

16

16. PLC KONFIGURACE A KONSTANTY

16.1 Konfigurace pro PLC program

PLC program má k dispozici pro přístup ke své konfiguraci instrukce **CNF_GET_INT**, **CNF_GET_REAL**, **CNF_GET_STR** a **CNF_GET_BIN**. Konfigurace pro PLC program je uložena například v souboru **Channelconfig**, který má XML formát pod elementem **Plc**. Zápis parametrů konfigurace není povinný a pokud se příslušný parametr v konfiguračním souboru nevyskytuje, v parametru pro PLC se objeví defaultní hodnota.

Konfigurace je pro PLC k dispozici už v době průchodu modulu **MODULE_INIT**, proto se doporučuje načíst celou konfiguraci v proceduře, která je zavolaná z modulu **MODULE_INIT**.

element Plc		Konfigurace pro PLC	
	element PlcParam	Jeden parametr konfigurace pro PLC.	
	atribut ID	Identifikátor parametru	
		Abcd	Textový řetězec, který slouží jako identifikátor parametru
	atribut Type	Typ parametru konfigurace	
		Int	Celočíselné a logické hodnoty
		Real	reálné hodnoty
		String	Textové řetězce
		Binary	Binární data
		Hodnota parametru	
		xx	Hodnota se zapisuje jako obsah elementu

Příklad:

Příklad zápisu parametrů PLC konfigurace v XML tvaru:

```
<Plc>
  <PlcParam ID="LIM_SWITCH" Type="Int">1</PlcParam> <!-- lim spínač -->
  <PlcParam ID="TIM_WATER" Type="Int">4</PlcParam> <!-- prodlení vody -->
</Plc>
```

Instrukce pro načtení konfigurace pro PLC:

instrukce	CNF_GET_INT
	CNF_GET_REAL
	CNF_GET_STR
	CNF_GET_BIN
	CNF_GET_BIT

funkce	CNF_GET_INT	Načtení celočíselné hodnoty z konfigurace
	CNF_GET_REAL	Načtení reálné hodnoty z konfigurace
	CNF_GET_STR	Načtení textového řetězce z konfigurace
	CNF_GET_BIN	Načtení binárního řetězce z konfigurace
	CNF_GET_BIT	Načtení bitové hodnoty z konfigurace
syntax	CNF_GET_xx	poin1, poin2 [, poin3]
	CNF_GET_xx	val1, poin2 [, val3]
	CNF_GET_xx	poin1, 'TEXT2' [, immed3]
	CNF_GET_xx	poin1, 'TEXT2' [, 'TEXT3']
1.parametr	„val1,poin1“	pointer nebo název cílové proměnné
2.parametr	„poin2,text“	pointer nebo textový řetězec identifikátoru
3.parametr	„val3,poin3,immed“	pointer, název nebo přímo defaultní hodnota

Vrácené datové hodnoty z konfigurace se zapisují do řetězce na který ukazuje parametr **poin1**, nebo přímo do datové proměnné **val1** a **bit**.

Všechny instrukce mají návratové hodnoty:

RLO=1 ... operace dokončena bez chyb

RLO=0 ... operace dokončena, ale konfigurační parametr se nenašel. Instrukce vrátí defaultní hodnotu.

parametr	název	význam	typ
1.	poin1	Ukazatel na buffer, do kterého se zkopírují požadovaná data, - návestí u řetězce definovaného instrukcí "str". Parametr může mít zadán offset v řetězci (+xx).	pointer
	val1	název datové proměnné do které se zkopírují požadovaná data, - typ BYTE, WORD, DWORD,...	data
	bit1	název bitové proměnné, která se nastaví podle konfigurace	bit
2.	poin2	Ukazatel na buffer, kde je umístěn text s klíčovým slovem konfiguračního parametru - návestí u řetězce definovaného instrukcí "str".	pointer
	text2	Přímé zadání textu s klíčovým slovem konfiguračního parametru v apostrofech	řetězec
3.	poin3	Ukazatel na buffer pro defaultní hodnotu, z které se zkopírují data do parametru 1, pokud se nenajde požadované klíčové slovo v konfiguraci. - návestí u řetězce definovaného instrukcí "str". Parametr může mít zadán offset v řetězci (+xx).	pointer
	val3	Název datové proměnné pro defaultní hodnotu, z které se zkopírují data do parametru 1, pokud se nenajde požadované klíčové slovo v konfiguraci. - typ BYTE, WORD, DWORD,...	data
	text3	Přímé zadání defaultního textu, který se zkopíruje do parametru 1, pokud se nenajde požadované klíčové slovo v konfiguraci	řetězec
	immed	Přímé zadání defaultní hodnoty, číslo se naplní do parametru 1 Reálná hodnota se zadává s desetinnou tečkou	číselná hodnota

