



UNIVERZITET U SARAJEVU  
ELEKTROTEHNIČKI FAKULTET  
ODSJEK ZA AUTOMATIKU I ELEKTRONIKU

---

## Dizajn kontrolabilnog modula za osvjetljenje

---

ZAVRŠNI RAD  
- PRVI CIKLUS STUDIJA -

**Student:**  
**Hasan Grošić**

**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 kontrolabilnog modula za osvjetljenje**

**Student: Hasan Grošić**

#### **Sažetak:**

Prilikom vizuelne inspekcije i korištenja algoritama obrade slike u industrijskom okruženju jedan od najvažnijih elemenata je odgovarajuće osvjetljenje. Zadatak ovog završnog rada je dizajniranje jednog modula za osvjetljenje radne površine vizujskog senzora, i to na način da se intenzitet i prostorna raspodjela svjetlosti koju generiše modul može upravljati putem udaljenog terminala (RS-485 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. ing. el.

## **Izjava o autentičnosti radova**

### **Završni rad I ciklusa studija**

Ime i prezime: Hasan Grošić

Naslov rada: Dizajn kontrolabilnog modula za osvjetljenje

Vrsta rada: Završni rad I ciklusa studija

Broj stranica: 66

#### **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 naznačeno;
- 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.

Potpis:

---

Hasan Grošić

## Sažetak

U radu su obrađeni svi aspekti dizajna i razvoja kontrolabilnog modula za osvjetljenje za potrebe vizuelne inspekcije. Opisane su aktuelne tehnike i tehnologije osvjetljavanja, sa posebnim osvrtom na LED tehnologiju. Predstavljeni su načini upravljanja osvjetljenjem, kao i konkurentna komercijalna rješenja. Prezentirana je hardverska realizacija sistema, uključujući mikrokontrolersku jedinicu, RS485 komunikaciju, LED rasvjetno tijelo, kao i fleksibilnu driversku jedinicu za LED segmente. Pojašnjeni su mogući načini povezivanja LED segmenata sa driverskom jedinicom. Razvijen je protokol za komunikaciju modula sa računarom putem RS485 sabirnice. Opisane su sve funkcionalnosti sistema, te odgovarajuće softverske implementacije.

**Ključne riječi:** osvjetljenje, vizuelna inspekcija, LED, upravljanje, hardver, mikrokontroler, RS485, driver, protokol, softver

## Abstract

The aim of this work is to give insight into all aspects of the design and development of a controllable illumination module for vision inspection purposes. Contemporary illumination techniques and technologies are described, including a special focus on LED technology. Illumination control methods as well as competitive commercial solutions are introduced. The hardware implementation of the system, including the microcontroller unit, RS485 communication, a LED lighting fixture, as well as a flexible driver unit for the LED segments are presented. All possible ways to connect the LED segments to the driver unit are explained. A protocol for communicating the module with the computer via the RS485 bus was developed. All system functionalities and their corresponding software implementations are described.

**Key words:** illumination, vision inspection, LED, control, hardware, microcontroller, RS485, driver, protocol, software

# Sadržaj

<b>Popis slika</b>	<b>v</b>
<b>Popis tabela</b>	<b>vi</b>
<b>1 Uvod</b>	<b>1</b>
1.1 Tehnologije i tehnike osvjetljavanja . . . . .	2
1.1.1 Vrste izvora svjetlosti i spektralni sadržaj . . . . .	2
1.1.2 Tehnike osvjetljavanja . . . . .	5
1.2 Upravljanje osvjetljenjem . . . . .	8
1.2.1 <i>Triniti</i> LED kontroleri . . . . .	8
1.2.2 <i>ULC-2</i> LED kontroler . . . . .	9
1.2.3 <i>DCS-100E</i> LED kontroler . . . . .	10
1.3 Razvoj fleksibilnog modula za osvjetljenje . . . . .	12
<b>2 Hardverska komponenta modula za osvjetljenje</b>	<b>13</b>
2.1 Mikrokontroler . . . . .	14
2.1.1 Atmel ATmega328p . . . . .	14
2.1.2 Funkcionalni moduli i mogućnosti mikrokontrolera ATmega328P . . .	14
2.1.3 Konfiguracija pinova mikrokontrolera ATmega328P (32-pin TQFP) . .	16
2.1.4 Minimalno elektroničko okruženje mikrokontrolera ATmega328p . .	18
2.1.5 Programiranje programske memorije - ISP . . . . .	18
2.2 Komunikaciona komponenta modula za osvjetljenje . . . . .	20
2.2.1 Implementacija RS485 komunikacije . . . . .	20
2.3 Adresiranje modula za osvjetljenje . . . . .	23
2.4 Rasvjetno tijelo . . . . .	25
2.4.1 LED tehnologija . . . . .	25
2.4.2 LED segmenti . . . . .	25
2.5 LED driveri . . . . .	27
2.5.1 Upravljanje LED osvjetljenjem . . . . .	27
2.6 Dizajn štampanih pločica – PCB . . . . .	29
2.6.1 Projektovanje štampanih pločica u programskom paketu EAGLE . . .	29
2.6.2 Fizička realizacija štampanih pločica . . . . .	34
2.7 Načini povezivanja LED segmenata sa kontrolerom . . . . .	36
<b>3 Softversko upravljanje modulom za osvjetljenje</b>	<b>38</b>
3.1 Softverske funkcionalnosti . . . . .	39
3.2 Razvijeni komunikacijski protokol . . . . .	41
3.2.1 Format poruke . . . . .	41
3.2.2 Implementirane instrukcijske poruke . . . . .	41

<b>4 Testiranje rada modula za osvjetljenje</b>	<b>44</b>
<b>Prilozi</b>	<b>48</b>
<b>A Teoretski aspekti komunikacije</b>	<b>49</b>
A.1 RS485 . . . . .	49
A.2 Terminacija prenosnih vodova . . . . .	49
A.3 Topologija mreže . . . . .	50
<b>B Programske kodove</b>	<b>52</b>
B.1 Program mikrokontrolera . . . . .	52
<b>Literatura</b>	<b>65</b>

# Popis slika

1.1	Komparacija najčešće korištenih izvora svjetlosti . . . . .	3
1.2	Relativni intenzitet izvora svjetlosti u zavisnosti od valne dužine . . . . .	4
1.3	Apsolutna kvantna efikasnost različitih tipova senzora u zavisnosti od valne dužine . . . . .	4
1.4	Pozadinsko osvjetljenje . . . . .	6
1.5	Različite implementacije prigušenog osvjetljenja . . . . .	6
1.6	Direkciono osvjetljenje . . . . .	7
1.7	Osvjetljenje sa tamnim poljem . . . . .	7
1.8	Fizički izgled <i>Triniti</i> LED kontrolera . . . . .	9
1.9	Fizički izgled <i>ULC-2</i> LED kontrolera . . . . .	10
1.10	Fizički izgled <i>DCS-100E</i> LED kontrolera . . . . .	11
2.1	Blok struktura modula . . . . .	13
2.2	Fizički izgled mikrokontrolera ATmega328P (32-pin TQFP) . . . . .	15
2.3	Blok dijagram mikrokontrolera ATmega328P [1] . . . . .	16
2.4	Raspored pinova mikrokontrolera ATmega328P (32-pin MLF) . . . . .	17
2.5	Minimalno električko okruženje mikrokontrolera ATmega328P . . . . .	18
2.6	Raspored pinova ISP konektora . . . . .	19
2.7	Primjer implementacije RS485 komunikacije . . . . .	21
2.8	Raspored pinova MAX485 IC . . . . .	21
2.9	Shema sklopa za RS485 komunikaciju . . . . .	22
2.10	Načini hardverskog adresiranja . . . . .	23
2.11	Shema sklopa za učitavanje adrese uređaja . . . . .	24
2.12	Dijagram četverosegmentnog rasporeda rasvjetnog tijela . . . . .	25
2.13	Shema spajanja LED segmenta . . . . .	26
2.14	Upravljanje LED osvjetljenjem pomoću promjenjivog otpora . . . . .	27
2.15	Upravljanje LED osvjetljenjem pomoću NPN tranzistora . . . . .	28
2.16	Upravljanje LED osvjetljenjem pomoću PNP i NPN tranzistora . . . . .	28
2.17	Električna shema spajanja dijela sa mikrokontrolerom . . . . .	30
2.18	Izgled štampane ploče dijela sa mikrokontrolerom . . . . .	30
2.19	Električna shema spajanja dijela za komunikaciju i adresiranje . . . . .	31
2.20	Izgled štampane ploče dijela za komunikaciju i adresiranje . . . . .	31
2.21	Električna shema spajanja dijela sa driverima . . . . .	32
2.22	Izgled štampane ploče dijela sa driverima . . . . .	32
2.23	Električna shema spajanja rasvjetnog tijela . . . . .	33
2.24	Izgled štampane ploče rasvjetnog tijela . . . . .	33
2.25	Fizički izgled štampanih pločica modula za osvjetljenje (bez komponenti) . . . . .	34
2.26	Fizički izgled štampanih pločica modula za osvjetljenje (sa komponentama) . . . . .	34
2.27	Fizički izgled modula za osvjetljenje nakon povezivanja pojedinačnih dijelova . . . . .	35

2.28 Fizički izgled rasvjetnog tijela . . . . .	35
2.29 Raspored ulaza i izlaza modula . . . . .	36
2.30 Mogući načini povezivanja LED segmenata sa kontrolerom . . . . .	37
3.1 Dijagram toka softvera mikrokontrolera . . . . .	38
A.1 Diferencijalni naponi RS485 predajnika . . . . .	49
A.2 Terminacija vodova . . . . .	50
A.3 Topologije mreže . . . . .	51

# Popis tabela

1.1	Osnovne karakteristike LED kontrolera TR-RT220 i TR-RT420 . . . . .	9
1.2	Osnovne karakteristike LED kontrolera ULC-2 . . . . .	10
1.3	Osnovne karakteristike LED kontrolera DCS-100E . . . . .	11
2.1	Oznake elemenata modula . . . . .	36
3.1	Modovi rada <i>sink</i> drivera i pripadni parametri . . . . .	40
3.2	Format poruke korišten za realizaciju komunikacije uređaja . . . . .	41
3.3	Interpretacija sadržaja poruke za konfiguraciju <i>sink</i> drivera . . . . .	42
3.4	Kodovi poruke greške . . . . .	43
4.1	Korišteni način povezivanja modula prilikom testiranja . . . . .	44
4.2	Odgovori modula na poslane instrukcijske poruke prilikom testiranja . . . . .	45

# Poglavlje 1

## Uvod

Industrijski sistem za vizuelnu inspekciju predstavlja kompleksno okruženje koje se sastoji od elektroničke, optičke i rasvjetne opreme. Daleko najskuplja i najvažnija komponenta tog sistema jeste procesor slike i kao rezultat takvog poretka, većina razmatranja prilikom projektovanja sistema se fokusiraju na tu komponentu, često zanemarujući druge aspekte sistema [2].

Jedan od tih zanemarenih aspekata, iako često odlučujući faktor za uspjeh aplikacija za obradu slike, jeste osvjetljenje (iluminacija) okruženja inspekcije. Kako bi se ostvarili najbolji rezultati i smanjilo vrijeme potrebno za naknadnu obradu slike, neophodno je osigurati određeni kvalitet i kvantitet svjetlosti, a to se u praksi postiže pomoću kontrolabilnih modula za osvjetljenje.

Općenito postoje tri ključna aspekta prilikom projektovanja sistema iluminacije [3]: prvo, razumijevanje uticaja komponente osvjetljenja u vizijskim aplikacijama; drugo, poznavanje načina interakcije svjetlosti i date površine, tj. geometrije i boje; i konačno, upoznavanje sa dostupnim tehnikama i tehnologijama iluminacije koje će osigurati uspješnu ekstrakciju relevantnih karakteristika analiziranog objekta. Kvalitetnim projektovanjem i praćenjem određenih koraka u analizi osvjetljenja, osiguravamo konzistentno i robusno okruženje, maksimizirajući vrijeme, trud i resurse utrošene na ostale aspekte dizajna, testiranja i implementacije sistema vizuelne inspekcije.

Ovaj završni rad je struktuiran tako da svojim poglavljima pokrije sve aspekte projektovanja jednog funkcionalnog i robusnog sistema osvjetljenja.

U prvom poglavlju opisane su neke od aktuelnih tehnika i tehnologija osvjetljavanja. Dat je kratak uvod u upravljanje LED osvjetljenjem i predstavljeno je nekoliko komercijalnih upravljačkih jedinica.

U drugom poglavlju opisan je dizajn hardverskog dijela modula. Opisane su moguće konfiguracije izlaznih kanala, 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 obuhvata softverski dio sistema. U ovom poglavlju objašnjeno je korištenje odgovarajućih mogućnosti mikrokontrolera potrebnih pri radu modula, te komunikacijski protokol.

U četvrtom poglavlju navedeni su načini rada i rezultati testiranja modula, dok je na kraju rada dat odgovarajući zaključak.

## 1.1 Tehnologije i tehnike osvjetljavanja

Historijski gledano, osvjetljenje je često posljednji specificiran, razvijan i/ili finansiran, a ponekad i u potpunosti zanemaren aspekt. Ovakav pristup nije u potpunosti ni neopravdan, obzirom da donedavno na tržištu nije bilo realnih opcija koje bi odgovarale specificiranom problemu, već su se rješenja osvjetljenja uglavnom sastojala od standardnih inkadescentnih ili fluorescentnih proizvoda opće namjene, uz doprinos raznih elemenata ambijentalne svjetlosti.

Do današnjeg dana bilo je određenih napredaka u ovoj oblasti, te u cilju što boljeg pregleda trenutnog stanja, u nastavku će biti pobrojane, objašnjene i kritički sagledane mnoge aktuelne tehnologije i tehnike osvjetljavanja.

### 1.1.1 Vrste izvora svjetlosti i spektralni sadržaj

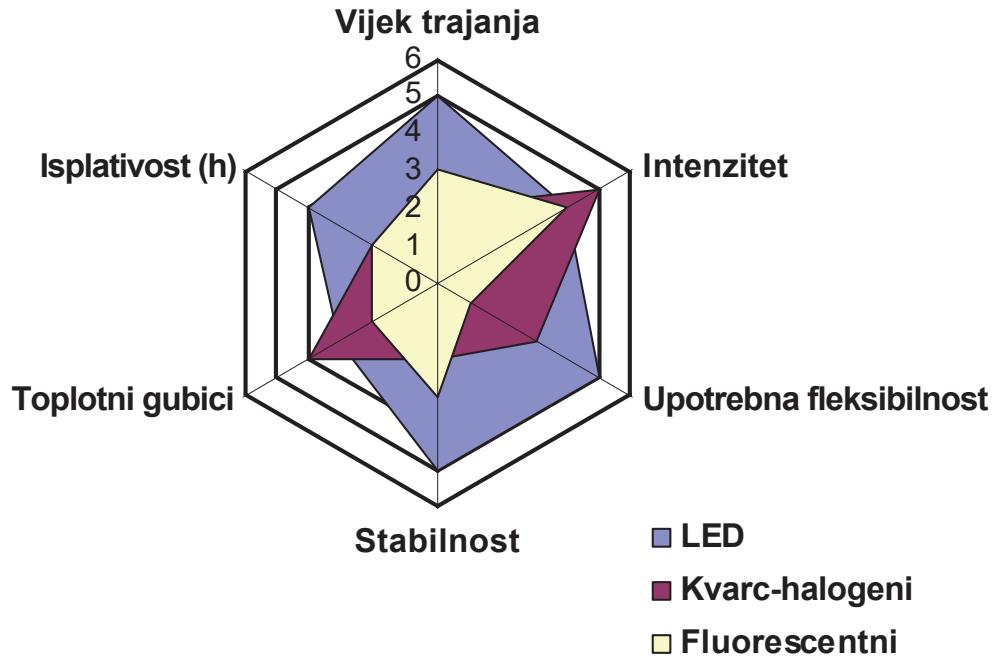
Za potrebe sistema vizuelne inspekcije, trenutno najkorišteniji izvori svjetlosti su [4]:

- fluorescentni,
- kvarc-halogeni,
- LED – svjetlo-emitirajuće diode (eng. *Light-Emitting Diode*),
- metal-halidni (živa),
- ksenonski i
- HPS – natrijski (eng. *High Pressure Sodium*).

