

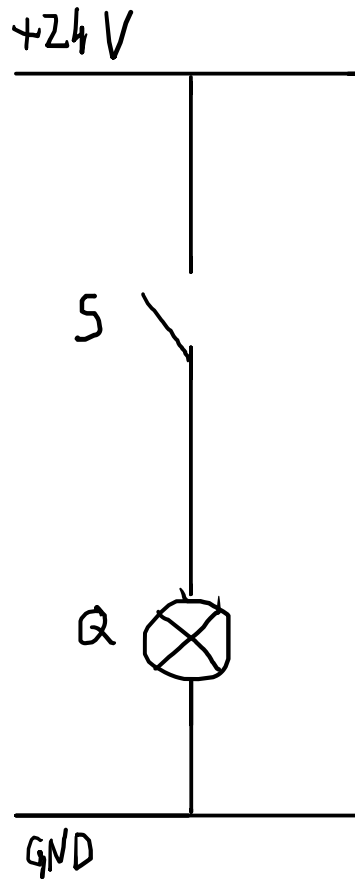
Programirljivi logični krmilnik: programiranje - 1. del

Procesna avtomatika

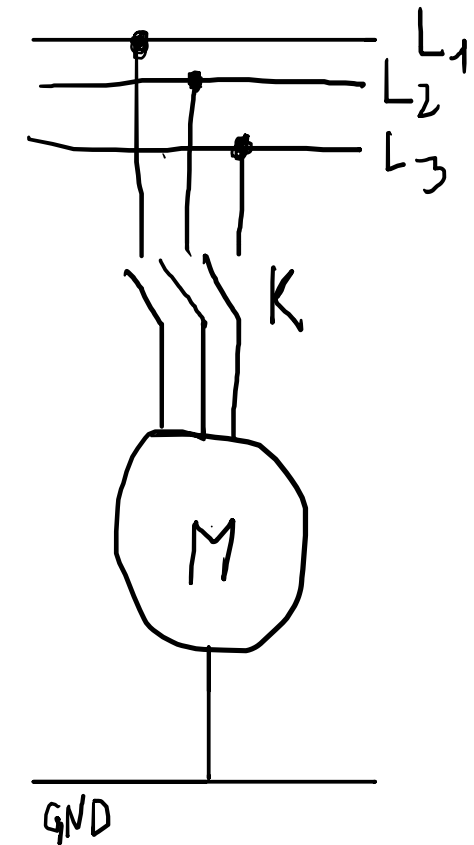
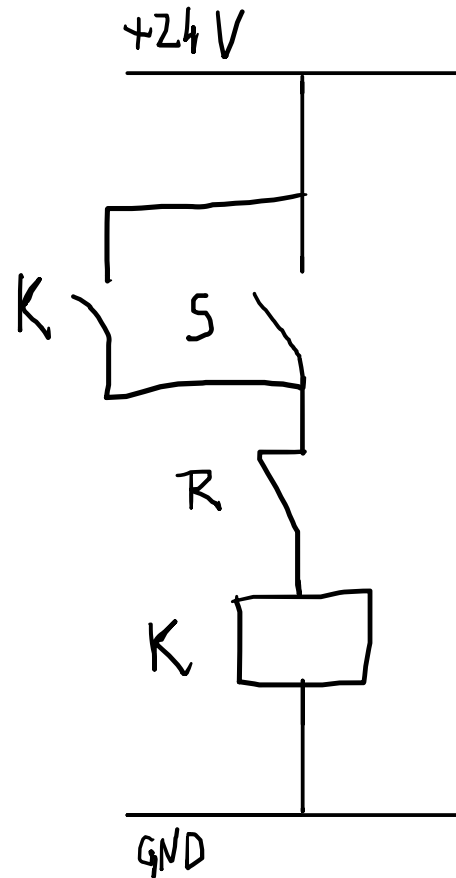
Uroš Lotrič, Nejc Ilc

Primer: programiranje brez PLK

Primer: stikalo in žarnica



Primer: krmiljenje elektromotorja



Razvoj standarda

Prej

- Lestvični diagrami slabo strukturirani
- Kompleksni visokonivojski jeziki za določene aplikacije in jeziki z namenom poenostavljanja programiranja

Standard IEC 61131

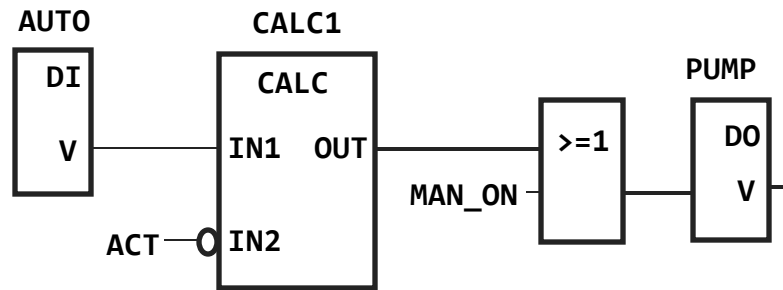
- Postavljen leta 1993, trenutna 3. izdaja iz leta 2013 (opisuje objektni pristop k programiranju)
- Poskus poenotenja raznoraznih konceptov programiranja PLK
- 3. del standarda (IEC 61131-3) definira programske strukture in programske jezike:
 - lestvični diagram (ladder diagram)
 - funkcijski načrt (function block diagram)
 - diagram poteka (sequential function chart)
 - strukturirano besedilo (structured text)
 - lista ukazov (instruction list) - v 3. izdaji je označen kot zastarel ([opomba za TwinCAT](#))

Zdaj

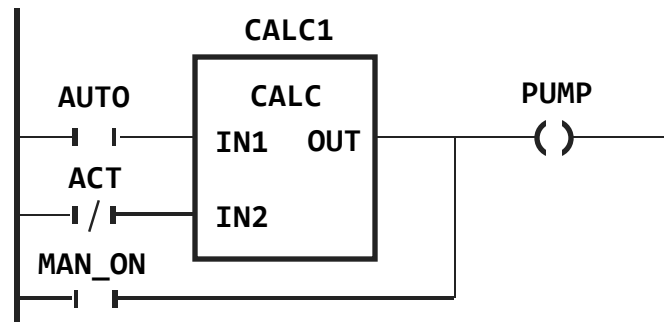
- Programi *skorajda* prenosljivi med različnimi proizvajalci ali serijami,
- koncepti programiranja pa zagotovo.

Standard IEC 61131-3

- Funkcijski načrt (FBD)



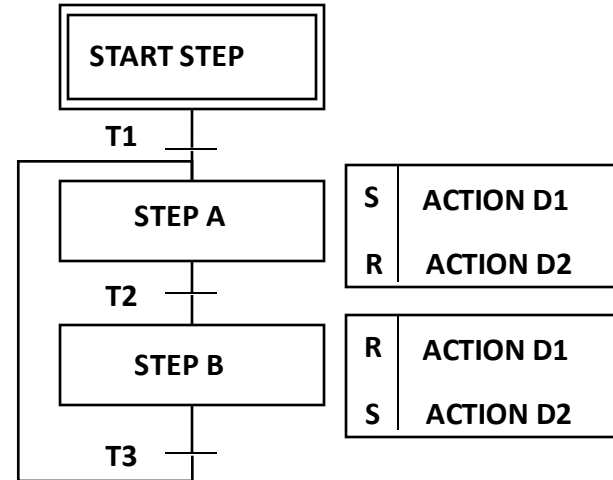
- Lestvični diagram (LD)



- Lista ukazov (IL)

```
A: LD    %IX1 (* PUSH BUTTON *)
      ANDN %MX5 (* NOT INHIBIT *)
      ST  %QX2 (* FAN ON *)
```

- Diagram poteka (SFC)



