

UNIVERZA V NOVI GORICI
POSLOVNO-TEHNIŠKA FAKULTETA

PREIZKUŠEVALNIK FUNKCIJ STROJEV

DIPLOMSKO DELO

Boštjan Gojkovič

Mentor: doc. dr. Maja Bračič Lotrič

Nova Gorica, 2007

ZAHVALA

Iskreno se zahvaljujem mentorici diplomske naloge doc. dr. Maji Bračič Lotrič, ki je bila vedno dostopna za vsa vprašanja in mi bila vedno pripravljena pomagati pri izdelavi diplomske naloge, mentorju praktičnega usposabljanja prof. dr. Jušu Kocijanu, ki me je vodil skozi praktično usposabljanje, podjetju VeinEN, s. p., Nova Gorica, ki mi je omogočilo opravljanje diplomske naloge, očetu in stricu, ki sta mi posredovala svoje bogate praktične izkušnje, ter mami in ostalim članom družine, ki so mi ves čas stali ob strani.

POVZETEK

Diplomska naloga zajema razvoj in realizacijo naprave, ki naj bi že med izdelavo novih strojev odkrivala morebitne nepravilnosti v delovanju posameznih mehanskih sklopov. Poimenovali smo jo preizkuševalnik strojev. Preizkuševalnik omogoča testiranje raznih funkcij, kot so vklapljanje ventilov, cilindrov, kontaktorjev, žarnic in vseh ostalih naprav, ki jih je mogoče krmiliti s pomočjo digitalnih in analognih izhodov programskega logičnega krmilnika. Na ta način je mogoče vse kritične in najobčutljivejše mehanizme stroja preizkusiti še pred vgradnjo v kompleksne stroje, pred električnim ožičenjem in namestitvijo ustreznega programa v krmilnik stroja. V diplomski nalogi je predstavljen celoten postopek izdelave od ideje, izbire strojne opreme, izdelave naprave pa do izdelave programske opreme in uporabniškega vmesnika.

ABSTRACT

The diploma project involves development and realization of a control device which detects errors and irregularities in activities of mechanical components in the phase of a machine construction. The control device enables testing of different components, such as valves, contactors, lamps, which can be controlled via digital or analog outputs of a programmable logical controller. In this way, all critical and most delicate mechanisms can be tested prior to electrical wiring and installation of a proper program into the machine »programmable logic controller«. In the thesis, the complete development procedure is given, from the basic idea, hardware selection, PLC programming and user interface development, is given.

OSNOVNI IZRAZI IN OKRAJŠAVE

PLK ... programski logični krmilnik

CPE ... centralno procesna enota

ON... vključeno

OFF... izključeno

Analogni signal ... zvezen (neprekinjen) signal

Digitalni signal ... digitalni signal je opisan z dvema stanjema, to je z logičnim stanjem 1 in z logičnim stanjem 0

I – vhodi ... vhodi PLK-ja

O – izhodi ... izhodi PLK-ja

I/O – vhodno/izhodni sistem

A/D – analogno-digitalni pretvornik ... pretvori analogni signal v digitalnega

D/A – digitalno-analogni pretvornik ... pretvori digitalni signal v analognega

Ventil ... naprava, ki regulira pretok fluida

Cilinder ... mehanski aktuator, ki se uporablja za ustvarjanje potisne (vlečne) sile

PT100 ... sonda za merjenje temperature

Linearna merilna sonda ... sonda za merjenje dolžine (globine)

Preklopnik ... priprava za vključevanje in izključevanje tokokroga ali za preklapljanje med različnimi električnimi tokokrogi

Prikazovalnik ... naprava za vidno prikazovanje strani za vnašanje podatkov ali za izpisovanje sporočil

LAD ... Ladder Logic = Lestvični program

FBD ... Function Block Diagram = Funkcijski blokovni program

SCL ... Structured Control Language = Strukturno krmilni program

STL ... Statement List = Seznam ukazov

KLJUČNE BESEDE

PLK, merilne sonde, krmiljenje ventilov, krmiljenje cilindrov, prikazovalnik, preizkušanje mehanskih sklopov

KAZALO

1.	UVOD.....	9
2.	MEHANSKI SKLOPI.....	11
2.1.	Ventilski otok.....	11
2.2.	Cilinder.....	12
2.3.	Merilne sonde.....	14
2.3.1.	Linearna merilna sonda.....	14
2.3.2.	Temperaturna merilna sonda.....	14
3.	STROJNA OPREMA.....	16
3.1.	Programski logični krmilnik (PLK).....	17
3.2.	Idejna zgradba programskega logičnega krmilnika PLK.....	17
3.3.	Način delovanja PLK-ja.....	18
3.4.	Digitalni vhodno/izhodni sistemi.....	20
3.4.1.	Digitalni vhodi.....	21
3.4.2.	Digitalni izhodi.....	22
3.5.	Analogni vhodno/izhodni sistemi.....	24
3.5.1.	Vhodni analogni signali.....	24
3.5.2.	Analogni izhodni signali.....	27
3.6.	Specifikacija uporabljene strojne opreme.....	29
3.6.1.	Programski logični krmilnik Siemens CPU 313C.....	29
3.6.2.	Prikazovalnik Siemens Simatic OP7.....	30
4.	PROGRAMSKA OPREMA.....	32
4.1.	Simatic Step 7.....	32
4.1.1.	Ladder logic LAD (Lestvični program).....	34
4.1.2.	Function Block Diagram FBD (Funkcijsko blokovni program).....	35
4.1.3.	Statement List STL (Seznam ukazov).....	37
4.1.4.	Structure Control Language SCL (Strukturno programski jezik).....	38
4.2.	Protool.....	43
5.	IZDELAVA PREIZKUŠEVALNIKA.....	44
5.1.	Izvedba osnovnih funkcij.....	46
5.1.1.	Izvedba poljubnega izbiranja vhodov.....	47
5.1.2.	Izvedba nastavljanja časovnih zakasnitev.....	48
5.1.3.	Izvedba izpisovanja napak na prikazovalniku.....	48
5.1.4.	Izvedba poljubnega določanja izhodov.....	49
5.1.5.	Izvedba merjenja z merilno sondo.....	50
5.1.6.	Izvedba merjenja temperature s temperaturno sondo.....	51
5.2.	Izbira načina delovanja.....	51
5.3.	Vnos parametrov.....	52
5.4.	Načini delovanja.....	56
5.4.1.	Samodejni način delovanja.....	56
5.4.2.	Koračni način delovanja.....	58
5.4.3.	Ročni način delovanja.....	59
6.	PREIZKUS DELOVANJA PREIZKUŠEVALNIKA.....	61
6.1.	Preverjanje izpisovanja načina delovanja.....	63
6.2.	Preverjanje delovanja samodejnega načina delovanja.....	64
6.3.	Preverjanje delovanja koračnega načina.....	65
6.4.	Preverjanje delovanja ročnega načina.....	65
6.5.	Preverjanje delovanja časovnih zakasnitev.....	65
6.6.	Preverjanje izpisovanja mehanske napake.....	66

6.7.	Preverjanje izpisovanja napake pri vnosu parametrov	67
6.8.	Preverjanje točnosti ponavljanja gibov mehanskih sklopov z merilno sondo 67	
6.9.	Preverjanje delovanja temperaturne sonde.....	68
7.	ZAKLJUČEK	69
8.	LITERATURA	71

KAZALO SLIK

Slika 1:	Namen preizkuševalnika	9
Slika 2:	Ventilski otok	12
Slika 3:	Cilinder	12
Slika 4:	Delovni položaj cilindra krmiljen z dvostranskim ventilom	13
Slika 5:	Vrnitev v začetni položaj cilindra, krmiljen z dvostranskim ventilom	13
Slika 6:	Linearna merilna sonda	14
Slika 7:	Temperaturna sonda PT100.....	15
Slika 8:	PLK-aplikacija.....	16
Slika 9:	Sestava PLK-ja	18
Slika 10:	Glavni deli CPE.....	18
Slika 11:	Delovanje CPE	19
Slika 12:	I/O-vmesnik	20
Slika 13:	Grafični prikaz prenosa vhodnega bita.....	21
Slika 14:	Blokovni prenos ali ukaz za nalaganje podatkov	22
Slika 15:	Grafični prikaz prenosa izhodnega bita	23
Slika 16:	Prenos podatkov v izhodno enoto.....	24
Slika 17:	Analogna vhodna enota	25
Slika 18:	Vzorčenje analognega signala	25
Slika 19:	Prikaz korakov pri pretvarjanju analognega signala v binarno obliko	27
Slika 20:	Krmiljenje dušilnega ventila.....	27
Slika 21:	Digitalno analogna pretvorba signala.....	28
Slika 22:	Koraki pri pretvarjanju diskretne vrednosti v analogni signal	29
Slika 23:	Siemens CPU 313C	30
Slika 24:	Siemens Simatic OP7	30
Slika 25:	Primer prvega lestvičnega programa	34
Slika 26:	Primer drugega lestvičnega programa	35
Slika 27:	Povezava vhodne spremenljivke in vhod bloka	35
Slika 28:	Povezava izhoda bloka in vhod naslednjega bloka	36
Slika 29:	Povezava izhoda bloka in izhodne spremenljivke.....	36
Slika 30:	Izgled programa napisanega v FBD in STL	37
Slika 31:	Diagram poteka repeat ukaza	40
Slika 32:	Diagram poteka stavka IF (pogojni stavek).....	41
Slika 33:	Diagram poteka stavka CASE	42
Slika 34:	Preizkuševalnik.....	44
Slika 35:	Delovanje preizkuševalnika.....	45
Slika 36:	Programska koda aplikacije.....	46
Slika 37:	Primerjava vhodov na PLK-ju z izbranimi vhodi na prikazovalniku.....	47
Slika 38:	Prikaz časovne zakasnitve	48
Slika 39:	Prvo alarmno sporočilo.....	49
Slika 40:	Drugo alarmno sporočilo	49
Slika 41:	Izbiranje izhodov	50

Slika 42: Graf funkcije diskretne spremenljivke	50
Slika 43: Zagon preizkuševalnika.....	51
Slika 44: Izbor načina delovanja.....	52
Slika 45: Izpis izbranega načina delovanja.....	53
Slika 46: Izpis na zaslonu pri izbiri koraka.....	53
Slika 47: Izbira vhodov v prvem koraku	54
Slika 48: Izbira izhodov v prvem koraku.....	54
Slika 49: Izbira meritve v koraku	55
Slika 50: Nastavljanje parametrov za merjenje dolžine.....	55
Slika 51: Nastavitev parametrov za merjene temperature	56
Slika 52: Samodejni način delovanja.....	57
Slika 53: Koračni način delovanja	59
Slika 54: Ročni način delovanja	60
Slika 55: Testiranje preizkuševalnika.....	61
Slika 56: Položaji cilindrov v posameznem koraku.....	62

KAZALO TABEL

Tabela 1: Pomen posameznih vhodov	62
Tabela 2: Pomen posameznih izhodov	63
Tabela 3: Rezultati kontrole izpisovanja načina delovanja	63
Tabela 4: Rezultati kontrole samodejnega načina delovanja.....	64
Tabela 5: Rezultati kontrole delovanja tipke NAPREJ	65
Tabela 6: Rezultati kontrole delovanja ročnega načina	65
Tabela 7: Rezultati kontrole časovnih zakasnitev.....	66
Tabela 8: Rezultati kontrole izpisovanja mehanske napake	66
Tabela 9: Rezultati kontrole izpisovanja napake pri vnosu parametrov	67
Tabela 10: Rezultati merjenja ponavljivosti giba cilindra z merilno sondo	68
Tabela 11: Rezultati kontrole delovanje temperature sonde.....	68

1. UVOD

Pogosto se zgodi, da je treba nove stroje ali cela postrojenja popravljati že pri njihovem prvem zagonu v proizvodnji ali pri kupcu. Nemalokrat gre za visoke stroške, za večje zamude in posledično izgube. Ogroženo je tudi dobro ime proizvajalca. Da bi se izognili takim nevšečnostim, je nastala zamisel za izdelavo preizkuševalnika, s katerim bi že v proizvodnji med izdelavo posameznih sklopov stroja preverili njihovo brezhibno delovanje. Tako bi se lahko odpravile vse nepravilnosti, še preden bi stroje predali končnemu kupcu v uporabo, s čimer bi se izognili velikanskim stroškom popravil strojev po postavitvi v proizvodne linije.

S preizkuševalnikom bi že med izdelavo novih strojev odkrivali nevšečnosti in nepravilnosti v delovanju posameznih mehanskih sklopov stroja v izgradnji. Tako bi vse kritične in najobčutljivejše mehanizme, ki bi zaradi nepravilnega delovanja naredili škodo, preizkušali že pred električnim ožičenjem in namestitvijo programa v programski logični krmilnik stroja (slika 1).



Slika 1: Namen preizkuševalnika

Gre za specifično napravo, zato na trgu verjetno ne bi uspeli najti izdelka, ki bi ustrezal potrebam naročnika, ki so se izoblikovale skozi dolgoletne izkušnje pri izdelavi in preizkušanju strojev. Zato smo se v podjetju odločili za razvoj in realizacijo preizkuševalnika.

V diplomski nalogi je predstavljen razvoj, izdelava in preizkus naprave, ki temelji na naslednjih zahtevah naročnika:

- Preizkuševalnik naj deluje v samodejnem, koračnem in ročnem načinu delovanja. Možnost naj ima za preklapljanje med naštetimi načini in za potrditev izbranega načina delovanja. Izpis delovanja naj bo viden na prikazovalniku.

- Operater naj ima možnost uporabljati katerikoli vhod ali izhod (od šestnajstih možnih) za povezavo stroja s preizkuševalnikom.
- Za preprečevanje trkov in lomov naj ima preizkuševalnik tudi možnost nastavljanja časovnih zakasnitev v posameznih korakih. Torej uporaba časovnih zakasnitev zakasni delovanje korakov in tako preprečuje, da bi posamezni mehanski sklopi trčili med seboj in povzročili škodo.
- Program naj ima tudi možnost kontrole fizikalnih količin (temperatura, dolžina) z ustreznimi merilnimi sondami.
- Na prikazovalniku preizkuševalnika naj se izpisujejo alarmna sporočila ob nepravilnem ali pomanjkljivem vnosu parametrov in nepravilnem delovanju naprave, ki jo testiramo.

V diplomski nalogi so najprej opisani mehanski sklopi, njihovi gradniki in njihovo delovanje. V nadaljevanju je opisana vsa uporabljena strojna oprema za izdelavo preizkuševalnika, postopek izbire strojne opreme, zgradba programskega logičnega krmilnika in njegovo delovanje. V četrtem poglavju je podrobno opisan postopek izbire programske opreme in izdelava aplikativnega programa. Temu sledi izdelava preizkuševalnika na osnovi zahtev naročnika. Preizkus njegovega delovanja je predstavljen v šestem poglavju.

2. MEHANSKI SKLOPI

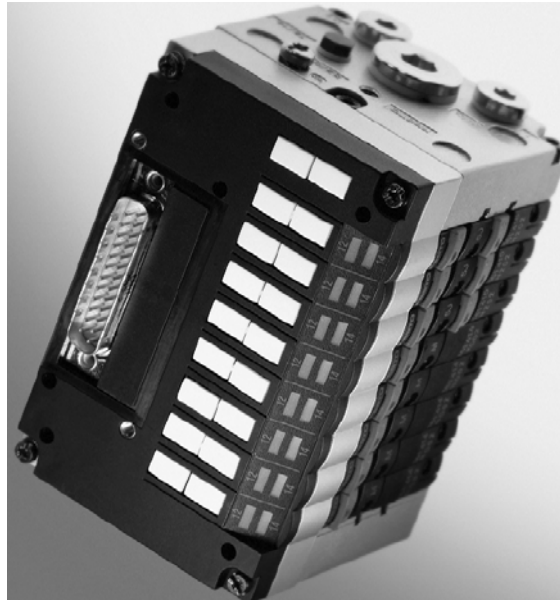
Diplomska naloga opisuje razvoj preizkuševalnika, t. j. naprave, prilagojene za preizkušanje delovanja mehanskih sklopov. Mehanski sklopi, ki so sestavljeni iz ventilskih otokov, pnevmatskih ali hidravličnih cilindrov, merilnih sond, pozicionirnih naprav, so gradniki raznovrstnih kompleksnih strojev. Za delovanje kompleksnih strojev, ki so sestavljeni iz večjega števila mehanskih sklopov, je pomembno, da predhodno preizkusimo vsakega od teh sklopov. Tako je pri zagonu stroja zagotovljeno, da osnovni sestavni deli delujejo brezhibno. S tem skrajšamo čas, potreben za zagon stroja, in olajšamo iskanje napak.

2.1. Ventilski otok

Ventilski otok (slika 2) zajema sklop razdelilnih ventilov (razdelilnikov), ki so po izvedbi lahko enostranski, dvostranski in dvostranski z ničelnim položajem, ter vse pnevmatske dovodne, odvodne in delovne cevi. Dovodne cevi so osrednji element ventilskih odprtih, ki dovoljujejo direkten pretok zraka skozi ventilske odprtine in doseganje maksimalnega pretoka zraka. Vsi ventili imajo pnevmatski krmilnik za optimizacijo delovnega učinka. Razdelilno funkcijo opravlja batno vretenčni sistem s tesnilnimi špranjami, kar zagotavlja njegovo primernost za široko paleto aplikacij in dolgo življenjsko dobo. Pnevmske komponente in njihove funkcije so standardizirane za vse modele pogonskih naprav (Priročnik Festo, 2005). Enostranski razdelilni ventil ima eno tuljavo in potrebuje za svoje delovanje stalno napetost 24 V, ki prekrmi ventil v delovni položaj. Ko napetosti na tuljavi ni, vzmet vrne ventil v njegov izhodiščni položaj.

Dvostranski razdelilni ventil ima dve tuljavi in potrebuje za svoje delovanje trenutno napetost 24 V, ki postavi ventil v ustrezen delovni položaj (glede na tuljavo, ki je dobila napetostni impulz). V tem položaju ostane, dokler nasprotna tuljava ne prejme impulza za prekrmljenje ventila v drugi položaj.

Dvostranski razdelilni ventil z ničelnim položajem ima tudi dve tuljavi in za svoje delovanje potrebuje na tuljavi stalno napetost 24 V. Ko napetosti ni ne na eni ne na drugi tuljavi, vzmet vrne ventil v njegov ničelni položaj.

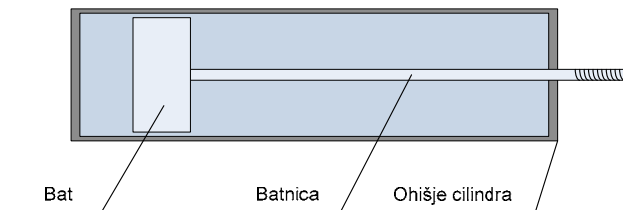


Slika 2: Ventilski otok

2.2. Cilinder

Pnevmatski cilindri (slika 3) so mehanske naprave, uporabljajo pa se za prenašanje moči, ki jo daje tlačna energija tekočine; največkrat gre za stisnjen zrak. Običajen pnevmatski cilinder je sestavljen iz bata, batnice in ohišja cilindra.

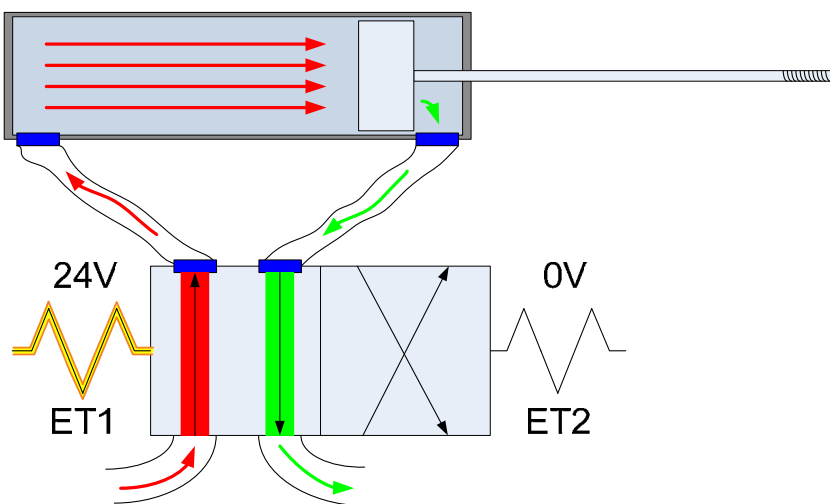
Poznamo enostransko in dvostransko delujoče cilindre. Pri enostranskih cilindrih tlačna sila fluida potisne batnico ven (odprt položaj), vzmet pa vrne batnico v začetno stanje (zaprt položaj cilindra). Pri dvosmerno delujočih batih se za gibanje batnice ven in noter uporablja tlačna sila fluida.