Fluorescentni, kvarc-halogeni i LED su daleko najšire korišteni tipovi osvjetljenja u mašinskoj viziji, naročito za manje i srednje veličine inspekcijskih stanica, dok se metal-halidni, ksenonski i HPS izvori tipično koriste u aplikacijama većih razmjera ili u područjima gdje je potreban izvor sa značajno većim intezitetom svjetlosti. Metal-halidni svjetlosni izvori, također poznati pod nazivom *živine svjetiljke*, se često koriste u mikroskopiji iz razloga što posjeduju više diskretnih vrhova (eng. *peaks*) valnih dužina što je komplementarno upotrebi filtera u izučavanju fluorescencije. Ksenonski izvori su korisni u aplikacijama gdje postoji potreba za jako intenzivnim pulsirajućim osvjetljenjem. Slika 1.1 prikazuje prednosti i nedostatke fluorescentnih, kvarc-halogenih i LED tipova osvjetljenja i kriterije izbora relevantne za potrebe mašinske vizije [5]. Naprimjer, iako LED osvjetljenje ima duži vijek trajanja, kvarc-halogeni izvori mogu biti dobar izbor za određenu aplikaciju jer nude veći intenzitet emisije.

U prošlosti, fluorescentni i kvarc-halogeni izvori su bili najčešće korišteni. U posljednje vrijeme, LED tehnologija je uznapredovala po pitanju stabilnosti, intenziteta i isplativosti, dok unatoč tim naprecima još uvijek nije dostigla stepen isplativosti za razvoj sistema velikih razmjera, naročito u uporedbi sa fluorescentnom tehnologijom. Međutim, s druge strane, ukoliko su upotrebna fleksibilnost, izlazna stabilnost i dugotrajnost bitni parametri, onda je LED tehnologija ipak svrshodnija.

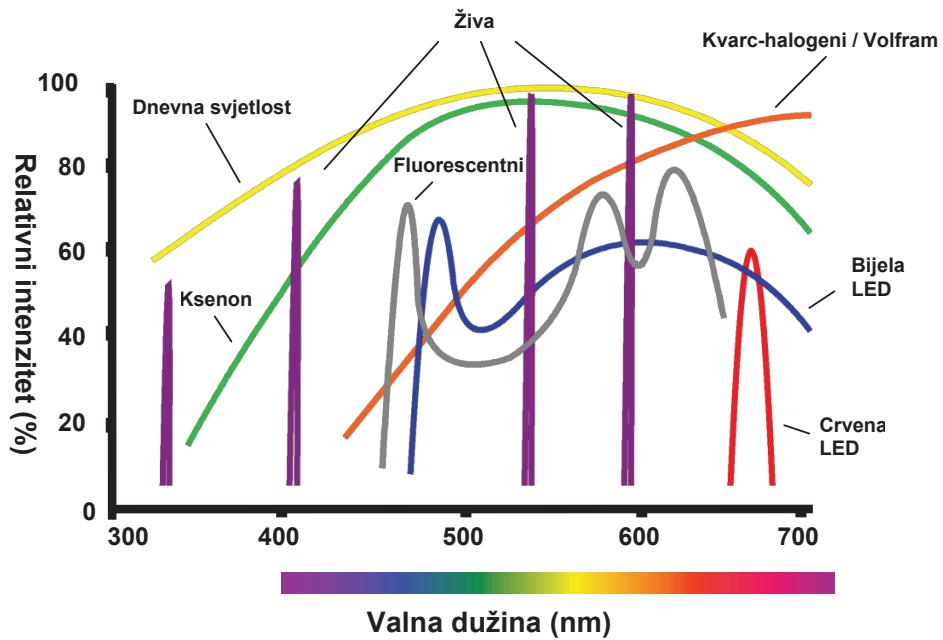
Zavisno od egzaktnih zahtjeva na osvjetljenje, često za specifičnu implementaciju može biti korišteno više od jednog tipa izvora i većina eksperta u polju mašinske vizije se slaže da samo jedan tip izvora svjetlosti ne može adekvatno riješiti sve probleme vezane za osvjetljenje.



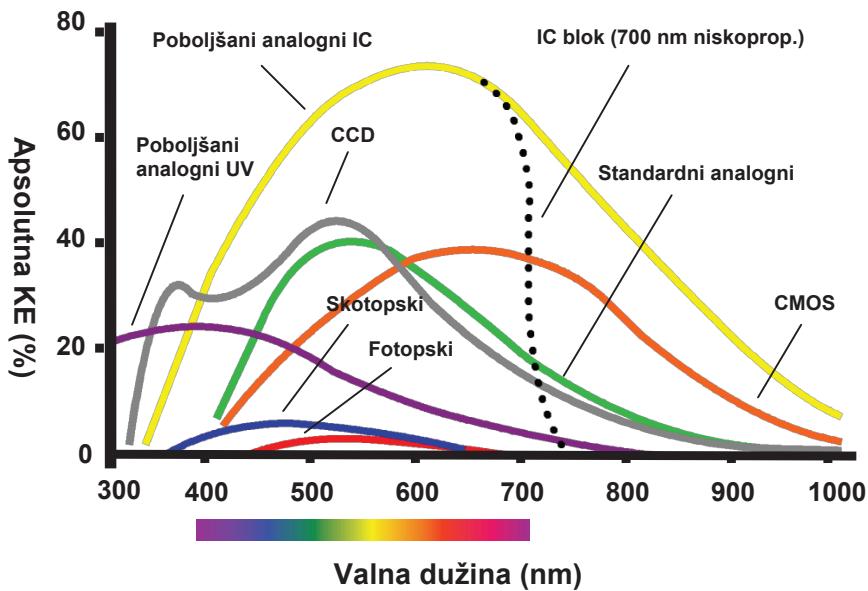
Slika 1.1: Komparacija najčešće korištenih izvora svjetlosti

Jako je bitno, pored jačine izvora, razmotriti i njegov spektralni sadržaj (slika 1.2) [5]. Mikroskopske aplikacije, naprimjer, koriste potpuno-spektralne kvarc-halogene, ksenonske ili metal-halidne izvore, naročito ukoliko se radi sa slikom u boji. Međutim, isto tako su monohromatski LED izvori korisni za crno-bijele CCD kamere, a od skorijeg vremena i za aplikacije u boji, uzimajući u obzir pojavu "svebojnih" RGB (eng. *Red-Green-Blue*) i bijelih LED rasvjetnih tijela.

U aplikacijama koje zahtijevaju veliki intenzitet svjetlosti, kao što su inspekcije pri velikim brzinama, može biti korisno prilagoditi spektralni učinak izvora spektralnoj osjetljivosti kamere koja se koristi (slika 1.3) [5]. Naprimjer, kamere bazirane na CMOS senzorima su daleko osjetljivije na infracrvenu svjetlost, nego li kamere bazirane na CCD tehnologiji, te na taj način imaju prednost u okolnostima sa oskudnim osvjetljenjem kada se koriste infracrveni LED ili užareni (volfram) izvori bogati infracrvenom svjetlošću.



**Slika 1.2:** Relativni intenzitet izvora svjetlosti u zavisnosti od valne dužine



**Slika 1.3:** Apsolutna kuantna efikasnost razlicitih tipova senzora u zavisnosti od valne duzine

Analizom karakteristika na slikama 1.2 i 1.3, mogu se zaključiti sljedeće činjenice važne za izbor tehnologije rasvjetnog tijela:

- Potrebno je uskladiti vršne osjetljivost senzora kamere sa vršnom valnom dužinom izvora kako bi učinkovitost bila maksimalna.
- Izvori sa uskim spektralnim pojasom, kao što su monohromatski LED ili metal-halidni,

su djelotvorni pri propuštanju strateških valnih dužina kada se koriste sa odgovarajućim uskopropusnim filtrima. Naprimjer, filter propusnik crvenog 660nm pojasa zajedno sa odgovarajućim crvenim LED svjetlom je veoma efikasan pri blokiranju ambijentalne svjetlosti iz okolnih fluorescentnih ili živinih izvora.

- Sunčeva svjetlost ima uglavnom konzistentan spektralni sadržaj na širem rasponu valnih dužina, stoga bitno utiče na rezultate vizuelne inspekcije. Iz navedenog razloga je potrebno koristiti određene metode zaštite od sunčeve svjetlosti, što je često realizirano u vidu svjetlo-nepropusnog kućišta.

### 1.1.2 Tehnike osvjetljavanja

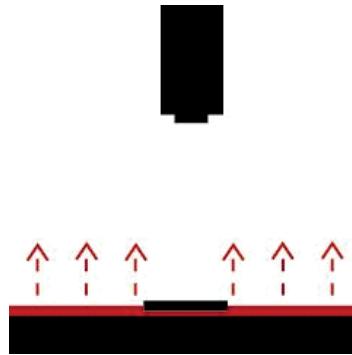
Tehnike osvjetljavanja uključuju sljedeće [5]:

- *pozadinsko osvjetljenje* (eng. *Back Lighting*),
- *prigušeno osvjetljenje*, odnosno, *potpuno osvjetljeno polje* (eng. *Diffuse Lighting*, odnosno, *Full Bright Field Lighting*),
- *direkciono osvjetljenje*, odnosno, *djelimično osvjetljeno polje* (eng. *Directional Lighting*, odnosno, *Partial Bright Field Lighting*) i
- *osvjetljenje sa tamnim poljem* (eng. *Dark Field Lighting*).

Upotreba nekih od ovih tehnika zahtjeva specifičnu svjetlost i geometriju ili relativni položaj kamere, uzorka i rasvjetnog tijela, dok druge tehnike isto ne zahtijevaju. Mnogi proizvođači rasvjetnih proizvoda za vizijske sisteme, između ostalog, nude i rasvjetna tijela koja inkorporiraju različite kombinacije pobrojanih tehnika u sklopu jednog uređaja, barem ukoliko se u obzir uzmu LED bazirani uređaji, gdje svaka od pobrojanih tehnika može biti pojedinačno adresirana. Ova činjenica omogućuje veću fleksibilnost i također smanjuje potencijalne troškove tako što više različitih aplikacija može biti zadovoljeno jednom stanicom, nego li sa više njih.

#### Pozadinsko osvjetljenje

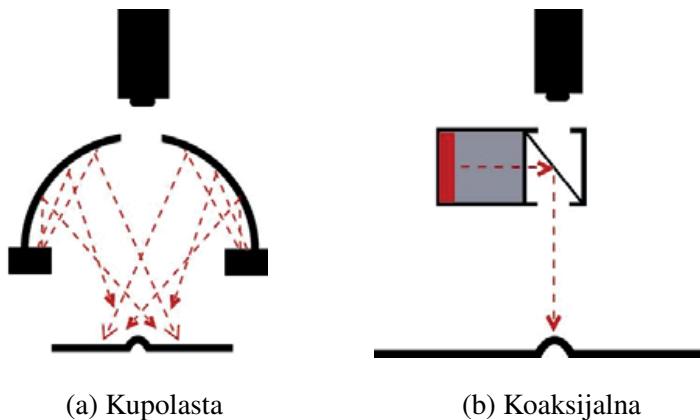
Pozadinsko osvjetljenje generira trenutni kontrast tako što kreira tamne siluete nasuprot svijetle pozadine (slika 1.4). Najčeća primjena ove tehnike se ogleda u detekciji prisustva/odsustva rupa i šupljina, pozicioniranju dijelova i orientaciji istih ili u mjerenu dimenzija objekata. Često je korisno koristiti monohromatsku svjetlost, kao što je crvena, zelena ili plava, sa određenom mjerom polarizacije ukoliko je potrebna veoma precizna detekcija ivice.



**Slika 1.4:** Pozadinsko osvjetljenje

### Prigušeno osvjetljenje (potpuno osvjetljeno polje)

Prigušeno osvjetljenje se najčešće koristi na sjajnim reflektirajućim površinama ili na uzorcima djelimične reflektivnosti gdje je potrebno osigurati ravnomjerno multidirekciono osvjetljenje. Generalno postoji više implementacija prigušenog osvjetljenja, ali dvije implementacije se najčešće pojavljuju u praksi: hemisferna kupolasta/cilindrična (slika 1.5a) i koaksijalna (slika 1.5b).



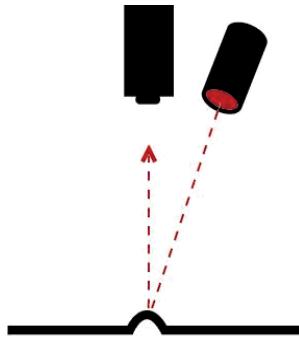
**Slika 1.5:** Različite implementacije prigušenog osvjetljenja

Kupolasta implementacija je veoma efektivna pri osvjetljavanju zakriviljениh, sjajnih površina, kakve su naprimjer, često prisutne u automobilskoj industriji. Koaksijalna implementacija funkcioniра na sličan način za ravne površine i naročito je efektivna pri naglašavanju diferencijalno nagnutih, teksturiranih ili topografskih odlika na relativno ravnim objektima. U cilju efikasnosti prigušenih izvora svjetlosti, naročito kupolaste implementacije, potrebno je osigurati blizinu uzorka.

### Direkciono osvjetljenje (djelimično osvjetljeno polje)

Direkciono osvjetljenje je najčešće korištena tehnika osvjetljavanja u vizijskim sistemima i također je najpoznatiji tip osvjetljenja koji se koristi svakodnevno, uključujući i sunčevu svjetlost. Ovaj tip osvjetljenja se razlikuje od prigušenog osvjetljenja po tome što je usmjeren, tipično izvirući iz tačkastog izvora i zbog svoje direkcione prirode, dobar je izbor za generisanje kontrasta

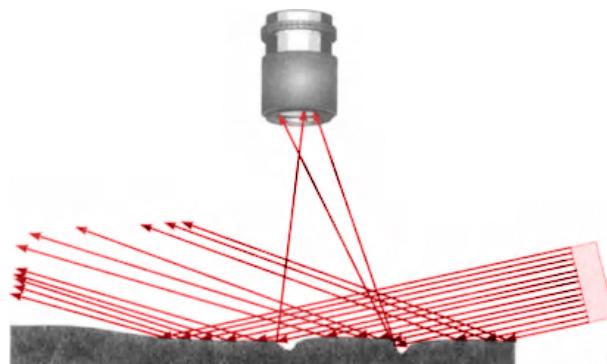
i naglašavanje topografskih detalja. Međutim, manje je efektivan kada se koristi *koaksijalno* na sjajnim površinama, iz razloga što dolazi do generisanja poznatog *odbljeska* (eng. *hotspot reflection*).



**Slika 1.6:** Direkciono osvjetljenje

### Osvjetljenje sa tamnim poljem

Ovaj tip osvjetljenja je okarakterisan malim ili umjerenim uglom incidencije svjetlosti (figura 1.7) i tipično zahtijeva blizinu objekta za kvalitetne rezultate inspekcije. Glatka reflektirajuća površina neće reflektovati svjetlost u kameru, te će zbog toga površina izgledati tamno. Za razliku od toga, neravnoće na površini će izgledati svjetlo iz razloga što će ivice orijentisane prema izvoru svjetlosti reflektovati svjetlost u kameru. Površine sa jako izraženim teksturama izgledaju svjetlo u ovakvoj konfiguraciji osvjetljenja, pošto sadrže mnoštvo mikroskopskih indentacija koje rasipaju incidentne zrake, tako da neke od njih dospijevaju do sočiva kamere. Na takvoj površini, suprotne ivice istih indentacija stvaraju sjene. Stoga, nezavisno od strukture površine, ova tehnika iluminacije će uzrokovati znatno drugačiju prezentaciju površine i njezinih karakteristika.



**Slika 1.7:** Osvjetljenje sa tamnim poljem

## 1.2 Upravljanje osvjetljenjem

Pored konkretnih tehnika i tehnologija, bitnu ulogu unutar jednog robusnog sistema osvjetljenja igra i sama upravljačka jedinica. Obzirom da se na osnovu ranije izloženog može zaključiti da je za potrebe ovog rada najprikladnija LED tehnologija, tako će se u nastavku dati kraći pregled mogućih načina upravljanja LED osvjetljenjem.

