

RsLogix5000 - Bitové inštrukcie

Körösi Ladislav · Elektrotechnika, Informačné technológie

04.08.2014

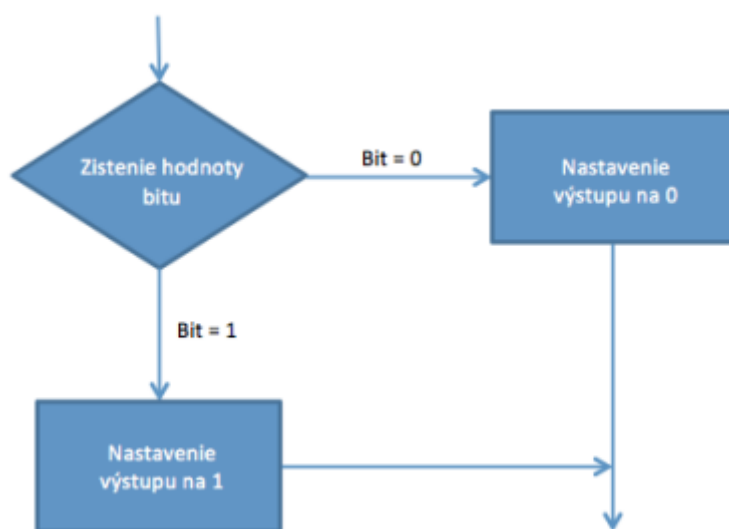


Cieľom článku je oboznámiť čitateľa so základnými bitovými inštrukciami softvéru RsLogix 5000 v jazyku rebríkových schém (LD - Ladder diagram). Softvér RsLogix 5000 je konfiguračný a programovací nástroj programovateľných logických automatov (PLC - Programmable logic controller) spoločnosti Rockwell Automation pre rady ControlLogix, CompactLogix, FlexLogix, SoftLogix5800 a DriveLogix.

Inštrukcia „Examine If Closed“ (XIC)

Inštrukcia XIC zisťuje či je hodnota bitu, ktorý je zadáný ako parameter inštrukcie rovná logickej jednotke (TRUE). Slúži na povolenie výstupu („vykonanie inštrukcie“), ak má bit stav TRUE. Táto inštrukcia je dostupná iba v programovacom jazyku LD a nie je dostupná v ďalších jazykoch (FBD, ST). Výstup inštrukcie je závislý od hodnoty bitu, stavu rebríka ako aj krokov vykonania:

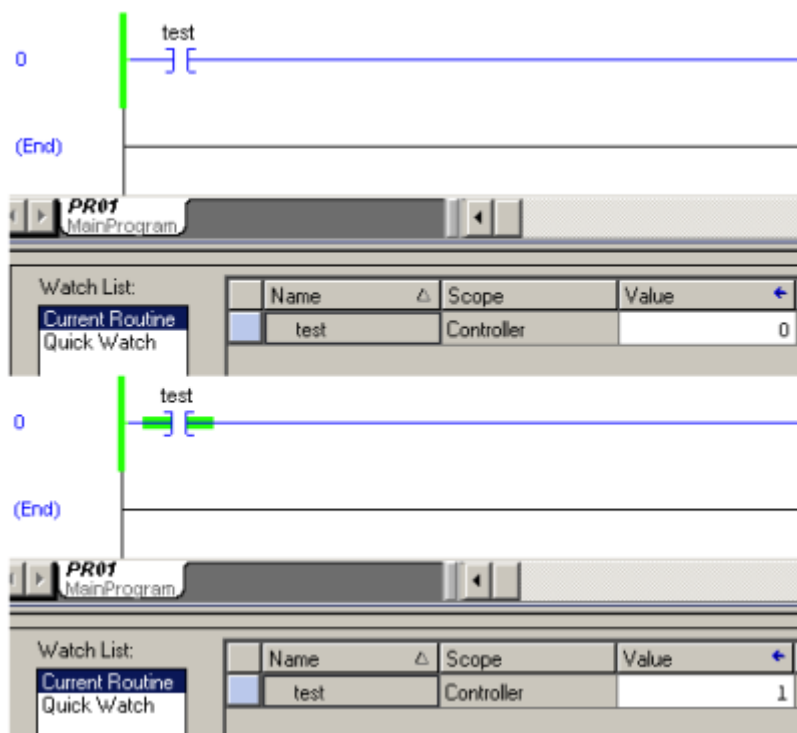
- V stave PRESCAN je výstupom inštrukcie logická nula (FALSE).
- Ak je stav rebríka FALSE, potom výstupom inštrukcie je FALSE.
- Ak je stav rebríka TRUE, potom výstupom inštrukcie je TRUE alebo FALSE podľa vývojového diagramu (obr. 1).
- V stave POSTSCAN je hodnota inštrukcie FALSE.



