



UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET
ODSJEK ZA AUTOMATIKU I ELEKTRONIKU

Dizajn ekspanzionog I/O modula za PC

ZAVRŠNI RAD
- PRVI CIKLUS STUDIJA -

Student:
Jasmin Hadžajlić

Mentor:
Doc.dr Emir Sokić, dipl.ing.el.

Sarajevo,
septembar 2017.

Doc. dr Emir Sokić, dipl.el.ing
Jasmina Zubača, MoE
Odsjek za automatiku i elektroniku
Sarajevo, 28.09.2017.

Tema za završni rad

studenta I ciklusa studija koji studira na ETF-u u skladu sa principima Bolonjskog procesa na Odsjeku za automatiku i elektroniku (šk. 2016/17.).

Tema: Dizajn ekspanzionog I/O modula za PC

Student: Jasmin Hadžajlić

Sažetak:

Prilikom vizuelne inspekcije i korištenja algoritama obrade slike u industrijskom okruženju, pojavljuje se potreba za interakcijom sistema za obradu slike sa okolnim hardverom (senzorima i aktuatorima). Jedan od nedostataka računara opće namjene je nemogućnost povezivanja, mjerjenja, akvizicije i generisanje odgovarajućih ulaznih i izlaznih (I/O) signala senzora odnosno aktuatora. Zadatak rada je dizajniranje elektroničkog modula koji se sastoji iz određenog broja analognih/digitalnih ulaza i izlaza koji zadovoljavaju neophodne uvjete za korištenje u industrijskom okruženju. Vezu između računara i modula je potrebno ostvariti putem RS485 sabirnice.

Polazna literatura:

- [1] Batchelor, Bruce G., and Paul F. Whelan. Intelligent vision systems for industry. Springer Science & Business Media, 2012.
- [2] Demant, Christian, C. Demant, and Bernd Streicher-Abel. Industrial image processing. Springer-Verlag, 1999.
- [3] Szeliski, Richard. Computer vision: algorithms and applications. Springer Science & Business Media, 2010.
- [4] Russ, John C. Image analysis of food microstructure. CRC press, 2004.
- [5] McRoberts, Michael, Brad Levy, and Cliff Wootton. Beginning Arduino. New York.: Apress, 2010.
- [6] Oxer, Jonathan, and Hugh Blemings. Practical Arduino: cool projects for open source hardware. Apress, 2011.
- [7] Perrin, Bob. "The art and science of RS-485.", 2015.

Mentor:

Doc. dr Emir Sokić, dipl.el.ing

Izjava o autentičnosti radova

Završni rad I ciklusa studija

Ime i prezime: Jasmin Hadžajlić
Naslov rada: Dizajn ekspanzionog I/O za PC
Vrsta rada: Završni rad I ciklusa studija
Broj stranica: 65

Potvrđujem:

- da sam pročitao dokumente koji se odnose na plagijarizam, kako je to definirano Statutom Univerziteta u Sarajevu, Etičkim kodeksom Univerziteta u Sarajevu i pravilima studiranja koja se odnose na I i II ciklus studija, integrirani studijski program I i II ciklusa i III ciklus studija na Univerzitetu u Sarajevu, kao i uputama o plagijarizmu navedenim na web stranici Univerziteta u Sarajevu;
- da sam svjestan univerzitetskih disciplinskih pravila koja se tiču plagijarizma;
- da je rad koji predajem potpuno moj, samostalni rad, osim u dijelovima gdje je to naznaceno;
- da rad nije predat, u cjelini ili djelimično, za stjecanje zvanja na Univerzitetu u Sarajevu ili nekoj drugoj visokoškolskoj ustanovi;
- da sam jasno naznačio prisustvo citiranog ili parafraziranog materijala i da sam se referirao na sve izvore;
- da sam dosljedno naveo korištene i citirane izvore ili bibliografiju po nekom od preporučenih stilova citiranja, sa navođenjem potpune reference koja obuhvata potpuni bibliografski opis korištenog i citiranog izvora;
- da sam odgovarajuće naznačio svaku pomoć koju sam dobio pored pomoći mentora i akademskih tutora/ica.

Sarajevo, 28. septembar 2017.

Jasmin Hadžajlić

Sažetak

U okviru ovog rada obrađeni su svi aspekti dizajna i implementacije ekspanzionog I/O modula za PC. Opisani su aktuelni industrijski standardi u oblasti I/O uređaja i različiti tipovi ulazno-izlaznih kanala. Predstavljena su konkurentna komercijalna rješenja, a zatim je koncipirano idejno rješenje jednostavnog I/O modula. Prezentirana je hardverska realizacija sistema, uključujući mikrokontrolersku jedinicu, RS485 komunikaciju, kao i električne sheme prilagođenja industrijskih ulaza i izlaza na standardne naponske nivoje. Pojašnjene su moguće konfiguracije upotrebe modula. Razvijen je protokol za komunikaciju modula sa računarom putem RS485 sabirnice. Opisane su sve funkcionalnosti sistema, te odgovarajuće softverske implementacije.

Abstract

The aim of this work is to give insight into all aspects of the design and development of an I/O expansion module for the PC. Contemporary industrial standards in the field of I/O devices, along with different types of input-output channels are described. Competitive commercial solutions were introduced, and subsequently a conceptual solution of a simple I/O module was conceived. The hardware implementation of the system, including the microcontroller unit, RS485 communication, as well as the electric circuits required for interfacing the industrial inputs and outputs to standard voltage levels are presented. Possible usage configurations of the module are explained. A protocol for communicating the module with computer via the RS485 bus was developed. All system functionalities and their corresponding software implementations are described.

Sadržaj

Popis slika	vii
Popis tabela	viii
1 Uvod	1
1.1 Pregled stanja u oblasti istraživanja	2
1.2 Prijedlog rješenja	4
2 Hardverska komponenta IO modula	5
2.1 Mikrokontroler	5
2.1.1 Atmel ATmega328p	6
2.1.2 Funkcionalni moduli i mogućnosti mikrokontrolera ATmega328P . . .	6
2.1.3 Konfiguracija pinova mikrokontrolera ATmega328P (32-pin TQFP) . .	7
2.1.4 Minimalno električko okruženje mikrokontrolera ATmega328p . .	9
2.1.5 Programiranje programske memorije - ISP	10
2.2 Komunikaciona komponenta IO sistema	12
2.2.1 Implementacija RS485 komunikacije	12
2.3 Adresiranje IO modula	14
2.4 Ulazni portovi	16
2.4.1 Digitalni ulaz	16
2.4.2 Analogni ulaz	19
2.4.3 Enkoderski ulaz	21
2.5 Izlazni portovi	22
2.5.1 Digitalni izlaz	22
2.5.2 Analogni izlaz	24
2.6 Dizajn štampane pločice - PCB	27
2.6.1 Projektovanje štampane pločice u programskom paketu Eagle	27
2.6.2 Raspored ulaza i izlaza na štampanoj pločici	31
2.6.3 Fizička realizacija štampanih pločica	33
3 Softverska komponenta IO modula	35
3.1 Softverske funkcionalnosti	35
3.2 Razvijeni komunikacijski protokol	36
3.2.1 Format poruke	36
3.2.2 Implementirane poruke	37
4 Testiranje rada modula	40
5 Zaključak	43

Prilozi	44
A Teoretski aspekti komunikacije	45
A.1 RS485	45
A.1.1 Terminacija prenosnih vodova	45
A.2 Topologija mreže	46
B Programske kodove	48
B.1 Program mikrokontrolera	48
Literatura	64

Popis slika

1.1	Izgled inkrementalnog enkodera	3
2.1	Blok struktura modula	5
2.2	Fizički izgled mikrokontrolera ATmega328P (32-pin TQFP)	6
2.3	Blok dijagram mikrokontrolera ATmega328P [1]	8
2.4	Raspored pinova mikrokontrolera ATmega328P (32-pin MLP)	9
2.5	Minimalno elektroničko okruženje mikrokontrolera ATmega328P	10
2.6	Raspored pinova ISP konektora	11
2.7	Primjer implementacije RS485 komunikacije	13
2.8	Shema sklopa za RS485 komunikaciju	14
2.9	Raspored pinova MAX485 IC	14
2.10	Načini hardverskog adresiranja	15
2.11	Sklop za učitavanje adrese uređaja	16
2.12	Prilagođenje naponskih digitalnih ulaza za mikrokontroler	17
2.13	Prilagođenje naponskih digitalnih ulaza PNP tipa za mikrokontroler	18
2.14	Prilagođenje strujnog digitalnog ulaza na mikrokontroler	18
2.15	Finalna shema digitalnog ulaza bez izolacije	18
2.16	Finalna shema digitalnog ulaza sa izolacijom	19
2.17	Shema prilagođenja analognog unipolarnog napona za mikrokontroler	20
2.18	Shema prilagođenja analognog bipolarnog napona za mikrokontroler	20
2.19	Shema prilagođenja analognog strujnog signala za mikrokontroler	20
2.20	Shema prilagođenja analognog ulaza korištena u realizaciji modula	21
2.21	Shema prilagođenja enkoderskog ulaza korištena u realizaciji modula	22
2.22	Shema spajanja reljevnih izlaza na mikrokontroler	22
2.23	Shema spajanja tranzistorskih izlaza na mikrokontroler	23
2.24	Shema spajanja naponskog digitalnog izlaza sa komplementarnim tranzistorima	23
2.25	Shema spajanja tranzistorskih izlaza na mikrokontroler korištena u realizaciji modula	24
2.26	Shema spajanja relejnog izlaza korištena u realizaciji modula	25
2.27	Shema spajanja naponskog analognog izlaza	25
2.28	Električna shema spajanja mikrokontrolera sa breakout pinovima	27
2.29	Izgled štampane ploče mikrokontrolerskog dijela	28
2.30	Električna shema spajanja dijela za komunikaciju i adresiranje	28
2.31	Izgled štampane ploče dijela za komunikaciju i adresiranje	29
2.32	Električna shema prilagođenja industrijskih ulaza i izlaza na mikrokontroler	29
2.33	Izgled štampane ploče dijela sa ulazima i izlazima	30
2.34	Raspored ulaza, izlaza i konfiguracijskih headera	31
2.35	Izgled štampanih pločica IO modula, nakon izrade	33

2.36 Izgled štampanih pločica IO modula, nakon lemljenja	33
2.37 Izgled IO modula bez kućišta	34
2.38 Izgled IO modula sa kućištem	34
3.1 Dijagram toka softvera mikrokontrolera	35
A.1 Diferencijalni naponi RS485 predajnika	45
A.2 Terminacija vodova	46
A.3 Topologije mreže	47

Popis tabela

1.1	Osnovne karakteristike <i>Modicon</i> kontrolera	2
2.1	Mogućnosti konfiguracije digitalnih ulaza bez izolacije	31
2.2	Mogućnosti konfiguracije digitalnih ulaza sa izolacijom	31
2.3	Mogućnosti konfiguracije analognih ulaza	32
2.4	Mogućnosti konfiguracije relejnih izlaza	32
2.5	Oznake ulaza i izlaza IO modula	32
3.1	Format poruke korišten za realizaciju komunikacije uređaja	36
3.2	Oznake ulaza i izlaza IO modula	37
3.3	Kodovi poruke greške	39
4.1	Odgovori uređaja na poslane poruke	41

Poglavlje 1

Uvod

U industrijskom sistemu za vizuelnu inspekciju neophodna je interakcija sistema sa okolinom, tj. mjerjenje određenih parametara i izvršavanje zadanih akcija [2]. Navedene funkcije u takvom sistemu obavljaju ulazno-izlazni moduli.

Ulazno-izlazni moduli (eng. *input-output modules*, skr. IO moduli) su električni uređaji koji povezuju sistem za obradu informacija, najčešće računar, sa ostatkom sistema upravljanja. Dakle, ovi uređaji vrše očitanje stanja sistema i mjerjenje karakterističnih veličina (ulaz), te djeluju na sistem u zavisnosti od instrukcije sistema za obradu informacija (izlaz).

Ulazi IO modula u industriji su izlazi različitih električnih senzora, koji mjere različite veličine, kako električne, tako i neelektrične. Izlazi senzora mogu biti naponski i strujni, te mogu biti analogni i diskretni. Prema tome i ulazi IO modula se dijele na:

- Digitalni strujni ulaz,
- Digitalni naponski ulaz,
- Analogni naponski ulaz,
- Analogni strujni ulaz.

U industrijskom postrojenju neizostavno je korištenje električnog motora. Za upravljanje motorom potrebno je na neki način mjeriti njegovu brzinu, što se najčešće radi upotrebom enkodera. Prema tome pored navedenih ulaza potrebno je da IO modul posjeduje i enkoderski ulaz (ulaz na koji se direktno spajaju izlazi enkodera).

Od izlaza IO modula u industriji zahtjeva se da mogu upravljati različitim uređajima na osnovu instrukcije računara. Uređaji kojima se upravlja su različiti električni motori (AC, DC), zatim grijaci, rasvjetna tijela i drugi, u zavisnosti od potrebe. Prema tome izlazi IO modula trebaju imati mogućnost upravljanja pomenutim uređajima, pa se razlikuju sljedeće vrste izlaza:

- Analogni/digitalni strujni izlaz,
- Analogni/digitalni naponski izlaz,
- Relejni izlaz,
- PWM izlaz.

U nastavku poglavlja dat je pregled stanja u oblasti istraživanja, a zatim i prijedlog rješenja.

1.1 Pregled stanja u oblasti istraživanja

Uređaj koji se u velikoj mjeri koristi u industriji za automatizirano upravljanje dijelovima sistema je PLC (programabilni logički kontroler, eng. *Programmable logic controller*). PLC je elektronski uređaj koji posjeduje programabilnu memoriju za smještanje instrukcija kojima se realizuju specifične funkcije, kao što su logičke operacije i sekvenciranje, brojanje i aritmetičke operacije, a u cilju upravljanja različitim mašinama i procesima putem digitalnih i/ili analognih ulazno/izlaznih modula [3].

Karakteristike PLC-ova (broj i vrsta ulaza i izlaza, količina memorije, vrsta komunikacije itd.) zavise od grane industrije u kojoj se koriste. Tako imamo široku paletu PLC-ova različitih karakteristika. Kompanije koje proizvode PLC-ove su *Siemens*, *Omron*, *Allan Bradley*, *Schneider electric* i drugi. Svaki od navedenih proizvođača nudi široku paletu PLC-ova za različite namjene, od komercijalnih, industrijskih preko sigurnosnih, pa sve do PLC-ova posebne namjene [4][5][6][7]. U nastavku će biti pobrojane karakteristike osnovnih PLC-ova, navedenih kompanija, koji se koriste u industriji.

Kompanija *Schneider Electric* proizvodi *Modicon* kontrolere [4]. U tabeli 1.1 pobrojani su ulazi i izlazi tih kontrolera. Vidi se da izlazni signali ovih kontrolera posjeduju iste osobine. Također uočava se da *Modicon M241* ne posjeduje analogne ulaze. Moguće je proširiti mogućnosti ovih kontrolera dodavanjem ekspanzionih modula.

Model	DI	AI	DO	Opseg analognog ulaza	Tip izlaza	Izlazni signal
Modicon M221	8	2	8	0...10 V	Tranzistorski	24 VDC, 0,5 A
Modicon M241	14	0	10	-	Tranzistorski, Relejni	24 VDC, 0,5 A, 125 VDC, 2 A, 250 VAC, 2 A
Modicon M258	26	4	16	4...20 mA, -10...10V	Tranzistorski	24 VDC, 4 mA

Tabela 1.1: Osnovne karakteristike *Modicon* kontrolera

Ekspanzionalni moduli kompatibilni sa *Modicon* kontrolerima su *Modicon TM3* ekspanzionalni moduli. U katalogu ovog proizvoda pobrojane su osnovne karakteristike ovih modula [8].

Digitalni ekspanzionalni moduli mogu sadržavati naizmjenične i istosmjerne ulaze, čiji su naponski nivoi 120 VAC, odnosno 24 VDC. Moguće je izvršiti izbor smjera toka struje (sink/source) ovih ulaza. Izlazi digitalnih modula su tranzistorski i relejni. Nazivna struja relejnih izlaza je 2 A, 7 A ili 8 A, u zavisnosti o kojem se proizvodu konkretno radi. Tranzistorski izlazi su sink ili source, nazivnih struja 0,1 A, 0,5 A, 2 A ili 4 A, u zavisnosti od konkretnog proizvoda.

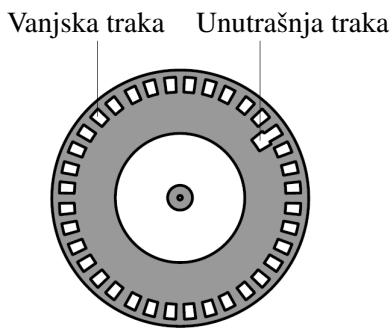
Analogni ekspanzionalni moduli TM3 mogu sadržavati strujne i naponske ulaze. Postoje dva opsega za strujne ulaze 0...20 mA i 4...20 mA. Opsezi naponskih ulaza su -10...10 V i 0...10 V. Rezolucije za obe vrste ulaza su 16 bita (ili 15 bita i bit za znak), te 12 bita (ili 11 bita i bit za znak), u zavisnosti od modula. Vrijeme čitanja analognih ulaza je između 1ms i 10ms (podesivo). Osim naponskih i strujnih analognih ulaza, ovi moduli mogu sadržavati i ulaze za temperaturne senzore (termoparove i temperaturne sonde). Izlazi analognih ekspanzionalnih modula također mogu biti strujni i naponski. Opsezi ovih izlaza su isti kao i kod analognih ulaza. Vrijeme konverzije signala je podesivo u intervalu od 1 ms do 10 ms. Rezolucija analognih izlaza je 11 bita (ili 10 bita i bit znaka).

Kompanija *Siemens* proizvodi kontroler *S7-1200* [5], koji se može koristiti za iste potrebe kao i navedeni *Modicon* kontroleri. *S7-1200* objedinjava mikroprocesor, napojnu jedinicu, ulaze i izlaze za upravljanje motorom, analogne ulaze, te port za komunikaciju sa ostalim dijelovima sistema. Navedeni modul ima i ekspanzionate module, kojima se povećava opseg njegovih mogućnosti. Ulazi i izlazi ovog PLC i njegovih ekspanzionih modula imaju identične naponske nivoe kao i Modicon kontroleri s razlikom što *S7-1200* dodatno ima digitalne ulaze i izlaze digitalnog naponskog nivoa 5 V. Razlike se uočavaju i u izlaznim strujama digitalnih izlaza, ali su istog reda veličine kao i kod Modicon kontrolera. Analogni ulazi *Siemens*-ovog kontrolera imaju manju rezoluciju, tj. 11 bita (10 bita i 1 bit za znak), ali i manje vrijeme konverzije (oko 600 μ s).

Kontroleri kompanije *Allan Bradley* i *Omron*, koji su sličnih mogućnosti kao gore navedeni kontrolери, su *Micro800* i *CP1* kontroleri, respektivno [7][6]. Naponski nivoi ulaza i izlaza ovih kontrolera su isti kao i kod *Modicon* kontrolera. *CP1* serija dodatno sadrži i naponski nivo od 5 V za digitalne ulaze, kao i naponske nivoe 0...5 V i 1...5 V za analogne ulaze i izlaze. *Micro800* familija kontrolera dodatno ima i istosmjerne digitalne ulaze naponskog nivoa 12 V. Ostale karakteristike ova dva kontrolera su dosta slične karakteristikama gore navedenih kontrolera.

Osim trenutnog stanja IO modula na tržištu potrebno je navesti i karakteristike aktualnih senzora i aktuatora. U ponudi kompanije *Schneider electric* za senzore temperature, pritiska, detektore dima, te senzore svjetlosti [9], uočava se da su izlazni signali analognih senzora između 0 i 10 V za naponske, te između 4 i 20 mA za strujne izlaze. Za induktivne senzore [10], koje nudi pomenuta kompanija, postoje različite vrste digitalnih izlaznih signala. Izlazni signali ovih senzora su NPN i ili PNP tipa, sa mirnim i ili radnim kontaktima, najčešće 24 VDC naponskog nivoa u zavisnosti od senzora.

Pored navedenih senzora u industriji se za mjerjenje pozicije i brzine koristi enkoder, što je navedeno na početku. U tu svrhu se koristi inkrementalni enkoder. Inkrementalni enkoder je kružnog oblika (slika 1.1) i sadrži dvije vanjske trake prozora istog broja (oznake A i B), koje su međusobno pomjerene. Princip rada bazira se na brojanju impulsa koji se formiraju prekidanjem svjetlosti (optički enkoder) pri rotaciji, zbog prozora na trakama. Broj prozora predstavlja rezoluciju enkodera. Također postoji i jedan prozor na unutrašnjoj traci (oznaka 0).



Slika 1.1: Izgled inkrementalnog enkodera

Izlazi inkrementalnog enkodera su kanali A, B i 0, koji se vode na ulaze PLC-a. Ulazi PLC na koje se enkoder spaja su tzv. brzi ulazi/brojači (eng. *high speed counters*), dok se brzina i pozicija računaju softverski. Kompanija *Schneider electric* proizvodi inkrementalne enkodere *OsiSense XCC* [11]. Naponski nivoi na izlazu ovih enkodera su 5 V ili u opsegu 11...30 V, što

zavisi od izbora enkodera i napona napajanja. Rezolucije enkodera su 100, 360, 500, 1000 ili 1024 ppr (eng. *pulses per revolution*, broj impulsa po obrtaju).

Iz priloženog se može zaključiti da su karakteristike ulaza i izlaz PLC-ova različitih proizvođača dosta slične, tj. standardizovane, te prilagođene različitim vrstama senzora tih proizvođača. Prema tome moguće je uzeti jedan uređaj kao referentni i na osnovu njega vršiti izbor i proračun korištenih komponenti IO modula. Kao referentni PLC u okviru ovog rada uzet je *Modicon M241* kontroler zajedno sa svojim ekspanzionim modulima.

1.2 Prijedlog rješenja

Na osnovu pregleda stanja u oblasti istraživanja, te zaključka o korištenju *Modicon M241* uređaja kao referentnog, u ovom dijelu rada potrebno je specificirati kakav uređaj je potrebno realizovati. Kako će uređaj biti korišten u industrijskom sistemu za vizuelnu inspekciju, potrebno je da posjeduje određene mogućnosti, kao što su: upravljanje različitom vrstom motora, očitavanje vrijednosti senzora za položaj objekata i slično. Shodno navedenom, potrebno je da uređaj posjeduje određen broj digitalnih i analognih ulaza, odnosno izlaza, te enkoderski ulaz i PWM izlaze. U zavisnosti od broja potrebnih uređaja i njihove upotrebe, moguće je implementirati više IO modula za specifične namjene (npr. modul za upravljanje DC motorom, koračnim motorom, AC motorom, modul za akviziciju podataka, sigurnosni modul i slično). U ovoj fazi razvoja, dizajniran je univerzalni IO modul, koji služi za testiranje i demonstraciju mogućnosti ovakvih modula, te poređenje sa komercijalnim rješenjima. Osim zahtjeva na ulaze i izlaze modula, potrebno je postaviti zahtjeve za komunikaciju. U ovom radu korišten je RS485 komunikacijski standard, koji povezuje module sa računarom.

Na osnovu izloženog u okviru ovog rada dizajniran je modul sljedećih karakteristika:

- 4 digitalna ulaza
- 2 analogna ulaza
- 1 enkoderski ulaz
- 2 digitalna izlaza
- 2 digitalna/PWM izlaza
- 1 analogni izlaz

Ovaj završni rad je struktuiran na način opisan u nastavku.

