

RsLogix5000 - Porovnávacie bloky

Körösi Ladislav · Elektrotechnika

20.10.2014



Porovnávacie inštrukcie sú riadiace bloky, ktoré sa môžu nachádzať na ľubovoľnej pozícii v rebríkovej schéme (okrem polohy úplne vpravo). Slúžia na porovnanie hodnôt nebinárnych operandov, ktoré sú zvyčajne tagy (premenné) alebo konštanty. Ich logický výstup je závislý od typu porovnania. Tieto inštrukcie majú široké použitie a pre je cieľom článku oboznámiť čitateľa so základnými porovnávacími inštrukciami softvéru RsLogix 5000 v jazyku rebríkových diagramov. Článok je pokračovaním predošlých troch článkov o základných inštrukciách softvéru RsLogix5000 od spoločnosti Rockwell Automation.

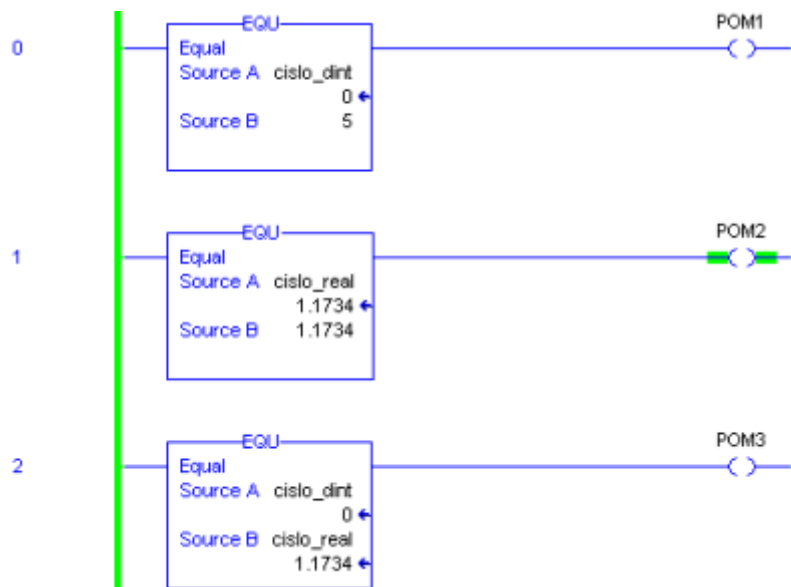
Pre všetky inštrukcie predstavené v článku platia nasledovné stavy:

- 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 a vykoná sa porovnanie dvoch operandov podľa zvolenej inštrukcie (EQU - „=“, NEQ - „<>“, GRT - „>“, GEQ - „>=“, LES - „<“, LEQ - „<=“, atď.)
- V stave POSTSCAN je výstupom inštrukcie logická nula (FALSE).

Inštrukcia "Equal To" (EQU)

Inštrukcia EQU testuje či sa operandy Source A a Source B rovnajú. Operandami inštrukcie môžu byť tagy dátových typov SINT, INT, DINT, REAL a String alebo konštanty. Oba operandy môžu byť teoreticky konštanty, ale inštrukcia v takom prípade stráca zmysel. Ak sa hodnoty operandov rovnajú, potom je výstupom inštrukcie hodnota TRUE, inak FALSE. Dátové typy SINT a INT sú automaticky konvertované na dátový typ DINT. Pre porovnanie dátových typov REAL sa odporúča použiť inštrukciu LIM.

Na obr. 1 sú tri príklady inštrukcie EQU. V prvom príklade sa testuje či je hodnota tagu `cislo_dint` (dátového typu DINT) väčšia ako číslo 5. Aktuálna hodnota tagu je 0, preto je výstupom inštrukcie hodnota FALSE, t.j. tag `POM1` má hodnotu FALSE. V druhom príklade sa tag `cislo_real` (dátového typu REAL) porovnáva s hodnotou 1.1734. Nakoľko aktuálna hodnota tagu `cislo_real` sa rovná operandu Source B, tak je výstupom inštrukcie hodnota TRUE, t.j. tag `POM2` má hodnotu TRUE. V poslednom príklade je ukážka porovnania operandov rôznych dátových typov.