Slika 3: Cilinder

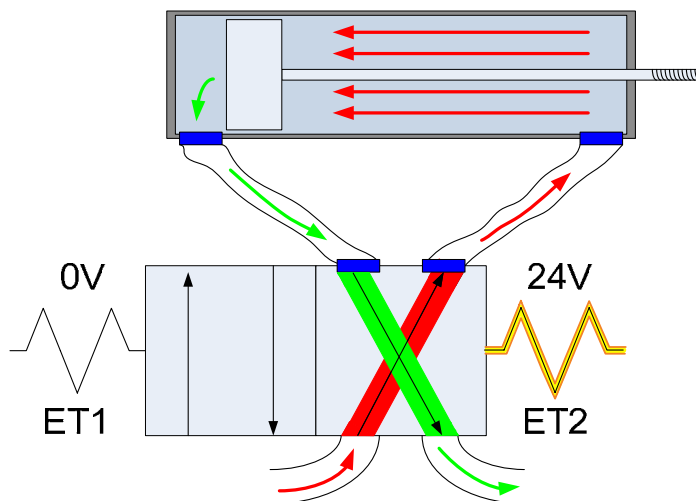
Delovanje mehanskega sklopa je shematično prikazano na slikah 3 in 4. Pnevmatski cilinder se nahaja v začetnem (zaprtem) položaju, ko je napetost na elektro tuljavah ET1 in ET2 0 V.

Ko je napetost ventila na ET1 enaka 24 V (slika 4), se vklopi ventil, ki odpre dovod zraka v cilindar – tlačna sila tekočine povzroči gibanje bata naprej (batnica zleze ven iz cilindra).



Slika 4: Delovni položaj cilindra krmiljen z dvostranskim ventilom

Položaj ventila in cilindra ostaneta nespremenjena, tudi ko je napetost na ET1 ponovno 0 V. Ventil se vrne v nasprotni položaj šele, ko je napetost na ET2 24 V (slika 5). Tedaj spusti ventil zrak skozi drugi priključek cilindra, ki potisne bat nazaj v prvotni (zaprt) položaj (batnica se potopi v cilindar).



Slika 5: Vrnitev v začetni položaj cilindra, krmiljen z dvostranskim ventilom

2.3. Merilne sonde

Pri preizkušanju mehanskih sklopov uporabljamo merilne sonde za merjenje fizikalnih količin, kot so temperatura, dolžina, sila ipd. Merilno sondo uporabljamo za merjenje odstopanj od delovnih položajev premičnih delov mehanskih sklopov, temperaturno merilno sondo pa za kontrolo temperature fluida.

2.3.1. Linearna merilna sonda

Linearno merilno sondo (slika 6) se uporablja za merjenje dolžine in globine. Upornostna sonda deluje na principu drsnega upornika. Upornost drsnika se izračuna po enačbi

$$R = \zeta \frac{l}{S},$$

kjer pomeni

R – upornost žice (vodnika)

ζ – specifično upornost žice v Ωm

l – dolžino žice

S – presek žice



Slika 6: Linearna merilna sonda

V začetni legi ima sonda upornost 0Ω in napetost 0 V . Takrat kaže sonda dolžino $0,0 \text{ mm}$. Sonda kaže maksimalno dolžino pri maksimalni upornosti in napetosti 10 V (Contelec, 2006).

2.3.2. Temperaturna merilna sonda

Upornost kovin se spreminja s temperaturo. Za manjša temperaturna območja je relacija med temperaturo in upornostjo skoraj linearna in jo lahko opišemo z relacijo

$$R_D = R_R (1 + \alpha [T_D - T_R]).$$

(Pico Technology, 2006)

T_D – dejanska temperatura

R_D – dejanska upornost

R_R – referenčna upornost pri temperaturi 20 °C

α – temperaturni koeficient za temperaturno sondo PT100 znaša $3.9083 \cdot 10^{-3} \text{ K}^{-1}$

T_R – referenčna temperatura je enaka 293 K (20 °C)

Relacijo med upornostjo in temperaturo določa mednarodni temperaturni standard ITS-90. Merilna sonda PT100, prikazana na sliki 7, ima pri temperaturi 0 °C upornost 100 Ω , pri 100 °C pa 138.4 Ω . Območje linearnosti temperature sonde PT100 se giblje od 0 °C do 850 °C.

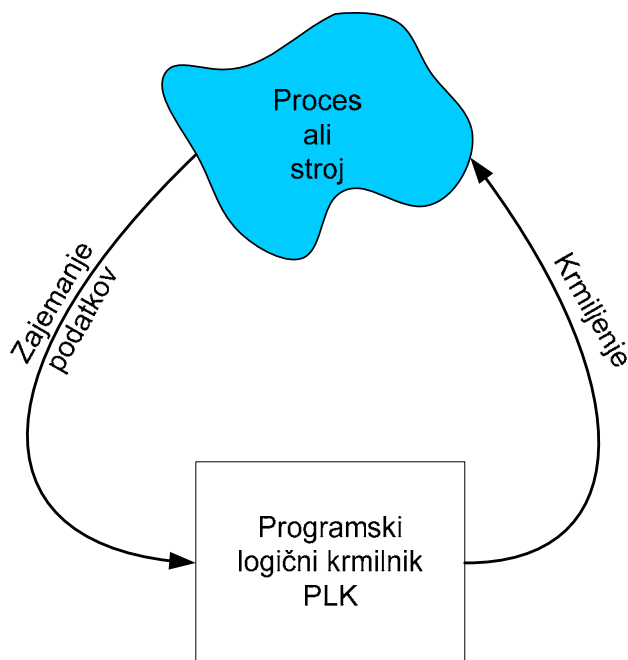


Slika 7: Temperaturna sonda PT100

3. STROJNA OPREMA

Za izdelavo preizkuševalnika je bilo najprej potrebno izbrati primerno opremo. Uvodoma podane zahteve naročnika navajajo potrebo po samostojnem delovanju naprave pod vplivom določenih vhodnih signalov in parametrov, vključujejo pa tudi prikaz stanja naprave in zahtevo za nastavljanje parametrov glede na postopek testno proceduro. Za realizacijo teh zahtev je najbolj smiselna uporaba programabilnega logičnega krmilnika (PLK), ki nam omogoča visoko fleksibilnost aplikacije v kombinaciji s prikazovalnikom, ki služi za prijazno in pregledno komunikacijo človek-stroj.

V njih se shranjujejo navodila za sekvenciranje (zaporedja ukazov ali kaj podobnega ter za časovna usklajevanja in merjenja), izvajajo matematične operacije, obdelujejo podatki in na njih potekajo komunikacije, s čimer se krmilijo proizvodni stroji in procesi. Na sliki 8 je predstavljena shema PLK-aplikacije.



Slika 8: PLK-aplikacija

Kar nekaj definicij obstaja za PLK. Predstavljamo si jih lahko kot industrijske računalnike specialnih konstrukcij s centralno enoto (sam PLK) in vhodno/izhodnimi enotami (I/O-vmesnik), preko katerih se povezujejo s senzorji in aktuatorji posameznega procesa (L. A. Bryan, E. A. Bryan, 1997, str. 4).

3.1. Programski logični krmilnik (PLK)

Programski logični krmilniki, na kratko PLK, so elektronski (računalniški) elementi, ki za opravljanje krmilnih funkcij uporabljajo integrirna vezja namesto elektromehanskih. Prva merila za PLK so napisana že v letu 1968. Izdelalo jih je podjetje General Motors. Njihov glavni cilj je bil zmanjšati visoke cene, ki so spremljale toge, relejske krmilne sisteme. Specifikacije so zahtevale stabilen sistem z računalniško fleksibilnostjo, ki bo funkcionalen v industrijskem okolju, enostaven za programiranje in vzdrževanje in bo uporaben tudi v naslednjih projektih. Takšen krmilni sistem bi zmanjšal zastoje strojev in zagotovil nadaljnji razvoj. Prve specifikacije so vsebovale naslednje zahteve:

- Novi krmilni sistem naj bo cenovno ugodnejši od relejnih sistemov.
- Sistem naj podpira industrijsko okolje.
- Vhodno-izhodni vmesniki naj bodo enostavno zamenljivi.
- Zgradba krmilnika naj bo modulna, da se zagotovi enostavna zamenjava ali popravilo podsklopov.
- Krmilni sistem naj prenaša podatke v centralni sistem.
- Sistem naj bo uporaben tudi po odsluženju projekta, za katerega je bil narejen.
- Programiranje krmilnika naj bo enostavno in razumljivo tehničnemu osebju.

(General Motors, 2007)

3.2. Idejna zgradba programskega logičnega krmilnika PLK

Prvi PLK-ji so le zamenjali releje. Njihova poglobljena funkcija je bila izvajanje operacij, ki so jih prej izvajali releji. Te operacije so vključevale vklop/izklop stružnic in vrtalnih strojev ali procesov, ki so zahtevali ponavljajoča se dela, npr. v proizvodnji na tekočih trakovih. Kljub temu so ti zgodnji PLK-ji pomenili veliko izboljšavo v primerjavi z releji. Vgradnja je bila enostavnejša, zavzemali so manj prostora in trošili manj energije. Opremljeni so bili z diagnostičnimi indikatorji, ki so pripomogli k odpravljanju tehničnih motenj, in za razliko od relejev se jih je dalo ponovno uporabiti po izteku projekta. Čeprav so se funkcije PLK-jev, kot so delovna hitrost, tip vmesnih enot in zmogljivost obdelovanja podatkov, v vseh teh letih

izboljševale, pa je v njihovih specifikacijah še vedno prisoten prvotni konstrukcijski cilj – da morajo biti enostavni za uporabo in vzdrževanje.

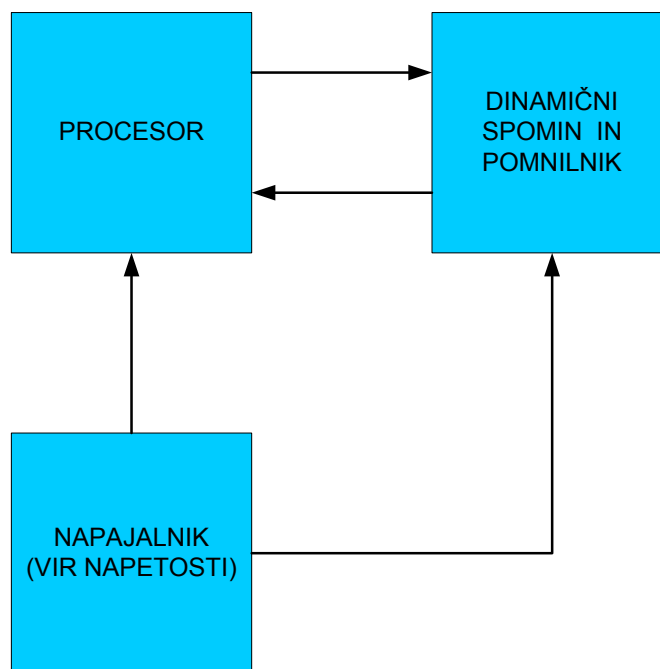
3.3. Način delovanja PLK-ja

Kot prikazuje slika 9 sestavljata PLK dva osnovna dela: centralno procesna enota in vhodno/izhodni enoti.



Slika 9: Sestava PLK-ja

Centralna procesna enota CPE upravlja vse dejavnosti PLK-ja. Sestavljajo jo procesor, dinamični spomin in pomnilnik ter vir napetosti (slika 10).



Slika 10: Glavni deli CPE

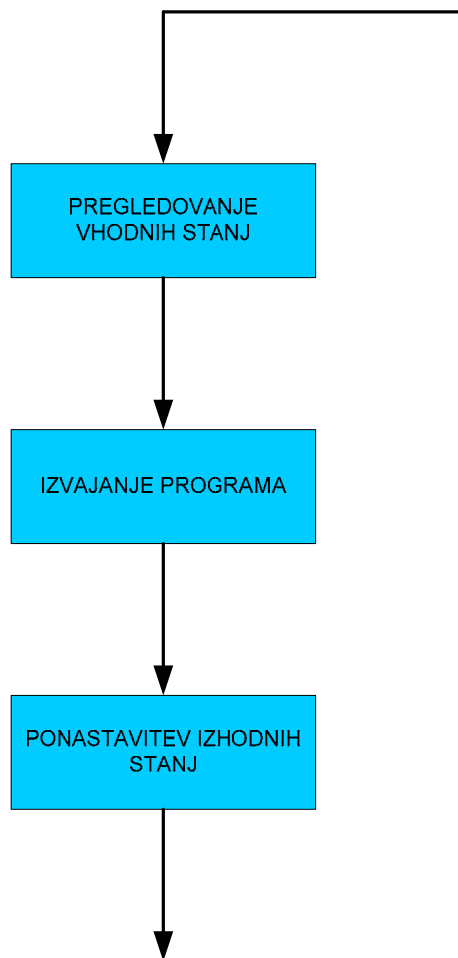
I/O-vmesnik je fizično povezan z obstoječimi priključnimi napravami, ki jih običajno dobimo v strojih ali uporabljamo v proizvodni krmilni tehniki. Te priključne enote (priključki) so lahko diskretne ali analogne naprave, kot so stikala, tlačni pretvorniki,

tipkala, koračni motorji, elektromagneti itd. I/O-vmesnik zagotavlja povezavo CPE s oskrbovalci informacij (vhodi) in s krmilno napravo (izhodi).

CPE izvaja tri operacije:

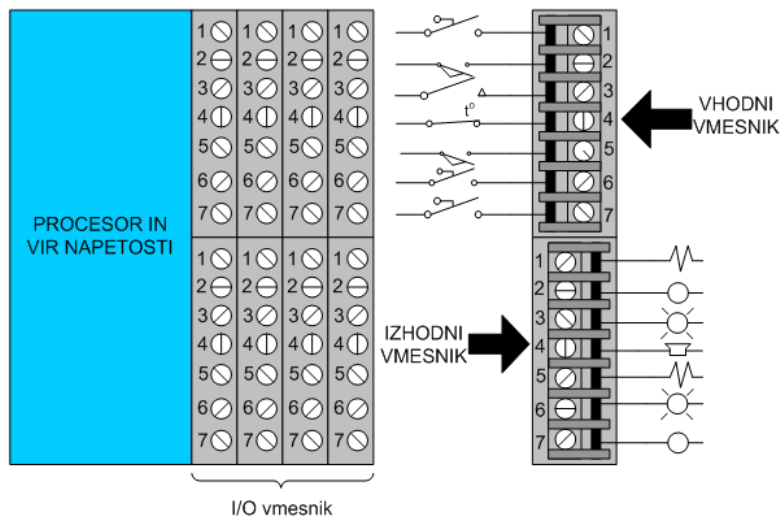
- bere (ali sprejema) vhodne podatke od priključenih naprav preko vhodnih vmesnikov,
- izvaja program. shranjen v pomnilniku, in
- sporoča (ali obnavlja sporočila) izhodnim napravam preko izhodnih vmesnikov.

Ta način delovanja je poznan kot »skeniranje«. Funkcijsko shemo delovanja CPE prikazuje slika 11.



Slika 11: Delovanje CPE

Vhodno/izhodni sistem je vmesnik, preko katerega povezujemo zunanje (periferne) naprave s PLK-jem (slika 12). Glavni namen vmesnika je pretvoriti signale, ki so bili sprejeti od zunanjih enot ali poslani vanje, v odgovarjajočo obliko. Senzorje (kot so tipkala, mejna in druga stikala, analogni senzorji ...) priključimo na vhodni vmesnik, krmilne naprave, kot so koračni motorji, elektromagnetni ventili, kontrolne lučke in pozicijski ventili, pa na izhodni vmesnik. Napajalnik zagotavlja potrebno električno napetost za pravilno delovanje vseh delov CPE.



Slika 12: I/O-vmesnik

Krmilni program v pomnilnik krmilnika nameščamo s pomočjo osebnega računalnika (PC) ali proizvajalčevim (mini)programatorjem. Programator je povezan s krmilnikom, ko se vnaša ali preverja krmilni program (L.A. Bryan, E.A. Bryan, 1997).

3.4. Digitalni vhodno/izhodni sistemi

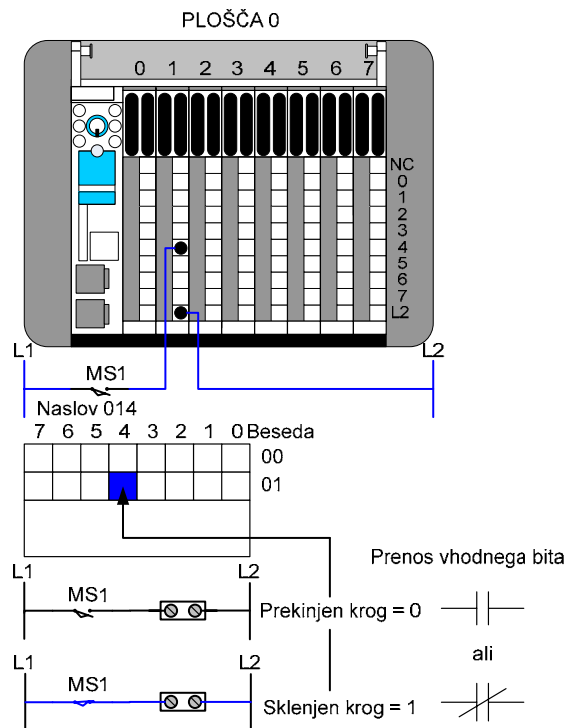
Digitalni vhodno/izhodni (I/O) sistem skrbi za fizično povezavo med CPE in zunanjimi napravami, ki prenašajo in sprejemajo digitalne signale. Digitalni signali so signali, ki imajo le dve stanji 1 (vklopljeno) in 0 (izklopljeno). S pomočjo različnih vmesnih elektronskih vezij in priključnih naprav (senzorji, tipala, stikala) dobi PLK preko digitalne vhodne enote informacijo o stanju naprave, ki se v krmilnem programu ustrezno obdelata in uporabi za krmiljenje naprav. Digitalna izhodna enota omogoča prenos informacij do krmilnih naprav.

3.4.1. Digitalni vhodi

Vhodne digitalne naprave se preko vhodnega digitalnega vmesnika povezuje z vhodno enoto PLK-ja. Digitalni (nezvezni) signali vhodnih digitalnih vmesnikov imajo le dve stanji (npr. ON/OFF, odprto/zaprto, pravilno/napačno itd.). Med vhodne digitalne priključke (zunanje naprave) spadajo prekinjala, nivojski mejniki, mejna stikala, motorna koračna stikala, fotocelice, induktivna stikala, tipkala, relejna stikala, selektorji in krilna stikala.

Za delo z digitalnimi vhodi obstaja več ukazov. Ti ukazi imajo opravka z enim bitom, ko kontrolirajo en vhod, ali z več biti, ko kontrolirajo več vhodov. Ne glede na to, ali ukaz kontrolira en ali več digitalnih vhodov, je informacija vedno ista – vključeno ON ali izključeno OFF.

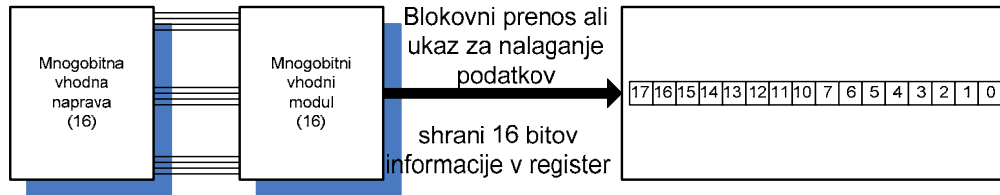
Slika 13 grafično prikazuje poenostavljeno 8-bitno ploščo v vhodni enoti, na katero je priključeno mejno stikalo MS1. Na eno ploščo se lahko poveže 8 vhodov (0-7). Naslove vhoda, pod katerimi jih krmilnik prepozna, določamo s strojno konfiguracijo krmilnika. Uporabljeno konkretno mejno stikalo MS1 nosi naslov 014, s katerim je določeno njegovo priključno mesto: 0 za ploščo, 1 za utor in 4 za priključek.



