ŠOLSKI CENTER PTUJ VIŠJA STROKOVNA ŠOLA

Jure Erman

# PREDSTAVITEV PLK OMRON ZEN IN NAVODILA ZA UPORABO V SLOVENŠČINI

Diplomska naloga

Ptuj, junij 2011



# PREDSTAVITEV PLK OMRON ZEN IN NAVODILA ZA UPORABO V SLOVENŠČINI

Diplomska naloga višješolskega študijskega programa

Študent:	Jure Erman
Študijski program:	Mehatronika
Modul:	Programiranje v avtomatiki
Mentor - predavatelj:	Slavko Murko, univ. dipl. inž.
Somentor - VSŠ:	Bojan Brečko, univ. dipl. inž. el.

Ptuj, junij 2011



Šolski center Ptuj Višja strokovna šola Volkmerjeva cesta 19 2250 Ptuj Slovenija 2 +386 (0)2 787 18 12 vss.scptuj.si

vss@scptuj.si

Številka: M1-47 Datum: 1.6.2011

#### SKLEP O DIPLOMSKI NALOGI

- 1. Jure ERMAN, študent višješolskega strokovnega programa mehatronika, izpolnjuje pogoje za izdelavo diplomske naloge.
- 2. Tema diplomskega naloge se nanaša na predmet Programiranje v avtomatiki.

Mentor predavatelj: Somentor: mag. Slavko Murko, univ. dipl. inž. el. Bojan Brečko, univ. dipl. inž. el.

3. Naslov diplomske naloge

## PREDSTAVITEV PLK OMRON ZEN IN NAVODILA ZA UPORABO V SLOVENŠČINI

#### 4. Vsebina diplomske naloge:

Spoznati uporabo krmilnika OMRON ZEN. Napisati navodila za uporabo z učnimi primeri za OMRON ZEN. Predstaviti rešitve ter preizkusiti primere.

Diplomsko nalogo izdelajte v skladu z "Navodili za izdelavo diplomske naloge" in "Pravilnikom o pripravi diplomske naloge". Dva trdo vezana izvoda, ter izvod v elektronski obliki, oddajte v referat VSŠ Ptuj. En izvod trdo vezane diplomske naloge oddajte mentorju v podjetju.

Mentor predavatelj mag. Slavko Murko, univ. dipl. inž. el.

JZ Šolski center Ptuj je vpisan pri Okrožnem sodišču Ptuj, vložna številka 10070500, matična številka 5064678000, davčna številka 23369809

Obrazec: VSŠ -P8

## ZAHVALA

Zahvaljujem se mentorju prof. univ. dipl. inž. Slavku Murku in somentorju prof. univ. dipl. inž. el. Bojanu Brečku za strokovne nasvete, pomoč in vodenje pri opravljanju diplomske naloge.

Posebna zahvala velja tudi staršem za spodbudo in podporo v času mojega šolanja.

## PREDSTAVITEV PLK OMRON ZEN IN NAVODILA ZA UPORABO V SLOVENŠČINI

Ključne besede:	Omron Zen, PI	LK,	programiranje,	avtomatizacija,	logična	funkcija,
	lestvični diagran	n				

UDK: 681.527.7:004.4(075.8) 681.5(075.3)

#### **Povzetek:**

Diplomsko delo predstavlja industrijski programirljivi logični krmilnik (PLK) Omron Zen in zajema navodila za uporabo krmilnika v avtomatiki, podprtega s praktičnim primerom koračnega in kombinacijskega (logičnega) krmilja.

V osnovi se PLK Omron Zen lahko uporablja za domače namene (npr. za nadzor sistema osvetlitve, za kontrolo nivoja rezervoarja vode ali za spremljanje temperature v vrtnem rastlinjaku), kot tudi drugod (npr. v nakupovalnih centrih lahko avtomatsko upravljamo tekoče stopnice, različne avtomate - kavomat, ročna avtopralnica, inp.). PLK Omron Zen je idealen tip krmilnika, s katerim lahko naredimo začetne prve korake (lestvičnega) programiranja v avtomatizaciji, saj je enostaven za uporabo in je tako primeren za učenje. Namenjen je reševanju krmilnih problemov manjšega obsega, ima možnost nadgradnje vhodno/izhodnih enot oz. razširitve vhodov (do 24) in izhodov (do 20), nima pa možnosti priključitve na industrijsko podatkovno omrežje. Programiramo lahko s pomočjo osebnega računalnika (s programsko opremo ZEN), kot tudi interno na krmilnik. Prednost ima programiranje s pomočjo računalnika, kajti programska oprema ponuja funkcijo simulacije, ki nam lahko programiranje poenostavi ter je bolj pregledno, obenem pa je glavni namen simulacije ta, da program simulacijsko preizkusimo in tako lažje odkrijemo oz. odpravimo morebitne težave, napake.

Delo je razdeljeno na dva sklopa. V prvem delu so podane osnove splošne teorije industrijskih krmilnikov. Drugi del pa se osredotoča na predstavitev PLK Omron Zen ter njegovo praktično uporabo.

## PRESENTATION OF PLC OMRON ZEN AND INSTRUCTION MANUAL IN SLOVENIAN LANGUAGE

Key words: Omron Zen, PLC, programming, automation, logical function, ladder diagram

UDK: 681.527.7:004.4(075.8) 681.5(075.3)

#### Abstract:

The diploma paper presents the industrial programmable logic controller (PLC) Omron Zen and contains the instructions for the use of the controller in automatics, supported with the practical case of the stepper and combinational (logical) control.

Basically, the PLC Omron Zen can be used for domestic purposes (for example: for the control of the lightening system, for the control of the water tank level or for monitoring the temperature in the greenhouse) as well as elsewhere (for example: in shopping centers we can steer the escalator automatically, different vending machines - coffee machines, self-service car wash, etc.). The PLC Omron Zen is the ideal type of a controller for the first steps (scale) of programing in automation, as it is easy to use and therefore appropriate for learning. It is intended for the solvation of control problems of a smaller range, it has got an option for upgrading of the input/output units or the expansion of inputs (up to 24) and outputs (up to 20), and it does not have an option for the connection to the industrial data network. We can program with the help of a personal computer (with the program equipment ZEN) as well as internally on the controller. Programing with the help of a computer is prior, as the program equipment offers the simulation function that can simplify the programing and it is more transparent; at the same time the purpose of the simulation is to test the program as a simulation and so to discover or eliminate the eventual problems, errors.

The diploma paper is divided into two assemblies. The basic general theories of the industrial controllers are given in the first part. The second part focuses on the presentation of the PLC Omron Zen and its practical use.

## KAZALO

1.	UVOD	1
1.1	Zgodovina programirljivih krmilnikov	1
1.2	Standard SIST EN 61131	2
2.	PROGRAMIRLJIVI LOGIČNI KRMILNIK	3
2.1	Osnovna predstavitev	
2.2	Zgradba in delovanje	4
2.2.1	Kompaktni krmilniki	10
2.2.2	Modularni krmilniki	11
3.	OSNOVNE LOGIČNE FUNKCIJE	13
3.1	Logične povezave med signali	
3.1.1	Logična funkcija NE (negacija)	15
3.1.2	Logična funkcija IN (konjunkcija)	15
3.1.3	Logična funkcija ALI (disjunkcija)	16
3.2	Pomnilne funkcije	16
3.2.1	RS pomnilna celica	16
3.3	Časovne funkcije	18
3.3.1	Zakasnitev vklopa	20
3.3.2	Zakasnitev izklopa	
3.4	Samodržna vezava	22
4.	PROGRAMIRANJE	25
4.1	Osnovni načini programiranja	
4.1.1	Lestvični diagram (LAD)	
4.1.2	Funkcijski blokovni diagram (FBD)	
4.1.3	Seznam ukazov (STL)	
5.	PLK OMRON ZEN	
5.1	Predstavitev krmilnika Omron Zen	
5.2	Programiranje s pomočjo programske opreme ZEN	35
5.2.1	Programska oprema ZEN	
5.2.2	Interno programiranje	52
5.2.3	Uporaba časovnikov, števcev in ostalih funkcij	61

5.3	Priključitev krmilnika Omron Zen	73
5.3.1	Vhodne povezave	73
5.3.1.1	Analogni vhodi	76
5.3.2	Izhodne povezave	78
5.3.2.1	Enote s tranzistorskim izhodom	78
5.3.2.2	Enote z relejskim izhodom	79
6.	PRAKTIČNA PRIMERA ZA UČENJE UPORABE PLK OMRON ZEN	81
6.1	Primer koračnega krmilja	83
6.2	Primer kombinacijskega krmilja	90
7.	ZAKLJUČEK	93
8.	IZJAVA AVTORJA	95
9.	LITERATURA IN VIRI	97
10.	PRILOGE	99
10.1	Seznam slik	99
10.2	Seznam preglednic	. 102
10.3	Ostale priloge	. 103

## **UPORABLJENI SIMBOLI, KRATICE**

- A/D Analogno-digitalni (pretvornik)
- CPE Centralna procesna enota
- D/A Digitalno-analogni (pretvornik)

EEPROM - Electricaly Programmable Read Only Memory (Električno zbrisljiv in programirljiv bralni pomnilnik)

I - Izhod

- IEC International Electrotechnical Commision (Mednarodna elektrotehniška komisija)
- PLK Programirljivi logični krmilnik
- RAM Random Access Memory (Bralno-pisalni pomnilnik)
- ROM Read Only Memory (Bralni pomnilnik)
- V Vhod
- VSŠ Višja strokovna šola
- ZOI Zajem obdelava izdaja

## 1. UVOD

## 1.1 Zgodovina programirljivih krmilnikov

Programirljivi krmilniki so se pojavili proti koncu šestdesetih let 20. stoletja. Podjetje General Motors je leta 1968 sestavilo kriterije oz. merila za programirljive krmilnike. Njihov glavni cilj nastajajoče generacije programirljivih krmilnikov je bil zmanjšati visoke cene, ki so spremljale tedanje relejsko-krmilne sisteme. Tehnologija je napredovala in tako tudi potreba po stabilnem sistemu s pomočjo računalnika, ki bi naj zadovoljila potrebe (avtomobilske) industrije in bi bila enostavna za programiranje ter vzdrževanje [6].

Kriteriji za programirljive krmilnike so vsebovali naslednje zahteve, po katerih bi:

- bili cenovno ugodnejši od relejnih sistemov;
- imeli enostavno nadomestljive vhodno-izhodne enote;
- bili sposobni delovati v industrijskem okolju (vibracije, visoke temperature ipd.);
- bili modularni, kar bi omogočilo enostavno zamenjavo ali popravilo podsklopov;
- imeli možnost reprogramiranja (zaradi uporabe pri različnih nalogah);
- bili enostavno programirljivi in razumljivi tehničnemu osebju.

Prvi uspešen programirljivi krmilnik je bil Modicon 084, podjetja Bedford Associates, saj je bil med prvimi, ki je imel možnost programiranja z uporabo lestvičnega diagrama [2].

Proces ožičenja so tako poenostavili s programirljivimi krmilniki, kar je pripomoglo k poenostavitvi proizvodnje in industrije. Vgradnja je bila enostavnejša, saj so zavzemali manj prostora in za razliko od relejev se jih je dalo ponovno uporabiti pri novih projektih.

Tudi v današnjih časih še vedno veljajo iste zahteve krmilnih sistemov oz. prvi kriteriji za programirljive krmilnike kot osnovno vodilo pri razvoju krmilnikov, čeprav je prišlo do radikalnih sprememb v načinu in filozofiji uporabe krmilij, ki so sledile industrijskemu razvoju.

Relejno-krmilne sisteme oz. elektromehanske elemente (releji) v sedanji praksi še vedno uporabljamo predvsem kot enote za galvansko ločitev med krmilnim (vhodi) in močnostnim delom (izhodi).

#### 1.2 Standard SIST EN 61131

V začetku nastajanja prvih programirljivih krmilnikov je na področju standardov in predpisov vladala precejšnja zmeda med proizvajalci, kar je pripomoglo k potrebi oz. uvedbi standardov.

Mednarodna skupina strokovnjakov in proizvajalci krmilnikov, programske opreme so se v okvirju IEC prvič sestali leta 1979. Pod oznako IEC 1131 sta leta 1990 izšla prva dva zvezka novega standarda z osnovnimi smernicami s področja programirljivih krmilnikov. V Sloveniji poznamo omenjeni standard pod oznako SIST EN 61131 [2]. Standard EN 61131 je razdeljen na več posameznih enot, kar prikazuje tabela 1.

Št. standarda	Opis
61131-1	Splošne informacije
61131-2	Aparaturne zahteve in preizkušanje
61131-3	Programski jeziki
61131-4	Vodnik za uporabnika
61131-5	Komunikacija
61131-7	Programiranje krmiljenja z mehko (angl. fuzzy) logiko
61131-8	Vodnik za uporabo in izvedbo programskih jezikov

Tabela 1: Enote standarda EN 61131 [2]

EN 61131-3 je standard, s katerim definiramo programski jezik za programiranje PLK-jev. Uporaba tega standarda v industriji je posledica zahtev po avtomatizaciji in ima pozitiven vpliv na življenjsko dobo programske opreme, ki vključuje zahteve po analizi, načrtovanju, izdelavi, potrditvi, namestitvi, delovanju in vzdrževanju. Standard EN 61131-3 zagotavlja podporo več programskim jezikom za PLK: lestvični diagram (LD), funkcijski blokovni diagram (FBD), sekvenčni funkcijski diagram (SFD), strukturiran tekst (ST) in seznam navodil (IL) [7].

## 2. PROGRAMIRLJIVI LOGIČNI KRMILNIK

#### 2.1 Osnovna predstavitev

Programirljivi logični krmilnik (PLK) je digitalni računalnik namenjen avtomatizaciji elektromehanskih procesov. Vsebuje večino enot, ki jih srečamo pri osebnih računalnikih vendar je zasnovan za druge vrste opravil. PLK se uporablja v številnih industrijah za upravljanje, krmiljenje in nadzorovanje strojev kot tudi za druge poslovne namene.

PLK je mikroprocesorsko krmiljen sistem, program imamo shranjen v pomnilniku in ga lahko tudi vedno spremenimo ter tako vplivamo na medsebojne povezave vhodov in izhodov [1]. Krmilnik deluje po principu zajem - obdelava - izdaja (ZOI) (slika 1).



Slika 1 - Princip ZOI (prirejeno po [1])

Osnovni cilj krmilnika je, da je enostaven za uporabo, vzdrževanje (odkrivanje napak) in programiranje z različnimi programskimi jeziki.

## 2.2 Zgradba in delovanje

PLK je mikroračunalnik namenjen avtomatiziranim procesom. Sestavljen je iz centralno procesne enote, vhodne enote, izhodne enote in napajalnega dela. Na podlagi ukazov, shranjenih v pomnilniku, izvaja PLK različne logične, sekvenčne, časovne in aritmetične operacije ter tako upravlja z različnimi napravami, procesi preko digitalnih in analognih vhodov/izhodov [8].

• Centralna procesna enota predstavlja "možgane" računalnika oz. z drugimi besedami bi lahko rekli, da je CPE osrednji in najpomembnejši del PLK-ja. CPE služi za obdelavo podatkov (uporabniškega programa) ter skrbi za izvajanje različnih matematičnih in logičnih funkcij [2]. V centralni procesni enoti se nahaja:

- mikroprocesor, ki skrbi za izvajanje ukazov uporabniškega programa ter sinhronizirano nadzira delovanje ostalih enot (vhodne in izhodne enote);
- različni tipi pomnilnika (programski, delovni, sistemski);
- sistemsko vodilo, ki jih povezuje.

Uporabniški program napisan po zahtevah uporabnika se nahaja v trajnem (ROM) in delovnem (RAM) pomnilniku, kar prikazuje slika 2.



Slika 2 - Primer prenosa uporabniškega programa v pomnilnik PLK-ja (prevedeno in prirejeno po [8])

Programski (bralni) pomnilnik je najpomembnejši tip pomnilnika, saj se v njem nahaja celoten uporabniški program, kot tudi konfiguracija in parametri modulov krmilnika. V bralni pomnilnik (ROM) med obratovanjem krmilnika ne moremo vpisovati, spreminjati ali dodajati vsebine programa. Pri sodobnih krmilnikih imamo pogosto v praksi največ opravka z EEPROM tipom programskega pomnilnika. Izveđen je lahko integrirano ali pa preko pomnilniške kartice (*angl. Memory Card*) vstavljene v razširitveno režo. Različica EEPROM pomnilnika nam omogoča hitro in sprotno spreminjanje vsebine.

Naslednji tip pomnilnika, ki se nahaja v CPE je delovni oz. bralno-pisalni pomnilnik (RAM). Vsebuje dele uporabniškega programa - to sta programska koda in uporabniški podatki, ki se med procesom izvajajo in se lahko spreminjajo. Uporabniku omogoča poljubno spreminjanje vsebine programa (tj. vpisovanje, branje, brisanje). Pomanjkljivost RAM-a je potreba po konstantnem napajanju, kajti ob izpadu električne napetosti bi se vsebina izbrisala. Lahko pa imamo vgrajeno pomožno baterijo, ki bi poskrbela, da se vsebina nebi izbrisala.

Zadnji tip pomnilnika, ki sestavlja CPE je sistemski pomnilnik, ki se nahaja v območju delovnega pomnilnika. Vsebuje spremenljivke oz. operande, ki jih uporabniški program bere in so razporejeni v naslednje skupine:

- vhodi (angl. Input I): programska preslikava digitalnih vhodov;
- izhodi (*angl. Output Q*): programska preslikava digitalnih izhodov;
- zastavice (*angl. Flag*<sup>\*</sup> *F*): pomožne spremenljivke oz. pomožni pomnilniki, ki so dostopni iz celotnega programa (pogl. 5.3.2);
- časovne funkcije (*angl. Timer T*): časovniki za realizacijo nadzornih časov, časovnih zakasnitev (pogl. 3.3);
- števne funkcije (*angl. Counter C*): programski števci za štetje navzgor ali navzdol (pogl. 5.3.2);
- začasni lokalni podatki služijo za dinamično shranjevanje vmesnih podatkov [1].

V sistemskem pomnilniku se poleg naštetih (remanentnih) spremenljivk nahajajo še računske enote, enote za primerjalne funkcije, funkcije za pomik podatkovnih blokov, pomnilne funkcije, enote za nadzor, komunikacijske enote in regulacijske enote [8].

• Vhodne in izhodne enote služijo za povezavo med PLK (CPE) in zunanjimi (perifernimi) napravami, kar prikazuje slika 3. Glavni namen vhodno-izhodnega sistema je pretvoriti signale, ki so bili sprejeti od zunanjih enot ali poslani vanje, v odgovarjajočo obliko.

Vhodne in izhodne enote opravljajo naslednje naloge:

- prilagajanje napetostnih in tokovnih nivojev vhodno-izhodnih signalov internim napetostnim nivojem v krmilniku (digitalne enote);
- pretvorba neke zunanje napetosti ali toka v digitalni podatek, ki ga lahko PLK obdela, in obratno (analogne enote);
- zaščita pred motilnimi signali iz industrijskega okolja z galvansko ločitvijo krmilnega (notranji/sistemski del) in procesnega (zunanji del) tokokroga;
- zaščita pred kratkim stikom (morebiten kratek stik ne sme uničiti krmilnika) [2].



Slika 3 - Vhodne in izhodne enote PLK-ja (prevedeno po [10])

Pri krmiljenju avtomatiziranih procesov imamo opravka z dvema oblikama signala. Poznamo digitalne in analogne vhodno-izhodne sisteme.

**Digitalna oblika vhodno-izhodnih sistemov** pride v poštev takrat, ko imamo preko digitalnih enot fizično povezavo med CPE in zunanjimi napravami, ki prenašajo in sprejemajo digitalne signale<sup>\*</sup>. Vhodne signale iz procesa posredujejo različni (priključeni) senzorji in inkrementalni dajalniki, kot so tipkala, mejna stikala, končna stikala, kontakti, releji. Napetosti, ki pridejo iz teh dajalnikov so lahko različne, kar je razvidno iz tabele 2.

	Napetost	Low - logična "0"	High - logična "1"
Enosmerna	24 V	0 do 5 V	11 do 32 V
	110 V	0 do 35 V	88 do 127 V
Izmenična	115 V	0 do 40 V	85 do 135 V
	230 V	0 do 70 V	176 do 230 V

Tabela 2: Glede na napetosti, ki pridejo iz dajalnikov so definirani logični nivoji [8]

PLK dobi preko vhodne digitalne enote informacijo o stanju naprave, ki se v krmilnem programu ustrezno obdela in uporabi za krmiljenje naprav. Digitalni izhodi so namenjeni za prenos in oblikovanje signala do krmilnih naprav. Ti signali pridejo iz CPE in so navadno TTL<sup>\*</sup> nivoja (5 V napajanje). S pomočjo različnih elektronskih vezij (tranzistorski ojačevalnik, releji, triaki) lahko dosežemo signale energijsko višjih nivojev. Upravljamo lahko z izvršilnimi členi, katerih tokovna obremenitev je v območju od nekaj mA (npr. 500 mA) do nekaj A. Na izhodne enote krmilnika priključimo krmilne naprave oz. aktuatorje, kot so koračni motorji, krmilni releji, elektromagnetni ventili, kontrolne lučke.

Nove tehnologije in aplikacije prinašajo programsko krmilnim procesom tudi takšne zahteve, ki jih digitalni vhodno-izhodni sistemi ne morejo izpolniti (tam, kjer za analiziranje signalov nista dovolj le dve stanji).

Rešitev je v **analognih vhodno-izhodnih sistemih**, ki lahko registrirajo analogne (zvezne) signale, kot npr. spreminjanje napetosti ali temperature (slika 4). Analogni vhodno-izhodni signali so tokovni ali napetostni. Krmilniki podpirajo več standardnih območij za analogne vhodno-izhodne signale (odvisno od proizvajalca): pri tokovnih analognih signalih se najpogosteje uporablja območje od 4 do 20 mA, pri napetostnih analognih signalih pa območje od 0 do 10 V.



Slika 4 - Analogni signal [Vir: lasten]

<sup>\*</sup> TTL standard določa nazivno napetost 0 V za signal "0" in 5 V za signal "1".

PLK razume le binarni stanji "1" in "0", zato ne more interpretirati analognih signalov, ki imajo neskončno število stanj. Zato analogne enote dospele analogne signale pretvorijo v digitalne vrednosti s pomočjo A/D pretvornikov. Ti so pri krmilnikih predvideni za zajemanje počasi se spreminjajočih veličin (npr. temperaturni signal se stalno spreminja za neskončno majhne vrednosti). Analogne izhodne enote se uporabljajo za posredovanje analognih signalov v proces, ki zahteva nadzor zunanjih naprav, le-te pa se odzivajo na (zvezno) spreminjanje napetosti in toka. Med analogne izhode uvrščamo analogne ventile, analogne števce, tlačne pretvornike, frekvenčne pretvornike. Za krmiljenje analognih naprav je potrebno digitalni podatek pretvoriti v analognega. To storimo z D/A pretvornikom, ki vsakemu digitalnemu številu pripiše neko vrednost napetosti ali toka [6].