Obr. 1 Príklady inštrukcie EQU v jazyku LD

Ekvivalentnou inštrukciu v jazyku textovom jazyku ST pre prvý príklad na obr. 1 je

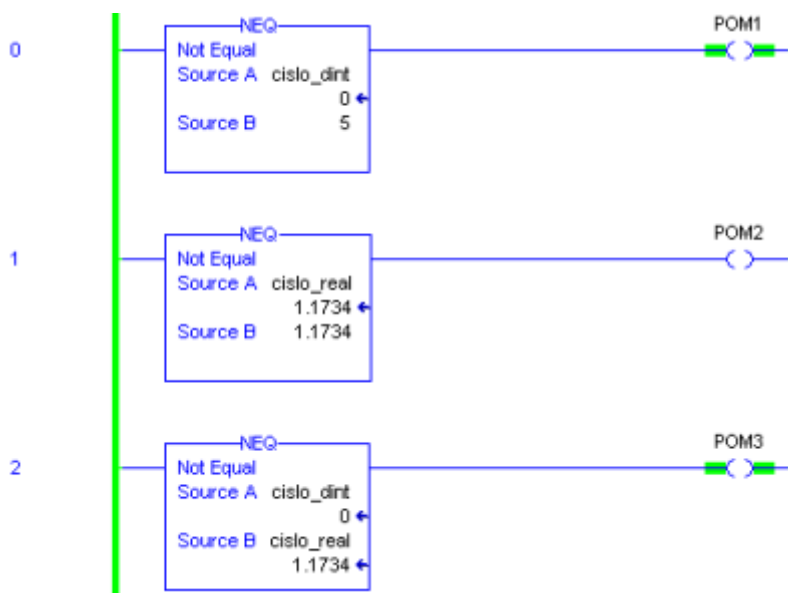
```
POM1 :=(cislo_dint = 5);
```

alebo

```
IF (cislo_dint = 5) THEN
  POM := TRUE;
ELSE
  POM := FALSE;
END_IF;
```

Inštrukcia "Not Equal To" (NEQ)

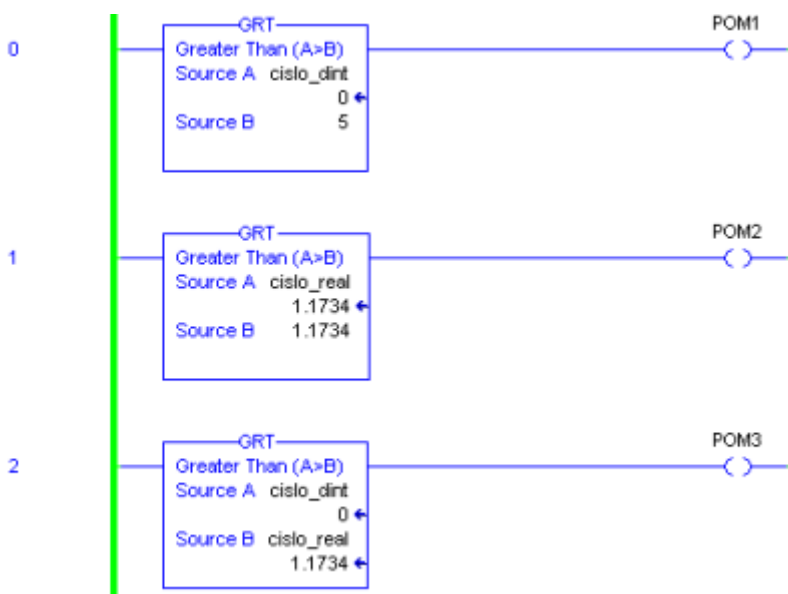
Inštrukcia NEQ testuje kedy sa operandy Source A a Source B nerovnajú. Ak sa hodnoty operandov rovnajú, potom je výstupom inštrukcie hodnota FALSE, inak TRUE. V prvom príklade sa na obr. 2 testuje či je aktuálna hodnota tagu cislo_dint rôzna od hodnoty 5. Aktuálna hodnota tagu je 0 preto je podmienka platná. V druhom príklade sa hodnoty operandov rovnajú, preto je výstupom inštrukcie hodnota FALSE. V poslednom príklade sú hodnoty operandov rôzne, preto inštrukcia dáva na výstupe hodnotu TRUE.



