

27

27. LOGICKÉ VSTUPY A VÝSTUPY MODULŮ SYSTÉMU

27.1 Princip

V kapitole jsou popsány prostředky pro logické propojování obecných vstupů a výstupů různých modulů systému (včetně PLC). Nové možnosti logického mapování pomocí instrukcí DEF_IN a DEF_OUT se neomezuje jen na CAN-BUS periferie MEFI, jako je to v případě mapování pomocí instrukcí MAP_IN a MAP_OUT, ale umožní například propojení PLC - EtherCAT apod.

Definice logických vstupů a výstupů umožňuje jejich propojování jen na základě konfigurace a bez nutnosti změny PLC programu. PLC program může mít nadefinován velké množství vstupů a výstupů ale jen některé z nich se propojí podle konfigurace na danou specifikaci stroje.

Jeden logický spoj je definován dvěma textovými identifikátory pro vstup a pro výstup.

Pokud logické vstupy a výstupy nejsou propojeny, může se uplatnit přednastavení buněk podle defaultní hodnoty.

Pro každý logický vstup a výstup možno určit požadovanou dobu obsluhy „virtuálního spoje“ (Service Period). Tem může být podle cyklu PLC (SLOW), podle cyklu interpolátoru (FAST), podle cyklu EtherCAT Mastra (FASTEST), nebo „na vyžádání“ (ONDEMAND).

Při definici logických prvků pro „virtuální spoje“ v PLC programu se proměnné automaticky deklarují – vymezí se jim požadovaný paměťový prostor.

Datové prvky, které definují logický spoj, mohou být definovány jako pole s ohledem na typ proměnné. V tomto případě logický spoj k danému prvku určuje jeho jméno a koncový index.

Pro každý logický spoj možno definovat konverzi, která se uplatní při propojení prvků.

Při propojování prvků s různými datovými typy dochází automaticky k přetypování proměnné podle typu cílové proměnné. Například se provede znaménkové rozšíření, nebo převod na reálné číslo.

Všechny logické vstupy a výstupy mají automaticky nastaveno „**přímé sdílení**“ a tak se značně zjednoduší přístup uživatelských obrazovek a dialogů k jejím hodnotám. Na logické vstupy a výstupy není potřeba používat instrukce SHARE_VAR a SHARE_BIT.

27.2 Popis modulů pro logické spoje

V „modulech“ jsou definovány logické vstupy a výstupy pro tvorbu virtuálních spojů. Název modulu se uvádí v identifikátoru prvku na prvním místě.

Modul	popis
CAN[1]	Druhý CAN-BUS kanál pro obsluhu periférií vstupů, výstupů, panelů a pod. (První CAN-BUS kanál je určen pro obsluhu pohonů). Jedná se o možnost připojení k perifériím MEFI: INOUT08. Z hlediska propojování jsou fyzické binární vstupy definovány jako logické výstupy a fyzické binární výstupy jsou logické vstupy. Typ prvků je BIT nebo BYTE a možnost obsluhy je FAST (viz dále).
VKeyboard	Virtuální klávesnice. Konfiguruje se v pořadí jako 7. klávesnice (pořadí 6 od 0). Nejedná se o fyzickou klávesnici nebo panel. Virtuální klávesnice má definováno 256 virtuálních tlačítek (s 256 SCAN kódy), ke kterým se možno připojit a ovládat je jako kdyby to byly standardní tlačítka. Virtuální tlačítka svými stisky tak mohou vykonávat příkazy podle konfigurace, která je zadána v souboru typu „KbdConfig“. Tanto soubor musí být zaregistrován pro 7.klávesnici. Virtuální klávesnice v podstatě umožňuje PLC programu pomocí bitu spouštět příkazy definovány pomocí atributů: Command, RtmCommand, PlcCommand, nebo přímo zavolat dialogové okno definováno v elementu <Dialog>...</Dialog> (viz návod „Ovládací panely“). Typ prvků je BIT a možnost obsluhy je SLOW (viz dále).
PLC.Input PLC.Output	Logické vstupy a výstupy definované v PLC programu pomocí instrukcí DEF_IN a DEF_OUT (viz dále). Perioda obsluhy a typ jsou definovány v instrukcích v PLC programu.
ECAT	Připojení k logickým vstupům a výstupům definovaným v „Process image“ pro EtherCAT komunikaci. Pro jeho tvorbu je nutno použít „EtherCAT Studio“. (viz návod „Periferie připojené na EtherCAT“). Typ je definován v „Process image“. Možnost periody obsluhy: FASTEST (viz dále).
RTM.Axis[x]	Logické vstupy a výstupy definované v systémovém programu pro NC osy (RTM je systémový program reálného času). Index NC osy „x“ může nabývat hodnot 0 – 15 podle pořadového čísla osy. Výčet hodnot je uveden v následující tabulce. Možnost periody obsluhy: FAST (viz dále).
RTM.Servo[x]	Logické vstupy a výstupy definované v systémovém programu pro servosmyčky. Index servosmyčky „x“ může nabývat hodnot 0 – 15 podle pořadového čísla serva. Výčet hodnot je uveden v následující tabulce. Možnost periody obsluhy: FAST.
RTM.Spindle	Logické vstupy a výstupy definované v systémovém programu pro vřetena. Výčet hodnot je uveden v následující tabulce. Možnost periody obsluhy: FAST.
RTM.Man	Logické vstupy a výstupy definované v systémovém programu pro manuální režim. Výčet hodnot je uveden v následující tabulce. Možnost periody obsluhy: FAST.
RTM.Control	Logické vstupy a výstupy definované v systémovém programu pro řízení běhu PLC apod.). Výčet hodnot je uveden v následující tabulce. Možnost periody obsluhy: FAST.

27.3 Zásady tvorby identifikátorů logických prvků

Logické virtuální spoje jsou definovány dvěma textovými identifikátory. Identifikátor logického výstupu a identifikátor logického vstupu. Textové identifikátory používají „tečkovou konvenci“ podle dále popsanych zásad.

