Programovacie jazyky PLC: Instruction List (IL), Structured Text (ST), Ladder Diagrams (LD), Function Block Diagram (FBD), Sequential Function Chart (SFC) :)

V rokoch 1993÷2003 vydal International Electrotechnical Commission (IEC) štandard známy ako 61131. V danom štandarde nájdeme sekciu IEC 61131-3 ktorá obsahuje dokumenty venované programovacím jazykom PLC. Hlavným dôvodom pre vytvorenie tohto štandardu bolo práve zjednotenie – štandardizácia existujúcich programovacích jazykov rôznych výrobcov PLC automatov.

Povedzme si úprimne o koľko komplikovanejší by bol život programátora, keby sa pri rôznych výrobcoch PLC musel učiť nový spôsob programovania. Či už teda programujete PLC od Mitsubishi, Siemens, Beckhoff či iného výrobcu, vždy sa stretnete s niektorými z programovacích jazykov popísaných v danom štandarde.

Štandard IEC 61131-3 hovorí práve o piatich nasledovných spôsoboch programovania PLC.

Instruction List (IL) – postupnosť inštrukcii (nemecky AWL)

Ide o programovací jazyk nižšej úrovne, ktorý je veľmi podobný assembleru. Takýto jazyk je predovšetkým vhodný skôr iba pre malé projekty, alebo pre tvorbu uzavretých funkcií. V dnešnej dobe je však pre svoju zložitosť využívaný iba veľmi málo pretože v súčasnosti nájdeme iné jednoduchšie, a tiež prehľadnejšie spôsoby programovania.

```
      U(
      U(
      UN    "First_Scan"              M50.7
      SPBNB _00c
      L     100
      T     "MaxEngINT".Umid          DB5.DBW4
      SET
      SAVE
      CLR
_00c: U     BIE
      )
      SPBNB _00d
      L     "MaxEngINT".Umid          DB5.DBW4
      ITD
      T     "MaxEngDINT".Umid         DB6.DBD8
      SET
      SAVE
      CLR
_00d: U     BIE
      )
      SPBNB _00e
      L     "MaxEngDINT".Umid         DB6.DBD8
      DTR
      T     "MaxEngREAL".Umid         DB7.DBD8
_00e: NOP   0
```

Structured Text (ST) – štruktúrovaný text

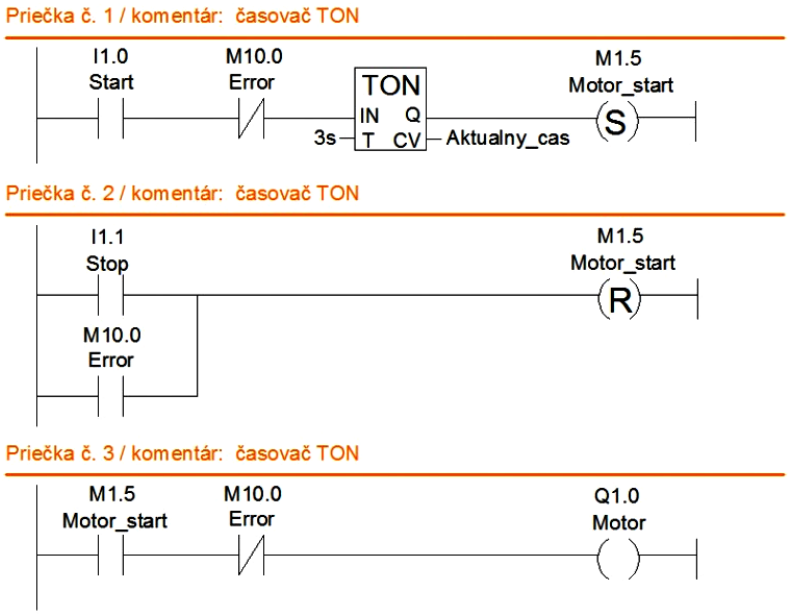
Pri ST ide o programovací jazyk vyššej úrovne. Môžeme ho prirovnať programovacím jazykom „C“, Pascal a iné. Svojich priaznivcov si „ST“ preto nachádza predovšetkým medzi programátormi, ktorí sú zvyknutý pracovať práve s programovacími jazykmi vyššej úrovne.

```
        (* TEST CYCLE SETUP  *)
        cycle_In1 := tb_In1[testCycleNum];
        cycle_In2 := tb_In2[testCycleNum];
        cycle_In3 := tb_In3[testCycleNum];
        cycle_Out := tb_Out[testCycleNum];
        IF testCycleNum = 0 THEN
            (* INIT  *)
            PID_Subsystem(i0_PID_Subsystem, 0, cycle_In1, cycle_In2, cycle_In3, out_Out);
        END_IF;
        (* STEP  *)
        PID_Subsystem(i0_PID_Subsystem, 1, cycle_In1, cycle_In2, cycle_In3, out_Out);
        (* VERIFY  *)
        IF testVerify THEN
            IF cycle_Out = 0.0 THEN
                IF ABS(out_Out) > 9.9999997473787516E-5 THEN
                    testVerify := 0;
                END_IF;
            ELSIF ABS(out_Out - cycle_Out) > (9.9999997473787516E-5 * ABS(cycle_Out)) THEN
                testVerify := 0;
            END_IF;
        END_IF;
        testCycleNum := testCycleNum + 1;
    END_IF;
END_IF;
```