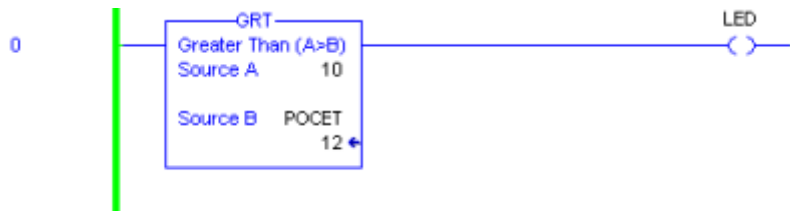
Obr. 2 Príklady inštrukcie NEQ v jazyku LD

Inštrukcia " Greater Than " (GRT)

Inštrukcia GRT testuje kedy je operand Source A väčší ako operand Source B. Ak platí podmienka $\text{Source A} > \text{Source B}$, potom je výstupom inštrukcie TRUE, inak FALSE. Vo všetkých troch príkladoch na obr. 3 neplatí podmienka porovnania, preto je výstupom inštrukcie hodnota FALSE. Inštrukciu GRT, ale aj ďalšie inštrukcie nižšie (GEQ, LES, LEQ) je možné použiť „neštandardne“ ako je to v príklade na obr. 4. Jedná sa o výmenu poradia operandov v Source A a Source B. Zaužívané je však použitie neznámej (premennej) ako prvého operandu a preto je viac intuitívne použitie inštrukcie LES s vymenenými operandami, lebo LED bude mať hodnotu TRUE len v prípade ak je $\text{POCET} \leq 10$.



Obr. 3 Príklady inštrukcie GRT v jazyku LD

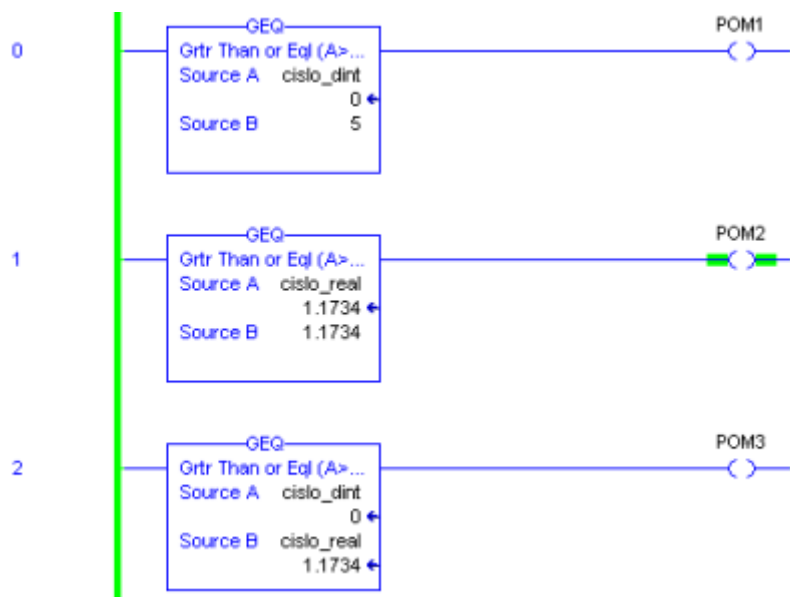


Obr. 4 Neštandardné použitie inštrukcie GRT

Inštrukcia "Greater Than or Equal To" (GEQ)

Inštrukcia GEQ testuje kedy je operand Source A väčší alebo rovný ako operand Source B. Ak platí podmienka $\text{Source A} \geq \text{Source B}$, potom je výstupom inštrukcie TRUE, inak FALSE.

