



Albatros
3.2.3

Číslicové řízení



Tecnologie e Prodotti per l'Automazione

Tato dokumentace je majetkem T.P.A S.r.l.
Její duplikace bez autorizace je zakázána.
Společnost si vyhrazuje právo na změnu jejího obsahu v
kterémkoli okamžiku a bez předešlého upozornění.

Obsah

1	Úvod	1
1.1	Způsob použití příručky	1
1.2	Pracovní okna	1
2	Složení systému	2
2.1	Práva pro přístup do systému	2
2.2	Multijazyková podpora	2
2.3	Typická architektura systému	2
2.4	Logická organizace a konfigurace	4
2.5	Zařízení	5
3	Grafické zobrazení	7
3.1	Použití Grafického Zobrazení	7
3.2	Zásahy do Grafického Zobrazení	7
3.3	Zásahy do Zařízení	7
3.4	Pohyb Jednotek pohybu v osách v Manuálním režimu	7
4	Technologické Parametry a Nástroje	9
4.1	Okno Technologické Parametry	9
4.2	Okno Parametry Nástrojů	10
5	Diagnostika	11
5.1	Okno Diagnostiky	11
5.2	Složení okna Diagnostiky	11
5.3	Znázornění Zařízení	11
5.4	Interakce se Zařízeními	12
5.5	Seznam tlačítek pro pohyb uvnitř struktury ve formě stromu	12
5.6	Korektory linearity	13
5.7	Panel kalibrace jednotek pohybu v osách	13
6	Chyby a Zprávy	16
6.1	Úvod	16
6.2	Chyby systému	17
6.2.1	Chyby vytvořené správou jednotek pohybu v osách	17
	1 NázevJednotkyPohybuvOse: Chybné připojení snímače impulzů	17
	2 NázevJednotkyPohybuvOse: Nedokončený pohyb	17
	3 NázevJednotkyPohybuvOse: Chyba servořízení	17
	4 NázevJednotkyPohybuvOse: Překročení kladného limitu	17
	5 NázevJednotkyPohybuvOse: Překročení záporného limitu	18

	10 NázevJednotkyPohybuvOse: Provedení v reálném čase (Real-Time) je rychlejší než vytváření profilu	18
6.2.2	Chyby vytvořené správou vzdálených VST./VÝST.	18
	2049 Přijímač číslo: Chybná konfigurace	18
	2050 Přijímač číslo: Odpojený	18
	2051 Přijímač číslo: Znovu připojený	18
	2052 Přijímač číslo: Chyba při opětovném čtení nepřipojeného výstupu číslo ČísloVýstupu	18
	2054 Přijímač číslo: Chybný druh	19
	2055 Přijímač číslo: Inicializován	19
	2056 Přijímač číslo: Chyba napájení +24 V=	19
	2057 Chyba napájení GreenBUS	19
	2058 Přijímač číslo: Chyba opětovného čtení DruhZařiz NázevZařiz	19
	2059 Nezdařila se zkouška paměti s duálním portem vysílače	19
	2060 Chyba inicializace vysílače	19
	2061 Výskyt chyby během přenosu firmwaru do vysílače	20
	2062 Výskyt chyby během přenosu konfigurace do vysílače	20
	2063 Výskyt chyby během přenosu konfigurace do přijímače	20
	2064 Přijímač číslo: Nekompatibilní verze firmwaru	20
	2065 Přijímač číslo: Chyba asynchronní komunikace	20
	2066 Přijímač číslo: Všeobecná chyba	20
	2067 Přijímač číslo: Chyba během přenosu konfigurace	21
	2068 Přijímač číslo: Interní chyba č. číslochyby	21
	2069 Přijímač číslo: Chyba napájení +24 V= pracovního stolu číslo	21
6.2.3	Chyby vytvořené správou MECHATROLINK-II	21
	2308 Karta ČísloKarty: Inicializace se nezdařila v důsledku nesprávného nastavení parametru konfigurace	21
	2341 Karta ČísloKarty: Číslo servopohonů překračuje maximální dovolený počet	21
	2342 Karta ČísloKarty: Adresa hardwaru servopohonu Servo překračuje maximální dovolenou hodnotu	22
	2349 Karta ČísloKarty: Servopohon Servo není připojen	22
6.2.4	Chyby vytvořené správou CanBUS	22
	2761 Uzel číslo: Odpojený	22
	2762 Uzel číslo: Znovu připojený	22
	2763 Chyba způsobená chybějícím přenosem	22
	2764 Uzel číslo: Chyba způsobená chybějícím příjmem	22
	2765 Uzel číslo: Inicializovaný	23
	2766 Stav poruchy na rozhraní sběrnice CAN	23
	2767 Chyba ztráty stavu CANopen	23
	2768 Uzel číslo: Chyba způsobená chybějícím příjmem PDO	23
	2769 Uzel číslo: Chyba příjmu nekonfigurovaného uzlu	23
	2770 Uzel číslo: Chybná konfigurace	23
	2771 Uzel číslo: Chyba komunikace SDO	23
	2772 Časový limit cyklu sběrnice CAN dotazování uzlů	23
	3073 Uzel číslo: Chyba Nouzového stavu č. číslochyby	24
	3074 Uzel číslo: Všeobecná chyba sběrnice CAN č. číslochyby	24
	3088 Karta CAN číslo: Uzel ČísloUzlu: Chyba komunikace SDO č. číslochyby - popis	24
6.2.5	Chyby generované provozní EtherCAT sběrnice	24
	3329 Chyba inicializace soketu komunikace	24
	3330 Chyba během skenování sítě EtherCAT	24
	3331 Chyba v konfiguraci poštovní schránky vysílání	24
	3332 Chyba v konfiguraci poštovní schránky příjmu	25
	3333 Karta EtherCAT číslo: Chyba v druhu rozšíření uzlu ČísloUzlu	25
	3334 Chyba ve fázi konfigurace PDO	25
	3335 Uzel ČísloUzlu - Alarm (ČísloChyby)	25
	3336 Karta EtherCAT číslo: Chybný počet rozšíření uzlu ČísloUzlu je chybný	26
	3337 Karta EtherCAT: Uzel ČísloUzlu odpojený	26
	3338 Karta EtherCAT: Uzel ČísloUzlu znovu připojený	27

	3340 Karta EtherCAT: Uzel ČísloUzlu neodpověděl na žádost (Kód)	27
	3341 Karta EtherCAT: Uzel ČísloUzlu neexistuje	27
	3342 Odpojený kabel	27
	3343 Karta EtherCAT číslo: Uzel ČísloUzlu nepřechází do stavu SAFE-OPERATIONAL (Kód)	27
	3344 Karta EtherCAT číslo: Uzel ČísloUzlu nepřechází do stavu OPERATIONAL (Kód)	27
	3345 Karta EtherCAT: Nestabilní komunikace	28
	4400 Příliš mnoho aktivních jednotek pohybu v ose ve FASTREAD (Funkce:NázevFunkce řádek:ČísloŘádku)	28
6.2.6	Chyba vytvořená inicializací	28
	769 Chybná konfigurace softwaru	28
	770 Chybné nakonfigurované číslo IRQ	28
	772 Chyba při čtení části vyrovnávací paměti během inicializace	28
	773 Bylo dosaženo maximálního počtu konfigurovatelných jednotek pohybu v osách	28
	774 Nebylo zahájeno provádění v reálném čase (Real-Time) jednotek pohybu v osách	29
	775 Nedostatečná doba pro provedení GPL	29
	776 Nadměrná doba provádění v reálném čase (Real-Time)	29
	777 Uplynul časový limit sledovacího zařízení	29
	778 Hlavní část kódu (Main) firmwaru je zablokována	29
	1025 Karta ČísloKarty: Neodpovídá na příkazy	29
	1026 Karta ČísloKarty: Chyba při odesílání firmwaru na osovou kartu	29
	1028 Karta ČísloKarty: Není přítomen firmware	30
	1029 Karta ČísloKarty: Zablokovaná hlavní část kódu (MAIN)	30
	1031 Karta ČísloKarty: Chyba inicializace	30
	1032 Karta ČísloKarty: Nezdařila se zkouška paměti s duálním portem	30
	1033 Karta ČísloKarty: Neprobíhá provádění spouštěcího kódu (Boot) firmwaru	30
	1035 Karta ČísloKarty: Nepřítomná	30
	1037 Karta ČísloKarty: Nezdařilo se otevření paměti s duálním portem	31
	1039 Karta ČísloKarty: Uplynul časový limit sledovacího zařízení	31
	1040 Karta ČísloKarty: Chyba napájení +24 V=	31
	1047 Karta ČísloKarty: Nedovolená konfigurace softwaru	31
	1052 Karta ČísloKarty: Provádí se spouštěcí kód (Boot)	31
	1053 Karta ČísloKarty: Uplynul časový limit sledovacího zařízení jednotek pohybu v osách	31
	1055 Uplynula kontrolní doba („Watchdog“) na karte ČísloKarty	31
	1056 Karta ČísloKarty: Chyba napájení rozhraní sběrnice CAN	32
	1057 Karta ČísloKarty: Interní chyba číslo ČísloChyby	32
6.2.7	Chyby vytvořené správou paměti	32
	1281 Chyba přidělení paměti v oblasti haldy	32
	1286 Chyba správy haldy	32
	1287 Příliš mnoho přidělení paměti z haldy	32
	1289 Chyba při vytváření globálních proměnných	32
	1290 Chyba rozměru stálých proměnných	32
	1291 Chyba rozměru proměnných jen pro čtení	33
6.2.8	Chyby vytvořené poruchami	33
	1559 Trasy zarážky	33
	1569 Neplatný operační kód mikroprocesoru	33
	1586 Hodnota celého čísla (INTEGER) vydělená nulou	33
	1600 Přetečení ve výsledku operace s plovoucí desetinnou čárkou	33
	1601 Podtečení ve výsledku operace s plovoucí desetinnou čárkou	33
	1602 Neplatný argument pro operaci s plovoucí desetinnou čárkou	34
	1603 Hodnota s plovoucí desetinnou čárkou vydělená nulou	34
	1604 Nepřesný výsledek operace s plovoucí desetinnou čárkou	34
	1605 Byla použita chybná hodnota s plovoucí desetinnou čárkou	34
	1728 Byl zaznamenán pokus o přístup na neplatnou adresu	34
	1735 Všeobecná výjimka	34

1736	Údaje nejsou v souladu	34
1801	Alarm teploty	35
1802	Alarm ventilátoru	35
1803	Frekvence CPU je nestabilní	35
6.2.9	Chyby vytvořené instrukcemi GPL	35
4097	Zařízení DruhZařízení NázevZařízení není nakonfigurováno	35
4098	Globální proměnná NázevProměnné neexistuje	35
4099	Funkce NázevFunkce nebyla nalezena	35
4101	Nevyhovující správa jednotky pohybu v ose NázevJednotkyPohybuOse	35
4105	Danou instrukci nelze provést na jednotce pohybu v ose NázevJednotkyPohybuOse	36
4106	Dálkové zařízení přiřazené k jednotce pohybu v ose v krokovém režimu NázevJednotkyPohybuOse není připojeno	36
4107	Instrukce SYSOK s chybnými argumenty	36
4108	Jednotka pohybu v ose NázevJednotkyPohybuOse: Finální cílová poloha mimo softwarové rozmezí	36
4110	Chybná rychlost	36
4111	Záporné zrychlení jednotky pohybu v ose NázevJednotkyPohybuOse	37
4112	Záporné zpomalení jednotky pohybu v ose NázevJednotkyPohybuOse	37
4114	Jednotka pohybu v ose NázevJednotkyPohybuOse: Neprovedené vynulování rychlého vstupu	37
4115	Jednotka pohybu v ose NázevJednotkyPohybuOse: Nenalezená značka nuly	37
4353	Neznámá instrukce operačního kódu (Funkce:NázevFunkce řádek:ČísloŘádku)	37
4354	Chybná matematická operace (Funkce:NázevFunkce řádek:ČísloŘádku)	37
4355	Chybná adresa matrice nebo vektoru (Funkce:NázevFunkce řádek:ČísloŘádku)	38
4356	Instrukce RET nebyla vyvolána z CALL (Funkce:NázevFunkce řádek:ČísloŘádku)	38
4357	Neexistující místní proměnná (Funkce:NázevFunkce řádek:ČísloŘádku)	38
4358	Neexistující návěstí skoku (Funkce:NázevFunkce řádek:ČísloŘádku)	38
4359	Chybný argument makra (Funkce:NázevFunkce řádek:ČísloŘádku)	38
4360	Chyba v přidělení paměti během provádění (Funkce:NázevFunkce řádek:ČísloŘádku)	39
4361	Příliš mnoho aktivních úkolů (Funkce:NázevFunkce řádek:ČísloŘádku)	39
4362	Chybný formát matrice (Funkce:NázevFunkce řádek:ČísloŘádku)	39
4363	Příliš mnoho aktivních instrukcí ONINPUT (Funkce:NázevFunkce řádek:ČísloŘádku)	39
4364	Jednotka pohybu v ose je již obsazena v místním odkazu (Funkce:NázevFunkce řádek:ČísloŘádku)	39
4365	Instrukce ONINPUT byla aktivována na stejném vstupu (Funkce:NázevFunkce řádek:ČísloŘádku)	40
4366	Příliš mnoho aktivních instrukcí ONFLAG (Funkce:NázevFunkce řádek:ČísloŘádku)	40
4367	Instrukce ONFLAG byla aktivována na stejném PŘÍZNAKU (Funkce:NázevFunkce řádek:ČísloŘádku)	40
4368	Pokus o zápis proměnné určené pouze ke čtení - ReadOnly (Funkce:NázevFunkce řádek:ČísloŘádku)	40
4369	Příliš mnoho aktivních jednotek pohybu v osách typu master (Funkce:NázevFunkce řádek:ČísloŘádku)	40
4370	Příliš mnoho aktivních jednotek pohybu v osách typu slave (Funkce:NázevFunkce řádek:ČísloŘádku)	40
4372	Chybné použití instrukce (Funkce:NázevFunkce řádek:ČísloŘádku)	41
4373	Nelze provést čtení rychlosti posuvu (Feed rate) (Funkce:NázevFunkce řádek:ČísloŘádku)	41
4374	Příliš mnoho prováděných instrukcí typu IPC (Funkce:NázevFunkce řádek:ČísloŘádku)	41
4375	FASTREAD provedena na jednotkách pohybu v osách odlišných karet (Funkce:NázevFunkce řádek:ČísloŘádku)	41
4378	Neaktivovaná instrukce (Funkce:NázevFunkce řádek:ČísloŘádku)	41
4379	Instrukce nepoužitelná ve funkcích spuštěných přerušením - Interrupt (Funkce:NázevFunkce řádek:ČísloŘádku)	42
4380	Příliš mnoho požadavků na zápis do oblasti vyrovnávací paměti (Funkce:NázevFunkce řádek:ČísloŘádku)	42

	4381 Není možné používat sériové vedení, které ještě nebylo otevřeno (Funkce:NázevFunkce řádek:ČísloŘádku)	42
	4382 Není možné otevřít sériové vedení, které je již otevřeno (Funkce:NázevFunkce řádek:ČísloŘádku)	42
	4383 Byla zaznamenána snaha o otevření příliš velkého počtu pomocných procesů (Funkce:NázevFunkce řádek:ČísloŘádku)	42
	4384 Pomocný proces není v oběhu (Funkce:NázevFunkce řádek:ČísloŘádku)	42
	4385 Byla zaznamenána snaha o otevření pomocného procesu z jiného úkolu (Funkce:NázevFunkce řádek:ČísloŘádku)	43
	4391 Chyba během aktivace SYSOK (Funkce:NázevFunkce řádek:ČísloŘádku)	43
	4394 Příliš mnoho chyb cyklu (Funkce:NázevFunkce řádek:ČísloŘádku)	43
	4395 Příliš mnoho hlášení (Funkce:NázevFunkce řádek:ČísloŘádku)	43
	4397 Přetečení zásobníku (Funkce:NázevFunkce řádek:ČísloŘádku)	43
	4398 Podtečení zásobníku (Funkce:NázevFunkce řádek:ČísloŘádku)	43
	4399 Parametr mimo rozsah (Funkce:NázevFunkce řádek:ČísloŘádku)	43
	4865 Chybí stanovení stroje pro interpolaci (G216 nebo G217)	44
	4866 Chybí stanovení indexů konfigurace stroje při uchopení (M6)	44
6.2.10	Chyby vytvořené ovladačem komunikací CNCTPA	44
	16385 Odpojený modul	44
	16386 Připojený modul	44
	16387 Znovu připojený modul	44
	16388 Inicializovaný modul	44
	16389 Modul přerušil spojení	44
	16641 Řídící firmware neodpovídá na příkazy	45
	16642 TpaSock neodpovídá na příkazy	45
	16643 Operační systém neumožňuje používat RTX	45
	16645 Chyba při odesílání kódu firmwaru	45
	16646 Nebylo možné znovu spustit kód firmwaru	45
	16897 Nebyl nainstalován RTX	45
	16898 Uživatel nedisponuje právy správce	46
	16899 Rozměr RAM modulu je chybný	46
	16900 IP adresa modulu je chybná	46
	16901 Modul je již odpojen od jiného výrobního zařízení	46
	16902 Modul není nakonfigurován	46
	16903 Nastavení firewallu brání komunikaci	46
	16904 Síťová karta není k dispozici nebo je vyřazena z činnosti	46
	16905 Chybí kód firmwaru řízení	47
	16906 Verze RTX není kompatibilní s kódem firmwaru řízení	47
	16907 Verze operačního systému není kompatibilní s kódem firmwaru řízení	47
	17153 DruhKarty: Chybí kód firmwaru vysílače GreenBUS	47
	17154 DruhKarty: Chybí část kódu firmwaru vysílače GreenBUS	47
	17155 DruhKarty: Chyba v odesílání spouštěcího kódu (Bootstrap) vysílače GreenBus	47
	17156 DruhKarty: Chyba v odesílání hlavní části kódu (Main) vysílače GreenBus	48
	17157 DruhKarty: Chybí spouštěcí kód (Bootstrap)	48
	17158 DruhKarty: Chybí hlavní část kódu (Main)	48
	17159 DruhKarty: Chyba v odesílání spouštěcího kódu (Bootstrap)	48
	17160 DruhKarty: Chyba v odesílání hlavní části kódu (Main)	48
	17409 Nebylo možné odeslat pomocný spustitelný soubor	48
	17410 Nebylo možné provést pomocný spustitelný soubor	48
	17667 NázevDLL: Nebylo možné znovu spustit kód firmwaru	49
	17668 NázevDLL: Nebylo možné získat ukazatel do sdílené RAM	49
	17921 Nebylo možné odeslat NODETPA	49
	17922 NODETPA nebyl znovu uveden do činnosti	49
	17923 NODETPA není spuštěn	49
	18177 NODETPA se pokusila o přístup na neplatnou adresu	49
6.3	Obecné signalizace	50
6.3.1	Albatros zahajuje provádění	50

6.3.2	Albatros končí provádění	50
6.3.3	Počítač vstupuje do pohotovostního režimu	50
6.3.4	Počítač vychází z pohotovostního režimu	50
6.3.5	Vypnutí počítače	50
6.3.6	Aktuální úroveň přístupu	50
6.3.7	Aktualizace softwaru modulů	50
6.3.8	Odeslání konfigurace do modulů	50
7	Konfigurace Systému	51
7.1	Úvod	51
7.2	Konfigurace Zařízení	51
7.2.1	Úvod	51
7.2.2	Všeobecné zařízení	51
7.2.3	Digitální výstup	52
7.2.4	Analogový vstup	52
7.2.5	Osa	52
	Základní údaje	52
	Parametry pohybu	52
	Parametry interpolace	53
	Ostatní parametry	53
	Referenční parametry	53
	Úrovně přístupu	54
	Provázání Jednotek pohybu v osách	54
	Korektory linearity	54
7.3	Logická konfigurace	55
7.3.1	Konfigurace výrobního zařízení	55
7.3.2	Konfigurace skupin	55
7.4	Fyzická konfigurace	57
7.4.1	Konfigurace systému	57
7.4.2	Konfigurace Hardwaru	57
	Přednastavené konfigurace	58
	Konfigurace uzlu na sběrnici TPA	58
	Konfigurace uzlu sběrnice CAN	60
	Kontrolní karta sběrnice	60
	Uzel CAN	60
	Vložení nového uzlu	60
	Konfigurace uzlu	61
	Vlastností řízení EtherCAT v softwaru Albatros	61
	Výchozí předpoklady	61
	Konfigurace hardwaru sítě EtherCAT	61
	Popis PDO	62
	Změna PDO pohonu	62
	Doplňkové PDO	64
	Automatické získání uzlů EtherCAT	64
7.4.3	Konfigurace Virtuální-Fyzické	65
7.4.4	Mapy kabeláže	66
7.5	Seznam tlačítek pro pohyb uvnitř struktury ve formě stromu	66
8	Vývojové nástroje	67
8.1	Editor GPL	67
8.1.1	Funkce editoru GPL	67
	Použití pravidelných výrazů	68
8.1.2	Vložit hlášení	69

8.1.3	Šifrování	70
8.1.4	Seznam dostupných tlačítek rychlé volby	71
8.2	Knihovny	72
8.3	Odlad'ování	73
8.3.1	Ladicí program	73
8.3.2	Provádění úkolu	73
8.3.3	Všechny úkoly	74
8.3.4	Vyvolání funkcí	74
8.3.5	Body přerušení	74
8.3.6	Obsah proměnné	75
8.3.7	Seznam dostupných tlačítek rychlé volby	75
8.4	Inicializace řízení	76
8.4.1	Síťová připojení	76
8.4.2	Diagnostika hardwaru	76
	Topologie sítě EtherCAT	76
	Zobrazování a změna objektů v uzlech	76
8.5	Zkouška	77
8.5.1	Uložit do paměti globální proměnnou	77
8.5.2	Provést funkci	78
8.5.3	Importování hlášení	78
8.5.4	Poznámka uživatele do souboru sestavy alarmů	80
8.6	Nástroje	80
8.6.1	Přizpůsobit...	80
8.7	Prohlížeč	81
8.7.1	Prohlížeč	81
8.7.2	Vyhledat Identifikátor	82
8.7.3	Seznam dostupných tlačítek rychlé volby	82
9	Doplňkové programy	83
9.1	XConfMerge: program pro sloučení konfiguračního souboru	83
9.2	XParMerge: program pro sloučení dvou souborů parametrizování	84
10	Jazyk GPL	85
10.1	Základní principy	85
10.1.1	Úvod a jazyk GPL	85
10.1.2	Konvence a terminologie	85
10.1.3	Proměnné	87
	Druhy dat	87
	Konverze dat	89
	Deklarování a viditelnost proměnných	89
	Modifikátory	90
	Přiřazení RANGE	90
	Práva na Čtení / Zápis	91
10.1.4	Konstanty	91
	Přednastavené konstanty s hodnotou přiřazenou	92
	Přednastavené konstanty s hodnotou přiřazenou při spuštění softwaru Albatros	93
10.1.5	Klíčová slova	93
10.1.6	Funkce	94
10.1.7	Parametry druhu zařízení	96
10.1.8	Zpracování více úloh (Multitasking)	96
10.1.9	Komunikace	98

10.1.10	Proměnné určené k použití při programování	98
10.1.11	Jednotky pohybu v osách	98
10.1.12	Korektory linearity	101
10.1.13	Správa hlášení v jazyce	101
10.1.14	Správa chyb systému	101
10.2	Speciální funkce	102
10.2.1	Uživatelské přizpůsobení pohybu jednotek pohybu v osách	102
10.2.2	Standardní funkce pohybu a kalibrace	104
10.2.3	Funkce OnUIEnd#	107
10.2.4	Funkce OnUIPlugged#	107
10.2.5	Funkce OnUIUnPlugged#	107
10.3	Instrukce	108
10.3.1	Konvence	108
10.3.2	Druh instrukcí jazyka GPL	108
10.3.3	Vstup/Výstup	114
	GETFEED	114
	INPANALOG	114
	INPFLAGPORT	115
	INPPORT	115
	MULTIINPPORT	115
	MULTIOUTPUTPORT	115
	MULTIRESETFLAG	116
	MULTIRESETOUT	116
	MULTISETFLAG	116
	MULTISETOUT	116
	MULTIWAITFLAG	117
	MULTIWAITINPUT	117
	OUTANALOG	118
	OUTFLAGPORT	118
	OUTPORT	118
	RESETFLAG	118
	RESETOUT	118
	SETFLAG	119
	SETOUT	119
	WAITFLAG	119
	WAITINPUT	119
	WAITPERSISTINPUT	120
10.3.4	Jednotky pohybu v osách	120
	CHAIN	120
	CIRCABS	121
	CIRCINC	122
	CIRCLE	123
	COORDIN	123
	DISABLECORRECTION	124
	EMERGENCYSTOP	125
	ENABLECORRECTION	125
	ENDMOV	126
	FASTREAD	126
	FREE	127
	HELICABS	127
	HELICINC	128
	JERKCONTROL	128
	JERKSMOOTH	129
	LINEARABS	129
	LINEARINC	130
	MOVABS	130

MOVINC	131
MULTIABS	132
MULTIINC	133
NORMAL	133
RESRIFLOC	133
SETINDEXINTERP	134
SETLABELINTERP	134
SETPFLY	134
SETPFLYCHAINSTRAT	135
SETPZERO	135
SETPZEROCHAINSTRAT	135
SETQUOTE	136
SETQUOTECHAINSTRAT	136
SETRIFLOC	136
SETTOLERANCE	137
START	138
STARTINTERP	138
STOP	138
SWITCHENC	139
WAITACC	139
WAITCOLL	140
WAITDEC	140
WAITREG	141
WAITSTILL	141
WAITTARGET	141
WAITWIN	141
Parametry Jednotek pohybu v osách	142
Čtení/Psaní	142
DEVICEID	142
GETAXIS	142
Pohybu z bodu do bodu	148
SETACC	148
SETDEC	148
SETDERIV	148
SETFEED	148
SETFEEDF	149
SETFEEDFA	149
SETINTEG	149
SETMULTIFEED	149
SETPROP	150
SETSLOPE	150
SETVEL	150
Interpolovaný pohyb	150
LOOKAHEAD	150
SETACCI	151
SETACCLIMIT	151
SETACCSTRATEGY	151
SETAXPARTYPE	152
SETCONTORNATURE	152
SETDECI	152
SETDERIVI	153
SETFEEDFAI	153
SETFEEDI	153
SETFEEDFI	153
SETINTEGI	154
SETPROPI	154
SETSLOPEI	154

	SETSLOWPARAM	154
	SETVELI	155
	SETVELILIMIT	155
	Koordinovaný pohyb	155
	SETFEEDCOORD	155
	SETOFFSET	157
	Zřetězený pohyb	157
	RATIO	157
	SETDYNRATIO	158
	Všeobecné parametry	158
	DYNLIMIT	158
	ENABLESTARTCONTROL	159
	NOTCHFILTER	159
	RESLIMNEG	159
	RESLIMPOS	160
	SETADJUST	160
	SETBACKLASH	160
	SETBIGWINFACTOR	162
	SETDEADBAND	162
	SETENCLIMIT	163
	SETINDEXEN	163
	SETINTEGTIME	163
	SETIRMPP	163
	SETLIMNEG	164
	SETLIMPOS	164
	SETMAXER	164
	SETMAXERNEG	164
	SETMAXERPOS	165
	SETMAXERTYPE	166
	SETPHASESINV	167
	SETREFINV	167
	SETRESOLUTION	167
10.3.5	Počítadla	168
	DECOUNTER	168
	INCOUNTER	168
	SETCOUNTER	168
10.3.6	Časovače	168
	HOLDTIMER	168
	SETTIMER	168
	STARTTIMER	169
10.3.7	Proměnné, vektory a matice	169
	CLEAR	169
	FIND	169
	FINDB	170
	LASTELEM	170
	LOCAL	170
	MOVEMAT	171
	PARAM	171
	SETVAL	171
	SORT	172
10.3.8	Řetězce	172
	ADDSTRING	172
	CONTROLCHAR	173
	LEFT	173
	LEN	173
	MID	173
	RIGHT	174

	SEARCH	174
	SETSTRING	174
	STR	175
	VAL	175
10.3.9	Komunikace	175
	CLEARRECEIVE	175
	COMCLEARRXBUFFER	175
	COMCLOSE	175
	COMGETERROR	176
	COMGETRXCOUNT	176
	COMOPEN	176
	COMREAD	177
	COMREADSTRING	177
	COMWRITE	177
	COMWRITESTRING	178
	RECEIVE	178
	SEND	182
	SENDIPC	188
	WAITIPC	188
	WAITRECEIVE	189
10.3.10	Matematika	189
	ABS	189
	ADD	189
	AND	190
	ARCCOS	190
	ARCSIN	190
	ARCTAN	190
	COS	191
	DIV	191
	EXP	191
	EXPR	191
	LOG	193
	LOGDEC	193
	MOD	193
	MUL	193
	NOT	194
	OR	194
	RANDOM	194
	RESETBIT	195
	ROUND	195
	SETBIT	196
	SHIFTL	196
	SHIFTR	197
	SIN	199
	SQR	199
	SUB	199
	TAN	200
	TRUNC	200
	XOR	200
10.3.11	Multitasking	201
	ENDMAIL	201
	ENDREALTIMETASK	201
	ENDTASK	201
	GETPRIORITYLEVEL	201
	GETREALTIME	202
	GETREALTIMECOUNT	202
	HOLDTASK	202

	RESUMETASK	202
	SENDMAIL	202
	SETPRIORITYLEVEL	203
	STARTREALTIMETASK	203
	STARTTASK	204
	STOPTASK	204
	WAITMAIL	204
	WAITTASK	204
10.3.12	Správa Toku	205
	CALL	205
	DELONFLAG	205
	DELONINPUT	205
	FCALL	205
	FOR/NEXT	206
	FRET	206
	GOTO	207
	IF/IFVALUE/IF-THEN-ELSE	207
	IFACC	208
	IFAND	208
	IFBIT	209
	IFBLACKBOX	209
	IFCHANGEVEL	210
	IFCOUNTER	210
	IFDEC	211
	IFDIR	211
	IFERRAN	211
	IFERROR	212
	IFFLAG	213
	IFINPUT	213
	IFMESSAGE	214
	IFOR	214
	IFOUTPUT	215
	IFQUOTER	215
	IFQUOTET	216
	IFRECEIVED	216
	IFREG	217
	IFSAME	217
	IFSTILL	217
	IFSTR	218
	IFTARGET	218
	IFTASKHOLD	219
	IFTASKRUN	219
	IFTIMER	219
	IFVEL	220
	IFWIN	220
	IFXOR	221
	ONERRSYS	221
	ONFLAG	222
	ONINPUT	222
	REPEAT/ENDREP	223
	RET	223
	SELECT	223
	TESTIPC	224
	TESTMAIL	225
10.3.13	Různé	225
	CLEARERRORS	225
	CLEARMESSAGES	225

	DEFMSG	226
	DELAY	227
	DELError	227
	DELMESSAGE	227
	ERROR	228
	IFDEF/ELSEDEF/ENDDEF	229
	MESSAGE	231
	SYSFAULT	233
	SYSOK	233
	TYPEOF	233
	WATCHDOG	233
10.3.14	MECHATROLINK-II	234
	MECCOMMAND	234
	MECGETPARAM	235
	MECGETSTATUS	235
	MECSETPARAM	237
10.3.15	Standardní sběrnice pole	237
	AXCONTROL	237
	AXSTATUS	239
	CNBYDEVICE	240
	READDICTIONARY	240
	WRITEDICTIONARY	241
10.3.16	EtherCAT	241
	ACTIVATEMODE	241
	ECATGETREGISTER	241
	ECATSETREGISTER	242
	GETPDO	242
	SETEOE	242
	SETPDO	243
10.3.17	Karta TMSbus s řízením CAN	243
	GETCNSTATE	243
	GETSDOERROR	243
	GETMNSTATE	243
	RECEIVEPDO	244
	SENDPDO	244
	SETNMTSTATE	244
10.3.18	Simulace	244
	DISABLE	244
	DISABLEFORCEDINPUT	245
	ENABLE	245
	ENABLEFORCEDINPUT	245
	RESETFORCEDINPUT	245
	SETFORCEDANALOG	246
	SETFORCEDINPUT	246
	SETFORCEDPORT	246
10.3.19	BlackBox	246
	ENDBLACKBOX	247
	PAUSEBLACKBOX	247
	STARTBLACKBOX	247
10.3.20	ISO	248
	ISOG0	248
	ISOG1	248
	ISOG9	249
	ISOG90	249
	ISOG91	249
	ISOG93	250
	ISOG94	250

	ISOG216	250
	ISOG217	251
	ISOM2	251
	ISOM6	251
	ISOSETPARAM	252
	KINEMATICEXPR	254
10.3.21	Instrukce, které již nejsou k dispozici	255
10.3.22	Instrukce nepoužitelné při přerušení (interrupt)	255
10.4	Příklady	257
10.4.1	Vynulování při Přerušení (Interrupt)	257
10.4.2	Server pohybu jednotek pohybu v osách	258
10.4.3	Cyklus hlavní části kódu (Main) se správou chyb	260
10.4.4	Operace na řetězcích	261
10.4.5	Sekvenční / Paralelní Provedení	262
10.4.6	Nulovací Rutina	262
10.4.7	Pohyby ISO	263

1 Úvod

1.1 Způsob použití příručky

Tato nápověda popisuje funkce číslíkového řízení Albatros.

Nápověda byla nastavena tak, aby vedla obsluhu při pochopení systému a jeho použití.

K významným bodům každé sekce nápovědy patří:

- okna a nástroje softwaru Albatros.
- popis typické Architektury systému Albatros.
- způsob zobrazování zařízení a provádění zásahů do nich s funkcemi manuálního ovládání a diagnostiky při využití Grafického zobrazení.
- způsob zobrazování a modifikace Technologických a Geometrických Parametrů a Parametrů Nástrojů.
- způsob zobrazování Zařízení a provádění zásahů do nich s funkcemi manuálního ovládání a diagnostiky

Abychom se vyhnuli nadměrnému zatížení této nápovědy, odkazujeme čtenáře na návody Operačního Systému Windows kvůli podrobnějšímu pochopení mechanismů používání *myši, menu, lišt, panelů a řádků* a všech klasických operačních funkcí operačního systému Windows.

1.2 Pracovní okna

Pracovní okna jsou různého druhu v závislosti na operaci, která má být provedena, a mohou být otevřena všechna současně.

Okna jsou rozdělena do níže uvedených druhů:

Okno

Hlavní

[Grafické zobrazení](#)

[Technologické Parametry](#)

[Parametry Nástrojů](#)

[Diagnostika](#)

[Chyba systému](#)

[Konfigurace Systému](#)

Popis

okno softwaru Albatros. Umožňuje volání funkcí a slouží jako rámec pro další okna, jejichž obsah závisí na specifické aplikaci, kterou zastupují.

obsahuje grafické znázornění stroje nebo jeho částí, přičemž umožňuje na ně působit.

umožňuje zobrazování a změnu technologických parametrů.

umožňuje zobrazování a změnu parametrů nástrojů.

umožňuje zobrazení stavu zařízení a v případech, kdy je to dovoleno, také zásahy do nich.

okno se seznamem posledních vzniklých chyb systému. Je možné zobrazovat také chyby cyklu a hlášení.

umožňuje zobrazování a změnu logických a fyzických zařízení stroje.


2 Složení systému

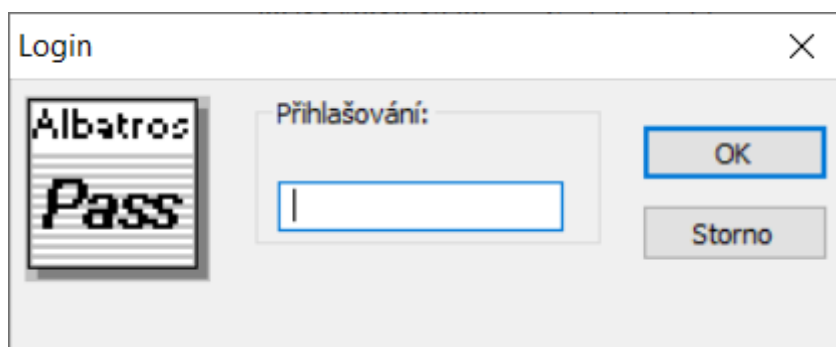
2.1 Práva pro přístup do systému

Albatros počítá se čtyřmi úrovněmi přístupu do systému:

- **Uživatel:** Jedná se o úroveň s největšími omezeními přístupu. Neumožňuje změnit žádné z nastavení týkajících se zařízení. Tato úroveň se používá pro provedení obrábění a pro běžné operace stroje. Při spuštění systému se automaticky aktivuje právě tato úroveň přístupu.
- **Servisní služba:** Jedná se o úroveň určenou k provádění běžné údržby stroje. Obsluha by měla mít možnost měnit některé parametry konfigurace, ale neměla by mít možnost provádět změny ve struktuře stroje.
- **Výrobce:** Jedná se o úroveň používanou pro konfiguraci výrobních zařízení a strojů. Na této úrovni se mohou provádět téměř všechny možné změny. Používají ji vývojoví pracovníci.
- **Tpa:** Jedná se o maximální úroveň přístupu do systému. Slouží k ochraně některých mimořádně choulostivých nastavení, jejichž změna vyžaduje hloubkovou znalost Albatros. Používá se velmi zřídka. Heslo pro přístup na tuto úroveň musí být dodáno přímo firmou TPA.

Pro přístup do systému s vyšší úrovní přístupu, než je úroveň Uživatele, nebo pro přepnutí systému na úroveň Uživatele po provedení změn s vyšší úrovní je třeba zadat odpovídající heslo.

Pro zobrazení okna, do kterého se zadává heslo, je třeba použít kombinaci tlačítek **Ctrl+*** (hvězdička). Alternativou může být použití **Hlavního panelu** Windows, který se nachází v pravé části a jehož součástí je i ikona . Kliknutím pravým tlačítkem na myši na uvedenou ikonu je možné zobrazit menu, ve kterém se objeví položka **Change pass level** (Změna úrovně přístupu). Okno, které se zobrazí, má níže uvedenou podobu:



Přihlašovací okno

Nyní je třeba zadat heslo a potvrdit jej stisknutím tlačítka **[OK]**. Namísto znaků, které tvoří heslo, budou zobrazeny znaky „*“, aby nikdo nemohl přečíst právě zadávané heslo. Po zadání hesla se ihned nacházíte na odpovídající úrovni přístupu. Za účelem potvrzení aktuální úrovně přístupu lze zvolit položku **Informace o Albatros** v menu ?.


Když je zadané heslo nesprávné, dojde k zobrazení chyby „Pozor! Chybné heslo!“.

2.2 Multijazyková podpora

Albatros podporuje zobrazování textu v různých jazycích.

Změna jazyka

Změna jazyka může být provedena na kterékoli [úrovni přístupu](#) do systému. Pro změnu volby jazyka musíte použít

kombinaci tlačítek **Ctrl + /** nebo klikněte na ikonu  v „**hlavním panelu**“ Windows.

V zobrazeném okně zvolte požadovaný jazyk a klikněte na tlačítko **[OK]**.

Změna jazyka nebude provedena okamžitě, ale při následujícím spuštění softwaru Albatros.

2.3 Typická architektura systému

Vzhledem k tomu, že mnoho aspektů v grafických zobrazeních i ve struktuře základních údajů stroje výrazně závisí na druhu Stroje, poskytuje tato příručka kromě některých informací všeobecného charakteru i popis složení typického systému jako zjednodušující příklad.

Reálná indikace, schémata a grafické stránky samotného systému závisí na specifické aplikaci, a proto za jejich přípravu odpovídá Výrobce obráběcího stroje.

Číslicové řízení Albatros je tvořeno dozorcujícím PC, které poskytuje rozhraní Obsluha-Stroj, a proměnlivým počtem modulů (od 1 do 16) pro řízení a kontrolu všech provozních zdrojů Obráběcího stroje nebo Výrobního zařízení.

Proto lze rozlišit dva druhy výrobního zařízení:

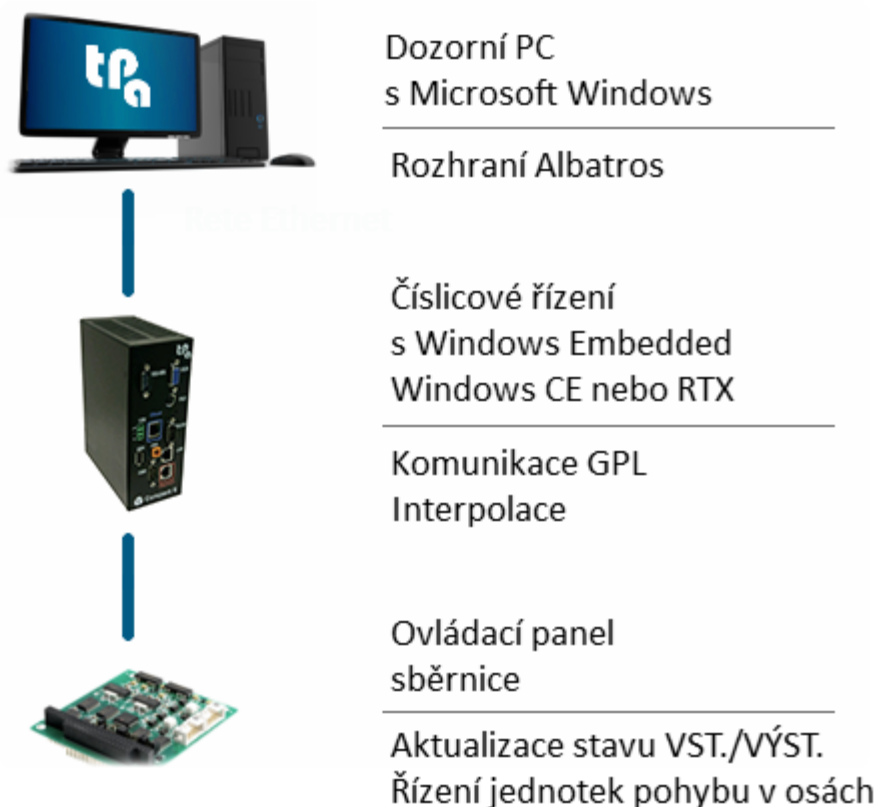
Monomodul , tvořený jediným modulem připojeným přímo ke sběrnici PC.

Multimodul , tvořený minimálně 1 a maximálně 16 moduly, který se obvykle používá pro aplikace na Výrobních zařízeních nebo Linkách tvořených více stroji; jednotka PC je v tomto případě fyzicky oddělena od modulů, které mohou být uloženy na odlišných místech linky nebo výrobního zařízení.

U obou těchto architektur jsou moduly tvořeny jednou nebo více kartami pro přímé řízení Jednotek pohybu v osách stroje a logickou správu vstupních/výstupních zařízení.

U verze monomodul jsou karty jednotek pohybu v osách nainstalovány přímo v průmyslovém PC (s displejem a klávesnicí nebo bez nich), připojeném k dozorcujícímu PC prostřednictvím sítě ethernet. Na následující obrázku je uvedeno schéma zapojení mezi dozorcujícím PC a vzdáleným modulem. Dále jsou na něm uvedeny hlavní činnosti prováděné jednotlivými komponenty.

Schéma připojení vzdáleného modulu



Inteligentní vzdálená zařízení ovládají VST./VÝST. zařízení a jednotky pohybu v osách (vzdálené TRS-AX) přímo na samotném stroji. Tato zařízení zajišťují čtení vedení Digitálních (ZAP / VYP) nebo Analogových vstupů a obnovení digitálních nebo analogových výstupů a jsou připojena k Modulům prostřednictvím sběrnice GreenBUS (sériová sběrnice RS485 - 1 Mbaud) sběrnice CAN a EtherCAT. Činnost Albatros ve stroji je chráněna přítomností hardwarového USB klíče, nakonfigurovaného firmou TPA.

2.4 Logická organizace a konfigurace

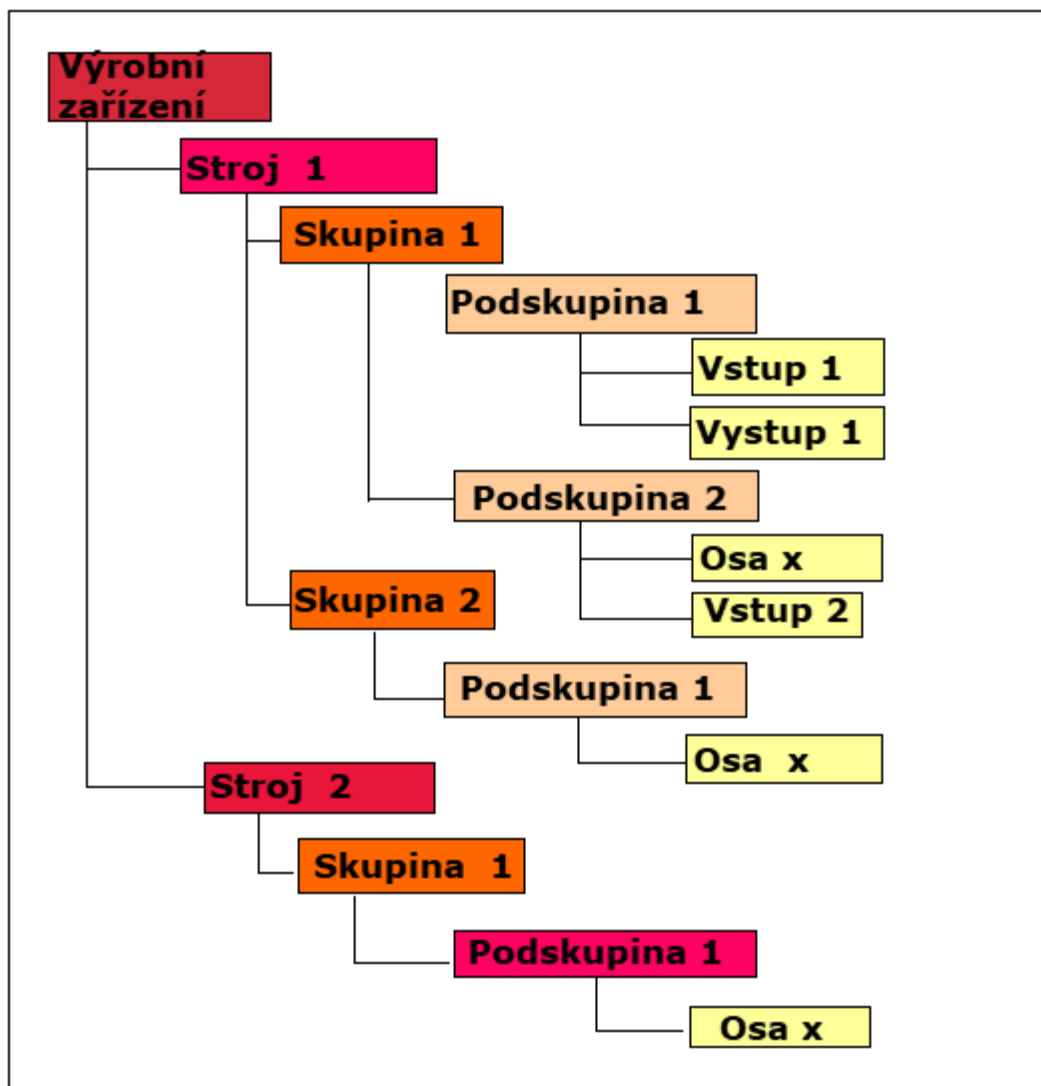
V systému Albatros je popisná struktura výrobního zařízení nebo jednotlivého obráběcího stroje zorganizována do technologického archivu s hierarchickou strukturou.

Toto nastavení odráží potřebu uchovat na úrovni parametrů konfigurace a režimu přístupu případnou modulární strukturu strojů, a to na základě její klasifikace z hlediska dynamického seskupení různých modulů, agregátů a zařízení, které lze aktivovat nebo vyřadit na základě jednotlivých možných uspořádání.

Na základě této logické struktury máme v tom nejobecnějším a nejvíce komplexním případě horní hierarchickou strukturu tvořenou z níže uvedených prvků:

- 1. Výrobní zařízení** jedná se o seskupení strojů. Považuje se za souhrn provozních prvků spravovaných číslicovým řízením. Výrobní zařízení je vždy přítomné, a to i v případě jediného stroje, a nesmí být nikdy zadefinováno explicitně.
- 2. Stroj** Z „logického“ hlediska se pojímá jako souhrn zařízení (jednotky pohybu v osách, časovače apod.) a kontrolních cyklů, tj. kódů napsaných v GPL, které implementují algoritmy pro řízení samotného stroje. Obecně se na stroji nachází vysoký počet zařízení, která jsou zorganizována do skupin
- 3. Skupiny** představují „obaly“, které umožňují zorganizovat součásti stroje podle logických kritérií. Například lze stanovit skupinu „jednotky pohybu v osách“, která obsahuje všechny jednotky pohybu v osách stroje, přepínače koncových spínačů, cykly, které provedou vynulování jednotek pohybu v osách, apod.
- 4. Podskupiny** představují další specializaci v rámci jedné skupiny. Například skupina „jednotky pohybu v osách“ může být rozdělena na „digitální jednotky pohybu v osách“ a „jednotky krokového pohybu v osách“.
- 5. Zařízení** představují nejnižší úroveň hierarchie. Jsou logickým zastoupením elektrických a mechanických součástí stroje a jsou nezávislé na podřízeném hardwaru.

Na níže uvedeném obrázku je schematicky znázorněna struktura hypotetického výrobního zařízení tvořeného dvěma stroji:



Příklad hierarchické organizace výrobního zařízení.

POZNÁMKA: Skupiny nemusí být nutně podrobeny dílčímu rozdělení na Podskupiny, a mohou být tvořeny přímo Zařízeními.




Pro přístup k některým funkčním blokům, jako je Diagnostika, Konfigurace systému, Parametrizace v případě výrobního zařízení se dvěma stroji, je třeba zvolit stroj, jehož údaje chcete zobrazit.





2.5 Zařízení

Zařízení lze seskupit do dvou kategorií: fyzická zařízení a logická zařízení. V systému jsou všechna zařízení odlišena prostřednictvím názvu, který identifikuje jejich použití.

Fyzická zařízení













Za fyzická zařízení se považují ta, která působí na elektrické nebo pneumatické součásti stroje nebo která zaznamenávají jejich stav, a patří k nim:

Symbol	Zařízení	Funkčnost
	<i>Digitální vstup</i>	zaznamenává „zapnutý“ nebo „vypnutý“ stav zařízení. Například bezpečnostní spínač ochranných dveří.
	<i>Digitální výstup</i>	slouží k aktivaci nebo zakázání zařízení, tedy k jeho uvedení do „zapnutého“ nebo „vypnutého“ stavu. Může se používat například pro řízení elektrického ventilu.
	<i>Analogový vstup</i>	zaznamenává hodnotu napětí na vstupu na odpovídající svorce. Například hodnotu vytvářenou tachymetrickým dynamem.

Symbol	Zařízení	Funkčnost
	<i>Analogový výstup</i>	produkuje výstupní napětí na odpovídající svorce. Může se používat například pro řízení frekvenčního měniče.
	<i>Port vstupů</i>	je tvořen 8 vedeními digitálních vstupů.
	<i>Port výstupů</i>	je tvořen 8 vedeními digitálních výstupů.
	<i>Jednotka pohybu v ose</i>	spravuje pohyb elektricky řízené jednotky pohybu v ose. Je možné provádět správu jednotek pohybu v osách různého druhu: s analogovou kontrolou, s digitální kontrolou, krokových motorů, jednotek pohybu v osách počítání (pouze čtení snímače impulzů).

Logická zařízení

Logická zařízení jsou prvky, které působí výhradně uvnitř pracovních programů a nemají fyzickou protihodnotu:

Symbol	Zařízení	Funkčnost
	<i>Časovač</i>	prvek počítání času. Jeho měrnou jednotkou je sekunda. Vyznačuje se rozlišením 4 ms. Může nabýt pouze kladných hodnot a jeho maximální vyjádřitelná hodnota je 8.589.934 sekund (přibližně 99 dnů) (při real-time s hodnotou 250 Hz). Jeho hodnota se ukládá do stálé paměti karty jednotek pohybu v osách.
	<i>Počítadlo</i>	prvek počítání operací. Může nabýt hodnoty v rozsahu od -2 147 483 648 do +2 147 483 647. Jeho hodnota se ukládá do stálé paměti karty jednotek pohybu v osách.
	<i>Bit příznaku</i>	prvek, který může nabýt hodnoty „zapnutý“ nebo „vypnutý“.
	<i>Přepínač příznaku</i>	jedná se o speciální příznaky, které mohou být přiřazeny některým tlačítkům, jako například příznak Startu.
	<i>Port příznaku</i>	je tvořen 8 vedeními bitů příznaků.
	<i>Proměnná</i>	globální proměnná typu <i>integer</i> (celé číslo) kódu GPL.
	<i>Proměnná</i>	globální proměnná typu <i>char</i> (znak) kódu GPL.
	<i>Proměnná</i>	globální proměnná typu <i>float</i> (hodnota s plovoucí desetinnou čárkou) kódu GPL.
	<i>Proměnná</i>	globální proměnná typu <i>double</i> (hodnota reálného čísla se zvýšenou přesností) kódu GPL.
	<i>Proměnná</i>	globální proměnná typu <i>string</i> (řetězec) kódu GPL.
	<i>Proměnná</i>	globální proměnná typu <i>array</i> (pole) kódu GPL.
	<i>Proměnná</i>	globální proměnná typu <i>matrix</i> (matrice) kódu GPL.

3 Grafické zobrazení

3.1 Použití Grafického Zobrazení

Během činnosti stroje je možné otevřít okno *Grafického zobrazení*, které umožňuje zobrazit stav nejvýznamnějších zařízení.

Informace přítomné na grafických zobrazeních se shodují s těmi, které jsou uvedeny v okně diagnostiky. Avšak zatímco v okně diagnostiky jsou informace představeny v podobě struktury ve formě stromu (která zahrnuje všechna zařízení stroje), grafická zobrazení umožňují představit informace grafickým způsobem (například zobrazením obrázku stroje a uvedením poloh jednotek pohybu v osách v jejich blízkosti). Dále umožňují zvolit nejvýznamnější informace a seskupit je do vedlejších stran, které může zobrazit uživatel.

3.2 Zásahy do Grafického Zobrazení

Pro diagnostické účely může Obsluha zvolit různé strany, které tvoří grafické zobrazení, a to *dvojklikem* myši na jednu ze dvou oblastí stroje, které jsou na obrázku ohraničeny čárkovaným obdélníkem a nazývají se také „horké oblasti“.

Za účelem nasměrování na „horkou oblast“ zařízení nebo jednotku pohybu v ose stačí přesunout kurzor myši na obrázek požadovaného objektu. Současně se ve Stavovém řádku zobrazí název zařízení, po kterém prochází kurzor myši.

Kurzor myši získává různé podoby v závislosti na druhu objektu, na který je nasměrován, přičemž takto uvádí úkony, které lze na objektu provést. Patří k nim:



lupa

, jedná-li se o „horkou oblast“



ruka

, jedná-li se o zařízení druhu výstup



textový kurzor

, jedná-li se o pole, ve kterém je možné nastavit hodnotu

3.3 Zásahy do Zařízení

Působení na zařízení se provádí nastavením kurzoru myši na požadované zařízení a dokončením postupu níže uvedeným způsobem (tento se mění v závislosti na druhu zařízení).

Způsob znázornění

Ikona zařízení

Úkon

nasměrovat kurzor a *kliknout*

Zařízení

Digitální výstup
Přepínač příznaku
Bit příznaku
Analogový výstup
Port výstupu
Port příznaku
Poloha jednotky pohybu v ose
Časovač
Počítadlo

Pole pro nastavení hodnoty

nasměrovat kurzor a *kliknout* a nastavit hodnotu




3.4 Pohyb Jednotek pohybu v osách v Manuálním režimu

Za účelem přístupu k funkci manuálního pohybu jednotek pohybu v osách je třeba disponovat příslušnými [přístupovými právy](#). Tato práva jsou přiřazena výrobcem stroje.

Pro interakci s jednou z jednotek pohybu v osách stačí *dvojkliknutí* myši v místě pole zobrazování poloh požadované jednotky pohybu v ose. Dojde tak k otevření okna pro Pohyb jednotky pohybu v ose. V případě Virtuálních, Krokových a Počítacích jednotek pohybu v ose bude okno obsahovat omezený počet informací. Například když je jednotka pohybu v ose druhu Počítání, budou zobrazeny pouze hodnoty Reálné Polohy a Rychlosti.



V okně se nacházejí dvě oblasti, které obsahují:





Oblast zobrazování

- Tři pole obsahující zobrazování *Reálné polohy* jednotky pohybu v ose [mm], její *Rychlosti* [m/min] a *Chyby smyčky* neboli sledování.
- Dvě tlačítka volby, která signalizují *Stav* jednotky pohybu v ose (*Free* = s otevřenou smyčkou, např. z důvodu chyby systému, *Normal* = s uzavřenou smyčkou, tj běžný stav kontroly polohy). Tato tlačítka umožňují také nastavit stav.
- Signalizace *Stavu* jednotky pohybu v ose (např. Zrychlení) během pohybu.
- Dvě tlačítka pro provedení přesunu jednotky pohybu v ose v záporném  nebo kladném  směru.
- Tlačítko  pro Zastavení pohybu jednotky pohybu v ose v kterémkoli okamžiku během přesunů v Absolutním a Krokovém režimu.




Oblast Pohybu

- Dvě pole pro nastavení *Záporné Polohy* a *Kladné polohy*, která se používají v *Absolutním* režimu.
- Pole pro nastavení *Rychlosti*, která má být vnucena jednotce pohybu v ose během pohybů v manuálním režimu.
- Tři tlačítka pro volbu režimu, ve kterém bude proveden pohyb: *Pomalý pohyb (Jog)*, *Pohyb s Absolutní polohou* nebo *Krokový pohyb (Step)*.
- Jedno pole pro nastavení hodnoty *Kroku* pro použití v režimu *Step*.

Pro pohyb jednotky pohybu v ose bude třeba vhodně nastavit popsané parametry. Bude třeba zvolit režim pohybu a stisknout tlačítko  (pro pohyb jednotky pohybu v ose v kladném směru) nebo tlačítko  (pro pohyb jednotky pohybu v ose v záporném směru).

V režimu *Jog* se bude jednotka pohybu v ose pohybovat, dokud zůstane tlačítko  nebo tlačítko  stisknuté. V režimu *Step* se jednotka pohybu v ose bude pohybovat v rozsahu určeném polem „krok“ při každém stisknutí tlačítka  nebo tlačítka .

V režimu *Absolutního pohybu* jednotka pohybu v ose přímo dosáhne polohy nastavené v poli kladné nebo záporné polohy.

Kromě pohybu tlačítka ,  a  je možné použít tlačítka „+“ (nebo Ctrl+P), „-“ (nebo Ctrl+M) a „mezerník“ klávesnice.

4 Technologické Parametry a Nástroje

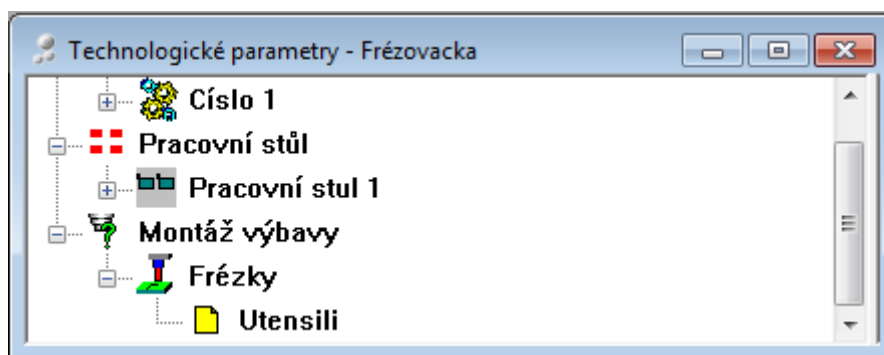
4.1 Okno Technologické Parametry

Archiv Technologických Parametrů umožňuje uložit do paměti informace geometrického a technologického charakteru stroje. Tyto informace jsou potřebné pro číslíkové řízení ke správné správě činnosti stroje. Otevírá se z menu **Soubor->Otevřít Technologické Parametry**.



Technologické Parametry jsou obvykle organizovány do Skupin / Podskupin (obecně skupiny a podskupiny Technologických Parametrů nejsou spojeny se skupinami a podskupinami, ve kterých jsou organizována zařízení stroje). Způsob zobrazování je definován Výrobcem Stroje a odlišen podle funkce specifické aplikace.

Hodnoty přítomné v archivu jsou obvykle určovány výrobcem ve fázi kalibrace stroje a obvykle nemohou být měněny Uživatelem, s výjimkou příležitostných změn. Proto je možné, že některá data jsou chráněna Heslem, aby se zabránilo náhodným změnám, které by mohly negativně ovlivnit správnou činnost systému.

Okno týkající se Technologických Parametrů zobrazuje strukturu ve formě stromu všech Skupin a Podskupin, které tvoří archiv - viz níže uvedený obrázek.



Struktura archivu Technologických Parametrů

Okno obsahuje některé Skupiny s příslušnými Podskupinami parametrů, zobrazené ve formě stromu. Struktura ve formě stromu může být rozbalena nebo sbalena prostřednictvím tlačítek  a  každého uzlu. Otevření a zavření částí, které tvoří strom, může být provedeno také tlačítky: +, -, šipky doprava a doleva.

Jak postupovat na Technologických Parametrech

Po otevření stromu požadované Skupiny / Podskupiny je možný přístup na stranu, která obsahuje údaje. Údaje mohou být zobrazeny v podobě tabulky nebo v podobě textových polí a volby; závisí to na druhu samotných dat a na tom, zda byla nastavena Výrobcem.

V případě provedení změn dat je třeba stisknout **[OK]**, aby se provedené změny staly trvalými.

Montáž výbavy

Specifickým případem dat týkajících se stroje je montáž výbavy. Obvykle jsou informace týkající se celku nástrojů, kterými je stroj vybaven (montáž výbavy), ukládány do archivu Technologických parametrů. Informace týkající se nástrojů jsou ukládány do archivu Parametrů Nástrojů.

Pro definování montáže výbavy stroje je třeba si přečíst informace obou archivů. Když to aplikace umožňuje, bude proto možné zobrazit z archivu Technologických Parametrů informace archivu Parametrů Nástrojů. Obvykle je spojení realizováno prostřednictvím tlačítka s ikonou podobnou níže uvedené ikoně.



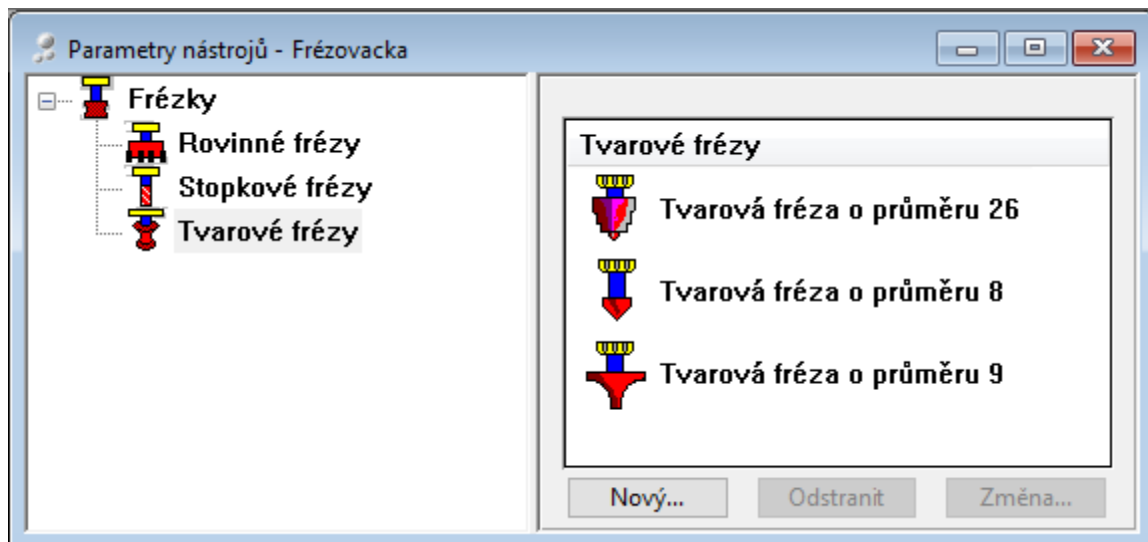
Volbou ikony a dvojklikem levým tlačítkem myši se otevře okno se seznamem nástrojů definovaných v archivu Parametrů Nástrojů, ve kterém je možné zvolit požadovaný nástroj. Po provedení volby se ikony tlačítka změň a bude zobrazena ta, která je přiřazena specifickému nástroji.

Dále je možné zobrazit údaje týkající se nástroje, a to dvojklikem pravým tlačítkem myši.

4.2 Okno Parametry Nástrojů



Okno Parametry Nástrojů se otevírá z menu **Soubor->Otevřít Parametry nástrojů**.

Parametry nástrojů uzpůsobené Výrobce na základě obrábění, které je stroj schopen provádět, jsou obvykle organizovány způsobem znázorněným na níže uvedeném obrázku:



Příklad okna Parametrů Nástrojů

Okno Parametrů Nástrojů je tvořeno dvěma oblastmi:

- *levá oblast* obsahuje Skupiny s příslušnými Podskupinami nástrojů, zobrazené ve formě stromu. Struktura ve formě stromu může být rozbalena nebo sbalena prostřednictvím tlačítek  a  každého uzlu. Např. je možné mít k dispozici Skupinu Frézy tvořenou Podskupinami fréz se vzájemně odlišnými vlastnostmi, jako jsou tvarové frézy, výkyvné frézy apod. Každá z těchto podskupin přiřadila jeden nebo více nástrojů, jejichž charakteristiky se přiřazují v dialogovém okně definovaném výrobcem. Nástroje, které se nacházejí v každé podskupině, jsou zobrazovány v pravé oblasti displeje.
- *pravá oblast*, která je označena názvem zvolené Podskupiny, obsahuje seznam Nástrojů, které jsou její součástí. Nástroje definované na tomto místě se nemusí nutně nacházet na stroji. Přiřazení mezi nástrojem a polohou na stroji (montáž výbavy) se obvykle provádí v archivu technologických parametrů.

Jak postupovat na Parametrech Nástrojů

Operace vložení, změny a vymazání nástrojů z archivu jsou možné *tlačítky*, která se nacházejí ve spodní části okna:

- [Nový...]** umožňuje vložit do Podskupiny nový nástroj. Otevře dialogové okno „Nový nástroj“, ve kterém je možné přiřadit níže uvedené údaje:
- *Popis*: Jedná se o hlášení, které identifikuje nástroj. Popis může být zvolen z popisů, které se nacházejí v seznamu, pokud již nebyl přiřazen jinému nástroji, nebo je možné definovat nový popis.
 - *Obrázek*: Jedná se o ikonu, která identifikuje nástroj. Může být zvolen z těch, které jsou uvedeny v seznamu, nebo může být vyvolán z adresáře tlačítkem **[Obrázek]**. Nástroj je vložen za dodržení abecedního pořadí popisů.
- [Odstranit]** Umožňuje odstranit nástroj z Podskupiny na základě potvrzení; příslušný popis nebude odstraněn a zůstane k dispozici pro jiný nástroj.
- [Změna...]** umožňuje nahradit *popis* nebo *obrázek* zvoleného nástroje; dojde k zobrazení stejného dialogového okna popsaného v příkazu **[Nový...]**.

5 Diagnostika

5.1 Okno Diagnostiky

Během výkonné fáze stroje je možné otevřít okno *Diagnostiky*, které umožňuje Obsluze mít pod kontrolou stav činnosti stroje a monitorovat logický stav signálů digitálních VST./VÝST., hodnotu analogových VST./VÝST., stav počítadel a časovačů a pohyb jednotek pohybu v osách.

V závislosti na [přístupových právech](#) přiřazených výrobcem je dále možné měnit stav zařízení.

Umožňuje-li to úroveň přístupu, je dále možné získat v reálném čase:

- Zobrazení stavu (ZAP./VYP.) všech signálů Digitálních vstupů a Digitálních výstupů
- možnost aktivace a zakázání signálů Digitálních výstupů
- zobrazení aktuálních hodnot (v intervalu +/- 10V) Analogových vstupů
- možnost přiřadit hodnotu (v intervalu +/- 10V) všem Analogovým výstupům
- možnost pohybovat jednotkou pohybu v ose v Manuálním režimu na základě volby její rychlosti, hodnoty Kroku nebo Absolutní koncové polohy se zobrazením aktuální polohy, rychlosti a chyby smyčky.
- zobrazování a změna globálních proměnných

V níže uvedených odstavcích budou podrobně popsána zařízení a globální proměnné a jejich grafické zobrazení.

POZNÁMKA: V okně diagnostiky budou zobrazena výhradně zařízení aktivovaná pro aktuální úroveň přístupu.

5.2 Složení okna Diagnostiky

S odvoláním na strukturu „Skupiny / Podskupiny“, která již byla popsána v kapitole [Složení systému](#), je možný přístup k zařízením, která jsou poté zobrazována ve struktuře ve formě stromu. Hlavním prvkem struktury je skupinu zastoupený ikonou



, po které následuje Název a Komentář.

Struktura se rozvíjí nebo zavírá *kliknutím* na tlačítko nebo . Otevření a zavření částí, které tvoří strom, lze provést i tlačítky: +, -, šipkovými klávesami pro pohyb doprava a doleva.

Po otevření Skupiny se ve stromu zobrazí

- „Seznam zařízení“ Skupiny zastoupený symbolem
- případné Podskupiny, které tvoří Skupinu.

Otevřením jedné z Podskupin se objeví zařízení, které ji tvoří.

5.3 Znázornění Zařízení

Pro každé ze zobrazených zařízení budou zobrazeny níže uvedené informace:

- Grafický symbol;
- aktuální Stav nebo Hodnota;
- Název;
- Komentář.

Níže je zobrazeno grafické znázornění zařízení, druh zařízení a to, která jeho veličina bude zobrazována v reálném čase.

Stav digitálních vstupů, digitálních výstupů a příznaků je znázorněn graficky prostřednictvím LED, která získá určitou barvu v závislosti na tom, zda je vstup zapnutý nebo vypnutý.

V případě Portů, tedy současného znázornění více vedení současně (8), se zobrazí řada LED; první vedení skupiny je znázorněno prostřednictvím LED nejvíce napravo a poslední prostřednictvím LED nejvíce vlevo.

Zařízení	Symbo	Stav	Zobrazení v reálném čase
Digitální vstup			stav: Zapnutý = ZELENÝ, Vypnutý = ŠEDÝ
Digitální výstup			stav: Zapnutý = ČERVENÝ, Vypnutý = ŠEDÝ
Analogový vstup		22.000	aktuální hodnota

Analogový výstup		22.000	číselná hodnota, vyjádřená ve voltech
Port vstupů			stav každého vedení (jako Digitální vstup). Stav: Zapnutý = ZELENÝ, Vypnutý = ŠEDÝ
Port výstupů			stav každého vedení (jako Digitální výstup). Stav: Zapnutý = ČERVENÝ, Vypnutý = ŠEDÝ
Jednotka pohybu v ose		100.000	absolutní aktuální poloha
Časovač		12.000	aktuální hodnota v sekundách
Počítadlo		58	aktuální číselná hodnota
Bit příznaku			stav: Zapnutý = ŽLUTÝ, Vypnutý = ŠEDÝ
Přepínač příznaku			stav (jako Bit příznaku). stav: Zapnutý = ŽLUTÝ, Vypnutý = ŠEDÝ
Port příznaku			stav každého vedení (jako Bit příznaku). stav: Zapnutý = ŽLUTÝ, Vypnutý = ŠEDÝ
Globální proměnná		2	globální proměnná typu integer (celé číslo) kódu GPL
Globální proměnná		127	globální proměnná typu char (znak) kódu GPL.
Globální proměnná		50.00000000	globální proměnná typu float (hodnota s plovoucí desetinnou čárkou) kódu GPL
Globální proměnná		200.00000000	globální proměnná typu double (hodnot reálného čísla se zvýšenou přesností) kódu GPL
Globální proměnná		Area	globální proměnná typu string (řetězec) kódu GPL
Globální proměnná		[256]	globální proměnná typu array (pole) kódu GPL
Globální proměnná		[10][3]	globální proměnná typu matrix (matrice) kódu GPL

5.4 Interakce se Zařízením

Pro diagnostické účely je možné provádět interakci se zařízením a přečíst si tak jeho stav nebo změnit jeho hodnotu. Není to možné u některých druhů zařízení, jako například u vstupů a zařízení chráněných Výrobce. V tomto případě příslušné hlášení informuje Obsluhu v případě pokusu o provedení operací na zařízení.

Po provedení volby zařízení *dvakrát klikněte* myší nebo stiskněte tlačítko **Enter** či **Mezerník** za účelem přístupu k oknu, ve kterém lze měnit stav nebo hodnotu zařízení.

V případě, že se jedná o **Digitální výstup** nebo **Příznak bitu**, nezobrazí se žádné okno, ale bude pouze poskytnuta informace o stavu zařízení. Vyhovující činnost výstupu bude signalizována změnou barvy LED, která informuje o jeho stavu.

V případě **Portů Výstupů** je třeba myší zvolit LED, která odpovídá požadovanému výstupu, a *dvakrát kliknout* za účelem realizace změny jejího stavu.

Totéž platí i pro **Příznak přepínače** a pro **Příznak portu**.

Pro **Analogové výstupy**, **Časovače** a **Počítadla** se zobrazí Dialogové okno, které zobrazuje aktuální hodnotu a které umožňuje nastavit novou hodnotu, kterou má zařízení okamžitě nabýt.

Způsob interakce s **Jednotkou pohybu v ose** je popsán v části [Pohyb Jednotek pohybu v osách v Manuálním režimu](#), na kterou vás tímto odkazujeme.

5.5 Seznam tlačítek pro pohyb uvnitř struktury ve formě stromu

Tlačítko	Popis
Šipka Nahoru	slouží k přemístění volby na bezprostředně předchozí nebo následující řádek
Šipka Dolů	
Šipka Doprava	slouží k rozbalení úrovně zvolené větve, a v případě, že je větev již rozbalena, k přemístění na následující větev
Šipka Doleva	slouží k zavření zvolené větve, a v případě, že je větev již zavřena, k přemístění na předchozí větev
+	slouží k rozbalení úrovně zvolené větve
-	slouží k zavření zvolené větve
*	slouží k rozbalení všech úrovní zvolené větve

Ctrl+Alt+Shift a Enter
Ctrl+Alt+Shift a kliknutí
levým tlačítkem myši

slouží k zobrazení tabulek korektorů linearity, přiřazených k jednotce pohybu v ose. Když je kombinace tlačítek aktivovaná, zatímco je zvolená jednotka pohybu v ose ve stromu zařízení modulu, budou zobrazeny všechny korektory přiřazené k jednotce pohybu v ose, jako by se jednalo o novou matici, ve které sloupce představují přiřazené jednotky pohybu v ose (první sloupec je vyhrazený pro samokorektory) a řádky představují hodnoty korekce. Případně změněné hodnoty budou zohledněny během pohybu jednotky pohybu v ose, ale nebudou uloženy do paměti na disku.

5.6 Korektory linearity

V diagnostice lze zobrazovat a měnit korektory linearity zvolené jednotky pohybu v ose, a to otevřením kontextuální nabídky a volbou položky **Korektory linearity**. Tato položka je viditelná pouze v případě, když byly pro danou jednotku pohybu v ose nadefinovány v rámci konfigurace korektory linearity. Jako alternativu k nabídce lze použít kombinaci tlačítek **[Ctrl+Shift+Enter]**.

5.7 Panel kalibrace jednotek pohybu v osách

Panel pro kalibraci jednotek pohybu v osách představuje nástroj, který umožňuje měnit parametry konfigurace jednotky pohybu v ose a současně umožňuje pohybovat jednotkou pohybu v ose a zobrazovat chybu smyčky na virtuálním osciloskopu.

Přístup na panel pro kalibraci jednotek pohybu v osách vyžaduje **úroveň přístupu** do systému vyšší nebo rovnou úrovni „Výrobce“. Přístup na panel pro kalibraci je možný z diagnostického režimu, nebo z manuálního režimu dvojitým kliknutím na jednotku pohybu v ose, jejíž kalibrace má být provedena za přidržení tlačítka **[shift]**, nebo volbou položky „Kalibrace“ v souvisejícím menu.

Tím se umožní přístup na panel kalibrace, který je zobrazen na níže uvedeném obrázku:

Za účelem ověření chování jednotky pohybu v ose po provedení změny parametrů se provádí její nepřetržitý pohyb mezi dvěma mezními polohami nazvanými **Kladná cíl. poloha** a **Záporná cíl. poloha**. Kromě těchto parametrů je třeba nastavit **Rychlost** pohybu jednotky pohybu v ose. V prvních fázích kalibrace se doporučuje

používat nízkou hodnotu rychlosti. Dále je možné nastavit **Opoždění**, které bude aplikováno mezi dvěma po sobě následujícími pohyby.

V okně osciloskopu bude zobrazen průběh chyby smyčky nebo jedna z dalších veličin jednotky pohybu v ose. Obdobně je možné i na osciloskopech na pracovním stole vhodně upravit měřítko grafu tak, aby došlo k jeho přizpůsobení rozměru okna, nebo za účelem zvýraznění detailů. Prostřednictvím myši a ovládacích tlačítek je možné si znovu prohlédnout poslední minutu kalibrace, zobrazit jeden nebo dva kurzory pro potřeby měření a kontroly vzorkovaného údaje, zvětšit část grafu za účelem analyzování detailů vzorkovaného údaje za účelem změny posunu a měřítka pořadnicové i souřadnicové osy. Dále je možné stisknutím tlačítka **Zastavit** zablokovat posouvání grafu, aby bylo možné jej v klidu prohlížet bez nutnosti zastavení jednotky pohybu v ose. Kromě grafu jsou ve dvou polích zobrazeny: Reálná poloha (vlevo) a veličina zobrazovaná na grafu (vpravo). Tuto lze nastavit prostřednictvím pole se seznamem, které se nachází nad polem pro zobrazování.

Pro změnu parametrů jednotky pohybu v ose je třeba stisknout tlačítko **[Parametry...]**, které aktivuje okno viditelné na níže uvedeném obrázku, v němž je možné bezprostředně měnit část parametrů jednotky pohybu v ose. Po provedení změny jednoho nebo více parametrů bude možné provést její aplikaci stisknutím tlačítka **[Aplikovat]**.

Toto tlačítko umožňuje okamžitě vidět účinek změny v dynamickém chování jednotky pohybu v ose. Při stisknutí **[Ok]** zůstanou změny v platnosti, zatímco při stisknutí **[Zrušit]** budou obnoveny hodnoty, které byly přítomny před otevřením okna **[Parametry...]**

K parametrům, na kterých se budou obvykle provádět změny, patří:

- **Proporcionální** koeficient
- **Integrační** koeficient
- **Derivační** koeficient
- **Feed Forward**: Procentuální vyjádření aktuální rychlosti, která je poskytována přímo do pohonu (nezávisle na chybě smyčky)
- **Feed Forward Zrychlení**: Procentuální vyjádření řídicího signálu rychlosti, který je dodáván přímo do pohonu během fází zrychlení a zpomalení jednotky pohybu v ose (jako přídavek k dopřednému řízení)
- **Zrychlení**: Doba trvání rampy zrychlení
- **Zpomalení**: Doba trvání rampy zpomalení

Kursor na grafické ploše

Kursor je nástroj, který umožňuje měřit a zobrazovat některé údaje stopy. Je tvořen svislou čarou barvy, přiřazené k zvolené stopě. Je možné přemístit se dovnitř grafu kurzorovými tlačítky nebo myší. V hlášení s pomocným textem (tooltip), které se vztahuje na svislou čáru, se mohou zobrazit různé hodnoty, které lze zvolit prostřednictvím překryvného okna nabídky, které lze vyvolat kliknutím pravým tlačítkem myši v blízkosti kurzoru.

V překryvném okně nabídky lze zvolit následující možnosti:

- **Kanál**: překládá seznam stop, zobrazených na grafu a zvýrazňuje symbolem zaškrtnutí tu, na kterou se vztahuje kurzor. Dále je možné zvolit novou stopu, která má být přiřazena ke kurzoru.
- **Styl**: předkládá seznam dat, které mohou být zobrazeny v obdélníku hlášení s pomocným textem (tooltip).
- **Hodnota X-Y**: zobrazuje hodnotu na ose x a na ose y bodu stopy, ve které se nachází kurzor.
- **Hodnota X**: zobrazuje hodnotu na ose x bodu stopy, ve které se nachází kurzor.
- **Hodnota Y**: zobrazuje hodnotu na ose y bodu stopy, ve které se nachází kurzor.
- **Období**: zobrazuje hodnotu vzdálenosti mezi dvěma zobrazenými kurzory, když se provádí měření podél osy X.
- **Špička-špička**: zobrazuje hodnotu vzdálenosti mezi dvěma zobrazenými kurzory, když se provádí měření mezi dvěma body podél osy Y.
- **Frekvence**: zobrazuje převrácenou hodnotu vzdálenosti podél osy X.
- **Možnosti**: slouží ke konfiguraci zobrazení kurzoru a přiřazeného hlášení s pomocným textem (tooltip).
- **Použití barvy kanálu**: když je tato možnost aktivována, kurzor je nakreslen barvou vzorkovaného údaje, na který se vztahuje; v opačném případě je barva, kterou je zobrazen kurzor, náhodná.
- **Ukrytí radu při uvolnění**: když je tato možnost aktivována, přiřazené hlášení s pomocným textem (tooltip) je zobrazeno pouze po dobu přidržení levého tlačítka myši ve stisknutém stavu.
- **Obrátit vyrovnání**: hlášení s pomocným textem (tooltip) je obvykle zobrazeno napravo od kurzoru. Když je tato možnost aktivována, hlášení s pomocným textem (tooltip) bude zobrazeno nalevo od kurzoru.
- **Polohování podle vzorku**: když je tato možnost aktivována, umístí kurzor pouze na vzorkované hodnoty.

Kalibrace jednotky pohybu v ose

Kalibrace jednotky pohybu v ose představuje choulostivou operaci, kterou je třeba provádět s maximální pozorností a opatrností.

Prostřednictvím položky CalibSampleTime v části [Albatros] v Tpa.ini je možné měnit dobu vzorkování dat jednotky pohybu v ose pro okno Kalibrace. Hodnota je v milisekundách a nemůže být menší než frekvence řízení jednotek pohybu v osách ani větší než 100.

Před provedením kalibrace jednotky pohybu v ose z panelu je třeba nastavit všechny parametry v konfiguraci a bude třeba také nastavit maximální hodnotu rozsahu stupnice rychlosti pohonu. U analogových jednotek pohybu v osách je hodnota napětí, které v softwaru Albatros odpovídá maximální rychlosti 9 voltů.

Aby se zabránilo poškození stroje použitím chybných parametrů, je třeba nastavit nízkou rychlost, například rovnající se 10 % maximální rychlosti určené pro danou jednotku pohybu v ose. Tímto způsobem se zabrání příliš prudkým reakcím jednotky pohybu ose i při nastavení příliš velkého zisku.

Obvykle se nejdříve provádí kalibrace pro pohyby z bodu do bodu, a teprve poté kalibrace pro interpolované pohyby.

První operací, pokud již nebyla provedena v rámci konfigurace, je nastavení dob zrychlení a zpomalení. Čím jsou tyto doby delší, tím menší bude zrychlení, kterému bude vystavena jednotka pohybu v ose.

Poté se přejde na nastavení minimálního zisku, který umožňuje pohyb jednotky pohybu v ose. Tím lze ověřit správnou kalibraci pohonu. Albatros je vskutku nakonfigurován tak, aby dodával řídicí signál 9 voltů, který odpovídá maximální rychlosti, která je nastavena v rámci konfigurace jednotky pohybu v ose. Když je například prováděn pohyb jednotky pohybu v ose rychlostí rovnající se 10 % maximální rychlosti, bude možné odečítat řídicí napětí rovnající se 10 % maximálního řídicího napětí, tedy 0,9 voltů. Když nebude dosaženo právě této hodnoty řídicího signálu, je třeba změnit maximální hodnotu rozsahu stupnice pohonu.

Po provedení kalibrace pohonu se začne po malých hodnotách a stále s maximální opatrností zvyšovat zisk smyčky sledování polohy. Čas od času je třeba ověřit, zda nedochází k projevům stavu přílišné elongace nebo nestability. Rychlost musí být v této fázi udržována na hodnotě menší nebo rovné 10 % maximální rychlosti. V této fázi je vždy vhodné podrobně analyzovat profil rychlosti získaný virtuálním osciloskopem a co nejvíce zvětšit obraz za účelem zvýraznění detailů.

Když jednotka pohybu v ose dosáhne chování, které je zároveň reaktivní i stabilní, bude možné po troškách zvyšovat rychlost a čas, od času ověřit chování jednotky pohybu v ose v případě, že by nebylo uspokojivé. Zisk a rychlost se nikdy nesmí měnit prudce. Podmínky kalibrace, které se při nízkých rychlostech jeví jako stabilní, by takové nemusely být při vyšších rychlostech.

Po určení optimální hodnoty zisku bude možné postupně zvýšit Integrační a derivační koeficient a následně i feed forward (dopředné řízení) za účelem snížení chyby smyčky a dosažení jejich hodnot přijatelných z hlediska přesnosti jednotky pohybu v ose. Feed forward (dopředné řízení) umožňuje téměř úplně potlačit chybu sledování během fáze režimu pohybu, ale ne během fází zrychlení a zpomalení. Pro další snížení chyby smyčky je v těchto fázích možné zvýšit feed forward zrychlení. Obecně platí, že i velmi nízké hodnoty tohoto parametru stačí pro dosažení dobrých výsledků.

Co se týká kalibrace jednotky pohybu v ose pro interpolované pohyby, je možné použít již určené hodnoty pro pohyby z bodu do bodu, avšak je třeba zohlednit přítomnost dalších jednotek pohybu v osách stroje. Zejména pro dosažení dobré přesnosti interpolovaných pohybů je třeba vyrovnat chyby smyčky jednotek pohybu v osách. Proto bude třeba určit jednotku pohybu v ose s nejvyšší chybou smyčky (při určité rychlosti) a „zhoršit“ (omezeně na parametry interpolace) nastavení dalších tak, aby byly chyby smyčky stejné.

6 Chyby a Zprávy

6.1 Úvod

Albatros zpracuje události chyby a signalizace

Chyby systému

Jedná se o chyby, které je systém Albatros schopen automaticky zaznamenat, a to jak během fáze provádění programů, tak během operací údržby a diagnostiky výrobního zařízení.

Tyto chyby jsou různého původu a pohybují se od problémů správy jednotky pohybu v ose po problémy, které se mohou projevit během provádění programu.

Chyby systému mohou být spravovány uvnitř pracovních programů prostřednictvím instrukce [ONERRSYS](#); v opačném případě bude při jejich výskytu ukončeno provádění programů na modulu, na kterém se chyba vyskytla. Vysvětlení každé chyby systému je popsáno na následujících stranách tohoto návodu.

Chyby cyklu

Jedná se o chyby, které se vyskytují během provádění programu, ale které obvykle umožňují jeho další pokračování po odstranění samotné chyby. Chyby cyklu lze vytvářet prostřednictvím instrukce [ERROR](#).

Hlášení

Jedná se o upozorňující hlášení, která vznikají ve specifických situacích provádění programu, nebo o signalizace požadavku na zákrok ze strany operátora, které však nezastavují provádění samotného programu. Hlášení lze vytvářet prostřednictvím instrukce GPL [MESSAGE](#)

Signalizace

Jedná se o obecné signalizace, které nesouvisí s cyklem stroje, a které jsou zapisovány systémem Albatros. Nejsou zobrazovány v systému Albatros. Popis každé signalizace je přítomný na následujících stranách tohoto návodu.

Chybová úsečka

V chybovém řádku je zobrazována poslední chyba systému, která se vyskytla v čase, spolu s poslední chybou cyklu a posledním hlášením.

Chyby systému jsou odlišeny červenou barvou.

Chyby cyklu jsou odlišeny žlutou barvou. Jedná se o chyby, které se vyskytují během provádění programu, ale které obvykle umožňují jeho další pokračování po odstranění samotné chyby.

Hlášení jsou odlišena zelenou barvou.

Vrtacka: Modul není nakonfigurován	00001
Frézovacka: Vynulování jednotek pohybu v osách ještě nebylo provedeno	00010
Frézovacka: Probíhá inicializace...	00002

Chybová úsečka

Chyby, které se vyskytly od zapnutí systému, lze zobrazit v okně, které lze otevřít dvojitým kliknutím myši na *Chybová úsečka* nebo z nabídky **Zobrazit**. V tomto okně jsou dále zobrazovány přídavné informace o chybách systému.

Okno je rozděleno do níže uvedených oblastí. V **horní části** jsou zobrazeny následující informace:

- Druh: identifikuje druh chyby (chyba systému, chyba cyklu a hlášení).
- Čas a Datum: Jedná se o čas a datum vzniku chyby.
- Popis: Obsahuje popis chyby.
- Kód: Jedná se o číslo chybového hlášení.
- Úkol: Jedná se o název úkolu, který způsobil vznik chyby (není přítomen v Chybovém řádku).

Dvoklíkem na jeden z těchto sloupců dojde k uspořádání informací na základě obsahu sloupce.

V **níže uvedené části** se nacházejí tyto pole:

- Chyby cyklu: Je-li aktivováno, zobrazují se v něm také tyto chyby.
- Hlášení: Je-li aktivováno, zobrazují se v něm také hlášení.
- Všechna: Je-li aktivováno, zobrazují se hlášení všech modulů systému, která se týkají druhu zobrazovaných informací.

- Pole názvu modulu: Zobrazuje název modulu, jehož informace jsou zobrazovány, a umožňuje v případě systému s více moduly dále zvolit modul, jehož informace mají být zvoleny.

Kromě výše uvedených informací zde najdeme tato ovládací **tlačítka**:

- **[Odstranit všechny]** slouží k vymazání všech informací z paměti, avšak bez jejich odstranění z archivu.
- **[Odstranit]** umožňuje vymazat aktuální informaci z paměti, avšak bez toho, aby byla odstraněna z archivu.
- **[OK]** umožňuje zavřít okno.

Ukládání chyb do paměti a signalizace do souboru sestavy

Všechny chyby jsou ukládány do paměti v podobě souboru za účelem získání jejich archivního přehledu. Jedná se o textový soubor ve formátu TSV. Název souboru je MONTH (aktuální měsíc).TER. V sadě systému Albatros je dodáván program ViewRER, který načítá a zobrazuje soubory .TER.

6.2 Chyby systému

6.2.1 Chyby vytvořené správou jednotek pohybu v osách

1 NázevJednotkyPohybuOse: Chybné připojení snímače impulzů

Příčina:

Při zastavení jednotky pohybu v ose byl zaznamenán rozdíl mezi teoretickou a reálnou polohou jednotky pohybu v ose, který převyšuje 1024 kroků snímače impulzů.

Obvykle se tento problém vyskytuje během uvedení jednotky pohybu v ose do provozu, když jsou vzájemně přehozené fáze snímače impulzů. Během běžné činnosti se tento problém může vyskytnout v případě, že se jednotkou pohybu v ose pohybuje manuálně při vypnutém pohonu, aniž by byla předem přepnuta do stavu FREE, nebo když je jednotka pohybu v ose následkem nesprávné kalibrace vystavena přílišné elongaci ve fázi dosažení cílové polohy (overshoot).

Po výskytu této chyby dojde k vynulování řídicího signálu a k přepnutí jednotky pohybu v ose do stavu FREE.

Řešení:

Ve fázi uvedení jednotky pohybu v ose do provozu zkontrolujte připojení fází snímače impulzů příslušné jednotky pohybu v ose (případně aktivujte volitelnou funkci záměny fází snímače impulzů v konfiguraci jednotky pohybu v ose). Zkontrolujte kalibraci jednotky pohybu v ose prostřednictvím příslušného režimu Diagnostiky.

2 NázevJednotkyPohybuOse: Nedokončený pohyb

Příčina:

Na konci přesunu, do 5 sekund po skončení teoretického pohybu, je rozdíl mezi teoretickou polohou a reálnou polohou jednotky pohybu v ose vyšší než toleranční okno uvedené v Konfiguraci. Může to jednoduše souviset s tím, že pohon je vypnutý nebo není aktivovaný, důvodem může být dále nesprávné nastavení posunu pohonu. Důvodem může být i mechanická vůle nebo příliš nízký zisk smyčky sledování polohy jednotky pohybu v ose.

Řešení:

Zkontrolujte, zda je pohon zapnutý a aktivovaný. Zkontrolujte kalibraci jednotky pohybu v ose a nastavte pohon příslušné jednotky pohybu v ose.

3 NázevJednotkyPohybuOse: Chyba servořízení

Příčina:

Během jakéhokoli pohybu se rozdíl mezi teoretickou a reálnou polohou jednotky pohybu v ose stal větším, než je maximum uvedené v Konfiguraci nebo nastavené instrukcí [SETMAXER](#).

Obvykle je důvodem nesprávné nastavení zisku sledování polohy nebo rozsah stupnice rychlosti pohonu nebo nadměrná setrvačnost jednotky pohybu v ose.

Řešení:

Zkontrolujte nastavení zisku a rozsah stupnice rychlosti pohonu.

Zkontrolujte správnou činnost snímače impulzů a jednotky motoru/pohonu.

Dále zkontrolujte případná mechanická zablokování.

4 NázevJednotkyPohybuOse: Překročení kladného limitu

Příčina:

Teoretická poloha jednotky pohybu v ose překročila kladný limit nastavený v Konfiguraci nebo nastavený instrukcí [SETLIMPOS](#).

Řešení:

Provedte v programu korekci polohy, která přesahuje kladný limit polohy, nebo znovu definujte limity polohy jednotky pohybu v ose.

5 NázevJednotkyPohybuOse: Překročení záporného limitu

Příčina:

Teoretická poloha jednotky pohybu v ose překročila záporný limit nastavený v Konfiguraci nebo nastavený instrukcí [SETLIMNEG](#).

Řešení:

Provedte v programu korekci polohy, která přesahuje záporný limit polohy, nebo znovu definujte limity polohy jednotky pohybu v ose.

10 NázevJednotkyPohybuOse: Provedení v reálném čase (Real-Time) je rychlejší než vytváření profilu

Příčina:

Provádění v reálném čase (real-time) profilu pohybu je rychlejší než vytváření GPL samotného profilu. Vyhledávání předem je vyprázdněno rychleji, než je naplňováno. Chyba může být způsobena dvěma příčinami, které se obvykle vyskytují najednou:

- Rychlost interpolace je příliš vysoká vzhledem k rozměrům úseků, které je třeba projít
- Úseky, které je třeba projít, jsou příliš krátké.

Řešení:

Zkontrolujte, zda nastavená rychlost interpolace není příliš vysoká vzhledem k rozměrům úseků, které je třeba projít, a ověřte, zda úseky interpolace, které je třeba projít, nejsou příliš krátké.

6.2.2 Chyby vytvořené správou vzdálených VST./VÝST.

2049 Příjímač číslo: Chybná konfigurace

Příčina:

Vzdálený přijímač přijal konfiguraci odlišnou od té, která byla zaznamenána v poli. Situace tohoto druhu může nastat, když vzdálený modul není stejný jako ten, který byl zvolen v konfiguraci hardwaru řízení Albatros. Když je například vzdáleným přijímačem Albre16 a v řízení Albatros je nakonfigurován vzdálený modul Albre24 (GreenBus v3.0) nebo ještě TRS-IO s chybným počtem rozšíření TRS-IO-E (GreenBus v4.0).

Řešení:

Zkontrolujte Konfiguraci Hardwaru.

2050 Příjímač číslo: Odpojený

Příčina:

Vzdálený přijímač neodpovídá na příkazy vysílače.

Řešení:

Zkontrolujte napájení přijímače a sériové připojení.

2051 Příjímač číslo: Znovu připojený

Příčina:

Přepojení mezi vysílačem a přijímačem bylo obnoveno.

2052 Příjímač číslo: Chyba při opětovném čtení nepřipojeného výstupu číslo ČísloVýstupu

Příčina:

Uvedený digitální výstup se nachází ve stavu zásahu ochrany nebo zkratu, a v každém případě se nachází ve stavu odlišném od stavu určeného řízením. Výstup není přiřazen žádnému logickému zařízení v Konfiguraci Virtuální-Fyzické, a proto se jedná o projev neshody mezi Konfigurací a reálnou kabeláží stroje.

Řešení:

Zkontrolujte Konfiguraci Virtuální-Fyzické. Odstraňte zkrat nebo zkontrolujte, zda aplikovaná zátěž nepřesahuje maximální limity (přečtěte si technickou dokumentaci).

2054 Příjímač číslo: Chybný druh

Příčina:

Během fáze inicializace vzdálených modulů byla na jisté adrese zaznamenána přítomnost jiného přijímače, než je uveden v konfiguraci.

Řešení:

Zkontrolujte, zda Konfigurace Hardwaru odpovídá nastavením vzdálených modulů.

2055 Příjímač číslo: Inicializován

Příčina:

Příjímač se připojil k vysílači po odpojení způsobeném chybějícím napájením.

2056 Příjímač číslo: Chyba napájení +24 V=

Příčina:

Polní napájení (+24 V=) modulu vzdáleného VST./VÝST. není aktivní nebo nefunguje správně.

Řešení:

Zkontrolujte činnost napájení +24 V=.

2057 Chyba napájení GreenBUS

Příčina:

Napájení průmyslové sběrnice, která je použita pro propojení modulů vzdáleného VST./VÝST. s řízením, nefunguje správně. Toto napájení musí mít jmenovitou hodnotu +12 V= a je dodáváno řízením.

Řešení:

Zkontrolujte přítomnost napájení GreenBUS a kabely GreenBUS. Provedte vypnutí a opětovné zapnutí. Případně vyměňte kartu řízení.

2058 Příjímač číslo: Chyba opětovného čtení DruhZařiz NázevZařiz

Příčina:

Stav uvedeného výstupu neodpovídá nastavenému stavu. Může to být způsobeno přítomností zkratu, zásahem ochrany (nadměrná zátěž) nebo jednoduše chybějícím napájením. Uvedeným výstupem může být digitální výstup, analogový výstup nebo výstup řízení jednotky pohybu v ose. Druh výstupu je uveden v zobrazení signalizace chyby.

Řešení:

V případě, že se jedná o digitální výstup, zkontrolujte napájení +24 V (strana pole), odstraňte případnou situaci zkratu nebo nadměrného proudového odběru výstupu (přečtěte si příslušnou technickou dokumentaci). V případě, že se jedná o analogový výstup nebo o výstup řízení jednotky pohybu v ose, zkontrolujte přítomnost a hodnotu napětí nastaveného na výstupu (pomocí testeru nebo osciloskopu), odstraňte případnou situaci zkratu nebo nadměrného proudového odběru výstupu (přečtěte si příslušnou technickou dokumentaci).

2059 Nezdařila se zkouška paměti s duálním portem vysílače

Příčina:

Došlo k výskytu chyby během zkoušek provedených ve fázi inicializace karty jednotek pohybu v osách. Konkrétně se nezdařila inicializace vysílače GreenBUS (mikrokontroler i296). Může to být důsledkem chybné konfigurace adres VST./VÝST. a IRQ karty nebo důsledkem konfliktu s jinými periferními zařízeními, která jsou součástí systému. Kromě toho se může jednat o následek poškození karty jednotek pohybu v osách.

Řešení:

Zkontrolujte konfiguraci karty a dále zkontrolujte, zda se nevyskytují konflikty s ostatními periferními zařízeními. V případě použití vzdáleného modulu zopakujte přenos firmwaru do modulu. Kvalifikovaní technici mohou provést zkoušku Hardwaru paměti s duálním portem mikrokontroleru i296. V případě opětovného výskytu problému se obraťte na výrobce.

2060 Chyba inicializace vysílače

Příčina:

Došlo k výskytu chyby během zkoušek provedených ve fázi inicializace karty jednotek pohybu v osách. Konkrétně se nezdařilo odeslání firmwaru do vysílače GreenBus (mikrokontroler i296). Může to být důsledkem chybné konfigurace adres VST./VÝST. a IRQ karty nebo důsledkem konfliktu s jinými periferními zařízeními, která jsou součástí systému. Kromě toho se může jednat o následek poškození karty jednotek pohybu v osách.

Řešení:

Zkontrolujte konfiguraci karty a dále zkontrolujte, zda se nevyskytují konflikty s ostatními periferními zařízeními. V případě použití vzdáleného modulu zopakujte přenos firmwaru do modulu. Kvalifikovaní technici mohou provést zkoušku Hardwaru Paměti s Duálním Portem mikrokontroleru i296. V případě opětovného výskytu problému se obraťte na výrobce.

2061 Výskyt chyby během přenosu firmwaru do vysílače**Příčina:**

Došlo k výskytu chyby během zkoušek provedených ve fázi inicializace karty jednotek pohybu v osách. Konkrétně se nezdařilo odeslání konfigurace modulů vzdáleného VST./VÝST. do vysílače GreenBus (mikrokontroler i296).

Řešení:

Zkontrolujte konfiguraci hardwaru a používá-li se vzdálený modul, zopakujte přenos firmwaru do modulu. Kvalifikovaní technici mohou provést zkoušku Hardwaru RAM mikrokontroleru i296. V případě opětovného výskytu problému se obraťte na výrobce.

2062 Výskyt chyby během přenosu konfigurace do vysílače**Příčina:**

Došlo k výskytu chyby během zkoušek provedených ve fázi inicializace karty jednotek pohybu v osách. Konkrétně se nezdařila inicializace modulů vzdáleného VST./VÝST.

Řešení:

Zkontrolujte konfiguraci hardwaru a používá-li se vzdálený modul, zopakujte přenos firmwaru do modulu. Kvalifikovaní technici mohou provést zkoušku Hardwaru RAM mikrokontroleru i296. V případě opětovného výskytu problému se obraťte na výrobce.

2063 Výskyt chyby během přenosu konfigurace do přijímače**Příčina:**

Došlo k výskytu chyby během inicializace vzdáleného modulu.

Řešení:

Zkontrolujte konfiguraci hardwaru. Kvalifikovaní technici mohou provést zkoušku Hardwaru vzdáleného modulu. V případě opětovného výskytu problému se obraťte na servisní službu.

2064 Přijímač číslo: Nekompatibilní verze firmwaru**Příčina:**

Vzdálený přijímač je vybaven verzí firmwaru, která není kompatibilní s firmwarem řízení.

Řešení:

Zkontrolujte instalaci řízení. V případě přetrvání uvedeného problému se obraťte na servisní službu.

2065 Přijímač číslo: Chyba asynchronní komunikace**Příčina:**

Došlo k výskytu chyby nebo k zaznamenání chybějící odpovědi během komunikace příkazu směrem do vzdáleného modulu (GreenBus v4.0).