Datové prvky, které definují logický spoj, mohou být v PLC programu definovány jako pole s ohledem na typ proměnné. V tomto případě logický spoj k danému prvku určuje jeho jméno a také koncový index. Proto koncový index není součástí identifikátoru. Identifikátor ale může obsahovat i další indexy (které nejsou koncové) a ty pak musí být jeho součástí.

Identifikátory pro vstupy a výstupy EtherCATu je nutno podle potřeby upravit v EtherCAT studiu.

Možnosti tvorby textových identifikátorů

Skupiny pro logické spoje	Popis
Periférie CAN-BUS INOUT08 – logické výstupy – bitový přístup	
CAN[1].INOUT08[xx].IP0.Bit[xx] CAN[1].INOUT08[xx].IP1.Bit[xx] CAN[1].INOUT08[xx].IP2.Bit[xx] CAN[1].INOUT08[xx].IP3.Bit[xx]	Připojení k fyzickým vstupům INOUT08. Typ: BIT. Možnost periody obsluhy: FAST
Periférie CAN-BUS INOUT08 – logické vstupy – bitový přístup	
CAN[1].INOUT08[xx].OP0.Bit[xx] CAN[1].INOUT08[xx].OP1.Bit[xx] CAN[1].INOUT08[xx].OP2.Bit[xx]	Připojení k fyzickým výstupům INOUT08. Typ: BIT. Možnost periody obsluhy: FAST
Periférie CAN-BUS INOUT08 – logické výstupy – bajtový přístup	
CAN[1].INOUT08[xx].IP0 CAN[1].INOUT08[xx].IP1 CAN[1].INOUT08[xx].IP2 CAN[1].INOUT08[xx].IP3	Připojení k fyzickým vstupům INOUT08. Typ: BYTE. Možnost periody obsluhy: FAST
Periférie CAN-BUS INOUT08 – logické vstupy – bajtový přístup	
CAN[1].INOUT08[xx].OP0 CAN[1].INOUT08[xx].OP1 CAN[1].INOUT08[xx].OP2	Připojení k fyzickým výstupům INOUT08. Typ: BYTE. Možnost periody obsluhy: FAST
Virtuální klávesnice – logické vstupy	
VKeyboard.VKKey[xx]	Připojení na virtuální klávesnici. Typ: BIT Možnost periody obsluhy: SLOW
PLC proměnné – logické vstupy	
PLC.Input.<jmeno>[xx]	Připojení k logickým vstupům definovaných v PLC programu (<jmeno> je definováno instrukcí DEF_IN) Perioda obsluhy a typ jsou definovány v instrukci DEF_IN
PLC proměnné – logické výstupy	
PLC.Output.<jmeno>[xx]	Připojení k logickým výstupům definovaných v PLC programu (<jmeno> je definováno instrukcí DEF_OUT) Perioda obsluhy a typ jsou definovány v instrukci DEF_OUT
EtherCAT – logické vstupy a výstupy	
ECAT.<jmeno>[xx]	Připojení k logickým vstupům a výstupům definovaným v „Process image“ pro EtherCAT komunikaci. Typ je definován v „Process image“. Možnost periody obsluhy: FASTEST

RTM proměnné – NC osy	
RTM.Axis[xx].<jmeno>	Připojení k logickým vstupům a výstupům definovaných v RTM programu pro NC osy (Výčet hodnot <jmeno> je uveden v následující tabulce). Možnost periody obsluhy: FAST
RTM proměnné – servosmyčky	
RTM.Servo[xx].<jmeno> RTM.Servo[xx].Edrv.<jmeno> RTM.Servo[xx].Enc.<jmeno>	Připojení k logickým vstupům a výstupům definovaných v RTM programu pro servosmyčky (Výčet hodnot <jmeno> je uveden v následující tabulce). Možnost periody obsluhy: FAST
RTM proměnné – vřetena	
RTM.Spindle.<jmeno>	Připojení k logickým vstupům a výstupům definovaných v RTM programu pro vřetena (Výčet hodnot <jmeno> je uveden v následující tabulce). Možnost periody obsluhy: FAST
RTM proměnné – manuální režim	
RTM.Man.<jmeno>[xx]	Připojení k logickým vstupům a výstupům definovaných v RTM programu pro manuální režim (Výčet hodnot <jmeno> je uveden v následující tabulce) Možnost periody obsluhy: FAST
RTM proměnné – řízení	
RTM.Control.<jmeno>[xx]	Připojení k logickým vstupům a výstupům definovaných v RTM programu pro řízení běhu PLC programu apod. (Výčet hodnot <jmeno> je uveden v následující tabulce) Možnost periody obsluhy: FAST

Pravidla pro definici identifikátoru prvku v PLC, v EtherCAT apod. (pomocí instrukce DEF_IN, DEF_OUT)

- Při definici v PLC programu se uvádí jen jméno prvku <jmeno> bez názvu skupiny.
- Na velikosti písmen nezáleží.
- Použití nulových indexů ([0], [00], [000]...) je ekvivalentní zápisu bez indexů. Možno je libovolně používat. Nulové indexy nejsou součástí identifikátoru.
- Vedoucí nuly v indexech se automaticky odstraňují ([05] = [5]), nejsou součástí identifikátoru. Možno je používat.
- Nesmí se v definici uvést žádný koncový index [xx], ten není součástí identifikátoru a může se uvést až v pravidlech pro vyhledávání log.spoje.
- Jméno prvku <jmeno> může obsahovat tečky a indexy [xx], které jsou součástí identifikátoru.