Slika 13: Grafični prikaz prenosa vhodnega bita

Vhodni vmesnik registrira napetostni signal zunanje naprave in ga pretvori v logični signal 1 ali 0. Logično stanje 1 na vhodni enoti kaže na vključeno ON ali sklenjen krog, logično stanje 0 pa kaže na izključeno OFF ali prekinjen krog.

Pri večbitnih vhodnih modulih, ki sprejemajo večje število vhodov, npr. od krilnih stikal, ki se uporabljajo pri BCD-registrih, se vhodne vrednosti nalagajo v registrski pomnilnik (slika 14) (L. A. Bryan, E. A. Bryan, 1997).



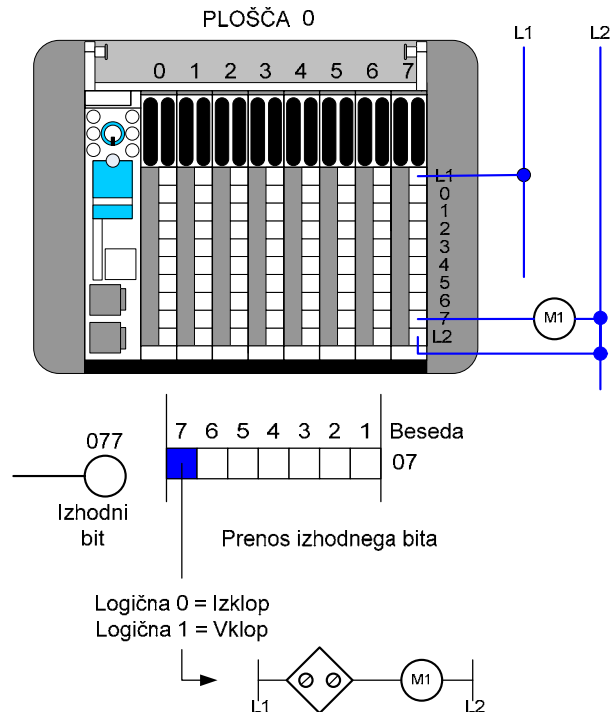
Slika 14: Blokovni prenos ali ukaz za nalaganje podatkov

Vhodni digitalni vmesnik prejema električno napetost in tok od napajalnika, ki oskrbuje ploščo, v kateri je vgrajen. Signali, ki prihajajo iz vhodnih (priključnih) naprav, so lahko različnih vrst in/ali velikosti. Tako obstajajo vhodni digitalni vmesniki, ki se napajajo z različnimi AC- in DC-napetostmi (izmeničnimi in enosmernimi). Standardna območja vhodnih napetosti so 24 V AC/DC, 48 V AC/DC, 120 V AC/DC, 230 V AC/DC, breznapetostni vhod, izolirani vhod in 5 – 50 V DC (izliv, ponor).

3.4.2. Digitalni izhodi

Najpogostejši izhodi PLK-ja so izhodni digitalni vmesniki. Le-ti povezujejo PLK z digitalnimi izhodnimi priključnimi napravami. Obstaja več ukazov za delo z enim ali več digitalnimi izhodi.

Slika 15 grafično prikazuje poenostavljeno 8-bitno izhodno ploščo. Tuljava koračnega motorja M1 je povezana z digitalnim izhodnim modulom (utor 7) na plošči 0, na katero se lahko veže 8 vhodnih priključnih naprav (0-7). Konkretni koračni motor M1 ima naslov 077, kar pomeni mesto njegove priključitve: 0 za ploščo, 7 za utor in 7 za priključek.



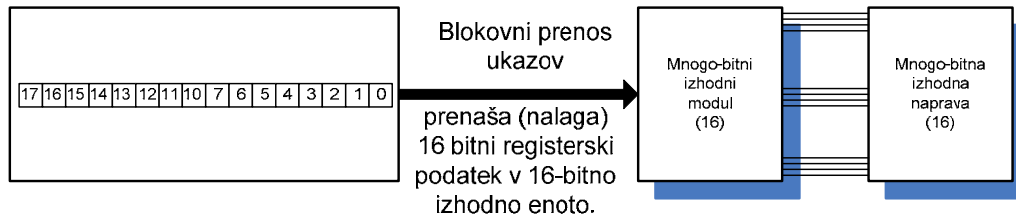
Slika 15: Grafični prikaz prenosa izhodnega bita

Pri kontroli ugotovljeno stanje (1 ali 0) pošlje program v izhodno enoto za ponastavitev izhoda. Glede na stanje odgovarjajočega bita na izhodni plošči izhodni digitalni vmesnik vklaplja ali izklaplja napetostni vir. Glede na vir napetosti ločimo več tipov izhodov :

- Tranzistorski izhodi se napajajo z napetostjo PLK-ja, njihova krmilna napetost znaša 24 V in tok ne večji od 1 A.
- Triak izhodi se napajajo z izmenično napetostjo krmilnika. Uporabljajo se v primeru izmeničnih napetosti manjših moči.
- Relejske izhode napaja zunanja napetost preko releja, ki ga krmili izhod v izhodno napravo. Relejski izhodi služijo za galvansko ločevanje krmilne napetosti od visokih uporabniških napetosti (npr. 230 V) in tokov do jakosti 5 A. Odzivni čas relejnih izhodov je daljši kot pri tranzistorskih izhodih, zato se uporabljajo v krmilnih sistemih, kjer odzivni čas ni pomemben.

Med izhodne priključne naprave sodijo zvonec, krmilni releji, ventilatorji, hupe, žarnice, koračni motorji, elektromagneti in ventili.

Večbitni izhodi, kot je BCD-register izhodov, uporabljajo funkcijske blokovne ukaze (npr. pri blokovnem prenosu) za prenos besede ali registerske vsebine na izhodno enoto (slika 16). Ta tip izhodnih ukazov (skupaj z vhodnimi ukazi) se veliko uporablja pri programiranju in kontroli I/O-signalov (L. A. Bryan, E.A. Bryan, 1997, str. 163).



Slika 16: Prenos podatkov v izhodno enoto

3.5. Analogni vhodno/izhodni sistemi

Digitalni vhodno/izhodni sistemi so dragocena orodja programsko krmilnih procesov, vendar ne morejo izpolniti vseh zahtev, ki jih prinašajo nove tehnologije in aplikacije. Tam, kjer za analiziranje signalov nista dovolj le dve stanji, se uporabljajo analogni vmesniki I/O, ki lahko registrirajo analogne (zvezne) signale, kot na primer spreminjanje napetosti ali temperature. Analogni vhodno/izhodni signali so lahko tokovni ali napetostni. Za tokovne analogne vhodno/izhodne signale se običajno uporablja območje od 4 do 20 mA, za napetostne analogne vhodno/izhodne signale pa območje od 0 do 10 V.

3.5.1. Vhodni analogni signali

Vhodno analogne module (slika 17) uporabljamo v aplikacijah, kjer je signal zunanje naprave analogen (slika 18). Z razliko od digitalnih signalov, ki poznajo le stanji 1 in 0, imajo analogni signali neskončno število stanj. Tak signal je na primer temperaturni signal, ki se stalno spreminja za infinitezimalne (neskončno majhne) vrednosti. Za PLK sprememba temperature, npr. od 20 °C do 21 °C, ne predstavlja spremembo samo za 1 °C, temveč za končno število nivojev, ki je odvisno od števila bitov A/D-pretvornika.

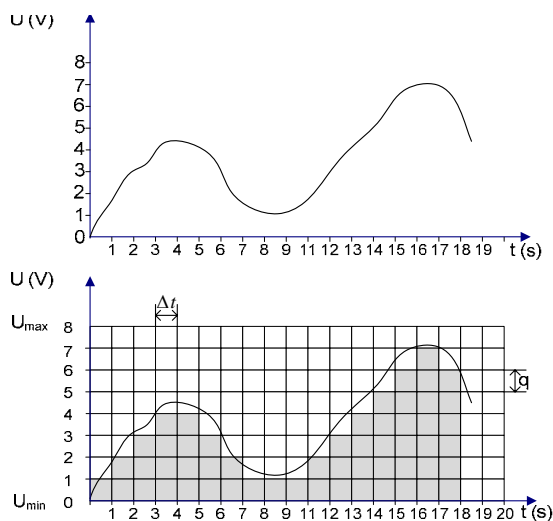


Slika 17: Analogni vhodni enota

PLK-ji, prav tako kot računalniški digitalni sistemi, razumejo le stanji 1 in 0, zato ne morejo interpretirati analognih (zveznih) signalov. Zato analogni vmesniki dosepele analogne signale pretvorijo v digitalne vrednosti, ki jih PLK-jev procesor lahko interpretira. Pri digitalizaciji signala gre za časovno in amplitudno diskretizacijo.

Amplitudna diskretizacija

Pri amplitudni diskretizaciji pretvorimo realne vrednosti napetostnega signala v enega od standardnih nivojev A/D-pretvornika, v drugem koraku pa pretvorimo diskretne vrednosti v njen dvojiški ekvivalent.



Slika 18: Vzorčenje analognega signala

Število diskretnih nivojev je odvisno od n števila bitov A/D-pretvornika in je enako

$$M = 2^n .$$

Na primer 3-bitni A/D-pretvornik ima na voljo

$$M = 2^3 = 8$$

diskretnih nivojev, s katerimi predstavimo analogni signal v digitalni obliki (slika 18). Postopek analogno/digitalne pretvorbe imenujemo kvantizacija. Pri postopku kvantizacije nastane kvantizacijski pogrešek \mathcal{E} (izguba ločljivosti zaradi diskretizacije sistema). Razliko med dvema sosednjima diskretnima nivojema pa imenujemo kvant (q), ki ga izračunamo po obrazcu:

$$q = \frac{U_{\max} - U_{\min}}{M}.$$

Velikost kvanta nam predstavlja resolucijo (ločljivost) A/D-pretvorbe. (Introduction to Digital Recording Techniques, 1989–2005)

Časovna diskretizacija

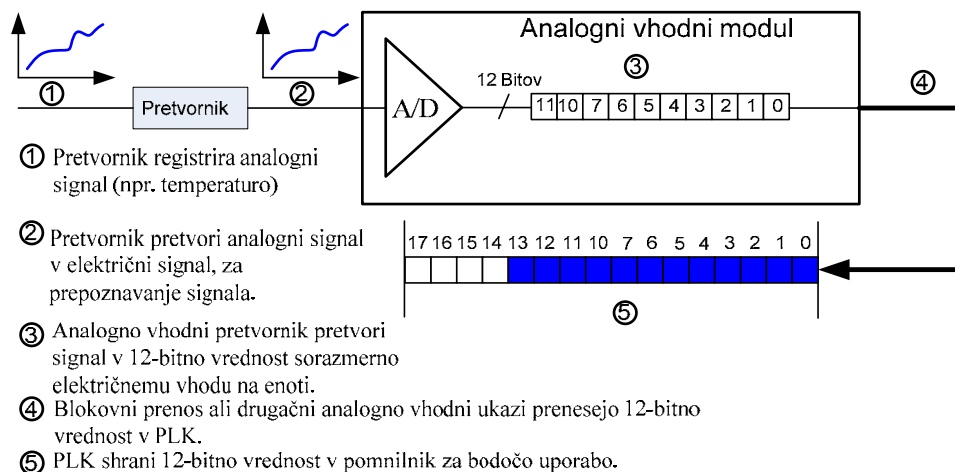
Poleg kvantizacije je potrebno opraviti tudi časovno vzorčenje. Frekvenco časovnega vzorčenja izračunamo z izrazom

$$\nu = \frac{1}{\Delta t},$$

kjer Δt pomeni periodo (interval) vzorčenja. Frekvenca časovnega vzorčenja nam pove, kako pogosto se analogni signal vertikalno preseka. Po Shannonovem teoremu mora biti ta frekvenca vsaj 2x večja od maksimalne frekvence (Introduction to Digital Recording Techniques, 1989–2005).

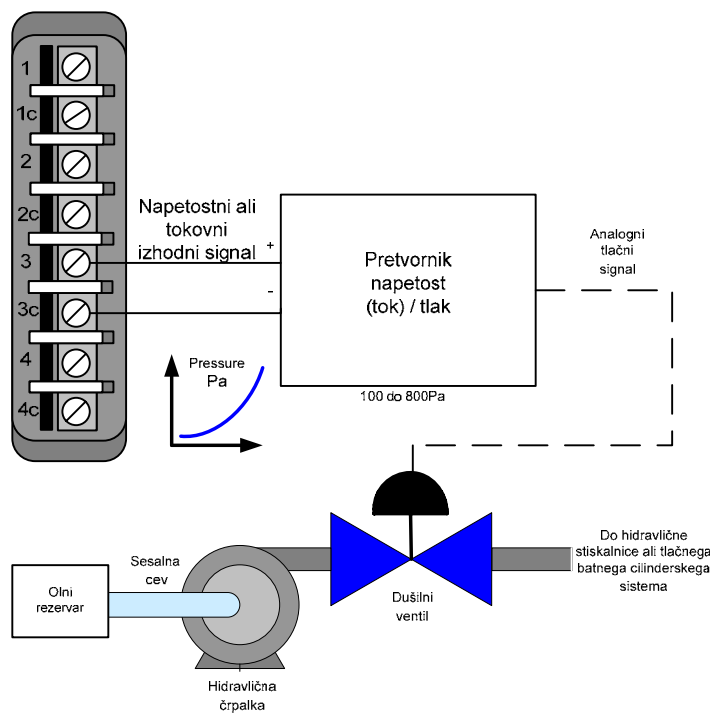
Na tak način dobljene digitalne vrednosti se potem uporabljajo v krmilnem programu. Vhodne analogne enote digitalizirajo vhodne analogne signale, ki jih nato prenesejo v PLK (slika 19). Te večbitne informacije moduli spravijo v registrski pomnilnik PLK-ja za delo s signali. Krmilni program dodeli informaciji registrski naslov za kasnejše matematične izračune, primerjave ipd. Analogna vrednost, ki se hrani v registrskem pomnilniku, je zapisana v binarni ali BCD-obliki (L. A. Bryan, E. A. Bryan, 1997).

Ukazi za delo s signali so podobni, če ne tudi enaki ukazom, ki se uporabljajo pri večbitnih digitalnih vhodih. Edina razlika med njimi je, da gre pri analognih vhodih za večbitne ukaze, ki so posledica digitalizacije analognega signala, pri digitalnih vhodih pa za večbitne ukaze, ki so posledica signalov multibitnih naprav ali ločenih signalov, povezanih z enakim številom digitalnih vhodov.



Slika 19: Prikaz korakov pri pretvarjanju analognega signala v binarno obliko

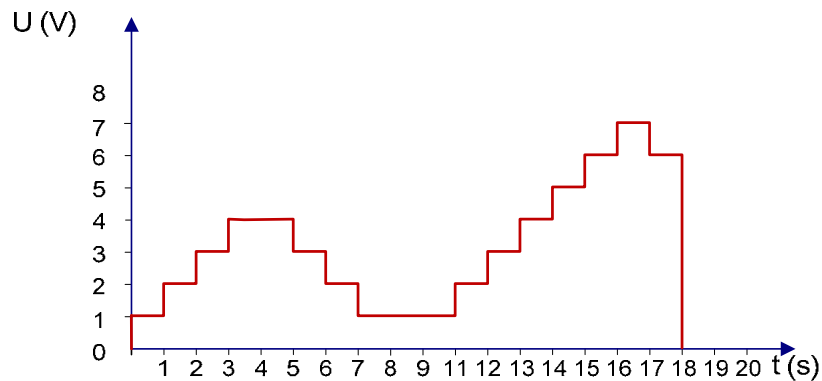
3.5.2. Analogni izhodni signali



Slika 20: Krmiljenje dušilnega ventila

Izhodni analogni vmesniki se uporabljajo v aplikacijah, ki zahtevajo kontrolo zunanjih naprav, le-te pa se odzivajo na zvezno spreminjanje električne napetosti in toka. Med analogne izhode uvrščamo analogne ventile, grafične risalnike, električne motorne pogone, analogne števec in tlačne pretvornike.

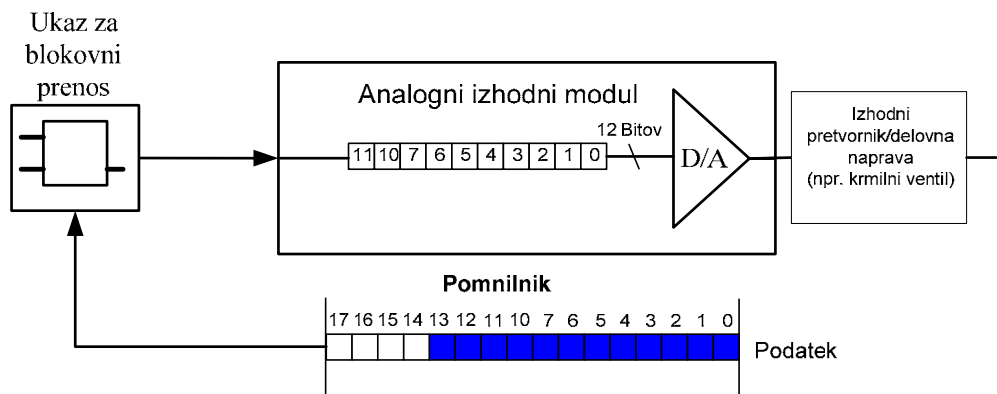
Primer zunanje naprave je dušilka (dušilni ventil) na sliki 20, ki se uporablja pri hidravličnih stiskalnicah. Dušilka regulira dotok olja v tlačne (hidravlične) cilindre, s čimer spreminja hitrost premikanja bata ali preše. Za spreminjanje pretoka olja potrebuje signal 0-10 V DC. Za krmiljenje analognih vhodov je potrebno digitalni podatek pretvoriti v analognega (slika 21). To storimo z digitalno-analognim pretvornikom D/A, kjer vsakemu digitalnemu številu pripišemo neko vrednost napetosti oziroma toka.



Slika 21: Digitalno analogna pretvorba signala

Večbitni ukazi pri analognih izhodih so podobni ukazom, ki jih uporabljamo pri večbitnih digitalnih izhodih. Uporabljajo se za prenos analogne informacije do zunanjih naprav. Z ukazom krmilnik prenese vsebino pomnilnika na izhodno enoto, ki digitalne podatke, zapisane v binarni ali BCD-obliki, pretvori v analogne vrednosti, ki jih nato pošlje naprej do zunanje naprave.

Slika 22 prikazuje večbitni ukaz za prenos 12-bitnega podatka iz pomnilnika PLK-ja v izhodno analogno enoto, povezano s krmilnim ventilom. Prenešana 12-bitna informacija je lahko rezultat tudi drugih aplikacijskih izračunov (L. A. Bryan, E. A. Bryan, 1997, str. 203).



Slika 22: Koraki pri pretvarjanju diskretne vrednosti v analogni signal

3.6. Specifikacija uporabljene strojne opreme

Na tržišču je ponudba krmilnikov velika. Odločili smo se za strojno in programsko opremo podjetja Siemens, ki je v Evropi med najbolj razširjenimi, na voljo pa je bila tudi v podjetju, kjer smo razvijali preizkuševalnik.

Uporabili smo naslednjo strojno opremo:

- krmilnik Siemens CPU313C
- prikazovalnik Siemens Simatic OP7

3.6.1. Programski logični krmilnik Siemens CPU 313C

Siemensov krmilnik CPU 313C (slika 23) ima na voljo 24 digitalnih vhodov, 16 digitalnih izhodov, 5 analognih vhodov in 2 analogna izhoda.