Copyright © 2013 - 2026 Encyklopédia poznania

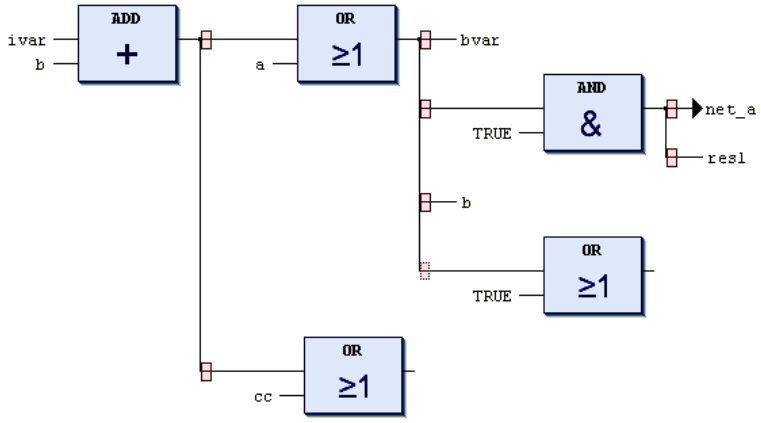
Ladder Diagrams (LD) – jazyk kontaktných (reléových) schém (nemecky KOP)

Daný programovací jazyk môžeme prirovnať ku tvorbe elektrických reléových schém. Na rozdiel od predošlých programovacích jazykoch (IL, ST) je LD grafický spôsob programovania. V dnešnej dobe je veľmi obľúbeným a využívaným jazykom z dôvodu svojej jednoduchosti tvorby vlastného programu, prehľadnosti zápisu, čítania kódu a jednoduchej diagnostiky v prípade poruchy zariadenia.



Function Block Diagram (FBD) – jazyk funkčných blokov (nemecky FUP)

Ide opäť o grafický spôsob programovania, kde využívame na programovanie funkčné bloky, na ktoré pripájame vstupy a výstupy. Každá z funkcii (časovače, matematické operácie a iné) má svoje označenie a svoj vlastný blok, ktorý následne spájame dokopy s ostatnými blokmi s cieľom vytvorenia vlastného programu podľa vopred nadefinovaného algoritmu.



Sequential Function Chart (SFC)

Tak ako predošlé dva programovacie jazyky tak aj tento môžeme považovať sa grafický programovací jazyk. V súčasnej dobe je veľmi obľúbený, a to predovšetkým pre vytváranie sekvenčných programov. V preklade, ak potrebujeme naprogramovať sled po sebe nasledujúcich udalostí je vhodné využiť práve daný typ programovania – SFC. Ako príklad si môžeme uviesť jednoduchú postupnosť krokov potrebných pre naprogramovanie:

1. kontrola inicializácie pracoviskám,

2. vlož diel,

3. skontroluj prítomnosť dielu,

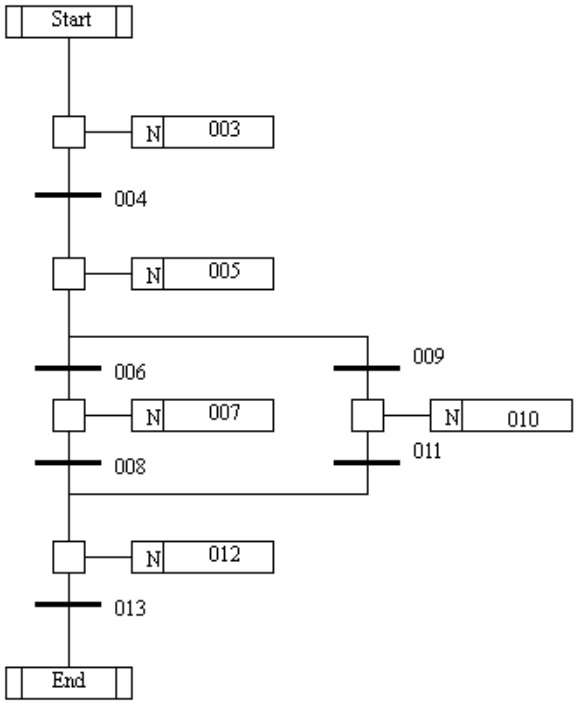
4. uzavri úpinky,

Copyright © 2013 - 2026 Encyklopédia poznania

5. opracuj diel,

6. odopni úpinky,

7. zinicializuj pracovisko.



Zdroje

Prevzaté a upravené z:

• **https://www.dailyautomation.sk/01-programovacie-jazyky-plc/.**

Copyright © 2013 - 2026 Encyklopédia poznania