

Programovanie PLC a norma IEC 61131-3

Norma IEC 61131-3 je výsledkom štandardizácie programovacích jazykov pre priemyselnou automatizáciu. Norma má celosvetovú podporu a preto nie je závislá od jednej spoločnosti.

IEC 61131-3 je treťou časťou normy IEC 61131. Skladá sa piatich častí:

1. Všeobecný prehľad
2. Hardware
3. Programovacie jazyky
4. Pokyny pre užívateľov
5. Komunikácia

Norma sa dá rozdeliť na dve hlavné skupiny:

1. Spoločné prvky
2. Programovacie jazyky

Spoločné prvky

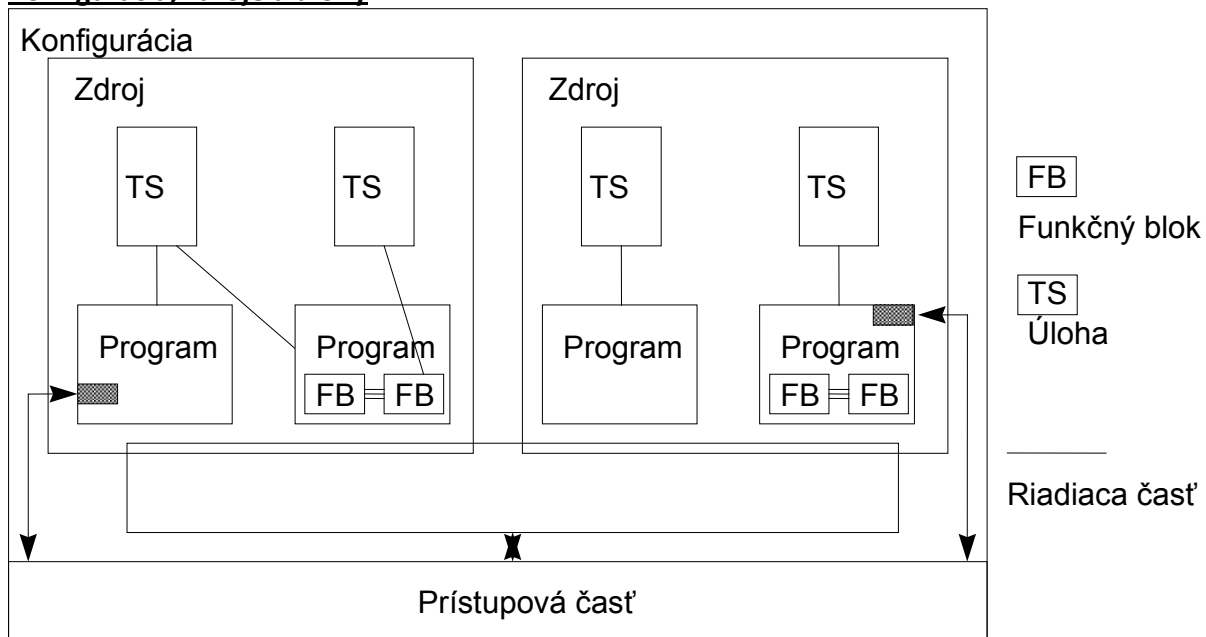
Dátové typy

Dátové prvky sú definované v rámci spoločných prvkov. Určenie správnych dátových prvkov zabraňuje chybám už v rannom štádiu. Používa sa na definovanie všetkých použitých parametrov. Zabráni sa tým delenie dátumu celým číslom. Dátové prvky sú definované už v rámci spoločných prvkov. Výber správnych dátových typov zabraňuje vývojovým chybám už v rannom štádiu. Definovanie všetkých použitých parametrov zabráni napríklad deleniu dátumu celým číslom. Spoločné dátové typy sú Boolean, Integer, Real a Byte a Word, ale aj dátum, TIME_OF_DAY a String. Na ich základe je možné nadefinovať vlastné, odvodené dátové typy. Týmto spôsobom je možné definovať aj analógový kanál ako dátový typ.

Premenné

Premenné sú explicitne priradené hardvérovým adresám (napr. vstupy a výstupy) zdrojov alebo programom. Týmto spôsobom vzniká vysoká miera nezávislosti hardvéru s možnosťou opätovného použitia softvéru. Platnosť definovaných premenných sa zvyčajne obmedzuje na organizačnú jednotku, v ktorej sú deklarované (lokálne premenné). Znamená to, že sa dajú použiť v ďalších častiach bez akéhokoľvek konfliktu. Ak majú mať globálny dosah, musia byť definované globálne (VAR_GLOBAL). Premenným sa počítačové hodnoty priradia pri štarte alebo studenom reštarte.