Obr. 1 Vývojový diagram inštrukcie XIC

Zjednodušene povedané, výstupom inštrukcie XIC je hodnota jej parametra (bitu). Ukážka vyhodnotenia premennej (tagu) test je na obr. 2. Na obr. 2 vpravo je zelenou farbou animovaná logická jednotka, ktorá je výstupom inštrukcie. Ekvivalentom v textových jazykoch je:

```
IF test THEN
ELSE
END_IF;
```

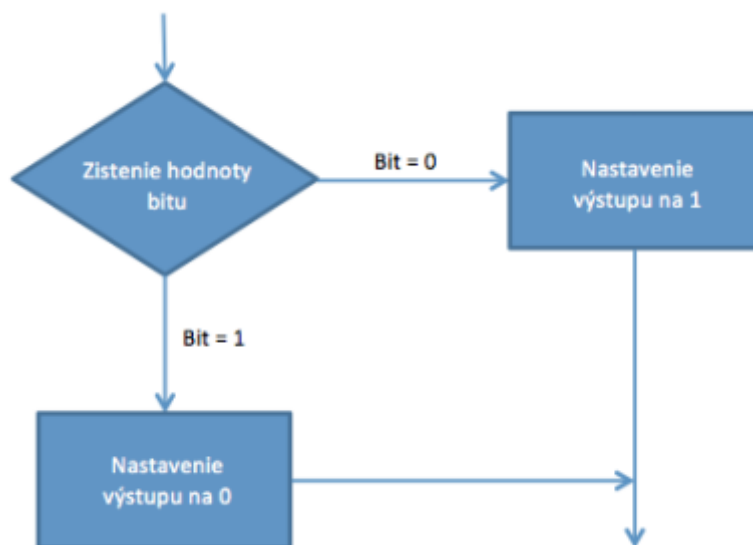


Obr. 2 Ukážka inštrukcie XIC

Inštrukcia „Examine If Open“ (XIO)

Inštrukcia XIO zisťuje či sa hodnota parametra inštrukcie (hodnota bitu) rovná logickej nule (FALSE). Slúži na povolenie výstupu ak má bit hodnotu FALSE. Táto inštrukcia podobne inštrukcia XIC nie je dostupná v ďalších programovacích jazykoch. Výstup inštrukcie je závislý od hodnoty bitu ako aj krokov vykonania:

- V stave PRESCAN je výstupom inštrukcie logická nula (FALSE).
- Ak je stav rebríka FALSE, potom výstupom inštrukcie je FALSE.
- Ak je stav rebríka TRUE, potom výstupom inštrukcie je TRUE alebo FALSE podľa vývojového diagramu (Obr. 3).
- V stave POSTSCAN je hodnota inštrukcie FALSE.