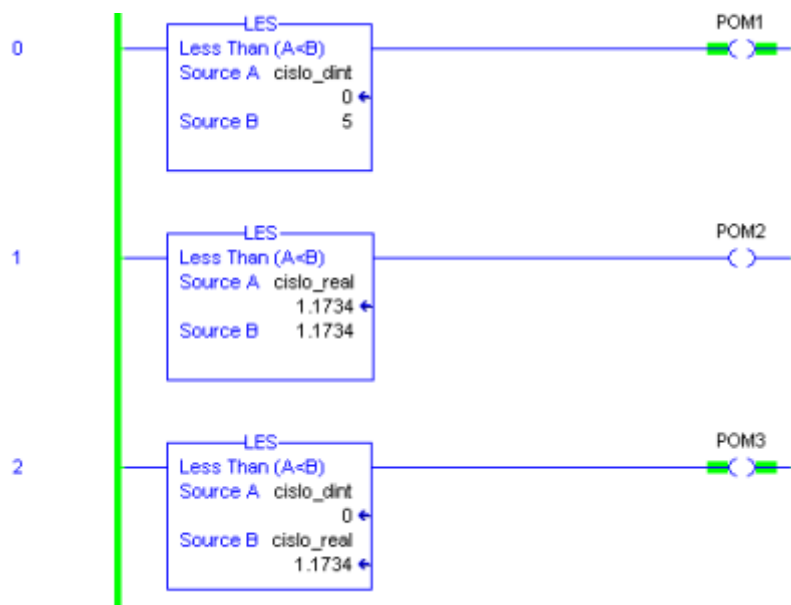
Na obr. 5 sú porovnané rovnaké operandy s rovnakými aktuálnymi hodnotami ako to bolo na obr. 3, len je použitá inštrukcia GEQ (Source A \geq Source B) namiesto GRT (Source A $>$ Source B). Z toho dôvodu na obr. 4 už platí podmienka v strednom príklade, preto má tag POM2 hodnotu TRUE.



Obr. 5 Príklady inštrukcie GEQ v jazyku LD

Inštrukcia "Less Than" (LES)

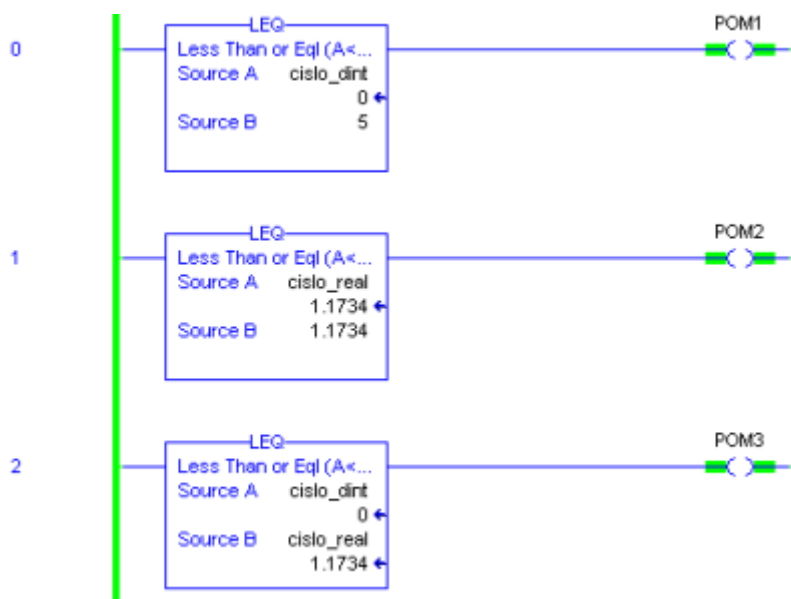
Inštrukcia LES testuje kedy je operand Source A menší ako operand Source B. Ak platí podmienka $\text{Source A} < \text{Source B}$, potom je výstupom inštrukcie TRUE, inak FALSE.



Obr. 6 Príklady inštrukcie LES v jazyku LD

Inštrukcia "Less Than or Equal To" (LEQ)

Inštrukcia LEQ testuje kedy je operand Source A menší alebo rovný ako operand Source B. Ak platí podmienka $Source\ A \leq Source\ B$, potom je výstupom inštrukcie TRUE, inak FALSE.



