

20

20. RUČNÍ POJEZDY

20.1 Všeobecný popis

Ruční pojezdy není samostatný režim systému. Pomocné ruční pojezdy se takto jeví jako okamžitý přechod do režimu MAN, ale bez změny režimu. Z toho je patrné, že nejvýznamnější využití pomocných ručních pojezdů bude v režimech AUT, AUT po stopu, AUT – BB. Využije se ale i rychlé operativní popojíždění například v režimu CA (centrální anulace).

Pro pomocné ruční pojezdy budeme používat označení: „**AUTMAN**“.

Při ukončení pomocných ručních pojezdů ve stopnutém režimu AUT nebo v režimu AUT -BB máme na výběr tyto způsoby :

- Proveďte se návrat na dráhu v režii pomocných ručních pojezdů a automatický režim po opětovném startu pokračuje přesně podle programované dráhy.
- Pomocné ruční pojezdy se použijí jen pro odjetí z místa stopu (například s točícím se vřetenem). Potom následuje **centrální anulace** a opětovný start programu pomocí **volby bloku**, nebo zrychlenou volbou **cont**.

Pro řízení ručních pojezdů možno využít sadu systémových příkazů „**Command**“ a sadu příkazů pro reálný čas „**RtmCommand**“, které jsou přímo definované v definičních souborech klávesnic typu „**KbdConfig**“. Popis příkazů a jejich použití je popsáno v kapitole: „**Ovládací panely**“.

V této kapitole bude popsán i způsob nezávislého posouvání dráhy, tzv režim „**SHIFT**“.

20.2 Bitové signály pro PLC program

PLC program má pro pomocné ruční pojezdy a nezávislé posouvání dráhy k dispozici tyto bitové signály:

20.2.1 Základní bitové signály

Základní bitové signály pro pomocné ruční pojezdy slouží pro PLC program na určení pohybu, směru pohybu, povolení pohybu, návratu do místa stopu a pod.

- Bit **ACK_AUTMAN** definovaný v BZH00. Hodnota log.1 signalizuje, že je požadován a kazetou byly potvrzeny pomocné ruční pojezdy (AUTMAN). Hodnota log.1 trvá po celou dobu aktivace pomocných ručních pojezdů.
- Bity **PO_OSxPI** definované v PB20PI se dynamicky nahazují na hodnotu log.1 při požadavku na pohyb v dané ose a při ukončení pohybu přejdou na hodnotu log.0. Bity se nahodí na hodnotu log.1 už při požadavku na pohyb i v případě, že je pohyb zakázán například od signálu MPxPI. Tyto bity proto může PLC program využít na zapínání vazby, řazení spojek a uvolňování os (pokud se nutno provádět) a až po provedení akce povolit příslušný pohyb. Bity PO_OSxPI není možno využít pro řízení uvolňování os tehdy, když požadujeme pohyb od ručního točítka bez předchozího pohybu v dané souřadnici. Na řízení uvolňování os pro točítka je možno využít bitů AUTMAN_CONT_SELECT a MM_x0, jak o tom bude popsáno dále.
- Bity **SM_POxPI** definované v PB20PIS určují směr pojezdu a jsou platné v okamžiku nahození signálů PO_OSxPI ještě před startem pohybu. Hodnota log.0 určuje kladný směr a hodnota log.1 určuje záporný směr pohybu. Tyto bity možno využít například pro řazení směrových spojek.
- Bity **MPxPI** definované v BZH08PI mohou sloužit pro povolování pohybu stejně jako u standardních pohybů systému. Podrobně je jejich význam popsán v PLC návodu kapitole “Důležité bitové proměnné pro PLC program”. Bity MPxPI slouží pro povolování pohybu jen v případě, když 4.dekáda strojní konstanty R233 je nastavena na hodnotu 1. Tehdy na povolení pohybu bity MPxMAN nemají vliv.
- Bity **MPxMAN** definované v BZH08MAN 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. V tomto případě povolení pohybu nezávisí od bitů MPxPI, ale jen od povolovacích bitů MPxMAN určených pro pomocné ruční pojezdy.
- Bit **INPOS_STOP** definovaný v EXT_CONT_AUTMAN3 svou hodnotou log.1 určuje, že systém je v poloze posledního stopu. Signál je definovaný jen v režimu AUT a ve všech jiných režimech má hodnotu log.1. Při pomocném ručním pojezdu v libovolné ose se signál INPOS_STOP nastaví na hodnotu log.0. Pomocí tohoto signálu může PLC program například blokovat opětovný START programu, pokud nebudou všechny souřadnice v poloze po stopu. Signál se nastaví na hodnotu log.1 opět tehdy, když bude proveden návrat na dráhu (v režii pomoc.ručních pojezdů) pro všechny osy.
- Bit **EN_AUTMAN** definovaný v BZH08MAN svou hodnotou log.1 povoluje pomocné ruční pojezdy. Bit je přednastaven na hodnotu log.1 při startu systému a dál je plně k dispozici pro PLC program. PLC program může pomocí tohoto bitu zakázat aktivaci pomocných ručních režimů.
- Bit **REQ_ESHIFT** je nastaven na hodnotu „1“ po celou dobu zvoleného režimu nezávislého posouvání dráhy „SHIFT“. PLC program může číst i nastavovat (viz dále).
- Bit **ACT_ESHIFT** je nastaven na hodnotu „1“ po celou dobu aktivace posouvání dráhy „SHIFT“. PLC program může číst i nastavovat (viz dále).