Řešení:

Zkontrolujte připojení a napájení GreenBus. V případě přetrvávání uvedeného problému se obraťte na servisní službu.

2066 Přijímač číslo: Všeobecná chyba**Příčina:**

Došlo k výskytu všeobecné chyby během komunikace události nebo alarmu ze vzdáleného modulu (GreenBus v4.0)

Řešení:

Zkontrolujte připojení a napájení GreenBus. V případě přetrvávání uvedeného problému se obraťte na servisní službu.

2067 Příjímač číslo: Chyba během přenosu konfigurace

Příčina:

Došlo k výskytu chyby komunikace při přenosu dat konfigurace do vzdáleného modulu (GreenBus v4.0).

Řešení:

Zkontrolujte připojení a napájení GreenBus. Provedte vypnutí a opětovné zapnutí. V případě přetrvávání uvedeného problému se obraťte na servisní službu.

2068 Příjímač číslo: Interní chyba č. číslochyby

Příčina:

Došlo k výskytu interní chyby na uvedeném vzdáleném modulu.

Řešení:

Obraťte se na výrobce stroje.

2069 Příjímač číslo: Chyba napájení +24 V= pracovního stolu číslo

Příčina:

Napájení (+24 V=) jednotky výstupu připojeného ke stejné napájecí svorce není aktivní nebo nefunguje správně.

Řešení:

Zkontrolujte činnost napájení +24 V=.

6.2.3 Chyby vytvořené správou MECHATROLINK-II

2308 Karta ČísloKarty: Inicializace se nezdařila v důsledku nesprávného nastavení parametru konfigurace

Příčina:

V konfiguraci virtuální fyzické nebyla připojena žádná jednotka pohybu v ose (logické zařízení) ke kartě se sběrnici MECHATROLINK-II (fyzické zařízení).

Řešení:

Zkontrolujte propojení v Konfiguraci Virtuální-Fyzické.

2341 Karta ČísloKarty: Číslo servopohonů překračuje maximální dovolený počet

Příčina:

Ke kartě se sběrnici MECHATROLINK-II bylo připojeno více servopohonů, než je nastaveno v příslušné konfiguraci.

Řešení:

Zkontrolujte v konfiguraci systému hodnotu Frekvence Řízení Jednotek pohybu v osách. V tabulce jsou uvedeny správné hodnoty, které je třeba nastavit v závislosti na počtu servořízení řízených z karty.

Karta	Frekvence Řízení Jednotek pohybu v osách (Hz)	Maximální počet servořízení
AlbMech	1000	8
AlbMech	<=500	16
DualMech Mono	1000	8
DualMech Mono	500	20
DualMech Mono	250	30
DualMech	1000	16
DualMech	500	40
DualMech	250	60

2342 Karta ČísloKarty: Adresa hardwaru servopohonu Servo překračuje maximální dovolenou hodnotu

Příčina:

V kartě se sběrnici MECHATROLINK-II byla jedna jednotka pohybu v ose (logické zařízení) připojena k vyšší adrese hardwaru (fyzické zařízení), než je počet servořízení, která lze řídit z karty.

Řešení:

Zkontrolujte v konfiguraci systému hodnotu Frekvence Řízení Jednotek pohybu v osách. V tabulce jsou uvedeny správné hodnoty, které je třeba nastavit v závislosti na počtu servořízení řízených z karty.

Karta	Frekvence Řízení Jednotek pohybu v osách (Hz)	Maximální počet servořízení
AlbMech	1000	8
AlbMech	<=500	16
DualMech Mono	1000	8
DualMech Mono	500	20
DualMech Mono	250	30
DualMech	1000	16
DualMech	500	40
DualMech	250	60

Zkontrolujte v Konfiguraci Virtuální Fyzické spojení mezi logickým zařízením a fyzickým zařízením. Když je například maximální počet pohonů 8, propojení mezi logickým zařízením a fyzickým zařízením musí být provedeno v rámci prvních 8 jednotek pohybu v osách (od Ax1 do Ax8).

2349 Karta ČísloKarty: Servopohon Servo není připojen

Příčina:

Fyzické připojení karty uvedené prostřednictvím položky k servopohonu je přerušeno

Řešení:

Zkontrolujte kabeláže sběrnice MECHATROLINK-II a servopohonu.

6.2.4 Chyby vytvořené správou CanBUS

2761 Uzel číslo: Odpojený

Příčina:

Uvedený uzel CAN se aktuálně jeví jako nepřipojený k průmyslové sběrnici, která se vztahuje na uvedenou kartu, navzdory tomu, že je přítomen v konfiguraci.

2762 Uzel číslo: Znovu připojený

Příčina:

Uvedený uzel CAN se jeví, jakoby byl právě znovu připojen k polní sběrnici, která se vztahuje na uvedenou kartu.

2763 Chyba způsobená chybějícím přenosem

Příčina:

Vnitřní chyba uvedené. Vysílání dat očekávaných uzlem CAN se nezdařilo.

Řešení:

Obraťte se na výrobce stroje.

2764 Uzel číslo: Chyba způsobená chybějícím příjmem

Příčina:

Příjem dat očekávaných uzlem CAN se nezdařil.

Řešení:

Zkontrolujte připojení a napájení uvedeného zařízení CAN. Zkontrolujte kabeláž celého vedení CAN. Zkontrolujte připojení vedení k číslicovému řízení. Zkontrolujte soulad mezi nastaveními protokolu uvedeného zařízení CAN vzhledem k nastavením vysílače, který se nachází na číslicovém řízení (přenosová rychlost, adresa, specifická nastavení přijatého protokolu).

2765 Uzel číslo: Inicializovaný**Příčina:**

Uvedený uzel CAN byl připojen k polní sběrnici. Následně byl inicializován správným způsobem.

2766 Stav poruchy na rozhraní sběrnice CAN**Příčina:**

Je signalizována porucha vnitřního napájení zařízení rozhraní CAN.

Řešení:

Obraťte se na výrobce stroje.

2767 Chyba ztráty stavu CANopen**Příčina:**

Z důvodu vážného problému se vysílač CAN již nenachází v operačním stavu (Operational).

Řešení:

Obraťte se na výrobce stroje.

2768 Uzel číslo: Chyba způsobená chybějícím příjmem PDO**Příčina:**

Číslicové řízení nepřijalo PDO, který očekávalo od uzlu sběrnice CAN.

Řešení:

Zkontrolujte:

- napájení uzlu;
- zda uzel zůstal v režimu pre-Operational.
- údaje PDO a sběrnice CAN, nakonfigurované v systému Albatros.

2769 Uzel číslo: Chyba příjmu nekonfigurovaného uzlu**Příčina:**

Byla zjištěna přítomnost uzlu na síti CAN, který nebyl uveden v konfiguraci hardwaru systému Albatros.

Řešení:

Ověřte hardwarovou adresu uzlu a adresu uzlu nastaveného v konfiguraci hardwaru.

Ověřte, zda je uzel uvedený v konfiguraci hardwaru. Pokud ne, je nutné jej přidat.

2770 Uzel číslo: Chybná konfigurace**Příčina:**

Je chybný popis dat RPDO a TPDO.

Řešení:

Opravte popis dat PDO vysílání a PDO příjmu v konfiguraci hardwaru.

2771 Uzel číslo: Chyba komunikace SDO**Příčina:**

Uvedený uzel CAN neodpověděl v rámci asynchronní komunikace (SDO).

Řešení:

Zkontrolujte stav připojení uzlu. Pokud problém přetrvává, obraťte se na výrobce stroje.

2772 Časový limit cyklu sběrnice CAN dotazování uzlů**Příčina:**

Došlo k výskytu chyby časového limitu cyklu CAN dotazování uzlů

Řešení:

Změňte v konfiguraci hardwaru nastavenou hodnotu doby vzorkování.

3073 Uzel číslo: Chyba Nouzového stavu č. číslochyby

Příčina:

Zařízení CANopen zaznamenalo situaci chyby uzlu vymezenou zobrazeným kódem. Kód chyby je ve formě šestnáctkového čísla.

Jedná se o situace chyby, týkající se jednoho uzlu, které jsou v souladu se standardem CiA DS301-protokol EMERGENCY.

Řešení:

Vycházejte z dokumentace uzlu.

3074 Uzel číslo: Všeobecná chyba sběrnice CAN č. číslochyby

Příčina:

Došlo k výskytu interní chyby na uvedeném uzlu. Kód chyby je ve formě šestnáctkového čísla.

Řešení:

Obraťte se na výrobce.

3088 Karta CAN číslo: Uzel ČísloUzlu: Chyba komunikace SDO č. číslochyby - popis

Příčina:

V instrukci READDICTIONARY nebo WRITEDICTIONARY se nezdařila jedna nebo více žádostí o čtení/zápis SDO. Nezdaření instrukcí může být způsobeno například žádostí o čtení objektu CANOpen, který není implementován v zařízení, na které se žádost vztahuje; nebo může souviset se zápisem data, které není kompatibilní s druhem objektu (například pokus o zápis řetězce do objektu, který je typu integer). Dodaný kód chyby je ve shodě s pravidly DS402 a kromě číselného kódu je dodán také textový popis. Kód chyby je ve formě šestnáctkového čísla.

Řešení:

Zkontrolujte správnost parametrů BaudRate, Doba vzorkování, apod. nastavených v konfiguraci hardwaru a parametrů případných instrukcí [READDICTIONARY](#) a/nebo [WRITEDICTIONARY](#), které se nacházejí v kódu GPL.

6.2.5 Chyby generované provozní EtherCAT sběrnice

3329 Chyba inicializace socketu komunikace

Příčina:

Firmware nedokázal komunikovat se síťovou kartou.

Řešení:

Když byla karta nakonfigurována v systému RTX, ověřte, zda jsou soubory .ini, které se nacházejí v podadresáři FW v softwaru Albatros, napsány správně. Ohledně kontroly syntaxe souborů si přečtěte návod k instalaci RTX do systému Albatros (installationRTXGuide.pdf).

3330 Chyba během skenování sítě EtherCAT

Příčina:

Během předběžného skenování sítě EtherCAT nedostal Master odpověď od některých nebo od všech nakonfigurovaných Slave nebo konfigurace neodpovídá skutečné síti EtherCAT, která se nachází v poli.

Řešení:

Ověřte kabeláž mezi zařízením master EtherCAT a zařízením slave. Zkontrolujte popisy zařízení v konfiguraci hardwaru. Pomocí při identifikaci chyby je okno Diagnostiky hardwaru. Zde jsou zobrazeny přítomné uzly a v případě, že jsou nakonfigurovány chybně, bude kromě nalezeného názvu zařízení zobrazen také název očekávaného zařízení.

3331 Chyba v konfiguraci poštovní schránky vysílání

Příčina:

Uzel sítě EtherCAT neodpovídá na příkaz ze zařízení master. Příčinami mohou být: chybějící komunikace, vadný uzel..

Řešení:

Zkontrolujte kabeláž a vzdálenou činnost.

3332 Chyba v konfiguraci poštovní schránky příjmu**Příčina:**

Uzel sítě EtherCAT neodpovídá na příkaz ze zařízení master. Příčinami mohou být: chybějící komunikace, vadný uzel...

Řešení:

Zkontrolujte kabeláž a vzdálenou činnost.

3333 Karta EtherCAT číslo: Chyba v druhu rozšíření uzlu ČísloUzlu**Příčina:**

Druh rozšíření nakonfigurovaných v uzlu EtherCAT v konfiguraci hardwaru neodpovídá druhu reálně přítomných rozšíření. (Například v konfiguraci hardwaru byl zadefinován TRS-CAT s rozšířením TRS-IO-E, zatímco v systému je přítomen TRS-CAT s rozšířením TRS-AN-E).

Řešení:

Ověřte, zda zařízení popsaná v konfiguraci hardwaru odpovídají přítomným zařízením.

3334 Chyba ve fázi konfigurace PDO**Příčina:**

Uzel EtherCAT, pro který byl zaznamenán pokus o konfiguraci PDO, se nenachází v síti nebo je vadný.

Řešení:

Zkontrolujte, zda konfigurace sítě EtherCAT popsaná v konfiguraci Albatros odpovídá fyzické konfiguraci sítě.

3335 Uzel ČísloUzlu - Alarm (ČísloChyby)**Příčina:**

Uvedený uzel se nachází v situaci alarmu.

Řešení:

Ověřte kód alarmu v níže uvedené tabulce

Kód alarmu	Popis
0x0001	Nespecifikovaná chyba
0x0002	Chybí paměť
0x0011	Chybný požadavek na změnu stavu
0x0012	Neznámý požadovaný stav
0x0013	Nepodporovaný spouštěcí kód
0x0014	Neplatný firmware
0x0015	Chybná konfigurace poštovní schránky
0x0016	Chybná konfigurace poštovní schránky
0x0017	Chybná konfigurace správce synchronizace
0x0018	Nejsou k dispozici žádné platné vstupy
0x0019	Žádné platné výstupy
0x001A	Chyba synchronizace
0x001B	Sledovací zařízení správce synchronizace
0x001C	Chybné druhy správce synchronizace
0x001D	Chybná konfigurace výstupu
0x001E	Chybná konfigurace vstupu
0x001F	Chybná konfigurace sledovacího zařízení
0x0020	Zařízení slave vyžaduje studený start

0x0021	Zařízení slave potřebuje INICIALIZ.
0x0022	Zařízení slave potřebuje PŘÍPR. OP.
0x0023	Zařízení slave potřebuje BEZP. OP.
0x0024	Chybné mapování vstupu
0x0025	Chybné mapování výstupu
0x0026	Nekonzistentní nastavení
0x0027	Volný běh není podporován
0x0028	Synchronizace není podporována
0x0029	Volný běh vyžaduje 3 režimy vyrovnávací paměti
0x002A	Sledovací zařízení na pozadí
0x002B	Žádné platné vstupy a výstupy
0x002C	Závažná chyba synchronizace
0x002D	Žádná chyba synchronizace
0x0030	Chybná konfigurace SYNCHR. DC
0x0031	Chybná konfigurace Nulové polohy motoru DC
0x0032	Chyba PLL
0x0033	Chyba způsobená chybným VST./VÝST DC
0x0034	Chyba způsobená časovým limitem DC
0x0035	Chybná doba cyklu synchronizace DC
0x0036	Doba cyklu synchr. 0 DC
0x0037	Doba cyklu synchr. 1 DC
0x0041	MBX_AOE
0x0042	MBX_EOE
0x0043	MBX_COE
0x0044	MBX_FOE
0x0045	MBX_SOE
0x004F	MBX_VOE
0x0050	Žádný přístup do EEPROM
0x0051	Chyba EEPROM
0x0060	Místní restartování zařízení slave

3336 Karta EtherCAT číslo: Chybný počet rozšíření uzlu ČísloUzlu je chybný

Příčina:

Počet rozšíření nakonfigurovaných na jednom uzlu EtherCAT v Albatros neodpovídá počtu přítomných rozšíření. (Například v konfiguraci hardwaru byl zadefinován TRS-CAT se dvěma rozšířeními TRS-IO-E, zatímco na systému existuje pouze jedno rozšíření).

Řešení:

Ověřte, zda zařízení popsaná v konfiguraci hardwaru odpovídají přítomným zařízením.

3337 Karta EtherCAT: Uzel ČísloUzlu odpojený

Příčina:

Uzel EtherCAT nevrátil do řízení žádnou odpověď. Uzel by mohl být odpojen ze sítě nebo by mohl být vypnutý.

Řešení:

Zkontrolujte kabeláž a napájení uzlu.

3338 Karta EtherCAT: Uzel ČísloUzlu znovu připojený

Příčina:

Uvedený uzel EtherCAT byl znovu připojen k síti a začal reagovat na požadavky řízení.

3340 Karta EtherCAT: Uzel ČísloUzlu neodpověděl na žádost (Kód)

Příčina:

Uzel EtherCAT, dotázaný prostřednictvím služby SDO, neodpověděl na žádost. Kromě toho v případě, že zařízení poskytlo kód pro zrušení, tento kód bude uveden v zobrazeném hlášení chyby (kód). Je možné, že se tato chyba zobrazí během provádění pokynů [SETPZERO](#), [SETPFLY](#) a [FASTREAD](#) na jednotkách pohybu v osách EtherCAT, vzhledem k tomu, že používají komunikaci SDO pro konfiguraci pohonu. V tomto specifickém případě může hlášení chyby kromě standardního kódu pro zrušení obsahovat následující kódy:

- 1: Uplynutí interního časového limitu: uzel neodpověděl do 250 ms.
- 00000005: interní index chybné poštovní schránky.
- 00000006: přijatý údaj neodpovídá žádosti.

Řešení:

Obraťte se na výrobce.

3341 Karta EtherCAT: Uzel ČísloUzlu neexistuje

Příčina:

Byl zaznamenán pokus nechat provést ovládací příkaz neexistujícímu uzlu.

Řešení:

Zkontrolujte číslo uzlu v instrukci GPL, která vytvořila chybu.
Připojte uzel.

3342 Odpojený kabel

Příčina:

Kabel EtherCAT není připojen k řízení.

Řešení:

Zkontrolujte, zda je kabel správně připojen k řízení a zda není poškozen.

3343 Karta EtherCAT číslo: Uzel ČísloUzlu nepřechází do stavu SAFE-OPERATIONAL (Kód)

Příčina:

Uzel EtherCAT nepřešel do stavu SAFE-OPERATIONAL.

Řešení:

Obraťte se na výrobce a oznamte mu uvedenou chybu (kód), která představuje kód ALstatuscode.

3344 Karta EtherCAT číslo: Uzel ČísloUzlu nepřechází do stavu OPERATIONAL (Kód)

Příčina:

Uzel EtherCAT nepřešel do stavu OPERATIONAL.

Řešení:

Obraťte se na výrobce a oznamte mu uvedenou chybu (kód), která představuje kód ALstatuscode.

3345 Karta EtherCAT: Nestabilní komunikace

Příčina:

Komunikace v síti EtherCAT je nestabilní z důvodu rušení nebo vadných kabelů či uzlů.

4400 Příliš mnoho aktivních jednotek pohybu v ose ve FASTREAD (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Byl zaznamenán pokus o použití příkazu FASTREAD s počet jednotek pohybu v osách větším než je dovoleno. Je třeba pamatovat, že jednotky pohybu v ose, které jsou vybavené přídavným snímačem impulzů, odpovídají dvojitě jednotce pohybu v ose (viz příkaz [SWITCHENC](#)).

Řešení:

Snižte počet použitých jednotek pohybu v osách s přídavným snímačem impulzů.

6.2.6 Chyba vytvořená inicializací

769 Chybná konfigurace softwaru

Příčina:

Konfigurace hardwaru vzdáleného modulu neodpovídá konfiguraci softwaru, která je uvedena v konfiguraci systému.

Řešení:

Zkontrolujte soulad mezi parametry hardwaru vzdáleného modulu a parametry softwaru.

770 Chybné nakonfigurované číslo IRQ

Příčina:

Nebylo správně nastaveno IRQ karty jednotek pohybu v osách v konfiguraci Modulu. Obvykle dochází k výskytu konfliktu hardwaru s jinými periferními zařízeními, která se nacházejí v systému.

Řešení:

Zkontrolujte v nastaveních BIOS základní desky, zda je IRQ použité kartou jednotek pohybu v osách vyhrazeno „Legacy ISA“. Zkontrolujte, zda nejsou přítomna další periferní zařízení, která používají stejné IRQ jako to, které bylo přiřazeno kartě jednotek pohybu v osách. Je-li to možné, změňte IRQ periferního zařízení, které je v konfliktu s kartou jednotek pohybu v osách; v opačném případě změňte IRQ jednotky pohybu v osách.

772 Chyba při čtení části vyrovnávací paměti během inicializace

Příčina:

Došlo k výskytu chyby během zkoušek provedených ve fázi inicializace karty jednotek pohybu v osách. Konkrétně se nezdařila zkouška vyrovnávací RAM (Dallas). Může to být důsledkem chybné konfigurace adres VST./VYST. a IRQ karty nebo důsledkem konfliktu s jinými periferními zařízeními, která jsou součástí systému. Kromě toho se může jednat o následek poškození karty jednotek pohybu v osách.

Řešení:

Zkontrolujte konfiguraci hardwaru. Kvalifikovaní technici mohou provést zkoušku Hardwaru RAM mikrokontroleru i296. Všimněte si, že zkouška Hardwaru RAM způsobí vymazání všech dat, která jsou v ní uložena. Ve vyrovnávací RAM jsou uchovávány hodnoty některých zařízení, například počítadel, časovačů a posunů DAC jednotek pohybu v osách. Před provedením zkoušky je třeba provést uložení těchto hodnot. V případě opětovného výskytu problému se obraťte na výrobce.

773 Bylo dosaženo maximálního počtu konfigurovatelných jednotek pohybu v osách

Příčina:

Byla zaznamenána snaha o konfiguraci počtu jednotek pohybu v osách, který překračuje maximální povolený počet.

Řešení:

Snižte nakonfigurovaný počet jednotek pohybu v osách. Obraťte se na výrobce stroje ohledně dalších informací.

774 Nebylo zahájeno provádění v reálném čase (Real-Time) jednotek pohybu v osách

Příčina:

Firmware správy jednotek pohybu v osách byl inicializován, ale nefunguje správně. Obvykle dochází k výskytu konfliktu hardwaru s jinými periferními zařízeními, která se nacházejí v systému.

Řešení:

Zkontrolujte, zda nedochází ke konfliktům s jinými periferními zařízeními, změňte konfiguraci periferních zařízení, která způsobila konflikt, nebo je odstraňte ze systému.

775 Nedostatečná doba pro provedení GPL

Příčina:

Provádění úkolu v reálném čase (real-time) zabírá příliš velkou část doby cyklu. K vytvoření této chyby dochází, když úkol prováděný v reálném čase (real-time) neskončí do začátku následujícího provádění jednotek pohybu v ose v reálném čase (real-time) (např. při vytvoření nekonečného cyklu).

Řešení:

Změňte kód GPL tak, aby došlo ke snížení doby trvání úkolu prováděného v reálném čase (real-time).

776 Nadměrná doba provádění v reálném čase (Real-Time)

Příčina:

Provádění úkolu v reálném čase (real-time) zabírá příliš velkou část doby cyklu. Doba provádění je mírně vyšší, než je maximální dovolený limit.

Řešení:

Změňte kód GPL tak, aby došlo ke snížení doby trvání úkolu prováděného v reálném čase (real-time).

777 Uplynul časový limit sledovacího zařízení

Příčina:

Firmware je zablokován.

Řešení:

Obraťte se na výrobce stroje.

778 Hlavní část kódu (Main) firmwaru je zablokována

Příčina:

Firmware se zablokoval na více než 5 provedení v reálném čase (real-time).

Řešení:

Obraťte se na výrobce stroje.

1025 Karta ČísloKarty: Neodpovídá na příkazy

Příčina:

Při inicializaci byla zaznamenána přítomnost karty jednotek pohybu v osách, ale tato karta neodpovídá správně na ovládací příkazy.

Řešení:

Kvalifikovaní technici mohou provést hardwarovou kontrolu osové karty. V případě opětovného výskytu problému se obraťte na výrobce.

1026 Karta ČísloKarty: Chyba při odesílání firmwaru na osovou kartu

Příčina:

Nebylo možné odeslat firmware do karty.

Řešení:

Obraťte se na výrobce.

1028 Karta ČísloKarty: Není přítomen firmware

Příčina:

Firmware přítomné na kartě nejsou správné pro zaznamenaný druh karty.

Řešení:

Odešlete správný firmware.

1029 Karta ČísloKarty: Zablokovaná hlavní část kódu (MAIN)

Příčina:

Firmware karty nebyl spuštěn.

Řešení:

Obraťte se na výrobce.

1031 Karta ČísloKarty: Chyba inicializace

Příčina:

Došlo k výskytu chyby během postupu inicializace karty jednotek pohybu v osách.

Řešení:

Zkontrolujte a vyřešte příčiny chyb systému, které se vyskytly těsně před výskytem aktuální chyby. Poté proveďte inicializaci systému.

1032 Karta ČísloKarty: Nezdařila se zkouška paměti s duálním portem

Příčina:

Došlo k výskytu chyby během zkoušek provedených ve fázi inicializace karty jednotek pohybu v osách.

Nezdařila se inicializace paměti s duálním portem.

Obvykle je to způsobeno konfliktem hardwaru s ostatními periferními zařízeními, která jsou přítomna v systému, ale může to souviset také s poškozením karty.

Řešení:

Zkontrolujte konfiguraci karty a dále zkontrolujte, zda se nevyskytují konflikty s ostatními periferními zařízeními. V případě použití vzdáleného modulu zopakujte přenos firmwaru do modulu. Kvalifikovaní technici mohou provést zkoušku Hardwaru paměti s duálním portem. V případě opětovného výskytu problému se obraťte na výrobce.

1033 Karta ČísloKarty: Neprobíhá provádění spouštěcího kódu (Boot) firmwaru

Příčina:

Firmware spouštění karty byl inicializován, ale nefunguje správně. Obvykle dochází k výskytu konfliktu hardwaru s jinými periferními zařízeními, která se nacházejí v systému.

Řešení:

Zkontrolujte konfiguraci karty a dále zkontrolujte, zda se nevyskytují konflikty s ostatními periferními zařízeními.

V případě použití vzdáleného modulu zopakujte přenos firmwaru do modulu. Kvalifikovaní technici mohou provést zkoušku Hardwaru paměti s duálním portem, která se nachází na kartě. V případě opětovného výskytu problému se obraťte na výrobce.

1035 Karta ČísloKarty: Nepřítomná

Příčina:

Došlo k výskytu chyby během zkoušek provedených ve fázi inicializace karty jednotek pohybu v osách. Konkrétně nebyla zaznamenaná karta.

Řešení:

Zkontrolujte, zda je karta přítomna v systému a zda není poškozena. Kvalifikovaní technici mohou provést zkoušku hardwaru karty. V případě opětovného výskytu problému se obraťte na výrobce.

1037 Karta ČísloKarty: Nezdařilo se otevření paměti s duálním portem

Příčina:

Otevření paměti s duálním portem karty se nezdařilo.

Řešení:

Kvalifikovaní technici mohou provést zkoušku hardwaru karty. obraťte se na výrobce.

1039 Karta ČísloKarty: Uplynul časový limit sledovacího zařízení

Příčina:

Firmware na kartě jednotek pohybu v osách ČísloKarty je zablokován.

Řešení:

Obraťte se na výrobce stroje.

1040 Karta ČísloKarty: Chyba napájení +24 V=

Příčina:

Polní napájení (+24 V=) výstupů není přítomno nebo nefunguje správně.

Řešení:

Zkontrolujte činnost napájení +24 V= pole.

1047 Karta ČísloKarty: Nedovolená konfigurace softwaru

Příčina:

Zařízení přijalo konfiguraci, která není kompatibilní s přítomným nebo aktivovaným hardwarem. Například se může jednat o požadavek na konfiguraci jednotky pohybu v ose, která není aktivována nebo není přítomna na zařízení.

Řešení:

Zkontrolujte soulad mezi parametry hardwaru karty a parametry softwaru.

1052 Karta ČísloKarty: Provádí se spouštěcí kód (Boot)

Příčina:

Karta je nastavena v provizorním režimu a provádí spouštěcí kód.

Řešení:

Obraťte se na výrobce.

1053 Karta ČísloKarty: Uplynul časový limit sledovacího zařízení jednotek pohybu v osách

Příčina:

Došlo k výskytu vážné chyby během provádění firmwaru karty řízení jednotek pohybu v osách. Jednotky pohybu v osách budou zakázány a případný signál SYSOK bude vypnut. Neprovádějte obnovení činnosti systému.

Řešení:

Obraťte se na výrobce stroje.

1055 Uplynula kontrolní doba („Watchdog“) na karte ČísloKarty

Příčina:

Firmware na kartě ČísloKarty je zablokován.

Řešení:

Obraťte se na výrobce stroje.

1056 Karta ČísloKarty: Chyba napájení rozhraní sběrnice CAN

Příčina:

Vysílací zařízení vedení CanBus, které se nachází na uvedené kartě, vykazuje chybu napájení. Může to souviset s přítomností zkratu, s chybou kabeláže sběrnice nebo s poškozením karty.

Řešení:

Zkontrolujte kabeláž celého vedení CAN. Zkontrolujte připojení vedení k číslíkovému řízení. Odstraňte případný zkrat. V případě, že by nedošlo k obnovení komunikace, obraťte se na výrobce stroje.

1057 Karta ČísloKarty: Interní chyba číslo ČísloChyby

Příčina:

Chyba v hardwaru uzlu.

Řešení:

Obraťte se na výrobce stroje.

6.2.7 Chyby vytvořené správou paměti

1281 Chyba přidělení paměti v oblasti haldy

Příčina:

Dostupná vyrovnávací paměť RAM je menší, než je vyžadováno například globální maticí.

Řešení:

Zmenšíte rozměr globálních proměnných přidělených v RAM.

1286 Chyba správy haldy

Příčina:

Chyba ve správě paměti ze strany firmwaru.

Řešení:

Obraťte se na výrobce stroje.

1287 Příliš mnoho přidělení paměti z haldy

Příčina:

Chyba ve správě paměti ze strany firmwaru.

Řešení:

Obraťte se na výrobce stroje.

1289 Chyba při vytváření globálních proměnných

Příčina:

Byl definován příliš vysoký počet [HID_GPL_VALCHAR_2](#) nebo byly definovány globální matrice příliš velkých rozměrů.

Řešení:

Snižte počet globálních proměnných nebo rozměr matic.

1290 Chyba rozměru stálých proměnných

Příčina:

Byl definován příliš vysoký počet stálých proměnných nebo byly definovány stálé matrice příliš velkých rozměrů.

Řešení:

Snižte počet stálých proměnných nebo rozměr stálých matic.

1291 Chyba rozměru proměnných jen pro čtení

Příčina:

Byl definován příliš vysoký počet proměnných jen pro čtení nebo byly definovány matrice jen pro čtení příliš velkých rozměrů.

Řešení:

Snižte počet proměnných jen pro čtení nebo rozměr matic jen pro čtení.

6.2.8 Chyby vytvořené poruchami

1559 Trasy zarážky

Příčina:

Vážná chyba firmwaru.

Řešení:

Obraťte se na výrobce stroje.

1569 Neplatný operační kód mikroprocesoru

Příčina:

Mikroprocesor narazil na neznámou instrukci. Může to souviset s problémy hardwaru PC i s poškozením souborů, které jsou obsaženy ve firmwaru řízení Albatros.

Řešení:

V případě místního modulu zkontrolujte, zda soubory nejsou poškozené a případně přeinstalujte Albatros. V případě vzdálených modulů proveďte aktualizaci firmwaru. Proveďte zkoušku hardwaru PC, zejména RAM. Pokud problém přetrvává, obraťte se na výrobce stroje.

1586 Hodnota celého čísla (INTEGER) vydělená nulou

Příčina:

Byl zaznamenán pokus o dělení celého čísla (INTEGER) nulou.

Řešení:

Zkontrolujte ve funkcích GPL, zda jsou všechna dělení správná.

1600 Přetečení ve výsledku operace s plovoucí desetinnou čárkou

Příčina:

Výsledek operace mezi hodnotami s plovoucí desetinnou čárkou (FLOAT) je větší, než je kapacita adresáta:

$\pm 3,402823E+38$	pro float
$\pm 1,79769313486231E+308$	pro double.

Řešení:

Zkontrolujte ve funkcích GPL, zda jsou výpočty týkající se hodnot s plovoucí desetinnou čárkou (float) správné.

1601 Podtečení ve výsledku operace s plovoucí desetinnou čárkou

Příčina:

Výsledek operace mezi hodnotami s plovoucí desetinnou čárkou (FLOAT) je menší, než je kapacita adresáta:

$\pm 1,401298E-45$	pro float
$\pm 4,94065645841247E-324$	pro double.

Řešení:

Zkontrolujte ve funkcích GPL, zda jsou výpočty týkající se hodnot s plovoucí desetinnou čárkou (float) správné.

1602 Neplatný argument pro operaci s plovoucí desetinnou čárkou

Příčina:

V jedné operaci týkající se hodnot s plovoucí desetinnou čárkou (float) byl použit jiný operand než operand typu float.

Řešení:

Zkontrolujte ve funkcích GPL, zda jsou výpočty týkající se hodnot s plovoucí desetinnou čárkou (float) správné.

1603 Hodnota s plovoucí desetinnou čárkou vydělená nulou

Příčina:

Byla zaznamenána snaha o vydělení hodnoty s plovoucí desetinnou čárkou (float) nebo hodnoty celého čísla s dvojitou přesností (double) nulou. Tato chyba bude vytvořena i v případě provedení logaritmu nuly.

Řešení:

Zkontrolujte ve funkcích GPL, zda jsou všechna dělení správná.

1604 Nepřesný výsledek operace s plovoucí desetinnou čárkou

Příčina:

Výsledek operace mezi hodnotami s plovoucí desetinnou čárkou (float) je nesprávný.

Řešení:

Zkontrolujte ve funkcích GPL, zda jsou výpočty týkající se hodnot s plovoucí desetinnou čárkou (float) správné.

1605 Byla použita chybná hodnota s plovoucí desetinnou čárkou

Příčina:

Byla použita menší hodnota s plovoucí desetinnou čárkou, než je minimální zastupitelná hodnota:

$\pm 1,401298E-45$ pro float
 $\pm 4,94065645841247E-324$ pro double.

Řešení:

Zkontrolujte ve funkcích GPL, zda jsou výpočty týkající se hodnot s plovoucí desetinnou čárkou (float) správné.

1728 Byl zaznamenán pokus o přístup na neplatnou adresu

Příčina:

Program provedl přístup do neplatného prostoru paměti.

Řešení:

Zkontrolujte případné nesrovnalosti globálních/místních proměnných; při přetrvávání problému oznamte poruchu.

1735 Všeobecná výjimka

Příčina:

Byla zaznamenána neznámá výjimka.

Řešení:

Obraťte se na výrobce stroje.

1736 Údaje nejsou v souladu

Příčina:

Vážná chyba firmwaru.

Řešení:

Obraťte se na výrobce stroje.

1801 Alarm teploty

Příčina:

Teplota CPU řízení překročila maximální dovolené limity.

Řešení:

Zkontrolujte, zda se nevyskytují problémy s ventilací nebo jiné důvody přehřátí. V případě opětovného výskytu problému se obraťte na výrobce.

1802 Alarm ventilátoru

Příčina:

Ventilátor CPU řízení nefunguje správně. Uvedený problém může v průběhu krátké doby způsobit přehřátí CPU.

Řešení:

Obraťte se na výrobce stroje.

1803 Frekvence CPU je nestabilní

Příčina:

Pracovní frekvence CPU není stabilní.

Řešení:

Obraťte se na výrobce stroje.

6.2.9 Chyby vytvořené instrukcemi GPL

4097 Zařízení DruhZařízení NázevZařízení není nakonfigurováno

Příčina:

Instrukce GPL použila nenakonfigurované zařízení, tj. zařízení, které nemá Virtuální-Fyzické spojení. Může být vytvořeno všemi instrukcemi, kterým je jako parametr předáno zařízení.

Řešení:

Zkontrolujte v kontrolních konfiguracích, zda všechna zařízení používaná funkcemi mají Virtuální-Fyzické spojení. Poté znovu přeneste konfigurace do karty.

4098 Globální proměnná NázevProměnné neexistuje

Příčina:

Instrukci GPL byla jako argument předána globální proměnná, která nebyla definována. Obvykle k tomu dojde v případě, kdy řízení nebylo správně inicializováno.

Řešení:

Znovu proveďte kompilaci veškerého kódu GPL a znovu inicializujte řízení.

4099 Funkce NázevFunkce nebyla nalezena

Příčina:

Bylo provedeno vyvolání funkce, která není přítomna. Může se vyskytnout v případě, kdy následkem změny kódu GPL nebyla provedena inicializace řízení.

Řešení:

Znovu proveďte kompilaci veškerého kódu GPL a znovu inicializujte řízení.

4101 Nevyhovující správa jednotky pohybu v ose NázevJednotkyPohybuOse

Příčina:

Byl provedena nedovolená změna stavu na jednotce pohybu v ose. Ohledně změn stavu si přečtěte příslušnou dokumentaci.

Chyba může být vytvořena všemi instrukcemi, které spravují jednotky pohybu v ose, a obvykle se vyskytuje v těchto případech:

- Při snaze o interpolaci a koordinaci s jednotkou pohybu v ose, která je již obsazena v pohybu z bodu do bodu (nebo opačně).
- Při provedení instrukce Chain, SetPFly, SetPZero na jednotce pohybu v ose, která se nachází v transparentním režimu (transparent mode).
- Při snaze o interpolaci a koordinaci s jednotkou pohybu v ose, která již vystupuje jako slave jiného zařízení.

Řešení:

Zkontrolujte, zda všechny pohyby jednotek pohybu v osách skončily instrukcí čekání v dosažené poloze, zejména zda jednotky pohybu v ose střídají pohyby různého druhu (z bodu do bodu, interpolace apod.).

4105 Danou instrukci nelze provést na jednotce pohybu v ose NázevJednotkyPohybuOse

Příčina:

Byla zaznamenána snaha o provedení instrukce na jednotce pohybu v ose, která ji nepodporuje. Například instrukce interpolace na jednotce krokového pohybu v ose.

Řešení:

Provedte korekci kódu GPL.

4106 Dálkové zařízení přiřazené k jednotce pohybu v ose v krokovém režimu NázevJednotkyPohybuOse není připojeno

Příčina:

Byla zaznamenána snaha o používání krokové jednotky pohybu v ose, která není připojena k řízení.

Řešení:

Zkontrolujte zapojení vzdáleného modulu, který řídí uvedenou jednotku pohybu v ose.

4107 Instrukce SYSOK s chybnými argumenty

Příčina:

Byla provedena instrukce SYSOK s chybnými argumenty. Dochází k tomu v případě, kdy není správně nakonfigurován jeden nebo více digitálních výstupů předaných jako argumenty instrukci.

Řešení:

Zkontrolujte kód GPL a Virtuální-fyzickou konfiguraci.

4108 Jednotka pohybu v ose NázevJednotkyPohybuOse: Finální cílová poloha mimo softwarové rozmezí

Příčina:

Byla zaznamenána snaha o pohyb jednotky pohybu v ose mimo rozmezí nastavené v konfiguraci nebo kódem GPL.

Řešení:

Provedte korekci programu obrábění, který způsobil chybu. Případně provedte korekci kódu GPL nebo konfigurace jednotky pohybu v ose.

4110 Chybná rychlost

Příčina:

Byla zaznamenána snaha o přiřazení nulové nebo záporné rychlosti jednotce pohybu v ose.

Řešení:

Provedte korekci kódu GPL.

4111 Záporné zrychlení jednotky pohybu v ose **NázevJednotkyPohybu**Ose

Příčina:

Byla zaznamenána snaha o přiřazení záporného zrychlení jednotce pohybu v ose.

Řešení:

Provedte korekci kódu GPL.

4112 Záporné zpomalení jednotky pohybu v ose **NázevJednotkyPohybu**Ose

Příčina:

Byla zaznamenána snaha o přiřazení záporného zpomalení jednotce pohybu v ose.

Řešení:

Provedte korekci kódu GPL.

4114 Jednotka pohybu v ose **NázevJednotkyPohybu**Ose: Neprovedené vynulování rychlého vstupu

Příčina:

Vynulování polohy na rychlém vstupu (okamžité vynulování) nebylo dokončeno správně. Tento postup umožňuje vynulovat polohu jednotky pohybu v ose během pohybu v okamžiku, kdy dojde ke změně stavu odpovídajícího rychlého vstupu. Když jednotka pohybu v ose dokončí probíhající pohyb, aniž by došlo k přepnutí vstupu, bude vytvořena chyba systému. Příčinou může být chybné nastavení parametrů pohybu jednotky pohybu v ose nebo problém kabeláže rychlého vstupu.

Řešení:

Zkontrolujte kód GPL, který implementuje okamžité vynulování, a zkontrolujte kabeláž rychlého vstupu.

4115 Jednotka pohybu v ose **NázevJednotkyPohybu**Ose: Nenalezená značka nuly

Příčina:

Vynulování polohy na značce nuly snímače impulzů neproběhlo správně. Tento postup umožňuje vynulovat polohu jednotky pohybu v ose během pohybu v okamžiku, kdy dojde k zaznamenání značky nuly snímače impulzů. Když jednotka pohybu v ose dosáhne polohy vyhledání značky nuly, aniž by došlo k jejímu zaznamenání, bude vytvořena chyba systému. Příčinou může být chybné nastavení parametrů pohybu jednotky pohybu v ose nebo problém kabeláže signálu značky (fáze C konektoru jednotky pohybu v ose).

Řešení:

Zkontrolujte kód GPL, který implementuje vynulování na značce, a zkontrolujte kabeláž jednotky pohybu v ose.

4353 Neznámá instrukce operačního kódu (**Funkce:NázevFunkce** řádek:**ČísloŘádku**)

Příčina:

Při provádění funkce GPL byla zaznamenána nelegální instrukce. Obecně platí, že je příznakem poškození souborů, které obsahují kompilovaný kód GPL. Pokud byla provedena aktualizace softwaru řízení, zkontrolujte, zda byla provedena opětovná kompilace. Předchozí verze softwaru řízení by totiž mohla obsahovat instrukce, které již nejsou podporovány novou verzí.

Řešení:

Znovu proveďte kompilaci veškerého kódu GPL a znovu inicializujte řízení. Pokud problém přetrvává, obraťte se na výrobce.

4354 Chybná matematická operace (**Funkce:NázevFunkce** řádek:**ČísloŘádku**)

Příčina:

Byla zaznamenána snaha instrukce GPL o provedení chybné matematické operace, například dělení nulou. Nebo jsou některá data zavedená do instrukcí GPL nevyhovující. Obvykle je tato chyba vytvářena instrukcemi pohybu v rámci interpolace, protože právě část firmwaru provádí matematické výpočty.

Řešení:

Zkontrolujte, zda jsou správné parametry, které byly předány instrukcím interpolace. Když problém přetrvává, oznamte nesprávnou činnost výrobci.

4355 Chybná adresa matrice nebo vektoru (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Byla zaznamenána snaha instrukce GPL o přístup k prvku pole nebo matrice, která překračuje maximální rozměr. Například byl zaznamenán pokus o přístup k prvku 10 pole s 5 prvky. Může být vytvořena všemi instrukcemi, které akceptují v úloze parametru pole nebo matici.

Řešení:

Zkontrolujte, zda se všechny indexy matrice a pole předané instrukcím nacházejí uvnitř rozměrů pole a matrice.

4356 Instrukce RET nebyla vyvolána z CALL (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Byla provedena instrukce RET, aniž by se v zásobníku nacházela příslušná zpětná adresa. Nejběžnější příčinou může být stanovení podprocedury před instrukcí FRET výstupu z funkce bez jejího ochránění s GOTO, která zabrání jejímu náhodnému provedení. Nebo byl proveden nechtěný skok dovnitř podprocedury.

Řešení:

Zkontrolujte tok programu GPL. Podprocedury umístěte v rámci možností přednostně na konec těla funkcí (po instrukci FRET).

4357 Neexistující místní proměnná (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Byla zaznamenána snaha Instrukce GPL o přístup k místní proměnné, která nebyla přidělena.

Řešení:

Provedte opětovnou kompilaci a přenos všech funkcí do karty. Když problém přetrvává, oznamte nesprávnou činnost.

4358 Neexistující návěstí skoku (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Instrukce GPL provedla skok na neexistující návěstí. Může být vytvořena funkcemi GOTO, CALL, FCALL a všemi IF.

Řešení:

Provedte opětovnou kompilaci a přenos všech funkcí do karty. Když problém přetrvává, oznamte nesprávnou činnost.

4359 Chybný argument makra (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Instrukci GPL byly předány neplatné argumenty. Může být vytvořena všemi funkcemi. Avšak ve většině případů se systém GPL snaží automaticky korigovat tuto situaci provedením automatických konverzí druhu (cast), které však představují časovou ztrátu. Chyba bude vytvořena tehdy, když tyto konverze nejsou možné, a to zejména v níže uvedených případech:

- Instrukce, které působí na specifických zařízeních (SETTIMER, SETCOUNTER) a kterým je předán odlišný druh.
- instrukce, které působí na bitu a kterým je předáno číslo s plovoucí desetinnou čárkou (AND, OR apod.)
- instrukce, které působí na maticích nebo polích a kterým je předána jednoduchá proměnná (SORT, MOVEMAT apod.)
- instrukce, které působí na řetězcích a kterým nejsou předány řetězce.

K vytvoření chyby dojde také při snaze provést instrukci na kartě, která tuto instrukce nespravuje. (Například instrukce [SENDPDO](#) nebo instrukce [RECEIVEPDO](#) na kartě odlišné od TMSCan nebo TMSCan+).

Řešení:

Provedte korekci kódu GPL.

4360 Chyba v přidělení paměti během provádění (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Funkce GPL se snažila přidělit prostor v paměti pro vnitřní použití, ale nenašla dostupnou paměť. Může se stát, že signalizace bude poukazovat na fyziologickou situaci způsobenou například nadměrně velkým počtem současně prováděných úkolů nebo globálními proměnnými s příliš velkým rozměrem.

Řešení:

Zkontrolujte rozměrový návrh globálních a místních proměnných a zkuste omezit jejich rozměry. Ověřte, zda není současně prováděno příliš mnoho úkolů, a je-li tomu tak, omezte jejich počet.

4361 Příliš mnoho aktivních úkolů (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Byla zaznamenána snaha o spuštění provádění více než 256 úkolů současně.

Řešení:

Snižte počet současně aktivních úkolů.

4362 Chybný formát matrice (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Instrukce, která působí na matici, našla neplatný formát. K instrukcím, které mohou vytvářet tuto chybu systému, patří:

- MOVEMAT v případě, že se formát zdrojové a cílové matrice neshodují.
- CLEAR, když má být vynulován řádek matrice, který neexistuje.
- GETAXIS v případě, že formát matrice poskytnutý parametru neodpovídá očekávanému formátu (přečtěte si dokumentaci jazyka GPL).

Řešení:

Zkontrolujte uvnitř úkolu, který vytvořil chybu, výše uvedené instrukce. Zkontrolujte zejména, zda matrice, na které byla aplikována instrukce MOVEMAT, mají stejný počet sloupců stejného druhu a zda má matrice, na kterou byla aplikována funkce GETAXIS, správný formát.

4363 Příliš mnoho aktivních instrukcí ONINPUT (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Bylo aktivováno více než 128 instrukcí OnInput.

Řešení:

Snižte počet ONINPUT.

4364 Jednotka pohybu v ose je již obsazena v místním odkazu (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Chyba se vztahuje na aktivaci trojic jednotek rotačního-posuvného pohybu v osách za účelem interpolace na větším počtu kartézských os.

Byla zaznamenána snaha o provedení instrukce SETRIFLOC, přičemž jí byla předána jednotka pohybu v ose, která je již vázána v trojici vztažných jednotek pohybu v ose. Tato chyba vzniká také v případě, že je provedena instrukce RESRIFLOC na jednotce pohybu v ose, která nebyla vázána v žádné trojici jednotek pohybu v ose. Posledním případem vytvoření této chyby je případ, kdy již nejsou k dispozici žádné dostupné vztažné trojice (maximálně jich může být 32).

Řešení:

Zkontrolujte, zda trojice předané instrukci SETRIFLOC nemají společné jednotky pohybu v ose.

Zkontrolujte RESRIFLOC.

Dále zkontrolujte, zda neexistují čekací instrukce v příslušné poloze ještě před RESRIFLOC.

V každém případě vezměte v úvahu, že dokud interpolace neskončí, RESRIFLOC nebude skutečně provedena.

4365 Instrukce ONINPUT byla aktivována na stejném vstupu (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Instrukci ONINPUT byl předán stejný příznak více než jednou.

Řešení:

Zkontrolujte, zda neexistují dvě ONINPUT, kterým je předán jako parametr stejný vstup.

4366 Příliš mnoho aktivních instrukcí ONFLAG (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Bylo aktivováno více než 128 instrukcí OnFlag.

Řešení:

Snižte počet ONFLAG.

4367 Instrukce ONFLAG byla aktivována na stejném PŘÍZNAKU (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Instrukci ONFLAG byl předán stejný příznak více než jednou.

Řešení:

Zkontrolujte, zda neexistují dvě ONFLAG, kterým je předán jako parametr stejný příznak.

4368 Pokus o zápis proměnné určené pouze ke čtení - ReadOnly (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Byla zaznamenána snaha o přístup do proměnné určené pouze ke čtení (readonly) v rámci zápisu. Proměnné určené pouze ke čtení jsou vždy globální a jsou uloženy v řízení v paměti typu FLASH. V editoru globálních proměnných se označují jako „statické“ (static). Při pokusu o zápis na tyto globální proměnné bude vytvořena chyba systému.

Chyba bude dále vytvořena v případě použití proměnných, které jsou uloženy ve vyrovnávací („stálé“) paměti RAM jako argument některých instrukcí při zápisu.

Například v příkazu COORDIN je předaná proměnná za účelem uvedení toho, že se zpracováváný řádek musí nacházet v RAM.

Řešení:

Zkontrolujte všechny proměnné typu static ve stálé paměti.

4369 Příliš mnoho aktivních jednotek pohybu v osách typu master (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Byla zaznamenána snaha o aktivaci více než čtyř jednotek pohybu v osách typu master současně. Tato chyba je vytvářena pouze při provádění instrukce CHAIN.

Řešení:

Snižte počet jednotek pohybu v osách typu master.

4370 Příliš mnoho aktivních jednotek pohybu v osách typu slave (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Byla zaznamenána snaha o aktivaci více než osmi jednotek pohybu v osách typu slave jediné jednotky pohybu v ose typu master.

Tato chyba je vytvářena při provádění instrukce CHAIN.

Řešení:

Snižte počet jednotek pohybu v osách typu slave.

4372 Chybné použití instrukce (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Tato chyba vzniká v jedné z následujících situací:

1. používá se instrukce pro správu mailbox (sendmail, waitmail, endmail, ifmail) nebo instrukce pro správu IPC (sendipc, testipc, waitipc) uvnitř funkce volání z instrukce Errsys, OnInput nebo OnFlag.
2. používá se instrukce IfError nebo instrukce IfMessage bez aktivace správy alarmů a stavu.
3. používá se instrukce Watchdog, aniž by byla součástí karta TMSWD.
4. Parametry zadefinované v instrukci interpolace (linearinc, linearabs, circle, circinc, circabs, helicinc a helicabs) nejsou koherentní. Například prohlášený počet jednotek pohybu v osách se liší od prohlášeného počtu poloh nebo bylo prohlášeno více jednotek pohybu v osách, než instrukce může spravovat.

Řešení:

Níže jsou popsána řešení pro každý důvod uvedený v seznamu.

1. Přemístěte instrukci, která způsobuje chybu v jiné funkci, nebo odstraňte danou instrukci.
2. Zkontrolujte, zda je aktivována správa alarmů podle stavu. V souboru tpa.ini, v části [ALBATROS], pod položkou AlarmsHaveState musí být přiřazena hodnota 1.
3. Odstraňte instrukci WatchDog nebo si zajistěte kartu TMSWD.
4. Zkontrolujte, zda jsou parametry instrukce GPL správné. Každé jednotce pohybu v ose musí odpovídat příslušná poloha. Počet jednotek pohybu v osách prohlášený v instrukci nesmí být větší než počet jednotek pohybu v osách, který funkce může spravovat. Například instrukce LINEARABS může spravovat až do 6 jednotek pohybu v osách. Když jich v rámci parametrů není prohlášeno více než 6, instrukce vytváří chybu systému.

4373 Nelze provést čtení rychlosti posuvu (Feed rate) (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Došlo k použití instrukce [GETFEED](#) na kartě TmsBus nebo TmsCan, která není Master.

Řešení:

Zkontrolujte v konfiguraci hardwaru, zda je karta, ke které je připojen signál Feedrate, typu Master.

4374 Příliš mnoho prováděných instrukcí typu IPC (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Došlo k překročení maximálního limitu 16 současně prováděných instrukcí IPC.

Řešení:

Změňte kód GPL.

4375 FASTREAD provedena na jednotkách pohybu v osách odlišných karet (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Byla zaznamenána snaha o provedení instrukce FASTREAD, přičemž jí byly předány jako parametry jednotky pohybu v osách, které nejsou všechny připojeny ke stejné kartě.

Řešení:

Změňte vhodně kód GPL a konfiguraci Virtuální-Fyzické.

4378 Neaktivovaná instrukce (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Byla zaznamenána snaha o použití instrukce, jejíž provádění není aktivováno. Hardwarový klíč pravděpodobně není správně zasunutý nebo chybí.

Řešení:

Správně zasuňte hardwarový klíč. Pokud problém přetrvává, obraťte se na výrobce.

4379 Instrukce nepoužitelná ve funkcích spuštěných přerušením - Interrupt (Funkce:NázevFunkce řádek:ČísloŘádku)**Příčina:**

Byla zaznamenána snaha o použití instrukce nedovolené v rámci funkce spuštěné přerušením (interrupt). Funkce spuštěné přerušením (interrupt) jsou ty, které byly předány úloze parametru instrukcím ONERSYS, ONINPUT a ONFLAG.

Řešení:

Změňte kód GPL. Přečtěte si [seznam instrukcí nepoužitelných v rámci přerušení \(interrupt\)](#)

4380 Příliš mnoho požadavků na zápis do oblasti vyrovnávací paměti (Funkce:NázevFunkce řádek:ČísloŘádku)**Příčina:**

Byla zaznamenána snaha o provedení příliš mnoha operací zápisu do vyrovnávací paměti současně (vyrovnávací paměť je charakterizována poměrně vysokou dobou přístupu).

Řešení:

Zkontrolujte instrukce, které provádějí operace na proměnných přidělených ve vyrovnávací paměti: počítadla, časovače, matrice a proměnné stanovené jako „nonvolatile“.

4381 Není možné používat sériové vedení, které ještě nebylo otevřeno (Funkce:NázevFunkce řádek:ČísloŘádku)**Příčina:**

Byla zaznamenána snaha o provedení instrukce, která působí na sériový port ještě před provedením instrukce COMOPEN pro stejný port.

Řešení:

Změňte kód GPL.

4382 Není možné otevřít sériové vedení, které je již otevřeno (Funkce:NázevFunkce řádek:ČísloŘádku)**Příčina:**

Byla zaznamenána snaha o provedení instrukce COMOPEN na sériovém portu, který již byl otevřen stejnou instrukcí.

Řešení:

Změňte kód GPL.

4383 Byla zaznamenána snaha o otevření příliš velkého počtu pomocných procesů (Funkce:NázevFunkce řádek:ČísloŘádku)**Příčina:**

Byla zaznamenána snaha o otevření více než 4 pomocných procesů současně.

Řešení:

Změňte kód GPL.

4384 Pomocný proces není v oběhu (Funkce:NázevFunkce řádek:ČísloŘádku)**Příčina:**

Byla zaznamenána snaha o přístup do pomocného procesu, který není prováděn.

Řešení:

Změňte kód GPL.

4385 Byla zaznamenána snaha o otevření pomocného procesu z jiného úkolu (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Byla zaznamenána snaha o otevření pomocného procesu z jiného úkolu než z toho, který spustil provádění. Pomocný proces se dá použít pouze z úkolu, který začal jeho provádění.

Řešení:

Změňte kód GPL.

4391 Chyba během aktivace SYSOK (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Aktivace signálu SYSOK nebyla správně ukončena. Obvykle to může souviset s nesprávnou činností vysílače Greenbus na kartě jednotek pohybu v osách.

Řešení:

Kvalifikovaní technici mohou provést zkoušku Hardwaru paměti s duálním portem mikrokontroleru i296. V případě opětovného výskytu problému se obraťte na výrobce.

4394 Příliš mnoho chyb cyklu (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Je aktivních více než 2000 chyb cyklu.

Řešení:

Omezte počet signalizací korekcí kódu GPL.

4395 Příliš mnoho hlášení (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Je aktivních více než 2000 hlášení.

Řešení:

Omezte počet signalizací korekcí kódu GPL.

4397 Přetečení zásobníku (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Zásobník funkce GPL překročil maximální limit 2 Kbyte.

Řešení:

Provedte opětovnou kompilaci kódu GPL a zkontrolujte v záznamu kompilátoru odhad obsazení zásobníku funkcí, který způsobil chybu systému. Zmenšete počet místních proměnných a parametrů předaných funkcím (například jejich nahrazením globálními proměnnými). Zmenšete počet VYVOLÁNÍ.

4398 Podtečení zásobníku (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Může se vyskytnout pouze vážná chyba Firmware, například v případě chybné správy parametrů funkce nebo místních proměnných.

Řešení:

Obraťte se na výrobce.

4399 Parametr mimo rozsah (Funkce:NázevFunkce řádek:ČísloŘádku)

Příčina:

Proměnné GPL nebo zařízení byla přiřazena hodnota, která se nenachází v dovoleném rozsahu.

Řešení:

Provedte korekci a opětovnou kompilaci kódu GPL.

4865 Chybí stanovení stroje pro interpolaci (G216 nebo G217)

Příčina:

Byla zaznamenána snaha o pohyb jednotek pohybu v osách s interpolací ISO nebo byly nastaveny indexy konfigurace, aniž by byly nejdříve definovány matrice konfigurace a jednotky pohybu v osách, které tvoří stroj.

Řešení:

Provedte korekci a opětovnou kompilaci kódu GPL s použitím instrukce [ISOG216](#).

4866 Chybí stanovení indexů konfigurace stroje při uchopení (M6)

Příčina:

Byla zaznamenána snaha o pohyb jednotek pohybu v osách s interpolací ISO, aniž by byly nejdříve definovány indexy matic konfigurace stroje

Řešení:

Provedte korekci a opětovnou kompilaci kódu GPL s použitím instrukce [ISOM6](#).

6.2.10 Chyby vytvořené ovladačem komunikací CNCTPA

16385 Odpojený modul

Příčina:

Propojení Dozorného PC s modulem bylo přerušeno.

K možným příčinám patří:

- Chybějící napájení vzdáleného modulu
- přerušení připojení kabelů sítě ethernet, včetně dočasného, způsobené nespolehlivým kontaktem v konektorech nebo poškozením samotných kabelů
- chybějící napájení nebo nesprávná činnost rozbočovače sítě Ethernet (je-li součástí)
- zablokování firmwaru vzdáleného modulu způsobené poškozením souborů konfigurace
- vynulování CPU vzdáleného modulu způsobené přehřátím nebo rušením EM

Řešení:

Zkontrolujte, zda je modul zapnutý. Zkontrolujte kabely a konektory sítě Ethernet. Provedte aktualizaci firmwaru ve vzdáleném modulu. Zkontrolujte, zda není vystaven přehřívání v důsledku nedostatečné ventilace a zda není vystaven EM rušení. Pokud problém přetrvává, obraťte se na výrobce stroje.

16386 Připojený modul

Příčina:

Vzdálený modul byl připojen k Dozornému PC po fázi inicializace řízení Albatros. Během spuštění se Albatros snaží připojit všechny moduly stanovené v Konfiguraci Systému. Tato fáze trvá přibližně 4 sekundy a moduly, které se připojí později, způsobují vznik chyby.

16387 Znovu připojený modul

Příčina:

Vzdálený modul se znovu připojil k Dozornému PC poté, co byl odpojen. Tato chyba je proto vždy následkem chyby 16385 „Odpojený modul“.

16388 Inicializovaný modul

Příčina:

Vzdálený modul byl znovu inicializován během běžné činnosti. Z toho vyplývá, že modul byl odpojen a znovu připojen k Dozornému PC. Tato chyba je proto vždy následkem chyby 16385 „Odpojený modul“.

Tato chyba informuje vykonané vynulování modulu způsobené například chybějícím napájením.

16389 Modul přerušil spojení

Příčina:

Vzdálený modul přerušil spojení se řízením Albatros. Tento problém se vyskytuje v případě, že modul delší dobu nepřijímá příkazy nebo dotazování ze strany dozorného PC. Tato chyba proto poukazuje na problém (výrazné zpomalení nebo zablokování) na Dozorném PC.

Řešení:

Zkontrolujte, zda na Dozorném PC nejsou programy, které blokují nebo zpomalují systém. Zakažte šetřič obrazovky na Dozorném PC. Pokud problém přetrvává, obraťte se na výrobce stroje.

16641 Řídicí firmware neodpovídá na příkazy

Příčina:

Došlo k výskytu chyby během inicializace systému. Konkrétně firmware neodpovídá správným způsobem. Problém může souviset s poškozením souboru firmwaru.

Řešení:

Zkuste vynulovat systém a případně znovu nainstalovat software Albatros. Pokud problém přetrvává, obraťte se na výrobce.

16642 TpaSock neodpovídá na příkazy

Příčina:

Došlo k výskytu chyby během inicializace systému. Konkrétně software pro komunikaci se vzdálenými moduly neodpovídá správně. Problém může souviset s poškozením souborů.

Řešení:

Zkuste vynulovat systém a případně znovu nainstalovat software Albatros. Pokud problém přetrvává, obraťte se na výrobce stroje.

16643 Operační systém neumožňuje používat RTX

Příčina:

Operační systém nainstalovaný na PC neumožňuje použití programu RTX, a proto neumožňuje správnou činnost verzí softwaru Albatros, které počítají s jeho přítomností.

Řešení:

Proveďte aktualizaci operačního systému PC. Přečtěte si minimální požadavky systému v návodu k instalaci softwaru Albatros (InstallationGuide.pdf).

16645 Chyba při odesílání kódu firmwaru

Příčina:

Došlo k výskytu chyby během inicializace systému. Konkrétně se nezdařila zkouška odeslání souboru firmwaru do jednoho z modulů.

Řešení:

Zkuste resetovat řízení. Pokud problém přetrvává, obraťte se na výrobce stroje.

16646 Nebylo možné znovu spustit kód firmwaru

Příčina:

Došlo k výskytu chyby během opětovné inicializace systému. Konkrétně se nezdařilo spuštění firmwaru po jeho předešlém zastavení.

Řešení:

Zkuste resetovat systém. Pokud problém přetrvává, obraťte se na výrobce stroje.

16897 Nebyl nainstalován RTX

Příčina:

Nainstalovaná verze softwaru Albatros vyžaduje přítomnost programu RTX, nainstalovaného v PC, ale tento nebyl nalezen.

Řešení:

Nainstalujte RTX nebo v případě, že je již nainstalováno, proveďte jeho opětovnou instalaci. Přečtěte si návod k instalaci softwaru Albatros (InstallationGuide.pdf).

16898 Uživatel nedisponuje právy správce

Příčina:

Albatros Uvedení do činnosti bylo provedeno uživatelem, který nedisponuje právy Správce PC. Práva Správce jsou potřebná pro správnou činnost softwaru Albatros.

Řešení:

Zavřete probíhající pracovní relaci a vstupte do systému jako „Správce“ nebo jako uživatel s právy správce.

16899 Rozměr RAM modulu je chybný

Příčina:

Velikost RAM zaznamenaná na vzdáleném modulu neodpovídá naplánované hodnotě. Obvykle tato chyba poukazuje na poškození hardwaru.

Řešení:

Pokud problém přetrvává, obraťte se na výrobce stroje.

16900 IP adresa modulu je chybná

Příčina:

Byl zaznamenan vzdálený modul, jehož adresa nepatří do podsítě dozorného PC. Albatros Proto nemůže správně komunikovat s modulem.

Řešení:

Zkontrolujte, zda jsou nastavení služby AlbdHCP a síťové karty PC správná. Přečtěte si návod k instalaci programu Albatros (InstallationGuide.pdf).

16901 Modul je již odpojen od jiného výrobního zařízení

Příčina:

Byl zaznamenan vzdálený modul, který je již připojen k jinému dozornému PC. Může to souviset s přítomností jiného PC, na kterém je prováděn software Albatros a které používá stejný modul. Dále to může souviset s nesprávnou činností komunikačního softwaru modulu.

Řešení:

Zkontrolujte, zda vzdálený modul není používán jiným dozorným PC. Provedte resetování modulu. Pokud problém přetrvává, obraťte se na výrobce stroje.

16902 Modul není nakonfigurován

Příčina:

Byl zaznamenan modul, který není nakonfigurován v „Konfiguraci Systému“ Albatros.

Řešení:

Provedte konfiguraci modulu.

16903 Nastavení firewallu brání komunikaci

Příčina:

Byla zaznamenaná přítomnost brány firewallu nainstalované na PC, která brání komunikaci mezi řízením Albatros a vzdálenými moduly.

Poznámka: Albatros je schopen identifikovat přítomnost brány firewall operačního systému Windows, ne však jiných bran firewallu včetně těch, které jsou zahrnuty v některých antivirových programech.

Řešení:

Změňte nastavení brány firewallu nebo ji zakažte.

16904 Síťová karta není k dispozici nebo je vyřazena z činnosti

Příčina:

Nebyla zaznamenaná síťová karta použitelná pro připojení vzdálených modulů.

Poznámka: Skutečnost, že dojde k zaznamenání karty, ještě nezaručuje, že je tato je správně konfigurována a připojena.

Řešení:

Zkontrolujte přítomnost a správnost konfigurace síťové karty. Pokud problém přetrvává, obraťte se na výrobce stroje.

16905 Chybí kód firmwaru řízení

Příčina:

Albatros Nebyl nalezen soubor firmwaru na pevném disku PC. Problém se může projevit po náhodném vymazání souborů firmwaru nebo po chybné aktualizaci.

Řešení:

Zkontrolujte, zda jsou přítomné soubory, které se nacházejí v instalačním adresáři FW softwaru Albatros, přítomné a zda patří správné verzi. Obraťte se na výrobce stroje.

16906 Verze RTX není kompatibilní s kódem firmwaru řízení

Příčina:

Byla zaznamenána verze programu RTX nekompatibilní s nainstalovaným firmwarem.

Řešení:

Nainstalujte správnou verzi RTX nebo proveďte aktualizaci firmwaru. Obraťte se na výrobce stroje.

16907 Verze operačního systému není kompatibilní s kódem firmwaru řízení

Příčina:

Verze operačního systému vzdáleného modulu není kompatibilní s nainstalovaným firmwarem.

Řešení:

Nainstalujte do vzdáleného modulu správnou verzi operačního systému nebo proveďte aktualizaci firmwaru. Obraťte se na výrobce stroje.

17153 DruhKarty: Chybí kód firmwaru vysílače GreenBUS

Příčina:

Nebyl nalezen soubor firmwaru v adresáři FW. Obvykle tento druh problému souvisí s náhodným vymazáním souboru nebo nekompletní či poškozenou instalací.

Řešení:

Proveďte opětovnou instalaci softwaru Albatros po provedení zálohování systému. Obraťte se na výrobce stroje.

17154 DruhKarty: Chybí část kódu firmwaru vysílače GreenBUS

Příčina:

Soubor obsahující kód firmwaru vysílače GreenBus se nachází v adresáři FW, ale je poškozený nebo neúplný.

Řešení:

Proveďte opětovnou instalaci softwaru Albatros po provedení zálohování systému. Obraťte se na výrobce stroje.

17155 DruhKarty: Chyba v odesílání spouštěcího kódu (Bootstrap) vysílače GreenBus

Příčina:

Došlo k výskytu chyby během inicializace systému. Konkrétně se nezdařila zkouška odeslání souboru firmwaru do jednoho z modulů.

Řešení:

Zkuste resetovat řízení. Pokud problém přetrvává, obraťte se na výrobce stroje.

17156 DruhKarty: Chyba v odesílání hlavní části kódu (Main) vysílače GreenBus**Příčina:**

Došlo k výskytu chyby během inicializace systému. Konkrétně se nezdařila zkouška odeslání souboru firmwaru do jednoho z modulů.

Řešení:

Zkuste resetovat řízení. Pokud problém přetrvává, obraťte se na výrobce stroje.

17157 DruhKarty: Chybí spouštěcí kód (Bootstrap)**Příčina:**

Nebyl nalezen soubor firmwaru v adresáři FW. Obvykle tento druh problému souvisí s náhodným vymazáním souboru nebo nekompletní či poškozenou instalací.

Řešení:

Provedte opětovnou instalaci softwaru Albatros po provedení zálohování systému. Obraťte se na výrobce stroje.

17158 DruhKarty: Chybí hlavní část kódu (Main)**Příčina:**

Nebyl nalezen soubor firmwaru v adresáři FW. Obvykle tento druh problému souvisí s náhodným vymazáním souboru nebo nekompletní či poškozenou instalací.

Řešení:

Provedte opětovnou instalaci softwaru Albatros po provedení zálohování systému. Obraťte se na výrobce stroje.

17159 DruhKarty: Chyba v odesílání spouštěcího kódu (Bootstrap)**Příčina:**

Došlo k výskytu chyby během inicializace systému. Konkrétně se nezdařila zkouška odeslání souboru firmwaru do jednoho z modulů.

Řešení:

Zkuste resetovat řízení. Pokud problém přetrvává, obraťte se na výrobce stroje.

17160 DruhKarty: Chyba v odesílání hlavní části kódu (Main)**Příčina:**

Došlo k výskytu chyby během inicializace systému. Konkrétně se nezdařila zkouška odeslání souboru firmwaru do jednoho z modulů.

Řešení:

Zkuste resetovat řízení. Pokud problém přetrvává, obraťte se na výrobce stroje.

17409 Nebylo možné odeslat pomocný spustitelný soubor**Příčina:**

Tato chyba se může vyskytnout během aktualizace firmwaru vzdáleného modulu. Může být způsobena okamžitou nesprávnou činností sítě, ale také poškozením firmwaru modulu. Chybové hlášení může zahrnovat chybový kód.

Řešení:

Zkuste vypnout a znovu zapnout vzdálený modul a poté zopakovat postup aktualizace. Pokud problém přetrvává, obraťte se na výrobce stroje.

17410 Nebylo možné provést pomocný spustitelný soubor**Příčina:**

Došlo k výskytu chyby během inicializace systému. Konkrétně nebylo možné provést pomocný program. Chybové hlášení uvádí také název pomocného programu a případně kód chyby.

Řešení:

Zkuste resetovat systém. Pokud problém přetrvává, obraťte se na výrobce stroje.

17667 NázevDLL: Nebylo možné znovu spustit kód firmwaru**Příčina:**

Došlo k výskytu chyby během inicializace systému. Konkrétně nebylo možné provést kód firmwaru. „NázevDLL“ odpovídá komponentu, který způsobil vznik chyby.

Řešení:

Zkuste resetovat systém. Pokud problém přetrvává, obraťte se na výrobce stroje.

17668 NázevDLL: Nebylo možné získat ukazatel do sdílené RAM**Příčina:**

Došlo k výskytu chyby během inicializace systému. Konkrétně nebylo možné otevřít kanál pro komunikaci s firmwarem. „NázevDLL“ odpovídá komponentu, který způsobil vznik chyby.

Řešení:

Zkuste resetovat systém. Pokud problém přetrvává, obraťte se na výrobce stroje.

17921 Nebylo možné odeslat NODETPA**Příčina:**

Tato chyba se může vyskytnout během aktualizace firmwaru vzdáleného modulu. Může být způsobena okamžitou nesprávnou činností sítě, ale také poškozením firmwaru modulu. Chybové hlášení může zahrnovat chybový kód.

Řešení:

Zkuste vypnout a znovu zapnout modul a poté zopakovat postup aktualizace. Pokud problém přetrvává, obraťte se na výrobce stroje.

17922 NODETPA nebyl znovu uveden do činnosti**Příčina:**

Tato chyba se může vyskytnout během aktualizace firmwaru vzdáleného modulu. Může být způsobena okamžitou nesprávnou činností, ale také poškozením firmwaru modulu. Chybové hlášení může zahrnovat chybový kód.

Řešení:

Zkuste vypnout a znovu zapnout modul a poté zopakovat postup aktualizace. Pokud problém přetrvává, obraťte se na výrobce stroje.

17923 NODETPA není spuštěn**Příčina:**

V síti byl zaznamenán vzdálený modul s nespouštěným komunikačním softwarem. Obvykle se jedná o příznak nesprávné činnosti komunikačního softwaru. Chybové hlášení může zahrnovat chybový kód.

Řešení:

Zkuste vypnout a znovu zapnout modul. Pokud problém přetrvává, obraťte se na výrobce stroje.

18177 NODETPA se pokusila o přístup na neplatnou adresu**Příčina:**

Komunikační software vzdáleného modulu způsobil vznik chyby. Chybové hlášení může zahrnovat chybový kód.

Řešení:

Zkuste vypnout a znovu zapnout modul. Pokud problém přetrvává, obraťte se na výrobce stroje.

6.3 Obecné signalizace

6.3.1 Albatros zahajuje provádění

Signalizuje spuštění systému Albatros a zobrazuje některé užitečné informace o verzi programu a prostředí provádění.

6.3.2 Albatros končí provádění

Signalizuje, že Albatros zanedlouho ukončí provádění.

6.3.3 Počítač vstupuje do pohotovostního režimu

Signalizuje, že počítač zanedlouho vstoupí do režimu pozastavení. Od tohoto okamžiku systém Albatros není dále schopen odpovídat na požadavky a signalizace, přicházející z cyklu GPL.

6.3.4 Počítač vychází z pohotovostního režimu

Signalizuje, že počítač právě ukončil režim pozastavení. Albatros obnoví provádění bez toho, aby byl restartován.

6.3.5 Vypnutí počítače

Signalizuje, že počítač bude zanedlouho vypnut, zatímco systém Albatros je ještě spuštěn.

6.3.6 Aktuální úroveň přístupu

Signalizuje změnu úrovně přístupu k funkcím systému Albatros, obvykle pro provádění údržby nebo změny cyklu nebo konfigurace.

6.3.7 Aktualizace softwaru modulů

Signalizuje, že bylo požádáno o aktualizaci softwaru a firmwaru, které se nacházejí ve vzdálených řízeních.

6.3.8 Odeslání konfigurace do modulů

Signalizuje, že bylo požádáno o aktualizaci konfigurace a cyklu, které se nacházejí ve vzdálených řízeních.

7 Konfigurace Systému

7.1 Úvod

V kapitole věnované složení systému již bylo možné vidět, jak je systém Albatros složen z jednoho nebo více modulů, které tvoří výrobní zařízení, a jak je každý z nich zorganizován do hierarchické struktury. Pro konfiguraci stroje z hlediska Albatros je třeba sledovat posloupnost operací, jejichž prostřednictvím kterých se provádí konfigurace jednotlivých logických úrovní a podléhajících hardwaru. Přibližné pořadí, které je třeba dodržovat při konfiguraci systému, je:

- [Konfigurace Modulu](#)
- [Definice Skupin a Podskupin](#)
- [Konfigurace Zařízení](#)
- [Konfigurace Systému](#)
- [Konfigurace Hardwaru](#)
- [Konfigurace Virtuální-Fyzické](#)

Konfigurace Modulu, Skupiny a Stroje definují logickou strukturu stroje, Konfigurace Systému, Hardwaru a Virtuální-Fyzické definují jeho strukturu na fyzické úrovni. Podrobně tyto body probereme v následujících odstavcích.

7.2 Konfigurace Zařízení

7.2.1 Úvod

V kapitole týkající se složení systému Albatros byly představeny druhy zařízení, které se mohou objevit v modulu. Níže jsou zařízení představena znovu, ale z hlediska své konfigurace.

Pro každý druh existuje maximální počet nakonfigurovatelných zařízení v souladu s níže uvedeným seznamem:

Druh zařízení	Max. počet
Analogový vstup	128
Analogový výstup	128
Digitální vstup	4096
Digitální výstup	4096
Port vstupu	512
Port výstupu	512
Jednotky pohybu v ose	240
Časovače	128
Počítadlo	128
Příznak bitu	1024
Příznak přepínače	256
Příznak portu	256

7.2.2 Všeobecné zařízení

Větší část zařízení vyžaduje stejné parametry konfigurace. Níže je uvedena konfigurace Digitálního vstupu, ale stejné úvahy se vztahují i na:

- Příznak bitu
- Příznak přepínače
- Analogové výstupy
- Port vstupů
- Port výstupů
- Porty příznaků
- Časovače
- Počítadla

Pro konfiguraci některého zařízení ve výše uvedeném seznamu je třeba uvést následující nastavení:

- **Název:** Název zařízení s maximální délkou 40 znaků.
- **Komentář:** Stručný popis zařízení, který může být přeložen do různých jazyků.
- **Přístupy při čtení:** Vymezuje minimální požadovanou úroveň přístupu, která umožňuje, aby bylo zařízení zobrazováno v oknech Diagnostiky a v Grafických zobrazeních.
- **Přístupy při zápisu:** Vymezuje minimální požadovanou úroveň přístupu potřebnou ke změně stavu zařízení.

- **Veřejné:** Vymezuje, zda stav zařízení může být čten a měněn kódem GPL, který však nepatří do skupiny, v níž je zařízení umístěno.

7.2.3 Digitální výstup

Digitální výstup má ve srovnání se standardními zařízeními jeden parametr navíc: Monostabilní

Pro konfiguraci digitálního výstupu musí být uvedena následující nastavení:

- **Název:** Název zařízení s maximální délkou 40 znaků.
- **Komentář:** Stručný popis zařízení, který může být přeložen do různých jazyků.
- **Monostabilní:** Je-li zvoleno, konfiguruje výstup jako monostabilní, to znamená, že když bude výstup nastaven na ZAPNUTÝ, po uplynutí 200 ms se automaticky vrátí do stavu VYPNUTÝ.
- **Přístupy kvůli čtení:** Vymezuje minimální požadovanou úroveň přístupu, která umožňuje, aby bylo zařízení zobrazováno v oknech Diagnostiky a v Grafických zobrazeních.
- **Přístupy kvůli zápisu:** Vymezuje minimální požadovanou úroveň přístupu potřebnou ke změně stavu zařízení.
- **Veřejné:** Vymezuje, zda stav zařízení může být čten a měněn kódem GPL, který však nepatří do skupiny, v níž je zařízení umístěno.

7.2.4 Analogový vstup

Analogový vstup má ve srovnání se standardními zařízeními o jeden parametr navíc: *Druh vstupního napětí*.

Pro konfiguraci analogového vstupu musí být uvedena následující nastavení:

- **Název:** Název zařízení s maximální délkou 40 znaků.
- **Komentář:** Stručný popis zařízení, který může být přeložen do různých jazyků.
- **Druh:** Umožňuje zvolit interval napětí čtených na vstupu.
- **Přístupy kvůli čtení:** Vymezuje minimální požadovanou úroveň přístupu, která umožňuje, aby bylo zařízení zobrazováno v oknech Diagnostiky a v Grafických zobrazeních.
- **Přístupy kvůli zápisu:** Vymezuje minimální požadovanou úroveň přístupu potřebnou ke změně stavu zařízení.
- **Veřejné:** Vymezuje, zda stav zařízení může být čten a měněn kódem GPL, který však nepatří do skupiny, v níž je zařízení umístěno.

7.2.5 Osa

Základní údaje

Je třeba uvést následující základní údaje:

- **Název:** Název zařízení s maximální délkou 40 znaků.
- **Popis:** Stručný popis zařízení, který může být přeložen do různých jazyků.
- **Rozlišení:** Rozlišení snímače impulzů, které závisí na charakteristikách snímače impulzů a vymezené měrné jednotce. Je třeba mít na paměti, že karty jednotek pohybu v osách Albatros počítají za impuls náběžné i sestupné hrany obou fází snímače impulzů (snímač impulzů s 2500 impulzy na jednu otáčku bude proto patrný jako snímač impulzů s 10000 impulzy na jednu otáčku).
- **Druh:** Druh jednotky pohybu v ose. K možným druhům patří **Analogový** (řízený analogově), **Krokový**, **Digitální**, **Počítání** (pouze čtení snímače impulzů), **Virtuální**.
- **Měrná jednotka:** Měrná jednotka používaná pro vyjádření cílových poloh jednotky pohybu v ose. Na ní závisí všechny odvozené veličiny, a proto se doporučuje nastavit ji před jakýmkoli dalším parametrem.
- **Inverze fází:** Umožňuje kompenzovat prostřednictvím softwaru případnou inverzi kabeláže fází snímače impulzů.
- **Inverze řídicího signálu:** Umožňuje obrátit řídicí signál rychlosti jednotky pohybu v ose. V případě použití tohoto parametru spolu s inverzí fází umožňuje provést inverzi směru jednotky pohybu v ose (za předpokladu správnosti provedení kabeláže).
- **Aktivace značky:** Je k dispozici pouze pro jednotky pohybu v ose počítání a slouží k automatickému vynulování polohy při zaznamenání značky nuly snímače impulzů.

Parametry pohybu

Parametry používané pro pohyb jednotky pohybu v ose z bodu do bodu.

- **Maximální rychlost:** Maximální rychlost jednotky pohybu v ose.
- **Zrychlení:** Doba trvání rampy zrychlení

- **Zpomalení:** Doba trvání rampy zpomalení
- **Minimální rychlost:** Rychlost, které motor dosáhne v jediném kroku, lze ji nastavit pouze pro jednotky krokového pohybu v ose.
- **Druh rampy:** Druh rampy zrychlení a zpomalení. Nenastavitelný pro krokové motory
- **Proporcionální:** Proporcionální koeficient regulátoru PID smyčky sledování polohy
- **Integrační:** Integrační koeficient regulátoru PID smyčky sledování polohy
- **Derivační:** Derivační koeficient regulátoru PID smyčky sledování polohy
- **Feed forward:** Procentuální podíl feed forward (dopředné řízení). Umožňuje odpovídající potlačení chyby smyčky úměrně s rychlostí.
- **Feed forward zrychlení:** Procentuální podíl feed forward zrychlení (dopředné řízení). Umožňuje potlačení zbytkové chyby smyčky (nepotlačené dopředným řízením) během fází zrychlení a zpomalení jednotky pohybu v ose.
- **Vzorky integrační složky** Slouží k nastavení počtu vzorků chyby sledování, použitých pro výpočet integrační složky. Platné hodnoty jsou v rozsahu od 1 do 200. Přednastavená hodnota je 50. Viz instrukce GPL [SETINTEGRIME](#).

Parametry interpolace

Parametry používané pro interpolovaný pohyb jednotky pohybu v ose.

Mají stejný význam jako parametry, které již byly uvedeny v nastaveních Pohybu, s výjimkou Minimální rychlosti, která se určuje pouze pro Jednotky krokového pohybu v osách při pohybech z bodu do bodu. Tyto se však používají pro interpolované pohyby.

POZN.: Hodnoty zrychlení a zpomalení nastavené v rámci parametrů interpolace nemohou být menší než odpovídající hodnoty v parametrech pohybu.

Ostatní parametry

- **Rychlost v manuálním režimu:** Vymezuje maximální rychlost konfigurace použitelnou při manuálně ovládaných pohybech. Nebude nikdy vyšší, než je maximální nastavená rychlost.
- **Dynamická chyba servořízení:** slouží k aktivaci nebo zrušení dynamické chyby servořízení. Přednastavenou hodnotou je zrušení dynamické chyby servořízení, a proto zůstává aktivována chyba servořízení s mezní hodnotou. Viz instrukce GPL [SETMAXERTYPE](#).
- **Řídicí rychlost a Chyba smyčky:** tyto dvě hodnoty slouží k výpočtu reálného lineárního vztahu dvou veličin ve stroji. Aby byly hodnoty zohledněny, musí být obě kladné a odlišné od nuly, a dále musí být aktivováno pole **Dynamická chyba servořízení**.
- **Čekání na zastavení jednotky pohybu v ose:** Slouží k aktivaci nebo zakázání funkce návratu po přejetí koncové polohy. Slouží k zavedení doby čekání s hodnotou 50 ms na konci každého pohybu.
- **Časový limit pohybu Jednotky pohybu v ose:** Platné hodnoty jsou v rozsahu od 0 do 1024. Viz instrukce GPL [ENABLESTARTCONTROL](#).
- **Mezní hodnota odpovídající chybnému připojení snímače impulsů:** Nastavené hodnoty jsou vyjádřeny v měrných jednotkách, ve kterých je vyjádřeno rozlišení jednotky pohybu v ose. Nastavené hodnoty musí být v rozsahu od 128/rozlišení jednotky pohybu v ose do 16384/rozlišení jednotky pohybu v ose. Přednastavená hodnota je vypočtena na základě počtu kroků, který se rovná 1024, tj. 1024/rozlišení jednotky pohybu v ose.
- **Kladný limit chyby servořízení:** Maximální hodnota chyby sledování pro regulační smyčku v kladném směru.
- **Záporný limit chyby servořízení:** Maximální hodnota chyby sledování pro regulační smyčku v záporném směru.
- **Kladný limit jednotky pohybu v ose:** Maximální hodnota dráhy jednotky pohybu v ose v kladném směru.
- **Záporný limit jednotky pohybu v ose:** Maximální hodnota dráhy jednotky pohybu v ose v záporném směru.
- **Okno kladného příchodu do cílové polohy:** Tolerance cílové polohy v kladném směru.
- **Okno záporného příchodu do cílové polohy:** Tolerance cílové polohy v záporném směru.

Referenční parametry

- **Řídicí signál:** Hodnota napětí řídicího signálu, která odpovídá maximální rychlosti.
- **Automatické doladění:** Slouží k aktivaci nebo zakázání výpočtu opětovného získání automatického posunu. Za běžných podmínek je aktivováno.
- **Počáteční posun:** Hodnota, na kterou je třeba nastavit počáteční vztažný posun. Hodnota se musí nacházet v rozmezí od -10 do 10. Přednastavená hodnota je 0.
- **Frekvence Notchova filtru:** Hodnota frekvence, která má být filtrována. Hodnota se musí nacházet v rozmezí od 0 do 500.

- **Minimální napětí:** Slouží k nastavení parametrů minimálního napětí pro uvedenou jednotku pohybu v ose. Záporná hodnota se musí nacházet v rozmezí od -10 do 0, zatímco kladná hodnota v rozmezí od 0 do +10. Viz instrukce GPL [SETDEADBAND](#).
- **Mezní hodnota:** Slouží k nastavení mezních hodnot. Hodnoty musí být vždy menší nebo se musí rovnat příslušným hodnotám minimálního napětí, a proto se záporná mezní hodnota musí nacházet v rozsahu od 0 do hodnoty minimálního záporného napětí. Maximální mezní hodnota se musí nacházet v rozmezí od 0 do hodnoty minimálního kladného napětí.

Úrovně přístupu

- **Přístupy kvůli čtení:** Vymezuje minimální požadovanou úroveň přístupu, která umožňuje, aby byla jednotka pohybu v ose zobrazována v oknech Diagnostiky a v Grafických zobrazeních.
- **Přístupy kvůli zápisu:** Vymezuje minimální požadovanou úroveň přístupu potřebnou ke změně stavu jednotky pohybu v ose.
- **Veřejné:** Vymezuje, zda stav osy může být čten a měněn kódem GPL, který však nepatří do skupiny, v níž je osa umístěna.

Provázání Jednotek pohybu v osách

Parametry provázání jednotek pohybu v osách. Jedná se o koeficienty řídicího zařízení PID, které kompenzuje rozdíly chyby smyčky mezi jednotkou pohybu v ose typu master a jednotkami pohybu v osách typu slave.

- **Proporcionální:** proporcionální koeficient
- **Integrační:** integrační koeficient
- **Derivační:** derivační koeficient

Korektory linearity

Nastavení korektorů linearity jednotky pohybu v ose. Korektory umožňují kompenzovat chyby polohování jednotky pohybu v ose způsobené nepřesností mechaniky samotné jednotky pohybu v ose (samokorektory) a chyby způsobené účinkem ostatních jednotek pohybu v osách stroje (křížové korektory), které obvykle souvisejí s ohybem konstrukce. Korektory nejsou aktivovány automaticky, ale aktivují se v okně pro změnu hodnot korekce (tlačítko **[Změna...]**) a aktivují se kódem GPL prostřednictvím instrukce [ENABLECORRECTION](#).

- **Interval korekce:** Umožňuje nastavit vzdálenost mezi dvěma po sobě následujícími korekcemi. Počet měření bude výsledkem délky jednotky pohybu v ose vydělené délkou intervalu korekce.
- **Název souboru korektorů:** Umožňuje nastavit název souboru, do kterého jsou hodnoty korekce ukládány. Jedná se o soubor ASCII, ve kterém jsou hodnoty vzájemně odděleny znakem „;“. To umožňuje jeho změnu prostřednictvím běžného textového editoru. Příponu souboru není třeba uvádět, protože bude automaticky provedeno přidání přípony „.csv“ (comma separated values).
- **Údaje korekce:** Umožňuje uvést seznam jednotek pohybu v osách, pro které aktuální jednotka pohybu v ose vytváří korekci. Aktuální jednotka pohybu v ose je vždy zahrnuta do seznamu, tj. samokorektor je vždy přítomen. Je možné upřesnit až do 5 dalších jednotek pohybu v ose. Za účelem přidání jednotky pohybu v ose ji zvolte v seznamu v levé části a stiskněte tlačítko **[>>Přidat]**. Za účelem odstranění jednotky pohybu v ose proveďte její volbu v seznamu v pravé části a stiskněte tlačítko **[>>Odstranit]**. Za účelem uvedení hodnot korekce zvolte jednotku pohybu v ose ze seznamu v pravé části a stiskněte tlačítko **[Změna...]**; bude otevřeno okno s tabulkou, do které je třeba zadat hodnoty korekce.

POZNÁMKA: Existuje maximální limit **235** korektorů linearity, které jsou spravovány systémem pro každou jednotku pohybu v ose. Z toho vyplývá, že na základě délky jednotky pohybu v ose interval měření bude muset mít délku větší nebo rovnou dvěstětřicetpětinové části délky jednotky pohybu v ose. Například pro jednotku pohybu v ose o délce 2500 mm bude třeba nastavit interval korekce větší nebo rovný 10,63 mm. Kromě toho existuje limit maximální hodnoty jednotlivé korekce, která bude muset být menší než 1024 kroků snímače impulzů; například pro jednotku pohybu v ose s rozlišením 256 kroků/mm bude maximální přípustná korekce ± 4 mm.

7.3 Logická konfigurace

7.3.1 Konfigurace výrobního zařízení

Pro zadefinování nového stroje nebo pro změnu již existujícího stroje je třeba provést přístup na stranu Konfigurace modulu. Konfigurace modulu a konfigurace modulů, které tvoří výrobní zařízení.

Otevření prostředí Konfigurace je možné nacházíme-li se na úrovni přístupu výrobce nebo na vyšší úrovni.

Přístup ke Konfiguraci

Zvolte v menu **Soubor** položku **Otevřít Konfiguraci**

Když se na výrobním zařízení nenacházejí již nakonfigurované moduly, bude automaticky otevřena Konfigurace modulu; v opačném případě bude otevřena Konfigurace stroje. V tomto případě lze přejít na Konfiguraci modulu níže uvedeným způsobem:

Zvolte v menu **Úpravy** položku **Konfigurace modulu**

Pro přidání modulu do výrobního zařízení stačí stisknout tlačítko **[Nový]**. Tlačítko **[Úpravy]** umožňuje změnit údaje již existujícího modulu, tlačítko **[Odstranit]** umožňuje vymazání modulu a tlačítko **[Zavřít]** umožňuje zavřít zobrazování konfigurace výrobního zařízení.

K údajům, které identifikují stroj a které je třeba uvést, patří:

- Číslo modulu: Celé pořadové číslo, které, nebude-li uvedeno, bude přiřazeno systémem
- krátký popis.

Některé údaje, které se týkají hardwaru.

- **Frekvence řízení jednotek pohybu v osách:** určuje frekvenci, se kterou je pravidelně prováděna výměna dat mezi číslicovým řízením a zařízeními, která jsou k němu připojena prostřednictvím polní sběrnice.
- **Číslo kanálu interpolace:** určuje maximální počet kanálů interpolace (tj. skupin jednotek pohybu v osách, které provádějí pohyb interpolovaného typu), které mohou být řízeny současně.
- **Procentuální vyjádření použití CPU:** určuje procentuální podíl časového úseku, který se vztahuje na období řízení jednotek pohybu v osách (tj. opačně ve srovnání s „frekvencí řízení jednotek pohybu v osách“), který je vyhrazen pro provádění firmwaru.

Stejně okno lze otevřít z větve modulu pro konfiguraci skupin, z větve modulu pro konfiguraci stroje a z větve modulu pro konfiguraci hardwaru.

7.3.2 Konfigurace skupin

Při prvním návrhu stroje je třeba definovat každý jeho komponent a zapsat všechny řídicí cykly. V mnoha případech je však návrh proveden z již vytvořeného stroje, který je změněn na základě parametrů nového stroje.

Vytvoření Skupiny

Pro vytvoření nové skupiny je třeba přistoupit k zobrazení konfigurace skupin. První větví stromu je modul, ze kterého jsou odvozeny všechny skupiny, podskupiny a zařízení. Stisknutím tlačítka **[ENTER]** nebo tlačítka **[Úpravy]** dojde k otevření dialogového okna pro změnu dat modulu.

Zvolte v menu **Úpravy** položku **Skupiny**

Z tohoto místa lze vytvořit nové skupiny, měnit nebo vymazat ty existující a zkopírovat skupinu.

Seznam příkazů pro vytvoření, změnu, vymazání, kopírování a vkládání skupin, podskupin a zařízení

Ovládací příkaz	Úkon
Slouží k vytvoření nové skupiny, podskupiny, zařízení	[Ctrl+Enter], Tlačítko [Nový], Úpravy->Nový, kontextové menu
Slouží ke změně skupiny, podskupiny, zařízení	[Enter], Tlačítko [Změna], Úpravy->Úpravy..., kontextové menu
Slouží k odstranění skupiny, podskupiny, zařízení	[Del], Tlačítko [Odstranit], Úpravy->Odstranit, kontextové menu

Aktivuje či deaktivuje použití skupiny ve stroji	Kontextové menu, Tlačítko [Aktivovat]
Slouží ke kopírování skupiny, podskupiny, zařízení	[Ctrl+C], Tlačítko [Kopírovat], Úpravy->Kopírovat, kontextové menu
Slouží k vložení skupiny, podskupiny, zařízení	[Ctrl+V], Tlačítko [Vložit], Úpravy->Vložit, kontextové menu

Při vytváření nové skupiny se zobrazí okno, ve kterém je třeba nastavit:

- Název skupiny
- Komentář (lze jej přeložit do jazyků, které je možné spravovat z Albatrosu)

Dále je možné označit skupinu jako **Meziskupinu**. Nejméně jedna skupina musí být nastavena jako meziskupina, protože tuto volbu používá Albatros pro identifikaci „hlavní“ skupiny stroje. To je skupina, jejíž hlavní funkce (se stejným názvem jako skupina) je automaticky spuštěna při uvedení do činnosti. Tento mechanismus slouží k inicializaci stroje a tím k ověření celkové správné činnosti před odevzdáním řízení obsluze.

Při deaktivaci skupiny, jejíž zařízení je připojeno k fyzickým zařízením, se zobrazí dotaz, zda je požadováno zrušení virtuálního-fyzického připojení. Pokud je zvoleno zachování připojení, piny fyzických zařízení, k nimž jsou zařízení připojena, budou v grafickém znázornění virtuálního-fyzického připojení zobrazeny šedou barvou.

Přidání podskupiny do skupiny

Pro vytvoření podskupiny skupiny je třeba, abyste se nacházeli ve skupině.

Když podskupině nemá být přiřazen specifický název, je možné zaškrtnout políčko *Seznam zařízení* a stisknout tlačítko **[OK]**. Název podskupiny bude přiřazen automaticky.

Nyní je možné vložit jednotlivá zařízení dovnitř podskupiny. Mechanismus je podobný tomu, který se používá pro vytváření podskupin. Zobrazí se okno se seznamem dostupných zařízení.

Po zvolení zařízení, které je předmětem zájmu, je třeba potvrdit volbu stisknutím tlačítka **[OK]**.

Zobrazí se okno, které umožňuje nastavit název, komentář a další údaje, které se mění v závislosti na zvoleném zařízení. Detailní popis typů zařízení a jejich nastavení je uveden v kapitole [Konfigurace Zařízení](#).

Kopírování zařízení

Funkce kopírování zařízení umožňuje zkopírovat zařízení. Především je třeba zvolit zařízení a poté tlačítko **[Kopírovat]**. Pro vložení zařízení do seznamu je třeba, zvolit větev, do které má být zařízení vloženo, a aktivovat příkaz **[Vložit]**. V dialogovém okně musí být zadán nový název zařízení.

Kopírování podskupiny

Funkce kopírování podskupiny umožňuje kopírovat podskupinu zahrnující všechna zařízení, která se v ní nacházejí. Pro vložení podskupiny je třeba zvolit větev, do které má být podskupina vložena, a aktivovat příkaz **[Vložit]**. V dialogovém okně musí být zadán nový název podskupiny.

Kopírování skupiny

Funkce kopírování skupiny umožňuje kopírovat skupinu zahrnující všechny obsažené podskupiny a zařízení. Kromě toho bude zkopírováno případné grafické zobrazení skupiny, které je k ní přiřazeno (grafické zobrazení, jehož název se shoduje s názvem skupiny).

Umožňuje rychle vytvořit skupiny, které mají strukturu podobnou struktuře již existující skupiny, aniž by bylo třeba znovu vytvořit jedno po druhém všechna zařízení. Pro zkopírování skupiny je třeba zvolit skupinu, která má být zkopírována, zvolit tlačítko **[Kopírovat]** a vložit do dialogového okna název nové skupiny.

Kopírování zařízení, podskupin a skupin může být provedeno také mezi odlišnými moduly.

Volba skupin náležejících ke stroji

Po vytvoření archivu skupin je nutno deaktivovat či aktivovat skutečně přítomné skupiny.

Všechny skupiny jsou ve stroji přítomné, pokud nedojde k jejich deaktivaci pomocí tlačítka **[Deaktivovat]** či zvolením téhož příkazu v kontextovém menu. Pokud došlo k deaktivaci určité skupiny, zobrazí se text **Nepřítomná** vedle názvu skupiny.

Pro zobrazení pouze skupin přítomných ve stroji zvolte v menu **Změna** položku **Stroj**.

Pro vložení nové skupiny je třeba stisknout tlačítko **[Vložit]**. Zobrazí se okno obsahující seznam skupin přítomných v archivu Skupiny, které ještě nebyly vloženy do stroje.

Nyní je třeba zvolit vybranou skupinu a **přetáhnout ji myší** do okna Konfigurace stroje nebo zvolit tlačítko **[Vložit]**.

Dále je možné odstranit pomocí tlačítka **[Odstranit]** existující skupinu nebo provést vyhledání názvu skupiny nebo zařízení uvnitř stromu sestavy stroje.

Ve stroji se musí nacházet jediná meziskupina.


7.4 Fyzická konfigurace

7.4.1 Konfigurace systému

Konfigurace systému umožňuje přiřadit fyzické zdroje (řídící jednotky) modulům zadaným v logické konfiguraci. Její provedení se může uskutečnit v dialogovém okně Konfigurace systému. Bude představen seznam **modulů** výrobního zařízení a každému z nich lze přiřadit jeden **Síťový uzel**.

- **Místní uzel:** „Místní“ systémy, v nichž je HW, který má implementované řízení, namontován přímo v PC, které představuje uživatelské rozhraní systému
- **Název síťového uzlu:** „Vzdálené“ systémy, v nichž je HW, který má implementované řízení, připojen k PC prostřednictvím sítě
- **Nenakonfigurován:** Bez jakékoli konfigurace. Jedná se o výchozí přednastavený stav. V případě ponechání této volby bude později možné přiřadit v dialogovém okně **Připojení síťových modulů** vzdálený modul.

Je možné nakonfigurovat až 16 modulů, ale pouze jeden může být nakonfigurován jako místní.

Za účelem přiřazení modulu zvolte tlačítko **[Změna]** nebo proveďte dvojité kliknutí myší v místě síťového modulu, který má být změněn. Při otevření roletového menu bude zobrazen seznam dostupných vzdálených modulů a volba použití Místního modulu nebo volba nastavení modulu jako nenakonfigurovaného. Pro potvrzení provedené volby je třeba zvolit tlačítko .

POZN.: Činnost Albatros ve stroji je chráněna přítomností hardwarového USB klíče, nakonfigurovaného firmou TPA.

7.4.2 Konfigurace Hardwaru

V konfiguraci hardwaru se definují karty a uzly, které tvoří systém. Karta, která zabírá první polohu v seznamu, se nazývá karta Master.

Ke konfigurovatelným druhům karet patří:

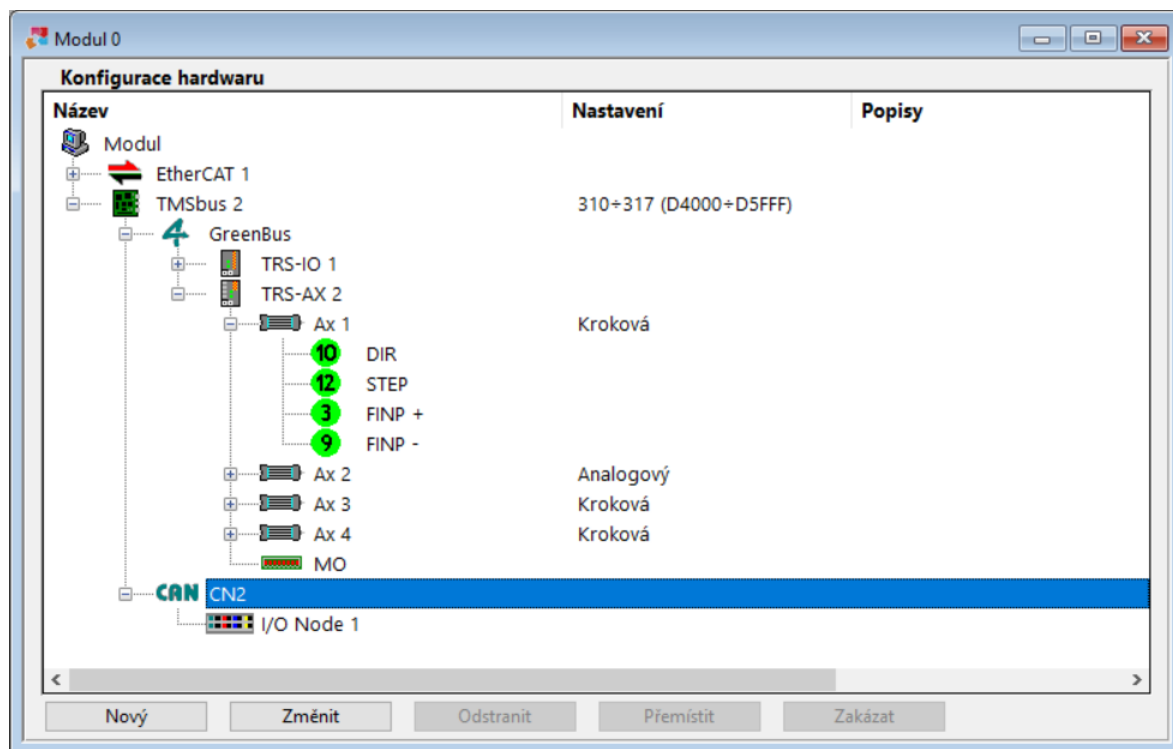
- TMSbus max. dvě
- TMSbus+ max. čtyři
- TMSCombo+ max. čtyři
- DualMech max. čtyři
- DualMech Mono max. čtyři
- TMSScan max. dvě
- TMSScan+ max. čtyři
- AlbMech max. dvě
- EtherCAT jedna

Popis okna pro konfiguraci hardwaru

Okno pro konfiguraci hardwaru se otevře volbou položky **Úpravy->Hardware** v příslušném menu.