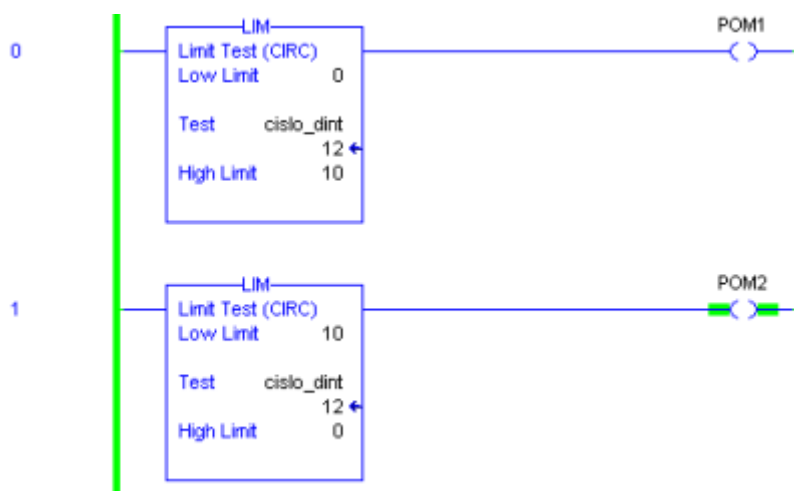
Obr. 7 Príklady inštrukcie LEQ v jazyku LD

Inštrukcia "Limit" (LIM)

Inštrukcia LIM testuje či je hodnota operandu Test v rozmedzí od operandu Low Limit po operand High Limit. Ekvivalentom inštrukcie LIM v jazyku ST je

```
IF (LowLimit <= HighLimit AND (Test >= LowLimit AND Test <=
HighLimit)) OR (LowLimit >= HighLimit AND (Test <= LowLimit
OR Test >= HighLimit)) THEN
  <príkazy>
END_IF;
```

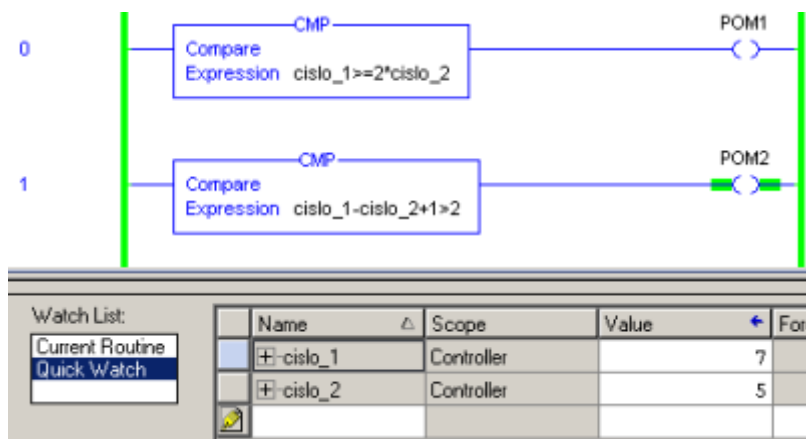
V prvom príklade na obr. 8 je ukážka použitia inštrukcie LIM pri klasickom programovaní. Inštrukcia slúži na zistenie či sa hodnota tagu `cislo_dint` nachádza v rozmedzí od 0 po 10. Aktuálna hodnota tagu je 12, preto je na výstupe inštrukcie signál FALSE. V druhom príklade je pri Low Limit zadaná vyššia hodnota ako pri High Limit. Takto zadané operandy slúžia na testovanie, či je `cislo_dint` mimo rozsahu 0 až 10. Aktuálna hodnota tagu je 12, preto je na výstupe inštrukcie signál TRUE. Využíva sa to napr. pri generovaní alarmu analógovej veličiny. Samozrejme inštrukcia LIM môže byť nahradená inštrukciami GRT a LES (resp. GEQ a LEQ), ktoré sú zapojené do série alebo paralelne v závoslosti od požadovanej funkcionality.