20.2.2 Informační signály

Informační bitové signály jsou určeny jen pro čtení. PLC program si může na základě nich zjistit způsob řízení pomocných ručních pojezdů z panelu systému nebo z panýlku točítka.

- Bit **AUTMAN_CONT_NC** definovaný v `MAN_NC` svou hodnotou log.1 informuje PLC program, že řízení pomocných ručních pojezdů se provádí z panelu systému.
- Bit **AUTMAN_CONT_TOC** definovaný v `CONT_AUTMAN2` svou hodnotou log.1 informuje PLC program, že řízení pomocných ručních pojezdů se provádí z panýlku točítka.
- Bit **AUTMAN_CONT_SELECT** definovaný v `CONT_AUTMAN` svou hodnotou log.1 informuje PLC program, že se provádí předvolba pohybu pro pomocné ruční pojezdy z panelu systému nebo z panýlku točítka. Například, když se na panýlku točítka stiskne souřadnice, pohyb se nevykoná, ale příslušná souřadnice se předvolí.
- Bity **MM_x0** a **MM_x1** definované v `AX_AUTMAN` informují PLC program o řízení ručních pojezdů. Bity `MM_X0`, `MM_Y0`, . . . ,`MM_60` určují požadavek na pohyb v kladném směru a bity `MM_X1`, `MM_Y1`, . . . ,`MM_61` určují požadavek na pohyb v záporném směru.
- Bit **AUTMAN_CONT_G00** definovaný v `CONT_AUTMAN` svou hodnotou log.1 informuje PLC program, že při řízení pohybu z panelu systému nebo z panýlku točítka je právě požadován rychloposuv. Každá souřadnice pojedí svým vlastním rychloposuvem, který je zadán v konfiguraci.
- Buňka **SHIFT_CONTROL**. Jednotlivé bity jsou nastaveny na hodnotu „1“ při aktivním posunu konkrétní souřadnice (viz dále).
- Pole **SHIFT_x** typu `DWORD`. Aktuální hodnota nezávislého posunutí pro jednotlivé souřadnice [1/8 μm].

20.2.3 Přímé sdílené proměnné

Přímé sdílené proměnné typu „**RtmVar**“ slouží pro předání stavu ručních pojezdů a režimu „SHIFT“ pro indikaci přímo do HTML oken a dialogů.

- Pole **KnobSelAx0**, **KnobSelAx1**, . . . typu `BYTE` slouží pro indikaci předvolené osy pro ruční pojezdy a pro nezávislé posouvání dráhy. Na základě hodnoty v buňce se například mohou měnit obrázky pro indikaci v HTML okně (viz příklad dále). Každá buňka indikuje stav jedné NC osy a může nabývat hodnot:
 - 0 = osa nevyvolena pro ruční pojezdy
 - 1 = osa je předvolena pro ruční pojezdy a případně pro posun z točítka
 - 2 = osa je předvolena pro nezávislý posun „SHIFT“
- Pole **Pos6ShiftAx0**, **Pos6ShiftAx1**, . . . typu `REAL` slouží pro indikaci aktuálního nezávislého posunutí pro jednotlivé osy „SHIFT“ [mm].
- Symbol **ReqShift** je přímé sdílení bitu `REQ_ESHIFT`
- Symbol **ActShift** je přímé sdílení bitu `ACT_ESHIFT`
- Symbol **KnobStep** je přímé sdílení buňky `NOB_STEP` typu `WORD`.
- Symbol **ShiftEnable** je přímé sdílení buňky `SHIFT_ENABLE` typu `BYTE`

20.2.4 Externí řízení ručních pojezdů z PLC

Bity pro externí řízení pohybu v pomocných ručních pojezdech slouží pro zadávání pohybu, směru pohybu, rychlosti pohybu.

