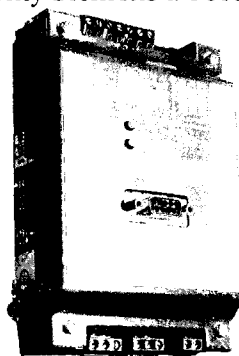


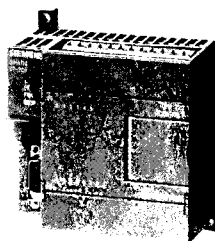
**Lehký úvod do
programování PLC
v jazyce IL**

Co je to PLC?

Programovatelný logický automat či **PLC** (z anglického **Programmable Logic Controller**) je malý specializovaný počítač používaný pro automatizaci procesů v reálném světě, jako je řízení strojů nebo výrobních linek (či jiných technologických procesů). V naší laboratoři se setkáte s automaty Simatic S7 od firmy Siemens a Tecomat od české firmy Teco.



Obrázek 1: Tecomat



Obrázek 2: Simatic

Každý počítač zpracovává informace, stejně je tomu i u PLC. Ten je ale navíc schopen získávat informace přímo z technologického procesu (k tomu slouží **vstupní signály**, které jsou přivedeny na **vstupní svorky** automatu) a také tento proces přímo ovlivňovat (k tomu slouží **výstupní signály** na **výstupních svorkách**). Stručně je označujeme jako **vstupy** a **výstupy**. Aby automat mohl s informacemi manipulovat, ukládají se do jeho vnitřní datové **paměti**. Soulad obsahu paměti se skutečným stavem vstupů a výstupů zajišťuje **systémový program** automatu. Zjišťuje stav signálů na vstupních svorkách a zapisuje jej do určité vyhrazené oblasti paměti (**obrazu vstupů**) a naopak z jiné oblasti (**obrazu výstupů**) čte hodnoty výstupů a generuje odpovídající signály na výstupních svorkách. To, jaké mají být hodnoty výstupů se „vypočítá“ provedením **uživatelského programu**, který do paměti automatu uložíme – a tento text se právě zabývá tím, jak takový program vytvořit. Čtení vstupů, vyhodnocení uživatelského programu a následný zápis výstupních hodnot se neustále opakuje v **cyklu automatu**. Uživatelský program je jen jednoduchý postup, jak se od hodnot vstupů „dopočítat“ k hodnotám výstupů.

Zápisníková paměť

Jak již bylo řečeno, automat má vnitřní paměť. Její část určená pro ukládání dat se nazývá **zápisníková paměť**. Tu můžeme dále členit na oblasti podle druhu dat. Jsou zde oblasti obrazu vstupů, obrazu výstupů a další. Oblasti se značí písmeny, uvedenými v tabulce

Oblast zápisníkové paměti	Simatic	Tecomat
Obraz vstupů	I	X
Obraz výstupů	Q	Y
Pomocné proměnné	M	R

Oblast obsahuje více paměťových buněk, na které se odkazujeme pořadovými čísly, počínaje nulou. Například **X0** je první buňka obrazu vstupů Tecomatu. Paměťové buňky mají tradičně velikost 1 bajt, tedy 8 bitů. Pro uložení jedné logické (= binární) hodnoty postačí jeden bit, můžeme se odkazovat i na jednotlivé bity jejich pořadovým číslem odděleným tečkou. Pak třeba **Q2.0** bude označovat první bit třetí paměťové buňky obrazu výstupů Simaticu. Označení oblasti a pořadového čísla buňky a případně i bitu tvoří **adresu** v zápisníkové paměti. Jaký signál je připojen k určité svorce a jaká adresa mu odpovídá určuje projektant řídicího systému. Je to dáno hardwarovým zapojením automatu a případně i jeho softwarovou konfigurací. Pro přehlednost bývá zvykem všechny použité signály pojmenovat tzv. **symbolickými názvy**. Místo adresy signálu zavedeme jméno, které bude vystihovat jeho význam. Tak třeba pro signál s adresou **Q25.4** spouštějící motor použijeme název „**MOTOR**“. **Přiřazení symbolických názvů** bývá součástí projektu, ale může si jej i vytvořit programátor. V případě našich laboratorních úloh je provedeno a obsaženo v zadání úlohy. Dobrou zásadou při **programování je používat pouze symbolické názvy**, které představují **logické proměnné**.