Pravidla pro vyhledávání pro předpis log.spoje v konfiguraci ... element <Connection> a v HTML stránkách

- Na začátku může být nepovinně klíč „CNC.“
- Zápis s indexem ([0], [00], [000]...) na libovolné pozici je ekvivalentní zápisu bez indexu. Možno jej používat
- Vedoucí nuly v indexech se automaticky odstraňují a možno je používat ([05] = [5]).
- Na velikosti písmen nezáleží.
- Analogové hodnoty mohou mít koncový index [xx], který se uplatní podle typu proměnné.
- Jméno prvku <jmeno> může obsahovat tečky a indexy [xx], které jsou součástí identifikátoru.
- Jméno prvku <jmeno> může mít přetypování na anonymní přístup k bitu „.Bit[xx]“

Příklady pro definici instrukcí DEF_IN, DEF_OUT v PLC programu:

```
DEF_IN flBitIn1,          `BitIn1`,          TYPE_BIT
DEF_IN flBitOut2,        `BitOut2`,         TYPE_BIT
DEF_IN rBunOut3[10],     `BunOut3`,         TYPE_REAL
DEF_IN dwBunIn1,        `BunIn1[2].Prvek`,  TYPE_DWORD
DEF_IN inLaserEgBodyTouch, `PrecitecEG8030[0].BodyTouch`, TYPE_BIT
DEF_IN rLaserEgDistanceLinear, `PrecitecEG8030.DistanceLinear`, TYPE_REAL
```

Příklady pro předpis v konfiguraci v elementu <Connection> :

```
Source="PLC.Output.BitOut2"
Destination="CAN[1].INOUT08[2].OP1.Bit[2]"
```

```
Source="PLC.Output.BitOut2"
Destination="VKeyboard.Vkey[22]"
```

```
Source="PLC.Output.BunOut3[4]"
Destination="PLC.Input.dwBunIn1"
```

```
Source="ECAT.ServoX.Inputs.Status word"
Destination="PLC.Input.Servo[8].StatusWord"
```

```
Source="ECAT.ServoX.Inputs.Position actual value"
Destination="PLC.Input.Servo[8].PositionAct"
```

```
Source="ECAT.CANOpen.EG8030_Inputs.TxPDO1_wDeviceStateEG.Bit[4]"
Destination="PLC.Input.PrecitecEG8030.TipTouch"
```

```
Source="RTM.Axis[2].Pos0_AX"
Destination="PLC.Input.Position_2"
```

27.4 Tabulka RTM prvků pro logické spoje

Tabulka systémových „RTM“ prvků pro možnost vytváření logických spojů.

RTM. Axis [xx] .		
ReqShift	BIT	Požadavek na režim SHIFT
ActShift	BIT	Režim SHIFT aktivní
ShiftEnable_Ax	pole BIT (x = 0,1,..,15)	Povolení SHIFT z PLC
Pos5Inc	REAL [mm/T]	Prostorový aktuální přírůstek dráhy v GEO osách v POS5
Pos6Inc	REAL [mm/T]	Prostorový aktuální přírůstek dráhy v GEO osách v POS6
Pos5IncPm	REAL [mm/min]	Prostorový aktuální přírůstek dráhy v GEO osách v POS5
Pos6IncPm	REAL [mm/min]	Prostorový aktuální přírůstek dráhy v GEO osách v POS6
IncPt_Ax	pole REAL [mm/T] (x = 0,1,..,15)	Aktuální přírůstek dráhy v ose
Inc_Ax	pole REAL [mm/min] (x = 0,1,..,15)	Aktuální přírůstek dráhy v ose
Angle5Ax1	REAL [rad]	1.fyzická osa rotace pro naklápění
Angle5Ax2	REAL [rad]	2.fyzická osa rotace pro naklápění
Pos0_Ax	pole REAL [mm] (x = 0,1,..,15)	Poloha osy v POS0
Pos1_Ax	pole REAL [mm] (x = 0,1,..,15)	Poloha osy v POS1
Pos2_Ax	pole REAL [mm] (x = 0,1,..,15)	Poloha osy v POS2
Pos3_Ax	pole REAL [mm] (x = 0,1,..,15)	Poloha osy v POS3
Pos4_Ax	pole REAL [mm] (x = 0,1,..,15)	Poloha osy v POS4
Pos5_Ax	pole REAL [mm] (x = 0,1,..,15)	Poloha osy v POS5
Pos6_Ax	pole REAL [mm] (x = 0,1,..,15)	Poloha osy v POS6
ToolVectorX	REAL	Jednotkový prostorový vektor orientace nástroje - složka X
ToolVectorY	REAL	Jednotkový prostorový vektor orientace nástroje - složka Y
ToolVectorZ	REAL	Jednotkový prostorový vektor orientace nástroje - složka Z
KnobStep	UNS16	Krok točítka
Ax1BevelAngle	REAL [rad]	Fyzický úhel 1.rotace při úkosové interpolaci (mezivýsledek)

Ax2BevelAngle	REAL [rad]	Fyzický úhel 2.rotace při úkosové interpolaci (mezivýsledek)
CircleAngle	REAL [rad]	Aktuální úhel kruhové interpolace
CircleDist	REAL [mm]	Distance pro kruhovou interpolaci
Distanc_Ax	pole REAL [mm] (x = 0,1,,15)	Složkové distance pro jednotlivé osy
Distanc	REAL [mm]	Prostorová distance os
ToolTangentialAngleXY	REAL [rad]	Průmět aktuálního tečnového úhlu úkosu ToolTangentialAngle
TangentialAngleAct	REAL [rad]	Aktuální tečnový úhel bloku
ToolBevelAngleAct	REAL [rad]	Aktuální úhel úkosu pro úkosovou interpolaci
ToolTangentialAngleAct	REAL [rad]	Aktuální tečnový úhel úkosu v úkosové rovině
Pos6Shift_Ax	pole REAL [mm/min] (x = 0,1,,15)	Posun SHIFT
TouchPos0_Ax	pole REAL [mm] (x = 0,1,,15)	Poloha osy v POS0 naměřena dotykovou sondou
TouchPos1_Ax	pole REAL [mm] (x = 0,1,,15)	Poloha osy v POS1 naměřena dotykovou sondou
TouchPos2_Ax	pole REAL [mm] (x = 0,1,,15)	Poloha osy v POS2 naměřena dotykovou sondou
TouchPos3_Ax	pole REAL [mm] (x = 0,1,,15)	Poloha osy v POS3 naměřena dotykovou sondou
TouchPos4_Ax	pole REAL [mm] (x = 0,1,,15)	Poloha osy v POS4 naměřena dotykovou sondou
TouchPos5_Ax	pole REAL [mm] (x = 0,1,,15)	Poloha osy v POS5 naměřena dotykovou sondou
TouchPos6_Ax	pole REAL [mm] (x = 0,1,,15)	Poloha osy v POS6 naměřena dotykovou sondou
TouchProbe0Stop	BIT	Stop pohybu od dotykové sondy
TouchProbe0	BIT	Kontakt dotykové sondy
IncManPos1_Ax	pole REAL [mm/min] (x = 0,1,,15)	Aktuální přírůstek dráhy MAN v POS1
IncManPos2_Ax	pole REAL [mm/min] (x = 0,1,,15)	Aktuální přírůstek dráhy MAN v POS2
IncManPos3_Ax	pole REAL [mm/min] (x = 0,1,,15)	Aktuální přírůstek dráhy MAN v POS3
IncManPos4_Ax	pole REAL [mm/min] (x = 0,1,,15)	Aktuální přírůstek dráhy MAN v POS4
IncManPos5_Ax	pole REAL [mm/min] (x = 0,1,,15)	Aktuální přírůstek dráhy MAN v POS5

