

8

8. ROZHRANÍ CNC SYSTÉM - PLC PROGRAM

Komunikační rozhraní CNC systém - PLC program tvoří dva bloky:

POVELOVÝ BLOK	povel z CNC systému do PLC programu (bude označován PBxx)
BLOK ZPĚTNÉHO HLÁŠENÍ	zpětné hlášení z PLC programu do CNC systému (bude označováno BZHxx)

Položky uvedené v tabulce komunikačního pole se nedeklarují a jsou programátorovi přístupné ze všech modulů PLC programu. Pro povelový blok a blok zpětného hlášení je pomocí komunikace zabezpečen přenos mezi panelem a kazetou systému. PLC program je umístěn v kazetě systému a tak může mít teoreticky navíc přístupné všechny symboly, které jsou přístupné v kazetě systému. Kromě povelového bloku (PBxx) a bloku zpětného hlášení (BZHxx) jsou takto programátorovi přístupné některé proměnné, které jsou definovány v modulu **INCLEXT** typu "INCLUDE" jako "EXTERNAL". Tento modul se připojí k překladu PLC programu a kromě symbolů obsahuje také definici maker, struktur a záznamů. Do tohoto modulu je umožněno přidávat další symboly.

8.1 Modul "INCLEXT" (EXT_04)

V modulu "INCLEXT" jsou kromě proměnných deklarovány také struktury pro sady parametrů regulátorů, některá makra a struktury. Do tohoto modulu je umožněno přidávat další odkazy na proměnné pomocí direktivy **EXTRN** (ASM86) nebo si definovat vlastní makra napsaná pomocí jazyka INTEL-ASM86, nebo BORLAND-TASM. Doporučení pro další proměnné poskytne výrobce systému. Při použití překladače od firmy INTEL má modul INCLEXT příponu **".A86"** a při použití překladače od firmy BORLAND musí mít modul INCLEXT příponu **".ASH"** (viz kapitolu "Překlad PLC programu , překladač TECHNOL"). Pro procesor CPU04 (80486DX) se mění název modulu INCLEXT.ASH na název **EXT_04.ASH**.

Při použití překladače od firmy BORLAND a při nutnosti přidat další odkazy na proměnné, se nedoporučuje měnit modul INCLEXT (EXT_04), ale je možno je přidat přímo do modulu PLC programu do oblasti deklarace dat.

V případě, že požadujeme aby jazyk PLC836 rozpoznal typ proměnné deklarované v modulu INCLEXT, je potřeba pro každou definici EXTRN použít definici makra podle následujícího příkladu. Makro definuje pro proměnnou dočasný symbol se stejným názvem, kterému je přiřazena hodnota:

- ♦ 0 pro konstanty
- ♦ 1 pro proměnné typu BYTE
- ♦ 2 pro proměnné typu WORD

Příklad:

Zpřístupnění konstanty KONST, byteové buňky BUNKA_BYTE a wordové buňky BUNKA_WORD.

V modulu INCLEXT.A86 (překláč INTEL):

```
EXTRN KONST:ABS
%SET(KONST,0)

EXTRN BUNKA_BYTE:BYTE
%SET(BUNKA_BYTE,1)

EXTRN BUNKA_WORD:WORD
%SET(BUNKA_WORD,2)
```

v modulu INCLEXT.ASH, EXT_04.ASH (BORLAND):

```
EXTRN KONST:ABS
_?KONST = 0

EXTRN BUNKA_BYTE:BYTE
_?BUNKA_BYTE = 1

EXTRN BUNKA_WORD:WORD
_?BUNKA_WORD = 2
```

V PLC programu:

```
LOD   KONST           ;načte konstantu do DR

LOD   BUNKA_BYTE      ;načte byteovou buňku do DR

LOD   BUNKA_WORD      ;načte wordovou buňku do DR
```

V případě, že je v modulu INCLEXT.A86 (EXT_04.ASH) definován jen samotný EXTRN pro proměnnou, je nutno v jazyku PLC836 použít deklarování typu (.BYTE .WORD .DWRD .CNST):

Příklad:

Zpřístupnění strojní konstanty BUKON60 pomocí modulu INCLEXT pro PLC program.

V modulu INCLEXT.A86:

```
EXTRN BUKON60:DWORD
```

V PLC programu:

```
LOD   WORD.BUKON60    ;načte WORD do DR
LOD   DWRD.BUKON60    ;načte DWORD do DR
```

8.2 Proměnné deklarované v modulu "INCLEXT" (EXT_04)

Některé důležité proměnné, které jsou v tomto modulu deklarovány.

8.2.1 Strojní konstanty pro PLC

BUKONxy Strojní konstanta xy

Strojní konstanty 90 až 94 a 170 až 229 jsou nastavovatelné v režimu strojních konstant a jsou vyhrazeny pro použití v PLC programu. V systému mají název **BUKONxx**, jsou typu DWORD a hodnota v nich je v BCD tvaru. Jednu konstantu je možno s výhodou použít pro nastavení dvou buněk typu WORD. Převod na binární hodnotu je možno provést jednorázově v modulu PIS_INIT a PIS_CLEAR (viz Struktura PLC programu). Konstanty 171 až 229 jsou přístupné relativně (pokud se nepřidají jejich definice pomocí příkazu EXTRN). Konstantu "n" načteme příkazem: LOD WORD.(BUKON170+4n).

BUKON90	typ WORD v BCD kódu
BUKON91	typ WORD v BCD kódu
BUKON92	typ WORD v BCD kódu
BUKON93	typ WORD v BCD kódu
BUKON94	typ WORD v BCD kódu
BUKON170	typ WORD v BCD kódu

Příklad:

Nastavení doby mazání souřadnice X podle konstanty 172 a doby mazání souřadnice Y podle konstanty 173 v modulu PIS_CLEAR :

LOD	WORD.(BUKON170+8)	;načte konstantu 172 do DR
BIN		;převod do bináru
STO	DOBA_MAZ_X	;zápis do konfigurovatelné proměnné
LOD	WORD.(BUKON170+12)	;načte konstantu 173 do DR
BIN		;převod do bináru
STO	DOBA_MAZ_Y	;zápis do konfigurovatelné proměnné

8.2.2 Odměřování a difference

B_POL Buffer polohy
B_INK Buffer inkrementu
DIFCIT_x Diferenční čítač

PLC program má k dispozici odměřování systému pro zjištění okamžité polohy stroje. Může jej využít například pro řízení mazání souřadnic od polohy stroje. Odměřování je součet **bufferu polohy B_POL** a **bufferu inkrementu B_INK**. Oba buffery jsou typu DWORD v doplňkovém kódu a odměřování je v osminách mikronu. Odměřování je platné od najetí do reference. Přístup k jednotlivým osám je relativní.

PLC program má k dispozici **diferenční čítače DIFCIT_X** pro všechny souřadnice. Pomocí diferenčních čítačů může například hlídat přetečení diferenční odchylky. Diferenční čítače DIFCIT_X pro jednotlivé souřadnice jsou typu DWORD, ale hodnota je v mikronech.

B_POL	typ DWORD buffer polohy (pro 6 os)	- doplňkový kód v osminách mikronu
B_INK	typ DWORD buffer inkrementu (pro 6 os)	- doplňkový kód v osminách mikronu
DIFCIT_X	typ DWORD diferenční čítač (6 os)	- doplňkový kód v mikronech

Příklad:

Hlídní přetečení difference v ose Y podle limitu LIMIT_Y

CLI		;zákaz přerušení (ASM86)
LOD	DWRD.(DIFCIT_X+4)	;načte diferenci Y
STI		;povolení přerušení (ASM86)
ABS	DWRD	;absolutní hodnota DR 32 bitů
GE	DWRD.LIMIT_Y	;porovnání
JL1	ERROR_Y	;skok na chybu

Příklad:

Zjištění polohy v ose Y (- jen kladné míry)

CLI		;zákaz přerušení (ASM86)
LOD	DWRD.(B_INK+4)	;načte buffer inkrementu Y
AD	DWRD.(B_POL+4)	;načte buffer polohy Y
STI		;povolení přerušení (ASM86)
RR	CNST.3,DWRD	;převod na mikrony
STO	DWRD.POLOHA_Y	;zápis do polohy Y

Příklad mazání podle ujeté dráhy je v kapitole "Užitečné příklady."

8.2.3 Otáčky vřetene s ohledem na %S

VYSOVERS

....Otáčky vřetene

Programované otáčky vřetene se PLC program dozví pomocí povelového bloku PB11 a PB12 (viz Povelový blok). Programované otáčky, které jsou přepočteny s ohledem na %S jsou přístupné v buňce **VYSOVERS** typu WORD.

VYSOVERS typ WORD otáčky vřetene

Podrobněji se o problematice vřeten a jejich řízení píše v kapitole "Zadávání otáček vřetena - Otáčky vřetene řízené pomocí %S.

8.2.4 Vstupy z panelu systému CNC836

TLINVstupy z panelu na standardní desce
PxINVstupy z panelu na přídatné desce

V panelu systému může být kromě standardní desky vstupů a výstupů ještě přídatná deska naadresovaná na ADDRPORT_IO1, která podle způsobu použití může obsahovat 4 vstupní porty. Standardní deska pro snímání matice tlačítek má k dispozici také jeden vstupní port. Blíže je problematika přídatných vstupů vysvětlena v kapitole "Přídatné vstupy a výstupy v panelu systému". V modulu INCLEXT.A86 jsou zpřístupněny pro snímání vstupů z panelu tyto symboly:

TLIN	typ BYTE na desce tlačítek (ADDRPORT_TLAC+1)	
P1IN	typ BYTE na desce INOUT02 (ADDRPORT_IO1),	nebo INOUT07 (3)
P2IN	typ BYTE na desce INOUT02 (ADDRPORT_IO1+1),	nebo INOUT07 (3)
P3IN	typ BYTE na desce INOUT02 (ADDRPORT_IO1+2),	nebo INOUT07 (3)
P4IN	typ BYTE na desce INOUT02 (ADDRPORT_IO1+3),	nebo INOUT07 (3)
P5IN až P8IN	typ BYTE na desce INOUT07 (4)	

8.2.5 Rozšíření tlačítek systému o panel stroje

PANEL_STROJEMatice tlačítek panelu stroje
--------------	-----------------------------------

Na desku INOUT02 (ADDRPORT_TLAC) pro snímání matice tlačítek systému v panelu systému nebo na jednotku INOUT07 (2), je možno připojit panel stroje se 64 tlačítky (pro INOUT07 jen 56 tlačítek) zapojených do matice. PLC program má přístup k matici panelu stroje 8 BYTE přes návěští **PANEL_STROJE**. Blíže je problematika panelu stroje popsána v kapitole "Panel stroje zapojen jako rozšíření tlačítek systému".

PANEL_STROJE	typ 8 BYTE matice tlačítek panelu stroje
--------------	--

8.2.6 Režim a tlačítka systému

MATTLTlačítko matice systému
REZIMPIRežim systému

PLC program může sejmout tlačítko matice systému pomocí buňky **MATTL**. Podrobně je problematika snímání tlačítek napsána v kapitole "Snímání tlačítek z panelu systému CNC836 do PLC".

MATTL	typ BYTE	tlačítko matice systému
-------	----------	-------------------------

PLC program dostává informaci o právě navoleném režimu pomocí buňky **REZIMPI**, která je definována v modulu INCLEXT.A86. Buňka REZIMPI je průběžně aktualizována.

REZIMPI	typ BYTE	režim
---------	----------	-------

Buňka REZIMPI nabývá hodnot podle navoleného režimu, například:

- ◆ 1Fh Centrální anulace
- ◆ 14h AUT
- ◆ 17h MAN+
- ◆ 19h MAN-
- ◆ 1BhJOG+
- ◆ 1DhJOG-
- ◆ 13h TOC
- ◆ 0Fh RUP
- ◆ 15h REF

8.2.7 Zadávání externí rychlosti v pomocných ručních pojezdech

EXT_FEED_AUTMAN ...Externí řízení rychlosti

Buňka EXT_FEED_AUTMAN slouží pro zadání externí rychlosti v pomocných ručních pojezdech z PLC programu. Takto zadaná rychlost se projeví, jen když je bit REQ_EXT_FEED_AUTMAN nastaven na hodnotu log.1. Hodnota rychlosti se zadává v binárním kódu v mm/min nebo v $\mu\text{m}/\text{ot}$. Podrobně je tato problematika popsána v kapitole "Pomocné ruční pojezdy".

8.2.8 Pomocné proměnné

Dále uvedeme některé pomocné proměnné definované v modulu INCLEXT.A86. Návrhář PLC programu s těmito buňkami nepříjde do styku, protože je většinou využívají přímo instrukce jazyka PLC836.

- Příznak přepnutí osy na vřetenno

VRETENO_X typ BYTE příznak pro osu X
VRETENO_Y typ BYTE
.....
VRETENO_6 typ BYTE

- Aktuální porty pro vřetenno

AKTUAL_PORT_VR typ DWORD pro obyčejné vřetenno
AKTUAL_PORT_VRAX typ DWORD pro rotační souřadnici

- Break PLC programu

AKTUAL_LINE linka programu, na které je zastavený interfejs ve stavu "break" (viz kapitolu o ladění interfejsu).

- Řízení mikroposuvu u potenciometrů

Při řízení posuvu pomocí potenciometrů jsou pomocí strojních konstant 68 a 69 zadány dvě rychlosti, které odpovídají maximálnímu vytočení potenciometru. PLC program pomocí buňky s bitovou strukturou **BUN_JOGMIK** řídí, která rychlost má být aktuální. Podrobně je tato problematika popsána v kapitole "Řízení posuvu pomocí potenciometrů".

BUN_JOGMIK typ BYTE s bitovou strukturou

- Snímání potenciometrů z panelu

Pro případ řízení pohybu bez polohové vazby pomocí potenciometrů snímaných v panelu systému CNC836 je potřeba snímat hodnotu maximálně šesti řídicích potenciometrů. PLC program může získat tuto informaci pomocí 6 buněk typu BYTE přístupných od adresy **BUF_POTENC**. Podrobně je tato problematika popsána v kapitole "Řízení posuvu pomocí potenciometrů".