Za zanesljivo razlikovanje vhodnih stanj ("0" in "1") in predvsem zaradi zaščite CPE ima vsak digitalna vhodna enota (slika 5):

- filter, ki izloča motnje;
- optični spojnik za galvansko ločitev (izloča motnje in predstavlja povezavo s CPE);
- pragovno preklopno stikalo skrbi za pravilno obliko binarnega signala za CPE.



Slika 5 - Vhodna enota PLK [Vir: lasten]

Za zaščito pred napačno priključitvijo napetosti imajo vhodne enote vgrajene diode ter usmerniška vezja za izmenične vhodne signale. CPE vključuje/izključuje aktuatorje in izvršilne člene (npr. ventile). V izhodni enoti CPE pred uničenjem ščitijo spojniki (galvanska ločitev med krmilnikom in procesom), tako da morebiten kratek stik v izvršilnem členu nebi mogel uničiti krmilnika [1].

Vhodom in izhodom so dodeljena ustrezna vmesna pomnilniška področja - procesne preslikave. V tem delu pomnilnika se nahajajo slike stanj vhodnih in izhodnih enot. Na začetku programskega cikla (pred programsko obdelavo) preslika CPE stanje iz vhodnih enot (vhodni signali) v procesne preslikave vhodov. Tako pridejo vhodni signali v CPE krmilnika. Ukazi tako dostopajo do vhodnih slik (dejanskih vhodov), zajetih na začetku cikla. Enako velja tudi za izhodne signale. Če se v nekem ukazu nek binarni izhod postavi v določeno stanje (pravimo, da se postavi določen bit v enoti), to stanje CPE vpiše v procesno preslikavo izhodov. Iz področja procesnih preslikav izhodov se stanja prenesejo na izhodne enote, ki se povezane s procesom. Do osvežitve izhodov in preslikave stanj iz procesne preslikave izhodov na izhode (Q) pride šele na koncu programskega cikla. Koncept procesnih preslikav ima številne prednosti, kar nam je v pomoč zlasti pri programiranju, vendar ne smemo pozabiti, da vedno, kadar naslavljamo "vhode" in "izhode", pravzaprav naslavljamo slike "vhodov" in "izhodov" [1, 2].

• Vodila so namenjena pretoku podatkov, naslovov ter krmilnih signalov med internimi enotami PLK-ja in kompatibilnimi zunanjimi enotami (slika 6).



Slika 6 - Vodila PLK-ja [2]

Napajalni del: PLK je običajno priključen (odvisno od proizvajalca) na enosmerno napetost 24 V. V ta namen se v napajalni enoti nahaja napajalnik, ki je priključen na zunanjo izmenično napetost 120 V ali 230 V (vrednost nastavimo na napajalniku). Napajalnik zagotavlja električno napetost za pravilno delovanje vseh enot.

## 2.2.1 Kompaktni krmilniki

Kompaktni krmilniki združujejo vse komponente v enem ohišju: napajalna enota, CPE, vhodne in izhodne enote. Kompaktni krmilniki sodijo v nižji cenovni in zmogljivostni razred in so namenjeni reševanju nalog manjšega obsega - zmorejo krmiljenje z majhnim številom vhodov in izhodov (odvisno od proizvajalca in tipa krmilnika). Vsebujejo digitalne vhode in izhode, nekatere izvedbe pa še dodatne analogne vhode in izhode.

PLK Omron Zen (slika 7) je primer krmilnika kompaktne izvedbe, poleg njega pa še v avtomatizaciji pogosto srečamo naslednje krmilnike iste izvedbe: Siemens S7-200, Logo, Moeller - Easy, Mitsubishi Alpha.



Slika 7 - Kompaktni PLK - Omron Zen [4]

Glavna prednost kompaktnih krmilnikov je možnost enostavne uporabe in programiranja, nizka cena ter možnost nadgradnje. Nadgradnja takšnih krmilnikov v primeru spremenjenih zahtev v procesu ni možna oz. je izvedljiva z (nesorazmernimi) visokimi stroški [2].

## 2.2.2 Modularni krmilniki

Modularni krmilniki so sestavljeni iz posameznih enot. Kot prikazuje slika 8, so enote vstavljene oz. nameščene po določenem vrstnem redu na montažni letvi. Napajalna enota je nameščena na prvem mestu (na levi strani), CPE se nahaja tik ob napajalni enoti, na drugem mestu. Na tretjem mestu so nameščene vhodno-izhodne enote, ki so lahko digitalne ali analogne. Sistem lahko razširimo in nadgradimo, nameščenih lahko imamo (desno od CPE) več posameznih enot, ki so med seboj povezane z internim vodilom [1]. Modularni krmilniki imajo možnost priključitve na industrijsko-podatkovno omrežje.



Slika 8 - Modularni PLK - Siemens S7-300 [9]

V avtomatizaciji pogosto srečamo naslednje modularne krmilnike: Siemens S7-300, S7-400, Mitsubishi AnSH.

Pri modularni izvedbi krmilnikov se lahko prilagajamo zahtevam procesov brez zamenjave osnovnih enot (napajalna enota, modul CPE). Odprtost njegove arhitekture omogoča priključitev najnovejših enot iste serije. Namenjeni so reševanju srednje-visokih zahtevnih problemov. Glavna pomanjkljivost modularnih krmilnikov v primerjavi s kompaktnimi krmilniki je visoka cena [2].

## 3. OSNOVNE LOGIČNE FUNKCIJE

Vsako digitalno krmilje deluje po nekem (napisanem) programu, ki ga imenujemo uporabniški program. Sestavljen je iz logičnih in uporabniških krmilnih funkcij. Ker imamo v praksi različne primere oz. zahteve nalog, so tudi uporabniški programi med seboj različni, vendar pa so posamezni elementi vedno enaki. Programe torej sestavljajo vedno enaki osnovni elementi (podsestavi), različno je le njihovo število in medsebojne povezave.

V praksi uporabljamo štiri osnovne skupine logičnih funkcij:

- kombinacijske funkcije;
- pomnilne ali sekvenčne funkcije;
- časovne funkcije;
- funkcije štetja [11].

## 3.1 Logične povezave med signali

Sisteme za vodenje obvladujemo z digitalnimi signali, katere lahko medsebojno sestavimo tako, da nam izvedejo določeno logiko delovanja. Osnovno logiko delovanja zgradimo iz osnovnih logičnih funkcij. Vso logiko, ki jo mora sistem vodenja izvesti, sprogramiramo v krmilnikih.

V avtomatiziranih procesih pogosto pride do primera z zelo zahtevnim logičnim delovanjem, pri katerem imamo več povezanih (enakih) signalov, ki se nam pojavljajo večkrat. V tem primeru je rešitev uporabe teoretičnega znanja in sicer tako, da logične funkcije pred realizacijo poenostavimo s pravili Boolove algebre. S temi pravili lahko enostavno zapišemo logične povezave signalov in jih tudi minimiziramo (s pomočjo Veitchevega diagrama) [8].

#### Zakonitosti Boolove algebre:

1. Komutativnost ali zamenljivost:

 $a \wedge b = b \wedge a$  $a \vee b = b \vee a$ 

2. Asociativnost ali združljivost:

 $(a \wedge b) \wedge c = a \wedge (b \wedge c)$  $(a \vee b) \vee c = a \vee (b \vee c)$ 

- 3. Operaciji konjunkcije in disjunkcije sta druga na drugo distributivni oz. razdružljivi:
  a ∧ (b ∨ c) = (a ∧ b) ∨ (a ∧ c)
  a ∨ (b ∧ c) = (a ∨ b) ∧ (a ∨ c)
- 4. Operaciji konjunkcije in disjunkcije sta medsebojno absorbtivni:

 $a \land (a \lor b) = a$  $a \lor (a \land b) = a$ 

- 5. Operaciji konjunkcije in disjunkcije sta antipodni:
  - $a \wedge \bar{a} = 0$  $a \vee \bar{a} = 1$

Logične povezave izvedemo s povezovanjem posameznih signalnih elementov. Poznamo tri osnovne logične funkcije: NE (negacija), IN (konjunkcija) in ALI (disjunkcija).

Značilnost kombinacijskih logičnih funkcij je, da je izhodni signal odvisen od (prisotnih) vhodnih signalov. To pomeni, da določena kombinacija vhodnih signalov daje na izhodu ustrezen logičen signal "0" ali "1", v skladu s pripadajočo logično pravilnostno tabelo [11].

## 3.1.1 Logična funkcija NE (negacija)

Logična funkcija NE ima en vhod in en izhod. Kot prikazuje tabela 3, ima na izhodu logično stanje "1", če je na vhodu logično stanje "0", in obratno. Logična funkcija NE povzroči nasprotno stanje. Elektrotehniško realiziramo negacijo z mirovnim kontaktom.

Tabela 3: Logična funkcija NE [Vir: lasten]

Algebraična	Simbol	Izvedba s stikali	Pravilnostna
logična enačba		(vezje)	tabela
$X = \bar{A}$	<b>▲ 1</b> ×		A         X           0         1           1         0

## 3.1.2 Logična funkcija IN (konjunkcija)

Logična funkcija IN ima dva ali več vhodov in en izhod. Na izhodu dobimo logično stanje "1", če se na vseh prisotnih vhodih hkrati pojavi logično stanje "1" (tabela 4). V vseh drugih primerih je na izhodu logični signal "0".

Algebraična	Simbol	Izvedba s stikali	Pravilnostna
logična enačba		(vezje)	tabela
$X = A \wedge B$	<u>в <b>&amp;</b> х</u>		A         X           0         0           1         0         0           0         1         0           1         1         1

Tabela 4: Logična funkcija IN [Vir: lasten]

## 3.1.3 Logična funkcija ALI (disjunkcija)

Logična funkcija ALI ima dva ali več vhodov in en izhod. Na izhodu dobimo logično stanje "1", če se vsaj na enem izmed vhodov pojavi logično stanje "1" (tabela 5).

Algebraična	Simbol	Izvedba s stikali	Pravilnostna
logična enačba		(vezje)	tabela
$X = A \lor B$	≜ <b>≥1</b> ×	< E-↓ B E-↓ × ×	A         B         X           0         0         0           1         0         1           0         1         1           1         1         1

Tabela 5: Logična funkcija ALI [Vir: lasten]

#### **3.2** Pomnilne funkcije

Pomnilne funkcije v digitalni elektroniki nastopajo z binarnimi logičnimi elementi (NE, IN, ALI) in imajo zmožnost shranjevanja (dokler je vezje pod napetostjo) enobitne informacije ("0" ali "1"). Pri pomnilnih funkcijah (za razliko od kombinacijskih funkcij) stanje izhoda ni vedno odvisno od trenutnega stanja vhoda. Osnovna značilnost pomnilnih funkcij je, da izhod ohrani (si zapomni) svoj zadnji prejeti signal (vhoda) poljubno dolgo še potem, ko tega signala ni več oz. se je spremenil. To stanje izhod ohrani, dokler ga drugi vhod ne postavi v drugačno stanje [11].

V elektroniki poznamo pomnilno funkcijo pod nazivom flip-flop, ki ga dovoljuje standard SIST EN 61131-3.

## 3.2.1 RS pomnilna celica

Pomnilna celica ima v osnovi dva vhoda, in sicer vhod za postavljanje (*angl. Set - S*) izhoda v stanje "1" in vhod za brisanje (*angl. Reset - R*) izhoda. Pomnilna celica je na voljo v dveh izvedbah: RS in SR pomnilna celica.



Slika 9 - Simbol in pravilnostna tabela RS pomnilne celice s prioriteto R [Vir: lasten]

Pomnilna celica ima v osnovi začetno stanje izhoda (Q) "0". Vhod za postavljanje (S) s svojim stanjem "1" postavi izhod v aktivno stanje "1", kar prikazuje slika 10. To stanje si izhod zapomni tudi po spremembi vhoda S na stanje "0". Niti ponovna postavitev vhoda S na stanje "1" nebo spremenila stanja izhoda. Do spremembe izhoda pride le ob postavitvi vhoda za brisanje (R) v aktivno stanje "1". Izhod se postavi v stanje "0" in si to stanje enako kot prej zapomni tudi po deaktiviranju signala na vhodu R.

Če opazujemo le izhod je ta v stanju "1", ko je vhod S enak "1" in v stanju "0", ko je vhod R aktiven. Če sta oba vhoda enaka stanju "0", pomeni da ni zahtev za spremembo izhoda in zato izhodno stanje ostane nespremenjeno.



Slika 10 - Časovni potek signalov pri RS pomnilni celici [Vir: lasten]

Pomembno je, da se pravilno odločimo katero izvedbo pomnilne celice (RS ali SR) bomo uporabili v nekem procesu, kajti tako določimo kateri izmed vhodov bo definiran kot prednostni vhod (slednji je definiran v tabelici znotraj simbola pomnilne celice).

Pri RS pomnilni celici (slika 9) prevladuje vhod za brisanje (nad vhodom za postavljanje). To pomeni, da se izhod celice briše (oz. ostane izbrisan), če je na obeh vhodih istočasno prisotno logično stanje "1". Signal se pojavi na dodatnem - invertiranem izhodu Q'. Pri SR pomnilni celici pa prevladuje postavitveni vhod: če sta oba vhoda v stanju "1", se izhod te celice postavi v stanje "1" (oz. ostane postavljen).

V industrijski praksi največkrat srečamo RS pomnilno celico, kajti v primeru napake ali okvare v sistemu, pride do (varnega) izklopa naprave (npr. motorja) - zaradi prevladujočega vhoda za brisanje. V splošni teoriji se pri načrtovanju procesov izogibamo stanju "1" na obeh vhodih - prepovedano stanje [2].

## 3.3 Časovne funkcije

V krmiljih se pogosto pojavljajo zahteve za časovno uskladitev krmilnih funkcij s procesom. Takšne zahteve opravljajo časovne funkcije.



Slika 11 - Časovne funkcije [Vir: lasten]

Časovne funkcije uporabljamo za določanje trajanja logičnih signalov. Čas trajanja logičnih signalov lahko skrajšamo, podaljšamo ali časovno premaknemo. Poznamo tri osnovne vrste časovnih funkcij: skrajševanje dolgih signalov (pulzov), podaljševanje kratkih signalov, časovna premaknitev signalov.

Vsaka časovna funkcija (slika 11) ima tri vhode in dva izhoda:

- Vhod S je binarni postavitveni vhod, ki sproži delovanje časovnika;
- Časovniku priredimo parameter, tj. časovno konstanto na vhodu TW, s čimer definiramo dolžino signala;
- Vhod R predstavlja ukaz za zaustavljanje časovnika;
- Na izhodu Dual / Dez lahko odčitamo trenutno stanje časovnika (večbitni podatek);
- Izhod Q je binarni izhod, ki se postavi v določeno stanje po poteku časa [3].



Slika 12 - Simbol in časovni potek časovnega člena za podaljšanje/skrajšanje signalov (prirejeno po [12])

Kot prikazuje slika 12, lahko s tem členom skrajšujemo ali podaljšujemo različno dolge vhodne signale (A) na želen čas oz. dolžino (trajanje t1 sekund). Kratkotrajne vhodne signale podaljšujemo v kombinaciji z logično ALI funkcijo.

Časovni člen, ki omogoča časovne premike signalov, kaže slika 13. Kot je razvidno iz časovnega diagrama, je izhod (Q) časovno zakasnjen za čas t1, glede na začetek signala "1" na vhodu (A) - zakasnitev vklopa. S prenehanjem vhodnega signala se izhod zakasni za čas t2 - zakasnitev izklopa. Čas trajanja t1 in t2 lahko poljubno vnaprej nastavimo (v določenih mejah) [12].



Slika 13 - Simbol in časovni potek časovnega člena za premikanje signalov [12]

Časovno zakasnitev uporabljamo, da preprečimo trke cilindrov ali drugih delov stroja. V krmilni tehniki se pogosto uporabljata še dva načina posebnih časovnih funkcij. To sta (zgoraj omenjeni) zakasnitvi vklopa in izklopa (pogl. 3.3.1 in 3.3.2).

## 3.3.1 Zakasnitev vklopa

Časovno funkcijo zakasnitev vklopa prikazuje slika 15. Časovnik aktiviramo s stanjem "1" na vhodu S. Na izhodu dobimo signal, ko poteče nastavljeni čas Ti (nastavljiv na TW), ob postavki da imamo na vhodu S še vedno prisotno stanje "1". Če je stanje vhodnega signala krajše od nastavljenega (zakasnilnega) časa, se časovnik ustavi in dobimo na izhodu stanje "0". Z aktiviranjem vhoda R pred potekom nastavljenega časa brišemo stanje časovne funkcije Ti (postavimo jo v stanje "0"), čeprav je na vhodu S še vedno logično stanje "1". Tudi po preteku nastavljenega časa lahko izhodni signal časovnika prekinemo (brišemo) z aktiviranjem vhoda R - stanje izhoda se postavi nazaj na "0" [1].

Na sliki 14 je prikazan simbol časovne funkcije zakasnitev vklopa.



Slika 14 - Časovnik kot vklopna zakasnitev (prirejeno po [3])



Slika 15 - Časovni potek zakasnitve vklopa [Vir: lasten]

## 3.3.2 Zakasnitev izklopa

Pri funkciji zakasnitev izklopa (slika 17) se izhod postavi v stanje 1 skupaj z vhodom S. Ob prehodu vhoda iz stanja "1" v stanje "0" se časovnik aktivira. Ko poteče nastavljeni čas Ti (nastavljiv na TW) se izhod postavi v stanje "0". Časovnikov čas se ustavi s ponovnim aktiviranjem vhoda S. Če je čas trajanja vhoda v neaktivnem stanju krajši od nastavljenega (zakasnilnega) časa Ti, se časovnik ne more aktivirati (sredi poteka se ustavi). Izhod je v stanju "1" tako dolgo, dokler časovnik neovirano ne doseže aktivnega stanja. Izhodni signal časovnika lahko prekinemo (brišemo) z aktiviranjem vhoda R ter postavimo izhodni signal (nazaj) v stanje "0" [2].

Na sliki 16 je prikazan simbol časovne funkcije zakasnitev izklopa.



Slika 16 - Časovnik kot izklopna zakasnitev (prirejeno po [3])



Slika 17 - Časovni potek zakasnitve izklopa [Vir: lasten]

## 3.4 Samodržna vezava

V elektropnevmatiki moramo pogosto impulzne signale pretvoriti v trajne signale. Za-to je potrebna t.i. samodržna vezava.

Samodržna vezava se uporablja za krmiljenje monostabilnih elektromagnetnih ventilov, pri katerih je pomembno, da prekrmiljeno stanje traja samo v času aktiviranja ventila. Ko ventil ni več aktiviran, vzmet vrne drsnik ventila nazaj v začetni osnovni položaj. Samodržna vezava, kratkotrajni impulz pretvori v trajni signal in ga preko releja posreduje monostabilnem ventilu.

Obstajata dve izvedbi samodržne vezave:

- Vezava s prevladujočim vklopom;
- Vezava s prevladujočim izklopom.

Funkcijo remanentnega releja - da bo sposoben obdržati (si zapomniti) trenutno stanje tudi po odklopu krmilnega signala, lahko realiziramo z običajnim relejem s pomočjo samodržne vezave.

#### Delovanje vezave s prevladujočim vklopom

S pritiskom na tipko START, se sklene tokovna veja 1 in navitje releja K1 dobi napetost in pritegne svoje kontakte (kateri so lahko mirovni ali delovni). Pri tem se sklene delovni kontakt K1 v tokovni veji 2 in 3. V veji 3 se (preko delovnega kontakta K1) aktivira magnetni ventil 1Y1. Če sedaj tipko START izpustimo, ostane navitje K1 pod napetostjo preko pritegnjenega kontakta K1 in mirovnega kontakta tipke STOP. Tako rele drži sam sebe, čeprav tipka START ni več aktivirana. Vezavo s prevladujočim vklopom sprostimo z mirovnim kontaktom tipke STOP. Vezava s prevladujočim vklopom se imenuje zato, ker če v trenutku, ko je rele pritegnjen, istočasno pritisnemo na tipki START in STOP, prevlada tipka START - rele K1 ostane vzbujen (pod napetostjo).

#### Delovanje vezave s prevladujočim izklopom

Če pritisnemo tipko START, dobi navitje releja K1 napetost (tokovna veja 1) prek delovnega kontakta tipke START in mirovnega kontakta tipke STOP. Pri tem se sklene delovni kontakt releja K1 v tokovni veji 2 in 3. V veji 3 delovni kontakt K1 aktivira magnetni ventil 1Y1. Če sedaj tipko START izpustimo, ostane navitje K1 pod napetostjo preko pritegnjenega kontakta K1. Tako rele drži sam sebe, čeprav tipka START ni več aktivirana. Vezavo s prevladujočim izklopom sprostimo s tipko STOP. Vezava s prevladujočim izklopom se imenuje zato, ker če v trenutku, ko je rele pritegnjen, hkrati [1].

Delovanje samodržne vezave s prevladujočim vklopom in s prevladujočim izklopom prikazuje slika 18.



Slika 18 - Samodržna vezava (prirejeno po [1])

## 4. PROGRAMIRANJE

## 4.1 Osnovni načini programiranja

Standard SIST EN 61131-3 definira programske jezike za programiranje PLK-jev. V splošnem delimo programske jezike na grafične (slika 19) in tekstualne jezike:

- Grafični jeziki: lestvični diagram (LAD), funkcijski blokovni diagram (FBD), sekvenčni funkcijski diagram (SFD);
- Tekstualni jeziki: strukturirani tekst (STL), seznam navodil (IL).



Slika 19 - Programski grafični jezik - Lestvični diagram [4]

Najbolj pogosti način programiranja je lestvični diagram (*angl. Ladder diagram*), ki ima zgodovinski pomen programirljivih krmilnikov (Modicon 084, pogl. 1.1). Pri programiranju FBD (*angl. Function block diagram*) sestavljamo programske gradnike v celoto. V programskem jeziku STL (*angl. Statement list*) programiramo opisno - podajanje kode v jeziku, ki spominja na jezik Pascal ali C. Programski jeziki različnih proizvajalcev niso združljivi. Pri programskih jezikih ne gre pozabiti, da je dejanski program v krmilniku napisan v strojni kodi. Grafični jeziki so namenjeni za realizacijo enostavnejših funkcij, temeljijo na standardiziranem načinu zapisa relejnih in logičnih povezav, zato so grafični programski jeziki enako razumljivi vsem uporabnikom. Medtem, ko pa so tekstualni jeziki namenjeni predvsem uporabi zapletenih ukazov in struktur programa [2, 8].