RTM. Servo [xx] .		
ICounter	pole REAL [mm] (xx = 0,1,..,15)	Diferenční čítač polohové servosmyčky
ICounter_S	pole INT32 [µm] (xx = 0,1,..,15)	Diferenční čítač polohové servosmyčky
ServoPosition	pole REAL [mm] (xx = 0,1,..,15)	Poloha servosmyček podle odměřování
ServoPosition_S	pole INT32 [µm] (xx = 0,1,..,15)	Poloha servosmyček podle odměřování
ServoPosition_Inc	pole INT32 [inc] (xx = 0,1,..,15)	Poloha servosmyček v inkrementech pohonu
ServoReqPosition	pole REAL [mm] (xx = 0,1,..,15)	Požadovaná poloha servosmyček
ServoReqPosition_S	pole INT32 [µm] (xx = 0,1,..,15)	Požadovaná poloha servosmyček
IncCan_Ax	pole INT32 [inc] (xx = 0,1,..,15)	Přírůstky dráhy pro CAN-BUS a EtherCAT trajektorie
Edrv .		EtherCAT pohon „EDRV“
State	pole UNS8 (xx = 0,1,..,15)	Stav EtherCAT pohonu EDRV (EDRV_INIT, EDRV_DONE,..)
MoveDisReq	pole BIT (xx = 0,1,..,15)	Žádost o blokování pohybu od pohonů EDRV
Error	pole BIT (xx = 0,1,..,15)	Chyba EtherCAT pohonu EDRV
Simul	pole BIT (xx = 0,1,..,15)	EDRV pohon v simulaci
Init	pole BIT (xx = 0,1,..,15)	Žádost o INIT EtherCAT pohonu typu EDRV
Done	pole BIT (xx = 0,1,..,15)	Žádost o DONE EtherCAT pohonu typu EDRV
Enable	pole BIT (xx = 0,1,..,15)	Žádost o ENABLE EtherCAT pohonu typu EDRV
Disable	pole BIT (xx = 0,1,..,15)	Žádost o DISABLE EtherCAT pohonu typu EDRV
On	pole BIT (xx = 0,1,..,15)	Žádost o ON (zapnutí) EtherCAT pohonu typu EDRV
Erase	pole BIT (xx = 0,1,..,15)	Žádost o ERASE EtherCAT pohonu EDRV
DriveStatus	pole UNS16 (xx = 0,1,..,15)	[in] „Drive status word“ EDRV pohonu (CoE, SoE)
DrivePos	pole INT32 [inc] (xx = 0,1,..,15)	[in] „Position feedback“ EDRV pohonu
DriveErrReg	pole UNS16 (xx = 0,1,..,15)	[in] „Drive error registr“ EDRV pohonu
DriveControl	pole UNS16 (xx = 0,1,..,15)	[out] „Master control word“ EDRV pohonu
DriveVeloReq	pole INT32 [inc] (xx = 0,1,..,15)	[out] „Velocity demand value“ EDRV pohonu
DrivePosReq	pole INT32 [inc] (xx = 0,1,..,15)	[out] „Position demand value“ EDRV pohonu
ScaleSpeed	pole REAL (x = 0,1,..,15)	[in] Měřitko pro výstup rychlosti EDRV pohonu

Enc .		Externí odměřování EtherCAT	
	CounterValue	pole INT32 [inc] (xx = 0,1,..,15)	[in] „Encoder counter value“ (EL5101 EDRV)
	LatchValue	pole INT32 [inc] (xx = 0,1,..,15)	[in] „Encoder latch value“ (EL5101 EDRV)
	LatchEn	pole BIT (xx = 0,1,..,15)	[out] „Encoder Enable latch“ (EL5101 EDRV)
	LatchValid	pole BIT (xx = 0,1,..,15)	[in] „Encoder Latch C valid“ (EL5101 EDRV)

RTM. Spindle .		
Spindle1Output	REAL [10V]	Výstupní napětí pro 1.vřeteno (bez procenta S)
Spindle2Output	REAL [10V]	Výstupní napětí pro 2.vřeteno (bez procenta S)
Spindle1PrsOutput	REAL [10V]	Výstupní napětí pro 2. vřeteno (s ohledem na procento S)
ReqSpeedPm	REAL [ot/min]	Požadované otáčky vřetene. (s ohledem na procento S)
ReqSpeedPt	REAL [ot/T]	Požadované otáčky vřetene. (s ohledem na procento S)
ActSpeedPm	REAL [ot/min]	Aktuální okamžité otáčky vřetene
ActSpeed2Pm	REAL [ot/min]	Aktuální okamžité otáčky 2.vřetene
ActAvgSpeedPm	REAL [ot/min]	Aktuální průměrované otáčky vřetene
ActAvgSpeed2Pm	REAL [ot/min]	Aktuální průměrované 2.otáčky 2.vřetene [ot/min]
ActSpeedPt	REAL [ot/T]	Aktuální okamžité otáčky vřetene
ActSpeed2Pt	REAL [ot/T]	Aktuální okamžité otáčky 2.vřetene
ActAvgSpeedPt	REAL [ot/T]	Aktuální průměrované otáčky vřetene
ActAvgSpeed2Pt	REAL [ot/T]	Aktuální průměrované otáčky 2.vřetene
SpindleGearAct1	UNS16	Číslo aktuálního převodového stupně pro 1.vřeteno
SpindleGearAct2	UNS16	Číslo aktuálního převodového stupně pro 2.vřeteno