8.3 Důležité bitové proměnné pro PLC program

Dále jsou k dispozici tyto důležité bitové proměnné:

Speciální bitové proměnné					
	název	Bit 7.	bit 6.	bit 5.	bit 4.
		Bit 3.	bit 2.	bit 1.	bit 0.
	BZH08PI	MP4PI	MPZPI	MP6PI MPYPI	MP5PI MPXPI
	PB20PI	PO_OS4PI	PO_OSZPI	PO_OS6PI PO_OSYPI	PO_OS5PI PO_OSXPI
	PB20PIS	SM_PO4PI	SM_POZPI	SM_PO6PI SM_POYPI	SM_PO5PI SM_POXPI
	VAZBAPI	VAZBA_4	VAZBA_Z	VAZBA_6 VAZBA_Y	VAZBA_5 VAZBA_X
	POLOHV	POLOHV_4	POLOHV_Z	POLOHV_6 POLOHV_Y	POLOHV_5 POLOHV_X
	INDIKPI	INDIK_4	INDIK_Z	INDIK_6 INDIK_Y	INDIK_5 INDIK_X
		VR_AVP	VR_M01	VR_ND VR_BB	VR_LOM VR_AUT
	BZH08MAN	EN_AUTMAN MP4MAN	MPZMAN	MP6MAN MPYMAN	MP5MAN MPXMAN
	CONT_AUTMAN	AUTMAN_CONT_SEL.. AM_40	AUTMAN_CONT_G00 AM_Z0	AM_60 AM_Y0	AM_50 AM_X0
	CONT_AUTMAN2	AUTMAN_CONT_TOC AM_40	AUTMAN_CONT_NC AM_Z1	AM_61 AM_Y1	AM_51 AM_X1
	EXT_CONT.. AUTMAN	REQ_EXT_SELECT_AUT.. EXM_40	REQ_EXT_G00_AUTMAN EXM_Z0	EXM_60 EXM_Y0	EXM_50 EXM_X0
	EXT_CONT.. AUTMAN2	EXM_41	EXM_Z1	EXM_61 EXM_Y1	EXM_51 EXM_X1
	EXT_CONT.. AUTMAN3	INPOS_STOP	G95_AUTMAN	REQ_EXT_FEED_AUT..	REQ_EXT_CONT_AUT..

8.3.1 Povolení pohybu od PLC (BZH08PI)

MPxPI

....Povolení pohybu

PLC program může povolit nebo zakázat pohyb pro jednotlivé osy bity **MPxPI** v **BZH08PI**.

- ♦ log. 1 povolení pohybu z PLC
- ♦ log. 0 zákaz pohybu z PLC

Implicitně jsou bity MPxPI nastaveny na hodnotu log. 1, takže je pohyb povolen. Implicitní nastavení se provede v rámci instrukce PIS_INIT a PIS_CLEAR.

V bloku zpětného hlášení jsou bity MPx, které jsou výsledkem všech podmínek povolení pohybu. Systémový software zapisuje do bitu MPx výsledek součinu povolení pohybu od různých zdrojů. Do tohoto součinu je zařazeno povolení pohybu např. od modulu interpolace, od modulu reálného času - kde jsou zařazené i koncové spínače (i softwarové) a od modulu servosmyčky. Do součinu jsou zahrnuty i bity MPxPI z BZH08PI, které umožňují blokovat pohyb z uživatelského PLC programu. Supervizor interfejsu umožní pohyb na základě nastaveného bitu MPx do log.1 a na základě ukončení průchodu modulem přípravných funkcí.

PLC program může zakázat pohyb i počas pohybu. V tomto případě se pohyb zastaví dojezdovou rampou. Při nastavení log.1 do MPxPI bude pohyb povolen a souřadnice se rozjede rozjezdovou rampou.

Skutečné povolení pohybu nastane v případě požadavku na pohyb PO_OSxPI (viz dále), nastavením log.1 v MPxPI a celým průchodem modulu přípravné funkce (viz kapitola "Struktura PLC programu"). Pro případ, že PLC program nepotřebuje blokovat pohyb od poruch (blokování pohybu od průchodu PLC programu přípravnými a závěrečnými funkcemi zabezpečuje supervizor interfejsu), nemusí být bity MPxPI v PLC programu vůbec nastavovány, protože mají implicitní hodnotu log.1. Ovládání bitů MPxPI se doporučuje prakticky jen v případě poruch souřadnic, které vznikly počas pohybu.

Příklad:

Zablokujte pohyb v ose Z od poruchy ERROR_Z .

LDR	ERROR_Z	;načte poruchový signál
FLI	0,MPYPI	;podmíněný zákaz pohybu

8.3.2 Pohyb v osách (PB20PI)

PO_OSxPIPohyb v osách

Požadavek na pohyb v osách. Pro jednotlivé souřadnice je v bitech **PO_OSxPI** v **PB20PI**.

- ♦ log. 1 požadavek na pohyb
- ♦ log. 0 není požadavek na pohyb

Bit **PO_OSxPI** je součtem všech požadavků na pohyb. Na rozdíl od **PO_OSx** v povelovém bloku **PB20** (viz dále) obsahuje i pohyb od řízení potenciometrů, který zadal PLC program. Po skončení pohybu se bit **PO_OSxPI** vynuluje.

Bit **PO_OSxPI** možno jenom číst v případě řízení pohybu z CNC systému (programově nebo ručně) nebo do něj zapisovat požadavek na pohyb z PLC programu (případ potenciometrů = viz kapitolu "Řízení pohybu pomocí potenciometrů").

Příklad:

Nastartujte mechanismus uvolnění osy X - **UVOL_X** v přípravných funkcích, když je pohyb programován

```

LDR PO_OSXPI      ;načte požadavek na pohyb
JL0 NENI_POX      ;obskok, když není pohyb
FL 1,UVOL_X       ;start mechanismu uvolnění osy X
EX                ;
LDR UVOL_X        ;čekání na uvolnění
EX1               ;
NENI_POX:         ;

```

8.3.3 Směr pohybu (PB20PIS)

SM_POxPISměr pohybu

Směr pohybu pro jednotlivé souřadnice je v bitech **SM_POxPI** v **PB20PIS**.

- ♦ log. 0 kladný směr pohybu
- ♦ log. 1 záporný směr pohybu

Směr pohybu je platný až po skutečném rozjetí pohybu, to znamená až po průchodu modulem přípravných funkcí. Pro případ, že je potřeba řídit nějakou činnost na základě směru pohybu v přípravných funkcích (například sepnutí směrových spojek), je potřeba spustit mechanismus od průchodu přípravných funkcí a tak se zabezpečí jeho paralelní chod s pohybem. Tento případ je vysvětlen na následujícím příkladu:

Příklad:

Ovládání směrových spojek stroje pro osu Y:

Konec modulu PRIPRAVNE_FUNKCE:

```

                LDR    PO_OSYPI      ;načtení požadavku na pohyb
                FLI     1,SPOJKA_Y   ;podmíněná aktivace mechanismu SPOJKA_Y
PRIPRAVNE_FUNKCE_END      ;konec přípravných funkcí - nečekáme na dokončení
                        ;mechanismu SPOJKA_Y

```

V modulu PROVOZ je definován mechanismus:

```

MECH_BEGIN  SPOJKA_Y
            EX                      ;začátek pohybu na 1 průchod PLC programu
            FL      0,MPYPI        ;zákaz pohybu
            LDR     SM_POYPI       ;načte platný směr pohybu
            WR      SPOJ_MINUS_Y_O ;nastaví zápornou spojku log.1=minus
            CA                      ;negace RLO
            WR      SPOJ_PLUS_Y_O  ;nastaví kladnou spojku log.1=plus
            LDR     KSOJ_Y_I       ;kontrolní vstup
            EX0                  ;čekání na sepnutí spojky
            FL      1,MPYPI        ;povolení pohybu
MECH_END    SPOJKA_Y              ;konec mechanismu

```

8.3.4 Vypínání a zapínání polohové vazby

VAZBA_x Polohová vazba

Vypínání a zapínání polohové vazby se řídí bitem **VAZBA_x** v **VAZBAPI**.

- ♦ log. 0 vypnutí polohové vazby
- ♦ log. 1 zapnutí polohové vazby

Systém CNC836 má softwarovou polohovou vazbu. PLC program může polohovou vazbu nejen zapínat a vypínat, ale může také nastavovat parametry polohové servosmyčky. Podrobně je tato problematika rozepsána v kapitole "Nastavení parametrů servopohonů a jejich řízení PLC programem".

Implicitně jsou bity **VAZBA_x** nastaveny do log.1, to znamená že je polohová vazba zapnuta. Implicitní nastavení se provede v rámci instrukce **PIS_INIT** a **PIS_CLEAR**. Pokud PLC program nepotřebuje polohovou vazbu vypínat, nemusí bity **VAZBA_x** vůbec obsluhovat.

Vypnutí polohové vazby by se mělo provádět, když je souřadnice v klidu. Nastavením log.0 do příslušného bitu **VAZBA_x** se odpojí účinek diferenčního čítače na vysílání napětí pro pohon. **Výstupní napětí pro pohon se neobsluhuje. Odměřování souřadnice zůstává dál plně v činnosti.** Pokud nebudou použity ještě jiné prostředky (například přepnutí do indikace nebo nulování difference), není v tomto případě vhodné, aby souřadnice jezdila. Odměřování je stále v činnosti a ujetá dráha by se kumulovala v diferenčním čítači. Po čase by mohlo dojít k jeho přetečení. Při opětovném zapnutí polohové vazby by pohon najednou dostal velký napěťový skok, protože polohová vazba by se snažila vynulovat diferenční odchylku.

Vypnutí polohové vazby je výhodné použít například pro upínání souřadnic, pro řízení souřadnic mimo CNC systém a pro případ, když je víc souřadnic řízeno stejným pohonem.

Ve zvláštních případech je možno použít ve spojitosti s vypínáním vazby nulování diferenčního čítače přepisem do odměřování systému. Příklad na nulování difference je v kapitole "Užitečné příklady".

Případ přepínání systému do tzv. **INDIKACE** nebo řízení souřadnic mimo CNC systém bude podrobně rozepsán v kapitole "Přepínání režimu INDIKACE-CNC u systému CNC836". V tomto případě se na řízení používají bity **INDIK**, kde se kromě jiného také vypíná polohová vazba.

Příklad:

Zapínání polohové vazby při uvolnění souřadnice Z.

V přípravných funkcích:

LDR	PO_OSZPI	;požadavek na pohyb
JL0	POZ_E	;obskok
FL	1,UVOL_Z	;start mechanismu uvolňování
EX		;
LDR	UVOL_Z	;čekáme na uvolnění
EX1		;
FL	1,VAZBA_Z	;zapnutí polohové vazby

POZ_E: ;v závěrečných funkcích bude vypnutí vazby společně
;s upnutím souřadnice

8.3.5 Polohování vřetene

POLOHV_xPolohování vřetene

Příznak pro polohování vřetene je v bitu **POLOHV_x** v **POLOHV**.

- ♦ log. 1 rotační souřadnice je ve stavu nájezdu na nulový puls
- ♦ log. 0 rotační souřadnice dosáhla nulový puls

Příznak **POLOHV_x** slouží jenom pro čtení a používá se v režimu přepnutí vřetene na rotační souřadnici (polohování rotační souřadnice). Používá se v součinnosti instrukce **SPI_AX_x**, která změní rychlostní vazbu rotační souřadnice na polohovou a tato se začne pohybovat dojížděcím posuvem a zastaví se až po dosažení nulového pulsu. Nájezd na nulový puls trvá určitou dobu, a proto PLC program musí pozastavit vykonávání dalších funkcí do dokončení nájezdu. Pro indikování najetí rotační souřadnice slouží bitové proměnné **POLOHV_x** (viz kapitolu "Popis řízení regulátorů pohonů rotačních os a vřeten").

8.3.6 Řízení zobrazení os v indikaci

INDIK_xŘízení os v indikaci

Bitové proměnné, které povolují zobrazení os v indikaci jsou **INDIK_x** v **INDIKPI**.

- ♦ log. 0 zobrazení indikační souřadnice zakázáno
- ♦ log. 1 zobrazení indikační souřadnice povoleno

Systém CNC836 může mít kromě souřadnic řízených polohovou vazbou i několik souřadnic bez polohové vazby, které jsou trvale v režimu "INDIKACE".

Je důležité rozlišit případ přepínání celého systému do režimu "INDIKACE", kdy se dočasně vypíná polohová vazba klasických NC souřadnic (řídí se bitem **INDIK** - popsáno dále) a možností mít kromě klasických NC souřadnic několik dalších trvale jen v indikaci. PLC program jenom řídí jejich zobrazení na obrazovce systému.

Při zobrazování indikačních os se na obrazovce nezobrazuje difference a distance. Ve strojních konstantách musí být nastavená v příslušné strojní konstantě 00 ... 05 třetí dekáda na hodnotu "2" (viz kapitolu "Strojní konstanty systému"). V systému musí být indikační osy umístěny na konci.

Podrobně je problematika indikačních os popsána v kapitole "Přepínání režimů **INDIKACE-NC** u systému CNC836".

8.3.7 Informační bity o volbě modifikace režimu AUT

VR_AUT, VR_BB, VR_M01, VR_AVP, VR_LOM, VR_ND ...Modifikace AUT

Všechny bity jsou průběžně aktualizovány a nastavují se na hodnotu log.1 při volbě příslušné modifikace režimu AUT. PLC program tyto bity může využít na blokování nebo aktivování různých akcí na stroji v závislosti na modifikaci režimu AUT. Doporučuje se bity testovat v modulu přípravných funkcí.

8.3.8 Povolení pohybu pro pomocné ruční pojezdy (BZH08MAN)

MPxMAN ...Povolení pohybu pro pomocné ruční pojezdy

Bity MPxMAN slouží speciálně pro povolování pomocných ručních pohybů, podobně jako bity MPxPI, ale jen v případě, že 4.dekáda strojní konstanty R233 je nastavena na hodnotu 2. Podrobně je tato problematika popsána v kapitole "Pomocné ruční pojezdy".

8.3.9 Informační bity pomocných ručních pojezdů (CONT_AUTMAN)

AM_x0, AM_x1, AUTMAN_CONT_NC, AUTMAN_CONT_TOC,
AUTMAN_CONT_SELECT, AUTMAN_CONT_G00

Tyto bity informují PLC program o řízení pomocných ručních pojezdů z panelu systému nebo z panýlku točítka. Bity AM_x0 a AM_x1 určují požadavek na pohyb v kladném a v záporném směru. Podrobně jsou významy všech bitů popsány v kapitole "Pomocné ruční pojezdy".

8.3.10 Bity pro externí řízení pomocných ručních pojezdů (EXT_CONT_AUTMAN)

EXM_x0, EXM_x1, REQ_EXT_G00_AUTMAN, REQ_EXT_SELECT_AUTMAN
REQ_EXT_CONT_AUTMAN, REQ_EXT_FEED_AUTMAN, G95_AUTMAN

Tyto bity slouží pro možnost zadávání pohybu, směru pohybu a rychlosti pohybu v pomocných ručních pojedech z PLC programu. Externí řízení pomocných ručních pojezdů je umožněno nastavením hodnoty log.1 v bitu REQ_EXT_CONT_AUTMAN. Bity EXM_x0 a EXM_x1 slouží pro zadávání pohybu a směru v pomocných ručních pojedech. Podrobně je tato problematika popsána v kapitole "Pomocné ruční pojezdy".

8.4 Povelový blok

Povel z CNC systému do PLC programu. Povelový blok se vysílá z panelu do kazety systému jednorázově po startu bloku. Výjimkou jsou bitové proměnné umístěné v **PB02**, které se přenášejí do kazety systému průběžně.