U drugom poglavlju dat je dizajn hardvera IO modula. Navedeni su mogući načini spajanja ulaza i izlaza, kao i njihove električne sheme, zatim je pojašnjeno korištenje mikrokontrolera kao upravljačkog dijela ovog modula, te komunikacijski standardi u industrijskom okruženju. Na kraju poglavlja dat je dizajn štampane pločice samog modula.

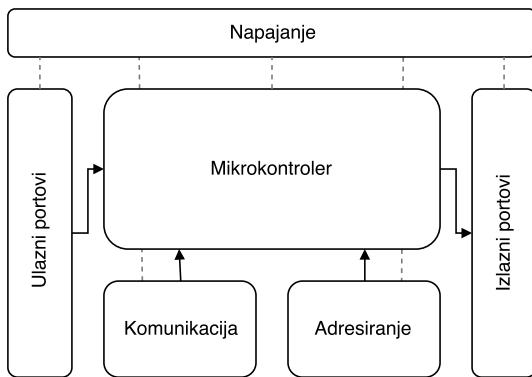
Treće poglavlje je sačinjeno od softverskog dijela. U ovom poglavlju navedene su realizovane funkcionalnosti modula, te formati poruka i naredbi.

U četvrtom poglavlju rada izloženi su načini i rezultati testiranja rada modula. Na osnovu postavljenog scenarija slanja poruka, posmatrano je ponašanje sistema.

Poglavlje 2

Hardverska komponenta IO modula

U ovom poglavlju će biti predstavljeni načini realizacije hardverskog dijela IO modula, a zatim će se odabrat odgovarajuća rješenja za konkretnu realizaciju. Osnovni dijelovi modula su prikazani na slici 2.1. Dijelovi modula opisani su u nastavku.¹



Slika 2.1: Blok struktura modula

2.1 Mikrokontroler

Mikrokontroler (skraćeno MCU od eng. *microcontroller unit*) je računar malih dimenzija implementiran na jednom integriranom kolu (*čipu*) [3]. U modernoj tehnologiji, to su SoC sistemi (*Sistem na čipu*, eng. *System on a Chip*). Jedna mikrokontrolerska jedinica sadrži jednu ili više procesorskih jezgri, zajedno sa memorijom i programabilnim perifernim ulazno/izlaznim modulima. Programska memorija u obliku feroelektrične RAM, NOR Flash ili OTP ROM memorije je također često prisutna, kao i određena količina RAM memorije. Mikrokontroleri se dizajniraju za ugradbene (eng. *embedded*) aplikacije, za razliku od mikroprocesora koji se koriste u personalnim računarama ili u aplikacijama opće namjene.

Mikrokontroleri se koriste pri realizaciji proizvoda i uređaja sa automatskim upravljanjem, kao što su upravljački sistemi za pogone automobila, implantacijski medicinski uređaji, daljinski upravljači, uredske mašine, alati, igračke i drugi ugradbeni sistemi. Smanjenjem veličine i

¹Dizajn i implementacija dijela sa mikrokontrolerom, te dijela sa komunikacijom i adresiranjem realizovani su u suradnji sa kolegom Hasanom Grošićem, te su sadržani i u završnom radu "Dizajn kontrolabilnog modula za osvjetljenje"

cijene u odnosu na dizajn koji koristi diskretne elemente mikroprocesora, memorije i ulazno/izlaznih modula, mikrokontroleri omogućuju ekonomičniji način digitalnog upravljanja mnogo više uređaja i procesa.

2.1.1 Atmel ATmega328p

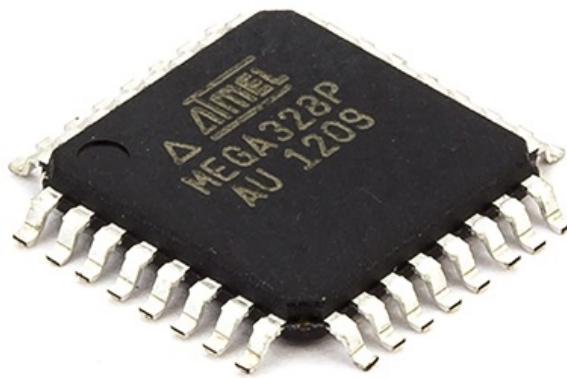
Atmel ATmega328P je 8-bitni CMOS mikrokontroler male snage baziran na tzv. AVR poboljšanoj (eng. *enhanced*) RISC arhitekturi [1].

Atmel AVR jezgra kombinuje bogat instrukcijski set sa 32 regista opće namjene. Svaki od tih 32 registra je direktno spojen na aritmetičko-logičku jedinicu (ALU), što omogućuje pristup ka dva nezavisna registra pomoću jedne instrukcije koja se izvršava u jednom takt ciklusu. Rezultujuća arhitektura je efikasnija u smislu programiranja, te postiže procesorske brzine blizu 1 MIPS (*milioni instrukcija u sekundi*) pri radnoj frekvenciji od 1 MHz, što je i do deset puta brže od konvencionalnih CISC mikrokontrolera. Ova činjenica omogućuje projektantima sistema da optimiziraju dizajn uređaja u smislu potrošnje, odnosno, u smislu brzine procesiranja.

Mikrokontroler ATmega328P se proizvodi u četiri fizički različite varijante (pakovanja):

- 28-pin PDIP,
- 28-pin MLF,
- 32-pin TQFP i
- 32-pin MLF.

Za potrebe ovog rada razmatrat će se varijanta **32-pin TQFP**, koja je prikazana na slici 2.2.



Slika 2.2: Fizički izgled mikrokontrolera ATmega328P (32-pin TQFP)

2.1.2 Funkcionalni moduli i mogućnosti mikrokontrolera ATmega328P

Mikrokontroler ATmega328P posjeduje sljedeće funkcionalnosti:

- 32KB sistemski programabilne Flash memorije sa Čitaj-Dok-Pišeš (eng. *Read-While-Write*) sposobnostima,
- 1KB EEPROM memorije,
- 23 I/O linije opće namjene,
- 32 radna registra opće namjene,
- prekidni sistem sa 26 različitih prekida,
- Real Time brojač (RTC),
- tri fleksibilna tajmera/brojača (2x8-bitni i 1x16-bitni) sa poredbenim modovima i mogućnošću širinsko-impulsne modulacije (PWM),
- 1 serijski programabilni USART interfejs,
- 1 bajtno-orjentirani 2-žični serijski interfejs (I2C),
- 8-kanalni 10-bitni AD konvertor,
- programabilni Watchdog tajmer sa internim oscilatorom,
- SPI serijski port i
- 6 softverski selektabilnih modova uštede energije.

ISP Flash (skr. od eng. *In-system Programming*) koji je prisutan na čipu, dozvoljava programskoj memoriji da bude reprogramirana *kroz sistem* preko SPI serijskog interfejsa, konvencionalnim programatorima neizbrisive (eng. *nonvolatile*) memorije, ali i *bootloader* programima smještenim na AVR jezgri. O načinima programiranja programske memorije će više biti riječi u kasnijim odjeljcima ovog rada.

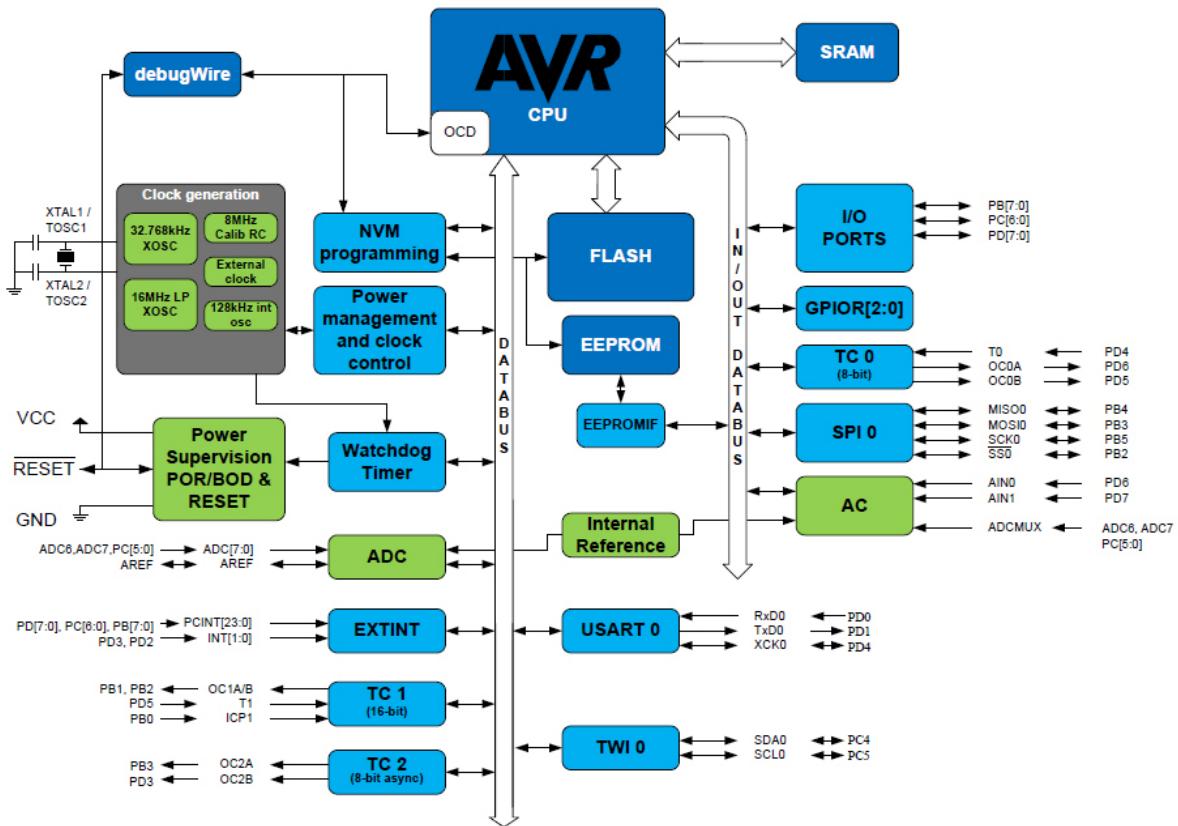
2.1.3 Konfiguracija pinova mikrokontrolera ATmega328P (32-pin TQFP)

Verzija mikrokontrolera razmatrana u ovom završnom radu (32-pin TQFP) ima 32 pina od kojih je većina multipleksirana, pa jedan pin može da obavlja više uloga u zavisnosti od aplikacije. Na slici 2.4 prikazan je raspored pinova sa odgovarajućim multipleksiranim funkcijama istih, a njihov opis dat je u nastavku.

VCC: Napon napajanja.

GND: Masa.

Port B (PB[7:0]) XTAL1/XTAL2/TOSC1/TOSC2: Port B je 8-bitni bidirekcion I/O port sa internim pull-up otpornicima (odabranim za svaki bit). Pinovi porta B prelaze u stanje visoke impedanse kada neki od uslova reseta postane aktivran, čak i kada je sat deaktiviran. Zavisno od postavki konfiguracijskih bita za selekciju takta (eng. *clock selection fuse*), pin PB6 može se koristiti kao ulaz u invertirajuće oscilatorsko pojačalo i kao ulaz u operativni krug internog sata, a PB7 kao izlaz iz invertirajućeg oscilatorskog pojačala. Ako se za generisanje takta čipa koristi interni kalibrirani RC oscilator, onda se PB[7:6] koristi kao TOSC[2:1] ulaz za asinhroni tajmer/brojač2 ako je AS2 bit u ASSR registru setovan.



Slika 2.3: Blok dijagram mikrokontrolera ATmega328P [1]

Port C (PC[5:0]): Port C je 7-bitni bidirekcion I/O port sa internim pull-up otpornicima (odabranim za svaki bit). Pinovi porta C prelaze u stanje visoke impedanse kada neki od uslova reseta postane aktivan, čak i kada je sat deaktiviran. Jedna od više funkcija pinova porta C jesu i analogni ulazi u AD konvertor.

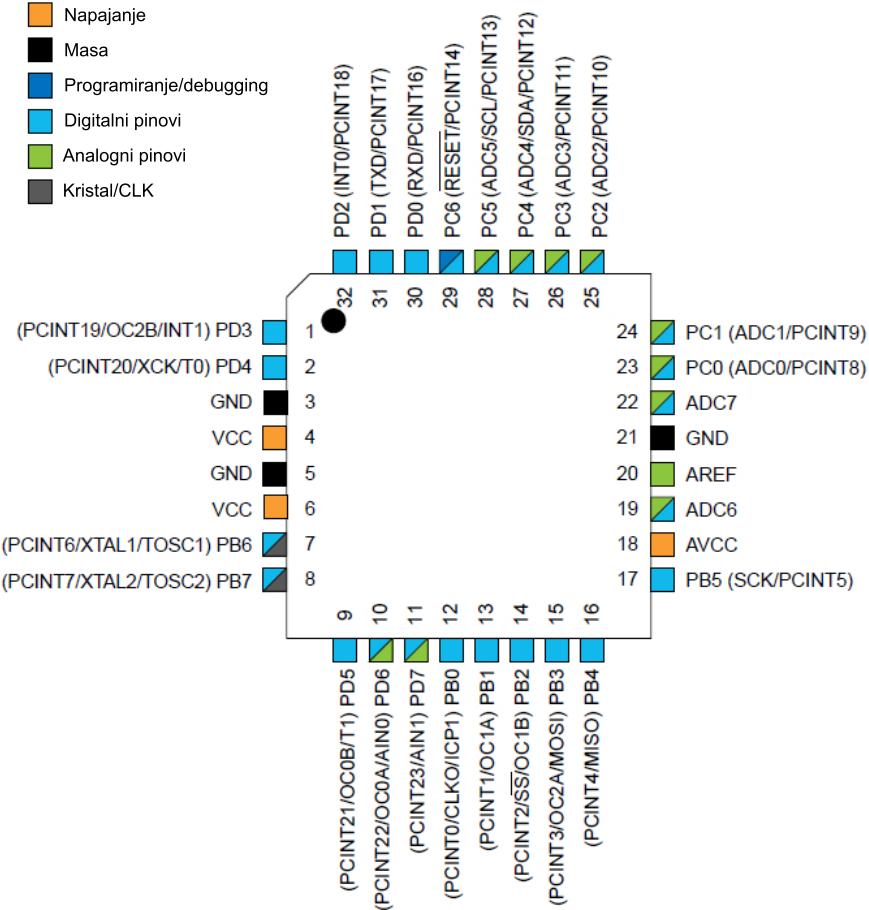
PC6/RESET: Ako je RSTDISBL konfiguracijski bit programiran, PC6 se koristi kao I/O pin. Bitno je naglasiti da se električne karakteristike pina PC6 razlikuju od karakteristika ostalih pinova porta C. Ako je RSTDISBL bit neprogramiran, PC6 se koristi kao RESET ulaz. Ulaz nivoa logičke nule na ovom pinu sa trajanjem dužim od barem jedne dužine takt impulsa, generisat će RESET, čak i kada sat nije aktiviran.

Port D (PD[7:0]): Port D je 8-bitni bidirekcion I/O port sa internim pull-up otpornicima (odabranim za svaki bit). Pinovi porta D prelaze u stanje visoke impedanse kada neki od uslova reseta postane aktivan, čak i kada je sat deaktiviran.

AVCC: AVCC služi kao ulaz za napon napajanja AD konvertora. Potrebno je da bude eksterni spojen na VCC, čak i kada se ADK ne koristi. Ako se ADK koristi, AVCC je potrebno spojiti na VCC preko niskopropusnog filtra.

AREF: AREF služi kao ulaz za analogni referentni naponski nivo za AD konvertor.

ADC[7:6]: ADC[7:6] služe kao analogni 10-bitni ulazni kanali u AD konvertor.



Slika 2.4: Raspored pinova mikrokontrolera ATmega328P (32-pin MLF)

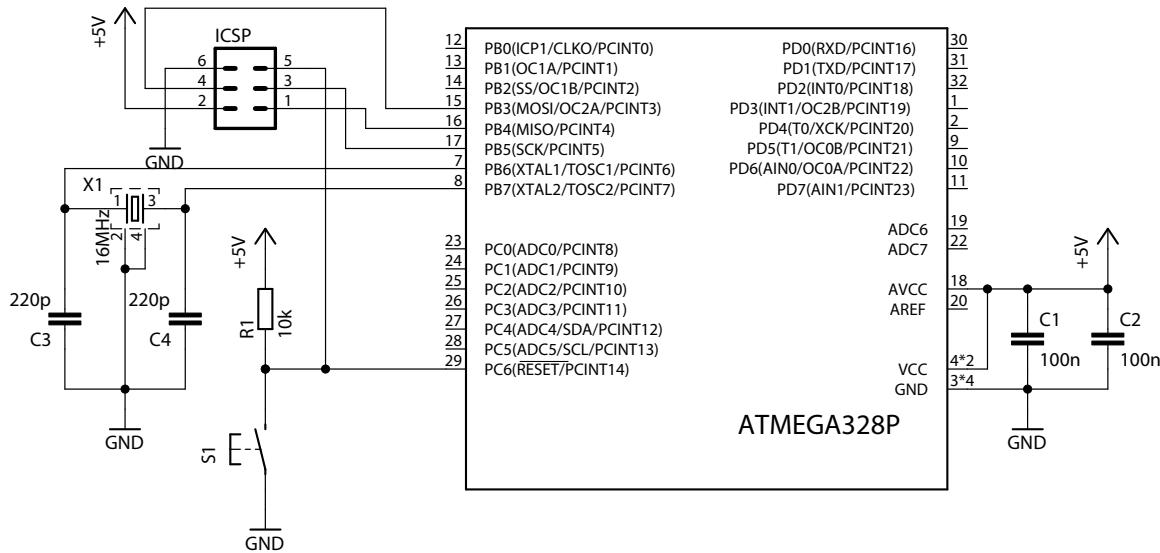
2.1.4 Minimalno elektroničko okruženje mikrokontrolera ATmega328p

Za pravilan rad mikrokontrolera ATmega328P, potrebno je osigurati odgovarajuće elektroničko okruženje, što podrazumijeva stabilan napon napajanja sa adekvatnim naponskim nivoima, kolo vanjskog oscilatora, kolo za resetovanje mikrokontrolera, te ISP interfejs za programiranje [12].

Obzirom da se mikrokontroler može napajati sa izvora čiji se naponi kreću između 1.8V i 5.5V, kao optimalan izbor odabire se napon napajanja od 5V kako bi se postigla kompatibilnost sa naponskim nivoima standardne 5V TTL logike. Svi pinovi mikrokontrolera labelirani sa VCC i GND se spajaju na odgovarajuće naponske nivoe izvora napajanja. Za dodatnu stabilizaciju napona napajanja i eliminaciju šumova koriste se razdvojni (eng. *decoupling*) kondenzatori sa kapacitetom od 100nF. Kolo vanjskog oscilatora sačinjeno je od kvarc kristala rezonantne frekvencije 16MHz i dva pločasta kondenzatora kapaciteta 22pF koji su odabrani shodno frekvenciji oscilatora. Uloga kristala se ogleda u stabilizaciji frekvencije *clock* signala. Za dovođenje vanjskog RESET signala na mikrokontroler koristi se taster/prekidač sa odgovarajućim *pull-up* otpornikom, dok je ISP interfejs za programiranje programske memorije realiziran u vidu 6-pinskog konektora.

Potpuna shema opisanog okruženja, zajedno sa mikrokontrolerom ATmega328P, prikazana je

na slici 2.5.



Slika 2.5: Minimalno elektroničko okruženje mikrokontrolera ATmega328P

2.1.5 Programiranje programske memorije - ISP

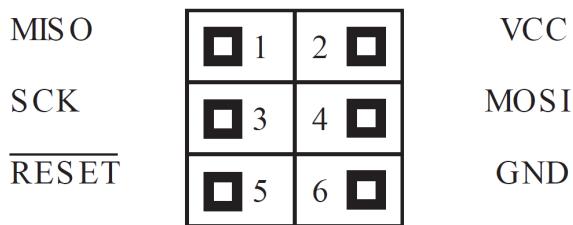
ISP programiranje (eng. *In-System Programming*), također poznato i kao ICSP (eng. *In-Circuit Serial Programming*), omogućuje programiranje i reprogramiranje programske memorije mikrokontrolera smještenog unutar krajnjeg sistema. Putem jednostavnog trožičnog SPI interfejsa, ISP programator serijski komunicira sa mikrokontrolerom i ima mogućnost reprogramiranja cjelokupne trajne memorije na čipu.

ISP eliminiše potrebu za fizičkim odstranjivanjem čipa iz sistema u cilju programiranja. Na taj način dolazi do velike uštede vremena i novčanih sredstava, kako tokom razvojne faze u laboratoriji, tako i u procesu terenske nadogradnje softvera ili parametara programa.

Programiranje pomoću ISP programatora se vrši preko 6-pinskog konektora čiji je raspored pinova prikazan na slici 2.6. Serijski periferalni interfejs - SPI (skr. od eng. *Serial Peripheral Interface*) se sastoji od tri signala: serijskog sata (SCK), Master In - Slave Out (MISO) i Master Out - Slave In (MOSI). Tokom programiranja mikrokontrolera, ISP se uvijek ponaša kao *master*, dok odredišni sistem operira kao *slave*.

ISP (*Master*) pomoću SCK linije obezbjeđuje takt potreban za komunikaciju. Svaki impuls na SCK liniji proslijedi jedan bit od programatora (*master*) do odredišnog mikrokontrolera (*slave*) na MOSI liniji. U isto vrijeme, svaki impuls na SCK liniji također proslijedi jedan bit od odredišnog mikrokontrolera (*slave*) do programatora (*master*) preko MISO linije.

Da bi se započelo serijsko programiranje, potrebno je aktivirati i držati aktivnom RESET liniju mikrokontrolera (niski log. nivo). Kako bi se automatizirao ovaj proces, preferira se da se programatoru omogući preuzimanje kontrole nad RESET linijom odredišnog mikrokontrolera



Slika 2.6: Raspored pinova ISP konektora

pomoću četvrte kontrolne linije ISP-a (RESET).

Ukoliko odredišni sistem nema vlastito napajanje, tada se za napajanje mikrokontrolera tokom programiranja koristi VCC linija programatora.

2.2 Komunikaciona komponenta IO sistema

Za realizaciju industrijskog sistema, bilo kakvog tipa, neophodna je komunikacija između komponenti sistema. U ovom završnom radu to je komunikacija između računara koji upravlja sistemom i samih modula sistema. Osobine koje treba da posjeduje takva komunikacija su:

- Velika brzina prenosa signala (reda Mbps),
- Prenos na veće udaljenosti (do 50 m),
- Otpornost na šumove,
- Velik broj učesnika u komunikaciji (do 32).

Razlikujemo dva osnovna načina na koje je moguće vršiti prenos podataka između predajnika i prijemnika, a to su paralelni i serijski način [13]. Prema tome razlikujemo paralelnu i serijsku komunikaciju. Paralelna komunikacija je prenos više podataka istovremeno između dva učesnika. Serijska komunikacija, za razliku od paralelne, je prenos više podataka preko jedne veze, na način da se podaci šalju jedan za drugim sekvencijalno u vremenu. Serijska komunikacija u odnosu na paralelnu zahtijeva manje veza, čime se smanjuje cijena njene implementacije. Paralelna komunikacija je brža i koristi se za prenos podataka na kraće udaljenosti (npr. računar - printer) [13].