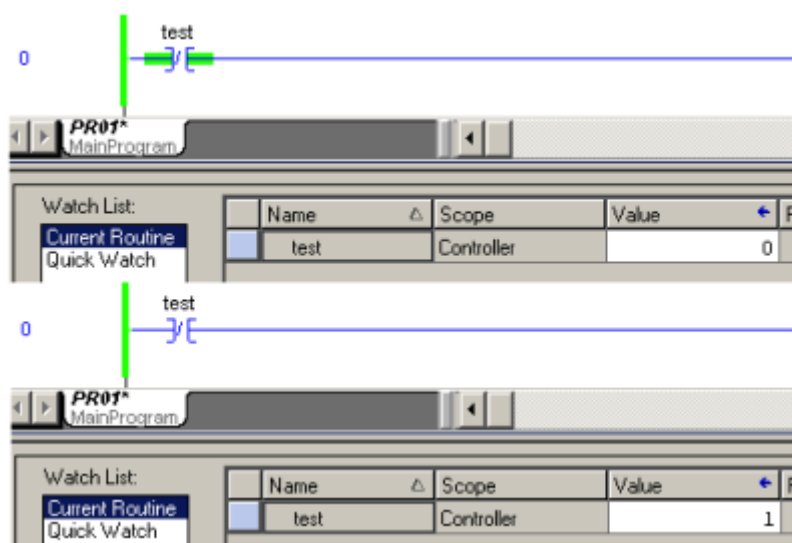


Obr. 3 Vývojový diagram inštrukcie XIO

Výstupom inštrukcie XIO je negovaná hodnota jej parametra (bitu). Na obr. 4 vľavo vidno, že stav bitu je 0 (FALSE) a výstupom inštrukcie je 1 (TRUE), čo je animované zelenou farbou. Ekvivalentom v textových jazykoch je:

```

IF not(test) THEN
ELSE
END_IF;
  
```



Obr. 4 Ukážka inštrukcie XIO

Inštrukcia „Output Energize“ (OTE)

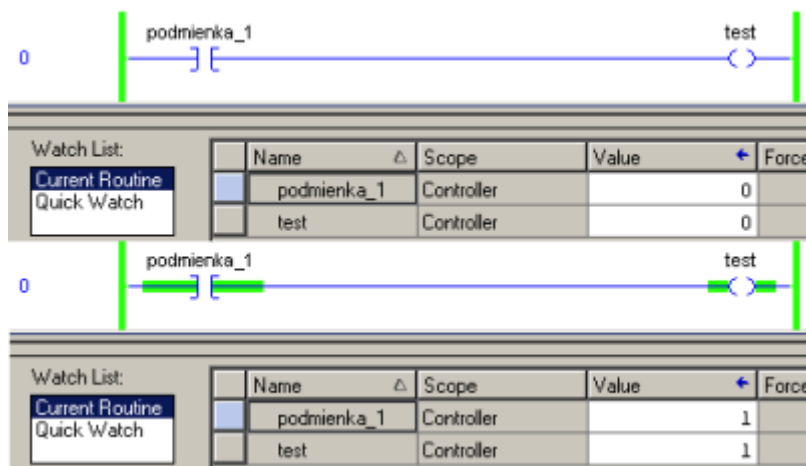
Inštrukcia OTE nastavuje hodnotu bitu (parametra) v závislosti od vstupného signálu (podmienky) inštrukcie. Ak je vstupná podmienka TRUE, t.j. inštrukcia je povolená, potom sa hodnota bitu nastaví na TRUE. Ak je vstupná podmienka FALSE, t.j. inštrukcia nie je povolená, potom sa hodnota bitu nastaví na FALSE. Inštrukcia je dostupná iba v jazyku LD. Hodnota bitu závisí od nasledujúcich stavov:

- V stave PRESCAN je hodnota bitu FALSE.
- Ak je stav rebríka FALSE, potom hodnota bitu je FALSE.

- Ak je stav rebríka TRUE, potom hodnota bitu je TRUE.
- V stave POSTSCAN je hodnota bitu FALSE.