8.4.1 Společné zásady pro povelový blok

Většinu bitových proměnných umístěných v PB00, PB01 a v PB02 využívá supervizor interfejsu a pro vlastní návrh PLC programu mají menší význam.

Změnové signály se nastaví na hodnotu "1", když je změna v dané skupině technologické funkce nebo když je v bloku programována.

Dekódované funkce jsou umístěny v PB17 a PB18. Platí pravidlo, že bity, které jsou ukončeny příponou "**..PI**", mají trvalou platnost a mění se až při změně technologické funkce. Bity, které mají příponu "**..PID**", jsou nastaveny jenom v bloku, ve kterém byly programovány.

8.4.2 Přehled signálů povelového bloku

V tabulce je přehled všech signálů povelového bloku. Podrobné vysvětlení bude v dalším popisu.

Bity v tabulce, které jsou uvedeny slabě kurzívou, využívá supervizor interfejsu a při návrhu PLC programu se nevyužívají. Ve speciálních případech je možné je použít jenom na čtení.

Záznamy kanálu PLC→CNC					
	název	bit 7. bit 3.	bit 6. bit 2.	bit 5. bit 1.	bit 4. bit 0.
16	PB00	<i>UHEL_SPIR</i> RUPI	NDPI REFPI	DGNPI CAPI	AUTPI <i>VLEC</i>
17	PB01	POTENC <i>KOPIR1</i>	<i>VCHODU</i> <i>KOPIR2</i>	GOOPI <i>SPIRAL</i>	<i>G33PI</i> <i>PLYNMPI</i>
18	PB02	STOPTL STOPC	<i>RESTART</i> GOOTLPI	<i>PO_POH</i> <i>GOOPR10PI</i>	STARTC <i>END_BLOK</i>
19	PB03	BCD kód M funkce uživatelské skupiny 10			
1A	PB04	BCD kód P funkce			
1B	PB05	BCD kód H funkce			
1C	PB06	BCD kód M funkce skupiny 2			
1D	PB07	BCD kód TO funkce			
1E	PB08	BCD kód T1 funkce			
1F	PB09	BCD kód T2 funkce			
20	PB10	BCD kód T3 funkce			
21	PB11	BCD kód S0 funkce			
22	PB12	BCD kód S1 funkce			
23	PB13	BCD kód M funkce uživatelské skupiny 11			
24	PB14	BCD kód M funkce uživatelské skupiny 12			
25	PB15	BCD kód M funkce uživatelské skupiny 13			
26	PB16	BCD kód M funkce uživatelské skupiny 14			
27	PB17	M02PID M07PI	M03PI M08PI	M04PI M50PI	M19PI M51PI
28	PB18	M44PI M10PID	M43PI M11PID	M42PI M06PID	M41PI M60PI
29	PB19	ZMSHPI	ZMSOPI	ZMSMPI ZMSTPI	ZMSPPI ZMSSPI
2A	PB20	PO_OS4	PO_OSZ	PO_OS6 PO_OSY	PO_OS5 PO_OSX
2B	PB21	PSEU_4	PSEU_Z	PSEU_6 PSEU_Y	PSEU_5 PSEU_X
2C	PB22	ZMM8PI ZMM4PI	ZMM7PI ZMM3PI	ZMM6PI ZMM2PI	ZMM5PI ZMM1PI
2D	PB23	ZMM12PI	ZMM11PI	ZMM14PI ZMM10PI	ZMM13PI ZMM9PI
2E	PB24,25	čítač časové prodlevy low			
30	PB26	M30_PID	M02_PID	M01_PID	M00_PID

Význam signálů pro povelový blok CNC → PLC		
UHEL_SPIR	Zadání úhlu pro spirálu	1 = zadávání G10
NDPI	Návrat na dráhu	1 = návrat na dráhu
DGNPI	Diagnostický režim	1 = zvolen
AUTPI	Automatický režim	1 = zvolen
RUPI	Ruční režim	1 = zvolen
REFPI	Režim REF	1 = zvolen
CAPI	Režim CA	1 = zvolen
VLEC	Vlečení	1 = aktivní
END_BLOK	Potvrzení PO_F	1 = konec bloku
PO_POH	Potvrzení pohybu	1 = potvrzení
GOOPR10PI	10 proc. G00	1 = je zmáčkuto
PSEUDO	Pseudoreference	1 = odstartována
VCHODU	Systém v chodu	1 = v chodu
GOOPI	Rychloposuv	1 = G00
GOOTLPI	Tlač. rychloposuvu	1 = zmáčkuto
G33PI	Závitování	1 = G33
STARTC	Start celkový	1 = start
STOPC	Stop celkový	1 = stop
SRTOPTL	Stop od tlačítka	1 = stop
SPIRAL	Řízení spirály (G09)	1 = aktivní G09
PLYNMPI	Plynulá návaznost	1 = plynule
M02PID	M02, M30 dynamické	1 = M02, M03
M03PI, M04PI, M19PI, M07PI	Dekódované funkce	1 = vyslaná
M08PI, M50PI, M51PI, M41PI	Dekódované funkce	1 = vyslaná
M42PI, M43PI, M44PI, M10PI	Dekódované funkce	1 = vyslaná
M11PID	Dekódované funkce dynamické	1 = vyslaná
M06PID	Dekódované funkce dynamické	1 = vyslaná
M60PI	Dekódované funkce	1 = vyslaná
ZMSMPI	Změn. signál M sdružený	1 = vyslaný
ZMMxPI	Změn. signál Mx	1 = vyslaný
ZMSPPi	Změn. signál P	1 = vyslaný
ZMSHPi	Změn. signál H	1 = vyslaný
ZMSOPI	Změn. signál O	1 = vyslaný
ZMSTPI	Změn. signál T	1 = vyslaný
ZMSSPI	Změn. signál S	1 = vyslaný
PO_OSx	Pohyb v osách	1 = je pohyb
PSEU_x	Pseudoreference v ose	1 = je požadována
M00_PID	Dekódovaná funkce M00	1 = M00
M01_PID	Dekódovaná funkce M01	1 = M01
M02_PID	Dekódovaná funkce M02	1 = M02
M30_PID	Dekódovaná funkce M30	1 = M30

8.4.3 Přehled použitých M funkcí podle skupin

Skupina 1:	M00	programový stop (řídí M00_PID, M01_PID, M02_PID, M30_PID)	
	M01	volitelný stop	
	M02	konec partprogramu	(řídí M02PID)
	M30	konec partprogramu	
Skupina 2:	M03	start vřetene CW (řídí M03PI, M04PI, M19PI a PB06)	
	M04	start vřetene CCW	
	M05	stop vřetene	
	M19	stop vřetene v orientovaném bodě	
Pozn.:	Do skupiny 2 lze zadat dalších 16 M-funkcí, které se definují ve strojních konstantách číslo R160 - R163. (např. M13 = M3 + M07 ... START vřetene a chlazení)		
Skupina 3:	M41	otáčky vřetene - rozsah 1	(řídí M41PI, M42PI, M43PI, M44PI)
	M42	otáčky vřetene - rozsah 2	
	M43	otáčky vřetene - rozsah 3	
	M44	otáčky vřetene - rozsah 4	
	M40	automatická převodovka	
Skupina 5:	M07	zapnutí chlazení 2	(řídí M07PI, M08PI)
	M08	zapnutí chlazení 1	
	M09	vypnutí chlazení 1 a 2	
	M17	zapnutí chlazení 1 a 2	
Skupina 6:	M50	zapnutí chlazení 3	(řídí M50PI, M51PI)
	M51	zapnutí chlazení 4	
	M52	zapnutí chlazení 3 a 4	
	M53	vypnutí chlazení 3 a 4	
Skupina 7:	M10	upnutí obrobku	(řídí M10PID , M11PID)
	M11	uvolnění obrobku	
Skupina 8:	M49	Překlenutí ručního FEED OVERRIDE	(nemá odezvu v PB)
	M48	Zrušení překlenutí FEED OVERRIDE	
Skupina 9:	M06	Výměna nástroje	(řídí M06PID a M60PI)
	M60	Výměna obrobku	
Skupina 10:	...	Funkce dle strojních konstant (R56)	(řídí PB03)
Skupina 11:	...	Funkce dle strojních konstant (R57)	(řídí PB13)
Skupina 12:	...	Funkce dle strojních konstant (R58)	(řídí PB14)
Skupina 13:	...	Funkce dle strojních konstant (R59)	(řídí PB15)
Skupina 14:	...	Ostatní pomocné funkce, které nejsou uvedeny v předešlých skupinách. (PB16)	

8.4.4 Bitové proměnné v povelovém bloku využívané supervizorem interfejsu

V následujícím textu jsou popsány bitové proměnné z povelového bloku, které využívá supervizor interfejsu nebo jiné systémové prostředky v kazetě systému. **PLC program tyto bitové proměnné většinou nepotřebuje** a ve speciálních případech mohou sloužit jen pro čtení.

UHEL_SPIRZadání úhlu pro spirálu

Zadání úhlu pro spirálu využívají systémové prostředky interpolátoru v kazetě CNC systému. Systém umožňuje řídit 2 souřadnice v kruhové interpolaci a současně 1 souřadnici v lineární interpolaci. Pomocí funkce G10 se zadává stoupání na jednu otáčku kruhové interpolace (viz Návod na programování).

SPIRALŘízení spirály

Blok, ve kterém je programován pohyb po spirále pomocí funkce G09. Bitovou proměnnou využívají systémové prostředky interpolátoru v kazetě CNC systému.

VLECVlečení

Systém CNC836 umožňuje programovat vlečení maximálně dvou souřadnic od jiných dvou souřadnic. Vlečení se programuje pomocí funkce G08 (viz Návod na programování). Tímto způsobem je umožněno řídit například dvojsupportový soustruh. Vlečené souřadnice mohou být vlečeny i zrcadlově. Vlečení využívají systémové prostředky interpolátoru v kazetě CNC systému.

VCHODUSystém v chodu

Bit **VCHODU** indikuje hodnotou log.1, že byl v panelu odstartován blok. Bit je umístěn v povelovém bloku, který se přepisuje jen novým startem, proto zůstává trvale nahozen do log.1. Pokud by PLC program potřeboval tento bit snímat, musel by jej po sejmutí vynulovat. Struktura PLC programu a existence modulu přípravných funkcí (viz "Struktura PLC programu") umožňují lepším způsobem řídit akce od odstartování bloku. Proto je tento bit uveden jen pro kompatibilitu se staršími verzemi automatů a je neperspektivní.

G33PIZávitování

Bit **G33PI** indikuje hodnotou log.1, že v bloku je programováno závitování funkcí G33. Bit G33PI využívají systémové prostředky interpolátoru v kazetě CNC systému.

KOPIR1, KOPIR2Kopírování

Systém CNC836 umožňuje využívat funkce na kopírování dle předlohy. Bit signalizuje hodnotou log.1, že systém je nastaven do režimu kopírování. Jedná se o speciální režim, který umožňuje pomocí odměřovací dotykové sondy obrábění v jedné ose podle vzorového kusu - šablony. Kontura vzorového kusu je sledována odměřovací sondou a naměřené údaje jsou vstupem pro pohyb řízené souřadnice. Systém umožňuje tento způsob řízení v jedné souřadnici a programuje se funkcí G05.

Bity jsou kromě kopírování využity i k jiným účelům, například pro funkci G24 - těsné navázání bloků. PLC program tyto bity ke své činnosti nepotřebuje. Bity využívají systémové prostředky a supervizor interfejsu.

PLYNMPI**....Plynulá návaznost**

Bit **PLYNMPI** hodnotou log. 1 indikuje, že bude plynulé navázání bloků. Pro plynulou návaznost je potřeba, aby směrnice sousedních bloků byly shodné a byla nastavena 1. dekáda strojní konstanty 97 na hodnotu 0 (viz Návod na obsluhu). Pro PLC program je tento bit pouze informativní.

RESTART**....Start po stopu v režimu AUT**

Bit **RESTART** se nastaví na hodnotu log.1 v případě, že je odstartován blok po předchozím stopu v režimu AUT. Bit má jenom pulsní průběh a je aktualizován průběžně. V tomto případě se nevysílá nový povelový blok. Supervizor interfejsu zabezpečí pokračování průchodu bloku od předchozího přerušení stopem v případě, že je nastavena 7. dekáda strojní konstanty 98 na hodnotu 0. V případě, že je 7. dekáda strojní konstanty 98 nastavena na hodnotu 1, supervizor interfejsu po startu zopakuje průchod bloku od začátku (viz "Strojní konstanty systému"). Pro PLC program je tento bit pouze informativní.

PO_POH**....Potvrzení pohybu**

Bit **PO_POH** slouží pro synchronizaci mezi přípravou bloku probíhající v panelu a supervizorem interfejsu. Bit má jenom pulsní průběh a je aktualizován průběžně. PLC program tento bit nepotřebuje využívat.

G00PR10PI**....10 % rychloposuvu v ručním režimu**

Bit **G00PR10PI** hodnotou log.1 signalizuje, že byl zvolen režim snížení rychlosti rychloposuvu. Bit je aktualizován průběžně. Pro PLC program má bit pouze informační charakter.

Režim sníženého rychloposuvu se vyvolá stisknutím tlačítka 10% panelu obsluhy. Nastavení tohoto režimu je indikováno na monitoru zobrazením značky 10%. Zrušení režimu snížení rychloposuvu se provede opětovným stisknutím tlačítka 10%.

Pátou dekádou strojní konstanty 97 je možno tuto funkci modifikovat (viz "Strojní konstanty systému"). Možno nastavit plynulou změnu rychloposuvu podle potenciometru FEED-OVERRIDE na panelu systému. V tomto případě se indikuje na monitoru značka %G0.

END_BLOK**....Signál END_BLOK**

Bit **END_BLOK** je synchronizační bit mezi supervizorem interfejsu a přípravou bloku v panelu, aktualizovaný průběžně. Pro PLC program tento bit nemá význam.

PB24,PB25**....Časová prodleva**

Wordová hodnota v buňkách **PB24, PB25** je využívána supervizorem interfejsu, který do bloku vloží časovou prodlevu naprogramovanou pomocí funkce G04. Časová prodleva se vloží po vykonání pohybu v bloku a před vykonáním modulu ZAVERCNE_FUNKCE.

8.4.5 Informace o režimu systému v povelovém bloku

V povelovém bloku je několik informačních bitů, které určují, v jakém režimu je systém. Informace o režimu je také obsažena v buňce REZIMPI definované v modulu INCLEXT.A86 a popsané v úvodu kapitoly. V povelové bloku je několik základních režimů dekodováno do bitů. Jedná se o režimy simulace, automatický režim, ruční, potenciometry, centrální anulace a nájezd do reference. Všechny tyto bity nejsou průběžné (na rozdíl od buňky REZIMP) a jsou aktualizovány jen po startu bloku ze systému.

Dále jsou popsány bity informující o stopu, startu a ryhloposuvu, které jsou průběžně aktualizovány.

