

# Programování PLC Tecomat

## Spuštění vývojového prostředí Mosaic pro tvorbu programu

Ikonou na ploše spustíte program **Mosaic**. Při výběru projektu vyberte projekt **Uloha1.mpr** v adresáři Uloha1 pro úlohu **L1 - Otočný stůl**, resp. **Uloha2.mpr** v adresáři Linka pro úlohu **L5 - Pneumomotor s přísavkou**. Otevře se editor souborů. Jako jedno okno se záložkou se otevře i soubor **Main.mos**, do kterého je možno mezi instrukce **P0** a **E0** psát program.

## Spuštění vytvořeného programu

Program přeložíte klávesou **F9**, přenesete do automatu klávesovou zkratkou **Shift-F9** a následně spustíte v automatu pomocí kláves **Ctrl-F9**.

## Instrukční soubor PLC Tecomat

Pro řešení zadané úlohy postačí tyto instrukce:

instrukce	význam
<b>LD</b>	načtení hodnoty proměnné
<b>LDC</b>	načtení negované hodnoty proměnné
<b>WR</b>	zápis hodnoty do proměnné
<b>WRC</b>	zápis negované hodnoty do proměnné
<b>AND</b>	logický součin
<b>ANC</b>	logický součin s negovaným operandem
<b>OR</b>	logický součet
<b>ORC</b>	logický součet s negovaným operandem
<b>SET</b>	podmíněné nastavení proměnné
<b>RES</b>	podmíněné nulování proměnné

Každá instrukce má jediný argument, druhým argumentem je vnitřní registr (zásobník), do kterého se po provedení logické operace uloží její výsledek. Například **AND** provádí logický součin mezi výsledkem minulé operace (uloženým v zásobníku) a svým argumentem a výsledek je uložen zpět do zásobníku.

V programu lze použít vedle absolutního i symbolické adresování. Absolutní adresy se skládají z písmene (X – vstup, Y – výstup, S – systémové registry, R – uživatelské registry ad.) a číslic vztahujících se ke konkrétnímu vstupu, výstupu atd.. Symbolické označení proměnné jménem se zadává pomocí direktivy #reg následované specifikací typu proměnné (bit, byte, word, long, float) a provádí se obvykle mimo základní proces P0.

## Několik příkladů k objasnění problematiky

Realizace vztahu  $Y = A \cdot \bar{B}$  :

LD            A  
ANC          B  
WR            Y

Realizace vztahu  $Y = \bar{A} + B$  :

LDC          A  
OR            B  
WR            Y

Realizace paměťové funkce set  $Y_{\text{set}} = A + B$  :

```
LD      A
OR      B
SET     Y
```

V případě platnosti vztahu  $A + B$  nastaví automat na výstup Y logickou jedničku, jinak je instrukce SET při vykonávání programu přeskakována.

Realizace paměťové funkce reset  $Y_{\text{reset}} = A \cdot B$  :

```
LD      A
AND     B
RES     Y
```

V případě platnosti vztahu  $A \cdot B$  nastaví automat na výstup Y logickou nulu, jinak je instrukce RES při vykonávání programu přeskakována.

Realizace vztahu se závorkami  $Y = ((A + B) + C \cdot \bar{D}) \cdot (E)$  :

```
LD      A
OR      B
LD      C
AND     D
OR
LD      E
AND
WR      Y
```

Protože automat vyhodnocuje operace jednu po druhé a dále pracuje s výsledkem předchozí operace uložené v zásobníku, je výhodné předchozí vztah přepsat na tvar  $Y = (C \cdot \bar{D} + A + B) \cdot E$  .

Realizace tohoto vztahu:

```
LD      C
AND     D
OR      A
OR      B
AND     E
WR      Y
```

**Pozn. 1:** Zdrojový kód lze zkopírovat jako text přes schránku do jiného programu, např. do MS Word, Poznámkového bloku apod..

**Pozn. 2:** Ve vývojovém prostředí Mosaic lze přepnout zobrazení programu z jazyka symbolických instrukcí do jazyka kontaktních schémat.