Upravljanje LED osvjetljenjem ostvaruje se pomoću LED kontrolerā i oni predstavljaju esencijalni element bilo kojeg vizujskog sistema u kojem je potrebno optimizirati intenzitet osvjetljenja i preciznost sekvencije okidanja između kamere i rasvjete.

Korištenjem kontrolera omogućuje se:

- upravljanje impulsima (eng. *Pulse*), odnosno, bljeskanjem (eng. *Strobe*) - gdje je potrebna sinhronizacija vremena aktivnosti osvjetljenja sa kamerom i cilnjim objektom (sa rezolucijama i u domenu nanosekundi),
- preopterećavanje (eng. *Overdriving*) - gdje je potreban pojačan intenzitet svjetlosti unutar kratkih i definisanih vremenskih intervala,
- konstantan strujni izvor - gdje je potreban visoko stabilan izvor konstantne struje za napajanje LED komponenti,
- upravljanje kompozicijama osvjetljenja (eng. *Multi-lighting*) - za sisteme sa konfiguracijama koje uključuju više rasvjetnih tijela koji zahtijevaju upravljanje intenzitetom i sinhronizaciju pri velikim brzinama, sa jednim ili više okidača,
- daljinske izmjene konfiguracije - za sisteme gdje je korisno imati mogućnost daljinskog podešavanja sistemskih parametara.

LED su poluprovodnički uređaji čiji je svjetlosni izlaz direktni rezultat protjecanja struje, a ne napona, te zbog toga svi proizvođači LED komponenti preporučuju implementaciju nekog vida upravljanja strujom, kako bi se osigurala efikasnost. Upravljanje strujom omogućuje precizniju kontrolu nad izlazom LED osvjetljenja.

U cilju ostvarivanja boljeg uvida u trenutnu ponudu komercijalnih rješenja u oblasti upravljanja osvjetljenjem, u nastavku će biti dat kratak osvrt na nekoliko konkurentnih proizvoda.

### 1.2.1 *Triniti* LED kontroleri

Kontroleri iz *TR* serije LED kontrolera su osnovna komponenta *Triniti* platforme za inteligentne sisteme osvjetljenja britanskog proizvođača *Gardasoft*. Fizički prikaz tri različita modela iz ove serije dat je na slici 1.8, dok su osnovne karakteristike dva modela kontrolera prikazane u tabeli 1.1 [6].

**Tabela 1.1:** Osnovne karakteristike LED kontrolera TR-RT220 i TR-RT420

Model	TR-RT220	TR-RT420
Komunikacija	Ethernet	
Izlazni kanali	Dva nezavisna strujna izlaza sa konst. strujom	Četiri nezavisna strujna izlaza sa konst. strujom
Izlazna struja	Do 3 A po kanalu kontinualno ili 20 A po kanalu pulsirano	
Izlazna snaga	Maks. 40 W po jedinici	Maks. 50 W po jedinici
Okidački ulaz	Dva opto-izolirana dig. ulaza, 3-24 V	Četiri opto-izolirana dig. ulaza, 3-24 V
Trajanje impulsa	20 $\mu$ s do 1 s	
Vrijeme odgode impulsa	20 $\mu$ s do 1 s	
Prekidačko kašnjenje	Maks. 20 $\mu$ s	
Frekvencija okidanja	Maks. 1 kHz	
Izlazni napon	0 V do 46 V	
Napon napajanja	24-48 VDC	

**Slika 1.8:** Fizički izgled Triniti LED kontrolera

### 1.2.2 ULC-2 LED kontroler

*Universal LED Controller (ULC-2)*, američkog proizvođača *Metaphase Technologies*, je sveobuhvatni upravljački uređaj dizajniran da bude kompatibilan sa standardnim LED osvetljenjem. Fizički prikaz kontrolera dat je na slici 1.9, dok su osnovne karakteristike prikazane u tabeli 1.2 [7].

**Tabela 1.2:** Osnovne karakteristike LED kontrolera ULC-2

<b>Model</b>	<b>ULC-2</b>
<b>Komunikacija</b>	Ethernet i USB
<b>Izlazni kanali</b>	Dva nezavisna strujna izlaza sa konst. strujom
<b>Izlazna struja</b>	Do 4 A po kanalu kontinualno ili 40 A po kanalu pulsirano
<b>Izlazna snaga</b>	Maks. 60 W po kanalu
<b>Okidački ulaz</b>	Dva opto-izolirana digitalna ulaza, 3-30 V
<b>Trajanje impulsa</b>	2 $\mu$ s do 60 ms
<b>Vrijeme odgode impulsa</b>	6 $\mu$ s do 60 ms
<b>Prekidačko kašnjenje</b>	Maks. 6 $\mu$ s
<b>Frekvencija okidanja</b>	Maks. 50 kHz
<b>Izlazni napon</b>	0 V do 60 V
<b>Napon napajanja</b>	24 VDC

**Slika 1.9:** Fizički izgled ULC-2 LED kontrolera

### 1.2.3 DCS-100E LED kontroler

Kontroler *DCS-100E*, američkog proizvođača *Advanced Illumination*, je snažni LED kontroler u kompaktnom kućištu. Fizički prikaz kontrolera dat je na slici 1.10, dok su osnovne karakteristike prikazane u tabeli 1.3 [8].

**Tabela 1.3:** Osnovne karakteristike LED kontrolera DCS-100E

<b>Model</b>	<b>DCS-100E</b>
<b>Komunikacija</b>	Ethernet
<b>Izlazni kanali</b>	Tri nezavisna strujna izlaza sa konst. strujom
<b>Izlazna struja</b>	Do 4.5 A po kanalu kontinualno ili 45 A po kanalu pulsirano
<b>Izlazna snaga</b>	Maks. 30 W po kanalu
<b>Okidački ulaz</b>	Tri opto-izolirana digitalna ulaza, 5-30 V
<b>Trajanje impulsa</b>	10 $\mu$ s do 65 ms
<b>Vrijeme odgode impulsa</b>	1 $\mu$ s do 10 ms
<b>Prekidačko kašnjenje</b>	Maks. 10 $\mu$ s
<b>Frekvencija okidanja</b>	Maks. 2 kHz
<b>Izlazni napon</b>	0 V do 50 V
<b>Napon napajanja</b>	24 VDC

**Slika 1.10:** Fizički izgled DCS-100E LED kontrolera

### 1.3 Razvoj fleksibilnog modula za osvjetljenje

Na osnovu podataka ponuđenih u tabelama 1.1, 1.2 i 1.3 može se zaključiti da prethodno opisani kontroleri imaju prednost u aplikacijama gdje su potrebne velike brzine uzorkovanja i značajni iznosi jačine izlazne struje, dok im je glavni nedostatak ograničenost na dva, tri, odnosno, četiri nezavisna izlazna kanala, što u nekim konkretnim slučajevima upotrebe može biti nedovoljno. Ovi nedostaci daju opravdanje pokušaju da se projektuje jedan fleksibilniji sistem osvjetljenja sa većom kontrolom nad izlaznim funkcionalnostima, što i jeste osnovna motivacija ovog završnog rada.

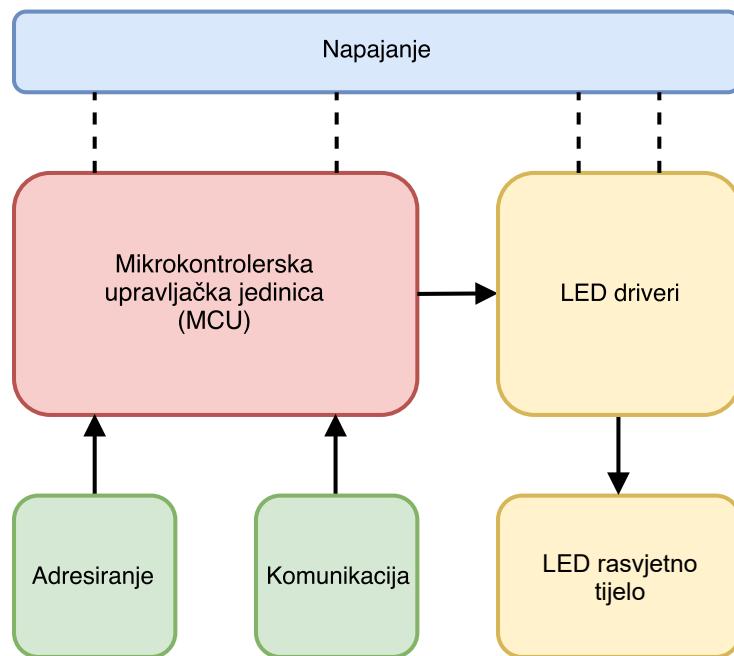
Modul osvjetljenja koji se razvija treba da ima kontrolersku jedinicu, ali i rasvjetno tijelo kojim bi se upravljalo. Kontroler treba da bude mikrokontrolerski bazirani sistem sa omogućenom komunikacijom putem sabirnice koja odgovara industrijskim standardima. Funkcionalnosti kontrolera treba poopćiti tako da u dovoljnoj mjeri ne ovise o obliku, veličini i snazi rasvjetnih tijela koja se povezuju na kontroler. Akcije poput uključivanja, odnosno, isključivanja rasvjetnog tijela, izmjene parametara modula, kao što su intenzitet svjetlosti, vremena trajanja impulsa i sl., treba da se obavljaju na zahtjev koji generiše računarski program i koji se modulu dostavlja putem komunikacionog kanala.

U narednim poglavljima rada bit će opisani svi pomenuti aspekti razvoja jednog fleksibilnog modula za osvjetljenje.

## Poglavlje 2

# Hardverska komponenta modula za osvjetljenje

U ovom poglavlju će biti predstavljeni načini realizacije hardverskog dijela modula za osvjetljenje, a zatim će se odabrat odgovarajuća rješenja za konkretnu realizaciju. Osnovni funkcionalni blokovi modula su prikazani na slici 2.1. Dijelovi modula opisani su u nastavku<sup>1</sup>.



Slika 2.1: Blok struktura modula

<sup>1</sup>Dizajn i implementacija dijela sa mikrokontrolerom, te dijela sa komunikacijom i adresiranjem realizovani su u saradnji sa kolegom Jasminom Hadžajlićem, te su sadržani i u završnom radu "Dizajn ekspanzionog I/O modula za PC".

## 2.1 Mikrokontroler

Mikrokontroler (skraćeno MCU od eng. *microcontroller unit*) je računar malih dimenzija implementiran na jednom integriranom kolu (*čipu*) [9]. U modernoj tehnologiji, to su sistemi na čipu – SoC (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 ferolektrične 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 računarima 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 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 registra 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 (*miliona 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):

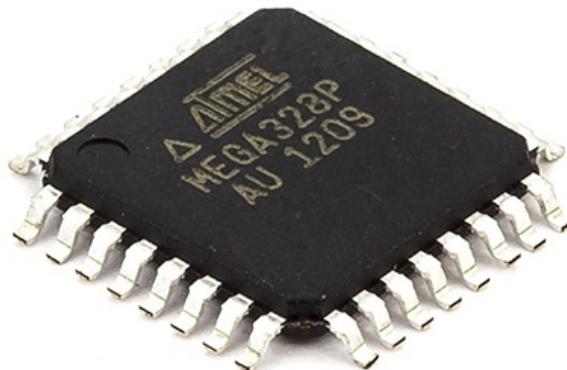
- 28-pin PDIP,
- 28-pin MLF,
- 32-pin TQFP i
- 32-pin MLF.

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

### 2.1.2 Funkcionalni moduli i mogućnosti mikrokontrolera ATmega328P

Mikrokontroler ATmega328P posjeduje sljedeće funkcionalnosti:

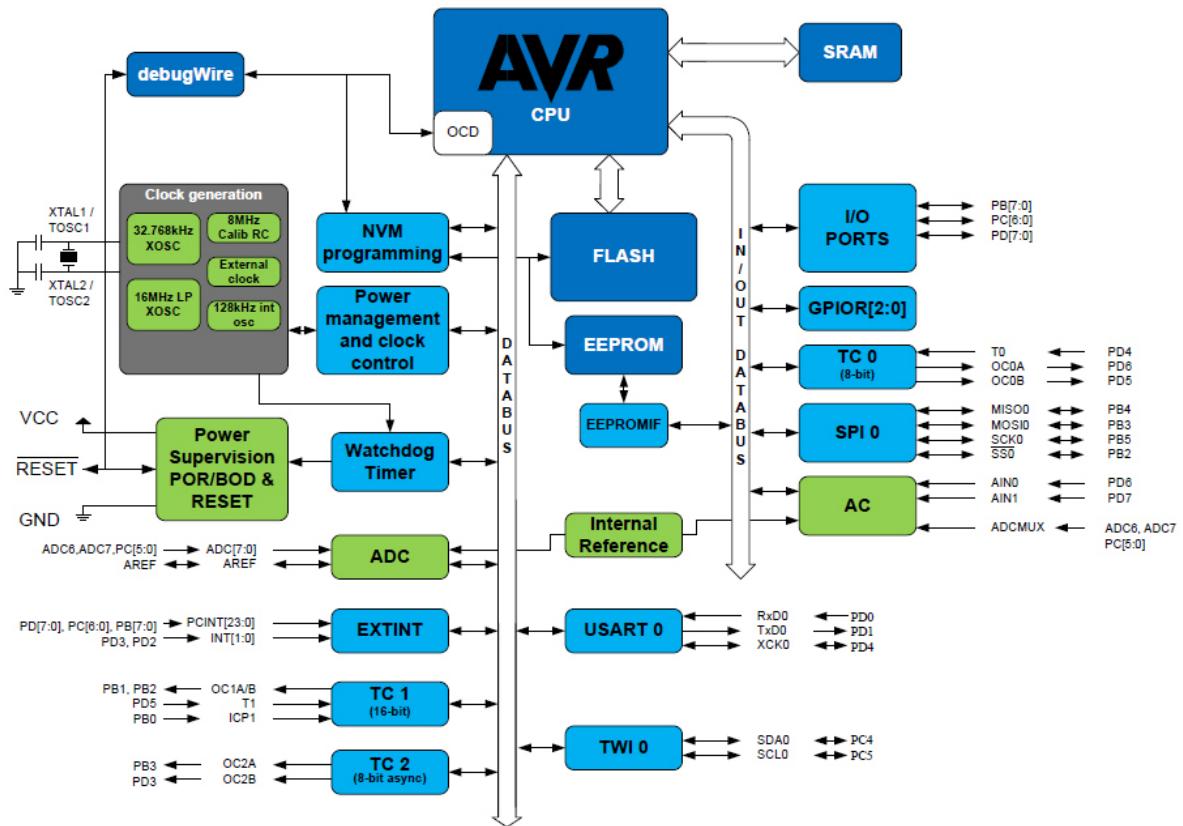
- 32KB sistemski programabilne Flash memorije sa Čitaj-Dok-Pišeš (eng. *Read-While-Write*) sposobnostima,
- 1KB EEPROM memorije,
- 23 I/O linije opće namjene,



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

- 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-orientirani 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.