DGNPIRežim simulace

- ♦ log.1 režim simulace
- ♦ log.0 normální provozní stav

Režim DGN je vhodné používat pro ladění programu z hlediska geometrie dráhy, bez skutečných pohybů a bez technologických funkcí stroje.

Režim simulace se na systému zvolí dvěma způsoby:

- ♦ Nastavením první dekády strojní konstanty 98 na hodnotu 1 (viz Popis strojních konstant). V tomto případě je systém trvale v režimu simulace.
- ♦ Před odstartováním programu v režimu AUT se zvolí modifikace AVP (zrychlení posuv) a v dotazu, jestli má být simulační běh, se potvrdí volba nastavením "ANO". V tomto případě se odstartuje jen příslušný program v režimu simulace.

Simulační režim spočívá v následujících změnách systému:

- ♦ nevysílají se vypočtené hodnoty z interpolace do pohonu, přesněji - nezadáva se přírůstek do softwerového diferenčního čítače
- ♦ udržuje se nadále softwerová polohová vazba
- ♦ do PLC programu se nevysílají žádné změnové signály v povelovém bloku
- ♦ supervizor interfejsu neprovádí moduly PRIPRAVNE_FUNKCE a ZAVERECNE_FUNKCE
- ♦ modul PROVOZ_VYSTUP probíhá normálně

CNC systém tak umožní jetí programu jen vnitřně bez skutečných pohybů a bez technologických funkcí stroje. Pohyb možno sledovat například pomocí formátu grafiky.

PLC program by měl bitem DGNPI blokovat všechny akce, které jsou spustitelné v modulu PROVOZ_VYSTUP bez změnových signálů.

AUTPIRežim AUT

- ♦ log.1 režim AUT
- ♦ log.0 není režim AUT

Bit **AUTPI** signalizuje hodnotou log.1, že byl odstartován blok v automatickém režimu (AUT), včetně všech možností modifikace (BB, AVP, M01, ND a LOM). Bit se vysílá jen po startu bloku a není průběžně aktualizován. Případná změna režimu se projeví až novým startem bloku, například odstartováním pohybu v ručním režimu nebo odstartováním centrální anulace.

RUPI**....Ruční režimy**

- ♦ log.1 některý z ručních režimů
- ♦ log.0 není ruční režim

Bit **RUPI** signalizuje hodnotou log.1, že byl odstartován blok v některém z ručních režimů. Jedná se o režimy:

- ♦ MAN+ MAN- režim "MANUAL"
- ♦ JOG+ JOG- režim "JOG"
- ♦ POT režim potenciometrů
- ♦ TOC režim ručního točítka

Bit se vysílá jen po startu bloku a není průběžně aktualizován. Případná změna režimu se projeví až novým startem bloku.

V případě, když PLC program potřebuje provádět určité činnosti jen v ručním režimu a testoval by za tímto účelem bit RUPI, musel by se nejdříve odstartovat blok v ručním režimu, což může způsobit pohyb souřadnice. Proto je v tomto případě vhodnější testovat buňku REZIMPI definovanou v modulu INCLEXT.A86, která je aktualizována průběžně.

REFPI**....Režim nájezdu do reference**

- ♦ log.1 je režim nájezdu do reference
- ♦ log.0 není režim nájezdu do reference

Bit **REFPI** signalizuje hodnotou log.1, že byl odstartován blok v režimu nájezdu do reference. Bit je nastavován při skutečném nájezdu do reference a při pseudoreferenci. V případě simulace reference a nulování reference nastavován není. Bit se vysílá jen po startu bloku a není průběžně aktualizován. Případná změna režimu se projeví až novým startem bloku.

CAPI**....Režim centrální anulace**

- ♦ log.1 režim centrální anulace
- ♦ log.0 není režim centrální anulace

Bit **CAPI** signalizuje hodnotou log.1, že byl odstartován blok v režimu centrální anulace. Bit se vysílá jen po startu bloku a není průběžně aktualizován. Případná změna režimu se projeví až novým startem bloku.

Supervizor interfejsu při centrální anulaci řídí průchod moduly PLC programu nestandardně. Když je potřeba vykonat nějakou akci od centrální anulace, doporučuje se, aby byl daný úsek programu umístěn v modulu PROVOZ_VYSTUP a startován od bitu CAPI v součinu s bitem STARTC.

POTENC**....Režim potenciometrů**

- ♦ log.1 režim potenciometrů
- ♦ log.0 není režim potenciometrů

Bit **POTENC** signalizuje hodnotou log.1, že byl odstartován blok v režimu řízení posuvu potenciometry. Režim potenciometrů je modifikací ručního režimu, a tak s bitem POTENC je současně nastaven bit RUPI.

Bit se vysílá jen po startu bloku a není průběžně aktualizován. Případná změna režimu se projeví až novým startem bloku.

Bit je shozen do Log0 až při zvolení a odstartování jiného režimu. Z toho vyplývá, že je vhodné, aby PLC program sledoval stav tlačítka STOP (bit PO_STOP v BZH01), stav tohoto bitu je aktualizován průběžně a v případě stisku tohoto tlačítka zastavil pohyb odvozený z potenciometru.

Režim MAN s modifikací POTENC umožní pomocí systému simulovat ruční řízení pohybu os stroje potenciometry. Ovládání os potenciometry je obdobné, jako by na řídicí vstupy regulátoru pohonu byly připojeny potenciometry pro ruční zadání rychlosti. Systém však udržuje řízené souřadnice v polohové vazbě, takže je možno provádět přepínání mezi tímto a standardním způsobem řízení.

Je-li na stroji realizováno řízení potenciometry, využívá PLC program bit POTENC k zjištění, že je obsluhou zvolen tento režim. Je-li tento režim zvolen, musí uživatelský program PLC obsluhovat bity POTx v BZH09 a POT_SGNx v BZH10.

PLC program však může řídit pohyby os pomocí potenciometrů i bez polohové vazby. Blíže o celé problematice řízení posuvu pomocí potenciometrů je v kapitole "Řízení pohybu pomocí potenciometrů".

G00PI**....Rychloposuv**

- ♦ log.1 rychloposuv
- ♦ log.0 pracovní posuv

Bit **G00PI** signalizuje hodnotou log.1, že byl odstartován blok, ve kterém je programován rychloposuv. Bit se vysílá jen po startu bloku a není průběžně aktualizován. Případná změna režimu se projeví až novým startem bloku.

Bit G00PI se dá využít například pro uvolnění zpomalovacích limitních spínačů, a tak se zabezpečí, že nebudou reagovat pro pracovní posuv, ale jen pro rychloposuv.

STOPTL**....Tlačítko stop**

- ♦ log.1 stop od tlačítka
- ♦ log.0 není stop

Do bitu **STOPTL** je vysílán z panelu systému jedničkový puls, když je stisknuto tlačítko STOP na panelu systému CNC836. Puls STOPTL je vysílán jenom tehdy, když má stisknutí tlačítka STOP na panelu smysl, například při rozpracovaném bloku v režimu AUT. Bit STOPTL je aktualizován průběžně.

STOPC**....Celkový stop**

- ♦ log.1 stop systému
- ♦ log.0 není stop systému

Bit **STOPC** je statický signál a úroveň log.1 signalizuje, že je systém ve stavu STOP (žádný blok není rozpracován). Systém se dostane do stavu STOP například zmačknutím tlačítka STOP na panelu systému nebo z PLC programu bitem STOPPI v bloku zpětného hlášení. Bit STOPC je aktualizován průběžně.

STARTC**....Start celkový**

- ♦ log.1 start systému
- ♦ log.0 není start systému

Bit **STARTC** nabývá hodnoty log.1 po dobu držení tlačítka START na panelu systému (kopie tlačítka START do povelového bloku) nebo při odstartování z PLC programu bitem STARTPI v bloku zpětného hlášení a také odstartováním pomocí sekvencí stisku tlačítek (viz kapitolu "Způsoby dálkového ovládání CNC systému z PLC programu"). Bit je aktualizován průběžně.

G00TLPI**....Tlačítko rychloposuvu**

- ♦ log.1 stisknuto tlačítko rychloposuvu
- ♦ log.0 není stisknuto tlačítko rychloposuvu

Bit **G00TLPI** hodnotou log.1 signalizuje, že bylo stisknuto tlačítko rychloposuvu v ručním režimu. Rychloposuv v ručním režimu platí jen po dobu držení tlačítka. Bit je aktualizován průběžně.

8.4.6 BCD funkce v povelovém bloku

V další části budou popsány funkce, které vystupují v povelovém bloku v BCD tvaru. Jedná se o všechny skupiny M-funkcí a funkce S, T, P, H. Změna v BCD hodnotách u těchto funkcí je vždy doprovázena příslušným změnovým bitem. Změnové bity povelového bloku budou popsány v další podkapitole.

PB03**....M funkce skupiny 10**

V **PB03** je BCD kód M funkce ze skupiny 10. Funkce sdružené do skupiny 10 jsou určeny k volnému použití. Do skupiny 10 jsou zařazeny funkce, které jsou určeny strojní konstantou 56 (viz kapitolu "Strojní konstanty systému"). Pomocí strojní konstanty 56 je možno zařadit do skupiny 10 maximálně čtyři libovolné M funkce, které nejsou zařazeny do ostatních skupin.

Změnový bit této skupiny je ZMM10PI v PB23

PB13**....M funkce skupiny 11**

V **PB13** je BCD kód M funkce ze skupiny 11. Funkce sdružené do skupiny 11 jsou určeny k volnému použití. Do skupiny 11 jsou zařazeny funkce, které jsou určeny strojní konstantou 57 (viz kapitolu "Strojní konstanty systému"). Pomocí strojní konstanty 57 je možno zařadit do skupiny 11 maximálně čtyři libovolné M funkce, které nejsou zařazeny do ostatních skupin.

Změnový bit této skupiny je ZMM11PI v PB23

PB14**....M funkce skupiny 12**

V **PB14** je BCD kód M funkce ze skupiny 12. Funkce sdružené do skupiny 12 jsou určeny k volnému použití. Do skupiny 12 jsou zařazeny funkce, které jsou určeny strojní konstantou 58 (viz kapitolu "Strojní konstanty systému"). Pomocí strojní konstanty 58 je možno zařadit do skupiny 12 maximálně čtyři libovolné M funkce, které nejsou zařazeny do ostatních skupin.

Změnový bit této skupiny je ZMM12PI v PB23.

PB15**....M funkce skupiny 13**

V **PB15** je BCD kód M funkce ze skupiny 13. Funkce sdružené do skupiny 13 jsou určeny k volnému použití. Do skupiny 13 jsou zařazeny funkce, které jsou určeny strojní konstantou 59 (viz kapitolu "Strojní konstanty systému"). Pomocí strojní konstanty 59 je možno zařadit do skupiny 13 maximálně čtyři libovolné M funkce, které nejsou zařazeny do ostatních skupin.

Změnový bit této skupiny je ZMM13PI v PB23.

PB16**....M funkce skupiny 14**

V **PB16** je BCD kód M funkce ze skupiny 14. Funkce sdružené do skupiny 14 jsou určeny k volnému použití. Do skupiny 14 jsou automaticky zařazeny všechny M funkce funkce, které nejsou sdruženy do jiných skupin. BCD kód v PB16 může nabývat hodnot 10 - 99. Tato skupina může obsahovat větší počet M-funkcí, které PLC program využívá.

Změnový bit této skupiny je ZMM14PI v PB23

PB06**....M funkce skupiny 2**

V **PB06** je BCD kód M funkce ze skupiny 2. Funkce ze skupiny 2 jsou určeny pro řízení vřetene a obsahují funkce M03, M04, M05 a M19 (viz Přehled použitých M funkcí.). Funkce ze skupiny 2 mají v povelovém bloku současně i dekodovaný bitový výstup (viz "Dekodované funkce povelového bloku").

Často je ale potřeba k těmto funkcím združit ještě další funkce, například start vřetene a současně start chlazení, stop vřetene a stop chlazení. Užitečná je také funkce start vřetene v přípravných funkcích a stop vřetene v tom samém bloku v závěrečných funkcích. Proto do skupiny 2 lze zadat dalších 16 M-funkcí, které se definují ve strojních konstantách číslo 160 - 163.

Funkce definované ve strojní konstantě 160 patří k základní funkci M3 a při jejich naprogramování se automaticky vysílá i dekodovaný bit M03PI.

Funkce definované ve strojní konstantě 161 patří k základní funkci M4 a při jejich naprogramování se automaticky vysílá i dekodovaný bit M04PI.

Funkce definované ve strojní konstantě 162 a 163 patří k základní funkci M5 a při jejich naprogramování se automaticky vynulují bity M03PI a M04PI (např. M13 = M3 + M07 ... START vřetene a chlazení).

PB04**....Funkce P**

V **PB04** je BCD kód funkce P. Funkce je určena k volnému použití a může nabývat hodnot 0-99.

Změnový bit této skupiny je ZMSPPI v PB19.

PB05**....Funkce H**

V **PB05** je BCD kód funkce H. Funkce je určena k volnému použití a může nabývat hodnot 0-99. Na panelu systému CNC836 není standardně tlačítko H. V případě použití této funkce je možnost definovat tlačítko H v konfiguračním souboru CNC836.KNF například místo 6 souřadnice.

Změnový bit této skupiny je ZMSHPI v PB19.

PB07**....Funkce T - 1. BYTE****PB08****....Funkce T - 2.BYTE****PB09****....Funkce T - 3.BYTE****PB10****....Funkce T - 4.BYTE**

Do byte **PB07** jsou předávány dvě nejnižší dekády funkce T v BCD kódu (1. a 2. dekáda).

Do byte **PB08** jsou předávány 3. a 4. dekáda funkce T v BCD kódu.

Do byte **PB09** jsou předávány 5. a 6. dekáda funkce T v BCD kódu.

Do byte **PB10** jsou předávány dvě nejvyšší dekády funkce T v BCD kódu. (7. a 8. dekáda).

Funkce T je určena k programování čísla nástroje. Data předávaná do PB07 až PB10 jsou ve formátu BCD. Funkci T je možno programovat v rozsahu osmi dekád, tedy to je v rozsahu 0 až 99999999. Do povelového bloku je hodnota předána do PB07, PB08 PB09 PB10 jako osm BCD dekád, přičemž platí, že v PB07 jsou dvě nejnižší dekády a v PB10 dvě nejvyšší dekády.

Změnový bit pro funkci T je ZMSTPI v PB19.

PB11Funkce S - 1.BYTE
PB12Funkce S - 2.BYTE

Byte **PB11** obsahuje spodní byte hodnoty programovaných otáček vřetene. Byte **PB12** obsahuje horní byte hodnoty programovaných otáček vřetene.

Formát dat o programované funkci S je předáván do PLC do PB11 a PB12 a je závislý na nastavení strojních konstanta 60 a 61. Formát může být v binárním tvaru nebo BCD tvaru nebo v procentech z určené hodnoty. Podrobně je o této problematice napsáno v kapitole "Zadávaní otáček vřetena".