Obr. 8 Príklady inštrukcie LIM v jazyku LD

Inštrukcia "Compare" (CMP)

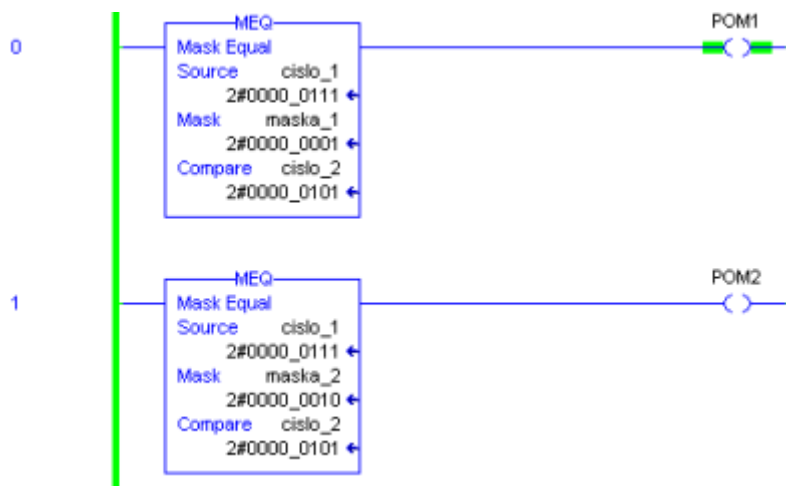
Operandom inštrukcie CMP je Expression (výraz), ktorý realizuje ľubovoľné porovnanie (<, <=, >, >=) pričom na oboch stranách porovnania môžu byť rôzne matematické výrazy. Vykonanie inštrukcie zaberie viac času z cyklu PLC ako pri použití základných porovnávacích resp. matematických inštrukcií, preto jeho použitie je na zváženie. V prvom príklade na obr. 9 je porovnanie výrazu $\text{cislo}_1 \geq 2 * \text{cislo}_2$ pomocou inštrukcie CMP. Po dosadení aktuálnych hodnôt $2 \geq 2 * 5$ je zrejmé, že výraz neplatí a preto výstupom inštrukcie je FALSE. V druhom príklade je výsledok matematického výrazu viac ako 2, preto je na výstupe inštrukcie hodnota TRUE.



Obr. 9 Príklady inštrukcie CMP v jazyku LD

Inštrukcia "Mask Equal To" (MEQ)

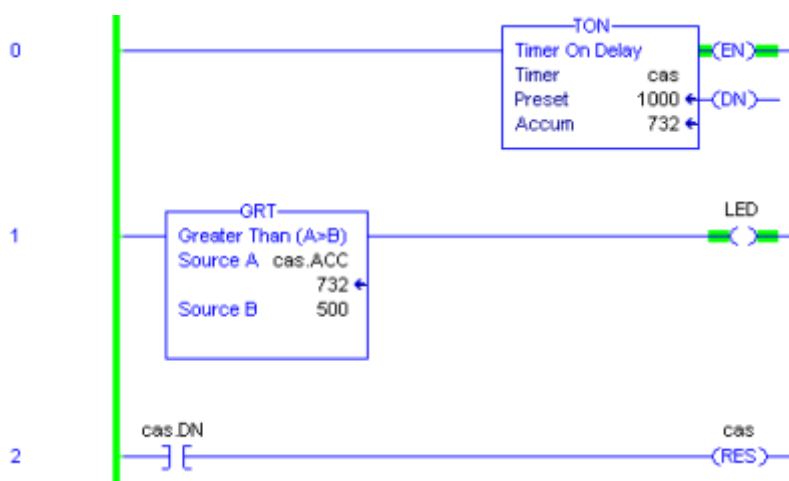
Inštrukcia MEQ vykonáva porovnanie operandov Source a Compare s prihliadnutím na masku Mask. Jedná sa o porovnanie po bitoch, preto ak sa hodnoty bitov Source a Compare rovnajú na všetkých miestach kde sa v maske Mask nachádza logická jednotka, tak je výstupom inštrukcie hodnota TRUE. V prvom príklade na obr. 10 sa porovnávajú dva tagy cislo_1 a cislo_2 oboj dátových typov SINT. Podľa masky maska_1 sa bude v každom cykle porovnávať nultý (najnižší) bit operandov cislo_1 a cislo_2. Oba bity operandov majú hodnotu TRUE (t.j. sú rovnaké), preto je výstupom inštrukcie MEQ hodnota TRUE. V druhom príklade sa porovnáva bit 1 (2. Najnižší bit) podľa masky maska_2. Pri operande cislo_1 je hodnota bitu TRUE a pri operande cislo_2 hodnota FALSE, preto je výstupom inštrukcie hodnota FALSE.