RTM . Man .		
AckMan	BIT	Pomocné ruční pojezdy jsou aktivní (ACK AUTMAN)
InposStop	BIT	System je v poloze posledního stopu (INPOS_STOP)
ManEnable	BIT	Povolení ručních pojezdů (EN AUTMAN)
ManControlNC	BIT	Řízení MAN je z panelu NC systému (AUTMAN CONT NC)
ManControlKnob	BIT	Řízení MAN je z panýlku točítka (AUTMAN CONT TOC)
ManControlSelect	BIT	Předvolba pohybu MAN (AUTMAN CONT SELECT)
ManMovePlus_Ax[xx]	pole BIT (x = 0,1,..,5)	Pohyb v režimu MAN v kladném směru (MM_x0)
ManMoveMinus_Ax[xx]	pole BIT (x = 0,1,..,5)	Pohyb v režimu MAN v záporném směru (MM_x1)
ManControlRT	BIT	Řízení MAN rychloposuvem (AUTMAN CONT G00)
Shift_Ax[xx]	pole BIT (x = 0,1,..,5)	Pohyb v režimu SHIFT (SHIFT_CONTROL)
ExtManMovePlus_Ax[xx]	pole BIT (x = 0,1,..,5)	Externí pohyb v režimu MAN v kladném směru (EXM_x0)
ExtManMoveMinus_Ax[xx]	pole BIT (x = 0,1,..,5)	Externí pohyb v režimu MAN v záporném směru (EXM_x0)
ReqExtManRT	BIT	Externí požadavek pro G00 MAN (REQ_EXT_G00_AUTMAN)
ReqExtManSelect	BIT	Externí požadavek pro předvolbu REQ_EXT_SELECT_AUTMAN
ReqExtMan	BIT	Externí požadavek na pohyb (REQ_EXT_CONT_AUTMAN)
ReqExtFeedMan	BIT	Externí požadavek na rychlost (REQ_EXT_FEED_AUTMAN)
ExtFeedMan	REAL [mm/min]	Hodnota externí rychlosti v MAN (EXT_FEED_AUTMAN)
ExtFeedRTMan	REAL [mm/min]	Hodnota externí rychlosti v MAN (EXT_FEED_G00_AUTMAN)
ReqG95Man	BIT	Požadavek na otáčkový posuv pro MAN (G95_AUTMAN)
Knob1Disable	BIT	Zákaz ovládní z 1. panýlku (AUTMAN_KNOB1_DIS)
Knob2Disable	BIT	Zákaz ovládní z 2. panýlku (AUTMAN_KNOB2_DIS)

RTM.Control .		
MpPlc_Ax[xx]	pole BIT (xx = 0,1,..15)	Povolení pohybu z PLC pro osy
InRef_Ax[xx]	pole BIT (xx = 0,1,..15)	NC osa v referenci
MpMan_Ax[xx]	pole BIT (xx = 0,1,..15)	Povolení pohybu pro osy pro manuální režim (MPxMan)
Mp_Ax[xx]	pole BIT (xx = 0,1,..15)	Celkové povolení pohybu pro osy (MPx)
Move_Ax[xx]	pole BIT (xx = 0,1,..15)	Pohyb v osách (PO_OSxPI)
Direct_Ax[xx]	pole BIT (xx = 0,1,..15)	Směr pohybu pro osy (SM_POxPI)
Coupling_S[xx]	pole BIT (xx = 0,1,..15)	Vazba polohové servosmyčky (VAZBA_x)
Homing_Ax[xx]	pole BIT (xx = 0,1,..15)	Polohování osy (POLOHV_x)
StartDisable	BIT	Zákaz startu (START_DISABLE)
BlockInProgress	BIT	Blok rozpracován (PO_F)
PosReached	BIT	Programovaná poloha dosažena (PO_POL)
StopAck	BIT	Potvrzení stopu (PO_STOP)
InPos	BIT	Poloha v toleranci (INPOS)
ExtFeedOvr	BIT	Externí řízení FeedOverride (FEED_OVR)
ExtSpeedOvr	UNS16	Externí řízení procenta otáček z PLC (SPEED_OVR)
Delay	BIT	Prodleva G04
LimPlus_Ax[xx]	pole BIT (xx = 0,1,..15)	Limitní spínače v kladném směru (KHx0)
LimMinus_Ax[xx]	pole BIT (xx = 0,1,..15)	Limitní spínače v záporném směru (KHx1)
DragLimPlus_Ax[xx]	pole BIT (xx = 0,1,..5)	Zpomalovací limitní spínače v kladném směru (ZPx0)
DragLimMinus_Ax[xx]	pole BIT (xx = 0,1,..5)	Zpomalovací limitní spínače v záporném směru (ZPx1)
ModeAut	BIT	Režim AUT (AUTPI)
ModeRefer	BIT	Ržim REFER (REFPI)
ModeCanul	BIT	Režim CANUL (CAPI)
Thread	BIT	Závitování G33 (G33PI)
RapidTraverse	BIT	Rychloposuv G00 (G00PI)
SwLimPlus_Ax[xx]	pole BIT (xx = 0,1,..5)	SW limitní spínače v kladném směru (SLS_LIMIT_PLUS)
SwLimMinus_Ax[xx]	pole BIT	SW limitní spínače v záporném