- Strukturirano besedilo (ST)

```
VAR CONSTANT
X : REAL := 53.8; Z : REAL; END_VAR
VAR aFB, bFB : FB_type; END_VAR

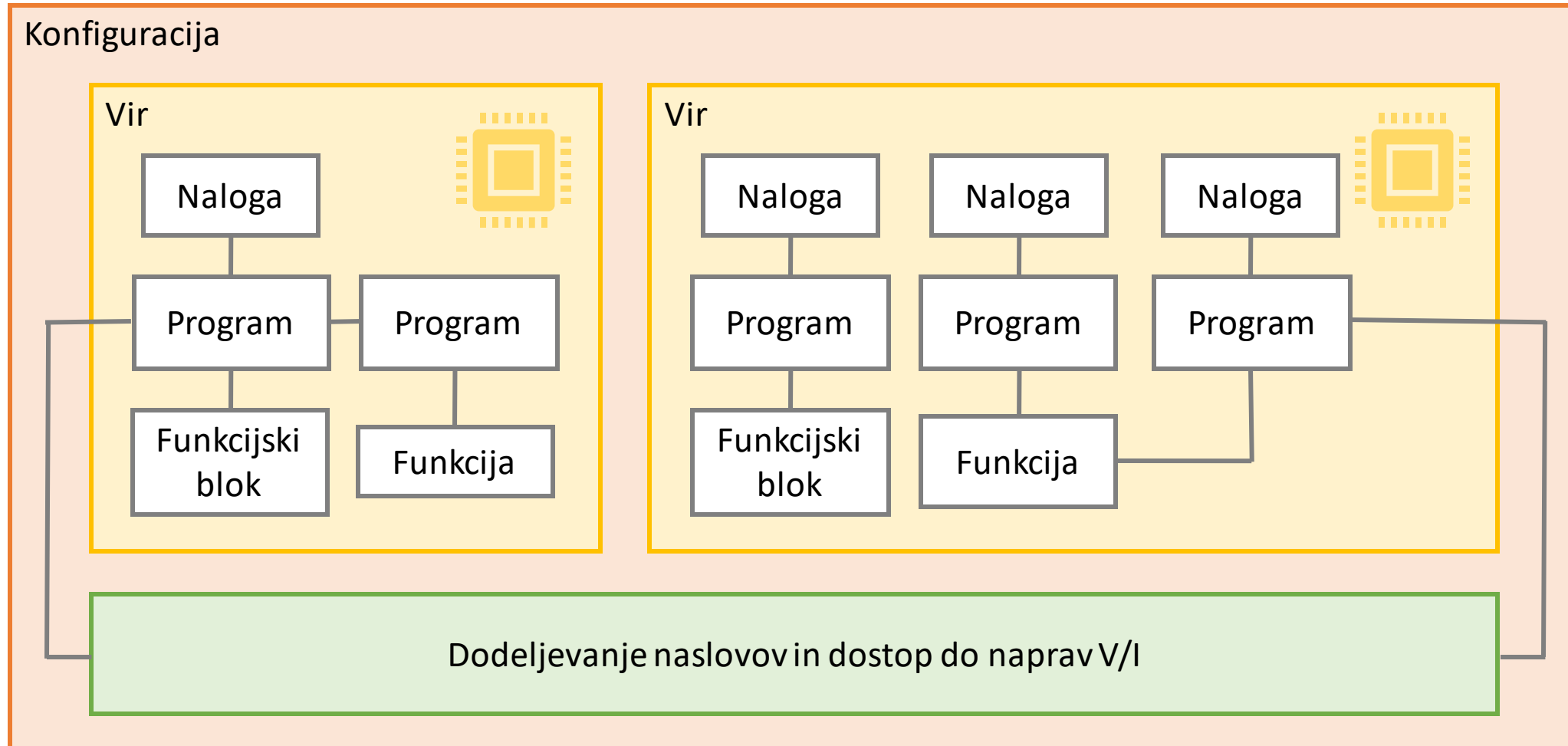
bFB(A:=1, B:='OK');
Z := X - INT_TO_REAL (bFB.OUT1);
IF Z > 57.0 THEN aFB(A:=0, B:="ERR");
ELSE aFB(A:=1, B:="Z is OK");
END_IF
```

Tipične programske strukture in podatkovni tipi

- Digitalni vhodi: %I*
- Digitalni izhodi: %Q*
- Spomin: %M*
- Časovniki
 - Ob izpolnjenem pogoju meri čas. Ko doseže nastavljeni čas sproži svoj kontakt.
- Števniki
 - Ob izpolnjenem pogoju šteje navzgor ali navzdol.
- Združevanje binarnih spremenljivk
 - bajt (byte) – 8 bit, beseda (word) – 16 bit, dvojna beseda (double word) – 32 bit
- Celoštevilčne spremenljivke (16- ali 32-bitne)
- Necela števila v formatu plavajoče vejice (3.14, 1.64e+009)
- Nizi ('Zivjo! '),
- Čas (T#12h34m15s),
- Datum (D#2023-07-23)

Programski objekti – model

Standard definira model dodeljevanja virov posameznim objektom.



Objekti, ki vsebujejo programsko kodo za PLK – POU (*program organization unit*)

Program – PRG

- Organizacijska enota na najvišjem nivoju.
- Vsak projekt potrebuje vsaj en program (tipično MAIN).
- Program lahko vsebuje klice drugih programov, funkcijskih blokov in funkcij.
- Program lahko kličemo iz drugih programov in funkcijskih blokov, ne moremo pa ga iz funkcij.
- Vrstni red klicev programov v projektu je definiran v objektu *naloga* (task). Naloga inicializira program.
- Ko se program zaključi, se vrednosti spremenljivk ohranijo do naslednjega klica programa.

Funkcijski blok – FB

- Ima lahko več vhodov (VAR_INPUT) in izhodov (VAR_OUTPUT).
- Ima svoj lastni spomin za notranje spremenljivke (VAR, VAR_STAT).
- Notranje in izhodne spremenljivke ohranijo svojo vrednost do naslednjega klica,
- kar pomeni, da FB lahko vrne različne vrednosti ob zaporednih klicih z istimi vhodnimi argumenti.
- Funkcijski blok se vedno kliče preko instance, tj. kopije. Vsaka instanca ima svojo kopijo spremenljivk.
- Najbolj pogost tip POU.

Funkcija – FUN

- Ob klicu vrne natanko en element (lahko je polje ali struktura).
- Ima lahko več vhodov (VAR_INPUT) in izhodov (VAR_OUTPUT).
- Nima spomina, torej se vrednosti spremenljivk med zaporednimi klici ne ohranjajo,
- kar pomeni, da ob istih vrednostih vhodnih parametrov vedno vrne isto izhodno vrednost.
- PS. Vseeno pa omogoča uporabo statičnih spremenljivk VAR_STAT ...

Programski objekti

Vsak programski objekt ima dva dela:

- Deklaracije

- Spremenljivke, ki definirajo vmesnik (*interface*):

- Vhod: VAR_INPUT
- Izhod: VAR_OUTPUT
- Vhod/izhod: VAR_IN_OUT

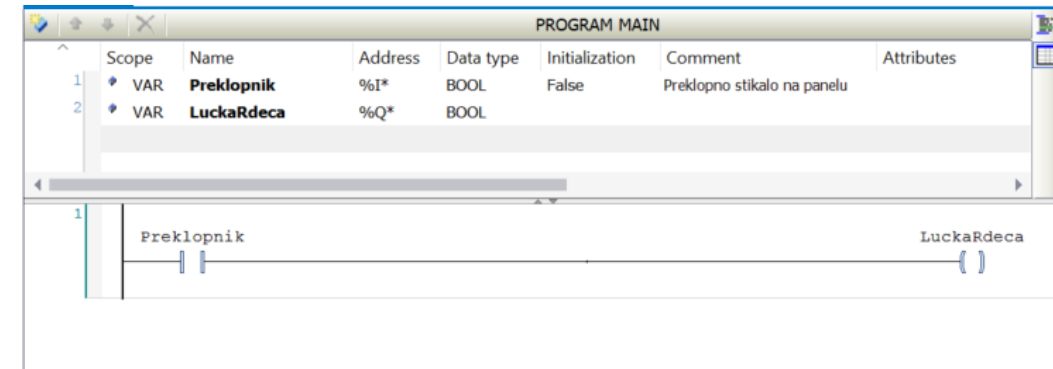
- Lokalne spremenljivke:

- VAR
- VAR_TEMP (samo v PRG in FB): inicializacija ob vsakem klicu bloka
- VAR_STAT (samo v FUN in FB): inicializacija samo ob prvem klicu, samo en original (gre za isto spremenljivko, četudi imamo več klicev FUN ali več instanc FB)

- Globalne spremenljivke (samo v GVL, ki nima implementacijskega dela): VAR_GLOBAL

- Implementacija

- Programska koda v izbranem jeziku IEC, ki implementira določeno funkcionalnost



Zagon

Začetni zagon (*reset original*)

- Vse spremenljivke se ponastavijo na začetne vrednosti, prav tako se ponastavi projekt na krmilniku

Hladen zagon (*reset cold*)

- Vse spremenljivke se ponastavijo na začetne vrednosti, razen tiste, ki so nastavljene kot [PERSISTENT](#) oziroma [RETAIN](#)

Naslavljanje

Sintaksa:

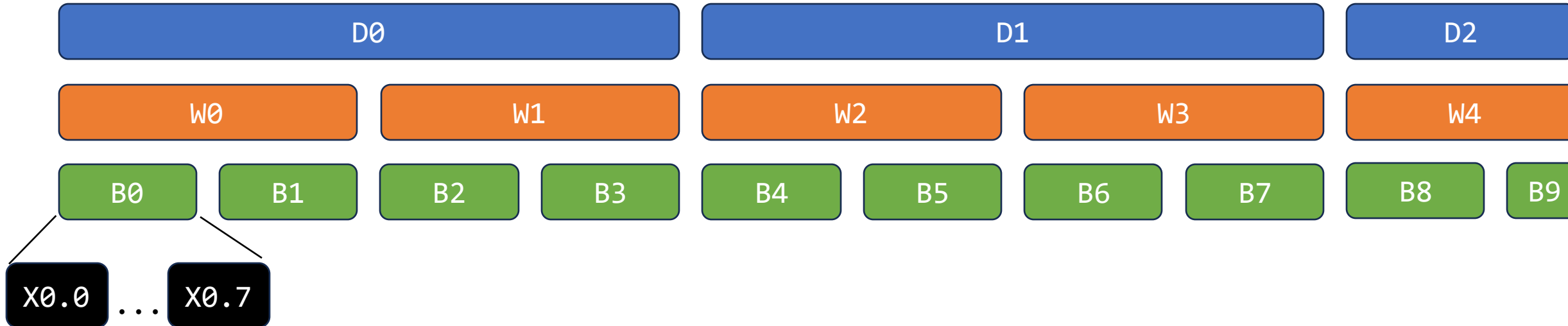


Primeri

- %IX1.0 : vhodni bit 0 v bajtu z naslovom 1 (1 bit)
- %QB2 : izhodni bajt na naslovu 2 (8 bitov)
- %MW4 : podatek v spominu, 5. zaporedna beseda (16 bitov)
- %QD24 : izhodna dvojna beseda, 25. zaporedna (32 bitov)

Naslavljanje – prekrivanje

V pomnilniku so biti, bajti, besede in dvojne besede razporejeni tako:



POZOR: Če zapišemo nekaj na %MW0 in nato nekaj na %MB1, smo pisali na prekrivajoče območje v pomnilniku.

NASVET: v TwinCAT svetujemo avtomatično dodelitev naslovov z uporabo nadomestnih znakov: %I*, %Q*, %M*. S tem omogočimo optimizacija dostopov in prilagodljivost naslovnega prostora.

Podatkovni tipi

Podatkovni tip	Spodnja meja	Zgornja meja	Velikost (biti)
BIT	FALSE (0)	TRUE (1)	1
BOOL	FALSE (0)	TRUE (1)	8
BYTE	0 (desetiško) 2#0000_0000 (dvojiško) 8#000 (osmiško) 16#00 (šestnajstiško)	255 2#1111_1111 8#377 16#FF	8
WORD	0	65535	16
DWORD	0	4294967295	32
LWORD	0	$2^{64}-1$	64
TIME	0 = T#0ms	4294967295 T#49d17h2m47s295ms	32
LTIME	0 = T#0ns	T#213503d23h34m33s709ms551us615ns	64
DATE	0 = D#1970-01-01	4294967295 = D#2106-02-07	32
DATE_AND_TIME	0 = DT#1970-1-1-0:0:0	4294967295 = DT#2106-02-07-06:28:15	32

Velja za vnašanje konstant pri vseh številskih tipih

Podatkovni tipi - nadaljevanje

Podatkovni tip	Spodnja meja	Zgornja meja	Velikost (biti)
SINT	-128	127	8
USINT	0	255	8
INT	-32768	32767	16
UINT	0	65535	16
DINT	-2147483648	2147483647	32
UDINT	0	4294967295	32
LINT	-2^{63}	$2^{63}-1$	64
ULINT	0	$2^{64}-1$	64
REAL	$-3.402823e+38$	$3.402823e+38$	32
LREAL	$-1.7976931348623158e+308$	$1.7976931348623158e+308$	64

Glejte tudi [STRING](#), [WSTRING](#), [ARRAY](#), [ANY](#)

Lestvični diagrami: standard

Najbolj uporabljan jezik za PLK

Grafični jezik, ki omogoča enostaven prehod iz električnih shem v programiranje

Standard 61131-3

- Navpični tirnici
 - Osnova lestvičnega diagrama
 - Leva tirnica je priključena na visoko napetost in predstavlja logično 1
 - Desna tirnica je priključena na nizko napetost in predstavlja logično 0
- Priključeni elementi so predstavljeni z vodoravnimi povezavami
 - Stanje priključenega elementa je definirano kot “vklopljen” ali “izklopljen” in ustreza logičnim vrednostim 1 in 0. V prvem primeru prepušča električni tok, v drugem ne.
- Vodoravna povezava
 - Stanje vodoravne se spreminja glede na to ali skozi element teče električni tok ali ne
 - Kadar je element vklopljen vodoravna povezava prenaša stanje vodoravne povezave na levi strani elementa na vodoravno povezavo na njegovi desni strani
- Navpične povezave povezujejo enega ali več priključenih elementov
 - Stanje navpične povezave je “izklopljeno”, če so vse vodoravne povezave izklopljene
 - Stanje navpične povezave je “vklopljeno”, če je vsaj ena vodoravna povezava vklopljena
 - Stanje navpične povezave se prenese na vse vodoravne povezave na njeni desni strani
 - Prenos stanja navpične povezave na levo stran ni dovoljeno

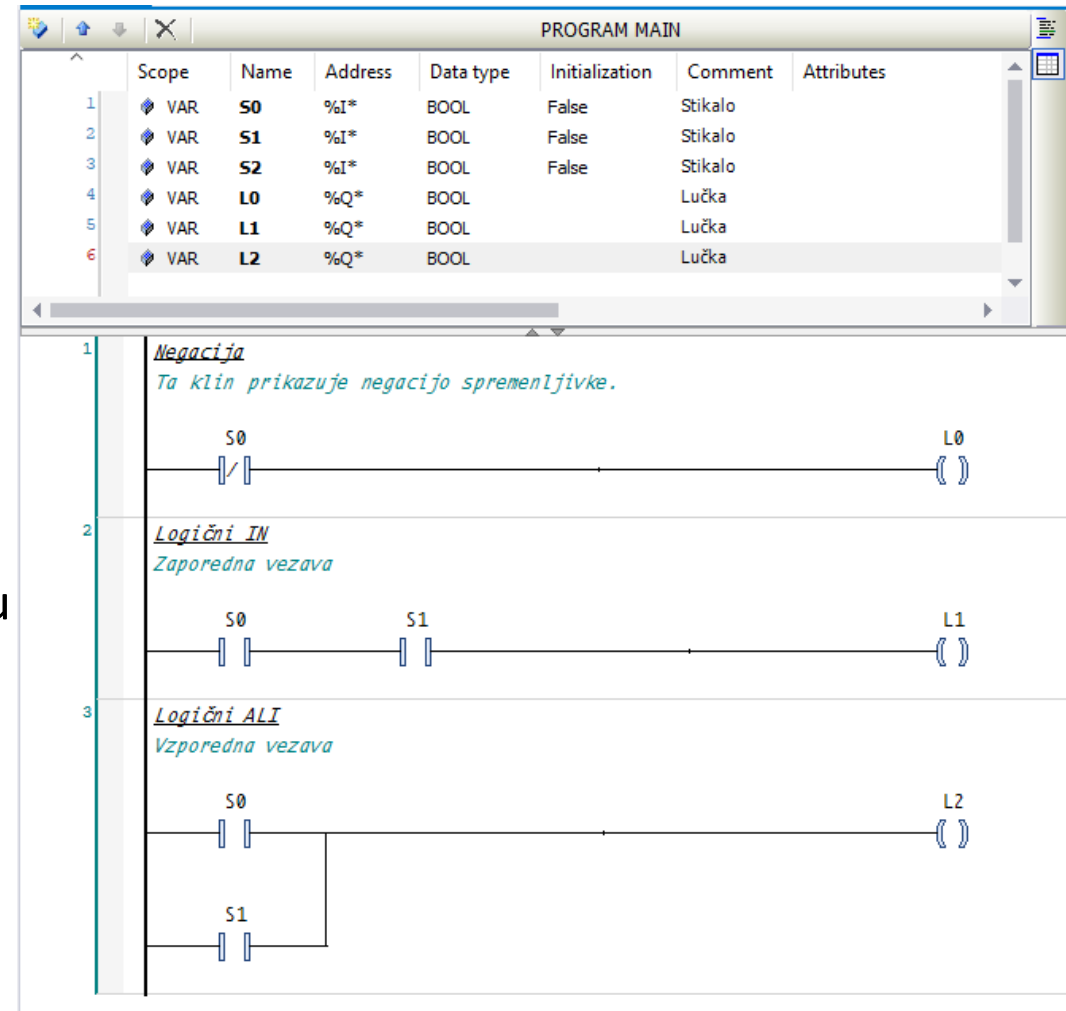
Lestvični diagrami: standard

Standard 61131-3

- Elementi in povezave tvorijo na lestvi kline (*network*)
- Običajno ima vsak klin množico ustrezno povezanih vhodnih pogojev, s katerimi nadziramo eno ali več tuljav (izhodov)
 - Dobra praksa: vsaka tuljava (izhod) naj se v programu pojavi samo na enem mestu