#### 4.1.1 Lestvični diagram (LAD)

Lestvični diagram imenovan tudi kontaktni načrt je grafični programski jezik za programiranje logičnih krmilnikov. Pri lestvičnem diagramu so ukazi predstavljeni z grafičnimi simboli, ki izhajajo iz stikalno-relejne tehnike. Osnovna elementa sta delovno (NO) in mirovno (NC) stikalo, poleg le-teh pri programiranju uporabljamo še druge funkcije. Programiranje po tej metodi je zelo pregledno in se ga da dokaj hitro naučiti (hitra in razumljiva uporaba programa) [13].

Način programiranja z uporabo lestvičnega diagrama se pogosto uporablja, predvsem pri reševanju manj zahtevnih krmilnih problemov. Glavna pomanjkljivost lestvičnega diagrama je, da ta metoda programiranja ni primerna za zahtevnejše obdelave digitalnih vrednosti (npr. računske operacije z 32-bitnimi števili) [2].

Lestvični diagram si lahko predstavljamo kot niz povezav med relejnimi kontakti in tuljavami. Ko obstaja pot oz. povezava med levo stranjo lestve in izhodom preko zahtevanih kontaktov (kot prikazuje slika 20), se izhod tuljave aktivira - postavi se na logično stanje "1".



Slika 20 - Primer lestvičnega diagrama [Vir: lasten]

Lestvični diagram se bere od leve proti desni (in z vrha proti dnu). Lestev ima veliko vrstic, oštevilčenih po vrstnem redu. Vsaka vrstica ima običajno eno tuljavo (izhod) na desni strani lestve, medtem ko se t.i. kontakti nahajajo (v prvih stolpcih) na levi strani. Kontakti pomenijo vhodni signal za PLK, ki ga dajejo fizične naprave (tipkala, preklopniki - mejna stikala) ali pa lahko predstavljajo stanja notranjih ali izhodnih bitov (tu pogosto nastopajo zastavice, pogl. 2.2).

#### Zapiralni, odpiralni kontakt

Osnovni element za oddajanje signalov je kontakt, ki je lahko zapiralni (*angl. Normally Open - NO*) ali odpiralni (*angl. Normally Closed - NC*). S tema dvema vrstama kontaktov preverjamo po stanju (vrednosti) binarnih operandov ("0" ali "1"). Vsak binarni operand ima svoj naslov (*angl. address*), ki je napisan nad kontaktom, kar prikazuje slika 21.



Slika 21 - Binarni operand v kombinaciji z zapiralnim in odpiralnim kontaktom [Vir: lasten]

S slike 22 je razvidno, če je na binarnem operandu I0 stanje "1", je **zapiralni kontakt** (imenovan tudi delovni kontakt) aktiviran in tokokrog sklenjen - rele vklopi. V primeru, da je vhod neaktiven (tipka I0 ni pritisnjena), na operandu ni signala, zato rele spusti oz. izklopi.



Slika 22 - Delovanje zapiralnega (delovnega) kontakta [Vir: lasten]

**Odpiralni kontakt** (slika 23) se aktivira pri logičnem stanju "0" na vhodu, torej, ko tipka I1 ni pritisnjena - navitje v releju vklopi. Tokokrog se prekine z aktiviranjem vhoda (tipke I1), zato rele spusti oz. izklopi.


Slika 23 - Delovanje odpiralnega (mirovnega) kontakta [Vir: lasten]

Kontakti zapirajo (sklenejo) ali odpirajo (prekinejo) tokokrog in s tem posredujejo signale, ki so posledica delovanja na kontakte.

V programu posameznem binarnem operandu ("vhodu") določimo naslov in vrsto kontakta, kajti CPE ne more ugotoviti, ali imamo na določen vhod priključen zapiralni ali odpiralni kontakt (CPE prepozna le logično stanje 0 ali 1) [1].

# 4.1.2 Funkcijski blokovni diagram (FBD)

Funkcijski blokovni diagram je programski jezik, pri katerem ukaze vnašamo z grafično metodo. Uporablja se za realizacijo kompleksnejših funkcij, ki so predstavljene direktno z logičnimi bloki Boolove algebre (slika 24). Programiranje je enostavno in razumljivo, sam program je pregleden, saj zelo nazorno kaže logične povezave med spremenljivkami, kar nam olajša odkrivanje napak v procesu (neposredno lahko opazujemo delovanje programa oz. spremenljivk). Ta programski jezik podobno kot lestvični diagram ni primeren za programiranje zahtevnejših računskih operacij [13].



Slika 24 - Primer funkcijskega blokovnega diagrama (z logičnimi bloki) [Vir: lasten]

### 4.1.3 Seznam ukazov (STL)

Seznam ukazov ali strukturirani tekst je besedilni programski jezik, pri katerem ukaze vnašamo v tekstovni obliki. Namenjen je zahtevnejšim operacijam. Sestavljen je iz zaporedja ukazov, ki skrbijo za hitro izvajanje programa (čas obdelave podatkov in lokacija pomnilnika sta optimizirana). Vsak ukaz je sestavljen iz operacijske kode in operanda. Operand sestavlja oznaka (vhod, zastavica...) in parameter (naslov operanda).

Celoten program se napiše v urejevalnik besedila (med najbolj poznanimi je beležnica - *angl. notepad*). V samem urejevalniku lahko uporabimo simbolično opisovanje, možne so tudi funkcije za iskanje ter sprotno preverjanje pravilnosti vnosa. Pred vsakim ukazom lahko vpišemo oznako, ki jo lahko uporabimo za sklicevanja, na primer ob skokih. Program na koncu prevedemo v ustrezno obliko.

Slika 25 prikazuje program, napisan iz grafičnega jezika FBD v tekstualni jezik STL, pri katerem:

- ukaz A pomeni logični IN;
- ukaz AN pomeni logični NE-IN;
- ukaz O pomeni logični ALI;
- simbol za enakost določa mesto hranjenja določenih rezultatov.



Slika 25 - Primer tekstualnega jezika STL [Vir: lasten]

Največja slabost tega jezika je precejšnja nepreglednost pri programu, kar oteži delo pri odkrivanju napak v procesu. Težave lahko nastopijo, če ne dopolnjujemo posameznih ukazov s komentarji. Ti so nam še posebej v pomoč za lažje razumevanje programa [14].

# 5. PLK OMRON ZEN

### 5.1 Predstavitev krmilnika Omron Zen

Krmilnik (PLK) Omron Zen predstavlja spodnji zmogljivostni razred. Namenjen je za avtomatizacijo enostavnejših industrijskih sistemov (procesov), kot so krmiljenje strojev, hidravličnih dvigal, tekočih stopnic, procesov v industriji. Uporaben je tudi za domače namene, npr. krmiljenje v elektroinštalacijah, kontrola nivoja rezervoarja vode, spremljanje temperature v vrtnem rastlinjaku. PLK Omron Zen je v kompaktni izvedbi in ga odlikuje predvsem enostavna uporaba, preprosto programiranje s pomočjo programske opreme ZEN in nizka cena. Družina Omron ponuja krmilnik ZEN v izvedbi s fiksnimi 10 ali 20 V/I z možnostjo razširitve do 44 V/I. Na voljo so z relejskimi ali tranzistorskimi izhodi.

Najpomembnejše komponente, ki jih srečamo pri krmilniku Omron Zen (slika 26) se nahajajo v kompaktnem ohišju krmilnika. Komponente, ki sestavljajo krmilnik so:

- Centralna procesna enota;
- Vhodne in izhodne enote (digitalne, analogne);
- Napajalna enota.



Slika 26 - Kompaktni krmilnik Omron Zen z 20 V/I [Vir: lasten]

Krmilnik Omron Zen nima možnosti priključitve na industrijsko-podatkovno omrežje in ga glede na svoje lastnosti uvrščamo pod programirljiv rele. Tak krmilnik je zelo zadržan oz. omejen s funkcijami, ki jih nudi. Jedro celotnega sistema je centralna procesna enota, ki skrbi za izvajanje programa ter za komunikacijo z drugimi enotami. Glede na zahtevnost naloge imamo na voljo več različnih tipov krmilnika Omron Zen: standardni, LED, ekonomični in komunikacijski (tehnične podatke najdemo v priloženi prilogi H).

Z različno kombinacijo lahko sestavimo sistem, ki bo zadostoval potrebam procesa oz. projekta. Prilagodimo lahko CPE in širitvene V/I enote [4]:

• Centralna procesna enota



- Število vhodov in izhodov
   10: 6 vhodov in 4 izhodov
  - 20: 12 vhodov in 8 izhodov
- 2. Tip razvrščanja
  - 1: Standardni LCD tip z zaslonom
  - 2: LED tip brez zaslona<sup>\*</sup>

3: Ekonomični tip z zaslonom
(priključitev širitvenih V/I enot ni možen)
4: Komunikacijski tip z zaslonom
(komunicira preko RS-485)

• Širitvena V/I enota



- Število vhodov in izhodov
   8: 4 vhodov in 4 izhodov
- Izvedba razvrščanja enot
   E1: Povežete lahko k V2 CPE

- Tip vhoda
   A: AC izmenični vhod
   D: DC enosmerni vhod
- 4. Tip izhoda
  - R: Relejski izhod
  - T: Tranzistorski izhod
- 5. Napajalna električna napetost
  - A: AC izmenični napajalnik
  - D: DC enosmerni napajalnik

- 3. Tip vhoda
  - A: AC izmenični vhod D: DC - enosmerni vhod
- 4. Tip izhoda
  - R: Relejski izhod
  - T: Tranzistorski izhod

Kot prikazuje slika 27, se na sprednji strani krmilnika nahajajo operacijske tipke, majhen LCD zaslon, priključne sponke za napajalno napetost, priključki za vhodni/izhodni del ter priključek za povezavo krmilnika z osebnim računalnikom s pomočjo programske opreme Zen. V jašek tega priključka lahko vstavimo tudi pomnilniško kaseto (EEPROM), katero lahko uporabimo za shranjevanje in kopiranje uporabniškega programa ter nastavitev.





Na levi strani krmilnika se pod pokrovčkom nahaja poseben jašek, v katerega namestimo podporno baterijo (slika 28). Naloga baterije je, da ob izpadu električne napetosti ohrani vsebino (koledar, ura, držalni časovnik). Brez podporne baterije bi ob izpadu napajanja svojo vrednost ohranili remanentni elementi na CPE (pomožni pomnilniki, števci, časovniki). Iztrošeno podporno baterijo zamenjamo, ko je krmilnik izključen (življenjska doba baterije je 10 let). V nasprotnem primeru se naložena vsebina izbriše.



Slika 28 - Dodatna oprema krmilnika [4]

Na desni strani krmilnika imamo možnost razširitve vhodov in izhodov krmilnika. Na osnovno enoto (krmilnik) lahko priključimo največ 3 širitvene V/I enote in tako razširimo V/I kapaciteto do 24 vhodov in 20 izhodov. Širitvene V/I enote ne morejo biti priključene k ekonomičnemu tipu krmilnika.

Pri krmilniku imamo na izbiro 6 različnih jezikov (španščina, italijanščina, francoščina, nemščina, japonščina, angleščina), privzeti nastavljeni jezik je angleščina.



Slika 29 - Zaslon krmilnika [4]

Na sliki 29 je prikazan zaslon krmilnika Omron Zen, pri katerem so prikazane ikone pojasnjene v tabeli 6.

Ikona	Pomen
RUN	Označuje izvajanje programa.
ERR	Označuje napako.
<b>A</b>	Prikaže, ko obstaja višja raven menija ali linija
	lestvičnega programa (v trenutnem položaju).
▼	Prikaže, ko obstaja nižja raven menija ali linija
	lestvičnega programa (v trenutnem položaju).
0 <del></del>	Prikaže, ko je bilo nastavljeno geslo.

Tabela 6: Pomen ikon na zaslonu krmilnika (prevedeno po [4])

Za povezavo oz. serijsko komunikacijo krmilnika z računalnikom uporabimo Omron Zen povezovalni kabel - RS-232C (9-pinski D-sub priključek). V primeru, če računalnik nima serijskih vrat RS232 (COM1), je rešitev v tem, da uporabimo pretvorbeni USB kabel.

Podatke o porabi električne energije, vklopnemu toku ter pogojih delovanja (temperaturno območje, vlažnost) - posameznih tipov krmilnika Zen in širitvenih V/I enot najdemo v prilogi I. Dimenzije krmilnika, napajalne in širitvene V/I enote so podane v prilogi K.

## 5.2 Programiranje s pomočjo programske opreme ZEN

Za programiranje kompaktnih krmilnikov Omron Zen se uporablja programa oprema ZEN. Programsko orodje omogoča programiranje skladno s standardom EN 61131-3 in vključuje programski jezik - lestvični diagram (LD).

Krmilniki Zen uporabljajo zelo enostavno metodo programiranja tj. programiranje organizacijskih blokov (OB1); ves uporabniški program je sklenjen v enem bloku. V OB1 običajno kličemo uporabniške funkcije in signale. Z drugimi besedami povedano, uporabniški program sestavljajo spremenljivke - biti (vhodi, izhodi, časovniki, števci, tipke itd.). Uporabniški program in vsi podatki so shranjeni v EEPROM pomnilniku. Procesor izvaja program program linearno po stavkih, kot so vpisani. Dodatno se lahko uporablja tehnika podprogramov.

Izvajanje programa v/na krmilniku poteka v programskih ciklih. V vsakem programskem ciklu, ki običajno traja nekaj milisekund, se:

- najprej preberejo vhodi;
- nato se izvede uporabniški program;
- nazadnje pa se na izhode zapišejo izračunane vrednosti.

Ko se programski cikel zaključi, se takoj sproži nov programski cikel. Izvajanje programskega cikla je lahko prekinjeno zaradi sistemskih operacij v krmilniku ali zaradi zahteve uporabnika (npr. če je potreben temeljit pregled ali servis).

Programiranje krmilnika Omron Zen je možno na dva načina:

- Programiranje s programsko opremo ZEN z uporabo osebnega računalnika, katerega serijska vrata so preko posebnega vmesnika (RS-232) povezana na priključek krmilnika;
- Interno programiranje (ročni vnos programa v krmilnik).

## 5.2.1 Programska oprema ZEN

Programska oprema ZEN je enostavna za uporabo in učenje. Temelji na uporabi menijske vrstice in oken. Program omogoča grafično programiranje z lestvičnim diagramom in uporabo različnih skupin ukazov.

S programsko opremo ZEN lahko simuliramo (*angl. offline* oz. simulacija brez povezave) izdelane programe in ugotovimo nepravilnosti v delovanju še preden ga prenesemo v PLK Zen. Zelo dobra lastnost programske opreme ZEN je možnost, da lahko program med delovanjem v krmilniku (*angl. online*) nadziramo na monitorju računalnika (več o poteku simulacije v programski opremi ZEN v nadaljevanju). Pri tem načinu delovanja lahko neposredno spremljamo trenutna stanja različnih spremenljivk. Tako lahko opazujemo vhode, izhode, zastavice, hitre števce in še nekatere druge spremenljivke. Med drugim lahko program iz krmilnika shranimo v računalnik ali pa ga natisnemo.

Vse opisane možnosti omogočajo uporabniku hitro in preprosto implementacijo njegovega sistema.

Za namestitev programske opreme ZEN je potrebno imeti osebni računalnik z (vsaj) naslednjimi zahtevami, ki so podane v tabeli 7. Namestitev programske opreme poteka po klasičnem postopku nameščanja programov.

Postavka	Pogoji
Operacijski sistem	Microsoft Windows 95, 98, 2000, ME,
	XP, NT 4.0, Vista ali 7 (32/64-bit)
СРЕ	Pentium 200 MHz ali več
Spomin (RAM)	64 Mb ali več
Trdi disk	40 Mb ali več prostega prostora
CD-ROM	Zahtevano.
Komunikacija	1 serijska vrata (COM1)
Tipkovnica in miška	Zahtevano.
Monitor	800 x 600 (SVGA), 256 barv ali več

Tabela 7: Zahteve programske opreme (prevedeno po [5])

#### Priprave na programiranje s programsko opremo ZEN

Nameščeno programsko opremo ZEN najdemo v meniju Start/Programi/OMRON/Zen Support Software/Zen Support Software. Ob zagonu programske opreme ZEN se nam prikaže začetni zaslon z imenom programske opreme (slika 30).



Slika 30 - Programska oprema ZEN - Verzija 4.14 [Vir: lasten]

Ko se prikazani začetni zaslon zapre, se nam pojavi naslednja slika. Izberemo lahko med tremi opcijami (slika 31):

- Create a new program Ustvari nov program;
- Load programs from files Naloži programe iz datotek;
- Read program from ZEN Beri program iz krmilnika ZEN.



Slika 31 - Ustvarjanje novega programa [Vir: lasten]

Za ustvarjanje novega programa izberemo prvo opcijo in kliknemo na tipko OK.

Prikaže se zaslon Property Settings (slika 32). Izberemo tip krmilnika Omron Zen, ki ga bomo uporabili in nastavimo konfiguracijske nastavitve (če imamo priključene dodatne širitvene V/I enote, jih označimo). Podatek o tipu krmilnika najdemo na ohišju krmilnika. Projektu lahko damo ime in komentar. Za zagon programske opreme kliknemo na tipko OK.

Property Settings			
CPU Unit type:	-V2	•	ОК
ZEN types:	20C1D*-D-V2	•	Cancel
Expansion Unit 1:	No	•	Help
Expansion Unit 2:	No	~	
Expansion Unit 3:	No	Υ.	
Project Name:	Nov projekt		
Comment:	-		

Slika 32 - Konfiguracijske nastavitve [Vir: lasten]

Programska oprema ZEN se zažene. Prikaže se zaslon v načinu/pogledu lestvičnega diagrama. Pred začetkom programiranja je potrebno vedeti osnove za delo s programsko opremo. Programe pišemo z uporabo različnih ukazov. Vsi ukazi, ki jih uporabljamo med programiranjem se nahajajo v menijski vrstici. Vsak ukaz ima svoj pomen. Meniji so podrobno predstavljeni na sliki 33.

Menijska vrstica z šestimi skupinami –	File Edit View	Insert ZEN Help				
ukazov Orodjarna z – najpogosteje uporabljenimi ukazi menijske vrstice			× <u>2</u> 2 4 1 8 7 7 1		-	Komentar vrstice
Lestvični pogled _ programa _	2 3 4 5	· ·		2 - 2 		
Informacije o posameznem bitu – Statusna vrstica, v kateri se izpisuje položaj kurzorja (miške)	Relay Type: Comment: Set Value: Present Value: Double-click (pressi	ng Enter key) enables yo	u to enter contacts/coils.	20C1D*-D-V2	Offine /	

Slika 33 - Meniji programske opreme [Vir: lasten]

Na zgornji strani programske opreme imamo menijsko vrstico s šestimi skupinami ukazov:

- Skupina FILE omogoča ustvarjanje in poimenovanje novega projekta, shranjevanje programa ali tiskanje lestvičnega diagrama. Spremenimo lahko tudi začetne konfiguracijske nastavitve (tip krmilnika, širitvene V/I enote, naslov projekta);
- Ukazi skupine EDIT omogočajo urejanje. Če poznamo naslov posamezne spremenljivke ali vrstico na kateri se nahaja, lahko uporabimo ukaz Najdi (Find), ki nam na hiter način najde iskano spremenljivko ali vrstico;
- V skupini VIEW lahko prilagodimo pogled programske opreme, dodamo ali odstranimo posamezne skupine (oz. bližnjice orodne vrstice) in izberemo med dvema prikazanima pogledoma - lestvični diagram ali električni (relejski) diagram;
- V skupini INSERT lahko dodajamo vrstice, bite, vodoravne ali navpične povezave med biti. Ti ukazi so namenjeni načrtovanju programov;
- Ukazi skupine ZEN se nanašajo predvsem na napisan uporabniški program kot tudi na krmilnik. S pomočjo programske opreme lahko pred prenosom programa v krmilnik preverimo, če smo bili uspešni pri pisanju programa ali pa lahko s pomočjo simulacije virtualno preizkusimo delovanje programa (tako lažje odkrijemo napake).

Po končanem programiranju je pred prenosom programa v krmilnik, smiselno preveriti naše povezave, bite. Lahko, da se nam je kje zalomilo, da smo naredili napako. Zaradi napak lahko pride do posledic, ki bi lahko negativno vplivale na proces avtomatizacije. Zato imamo na izbiro več ukazov, ki nam lahko olajšajo delo pri odkrivanju napak v programu inp.

Najpogosteje uporabljeni ukazi v skupini ZEN so: Preverjanje programa, Prenos programa iz programske opreme v/na krmilnik (in obratno) ter Spremljanje delovanja programa in spremenljivk - programska oprema omogoča simulacijo z ali brez povezave.

Krmilnik lahko zaganjamo in ustavljamo s programsko opremo ZEN in tako vplivamo na delovanje krmilnika. Preklapljamo lahko med dvema stanjema oz. načinoma obratovanja: RUN način omogoča preklop v stanje izvajanja programa (delovanje), STOP način pa omogoča preklop v stanje prekinitve izvajanja programa (čakanje). Pred pričetkom programiranja še lahko spremenimo oz. nastavimo naslednje parametre, kot so jezik, čas, datum in geslo. Na izbiro imamo 6 različnih jezikov (privzeti nastavljeni jezik je angleščina). Programska oprema ima možnost sinhronizacije časa in datuma neposredno iz operacijskega sistema Windows. Krmilniku lahko zaradi varnostnih razlogov nastavimo geslo. Geslo lahko vsebuje le 4 znake. Pri nastavljanju gesla je pomembno, da si jo zapomnimo. Kajti v primeru, če pozabimo geslo, nam bo onemogočeno urejanje in brisanje vsebine programa. Uporabniku je na voljo tudi možnost brisanja celotnega pomnilnika CPE krmilnika. S tem bi izbrisali obstoječi uporabniški program, V/I bite in nastavljeno geslo krmilnika. V primeru, ko ne vemo, kaj se nahaja na krmilniku in v kakšnih stanjih so npr. remanentni pomožni pomnilniki (zastavice), je smiselno krmilnik izbrisati. Širitvenim V/I enotam lahko po potrebi nastavimo tudi vhodne filtre, ki bi poskrbeli za izločanje hitrih preklopov zaradi zunanjih motenj;

 V skupini HELP imamo možnost uporabe pomoči. Programska oprema vključuje obširno interaktivno pomoč. Uporabimo lahko tipko F1 na izbranem objektu ali z miško kliknemo v meni na opcijo Pomoč (Help). Sprotna pomoč nam hitro razreši marsikatero dilemo.

Pod menijsko vrstico se nahaja orodna vrstica, v kateri se nahajajo najpogosteje uporabljeni ukazi menijske vrstice. Simbole v orodni vrstici prikazuje slika 34. Na izbiro imamo ukaze, ki so razporejeni v standardne, lestvične, ZEN in simulacijske skupine. V času programiranja imamo največ opravka s skupino lestvični diagram, v kateri najdemo pod ukazoma "Vstavi kontakt" in "Vstavi tuljavo" različne spremenljivke (vhodi, izhodi, zastavice, časovniki, števci...), ki jih uporabimo pri programiranju oz. načrtovanju lestvičnega diagrama. V skupini simulacija najdemo vse potrebne ukaze za postopek simulacije programa. Ko imamo program pripravljen za prenos v krmilnik, ga lahko v skupini ZEN prenesemo v/na krmilnik.