Pro vložení karty nebo vzdáleného modulu VST./VÝST. nebo uzlu CAN nebo EtherCAT je třeba stisknout tlačítko **[Nový]**. Dojde k otevření okna, které umožňuje volbu karty nebo vzdáleného modulu VST./VÝST. a pro sběrnice CAN a GreenBUS a polohy, do které má být vložena.

Obecně platí, že pro každý modul lze nakonfigurovat maximálně 4 karty a v závislosti na typu karty a sběrnice variabilní počet modulů vzdálených VST./VÝST.



Konfigurace Hardwaru

V sloupci **Nastavení** se přiřazují informace související s kartou nebo s uzlem. Příkazem **[Přemístit]** je možné přemístit kartu, uzel nebo rozšíření modulu TRS-IO nebo TRS-CAT z jedné polohy stromu do druhé. Rozšíření mohou být přemístěna pouze v rámci stejného uzlu. Tato operace zachovává propojení přítomná v konfiguraci [Virtuální-Fyzické](#).

Dále je možné zrušit činnost uzlu nebo rozšíření modulu TRS-IO nebo TRS-CAT. Zrušení činnosti způsobí, že zůstanou zachována spojení v konfiguraci Virtuální-Fyzické. Když uzel přináleží sběrnici GreenBUS, samotný uzel i zařízení, která jsou k němu připojena, jsou úplně ignorována systémem. Nevznikne proto žádná chyba, pokud modul nebude zaznamenán v rámci inicializace, a žádná chyba nevznikne ani v případě, že bude provedena instrukce GPL na zařízení přiřazeném k modulu. Když uzel nebo rozšíření přináleží síti EtherCAT nebo když rozšíření přináleží sběrnici GreenBUS, nesmí být přítomen/no v síti. Když se v GPL provádí přístup k nepřipojenému zařízení, dojde k výskytu chyby systému. *Je třeba použití této funkce věnovat mimořádnou pozornost*

Pro zrušení činnosti uzlu nebo rozšíření použijte příkaz **[Deaktivovat]**, zatímco pro opětovnou aktivaci uzlu nebo rozšíření použijte příkaz **[Aktivovat]**.

Přednastavené konfigurace

K dispozici je více přednastavených konfigurací. Z nabídky **Úpravy->Změna typu řízení** nebo z kontextuální nabídky na větvi Modul je možné zvolit požadovanou konfiguraci. Když je konfigurace nová, strom bude zaplněn definovanými kartami a uzly; v opačném případě budou provedeny kontroly pro ověření kompatibility přítomného hardwaru se zvoleným typem modulu. Vše, co není kompatibilní, bude odstraněno.

Konfigurace uzlu na sběrnici TPA

Typy vzdálených I / O modulů konfigurovatelných na GreenBus (v3.0) jsou:

- Albre8 8 digitálních vstupů a 8 digitálních výstupů
- Albre16 16 kanálů konfigurovatelných jako digitální vstup nebo digitální výstup
- Albre24 24 digitálních vstupů a 24 digitálních výstupů
- Albre48 48 digitálních vstupů a 48 digitálních výstupů
- Albrem 10 portů na vstupu a 10 portů na výstupu
- AlbSTEP 8 digitálních vstupů a 6 digitálních výstupů, jeden krokový motor
- AlbEV 20 nebo 24 elektrických ventilů (25kolíkový konektor typu D-sub)
- Albre4 4 analogové vstupy a 4 analogové výstupy

Ke konfigurovatelným druhům vzdáleného modulu na GreenBus (v4.0) patří:

- TRS-AX 4 analogové nebo krokové jednotky pohybu v ose
- TRS-EV-24 24 elektrických ventilů (25kolíkový konektor typu D-sub)
- TRS-16 16 kanálů konfigurovatelných jako digitální vstup nebo digitální výstup
- TRS-IO 16 kanálů, konfigurovatelných prostřednictvím softwaru jako digitální vstup nebo digitální výstup, který lze rozšířit moduly typu TRS-IO-E a moduly TRS-AN-E až do celkového max. počtu 5 a moduly TRS-AC-E
- TRS-IO-E 16 kanálů konfigurovatelných prostřednictvím softwaru jako digitální vstup nebo digitální výstup, použitelný pouze jako rozšíření modulu TRS-IO
- TRS-AN-E 1 analogový vstup a 1 analogový výstup použitelný pouze jako rozšíření modulu TRS-IO
- TRS-REM všeobecný vzdálený pro realizaci speciálních modulů. Lze je připojit až do 12 portů na vstupu a 12 portů na výstupu.
- TRS-AC-E 1 jednotka počítání pohybu v ose a 2 digitální vstupy konfigurovatelné jako značka nuly a rychlý vstup („fast input“). Níže uvedená tabulka popisuje maximální počet TRS-AC-E, konfigurovatelných v TRS-IO.
Když celkově existují 3 rozšíření TRS-IO-E a TRS-AN-E, může být konfigurováno jediné rozšíření TRS-AC-E. Když je konfigurováno jediné rozšíření (TRS-IO-E nebo TRS-AN-E), je možné mít až do 2 rozšíření TRS-AC-E.

Vzdálené moduly TRS-AX, TRS-IO, TRS-REM a TRS-16 lze připojit výhradně ke kartám TMSbus, TMSbus+ a TMSCombo+.

Ke každé kartě TMSbus a TMSbus+ lze připojit maximálně 4 vzdálené moduly TRS-AX.

Ke konfigurovatelným druhům vzdáleného modulu od TPA na sběrnici EtherCAT patří:

- TRS-CAT 16 kanálů, konfigurovatelných prostřednictvím softwaru jako digitální vstup nebo digitální výstup, rozšiřitelný moduly typu TRS-IO-E, moduly TRS-AN-E a moduly TRS-AC-E
- STAR-CAT transformuje topologii sítě lineární EtherCAT na hvězdu s použitím jednoho kanálu na vstupu a až do 3 různých kanálů na výstupu

Níže uvedená tabulka popisuje maximální počet konfigurovatelných rozšíření v TRS-CAT

Počet rozšíření TRS-IO-E a TRS-AN-E	Počet rozšíření TRS-AC-E
7	0
5	1
3	2
1	3

Pro vzdálené moduly typu TRS-AX platí, že zvýšením počtu vložených TRS-AX se zmenší počet TRS-16 a TRS-IO, které lze použít.

Pro výpočet maximálního počtu vzdálených modulů typu TRS-16 a TRS-IO, které lze vložit do systému, je třeba aplikovat níže uvedený vzorec: počet dalších vzdálených modulů = $32 - (\text{počet TRS-AX} * 4)$. Když jsou například ke kartě TMSbus připojené 3 TRS-AX, aplikací uvedeného vzorce dostaneme: Počet dalších vzdálených modulů = $32 - (3*4)$, do systému je tedy možné vložit maximálně 20 vzdálených modulů typu TRS-16 a/nebo TRS-IO. Volba polohy vzdáleného modulu musí být provedena na základě adresy nastavené prostřednictvím prepínačů na vzdáleném modulu. Vycházejte přitom z dokumentace k hardwaru konkrétního vzdáleného modulu.

V případě volby vzdáleného modulu TRS-AX může být zapotřebí nastavit řízený typ jednotek pohybu v osách.

V níže uvedeném schématu jsou popsány druhy jednotek pohybu v osách, které lze přiřadit k jednotlivým hardwarům.

- karta AlbMech digitální jednotky pohybu v osách
- karta DualMech digitální jednotky pohybu v osách
- karta DualMech Mono digitální jednotky pohybu v osách
- vzdálený modul TRS-AX analogové jednotky pohybu v osách (když je specifický konektor analogového typu), jednotky pohybu v osách počítání (když je specifický konektor analogového typu), krokové jednotky pohybu v osách (když je specifický konektor krokového typu)
- vzdálený modul AlbStep krokové jednotky pohybu v osách
- rozšíření TRS-AC-E jednotky počítání pohybu v osách

Na kartách MECHATROLINK-II může být každá jednotka pohybu v ose spravována formou řízení polohy nebo řízení rychlosti (přednastavená hodnota). Typ řízení pro každou jednotku pohybu v ose může být změněn v okně

Konfigurace hardwaru, ve sloupci **Nastavení**. Počet konfigurovatelných jednotek pohybu v osách mění na základě hodnoty nastavené frekvence řízení jednotek pohybu v osách:

Karta	Frekvence Řízení Jednotek pohybu v osách (Hz)	Maximální počet servopohonů
AlbMech	1000	8
AlbMech	<=500	16
DualMech Mono	1000	8
DualMech Mono	500	20
DualMech Mono	250	30
DualMech	1000	16
DualMech	500	40
DualMech	250	60

Konfigurace uzlu sběrnice CAN

Kontrolní karta sběrnice

Albatros může řídit zařízení na sběrnici pole CAN pomocí karet Tpa vybavených konektorem pro sběrnici CAN. Sběrnice CAN je k dispozici na kartách **TMSbus, TMSbus+, TMSCan+ a TMSCan**.

Konfigurace základních dat a služeb

Data sběrnice CAN se definují v hardwarové konfiguraci. Vyberte sběrnici CAN, u které chcete definovat parametry, a klikněte na tlačítko **[Změna]**.

Základní údaje jsou následující:

- **Doba vzorkování (TIME)**: doba vzorkování v ms. Nesmí mít vyšší hodnotu než 60000 (60 sekund). Přednastavená hodnota je 2. Na sběrnici S-CAN je akceptována pouze hodnota 2.
- **Doba pro synchronní komunikaci PDO (TIMEPDO)**: doba vyjádřená v ms. Informuje o době věnované synchronní komunikaci PDO. Nastavená hodnota nesmí být vyšší, než je hodnota atributu TIME (Nejedná se o povinnou hodnotu).
- **Čekací doba (TIMEAFTERRESET)**: doba vyjádřená v ms. Označuje dobu čekání během počáteční fáze následující po vynulování softwaru uzlů v síti. Nesmí mít vyšší hodnotu než 60000 (60 sekund)
- **Počet cyklů CAN bez odpovědi (LIFETIMEFACTOR)**: je počet cyklů CAN bez odpovědi na volání Node Guarding dříve, než bude vygenerována chyba odpojeného uzlu. Hodnota nemůže být vyšší než 100 ani nižší než 1. (Přednastavená hodnota je 3).
- **Baud rate**: číslo rychlosti komunikace CAN v kilobitech za sekundu (může mít hodnotu 1000, 500, 250, 125, 100)

Služby mohou být aktivovány či deaktivovány zvolením či zrušením možnosti vztahujícího se k dané službě. Poli **Extra** mohou být přiděleny hodnoty mající význam definovaný výrobcem stroje. U této hodnoty neprobíhá žádná kontrola. Přednastavená hodnota je 16.

Uzel CAN

Vložení nového uzlu

Uzel se vloží zvolením větve CAN stromového zobrazení a kliknutím na tlačítko **[Nový]**. Typ uzlu se odvodí od typu sběrnice. Pokud je sběrnice typu CAN, uzel bude I/O Node, pokud je sběrnice S-CAN, typ vložené sběrnice bude typu Servo. V dialogovém okně pro vložení uzlu je nutno zvolit:

- **Poloha**: je číslo uzlu (počínaje 1).
- **AutoOp**: pokud je zvolen, zařízení připustí automatický přechod do Provozního stavu po opětovném připojení.

Konfigurace uzlu

Pro každý uzel je možno definovat PDO vysílání a PDO příjmu. Pokud je sběrnice CanOpen, je možno definovat maximálně 8 PDO pro karty TMSBus a TMSBus+, maximálně 4 pro karty TMSCan a TMSCan+. Ve stromovém zobrazení se zvolí daný uzel a poté se klikne na tlačítko **[Nový]**. Data k nastavení v dialogovém okně jsou následující:

- **Typ PDO:** zvolte **Vysílání** nebo **Příjem**, pokud chcete definovat TPDO či RPDO.
- **Rozměr:** rozměr PDO příjmu či vysílání.
- **COB-ID:** hodnota, kterou lze definovat pouze u karet TMSBus a TMSBus+. Hodnota je zobrazována a ukládána do paměti v desítkovém formátu. Pro zobrazení hodnoty v šestnáctkovém formátu je třeba zaškrtnout zaškrtačací políčko **Šestnáctkový**, které se nachází vedle samotné hodnoty.
- **Asynchronní:** pokud je funkce aktivována, je možné nakonfigurovat asynchronní PDO, tj. takové, které nejsou aktualizovány při každém cyklu. Tuto volbu lze použít pouze s kartami TMSCan a TMSCan+. Přijímání asynchronních PDO je třeba provést v kódu GPL pomocí instrukce [RECEIVEPDO](#)

Vlastnosti řízení EtherCAT v softwaru Albatros

Způsobem komunikace je vždy DC-Synchronous. První uzel sítě poskytuje takt, a proto je zcela nezbytné se ujistit, že tento uzel dodává přesný a stabilní takt v souladu s příkladem z TRS-CAT. Není možné používat jiné režimy, jako je například Free-Run.

K spravovaným protokolům patří: CoE (protokol pro aplikaci CAN na EtherCAT) a EoE (Ethernet na EtherCAT). Uvnitř CoE jsou spravovány profily zařízení („device profiles“) DS401 a DS402 s přednastaveným provozním režimem řízení jednotky pohybu v ose *cyklický synchronní režim řízení rychlosti („cyclic synchronous velocity mode“)*.

Maximální počet uzlů v EtherCAT je 200.

Výchozí předpoklady

Každému fyzickému zařízení sítě EtherCAT je přiřazen soubor ESI (EtherCAT Slave Information), který popisuje vlastnosti a funkce, které zařízení nabízí. Soubor je ve formátu XML. Pro každé zařízení musí existovat pouze jeden soubor ESI. Soubory ESI lze obvykle stáhnout z internetové stránky výrobce. Albatros hledá tyto soubory ve složce zadané v Tpa.ini v sekci [tpa] pod položkou DirESIFiles. Přednastavenou je podsložka „\EtherCAT“ v SYSTEM.

Ze souborů ESI Albatros získává informace o zařízení na základě analýzy všech prvků „/Devices/Device/Type“. Každé zařízení je identifikováno prostřednictvím Vendor ID, prostřednictvím Product ID a prostřednictvím Revision Number.

Ze souborů ESI se dále získávají informace o rozšířeních (nazvaných také moduly) zařízení. Albatros vyhledává informace, které se týkají druhů rozšíření, na základě vyhledání prvků „ESIModules/Module“ v souboru ESI.

Konfigurace hardwaru sítě EtherCAT

Hardware je konfigurován popisem karet master a pro každou kartu seznamem fyzických zařízení, připojených na sběrnici k této kartě. Fyzická zařízení se nazývají také „uzly“ polní sběrnice. Pro EtherCAT není karta master specifickou kartou řízení sběrnice, ale používá se síťové připojení modulu.

Pokud jde o lokální moduly síťového připojení, musí patřit k těm, které spravuje RTX, zatímco pro vzdálené moduly se používá specifické síťové připojení modulu z těch, která spravuje Windows CE 6.0. Pro každý místní nebo vzdálený modul je možné konfigurovat pouze jeden master.

Kartu master je nutno vložit do konfigurace hardwaru zvolením větve Modul a stisknutím tlačítka **[Nový]**.

Uzel EtherCAT

Pro vložení uzlu zvolte větev karty stromového zobrazení a klikněte na tlačítko **[Nový]**. Indexy uzlů musí jít po sobě, proto ke každému příkazu **[Nový]** je přiřazen uzel v řadě za ostatními. Pro každý uzel je nutno definovat následující údaje:

- **Název zařízení:** vyberte název zařízení ze seznamu. Zobrazí se pouze zařízení s nejnovější revizí; pokud chcete zobrazit všechny revize, je nutné zvolit položku **Zobrazení revizí**.
- **Režim jednotky pohybu v ose:** zvolte provozní režim určený pro uzly typu pohon, v závislosti na specifikaci v předpisu DS402, ve vztahu k objektu 6060₁₆ „mode of operation“ (režim činnosti). Výběr je mezi **Cyklicky synchronizovaná poloha** a **Cyklicky synchronizovaná rychlost**. Ten poslední je výchozí.

- **Nucené nastavení jako uzel &VST./VÝST.:** je-li aktivován, nuceně nastaví číslíkové řízení tak, aby považovalo pohon za uzel VST./VÝST. Tento atribut se aplikuje pouze na uzly, které podporují DS402 (servopohonu).

Režim jednotky pohybu v ose je jediným datem, který lze změnit po vložení uzlu.

Uzel sítě EtherCAT může být odstraněn, přemístěn, zrušen nebo kopírován. Příkazy jsou dostupné v kontextuální nabídce každé větve, v hlavní nabídce systému Albatros a v některých tlačítkách, které se nacházejí v okně pro konfiguraci hardwaru.

Příkaz **Odstranit** vymaže zvolený uzel, všechna jeho rozšíření a jeho virtuální fyzické propojení, a všechny následující uzly budou přemístěny na předchozí adresu.

Příkaz **Přemístit** přemístí zvolený uzel na novou adresu. Následující uzly na zvolené adrese budou přemístěny o jednu polohu.

Příkaz **Kopírovat** uloží údaje zvoleného uzlu do paměti, aby je mohl použít při následujícím příkaze Vložit.

Příkaz **Vložit** předtím zkopírovaný uzel do stromu zařízení. Uzel je vložen do polohy zvolené uživatelem. Všechny následující uzly budou přemístěny.

Rozšíření sítě EtherCAT

Pro vložení rozšíření zvolte uzel, do kterého chcete přidat rozšíření a klikněte na tlačítko **[Nový]**. Indexy rozšíření musí být následné, a proto je ke každému příkazu **[Nový]** přidáno rozšíření na konec řady ostatních zařízení.

Objekty PDO rozšíření nelze měnit.

Popis PDO

Je možné zadefinovat až do 16 PDO odeslaných z uzlu (TxPDO) a až do 16 PDO přijatých z uzlu (RxPDO). Každý RxPDO popisuje jediný PDO, který uzel přijme od zařízení master, a tedy digitální a analogové výstupy pro uzly VST./VÝST. nebo cílové rychlosti a controlword pro uzly jednotky pohybu v ose. Každý TxPDO popisuje jediný PDO, který uzel odesílá do zařízení master, a tedy digitální a analogové vstupy pro uzly VST./VÝST. nebo aktuální polohu a statusword pro uzly jednotky pohybu v ose.

Ohledně seznamu a popisu PDO a zmapovatelných objektů na PDO, je třeba vycházet z dokumentace specifického zařízení EtherCAT a z jeho souboru ESI..

Existují tři způsoby popisu PDO uzlu:

- Neuvést žádný PDO.
Při tomto způsobu začne číslíkové řízení používat PDO nakonfigurované v rámci přednastavení v zařízení.
- Uvést pouze PDO bez poskytnutí seznamu objektů.
Použít tehdy, pokud CN má různé alternativní a nenaprogramovatelné PDO.
- Popsat kompletně PDO a uvést komunikační objekt a seznam objektů ke zmapování.
Tento režim zajistí největší kontrolu informací zasílaných a přijímaných z uzlu.

Každý objekt je popsán svým indexem ve slovníku objektu („object dictionary“) v NC, za kterým může jako nepovinný údaj následovat podindex.

Když podindex není součástí, vychází se z předpokladu, že má hodnotu 0.

Předmět slovníku („object dictionary“) je srdcem každého zařízení. Aktivujte přístup ke všem druhům dat zařízení, ke komunikačním parametrům, ke konfiguračním parametrům a ke zpracování dat.

Upozornění: Ne všechny objekty slovníku objektu mohou být zmapovány v PDO.

Změna PDO pohonu

Co se týče uzlů servopohonů, je k dispozici jeden PDO pro každou jednotku pohybu v ose, tudíž n-tý TxPDO a n-tý RxPDO uzlu se vztahuje na n-tou jednotku pohybu v ose uzlu. První dva objekty každého RxPDO a TxPDO se vyznačují předem přiřazeným významem a předem přiřazenou velikostí. Níže uvedená tabulka popisuje způsob konfigurace PDO týkajících se jednotek pohybu v osách stejného uzlu, řízených v režimu jednotky pohybu v ose **Cyklicky synchronizovaná rychlost**.

Pohon	RxPDO		TxPDO	
	1. objekt 16 bitů Kontrolní slovo (Controlword)	2. objekt 32 bitů Cílová rychlost	1. objekt 16 bitů Stavové slovo (Statusword)	2. objekt 32 bitů Aktuální poloha
1. jednotka pohybu v ose	6040 ₁₆	60FF ₁₆	6041 ₁₆	6064 ₁₆
2. jednotka pohybu v ose	6840 ₁₆	68FF ₁₆	6841 ₁₆	6864 ₁₆
n-tá jednotka pohybu v ose	Přidejte 800 ₁₆ každému objektu jednotky pohybu v ose, která se nachází před ním.			

Níže uvedená tabulka popisuje způsob konfigurace PDO týkajících se jednotek pohybu v osách stejného uzlu CN, řízených v režimu jednotky pohybu v ose **Cyklicky synchronizovaná poloha**.

Pohon	RxPDO		TxPDO	
	1. objekt 16 bitů Kontrolní slovo (Controlword)	2. objekt 32 bitů Cílová poloha	1. objekt 16 bitů Stavové slovo (Statusword)	2. objekt 32 bitů Aktuální poloha
1. jednotka pohybu v ose	6040 ₁₆	607A ₁₆	6041 ₁₆	6064 ₁₆
2. jednotka pohybu v ose	6840 ₁₆	687A ₁₆	6841 ₁₆	6864 ₁₆
n-tá jednotka pohybu v ose	Přidejte 800 ₁₆ ke každému objektu jednotky pohybu v ose, který se nachází před ní.			

Pro změnu objektů jednotky pohybu v ose zvolte jednotku pohybu v ose ve stromovém zobrazení a klikněte na tlačítko **[Změna]**. V dialogovém okně se zobrazí index PDO vysílání, index PDO příjmu a seznam nakonfigurovaných objektů.

Konkrétně jsou údaje v dialogovém okně následující:

- **Index PDO vysílání:** je hodnota PDO vysílání jednotky pohybu v ose. Tuto hodnotu nelze změnit.
- **Objekty:** seznam objektů PDO vysílání. První dva objekty se určí na základě zvoleného režimu jednotky pohybu v ose.
- **Seznam objektů:** seznam použitelných objektů PDO vysílání. Pro každý objekt je nutno kromě indexu definovat rovněž subindex a popis.
- **Index příjmacího PDO:** je hodnota PDO příjmu jednotky pohybu v ose. Tuto hodnotu nelze změnit.
- **Objekty:** seznam objektů PDO příjmu. První dva objekty se určí na základě zvoleného režimu jednotky pohybu v ose.
- **Seznam objektů:** seznam použitelných objektů PDO příjmu. Pro každý objekt je nutno kromě indexu definovat rovněž subindex a popis.

Tlačítka **[Nahoru]** a **[Dolů]** slouží pro provádění změn pořadí objektů v seznamu v okně **Objekty**.

Tlačítko **[Přidat]** umožňuje vložit objekt, který není zahrnutý v **Seznamu objektů**.

Pro změnu hodnoty objektu je nutno zvolit daný objekt a dvakrát na něj kliknout myší. Syntax pro definování nového objektu je:

- **číslo objektu:** je číslo objektu v hexadecimálním formátu.
- **Podobsah:** je číslo subindexu. Musí být odděleno od čísla objektu pomocí znaku `.` (tečka). Jedná se o volitelnou hodnotu, která, pokud není definována, nastaví se na výchozí hodnotu 0. Je v decimálním formátu.

- **L**: rozměr objektu v bitech. Musí to být násobek čísla 8.

Neuvedené hodnoty se získají z dat v object dictionary, nebo dojde k nastavení výchozích hodnot.

Příklad:

kompletní objekt včetně subindexu a délky: 1600.1L16

objekt bez subindexu: 1601L24

jednoduchý objekt: 1603

Tyto hodnoty mohou být přečteny z GPL prostřednictvím instrukce GETAXIS, na kterou vás odkazujeme. Dále je možné stopovat přídavné objekty jednak z okna pro kalibraci, jednak pomocí osciloskopu.

Obecněji lze říci, že z GPL je možný přístup prostřednictvím čtení a zápisu ke specifickým objektům uvnitř PDO prostřednictvím instrukcí. [GETPDO](#) e [SETPDO](#), na které vás odkazujeme.

Doplňkové PDO

Pro definování doplňkových PDO vysílání a příjmu je nutno zvolit uzel ve stromovém zobrazení a poté kliknout na tlačítko **[Změna]**.

V dialogovém okně pro přidání PDO vysílání klikněte na tlačítko **[Doplňkové TPDO]**, pro přidání PDO příjmu klikněte na tlačítko **[Doplňkové RPDO]**.

Dialogové okno pro konfiguraci doplňkového PDO je podobné jako dialogové okno pro vložení objektů.

V tomto okně lze zvolit index PDO a objekty k přiřazení. Pokud má PDO povinné nezměnitelné objekty, tyto se zobrazí a všechna tlačítka pro změny budou deaktivovaná. Nový PDO bude přidán do stromového zobrazení za pohony. Pro úpravu dat je nutno zvolit a kliknout na tlačítko **[Změna]**. Pro odstranění doplňkového PDO je nutno zvolit daný PDO ve stromovém zobrazení a poté kliknout na tlačítko **[Odstranit]**.

Automatické získání uzlů EtherCAT

Když se v konfiguraci hardwaru nachází síť EtherCAT, je možné získat ze sítě připojené uzly, a to prostřednictvím příkazu **Automatické získání uzlů**, který lze aktivovat z kontextové nabídky. Aby mohl být příkaz proveden, údaje konfigurace hardwaru a údaje, které se nacházejí v číslicovém řízení, musí být sladěny a soubory ESI, které popisují uzel na síti, se musí nacházet v adresáři definovaném Tpa.ini, v sekci [tpa], pod položkou DirESIFiles.

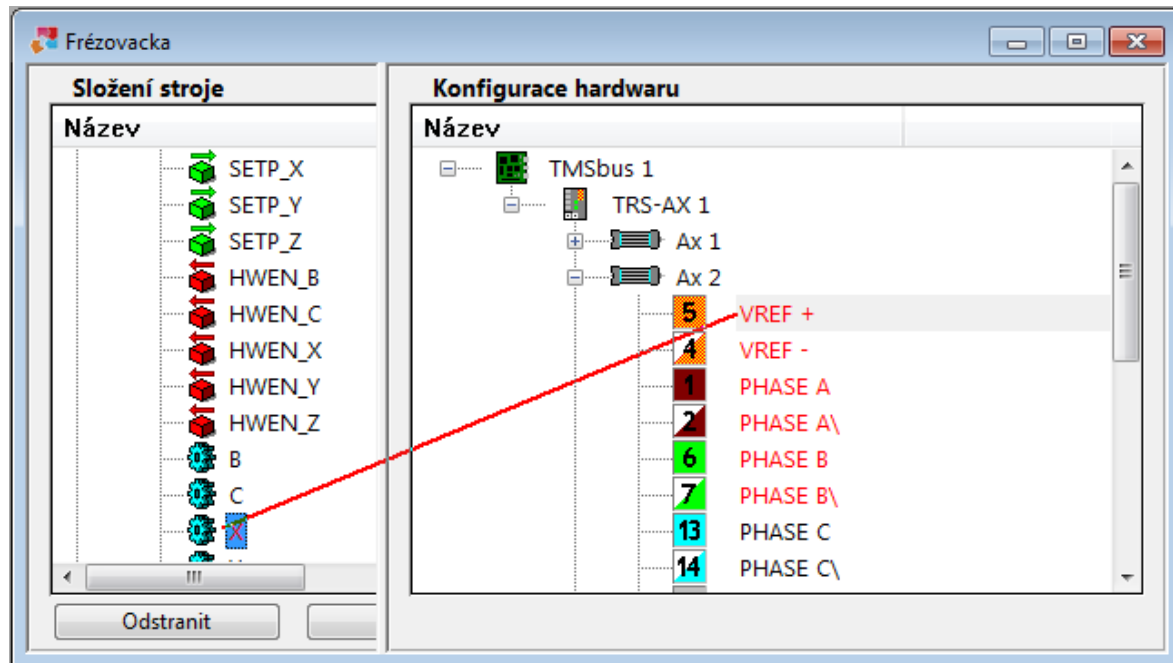
Získání uzlů sítě EtherCAT ze sítě postupuje v souladu s níže uvedenými pravidly:

- konfigurace hardwaru není nakonfigurován žádný uzel: údaje týkající se uzlů budou přečteny ze sítě a zobrazeny v konfiguraci
- do konfigurace hardwaru již byly vloženy uzly sítě EtherCAT:
 - když má uzel přečtený ze sítě stejnou adresu, stejné OZN. dodavatele (Vendor ID), stejné OZN. produktu (Product ID) a stejné číslo revize (Revision Number) nakonfigurovaného uzlu, bude zachován nakonfigurovaný uzel se všemi svými PDO
 - když má uzel přečtený ze sítě stejnou adresu jako nakonfigurovaný uzel, ale má odlišné OZN. dodavatele (Vendor ID) nebo odlišné OZN. produktu (Product ID), uzel konfigurace bude odstraněn spolu se svými virtuálními-fyzickými, a bude nahrazen uzlem sítě. Když je uzlem sítě pohon, PDO budou přečtené ze souboru ESI; v opačném případě budou přečteny z uzlu sítě.
 - když má uzel přečtený ze sítě stejnou adresu, stejné OZN. dodavatele (Vendor ID), stejné OZN. produktu (Product ID) a odlišné číslo revize (Revision Number) nakonfigurovaného uzlu, bude nahrazeno pouze číslo revize (Revision Number) a PDO budou zachovány.
 - když je uzel definovaný v konfiguraci nemá odpovídající uzel na síti EtherCAT, který zabírá stejnou adresu, bude zrušen v konfiguraci a virtuální-fyzický bude zachován.

7.4.3 Konfigurace Virtuální-Fyzické

Konfigurace virtuální fyzické je poslední fází konfigurace a spočívá v přiřazení logických zařízení komponentům hardwaru.

Při otevření Konfigurace Virtuální-Fyzické se zobrazí dvě okna: Nalevo okno Složení Stroje (virtuální), napravo okno Konfigurace Hardwaru (fyzické). V obou jsou v grafické podobě a v podobě struktury stromu zobrazeny všechny prvky, které tvoří systém.



Konfigurace Virtuální-Fyzické

Již vytvořená propojení virtuální-fyzické jsou zvýrazněna v okně „Složení stroje“ názvem zařízení červené barvy, zatímco v okně „Konfigurace hardwaru“ názvem druhu signálu, který následuje po čísle svorky, rovněž červené barvy.

Pro každou jednotku pohybu v ose karty MECHATROLINK-II lze v rámci virtuálního fyzického stavu nakonfigurovat 6 vstupů a 1 digitální výstup. Podrobnější popis je uveden v kapitole **Jazyk GPL->Instrukce->MECHATROLINK-II->MECGETSTATUS**.

Když se na modulu nachází sběrnice EtherCAT, je v každém případě možné provést konfiguraci karet pro sběrnici MECHATROLINK-II, ale s několika omezeními: s hodnotou real-time 1 ms nesmí být připojeno více než šest jednotek pohybu v osách MECHATROLINK-II (pro každou sběrnici); s hodnotou real-time 2 ms limit stoupne na 16 jednotek pohybu v osách.

Zařízení nebo svorky, které je třeba ještě připojit, jsou zobrazeny černou barvou. Šedě jsou znázorněna zařízení, která přináležejí ke zrušené skupině, jejíž konfigurace ve virtuálně-fyzické sekci má být zachována.

Signálům, které se týkají jednotek pohybu v osách, v okně „Konfigurace stroje“ předchází obdélník, jehož barva odpovídá barvě izolace vodiče uvnitř spojovacího kabelu.

Propojení je možné zvýraznit volbou logického zařízení (nebo komponentu hardwaru) a stisknutím mezerníku bude propojení zvýrazněno jako červená čára, která spojuje zařízení s komponentem hardwaru. Dále je možné zachovat stále aktivní zobrazení propojení prostřednictvím příkazu **[Alt+Enter]**.

Pro zobrazení toho, které logické zařízení je připojeno ke komponentu hardwaru, je třeba zvolit komponent hardwaru a provést dvojité kliknutí myši.

Za účelem volby logického a fyzického zařízení, které má být připojeno, je možné použít více postupů:

první postup

- Zviditelnit na displeji, v okně „Konfigurace hardwaru“, fyzickou svorku, ke které má být zařízení připojeno
- Zvolte nebo zaškrtněte logické zařízení v okně „Složení stroje“

druhý postup

- Zvolte nebo zaškrtněte požadované logické zařízení v okně „Složení stroje“
- Zvolte příkaz v menu **Úpravy->Najít vhodné fyzické zařízení** nebo kombinaci tlačítek **[Ctrl+mezera]**. V automatickém režimu Albatros zobrazuje v okně „Konfigurace hardwaru“ první volné fyzické zařízení, ke kterému může být připojeno logické zařízení.

třetí možný postup

- Zvolte nebo zaškrtněte požadované virtuální zařízení v okně „Složení stroje“
- Zvolte příkaz v menu **Úpravy->Najít následující nepřipojené zařízení** nebo kombinaci tlačítek **[Ctrl+NumPad+]** nebo příkaz **Úpravy->Najít předchozí nepřipojené zařízení** nebo kombinaci tlačítek **[Ctrl+NumPad-]**.

Za účelem provedení propojení mezi zvolenými zařízeními:

- Na logickém zařízení, které má být připojeno, stiskněte levé tlačítko myši a držte jej stisknuté a přetáhněte myš ve směru zvolené svorky. Zobrazí se červený řádek, který zvýrazňuje probíhající připojení. Po přesunutí myši na řádek svorky ukončete operaci uvolněním tlačítka
- Zvolte příkaz **Připojit!** v menu **Úpravy** nebo kombinaci tlačítek **[Ctrl+L]**.

Za účelem odstranění připojení je třeba zvolit zařízení nebo komponent a stisknout tlačítko **[Odstranit]** nebo tlačítko **[Delete]** na klávesnici.

7.4.4 Mapy kabeláže

Po provedení spojení mezi virtuálními a odpovídajícími fyzickými zařízeními je možné vytisknout mapy nebo seznamy, ve kterých je uvedeno přiřazení mezi fyzickými a virtuálními zařízeními.

Za účelem provedení této operace je třeba, aby byl v systému nainstalován MS-Word (verze 6 nebo vyšší verze); Albatros využívá jeho funkce pro formátování map.

Kromě toho musí být systém nakonfigurován správně. To znamená, že v systému musí být přítomné soubory modelu, použité pro vyplnění map. Tyto soubory tvoří sérii souborů s příponou „.doc“, které jsou obvykle uloženy v adresáři System nebo v jiném adresáři instalace (obvykle se jedná o adresář „Mappe“ (Mapy)). Podstatné je, aby se adresář, ve kterém se nacházejí tyto soubory, shodoval s adresářem uvedeným v souboru tpa.ini prostřednictvím klíče „DirMaps“. Například:

```
[TPA]
DirMaps=C:\<Albatros/>\Mappe
```

Pro tisk map kabeláže je třeba zvolit kterýkoli komponent hardwaru, který se proto nachází v okně napravo od [konfigurace Virtuálních-Fyzických zařízení](#) nebo který se nachází v okně [Konfigurace hardwaru](#).

Stisknutím tlačítka „Tisk“ panelu nástrojů nebo volbou položky **Tisk** v menu **Soubor** se zobrazí běžné okno pro volbu možností tisku. Po provedení našich voleb je třeba provést jejich potvrzení stisknutím tlačítka **[OK]**. Poté se zobrazí okno se seznamem komponentů hardwaru přítomných v konfiguraci.

V tomto okně můžeme zvolit komponenty, pro které si přejeme vytisknout mapu kabeláže.

Po stisknutí tlačítka **[OK]** se vytisknou mapy kabeláže. V případě volby možnosti **Tisk na papír** budou mapy uloženy jako dokumenty MS-Word do adresáře „mappe“ + název aktuálního modulu (Mod. 0 apod.).

Vzhledem k vysokému počtu stran, které se dají vytisknout, se doporučuje provést zkoušku, a to tak, že se vytiskne pouze mapa jediného komponentu hardware za účelem ověření toho, že vše funguje předepsaným způsobem. Když bude namísto map vytištěn seznam logických zařízení, pravděpodobně nebyl zvolen komponent (například karta jednotek pohybu v osách nebo vzdálený komponent) v okně týkajícím se hardwaru.

7.5 Seznam tlačítek pro pohyb uvnitř struktury ve formě stromu

Tlačítko	Popis
Šipka Nahoru	slouží k přemístění volby na bezprostředně předchozí nebo následující řádek
Šipka Dolů	
Šipka Doprava	slouží k rozbalení úrovně zvolené větve a v případě, že je větev již rozbalena, k přemístění na následující větev
Šipka Doleva	slouží k zavření zvolené větve a v případě, že je větev již zavřena, k přemístění na předchozí větev
+	slouží k rozbalení úrovně zvolené větve
-	slouží k zavření zvolené větve
*	slouží k rozbalení všech úrovní zvolené větve

8 Vývojové nástroje

8.1 Editor GPL

8.1.1 Funkce editoru GPL

Editor GPL představuje nástroj, který umožňuje vytváření a změnu souborů obsahujících kód GPL softwaru Albatros. Tato funkce je aktivní pouze v případě úrovně hesla, která je stejná jako úroveň výrobce nebo ji převyšuje. Každému souboru funkcí jsou přiřazeny informace, které lze zobrazit prostřednictvím menu **Soubor->Informace**.

Funkčními režimy jsou obvyklé režimy editoru textu, a proto zde najdeme příkazy **Kopírovat**, **Vložit**, **Najít**, **Nahradit** apod. Všechny tyto příkazy lze zvolit v menu **Úpravy**.

Zrušit	Umožňuje, pokud je to možné, zrušit provedenou operaci. Bude provedeno obnovení situace, která předcházela poslední provedené operaci.
Zopakovat	Slouží k obnovení situace, která předcházela poslednímu příkazu Zrušit
Vyjmout	Bude eliminován text nebo zvolená data a budou kopírovány do vyrovnávací paměti s cílem umožnit případné opětovné vložení prostřednictvím příkazu <i>Vložit</i> .
Kopírovat	Text nebo zvolený prvek bude kopírován do vyrovnávací paměti s cílem umožnit případné opětovné vložení prostřednictvím příkazu <i>Vložit</i> .
Vložit	Obsah vyrovnávací paměti bude vložen s použitím kritérií, která se budou v jednotlivých případech lišit v závislosti na aktivní funkci.
Odstranit	Tento příkaz slouží k vymazání zvoleného textu, řádků nebo prvku. To, co bylo vymazáno, lze obnovit prostřednictvím příkazu <i>Zrušit</i> .
Zvolit Vše	Umožňuje zvolit veškerý text aktivního souboru Na zvolené řádky lze aplikovat příkazy Kopírovat, Vyjmout, Vložit.
Najít...	Slouží k vyhledání textu v aktuálním dokumentu. Je možné nastavit kritéria pro použití během vyhledávání: směr vyhledávání, rozlišování malých a velkých písmen, vyhledávání celého slova a vyhledávání prostřednictvím regulérních výrazů.
Najít Následující	Umožňuje zopakovat předchozí vyhledávání s možností změnit kritéria vyhledávání, nastavená příkazem Najít.
Nahradit	Umožňuje vyhledat text v aktuálním dokumentu a nahradit jej jiným textem.
Vložit zařízení	Slouží k vložení zařízení na základě jeho volby v seznamu zařízení. Tato funkce je mimořádně pohodlná při práci s vysokým počtem zařízení, u kterých může být obtížné pamatovat si jejich názvy. Budou zobrazena pouze ta zařízení aktuálního modulu, která mohou být vyvolána, a všechna veřejná zařízení ostatních modulů
Vložit funkci	Slouží k vložení funkce nebo části funkce počínaje polohou kurzoru, a to po jejím přečtení ze souboru prototypu, napsaného výrobcem stroje. Je možné napsat více souborů prototypu. Soubor prototypu je textový soubor, jehož název musí začínat předponou GPL a přípona musí být TXT. Musí být uložen v adresáři, ve kterém jsou archivovány knihovny (obvykle system\lib). V případě definování více souborů prototypu dojde při volbě příkazu k otevření dialogového okna, ve kterém je předložen seznam názvů souborů prototypu bez předpony a bez přípony. Soubory prototypu mohou obsahovat například definice konstant pro společné použití, funkce pro správu chyb systému, všeobecné funkce, kódy, které implementují algoritmy pro všeobecné použití apod. Mohou obsahovat také komentáře. Soubor prototypu může být vytvořen uložením zvoleného textu do souboru funkcí GPL. Příkaz je k dispozici pouze v podobě rychlé volby z klávesnice [Ctrl+Shift+C]. Dojde k otevření dialogového okna pro zadání názvu, který má být přiřazen části kódu.
Vložit hlášení	Slouží k vložení číselného kódu do textu GPL, přiřazeného ke zvolenému hlášení. Umožňuje vložení nových hlášení do souboru jazyka.
Slouží k aktivaci/zrušení přeskočení strany	Slouží k vložení nebo zrušení přeskočení strany  . Přeskočení strany lze použít jako záložku pro přeskočení do významných poloh uvnitř souboru funkcí.
Přejít na následující stranu	Slouží k přemístění kurzoru editace na řádek pro přeskočení strany následující po jeho poloze
Přejít na předchozí stranu	Slouží k přemístění kurzoru editace na řádek pro přeskočení strany předchozí vůči jeho poloze

```

Function AbsMovement
  param axisname as axis
  param speed as float
  param position as double

  iftarget axisname goto move
  ifstill axisname goto move
  fret
move:
  setvel axisname, speed
  movabs axisname, position
  waitstill axisname
  fret

```

Editor GPL

Kontrola správnosti syntaxe je prováděna ve fázi archivace, když je text také podroben kompilaci. V každém případě je možná první vizuální analýza ze strany programátora, protože text je zobrazován různými barvami v závislosti na tom, co představuje. Například instrukce jsou modré barvy, komentáře zelené a návěští červené. Je možné měnit hodnotu tabulací v menu **Možnosti->Zarážky tabulátorů....** Je možné definovat dva druhy zarážek tabulátorů:

- absolutní zarážka tabulátoru: Nastavuje počáteční polohu pro instrukce kódu GPL, počáteční polohu prvního argumentu instrukcí a počáteční polohu pro komentář.
- relativní zarážka tabulátoru (mezery): Nastavuje, po jakém počtu mezer má být umístěna zarážka tabulátoru. Zarážky tabulátorů představují pomoc ve snaze učinit kód GPL vizuálně srozumitelnějším.

Pro každou instrukci nebo klíčové slovo je dostupná nápověda online jako podpora při definici funkce. Pro vyvolání nápovědy stačí umístit kurzor na instrukci a stisknout tlačítko **[F1]**.

Na každý řádek textu lze napsat pouze jednu instrukci. Je možné pokračovat v instrukci na následujícím řádku zadáním znaku '_' (kterému předchází jedna mezera) na konci řádku. To umožňuje vložit komentáře doprostřed instrukce:

```

Message
1000      ;kód hlášení, které bude zobrazeno      _
3         ;pole grafického zobrazení, ve kterém bude zobrazeno [Enter].

```

Použití pravidelných výrazů

V okně **Najít** a v okně **Nahradit** je možné použít pravidelné výrazy. Pravidelný výraz je posloupnost číslíkových a abecedních znaků, použitých ve snaze vyhledat a nahradit části textu uvnitř většího textu. Albatros používá gramatiku pravidelných výrazů ECMAScript. V tomto odstavci jsou popsány hlavní použití pravidelných výrazů v softwaru Albatros.

Nejjednodušší případ pravidelného výrazu je tvořen samostatným znakem. Existují výjimky, tvořené následujícími znaky:

- **.** (**bod**): bod, ve kterém se nachází jakýkoli znak. Například „A.“ najde všechny výskyty velkého písmena A, po kterém následují dva libovolné znaky.

- **[] (hranaté závorky)**: závorky, které umožňují mezi nimi seznam znaků. V textu budou hledány výskyty pro každý z přítomných znaků. Když je prvním znakem **^ (stříška)** budou hledány všechny znaky, s výjimkou těch v hranatých závorkách.

Například:

[<>]: budou hledány všechny výskyty znaku < a znaku >.

[.]AX: budou hledány všechny položky, které mají uvnitř řetězec „AX“.

[a-d]: budou hledány všechny výskyty znaků a, b, c a d. Pomlčka označuje soubor znaků.

[\[]]: budou hledány všechny výskyty znaku [a znaku].

[^+]: budou hledány všechny znaky s výjimkou znaku +.

- ***(hvězdička)**: budou hledány všechny výskyty znaku (nebo skupiny znaků), které se nacházejí před hvězdičkou. Hvězdička má vliv pouze na jeden znak, který jí předchází: aby působil na skupinu znaků, je třeba použít hranaté závorky.

Například:

;-*: budou hledány všechny výskyty znaků ; a ;- a ;-----.

- **+(znaménko plus)**: vyhledá jeden nebo více výskytů znaku, které mu předchází. Liší se od použití hvězdičky, protože znak, který předchází znaménku + musí být vždy přítomen. Znaménko plus působí pouze na znak, který mu předchází: aby působil na skupinu znaků, je třeba použít hranaté závorky. Vycházejme znovu z příkladu, použitého pro znak hvězdičky:

;-+: budou hledány všechny výskyty znaků ;- a ;---. Nebude hledán pouze znak ; **(středník)**.

- **? (otazník)**: dělá volitelným znak, který mu předchází, který proto může být přítomen maximálně jednou.

Například:

Setfeedi?: budou hledány všechny výskyty slova Setfeed a slova Setfeedi.

- **{ } (složené závorky)**: určují, kolikrát se znak (nebo skupina znaků) musí nacházet uvnitř textu.

Například:

ee{2}: jsou hledány všechny výskyty dvou následných ee.

- **^ (stříška a tečka)**: najde první znak každého řádku.

- **^ (stříška)**: najde hledaný výraz pouze v případě, když se nachází na začátku řádku

- **| (kanál)**: najde výrazy, které se zobrazí před a po znaku `|`.

Například:

Send|Const: budou hledány všechny výskyty slova Send a slova Const.

- **\ (zpětné lomítko)**: zpětné lomítko plní dvojí funkci:

1) transformuje běžný znak na funkci.

\b najde počáteční nebo koncový okraj slova

\B najde slovo kromě počátečního okraje

\d najde pouze číslice

\D najde všechno kromě číslic

\s najde mezeru

\S najde všechno kromě mezery

Příklady:

\bi najde všechna slova v textu, která začínají písmenem i.

2) transformuje speciální znak na běžný znak. Například pro vyhledání speciálního znaku **^ (stříška)** stačí umístit před něj zpětné lomítko.

8.1.2 Vložit hlášení

V softwaru Albatros existují dva druhy hlášení: hlášení modulu a hlášení skupiny.

Příkaz lze zvolit v menu **Úpravy->Vložit hlášení**.

Hlášení skupiny budou vložena přímo v editoru během zápisu kódu GPL prostřednictvím instrukce [DEFMSG](#). Jsou viditelná a použitelná pouze uvnitř skupiny, ve které byla definována, a proto je možné mít stejnou definici hlášení ve více skupinách, aniž by došlo k jakémukoli překrývání.

Na rozdíl od hlášení skupiny mohou být hlášení modulu použita všemi skupinami. Mohou být vložena s použitím dialogového okna, které umožňuje volat hlášení, které se již nachází v souboru jazyka, nebo vložit nové.

The screenshot shows a dialog box titled "Správa hlášení a chyb cyklu". On the left, there is a "Popis" label above a dropdown menu. On the right, there is a "Kód" label above a vertical stack of four buttons: "Upravit text", "Zavřít", "Nový", and "Úpravy". The "Upravit text" button is highlighted with a blue border.

Dialogové okno hlášení

V tomto režimu je třeba pamatovat, že třeba změnit soubor. Hlášení bude vloženo v aktuálním jazyce a následně bude třeba provést jeho překlad do ostatních jazyků (s použitím příslušného programu TpaLangs).

Všechna hlášení přítomná v souboru jazyka jsou uvedena v seznamu pod položkou **Popis**. Pro vložení hlášení dovnitř funkce je třeba po provedení volby textu zvolit tlačítko **[Upravit text]**.

Za účelem změny existujícího hlášení **[Úpravy]** nebo vytvoření nového **[Nový]** je třeba jako první operaci zadat změnu nebo nový text a poté stisknout odpovídající tlačítko.

8.1.3 Šifrování

V softwaru Albatros je možné prostřednictvím šifrování znemožnit zobrazování zdrojového textu funkcí.

Aktivace šifrování se provádí nastavením položky Tele+= 1. Přednastavenou hodnotou je hodnota 0, která odpovídá zrušenému šifrování.

Při uložení souboru funkcí při aktivovaném šifrování se zobrazí hlášení: „Přejete si zašifrovat soubor?“ Při odpovědi [Ano] bude soubor zašifrován. Je možné zašifrovat soubor, který byl předtím uložen bez šifrování, zatímco zašifrovaný soubor zůstane zašifrovaný.

Když je aktivováno šifrování a heslo je typu Denní heslo výrobce, soubor uložený bez šifrování nebude šifrován. Soubor se zašifrovanými funkcemi bude muset být zobrazen nebo měněn v softwaru Albatros pouze uživatelem, který jej předtím uložil. Majitel zašifrovaného souboru funkcí nemůže tento soubor měnit!

V případě, že by bylo třeba odstranit zašifrování souboru, bude třeba použít externí program nazvaný SBIANCA, který se nachází v adresáři Bin softwaru Albatros.

Jedním ze způsobů použití programu Sbianca je jeho použití prostřednictvím grafického rozhraní. Provede se volba souborů, které mají být dešifrovány. Pro každý soubor budou zobrazeny vlastnosti **Stav** a **Přístupové údaje**. **Stav** může být **Bez šifrování**, když je soubor uložen bez šifrování, nebo **Šifrovaný**, když je soubor zašifrovaný. Vlastnost **Přístupové údaje** poskytuje informace ohledně viditelnosti souboru: **Čitelný** znamená, že soubor může být zobrazován z aktuální úrovně hesla, zatímco **Zablokovaný** znamená, že soubor nemůže být zobrazován. Po provedení volby souborů je třeba kliknout na **[Dešifrovat!]** nebo na **[Zašifrovat!]** za účelem jejich zašifrování/dešifrování.

Druhým způsobem použití je z příkazového řádku. Příkazy a soubory jsou předány programu jako argumenty. Když nejsou přítomné argumenty, bude aktivován způsob použití prostřednictvím grafického rozhraní. Příslušná syntax je:

```
[-l|-e|-d] soubor1 soubor2 ...
```

kde:

- -l, nebo žádná možnost, pro zobrazení stavu každého souboru
- -d pro dešifrování souborů
- -e pro šifrování souborů

Po ukončení provedení program odešle na výstup výsledky požadované operace.

Výstup programu je ve formátu markdown.

8.1.4 Seznam dostupných tlačítek rychlé volby

<ul style="list-style-type: none"> [-] Vymazání textu Tlačítko Backspace Ctrl+Backspace Delete Ctrl+T Ctrl+Delete 	<p>Popis</p> <p>slouží k odstranění znaku vlevo nebo k odstranění zvoleného textu</p> <p>slouží k odstranění slova vlevo</p> <p>slouží k odstranění znaku vpravo nebo k odstranění zvoleného textu</p> <p>slouží k odstranění slov nebo mezer vpravo</p> <p>slouží k odstranění slova vpravo a všech následujících mezer až do začátku nového slova</p>
<ul style="list-style-type: none"> [-] Komentář více řádků textu Tlačítko Ctrl+';'. Na italských klávesnicích je třeba stisknout také tlačítko [Shift] 	<p>Popis</p> <p>slouží k přidání nebo odebrání znaku komentáře ve zvolených řádcích.</p>
<ul style="list-style-type: none"> [-] Pohyb kurzoru Tlačítko Šipka Nahoru Šipka Dolů Šipka Doprava Šipka Doleva Home End Ctrl+Home Ctrl+End Ctrl+Šipka Doleva Ctrl+Šipka Doprava Ctrl+Enter 	<p>Popis</p> <p>slouží k přesunu kurzoru ve zvoleném směru</p> <p>slouží ke střídavému přesouvání kurzoru na začátek řádku a na první znak řádku</p> <p>slouží k přesunutí kurzoru na konec řádku</p> <p>slouží k přesunutí kurzoru na začátek dokumentu</p> <p>slouží k přesunutí kurzoru na konec dokumentu</p> <p>slouží k přesunu kurzoru o jedno slovo doleva</p> <p>slouží k přesunu kurzoru o jedno slovo doprava</p> <p>slouží k přesunu kurzoru na první znak následujícího řádku</p>
<ul style="list-style-type: none"> [-] Volba Tlačítko Shift+Home Ctrl+Shift+Home Ctrl+Shift+End Ctrl+Shift+Šipka Doleva Ctrl+Shift+Šipka Doprava Shift+Stránka Nahoru Shift+Stránka Dolů Ctrl+W Ctrl+A 	<p>Popis</p> <p>slouží k volbě až po začátek řádku počínaje polohou kurzoru</p> <p>slouží k volbě až po začátek dokumentu počínaje polohou kurzoru</p> <p>slouží k volbě až po konec dokumentu počínaje polohou kurzoru</p> <p>slouží k volbě slova nebo mezer nalevo od kurzoru</p> <p>slouží k volbě slova nebo mezer napravo od kurzoru</p> <p>slouží k volbě strany směrem nahoru počínaje polohou kurzoru</p> <p>slouží k volbě strany směrem dolů počínaje polohou kurzoru</p> <p>slouží k volbě slova v místě, na kterém se nachází kurzor</p> <p>slouží k volbě celého dokumentu</p>
<ul style="list-style-type: none"> [-] Čtvercová volba Tlačítko Alt+ Shift+Šipka Nahoru Shift+Šipka Dolů Shift+Šipka Doleva Shift+Šipka Doprava nebo Alt+volba myši 	<p>Popis</p> <p>slouží k volbě čtvercového bloku kódu</p>
<ul style="list-style-type: none"> [-] Zarážky tabulátoru Tlačítko Tab Shift+Tab 	<p>Popis</p> <p>v případě, že není zvolen text, slouží k vložení mezer mezi znaky; mezery jsou definovány v Možnosti->Zarážky tabulátoru. V případě, že bylo zvoleno více řádků, vloží napravo mezery nastavené pro příslušnou zarážku tabulátoru.</p> <p>v případě, že není zvolen text, slouží k přesunu kurzoru nalevo od mezery definované v Možnosti->Zarážky tabulátoru. V případě, že byl zvolen jeden nebo více řádků, tyto budou přesunuty nalevo od mezery nastavené pro příslušnou zarážku.</p>
<ul style="list-style-type: none"> [-] Kopírovat a Vložit Tlačítko Ctrl+C Ctrl+Ins Ctrl+X Shift+Del 	<p>Popis</p> <p>slouží ke zkopírování zvoleného textu do Schránky</p> <p>slouží k odstranění zvoleného textu a k jeho zkopírování do Schránky</p>

Ctrl+V Shift+Ins Ctrl+Y	slouží k vložení obsahu Schránky počínaje polohou kurzoru
Přetažení pomocí myši (Drag'n'drop) Ctrl+Přetažení pomocí myši (Drag'n'drop)	slouží k odstranění řádku, na kterém se nachází kurzor, a k jeho zkopírování do Schránky slouží k přetažení zvoleného textu a k jeho přesunutí do nové polohy
☐ Zrušit / Obnovit Tlačítko Ctrl+Z Alt+BackSpace Ctrl+Shift+Z	Popis slouží ke zrušení posledního zadání slouží k obnovení posledního zadání
☐ Vyhledání a Výměna Tlačítko Ctrl+F3 Ctrl+Shift+F3 F3 Shift+F3 Alt+F3	Popis slouží k vyhledání slova, na kterém je umístěn kurzor, směrem dolů v celém dokumentu slouží k vyhledání slova, na kterém je umístěn kurzor, směrem nahoru v celém dokumentu slouží k vyhledání následujícího výskytu. Dialogové okno Vyhledat musí být zavřeno. slouží k vyhledání předchozího výskytu. Dialogové okno Vyhledat musí být zavřeno. slouží k otevření dialogového okna Vyhledat a k nastavení slova, které se nachází v místě kurzoru, jako textu k vyhledání
☐ Zobrazení chyb kompilace Tlačítko Dvojitě kliknutí na chybu F4 Shift+F4	Popis slouží k přemístění kurzoru na řádek funkce GPL, ve kterém došlo k výskytu popsané chyby slouží k přemístění kurzoru na řádek funkce GPL, ve kterém došlo k výskytu chyby, která následuje po poslední zvolené chybě slouží k přemístění kurzoru na řádek funkce GPL, ve kterém došlo k výskytu chyby, která předchází poslední zvolené chybě.
☐ Vytvoření souboru prototypu Tlačítko Ctrl+Shift+C	Popis slouží k uložení zvoleného textu do souboru funkcí GPL. Dojde k otevření dialogového okna pro zadání názvu, který má být přiřazen části kódu.
☐ Správa skládání Tlačítko Ctrl+M	Popis Slouží k otevření nebo zavření zvoleného skládání.

8.2 Knihovny

Knihovna je tvořena sbírkou funkcí GPL, které mohou být vyvolány uvnitř uživatelského kódu GPL, aniž by musely být navázány na specifickou konfiguraci. Knihovny jsou velmi užitečné, protože mohou být snadno kopírovány z jednoho stroje do druhého. Zabrání se tak nutnosti opětovného přepisování všeobecného kódu při implementování nových strojů. Například je takto možné vytvořit knihovnu geometrických a matematických funkcí.

Soubory knihovny jsou archivovány do adresáře system\lib. Budou podrobeny kompilaci při provedení jednoho z níže uvedených příkazů: **CNC->Inicializace, Soubor->Kompilovat vše, Uložit** soubor knihovny nebo soubor globálních proměnných.

Když bude v kódu GPL některého stroje provedeno přiřazení názvu funkce nebo proměnné, která již existuje v knihovně, ve fázi kompilace bude mít přednost kód stroje. V případě, že stejný název existuje ve dvou odlišných knihovnách, je k identifikaci toho, který z nich je předmětem zájmu, vhodné v kódu GPL použít níže uvedenou syntaxi: **názevknihovny.názevfunkce**. Například funkce DélkaSegmentu existuje v knihovně LIBGEO i v knihovně LIBMAT, a když si přejete použít funkci knihovny LIBGEO, je třeba napsat: LIBGEO.DélkaSegmentu.

Všechny operace, které se týkají jedné knihovny, jsou spravovány prostřednictvím dialogového okna. Je možné vytvořit **[Novou]** knihovnu. Přiřazený název bude uveden v seznamu nainstalovaných knihoven. Je možné importovat již existující knihovny nebo přetransformovat soubor skupin na knihovny jejich vyvoláním prostřednictvím dialogového okna otevřeného tlačítkem **[Importovat...]**. Knihovny odstraněné tlačítkem **[Odstranit]** budou přesunuty do koše v operačním systému Windows.

Pro změnu kódu knihovny je třeba zvolit tlačítko **[Změna]**. Knihovna tak bude otevřena editorem GPL. Během vytváření funkcí knihovny je třeba mít na paměti tato základní pravidla:

- Není možný přístup ke specifickým zařízením, funkcím, proměnným konfigurace, uvnitř které se zapisuje funkce.
- je možné vyvolat funkce a veřejné proměnné z jiných knihoven
- funkce deklarované uvnitř knihovny jsou v rámci přednastavení definovány jako soukromé. Aby bylo možné je vyvolat z jiných souborů funkcí, je třeba, aby byly deklarovány jako **PUBLIC**.

Možnost změny knihovny je podřízena úrovni přístupu používaného softwarem Albatros. Je možné přiřadit nebo změnit autorizace pro přístup do knihovny volbou tlačítka **[Vlastnosti]**.

Globální proměnné deklarované v knihovně jsou zobrazovány v **Diagnostic**, v její specifické sekci. Zobrazování prvků knihovny souvisí s přístupovými právy toho, kdo používá Albatros.

8.3 Odlad'ování

8.3.1 Ladicí program

Ladicí program představuje funkci softwaru Albatros, která umožňuje sledovat krok za krokem posloupnost instrukcí úkolu GPL a umožňuje tedy pochopit a korigovat případné logické chyby a nechtěná chování kódu.

Tato funkce je aktivní pouze v případě úrovně hesla, která je stejná jako úroveň výrobce nebo ji převyšuje. S ladicím programem je například možné:

- Přiřadit body přerušení
- přerušit provádění úkolu a zobrazit hodnotu proměnné
- sledovat posloupnost provádění funkce
- zkontrolovat hodnotu, kterou nabyla místní proměnná
- zkontrolovat v případě instrukce If, zda je zvolená větev správná

Příkazy použitelné v režimu ladění lze zvolit v menu **Odlad'ování**. K hlavním příkazům patří:

Pokračovat	Slouží k obnovení zablokovaného úkolu. Úkol bude pokračovat až do svého ukončení, nebude znovu zablokován ani nebude dosaženo bodu přerušení.
Restartovat	Slouží k restartování ladění zvoleného úkolu.
Přerušit nyní	Slouží k zablokování provádění úkolu, jehož ladění je právě prováděno. Kurzor se přesune na řádek, na kterém došlo k přerušení instrukce. Po zablokování úkolu je možné pilotovat jeho provádění a zkontrolovat stav místních proměnných
Provést instrukci	Provést jednotlivou instrukci GPL. Předpokladem je předchozí přerušení úkolu.
Provést až po konec funkce	Slouží k provedení instrukcí až po první instrukci, která následuje po aktuální funkci
Provést instrukci/funkci	Slouží k provedení jednotlivé instrukce GPL nebo k provedení celé funkce, když je instrukcí volání funkce
Provést až po kurzor	Slouží k provedení instrukcí až po polohu kurzoru.
Ukončit odlad'ování	Ukončit ladicí činnost. Soubor funkcí, jehož ladění je právě prováděno, bude otevřen v režimu editoru.

Za účelem přístupu k ladicímu programu je třeba zobrazit **seznam prováděných úkolů** (v menu **Odlad'ování->Prováděný úkol**) nebo **seznam všech úkolů** (z menu **Odlad'ování->Všechny úkoly**) a poté zobrazit úkol, jehož ladění má být provedeno.

Před provedením ladění je třeba, aby nebyly přítomné chyby kompilace funkcí (například: chyby syntaxe, nedeklarované proměnné) a aby byl modul, jehož ladění má být provedeno, správně inicializován.

Okno ladění je podobné jako okno editoru GPL, není v něm však možné provádět změny kódu. Pozadí okna je šedé a čára provádění je výrazně žlutou barvou.

Poznámka: Není možné současně provést ladění více úkolů, které patří do stejného modulu.

8.3.2 Provádění úkolu










Příkaz lze zvolit v menu **Odlad'ování->Prováděný úkol**. Slouží k zobrazení seznamu prováděných úkolů přiřazených stroji nebo modulu. Volbou úkolu je možné **provést jeho ladění** nebo ukončit jeho provádění s použitím tlačítek **[Odlad'ování]** nebo **[Ukončit]**.

8.3.3 Všechny úkoly

Slouží k zobrazení všech úkolů, které byly definovány v kódu GPL v dialogovém okně. Graficky jsou znázorněny ve struktuře ve formě stromu. Při zvolení funkce dojde k otevření souboru, ve kterém je tato funkce definována, a kurzor bude přemístěn na jeho první instrukci. Tímto způsobem je možné nastavit [Body přerušení](#) ještě před provedením.

Je možné provést odladění jakéhokoli úkolu a funkce bez parametrů.

Níže je uveden popis významu každého symbolu použitého pro složení stromu provádění úkolů. Speciální symbol je ten, který označuje odvolávací funkci, která vyvolá ve svém vnitřku funkci, z níž je vyvolána.

Symbol	Popis
	úkol hlavní funkce meziskupiny
	úkol automatického spuštění (autorun)
	všeobecný úkol
	úkol v reálném čase (real-time)
	funkce skupiny
	funkce skupiny provedená z instrukcí jako ONINPUT, ONFLAG
	funkce knihovny
	funkce knihovny provedená z instrukcí jako ONINPUT, ONFLAG
	rekurzivní funkce

8.3.4 Vyvolání funkcí

Během ladění je možné zobrazit seznam funkcí, které byly vyvolány, ale které se ještě nevrátily (tedy všech funkcí, ve kterých dosud nebyla provedena instrukce FRET). Bude zobrazeno dialogové okno, v němž je uveden seznam vyvolání funkcí, které vedly až k aktuální instrukci. Na začátku seznamu se nachází poslední provedená funkce. Na začátku seznamu se nachází poslední provedená funkce.

Za účelem pozorování chování vyvolání funkce:

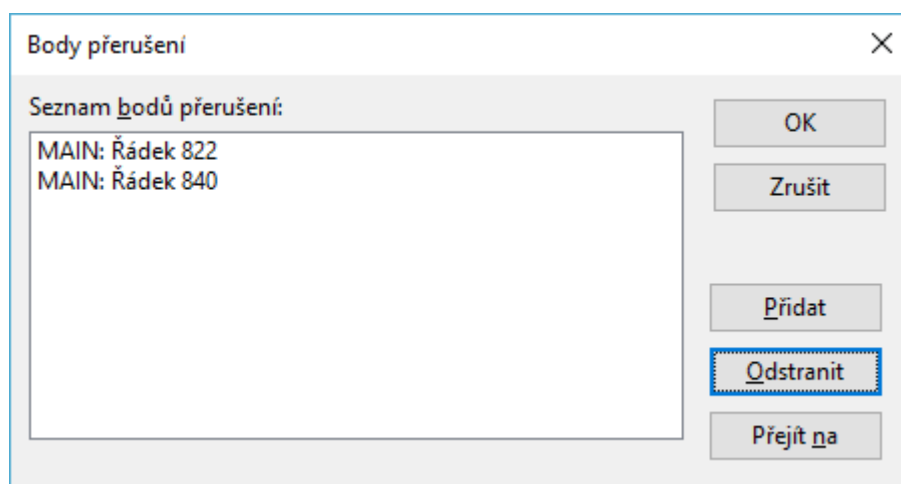
- Přesuňte kurzor do požadované polohy uvnitř funkce.
- zvolte **Odlad'ování->Provést až po kurzor** za účelem provedení programu do požadované polohy.
- zvolte **Odlad'ování->Vyvolání funkcí** nebo tlačítko rychlé volby Ctrl+K.
- v dialogovém okně Volání je možné zvolit název funkce. Kurzor se přesune na první instrukci zvolené funkce.

8.3.5 Body přerušení

Bod přerušení umožňuje podrobně si prohlédnout posloupnost provádění instrukcí, zhodnotit nebo změnit proměnné a zařízení, prozkoumat seznam volání funkcí apod.

Provádění úkolu se zablokuje po dosažení instrukce, do které byl vložen bod přerušení.

Body přerušení mohou být nastaveny předtím, než bude zahájeno provádění některého úkolu, i během jeho provádění (menu **Odlad'ování->Body přerušení**). Je možné také odstranit body přerušení poté, co již nebudou potřebné.



Seznam bodů přerušení

Existují situace, při kterých nedojde k zablokování úkolu, navzdory tomu, že byly vloženy body přerušení, protože provádění nikdy nedosáhne místa přerušení. V těchto případech lze zablokovat úkol příkazem **Odlad'ování->Přerušit nyní**. Kurzor bude přemístěn na instrukci GPL, která měla být provedena v okamžiku přerušení.

8.3.6 Obsah proměnné

Příkaz lze zvolit v menu **Odlad'ování->Obsah proměnné**.

Po přerušení provádění úkolu je možné zobrazit:

- Hodnota místních proměnných deklarovaných ve funkci, ve které je úkol zastaven
- globální proměnné
- hodnota přijatá vztahem
- stav zařízení a parametrů druhu zařízení.

Zobrazení/Změna obsahu proměnné

Když proměnná (nebo zařízení) není typu samotného čtení, její obsah lze měnit: Změna samozřejmě ovlivní následující průběh úkolu.

Změna hodnoty proměnné nebo zařízení umožňuje vyzkoušet provádění v podmínkách odlišných od běžného provádění, korigovat zjištěné chyby a pokračovat prováděním následujících instrukcí.

Je možné zobrazit obsah proměnné, zařízení nebo konstanty i přesunutím myši na proměnnou, na název zařízení nebo na konstantu. Dojde k zobrazení popisu tlačítka, ve kterém je uveden druh, název a hodnota údaje. V případě volby vztahu dojde k zobrazení výsledku. Když se kurzor myši nachází uvnitř volby, bude použita celá volba; v opačném případě bude použito pouze slovo, na kterém se nachází kurzor myši. Když se kurzor myši nenachází uvnitř slova, použije se celý argument.

Například pro zobrazení hodnoty matrice $Mx[3][\text{slopec}]$ v případě, že se kurzor myši nachází nad „3“, bude v popisu tlačítka zobrazeno 3. Když se kurzor nachází na „slopec“, bude zobrazena hodnota sloupce. A když se kurzor nachází na „matrice“, nebude zobrazeno nic. Když se kurzor nachází na hranaté závorce, bude zobrazena hodnota $Mx[3][\text{slopec}]$.

8.3.7 Seznam dostupných tlačítek rychlé volby

Pro aktivaci příkazů **Odlad'ování** je možné zvolit položky menu **Odlad'ování** nebo působit přímo z klávesnice. Je třeba použít tato tlačítka:

Tlačítko	Popis
Ctrl+F5	slouží k otevření dialogového okna se seznamem prováděných úkolů
Ctrl+Shift+F5	slouží k otevření dialogového okna se seznamem všech úkolů
Ctrl+B	slouží k otevření dialogového okna pro vložení nebo vymazání bodů přerušení
Ctrl+F9	slouží k vložení nebo odstranění bodu přerušení v místě řádku, na kterém se nachází kurzor
Ctrl+K	slouží k otevření dialogového okna pro zobrazení seznamu funkcí, které byly vyvolány, ale které se dosud nevrátily
Shift+F9	slouží k otevření dialogového okna pro zobrazení obsahu proměnné
F8	slouží k provedení instrukce. Jedná-li se o funkci, bude uvedena do činnosti.
Shift+F7	slouží k provedení všech instrukcí funkce
F10	slouží k provedení instrukce. Jedná-li se o funkci, zajistí její provedení bez vstupu dovnitř.
F7	slouží k provedení všech instrukcí až po instrukci, na které se nachází kurzor. Kurzor musí být umístěn na instrukci uvnitř funkce
Alt+Pauza	slouží k přerušení provádění kódu na poslední provedené instrukci
F5	slouží k obnovení provádění kódu po posledním přerušení
Shift+F5	slouží k ukončení aktuálního úkolu a k jeho opětovnému provedení
Alt+F5	slouží k ukončení ladění

8.4 Inicializace řízení

8.4.1 Sítová připojení

Činnost Albatros ve stroji je chráněna přítomností hardwarového USB klíče, nakonfigurovaného firmou T.P.A. Příkaz lze zvolit v menu **Cnc-> Sítová připojení**. Slouží k zobrazení stavu vzdálených modulů připojených k systému. V případě, že by daný modul nebyl připojen, bude symbol, kterým je znázorněn, označen červeným X. Pro každý modul jsou uvedena dvě pole. První pole představuje název modulu a druhé pole název stanice sítě. Obvykle má toto poslední uvedené pole první znaky s neměnným názvem „TPA“, po kterém následuje výrobní číslo vzdáleného modulu.

Přiřazení síťového uzlu (vzdálený modul) logickému modulu

Síťový uzel je možné přiřadit zvolenému modulu volbou nápisu „Nekonfigurováno“ kurzorem myši nebo volbou tlačítka **[Změnit]**. Po několika sekundách se objeví okno se seznamem vzdálených modulů dostupných v síti (každý vzdálený modul musí být zapnutý a musí již mít správně získanou IP adresu).

Nyní je možné zvolit síťový uzel, který má být připojen k logickému modulu, a potvrdit volbu stisknutím tlačítka



Všimněte si, že tato operace může být provedena s úrovní hesla „Servisní služba“, bez potřeby přístupu do konfigurace Systému Albatros, která vyžaduje úroveň hesla „Výrobce“.

Aktualizace softwaru vzdáleného modulu

Je možné provést kompletní aktualizaci řídicího softwaru, který se nachází ve vnitřní paměti vzdáleného modulu, a to volbou tlačítka **[Aktualizovat]**. Aby Albatros provedl aktualizaci, je třeba zvolit vzdálený modul, který si přejete aktualizovat. Tento modul musí být připojen a musí komunikovat se softwarem Albatros.

8.4.2 Diagnostika hardwaru

Příkaz lze zvolit v menu **Cnc->Diagnostika hardwaru**.

Diagnostika hardwaru zobrazuje seznam a stav konfigurovaných modulů, karet a uzlů, které k nim patří, v souladu s definováním v konfiguraci hardwaru.

Když je symbol karty nebo uzlu označen červeným X, znamená to, že daná karta nebyla identifikována hardwarem, který je součástí řízení, nebo že nebylo možné správně provést její inicializaci. Je-li označena žlutým otazníkem, znamená to, že systém zaznamenal přítomnost karty nebo uzlu, ale že se neshoduje s druhem definovaným v konfiguraci.

Topologie sítě EtherCAT

V diagnostickém okně hardwaru, když je v *stromu* zvolen uzel sítě EtherCAT, dojde k aktivaci tlačítka **[Detaily]**, které zas aktivuje zobrazování topologie sítě EtherCAT v grafickém formátu.

V tomto grafu jsou zobrazeny informace uzlů, které jsou fyzicky přítomné na síti: stav každého uzlu, stav jednotek pohybu v osách, když je uzlem servopohon, a kvalita komunikace. Každý uzel a každá jednotka pohybu v ose je znázorněn jako obdélník, jehož barva určuje stav.

Při přechodu kurzorem myši po obdélníku se zobrazí hlášení s pomocným textem (tooltip), které popisuje stav a chyby komunikace uzlu a stav jednotky pohybu v ose.

Zobrazování a změna objektů v uzlech

V diagnostickém okně hardwaru, když je v *stromu* zvolen uzel sítě EtherCAT, dojde k aktivaci tlačítka **[Slovník objektu]**, které slouží k zobrazení a změně objektů uzlu. Změna dat je možná pouze na úrovni **Výrobce**.

Objekty jsou seskupeny na základě jejich adresy:

Počáteční adresa	Koncová adresa	Název plochy
------------------	----------------	--------------

0x0000	0x0FFF	Data type area
0x1000	0x1FFF	Communication area
0x2000	0x5FFF	Manufacture specific area
0x6000	0x6FFF	Input area
0x7000	0x7FFF	Output area
0x8000	0x8FFF	Configuration area
0x9000	0x9FFF	Information area
0xA000	0xAFFF	Diagnosis area
0xB000	0xBFFF	Service transfer area
0xC000	0xEFFF	Reserved area
0xF000	0xFFFF	Device area

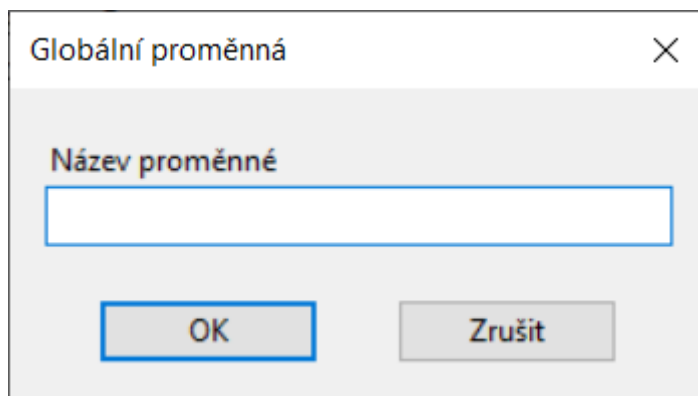
Celé číselné hodnoty jsou zobrazovány v desítkovém i v šestnáctkovém základu. Při změně jejich hodnoty může být číslo zadáno i v šestnáctkovém formátu, použitím označení \$...H, a v binárním, s \$...B. (jak se to děje v GPL).

8.5 Zkouška

8.5.1 Uložit do paměti globální proměnnou

Příkaz lze zvolit v menu **Zkouška->Uložit do paměti globální proměnnou**.

Slouží k uložení globální proměnné do paměti na disk v podobě formátovaného textového souboru. Soubor má název tvořený *názevproměnné.txt* a bude uložen do adresáře *Report*. Operace je možná pouze v případě, že je úroveň přístupu se čtením globální proměnné kompatibilní s aktuální úrovní přístupu.



Uložení globální proměnné do paměti

8.5.2 Provést funkci

Příkaz lze zvolit v menu **Zkouška->Provést funkci**.

Slouží k provedení funkce způsobem nezávislým na zbytku systému. Dojde tedy k vytvoření úkolu, který začíná své provádění ze zvolené funkce a bude z ní mít odvozený název. Spustitelné funkce jsou pouze ty, které nemají parametry na vstupu a jejichž úroveň přístupu při čtení je kompatibilní s aktuální úrovní přístupu. Když je prováděná funkce hlavní funkcí mezikupiny, poté budou provedeny také všechny úkoly automatického spuštění (Autorun).

8.5.3 Importování hlášení

Hlášení skupiny přiřazená prostřednictvím instrukce GPL [DEFMSG](#) jsou archivována ve formátu xmlng. Hlášení, která se nacházejí v souboru, mohou být změněna, přidána nebo odstraněna. Pro zobrazení změn v kódu GPL je třeba použít příkaz **Zkouška->Importovat hlášení skupiny**, který je aktivován pouze když nejsou okna otevírána.

Za účelem provedení importování musí být veškerý kód GPL podroben kompilaci bez výskytu chyb. V opačném případě bude uživatel upozorněn hlášením „Ne celý kód GPL byl podroben kompilaci“. Nebudou importována hlášení skupiny patřící do [šifrovaných](#) souborů (viz kapitola **Vývojové nástroje->Editor GPL->Šifrování**), pro které uživatel nedovolil jejich průhledné zobrazování.

Budou importována pouze hlášení, která již byla definována v kódu GPL. Změna textu GPL nebude provedena, když je přítomna nejméně jedna DEFMSG, která následuje po instrukci IFDEF.

Během importování mohou být zaznamenány chyby, když:

- Je v textech některého hlášení skupiny identifikátor jazyka přítomný víckrát než jednou
- kterýkoli text je prázdný (tedy: „“)
- název skupiny nebo knihovny je definován víckrát než jednou.

Po skončení importování bude provedena kompilace všech modulů, ve kterých byly provedeny změny skupin nebo knihoven.

Všechny níže uvedené jazyky mohou být použity v souboru XMLNG

"AFK" Afrikánština
 "ARA" Arabština
 "AZE" Azerština
 "BAS" Baškirština
 "BEL" Běloruština
 "BGR" Bulharština
 "BSB" Bosenština (latinka)
 "BSC" Bosenština (azbuka)
 "BRE" Bretonština
 "CAT" Katalánština
 "CHS" Zjednodušená čínština
 "CHT" Standardní čínština
 "COS" Korsičtina
 "CSY" Čeština
 "CYM" Gaelština
 "DAN" Dánština
 "DEA" Němčina (Rakousko)
 "DEU" Němčina (Německo)
 "ELL" Řečtina

"ENG" Angličtina
"ENU" Angličtina (Spojené státy)
"ESP" Španělština
"ETI" Estonština
"EUQ" Baskičtina
"FAR" Perština
"FIN" Finština
"FRA" Francouzština
"FPO" Filipínština
"FRB" Francouzština (Belgie)
"FYN" Fríština
"GLC" Galicijština
"HAU" Hauština
"HEB" Hebrejština
"HRB" Chorvatština (Bosna-Hercegovina)
"HRV" Chorvatština (Chorvatsko)
"HUN" Maďarština
"IBO" Igboština
"IND" Indonéština
"IRE" Irština
"ISL" Islandština
"ITA" Italština
"JPN" Japonština
"KAL" Grónština
"KOR" Korejština
"SAH" Jakutština
"KYR" Kyrgyzština
"LVI" Lotyština
"LTH" Litevština
"LBX" Lucemburština
"MNN" Mongolština
"NON" Norština Nynorsk
"NOR" Norština Bokmal
"NLB" Nizozemština (Belgie)
"NLD" Nizozemština (Holandsko)
"OCI" Okcitánština
"PLK" Polština
"PTB" Portugalština (Brazílie)
"PTG" Portugalština (Portugalsko)
"RMC" Rétorománština
"ROM" Rumunština
"RUS" Ruština
"SKY" Slovenština
"SLV" Slovinština
"SQI" Albánština
"SRM" Srbština (latinka, Srbsko)
"SRN" Srbština (azbuka, Bosna-Hercegovina)
"SRO" Srbština (azbuka, Srbsko)
"SRP" Srbština (latinka, Montenegro)
"SRQ" Srbština (azbuka, Montenegro)
"SRS" Srbština (latinka, Bosna-Hercegovina)
"SVE" Švédština
"TAJ" Tádžičtina
"TRK" Turečtina
"TTT" Tatarština
"TUK" Turkmenština
"UKR" Ukrajínština
"URD" Urdština
"UZB" Uzbečtina
"VIT" Vietnamština
"WOL" Wolofčina
"XHO" Xhoština
"YOR" Jorubština
"ZUL" Zuluština

8.5.4 Poznámka uživatele do souboru sestavy alarmů

Prostřednictvím příkazu **Zadat poznámku** v nabídce **Zkouška** je možné vložit signalizaci do souboru sestavy alarmů aktuálního měsíce (MONTHxx.TER). Položka nabídky je aktivována se stejnou nebo vyšší úrovní hesla než je heslo servisní služby.

8.6 Nástroje

8.6.1 Přizpůsobit...

Příkaz lze zvolit v menu **Nástroje->Přizpůsobit...**

Umožňuje nastavení maximálně 20 programů, jejichž provádění bude muset být zahájeno z menu **Nástroje** softwaru Albatros.

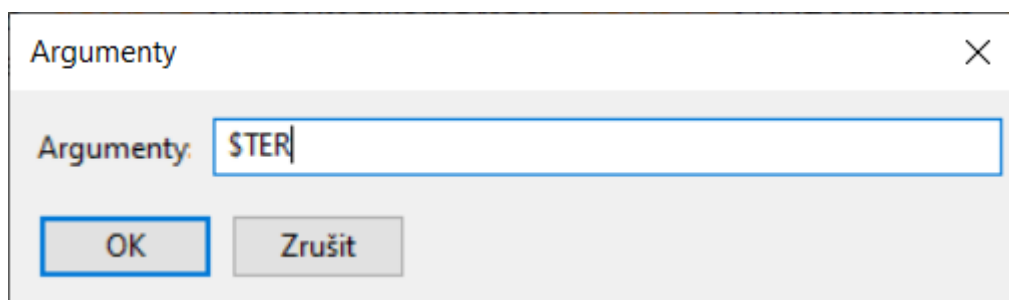
Konfigurace menu Nástroje

Struktura menu:
Příkaz:

Je zde uveden seznam programů zobrazených v menu **Nástroje**.
Název programu, který musí být proveden. V rámci volitelných funkcí může být definován také adresář, ve kterém je archivován program v případě, že by byl odlišný od adresáře, ze kterého byl spuštěn Albatros ze složky, ve kterém operační systém hledá spustitelné soubory (windows proměnné prostředí PATH).

Text v menu:	Jedná se o název, který má být zobrazen v menu Nástroje za účelem identifikace spustitelného programu.
Argumenty:	Jakákoli kombinace argumentů příkazového řádku, který program potřebuje ke svému správnému spuštění. Je možné vložit dynamické argumenty. \$Ter Například použitím řetězce \$TER v provedení ViewRER se otevře soubor protokolu aktuálního měsíce.
	Následuje seznam argumentů:
\$File	Kompletní cesta aktuálního souboru.
\$FileName	Název souboru a přípony aktuálního souboru.
\$FileDir	Disk a adresář aktuálního souboru.
\$Ter	Kompletní cesta souboru protokolu chyb aktuálního měsíce.
\$DirModule	Disk a adresář obsahující MODx aktuálního souboru.
\$Module	Název modulu aktuálního souboru.
\$Bin	Disk a adresář obsahující spustitelné soubory softwaru Albatros.
\$TpaIni	Kompletní cesta inicializačního souboru tpa.ini
\$ReqDirModule	Cesta (jednotku a složku) ve formě Albatros. Pokud máte více Moduly se otevře okno pro výběr modulu.
\$ReqModule	Modul počet albatrosa. Pokud máte více modulů je otevřen Výběr okno číslo modulu.
\$ReqFile	Název souboru. Otevře se okno s požadavkem na název souboru. Vybraný soubor bude předán mezi tématy programu, která budou provedena.

Žádost o argumenty: V případě této volby bude při každém spuštění programu zobrazeno dialogové okno pro zadání argumentů odlišných od těch, které byly nastaveny prostřednictvím položky Argumenty, a které lze měnit v závislosti na způsobu spuštění programu.



Slouží k vymezení argumentů spuštění programu

Úroveň aktivace: Slouží k nastavení zobrazování programu v menu **Nástroje**. Programům pro zkoušení nebo změnu dat softwaru Albatros se obvykle přiřazuje úroveň výrobce. Programům pro editování obrábění prováděných na stroji se přiřazuje úroveň aktivace uživatele.

Editování některých polí může proběhnout také prostřednictvím použití tlačítka **[Přidat]**. Toto tlačítko otevře dialogové okno **Přidat nástroj**, které slouží k volbě programu, který má být spuštěn. K druhým povolených spustitelných souborů patří .EXE, .COM, .PIF, .BAT.

Při zavření dialogového okna s potvrzením dat je program vložen do okna **Struktura menu** a název programu a adresáře, do kterého patří, do řádku **Příkaz**.

K ostatním přítomným tlačítkům patří **[Odstranit]**, **[Přemístit nahoru]**, **[Přemístit dolů]** a tato tlačítka se používají pro vymazání programu a pro seřazení seznamu programů.

8.7 Prohlížeč

8.7.1 Prohlížeč

Prohlížeč představuje funkci softwaru Albatros, která používá informace vytvořené kompilátorem pro vytvoření databáze pro rychlé hledání symbolů definovaných ve funkcích.

Tato funkce je aktivní pouze v případě úrovně hesla, která je stejná jako úroveň výrobce nebo ji převyšuje.

Použitelné příkazy lze zvolit v menu **Ladění**.

S prohlížečem je možné:

- Umístit kurzor na řádek, ve kterém je poprvé definována funkce, proměnná nebo konstanta modulu, skupiny nebo knihovny. (v menu **Ladění->Přejít na definici**)
- Umístíte kurzor na řádky, ve kterých se nacházejí reference funkce, zařízení, proměnné modulu nebo skupiny, instrukce GPL (s výjimkou instrukcí FCALL a FRET). (v menu **Ladění->Přejít na odkaz** za účelem zobrazení předchozího nebo následujícího odkazu zvolte v menu položky **Ladění->Předchozí** nebo **Ladění->Následující**)

Proměnné skupiny jsou spravovány v okně editování skupiny, do které patří.

Za účelem aktualizace prohlížeče při přechodu na novou verzi se doporučuje jako první operaci provést uložení globálních proměnných a poté provést příkaz **Soubor->Kompilovat Vše**. Během fáze editování funkcí se ztratí souvislost mezi textem a vyhledávanými symboly a bude obnovena teprve ve fázi archivování.

8.7.2 Vyhledat Identifikátor
















Příkaz lze zvolit v menu **Odladování->Vyhledat Identifikátor**. Slouží k otevření dialogového okna pro zadání názvu symbolu, který má být vyhledán v kódu GPL. Na základě nastaveného **Druhu vyhledání** je identifikována definice nebo první odkaz symbolu.

Zadaný název může mít níže uvedené charakteristiky:

- Neobsahuje žádný znak '.' (tečka): Bude vyhledán ve všech souborech funkcí
- obsahuje pouze jeden znak '.' (tečka): Název, který předchází tečce, bude považován za název skupiny a symbol bude vyhledán pouze v této skupině. Když jsou například definované dvě funkce s názvem VisError, jedna ve skupině MAIN a druhá ve skupině JEDNOTKY POHYBU V OSÁCH (JPVO), při napsání identifikátoru JPVO.VisError se kurzor přesune na první řádek funkce VisError skupiny JPVO.
- obsahuje dva znaky '.' (tečka): Název, který předchází první tečce, je považován za název skupiny a název, který předchází druhé tečce, je považován za název podskupiny; symbol bude vyhledán v této podskupině.
- když končí znakem '*' (hvězdička), budou se brát v úvahu všechny symboly, které začínají znaky, které předchází hvězdičce.

V případě, že se vyskytne nejednoznačnost ve vyhledání symbolu, bude otevřeno dialogové okno, v němž budou zobrazeny všechny symboly s požadovaným názvem. V tomto okně bude možné zvolit požadovaný symbol.

Níže je uveden popis významu speciálních symbolů použitých v seznamu pro volbu identifikátoru.

Symbol	Popis
	instrukce GPL
	konstanta modulu nebo skupiny nebo knihovny
	proměnná modulu nebo skupiny
	proměnná knihovny
	vektor knihovny
	matrice knihovny
	funkce knihovny
	hlášení skupiny
	návěští
	místní proměnná
	místní vektor
	místní matrice
	jednoduchý parametr
	parametr vektor
	parametr matrice

8.7.3 Seznam dostupných tlačítek rychlé volby

Pro aktivaci příkazů Prohlížeče je možné zvolit položky menu **Ladění** nebo působit přímo z klávesnice. Je třeba použít tato tlačítka:

Tlačítko	Popis
F2	slouží k přemístění kurzoru na řádek, ve kterém je definován zvolený symbol. Když je v databázi prohlížeče definováno více symbolů se zvoleným názvem, dojde k otevření dialogového okna, ve kterém bude uživateli poskytnuta možnost volby požadovaného symbolu.
Shift+F2	slouží k přemístění kurzoru na první odkaz zvoleného symbolu. V případě nejednoznačnosti dojde k otevření dialogového okna pro volbu zvoleného symbolu.
Ctrl+F2	slouží k otevření dialogového okna pro volbu zvoleného symbolu
Ctrl+'+' nebo Ctrl+PgUp	slouží k přemístění kurzoru na následující odkaz (použijte '+' číselné klávesnice)
Ctrl+'-' nebo Ctrl+PgDown	slouží k přemístění kurzoru na předchozí odkaz (použijte '-' číselné klávesnice)

9 Doplňkové programy

9.1 XConfMerge: program pro sloučení konfiguračního souboru

XConfMerge je nástroj, který provádí sloučení konfiguračního souboru. Spouští se z příkazového řádku ze složky bin, protože XConfMerge čte soubor tpa.ini.

Soubory, přečtené z XConfMerge jsou:

- hardware.xconf: obsahuje data virtuálně-fyzického a konfiguraci hardwaru (fyzické)
- devices.xconf: obsahuje data konfigurace skupin, podskupin a zařízení (logických)
- devices.xmlng: obsahuje přeložitelná hlášení. V každém případě jsou zohledněny všechny soubory jazyka, přítomné ve složce
- addresses.xdb: obsahuje logické adresy zařízení.

K argumentům, které je třeba předat, patří:

- složka, ze které mají být přečteny nové soubory;
- číslo modulu, na který se vztahuje uživatelsky přízpůsobená funkce. Když není uvedeno žádné číslo, bude jako přednastavená hodnota považována 0.

Cesta, ze které lze číst soubor určený k aktualizaci, která je stejná jako ta, na kterou se zapisuje soubor získaný postupem sloučení, je odvozena z dat, nastavených v tpa.ini.

Upozornění: soubory, které jsou určeny k aktualizaci, jsou přepsány, a nebude provedeno žádné automatické zálohování.

Pravidla pro sloučení konfiguračních souborů skupin jsou:

1. Když v obou souborech existuje stejná skupina, podskupina nebo zařízení, zachovávají se data a aktivace starého souboru.
2. Když skupina, podskupina nebo zařízení existuje pouze v novém souboru, data a aktivace budou zkopírována.
3. Když skupina, podskupina nebo zařízení existuje pouze ve starém souboru, tento bude odstraněn.

Pravidla pro sloučení konfiguračního souboru hardwaru a virtuálně-fyzického:

1. Když v obou souborech existuje stejný hardware, zachová se hardware a virtuálně-fyzické nového souboru s aktivacemi, určenými ve starém souboru.
2. Když hardware existuje pouze v novém souboru, zachová se nový hardware se svou aktivací a nové virtuálně-fyzické.
3. Když hardware existuje pouze ve starém souboru, bude odstraněn spolu s virtuálně fyzickým.

Pravidla pro sloučení souboru logických adres:

1. Když se soubor nachází ve složce souborů, určených k importu, budou zachovány logické adresy uživatelsky přízpůsobené funkce a budou přiřazeny nové adresy novým zařízením, jsou-li nějaké součástí.
2. Když se soubor nachází ve složce souborů, určených k importu, logické adresy, které se v ní nacházejí, budou přečteny a použity.

Pravidla pro sloučení souborů hlášení:

1. Nebude provedeno žádné sloučení, ale nový soubor bude zkopírován do složky modulu.

Po ukončení provádění nástroj pro sloučení vrátí následující hodnoty:

0	všechno OK;
---	-------------

1	sloučení souborů nebylo ukončeno úspěšně;
2	nebyly definovány argumenty;
3	číslo modulu, které bylo předáno jako argument, je chybné;
4	složka, uvedena jako první argument, neexistuje.

9.2 XParMerge: program pro sloučení dvou souborů parametrizování

XParMerge je nástroj, který provádí sloučení dvou souborů parametrizování.

Spouští se z příkazového řádku ze složky bin, protože XParMerge čte soubor tpa.ini. K argumentům, které je třeba předat, patří: název nového souboru a číslo modulu, na který se vztahuje.

Cesta, ze které lze číst soubor určený k aktualizaci, a ta, na kterou se zapisuje soubor získaný postupem sloučení, je odvozena z dat, nastavených v tpa.ini.

Upozornění: nebude provedeno žádné automatické zálohování starého souboru parametrizování, ale bude přepsán.

Pravidla pro sloučení souborů technologického parametrizování:

- 1) Když v obou souborech existuje stejná kontrola, bude podržena hodnota starého souboru, ale aktualizují se ostatní parametry, které ji definují (zrušený, viditelný, název proměnné GPL...), po jejich převzetí z nového souboru;
- 2) Když je v novém souboru definována kontrola, která neexistuje ve starém souboru, kontrola bude podržena;
- 3) Když je ve starém souboru definována kontrola, která neexistuje v novém souboru, kontrola bude odstraněna.

Pravidla pro sloučení souborů technologického parametrizování:

- 1) Když je související funkce dialog nového souboru odlišná od související funkce dialog nového souboru, když jsou součástí, budou aktualizovány ty starého souboru (odebrány nebo přidány nové kontroly);
- 2) Když je související funkce dialog nového souboru stejná jako související funkce dialog starého souboru, funkce dialog nového se přidají k funkcím dialog starého;
- 3) Když je související funkce dialog přítomna pouze ve starém souboru, bude odstraněna spolu s jejími funkcemi dialog.

10 Jazyk GPL

10.1 Základní principy

10.1.1 Úvod a jazyk GPL

Jazyk GPL (General Purpose Language - Jazyk pro všeobecné účely) je jazyk používaný pro vytváření funkcí systému Albatros.

Jeho struktura je velmi podobná jazyku BASIC, ale je charakterizována přítomností mnoha specifických instrukcí pro řízení zařízení.

Jazyk je tvořen více než 200 instrukcemi nazvanými *instrukce*, které jsou kvůli výhodnosti rozděleny do více skupin tvořených instrukcemi se souvisejícími funkcemi.

Jazyk je dále [multitaskový](#), tj. umožňuje současné provádění více úkolů.

Typická syntaxe instrukce GPL

Instrukce GPL mají všechny podobnou strukturu, která je zachycena na níže uvedeném schématu:

nazevinstrukce parametr-1, parametr-2, parametr-N

Počet skutečně přítomných parametrů bude záviset na samotné instrukci a na kontextu, ve kterém je použita; obecně platí, že maximální přípustný počet parametrů pro instrukci nebo funkci GPL je 120. V některých případech se může stát, že parametry nebudou přítomné.

Minimálním blokem kódu GPL je [funkce](#).

Rozdělení kódu do skupin

Kód GPL se rozděluje do bloků, které zrcadlí logické rozdělení stroje do skupin. Pro každou skupinu proto budeme mít soubor obsahující jemu přiřazený kód. K souborům obsahujícím kód skupin, přítomným ve stroji, se přidružuje soubor globálních proměnných a konstant, které jsou patrné z kódu GPL všech skupin, a [knihovny](#). Ty obsahují kód, který není vázán na konfiguraci stroje, a proto jej lze snadno přenést z jednoho stroje do druhého.

10.1.2 Konvence a terminologie

Hlavní přijaté výrazy

ARGUMENT	Jedná se o jeden z argumentů instrukce; v závislosti na druhu instrukce může být definován jako <i>konstanta</i> , <i>proměnná</i> nebo jako <i>parametr</i> , v závislosti na druhu instrukce; nachází-li se v hranatých závorkách ([]), znamená to, že může být vynechán, což má samozřejmě za následek odlišný způsob provedení instrukce
KLÍČOVÉ SLOVO	Jedná se o argument, který se volí mezi argumenty s přednastavenou hodnotou, a obvykle je psán velkými písmeny; seznam klíčových slov je dodán na specifické straně nápovědy
PARAMETR	Jedná se o argument instrukce nedefinovaný uvnitř samotné instrukce, který je stejně jako parametr předán funkci při jejím provedení; někdy se také označuje výrazem <i>parametrizovaný argument</i>
KONSTANTA	jedná se o argument definovaný neměnným způsobem prostřednictvím metapříkazu CONST nebo argument, který je stanoven neměnným způsobem uvnitř instrukce
PROMĚNNÁ	Jedná se o argument, který byl definován jako globální proměnná modulu nebo skupiny, prostřednictvím instrukce LOCAL, a může být zorganizován jako jednoduchá proměnná, vektor nebo matrice. Viz Proměnné
PARAMETR KONFIGURACE	Jedná se o argument, který byl definován v konfiguraci, jako například parametry jednotky pohybu v ose

Argumenty, které se nejvíc vyskytují v popisu instrukcí

Níže jsou uvedeny výrazy vztahující se na argumenty, které se často používají v syntaxi instrukcí GPL, přičemž za každým je uveden příslušný popis. V případech, kdy argument může nabýt hodnoty odlišné od níže popsanych hodnot, bude jeho popis uveden v části *Argumenty* na straně věnované nápovědě k instrukci.

nazevvstupu	název zařízení digitálního vstupu
nazevvystupu	název zařízení digitálního výstupu
nazevpriznaku	název zařízení přepínače příznaku nebo bitu příznaku
nazevportu	název zařízení portu vstupu, portu výstupu nebo portu příznaku
nazevcasovace	název zařízení časovače
nazevpocitadla	název zařízení počítačidla
nazevfunkce	název funkce (platí také jako parametr zařízení v případě ERRSYS.)
nazevprogramu	název podprogramu, odpovídá <i>návěstí</i> , ze kterého je třeba vycházet kvůli vysvětlení; vyvolání podprogramu se provádí prostřednictvím instrukce "CALL nazevpodprogramu".
jednotkapohybuose	název jednotky pohybu v ose
konstanta	znak, celé číslo, double nebo klíčové slovo
hodnota	konstanta nebo proměnná (<i>typ závisí na instrukci</i>)
promenna	název proměnné, prvek vektoru nebo prvek matrice
promennazarizeni	název <i>parametru zařízení</i>
matrice	název matrice
vektor	název vektoru
navesti	název návěstí skoku nebo název podprogramu.
stav	logický stav, který může být ZAP. (ON) nebo VYP. (OFF), nebo 1 nebo 0
casovylimit	časový limit, do kterého musí dojít k určité události, nebo doba opoždění (konstanta nebo proměnná)
poloha	hodnota polohy (konstanta double nebo proměnná double)
polomer	hodnota poloměru (konstanta double nebo proměnná double)
uhel	hodnota úhlu (konstanta double nebo proměnná double)
pocetotacek	počet otáček (konstanta double nebo proměnná double)
rychlost	hodnota rychlosti (konstanta float nebo proměnná float)
smer	otáčení ve směru nebo proti směru hodinových ručiček (proměnná nebo konstanta: CW nebo CCW)
operand	(konstanta nebo proměnná nebo nazevzarizeni)
vysledek	výsledek operace (proměnná nebo nazevzarizeni)
nazevzarizeni	název typu jakéhokoli zařízení (nebo parametr zařízení)
konstantestr	posloupnost znaků uzavřena uvozovkami (např. "string")
promennaestr	název vektoru znaků, tj. řetězec
operator	operátory porovnání, také ve vzájemné kombinaci: > (větší) = (rovný) < (menší) mohou být kombinovány svým vzájemným přiblížením, např. >= (tj. větší nebo rovný)
typ	typ konstanty nebo proměnné: "char" (8 bitů), "integer" (32 bitů), "float" (32 bitů), "double" (64 bitů), "string"
parametr zařízení	jedná se o proměnnou, která zastupuje zařízení. Zařízení jsou definována v Konfiguraci.

Hlavní výrazy používané pro jednotku pohybu v ose

teoretická (nebo cílová)	Aktuální „teoretická“ hodnota vnucovaná číslicovým řízením na základě algoritmu vytváření profilu rychlosti.
hodnota	
reálná poloha	Skutečná poloha jednotky pohybu v ose tak, jak je zaznamenána snímačem polohy. Liší se od teoretické polohy o hodnotu nazvanou „chyba sledování“ nebo „chyba smyčky“.
výsledná poloha	Odpovídá naprogramované cílové poloze pohybu. Algoritmus výpočtu profilu rychlosti zajistí, že teoretická poloha přesně dosáhne výsledné polohy.
okno dosažení polohy	Programovatelný interval, který má jako centrální bod výslednou teoretickou polohu: Když se reálná poloha dostane dovnitř tohoto intervalu, skutečný pohyb se považuje za ukončený.
velké okno dosažení polohy	a okno dosažení polohy vynásobené faktorem nastavitelným prostřednictvím instrukce SETBIGWINFACTOR .

chyba sledování	Rozdíl, okamžik po okamžiku, mezi teoretickou a reálnou polohou jednotky pohybu v ose: Obvykle je přímo úměrný rychlosti přesunu a nepřímo úměrný „zisku proporcionální smyčky“.
proporcionální zisk [smyčky]	Programovatelný parametr regulace jednotky pohybu v ose: Určuje poměr mezi aktuální rychlostí a příslušnou chybou sledování.
feed forward	Programovatelný parametr regulace jednotky pohybu v ose: Určuje přímý příspěvek (proporcionální programované rychlosti) aplikovaný na ovládání rychlosti pohonu (dopředné řízení). Umožňuje snížit hodnotu chyby sledování při zachování rychlosti a proporcionálního zisku.
feed rate override	Procentuální podíl z naprogramované rychlosti (překrytí rychlosti dopředného řízení). Tento parametr umožňuje snížit rychlost provádění vůči naprogramované rychlosti v procentuálním poměru, který se může měnit v rozsahu od 0% do 100%.
tolerance	Hodnota přesunu, u které se jednotka pohybu v ose liší od původní dráhy pohybu v interpolaci více jednotek pohybu v osách na rozhraní dvou po sobě následujících bloků přesunu.
mechanická vůle	Prostor mezi drážkou a zubem ve dvojici ozubených kol

10.1.3 Proměnné

Proměnné jsou schránky na informace, které jsou používány jazykem GPL k uložení hodnot potřebných pro průběh programu.