Primer

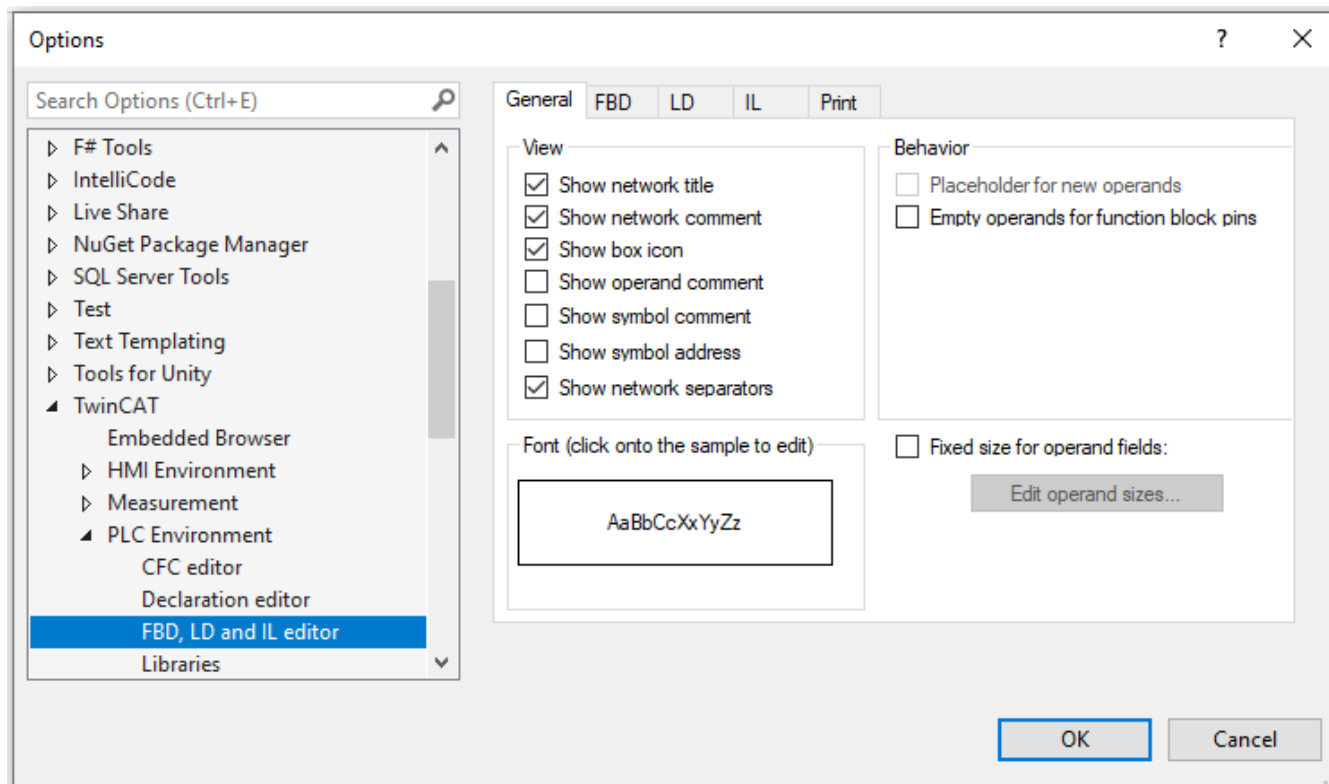
- Osnovne logične operacije



Lestvični diagrami: nastavitve v TwinCAT

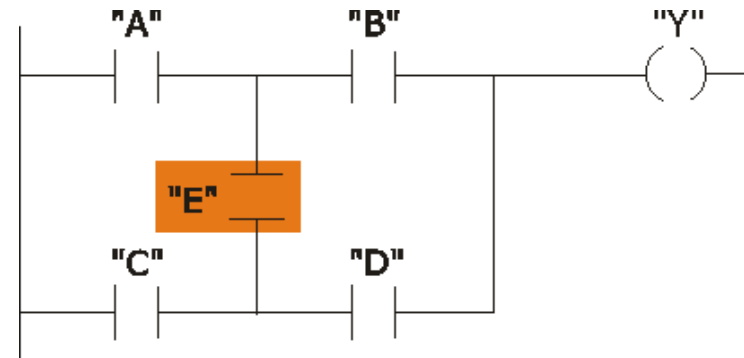
Tools → Options → TwinCAT → PLC Environment

- Prikaz naslovov in komentarjev klinov
- Pisava ...



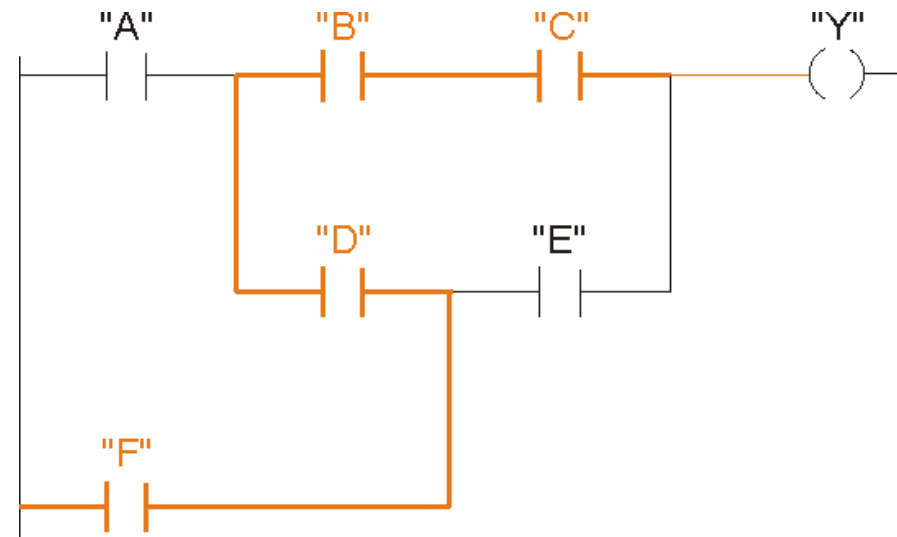
Lestvični diagrami: omejitve

Navpičnih vezav elementov ni



Tok teče samo iz leve na desno

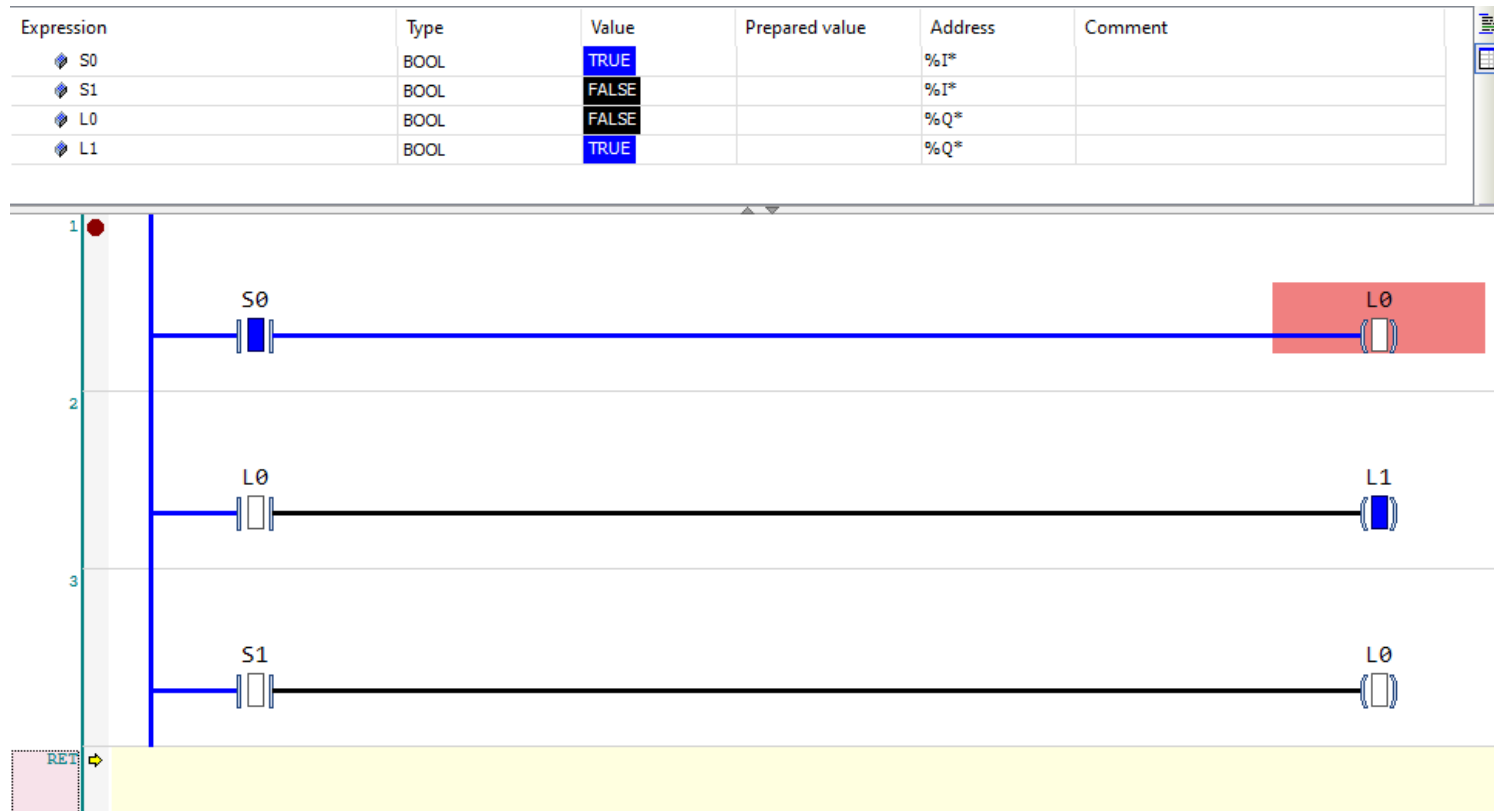
- Običajno orodja izdelave takih vezij ne omogočajo