	(xx = 0,1,..5)	směru (SLS_LIMIT MINUS)
SwDragPlus_Ax[xx]	pole BIT (xx = 0,1,..5)	SW zpomal. spínače v kladném směru (SLS_DRAG PLUS)
SwDragMinus_Ax[xx]	pole BIT (xx = 0,1,..5)	SW zpomal. spínače v záporném směru (SLS_DRAG MINUS)
SwRefEnable_Ax[xx]	pole BIT (xx = 0,1,..5)	Povolení softwarových spínačů od referencie (SLS_REFER)
HomingPoint_Ax[xx]	pole BIT (xx = 0,1,..5)	Referenční spínače (KRx)
DragHoming_Ax[xx]	pole BIT (xx = 0,1,..5)	Zpomalovací referenční spínače (ZPRx)
HomingRevers_Ax[xx]	pole BIT (xx = 0,1,..5)	Referenční spínače reverzační (KRRx)
M00	BIT	Dekódovaná funkce M00 1.skupina M (M00_PID)
M01	BIT	Dekódovaná funkce M01 1.skupina M (M01_PID)
M02	BIT	Dekódovaná funkce M02 1.skupina M (M02_PID)
M30	BIT	Dekódovaná funkce M02 1.skupina M (M30_PID)
M03	BIT	Dekódovaná funkce M03 2.skupina M (M03PI)
M04	BIT	Dekódovaná funkce M04 2.skupina M (M04PI)
M19	BIT	Dekódovaná funkce M19 2.skupina M (M19PI)
M41	BIT	Dekódovaná funkce M41 3.skupina M (M41PI)
M42	BIT	Dekódovaná funkce M42 3.skupina M (M42PI)
M43	BIT	Dekódovaná funkce M43 3.skupina M (M43PI)
M44	BIT	Dekódovaná funkce M44 3.skupina M (M44PI)
M07	BIT	Dekódovaná funkce M07 5.skupina M (M07PI)
M08	BIT	Dekódovaná funkce M08 5.skupina M (M08PI)
M50	BIT	Dekódovaná funkce M50 6.skupina M (M50PI)
M51	BIT	Dekódovaná funkce M51 6.skupina M (M51PI)
M10	BIT	Dekódovaná funkce M10 7.skupina M (M10PID)
M11	BIT	Dekódovaná funkce M11 7.skupina M (M11PID)
M06	BIT	Dekódovaná funkce M06 8.skupina M (M06PID)
M60	BIT	Dekódovaná funkce M60 8.skupina M (M60PI)
SF_Run	BIT	System v chodu, stroj jede, nebo jsou technologické funkce
SF_BlockInProgr	BIT	Blok rozpracován
SF_BlockFinished	BIT	Blok ukončen

SF_Stop	BIT	Všechny druhy stopu bloku
SF_NotInpos	BIT	Dosud nebylo dosaženo požadované polohy
SF_DelayInProgr	BIT	Časová prodleva
SF_IndicationMode	BIT	Indikační režim
SF_SimulationMode	BIT	Simulační režim
SF_HighspeedMode	BIT	Zrychlený režim
SF_CondstopMode	BIT	Podmíněný stop aktivní (pro bloky s M01)
SF_ManualMode	BIT	Manuální režim
SF_BlockByBlock	BIT	Režim blok po bloku
SF_LS	BIT	Limitní spínač
SF_SLS	BIT	Softwarový limitní spínač
SF_Backward	BIT	Couvání
SF_StopPosChanged	BIT	Bude nastaven, když při stopu programu bylo ručně popojeto
BlockToggle	BIT	Klopni bit pro nový blok
MeasStart	BIT	Start pro dávková měření
MeasStop	BIT	Stop pro dávková měření
MeasErr	BIT	Chyba dávkového měření
MeasControl	UNS32	Řízení pro dávková měření
BlockRestart	BIT	Start po stopu bloku
LimitFeedSpotReq	BIT	Požadavek na okamžitý limit (složkový)
G33Gradual	BIT	G33 s plynulým koncem (G33 -> G95)
TouchProbe0Stop	BIT	Stop pohybu od dotykové sondy
TouchProbe0	BIT	Dotyková sonda
KnobSel_Ax[xx]	pole UNS8 (xx = 0,1,..15)	Volba osy z točítka pro indikaci
PlcStatus	UNS8	Stav PLC
FeedOvrVal	UNS16	Hodnota promile pro externí řízení rychlosti
FeedOvrMultiplier	UNS16	Násobící koeficient pro externí řízení rychlosti
SpeedOvrVal	UNS16	Hodnota promile pro řízení otáček (SPEED_OVR_EXT)
SpeedOvrMultiplier	UNS16	Násobící koeficient pro otáčky

SF_PTYPE	UNS32	Výčet pro programovou transformaci
SF_WTYPE	UNS32	Výčet pro transformaci polotovaru
PLCPeriod	REAL [s]	Definovaná hodnota periody pro základní běh PLC (slow .. 20ms)
PLCPeriodFast	REAL [s]	Definovaná hodnota periody pro rychlý běh PLC (1 ms)
PLCPeriodFastest	REAL [s]	Definovaná hodnota periody pro nejrychlejší běh PLC (200 µs)
FastestModuleCount	UNS32	Diagnostický čítač průběhu MODULE FASTEST
FastestModulePresent	BIT	Modul MODULE_FASTEST je aktivní
FastestInterrupt	REAL [s]	Definovaná hodnota periody pro nejrychlejší přerušování (200 µs)
FlagG33	UNS8	Fáze průběhu závitování G33
LOCATE_ANGLE	INT32	Žádaný uhel pro tečnovou interpolaci
ACTUAL_ANGLE	INT32	Aktuální uhel pro tečnovou interpolaci
CANErrorCode	UNS8	Chybový registr pro PLC od CAN2
CANErrorNum	UNS8	Číslo jednotky v poruše na CAN2
CANTimeOutCode	UNS8	Chyba time-out pro PLC od CAN2
CANTimeOutNum	UNS8	Číslo jednotky v poruše time-out na CAN2
PLCMemBackup[xx]	pole UNS8 (xx = 0,1,..9999)	Zálohovaná oblast PLC paměti (PLC_MEM_BACKUP)
BSPProcNumber	UNS32	Číslo procesoru BSP
SECProcNumber	UNS32	Číslo procesoru SEC