Na obr. 5 je ukážka nastavenia hodnoty bitu test v závislosti od bitu podmienka_1. V prvom príklade je na vstupe inštrukcie OTE hodnota FALSE lebo inštrukcia XIC s parametrom podmienka_1 dáva na výstup inštrukcie hodnotu FALSE. V druhom príklade je na vstupe inštrukcie OTE hodnota TRUE lebo inštrukcia XIC s parametrom podmienka_1 dáva na výstup inštrukcie hodnotu TRUE. V textových programovacích jazykoch je ekvivalentnou inštrukciou:

```
test := podmienka_1;
```



Obr. 5 Ukážka inštrukcie OTE

Inštrukcia „Output Latch” (OTL)

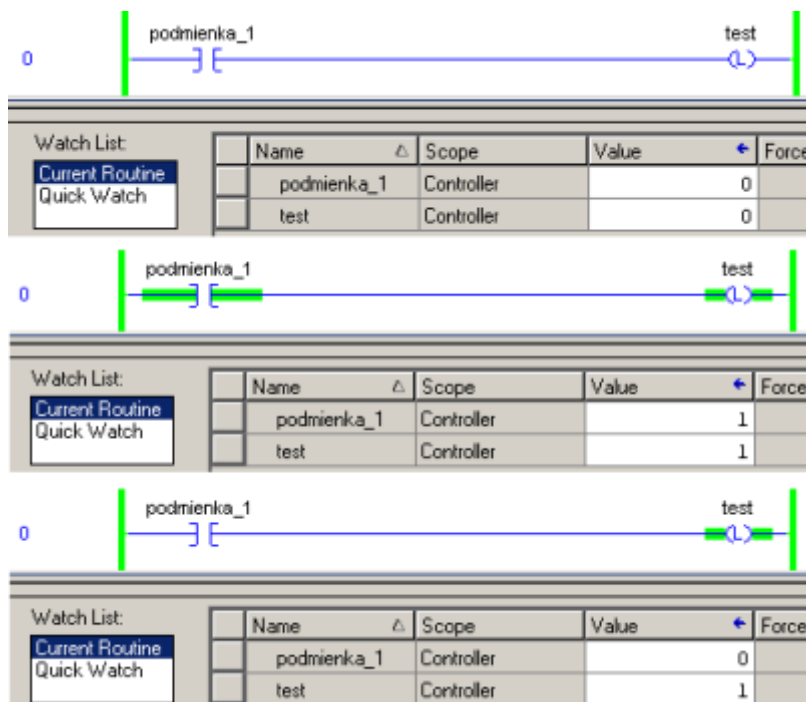
Inštrukcia pri platnej vstupnej podmienke nastavuje hodnotu bitu (parametra) na hodnotu TRUE. Inštrukcia nie je dostupná v ďalších programovacích jazykoch. Stavy, ktoré môže nadobúdať:

- V stave PRESCAN sa hodnota bitu nemení.
- Ak je stav rebríka FALSE, potom sa hodnota bitu nemení.
- Ak je stav rebríka TRUE, potom sa hodnota bitu nastaví na TRUE.
- V stave POSTSCAN sa hodnota bitu nemení.

Na obr. 6 je príklad inštrukcie OTL pre zmenu vstupnej podmienky podmienky_1 z FALSE na TRUE a následne opäť na FALSE. Z posledného príkladu vidno, že po zániku podmienky sa hodnota tagu test nemodifikuje. Ekvivalentom inštrukcie v textovom programovacom jazyku je:

```
IF podmienka_1 THEN
    test := TRUE;
END_IF;
```

Nulovanie (resetovanie) hodnoty bitu je realizované typicky inštrukciou OTU (Output unlatch). Samozrejme aj iné inštrukcie môžu daný stav resetovať. Medzi ne patrí už opísaná inštrukcia OTE, ktorá prepíše stav bitu na základe vstupnej podmienky.



Obr. 6 Ukážka inštrukcie OTL

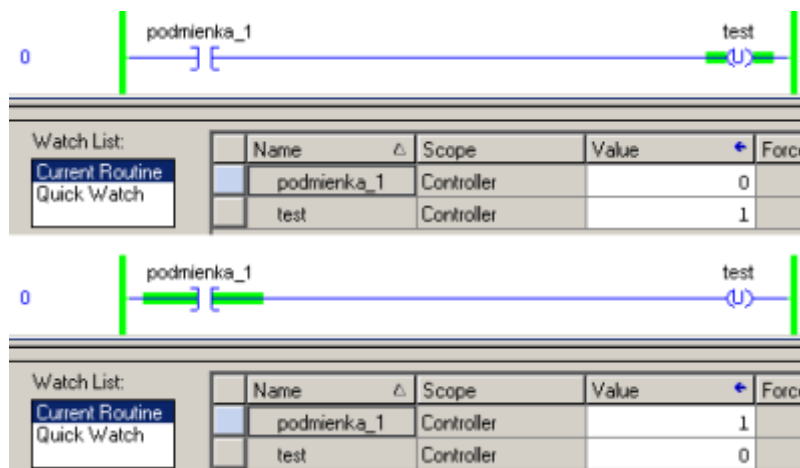
Inštrukcia „Output Unlatch” (OTU)