Lestvični diagrami: spremembe

Spreminjanje vhodov in izhodov

- Vhodi so skozi cel programski cikel konstantni (paralelno)
- Izhodi se spreminjajo klin za klinom (sekvenčno)
 - Na fizične izhode se prenese zadnja vrednost izhoda v pomnilniški sliki



Lestvični diagrami: razširitve

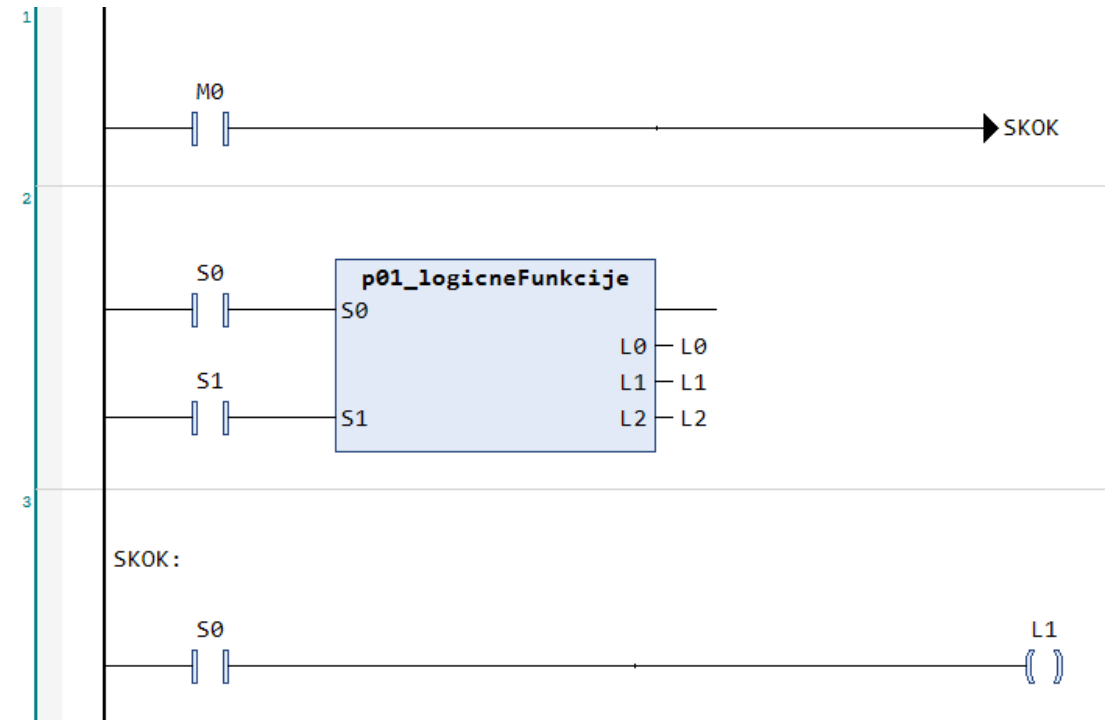
Osnovni lestvični diagram pozna samo stikala in releje

Nujne razširitve

- Klici funkcij in podprogramov
- Skoki
- Uporaba strukturiranih podatkovnih tipov
- Procesiranje analognih vrednosti

Programi postanejo težje berljivi

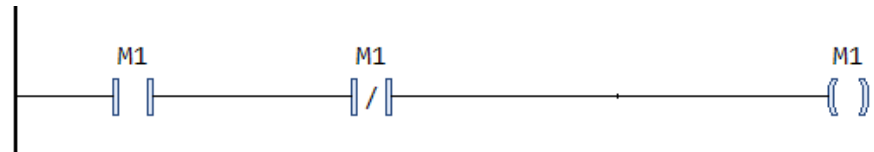
- Intuitivnost stikal in relejev se izgubi
- Niso primerni za velike, kompleksne projekte



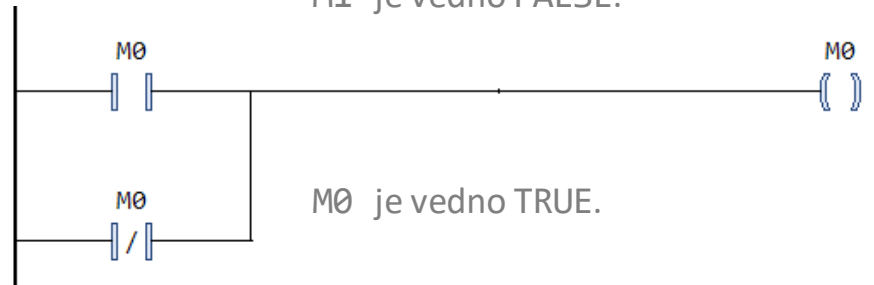
Lestvični diagrami: primeri



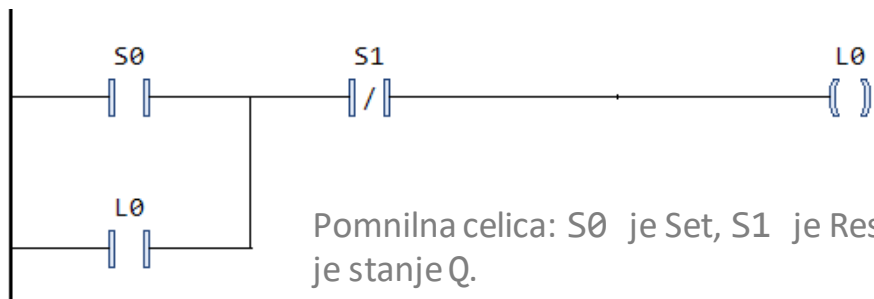
Bit M0 vsak cikel obrne svojo vrednost – dobimo uro.



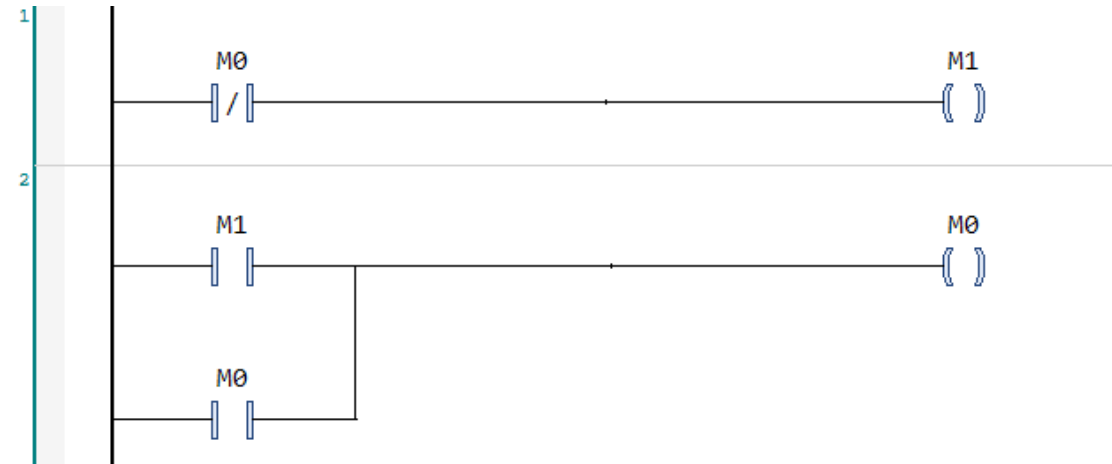
M1 je vedno FALSE.