Slika 2.3: Blok dijagram mikrokontrolera ATmega328P [1]

### 2.1.3 Konfiguracija pinova mikrokontrolera ATmega328P (32-pin TQFP)

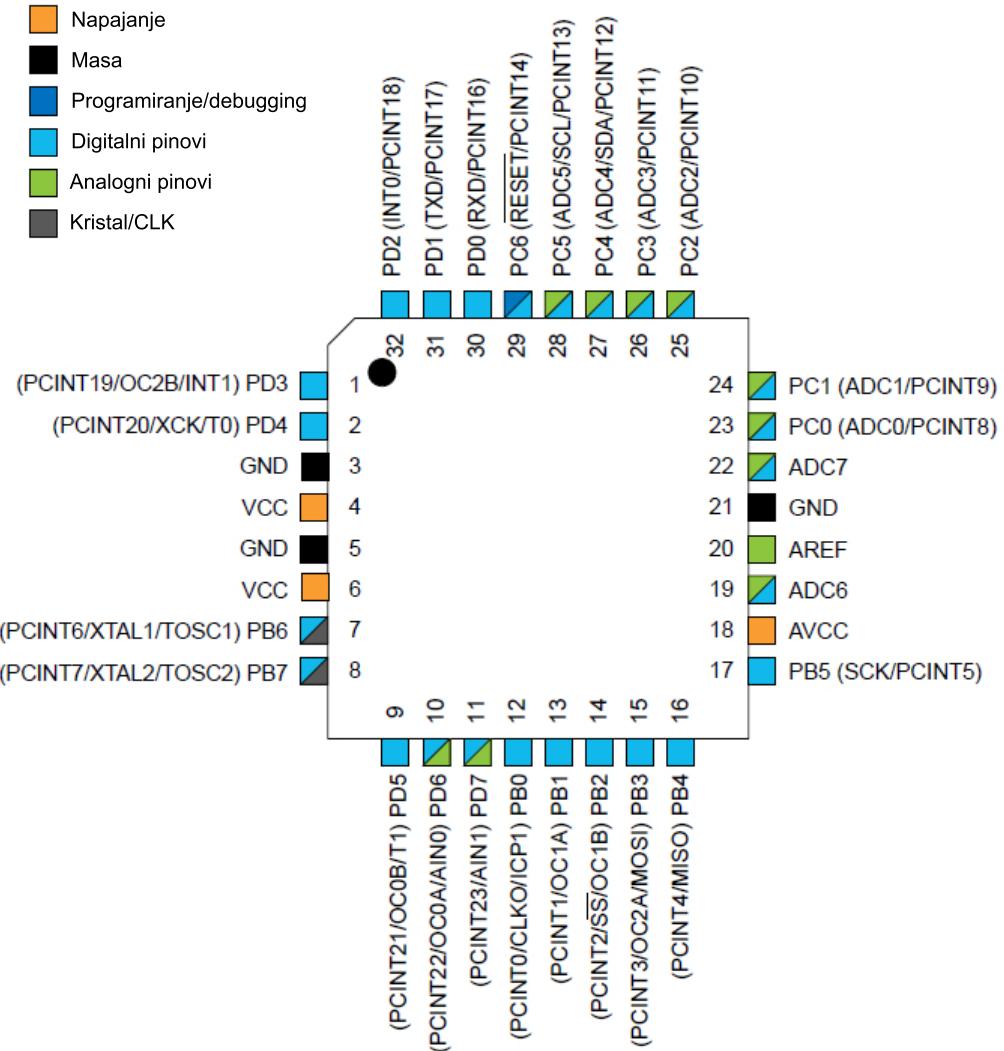
Verzija mikrokontrolera razmatrana u ovom završnom radu (32-pin TQFP) ima 32 pina od kojih je većina multipleksirana, pa jedan pin može da obavlja više uloga u zavisnosti od aplikacije. Na slici 2.4 prikazan je raspored pinova sa odgovarajućim multipleksiranim funkcijama istih, a njihov opis dat je u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

**VCC:** Napon napajanja.

**GND:** Masa.

**Port B (PB[7:0]) XTAL1/XTAL2/TOSC1/TOSC2:** Port B je 8-bitni bidirekcionni 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 aktiviran, 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.

**Port C (PC[5:0]):** Port C je 7-bitni bidirekcionni I/O port sa internim pull-up otpornicima (odabranim za svaki bit). Pinovi porta C prelaze u stanje visoke impedanse kada neki od uslova



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

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 (odabaranim 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 eks- terno 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.

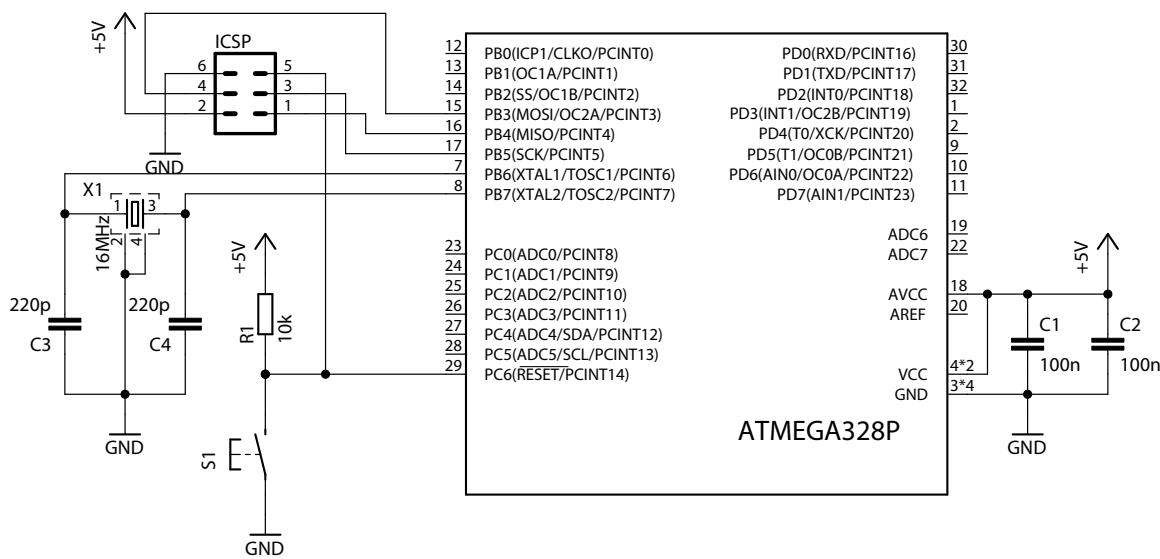
**ADC[7:6]:** ADC[7:6] služe kao analogni 10-bitni ulazni kanali u AD konvertor.

## 2.1.4 Minimalno električko okruženje mikrokontrolera ATmega328p

Za pravilan rad mikrokontrolera ATmega328P, potrebno je osigurati odgovarajuće električko okruženje, što podrazumijeva stabilan napon napajanja sa adekvatnim naponskim nivoima, kolo vanjskog oscilatora, kolo za resetovanje mikrokontrolera, te ISP interfejs za programiranje [10].

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 električko okruženje mikrokontrolera ATmega328P

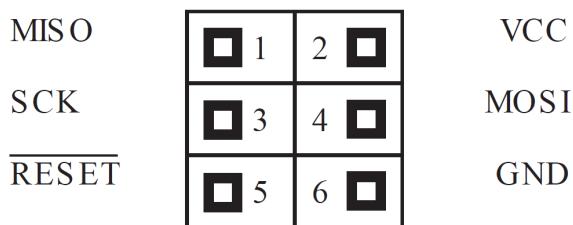
## 2.1.5 Programiranje programske memorije - ISP

ISP programiranje (eng. *In-System Programming*), također poznato i kao ICSP (eng. *In-Circuit Serial Programming*), omogućuje programiranje i reprogramiranje programske memorije mi-

krokontrolera 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 odstranjuvanjem 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*.



**Slika 2.6:** Raspored pinova ISP konektora

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

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 [11]. 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 sekvencialno 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) [11].

Prema smjeru toka podataka komunikaciju možemo podijeliti na tri vrste [11]:

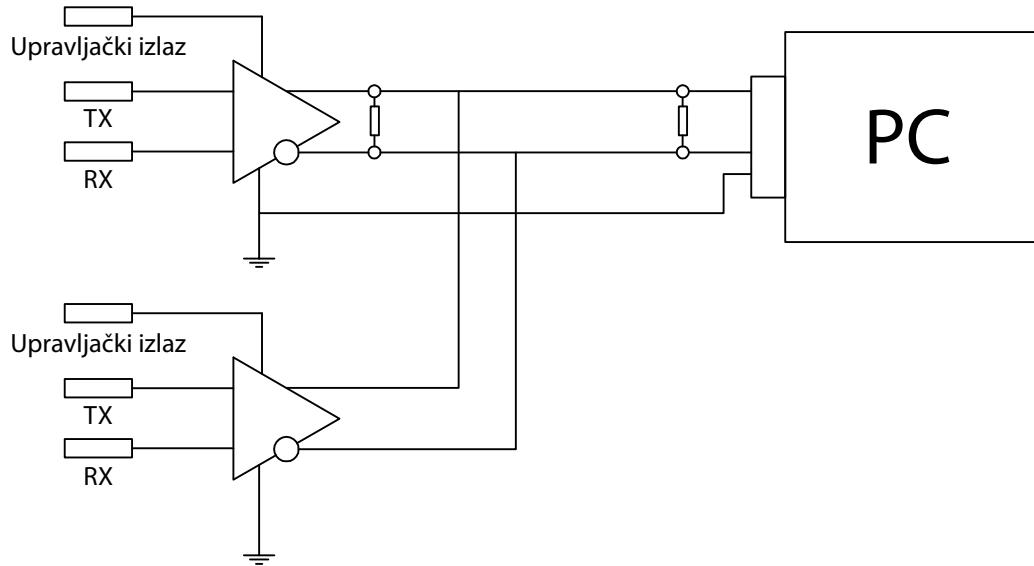
- *Simplex* – komunikacija između učesnika se uvijek obavlja u istom smjeru. Jedan uređaj samo šalje, a drugi samo prima podatke.
- *Half-duplex* – komunikacija je dvosmjerna, svi učesnici mogu slati i primati podatke, ali ne istovremeno. Jedan učesnik šalje podatke, a ostali primaju.
- *Full-duplex* – 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 [12]. Detaljnija teoretska razmatranja oko RS485 standarda i drugih aspekata komunikacije, data su na kraju rada u prilogu A.

### 2.2.1 Implementacija RS485 komunikacije

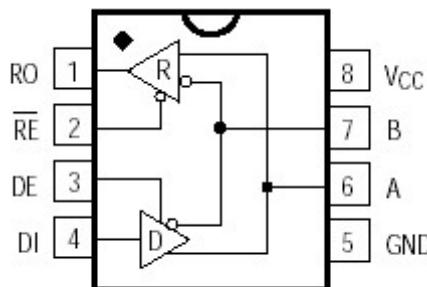
Komunikacija treba da se vrši između glavnog računara i modula sistema, kako je ranije spomenuto. Moduli sistema su bazirani na mikrokontroleru. Mikrokontroler posjeduje pinove za serijsku komunikaciju (RX, TX). Osim ova dva pina, potreban je još jedan izlazni pin, koji određuje u kojem smjeru se vrši 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 2.9 prikazan je elektronički sklop koji realizira RS485 komunikaciju korištenu u sklopu ovog rada i raspored pinova na korištenom RJ-45 konektoru. U nastavku su date osnovne osobine integrisanog kola MAX485, korištenog za implementaciju komunikacije sistema.



**Slika 2.7:** Primjer implementacije RS485 komunikacije

### Integrисано коло MAX485

Integrисана кола MAX485 су примопредајници male snage koji se користе за implementaciju RS485 i RS422 komunikacije. На слици 2.8 је приказан raspored pinova MAX485 IC. Pinovi DE i RE služe за управљање смјером комуникације, pinovi RO i DI služe за примање, односно слanje података. Pinovi A i B су диференцијалне линије RS485 комуникације.



**Slika 2.8:** Raspored pinova MAX485 IC

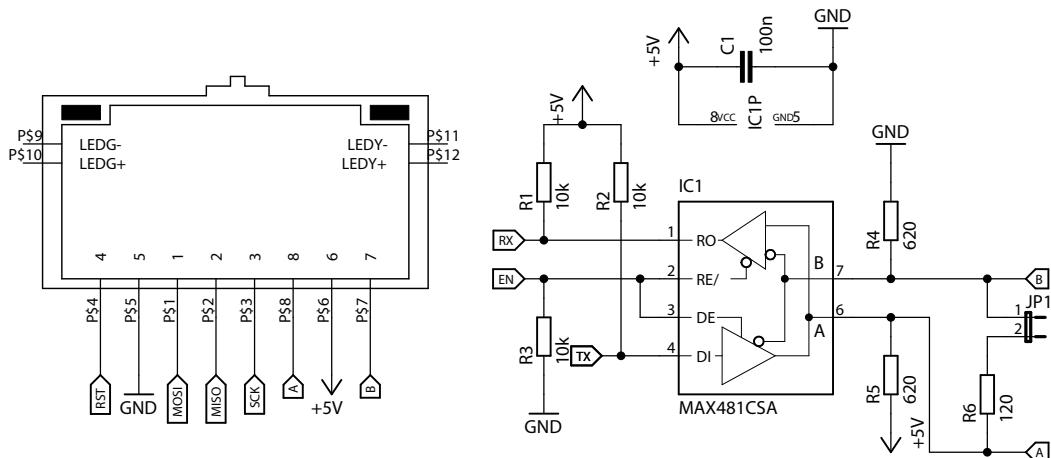
Ukoliko su  $DE$  i  $\overline{RE}$  у стању логичке "0", диференцијални сигнал са A и B се преbacuje на pin RO, одакле приjemник очитава податак. Када су  $DE$  и  $\overline{RE}$  на нивоу логичке "1", податак на pinu DI се проследжује на A и B, и тако поставља на sabirnicu.

Osnovne karakteristike integralnog kola [13]:

- Ulazna otpornost  $R_I = 12k\Omega$
- Napon napajanja  $V_{CC} = 5V$
- Minimalni napon логичке "1" на улазу  $V_{IHmin} = 2V$
- Maksimalni napon логичке "0" на улазу  $V_{ILmax} = 0.8V$
- Brzina prenosa података до 2.5 Mbps

- Half-duplex komunikacija
- Maksimalno 32 korisnika

MAX485 integrисано kolo ima zaštitu od 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.



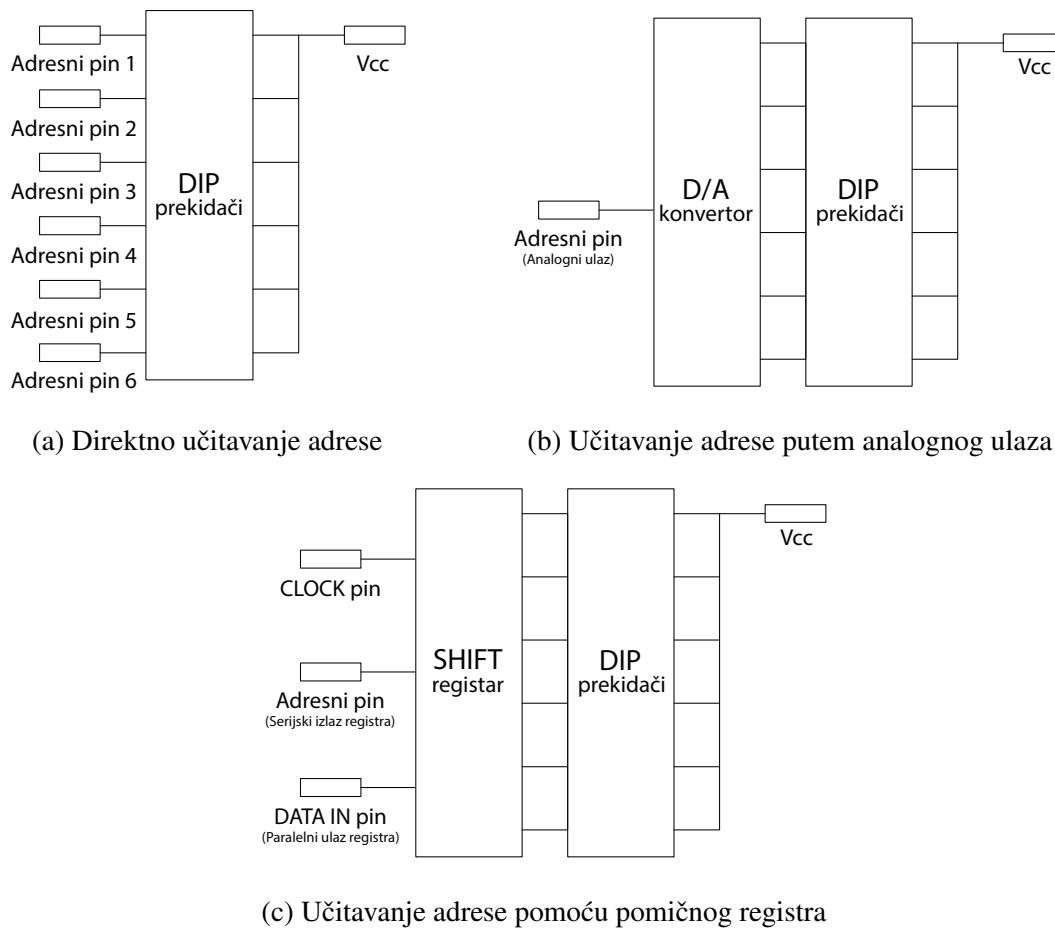
**Slika 2.9:** Shema sklopa za RS485 komunikaciju

## 2.3 Adresiranje modula za osvjetljenje

U *master/slave* komunikaciji *master* uređaj mora znati kome šalje 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.