- ▶ Bity **EXM_x0** a **EXM_x1** definované v **EXT_CONT_AUTMAN** slouží pro zadávání pohybu a směru pohybu v pomocných ručních pojezdech. Bit **EXM_X0** při nastavení na hodnotu log.1 je požadavkem na pohyb v ose X v kladném směru. Bit **EXM_X1** při nastavení na hodnotu log.1 je požadavkem na pohyb v ose X v záporném směru. Nastavení bitů **EXM_x0** a **EXM_x1** je platné jen tehdy, když je bit **REQ_EXT_CONT_AUTMAN** nastaven na log.1 (bude pojednáno dále). V opačném případě na hodnotách bitů **EXM_x0** a **EXM_x1** nezáleží. Pro požadavky na pohyb může být nastaveno i víc bitů najednou.
- ▶ Bit **REQ_EXT_G00_AUTMAN** definovaný v **EXT_CONT_AUTMAN** je externí požadavek pro rychloposuv. Když PLC program požaduje posun v pomocných ručních pojezdech rychloposuvem, nastaví bit **REQ_EXT_G00_AUTMAN** na hodnotu log.1. Rychloposuv pro každou souřadnici je určen v konfiguraci. Když má tento bit hodnotu log.0, souřadnice se pohybují rychlostí podle atributu „Feed“ nebo rychlostí zadanou externě z PLC programu. V každém případě se rychlost ovlivňuje pomocí potenciometru procenta F nebo externím řízením procenta rychlosti z PLC programu.
- ▶ Bit **REQ_EXT_SELECT_AUTMAN** definovaný v **EXT_CONT_AUTMAN** je externí požadavek na předvolbu pohybu v pomocných ručních pojezdech. PLC program hodnotou log.1 v bitu **REQ_EXT_SELECT_AUTMAN** a nastavením příslušného bitu **EXM_x0** předvolí pohyb v dané souřadnici. Osa se nerozjede, ale bude předvolena pro následující pohyb zadávaný od ručního kolečka.
- ▶ Bit **REQ_EXT_CONT_AUTMAN** definovaný v **EXT_CONT_AUTMAN3** je požadavek na externí řízení pohybu v pomocných ručních pojezdech. Když je bit nastaven na hodnotu log.0, zadávání souřadnic a směru pohybu se provádí z panelu systému nebo z panýlku točítka (interní řízení pohybu). V tomto případě na nastavení bitů **EXM_x0** a **EXM_x1** nezáleží. Když je bit **REQ_EXT_CONT_AUTMAN** nastaven na hodnotu log.1, zadávání souřadnic a směru pohybu se provádí externě z PLC programu pomocí bitů **EXM_x0** a **EXM_x1** (externí řízení pohybu). V tomto případě se nedá pohyb ovlivnit z panelu systému nebo z panýlku točítka. Bit je po zapnutí systému přednastaven na hodnotu log.0.
- ▶ Bit **REQ_EXT_FEED_AUTMAN** definovaný v **EXT_CONT_AUTMAN3** je požadavek na externí řízení rychlosti pohybu v pomocných ručních pojezdech. Když je bit nastaven na hodnotu log.0, zadávání rychlosti pohybu se provádí z panelu systému (interní řízení rychlosti). V tomto případě na nastavení buňky **EXT_FEED_AUTMAN** nezáleží. Když je bit **REQ_EXT_FEED_AUTMAN** nastaven na hodnotu log.1, zadávání rychlosti pohybu se provádí externě z PLC programu pomocí buňky **EXT_FEED_AUTMAN** (externí řízení rychlosti). V tomto případě se nedá rychlost pohybu ovlivnit z panelu systému. Bit je po zapnutí systému přednastaven na hodnotu log.0.
- ▶ Buňka **EXT_FEED_AUTMAN** typu WORD 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 µm/ot. V každém případě se rychlost ovlivňuje pomocí potenciometru procenta F nebo externím řízením procenta rychlosti z PLC programu.
- ▶ Bit **REQ_EXT_FEED_G00_AUTMAN** je požadavek na externí řízení rychlosti pohybu v pomocných ručních pojezdech rychloposuvem. Když je bit **REQ_EXT_FEED_G00_AUTMAN** nastaven na hodnotu log.1, zadávání rychloposuvu se provádí externě z PLC programu pomocí buňky **EXT_FEED_G00_AUTMAN** (externí řízení rychloposuvu). Bit je po zapnutí systému přednastaven na hodnotu log.0.
- ▶ Buňka **EXT_FEED_G00_AUTMAN** typu WORD slouží pro zadání externí rychlosti pro rychloposuv v pomocných ručních pojezdech z PLC programu. Takto zadaná rychlost se projeví, jen když je bit **REQ_EXT_G00_AUTMAN** nastaven na hodnotu log.1. Hodnota rychlosti se zadává v binárním kódu v mm/min.