Prema smjeru toka podataka komunikaciju možemo podijeliti na tri vrste [13]:

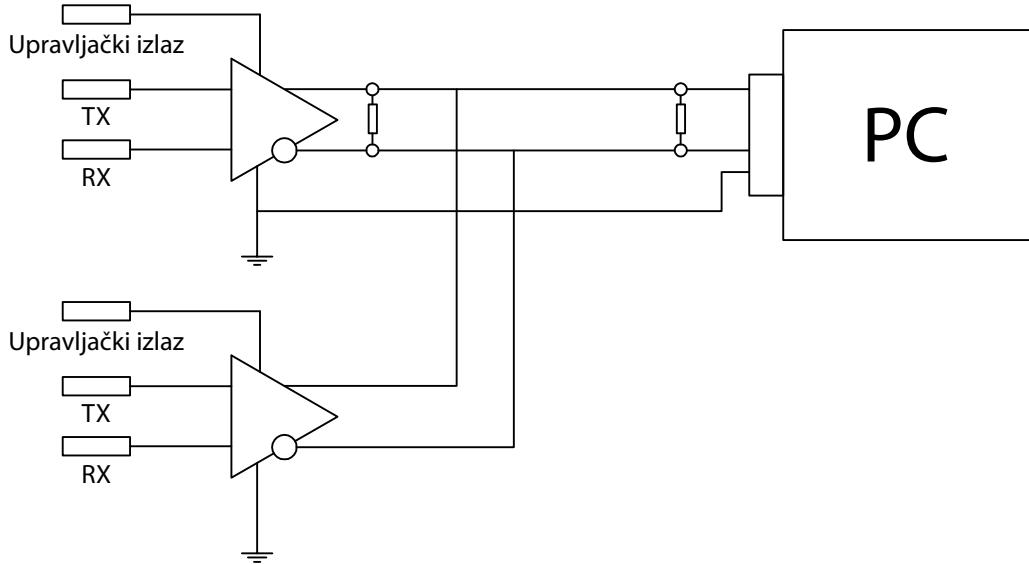
- *Simplex* – komunikacija između učesnika se uvijek obavlja u istom smjeru. Jedan uređaj samo šalje, a drugi samo prima podatke.
- *Half-duplex* – komunikacija je dvosmjerna, svi učesnici mogu slati i primati podatke, ali ne istovremeno. Jedan učesnik šalje podatke, a ostali primaju.
- *Full-duplex* – dvosmjerna komunikacija, moguće je istovremeno vršiti komunikaciju u oba smjera.

U ovom radu bit će korištena *half-duplex* komunikacija, u kojoj će računar upravljati komunikacijom nad uređajima u mreži.

Zbog industrijskih uslova rješenje koje se najbolje pokazalo u praksi je RS485 komunikacijski standard, koji se odnosi na fizički nivo komunikacije [14]. Detaljnija teoretska razmatranja oko RS485 standarda i drugih aspekata komunikacije, data su na kraju rada u prilogu A.

2.2.1 Implementacija RS485 komunikacije

Komunikacija treba da se vrši između glavnog računara i modula sistema, kako je ranije spomenuto. Moduli sistema su bazirani na mikrokontroleru. Mikrokontroler posjeduje pinove za serijsku komunikaciju (RX, TX). Osim ova dva pina, potreban je još jedan izlazni pin, koji određuje u kojem smjeru se vrši komunikacija. Pomenute pinove mikrokontrolera spajamo na integrisano kolo MAX485, koje služi za konverziju signala iz unipolarnog u diferencijalni. Također dodajemo otpornike za zatvaranje vodova. Blok shema je prikazana na slici 2.7. Na slici 2.8 prikazan je elektronički sklop koji realizira RS485 komunikaciju korištenu u sklopu ovog rada i raspored pinova na korištenom RJ-45 konektoru.



Slika 2.7: Primjer implementacije RS485 komunikacije

U nastavku su prikazane osnovne osobine integrisanog kola MAX485, korištenog za implementaciju komunikacije sistema.

Integrисано кло MAX485

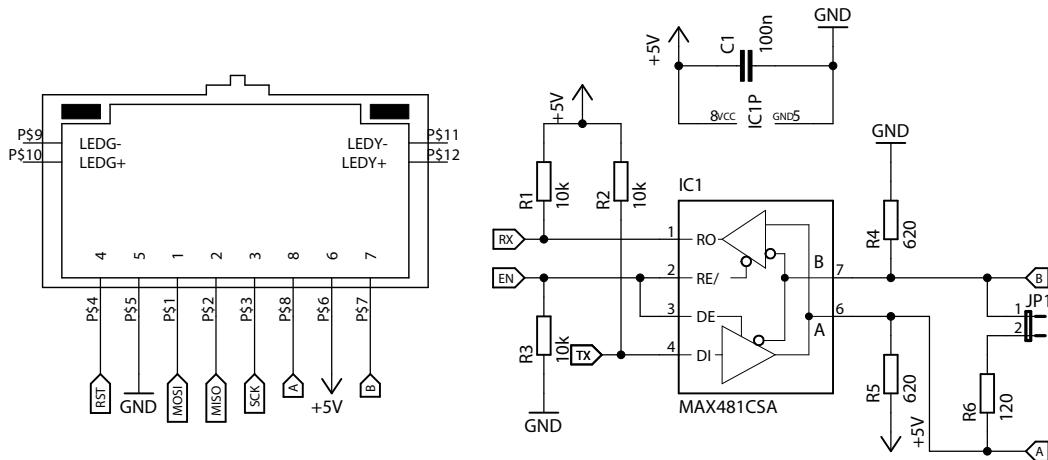
Integrисана кло MAX485 су примопредајници мале snage koji se koriste за implementaciju RS485 i RS422 komunikacije. Na slici 2.9 je prikazan raspored pinova MAX485 IC. Pinovi DE i RE služe za upravljanje smjerom komunikacije, pinovi RO i DI služe za primanje, odnosno slanje podataka. Pinovi A i B su diferencijalne linije RS485 komunikacije.

Ukoliko su DE i \overline{RE} u stanju logičke "0", diferencijalni signal sa A i B se prebacuje na pin RO, odakle prijemnik očitava podatak. Kada su DE i \overline{RE} na nivou logičke "1", podatak na pinu DI se prosljeđuje na A i B, i tako postavlja na sabirnicu.

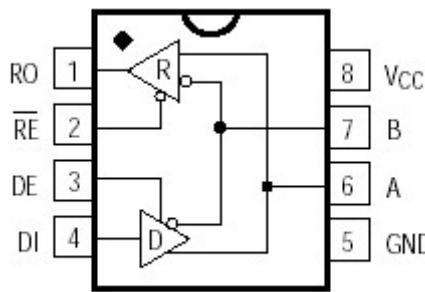
Osnovne karakteristike integralnog kola [15]:

- Ulazna otpornost $R_I = 12k\Omega$
- Napon napajanja $V_{CC} = 5V$
- Minimalni napon logičke "1" na ulazu $V_{IHmin} = 2V$
- Maksimalni napon logičke "0" na ulazu $V_{ILmax} = 0.8V$
- Brzina prenosa podataka do 2.5 Mbps
- *Half-duplex* komunikacija
- Maksimalno 32 korisnika

MAX485 integrисано кло има заштиту од prevelike disipacije snage, tako što izlaz postavi u stanje visoke impedanse. Također prijemni ulaz u kolo je postavljen na nivo logičke "1" ukoliko je odspojen.



Slika 2.8: Shema sklopa za RS485 komunikaciju



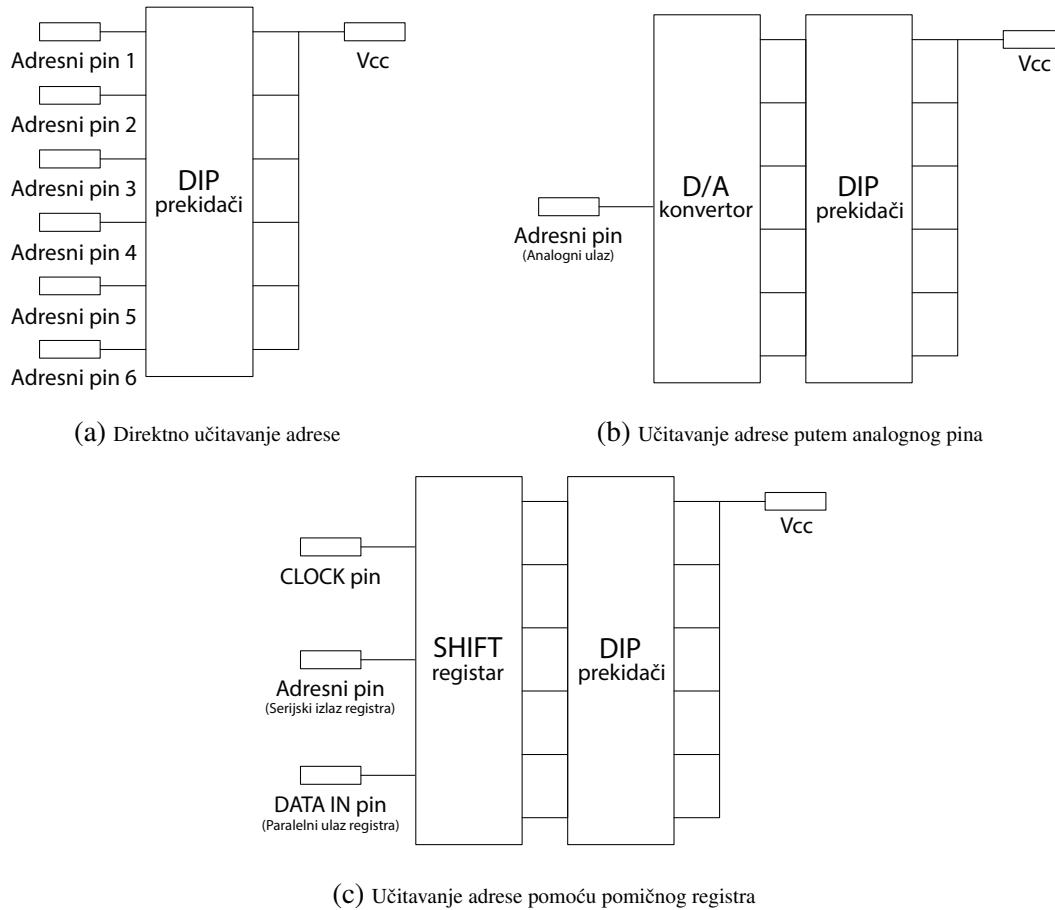
Slika 2.9: Raspored pinova MAX485 IC

2.3 Adresiranje IO modula

U *master/slave* komunikaciji *master* uređaj mora znati kome šalje koje podatke, a *slave* mora znati koji su podaci upućeni njemu. Zaključno tome, neophodan je neki način adresiranja prijemnika (*slave*-ova). Adresiranje se može implementirati na više načina, od kojih svaki ima svoje prednosti i nedostatke.

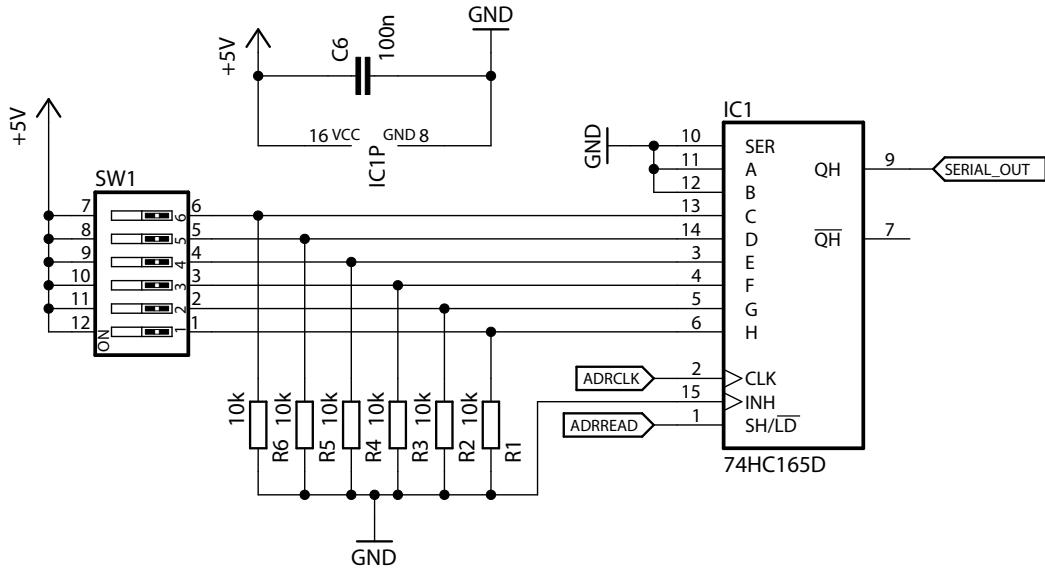
Adresiranje je moguće vršiti tako što se svakom prijemniku softverski unaprijed dodijeli adresa, koja se ne može mijenjati. Osim toga moguće je prijemnicima, uključivanjem jednog po jednog na mrežu, softverski dodjeljivati adrese. Drugi način je komplikovaniji, ali osigurava izbjegavanje preklapanja adresa prijemnika, odnosno da dva prijemnika imaju istu adresu.

Adresiranje je također moguće obaviti i hardverski. Jedan od načina je da se na ulaze uređaja preko više prekidača binarno postavlja adresa (slika 2.10a). Prednost ovakvog adresiranja leži u tome što se uvijek može vidljivo očitati adresa uređaja. Međutim, nedostatak je korištenje većeg broja pinova uređaja samo za adresiranje, što znači da se umanjuje opseg mogućnosti uređaja. Ovaj nedostatak je moguće u jednoj mjeri ispraviti korištenjem D/A konvertora, pri čemu se adresa učitava kao analogna vrijednost (slika 2.10b). Ista stvar se postiže ukoliko se za adresiranje koristi pomicni registar (eng. *Shift Register*) sa paralelnim ulazom i serijskim izlazom (slika 2.10c). Ovaj način podrazumijeva korištenje tri digitalna pina mikrokontrolera: izlaz koji generiše *clock* signal za pomijeranje registra, izlaz koji generiše signal za paralelno učitavanje u registar i ulaz koji učitava serijske podatke u mikrokontroler.



Slika 2.10: Načini hardverskog adresiranja

Način učitavanja adrese kao analogne vrijednosti, iako koristi samo jedan pin mikrokonrolera, jako je podložan greškama u očitanju koje su karakteristične za A/D konverziju. Naime, pojava šuma tokom konverzije može uzrokovati pogrešno očitanje nižih bita, a samim time i adrese. Stoga jasno je da je opravdano koristiti način učitavanja adrese koji se realizira pomoću pomočnog registra i tri digitalna pina mikrokontrolera. Na slici 2.11 prikazana je shema sklopa koji realizira opisani način adresiranja. Kao pomočni registar korišteno je integrисано kolo 74HC165D [16].



Slika 2.11: Sklop za učitavanje adrese uređaja

2.4 Ulazni portovi

2.4.1 Digitalni ulaz

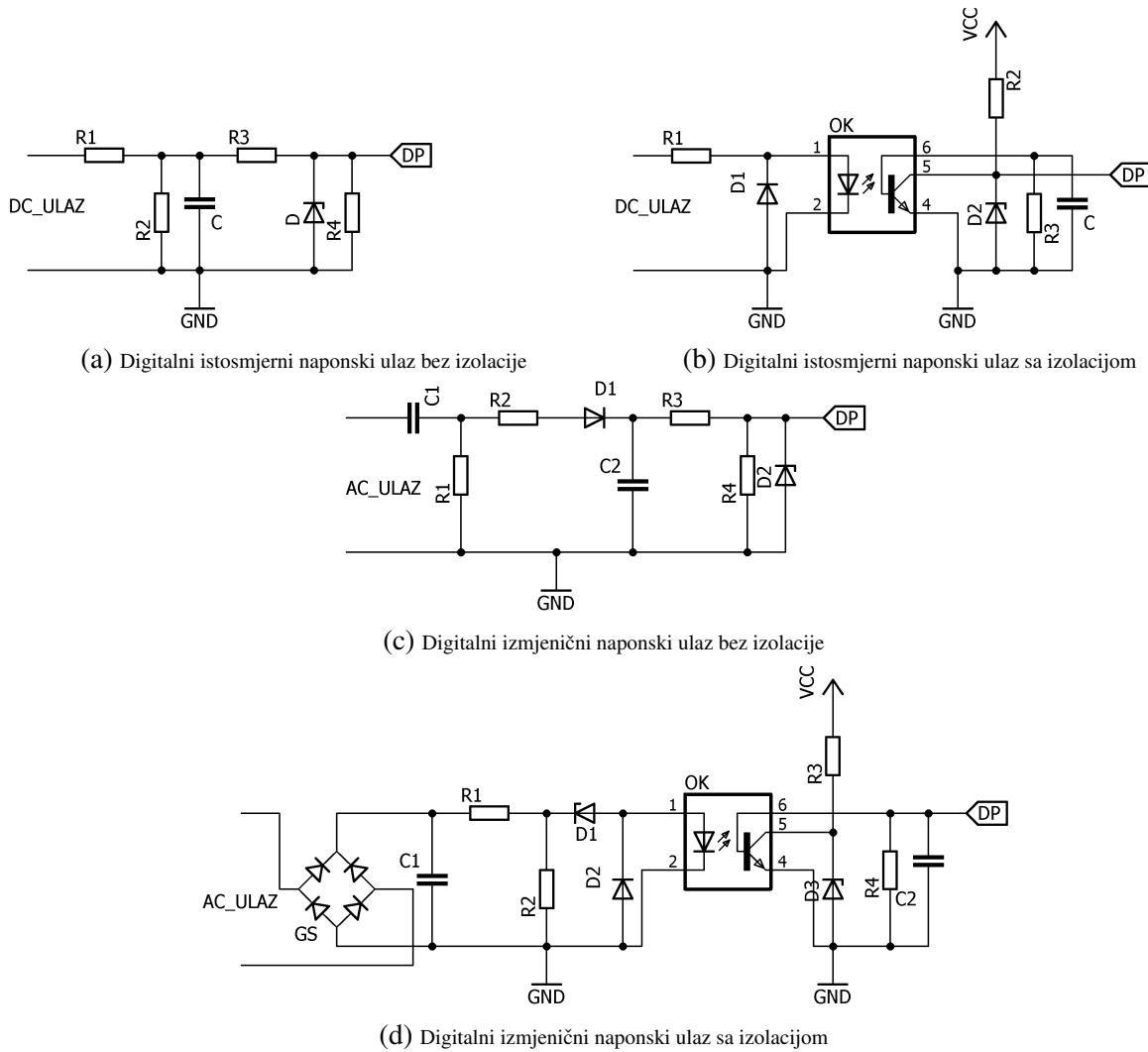
Digitalni ulaz služi za pretvaranje dovedenog električnog signala u signal koji je kompatibilan sa ulazom u mikrokontroler. Ulazni digitalni signal može biti strujni ili naponski, izmjenični ili istosmjerni. U zavisnosti od ulaznog signala postoje različiti elektronički krugovi, kojima se taj signal prevodi u signal pogodan za mikrokontroler. Mikrokontroler ATmega328P baziran je na CMOS tehnologiji i posjeduje digitalne naponske ulaze nivoa 0-5V.

Bitne karakteristike digitalnog ulaza su ulazna otpornost, te brzina reagovanja. Također bitna karakteristika su i minimalne vrijednosti logičke jedinice, te maksimalne vrijednosti logičke nule. Osim ovih parametara potrebno je postići zaštitu mikrokontrolera od eventualnih prenapona i prekostruja na ulaznoj strani.

Shodno tome postoji niz različitih shema spajanja, koje su prikazane na slici 2.12, za naponski digitalni signal. Na slici je prikazana shema prilagođenja ulaza ukoliko se radi o DC signalu (slika 2.12a), pri čemu ne postoji galvansko razdvajanje prema pinu mikrokontrolera. Galvanskim razdvajanjem se postiže zaštita mikrokontrolera od spoljašnjih utjecaja (slika 2.12b).

Na shemi sa slike 2.12a, otpornici R1, R2 i kondenzator C1 služe kao NF filter kako bi se odstranio eventualni utjecaj viših harmonika ulaznog signala, koji mogu dovesti do pogreške u očitanju digitalnog signala. Otpornici R3 i R4, zajedno sa otpornicima R1 i R2 služe sa snižavanje ulaznog istosmjernog napona na napon pogodan za mikrokontroler (5 V). Zener dioda D sprečava pojavu inverznog napona na ulazu u mikrokontroler, te štiti kontroler od pojave napona većih vrijednosti od 5 V. Mnogo efikasnija zaštita mikrokontrolera prikazana je na shemi 2.12b, gdje je izvršeno galvansko razdvajanje ulaza u sistem i ulaza u mikrokontroler pomoću optokaplera. Otpornik R3 i kondenzator C služe za stabilizaciju rada optokaplera.

Na slikama 2.12c i 2.12d prikazane su sheme spajanja naizmjeničnog naponskog digitalnog ulaza bez i sa galvanskom izolacijom. Kondenzatori i otpornici na shemi 2.12c služe za filtriranje i snižavanje napona, dok dioda D1 služi za ispravljanje izmjeničnog napona. Na 2.12d, umjesto poluvalnog ispravljanja korišten je Grecov spoj dioda. Zener dioda D1 služi za sniženje napona i ograničenje ulazne struje optokaplera. Uloga ostalih komponenti je ista kao na prethodnim shemama.

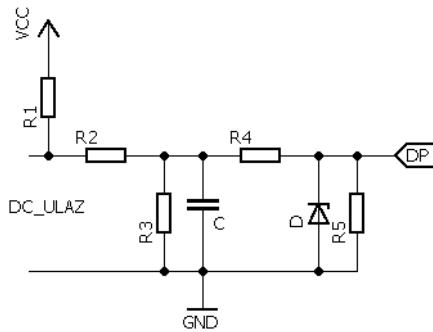


Slika 2.12: Prilagođenje naponskih digitalnih ulaza za mikrokontroler

U uvodnom poglavlju navedeno je da industrijski senzori mogu imati NPN i/ili PNP izlaze. Prema tome potrebno je ostvariti mogućnost očitanja za oba tipa. Prethodno navedeni istosmjerni ulazi su NPN tipa. Digitalni ulaz PNP tipa može se dobiti dodavanjem otpornika prema napajajuću, što je prikazano na slici 2.13. Dodavanje otpornika može se primijeniti i na shemu 2.12b.

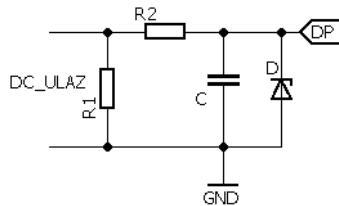
Osim naponskih ulaza potrebno je prikazati prilagođenje strujnih digitalnih ulaza, koji se često koriste u praksi zbog efikasnijeg prenosa signala na veće udaljenosti. Kako mikrokontroler ATmega328p posjeduje samo naponske ulaze, potrebno je strujni signal pretvoriti u naponski.

Jedna od shema koja omogućava pretvaranje naponskog u strujni signal prikazana je na slici



Slika 2.13: Prilagođenje naponskih digitalnih ulaza PNP tipa za mikrokontroler

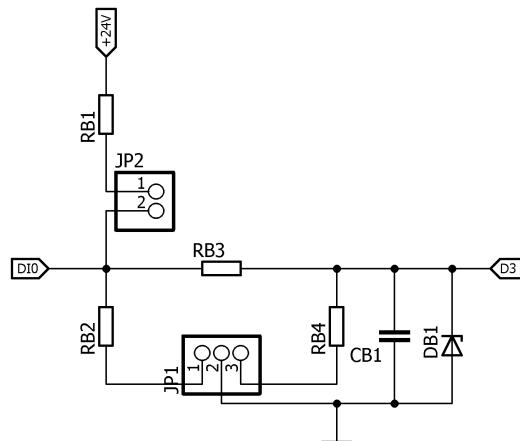
2.14. Pretvaranje se vrši pomoću otpornika spojenog paralelno na ulaz. Proticanjem struje kroz ovaj otpor stvara se pad napona na njemu koji mikrokontroler mjeri. Također, razlikuju se ulazi sa i bez galvanske izolacije.