Slika 34 - Menijske vrstice (prevedeno po [5])

V vrstici posameznega bita najdemo informacije, kot so tip spremenljivke (bita), komentar, nastavljene vrednosti časovnika ali števca, vhodne informacije ter trenutne vrednosti časovnikov ali števcev.

Na koncu (spodnja vrstica) pa še imamo statusno vrstico, v kateri se izpisuje položaj miške. V statusni vrstici so podani še naslednji podatki: priključen tip krmilnika Zen, status povezave (Online/Offline - povezano/nepovezano), obratovalni način (RUN/STOP - delovanje/zaustavitev) in komentarji (pojasnila, napake).

### Programiranje v programski opremi ZEN

Sedaj, ko poznamo pomene posameznih ukazov v programski opremi ZEN, lahko v osnovnem polju ustvarimo program v obliki lestvičnega diagrama. Programiranje poteka v grafičnem načinu, kjer vnašamo in povezujemo spremenljivke. Z dvojnim klikom miške ali s pritiskom tipke ENTER na označeni poziciji (svetlomodro polje), lahko programu dodamo prvi bit (vhod) in začnemo s programiranjem.

Prikazalo se bo okno, v kateri izberemo spremenljivko in ji nastavimo parametre. Na sliki 35 so prikazani posamezni sklopi nastavitev.

Edit Contact							×	
# @   Q	× ×	C Y	F	A H	P B	G		
Relay Name: Cl Contact:	PU Unit Inpu	ut Relay Normally O Normally Cl	pen –  osed – ,	⊦ ≁				<ul> <li>Ime bita: I (vnod)</li> <li>Kontakt: Normalno odprti (NO) Normalno zaprti (NC)</li> </ul>
Relay No.: Comment:		TH O	ъ			<b>-</b>		— Številka bita: 0 — Barva komentarja — Komentar

Slika 35 - Parametri vhoda [Vir: lasten]

Vhodu lahko dodamo komentar. Kontakt in številko bita pa lahko pustimo na začetni privzeti vrednosti. Na naslednji označeni poziciji (svetlomodro polje) lahko poljubno dodamo še naslednji vhod oz. bit (slika 36). Postopek je enak kot za prvi vstavljeni bit (vhod). Na vrstico lahko damo največ tri spremenljivke.

File Edit View Insert ZEN Help	
D <b>26 50 100 100 100 100 100 100 100 100 100 </b>	
	<b>_</b>
Vhod 0	_
1	
2	

Slika 36 - Dodajanje vhodnih spremenljivk [Vir: lasten]

Na koncu vrstice dodamo izhod, katerem določimo funkcijo, številko bita in komentar. Posamezne funkcije izhoda prikazuje slika 37. Vsaki spremenljivki lahko nastavimo različne parametre.

Edit	Coll					
Q	Y M	Н  Т  #  С	F	D		Ime bita: O (izbod)
F	Relay Name: CPU	Unit Output Relay				
	Function:	<ul> <li>Normal output operation</li> <li>Set/Reset operation</li> </ul>	C			— Funkcija: Normalni izhod - [ Postavi - S / Briši - R
		C Set operation	s			Izmenjalka - A
		C Reset operation	R			
		C Alternate operation	A			— Številka bita: 0
	Relay No .:	0-7				
	Comment:					— Barva komentarja
						Komentar

Slika 37 - Funkcije izhoda [Vir: lasten]

Preprosti program za lažje razumevanje programiranja v programski opremi Zen je končan. Program je v obliki lestvičnega diagrama in vsebuje tri vhode in en izhod, kar prikazuje slika 38. Vhodi so na levi strani, izhodi pa na desni strani (na koncu vrstice) lestvičnega diagrama. Vhodi ne morejo biti postavljeni za izhodi.



Slika 38 - Eno vrstico sestavljajo trije vhodi in en izhod [Vir: lasten]

Vsaki vrsti lahko dodamo komentar in tako program naredimo še preglednejši. Komentar lahko dodamo desno od izhoda z dvojnim klikom miške. V programskem orodju Zen lahko uporabimo do 96 vrstic programa, s tremi vhodi in enim izhodom na vrstico.

Uporabniški program lahko razširimo z vstavljanjem novih spremenljivk in funkcij. Programerji zelo pogosto pri načrtovanju programov, najprej naredijo osnovni program, katerega pa kasneje razširijo ter prilagodijo potrebam nekega avtomatiziranega procesa oz. naloge. Programska oprema ZEN omogoča, da lahko program prilagodimo, npr. z dodajanjem spremenljivk in funkcij v vezje naredimo program bolj funkcionalen, zahteven. Tu pogosto nastopi problem, ki se nanaša na prostorsko stisko med vrsticami programa. Kot smo že omenili, poteka programiranje v lestvičnem diagramu z vstavljanjem spremenljivk, ta postopek se ponavlja vse do zadnjega bita - do zadnje vrstice. Tako dodobra zapomnimo posamezne vrstice. Kaj pa narediti v primeru, ko želimo dodati oz. priključiti spremenljivko k vrstici, na kateri ni več prostora za vhode? Rešitev je v tem, da dodamo novo vrstico.

Pri dodajanju nove vrstice moramo najprej označiti pozicijo (svetlomodro polje), na kateri želimo dodati prazno vrstico. Vrstico dodamo z izbiro Insert/Rung (kombinacija tipk Ctrl+Alt+Down) iz menijske vrstice ali pa z desnim klikom miške izberemo Rung/Insert. Kot prikazuje slika 39, bo prazna vrstica vstavljena nad označeno pozicijo.



Slika 39 - Dodajanje vrstice [Vir: lasten]

Nadgrajevanje programa poteka z urejanjem vhodnih in izhodnih spremenljivk. Spremenljivke lahko z uporabo standardnih ukazov brišemo oz. kopiramo/izrežemo iz enega mesta na drugo mesto. Za urejanje lestvičnega programa uporabljamo Edit ukaze, ki se nahajajo v menijski vrstici. Posamezni spremenljivki lahko spremenimo tip in parametre s pritiskom tipke ENTER ali z dvojim klikom na označen bit. V našem primeru bomo na mesto izhoda postavili zastavico (M) in jo kasneje uporabili kot vhod. Posamezne spremenljivke lahko med seboj povežemo z vstavljanjem horizontalnih ali vertikalnih povezav. Povezavo lahko naredimo z vlečenjem miške ali pa si pomagamo z orodno vrstico.

S slike 40 je razvidno, da smo k programu dodali še časovnik (T), katerem smo definirali zakasnilni čas vklopa 5 sekund. Normalno odprti (NO) kontakt tipke I1 pa smo uporabili za prekinitev izvajanja programa. Več o zakasnitvi vklopa v poglavju 3.3.1.



Slika 40 - Uporabniški program lahko nadgradimo [Vir: lasten]

Tako smo dobili novo podobo uporabniškega programa v obliki lestvičnega diagrama.

Pred prenosom programa v krmilnik, ga je potrebno preveriti. To storimo z izbiro ZEN/Program Check iz menijske vrstice. Če med preverjanjem programa ni bilo najdene nobene napake, se prikaže naslednje sporočilo (slika 41).



Slika 41 - Preverjanje programa [Vir: lasten]

Če pozabimo katero od spremenljivk povezati, nas na to opozori programska oprema ZEN.

Lestvični program lahko zaženemo oz. testiramo, še preden ga prenesemo v/na krmilnik. S pomočjo simulacije lahko testiramo delovanje programa. Vse vhodne signale lahko s tem orodjem simuliramo. Tako si lahko ustvarimo virtualno sliko delovanja našega programa.

Funkcijo simulacije zaženemo s klikom na Start/Stop Simulator v orodni vrstici ali z izbiro ZEN/Start/Stop Simulator iz menijske vrstice. Ko se simulator zažene, se ozadje obarva v rumeno barvo. Hkrati se v orodni vrstici omogočijo naslednje tipke: Zaženi, Zaustavi, Prikaži ZEN sliko, Prikaži seznam trenutnih vrednosti, Prikaži uro, Zagon/Izhod simulacije. Primer simulacijskega načina prikazuje slika 42.

Simulacijo zaženemo s klikom na tipko Zaženi (RUN). Simulacijo ustavimo s tipko Zaustavi (STOP). Aktivni so tisti elementi oz. spremenljivke, ki se obarvajo v zeleno.

Ed N	lew Projec	t.zen									_ 🗆 🗵
File	Edit Viev	w Insert	ZEN H	lelp							
	) 📽 日		* *	n R X	2	2 4	2				°a "*
] [	8 -1	> -	٩	<u>a</u> 183	P	B			0	9	
0	M	10					RUN			TO (T)	<b>_</b>
1	Zast I	avica :0		11	•	•		•	•	Časovnik M0	
2	Vho T	od 0 70		Vhod 1				•		Zastavica Q0	
3	Časo I	ovnik 1			•	•		•		Izhod 0 M0	
4	Vho	od 1	• •		•	·		·		Zastavica	
5	-				•	•		1.	•		
Rela Con Set Pres	ay Type: nment: Value: sent Value:										
Exec	ute ladder p	program						20C1D*-	D-V2		Simu //

Slika 42 - Primer simulacije v programski opremi ZEN [Vir: lasten]

V simulatorju lahko s pomočjo klika miške vklopimo ali izklopimo posamezne vhode (spreminjamo stanja vhodnih digitalnih signalov) ter opazujemo, kaj se v tem času dogaja s posameznim izhodom (v simulaciji je izhod predstavljen kot žarnica) oz. izhodi, zastavicami, števci itd. Vhodnim tipkam lahko nastavimo položaje kontaktov, kar prikazuje slika 43. Simulacija programa se lahko izvede brez priključenega krmilnika. Spreminjamo lahko tudi analogne vrednosti (v simulaciji z desnim klikom na analogni vhod nastavimo opcijo "Analog Input" in prikaže se nam drsnik z možnostjo spreminjanja napetosti 0-10 V).



Slika 43 - Vhodu lahko nastavimo več tipov kontakta (NO, NC, Alternativni) [Vir: lasten]

Če imamo k krmilniku priključene in definirane dodatne širitvene V/I enote, se v načinu simulacije poleg krmilnika pojavijo še širitvene V/I enote. Širitvene vhode označujemo z X, izhode z Y. Če imamo v programu (vhodi) definirane operacijske tipke, lahko v simulaciji uporabimo tudi te tipke.

Če smo v programu uporabili tedenski ali koledarski časovnik, lahko uporabimo opcijo Prikaži uro v orodni vrstici. S tem ukazom lahko simuliramo datum in uro. Tako lahko nastavimo čas naprej ali nazaj. Več o tedenskem in koledarskem časovniku v poglavju 5.2.3.

Simulator zaustavimo s klikom na tipko Zagon/Izhod simulacije (Start/Stop Simulator).

Pred prenašanjem programa na krmilnik je potrebno preveriti pravilnost povezave med osebnim računalnikom (COM1) in krmilnikom Omron Zen (slika 44). Krmilnik je potrebno pred priključitvijo ugasniti, šele nato ga lahko povežemo z osebnim računalnikom. Ko imamo povezavo, lahko krmilnik ponovno vklopimo.



Slika 44 - Preverjanje povezave [5]

Nato odpremo projekt, kliknemo na opcijo Poveži se (Go Online) z orodne vrstice ali pa izberemo ZEN/Go Online (kombinacija tipk Ctrl+W) iz menijske vrstice. Prikaže se naslednje sporočilo (slika 45). Kliknemo na tipko V redu.



Slika 45 - Povezovanje [Vir: lasten]

V statusni vrstici vidimo, da se je spremenil status povezave iz privzetega načina Nepovezano (Offline) v način Povezano (Online).

Za prenos uporabniškega programa v krmilnik kliknemo na opcijo Prenesi k ZEN-u (Transfer to Zen) z orodne vrstice oz. izberemo ZEN/Transfer/Transfer to ZEN iz menijske vrstice. Pred prenosom se nam pojavi okno, da smo tik pred prenosom programa v krmilnik (slika 46). Na krmilnik moramo uporabniški program naložiti skupaj z konfiguracijskimi nastavitvami. Za potrditev prenosa kliknemo na tipko OK. Pred nalaganjem podatkov nas lahko programska oprema opozori, če na krmilniku obstaja nek drugi program.

Transfer To ZEN	<u>&gt;</u>
About to download the program to ZEN. Do you wish to continue?	0K
	Cancel
The settings are downloaded too	1
Protection is set	Help

Slika 46 - Prenos uporabniškega programa in konfiguracijskih nastavitev v/na krmilnik [5]

Programska oprema ZEN ponuja tudi možnost, da lahko prenesemo lestvični program iz krmilnika v programsko opremo. Za prenos uporabniškega programa iz krmilnika kliknemo na opcijo Prenesi od ZEN-a (Transfer from Zen) z orodne vrstice ali pa izberemo ZEN/Transfer/Transfer from ZEN iz menijske vrstice. Prikaže se okno, na kateri kliknemo na tipko OK (slika 47). Tako potrdimo prenos programa in konfiguracijskih nastavitev iz krmilnika na osebni računalnik.

Transfer From ZEN	X
About to upload the program from ZEN. Do you wish to continue?	0K
	Cancel
The settings are uploaded too.	Help

Slika 47 - Prenos uporabniškega programa in konfiguracijskih nastavitev iz krmilnika [5]

Pravkar naloženi program v/na krmilnik lahko preverimo, če je enak tistemu, ki ga imamo v programski opremi. Med prenosom programa lahko pride do kakšne napake, ki je morda programa oprema ni zaznala. S klikom na opcijo Primerjaj z ZEN-om (Compare with Zen) z orodne vrstice ali z izbiro ZEN/Transfer/Compare with ZEN iz menijske vrstice lahko to preverimo. Prikaže se okno, na kateri kliknemo na tipko OK (slika 48).

Compare with ZEN	
About to compare the program with ZEN. Do you wish to continue?	ОК
	Cancel
The settings are compared too	
	Help

Slika 48 - Primerjava uporabniških programov [5]

Ko se postopek preverjanja uporabniških programov zaključi, se prikaže eno izmed potrditvenih oken, kar prikazuje slika 49.

Programa sta enaka	Programa sta različna
ZEN Support Software	ZEN Support Software
Verify was successful.	Verify failed.
CK I	CK I

Slika 49 - Zaključek preverjanja uporabniških programov [5]

Če med preverjanjem ni bilo najdenih napak, lahko uporabimo funkcijo, s katero bomo preko osebnega računalnika imeli pregled nad uporabniškim programom. Tako lahko opazujemo stanja različnih spremenljivk.



Slika 50 - Funkcija spremljanja delovanja programa [5]

Delovanja uporabniškega programa v/na krmilniku lahko opazujemo s funkcijo spremljanja (Toggle Monitoring), kar prikazuje slika 50. Slednjo zaženemo s klikom na opcijo Preklopi v spremljanje (Toggle Monitoring) z orodne vrstice ali z izbiro ZEN/Monitor iz menijske vrstice. Ko imamo aktiviran način spremljanja, so aktivne spremenljivke obarvane v zeleno. Med spremljanjem programa se nam prikazujejo trenutne vrednosti spremenljivk ali različna stanja vhodov (vrstica - informacije o posameznem bitu). Monitorski režim se uporablja za opazovanje krmilnika, ki že izvaja program in je priključen na osebni računalnik. Parametrov sedaj ni več mogoče nastavljati, saj je program že v krmilniku, zato moramo ustaviti delujoči program, ga popraviti in ponovno vnesti v krmilnik. Z možnostjo delovanja v povezanem načinu (Online) lahko neposredno med izvajanjem programa preko monitorja osebnega računalnika spremljamo trenutna stanja različnih spremenljivk. Tako lahko opazujemo vhode, izhode, hitri števec in še nekatere druge spremenljivke.

Celoten postopek programiranja imamo na kratko prikazan v priloženi prilogi F.

Uporabniški program lahko natisnemo s klikom miške na opcijo Natisni (Print) v orodni vrstici oz. z izbiro File/Print iz menijske vrstice. Pred tiskanjem lahko naredimo predogled programa v obliki lestvičnega diagrama, kar prikazuje slika 51.

New Project.zen			LØ X
Nateri Ileverteration Erelies anan Svenstor Poyeta Pgnar	staj Zapri		
Nerrow Interference Device Prove Interference Interference Prove Interference Inter			
14			
Stran 1		20C1D*-0-V2	Offine

Slika 51 - Tiskanje uporabniškega programa [Vir: lasten]

### 5.2.2 Interno programiranje

Drugi način programiranja krmilnika Omron Zen je programiranje z ročnim vnosom programa v krmilnik - interno programiranje. Programiramo lahko direktno s pomočjo tipk na sprednji strani krmilnika. Ta način programiranja je uporaben za izkušene programerje. Programiranje s pomočjo direktnega programiranja je zelo zahtevno, omejeno, slabo pregledno. Primerno je za manjša popravila ali dopolnila programa. Veliko bolj primeren in eleganten način programiranja je programiranje s programsko opremo ZEN.

Po priklopu krmilnika na električno napetost, se na zaslonu prikaže glavni meni. Pred pričetkom programiranja lahko nastavimo parametre, kot so jezik, ura, datum, časovno lahko omogočimo prehod na poletni čas, nastavimo lahko komunikacijo (velja za komunikacijski tipa krmilnika) ter nekatere druge nastavitve. Začetni zaslon (glavni meni) krmilnika Omron Zen in ostali meniji so prikazani v priloženi prilogi G.

Programiranje poteka z uporabo operacijskih tipk, ki se nahajajo na sprednji strani krmilnika. S tipkami lahko v krmilnik vnašamo spremenljivke, se premikamo po meniju oz. programu, izbiramo programske možnosti itd. Kot prikazuje slika 52, ima krmilnik 8 operacijskih tipk (B0 - B7). Pomen oznak tipk je prikazan v tabeli 8.



Slika 52 - Operacijske tipke na sprednji strani krmilnika (prevedeno po [4])

Tipka	Funkcija					
	Meniji	Pisanje lestvičnega programa	Nastavitev parametrov	Operacijs ka tipka (naslov)		
DEL DEL 6	/	Izbriše vhode, izhode ter povezovalne in prazne linije.	/	B6		
ALT ALT 7	/	<ul> <li>Preklop med normalno odprtim in normalno zaprtim pogojem.</li> <li>Spremeni povezovalno linijo.</li> <li>Vstavi linijo.</li> </ul>	/	B7		
Gor	Premakne smernik gor in dol	<ul> <li>Premakne smernik gor in dol.</li> <li>Izbira vrste bita in</li> </ul>	<ul> <li>Premakne smernik gor in dol.</li> <li>Sprememba</li> </ul>	B5		
Dol 2		funkcije.	številk in parametrov.	B2		
Levo	/	Premakne smernik levo in desno.	Premakne smernik levo in desno.	B3		
Desno				B4		
ESC (ESC 0	Vrnitev na prejšnji zaslon.	Prekliče nastavitev in se vrne na prejšnjo operacijo.	Prekliče nastavitev in se vrne na prejšnjo operacijo.	В0		
OK OK 1	Izbira menija glede na položaj smernika.	Potrdi nastavitev.	Potrdi nastavitev.	B1		

Tabela 8: Oznake in funkcije operacijskih tipk (prevedeno po [4]
--

Tipke na sprednji (čelni) strani lahko uporabimo kot vhodne signale pri pisanju lestvičnega programa. Tako lahko uporabimo še dodatnih 8 vhodov. V tem primeru je njihova primarna funkcija pomikanja in potrjevanja izključena.

## Interno programiranje

Interno programiranje bomo predstavili na preprostem primeru lestvičnega programa. Lestvični diagram bomo napisali na osnovi preprostega vezja (slika 53).



Slika 53 - Ustvarjanje programa iz preprostega vezja [15]

Program sestavljata dve tipki SW1 in SW2 in en izhod H (žarnica). Tipki SW1 in SW2 priključimo na vhod krmilnika (I0, I1), žarnico H pa na izhod krmilnika (Q0). Slika 54 prikazuje priključitev vhodov, izhoda, napajanja ter shemo delovanja uporabniškega programa.



Slika 54 - Prikaz notranjega delovanja v krmilniku (prevedeno po [15])

Pred pričetkom pisanja lestvičnega programa je priporočljivo, da izbrišemo morebiten shranjen program v/na krmilniku. Z brisanjem ne bomo izbrisali celotnega pomnilnika krmilnika, ampak samo lestvični program. Parametri, kot so jezik, datum, čas in vse ostale nastavitve bodo ostale na krmilniku.

Za preklop v glavni meni pritisnemo tipko OK. Postopek brisanja uporabniškega programa prikazuje slika 55. V meni Program vstopimo z izbiro prve opcije. Izbiro potrdimo s tipko OK. Za brisanje programa izberemo drugo opcijo Izbriši program (DELETE PROG). S pritiskom na tipko OK se prikaže potrditveno sporočilo, kjer izbris programa potrdimo z izbiro Da (YES). Po končanem brisanju, se nam prikaže meni Program.



Slika 55 - Brisanje programa [15]

Z izbiro prve opcije Uredi program (EDIT PROG) začnemo s pisanjem novega lestvičnega programa. Izbiro potrdimo s tipko OK. Prikaže se nam potrditveno sporočilo, pri kateri izberemo Da (YES) ter tako vstopimo v nov program. Na spodnjem zaslonu slike 56, zgornja puščica označuje, v kateri vrstici programa se nahajamo. Spodnja puščica prikazuje utripajoči kazalec, kateri prikazuje tisti del vrstice, ki se bo odzval na vnos bita.



Slika 56 - Ustvarjanje novega programa [15]

Pred nadaljevanjem programiranja je potrebno vedeti, da lahko na eno vrstico napišemo tri vhode in en izhod. Vhodi se nahajajo na levi strani, izhodi pa na desni strani lestvičnega diagrama. Krmilnik dopušča, da lahko napišemo do 96 vrstic. Pri pisanju programa lahko imamo hkrati prikazani dve vrstici. Slika 57 prikazuje mesta posameznih bitov in opisuje nekatere funkcije.



Slika 57 - Pomen oznak v lestvičnem programu pri internemu programiranju (prevedeno po [15])

Spremenljivke lahko dodajamo z leve proti desni strani lestvičnega programa. S pritiskom na tipko OK se pokaže prva vhodna spremenljivka I0. S pomočjo smernih tipk lahko utripajoči kazalec premaknemo na tip ali številko (naslov) bita. Če se nahajamo na položaju I (utripa), lahko s tipkama Gor/Dol (Up/Down) spremenimo tip spremenljivke (privzeti tip vhodne spremenljivke je vhod). S pritiskom na desno tipko lahko utripajoči kazalec pomaknemo iz položaja I na položaj 0. Z uporabo tipk Gor/Dol (Up/Down) lahko spremenimo naslov bita. Pisanje prvega vhoda I0 zaključimo z dvakratnim pritiskom na tipko OK. Utripajoči kazalec se pomakne na naslednji vhodni položaj (slika 58).