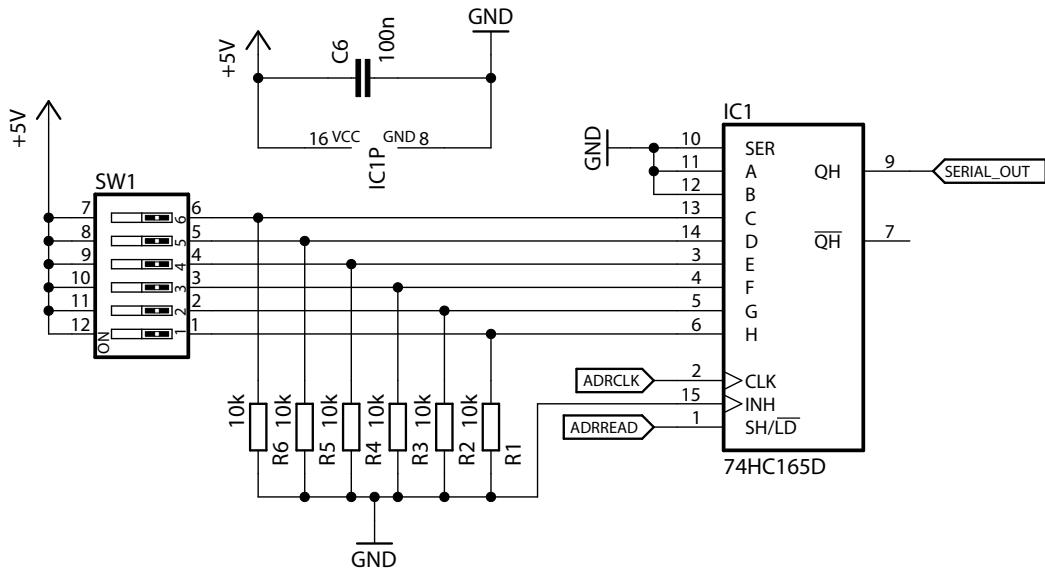
Adresiranje je moguće vršiti tako što se svakom prijemniku softverski unaprijed dodijeli adresa, koja se ne može mijenjati. Osim toga moguće je prijemnicima, uključivanjem jednog po jednog na mrežu, softverski dodjeljivati adrese. Drugi način je komplikovaniji, ali osigurava izbjegavanje preklapanja 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 pomični 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 pomijerenje 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 mikrokontrolera, 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 pomicnog registra i tri digitalna pina mikrokontrolera. Na slici 2.11 prikazana je shema sklopa koji realizira opisani način adresiranja. Kao pomicni registar korišteno je integrисано kolo 74HC165D [14].



Slika 2.11: Shema sklopa za učitavanje adrese uređaja

## 2.4 Rasvjetno tijelo

Kao što je pokazano u uvodnom dijelu rada, veoma bitnu ulogu u jednom sistemu osvjetljenja ima tehnologija osvjetljavanja. Iako je primarni cilj ovog rada sinteza fleksibilnog kontrolera osvjetljenja, radi demonstracije svih funkcionalnosti uređaja bilo je potrebno realizirati rasvjetno tijelo bazirano na korištenoj LED tehnologiji.

### 2.4.1 LED tehnologija

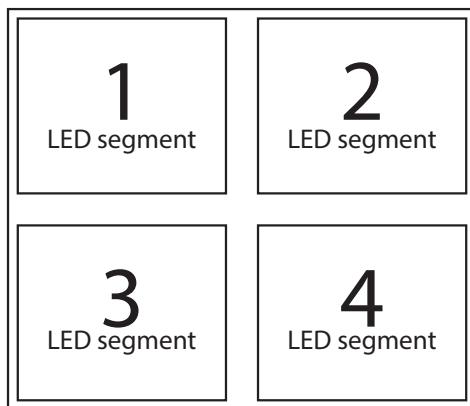
Svjetlo-emitirajuće diode – LED (eng. *Light-Emitting Diodes*) spadaju u poluprovodničke izvore svjetlosti. Kada se dioda pozitivno polarije (uključi), dolazi do rekombinacije elektrona i šupljina u poluprovodniku, pri čemu se oslobođa energija u obliku fotona. Ovaj efekat se zove elektroluminiscencija (eng. *electroluminescence*) i valna dužina emitovane svjetlosti (u skladu sa energijom fotona) je određena širinom *zabranjene zone* poluprovodničkog materijala [15].

LED osvjetljenje predstavlja najbolju opciju kada su u pitanju inspekcijski sistemi manjih razmjera. Sa stanovišta performanse, LED, koje se napajaju istosmjernom strujom, nude visok intenzitet emitovane svjetlosti i veoma ujednačen i stabilan izlaz sa nižim stepenom treperenja, rezultirajući u čišćim i bešumnim inspekcijskim slikama, u odnosu na ostale tehnologije koje se napajaju naizmjeničnom strujom. LED osvjetljenje može biti korišteno za sve potrebe iluminacije uključujući glavno osvjetljenje, pozadinsko osvjetljenje, kao i reflektivno osvjetljenje, a kombinovana rasvjetna tijela se mogu jednostavno upravljati pomoću PC korisničkog interfejsa kako bi se postigla optimalna inspekcija na različitim vrstama podloga.

Pored toga što nude bolje inspekcijske performanse i pouzdanost, LED sistemi zahtijevaju jako malo održavanja. Kvalitetnije LED diode mogu da izdrže i do 100000 radnih sati (11.5 godina), a ukoliko se koristi aktivno hlađenje moguće je maksimizirati radni vijek [16].

### 2.4.2 LED segmenti

Glavna ideja prilikom implementacije rasvjetnog tijela jeste da se tijelo podijeli na više nezavisno upravljenih segmenata. Na slici 2.12 prikazan je raspored sa četiri nezavisna LED segmenta koji je implementiran u nastavku rada.



**Slika 2.12:** Dijagram četverosegmentnog rasporeda rasvjetnog tijela

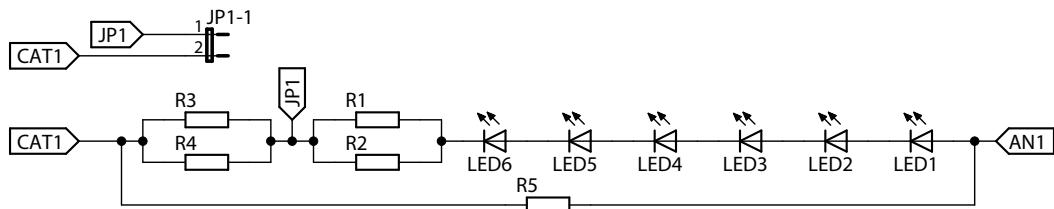
Obzirom da bijele LED diode koje su odabrane za realizaciju ovog rada imaju napon vođenja  $V_f = 3.4V$  pri nominalnoj struci  $I_f = 70mA$  [17], to znači da pri naponu napajanja od +24V koji

je standardan u industrijskim postrojenjima, maksimalni broj LED dioda koje se mogu serijski vezati iznosi  $n = 6$ . Tako će svaki segment biti realiziran u vidu serijske veze šest LED dioda i odgovarajućeg otpornika čija je uloga ograničavanje struje.

Ukoliko se koristi pulsni režim rada osvjetljenja, poželjno je zbog kratkih vremenskih intervala imati mogućnost višestrukog pojačanja intenziteta emitovane svjetlosoti. To se postiže tako što se LED diode do određene mjeri svjesno preoptereće, u smislu povećanja struje, a kako je intenzitet direktno vezan za iznos struje koja protiče kroz diode, tako će doći do srazmernog povećanja intenziteta svjetlosti. Obzirom da je iznos struje koja teče kroz diode ograničena serijskim otporom, povećanje struje je uvjetovano smanjenjem istog. Moguće je implementirati mogućnost selekcije moda rada (normalnog ili preopteretnog) tako što se ukupni otpor podijeli na dva jednaka serijska otpora, a da se korištenjem kratkospojnika prilikom promjene moda ručno prespoji jedan od otpora.

Ukoliko se u obzir uzmu dinamičke osobine LED dioda u pulsnom režimu rada primjetno je veliko vremensko kašnjenje uzrokovano vremenskom konstantom pražnjenja naboja u diodama [18]. To vremensko kašnjenje ograničava maksimalnu frekvenciju PWM signala kojim bi se modulisao intenzitet svjetlosti. Iz tog razloga potrebno je izvršiti kompenzaciju, tj. smanjenje vremenske konstante. To se najlakše postiže dodavanjem paralelnog otpora relativno male vrijednosti preko kojeg se diode mogu prazniti u trenutku isključivanja.

Shema električkog sklopa koji realizira jedan opisani LED segment prikazana je na slici 2.13.



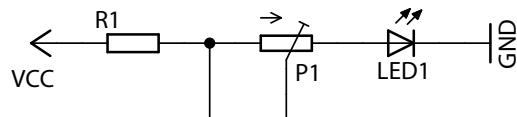
Slika 2.13: Shema spajanja LED segmenta

## 2.5 LED driveri

Cilj ovog rada jeste da se razvije jedan fleksibilni LED kontroler, a ta fleksibilnost se ogleda u povećanoj slobodi nad mogućnostima konfiguracije izlaznih kanala. Prilikom upravljanja LED osvjetljenjem glavni parametri su intenzitet osvjetljenja, trajanje impulsa u pulsnom režimu rada, kao i period pulsiranja ukoliko se radi o periodičnom režimu. U nastavku je data kratka diskusija o nekoliko načina upravljanja osvjetljenjem.

### 2.5.1 Upravljanje LED osvjetljenjem

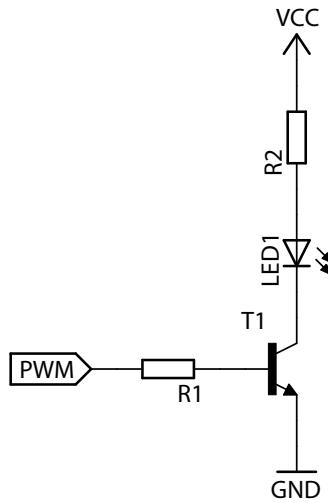
Kao što je pokazano u prethodnom dijelu rada, intenzitet svjetlosti je direktno ovisan o struji koja protiče kroz LED diode. Struja se podešava pomoću serijskog otpora koji ograničava njen iznos. Stoga, način upravljanja intenzitetom svjetlosti koji se po automatizmu nameće jeste korištenje promjenjivog serijskog otpora (reostata). Na slici 2.14 prikazana je električna shema koja realizira opisani način upravljanja. Fiksni otpornik služi kao predotpor koji u slučaju nultog položaja reostata ograničava maksimalnu vrijednost struje kroz diode.



**Slika 2.14:** Upravljanje LED osvjetljenjem pomoću promjenjivog otpora

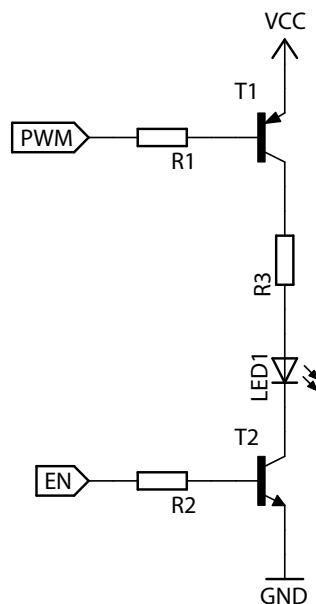
Ovaj način je poprilično jednostavan za implementaciju, ali nažalost ne pokazuje dovoljnu mjeru linearnosti u odnosu između promjene otpornosti i promjene intenziteta svjetlosti, te zahtijeva manuelno podešavanje otpornosti, što nije u skladu sa ciljem elektroničke automatizacije sklopa.

Iz tog razloga se upravljanje intenzitetom svjetlosti najčešće obavlja pomoću impulsno-širinske modulacije – PWM (eng. *Pulse Width Modulation*) i prekidačkih komponenti kao što su tranzistori. Prednosti ovakvog načina upravljanja jesu linearnost intenziteta svjetlosti o *duty cycle*-u upravljačkog PWM signala, ali i dodatne mogućnosti podešavanja intervala vodenja tranzistora pomoću mikrokontrolera, što omogućuje implementaciju pulsnog moda rada osvjetljenja. Na slici 2.15 prikazana je shema koja realizira upravljanje pomoću NPN tranzistora i PWM signala.



**Slika 2.15:** Upravljanje LED osvjetljenjem pomoću NPN tranzistora

Korištenjem jednog tranzistora omogućuje se upravljanje intenzitetom i trajanjem impulsa samo jednog nezavisnog kanala. Za neke primjene korisno je rastaviti ta dva upravljanja na dva nezavisna upravljačka kruga. Na taj način moguće je ostvariti razne kombinacije upravljanja, kao npr. korištenje jednog zajedničkog drivera za intenzitet, dok bi se impulsima upravljalo nezavisno za svaki individualni kanal pomoću više drivera za pulsiranje. U tu svrhu može se koristiti PNP tranzistor kao PWM modulisani izvor (eng. *Source*) za upravljanje intenzitetom, dok se za upravljanje impulsima može koristiti NPN tranzistor u ulozi ponora (eng. *Sink*). LED segment se spaja između kolektorâ ova dva tranzistora kao što je pokazano na slici 2.16. Upravljački signal u bazi NPN tranzistora se može generisati pomoću digitalnog pina mikrokontrolera. Ovaj način upravljanja se po svojoj strukturi slaže sa konceptom fleksibilnosti o kojem je ranije bilo riječi, te će se zbog toga koristiti pri realizaciji ovoga rada.



**Slika 2.16:** Upravljanje LED osvjetljenjem pomoću PNP i NPN tranzistora

## 2.6 Dizajn štampanih pločica – PCB

Jedan od finalnih dijelova realizacije hardverskog dijela modula je dizajn štampanih pločica – PCB (eng. *Printed Circuit Board*). U okviru ovog rada dizajn modula je podijeljen na četiri manje cjeline koje čine kontroler i rasvjetno tijelo. Razlog za razdvajanje leži u tome što se vertikalnim povezivanjem manjih cjelina postiže veća iskoristivost prostora.

Prva cjelina je mikrokontroler sa osnovnom konfiguracijom, druga cjelina obuhvata komunikaciju i adresiranje uređaja, treća izlazne drivere, a rasvjetno tijelo obuhvata LED segmente. Na osnovu ove četiri cjeline projektovane su i štampane pločice. Povezivanje pločica vrši se pomoću izvedenih pinova.