- Bit **G95_AUTMAN** je požadavek na otáčkový posuv. Když PLC program nastaví bit G95_AUTMAN na hodnotu log.1, jsou všechny rychlosti posuvů (interní i externí) v pomocných ručních pojezdech vykonány otáčkovým posuvem v $\mu\text{m}/\text{ot}$. Tento bit neovlivňuje rychloposuv signalizovaný v AUTMAN_CONT_G00 nebo zadaný v REQ_EXT_G00_AUTMAN. Bit G95_AUTMAN je po zapnutí systému přednastaven na hodnotu log.0.
- Bit **AUTMAN_REQ** definovaný v BZH00 je 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 na hodnotu log.0, budou pomocné ruční pojezdy deaktivovány.
- Bit **AUTMAN_SELECT** definovaný v AUTMAN_SEL je požadavek na aktivaci pomocných ručních pojezdů. Hodnota log.1 aktivuje pomocné ruční pojezdy, pokud jejich aktivace je v systému povolena. PLC program jen nastavuje žádost hodnotou log.1 a systém po převzetí žádosti bit vždy vynuluje, a to bez ohledu na úspěšnost žádosti.
- Bit **AUTMAN_CANCEL** definovaný v AUTMAN_SEL je požadavek na deaktivaci pomocných ručních pojezdů. Hodnota log.1 zruší pomocné ruční pojezdy, pokud jejich deaktivace je v systému povolena (mohou být ještě jiné požadavky). PLC program jen nastavuje žádost hodnotou log.1 a systém po převzetí žádosti bit vždy vynuluje, a to bez ohledu na úspěšnost žádosti.
- Buňka **POS_SPACE_PLC** typu BYTE slouží pro určení prostoru, ve kterém má externí řízení pohybu probíhat. Systém standardně používá několik transformací, které jsou zařazeny na výstup interpolátoru. Jednotlivé transformace oddělují „**prostory**“, ve kterých je definovaná aktuální poloha (viz „Rozhraní CNC-PLC“).
- Buňka **POS_SPACE_MMM** typu BYTE slouží pro určení prostoru, ve kterém má probíhat pohyb řízený pomocí příkazů typu „RtmCommand“. Příkazy jsou konfigurovány například v souborech typu „*.KbdConfig“ (například RtmCommand="RTMID_IMMM_MOVEY", viz návod „Ovládací panely“).
- Bit **AUTMAN_KNOB1_DIS** definovaný v AUTMAN_SEL řídí povolení ovládání z panelu 1.točítka. Hodnota log.1 zakáže funkčnost tlačítek z 1.točítka.
- Bit **AUTMAN_KNOB2_DIS** definovaný v AUTMAN_SEL řídí povolení ovládání z panelu 2.točítka. Hodnota log.1 zakáže funkčnost tlačítek z 2.točítka.
- Bit **AUTMAN_WHEEL_DIS** definovaný v AUTMAN_SEL2 zakáže automatickou volbu osy v ručních posuvech (RTMCommand) pro další ovládání točítkem.
- Bit **AUTMAN_KNOB_TOGGLE** definovaný v AUTMAN_SEL2 modifikuje způsob předvolby na ovládacím točítku. Příkazy RTMCommand (RTMID_MMM_SEL. .) pro předvolbu os střídavě aktivují a deaktivují možnost ovládání os z panýlku točítka.
- Buňka **KNOB_STEP** typu WORD slouží pro zadání kroku točítka.
- Buňka **SHIFT_ENABLE**. Jednotlivé bity slouží pro povolení nezávislého posouvání dráhy (viz dále).

20.3 Ovládání ručních pojezdů pomocí Joysticku

Systém umožňuje připojit ruční panálek s Joystickem pro ovládání ručních pojezdů (viz. Návod „Ovládací panely“).