Konfigurácia, zdroje a úlohy



Obr. 1 Softvérový model

Softvér potrebný na vyriešenie konkrétneho riadiaceho problému je na najvyššej úrovni definovaný ako Konfigurácia. Konfigurácia je špecifická pre určitý typ riadiaceho systému – vrátane zabezpečenia hardvéru, spracovania zdrojov, pamäťových adries pre I/O kanály a systémových možností. V rámci nastavení je možné zdefinovať jeden alebo viac Zdrojov. Na zdroje je možné nazerať ako na technologické zariadenie, ktoré je schopné vykonávať programy IEC. V zdrojoch môžu byť nedefinované úlohy. Úlohy riadia vykonávanie želaných programov alebo funkčných blokov. Úlohy sa vykonávajú buď pravidelne alebo na základe spúšťača alebo po zmene premennej. Programy sú zostavené zo softvérových prvkov a sú napísané v IEC jazyku. Program sa typicky skladá z množiny funkcií a funkčných blokov, ktoré sú schopné výmeny dát. Funkcie a funkčné bloky sú základné stavebné kamene obsahujúce dátové štruktúry a algoritmus. V porovnaní s klasickým PLC, ktoré obsahuje jeden zdroj, jednu bežiacu úlohu a jeden riadiaci program bežiaci v uzavretej slučke, norma IEC 61131-3 pridáva ďalšie možnosti – multiprocessing a udalosťami riadené programy. Norma je určená pre širokú škálu aplikácií, aj bez znalosti ďalších programovacích jazykov.

Programové organizačné jednotky

V norme IEC 61131-3 sú programy, funkčné bloky a funkcie nazývané Programové organizačné jednotky - POU.

Funkcie

Norma IEC rozdeľuje funkcie na štandardné a užívateľom definované. Štandardné funkcie sú napríklad ADD, ABS, SQRT, SIN a COS. Užívateľom definované funkcie sa nedefinujú raz a môžu používať stále.

Funkčné bloky, FB

Funkčné bloky sú ekvivalentom integrovaných obvodov a predstavujú špecializované riadiace funkcie. Funkčné bloky obsahujú aj údaje aj algoritmus, čiže môžu sledovať aj minulosť (na rozdiel od funkcií). Rozhranie majú definované ako integrovaný obvod alebo čierna skrinka, čím jasne rozdeľujú úrovne programátorov alebo personál

údržby. Vynikajúcim príkladom funkčného bloku je regulácia teploty alebo PID. Funkčné bloky je možné použiť v tom istom programe, v inom programe, no aj v iných projektoch opakovane. Programovanie funkčných blokov je možné v IEC jazykoch, no často sa používa jazyk C. Odvođené funkčné bloky sú založené na štandardných blokoch, ale nie je problém vytvoriť nové FB podľa poskytnutého frameworku.

Funkcie a funkčné bloky sú popísané rovnakým spôsobom:

Príklad FUNCTION_BLOCK

VAR_INPUT:

X: Bool;

Y: Bool;

END_VAR

VAR_OUTPUT

Z: Bool;

END_VAR

(* Výkazy * functionblock tela)

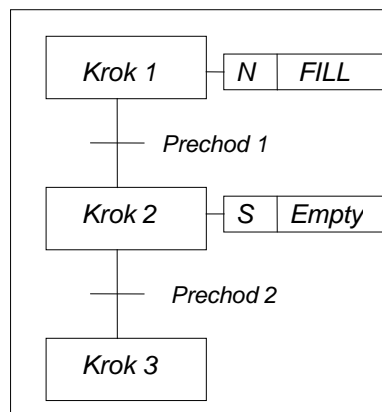
END_FUNCTION_BLOCK

Deklarácia popisuje rozhranie funkčného bloku s dvoma logickými vstupnými parametrami a jedným výstupným logickým parametrom.

Programy

Základným stavebným kameňom je program s množinou funkcií a funkčných blokov.