### 2.6.1 Projektovanje štampanih pločica u programskom paketu EAGLE

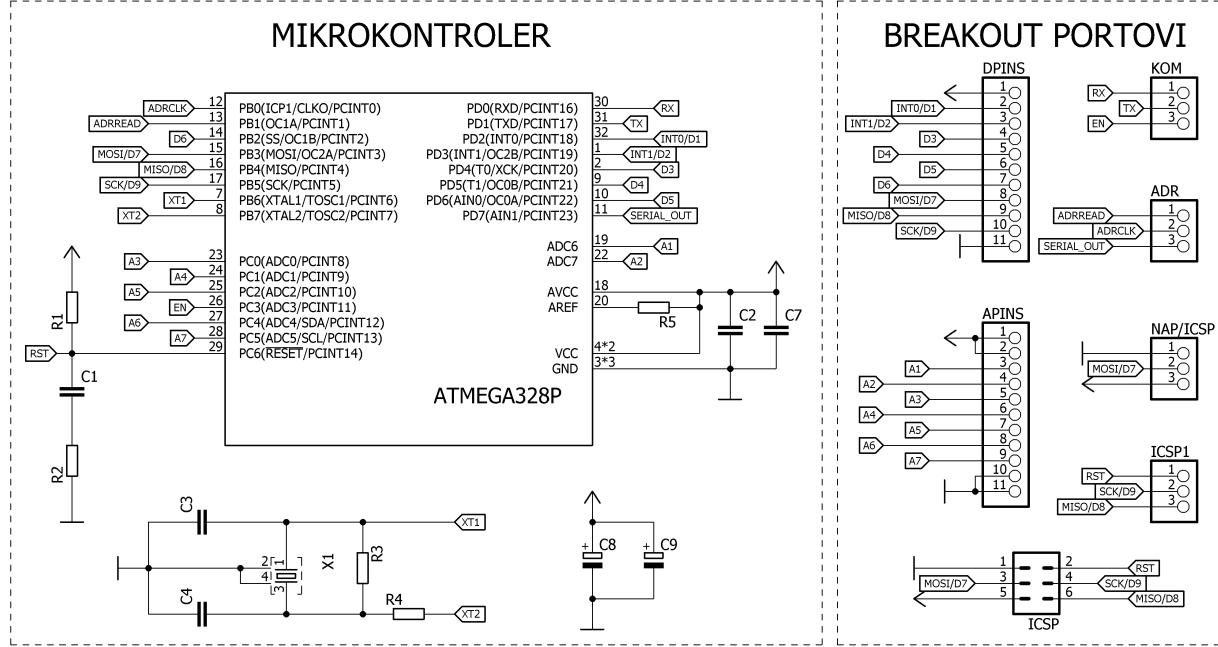
Za računarsko projektovanje štampanih pločica korišten je programski paket EAGLE.

Na slici 2.17 prikazana je shema osnovne konfiguracije mikrokontrolera sa izvedenim ulaznim i izlaznim pinovima, te pinovima za komunikaciju i adresiranje. Dizajn štampane pločice mikrokontrolera je prikazan na slici 2.18.

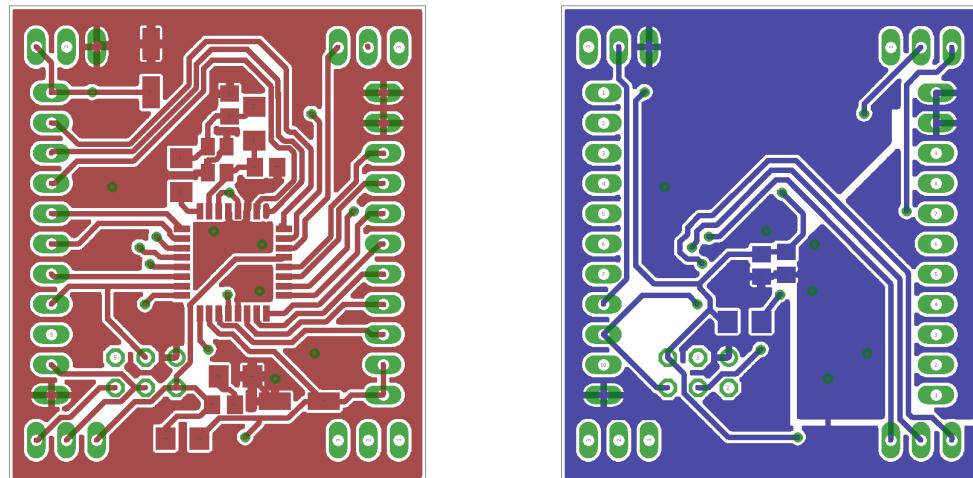
Druga cjelina se sastoji od dijela za adresiranje, te dijela za komunikaciju. Električna shema spajanja je data na slici 2.19. Štampana pločica je prikazana na slici 2.20. Povezivanje između pločice za adresiranje i komunikaciju i pločice sa mikrokontrolerom vrši se preko 12 pinova, raspoređenih u četiri ugla po tri pina, kao što se vidi na štampanoj pločici.

Treća cjelina se sastoji od 9 *sink* drivera i 5 *source* drivera i predstavlja bazu uređaja. Električna shema spajanja je data na slici 2.21. Štampana pločica je prikazana na slici 2.22. Povezivanje između pločice sa mikrokontrolerom i pločice sa driverima vrši se preko 22 pina, koji su podijeljeni u dvije linije: analogni i digitalni pinovi.

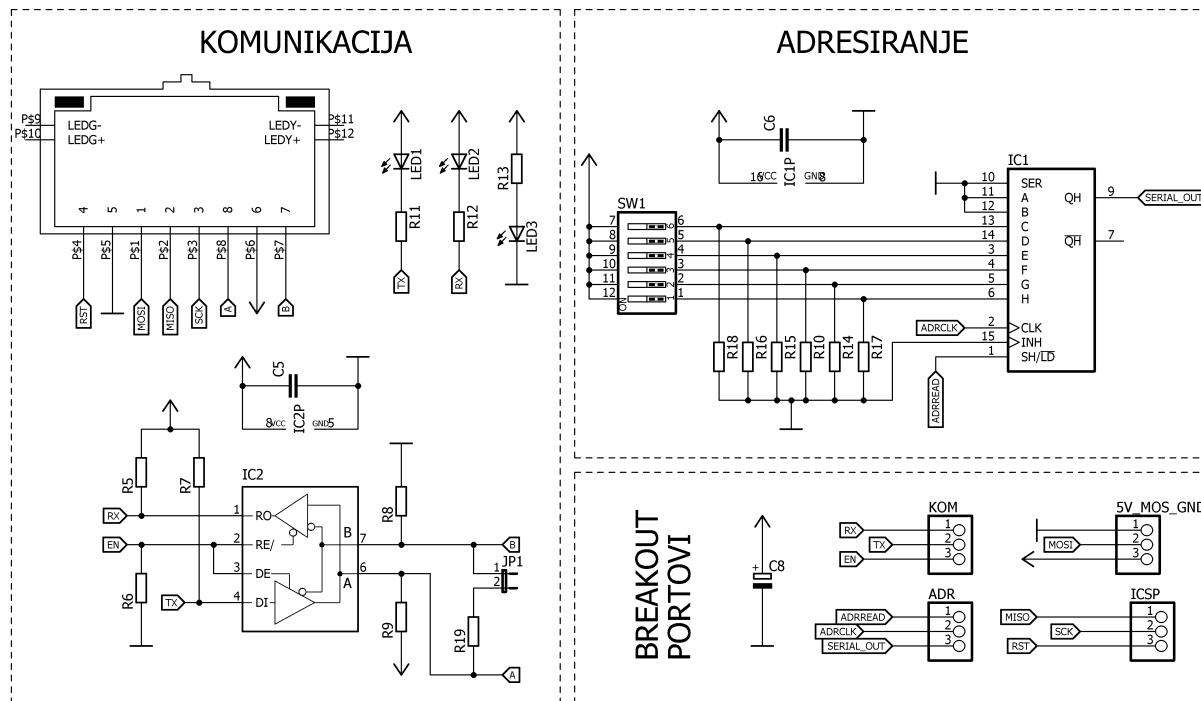
Četvrta cjelina jeste rasvjetno tijelo i ono se sastoji od četiri nezavisna LED segmenta sa pripadajućim kratkospojnicima za odabir moda rada. Električna shema ove cjeline je data na slici 2.23. Štampana pločica je prikazana na slici 2.24.



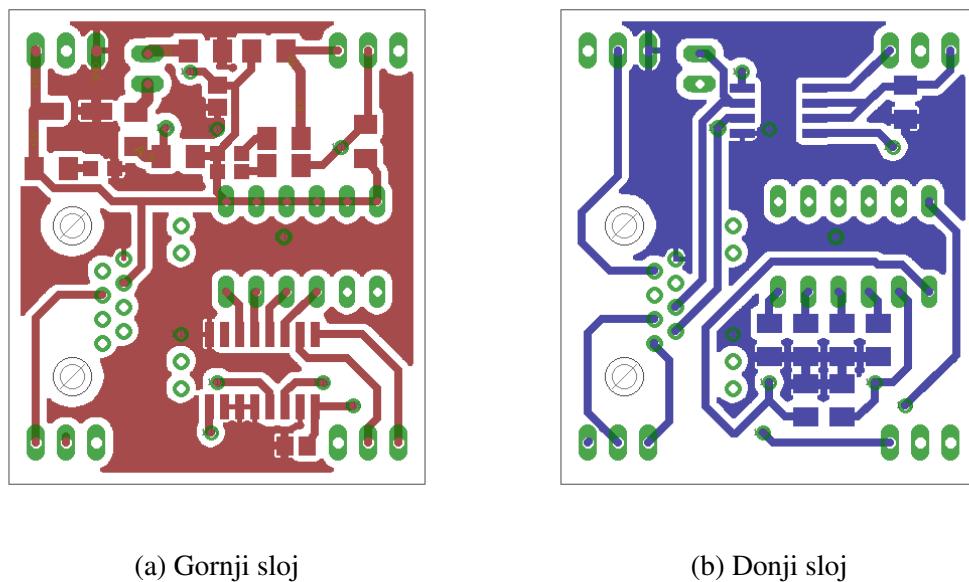
Slika 2.17: Električna shema spajanja dijela sa mikrokontrolerom



Slika 2.18: Izgled štampane ploče dijela sa mikrokontrolerom



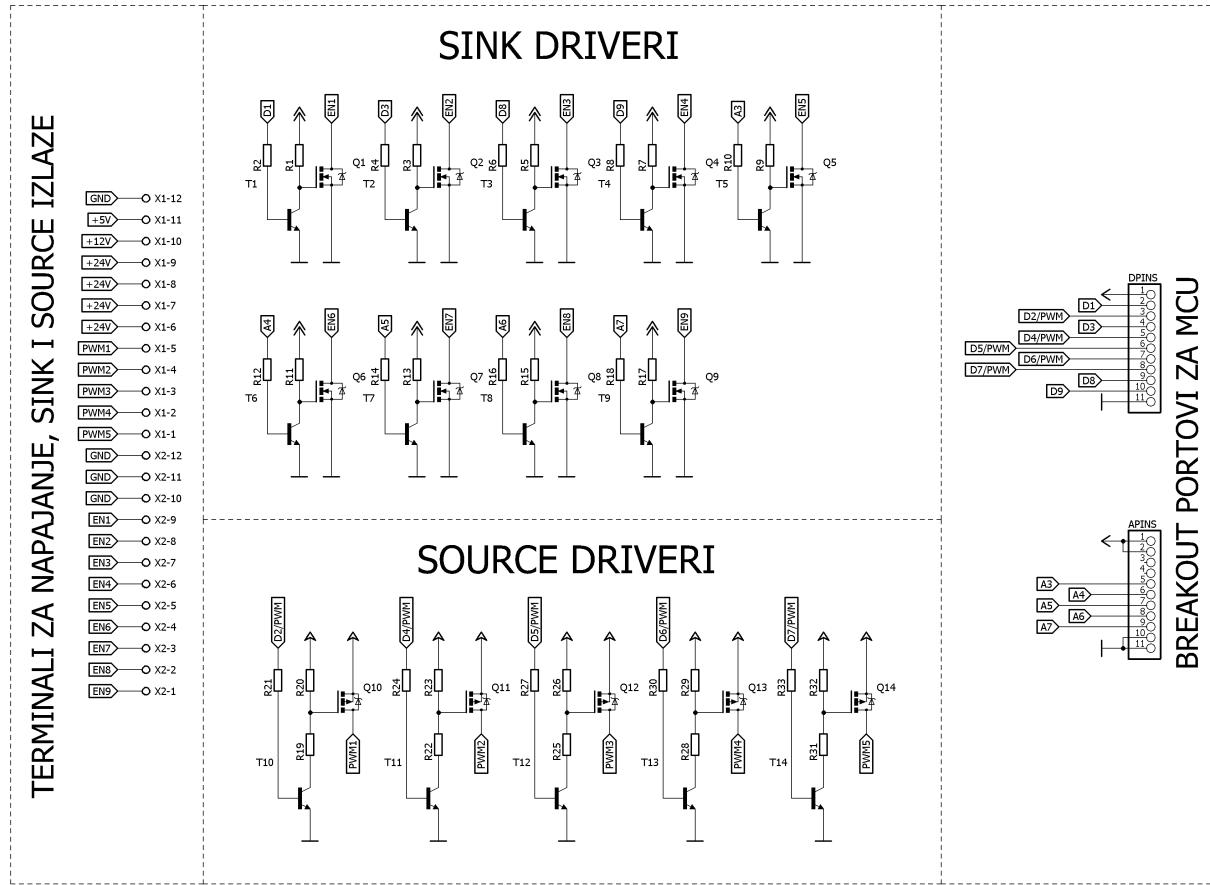
Slika 2.19: Električna shema spajanja dijela za komunikaciju i adresiranje



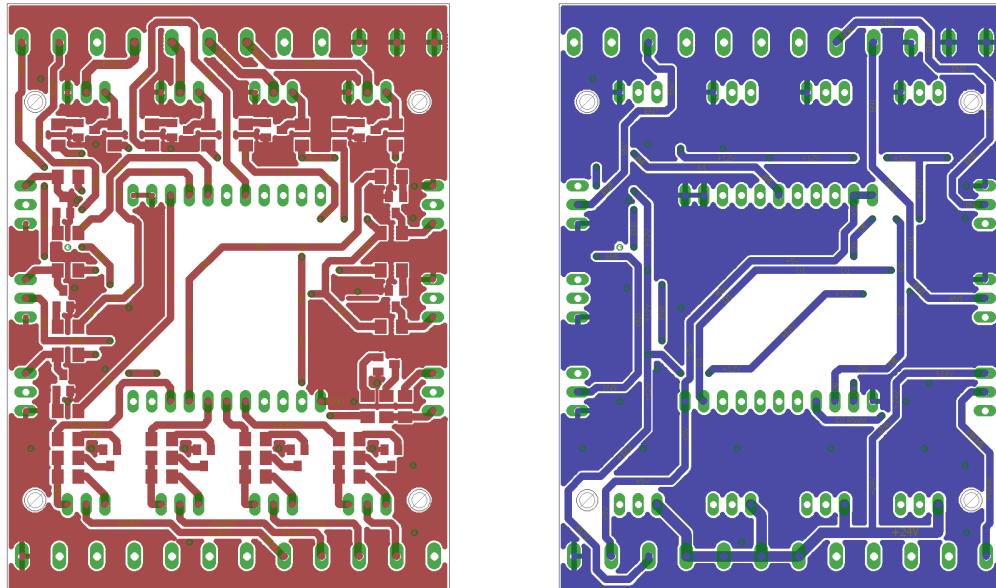
(a) Gornji sloj

(b) Donji sloj

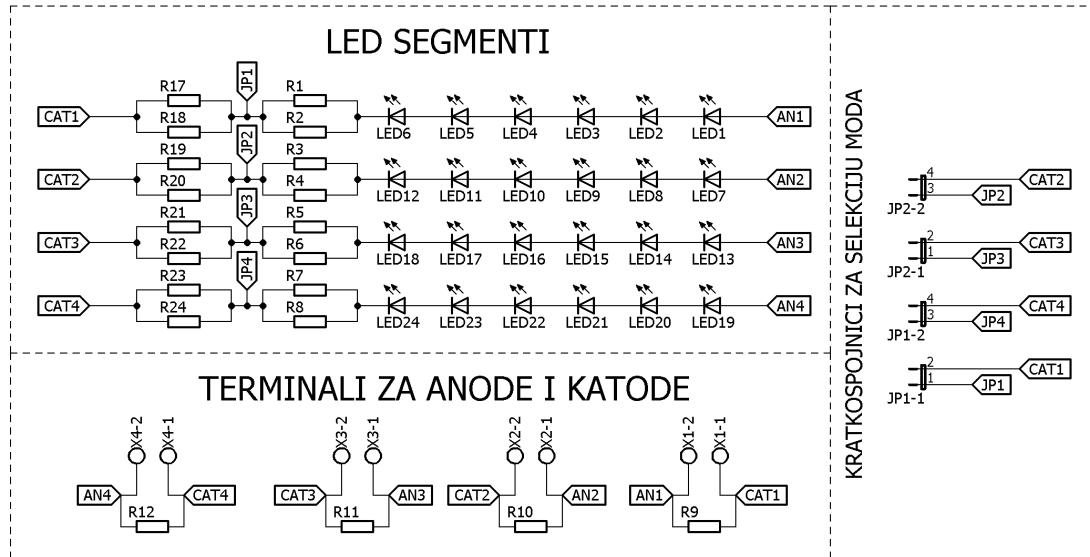
Slika 2.20: Izgled štampane ploče dijela za komunikaciju i adresiranje