Proměnné jsou charakterizované „druhem“, který odzrcadluje vlastnosti informace, která do nich bude ukládána. Každé proměnné je dále přiřazena viditelnost definovaná jako celky nebo podcelky kódu, které na ni mohou působit (provádět čtení a zápis).

Druhy dat

JEDNODUCHÁ nebo SKALÁRNÍ DATA

GPL podporuje jednoduché a sdružené druhy dat. Druhy jednoduchých dat jsou podobné těm, které jsou k dispozici ve většině programovacích jazyků:

Char

Jedná se o celé číslo se znaménkem v intervalu [-128 ; +127] a jeho délka je 1 byte.

Deklarování proměnné druhu Char je prováděno prostřednictvím níže uvedené syntaxe:

```
NazevPromenne as char
```

Integer

Jedná se o celé číslo se znaménkem v intervalu [-2147483647 ; +2147483647] a jeho délka je 4 byte (odpovídá druhu long v C).

Deklarování proměnné druhu Integer se provádí prostřednictvím níže uvedené syntaxe:

```
NazevPromenne as integer
```

Hodnota s plovoucí desetinnou čárkou (float)

Jedná se o číslo s plovoucí desetinnou čárkou v intervalu [-3,402 E+38 ; +3,402 E+38] a jeho délka je 4 bajty (obvykle se používá pro vyjádření rychlosti).

Deklarování proměnné druhu Float se provádí prostřednictvím níže uvedené syntaxe:

```
NazevPromenne as float
```

Double Jedná se o číslo s plovoucí desetinnou čárkou v intervalu [-1,797 E+308 ; 1,797 E+308] a jeho délka je 8 bajtů (obvykle se používá pro vyjádření poloh).

Deklarování proměnné druhu Double se provádí prostřednictvím níže uvedené syntaxe:

```
NazevPromenne as double
```

Tyto druhy dat mohou být použity současně ve stejném výrazu. GPL provede automatickou konverzi, aniž by na to nějak upozorňoval. Proto je třeba věnovat pozornost možné ztrátě informace, ke které by mohlo dojít při použití různých druhů dat ve stejném výrazu.

V některých situacích konverze není povolena. V těchto případech jsou obvykle výsledkem upozornění ze strany kompilátoru nebo chyby systému.

SDRUŽENÁ DATA

Pole (Array)

Jedná se o celek proměnných jednoduchého druhu, kdy všechny jsou stejného druhu, získaný přiřazením indexu k názvu proměnné. Index je uzavřen do hranatých závorek. Když se pole (array) nazývá „parametry“, první prvek celku bude označen „parametry[1]“, druhý „parametry[2]“ apod.

Pole (Array) má pevný počet prvků, který musí být definován při deklarování. Typická struktura deklarování pole se řídí níže uvedenou syntaxí:

```
parametry[10] as integer
```

Kde *parametry[10]* specifikuje, že název pole (array) je „parametry“ a že je tvořeno 10 prvky; *as integer* definuje jednoduchý druh data jednotlivých prvků pole, kterým je v tomto případě integer.

Pole (Array) mohou být tvořena jednoduchými daty nebo řetězci.

Jedno pole může být tvořeno maximálně 262144 prvky.

Vektory mohou být inicializovány přímo v kódu GPL v okamžiku deklarování. Syntaxe GPL může být:

```
[READONLY] vektor[pocetradku] as integer = 1,2,3,4
```

```
[READONLY] vektor[pocetradku] as string = „jeden“, „dva“, „tři“, „čtyři“
```

Matrice

Matrice jsou dvourozměrná pole (array), tedy proměnné, kterým jsou přiřazeny dva indexy. Matici můžeme přiřadit tabulce dat rozdělené na řádky a sloupce. Pro označení polí tabulky můžeme definovat, ve kterém řádku a ve kterém sloupci tabulky se nachází. První index odpovídá číslu řádku a druhý číslu sloupce.

Na rozdíl od polí (array) mohou matrice obsahovat vzájemně odlišné druhy dat pro každý sloupec, ale nemůžeme měnit druh uvnitř samotného sloupce.

Můžeme například definovat matici, ve které je první sloupec druhu integer a druhý druhu float. Nemůžeme však mít matici, ve které máme v prvním sloupci integer a float a ve druhém máme char a double. Pokud jde o druhy dat prvků, které tvoří řádky, musí být všechny řádky stejné.

Deklarování matrice může být provedeno podle níže uvedených syntaxí:

```
offset[10] as double double double
```

```
rozmery_dilu[50] as float:delka float:sirka float:tloustka
```

Při druhém druhu deklarování je přiřazen popisec nebo symbolický název každému sloupci. Symbolické názvy sloupců jsou užitečné při práci s maticemi velkých rozměrů. V těchto situacích je těžké pamatovat si význam veličin uložených v každém sloupci matrice. Symbolický název nám umožňuje okamžitě identifikovat druh dat, se kterým pracujeme. Např. „Posun[1][3]“ je méně jasný než „Posun[1].jednotkapohybuvose_X“. Matrice mohou obsahovat pouze data jednoduchého druhu. Nemohou se například vytvářet matrice, které obsahují řetězce. Matrice může být tvořena maximálně 262144 řádky.

Matrice mohou být inicializovány přímo v kódu GPL v okamžiku jejich deklarování. Syntaxe GPL může být:

```
[READONLY] nazevmatrice[cisloradku] as double double integer double = _
```

```
1.1, 2.2, 3, 0.1 _
```

```
1.2, 3.4, 5, 0.1 _
```

```
2.1, 5.6, 6, 0.1
```

Řetězce

Řetězce jsou soubory znaků, tedy daty typu char, které jsou však spravovány speciálním způsobem, protože představují čitelný text.

Řetězec je velmi podobný poli (array) prvků char. Největším rozdílem je přítomnost ukončovacího znaku, který je automaticky přidáván na konec řetězců. GPL poskytuje některé instrukce, které umožňují manipulovat řetězce.

Obvykle jsou řetězce použity pro zápis hlášení čitelných uživatelem na displej nebo do souboru záznamů.

Deklarování proměnné druhu Řetězec může být provedeno podle níže uvedených syntaxí:

```
NazevPromenne as String
```

```
NazevPromenne as String[20]
```

V případě prvního druhu deklarování je přiřazen přednastavený rozměr 256 bajtů. Ve druhém případě se definuje maximální rozměr řetězce.

Hodnoty řetězce jsou sekvence znaků, ohraničené uvozovkami. Příklad: „Stiskněte tlačítko“.

Pro zadání znaku "" (uvozovky) je třeba jej zadat dvakrát. Příklad: "Stiskněte tlačítko ""Start""".

Pro zadání znaků prostřednictvím číselného kódu se dovnitř řetězce zapíše znaky \u, po kterých následuje číselná hodnota šestnáctkového znaku. Příklad: \u20ac je symbol eura. Při napsání "\u20ac 15,6" je výsledkem € 15,6.

Konverze dat

Ve všech matematických výrazech, s výjimkou instrukce EXPR, jsou datové druhy operandů konvertovány na datový typ proměnné výsledku, a poté je provedena operace. Je důležité věnovat velkou pozornost deklarování datových druhů, protože mohou ovlivnit výsledek. Níže uvedená tabulka je příkladem možných změn výsledků na základě datového druhu:

dílků	Operand 1 (Integer)	Operand 2 (Double)	Výsledek (char)
3		5,0	0
5		1,9	5
1200		107,2	Nedefinován
1200		250,0	Nedefinován

dílků	Operand 1 (Double)	Operand 2 (Double)	Výsledek (Double)
3		5,0	0,6
5		1,9	2 631
1200		107,2	11 194
1200		250,0	4,8

V případě, že operandy nejsou stejného druhu, bude v instrukci EXPR provedena automatická konverze a druh výsledku operace je stejný jako ten z obou, který je větší, na základě níže uvedeného pravidla:

- char < integer
- float < double
- char nebo integer < float nebo double.

Po vyřešení výrazu je výsledek konvertován na druh proměnné výsledku.

EXPR	Operand 1 (Double)	+	Operand 2 (Integer)	/	Operand 3 (Float)	Výsledek (Integer)
	900,0	+	100	/	400,0	900

EXPR	Operand 1 (Double)	+	Operand 2 (Integer)	/	Operand 3 (Float)	Výsledek (Double)
	900,0	+	100	/	400,0	900,25

Deklarování a viditelnost proměnných

Prohlášení proměnných a konstant mohou být provedena pouze v některých specifických místech kódu GPL.

Lze definovat proměnné:

- Globální proměnné modulu
- Globální proměnné skupiny
- Místní (pouze proměnné)
- Globální proměnné knihovny

Maximální počet deklarovatelných globálních proměnných (modulu a skupiny) je 2048.

Je možné definovat *modifikátory*, které přiřazují proměnným přídavné vlastnosti.

Globální proměnné modulu

Globální proměnné modulu jsou obsaženy ve specifickém souboru, ke kterému je možný přístup volbou položky **Soubor->Otevřít globální proměnné**.

Deklarování se v souladu s předchozími odstavci provádí určením názvu proměnné, po kterém následuje klíčové slovo „AS“, a poté druh data (nebo druhy dat v případě matic).

Tyto proměnné jsou viditelné přímo kódem všech skupin.

Globální proměnné skupiny

Globální proměnné skupiny jsou definovány na začátku kódu týkajícího se skupiny. Musí být deklarovány před funkcemi GPL.

Tyto proměnné jsou viditelné přímo celým kódem obsaženým uvnitř skupiny. Dále je možné rozšířit viditelnost těchto proměnných zvenčí skupiny jejich deklarováním jako „Veřejné“ (Public).

Veřejné proměnné nejsou přímo přístupné zvenčí skupiny. Za účelem přístupu k těmto proměnným se používá jejich název, před kterým je uveden název skupiny, do které patří. Například když je třeba změnit veřejnou proměnnou „posun“ skupiny „jednotkypohybuvosach“ kódem skupiny „main“, bude třeba napsat „SETVAL 10 jednotkypohybuvosach.posun“.

Pro deklarování globální proměnné skupiny se používá stejná syntaxe jako v případě globálních proměnných modulu. Hlavní rozdíl spočívá v definici veřejných proměnných.

Pro definování jedné nebo více veřejných a soukromých proměnných se používají popisky „Public:“ a „Private:“. Například:

```
Public:
  offset as double
  rychlost as float
Private:
  nástroj as integer
```

Místní proměnné

Místní proměnné se deklarují uvnitř těla funkce. Musí být deklarovány před jakoukoli jinou instrukcí s výjimkou deklarování parametrů funkce.

Místní proměnné jsou přístupné uvnitř samotné funkce.

Tyto proměnné jsou vytvářeny s hodnotou 0 (je provedeno přidělení potřebné paměti) pouze na začátku provádění samotné funkce a po skončení provádění dojde k jejich zničení (příslušná část paměti je uvolněna). Globální proměnné jsou vytvářeny při inicializaci modulu a jsou neustále viditelné v části „Diagnostika“.

Deklarování místní proměnné dodržuje již prezentovanou syntaxi, ale musí ji předcházet klíčové slovo „LOCAL“. Například:

```
Function obrabeni
  local poloha_stredu as double
  movabs X,poloha_stredu
  fret
```

Globální proměnné knihovny

Globální proměnné knihovny jsou obsaženy v [knihovnách kódu GPL](#). Jsou obdobou globálních proměnných skupiny.

Modifikátory

Modifikátory: READONLY

Globální proměnné modulu a globální proměnné skupiny mohou být deklarovány jako READONLY.

Proměnná druhu readonly je proměnná, jejíž hodnota nemůže být měněna kódem GPL, ale může být změněna „zvenčí“, tj. z archivu technologických parametrů softwaru Albatros.

Archiv technologických parametrů je tvořen databází, do které jsou ukládány hodnoty, které charakterizují stroj, ale které se mohou dlouhodobě měnit v důsledku změn samotného stroje nebo v důsledku provedených zásahů mimořádné údržby. Tyto informace jsou obvykle vkládány do matrice GPL během inicializace řízení.

Příkladem těchto informací jsou posuny pracovních ploch nebo rozměry a technologické parametry nástrojů.

Při deklarování těchto proměnných jako readonly se ochráníme před náhodnými změnami uložených informací, které se obvykle nemusí měnit během běžné činnosti stroje.

Maximální rozměr proměnné readonly je 128 Kbytů.

Deklarování proměnné readonly se provádí podle níže uvedené syntaxe:

```
readonly NavezvPromenne as druh
```

Modifikátory: NONVOLATILE

Proměnné deklarované **NONVOLATILE**, se ukládají do stálé oblasti RAM (vybavené baterií) namísto běžné RAM.

Proto při vypnutí číslicového řízení nedochází ke ztrátě hodnot uložených do těchto proměnných.

Deklarování proměnné nonvolatile se provádí podle níže uvedené syntaxe:

```
nonvolatile NavezvPromenne as druh
```

Například:

```
nonvolatile PosunPloch[2] as double:posunX double:posunY double:posunZ
```

Jako „nonvolatile“ mohou být označeny pouze globální proměnné skupiny a modulu.

Maximální celkový rozměr proměnných uložených ve stálé RAM je 65536 bytů. Maximální rozměr jednotlivé stálé matrice je 1024 bytů.

Přiřazení RANGE

V okamžiku deklarování proměnné jí lze přiřadit rozsah hodnot. Aktuálně se však neprovádí kontrola dodržení limitů ve fázi provedení, ale je prováděna pouze kontrola kompilátorem v případě přiřazení konstantních hodnot (např. pro inicializaci proměnné).

Hlavní výhodou je tedy jakési autodokumentování kódu.

Definice rozsahu se provádí podle níže uvedené syntaxe:

NazevProměnně Range:valmin..valmax AS druh

Například:

CisloNastroje Range:1..100 as integer

Práva na Čtení / Zápis

Práva na čtení a zápis umožňují určit [minimální úroveň přístupu](#) do systému, potřebnou pro zobrazení (právo na čtení) a změnu (právo na zápis) jejich hodnoty.

Syntaxe je následující:

NazevPromenne Read=S Write=M AS druh

Klíčovými slovy používanými při určování práv jsou:

- READčtení
- WRITE zápis

Přípustnými hodnotami jsou:

- U nebo USER uživatel
- S nebo SERVICE servisní služba
- M nebo MANUFACTURER výrobce
- T nebo TPA firma TPA

K přednastaveným hodnotám patří:

- READčtení pro servisní službu (S nebo SERVICE)
- WRITE zápis pro výrobce (M nebo MANUFACTURER) a firmu TPA (T nebo TPA)

10.1.4 Konstanty

GPL počítá se čtyřmi druhy konstant:

- Integer
- Double
- Char
- String

Konstanty druhu Char jsou deklarovány s použitím jednoduchých uvozovek:

```
Const COD = 'A'
```

Konstanty druhu String jsou deklarovány s použitím dvojitých uvozovek:

```
Const MSG = "Zahájení obrábění"
```

Konstanty druhu Integer a druhu Double jsou deklarovány s následující syntaxí:

```
Const PI = 3.14
Const MSGBOX = 12
```

Pro konstanty druhu Integer se počítá také s binární a hexadecimální notací:

```
Const MASK = $11001001b ; binární
Const MASK = $F5h ; hexadecimální
```

Také konstanty skupiny a knihovna mohou být veřejné nebo soukromé.

Syntaxe je obdobná té, která byla již uvedena pro proměnné.

Příklad:

```
Public:
Const PI = 3.14
Const MSGBOX = 12
Private:
Const MASK = $11001001b
```

POZNÁMKA: Neexistují konstanty druhu Float. Desetinné hodnoty musí být proto jednoznačně deklarovány jako Double. Někdy to může způsobit upozorňující hlášení ze strany kompilátoru (když se používají instrukce GPL optimalizovaná pro použití druhů Float).

Konstanty je možné definovat jako výsledek výrazů výpočtu s následující syntaxí:

```
Const a = 10
Const b = 20
Const c = a + b
```

Připustnými operátory jsou stejné operátory, jaké jsou použity v instrukci [EXPR](#).

Přednastavené konstanty s hodnotou přiřazenou

Jazyk GPL počítá s některými přednastavenými konstantami. Tyto mohou být použity přímo a bez potřeby jejich definování.

K přednastaveným konstantám a odpovídajícím hodnotám patří:

ON	1
OFF	0
UP	+1
DOWN	-1
POSITIVE	+1
NEGATIVE	-1
CW	1
CCW	0
TRUE	1
FALSE	0
NOWAIT	0
WAIT	1
WAITACK	2
STORE	1
NOSTORE	0
NOPLACE	0
COM1	0
COM2	1
COM3	2
COM4	3
COM5	4
COM6	5
COM7	6
COM8	7
NOPARITY	0
ODDPARITY	1

EVENPARITY 2**Přednastavené konstanty s hodnotou přiřazenou při spuštění softwaru Albatros**

Jazyk GPL počítá s některými přednastavenými konstantami, jejichž hodnota je určena při spuštění softwaru Albatros. Mohou být použity v instrukci [IFDEF](#).

_ID_MODULE	číslo aktuálního modulu. Hodnota modulu se musí nacházet v rozmezí od 0 do 15.
_REMOTE_MODULE	typ modulu. Konstanta má hodnotu 1, jedná-li se o vzdálený modul, hodnotu 0, když se jedná o místní modul nebo není určena, když modul není nakonfigurován v Konfiguraci systému.
_VER_MAJOR	číslo hlavní verze softwaru Albatros. Když je verze softwaru Albatros 3.2.1, hodnota hlavní verze je 3.
_VER_MINOR	číslo vedlejší verze softwaru Albatros. Když je verze softwaru Albatros 3.2.1, hodnota vedlejší verze je 2.
_VER_REVISION	číslo revize softwaru Albatros. Když je verze softwaru Albatros 3.2.1, hodnota revize je 1.
_VER_SP	řetězec, který popisuje Service Pack, je-li nainstalován (například „Service Pack 1f“); v opačném případě není určena.
_VER_FULL	číslo kompletní verze. V případě verze 3.2.1 platí \$00030201H.

10.1.5 Klíčová slova

Klíčová slova jsou identifikátory s vyhrazeným použitím, které nemohou být používány jinak.

K dostupným klíčovým sloům patří:

Všechny názvy instrukcí GPL	Popis všech instrukcí GPL je uveden v návodu „Instrukce“
Všechny druhy dat	Viz Proměnné
Parametry druhu zařízení	Viz Parametry druhu zařízení
EXIST	používané v instrukci IFDEF pro ověření existence skupiny. Viz instrukce IFDEF
NOTEXIST	používané v instrukci IFDEF pro ověření neexistence skupiny. Viz instrukce IFDEF
LINKED	používá se v instrukci IFDEF pro aktivaci kompilace bloků kódu, když je zařízení připojeno ve virtuálně-fyzickém deklarování. Viz instrukce IFDEF
UNLINKED	používá se v instrukci IFDEF pro aktivaci kompilace bloků kódu, když zařízení není připojeno ve virtuálně-fyzickém deklarování. Viz instrukce IFDEF
FUNCTION	Deklarování funkce. Viz Funkce
AS	Používá se pro deklarování proměnných. Viz Proměnné
PUBLIC	Jedná se o atribut funkce. Viz Funkce
AUTORUN	Jedná se o atribut funkce. Informuje o tom, že se jedná o automaticky spustitelnou funkci. Viz Funkce
R= nebo READ	Jedná se o atribut funkce nebo proměnné. Informuje o úrovni přístupu při čtení. Viz Funkce , Proměnné a Přístupová práva
W= nebo WRITE	Jedná se o atribut funkce nebo proměnné. Informuje o úrovni přístupu při zápisu. Viz Funkce , Proměnné a Přístupová práva

CONST	Umožňuje přiřadit příznačný název, nazvaný symbolická konstanta, namísto čísla, znaku nebo řetězce. Viz Proměnné
READONLY	Jedná se o atribut globální proměnné. Viz Proměnné
NONVOLATILE	Jedná se o atribut globální proměnné. Viz Proměnné
PRIVATE	Jedná se o atribut funkce. Viz Funkce
RANGE	Používá se pro definici intervalu hodnot proměnné. Viz Proměnné
USER	Jedná se o atribut funkce nebo proměnné. Signalizuje druh přístupu, v tomto případě uživatele. Viz Funkce nebo Proměnné .
SERVICE	Jedná se o atribut funkce nebo proměnné. Signalizuje druh přístupu, v tomto případě údržbu. Viz Funkce nebo Proměnné .
MANUFACTURER	Jedná se o atribut funkce nebo proměnné. Signalizuje druh přístupu, v tomto případě výrobce. Viz Funkce nebo Proměnné .
TPA	Jedná se o atribut funkce nebo proměnné. Signalizuje druh přístupu, v tomto případě TPA. Viz Funkce nebo Proměnné .

10.1.6 Funkce

Funkce představují minimální blok kódu GPL. Instrukce GPL nemohou být vloženy sekvenčně do souboru, ale je třeba je seskupit do funkcí. Maximální počet prohlásitelných funkcí je 8191.

Z hlediska kompilátoru jsou funkcemi všechny bloky kódu GPL, které začínají řádkem, na němž je prvním slovem slovo `FUNCTION`. Neexistuje však klíčové slovo, které identifikuje konec textu funkce: Funkce končí na řádku, který předchází začátek další funkce nebo konec souboru, který obsahuje funkce.

Syntax pro definování funkce je:

```
FUNCTION   NázevFunkce Attributy
            Parametry
            Místní proměnné
            Seznam instrukcí GPL
```

Funkce je dále speciálním druhem zařízení softwaru Albatros. Sdílí se zařízeními některé vlastnosti: jedinečný název (nepřeložitelný), indikátor viditelnosti (zda se jedná o zařízení veřejné, nebo ne), [úroveň přístupu](#) při čtení a úroveň přístupu při zápisu (viz níže uvedený odstavec).

Přístupová práva

Přístupovým právům umožňují vymezit minimální úroveň přístupu do systému, která umožňuje jeho viditelnost (právo na čtení) a proveditelnost (právo na zápis).

Syntaxe je následující:

```
Function   NázevFunkce   READ=S WRITE=M
```

Práva jsou identifikována klíčovými slovy `READ` (čtení) a `WRITE` (provádění) K hodnotám, které odpovídají úrovním přístupu a které lze přiřadit, patří:

- U nebo `USER` uživatel
- S nebo `SERVICE` servisní služba
- M nebo `MANUFACTURER` výrobce
- T nebo `TPA` tpa

K přednastaveným hodnotám patří:

- `READ` čtení pro servisní službu (S nebo `SERVICE`) a uživatel (U nebo `USER`)
- `WRITE` zápis pro výrobce (M nebo `MANUFACTURER`) a tpa (T nebo `TPA`)

Funkce Autorun

Funkce druhu autorun je prováděna automaticky při inicializaci stroje.

Funkce autorun se vyznačují tím, že po svém ukončení v důsledku chyby systému jsou následně spouštěny automaticky.

Syntaxe je následující:

Function NázevFunkce autorun

Stačí tedy přidat k deklarování funkce modifikátor „autorun“.

Funkce Public

Obvykle může být provádění funkce zahájeno (vyvoláno) výhradně z kódu, který se nachází uvnitř souboru skupiny. Aby mohlo být provádění funkce zahájeno z kódu GPL jiné skupiny, musí být funkce druhu **public**. Syntax pro definování funkce public je následující:

Function NázevFunkce public

Stačí tedy přidat k deklarování funkce modifikátor „public“.

Výjimkou jsou funkce, které náležejí do meziskupiny, které jsou vždy druhu **public**.

Funkce Podskupiny

Funkce může být přiřazena podskupině jednoduše tak, že se před názvem funkce uvede název podskupiny. Název podskupiny a název funkce se musí oddělit tečkou „.“. Například níže uvedená funkce patří do podskupiny „X“ skupiny „JednotkyPohybuOse“.

Function X.vynulovani
 local vel as float
 movabs X,100
 waitstill X
Fret

Asynchronní funkce

Asynchronní funkce bude automaticky vyvolána číslicovým řízením při vzniku události, na kterou je tato funkce vázána.

Existují tři druhy těchto událostí:

- Změna stavu digitálního vstupu: instrukce ONINPUT
- Změna stavu bitu příznaku nebo přepínače příznaku: instrukce ONFLAG
- Vznik chyby systému: instrukce ONERRSYS

Při vzniku události bude vyvolána funkce (ne jako samostatná úloha, ale v kontextu úloh, v němž byla provedena příslušná instrukce ON...) jako implicitní FCALL implicita, a to ihned poté, co aktuální instrukce ukončila své provádění.

Asynchronní funkce typicky slouží pro správu nouzových situací a musí být velmi rychlé. Z tohoto důvodu tyto funkce nemohou používat jakoukoli instrukci GPL, ale podmnožinu, která zaručuje krátké doby provádění.

Funkce se vstupními parametry (parametrické)

Funkce může mít deklarované parametry na vstupu, ale v žádném případě nevrací žádnou hodnotu.

Parametry mohou být viděny jako speciální místní proměnné, jejichž hodnota je inicializována zvenčí v okamžiku, kdy je funkce spuštěna. Parametry jsou deklarovány klíčovým slovem **PARAM** a dodržují stejnou syntaxi, jaká se používá pro místní proměnné. Parametry musí být uvedeny v seznamu na prvních dvou řádcích těla funkce, před jakoukoli jinou instrukcí a před místními proměnnými.

Existují dva způsoby předání parametrů:

- **Prostřednictvím hodnoty:** Prostřednictvím hodnoty jsou předávány všechny druhy jednoduchých dat, tj. CHAR, INTEGER, FLOAT a DOUBLE. Předání prostřednictvím hodnoty znamená vytvoření kopie původní hodnoty. Změny parametrů se projeví pouze v rámci dané funkce.
- **Prostřednictvím odkazu:** Prostřednictvím odkazu jsou předávány strukturované druhy dat, jako jsou ARRAY, MATRICE a ŘETĚZCE. Předání prostřednictvím odkazu znamená použití původní proměnné, a proto provedené změny parametru mají vliv v kontextu vyvolávající funkce. Tuto charakteristiku lze využít pro vrácení zpětných hodnot vyvolávající funkci.

Obvykle je provádění funkce zahájeno instrukcí FCALL. Když se jedná o parametrickou funkci, po názvu funkce bude třeba uvést seznam hodnot, které mají být přiřazeny parametrům.

V níže uvedeném příkladě najdete funkci, která provádí obrábění vrtáním. Polohy středu otvoru a rychlost posuvu jednotky pohybu v ose Z jsou funkcí předány jako parametry.

Function Vrtani
 Param Qx as Double ; poloha X středu otvoru
 Param Qy as Double
 Param vel as Float ; rychlost posuvu

 Movabs X, Qx, Y, Qy
 Waitstill X,Y

Fret

Vyvolání této funkce, například kvůli vytvoření otvoru v polohách (12.5 , 25.7) s rychlostí posuvu 3 m za minutu, by mohlo být provedeno níže uvedeným způsobem:

```
Fcall Vrtani 12.5, 25.7, 3.0
```

Parametry předané funkci musí odpovídat z hlediska počtu a druhu těm, které jsou deklarovány ve vyvolané funkci. Provádění vyvolávající funkce bude pokračovat po ukončení provádění vyvolávané funkce.

Jako parametr funkce může být [deklarováno také zařízení](#). To umožňuje zápis funkcí všeobecného použití.

Například funkce vynulování použitelná se všemi jednotkami pohybu v osách, které se nacházejí na stroji, je:

```
Function VYNULOvani PUBLIC
    param      jednotkapohybuose as Axis
    movabs     jednotkapohybuose,100
```

```
Fret
```

```
Function Main
```

```
.....
```

```
Jednotkypohybuose.Vynulovani x
```

```
Fret
```

Funkce vynulovani patří do skupiny Jednotkypohybuose a je deklarována druhem PUBLIC, aby mohla být viděna funkcemi deklarovanými v ostatních skupinách. Funkce Main vyvolá funkci vynulování skupiny jednotek pohybu v osách, přičemž jako parametr na vstupu se uvede jednotka pohybu v ose, kterou je třeba pohybovat.

10.1.7 Parametry druhu zařízení

Druhy zařízení jsou speciální proměnné, které umožňují vztahovat se na zařízení stroje.

Tyto druhy dat mohou být použity **výhradně** v deklarování [parametrů funkce](#). Proto nemohou být deklarovány proměnné tohoto druhu. Definice názvů a dalších charakteristik zařízení i nadále náleží do Konfigurace systému. V následující tabulce jsou uvedeny druhy Zařízení a příslušná klíčová slova pro použití při deklarování odpovídajících parametrů.

Druh	Klíčové slovo
Digitální vstup	INPUTDIG
Digitální výstup	OUTPUTDIG
Analogový vstup	INPUTANALOG
Analogový výstup	OUTPUT ANALOG
Jednotka pohybu v ose	AXIS
Časovač	TIMER
Počítadlo	COUNTER
Bit příznaku	FLAGBIT
Přepínač příznaku	FLAGSWITCH
Port příznaku	FLAGPORT
Port vstupu	INPUTPORT
Port výstupu	OUTPUTPORT
Funkce	FUNCTION
Všeobecné zařízení	DEVICE
Úloha	TASK

V níže uvedeném příkladu je deklarován a používán parametr druhu jednotka pohybu v ose:

```
Function zkouska
    Param jednotkapohybuose as axis
```

```
    MovAbs     jednotkapohybuose,100
    WaitStill  jednotkapohybuose
```

```
Fret
```

10.1.8 Zpracování více úloh (Multitasking)

Systém Albatros je multitaskového typu, a proto je možné mít více úloh GPL prováděných současně, kde pod pojmem úloha se má na mysli proces správy logické entity (obvykle skupiny).

Druhy poskytnutých úloh jsou dva: běžné úlohy a úlohy zpracovávané v reálném čase (real-time).

Běžné úkoly

Multitasking je založen na algoritmu spolupráce, který vychází z priority. Ten zaručuje všem úlohám cyklické provádění a mění jejich prioritu. Algoritmus plánování procesů počítá s provedením jedné instrukce pro každou aktivní úlohu (stav běhu (running)). Každému úloze je přiřazena priorita nastavitelná instrukcí [SETPRIORITYLEVEL](#). Priorita je identifikována celým číslem z rozsahu od 0 (maximální priorita) do 255 (minimální priorita). U úloh s prioritou 0 (nula) bude provedena jedna instrukce při každém cyklu plánování procesů, u úloh s prioritou 1 bude provedena jedna instrukce každé dva cykly plánování procesů a tak dále až po úlohy s prioritou 255, u nichž bude provedena jedna instrukce každých 256 cyklů plánování procesů.

Provádění běžných úloh je asynchronní vzhledem k frekvenci aktualizace stavu jednotek pohybu v osách. To znamená, že neexistuje záruka toho, že funkce GPL bude dokončena v časovém intervalu, který uplyne mezi dvěma aktualizacemi jednotek pohybu v osách.

Úloha je identifikována názvem funkce GPL, ze které bylo zahájeno její provádění.

Provádění úlohy může začít:

- automaticky při inicializaci systému: Hlavní funkce meziskupiny a funkce autorun.
- po provedení instrukce [STARTTASK](#).
- po spuštění grafického rozhraní softwaru Albatros v manuálním režimu.

Každá úloha je charakterizována vnitřním stavem:

RUNNING	úloha je prováděna
HOLD	úloha je pozastavena
BREAK	úloha je přerušena ladicím programem

Hierarchie úloh je zorganizována ve formě stromu. Každá úloha je vytvořena jinou úlohou, z níž pochází, to znamená, že když je ukončena rodičovská úloha, budou ukončeni i všichni potomci.

Maximální počet současně prováděných úloh je 500.

Mějte na paměti, že vysoký počet prováděných úloh znamená snížení rychlosti, se kterou jsou prováděny jednotlivé úlohy.

V případě, že si myslíte, že by vytvářená aplikace mohla vyžadovat více než 200 úloh, je třeba použít vhodný hardware, jako jsou moduly Cn2128.

Úloha v reálném čase (Real-Time)

Úlohy v reálném čase se odlišují od předcházejících tím, že nepodléhají mechanismu plánování procesů a nemají žádnou přiřazenou prioritu, ale jsou prováděny kompletně při každé aktualizaci stavu jednotek pohybu v osách (provádění jednotek pohybu v osách v reálném čase (real-time)).

Je jednoznačně potřebné, aby provádění těchto úloh skončilo v určené době, protože provádění výše popsaných úloh GPL zůstane během provádění úloh v reálném čase pozastaveno.

Systém provede kontroly doby provedení úloh v reálném čase a v případě, že překročí maximální povolenou dobu, bude vytvořena chyba systému.

Proto se doporučuje vytvořit nekonečné cykly (např. s použitím GOTO) uvnitř těchto úloh, cykly, které ani nejsou potřebné, protože provádění kódu začíná od začátku při každém provedení jednotek pohybu v osách v reálném čase.

Aby se zabránilo nadměrným dobám provádění úloh, úlohy vykonávané v reálném čase podléhají omezením při použití některých instrukcí GPL. Instrukce, jejichž použití není dovoleno, jsou ty, které [nejsou použitelné při přerušení \(interrupt\)](#).

Použití úlohy vykonávané v reálném čase se doporučuje pouze u těch druhů činnosti, které musí být bezpodmínečně provedeny synchronním způsobem s aktualizací poloh jednotek pohybu v osách. Pro většinu kontrolních činností je vhodné použít běžné úlohy.

Úlohy vykonávané v reálném čase se spouštějí instrukcí [STARTREALTIMETASK](#) a mohou být přerušeny instrukcí [ENDREALTIMETASK](#). Je možné aktivovat maximálně 256 úloh prováděných v reálném čase současně.

Neplatí mechanismus dědičnosti, a proto když úloha, která spustila úlohu vykonávanou v reálném čase, skončí, spuštěná úloha bude vykonávána i nadále.

Místní proměnné deklarované v úloze prováděné v reálném čase jsou inicializovány teprve při spuštění úlohy a poté si udržují hodnotu posledního provedení.

Úlohy vykonávané v reálném čase nejsou charakterizovány stavy, se kterými se počítá u běžných úloh. Je možné provést ladění úlohy vykonávané v reálném čase, ale v tomto případě systém automaticky deklaruje úlohu na „běžnou úlohu“ po celou dobu trvání činnosti ladění.

Když se v některé úloze vykonávané v reálném čase vyskytne chyba systému, úloha bude deklasována na běžnou úlohu a bude uvedena do stavu HOLD, aby byla umožněna analýza pomocí ladicího programu.

10.1.9 Komunikace

Komunikace mezi GPL a okolním světem probíhají prostřednictvím tří odlišných způsobů:

- SEND / RECEIVE
- Sériové komunikace
- IPC

Send / Receive

Instrukce [SEND](#) a [RECEIVE](#) implementují mechanismus komunikace orientovaný na hlášení.

Komunikace může probíhat buď uvnitř samotného modulu (málo výhodná), mezi moduly jednoho vedení nebo mezi moduly a dozorným softwarem Albatros nebo s aplikacemi OLE.

Fungování je podobné výše uvedenému; každému hlášení je přiřazený adresát, identifikátor odeslané (nebo vyžádané) informace, samotná informace plus služební informace. Albatros Plní funkci sběru a třídění informací a v některých případech dodává přímo požadované informace.

Tento režim komunikace se obvykle používá pro odesílání programu obrábění mezi obráběním, dozorným PC a řídicími moduly pro synchronizaci činnosti strojů linky a pro vzájemné spojení s externími aplikacemi (server OLE).

Sériové komunikace

Jazyk GPL poskytuje některé instrukce, například [COMREAD](#) a [COMWRITE](#), které umožňují odesílat data prostřednictvím sériových portů řízení. Proto je možné vzájemné přepojení řízení s externími zařízeními, jako jsou frekvenční měniče, terminály nebo PLC. Pokud se tyto instrukce používají vhodně, umožňují implementovat protokoly sériové komunikace jako MODBUS-RTU apod.

IPC

IPC neboli Inter Process Communication představuje jeden ze způsobů komunikace mezi procesy. Tento způsob konkrétně umožňuje definovat oblast paměti sdílenou dvěma nebo více procesy, která může být použita pro výměnu informací.

Obvykle se používá v případě, když výkon dodaný rozhraním OLE systému Albatros není vhodný.

Na straně GPL bude komunikace IPC implementována instrukcemi [SENDIPC](#), [WAITIPC](#) a [TESTIPC](#).

V případě místního modulu se mohou externí procesy vztahovat na API dodané od RTX, nebo na komponent COM gplipc2.dll dodaný firmou TPA, který zjednodušuje jeho použití.

V případě vzdáleného modulu používají procesy spuštěné v dozorném PC komponent COM **gplipc2net.dll**.

Pro podrobnější informace kontaktujte firmu TPA.

10.1.10 Proměnné určené k použití při programování

Větší část instrukcí byla vytvořena s různými druhy proměnných (CHAR, INTEGER, FLOAT, DOUBLE). Platí však, že každá instrukce má svůj přednostní druh proměnné, pro který byla optimalizována. Za účelem dosažení nejlepší výkonnosti během provádění kódu GPL se doporučuje používat druhy proměnných doporučené v popisu každé instrukce. Obecně platí, že se doporučuje dodržovat níže uvedenou tabulku, která přiřazuje hlavním veličinám používaným v programování příslušné optimální druhy proměnných:

veličina	druh
poloha	<i>double</i>
rychlost	<i>float</i>
doba	<i>double</i>
počítadlo	<i>integer</i>
hodnota portu / port příznak	<i>integer</i>
časový limit	<i>double</i>
analogový vstup / výstup	<i>float</i>
směrové kosiny	<i>double</i>
kontrolní znak řetězce	<i>char</i>
zrychlení / zpomalení	<i>integer</i>

10.1.11 Jednotky pohybu v osách

Výrazem „jednotka pohybu v ose“ je obvykle označován elektromechanický systém, jehož účelem je řízené pohybování součásti obráběcího stroje.

Tento systém můžeme popsat z hlediska prvků, které jej tvoří, a tyto prvky mohou být rozděleny na základě svých technologických charakteristik.

Budeme tedy mít mechanické součásti, ke kterým patří:

- nosná konstrukce
- vedení
- ložiska
- šroub + ložiska kuličkového šroubu

Jejichž úkolem je potlačovat přítomné síly, snižovat tření, transformovat rotační pohyb na posuvný apod.

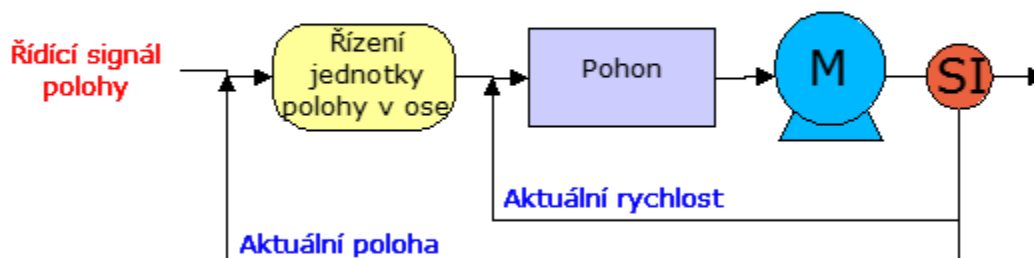
Budeme tak mít elektrické a elektronické komponenty, ke kterým patří:

- motor
- koncové spínače
- snímač impulzů
- tachometrické dynamo,

jejichž úkolem je dodávat výkon potřebný pro pohyb a zaznamenávat stav systému.

Tyto prvky jsou vzájemně spojeny tak, aby umožňovaly provedení pohybů řízeným způsobem.

Schéma řízení se zpětnou vazbou

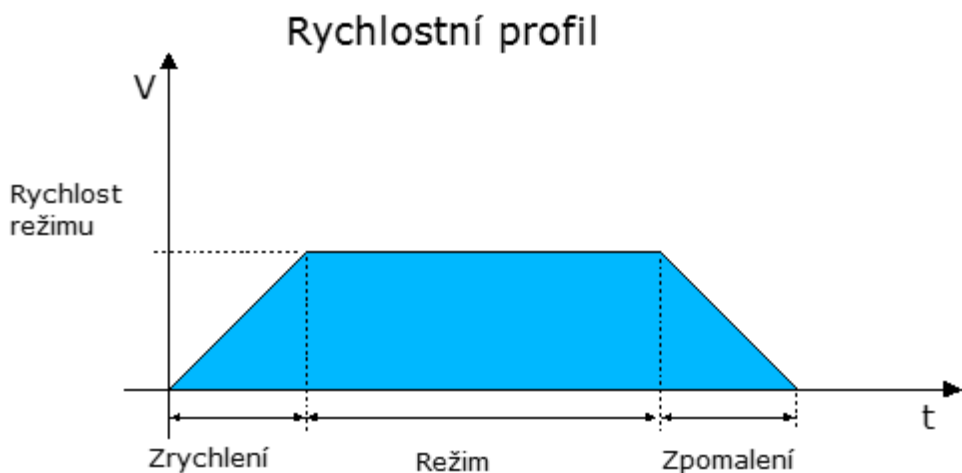


Úkolem číslicového řízení je řídit polohu a pohyb jednotky pohybu v ose.

Pohyb jednotky pohybu v ose je složen z 5 fází:

Zrychlení	počáteční fáze, během které jednotka pohybu v ose postupně zvyšuje svou rychlost až po dosažení požadované hodnoty
Režim	střední fáze, během které se jednotka pohybu v ose pohybuje konstantní rychlostí (tato fáze může také chybět, když je prostor, který má být překonán, menší než prostor zrychlení a zpomalení)
Zpomalení	fáze, během které dochází k postupnému snížení rychlosti až na 0
Okno	fáze čekání na omezení chyby smyčky na hodnotu uvedenou v konfiguraci pod názvem „okno dosažení cílové polohy“
Poloha	ukončení pohybu

Na konci pohybu musí být jednotka pohybu v ose umístěna uvnitř intervalu nazvaného okno dosažení cílové polohy (definuje toleranci polohování jednotky pohybu v ose). Když k tomu nedojde v průběhu 5 sekund od očekávaného ukončení pohybu, systém vytvoří chybu systému „nedokončený pohyb“.



Pro každý pohyb číslicové řízení vypočítává profil rychlosti jako ten, který je patrný na obrázku, a poté vypočítává cílové polohy rozdělením profilu rychlosti do časových intervalů rovnajících se době potřebné na obnovení stavu jednotky polohy v ose a výpočet oblasti každé části. Oblast představuje zvýšení polohy, které musí dosáhnout v časovém intervalu za účelem dodržení výše uvedeného profilu rychlosti.

Řízení jednotky pohybu v ose je implementováno prostřednictvím regulátoru PID, který zajišťuje „uzavření smyčky sledování polohy“, tj. poskytnutí řídicího signálu rychlosti pohonu vypočteného na základě polohy, které má být dosaženo (cílová poloha) a reálné polohy čtené snímačem impulzů. Rozdíl mezi reálnou a cílovou polohou se nazývá **Chyba Smyčky**.

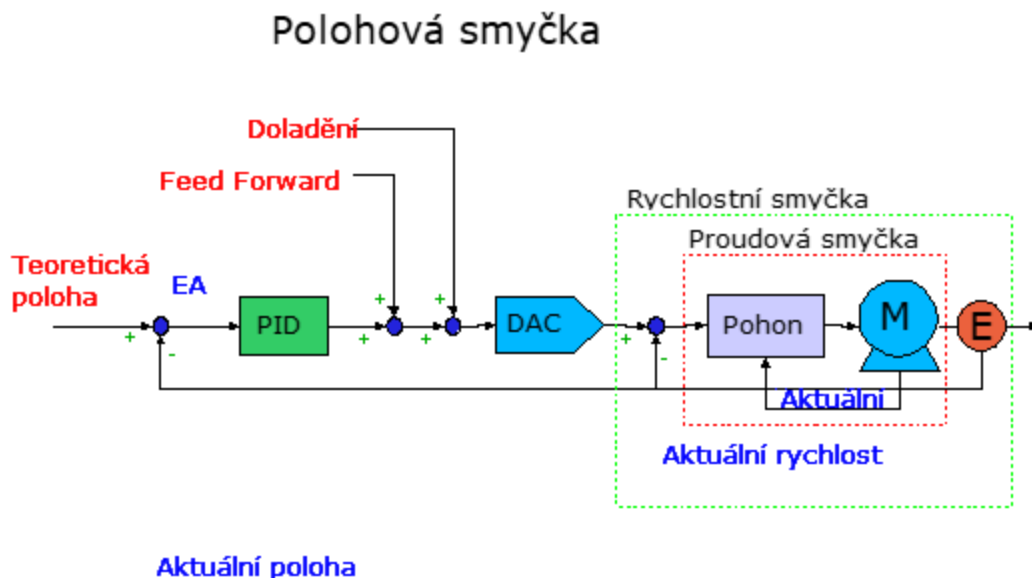


Schéma řízení jednotky pohybu v ose Albatros

10.1.12 Korektory linearity

Tabulka korektorů linearity jednotky pohybu v ose je viděna jako matice, s názvem *NázevSkupiny.NázevPodskupiny.NázevJednotkypohybuose#correctors* o *NázevSkupiny.NázevJednotkypohybuose#correctors* a může být použit ve všech příkazech, ve kterých se provádí přístup k maticím, prvkům matice a řádkům matice.

Počet sloupců matice odpovídá počtu jednotek pohybu v osách, nastavených v okně korektorů linearity v konfiguracích jednotek pohybu v osách. Samokorektory jsou vloženy do prvního sloupce. Všechny hodnoty korekce jsou typu float. Celkový počet řádků matice lze získat prostřednictvím příkazu GPL [LASTELEM](#).

Tyto matice jsou přístupné v rámci čtení i v rámci zápisu, a v případě jejich změny jsou jejich hodnoty okamžitě použity pro korekci polohy, byla-li korekce aktivována.

Příklad:

```
Function ReadCorr
local i as integer
local j as integer
local row as integer
local column as integer
local firstvalue as float

; čtení první hodnoty samokorekce jednotky pohybu v ose AX
firstvalue = X.Ax#correctors[1][1]
; počet jednotek pohybu v osách, v okně korektorů linearity
setval 3 column
lastelem X.Ax#correctors row
; zvýšení všech korekcí o konstantní hodnotu
for i 1 row
for j 1 column
X.Ax#correctors[i][j] = X.Ax#correctors[i][j] + 0.025
next
next

fret
```

10.1.13 Správa hlášení v jazyce

Jak již bylo popsáno v kapitole Složení systému, Albatros podporuje [zobrazování textových hlášení ve více jazycích](#).

Tato podpora je poskytována prostřednictvím použití programu TpaLangs, který je externí vůči softwaru Albatros, a který spravuje archivy hlášení. Tento program pomáhá v činnosti překladu hlášení do jednotlivých jazyků.

Text spojený s Chybami cyklu a Hlášeními

Specifickým druhem textů, které jsou obvykle zobrazovány softwarem Albatros, jsou Hlášení a Chyby cyklu vytvářené kódem GPL.

Ty jsou obvykle definovány tím, kdo vyvíjí kód GPL během zápisu samotného kódu. Pro usnadnění práce programátora editor GPL umožňuje vložit text hlášení přímo ze softwaru Albatros, tj. bez potřeby použít TpaLangs.

Druhou možností pro správu hlášení v jazyce je použití instrukce GPL [DEFMSG](#).

10.1.14 Správa chyb systému

Když dojde k výskytu [chyby systému](#) (Viz kapitola **Chyby systému->Úvod k Chybám Systému**), běžným chováním řízení je ukončení všech úkolů: Správa chyb systému umožňuje vyhnout se ukončení úkolů u těch úkolů, pro které byla aktivována.

Chyby systému vytvořené poruchou, podtečením zásobníku a přetečením zásobníku jsou spravovány přímo řízením bez vyvolání funkce pro správu chyb systému: Úkol je uveden do stavu HOLD.

Funkce správy chyb

Uvnitř kódu GPL musí být definována jedna nebo více funkcí pro analýzu chyby systému a pro pozdější definici vhodných akcí za účelem zabezpečení stroje. Funkce, kterou je třeba vyvolat, je předána v podobě parametru instrukci GPL [ONERRSYS](#). (Viz kapitola **Jazyk GPL->Instrukce->Správa toku->ONERRSYS**).

Při výskytu chyby systému bude úkol, který vytvořil chybu, uveden do stavu HOLD. Když úkoly autorun vytvářejí chyby systému, budou znovu spuštěny teprve poté, co se již systém nebude nacházet ve stavu poruchy (FAULT).

Když je chyba systému vytvořena bez čísla úkolu, bude do stavu HOLD uveden aktuální úkol.

10.2 Speciální funkce

10.2.1 Uživatelské přizpůsobení pohybu jednotek pohybu v osách

Grafické rozhraní systému Albatros počítá s možností pohybovat jednotkami pohybu v ose v manuálním režimu a dále poskytuje grafickou podporu pro jejich kalibraci.

Pohyb jednotek pohybu v osách v manuálním režimu probíhá prostřednictvím ovládacího panelu pro pohyb jednotky pohybu v ose a kalibrace může být provedena prostřednictvím panelu kalibrace. Oba panely lze zobrazit z funkce Diagnostika a z grafických zobrazení.

V obou případech je pohyb jednotky pohybu v ose ovládán skupinou funkcí GPL, jejichž provádění zůstane pro uživatele neviditelné.

Systém má k dispozici přednastavený soubor těchto funkcí, které jsou vhodné ve většině případů. Však může být potřebné je uživatelsky přizpůsobit, například kvůli zavedení některých omezení pohybu jednotek pohybu v osách souvisejících se stavem stroje nebo pro správu pomocných zařízení, jako jsou například brzdy jednotky pohybu v ose.

Uživatelské přizpůsobení probíhá prostřednictvím vytvoření dvou funkcí GPL pro každou jednotku pohybu v ose: Jedné pro pohyb v manuálním režimu, druhé pro kalibraci. Tyto funkce jsou volitelné, a proto v případě, že je systém zaznamená, používají se; v opačném případě jsou používány standardní funkce. Dále se počítá také s částečným uživatelským přizpůsobením funkcí pohybu.

Pohyb v manuálním režimu

Uživatelsky přizpůsobené funkce *manuálního pohybu* musí dodržovat níže uvedená pravidla:

- Funkce musí patřit do stejné skupiny nebo podskupiny, do které patří jednotka pohybu v ose, na kterou se odvolává.
- Název funkce bude muset být **MoveAx#navez_jednotkypohybuose**, kde navez_jednotkypohybuose je třeba nahradit názvem definovaným v konfiguraci. Například pro jednotku pohybu v ose X bude název funkce: MoveAx#X .
- Funkce bude muset mít níže uvedené parametry:
 1. **Požadovaný úkon.** Může jím být pohyb do absolutní polohy, inkrementální pohyb, zastavení apod. Úkony jsou identifikovány celým číslem, přičemž kompilátor GPL používá přednastavené konstanty přiřazené jednotlivým úkonům:

_MOVAXABS	pohyb do absolutní polohy
_MOVAXINC	pohyb do inkrementální polohy
_MOVAXSET	nastavení polohy
_MOVAXFREE	nastavení stavu free
_MOVAXNORMAL	nastavení stavu normal
_MOVAXEND	obnovení stavu jednotky pohybu v ose na konci pohybu (neslouží k zastavení jednotky pohybu v ose)
 2. **Výsledek.** Slouží systému k tomu, aby věděl, zda byl požadovaný úkon spravován uživatelsky přizpůsobenou funkcí. Když úkon není spravován, bude použita odpovídající standardní funkce. Jedná se tedy o návratovou hodnotu, kterou musí uživatelsky přizpůsobená funkce nastavit, a za tímto účelem je definován jako parametr, který prošel odkazem (pole jednoho prvku).
 3. **Rychlost.** Je významná pouze v případě, že je požadovaným úkonem pohyb, a v takovém případě nastaví jeho rychlost.
 4. **Poloha.** Je významná pouze při úkonech pohybu a nastavení polohy.

Příklad uživatelsky přizpůsobené funkce pohybu:

```
Function MoveAx#X
  param action as integer
  param result[1] as integer
  param speed as float
  param position as double

  setval      1,result[1]

  select action
  case _MOVAXEND
```

```

        fcall EndMovement
case _MOVAXABS
        fcall AbsMovement X, speed, position
case _MOVAXINC
        fcall IncMovement X, speed, position
case _MOVAXSET
        fcall PositionSet X, position
case _MOVAXFREE
        fcall FreeAxis
case _MOVAXNORMAL
        fcall NormalAxis
case else
        call Unknown
endselect

fret

Unknown:
        setval      0, result[1]
        ret

```

Funkce EndMovement, AbsMovement apod. (názvy nejsou závazné) budou muset implementovat uživatelsky přizpůsobenou správu požadovaných úkonů. Pro usnadnění práce programátora uvádíme [standardní funkce pohybu](#), které mohou sloužit jako příklad pro vývoj uživatelsky přizpůsobených funkcí.

Nastavení

Uživatelsky přizpůsobené funkce kalibrace musí dodržovat níže uvedená pravidla:

- Funkce musí patřit do stejné skupiny nebo podskupiny, do které patří jednotka pohybu v ose, na kterou se odvolává.
- Název funkce bude muset být **CalibAx#nazev_jednotkypohybuose** kde nazev_jednotkypohybuose je třeba nahradit názvem definovaným v konfiguraci. Například pro jednotku pohybu v ose X bude název funkce: CalibAx#X .
- Funkce bude muset mít níže uvedené parametry:
 1. **Požadovaný úkon.** Může se jednat o pohyb z bodu do bodu nebo o interpolovaný pohyb.
 2. **Výsledek.** Slouží systému k tomu, aby věděl, zda je požadovaný úkon spravován uživatelsky přizpůsobenou funkcí. Když daný úkon není spravován, bude použita odpovídající standardní funkce.
 3. **Rychlost.** Rychlost pohybu během kalibrace
 4. **Kladná poloha.** Kladná poloha střídavého pohybu v rámci kalibrace.
 5. **Záporná poloha.** Záporná poloha střídavého pohybu v rámci kalibrace.
 6. **Čekání.** Čekání mezi dvěma po sobě následujícími pohyby

POZNÁMKA: Je třeba mít na paměti, že v některých případech mají úkony prováděné na panelu kalibrace za následek provedení funkce pohybu jednotky pohybu v ose. Například na konci pohybu kalibrace (při stisknutí tlačítka zastavení) bude provedena operace obnovení stavu jednotky pohybu v ose a za tímto účelem bude vyvolána funkce pohybu v ose, která je uživatelsky přizpůsobena parametrem úkonu nastaveným na _MOVAXEND. Stejně tak při změně polohy jednotky pohybu v ose na panelu kalibrace bude vyvolána funkce pohybu jednotky pohybu v ose s parametrem úkonu nastaveným na _MOVAXSET.

Příklad uživatelsky přizpůsobené funkce kalibrace:

```

Function CalibAx#X
    param action as integer
    param result[1] as integer
    param speed as float
    param PosPosition as double
    param NegPosition as double
    param waitTime as float

    setval      1,result[1]

    select action

```

```

case _CALAXPP
    fcall PPCalibration X, speed, PosPosition, NegPosition, _
        waitTime
case _CALAXINT
    fcall IntCalibration X, speed, PosPosition, NegPosition, _
        waitTime
case else
    call Unknown
endselect

fret

Unknown:
    setval      0, result[1]
    ret

```

Funkce PPValibration a IntCalibration (názvy nejsou závazné) budou muset implementovat uživatelsky přizpůsobenou správu požadovaných úkonů. Pro usnadnění práce programátora uvádíme [standardní funkce kalibrace](#), které mohou sloužit jako příklad pro vývoj uživatelsky přizpůsobených funkcí.

Interakce s oknem Pohybování jednotkou pohybu v ose v manuálním režimu

Funkce pro interakci s oknem manuálního pohybu musí dodržovat níže uvedená pravidla:

- Funkce musí patřit do stejné skupiny nebo podskupiny, do které patří jednotka pohybu v ose, na kterou se odvolává.
- Název funkce bude muset být **MoveAx#navez_jednotkypohybuose#Úkon**, kde navez_jednotkypohybuose je třeba nahradit názvem jednotky pohybu v ose definovaným v konfiguraci a Úkon bude moci nabýt jedné z níže uvedených definic:

OPEN	poukazuje na to, že uživatel právě otevřel okno pohybování jednotkou pohybu v ose
CLOSE	poukazuje na to, že uživatel právě zavírá okno pohybování jednotkou pohybu v ose
ACTIVE	signalizuje, že okno pohybování jednotkou pohybu v ose je aktivní
INACTIVE	signalizuje, že okno pohybování jednotkou pohybu v ose není aktivní
JOG	poukazuje na to, že je nastaven pohyb pro přesun spravovaný v režimu runtime obsluhou
STEP	poukazuje na to, že byl nastaven pohyb s přednastaveným krokovým přesunem
ABSOLUTE	poukazuje na to, že byl nastaven pohyb s přesunem do určené polohy

V případě, že bylo např. otevřeno okno pohybování jednotkou pohybu v ose pro jednotku pohybu v ose X, bude vyvolána funkce s názvem MoveAx#X#Open.

Změny okna Pohybování jednotkou pohybu v ose v manuálním režimu

Je možné přidat až do 4 tlačítek oknu pro pohybování jednotkou pohybu v ose. Musí být definovány funkce GPL s něměným názvem MoveAx#NavezJednotkypohybuose#BUTTONtext, a to ve stejné skupině nebo podskupině, ve které je definována příslušná jednotka pohybu v ose. NavezJednotkypohybuose představuje název příslušné jednotky pohybu v ose a text představuje text, který se zobrazí na tlačítku. Text může obsahovat znak '&' kvůli zavedení rychlé volby z klávesnice. Když text začíná číslem v rozsahu od 1 do 4, toto číslo bude považováno za polohu, do které bude vloženo tlačítko v okně pohybování jednotkou pohybu v ose. Text tlačítka může být přeložen zavedením DEFMSG do skupiny, ve které se nachází jednotka pohybu v ose, která má v úloze identifikátora MOVEAX#BUTTONtext. Stisknutí uživatelsky přizpůsobeného tlačítka bude mít za následek provedení přiřazené funkce GPL. Nebude provedeno žádné čekání na konec provádění ani žádné ověření zahájení provádění funkce.

10.2.2 Standardní funkce pohybu a kalibrace

Následují standardní funkce používané ovládacím panelem pohybu jednotky pohybu v ose a ovládacím panelem kalibrace.

Funkce se mění v závislosti na druhu příslušné jednotky pohybu v ose: počítání, kroková apod. Tyto funkce mohou být [uživatelsky přizpůsobené](#).

Standardní funkce manuálně ovládaného pohybu

Pohyb do absolutní polohy

```

; pro jednotky krokového pohybu v ose
Function AbsMovement
    param axisname as axis

```



```

param speed as float
param position as double

ifstill      axisname goto move
fret

move:
setvel      axisname, speed
movabs     axisname, position
waitstill  axisname
fret

; pro všechny ostatní druhy jednotek pohybu v ose
Function AbsMovement
param axisname as axis
param speed as float
param position as double

iftarget    axisname goto move
ifstill     axisname goto move
fret

move:
setvel      axisname, speed
movabs     axisname, position
waitstill  axisname
fret

```

Inkrementální pohyb

```

; pro jednotky krokového pohybu v ose
Function IncMovement
param axisname as axis
param speed as float
param position as double

ifstill      axisname goto move
fret

move:
setvel      axisname, speed
movinc     axisname, position
waitstill  axisname
fret

; pro všechny ostatní druhy jednotek pohybu v ose
Function IncMovement
param axisname as axis
param speed as float
param position as double

iftarget    axisname goto move
ifstill     axisname goto move
fret

move:
setvel      axisname, speed
movinc     axisname, position
waitstill  axisname
fret

```

Nastavení polohy

```

; pro jednotky pohybu v ose počítání
Function PositionSet
param axisname as axis
param position as double

setquote    axisname, position

```

```

fret

; pro jednotky krokového pohybu v ose
Function PositionSet
  param axisname as axis
  param position as double

  ifstill      axisname goto set
  fret
set:
  setquote    axisname, position
  fret

; pro všechny ostatní druhy jednotek pohybu v ose
Function PositionSet
  param axisname as axis
  param position as double

  iftarget    axisname goto set
  ifstill     axisname goto set
  fret
set:
  setquote    axisname, position
  fret

```

Nastavení stavu free

```

Function FreeAxis
  param axisname as axis

  free      axisname
  fret

```

Nastavení stavu normal

```

Function NormalAxis
  param axisname as axis

  normal    axisname
  fret

```

Standardní funkce kalibrace**Kalibrace pohybu z bodu do bodu**

```

; pro jednotky krokového pohybu v ose
Function PPCalibration
  param axisname as axis
  param speed as float
  param PosPosition as double
  param NegPosition as double
  param WaitTime as float

  setvel    axisname, speed
loop:
  movabs    axisname, PosPosition
  waitstill axisname
  delay     WaitTime
  movabs    axisname, NegPosition
  waitstill axisname
  delay     WaitTime
  goto     loop
  fret

; pro všechny ostatní druhy jednotek pohybu v ose
Function PPCalibration
  param axisname as axis

```

```

param speed as float
param PosPosition as double
param NegPosition as double
param WaitTime as float

setvel      axisname, speed
loop:
movabs     axisname, PosPosition
waitstill  axisname
ifquotet   axisname,<>,PosPosition goto exit
delay     WaitTime
movabs     axisname, NegPosition
waitstill  axisname
ifquotet   axisname,<>,NegPosition goto exit
delay     WaitTime
goto      loop
exit:
fret

```

Kalibrace při interpolaci

```

Function IntCalibration
param axisname as axis
param speed as float
param PosPosition as double
param NegPosition as double
param WaitTime as float

setveli    axisname, speed
loop:
linearabs  axisname, PosPosition
waitstill  axisname
ifquotet   axisname,<>,PosPosition goto exit
delay     WaitTime
linearabs  axisname, NegPosition
waitstill  axisname
ifquotet   axisname,<>,NegPosition goto exit
delay     WaitTime
goto      loop
exit:
fret

```

10.2.3 Funkce OnUIEnd#

Funkce „OnUIEnd#“ bude provedena, je-li přítomna, softwarem Albatros před ukončením všech úloh modulu. Musí být definována v souboru funkcí meziskupiny. Funkce „OnUIEnd#“ se vyznačuje maximální dobou realizace 2 sekundy, po jejímž uplynutí software Albatros ukončí všechny úlohy.

Maximální dobu čekání na ukončení provádění této funkce lze konfigurovat v Tpa.ini, v sekci [Albatros], v položce Timeout.OnUIEnd=*hodnota*, kde *hodnota* je vyjádřena v milisekundách a nemůže být větší než 60000.

10.2.4 Funkce OnUIPlugged#

Funkce OnUIPlugged# bude provedena, když je např. při zapnutí výrobního zařízení třeba vědět, zda Albatros komunikoval se vzdáleným modulem. Tato funkce musí být definována v meziskupině.

10.2.5 Funkce OnUIUnPlugged#

Funkce „OnUIUnplugged#“ bude provedena před ukončením provádění softwaru Albatros (a tedy dříve, než se Albatros odpojí od modulu). Tato funkce musí být definována v meziskupině. Albatros provede tuto funkci maximálně do 2 sekund, během kterých budou přečteny:

- chyby cyklu

- chyby systému
- hlášení

Na konci provádění dojde k zavření softwaru Albatros.

Maximální dobu čekání na ukončení provádění této funkce lze konfigurovat v Tpa.ini, v sekci [Albatros], v položce Timeout.OnUIUnplugged=*hodnota*, kde *hodnota* je vyjádřena v milisekundách a nemůže být větší než 60000.

10.3 Instrukce

10.3.1 Konvence

Níže uvedené strany jsou zorganizovány v podobě listů a popisují ohledně každé instrukce:

- syntax
- popis argumentů: datový druh a možné hodnoty
- popis činnosti
- případné poznámky
- případné příklady

Instrukce stejného druhu jsou seskupeny kvůli usnadnění pochopení a konzultace.

10.3.2 Druh instrukcí jazyka GPL

Jazyk je tvořen instrukcemi, které mohou být seskupeny níže uvedeným způsobem:

Instrukce pro správu Vstupů/Výstupů

GETFEED	čte feed rate override
INPANALOG	čte analogový vstup
INPFLAGPORT	čte port příznaku
INPPORT	čte digitální port
MULTIINPPORT	čte až do 4 portů výstupu
MULTIOUTPORT	nastavuje až do 4 portů výstupu
MULTISETFLAG	nastaví více příznaků na 1
MULTISETFLAG	nastaví více výstupů na 1
MULTIRESETFLAG	nastaví více příznaků na 0
MULTIRESETOUT	nastaví více výstupů na 0
MULTIWAITFLAG	čeká na stav bitu příznaku nebo přepínače příznaku
MULTIWAITINPUT	čeká na stav více vstupů
OUTANALOG	mění analogový výstup
OUTFLAGPORT	mění port příznaku
OUTPORT	mění digitální port
RESETFLAG	nastaví příznak na 0
RESETOUT	nastaví výstup na 0
SETFLAG	nastaví příznak na 1
SETOUT	nastaví výstup na 1
WAITFLAG	čeká na stav bitu příznaku nebo přepínače příznaku
WAITINPUT	čeká na stav vstupu
WAITPERSISTINPUT	čeká na perzistentní stav vstupu

Instrukce pro správu Jednotek pohybu v osách

CHAIN	zřetězí jednu jednotku pohybu v ose s jinou
CIRCABS	absolutní kruhová interpolace
CIRCINC	inkrementální kruhová interpolace
CIRCLE	vytvoří kruh
COORDIN	pohyb koordinovaných jednotek pohybu v ose
DISABLECORRECTION	zakazuje lineární korekci pro uvedenou jednotkupohybuose
EMERGENCYSTOP	nucené nouzové zastavení jednotky pohybu v ose
ENABLECORRECTION	aktivuje lineární korekci pro uvedenou jednotkupohybuose
ENDMOV	ukončuje pohyb jednotky pohybu v ose
FASTREAD	rychlé čtení poloh jednotek pohybu v osách
FREE	uvede jednotku pohybu v ose do stavu free
HELICABS	absolutní spirálová interpolace

HELICINC	inkrementální spirálová interpolace
JERKCONTROL	aktivuje nebo zakazuje kontrolu pohybů při interpolaci
JERKSMOOTH	plynule spojuje zrychlení a rychlosti, profily rychlosti sledované jednotkami
LINEARABS	pohyb jednotek pohybu v osách v absolutním režimu
LINEARINC	pohyb jednotek pohybu v osách v inkrementálním režimu
MOVABS	absolutní lineární interpolace
MOVINC	inkrementální lineární interpolace
MULTIABS	lineární interpolace více jednotek pohybu v osách v absolutním režimu
MULTIINC	lineární interpolace více jednotek pohybu v osách v inkrementálním režimu
NORMAL	zruší stav free jednotky pohybu v ose
RESRIFLOC	obnoví počáteční vztažný systém
SETINDEXINTERP	přiřadí jednotce pohybu v ose proměnnou pro počítání provedených bloků interpolace
SETLABELINTERP	přiřadí jednotce pohybu v ose proměnnou pro identifikaci bloku přesunu
SETPFLY	okamžité vynulování
SETPFLYCHAINSTRAT	kontroluje chování jednotky pohybu v ose slave v případě instrukce setpfly na jednotce master
SETPZERO	vynulování na značce nuly
SETPZEROCCHAINSTRAT	kontroluje chování jednotky pohybu v ose slave v případě instrukce setpzero na jednotce master
SETQUOTE	nastavení polohy
SETQUOTECHAINSTRAT	kontroluje chování jednotky pohybu v ose slave v případě instrukce setquote na jednotce master
SETRIFLOC	nastavuje prostorový vztažný systém
SETTOLERANCE	nastavuje hodnoty tolerance pro lineární interpolaci
START	obnovuje pohyb jednotky pohybu v ose
STARTINTERP	nuceně nastavuje zahájení interpolace
STOP	pozastavuje pohyb jednotky pohybu v ose
SWITCHENC	umožňuje provést výměnu snímače impulzů za snímač impulzů jiné jednotky pohybu v ose
WAITACC	čekání na zrychlení jednotky pohybu v ose
WAITCOLL	čekání na překročení polohy jednotkou pohybu v ose, od které začíná ověřování přítomnosti kolize
WAITDEC	čekání na zpomalení jednotky pohybu v ose
WAITREG	čekání na režim jednotky pohybu v ose
WAITSTILL	čekání na rovnost finální polohy s cílovou polohou jednotky pohybu v ose
WAITTARGET	čekání na dosažení cílové polohy jednotky pohybu v ose
WAITWIN	čekání na dosažení okna jednotkou pohybu v ose

Instrukce pro správu Parametrů jednotky pohybu v ose Čtení/Zápis

DEVICEID	zapisuje logickou adresu přiřazenou zařízení
GETAXIS	čtení jednoho nebo více dat jednotky pohybu v ose

Pohyb z bodu do bodu

SETACC	nastavení zrychlení
SETDEC	nastavení zpomalení
SETDERIV	nastavení koeficientu derivačního účinku
SETFEED	nastavení feed rate pohybu z bodu do bodu
SETFEEDF	nastavení feed forward
SETFEEDFA	nastavení feed forward zrychlení
SETINTEG	nastavení koeficientu integračního účinku
SETMULTIFEED	nastavení procentuální hodnoty feed rate override uvedených jednotek pohybu v osách
SETPROP	nastavení koeficientu proporcionálního účinku
SETSLOPE	nastavení druhu rampy v rychlých pohybech
SETVEL	nastavení rychlosti

Interpolovaný pohyb

LOOKAHEAD	nastavení lookahead interpolace
SETACCI	nastavení zrychlení pro interpolaci
SETACCLIMIT	aktive a zakázání automatického výpočtu rychlosti režimu interpolace
SETACCSTRATEGY	volba druhu zrychlení
SETAXPARTYPE	změna používané sady parametrů jednotky pohybu v ose

SETCONTORNATURE	nastavení úhlu obrubování
SETDECI	nastavení zpomalení pro interpolaci
SETDERIVI	nastavení koeficientu derivačního účinku interpolace
SETFEEDFAI	nastavení feed forward zrychlení v interpolaci
SETFEEDI	nastavení feed rate v interpolaci
SETFEEDFI	nastavení feed forward v interpolaci
SETINTEGI	nastavení koeficientu integračního účinku interpolace
SETPROPI	nastavení koeficientu proporcionálního účinku interpolace
SETSLOPEI	nastavení druhu rampy v interpolovaných pohybech
SETSLOWPARAM	změna parametrů potřebných pro výpočet rychlosti zpomalení v případě, že je aktivní funkčnost zpomalení při obrubování
SETVELI	nastavení rychlosti interpolace
SETVELILIMIT	nastavení jednotlivých komponentů rychlosti specifikované jednotky pohybu v ose

Koordinovaný pohyb

SETFEEDCOORD	nastavení procentuální hodnoty maximální okamžité změny feed rate jednotky pohybu v ose
SETOFFSET	nastavení posunu polohy

Zřetězený pohyb

RATIO	nastavení poměru zřetězení jednotky pohybu v ose slave vůči své jednotce master
SETDYNRATIO	dynamickým způsobem mění poměr zřetězení během pohybu jednotky pohybu v ose master

Všeobecné parametry

DYNLIMIT	aktivace nebo zakázání ověření překročení limitů jednotky pohybu v ose dynamickým způsobem
ENABLESTARTCONTROL	aktivace a nastavení časového limitu pro kontrolu chybějícího rozjezdu nebo náhlého zastavení jednotky pohybu v ose
NOTCHFILTER	nastavení mezní frekvence Notchova filtru pro uvedenou jednotku pohybu v ose
RESLIMNEG	zakázání záporného limitu jednotky pohybu v ose
RESLIMPOS	zakázání kladného limitu jednotky pohybu v ose
SETADJUST	nastavení adjust jednotky pohybu v ose
SETBACKLASH	snížení nebo odstranění následků mechanické vůle na dráhu pohybu jednotky pohybu v ose
SETBIGWINFACTOR	změna faktoru násobení pro výpočet velkého okna na zvolené jednotce pohybu v ose
SETDEADBAND	nastavení parametrů minimálního napětí pro uvedenou jednotku pohybu v ose
SETENCLIMIT	změna limitů chybného zapojení snímače impulzů
SETINDEXEN	aktivace nebo zakázání vynulování polohy v místě značky nuly
SETINTEGRIME	nastavení počtu vzorků chyby smyčky, které byly použity pro výpočet integrační složky
SETIRMP	nastavení rychlosti zahájení rampy
SETLIMNEG	nastavení záporného limitu jednotky pohybu v ose
SETLIMPOS	nastavení kladného limitu jednotky pohybu v ose
SETMAXER	nastavení maximální hodnoty tolerovaného sledování
SETMAXERNEG	nastavení maximální hodnoty tolerovaného sledování (záporný směr)
SETMAXERPOS	nastavení maximální hodnoty tolerovaného sledování (kladný směr)
SETMAXERTYPE	nastavení druhu zkoušky na chybě servořízení
SETPHASESINV	aktivace nebo zakázání inverze fází na uvedené jednotce pohybu v ose
SETREFINV	aktivace nebo zakázání inverze řídicího signálu rychlosti na uvedené jednotce pohybu v ose
SETRESOLUTION	změna rozlišení jednotky pohybu v ose

Instrukce pro správu Počítadla

DECOUNTER	snižuje hodnotu počítadla
INCOUNTER	zvyšuje hodnotu počítadla

[SETCOUNTER](#) nastavuje počítadlo

Instrukce pro správu Časovačů

[HOLDTIMER](#) blokuje časovač
[SETTIMER](#) nastavuje časovač
[STARTTIMER](#) spouští časovač

Instrukce pro správu Matric

[CLEAR](#) vynulování proměnné, vektoru, matrice
[FIND](#) vyhledání prvku
[FINDB](#) vyhledání prvku ve vektoru nebo matici, které jsou uspořádány ve vzestupném pořadí
[LASTELEM](#) poslední prvek vektoru nebo matrice
[LOCAL](#) deklarování proměnné, vektoru, místní matrice
[MOVEMAT](#) kopíruje jeden řádek matrice do jiného řádku
[PARAM](#) deklarování parametru funkce
[SETVAL](#) změna proměnné
[SORT](#) uspořádání vektoru nebo matrice

Instrukce pro správu Řetězců

[ADDSTRING](#) provázání dvou řetězců
[CONTROLCHAR](#) nastavení kontrolního znaku do proměnné řetězec
[LEFT](#) vyjmutí prvních znaků
[LEN](#) čtení délky řetězce
[MID](#) vyjmutí některých znaků
[RIGHT](#) vyjmutí posledních znaků
[SEARCH](#) vyhledání řetězce
[SETSTRING](#) změna proměnné druhu řetězec
[STR](#) konverze z čísla na řetězec
[VAL](#) konverze z řetězce na číslo

Instrukce pro správu Komunikací

[CLEARRECEIVE](#) vyprazdňuje seznam RECEIVE požadavek na uspokojení
[COMCLEARRXBUFFER](#) vyprazdňuje přijímací vyrovnávací paměť sériového portu
[COMCLOSE](#) zavírá sériový port
[COMGETERROR](#) čte chybový kód
[COMGETRXCOUNT](#) čte počet bytů přítomných v přijímací vyrovnávací paměti
[COMOPEN](#) otevírá sériový port
[COMREAD](#) čte sériový port
[COMREADSTRING](#) čte řetězec ze sériového portu
[COMWRITE](#) zapisuje řetězec na sériový port
[COMWRITESTRING](#) zapisuje řetězec na sériový port
[RECEIVE](#) příjem dat zvenčí
[SEND](#) odesílání dat ven
[SENDIPC](#) odešle informace IPC
[WAITIPC](#) vyčká na příchod informace IPC
[WAITRECEIVE](#) příjem dat zvenčí s čekáním

Matematické Instrukce

[ABS](#) absolutní hodnota
[ADD](#) součet
[AND](#) binární součin - AND
[ARCCOS](#) arkus kosinus
[ARCSIN](#) arkus sinus
[ARCTAN](#) arkus tangens
[COS](#) kosinus
[DIV](#) dělení
[EXP](#) exponenciální funkce
[EXPR](#) řešení matematických výrazů
[LOG](#) přirozený logaritmus
[LOGDEC](#) logaritmus se základem 10

MOD	modul
MUL	násobení
NOT	binární negace - NOT
OR	binární součet - OR
RANDOM	vytvoření náhodného čísla
RESETBIT	nastavení bitu na 0
ROUND	zaokrouhlení
SETBIT	nastavení bitu na 1
SHIFTL	otočení bitu doleva
SHIFTR	otočení bitu doprava
SIN	sinus
SQR	druhá odmocnina
SUB	odečítání
TAN	tangens
TRUNC	useknutí
XOR	exkluzivní binární součet - XOR

Instrukce pro správu Multitaskingu

ENDMAIL	signalizuje konec provádění příkazu
ENDREALTIMETASK	ukončí úlohu prováděnou v reálném čase (real-time)
ENDTASK	ukončí úlohu
GETPRIORITYLEVEL	čte úroveň priority aktuální úlohy
GETREALTIME	vrací hodnotu času uplynulého od zahájení provádění v reálném čase (real-time) jednotek pohybu v osách
GETREALTIMECOUNT	vrací počet uplynulých RealTime
HOLDTASK	pozastaví provádění úlohy
RESUMETASK	obnoví provádění úlohy
SENDMAIL	odešle příkaz do poštovní schránky 'mail'
SETPRIORITYLEVEL	nastaví úroveň priority pro aktuální úlohu
STARTREALTIMETASK	zahájí úlohu real-time
STARTTASK	zahájí provádění úlohy
STOPTASK	pozastaví provádění úlohy a zastavení pohyb přiřazených jednotek pohybu v osách
WAITMAIL	přijme příkaz z poštovní schránky 'mail'
WAITTASK	vyčká na dokončení provádění úlohy

Instrukce pro správu Toků

CALL	vyvolání podprogramu
DELONFLAG	zakázání správy nouzového stavu na základě bitu příznaku nebo přepínače příznaku
DELONINPUT	zakázání správy nouzového stavu na základě digitálního vstupu
ENDREP	konec opakování bloku s instrukcí REPEAT
FCALL	vyvolání funkce
FOR	rozšíření instrukce REPEAT
FRET	návrat z vyvolání funkce
GOTO	skok na návěstí
IF	ověření proměnné
IFACC	ověření probíhajícího zrychlování jednotky pohybu v ose
IFAND	ověření operace AND
IFBIT	ověření bitu
IFBLACKBOX	zkouška aktivního stavu zaznamenávání aktivity logických zařízení.
IFCHANGEVEL	ověření probíhající změny rychlosti jednotky pohybu v ose
IFCOUNTER	ověření počítadla
IFDEC	ověření probíhajícího zpomalování jednotky pohybu v ose
IFDIR	ověření směru jednotky pohybu v ose
IFERRAN	ověření chyby smyčky
IFERROR	ověření aktivní chyby cyklu
IFFLAG	ověření příznaku
IFINPUT	ověření vstupu
IFMESSAGE	ověření aktivního hlášení
IFOR	ověření operace OR
IFOUTPUT	ověření výstupu
IFQUOTER	ověření reálné polohy
IFQUOTET	ověření cílové polohy
IFRECEIVED	ověření příjmu dat
IFREG	ověření probíhajícího režimu jednotky pohybu v ose

IFSAME	ověření vazby dvou argumentů na stejná data
IFSTILL	ověření zastavení jednotky pohybu v ose
IFSTR	ověření řetězce
IFTARGET	ověření jednotky pohybu v ose v cílové poloze
IFTASKHOLD	ověření pozastavení paralelní funkce
IFTASKRUN	ověření provádění paralelní funkce
IFTIMER	ověření časovače
IFVALUE	ověření proměnné
IFVEL	ověření rychlosti jednotky pohybu v ose
IFWIN	ověření jednotky pohybu v ose v okně
IFXOR	ověření operace XOR
NEXT	konec opakování bloku s FOR
ONERRSYS	nastavení vyvolání funkce na základě chyby systému
ONFLAG	nouzový stav na základě bitu příznaku nebo přepínače příznaku
ONINPUT	nouzový stav na základě digitálního vstupu
REPEAT	opakování bloku instrukcí
RET	návrat z podprogramu
SELECT	vícenásobná volba se skokem
TESTIPC	ověření přítomnosti informace IPC
TESTMAIL	ověření a příjem příkazu

Různé Instrukce

CLEARERRORS	vymazání všech chyb cyklu modulu
CLEARMESSAGES	vymazání všech hlášení modulu
DEFMSG	definování hlášení skupiny
DELAY	zablokování aktuální funkce po určené časové období
DELERROR	vymazání předchozí chyby cyklu
DELMESSAGE	vymazání předchozího hlášení
ERROR	odeslání chyby cyklu do PC
IFDEF/ELSEDEF/ENDIF	ověření podmíněné kompilace
MESSAGE	odeslání hlášení do PC
SYSFAULT	vypnutí signálu SYSOK
SYSOK	zapnutí signálu SYSOK
TYPEOF	druh argumentu
WATCHDOG	slouží k aktivaci, aktualizaci a zrušení sledovacího zařízení (watchdog) z GPL na hardwarovém modulu TMSWD

Instrukce pro správu MECHATROLINK-II

MECCOMMAND	odeslání příkazu pro pohon jednotky pohybu v ose
MECGETPARAM	čtení parametru uvedené jednotky pohybu v ose
MECGETSTATUS	čtení hodnot STATUS, ALARM a IO_MON
MECSETPARAM	zápis parametru uvedené jednotky pohybu v ose

Instrukce pro správu standardních polních sběrnic

AXCONTROL	nastavení hodnoty pro ControlWord
AXSTATUS	vrácení hodnoty obsažené ve StatusWord
CNBYDEVICE	vrací číslo karty a uzel zařízení
READDICTIONARY	čtení obsahu objektu slovníku
WRITEDICTIONARY	zápis obsahu objektu slovníku

Instrukce pro správu EtherCAT

ACTIVATEMODE	nastavení provozního režimu
ECATGETREGISTER	vrácení obsahu registru ESC (EtherCAT Slave Controller).
ECATSETREGISTER	zápis obsahu registru ESC (EtherCAT Slave Controller).
GETPDO	vrácení objektu uvnitř PDO sítě EtherCAT
SETEOF	aktivace nebo zrušení položky Sniffer
SETPDO	nastavení objektu uvnitř PDO sítě EtherCAT

Instrukce pro správu sběrnice CAN

GETCNSTATE	vrácení stavu protokolu NMT pro uzel karty
----------------------------	--

[GETSDOERROR](#)
[GETMNSTATE](#)
[RECEIVEPDO](#)
[SENDPDO](#)
[SETNMTSTATE](#)

vrácení poslední chyby, ke které došlo
 vrácení stavu protokolu NMT pro uzel master karty
 slouží k přečtení obsahu asynchronního PDO
 slouží k zápisu obsahu asynchronního PDO
 nastavení stavu protokolu NMT pro uzel master karty

Instrukce pro Simulaci

[DISABLE](#)
[DISABLEFORCEDINPUT](#)
[ENABLE](#)
[ENABLEFORCEDINPUT](#)
[RESETFORCEDINPUT](#)
[SETFORCEDANALOG](#)
[SETFORCEDINPUT](#)
[SETFORCEDPORT](#)

zakazuje jednu nebo více jednotek pohybu v osách
 zakazuje nucené nastavení vstupů
 aktivuje jednu nebo více jednotek pohybu v osách
 aktivuje nucené nastavení vstupů
 provádí nucené nastavení vstupu na OFF
 provádí nucené nastavení analogového vstupu
 provádí nucené nastavení vstupu na ON
 provádí nucené nastavení portu vstupů

Instrukce pro funkci „Blackbox“

[ENDBLACKBOX](#)
[PAUSEBLACKBOX](#)
[STARTBLACKBOX](#)

ukončení funkce zaznamenávání
 pozastavení funkce zaznamenávání
 zahájení funkce zaznamenávání

Instrukce pro správu ISO

[ISOG0](#)
[ISOG1](#)
[ISOG9](#)
[ISOG90](#)
[ISOG91](#)
[ISOG93](#)
[ISOG94](#)
[ISOG216](#)
[ISOG217](#)

nastavení rychlého pohybu
 nastavení interpolovaného pohybu
 nastavení nuceného zastavení pohybu
 nastavení interpretace poloh jako absolutních poloh
 nastavení interpretace poloh jako relativních poloh
 nastavení interpretace rychlosti jako obrácené hodnoty času
 nastavení interpretace rychlosti jako měrné jednotky za minutu
 definování matic pro parametrizaci stroje
 popisování fyzických jednotek pohybu v ose a virtuálních jednotek pohybu v ose, které tvoří stroj
 uvolnění od pohybu ISO
 volba indexů matic parametrizace
 nastavení některých parametrů charakterizujících plynulosti interpolovaného pohybu ISO
 nastavení jednotlivých výrazů obrácené a přímé kinematiky

[ISOM2](#)
[ISOM6](#)
[ISOSETPARAM](#)
[KINEMATICEXPR](#)

10.3.3 Vstup/Výstup

GETFEED

Syntaxe

GETFEED **promenna**

Argumenty

promenna feed rate

Popis

Slouží ke kopírování hodnoty feed rate, přečtené z karty VST./VÝST. vzdáleného modulu, do uvedené parametru **promenna**. Hodnota feedrate je v rozsahu od 0 do 100 a jedná se o procentuální hodnotu. Působí na analogový vstup, který není viditelný v konfiguraci. Všechny karty TPA, které provádějí správu feed rate, jsou vybaveny specifickým konektorem.

INPANALOG

Syntaxe

INPANALOG **nazevanalogovehovstupu, promenna**

Argumenty

nazevanalogovehovstupu název zařízení druhu analogový vstup
promenna proměnná

Popis

Kopíruje hodnotu analogového vstupu určeného parametrem **nazevanalogovehovstupu** do proměnné určené parametrem **promenna**.

INPFLAGPORT

Syntaxe

INPFLAGPORT **nazevportupriznaku, promenna**

Argumenty

nazevportupriznaku název zařízení druhu port příznaku
promenna proměnná

Popis

Kopíruje stav portu příznaku definovaného argumentem **nazevportupriznaku** do proměnné určené argumentem **promenna**.

Port příznaku je viděn jako maska bitů. Každému portu příznaku je přiřazen jeden bit. Když se příznak nachází ve stavu „ON“, odpovídající bit je nastaven na 1.

INPPORT

Syntaxe

INPPORT **nazevportu, promenna**

Argumenty

nazevportu název zařízení druhu port vstupu
promenna proměnná integer nebo char

Popis

Kopíruje stav portu vstupu **nazevportu** do proměnné definované argumentem **promenna**.

Port vstupu je viděn jako maska bitů. Když se port nachází ve stavu „ON“, odpovídající bit je nastaven na 1.

MULTIINPPORT

Syntaxe

MULTIINPPORT **port1[,...,port4],proměnná**

Argumenty

port1 poskytuje bity od 0 do 7
port2 poskytuje bity od 8 do 15
port3 poskytuje bity od 16 do 23
port4 poskytuje bity od 24 do 31
proměnná proměnná integer, která přijímá porty vstupů

Popis

Čte až do čtyř vstupních portů současně a zapisuje je do **proměnné**. Čtení portů probíhá atomickým způsobem, což zajišťuje, že čtení bude provedeno v rámci stejného provádění v reálném čase (real-time). Port1 odpovídá nejnižšímu bytu, zatímco port4 odpovídá nejvyššímu bytu.

Port 4	Port 3	Port 2	Port 1
31	23	15	7
0 bit			

MULTIOUTPORT

Syntaxe

MULTIOUTPORT **hodnota, nazevportu1[,...,nazevportu4]**

Argumenty

hodnota číslo nebo proměnná integer, určená k zápisu do portů výstupu
nazevportu1 přijímá bit od 0 do 7
nazevportu2 přijímá bit od 8 do 15
nazevportu3 přijímá bit od 16 do 23
nazevportu4 přijímá bit od 24 do 31

Popis

MULTIWAITFLAG

Syntaxe

MULTIWAITFLAG	maska, priznak1[, ..., priznak32], stav [, casovylimit [, GOTO navesti]]
MULTIWAITFLAG	maska, priznak1[, ..., priznak32], stav [, casovylimit [, CALL nazevpodprogramu]]
MULTIWAITFLAG	maska, priznak1[, ..., priznak32], stav [, casovylimit [, nazevfunkce]]

Argumenty

maska	konstanta nebo proměnná. Maska zainteresovaných příznaků
priznak1[,...priznak32]	název zařízení druhu příznak
stav	přednastavená konstanta. Příпустnými hodnotami jsou: - ON stav zapnutého příznaku - OFF stav vypnutého příznaku
casovylimit	konstanta nebo proměnná. Jedná se o mezní hodnotu doby čekání
navesti	návěští skoku (GOTO)
nazevpodporogramu	navesti podprogramu (CALL)
nazevfunkce	název funkce

Popis

Čeká, dokud se uvedené příznaky **priznak1...priznak32** nebudou nacházet ve stavu uvedeném parametrem **stav** (ON/OFF).

Ze všech příznaků ověří ty, jejichž bit v argumentu **maska** je zapnutý (ON). Bit 0 argumentu **maska** (ten s nejnižší vahou) odpovídá bitu definovanému parametrem **priznak1**, bit 1 odpovídá bitu definovanému parametrem **priznak2** apod. až po bit definovaný parametrem **priznak32**.

Parametr **casovylimit** umožňuje nastavit časový limit odlišný od přednastaveného limitu, který je jedna sekunda.

Když je přítomno **navesti** nebo **nazevpodprogramu** nebo **nazevfunkce**, při uplynutí časového limitu program přeskočí na **navesti** nebo vyvolá **nazevpodprogramu** nebo **nazevfunkce**.

MULTIWAITINPUT

Syntaxe

MULTIWAITINPUT	maska, vstup1[, ..., vstup32], stav [, casovylimit [, GOTO navesti]]
MULTIWAITINPUT	maska, vstup1[, ..., vstup32], stav [, casovylimit [, CALL nazevpodprogramu]]
MULTIWAITINPUT	maska, vstup1[, ..., vstup32], stav [, casovylimit [, nazevfunkce]]

Argumenty

maska	konstanta nebo proměnná. Maska zainteresovaných vstupů
vstup1[,...vstup32]	název vstupu
stav	přednastavená konstanta. Příпустnými hodnotami jsou: - ON stav zapnutého vstupu - OFF stav vypnutého vstupu
casovylimit	konstanta nebo proměnná. Jedná se o mezní hodnotu doby čekání
navesti	návěští skoku (GOTO)
nazevpodporogramu	navesti podprogramu (CALL)
nazevfunkce	název funkce

Popis

Čeká, dokud se uvedené vstupy **vstup1...vstup32** nebudou nacházet ve stavu uvedeném parametrem **stav** (ON/OFF).

Ze všech vstupů ověří ty, jejichž bit v argumentu **maska** je zapnutý (ON). Bit 0 argumentu **maska** (ten s nejnižší vahou) odpovídá bitu definovanému parametrem **vstup1**, bit 1 odpovídá bitu definovanému parametrem **vstup2** apod. až po bit definovaný parametrem **vstup32**.

Když nebyly určeny volitelné argumenty, po uplynutí jedné sekundy od začátku provádění instrukce (přednastavený časový limit) bude automaticky vytvořeno parametrizované hlášení: „Wait inputn ON/OFF“.

Název signalizovaného vstupu je ten, který odpovídá prvnímu aktivovanému vstupu, který ještě neuspokojil stav. Když je přítomen parametr **casovylimit**, výše popsané zobrazení bude vytvořeno po uplynutí požadovaného časového limitu. Když po uplynutí časového limitu dojde ke splnění požadované podmínky, bude automaticky vytvořeno parametrizované hlášení, aby se vymazalo předtím odeslané hlášení.

Když je přítomno **navesti** nebo **nazevpodprogramu** nebo **nazevfunkce**, při uplynutí časového limitu program přeskočí na **navesti** nebo vyvolá **nazevpodprogramu** nebo **nazevfunkce**.

OUTANALOG

Syntaxe

OUTANALOG

nazevanalogovehovystupu, hodnota

Argumenty

nazevanalogovehovystupu
hodnota

název zařízení druhu analogový výstup nebo jednotka pohybu v ose
konstanta nebo proměnná

Popis

Slouží k nastavení napětí určeného parametrem **hodnota** na analogovém výstupu nebo na jednotce pohybu v ose určeném parametrem **nazevanalogovehovystupu**.

OUTFLAGPORT

Syntaxe

OUTFLAGPORT

nazevportupriznaku, hodnota

Argumenty

nazevportupriznaku
hodnota

název zařízení druhu port příznaku
konstanta nebo proměnná

Popis

Kopíruje **hodnotu** do portu příznaku uvedeného v parametru **nazevportuproznaku**.

Parametr **hodnota** je viděn jako maska bitů. Každému bitu je přiřazen jeden příznak portu. Když má bit hodnotu 1, odpovídající příznak bude nastaven do stavu „ON“.

OUTPORT

Syntaxe

OUTPORT

nazevportu, hodnota

Argumenty

nazevportu
hodnota

název zařízení druhu port výstupu
konstanta nebo proměnná, integer nebo char

Popis

Kopíruje obsah parametru **hodnota** do portu výstupu **nazevportu**.

Port výstupu je viděn jako maska bitů. Když má bit hodnotu 1, odpovídající výstup bude nastaven do stavu „ON“.

RESETFLAG

Syntaxe

RESETFLAG

nazevpriznaku

Argumenty

nazevpriznaku

název zařízení druhu příznak

Popis

Vypne (uvede do stavu OFF) příznak **nazevpriznaku**.

RESETOUT

Syntaxe

RESETOUT

nazevvystupu

Argumenty

nazevvystupu

název zařízení druhu digitálního výstupu

Popis

Slouží k vypnutí (uvedení do stavu OFF) výstupu **nazevvystupu**.

SETFLAG

Syntaxe

SETFLAG **nazevpriznaku**

Argumenty

nazevpriznaku název zařízení druhu příznak

Popis

Zapne (uvede do stavu ON) příznak **nazevpriznaku**.

SETOUT

Syntaxe

SETOUT **nazevvystupu**

Argumenty

nazevvystupu název zařízení druhu digitálního výstupu

Popis

Slouží k zapnutí (uvedení do stavu ON) výstupu **nazevvystupu**. Když je výstup monostabilního druhu, bude vypnut automaticky, po uplynutí pevně nastaveného časového limitu 200 milisekund.

WAITFLAG

Syntaxe

WAITFLAG **nazevpriznaku, stav [, casovylimit [, GOTO navesti]]**
WAITFLAG **nazevpriznaku, stav [, casovylimit [, CALL nazevpodprogramu]]**
WAITFLAG **nazevpriznaku, stav [, casovylimit [, nazevfunkce]]**

Argumenty

nazevpriznaku název zařízení druhu příznaku
stav přednastavená konstanta. Přípustnými hodnotami jsou:
- **ON** stav zapnutého příznaku
- **OFF** stav vypnutého příznaku
casovylimit konstanta nebo proměnná. Jedná se o mezní hodnotu doby čekání
navesti návěstí skoku (GOTO)
nazevpodprogramu navesti podprogramu (CALL)
nazevfunkce název funkce

Popis

Čeká, dokud příznak **nazevpriznaku** nedosáhne stavu uvedeného parametrem **stav** (ON/OFF). Když je z volitelných argumentů přítomný pouze **casovylimit**, chyba cyklu „Příznak **nazevpriznaku** čeká na **stav**“ bude vytvořena po uplynutí časového limitu. Když dojde ke splnění uvedené podmínky po uplynutí časového limitu, bude automaticky vymazána chyba cyklu, která byla předtím odeslána této úloze. Když je přítomno **navesti** nebo **nazevpodprogramu** nebo **nazevfunkce**, při uplynutí časového limitu program přeskočí na **navesti** nebo vyvolá **nazevpodprogramu** nebo **nazevfunkce** bez vytvoření jakéhokoliv automatického zobrazení.

Poznámka

Aby se zabránilo situaci čekání na příznak během pracovního cyklu, doporučuje se nastavit časový limit.

WAITINPUT

Syntaxe

WAITINPUT **nazevvstupu, stav [, casovylimit [, GOTO navesti]]**
WAITINPUT **nazevvstupu, stav [, casovylimit [, CALL nazevpodprogramu]]**
WAITINPUT **nazevvstupu, stav [, casovylimit [, nazevfunkce]]**

Argumenty

nazevvstupu název vstupu
stav přednastavená konstanta. Přípustnými hodnotami jsou:
- **ON** stav zapnutého vstupu
- **OFF** stav vypnutého vstupu
casovylimit konstanta nebo proměnná. Časový limit čekání
navesti návěstí skoku (GOTO)

nazevpodprogramu navesti podprogramu (CALL)
nazevfunkce název funkce

Popis

Čeká, dokud vstup **nazevvstupu** nedosáhne stavu uvedeného parametrem **stav** (ON/OFF). Když nejsou uvedeny volitelné argumenty, po uplynutí 20 sekund od začátku provádění instrukce bude automaticky vytvořena chyba cyklu: „Digitální vstup **nazevvstupu** čeká na **stav**“.
 Když je z volitelných argumentů přítomen pouze **casovylimit**, výše uvedené zobrazení bude vytvořeno po jeho uplynutí.
 Když dojde ke splnění uvedené podmínky po uplynutí hodnoty parametru **casovylimit**, bude automaticky vymazána chyba cyklu, která byla předtím odeslána této úloze.
 Když je přítomno **navesti** nebo **nazevpodprogramu** nebo **nazevfunkce**, při uplynutí časového limitu program přeskočí na **navesti** nebo vyvolá **nazevpodprogramu** nebo **nazevfunkce** bez vytvoření jakéhokoli automatického zobrazení.

Poznámka

Aby se zabránilo situaci čekání na vstupní signál, během pracovního cyklu se doporučuje nastavit časový limit menší, než je přednastavená hodnota (20 sekund).

Příklad

[Rutina Vynulování jednotky pohybu v ose](#)

WAITPERSISTINPUT**Syntaxe**

WAITPERSISTINPUT **nazevvstupu, stav, dobasevvanistavu [, casovylimit [, GOTO navesti]]**
WAITPERSISTINPUT **nazevvstupu, stav, dobasevvanistavu [, casovylimit [, CALL nazevpodprogramu]]**
WAITPERSISTINPUT **nazevvstupu, stav, dobasevvanistavu [, casovylimit [, nazevfunkce]]**

Argumenty

nazevvstupu název zařízení druhu digitální vstup
stav přednastavená konstanta. Přípustnými hodnotami jsou:
 - **ON** stav zapnutého vstupu
 - **OFF** stav vypnutého vstupu
dobasevvanistavu konstanta nebo proměnná
casovylimit konstanta nebo proměnná. Časový limit čekání
navesti návěstí skoku (GOTO)
nazevpodprogramu navesti podprogramu (CALL)
nazevfunkce název funkce

Popis

Čeká, dokud se vstup **nazevvstupu** nebude nacházet ve stavu určeném parametrem **stav** (ON/OFF), a zůstane v tomto stavu po dobu uvedenou v parametru **dobasevvanistavu**. Když nejsou uvedeny volitelné argumenty, po uplynutí 20 sekund od začátku provádění instrukce bude automaticky vytvořena chyba cyklu: „Digitální vstup **nazevvstupu** čeká na **stav**“.
 Když je z volitelných argumentů přítomen pouze **casovylimit**, výše uvedené zobrazení bude vytvořeno po jeho uplynutí.
 Když dojde ke splnění uvedené podmínky po uplynutí hodnoty parametru **casovylimit**, bude automaticky vymazána chyba cyklu, která byla předtím odeslána této úloze.
 Když je přítomno **navesti** nebo **nazevpodprogramu** nebo **nazevfunkce**, při uplynutí časového limitu program přeskočí na **navesti** nebo vyvolá **nazevpodprogramu** nebo **nazevfunkce** bez vytvoření jakéhokoli automatického zobrazení.

Poznámka

Aby se zabránilo situaci čekání na vstupní signál, během pracovního cyklu se doporučuje nastavit časový limit menší, než je přednastavená hodnota (20 sekund).

10.3.4 Jednotky pohybu v osách**CHAIN****Syntaxe**

CHAIN **jednotkapohybuose_master, jednotkapohybuose_slave1 [, ...jednotkapohybuose_slave5]**

Argumenty

jednotkapohybuose_master název zařízení druhu jednotka pohybu v ose, které bude fungovat jako master

jednotkapohybuose_slav1... název zařízení druhu jednotka pohybu v ose, které bude fungovat jako slave

jednotkapohybuose_slave5

Popis

Po provedení této instrukce **jednotkypohybuose_slave** (1÷5) provedou pohyby související s pohyby jednotky pohybu v ose master na základě vztahu řetězení nastaveného instrukcí **RATIO**. Výsledkem budou zřetězené pohyby z bodu do bodu i interpolované pohyby.

Jednotkapohybuose_slave1 není volitelným parametrem, ale musí být vždy definována.

Aby mohla být jednotka pohybu v ose slave zřetězena, nesmí být vázána v interpolaci a nesmí sama představovat master pro jiné jednotky.

Jednotka pohybu v ose master zase nesmí sama představovat slave pro jiné jednotky pohybu v ose.

Zřetězení může být provedeno s jednotkami pohybu v ose v cílové poloze i s jednotkami pohybu v ose v pohybu.

Pro zrušení zřetězení jednotek pohybu v ose stačí provést instrukci **NORMAL** na jednotce pohybu v ose master. Posledně uvedená operace může být provedena s jednotkami pohybu v ose v cílové poloze i s jednotkami pohybu v ose v pohybu. Když je zřetězení zrušeno během pohybu jednotek pohybu v osách, jednotka pohybu v osách typu slave provede rampu zpomalení a zastaví se.

Je možné definovat maximálně 8 jednotek pohybu v osách typu master současně.

Instrukci lze provést také s jednotkami krokového pohybu v osách (stepper) za podmínky, že jsou všechny řízeny prostřednictvím TRS-AX.

Kromě toho platí, že všechny jednotky pohybu v ose musí mít reálný, nikoli simulovaný snímač impulzů. V opačném případě bude vytvořena chyba systému „4101 Nevyhovující správa jednotky pohybu v ose nazevejednotkypohybuose“.

Viz také [RATIO](#).