Slika 58 - Ustvarjanje prvega vhoda I0 [15]

Na utripajočem položaju lahko dodamo naslednji vhod (slika 59). S pritiskom na tipko OK se prikaže vhodna spremenljivka z istim naslovom in kontaktom, kot prejšnja spremenljivka (I0, NO). S pritiskom alternativne tipke (ALT) lahko preklopimo oz. spremenimo kontakt iz normalno odprtega (NO) v normalno zaprtega (NC). Z desno tipko premaknemo utripajoči kazalec na položaj, kateri prikazuje naslov (številko) spremenljivke. Ker že imamo vhod z naslovom 0, je potrebno drugemu vhodu spremeniti naslov. Z uporabo tipke Gor (Up) spremenimo naslov vhodne spremenljivke (I1). S pritiskom tipke OK se kazalec pomakne na naslednji položaj. Med spremenljivkama I0 in I1 se naredi horizontalna povezava.



Slika 59 - Ustvarjanje drugega vhoda I1 [15]

Ko smo končali s pisanjem vhodnih spremenljivk se lahko pomaknemo na izhod (na vrstico bi lahko dodali še eno vhodno spremenljivko, toda v našem primeru je ne potrebujemo). Ker imamo v našem programu samo dva vhoda, je potrebno naslednji položaj preskočiti oz. narediti povezavo med zadnjo vhodno spremenljivo I1 in izhodom Q0. Horizontalno povezavo omogočimo s pritiskom na alternativno tipko (ALT). Povezavo naredimo s pritiskom na desno tipko (Right), kar prikazuje slika 60.



Slika 60 - Dodajanje horizontalne povezave [15]

Pred dodajanjem izhoda v lestvični program še je potrebno narediti povezavo. S pritiskom na desno tipko premaknemo utripajoči kazalec na položaj izhoda (desna stran lestvičnega programa). S pritiskom na tipko OK se prikaže izhodna spremenljivka, obenem pa se naredi povezava. Če se z utripajočim kazalcem nahajamo na položaju Q, lahko z uporabo tipk Gor/Dol (Up/Down) spremenimo oz. izberemo drugi tip spremenljivke. Z uporabo smernih tipk Desno/Levo (Right/Left) lahko kazalec premikamo, z uporabo tipk Gor/Dol (Up/Down) pa lahko izberemo funkcijo izhoda ter spremenimo naslov izhodne spremenljivke. Prvo vrstico zaključimo z dvakratnim pritiskom na tipko OK. Kot prikazuje slika 61 imamo na vrstici dva vhoda in en izhod. Utripajoči kazalec se pomakne na začetek naslednje vrstice.



Slika 61 - Dodajanje izhoda Q0 [15]

V internem programiranju moramo vsako povezavo narediti ročno s pomočjo operacijskih tipk. Kot prikazuje slika 62, imamo na voljo dve povezovalni liniji - horizontalno in vertikalno. Med programiranjem lahko z alternativno tipko (ALT) izberemo želeno povezovalno linijo.



Slika 62 - Izbira povezovalne linije (prevedeno po [15])

Uporabniški program bomo zaključili z dodajanjem vhodne spremenljivke na drugo vrstico (slika 63). S pritiskom na tipko OK se prikaže vhodna spremenljivka I0. Ker bomo uporabili izhod Q0 kot vhodno spremenljivko, je potrebno spremenljivki I0 spremeniti tip. Spremembo naredimo tako, da se s kazalcem premaknemo na položaj I, nato pa s pomočjo tipke Gor (Up) spremenimo tip spremenljivke v Izhod (Q). Z dvakratnim pritiskom na

tipko OK potrdimo paralelni vhod Q0. Utripajoči kazalec se pomakne na naslednji vhodni položaj v vrstici. Vhodno spremenljivko Q0 bomo povezali k prvi vrstici. S pritiskom alternativne tipke (ALT) naredimo horizontalno linijo med prvo spremenljivo in drugim položajem v vrstici.



Slika 63 - Uporaba izhoda kot vhodno spremenljivko [15]

Povezavo druge vrstice k prvi vrstici zaključimo s pritiskom na tipko Gor (Up) - ustvari se vertikalna povezovalna linija (oznaka + pomeni sekajočo linijo). S tipko OK potrdimo spremembo ter tako zaključimo s povezovanjem. Tako smo preko horizontalne in vertikalne linije povezali vhodno spremenljivko k prvi vrstici (slika 64).



Slika 64 - Paralelna vezava spremenljivke [15]

Lestvično interno programiranje zaključimo s tipko ESC. S ponovnim pritiskom tipke ESC se vrnemo v glavni meni (enako velja med izvajanjem programa).

Pri programiranju v obeh načinih (z uporabo programske opreme ZEN, interno) imamo na razpolago 17 spremenljivk (bitov). Vsaka izmed njih ima svojo funkcijo ter namen uporabe v avtomatizaciji. Spremenljivke so prikazane in opisane v priloženi prilogi J. Glavni meni zapustimo s tipko ESC. Na zaslonu krmilnika Omron Zen se nam prikažejo naslednji podatki, kar prikazuje in opisuje slika 65.



Slika 65 - Prikaz glavnega zaslona (prevedeno po [15])

Shranjeni program zaženemo tako, da spremenimo začetni zaustavljeni način (STOP) v način delovanja (RUN). S pritiskom na tipko OK, se nam prikaže glavni meni, v katerem s tipko Dol (Down) pomaknemo utripajoči kazalec na opcijo Zaženi (RUN). S tipko OK potrdimo zagon lestvičnega programa.

Delovanje programa lahko spremljamo preko glavnega zaslona ali pa izberemo opcijo Spremljanje (MONITOR). Oba načina spremljanja delovanja programa prikazuje in opisuje slika 66.



Slika 66 - Delovanje programa lahko spremljamo (prevedeno po [15])

Uporabniški program lahko tudi spremenimo. Posameznim spremenljivkam lahko spremenimo tip, naslov, funkcijo ali parametre. Npr. na mesto vhodne spremenljivke Q0, lahko postavimo zastavico M1. Spremembo naredimo tako, da premaknemo utripajoči kazalec na mesto spremenljivke, kjer želimo narediti spremembo (tip ali naslov bita). Če postavimo kazalec na tip bita, lahko s pritiskom na tipko OK naredimo spremenimo. Z uporabo tipk Gor/Dol (Up/Down) izberemo spremenljivko, katero želimo uporabiti. V našem primeru izberemo zastavico (M). Zastavici še lahko določimo naslov. To storimo, da premaknemo utripajoči kazalec na mesto naslova spremenljivke ter nato s tipkama Gor/Dol (Up/Down) določimo naslov (1). S tipko OK potrdimo spremembo. Na ta način zamenjamo vhodno spremenljivko Q0 z zastavico M1. Na podoben način lahko spremenimo oz. vplivamo tudi na druge spremenljivke. Izhodu lahko spremenimo tip, naslov ali funkcijo. Spremenljivkam, kot so časovniki, števci ali primerjalniki lahko nastavimo tudi različne parametre (več v pogl. 5.2.3).

V programu lahko brišemo spremenljivke (vhodi, izhodi) ali povezovalne linije. To storimo tako, da premaknemo utripajoči kazalec na položaj spremenljivke ali povezovalne linije, katero želimo izbrisati ter nato pritisnemo tipko Izbriši (DEL). V uporabniški program lahko dodamo tudi novo vrstico. Prazno vrstico dodamo, da najprej premaknemo utripajoči kazalec na začetek (naslednje) vrstice, kjer želimo vstaviti novo vrstico. Prazno vrstico dodamo s pritiskom na alternativno tipko (ALT). V primeru, če pa želimo izbrisati kakšno izmed vrstic, storimo to tako, da pomaknemo utripajoči kazalec na začetek vrstice (katero želimo izbrisati), ter pritisnemo na tipko Izbriši (DEL) [15].

## 5.2.3 Uporaba časovnikov, števcev in ostalih funkcij

V avtomatizaciji se pojavlja potreba po štetju različnih dogodkov, časovno pogojenih krmiljenih procesih itd. Pogosto se moramo prilagoditi potrebam nekih avtomatiziranih procesov, zunanjih okoliščin... Pri programiranju lahko prilagodimo delovanje programa z uporabo časovnikov, števcev, primerjalnikov ter še z nekaterimi drugimi funkcijami.

Posamezni postopki so v avtomatizaciji časovno pogojeni. Postopki se morajo začeti oz. končati po nekem pretečenem času, pogosto morajo trajati nek predpisani čas, včasih se pojavi tudi potreba po merjenju časa med izbranimi dogodki [2].

Za realizacijo tega imamo pri krmilniku Omron Zen na razpolago 4 skupine časovnikov:

- Časovnik: T0 Tf;
- Držalni časovnik: #0 #7;
- Tedenski časovnik: @0 @f;
- Koledarski časovnik: \*0 \*f.

Na voljo imamo 56 časovnikov, med katerimi je 16 časovnikov iz skupine osnovnih časovnikov, 8 časovnikov je iz držalnih časovnikov, 16 časovnikov lahko uporabimo kot tedenske in 16 za koledarske časovnike.

Časovnik (T) lahko uporabimo na 5 načinov, in sicer kot zakasnitev vklopa, zakasnitev izklopa, impulzni en-strel, utripajoči impulzni in dvojni časovnik. Vsak izmed naštetih načinov ima določeno funkcijo uporabe. Vrste časovnih funkcij prikazuje slika 67.

Edit Contact		×
# @ *	C F A P C Y M H B	G G
Relay Name: Timer	Allow monitoring	
Contact:	Normally Open + +     Normally Closed +/+	
Relay No.:	1~0 <u>÷</u> 0	
Comment:		
Timer Type:	Image: Constant of the second sec	
Time Setting Value:	Second:10Millisecond(S)     0     1	00:01~99:99
ON time	C Minute:Second(M:S) C Time:Minute(H:M)	
OFF time	Second:10Millisecond(S)     1     Minute:Second(M:S)     Time:Minute(H:M)	00:01~99:99

Slika 67 - Vrste časovnih funkcij [Vir: lasten]

#### Delovanje časovnih funkcij

Časovni funkciji zakasnitev vklopa (On-delay Timer) in izklopa (Off-delay Timer) sta pojasnjeni v pogl. 3.3.1 in 3.3.2. Časovno funkcijo - impulzni en-strel (One-shot Pulse) uporabimo takrat, ko želimo, da pri prehodu vhodnega stanja iz "0" v "1" ostane izhod določen čas aktiven (širino impulza nastavimo v nastavitvah). Z utripajočim impulznim časovnikom (Flashing Pulse) lahko ponavljajoče v določenem časovnem presledku vklopimo in izklopimo izhod. Kako dolgo bo izhod aktiviran oz. deaktiviran je odvisno od nastavitve časovnega presledka, ki je lahko podan v sekundah, minutah ali urah. Funkcija uporabe dvojnega časovnika (Twin Timer) je podobna kot pri utripajočem časovniku z razliko, da lahko neodvisno nastavimo čas vklopa in izklopa.

Držalni časovnik se aktivira, ko trenutna vrednost (Present value) doseže oz. preseže nastavljeno vrednost. Trenutna vrednost se poveča vsakič, ko se aktivira oz. sproži vhod držalnega časovnika. Trenutna vrednost se ohrani, ko je vhod v neaktivnem stanju. Izhod je tako dolgo v aktivnem stanju, dokler ne aktiviramo vhoda R (brisanje). Pri držalnem časovniku je podprta samo funkcija zakasnitve vklopa.

Vsakemu časovniku moramo nastaviti časovno enoto (sekunda, minuta ali ura), po kateri se bo posamezna časovna funkcija ravnala. S pomočjo časovnikov lahko realiziramo različne časovne premike signalov.

V primeru lestvičnega programa sta prikazana vhoda (I0, I1) in izhoda za proženje ter brisanje časovnika (slika 68). Spremenljivka T0 aktivira izhod časovnika. Parametre in funkcije nastavimo v programu.



Slika 68 - Nastavitve časovnika v lestvičnemu programu (prevedeno po [15])

Način spremljanja parametrov časovnika med delovanjem uporabniškega programa prikazuje slika 69.


Slika 69 - Način spremljanja na krmilniku (prevedeno po [15])

Tedenski časovnik ima dodeljene tri operacije, kar prikazuje tabela 9. Operacije se razlikujejo po dnevu v tednu, času in izhodnih nastavitvah časa. Ena izmed operacij je lahko izbrana za tedenski časovnik.

Vrsta operacije	N	Normalno
	D	Med dnevi
	Р	Impulzna
Dan	Začetni dan	Ned/Pon/Tor/Sre/Čet/Pet/Sob
	Končni dan	Ned/Pon/Tor/Sre/Čet/Pet/Sob/Noben
Čas	Začetni čas	00:00 - 23:59
	Končni čas	00:00 - 23:59

Tabela 9: Operacije tedenskega časovnika (prevedeno po [15])

V primeru lestvičnega programa je prikazan vhod tedenskega časovnika (@0), ki aktivira izhod. Parametre in operacijo tedenskega časovnika nastavimo v programu. Način spremljanja parametrov tedenskega časovnika med delovanjem uporabniškega programa prikazuje slika 70.





Koledarski časovnik uporabimo takrat, ko želimo določiti med katerima datumoma naj bo izhod aktiven. Na sliki 71 je prikazan primer lestvičnega programa z uporabo koledarskega časovnika. Definirani vhod (\*0) koledarskega časovnika aktivira izhod (Q0).



Slika 71 - Spremljanje delovanja koledarskega časovnika (prevedeno po [15])

V industriji imamo tudi pogosto opravka s štetjem različnih dogodkov (štetje impulzov, kosov). Tipična področja uporabe so kovinska industrija, tekstilna industrija, papirna industrija in industrija pakiranja. Npr. v pivovarni nas zanima število steklenic, ki jih je napolnila polnilna linija, na parkirnem prostoru nas lahko zanima število parkiranih avtomobilov [2]. Napisali bi lahko program za štetje dogodkov z uporabo funkcije štetja (T). Krmilniki imajo za programiranje pripravljene različne spremenljivke in funkcije za uporabo.

Tako imamo pri krmilniku Omron Zen na voljo dve različici števca:

- Števec: na voljo imamo 16 števcev, poimenovanih od C0 do Cf;
- 8-bitni digitalni števec: F0.

Števec nam omogoča štetje navzgor in navzdol v obsegu od 1 do 9999 dogodkov, vanj lahko postavimo neko začetno vrednost ter določimo funkcijo (smer štetja, lahko ga tudi brišemo - postavimo na 0), kar prikazuje slika 72.

Edit Coil		×
Q Y M Relay Name: Counter	н  т  # С	FD
Function:	<ul> <li>Counter</li> <li>Designate count direction</li> <li>Reset operation</li> </ul>	C D R
Relay No.: Comment: Counter Set Value:	0 <u>÷</u> 0~f  	

Slika 72 - Določitev funkcije števca [Vir: lasten]

#### Delovanje števca

Števec se aktivira, ko trenutna vrednost doseže oz. preseže nastavljeno vrednost. Trenutna vrednost se stopnično poveča (ali zmanjša) vsakič, ko vhodni števec C (CNT) zazna oz. zabeleži dogodek (se aktivira). Izhod števca zbrišemo (ponastavimo) z aktiviranjem vhoda R (RES). Koraki štetja se ne upoštevajo tako dolgo, dokler je vhod R v stanju "1". Z aktiviranjem vhoda D (DIR) spremenimo smer štetja iz štetja navzgor v štetje navzdol (števec odšteje od števila dogodkov en dogodek). Upoštevani so vsi možni vhodi, kar pa pri programiranju ni nujno. Uporabimo lahko le tiste, ki jih potrebujemo. Delovanje števca prikazuje slika 73.



Slika 73 - Časovni potek signalov pri števcu [15]

V primeru lestvičnega programa so prikazani vhodi (I0, I1, I3) in izhodi za vhodni števec (štetje dogodkov), smer štetja ter za brisanje števca (slika 74). Spremenljivka C0 aktivira izhod števca. Parametre in funkcije nastavimo v programu.



Slika 74 - Nastavitve štetja v lestvičnemu programu (prevedeno po [15])

8-bitni digitalni števec je namenjen štetju hitrih dogodkov v obsegu od 1 do 99999999 dogodkov. Vanj lahko postavimo začetno vrednost, določimo funkcijo ter nastavimo

hitrost štetja. Največja hitrost štetja 8-bitnega digitalnega števca je 150 Hz (ne glede na to, ali se uporablja vhodni filter). Uporabimo lahko dve hitrosti štetja (tabela 10).

Tabela 10: Hitrosti štetja pri 8-bitnemu digitalnemu števcu (prevedeno po [15])

Hitrost števca	Η	Visoka hitrost (150 Hz). Podprto za tip Omron Zen krmilnika z
(štetja)		enosmerno DC priključitveno napetostjo.
	L	Nizka hitrost (odvisno od tega, kakšen vhodni filter uporabljamo).

Način spremljanja parametrov števca med delovanjem uporabniškega programa prikazuje slika 75 (v internemu načinu programiranja).





Poleg časovnikov in števcev še imamo na razpolago naslednje funkcije:

- Prikaz sporočil bitov (spremenljivk): na voljo imamo 16 bitov (D0 Df);
- Analogni primerjalnik: na voljo imamo 4 analogne primerjalnike (A0 A3);
- Časovni/števni primerjalnik: na voljo imamo 16 primerjalnikov (P0 Pf);
- 8-bitni števni primerjalnik: na voljo imamo 4 primerjalnike (G0 G3);
- Uporaba operacijskih tipk vhodnih bitov: na voljo imamo 8 bitov (B0 B7).

Z uporabo funkcije prikaza sporočil lahko na LCD zaslonu krmilnika prikažemo uporabniško sporočilo, čas prikazanega sporočila, trenutno vrednost časovnika/števca ali analogno pretvorbeno vrednost. Parametre nastavimo v programu. Na istem zaslonu lahko prikažemo hkrati več prikazovalnih funkcij. V praksi bi funkcijo lahko uporabili, da bi nas opozorila na določena stanja oz. spremembe v poteku avtomatiziranega procesa (npr. motor vklopljen). Na sliki 76 je prikazan zaslon v načinu spremljanja ter primer uporabe te funkcije v lestvičnemu programu.



Slika 76 - Funkcija prikaza sporočil - bitov (spremenljivk) (prevedeno po [15])

V programiranju lahko izberemo objekt, katerega želimo prikazati. Izbiramo lahko med devetimi prikazi objekta, kar je razvidno iz tabele 11.

CHR	Znaki, simboli (največ 12 znakov)
DAT	Mesec/dan (5 mestno število)
DAT1	Dan/mesec (5 mestno število)
CLK	Ura/minute (5 mestno število)
Ia, Ib	Analogna pretvorba (4 mestno število)
T0 - Tf	Trenutna vrednost časovnika (5 mestno število)
#0 - #7	Trenutna vrednost držalnega časovnika (5 mestno število)
C0 - Cf	Trenutna vrednost števca (5 mestno število)
F0	Trenutna vrednost 8-bitnega digitalnega števca (8 mestno število)

Tabela 11: Izbira prikazanega objekta (prevedeno po [15])

Primer uporabe funkcije prikaza sporočil prikazuje in opisuje slika 77. Na zaslonu krmilnika se izpišejo tri sporočila: dan/mesec, ura/minute in definirana oznaka (PIT#2).

05/02 20:18 PIT#2 EMPTY!		
Izpis datuma in u pojavila napaka	ure, ko se je	
Podrobnosti nas	tavitev:	
D1 L3 TRG X00Y0 (00) DAT [) ]][ ]	D2 L3 A TRG X06Y0 (00) CLK [> !][ ]	D3 L3 A TRG X00Y2 (00) CHR [0PQ][PIT#2]

Slika 77 - Primer uporabe funkcije prikaza sporočil

(prevedeno po [15])

Analogni primerjalnik lahko uporabimo za primerjanje analognih vhodov (Ia, Ib) oz. analognih vhodnih napetosti med 0 in 10 V. Funkcija analognega primerjanja je na voljo samo izvedbam krmilnika Omron Zen z enosmernim tipom priključitve napajanja. Na sliki 78 je prikazan način priključitve analogne naprave na krmilnik Omron Zen.



Slika 78 - Uporaba analognih signalov (Ia, Ib) (prevedeno po [15])

Analogne signale lahko primerjamo na tri načine (tabela 12). Primerjamo lahko analogna vhoda med seboj, kot tudi posamezni analogni vhod z neko določeno konstanto. Slednjo definiramo oz. podamo v nastavitvah analognega primerjalnika (med programiranjem). Analognemu primerjalniku moramo za primerjanje signalov podati operater (večje ali enako; manjše ali enako). Izhod se bo odzval tako, kot bomo v programu definirali operater med dvema vrednostima. Več o analognih vhodih v pogl. 5.3.1.1.

Tabela 12: Nastavitev parametrov za analogni primerjalnik (prevedeno po [15])
---

Primerjava	1	Ia (I4): Analogni vhod 1Primerjava vzorcev:
podatkov		Ib (I5): Analogni vhod 2 • Primerjava med Ia in Ib.
	2	Ib (I5): Analogni vhod 2• Primerjava med Ia in konstanto.
		Konstanta: 00.0 - 10.5 • Primerjava med Ib in konstanto.
Operater		>=: Izhod se aktivira, ko je primerjalni podatek 1 ≥
		primerjalnemu podatku 2
		<≕ Izhod se aktivira, ko je primerjalni podatek 1 ≤
		primerjalnemu podatku 2

Na sliki 79 je prikazan zaslon v načinu spremljanja ter primer uporabe funkcije analognega primerjanja v lestvičnemu programu. Izhod analognega primerjalnika se bo aktiviral, ko bo vhod (A0) dosegel vrednost 5.2 V.



Slika 79 - Prikaz uporabe funkcije analognega primerjalnika (prevedeno po [15])

Primerjalnik lahko uporabimo tudi za primerjanje trenutnih vrednosti časovnikov, držalnih časovnikov ali števcev. To funkcijo imenujemo časovni/števni primerjalnik (P). Primerjamo lahko samo med istim tipom funkcije (T - časovnik, # - držalni časovnik ali C - števec), vrednost le-te pa lahko primerjamo tudi z definirano konstanto, kar prikazuje tabela 13.

Primer uporabe funkcije - držalno-časovni primerjalnik s konstanto bomo prikazali v lestvičnemu programu (slika 80). Vsakič, ko se aktivira vhod, se s tem poveča trenutna vrednost držalnega časovnika (#0). Trenutna vrednost se sešteva tako dolgo, dokler ne doseže meje 12,34 sekund - definirana konstanta. Posledica tega je aktiviranje izhoda.



Slika 80 - Primerjanje držalnega časovnika s konstanto (prevedeno po [15])

V nastavitvah moramo primerjalniku nastaviti vrsto in operater primerjanja (>=). V našem primeru, ko imamo uporabljeno konstanto, še moramo določiti vrednost le-te (12 sekund in 34 milisekund).