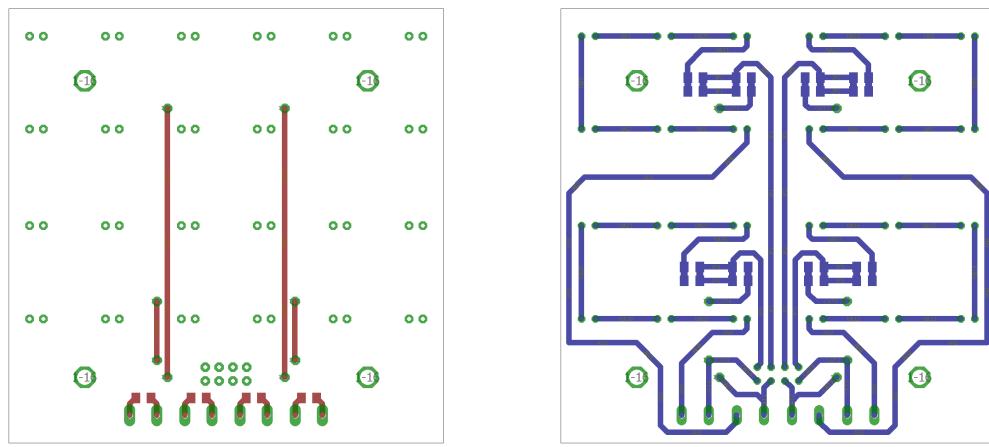
Slika 2.21: Električna shema spajanja dijela sa driverima



Slika 2.22: Izgled štampane ploče dijela sa driverima



Slika 2.23: Električna shema spajanja rasvjetnog tijela



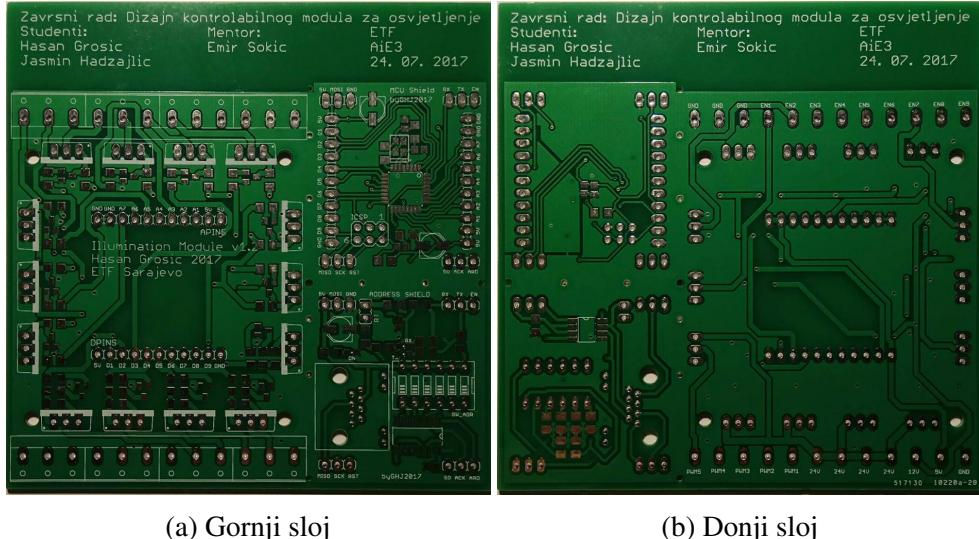
(a) Gornji sloj

(b) Donji sloj

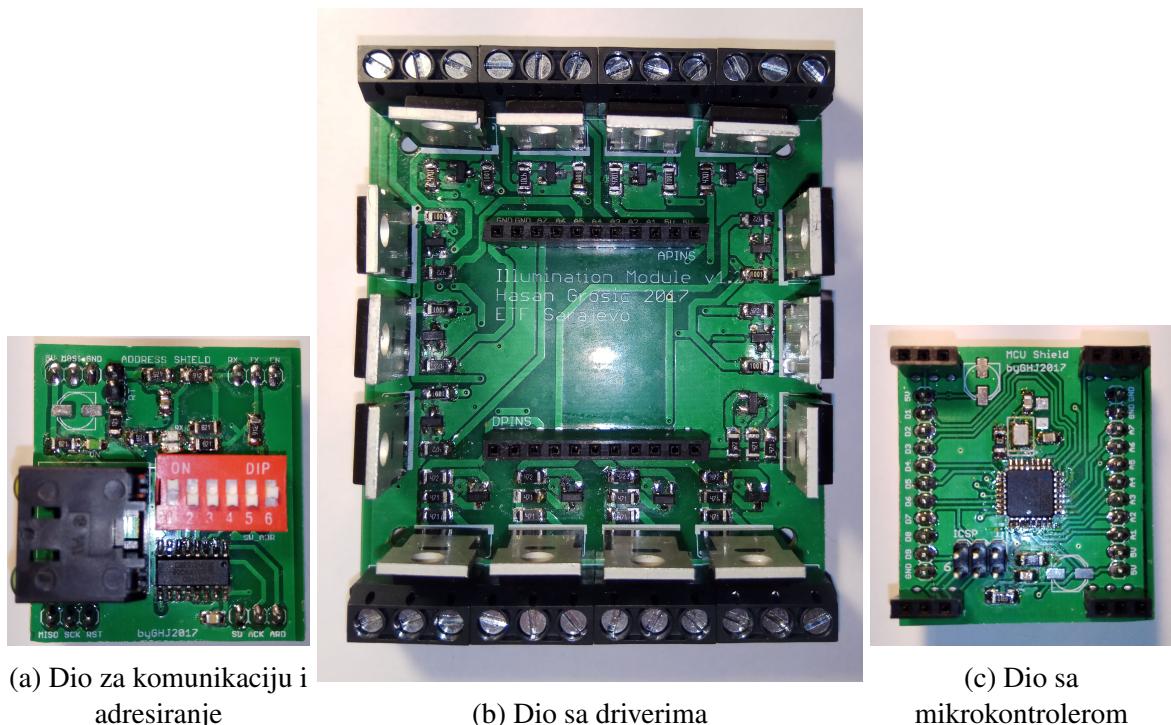
Slika 2.24: Izgled štampane ploče rasvjetnog tijela

## 2.6.2 Fizička realizacija štampanih pločica

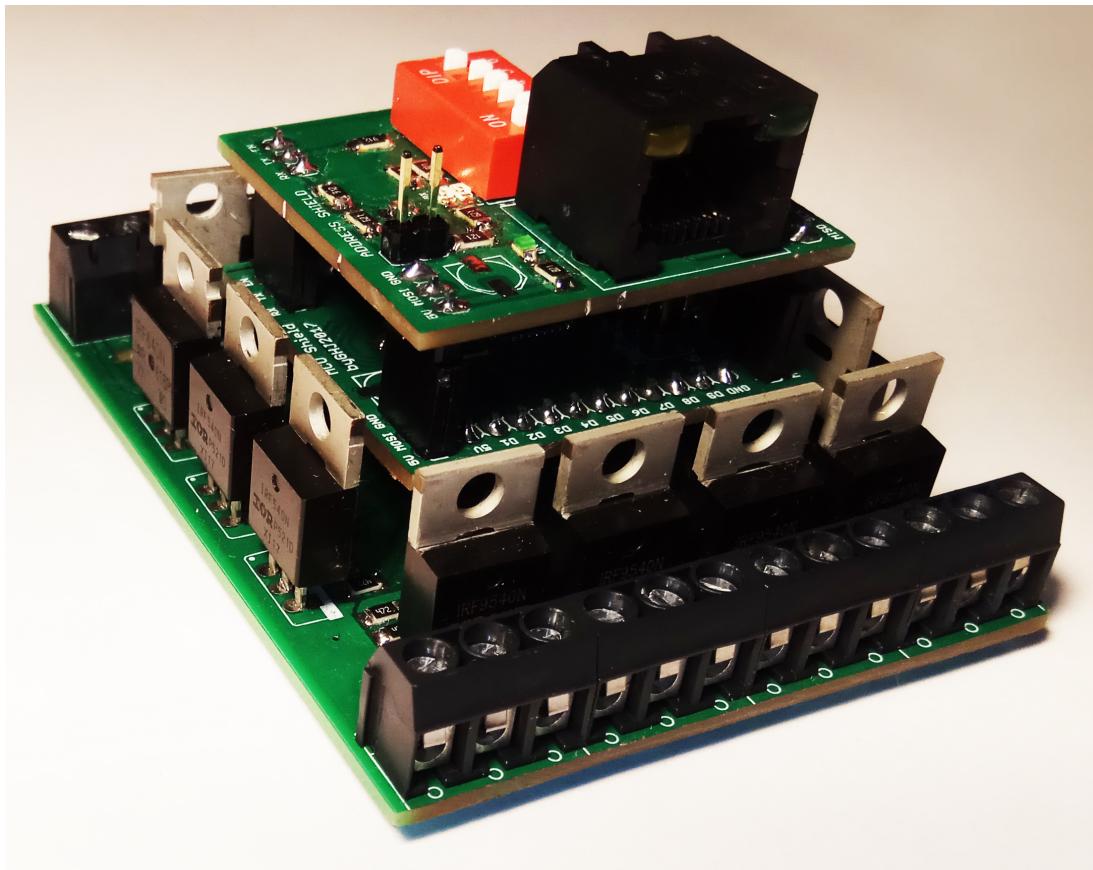
Na slikama 2.25 i 2.26 prikazan je izgled štampanih pločica nakon izrade. Prilikom izrađivanja, štampane pločice svih dijelova su objedinjene u jednu cjelinu, ali su prije lemljenja komponenti ponovo rastavljene. Izgled cijelog modula nakon povezivanja pojedinačnih pločica prikazan je na slici 2.27.



**Slika 2.25:** Fizički izgled štampanih pločica modula za osvjetljenje (bez komponenti)

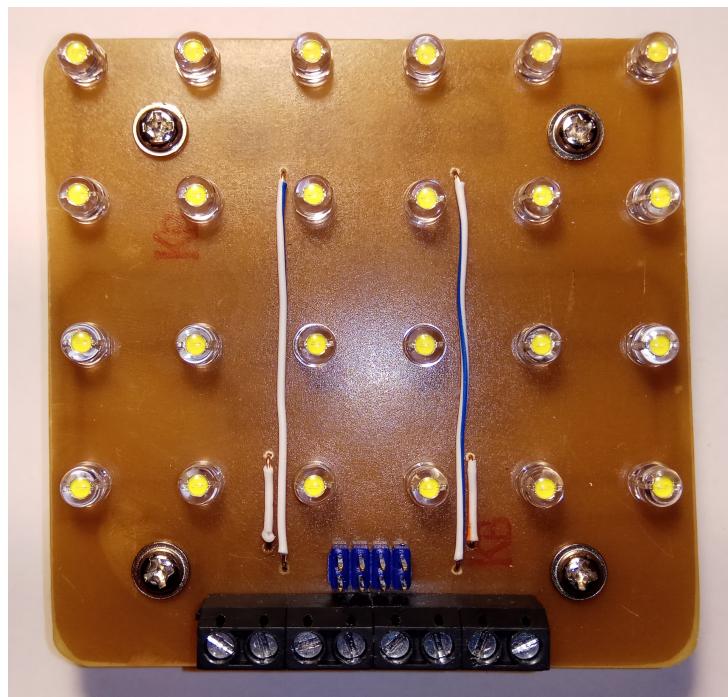


**Slika 2.26:** Fizički izgled štampanih pločica modula za osvjetljenje (sa komponentama)



**Slika 2.27:** Fizički izgled modula za osvjetljenje nakon povezivanja pojedinačnih dijelova

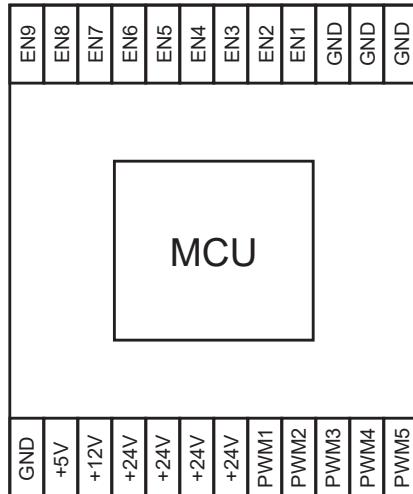
Na slici 2.28 prikazan je fizički izgled rasvjetnog tijela.



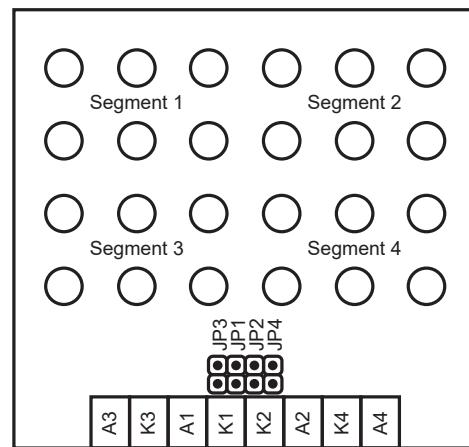
**Slika 2.28:** Fizički izgled rasvjetnog tijela

## 2.7 Načini povezivanja LED segmenata sa kontrolerom

U ovom dijelu rada bit će detaljno objašnjene sve moguće konfiguracije upravljanja LED segmentima. Na slici 2.29a su prikazani položaji driverskih izlaza kontrolera, kao i terminala na koje se dovodi napajanje (GND, +5V, +12V i +24V), dok su na slici 2.29b prikazane anode, katode i konfiguracioni kratkospojnici koji odgovaraju pojedinačnim segmentima rasvjetnog tijela. Kao što je ranije objašnjeno, kratkospojnici služe za selekciju strujnog nivoa LED segmenta (bez kratkospojnika – normalni mod, sa kratkospojnikom – preopteretni mod).



(a) Kontroler



(b) Rasyjetno tijelo

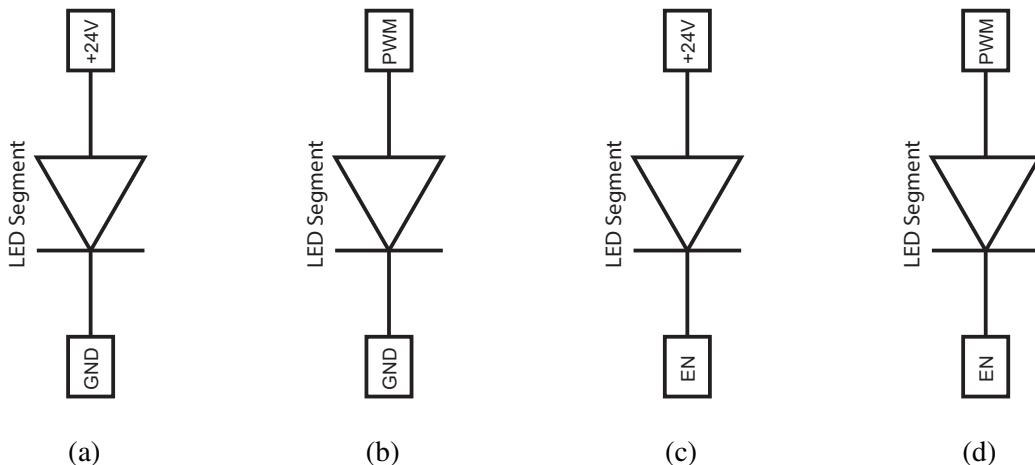
**Slika 2.29:** Raspored ulaza i izlaza modula

U tabeli 2.1 su prikazane oznake pojedinih elemenata, zajedno sa njihovim ulogama.