Inštrukcia pri platnej vstupnej podmienke nastavuje (resetuje, nuluje) hodnotu bitu na hodnotu FALSE. Inštrukcia ako aj OTL nie je dostupná v ďalších programovacích jazykoch. Stav, ktoré môže nadobúdať:

- V stave PRESCAN sa hodnota bitu nemení.
- Ak je stav rebríka FALSE, potom sa hodnota bitu nemení.
- Ak je stav rebríka TRUE, potom sa hodnota bitu nastaví na FALSE.
- V stave POSTSCAN sa hodnota bitu nemení.

Na obr. 7 je príklad inštrukcie OTU pre zmenu vstupnej podmienky podmienky_1 z FALSE na TRUE. V prvom prípade tag test nie je modifikovaný lebo nie je platná vstupná podmienka inštrukcie OTU. V druhom prípade pri platnej podmienke sa hodnota bitu vynulovala. Ekvivalentom inštrukcie OTU v textovom programovacom jazyku je:

```
IF podmienka_1 THEN
    test := FALSE
END_IF;
```



Obr. 7 Ukážka inštrukcie OTU

Inštrukcia „One Shot“ (ONS)

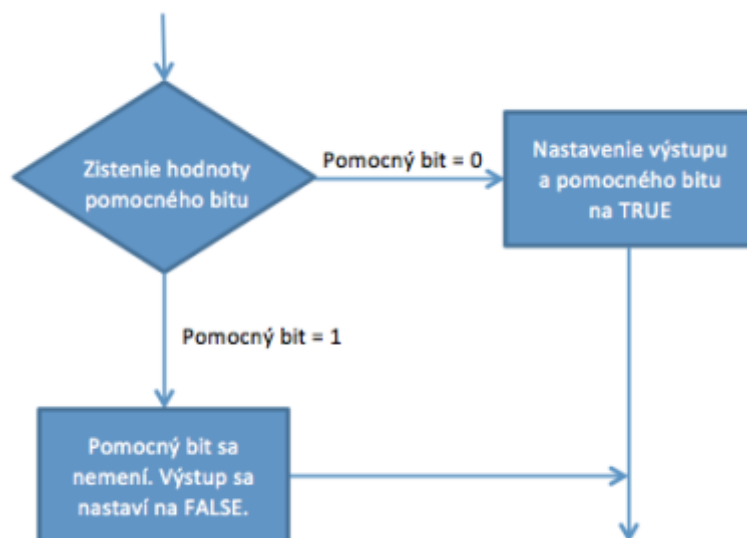
Inštrukcia ONS slúži na detekciu nábežnej hrany vstupného signálu, t.j. pri zmene vstupnej podmienky z FALSE na TRUE dáva na výstup hodnotu TRUE. V opačnom prípade je výstupom inštrukcie FALSE. Pomocný bit (tzv. storage bit), ktorý je parametrom inštrukcie, slúži na zapamätanie aktuálneho stavu vstupu pre ďalší cyklus PLC, preto zmena nastane ak aktuálny vstup do inštrukcie je TRUE a pomocný bit má hodnotu FALSE. Je potrebné upozorniť, že do pamäťovej premennej použitej na uloženie pomocného bitu nie je vhodné zapisovať z iného miesta programu, inak inštrukcia ONS nebude poskytovať adekvátny výsledok. Výstup inštrukcie ovplyvňuje tok signálu v aktuálnom rebríku, preto sa používa na podmienené vykonávanie ďalších inštrukcií. Postupnosť stavov:

- V stave PRESCAN sa hodnota pomocného bitu nastaví na TRUE a výstup inštrukcie na FALSE. Tento stav zabezpečí, aby došlo k chybnej detekcii.
- Ak je stav rebríka FALSE, potom sa hodnota pomocného bitu a výstup inštrukcie nastaví na FALSE.
- Ak je stav rebríka TRUE, potom sa vykoná inštrukcia podľa vývojového diagramu (obr. 8).
- V stave POSTSCAN sa hodnota pomocného bitu a výstup inštrukcie nastaví na FALSE.