Vrsta		T: Časovnik		
primerjave		#: Držalni časovnik		
		C: Števec		
Primerjava	1	T: T0 - Tf	Primerjava vzorcev:	
podatkov		#: #0 - #7	• Primerjava med T in	
		C: C0 - Cf	T ali med T in	
	2	T: T0 - Tf	konstanto.	
		#: #0 - #7	• Primerjava med # in	
		C: C0 - Cf	# ali med # in	
		Konstanta:	konstanto.	
		• 00.00 do 99.99 (primerjava	• Primerjava med C in	
		časovnikov in držalnih časovnikov).	C ali med C in	
		• 0000 do 9999 (primerjava števcev).	konstanto.	
Operater		>=: Izhod se aktivira, ko je primerjalni podatek 1 ≥ primerjalnemu		
		podatku 2		
		<=: Izhod se aktivira, ko je primerjalni podatek 1 ≤ primerjalnemu		
		podatku 2		

Tabela 13: Časovni/števni primerjalnik (prevedeno po [15])

Naslednja funkcija se imenuje 8-bitni števni primerjalnik (G). S to funkcijo lahko primerjamo vrednost 8-bitnega digitalnega števca s konstantno vrednostjo. Na sliki 81 je prikazan lestvični program v načinu internega programiranja z uporabo 8-bitnega števnega primerjalnika. Primerjalnik dobi signal od 8-bitnega števca (F0). Z vsakim impulzom 8-bitnega števca se vrednost primerjalnika povečuje (sešteva se). Izhod se aktivira, ko primerjalnik (G0) doseže vrednost 12000000. Primerjalniku je potrebno v nastavitvah definirati operater in konstanto (12000000). Delovanje operatera opisuje tabela 14.



Slika 81 - Uporaba 8-bitnega števnega primerjalnika (prevedeno po [15])

Primerjava podatkov	Konstanta: 00000000 do 99999999	
Operater	>=: Izhod se aktivira, ko je trenutna vrednost 8-bitnega	
	števca ≥ konstantni vrednosti	
	<=: Izhod se aktivira, ko je trenutna vrednost 8-bitnega	
	števca ≤ konstantni vrednosti	

Tabela 14: Nastavitev konstante in operatera (prevedeno po [15])

Operacijske tipke na sprednji strani krmilnika so dodeljene operacijam vhodnih bitov. To pomeni, da lahko funkcijo operacijskih tipk med programiranje uporabimo kot kontakte vhodnih bitov. Primer uporabe operacijskih tipk, je prikazan na sliki 82. S kombinacijo tipk B6+B7 lahko brišemo (ponastavimo) trenutno vrednost števca (C2) in stanje držalnega releja (H5).



Slika 82 - Operacijske tipke na sprednji strani krmilnika kot vhodni biti [15]

Časovni poteki signalov posameznih časovnih, števnih in vseh ostalih funkcij, ki so nam na voljo pri programiranju so prikazani v priloženi prilogi J.

Vse opisane funkcije nas spremljajo od začetka programiranja oz. ustvarjanja projekta pa vse do njegovega zaključka. Funkcije imajo vgrajeno uro realnega časa. Pri programiranju je samo LED tip Omron Zen krmilnika omejen s funkcijami (ne podpira koledarskega/tedenskega časovnika in 8-bitnega digitalnega števca) [15].

## 5.3 Priključitev krmilnika Omron Zen

## 5.3.1 Vhodne povezave

Vhodne signale iz procesa posredujejo različni (priključeni) senzorji in inkrementalni dajalniki. Pri avtomatizaciji imamo največ opravka predvsem s senzorji. Izbiramo lahko med senzorji, ki delujejo z enosmernim DC napajanjem in senzorji z izmeničnim AC napajanjem.

Senzorji z enosmernim DC napajanjem so lahko v dvožični ali trožični izvedbi. Dvožična izvedba se uporablja le pri kapacitivnih in induktivnih senzorjih. Pri tej izvedbi je senzor priključen zaporedno k bremenu (paziti je treba na polariteto senzorja). Paziti moramo tudi na tokovno omejitev senzorja in največjo dovoljeno napetost, ki jo lahko priključimo na senzor.

Slika 83 prikazuje dvožično vezavo senzorja, pri kateri moramo senzor najprej vezati na rele, kateri bo vklapljal večje porabnike. Pri vezavi senzorja na vhod PLK-ja ni potrebe po "posredniku", zato lahko senzor vežemo direktno na vhod. PLK ima veliko vhodno upornost (teče majhen tok) in senzorja ne moremo uničiti.



Slika 83 - Dvožična vezava senzorja [Vir: lasten]

Pri standardni trožični izvedbi razlikujemo med NPN in PNP tipom vezave. Pri NPN tipu je breme priključeno na pozitivni (+) pol, medtem ko je PNP tip priključen na maso. Razlikujemo več tipov trožičnih senzorjev: induktivni, kapacitivni, optični itd. Za zaznavanje končnih položajev zelo pogosto uporabljamo Reed-kontakte (brezkontaktna, magnetna stikala). Trožično vezavo senzorja prikazuje slika 84, pri kateri moramo senzor najprej vezati na rele, kateri bo vklapljal večje porabnike. Podobno kot pri dvožični izvedbi lahko senzor vežemo direktno na vhod, le da imamo pri trožični izvedbi še skupno maso.



Slika 84 - Trožična vezava senzorja [Vir: lasten]

Glede na tip krmilnika lahko imamo ožičenje krmilnika z izmenično AC ali z enosmerno DC napetostjo.

Pri priključevanju izmeničnega AC napajanja, vežemo linijski oz. fazni vodnik (črna ali rjava barva) na sponko L, nevtralni vodnik (modra barva) pa na N. Priključevanje izmeničnega napajanja k krmilniku prikazuje slika 85. Pri priključitvi moramo biti pozorni, da ne zamenjamo sponk.



Slika 85 - Ožičenje krmilnika in širitvene V/I enote z izmenično AC napetostjo [4]

Vhodna impedanca pri izmeničnih vhodih znaša 680 k $\Omega$ , vhodni tok pa 0.35 mA pri izmeničnem AC napajanju 230 V. Odzivni čas in visoko/nizko stanje izmeničnih vhodov je podano v tabeli 15.

Postavka	Specifikacije	Vezje
Vhodna napetost	100 - 230 V AC +10%, -15%,	
	50/60 Hz	
Vhodna impedanca	680 kΩ	ri
Vhodni tok	0.15 mA/100 V AC, 0.35	5 IN 330 κΩ 300 κΩ
	mA/240 V AC	IN ξ51 kΩ Internal circuit
Visoko stanje	80 V AC min.	
Nizko stanje	25 V AC max.	
Odzivni čas vklopa	50 ms ali 70 ms/100 V AC	
Odzivni čas izklopa	100 ms ali 120 ms/240 V AC	

Tabela 15: Specifikacije za izmenične AC vhode (vhodi I0 - I9) (prevedeno po [4])

Pri priključevanju enosmernega DC napajanja (12 - 24 V DC), povežemo pozitivni vodnik (rdeča barva) s sponko "+", negativni vodnik (črna barva) pa s sponko "-".

Pri vezavi krmilnika z enosmerno DC napetostjo moramo biti pozorni na tip povezave - PNP ali NPN. Pri PNP vhodni vezavi (slika 86) imajo vhodi skupni "+" (izvor), pri NPN vezavi (slika 87) pa imajo vhodi skupni "-" (ponor).



Slika 86 - Ožičenje krmilnika z enosmerno DC napetostjo (PNP povezava) [4]





Slika 87 - Ožičenje krmilnika z enosmerno DC napetostjo (NPN povezava) [4]

Vhodna impedanca znaša 5.3 k $\Omega$ , vhodni tok pa 4.5 mA pri enosmernem DC napajanju 24 V. Odzivni čas in visoko/nizko stanje enosmernih vhodov je podano v tabeli 16.

Postavka	Specifikacije	Vezje
Vhodna napetost	12 - 24 V DC +20%, -10%	
Vhodna impedanca	5.3 kΩ	l
Vhodni tok	4.5 mA/24 V DC	5.1 kΩ 1.8 kΩ
Visoko stanje	8 V DC min.	12 to 24 VDC IN
Nizko stanje	5 V DC max.	
Odzivni čas vklopa	15 ms ali 50 ms	
Odzivni čas izklopa	(z nastavitvami vhodnega filtra)	

Tabela 16: Specifikacije za enosmerne DC vhode (vhodi I0 - I9) (prevedeno po [4])

## 5.3.1.1 Analogni vhodi

Analogni vhodi lahko registrirajo zvezne signale. Preko analognih vhodov lahko reguliramo temperaturo ali spreminjamo napetost, kar pri digitalnih vhodih ne moremo. Krmilnik Omron Zen podpira analogne napetostne signale. PLK razume le digitalne signale, zato je pri analognih vhodih potreba po pretvorbi signalov oz. potreba po uporabi A/D oz. D/A pretvornikov. Več o teoriji analognih vhodov v pogl. 2.2 (vhodne in izhodne enote).

Slika 88 prikazuje povezavo analognih vhodov Ia in Ib k enosmernemu DC napajanju.



Note: When connecting an analog input device, always connect the negative side to the COM terminal.

Slika 88 - Ožičenje analognih vhodov z enosmerno DC napetostjo [4]

Vhodno napetostno območje pri analognih vhodih znaša od 0 do 10 V, zunanja vhodna impedanca pa 100 k $\Omega$ . Celotna točnost analognih vhodov je ±1.5 %, kar krmilnik Omron Zen uvršča na sam vrh v primerjavi z drugimi kompaktnimi PLK-ji. V tabeli 17 sta še podana podatka za resolucijo in AD pretvorbo podatkov.

Postavka	Specifikacije	Vezje
Vhodna napetost	0 - 10 V	
Zunanja vhodna	100 kΩ min.	
impedanca		
Resolucija	0.1 V (1/100 FS)	
Natančnost	±1.5% FS (pri sobni	
	temperaturi)	12 to 24 VDC
AD pretvorba	0 - 10.5 V (v	
podatkov	poviških 0.1 V)	

Tabela 17: Specifikacije za analogna vhoda Ia in Ib (prevedeno po [4])

### 5.3.2 Izhodne povezave

#### 5.3.2.1 Enote s tranzistorskim izhodom

Pri tranzistorski vezavi je mogoč samo skupen "+" (izvor). Na tranzistorske izhode lahko priključimo 24 V DC porabnike z največjo dovoljeno tokovno obremenitvijo 500 mA. Enako velja tudi za širitvene V/I enote s tranzistorskim izhodom. V primeru če je izhod preveč obremenjen, lahko pride do kratkega stika in posledično do uničenja (izhoda) krmilnika. Zato je priporočljivo, da vstavimo oz. priključimo zaščitno varovalko 2 A k posameznemu izhodu. Enote s tranzistorskim izhodom imajo zelo kratek odzivni čas.



Slika 89 - Enote s tranzistorskim izhodom [4]

Slika 89 prikazuje izhodno tranzistorsko vezavo. V tabeli 18 so podani še podatki o odzivnosti (čas vklopa ali izklopa) in o največjem prepustnem odprtozančnem toku tranzistorskih izhodov.

Postavka	Specifikacije	Vezje
Največja zmogljivost	24 V DC	Each circuit is configured with an independent common circuit
preklapljanja	+20%, 500 mA	
Odvodni tok	0.1 mA max.	
Preostala napetost	1.5 V max.	
Odzivni čas vklopa	1 ms max.	circuit
Odzivni čas izklopa	1 ms max.	Q4/Q6 CUIO points only Q5/Q7

Tabela 18: Specifikacije za tranzistorski izhod (prevedeno po [4])

### 5.3.2.2 Enote z relejskim izhodom

Na relejske izhode lahko priključimo 230 V AC ali 24 V DC porabnike s skupno porabo do največ 40 A (velja za izvedbe krmilnika z 20 V/I). Največja dovoljena tokovna obremenitev za posamezen izhod je 8 A za porabnike 230 V AC oz. 5 A za porabnike 24 V DC. Za širitvene V/I enote je največja dovoljena tokovna obremenitev za vse izhode 20 A (na enoto). Izhodi so med seboj izolirani. Slika 90 prikazuje izhodno relejsko vezavo.



Slika 90 - Enote z relejskim izhodom [4]

Relejski izhodi služijo za galvansko ločevanje krmilne napetosti od visokih uporabniških napetosti in tokov. Odzivni čas relejskih izhodov je v primerjavi s tranzistorskimi izhodi daljši, zato se uporabljajo predvsem v krmilnih sistemih, kjer odzivni čas ni pomemben.

Podatki o življenjski dobi kontaktov ter o odzivnem času relejskih izhodov so podani v tabeli 19.

Posta	ivka	Specifikacije	Vezje
Največja zm	ogljivost	250 V AC/8 A	
preklapljanja		(uporovno breme: $\cos \Phi = 1$ )	
		24 V DC/5 A	
		(uporovno breme)	
		Skupna poraba izhodov:	
		• Tip krmilnika z 10 V/I:	
		20 A max. (15 A max. za	
		komunikacijski tip)	
		• Tip krmilnika z 20 V/I:	
		40 A max.	
		• Širitvene V/I enote:	
		20 A max.	Models with 20 I/O points
Najmanjša z	mogljivost	5 V DC/10 mA	
preklapljanja		(uporovno breme)	······
Življenjska	Električni	Uporovno breme: 50,000	
doba releja		ciklov ( $\cos\Phi = 1$ )	
		Induktivno breme: 50,000	
		ciklov ( $\cos\Phi = 0.4$ )	
	Mehanski	10 milijonov ciklov	
Odzivni čas vklopa		15 ms max.	
Odzivni čas izklopa		5 ms max.	

Tabela 19: Specifikacije za relejski izhod (prevedeno po [4])

# 6. PRAKTIČNA PRIMERA ZA UČENJE UPORABE PLK OMRON ZEN

Uporabo krmilnika Omron Zen bomo predstavil na primeru koračnega in kombinacijskega krmilja. Za lažje razumevanje primerov je potrebno razjasniti razliko med obema tipoma krmilij. Krmilje lahko izvedemo kot koračno ali pa kot kombinacijsko (logično). Če si proizvodni postopki sledijo v nekem stalnem in vnaprej določenem zaporedju (korak za korakom), govorimo o koračnemu krmiljenju. Medtem, ko pa je osnovna značilnost kombinacijskega krmilja v tem, da so logična stanja posameznih izhodov odvisna le od trenutne kombinacije stanj vhodnih spremenljivk (deluje kot preslikava vhod - izhod). Kombinacijsko krmilje realiziramo tako, da s pomočjo pravilnostne tabele zapišemo logično enačbo, katera bo upoštevala vse možne vhodne/izhodne kombinacije. Pravilnostna tabela podaja odvisnost izhodnih funkcij od vhodnih kombinacij. Vezje lahko realiziramo z logičnimi funkcijami/elementi (IN, ALI, NE).

Pri projektiranju dajemo v primerjavi z kombinacijskim krmiljem prednost koračnemu krmilju, ker:

- je osnutek veliko enostavnejši;
- uporabniški program je preglednejši, bolj razumljiv;
- razmeroma enostavno se da naknadno spremeniti funkcijski proces;
- se da v primeru napak enostavno lokalizirati in odpraviti napako.

Koračna krmilja so zgrajena tako, da se v danem trenutku izvaja le ena operacija, ki traja tako dolgo, dokler je ne potrdi vhod (senzor) za dano operacijo (temperatura, pomik, nivo tekočine v rezervoarju). Potrjen predhodni korak (senzor) omogoča vklop naslednjega koraka. Pri tej vrsti krmilij so izhodi odvisni od vhodov in notranjih stanj krmilja. Notranja stanja so lege ročic, stanja časovnikov, števcev... Koračno krmilje lahko izvedemo kot sistem s trajnim ožičenjem ali kot sistem s prostim programiranjem. Pogosto imamo opravka s trajno-ožičenimi sistemi, kjer so osnovne enote posameznih korakov flip-flopi.

Kot prikazuje slika 91 so povezave med flip-flopi zasnovane po naslednjih principih:

- Na postavitveni vhod (SET) posameznega flip-flopa je priključen izhod predhodnega koraka. Izhod iz vezja določa pogoj za napredovanje v naslednji korak;
- Na ponastavitveni vhod (RESET) je priključen izhod flip-flopa iz naslednjega koraka (s tem briše izhod svojemu predhodniku). Ta vhod je namenjen brisanju izhodnega stanja [3].



Slika 91 - Funkcijski načrt v obliki koračne verige [Vir: lasten]

Ker se mora krmilje izvajati po korakih, je potrebno krmiljeni proces razstaviti na delne funkcije, t.i. korake. Delovanje korakov je tako bolj pregledno in nazorno. Med koraki so potrebni predhodni pogoji. Če so prehodi odvisni samo od stanj procesa, pravimo da je sistem procesno voden [1].

## 6.1 Primer koračnega krmilja

Na primeru koračnega krmilja bomo predstavili podajalno napravo prikazano na sliki 92. Podajalno napravo sestavljata dva dvosmerna pnevmatična cilindra, ki bosta prisotne pakete potiskala v škatlo. Paketi so zloženi v zalogovniku drug nad drugim. Prisotnost paketa na ploščadi zaznava optični senzor. Podajalno napravo upravljamo preko petih tipk:

- Start (S1);
- Stop (S2);
- Izklop v sili (S3);
- Reset (S4);
- Način delovanja (S5).



Slika 92 - Shema podajalne naprave [Vir: lasten]

#### **Delovanje naprave**

Pakete izpod zalogovnika na klančino podaja dvosmerno delujoči pnevmatični cilinder 1A. Podajanje paketov se izvrši s pomočjo dvosmernega pnevmatičnega cilindra 2A, kateri potiska pakete iz klančine v škatlo. Pnevmatična cilindra se gibljeta z dvema hitrostma: hitro in počasi. Počasen gib se izvaja med procesom podajanja, hitro gibanje pa izvaja med vračanjem batnice (nazaj) v začetni (uvlečeni) položaj. Končne položaje cilindrov zaznavamo z Reedovimi magnetnimi senzorji. Proces podajanja paketov se konča oz. prekine, ko pnevmatičnemu podajalnemu cilindru 1A zmanjkajo paketi ali pa ko optični senzor B5 na ploščadi ne zazna prisoten paket.

Programsko krmilje se izvede v naslednjih korakih:

- Cilinder 1A poda paket izpod zalogovnika;
- Cilinder 2A potisne paket iz klančine v škatlo, sočasno se cilinder 1A uvleče;
- Cilinder 2A se vrne v začetni položaj.

#### Zagon in upravljanje podajalne naprave

Pred prvim zagonom podajalne naprave je zaradi varnostnih razlogov potrebno pritisniti na tipki Izklop v sili (S3) in Reset (S4). S tem potrdimo ponastavljanje (brisanje) vseh izhodov in korakov (zastavic). Na takšen način postavimo aktuatorje (ventile) v začetne položaje. Program krmilja je možno zagnati šele, ko so vsi aktuatorji zagotovo v začetnih položajih in so izpolnjeni ostali pogoji za zagon podajalne naprave. Indikator Start (LED dioda, tipk ob tipki Start) sveti, ko je naprava pripravljena na zagon.

Zagon programa sprožimo s tipko Start (S1) ob pogoju, da sta oba cilindra v svojih začetnih položajih ter da imamo prisoten paket (senzor B5). Na voljo imamo 2 načina delovanja - avtomatski in ročni. Ročni način aktiviramo oz. izberemo s tipko S5 (če tipka ni aktivirana imamo izbrani avtomatski režim delovanja). Izvajanje programa lahko zaustavimo s tipko Stop (S2). Program izvede cikel do konca, šele nato se varno zaustavi. V primeru zasilnega (nujnega) izklopa pritisnemo tipko Izklop v sili (S3). Pri tem se aktuatorji prisilno zaustavijo ter ostanejo v trenutnih položajih. Nadaljnji zagon je možen šele takrat, ko so aktuatorji spet na svojih začetnih položajih. Aktivirani indikator Reset (LED dioda, tik ob tipki Reset) nas obvesti, da če želimo nadaljevati z avtomatiziranim podajanjem paketov, moramo pritisniti tipko Reset (S4) - tako postavimo aktuatorje na začetne položaje. Funkcija ponastavljanja ni izvedljiva tekom delovanja podajalne naprave.

Pri avtomatskem režimu (slika 93) se program izvaja nepretrgoma tako dolgo, dokler so prisotni vsi pogoji za normalno izvajanje programa. Izvajanje se lahko prekine tudi na željo uporabnika (tipka Stop). Med potekom programa lahko avtomatski način delovanja spremenimo v ročni način (ali obratno). Ko imamo pritisnjeno tipko Način delovanja (S5), imamo izbrani avtomatski način izvajanja programa. Če pa tipko sprostimo, preklopimo v

način ročnega delovanja. V ročnem režimu se cikel programa izvede le enkrat. Nadaljnjo izvajanje programa dosežemo s pritiskom na tipko Start (S1).



Slika 93 - Princip samodejnega načina delovanja [Vir: lasten]

Najvažnejši del koračnega krmilja je sekvenčni funkcijski diagram (koračna veriga), ki razčleni program v posamezne zaporedne korake, pri čemer ima vsak korak pripadajočo SR pomnilno celico - flip-flop. Tem korakom so prirejeni ukazi, ki se izvedejo v določenem zaporedju. Desno od koraka (flip-flopa) so pripadajoča ukazna polja, ki sprožijo izvedbo akcij (slika 94). Akcije krmilijo aktuatorje in zastavice v odvisnosti od sekvenčnega funkcijskega diagrama.

V priloženi prilogi C najdemo 3 sekvenčne funkcijske diagrame:

- Začetni vklop programa;
- Program za podajalno napravo;
- Tipka STOP v avtomatskem režimu.



Slika 94 - Ukazno polje [Vir: lasten]

V primeru podajalne naprave ostanejo ukazi aktivni skozi več korakov. Ukaz S ohranja stanje tudi po preteku koraka. V določenem koraku lahko ukaz vklopi ter v nekem drugem koraku izklopi SR pomnilno celico [1]. V lestvičnem programu podajalne naprave npr. 1. korak vklopi ter 3. korak izklopi ventil 1Y1. Najpogosteje uporabljene oznake za ukaze so podane v tabeli 20.

Tabela 20: Ukazni znaki [1]

Oznaka	Opis
Ν	Ne shranimo
S	Postavitev
R	Brisanje (ponastavljanje)
D	Časovno zakasnjen
L	Časovno omejen
Р	Impulz (fronta)
SD	Shranjen in časovno zakasnjen
DS	Zakasnjen in shranjen

#### Ostale podrobnosti delovanja

Pnevmatični cilinder 1A krmilimo z monostabilnim ventilom 1V1, cilinder 2A pa z bistabilnim ventilom 2V1. Na obeh cilindrih imamo nameščene Reedove magnetne senzorje. Na cilindru 1A imamo senzorja 1B1 in 1B2 v dvožični izvedbi. Na dvosmernem cilindru 2A pa se nahajata senzorja 2B1 in 2B2 v trožični izvedbi (PNP tip). Naloga posameznega senzorja je, da zaznava položaj batnice. Ob enem pa te podatke sproti pošilja (sporoča) krmilniku - tako CPE ve kaj se dogaja v procesu. Optični senzor B5 v trožični izvedbi (PNP tip) preverja prisotnost paketov. Ponoven zagon programa ni možen, če optični senzor ne zazna paketa na ploščadi.