Hodnota v PB11 a v PB12 je aktualizována jen na začátku bloku a bez ohledu na nastavení potenciometru %S. Pokud PLC program potřebuje řídit otáčky vřetene s ohledem na %S, musí využívat buňku VYSOVERS, která je průběžně aktualizována a je definována v souboru INCLEXT.A86.

Změnový signál se nazývá ZMSSPI a je v PB19.

8.4.7 Dekódované funkce v povelovém bloku

V povelovém bloku je umístěno 20 dekodovaných M funkcí.

M02PIDDekódovaná funkce M00,M01,M02, M30
---------------	---

Bit **M02PID** se nastaví do log.1 na dobu jednoho bloku, je-li v bloku programována některá z funkcí M02, M00, M01 a M30. Systém vyhodnotí konec a v případě funkcí M02 a M30 nastaví ještě další funkce v povelovém bloku, například stop vřetena a stop chlazení.

M00_PID	...Dekódovaná funkce M00
----------------	---------------------------------

Bit **M00_PID** se nastaví do log.1 na dobu jednoho bloku, je-li v bloku programována funkce M00. Změnový signál pro M00_PID je ZMM1PI.

M01_PID	...Dekódovaná funkce M01
----------------	---------------------------------

Bit **M01_PID** se nastaví do log.1 na dobu jednoho bloku, je-li v bloku programována funkce M01. Jestli se funkce M01 v bloku skutečně uplatní, závisí od toho, zda je aktivní modifikace M01 režimu AUT. PLC program si může modifikaci režimu AUT přechíst pomocí informačního bitu VR_M01. Změnový signál pro M01_PID je ZMM1PI.

M02_PID	...Dekódovaná funkce M02
----------------	---------------------------------

Bit **M02_PID** se nastaví do log.1 na dobu jednoho bloku, je-li v bloku programována funkce M02. Změnový signál pro M02_PID je ZMM1PI.

M30_PID	...Dekódovaná funkce M30
----------------	---------------------------------

Bit **M30_PID** se nastaví do log.1 na dobu jednoho bloku, je-li v bloku programována funkce M30. Změnový signál pro M30_PID je ZMM1PI.

M03PI**....Dekódovaná funkce M03**

Dekódovaný bit **M03** je statický a je určen pro roztočení vřetena ve směru točení hodinových ručiček. Je-li v bloku technologického programu programována funkce M03 nebo některá funkce určena strojní konstantou 160, je na počátku výkonu tohoto bloku nastaven bit M03PI do log.1. Tento stav setrvává do doby, než je jinou funkcí z 2. skupiny M změněn.

Pomocí strojních konstant 160, 161, 162 a 163 je možné provést rozšíření této skupiny o maximálně dalších 16 M funkcí, které nejsou zařazeny do skupiny 2 a jsou vysílány do PB06. Je-li programována funkce, jenž je uvedena ve strojní konstantě 160, je bit M03PI nastaven do log.1 (rozšíření M03). Je-li programována funkce, jenž je uvedena ve strojní konstantě 162 nebo 163 jsou bity M03PI, M04PI vynulovány.

Po naprogramování funkce M03 se předpokládá naprogramování funkce M03, M05 nebo M19 (zaploňování vřetene). Je-li konec programu dřív než je programováno zastavení vřetene, vydá systém automaticky funkci M05.

M04PI**....Dekódovaná funkce M04**

Dekódovaný bit **M04** je statický a je určen pro roztočení vřetena proti směru točení hodinových ručiček. Je-li v bloku technologického programu programována funkce M04 nebo některá funkce určena strojní konstantou 161, je na počátku výkonu tohoto bloku nastaven bit M04PI do log.1. Tento stav setrvává do doby, než je jinou funkcí z 2. skupiny M změněn.

Pomocí strojních konstant 160, 161, 162 a 163 je možné provést rozšíření této skupiny o maximálně dalších 16 M funkcí, které nejsou zařazeny do skupiny 2 a jsou vysílány do PB06. Je-li programována funkce, jenž je uvedena ve strojní konstantě 161, je bit M04PI nastaven do log.1 (rozšíření M04). Je-li programována funkce, jenž je uvedena ve strojní konstantě 162 nebo 163 jsou bity M03PI, M04PI vynulovány.

Po naprogramování funkce M04 se předpokládá naprogramování funkce M03, M05 nebo M19 (zaploňování vřetene). Je-li konec programu dříve, než je programováno zastavení vřetene, vydá systém automaticky funkci M05.

Změnový signál pro M04PI je ZMM2PI.

M19PI**....Dekódovaná funkce M19**

Dekódovaný bit **M19** je statická a slouží pro programování zaploňování vřetena. Celé zaploňování vřetena musí provést PLC program. Bit M19PI se nastaví na hodnotu log.1, když je funkce M19 v bloku programována a tento stav trvá do doby, než je jinou funkcí z 2. skupiny M změněn.

Funkce M19 nastaví v BCD tvaru buňku PB06 na hodnotu 19h.

Změnový signál pro M04PI je ZMM2PI.

M07PI**....Dekódovaná funkce M07**

Dekódovaný bit **M07PI** je statický a je určen pro zapnutí chlazení 1. Je-li v bloku technologického programu programována funkce M07 nebo M17, je na počátku výkonu tohoto bloku nastaven bit M07PI do log.1. Funkce M07, M08, M09 a M17 patří do 5. skupiny M funkcí.

Je-li v bloku technologického programu programována funkce M09, je na počátku výkonu tohoto bloku shoen bit M07PI do log.0.

Po zapnutí systému a po centrální anulaci je bit M07PI vynulován.

Změnový signál pro M07PI je ZMM5PI.

M08PI**....Dekódovaná funkce M08**

Dekódovaný bit **M08PI** je statický a je určen pro zapnutí chlazení 2. Je-li v bloku technologického programu programována funkce M08 nebo M17, je na počátku výkonu tohoto bloku nastaven bit M08PI do log.1. Funkce M07, M08, M09 a M17 patří do 5. skupiny M funkcí.

Je-li v bloku technologického programu programována funkce M09, je na počátku výkonu tohoto bloku shoen bit M08PI do log.0.

Po zapnutí systému a po centrální anulaci je bit M08PI vynulován.

Změnový signál pro M08PI je ZMM5PI.

M50PI**....Dekódovaná funkce M50**

Dekódovný bit **M50PI** je statický a je určen pro zapnutí chlazení 3. Je-li v bloku technologického programu programována funkce M50 nebo M52, je na počátku výkonu tohoto bloku nastaven bit M50PI do log.1. Funkce M50, M51, M52 a M53 patří do 6. skupiny M funkcí.

Je-li v bloku technologického programu programována funkce M53, je na počátku výkonu tohoto bloku shozen bit M50PI do log.0.

Po zapnutí systému a po centrální anulaci je bit M50PI vynulován.

Změnový signál pro M50PI je ZMM6PI.

M51PI**....Dekódovaná funkce M51**

Dekódovný bit **M51PI** je statický a je určen pro zapnutí chlazení 4. Je-li v bloku technologického programu programována funkce M51 nebo M52, je na počátku výkonu tohoto bloku nastaven bit M51PI do log.1. Funkce M50, M51, M52 a M53 patří do 6. skupiny M funkcí.

Je-li v bloku technologického programu programována funkce M53, je na počátku výkonu tohoto bloku shozen bit M51PI do log.0.

Po zapnutí systému a po centrální anulaci je bit M51PI vynulován.

Změnový signál pro M51PI je ZMM6PI v PB22.

M41PI**....Dekódovaná funkce M41****M42PI****....Dekódovaná funkce M42****M43PI****....Dekódovaná funkce M43****M44PI****....Dekódovaná funkce M44**

Dekódovné bity **M41PI, M42PI, M43PI a M44PI** jsou statické a nastavují se podle funkcí M41, M42, M43 a M44. Funkce M41 je určena k zařazení prvního převodového stupně vřetene. Funkce M42, M43 a M44 je určena k zařazení druhého, třetího a čtvrtého převodového stupně vřetene.

Je-li v bloku technologického programu programována funkce M41, je na počátku výkonu tohoto bloku nastaven bit M41PI do log.1. Je-li v některém dalším bloku technologického programu programována některá z funkcí M42, M43, M44, je na počátku výkonu tohoto bloku shozen bit M41PI do log.0 a nahozen příslušný bit M42PI, M43PI nebo M44PI podle převodového stupně. CNC systém kontroluje, zda zadané otáčky odpovídají převodovému stupni podle strojních konstant 61, 62, 63 a 64.

Funkce M41, M42, M43, M44 a M40 patří do 3. skupiny M funkcí a mají změnový signál ZMM3PI v PB22.

Funkce M40 je automatická převodovka a způsobuje, že systém automaticky nastavuje proměnné M41PI, M42PI, M43PI, M44PI v závislosti na programované funkci S. Automatický výběr převodového stupně probíhá v závislosti na nastavení strojních konstant 61, 62, 63 a 64.

Podrobněji je o problematice řízení vřetena pojednáno v kapitole "Zadávání otáček vřetena".

M10PID**....Dekódovaná funkce M10****M11PID****....Dekódovaná funkce M11**

Bity **M10PID a M11PID** jsou dynamické a nastavují se jen po dobu trvání jednoho bloku. Nastavením hodnoty log.1 se určuje, že v bloku byla programována funkce upnutí obrobku M10 nebo funkce uvolnění obrobku M11.

Funkce M10 a M11 patří do 7. skupiny M funkcí a mají změnový signál ZMM7PI v PB22.

M06PID**....Dekódovaná funkce M06****M60PI****....Dekódovaná funkce M60**

Bit **M06PID** je dynamický s platností jednoho bloku a je nastaven na hodnotu log.1, když je v bloku programována funkce pro výměnu nástroje M06.

Funkce M06 a M60 patří do 9. skupiny M funkcí a mají změnový signál ZMM9PI.

Bit M60PI je rezervován pro výměnu obrobku u robotizovaných pracovišť a nastavuje se funkcí M60.

8.4.8 Změnové signály v povelovém bloku

Všechny změnové signály jsou bity umístěny v povelovém bloku v PB19, PB22 a v PB23. Všechny změnové signály jsou dynamické, to znamená, že jsou nastaveny jen po dobu trvání jednoho bloku.

Systém nastavuje změnové signály ve dvou případech:

- ♦ Je změna v dané technologické funkci (BCD výstup nebo dekodované funkce). Změna vznikla naprogramováním nové hodnoty, která je jiná než předcházející hodnota.
- ♦ Není změna hodnoty v dané technologické funkci, ale tato byla opět v bloku programována. Tato vlastnost je důležitá tehdy, když technologické funkce se ovládají ještě jiným (ručním) způsobem. Například když systém vyšle funkci M5 (stop vřetene) a pak PLC program na povel přímo od stroje vřeteno opět roztočí, může jej systém novým vysláním M5 zastavit, i když nebyla změna ve 2. skupině M funkcí.

ZMSMPIZdružený změnový bit pro všechny M funkce
ZMSPPIZměnový bit funkce P
ZMSHPIZměnový bit funkce H
ZMSTPIZměnový bit funkce T
ZMSSPIZměnový bit funkce S
ZMMxPIZměnový bit funkce Mx x=1,2,...,14

Změnové signály se v PLC programu testují hlavně v přípravných a závěrečných funkcích a na základě nich se až dekodují BCD výstupy v povelovém bloku nebo dekodované funkce. Aby nedocházelo ke zpomalení průchodu modulem přípravných a závěrečných funkcí, první instrukce typu EX se napíše až za obskokem, jak je patrné na příkladu:

Příklad:

Zapnutí vřetena.

```

LDR    ZMM2PI      ;test změnového signálu 2. skupiny M funkcí
JLO    OBSKOK        ;obskok
LDR    M03PI         ;test funkce M03
JLO    SKOKM4        ;skok na další testy
FL1    CW            ;start mechanismu CW
EX                      ;
LDR    CW            ;čekání na dokončení mechanismu CW
EX1                      ;
JUM     OBSKOK        ;skok na konec

```

8.4.9 Rozšíření rozhraní povelového bloku

Od verze software panelu 40.28 a sekundárního procesoru 6.313 je v PLC programu umožněno využívat rozšířenou část povelového bloku.

Název	Typ	Popis
PB_AX_X	DWORD	Programovaná hodnota 1. souřadnice v BCD tvaru.
PB_AX_Y	DWORD	Programovaná hodnota 2. souřadnice v BCD tvaru.
PB_AX_Z	DWORD	Programovaná hodnota 3. souřadnice v BCD tvaru.
PB_AX_4	DWORD	Programovaná hodnota 4. souřadnice v BCD tvaru.
PB_AX_5	DWORD	Programovaná hodnota 5. souřadnice v BCD tvaru.
PB_AX_6	DWORD	Programovaná hodnota 6. souřadnice v BCD tvaru.
PB_AX_I	DWORD	Programovaná hodnota I funkce v BCD tvaru jen v nepohybovém bloku.
PB_AX_J	DWORD	Programovaná hodnota J funkce v BCD tvaru jen v nepohybovém bloku.
PB_AX_K	DWORD	Programovaná hodnota K funkce v BCD tvaru jen v nepohybovém bloku.
PB_AX_E	DWORD	Programovaná hodnota E funkce v BCD tvaru, pokud není zařazeno zpřesňování kruhové interpolace (R325), jinak jen v nepohybovém bloku.

Změnové signály pro rozšířené rozhraní:

Byte:	bit 5.	bit 4.	bit 3.	bit 2.	bit 1.	bit 0.
PB_AX_ZM1	ZMS6PI	ZMS5PI	ZMS4PI	ZMSZPI	ZMSYPI	ZMSXPI
PB_AX_ZM2			ZMSEPI	ZMSKPI	ZMSJPI	ZMSIPI

Programované hodnoty souřadnic v BCD tvaru v double-wordových položkách **PB_AX_X**, **PB_AX_Y**,... je umožněno využívat v PLC programu. Praktický význam se získá tehdy, když PLC program využívá hodnotu souřadnice, která na stroji už není obsažena. Například když má stroj použity 4 souřadnice, tak PLC program může použít pátou souřadnici pro zadání double-wordových hodnot třeba pro zadání otáček druhého vřetene. V tomto případě se musí pátá souřadnice nastavit ve strojní konstantě R04 jako standardní osa s tím rozdílem, že ve třetí dekádě se nastaví neviditelnost a neaktivnost souřadnice (třetí dekáda R04 = 3).

Programované hodnoty funkcí I, J a K se do rozhraní pro PLC přepíší do double-wordových buněk **PB_AX_I**, **PB_AX_J** a **PB_AX_K**. Systém je do rozhraní pro PLC přepisuje jen pokud jsou nově programovány v nepohybových blocích.