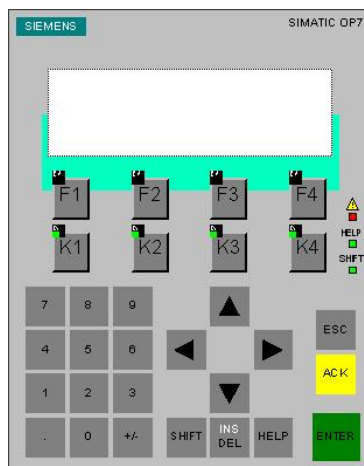


Slika 23: Siemens CPU 313C

Siemensove krmilnike se programira z programskim orodjem STEP7. Podroben opis PLK-ja in programskih jezikov se nahaja v poglavju 4.

3.6.2. Prikazovalnik Siemens Simatic OP7

Prikazovalnik je naprava, ki služi za komuniciranje med uporabnikom in PLK-jem. Povezava se imenuje HMI (Human Machine Interface). Operacijska plošča Siemens Simatic OP7 (slika 24) omogoča operacije s stanji, obdelavo podatkov in prikazovanje napak v delovanja priključenega PLK-ja. Kot dodatek lahko vhode prikažemo na prikazovalniku in jih neposredno vpisujemo v PLK. Nekatere funkcije, vezane na strojno diagnozo, je mogoče videti tudi na prikazovalniku.



Slika 24: Siemens Simatic OP7

Siemensov prikazovalnik OP7 je štirivrstični. Tipke od F1 do F4 ter od K1 do K4 so funkcijske tipke, ki se uporabljajo za izbiranje strani na prikazovalniku ali pa lahko nadomeščajo preklopnik ali katero drugo programsko funkcijo. Navigacijske tipke gor ▲, dol ▼, levo ◀ in desno ▶ služijo za premikanje po zaslonu. Numerične tipke se uporabljajo za vpisovanje potrebnih parametrov. Medtem ko se s tipko ENTER potrjujejo vnosi parametrov, se s tipko ACK potrjujejo alarmna sporočila. Prižgana led dioda pod ikono ▲ opozarja, da alarmno sporočilo ni potrjeno; utripanje te iste diode pa, da ni potrjeno več alarmnih sporočil. Tipka HELP prikličje navodila za ukrepanje v določenih situacijah. Tipka SHIFT služi za aktiviranje še druge funkcije, ki jo ima posamezna tipka. Tipka ESC ima funkcijo preklica zadnjega vnosa (če le-ta ni bil še potrjen s tipko ENTER), funkcijo vračanja prikazovalnika v prejšnje stanje,

funkcijo skrivanja sistemskih sporočil, funkcijo preklica navodil za pomoč, funkcijo resetiranja vseh sporočil med brskanjem in funkcijo prekinitve nameščanja programa v prikazovalnik (Priročnik Siemens, 1999).

4. PROGRAMSKA OPREMA

Izbira proizvajalca in tipa krmilnika definira tudi ustrezno programsko opremo, ki omogoča delo s krmilnikom. Za programiranje krmilnika in panela smo uporabili:

- Simatic Step 7 za programiranje PLK
- Protool za programiranje panela

4.1. Simatic Step 7

Programsko okolje STEP7 omogoča pisanje aplikativnih programov s pomočjo štirih programskih jezikov LAD, FBD, STL in SCL.

Siemensove PLK-je lahko programiramo s štirimi orodji:

- Ladder Logic LAD (Lestvični program),
- Function Block Diagram FBD (Funkcijsko blokovni program)
- STL Statement List (Seznam ukazov) in
- Structured Control Language SCL (Strukturno krmilni program)

Za vse aplikativne programe, izdelane z orodjem STEP 7, velja, da so napisani znotraj svojih blokov, kar omogoča klicanje drugih blokov znotraj bloka. Tako se izvede strukturiranje aplikativnega programa, ki izboljšuje organiziranost, preglednost in enostavnost vzdrževanja programov shranjenih v PLK-ju.

Orodje STEP 7 razpolaga z naslednjimi bloki (Edison Industrial Automatisation, 2007):

- Organizacijski blok (OB – Organization Block) nadzira zagon programa.
 - OB-ji so razdeljeni v razrede glede na aktiviranje (časovni, alarmni). Razredi imajo prioriteto v aktiviranju, zato lahko pride do prekinitve med OB-ji.
 - Pri aktiviranju OB-ja se izpiše tudi vzrok za aktivacijo. Informacijo se lahko ovrednoti v aplikativnem programu.
- Funkcijski blok (FB – Function Block) vsebuje uporabniški program.
 - FB se lahko oskrbi z različnimi podatki takoj, ko so le-ti poklicani iz tako imenovanih instančnih blokov. Ti podatki, kot tudi notranje

- spremenljivke (za vmesne vrednosti) in izidi logičnih funkcij so hranjeni v instančnih podatkovnih blokih, s katerimi sistem avtomatsko upravlja.
- Instančni podatkovni bloki so dodeljeni FB/SFB, ko je le-ta poklican, in se avtomatično generirajo, ko so bloki kompilirani (sestavljani).
 - Uporabnik lahko dostopa do instantnih podatkov (tudi simbolično) iz katerekoli točke aplikativnega programa kot tudi iz uporabniškega vmesnika (HMI – Human Machine Interface).
- Funkcija (FC – Function) vsebuje rutinske programe za najpogosteje uporabljene funkcije.
 - Vsaka funkcija ima določeno funkcijsko vrednost. Izhodni parametri se obdelajo takoj po klicu, zato funkcija ne zahteva nobenih instančnih podatkovnih blokov.
 - Podatkovni blok (DB – Data Block) je področje, kjer se hranijo uporabniški podatki.
 - Poleg instančnih podatkov, ki se v vsakem primeru dodelijo funkcijskemu bloku, se lahko definirajo globalni podatki, ki jih lahko uporabljajo katerikoli programski moduli.
 - Elementarni ali sestavljen podatek se lahko dodeli komponentam podatkovnega bloka. Predstavniki elementarnih podatkov so: BOOL, REAL ali INTEGER. Sestavljeni podatki (polja in strukture) vključujejo elementarne podatke. Podatke in podatkovne bloke se lahko naslavlja simbolično, kar olajšuje programiranje in čitljivost programa.
 - Sistemski funkcijski blok (SFB – System Function Block) je vgrajen v računalnikov operacijski sistem. Spremenljivke sistemskih funkcijskih blokov so arhivirane v instančnih podatkovnih blokih.
 - Sistemska funkcija (SFC – System Function) je vgrajena v računalnikov operacijski sistem.
 - Sistemski podatkovni bloki (SDB – System Data Block) so podatki za računalnikov operacijski sistem, ki vsebujejo sistemske nastavitve.

4.1.1. Ladder logic LAD (Lestvični program)

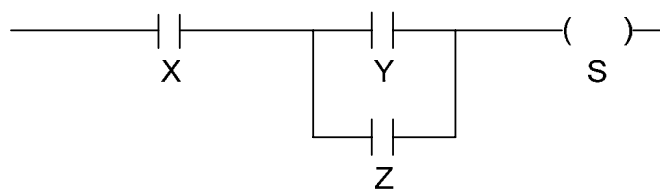
Lestvični program LAD – zelo priljubljen grafični jezik za programiranje logičnih krmilnikov – je metoda za risanje električnih logičnih shem. Ime je dobil po izpisovanju programskega jezika, ki je bil sprva podoben lestvam – dve navpičnici s serijo stopničk med njima. Lestvični program upravlja tuljave s pomočjo kontaktov, ki povezujejo ali prekinjajo tokokroge.

Jezik si lahko predstavljamo kot niz povezav med logičnimi pregledovalniki (relejnimi kontakti) in sprožilci (tuljavami). Ko obstaja pot med levo stranjo lestve in izhodom preko zahtevanih (pravih ali »zaprtih«) kontaktov, se izhod tuljave postavi na logično stanje 1 ali pravilno. Ko takšne poti ni, potem je izhod napačen (0) – tuljava se ne inducira.

Vsaka tuljava ali stik odgovarjata stanju enega bita v PLK-spominu. Lestvični program se lahko pogosto sklicuje na stanje enega bita, kar odgovarja releju z velikim številom kontaktov.

Tako imenovani kontakti lahko pomenijo vhodni signal za programski krmilnik, ki ga dajejo fizične naprave, kot so tipkala in preklopniki (mejna stikala), ali lahko predstavljajo stanje notranjih bitov ali stanje izhodnih bitov, ki so nastali nekje drugod v programu.

Vsaka vrstica lestvičnega programa LAD (slika 25) ima običajno eno tuljavo na skrajni desni strani. Izgled vrstice v lestvičnem programu LAD za funkcijo : $S = X \text{ in } (Y \text{ ali } Z)$.



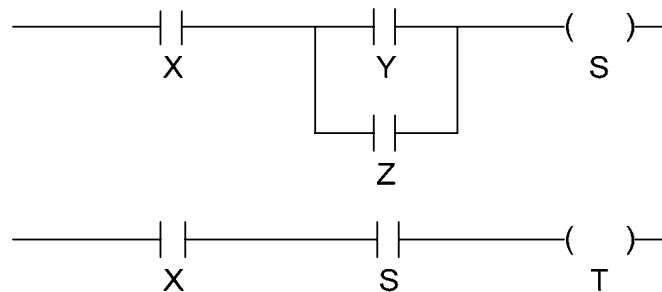
Slika 25: Primer prvega lestvičnega programa

Pomen grafičnih simbolov:

-()- – običajna tuljava; pravilno 1, ko je vrstica pravilna 1

┆┆ – simbol stikala

Ponavadi se daljši lestvični program LAD bere od leve proti desni in z vrha proti dnu. Vrednost izračunane vrstice, tj. njen tuljavni izhod, se lahko vnese kot vhodno vrednost v naslednjo vrstico programa (slika 26). V sestavljenih sistemih ima lestev veliko vrstic (stopnic), oštevilčenih po vrstnem redu ocenjevanja.



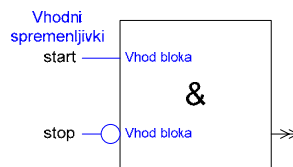
Slika 26: Primer drugega lestvičnega programa

V funkciji $T = X \text{ in } S$ predstavlja S vrednost iz prejšnje funkcije. Sedaj je podan nekoliko zahtevnejši sistem za drugo vrstico. Vrednost prve vrstice, tj. stanje na izhodni tuljavi S, se prenese v drugo vrstico, ki se potem pregleda. Dobljen rezultat T se lahko prenese na izhodno napravo (zvonec, žarnico itd.) ali v naslednjo programsko vrstico. Na takšen način lahko zelo zahtevne sisteme razčlenimo in ocenjujemo (Priročnik Siemens, 2001).

4.1.2. Function Block Diagram FBD (Funkcijsko blokovni program)

Funkcijsko blokovni program FBD opisuje funkcijo med vhodnimi in izhodnimi spremenljivkami. Funkcija je opisana kot zaporedje elementarnih blokov. Vhodne in izhodne spremenljivke povezujemo s črtami. Izhod bloka je lahko vezan z vhomom drugega bloka. Vhodi in izhodi blokov so povezani s črtami ali veznimi členi. Enojne črte se lahko uporabijo za povezovanje dveh logičnih točk diagrama:

- vhodne spremenljivke in vhod bloka

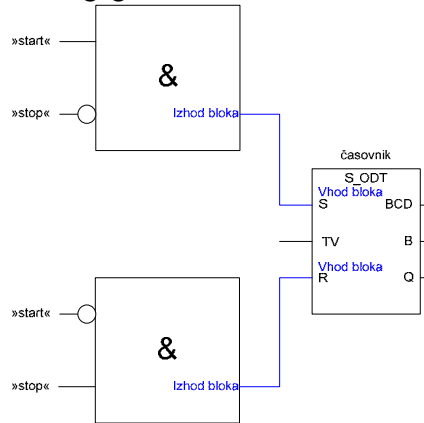


Slika 27: Povezava vhodne spremenljivke in vhod bloka

Vhodni spremenljivki start in stop sta povezani na vhod bloka logičnega IN-člena (slika 27). Operand \bigcirc pomeni negacijo spremenljivke stop. Prvi primer zahteva, da

je spremenljivka start na logični 1, spremenljivka stop pa na logični 0. Ko je pogoj izpolnjen, se izhod postavi na logično 1, v nasprotnem primeru ostane na logični 0.

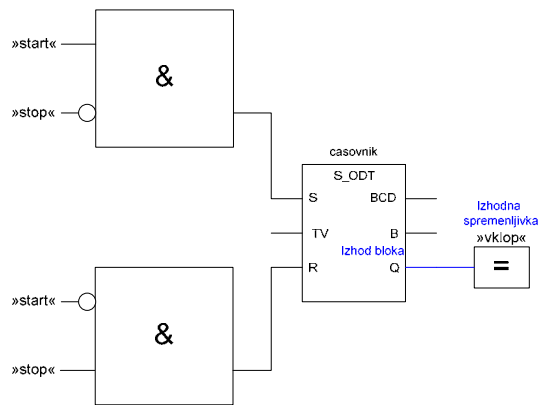
- Izhod bloka in vhod drugega bloka



Slika 28: Povezava izhoda bloka in vhod naslednjega bloka

Oba izhoda logičnih IN-blokov sta povezana na vhod časovnega bloka (slika 28) (ODT – zakasnitveni časovnik), in sicer izhod zgornjega logičnega IN-bloka je vezan na S (nastavi), izhod spodnjega logičnega IN-bloka pa na R (ponovno nastavi). Časovnik se aktivira, ko je izpolnjen pogoj zgornjega IN-bloka in izklopi, ko je izpolnjen pogoj spodnjega IN-bloka.

- Izhod bloka in izhodne spremenljivke



Slika 29: Povezava izhoda bloka in izhodne spremenljivke

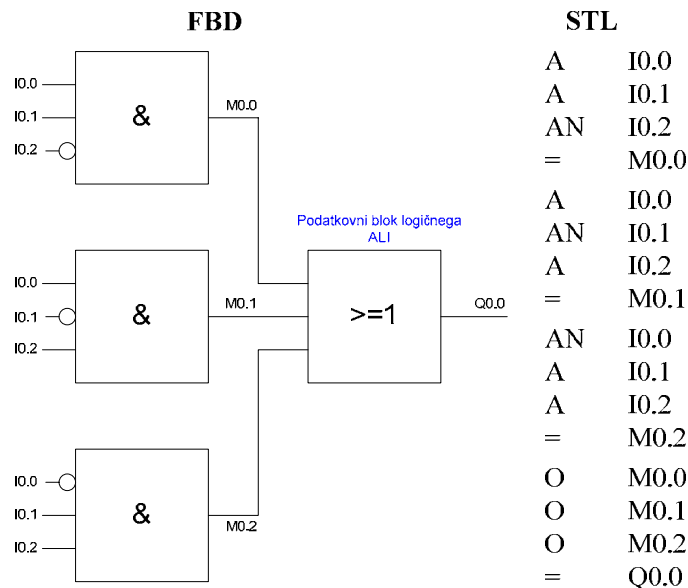
Izhod časovnega bloka je povezan na določen blok (slika 29). Ko je S na logični 1, se po izteku časa časovne zakasnitve TV postavi izhod Q na logično vrednost 1, ki postavi spremenljivko vklop na logično 1. Izhod Q in spremenljivka vklop sta izključeni, ko je R na logični 1.

Usmerjena povezava je linija, ki od leve proti desne nosi z njo povezane podatke. Začetek in konec linije morata biti istega tipa. Razvejana ali divergentna povezava se lahko uporablja za oddaljeno prenašanje informacij od njenega začetka do njenih desnih zaključkov. Vsi zaključki povezave morajo biti istega tipa (Priročnik Siemens, 2001).

4.1.3. Statement List STL (Seznam ukazov)

Seznam ukazov STL, besedilni programski jezik, nam omogoča pisanje aplikativnih programov, ki optimizirajo operacijski čas in spominski prostor glede na strojno opremo.

Program je mogoče pisati v inkrementacijskem ali tekstualnem načinu. Pri inkrementacijskem načinu lahko uporabnik takoj preveri pravilnost vsakega vpisa, pri tekstualnem načinu pa uporabnik najprej napiše program v celoti z urejevalnikom besedil z uporabo samo simbolov in nato ga prevede z ustrezno simbolno tabelo. Slika 30 prikazuje delček programa, ki je napisan na oba načina.



Slika 30: Izgled programa napisanega v FBD in STL

Oznake I0.0 do I0.2 so naslovi vhodov, oznake M0.0 do M0.2 so rezervirana spominska območja (merker) za vmesne rezultate logičnih operacij, oznaka Q0.0 je naslov izhoda. V STL-programskem jeziku pomeni ukaz A logični in (AND), ukaz

AN logični ne-in (N-AND) in ukaz O logični ali (OR). Simbol za enakost določa mesto hranjenja logičnih rezultatov (Edison Industrial Automatisation, 2007).

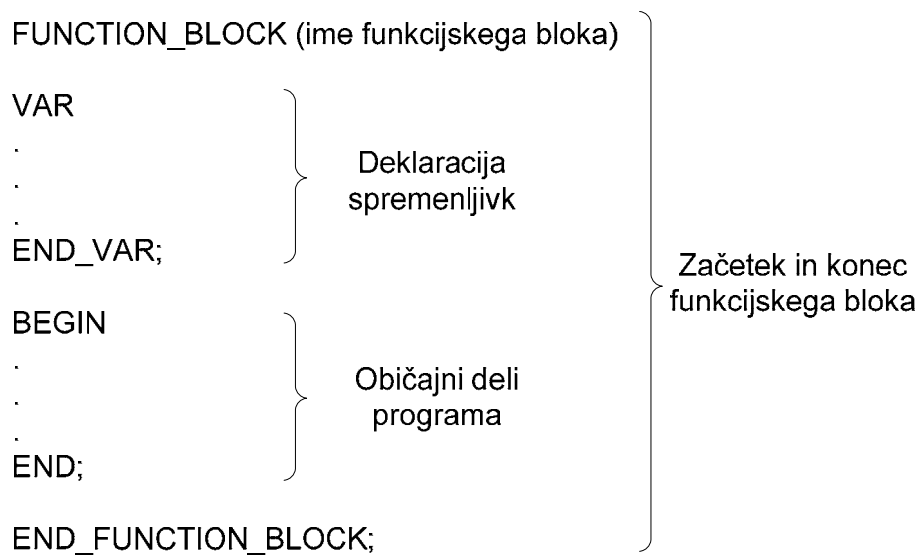
4.1.4. Structure Control Language SCL (Strukturno programski jezik)

Strukturno programski jezik SCL je zahtevnejši programski jezik, pisan v Pascalu. Grajen je na standardih za programske elektronske krmilnike. Z mednarodnim standardom IEC 1131-3 so standardizirani programski jeziki za PLC-je. Strukturno programski jezik SCL izpolnjuje osnovni nivo za strukturno besedilni jezik ST, ki ga določa ta standard. Poleg tega uporablja programske elemente tipične za PLC-je, kot so vhodi, izhodi, časovniki, bitni spomin, blokovni sklici itd. Z drugimi besedami, strukturno programski jezik SCL dopolnjuje in razširja STEP 7 programsko orodje in njegove programske jezike Ladder Logic LAD, Function Block Diagram FBD in Statement List STL. Strukturno programski jezik SCL ne pomaga uporabniku le pri običajnih krmilnih nalogah, temveč tudi pri obsežnejših aplikacijah, kar ga dela za neprekosljivega na področjih, kot so delo s podatki, optimizacija procesov, upravljanje z recepti in v matematično/statističnih operacijah. Strukturno programski jezik SCL podpira blokovni koncept STEP 7 in tako omogoča standardizirano blokovno programiranje, prav tako kot se to počne s Statement List STL, Ladder Logic LAD in Function Block Diagram FBD. Za izmenjavo podatkov med Simatic S7 PLC-ji in ostalimi napravami, ki podpirajo S7 protokol, npr. s PC-ji, se uporablja S7-protokol.

Pomembna razlika med strukturno programskim jezikom SCL in drugimi programskimi jeziki je v izvajanju krmilnega programa. Strukturno programski jezik SCL bo pretekel pot od začetka do konca programa večkrat v sekundi, medtem ko ostali programi ne dosežejo konca, dokler se popolnoma ne zaključijo. V prejšnjem primeru bi zanka lahko vodila v program, ki bi se (z nekaj spremembami) lahko ujel v neskončno zanko. Če do tega pride med samim krmiljenjem, lahko proces postane nevaren, zato se bo krmilnik nehal odzivati, njegov varnostni časovnik pa bo javil napako.