20.3.1 Konfigurace pro Joystick

Joystick je připojen pomocí sběrnice CAN-BUS na jednotku EtherCAT KLA60, jako další typ panelu. Proto se konfiguruje v ChannelConfigu jako samostatné HMI zařízení, například:

```
<HMIDevice No="0"
  Active="1"
  DeviceType="PanelKla60"
  DeviceName="ECAT.KLA60_1">
</HMIDevice>
<HMIDevice No="1"
  Active="1"
  DeviceType="JoystickKla50"
  DeviceName="CAN[1].KLA50[2]">
</HMIDevice>
```

Tlačítka na panýlku Joysticku budou mít standardně svůj konfigurační soubor typu „KbdConfig“, který je potřeba vytvořit a také zaregistrovat (viz konfigurace tlačítek v návodu „Ovládací panely“).

```
#{registry::Write}
"HKLM\Software\MEFI\WinCNC\Machine\Channels\Channel0\Keyboards\Keyboard1"
"ConfigFiles" "JoystickArea.KbdConfig" "REG_MULTI_SZ" $R0
```

Příklad pro soubor JoystickArea.KbdConfig:

```
<KeyboardConfig>
  <Comment>
    Tlačítka vzdáleného ovládání s joystickem.
  </Comment>

  <KeyConfig ScanCode="0x17" Type="Normal" PlcCommand="cmdJoystick"></KeyConfig>
  <KeyConfig ScanCode="0x0f" Type="Normal" Command="0" ></KeyConfig>
  <KeyConfig ScanCode="0x07" Type="Normal" Command="0" ></KeyConfig>

</KeyboardConfig>
```

Dále je nutno v souboru typu ChannelConfig konfigurovat logické potenciometry „Override“, například: Systém vynechané kroky v tabulce interpoluje a hodnota pro Joystick musí být z intervalu (-1, 1).

```
<Override No="3"
  DefaultValueNo="128"
  ConnectToPhysicalPotentiometer="1"
  PhysicalPotentiometerNo="2">
  <OverrideValue No="0"> -1.0 </OverrideValue>
  <OverrideValue No="30"> -1.0 </OverrideValue>
  <OverrideValue No="118"> 0.0 </OverrideValue>
  <OverrideValue No="128"> 0.0 </OverrideValue>
  <OverrideValue No="138"> 0.0 </OverrideValue>
  <OverrideValue No="220"> 1.0 </OverrideValue>
  <OverrideValue No="255"> 1.0 </OverrideValue>
</Override>
```

```

<Override No="4"
  DefaultValueNo="128"
  ConnectToPhysicalPotentiometer="1"
  PhysicalPotentiometerNo="3">
  <OverrideValue No="0">    -1.0 </OverrideValue>
  <OverrideValue No="30">   -1.0 </OverrideValue>
  <OverrideValue No="118">   0.0 </OverrideValue>
  <OverrideValue No="128">   0.0 </OverrideValue>
  <OverrideValue No="138">   0.0 </OverrideValue>
  <OverrideValue No="220">   1.0 </OverrideValue>
  <OverrideValue No="255">   1.0 </OverrideValue>
</Override>

```

20.3.2 Ovládání Joysticku z PLC

PLC soubor musí mít připojen systémový soubor include **SysMac.inc**:

```
t_include 'SysMac.inc'
```

V tomto souboru jsou definovány makra pro inicializaci a ovládání Joysticku:

```

;Makro pro nastavení Joysticku pro osu
;Parametry:

;Axis .....Index NC osy (0,1,2,3,..)
;MaxFeed ....Maximalní rychlost [mm/min],
;             hodnota 0 znamená, že se použije rychlosti "Feed,FeedSafe" ze ;
             ChannelConfigu
;Override ...číslo override podle elementu "Override" v Channelconfigu (0,1,2,..) ;
             (ne podle fyzického potenciometru)
;Dir .....priznak otocení směru

DEF_T_MACRO JOYSTICK_INIT Axis, MaxFeed, Override, Dir

;.....

;Makro aktivace a deaktivace joysticku pro osu
;Parametry:

;Axis .....      Index NC osy (0,1,2,3,..)
;Active .....    0/1 aktivace řízení joystickem

DEF_T_MACRO JOYSTICK Axis, Active

```

(Příklad PLC pro ovládání Joysticku je v projektu „PLC Sample Joystick Ecat“.)

20.4 Nezávislé posouvání dráhy

Jedná se o možnost nezávislého posouvání dráhy pomocí ručního kolečka i v průběhu jetí programu. Posun je k dráze připočten až na výstupu z interpolátoru a tak neovlivní průběh programu, všechny typy posunutí dráhy a ani interpolaci v rámci bloku.

Pro nezávislé posouvání dráhy budeme používat označení: „**SHIFT**“.