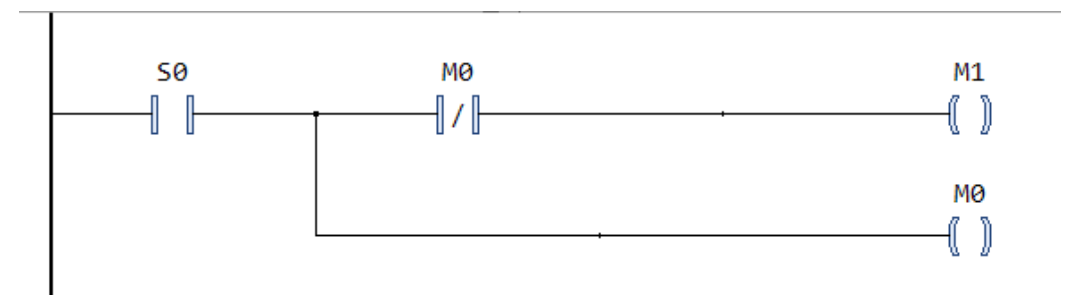
M0 je vedno TRUE.



Pomnilna celica: S0 je Set, S1 je Reset, L0 je stanje Q.



M1 bo TRUE samo prvi cikel izvajanja programa.

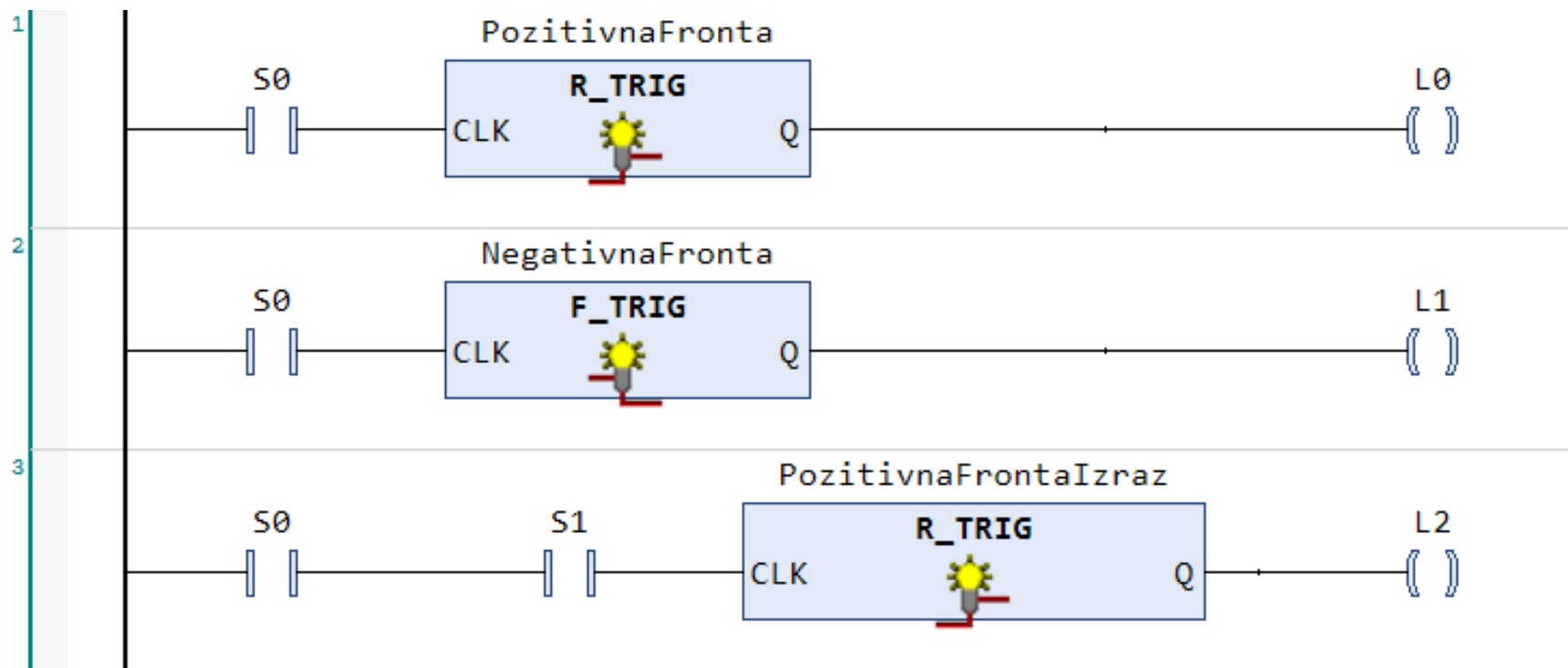


Detekcija pozitivne fronte na S0. M1 bo TRUE natanko en cikel - takrat, ko pride do spremembe na S0 iz FALSE na TRUE.

Lestvični diagrami: zaznavanje fronte

Fronta je sprememba vrednosti enega bita in traja en cikel

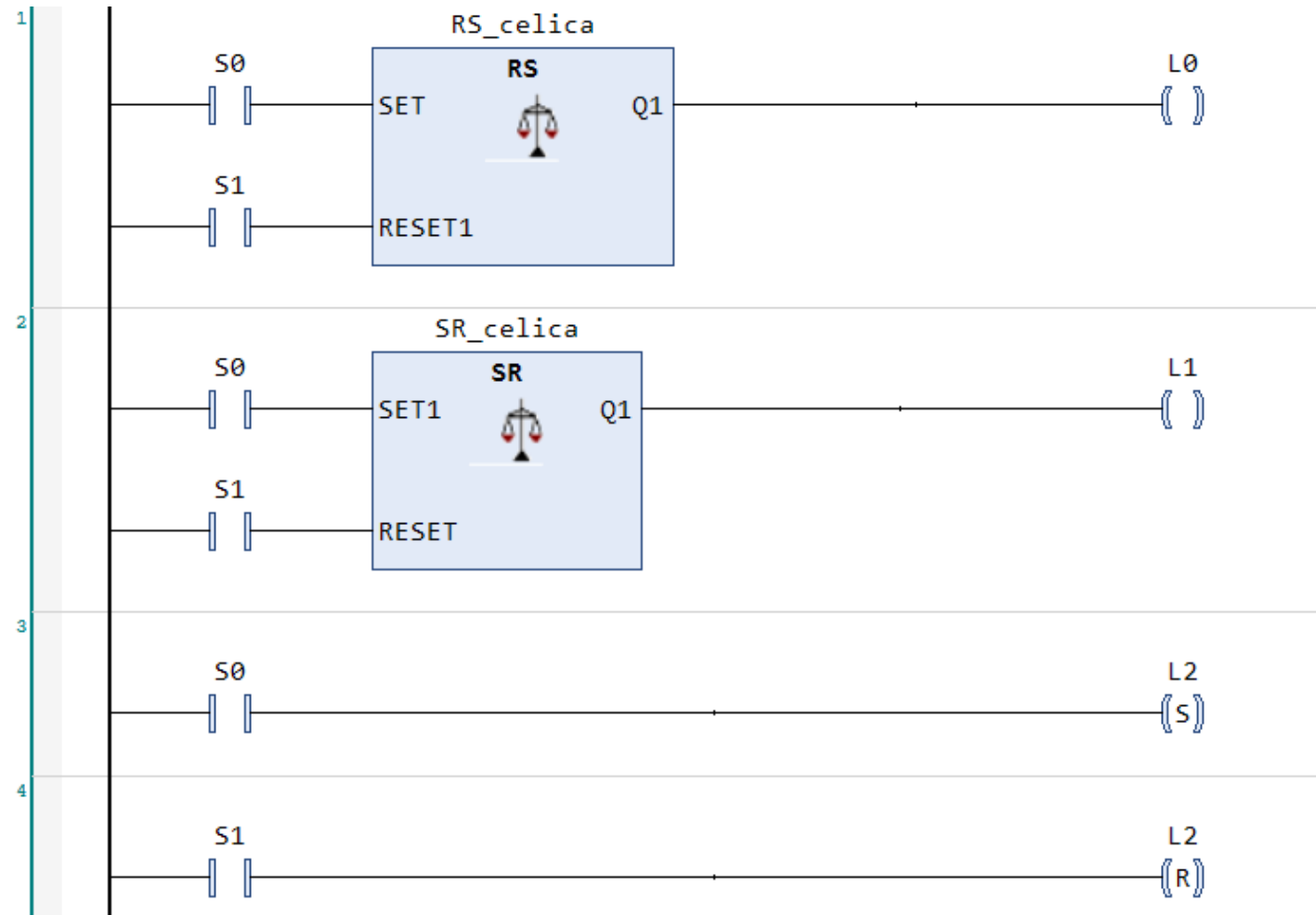
- Pozitivna: FALSE → TRUE, zaznamo jo s funk. blokom R_TRIG
- Negativna: TRUE → FALSE, zaznamo jo z F_TRIG



Lestvični diagrami: pomnilna celica

Pomnilni celici RS (RESET ima prednost pred SET) in SR (SET ima prednost)