Programovaná hodnota funkce E se do rozhraní pro PLC přepíše do double-wordové buňky **PB_AX_E**. Pokud není zařazeno zpřesňování kruhové interpolace (strojní konstanta R325 je nulová), systém vždy přepíše do rozhraní aktuální hodnotu funkce E. Pokud je zařazeno zpřesňování kruhové interpolace, systém přepíše funkci E do rozhraní pro PLC jen pokud je tato funkce nově programovaná v nepohybovém bloku.

Příklad:

V přípravních funkcích se snímá hodnota 6.souřadnice:

```

LDR    ZMS6PI
JL0    W_AX_NO
LOD    DWRD.PB_AX_6      ;sejme hodnotu 6. osy
BIN    DWRD              ;převod z BCD na BIN
STO    DWRD.BUN_AX_6

W_AX_NO:
```

8.5 Blok zpětného hlášení

Blok zpětného hlášení je zprávou z PLC programu pro CNC systém. Na rozdíl od povelového bloku se přenáší z PLC programu do CNC systému průběžně.

8.5.1 Přehled signálů bloku zpětného hlášení

V následující tabulce je uveden přehled signálů bloku zpětného hlášení.

Bity uvedené **tlustým písmem** jsou vyhrazeny jen pro ovládání z PLC programu.

Bity uvedené *kurzívou* používá supervizor interfejsu a pro PLC program mají maximálně informační význam.

PLC program nesmí zapisovat do bitů bloku zpětného hlášení, které jsou uvedeny kurzívou.

Bity uvedené v závorce jsou uvedeny jen pro kompatibilitu se staršími verzemi a při návrhu nového PLC programu se nepoužívají. V tomto návodu nebudou zvlášť popsány.

Význam signálů pro kanál PLC → CNC		
ST_RDY	Stroj připraven	1 = připravenost stroje
TOC_ENABLE	Povolení točítka	1 = točítko povoleno
PIS_RDY	Start PIS připraven	1 = připraven
AUTMAN_REQ	Požadavek na pom. ruční pojezdy	1 = požadavek na AUTMAN
ACK_AUTMAN	Potvrzení pomocných ručních poj.	1 = potvrzení AUTMAN
BLOK_STR	Blokování startu	1 = blokování
PO_F	Potvrzení M, T, P, H, O, S	1 = převzetí povelu
PO_POL	Potvrzení progr. polohy	1 = poloha dosažena
PO_STOP	Potvrzení stop	1 = stop
ZAV_FCE	Příznak závěreč. funkce	1 = budou závěreč. funkce
EXTPAN	Externí panel	1 = zařazen
FEED_OVR	Řízení procenta rychlosti	1 = zařazeno
PRODLEVA	Časová prodleva	1 = zařazena
VAZBA	Polohová vazba	1 = zařaz. pro všechny osy
BUFINRDY	Příjmový buffer RTM	1 = je volný
INPOS	Poloha v toleranci	1 = dosaženo
KRx0	Reference	1 = najeto do reference
KHx0	Koncový limitní spínač +	1 = najeto
KHx1	Koncový limitní spínač -	1 = najeto
ZPx0	Zpomalovací lim. spínač +	1 = najeto
ZPx1	Zpomalovací lim. spínač -	1 = najeto
STARTPI	Start od PI	1 = start
STOPPI	Stop od PI	1 = stop
PSEU_REQ	Pseudoreference	1 = nastavení
INDIK	Indikace	1 = zařazena
CLR_REF	Zrušení reference	1 = aktivace
USER_REQ	Uživatelská sekvence	1 = aktivace
MAN_REQ	Manuální řízení os	1 = aktivace
JOGx0	Ruční posuv v ose +	1 = posuv
JOGx1	Ruční posuv v ose -	1 = posuv
MPx	Dynamické povolení pohybu	1 = povoleno (nepoužívat)
MIKx	Inkr. jog v mikronech	1 = zadáno
ZPRx	Zpomalovací refer. spínač	1 = najeto

Záznamy kanálu PLC → CNC					
	název	bit 7. bit 3.	bit 6. bit 2.	bit 5. bit 1.	bit 4. bit 0.
0	BZH00	ST_RDY ACK_AUTMAN	TOC_ENABLE BLOK_STR	PIS_RDY PO_F	AUTMAN_REQ PO_POL
1	BZH01	ZAV_FCE PO_STOP	EXTPAN VAZBA	FEED_OVR BUFINRDY	PRODLEVA INPOS
2	BZH02	KR4	KRZ	KR6 KRY	KR5 KRX
3	BZH03	KH40	KHZ0	KH60 KHY0	KH50 KHX0
4	BZH04	KH41	KHZ1	KH61 KHY1	KH51 KHX1
5	BZH05	ZP40	ZPZ0	ZP60 ZPY0	ZP50 ZPX0
6	BZH06	ZP40	ZPZ1	ZP61 ZPY1	ZP51 ZPX1
7	BZH07	PIS_CONTROL INDIK	STARTPI CLR_REF	STOPPI USER_REQ	PSEU_REQ MAN_REQ
8	BZH08	MP4	MPZ	MP6 MPY	MP5 MPX
9	BZH09	POT4	POTMIK POTZ	POT6 POTY	POT5 POTX
0A	BZH10	POT_SGN4	POT_SGNZ	POT_SGN6 POT_SGNY	POT_SGN5 POT_SGNX
0B	BZH11	číslo erroru z PLC			
0C	BZH12	číslo erroru z interpolací			
0D	BZH13	číslo erroru z RTM			
0E	BZH14	ZPR4	ZPRZ	ZPR6 ZPRY	ZPR5 ZPRX
0F	BZH15	JOG4	JOGZ	JOG6 JOGY	JOG5 JOGX
10	BZH16	JOGM4	JOGMZ	JOGM6 JOGMY	JOGM5 JOGMX
11	BZH17	standardní výstupní port v panelu (TLOUT)			
12	BZH18	1. přídavný výstupní port v panelu (P1OUT)			
13	BZH19	2. přídavný výstupní port v panelu (P2OUT)			
14	BZH20	potvrzení převzetí tlačítka z MATTL			
15	BZH21	informační hlášení z PLC			
16	BZH22	podskupina chyby z PLC programu			
17	BZH23	3. přídavný výstupní port v panelu (P3OUT)			
18	BZH24	4. přídavný výstupní port v panelu (P4OUT)			
19	BZH25	5. přídavný výstupní port v panelu (P5OUT)			

8.5.2 Popis signálů bloku zpětného hlášení, které používá supervizor interfejsu

Bity v tabulce uvedené kurzívou používá supervizor interfejsu a pro PLC program mají maximálně informační význam. **PLC program nesmí zapisovat do bitů bloku zpětného hlášení, které jsou uvedeny kurzívou.**

PIS_RDYStart PLC připraven

Synchronizační signál pro supervisor dává na vědomí, že se může přesunout další povelový blok. PLC program jej nesmí nastavovat.

Nuluje se na začátku počátečních funkcí, do log.1 se nastaví po přesunu nového bloku z mezibufferu příjmu do aktivního bufferu.

PO_FPotvrzení funkcí

Bit **PO_F** hodnotou log.1 potvrzuje převzetí funkcí M, T, P, H, O, S a pohybu. Je řízen supervisorem interfejsu a na panelu systému se podle něj řídí druhá signálka: FUNKCE ROZPRACOVÁNY. Po vykonání všech funkcí, to znamená na konci bloku, bit **PO_F** je shozen do log.0. Při stopu, když blok není ukončen zůstává bit **PO_F** ve stavu log.1. PLC program může bitem **PO_F** zjišťovat, zda je stroj v klidu a podle toho řídit případné další akce.

PO_POLPotvrzení dosažení polohy

Bit **PO_POL** hodnotou log.1 potvrzuje, že byla dosažena zadaná poloha souřadnic. Bit nastavují moduly interpolace v kazetě systému. Bit je nahozen až po dosažení zadané tolerance polohy v 6. a 7. strojní konstantě. Po stopu pohybu se bit **PO_POL** nenastaví, protože naprogramovaná poloha nebyla dosažena.

PO_STOPPotvrzení stopu

Bit **PO_STOP** hodnotou log.1 potvrzuje vykonání stopu. Uživatelský PLC program smí tento bit pouze číst, nastavení provádí supervisor PLC programu.

Bit **PO_STOP** nabývá stavu log.1 jen tehdy, je-li proveden stop běžícího programu a ne je-li pouze stlačeno tlačítko STOP v době, kdy není co stopovat.

Stop může být vyvolán různým způsobem, například také signálem STOPPI z PLC programu.

Bit **PO_STOP** se po stopu dostane do stavu log.1 a v něm zůstane až po nový start systému, kdy se vrátí do stavu log.0. Nejedná se o kopírování tlačítka STOP.

Bit je pro PLC program vhodný pro zjištění, zda je systém ve stavu STOP .

BUFINRDYPříjmový buffer PLC programu

Bit **BUFINRDY** slouží pro synchronizaci přesunu dat povelového bloku z příjmového bufferu do pracovní paměti PLC rozhraní. Je-li v log.1, je příjmový buffer přenesen do pracovní paměti PLC rozhraní a příjmový buffer může být plněn novými daty.

Stav bitu **BUFINRDY** řídí supervizor interfejsu a uživatelský PLC program jej nesmí přepisovat.

Uživatelský program bit **BUFINRDY** nepotřebuje ani číst.

INPOSPoloha v toleranci

Je-li bit **INPOS** ve stavu log.1, je poloha os v toleranci. Bit je řízen programem interpolace, není-li v žádné ose polohová odchylka serva větší než je dovoleno pro jednotlivé osy ve strojních konstantách 06 a 07.

Je-li bit INPOS v log.0, svítí na panelu signálka INPOS (krajní signálka vpravo, označená vlnovkou).

Program interpolace nastavuje bit INPOS jednorázově po dosažení požadované polohy, to znamená, že bude-li některá osa "vytlačována" a vytlačena proti působení polohové vazby z dané polohy cizí mechanickou silou, bit INPOS nebude shozen do log.0.

Rozdíl mezi bity PO_POL a INPOS je ten, že bit INPOS netestuje, zda byla dosažena programovaná poloha. To znamená, že bit INPOS se nastaví do log.1 i při stopu pohybu, pokud byla dosažena tolerance polohové odchylky.

Bit INPOS je pro PLC program užitečný, když PLC program musí zjistit, zda se vykonává pohyb os. PLC program nesmí bit INPOS nastavovat.

PIS_CONTROL Řízení PLC

Bit **PIS_CONTROL** nastavuje supervizor interfejsu a potvrzuje stav PLC programu pro řízení PLC z programu. PLC program bit nesmí nastavovat a nepotřebuje jej ani číst.

MPx Povolení pohybu

Bity **MPx** jsou výsledkem všech podmínek povolení pohybu. Systémový software zapisuje do bitu MPx výsledek součinu povolení pohybu od různých zdrojů. Do tohoto součinu je zařazeno povolení pohybu např. od modulu interpolace, od modulu reálného času - kde jsou zařazené i koncové spínače (i softwarové) a od modulu servosmyčky. Do součinu jsou zahrnuty i bity MPxPI z BZH08PI, které umožňují blokovat pohyb z uživatelského PLC programu. Supervizor interfejsu umožní pohyb na základě nastaveného bitu MPx do log.1 a na základě ukončení průchodu modulem přípravných funkcí.

PLC program povoluje pohyb prostřednictvím signálů MPxPI. Problematika povolení pohybu byla popsána v podkapitole "Povolení pohybu" v úvodu této kapitoly. PLC program nesmí do bitů MPx zapisovat.

BZH12 Číslo chyby z interpolací BZH13 Číslo chyby z RTM

PLC program může zjistit čísla chyb z interpolací a z reálného času řízení servosmyčky. Jedná se například o dosažení koncových spínačů (i softwarových), o chyby snímačů (chyba kontrolních čítačů IRC, chyba fáze, přetečení dif.čítače atd.)

PLC program může tyto hodnoty pouze číst.

ACK_AUTMAN Potvrzení AUTMAN

Bit **ACK_AUTMAN** hodnotou log.1 signalizuje, že jsou požadovány a kazetou byly potvrzeny pomocné ruční pojezdy (AUTMAN). Hodnota log.1 trvá po celou dobu aktivace pomocných ručních pojezdů. Podrobněji je o této problematice pojednáno v kapitole "Pomocné ruční pojezdy".

8.5.3 Důležité řídicí funkce bloku zpětného hlášení

ST_RDY

....Stroj připraven

Bit **ST_RDY** hodnotou log.1 hlásí CNC části systému připravenost stroje. Po startu systému je automaticky nastaven do log.1

PLC program tento bit může shodit, jestliže se vyskytne závažná chyba.

Je-li Tento bit **ST_RDY** shozen do log.0, dojde k přerušení programu a shoení všech dekodovaných funkcí, mimo bitu M41PI, M42PI, M43PI, M44PI pro řazení převodových stupňů vřetene. Je-li **ST_RDY** v log.0, nelze odstartovat blok ani program.

Je-li **ST_RDY** ve stavu log.0, svítí signálka STROJ na ovládacím panelu. Bit **ST_RDY** je jediným zdrojem ovládání signálky Stroj.

V případě, že PLC program shodí bit **ST_RDY** do log.0, je vhodné na obrazovku vypsat hlášení o příčině.

Bit **ST_RDY** se nastaví do stavu log.1 v rámci instrukce **PIS_INIT** a **PIS_CLEAR**. Je-li z nějaké příčiny potřeba nahazovat tento signál až od splnění určitých podmínek, provede se to tak, že do modulu **PIS_INIT** se vloží instrukce, která tento signál shodí a nahození se provede v modulu **PROVOZ_VYSTUP**.

BLOK_STR

....Blokování startu

Je-li bit **BLOK_STR** nastaven do log.1, nelze na systému nic odstartovat. Při pokusu o start se hlásí chyba "ERROR 9.20 - PLC má blokování startu".

Stav bitu **BLOK_STR** se testuje při naměřené hraně startu. Blokování se týká všech existujících způsobů startu. Bit **BLOK_STR** se využije například při ošetření méně závažných chyb, jako je například zanesení filtru, při kterých je možné ještě dokončit rozpracovaný blok.

Po startu systému je bit **BLOK_STR** nastaven do log.0 a tak bit **BLOK_STR** není třeba programem PLC ošetřit.

ZAV_FCE

....Závěrečné funkce

Je-li bit **ZAV_FCE** nastaven do log.1, provede se modul **ZAVRECNE_FUNKCE**.

Je-li v bloku programována závěrečná funkce, kterou je třeba provést až po dokončení pohybu, je potřeba v modulu **PRIPRAVNE_FUNKCE** nahodit bit **ZAV_FCE** do log.1. Na základě stavu log.1 v bitu **ZAV_FCE** bude supervisor PLC programu volat modul **ZAVRECNE_FUNKCE**. V modulu **ZAVRECNE_FUNKCE** se pak zajistí provedení této funkce. Bit **ZAV_FCE** je shozen do log.0, je-li dokončen modul **ZAVRECNE_FUNKCE**. Shoení bitu **ZAV_FCE** provede supervisor PLC programu.