20.4.1 Volba režimu a aktivace posouvání dráhy

Volba režimu posouvání se provádí například pomocí tlačítka panelu systému, které má v definičním souboru typu „KbdConfig“ nastaven `RtmCommand="RTMID_MMM_SHIFT_REQ"`, viz návod: „Ovládací panely“.

Volbu režimu posouvání `SHIFT` lze provést i z PLC nastavením bitu `REQ_ESHIFT` na hodnotu log.1

Zrušení režimu posouvání se provádí pomocí stejného příkazu, jako pro volbu posouvání. Při zrušení režimu posouvání dráhy systém odjede na původní místo před posunem dráhy.

Aktivace posunu dráhy se může provést například pomocí tlačítka panelu systému, které má v definičním souboru typu „KbdConfig“ nastaven `RtmCommand="RTMID_MMM_SHIFT_ACT"`.

Aktivaci posouvání `SHIFT` lze provést i z PLC nastavením bitu `ACT_ESHIFT` na hodnotu log.1.

V aktivním stavu posouvání dráhy je možno volit osu pro posun pomocí standardních příkazů předvolby. (například `RtmCommand="RTMID_MMM_SEL00"`), viz návod: „Ovládací panely“. Pomocí ručního kolečka je možno aktuální posun kdykoli měnit.

Po posunutí dráhy je vhodné aktivaci zrušit stejným příkazem, jak byla aktivace zvolena. Ruční točítko přestane dráhu posouvat a zruší se poslední volba osy. Všechny aktuální posuny dráhy zůstanou nadále platné.

Na obrazovce systému je vhodné indikovat:

- režim `SHIFT` je aktivní
- Která osa se právě posouvá
- Aktuální posun pro všechny osy

Pro snadnou tvorbu HTML oken slouží přímo sdílené systémové proměnné (už bylo popsáno) :

- `KnobSelAx0, ..`
- `Pos6ShiftAx0, ..`
- `ActShift`

Příklad tvorby HTML okna bude uveden dále.

20.4.2 Informace pro PLC pro nezávislý posun

PLC program má k dispozici několik signálů, které se týkají nezávislých posunů dráhy:

Název	Typ	Popis
REQ_ESHIFT	BIT	PLC program může číst a nastavovat. Bit je nastaven na hodnotu "1" po celou dobu zvoleného režimu posouvání.
ACT_ESHIFT	BIT	PLC program může číst a nastavovat. Bit je nastaven na hodnotu "1" po dobu aktivace posunu dráhy v platném režimu posouvání.
SHIFT_CONTROL	BYTE	PLC program může jen číst. Jednotlivé bity jsou nastaveny na hodnotu "1" při aktivním posunu konkrétní souřadnice. Bity jsou přiřazeny osám postupně od nejnižší váhy podle pořadí definice os (bit 0=1.osa, bit 1=2. osa, ...). Hodnota v buňce SHIFT_CONTROL je platná pouze, když je bit ACT_ESHIFT nastaven na hodnotu "1".
SHIFT_ENABLE	BYTE	PLC program může nastavovat. Jednotlivé bity jsou svou hodnotu "1" povolují pohyb při aktivním posunu pro konkrétní souřadnice. Bity jsou přiřazeny osám postupně od nejnižší váhy podle pořadí definice os (bit 0=1.osa, bit 1=2. osa, ...). Proměnná SHIFT_ENABLE je přednastavena na hodnotu 3Fh.
SHIFT_x	DWORD	PLC program může jen číst. Pole 6 double-wordů. Aktuální hodnota posunutí pro jednotlivé souřadnice v osminách mikronu

20.5 Příklad tvorby okna pro ruční režimy

Dále uvedeme příklad tvorby okna pro indikaci předvolby osy točítka a nezávislého posunu dráhy. Upraví se například okno „MAIN“ (soubor MAIN.HTML) tak, aby zeleným kolečkem před názvem souřadnice se indikovala předvolená osa pro točítka a červeným kolečkem osa, která má aktivní nezávislý posuv (SHIFT). Červený údaj udává o kolik je osa posunuta v režimu SHIFT. Kompletní řešení je uvedeno ve vzorovém projektu PLC pro školení.