Sekvenčný funkčný diagram



Obr.2 Sekvenčný funkčný diagram

Sekvenčný funkčný diagram popisuje sekvenčné správanie riadiaceho programu a je odvodený z Petriho siete a IEC 848. SFC rozdeľuje vnútornú organizáciu programu a pomáha rozdeliť riadenie do ovládateľných častí. SFC sa skladá z krokov prepojených s akčnými blokmi a prechodmi. Každý krok predstavuje konkrétny stav riadeného systému. Kroky prepojené s akčnými blokmi vykonávajú riadiace akcie. Prvky môžu byť naprogramované v IEC jazyku, vrátane SFC. Je možné používať alternatívne sekvencie či paralelné sekvencie, ktoré sa bežne vyskytujú u dávkových aplikácií. Jedna sekvencia môže byť použitá napríklad pre hlavný proces, druhá zase na sledovanie celkových prevádzkových vlastností.

Programovacie jazyky

Štandard definuje štyri programovacie jazyky, čiže syntax a sémantika je už pripravené. Jazyky sa dajú rozdeliť na textové a grafické verzie:

Textové:

- Zoznam príkazov, IL
- Štruktúrovaný text, ST

Grafické:

- Ladder diagram, LD
- Funkčná bloková schéma, FBD

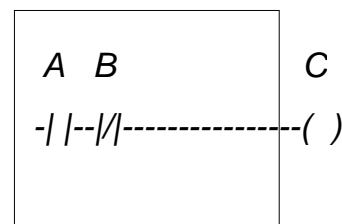
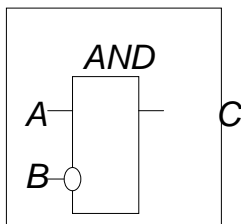
Zoznam príkazov (IL)

```
LD  A
ANDNB
ST  C
```

Štruktúrovaný text (ST)

```
C:= A AND NOT B
```

Funkčná bloková schéma (FBD) Ladder Diagram (LD)



Obr.3 Všetky programovacie jazyky popisujú rovnaký program

Voľba programovacieho jazyka je závislá na:

- znalostiach programátora
- riešenom probléme
- úrovni popisu problému
- štruktúre riadiaceho systému
- prepojení s ostatnými ľuďmi / oddeleniami

Všetky štyri jazyky sú vzájomne prepojené, pretože poskytujú spoločný súbor programov.

Ladder Diagram, má korene v USA. Je založený na grafickej prezentácii Relay Ladder Logic. **Zoznam príkazov** je jeho európsky náprotivok. Jeho textový jazyk pripomína assembler. **Funkčná bloková schéma** sa veľmi často používa vo výrobnom priemysle. Vyjadruje správanie funkcií, funkčných blokov a programov ako súbor vzájomne prepojených grafických blokov, podobne ako elektrotechnických schém. Na systém sa pozerá z hľadiska toku signálu medzi prvkami.

Štruktúrovaný text je veľmi silný nástroj na úrovni jazykov Ada, Pascal a C. Obsahuje všetky podstatné prvky moderného programovacieho jazyka, vrátane vetvenia (IF-THEN-ELSE a CASE OF) a iteračných slučiek (FOR, WHILE a REPEAT). Tieto prvky môžu byť vnorené. Štruktúrovaný text môže byť použitý na definovanie komplexných funkčných blokov, ktoré sa dajú použiť v ďalších jazykoch.

Ukážka:

```
I: = 25;
WHILE J <5 sa
    Z: = f (I + J);
END_WHILE

IF B_1 THEN
    % QW100: = INT_TO_BCD (Display)
ENDIF

CASE TW
    1,5: TEMP: = TEMP_1;
    2: TEMP: = 40;
    4: TEMP: = FTMP (TEMP_2);
ELSE
    TEMP: = 0;
    B_ERROR: = 1;
END_CASE
```

Zhora nadol vs zdola nahor

Štandard umožňuje rozvíjať program dvoma spôsobmi: zhora nadol a zdola nahor. Buď špecifikujete celú aplikáciu a rozdelíte ju na jednotlivé časti, deklarujete premenné a podobne. Alebo môžete začať programovať aplikáciu "zdola", prostredníctvom odvodených funkcií a funkčných blokov.

Implementácia

Nie je ľahké splniť všeobecné požiadavky na normu IEC 61131-3. Z tohto dôvodu norma umožňuje čiastočnú implementáciu v rozličných oblastiach. To sa týka počtu podporovaných jazykov, funkcií a funkčných blokov.

www.openplc.org