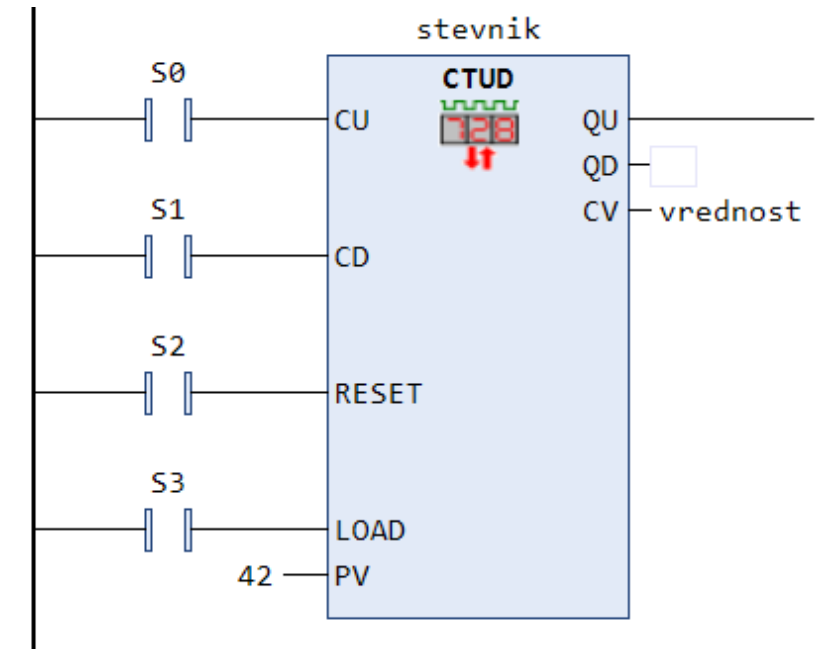
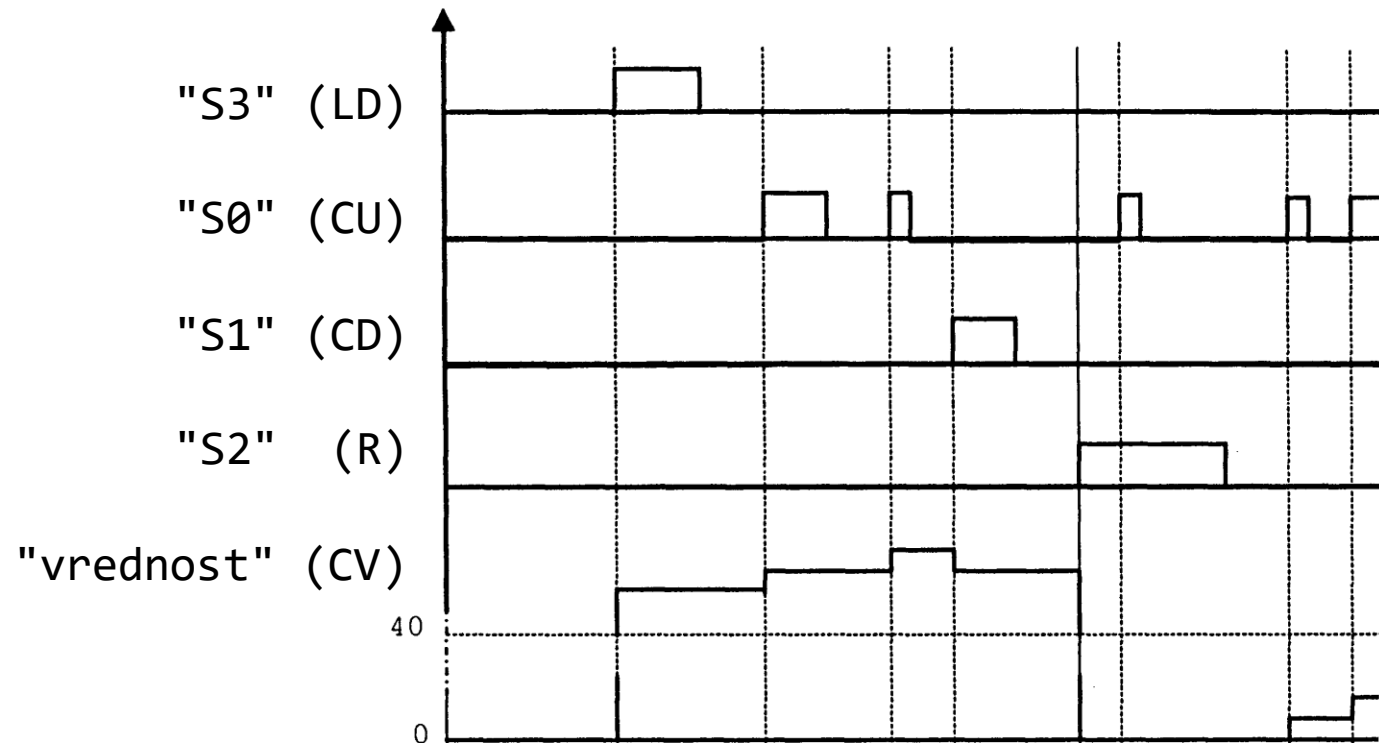
- Način s funkcijskim blokom (klina 1 in 2)



- Način s tuljavami (klina 3 in 4)

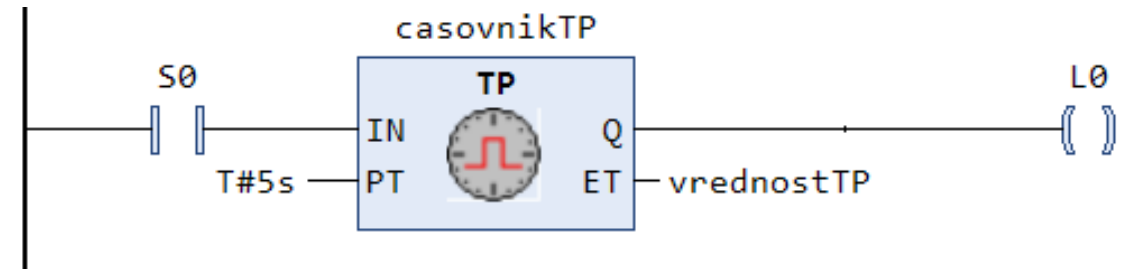
Lestvični diagrami: števniki

Časovni potek za števnik, ki šteje navzgor in navzdol.

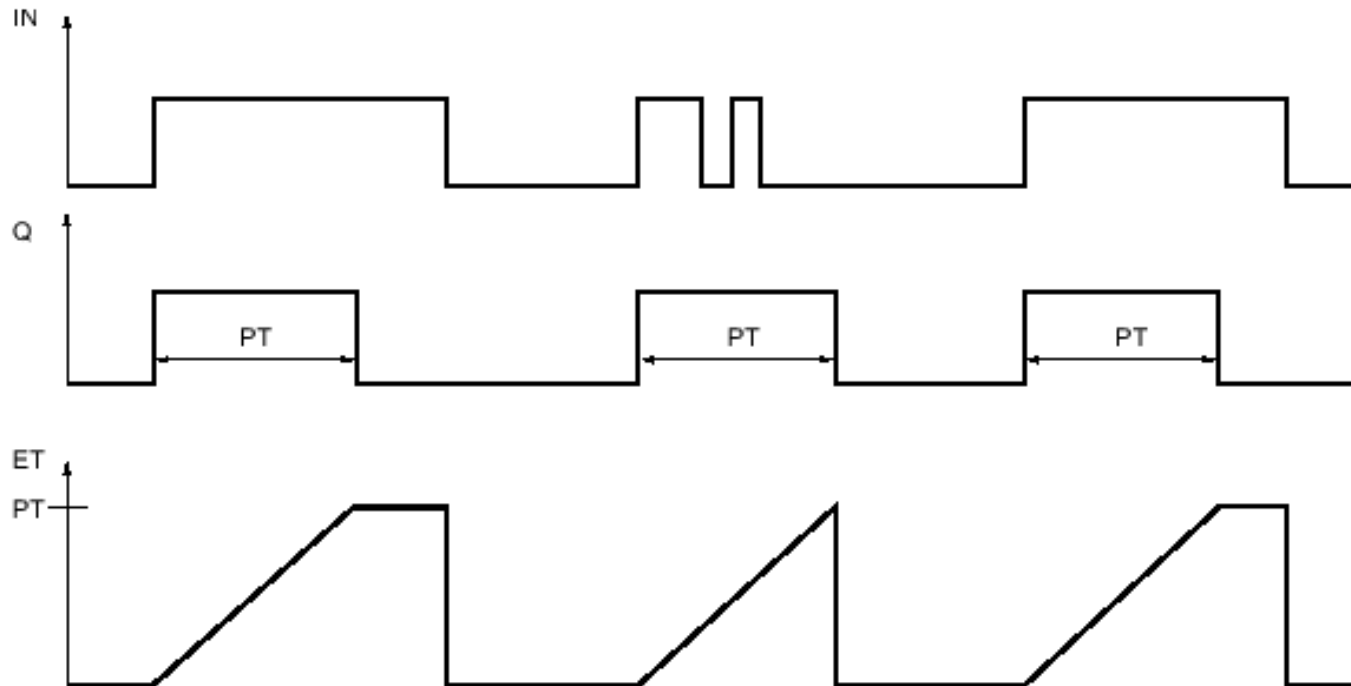


Lestviční diagrami: časovník TP

Generator pulza (timer pulse)