27.5 Konfigurace pro logické spoje

Konfigurace pro logické spoje jsou v souboru typu „ChannelConfig“ v elementu „Connections“.

element Connections		Konfigurace pro logické (virtuální) spoje	
element Connection		Konfigurace pro jeden logický (virtuální) spoj.	
atribut Source		Identifikátor výstupu	
		Abcd	Textový řetězec, který slouží jako identifikátor výstupu
atribut Destination		Identifikátor vstupu	
		Abcd	Textový řetězec, který slouží jako identifikátor vstupu
atribut Invert		Inverze (platí pro bitové spoje)	
		0	logický vstup je při přímo propojen s výstupem (default)
		1	logický výstup je při propojení na logický vstup invertován
atribut Scale		Měřítko (platí pro „analogové“ spoje)	
		1.0	bez měřítka (default)
		xx.xx	Měřítko se uplatní při propojení logického výstupu na analogový logický vstup
atribut Offset		Posun (platí pro „analogové“ spoje)	
		0.0	bez posunu (default)
		xx.xx	Posun se uplatní při propojení logického výstupu na analogový logický vstup
atribut Connected		Propojení logického vstupu a výstupu	
		0	logický vstup není propojen s logickým výstupem (default)
		1	logický vstup je propojen s logickým výstupem

Příklad:

Příklad logických spojů: (flOut1 -> flIn1), (flOut1 -> INOUT08.OP0.Bit[7])
(flOut1 -> Bun1.Bit[2])

```
<Connections>
```

```
<Connection Source="PLC.Output.flOut1" Destination="PLC.Input.flIn1"
Invert="0" Connected="1"></Connection>
```

```
<Connection Source="PLC.Output.flOut1"
Destination="CAN[1].INOUT08.OP0.Bit[7]" Connected="1"></Connection>
```

```
<Connection Source="PLC.Output.flOut1"
Destination="PLC.Input.Bun1.Bit[2]" Connected="1"></Connection>
```

```
</Connections>
```

27.6 Definice logických vstupů a výstupů v PLC programu

Pro definici logických (bitových i analogových) vstupů a výstupů s možností propojování slouží instrukce **DEF_IN** a **DEF_OUT**. Instrukce musí být umístěny (nebo volány) v inicializačním modulu **MODULE_INIT**. Instrukce automaticky definují vstupní nebo výstupní buňku podle zadaného typu (může být i pole) a automaticky nastaví přímé sdílení pro proměnné (už se nemusí použít instrukce **SHARE_VAR**).

instrukce	DEF_IN DEF_OUT
-----------	---------------------------------

funkce	DEF_IN	Definice logického vstupu s možností propojování
	DEF_OUT	Definice logického výstupu s možností propojování

syntax	DEF_IN (DEF_OUT)	val , poin , typ
	DEF_IN (DEF_OUT)	val , `TEXT` , typ
	DEF_IN (DEF_OUT)	val[x] , `TEXT` , typ
	DEF_IN (DEF_OUT)	val , `TEXT` , typ [, def]
	DEF_IN (DEF_OUT)	val , `TEXT` typ [, def , period]

1. parametr	„val“	název bitu nebo proměnné
2. parametr	„poin, TEXT“	pointer nebo textový řetězec identifikátoru
3. parametr	„typ“	zadání typu proměnné (viz dále)
4. parametr	„def“	defaultní hodnota
5. parametr	„period“	požadovaná doba obsluhy spoje (viz dále)

parametr	název	význam	typ
1.	val	Název bitu nebo datové proměnné pro vstup nebo výstup. Proměnná se automaticky definuje a proto v PLC programu se už nepoužívá definice pomocí instrukce DFM, a DS. Název může obsahovat počet prvků pole v hranaté závorce. Pokud se požaduje pole, automaticky se vymezi prostor pro celé pole s ohledem na typ proměnné.	Data podle typu
2.	poin	Ukazatel na buffer, kde je umístěný text s klíčovým slovem konfiguračního parametru - náveští u řetězce definovaného instrukcí "str".	pointer
	text	Přímé zadání textu s klíčovým slovem konfiguračního parametru v apostrofech	řetězec
3.	typ	Zadání typu proměnné (viz dále)	klíčové slovo
4.	def	Nepovinný parametr. Přímé zadání defaultní hodnoty, číslo se naplní do vstupu nebo výstupu.	číselná hodnota
5.	period	Nepovinný parametr pro požadovanou obsluhu virtuálního spoje. Mohou být použity klíčová slova: SLOW, FAST, FASTEST, ONDEMAND (viz dále)	klíčové slovo

Tabulka typů proměnné pro sdílení (3.parametr instrukce DEF_IN, DEF_OUT):

Typ	popis
TYPE_BIT	bitová proměnná
TYPE_INT 8	celočíslný typ se znaménkem – 8 bitů
TYPE_UNSC 8, TYPE_BYTE	celočíslný typ bez znaménka – 8 bitů
TYPE_INT 16	celočíslný typ se znaménkem – 16 bitů
TYPE_UNSC 16, TYPE_WORD	celočíslný typ bez znaménka – 16 bitů
TYPE_INT 32	celočíslný typ se znaménkem – 32 bitů
TYPE_UNSC 32, TYPE_DWORD	celočíslný typ bez znaménka – 32 bitů
TYPE_INT 64	celočíslný typ se znaménkem – 64 bitů
TYPE_UNSC 64, TYPE_QWORD	celočíslný typ bez znaménka – 64 bitů
TYPE_REAL	reálný typ
TYPE_STR	textový řetězec
TYPE_BIN	binární řetězec

Požadovaná obsluha virtuálního spoje (5.parametr instrukce DEF_IN, DEF_OUT):

Period	popis
SLOW	Systém zabezpečí periodickou obsluhu virtuálního spoje v taktu základní smyčky PLC (20 ms). Defaultní obsluha. (viz dále).
FAST	Systém zabezpečí periodickou obsluhu virtuálního spoje v rychlém taktu interpolátoru a servosmyček (1 ms). Pro rychlou obsluhu spoje je nutné, aby logický vstup a současně i logický výstup měly nastaveny atributy FAST. (viz dále).
FASTEST	Systém zabezpečí periodickou obsluhu virtuálního spoje v nejrychlejším taktu pro EtherCAT (250 us). (viz dále).
ONDEMAND	Na vyžádání. Systém nezabezpečí periodickou obsluhu spoje. PLC program má možnost okamžitě zjistit stav logického vstupu, na který směřuje virtuální spoj, pomocí instrukce UPDATE_IN (viz dále).