Příklad

```
CHAIN X, Y ; zřetězení jednotky pohybu v ose Y s jednotkou pohybu
          v ose X
MOVINC X, 100 ; pohybují jednotkou pohybu v ose X. Y
              ; replikuje pohyb X
```

CIRCABS

Syntaxe

CIRCABS [**navesti**],**jednotkapohybuose1**, **poloha1**, **jednotkapohybuose2**, **poloha2**, **směr**, **±poloměr** [, **uhel**]

Argumenty

navesti konstanta nebo proměnná integer. Návěstí, které identifikuje blok přesunu názvy zařízení druhu jednotka pohybu v ose

jednotkapohybuose1,
jednotkapohybuose2

poloha1,**poloha2**

směr

konstanta nebo proměnná. Představuje absolutní polohu přesunu
proměnná integer. Určuje druh otáčení; přípustné jsou dvě hodnoty:
CW ve směru hodinových ručiček
CCW proti směru hodinových ručiček

poloměr

uhel

konstanta nebo proměnná. Představuje hodnotu poloměru kruhu
konstanta nebo proměnná. Představuje výchozí úhel

Popis

Kruhová interpolace 2 jednotek pohybu v osách s *absolutním přesunem* založená na naprogramovaných argumentech **poloha1** a **poloha2**.

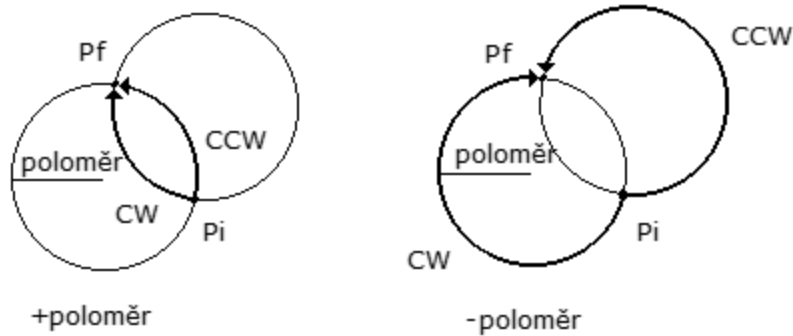
Kruh je dán výchozím bodem (aktuálním bodem), koncovým bodem, hodnotou **poloměru** a **směrem** dráhy.

Znaménko aplikované na **poloměr** umožňuje zvolit menší kruh (+poloměr) nebo větší kruh (-poloměr).

Ve specifickém případě, kdy se počáteční poloha jednotky pohybu v ose 1 shoduje s finální polohou **poloha1** a počáteční poloha jednotkypohybuose2 se shoduje s finální polohou **poloha2**, bude určen kompletní kruh: V tomto případě je třeba uvést argument **uhel** se stejným významem instrukce **CIRCLE** (ze které je třeba vycházet).

Parametr **uhel** slouží k jedinečnému určení středu kruhu se stejným významem instrukce **CIRCLE**. Bere se v úvahu pouze tehdy, když ještě před provedením instrukce **poloha1** a **poloha2** odpovídají aktuálním polohám jednotek pohybu v osách. Volitelný parametr **navesti** se používá ve spojení s instrukcí [SETLABELINTERP](#) pro jedinečnou identifikaci bloku přesunu.

Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX. V tomto případě je třeba brát v úvahu, že výrazem interpolace se má na mysli koordinovaný pohyb více jednotek pohybu v osách ovlivněný diskrétní chybou způsobenou metodou řízení jednotky pohybu v ose.



CIRCINC

Syntaxe

CIRCINC

[navesti],jednotkapohybuose1, poloha1, jednotkapohybuose2, poloha2, smer, \pm polomer [, uhel]

Argumenty

navesti
jednotkapohybuose1,
jednotkapohybuose2
poloha1, poloha2
smer

konstanta nebo proměnná integer. Návěští, které identifikuje blok přesunu názvy zařízení druhu jednotka pohybu v ose

konstanta nebo proměnná. Představuje inkrementální polohu přesunu
proměnná integer. Určuje druh otáčení; přípustné jsou dvě hodnoty:
CW ve směru hodinových ručiček

CCW proti směru hodinových ručiček

poloměr
uhel

konstanta nebo proměnná. Představuje hodnotu poloměru kruhu
konstanta nebo proměnná. Představuje výchozí úhel

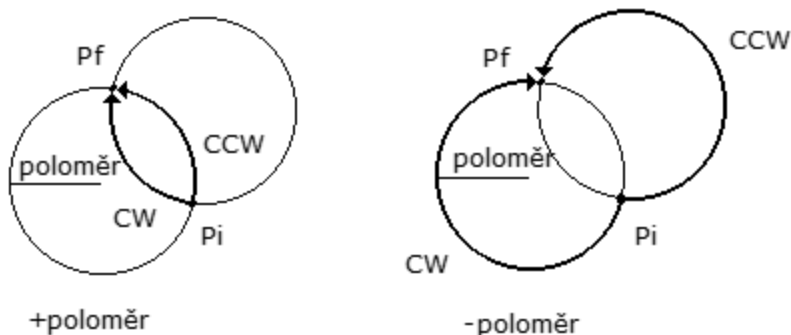
Popis

Kruhová interpolace 2 jednotek pohybu v osách s *inkrementálním přesunem* založená na naprogramovaných argumentech **poloha1** a **poloha2**.

Kruh je dán výchozím bodem (aktuálním bodem), koncovým bodem, hodnotou **poloměru** a **směrem** dráhy. Znaménko aplikované na **poloměr** umožňuje zvolit menší kruh (+poloměr) nebo větší kruh (-poloměr).

Ve specifickém případě, když $poloha1 = poloha2 = 0$, se určí kompletní kruh: V tomto případě je třeba uvést argument **uhel** se stejným významem jako instrukce [CIRCLE](#) (ze které je třeba vycházet).

Parametr **uhel** slouží k jedinečnému určení středu kruhu se stejným významem instrukce [CIRCLE](#). Volitelný parametr **navesti** se používá ve spojení s instrukcí [SETLABELINTERP](#) pro jedinečnou identifikaci bloku přesunu. Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX. V tomto případě je třeba brát v úvahu, že výrazem interpolace se má na mysli koordinovaný pohyb více jednotek pohybu v osách ovlivněný diskrétní chybou způsobenou metodou řízení jednotky pohybu v ose.



CIRCLE

Syntaxe

CIRCLE

[navesti],jednotkapohybuose1, jednotkapohybuose2, smer, polomer, uhel

Argumenty

navesti
jednotkapohybuose1,
jednotkapohybuose2
smer

konstanta nebo proměnná integer. Návěstí, které identifikuje blok přesunu názvy zařízení druhu jednotka pohybu v ose

proměnná integer. Určuje druh otáčení; přípustné jsou dvě hodnoty:
CW ve směru hodinových ručiček
CCW proti směru hodinových ručiček

poloměr
uhel

konstanta nebo proměnná. Představuje hodnotu poloměru kruhu
konstanta nebo proměnná. Představuje výchozí úhel

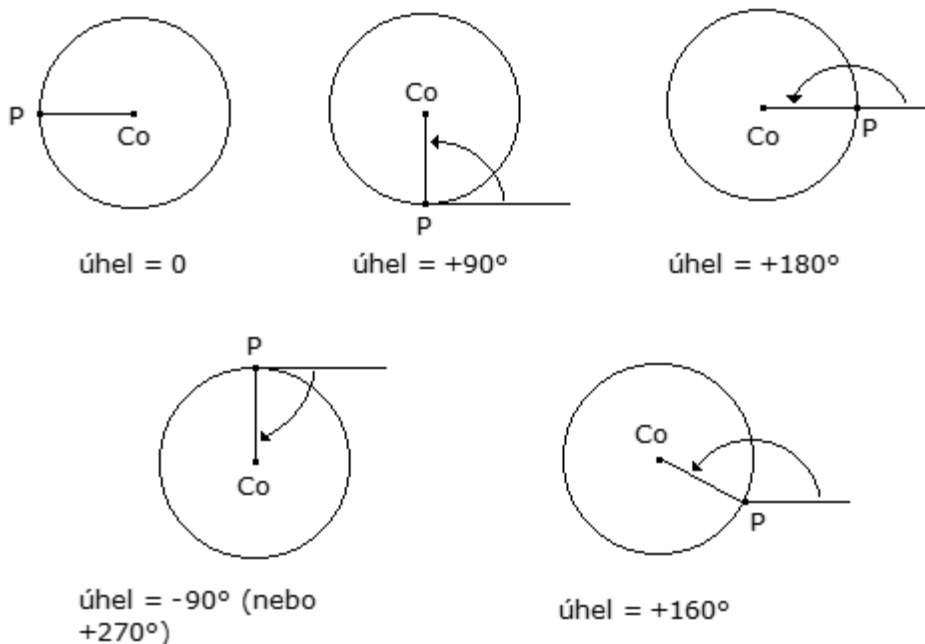
Popis

Kompletní kruhová interpolace.

Vytváří kruh s **jednotkoupohybuose1** a **jednotkoupohybuose2** v uvedeném parametru **smer**, s parametrem **polomer**, který se rovná hodnotě přiřazené na základě vnuceného výchozího parametru **uhel**. Parametr **polomer** může nabýt pouze kladných hodnot.

Parametr **uhel** je třeba uvést s kladnou trigonometrickou konvencí proti směru hodinových ručiček, přičemž je nutné vycházet z jednotkypohybuose X. Zůstane vskutku určena poloha středu Co kruhu s uvedením úhlu vytvořeného poloměrem procházejícím počátečním naprogramovaným bodem P (aktuální bod) a vodorovným směrem X+. Volitelný parametr **navesti** se používá ve spojení s instrukcí [SETLABELINTERP](#) pro jedinečnou identifikaci bloku přesunu.

Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX. V tomto případě je třeba brát v úvahu, že výrazem interpolace se má na mysli koordinovaný pohyb více jednotek pohybu v osách ovlivněný diskrétní chybou způsobenou metodou řízení jednotky pohybu v ose.



COORDIN

Syntaxe

COORDIN

matrice, hodnotadelta_t, smer, poc, konc, maska,
jednotkapohybuose1, c_sloupce_jednotkypohybuose1 [,
(jednotkapohybuose2, c_sloupce_jednotkypohybuose2) ÷
(jednotkapohybuose32, c_sloupce_jednotkypohybuose32)]

Argumenty

matrice	matrice dat
hodnotadelta_t	konstanta nebo proměnná. Základ dob
smer	přednastavená konstanta. Směr čtení dat matrice UP od nejspodnějšího po nejvyšší řádek DOWN od nejvyššího po nejspodnější řádek
poc	globální proměnná integer. Jedná se o číslo počátečního řádku
konc	globální proměnná integer. Jedná se o číslo koncového řádku
maska	maska jednotek pohybu v osách určených k aktivaci
jednotkapohybuose1 [... jednotkapohybuose32]	názvy zařízení druhu jednotka pohybu v ose
c_sloupc_jednotkapohybuose	číslo sloupce matrice vztahující se na jednotku pohybu v ose
1[... c_sloupc_jednotkapohybuose	
32]	

Popis

Tato funkce umožňuje provádět synchronizované pohyby jednotek pohybu v osách **jednotkapohybuose1**, **jednotkapohybuose2**. apod. prostřednictvím inkrementálních přesunů (mikrovektorů) definovaných **matricí** dat.

Parametry **jednotkapohybuose1** a **c_sloupc_jednotkypohybuose1** musí být vždy zadefinovány. Hodnoty obsažené v parametru **matrice** poukazují na absolutní polohy postupně dosahované jednotlivými jednotkami pohybu v osách.

Příslušné inkrementální přesuny (rozdíl poloh mezi řádkem (n) a řádkem (n-1)) budou provedeny v časovém intervalu rovnajícím se **násobku** základu dob (1 ms = Reálná doba aktualizace jednotek pohybu v osách), definovanému argumentem **hodnotaδελτα_t**, který tedy musí být vyjádřen celým číslem.

Po definování této časové hodnoty určí entita jednotlivých přesunů jednotky pohybu v ose jejich rychlost.

Tato instrukce umožňuje koordinovaný pohyb maximálně 32 jednotek pohybu v osách vytvořených technikami SPLINE podél jakékoli křivočaré dráhy v prostoru.

Instrukce je průchozí a nevyžaduje k zahájení instrukci STARTINTERP. Na jejím konci je však třeba uvést instrukci WAITSTILL kvůli čekání na správný příchod jednotek pohybu v osách do cílové polohy. Případné změny feedrate override (překrytí rychlosti dopředného řízení) musí být provedeny prostřednictvím instrukce SETFEEDI a jejich správa musí být prováděna prostřednictvím instrukce SETFEEDCOORD.

Parametr **smer** umožňuje určit směr procházení matrice a umožnit tak provedení dráhy pohybu v obou směrech.

Sloupce matrice, která má být procházena, mohou být druhu float nebo druhu double, ne však obou současně.

Kromě pohybu jednotek pohybu v osách podél hotové dráhy (zadefinované počtem řádku matrice) je možné vytvořit nekonečný pohyb použitím:

- Matrice s jediným řádkem. V tomto režimu činnosti řízení čte vždy jediný řádek matrice a aplikuje na jednotky pohybu v osách polohy, které jsou v něm uvedeny. Aby se jednotky pohybu v ose pohybovaly, bude třeba vhodně změnit řádek matrice, přednostně s použitím úlohy v reálném čase (real-time), která zaručuje synchronizaci aktualizace poloh s frekvencí aktualizace jednotek pohybu v osách. Tímto způsobem je možné realizovat elektronické vačky nebo zřetězení v poměru odlišném od 1:1. Pro aktivaci tohoto režimu činnosti bude třeba nastavit **poc = 1**, **konc = 0** a **smer = UP**. Při použití tohoto režimu činnosti NESMÍ být použita instrukce STOP.
- Matrice s více řádky. Je možné nekonečně procházet matici s cykly od prvního po poslední řádek nastavením hodnot **poc = 1**, **konc = 0** a **smer = UP**. V případě, že je třeba provést jediný řádek víceřádkové matrice, je třeba nastavit parametry **poc**, **konc** a **smer** níže uvedeným způsobem: **poc** = číslo řádku, který má být proveden, **konc** = číslo řádku, který předchází řádku, který má být proveden, **smer** = UP. V odlišném případě bude vytvořena chyba systému.

Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX.

DISABLECORRECTION

Syntaxe

DISABLECORRECTION **jednotkapohybuose** [, **jednotkapohybuose1**, ..., **jednotkapohybuose6**]

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
**jednotkapohybuose1,...,
jednotkapohybuose6]** názvy zařízení druhu jednotka pohybu v ose

Popis

Slouží k zakázání korekce pro určenou **jednotku pohybu v ose**.

Prvním parametrem je jednotka pohybu v ose, u které má být zakázána korekce; v případě, že se jedná o jediný uvedený parametr, budou zakázány všechny korekce přítomné v konfiguraci. Následující parametry

umožňují určit, které korekce zakázat; když se jeden z nich shoduje s prvním parametrem, bude zakázána samokorekce.

Ohledně podrobnějších informací viz [ENABLECORRECTION](#).

Příklad

; zakáže samotnou samokorekci pro jednotku pohybu v ose X
DISABLECORRECTION X, X

; zakáže křížovou korekci (vůči X a Y) pro jednotku pohybu v ose Z, ale nezakáže samokorekci

DISABLECORRECTION Z, X, Y

EMERGENCYSTOP

Syntaxe

EMERGENCYSTOP **jednotkapohybuose , doba**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
doba konstanta nebo proměnná integer. Doba rampy v ms

Popis

Slouží k zastavení uvedené jednotky pohybu v ose a případně i všech dalších, které jsou zainteresovány v interpolovaném pohybu. Pohyb bude zastaven se zpomalovací rampou, jejíž doba je stanovena proměnnou **[doba]**.

Když je při pohybech z bodu do bodu nastavena doba delší než doba zpomalení uvedená v konfiguraci, bude použita doba uvedená v konfiguraci.

Když je při interpolovaných pohybech nastavena doba delší než maximum dob zpomalení všech zainteresovaných jednotek pohybu v osách, bude použita maximální doba nastavená v konfiguraci.

Pohyb bude moci být obnoven instrukcí [START](#).

Instrukce nesmí být použita v případě, že je **[jednotkapohybuose]** typu Slave.

Instrukce může vytvořit níže uvedené chyby systému:

- „4101 - Nevyhovující správa jednotky pohybu v ose“, když **[jednotkapohybuose]** provádí synchronizovaný pohyb nebo multilineární interpolaci nebo pohyb ISO.
- „4105 - Instrukci nelze provést na dané jednotce pohybu v ose“, když je **[jednotkapohybuose]** jednotkou počítání pohybu v ose.
- „4399 - Parametr mimo určené rozmezí“, když je uvedená **[doba]** rovná nebo menší než 0.

ENABLECORRECTION

Syntaxe

ENABLECORRECTION **jednotkapohybuose [, jednotkapohybuose1, ..., jednotkapohybuose6]**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
jednotkapohybuose1, ..., jednotkapohybuose6] názvy zařízení druhu jednotka pohybu v ose

Popis

Slouží k aktivaci lineární korekce pro určenou **jednotku pohybu v ose**. Korekce se skládá ze samokorekce a z křížové korekce. Samokorekce je korekce reálné polohy jednotky pohybu v ose v závislosti na své poloze, zatímco křížová korekce je korekce reálné polohy jednotky pohybu v ose v závislosti na poloze ostatních jednotek pohybu v ose. Je možné definovat maximálně pět křížových korektorů.

Prvním parametrem je jednotka pohybu v ose, pro kterou má být aktivována korekce; v případě, že se jedná o jediný uvedený parametr, budou aktivovány všechny korekce přítomné v konfiguraci.

Následující parametry umožňují určit, které korekce aktivovat; když se jeden z nich shoduje s prvním parametrem, bude aktivována samokorekce.

Viz také [DISABLECORRECTION](#).

Poznámka

Aby měla instrukce efekt, korekce musí být aktivována v konfiguraci.

Příklad

; aktivuje všechny korekce určené v konfiguraci pro jednotku pohybu v ose X
ENABLECORRECTION X

; aktivuje samotnou samokorekci pro jednotku pohybu v ose X

ENABLECORRECTION X, X

; aktivuje samokorekci a křížovou korekci (vůči X a Y) pro jednotku pohybu v ose Z

ENABLECORRECTION Z, X, Y, Z

ENDMOV

Syntaxe

ENDMOV jednotka pohybu v ose [, poloha]

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
poloha konstanta nebo proměnná.

Popis

Slouží k zastavení pohybu vymezené jednotky pohybu v ose. Liší se od instrukce [STOP](#), protože pohyb je ukončen a nebude moci být obnoven případnou instrukcí [START](#). Je-li uveden parametr **poloha**, umožňuje nastavit polohu, ve které jednotka pohybu v ose ukončí svůj pohyb; v opačném případě bod, ve kterém dojde k zastavení jednotky pohybu v ose, závisí na aktuální rychlosti a na posledním naprogramovaném zpomalení. V případě potřeby za účelem dosažení koncového bodu pohybu řízení provede změnu směru pohybu jednotky pohybu v ose.

Poznámka

Parametr kóty se používá pouze v případě, že se jedná o pohyb z bodu do bodu. V případě interpolovaného pohybu dojde k zastavení pohybu jednotky pohybu v ose bez zohlednění hodnoty **polohy**.

Příklad

; zastaví se aktuální pohyb a jednotka pohybu v ose se přesune do polohy

0.0

ENDMOV X, 0.0

FASTREAD

Syntaxe

FASTREAD jednotkapohybuose1, stav, promenna1 [,jednotkapohybuose2, promenna2],[..., jednotkapohybuose8, promenna8]

Argumenty

jednotkapohybuose1... názvy zařízení druhu jednotka pohybu v ose. Jednotkapohybuose1 je
[...jednotkapohybuose8] jednotka pohybu v ose druhu master
stav přednastavená konstanta. Může nabýt níže uvedených hodnot:

ON náběžná hrana
OFF sestupná hrana
promenna1...[...promenna8] proměnná nebo prvek matrice/vektoru druhu double. Poloha uložená v paměti

Popis

Polohy uvedených **jednotekpohybuosach** jsou čteny a ukládány do paměti prostřednictvím parametru **proměnných** v okamžiku, kdy rychlý vstup **jednotkypohybuose1** (jednotka pohybu v ose Master) přepne do nastaveného stavu.

Když jsou uvedené jednotky pohybu v osách analogové, musí patřit do stejné karty (4 pro TRS-AX). Když jsou uvedené jednotky pohybu v osách digitální, signál rychlého vstupu se nachází přímo na pohonu, a proto je v případě fastread třeba připojit signál paralelně na jednotlivá zařízení.

Se sběrnici EtherCAT je pro každou jednotku pohybu v ose se záměnou snímačů impulzů (viz příkaz [SWITCHENC](#)) maximální počet jednotek pohybu v osách snížen o jeden. Po překročení tohoto limitu bude vytvořena chyba systému „4400 Příliš mnoho aktivních jednotek pohybu v ose ve FASTREAD“. Příkladový snímač impulzů musí být dále připojen k rozšíření TRS-AC-E na TRS-CAT. V opačném případě bude vytvořena chyba systému „4375 FASTREAD provedena na jednotkách pohybu v osách odlišných karet“

Když jsou uvedené jednotky pohybu v osách nakonfigurovány na sběrnici EtherCAT, musí patřit do stejného pohonu.

Instrukce je ukončena, když dojde k přepnutí vstupu do uvedeného **stavu (ON/OFF)**.

Při provedení instrukce STOP před přepnutím rychlého vstupu tyto instrukce zůstanou aktivní a budou obnoveny po instrukci START.

Je možné aktivovat více rychlých čtení současně na stejné kartě jednotek pohybu v osách.

Během provádění instrukce není možné současně provádět instrukce [SETPZERO](#) a [SETPLY](#) na stejné jednotce pohybu v ose, je-li připojena ke kartám se sběrnici MECHATROLINK-II.

Poznámka

Rychlý vstup pro jednotky pohybu v osách digitálního druhu na kartě se sběrníci MECHATROLINK-II je přítomen na vstupu **EXTI2** a není třeba jej konfigurovat ve virtuálním-fyzickém deklarování. Rychlé vstupy digitálních jednotek pohybu v osách MECHATROLINK-II musí být zkratovány, protože ukládání polohy jednotky pohybu v ose se provádí pouze v návaznosti na vlastní rychlý vstup.

FREE**Syntaxe**

FREE **jednotkapohybuose** [, **napeti**]

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
napeti konstanta float nebo proměnná float. Řídící napětí

Popis

Uvede **jednotkupohybuose** do stavu „otevřené smyčky“ (Free) zrušením *kontroly polohy*. Když je **jednotkapohybuose** slave v provázání s jinými jednotkami pohybu v osách, omezení bude zrušeno a pohyb **jednotkypohybuose** bude zastaven. V případě uvedení parametru **napeti** se řídící napětí jednotky pohybu v ose bude rovnat tomuto parametru.

Tato instrukce může být použita v případě jednotek měření v ose, které slouží pro zajišťování poloh, nebo v případě jednotek pohybu v osách, jejichž pohyb může být nuceně ovládn externími mechanickými součástmi, které by měnily jejich polohu.

Během provozu bude poloha jednotky pohybu v ose zaznamenávána a aktualizována a je tedy možné provádět polohování jednotky pohybu v ose v absolutním režimu po opětovné aktivaci kontroly polohy (instrukce [NORMAL](#)).

HELICABS**Syntaxe**

HELICABS [**navesti**], **jednotkapohybuose1**, **poloha1**, **jednotkapohybuose2**, **poloha2**, **jednotkapohybuose3**, **poloha3**, **smer**, **±polomer** [, **uhel** [, **pocetotacek** [, **jednotkapohybuose4**, **poloha4** [, ..., **jednotkapohybuose6**, **poloha6**]]]]

Argumenty

navesti konstanta nebo proměnná integer. Návěstí, které identifikuje blok přesunu
jednotkapohybuose1...jednotkapohybuose3[...jednotkapohybuose6] názvy zařízení druhu jednotka pohybu v ose
poloha1...poloha3[...poloha6] konstanta nebo proměnná. Poloha absolutního přesunu
smer proměnná integer. Druh otáčení ve směru/proti směru hodinových ručiček (CW/CCW)
polomer konstanta nebo proměnná. Poloměr spirály
uhel konstanta nebo proměnná. Výchozí úhel
pocetotacek konstanta nebo proměnná. Počet otáček

Popis

Spirálovitá interpolace s absolutním přesunem rovnajícím se naprogramovaným polohám **poloha1**, **poloha2** a **poloha3**. Pohyb je tvořen kruhovou interpolací přiřazenou jednotkám pohybu v ose **jednotkapohybuose1** a **jednotkapohybuose2** (se stejnými pravidly syntaxe jako v případě [CIRCABS](#) /[CIRCINC](#), vztahující se na argumenty **smer**, **±polomer** a **uhel**) a lineární interpolací přiřazenou jednotkám pohybu v osách **jednotkapohybuose3** (případně **jednotkapohybuose4**, **jednotkapohybuose5** a **jednotkapohybuose6**). Spirálovitý pohyb může být rozvinutý na více otáčkách určených argumentem **pocetotacek**. Poloha týkající se jednotky pohybu v ose s lineárním pohybem (jako případné polohy **jednotkapohybuose4**, **jednotkapohybuose5**, **jednotkapohybuose6**) se vztahuje na celkový přesun (tedy ne na přesun/otáčku). Volitelný parametr **navesti** se používá ve spojení s instrukcí [SETLABELINTERP](#) pro jedinečnou identifikaci bloku přesunu.

Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX. V tomto případě je třeba brát v úvahu, že výrazem interpolace se má na mysli koordinovaný pohyb více jednotek pohybu v osách ovlivněný diskrétní chybou způsobenou metodou řízení jednotky pohybu v ose.

Poznámka

- Podmínka sledování obrysu je vyhodnocována pouze na prvních třech jednotkách pohybu v ose, které tvoří vztahný systém. Výsledkem přidání další osy a případné změny je chybná správa profilu rychlosti. Pro dosažení správného pohybu je třeba mezi instrukci HELICABS a další instrukci vložit instrukci [WAITSTILL](#).

2. Při nastavení místního vztažného systému s použitím instrukce [SETRIFLOC](#) musí být tři jednotky pohybu v osách, které definují nový systém, vždy uvedeny mezi parametry instrukce HELICABS, i když nevykonávají přesuny.

HELICINC

Syntaxe

HELICINC

[navesti], jednotkapohybuose1, poloha1, jednotkapohybuose2, poloha2, jednotkapohybuose3, poloha3, smer, ± polomer [,uhel [, pocetotacek [, jednotkapohybuose4, poloha4 [, ..., jednotkapohybuose6, poloha6]]]]

Argumenty

navesti	konstanta nebo proměnná integer. Návěští, které identifikuje blok přesunu
jednotkapohybuose1...jednotkapohybuose3[...jednotkapohybuose6]	názvy zařízení druhu jednotka pohybu v ose
poloha1...poloha3[...poloha6]	konstanta nebo proměnná. Poloha inkrementálního přesunu
smer	proměnná integer. Druh otáčení ve směru/proti směru hodinových ručiček (CW/CCW)
polomer	konstanta nebo proměnná. Poloměr spirály
uhel	konstanta nebo proměnná. Výchozí úhel
pocetotacek	konstanta nebo proměnná. Počet otáček

Popis

Spirálovitá interpolace s inkrementálním přesunem rovnajícím se naprogramovaným polohám **poloha1**, **poloha2** a **poloha3**. Pohyb je tvořen kruhovou interpolací přiřazenou jednotkám pohybu v osách **jednotkapohybuose1** a **jednotkapohybuose2** (se stejnými pravidly syntaxe jako v případě [CIRCABS](#) /[CIRCINC](#), vztahující se na argumenty **smer**, **±polomer** a **uhel**), a lineární interpolací přiřazenou **jednotcepohybuose3** (případně **jednotcepohybuose4**, **jednotcepohybuose5** a **jednotcepohybuose6**). Spirálovitý pohyb může být rozvinutý na více otáčkách určených argumentem **pocetotacek**. Poloha týkající se jednotky pohybu v ose s lineárním pohybem (jako případné polohy **jednotkypohybuose4**, **jednotkypohybuose5**, **jednotkypohybuose6**) se vztahuje na celkový přesun (a ne na přesun/otáčku). Volitelný parametr **navesti** se používá ve spojení s instrukcí [SETLABELINTERP](#) pro jedinečnou identifikaci bloku přesunu. Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX. V tomto případě je třeba brát v úvahu, že výrazem interpolace se má na mysli koordinovaný pohyb více jednotek pohybu v osách ovlivněný diskrétní chybou způsobenou metodou řízení jednotky pohybu v ose.

Poznámka

1. Podmínka sledování obrysu je vyhodnocována pouze na prvních třech jednotkách pohybu v ose, které tvoří vztažný systém. Výsledkem přidání další osy a případné změny je chybná správa profilu rychlosti. Pro dosažení správného pohybu je třeba mezi instrukci HELICINC a další vložit instrukci [WAITSTILL](#).
2. Při nastavení místního vztažného systému s použitím instrukce [SETRIFLOC](#) musí být tři jednotky pohybu v osách, které definují nový systém, vždy uvedeny mezi parametry instrukce HELICINC, i když nevykonávají přesuny.

JERKCONTROL

Syntaxe

JERKCONTROL

jednotkapohybuose, stav

Argumenty

jednotkapohybuose	názvy zařízení druhu jednotka pohybu v ose
stav	přednastavená konstanta. Přípustnými hodnotami jsou: ON stav zapnutého příznaku OFF stav vypnutého příznaku

Popis

Aktivuje (když je parametru **stav** přiřazena hodnota ON) nebo zakáže (když je parametru **stav** přiřazena hodnota OFF) kontrolu trhavých pohybů (jerk) na interpolovaných pohybech a na pohybech z bodu do bodu **jednotkypohybuose**. Kontrola trhavých pohybů (jerk) se aktivuje pouze v případě jednotek pohybu v osách s konfigurovanou rampou zrychlení a zpomalení ve tvaru 'S'. V případě jednotky pohybu v ose s lineární rampou kontrola trhavých pohybů (jerk) nebude prováděna.

JERKSMOOTH

Syntaxe

JERKSMOOTH **jednotkapohybuose, hodnota**

Argumenty

jednotkapohybuose názvy zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná druhu float

Popis

Během klasických interpolovaných pohybů se jednotky pohybu v osách mohou pohybovat v podmínkách sledování obrysu, tj. bez zastavení mezi dvěma po sobě následujícími bloky přesunu, když je úhel mezi tangenty a dráhou pohybu menší než hodnota parametru „Maximální úhel sledování obrysu“ (přednastavená hodnota je 15 stupňů, a lze ji měnit instrukcí [SETCONTORNATURE](#)).

V opačném případě budou jednotky pohybu v ose zastaveny v bodě hrany dvou bloků řízeného zpomalení a znovu uvedeny do chodu podél nového bloku s řízenými zrychleními. Zastavení a rozjezd však snižují výkonnost pohybu stroje. V případech, kdy úhel obrysu nabude výrazných hodnot, jako například hodnota přetržitosti tangenty vyšší než 5 stupňů, vznikají rychlostní skoky nezanedbatelné pro jednotky pohybu v osách, které jsou zahrnuty do pohybu, s následnými nekonečnými hodnotami zrychlení, trhavými pohyby (jerk) a výrazným mechanickým namáháním, které se může projevit také na kvalitě obrábění.

Instrukce JERKSMOOTH umožňuje na základě hodnoty určené uživatelem spojovat profily rychlostí sledované jednotkami pohybu v osách během pohybů sledování obrysu, a to hladkým způsobem, tj. s plynulostí zrychlení a rychlosti. Je třeba si všimnout, že tento hladký spoj zavádí malé změny v prováděné dráze pohybu vzhledem k její teoretické podobě, protože v okolí bodu sledování obrysu jednotky pohybu v ose vykazují profil rychlosti odlišný od teoretického profilu.

Proměnná **hodnota**, která je vyjádřena prostřednictvím procentuální hodnoty v rozmezí od 0 do 100, definuje, do jaké míry mají být pospojovány profily rychlosti hladkým způsobem. Hodnota rovnající se 0 udržuje teoretický profil, protože vytváří přetržitosti ve zrychleních a v profilech rychlosti. Hodnota rovná 100 zajistí profily pospojované jemně, nejlepší výkonnost, ale také maximální odchylku od teoretické dráhy pohybu, proporcionální rychlosti udržované podél dráhy pohybu.

Poznámka

Uvedená instrukce se aplikuje pouze na pohyby s klasickou interpolací (instrukce [LINEARABS](#), [LINEARINC](#), [CIRCABS](#), [CIRCINC](#), [HELICABS](#), [HELICINC](#)). Neaplikuje se v pohybech multisosové interpolace (instrukce [MULTIABS](#) a [MULTIINC](#)).

LINEARABS

Syntaxe

LINEARABS **[navesti],jednotkapohybuose1, poloha1, [jednotkapohybuose2, poloha2 [, jednotkapohybuose3, poloha3 [, ..., jednotkapohybuose6, poloha6]]]**

Argumenty

navesti konstanta nebo proměnná integer. Návěstí, které identifikuje blok přesunu
jednotkapohybuose1[...jednotkapohybuose6] názvy zařízení druhu jednotka pohybu v ose
poloha1[...poloha2[...poloha6]] konstanta nebo proměnná. Poloha absolutního přesunu

Popis

Lineární interpolace s *absolutním přesunem* do poloh určených prostřednictvím **poloha1**, **poloha2** apod. Volitelný parametr **navesti** se používá ve spojení s instrukcí [SETLABELINTERP](#) pro jedinečnou identifikaci bloku přesunu.

Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX. V tomto případě je třeba brát v úvahu, že výrazem interpolace se má na mysli koordinovaný pohyb více jednotek pohybu v osách ovlivněný diskrétní chybou způsobenou metodou řízení jednotky pohybu v ose.

Poznámka

1. Podmínka sledování obrysu je vyhodnocována pouze na prvních třech jednotkách pohybu v ose, které tvoří vztahný systém. Výsledkem přidání další osy a případné změny je chybná správa profilu rychlosti. Pro dosažení správného pohybu je třeba mezi instrukci LINEARABS a další vložit instrukci [WAITSTILL](#).
2. Při nastavení místního vztahného systému s použitím instrukce [SETRIFLOC](#) musí být tři jednotky pohybu v osách, které definují nový systém, vždy uvedeny mezi parametry instrukce LINEARABS, i když nevykonávají přesuny.

LINEARINC

Syntaxe

LINEARINC

[navesti], jednotkapohybuose1, poloha1, [jednotkapohybuose2, poloha2 [, ednotkapohybuose3, poloha3 [, ..., ednotkapohybuose6, poloha6]]]

Argumenty

navesti konstanta nebo proměnná integer. Návěstí, které identifikuje blok přesunu
jednotkapohybuose1[...jednotkapohybuose2[...jednotkapohybuose6]] názvy zařízení druhu jednotka pohybu v ose
poloha1[...poloha2[...poloha6]] konstanta nebo proměnná. Poloha inkrementálního přesunu

Popis

Lineární interpolace s *inkrementálním přesunem*, do poloh určených prostřednictvím **poloha1**, **poloha2** apod. Volitelný parametr **navesti** se používá ve spojení s instrukcí [SETLABELINTERP](#) pro jedinečnou identifikaci bloku přesunu.

Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX. V tomto případě je třeba brát v úvahu, že výrazem interpolace se má na mysli koordinovaný pohyb více jednotek pohybu v osách ovlivněný diskrétní chybou způsobenou metodou řízení jednotky pohybu v ose.

Poznámka

- Podmínka sledování obrysu je vyhodnocována pouze na prvních třech jednotkách pohybu v ose, které tvoří vztahný systém. Výsledkem přidání další osy a případné změny je chybná správa profilu rychlosti. Pro dosažení správného pohybu je třeba mezi instrukci LINEARINC a další vložit instrukci [WAITSTILL](#).
- Při nastavení místního vztahného systému s použitím instrukce [SETRIFLOC](#) musí být tři jednotky pohybu v osách, které definují nový systém, vždy uvedeny mezi parametry instrukce LINEARINC, i když nevykonávají přesuny.

MOVABS

Syntaxe

MOVABS

jednotkypohybuose1, hodnota1 [, jednotkypohybuose2, hodnota2 [, ..., jednotkypohybuose6, hodnota6]]

Argumenty

jednotkypohybuose1...[...jednotkypohybuose6] názvy zařízení druhu jednotka pohybu v ose
hodnota1...[...hodnota6] konstanta nebo proměnná. Hodnota absolutního přesunu

Popis

Slouží k provedení *absolutního pohybu* uvedených jednotek pohybu v osách do poloh určených parametry **hodnota1** [,...**hodnota6**].

Pro provedení pohybu jednotky pohybu v ose se nesmí používat interpolovaný pohyb a jednotka pohybu v ose se musí nacházet v cílové poloze nebo v okně. Pohyb jednotky pohybu v ose začíná bezprostředně po provedení instrukce. V případě provedení více instrukcí pohybu z bodu do bodu ve stejné úloze budou tyto instrukce vzájemně zřetězeny. Pokud se druhá úloha snaží provést instrukce pohybu z bodu do bodu na jednotce pohybu v ose, která je již obsazena pohybem, tato úloha zůstane čekat na dokončení pohybu ovládaného z první úlohy.

Dále je možné měnit rychlost mezi dvěma po sobě následujícími pohyby z bodu do bodu prostřednictvím instrukce [SETVEL](#). Oba pohyby budou spojeny prostřednictvím rampy rychlosti bez zastavení jednotek pohybu v osách.

Když nebude použita instrukce [SETVEL](#), maximální možná rychlost je dána hodnotou rychlosti manuálního pohybu uvedenou v konfiguraci.

Pohyb z bodu do bodu může být přerušen instrukcí [STOP](#) a následně obnoven instrukcí [START](#). Během přerušování pohybu zůstane jednotka pohybu v ose ve stavu „v režimu“, i když se fyzicky nepohybuje.

Pohyb může být přerušen instrukcí [ENDMOV](#). V tomto případě nemůže být obnoven.

Poznámka

- V minulosti pohyby z bodu do bodu:
 - Neumožňovaly změnu rychlosti, jestliže jednotka pohybu v ose nebyla zastavena. Aktuální chování je podobné jako chování při interpolovaných pohybech.
 - Při jejich přerušování příkazem STOP odpovídající jednotka pohybu v ose nabývala stavu „dosažená poloha“.
- Když je počet bloků přesunu vyšší než 32 a bloky jsou tvořeny mikroúseky, doporučuje se použít instrukce lineární interpolace namísto pohybu z bodu do bodu. Za účelem získání podrobnějších informací si vyžádejte od firmy TPA dokument „Limiti Firmware Movimento Punto Punto.doc“ (Limity firmwaru pro pohyb z bodu do bodu - pozn. překl.).

Příklad 1

[Rutina Vynulování při Přerušení \(Interrupt\)](#)

Příklad 2

```
; změna rychlosti
Function změnarychl
setvel X, 20
setvel X, 20
movabs X, 100, Y, 200
movabs X, 150, Y, 180
setvel X, 5
movabs X, 80, Y, 100
waitstill X, Y
fret
```

MOVINC**Syntaxe**

MOVINC

jednotkapohybuose1, hodnota1 [, jednotkapohybuose2, hodnota2 [, ..., jednotkapohybuose6, hodnota6]]

Argumenty

jednotkapohybuose1...
[...jednotkapohybuose6]
hodnota1...[...hodnota6]

názvy zařízení druhu jednotka pohybu v ose

konstanta nebo proměnná. Hodnota inkrementálního přesunu

Popis

Zadá každé jednotce pohybu v ose, aby provedla *inkrementální pohyb* odpovídající parametru **hodnota**. Pro provedení pohybu jednotky pohybu v ose se nesmí používat interpolovaný pohyb a jednotka pohybu v ose se musí nacházet v cílové poloze nebo v okně. Pohyb jednotky pohybu v ose začíná bezprostředně po provedení instrukce. V případě provedení více instrukcí pohybu z bodu do bodu ve stejné úloze budou tyto instrukce vzájemně zřetězeny. Pokud se druhá úloha snaží provést instrukce pohybu z bodu do bodu na jednotce pohybu v ose, která je již obsazena pohybem, tato úloha zůstane čekat na dokončení pohybu ovládaného z první úlohy.

Dále je možné měnit rychlost mezi dvěma po sobě následujícími pohyby z bodu do bodu prostřednictvím instrukce [SETVEL](#). Oba pohyby budou spojeny prostřednictvím rampy rychlosti bez zastavení jednotek pohybu v osách.

Když nebude použita instrukce [SETVEL](#), maximální možná rychlost je dána hodnotou rychlosti manuálního pohybu uvedenou v konfiguraci.

Pohyb z bodu do bodu může být přerušen instrukcí [STOP](#) a následně obnoven instrukcí [START](#). Během přerušení pohybu zůstane jednotka pohybu v ose ve stavu „v režimu“, i když se fyzicky nepohybuje.

Pohyb může být přerušen instrukcí [ENDMOV](#). V tomto případě nemůže být obnoven.

Poznámka

1) V minulosti pohyby z bodu do bodu:

- Neumožňovaly změnu rychlosti, jestliže jednotka pohybu v ose nebyla zastavena. Aktuální chování je podobné jako chování při interpolovaných pohybech.

- Při jejich přerušení příkazem STOP odpovídající jednotka pohybu v ose nabývala stavu „dosažená poloha“.

2) Když je počet bloků přesunu vyšší než 32 a bloky jsou tvořeny mikroúseky, doporučuje se použít instrukce lineární interpolace namísto pohybu z bodu do bodu. Za účelem získání podrobnějších informací si vyžádejte od firmy TPA dokument „Limity Firmware Movimento Punto Punto.doc“ (Limity firmwaru pro pohyb z bodu do bodu - pozn. překl.).

Příklad 1

[Rutina Vynulování jednotky pohybu v ose](#)

Příklad 2

```
; změna rychlosti
Function změnarychl
setvel X, 20
setvel Y, 20
movinc X, 100, Y, 200
movinc X, 150, Y, 180
setvel X, 5
movinc X, 80, Y, 100
waitstill X, Y
fret
```

MULTIABS

Syntaxe

MULTIABS [navesti],jednotkapohybuose1, hodnota1, [jednotkapohybuose2, hodnota2 [, jednotkapohybuose3, hodnota3 [, ..., jednotkapohybuose16, hodnota16]]]

Argumenty

navesti konstanta nebo proměnná integer. Návěstí, které identifikuje blok přesunu
jednotkapohybuose1...jednotk apohybuose16 názvy zařízení druhu jednotka pohybu v ose
hodnota1...[...hodnota16] konstanta nebo proměnná. Hodnota teoretické polohy konce bloku přesunu

Popis

Absolutní multilineární interpolace až o 16 jednotek pohybu v osách. Tento pohyb interpolace umožňuje předstih profilů rychlosti na základě nastavení příslušných tolerancí na jednotkách pohybu v osách, prostřednictvím příkazu [SETTOLERANCE](#) (pod pojmem tolerance jednotky pohybu v ose se má na mysli porce dráhy pohybu, na které nemusí existovat konstantní poměr interpolace). Pořadí zařazování jednotek pohybu v osách do instrukce MULTIABS **musí** vždy být stejné a **všechny** jednotky pohybu v ose zainteresované do pohybu musí být přítomné. Bloky přesunu jsou zařazovány do řady v běžném lookahead a pohyb je zahájen při provedení instrukce [WAITSTILL](#), [STARTINTERP](#) nebo při naplnění samotného lookahead. Z jednotek pohybu v osách zahrnutých do pohybu může být jedna použita jako collider prostřednictvím instrukce [WAITCOLL](#). Volitelný parametr **navesti** se používá ve spojení s instrukcí [SETLABELINTERP](#) pro jedinečnou identifikaci bloku přesunu. Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX. V tomto případě je třeba brát v úvahu, že výrazem interpolace se má na mysli koordinovaný pohyb více jednotek pohybu v osách ovlivněný diskrétní chybou způsobenou metodou řízení jednotky pohybu v ose.

Poznámka

S tímto druhem interpolace není možné používat virtuální vztažné systémy (instrukce [SETRIFLOC](#) a [RESRIFLOC](#)). Je možné provést pohyby s řetězenými jednotkami pohybu v osách (v [CHAIN](#)). Jednotky pohybu v osách zainteresované do interpolovaného pohybu více jednotek pohybu v osách musí být deklarovány master jiných jednotek pohybu v osách, které nejsou zainteresovány v pohybu. Dále je možné aplikovat FeedRate Override.

Příklad

```
setquote      x, 0
setquote      y, 0
setquote      z, 0
; první blok
setveli       x, velx1
setveli       y, vely1
setveli       z, velz1
multiabs      x, quotax1, y,quotay1, z,quotaz1
; druhý blok
settolerance  x,tollx2, y,tolly2, z,tollz2
setveli       x, velx2
setveli       y, vely2
setveli       z, velz2
multiabs      x,quotax2, y,quotay2, z,quotaz2
; třetí blok
settolerance  x,tollx3, y,tolly3, z,tollz3
setveli       x, velx3
setveli       y, vely3
setveli       z, velz3
multiabs      x,quotax3, y,quotay3, z,quotaz3
; čtvrtý blok
settolerance  x,tollx4, y,tolly4, z,tollz4
setveli       x, velx4
setveli       y, vely4
setveli       z, velz4
multiabs      x,quotax4, y,quotay4, z,quotaz4
waitstill     x, y, z
```

MULTIINC

Syntaxe

MULTINC **[navesti],jednotkapohybuvose1, hodnota1, [jednotkapohybuvose2, hodnota2 [, jednotkapohybuvose3, hodnota3 [, ..., jednotkapohybuvose16, hodnota16]]]**

Argumenty

navesti konstanta nebo proměnná integer. Návěstí, které identifikuje blok přesunu
jednotkapohybuvose1...jednotk názvy zařízení druhu jednotka pohybu v ose
apohybuvose16
hodnota1...[...hodnota16] konstanta nebo proměnná. Hodnota zvýšení teoretické polohy konce bloku přesunu

Popis

Inkrementální multilineární interpolace až o 16 jednotek pohybu v osách. Tento pohyb interpolace umožňuje předstih profilů rychlosti vhodným nastavením prostřednictvím instrukce [SETTOLERANCE](#) příslušných tolerancí na jednotkách pohybu v osách (pod pojmem tolerance jednotky pohybu v ose se má na mysli porce dráhy pohybu, na které nemusí existovat konstantní poměr interpolace). Pořadí zařazování jednotek pohybu v osách do instrukce MULTIINC **musí** být vždy stejné a **všechny** jednotky pohybu v ose zainteresované do pohybu musí být přítomné. Bloky přesunu jsou zařazovány do řady v běžném lookahead a pohyb je zahájen při provedení instrukce [WAITSTILL](#), [STARTINTERP](#) nebo při naplnění samotného lookahead. Z jednotek pohybu v osách zahrnutých do pohybu může být jedna použita jako collider prostřednictvím instrukce [WAITCOLL](#). Volitelný parametr **navesti** se používá ve spojení s instrukcí [SETLABELINTERP](#) pro jedinečnou identifikaci bloku přesunu. Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX. V tomto případě je třeba brát v úvahu, že výrazem interpolace se má na mysli koordinovaný pohyb více jednotek pohybu v osách ovlivněný diskrétní chybou způsobenou metodou řízení jednotky pohybu v ose.

Poznámka

S tímto druhem interpolace není možné používat virtuální vztažné systémy (instrukce [SETRIFLOC](#) a [RESRIFLOC](#)). Je možné provést pohyby s řetězenými jednotkami pohybu v osách (v [CHAIN](#)). Jednotky pohybu v osách zainteresované do interpolovaného pohybu více jednotek pohybu v osách musí být deklarovány master jiných jednotek pohybu v osách, které nejsou zainteresovány v pohybu. Dále je možné aplikovat FeedRate Override.

NORMAL

Syntaxe

NORMAL **jednotkapohybuvose**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose

Popis

Slouží k aktivaci kontroly polohy na **jednotce pohybuvose** a k zakázání řetězení jednotek pohybu v osách. Při zapnutí systému budou všechny konfigurované jednotky pohybu v ose uvedeny do stavu free a poté přejdou do stavu normal, když je provedena tato instrukce nebo když je proveden první pohyb. V každém případě se doporučuje provést tuto instrukci před provedením procedury vynulování jednotek pohybu v osách, aby se zrušil případný nouzový stav.

RESRIFLOC

Syntaxe

RESRIFLOC **jednotkapohybuvose1, jednotkapohybuvose2, jednotkapohybuvose3**

Argumenty

jednotkapohybuvose1...jednotk názvy zařízení druhu jednotka pohybu v ose
apohybuvose3

Popis

Slouží k obnovení absolutního vztažného systému pro jednotky pohybu v ose X Y Z (**jednotkapohybuvose1, jednotkapohybuvose2, jednotkapohybuvose3**). Tato instrukce se obvykle používá po nastavení vztažného systému jednotek rotačního-posuvného pohybu v osách prostřednictvím instrukce [SETRIFLOC](#).

SETINDEXINTERP

Syntaxe

SETINDEXINTERP

jednotkapohybuvose, **nazevpromenne**

Argumenty

jednotkapohybuvose
nazevpromenne

název zařízení druhu jednotka pohybu v ose
název globální proměnné druhu integer

Popis

Definuje index, který počítá počet bloků interpolace provedených jednotkou pohybu v ose. Během pohybu v interpolaci je proměnná **nazevpromenne** při každé změně loku zvýšena o 1.

Poznámka

Proměnná použitá jako index musí být globální proměnnou skupiny nebo globální proměnnou stroje.

SETLABELINTERP

Syntaxe

SETLABELINTERP

jednotkapohybuvose, **hodnota**

Argumenty

jednotkapohybuvose
hodnota

název zařízení druhu jednotka pohybu v ose
globální proměnná druhu integer

Popis

Proměnné **hodnota** je během pohybu v interpolaci při každém změně bloku přiřazena hodnota návěští nového bloku. Návěští je definováno v instrukcích interpolovaného pohybu.

Poznámka

Proměnná **hodnoty** musí být globální proměnná skupiny nebo globální proměnná modulu.

SETPFLY

Syntaxe

SETPFLY

jednotkapohybuvose, **stav**, **rychlost**, **poloha**, [**chyba**]

Argumenty

jednotkapohybuvose
stav

název zařízení druhu jednotka pohybu v ose
přednastavená konstanta. Označuje ověřovaný stav mikrospínačů.
Přípustnými hodnotami jsou:

rychlost
poloha
chyba

ON (zapnutý)
OFF (vypnutý)
konstanta nebo proměnná float
konstanta nebo proměnná
proměnná integer. Kód chyby

Popis

Umožňuje okamžitě vynulovat polohu jednotky pohybu v ose. Vynulování je řízeno jedním spínačem, který je připojen k rychlému vstupu konektoru jednotky pohybu v ose (na kartách se sběrnici MECHATROLINK-II se odkazuje na EXT11).

Během pohybu **jednotkypohybu**vose, který se provádí prostřednictvím instrukce MOVABS, instrukce vyčká na přepnutí mikrospínače vynulování do uvedeného **stavu**. Po zaznamenání této změny stavu dojde k vynulování reálné polohy jednotky pohybu v ose bez zastavení pohybu a bude automaticky dynamicky znovu definována cílová **poloha** a **rychlost**. Proto bude znovu obnoven správný pohyb pro dosažení cílové polohy a dle potřeby bude také provedena změna směru pohybu. Pokud bude dosaženo nastavené polohy, aniž by bylo zaznamenáno přepnutí stavu vstupu, a nebyl nastaven parametr **chyba**, bude vytvořena chyba systému. V případě nastavení parametru **chyba** bude tento parametr obsahovat číselný kód odpovídající chyby systému. V takovém případě vynulování nebylo provedeno, a proto je třeba provést instrukci [SETQUOTE](#) za účelem opětovného nastavení vyhledání mikrospínače.

Aby se přerušilo provádění okamžitého vynulování, stačí provést instrukci NORMAL na dané jednotce pohybu v ose nebo ukončit úlohu, která požádala o provedení vynulování.

Během provádění instrukce není možné současně provádět instrukce [SETPZERO](#) a [FASTREAD](#) na stejné jednotce pohybu v ose, je-li připojena ke kartám se sběrnici MECHATROLINK-II.

Příklad

[Rutina Vynulování při Přerušení \(Interrupt\)](#)

SETPFLYCHAINSTRAT

Syntaxe

SETPFLYCHAINSTRAT **jednotkapohybuose, [druh]**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
druh Konstanta druhu integer. Přípustnými hodnotami jsou:
0 = pouze jednotka pohybu v ose master vynuluje polohu, zatímco
jednotka pohybu v ose slave udržuje předchozí polohu
odlišná od 0 = jednotka pohybu v ose master a jednotka pohybu v ose
slave vynulují polohu synchronním způsobem

Popis

Tato instrukce umožňuje nastavit způsob, jakým se bude chovat uvedená jednotka pohybu v ose slave v případě instrukce [SETPFLY](#) na jednotce master.
Tato instrukce se provádí na jednotce pohybu v ose slave. Když je proměnná druh vynechána, bude použita přednastavená hodnota rovnající se 0.

SETPZERO

Syntaxe

SETPZERO **jednotkapohybuose, poloha [,chyba]**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
poloha konstanta nebo proměnná. Jedná se o inkrementální polohu
chyba proměnná integer. Kód chyby

Popis

Slouží k zahájení inkrementálního pohybu **jednotkypohybuose** do uvedené **polohy** a vyčká na zaznamenání značky nuly snímače impulzů (před dosažením uvedené polohy).
V okamžiku zaznamenání značky nuly je reálná poloha vynulována a jednotka pohybu v ose je zastavena. Pokud bude dosaženo nastavené polohy, aniž by byla zaznamenána značka Nuly, a nebyl nastaven parametr **chyba**, bude vytvořena chyba systému. V případě nastavení parametru **chyba** bude tento parametr obsahovat číselný kód odpovídající chyby systému. V takovém případě cílová hodnota nebyla nastavena, a proto je třeba provést instrukci [SETQUOTE](#) za účelem opětovného nastavení vyhledání značky.

Pohyb jednotek v osách, který je vytvářen touto instrukcí, může být přerušen instrukcí STOP a obnoven instrukcí START.

Když je instrukce provedena s jednotkami pohybu v osách S-CAN a s jednotkami pohybu v osách EtherCAT, je třeba provést nejdříve instrukci FREE.

Během provádění instrukce není možné současně provádět instrukce [SETPFLY](#) a [FASTREAD](#) na stejné jednotce pohybu v ose, je-li připojena ke kartám se sběrnici MECHATROLINK-II.

Příklad

```
FREE                      X
SETPZERO                X, 100
```

SETPZEROCHAINSTRAT

Syntaxe

SETPZEROCHAINSTRAT **jednotkapohybuose, [hodnota]**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
hodnota proměnná druhu integer. Přípustnými hodnotami jsou:
0 = pouze jednotka pohybu v ose master vynuluje polohu, zatímco
jednotka pohybu v ose slave udržuje předchozí polohu
odlišná od 0 = jednotka pohybu v ose master a jednotka pohybu v ose
slave vynulují polohu synchronním způsobem

Popis

Tato instrukce umožňuje nastavit způsob, jakým se bude chovat uvedená jednotka pohybu v ose slave v případě instrukce [SETPZERO](#) na jednotce master.
Tato instrukce se provádí na jednotce pohybu v ose slave.

Když je proměnná **hodnota** vynechána, bude použita přednastavená hodnota rovnající se 0.

SETQUOTE

Syntaxe

SETQUOTE **jednotkapohybuvose, poloha**

Argumenty

jednotkapohybuvose **poloha** název zařízení druhu jednotka pohybu v ose
konstanta nebo proměnná

Popis

Tato instrukce provádí nucené nastavení teoretické polohy a reálné polohy jednotky pohybu v ose uvedené v parametru **poloha**. Když je jednotka pohybu v ose v pohybu, způsobuje prudké zastavení, protože pro řízení se jednotka pohybu v ose náhle ocitá v cílové poloze (reálná poloha se rovná teoretické poloze). Proto se nedoporučuje používat tuto instrukci na jednotce pohybu v ose během jejího pohybu, není-li pohyb prováděn velmi nízkou rychlostí.

Příklad

[Rutina Vynulování jednotky pohybu v ose](#)

SETQUOTECHAINSTRAT

Syntaxe

SETQUOTECHAINSTRAT **jednotkapohybuvose, [hodnota]**

Argumenty

jednotkapohybuvose **hodnota** název zařízení druhu jednotka pohybu v ose
Proměnná druhu integer. Přípustnými hodnotami jsou:
0 = pouze jednotka pohybu v ose master inicializuje na novou polohu,
zatímco jednotka pohybu v ose slave udržuje předchozí polohu
odlišná od 0 = polohy jednotek pohybu v osách slave a master jsou
inicializovány synchronním způsobem

Popis

Tato instrukce umožňuje nastavit způsob, jakým se bude chovat uvedená jednotka pohybu v ose slave v případě instrukce [SETQUOTE](#) na jednotce master.

Tato instrukce se provádí na jednotce pohybu v ose slave.

Když je proměnná **hodnota** vynechána, bude použita přednastavená hodnota rovnající se 0.

SETRIFLOC

Syntaxe

SETRIFLOC **poloha1_ax1, poloha2_ax1, poloha3_ax1,
poloha1_ax2, poloha2_ax2, poloha3_ax2,
poloha1_ax3, poloha2_ax3, poloha3_ax3,
jednotkapohybuvose1, jednotkapohybuvose2,
jednotkapohybuvose3**

Argumenty

poloha1_ax1...poloha3_ax3 směrové kosiny tří jednotek pohybu v osách
jednotkapohybuvose1...jednotk názvy zařízení. Druh jednotky pohybu v ose
apohybuvose3

Popis

Umožňuje aktivovat karteziánský vztažný systém X' Y' Z' jednotek rotačního-posuvného pohybu v osách vzhledem k absolutnímu vztažnému systému stroje X Y Z, zastoupeného fyzickými jednotkami pohybu v osách **jednotkapohybuvose1, jednotkapohybuvose2 a jednotkapohybuvose3**.

Devět argumentů určuje Směrové Kosiny tří místních jednotek pohybu v osách vzhledem k absolutním,

$$\begin{array}{ccc} \cos\alpha_1 & \cos\beta_1 & \cos\gamma_1 \\ \cos\alpha_2 & \cos\beta_2 & \cos\gamma_2 \\ \cos\alpha_3 & \cos\beta_3 & \cos\gamma_3 \end{array}$$

které tvoří matici transformace koordinát.

Výchozí poloha nového vztažného systému se bere v aktuálním bodě.

Všechny instrukce interpolovaného pohybu, do kterých jsou zainteresovány jednotky pohybu v osách X, Y a Z, se vztahují k tomuto vztažnému systému až do provedení instrukce [RESRIFLOC](#).

SETTOLERANCE

Syntaxe

SETTOLERANCE

**jednotkapohybuose1, hodnota1, [jednotkapohybuose2, hodnota2
[, jednotkapohybuose3, hodnota3 [, ..., jednotkapohybuose16,
hodnota 16]]]**

Argumenty

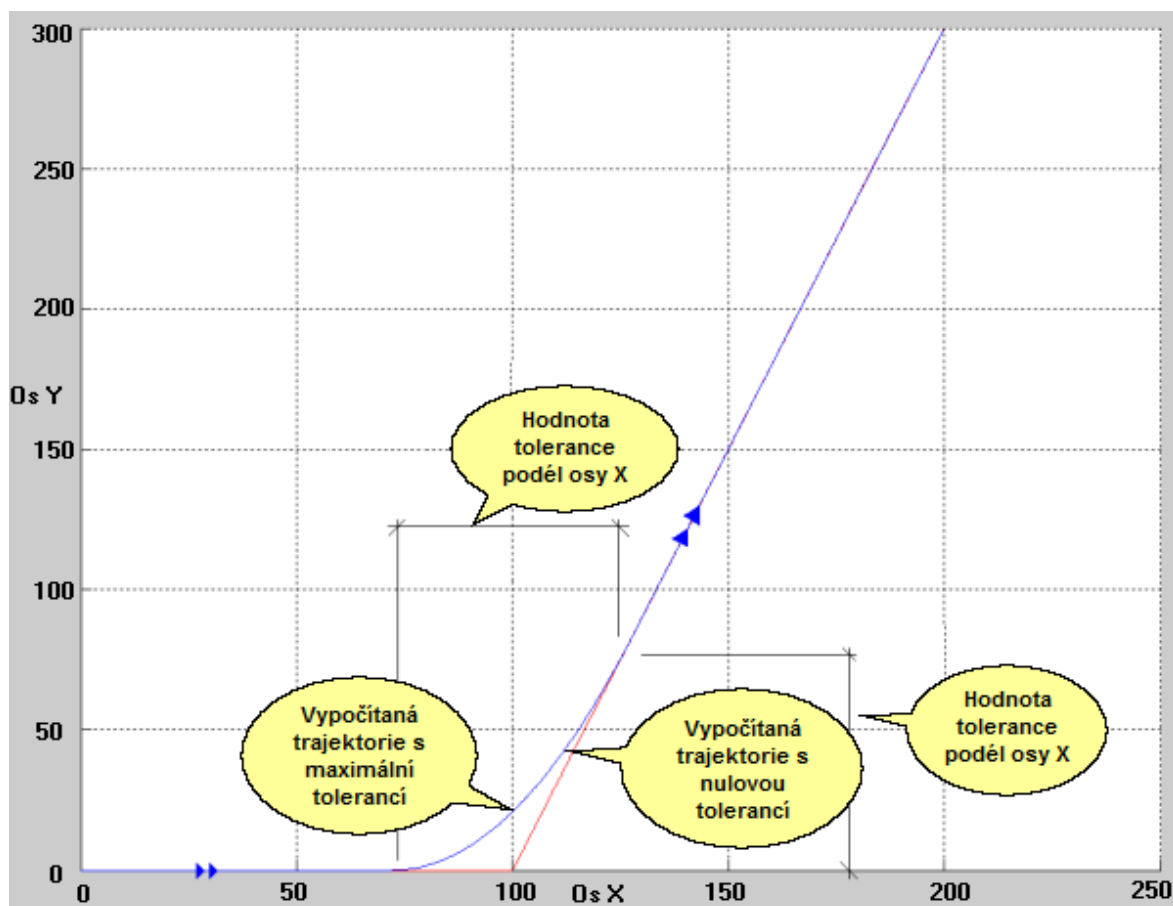
jednotkapohybuose1...jednotk názvy zařízení druhu jednotka pohybu v ose
apohybuose16

hodnota1...[...hodnota16]

konstanta nebo proměnná. Hodnota maximální tolerance aplikovatelná na jednotku pohybu v ose

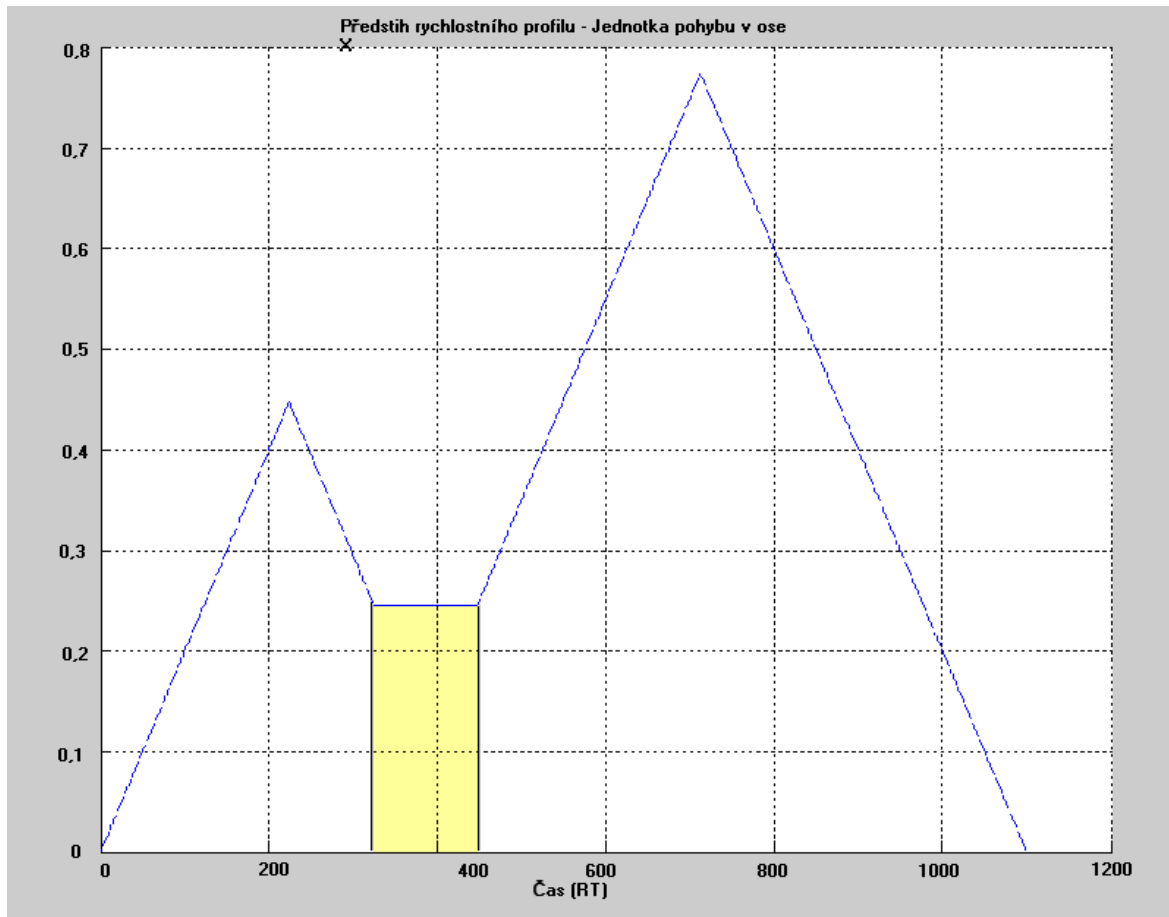
Popis

Nastavuje pro každou definovanou **jednotkupohybuose** **hodnotu** tolerance, která má být aplikována na interpolované pohyby více jednotek pohybu v osách. Hodnota tolerance je hodnotou přesunu, pro kterou se jednotka pohybu v ose odchýlí od původní dráhy pohybu při interpolaci více jednotek pohybu v osách. Tolerance se nastavuje pro každou **jednotkupohybuose** zainteresovanou do interpolace a systém zajistí předstih profilů rychlosti vůči tolerancím na všech jednotkách pohybu v ose, aniž by byl překročen prostor rampy, která představuje maximální limit pro předstih profilů. Chybějící přiřazení tolerance před instrukcí více jednotek pohybu v osách způsobuje aplikaci poslední nastavené tolerance na samotnou jednotku pohybu v ose. V případě, že by nikdy nebyla přiřazena hodnota tolerance konkrétní jednotce pohybu v ose, tato by byla považována za jednotku pohybu v ose s nulovou tolerancí. V tomto případě každý pohyb více jednotek pohybu v osách, který zahrnuje tyto jednotky pohybu v ose, nebude používat žádný předstih rampy. Proto se doporučuje systematické použití této instrukce pro každou zainteresovanou jednotku pohybu v ose před instrukcí pohybu interpolace více jednotek pohybu v osách.



Na výše uvedeném obrázku je znázorněna klasická dráha pohybu více jednotek pohybu v osách, která je tvořena dvěma bloky pohybu, přičemž první z nich spočívá v přesunu jednotky pohybu v ose X o 100, zatímco druhý spočívá v pohybu jednotky pohybu v ose Y o 300 a jednotky pohybu v X znovu o 100. Červená čára označuje dráhu pohybu v případě nulové tolerance, zatímco modrá je dráhou pohybu v případě maximální tolerance.

Tolerance může být viděna jako oblast uzavřená profilem rychlosti během intervalu předstihu v souladu s níže uvedeným obrázkem.



START

Syntaxe

START **jednotkapohybuose**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose

Popis

Obnovení pohybu **jednotkypohybuose** po jejím zastavení.

STARTINTERP

Syntaxe

STARTINTERP **jednotkapohybuose**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose

Popis

Slouží k zahájení interpolace, jejíž kanál je identifikován parametrem **jednotkapohybuose**. Obvykle pohyb jednotek pohybu v osách přiřazený k jednomu kanálu interpolace začíná po úplném naplnění vyrovnávací paměti interpolátoru (512 instrukcí) nebo po dosažení instrukce WAITSTILL, která ukončí pohyb. To umožní algoritmu interpolátoru určit optimální profily na základě informací týkajících se vysokého počtu (nebo všech) úseků pohybu v interpolace, které má k dispozici. Instrukce STARTINTERP umožňuje nuceně spustit pohyb jednotek pohybu v osách, i když nejsou splněny výše uvedené požadavky.

STOP

Syntaxe

STOP **jednotkapoihybuose**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose

Popis

Zastavení pohybu **jednotkypohybuvo**se. Jednotka pohybu v ose provede rampu zrychlení, jejíž trvání závisí na aktuální rychlosti a na parametrech konfigurace.

Příklad

[Rutina Vynulování jednotky pohybu v ose](#)

SWITCHENC**Syntaxe**

SWITCHENC **jednotkapohybuvo**1, [**jednotkapohybuvo**2, [**smer**, **poloha**]]

Argumenty

jednotkapohybuvo1 název zařízení druhu jednotka pohybu v ose
jednotkapohybuvo2 název zařízení druhu jednotka pohybu v ose. Označuje jednotku pohybu v ose počítání
smer přednastavená konstanta.
poloha **UP** = výměna snímačů impulzů po překročení polohy v kladném směru
DOWN = výměna snímačů impulzů po překročení polohy v záporném směru
konstanta double nebo proměnná double

Popis

Umožňuje vzájemně vyměnit snímač impulzů **jednotkypohybuvo**1 za snímač impulzů **jednotkypohybuvo**2. Vzájemná výměna snímačů impulzů se provádí při překročení uvedené **polohy** v kladném (UP) nebo záporném (DOWN) **smeru**.
Když jsou parametry **smer** a **poloha** vynechány, vzájemná výměna snímačů impulzů je provedena ihned, nezávisle na poloze jednotek pohybu v osách.

V případě deklarování samotné **jednotkypohybuvo**1 bude obnovena činnost samotného snímače impulzů.

Jednotkapohybuvo1 nemůže být druhu krokového pohybu, počítací a virtuální, zatímco **Jednotkapohybuvo**2 může být pouze jednotka pohybu v ose počítání. Kromě toho **jednotkapohybuvo**1 ani **jednotkapohybuvo**2 nemůže být zainteresována do pohybů v řetězení v úloze jednotky pohybu v ose slave.

Tento příkaz vytváří chybu systému 4101 - Nevyhovující správa jednotky pohybu v ose
NázevJednotkyPohybuOse, když je **jednotkapohybuvo**1 nebo **jednotkapohybuvo**2 prohlášena za slave v pohybu v řetězci nebo když **jednotkapohybuvo**1 provádí příkaz FASTREAD nebo příkaz SETPFLY.
Dále může být vytvořena chyba systému 4105 - Příkaz nelze provést na jednotce pohybu v ose
NázevJednotkyPohybuOse, když typ jednotky pohybu v ose není mezi těmi možnými.

WAITACC**Syntaxe**

WAITACC **jednotkapohybuvo**1 [, ..., **jednotkapohybuvo**6]

Argumenty

jednotkapohybuvo1 názvy zařízení druhu jednotka pohybu v ose
[...]**jednotkapohybuvo**6]

Popis

Čekání stavu zrychlení nebo jednoho z následujících určených stavů **jednotek pohybu v osách** (1÷6).
Úloha, ve které je provedena instrukce, bude uvedena do stavu čekání, dokud se jednotka pohybu v ose nebude nacházet ve stavu zrychlení nebo v jednom z následujících stavů.

Stavy jednotky pohybu v ose jsou stejné jako v případě celého čísla:

- zrychlení = 1
- režim = 2
- zpomalení = 3
- dosažená poloha = 4
- čekání na velké okno = 5
- čekání na malé okno = 6
- čekání na zastavení jednotky pohybu v ose = 7

WAITCOLL

Syntaxe

WAITCOLL **jednotkapohybuose, hodnota, casovylimit, delta**

Argumenty

jednotkapohybuose	název zařízení druhu jednotka pohybu v ose
hodnota	konstanta nebo proměnná. Hodnota absolutní polohy
casovylimit	konstanta nebo proměnná. Jedná se o dobu čekání na zastavenou jednotku pohybu v ose
delta	konstanta nebo proměnná. Jedná se o hodnotu okna pro dosažení zastavené jednotky pohybu v ose

Popis

Během pohybu jednotky pohybu v ose může být dosažení naprogramované polohy zabráněno překážkou mechanického charakteru, jakou může představovat například obráběný díl. Systém v tomto případě vytvoří chybu systému „chyba servořízení“ nebo „nedokončený pohyb“. Touto instrukcí se definuje **hodnota** polohy, na jejímž místě systém začne ověřovat přítomnost kolize; doba čekání (**casovylimit**) předtím, než bude **jednotkapohybuose** po vzniku kolize uvedena do stavu „dosažená poloha“, a **delta**, která definuje toleranci polohování jednotky pohybu v ose. Když jednotka pohybu v ose překročí polohu definovanou parametrem **hodnota**, dojde k ověření toho, zda je jednotka pohybu v ose ještě v pohybu. Po zachycení překážky bude identifikována kritická situace a i když bude zajištěna tažná síla motoru, nebude již kontrolováno překročení limitu chyba smyčky. Směr pohybu, ve kterém bude ověřena vzniklá kolize, je stejný jako směr posledního pohybu zařazeného do řady. Parametr **casovylimit** je vyjádřen v sekundách, hodnota **delta** musí být větší o 0,001 mm a menší než rozdíl mezi naprogramovanou cílovou polohou a polohou **hodnota**. Instrukce může být použita s interpolátorem více jednotek pohybu v osách vzhledem k tomu, že u interpolátoru tohoto druhu se akceptuje dočasná ztráta vazby interpolace.

Instrukce je aplikovatelná také na jednotky pohybu v osách Master pohybu v Chain.

Ke vzniku chyby systému dojde, když:

- **jednotka pohybu v ose** provádí klasický interpolovaný pohyb (viz instrukce LINEARBS, LINEARINC, CIRCABS, CIRCINC, HELICABS, HELICINC), koordinovaný pohyb nebo interpolované pohyby ISO.
- **jednotkapohybuose** je jednotka pohybu v ose Slave
- **jednotkapohybuose** je jednotky pohybu v ose počítání nebo jednotka krokového pohybu v ose nebo virtuální jednotka pohybu v ose
- nastavená **hodnota** je vyšší než poloha ukončení pohybu

Příklad

```
; nastavení polohy jednotky pohybu v ose X
SETQUOTE X, 0.0
; pohyb jednotky pohybu v ose X do absolutní polohy 1000
MOVABS X, 1000.0
; čekání na dosažení kolizní polohy, čekání 2 sekundy
; před uvedením jednotky pohybu v ose do stavu „dosažená poloha“ po
; zaznamenání kolize s přesností 0,01 mm
WAITCOLL X, 980.0,2.0,0.01
```

WAITDEC

Syntaxe

WAITDEC **jednotkapohybuose1 [, ..., jednotkapohybuose6]**

Argumenty

jednotkapohybuose1 [...],jednotkapohybuose6]	názvy zařízení druhu jednotka pohybu v ose
--	--

Popis

Čekání stavu zpomalení nebo jednoho z následujících určených stavů **jednotekpohybuosách** (1÷6). Úloha, ve které je provedena instrukce, bude uvedena do stavu čekání, dokud se jednotka pohybu v ose nebude nacházet ve stavu zpomalení, dosažení polohy, čekání na velké okno, čekání na malé okno a zastavení jednotky pohybu v ose.

Stavy jednotky pohybu v ose jsou stejné jako v případě celého čísla:

- zrychlení = 1
- režim = 2
- zpomalení = 3
- dosažená poloha = 4
- čekání na velké okno = 5
- čekání na malé okno = 6
- čekání na zastavení jednotky pohybu v ose = 7

WAITREG

Syntaxe

WAITREG **jednotkapohybuvose1 [, ..., jednotkapohybuvose6]**

Argumenty

jednotkapohybuvose1 názvy zařízení druhu jednotka pohybu v ose
[...,jednotkapohybuvose6]

Popis

Čekání stavu režimu nebo jednoho z následujících určených stavů **jednotekpohybuvosách** (1÷6). Úloha, ve které je provedena instrukce, bude uvedena do stavu čekání, dokud se jednotka pohybu v ose nebude nacházet ve stavu režimu, zpomalení, dosažení polohy, čekání na velké okno, čekání na malé okno a zastavení jednotky pohybu v ose.

Stavy jednotky pohybu v ose jsou stejné jako v případě celého čísla:

- zrychlení = 1
- režim = 2
- zpomalení = 3
- dosažená poloha = 4
- čekání na velké okno = 5
- čekání na malé okno = 6
- čekání na zastavení jednotky pohybu v ose = 7

WAITSTILL

Syntaxe

WAITSTILL **jednotkapohybuvose1 [, ..., jednotkapohybuvose6]**

Argumenty

jednotkapohybuvose1 názvy zařízení druhu jednotka pohybu v ose
[...,jednotkapohybuvose6]

Popis

Čekání všech uvedených **jednotekpohybuvose** (1÷6) na ukončení pohybu (stav Dosažená poloha).

Příklad

[Rutina Vynulování jednotky pohybu v ose](#)

WAITTARGET

Syntaxe

WAITTARGET **jednotkapohybuvose1 [, ..., jednotkapohybuvose6]**

Argumenty

jednotkapohybuvose1 názvy zařízení druhu jednotka pohybu v ose
[...,jednotkapohybuvose6]

Popis

Čekání všech uvedených **jednotekpohybuvosach** (1÷6) na dosažení rovnosti mezi aktuální teoretickou polohou a cílovou polohou. Reálná poloha se bude lišit od teoretické, dokud nebude provedeno vynulování chyby smyčky.

WAITWIN

Syntaxe

WAITWIN **jednotkapohybuvose1 [, ..., jednotkapohybuvose6]**

Argumenty

jednotkapohybuvose1 názvy zařízení druhu jednotka pohybu v ose
[...,jednotkapohybuvose6]

Popis

Čekání stavu okna nebo jednoho z následujících určených stavů **jednotekpohybuvosách** (1÷6). Úloha, ve které je provedena instrukce, bude uvedena do stavu čekání, dokud se jednotka pohybu v ose nebude nacházet ve stavu čekání na velké okno, čekání na malé okno a zastavení jednotky pohybu v ose.

Stavy jednotky pohybu v ose jsou stejné jako v případě celého čísla:

- zrychlení = 1
- režim = 2

konstanta	popis	druh	zpětná hodnota
_CFGUM	měrné jednotka	i	0 = millimetry, 1 = palce, 2 = stupně, 3 = otáčky
_CFGGRIS	rozišení	d	impulzy pro MJ
_CFGVMAX	maximální rychlost	f	m/1' nebo inch/1" nebo stupně/1" o ot./1'
_CFGVMAXD	maximální rychlost v manuálním režimu	f	m/1' nebo inch/1" nebo stupně/1" o ot./1'
_CFGVMAXI	maximální rychlost interpolace	f	m/1' nebo inch/1" nebo stupně/1" o ot./1'
_CFGPHINV	inverze fází snímače impulzů	b	0=žádná inverze, 1=inverze
_CFGRFINV	inverze řídicího signálu	b	0=žádná inverze, 1=inverze
_CFGZIND	aktivace vynulování polohy při dosažení značky nuly	b	0=zakázáno, 1=aktivováno
_CFGSLOPE	druh rampy pro zrychlení/zpomalení při pohybu z bodu do bodu	b	0=lineární, 1=ve tvaru 'S', 2=ve tvaru dvojitého 'S'
_CFGKFFA	feed forward zrychlení	f	
_CFGKFFAI	feed forward interpolace zrychlení	f	
_CFGSRPP	rychlost zahájení rampy při krokovém pohybu	f	m/1' nebo inch/1" nebo stupně/1" o ot./1'
_CFGACC	doba zrychlení z 0 na _CFGVMAX	i	ms
_CFGDEC	doba zpomalení z _CFGVMAX na 0	i	ms
_CFGACCI	doba zrychlení z 0 na _CFGVMAXI	i	ms
_CFGDECI	doba zpomalení z _CFGVMAXI na 0	i	ms
_CFGQLP	limit jednotky pohybu v ose v kladném směru	d	poloha
_CFGQLN	limit jednotky pohybu v ose v záporném směru	d	poloha
_CFGKP	proporcionální koeficient	f	
_CFGKI	integrační koeficient	f	
_CFGKD	derivační koeficient	f	
_CFGKFF	feed forward	f	procentuální podíl
_CFGKPS	proporcionální koeficient jednotky pohybu v ose druhu slave	f	
_CFGKIS	integrační koeficient jednotky pohybu v ose druhu slave	f	
_CFGKDS	derivační koeficient jednotky pohybu v ose druhu slave	f	
_CFGQEAP	chyba smyčky v kladném směru	d	poloha
_CFGQEAN	chyba smyčky v záporném směru	d	poloha
_CFGKPI	proporcionální koeficient interpolace	f	
_CFGKII	integrační koeficient interpolace	f	
_CFGKDI	derivační koeficient interpolace	f	
_CFGTMINP	minimální kladné napětí	f	Volt
_CFGTMINN	minimální záporné napětí	f	Volt
_CFGSTMINP	napětí kladné mezní hodnoty	f	Volt
_CFGSTMINN	napětí záporné mezní hodnoty	f	Volt

konstanta	popis	druh	zpětná hodnota
_CFGESC	časový limit pohybu jednotky pohybu v ose	i	ms
_CFGDSE	aktivace dynamické chyby servořízení	b	0=zakázáno, 1=aktivováno
_CFGAN	aktivace automatického doladění	b	0=zakázáno, 1=aktivováno
_CFGOFFSET	napětí doladování - počáteční posun	f	Volt
_CFGCEE	poloha odpovídající chybnému připojení snímače impulzů	d	poloha
_CFGNOTCH	frekvence Notchova filtru	i	Hz
_CFGBUFI	velikost vyrovnávací paměti integračního výpočtu	i	[1, 200]
_CFGQAP	okno kladného dosažení polohy	d	
_CFGQAN	okno záporného dosažení polohy	d	
_CFGSLOPEI	druh rampy pro zrychlení/zpomalení při interpolaci	f	0=lineární, 1=ve tvaru 'S', 2=ve tvaru dvojitého 'S'
_CFGKFFI	feed forward interpolace	f	procentuální podíl
_CFGAAF	čekání na zastavení jednotky pohybu v ose	b	0=zakázáno, 1=aktivováno
_CFGENTYPE	druh snímače impulzů	i	0=simulovaný nebo chybějící, 1=reálný
_SRPP	rychlost zahájení rampy při krokovém pohybu	f	m/1' nebo inch/1" nebo stupně/1" o ot./1'
_ACC	doba zrychlení z 0 na _VMAX	i	ms
_DEC	doba zpomalení z _VMAX na 0	i	ms
_ACCI	doba zrychlení z 0 na _VMAXI při interpolaci	i	ms
_DECI	doba zpomalení z _VMAXI na 0 při interpolaci	i	ms
_QLP	limit jednotky pohybu v ose v kladném směru	d	poloha
_QLN	limit jednotky pohybu v ose v záporném směru	d	poloha
_KP	proporcionální koeficient	f	
_KI	integrační koeficient	f	
_KD	derivační koeficient	f	
_KFF	feed forward	f	procentuální podíl
_KPS	proporcionální koeficient jednotky pohybu v ose druhu slave	f	
_KIS	integrační koeficient jednotky pohybu v ose druhu slave	f	
_KDS	derivační koeficient jednotky pohybu v ose druhu slave	f	
_QEAP	chyba smyčky v kladném směru	d	poloha
_QEAN	chyba smyčky v záporném směru	d	poloha
_VEL	rychlost pohybu z bodu do bodu	f	m/1' nebo inch/1" nebo stupně/1" o ot./1'
_VELI	rychlost interpolace	f	m/1' nebo inch/1" nebo stupně/1" o ot./1'
_MODE	režim činnosti jednotky pohybu v ose	b	1=běžný, 2=volný, 8=interpol., 10=koord.

konstanta	popis	druh	zpětná hodnota
_PHINV	inverze fází snímače impulzů	b	0=žádná inverze, 1=inverze
_RFINV	inverze řídicího signálu	b	0=žádná inverze, 1=inverze
_ZIND	aktivace vynulování polohy při dosažení značky nuly	b	0=zakázáno, 1=aktivováno
_KPI	proporcionální koeficient interpolace	f	
_KII	integrační koeficient interpolace	f	
_KDI	derivační koeficient interpolace	f	
_KFFI	feed forward interpolace	f	procentuální podíl
_KFFA	feed forward zrychlení	f	procentuální podíl
_KFFAI	feed forward zrychlení interpolace	f	procentuální podíl
_ESC	časový limit pohybu jednotky pohybu v ose	i	ms
_CEE	poloha odpovídající chybnému připojení snímače impulzů	d	poloha
_NOTCH	frekvence Notchova filtru	i	Hz
_BUFI	velikost vyrovnávací paměti integračního výpočtu	i	[1,200]
_QAP	okno kladného dosažení polohy	d	
_QAN	okno záporného dosažení polohy	d	
_QEAPINV	limit chyby sledování v kladném směru při inverzi	d	
_QEANINV	limit chyby sledování v záporném směru při inverzi	d	
_OFSCoord	poloha posunu při koordinovaném pohybu	d	
_MS	druh jednotky pohybu v ose master nebo slave	b	0=žádné zřetězení, 4=master, 5=slave
_QENC	poloha ze snímače impulzů	d	poloha
_QR	reálná poloha	d	poloha
_RIS	použití rozlišení jednotky pohybu v ose	d	
_ST	stav jednotky pohybu v ose	b	1=zrychlení, 2=režim, 3=zpomalení, 4=poloha, 5=čekání na velké okno, 6=čekání na zastavení jednotky pohybu v ose, 7=čekání na malé okno, 8=rozjezd
_QT	teoretická poloha	d	poloha
_EA	chyba smyčky	d	poloha
_FF	feed forward	i	
_VC	aktuální rychlost	f	
_P	proporcionální korektor	i	
_I	integrační korektor	i	
_D	derivační korektor	i	
_FLGS	příznaky jednotek pohybu v osách	b	
_VCR	reálná aktuální rychlost	f	
_ADJUST	kompensace posunu jednotky pohybu v ose	i	hodnota v podobě celého čísla, která představuje napětí, které má být předáno pohonu, na straně karty jednotek pohybu v ose v DAC. Rozsah stupnice pro pohon je 10 voltů a pro DAC 32767
_DAC	DAC	i	hodnota v podobě celého čísla, která představuje napětí, které má být předáno pohonu, na straně karty jednotek pohybu

konstanta	popis	druh	zpětná hodnota
			v ose v DAC. Rozsah stupnice pro pohon je 10 voltů a pro DAC 32767.
_ACCINST	okamžitá hodnota zrychlení	f	
_FFA	feed forward zrychlení	i	
_GONETIME	doba uplynulá od začátku pohybu	f	s (0 pro jednotky pohybu v ose druhu slave a pro jednotky krokového pohybu v ose)
_RESTIME	zbývající doba do ukončení pohybu. Hodnoty se vztahují na pohyb přidělený ve vyrovnávací paměti v okamžiku požadavku	f	s (0 pro jednotky pohybu v ose druhu slave, pro jednotky koordinovaného pohybu v ose a pro jednotky krokového pohybu v osách a pohybu ISO)
_TARGETTIME	doba potřebná pro vytvoření cílové polohy	f	mikrosekundy
_GONESPACE	vzdálenost ujetá od začátku pohybu Hodnoty se vztahují na pohyb přidělený ve vyrovnávací paměti v okamžiku požadavku	f	procentuální podíl (100 pro jednotky pohybu v ose druhu slave, pro jednotky koordinovaného pohybu v ose a pro jednotky krokového pohybu v ose)
_RESSPACE	vzdálenost zbývající do ukončení pohybu. Hodnoty se vztahují na pohyb přidělený ve vyrovnávací paměti v okamžiku požadavku	f	procentuální podíl (0 pro jednotky pohybu v ose druhu slave, pro jednotky koordinovaného pohybu v ose a pro jednotky krokového pohybu v ose)
_AXESJERK	aktivace kontroly trhavých pohybů (jerk) na jednotce pohybu v ose	b	1=kontrola aktivována , 0=kontrola neaktivována
_MOVEJERK	aktivace kontroly trhavých pohybů (jerk) pohybu, do kterého je zainteresována jednotka pohybu v ose	b	1=kontrola aktivována , 0=kontrola neaktivována
_MOVETYPE	druh pohybu, do kterého je zainteresována jednotka pohybu v ose	b	1=klasický interpolovaný pohyb, 2=interpolovaný pohyb více jednotek pohybu v osách, 3=koordinovaný pohyb, 4=pohyb z bodu do bodu, 5=zřetězený pohyb (pouze jednotky pohybu slave)
_PARTYPESET	druh parametrů jednotky pohybu v ose, které se používají během pohybu	i	1=interpolace, 0=z bodu do bodu
_AXINRIFLOC	jednotka pohybu v režimu uchopení v systému s místním řízením	i	1=ano, 0=ne
_QTARGETTOOL	cílová poloha jednotky pohybu v ose. V případě interpolace ISO se jedná o cílovou polohu souřadnice hrotu obráběcího nástroje odpovídajícího jednotce pohybu v ose	d	
_QREALTOOL	reálná poloha jednotky pohybu v ose. V případě interpolace ISO se jedná o reálnou polohu souřadnice hrotu obráběcího nástroje odpovídajícího jednotce pohybu v ose	d	
_BACKLASH	hodnota mechanické vůle, definovaná pro jednotku pohybu v ose	d	
_DISABLED	zakázání jednotky pohybu v ose	b	1=jednotka pohybu v ose zakázána, 0=jednotka pohybu v ose aktivována
_DYNLIMIT	aktivace dynamické kontroly limitů jednotky	b	1=kontrola aktivována , 0=kontrola neaktivována

konstanta	popis	druh	zpětná hodnota
	pohybu v ose		
_AXESFEED	hodnota feedrate override aplikovaná aktuálně na jednotku pohybu v ose	f	
_CORRLIN	používaný druh korektoru lineárnosti	i	0=žádný použitý korektor, 1=samokorektor, 2=křížový korektor, 3=samokorektor spolu s křížovým korektorem
_VELISO	rychlost hrotu obráběcího nástroje během pohybu 5 jednotek pohybu v osách	f	
_ISOSTOPS	počet nucených zastavení interpolovaného pohybu, způsobených mezními situacemi ve správě lookahead	i	
_CURRATIO	hodnota aktuálně používaného poměru zřetězení	d	
_DYNRATIO	návrat v případě probíhající dynamické změny vztahu zřetězení	i	0=ne, 1=ano
_RESBLOCK	počet bloků přesunu, které ještě mají být provedeny	i	
_EXECBLOCK	počet již provedených bloků přesunu	i	
_TOTALBLOCK	počet bloků přesunu celkově zařazených do řady v rámci pohybu (aktuální hodnota)	i	
_SWITCHENC	slouží k monitorování probíhající výměny snímače impulzů	i	-1=jednotka pohybu v ose nepoužívá instrukci SWITCHENC, 0=byla provedena instrukce SWITCHENC, ale jednotka pohybu v ose používá svůj snímač impulzů, 1=byla provedena instrukce SWITCHENC a jednotka pohybu v ose používá snímač impulzů jednotky pohybu v ose počítání
_QOFSENC	hodnota posunu snímače impulzů	d	
_LENSETPZERO	vzdálenost ujetá pro dosažení značky nuly	d	poloha
_TORQUEINST	okamžitá hodnota krouticího momentu	i	
_CURRSLOPE	vrací druh rampy, který se právě používá při rychlých pohybech	i	0=lineární, 1=ve tvaru 'S' , 2=ve tvaru dvojitého 'S'
_CURRSLOPEI	vrací druh rampy, který se právě používá při interpolovaných pohybech	i	0=lineární, 1=ve tvaru 'S' , 2=ve tvaru dvojitého 'S'
_REALSLOPEI	vrací druh rampy, který se používá při interpolovaných pohybech	i	0=lineární, 1=ve tvaru 'S' , 2=ve tvaru dvojitého 'S'
_AXISPAR1...AXISPAR8	přídavné volitelné parametry pro jednotku pohybu v ose EtherCAT	i	číslo
_ISOMOVETYPE	Prováděný typ pohyb ISO	i	0=rychlý pohyb ISO, 1 = interpolovaný pohyb ISO, -1 = jiné
_QMAINENC	Reálná poloha z hlavního snímače impulzů, když se používá vedlejší snímač impulzů	d	

Pohybu z bodu do bodu

SETACC

Syntaxe

SETACC **jednotkapohybu**vose, [hodnota]

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Doba zrychlení

Popis

Přiřadí **jednotcepohybu**vose dobu zrychlení identifikovanou parametrem **hodnota**. Doba zrychlení je vyjádřena v milisekundách. Když je parametr **hodnota** vynechán, použije se parametr konfigurace. Když je instrukce vložena mezi dvě instrukce MOVABS nebo MOVINC, bude provedena první instrukce pohybu (se zastavením pohybu) s použitím předtím nastavených parametrů zrychlení a zpomalení. Druhá instrukce je provedena aplikací nových parametrů zrychlení. Instrukce SETACC má vliv pouze na pohyby, které následují po jejím provedení. Když je uvedena **hodnota** menší než hodnota konfigurace, bude použita hodnota konfigurace.

Viz také [SETDEC](#), [SETACCI](#) a [SETDECI](#).

SETDEC

Syntaxe

SETDEC **jednotkapohybu**vose, [hodnota]

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Doba zpomalení

Popis

Přiřadí **jednotcepohybu**vose dobu zpomalení identifikovanou parametrem **hodnota**. Doba zpomalení je vyjádřena v milisekundách. Když je parametr **hodnota** vynechán, použije se parametr konfigurace. Když je instrukce vložena mezi dvě instrukce MOVABS nebo MOVINC, bude provedena první instrukce pohybu (se zastavením pohybu) s použitím předtím nastavených parametrů zrychlení a zpomalení. Druhá instrukce je provedena aplikací nových parametrů zpomalení. Instrukce SETACC má vliv pouze na pohyby, které následují po jejím provedení. Když je uvedena **hodnota** menší než hodnota konfigurace, bude použita hodnota konfigurace.

Viz také [SETACC](#), [SETACCI](#) a [SETDECI](#).

SETDERIV

Syntaxe

SETDERIV **jednotkapohybu**vose [, hodnota]

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Koeficient derivačního účinku. Nejsou přípustné druhy char a integer

Popis

Přiřazuje **jednotcepohybu**vose **hodnotu** *koeficientu derivačního účinku*. Když parametr **hodnota** není uveden, použije se koeficient derivačního účinku uvedený v konfiguraci. Tato instrukce není aplikovatelná na krokový motor. Viz také instrukci [SETDERIVI](#).

SETFEED

Syntaxe

SETFEED **jednotkapohybu**vose, hodnota

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Představuje procentuální hodnotu feed rate override

Popis

Slouží ke změně **hodnoty** procentuálního podílu feed rate override **jednotkypohybu**vose vztahující se na *pohyby z bodu do bodu*. Viz také [SETFEEDI](#).

SETFEEDF

Syntaxe

SETFEEDF **jednotkapohybu**vose, **hodnota**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Procentuální podíl feed forward

Popis

Přiřazuje **jednotcepohybu**vose procentuální **hodnotu** *feed forward*.
Když parametr **hodnota** není uveden, použije se koeficient feed forward uvedený v konfiguraci.
Při aplikaci této instrukce na krokový motor dojde k vytvoření chyby systému. Dojde k tomu i v případě, že je proměnné **hodnota** přiřazena hodnota mimo rozsah od 0 do 100.
Viz také instrukce [SETFEEDFI](#), [SETFEEDFA](#), [SETFEEDFAI](#).

SETFEEDFA

Syntaxe

SETFEEDFA **jednotkapohybu**vose [, **hodnota**]

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Procentuální podíl feed forward

Popis

Přiřazuje **jednotcepohybu**vose procentuální **hodnotu** *feed forward* zrychlení pro pohyby z bodu do bodu.
Když parametr **hodnota** není uveden, použije se koeficient feed forward uvedený v konfiguraci.
Při aplikaci této instrukce na krokový motor dojde k vytvoření chyby systému. Dojde k tomu i v případě, že je proměnné **hodnota** přiřazena hodnota mimo rozsah od 0 do 100.
Viz také instrukce [SETFEEDFI](#), [SETFEEDFI](#), [SETFEEDFAI](#).

SETINTEG

Syntaxe

SETINTEG **jednotkapohybu**vose [, **hodnota**]

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Koeficient integračního účinku. Nejsou přípustné druhy char a integer

Popis

Přiřazuje **jednotcepohybu**vose **hodnotu** *koeficientu integračního účinku*.
Když parametr **hodnota** není uveden, použije se koeficient integračního účinku uvedený v konfiguraci.
Tato instrukce není aplikovatelná na krokový motor.
Viz také instrukci [SETINTEGI](#).

SETMULTIFEED

Syntaxe

SETMULTIFEED **jednotkapohybu**vose1, **hodnota**1, [**jednotkapohybu**vose2, **hodnota**2 [, **jednotkapohybu**vose3, **hodnota**3 [, ..., **jednotkapohybu**vose16, **hodnota** 16]]]

Argumenty

jednotkapohybuvose1...**jednotk** názvy zařízení druhu jednotka pohybu v ose
apohybuvose16
hodnota1...[...**hodnota**16] konstanta nebo proměnná. Představuje procentuální hodnotu feed rate override

Popis

Slouží ke změně **hodnoty** procentuálního podílu feed rate override uvedených **jednotekpohybu**vose vztahující se na *pohyby z bodu do bodu*. Pro každou jednotku pohybu v ose může být nastavena odlišná hodnota.

SETPROP

Syntaxe

SETPROP **jednotkapohybuose [, hodnota]**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Koeficient proporcionálního účinku. Nejsou přípustné druhy char a integer

Popis

Přiřazuje **jednotcepohybuose hodnotu koeficientu proporcionálního účinku**. Když parametr **hodnota** není uveden, použije se koeficient proporcionálního účinku uvedený v konfiguraci. Tato instrukce není aplikovatelná na krokový motor. Viz také instrukci [SETPROPI](#).

SETSLOPE

Syntaxe

SETSLOPE **jednotka pohybu v ose [, hodnota]**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná integer. Druh rampy

Popis

Slouží k nastavení druhu rampy pro použití pro rychlý pohyb:

- 0 = lineární rampa
- 1 = rampa s es
- 2 = rampa s dvojitým es

Když je **hodnota** vynechána, bude obnovena rampa konfigurace. Výměna druhu rampy může být provedena pouze při zastavené jednotce pohybu v ose, ve stavu POLOHA. V opačném případě bude vytvořena chyba systému 4101 - Nevyhovující správa jednotky pohybu v ose nazevejednotkypohybuose.

Ve spojení s touto instrukcí lze ověřit druh rampy, které právě používá jednotka pohybu v ose, prostřednictvím instrukce [GETAXIS](#) s parametrem `_CURRSLOPE`.

Viz také instrukci [SETSLOPEI](#).

SETVEL

Syntaxe

SETVEL **jednotkapohybuose [, rychlost]**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
rychlost konstanta float nebo proměnná float

Popis

Souží k nastavení **maximální rychlosti jednotky pohybu v ose** pro pohyby z bodu do bodu.

Rychlost je vyjádřena v měrné jednotce jednotky pohybu v ose, která je uvedena v konfiguraci.

Když je naprogramovaná **rychlost** vyšší, než je maximální rychlost konfigurace, bude použita rychlost konfigurace.

Když je argument **rychlost** vynechán, bude použita rychlost konfigurace. Jsou přípustné pouze kladné hodnoty **rychlosti**.

Viz instrukce [SETVELI](#).

Příklad

[Rutina Vynulování jednotky pohybu v ose](#)

Interpolovaný pohyb

LOOKAHEAD

Syntaxe

LOOKAHEAD **[hodnota]**

Argumenty

hodnota konstanta nebo proměnná. Hodnota lookahead

Popis

Slouží k nastavení hodnoty lookahead interpolátoru. Lookahead je počet bloků interpolace, které jsou zpracovány před zahájením pohybu jednotek pohybu v osách. Lookahead umožňuje vytvářet optimální profily pohybu, zejména při použití ramp ve tvaru 'S'.

Když není uveden parametr **hodnota**, systém používá lookahead 512 bloků (přednastavená hodnota). Maximální přípustná hodnota se rovná **4096/pocetkanalu**, kde **pocetkanalu** odpovídá počtu kanálů interpolace, které jsou definovány v konfiguraci modulu. Minimální přípustná hodnota je 256.

Poznámka

Pod pojmem blok interpolace se má na mysli souhrn informací přiřazených jakékoli instrukci interpolovaného pohybu (např. LINEARABS).

Příklad

LOOKAHEAD 1024

SETACCI

Syntaxe

SETACCI **jednotkapohybuose1** [, ..., **jednotkapohybuose6**] [, **hodnota**]

Argumenty

jednotkapohybuose1,
[...**jednotkapohybuose6**]
hodnota názvy zařízení druhu jednotka pohybu v ose
konstanta nebo proměnná. Doba zrychlení

Popis

Přiřazuje jednotkám pohybu v osách **jednotkapohybuose1**, **jednotkapohybuose2** dobu zrychlení pro interpolované pohyby, která je určena parametrem **hodnota**. Tato doba se vyjadřuje v milisekundách. Když je parametr **hodnota** vynechán, použije se parametr konfigurace.

Viz také [SETACC](#), [SETDEC](#) a [SETDECI](#).

SETACCLIMIT

Syntaxe

SETACCLIMIT **jednotkapohybuose**, [**hodnota**]

Argumenty

jednotkapohybuose
hodnota název zařízení druhu jednotka pohybu v ose
konstanta času pohonu

Popis

Aktivuje a zakazuje automatický výpočet rychlostí režimu interpolace v závislosti na zrychleních tolerovaných jednotkami pohybu v osách. Parametr **hodnota** je konstantou času použitou pro určení schůdku rychlosti tolerovaného **jednotkoupohybuose**, vyjádřeného v milisekundách. Tento parametr je volitelný. Při jeho vynechání bude mít instrukce efekt zákazu automatického výpočtu. Standardní hodnotou pro tento parametr je 30 milisekund. Snížením hodnoty této doby dojde ke zpomalení profilu, a tedy k mírnějšímu pohybu. Proloužení této doby bude mít opačný účinek. Tato instrukce nemusí být aplikována na spirálové interpolace.

SETACCSTRATEGY

Syntaxe

SETACCSTRATEGY **jednotkapohybuose**, [**hodnota**]

Argumenty

jednotkapohybuose
[**hodnota**] název zařízení druhu jednotka pohybu v ose
proměnná nebo konstanta integer

Popis

Umožňuje zvolit druh zrychlení, které má být aplikováno na následující pohyb interpolace. Tuto instrukci je třeba provést pro všechny jednotky pohybu v osách zainteresované do interpolace.

K přípustným hodnotám parametru **hodnota** patří 0, 1 a 2. V případě předání hodnoty 0 se přizpůsobí obvyklé strategii zrychlení (bude zvoleno zrychlení profilu nejmenší ze všech jednotek pohybu v osách zainteresovaných do interpolace).