Strukturno programski jezik SCL je bil konstruiran za delo z ostalimi programskimi jeziki PLK. Tako lahko program v lestvični logiki LAD pokliče podprogram napisan v strukturno programskem jeziku SCL.

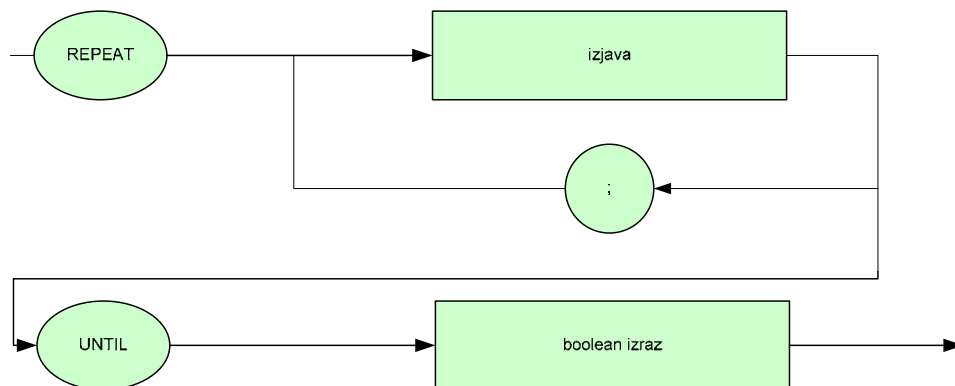
Pri pisanju programa v SCL-ju je potrebno najprej definirati začetek in ime uporabljenega funkcijskega bloka (FB) ali funkcije (FC), sledi deklariranje spremenljivk v prostoru med VAR in END_VAR, med ukazoma BEGIN in END pa se nahajajo običajni deli programa. Na koncu programa sledi ukaz END_FUNCTION_BLOCK, ki zaključi funkcijski blok.



Repeat, case in if so ukazi, ki se največ uporabljajo pri programiranju s strukturno programskim jezikom SCL (Priročnik Siemens, 2001).

4.1.4.1. Ukaz REPEAT

Ukaz 'repeat' (slika 31) omogoča ponavljanje zanke, dokler pogoj ni izpolnjen. Ukaza REPEAT in END_REPEAT definirata zanko. Ukaz UNTIL pove, kdaj se zanka zaključi.



Slika 31: Diagram poteka repeat ukaza

Primer uporabe ukaza repeat:

```

FUNCTION_BLOCK repeat
VAR
I: int := 0;
END_VAR

BEGIN

REPEAT
I:= I + 1;
UNTIL I>=10;
END_REPEAT;

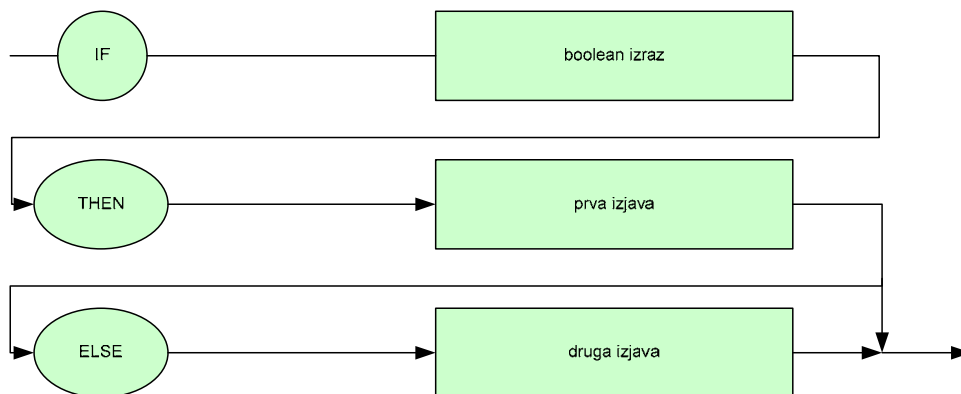
END;
END_FUNCTION_BLOCK

```

Med ukazoma VAR in END_VAR je spremenljivka i definirana kot celo število. Program sledi tem izjavam za spremenljivke in šteje po zanki od 0 do 10. Ko se program začne izvajati, se postavi vrednost za spremenljivko i na 0. Dodana je vrstica, ki poveča vrednost spremenljivke i za 1.

4.1.4.2. Ukaz IF

Ukaz IF (slika 32) omogoča izvajanje različnih vej programa. Ko je pogoj stavka IF uresničen, se izvrši izbrani ukaz oziroma zaporedje ukazov. Če pogoj stavka IF ni izpolnjen, potem se po izbiri izvede alternativno zaporedje ukazov. Pogoji so preračunani kot logični izrazi (obrazci, formule). Logični izraz je pravilen, če je vrednost 1, in napačen, če je vrednost 0. Ukaz je izpolnjen, ko se pride do konca stavka IF. Stavki IF se lahko vleče skozi več vrstic.



Slika 32: Diagram poteka stavka IF (pogojni stavek)

Primer uporabe stavka IF:

```

FUNCTION_BLOCK if
VAR
pogoj1: bool := 0;
l: bool :=0;
END_VAR

BEGIN

IF vhod1=true THEN
Izhod1:=true;
ELSE
Izhod1:=false;
END_IF;

END;
END_FUNCTION_BLOCK

```

Iz danega primera lahko preberemo: Izhod1 se postavi na logično 1, ko je izpolnjen pogoj »vhod1 postavljen na logični 1«, oziroma ostane na logični 0, ko ta pogoj ni izpolnjen.

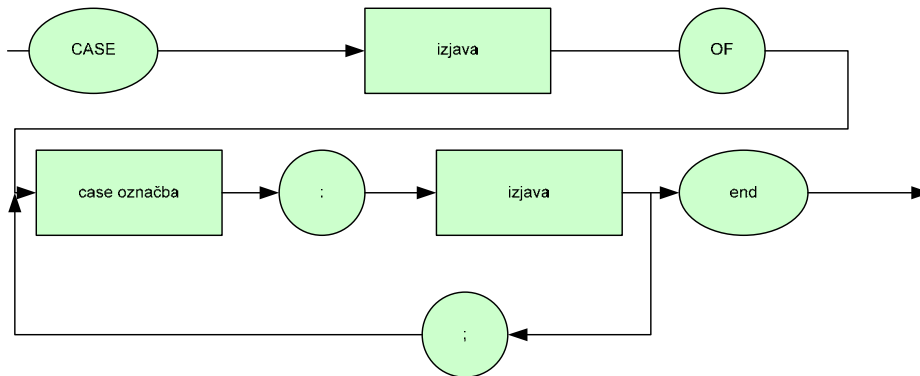
4.1.4.3. Ukaz CASE

Stavek CASE (slika 33) omogoči vrednostni spremenljivki ali izrazu, da kontrolira potek izvajanja programa. Običajna sintaksa stavka CASE je:

- Prva vrstica vsebuje aktualno besedo »CASE«. Na to spremenljivko ali stavek se obračamo kot na »kontrolno spremenljivko« stavka CASE (preklopnega stavka).

- Naslednje vrstice definirajo enega ali več blokov programske kode, ki predstavljajo možne poti, ki jih program lahko ubere.

Vsak blok se začne z vrstico, ki vsebuje vrednost, ki jo ima kontrolna spremenljivka. Ko vrednost kontrolne spremenljivke ustreza tej vrednosti, se izvajanje programa premakne na ta del programske kode. Sicer program pregleda vrednost naslednjega bloka (če je prisoten) in postopek se ponovi.



Slika 33: Diagram poteka stavka CASE

Primer uporabe ukaza CASE:

```

FUNCTION_BLOCK case
VAR
    korak: int := 0;
    izhod1: bool :=0;
    izhod2: bool :=0;
    izhod3: bool :=0;
    izhod4: bool :=0;
    izhod5: bool :=0;
    izhod6: bool :=0;
END_VAR

BEGIN

CASE korak OF
    1: izhod1:=true;
       izhod2:=false;
    2: izhod1:=false;
       izhod2:=true;
    3: izhod3:=true;
       izhod4:=false;
    4: izhod3:=false;
       izhod4:=true;
    5: izhod5:=true;
       izhod6:=false;
    6: izhod5:= false;

```

```
    Izhod6:=true;  
END_CASE;  
END;  
END_FUNCTION_BLOCK
```

Iz zgornjega primera lahko preberemo: Pri izvajanju stavka CASE za korak 1 se izhod1 postavi na logično vrednost 1, izhod2 pa ostane na logični vrednosti 0. Pri izvajanju stavka CASE za korak 2 pa se izhod1 postavi na logično vrednost 0, izhod2 pa na logično vrednost 1 itd.

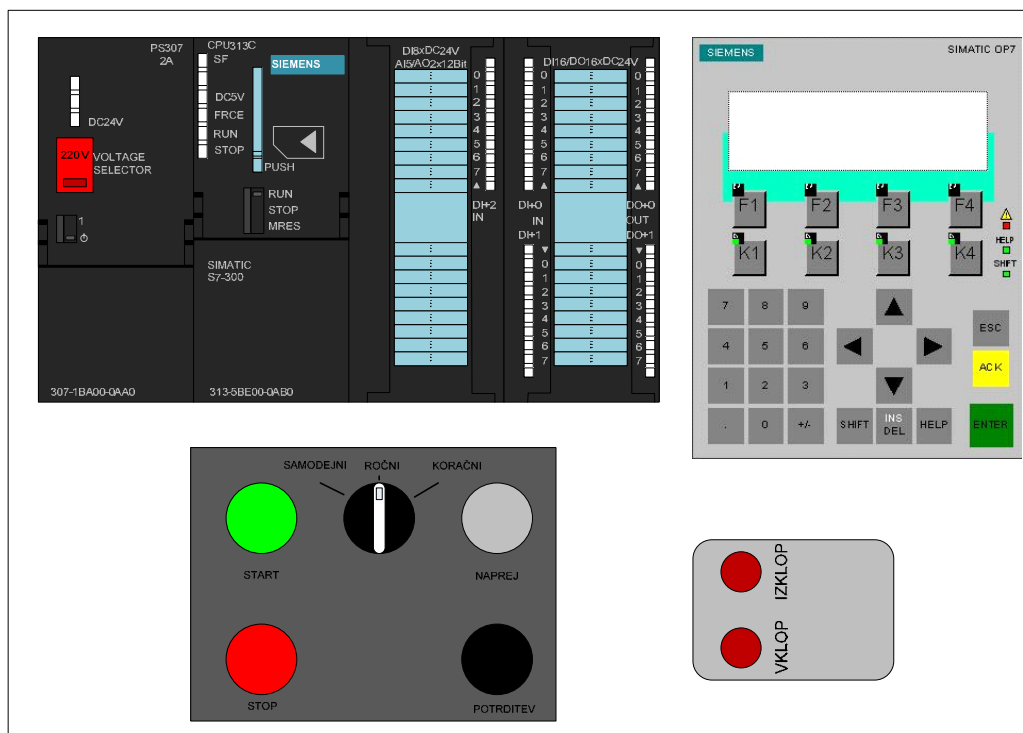
4.2. Protool

Program Protool služi izključno za izpisovanje teksta in izrisovanje grafov in slik na Siemens-Simaticovih prikazovalnikih. Program je enostaven za uporabo. Omogoča izdelovanje strani s pripadajočimi teksti in okenc za vnašanje parametrov. Pišemo lahko alarmna sporočila in druga sporočila. Za posebne funkcije so namenjeni prostorski kazalci. Pri vsakem projektu je potrebno izbrati uporabljen prikazovalnik in način povezave med prikazovalnikom in PLK-jem. Ko je projekt zaključen, ga je potrebno naložiti v prikazovalnik in preizkusiti njegovo delovanje (Priročnik Siemens, 1999).

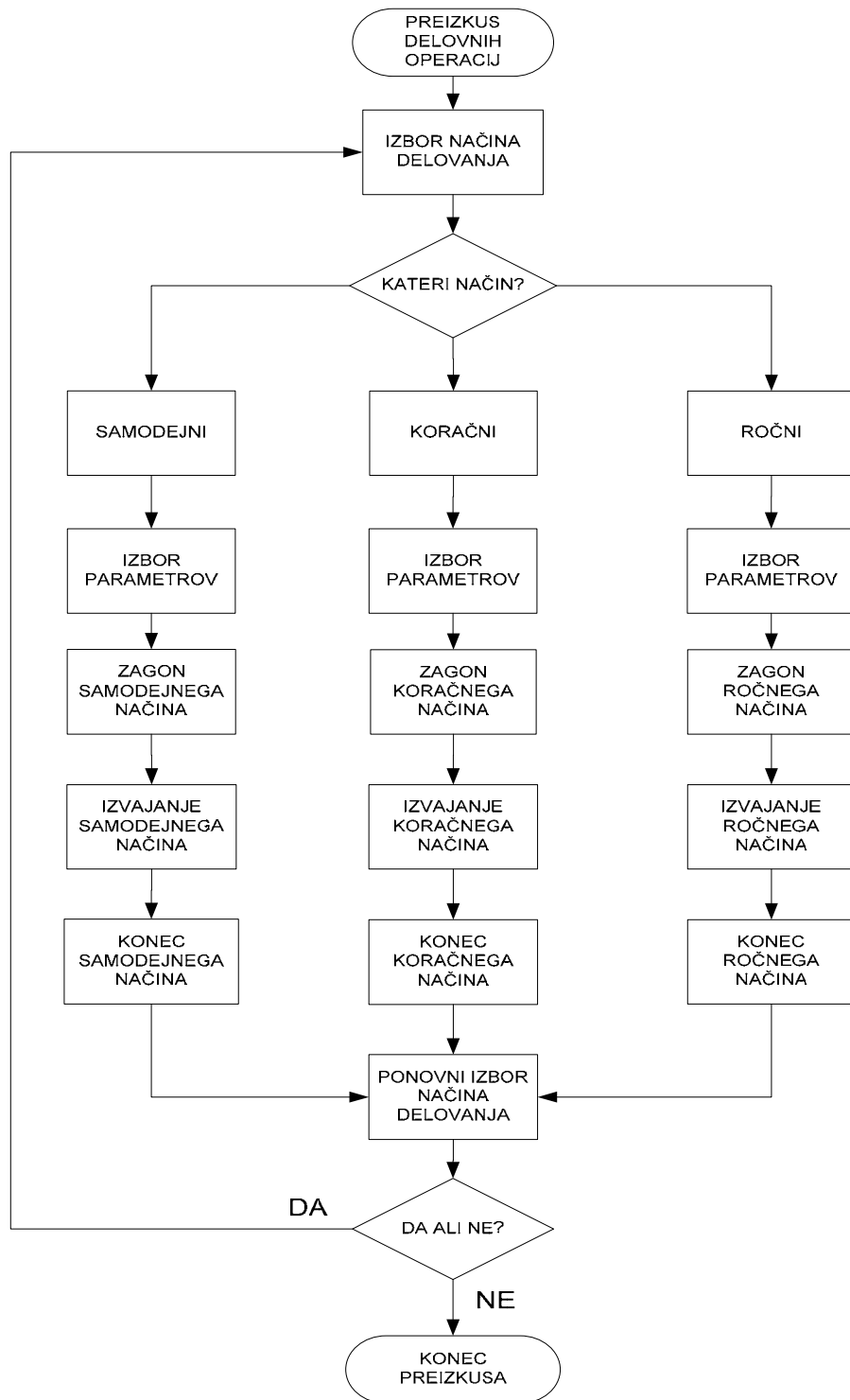
5. IZDELAVA PREIZKUŠEVALNIKA

V prejšnjih dveh poglavjih so predstavljena osnovna orodja za izdelavo preizkuševalnika – PLK kot osnova za izgradnjo sistema in pa programsko orodje STEP 7 kot orodje za izvedbo vseh zahtevanih funkcij. V nadaljevanju bo podan kratek opis realizacij osnovnih funkcij preizkuševalnika in načina delovanja. Preizkuševalnik (slika 34) je zasnovan tako, da omogoča uporabo več različnih načinov. Vsak preizkus zahteva natančen vnos parametrov glede na delovanje naprave (mehanskega sklopa) v preizkusu. Za vnos parametrov je izdelan uporabniški vmesnik, ki ga sestavljajo tipke in preklopniki ter operacijska plošča na preizkuševalni napravi.

S preklopnikom izbiramo med samodejnim, koračnim in ročnim načinom delovanja. Način zagona je odvisen od načina delovanja, sam postopek preizkušanja pa je v vseh primerih enak. Delovanja preizkuševalnika je v obliki diagrama prikazano na sliki 35.



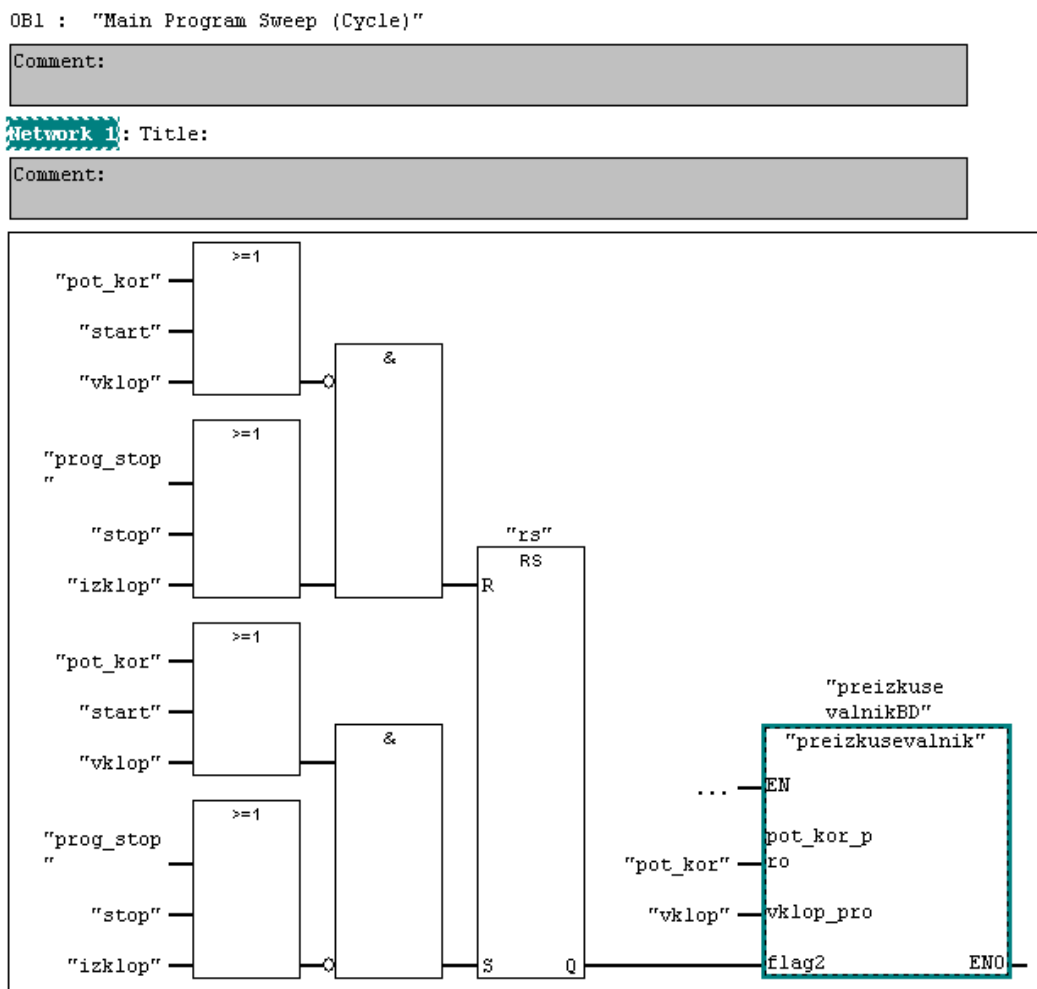
Slika 34: Preizkuševalnik



Slika 35: Delovanje preizkuševalnika

5.1. Izvedba osnovnih funkcij

Vse zahteve naročnika so bile izvedene s programi Siemens SIMATIC STEP 7 in ProTool. Prvi del krmilnega programa je napisan s programom FBD (slika 36), preostali del programa, ki se nahaja v označenem kvadratu, pa je napisan v programski obliki SCL. Tu se nahajajo vse ostale funkcije in ukazi za izpisovanja zaslona na prikazovalniku.