výčet možností syntaxe:

```

CNF_GET_xxx      poin1, poin2
CNF_GET_xxx      poin1, poin2,   poin3
CNF_GET_xxx      poin1, poin2,   val3
CNF_GET_xxx      poin1, poin2,   'TEXT3'
CNF_GET_xxx      poin1, poin2,   immed

CNF_GET_xxx      vall1,  poin2
CNF_GET_xxx      vall1,  poin2,   poin3
CNF_GET_xxx      vall1,  poin2,   val3
CNF_GET_xxx      vall1,  poin2,   'TEXT3'
CNF_GET_xxx      vall1,  poin2,   immed

CNF_GET_xxx      poin1, 'TEXT2'
CNF_GET_xxx      poin1, 'TEXT2', poin3
CNF_GET_xxx      poin1, 'TEXT2', val3
CNF_GET_xxx      poin1, 'TEXT2', 'TEXT3'
CNF_GET_xxx      poin1, 'TEXT2', immed

CNF_GET_xxx      vall1,  'TEXT2'
CNF_GET_xxx      vall1,  'TEXT2', poin3
CNF_GET_xxx      vall1,  'TEXT2', val3
CNF_GET_xxx      vall1,  'TEXT2', 'TEXT3'
CNF_GET_xxx      vall1,  'TEXT2', immed

```

Příklady:

; různé možnosti načtení konfigurace

```

CNF2:      STR    20, 'DOBA_MAZANI'
TEXT1:     STR    20
BITX:      DFM    ,,,,bit_A,,,
dwTime:    DS     4
rBun1:     DS     8      ;pro reálné hodnoty
rBunDef:    DS     8

CNF_GET_INT      dwTime, CNF2, 125
CNF_GET_INT      dwTime, 'DOBA_MAZANI', 125
CNF_GET_STR      TEXT1, 'TextMazani','Probiha mazani'

CNF_GET_REAL     rBun1, 'Zesileni', cnst.0.2
                                   ;reálná defaultní hodnota

```

; načte konfiguraci podle předchozího příkladu

```

CNF_GET_BIT      bit_A, 'LIM_SWITCH', 0 ;načte bitovou proměnnou
CNF_GET_INT      dwTime, 'TIM_WATER', 6  ;načte prodlení pro vodu

```

16.2 Konstanty pro PLC program

PLC konstanty slouží pro symbolické definování konstant, které mají být k dispozici jak v PLC programu, tak v „Panelu“ systému. Konstanty lze například využít na definici příkazů pro PLC nebo na definici symbolických offsetů pro orientaci ve sdílené paměti mezi PLC programem a Panelem systému (instrukce SA_READ a SA_WRITE).

PLC konstanty pro definici offsetů ve sdílené paměti jsou zadány v souboru „SAVars.PlcConstants“.
PLC konstanty pro definici příkazů pro PLC jsou zadány v souboru „Commands.PlcConstants“.

Všechny PLC konstanty se zadávají v XML tvaru pomocí elementu PlcConstants.

element PlcConstants		Konstanty pro PLC	
	element Constant	Jedna konstanta	
	atribut ID	Identifikátor parametru	
		Abcd	Textový řetězec, který slouží jako identifikátor konstanty
	atribut Value	Hodnota konstanty	
		xx	Přímo číselná hodnota PLC konstanty
		SA0INT	Konstanta je offset na DWORD ve sdílené paměti
		SA0REAL	Konstanta je offset na REAL ve sdílené paměti
		PLCCOMMAND	Konstanta je příkaz pro PLC

Pokud atribut „Value“ neobsahuje přímo číselnou hodnotu konstanty, ale některou z identifikátorů „SA0INT, SA0REAL a PLCCOMMAND“ dojde k automatickému číslování konstant s ohledem na typ. Tato metoda se doporučuje, protože nemůže docházet k překrytí číselných hodnot konstant.