Ekvivalentným príkazom v textovom jazyku je

```
OUT := TLACIDLO_1 AND (NOT(SB_1));
SB := TLACIDLO_1;
```

kde TLACIDLO_1 je vstupom do inštrukcie, SB_1 je pomocný bit a OUT je výstupom inštrukcie.

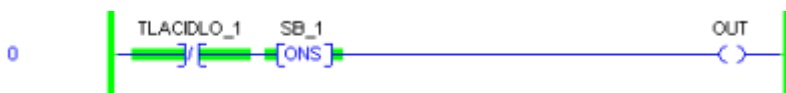


Obr. 8 Vývojový diagram inštrukcie ONS

Príklad použitia inštrukcie ONS v jazyku LD je na obr. 9. Príklad je z pohľadu funkcionality totožný vyššie uvedenému v textovom programovacom jazyku. Bit OUT bude mať stav TRUE na jeden cyklus len a len vtedy, ak TLACIDLO_1 = TRUE a SB_1 = FALSE (t.j. TLACIDLO_1 z predchádzajúceho cyklu mal stav FALSE). Pomocou inštrukcie ONS je možné detekovať aj dobežnú hranu, t.j. zmenu vstupného signálu z TRUE na FALSE a to tak, že sa na vstupe ONS použije negovaný signál (obr. 10).



Obr. 9 Ukážka inštrukcie ONS

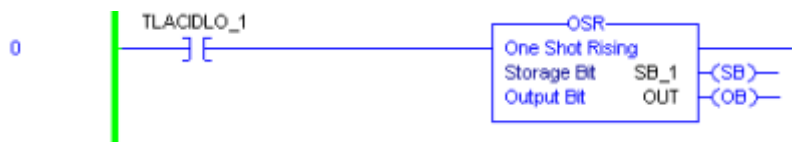


Obr. 10 Ukážka inštrukcie ONS pre detekciu dobežnej hrany

Inštrukcia „One Shot Rising“ (OSR)

Inštrukcia OSR podobne ako inštrukcia ONS slúži na detekciu nábežnej hrany. Na rozdiel od ONS výstup inštrukcie zapisuje do parametra (tzv. Output bit). Premennú použitú na uloženie pomocného bitu opäť nie je vhodné používať na inom mieste programu. Na obr. 9 a 11 sú ekvivalentné ukážkové programy. Postupnosť stavov:

- V stave PRESCAN sa hodnota pomocného bitu (storage bit) nastaví na TRUE a výstupný bit (output bit) na FALSE. Zároveň sa výstup inštrukcie, ktorý ovplyvňuje tok signálu nastaví na FALSE.
- Ak je stav rebríka FALSE, potom sa hodnota pomocného bitu a výstup inštrukcie nastavia na FALSE. Výstupný bit sa nemení.
- Ak je stav rebríka TRUE, potom sa vykoná inštrukcia podľa vývojového diagramu ako pri inštrukcii ONS (obr. 8).
- V stave POSTSCAN sa hodnota pomocného bitu a výstup inštrukcie nastavia na FALSE. Výstupný bit sa nemení.