Slika 2.14: Prilagođenje strujnog digitalnog ulaza na mikrokontroler

Na osnovu prethodnih shema formirane su sheme digitalnih ulaza korištene za realizaciju modula. Sheme sklopova za prilagođenje prikazane su na slikama 2.15 i 2.16.

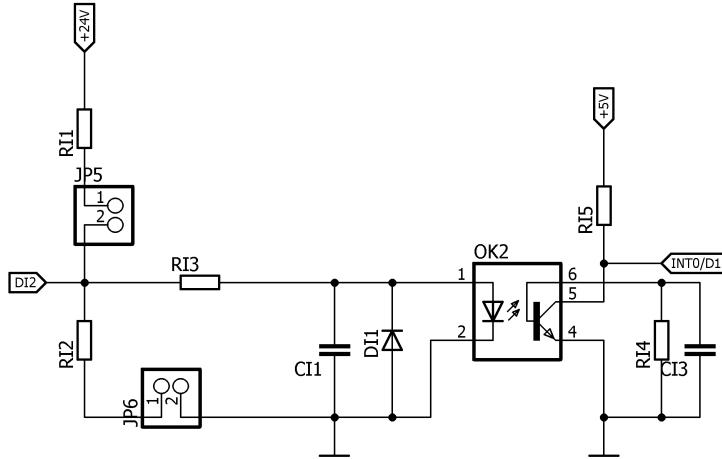
Sheme su identične kao gore navedene s tim što su NPN, PNP naponski ulaz i strujni ulaz spojeni zajedno. Korisnik postavljanjem kratkospojnika na odgovarajuća mjesta odabire konfiguraciju ulaza, koju želi da koristi.



Slika 2.15: Finalna shema digitalnog ulaza bez izolacije

Kratkim spajanjem pinova 2 i 3, headera JP1 (slika 2.15), digitalni ulaz predstavlja istosmjerni naponski ulaz NPN tipa. Ukoliko se kratko spoje i pinovi headera JP2 dobija se istosmjerni

naponski ulaz PNP tipa. Kratkim spajanjem pinova 1 i 2, headera JP1 formira se istosmjerni strujni ulaz.



Slika 2.16: Finalna shema digitalnog ulaza sa izolacijom

Mogućnosti konfiguracije digitalnog ulaza sa izolacijom (slika 2.16) date su u nastavku. Odabir naponskog istosmjernog ulaza NPN tipa vrši se nespajanjem headera kratkospojnicima. Za odabir naponskog istosmjernog ulaza PNP tipa potrebno je kratko spojiti pinove headera JP5, dok je za odabir strujnog digitalnog ulaza potrebno kratko spojiti pinove headera JP6.

2.4.2 Analogni ulaz

U svrhu korištenja analognih senzora potreban je sklop kojim se njihove vrijednosti na izlazu očitavaju i dalje unutar sistema procesiraju. Za ovu potrebu koriste se analogni ulazi. Analogni ulaz pretvara električnu analognu veličinu u digitalni signal, pogodan za kompjuter (ili mikrokontroler).

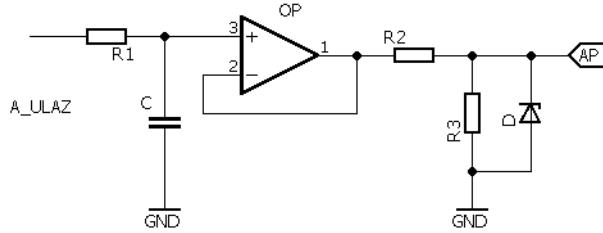
Mikrokontroler ATmega328p ima ugrađen A/D konvertor sa 6 kanala i rezolucijom od 10 bita. A/D konvertor je uređaj koji pretvara analogni električni signal u digitalnu riječ. Rezolucija konverzije analognog u digitalni signal pokazuje sa kojom preciznošću se signal konverte.

U industriji se koriste različiti analogni senzori, čiji izlazni signali mogu biti strujni ili naponski. Opseg promjene izlazne veličine analognih senzora je također različit od vrste senzora. Ovi opsezi su standardizovani kao što je navedeno u uvodu. Kako korišteni mikrokontroler ima naponski A/D konvertor maksimalnog opsega 0-5 V, potrebno je izvršiti prilagođenje ulaznog signala na ovaj opseg.

Bitne karakteristike analognih ulaza su ulazna otpornost, rezolucija i brzina konverzije signala. Pored toga, potrebno je zaštитiti mikrokontroler od spoljašnjih utjecaja.

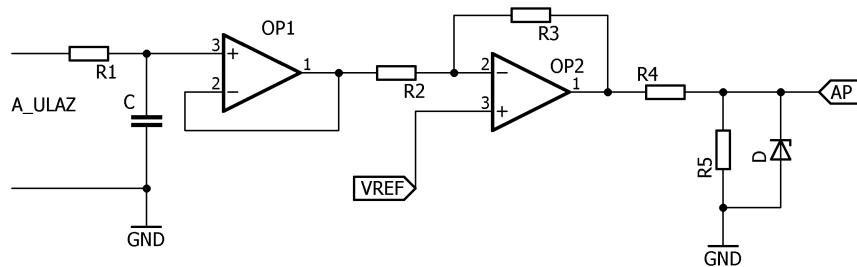
Jedna od mogućih shema prilagođenja analognog ulaza, pozitivnog napona (najčešće od 0 do 10 V) prikazana je na slici 2.17. Otpornik R1 i kondenzator C služe za filtriranje ulaznog napona, kako bi očitanje vrijednosti bilo preciznije. Operaciono pojačalo OP spojeno je kao naponsko sljedilo, koje impedantno razdvaja kontroler od ostatka sistema. Otpornici R2 i R3 spojeni kao

naponski djeljitelj smanjuju napon sa opsega 0-10 V na opseg 0-5 V pogodan za kontroler. Zener dioda D štiti kontroler od inverznog napona i direktnog napona većih vrijednosti.



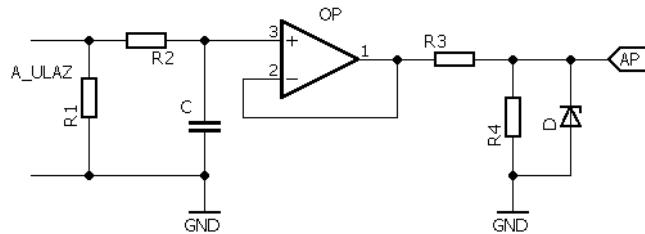
Slika 2.17: Shema prilagođenja analognog unipolarnog napona za mikrokontroler

Osim analognih signala pozitivne vrijednosti u industrijskim primjenama, potrebno je mjeriti bipolarne analogne napone. Kako bi se bipolarni napon prilagodio za mikrokontroler potrebno je pretvoriti ovaj napon u unipolarni. Jedna od mogućnosti je korištenje sumatora, što je prikazano na slici. Operaciono pojačalo OP2 zajedno sa otporima R2 i R3, te referentnim naponom na neinvertirajućem ulazu, pomjera ulazni napon, te ga pretvara u unipolarni. Uloga ostalih komponenti je ranije opisana.



Slika 2.18: Shema prilagođenja analognog bipolarnog napona za mikrokontroler

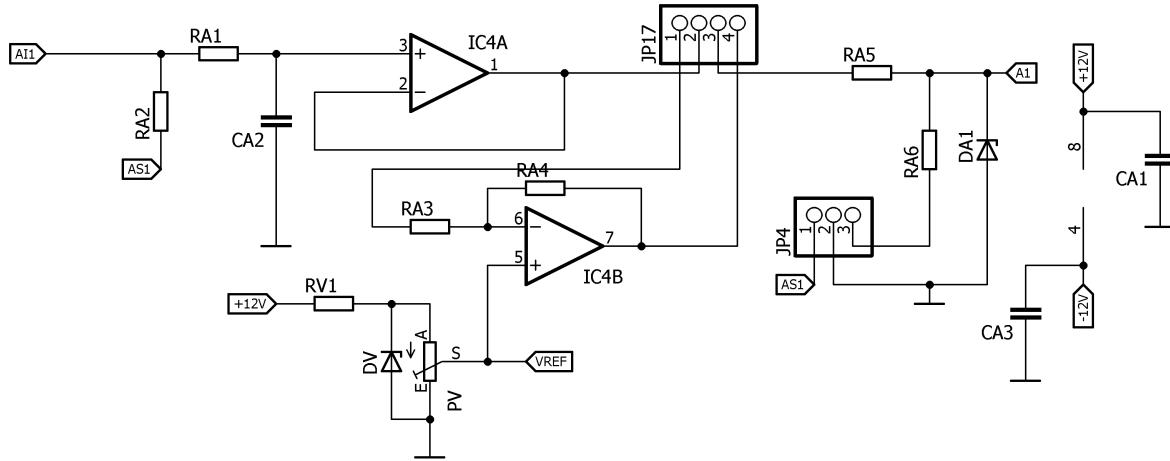
Često se javlja potreba za mjeranjem analognih strujnih signala, iz istih razloga kao i digitalnih strujnih signala. Prilagođenje strujnog analognog signala moguće je vršiti dodavanjem paralelnog otpora na ulaz sheme sa slike 2.17. Shema spajanja strujnog analognog ulaza prikazana je na slici 2.19. Ulagana struja stvara pad napona na otporu R1. Kontroler mjeri ovaj napon, čime je izvršena konverzija strujnog u naponski signal.



Slika 2.19: Shema prilagođenja analognog strujnog signala za mikrokontroler

Na osnovu prikazanih mogućih prilagođenja analognog signala za mikrokontroler, pri realizaciji modula formirana je finalna shema. Shema je prikazana na slici 2.20.

Primjećuje se da je shema analognog ulaza (slika 2.20) fuzija prethodno navedenih shema. Postavkom kratkospojnika odabire se konfiguracija ulaza. Kratkim spajanjem pinova 2 i 3, headera JP17, te pinova 2 i 3, headera JP4, formira se shema za prilagođenje unipolarnog analognog ulaza. Kratkim spajanjem pinova 2 i 3, headera JP17 i pinova 1 i 2, headera JP4, dobija se



Slika 2.20: Shema prilagođenja analognog ulaza korištena u realizaciji modula

shema prilagođenja strujnog analognog signala za mikrokontroler. Treća konfiguracija dobija se kratkim spajanjem pinova 1 i 2, te pinova 3 i 4, headera JP17, te pinova 2 i 3, headera JP4. Ova shema služi za mjerjenje bipolarnog napona. Operaciono pojačalo IC4B zajedno sa otpornicima RA3 i RA4, te referentnim naponom čini sumator. Referentni napon dobija se preko naponskog djeljitelja (potenciometar PV) i zener diode DV.

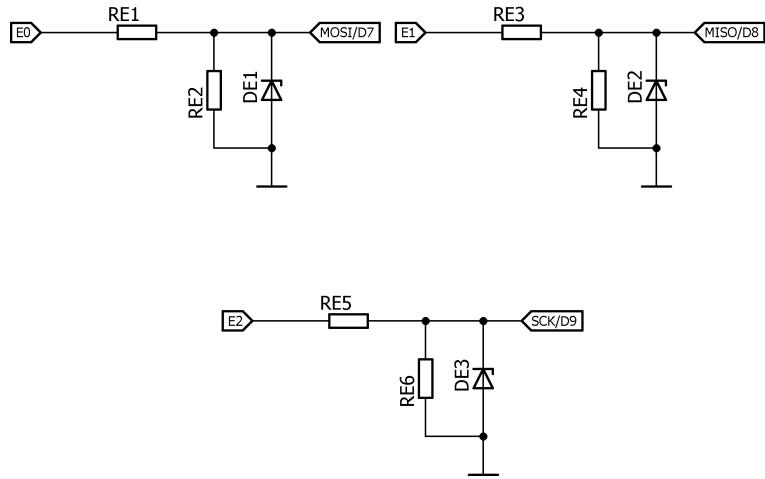
2.4.3 Enkoderski ulaz

Rad enkodera ukratko je pojašnjen u uvodnom poglavlju. Enkoderski ulaz sastoji se od tri digitalna ulaza, na osnovu kojih se računaju brzina i pozicija uređaja. Osnovne karakteristike enkodera su broj impulsa po obrtaju i naponski nivo izlaznog signala. Zaključuje se da su karakteristike enkoderskog ulaza frekvencija ulaznog signala i naponski nivo.

Enkoderski ulaz moguće je realizovati spajanjem izlaza enkodera preko naponskog djeljitelja na ulaz mikrokontrolera, sa odgovarajućom zaštitom. Ukoliko se koristi ovakav način realizacije potrebno je softverski računati brzinu i poziciju. Ograničenje ove realizacije zasniva se na maksimalnoj frekvenciji ulaznog signala, koju mikrokontroler može očitati. Može se poboljšati korištenjem mikrokontrolera sa većom frekvencijom očitanja ulaza.

Drugi način realizacije je korištenje hardverskih brojača, koji rade na velikim frekvencijama (reda 1 MHz). Integralno kolo CD74HC4017 je primjer takvog brojača [17]. Maksimalna frekvencija ovog brojača zavisi od napona napajanja i može biti do 23MHz. Ovaj brojač je 10-bitni sa nekodiranim izlazima, što znači da je potrebno 10 izlaza spojiti na mikrokontroler (direktno ili preko multipleksera, shift registra i sličnih komponenti).

Za realizaciju modula u ovom radu, korištena je prva, najjednostavnija shema enkoderskog ulaza. Shema spajanja prikazana je na slici 2.21.



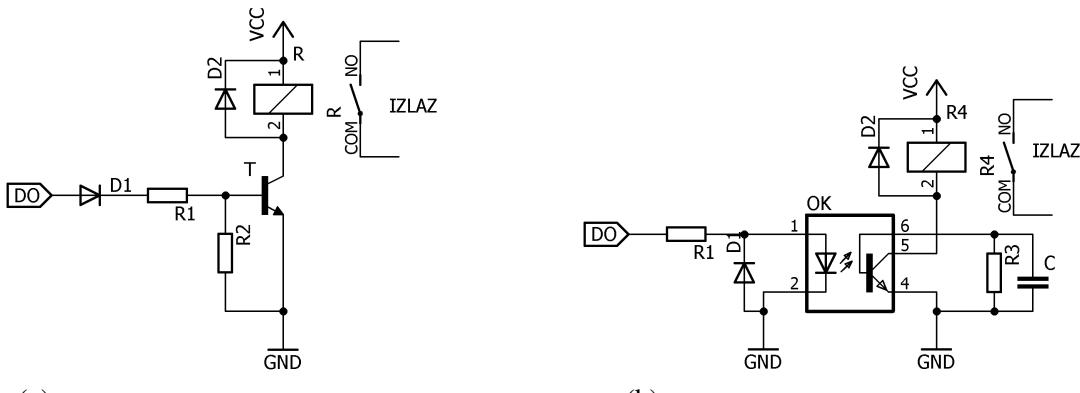
Slika 2.21: Shema prilagođenja enkoderskog ulaza korištena u realizaciji modula

2.5 Izlazni portovi

2.5.1 Digitalni izlaz

Digitalni izlaz služi za pretvaranje signala koji generiše logički dio modula (mikrokontroler) u signal kojim se upravlja nekim uređajem u sistemu. Najčešći uređaji kojima se upravlja su motori, rasvjetna tijela, različite vrste pumpi i dr. Digitalni izlazi mogu biti relejni ili tranzistorski, koji mogu biti izvor struje (eng. *source*, PNP) ili ponor struje (eng. *sink*, NPN).

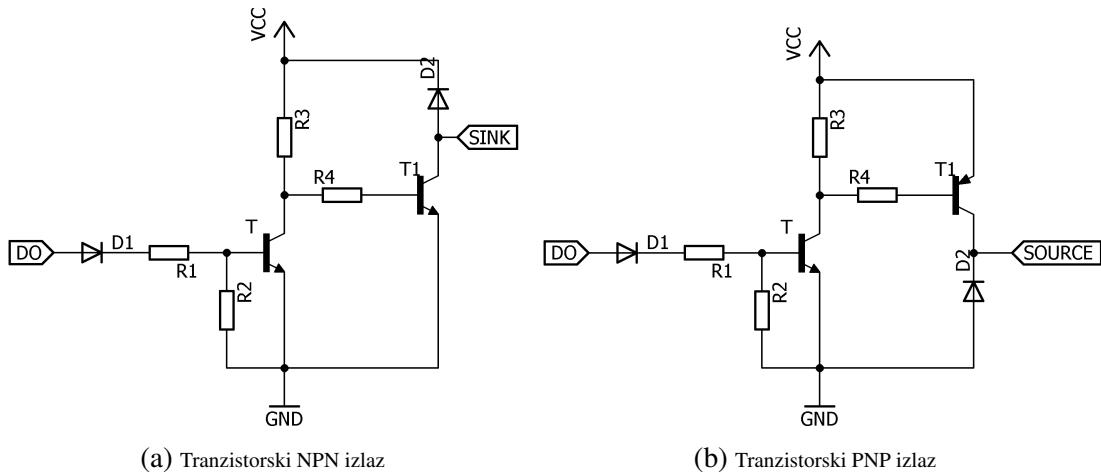
Bitne karakteristike digitalnog izlaza su izlazna otpornost, maksimalna izlazna struja, maksimalni napon na izlazu, te brzina promjene stanja. Kao i kod digitalnog ulaza, i u ovom slučaju je potrebno zaštiti mikrokontroler od spoljnog utjecaja.



Slika 2.22: Shema spajanja relejnih izlaza na mikrokontroler

Na slici 2.22 prikazane su sheme spajanja relejnih izlaza na mikrokontroler. Na shemi 2.22a prikazano je upravljanje relejem pomoću NPN tranzistora. Dioda D1 služi za zaštitu izlaznog pina mikrokontrolera. Otpornik R1 polarizuje tranzistor, tako da radi u prekidačkom režimu. Otpornik R2 se koristi za brže isključenje tranzistora. Dioda D2 štiti tranzistor od prenapona kad se on isključuje, zbog induktivnog karaktera primara releja. Na shemi 2.22b prikazana je ista funkcionalnost kao i na prethodnoj shemi, samo što se za upravljanje relejem koristi optokapler. Na ovaj način se postiže bolja zaštita mikrokontrolera, nego u prvom slučaju.

Kao što je već pomenuto osim relejnih, postoje i tranzistorski izlazi. Oni su prikazani na slici 2.23. Na shemi 2.23a prikazan je *sink* izlaz, što znači da struja teče od potrošača prema izlazu. U slučaju logičke jedinice izlaz je prespojen na masu preko tranzistora koji je u zasićenju, dok je u slučaju logičke nule izlaz odspojen (stanje visoke impedanse). Na shemi 2.23b prikazan je *source* izlaz, koji je suprotan od *sink* izlaza, tj. struja teče iz izlaza prema potrošaču. Upravljanje izlaznim tranzistorima T1 vrši se preko tranzistora T. Tranzistori T1 su većih snaga i maksimalnih struja, dok je tranzistor T manjih snaga. Shema za polarizaciju tranzistora T je ista kao i kod tranzistora kod relejnog izlaza. Otpornici R3 i R4 polarizuju tranzistore T1, tako da rade u prekidačkom režimu. Diode D2 štite tranzistore T1 od prenapona, ukoliko je izlazno opterećenje induktivnog karaktera.



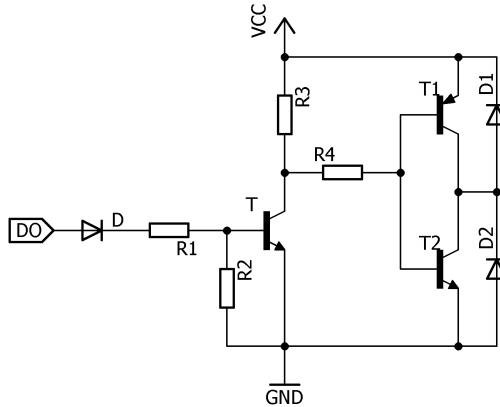
(a) Tranzistorski NPN izlaz

(b) Tranzistorski PNP izlaz

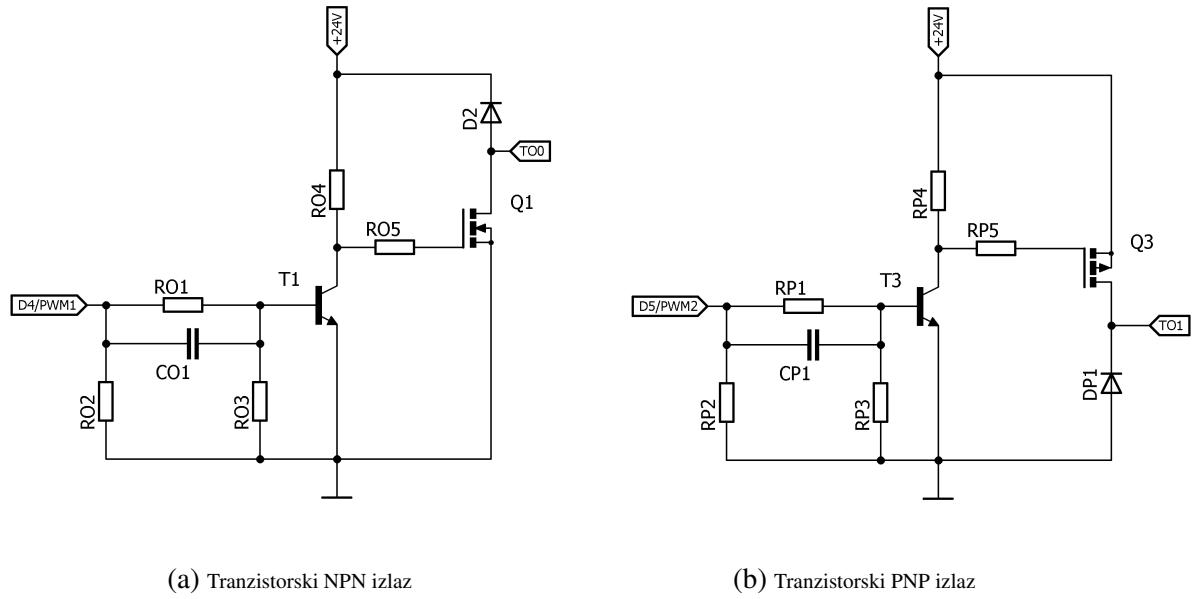
Slika 2.23: Shema spajanja tranzistorskih izlaza na mikrokontroler

Ukoliko se kao izlazni tranzistor koristi MOSFET, shema spajanja je potpuno ista, s tim što otpornik R4 može biti kratko spojen.

Osim navedenih izlaza postoji i izlaz koji daje naponski izlazni signal. On je mješavina *sink* i *source* tranzistorskih izlaza, koji rade komplementarno. Shema spajanja je prikazana na slici 2.24. Tranzistori T1 i T2 nikad ne vode istovremeno. Na izlazu imamo približno napon VCC za slučaj logičke jedinice, te približno 0 za slučaj logičke nule. Uloge dioda i otpornika su ranije navedene.



Slika 2.24: Shema spajanja naponskog digitalnog izlaza sa komplementarnim tranzistorima



Slika 2.25: Shema spajanja tranzistorskih izlaza na mikrokontroler korištena u realizaciji modula

Osim naponskih postoje i strujni izlazi. Shema spajanja je dosta slična kao i kod tranzistorских izlaza, samo u ovom slučaju je potrebno da tranzistori rade u linearnom režimu i struju održavaju konstantnom za slučaj logičke jedinice. Zbog toga se dodaje emiterski otpornik. Također se razlikuju *sink* i *source* strujni izlaz.