V případě hodnoty rovné 2 a lineární interpolace bude použito maximální zrychlení, které jednotlivé kroky mohou snést (se zohledněním jednotlivých komponentů všech jednotek pohybu v osách, lineárních a/nebo rotačních); v případě kruhové interpolace zůstane správa zrychlení nezměněna. V případě hodnoty 1 bude vyvolána zastaralá správa, která je zachovávána kvůli kompatibilitě.

SETDERIVI

Syntaxe

SETDERIVI **jednotkapohybuvose [, hodnota]**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Koeficient derivačního účinku. Nejsou přípustné druhy char a integer

Popis

Přiřazuje **jednotcepohybuvose hodnotu** *koeficientu derivačního účinku* použitou během pohybů interpolace jednotek pohybu v osách.

Když parametr **hodnota** není uveden, použije se koeficient derivačního účinku uvedený v konfiguraci.

Tato instrukce není aplikovatelná na krokový motor.

Viz také instrukci [SETDERIV](#).

SETFEEDFAI

Syntaxe

SETFEEDFAI **jednotkapohybuvose [, hodnota]**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Procentuální podíl feed forward

Popis

Přiřazuje **jednotcepohybuvose** procentuální **hodnotu** *feed forward* zrychlení pro interpolované pohyby.

Když parametr **hodnota** není uveden, použije se koeficient feed forward uvedený v konfiguraci.

Při aplikaci této instrukce na krokový motor dojde k vytvoření chyby systému. Dojde k tomu i v případě, že je proměnné **hodnota** přiřazena hodnota mimo rozsah od 0 do 100.

Viz také instrukce [SETFEEDF](#), [SETFEEDFI](#), [SETFEEDFA](#).

SETFEEDI

Syntaxe

SETFEEDI **jednotkapohybuvose, hodnota**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Představuje procentuální hodnotu feed rate override

Popis

Slouží ke změně **hodnoty** procentuálního podílu feed rate override dell **jednotkypohybuvose** vztahující se na *pohyby v interpolaci*. Viz také instrukci [SETFEED](#).

SETFEEDFI

Syntaxe

SETFEEDFI **jednotkapohybuvose [, hodnota]**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Procentuální podíl feed forward

Popis

Přiřazuje **jednotcepohybuvose** procentuální **hodnotu** *feed forward* pro interpolované pohyby.

Když je argument **hodnota** vynechán, systém použije procentuální hodnotu feed forward definovanou v parametrech konfigurace příslušného zařízení druhu jednotky pohybu v ose.

Tato instrukce není aplikovatelná na krokový motor.

Pro proměnnou **hodnota** jsou přípustné hodnoty v rozsahu od 0 do 100.

Viz také instrukce [SETFEEDF](#), [SETFEEDFA](#), [SETFEEDFAI](#).

SETINTEGI

Syntaxe

SETINTEGI **jednotkapohybuvose [, hodnota]**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Koeficient integračního účinku. Nejsou přípustné druhy char a integer

Popis

Přiřazuje **jednotcepohybuvose hodnotu koeficientu integračního účinku** použitou během pohybů interpolace jednotek pohybu v osách.

Když parametr **hodnota** není uveden, použije se koeficient integračního účinku uvedený v konfiguraci.

Tato instrukce není aplikovatelná na krokový motor.

Viz také instrukci [SETINTEG](#).

SETPROPI

Syntaxe

SETPROPI **jednotkapohybuvose [, hodnota]**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná. Koeficient proporcionálního účinku. Nejsou přípustné druhy char a integer

Popis

Přiřazuje **jednotcepohybuvose hodnotu koeficientu proporcionálního účinku** použitou během pohybů interpolace jednotek pohybu v osách.

Když parametr **hodnota** není uveden, použije se koeficient proporcionálního účinku uvedený v konfiguraci.

Tato instrukce není aplikovatelná na krokový motor.

Viz také instrukci [SETPROP](#).

SETSLOPEI

Syntaxe

SETSLOPEI **jednotka pohybu v ose [, hodnota]**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná *integer*. Druh rampy.

Popis

nastavení druhu rampy v interpolovaných pohybech (kde to je povoleno):

- 0 = lineární rampa
- 1 = rampa s es
- 2 = rampa s dvojitým es

Když je **hodnota** vynechána, bude obnovena rampa konfigurace.

Změna typu rampy může být provedena pouze s osou dosud nepoužívanou v interpolačním kanálu. V opačném případě bude vytvořena chyba systému 4101 - Nevyhovující správa jednotky pohybu v ose nazevejednotkypohybuvose.

Ve spojení s touto instrukcí lze ověřit druh rampy, prostřednictvím instrukce [GETAXIS](#), s parametrem `_CURRSLOPEI` můžete zkontrolovat typ rampy aktuálně nastavené pro osu a pomocí parametru `_REALSLOPEI` můžete vidět typ rampy (to je kanál, ve kterém je použita osa) použité osou.

Viz také instrukci [SETSLOPE](#).

SETSLOWPARAM

Syntaxe

SETSLOWPARAM **jednotkapohybuvose [,hodnota1,hodnota2]**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota1 konstanta nebo proměnná double. Faktor celkového omezení

hodnota2 konstanta nebo proměnná double. Faktor omezení inverze

Popis

Tato instrukce mění hodnotu parametrů potřebných k výpočtu rychlosti zpomalení v případě, že je aktivní funkce zpomalení při sledování obrysů (viz instrukce [SETCONTORNATURE](#)). Rychlost zpomalení je na začátku vypočtena teoretickým způsobem pro jednu jednotku pohybu v ose po druhé. V případě změny směru pohybu může být snížena použitím parametru **hodnota2 ve výpočtech**. Poté se bude ze všech vypočtených hodnot brát v úvahu minimální hodnota, aby se dodržela dynamika nejvíce omezující jednotky pohybu v ose. Dále je možné snížit rychlost zpomalení ještě o faktor nezávislý na parametru **hodnota1**.

Když je parametr **hodnota1** nebo **hodnota2** vynechán, budou použity přednastavené hodnoty, které odpovídají nulovému účinku. Parametr **hodnota1** představuje hodnotu procentuálního omezení teoretické rychlosti zpomalení. Aplikovaná rychlost zpomalení se rovná $(100 - \text{hodnota1}) / 100$ krát teoretická rychlost zpomalení. Maximální hodnota omezení je rovná 100. V takovém případě se výsledná rychlost rovná 1% teoretické rychlosti. Když je naopak nulová a vynechaná, bude se brát v úvahu přednastavená hodnota, tj. celá teoretická rychlost.

Parametr **hodnota2** představuje procentuální podíl omezení teoretické rychlosti zpomalení od 1 do 10 krát v případě, že dojde ke změně směru pohybu dané jednotky pohybu v ose. Konkrétně proto platí, že když je **hodnota2** rovná 100, rychlost je omezena 10 krát. Opačně platí, že když je nulová nebo vynechaná, rychlost nebude omezena.

Tato instrukce vytváří chybu systému 4399: „Parametr mimo rozsah“ při nastavení hodnoty menší než nula nebo větší než 100. Je důležité připomenout, že když je vynechán parametr **hodnota1**, musí být vynechán také parametr **hodnota2**.

Poznámka

Použití této instrukce je spojeno s použitím instrukcí [JERKSMOOTH](#) a [SETCONTORNATURE](#) a má vliv pouze na pohyby s klasickou interpolací (instrukce [LINEARABS](#), [LINEARINC](#), [CIRCABS](#), [CIRCINC](#), [HELICABS](#), [HELICINC](#)).

SETVELI

Syntaxe

SETVELI **jednotkapohybuose1 [, ..., jednotkapohybuose6] [, rychlost]**

Argumenty

jednotkapohybuose1, [...jednotkapohybuose6]	název zařízení druhu jednotka pohybu v ose určeného k interpolaci
rychlost	konstanta float nebo proměnná float

Popis

Slouží k nastavení **maximální** rychlosti jednotek pohybu v osách **jednotkapohybuose1,** **jednotkapohybuose2,** pro interpolované pohyby.

Rychlost je vyjádřena v měrné jednotce jednotek pohybu v osách, která je určena příslušným parametrem v konfiguraci. Když je argument **rychlost** vynechán, bude použita maximální rychlost konfigurace.

Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX.

Viz instrukce [SETVEL](#).

SETVELILIMIT

Syntaxe

SETVELILIMIT **jednotkapohybuose, rychlost**

Argumenty

jednotkapohybuose	název zařízení druhu jednotka pohybu v ose
rychlost	konstanta float nebo proměnná float

Popis

Nastavuje jednotlivé komponenty **rychlosti** uvedené **jednotkypohybuose** pro interpolované pohyby.

Rychlost je vyjádřena v měrných jednotkách jednotky pohybu v ose.

Koordinovaný pohyb

SETFEEDCOORD

Syntaxe

SETFEEDCOORD **jednotkypohybuose, hodnota1, hodnota2**

Argumenty

jednotkapohybuose	název zařízení druhu jednotka pohybu v ose
hodnota1	konstanta nebo proměnná double. Představuje maximální procentuální podíl okamžité změny feedrate
hodnota2	konstanta nebo proměnná integer. Představuje počet provádění v reálném čase (real-time), pro které se neaplikuje žádná změna feedrate

Popis

Slouží ke změně **hodnoty1** procentuálního podílu maximální okamžité změny feed rate **jednotkypohybuose**. Feed rate již nebude měněn v časovém intervalu vyjádřeném reálným časem, ale bude definován v proměnné **hodnota2**. Jinými slovy, poté, co bude aplikována změna feedrate override, rovnající se **hodnotě1**, v reálném čase (real-time) tolikrát, jaká je hodnota parametru **hodnota2**, již nebude aplikována žádná nová změna feed rate. Kombinace těchto dvou parametrů definuje určitý druh zrychlení/zpomalení, které může jednotka pohybu v ose snést. Modulací těchto dvou parametrů lze dosáhnout rampy se „schůdky“ s požadovaným sklonem.

Poznámka

U každé jednotky pohybu v ose se musí při koordinovaném pohybu nastavit hodnota feedrate a časový interval, protože v opačném případě budou použity přednastavené hodnoty, kterými jsou **hodnota1=100** a **hodnota2=1**. Během provádění koordinovaného pohybu (instrukce **COORDIN**) systém přepočítá parametry **hodnota1** a **hodnota2**, které mají být aplikovány na pohyb, na základě všech zainteresovaných jednotek pohybu v osách. Řízení vyloučí zastavené jednotky pohybu v osách. Uvedené dva parametry budou vypočteny níže uvedeným způsobem:

hodnota1: minimální nastavená hodnota na jednotkách pohybu v ose, které se pohybují

hodnota2: hodnota dosažená výpočtem, kdy se **hodnota1** vydělí minimálním poměrem

hodnota1/hodnota2.

Příklad

Function KoordinovanyPohyb

```

Setquote      X,0
Setquote      Y,0
Setquote      Z,0

setFeedCoord  X, 20, 80
setFeedCoord  X, 10, 1
setFeedCoord  X, 3, 3

coordin       matrix, delta_t, UP, rigaInit, rigaEnd,mask,_X,columnX,
              Y,columnY, Z,columnZ
waitstill     x,y,z

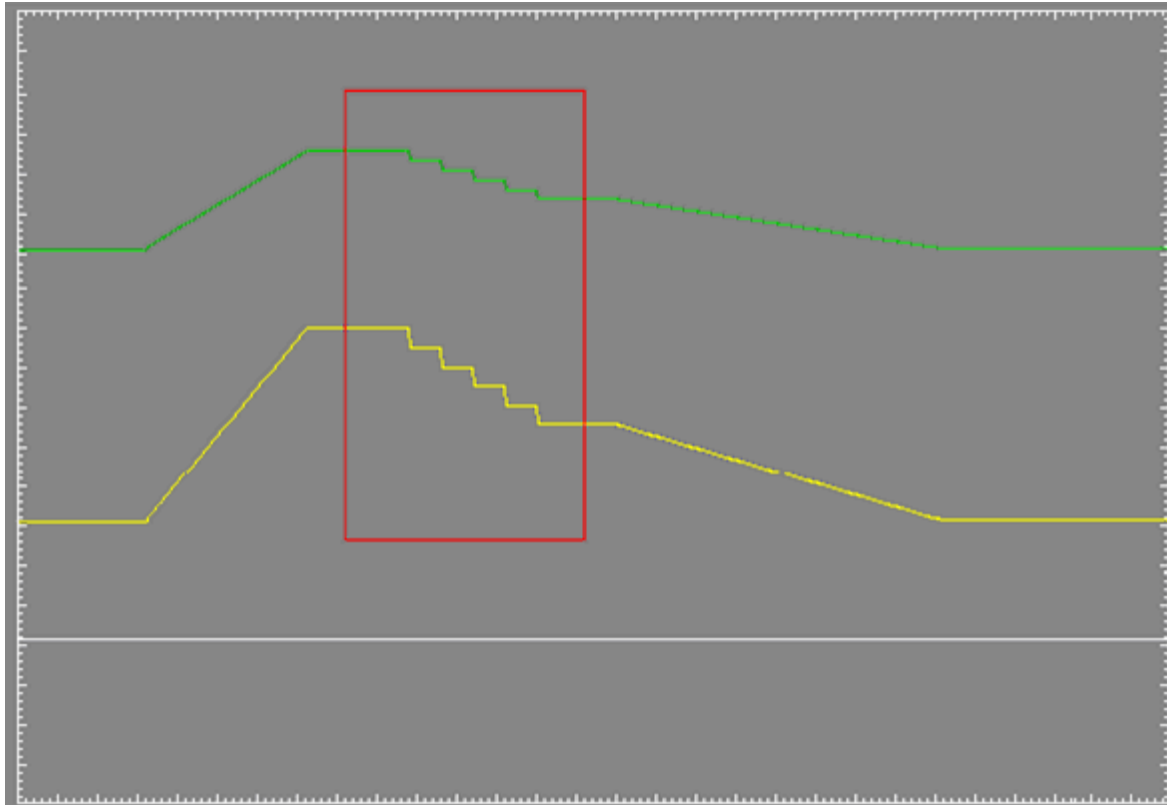
```

fret

Vycházejme z předpokladu, že v určitém úseku koordinovaného pohybu se jednotka pohybu v ose Z nepohybuje. Nastavené parametry jsou

Max_změna = 10
Delta_T = 10 / 0,25 = 40

Z toho vyplývá níže uvedená stopa osciloskopu, na které je zeleně zobrazen profil rychlosti jednotky pohybu v ose X a žlutě profil rychlosti jednotky pohybu v ose Y.



SETOFFSET

Syntaxe

SETOFFSET **jednotkapohybuose, poloha**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
poloha konstanta nebo proměnná. Posun pro koordinovaný pohyb

Popis

Umožňuje aplikovat posun na polohy koordinovaného pohybu. Posun uvedený parametrem **poloha** se použije v dalších koordinovaných pohybech, a to přičtením uvedené polohy ke všem polohám přítomným v tabulce. Viz také instrukci [COORDIN](#).

Zřetězený pohyb

RATIO

Syntaxe

RATIO **jednotkapohybuose, [hodnota]**

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose.
hodnota konstanta nebo proměnná. Redukční poměr.

Popis

Slouží k nastavení poměru zřetězení jednotky pohybu v ose druhu slave ke svému master. Pohyby jednotky pohybu v ose druhu slave budou odstupňovány vůči pohybům jednotky pohybu v ose druhu master nastaveným poměrem zřetězení. Při vynechání parametru **hodnota** bude obnoven poměr 1.0 (stejně pohyby). Tato instrukce vytváří chybu systému, když je provedena v době, kdy se jednotka pohybu v ose nebude nacházet ve stavu slave a odpovídající jednotka pohybu master se nenachází ve stavu dosažení polohy. Viz instrukce [CHAIN](#).

Příklad

CHAIN X, Y
RATIO Y, 0.5 ; redukční poměr 1/2

MOVABS X, 100 ; jednotka pohybu v ose Y se přesune do polohy 50
 WAITSTILL X

SETDYNRATIO

Syntaxe

SETDYNRATIO **jednotkapohybu**, **hodnota**

Argumenty

jednotkapohybu název zařízení druhu jednotka pohybu v ose
hodnota konstanta nebo proměnná double

Popis

Tato instrukce umožňuje dynamicky měnit poměr zřetězení během pohybu jednotky pohybu v ose master. Je možné aplikovat novou hodnotu poměru zřetězení, i když předchozí změna ještě nebyla ukončena.

Deklarovaná **jednotkapohybu** musí být jednotkou pohybu v ose slave.

V případě, že je instrukce provedena s jednotkou pohybu v ose master ve stavu DOSAŽENÁ POLOHA, nová hodnota poměru zřetězení **hodnota** bude aplikována okamžitě.

Změna poměru zřetězení proběhne prostřednictvím rampy lineárního zrychlení (nebo zpomalení). Použitá hodnota zrychlení je dána zrychlením jednotky pohybu v ose Master správně v záběru pro pohyb z bodu do bodu. To znamená, že je také možné měnit rampu prostřednictvím nastavení nové hodnoty zrychlení, a to použitím instrukce [SETACC](#).

Instrukce může vytvořit níže uvedenou chybu systému:

- „4101: Nevyhovující správa jednotky pohybu v ose“, v případě, že deklarovaná **jednotkapohybu** není jednotkou slave.

Všeobecné parametry

DYNLIMIT

Syntaxe

DYNLIMIT **jednotkapohybu**, **stav**

Argumenty

jednotkapohybu název zařízení druhu jednotka pohybu v ose
stav přednastavená konstanta. Přípustnými hodnotami jsou:
ON aktivace dynamické kontroly limitů jednotek pohybu v osách
OFF zakázání dynamické kontroly limitů jednotek pohybu v osách

Popis

Slouží k aktivaci nebo k zakázání dynamického ověření překročení limitu jednotky pohybu v ose. Dynamické ověření překročení limitů jednotky pohybu v ose se liší od statického ověření překročení limitů jednotky pohybu v ose, protože ověřuje při každém provádění v reálném čase (real-time), zda jednotka pohybu v ose na základě své aktuální rychlosti a svého maximálního zpomalení překročí své limity. Ověření statického druhu na rozdíl od ní ověřuje okamžik za okamžikem, zda se aktuální cílová hodnota každé jednotky pohybu v ose nachází uvnitř nastaveného kladného a záporného limitu jednotky pohybu v ose. Kromě toho před zahájením pohybu ověření statického druhu ověří, zda polohy předané prostřednictvím instrukcí pohybu překračují určené limity. Před instrukcí DYNLIMIT musí být nastavena instrukce [SETLIMPOS](#) a instrukce [SETLIMNEG](#) kvůli definování nových limitů jednotek pohybu v osách; v opačném případě se uvažované mezní hodnoty zmenšené o minimální prostor pro zastavení rovnají poloze 0.0 pro oba směry pohybu.

Příklad

Zkontrolujte limity jednotek pohybu v osách podle dvou druhů statického a dynamického ověření, s jednotkami pohybu v osách na stejné řídicí křivce pohybu.

Statické ověření

Ve všeobecném pohybu **Jednotka pohybu v ose X1** nemůže být vyšší než je počáteční kladný limit určený polohou **Jednotky pohybu v ose X2**. Ověření limitů jednotek pohybu v osách vytváří chybu systému č. 4108 „Jednotka pohybu v ose X1: Finální cílová poloha se nachází za softwarovým limitem“.

Dynamické ověření

Ve všeobecném pohybu nastane situace, že se okamžitá hodnota **poloha X1** se nachází v rozmezí určeném zmenšenými limity jednotky pohybu v ose s vhodným znaménkem v závislosti na směru pohybu jednotky pohybu v ose a na minimálním prostoru pro zastavení samotné jednotky pohybu v ose. Minimální prostor zastavení bude vypočítán jako funkce okamžité rychlosti a zpomalení nastaveného v konfiguraci pro pohyb z bodu do bodu. Kromě toho bude zakázána počáteční kontrola překročení limitů určených předanými polohami prostřednictvím instrukcí pohybu.



ENABLESTARTCONTROL

Syntaxe

ENABLESTARTCONTROL **jednotkapohybu**vose, [**casovylimit**]

Argumenty

jednotkapohybu vose	název zařízení druhu jednotka pohybu v ose
casovylimit	proměnná nebo konstanta integer. Jedná se o mezní hodnotu doby čekání, vyjádřenou v reálném čase.

Popis

Tato instrukce umožňuje aktivovat a nastavit **casovylimit** pro kontrolu chybějícího zahájení pohybu nebo náhlého zastavení jednotky pohybu v ose. Když se jednotka pohybu v ose nepohne nejméně o 2 kroky v průběhu 200 provedení v reálném čase (real-time), při požadavku na pohyb bude vytvořena chyba systému č. 3 „Chyba servořízení“.

Když je parametr **casovylimit** nastaven na nulu, kontrola bude zakázána. Instrukce nemá žádný vliv, když je teoretická hodnota menší než dva kroky v průběhu 200 provedení v reálném čase (real-time) nebo když je pohyb ukončen za méně než 200 provedení v reálném čase (real-time).

Příklad

```
ENABLESTARTCONTROL X, 10
```

; časový limit zahájení pohybu jednotek pohybu v osách se rovná 10 v reálném čase (real-time)

NOTCHFILTER

Syntaxe

NOTCHFILTER **jednotkapohybu**vose, [**hodnota**]

Argumenty

jednotkapohybu vose	název zařízení druhu jednotka pohybu v ose
hodnota	konstanta nebo proměnná. Hodnota frekvence [Hz]. Přípustné hodnoty jsou v rozmezí od 0 do 500 .

Popis

Slouží k nastavení frekvence řezání Notchova filtru pro uvedenou jednotku pohybu v ose. Když se **hodnota** rovná 0, filtr bude zrušen. V případě, že je parametr **hodnota** vynechán, bude použita hodnota nastavená v konfiguraci.

Příklad

```
NOTCHFILTER X, 97
```

; odřezání frekvence 97 Hz

RESLIMNEG

Syntaxe

RESLIMNEG **jednotkapohybu**vose

Argumenty

jednotkapohybu vose	název zařízení druhu jednotka pohybu v ose
----------------------------	--

Popis

Slouží ke zrušení ověření záporného limitu uvedené **jednotkypohybuvose**. Tyto instrukce se používají obvykle v rutinách vynulování k vyhledávání nulovacího spínače, čímž umožňují jednotkám pohybu v ose překročit nastavené hodnoty konfigurace. Viz také instrukce [SETLIMNEG](#), [SETLIMPOS](#), [RESLIMPOS](#).

Příklad

[Rutina Vynulování jednotky pohybu v ose.](#)

RESLIMPOS**Syntaxe**

RESLIMPOS **jednotkapohybuvose**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose

Popis

Slouží ke zrušení ověření kladného limitu uvedené **jednotkypohybuvose**. Tyto instrukce se používají obvykle v rutinách vynulování k vyhledávání nulovacího spínače, čímž umožňují jednotkám pohybu v ose překročit nastavené hodnoty konfigurace. Viz také instrukce [RESLIMNEG](#), [SETLIMPOS](#), [SETLIMNEG](#).

Příklad

[Rutina Vynulování jednotky pohybu v ose.](#)

SETADJUST**Syntaxe**

SETADJUST **jednotkapohybuvose, stav, [hodnota]**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
stav přednastavená konstanta. Může nabýt následujících hodnot:

- **ON** aktivovaná
- **OFF** zakázaná

[hodnota] proměnná nebo konstanta float. Napětí [Volt]

Popis

Aktivuje nebo zakazuje na **uvedené jednotce pohybu v ose** automatický výpočet rekuperace, tj. doladění. Adjust umožňuje kompenzovat mírné posuny polohy na konci pohybu jednotky pohyb v ose. Obvykle je adjust aktivován.

Může však být výhodné zakázat adjust u jednotek pohybu v osách poháněných motory, které se vyznačují vysokou hysterezí polohy a které by tedy nedosáhly žádné výhody použitím této funkce řízení.

Při opětovné aktivaci funkce adjust po jejím předešlém zakázání řízení nebude brát v úvahu předtím vypočtenou hodnotu. Tato instrukce proto může být použita pro vynulování hodnoty adjust, která byla akumulována jednotkou pohybu v ose, bez potřeby opětovné inicializace řízení.

Když je přítomen třetí parametr, posun bude nastaven na uvedenou **hodnotu** nezávisle na aktivaci nebo zakázání automatické doladění. Toto použití instrukce umožňuje kompenzovat softwarem případný posun řízení rychlosti namísto jeho kompenzace na pohonu, i když kompenzaci na pohonu je třeba upřednostnit před kompenzací prostřednictvím softwaru.

Tuto instrukci lze aplikovat pouze na jednotky pohybu v ose řízení analogicky.

SETBACKLASH**Syntaxe**

SETBACKLASH **jednotkapohybuvose, hodnota**

Argumenty

jednotkapohybuvose název zařízení druhu jednotka pohybu v ose
hodnota proměnná nebo konstanta float. Hodnota vůle

Popis

Tato instrukce umožňuje snížit nebo úplně odstranit účinky mechanické vůle na dráhu **jednotkypohybuvose**. **Hodnota** mechanické vůle může být nastavena v rozsahu od 0,0 do 3,0. Tato hodnota je nezávislá na zvolené měrné jednotce.

Ke speciálním situacím dochází v níže uvedených případech:

- Při zakázané jednotce pohybu v ose nebude funkce kompenzace vůle aplikována ani v případě, že bude vyžadována.
- Při jednotce pohybu v ose ve svislém směru se vzhledem její specifické konfiguraci neprojevuje žádná vůle.
- Při jednotce pohybu v ose se zátěží odpovídající střední setrvačnosti se může vyskytnout částečná nebo celková kompenzace zátěže. Může se opravdu stát, že vzhledem k hmotnosti zátěže dojde k zastavení jednotek pohybu v osách s opožděním vzhledem k zastavení motoru. Výsledné polohování zubů ozubeného převodu převodovky vzhledem k polohování zubů ozubeného převodu motoru může být proto takové, že bude snižovat nebo dokonce zcela nulovat vůli.
- Zobrazení reálné polohy a snímače impulzů jednotky **pohybu v ose**, vzorkované s použitím osciloskopu, vykazuje v místech aktivace kompenzace vůle (změna směru pohybu) vykazuje špičkovou hodnotu rovnající se hodnotě samotné vůle.

Tato instrukce způsobuje vytvoření chyby systému, je-li použita:

- Na jednotkách krokového pohybu v ose, které nejsou spravovány vzdálenými virtuálními moduly počítání TRS-AX.
- Na jednotkách krokového pohybu v ose, které jsou spravovány vzdálenými moduly TRS-AX se simulovaným snímačem impulzů.

Příklad

; Funkce s kompenzací vůle zakázána (červená stopa na výkresu)

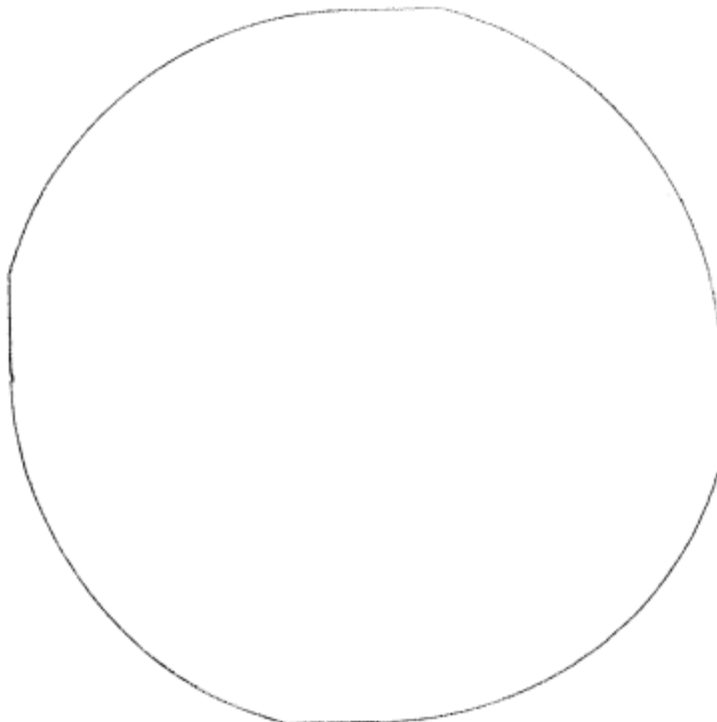
```
SETQUOTE      X, 0
SETQUOTE      Y, 0
SETVELI       X, 1.0
CIRCLE        X,Y,cw,100,90
WAITSTILL     X,Y
```

; Funkce s kompenzací vůle aktivována (černá stopa na výkresu)

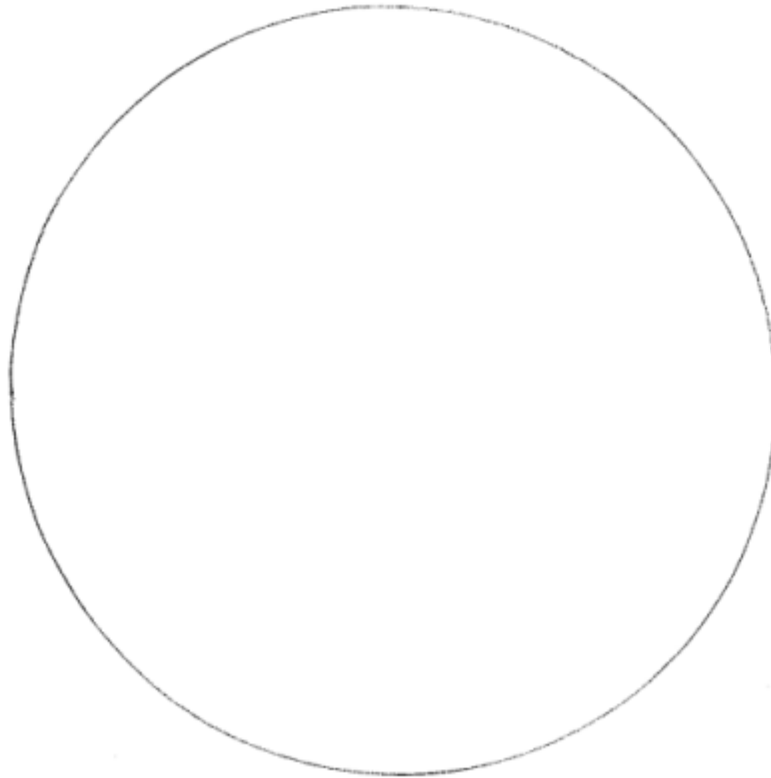
```
SETQUOTE      X, 0
SETQUOTE      Y, 0
SETVELI       X, 1.0
SETBACKLASH   X, 1.9
SETBACKLASH   Y, 1.8
CIRCLE        X,Y,cw,100,90
WAITSTILL     X,Y
```

Provedení obou funkcí vytváří dvě odlišné stopy.

První obrázek znázorňuje interpolaci dvou jednotek pohybu v osách, které vykazují mechanickou vůli ve spojení motor-převodovka.



Druhý obrázek znázorňuje stejnou interpolaci, ale s použitím instrukce pro kompenzaci vůle.



SETBIGWINFACTOR

Syntaxe

SETBIGWINFACTOR

jednotkapohybuvose, hodnota

Argumenty

jednotkapohybuvose
hodnota

název zařízení druhu jednotka pohybu v ose
proměnná nebo konstanta double. Faktor násobení pro výpočet velkého okna

Popis

Tato instrukce umožňuje změnit faktor násobení pro výpočet velkého okna na zvolené **jednotce pohybu**. Pro výpočet velkého okna se násobí proměnná **hodnota** parametrem Okna dosažení polohy, definovaným v konfiguraci jednotek pohybu v osách. **Hodnota**, která může být nastavena, se pohybuje v rozsahu od 1 do 257, včetně mezních hodnot. Přednastavená hodnota je 4,0.

SETDEADBAND

Syntaxe

SETDEADBAND

jednotkapohybuvose, VMinKlad, VMinZap, VMezKlad, VMezZap

Argumenty

jednotkapohybuvose
VMinKlad
VMinZap
VMezKlad
VMezZap

název zařízení druhu jednotka pohybu v ose
proměnná nebo konstanta float. Minimální kladné napětí [Volt]
proměnná nebo konstanta float. Minimální záporné napětí [Volt]
proměnná nebo konstanta float. Kladná mezní hodnota [Volt]
proměnná nebo konstanta float. Záporná mezní hodnota [Volt]

Popis

Slouží k nastavení parametrů minimálního napětí pro uvedenou jednotku pohybu v ose. Hodnoty minimálního napětí (kladného/záporného) se přičítají k teoretickému řídicímu napětí (kladnému/zápornému), když jeho hodnota překročí nastavenou mezní hodnotu (kladnou/zápornou). Když se teoretické řídicí napětí nachází uvnitř mezních hodnot, skutečné řídicí napětí je nuceně nastaveno na nulu. Je možné zakázat správu minimálního napětí nastavením všech hodnot na nulu. Mezní hodnoty musí být vždy menší než příslušné hodnoty minimálního napětí.

Při inicializaci systému je správa minimálního napětí zakázána.

SETENCLIMIT

Syntaxe

SETENCLIMIT **jednotkapohybu**vose, [hodnota]

Argumenty

jednotkapohybuvose **hodnota** název zařízení druhu jednotka pohybu v ose
konstanta nebo proměnná double

Popis

Slouží ke změně limitu chybného zapojení snímače impulzů. Tento parametr je vyjádřen v měrných jednotkách jednotky pohybu v ose. Přípustné hodnoty se musí nacházet v rozsahu intervalu od 128 po 16384 kroků snímače impulzů. Když je tento parametr vynechán, bude použita přednastavená hodnota rovnající se 1024 krokům.

Například pro jednotku pohybu v ose s rozlišením rovnajícím se 1000 impulzů/mm se budou přípustné hodnoty nacházet v rozsahu od 0,128 po 16,384 mm.

Když je parametr **hodnota** nastaven na nulu, bude zakázána kontrola limitu chybného zapojení snímače impulzů.

Příklad

; vnucený limit zapojení snímače impulzů rovnající se 3,5

```
SETENCLIMIT X, 3.5
```

SETINDEXEN

Syntaxe

SETINDEXEN **jednotkapohybu**vose, stav

Argumenty

jednotkapohybuvose **stav** název zařízení druhu jednotka pohybu v ose
přednastavená konstanta. Přípustnými hodnotami jsou:
ON stav značky nuly aktivován
OFF stav značky nuly je zrušen

Popis

Aktivuje nebo zakazuje na uvedené **jednotce**pohybu vynulování polohy v místě značky nuly. Za účelem provedené této instrukce musí být jednotky pohybu v ose typu počítací.

SETINTEGTIME

Syntaxe

SETINTEGTIME **jednotkapohybu**vose [, hodnota]

Argumenty

jednotkapohybuvose **hodnota** název zařízení druhu jednotka pohybu v ose
konstanta integer nebo proměnná

Popis

Slouží k nastavení počtu vzorků chyby sledování, použitých pro výpočet integrační složky. Platné hodnoty jsou v rozsahu od 1 do 200. Okamžitá změna tohoto parametru je možná, ale může způsobit schůdky řídicího signálu rychlosti jednotky pohybu v ose. Doporučuje se proto zasahovat do tohoto parametru při zastavených jednotkách pohybu v osách, které jsou zakázané a podle možnosti ve stavu free.

SETIRMPP

Syntaxe

SETIRMPP **jednotkapohybu**vose, rychlost

Argumenty

jednotkapohybuvose **rychlost** název zařízení druhu jednotka pohybu v ose
konstanta float nebo proměnná float. Počáteční rychlost rampy

Popis

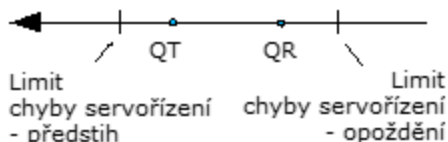
Přiřazuje **jednotce**pohybu hodnotu počáteční **rychlosti** rampy. Jedná se o minimální rychlost krokového motoru.

Tato instrukce se používá pro jednotky pohybu v ose, které se uvádějí do pohybu krokovými motory.

Přiřazuje **jednotcepohybuose** maximální hodnoty chyby sledování (**zpozdění**) a předstihu (**predstih**), které řízení připouští pouze pro záporný směr, před signalizací „chyby servořízení“.

- **zpozdění:** Jedná se o maximální hodnotu sledování tolerovanou v případě statické zkoušky chyby servořízení. Tato hodnota po přičtení k teoretické chybě proporcionální rychlosti určuje maximální hodnotu tolerovaného sledování.
- **predstih:** Jedná se o maximální hodnotu sledování tolerovanou během změny směru pohybu z pohybu v záporném směru na pohyb v kladném směru

Chyba sledování představuje rozdíl mezi teoretickou polohou (ve které by se jednotka pohybu v ose měla nacházet) a reálnou polohou. Když se jednotka pohybu v ose pohybuje v záporném směru, chyba sledování se záporným znaménkem poukazuje na stav zpoždění jednotky pohybu v ose, zatímco chyba sledování s kladným znaménkem poukazuje na stav předstihu. Když se nepoužívá tato instrukce, řízení používá maximální hodnoty chyby sledování uvedené v konfiguraci jednotky pohybu v ose a v takovém případě se limit předstihu rovná 1/4 limitu zpoždění.



Příklad

SETMAXERNEG Jednotkypohybuosach.X, 10, 5

; maximální zpoždění jednotky pohybu v ose je 10 mm , maximální předstih je 5 mm

SETMAXERPOS

Syntaxe

SETMAXERPOS jednotkapohybuose, zpozdění, predstih

Argumenty

jednotkapohybuose	název zařízení druhu jednotka pohybu v ose
zpozdění	konstanta nebo proměnná. Maximální chyba sledování
predstih	konstanta nebo proměnná. Maximální chyba předstihu

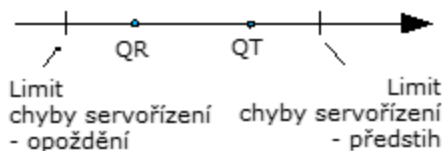
Popis

Přiřazuje **jednotcepohybuose** maximální hodnoty chyby sledování (**zpozdění**) a předstihu (**predstih**), které řízení připouští pouze pro kladný směr, před signalizací „chyby servořízení“.

- **zpozdění:** Jedná se o maximální hodnotu sledování tolerovanou v případě statické zkoušky chyby servořízení. Tato hodnota po přičtení k teoretické chybě proporcionální rychlosti určuje maximální hodnotu tolerovaného sledování.
- **predstih:** Jedná se o maximální hodnotu sledování tolerovanou během změny směru pohybu z pohybu v záporném směru na pohyb v kladném směru

Chyba sledování představuje rozdíl mezi teoretickou polohou (ve které by se jednotka pohybu v ose měla nacházet) a reálnou polohou. Když se jednotka pohybu v ose pohybuje v kladném směru, chyba sledování s kladným znaménkem poukazuje na stav zpoždění jednotky pohybu v ose, zatímco chyba sledování se záporným znaménkem poukazuje na stav předstihu.

Když se nepoužívá tato instrukce, řízení používá maximální hodnoty chyby sledování uvedené v konfiguraci jednotky pohybu v ose a v takovém případě se limit předstihu rovná 1/4 limitu zpoždění.



Příklad

SETMAXERPOS Jednotkypohybuosach.X, 10, 5

; maximální zpoždění jednotky pohybu v ose je 10 mm , maximální předstih je 5 mm

SETMAXERTYPE

Syntaxe

SETMAXERTYPE

jednotkapohybuvose, druh

Argumenty

jednotkapohybuvose
druh

název zařízení druhu jednotka pohybu v ose
Konstanta druhu integer. Přípustnými hodnotami jsou:
0 = nastavuje chybu servořízení podle mezní hodnoty (přednastavená hodnota)
1 = nastavuje dynamickou chybu servořízení

Popis

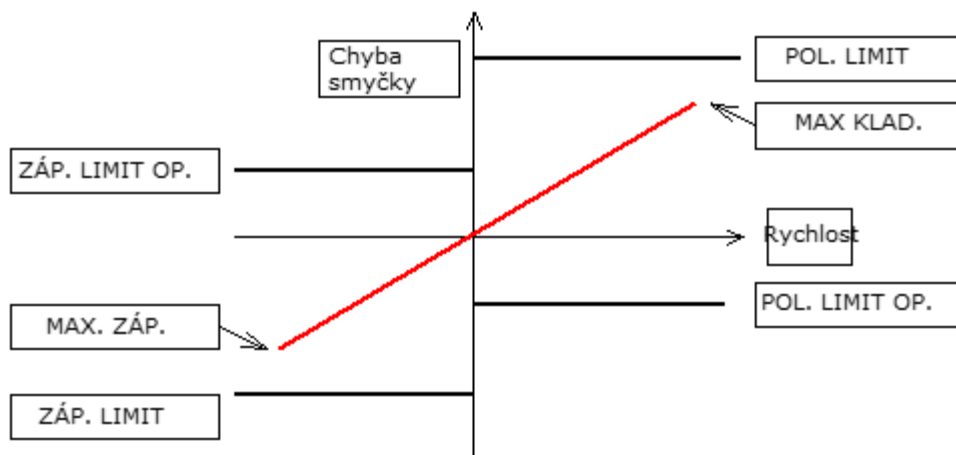
Tato instrukce umožňuje nastavit **druh** zkoušky chyby servořízení. Klasická správa chyby servořízení počítá s dvojicí limitů (kladný a záporný), které jsou při změnách rychlosti jednotky pohybu v ose konstantní. Tento druh správy vede k navržení velikosti uvedených limitů v závislosti na maximální rychlosti jednotky pohybu v ose, tj. nastaví se takový limit, aby nedocházelo k zásahu chyby v běžných podmínkách činnosti. Při nízkých hodnotách chyby smyčky má obvykle mnohem nižší hodnoty, než je nastavený limit, což znamená zpoždění při identifikaci chybového stavu.

Správa chyby servořízení prostřednictvím okna je založena na výpočtu chyby teoretické smyčky. Kladný a záporný limit chyby servořízení se vypočítávají v závislosti na tomto výpočtu a po odečtení mezní hodnoty. Když reálná hodnota chyby smyčky překročí tuto mezní hodnotu, dojde ke vzniku chyby servořízení.

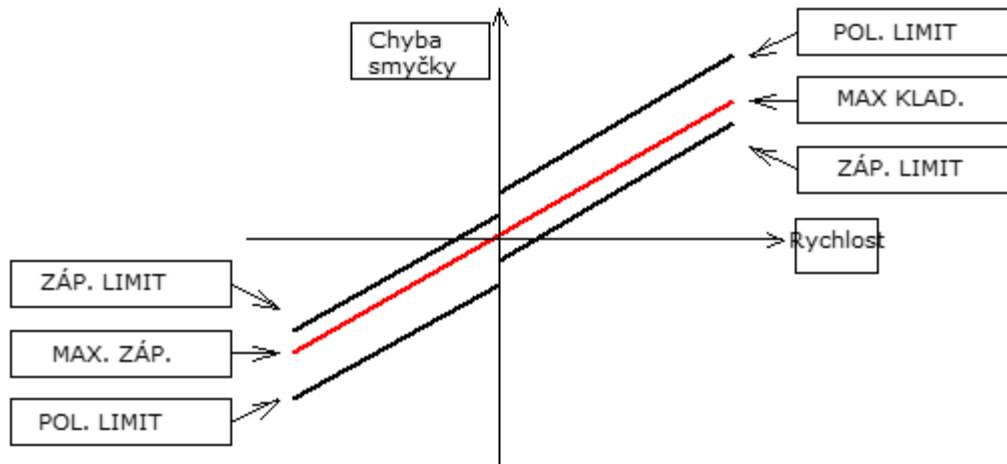
Poznámka

Při nastavení zkoušky dynamické chyby servořízení je obvykle třeba změnit hodnoty Kladný limit chyby servořízení a Záporný limit chyby servořízení, nastavené v Konfiguraci jednotek pohybu v osách pro chybu servořízení s mezní hodnotou. To proto, že výše uvedené hodnoty se používají jako počáteční hodnoty pro výpočet chyby smyčky.

Limit Chyby Servořízení s mezní hodnotou



Limit Dynamické Chyby Servořízení



SETPHASESINV

Syntaxe

SETPHASESINV

jednotkapohybuvose, stav

Argumenty

**jednotkapohybuvose
stav**

název zařízení druhu jednotka pohybu v ose
přednastavená konstanta. Přípustnými hodnotami jsou:
ON stav inverze fází zapnutý
OFF stav inverze fází vypnutý

Popis

Zapne nebo vypne na uvedené **jednotcepohybuvose** inverzi fází. Umožňuje kompenzovat prostřednictvím softwaru případnou inverzi kabeláže fází snímače impulzů. V případě použití tohoto parametru spolu s inverzí řídicího signálu umožňuje provést inverzi směru jednotky pohybu v ose (za předpokladu správnosti provedení kabeláže).

Za účelem provedení této instrukce se jednotka pohybu v ose musí nacházet ve stavu FREE.

SETREFINV

Syntaxe

SETREFINV

jednotkapohybuvose, stav

Argumenty

**jednotkapohybuvose
stav**

název zařízení druhu jednotka pohybu v ose
přednastavená konstanta. Přípustnými hodnotami jsou:
ON aktivuje inverzi řídicího signálu rychlosti
OFF zakazuje inverzi řídicího signálu rychlosti

Popis

Aktivuje nebo zakazuje na uvedené jednotce pohybu v ose inverzi řídicího signálu rychlosti. V případě použití tohoto parametru spolu s inverzí fází umožňuje provést inverzi směru jednotky pohybu v ose (za předpokladu správnosti provedení kabeláže).

Za účelem provedení této instrukce se jednotka pohybu v ose musí nacházet ve stavu FREE.

Viz také [SETPHASESINV](#).

SETRESOLUTION

Syntaxe

SETRESOLUTION

jednotkapohybuvose [, hodnota]

Argumenty

**jednotkapohybuvose
hodnota**

název zařízení druhu jednotka pohybu v ose
konstanta nebo proměnná double

Popis

Slouží ke změně rozlišení uvedené jednotky pohybu v ose. Když je **hodnota** vynechána, bude použita hodnota rozlišení nastavená v konfiguraci. Změna hodnoty rozlišení může být provedena pouze při zastavené

jednotce pohybu v ose (stav jednotky pohybu v ose = poloha); v opačném případě bude vytvořena chyba systému č. 4101 „Nevyhovující správa jednotky pohybu v ose“.

10.3.5 Počítadla

DECOUNTER

Syntaxe

DECOUNTER **nazevpocitadla [, hodnota]**

Argumenty

nazevpocitadla název zařízení druhu počítadla
valore konstanta nebo proměnná nebo zařízení druhu počítadlo

Popis

Slouží ke snížení hodnoty počítadla **nazevpocitadla** o uvedený parametr **hodnota**. Když je parametr **hodnota** vynechán, nabude hodnoty 1. Viz také instrukce [SETCOUNTER](#) a [INCOUNTER](#).

INCOUNTER

Syntaxe

INCOUNTER **nazevpocitadla [, hodnota]**

Argumenty

nazevpocitadla název zařízení druhu počítadla
valore konstanta nebo proměnná nebo zařízení druhu počítadlo

Popis

Slouží ke zvýšení hodnoty počítadla **nazevpocitadla** o uvedený parametr **hodnota**. Když je parametr **hodnota** vynechán, nabude hodnoty 1. Viz také instrukce [SETCOUNTER](#) a [DECOUNTER](#).

SETCOUNTER

Syntaxe

SETCOUNTER **nazevpocitadla [, hodnota]**

Argumenty

nazevpocitadla název zařízení druhu počítadla
valore konstanta nebo proměnná nebo zařízení druhu počítadlo

Popis

Slouží k nastavení počítadla **nazevpocitadla** na uvedený parametr **hodnota**.
Viz také [INCOUNTER](#) a [DECOUNTER](#).

10.3.6 Časovače

HOLDTIMER

Syntaxe

HOLDTIMER **nazevcasovace**

Argumenty

nazevcasovace název zařízení druhu časovač

Popis

Blokuje aktualizaci časovače **nazevcasovace**.
Viz také [STARTTIMER](#) a [SETTIMER](#).

SETTIMER

Syntaxe

SETTIMER **nazevcasovace, doba**

Argumenty

nazevcasovace název zařízení druhu časovač
hodnota konstanta nebo proměnná nebo zařízení druhu časovač

Popis

Slouží k nastavení **nazevcasovace** na uvedenou **dobu** (v sekundách). Jsou přípustné hodnoty větší než nula. Maximální přesnost časovače je 4 ms. Viz také [STARTTIMER](#) a [HOLDTIMER](#)

Příklad

```
; Function nastaví časovač
; nastaví se časovač CasovyLimit
; na hodnotu 20 sekund
SETTIMER CasovyLimit,20
; časovač je spuštěn v klesajícím režimu. když dosáhne 0, zastaví se

STARTTIMER CasovyLimit,DOWN
```

STARTTIMER

Syntaxe

STARTTIMER **nazevcasovace** [, **smer**]

Argumenty

nazevcasovace název zařízení druhu časovač
smer přednastavená konstanta. Přípustnými hodnotami jsou:
UP stoupající
DOWN klesající

Popis

Slouží ke spuštění časovače **nazevcasovace** způsobem případně vymezeným parametrem **smer**. Když je parametr **smer** vynechán, bude použit způsob **DOWN**. Když časovač (spuštěný v klesajícím režimu) dojde na nulu, automaticky se zastaví. Viz také [HOLDTIMER](#) a [SETTIMER](#).

10.3.7 Proměnné, vektory a matice

CLEAR

Syntaxe

CLEAR **nazevpromenne** nebo **vektor** nebo **matrice**[**radekmatrice**]

Argumenty

nazevpromenne název proměnné
vektor název vektoru
matrice název matrice
radekmatrice konstanta nebo promenna nebo pocitadlo. Řádek matrice

Popis

Inicializuje na hodnotu 0 prostor v paměti vyhrazený pro proměnné (**nazevpromenne**), vektory (**vektor**), matrice (**matrice**) nebo prvky řádku matrice.

FIND

Syntaxe

FIND **matrice, sloupec, min_limit, max_limit, hodnota, promenna**
FIND **vektor, min_limit, max_limit, hodnota, promenna**

Argumenty

matrice název matrice. Matrice, ve které je prováděno vyhledání
vektor název vektoru. Vektor, ve kterém je prováděno vyhledání
sloupec konstanta nebo proměnná nebo název počítadla. Číslo sloupce matrice, ve kterém má být provedeno vyhledání
min_limit konstanta nebo proměnná. Minimální index vektoru nebo matrice, ze kterého se má zahájit provádění vyhledávání
max_limit konstanta nebo proměnná. Maximální index vektoru nebo matrice, ve kterém se má ukončit provádění vyhledávání
hodnota konstanta nebo proměnná. Vyhledávaná hodnota
promenna proměnná. Výsledek vyhledávání

Popis

Slouží k provedení sekvenčního vyhledávání hodnoty uvnitř **vektoru** nebo **sloupce matrice** a uloží index prvku do **promenna**.
Když hodnota nebyla nalezena, **promenna** bude obsahovat hodnotu -1.

FINDB

Syntaxe

FINDB **matrice, sloupec, min_limit, max_limit, hodnota, promenna**
FINDB **vektor, min_limit, max_limit, hodnota, promenna**

Argumenty

matrice	název matrice. Matrice, ve které je prováděno vyhledání
vektor	název vektoru. Vektor, ve kterém je prováděno vyhledání
sloupec	konstanta nebo proměnná nebo název počítadla. Číslo sloupce matrice, ve kterém má být provedeno vyhledání
min_limit	konstanta nebo proměnná. Minimální index vektoru nebo matrice, ze kterého se má zahájit provádění vyhledávání
max_limit	konstanta nebo proměnná. Maximální index vektoru nebo matrice, ve kterém se má ukončit provádění vyhledávání
hodnota	konstanta nebo proměnná. Vyhledávaná hodnota
promenna	proměnná. Výsledek vyhledávání

Popis

Slouží k provedení rychlého vyhledání uvnitř **vektoru** nebo **sloupce matrice** a uloží index prvku do **promenna**. Aby bylo vyhledávání úspěšné, je třeba, aby byl **vektor** nebo **sloupec matrice** předtím uspořádán prostřednictvím instrukce SORT ve vzestupném pořadí.
Když hodnota nebyla nalezena, parametr **promenna** bude obsahovat hodnotu -1.

LASTELEM

Syntaxe

LASTELEM **vektor, prvkyvektoru**
LASTELEM **matrice, radkymatrice**

Argumenty

matrice	název matrice
vektor	název vektoru
prvkyvektoru	proměnná. Počet prvků vektoru
radkymatrice	proměnná. Počet řádků matrice

Popis

Zapíše do proměnné **prvkyvektoru** počet prvků **vektoru** nebo do proměnné **radkymatrice** počet řádků **matrice**.

LOCAL

Syntaxe

LOCAL **nazevpromenne AS druh**
LOCAL **vektor[pocet prvku] AS druh**
LOCAL **matrice[pocet radku] AS druh, druh, druh apod.**
LOCAL **matrice[pocet radku] AS druh:nazevs1, druh:nazevs2, druh:nazevs3 apod.**

Argumenty

nazevpromenne	název proměnné
[pocet prvku]	proměnná nebo konstanta (povinný argument). Počet prvků vektoru
[pocet radku]	konstanta nebo proměnná (povinný argument). Počet řádků matrice
druh	char, integer (32 bit), float (32 bit), double (64 bit), string, timer
nazevs1...nazevsN	název sloupce. Návěští

Popis

Deklarování místní proměnné. Před použitím této instrukce se může objevit pouze instrukce PARAM, která definuje parametry funkce.
Ohledně podrobnějších informací o místních proměnných viz [Místní proměnné](#).

MOVEMAT

Syntaxe

MOVEMAT	nazevzdrojovematrice, nazevcilovematrice
MOVEMAT	nazevzdrojovematrice[zdrojovyradek],
	nazevcilovematrice[cilovyradek]
MOVEMAT	nazevzdrojovematrice[zdrojovyradek],
	nazevcilovematrice[cilovyradek], pocetradku

Argumenty

nazevzdrojovematrice	název zdrojové matrice
zdrojovyradek	číslo výchozího řádku kopírování zdrojové matrice (povinný argument)
nazevcilovematrice	název cílové matrice
cilovyradek	číslo výchozího řádku pro kopírování do cílové matrice (povinný argument)
pocetradku	počet řádků určených ke kopírování

Popis

Kopíruje obsah celé matrice **nazevzdrojovematrice** do jiné matrice **nazevcilovematrice** nebo jeden či více řádků **pocetradku** od řádku matrice **nazevzdrojovematrice** na řádek matrice **nazevcilovematrice[cilovyradek]**. Když parametr **pocetradku** není uveden, bude kopírován pouze jeden řádek. Obě matrice musí mít stejný druh struktury (počet sloupců a druh data každého sloupce) a v případě celých matric musí mít také stejný počet řádků. Je možné přesouvat řádky dat uvnitř stejné matrice.

Příklad

```

Movemat Mx1, Mx2 ; kopíruje matici Mx1 do Mx2

; kopíruje řádek 10 matrice Mx1 do řádku 3 matrice Mx2
Movemat Mx1[10], Mx2[3]

; kopíruje řádek 1 matrice Mx1 do řádku 7 matrice Mx1
Movemat Mx1[1], Mx1[7]
; kopíruje 6 řádků počínaje řádkem 2 matrice Mx1 do matrice
; Mx2 počínaje řádkem 8
Movemat Mx1[2], Mx2[8], 6

; kopíruje 4 řádky počínaje řádkem 2
; matrice Mx1 do stejné matrice
; Mx1 počínaje řádkem 10
Movemat Mx1[2], Mx1[10], 4;

```

PARAM

Syntaxe

[PARAM]	nazevpromenne AS druh
[PARAM]	vektor[pocet prvku] AS druh
[PARAM]	matrice[pocet radku] AS druh, druh, druh apod
[PARAM]	matrice[pocet radku] AS druh:alias, druh:alias, druh:alias apod

Argumenty

nazevpromenne	název proměnné
[pocet prvku]	konstanta (povinný argument)
[pocet radku]	konstanta (povinný argument)
druh	char, integer (32 bitů), float (32 bitů), double (64 bitů), string, timer

Popis

Parametry se chovají jako místní proměnné (viz [LOCAL](#)), ale jsou inicializovány tím, kdo vyvolal funkci. Syntaxe pro deklarování parametrů je stejná jako ta, která se používá pro místní proměnné. Parametry jsou předávány prostřednictvím hodnoty nebo prostřednictvím odkazu ve funkci jejich druhu. Viz „[Funkce](#)“. Je třeba deklarovat před jakoukoli jinou instrukcí. Ohledně podrobnějších informací viz [Místní proměnné](#).

SETVAL

Syntaxe

SETVAL	hodnota, nazevpromenne
---------------	-------------------------------

Argumenty

hodnota	konstanta nebo proměnná nebo nazevzarizeni
nazevpromenne	proměnná nebo nazevzarizeni

Popis

Přirazuje uvedenou **hodnotu** proměnné **nazevpromenne** nebo i-tému prvku vektoru nebo matrice.

SORT**Syntaxe**

SORT **matrice, sloupec [, poradi], min_limit, max_limit**
SORT **vektor [, poradi], min_limit, max_limit**

Argumenty

matrice	název matrice
vektor	název vektoru
sloupec	konstanta nebo proměnná. Číslo sloupce matrice
poradi	přednastavená konstanta. Označuje způsob uspořádání Přípustnými hodnotami jsou: UP vzestupné uspořádání DOWN sestupné uspořádání
min_limit	konstanta nebo proměnná. Minimální index vektoru nebo matrice, ze kterého se má zahájit provádění uspořádání
max_limit	konstanta nebo proměnná. Maximální index vektoru nebo matrice, ze kterého se má zahájit provádění uspořádání

Popis

Provede uspořádání hodnot přítomných uvnitř **vektoru** nebo **matrice** se směrem určeným prostřednictvím konstanty **poradi**.

V případě matrice je uspořádání řádků stanoveno vzestupným (UP) nebo sestupným (DOWN) uspořádáním hodnot obsažených ve zvoleném **sloupci**.

Když je argument **poradi** vynechán, bude automaticky přijat vzestupný způsob uspořádání - UP.

**10.3.8 Řetězce****ADDSTRING****Syntaxe**

ADDSTRING **nazevretezce1, nazevretezce2, nazevretezce3**

Argumenty

nazevretezce1	konstanta nebo proměnná řetězec. Zdrojový řetězec
nazevretezce2	konstanta nebo proměnná řetězec. Řetězec, který má být přidán
nazevretezce3	proměnná řetězec. Výsledný řetězec

Popis

Řetězení dvou řetězců.

Přidá k řetězci identifikovanému prostřednictvím **nazevretezce1** řetězec identifikovaný prostřednictvím **nazevretezce2** a umístí výsledek do řetězce identifikovaného jako **nazevretezce3**.

Maximální rozměr řetězce je 255 znaků + ukončovací znak řetězce, a proto výsledek řetězení prvních dvou řetězců nebude moci překročit tento limit.

Příklad

Viz příklad [Operace na řetězcích](#).

CONTROLCHAR

Syntaxe

CONTROLCHAR **hodnota, nazevretezce**

Argumenty

hodnota konstanta char nebo integer nebo proměnná char nebo integer. Hodnota určená ke konverzi
nazevretezce proměnná řetězec. Výsledný řetězec

Popis

Tato instrukce slouží ke konverzi hodnoty identifikované prostřednictvím parametru **hodnota** na znak ASCII a uloží výsledek do řetězce **nazevretezce** (prakticky do prvního bytu).

Předchozí obsah řetězce bude ztracen. Tato instrukce je užitečná v případě, že je třeba do řetězce vložit kontrolní nebo netisknutelné znaky (například znak NULL = 0x00).

Akceptuje řetězce nejméně se 2 znaky: 1 znak + ukončovací znak. Když je řetězec tvořen jediným znakem `array[1] as char`, bude signalizována chyba systému „Chybný argument makra“.

Příklad

Viz příklad [Operace na řetězcích](#).

LEFT

Syntaxe

LEFT **navezdrojovehoretezce, pocetznaku, nazevcilovehoretezce**

Argumenty

navezdrojovehoretezce konstanta nebo proměnná řetězec. Zdrojový řetězec
pocetznaku konstanta nebo proměnná. Počet znaků, které mají být kopírovány
nazevcilovehoretezce proměnná řetězec. Cílový řetězec

Popis

Kopíruje první **pocetznaku** řetězce **navezdrojovehoretezce** do řetězce **nazevcilovehoretezce**.

Prakticky je odebrána levá část zdrojového řetězce. Viz také instrukce [MID](#) a [RIGHT](#).

Příklad

Viz příklad [Operace na řetězcích](#).

LEN

Syntaxe

LEN **nazevretezce, promenna**

Argumenty

nazevretezce proměnná řetězec. Řetězec
promenna proměnná

Popis

Slouží k výpočtu znaků obsažených v řetězci **nazevretezce** (s výjimkou terminátoru) a uloží výsledek do argumentu **promenna**.

Příklad

Viz příklad [Operace na řetězcích](#).

MID

Syntaxe

MID **navezdrojovehoretezce, prvniznak [, pocetznaku], nazevcilovehoretezce**

Argumenty

navezdrojovehoretezce konstanta nebo proměnná řetězec. Zdrojový řetězec
pocetznaku konstanta nebo proměnná. Počet znaků, které mají být kopírovány
nazevcilovehoretezce proměnná řetězec. Cílový řetězec
prvniznak konstanta nebo proměnná. Poloha znaku zahájení kopírování

Popis

Provádí extrakci z řetězce identifikovaného argumentem **nazevzdrojovehoretezce** počtu znaků určených argumentem **pocetznaku**, počínaje polohou **prvniznak**.

Vyjmutý podřetězec je umístěn do řetězce identifikovaného argumentem **nazevcilovehoretezce**.

Když je parametr **pocetznaku** vynechán, bude provedeno kopírování **nazevzdrojovehoretezce** počínaje polohou **prvniznak** až do konce řetězce. Prakticky je odebrána střední část zdrojového řetězce. Viz také instrukce [LEFT](#) a [RIGHT](#).

Příklad

Viz příklad [Operace na řetězcích](#).

RIGHT

Syntaxe

RIGHT **nazevzdrojovehoretezce, pocetznaku, nazevcilovehoretezce**

Argumenty

nazevzdrojovehoretezce konstanta nebo proměnná řetězec. Zdrojový řetězec
pocetznaku konstanta nebo proměnná. Počet znaků, které mají být kopírovány
nazevcilovehoretezce proměnná řetězec. Cílový řetězec

Popis

Kopíruje poslední **pocetznaku** řetězce **nazevzdrojovehoretezce** do řetězce **nazevcilovehoretezce**. Po praktické stránce je odebrána pravá část zdrojového řetězce. Viz také instrukce [LEFT](#) a [MID](#).

Příklad

Viz příklad [Operace na řetězcích](#).

SEARCH

Syntaxe

SEARCH **nazevretezce, znak, promenna**

Argumenty

nazevretezce proměnná řetězec.
znak konstanta char nebo konstanta řetězec nebo proměnná řetězec. Znak nebo řetězec k vyhledání
promenna proměnná

Popis

Vyhledá polohu znaku ASCII identifikovaného parametrem **znak** (kterým může být také řetězec) uvnitř řetězce **nazevretezce** a uloží index výsledku do parametru **promenna**. Když **znak** nebyl nalezen, **promenna** bude obsahovat hodnotu -1.

Příklad

Viz příklad [Operace na řetězcích](#).

SETSTRING

Syntaxe

SETSTRING **"hodnota", nazevretezce**

Argumenty

hodnota konstanta řetězec nebo proměnná řetězec (uzavřená v uvozovkách)
nazevretezce cílový řetězec

Popis

Provádí kopírování řetězce.

Slouží k provedení kopírování znaků ASCII, které se nacházejí v řetězci nazvaném **"hodnota"**, do řetězce identifikovaného názvem **nazevretezce**.

Ohledně vložení znaků, které nelze tisknout, do řetězce viz [CONTROLCHAR](#).

Příklad

Viz příklad [Operace na řetězcích](#).

STR

Syntaxe

STR **hodnota, nazevretezce**

Argumenty

hodnota konstanta nebo proměnná. Zdrojová hodnota určená ke konverzi
nazevretezce proměnná řetězec. Cílový řetězec

Popis

Slouží ke konverzi parametru **hodnota** na znaky ASCII a na uložení výsledku do řetězce **nazevretezce**. Lze ji použít také pro transformaci proměnné druhu integer na řetězec. Například číslo 10 bude přetransformováno na řetězec „10“.

Příklad

Viz příklad [Operace na řetězcích](#).

VAL

Syntaxe

VAL **nazevretezce, vysledek**

Argumenty

nazevretezce proměnná řetězec. Řetězec určený ke konverzi
vysledek promenna. Přetransformovaný řetězec

Popis

Trasformuje obsah řetězce **nazevřetězce** na desetinné číslo a uloží výsledek do parametru **vysledek**. Například řetězec „123“ bude transformován na hodnotu 123.

Příklad

Viz příklad [Operace na řetězcích](#).

10.3.9 Komunikace

CLEARRECEIVE

Syntaxe

CLEARRECEIVE

Argumenty

žádný argument

Popis

Slouží k vyprázdnění seznamu provedených, ale dosud neuspokojených RECEIVE.

COMCLEARRXBUFFER

Syntaxe

COMCLEARRXBUFFER **cisloCOM**

Argumenty

cisloCOM přednastavená konstanta. Číslo sériového portu. Hodnoty, které mohou být přiřazeny, jsou: od **COM1** do **COM8**

Popis

Tato instrukce vyprázdní vyrovnávací paměť příjmu sériového vedení **cisloCOM**. Všechna případně přítomná data budou vymazána.

COMCLOSE

Syntaxe

COMCLOSE **cisloCOM**

Argumenty

cisloCOM přednastavená konstanta. Číslo sériového portu. Hodnoty, které mohou být přiřazeny, jsou: od **COM1** do **COM8**

Popis

Slouží k zavření sériového vedení **CisloCOM** otevřeného instrukcí **COMOPEN**. Sériové vedení je třeba zavřít i v případě, že bude úloha, která otevřela sériový port, z jakýchkoli příčin ukončena.

COMGETERROR**Syntaxe**

COMGETERROR **cisloCOM, promenna**

Argumenty

cisloCOM přednastavená konstanta. Číslo sériového portu. Hodnoty, které mohou být přiřazeny, jsou: od **COM1** do **COM8**

promenna proměnná integer. Výsledek poslední operace provedené na sériové lince

Popis

Instrukce čte návratový kód poslední instrukce sériové komunikace vyvolané na portu **cisloCOM**. Prostřednictvím této instrukce je možné zjistit, zda operace čtení nebo zápisu byla úspěšná, a v záporném případě také návratový kód chyby. Níže jsou uvedeny kódy chyb:

Běžný návrat	0
Planá vyrovnávací paměť	2
Zařízení již otevřené	3
Neplatný nebo nekonfigurovaný port	6
Nezdařená aktivace portu VST./VÝST.	7
Chybějící možnost připojit se k přerušení (interrupt)	8
Ještě neotevřený sériový port (com)	9
Obsazené sériové zařízení (com)	12
Chybějící možnost připojit se k RTX	14

COMGETRXCOUNT**Syntaxe**

COMGETRXCOUNT **cisloCOM, pocetznaku**

Argumenty

cisloCOM přednastavená konstanta. Číslo sériového portu. Hodnoty, které mohou být přiřazeny, jsou: od **COM1** do **COM8**

pocetznaku počet znaků přítomných ve vyrovnávací paměti

Popis

Tato instrukce vrací informaci o počtu znaků přítomných ve vyrovnávací paměti. Umožňuje zjistit, zda byly přijaty znaky ze sériové linky.

COMOPEN**Syntaxe**

COMOPEN **čísloCOM, přenosová rychlost, velikostslova, stopbit, parita**

Argumenty

cisloCOM přednastavená konstanta. Číslo sériového portu. Hodnoty, které mohou být přiřazeny, jsou: od **COM1** do **COM8**

prenosovarychlost přenosová rychlost komunikace. K hodnotám, které lze přiřadit, patří: 2400, 4800, 9600, 19200, 38400, 57600, 115200

velikostslova rozměr slova dat. K přiřaditelným hodnotám patří: 5, 6, 7, 8

stopbit stop bit. K přiřaditelným hodnotám patří: 1, 2

parita přednastavená konstanta. Parita. K přiřaditelným hodnotám patří: **NOPARITY, ODDPARITY** a **EVENPARITY**

Popis

Slouží k otevření sériové linky. Je třeba ji provést před jakoukoli jinou instrukcí, která provádí správu sériové linky. V případě provedení jiné instrukce týkající se stejné sériové linky před **COMOPEN** bude vytvořena chyba systému. Předané parametry musí mít hodnoty, které se nacházejí ve výše uvedených rozsazích. Sériový komunikační kanál je přiřazen úloze, která provedla instrukci **COMOPEN**. Když úloha skončí, komunikační kanál bude automaticky zavřen.

Viz také [COMCLOSE](#), [COMREAD](#), [COMWRITE](#), [COMREADSTRING](#), [COMWRITESTRING](#).

Poznámka

Počet dostupných sériových linek závisí od hardwarového prostředí číslicového řízení (přečtěte si dokumentaci). V prostředí RTX jsou k dispozici pouze COM1 a COM2.

COMREAD

Syntaxe

COMREAD

cisloCOM, vyrovnavacipamet, pocetznakukectení, pocetprectenychznaku [,casovylimít]

Argumenty

cisloCOM

přednastavená konstanta. Číslo sériového portu. Hodnoty, které mohou být přiřazeny, jsou: od **COM1** do **COM8**

vyrovnavacipamet

vektor znaku. Jedná se o vektor, kde jsou ukládána skutečně přečtených znaků

pocetznakukectení

počet znaků, které mají být přečteny ze sériové linky

pocetprectenychznaku

počet skutečně přečtených znaků

casovylimít

časový limit čekání (v sekundách)

Popis

Tato instrukce čte znaky ze sériové linky **CisloCOM**. Znaky jsou ukládány do paměti prostřednictvím proměnné **vyrovnavacipamet**. Pole **Pocetznakukectení** označuje počet znaků, které má instrukce přečíst. Když se v přijímací vyrovnávací paměti sériové linky nachází méně znaků a parametr **casovylimít** není uveden, instrukce bude okamžitě ukončena a v parametru **pocetprectenychznaku** uvede počet skutečně přečtených znaků. Když je parametr **casovylimít** uvedený, instrukce bude muset čekat na příchod dalších znaků maximálně počet sekund uvedený v této proměnné. Když uplyne **casovylimít**, instrukce bude ukončena a v parametru **Pocetprectenychznaku** uvede počet znaků, které byly skutečně kopírovány do vyrovnávací paměti.

COMREADSTRING

Syntaxe

COMREADSTRING

cisloCOM, vyrovnavacipamet, pocetprectenychznaku [,terminator [,casovylimít]]

Argumenty

cisloCOM

přednastavená konstanta. Číslo sériového portu. Hodnoty, které mohou být přiřazeny, jsou: od **COM1** do **COM8**

vyrovnavacipamet

vektor znaku. Jedná se o vektor, kde jsou ukládána data.

pocetprectenychznaku

počet skutečně přečtených znaků

terminator

ukončovací znak přenosu

casovylimít

časový limit čekání (v sekundách)

Popis

Tato instrukce čte znaky ze sériové linky **CisloCOM**. Od instrukce **COMREAD** se liší v tom, že provádí čtení ze sériové linky dokud nenajde ukončovací znak. Znaky jsou ukládány do paměti prostřednictvím proměnné **vyrovnavacipamet**. Tato proměnná musí být typu vektor znaku. Pole **pocetprectenychznaku** označuje počet znaků, které instrukce opravdu přečetla ze sériové linky a který byl skutečně kopírován do **vyrovnavacipaměti**. Parametr **terminator** označuje znak, který bude sloužit jako ukončovací znak přenosu. V praxi to znamená, že instrukce bude muset číst znaky ze sériové linky dokud nenarazí na znak rovnající se znaku uvedenému v tomto parametru. Tento parametr je volitelný. Když není uveden, interpretuje se tak, že ukončovacím znakem je nula. Znak nula nebude kopírován do vyrovnávací paměti v úloze parametru, zatímco případný specifický ukončovací znak uvedený v instrukci ano. **Casovylimít** je dalším parametrem, který označuje, kolik sekund má instrukce čekat na příchod nových znaků, když vyprázdnila přijímací vyrovnávací paměť, aniž by našla ukončovací znak. Když parametr **casovylimít** není uveden, instrukce bude ukončena bezprostředně po vyprázdnění přijímací vyrovnávací paměti.

COMWRITE

Syntaxe

COMWRITE

cisloCOM, vyrovnavacipamet, pocetznakukzapisu

Argumenty

cisloCOM

přednastavená konstanta. Číslo sériového portu. Hodnoty, které mohou být přiřazeny, jsou: od **COM1** do **COM8**

vyrovnavacipamet

vektor znaku. Jedná se o vektor obsahující data určená k zápisu

pocetznakukzapisu

počet znaků, které mají být zapsány

Popis

Tato instrukce zapisuje znaky, které se nacházejí v proměnné **vyrovnavacipamet** na sériové lince **cisloCOM**. V parametru **pocetznakuzapisu** bude uveden počet znaků skutečně určených k zápisu.

COMWRITESTRING

Syntaxe

COMWRITESTRING **cisloCOM, vyrovnavacipamet [,terminator]**

Argumenty

cisloCOM	přednastavená konstanta. Číslo sériového portu. Hodnoty, které mohou být přiřazeny, jsou: od COM1 do COM8
vyrovnavacipamet	vektor znaku. Jedná se o vektor obsahující data určená k zápisu
terminator	ukončovací znak přenosu

Popis

Tato instrukce zapisuje znaky, které se nacházejí v proměnné **vyrovnavacipamet** na sériové lince **cisloCOM**. Od instrukce **COMWRITE** se liší v tom, že provádí zápis na sériovou linku dokud nenajde **terminator**. Parametr **terminator** je volitelný. Když není uveden, instrukce bude uskutečňovat přenos, dokud nenarazí na znak nuly. Nula nebude přenesena, zatímco případný specifický ukončovací znak ano.

RECEIVE

Syntaxe

RECEIVE **[zdroj,] identifikator, priznaky [, kontejner]**

Argumenty

zdroj	konstanta druhu řetězec
identifikator	konstanta druhu řetězec
priznaky	konstanta druhu integer
kontejner	název zařízení nebo proměnné (číselné nebo v podobě řetězce)

Popis

Tato instrukce se spolu i instrukcí **SEND** používá pro výměnu informací mezi moduly výrobního zařízení a dozorovým PC. **SEND** se používá pro odesílání informací a **RECEIVE** pro jejich příjem. Informace mohou být vyžádány od softwaru Albatros nebo od externího programu (Server OLE Automation). Ve druhém případě bude požadavek přijat softwarem Albatros, který zajistí jeho odeslání externímu programu.

Parametr **zdroj** je tvořen řetězcem, který umožňuje určit, komu je požadavek na informace určen. Existují tři třídy adresátů:

- Zdroje začínají znakem „@“ (viz níže uvedený seznam). Zdrojem je ve skutečnosti software Albatros nebo přesněji jedna z jeho funkcí.
- Zdroje, které nezačínají znakem „@“. Berou se jako Server OLE; při prvním požadavku na informace, který je jim adresován, se software Albatros bude snažit spustit je a poté jim předat požadavek na informace přijatý od modulu.
- Neurčený zdroj (parametr je skutečně volitelný). V tomto případě je informace přečtena v tabulce uložené v softwaru Albatros. Když informace není přítomna v tabulce, požadavek zůstane nevyřízený a bude uspokojen, jakmile se informace stane dostupnou (dodaná jiným modulem nebo externím programem).

Parametr **identifikator** představuje název požadované informace, a proto nemůže být vynechán. Nabývá různých významů v závislosti na funkci zdroje:

- Když je zdrojem software Albatros, bude se jednat o příkaz související s funkcí, ke které je prováděn přístup
- když je zdrojem Server OLE, bude se jednat o vlastnost požadovaného objektu OLE.
- když zdroj není uveden, bude se jednat o návštěvu, které identifikuje informaci v tabulce uskladněné v softwaru Albatros.

Parametr **priznaky** umožňuje uvést, jak má s požadovanou informací naložit Albatros. Níže jsou uvedeny přípustné hodnoty a příslušné odezvy:

hodnota	příkaz	popis
\$0008H	CancelAfter	Informace bude po přečtení vymazána
\$0800H	UpdateFlags	Změní stav informace (přečtená / ke čtení) bez změny dat
\$8000H	Delete	Vymaže informaci

Parametr **kontejner** tvoří proměnná (nebo zařízení), do které bude uložena požadovaná informace. Může být vynechána a v takovém případě bude požadováno oznámení udalosti (může být použita pro synchronizaci provádění kódu GPL na různých modulech).

Seznam **zdrojů** ve správě softwaru Albatros a příslušné příkazy:

„@List“

Umožňuje správu příkazů Sim (Simulovany) a Setp (Nastavená hodnota - Setpoint).

Jsou přípustné níže uvedené příkazy (parametr **identifikátor**):

- Sim,0,kontejner: Vyžaduje stav tlačítka Simulovaný, který je zapsán do příznaku flagSwitch Simulovany. Zpětná proměnná **kontejner** má hodnotu 1, jestliže nebyly zaznamenány chyby; v opačném případě má hodnotu 0.
- Setp,0,kontejner: Vyžaduje stav tlačítka Nastavená hodnota, který je zapsán do příznaku flagSwitch CmdSetP. Zpětná proměnná **kontejner** má hodnotu 1, jestliže nebyly zaznamenány chyby; v opačném případě má hodnotu 0.
- Esc,0,kontejner: Vyžaduje stav tlačítka Nastavená hodnota, který je zapsán do příznaku flagSwitch Vyrazeny. Zpětná proměnná **kontejner** má hodnotu 1, jestliže nebyly zaznamenány chyby; v opačném případě má hodnotu 0.

„@Environ“

Umožňuje získat informace o stavu systému: [úroveň přístupu](#) uživatele, moduly připojené k dozornému PC apod. Požadovaná informace je uložena do parametru **kontejner**. K přípustným hodnotám parametru **identifikátor** a k příslušným odpovědím patří:

„AccessLevel“ úroveň přístupu do systému 0=uživatel, 1=servisní služba, 2=výrobce, 3=tpa
 „MaskConfModules“ maska konfigurovaných modulů
 „MaskActiveModules“ maska připojených modulů
 „CurrentModule“ modul, od kterého pochází požadavek
 „mod:NamePC“ název PC odpovídající modulu „mod“. (mod v rozsahu od 0 do 15)
 „LocalDateTime“:

Parametr **kontejner** přijme datum a čas PC ve formátu souvisejícím s jeho typem:

- char: číslo dne v týdnu
- integer: počet sekund od 1. 1. 1970
- float: počet dnů a částí dne od 1. 1. 1900
- double: počet dnů a částí dne od 1. 1. 1900
- string: text „RRRR/MM/DD hh:mm:ss“

Masky připojených a konfigurovaných modulů jsou masky bitů. Bit s nejnižší vahou odpovídá modulu 0. Bit každého modulu bude mít hodnotu 1, jestliže je modul připojen a nakonfigurován. V případě „NamePC“ je číslo modulu fakultativní a v případě, že bude vynecháno, použije se číslo modulu, ze kterého vyšel požadavek.

„@Syn“

Komunikace mezi GPL a prohlížečem grafických zobrazení. Umožňuje otevření a zavření grafických zobrazení ovládané z GPL a požadavek na informace z pole grafického zobrazení. Jsou přípustné níže uvedené příkazy (parametr **identifikátor**):

„Open:nomefile“ otevření grafického zobrazení *nomefile.xsyn*
 „Close:nomefile“ zavření grafického zobrazení *nomefile.xsyn*

„nomecasella“ pole, ze kterého se čte požadovaná informace

Informace o okně pohybu jednotek pohybu v osách podle určených pravidel lze získat také prostřednictvím níže uvedeného parametru **zdroj** [„@Devices“](#).

„@FileName“

Slouží k uložení přiřazení mezi konstantou v podobě řetězce a názvem souboru sestavitelným pomocí proměnných druhu řetězec. Od okamžiku, ve kterém Albatros přijme komunikaci o přiřazení, nahradí všechny následující názvy souborů názvem přijatým prostřednictvím této instrukce. Parametr **identifikátor** představuje název souboru. Název souboru je tvořen proměnnou druhu řetězec. Když v parametru identifikátor není uvedena kompletní cesta pro archivaci souboru, Albatros bude za cestu považovat tu, která je přednastavena v souboru tpa.ini, v části [tpa], pod položkou dirreport. Hodnota parametru identifikátor je uložena do tpa.ini v části [GPLFileName] pod položkou Log, aby ji bylo možné použít i při následujících spuštěních softwaru Albatros. Za účelem odstranění přiřazení je třeba nastavit za parametr identifikátor prázdný řetězec. Takto definované přiřazení platí pro všechny moduly.

„@FileDelete“

Vymazání souboru. Parametr **identifikátor** představuje název souboru, který bude vymazán (kompletní cesta). Když v parametru identifikátor není uvedena kompletní cesta pro archivaci souboru, Albatros bude za cestu považovat tu, která je přednastavena v souboru tpa.ini, v části [tpa], pod položkou dirreport. Název souboru může být definován podle pravidel popsaných při parametru **zdroj** [@FileRead](#). Parametr **kontejner** bude obsahovat hodnotu:

- 1, jestliže byl soubor vymazán
- 0 v opačném případě

„@FileRead“

Slouží k přečtení obsahu souboru. Parametr **identifikátor** představuje název souboru, který bude přečten (kompletní cesta). Když v parametru identifikátor není uvedena kompletní cesta pro archivaci souboru, Albatros bude za cestu považovat tu, která je přednastavena v souboru tpa.ini, v části [tpa], pod položkou dirreport. Když identifikátor začíná a končí znakem %, řetězec uvnitř identifikátoru bude vyhledán

v tpa.ini, v části [tpa] a bude použit jako název souboru. Uvnitř názvu mohou být vloženy znaky, které budou nahrazeny během provádění instrukce:

- %n číslo modulu, který provede instrukci RECEIVE
- %h aktuální čas (formát 00-23)
- %d aktuální den (formát 01-31)
- %m aktuální měsíc (formát 01-12)
- %y aktuální rok (formát se čtyřmi číslicemi)

Když je parametr **kontejner** definován jako proměnná char, bude obsahovat jeden byte přečtený ze souboru; když je definován jako řetězec, bude obsahovat jeden celý řádek textového souboru; když je definován jako integer, bude obsahovat počet bytů, které chybí do konce souboru (0 = konec souboru). Za účelem přemístění kurzoru na začátek souboru musí být parametr **kontejner** vynechán.

„@FileExist“

Slouží k ověření existence souboru. Parametr **identifikátor** představuje název souboru, který bude přečten (kompletní cesta). Když v parametru identifikátor není uvedena kompletní cesta pro archivaci souboru, Albatros bude za cestu považovat tu, která je přednastavena v souboru tpa.ini, v části [tpa], pod položkou dirreport. Název souboru může být definován podle pravidel popsaných při parametru **zdroj @FileRead**. Parametr **kontejner** bude obsahovat hodnotu:

- Odlišnou od 0, jestliže soubor existuje
- 0, když soubor neexistuje

„@FileLastWrite“

Získá datum poslední provedené změny souboru. Parametr **identifikátor** představuje název souboru (kompletní cesta). Když v parametru identifikátor není uvedena kompletní cesta pro archivaci souboru Albatros, bude za cestu považovat tu, která je přednastavena v souboru tpa.ini, v části [tpa], pod položkou dirreport. Název souboru může být definován podle pravidel popsaných při parametru **zdroj @FileRead**. Parametr **kontejner** bude obsahovat datum poslední změny souboru ve formátu, který souvisí s druhem parametru:

- char: číslo dne v týdnu
- integer: počet sekund od 1. ledna 1970
- float: počet dnů a částí dne od 1. ledna 1900
- double: počet dnů a částí dne od 1. ledna 1900
- string: text ve formátu „RRRR/MM/DD hh:mm:ss“

„@FileInfo“

Slouží k přečtení informací ze souboru. Parametr **identifikátor** musí být vyjádřen ve formě „vlastnost:nazevsouboru“, kde **vlastnost** označuje název vlastnosti, která má být přečtena, a **nazevsouboru** je název souboru. Název souboru může být nastaven prostřednictvím položky „Name“. Parametr **kontejner** bude obsahovat data, přečtená ze souboru.

Seznam vlastností:

- „version:“: vrací do kontejneru údaj typu celé číslo (integer). Čtyři čísla, která identifikují verzi, se nacházejí ve 4 bytech proměnné kontejner. Když dojde k výskytu chyby, hodnota proměnné kontejner je 0.
- „size:“: vrací do kontejneru údaj typu celé číslo (integer), číslo s plovoucí čárkou (float) nebo dvojitě číslo (double). Údaj představuje rozměr souboru. Když dojde k výskytu chyby, hodnota proměnné kontejner je -1.

„@Devices“

Požadavek na otevření nebo zavření okna Diagnostika týkajícího se modulu, který odesílá informaci.

Parametr identifikátor může nabýt níže uvedených hodnot:

- „Open“ otevření Diagnostiky
- „Close“ zavření Diagnostiky

Parametr **identifikátor** může v případě, že má být prováděna interakce s oknem pro pohyb jednotky pohybu v ose, nabýt níže uvedených hodnot:

„MoveAX#nazev_jednpohvose#HasFocus“ Parametr **kontejner** bude obsahovat 1, když je okno pro pohybování uvedenou jednotkou pohybu v ose aktivní; v opačném případě bude obsahovat 0.

„MoveAX#nazev_jednpohvose#Jog“ Parametr **kontejner** bude obsahovat 1 při nastavení pohybu pro přesuny spravované v reálném čase (runtime) obsluhou; v opačném případě bude obsahovat 0.

„MoveAX#nazev_jednpohvose#Step“ Parametr **kontejner** bude obsahovat 1 při nastavení pohybu pro přesuny s přednastaveným krokem; v opačném případě bude obsahovat 0.

„MoveAX#nazev_jednpohvose#Absolute“ Parametr **kontejner** bude obsahovat 1 při nastavení pohybu s přesunem do určené polohy; v opačném případě bude obsahovat 0.

kde nome_asse představuje název jednotky pohybu v ose, která je zobrazena v okně. Když má být například ověřeno, zda je okno pro pohybování jednotkou pohybu v ose X aktivní, parametr **identifikátor** bude „@MoveAX#X#HasFocus“. Název jednotky pohybu v ose může mít jeden z níže uvedených tvarů:

1. Nazev_Skupiny.Nazev_Podskupiny.Nazev_Jednpohvose nebo Nazev_Skupiny.Nazev_Jednpohvose:
Bude dodána kompletní cesta jednotky pohybu v ose.

2. `Nazev_Jednpohvose`: Pro identifikaci správné jednotky pohybu v ose budou provedena níže uvedená ověření, a to v uvedeném pořadí:
- Když je úlohou, ze které pochází příkaz funkce podskupiny, jednotka pohybu v ose, bude hledána v té podskupině.
 - Když je úlohou, ze které pochází příkaz funkce hlavní podskupiny, jednotka pohybu v ose, bude hledána v celé skupině. Když je přítomno více jednotek pohybu v osách s daným názvem, hledání bude neúspěšné.
 - V případě, že se předchozí vyhledání nezdařila, jednotky pohybu v ose bude hledána ve skupinách modulu. Když je přítomno více jednotek pohybu v osách s názvem `Nazev_Jednpohvose`, hledání nebude úspěšné.

„@Vars“

Vyžaduje aktualizaci globální proměnné GPL. Umožňuje provést aktualizaci dat Technologických parametrů a nástrojů. Údaje parametrizování jsou obvykle odesílány do GPL během inicializace stroje. Parametr **identifikátor** nabude významu názvu globální proměnné (stroje nebo skupiny), jejíž aktualizace je vyžadována. Parametr **kontejner** bude obsahovat hodnotu:

- 1, když byla proměnná aktualizována správně
- 0 v opačném případě

„@Application“

Interakce se softwarem Albatros Umožňuje zobrazit „message box“ na displeji a zavřít software Albatros. Přípustnými hodnotami parametru **identifikátor** jsou:

„Quit“ (Kon Zavírá software Albatros.
ec)

„IsLocked“ Ověřuje, zda je ukončení softwaru Albatros zablokováno. Parametr **kontejner** bude obsahovat 1, když je rozhraní zablokováno, 0, když je možné ukončit provádění softwaru Albatros.

„MsgBox“ Čte odpověď z message box, který byl předtím otevřen instrukcí SEND

Parametr **kontejner** umožňuje v případě message box vědět, které tlačítko bylo stisknuto obsluhou:

- 1 tlačítko „OK“
- 2 tlačítko „Zrušit“
- 4 tlačítko „Opakovat“
- 6 tlačítko „Ano“
- 7 tlačítko „Ne“

V případě příkazu „Quit“ bude parametr **kontejner** obsahovat hodnotu:

- 1, když byl Albatros správně zavřen;
- 0 v opačném případě

„@Param“

Umožňuje poznat pořadové číslo záznamu souborů parametrizování Partec.xpar a Partool.xpar. Požadovaná informace je uložena do parametru **kontejner**. Přípustnými hodnotami parametru **identifikátor** jsou:

„partec“ vyžaduje pořadové číslo záznamu partec.xpar

„partool“ vyžaduje pořadové číslo záznamu partool.xpar

„@Ini“

Čte kombinaci klíč=hodnota ze souboru tpa.ini. Parametr **identifikátor** představuje název klíče, který má být přečten ze souboru tpa.ini v části [Tpa]. Za účelem přečtení specifické části je třeba přidat k názvu klíče název části v hranatých závorkách.(„[Část]Klíč“).

„@ShellExecute“

Žádá od operačního systému otevření souboru s použitím programu přiřazeného příponě souboru. Je možné také spustit spustitelný soubor. Parametr **identifikátor** představuje název souboru, který má být otevřen, nebo název programu, který má být spuštěn. Název souboru může být deklarován s kompletní cestou; v opačném případě bude hledán v aktuálním adresáři softwaru Albatros. Název souboru bude hledán také v těch, které jsou definované prostřednictvím „@FileName“. Parametr **kontejner** bude obsahovat hodnotu 0 když nedošlo k výskytu chyb při otevírání souboru; v opačném případě bude obsahovat kód chyby.

„@StartProg“

Slouží k provedení programu definovaného v parametru **identifikátor**. Není možné předávat argumenty programu, který má být spuštěn. Název programu musí obsahovat kompletní cestu, protože v opačném případě bude hledán v aktuálním adresáři softwaru Albatros. Název programu bude hledán také v těch, které jsou definovány prostřednictvím „@FileName“. Parametr **kontejner** bude obsahovat hodnotu 0, když bylo spuštění programu úspěšné; v opačném případě bude obsahovat kód příslušné chyby. Když byl program již spuštěn, kód chyby je 1056.

„@TermProg“

Ukončí program definovaný v parametru **identifikátor** a spuštěný prostřednictvím „@StartProg“. Název programu musí obsahovat kompletní cestu, protože v opačném případě bude hledán v aktuálním adresáři softwaru Albatros. Název programu bude hledán také v těch, které jsou definované prostřednictvím

„@FileName“. Parametr **kontejner** bude obsahovat hodnotu 0, když bylo spuštění programu úspěšné; v opačném případě bude obsahovat kód příslušné chyby. Když byl program již spuštěn, kód chyby je 1056.

„@ProgRunning“

Ověří, zda se program spuštěný prostřednictvím „@StartProg“ dosud provádí. Název programu je definován v parametru **identifikátor**. Název programu musí obsahovat kompletní cestu, protože v opačném případě bude hledán v aktuálním adresáři softwaru Albatros. Název programu bude hledán také mezi těmi, které jsou definovány prostřednictvím „@FileName“. Parametr **kontejner** bude obsahovat hodnotu 1, když je program ještě prováděn; v opačném případě bude obsahovat hodnotu 0.

„@DialogFile“

Slouží k otevření dialogového okna Soubor Otevřít nebo Soubor Uložit pro umožnění volby nového souboru. Pro otevření okna Soubor Otevřít nastavte parametr **identifikátor** = „Open“; pro otevření okna Soubor Uložit nastavte parametr **identifikátor** = „Uložit“. Název zvoleného souboru bude uložen do parametru **kontejner**.

„@AxisCorrectors“

Vyměňuje tabulku korektorů linearoty jednotky pohybu v ose za novou tabulku, načítanou ze souboru, která musí mít v každém případě stejný počet korektorů a stejný počet jednotek pohybu v osách pro křížové konektory.

Parametr **identifikátor** je název souboru, který má obvykle příponu .csv a nachází se v adresáři ...

\\Mod.n\Config (název souboru je 'normalizován' způsobem, jaký se používá například pro „@FileExist“).

Parametr **kontejner** je zadefinován jako proměnná integer a bude obsahovat hodnotu 1, když byly nové korektory odeslané a hodnotu 0 v opačném případě.

„@Language“

Přijímá přeložitelný text, odpovídající hlášení skupiny, knihovny nebo modulu, která/ý je přiřazená/ý příkazu MESSAGE nebo příkazu ERROR. Přípustnými hodnotami parametru **identifikátor** jsou:

- „DEFMSG:číslo“, kde číslo odpovídá posloupnosti číslic. Albatros zapisuje do parametru **kontejner** text hlášení modulu číslo „číslo“.
- „DEFMSG:název“, kde „název“ odpovídá DEFMSG, včetně skupiny a knihovny. Albatros zapisuje do parametru **kontejner** text uvedeného hlášení. Když chybí název skupiny nebo knihovny, použije se úkol, který odeslal příkaz RECEIVE.
- „DEFMSG:*“, Albatros zapisuje do **kontejneru** text hlášení skupiny nebo modulu, předtím uvedeného prostřednictvím příkazu SEND.

Příklad

```
; v GPL
RECEIVE "@Param", "partec", 0, prog
RECEIVE "@Param", "partool", 0, prog

; v GPL
; čte hodnotu klíče Radix v sekci [Albatros] souboru tpa.ini
RECEIVE "@INI", "[Albatros]Radix", 0, hodnota

; otevře okno Soubor Otevřít a uloží název souboru do proměnné
NavezSouboru
RECEIVE "@DialogFile", "Open", 0, NavezSouboru

; kompletní čtení souboru
Function ReadProperties
PARAM file AS STRING
LOCAL version AS INTEGER
LOCAL size AS DOUBLE

SEND "@FileName" "theFile" 0 file
WAITRECEIVE "@FileInfo", "version:theFile", 0, version
WAITRECEIVE "@FileInfo", "size:theFile", 0, size
```

SEND

Syntaxe

SEND [adresat,] identifikator, priznaky [, informace]

Argumenty

adresat	konstanta druhu řetězec
identifikator	konstanta druhu řetězec
priznaky	konstanta druhu integer

informace navez zarizeni nebo konstanta nebo promenna (číselná nebo v podobě řetězce)

Popis

Tato instrukce spolu s instrukcí RECEIVE používá pro výměnu informací mezi moduly výrobního zařízení a dozorným PC. SEND se používá pro odesílání informací a RECEIVE pro jejich příjem. Informace mohou být odesílány softwaru Albatros nebo externímu programu (Server OLE Automation). Ve druhém případě bude informace přijata softwarem Albatros, který zajistí její odeslání externímu programu.

Parametr **adresat** je tvořen řetězcem, který umožňuje určit komu je informace určena. Existují tři třídy adresátů:

- Adresáti, kteří začínají znakem „@“ (viz níže uvedený seznam). Adresátem je ve skutečnosti software Albatros nebo přesněji jedna z jeho funkcí.
- Adresáti, kteří nezačínají znakem „@“. Berou se jako Server OLE; při příjmu první informace, která je jim adresována, se software Albatros bude snažit spustit je a poté jim předat informaci přijatou od modulu.
- Neurčený adresát (parametr je skutečně volitelný). V tomto případě bude informace uchována softwarem Albatros v tabulce a zůstane k dispozici pro toho, kdo o ni požádá (další modul nebo externí program).

Parametr **identifikator** představuje název informace, a proto nemůže být vynechán. Nabývá různých významů v závislosti na adresátovi:

- Když je adresátem software Albatros, bude se jednat o příkaz související s funkcí, ke které je prováděn přístup
- Když je adresátem Server OLE, bude se jednat o vlastnost požadovaného objektu OLE.
- Když adresát není uveden, bude se jednat o návštěví, které identifikuje informaci v tabulce uskládněné v softwaru Albatros.

Parametr **příznaky** umožňuje uvést, jak má s požadovanou informací naložit Albatros. Níže jsou uvedeny přípustné hodnoty a příslušné odezvy:

hodnota	příkaz	popis
\$0001H	Broadcast	Běžné odeslání informace
\$0008H	CancelAfter	Informace bude po přečtení vymazána
\$0020H	ReadOnly	Informace může být vymazána pouze odesílatelem
\$1000H	UpdateFlags	Změní stav informace (přečtená / ke čtení) bez změny dat
\$8000H	Delete	Vymaže informaci

Parametr **informace** v závěru představuje samotnou odesílanou informaci. Tato může být vynechána a v takovém případě odeslání prázdné informace bude znamenat oznámení události (může být použita pro synchronizaci provádění kódu GPL na různých modulech). V úloze parametru informace jsou přípustná také zařízení (s výjimkou jednotek pohybu v ose), jednoduché proměnné GPL a řetězce.

Seznam **adresátů** ve správě softwaru Albatros a příslušné příkazy:

„@List“

Umožňuje správu příkazů Sim (Simulovany) a Setp (Nastavená hodnota - Setpoint)

Jsou přípustné níže uvedené příkazy (parametr **identifikator**):

- Sim: Oznámení změny stavu přepínače příznaku Sim (Simulovaný). Na základě stavu příznaku bude zobrazeno stisknuté nebo uvolněné tlačítko, které jej identifikuje na Panelu nástrojů (1=zvolený, 0=nezvolený)
- Setp: Oznámení změny stavu přepínače příznaku CmdSetp. Na základě stavu příznaku bude zobrazeno stisknuté nebo uvolněné tlačítko, které jej identifikuje na Panelu nástrojů (1=zvolený, 0=nezvolený)
- Esc: Oznámení změny stavu přepínače příznaku Vyrazený. Na základě stavu příznaku bude zobrazeno stisknuté nebo uvolněné tlačítko, které jej identifikuje (stejně jako v případě příznaku CmdSetp) na Panelu nástrojů (1=zvolený, 0=nezvolený)
- End: Ukončení provádění seznamu. Tento příkaz spustí tlačítka Start a Stop a zakáže položky menu Start a Stop
- Hold: Spustí tlačítko Stop a aktivuje položku menu Stop. Slouží ke zvednutí tlačítka Start a zrušení položky menu Start

„@Syn“

Komunikace mezi GPL a prohlížečem grafických zobrazení. Umožňuje otevření a zavření grafických zobrazení ovládané z GPL a odeslání informací do pole grafického zobrazení. Jsou přípustné níže uvedené příkazy (parametr **identifikator**):

- „Open:*nomefile*“ otevření grafického zobrazení *nomefile.xsyn*
- „Close:*nomefile*“ zavření grafického zobrazení *nomefile.xsyn*
- „Open“ otevření grafického zobrazení. Název souboru bude přečten z proměnné **informace**
- „Close“ zavření grafického zobrazení. Název souboru bude přečten z proměnné **informace**
- „*nazevpole*“ pole, ve kterém je zobrazována odeslaná informace

Interakce s oknem ohybu jednotek pohybu v osách podle určených pravidel je možné získat také prostřednictvím níže uvedeného parametru **adresat** „@Devices“.

„@File“

Zápis do souboru. Umožňuje vytvořit uživatelsky přizpůsobené soubory protokolu, do kterých mají být uloženy operace provedené strojem. Jedná se o textové soubory (ASCII). Parametr **identifikátor** je tvořen názvem souboru, do kterého bude proveden zápis.

Název souboru je tvořen konstantou druhu řetězec. Když v parametru identifikátor není uvedena kompletní cesta pro archivaci souboru, Albatros bude za cestu považovat tu, která je přednastavena v souboru tpa.ini, v části [tpa], pod položkou dirreport.

Když identifikátor začíná a končí znakem %, řetězec uvnitř identifikátoru bude vyhledán v tpa.ini, v části [tpa] a bude použit jako název souboru. Uvnitř názvu mohou být vloženy znaky, které budou nahrazeny během provádění instrukce:

- %n číslo modulu, který provede instrukci SEND
- %h aktuální čas (formát 00-23)
- %d aktuální den (formát 01-31)
- %m aktuální měsíc (formát 01-12)
- %y aktuální rok (formát se čtyřmi číslicemi)

Viz příklad.

Operace zápisu probíhají v režimu append (data jsou přidána na konec souboru). Do souboru mohou být odeslána číselná data (která jsou automaticky konvertována do formátu ASCII) nebo řetězce. Je možné zapisovat řetězce ve formátu datum a čas použitím znaků formátu %d pro datum a %t pro čas. Pro čas se používá formát „HH:mm:ss“ (tj.: hodiny, minuty a sekundy oddělené prostřednictvím „:“) a pro datum se používá formát, který závisí na národních nastaveních. Je možné použít další formát nastavením v tpa.ini, v části [Albatros], položka „LogNoLocale=1“ (přednastavená hodnota je LogNoLocale=0, tj. použití aktuálního formátu). Formát, který se má použít pro datum a čas, je možné nastavit i nezávisle na formátu nastaveném ve Windows a také v tomto případě je třeba nastavení provést v tpa.ini, v části [Albatros], prostřednictvím položek „LogDateFormat=“ a „LogTimeFormat=“ a přiřazením řetězce znaků podle níže uvedeného schématu. Když tyto položky nejsou přítomné nebo jsou prázdné, budou použity formáty nastavené ve Windows.

Formát času

h	čas ve formátu 12 hodin bez počátečních nul
hh	čas ve formátu 12 hodin s počátečními nulami
H	čas ve formátu 24 hodin bez počátečních nul
HH	čas ve formátu 24 hodin s počátečními nulami
m	minuty bez počátečních nul
mm	minuty s počátečními nulami
s	sekundy bez počátečních nul
ss	sekundy s počátečními nulami
t	jediny znak pro označení značky času, například A nebo P
tt	více znaků pro označení značky času, například AM nebo PM

Poznámka: Formáty „t“ a „tt“ používají značku času uvedenou na aktuálním ovládacím panelu. Nemusí to být pouze „AM“ a „PM“.

Příklad: Když je 11:29 odpoledne a řetězec je sestaven v podobě „hh':'mm':'ss tt“, bude napsáno „11:29:40 PM“.