Slika 36: Programska koda aplikacije

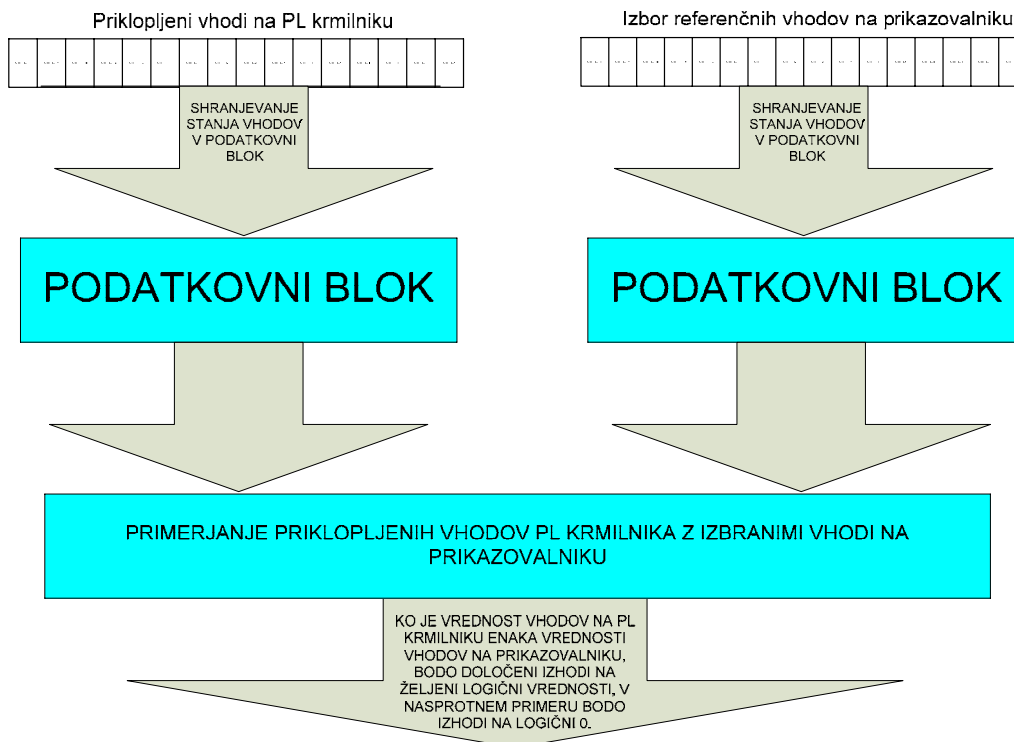
V diplomski nalogi so podrobneje opisani samo ključni deli programa, in sicer:

- izvedba poljubnega izbiranja vhodov
- izvedba nastavljanja časovnih zakasnitev
- izvedba izpisovanja napak na prikazovalniku
- izvedba poljubnega določanja izhodov

- izvedba merjenja z merilno sondo
- izvedba merjenja temperature z temperaturno sondo

5.1.1. Izvedba poljubnega izbiranja vhodov

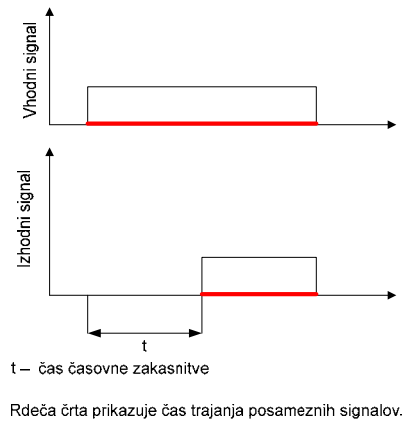
Ob zagonu koraka aplikativni program najprej primerja stanje priključenih vhodov na PLK-ju in referenčnih vhodov na prikazovalniku, kot prikazuje slika 37. Če želimo, da bo na priključenih izhodih zahtevano logično stanje, se morajo izbrani vhodi na prikazovalniku OP7 ujemati s priključenimi vhodi na PLK-ju. Izbranim vhodom se dodeli ustrezna logična vrednost in primerja s priključenimi vhodi na PLK-ju. Ko izbrane vrednosti vhodov na prikazovalniku ustrezajo vrednostim priključnih vhodov na PLK-ju, imamo na izhodu zahtevano logično stanje; če se vrednosti ne ujemajo, pa se na prikazovalniku pojavi napaka pri vnosu (poglavje 5.3.).



Slika 37: Primerjava vhodov na PLK-ju z izbranimi vhodi na prikazovalniku

5.1.2. Izvedba nastavljanja časovnih zakasnitev

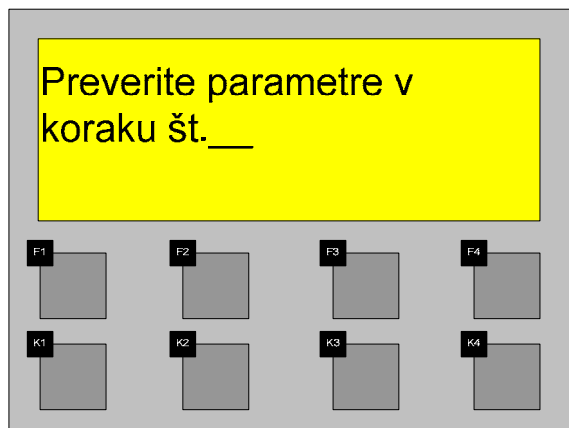
Prikazovalnik omogoča nastavljanje časovnih zakasnitev tipa ODT (On-Delay Timer). Časovno zakasnitev med koraki uporabljamo, da preprečimo trke cilindrov ali drugih delov stroja. V primeru časovne zakasnitve se izhodni signal postavi na logično 1 šele po izteku časa, predvidenem s časovno zakasnitvijo, vhodni signal pa ves ta čas stoji na logični 1. Paziti moramo, da časovna zakasnitev ni daljša od časa trajanja vhodnega signala. Slika 38 prikazuje delovanje časovne zakasnitve.



Slika 38: Prikaz časovne zakasnitve

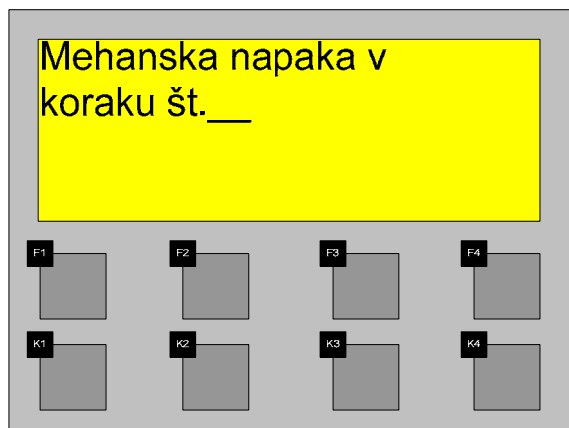
5.1.3. Izvedba izpisovanja napak na prikazovalniku

Uporabniški program izpisuje dve vrsti alarmnih sporočil. Prvo alarmno opozorilo dobimo, če se vrednosti priključenih vhodov na PLK-ju in referenčnih vhodov na začetku izvajanja korakov ne ujemajo, ali pa tolerančne meje za merilno ali temperaturno sondo niso bile vnešene. Na zaslonu prikazovalnika se izpiše alarmno sporočilo, ki od uporabnika zahteva, da preveri parametre v konkretnem koraku (slika 39).



Slika 39: Prvo alarmno sporočilo

Drugo alarmno poročilo sporoča, da je prišlo do mehanske napake v določenem koraku ali da je prekoračeno dopustno tolerančno območje za temperaturo ali mero (slika 40). V primeru, da se v koraku en od priključenih vhodov (v času 10 sekund) ne postavi na zahtevano stanje, se delovanje preizkuševalnika ustavi, na prikazovalniku se izpiše "Mehanska napaka v koraku št. ___", kar zahteva podroben pregled same povezave preizkuševalnika in stroja in pregled mehanskih sklopov stroja, ker je lahko prišlo do okvare ali loma. Na koncu delovanja lahko določen način delovanja ponovno zaženemo v izbranem načinu ali izberemo drugi način delovanja.

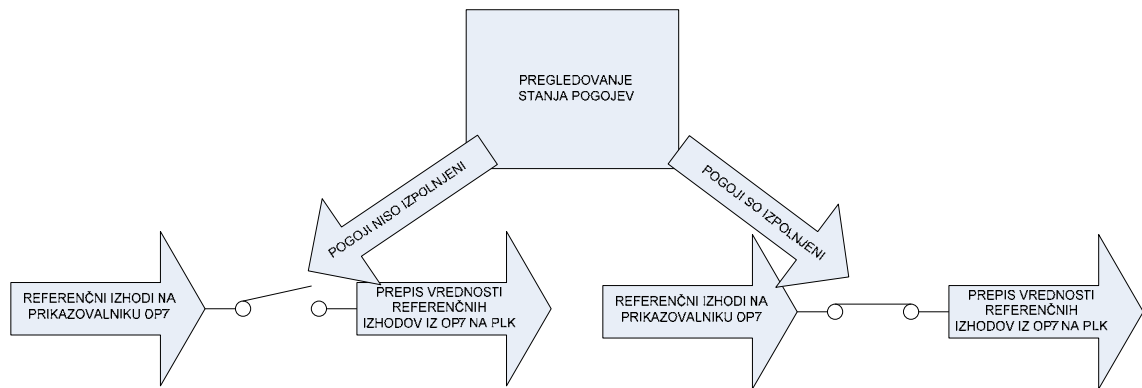


Slika 40: Drugo alarmno sporočilo

5.1.4. Izvedba poljubnega določanja izhodov

Na prikazovalniku OP7 postavimo izhod v željeni položaj tako, da jim pripišemo število 1, če gre za aktiven izhod, in število 0 za neaktiven izhod. Aktivni izhodi se postavijo na logično stanje 1, ko so pogoji izpolnjeni (slika 41). Neaktivni izhodi se

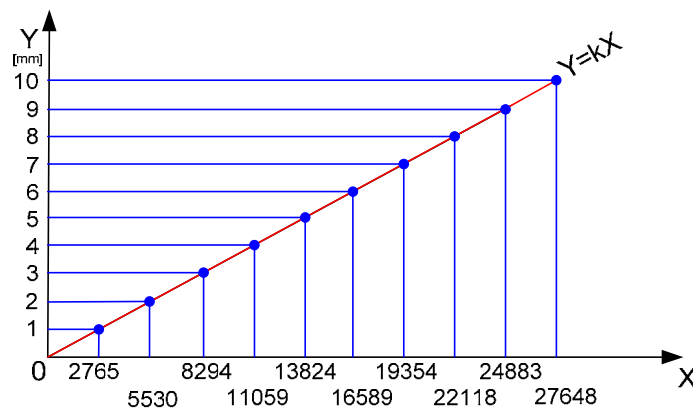
ob izpolnitvi pogojev ne vklopijo. Izhodne vrednosti se po izpolnitvi pogojev iz prikazovalnika OP7 prenesejo v uporabniški program.



Slika 41: Izbiranje izhodov

5.1.5. Izvedba merjenja z merilno sondo

Merilno sondo priključimo na analogni vhod PLK-ja, ki se napaja z električno napetostjo 0 V–10 V. Električna napetost je analogna spremenljivka in se spreminja premosorazmerno z upornostjo sonde. Njenim skrajnim mejam so predpisane diskretne vrednosti, in sicer 0 za 0 V in 27648 za 10 V. Vmesne diskretne vrednosti se računajo z linearno funkcijo. Dolžina merilne sonde je 10 mm, zato diskretna vrednost 0 ustreza meri 0.0 mm, diskretna vrednost 27648 pa dolžini 10.0 mm. Merilna sonda meri na 0,1 mm natančno. Vrednost meritve y pri ustrezni digitalni vrednosti x izračuna program kar z linearno funkcijo $y = k * x$, kjer je k smerni koeficient funkcije in ima vrednost $k = 10/27648 = 3,6169 \cdot 10^{-4}$ (Slika 42).



Slika 42: Graf funkcije diskretne spremenljivke

5.1.6. Izvedba merjenja temperature s temperaturno sondo

Temperaturna sonda PT100 služi za merjenje temperatur v merilnem območju od 0 °C do 800 °C. Sondo priključimo na analogni vhod PLK-ja, rezerviran za merjenje temperatur. Iz vhodne enote pride do PLK-ja vrednost, ki je deset krat večja od vrednosti dejanske temperature. PLK izračuna dejansko temperaturo po enačbi:

$$T_D = \frac{T_V}{10},$$

kjer pomeni:

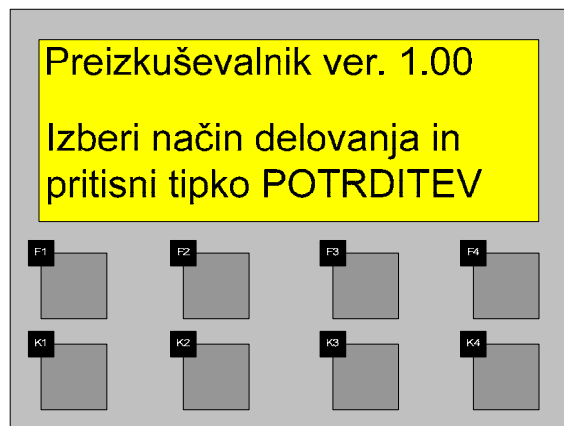
T_D – dejanska temperatura v °C

T_V – vrednost, ki jo pošlje modul PLK-ju

in je vključena v uporabniškem programu.

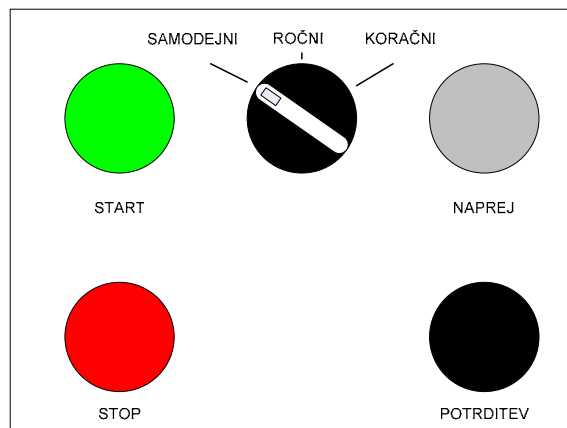
5.2. Izbira načina delovanja

Pri vsakem vklopu preizkuševalnika se na zaslonu prikazovalnika pojavi stran (slika 43),



Slika 43: Zagon preizkuševalnika

ki zahteva izbiro načina delovanja. Način delovanja izvedemo s preklopnikom med tremi možnimi načini: samodejnim, koračnim in ročnim načinom. Preklopnik obrnemo na ustreznemu način delovanja. S tipko POTRDITEV pa potrdimo to izbiro (slika 44),



Slika 44: Izbor načina delovanja

šele potem lahko vnašamo potrebne parametre. Samodejni način poženemo s tipko START, koračni način s tipko NAPREJ, ročni način pa s tipko VKLOP na daljinskem upravljalcu. Sledi izvajanje in zaključek delovanja. Na koncu lahko ponovno izbiramo med možnimi načini delovanja ali pa končamo preizkus.

5.3. Vnos parametrov

Pri vsakem načinu moramo najprej definirati parameter vsakega koraka. Ti parametri so:

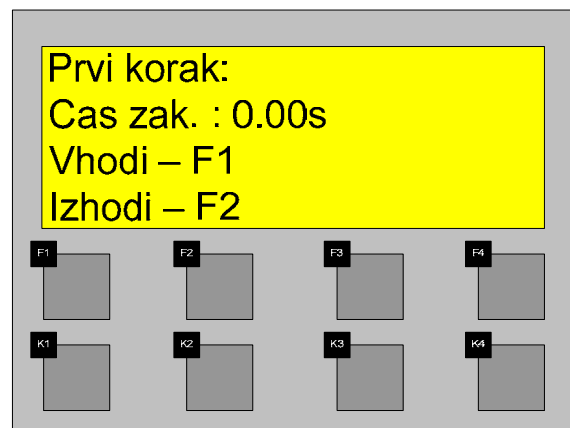
- Vhodi – določajo trenutni položaj mehanskega sklopa.
- Izhodi – glede na izpolnjene pogoje (stanja vhodov) aktiviramo oziroma deaktiviramo določen izhod oz. izhode.
- Časovna zakasnitev – določimo zakasnitev med izvajanjem posameznih korakov.
- Parametri merilne sonde:
 - Sp. meja – določanje spodnje meje histereze
 - Zg. meja – določanje zgornje meje histereze
 - Tre. vred. – trenutna dolžina merilne sonde
- Parametri temperaturne sonde:
 - Sp. T meja – določanje spodnje meje histereze
 - Zg. T meja – določanje zgornje meje histereze
 - Tre. vred. – trenutna temperatura sonde

Po potrditvi izbranega načina delovanja se na zaslonu prikazovalnika prikaže stran, kot kaže slika 45.



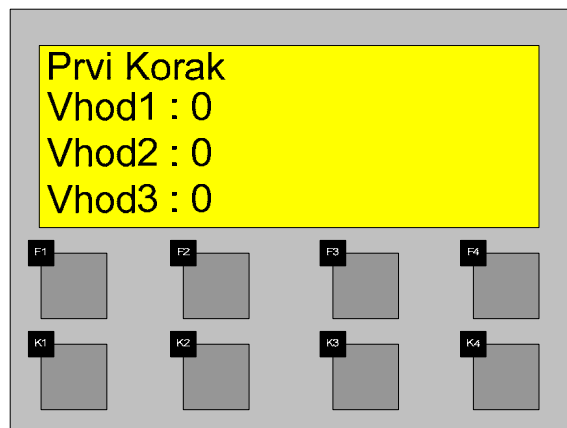
Slika 45: Izpis izbranega načina delovanja

Po zaslonu se premikamo z navigacijskimi tipkami gor ▲ in dol ▼ . S funkcijskimi tipkami F1, F2, F3 in F4 pa izbiramo korake. Pritisk na tipko F1 odpre na prikazovalniku stran za prvi korak (slika 46).



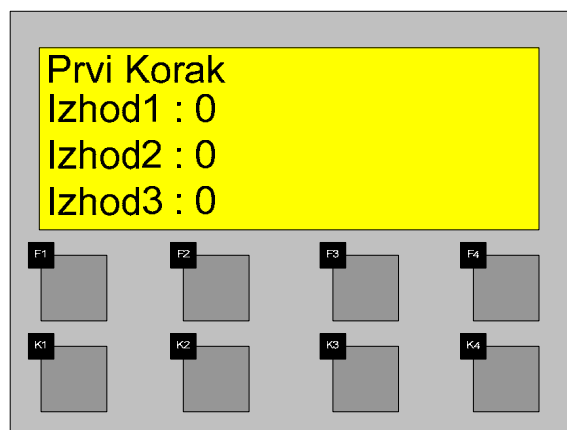
Slika 46: Izpis na zaslonu pri izbiri koraka

S pritiskom na tipko ENTER lahko vnesemo časovno zakasnitev iz od 0.01 s do 9.99 s. Vnos potrdimo s ponovnim pritiskom na tipko ENTER. S pritiskom na tipko F1 se prestavimo na stran za določanje vhodov pri prvem koraku (slika 47).



Slika 47: Izbira vhodov v prvem koraku

S pomočjo navigacijskih tipk postavimo vrednosti vhodov na zahtevano stanje. Ko so parametri vhodov ustrezno vnešeni, pritisnemo na tipko ESC, ki nas vrne nazaj na prejšnjo stran (slika 46). Pritisk na tipko F2 odpre stran za določanje izhodov v prvem koraku (slika 48).