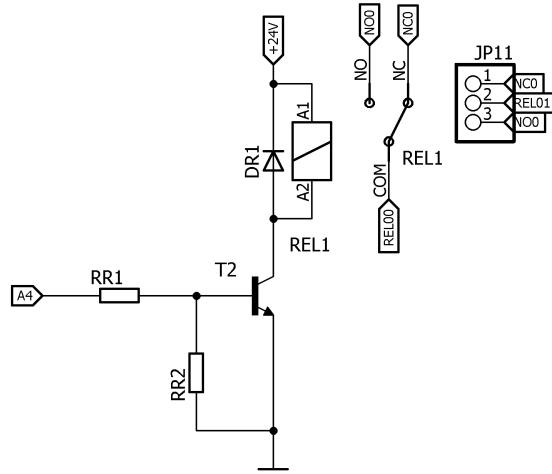
Na osnovu navedenih mogućnosti formirane su sheme korištene u realizaciji modula. Sheme su prikazane na slikama 2.25 i 2.26.

Sheme tranzistorskih izlaza (slika 2.25) su dosta slične navedenim. Razlika je u načinu polarizacije upravljačkog tranzistora T1 (T3). U realizaciji su dodani otpornik i kondenzator. Otpornici i kondenzator u bazi upravljačkog tranzistora služe za ubrzanje rada tranzistora. Kondenzator CO1 (CP1) i otpornik RO3 (RP3) spojeni su kao diferencijator. Pri promjeni signala na izlazu iz kontrolera dolazi do naglih skokova napona u bazi upravljačkog tranzistora, a samim tim i brže promjene stanja. Otpornik RO1 (RP1) služi za postavljanje radne tačke tranzistora, kako bi radio u prekidačkom režimu, te pražnjenje kondenzatora ukoliko dođe do odspajanja ulaza u prikazano kolo.

Shema spajanja relejnog izlaza (slika 2.26) je identična kao gore navedena. Međutim u realizaciji je korišten relej sa dva izlaza (NC i NO), pa je shodno tome dodan header, pomoću kojeg se vrši izbor izlaza, koji će se koristiti.

2.5.2 Analogni izlaz

U industrijskim postrojenjima često je potrebno određenim uređajima proslijediti neku referentnu vrijednost električnog signala (napona ili struje), te na taj način upravljati tim uređajima. Za te potrebe koriste se analogni izlazi. Cilj analognih izlaza je konverzija digitalnog u analogni signal pomoću odgovarajućih elektroničkih komponenti. Na izlazu iz mikrokontrolera je digitalni signal, koji je potrebno prilagoditi.



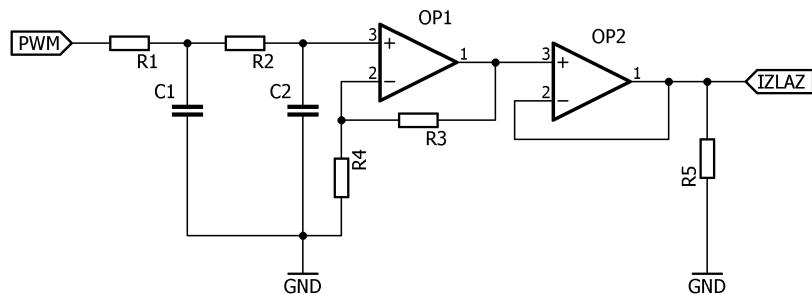
Slika 2.26: Shema spajanja relejnog izlaza korištena u realizaciji modula

Bitne karakteristike analognih izlaza su tačnost, rezolucija, te brzina promjene izlaza. Pored navedenih karakteristika, za naponske izlaze bitna je veličina izlazne struje, te izlazna otpornost.

Postoji više načina konverzije digitalnog (signala na izlazu mikrokontrolera) u analogni signal. Jedan od načina je korištenje integralnih kola za D/A konverziju. Primjer ovakvog integralnog kola je AD5750 [18], koji se upravlja SPI komunikacijom. AD5750 posjeduje naponski i strujni izlazni kanal, rezolucije 16 bita. Naponski nivoi izlaznih signala su standardni.

Drugi način realizacije analognog izlaza je korištenje otporničkih struktura i operacionog pojačala (DAC sa težinskom mrežom i R2R DAC). Za postizanje željenih nivoa signala potrebno je dodatno koristiti pojačavače, sumatore i slične module. Nedostatak ovih pretvarača je korištenje jednog izlaza mikrokontrolera za 1 bit u konverziji, što znači da je za rezoluciju od 16 bita potrebno 16 izlaza mikrokontrolera.

Treći i najjednostavniji način realizacije analognog izlaza je korištenje PWM signala na izlazu mikrokontrolera i NF-filtera. Rad ovog sklopa bazira se na izdvajaju istosmjernog signala iz PWM signala, tj. prigušenje njegovih izmjeničnih komponenti. Na taj način dobija se vrijednost analognog signala proporcionalna *duty cycle*-u PWM signala. Jedan od načina spajanja ove realizacije prikazan je na slici 2.27.



Slika 2.27: Shema spajanja naponskog analognog izlaza

Otpornici i kondenzatori na ulazu predstavljaju NF filter. Operaciono pojačalo OP1 sa dva otpornika (R3, R4) formira pojačavač kojim se prilagođava nivo izlaznog signala. Operaciono

pojačalo na izlazu, spojeno kao naponsko sljedilo, služi za impedantno razdvajanje, te smanjenje izlaznog otpora. Rezolucija ovog izlaza jednaka je rezoluciji PWM signala mikrokontrolera. Nedostatak ove realizacije je istovremeno zadovoljenje brzine promjene izlaza i izlazne greške. Brzina promjene se smanjuje smanjenjem vremenske konstante NF filtera, dok se izlazna greška smanjuje povećanjem vremenske konstante filtera. Karakteristike je moguće poboljšati povećanjem frekvencije PWM-a, što je ograničeno hardverom mikrokontrolera.

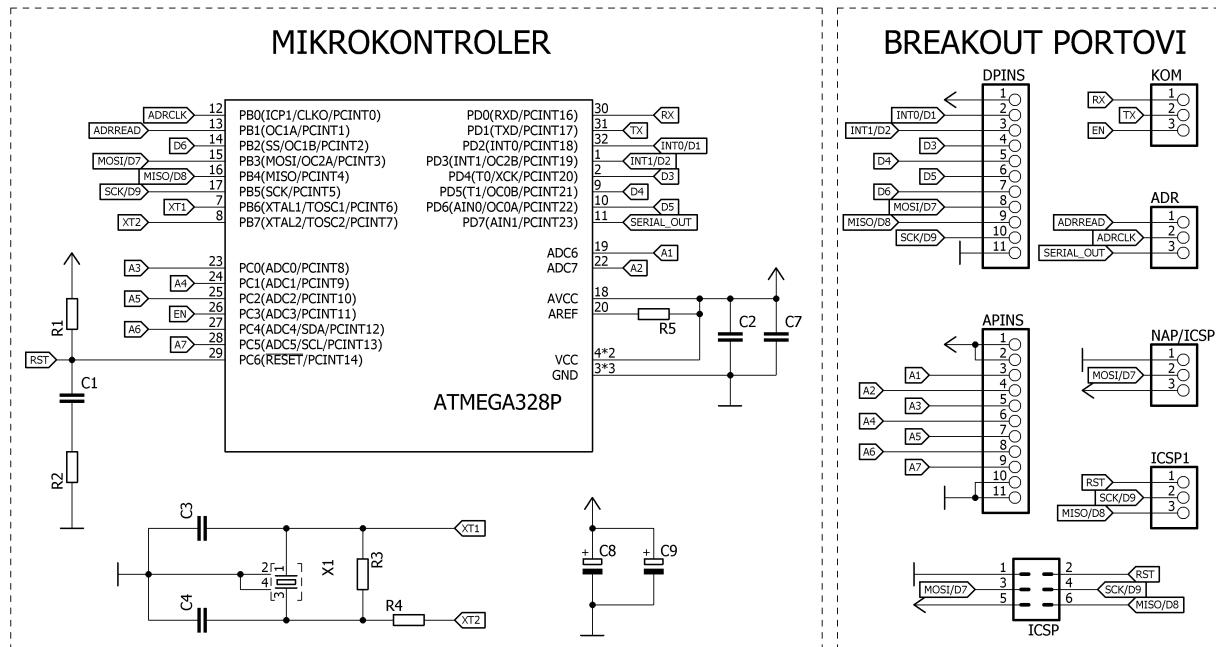
Za realizaciju modula u ovom radu korištena je shema spajanja analognog izlaza kao na slici 2.27

2.6 Dizajn štampane pločice - PCB

Jedan od bitnijih dijelova realizacije hardverskog dijela modula je dizajn štampanih pločica - PCB (eng. *Printed Circuit Board*). U okviru ovog rada dizajn modula je rastavljen na tri dijela. Razlog zbog kojeg je to urađeno je što se vertikalnim povezivanjem manjih pločica postiže veća iskoristivost prostora. Također postoji manja vjerovatnoća da dođe do greške pri projektovanju pločica, te ukoliko se pojavi greška na jednom dijelu, potrebno je samo taj dio ispraviti. Osim navedenog, omogućava se povezivanje sa drugim pločicama, različitog dizajna, ali istih funkcija. Prvi dio modula čini mikrokontroler sa osnovnom konfiguracijom, drugi dio obuhvata komunikaciju i adresiranje uređaja, dok treći obuhvata ulazne i izlazne portove. Na osnovu ova tri dijela projektovane su i štampane pločice. Povezivanje pločica vrši se pomoću headera (izvedenih pinova).

2.6.1 Projektovanje štampane pločice u programskom paketu Eagle

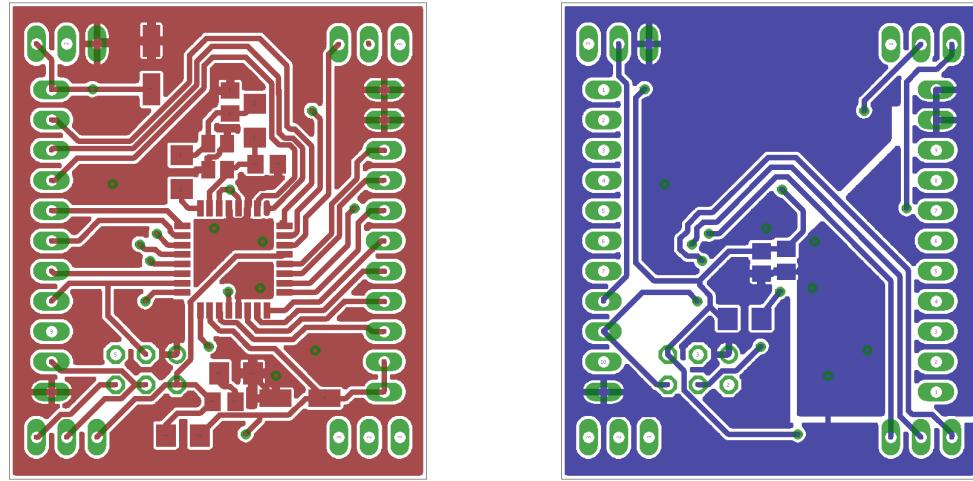
Na slici 2.28 je prikazana shema osnovne konfiguracije mikrokontrolera sa izvedenim ulaznim i izlaznim pinovima, te pinovima za komunikaciju i adresiranje. Dizajn štampane pločice mikrokontrolera je prikazan na slici 2.29.



Slika 2.28: Električna shema spajanja mikrokontrolera sa breakout pinovima

Drugi dio modula sastoji se od dijela za adresiranje, te dijela za komunikaciju. Električna shema spajanja je data na slici 2.30. Štampana pločica prikazana je na slici 2.31. Povezivanje između pločice za adresiranje i komunikaciju i pločice sa kontrolerom vrši se preko 12 headera, raspoređenih u četiri ugla po 3, kao što se vidi na štampanoj pločici.

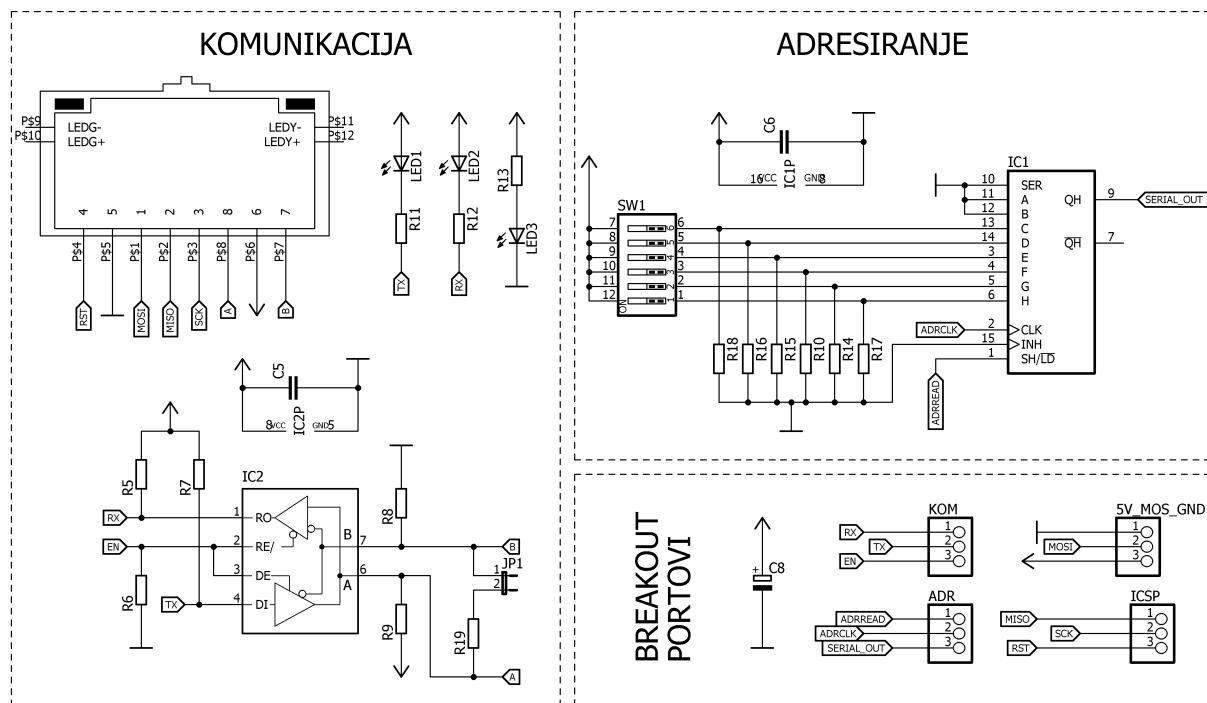
Treći dio modula koji obuhvata ulazne i izlazne portove predstavlja bazu uređaja. Na nju se dodaju prethodno navedeni dijelovi. Električna shema ovog dijela prikazana je na slici 2.32. Štampana pločica prikazana je na slici 2.33.



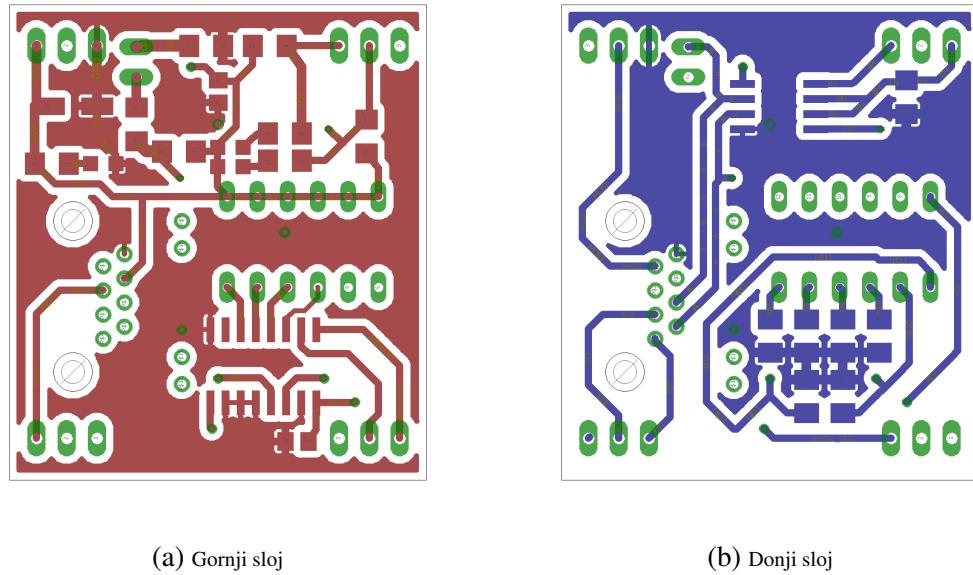
(a) Gornji sloj

(b) Donji sloj

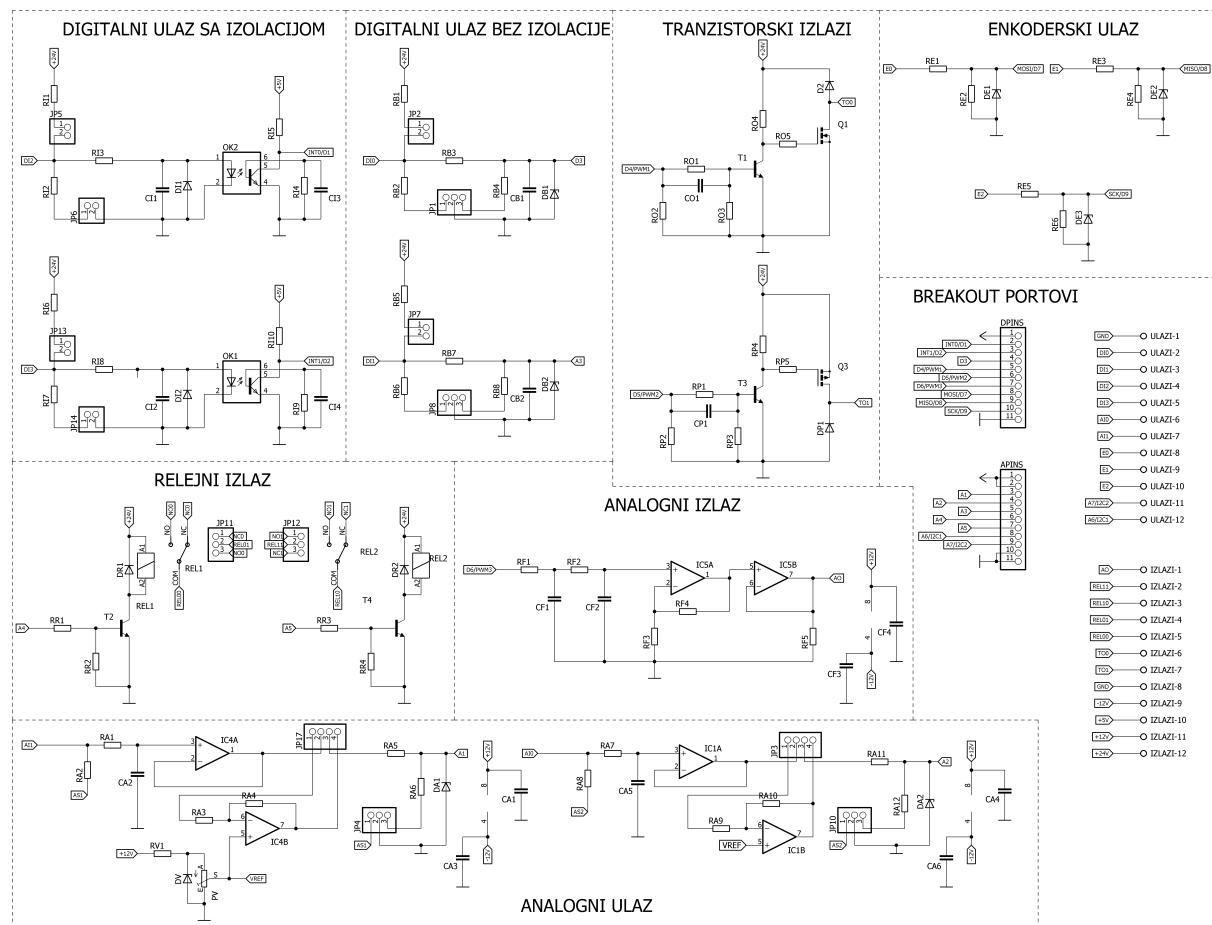
Slika 2.29: Izgled štampane ploče mikrokontrolerskog dijela



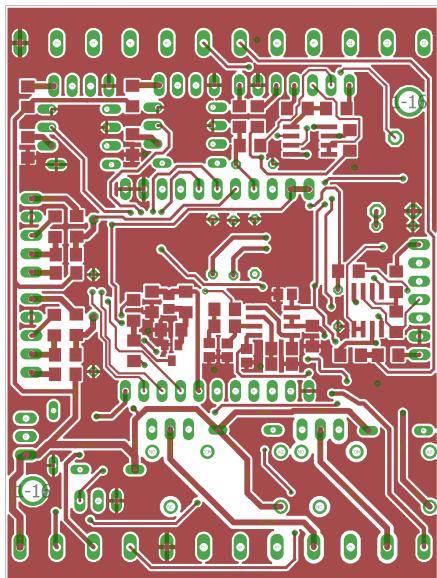
Slika 2.30: Električna shema spajanja dijela za komunikaciju i adresiranje



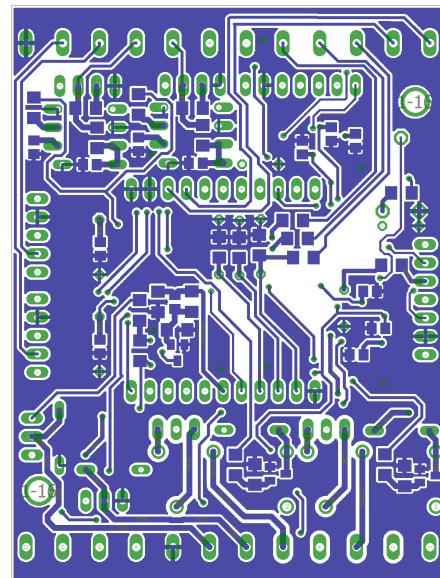
Slika 2.31: Izgled štampane ploče dijela za komunikaciju i adresiranje



Slika 2.32: Električna shema prilagođenja industrijskih ulaza i izlaza na mikrokontroler



(a) Gornji sloj

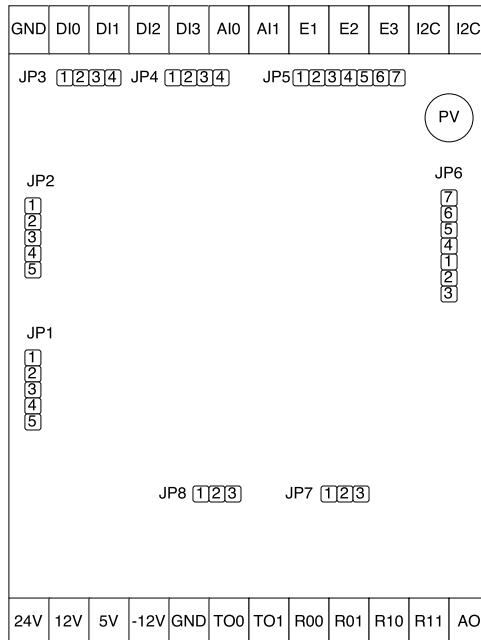


(b) Donji sloj

Slika 2.33: Izgled štampane ploče dijela sa ulazima i izlazima

2.6.2 Raspored ulaza i izlaza na štampanoj pločici

U ovom dijelu rada prikazan je raspored karakterističnih dijelova na štampanoj pločici. Na slici 2.34 su prikazani položaji ulaza i izlaza, te headeri, koji služe za konfiguraciju uređaja, na štampanoj pločici. Također na slici je prikazana pozicija potenciometra PV, koji služi za podešavanje referentnog napona za bipolarne analogne ulaze (objašnjeno u dijelu 2.4.2).