EXTPAN

....Externí panel

Je-li bit **EXTPAN** nastaven do log.1, je blokováno tlačítko START na ovládacím panelu.

Je-li systém vybaven externím ovládacím panelem, může se využít bit **EXTPAN** na blokování možnosti startu z hlavního ovládacího panelu. Start se pak provede zapsáním pulsu do bitu **STARTPI** nebo je možno start provést vysláním sekvence kódu tlačítek pomocí instrukce **SEKV**, **SEKV_END**.

Bit **EXTPAN** je možno ovládat přepínačem, který přepíná mezi hlavním ovládacím panelem a pomocným ovládacím panelem. Bit **EXTRN** nemá v systému žádný další účinek než blokování startu od fyzického tlačítka START na hlavním ovládacím panelu.

Předávání kódu stisknutého tlačítka do byte **MATTL** není stavem bitu **EXTPAN** ovlivněno.

Při pokusu o start fyzickým tlačítkem START na hlavním ovládacím panelu v době, kdy je bit **EXTPAN** nastaven do log.1, je hlášena chyba.

FEED_OVR**....Řízení procenta rychlosti z PLC**

Bit **FEED_OVR** hodnotou v log.1 provede odstavení potenciometru %F na hlavním ovládacím panelu a informaci o nastaveném procentu rychlosti přebírá z bytů BZH09 a BZH10.

Do bytů BZH09 a BZH10 se zapisuje promile rychlosti ve formátu šestnáctibitového slova BIN.

Osa pojede programovanou rychlostí, bude-li v BZH09 a BZH10 hodnota 1000.

Bit FEED_OVR se využije při dálkovém ovládání, například z pomocného ovládacího panelu.

Systém nekontroluje hodnoty zapsané do BZH09 a BZH10, souřadnice však nepojede v žádné ose větší rychlostí než je zadaná ve strojních konstantách pro rychloposuv : 10 až 15.

Systém udržuje tři proměnné rychlosti :

- 1) pro režim AUT,RUP - inicializace hodnotou 0
- 2) pro režim MAN - inicializace ve strojní konstantě 54
- 3) pro režim AUT-AVP - inicializace ve strojní konstantě 54

V jednotlivých režimech je údaj o rychlosti vypočten vynásobením příslušné proměnné s hodnotou od potenciometru.

Způsob reakce při požadavku na rychloposuv je volen pátou dekadou strojní konstanty 97 (viz kapitola "Popis strojních konstant"):

- ♦ 0 - Potenciometr %F se neuplatní, tlačítkem 10% lze přepínat mezi 100% a 10% rychloposuvu. Tlačítko má přepínací funkci. Je-li nastaven rychloposuv na 10%, je tento stav indikován na obrazovce textem "10%".
- ♦ 1 - Tlačítko 10% zapíná nebo vypíná uplatnění potenciometru %F při požadavku na rychloposuv. Tlačítko 10% má přepínací funkci. V případě, že je potenciometr zařazen, je tento stav indikován na obrazovce textem "%G0".
- ♦ 2 - Tlačítko 10% nemá význam, potenciometr 10% má trvalou funkci i při rychloposuvu, na obrazovce je tento stav indikován zobrazením textu "%G0".

V systému ještě existuje samostatný režim POTENC, na který se tento popis nevztahuje.

Příklad:

Řízení rychlosti z PLC programu.

FL	1,FEED_OVR	;nastavení požadavku na řízení rychlosti z PLC
LOD	PROMILE	;promile požadované rychlosti
STO	WORD.BZH09	;zápis do BZH09 a BZH10

PRODLEVA**....Signálka prodleva**

Nastavení bitu **PRODLEVA** do log.1 způsobí rozsvícení signálky "PRODLEVA" na ovládacím panelu. Signálka "PRODLEVA" se rozsvítí rovněž, je-li programována časová prodleva pomocí G04.

Signálka prodleva je na monitoru umístěna třetí zleva s obrázkem hodinek.

Jestli PLC program bit PRODLEVA využívá, je lépe jím blikat, aby jej bylo možno rozlišit od programování G04. Bit PRODLEVA je možné využít například při počátečním promazání stroje po zapnutí napájení, aby obsluha nebyla zmatená a netropila zbytečný povyk.

VAZBA**....Softwarová vazba a odměřování**

Bit **VAZBA** hodnotou log.1 zabezpečí uzavření polohové vazby v osách. Po startu systému se nahodí do log.1, stane se tak při výkonu instrukce PIS_INIT a PIS_CLEAR.

Je-li bit VAZBA v log.0, polohová vazba ve všech osách není uzavřena, systém nevydává výstupní napětí

pro řízení pohonu, ale ani nesnímá signály z odměřování.

Když je potřeba z nějakých příčin polohovou vazbu os uzavřít až později, například až po promazání, vloží se do modulu PIS_INIT instrukce, která bit shodí opět do log.0 a nahození do log.1 se provede v modulu PROVOZ_VYSTUP po splnění požadovaných podmínek.

PLC program by měl blokovat pohon a případně upnout osu, když není uzavřena polohová vazba, aby osa neujela driftem.

Když je potřeba zrušit vazbu a přitom zachovat funkci odměřování, použije se bit INDIK.

Když je potřeba zrušit polohovou vazbu, ale zachovat funkci odměřování pro jednotlivé osy, použije se bit VAZBA_x v byte VAZBAPI.

STARTPI

....Start z PLC programu

Bit **STARTPI** hodnotou log.1 vyvolá START obdobně jako stisknutí tlačítka START na ovládacím panelu. Nastavením bitu STARTPI se vyvolá start například výkonu partprogramu. Systém reaguje na nástupní hranu signálu.

Stav bitu STARTPI je potřeba v PLC programu vrátit do stavu log.0, to se provede například odčasnáním. Minimální délka pulsu signálu STARTPI je asi 200 ms.

Nejedná se o dálkové zmáčknutí tlačítka, ale o vyvolání činnosti, které toto tlačítko vyvolává. Nastavení bitu STARTPI se tedy nepromítne do proměnné MATTL jako stisknutí tlačítka na ovládacím panelu.

STOPPI

....Stop z PLC programu

Bit **STOPPI** hodnotou log.1 vyvolá STOP obdobně jako stisknutí tlačítka STOP na ovládacím panelu. Nastavením bitu STOPPI se vyvolá stop celého bloku. Systém reaguje na nástupní hranu signálu.

Stav bitu STOPPI je potřeba v PLC programu vrátit do stavu log.0, to se provede například odčasnáním. Minimální délka pulsu signálu STARTPI je asi 200 ms.

Nejedná se o dálkové zmáčknutí tlačítka, ale o vyvolání činnosti, které toto tlačítko vyvolává. Nastavení bitu STOPPI se tedy nepromítne do proměnné MATTL jako stisknutí tlačítka na ovládacím panelu.

V případě, že bude bit STOPPI nastaven na log.1 a bude nastaven také STARTPI, systém sice provede START bloku, ale vzápětí provede STOP bloku.

Signál STOPPI používá PLC program, když nastane v průběhu vykonávání bloku chyba. Například při chybě v přípravných funkcích se zastaví další chod pomocí "nekonečné instrukce typu EX", nastaví se bit STOPPI a supervizor interfejsu přeruší vykonávání bloku.

Příklad:

Ošetření chyby v přípravných funkcích:

LDR	TLAK_I		;načte vstup o dosažení tlaku
TEX0	CITAC_TL,CAS_ERROR,ERROR,15h		;časová kontrola tlaku
.....			
.....			
ERROR:	STO	BZH11	;nastavení chybového hlášení
FL	1,STOPPI		;STOP z PLC programu
LDR	JEDNA		;"nekonečný stav"
EX0			;zabránění dalšímu výkonu
			;přípravných funkcí

PSEU_REQ**....Nastavení pseudoreference z PLC**

Bit **PSEU_REQ** hodnotou log.1 provede nastavení pseudoreference. Informace o tom, ve které ose se má provést pseudoreference, je předána pro tento účel v BZH02 pomocí bitů KRx.

Stav bitu PSEU_REQ je potřeba v PLC programu vrátit do stavu log.0, to se provede například odčasnáním. Minimální délka pulsu signálu je asi 200 ms.

Příklad:

Nastavení pseudoreference v ose X prostřednictvím mechanismu.

FL	1,KRX	;příznak osy
FL	1,PSEU_REQ	;nastavení požadavku na pseudoreferenci
TIM	CITAC_RF,D200	;čas asi 200 ms
FL	0,PSEU_REQ	;vynulování akce pseudoreference
FL	0,KRX	;

CLR_REF**....Vynulování reference z PLC**

Bit **CLR_REF** hodnotou log.1 zruší referenci v daných osách.

Je-li v bitu CLR_REF log.1, dojde ke zrušení reference v těch osách, pro které je v BZH02 v bitech KRx nastavena log.1

Stav bitu CLR_REF je potřeba v PLC programu vrátit do stavu log.0, to se provede například odčasnáním. Minimální délka pulsu signálu je asi 200 ms.

Zrušení reference v ose se provede například při přepnutí rotační souřadnice na vřeteno.

Příklad:

Zrušení reference pomocí bitu "CLR_REF" v logickém sekv. celku:

FL	1,KRY	;nastavení pro osu Y
FL	1,CLR_REF	;zrušení reference pro osu Y
TIM	CITAC_doba	;čas cca 200 ms
FL	0,KRY	;
FL	0,CLR_REF	;shození povelu

USER_REQ**....Uživatelská sekvence**

Bit **USER_REQ** hodnotou log.1 zabezpečí vyslání sekvence stisku tlačítek panelu, režimů a formátů..

PLC progrm může simulovat stisky tlačítek na ovládacím panelu, nastavovat režimy, řídit obrazovky a menu, pomocí instrukcí SEKV a SEKV_END (popsané v kapitole "Způsoby dálkového ovládání systému").

Uživatelský PLC program běžně nepotřebuje tento bit nastavovat, neboť to zajistí instrukce SEKV_END. Program PLC využije tento bit jedině v případě, je-li sekvence vysílána dynamicky, pro testování dalšího vysílání.

Hodnota log.1 způsobí zahájení odesílání bufferu, v kterém je připravena řídicí sekvence do ovládacího panelu. Po odeslání celého bufferu komunikační program vrátí bit USER_REQ zpět do log.0. Změna stavu z log.1 do log.0 v bitu USER_REQ znamená, že sekvence byla odeslána do panelu, ale ne že byla již provedena.

MAN_REQ**....Manuální řízení os**

Bit **MAN_REQ** slouží pro dálkové ovládání pohybu os v režimu MAN. Je-li bit **MAN_REQ** nastaven do log.1, probíhá pohyb ve zvolené ose.

Osa, která bude vykonávat pohyb, se určí nastavením jedno z bitu **JOGx** a **JOGMx** v **BZH15** a **BZH16** do log.1. Je zakázáno v **BZH15** a **BZH16** nastavit do log.1 více bitů pro současný pohyb více os. Log.1 v **BZH16** určuje záporný směr pohybu pro danou osu.

Řízení rychlosti pohybu je možno provádět pomocí dálkového ovládání potenciometru %F. Dálkové ovládání potenciometru %F se vyvolá nastavením bitu **FEED_OVR** do log.1, pak byte **BZH09** a **BZH10** představují binární zadání promile rychlosti.

Bit **MAN_REQ** se využije například, je- i stroj vybaven přídatným ovládacím panelem, z kterého je potřeba ručně řídit pohyb.

Příklad:

Nastavte ruční řízení posuvu v ose Z v záporném směru.

FL	1,MAN_REQ	;požadavek na ruční řízení
FL	1,JOGMZ	;pohyb v ose -Z

INDIK**....Režim indikace**

Bit **INDIK** hodnotou log.1 umožní přepnout celý systém do režimu indikace.

V režimu indikace se na analogové výstupy pro servopohony nevysílá napětí. Indikace polohy na obrazovce zobrazuje stav čítače odměřování a ne polohu požadovanou, jak je tomu ve standardním režimu systému.

V režimu **INDIKACE** může řídit pohyb souřadnic přímo stroj nebo PLC program pomocí instrukcí **ANALOG** (popsaných v části "Nastavení parametrů servopohonů..").

Podobně je problematika přepínání do indikace popsána v kapitole "Přepínání režimů **INDIKACE-NC**".

POTx**....Řízení pomocí potenciometrů****POT_SGNx****....Nastavení směru pro potenciometry**

Zapsáním log.1 do bitu **POTx**, řídí PLC program pohyb v dané ose v režimu potenciometrů. V případě nastavení externího řízení %F pomocí bitu **FEED_OVR**, buňky **BZH09** a **BZH10** slouží pro zadání promile požadované rychlosti (viz bit **FEED_OVR**).

Režim řízení pohybu potenciometry umožní CNC systému simulovat ruční řízení stroje. Ovládání os potenciometry je obdobné, jako by na vstupy pohonu pro řídicí napětí byly připojeny potenciometry pro ruční zadání rychlosti. Systém však udržuje řízené souřadnice v polohové vazbě.

Při řízení potenciometry je možné vyvolat pohyb současně ve více osách. Směr pohybu je určen znaménkem. Znaménko se zadává pro jednotlivé osy do bitu **POT_SGNx**. Je-li v **POT_SGNx** log.1, je nastaven záporný směr pohybu.

K systému je možno připojit max. 4 potenciometry, pro každou osu se připojuje jeden potenciometr a tedy je možno pomocí potenciometru řídit max 4 osy systému. Pro připojení potenciometru je třeba do ovládacího panelu osadit přídatnou kartu vstupu a výstupu a provést nastavení strojních konstant 68 a 69. Strojní konstanta 68 nastavuje velikost rychlosti pracovního a mikro posuvu jednotlivých os při plném vytočení potenciometru. Strojní konstanta 69 nastavuje masku a citlivost potenciometru.

PLC program pomocí buňky s bitovou strukturou **BUN_JOGMIK** řídí, která rychlost má být aktuální. Podrobně

je tato problematika popsána v kapitole "Řízení posuvu pomocí potenciometrů".

JOGx
JOGMx

....Externí řízení pohybu
....Externí řízení pohybu v záporném směru

Bit **JOGx** hodnotou log.1 povoluje pohyb v kladném směru při externím řízení pohybu, to je při nastavení bitu MAN_REQ do log.1.

Bit MAN_REQ slouží pro dálkové ovládání pohybu os v režimu MAN. Je-li bit MAN_REQ nastaven do log.1, probíhá pohyb ve zvolené ose. Osa, která bude vykonávat pohyb, se určí nastavením jedno z bitů JOGx a JOGMx v BZH15 a BZH16 do log.1. Je zakázáno v BZH15 a BZH16 nastavit do log.1 více bitů pro současný pohyb více os. Hodnota log.1 v bitu JOGMx určuje záporný směr pohybu pro danou osu.

Příklad pro řízení os je při vysvětlení bitu MAN_REQ a v kapitole "Způsoby dálkového ovládání systému".