Trocha teorie...

Naším cílem bude realizovat logické řízení. Tento typ řízení lze matematicky popsat **logickou funkcí**. Existuje mnoho forem vyjádření logické funkce. Jazyk IL, který budeme používat, vychází z algebraického zápisu Booleovou algebrou. Logické funkce řízení pro jednotlivé výstupy automatu musíme tedy vyjádřit prostředky této algebry. Logické proměnné už máme definovány symbolickými názvy. Pro vyjádření funkce pak používáme formule s použitím základních algebraických operací – negace, součtu a součinu. Nejběžněji používáme **infixovou notaci**, kdy znaménka operací zapisujeme **mezi operandy**. Při vyhodnocování algebraických formulí respektujeme prioritu logických operací – první vyhodnocujeme negace (nejvyšší priorita), pak součiny a nakonec součty (nejnižší priorita). Pokud má být pořadí vyhodnocení jiné, použijeme závorky, které mají prioritu ještě vyšší. Tak např. ve funkci $y = a + \bar{b} \cdot c$ se nejdříve vypočte negace, pak součin a nakonec součet. U funkce $z = (a + \bar{b}) \cdot c$ se nejdříve vyhodnotí obsah závorky (negace, součet) a teprve pak součin. Můžeme použít i méně obvyklou **postfixovou notaci** (označovanou též **RPN = reverzní polská notace**). U té se znaménko operace zapisuje **za operandy**. Pořadí operací je dáno jednoznačně pořadím jejich zápisu, zabývat se prioritou není třeba a závorky nepotřebujeme. Pro první funkci bychom použili zápis $a b \neg + \cdot$ a pro druhou $a b \neg + c \cdot$. Výpočet podle této notace se velmi snadno technicky realizuje pomocí **zásobníkové paměti** – což je paměť, do které zapisujeme hodnoty v opačném pořadí je čteme (poslední zapsaná hodnota je první přečtena). Zápis funkce totiž pak přímo představuje postup jejího výpočtu. Proměnná znamená načtení hodnoty do zásobníkové paměti a znaménko operace provedení operace s načtenými hodnotami a zapsání výsledku zpět do zásobníku. Nakonec ze zásobníku můžeme přečíst výsledek a uložit ji do výsledné proměnné. Například funkci $z = (a + \bar{b}) \cdot c$ bychom po přepsání na $a b \neg + c \cdot$ vypočetli postupem:

- a ... načti obsah proměnné a*
- b ... načti obsah proměnné b*
- ¬ ... neguj (poslední načtenou hodnotu – tedy b)*
- + ... sečti (poslední dvě hodnoty uložené v zásobníku, tedy a s negovaným b)*
- c ... načti obsah proměnné c*
- ... násob (poslední dvě hodnoty uložené v zásobníku, tedy výsledek součtu s c)*
- =z ... výsledek zapiš do proměnné z*

A obdobným postupem se provádějí i výpočty v programovatelném automatu.