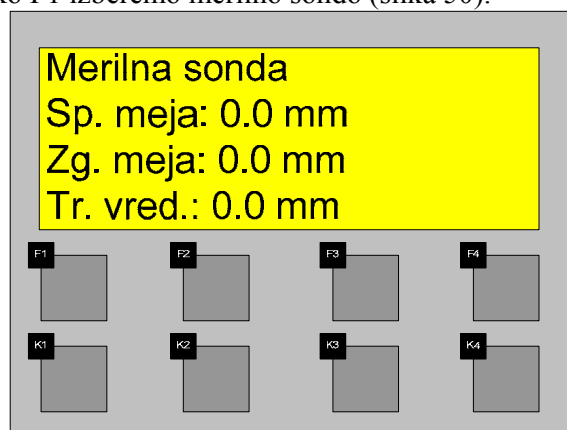
Slika 48: Izbira izhodov v prvem koraku

S pomočjo navigacijskih tipk postavimo vrednost izhodov na ustrezno vrednost. S pritiskom na tipko ESC se ponovno vrnemo na prejšnjo stran (slika 46). Z navigacijsko tipko dol odpremo stran, kjer lahko izberemo vrsto meritve (sliki 49).



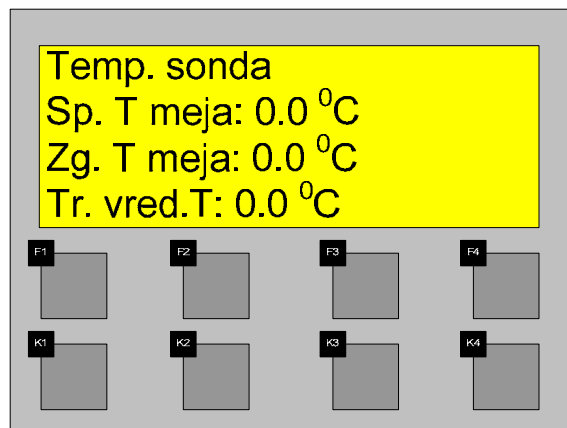
Slika 49: Izbira meritve v koraku

S pritiskom na tipko F1 izberemo merilno sondo (slika 50).



Slika 50: Nastavljanje parametrov za merjenje dolžine

Pritisk na tipko ENTER nam dovoli vnos podatka za najmanjšo mero, ki ga potrdimo s ponovnim pritiskom na tipko ENTER. Z navigacijsko tipko se spustimo na vrstico Zg. meja. Po pritisku na tipko ENTER lahko vnesemo podatek za zgornjo mejo, ki ga potrdimo s ponovnim pritiskom na tipko ENTER. Logično je, da mora biti zgornja dopustna meja večja od spodnje dopustne meje za merjenec. V vrstici Tr. vred. se prikazuje trenutna vrednost meritve, ki jo registrira merilna sonda. Tipka ESC nas vrne nazaj na prejšnjo stran (slika 49). S tipko F2 izberemo temperaturno sondo (slika 51).



Slika 51: Nastavitev parametrov za merjene temperature

Pritisk na tipko ENTER nam dovoli vnos podatka za najnižjo temperaturo, ki ga potrdimo s ponovnim pritiskom na tipko ENTER. Z navigacijsko tipko se premaknemo na vrstico Zg. T meja. Po pritisku na tipko ENTER lahko vnesemo podatek za najvišjo temperaturo, ki ga potrdimo s ponovnim pritiskom na tipko ENTER. V vrstici Tr. vred. se prikazuje trenutna temperatura. Zgornja dopustna temperatura mora biti večja od spodnje dopustne temperature. Z dvojnimi pritiskom na tipko ESC se vrnemo na stran (slika 49), kjer lahko vnašamo parametre za naslednji korak itn.

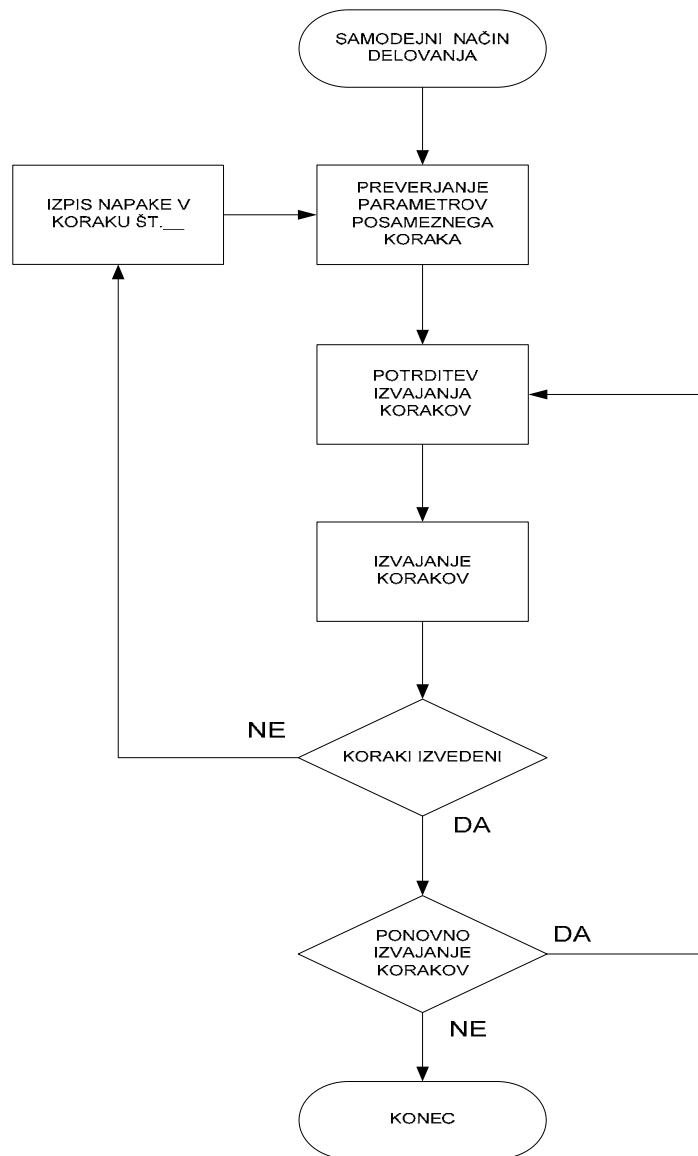
5.4. Načini delovanja

Za kontrolo funkcionalnosti mehanskih sklopov uporabljamo:

- samodejni,
- koračni in
- ročni način kontrole.

5.4.1. Samodejni način delovanja

Pri samodejnemu načinu delovanja (slika 52) program najprej preveri vnos parametrov v posameznem koraku. Vsaka napaka pri vnosu parametrov, ki jo je potrebno odpraviti, se pokaže na prikazovalniku. Ko se samodejni način izvede do konca, imamo na voljo ponovni zagon v samodejnem ali v katerem drugem načinu.



Slika 52: Samodejni način delovanja

S tipko START zaženemo samodejni način. Program preveri za vsak korak najprej (glede na izbrani način), če:

- je vrednost priključenih vhodov na PLK-ju enaka vrednosti referenčnih vhodov na prikazovalniku,
- sta določeni spodnja in zgornja meja za merilno sondo in
- sta določeni najvišja in najnižja temperatura za temperaturno sondo.

V primeru, da vrednosti vhodov ne ustrezajo ali pa manjka vnos dopustnih mej, se na prikazovalniku izpiše »Preverite parametre v koraku št. __«.

V primeru, da vrednosti vhodov ustrezajo in so vnešeni vsi podatki, korak pa se ni izvedel, se na prikazovalniku izpiše »Mehanska napaka v koraku št. __«. Ta izpis opozarja na odstopanje od dopustnih mej in na morebitni lom ali trk mehanskega dela stroja.

V primeru, da vrednosti vhodov ustrezajo in so vnešeni vsi potrebni podatki ter so izmerjeni podatki v dopustnih mejah, se prvi korak izvede, program pa stopi v naslednji korak, kjer se postopek kontrole ponovi. Po izvedbi vseh korakov, se lahko samodejni način ponovi ali izbere drug način delovanja. Samodejni način ustavimo s tipko STOP.

5.4.2. Koračni način delovanja

Koračni način delovanja (slika 53) se od samodejnega načina delovanja razlikuje le v tem, da je tu potreben pritisk na tipko NAPREJ za zagon posameznih korakov.

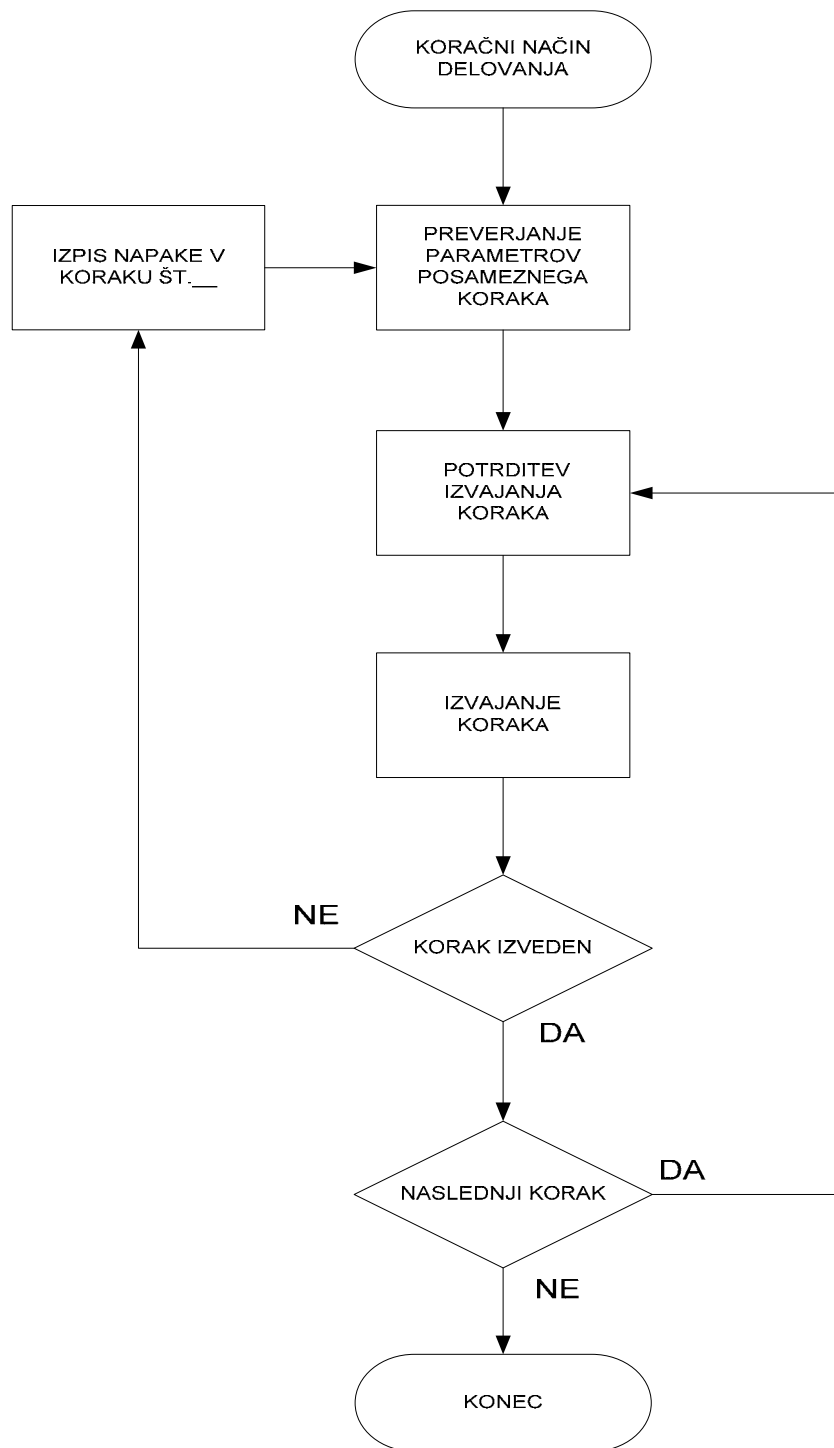
Koračni način delovanja izberemo s preklopnikom in izbiro potrdimo s tipko POTRDITEV.

S tipko NAPREJ poženemo koračni način, pri katerem program za vsak korak (glede na izbran način) preveri, če:

- je vrednost priključenih vhodov na PLK-ju enaka vrednosti referenčnih vhodov na prikazovalniku,
- sta določeni spodnja in zgornja meja za merilno sondo in
- sta določeni najvišja in najnižja temperatura za temperaturno sondo.

V primeru, da vrednosti vhodov ne ustrezajo ali manjka vnos dopustnih mej, se na prikazovalniku izpiše »Preverite parametre v koraku št. __«.

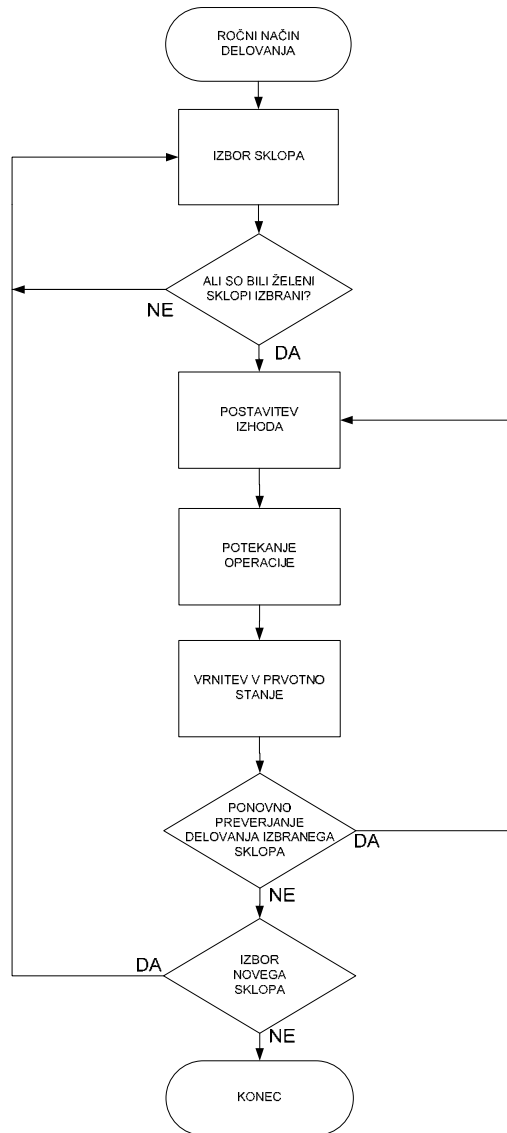
V primeru, da vrednosti vhodov ustrezajo in so vsi podatki vnešeni, korak pa se ni izvedel, se na prikazovalniku izpiše »Mehanska napaka v koraku št. __«. Ta izpis opozarja na odstopanja od dopustnih mej in na morebitni lom ali trk mehanskega dela stroja. Po uspešni izvedbi koraka se lahko pritisne ponovno na tipko NAPREJ za preizkušanje naslednjega koraka ali pa izbere drug način delovanja.



Slika 53: Koračni način delovanja

5.4.3. Ročni način delovanja

Pri ročnem načinu delovanja (slika 54) postavljamo izhode ročno. S tipkama VKLOP in IZKLOP vklapljamo oziroma izklapljamo delovanje mehanskega sklopa.



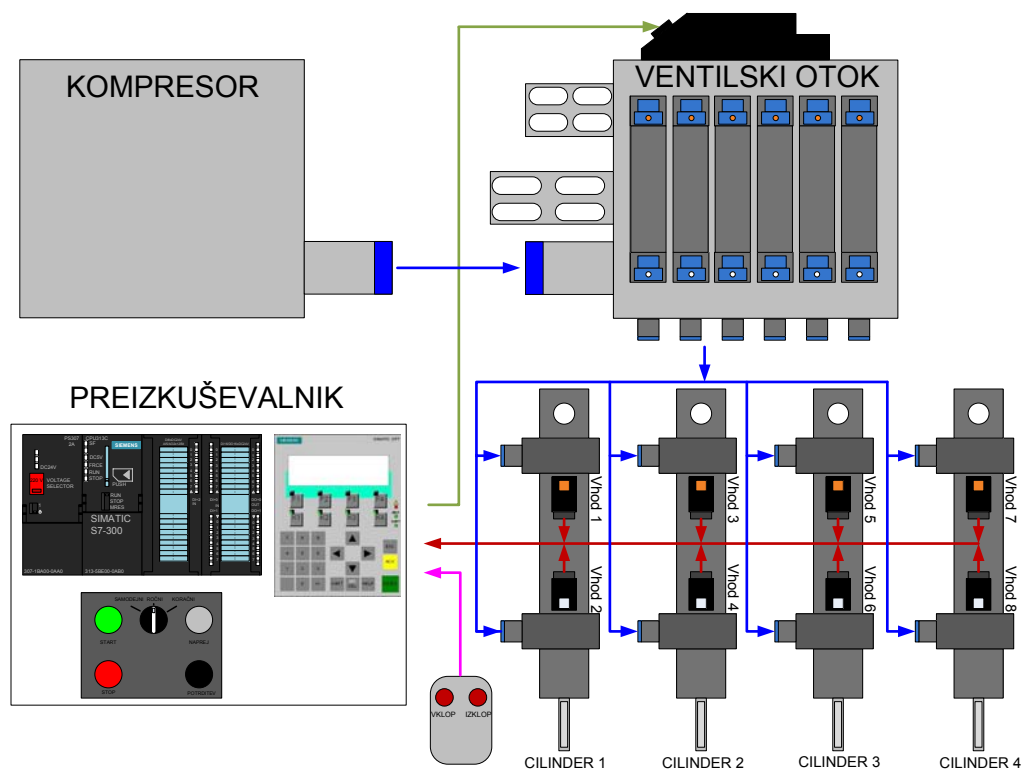
Slika 54: Ročni način delovanja

Ročni način delovanja izberemo s preklopnikom in izbiro potrdimo s tipko POTRDITEV. S pritiskom na tipko VKLOP spravimo izbrani mehanski sklop v delovni položaj, s tipko IZKLOP pa nazaj v prvotno stanje.

6. PREIZKUS DELOVANJA PREIZKUŠEVALNIKA

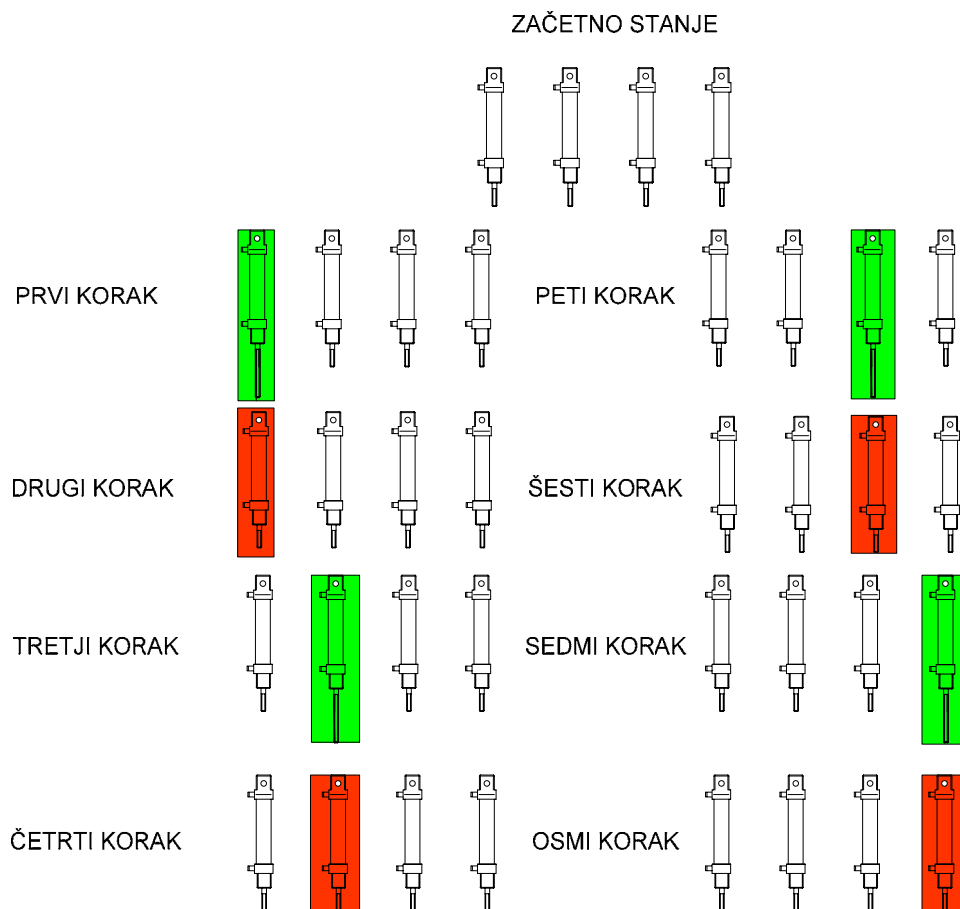
Preizkuševalnik smo testirali na napravi, ki je sestavljena iz kompresorja zraka; ventilskega otoka, ki ga sestavljajo štiri dvostranski ventili in iz štirih pnevmatskih cilindrov z magnetnimi stikali, ki signalizirajo dejanski položaj posamičnega cilindra (slika 55).