Slika 2.34: Raspored ulaza, izlaza i konfiguracijskih headera

U tabeli 2.1 prikazane su mogućnosti odabira konfiguracije digitalnih ulaza bez izolacije (DI0, DI1), odnosno postavka headera JP1 i JP2.

Funkcija	Postavka kratkospojnika
Naponski NPN ulaz	1-2
Naponski PNP ulaz	1-2, 4-5
Strujni ulaz	2-3

Tabela 2.1: Mogućnosti konfiguracije digitalnih ulaza bez izolacije

U tabeli 2.2 prikazane su mogućnosti odabira konfiguracije digitalnih ulaza sa izolacijom (DI2, DI3), odnosno postavka headera JP3 i JP4.

Funkcija	Postavka kratkospojnika
Naponski NPN ulaz	-
Naponski PNP ulaz	1-2
Strujni ulaz	3-4

Tabela 2.2: Mogućnosti konfiguracije digitalnih ulaza sa izolacijom

U tabeli 2.3 prikazane su mogućnosti odabira konfiguracije analognih ulaza, odnosno postavka headera JP5 i JP6.

Funkcija	Postavka kratkospojnika
Naponski analogni ulaz 0...10 V	1-2, 5-6
Naponski analogni ulaz -10...10 V	1-2, 4-5, 6-7
Strujni analogni ulaz 0...20 mA	2-3, 5-6

Tabela 2.3: Mogućnosti konfiguracije analognih ulaza

U tabeli 2.4 prikazane su mogućnosti odabira konfiguracije relejnih izlaza, odnosno postavka headera JP7 i JP8.

Funkcija	Postavka kratkospojnika
Mirni kontakt (NC)	2-3
Radni kontakt (NO)	1-2

Tabela 2.4: Mogućnosti konfiguracije relejnih izlaza

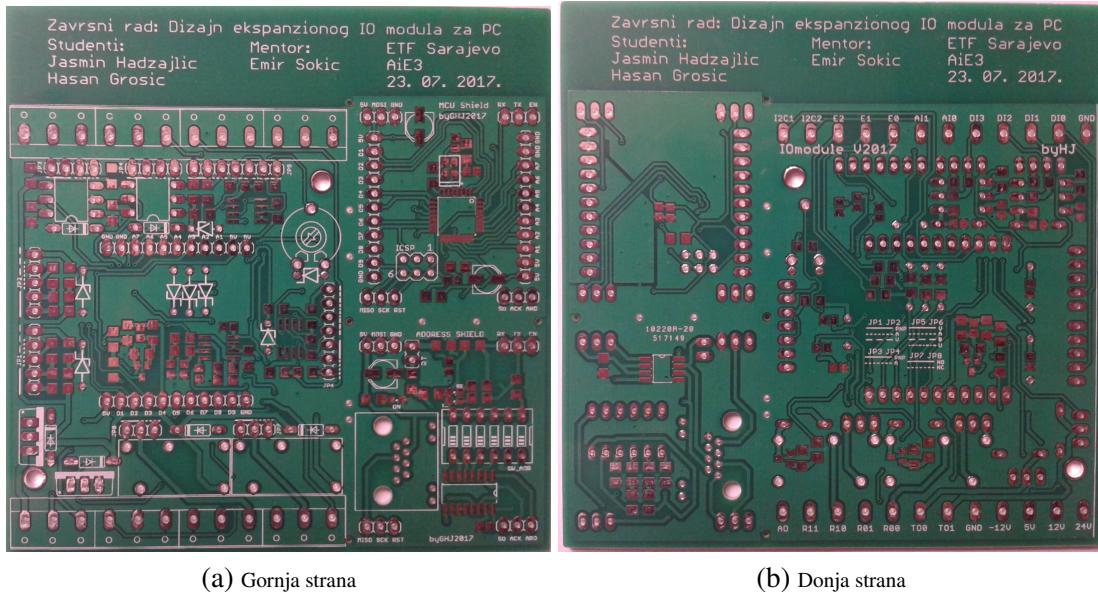
U tabeli 2.5 su prikazane oznake, odnosno adrese ulaza i izlaza, zajedno sa ulogama i znakovima u poruci, koja se šalje prilikom komunikacije.

Oznaka	Uloga	Poruka
DI0	Digitalni ulaz bez izolacije	'a'
DI1	Digitalni ulaz bez izolacije	'b'
DI2	Digitalni ulaz sa izolacijom	'c'
DI3	Digitalni ulaz sa izolacijom	'd'
AI0	Analogni ulaz	'e'
AI1	Analogni ulaz	'f'
TO0	Tranzistorski izlaz	'p'
TO1	Tranzistorski izlaz	'q'
R00	Relejni izlaz	'r'
R01		
R10	Relejni izlaz	's'
R11		
AO	Analogni izlaz	't'

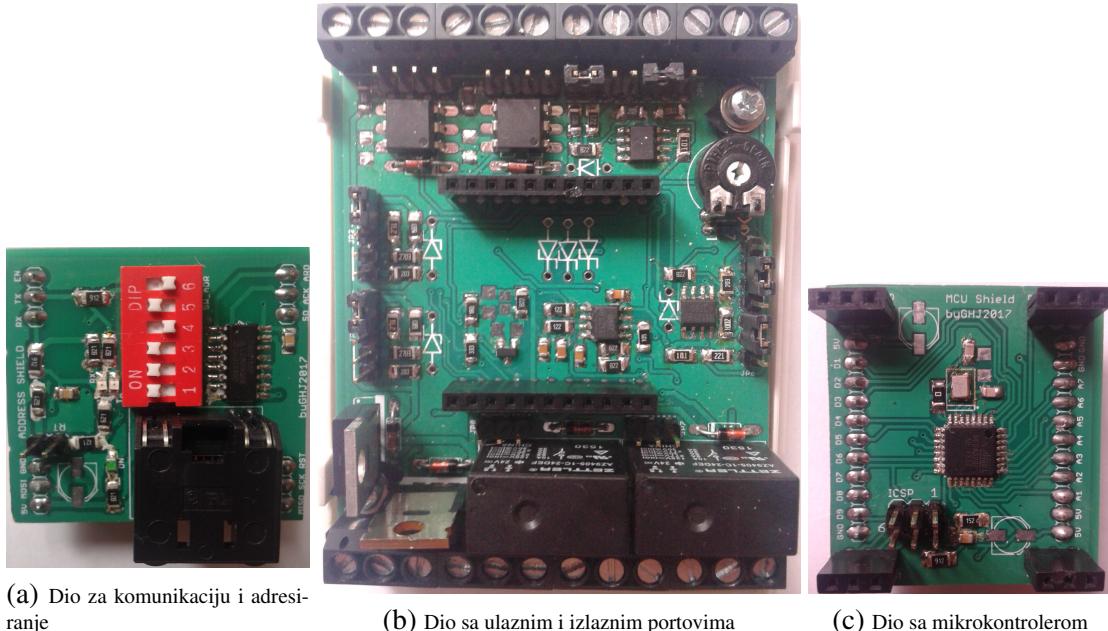
Tabela 2.5: Oznake ulaza i izlaza IO modula

2.6.3 Fizička realizacija štampanih pločica

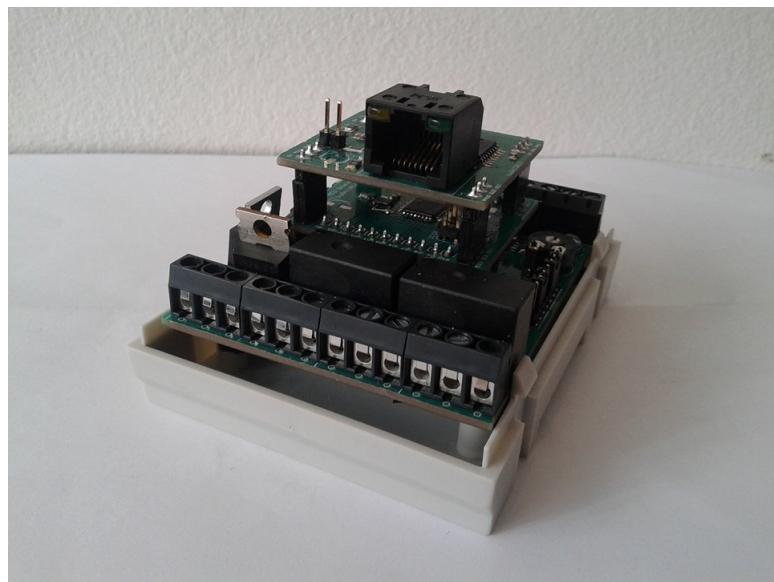
Na slici 2.35 je prikazan izgled štampane pločice nakon izrade. Štampane pločice svih cjelina su objedinjene u jednu. Prije lemljenja komponenti pločice su rastavljenje. Izgled štampanih pločica nakon lemljenja prikazan je na slici 2.36. Na slikama 2.37 i 2.38 prikazan je izgled IO modula, nakon povezivanje sva tri dijela u jednu cjelinu, prije i poslije pakovanja.



Slika 2.35: Izgled štampanih pločica IO modula, nakon izrade



Slika 2.36: Izgled štampanih pločica IO modula, nakon lemljenja



Slika 2.37: Izgled IO modula bez kućišta

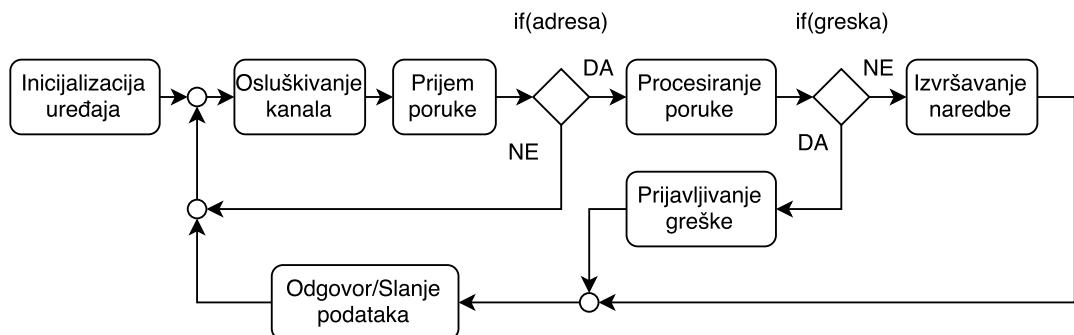


Slika 2.38: Izgled IO modula sa kućištem

Poglavlje 3

Softverska komponenta IO modula

U ovom poglavlju bit će opisana softverska realizacija sistema, što se odnosi na programiranje mikrokontrolera. Cilj softverskog dijela je omogućiti da IO modul na osnovu naredbe koju pošalje računar, izvrši odgovarajuću funkciju, te kao rezultat vrati povratnu poruku. Dijagram toka je prikazan na slici 3.1.



Slika 3.1: Dijagram toka softvera mikrokontrolera

Dijagram toka pokazuje na koji način funkcioniše uređaj. Pri pokretanju uređaja, njegovi dijelovi se inicijaliziraju. Nakon toga uređaj ulazi u petlju, u kojoj osluškuje kanale (očitava stanja ulaza). Kada računar pošalje naredbu, softver izlazi iz petlje, izvršava odgovarajuću naredbu, a zatim šalje odgovor nazad prema računaru. Ukoliko se detektuje greška u bilo kojem stanju, uređaj javlja računaru da je došlo do pogreške.

3.1 Softverske funkcionalnosti

U ovom dijelu rada bit će navedene funkcionalnosti uređaja, koje su softverski implementirane. Na postojeći hardver moguće je implementirani dodatne funkcionalnosti, u zavisnosti od potencijalne primjene sistema.

Implementirane funkcionalnosti su sljedeće:

- Očitanje zadanih ulaza na zahtjev,
- Postavljanje zadanih izlaza na zahtjev,
- Očitanje zadanih ulaza n puta, periodično periodom T,

- Očitanje ulaza na promjenu stanja,
- Postavljanje jednog izlaza,
- Očitanje jednog ulaza.

Očitanje zadanih ulaza na zahtjev sastoji se u slanju konfiguracijske poruke kojom se oda-biru ulazi čija se vrijednost očitava, te poruci za trigerovanje ulaza, kojom se očitane vrijednosti šalju nazad računaru.

Postavljanje zadanih izlaza na zahtjev je slično očitanju zadanih ulaza na zahtjev. Sastoji se od konfiguracijske poruke, te poruke za trigerovanje, tj. postavljanje izlaza na zadane vrijednosti.

Očitanje zadanih ulaza n puta, periodično periodom T je funkcionalnost kojom se vrijednosti konfigurisanih ulaza, očitavaju periodično zadanim periodom T , a zatim šalju računaru. Ovaj postupak se ponavlja n puta, koji računar postavlja.

Očitanje ulaza na promjenu stanja, ili tzv. prekidni mod je funkcionalnost kojom se prati promjena stanja konfigurisanih ulaza. Ukoliko dođe do promjene stanja bilo kojeg konfigurisanih ulaza uređaj šalje poruku prema računaru.

Postavljanje jednog izlaza omogućava korisniku da direktno postavi izlaz, bez potrebe za konfiguracijom i trigerovanjem.

Očitanje jednog ulaza omogućava očitanje jednog ulaza, bez konfiguracije i trigera.

3.2 Razvijeni komunikacijski protokol

Kod komunikacije je bitno definisati format poruke, tj. paketa, koji se šalje. Očito je da svaki paket treba da sadrži adresu uređaja kojem se šalje. Osim toga paket treba da sadrži podatak o početku i kraju poruke, te samu informaciju koju korisnik šalje. Pored navedenog, ponekad je dobro slati i pomoćne podatke, kojima smanjujemo mogućnost greške (kao što je bit pariteta).

3.2.1 Format poruke

Format poruke je isti u oba smjera, od računara ka uređaju i obratno. Poruka se sastoji od dijela za start, funkciju, parametre funkcije, te dijela za kraj. Zbog različitosti funkcija i parametara, dužina poruke nije konstantna. Format poruke prikazan je u tabeli 3.1.

Start (2 bajta)	Adresa (1 bajt)	Naredba (1 bajt)	Parametri (x bajta)	Kraj (3 bajta)
'#' '#'	0-255	'A' - 'J'	-	CR CR NL

Tabela 3.1: Format poruke korišten za realizaciju komunikacije uređaja

Kao što je prikazano, za start se koriste dva znaka '#', jer se rijetko pojavljuju u poruci kao parametri. Također za kraj poruke koriste se tri bajta, jer postoji mogućnost da se u parametrima pošalju dva znaka jedan iza drugog, koji mogu označiti kraj poruke, te će na taj način doći do pogreške u čitanju.

3.2.2 Implementirane poruke

U narednom dijelu navedene su implementirane poruke na uređaju, zajedno sa njihovim odgovorima.

Inicijalizacija

Poruka inicijalizacije služi za označavanje koji ulazi, odnosno izlazi će biti korišteni tokom rada. Naredba za inicijalizaciju je znak 'A'. Kao parametri poruke šalju se bajti koji označavaju ulaze i izlaze, dok se kod tranzistorских izlaza dodatno šalje odabir režima rada (digitalni ili PWM izlaz). Oznake ulaza i izlaza date su u tabeli 3.2. Odabir PWM tranzistorског izlaza vrši se slanjem bajta '1' uz oznaku tranzistorског izlaza, dok se slanjem bajta '0' odabire digitalni tranzistorски izlaz.

Oznaka	Uloga	Poruka
DI0	Digitalni ulaz bez izolacije	'a'
DI1	Digitalni ulaz bez izolacije	'b'
DI2	Digitalni ulaz sa izolacijom	'c'
DI3	Digitalni ulaz sa izolacijom	'd'
AI0	Analogni ulaz	'e'
AI1	Analogni ulaz	'f'
TO0	Tranzistorski izlaz	'p'
TO1	Tranzistorski izlaz	'q'
R00	Relejni izlaz	'r'
R01		
R10	Relejni izlaz	's'
R11		
AO	Analogni izlaz	't'

Tabela 3.2: Oznake ulaza i izlaza IO modula

Kao povratnu poruku uređaj vraća primljenu poruku ukoliko je inicijalizacija uspješna, odnosno poruku greške ukoliko inicijalizacija nije uspješna.

Konfiguracija ulaza

Poruka za konfiguraciju ulaza definiše ulaze, čiju će vrijednost modul vraćati računaru. Znak koji definiše ovu funkcionalnost je 'B'. Parametri poruke su znakovi koji predstavljaju ulaze, koji se žele očitati.

Uređaj kao povratnu poruku vraća primljenu poruku, ukoliko je uspješno izvršena konfiguracija, dok u suprotnom vraća poruku greške. Ukoliko je korisnik pokušao konfigurisati ulaz koji nije inicijaliziran, uređaj će također vratiti poruku greške.

Trigerovanje ulaza

Trigerovanje ulaza služi za očitanje vrijednosti ulaza koji su konfigurisani. Znak za trigerovanje ulaza je 'C'. Ova poruka nema parametara.

Kao povratnu poruku uređaj vraća vrijednosti ulaza, redoslijedom kojim su konfigurisani. Za

digitalne ulaze znak '0' označava logičku nulu, a znak '1' logičku jedinicu. Kao vrijednost analognih ulaza šalju se dva bajta, koji formiraju desetobitnu analognu vrijednost. Ukoliko dođe do greške, uređaj vraća poruku greške.

Konfiguracija izlaza

Poruka za konfiguraciju izlaza služi za postavljanje vrijedosti izlaza, koji će se promijeniti porukom za trigerovanje izlaza. Znak za konfiguraciju izlaza je 'D'. U poruci se šalju znakovi koji označavaju izlaze, iza kojih se šalje vrijednost izlaza. U slučaju da se radi o relejnim ili digitalnim tranzistorskim izlazima šalje se '0' ili '1' (isključen, odnosno uključen relej). Ukoliko se radi o analognom izlazu šalje se jedan bajt koji predstavlja vrijednost izlaza skaliranu na opseg od 0 do 10 V. Ukoliko je slučaj o tranzistorskem PWM izlazu, bajt koji se šalje kao parametar predstavlja vrijednost *duty cycle*.

Kao povratnu poruku uređaj šalje primljenu poruku, ukoliko je uspješno izvršena konfiguracija. U suprotnom se šalje poruka greške. Ukoliko je korisnik pokušao konfigurisati izlaz koji nije prethodno inicijaliziran, uređaj će prijaviti grešku.

Trigerovanje izlaza

Trigerovanje izlaza služi za postavljanje vrijedosti izlaza na konfigurisane vrijednosti. Znak za trigerovanje izlaza je 'E'. Ova poruka nema parametara.

Kao povratnu poruku uređaj vraća primljenu poruku, ukoliko je uspješno izvršeno trigerovanje. Ukoliko dođe do pogreške, uređaj vraća poruku greške.

Očitanje jednog ulaza

Poruka za očitanje jednog ulaza služi za očitanje nekog od ulaza. Znak za ovu funkcionalnost je 'F'. Kao parametar šalje se bajt koji označava ulaz čija se vrijednost želi očitati.

Ukoliko je očitanje uspješno uređaj vraća vrijednost traženog ulaza, u suprotnom vraća poruku greške. Kao i kod konfiguracije ulaza, i u ovom slučaju ukoliko se pokuša očitati vrijednost ulaza koji prethodno nije inicijaliziran, uređaj će vratiti poruku greške.

Postavljanje jednog izlaza

Postavljanje jednog izlaza je poruka čiji je znak 'G'. Ova poruka služi za postavljanje jednog od izlaza na zadalu vrijednost, tako što se kao parametri šalju znak željenog izlaza, te vrijednost na koju se postavlja.

Uređaj kao povratnu poruku vraća primljenu poruku. Ukoliko dođe do pogreške, vraća se poruka greške.

Trigerovanje čitanja i pisanja istovremeno

Trigerovanje čitanja i pisanja istovremeno je poruka koja trigeruje i čitanje i pisanje, tj. ima istu funkciju kao i prethodno navedene poruke za trigerovanje. Znak za ovu poruku je 'H'.

Strobe

Poruka strobe izvršava funkcionalnost čitanja ulaza n puta periodično periodom T . Znak za poruku je 'I'. Kao parametar šalje se broj čitanja n , te broj koji pomnožen sa 16 ms daje period T .

Prekidni mod

Poruka za prekidni mod ima znak 'J', i izvršava aktivaciju očitanja ulaza na promjenu stanja. Ova poruka ima jedan parametar ('0' za deaktivaciju, '1' za aktivaciju).

Povratna poruka je ista kao i primljena. Ukoliko dođe do promjene stanja jednog od ulaza, koji je prethodno konfigurisan, uređaj šalje vrijednosti svih konfigurisanih ulaza.

Poruka greške

Poruka greške formira se tako što se kao prvi parametar šalje znak '?', a zatim znak koji predstavlja kod greške. U tabeli 3.3 su data značenja kodova greške.

Znak	Značenje
'I'	Pogrešan znak. Nemoguće procesirati poruku.
'O'	Konfigurisani ulaz nije inicijaliziran.
'P'	Konfigurisani izlaz nije inicijaliziran.
'Q'	Pogrešna dužina poruke

Tabela 3.3: Kodovi poruke greške

Poglavlje 4

Testiranje rada modula

U ovom dijelu rada bit će testiran rad modula. Testiranje se sastoji od testiranja komunikacije, te električnih prilika modula, za unaprijed definisani scenario. Slanje podataka od strane računara vrši se pomoću programskog paketa *Matlab*. Definisan je scenario koji se sastoji u sljedećem:

- Inicijalizacija digitalnih ulaza DI0 i DI2, analognog ulaza AI0, reljnog izlaza R0, tranzistorskog PWM izlaza TO0 i analognog izlaza,
- Konfiguracija ulaza AI1,
- Konfiguracija izlaza TO1,
- Konfiguracija ulaza DI2 i AI0,
- Trigerovanje ulaza,
- Konfiguracija izlaza R0 i A0,
- Trigerovanje izlaza,
- Očitanje vrijednosti DI0 ulaza,
- Postavljanje tranzistorskog izlaza TO1,
- Aktivacija Strobe-a 10 puta, svakih 80 ms,
- Aktivacija prekidnog moda za zadatu konfiguraciju.

U tabeli 4.1 prikazane su poruke opisanog scenarija. Sa znakom 'X' u porukama označena je adresa uređaja na kojem je vršeno testiranje. Odabrana je adresa 215, pri čemu su položaji dip prekidača postavljeni na "010111" (23), na što se dodaje 192, jer se radi o IO modulu.

Kod očitanja vrijednosti analognog ulaza koriste se 2 bajta, pri čemu se koriste zadnja dva bita prvog bajta i čitav drugi bajt, što je ukupno 10 bita. Ukoliko je analogni ulaz konfigurisan za korištenje unipolarnog naponskog nivoa 0...10 V, onda je vrijednost napona jednaka $(256a + b)/1024 * 10$ V. Oznaka a je za prvi, a oznaka b za drugi bajt analognog ulaza.

Kod postavljanja vrijednosti analognog izlaza pretvorba se vrši na sličan način. Vrijednost napona na izlazu je $c/255 * 10$ V, gdje oznaka c označava bajt kojim se vrijednost izlaza postavlja.

Računanje broja slanja ulaznih vrijednosti kod funkcije za trigerovanje strobe-a vrši se dodavanjem jedinice na vrijednost poslanog bajta. Period slanja podataka se računa kao $16 * (d + 1)$ ms, gdje je d oznaka za bajt kojim se zadaje period strobe-a.