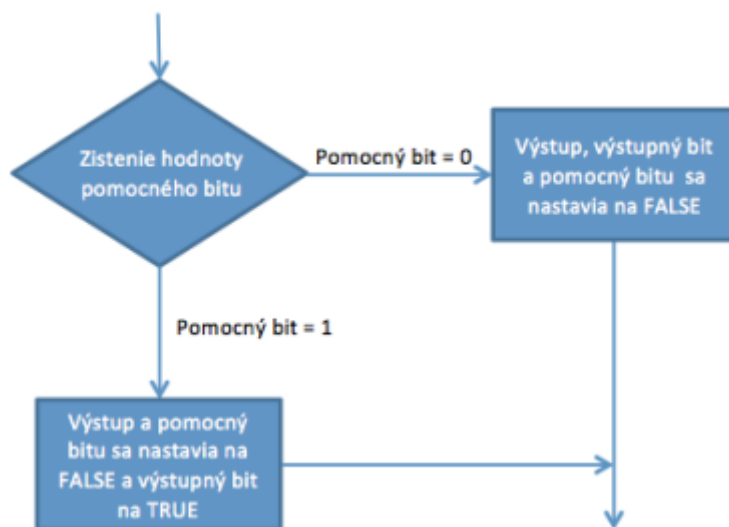


Obr. 11 Ukážka inštrukcie OSR

Inštrukcia „One Shot Falling“ (OSF)

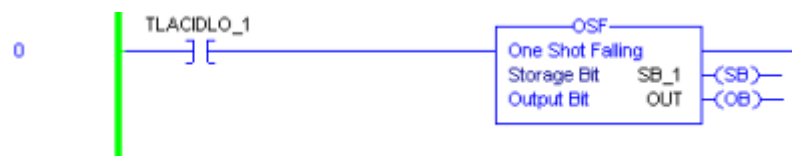
Inštrukcia OSF detekuje dobežnú hranu vstupného signálu. Má rovnaké parametre (storage bit a output bit) ako inštrukcia OSR. Nie je vhodné zapisovať do premennej použitej na uloženie pomocného bitu z iného miesta programu. Postupnosť stavov inštrukcie je nasledovný:

- V stave PRESCAN sa hodnota pomocného bitu (storage bit) nastaví na TRUE a výstupný bit (output bit) na FALSE. Zároveň sa výstup inštrukcie, ktorý ovplyvňuje tok signálu nastaví na FALSE.
- Ak je stav rebríka FALSE, potom sa vykoná inštrukcia podľa vývojového diagramu (obr. 12).
- Ak je stav rebríka TRUE, potom sa hodnota pomocného bitu a výstup inštrukcie nastaví na TRUE. Výstupný bit sa nastaví na FALSE.
- V stave POSTSCAN sa výstup inštrukcie nastaví na FALSE.



Obr. 12 Vývojový diagram inštrukcie OSF

Ukážka inštrukcie OSF sa nachádza na obr. 13. Po zatlačení a následnom uvoľnení tlačidla (TLACIDLO_1) sa bit OUT nastaví na jeden cyklus na hodnotu TRUE.

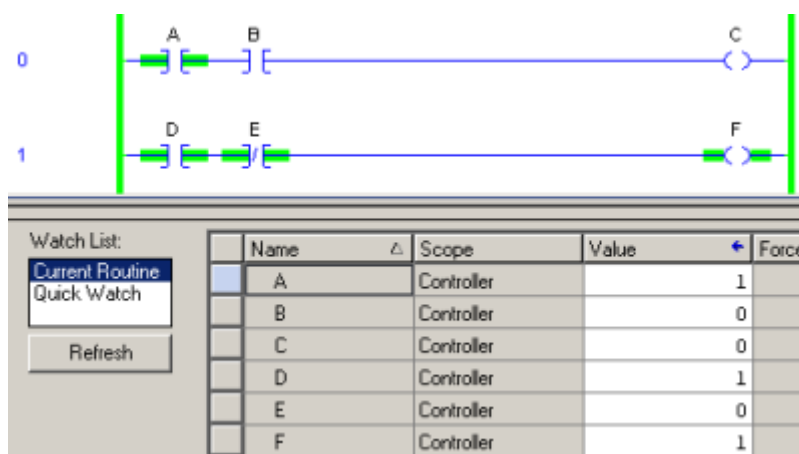


Obr. 13 Ukážka inštrukcie OSF

Príklady použitia inštrukcií

Príklad 1 - Operácia AND

Na obr. 14 sú zobrazené dva príklady operácie OR v kombinácii s operáciou AND. V rebríku č. 0 bude mať tag C hodnotu TRUE ak $A = \text{TRUE}$ a $B = \text{TRUE}$. V druhom príklade v rebríku č. 1 bude mať tag F hodnotu TRUE ak $D = \text{TRUE}$ a $E = \text{FALSE}$.