Obr. 10 Príklady inštrukcie MEQ v jazyku LD

Príklad 1 - Blikanie signálky

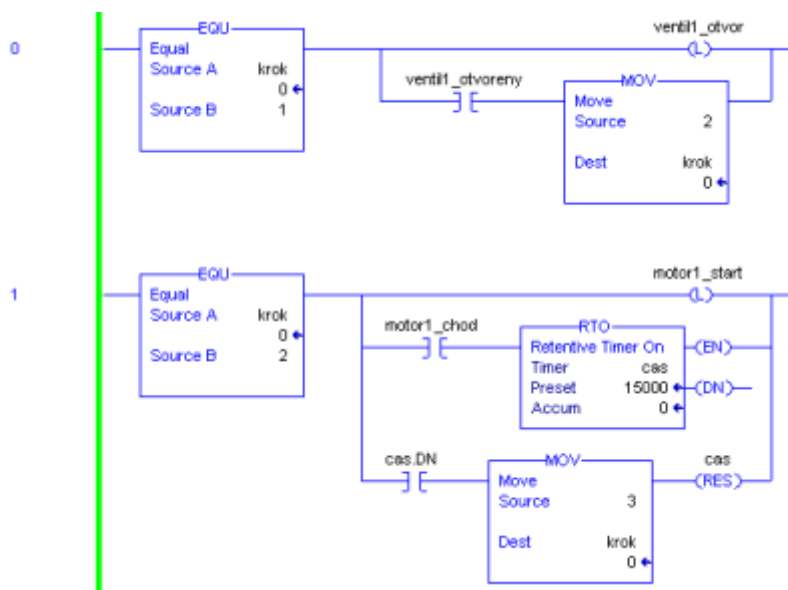
Porovnávacie bloky sa môžu využiť na vetvenie programu v závislosti napr. od uplynutého času. Na obr. 11 je ukážka jednoduchého programu generovania blikajúceho signálu LED pomocou časovača a porovnávacieho bloku GRT. Časovač cas sa spustí (na vstupe je TRUE priamo z ľavej strany rebríka - rebrík č. 0) a po 1 sekunde sa vynuluje (rebrík č. 2) čím začne v ďalšom cykle rátať čas odznova. V rebríku č. 1 sa porovnáva aktuálny čas časovača s hodnotou 500ms. Ak je uplynutý čas väčší ako 0,5s, tak LED má hodnotu TRUE (LED svieti), inak má hodnotu FALSE. Takto dostávame blikajúci signál 0,5s zapnutý a 0,5s vypnutý.



Obr. 11 Príklad generovania zmeny signálu

Príklad 2 - Krokové (sekvenčné) riadenie

Príklad jednoduchého krokového riadenia je na obr. 12. Jednotlivé operácie v krokoch sú realizované ak je aktívny krok riadenia. Testovanie čísla kroku od ktorého sa odvíja program je pomocou porovnávacieho bloku EQU. V rebríku č. 0 sa testuje či je krok riadenia rovný 1. Ak je podmienka splnená, tak sa nastaví príkaz na otváranie ventilu1 (ventil1_otvor) a po detekcii spätnej väzby o otvorení ventilu (ventil1_otvoreny) sa prechádza do kroku 2 (inštrukcia MOV). V kroku 2 (rebrík č. 1) sa zapne chod miešadla (motor1_start), ktoré mieša médium v nádrži. Po 15s miešania (cas.DN) sa prechádza na krok 3 (inštrukcia MOV), atď.



Obr. 12 Príklad krokového riadenie

Záver

Porovnávacie inštrukcie majú významné zastúpenie v praktických aplikáciách spojitého, ale i diskretného charakteru riadenia. Často sa využívajú na kontrolu, či napr. meraná hodnota je v danom rozsahu a prípadné generovanie alarmových hlásení. V diskretnom riadení je ich možné s výhodou využiť pri spomínanom sekvenčnom riadení, ale napr. i pri jednoduchých aplikáciách polohovania a pod.

PodĎakovanie

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, FEI STU, Ilkovičova 3, 812 19 Bratislava, Slovenská republika