Formát dne

d	den měsíce bez počátečních nul, znázorněný prostřednictvím čísla
dd	den měsíce s počátečními nulami, znázorněný prostřednictvím čísla
ddd	den týdne znázorněný znaky a zkrácený na tři písmena
dddd	den týdne znázorněný znaky, s úplným názvem
M	měsíc bez počátečních nul, znázorněný prostřednictvím čísla
MM	měsíc s počátečními nulami, znázorněný prostřednictvím čísla
MMM	měsíc znázorněný znaky a zkrácený na tři písmena
MMMM	měsíc znázorněný znaky, s úplným názvem
y	rok se dvěma číslicemi a bez počátečních nul pro roky s nižší hodnotou než 10
yy	rok se dvěma číslicemi, a s počátečními nulami pro roky s nižší hodnotou než 10
yyyy	rok znázorněný čtyřmi nebo pěti číslicemi v závislosti na druhu používaného kalendáře
yyyyy	rok znázorněný čtyřmi nebo pěti číslicemi v závislosti na druhu používaného kalendáře

Příklad: Když je středa, 31. srpna 1994, a řetězec je složen „ddd',' MMM dd yy“, bude zapsáno „Stř, Srpen 31 94“

V případě vynechání informace bude do souboru přidán „návrát na začátek řádku“.

„@FileName“

Slouží k uložení přiřazení mezi konstantou v podobě řetězce a názvem souboru sestavitelným pomocí proměnných druhu řetězec. Od okamžiku, ve kterém Albatros přijme komunikaci o přiřazení, nahradí všechny následující názvy souborů názvem přijatým prostřednictvím této instrukce. Parametr **identifikátor** je tvořen názvem souboru, do kterého bude proveden zápis. Název souboru je tvořen proměnnou druhu řetězec. Když v parametru identifikátor není uvedena kompletní cesta pro archivaci souboru, Albatros bude za cestu považovat tu, která je přednastavena v souboru tpa.ini, v části [tpa],

pod položkou dirreport. Hodnota parametru identifikátor je uložena do tpa.ini v části [GPLFileName] pod položkou Log, aby ji bylo možné použít i při následujících spuštěních softwaru Albatros. Za účelem odstranění přiřazení je třeba nastavit za parametr identifikátor prázdný řetězec. Takto definované přiřazení platí pro všechny moduly.

„@FileDelete“

Vymazání souboru. Parametr **identifikátor** představuje název souboru, který bude vymazán (kompletní cesta). Když v parametru identifikátor není uvedena kompletní cesta pro archivaci souboru, Albatros vezme v úvahu cestu definovanou v tpa.ini, v části [tpa], prostřednictvím položky dirreport. Název souboru může být definován podle pravidel popsanych u parametru **adresat @File**.

„@FileRead“

Přemístí kurzor souboru na začátek souboru. Parametr **identifikátor** představuje název souboru (kompletní cesta). Když v parametru identifikátor není uvedena kompletní cesta pro archivaci souboru, Albatros bude za cestu považovat tu, která je přednastavena v souboru tpa.ini, v části [tpa], pod položkou dirreport. Název souboru může být definován podle pravidel popsanych u parametru **adresat @FileRead**.

„@Axis“

Interakce s oknem manuálního pohybu jednotek pohybu v osách podle určených pravidel také prostřednictvím níže uvedeného parametru adresát „@Devices“. Když je již otevřeno jedno okno, které kontroluje pohyb jednotky pohybu v osách, příkaz působí na toto okno, bez ohledu na to, zda je otevřeno v grafickém zobrazení nebo v diagnostice. Když je okno zavřeno, příkaz se jej bude snažit otevřít v diagnostice nebo v jednom z již otevřených grafických zobrazení, které obsahuje tuto jednotku pohybu v ose.

„@Devices“

Požadavek na otevření nebo zavření okna Diagnostika týkajícího se modulu, který odesílá informaci. Provedení příkazů uvnitř okna pro pohybování jednotkou pohybu v ose v diagnostickém režimu. Parametr **identifikátor** může nabýt níže uvedených hodnot:

„Open“ otevření Diagnostiky

„Close“ zavření Diagnostiky

Parametr **identifikátor** může v případě, že má být prováděna interakce s oknem pro pohyb jednotky pohybu v ose, nabýt níže uvedených hodnot:

„MoveAX#nome_asse#Open“ otevření okna pro pohybování jednotkou pohybu v ose.

„MoveAX#nome_asse#Close“ zavření okna pro pohybování jednotkou pohybu v ose.

„MoveAX#nome_asse#Plus“ stisknutí tlačítka pohybování jednotkou pohybu v ose v kladném směru

„MoveAX#nome_asse#Minus“ stisknutí tlačítka pohybování jednotkou pohybu v ose v záporném směru

„MoveAX#nome_asse#Stop“ stisknutí tlačítka zastavení pohybu

„MoveAX#nome_asse#Jog“ nastavení režimu pohybu pro přesuny spravované v reálném čase (runtime) obsluhou

„MoveAX#nome_asse#Step“ nastavení režimu pohybu pro přesuny s přednastaveným krokem

„MoveAX#nome_asse#Absolute“ nastavení režimu pohybu s přesunem do určené polohy jednotky pohybu v ose

kde nome_asse představuje název jednotky pohybu v ose, která je zobrazena v okně. Například když má být otevřeno okno pro pohybování jednotkou pohybu v ose X, parametr **identifikátor** bude „@MoveAX#X#Open“. Název jednotky pohybu v ose může mít jeden z níže uvedených tvarů:

1. `Nazev_Skupiny.Nazev_Podskupiny.Nazev_Jednpohvose` nebo `Nazev_Skupiny.Nazev_Jednpohvose`: Bude dodána kompletní cesta jednotky pohybu v ose.

2. `Nazev_Jednpohvose`: Pro identifikaci správné jednotky pohybu v ose budou provedena níže uvedená ověření, a to v uvedeném pořadí:

- Když je úlohou, ze které pochází příkaz funkce podskupiny, jednotka pohybu v ose, bude hledána v té podskupině.
- Když je úlohou, ze které pochází příkaz funkce hlavní podskupiny, jednotka pohybu v ose, bude hledána v celé skupině. Když je přítomno více jednotek pohybu v osách s daným názvem, hledání bude neúspěšné.
- V případě, že se předchozí vyhledání nezdařilo, jednotky pohybu v ose bude hledána ve skupinách modulu. Když je přítomno více jednotek pohybu v osách s názvem `Nazev_Jednpohvose`, hledání nebude úspěšné.

Je možné zabránit uživateli, aby působil na tlačítka ovládání pohybování jednotkou pohybu v ose všech oken pohybování jednotkou pohybu v ose v diagnostickém režimu, a to níže uvedeným nastavením parametru **identifikátor** níže uvedeným způsobem:

- „MoveAX##UIENABLE“, když je parametr **informace** nastaven na 0, dojde k zákazu pohybování jednotkami pohybu v osách ze strany softwaru Albatros; při nastavení na 1 bude pohybování jednotkami pohybu v osách aktivováno softwarem Albatros.

Zákaz pohybování jednotkami pohybu v osách ze strany softwaru Albatros se doporučuje v případě pohybování jednotkami pohybu v osách z tlačítkového panelu stroje.

„@Vars“

Vyžaduje uložení obsahu globální proměnné GPL do archivu technologických parametrů nebo obráběcích nástrojů. Parametr **identifikátor** představuje název globální proměnné (stroje, skupiny nebo knihovny), pro kterou se vyžaduje aktualizace.

„@Application“

Interakce se softwarem Albatros. Umožňuje zavřít Albatros nebo zobrazit okno se správou („message box“) na displeji s cílem informovat uživatele nebo požádat o souhlas s následující aktivitou. Přípustnými hodnotami parametru **identifikátor** jsou:

„Quit“ (Konec) slouží k ukončení Albatros

„Lock“ (Zamk) zabraňuje zavření softwaru Albatros ze **Soubor->Konec** nebo kombinací kláves [ALT+F4] nout) nebo zavíracím tlačítkem.

„UnLock“ (Odeobnovuje možnost zavření softwaru Albatros mknout)

„MsgBox: přízn otevírá message box aky“

Chování message box je řízeno parametrem „příznaky“ řetězce **identifikátor**. Tou může být posloupnost následujících znaků (jsou přípustná bez rozdílů velká i malá písmena):

„O“	tlačítko „OK“
„C“	tlačítko „Zrušit“
„Y“	tlačítko „Ano“
„N“	tlačítko „Ne“
„R“	tlačítko „Opakovat“
„S“	ikona cedule Zastavení
„?“	„?“ informační ikona, tvořená malým písmenem 'i' v kruhu
„!“	ikona s vykřičníkem
„*“	informační ikona
„1“	první tlačítko je přednastavené
„2“	druhé tlačítko je přednastavené
„3“	třetí tlačítko je přednastavené
„4“	čtvrté tlačítko je přednastavené

Když to není uvedeno, přednastaveným tlačítkem je první.

Například „MsgBox: ?YN2“ definuje message box s informační ikonou, dvě tlačítka „Ano“ a „Ne“, přičemž druhé z nich je přednastaveno. Parametr **informace** může být tvořen řetězcem, který obsahuje text určený k zobrazení, nebo celým číslem, které je třeba interpretovat jako kód hlášení modulu spravovaného souborem TpaLangs.exe nebo návštěví hlášení skupiny definované instrukcí [DEFMSG](#).

Co se týče textu, když se v něm nachází znak návratu na začátek řádku, „\u000A“, text bude rozdělen na dvě části a bude zobrazena první část jako text okna se správou, zatímco druhá část bude zobrazena jako vysvětlení, nebo detail textu.

Jazyk, ve kterém jsou zobrazována tlačítka a jazyk operačního systému Windows.

„@Help“ (Nápověda)

Otevření souboru nápovědy. Umožňuje ovládat zobrazování souboru nápovědy určením tématu, které má být zobrazeno. Přípustnými hodnotami parametru **identifikátor** jsou:

„Open: *nazevsouboru*“ otevření souboru nápovědy

„Close: *nazevsouboru*“ zavření souboru nápovědy

Část „*nazevsouboru*“ řetězce určuje název souboru nápovědy, který má být otevřen.

Parametrem **informace** může být řetězec nebo číslo a může nabýt významu klíče nebo čísla kontextu (slouží k identifikaci tématu nebo strany nápovědy, které/á má být zobrazeno/a).

„@Report“ (Záznam)

Přidává signalizace do souboru záznamů Albatros (MONTH(n. měsíc).TER). Parametrem **identifikátor** je:

- „Add“ (Přidat)

Parametrem **informace** může být:

- Proměnná řetězec nebo konstanta řetězec: V tomto případě bude do paměti uložen text obsažený v řetězci
- Proměnná integer nebo číselná hodnota integer: V tomto případě bude do paměti uložen text definovaný instrukcí [DEFMSG](#).

„@Ini“

Zapisuje kombinaci klíč=hodnota do souboru tpa.ini. Parametr **identifikátor** představuje název klíče, který má být přidán do souboru tpa.ini, do části [Tpa]. Za účelem zápisu specifické části je třeba přidat k názvu klíče název části v hranatých závorkách („[Část]Klíč“).

Parametrem **informace** může být proměnná řetězec nebo číselná proměnná, konstanta řetězec nebo číselná konstanta.

„@ShellExecute“

Žádá od operačního systému otevření souboru s použitím programu přiřazeného příponě souboru. Je možné také spustit spustitelný soubor. Parametr **identifikátor** představuje název souboru, který má být otevřen, nebo název programu, který má být spuštěn. Název souboru může být deklarován s kompletní cestou; v

opačném případě bude hledán v aktuálním adresáři softwaru Albatros. Název souboru bude hledán také v těch, které jsou definované prostřednictvím „@FileName“.

„@StartProg“

Slouží k provedení programu definovaného v parametru **identifikátor**. Není možné předávat argumenty programu, který má být spuštěn. Název programu musí obsahovat kompletní cestu, protože v opačném případě bude hledán v aktuálním adresáři softwaru Albatros. Název programu bude hledán také v těch, které jsou definované prostřednictvím „@FileName“.

„@TermProg“

Ukončí program definovaný v parametru **identifikátor** a spuštěný prostřednictvím „@StartProg“ . Název programu musí obsahovat kompletní cestu, protože v opačném případě bude hledán v aktuálním adresáři softwaru Albatros. Název souboru bude hledán také v těch, které jsou definované prostřednictvím „@FileName“.

„@DialogFile“

Umožňuje nastavit některé parametry týkající se dialogového okna Soubor Otevřít nebo Soubor Uložit.

Přípustnými hodnotami parametru **identifikátor** jsou:

- „Extension“ když uživatel nezadá příponu, použije se ta, která je zadefinována v parametru **informace** (proměnná nebo konstanta typu řetězec)
- „Filter“ slouží k nastavení filtru na typy souborů, které mají být použity. Parametrem **informace** může být proměnná řetězec nebo konstanta řetězec a v tomto případě se jako filtr použije text, který se nachází v řetězci, proměnná typu integer nebo číselná hodnota integer, a v tomto případě se jako filtr použije text zadefinován v instrukci [DEFMSG](#).
- „Flags“ slouží k nastavení příznaku inicializace. Ohledně seznamu hodnot pro nastavení v poli **informace** (proměnná nebo konstanta typu integer) vycházejte z oficiální dokumentace společnosti Microsoft, týkající se člena Flags struktury OPENFILENAME.
- „InitalDir“ slouží k nastavení počáteční složky, zadefinované v poli **informace** (proměnná nebo konstanta typu řetězec)
- „Title“ slouží k nastavení názvu okna. Parametrem **informace** může být proměnná řetězec nebo konstanta řetězec a v tomto případě se jako filtr použije text, který se nachází v řetězci, proměnná typu integer nebo číselná hodnota integer, a v tomto případě se jako název použije text zadefinován v instrukci [DEFMSG](#).

„@Language“

Nastavuje číslo hlášení skupiny, modulu nebo knihovny, které bude použito v následujícím příkazu RECEIVE se stejným identifikátorem. Přípustnou hodnotou parametru **identifikátoru** je „DEFMSG:*“. Parametr **informace** může být proměnnou typu integer nebo konstantou integer a v tomto případě definuje číslo hlášení skupiny, která má být zobrazena. Může být typu řetězec nebo konstanta řetězec a v tomto případě definuje název DEFMSG.

Příklad

```
; Příklad instrukce send file s názvem vytvořeným během provádění.
; vycházejme z předpokladu, že datum provedení instrukce
; je 31-01-2000
```

```
; v GPL
SEND"@File", "%Log%", 0, "Zahájení provádění"
; přidá „návrát na začátek řádku“
; do souboru file tpa.ini, v části [TPA], bude přidáno
SEND"@File", "%Log%", 0
Log=c:\Albatros\report\%y\Rep%m%d.txt
```

```
; výsledný název souboru tedy bude:
c:\Albatros\report\2000\Rep0131.txt
```

```
; Příklad instrukce send Vars
; definuje se proměnná Var_SendVars
; as double v souboru globálních proměnných
; do technologického parametrizování se vloží Var_SendVars
; do pole NazevMatrice
; v GPL
SETVAL 100.0,Var_SendVars
; odešle hodnotu 100,0 parametru Technologického parametrizování
; přiřazený proměnné Var_SendVars
SEND"@Vars", "Var_SendVars", 0
```

```
; Příklad instrukce send INI
```

```
; do souboru tpa.ini se zapíše klíč Radix v části [Albatros]
; za účelem nastavení číselného základu desetinného zobrazování
SEND "@INI", "[Albatros]Radix", 0,1
```

```
; Příklad nastavení přiřazení mezi řetězcem konstantou GPL
; a názvem souboru.
```

```
; deklarování proměnné řetězec
navevsouboru as string
; složení názvu souboru
setstring "C:\albatros\report\LogFile.txt",navevsouboru
; přiřazení
SEND "@FileName", "LOG",0,navevsouboru
; všechny operace zápisu budou od této chvíle
; prováděny na souboru definovaném proměnnou navevsouboru
SEND "@File", "LOG",0,"Zapisuji do souboru LOG"
```

SENDIPC

Syntaxe

```
SENDIPC navevIPC, cekani [, navevpromenne1 [, navevpromenneN, ...]]
SENDIPC navevIPC, cekani , matrice[radek]
SENDIPC navevIPC, cekani , vektor
SENDIPC navevIPC, cekani , matrice
```

Argumenty

navevIPC	konstanta řetězec. Název IPC
cekani	přednastavená konstanta. Způsob čekání na přečtení příkazu Přípustnými hodnotami jsou: WAIT čeká na přečtení příkazu NOWAIT nečeká na přečtení příkazu
navevpromenne1[...navevpromenneN]	konstanta nebo proměnná. Názvy proměnných 1÷N
matrice[radek]	konstanta nebo proměnná integer. Číslo řádku matrice
vektor	Název vektoru
matrice	Název matrice

Popis

Odesílá příkaz IPC do sdílené paměti „**navevIPC**“.
Při prvním provedení instrukce SENDIPC bude přidělena sdílená paměť, jejíž velikost je vypočtena na základě rozměru dat, která jsou odesílána. Maximální velikost sdílené paměti je 64 Kb. Je možné definovat maximálně 48 sdílených pamětí identifikovaných 48 jedinečnými názvy.
Sdílené paměti je přiřazen semafor, který umožňuje synchronizovat provádění úloh, které do ní přistupují.
Úloha, která zapisuje data, aktivuje semafor na konci zápisu a úloha, která čte data, jej vypne na konci čtení.
Když byla v úloze parametru **cekani** uvedena hodnota WAIT, úloha, která zapsala data, vyčká, dokud data nebudou přečtena, než se bude pokračovat v provádění (vypnutý semafor).
Instrukce SENDIPC bez dat se stává pouhým synchronismem mezi úlohami. V tomto případě nebude provedeno přidělení sdílené paměti.

IPC intermodul

Dva vzdálené moduly mohou realizovat výměnu dat prostřednictvím IPC. Tyto IPC se nazývají IPC intermodul.
Pro zadefinování IPC intermodul je třeba zapsat **navevIPC** podle níže uvedeného formalismu:
číslo zdrojového modulu, „->“, číslo cílového modulu, „:“, a poté ostatní znaky názvu IPC.
Například „0->1:Základní Parametry“.

Viz také [WAITIPC](#) a [TESTIPC](#).

WAITIPC

Syntaxe

```
WAITIPC navevIPC [, navevpromenne1 [, navevpromenneN, ...]]
WAITIPC navevIPC, matrice[radek]
WAITIPC navevIPC, vektor
WAITIPC navevIPC, matrice
```

Argumenty

navevIPC	konstanta řetězec. Název IPC
-----------------	------------------------------

nazevpromenne1[...nazevpromenneN] konstanta nebo proměnná. Názvy proměnných 1÷N
matrice[radek] konstanta nebo proměnná integer. Číslo řádku matrice
vektor Název vektoru
matrice Název matrice

Popis

Přijímá příkaz IPC ze sdílené paměti „**nazevIPC**“.
 Při prvním provedení instrukce WAITIPC bude přidělena sdílená paměť, jejíž velikost je vypočtena na základě rozměru dat, která jsou odesílána. Maximální velikost sdílené paměti je 64 Kb. Je možné definovat maximálně 48 sdílených pamětí identifikovaných 48 jedinečnými názvy.
 Sdílené paměti je přiřazen semafor, který umožňuje synchronizovat provádění úloh, které do ní přistupují. Úloha, která čte data, čeká na zapnutí semaforu úlohou zapisující data; poté přečte data a vypne semafor. Instrukce WAITIPC bez dat se stává pouhým synchronizátorem mezi úlohami. V tomto případě nebude provedeno přidělení sdílené paměti.
 Viz také [SENDIPC](#) a [TESTIPC](#).

WAITRECEIVE

Syntaxe

WAITRECEIVE **[zdroj,] identifikator, priznaky [, kontejner]**

Argumenty

zdroj konstanta druhu řetězec
identifikator konstanta druhu řetězec
priznaky konstanta druhu integer
kontejner název zařízení nebo proměnné (číselné nebo v podobě řetězce)

Popis

Čeká, dokud požadovaná informace (určená parametrem **identifikator**) před dalším pokračováním v provádění programu GPL. Za účelem použití této instrukce vycházejte z dokumentace instrukce [RECEIVE](#).

10.3.10 Matematika

ABS

Syntaxe

ABS **operand, vysledek**

Argumenty

operand konstanta nebo proměnná nebo název zařízení
vysledek proměnná nebo název zařízení

Popis

Získá absolutní hodnotu **operandu** a uloží ji do **výsledku**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

SetVal -10,op ; přiřadí -10 proměnné op
Abs op,var ; hodnota umístěná do proměnné var bude 10

ADD

Syntaxe

ADD **operand1, operand2, vysledek**

Argumenty

operand1 konstanta nebo proměnná nebo název zařízení
operand2 konstanta nebo proměnná nebo název zařízení
vysledek proměnná nebo název zařízení

Popis

Slouží k provedení operace součtu mezi **operand1** a **operand2** a k uložení výsledku do parametru **výsledek**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

Setv 5,op1 ; přiřadí 5 proměnné op1
al

Add op1,3,var
; hodnota umístěná do proměnné var bude 8

AND

Syntaxe

AND operand1, operand2, vysledek

Argumenty

operand1	konstanta nebo proměnná nebo název zařízení
operand2	konstanta nebo proměnná nebo název zařízení
vysledek	proměnná nebo název zařízení

Popis

Slouží k provedení operace binárního součinu - AND (mezi dvěma bity je výsledkem 1 pouze v případě, že mají oba hodnotu 1) mezi **operand1** a **operand2** a umístí výsledek do parametru **vysledek**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

; hodnota umístěná do proměnné var bude 1
; (Binární notace: 5 = 0101, 3 = 0011, 1 = 0001)

And 5,3,var

ARCCOS

Syntaxe

ARCCOS operand, vysledek

Argumenty

operand	konstanta nebo proměnná nebo název zařízení
vysledek	proměnná nebo název zařízení

Popis

Slouží k provedení operace arkus kosinus na **operand** a umístí výslednou hodnotu do parametru **vysledek**, a to v stupních. Hodnota výsledku je dána v intervalu $0^\circ \div 180^\circ$. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

ARCSIN

Syntaxe

ARCSIN operand, vysledek

Argumenty

operand	konstanta nebo proměnná nebo název zařízení
vysledek	proměnná nebo název zařízení

Popis

Slouží k provedení operace arkus sinus na **operand** a umístí výslednou hodnotu do parametru **vysledek**, a to ve stupních. Hodnota výsledku je dána v intervalu $-90^\circ \div +90^\circ$. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

ARCTAN

Syntaxe

ARCTAN operand1, [, operand2], vysledek

Argumenty

operand1...[operand2]	konstanta nebo proměnná nebo název zařízení
vysledek	proměnná nebo název zařízení

Popis

Když je **operand2** vynechán, provede operaci arkus tangens na **operand1** a umístí výslednou hodnotu do parametru **vysledek**, a to ve stupních. Když je **operand2** přítomen, uvažovaným úhlem bude ten, jehož sinus určuje **operand1** a jehož kosinus určuje **operand2**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

COS

Syntaxe

COS **operand, vysledek**

Argumenty

operand konstanta nebo proměnná nebo název zařízení
vysledek proměnná nebo název zařízení

Popis

Slouží k provedení operace kosinus na parametru **operand** a umístí výslednou hodnotu do parametru **vysledek**.

Argument **operand** je vyjádřen ve stupních s případnou setinovou částí (např.: 30° 15" = 30,25.). Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 60,op ; přiřadí 60 proměnné op
Cos op,var
; hodnota umístěná do proměnné var bude 0.5
```

DIV

Syntaxe

DIV **operand1, operand2, vysledek**

Argumenty

operand1 konstanta nebo proměnná nebo název zařízení
operand2 konstanta nebo proměnná nebo název zařízení
vysledek proměnná nebo název zařízení

Popis

Slouží k provedení operace dělení parametrů **operand1** a **operand2** a k uložení výsledku do parametru **vysledek**.

Tato instrukce může vytvořit chybu systému, když se **operand2** rovná 0. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 10,op1 ; přiřadí 10 proměnné op1
SetVal 5,op2 ; přiřadí 5 proměnné op2
Div op1,op2,var
; hodnota umístěná do proměnné var bude 2
```

EXP

Syntaxe

EXP **operand, vysledek**

Argumenty

operand konstanta nebo proměnná nebo název zařízení
vysledek proměnná nebo název zařízení

Popis

Slouží k výpočtu exponentu parametru **operand** a uloží hodnotu do parametru **vysledek**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 2 302,op ; přiřadí 2 302 proměnné op
Exp op,var
; hodnota umístěná do proměnné var bude 10
```

EXPR

Syntaxe

EXPR **promenna = vyraz**

Argumenty

promenna název zařízení nebo proměnná

výraz

souhm operátorů

Popis

Tato instrukce umožňuje provést výpočet matematických výrazů. Operátory mohou být konstanty, názvy zařízení nebo proměnné. Syntax instrukce vyžaduje, aby mezi operátory byly mezery. V případě, že operandy výrazu nejsou stejného druhu, bude provedena automatická konverze a druh výsledku operace je stejný jako ten z obou, který je větší, na základě níže uvedeného pravidla:

- char < integer
- float < double
- char nebo integer < float nebo double.

Po vyřešení **výrazu** je výsledek konvertován na druh proměnné **vysledek**.

K přípustným operátorům seskupeným podle úrovně priority, patří:

()	závorky
-	operátor změny znaménka
ABS	absolutní hodnota operandu
ROUND	zaokrouhlení na jednotky
TRUNC	zkrácení na celou hodnotu
LOG	přirozený logaritmus
LOGDEC	logaritmus se základem 10
EXP	exponenciální funkce
SQR	druhá odmocnina
SIN	operace sinu. Operand je vyjádřen ve stupních s případnou setinovou částí (např.: 30° 15" = 30,25.)
COS	operace kosinu. Operand je vyjádřen ve stupních s případnou setinovou částí (např.: 30° 15" = 30,25.)
TAN	operace tangentu. Argument je vyjádřen ve stupních.
ARCSIN	operace arkus sinu. Výsledek je vyjádřen ve stupních a jeho hodnota se pohybuje v intervalu -90° ÷ +90°
ARCCOS	operace arkus kosinu. Výsledek je vyjádřen v stupních a jeho hodnota se pohybuje v intervalu 0° ÷ 180°
ARCTAN	provede operace arkustangens. Viz ARCTAN
^	operace umocnění

*	násobení
/	dělení
%	zbytek dělení (modul)
+	součet
-	odečítání

Tato instrukce umožňuje zjednodušit zápis kódu GPL v případech, kdy je třeba provést matematické výpočty, prostřednictvím výměny jednotlivých instrukcí GPL odpovídajících operátorům uvedeným v tabulce. Uvedené instrukce zůstávají k dispozici kvůli kompatibilitě.

Příklad

; výpočet vzdálenosti mezi dvěma body

```
EXPR vzdalenost = SQR ( ( Xb - Xa ) ^ 2 + ( Yb - Ya ) ^ 2 )
```

; konverze dat v instrukci EXPR

```
local op1 as double
local op2 as integer
local op3 as float
local vysledek as integer
setval 900,op1
setval 100,op2
setval 500.0,op3
EXPR vysledek = ( op1 + op2 ) / op3
; jako první je vyřešen výraz v double
; poté je provedena konverze na integer
; jako druh data vysledek
```


LOG

Syntaxe

LOG **operand, vysledek**

Argumenty

operand konstanta nebo proměnná nebo název zařízení
vysledek proměnná nebo název zařízení

Popis

Slouží k výpočtu přirozeného logaritmu parametru **operand** a uloží hodnotu do parametru **vysledek**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 10,op ; přiřadí 10 proměnné op
Log op,var

; hodnota umístěná do proměnné var bude 2.302585093
```

LOGDEC

Syntaxe

LOGDEC **operand, vysledek**

Argumenty

operand konstanta nebo proměnná nebo název zařízení
vysledek proměnná nebo název zařízení

Popis

Slouží k výpočtu logaritmu se základem 10 **operandu** a uloží hodnotu do parametru **vysledek**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 10,op ; přiřadí 10 proměnné op
Logdec op,var

; hodnota umístěná do proměnné var bude 1
```

MOD

Syntaxe

MOD **operand1, operand2, vysledek**

Argumenty

operand1 konstanta nebo proměnná integer nebo název zařízení
operand2 konstanta nebo proměnná integer nebo název zařízení
vysledek proměnná integer nebo název zařízení

Popis

Slouží k provedení operace modulu mezi parametry **operand1** a **operand2** a k uložení výsledku do parametru **vysledek**. Modul je zbytek vyplývající z vydělení prvního a druhého operandu. Tato instrukce může způsobit vznik chyby systému, když se **operand2** rovná 0. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 20,op1 ; přiřadí 20 proměnné op1
SetVal 3,op2 ; přiřadí 3 proměnné op2
Mod op1,op2,var

; hodnota umístěná do proměnné var bude 2
```

MUL

Syntaxe

MUL **operand1, operand2, vysledek**

Argumenty

operand1 konstanta nebo proměnná nebo název zařízení
operand2 konstanta nebo proměnná nebo název zařízení
vysledek proměnná nebo název zařízení

Popis

Slouží k provedení operace násobení mezi parametry **operand1** a **operand2** a k uložení výsledku do parametru **vysledek**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
Setv 5,op1; přiřadí 5 proměnné op1
a1
Setv 2,op2; přiřadí 2 proměnné op2
a1
Mul op1,op2,var
; hodnota umístěná do proměnné var bude 10
```

NOT**Syntaxe**

NOT **operand**

Argumenty

operand proměnná nebo název zařízení

Popis

Slouží k provedení operace binární negace NOT (*dojde ke změně hodnot jednotlivých bitů*) hodnoty vyjádřené argumentem **operand**. Výsledek je uložen do parametru **operand**.

Příklad

```
Setval 5,var ; přiřadí hodnotu 5 proměnné „var“
Not var
; výsledkem bude = -6
; Binární notace: 5 = 0000 0101,
; Binární notace: 10 = 0000 1010
; Hexadecimální notace 5 = 0000 0000 0000 0005
; Hexadecimální notace 10 = 0000 0000 0000 000A
; po provedení NOT na hodnotě 5 bude výsledkem 0xFFFF FFFF FFFF FFFA = -6
```

OR**Syntaxe**

OR **operand1, operand2, vysledek**

Argumenty

operand1 konstanta nebo proměnná nebo název zařízení
operand2 konstanta nebo proměnná nebo název zařízení
vysledek proměnná nebo název zařízení

Popis

Slouží k provedení operace binárního součtu OR (mezi dvěma bity; výsledkem je 1, když aspoň jeden z nich má hodnotu 1) mezi operandy **operand1** a **operand2** a výsledek je uložen do parametru **vysledek**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
; hodnota umístěná do proměnné var bude 7
; (Binární notace: 5 = 0101, 3 = 0011, 7 = 0111)
or 5,3,var
```

RANDOM**Syntaxe**

RANDOM **min, max, vysledek**

Argumenty

min konstanta nebo proměnná

max konstanta nebo proměnná
vysledek proměnná nebo název zařízení

Popis

Vrací v parametru **vysledek** pseudonáhodné číslo z rozsahu od **min** po **max** (včetně uvedených mezních hodnot). Opakovaným provedením této instrukce lze získat posloupnost pseudonáhodných čísel. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVa 2,op1 ; přiřadí 2 proměnné op1
|
SetVa 100,op2; přiřadí 100 proměnné op2
|
Rando op1,op2; hodnota uložená do proměnné var
m ,var ; bude se jednat o náhodné číslo
; v rozsahu od 2 do 100
```

RESETBIT

Syntaxe

RESETBIT **maska, cislobitu**

Argumenty

maska konstanta nebo proměnná integer nebo nazevpocitadla nebo nazevportu. Představuje hodnotu, která má být změněna (max. 32 bitů)
cislobitu konstanta nebo proměnná integer nebo nazevpocitadla. Číslo bitu, který má být změněn

Popis

Slouží k nastavení jednotlivého bitu, který je vymezen prostřednictvím **cislobitu**, v procházeném parametru **maska** bitů na hodnotu 0. Argument **maska** musí mít možnost odpovídat hodnotě celého čísla s maximem 32 bitů. Číslo bitu, **cislobitu**, se pohybuje od 1 do 32.

Příklad

Stav portu před provedením kódu



Stav portu po provedení kódu



```
-----
; Příklad pro zrušení řádku portu příznaku:
-----
```

```
SetVal 2,cislobitu
Resetbit FlagPort,cislobitu
; slouží ke zrušení řádku 2 portu příznaku
```

ROUND

Syntaxe

ROUND **operand, vysledek**

Argumenty

operand konstanta nebo proměnná nebo název zařízení
vysledek proměnná nebo název zařízení

Popis

Slouží k provedení operace zaokrouhlení na jednotky parametru **operand** a umístí výslednou hodnotu do parametru **vysledek**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 5.7,op ; přiřadí 5,7 proměnné op
Round op,var
```

; hodnota umístěná do proměnné var bude 6

```
SetVal 5.2,op ; přiřadí 5,2 proměnné op
Round op,var
```

; hodnota umístěná do proměnné var bude 5

SETBIT**Syntaxe**

```
SETBIT maska, cislobitu
```

Argumenty

maska konstanta nebo proměnná integer nebo nazevpocitadla nebo nazevportu. Představuje hodnotu, která má být změněna (max. 32 bitů)

cislobitu konstanta nebo proměnná integer nebo nazevpocitadla. Číslo bitu, který má být změněn (1÷32)

Popis

Slouží k nastavení jednotlivého bitu, který je vymezen prostřednictvím **cislobitu**, v procházeném parametru **maska** bitů na hodnotu 1. Argument **maska** musí mít možnost odpovídat hodnotě celého čísla s maximem 32 bitů. Číslo bitu, **cislobitu**, se pohybuje od 1 do 32.

Příklad

Stav portu před provedením kódu



Stav portu po provedení kódu



```
-----
; Příklad pro aktivaci řádku portu příznaku:
;
;
;-----
```

```
SetVal 2,cislobitu
Setbit FlagPort,cislobitu
```

; slouží k aktivaci řádku 2 portu příznaku

SHIFTL**Syntaxe**

```
SHIFTL operand1 [, operand2]
```

Argumenty

operand1 konstanta (integer nebo char) nebo název zařízení

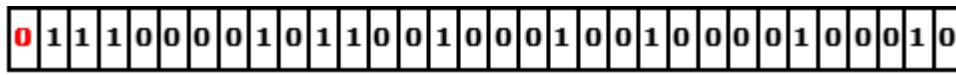
operand2 konstanta (integer nebo char) nebo název zařízení

Popis

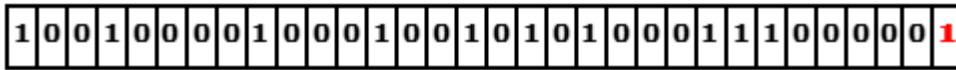
Není-li uveden **operand2**, provede operaci posuvu bitů, které tvoří **operand1**, doleva. Když je uveden také druhý operand, provede operaci otočení mezi bity parametru **operand2**, a bity parametru **operand1**. Na konci operace bude **operand2** obsahovat přenesení, tj. Vysoký bit **operand1**.

Příklad

Rotace operandů typu integer (shift doleva s přenesením)
Před rotací

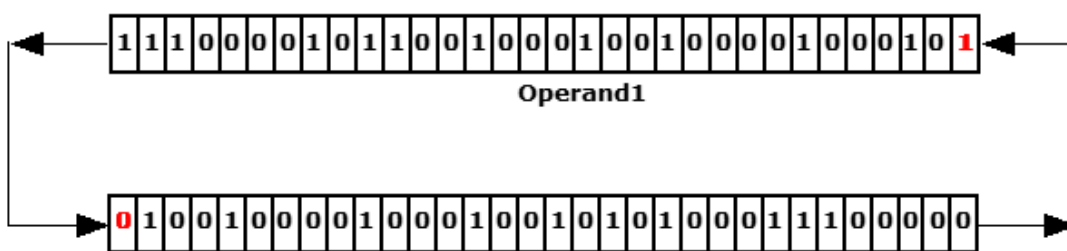


Operand1



Operand2

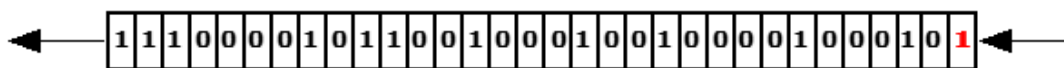
Po rotaci



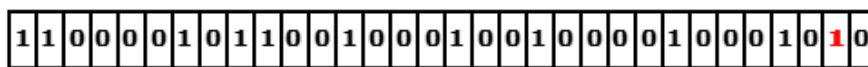
Operand1

Operand2

Posun (shift doleva bez přenesení)



Operand1



Operand1 po shift
doleva

SHIFTR

Syntaxe

SHIFTR

operand1 [, operand2]

Argumenty

operand1

proměnná (integer nebo char) nebo název zařízení

operand2

proměnná (integer nebo char) nebo název zařízení

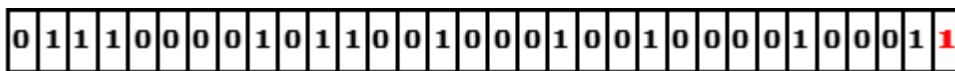
Popis

Není-li uveden **operand2**, provede operaci posuvu bitů, které tvoří **operand1**, doprava. Pokud je **operand1** prohlášen za typ char, vysoký bit na vstupu je vždy 0. Pokud je **operand1** prohlášen za typ integer, bit 32 na vstupu je značkový bit.

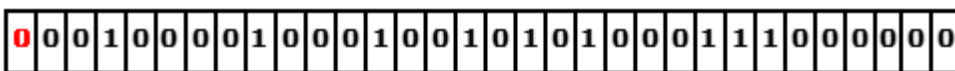
Když je uveden také druhý operand, provede operaci otočení mezi parametrem **operand2**, který je považován za hodnotu 0, nebo odlišným od 0, a bity parametru **operand1**. Na konci operace **operand2** bude obsahovat přenesení operace a bit s nejvyšší vahou parametru **operand1** získá hodnotu 0 nebo 1 v závislosti na počáteční hodnotě parametru **operand2** (0 nebo 1).

Příklad

Rotace operandů typu integer (shift doprava s přenesením)
Před rotací

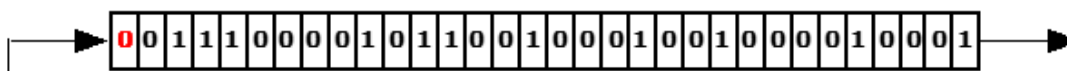


Operand1

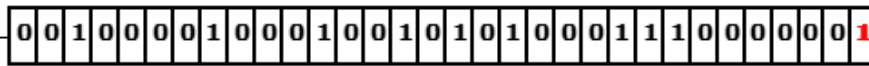


Operand2

Po rotaci

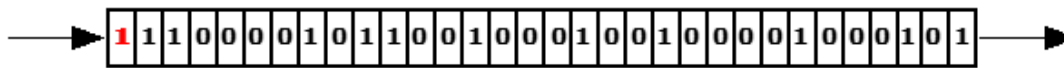


Operand1

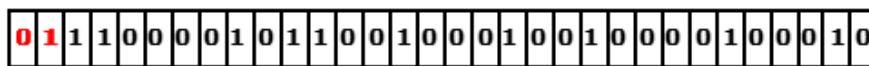


Operand2

Posun doprava charu (shift doprava bez přenesení)

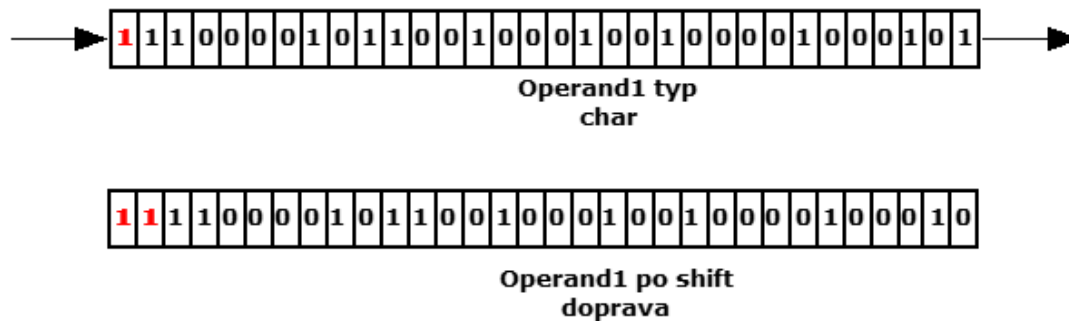


Operand1 typ char



Operand1 po shift doprava

Posun doprava integeru (shift doprava bez přenesení)

**SIN****Syntaxe****SIN****operand, vysledek****Argumenty****operand
vysledek**konstanta nebo proměnná nebo název zařízení
proměnná nebo název zařízení**Popis**

Slouží k provedení operace sinus na parametru **operand** a umístí výslednou hodnotu do parametru **vysledek**. Argument **operand** je vyjádřen ve stupních s případnou setinovou částí (např.: 30° 15" = 30,25). Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 30,op ; přiřadí 30 proměnné op
Sin op,var
```

; hodnota umístěná do proměnné var bude 0.5

SQR**Syntaxe****SQR****operand, vysledek****Argumenty****operand
vysledek**konstanta nebo proměnná nebo název zařízení
proměnná nebo název zařízení**Popis**

Slouží k provedení operace druhé odmocniny na parametru **operand** a umístění výsledné hodnoty do parametru **vysledek**. U parametru **operand** jsou přípustné pouze kladné hodnoty. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 81,op ; přiřadí 81 proměnné op
Sqr op,var
```

; hodnota umístěná do proměnné var bude 9

SUB**Syntaxe****SUB****operand1, operand2, vysledek****Argumenty**

operand1	konstanta nebo proměnná nebo název zařízení
operand2	konstanta nebo proměnná nebo název zařízení
vysledek	proměnná nebo název zařízení

Popis

Slouží k provedení operace odečítání mezi parametry **operand1** a **operand2** a k uložení výsledku do parametru **vysledek**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 10,op1 ; přiřadí 10 proměnné op1
SetVal 4,op2 ; přiřadí 4 proměnné op2
Sub op1,op2,var
```

; hodnota umístěná do proměnné var bude 6

TAN**Syntaxe**

TAN **operand, vysledek**

Argumenty

operand	konstanta nebo proměnná nebo název zařízení
vysledek	proměnná nebo název zařízení

Popis

Slouží k provedení operace tangens na parametru **operand** a umístí výslednou hodnotu do parametru **vysledek**. Argument **operand** je vyjádřen ve stupních. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 45,op ; přiřadí 45 proměnné op
Tan op,var
```

; hodnota umístěná do proměnné var bude 1

TRUNC**Syntaxe**

TRUNC **operand, vysledek**

Argumenty

operand	konstanta nebo proměnná nebo název zařízení
vysledek	proměnná nebo název zařízení

Popis

Odřízne na celé číslo hodnotu vyjádřenou parametrem **operand** a uloží ji do parametru **vysledek**. (Desetinná část bude ztracena). Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
SetVal 5.7,op ; přiřadí 5,7 proměnné op
Trunc op,var
```

; hodnota umístěná do proměnné var bude 5

XOR**Syntaxe**

XOR **operand1, operand2, vysledek**

Argumenty

operand1	konstanta nebo proměnná nebo název zařízení
operand2	konstanta nebo proměnná nebo název zařízení
vysledek	proměnná nebo název zařízení

Popis

Slouží k provedení operace exkluzivního binárního součtu XOR (mezi dvěma bity; výsledkem je 1 když pouze jeden ze dvou má hodnotu 1) mezi operandy **operand1** a **operand2** a výsledek je uložen do parametru **výsledek**. Ohledně konverze dat na základě deklarovaného druhu vycházejte z kapitoly [Konverze dat](#).

Příklad

```
xor 5,3,var
; hodnota umístěná do proměnné var bude 6
; (Binární notace: 5 = 0101, 3 = 0011, 6 = 0110)
```

10.3.11 Multitasking

ENDMAIL

Syntaxe

ENDMAIL **posta**

Argumenty

posta konstanta nebo proměnná druhu integer. Číslo poštovní schránky (1÷256)

Popis

Signalizuje ukončení provádění příkazu přiřazeného hlášení odebranému z poštovní schránky **posta**. Úloha, která odeslala hlášení (s použitím instrukce [SENDMAIL](#)) a zůstala čekat na provedení příkazu (argument čekání WAITACK), může takto pokračovat ve svém provádění. Tato instrukce **má vliv pouze v případě, že je provedena úlohou**, která předtím přijala hlášení (prostřednictvím instrukce WAITMAIL nebo TESTMAIL).

Viz také instrukce [SENDMAIL](#), [WAITMAIL](#) a [TESTMAIL](#)

Příklad

[Server pohybu jednotek pohybu v osách](#)

ENDREALTIMETASK

Syntaxe

ENDREALTIMETASK **nazevfunkce**

Argumenty

nazevfunkce název funkce

Popis

Slouží k ukončení [úlohy v reálném čase \(real-time\)](#). Viz také [STARTREALTIMETASK](#).

ENDTASK

Syntaxe

ENDTASK **[nazevulohy]**

Argumenty

nazevulohy název úlohy

Popis

Slouží k ukončení úloh a všech úloh, které tato úloha aktivovala (úlohy, které jsou jejími potomky). Tato instrukce dále pozastaví pohyb jednotek pohybu v osách, zruší nevyřízené RECEIVE a zavře případné připojení k sériovým portům. Když je proměnná **nazevulohy** vynechána, tato instrukce ukončí provádění aktuální úlohy.

GETPRIORITYLEVEL

Syntaxe

GETPRIORITYLEVEL **uroven [,nazevfunkce]**

Argumenty

uroven proměnná. Úroveň priority provádění
nazevfunkce název funkce

Popis

Vrací do proměnné **uroven** hodnotu priority úlohy definované prostřednictvím parametru **nazevfunkce**. Uvedená hodnota je celým číslem v rozsahu od 1 do 255, kde 1 představuje nejvyšší a 255 nejnižší úroveň. Když parametr **nazevfunkce** není definován, vrátí se hodnota priority aktuální úlohy, tj. funkce, ve které je prováděna instrukce GETPRIORITYLEVEL. Viz také [SETPRIORITYLEVEL](#).

GETREALTIME

Syntaxe

GETREALTIME **nazevpromenne**

Argumenty

nazevpromenne proměnná integer

Popis

Vrací do proměnné **nazevpromenne** dobu, která uplynula od začátku posledního provádění v reálném čase (real-time) v rámci správy jednotek pohybu v osách. Vrácená doba je vyjádřena v mikrosekundách. Viz také [GETREALTIMECOUNT](#).

GETREALTIMECOUNT

Syntaxe

GETREALTIMECOUNT **nazevpromenne**

Argumenty

nazevpromenne proměnná integer

Popis

Vrací do proměnné **nazevpromenne** počet provedení v reálném čase (real-time) správy jednotek pohybu v osách od poslední inicializace číslicového řízení. Viz také [GETREALTIME](#).

HOLDTASK

Syntaxe

HOLDTASK **[nazevulohy]**

Argumenty

nazevulohy název úlohy

Popis

Slouží k pozastavení provádění úlohy definované v parametru **nazevulohy**. Tato instrukce nepozastaví pohyb jednotek pohybu v osách, které je třeba zastavit instrukcemi STOP. Když je **nazevulohy** vynechán, tato instrukce pozastaví provádění aktuální úlohy.

RESUMETASK

Syntaxe

RESUMETASK **[nazevulohy]**

Argumenty

nazevulohy název úlohy

Popis

Slouží k obnovení provádění úlohy zdefinované v parametru **nazevulohy**. Když je **nazevulohy** vynechán, tato instrukce obnoví provádění aktuální úlohy. Když byla úloha pozastavena instrukcí STOPTASK, budou obnoveny také případné pohyby jednotek pohybu v osách.

SENDMAIL

Syntaxe

SENDMAIL **postovnischranka, cekani [, nazevpromenne1 [,..nazevpromenneN]]**
SENDMAIL **postovnischranka, cekani, matrice[radek]**

Argumenty

postovnischranka konstanta nebo proměnná integer. Číslo poštovní schránky (1÷256)
cekani přednastavená konstanta. Režim čekání na čtení nebo provedení příkazu.

K hodnotám, které mohou být přiřazeny konstantě cekani, patří:

- **WAIT** čeká na přečtení příkazu
- **NOWAIT** nečeká na přečtení příkazu
- **WAITACK** čeká na provedení příkazu

nazevpromenne1[...nazevpromenneN] konstanta nebo proměnná integer. Názvy proměnných 1÷20
matrice[radek] konstanta nebo proměnná integer. Číslo řádku matrice

Popis

Odesílá hlášení (nebo příkaz) do pole **postovnischranka**. Hlášení mohou být použita pro synchronizaci a výměnu informací mezi dvěma nebo více úlohami.

Když pole **postovnischranka** neexistuje, tedy nebyla ještě provedena instrukce [WAITMAIL](#) nebo [TESTMAIL](#), instrukce bude ignorována.

Když přijímací úloha nečeká na hlášení (instrukce [WAITMAIL](#)) nebo je obsazena, data (**nazevpromenne** (1÷20) nebo do řádku matrice uvedeného v **matrice[radek]**) předaná instrukcí budou uložena do řady. V tomto případě:

1. Když je argumentem cekani **NOWAIT**, provádění bude pokračovat následující instrukcí;
2. když je argumentem cekani **WAIT**, provádění počká na přečtení hlášení přijímací úlohy;
3. když je argumentem cekani **WAITACK**, provádění počká na přečtení hlášení a na potvrzení provedení příkazu od přijímací úlohy (prostřednictvím instrukce [ENDMAIL](#) nebo nové [WAITMAIL](#)).

Je velmi důležité, aby se počet předaných proměnných a jejich druhy shodovaly s těmi, které byly použity pro vytvoření poštovní schránky prostřednictvím instrukce [WAITMAIL](#). Řízení neumožňuje použití různých druhů a provede automatické konverze druhu (cast) jako v obvyklém případě.

Z instrukce [SENDMAIL](#) bez volitelných parametrů (data) se stává jednoduchý mechanismus synchronizace mezi úlohami.

Příklad

[Server pohybu jednotek pohybu v osách](#)

SETPRIORITYLEVEL

Syntaxe

SETPRIORITYLEVEL uroveň [,nazevfunkce]

Argumenty

uroveň konstanta nebo proměnná. Úroveň priority provádění
nazevfunkce název funkce

Popis

Přiřadte hodnotu priority obsaženou v proměnné level k úkolu definovanému ve **nazevfunkce**. Uvedená hodnota je celým číslem v rozsahu od 0 do 255, kde 0 představuje nejvyšší a 255 nejnižší úroveň. Když není definován název funkce v proměnné **nazevfunkce**, bude změněna hodnota priority aktuální úlohy, tedy úroveň provádění funkce, ve které je instrukce prováděna. Viz také [GETPRIORITYLEVEL](#).

STARTREALTIMETASK

Syntaxe

STARTREALTIMETASK nazevfunkce

Argumenty

nazevfunkce název funkce

Popis

Slouží k aktivaci provádění [úlohy v reálném čase \(real-time\)](#). Tato úloha je prováděna se stejnou frekvencí, jaká je aplikována na řízení jednotek pohybu v osách v reálném čase. Na rozdíl od běžných úloh GPL je každá real-time prováděna kompletně, od první instrukce function až po první instrukci [FRET](#). Viz také [ENDREALTIMETASK](#).

Poznámka

Místní proměnné deklarované v úloze prováděné v reálném čase jsou inicializovány teprve při spuštění úlohy a poté si udržují hodnotu posledního provedení.

Čeká na dokončení provádění úlohy **nazevulohy**.

Příklad

[Sekvenční / Paralelní Provedení](#)

10.3.12 Správa Toků

CALL

Syntaxe

CALL **nazevpodprogramu**

Argumenty

nazevpodprogramu název podprogramu, návěstí

Popis

Provede podprogram definovaný návěstím **nazevpodprogramu**. Každý podprogram musí končit v místě výstupu instrukcí [RET](#), aby se mohl vrátit na instrukci, která následuje po CALL.

Poznámka

Tato instrukce představuje spolu s RET typický zdroj chyb programování. Doporučuje se proto věnovat pozornost jejímu použití, zejména se doporučuje umísťovat podprocedury na konec funkce (po instrukci FRET), aby se zabránilo náhodnému provedení kódu podprocedury, jakoby se jednalo o hlavní kód. Výsledkem této situace je v nejlepším případě chyba systému a v ostatních případech je výsledkem poruchové chování stroje, jehož příčinu lze obtížně určit.

DELONFLAG

Syntaxe

DELONFLAG **nazevpriznaku**

Argumenty

nazevpriznaku název zařízení příznak

Popis

Slouží ke správě přerušení softwaru na stavu bitu příznaku nebo přepínače příznaku, která byla předtím aktivována prostřednictvím instrukce [ONFLAG](#).

DELONINPUT

Syntaxe

DELONINPUT **nazevvstupu**

Argumenty

nazevvstupu název vstupu

Popis

Slouží k zakázání správy softwarového přerušení podle stavu vstupu, která byla předtím aktivována instrukcí [ONINPUT](#).

FCALL

Syntaxe

[FCALL] **nazevfunkce** **nazevfunkce [, parametry]**
[parametry]

Argumenty

nazevfunkce název funkce, která má být vyvolána
parametry případné parametry jednotky pohybu v osách předané funkci

Popis

Provede vyvolání funkce, tj. bude provedena funkce **nazevfunkce**. Funkci mohou být předány případné **parametry**. Tyto musí odpovídat z hlediska počtu a druhu těm, které jsou deklarovány ve vyvolané funkci. Provedení vyvolávající funkce (té, ve které je provedena instrukce FCALL) bude obnoveno po provedení vyvolané funkce (té, která je uvedena v parametru **nazevfunkce**).

Všimněte si rozdílu ve srovnání s instrukcí [STARTTASK](#), která spustí další funkci paralelně s vyvolávající (používá se pro zajištění současného provádění více funkcí současně).

Příklad

[Sekvenční / Paralelní Provedení](#)

FOR/NEXT

Syntaxe

```
FOR          index, zacatek, konec [, krok]
  instrukce
  instrukce
  ...
NEXT
```

Argumenty

index	proměnná nebo nazevpocitadla
zacatek	konstanta nebo proměnná nebo nazevpocitadla. Počáteční hodnota
konec	konstanta nebo proměnná nebo nazevpocitadla. Závěrečná hodnota
krok	konstanta nebo proměnná nebo nazevpocitadla. Krok zvyšování nebo snižování

Popis

Zopakuje cyklicky provádění instrukcí uzavřených mezi instrukcí FOR a instrukcí NEXT. Během prvního cyklu je proměnná **index** inicializována na hodnotu proměnné **zacatek**. Při druhém cyklu bude mít proměnná **index** hodnotu rovnající se (**zacatek+krok**), a tak dále, dokud se proměnná **index** nestane větší, (nebo menší v případě záporné hodnoty proměnné **krok**), než proměnná **konec**. Když je proměnná **krok** vynechána, bude použita přednastavená hodnota rovnající se +1. Instrukce uzavřené mezi FOR a NEXT mohou měnit počet opakování změnou proměnné **index**. Po skončení opakování bude provedena instrukce, která následuje po instrukci NEXT.

Příklad

```
Function Loop
  Local i as integer
  local vektor[10] as integer

  For i,1,10
    setval i, vektor[i] ; naplní prvky vektoru
                       ; čísla 1,2, .... 10
  NEXT
Fret
```

```
Function loop2
  Local j as integer
  local vektor[10] as integer

  For j,1,10,2
    setval 27, vektor[j] ; vloží hodnotu 27 do následujících
                       ; prvků vektoru: 1,3,5,7,9
  NEXT
Fret
```

FRET

Syntaxe

```
FRET
```

Argumenty

žádný argument

Popis

Návrat z funkce. Způsobuje ukončení provádění funkce a uvolnění paměti přidělené pro místní proměnné. Když byla funkce spuštěna prostřednictvím instrukce FCALL, provádění vyvolávající funkce bude obnoveno od následující instrukce.

Když byly předtím provedeny instrukce WAITTASK s aktuální funkcí (s tou, ve které je provedena FRET) v úloze argumentu, bude provedeno odjištění čekajících úloh.

GOTO

Syntaxe

GOTO **navesti**

Argumenty

navesti návěstí

Popis

Provádí nepodmíněný skok na návěstí uvedené parametrem **navesti**.

Návěstí je definováno klíčovým slovem, po kterém následuje znak „:“.

Návěstí se musí nacházet uvnitř těla funkce, ve které je prováděna instrukce GOTO.

Poznámka

Tělo funkce se nachází v rozsahu od instrukce FUNCTION, která deklaruje název funkce po obdobnou instrukci, která definuje následující funkci (nebo od konce souboru). Z tohoto lze vyvodit možnost provedení skoků z hlavního těla funkce a případných podprocedur (viz instrukce [CALL](#) a [RET](#)). Výrazně se doporučuje tento styl programování, protože tato instrukce je zdrojem mnoha chyb, které lze jen velmi těžko identifikovat.

Příklad

```
; Function, která provádí blikání příznaku
; (např. kontrolka alarmu na grafickém zobrazení)
```

Function Loop

```
loop:
  setflag    alarm    ; zapne příznak
  delay     1
  resetflag  alarm    ; vypne příznak
  delay     1
  goto      loop
  Fret
```

IF/IFVALUE/IF-THEN-ELSE

Syntaxe

IF **nazevpromenne, operator porovnani, hodnota, GOTO navesti**

IF **nazevpromenne, operator porovnani, hodnota, CALL**

nazevpodprogramu

IF **nazevpromenne, operator porovnani, hodnota, nazevfunkce**

IF **nazevpromenne, operator porovnani, hodnota THEN**

instrukce

instrukce

...

ENDIF

IF **nazevpromenne, operator porovnani, hodnota THEN**

instrukce

instrukce

...

ELSE

instrukce

instrukce

...

ENDIF

Argumenty

nazevpromenne
operátor porovnání

konstanta nebo proměnná nebo nazevzarizeni

k symbolům, které mohou být použity pro provedení porovnání, patří:

< (menší) = (rovný)

hodnota	> (větší) =< (menší rovný)
návěstí	>= (větší rovný) <> (odlišný)
nazevpodprogramu	konstanta nebo proměnná nebo nazevzarizeni
nazevfunkce	název návěstí, na které má být proveden skok název podprogramu název funkce

Popis

Instrukce IF a IFVALUE představují synonyma. Doporučuje se používat kratší podobu.

Tato instrukce umožňuje provést porovnání mezi proměnnou určenou parametrem **nazevpromenne** a hodnotou určenou parametrem **hodnota** a na základě výsledku porovnání provést určený úkon.

V prvních třech podobách bude v případě kladného výsledku porovnání proveden skok na návěstí (GOTO) nebo bude provedeno vyvolání podprogramu (CALL) nebo vyvolání funkce (nazevfunkce). Po skončení provádění vyvolané funkce nebo vyvolaného podprogramu bude provádění obnoveno z následujícího řádku. Když je výsledek porovnání záporný, provádění programu bude pokračovat.

Vazba IF...THEN umožňuje provést jednu nebo více instrukcí podmíněným způsobem. Instrukce, které se nacházejí mezi klíčovými slovy THEN a ENDIF, budou provedeny, když je výsledek porovnání mezi parametry **nazevpromenne** a **hodnota** kladný.

Vazba IF...THEN...ELSE umožňuje definovat dva bloky instrukcí, ze kterých jeden bude proveden. Když je výsledek porovnání mezi parametry **nazevpromenne** a **hodnota** kladný, budou provedeny instrukce, které se nacházejí mezi klíčovými slovy THEN a ELSE; když je uvedený výsledek záporný, budou provedeny instrukce, které se nacházejí mezi klíčovými slovy ELSE a ENDIF. V obou případech bude poté provádění pokračovat instrukcí, která následuje po ENDIF.

Poznámka

IFVALUE je zachováno kvůli kompatibilitě s předchozími verzemi GPL.

IFACC**Syntaxe**

IFACC	jednotkapohybuose, GOTO navesti
IFACC	jednotkapohybuose, CALL nazevpodprogramu
IFACC	jednotkapohybuose, nazevfunkce

Argumenty

jednotkapohybuose	název zařízení druhu jednotka pohybu v ose
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověření toho, zda se jednotkapohybuose definovaná proměnnou **jednotkapohybuose** nachází ve stavu zrychlení.

Když je podmínka splněna, přeskočí na **navesti** nebo vyvolá **podprogram nazevpodprogramu** nebo funkci **nazevfunkce**.

IFAND**Syntaxe**

IFAND	operand1, operand2, hodnotaovereni, GOTO navesti
IFAND	operand1, operand2, hodnotaovereni, CALL nazevpodprogramu
IFAND	operand1, operand2, hodnotaovereni, nazevfunkce

IFAND	operand1, operand2, hodnotaovereni THEN
--------------	--

instrukce
instrukce
...

ENDIF

IFAND	operand1, operand2, hodnotaovereni THEN
--------------	--

instrukce
instrukce
...

ELSE

instrukce
instrukce
...

ENDIF

Argumenty

operand1	konstanta nebo proměnná nebo nazevzarizeni
operand2	konstanta nebo proměnná nebo nazevzarizeni
hodnotaovereni	konstanta. Hodnota použitá pro ověření výsledku operace. Může nabýt následujících hodnot: TRUE 1, FALSE 0
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Jsou prováděny dvě porovnání, první mezi **operand1** a **operand2** a druhé mezi výsledkem prvního porovnání a **hodnotaovereni**.

První porovnání spočívá v binárním součinu AND mezi **operand1** a **operand2**. Oba operandy jsou považovány za masky bitů. Když má výsledek binárního součinu AND alespoň jeden bit odlišný od 0, výsledkem prvního porovnání je TRUE. Ten je poté porovnán s **hodnotaovereni**. Když se obě hodnoty shodují, bude proveden skok na určené návěstí nebo vyvolání funkce nebo podprogramu.

Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFBIT**Syntaxe**

```
IFBIT maska, cislobitu, stav, GOTO navesti
IFBIT maska, cislobitu, stav, CALL nazevpodprogramu
IFBIT maska, cislobitu, stav, CALL nazevfunkce
```

```
IFBIT maska, cislobitu, stav THEN
```

```
instrukce
instrukce
```

```
...
```

```
ENDIF
```

```
IFBIT maska, cislobitu, stav THEN
```

```
instrukce
instrukce
```

```
...
```

```
ELSE
```

```
instrukce
instrukce
```

```
...
```

```
ENDIF
```

Argumenty

maska	konstanta nebo proměnná integer nebo nazevpocitadla nebo nazevportu. Hodnota, kterou je třeba ověřit
cislobitu	konstanta nebo proměnná integer nebo nazevpocitadla. Číslo bitu (1÷32)
stav	přednastavená konstanta. Stav, který je třeba ověřit na masce. Přípustnými hodnotami jsou: ON zvolený bit na 1 OFF zvolený bit na 0
navesti	návěstí skoku (GOTO)
nazevpodporogramu	vyvolání podprogramu (CALL)
nazevfunkce	název funkce

Popis

Ověření jednotlivého bitu procházeného parametru **maska**. Argument **maska** musí mít možnost odpovídat hodnotě celého čísla s maximem 32 bitů. Číslo, které má být přiřazeno proměnné **cislobitu** za účelem identifikace bitu, který má být ověřen, se pohybuje od 1 do 32. Když je podmínka uvedená v parametru **stav** ověřena, přeskočí na **navesti** nebo vyvolá **nazevpodprogramu** nebo **nazevfunkce**.

Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFBLACKBOX**Syntaxe**

```
IFBLACKBOX GOTO návěstí
IFBLACKBOX CALL nazevpodporogramu
IFBLACKBOX nazevfunkce
```

Argumenty

navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Když je záznam aktivní, přeskočí na **návěstí** nebo vyvolá **nazevpodprogramu** nebo funkci **nazevfunkce**.
Přečtete si také [STARTBLACKBOX](#), [PAUSEBLACKBOX](#) a [ENDBLACKBOX](#).

IFCHANGEVEL**Syntaxe**

IFCHANGEVEL	jednotkapohybuvose [, stav], GOTO navesti
IFCHANGEVEL	jednotkapohybuvose [, stav], CALL nazevpodprogramu
IFCHANGEVEL	jednotkapohybuvose [, stav], nazevfunkce

Argumenty

jednotkapohybuvose	název zařízení druhu jednotka pohybu v ose
stav	druh změny. Přípustnými hodnotami jsou: POSITIVE , NEGATIVE
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Slouží k ověření uskutečnění změny rychlosti jednotky pohybu v ose.
Když jednotka pohybu v ose definované proměnnou **jednotkapohybuvose** provádí změnu rychlosti během přesunu, bude proveden skok na **navesti** nebo vyvolání **nazevpodprogramu** nebo **nazevfunkce**.
Parametr **stav** upřesňuje, zda došlo ke zvýšení (POSITIVE) nebo ke snížení (NEGATIVE) rychlosti.

IFCOUNTER**Syntaxe**

IFCOUNTER	nazevpocitadla , operator srovnani , hodnota , GOTO navesti
IFCOUNTER	nazevpocitadla , operator srovnani , hodnota , CALL nazevpodporgramu
IFCOUNTER	nazevpocitadla , operator srovnani , hodnota , nazevfunkce
IFCOUNTER	nazevpromenne , operator srovnani , hodnota THEN
instrukce	
instrukce	
...	
ENDIF	
IFCOUNTER	nazevpromenne , operator srovnani , hodnota THEN
instrukce	
instrukce	
...	
ELSE	
instrukce	
instrukce	
...	
ENDIF	

Argumenty

nazevpocitadla	název počítadla
operátor porovnání	k symbolům, které mohou být použity pro provedení porovnání, patří: < (menší) = (rovný) > (větší) =< (menší rovný) >= (větší rovný) <> (odlišný)
hodnota	konstanta nebo proměnná nebo nazevpocitadla
navesti	název návěstí, na které má být proveden skok
nazevpodporogramu	název podprogramu
nazevfunkce	název funkce

Popis

Tato instrukce provádí ověření počítadla.
Když je obsah počítadla definován v proměnné **nazevpocitadla**, ověří podmínku určenou **operátorem porovnání** s hodnotou vyjádřenou proměnnou **hodnota**, bude proveden skok na návěstí určené parametrem **navesti** nebo vyvolání podprogramu definovaného parametrem **nazevpodprogramu** nebo funkce definované parametrem **nazevfunkce**.

Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFDEC

Syntaxe

IFDEC	jednotkapohybu vose, GOTO navesti
IFDEC	jednotkapohybu vose, CALL nazevpodprogramu
IFDEC	jednotkapohybu vose, nazevfunkce

Argumenty

jednotkapohybu vose	název zařízení druhu jednotka pohybu v ose
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověření toho, zda se jednotkapohybu vose definovaná proměnnou **jednotkapohybu**vose nachází ve stavu zpomalení.
Když je podmínka splněna, přeskočí na **navesti** nebo vyvolá **podprogram nazevpodprogramu** nebo funkci **nazevfunkce**.

IFDIR

Syntaxe

IFDIR	jednotkapohybu vose, smer , GOTO navesti
IFDIR	jednotkapohybu vose, smer , CALL nazevpodprogramu
IFDIR	jednotkapohybu vose, smer , nazevfunkce

IFDIR	jednotkapohybu vose, smer THEN
-------	--

```
instrukce
instrukce
...
```

ENDIF

IFDIR	jednotkapohybu vose, smer THEN
-------	--

```
instrukce
instrukce
...
```

ELSE

```
instrukce
instrukce
...
```

ENDIF

Argumenty

jednotkapohybu vose	název zařízení druhu jednotka pohybu v ose
smer	směr jednotky pohybu v ose. Přípustnými hodnotami jsou: POSITIVE kladný směr jednotky pohybu v ose NEGATIVE záporný směr jednotky pohybu v ose
návěstí	název návěstí, na které má být proveden skok
nazevpodporogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověření aktuálního směru jednotky pohybu v ose.
Když se **jednotka pohybu v ose** pohybuje ve směru určeném proměnnou **smer**, bude proveden skok na **navesti** nebo vyvolání **nazevpodprogramu** nebo **nazevfunkce**.
Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFERRAN

Syntaxe

IFERRAN	nazevpocitadla , operator porovnaní , hodnota , GOTO navesti
IFERRAN	nazevpocitadla , operator porovnaní , hodnota , CALL
	nazevpodprogramu
IFERRAN	nazevpocitadla , operator porovnaní , hodnota , nazevfunkce

IFERRAN	nazevpocitadla , operator porovnaní , hodnota THEN
---------	---

```

    instrukce
    instrukce
    ...
ENDIF

IFERRAN                nazevpocitadla, operator porovnaní, hodnota THEN
    instrukce
    instrukce
    ...
ELSE
    instrukce
    instrukce
    ...
ENDIF

```

Argumenty

jednotkapohybuvose	název zařízení druhu jednotka pohybu v ose
operátor porovnaní	k symbolům, které mohou být použity pro provedení porovnaní, patří: < (menší) = (rovný) > (větší) =< (menší rovný) >= (větší rovný) <> (odlišný)
hodnota	konstanta nebo proměnná nebo nazevpocitadla
návěští	název návěští, na které má být proveden skok
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověření hodnoty chyby sledování (chyby smyčky) jednotky pohybu v ose definované v proměnné

jednotkapohybuvose.

Když chyba smyčky **jednotkapohybuvose** ověří splnění podmínky vyjádřené **operátorem porovnaní** s hodnotou vyjádřenou parametrem **hodnota**, bude proveden skok na **navesti** nebo vyvolání podprogramu **nazevpodprogramu** nebo funkce **nazevfunkce**.

Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFERROR**Syntaxe**

```

IFERROR                cislo, OZNpol, GOTO navesti
IFERROR                cislo, OZNpol, CALL navesti
IFERROR                cislo, OZNpol, nazevfunkce
IFERROR                nazevzarizeni, stav,OZNpol, GOTO navesti
IFERROR                nazevzarizeni, stav,OZNpol, CALL navesti
IFERROR                nazevzarizeni, stav,OZNpol, nazevfunkce

```

Argumenty

cislo	DEFMSG nebo konstanta nebo proměnná integer
nazevzarizeni	název zařízení
stav	přednastavená konstanta. Může nabýt následujících hodnot: ON, OFF
OZNpol	konstanta nebo proměnná. Číselná hodnota používaná v grafických zobrazeních
navesti	název návěští, na které je třeba přeskočit
nazevfunkce	název funkce

Popis

Ověření chyby aktivního cyklu.

Když je chyba cyklu identifikována parametry **cislo** a **OZNpol** nebo parametry **nazevzarizeni**, **stav** a **OZNpol** aktivní, bude proveden skok na **navesti** nebo vyvolání funkce **nazevfunkce**, Parametr **cislo** může identifikovat chybu cyklu modulu (tedy hodnotu celého čísla) nebo skupiny (v tomto případě se používá DEFMSG).

Parametr **nazevzarizeni** je název zařízení a parametr **stav** představuje stav ON/OFF, ve kterém se musí nacházet zařízení v okamžiku vytvoření chyby.

Parametr **OZNpol** je volitelný parametr, který vymezuje číselnou hodnotu použitou pro třídění chyb cyklu do odlišných polí. Musí odpovídat hodnotě v programu pro tvorbu grafických znázornění pro konkrétní zobrazovací pole. Když není třeba uvádět specifické pole, uvádí se přednastavená konstanta NOPLACE. Rozsah nastavitelných hodnot je od 0 (NOPLACE) do 1023.

Když je instrukce použita, aniž by byla správa alarmů aktivována, bude vytvořena chyba systému.

Viz také instrukce [ERROR](#).

IFFLAG

Syntaxe

IFFLAG	nazevpriznaku, stav, GOTO navesti
IFFLAG	nazevpriznaku, stav, CALL nazevpodprogramu
IFFLAG	nazevpriznaku, stav, nazevfunkce

IFFLAG	nazevpriznaku, stav THEN
instrukce	
instrukce	
...	
ENDIF	

IFFLAG	nazevpriznaku, stav THEN
instrukce	
instrukce	
...	
ELSE	
instrukce	
instrukce	
...	
ENDIF	

Argumenty

nazevpriznaku	název zařízení druhu příznaku
stav	přednastavená konstanta. Stav, který má být ověřen. Přípustnými hodnotami jsou: ON zapnutý OFF vypnutý
navesti	název návěstí, na které má být proveden skok
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověření logického stavu příznaku.

Když se příznak definovaný proměnnou **nazevpriznaku** nachází ve stavu uvedeném parametrem **stav**, bude proveden skok na **navesti** nebo vyvolání podprogramu **nazevpodprogramu** nebo funkce **nazevfunkce**.

Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFINPUT

Syntaxe

IFINPUT	nazevvstupu, stav, GOTO navesti
IFINPUT	nazevvstupu, stav, CALL nazevpodprogramu
IFINPUT	nazevvstupu, stav, nazevfunkce

IFINPUT	nazevvstupu, stav THEN
instrukce	
instrukce	
...	
ENDIF	

IFINPUT	nazevvstupu, stav THEN
instrukce	
instrukce	
...	
ELSE	
instrukce	
instrukce	
...	
ENDIF	

Argumenty

nazevvstupu	název vstupu
stav	přednastavená konstanta. Stav, který má být ověřen Přípustnými hodnotami jsou: - ON zapnutý - OFF vypnutý
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu

nazevfunkce

název funkce

Popis

Ověření logického stavu vstupu.

Když se vstup definovaný proměnnou **nazevvstupu** nachází ve stavu uvedeném parametrem **stav**, bude proveden skok na **navesti** nebo vyvolání podprogramu **nazevpodprogramu** nebo funkce **nazevfunkce**. Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFMESSAGE**Syntaxe**

IFMESSAGE
IFMESSAGE
IFMESSAGE

cislo, OZNpol, GOTO navesti
cislo, OZNpol, CALL navesti
cislo, OZNpol, nazevfunkce

Argumenty

cislo
OZNpol

navesti
nazevfunkce

DEFMSG nebo konstanta nebo proměnná integer
konstanta nebo proměnná. Číselná hodnota používaná v grafických zobrazeních
název návěstí, na které je třeba přeskočit
název funkce

Popis

Ověření aktivního hlášení.

Když je hlášení identifikované parametry **cislo** a **OZNpol** aktivní, bude proveden skok na **navesti** nebo vyvolání funkce **nazevfunkce**, Parametr **OZNpol** je volitelný parametr, který vymezuje číselnou hodnotu použitou pro třídění chyb cyklu do odlišných polí. Musí odpovídat hodnotě v programu pro tvorbu grafických znázornění pro konkrétní zobrazovací pole. Když není třeba uvádět specifické pole, uvádí se přednastavená konstanta NOPLACE. Rozsah nastavitelných hodnot je od 0 (NOPLACE) do 1023.

Když je instrukce použita, aniž by byla správa alarmů aktivována, bude vytvořena chyba systému.

Viz také instrukce [MESSAGE](#).

IFOR**Syntaxe**

IFOR
IFOR
IFOR

operand1, operand2, hodnotaovereni, GOTO navesti
operand1, operand2, hodnotaovereni, CALL nazevpodprogramu
operand1, operand2, hodnotaovereni, nazevfunkce

IFOR
instrukce
instrukce
...
ENDIF

operand1, operand2, hodnotaovereni THEN

IFOR
instrukce
instrukce
...
ELSE
instrukce
instrukce
...
ENDIF

operand1, operand2, hodnotaovereni THEN

Argumenty

operand1
operand2
hodnotaovereni

konstanta nebo proměnná nebo nazevzarizeni
konstanta nebo proměnná nebo nazevzarizeni
konstanta. Hodnota použitá pro ověření výsledku operace.
Může nabýt následujících hodnot:

navesti
nazevpodprogramu
nazevfunkce

TRUE 1
FALSE 0
název návěstí, na které má být proveden skok
název podprogramu
název funkce

Popis

Jsou prováděny dvě porovnání, první mezi **operand1** a **operand2** a druhé mezi výsledkem prvního porovnání a **hodnotaovereni**.

První porovnání spočívá v binárním součtu OR mezi **operand1** a **operand2**. Oba operandy jsou považovány za masky bitů. Když má výsledek binárního součtu OR alespoň jeden bit odlišný od 0, výsledkem prvního porovnání je TRUE. Ten je poté porovnán s **hodnotaovereni**. Když se obě hodnoty shodují, bude proveden skok na určené návěští nebo vyvolání funkce nebo podprogramu. Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFOUTPUT

Syntaxe

```

IFOUTPUT nazevvystupu, stav, GOTO navesti
IFOUTPUT nazevvystupu, stav, CALL nazevpodprogramu
IFOUTPUT nazevvystupu, stav, nazevfunkce

IFOUTPUT nazevvystupu, stav THEN
    instrukce
    instrukce
    ...
ENDIF

IFOUTPUT nazevvystupu, stav THEN
    instrukce
    instrukce
    ...
ELSE
    instrukce
    instrukce
    ...
ENDIF

```

Argumenty

nazevvystupu	název výstupu
stav	přednastavená konstanta. Představuje stav, který má být ověřen na výstupu. Přípustnými hodnotami jsou: ON zapnutý OFF vypnutý
návěští	název návěští, na které má být proveden skok
nazevpodporogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověření logického stavu výstupu. Když se výstup definovaný proměnnou **nazevvystupu** nachází ve stavu uvedeném parametrem **stav**, bude proveden skok na **navesti** nebo vyvolání prodprogramu **nazevpodprogramu** nebo funkce **nazevfunkce**. Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFQUOTER

Syntaxe

```

IFQUOTER jednotkapohybuvoze, operator porovnaní, hodnota, GOTO navesti
IFQUOTER jednotkapohybuvoze, operator porovnaní, hodnota, CALL
nazevpodprogramu
IFQUOTER jednotkapohybuvoze, operator porovnaní, hodnota, nazevfunkce

IFQUOTER jednotkapohybuvoze, operator porovnaní, hodnota THEN
    instrukce
    instrukce
    ...
ENDIF

IFQUOTER jednotkapohybuvoze, operator porovnaní, hodnota THEN
    instrukce
    instrukce
    ...
ELSE
    instrukce
    instrukce
    ...
ENDIF

```

Argumenty

jednotkapohybu operátor porovnání	název zařízení druhu jednotka pohybu v ose k symbolům, které mohou být použity pro provedení porovnání, patří: < (menší) = (rovný) > (větší) =< (menší rovný) >= (větší rovný) <> (odlišný)
hodnota návěstí nazevpodprogramu nazevfunkce	konstanta nebo proměnná nebo nazevpocitadla název návěstí, na které má být proveden skok název podprogramu název funkce

Popis

Ověření reálné polohy vyjádřené proměnnou **jednotkapohybu**.
Když hodnota proměnné **jednotkapohybu** ověří splnění podmínky vyjádřené **operátorem porovnání** s hodnotou vyjádřenou parametrem **hodnota**, bude proveden skok na **navesti** nebo vyvolání podprogramu **nazevpodprogramu** nebo funkce **nazevfunkce**.
Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFQUOTET**Syntaxe**

IFQUOTET IFQUOTET	jednotkapohybu , operator porovnaní , hodnota , GOTO navesti jednotkapohybu , operator porovnaní , hodnota , CALL nazevpodprogramu
IFQUOTET	jednotkapohybu , operator porovnaní , hodnota , nazevfunkce
IFQUOTET instrukce instrukce ... ENDIF	jednotkapohybu , operator porovnaní , hodnota THEN
IFQUOTET instrukce instrukce ... ELSE instrukce instrukce ... ENDIF	jednotkapohybu , operator porovnaní , hodnota THEN

Argumenty

jednotkapohybu operátor porovnání	název zařízení druhu jednotka pohybu v ose k symbolům, které mohou být použity pro provedení porovnání, patří: < (menší) = (rovný) > (větší) =< (menší rovný) >= (větší rovný) <> (odlišný)
hodnota návěstí nazevpodprogramu nazevfunkce	konstanta nebo proměnná nebo nazevpocitadla název návěstí, na které má být proveden skok název podprogramu název funkce

Popis

Ověření teoretické polohy vyjádřené proměnnou **jednotkapohybu**.
Když hodnota proměnné **jednotkapohybu** ověří splnění podmínky vyjádřené **operátorem porovnání** s hodnotou vyjádřenou parametrem **hodnota**, bude proveden skok na **navesti** nebo vyvolání podprogramu **nazevpodprogramu** nebo funkce **nazevfunkce**.
Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFRECEIVED**Syntaxe**

IFRECEIVED	[zdroj,] identifikator , GOTO navesti
IFRECEIVED	[zdroj,] identifikator , CALL nazevpodprogramu
IFRECEIVED	[zdroj,] identifikator , nazevfunkce

Argumenty

zdroj	konstanta řetězec řetězec
identifikátor	konstanta řetězec řetězec
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověřuje, zda byla splněna instrukce [RECEIVE](#).
 V případě, že byla některá předcházející instrukce [RECEIVE](#) splněna, bude proveden skok na **navesti** nebo vyvolání podprogramu **nazevpodprogramu** nebo funkce **nazevfunkce**.
 Viz také instrukce [RECEIVE](#), [WAITRECEIVE](#), [SEND](#).

IFREG

Syntaxe

IFREG	jednotkapohybuvose, GOTO navesti
IFREG	jednotkapohybuvose, CALL nazevpodprogramu
IFREG	jednotkapohybuvose, nazevfunkce

Argumenty

jednotkapohybuvose	název zařízení druhu jednotka pohybu v ose
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověření toho, zda se jednotkapohybuvose definovaná proměnnou **jednotkapohybuvose** nachází ve stavu režimu.
 Když je podmínka splněna, přeskočí na **navesti** nebo vyvolá podprogram **nazevpodprogramu** nebo funkci **nazevfunkce**.

IFSAME

Syntaxe

IFSAME	operand1, operand2, GOTO navesti
IFSAME	operand1, operand2, CALL nazevpodprogramu
IFSAME	operand1, operand2, nazevfunkce

Argumenty

operand1	proměnná nebo nazevzarizeni
operand2	proměnná nebo nazevzarizeni
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověření mezi dvěma operandy.
 Ověřuje, zda se hodnoty definované v **operand1** a **operand2** se vztahují na stejné zařízení nebo na stejnou oblast paměti.
 Po splnění ověření mezi dvěma operandy bude proveden skok na **navesti** nebo vyvolání podprogramu **nazevpodprogramu** nebo vyvolání funkce **nazevfunkce**.

IFSTILL

Syntaxe

IFSTILL	jednotkapohybuvose, GOTO navesti
IFSTILL	jednotkapohybuvose, CALL nazevpodprogramu
IFSTILL	jednotkapohybuvose, nazevfunkce

Argumenty

jednotkapohybuvose	název zařízení druhu jednotka pohybu v ose
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověření toho, zda je jednotkapohybuvose definovaná proměnnou **jednotkapohybuvose** zastavena v poloze určené parametrem „poloha“.

Když je podmínka splněna, přeskočí na **navesti** nebo vyvolá **podprogram nazevpodprogramu** nebo funkci **nazevfunkce**.

Viz také [IFTARGET](#) a [IFWIN](#).

IFSTR

Syntaxe

IFSTR **retezec1, operator porovnani, retezec2, GOTO navesti**
IFSTR **retezec1, operator porovnani, retezec2, CALL nazevpodprogramu**
IFSTR **retezec1, operator porovnani, retezec2, nazevfunkce**

IFSTR **retezec1, operator porovnani, retezec2 THEN**

instrukce
instrukce

...

ENDIF

IFSTR **retezec1, operator porovnani, retezec2 THEN**

instrukce
instrukce

...

ELSE

instrukce
instrukce

...

ENDIF

Argumenty

retezec1

operátor porovnání

proměnná druhu řetězec. První řetězec ASCII

K symbolům, které mohou být použity pro provedení porovnání řetězců, patří:

< (menší) = (rovný)

> (větší) =< (menší rovný)

>= (větší rovný) <> (odlišný)

retezec2

návěští

nazevpodporogramu

nazevfunkce

proměnná druhu řetězec. Druhý řetězec ASCII

název návěští, na které má být proveden skok

název podprogramu

název funkce

Popis

Ověření řetězců ASCII.

Když hodnota definovaná v parametru **retezec1** ověří splnění podmínky vyjádřené **operátorem porovnání** s řetězcem obsaženým v parametru **retezec2**, bude proveden skok na **navesti** nebo vyvolání podprogramu **nazevpodprogramu** nebo funkce **nazevfunkce**.

Ohledně podrobnějších informací si prohlédněte vazba [IF-THEN-ELSE](#).

IFTARGET

Syntaxe

IFTARGET **jednotkapohybuose, GOTO navesti**
IFTARGET **jednotkapohybuose, CALL nazevpodprogramu**
IFTARGET **jednotkapohybuose, nazevfunkce**

Argumenty

jednotkapohybuose

navesti

nazevpodprogramu

nazevfunkce

název zařízení druhu jednotka pohybu v ose

název návěští, na které je třeba přeskočit

název podprogramu

název funkce

Popis

Ověření toho, zda jednotka pohybu v ose definovaná proměnnou **jednotkapohybuose** dosáhla naprogramované finální polohy. I když jednotka pohybu v ose dosáhla finální cílové polohy, nemusí být bezpodmínečně zastavená; obvykle ještě musí provést kompenzaci chyby smyčky. Když je podmínka splněna, přeskočí na **navesti** nebo vyvolá **podprogram nazevpodprogramu** nebo funkci **nazevfunkce**.

Viz také [IFSTILL](#) a [IFWIN](#).

IFTASKHOLD

Syntaxe

IFTASKHOLD	nazevulohy, GOTO navesti
IFTASKHOLD	nazevulohy, CALL navesti
IFTASKHOLD	nazevulohy, nazevfunkce

Argumenty

nazevulohy	název paralelní úlohy
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověřuje, zda je úloha pozastavena (stav HOLD).

Když je úloha s názvem **nazevulohy** pozastavena, bude proveden skok na **navesti** nebo vyvolání podprogramu určeného parametrem **nazevpodprogramu** nebo funkce určené parametrem **nazevfunkce**.

IFTASKRUN

Syntaxe

IFTASKRUN	nazevulohy, GOTO navesti
IFTASKRUN	nazevulohy, CALL nazevpodprogramu
IFTASKRUN	nazevulohy, nazevfunkce

Argumenty

nazevulohy	název paralelní úlohy
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověřuje, zda je úloha prováděna.

Když je úloha s názvem **nazevulohy** prováděna, bude proveden skok na **navesti** nebo vyvolání podprogramu určeného parametrem **nazevpodprogramu** nebo funkce určené parametrem **nazevfunkce**.

IFTIMER

Syntaxe

IFTIMER	nazevcasovace, operator porovnaní, hodnota, GOTO navesti
IFTIMER	nazevcasovace, operator porovnaní, hodnota, CALL
IFTIMER	nazevpodprogramu
IFTIMER	nazevcasovace, operator porovnaní, hodnota, CALL nazevfunkce

IFTIMER	nazevcasovace, operator porovnaní, hodnota THEN
----------------	--

```
instrukce
instrukce
```

```
...
ENDIF
```

IFTIMER	nazevcasovace, operator porovnaní, hodnota THEN
----------------	--

```
instrukce
instrukce
```

```
...
ELSE
```

```
instrukce
instrukce
```

```
...
ENDIF
```

Argumenty

nazevcasovace	název zařízení druhu časovač
operátor porovnání	K symbolům, které mohou být použity pro provedení porovnání řetězců, patří: < (menší) = (rovný) > (větší) =< (menší rovný) >= (větší rovný) <> (odlišný)
hodnota	konstanta nebo proměnná nebo nazevcasovace Hodnota, na které má být provedeno porovnání

Když je podmínka splněna, přeskočí na **navesti** nebo vyvolá **podprogram nazevpodprogramu** nebo funkci **nazevfunkce**.
 Viz také [IFTARGET](#) a [IFSTILL](#).

IFXOR

Syntaxe

IFXOR **operand1, operand2, hodnotaovereni, GOTO navesti**
IFXOR **operand1, operand2, hodnotaovereni, CALL nazevpodprogramu**
IFXOR **operand1, operand2, hodnotaovereni, nazevfunkce**

IFXOR **operand1, operand2, hodnotaovereni THEN**

instrukce
 instrukce

...

ENDIF

IFXOR **operand1, operand2, hodnotaovereni THEN**

instrukce
 instrukce

...

ELSE

instrukce
 instrukce

...

ENDIF

Argumenty

operand1	konstanta nebo proměnná nebo nazevzarizeni
operand2	konstanta nebo proměnná nebo nazevzarizeni
hodnotaovereni	konstanta. Hodnota použitá pro ověření výsledku operace. Může nabýt následujících hodnot: TRUE 1, FALSE 0
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Jsou prováděny dvě porovnání, první mezi **operand1** a **operand2** a druhé mezi výsledkem prvního porovnání a **hodnotaovereni**.

První porovnání spočívá v exkluzivním binárním součtu XOR mezi **operand1** a **operand2**. Oba operandy jsou považovány za masky bitů. Když má výsledek exkluzivního binárního součtu XOR alespoň jeden bit odlišný od 0, výsledkem prvního porovnání je **TRUE**. Ten je poté porovnán s **hodnotaovereni**. Když se obě hodnoty shodují, bude proveden skok na určené návěstí nebo vyvolání funkce nebo podprogramu.

Ohledně podrobnějších informací si prohlédněte vazba [_IF-THEN-ELSE](#).

ONERRSYS

Syntaxe

ONERRSYS **nazevfunkce**

Argumenty

nazevfunkce **název funkce**

Popis

Slouží k aktivaci správy chyb systému. Běžným chováním řízení při výskytu chyby systému je ukončení všech úloh. Správa chyb systému však umožňuje zabránit ukončení úloh, pro které byla aktivována.

Při výskytu chyby systému bude provedena funkce **nazevfunkce**. Úkolem této funkce je analyzovat chybu systému a provést vhodná opatření pro zabezpečení stroje.

Funkce **nazevfunkce** podléhá dvěma omezením. Prvním je akceptování níže uvedených parametrů:

- Číslo chyby systému jako Integer
- úloha, ve které se vyskytla chyba jako Function
- zařízení, které vytvořilo chybu jako device

Druhým je podmínka, že nesmí obsahovat určitý počet instrukcí GPL. Viz [seznam instrukcí nepoužitelných v rámci přerušení \(interrupt\)](#).

V případě vícenásobných Chyb Systému bude funkce vyvolána jednou pro každou vytvořenou chybu, a to sekvenčním způsobem. Když samotná funkce vytvoří Chybu Systému, budou ukončeny všechny úlohy.

Během provádění funkce bude úloha, pro kterou byla aktivovaná správa chyb, zastavena a bude obnovena teprve po skončení první funkce vyvolané instrukcí ONERRSYS. Konkrétně bude provádění úkolu obnoveno opětovným provedením příkazu přerušného chybou systému.

Příklad

[Cyklus hlavní části kódu \(Main\) se správou Chyb](#)

ONFLAG

Syntaxe

ONFLAG **nazevpriznaku, [stav,] nazevfunkce[,argumenty]**

Argumenty

nazevpriznaku	název zařízení příznak
stav	přednastavená konstanta. Stav, který má být ověřen. Přípustnými hodnotami jsou: - ON zapnutý - OFF vypnutý
nazevfunkce	název funkce
argumenty	případné argumenty funkce

Popis

Slouží k aktivaci softwarového přerušení úlohy, ve které je provedena, přiřazeného stavu uvedeného příznaku. Při přepnutí příznaku do uvedeného stavu (interrupt) bude provádění úlohy přerušeno a bude spuštěna funkce určená parametrem **nazevfunkce**. Po ukončení tohoto provádění bude úloha obnovena od stavu, ve kterém byla přerušena. Funkce provedená při výskytu přerušení (interrupt) podléhá omezení. Především tomu, že ne všechny instrukce GPL se mohou objevit v těle funkce. Účelem tohoto omezení je zabránit situacím kritického zablokování kódu GPL nebo dlouhým čekáním. Viz [seznam instrukcí nespustitelných v rámci přerušení \(interrupt\)](#).

Když je argument **stav** vynechán, funkce bude vyvolána při každé změně stavu příznaku.

Ověření stavu příznaku bude provedeno každých 5 ms, od změny příznaku po spuštění funkce tedy může uplynout maximálně 5 ms.

Není možné definovat více instrukcí ONFLAG na stejném příznaku.

V úloze argumentů funkce definované parametrem **nazevfunkce** nelze použít vektory nebo místní matrice. Viz také instrukce [DELONFLAG](#), [ONINPUT](#), [DELONINPUT](#).

ONINPUT

Syntaxe

ONINPUT **nazevvstupu, [stav,] nazevfunkce [,argumenty]**

Argumenty

nazevvstupu	název vstupu
stav	přednastavená konstanta. Stav, který má být ověřen. Přípustnými hodnotami jsou: ON zapnutý OFF vypnutý
nazevfunkce	název funkce
argumenty	případné argumenty funkce

Popis

Slouží k aktivaci softwarového přerušení úlohy, ve které je provedena, přiřazeného stavu uvedeného vstupu. Při přepnutí vstupu do uvedeného stavu (interrupt) bude provádění úlohy přerušeno a bude spuštěna funkce určená parametrem **nazevfunkce**. Po ukončení tohoto provádění bude úloha obnovena od stavu, ve kterém byla přerušena. Funkce provedená při výskytu přerušení (interrupt) podléhá omezení. Především tomu, že ne všechny instrukce GPL se mohou objevit v těle funkce. Účelem tohoto omezení je zabránit situacím kritického zablokování kódu GPL nebo dlouhým čekáním. Viz [seznam instrukcí nespustitelných v rámci přerušení \(interrupt\)](#).

Když je argument **stav** vynechán, funkce bude vyvolána při každé změně stavu vstupu.

Ověření stavu vstupu je prováděno každých 5 ms, ke kterým je třeba přičíst 4 ms filtru na ochranu před odražením ve správě vstupů, a proto před spuštěním funkce může uplynout doba 9 ms.

Není možné definovat více instrukcí ONINPUT na stejném vstupu.

Viz také instrukce [DELONINPUT](#), [ONFLAG](#) a [DELONFLAG](#).

REPEAT/ENDREP

Syntaxe

```

REPEAT                hodnota
    instrukce
    instrukce
    ...
ENDREP

```

Argumenty

hodnota konstanta nebo proměnná nebo nazevpocitadla. Počet opakování

Popis

Slouží k zopakování provádění instrukcí uzavřených mezi instrukci REPEAT a instrukci ENDREP, přičemž počet opakování bude odpovídat hodnotě proměnné **hodnota**. Když program narazí na instrukci ENDREP, počítadlo počtu opakování bude sníženo a v případě, že nebude menší nebo rovné nule, blok instrukcí bude znovu proveden počínaje instrukcí na řádce následujícím po REPEAT. Instrukce budou tedy provedeny nejméně jednou (i když je parametr hodnota nulový nebo záporný). Po skončení opakování bude provedena instrukce, která následuje po ENDREP. Viz také instrukci [FOR/NEXT](#).

Příklad

```

; příklad cyklu, který pohybuje jednotkou pohybu v ose
; 10-krát mezi dvěma polohami
Function Cyklus
Repeat 10
    MovAbs    jednotkapohybuose, 100
    waitinput spínač, ON
    Movabs    jednotkapohybuose, -100
    waitinput spínač, OFF
EndRep
Fret

```

RET

Syntaxe

```
RET
```

Argumenty

žádný argument

Popis

Slouží k ukončení podprogramu s návratem na instrukci následující bezprostředně po instrukci CALL použité k vyvolání.

Viz také instrukce [CALL](#).

Poznámka

Tato instrukce spolu s CALL představuje typický zdroj chyb programování. Doporučuje se proto věnovat pozornost jejímu použití, zejména se doporučuje umísťovat podprocedury na konci funkce (po instrukci FRET), aby se zabránilo náhodnému provedení kódu podprocedury, jakoby se jednalo o hlavní kód. Výsledkem této situace je v nejlepším případě chyba systému a v ostatních případech je výsledkem poruchové chování stroje, jehož příčinu lze obtížně určit.

SELECT

Syntaxe

```

SELECT nazevpromenne
    CASE hodnota
        instrukce
    CASE hodnota1 TO hodnota2
        instrukce
    CASE IS < = > hodnota
        instrukce
    CASE ELSE
        instrukce
ENDSELECT

```

pokyn musí být nahrazen jednou z následujících hodnot:

GOTO *navesti*
CALL *nazevpodprogramu*
[FCALL] *nazevfunkce [parametr1,...parametrN]*
[EXPR] *proměnná = výraz*
[EXPR] *zařízení = výraz*

Argumenty

nazevpromenne	konstanta nebo proměnná nebo nazevpocitadla
hodnota, hodnota1, hodnota2	konstanty integer
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce
parametr1...parametrN	parametr předaný funkci vyvolání
proměnná	název proměnné
zařízení	název zařízení
výraz	souhm operátorů

Popis

Vícenásobná volba na základě **hodnoty** proměnné **nomevar**. Bude proveden kód, který se nachází ve větvi CASE ověřené podmínky. Větev CASE - ELSE bude provedena v případě, že nebyla uspokojena žádná předešlá instrukce CASE.

Pro každou instrukci CASE (volitelná funkce) může být pouze jedna instrukce **GOTO**, **CALL** nebo **FCALL** nebo **EXPR**.

Mezi instrukcemi SELECT a ENDSELECT musí být nejméně jedna instrukce CASE. Instrukce ENDSELECT označuje konec instrukce SELECT.

Po každé instrukci CALL nebo FCALL nebo EXPR bude provádění funkce pokračovat na instrukci, která následuje po instrukci ENDSELECT.

Příklad

[Server pohybu jednotek pohybu v osách](#)

TESTIPC

Syntaxe

TESTIPC	nazevIPC, [, nazevpromenne1 [, nazevpromenneN, ...]], GOTO
	navesti
TESTIPC	nazevIPC, [, nazevpromenne1 [, nazevpromenneN, ...]], CALL
	nazevpodprogramu
TESTIPC	nazevIPC, [, nazevpromenne1 [, nazevpromenneN, ...]],
	nazevfunkce
TESTIPC	nazevIPC, matrice[radek], GOTO
TESTIPC	navesti
TESTIPC	nazevIPC, matrice[radek], CALL
	nazevpodprogramu
TESTIPC	nazevIPC, matrice[radek], nazevfunkce
TESTIPC	nazevIPC, vektor, GOTO
TESTIPC	navesti
TESTIPC	nazevIPC, vektor, CALL
	nazevpodprogramu
TESTIPC	nazevIPC, vektor, nazevfunkce

Argumenty

nazevIPC	konstanta řetězec. Název IPC
nazevpromenne1[...nazevpromenneN]	konstanta nebo proměnná. Názvy proměnných 1÷N
matrice[radek]	konstanta nebo proměnná integer. Číslo řádku matrice
vektor	název vektoru
matrice	název matrice
navesti	název návěstí, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověření a příjem příkazu IPC.

Při prvním provedení instrukce TESTIPC bude přidělena sdílená paměť, jejíž velikost je vypočtena na základě rozměru dat, která jsou odesílána. Maximální velikost sdílené paměti je 64 Kb.

Sdílené paměti je přiřazen semafor, který umožňuje synchronizovat provádění úloh, které do ní přistupují.

Úloha, která přistupuje, ověří přítomnost zapnutého semaforu, přečte data ze sdílené paměti a poté vypne

semafor. Poté bude provedena instrukce skoku na návěští nebo funkce či podprogram, které byly popsány jako poslední parametr instrukce TESTIPC.

Viz také [SENDIPC](#) a [WAITIPC](#).

TESTMAIL

Syntaxe

TESTMAIL	postovnischranka , [nazevpromenne1 [...nazevpromenneN]], GOTO navesti
TESTMAIL	postovnischranka , [nazevpromenne1 [...nazevpromenneN]], CALL navesti
TESTMAIL	postovnischranka , [nazevpromenne1 [...nazevpromenneN]], nazevfunkce
TESTMAIL	postovnischranka , matrice[radek] , GOTO navesti
TESTMAIL	postovnischranka , matrice[radek] , CALL nazevpodprogramu
TESTMAIL	postovnischranka , matrice[radek] , nazevfunkce

Argumenty

postovnischranka	konstanta nebo proměnná integer (1÷256). Číslo poštovní schránky
[nazevpromenne1[...nazevpromenneN]]	proměnná integer. Názvy proměnných 1÷20
matrice[radek]	konstanta nebo proměnná integer. Číslo řádku matrice
navesti	název návěští, na které je třeba přeskočit
nazevpodprogramu	název podprogramu
nazevfunkce	název funkce

Popis

Ověření a příjem hlášení.

První instrukce TESTMAIL aplikovaná na **postovnischranku** vytvoří poštovní schránku.

Když se v poštovní schránce **postovnischranka** nachází hlášení, data odeslaná spolu s hlášením budou uložena do proměnných **nazevpromenne** (1÷20) pouze v případě, že jsou tyto proměnné uvedeny v řádku matrice uvedeném v **matrice[radek]**; dále bude proveden skok na **navesti** nebo vyvolání **podprogramu** nebo **nazevfunkce**.

Během provádění je kontrolován soulad mezi předanými daty a daty očekávanými instrukcí.

Viz také instrukce [SENDMAIL](#), [WAITMAIL](#) a [ENDMAIL](#).

10.3.13 Různé

CLEARERRORS

Syntaxe

CLEARERRORS [OZNpol]

Argumenty

OZNpol konstanta nebo proměnná. Jedná s o hodnotu používanou v grafických zobrazeních

Popis

Jejím prostřednictvím provede dozorný PC vymazání všech chyb cyklu, které byly předem odeslány instrukcí ERROR a které se týkají modulu, který provede instrukci. Parametr **OZNpol** je volitelný parametr, který vymezuje číselnou hodnotu použitou pro třídění chyb cyklu do odlišných polí. Musí odpovídat hodnotě v programu pro tvorbu grafických znázornění pro konkrétní zobrazovací pole. Albatros Tuto identifikaci používá pro správu chyb cyklu v oddělených řadách. Pro každé OZNpol je vytvořena jedna řada. Rozsah nastavitelných hodnot je od 0 (NOPLACE) do 1023. Když parametr **OZNpol** není uveden, bude provedeno vymazání všech chyb cyklu v přednastavené řadě i v případných jiných řadách.

Viz také instrukce [ERROR](#) a [DELEERROR](#).

CLEARMESSAGES

Syntaxe

CLEARMESSAGES [OZNpol]

Argumenty

OZNpol konstanta nebo proměnná. Jedná s o hodnotu používanou v grafických zobrazeních

Popis

Jejím prostřednictvím provede dozorný PC vymazání všech chyb cyklu, které byly předem odeslány instrukcí MESSAGE a které se týkají modulu, který provede instrukci. Parametr **OZNpol** je volitelný parametr, který vymezuje číselnou hodnotu použitou v grafických zobrazeních pro třídění hlášení do odlišných polí. Musí odpovídat hodnotě v programu pro tvorbu grafických znázornění pro konkrétní zobrazovací pole. Albatros tuto identifikaci používá pro správu hlášení v oddělených řádcích. Pro každé OZNpol je vytvořena jedna řada. Rozsah nastavitelných hodnot je od 0 (NOPLACE) do 1023. Když parametr **OZNpol** není uveden, bude provedeno vymazání všech hlášení v přednastavené řadě i v případných jiných řádcích. Viz také instrukce [MESSAGE](#) a [DELMESSAGE](#).

DEFMSG

Syntaxe