Příklad:

Příklad pro konstanty offsetů ve sdílené paměti:

```
<PlcConstants>

  <Constant ID="OFFS_R_FEED" Value="SA0REAL"></Constant>  <!-- Feed -->
  <Constant ID="OFFS_I_STATE" Value="SA0INT"></Constant>   <!-- Stav -->

</PlcConstants>
```

Příklad pro konstanty příkazů pro PLC:

(příklad s přímým zadáním hodnot konstant, lepší je použít Value="PLCCOMMAND"):

```
<PlcConstants>

  <Constant ID="PUMP_START" Value="12"></Constant> <!-- Zapne pumpu -->
  <Constant ID="JET1_UP" Value="13"></Constant>   <!-- JET1 nahoru -->

</PlcConstants>
```

PLC konstanty zabezpečí provázanost mezi PLC programem, systémem a dialogovými okny.

Různé příklady použití PLC konstant mimo PLC program:

PLC konstantu pro příkaz budeme označovat: "Cmd"

PLC konstantu pro offset ve sdílené paměti: "Offs"

typ souboru	atribut s PLC konstantou	popis
*.KbdConfig	PlcCommand="Cmd"	Tlačítko pro PLC program definované v technologické části tlačítek panelu jako atribut elementu KeyConfig
*.HTML	PlcCommand="Cmd"	Softwarové tlačítko pro PLC program definované v dialogovém okně jako atribut v elementu BUTTON
*.SoftMenu	PlcCommand="Cmd"	Softwarové tlačítko pro PLC program definované v softwarovém menu jako atribut v elementu SoftMenuItem
*.HTML	Offset="Offs"	Vstupní hodnota z dialogového okna, která se запиše na zadaný offset do sdílené paměti, definovaný jako atribut elementu INPUT
*.HTML	Offset="Offs"	Zobrazovaná hodnota v okně, která se zobrazuje ze zadaného offsetu ze sdílené paměti, definovaný např. jako atribut elementu DIV s atributem: CNCType = "AsyncIndicatorPainter"
*.HTML	Offset="Offs"	Zobrazení obrázků podle hodnoty proměnné ve sdílené paměti, definovaný jako atribut elementu IMG s atributem: CNCType = "AsyncIndicatorPainter"

Všechny soubory typu „PlcConstants“ musí být zařazeny v projektu PLC a v PLC SETUPu musí být zabezpečeno, že se všechny přkopírují z projektu do adresáře PLC v prostředí „CNC Machine Files“. Proto v souboru „PLC.NSI“ musí být:

Část pro instalaci souborů:

Function InstallPlcFiles

File "/oname=Plc\Commands.PlcConstants" "..\Plc\Commands.PlcConstants"

File "/oname=Plc\SAVars.PlcConstants" "..\Plc\SAVars.PlcConstants"

Část pro odinstalování:

Function un.InstallPlcFiles

Delete "\$INSTDIR\Plc\Commands.PlcConstants"

Delete "\$INSTDIR\Plc\SAVars.PlcConstants"

instrukce	CONST_GET
------------------	------------------

funkce	CONST_GET	Načtení PLC konstanty
syntax	CONST_GET	val1, poin2
	CONST_GET	val1, 'TEXT2'
1.parametr	„val1“	název cílové proměnné kam se načte konstanta
2.parametr	„poin2,text“	pointer nebo textový řetězec identifikátoru

Vrácená datová hodnota PLC konstanty se naplní do DR registru (DWORD). Návrátové hodnoty jsou:

RLO=0, DR=0 PLC konstanta se nenašla
RLO=1, DR operace dokončena v pořádku

Konstanty pro PLC jsou k dispozici už v době průchodu modulu MODULE_INIT, proto se doporučuje načíst všechny konstanty v proceduře, která je zavolána z modulu MODULE_INIT.

parametr	název	význam	typ
1.	val1	název datové proměnné do které se zkopírují požadovaná data, - typ BYTE, WORD, DWRD	data
2.	poin2	Ukazatel na buffer, kde je umístěný text ID PLC konstanty - návěští u řetězce definovaného instrukcí "str".	pointer
	text	Přímé zadání textu ID PLC konstanty v apostrofech	řetězec

Konstanty příkazů se v PLC programu zpracovávají pomocí událostní procedury **_ON_CMD**.
(viz kapitolu „Základní instrukce“).