X	7.536	F	0
-0.114	0.000		
Y	-25.853	S	0
0.000	0.000		
Z	0.000	Dist	0.000
0.000	0.000		

```

<!-- Výpis názvu souřadnice -->
<DIV id="AxisX_Lbl" class="LabelBig">X</DIV>

<!-- Výpis aktuální polohy -->
<DIV id="AxisX_Pos" class="ValueBig" CNCType="AsyncIndicatorPainter"
  AIPLibrary="StdPlugins" AIPTType="Tool" AIPOptions="Channel: 0; Axis: X;
  ValType: Pos0;">+0000.000</DIV>

<!-- Výpis značky reference -->
<DIV id="AxisX_InRef" class="ValueMedium" CNCType="AsyncIndicatorPainter"
  AIPLibrary="StdPlugins" AIPTType="Axis" AIPOptions="Channel: 0; Axis: 0;
  ValType: InRef; GraphType: Image; ImgWidth: 15; ImgHeight: 15;">*</DIV>

<!-- Výpis distance -->
<DIV id="AxisX_Dist" class="ValueMedium" CNCType="AsyncIndicatorPainter"
  AIPLibrary="StdPlugins" AIPTType="Axis" AIPOptions="Channel: 0; Axis: 0;
  ValType: Distance;">+0000.000</DIV>

<!-- Výpis značky předvolené osy točítka a SHIFT -->
<DIV id="AxisX_Knob" CNCType="AsyncIndicatorPainter"
  AIPLibrary="StdPlugins" AIPTType="RtmVar" AIPOptions="Channel: 0;
  Variable: 'KnobSelAx0'; GraphType: Image; Img0: KnobSel0.png; Img1:
  KnobSel1.png; Img2: KnobSel2.png;">*</DIV>

<!-- Výpis aktuálního posunutí SHIFT -->
<DIV id="AxisX_Shift" class="ValueMedium" CNCType="AsyncIndicatorPainter"
  AIPLibrary="StdPlugins" AIPTType="RtmVar" AIPOptions="Channel: 0;
  Variable: 'Pos6ShiftAx0'; NumberPrecision: 3;">+00.000</DIV>

```

V hlavičce HTML souboru je definován „stylopis“ pro každý prvek. Například pro výpis značky předvolby točítka je:

```

<STYLE type="text/css">
  #AxisX_Knob, #AxisY_Knob, #AxisZ_Knob
  {position: absolute; left: 9px; top: 24px; width: 16px; height: 16px; }
</STYLE>

```

20.6 Některé možnosti řízení z PLC programu

Krátce pojednáme o jednotlivých možnostech řízení pomocných ručních pojezdů z PLC programu.

20.6.1 Příklad řízení, když jsou všechny souřadnice trvale v polohové vazbě

V případě, že na stroji jsou všechny souřadnice trvale v polohové vazbě, je situace velmi jednoduchá. PLC program nemusí mít žádnou podporu pro pomocné ruční pojezdy.

V případě, že je potřeba uvolňovat a upínat osy nebo ovládat spojky pro souřadnice, můžeme si vybrat, zda budeme pro pomocné ruční pojezdy blokovat pohyb pomocí signálů **MPxPI** nebo **MPxMAN**. Výběr provedeme pomocí nastavení 4.dekády strojní konstanty R233, jak už bylo popsáno výše.

20.6.2 Příklad řízení pro 1 osu s upínáním bez panýlku s ručním točítkem

Příklad pro řízení osy X v části `MODULE_MAIN`, pro blokování pohybu použijeme signál `MPxMAN` (4. dekáda strojní konstanty R233 je 2). Bity `MPxMAN` jsou v základním stavu v hodnotě `log.0`. V tomto případě není použitý panýlek s ručním točítkem. Ovládání můžeme napsat v části `MODULE_MAIN`:

```

.....
LDR  ACK_AUTMAN          ;test aktivace pomocných ručních pojezdů
LA   PO_OSXPI            ;je požadován pohyb v ose X ?
FL1  1,MCH_MAN_X        ;podmíněná aktivace mechanismu pohybu X
.....

MECH_BEGIN MCH_MAN_X          ;definice mechanismu pro řízení pohybu X
.....
....  uvolnění osy , zapnutí vazby ,...
.....
FL    1,MPXMAN              ;povolení pohybu v ose X pro ruční pojezdy
EX
LDR   PO_OSXPI
LA    ACK_AUTMAN
EX1   :čekání a ukončení pohybu pro ruční pojezdy
FL    0,MPXMAN              ;zákaz pohybu pro ruční pojezdy
.....
....  upnutí osy, vypnutí vazby ,...
.....
MECH_END  MCH_MAN_X

```

20.6.3 Příklad pro externí řízení pohybu pomocí potenciometrů

Na stroji jsou potenciometry pro zadávání rychlosti pohybu a přepínače pro volbu směru pohybu.