```
DEFMSG          navesti [, predponajazyka1], "retezechlasi", ... , [,
predponajazykaN, "retezechlasi"]
```

Argumenty

navesti	mnemotechnický název hlášení, které má být zobrazeno
predponajazyka	přednastavená konstanta. Jazyk, ve kterém je napsáno hlášení
retezechlasi	hlášení, které má být zobrazeno. Musí být zapsáno v uvozovkách (""))

Popis

Slouží k přiřazení parametru **navesti** hlášení. Instrukce DEFMSG musí být deklarována před implementací funkcí. Definování hlášení může být použito pouze uvnitř souboru (nebo skupiny), ve kterém je deklarováno. Je možné vložit hlášení v různých jazycích s použitím přednastavené konstanty **predponajazyka** (pro seznam předpon jazyka si přečtěte kapitolu „[Importovat hlášení](#)“). V tomto případě instrukce MESSAGE zobrazuje hlášení v jazyce odpovídajícím jazyku aktuálně používanému pro Albatros. Hlášení, kterému není přiřazena předpona, bude použito, když aktuálně používaný jazyk neodpovídá žádné z existujících předpon. Návěští jednotlivých jazyků musí být napsána všechna na jednom řádku nebo na více řádcích s tím, že přechod na nový řádek bude zadán prostřednictvím znaku "_", kterému předchází mezera. Instrukce DEFMSG může být předána jako parametr nebo jako funkce. Při tomto postupu může být funkce, která ji přijme, použita jako jeden ze tří argumentů instrukcí ERROR a MESSAGE. (Viz příklad 2). Viz také instrukce [MESSAGE](#), [DELMESSAGE](#), [ERROR](#), [DELEERROR](#).

Příklad 1

```
; přiřazení řetězci hlášení určitému návěští
; bez volby jazyka
DEFMSG MSG_GRU_1 "Hlášení skupiny 1"

; použití definice
MESSAGE MSG_GRU_1 ; zobrazuje: "Hlášení skupiny 1"

; přiřazení řetězce hlášení určitému návěští
; s volbou jazyka
DEFMSG MSG_GRU_1 ITA „Hlášení skupiny 1“
ENG „Message group 1“

; použití definice, když je jazykem
; softwaru Albatros ENG
MESSAGE MSG_GRU_1 ; zobrazuje: "Message group 1"
```

Příklad 2

```
; ve skupině:
DEFMSG MSG_TEST "Chyba provádění"

FUNCTION vyvolaTest
  Test MSG_TEST
FRET

; v knihovně:
DEFMSG MSG_BASE "Signalizace chyby: $1"
...
FUNCTION Test Public
  PARAM kód AS integer
  ERROR MSG_BASE NOPLACE NOSTORE kód
FRET
; Zobrazená chyba cyklu je: Signalizace chyby: Chyba provádění
```

DELAY

Syntaxe

DELAY **hodnota**

Argumenty

hodnota konstanta nebo proměnná. Doba opoždění vyjádřená v sekundách.

Popis

Vyčká na uplynutí doby specifikované prostřednictvím parametru **hodnota**. Po uplynutí uvedeného časového intervalu bude provedena následující instrukce. Minimální nastavitelná hodnota je 4 ms (0,004 sekundy).

DELError

Syntaxe

DELError **nazevzarizeni [,stav [, OZNpol [protokol]]]**
DELError **cislo [, OZNpol [protokol]]]**

Argumenty

nazevzarizeni název zařízení
cislo DEFMSG nebo konstanta nebo proměnná integer
OZNpol konstanta nebo proměnná. Jedná s o hodnotu používanou v grafických zobrazeních
stav přednastavená konstanta. Může nabýt následujících hodnot: **ON**, **OFF**
protokol přednastavená číselná konstanta nebo proměnná typu celé číslo (integer). Může nabýt následující hodnoty: **STORE** chyba uložená do souboru, **NOSTORE** chyba neuložená do souboru

Popis

Slouží k provedení vymazání chyby cyklu dozorným PC odeslané předtím instrukcí ERROR. V případě uvedení názvu zařízení bude namísto čísla předán dozornému PC druh a logická adresa zařízení. Aby mělo vymazání účinek, všechny hodnoty nastavené v parametrech se musí shodovat s těmi, které se používají pro vytvoření chyby. Nastavení parametru **protokol** na **STORE** znamená zaregistrování chyby cyklu do souboru záznamů chyb aktuálního měsíce. Chyba zaregistrovaná na záznamových souborech není odstraněna souborem, ale pouze oknem chyb. Do souboru bude přidán nový záznam vymazání chyby. Parametr **OZNpol** je volitelný parametr, který vymezuje číselnou hodnotu použitou pro třídění chyb cyklu do odlišných polí. Musí odpovídat hodnotě v programu pro tvorbu grafických znázornění pro konkrétní zobrazovací pole. Když není třeba uvádět specifické pole, uvádí se přednastavená konstanta NOPLACE. Rozsah nastavitelných hodnot je od 0 (NOPLACE) do 1023. Když se provádí správa chyb cyklu jako signalizací, jsou odesílány všechny požadavky vymazání. Když se správa alarmů provádí jako správa stavů, vymazání chyb cyklu bude odesláno pouze v případě, že byla chyba cyklu aktivní; v opačném případě bude instrukce DELError ignorována. Viz také instrukce [ERROR](#) a [CLEARERRORS](#).

DELMESSAGE

Syntaxe

DELMESSAGE **cislo [, OZNpol]**

Argumenty

cislo DEFMSG nebo konstanta nebo proměnná
OZNpol konstanta nebo proměnná. Číselná hodnotapoužívaná v grafických zobrazeních

Popis

Odesílá do PC požadavek na vymazání hlášení odeslaného předtím prostřednictvím instrukce MESSAGE. Když se provádí správa hlášení jako signalizací, budou vymazána všechna odpovídající hlášení. Když se provádí správa hlášení jako stavů, vymazání hlášení bude odesláno pouze v případě, že byl jeho stav aktivní; v opačném případě bude instrukce DELMESSAGE ignorována. Parametr **OZNpol** je volitelný parametr, který vymezuje číselnou hodnotu použitou pro třídění chyb cyklu do odlišných polí. Musí odpovídat hodnotě v programu pro tvorbu grafických znázornění pro konkrétní zobrazovací pole. Když není třeba uvádět specifické pole, uvádí se přednastavená konstanta NOPLACE. Rozsah nastavitelných hodnot je od 0 (NOPLACE) do 1023. Viz také instrukce [MESSAGE](#).

ERROR

Syntaxe

```
ERROR navezzarizeni [,stav [, OZNpol [, protokol ]]]
ERROR cislo [, OZNpol [, protokol [, arg1, ..., arg3]]]
```

Argumenty

navezzarizeni	název zařízení
cislo	DEFMSG nebo konstanta nebo proměnná
OZNpol	konstanta nebo proměnná. Číselná hodnota používaná v grafických zobrazeních
stav	přednastavená konstanta. Může nabýt následujících hodnot: ON, OFF
protokol	přednastavená číselná konstanta nebo proměnná integer. Může nabýt následujících hodnot: STORE chyba uložena do souboru NOSTORE chyba neuložena do souboru
arg1, ..., arg3	konstanta nebo zařízení nebo proměnná

Popis

Slouží k vytvoření chyby cyklu. Chyba je identifikována parametrem **cislo** nebo názvem zařízení. Parametr **cislo** může identifikovat chybu cyklu modulu (tedy hodnotu celého čísla) nebo skupiny (v tomto případě se používá DEFMSG). V případě uvedení názvu zařízení bude namísto čísla předán PC druh a logická adresa zařízení. Chyba cyklu je odeslána dozornému PC a zobrazena v chybovém řádku softwaru Albatros. Parametr **OZNpol** se používá v grafických zobrazeních pro třídění chyb cyklu do odlišných polí. Musí odpovídat hodnotě v programu pro tvorbu grafických záznamů pro konkrétní zobrazovací pole. Albatros Tuto identifikaci používá pro správu chyb cyklu v oddělených řadách. Pro každé OZNpol je vytvořena jedna řada. Když není uveden parametr OZNpol nebo když se používá přednastavená konstanta NOPLACE, chyba cyklu přejde do přednastavené řady (OZNpol=0). Rozsah nastavitelných hodnot je od 0 (NOPLACE) do 1023. Nastavení parametru **protokol** na **STORE** znamená zaregistrování chyby cyklu do souboru záznamů chyb aktuálního měsíce. Vytvoření vysokého počtu chyb (nebo jejich vymazání) může ohrozit úroveň výkonnosti vzdálených modulů. Dozorný PC musí provádět správu všech odeslaných chyb (a jejich případné vymazání), což může zpomalit odesílání důležitých dat, zejména pracovních programů, do řízení. Volitelné parametry **arg1, ..., arg3** umožňují definovat parametrická chybová hlášení. Do řetězce definice chybového hlášení budou vloženy značky, které budou nahrazeny při vytvoření chyby hodnotou nebo názvem zařízení nebo proměnné předané jako parametr. Značky, které se vkládají do řetězce, jsou:

- \$1, ... \$2 nahrazené **názvem** zařízení nebo proměnné (\$1 odpovídá arg1 apod.)
- \$(1), ..., \$(3) nahrazené **hodnotou** zařízení nebo proměnnou

K datům přípustným pro parametry arg1, ..., arg3 patří:

- CHAR
- INTEGER
- FLOAT
- DOUBLE (který je však automaticky konvertován na FLOAT)
- číslo hlášení (nebo návěští pro DEFMSG)
- zařízení
- globální nebo místní proměnná
- parametr funkce. Jako parametr funkce je možné používat také návěští definované instrukcí [DEFMSG](#)

Jako parametry nemohou být použity řetězce, matrice a vektory (jednotlivé prvky vektoru nebo matrice jsou přípustné). Pro místní proměnné je možné dekódovat pouze hodnotu, a ne název.

Za účelem vymazání hlášení prostřednictvím instrukce DELERROR budou parametry arg1, ...arg3 ignorovány.

Jsou definovány dva způsoby správy chyb cyklu určené výrobcem stroje:

Alarmy spravované jako signalizace: Budou odesílány všechny chyby cyklu. Albatros udržuje řadu posledních 100 chyb určené řady a posledních 100 chyb přednastavené řady.

Alarmy spravované jako stav: Chyba je považována za aktivní nebo neaktivní. Je-li aktivní, každé další odeslání stejné chyby cyklu (prostřednictvím instrukce ERROR) bude ignorováno.

Viz také instrukce [DELERROR](#) a [CLEARERRORS](#).

Příklad 1

```
DEFMSG ERR_TOOL "Nástroj není přítomen"
DEFMSG ERR_TOOL_P "Naložte nástroj $(1) do polohy $(2)"

; značka pro grafická zobrazení
CONST TOOLCHANGE = 5

; chyba zobrazovaná v chybovém řádku nebo
; v neoznačených polích grafického zobrazení.
ERROR ERR_TOOL
```

```

; chyba ukládaná do záznamů a zobrazovaná v polích
; grafického zobrazení s kódem 5
ERROR      ERR_TOOL, TOOLCHANGE, STORE

; chyba ukládaná do záznamů ale nepřirázovaná
; polím grafického zobrazení
ERROR      ERR_TOOL, NOPLACE, STORE

; chyba s parametry
ERROR      ERR_TOOL_P, NOPLACE, NOSTORE, MxUtensili[3].Cod, 5

```

Příklad 2

```

; definovaná ve skupině
DEFMSG     MSG_ERR_CARICO  "Nebylo provedeno naložení nástroje"

```

```

Function ZobrazitHlaseni
  MsgTool MSG_ERR_CARICO MxUtensili[3].Cod
fret

```

```

Function ZobrazitChybu
  ErrTool STORE MSG_ERR_CARICO MxUtensili[3].Cod
Fret

```

```

; definovaná v knihovně
DEFMSG     MSG_ERR_TOOL    "Chyba nástroje: $1 $(2)"

```

```

Function MsgTool public
PARAM parametr1 as integer
PARAM parametr2 as integer

MESSAGE MSG_ERR_TOOL NOPLACE parameter1 parameter2
fret

```

```

Function ErrTool public
PARAM log as integer
PARAM argument1 as integer
PARAM argument2 as integer

ERROR MSG_ERR_TOOL NOPLACE log argument1 argument2
fret

```

IFDEF/ELSEDEF/ENDDF**Syntaxe**

```

IFDEF          konstanta
  instrukce
  ...
ENDDF

IFDEF          konstanta, operátor porovnání, hodnota
  instrukce
  ...
ENDDF

IFDEF          EXIST, nazevskupiny
  instrukce
  ...
ENDDF

IFDEF          LINKED, názevzařízení
  instrukce
  ...
ENDDF

```

IFDEF instrukce ... ENDDF	UNLINKED, názevzařízení
IFDEF instrukce ... ELSEDEF instrukce ... ENDDF	konstanta, operátor porovnání, hodnota

Argumenty

konstanta	konstanta druhu integer, char, double, řetězec
operátor porovnání	k symbolům, které mohou být použity pro provedení porovnání, patří: < (menší) = (rovný) > (větší) =< (menší rovný) >= (větší rovný) <> (odlišný)
hodnota	konstanta nebo název zařízení
nazevskupiny	konstanta řetězec nebo název skupiny
nazevzarizeni	konstanta řetězec nebo název zařízení

Popis

Podmíněná kompilace umožňuje kontrolovat, které části souboru function GPL musí být kompilovány, tedy provedeny. Kompilátor ověří, zda je podmínka požadovaná v podobě argumentu instrukce IFDEF splněna. V tomto případě bude kód, který se nachází mezi instrukcí IFDEF a instrukcí ENDDF nebo ELSEDEF, kompilován. V případě přítomnosti instrukce ELSEDEF a nesplnění podmínky bude provedena kompilace kódu mezi instrukcí ELSEDEF a instrukcí ENDDF.

Podmínka kompilace může být vyjádřena různými způsoby:

- Po instrukci IFDEF je uveden název **konstanty**. V tomto případě je podmínka splněna, když existuje globální konstanta nebo aktuální skupiny s uvedeným názvem.
- Po instrukci IFDEF je uveden vztah mezi dvěma operátory a jedním operandem. Prvním operandem musí být konstanta. V tomto případě je podmínka splněna, když je vztah pravdivý (např. MAX_TOOLS = 100).
- po instrukci IFDEF bude uvedeno klíčové slovo EXIST nebo NOTEXIST, po kterém následuje název skupiny stroje nebo řetězec obsahující název skupiny stroje nebo název knihovny. V tomto případě je podmínka splněna, když se v Konfiguraci Stroje nachází nebo nenachází skupina se stejným názvem.
- po instrukci IFDEF se uvádí klíčové slovo LINKED nebo UNLINKED, po kterém následuje název zařízení. V tomto případě je podmínka ověřena, když zařízení je (LINKED) nebo není (UNLINKED) připojeno ve virtuálně-fyzickém deklarování. Název zařízení může být vyjádřen v podobě:
Název_Skupiny.Název_Podskupiny.Název_Zařízení nebo Název_Skupiny.Název_Zařízení nebo
Název_Podskupiny_NázevZařízení nebo Název_Zařízení. Když zařízení neexistuje v konfiguraci, považuje se za nepřipojené.

Je možné vložit více instrukcí IFDEF, přičemž je třeba mít na paměti, že každé instrukci IFDEF musí odpovídat instrukce ENDDF.

Příklad 1

```
; provádění kódu GPL se mění v závislosti na
; přítomnosti skupiny FREZA ve skupině
Const SkupinaFreza = "Freza"
IFDEF Exist SkupinaFreza
instrukce
instrukce
ELSEDEF
instrukce
instrukce
ENDDF
```

Příklad 2

```
; provádění kódu GPL se mění
; v závislosti na modulu
IFDEF _ID_MODULE = 1 ; provádí kompilaci instrukcí pro modul 1
instrukce
instrukce
ELSEDEF ; provádí kompilaci instrukcí pro všechny
ostatní moduly
instrukce
```

```

instrukce
ENDDF

; provádí kompilaci kódu pro verzi 3.2.0 softwaru Albatros

IFDEF _VER_MAJOR = 3
  IFDEF _VER_MINOR = 2
    IFDEF _VER_REVISION = 0
      instrukce
      instrukce
    ENDDF
  ENDDF
ENDDF

; provádí kompilaci kódu pro verzi
; service pack 10 Albatros
IFDEF _VER_SP = "Service Pack 10"
  instrukce
ENDDF

; provádí kompilaci kódu pouze v případě, že je systém
; konfigurován pro vzdálený modul
IFDEF _REMOTE_MODULE = 1 ; 1 = vzdálený modul, v opačném případě 0 =
místní modul
  instrukce
ENDDF

; provádí kompilaci kódu pro verzi
; 2.4 service pack 10 Albatros
IFDEF _VER_FULL = $0002040AH
  instrukce
ENDDF

```

Příklad 3

```

; provádění kódu GPL se mění
; v závislosti na tom, zda je zařízení připojeno ve virtuálně fyzickém
deklarování
IFDEF LINKED out1 ; když je Out1 připojeno, dojde ke spuštění
kódu

  instrukce
  instrukce
  instrukce
ENDIF

```

MESSAGE**Syntaxe**

```
MESSAGE cislo [, OZNpol [, arg1, ..., arg3]]
```

Argumenty

cislo	DEFMSG nebo konstanta nebo proměnná
OZNpol	kostanta nebo proměnná. Číselná hodnota používaná v grafických zobrazeních
arg1, ..., arg3	konstanta nebo zařízení nebo proměnná.

Popis

Vytváří hlášení pro obsluhu. Hlášení je identifikováno parametrem **cislo**. Parametr **cislo** může identifikovat chybu hlášení modulu (tedy hodnotu celého čísla) nebo skupiny (v tomto případě se používá DEFMSG). V rámci volitelných funkcí může být předán argument tvořený parametrem **IDposiz**. Určuje, ve kterém okně grafického zobrazení má být hlášení zobrazeno. Musí odpovídat hodnotě v programu pro tvorbu grafických znázornění pro konkrétní zobrazovací pole. Albatros Tuto identifikaci používá pro správu hlášení v oddělených řadách. Pro každé OZNpol je vytvořena jedna řada. Když OZNpol není určeno, hlášení bude zařazeno do přednastavené řady (OZNpol=0). Rozsah nastavitelných hodnot je od 0 (NOPLACE) do 1023.

Volitelné parametry **arg1, ..., arg3** umožňují definovat parametrická chybová hlášení. Do řetězce definice hlášení budou vloženy značky, které budou nahrazeny při vytvoření hlášení hodnotou nebo názvem zařízení nebo proměnnou předanou jako parametr. Značky, které se vkládají do řetězce, jsou:

- \$1, ... \$2 nahrazené **názvem** zařízení nebo proměnné (\$1 odpovídá arg1 apod.)

- \$(1), ..., \$(3) nahrazené **hodnotou** zařízením nebo proměnnou
- K datům přípustným pro parametry arg1, ..., arg3 patří:
 - CHAR
 - INTEGER
 - FLOAT
 - DOUBLE (který je však automaticky konvertován na FLOAT)
- číslo hlášení (nebo návěstí pro DEFMSG)
- zařízení
- globální nebo místní proměnná
- parametr funkce. Jako parametr funkce je možné používat také návěstí definované instrukcí [DEFMSG](#)

Jsou definovány dva způsoby správy hlášení určené výrobcem stroje:

Hlášení spravované jako signalizace: Jsou odesílána všechna hlášení. Albatros Udržuje řadu posledních 100 hlášení určené řady a posledních 100 hlášení přednastavené řady. Po naplnění řady hlášení dojde k přepsání nejstaršího hlášení. Když je předchozí hlášení řady identické s hlášením, které má být odesláno, hlášení odesláno nebude (stejně úloha, stejné číslo, stejný argument).

Hlášení spravované jako stavy: Hlášení je považováno za aktivní nebo neaktivní. Je-li aktivní, každé další odeslání stejného hlášení (prostřednictvím instrukce MESSAGE) bude ignorováno.

Jako parametry nemohou být použity řetězce, matrice a vektory (jednotlivé prvky vektoru nebo matrice jsou přípustné). Pro místní proměnné je možné dekódovat pouze hodnotu, a ne název.

Za účelem vymazání hlášení prostřednictvím instrukce DELMESSAGE budou parametry arg1, ...arg3 ignorovány.

Viz také instrukce [DELMESSAGE](#) a [CLEARMESSAGES](#).

Příklad 1

```
DEFMSG      MSG_TOOL      "Proveďte výměnu nástroje"
DEFMSG      MSG_TOOL_P    "Naložený nástroj č. $(1)"

; značka pro grafická zobrazení
CONST      TOOLCHANGE = 7

; hlášení zobrazované v chybovém řádku nebo
; v neoznačených polích grafického zobrazení.
MESSAGE    MSG_TOOL

; hlášení zobrazované v chybovém řádku a
; v polích grafického zobrazení označených kódem 7
MESSAGE    MSG_TOOL, TOOLCHANGE

; hlášení s parametry
MESSAGE    MSG_TOOL_P, NOPLACE, MxNastroje[3].Kod
```

Příklad 2

```
; definovaná ve skupině
DEFMSG      MSG_CARICO    "Naložení bylo provedeno"

Function ZobrazitHlaseni
MsgTool MSG_NALOZENI MxNastroje[3].Kod
fret

; definovaná v knihovně
DEFMSG      MSG_TOOL      "nástroj: $(1) $2"

Function MsgTool public
PARAM parametr1 as integer
PARAM parametr2 as integer

MESSAGE MSG_ERR_TOOL NOPLACE parametr1 parametr2
fret
```


SYSFAULT

Syntaxe

SYSFAULT

Argumenty

žádný argument

Popis

Slouží k vypnutí signálu SYSOK.

Vypnutí tohoto signálu se provádí kvůli informování o tom, že stroj není více zabezpečen (např.: Nejsou již prováděné úlohy GPL, které provádějí správu nouzového stavu).

Viz také instrukci [SYSOK](#).

SYSOK

Syntaxe

SYSOK

[nazevvstupu1 [, ... nazevvstupu8]]

Argumenty

nazevvstupu1 [...nazevvstupu8]název zařízení druhu digitální výstup

Popis

Informuje číslicové řízení o tom, které výstupy jsou připojeny k bezpečnostním obvodům stroje (může se jednat o jediný výstup připojený k bezpečnostnímu relé, které kontroluje napájení výkonových obvodů stroje). Výstupy jsou aktivovány poté, co číslicové řízení dokončilo inicializaci stroje a aktivovalo všechny úlohy správy nouzového stavu. Stroj může být tedy považován za zabezpečený. Je možné definovat celkově až do 8 digitálních výstupů. Na každém vzdáleném modulu může být aktivován pouze jeden výstup. Seznam výstupů deklarovaných v prvním použití instrukce SYSOK **nemůže** být změněn během případných následujících volání sysok s výjimkou případu, že k tomu dojde po inicializaci řízení. Při provedení instrukce bez parametrů bude obnoven signál SYSOK.

Viz také instrukci [SYSFAULT](#).

Poznámka

Instrukce SYSOK může být aktivována pouze:

- Na všech vzdálených modulech GreenBus v3.0 s digitálními výstupy.
- na vzdálených modulech GreenBus v4.0 typu TRS-IO.
- Na vzdálených modulech TRS-CAT, pouze na základní části a ne na rozšířeních, které obsahují verzi firmwaru 1.2 nebo vyšší (rev 1.00 vzdáleného modulu).

TYPEOF

Syntaxe

TYPEOF

nazev, vysledek

Argumenty

nazev

název zařízení, konstanta, nazevfunkce, proměnná, vektor, matrice nebo řádek matrice

vysledek

proměnná integer. Druh prvního argumentu

Popis

Vrátí do proměnné **vysledek** druh argumentu **nazev**.

WATCHDOG

Syntaxe

WATCHDOG

stav

Argumenty

stav

přednastavená konstanta. Může nabýt následujících hodnot: **ON, OFF**

Popis

Tento příkaz slouží k aktivaci použití sledovacího zařízení (watchdog), spojeného s hardwarem TMSWD.

Umožňuje identifikovat chybové situace, které vznikají během provádění kódu GPL.

Při prvním provedení parametru **stav** na hodnotu ON instrukce aktivuje použití watchdog. Ve všech dalších případech je třeba přiřadit ON parametru **stav**, aby se provedla aktualizace počítadla karty. Když aktualizace nebude provedena, dojde ke spuštění sledovacího zařízení a modul TMSWD zruší bezpečnostní výstup stroje.

Pro ukončení použití sledovacího zařízení je třeba přiřadit hodnotu VYP. parametru **stav**.

Tato instrukce je použitelná pouze s kartami TMSbus+, TMSCan+ a TMSCombo+ s verzí FPGA 2.0 nebo vyšší, s namontovaným hardwarovým modulem TMSWD.

Příklad

Funkce TestwatchDog autorun

watchdog ON ; aktivuje správu sledovacího zařízení

Loop:

watchdog ON ; provede aktualizaci počítadla karty

goto loop

fret

10.3.14 MECHATROLINK-II

MECCOMMAND

Syntaxe

MECCOMMAND

jednotkapohybuose,prikaz,parametry,odpoved,chyba

Argumenty

jednotkapohybuose	název zařízení druhu digitální jednotka pohybu v ose
prikaz	konstanta integer
parametry	vektor integer
odpoved	vektor integer
chyba	proměnná integer. Kód chyby

Popis

Odesílá pohonu uvedené **jednotkypohybuose prikaz** a čeká na její odpověď. Údaje potřebné k provedení příkazu jsou vloženy do vektoru **parametry**, zatímco údaje vrácené provedeným příkazem jsou uloženy do vektoru **odpoved**. Vektory **parametry** a **odpoved** musí mít stejnou velikost a maximální počet prvků 14. Uvažovanou hodnotou je nejnižší byte samostatných celých čísel. Parametr **chyba** obsahuje kódy případných chyb vytvořených během uvedené operace. Správa chybových kódů musí být prováděna z GPL formou chyb cyklu.

K vráceným kódům parametru **chyba** patří:

Kód Chyby	Hlášení
-40	Příkaz nedovolný v aktuálních podmínkách činnosti
-41	Chyba časového limitu během provádění příkazu MECHATROLINK-II
-44	Chyba časového limitu během provádění podpříkazu MECHATROLINK-II
-45	Chyba zapojení pohonu

Ohledně hodnot, které je třeba přiřadit parametrům **prikaz**, **parametry**, **odpoved** a **chyba**, je třeba vycházet z oficiální dokumentace Yaskawa MECHATROLINK-II. V ní jsou popsány hodnoty pro příkaz popsané počínaje indexem 2 až po index 15. Hodnoty, které je třeba nastavit pro podpříkazy, jsou v ní popsány počínaje indexem 18 až po index 32.

Hodnoty parametru **prikaz** se mohou následně měnit ve srovnání s oficiální dokumentací.

Příkazy mohou být rozděleny na:

- příkazy. Vyznačují se kódem v rozsahu od 0x00 do 0xFF. Z bezpečnostních důvodů jsou prováděny pouze při zakázaném servořízení.
- podpříkazy. Příkazy považované za podpříkazy musí přičíst k hodnotě uvedené v dokumentaci kód 0x100. Například příkaz NOP má dokumentovaný kód 0x00; při jeho interpretaci jako podpříkazu je to 0x100.
- procedury. Příkazy považované za procedury mají kód s hodnotou počínaje 0x200. Aktuálně jsou známe tyto procedury:
 - \$201H procedura aktivace parametrů offline (používá se při zakázané jednotce pohybu v ose)

Tato instrukce je použitelná pouze s kartami AlbMech, DualMech, DualMech Mono. Ohledně získání podrobnějších informací týkajících se použití této instrukce se obraťte na firmu TPA.

Poznámka

Tato instrukce působí na chování digitálních jednotek pohybu v ose, a proto musí být použita v kontrolovaném

MECGETPARAM

Syntaxe

MECGETPARAM **jednotkapohybu***vose*,**parametr**,**rozmer**,**data**,**chyba**

Argumenty

jednotkapohybu <i>vose</i>	název zařízení druhu digitální jednotka pohybu v ose
parametr	konstanta nebo proměnná integer
rozmer	konstanta nebo proměnná integer
data	proměnná integer
chyba	proměnná integer. Kód chyby

Popis

Čte parametr pohonu uvedené **jednotkypohybu***vose* a ukládá ho do proměnné **data**. Parametr **chyba** obsahuje kódy případných chyb vytvořených během uvedené operace. Správa chybových kódů musí být prováděna z GPL formou chyb cyklu.

K vráceným kódům parametru **chyba** patří:

Kód Chyby	Hlášení
-40	Příkaz nedovolený v aktuálních podmínkách činnosti
-41	Chyba časového limitu během provádění příkazu MECHATROLINK-II
-44	Chyba časového limitu během provádění podpříkazu MECHATROLINK-II
-45	Chyba zapojení pohonu

Ohledně hodnot, které mají být přiřazeny proměnným **parametr** a **rozmer**, vycházejte z oficiální dokumentace Yaskawa MECHATROLINK-II.

Tato instrukce je použitelná pouze s kartami AlbMech, DualMech, DualMech Mono. Ohledně získání podrobnějších informací týkajících se použití této instrukce se obraťte na firmu TPA.

MECGETSTATUS

Syntaxe

MECGETSTATUS **jednotkapohybu***vose*,**stav**,**vstupvystup**,**chyba**

Argumenty

jednotkapohybu <i>vose</i>	název zařízení druhu digitální jednotka pohybu v ose
stav	konstanta nebo proměnná integer
vstupvystup	konstanta nebo proměnná integer
chyba	proměnná integer. Kód chyby

Popis

Čte a ukládá do proměnné **stav** hodnotu STATUS a ALARM a do proměnné **vstupvystup** hodnotu IO_MON týkající se uvedené **jednotkypohybu***vose*. Ohledně hodnot STATUS, ALARM, IO_MON vycházejte z oficiální dokumentace Yaskawa MECHATROLINK-II.

Parametr **chyba** obsahuje kódy případných chyb vytvořených během uvedené operace. Správa chybových kódů musí být prováděna z GPL formou chyb cyklu.

K vráceným kódům **chyby** patří:

Kód Chyby	Hlášení
-40	Příkaz nedovolený v aktuálních podmínkách činnosti
-41	Chyba časového limitu během provádění příkazu MECHATROLINK-II
-44	Chyba časového limitu během provádění podpříkazu MECHATROLINK-II
-45	Chyba zapojení pohonu

Je definována série kategorií chyb. Kategorie představují hodnotu nejvyššího půlbajtu hodnoty ALARM.

Za účelem zrušení stavu alarmu při alarmech patřících do kategorií 0x30,0x70,0xD0,0xF0 musí být odeslán příkaz CLEAR (0x06). Alarmy, které spadají do kategorií 0x00,0x10,0x40,0xB0, nemohou být odstraněny žádným příkazem. V takovém případě je třeba vyřešit problém, který způsobuje alarm, vypnout servořízení a znovu jej zapnout.

Struktura proměnných **stav** a **inout** je tvořena maskou bitů organizovanou níže uvedeným způsobem:

STAV:	32	24	16	1
	0	ALARM	STATUS	
INOUT:	32	16	1	
	0	IO_MON		

STAV

8	7	6	5	4	3	2	1
PSET V_CMP	ZPOINT	Vyhrazenc	PON	SVON	CMDRDY	WRNG	ALM
16	15	14	13	12	11	10	9
Vyhrazenc	Vyhrazenc	N-SOT	P-SOT	NEAR V_LIM	L_CMP	T_LIM	DEN ZSPD

Význam bitů hodnoty STAV

Bit	Ovládací příkaz	Fyzické kolíky, které lze připojit ve virtuálním fyzickém zařízení
1	ALM (Alarm)	Digitální vstup
2	WRNG (Upozornění)	Digitální vstup
3	CMDRDY (Příkaz připraven)	
4	SVON (ZAPNUTÍ servořízení)	Digitální výstup
5	PON (ZAPNUTÍ hlavního napájení)	Digitální vstup
6	Vyhrazeno	
7	ZPOINT (Nulový bod)	
8	PSET (Dosažená poloha) V_CMP (Velocity Agreement)	
9	DEN (Příznak ukončené distribuce ovl. příkazu) ZSPD (Zero Velocity)	
10	T_LIM (Mezní hodnota krouticího momentu)	
11	L_CMP (Dosažení nulové polohy motoru)	
12	NEAR (Poloha bezdotykového spínače reagujícího na přiblížení) V_LIM (Velocity Limit)	
13	P-SOT (Koncový spínač pohybu vpřed)	
14	N-SOT (Koncový spínač zpětného pohybu)	
15	Vyhrazeno	
16	Vyhrazeno	

IO_MON

8	7	6	5	4	3	2	1
EXT2	EXT1	PC	PB	PA	DEC	N_OT	P_OT
16	15	14	13	12	11	10	9
IN4	IN3	IN2	IN1			BRK	EXT3

Význam bitů hodnoty IO_MON

Bit	Ovládací příkaz	Fyzické kolíky, které lze připojit ve virtuálním fyzickém zařízení
1	P_OT (Přejetí mezní polohy pohybu vpřed)	
2	N_OT (Přejetí mezní polohy zpětného pohybu)	
3	DEC (Koncový spínač zpomalení)	
4	PA (Fáze A)	
5	PB (Fáze B)	

6	PC (Fáze C)	Digitální vstup
7	EXT1 (První externí vstup nulové polohy motoru)	Digitální vstup
8	EXT2 (Druhý externí vstup nulové polohy motoru)	Digitální vstup
9	EXT3 (Třetí externí vstup nulové polohy motoru)	
10	BRK (Výstup ovládání brzdy)	
11		
12		
13	IN1 (Vstup 1 pro všeobecné účely)	
14	IN2 (Vstup 2 pro všeobecné účely)	
15	IN3 (Vstup 3 pro všeobecné účely)	
16	IN4 (Vstup 4 pro všeobecné účely)	

Tato instrukce je použitelná pouze s kartami AlbMech, DualMech a DualMech Mono. Ohledně získání podrobnějších informací týkajících se použití této instrukce se obraťte na firmu TPA.

MECSETPARAM

Syntaxe

MECSETPARAM **jednotkapohybu***vose*,**parametr**,**rozmer**,**data**,**chyba**

Argumenty

jednotkapohybu <i>vose</i>	název zařízení druhu digitální jednotka pohybu v ose
parametr	konstanta nebo proměnná integer
rozmer	konstanta nebo proměnná integer
data	proměnná integer
chyba	proměnná integer. Kód chyby

Popis

Slouží k zápisu **data** do **parametru** uvedené **jednotkypohybu***vose*. Ohledně hodnot, které mají být přiřazeny proměnným **parametr** a **rozmer**, vycházejte z oficiální dokumentace Yaskawa MECHATROLINK-II. Parametr **chyba** obsahuje kódy případných chyb vytvořených během uvedené operace. Správa chybových kódů musí být prováděna z GPL formou chyb cyklu. K vráceným kódům parametru **chyba** patří:

Kód Chyby	Hlášení
-40	Příkaz nedovoleno v aktuálních podmínkách činnosti
-41	Chyba časového limitu během provádění příkazu MECHATROLINK-II
-44	Chyba časového limitu během provádění podpříkazu MECHATROLINK-II
-45	Chyba zapojení pohonu

Tato instrukce je použitelná pouze s kartami AlbMech, DualMech, DualMech Mono. Ohledně získání podrobnějších informací týkajících se použití této instrukce se obraťte na firmu TPA.

Poznámka

Tato instrukce působí na chování digitálních jednotek pohybu v ose, a proto musí být použita v kontrolovaném kontextu.

Pro zápis **data** do stálé paměti je třeba použít instrukci [MECCOMMAND](#)

10.3.15 Standardní sběrnice pole

AXCONTROL

Syntaxe

AXCONTROL **jednotkapohybu***vose*,**data**

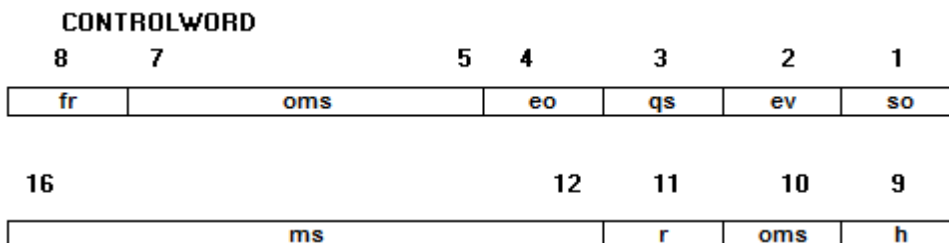
Argumenty

jednotkapohybu <i>vose</i>	název zařízení druhu jednotka pohybu v ose
data	proměnná nebo konstanta integer. Slouží k nastavení ControlWord

Popis

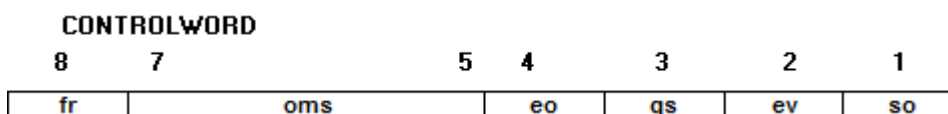
Slouží k nastavení **data** ControlWord způsobem, který je ve shodě s aktivní operativností, podle „CiA 402 CANopen device profile“.

Tabulka určení hodnot pro EtherCAT



Bit	Význam	Název ve virtuálním fyzickém
1	so=ZAPNUTÍ	CW1
2	ev=Aktivace napětí	EV
3	qs=Rychlé zastavení	STOP
4	eo=Aktivace činnosti	SVON
5	oms=Specifický režim činnosti	CW5
6	oms=Specifický režim činnosti	CW6
7	oms=Specifický režim činnosti	CW7
8	fr=Vynulování poruchy	RESALM
9	h=Ukončení	CW9
10	oms=Specifický režim činnosti	CW10
11	r=Vyhrazeno	CW11
12	ms=Specifický výrobce	CW12
13	ms=Specifický výrobce	CW13
14	ms=Specifický výrobce	CW14
15	ms=Specifický výrobce	CW15
16	ms=Specifický výrobce	CW16

Tabulka určení hodnot pro S-CAN



Bit	Význam	Název ve virtuálním fyzickém
1	Ten_cmd=příkaz aktivace krouticího momentu. 1:jednotka pohybu v ose ve fázi aplikace krouticího momentu 0:volná jednotka pohybu v ose	SVON
2	Ien_cmd=příkaz aktivace pohybů. 1:aktivované pohyby 0:jednotka pohybu v ose ve fázi zastavení aplikace krouticího momentu	ENMOVE
3	Stp_cmd=příkaz pro zastavení. 1:aktivovaný příkaz zastavení 0:neaktivovaný příkaz zastavení	STOP
4	Alm_rst=stav alarmů	RESALM

STATUSWORD

8	7	6	5	4	3	2	1
w	sod	qs	ve	f	oe	so	rtso

Bit	Význam	Název ve virtuálním fyzickém
1	Ten_st=stav aktivace krouticího momentu 1:jednotka pohybu v ose ve fázi aplikace krouticího momentu 0:volná jednotka pohybu v ose	SW1
2	Ien_st=stav aktivace pohybů 1:aktivované pohyby 0:jednotka pohybu v ose ve fázi zastavení aplikace krouticího momentu	SW2
3	Stp_st=stav zastavení 1:probíhá rampa zastavení... 0:zastavení není aktivováno nebo rampa není ukončena	SW3
4	Alm_st=stav alarmů 1:alarm stroje 0:nebyl zaznamenán žádný alarm	ALM
5	Ltc_st=stav nulové polohy 1:dosažení nulové polohy bylo provedeno a registr je připraven pro čtení 0:nebyla zaznamenána žádná nulová poloha	SW5
6	oms=specifický pro zvolený režim	SW6
7	oms=specifický pro zvolený režim	SW7
8	oms=specifický pro zvolený režim	SW8

CNBYDEVICE**Syntaxe****CNBYDEVICE****zarizeni, karta,cn****Argumenty****zarizeni**

název zařízení

karta

proměnná integer. Vrácené číslo karty

cn

proměnná integer. Vrácené číslo uzlu

Popis

Vrátí číslo karty a číslo uzlu zařízení definovaného v parametru **zařízení**. Tato instrukce slouží k umožnění použití instrukcí, které nemají přímé odkazy na zařízení, jako například instrukce [READDICTIONARY](#) a [WRITEDICTIONARY](#).

Příkaz může být použitý pro zařízení konfigurované na sběrnici CAN, S-CAN, GreenBUS nebo EtherCAT. Když je vrácené číslo uzlu záporná hodnota, znamená to, že uzel je zrušen.

READDICTIONARY**Syntaxe****READDICTIONARY****karta,cn,index,podindex,rozmerdata,data,chyba****Argumenty****karta**

konstanta nebo proměnná integer. Číslo karty

cn

konstanta nebo proměnná integer. Číslo uzlu

index

konstanta nebo proměnná integer. Index objekt ve slovníku

podindex

konstanta nebo proměnná integer. Podindex objektu ve slovníku

rozmerdata

proměnná integer. Velikost přečteného data.

data

proměnná char, integer, float, double, char vektor, string. Proměnná, která přijímá data

chyba proměnná integer. Kód chyby vrácen uzlem

Popis

Čte obsah objektu slovníku objektů obsažených v uzlu. Tato instrukce umožňuje číst prostřednictvím protokolu SDO všechny objekty definované podle „CiA 402 CANopen device profile“ plus všechny objekty zpřístupněné výrobcem uzlu. Ohledně významu parametrů **index**, **podindex** a **rozmerdata** vycházejte z „CiA 402 CANopen device profile“ nebo z předpisu výrobce uzlu. Pro zařízení S-CAN je třeba parametr podindex pokaždé nastavit na nulu.

WRITEDICTIONARY

Syntaxe

WRITEDICTIONARY karta, cn, index, podindex, rozmerdata, data, chyba

Argumenty

karta konstanta nebo proměnná integer. Číslo karty
cn konstanta nebo proměnná integer. Číslo uzlu
index konstanta nebo proměnná integer. Index objekt ve slovníku
podindex konstanta nebo proměnná integer. Podindex objektu ve slovníku
rozmerdata konstanta nebo proměnná integer. Rozměr data určeného k zápisu
data proměnná char, integer, float, double, char vektor, string. Proměnná, která obsahuje data
chyba proměnná integer. Kód chyby vrácen uzlem

Popis

Zapíše obsah objektu slovníku objektů obsažených v uzlu. Tato instrukce umožňuje zapsat prostřednictvím protokolu SDO všechny objekty zadané podle „CiA 402 CANopen device profile“ plus všechny objekty zpřístupněné výrobcem uzlu. Ohledně významu parametrů **index**, **podindex** a **rozmerdata** zapisovaných dat vycházejte z „CiA 402 CANopen device profile“ nebo z předpisu výrobce uzlu. Pro zařízení S-CAN je třeba parametr podindex pokaždé nastavit na nulu.

10.3.16 EtherCAT

ACTIVATEMODE

Syntaxe

ACTIVATEMODE jednotkapohybuose, data, chyba

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose
data konstanta nebo proměnná integer. Provozní režim
chyba proměnná integer. Kód chyby vrácený servořízením

Popis

Slouží k nastavení provozního režimu definovaného v proměnné **data** způsobem, který je ve shodě s „CiA 402 CANopen device profile“. Provozní režim jednotky pohybu v ose odpovídá při zahájení činností hodnotě **data=9**, tj. „režim cyklické synchronní rychlosti“. Instrukce vrací hodnotu **chyba=0** v případě úspěšného provedení příkazu; v opačném případě vrátí kód chyby. Níže je uvedena tabulka hodnot určených pro přiřazení **datu** pro volbu provozního režimu.

Hodnota	Stanovení
+6	režim homing
+9	režim cyklické synchronní rychlosti

ECATGETREGISTER

Syntaxe

ECATGETREGISTER uzel, adresa, velikost, data, chyba

Argumenty

uzel konstanta nebo proměnná integer. Pozice obsazená zařízením Slave v řetězu sítě EtherCAT (od 1 dále)
adresa konstanta nebo proměnná integer. Adresa registru ESC (EtherCAT Slave Controller), určená ke čtení (od 0 dále)
velikost konstanta nebo proměnná integer. Číslo bytů určených ke čtení (1, 2 nebo 4)
data proměnná integer. Kontejner přečteného data
chyba proměnná integer. Kód chyby

Popis

Vrací **[data]** obsah registru ESC (EtherCAT Slave Controller) uvedeného uzlu EtherCAT. Parametr **chyba** bude obsahovat číselný kód chyby nebo 0, když se nevyskytly chyby.

ECATSETREGISTER**Syntaxe**

ECATSETREGISTER **uzel,adresa,velikost,data,chyba**

Argumenty

uzel	konstanta nebo proměnná integer. Pozice obsazená zařízením Slave v řetězu sítě EtherCAR (od 1 dále)
adresa	konstanta nebo proměnná integer. Adresa registru ESC (EtherCAT Slave Controller), určená ke čtení (od 0 dále)
velikost	konstanta nebo proměnná integer. Číslo bytů určených ke čtení (1, 2 nebo 4)
data	konstanta nebo proměnná integer. Kontejner přečteného data
chyba	proměnná integer. Kód chyby

Popis

Zapíše **[data]** obsah registru ESC (EtherCAT Slave Controller) uvedeného uzlu EtherCAT. Parametr **chyba** bude obsahovat číselný kód chyby nebo 0, když se nevyskytly chyby.

GETPDO**Syntaxe**

GETPDO **karta,uzel,nPDO,nObj,data,[chyba]**

Argumenty

karta	konstanta nebo proměnná integer. Číslo karty
uzel	konstanta nebo proměnná integer. Pozice obsazená zařízením Slave v řetězu sítě EtherCAR (od 1 dále)
nPDO	konstanta nebo proměnná integer. Identifikační označení PDO (např. \$1600h) nebo jeho pozice v seznamu PDO nastavených v konfiguraci hardwaru pro daný uzel (od 1 do 16)
nObj	konstanta nebo proměnná integer. Identifikační označení objektu (např. \$6040h) nebo jeho pozice v seznamu konfigurovaných objektů v PDO (od 1 do 32)
data	proměnná integer. Přijímá hodnotu
chyba	proměnná integer. Kód chyby

Popis

Vrací v **[data]** obsah objektu vyměněného prostřednictvím PDO nakonfigurovaných pro uzel sítě EtherCAT. Když jsou prošlé argumenty chybné a nebyl nastaven parametr **chyby**, bude vytvořena chyba systému. V případě nastavení parametru **chyba** bude tento parametr obsahovat číselný kód odpovídající chyby systému.

SETEOE**Syntaxe**

SETEOE **stav,[chyba]**

Argumenty

stav	přednastavená konstanta. Přípustnými hodnotami jsou: - ZAP aktivuje správu protokolu EoE - VYP zruší správu protokolu EoE
chyba	proměnná integer. Kód chyby

Popis

Tato instrukce umožňuje aktivovat a zrušit správu protokolu EoE (Ethernet over EtherCAT). Tato funkce používá pomocný software SnifferCE nebo SnifferRTX, který provádí správu komunikace EoE mezi číslicovým řízením a dozorným PC. Proměnná chyby může nabýt níže uvedených hodnot:

- 0, když byl příkaz proveden správně;
- 1, když příkaz nebyl ukončen v určeném časovém limitu 5 sekund;
- 2, když došlo k výskytu chyby komunikace.

SETPDO

Syntaxe

SETPDO **karta, uzel, nPDO, nObj, data, [chyba]**

Argumenty

karta	konstanta nebo proměnná integer. Číslo karty)
uzel	konstanta nebo proměnná integer. Pozice obsazená zařízením Slave v řetězu sítě EtherCAR (od 1 dále)
nPDO	konstanta nebo proměnná integer. Identifikační označení PDO (např. \$1600h) nebo jeho pozice v seznamu PDO nastavených v konfiguraci hardwaru pro daný uzel (od 1 do 16)
nObj	konstanta nebo proměnná integer. Identifikační označení objektu (např. \$6040h) nebo jeho pozice v seznamu konfigurovaných objektů v PDO (od 1 do 32)
data	proměnná integer. Nastavená hodnota
chyba	proměnná integer. Kód chyby

Popis

Nastavuje obsah **[data]** objektu vyměněného prostřednictvím PDO nakonfigurovaných pro uzel sítě EtherCAT. Když jsou prošlé argumenty chybné a nebyl nastaven parametr **chyby**, bude vytvořena chyba systému. V případě nastavení parametru **chyba** bude tento parametr obsahovat číselný kód odpovídající chyby systému.

10.3.17 Karta TMSbus s řízením CAN

GETCNSTATE

Syntaxe

GETCNSTATE **karta, uzel, stav**

Argumenty

karta	konstanta nebo proměnná integer. Číslo karty
uzel	konstanta nebo proměnná integer. Číslo uzlu
stav	konstanta nebo proměnná integer

Popis

Vrací informaci o stavu protokolu NMT pro **uzel uvedené karty**. Podrobnější informace o významu těchto parametrů najdete přímo v dokumentaci příslušného zařízení.

GETSDOERROR

Syntaxe

GETSDOERROR **karta, chyba**

Argumenty

karta	konstanta nebo proměnná integer. Číslo karty (1 až 4)
chyba	proměnná integer. Kód chyby

Popis

Vrací poslední **chybu**, která se vyskytla, vztaženou na komunikaci SDO pro **uvedenou kartu**. Podrobnější informace o významu vráceného kódu chyby najdete přímo v dokumentaci příslušného zařízení.

GETMNSTATE

Syntaxe

GETMNSTATE **karta, stav**

Argumenty

karta	konstanta nebo proměnná integer. Číslo karty (1 až 4)
stav	konstanta nebo proměnná integer

Popis

Vrací informaci o stavu protokolu NMT pro uzel master **uvedené karty**. Podrobnější informace o významu těchto parametrů najdete přímo v dokumentaci příslušného zařízení.

RECEIVEPDO

Syntaxe

RECEIVEPDO **karta, uzel, čísloPDO**

Argumenty

karta konstanta nebo proměnná integer. Číslo karty (od 1 do 4)
uzel konstanta nebo proměnná integer. Číslo uzlu
čísloPDO konstanta nebo proměnná integer. Číslo PDO

Popis

Slouží ke čtení obsahu PDO vymezeného atributem **čísloPDO** pro uvedený uzel. Tato instrukce se používá pro čtení asynchronních PDO (tj. PDO, které v konfiguraci hardwaru povolily možnost **Asynchronní**).
 Přečtený údaj je kopírován ve zařízeních připojených ve virtuálního-fyzického připojení.
 Tuto instrukci lze použít pouze s kartami TMSCan a TMSCan+.

SENDPDO

Syntaxe

SENDPDO **karta, uzel, čísloPDO**

Argumenty

karta konstanta nebo proměnná integer. Číslo karty
uzel konstanta nebo proměnná integer. Číslo uzlu
čísloPDO konstanta nebo proměnná integer. Číslo PDO

Popis

Slouží k zápisu obsahu PDO vymezeného atributem **čísloPDO** pro uvedený uzel. Tato instrukce se používá pro zápis asynchronních PDO (tj. PDO, které v konfiguraci hardwaru povolily možnost **Asynchronní**).
 Tuto instrukci lze použít pouze s kartami TMSCan a TMSCan+.

SETNMTSTATE

Syntaxe

SETNMTSTATE **karta, uzel, stav**

Argumenty

karta konstanta nebo proměnná integer. Číslo karty (1 až 4)
uzel konstanta nebo proměnná integer. Číslo uzlu
stav konstanta nebo proměnná integer

Popis

Nastavuje stav protokolu NMT pro **uzel uvedené karty** CANOpen. Když se hodnota uzlu rovná 0 (zero) nebo je vyšší než 126, nastavení bude aplikováno na všechny konfigurované uzly přítomné na kanálu. Podrobnější informace o významu těchto parametrů najdete přímo v dokumentaci příslušného zařízení.

Hodnota	Stav protokolu
1	Provozní
128	Přípravný

10.3.18 Simulace

DISABLE

Syntaxe

DISABLE **jednotkapohybuose1,[...jednotkapohybuose6]**

Argumenty

jednotkapohybuose1... názvy zařízení druhu jednotka pohybu v ose
[...jednotkapohybuose6]

Popis

Slouží k zakázání uvedených jednotek pohybu v ose. Zakázání umožňuje provést simulace cyklů stroje bez reálného pohybu jednotek pohybu v osách. Zakázaná jednotka pohybu v ose nečte informace, které přicházejí ze snímače impulzů, ale simuluje chybu smyčky proporcionální aktuální rychlosti. Zakázání jednotky pohybu v ose však nezakáže řídicí signál rychlosti, a proto na konektoru jednotky pohybu v ose bude napětí

odlišné od nuly i v případě simulovaných pohybů. Proto je třeba vypnout napájení pohonů nebo je během simulovaných pohybů, tj. při zakázaných jednotkách pohybu v ose, odpojit z karty jednotek pohybu v osách. Viz také [ENABLE](#).

Poznámka

Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX.

DISABLEFORCEDINPUT

Syntaxe

DISABLEFORCEDINPUT

Argumenty

žádný argument

Popis

Zakáže možnost používání funkcí pro nucené nastavování vstupů. Když byly předtím nuceně nastaveny vstupy, provádění této instrukce je vrátí do reálného stavu. Viz také [ENABLEFORCEDINPUT](#), [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDPORT](#), [SETFORCEDANALOG](#).

ENABLE

Syntaxe

ENABLE **jednotkapohybuose1,[...jednotkapohybuose6]**

Argumenty

jednotkapohybuose1... názvy zařízení druhu jednotka pohybu v ose
[...jednotkapohybuose6]

Popis

Slouží k aktivaci uvedených jednotek pohybu v ose. Jednotky pohybu v ose jsou vždy aktivovány ve fázi inicializace. Tato instrukce je vyvolávána pouze v případě, že byly jednotky pohybu v ose předtím zakázány instrukcí [DISABLE](#).

Poznámka

Jednotky krokového pohybu v ose se mohou používat v této instrukci pouze v případě, že jsou spravovány ze vzdáleného TRS-AX.

ENABLEFORCEDINPUT

Syntaxe

ENABLEFORCEDINPUT

Argumenty

žádný argument

Popis

Aktivuje nucené nastavování vstupů. Před použitím instrukcí pro nucenou aktivaci nebo zakázání vstupních zařízení je třeba provést tuto instrukci. V opačném případě nebudou mít instrukce nuceného nastavení vstupů žádný vliv.

Viz také [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDPORT](#), [SETFORCEDANALOG](#).

RESETFORCEDINPUT

Syntaxe

RESETFORCEDINPUT **nazevvstupu**

Argumenty

nazevvstupu **název digitálního vstupu**

Popis

Slouží k nucenému nastavení vstupu uvedeného v parametru **nazevvstupu** na hodnotu OFF. Za účelem použití této instrukce je třeba předem aktivovat nucené nastavování vstupů prostřednictvím instrukce [ENABLEFORCEDINPUT](#). Viz také [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [SETFORCEDPORT](#), [SETFORCEDANALOG](#).

SETFORCEDANALOG

Syntaxe

SETFORCEDANALOG **analogovyvstup, hodnota**

Argumenty

analogovyvstup název zařízení druhu analogový vstup
hodnota konstanta nebo proměnná integer nebo float nebo double

Popis

Nuceně nastavuje **hodnotu** analogového vstupu určeného prostřednictvím parametru **analogovyvstup**. Za účelem použití této instrukce je třeba předem aktivovat nucené nastavování vstupů prostřednictvím instrukce [ENABLEFORCEDINPUT](#). Viz také [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDPORT](#).

SETFORCEDINPUT

Syntaxe

SETFORCEDINPUT **nazevvstupu**

Argumenty

nazevvstupu název digitálního vstupu

Popis

Slouží k nucenému nastavení vstupu uvedeného v parametru **nazevvstupu** na hodnotu ON. Za účelem použití této instrukce je třeba předem aktivovat nucené nastavování vstupů prostřednictvím instrukce [ENABLEFORCEDINPUT](#). Viz také [DISABLEFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDPORT](#), [SETFORCEDANALOG](#).

SETFORCEDPORT

Syntaxe

SETFORCEDPORT **nazevportu, hodnota**

Argumenty

nazevportu název zařízení druhu port vstupů
promenna konstanta nebo proměnná integer nebo char

Popis

Nuceně nastavuje **hodnotu** portu vstupů vyjádřenou parametrem **nazevportu**. Port vstupů je viděn jako maska bitů. Když má bit hodnotu 1, odpovídající vstup bude nastaven do stavu „ON“. Za účelem použití této instrukce je třeba předem aktivovat nucené nastavování vstupů prostřednictvím instrukce [ENABLEFORCEDINPUT](#). Viz také [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDANALOG](#).

10.3.19 BlackBox

Účelem funkce „BlackBox“ je zaznamenat do databáze veškerou aktivitu stroje, tj. místního nebo vzdáleného modulu. Pod pojmem „aktivita stroje“ se má na mysli změna podcelku všech logických zařízení, použitelných v GPL. Tímto způsobem je možné provést následnou analýzu chování stroje na základě korelace stavů zařízení uložených v paměti. Databáze má tabulku obsahující časovou informaci o čase a stavu všech zařízení v daném okamžiku, po jednom pro každý sloupec. V jazyce GPL byly zavedeny nové instrukce pro zahájení, ukončení a dotazování aktivity zaznamenávání, které jsou popsány níže.

Každý soubor blackbox představuje databázi SQLite a obsahuje informace, které se týkají pouze jediného modulu. Název souboru zahrnuje číslo modulu a datum a čas zahájení aktivity zaznamenávání.

Přidání záznamu do databáze se provádí prostřednictvím transakce. Každá transakce obsahuje více záznamů vytvořených v průběhu 1 sekundy. V případě výpadku dodávky elektrické energie je zaručena koherence souboru a může dojít ke ztrátě poslední transakce. Doba trvání transakce může být změněna prostřednictvím položky v Tpa.ini (ohledně podrobnějších informací se obraťte na firmu TPA.).

Byl zadán limit doby zachování záznamu, rovnající se 12 hodinám. To znamená, že každá databáze bude vždy obsahovat pouze posledních 12 hodin záznamu. Během zaznamenávání budou z databáze odstraněny starší

záznamy. Mezní dobu zachování archivního přehledu zaznamenaného v databázi lze měnit prostřednictvím položky v Tpa.ini (ohledně podrobnějších informací se obraťte na firmu TPA).

Tato funkce je dostupná pro fyzická zařízení na sběrnici GreenBus, EtherCAT, CAN, S-CAN, MECHATROLINK-II, která jsou připojena prostřednictvím sběrnice TMSbus, TMSbus+, TMScan, TMScan+, DualMech, DualMech Mono, AlbMech.

ENDBLACKBOX

Syntaxe

ENDBLACKBOX

Popis

Slouží k ukončení funkce zaznamenávání veškeré aktivity místního nebo vzdáleného modulu na soubor. Viz také [STARTBLACKBOX](#) a [PAUSEBLACKBOX](#).

PAUSEBLACKBOX

Syntaxe

PAUSEBLACKBOX

Popis

Slouží k pozastavení funkce zaznamenávání veškeré aktivity místního nebo vzdáleného modulu na soubor. Pro obnovení zaznamenávání stačí spustit instrukci [STARTBLACKBOX](#) bez argumentů. Viz také [ENDBLACKBOX](#).

STARTBLACKBOX

Syntaxe

STARTBLACKBOX

[hodnota][,chyba]

Argumenty

hodnota

konstanta nebo proměnná integer. Období zaznamenávání

chyba

proměnná integer. Kód chyby

Popis

Slouží k aktivaci funkce zaznamenávání veškeré aktivity místního nebo vzdáleného modulu na soubor. Aktivitou modulu se má na mysli změna stavu logických zařízení v čase s výjimkou zařízení typu přepínač příznaku.

Doba zaznamenávání (**hodnota**) je vyjádřena v milisekundách. Nesmí být menší než 10 a musí být násobkem doby real-time. Kdyby tomu tak nebylo, došlo by ke vzniku chyby systému č. 4399 (Parametr mimo rozsah). Když instrukce zahájí zaznamenávání a **hodnota** je vynechána, za přednastavenou hodnotu bude považována hodnota 20.

Když instrukce obnoví předtím přerušené zaznamenávání, nebude zohledněna žádná nastavená **hodnota**.

Když nebylo možné spustit zaznamenávání, **chyba** obsahuje hodnotu odlišnou od 0; v opačném případě obsahuje hodnotu 0:

Kód chyby	Popis
0	Žádná chyba
1	Existují rozdíly mezi konfigurací zařízení v číslicovém řízení a v systému Albatros
2	Počet zařízení, která mají být zaregistrována, přesahuje maximální počet určený pro systém
3	V konfiguraci nejsou přítomna zařízení
4	Komunikační software ve vzdáleném modulu nepodporuje funkci blackbox (pouze vzdálené moduly)
5	Číslicové řízení brání zahájení registrace
6	Vyskytla se chyba při načítání knihovny pro správu databáze
7	Počet sloupců na tabulku přesahuje maximální počet sloupců, které databáze dokáže spravovat
8	Nebylo možné otevřít databázi na disku
9	Není možné vytvořit v databázi tabulku zaznamenávání
10	Vyskytla se chyba v IP adrese pro komunikaci se vzdáleným modulem (pouze vzdálené moduly)
11	Nebylo možné vytvořit komunikační socket pro příjem dat (pouze vzdálené moduly)
12	Nebylo možné přiřadit místní adresu komunikačnímu socketu (pouze vzdálené moduly)
13	Nebylo možné se připojit ke vzdálenému socketu (pouze vzdálené moduly)
14	Nebyl možný přístup do paměti sdílené s číslicovým řízením
15	Konfigurace hardwaru brání použití funkce „BlackBox“
16	Daná funkce byla zrušena v tpa.ini

Viz také [PAUSEBLACKBOX](#) a [ENDBLACKBOX](#).

10.3.20 ISO

ISOGO

Syntaxe

ISOGO

navesti, jednotkapohybuose1, poloha1, jednotkapohybuose2, poloha2, jednotkapohybuose3, poloha3, jednotkapohybuose4, poloha4, jednotkapohybuose5, poloha5, [hodnota]

Argumenty

navesti

konstanta nebo proměnná integer. Návěští které identifikuje blok přesunu. N ve standardu ISO

jednotkapohybuose1 poloha1

název zařízení druhu jednotka pohybu v ose. (X ve standardu ISO)
konstanta nebo proměnná. Poloha v operačním prostoru
jednotkapohybuose1

jednotkapohybuose2 poloha2

název zařízení druhu jednotka pohybu v ose. (Y ve standardu ISO)
konstanta nebo proměnná. Poloha v operačním prostoru
jednotkapohybuose2

jednotkapohybuose3 poloha3

název zařízení druhu jednotka pohybu v ose. (Z ve standardu ISO)
konstanta nebo proměnná. Poloha v operačním prostoru
jednotkapohybuose3

jednotkapohybuose4 poloha4 jednotkapohybuose5 poloha5 hodnota

název zařízení druhu jednotka pohybu v ose. (C ve standardu ISO)
konstanta nebo proměnná. Poloha v prostoru spojů jednotkapohybuose4
název zařízení druhu jednotka pohybu v ose. (B ve standardu ISO)
konstanta nebo proměnná. Poloha v prostoru spojů jednotkapohybuose5
konstanta nebo proměnná double. Představuje procentuální hodnotu feed rate. (F ve standardu ISO)

Popis

Slouží k nastavení Rychlého pohybu. Úseky rychlého pohybu jsou řízeny v synchronizaci. Body definované uživatelem jsou hraničními body jednotlivého úseku přesunu, který je ujetý tak, aby byly jednotky pohybu v osách vzájemně synchronizovány. To znamená, že fyzické jednotky pohybu v ose se pohybují nezávisle jedna na druhé, ale zachovávají současnou rozjezd a příjezd podobně jako v instrukcích [MULTIABS](#) a [MULTIINC](#). Hrot obráběcího nástroje nepřejde čáru v provozním prostoru a jeho dráha nebude kontrolována. Parametr **navesti** se používá ve spojení s instrukcí [SETLABELINTERP](#) pro jedinečnou identifikaci bloku přesunu. První tři **polohy** identifikují polohu hrotu v provozním prostoru, zatímco další dvě definují hodnotu rotačních jednotek pohybu v osách v prostoru spojů. **Hodnota** FeedRate definuje procentuální hodnotu snížení vzhledem maximální možné rychlosti (V ISO: F0 maximální rychlost, F100 nulový FeedRate, a tedy zastavené jednotky pohybu v osách).

Tato instrukce způsobuje při použití na jednotkách krokového pohybu v ose vznik chyby systému (4105 - Danou instrukci nelze provést na jednotce pohybu `NazevJednotkypohybuose`).

Není možné použít instrukci [WAITCOLL](#), protože v okamžiku kolize vpředu by došlo ke ztrátě vztahu interpolace s ostatními jednotkami pohybu v osách, které se podílejí na pohybu, a výsledkem by byl vznik profilu rychlosti, který se liší od očekávaného profilu. V případě jejího použití bude vytvořena chyba systému 4101 - Nevhovující správa jednotky pohybu v ose `NazevJednotkypohybuose`.

ISOG1

Syntaxe

ISOG1

navesti, jednotkapohybuose1, poloha1, jednotkapohybuose2, poloha2, jednotkapohybuose3, poloha3, jednotkapohybuose4, poloha4, jednotkapohybuose5, poloha5, [hodnota]

Argumenty

návěští

konstanta nebo proměnná integer. Návěští které identifikuje blok přesunu. (N v standardu ISO)

jednotkapohybuose1 poloha1

název zařízení druhu jednotka pohybu v ose. (X ve standardu ISO)
konstanta nebo proměnná. Poloha v operačním prostoru
jednotkapohybuose1

jednotkapohybuose2 poloha2

název zařízení druhu jednotka pohybu v ose. (Y ve standardu ISO)
konstanta nebo proměnná. Poloha v operačním prostoru
jednotkapohybuose2

jednotkapohybuose3 poloha3

název zařízení druhu jednotka pohybu v ose. (Z ve standardu ISO)
konstanta nebo proměnná. Poloha v operačním prostoru
jednotkapohybuose3

jednotkapohybuose4

název zařízení druhu jednotka pohybu v ose. (C ve standardu ISO)

poloha4	konstanta nebo proměnná. Poloha v prostoru spojů jednotkypohybu4
jednotkapohybu4	název zařízení druhu jednotka pohybu v ose. (B ve standardu ISO)
poloha5	konstanta nebo proměnná. Poloha v prostoru spojů jednotkypohybu5
hodnota	konstanta nebo proměnná double. Představuje hodnotu posuvu. (F ve standardu ISO)

Popis

Definuje bod v provozním prostoru, kterého musí dosáhnout první hrot obráběcího nástroje na konci interpolace aktuálního bloku. Parametr **navesti** se používá ve spojení s instrukcí [SETLABELINTERP](#) pro jedinečnou identifikaci bloku přesunu. První tři **polohy** identifikují polohu hrotu v provozním prostoru, zatímco další dvě definují hodnotu rotačních jednotek pohybu v osách v prostoru spojů. **Hodnota** Feed definuje rychlost hrotu jako měrná jednotka (milimetry nebo stupně) za minutu (v případě přítomnosti instrukce [ISOG94](#)) nebo jako dobu provádění (v případě přítomnosti instrukce [ISOG93](#)). Parametr **hodnota** je povinný pro první instrukci ISOG1 pohybu interpolace.

Tato instrukce způsobuje při použití na jednotkách krokového pohybu v ose vznik chyby systému (4105 - Danou instrukci nelze provést na jednotce pohybu `NazevJednotkypohybu4`).

Není možné použít instrukci [WAITCOLL](#), protože v okamžiku kolize vpředu by došlo ke ztrátě vztahu interpolace s ostatními jednotkami pohybu v osách, které se podílejí na pohybu, a výsledkem by byl vznik profilu rychlosti, který se liší od očekávaného profilu. V případě jejího použití bude vytvořena chyba systému 4101 - Nevhovující správa jednotky pohybu v ose `NazevJednotkypohybu4`.

ISOG9**Syntaxe**

ISOG9 **jednotkapohybu4**

Argumenty

jednotkapohybu4 název zařízení druhu jednotka pohybu v ose

Popis

Umožňuje nucené zastavení pohybu. V případě přítomnosti této instrukce budou interpolace nebo rychlý pohyb zastaveny před přechodem na následující blok. Nejedná se proto o blokující instrukci, jako je tomu v případě instrukce [WAITSTILL](#). Řízení je informováno o nuceném zastavení a proces získání bloků pohybu bude pokračovat až do naplnění lookahead. Parametr **jednotkapohybu4** identifikuje kanál interpolace 5 Jednotek pohybu v osách po skončení předtím vypočteného bloku. V tomto případě nebude rozdíl mezi provedením instrukce [ISOG1](#) a instrukce [ISOG0](#).

ISOG90**Syntaxe**

ISOG90 **jednotkapohybu4**

Argumenty

jednotkapohybu4 název zařízení druhu jednotka pohybu v ose

Popis

Slouží k nastavení interpretace poloh jako absolutních. Parametr **jednotkapohybu4** identifikuje kanál interpolace 5 Jednotek pohybu v osách, který bude počínaje touto instrukcí interpretovat polohy jednotek pohybu v osách jako absolutní polohy (přednastavená podmínka). V tomto případě nebude rozdíl mezi provedením instrukce [ISOG1](#) a instrukce [ISOG0](#).

ISOG91**Syntaxe**

ISOG91 **jednotkapohybu4**

Argumenty

jednotkapohybu4 název zařízení druhu jednotka pohybu v ose

Popis

Slouží k interpretaci poloh jako relativních. Parametr **jednotkapohybu4** identifikuje kanál interpolace 5 Jednotek pohybu v osách, který bude počínaje touto instrukcí interpretovat polohy jednotek pohybu v osách jako relativní polohy. V tomto případě nebude rozdíl mezi provedením instrukce [ISOG1](#) a instrukce [ISOG0](#).

ISO93**Syntaxe****ISO93****jednotkapohybuosach****Argumenty****jednotkapohybuose**

název zařízení druhu jednotka pohybu v ose

Popis

Slouží k nastavení rychlostí jako obrácených hodnot doby provádění. Parametr **jednotkapohybuose** určuje kanál interpolace 5 Jednotek pohybu v osách, který bude počínaje touto instrukcí interpretovat hodnotu předanou prostřednictvím parametrů F instrukce [ISO91](#) jako obrácenou hodnotu doby provádění vyjádřené v minutách. Díky tomu je řízení schopno určit rychlost, která musí být udržována hrotem nástroje v blocích interpolace.

ISO94**Syntaxe****ISO94****jednotkapohybuose****Argumenty****jednotkapohybuose**

název zařízení druhu jednotka pohybu v ose

Popis

Slouží k nastavení rychlosti jako měrné jednotky za minutu. Parametr **jednotkapohybuose** určuje kanál.

ISO216**Syntaxe****ISO216**

nazevMatriceRotacnichjednotek, nazevMatriceDrzakuNastroju, nazevMatriceNastroju, maskaAktivace, jednotkapohybuose1, jednotkapohybuose2, jednotkapohybuose3, jednotkapohybuose4, jednotkapohybuose5

Argumenty

nazevMatriceRotacnichjednotek název matrice. Obsahuje údaje týkající se rotačních jednotek pohybu v osách

nazevMatriceDrzakuNastroju název matrice. Obsahuje údaje týkající se držáku nástrojů

nazevMatriceNastroju název matrice. Obsahuje údaje týkající se obráběcích nástrojů

maskaAktivace proměnná nebo konstanta integer. Maska aktivace jednotek pohybu v osách C a B

jednotkapohybuose1 název zařízení druhu jednotka pohybu v ose. (X ve standardu ISO)

jednotkapohybuose2 název zařízení druhu jednotka pohybu v ose. (Y v standardu ISO)

jednotkapohybuose3 název zařízení druhu jednotka pohybu v ose. (Z v standardu ISO)

jednotkapohybuose4 název zařízení druhu jednotka pohybu v ose. (C v standardu ISO)

jednotkapohybuose5 název zařízení druhu jednotka pohybu v ose. (B v standardu ISO)

Popis

Identifikuje tři matrice pro parametrizování stroje a pět zařízení druhu jednotka pohybu v ose, která tvoří samotný stroj. Tato instrukce **musí** být provedena před jakoukoli jinou instrukcí ISO. Parametr **maskaAktivace** definuje, které rotační jednotky pohybu v ose (C a/nebo B) aktivovat.

Ohledně hodnot, které je třeba nastavit, vycházejte z níže uvedené tabulky:

maskaAktivace	Popis
31	Aktivace jednotek pohybu v osách C a B
23	Aktivace samotné jednotky pohybu v ose B
15	Aktivace samotné jednotky pohybu v ose C
7	Zakázání jednotek pohybu v osách C a B

Poznámka

Měrnými jednotkami, ve kterých jsou vyjádřeny v konfiguraci hodnoty rotačních jednotek pohybu v osách, jsou stupně.

Souvislost mezi fyzickými jednotkami pohybu v osách a virtuálními jednotkami pohybu v osách ISO nastavená prostřednictvím této instrukce je zrušena v okamžiku ukončení úlohy, ve které je definována tato instrukce, nebo prostřednictvím instrukce [ISOM2](#). Jednotky pohybu v osách lze použít pro pohyby klasického druhu.

ISOG217

Syntaxe

ISOG2

jednotkapohybuose1, jednotkapohybuose2, jednotkapohybuose3, jednotkapohybuose4, jednotkapohybuose5, VirtualniJednotkapohybuose1, VirtualniJednotkapohybuose2, VirtualniJednotkapohybuose3, VirtualniJednotkapohybuose4, VirtualniJednotkapohybuose5

Argumenty

jednotkapohybuose1	název zařízení druhu jednotka pohybu v ose
jednotkapohybuose2	název zařízení druhu jednotka pohybu v ose
jednotkapohybuose3	název zařízení druhu jednotka pohybu v ose
jednotkapohybuose4	název zařízení druhu jednotka pohybu v ose
jednotkapohybuose5	název zařízení druhu jednotka pohybu v ose
VirtualniJednotkapohybuose1	název zařízení druhu virtuální jednotka pohybu v ose (X ve standardu ISO)
VirtualniJednotkapohybuose2	název zařízení druhu virtuální jednotka pohybu v ose. (Y ve standardu ISO)
VirtualniJednotkapohybuose3	název zařízení druhu virtuální jednotka pohybu v ose (Z ve standardu ISO)
VirtualniJednotkapohybuose4	název zařízení druhu virtuální jednotka pohybu v ose (C ve standardu ISO)
VirtualniJednotkapohybuose5	název zařízení druhu virtuální jednotka pohybu v ose (B ve standardu ISO)

Popis

Popisuje fyzické jednotky pohybu v osách a virtuální jednotky pohybu v osách, které tvoří stroj. Virtuální jednotky pohybu v osách popisují polohu a nasměrování nástroje a musí být deklarovány jako druh Virtuální v konfiguraci softwaru Albatros. Prvních pět jednotek pohybu v osách musí být fyzické jednotky pohybu v osách a jsou ovládány interpolátorem. Následujících pět jednotek musí být virtuální jednotky pohybu v osách a jsou to jednotky pohybu v osách, které používají instrukce [ISOGO](#) a [ISOG1](#).

Tato instrukce **musí** být provedena před jakoukoli jinou instrukcí ISO.

Vzorce pro přímou a obrácenou kinematiku na přechod od jedné polohy v prostoru spojují (fyzické jednotky pohybu v osách) do provozního prostoru (virtuální jednotky pohybu v osách) musí být upřesněny prostřednictvím instrukce [KINEMATICEXPR](#) pro každou z deseti jednotek pohybu v osách definovaných v instrukci ISOG217.

Tato instrukce způsobuje při použití na jednotkách krokového pohybu v ose vznik chyby systému (4105 - Danou instrukci nelze provést na jednotce pohybu `NazevJednotkypohybuose`).

Poznámka

Souvislost mezi fyzickými jednotkami pohybu v osách a virtuálními jednotkami pohybu v osách ISO nastavená prostřednictvím této instrukce je zrušena v okamžiku ukončení úlohy, ve které je definována tato instrukce, nebo při provedení instrukce [ISOM2](#). Jednotky pohybu v osách lze použít pro pohyby klasického druhu.

ISOM2

Syntaxe

ISOM2

jednotkapohybuose

Argumenty

jednotkapohybuose název zařízení druhu jednotka pohybu v ose

Popis

Uvolňuje jednotky pohybu v osách od pohybu ISO, který je nastaven instrukcí [ISOG216](#) nebo instrukcí [ISOG217](#).

ISOM6

Syntaxe

ISOM6

jednotkapohybuose, indexRadkuMatriceRotacnichjednotek, indexRadekMatriceDrzakuNastroju, indexRadekMatriceNastroju

Argumenty

jednotkapohybuose	název zařízení druhu jednotka pohybu v ose
indexRadekMatriceRotacnichjednotek	konstanta nebo proměnná integer. Index řádku matrice rotačních jednotek pohybu v osách
indexRadekMatriceDrzakuNastroju	konstanta nebo proměnná integer. Index řádku matrice držáku nástrojů

indexMatriceNastroju

konstanta nebo proměnná integer. Index řádku matrice nástrojů

Popis

Slouží k nastavení použití skupiny parametrů popisujících kinematiku stroje. **Indexy** se vztahují na tři matrice s názvem určeným uživatelem, deklarované v souboru globálních proměnných softwaru Albatros. Parametr **jednotkapohybuvose** identifikuje příslušný kanál interpolace. Níže uvedené tabulky popisují, jak musí být deklarovány tři matrice v souboru globálních proměnných.

Pole Matrice	Matrice Rotačních Jednotek pohybu v osách
Posun X	Posun podél X mezi otočným bodem a bodem ovládání hlavy
Posun Y	Posun podél Y mezi otočným bodem a bodem ovládání hlavy
Posun Z	Posun podél Z mezi otočným bodem a bodem ovládání hlavy
Posun os X	Odchylka v X mezi osou otáčení a osou natáčení (při poloze jednotky pohybu v ose C =0)
Posun os Y	Odchylka v Y mezi osou otáčení a osou natáčení (při poloze jednotky pohybu v ose C =0)
Posun os Z	Vzdálenost Nos-otočný bod
Úhel δ	Úhel kolem Z pro správné polohování hlavy vůči nule stroje
Úhel γ	Úhel mezi rotační a natáčecí plochou

Pole Matrice	Matrice DržakuNastroju
PosunPU X	Posun v X mezi bodem uchycení držáku nástrojů k motoru a bodem uchycení nástroje k držáku nástrojů (při poloze jednotky pohybu v ose C =0 a svislým motorem)
PosunPU Y	Posun v Y mezi bodem uchycení držáku nástrojů k motoru a bodem uchycení nástroje k držáku nástrojů (při poloze jednotky pohybu v ose C =0 a svislým motorem)
PosunPU Z	Posun v Z mezi bodem uchycení držáku nástrojů k motoru a bodem uchycení nástroje k držáku nástrojů (při poloze jednotky pohybu v ose C =0 a svislým motorem)
Úhel α	Úhel posunu os mezi osou motoru a osou nástroje (vůči Z)
Úhel β	Úhel posunu os mezi osou motoru a osou nástroje (vůči Y)

Pole Matrice	Matrice Nástrojů
Délka nástroje	Délka nástroje

ISOSETPARAM**Syntaxe****ISOSETPARAM****OznaceniParametru, hodnota****Argumenty****OznaceniParametru**

konstanta nebo proměnná integer. Jedná se o číslo, které označuje parametr

hodnota

konstanta nebo proměnná float. Jedná se o hodnotu, která má být nastavena.

Popis

Slouží k nastavení některých parametrů, které charakterizují plynulost interpolovaného pohybu ISO. Níže uvedená tabulka vysvětluje význam každého **OznaceniParametru**, rozmezí hodnot, ve kterém se musí pohybovat proměnná **hodnota** a přednastavené hodnoty.

OznaceniParametru	Rozsah	Přednastavení	Význam
0	0.0-100.0	50.0	Procentuální vyjádření zpomalení na lineárních jednotkách pohybu v osách v případě úhlového bodu. (0=žádné zpomalení, 100=maximální zpomalení dovolené interpolátorem)

1	0.0-100.0	50.0	Procentuální vyjádření zpomalení na rotačních jednotkách pohybu v osách v případě úhlového bodu. (0=žádné zpomalení, 100=maximální zpomalení dovolené interpolátorem)
2	0.5-1.0	0.9	Faktor snížení rychlosti na křivočaré pořadnici v případě úhlového bodu. (1=žádné snížení, 0,5=maximální dovolené snížení)
3	0.0-100.0	60.0	Procentuální vyjádření zpomalení v případě blízkých přetržitostí. (0=žádné zpomalení, 100=maximální zpomalení dovolené interpolátorem)
4	0.0-100.0	10.0	Procentuální vyjádření hodnoty smooth dráhy pohybu
5	(Větší než 0,0)-1,0	0.2	Minimální rozměr úseku, který lze projet se samotnými lineárními jednotkami pohybu v osách. Hodnota je vyjádřena v milimetrech.
6	(Větší než 0,0)-1,0	0.1	Minimální rozměr úseku, který lze projet se samotnými rotačními jednotkami pohybu v osách. Hodnota je vyjádřena ve stupních.
7	0.0-100.0	100.0	Dolní limit filtru vyhlazení
8	1.0-100.0	1.0	Multiplikační faktor, aplikovaný na zrychlení a na zpomalení, která jsou zadaná v konfiguraci. Zvyšuje maximální zrychlení a zpomalení samotných lineárních jednotek pohybu v osách. Hodnoty mimo interval způsobují chybu systému 4399 - Parametr mimo rozsah.
9	1.0-100.0	1.0	Multiplikační faktor, aplikovaný na zrychlení a na zpomalení, která jsou zadaná v konfiguraci. Zvyšuje maximální zrychlení a zpomalení samotných rotačních jednotek pohybu v osách. Hodnoty mimo interval způsobují chybu systému 4399 - Parametr mimo rozsah.


```

KinematicExpr X = Rx + 135 - ut * sin ( Rb ) * cos ( C )
; PŘÍMÁ EXPLICITNÍ KINEMATIKA JEDNOTKY POHYBU V OSE Y
KinematicExpr Y = Ry - posuny - ut * sin ( Rb ) * sin ( C )
; PŘÍMÁ EXPLICITNÍ KINEMATIKA JEDNOTKY POHYBU V OSE Z
KinematicExpr Z = Rz - posunz - ut * cos ( Rb )
; PŘÍMÁ EXPLICITNÍ KINEMATIKA JEDNOTKY POHYBU V OSE C
KinematicExpr C = Rc
; PŘÍMÁ EXPLICITNÍ KINEMATIKA JEDNOTKY POHYBU V OSE B
KinematicExpr B = Rb

; POHYB
ISOG0 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,0.0
ISOG1 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,1000.0
ISOG1 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,1000.0
ISOG1 1003,X 996.0,Y 600.0,Z 0.0,C 90.0,B 45.0,1000.0
WAITSTILL X
ISOM2 X
FRet

```

10.3.21 Instrukce, které již nejsou k dispozici

INPBCD	čte sérii digitálních půlbajtů ve formátu BCD
OUTBCD	mění sérii digitálních půlbajtů ve formátu BCD
SETFORCEDBCD	provádí nucené nastavení série půlbajtů do formátu BCD
CANOPENDRIVER	otevření komunikačního kanálu CANopen
CANCLOSEDRIVER	zavření komunikačního kanálu CANopen
CANRESETBOARD	provedení vynulování karty CANopen
CANSETOBJECT	zápis objektu CANopen
CANGETOBJECT	čtení objektu CANopen
SLMCOMMAND	provedení příkazu SLM
SLMEEPROMDISABLE	provedení příkazu zakázání zápisu do EEPROM
SLMEEPROMENABLE	provedení příkazu aktivace zápisu do EEPROM
SLMGETEEPROM	čtení místa v paměti EEPROM
SLMGETPARAM	čtení parametru SLM
SLMGETREGISTER	čtení registru SLM
SLMGETSTATUS	čtení veličiny pohonu
SLMSEEEPROM	zápis na místo v paměti EEPROM
SLMSETPARAM	nastavení parametru SLM
SLMSETREGISTER	nastavení registru SLM
HOMING	provedení „vyhledání nuly“
SYNCROPEN	otevřít kanál synchronizovaného pohybu
SYNCRCLOSE	zavřít kanál synchronizovaného pohybu
SYNCRMOVE	přiřazuje bod synchronizovaného pohybu
SYNCRSETACC	nastavuje zrychlení pro synchronizované pohyby
SYNCRSETDEC	nastavuje zpomalení pro synchronizované pohyby
SYNCRSETVEL	nastavuje rychlost pro synchronizované pohyby
SYNCRSETFEED	odstupňuje rychlost jednotek pohybu v osách v synchronizovaném pohybu
SYNCRSTARTMOVE	zahájí zpracování synchronizovaného pohybu
GETVF	čte hodnotu měniče napětí/frekvence

10.3.22 Instrukce nepoužitelné při přerušení (interrupt)

Níže uvedené instrukce nemohou být použity ve funkcích, které jsou vyvolávány instrukcemi [ONFLAG](#), [ONINPUT](#) a [ONERRSYS](#). Jejich použití je zakázáno také v [úlohách prováděných v reálném čase \(real-time\)](#).

Instrukce, které vyvolávají funkci přerušení (interrupt):

- ONFLAG
- ONINPUT
- ONERRSYS

Instrukce, které způsobují čekání:

- WAITINPUT
- WAITFLAG
- WAITACC
- WAITCOLL
- WAITDEC
- WAITREG
- WAITTARGET
- WAITWIN
- WAITSTILL
- WAITTASK
- WAITRECEIVE
- WAITPERSISTINPUT
- MULTIWAITFLAG
- MULTIWAITINPUT

Instrukce komunikace.

- SEND
- RECEIVE
- CLEARRECEIVE
- COMOPEN
- COMCLOSE
- COMREAD
- COMREADSTRING
- COMWRITE
- COMWRITESTRING
- COMGETERROR
- COMCLEARRXBUFFER
- COMGETRXCOUNT

Instrukce, které souvisí s pohybem jednotek pohybu v osách:

- MOVINC
- MOVABS
- LINEARINC
- LINEARABS
- CIRCLE
- CIRCINC
- CIRCABS
- HELICINC
- HELICABS
- COORDIN
- MULTIABS
- MULTIINC
- SETRIFLOC
- SETTOLERANCE
- RESRIFLOC
- SETPFLY
- SETPZERO
- SETINDEXINTERP
- STARTINTERP
- FASTREAD
- ENABLE
- DISABLE
- ENDMOV

Instrukce ISO:

- ISOG0
- ISOG1
- ISOG9
- ISOG90
- ISOG91
- ISOG93
- ISOG94
- ISOG216

- ISOG217
- ISOM2
- ISOM6
- ISOSETPARAM
- KINEMATICEXPR

Instrukce Blackbox:

- ENDBLACKBOX
- PAUSEBLACKBOX
- STARTBLACKBOX

Instrukce, které souvisí se správou EtherCAT:

- READDICTIONARY
- WRITEDICTIONARY

Instrukce související s víceúlohovým prováděním (multitasking):

- SENDMAIL
- WAITMAIL
- ENDMAIL
- SENDIPC
- WAITIPC
- TESTMAIL
- TESTIPC

Instrukce, které způsobují dlouhou dobu zpracování:

- SORT
- FIND
- FINDB
- MOVEMAT

10.4 Příklady

10.4.1 Vynulování při Přerušení (Interrupt)

```

;-----
; Příklad rutiny okamžitého vynulování
; Function provede níže uvedené operace:
;
; 1) Nastaví jednotku pohybu v ose zrušením softwarových limitů
;     a vynulováním polohy
; 2) Zkontroluje, zda se senzor již nenachází ve stavu ZAP. (ON).
;     Když se již nachází ve stavu ZAP. (ON), pohne jednotkou pohybu v ose a
;     vyčká na jeho
;     návrat do stavu VYP. (OFF). Kdyby k tomu nedošlo v průběhu 30 sekund,
;     vytvoří chybové hlášení.
; 3) Nastavení rychlost vyhledání senzoru
; 4) Spustí pohyb jednotky pohybu v ose a pro danou
;     jednotku pohybu v ose okamžité vynulování. Při zásahu přerušení
;     bude poloha jednotky pohybu v ose vynulována a bude automaticky
;     zahájen
;     pohyb do polohy mimo kolizní prostor.
; 5) Vyčká na příchod jednotky pohybu v ose do polohy mimo kolizní
;     prostor
; 6) Obnoví limity jednotky pohybu v ose
;
; © TPA
;-----

```

Function Rychlé_vynulování

```

ResLimPos   jednotkapohybuose ; Inicializace jednotky pohybu v ose
ResLimNeg   jednotkapohybuose
SetQuote    jednotkapohybuose,0

IfInput     FastInput,OFF,Goto Continue ; Zkouška obsazení senzoru

```

```

    SetVel      jednotkapohybuvose,5           ; Nastavení
rychlosti
    ; přesunu mimo kolizní
    prostor
    MovAbs      jednotkapohybuvose,30         ; pohyb
jednotky pohybu v ose
    WaitInput   FastInput,OFF,30,Call Error   ; kontrola přesunu mimo
kolizní prostor
    ; mikrospínač,
    ; chyba po TimeOut=30

    EndMov      jednotkapohybuvose           ; Zastavení jednotky
pohybu v ose
    WaitStill   jednotkapohybuvose ; Čekání na Zastavení jednotky pohybu v
ose

Continue:
    SetVel      jednotkapohybuvose,10        ; Rychlost vyhledání senzoru
vynuťování
    MovAbs      jednotkapohybuvose,-1000     ; Pohyb v záporném směru za
účelem vyhledání senzoru
    SetPFly     jednotkapohybuvose,ON,10,0    ; Připojení Přerušeni
    ; a nastavení polohy a rychlosti
    ; přesunu mimo kolizní prostor
    WaitStill   jednotkapohybuvose ; Čekání na Zastavení jednotky pohybu v
ose

    SetLimPos   jednotkapohybuvose ; Obnovení limitů jednotky pohybu v ose
    SetLimNeg   jednotkapohybuvose

    Fret

; podprocedura odeslání chybového hlášení
Error:
    Error      ERR_SETP      ; signalizace chyby zabraňující dalšímu
pokračování
    Ret

```

10.4.2 Server pohybu jednotek pohybu v osách

```

;-----
; Příklad serveru pohybování jednotek pohybu v osách:
;
; Server zajišťuje pohyb jednotek pohybu v osách stroje
; za jednotlivé úkoly.
;
; Úkoly client odesílají příkazy ve formě
; hlášení (mail) do poštovní schránky.
;
; Server přebere příkazy ze schránky a provede je.
;
; Požadavky jsou zařazeny na konec řady ve schránce, a proto v případě,
; že požadavek přijde v době, kdy je již server
; obsazen, nebude ztracena, ale bude vybavena jakmile to bude možné.
;
; Server představuje jediný úkol, kterým je pohybování jednotek pohybu v
osách. Jedná se o způsob, jak se vyhnout
; konfliktům.
;
; Server je implementován funkcí Master_assi.
;
; Příklad client je implementován funkcí Check_flag.
; Tato funkce pravidelně kontroluje stav
; příznaku a když jej najde ve stavu ZAP. (ON), odešle na server

```

```

; příkaz pro provedení vynulování jednotek pohybu v osách.
; příznak bude pravděpodobně nastaven na ZAP. (ON) manuálně
; obsluhou, například s použitím grafického zobrazení.
;
;-----
;-----
; -- GLOBÁLNÍ KONSTANTY STROJE --
;-----
Const MBOX = 101 ; identifikuje poštovní schránku
; příkazy
Const SETP = 10 ; vynulování jednotek pohybu v osách
Const CHG = 11 ; výměna nástroje
Const OTVOR = 12 ; provedení vrtání

;-----
; --- SKUPINA JEDNOTEK POHYBU V OSÁCH ---
;-----

; definice chybových hlášení
Defmsg ERR_CMD "Neznámý příkaz skupiny jednotek pohybu v osách"

; --- Server ---
Function Autorun master_jednotekpohybuvosach

    local cmd as integer ; příkaz
    local quota_X as double ; poloha X otvoru
    local quota_Y as double ; poloha Y otvoru

loop:
    waitmail MBOX,cmd,poloha_X,poloha_Y ; čekání na příkaz

    ; Po přijetí příkazu jej identifikují
    ; a provedou vhodný úkon
    select cmd

    case SETP
        fcall vynulovani_jednotekpohybuvosach ; vynulování jednotek
pohybu v osách
    case CHG
        fcall vymena_nastroje ; Provede výměnu nástroje
    case OTVOR
        fcall vrtani poloha_X,poloha_Y ; vrtání po
; určené polohy
    case else
        call chyba
    endselect

    endmail MBOX ; oznámení provedení příkazu
    goto loop ; vrátí se do stavu čekání na
nový ; příkaz

    fret

; podprocedura odeslání chybového hlášení
chyba:
    error ERR_CMD
    ret

```

```

;-----
; --- VŠEOBECNÁ ČÁST TÝKAJÍCÍ SE SKUPINY ---
;-----
; --- Client ---
Function Check_flag
Loop:
    ifflag Setp_jednotekpohybuvosach,OFF, goto loop ; text stavu příznaku
; OK, příznak má hodnotu ZAP. (ON), odešlu příkaz
sendmail MBOX,WAITTACK,SETP,0.0,0.0
resetflag Setp_jednotekpohybuvosach ; vynuluji příznak
goto loop ; vrátím se do stavu čekání
fret
; VŠIMNĚTE SI, ŽE:
; - Po příkazu "SETP" je třeba uvést
; dva parametry „poloha_X“ a „poloha_Y“, i když to
; nemá smysl pro operaci vynulování.
; Server totiž nemůže předem vědět, který příkaz
; mu bude odeslán, a proto musíme uvést dvě hodnoty
; stejného typu jako ty, jejichž přijetí server očekává
; v tomto případě dvě DOUBLE. Předané hodnoty jsou „0,0“ a "0,0".
; - Parametr „WAITACK“ zajišťuje, že client zůstane ve stavu čekání
; na provedení příkazu ze strany Serveru.
; Client bude pokračovat ve svém provádění teprve poté, co Server
; provedl ENDMAIL nebo zahájil zpracování nového
; příkazu (WAITMAIL).

```

10.4.3 Cyklus hlavní části kódu (Main) se správou chyb

```

;-----
; Hypotetická hlavní funkce
; slouží k inicializaci stroje a k provedení kontrolního cyklu
;-----
Function Main
    OnErrSys GestErrSys ; aktivuje správu chyb
    StartTask NouzovaZastaveni ; inicializuje
    StartTask RidiciZarizeni
    AktivovatJednotkypohybuvosach
Loop:
    IfFlag Flag,OFF, ZruseniNouzovehoStavu
    .....
    goto loop
Fret
;-----
; funkce správy chyb
;-----
Function GestErrSys
    Param nerrere as integer
    Param task as function
    Param tipodevice as device

```

```

EndTask      RidiciZarizeni          ; Ukončení úkolu
If           nerrorex, >, 5, goto noerraxis ; RidiciZarizeni
                                                ; Prvních 5 chyb
                                                ; se vztahuje
                                                ; na jednotky pohybu

v osách
ResetFlag   Flag
ZrusitJednotkypohybuvosach

noerraxis:
Fret

```

10.4.4 Operace na řetězcích

```

;-----;
; Příklad manipulace s řetězcí
;-----;
Function prikklad
Local      retezec1 as string
Local      retezec2 as string
Local      retezec3 as string
Local      delka as integer
Local      poloha as integer

SetString  "Zkusebni",retezec1          ; retezec1 obsahuje
                                         ; nyní "Zkusebni"

SetString  " retezec",retezec2

AddString  retezec1,retezec2,retezec3 ; retezec3 obsahuje
                                         ; "Zkusebni retezec"

Search     retezec3,'k',poloha          ; poloha má hodnotu 2
Search     retezec3,'y',poloha          ; poloha má hodnotu -1

Left       retezec3,7,retezec1          ; retezec1
                                         ; obsahuje "Zkusebni"

Right      retezec3,2,retezec2          ; retezec2
                                         ; obsahuje "ec"

Mid        retezec3,9,2,retezec3        ; retezec3
                                         ; obsahuje "re"

ControlChar 65,retezec1                 ; retezec1
                                         ; obsahuje "Z"

Len        retezec3,delka                ; delka má hodnotu 2

Str        delka,retezec3                ; retezec3
                                         ; obsahuje "2"

Val        poloha,retezec1               ; retezec1
                                         ; obsahuje "-1"

AddString  "Vysledek je ",retezec1,retezec2
; retezec2 obsahuje "Vysledek je -1"

Fret

```

10.4.5 Sekvenční / Paralelní Provedení

```

-----
; Příklad rutiny, která řídí vynulování
; stroje se 3 jednotkami pohybu v ose
; se zabráněním případným mechanickým kolizím.
;
; vynulování jednotlivých jednotek pohybu v ose
; jsou implementována funkcemi, jejichž text je vynechán.
; Viz příklad „Rutina vynulování“.
;
; Nejdříve je vykonáno samotné vynulování
; jednotky pohybu v ose Z (vycházíme přitom z předpokladu,
; že nemůže být vykonáno spolu s ostatními),
; po jehož ukončení budou současně provedena
; vynulování jednotek pohybu v osách
; X a Y.
-----

```

```

; hlášení pro obsluhu (přeloženo do jazyka)
DefMsg      MSG_SETP      ITA      "Probíhá vynulování..."
           MSG_SETP      ENG      "Homing in progress ..."

```

Function vynulovaniJednotkypohybuvosach

```

Message      MSG_SETP      ; informuje obsluhu

Fcall      vynulovaniJednotkypohybuoseZ      ; vynulování jednotky
pohybu v ose Z

; OK, vynulování jednotky pohybu v ose Z bylo ukončeno

StartTask    vynulovaniJednotkypohybuoseX      ; spustí vynulování X a Y
StartTask    vynulovaniJednotkypohybuoseY

WaitTask     vynulovaniJednotkypohybuoseX      ; čeká na dokončení
WaitTask     vynulovaniJednotkypohybuoseY

DelMessage   MSG_SETP      ; vymaže hlášení
              ; pro obsluhu

```

Fret

10.4.6 Nulovací Rutina

```

-----
; Příklad Nulovací rutiny jednotky pohybu v ose
;
; Function provede níže uvedené operace:
; 1) zruší softwarové limity jednotky pohybu v ose
; 2) nastaví rychlost vyhledání spínače
; 3) pohybuje jednotkou pohybu v ose do inkrementální polohy, která
;    zaručuje dosažení spínače
; 4) čeká, dokud jednotka pohybu v ose neaktivuje spínač
; 5) zastaví jednotku pohybu v ose a vyčká na konec pohybu
; 6) nastaví rychlost (nízkou) uvolnění spínače
; 7) provede takový zpětný pohyb jednotky pohybu v ose, který stačí
;    na uvolnění spínače
; 8) vyčká na uvolnění spínače
; 9) přiřadí novou polohu jednotce pohybu v ose
; 10) obnoví přednastavenou rychlost a softwarové limity
;
; © TPA.
-----

```

Function Vynuťování

```

ResLimPos   jednotkapohybu v ose      ; zruší softwarové limity
ResLimNeg   jednotkapohybu v ose

SetVel      jednotkapohybu v ose,10   ; nastaví rychlost

MovInc      jednotkapohybu v ose,10000 ; pohybuje jednotkou pohybu v ose
ose

WaitInput   spínač,ON ; čekání na spínač

EndMov      jednotkapohybu v ose      ; zastaví jednotku pohybu v ose
ose
waitStill   jednotkapohybu v ose      ; čeká na zastavení jednotky
pohybu v ose

SetVel      jednotkapohybu v ose, 0.1 ; nastaví rychlost uvolnění

MovInc      jednotkapohybu v ose,-100 ; pohybuje jednotkou pohybu v ose
ose

WaitInput   spínač,OFF ; čekání na uvolnění spínače

SetQuote    jednotkapohybu v ose,0    ; přiřadí novou polohu

SetVel      jednotkapohybu v ose      ; obnovení rychlosti
SetLimpos   jednotkapohybu v ose      ; obnovení softwarových limitů
SetLimneg   jednotkapohybu v ose

```

Fret

10.4.7 Pohyby ISO

```

-----
; Příklad pohybu ISO
;
; Bude vygenerován profil prostřednictvím instrukcí ISG00 a ISOG1
;
; © TPA.
;-----*

; Deklarování matic ISO
; Matrice rotačních jednotek pohybu v osách
MxRot[5] as double:off_X double:off_Y double:off_Z double:dis_X
double:dis_Y double:dis_Z double:delta double:gamma
; Matrice držáku nástrojů
MxDrzak[1] as double:off_X double:off_Y double:off_Z double:alpha
double:beta
; Matrice nástrojů
MxNastroje[10] as double:ut double

```

Function InterpolaceISO

```

; nastavení standardních hodnot parametrizace stroje
setval 90.0 MxRot[5].rada
setval 260.3 MxNastroje[10].ut
setval MxNastroje[10].ut ut

; nastavení parametrů algoritmu
IsoiParam 0 50
IsoiParam 1 50

```

```

IsosetError 2 0.9
IsosetError 3 60
IsosetError 4 30

; nastavení stroje: prohlášení tří matic
; použitých pro parametrizaci stroje
; a fyzických jednotek pohybu v ose v pohybech ISO
isog216 MxRot MxDrzak MxNastroje 31 X Y Z C B ; IMPLICITNÍ KINEMATIKA

; nastavení skupiny parametrů, které popisují
; kinematiku stroje
isom6 X 5 1 10 ; IMPLICITNÍ KINEMATIKA

; nastavení počátečních hodnot
setquote x 500
setquote y 300
setquote z 0
setquote c 0
setquote b 0
setvel x
setvel y
setvel z
setvel c
setvel b
setvel i x y z c b

; provedení profilu
isoG0 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,50.0
isoG1 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,10000.0
isoG1 1003,X 996.0,Y 600.0,Z 0.0,C 90.0,B 45.0,10000.0
isoG1 1002,X 600.0,Y 600.0,Z 0.0,C 90.0,B 45.0,10000.0
isoG1 1004,X 599.131759111665,Y 599.924038765061,Z 0,C 100,B
45.0,10000.0
isoG1 1006,X 598.289899283372,Y 599.69846310393,Z 0,C 110,B 45.0,10000.0
isoG1 1005,X 597.5,Y 599.330127018922,Z 0,C 120,B 45.0,10000.0
isoG1 1003,X 596.786061951567,Y 598.830222215595,Z 0,C 130,B
45.0,10000.0
isoG1 1002,X 596.169777784405,Y 598.213938048433,Z 0,C 140,B
45.0,10000.0
isoG1 1012,X 595.669872981078,Y 597.5,Z 0,C 150,B 45.0,10000.0
isoG1 1011,X 595.301536896071,Y 596.710100716628,Z 0,C 160,B
45.0,10000.0
isoG1 1031,X 595.075961234939,Y 595.868240888335,Z 0,C 170,B
45.0,10000.0
isoG1 1102,X 595.0,Y 0.0,Z 0.0,C 180.0,B 45.0,10000.0
waitstill X Y Z C B
fret

```




Tecnologie e Prodotti per l'Automazione s.r.l.

Via Carducci 221
I - 20099 Sesto S.Giovanni
(MI)
Tel. +39 02.36527550
Fax. +39 02.2481008
www.tpaspa.com