Jazyk IL

Pro zápis programu automatu existuje řada jazyků. Jsou textové (píše se text) i grafické (kreslí se schéma). V Evropě se nejvíce rozšířil textový jazyk nazývaný *jazyk symbolických instrukcí*, který se ovšem liší u různých výrobců. O sjednocení se pokusila norma IEC 61131-3 (v ČR přijata jako ČSN EN 61 131-3), která tento druh jazyka nazývá **IL (Instruction List = Seznam Instrukcí)**. Program je zapsán jako seznam instrukcí postupně vykonávaných automatem. Soubor všech možných instrukcí konkrétního automatu se pak nazývá **instrukční sada**. Instrukce je zapsána svým názvem a případně i názvem použitého operandu. Každá instrukce je na samostatném řádku. Tak například načtení hodnoty do zásobníku je označeno názvem „LD“. Načtení proměnné se symbolickým názvem „**TLACITKO**“ pak zapíšeme

LD TLACITKO

Firemní jazyky se snaží o slučitelnost s normou, ale stále zde jsou určité odlišnosti. Zejména se jedná o rozsah instrukční sady a rozdílné názvy instrukcí. Pokud chceme provádět vyhodnocení logické funkce, potřebujeme kromě čtení vstupních proměnných také vykonávat algebraické operace a zapisovat výsledné hodnoty. Názvy instrukcí pro námi používané automaty jsou

Simatic STEP7	Tecomat	Význam
LD JMENO	LD JMENO	Načtení hodnoty z proměnné s názvem „JMENO“
NOT	NEG	Negace
AND	AND	Součin
OR	OR	Součet
= JMENO	WR JMENO	Zápis hodnoty do proměnné s názvem „JMENO“

Těmito instrukcemi již jsme schopni provést výpočet libovolné funkce. Dříve uvedený postup výpočtu funkce $z=(a+b) \cdot c$ (pokud pro proměnné a, b, c a z použijeme názvy **START**, **STOP**, **VOLNO** a **MOTOR**) lze zapsat v jazce IL fragmentem kódu (ve variantě pro Simatic = STEP7):

```
LD START
LD STOP
NOT
OLD
LD VOLNO
ALD
= MOTOR
```

nebo ve variantě pro automaty Tecomat:

```
LD START
LD STOP
NEG
OR
LD VOLNO
AND
WR MOTOR
```

Existují i efektivnější instrukce. Například načtení hodnoty a její negaci můžeme provést jako jedinou instrukci. Stejně tak součet či součin s načtenou hodnotou a dokonce i s načtenou a negovanou hodnotou můžeme provést jedinou instrukcí. Odpovídající instrukce jsou

Simatic STEP7	Tecomat	Význam
LDN JMENO	LDC JMENO	Načtení negované hodnoty proměnné s názvem „JMENO“
A JMENO	AND JMENO	Součin s hodnotou proměnné „JMENO“
O JMENO	OR JMENO	Součet s hodnotou proměnné „JMENO“
AN JMENO	ANC JMENO	Součin s negovanou hodnotou proměnné „JMENO“
ON JMENO	ORC JMENO	Součet s negovanou hodnotou proměnné „JMENO“

Výše uvedený program pak lze úsporněji zapsat

```
LD START
ON STOP
A VOLNO
= MOTOR
```

respektive

```
LD START
ORC STOP
AND VOLNO
WR MOTOR
```

Další užitečné instrukce realizují funkci RS klopného obvodu. Provádějí nastavení (**SET**) a nulování (**RESET**) hodnoty proměnné. To se ale provede pouze za podmínky, pokud na vrchlou zásobníku je nenulová hodnota.

Simatic STEP7	Tecomat	Význam
S JMENO, 1	SET JMENO	Podmíněné nastavení proměnné
R JMENO, 1	RES JMENO	Podmíněné nulování proměnné

Instrukční sada běžných automatů je samozřejmě mnohem rozsáhlejší, uvedené instrukce představují jen její nepatrnou část. Pro řešení jednoduchých logických funkcí ale postačí. Aby uvedené fragmenty kódu vytvořily program proveditelný automatem, musí se ještě doplnit dalšími organizačními instrukcemi – které například určí, kdy se má program provádět, v jakém pořadí apod. V našem případě už tyto další instrukce budou v projektu připraveny a postačí pouze doplnit kód pro výpočet logických funkcí složený z uvedených instrukcí.

V Praze dne 8.3.2010 5:10

Filip ZÁMEK