Obr. 14 Ukážky operácie AND

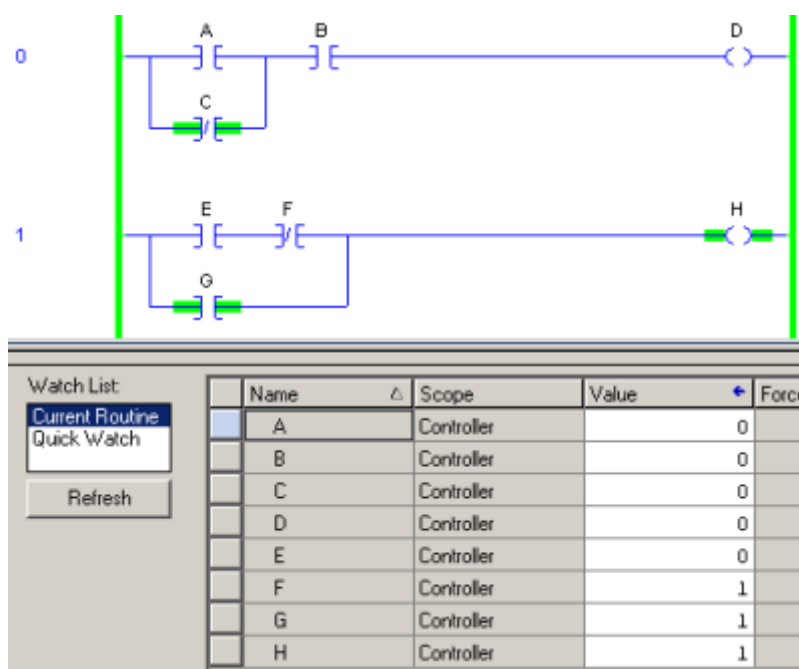
Príklad 2 - Operácia OR

Na obr. 15 sú zobrazené dva príklady operácie AND. V rebríku č. 0 bude mať tag D hodnotu TRUE ak $A = \text{TRUE}$ a $B = \text{TRUE}$ alebo $C = \text{FALSE}$ a $B = \text{TRUE}$. Zápis v textovom jazyku je napr. nasledovný:

$D := (A \text{ OR } \text{NOT}(C)) \text{ AND } B;$

V druhom príklade v rebríku č. 1 bude mať tag H hodnotu TRUE ak $E = \text{TRUE}$ a $F = \text{FALSE}$ alebo $G = \text{TRUE}$. Zápis v textovom jazyku:

$H := (E \text{ AND } \text{NOT}(F)) \text{ OR } G;$



Obr. 15 Ukážky operácie OR v kombinácii s AND

Záver

Vo väčšine prípadov každý výrobca PLC má (alebo môže mať) rôznu sadu inštrukcií, či sú to základné bitové inštrukcie alebo pokročilejšie inštrukcie na komunikáciu alebo samotná inštrukcia PID. Cieľom článku bolo oboznámiť čitateľov s bitovými inštrukciami softvéru RsLogix 5000.

Podakovanie

Tento príspevok vznikol vďaka podpore Nadácie Tatra banky - E-Talent z Nadácie Tatra banky č. 2013et030.

Literatúra

1. http://literature.rockwellautomation.com/idc/groups/public/documents/webassets/browse_category.hcst
2. Interné prednášky z predmetu Softvér riadiacich systémov na FEI STU Bratislava

Spoluautorom článku je Leo Mrafko, Ústav robotiky a kybernetiky, Fakulta elektrotechniky a informatiky, Slovenská technická univerzita, Ilkovičova 3, 812 19 Bratislava, Slovenská republika