### Izdelava lestvičnega programa v programski opremi ZEN

Iz opredelitvenih funkcionalnih zahtev podajalne naprave sledi, da za realizacijo naloge potrebujemo takšen tip Omron Zen krmilnika, ki ima 10 digitalnih vhodov in 6 izhodov. Uporabili bomo krmilnik Omron ZEN-20C1DR-D-V2 (brez dodatnih širitvenih V/I enot). Krmilnik ponuja 12 digitalnih vhodov od katerih sta lahko dva uporabljena kot analogna vhoda (Ia in Ib). Na izhodni strani ima krmilnik na voljo 8 relejskih izhodov. Vhodi se lahko priklopijo direktno na 12 - 24 V DC, relejski izhodi pa so lahko priključeni na 230 V AC/8 A ali na 24 V DC/5 A porabnike s skupno porabo do največ 40 A.

Glavne lastnosti te serije so:

- vhodi/izhodi: 12 vhodov (12 24 V DC) in 8 izhodov (relejski);
- napajanje krmilnika: 24 V DC, 5 W;
- visokotokovna izhodna zmogljivost;
- montaža na DIN letev;
- programska oprema ZEN;
- interno programiranje (na krmilniku);
- programski jezik: lestvični diagram (kontaktni načrt);
- programska kapaciteta: 96 vrstic (3 vhodi in 1 izhod na vrstico);
- čas izvedbe osnovne inštrukcije: 0.30 µs (na vrstico);
- komunicira preko RS-232;
- vgrajena ura realnega časa.

Krmiljenje podajalne naprave bo potekalo preko 10 vhodnih in šestih izhodnih signalov.

Potrebe po vhodnih signalih so naslednje:

- 1. Reedov magnetni dvožični senzor 1B1 (pomik obdelovanca na delovno mesto);
- 2. Reedov magnetni dvožični senzor 1B2 (obdelovanec na delovnem mestu);
- 3. Reedov magnetni trožični senzor 2B1 PNP (položaj batnice zgoraj);
- 4. Reedov magnetni trožični senzor 2B2 PNP (položaj batnice spodaj);
- 5. Optični trožični senzor B5 PNP (preverjanje prisotnosti paketa);
- 6. Tipka Start NO (delovni kontakt);
- 7. Tipka Stop NC (mirovni kontakt);
- 8. Tipka Izklop v sili NC (mirovni kontakt);

- 9. Tipka Reset NO (delovni kontakt);
- 10. Tipka Način delovanja NO (delovni kontakt).

Krmiljeni elementi (izhodni signali) so naslednji:

- El. signal 1Y1 podajanje paketa izpod zalogovnika (krmiljenje 5/2 elektromagnetnega monostabilnega ventila 1V1 z električnim signalom 1Y1 - 24 V DC);
- El. signal 2Y1 potisk paketa iz klančine v škatlo (krmiljenje 5/2 elektromagnetnega bistabilnega ventila 2V1 z električnim signalom 2Y1 - 24 V DC);
- 3. El. signal 2Y2 postavitev cilindra 2A v začetni položaj (krmiljenje 5/2 elektromagnetnega bistabilnega ventila 2V1 z električnim signalom 2Y2 24 V DC);
- El. signal 3Y1 zaprtje glavnega dovoda zraka (krmiljenje 3/2 elektromagnetnega monostabilnega ventila 3V1 z električnim signalom 3Y1 - 24 V DC);
- 5. Indikator Start LED dioda (opozorilo, da je naprava pripravljena na zagon);
- 6. Indikator Reset LED dioda (opozorilo o nujnem ponastavljanju).

Električni načrt priključitve krmilnega dela (vhodni in izhodni električni signali) na PLK Omron Zen je prikazan v priloženi prilogi B.



Slika 95 - SMC - Elektromagnetni ventili [Vir: lasten]

Dvosmerna cilindra 1A in 2A sta krmiljena preko 5/2 elektromagnetnih ventilov (slika 95). Kot prikazuje priložena priloga A dovajamo zrak na obe strani valja. Hitrost batnice uravnamo z dušenjem iztekajočega zraka. Z uporabo hitroodzračevalnega ventila iztekajočemu zraku skrajšamo pot do atmosfere. Hitrosti gibov pnevmatičnih cilindrov prilagodimo oz. nastavimo po želji. V primeru podajalne naprave se počasen gib izvaja med procesom podajanja, hitro gibanje pa se izvaja med vračanjem batnice (nazaj) v začetni (uvlečeni) položaj.

Slika 96 prikazuje dvosmerni pnevmatični cilinder SMC v začetnem položaju.



Slika 96 - SMC - Dvosmerni pnevmatični cilinder (valj) [Vir: lasten]

V primeru nujnega izklopa se z aktiviranjem tipke - Izklop v sili zapre glavni dovod zraka preko 3/2 elektromagnetnega monostabilnega ventila 3V1.

Lestvični diagram podajalne naprave je prikazan v priloženi prilogi D. Lestvični diagram je bil narejen oz. sprogramiran v programski opremi Zen. Posamezne vrstice zaradi lažjega pregleda nad programom vsebujejo komentarje. Komentar na kratko opiše vrstico, v kateri se nahajamo (tako imamo bolj jasen pregled nad tem, kje v programu se nahajamo).

V priloženi prilogi L najdemo slike podajalne naprave med preizkušanjem.

## 6.2 Primer kombinacijskega krmilja

Prav v času, ko domove zapuščamo za daljši čas, pogosteje pomislimo na varnost naših domov. Zato lahko z nekaj ukrepi poskrbimo za varnost. Učinkovita zaščita doma bi lahko bila kombinacija mehanske in elektronske zaščite - alarmna naprava, vodena preko krmilnika Omron Zen. Na primeru kombinacijskega krmilja bomo predstavili alarmno napravo (slika 97).

#### Delovanje alarmne naprave

Krmiljenje alarmne naprave poteka preko treh senzorjev. V stanovanjski hiši imamo na treh mestih postavljene senzorje gibanja. Senzorji se nahajajo v spalnici (B1), v dnevni sobi (B2) in na hodniku (B3). V primeru aktiviranja enega izmed senzorjev se aktivira alarm (Q1).



Slika 97 - Alarmna naprava (primer postavitve) [16]

#### Zagon alarmne naprave

Alarmno napravo aktiviramo s tipko Start (S1). Da imamo alarm aktiviran, nas opozori Indikator Start (LED dioda, tik ob tipki Start), ki v času delovanja alarmne naprave utripa. Alarmni sistem je sprogramiran tako, da se med tednom od 22. do 22.15 ure aktivira zvočna signalizacija (Q0), ki opozarja, da se bo aktiviral alarm. Alarm se samodejno aktivira ob 22.15. V času od 22.15 in 5.15 je prepovedan vstop v stanovanjsko hišo. V primeru, da senzor zazna gibanje (vsiljivca) v času delovanja alarmne naprave se aktivira alarm (Q1). Alarmno napravo ponastavimo (izklopimo) s tipko Reset (S3). Če pa se alarm čez noč ni sprožil, lahko delovaje alarmne naprave zaustavimo oz. prekinemo s tipko Stop (S2).

Lestvični diagram alarmne naprave v programski opremi ZEN je prikazan v priloženi prilogi E. V programu sta uporabljena tedenski časovnik (Weekly Timer) in utripajoči impulzni časovnik (Flashing Pulse). Slednji je uporabljen za utripanje indikatorja Reset v času aktiviranega alarma. S tedenskim časovnikom določimo uro, kdaj se naj posamezen izhod aktivira (zvočna signalizacija, alarm) in kdaj se naj deaktivira. Slika 98 prikazuje, da imamo za ponedeljek - torek nastavljeno začetno uro 22:15 in končno uro 5:15 (med tem časom je aktivirana alarmna naprava).



Slika 98 - Uporaba tedenskega časovnika pri alarmni napravi [Vir: lasten]

## 7. ZAKLJUČEK

Namen diplomske naloge je bila predstavitev krmilnika Omron Zen in izdelava navodil za uporabo v slovenščini. Preprosta in uporabniško prijazna serija Zen iz družine krmilnikov Omron je namenjena za rabo v tovarni, pisarni, domu itd., kjerkoli se čuti potreba po enostavnih funkcijah vodenja. Predstavljeni PLK se po svojih lastnostih lahko primerja z drugimi kompaktnimi krmilniki spodnjega zmogljivostnega razreda, vendar največjo težavo predstavlja nezmožnost priključitve na industrijsko-podatkovno omrežje.

Pri realizaciji diplomske naloge sem moral spoznati programsko opremo ZEN za programiranje krmilnika. Programska oprema Zen podpira grafični jezik lestvični diagram, kar naredi programiranje bolj pregledno, razumljivo. S pomočjo programske opreme sem napisal lestvični program za koračno krmilje in lestvični program za kombinacijsko (logično) krmilje. V primeru koračnega krmilja sem predstavil podajalno napravo, v primeru kombinacijskega krmilja pa sem podal realen primer, kako bi lahko krmilnik uporabili za domače namene (alarmna naprava).

Med razvojem podajalne naprave sem uporabil vse možne vhode. Težava bi lahko nastala, če bi potreboval še več vhodov. Ena izmed rešitev bi lahko bila, da bi uporabil operacijske tipke na sprednji strani krmilnika. Bolj primerna rešitev in s tem možnost izboljšanja programa (podajalne naprave) pa bi bila, če bi krmilniku dodal širitvene V/I enote ter tako pridobil dodatne vhode/izhode. Vendar sem se odločil za implementacijo zgolj tistih zahtev, katere so bile za izvedbo naloge potrebne. Pred programiranjem sem potreboval predstavo o zahtevah delovanja procesa (kako naj podajalna naprava deluje).

Delovanje uporabniških programov sem preizkusil in sem bil zadovoljen.

Lotil sem se področja, ki sem ga nekoliko poznal že od prej. Kljub nekaterim začetnim zapletom pri realizaciji podajalne naprave je naloga uspela in naučil sem se veliko novega.

Področje avtomatizacije mi je z realizacijo koračnega in kombinacijskega krmilja postalo še bolj poznano.

Krmilnik je po zmogljivosti in možnostih, ki jih ponuja primeren za številne enostavne naloge krmiljenja. Primeren je za učenje - s krmilnikom lahko naredimo prve začetne korake v avtomatizaciji. Zaradi svoje enostavnosti in preglednosti je uporaben za izobraževanje učencev, dijakov, študentov, ki šele vstopajo v svet elektronike in logičnih vezij.

## 8. IZJAVA AVTORJA

S podpisom potrjujem, da sem diplomsko nalogo izdelal samostojno in po virih, ki so navedeni v diplomski nalogi ter v skladu z Navodili za izdelavo diplomske naloge, ki jih je izdala Višja strokovna šola.

Diplomsko nalogo sem izdelal po izhodiščih, ciljih in dispoziciji potrjene Prijave teme in naslova diplomske naloge.

Ptuj, 14.10.2011

Podpis: \_\_\_\_\_

# 9. LITERATURA IN VIRI

[1]	Mehatronika, 2. izd., Pasadena, Ljubljana, 2009.
[2]	V. Ambrožič, D. Nedeljković, Uvod v programirljive krmilne sisteme, FE in
	FRI, Ljubljana, 2005.
[3]	C. Štandeker, Krmilja in regulacije. 1. del: Krmilja, TzS, Ljubljana, 1993.
[4]	Omron, Data Sheet, 2011. (online). (citirano 21.5.2011).
	Dostopno na naslovu:
	www.ia.omron.com/data_pdf/data_sheet/zen_ds_csm106.pdf
[5]	Omron, ZEN Support Software Operation Manual, 2008. (online). (citirano
	21.5.2011). Dostopno na naslovu:
	http://downloads.industrial.omron.eu/IAB/Products/Control%20Component
	s/Programmable%20Relays/Z184/Z184-E1-04.pdf
[6]	L. A. Bryan, E. A. Bryan, Programmable Controllers. Theory and
	Implementation, 2. izd., ZDA, 1997. (online). (citirano 11.8.2011).
	Dostopno na naslovu:
	http://www.bioingenieria.edu.ar/academica/catedras/control/material%20est
	udio/Anexo/ebook%20-
	PLC%20Programmable%20Logic%20Controllers%20theory%20and%20im
	plementation%202nd%20ed.pdf
[7]	M. Medved, Projektni razvoj programske opreme za programabilne logične
	krmilnike, Fakulteta za elektrotehniko, računalništvo in informatiko,
	Diplomsko delo, Maribor, 2010.
[8]	T. Pintarič, M. Hočevar, J. Čurk, A. Gorenc, Avtomatizacija in robotika,
	Višja strokovna šola Novo mesto, Novo mesto, 2011.
[9]	Siemens, S7-300 CPUs, 2010. (online). (citirano 22.8.2011).
	Dostopno na naslovu:
	http://www.automation.siemens.com/bilddb/picDetails.aspx?objID=101442
[10]	Siemens, Basics of PLCs, 2010. (online). (citirano 24.8.2011).

	Dostopno na naslovu:
	http://www.robicon.com/step/downloads.html
[11]	D. Bobnar, Prototipni model mikroprocesorskega krmiljenja stroja za
	peskanje, Fakulteta za organizacijske vede, Diplomsko delo, Kranj, 2007.
[12]	R. Cajhen, Sodobni elektronski krmilni sistemi, Fakulteta za elektrotehniko,
	Skripta, Ljubljana, 1985.
[13]	I. Šalković, Avtomatizacija brizgalno-hladilne linije za izdelavo polizdelkov
	gumenih plaščev, Fakulteta za elektrotehniko, Diplomsko delo, Ljubljana,
	2011.
[14]	H. Berger, Automating with STEP 7 in STL and SCL, 4. izd., Publicis
	Corporate Publishing, Berlin in München, 2007.
[15]	Omron, ZEN Operation Manual, 2008. (online). (citirano 2.10.2011).
	Dostopno na naslovu:
	http://downloads.industrial.omron.eu/IAB/Products/Control%20Component
	s/Programmable%20Relays/Z211/Z211-E1-03.pdf
[16]	Kigohuram, Floor Plans For Homes, 8.9.2011. (online). (citirano
	13.10.2011). Dostopno na naslovu:
	http://www.californiahomessales.net/floor-plans-for-homes/

# **10. PRILOGE**

## 10.1 Seznam slik

Slika 1 - Princip ZOI	
Slika 2 - Primer prenosa uporabniškega programa v pomnilnik PLK-ja	4
Slika 3 - Vhodne in izhodne enote PLK-ja	6
Slika 4 - Analogni signal	7
Slika 5 - Vhodna enota PLK	8
Slika 6 - Vodila PLK-ja	9
Slika 7 - Kompaktni PLK - Omron Zen	10
Slika 8 - Modularni PLK - Siemens S7-300	11
Slika 9 - Simbol in pravilnostna tabela RS pomnilne celice s prioriteto R	17
Slika 10 - Časovni potek signalov pri RS pomnilni celici	17
Slika 11 - Časovne funkcije	18
Slika 12 - Simbol in časovni potek časovnega člena za podaljšanje/skrajšanje signalov	v 19
Slika 13 - Simbol in časovni potek časovnega člena za premikanje signalov	19
Slika 14 - Časovnik kot vklopna zakasnitev	20
Slika 15 - Časovni potek zakasnitve vklopa	20
Slika 16 - Časovnik kot izklopna zakasnitev	21
Slika 17 - Časovni potek zakasnitve izklopa	21
Slika 18 - Samodržna vezava	23
Slika 19 - Programski grafični jezik - Lestvični diagram	25
Slika 20 - Primer lestvičnega diagrama	26
Slika 21 - Binarni operand v kombinaciji z zapiralnim in odpiralnim kontaktom	27
Slika 22 - Delovanje zapiralnega (delovnega) kontakta	27
Slika 23 - Delovanje odpiralnega (mirovnega) kontakta	28
Slika 24 - Primer funkcijskega blokovnega diagrama (z logičnimi bloki)	28
Slika 25 - Primer tekstualnega jezika STL	29
Slika 26 - Kompaktni krmilnik Omron Zen z 20 V/I	31
---	----
Slika 27 - Različni pogledi na krmilnik Omron Zen	33
Slika 28 - Dodatna oprema krmilnika	33
Slika 29 - Zaslon krmilnika	34
Slika 30 - Programska oprema ZEN - Verzija 4.14	37
Slika 31 - Ustvarjanje novega programa	37
Slika 32 - Konfiguracijske nastavitve	38
Slika 33 - Meniji programske opreme	38
Slika 34 - Menijske vrstice	41
Slika 35 - Parametri vhoda	42
Slika 36 - Dodajanje vhodnih spremenljivk	42
Slika 37 - Funkcije izhoda	43
Slika 38 - Eno vrstico sestavljajo trije vhodi in en izhod	43
Slika 39 - Dodajanje vrstice	44
Slika 40 - Uporabniški program lahko nadgradimo	45
Slika 41 - Preverjanje programa	45
Slika 42 - Primer simulacije v programski opremi ZEN	46
Slika 43 - Vhodu lahko nastavimo več tipov kontakta (NO, NC, Alternativni)	47
Slika 44 - Preverjanje povezave	48
Slika 45 - Povezovanje	48
Slika 46 - Prenos uporabniškega programa in konfiguracijskih nastavitev v/na krmilnik	49
Slika 47 - Prenos uporabniškega programa in konfiguracijskih nastavitev iz krmilnika	49
Slika 48 - Primerjava uporabniških programov	50
Slika 49 - Zaključek preverjanja uporabniških programov	50
Slika 50 - Funkcija spremljanja delovanja programa	50
Slika 51 - Tiskanje uporabniškega programa	51
Slika 52 - Operacijske tipke na sprednji strani krmilnika	52
Slika 53 - Ustvarjanje programa iz preprostega vezja	54
Slika 54 - Prikaz notranjega delovanja v krmilniku	54
Slika 55 - Brisanje programa	55
Slika 56 - Ustvarjanje novega programa	55
Slika 57 - Pomen oznak v lestvičnem programu pri internemu programiranju	56

Slika 58 - Ustvarjanje prvega vhoda I0	56
Slika 59 - Ustvarjanje drugega vhoda I1	57
Slika 60 - Dodajanje horizontalne povezave	57
Slika 61 - Dodajanje izhoda Q0	58
Slika 62 - Izbira povezovalne linije	58
Slika 63 - Uporaba izhoda kot vhodno spremenljivko	59
Slika 64 - Paralelna vezava spremenljivke	59
Slika 65 - Prikaz glavnega zaslona	60
Slika 66 - Delovanje programa lahko spremljamo	60
Slika 67 - Vrste časovnih funkcij	62
Slika 68 - Nastavitve časovnika v lestvičnemu programu	63
Slika 69 - Način spremljanja na krmilniku	64
Slika 70 - Normalna operacija tedenskega časovnika med spremljanjem delovanja	64
Slika 71 - Spremljanje delovanja koledarskega časovnika	65
Slika 72 - Določitev funkcije števca	65
Slika 73 - Časovni potek signalov pri števcu	66
Slika 74 - Nastavitve štetja v lestvičnemu programu	66
Slika 75 - Način spremljanja delovanja števca	67
Slika 76 - Funkcija prikaza sporočil - bitov (spremenljivk)	68
Slika 77 - Primer uporabe funkcije prikaza sporočil	68
Slika 78 - Uporaba analognih signalov (Ia, Ib)	69
Slika 79 - Prikaz uporabe funkcije analognega primerjalnika	70
Slika 80 - Primerjanje držalnega časovnika s konstanto	70
Slika 81 - Uporaba 8-bitnega števnega primerjalnika	71
Slika 82 - Operacijske tipke na sprednji strani krmilnika kot vhodni biti	72
Slika 83 - Dvožična vezava senzorja	73
Slika 84 - Trožična vezava senzorja	74
Slika 85 - Ožičenje krmilnika in širitvene V/I enote z izmenično AC napetostjo	74
Slika 86 - Ožičenje krmilnika z enosmerno DC napetostjo (PNP povezava)	75
Slika 87 - Ožičenje krmilnika z enosmerno DC napetostjo (NPN povezava)	76
Slika 88 - Ožičenje analognih vhodov z enosmerno DC napetostjo	77
Slika 89 - Enote s tranzistorskim izhodom	78

Slika 90 - Enote z relejskim izhodom	79
Slika 91 - Funkcijski načrt v obliki koračne verige	82
Slika 92 - Shema podajalne naprave	83
Slika 93 - Princip samodejnega načina delovanja	85
Slika 94 - Ukazno polje	86
Slika 95 - SMC - Elektromagnetni ventili	88
Slika 96 - SMC - Dvosmerni pnevmatični cilinder (valj)	89
Slika 97 - Alarmna naprava (primer postavitve)	90
Slika 98 - Uporaba tedenskega časovnika pri alarmni napravi	91

# 10.2 Seznam preglednic

Tabela 1: Enote standarda EN 61131	2
Tabela 2: Glede na napetosti, ki pridejo iz dajalnikov so definirani logični nivoji	7
Tabela 3: Logična funkcija NE	15
Tabela 4: Logična funkcija IN	15
Tabela 5: Logična funkcija ALI	16
Tabela 6: Pomen ikon na zaslonu krmilnika	34
Tabela 7: Zahteve programske opreme	36
Tabela 8: Oznake in funkcije operacijskih tipk	53
Tabela 9: Operacije tedenskega časovnika	64
Tabela 10: Hitrosti štetja pri 8-bitnemu digitalnemu števcu	67
Tabela 11: Izbira prikazanega objekta	68
Tabela 12: Nastavitev parametrov za analogni primerjalnik	69
Tabela 13: Časovni/števni primerjalnik	71
Tabela 14: Nastavitev konstante in operatera	72
Tabela 15: Specifikacije za izmenične AC vhode (vhodi I0 - I9)	75
Tabela 16: Specifikacije za enosmerne DC vhode (vhodi I0 - I9)	76
Tabela 17: Specifikacije za analogna vhoda Ia in Ib	77
Tabela 18: Specifikacije za tranzistorski izhod	78
Tabela 19: Specifikacije za relejski izhod	80
Tabela 20: Ukazni znaki	86

# 10.3 Ostale priloge

Priloga A: Pnevmatični načrt - Podajalna naprava	104
Priloga B: Električni načrt priključitve na PLK - Podajalna naprava	105
Priloga C: Sekvenčni funkcijski diagram (SFD) - Podajalna naprava	107
Priloga D: Lestvični diagram (LD) - Podajalna naprava	110
Priloga E: Lestvični diagram (LD) - Alarmna naprava	114
Priloga F: Postopek programiranja s krmilnikom Omron Zen	115
Priloga G: Menijski zaslon konfiguracijskih nastavitev	116
Priloga H: Tehnični podatki	117
Priloga I: Specifikacije	118
Priloga J: Funkcije, spremenljivke	120
Priloga K: Dimenzije krmilnika, napajalne in širitvene V/I enote	125
Priloga L: Podajalna naprava (slike)	126