Procedura se zavolá automaticky při stisku tlačítka nebo softwarového menu, které mají v konfiguraci nastaven "PLCCommand". Procedura v PLC programu je nepovinná. Umístění procedury může být v libovolném souboru. Překladač TECHNOLOG zkoumá existenci takovéto procedury a v případě, že existuje, spustí ji automaticky při stisku tlačítka.

Vstupní parametry procedury jsou:

RLO = 1 tlačítko stisknuto
RLO = 0 tlačítko puštěno
DR (DWORD) ... kód z konfigurace "PLCCommand"
kódem je PLC konstanta načtená pomocí CONST_GET

Konstanty symbolických offsetů pro orientaci ve sdílené paměti se použijí v parametru instrukce **SA_READ** a **SA_WRITE**.

Příklad:

Příklad načtení konstant z předchozího příkladu:

```

OFFS_R_FEED:      DS    2      ;jsem se načte offset pro FEED
OFFS_I_STATE:     DS    2      ;jsem se načte offset pro STATE
CMD_PUMP_START:   DS    4      ;buňka pro kód příkazu PUMP_START
CMD_JET1_UP:      DS    4      ;buňka pro kód příkazu JET1_UP
R_FEED:           DS    8      ;buňka pro načtení reálné hodnoty FEED
STATE:            DS    4      ;buňka pro načtení DWORD hodnoty STATE

PROC_BEGIN Inicializace
  CONST_GET  OFFS_R_FEED, 'OFFS_R_FEED'      ;načte offset pro FEED
  LA         FL_OK
  WR         FL_OK
  CONST_GET  OFFS_I_STATE, 'OFFS_I_STATE'     ;načte offset pro STATE
  LA         FL_OK
  WR         FL_OK
  CONST_GET  CMD_PUMP_START, 'PUMP_START'     ;načte kód pro PUMP_START
  LA         FL_OK
  WR         FL_OK
  CONST_GET  CMD_JET1_UP, 'JET1_UP'           ;načte kód pro JET1_UP
  LA         FL_OK
  WR         FL_OK

  LDR        -FL_OK
  ESET1      2                               ; PLC0002: Načítání PLC konstant se nepodařilo
PROC_END Inicializace

```

Příklad pro rozkódování příkazů v událostní proceduře _ON_CMD.

;Událostní procedura pro příkazy PLC

```

PROC_BEGIN _ON_CMD
  JL0        TL_PUSTENO                     ;tlačítko puštěno ?
  EQ         CMD_PUMP_START                 ;je start pumpy ?
  FL1        1,M_PUMP_START                 ;mechanismus start pumpy
  EQ         CMD_JET1_UP                    ;je příkaz JET1 nahoru ?
  FL1        1,JET1_UP_O                    ;výstup JET1 nahoru - začátek
  JUM        TL_END

  TL_PUSTENO:                               ;pro konec příkazu
  EQ         CMD_JET1_UP                    ;je příkaz JET1 nahoru ?
  FL1        0,JET1_UP_O                    ;výstup JET1 nahoru - konec
  TL_END:
PROC_END _ON_CMD

```

Příklad pro čtení se sdílené oblasti pomocí instrukci SA_READ a SA_WRITE.

```

MECH_BEGIN M_SA_READ
  EX
  SA_READ    0,OFFS_R_FEED,8,R_FEED        ;načte reální hodnotu FEED
  EX0
  SA_READ    0,OFFS_I_STATE,4,STATE        ;načte DWORD hodnotu STATE
  EX0
MECH_END M_SA_READ

```