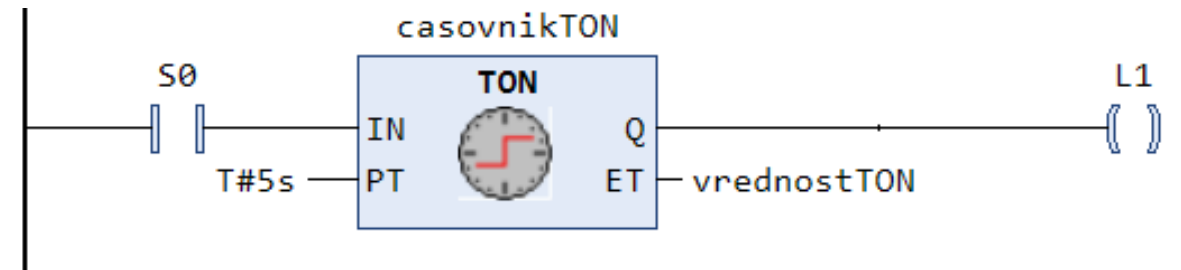


Časovní potek

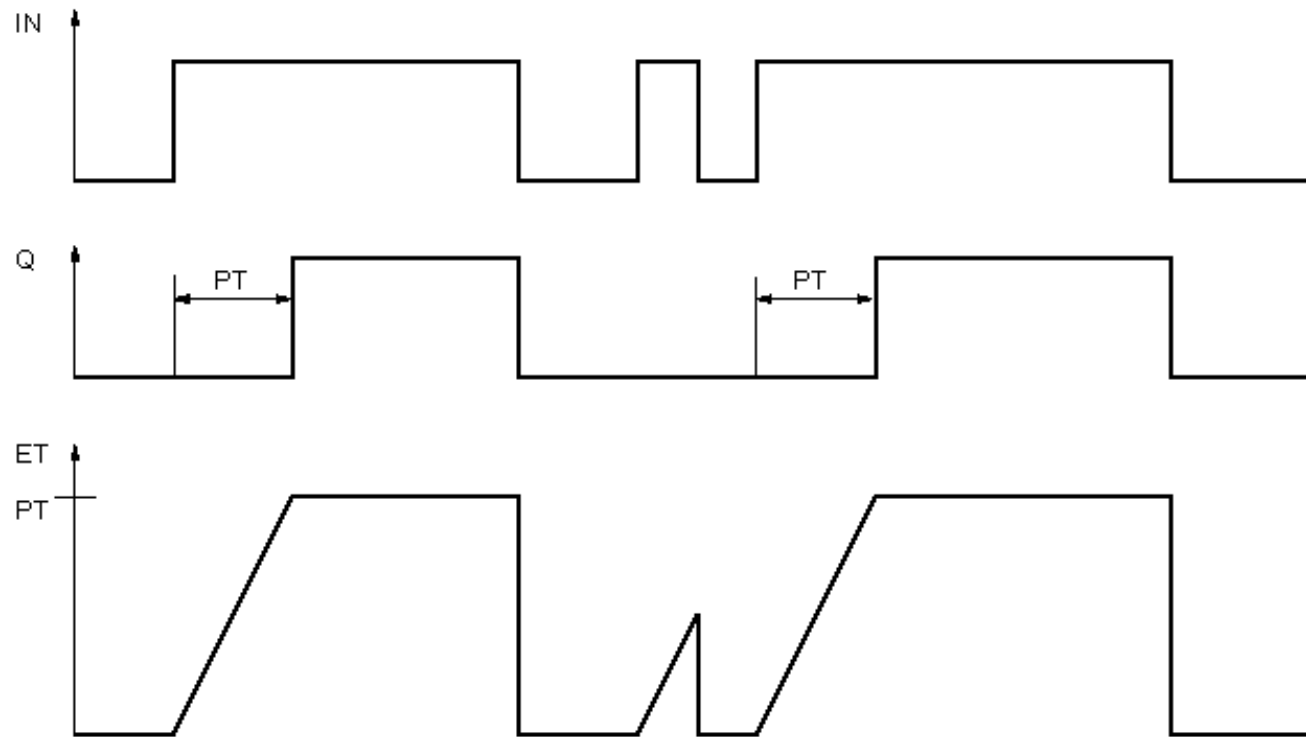


Lestvični diagrami: časovnik TON

Zakasnitev vklopa (timer on-delay)



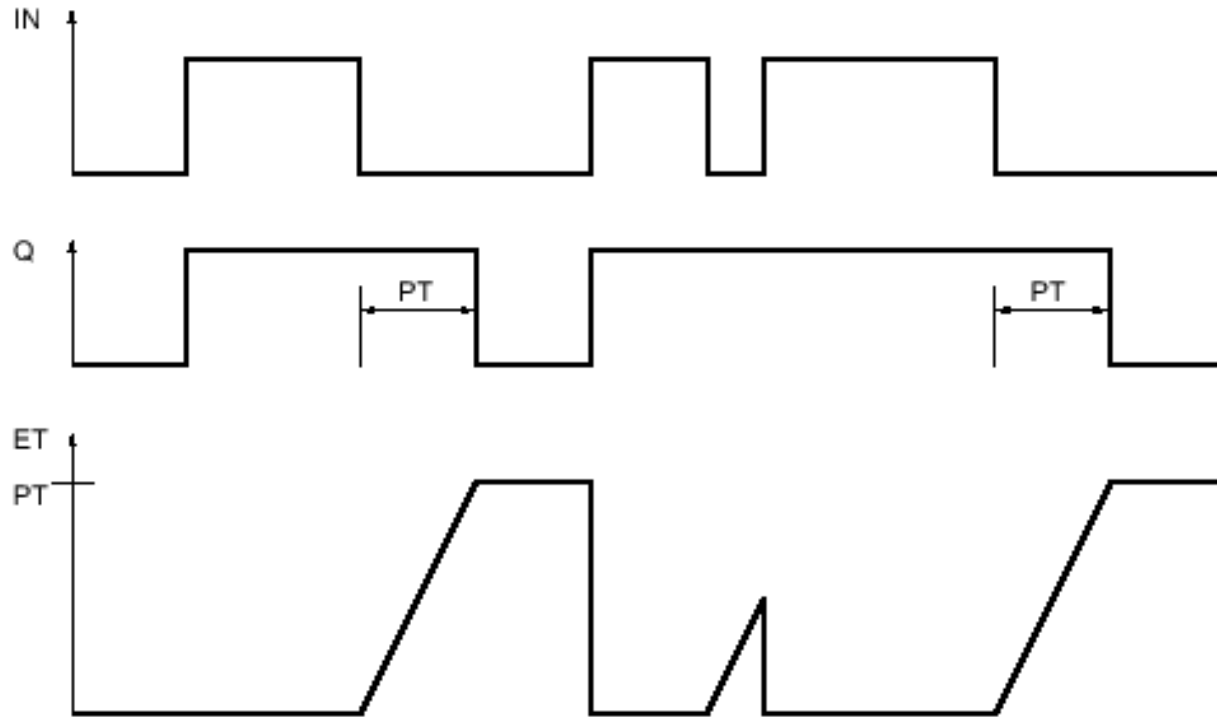
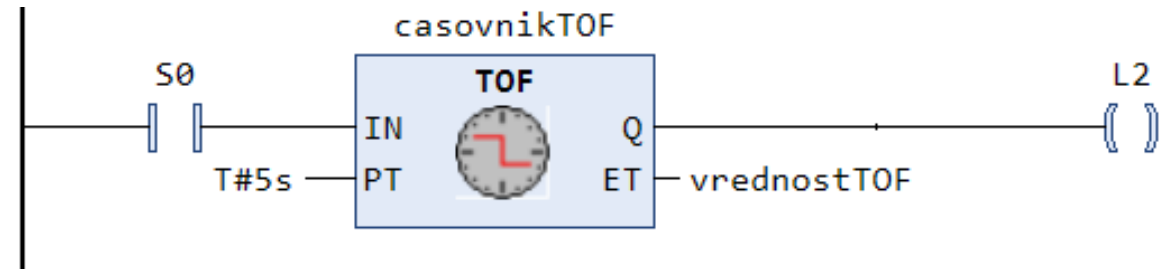
Časovni potek



Lestvični diagrami: časovnik TOF

Zakasnitev izklopa (timer off-delay)

Časovni potek



Lestvični diagrami: časovniki

Primer: blokada prehitre menjave smeri vrtenja motorja