Vstupy z panelu stroje:

```

;X_SLOW_I      pomalý stupeň
;X_RAPID_I     rychlý stupeň
;X_ON_I        potenciometr zapnut
;X_OFF_I       potenciometr vypnut
;X_MINUS_I     záporný směr
;X_PLUS_I      kladný směr
POT10: DFM X_SLOW_I,X_RAPID_I,X_ON_I,X_OFF_I,X_MINUS_I,X_PLUS_I,,

;minulé stavy a příznaky
POT11: DFM X_SLOW_M,X_RAPID_M,X_ON_M,X_OFF_M,X_MINUS_M,X_PLUS_M,,

ANALOG_POT_X:   DS    4
POT_FEED_X:     DS    4

;Mechanismus pro test volby směru pro potenciometry
MECH_BEGIN MechPotencSmerX
    fl    1,MechPotencFeedX           ;zadávání rychlosti pot.
    fl    1,MechPotencStopX          ;stop pohybu pot.
PotencSmerX:
    ex
    ldr   MechPotencStopX             ;čeká na konec případného stopu
    ex1
    ldr   X_MINUS_I                  ;čeká na volbu směru
    lo    X_PLUS_I
    ex0
    ldr   X_MINUS_I                  ;nesmi byt oboje
    la    X_PLUS_I
    jll   PotencSmerX

    ldr   X_MINUS_I                  ;paměť směru
    wr    X_MINUS_M
    ldr   X_PLUS_I
    wr    X_PLUS_M
    fl    1,MechPotencPohybX          ;mohou se testovat podmínky pohybu
    ex
    ldr   X_MINUS_I                  ;kontroluje změnu směru při běhu
    lx    X_MINUS_M
    ldr   X_PLUS_I
    lx    X_PLUS_M
    lo
    lo    -MechPotencPohybX
    ex0
    fl    1,MechPotencStopX          ;stop pohybu pot.
    jum   PotencSmerX
MECH_END   MechPotencSmerX

```

```

;Externí zadávání rychlosti pro potenciometry
MECH_BEGIN MechPotencFeedX
PoteFeedC:
    ex
    ldr MechPotencStopX                ;čeká na konec případného stopu
    ex1
    lod cnst.100
    ldr X_SLOW_I                       ;pomalá rychlost
    wr X_SLOW_M
    stol POT_FEED_X
    lod cnst.5000
    ldr X_RAPID_I                      ;velká rychlost
    wr X_RAPID_M
    stol POT_FEED_X
    ex
    lod POT_FEED_X
    mulb ANALOG_POT_X                 ;sejmuta analog.hodnota pot.
    divb cnst.1000
    sto EXT_FEED_AUTMAN               ;externí zadání rychlosti
    ldr X_SLOW_I                      ;test změny rychlosti
    lx X_SLOW_M
    ldr X_RAPID_I
    lx X_RAPID_M
    lo
    lo -MechPotencPohybX
    ex0
    fl 1,MechPotencStopX              ;stop
    jum PoteFeedC
MECH_END MechPotencFeedX

;Stop pohybu potenciometru
MECH_BEGIN MechPotencStopX
    fl 0,EXM_X0,EXM_X1                ;konec pohybu
    mech_init MechPotencPohybX
    ldr PrPosXVyp                     ;čeká na vytočení pot. do nuly
    la -X_ON_I
    ex0
MECH_END MechPotencStopX

;Test podmínek pohybu pro potenciometry
MECH_BEGIN MechPotencPohybX
    ldr X_ON_I                        ;čeká na pootočení pot.
    la -X_OFF_I
    ex0
    fl 1,AUTMAN_REQ                   ;požadavek na AUTMAN
    fl 1,REQ_EXT_CONT_AUTMAN,REQ_EXT_FEED_AUTMAN ;externí řízení
    ex
    ldr X_PLUS_M                      ;aktivace pohybu !
    wr EXM_X0
    ldr X_MINUS_M
    wr EXM_X1
    bex
    ldr X_ON_I                        ;testuje pootočení pot.
    la -X_OFF_I
    ex1
    fl 1,MechPotencStopX
MECH_END MechPotencPohybX

```

```
;Konec režimu pro potenciometry
MECH_BEGIN MechPotencEnd
    fl    0,EXM_X0,EXM_X1           ;konec pohybu
    fl    0,AUTMAN_REQ
    fl    0,REQ_EXT_CONT_AUTMAN,REQ_EXT_FEED_AUTMAN
    mech_init MechPotencSmerX
    mech_init MechPotencPohybX
    mech_init MechPotencFeedX
    mech_init MechPotencStopX
MECH_END    MechPotencEnd
```