Red. broj	Odlazna poruka	Dolazna poruka
1.	"##XAacep1rt"	"##XAacep1rt"
2.	"##XBf"	"##XB?O"
3.	"##XDq2"	"##XD?P"
4.	"##XBce"	"##XBce"
5.	"##XC"	[##XC", "1", 0, 171]
6.	"##XC"	[##XC", "1", 2, 48]
7.	"##XC"	[##XC", "0", 3, 191]
8.	"##XDr1td"	"##XDr1td"
9.	"##XE"	"##XE"
10.	"##XDr1ty"	"##XDr1ty"
11.	"##XE"	"##XE"
12.	"##XFa"	"##XF0"
13.	"##XFa"	"##XF1"
14.	"##XGpP"	"##XGpP"
15.	[##XI", 9,4]	"##XI"
16.	"##XJ1"	"##XJ"

Tabela 4.1: Odgovori uređaja na poslane poruke

U nastavku su opisana značenja poruka u koracima prikazanim u tabeli.

U prvom koraku poslana je poruka za inicijalizaciju ulaznih i izlaznih portova. Na osnovu sadržaja poruke zaključuje se da je pokušana inicijalizacija digitalnih ulaza DI0 i DI2, analognog ulaza AI0, tranzistorskog izlaza TO0 (za korištenje PWM-a), relejnog izlaza R0, te analognog izlaza AO. Uređaj je vratio istu poruku, što znači da je inicijalizacija uspješno izvršena.

U drugom i trećem koraku poslane su poruke za konfiguraciju analognog ulaza AI1, odnosno tranzistorskog izlaza TO1. Iz sadržaja dolaznih poruke primijećuje se da je došlo do greške pri konfiguraciji, jer analogni ulaz AI1, odnosno tranzistorski izlaz TO1 nisu prethodno inicijalizirani.

U četvrtom koraku poslana je poruka za konfiguraciju ulaza DI2, te AI0. Uređaj je vratio istu poruku, pa se zaključuje da je konfiguracija uspješno izvršena.

U naredna tri koraka poslane su poruke za trigerovanje ulaza, za različite vrijednosti napona na ulazima. Ulazi DI2 i AI0 su kratko spojeni. Prilikom prvog trigerovanja vrijednost napona na ulazima je bila 1.7 V. Na osnovu sadržaja dolazne poruke očitava se vrijednost logičke jedinice na ulazu DI2 (obrnuta logika), te napon od 1.67 V na analognom ulazu (formula navedena ranije). Kod drugog trigerovanja napon na ulazima je imao vrijednost od 5.52 V. Dešifrovanjem dolazne poruke dobija se vrijednost logičke jedinice na ulazu DI2, te vrijednost napona od 5.47 V. Trećim trigerovanjem dobijena je vrijednost logičke nule na ulazu DI2, te vrijednost napona 9.36 V na ulazu AI0. Vrijednost napona mjerena voltmetrom iznosila je 9.46 V.

U osmom koraku poslana je poruka za konfiguraciju izlaza. Poruka sadrži konfiguraciju za

uključenje relejnog izlaza, te postavljanje analognog izlaza na vrijednost 3.92 V (znak 'd' ima vrijednost 100, formula za računanje navedena ranije). Uređaj je vratio istu poruku, što znači da je konfiguracija uspješno izvršena.

U sljedećem koraku poslana je poruka za trigerovanje izlaza. Uređaj je vratio istu poruku, što znači da je prethodnu konfiguraciju izlaza uspješno primjenio. Relej se uključio, dok je na analognom izlazu voltmetrom izmjerena napon vrijednosti 3.95 V. Napon ripla izlaznog napona je izmjerena na osciloskopu i iznosi 40 mVpp (od vrha do vrha, eng. *peak to peak*).

U koracima deset i jedanaest poslane su također poruka za konfiguraciju izlaza, te poruka za trigerovanje izlaza. Konfiguracijom je relej ostavljen uključen, dok je vrijednost analognog izlaza promjenjena na 10 V. Voltmetrom je na analognom izlazu izmjerena napon od 10.04 V, greške 40 mVpp.

U naredna dva koraka poslana je poruka za čitanje jednog ulaza (DI2). Uređaj je u prvom slučaju, vratio vrijednost logičke nule, dok je voltmetrom izmjerena vrijednost napona 6.63 V. U drugom slučaju vraćena je vrijednost logičke jedinice, dok je vrijednost mjereno napona 13.17 V.

U četrnaestom koraku poslana je poruka za postavljanje tranzistorskog izlaza TO0 na vrijednost popunjenoosti PWM-a od 31.4%. Osciloskopom je izmjerena približno ista vrijednost.

U koraku petnaest poslana je poruka za trigerovanje strobe-a sa postavkama $n=10$, te $T=80$ ms. Uređaj je vratio 10 poruka formata kao kod trigerovanja ulaza.

Zadnja funkcija koja je testirana je trigerovanje prekidnog moda. Nakon postavke ove funkcije, promjenom vrijednosti napona na ulazu DI2, uređaj je slao vrijednosti svih konfigurisanih ulaza (DI2, AI0).

Na osnovu predloženog testiranja, zaključuje se da su dobijeni rezultati u skladu sa očekivanim. Komunikacija između računara i IO modula radi prema zadatom scenariju. Očitanja ulaza i postavljanje izlaza su u skladu sa očekivanim. Kod analognih ulaza i analognog izlaza, primjećuje se odstupanje, uzrokovano nepreciznošću voltmetra, kao i nepreciznošću samog modula. Odstupanja su u dozvoljenim granicama.

Poglavlje 5

Zaključak

U ovom radu prezentiran je jedan od načina realizacije IO modula. Sa ekonomskog aspekta, u poređenju sa komercijalnim PLC-ovima, ponuđeno rješenje je neuporedivo jeftinije. Međutim sa aspekta tehničkih mogućnosti, realizovani uređaj je dosta ograničen u odnosu na komercijalna rješenja. Ovo se ogleda u broju ulaza i izlaza, njihovim električnim karakteristikama, te brzini rada. Jedan od većih nedostataka realizovanog modula je nestandardno napajanje. Nameće, potrebna su 4 naponska nivoa za pravilan rad uređaja.

Smjernice za budući rad

Za realizaciju modula u ovom radu, kao upravljačka jedinica korišten je mikrokontroler Atmel ATmega328p. Povećanje mogućnosti i poboljšanje karakteristika realizovanog modula moguće je izvršiti korištenjem mikrokontrolera boljih karakteristika. Ove karakteristike se ogledaju uglavnom u brzini rada, te broju ulaznih i izlaznih portova, kao i dodatnim hardverskim modulima implementiranim na kontroleru.

Za potrebe komunikacije realizovanog modula korišten je RS485 standard, dok je protokol pravljen od strane autora. U budućem radu bi bilo dobro koristiti neki od standardizovanih protokola, kako bi se uređaj mogao povezati na ostale (komercijalne) uređaje bez potrebnih prilagođenja.

Prilozi

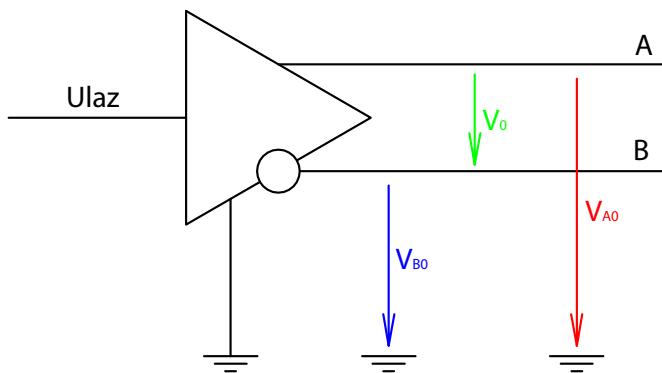
Prilog A

Teoretski aspekti komunikacije

A.1 RS485

RS485 (puni naziv *TIA/EIA-485*) je komunikacijski standard koji definiše električne karakteristike predajnika i prijemnika u okviru serijske komunikacije unutar sistema. Pri tome se ne dotiče komunikacijskog protokola koji sistem koristi, dakle zadržava se na fizičkom nivou.

RS485 se zasniva na diferencijalnom načinu rada. Podaci između korisnika prenose se preko dvije linije (A i B, slika A.1). Predajnik generiše signale na linijama A i B na osnovu digitalnog podatka na ulazu. Ukoliko je ulaz logička "1", na izlazu iz predajnika vrijedi $V_{A0} > V_{B0}$, dok za logičku "0" vrijedi suprotno. Ovakav način rada smanjuje utjecaj smetnji na podatke koji se prenose, pa se na taj način smanjuje vjerovatnoća za nastankom greške prilikom prenosa podataka. Kako se vodovi A i B nalaze u istom okruženju, to na njih djeluju isti šumovi (indukovani elektromagnetski valovi). Zbog toga razlika između napona V_{A0} i V_{B0} ostaje ista, tj. ne dolazi do greške pri prenosu.

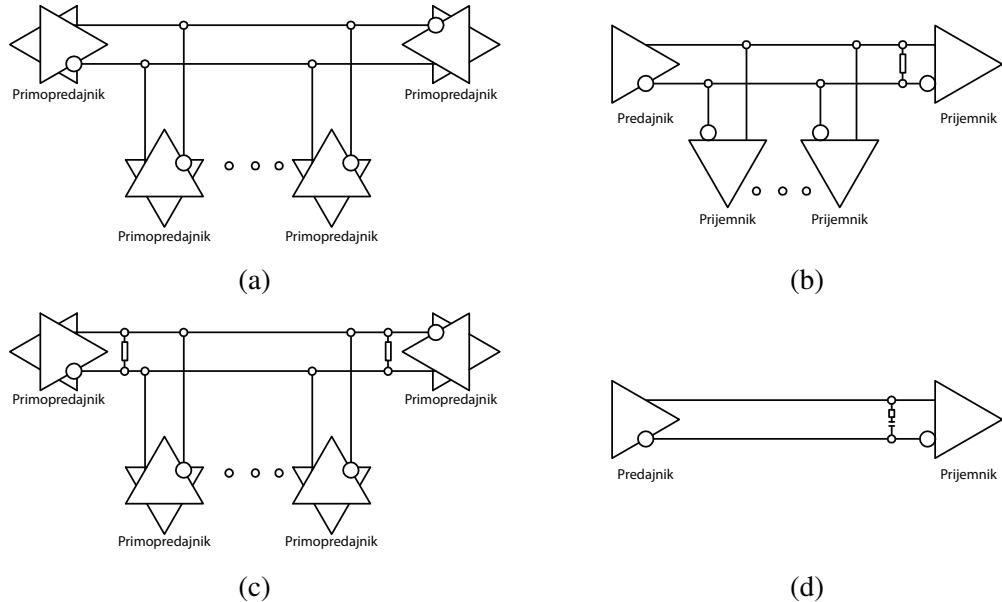


Slika A.1: Diferencijalni naponi RS485 predajnika

A.1.1 Terminacija prenosnih vodova

Uzrok česte pojave greške pri prenosu podataka je refleksija napona na krajevima vodova. Ovo se najčešće dešava na vodovima dužina preko 50 m, pri frekvenciji signala oko 100 kHz. Kako bi se smanjio utjecaj refleksije napona, vodove je potrebno zatvoriti (terminirati) otpornicima.

Postoji više različitih načina kako implementirati terminaciju [19]. Na slici A.2 je prikazano nekoliko tipičnih konfiguracija [14].



Slika A.2: Terminacija vodova

- *Bez zatvaranja* (slika A.2a) - mala cijena, mala snaga i jednostavnost, manje brzine prenosa podataka, manja udaljenost,
- *Zatvaranje otpornikom sa jedne strane voda* (slika A.2b) - odličan pristup kada se u mreži nalazi jedan predajnik, potrebno zatvoriti kraj voda koji se nalazi kod prijemnika, vrijednost otpornika jednaka vrijednosti karakteristične impedanse voda (najčešće 120Ω),
- *Dvostrano zatvaranje otpornikom* (slika A.2c) - predajnik se može nalaziti bilo gdje u mreži, odlikuje ga veća potrošnja, najčešće u primjeni,
- *AC terminacija* (slika A.2d) - kondenzator se uvodi zbog eliminacije DC komponente (manja potrošnja), ograničava brzinu slanja podataka te dužinu vodova, rijetko se koristi.

Pored zatvaranja vodova koji se koriste u komunikaciji, potrebno je zatvoriti i vodove, koji se ne koriste, a nalaze se unutar kabla. Ukoliko se ostave otvoreni, u njima se indukuje napon i na taj način utiču na ostale vodove. Spajanjem krajeva nekorištenih vodova na masu, njihov utjecaj se smanjuje.

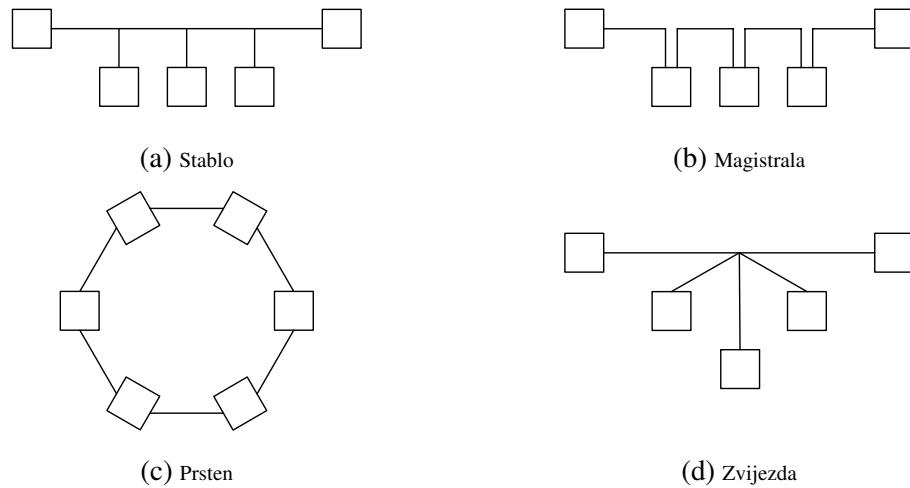
A.2 Topologija mreže

Topologija mreže podrazumijeva međusobni raspored, te povezanost uređaja koji učestvuju u komunikaciji. Četiri osnovne topologije su (slika A.3):

- Stablo (eng. *Bus*) – svi uređaji su povezani na jednu liniju (stablo) (slika A.3a).
- Magistrala (eng. *Daisy chain*) – svaki uređaj je povezan sa svojim susjedom (slika A.3b).
- Prsten (eng. *Ring*) – uređaji su povezani u krug (slika A.3c).

- Zvijezda (eng. *Star*) – svi uređaji u mreži spojeni su u jednu tačku (slika A.3d).

Najbolju otpornost na refleksije pokazuje topologija magistrale, pa je i najčešće u upotrebi. Najjednostavnija za izvedbu ovog projekta je topologija stabla. Računar i module sistema spajamo na "stablo". Komunikacija između računara i bilo kojeg od modula obavlja se približno istom brzinom.



Slika A.3: Topologije mreže

Prilog B

Programski kodovi

B.1 Program mikrokontrolera

```
1  **** ZAVRSNI RAD ****
2  ***** Dizajn ekspanzionog IO modula za PC ****
3  ***** Mentor: Emir Sokic ****
4  ***** Student: Jasmin Hadzajlic ****/
5
6  /* Varijable */
7  // Adresa
8  byte adresa;
9  const byte broadcast = 0b10101010;
10
11 // Ulazi
12 boolean konfigurisani_ulazi[4] = {};
13 uint8_t prethodna_stanja_ulaza[4];
14 uint8_t DI0, DI1, DI2, DI3;
15 uint16_t AI0, AI1;
16 /* Niz flagova za inicializaciju
17 Ulazi
18 Oznaka | U poruci | Indeks
19 DI0    | a        | 0
20 DI1    | b        | 1
21 DI2    | c        | 2
22 DI3    | d        | 3
23 AI0    | e        | 4
24 AI1    | f        | 5
25 Izlazi
26 TO0    | 1        | 6
27 TO1    | 2        | 7
28 REL0   | 3        | 8
29 REL1   | 4        | 9
30 AO     | 5        | 10
31
32 U poruci | U nizu
33 TO0 i TO1
34 0        | 1 - digitalni izlaz
35 1        | 2 - pwm izlaz
36 Ostalo
37 -        | 1
38
39 0 za nekoristene ulaze i izlaze
40 */
```

```
41 int inicijalizacija[11];
42 /* Niz za konfiguraciju ulaza
43 konfig_ulaza
44 Sadrzi znak (char) odgovarajucih ulaza cija se vrijednost trazi*/
45 byte konfig_ulaza[6];
46 int ulaz_velicina;
47 /* Niz za konfiguraciju izlaza
48 konfig_izlaza
49 Sadrzi znak (char) odgovarajucih izlaza ciji se izlaz postavlja
50 vrijednost_izlaza
51 Sadrzi vrijednost, pri cemu se indeksi poklapaju sa konfig_izlaza
52 Digitalni izlazi '0' ili '1' (char)
53 Analogni i PWM 0-255 */
54 byte konfig_izlaza[5];
55 byte vrijednost_izlaza[5];
56 int izlaz_velicina;
57 /* Konfiguracija strobe-a
58 Brojaci za odbrojavanje broja slanja i perioda slanja
59 broj_slanja i period_slanja - zadane vrijednosti*/
60 boolean strobe_flag;
61 int brojac_strobe;
62 int broj_slanja;
63 int period_slanja;
64 int brojac_slanja;
65
66 /* Poruke
67 Nizovi dolazne i odlazne poruke i duzine poruka*/
68 int duzina_dolazne_poruke, duzina_odlazne_poruke;
69 byte dolazna_poruka[40], odlazna_poruka[40];
70 /* Flagovi za slanje, gresku, procesiranje, prekidni mod*/
71 boolean greska_flag;
72 boolean slanje_flag;
73 boolean procesiranje_flag;
74 boolean prekidni_mod_flag;
75
76 /* Funkcija za ocitanje bita iz registra*/
77 byte citajBit(volatile uint8_t &registar, uint8_t redni_broj) {
78     if (registar & (1 << redni_broj))
79         return '1';
80     return '0';
81 }
82
83 /* Funkcija za setovanje bita u registru*/
84 void setujBit(volatile uint8_t &registar, uint8_t redni_broj) {
85     registar |= 1 << redni_broj;
86 }
87
88 /* Funkcija za resetovanje bita u registru*/
89 void resetujBit(volatile uint8_t &registar, uint8_t redni_broj) {
90     registar &= ~ (1 << redni_broj);
91 }
92
93 /* Funkcija za formiranje poruke greske i indikaciju*/
94 void greska(byte sifra) {
95     greska_flag = true;
96     odlazna_poruka[4] = '?';
97     odlazna_poruka[5] = sifra;
98     duzina_odlazne_poruke = 6;
```

```

99    }
100
101 /* Funkcija za citanje vrijednosti analognog ulaza*/
102 uint16_t ADCitaj(uint8_t kanal) {
103     // Odabir kanala
104     ADMUX &= 0b11110000;
105     ADMUX |= 0b00001111 & kanal;
106     // Pocetak konverzije
107     ADCSRA |= (1 << ADSC);
108     // Cekanje kraja konverzije
109     while (!(ADCSRA & (1 << 4)));
110     resetujBit(ADCSRA, 4);
111     // Povratak 10-bitne vrijednosti
112     return ADC;
113 }
114
115 /* Funkcija za inicijalizaciju timera 0 za PWM na TO0 i TO1 */
116 void PWMinit() {
117     // Preskaler 1/8, Fast PWM
118     TCCR0A = 0b00000011;
119     TCCR0B = 0b11000011;
120     TCNT0 = 0x00;
121 }
122
123 /* Funkcija za inicijalizaciju ulaza i izlaza koji se koriste*/
124 void inicijaliziraj() {
125     // Resetovanje inicijalizacije
126     for (int i = 0; i < 11; i++)
127         inicijalizacija[i] = 0;
128     // Ucitavanje inicijalizacije u niz flagova
129     for (int i = 4; i < duzina_dolazne_poruke - 3; i++) {
130         // Formiranje odlazne poruke
131         odlazna_poruka[duzina_odlazne_poruke] = dolazna_poruka[i];
132         duzina_odlazne_poruke++;
133         switch (dolazna_poruka[i]) {
134             case 'a': // DIO
135                 inicijalizacija[0] = 1;
136                 break;
137             case 'b': // DI1
138                 inicijalizacija[1] = 1;
139                 break;
140             case 'c': // DI2
141                 inicijalizacija[2] = 1;
142                 break;
143             case 'd': // DI3
144                 inicijalizacija[3] = 1;
145                 break;
146             case 'e': // AI0
147                 inicijalizacija[4] = 1;
148                 break;
149             case 'f': // AI1
150                 inicijalizacija[5] = 1;
151                 break;
152             case 'p': // TO0
153                 i++;
154                 odlazna_poruka[duzina_odlazne_poruke] = dolazna_poruka[i];
155                 duzina_odlazne_poruke++;
156                 if (dolazna_poruka[i] == '0') { // digitalni izlaz

```

```

157         inicijalizacija[6] = 1;
158         resetujBit(TCCR0A, COM0B1); // Iskljucenje PWM-a
159     }
160     else if (dolazna_poruka[i] == '1') { // PWM izlaz
161         inicijalizacija[6] = 2;
162         setujBit(TCCR0A, COM0B1); // Ukljucenje PWM-a
163     }
164     else
165         greska('I'); // Pogresan znak
166         break;
167     case 'q': // TO1
168         i++;
169         odlazna_poruka[duzina_odlazne_poruke] = dolazna_poruka[i];
170         duzina_odlazne_poruke++;
171         if (dolazna_poruka[i] == '0') { // digitalni izlaz
172             inicijalizacija[7] = 1;
173             resetujBit(TCCR0A, COM0A1); // Iskljucenje PWM-a
174         }
175         else if (dolazna_poruka[i] == '1') { // PWM izlaz
176             inicijalizacija[7] = 2;
177             setujBit(TCCR0A, COM0A1); // Ukljucenje PWM-a
178         }
179         else
180             greska('I'); // Pogresan znak
181             break;
182     case 'r': // REL0
183         inicijalizacija[8] = 1;
184         break;
185     case 's': // REL1
186         inicijalizacija[9] = 1;
187         break;
188     case 't': // AO
189         inicijalizacija[10] = 1;
190         break;
191     default:
192         greska('I'); // Pogresan znak
193         break;
194     }
195     if (greska_flag) // Greska - napustanje funkcije
196         break;
197     }
198 }
199
200 /* Funkcija za konfiguraciju citanja (ulaza) */
201 void konfigurisiCitanje() {
202     for (int i = 0; i < 4; i++)
203         konfigurisani_ulazi[i] = false;
204     ulaz_velicina = 0;
205     for (int i = 4; i < duzina_dolazne_poruke - 3; i++) {
206         switch (dolazna_poruka[i]) // Provjera da li je ulaz inicijalizovan
207         {
208             case 'a':
209                 if (inicijalizacija[0] == 0)
210                     greska('O');
211                 konfigurisani_ulazi[0] = true;
212                 break;
213             case 'b':
214                 if (inicijalizacija[1] == 0)