AUTMAN_REQ

....Požadavek na AUTMAN

Požadavek na aktivaci pomocných ručních pojezdů. Hodnota log.1 v bitu AUTMAN_REQ aktivuje pomocné ruční pojezdy, pokud jejich aktivace je v systému povolena. PLC program se o aktivaci pomocných ručních pojezdů může přesvědčit pomocí signálu ACK_AUTMAN. Když PLC program vyžaduje aktivaci pomocných ručních pojezdů, musí držet bit AUTMAN_REQ na hodnotě log.1 po celou dobu požadované aktivace. Pokud byly pomocné ruční pojezdy aktivovány z PLC programu a bit AUTMAN_REQ se shodí no hodnotu log.0, budou pomocné ruční pojezdy deaktivovány. Podrobněji je o této problematice pojednáno v kapitole "Pomocné ruční pojezdy".

8.5.4 Limitní, zpomalovací a referenční spínače

PLC program může dát systému informaci o těchto spínačích:

- ♦ **limitní spínače** - jedná se o koncové spínače, na základě kterých systém dojezdovou rampou zastaví pohyb (KHx0, KHx1)
- ♦ **limitní zpomalovací spínače** - jedná se o spínače, na základě kterých systém zpomalí rychlost na dojezdovou, danou strojní konstantou 51 (ZPx0, ZPx1)
- ♦ **referenční spínače** - jedná se jen o režim nájezdu do reference. Při referenčním spínači už musí být posuv zpomalen na dojezdovou rychlost, danou strojní konstantou 51 a od tohoto spínače se vyhledává nulový puls se snímače (KRx)
- ♦ **zpomalovací referenční spínače** - jedná se jen o režim nájezdu do reference. Od tohoto spínače systém zpomalí pohyb na dojezdovou rychlost, danou strojní konstantou 51 (ZPRx).

KHx0
KHx1

....Limitní spínače v kladném směru
....Limitní spínače v záporném směru

Hodnota log.1 v **KHx0** nebo v **KHx1** způsobí zastavení pohybu v dané ose. Zastavení řídí interpolátor dojezdovou rampou podle strojní konstanty 52.

Je-li v bitu KHx0 nebo KHx1 log.1, vypíše se na monitoru hlášení "ERROR: Osa x najela na limitní koncový spínač v kladném/záporném směru".

Vyjetí z limitního spínače je možné pouze v režimu MAN. V režimu AUT a RUP způsobí najetí na jakýkoliv limitní koncový spínač zastavení pohybu. Jestliže by se signál KHx0 nebo KHx1 opět vrátil do stavu log.0, pohyb souřadnice bude pokračovat (není samodrž). Případnou samodrž musí zabezpečit PLC program (viz ještě obrázek u zpomalovacích spínačů).

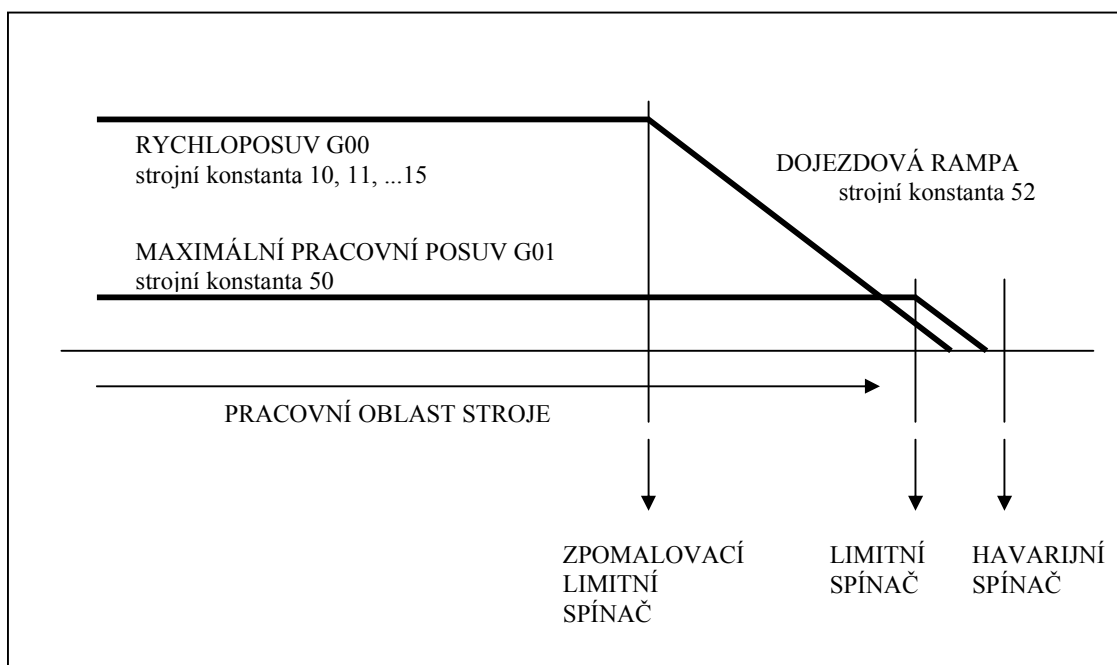
ZPx0
ZPx1

....Zpomalovací limitní spínač
....Zpomalovací limitní spínač

Hodnota log.1 v bitech **ZPx0** a **ZPx1** způsobí přechod na zpomalovací rampu zadanou ve strojní konstantě 52, až pokud nebude dosažena velikost zpomalovacího posuvu zadaného strojní konstantou 51. CNC systém u zpomalovacích limitních spínačů nevyhodnocuje směr. Když je potřeba vyhodnocovat směr pohybu, musí to provést PLC program na základě signálů SM_POxPI.

Zápis do bitu rozhraní ZPx0 (limitní zpomalovací koncový spínač) a do bitu rozhraní ZPRx (zpomalovací referenční koncový spínač) lze odvozovat z jednoho fyzického koncového spínače umístěného na ose stroje.

Signál z fyzického koncového zpomalovacího spínače je do rozhraní vhodné předávat pouze při zařazení rychloposuvu. Tímto řešením lze dosáhnout zvětšení pracovního prostoru stroje až po limitní koncové spínače. Výhoda tohoto řešení se uplatní zejména u strojů s vysokým rychloposuvem.



KRx
ZPRx

....Referenční spínače
....Zpomalovací referenční spínače

Signál **KRx** informuje systém o najezení na referenční koncový spínač v příslušné ose. Je-li systém v režimu REF a je-li do bitu KRx zapsaná hodnota log.1, přejde systém do očekávání příchodu nulového pulsu od odměřování. Předpokládá se, že předcházelo zpomalení pomocí signálu **ZPRx**.

PLC program musí zajistit, aby během nájezdu do referenčního bodu došlo k nastavení bitu KRx do stavu log.1. PLC program zpravidla nastaví bit KRx do stavu log.1 od najezení na fyzický referenční koncový spínač.

Signály od koncových spínačů os jsou zavedeny na libovolné PLC vstupy. PLC program je zpracovává a přenese do rozhraní na bity KRx a ZPRx. Zpracování signálů spínačů os PLC programem umožňuje vyhodnocování různých variant a umístění spínačů os. Je možné i simulovat spínače pomocí časových členů v PLC programu.

Postup nájezdu do referenčního bodu :

Po odstartování nájezdu do referenčního bodu se osa rozjede směrem k referenčnímu koncovému spínači. Najezením na referenční zpomalovací spínač **ZPRx** se posuv v ose zpomalí. Po dosažení zpomalené rychlosti se očekává nájezd na referenční spínač. Po nájezdu na referenční spínač se pokračuje stejnou rychlostí a očekává se příchod nulového pulsu odměřování. Po příchodu nulového pulsu se nastaví čítač odměřování na nulový bod stroje podle strojních konstant a zastaví se posuv osy.

Směr nájezdu do referenčního bodu je pro jednotlivé osy určen znaménkem ve strojních konstantách 00 až 05. Rychlost nájezdu na zpomalovací referenční spínač je pro jednotlivé osy zadána ve strojních konstantách 10 až 15 - "Rychloposuv". Je-li rychlost nájezdu rychloposuvem příliš velká, je možno ve strojních konstantách 10 až 15 - "Rychloposuv" v dekadách 6,7,8 zadat procento z rychloposuvu. Rampa zpomalení po nájezdu na zpomalovací referenční spínač je zadána ve strojní konstantě 52 - "zrychlení". Rychlost, na kterou se zpomaluje, je zadána ve strojní konstantě 51 - "Zpomalovací Posuv".

Není-li na stroji k dispozici fyzicky referenční koncový spínač, je potřeba tento signál do rozhraní do bitu KRx odvodit například časově od nájezdu na referenční zpomalovací koncový spínač. Jiná možnost je odvození signálu KRx od ujeté dráhy od zpomalovacího spínače (viz příklad v kapitole "Užitečné příklady pro PLC program").

Ztotožnění zpomalovacího koncového spínače referenčního a zpomalovacího koncového spínače limitního je dovoleno.

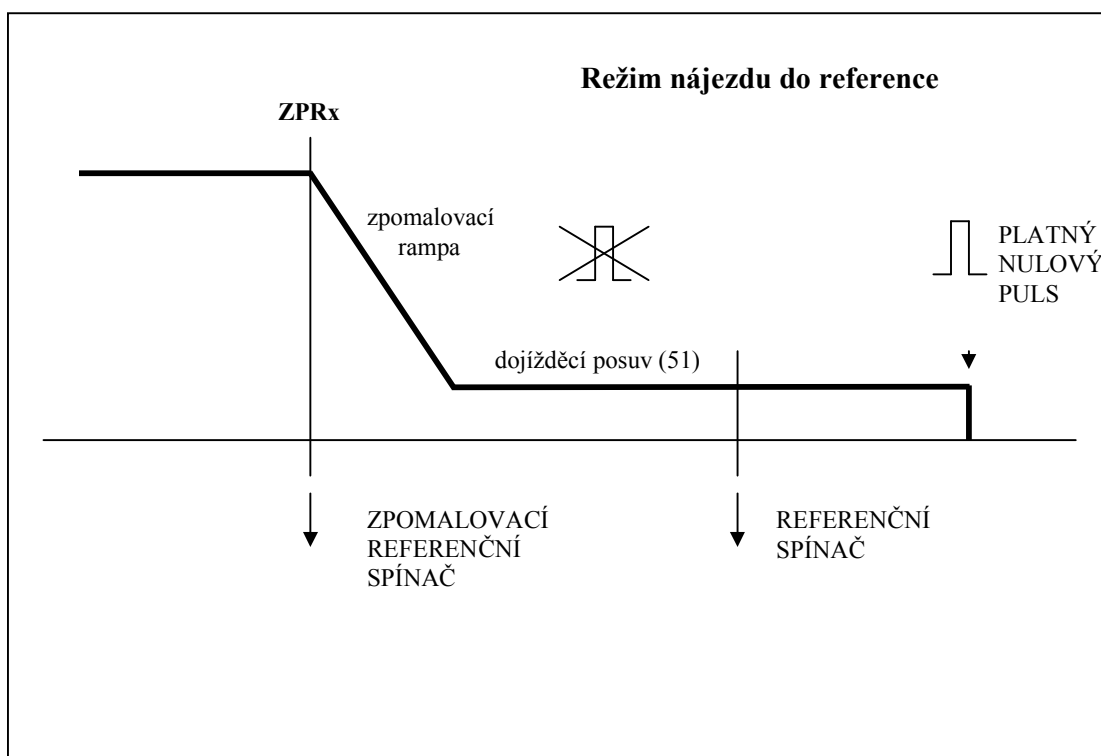
Není-li režim nájezdu do reference nebo neprobíhá-li proces pseudoreference, tzn. není nastaven bit PSEU_REQ, systém nereaguje na signály v bitech ZPRx a KRx.

Zrušení reference je možno provést signálem CLR_REF. Signály zapisované do ZPRx a do KRx mohou mít charakter pulsu, protože systém je vyhodnocuje paměťově.

Příklad:

Řízení reference od jednoho fyzického spínače

LDR	REF_X_I	;vstup od spínače
LO	ZPRX	;přidrží od nastavení zpomalovacího ref. spínače
LA	REFPI	;blokování režimem reference
WR	ZPRX	;zápis do zpomalovacího referenčního spínače
LOD	CAS_REF	;doba pro zabrzdění na dojížděcí posuv
TM	CITAC_REF	;časování
WR	KRX	;nastavení referenčního spínače



8.5.5 Bajtové hodnoty v bloku zpětného hlášení

BZH11Číslo chyby z PLC

Do buňky **BZH11** nastavuje PLC program chyby v BCD tvaru. Podrobnější popis problematiky chybových hlášení, přiřazení textů k jednotlivým chybám j v kapitole "Chybová hlášení, varování a informační hlášení z PLC programu."

BZH22Číslo podskupiny chyby z PLC

Do buňky **BZH22** nastavuje PLC program podskupinu chyby v BCD tvaru. Podrobnější popis problematiky chybových hlášení, přiřazení textů k jednotlivým chybám j v kapitole "Chybová hlášení, varování a informační hlášení z PLC programu."

BZH21Informační hlášení z PLC programu

Do buňky **BZH21** nastavuje PLC program informační hlášení v binárním tvaru. Podrobnější popis problematiky informačních hlášení, přiřazení textů k jednotlivým informacím je v kapitole "Chybová hlášení, varování a informační hlášení z PLC programu."

BZH20 Potvrzení převzetí tlačítka

V buňce **BZH20** je potvrzení převzetí kódu tlačítka předávaného v buňce MATTL.

Do buňky MATTL jsou předávány ASCII kódy standardních tlačítek ovládacího panelu systému. Pro potvrzení převzetí kódu tlačítka z MATTL překopíruje PLC program obsah buňky MATTL do buňky BZH20. Je-li převzetí kódu tlačítka potvrzeno je buňka MATTL nulována. Systém nevytváří frontu nepřevzatých tlačítek pro PLC. Podrobně je tato problematika popsána v kapitole "Snímání tlačítek z panelu systému CNC836 do PLC".

BZH17Standardní výstupní port v panelu (TLOUT)
BZH181. přídavný výstupní port v panelu (P1OUT)
BZH192. přídavný výstupní port v panelu (P2OUT)
BZH233. přídavný výstupní port v panelu (P3OUT)
BZH244. přídavný výstupní port v panelu (P4OUT)
BZH255. přídavný výstupní port v panelu (P5OUT)

PLC program může využívat i vstupy a výstupy, které jsou připojeny do panelu systému. Na standardní desce pro snímání tlačítek systému (INOUT02) je k dispozici 8 výstupních bitů ovládaných prostřednictvím **BZH17**. Na přídavné desce (INOUT02, INOUT07) jsou k dispozici 2 výstupní porty, ovládané prostřednictvím **BZH18** a **BZH19**. Další tři výstupní porty jsou k dispozici na další přídavné desce INOU07. Tato problematika je podrobně rozepsána v kapitole "Přídavné vstupy a výstupy v panelu systému CNC836. Panel stroje zapojen na přídavné vstupy a výstupy."