Příklad:

MODULE_INIT

```

DEF_IN    inLimX,    `inLimX`,    TYPE_BIT, -, FAST
DEF_IN    flBitIn1, `BitIn1`,    TYPE_BIT
DEF_OUT   flBitOut, `BitOut`,    TYPE_BIT, 1

DEF_IN    dwBunIn1, `BunIn1`,    TYPE_DWORD, 222, ONDEMAND
DEF_OUT   rBunOut1, `BunOut1`,    TYPE_REAL, 1.7

DEF_IN    dwBunIn2[5], `BunIn2`,    TYPE_DWORD
DEF_OUT   rBunOut2[8], `BunOut2`,    TYPE_REAL

```

MODULE_INIT_END

27.7 Stav logického vstupu na vyžádání z PLC programu

PLC program má možnost zjistit okamžitě stav logického vstupu, na který je nastaven „virtuální spoj“. Čas zjištění aktuální hodnoty logického vstupu je tak určen PLC programem a nevyužije se periodická obsluha spoje pomocí systémových prostředků.

Požadovaná obsluha virtuálního spoje se udává jako 5. parametr v instrukcích DEF_IN a DEF_OUT. V tomto případě se musí nastavit do 5. parametru klíčové slovo „ONDEMAND“.

instrukce	UPDATE_IN
------------------	------------------

funkce **UPDATE_IN** **Okamžité zjištění stavu propojeného logického vstupu**

syntax **UPDATE_IN val**
 UPDATE_IN val[x]

1.parametr „**val**“ **název bitu nebo proměnné**

parametr	název	význam	typ
1.	val	Název bitu nebo datové proměnné pro vstup. Název proměnné musí být stejný, jaký je použit u instrukci DEF_IN, kde je proměnná definovaná a kde jsou další informace potřebné pro logický spoj. Název může obsahovat číslo prvku v hranaté závorce. V tomto případě se jedná a prvek pole a zjistí se aktuální stav jen toho jednoho prvku z pole.	Data podle typu

Příklad:

V inicializaci jsou definovány logické vstupy:

```
DEF_IN        flBitIn1,    `BitIn1`,    TYPE_BIT,    -,    ONDEMAND
DEF_IN        dwBunIn1,    `BunIn1`,    TYPE_DWORD, 222, ONDEMAND
DEF_IN        rBunIn2[10], `BunIn2`,    TYPE_REAL,    -,    ONDEMAND
```

Na vhodných místech v PLC programu je čtení log.vstupů „na vyžádání“.

```
UPDATE_IN    flBitIn1
UPDATE_IN    dwBunIn1
UPDATE_IN    rBunIn2[6]
```

27.8 Perioda obsluhy virtuálního spoje

Systém zabezpečí periodickou obsluhu virtuálního spoje. Jako požadovaná perioda obsluhy spoje se nastaví maximální hodnota prvků, mezi kterými je spoj definován.

27.8.1 Perioda obsluhy „SLOW“

Systém zabezpečí periodickou obsluhu virtuálního spoje v taktu základní smyčky PLC (20 ms). Jedná se o defaultní periodu obsluhy (instrukce `DEF_IN` a `DEF_OUT` nemusí mít 5. parametr vyplněný).

Propojení spojů s periodou SLOW se provede před voláním běžného průchodu PLC a také po volání běžného průchodu PLC.

27.8.2 Perioda obsluhy „FAST“

Systém zabezpečí periodickou obsluhu virtuálního spoje v taktu interpolátoru, servosmyček a PLC modulu `MODULE_FAST` (1 ms). Propojení spojů s periodou FAST se provede tehdy, když instrukce `DEF_IN` a `DEF_OUT` mají 5. parametr nastavený na FAST a druhý prvek spoje má možnost periody obsluhy alespoň FAST nebo FASTEST.

Data propojované s periodou FAST jsou určeny pro zpracování v modulu `MODULE_FAST` a propojení spojů se provede před jeho průchodem. To samé platí i pro případné použití funkce `RTMDLL: RtmFunServo()`.

Pro cizí reálný čas RTX může být v PLC použit modul `MODULE_CYCLICOPERATION`, který běží v taktu FAST (1 ms). Tento modul se časově zpracovává na začátku obsluhy smyčky FAST. V tomto případě propojení spojů se provede i před průchodem tohoto modulu. Druhé propojení spojů se provede standardně po jeho průchodu a před průchodem `MODULE_FAST`.

27.8.3 Perioda obsluhy „FASTEST“

Perioda obsluhy FASTEST se používá jen pro cizí reálný čas RTX. Systém zabezpečí periodickou obsluhu virtuálního spoje v taktu FASTEST, který je definován nastavením v registru `FastestPeriod`. Pokud je v registru defaultní hodnota, tak se použije nastavení podle `MasterCycleTime` pro sběrnici EtherCAT. Defaultní hodnota pro periodu `MasterCycleTime` je 1000 μ s (podle cyklu EtherCAT Mastra). Perioda může nabývat hodnot: 100, 200, 250, 500, 1000 μ s.

Pokud existují virtuální spoje FASTEST, takže instrukce `DEF_IN` a `DEF_OUT` mají 5. parametr nastavený na FASTEST a druhý prvek spoje má možnost periody obsluhy FASTEST, systém v tomto rastru volá modul `MODULE_FASTEST` a také funkci `RTMDLL: RtmFunPlcFastest()`. Propojení spojů FASTEST se provede před i po průchodu modulu `MODULE_FASTEST`.