```

```

215         greska('O');
216         konfigurisani_ulazi[1] = true;
217         break;
218     case 'c':
219         if (inicijalizacija[2] == 0)
220             greska('O');
221         konfigurisani_ulazi[2] = true;
222         break;
223     case 'd':
224         if (inicijalizacija[3] == 0)
225             greska('O');
226         konfigurisani_ulazi[3] = true;
227         break;
228     case 'e':
229         if (inicijalizacija[4] == 0)
230             greska('O');
231         break;
232     case 'f':
233         if (inicijalizacija[5] == 0)
234             greska('O');
235         break;
236     default:
237         greska('I');
238         break;
239     }
240     if (greska_flag)
241         return;
242
243     // Konfiguracija
244     konfig_ulaza[ulaz_velicina] = dolazna_poruka[i];
245     ulaz_velicina++;
246
247     // Formiranje odlazne poruke
248     odlazna_poruka[duzina_odlazne_poruke] = dolazna_poruka[i];
249     duzina_odlazne_poruke++;
250 }
251 }
252
253 /* Funkcija za konfiguraciju pisanja (izlaza) */
254 void konfigurisiPisanje() {
255     izlaz_velicina = 0;
256     for (int i = 4; i < duzina_dolazne_poruke - 3; i++) {
257         switch (dolazna_poruka[i]) {
258             case 'p':
259                 if (inicijalizacija[6] == 0)
260                     greska('P');
261                 break;
262             case 'q':
263                 if (inicijalizacija[7] == 0)
264                     greska('P');
265                 break;
266             case 'r':
267                 if (inicijalizacija[8] == 0)
268                     greska('P');
269                 break;
270             case 's':
271                 if (inicijalizacija[9] == 0)
272                     greska('P');

```

```

273     break;
274 case 't':
275     if (inicijalizacija[10] == 0)
276         greska('P');
277     break;
278 default:
279     greska('I');
280     break;
281 }
282 if (greska_flag)
283     return;
284
285 // Konfiguracija izlaza
286 konfig_izlaza[izlaz_velicina] = dolazna_poruka[i];
287 i++;
288 // POTREBNO PROVJERITI VRIJEDNOST '0' ili '1'
289 vrijednost_izlaza[izlaz_velicina] = dolazna_poruka[i];
290 izlaz_velicina++;
291
292 // Formiranje odlazne poruke
293 odlazna_poruka[duzina_odlazne_poruke] = dolazna_poruka[i - 1];
294 duzina_odlazne_poruke++;
295 odlazna_poruka[duzina_odlazne_poruke] = dolazna_poruka[i];
296 duzina_odlazne_poruke++;
297 }
298 }
299
300 /* Funkcija za trigerovanje citanja */
301 void trigerujCitanje() {
302     for (int i = 0; i < ulaz_velicina; i++) {
303         switch (konfig_ulaza[i]) {
304             case 'a':
305                 odlazna_poruka[duzina_odlazne_poruke] = DI0;
306                 duzina_odlazne_poruke++;
307                 break;
308             case 'b':
309                 odlazna_poruka[duzina_odlazne_poruke] = DI1;
310                 duzina_odlazne_poruke++;
311                 break;
312             case 'c':
313                 odlazna_poruka[duzina_odlazne_poruke] = DI2;
314                 duzina_odlazne_poruke++;
315                 break;
316             case 'd':
317                 odlazna_poruka[duzina_odlazne_poruke] = DI3;
318                 duzina_odlazne_poruke++;
319                 break;
320             case 'e':
321                 odlazna_poruka[duzina_odlazne_poruke] = (AI0 & 0x0300) >> 8;
322                 duzina_odlazne_poruke++;
323                 odlazna_poruka[duzina_odlazne_poruke] = AI0 & 0xFF;
324                 duzina_odlazne_poruke++;
325                 break;
326             case 'f':
327                 odlazna_poruka[duzina_odlazne_poruke] = (AI1 & 0x0300) >> 8;
328                 duzina_odlazne_poruke++;
329                 odlazna_poruka[duzina_odlazne_poruke] = AI1 & 0xFF;
330                 duzina_odlazne_poruke++;

```

```

331     break;
332     default:
333         greska('I');
334         break;
335     }
336     if (greska_flag)
337         return;
338 }
339 }
340
341 /* Funkcija za trigerovanje pisanja */
342 void trigerujPisanje() {
343     for (int i = 0; i < izlaz_velicina; i++) {
344         switch (konfig_izlaza[i]) {
345             case 'p':
346                 // TO0
347                 switch (inicijalizacija[6])
348                 {
349                     // Digitalni izlaz
350                     case 1:
351                         switch (vrijednost_izlaza[i]) {
352                             case '0':
353                                 resetujBit(PORTD, PD5);
354                                 break;
355                             case '1':
356                                 setujBit(PORTD, PD5);
357                                 break;
358                         }
359                         break;
360                         // PWM izlaz
361                     case 2:
362                         OCR0B = vrijednost_izlaza[i];
363                         break;
364                     }
365                     break;
366                 case 'q':
367                     // TO1
368                     switch (inicijalizacija[7])
369                     {
370                         // Digitalni izlaz
371                         case 1:
372                             switch (vrijednost_izlaza[i]) {
373                                 case '0':
374                                     resetujBit(PORTD, PD6);
375                                     break;
376                                 case '1':
377                                     setujBit(PORTD, PD6);
378                                     break;
379                             }
380                             break;
381                             // PWM izlaz
382                         case 2:
383                             OCR0A = vrijednost_izlaza[i];
384                             break;
385                         }
386                         break;
387                     case 'r':
388                         // REL0

```

```

389     switch (vrijednost_izlaza[i]) {
390     case '0':
391         resetujBit(PORTC, PC1);
392         break;
393     case '1':
394         setujBit(PORTC, PC1);
395         break;
396     }
397     break;
398     case 's':
399         // REL1
400         switch (vrijednost_izlaza[i]) {
401             case '0':
402                 resetujBit(PORTC, PC2);
403                 break;
404             case '1':
405                 setujBit(PORTC, PC2);
406                 break;
407             }
408             break;
409     case 't':
410         // AO
411         OCR1B = vrijednost_izlaza[i];
412         break;
413     default:
414         greska('I');
415         break;
416     }
417     if (greska_flag)
418         return;
419 }
420 }
421
422 /* Funkcija za citanje ulazne poruke */
423 void citajSerial() {
424     resetujBit(PORTC, PC3);
425     duzina_dolazne_poruke = 0;
426     int i = 0;
427     while (i < 1000) {
428         delay(1);
429         if (Serial.available()) {
430             dolazna_poruka[i] = Serial.read();
431             if (dolazna_poruka[i] == '\n' && dolazna_poruka[i - 1] == '\r' &&
432                 dolazna_poruka[i - 2] == '\r') {
433                 i++;
434                 break;
435             }
436             i++;
437         }
438     }
439     duzina_dolazne_poruke = i;
440     if (duzina_dolazne_poruke < 7) {
441         slanje_flag = false;
442         procesiranje_flag = false;
443     }
444 }
445 /* Funkcija za pisanje izlazne poruke */

```

```

446 void pisiSerial() {
447     setujBit(PORTC, PC3);
448     // Znakovi za kraj poruke
449     odlazna_poruka[duzina_odlazne_poruke++] = '\r';
450     odlazna_poruka[duzina_odlazne_poruke++] = '\r';
451     odlazna_poruka[duzina_odlazne_poruke++] = '\n';
452
453     for (int i = 0; i < duzina_odlazne_poruke; i++) {
454         Serial.write(odlazna_poruka[i]);
455         while (ocitajBit(UCSR0A, TXC0) == '0');
456     }
457     duzina_odlazne_poruke = 3;
458     resetujBit(PORTC, PC3);
459 }
460
461 /* Funkcija za ocitanje jednog ulaza */
462 void ocitajUlaz() {
463     switch (dolazna_poruka[4]) {
464     case 'a':
465         if (inicijalizacija[0] == 0)
466             greska('O');
467         else {
468             odlazna_poruka[duzina_odlazne_poruke] = DI0;
469             duzina_odlazne_poruke++;
470         }
471         break;
472     case 'b':
473         if (inicijalizacija[1] == 0)
474             greska('O');
475         else {
476             odlazna_poruka[duzina_odlazne_poruke] = DI1;
477             duzina_odlazne_poruke++;
478         }
479         break;
480     case 'c':
481         if (inicijalizacija[2] == 0)
482             greska('O');
483         else {
484             odlazna_poruka[duzina_odlazne_poruke] = DI2;
485             duzina_odlazne_poruke++;
486         }
487         break;
488     case 'd':
489         if (inicijalizacija[3] == 0)
490             greska('O');
491         else {
492             odlazna_poruka[duzina_odlazne_poruke] = DI3;
493             duzina_odlazne_poruke++;
494         }
495         break;
496     case 'e':
497         if (inicijalizacija[4] == 0)
498             greska('O');
499         else {
500             odlazna_poruka[duzina_odlazne_poruke] = (AI0 & 0x0300) >> 8;
501             duzina_odlazne_poruke++;
502             odlazna_poruka[duzina_odlazne_poruke] = AI0 & 0xFF;
503             duzina_odlazne_poruke++;
504         }
505     }
506 }

```

```

504     }
505     break;
506 case 'f':
507     if (inicijalizacija[5] == 0)
508         greska('O');
509     else {
510         odlazna_poruka[duzina_odlazne_poruke] = (AI1 & 0x0300) >> 8;
511         duzina_odlazne_poruke++;
512         odlazna_poruka[duzina_odlazne_poruke] = AI1 & 0xFF;
513         duzina_odlazne_poruke++;
514     }
515     break;
516 default:
517     greska('I');
518     break;
519 }
520 }

/* Funkcija za postavljanje jednog izlaza */
521 void postaviIzlaz() {
522     odlazna_poruka[duzina_odlazne_poruke] = dolazna_poruka[4];
523     duzina_odlazne_poruke++;
524     odlazna_poruka[duzina_odlazne_poruke] = dolazna_poruka[5];
525     duzina_odlazne_poruke++;

526     switch (dolazna_poruka[4]) {
527     case 'p':
528         switch (inicijalizacija[6]) // T00
529         {
530             case 0:
531                 greska('P');
532                 break;
533             case 1: // Digitalni izlaz
534                 switch (dolazna_poruka[5]) {
535                     case '0':
536                         resetujBit(PORTD, PD5);
537                         break;
538                     case '1':
539                         setujBit(PORTD, PD5);
540                         break;
541                     default:
542                         greska('P');
543                         break;
544                     }
545                     break;
546                 }
547             break;
548         case 2:
549             OCR0B = dolazna_poruka[5]; // PWM izlaz
550             break;
551         }
552     }
553     break;
554 case 'q': // T01
555     switch (inicijalizacija[7])
556     {
557         case 0:
558             greska('P');
559             break;
560         case 1: // Digitalni izlaz
561             switch (dolazna_poruka[5]) {

```

```

562     case '0':
563         resetujBit(PORTD, PD6);
564         break;
565     case '1':
566         setujBit(PORTD, PD6);
567         break;
568     default:
569         greska('P');
570         break;
571     }
572     break;
573 case 2: // PWM izlaz
574     OCR0A = dolazna_poruka[5];
575     break;
576 }
577 break;
578 case 'r': // REL0
579     if (inicijalizacija[8] == 0)
580         greska('P');
581     else {
582         switch (dolazna_poruka[5]) {
583             case '0':
584                 resetujBit(PORTC, PC1);
585                 break;
586             case '1':
587                 setujBit(PORTC, PC1);
588                 break;
589             default:
590                 greska('I');
591                 break;
592             }
593         }
594         break;
595 case 's': // REL1
596     if (inicijalizacija[9] == 0)
597         greska('P');
598     else {
599         switch (dolazna_poruka[5]) {
600             case '0':
601                 resetujBit(PORTC, PC2);
602                 break;
603             case '1':
604                 setujBit(PORTC, PC2);
605                 break;
606             default:
607                 greska('I');
608                 break;
609             }
610             break;
611 case 't': // AO
612     if (inicijalizacija[10] == 0)
613         greska('P');
614     else
615         OCR1B = dolazna_poruka[5];
616         break;
617     default:
618         greska('I');
619         break;

```

```

620     }
621 }
622 }
623
624 /* Prekidna rutina svakih 16ms ako je aktivan strobe */
625 ISR(TIMER2_OVF_vect) {
626     if (strobe_flag) {
627         brojac_strobe++;
628         if (brojac_strobe >= period_slanja) { // Slanje stanja ulaza
629             trigerujCitanje();
630             pisiSerial();
631             brojac_strobe = 0;
632             brojac_slanja++;
633             if (brojac_slanja >= broj_slanja) { // Posljednje slanje
634                 brojac_slanja = 0;
635                 strobe_flag = false;
636             }
637         }
638     }
639 }
640
641 /* Funkcija za inicijalizaciju timer2 koji se koristi za prekide */
642 void inicijalizirajPrekide() {
643     // Postavljanje prescalera na 128, 16ms prekidi
644     TCCR2B = 0b00000101;
645 }
646
647 /* Funkcija za trigerovanje strobe-a */
648 void trigerujStrobe() {
649     strobe_flag = true;
650     broj_slanja = 1 + dolazna_poruka[4]; // 1 - 256
651     period_slanja = 1 + dolazna_poruka[5] * 4; // Minimalno 16ms, maksimalno
652     4,096s
653     TCNT2 = 0; // Postavljanje brojaca na 0
654     setujBit(TIMSK2, TOIE2); // Aktiviranje prekida na timer2
655 }
656
657 /* Trigerovanje citanja i pisanja istovremeno*/
658 void trigerujSve() {
659     trigerujPisanje();
660     if (!greska_flag)
661         trigerujCitanje();
662 }
663
664 /* Funkcija za trigerovanje prekidnog moda*/
665 void trigerujPrekidniMod() {
666     switch (dolazna_poruka[4]) {
667     case '0':
668         prekidni_mod_flag = false;
669         break;
670     case '1':
671         prekidni_mod_flag = true;
672         break;
673     default:
674         greska('I');
675         break;
676     }
677 }

```

```

677 /* Funkcija za procesiranje poruke */
678 void procesirajPoruku() {
679     // Provjera pocetka i kraja
680     if (dolazna_poruka[0] != '#' || dolazna_poruka[1] != '#' || 
681         dolazna_poruka[duzina_dolazne_poruke - 2] != '\r' || dolazna_poruka[ 
682         duzina_dolazne_poruke - 1] != '\n') {
683         greska('L');
684         return;
685     }
686     // Provjera adresu
687     if (dolazna_poruka[2] != adresa) {
688         slanje_flag = false;
689         if (dolazna_poruka[2] != broadcast) // procesiraj poruku, ali ne
690             odgovaraj (BROADCAST)
691         return;
692     }
693     // Formiranje odlazne poruke
694     odlazna_poruka[duzina_odlazne_poruke] = dolazna_poruka[3];
695     duzina_odlazne_poruke++;
696     // Provjera zadane funkcije
697     switch (dolazna_poruka[3]) {
698     case 'A': // Inicijalizacija
699         inicijaliziraj();
700         break;
701     case 'B': // Konfiguracija za citanje
702         konfigurisiCitanje();
703         break;
704     case 'C': // Trigger za citanje
705         if (duzina_dolazne_poruke != 7)
706             greska('Q');
707         else
708             trigerujCitanje();
709         break;
710     case 'D': // Konfiguracija za pisanje
711         konfigurisiPisanje();
712         break;
713     case 'E': // Trigger za pisanje
714         if (duzina_dolazne_poruke != 7)
715             greska('Q');
716         else
717             trigerujPisanje();
718         break;
719     case 'F': // Ocitaj ulaz
720         if (duzina_dolazne_poruke != 8)
721             greska('Q');
722         else
723             ocitajUlaz();
724         break;
725     case 'G': // Postavi izlaz
726         if (duzina_dolazne_poruke != 9)
727             greska('Q');
728         else
729             postaviIzlaz();
730         break;
731     case 'H': // Trigger i za citanje i za pisanje
732         if (duzina_dolazne_poruke != 7)
733             greska('Q');

```

```

732     else
733         trigerujSve();
734     break;
735 case 'I': // Strobe
736     if (duzina_dolazne_poruке != 9)
737         greska('Q');
738     else
739         trigerujStrobe();
740     break;
741 case 'J': // Prekidni mod
742     if (duzina_dolazne_poruке != 8)
743         greska('Q');
744     else
745         trigerujPrekidniMod();
746     break;
747 default: // Greska
748     greska('I');
749     break;
750 }
751 }
752
753 /* Funkcija za inicijalizaciju izlaza */
754 void inicijalizirajIzlaze() {
755     setujBit(DDRC, PC1);
756     setujBit(DDRC, PC2);
757     setujBit(DDRD, PD5);
758     setujBit(DDRD, PD6);
759 }
760
761 /* Funkcija za inicijalizaciju komunikacije */
762 void inicijalizirajSerial() {
763     Serial.begin(115200);
764     setujBit(DDRC, PC3); // Enable pin
765     resetujBit(PORTC, PC3);
766 }
767
768 /* Funkcija za inicijalizaciju analognog izlaza */
769 void inicijalizirajAnalogniIzlaz() {
770     setujBit(DDRB, PB2);
771     // TIMER 1 - Fast PWM (8 bit) - Clear on Compare Match, set at BOTTOM (non-inverting mode)
772     TCCR1A = 0b00100001;
773     TCCR1B = 0b00001001;
774     TCNT1 = 0x0000;
775
776     OCR1B = 0; // Analogni izlaz
777 }
778
779 /* Funkcija za inicijalizaciju PWM-a za TO0 i TO1 */
780 void inicijalizirajPWM() {
781     // TIMER 0 - Fast PWM - Set on Compare Match, clear at BOTTOM (inverting mode)
782     TCCR0A = 0b00000011;
783     TCCR0B = 0b00000010;
784     TCNT0 = 0x00;
785
786     // TO0 - PWM
787     OCR0B = 0;

```

```

788     // TO1 - PWM
789     OCR0A = 0;
790 }
791
792 /* Funkcija za adresiranje modula */
793 void adresiraj() {
794     // Resetovanje adrese
795     adresa = 0x00;
796     setujBit(DDRB, PB0);
797     setujBit(DDRB, PB1);
798     // Paralelno ucitavanje u shift registar
799     resetujBit(PORTB, PB1);
800     delay(1);
801     setujBit(PORTB, PB1);
802     delay(1);
803     // Ucitavanje adrese u mikrokontroler
804     for (int i = 0; i < 8; i++) {
805         // Citanje adrese
806         if (ocitajBit(PIND, PD7) == '1')
807             setujBit(adresa, i);
808         // Clock za shiftanje
809         setujBit(PORTB, PB0);
810         delay(1);
811         resetujBit(PORTB, PB0);
812         delay(1);
813     }
814     adresa |= 0b11000000;
815 }
816
817 /* SETUP - inicializacija svih koristenih modula kontrolera i varijabli*/
818 void setup() {
819     inicializirajIzlaze();
820     inicializirajAnalogniIzlaz();
821     inicializirajSerial();
822     inicializirajPWM();
823     inicializirajPrekide();
824     strobe_flag = false;
825     brojac_slanja = 0;
826     brojac_strobe = 0;
827     adresiraj();
828
829     // Formiranje pocetka odlazne poruke
830     odlazna_poruka[0] = '#';
831     odlazna_poruka[1] = '#';
832     odlazna_poruka[2] = adresa;
833     duzina_odlazne_poruke = 3;
834     greska_flag = false;
835     slanje_flag = true;
836     procesiranje_flag = true;
837     prekidni_mod_flag = false;
838 }
839
840 void loop() {
841
842     // Cekanje na poruku
843     if (Serial.available()) {
844         citajSerial();
845         if (procesiranje_flag)

```

```
846     procesirajPoruku();
847     if (slanje_flag)
848         pisiSerial();
849
850     // Resetovanje flagova
851     greska_flag = false;
852     slanje_flag = true;
853     procesiranje_flag = true;
854     duzina_odlazne_poruke = 3;
855 }
856 // Kraj strobe-a
857 if (!strobe_flag)
858     resetujBit(TIMSK0, TOIE0);
859 // Osvjezavanje ulaza
860 DIO = ocitajBit(PIND, PD4);
861 DII = ocitajBit(PINC, PC0);
862 DIZ = ocitajBit(PIND, PD2);
863 DIZ = ocitajBit(PIND, PD3);
864 AI0 = ADCitaj(0x07);
865 AI1 = ADCitaj(0x06);
866 // Ako je aktivran prekidni mod
867 if (prekidni_mod_flag && ((DIO != prethodna_stanja_ulaza[0] &&
868     konfigurisani_ulazi[0]) || (DI1 != prethodna_stanja_ulaza[1] &&
869     konfigurisani_ulazi[1]) || (DI2 != prethodna_stanja_ulaza[2] &&
870     konfigurisani_ulazi[2]) || (DI3 != prethodna_stanja_ulaza[3] &&
871     konfigurisani_ulazi[3]))) {
872     odlazna_poruka[duzina_odlazne_poruke++] = 'J';
873     trigerujCitanje();
874     pisiSerial();
875 }
876 }
```

Literatura

- [1] ATmega328/P 8-bit AVR Microcontroller, Atmel Corporation, 2016, datum pristupa: 26.12.2016, dostupno na: <http://www.atmel.com/>
- [2] Batchelor, B. G., Whelan, P. F., (ur.), Intelligent Vision Systems for Industry. Cardiff, Wales: University of Wales, Cardiff, 2012.
- [3] Kreso, S., Huseinbegović, S., Projektovanje mikroprocesorskih sistema. Elektrotehnički fakultet Sarajevo, 2012.
- [4] Logic controller Modicon M241, Schneider electric, 2017, datum pristupa: 16.03.2017, dostupno na: <http://www.schneider-electric.us>
- [5] Siemens S7-1200 Programmable controller, Siemens, 2017, datum pristupa: 20.08.2017, dostupno na: <https://www.siemens.com/>
- [6] Omron CP series controller, Omrons, 2017, datum pristupa: 13.08.2017, dostupno na: <https://www.omron.com/>
- [7] Micro800 Programmable Controller Family, Allan Bradley, 2017, datum pristupa: 20.08.2017, dostupno na: <http://ab.rockwellautomation.com/>
- [8] Modicon TM3 expansion modules, Schneider electric, 2014, datum pristupa: 13.08.2017, dostupno na: <http://www.schneider-electric.us>
- [9] Sensor Catalogue, Schneider electric, 2014, datum pristupa: 16.03.2017, dostupno na: <http://www.schneider-electric.us>
- [10] OsiSense XS Inductive proximity sensors, Schneider electric, 2017, datum pristupa: 16.03.2017, dostupno na: <http://www.schneider-electric.us>
- [11] Osisense XCC Rotary Encoders Catalog, Schneider electric, 2011, datum pristupa: 16.03.2017, dostupno na: <http://www.schneider-electric.us>
- [12] Arduino Uno Reference Design, Rev. 3, Arduino, 2016, dostupno na: https://www.arduino.cc/en/uploads/Main/Arduino_Unc_Rev3-schematic.pdf
- [13] Stojanović, Z., Arhitektura, protokoli i servisi interneta. Evropski univerzitet Brčko distrikta, 2015.
- [14] Perrin, B., The Art and Science of RS-485, (Juli 1999), dostupno na: www.circuitcellar.com

- [15] MAX481/MAX483/MAX485/MAX487/MAX491/MAX1487 Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers, Maxim Integrated, 2014, datum pristupa: 09.03.2017, dostupno na: <http://www.maximintegrated.com./>
- [16] SNx4HC165 8-Bit Parallel-Load Shift Registers, Texas Instruments, 2017, datum pristupa: 18.09.2017, dostupno na: <http://www.ti.com/>
- [17] High-Speed CMOS Logic Decade Counter/Divider CD54HC4017, Texas instruments, 2003, datum pristupa: 21.09.2017, dostupno na: <http://www.ti.com>
- [18] Industrial Current/Voltage Output Driver with Programmable Ranges, Analog Devices, 2014, datum pristupa: 10.09.2017, dostupno na: <http://www.analog.com>
- [19] Vo, J., AN-903 A Comparison of Differential Termination Techniques, National Semiconductor, 1993.