## Priloga A: Pnevmatični načrt - Podajalna naprava



## Priloga B: Električni načrt priključitve na PLK - Podajalna naprava





### Priloga C: Sekvenčni funkcijski diagram (SFD) - Podajalna naprava

Začetni vklop - izklop v sili in funkcija RESET [Vir: lasten]



Program: Podajalna naprava (glavni del) [Vir: lasten]



Tipka STOP v avtomatskem režimu (osnovni del) [Vir: lasten]



### Priloga D: Lestvični diagram (LD) - Podajalna naprava







Lestvični diagram podajalne naprave [Vir: lasten]



### Priloga E: Lestvični diagram (LD) - Alarmna naprava

Lestvični diagram alarmne naprave [Vir: lasten]

### Priloga F: Postopek programiranja s krmilnikom Omron Zen



Priprave na delovanje avtomatiziranega procesa

(prevedeno po [15])

## Priloga G: Menijski zaslon konfiguracijskih nastavitev



Menijski zaslon krmilnika Omron Zen

(prevedeno po [15])

Unit	Name	No. of I/O points	LCD display	Power supply voltage		Inputs		Outputs	Buttons, calendar, and clock	Analog input	Model										
CPU	Standard	10	Yes	100 to 240 VAC	6	100 to 240 VAC	4	Relays	Yes	No	ZEN-10C1AR-A-V2										
Units	LCD type			12 to 24 VDC	1	12 to 24 VDC				Yes	ZEN-10C1DR-D-V2										
								Transistors			ZEN-10C1DT-D-V2										
		20		100 to 240 VAC	12	100 to 240 VAC	8	Relays		No	ZEN-20C1AR-A-V2										
				12 to 24 VDC	1	12 to 24 VDC	1			Yes	ZEN-20C1DR-D-V2										
							Transistors			ZEN-20C1DT-D-V2											
	LED type	10	No	100 to 240 VAC	6	100 to 240 VAC	4	Relays	No	No	ZEN-10C2AR-A-V2										
	display			12 to 24 VDC	1	12 to 24 VDC	1					Yes	ZEN-10C2DR-D-V2								
	(See note 1.)							Transistors			ZEN-10C2DT-D-V2										
		20	1	100 to 240 VAC	12	100 to 240V AC	8	Relays		No	ZEN-20C2AR-A-V2										
				12 to 24 VDC	1	12 to 24 VDC	1			Yes	ZEN-20C2DR-D-V2										
									Transistors			ZEN-20C2DT-D-V2									
	Economy type (Expansion I/O Units	10	10 Yes	100 to 240 VAC	6 100 to 240 VA 12 to 24 VDC	100 to 240 VAC	4	Relays	Yes	No	ZEN-10C3AR-A-V2										
				12 to 24 VDC		12 to 24 VDC				Yes	ZEN-10C3DR-D-V2										
		20	1	100 to 240 VAC	12	100 to 240 VAC	8	Relays		No	ZEN-20C3AR-A-V2										
	connected)			12 to 24 VDC	1	12 to 24 VDC				Yes	ZEN-20C3DR-D-V2										
	Communica-	9	1	100 to 240 VAC	6	100 to 240 VAC	3	3	3	3	3	3	3	3	3	3	3	Relays	_	No	ZEN-10C4AR-A-V2
	tions type			12 to 24 VDC	-	12 to 24 VDC	12 to 24 VDC				Yes	ZEN-10C4DR-D-V2									
ZEN Kit		Set con	taining CP	U Unit (ZEN-10C1A	R-A-	V2), Connecting Ca	ble, Z	EN Support Softw	vare, and manual		ZEN-KIT01-EV4										
		Set cont	taining CP	U Unit (ZEN-10C10	DR-D-	V2), Connecting Ca	ble, Z	EN Support Softw	vare, and manual		ZEN-KIT02-EV4										
Expansio	on I/O Units 8		100 to 240 VAC	4	100 to 240 VAC	4	Relays			ZEN-8E1AR (See notes 2, 3.)											
				12 to 24 VDC		12 to 24 VDC	1	1	7	7	1	1	1				ZEN-8E1DR (See note 2.)				
								Transistors			ZEN-8E1DT										

### Priloga H: Tehnični podatki

Note: 1. Programming is not possible using only the CPU in the LED-type CPU Unit. ZEN Support Software or a Memory Cassette is required. 2. Cannot be connected to pre-V1 and V1 CPU Units.

3. The ZEN-8E1AR cannot be connected to a CPU Unit with DC power supply.

Power ratings	Input voltage	Output voltage	Output current	Model
30 W	100 to 240 VAC	24 VDC	1.3 A	ZEN-PA03024

#### CPE, širitvene V/I enote in napajalna enota [4]

Name	Specifications		Remarks					
Memory Cassette	EEPROM (for data security and copying)	Enables programs and p ZEN. (See note 1.)	ZEN-ME01					
	2		LCD-type CPU Unit with display (See note 2.)	LED-type CPU Unit without display (See note 3.)				
		Transfer from ZEN to Memory	Supported	Not supported				
		Transfer from Memory Cassette to ZEN	Supported	Automatic transfer when power turned ON				
		Memory Cassette initialization	Supported	Not supported				
Connecting Cable	2 m RS-232C (9-pin D- sub connector)				ZEN-CIF01			
Battery Unit	10 years min. Battery life (at 25°C)	Ladder programs and pa but calendar, clock, and values are held by the c for 2 days or more (at 25 systems where the pow	ZEN-BAT01					
ZEN Support Software	Runs on Windows 95, 98, 2000, ME, XP, NT 4.0, Vista or 7. (See note 4.)	Specifically designed fo	ZEN-SOFT01-V4					

Memory Cassettes created using a CPU Unit can be read to other CPU Units, regardless of which model is used. Restrictions, apply, however, to the functions that can be used. depending on the CPU Unit version combination. For details, refer to *Memory Cassette and CPU Unit Combinations* on page 31.
 Standard LCD-type, Economy-type, and Communications-type CPU Units (i.e., excluding ZEN-020-0-V2 models).
 LED-type CPU Unit without display (i.e., ZEN-020-0-V2 models).
 Except for 64-bit versions of Windows.

Dodatna oprema [4]

## Priloga I: Specifikacije

Item	Specification					
	ZEN-CAR-A-V2/ZEN-8E1AR	ZEN-CDD-D-V2/ZEN-8E1D				
Rated supply voltage	100 to 240 VAC, 50/60 Hz	12 to 24 VDC (DC ripple rate: 5% max.)				
Operating voltage range	85 to 264 VAC	10.8 to 28.8 VDC				
Power consumption	CPU Units without Expansion I/O Units           • ZEN-10C1AR-A-V2/ZEN-10C2AR-A-V2/ ZEN-10C3AR-A-V2           100 V AC: 5 VA max.           240 V AC: 7 VA max.           • ZEN-10C4AR-A-V2           100 V AC: 6 VA max.           • ZEN-10C4AR-A-V2           100 V AC: 6 VA max.           • ZEN-10C4AR-A-V2           100 V AC: 6 VA max.           240 V AC: 8 VA max.           • ZEN-20C□AR-A-V2           100 V AC: 7 VA max.           240 V AC: 10 VA max.           240 V AC: 10 VA max.           240 V AC: 6 VA max.           240 V AC: 8 VA max.           240 V AC: 9 VA max.           240 V AC: 11 VA max.           240 V AC: 11 VA max.           240 V AC: 3 VA max.           240 V AC: 3 VA max.	CPU Units without Expansion I/O Units • ZEN-10C□DR-D-V2 12/24 V DC: 3 W max. (ZEN-10C3DR-D-V2: 2.8 W max.) • ZEN-10C□DT-D-V2 12/24 V DC: 2 W max. • ZEN-20C□DR-D-V2 12/24 V DC: 4 W max. • ZEN-20C□DT-D-V2 12/24 V DC: 2 W max. CPU Units with three Expansion I/O Units • ZEN-10C□DR-D-V2 12/24 V DC: 4 W max. • ZEN-10C□DT-D-V2 12/24 V DC: 3 W max. • ZEN-20C□DR-D-V2 12/24 V DC: 3 W max. • ZEN-20C□DT-D-V2 12/24 V DC: 3 W max. • ZEN-20C□DT-D-V2 12/24 V DC: 3 W max. • ZEN-20C□DT-D-V2 12/24 V DC: 3 W max. • ZEN-8E1DR 12/24 V DC: 2 W max.				
Inrush current	ZEN-10C□AR-A-V2:         4.5 A max.         ZEN-10C□D□-D-V2:         30 A max.           ZEN-20C□AR-A-V2:         4.5 A max.         ZEN-20C□D□-D-V2:         30 A max.           ZEN-8E1AR:         4 A max.         ZEN-8E1DR:         15 A max.					
Ambient temperature	0 to 55°C (-25 to 55°C for ZEN-C2C	odels)				
Ambient storage temperature	-20 to 75°C (-40 to 75°C for ZEN-C2C	models)				
Ambient humidity	10% to 90% (with no condensation)					
Ambient conditions	No corrosive gases					
Mounting method	Surface mounting, DIN track mounting (standard (vertical) installation and horizontal installation) (See notes 1 and 2.)					
Terminal block	Solid-line terminal block (use solid wire or fine-stranded wire)					
Terminal screw tightening torque	0.565 to 0.6 N·m (5 to 5.3 in-lb)					
Degree of protection	P20 (Mounted inside a control panel)					

Note: 1. Can be mounted to 35-mm DIN Track.

2. Standard (Vertical) installation Horizontal installation



Poraba električne energije, vklopni tok in pogoji delovanja [4]

Item	Specification
Control method	Stored program control
I/O control method	Cyclic scan
Programming language	Ladder diagram
Program capacity	96 lines (3 input conditions and 1 output per line)
Max. No. of control I/O points	44 points (See note 1.) CPU Units with 20 I/O points: 12 inputs and 8 outputs Expansion I/O Units: 4 inputs and 4 outputs each, up to 3 Units.
LCD display (See note 2.)	12 characters × 4 lines, with backlight
Operation buttons (See note 2.)	8 (4 cursor buttons and 4 operation buttons)
User program backup	Internal EEPROM, Memory Cassette (optional)
Power interruption hold	Internal holding bit status, holding timer/counter present values, calendar and clock (year, month, day of month, day of week, time) • Super capacitor backup time:2 days min. (25°C) • Life of optional battery: 10 years min. (25°C)
Calendar and clock function (See note 2.)	Accuracy: ±15 s/month (at 25°C)
Timer accuracy	0.01 s unit: -0.05% -10 ms max. (rate for set value) min/s unit: -0.05% -1 s max. (rate for set value) h/min unit: -0.05% -1 min max. (rate for set value)
Maximum counting speed	150 Hz: 8-Digit counter (F) set to high-speed operations (CPU Units with DC power supplies only) (The counting speed may be less than 150 Hz depending on the cycle time of the program. See page 21.)
Insulation resistance	$20 \text{ M}\Omega$ (at 500 VDC) min.: Between power supply terminals and all output terminals. Between terminals of different output circuits. Between all terminals of CPU Unit and all terminals of Expansion I/O Unit.
Insulation	<ul> <li>Reinforced insulation Between power supply or input terminals and output terminals. Between terminals of different output circuits. Between all terminals of CPU Unit and all terminals of Expansion I/O Unit.</li> <li>No separation Between power supply and input terminals of the same unit. Between power supply terminals of CPU Unit and computer connector, Battery Unit connector, or all Expansion Unit connectors (all interfaces are live parts).</li> </ul>
Dielectric strength	2,300 VAC, 50/60 Hz for 1 min (leakage current 1 mA max.): Between power supply terminals and all output terminals. Between terminals of different output circuit. Between all terminals of CPU Unit and all terminals of Expansion I/O Unit.
Vibration resistance	Conforms to IEC60068-2-6, 5 to 9 Hz with 3.5-mm single amplitude, 9 to 150 Hz acceleration 9.8 m/s <sup>2</sup> , 10 sweeps each in X, Y, and Z directions (1 octave/min)
Shock resistance	Conforms to IEC60068-2-27, 147 m/s <sup>2</sup> , 3 times each in X, Y, and Z directions.
Weight	CPU Unit with 10 I/O points: Approx. 300 g max. CPU Unit with 20 I/O points: Approx. 350 g max. Expansion I/O Unit: Approx. 120 g max.

Note: 1. Up to 34 points for CPU Units with 10 I/O points. With Communications-type CPU Units, however, the CPU Unit has 6 inputs and 3 outputs, for a maximum of 33 I/O points.
 Not provided for LED-type CPU Unit without display (i.e., ZEN-□C2□-□-V2 models).

### Tehnične karakteristike [4]

Item	Specifications	Circuit drawing
Input voltage	100 to 240 VAC +10%, -15%, 50/60 Hz	2 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100
Input impedance	680 kΩ	
Input current	0.15 mA/100 VAC, 0.35 mA/240 VAC	
ON voltage	80 VAC min.	
OFF voltage	25 VAC max.	100 to 240 VAC
ON response time	50 ms or 70 ms at 100 VAC (See note.)	Q NO
Off response time	100 ms or 120 ms at 240 VAC (See note.)	The last a second

Note: Can be selected using the input filter settings.

Item	Specifications	Circuit drawing
Input voltage	12 to 24 VDC +20%, -10%	·····
Input impedance	6.5 kΩ	
Input current	3.7 mA (typ.)/24 VDC	<sup>5</sup> IN <sub>1</sub> 6.2 kΩ 1.8 kΩ
ON voltage	8 VDC min.	
OFF voltage	5 VDC max.	
ON response time	15 ms or 50 ms (See note 1.)	
OFF response time		L

Note: 1. Can be selected using the input filter settings.

2. The ZEN-8E1DT has no +/- terminals. There is no need to supply power.

Specifikacije širitvenih V/I enot za izmenične AC in enosmerne DC vhode [4]

## Priloga J: Funkcije, spremenljivke

Name	Symbol	Bit addresses	No. of points		Operation	1	Details (See note 1.)
Input bits	1	I0 to Ib (See note 2.)	12	Reflect the ON/OFF status of the input devices connected to the input erminals on the CPU Unit.			
Expansion input bits	x	X0 to Xb	12	Reflect the ON/OFF terminals on the Exp	eflect the ON/OFF status of the input devices connected to the input rminals on the Expansion I/O Units.		
Output bits	Q	Q0 to Q7 (See note 2.)	8	The ON/OFF status of connected to the out	he ON/OFF status of these output bits is used to control the output devices onnected to the output terminals on the CPU Unit.		1
Expansion output bits	Y	Y0 to Yb	12	The ON/OFF status of connected to the out	he ON/OFF status of these output bits is used to control the output devices onnected to the output terminals on the Expansion I/O Units.		
Work bits	М	M0 to Mf	16	Work bits can be use cannot be made (i.e.	e used only within the ZEN program. I/Os for external devices a (i.e., all I/O is internal).		
Holding bits	н	H0 to Hf	16	Used the same as th OFF, these bits also	e work bits. However, if the power to the ZEN is turned maintain the previous ON/OFF status.		
Timers	Т	T0 to Tf	16	X: ON-delay timer	Functions are	Time units can be selected	
				: OFF-delay timer	selected from the screen when parameter settings are made.	from the following: 0.01-s unit: 0.01 to 99.99 s min/s unit: 00 min 01 s to 99 min 59 s h/s unit: 00 h 01 min to 99 h 59 min	2
				O: One-shot pulse timer			
				F: Flashing pulse timer			
				W: Twin timer	1		
Holding timers	#	#0 to #7	8	Hold the present value being counted even if the trigger input or power supply is turned OFF and continue timing when the trigger input or power is restored.		3	
Counters	С	C0 to Cf	16	Reversible 4-digit counters that can be incremented and decremented.		4	
8-digit counter	F	F0	1	Reversible 8-digit counters that can be incremented and decremented. CPU Units with DC power supply support a high-speed counter up to 150 Hz. For details, refer to 8-digit Counter Operation on page 21.		5	
Weekly timers (See note 3.)	@	@0 to @f	16	Turn ON and OFF during specified times on specified days.		6	
Calendar timers (See note 3.)	*	*0 to *f	16	Turn ON and OFF between specified dates.		7	
Display bits (See note 3.)	D	D0 to Df	16	Display user-specifie analog-converted val	d messages, times, tin ues.	ner/counter present values, or	8
Analog comparator bits	A	A0 to A3	4	Used as program input conditions to output analog comparator comparison results. These bits can be used only for CPU Units with DC power supply.		9	
Timer/counter comparator bits	Р	P0 to Pf	16	Compare the present values of timers (T), holding timers (#), and counters (C). Comparison can be made between the same two counters or timers, or with constants.		10	
8-digit counter comparator bits	G	G0 to G3	4	Used to compare the present values of 8-digit counters (F) and output the comparison results.		11	
Button input bits	В	B0 to B7	8	Used as program input conditions and turn ON when operation buttons are		12	

Note: 1. For details, refer to the indicated item numbers on the following pages.

POI details, refer to the indicated term numbers on the following pages.
 CPU Units with 10 I/O points have 6 input bits (I0 to I5) and 4 output bits (Q0 to Q3). Output bit Q3 of Communications-type CPU Units, however, cannot be output externally. It can be used as a work bit.
 These input bits are not supported by LED-type CPU Units without display.

#### Biti, funkcije [4]

[: Normal	Set/Reset	A: Alternate	
Crigar.	S: Set	R: Reset	
[Q0	-  <sup>11</sup> SQ1	-  <sup>12</sup> 	-  <sup>13</sup>   AQ2
10	" <u>`</u> ĹĹŢ	12	13
	Q1	Q1	
Q0 will turn ON or OFF depending on the ON/OFF status of the execution condition I0.	Q1 will stay ON once the execution condition I1 has turned ON once. A reset is used to turn Q1 OFF.	Q1 is forced OFF when the execution condition I2 is turned ON.	Q2 alternates between turning ON and OFF when the execution condition I3 turns ON.

Dodatne funkcije izhodnega bita (številka 1) [4]

X: ON-delay timer	
Trigger input Reset input Setting Present value 0 Timer bit	Turns ON after set delay after the trigger input turns ON. Basic Operation Trigger input Output Main Applications When delayed operation or a time lag is required.
OFF-delay timer	
Trigger input Reset input Setting Present value 0 Timer bit	Stays ON while the trigger input is ON and turns OFF after a set delay after the trigger input has turned OFF. Basic Operation Trigger input Output Main Applications Useful for OFF delay circuits for lights or fans
O: One-shot pulse timer	
Trigger input Reset input Setting Present value 0 Timer bit	Turns ON for a set period after the trigger input turns ON and regardless of how long the trigger input remains ON. Basic Operation Trigger input Output Main Applications Useful for set operations where operation is always required during a regular period only.
F: Flashing pulse timer	
Trigger input Reset input Setting Present value 0 Timer bit	Repeatedly turns ON and OFF in a set cycle while the trigger input is ON. Basic Operation Trigger input Output Main Applications Useful for flashing emergency lights or sounding buzzers as the output for an alarm circuit.
w: iwin umer	Percentedly trees ON and OFF is a set such while the tries of the OV
Trigger input Reset input ORF-time OFF-time Present value 0 Timer bit	Hepeatedry turns ON and OFF in a set cycle while the trigger input is ON. Independent ON- and OFF-time settings are possible. Basic Operation Trigger input (OFF-time) (OFF-time) (OFF-time) Output Main Applications Useful for intermittent (ON/OFF) operation, such as that used for fans.

Uporaba časovnikov (številka 2) [4]

Trigger input Reset input Setting Present value	Turns ON after set delay after the trigger input turns ON. The present value is held while the trigger input is OFF. <b>Main Applications</b> To continue operation after momentary power loss or power interruptions Also can be used when delayed operation or a time lag is required.
--	--

Uporaba držalnih časovnikov (številka 3) [4]

Counter direction specification input Counter input Reset input Set value Present value 0000 Counter bit					The counter bit turns ON when the counter value (present value) reaches the set value (present value ≥ set value). The counter returns to 0 and the counter bit turns OFF when the reset input turns ON. Count inputs are not accepted while the reset input is turned ON. The counter present value and counter bit (ON/OFF) are held even if the operating mode is changed or the power supply is interrupted.
---	--	--	--	--	--

### Števec (številka 4) [4]



#### 8-bitni digitalni števec (številka 5) [4]



Tedenski časovnik (številka 6) [4]



Koledarski časovnik (številka 7) [4]

Backlight/display function screen switching	<ul> <li>L0: No backlight; No switching to display function screen (See note 1.)</li> <li>L1: Backlight; No switching to display function screen (See note 1.)</li> <li>L2: No backlight; Switching to display function screen (See note 2.)</li> <li>L3: Backlight; Switching to display function screen (See note 2.)</li> </ul>				
Display start position	X (digit): 00 to 11 x0 x11 Y (line): 0 to 3 y010/3				
Display object	CHR	Characters (Up to 12 characters - English, numerals, symbols)			
	DAT	Γ Month/day (5 digits □□/□□)			
	CLK	hour/minute (5 digits and a contract of the co			
	14 to 15	Analog-converted value (4 digits			
	T0 to Tf	Timer present value (5 digits)			
	#0 to #7	Holding timer present value (5 digits)			
	C0 to Cf	Counter present value (4 digits)			
	F0	8-digit counter present value(8 digits DDDDDDD)			
Monitoring	A: Can read settings during operation.				

- Note: 1. When L0 or L1 are selected to disable the display function screen, the display function screen will not be displayed automatically. Use operation buttons to move to the display function screen.
  - When L2 or L3 are selected, the ZEN switches to the display function screen if the display function is enabled and the specified data is displayed. The Main Screen will no longer be displayed. To display the Main Screen, change the CPU Unit to STOP mode.

Prikaz sporočil - bitov (številka 8) [4]



#### Analogni primerjalnik (številka 9) [4]



Časovni/števni primerjalnik (številka 10) [4]



8-bitni števni primerjalnik (številka 11) [4]



Uporaba operacijskih tipk - vhodnih bitov (številka 12) [4]



## Priloga K: Dimenzije krmilnika, napajalne in širitvene V/I enote

Dimenzije krmilnika Omron Zen v izvedbi s fiksnimi 20 V/I [4]



### Dimenzije napajalne enote [4]



Dimenzije širitvene V/I enote [4]

## Priloga L: Podajalna naprava (slike)



Pnevmatična cilindra (Šolski center Ptuj) [Vir: lasten]



Uporabljeni ventili (Šolski center Ptuj) [Vir: lasten]



Krmilnik Omron Zen (Šolski center Ptuj) [Vir: lasten]



Primer podajalne naprave (Šolski center Ptuj) [Vir: lasten]