Na sliki 55 je z rdečo črto predstavljena informacija o položaju cilindrov, z zeleno črto zahteva po vklopu ali izklopu željenih izhodov, z vijolično črto komanda na daljinskem upravljalcu, z modro pa potek zraka od kompresorja do cilindrov.



Slika 55: Testiranje preizkuševalnika

Pri samodejnem in koračnem načinu se morajo cilindri premikati, kot prikazuje slika 56. Gibanje cilindrov je ponazorjeno z rdečo barvo, ki predstavlja začetni položaj in zeleno barvo, ki predstavlja delovni položaj.



Slika 56: Položaji cilindrov v posameznem koraku

Glede na zahtevano zaporedje premikov nastavimo vrednosti vhodov in izhodov. Vrednosti moramo po postopku za nastavljanje parametrov nastaviti za vsak korak, osnova za nastavitve pa je pomen posameznega vhoda in izhoda. Pomen posameznih vhodov je prikazan v tabeli 1, pomen posameznih izhodov pa v tabeli 2.

Tabela 1: Pomen posameznih vhodov

VHOD	POLOŽAJ CILINDRA
Vhod1	cilinder 1 v začetnem položaju
Vhod2	cilinder 1 v delovnem položaju
Vhod3	cilinder 2 v začetnem položaju
Vhod4	cilinder 2 v delovnem položaju
Vhod5	cilinder 3 v začetnem položaju
Vhod6	cilinder 3 v delovnem položaju
Vhod7	cilinder 4 v začetnem položaju
Vhod8	cilinder 4 v delovnem položaju

Tabela 2: Pomen posameznih izhodov

IZHOD	PREMIK CILINDRA
Izhod 1	premik cilinder 1 v začetni položaj
Izhod 2	premik cilinder 1 v delovni položaj
Izhod 3	premik cilinder 2 v začetni položaj
Izhod 4	premik cilinder 2 v delovni položaj
Izhod 5	premik cilinder 3 v začetni položaj
Izhod 6	premik cilinder 3 v delovni položaj
Izhod 7	premik cilinder 4 v začetni položaj
Izhod 8	premik cilinder 4 v delovni položaj

Postopek kontrole delovanja preizkuševalnika:

- Preverjanje izpisovanja načina delovanja
- Preverjanje delovanja samodejnega načina delovanja
- Preverjanje delovanja koračnega načina delovanja
- Preverjanje ročnega načina delovanja
- Preverjanje delovanja časovnih zakasnitev
- Preverjanje izpisovanja napake pri vnosu na prikazovalniku
- Preverjanje izpisovanja mehanske napake na prikazovalniku
- Preverjanje točnosti ponavljanja gibov mehanskih sklopov z merilno sondo
- Preverjanje delovanja temperaturne sonde

6.1. Preverjanje izpisovanja načina delovanja

Za vsak način delovanja je potrebno preveriti, če izpis na prikazovalniku ustreza izbranemu načinu delovanja.

Iz rezultatov v tabeli 3 je razvidno, da se način delovanja v vseh primerih izpisuje pravilno.

Tabela 3: Rezultati kontrole izpisovanja načina delovanja

POLOŽAJ PREKLOPNIKA	IZPIS NA PRIKAZOVALNIKU	STATUS
Ročni način	Rocni nacin	OK
Koračni način	Koracni nacin	OK
Samodejni način	Samodejni nacin	OK

6.2. Preverjanje delovanja samodejnega načina delovanja

Pri samodejnem načinu delovanja je potrebno preveriti, če se koraki izvajajo v skladu z vnešenimi parametri in če se po zadnjem koraku ponovno izvaja prvi korak. V tabeli 3 so z zeleno barvo pobarvani izhodi, ki pošiljajo cilindre v delovni položaj; z rdečo pa izhodi, ki pošiljajo cilindre v začetni položaj.

Iz rezultatov v tabeli 4 je razvidno, da se položaji cilindrov v posameznem koraku ujemajo s testnim postopkom.

Tabela 4: Rezultati kontrole samodejnega načina delovanja

	KORAK 1	KORAK 2	KORAK 3	KORAK 4	KORAK 5	KORAK 6	KORAK 7	KORAK 8
REFERENČNA VREDNOST VHODA 1	1	0	1	1	1	1	1	1
REFERENČNA VREDNOST VHODA 2	0	1	0	0	0	0	0	0
REFERENČNA VREDNOST VHODA 3	1	1	1	0	1	1	1	1
REFERENČNA VREDNOST VHODA 4	0	0	0	1	0	0	0	0
REFERENČNA VREDNOST VHODA 5	1	1	1	1	1	0	1	1
REFERENČNA VREDNOST VHODA 6	0	0	0	0	0	1	0	0
REFERENČNA VREDNOST VHODA 7	1	1	1	1	1	1	1	0
REFERENČNA VREDNOST VHODA 8	0	0	0	0	0	0	0	1
REFERENČNA VREDNOST IZHODA 1	0	1	1	1	1	1	1	1
REFERENČNA VREDNOST IZHODA 2	1	0	0	0	0	0	0	0
REFERENČNA VREDNOST IZHODA 3	1	1	0	1	1	1	1	1
REFERENČNA VREDNOST IZHODA 4	0	0	1	0	0	0	0	0
REFERENČNA VREDNOST IZHODA 5	1	1	1	1	0	1	1	1
REFERENČNA VREDNOST IZHODA 6	0	0	0	0	1	0	0	0
REFERENČNA VREDNOST IZHODA 7	1	1	1	1	1	1	0	1
REFERENČNA VREDNOST IZHODA 8	0	0	0	0	0	0	1	0
POLOŽAJ CILINDRA 1	DELOVNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ
POLOŽAJ CILINDRA 2	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	DELOVNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ
POLOŽAJ CILINDRA 3	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	DELOVNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ
POLOŽAJ CILINDRA 4	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	ZAČETNI POLOŽAJ	DELOVNI POLOŽAJ	ZAČETNI POLOŽAJ
SAMODEJNO IZVAJANJE KORAKOV	OK	OK	OK	OK	OK	OK	OK	OK
PRAVILNO POLOŽAJ CILINDROV	OK	OK	OK	OK	OK	OK	OK	OK

6.3. Preverjanje delovanja koračnega načina

Pri koračnem načinu delovanja je potrebno preveriti, če se koraki izvajajo ustrezno z vnešenimi parametri, če se posamezni koraki izvajajo vsakič po pritisku na tipko NAPREJ in če se po zadnjem koraku ponovno izvaja prvi korak, ko pritisnemo tipko NAPREJ.

Vnos parametrov v posameznem koraku in položaji cilindrov so enaki kot pri kontroli delovanja samodejnega načina (tabela 3). Rezultati kontrole delovanja tipke NAPREJ so prikazani v tabeli 5.

Tabela 5: Rezultati kontrole delovanja tipke NAPREJ

	KORAK 1	KORAK 2	KORAK 3	KORAK 4	KORAK 5	KORAK 6	KORAK 7	KORAK 8
ZAGON POSAMEZNIH KORAKOV	OK	OK	OK	OK	OK	OK	OK	OK
PRAVILEN POLOŽAJ CILINDROV	OK	OK	OK	OK	OK	OK	OK	OK

6.4. Preverjanje delovanja ročnega načina

Pri ročnem načinu delovanja je potrebno preveriti, če se izhod postavi na logično 1 po pritisku na tipko VKLOP oziroma na logično 0 po pritisku na tipko IZKLOP.

Iz rezultatov v tabeli 6 je razvidno, da se izhodi s pritiskom na tipko VKLOP postavijo na logično vrednost 1, s pritiskom na IZKLOP pa na logično vrednost 0.

Tabela 6: Rezultati kontrole delovanja ročnega načina

REFERENČNI IZHODI	MENI VKLOP	MENI IZKLOP	STANJE IZHODA S TIPKO VKLOP	POLOŽAJ CILINDRA	STANJE IZHODA S TIPKO IZKLOP	POLOŽAJ CILINDRA	STANJE
IZHOD 1	1	0	1	začetni	0	Začetni	OK
IZHOD 2	1	0	1	delovni	0	delovni	OK
IZHOD 3	1	0	1	začetni	0	začetni	OK
IZHOD 4	1	0	1	delovni	0	delovni	OK
IZHOD 5	1	0	1	začetni	0	začetni	OK
IZHOD 6	1	0	1	delovni	0	delovni	OK
IZHOD 7	1	0	1	začetni	0	začetni	OK
IZHOD 8	1	0	1	delovni	0	delovni	OK

6.5. Preverjanje delovanja časovnih zakasnitev

Preveriti je potrebno, če se posamezni koraki izvajajo v skladu z nastavljenimi časovnimi zakasnitvami. Čas časovne zakasnitve je bil dodatno merjen s štoparico.

Pri samodejnem načinu delovanja smo merili čas, ki so ga potrebovali cilindri za preklop iz delovnega v začetni položaj in obratno.

Pri koračnem načinu delovanja smo merili čas, ki ga je potreboval cilindri, da se je postavil v en ali drugi položaj, po pritisku na tipko naprej. V spodnji tabeli so prikazani rezultati meritev. Rezultati so ustrezni v okviru napake štoparice in reakcijskega časa merilca.

Tabela 7: Rezultati kontrole časovnih zakasnitev

	REFERENČNA VREDNOST ČASOVNE ZAKASNITVE	IZMERJENA ČASOVNA ZAKASNITEV PRI SAMODEJNEM NAČINU	IZMERJENA ČASOVNA ZAKASNITEV PRI KORAČNEM NAČINU	STATUS
Korak 1	5 s	4,78 s	5,17 s	OK
Korak 2	5 s	4,91 s	5,41 s	OK
Korak 3	5 s	5,14 s	5,23 s	OK
Korak 4	5 s	5,22 s	4,99 s	OK
Korak 5	5 s	4,69 s	5,02 s	OK
Korak 6	5 s	4,90 s	5,35 s	OK
Korak 7	5 s	5,07 s	5,13 s	OK
Korak 8	5 s	4,96 s	5,58 s	OK

6.6. Preverjanje izpisovanja mehanske napake

Preveriti je potrebno, če sistem uspešno detektira mehanske napake in jih izpisuje na prikazovalniku.

Postopek: Mehansko napako generiramo (simuliramo) z izklapljanjem kompresorja v posameznih korakih in ugotovljamo, če prikazovalnik vsakič opozori na mehansko napako.

Iz rezultatov v tabeli 8 je razvidno, da se mehanske napake v posameznem koraku pravilno prikazujejo.

Tabela 8: Rezultati kontrole izpisovanja mehanske napake

IZKLOP KOMPRESORJA	PRIKAZ NA PRIKAZOVALNIKU	STATUS
Korak 1	Mehanska napaka v koraku st.1	OK
Korak 2	Mehanska napaka v koraku st.2	OK
Korak 3	Mehanska napaka v koraku st.3	OK
Korak 4	Mehanska napaka v koraku st.4	OK

Korak 5	Mehanska napaka v koraku st.5	OK
Korak 6	Mehanska napaka v koraku st.6	OK
Korak 7	Mehanska napaka v koraku st.7	OK
Korak 8	Mehanska napaka v koraku st.8	OK

6.7. Preverjanje izpisovanja napake pri vnosu parametrov

Preveriti je treba, če sistem uspešno detektira napake pri vnosu parametrov in jih izpisuje na prikazovalniku.

Postopek: Napako pri vnosu generiramo (simuliramo) tako, da pri vsakem koraku vpišemo napačno vrednost enega vhoda in ugotavljamo, če prikazovalnik vsakič javi napako pri vnosu.

Iz rezultatov v tabeli 9 je razvidno, da se napake pri vnosu v posameznem poraku pravilno prikazujejo.

Tabela 9: Rezultati kontrole izpisovanja napake pri vnosu parametrov

	GENERIRANJE NAPAK	DEJANSKI PRIKAZ NA PRIKAZOVALNIKU	STATUS
Korak 1	Korak 1 – vhod 1 postavimo na logično 0 namesto na 1	Preverite parametre v koraku st.1	OK
Korak 2	Korak 2 – vhod 2 postavimo na logično 0 namesto na 1	Preverite parametre v koraku st.2	OK
Korak 3	Korak 3 – vhod 3 postavimo na logično 0 namesto na 1	Preverite parametre v koraku st.3	OK
Korak 4	Korak 4 – vhod 4 postavimo na logično 0 namesto na 1	Preverite parametre v koraku st.4	OK
Korak 5	Korak 5 – vhod 5 postavimo na logično 0 namesto na 1	Preverite parametre v koraku st.5	OK
Korak 6	Korak 6 – vhod 6 postavimo na logično 0 namesto na 1	Preverite parametre v koraku st.6	OK
Korak 7	Korak 7 – vhod 7 postavimo na logično 0 namesto na 1	Preverite parametre v koraku st.7	OK
Korak 8	Korak 8 – vhod 8 postavimo na logično 0 namesto na 1	Preverite parametre v koraku st.8	OK

6.8. Preverjanje točnosti ponavljanja gibov mehanskih sklopov z merilno sondo

Preveriti je treba natančnost ponovljivosti giba mehanskega sklopa (cilindra) z merilno sondo. Cilinder smo postavili v delovni položaj. Merilno sondo smo postavili k cilindru tako, da je kazala 2,3 mm. Potem smo preverili, ali merilna sonda po vsaki ponovitvi giba cilindra pokaže enako vrednost. Rezultati meritev so prikazani v tabeli 10.

Tabela 10: Rezultati merjenja ponavljivosti giba cilindra z merilno sondo

PONOVITEV PREMIKA CILINDRA	POLOŽAJ CILINDRA	STANJE
1	2,3 mm	OK
2	2,3 mm	OK
3	2,3 mm	OK
4	2,3 mm	OK
5	2,3 mm	OK

6.9. Preverjanje delovanja temperaturne sonde

Preveriti je treba, če so meritve temperature s temperaturno sondo natančne. Meritve preverjamo z ustreznim digitalnim termometrom. Postopek: Merilno sondo in digitalni termometer smo postavili v lonček vode, ki smo ga segrevali na štedilniku. Vsake toliko časa smo izpisali temperaturi, ki sta jih kazala oba instrumenta. Rezultati meritev, prikazani v spodnji tabeli, vključujejo merilne napake temperaturne sonde.

Tabela 11: Rezultati kontrole delovanje temperaturne sonde

DEJANSKA TEMPERATURA MERJENCA	MERJENA TEMPERATURA S SONDO	STANJE
24,3 °C	24,7 °C	OK
33,5 °C	33,3 °C	OK
42,8 °C	43,2 °C	OK
55,5 °C	55,1 °C	OK
64,8 °C	65,2 °C	OK
77,1 °C	77,6 °C	OK
83,2 °C	83,5 °C	OK
95,3 °C	94,6 °C	OK
98,4 °C	97,9 °C	OK

7. ZAKLJUČEK

Podjetje, ki je naročnik preizkuševalnika ima dolgoletne izkušnje na področju izdelovanja in rekonstruiranja strojev. V tem času so se napake in nevšečnosti odkrivale v zadnji fazi, ko je bil stroj že izdelan. Takšen način dela je povzročal dodatne stroške materiala, energije, delovne sile in celo zamude rokov izvedbe projektov. Sčasoma je naročnik prišel do spoznanja, da je potrebno te nevšečnosti odpraviti v čim zgodnejši fazi izdelave sestavnih delov.

Tako je nastala ideja o napravi – preizkuševalniku, s katero bi se testirali posamezni mehanski sklopi strojev že med izdelovanjem stroja. Naročnik je na osnovi svojih izkušenj izdelalo zahteve, kako naj naprava deluje. V podjetju smo poiskali ustrezno strojno opremo. Odločili smo se za Siemensov programabilni logični krmilnik S7-313C, ki razpolaga s 24 digitalnimi vhodi, 16 digitalnimi izhodi ter z 2 analognima vhodoma in izhodoma. Za prikazovanje teksta smo izbrali štirivrstični OP7-prikazovalnik. Kot programsko orodje smo uporabili Simensov STEP 7.

V okviru diplomske naloge smo izdelali prototip preizkuševalnika in zanj napisali aplikativni program vključno z uporabniškim vmesnikom. Program omogoča preizkušanje mehanskih sklopov stroja v ročnem, koračnem in samodejnem načinu delovanja. Ker lahko s pomočjo uporabniškega vmesnika parametriramo poljubne vhodno-izhodne kombinacije, ga lahko uporabljamo na različnih napravah. Preizkuševalnik z uporabo merilnih sond in analognih vhodov krmilnika omogoča tudi merjenje temperature in dolžine. Delovanje preizkuševalnika smo preverjali že v fazi programiranja. Na koncu smo izdelali testno napravo s štirimi pnevmatskimi cilindri, na kateri smo preverili delovanje vseh preizkuševalnikovih funkcij.

Preizkuševalnik je odprt za nadaljnjo izpopolnjevanje in nadgradnjo. Z močnejšim krmilnikom, ki bi imel tudi več vhodno-izhodnih točk in bi omogočal tudi Profibus komunikacijski protokol, bi se lahko lotili tudi testiranja kompleksnejših strojev in naprav, kot so servomotorji, ventili, ki obratujejo le preko profibusa. Za obsežnejša testiranja bi bilo potrebno nadgraditi tudi uporabniški vmesnik, bodisi z močnejšim grafičnim prikazovalnikom bodisi pa s SCADA sistemom. To bi omogočilo večjo

preglednost pri nastavljanju parametrov v posameznih korakih ali celo izdelavo receptur za posamezne sklope.

8. LITERATURA

S. Strmčnik (1998). Celostni pristop k računalniškemu vodenju procesov. 1. izdaja. Ljubljana: Fakulteta za elektrotehniko.

Priročnik Siemens (1999). Simatic HMI OP7, OP17 Operator panel.

Priročnik Siemens (2001). S7-300 Programmable Controller CPU Specifications, CPUs 312C to 314C-2DP/PtP.

Ambrožič V., Nedeljković D. (2002). Digitalna krmilja in teorija regulacij. Ljubljana: Laboratorij za regulacijsko tehniko in močnostno elektroniko, Fakulteta za elektrotehniko.

Siemens automation group (2004). Sistemski tečaj ST-7PRO.

Your personal PLC tutor site. Pridobljeno 27. 3. 2006 s svetovnega spleta: <http://www.plcs.net/>.

Platinum resistance thermometers. Pridobljeno 2. 2. 2007 s svetovnega spleta: <http://www.picotech.com/applications/pt100.html>

L.A. Bryan, E.A. Bryan (1997). Programmable Controllers Theory and Implementation Second Edition.

Priročnik Festo (2005). Valve terminal type 10 CPV, Compact Performance.

Pico Technology (2007). Pridobljeno 10. 1. 2007 s svetovnega spleta: <http://www.picotech.com/applications/pt100.html>

Contelec (2007). Pridobljeno 20. 1. 2007 s svetovnega spleta: <http://www.contelec.de/>

Edison Instruments Automation (2007). Pridobljeno 25. 1. 2007 s svetovnega spleta: http://ia.edisonautomation.com/software/siemens_software_step7.html

General Motors (2007). Pridobljeno 10. 1. 2007 s svetovnega spleta: <http://www.gm.com/>

Introduction to Digital Recording Techniques(1989-2005). Pridobljeno 16. 2. 2007 s svetovnega spleta: <http://www.digital-recordings.com/publ/pubrec.html>