**Tabela 2.1:** Oznake elemenata modula

Oznaka	Uloga
PWM[1:5]	<i>Source</i> driveri kontrolera
EN[1:9]	<i>Sink</i> driveri kontrolera
A[1:4]	Anode LED segmenata
K[1:4]	Katode LED segmenata
JP[1:4]	Kratkospojnici za odabir strujnog nivoa LED segmenata

Pored izvoda *sink* i *source* drivera, za povezivanje segmenata na kontroler predviđena su i po tri izvoda GND i +24V. Ukupno postoje četiri moguća načina povezivanja, zavisno od toga na koje izvode kontrolera se spajaju anoda i katoda segmenta. Načini povezivanja prikazani su na slici 2.30.



**Slika 2.30:** Mogući načini povezivanja LED segmenata sa kontrolerom

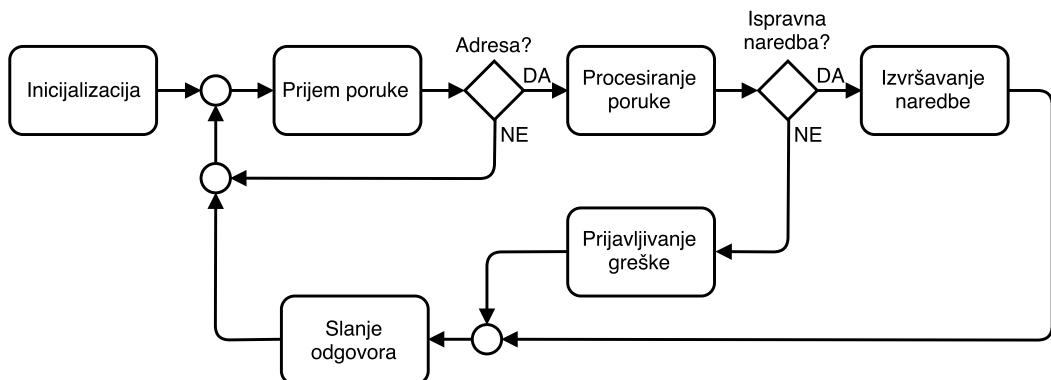
Načini povezivanja su:

- **Anoda → +24V | Katoda → GND** (slika 2.30a):  
Segment kontinualno svijetli, bez mogućnosti upravljanja intenzitetom ili impulsima.
- **Anoda → PWM | Katoda → GND** (slika 2.30b):  
Segment kontinualno svijetli, uz mogućnost upravljanja intenzitetom, ali ne i impulsima.
- **Anoda → +24V | Katoda → EN** (slika 2.30c):  
Segment svijetli u intervalima, uz mogućnost upravljanja impulsima, ali ne i intenzitetom.
- **Anoda → PWM | Katoda → EN** (slika 2.30d):  
Segment svijetli u intervalima, uz mogućnost upravljanja impulsima i intenzitetom.

## Poglavlje 3

# Softversko upravljanje modulom za osvjetljenje

U ovom poglavlju bit će opisana softverska realizacija sistema, što se odnosi na programiranje mikrokontrolera. Cilj softverskog dijela je omogućiti da se na osnovu naredbe koju pošalje računar, izvrši odgovarajuća funkcija, te kao rezultat vrati povratna poruka. 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, svi korišteni moduli mikrokontrolera se inicijaliziraju. Nakon toga uređaj ulazi u petlju, u kojoj očekuje poruku sa naredbom. Kada računar pošalje naredbu, softver izlazi iz petlje, provjerava da li je naredba upućena na njegovu adresu, te ukoliko jeste, 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 implementirati dodatne funkcionalnosti, u zavisnosti od potencijalne primjene sistema. Potpuni programski kod koji realizira sve implementirane funkcionalnosti dat je u prilogu B.1.

U sklopu ovog rada softverski su implementirane sljedeće funkcionalnosti za manipulaciju sa modulom:

- podešavanje intenziteta pojedinačnih *source* drivera,
- postavljanje modova rada pojedinačnih *sink* drivera,
- podešavanje parametara pojedinačnih *sink* drivera
- trigerovanje *sink* drivera u aktivnom FLASH modu,
- momentalno isključivanje svih drivera.

#### Postavljanje intenziteta pojedinačnih *source* drivera

Obzirom da je svaki *source* driver hardverski implementiran kao PWM modulisani izvor, intenzitet linearno ovisi o *duty cycle*-u upravljačkog PWM signala, stoga se promjena intenziteta ogleda u upisivanju odgovarajućih 8-bitnih vrijednosti *duty cycle*-a u kontrolne registre pripadajućih PWM kanala mikrokontrolera. Moguće je podešavati intenzitet na cijelom rasponu vrijednosti, tj. od 0 do 100%.

#### Postavljanje modova rada pojedinačnih *sink* drivera

Ova funkcionalnost omogućuje korisniku da za svaki *sink* driver odabere nezavisni režim rada. Četiri moda koja su implementirana su:

- **OFF** – driver isključen,
- **ON** – driver kontinualno uključen,
- **FLASH** – driver na zahtjev generiše impuls podesive dužine trajanja i vremena odgode (eng. *Delay Time*),
- **STROBE** – driver periodično generiše impulse podesivog perioda i *duty cycle*-a.

Ovakvom implementacijom otvara se mogućnost korištenja jednog modula za više iluminacijskih postavki, npr. da isti modul generiše kontinualno ambijentalno osvjetljenje, ali i snažne diskretne svjetlosne impulse u trenucima uzimanja uzoraka kamerom.

#### Podešavanje parametara pojedinačnih *sink* drivera

Posljednja dva od prethodno pobrojanih modova rada *sink* drivera karakterišu njihovi parametri. Ti parametri određuju vremenski oblik generisanih svjetlosnih impulsa. FLASH mod je karakterističan po vremenu trajanja i vremenu odgode impulsa, dok STROBE mod karakterišu period i *duty cycle* generisanih impulsa. U FLASH modu, vrijeme odgode impulsa se može

podešavati od 0 do 1000 ms, a vrijeme trajanja impulsa od 1 do 1000 ms. U STROBE modu, period impulsa se može podešavati od 10 do 1000 ms, a *duty cycle* od 10 do 90%. Tabelarni prikaz svi modova rada *sink* kanala i pripadajućih parametara dat je u tabeli 3.1.

### Trigerovanje *sink* drivera u aktivnom FLASH modu

Radi precizne vremenske sinhronizacije potrebno je omogućiti brzo okidanje svjetlosnih impulsa. To se obavlja na način da se prvo potrebeni *sink* driveri postave u FLASH mod sa odgovarajućom konfiguracijom parametara, a zatim posebnom instrukcijom za okidanje (trigerovanje) koja se modulu prosljeđuje od strane računara u trenutku uzorkovanja, generišu impulsi predviđenih vremenskih karakteristika.

### Momentalno isključivanje svih drivera

Ponekad je potrebno istovremeno isključiti sve rasvjetne elemente, tako što se modulu pošalje samo jedna instrukcija. Ova funkcionalnost je implementirana tako da na zahtjev postavlja intenzitete svih *source* drivera na 0%, a modove rada svih *sink* drivera na OFF.

**Tabela 3.1:** Modovi rada *sink* drivera i pripadni parametri

Mod rada	Parametar 1	Vrijednost 1	Parametar 2	Vrijednost 2
OFF	-	-	-	-
ON	-	-	-	-
FLASH	Vrijeme odgode	0 – 1000 ms	Trajanje impulsa	1 – 1000 ms
STROBE	Period impulsa	10 – 1000 ms	Duty cycle	10 – 90%

## 3.2 Razvijeni komunikacijski protokol

Prilikom uspostavljanja bilo kakvog vida komunikacije od esencijalne važnosti je da se definiše format poruke, tj. paketa, koji se šalje. Očito je da svaki paket treba da sadrži adresu uređaja kojem je namijenjen. Osim toga, paket treba da sadrži podatak o početku i kraju poruke, te samu informaciju koju korisnik šalje. Pored ovoga ponekad je dobro slati i pomoćne kontrolne podatke, pomoću kojih se umanjuje 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 (terminatora). Zbog različitosti funkcija i parametara, dužina poruke nije konstantna. Format poruke prikazan je u tabeli 3.2.

**Tabela 3.2:** Format poruke korišten za realizaciju komunikacije uređaja

Start (2 bajta)	Adresa (1 bajt)	Znak naredbe (1 bajt)	Parametri (x bajta)	Kraj (3 bajta)
'#' '#'	0-255	-	-	CR CR NL

Kao što se vidi, za start se koriste dva znaka '#', jer se rijetko pojavljuju u poruci kao parametri. Također za kraj poruke koriste se tri bajta, dva *carry return* ('\r') i jedan *new line* ('\n') terminator, jer postoji mogućnost da se unutar dijela poruke koji sadržava parametre naredbe, pošalju dva znaka jedan iza drugog, koji mogu označiti kraj poruke, te na taj način dovesti do pogreške u čitanju.

### 3.2.2 Implementirane instrukcijske poruke

U sklopu razvoja komunikacijskog protokola, implementirane su i određene instrukcijske poruke pomoću kojih se upravlja funkcionalnostima modula za osvjetljenje. U narednom dijelu opisane su sve implementirane poruke, zajedno sa pripadnim odgovorima koje generiše modul.

#### Konfiguracija *source* drivera

Ova poruka služi za postavljanje intenziteta *source* drivera. Znak za konfiguraciju *source* drivera je 'A'. U poruci se šalje 5 bajta koji sadrže informaciju o *duty cycle-u* pojedinačnih *source* drivera. Radi eliminacije mogućnosti pogrešnog očitavanja terminadora poruke, između svakog od bajta dodaje se znak 'I'.

Dužina poruke, uključujući i zaglavljene sa startom i adresom i terminatom, iznosi 16 bajta. Kao odgovor modul vraća primljenu poruku ukoliko je konfiguracija uspješna, odnosno poruku greške ukoliko nije uspješna.

#### Konfiguracija *sink* drivera

Ova poruka služi za postavljanje moda i parametara *sink* drivera. Znak za konfiguraciju *sink* drivera je 'B'. U poruci se šalje 9 snopova od po tri bajta, gdje prvi bajt predstavlja mod rada drivera, a naredna dva bajta skalirane vrijednosti pripadnih parametara. Ukoliko se radi o ON

ili OFF modu, vrijednosti ta dva bajta se zanemaruju. U tabeli 3.3 su opisane interpretacije pojedinih bajta u zavisnosti od moda rada.

**Tabela 3.3:** Interpretacija sadržaja poruke za konfiguraciju *sink* drivera

Vrijednost 1. bajta	Mod rada	Značenje 2. bajta	Značenje 3. bajta
'A'	OFF	Zanemaruje se	Zanemaruje se
'B'	ON	Zanemaruje se	Zanemaruje se
'C'	FLASH	Trajanje impulsa	Vrijeme odgode
'D'	STROBE	Duty cycle	Period

Dužina poruke, uključujući i zaglavljje sa startom i adresom i terminator, iznosi 34 bajta. Kao odgovor modul vraća primljenu poruku ukoliko je konfiguracija uspješna, odnosno poruku greške ukoliko nije uspješna.

### Trigerovanje FLASH drivera

Ova poruka služi za okidanje (trigerovanje) *sink* drivera koji su prethodno konfigurisani za rad u FLASH modu. Znak za trigerovanje FLASH drivera je 'T'. Poruka ne sadrži dodatne parametre.

Dužina poruke, uključujući i zaglavljje sa startom i adresom i terminator, iznosi 7 bajta. Kao odgovor modul vraća primljenu poruku ukoliko je trigerovanje uspješno, odnosno poruku greške ukoliko nije uspješno.

### Resetovanje svih drivera

Ova poruka služi za momentalno isključivanje svih drivera, uključujući i vraćanje odgovarajućih parametara na nulte vrijednosti. Znak za resetovanje svih drivera je 'R'. Poruka ne sadrži dodatne parametre.

Dužina poruke, uključujući i zaglavljje sa startom i adresom i terminator, iznosi 7 bajta. Kao odgovor modul vraća primljenu poruku ukoliko je trigerovanje uspješno, odnosno poruku greške ukoliko nije uspješno.

### Poruka greške

Pomoću poruke greške modul signalizira neočekivani ili nepravilni oblik ili sadržaj dolazne poruke. Formira se tako što se u bafer odlazne poruke smjesti prvi bajt dolazne poruke, znak '?' koji signalizira grešku i dodatni znak koji predstavlja kod greške. U tabeli 3.4 su data značenja pojedinih kodova greške.

**Tabela 3.4:** Kodovi poruke greške

Znak	Značenje
'K'	Nepoznata komanda
'F'	Nepravilan format dolazne poruke
'M'	Nepoznat mod <i>sink</i> drivera
'P'	Neočekivan broj parametara

## Poglavlje 4

# Testiranje rada modula za osvjetljenje

U ovom dijelu rada izvršeno je testiranje rada modula. Testiranje se sastoji od testiranja komunikacije, te testiranja odziva modula (u smislu odgovarajućih promjena na segmentima rasvjetnog tijela), za unaprijed definisani scenario. Slanje podataka od strane računara vrši se pomoću programskog paketa MATLAB. U tabeli 4.1 dat je način povezivanja rasvjetnog tijela i kontrolera koji je korišten prilikom testiranja.

**Tabela 4.1:** Korišteni način povezivanja modula prilikom testiranja

Rasvjetno tijelo	Kontroler
A1	PWM1
K1	EN1
A2	PWM2
K2	EN2
A3	PWM3
K3	EN3
A4	PWM4
K4	EN4

Definisan je testni scenario koji se sastoji u sljedećem:

- konfiguracija *source* drivera,
- konfiguracija *sink* drivera sa nedovoljnim brojem parametara,
- konfiguracija *sink* drivera sa tačnim brojem parametara,
- slanje instrukcije sa nepoznatim znakom funkcije,
- trigerovanje FLASH drivera,
- resetovanje svih drivera.

U tabeli 4.2 prikazane su instrukcijske poruke koje čine opisani scenarij, zajedno sa odgovorima koje je generisao modul. Sa znakom 'X' u porukama označena je adresa uređaja na kojem je vršeno testiranje. Odabrana je adresa 33, pri čemu su položaji DIP prekidača postavljeni na "100001", a vodeća dva bita softverski postavljena na nulu jer se radi o modulu za osvjetljenje.

**Tabela 4.2:** Odgovori modula na poslane instrukcijske poruke prilikom testiranja

R.b.	Poslana poruka	Odgovor
1.	[##XA",255,"I",192,"I",128,"I",64,"I",0]	Dolazna poruka
2.	[##XB","B",0,0,"C",128,128,"D",64,255]	[##XB?P"]
3.	[##XB","B",0,0,"C",128,128,"D",64,255,"B",0,0,"A",0,0,"A",0,0,"A",0,0,"A",0,0,"A",0,0]	Dolazna poruka
4.	[##XM",120,255]	[##XM?K"]
5.	[##XT"]	Dolazna poruka
6.	[##XR"]	Dolazna poruka

U prvom koraku poslana je poruka za konfiguraciju *source* drivera. Na osnovu sadržaja poruke zaključuje se da je potrebno postaviti *duty cycle sink* drivera PWM[1:5] na vrijednosti 255, 192, 128, 64 i 0, što odgovara relativnim vrijednostima intenziteta 100%, 75%, 50%, 25% i 0%, respektivno. Uređaj je vratio istu poruku, što znači da je konfiguracija uspješno izvršena.

U drugom koraku poslana je poruka za konfiguraciju *sink* drivera sa nedovoljnim brojem parametara, što znači da bi modul trebao signalizirati grešku. Uređaj je vratio poruku [##XB?P"], gdje znak '?' signalizira grešku, a znak 'P' daje informaciju o tome da se radi o nedovoljnem broju parametara.

U trećem koraku također je poslana poruka za konfiguraciju *sink* drivera, ali sa potrebnim brojem parametara. Na osnovu sadržaja poruke zaključuje se da je potrebno postaviti:

- Drivere EN1 i EN4 u ON mod,
- Driver EN2 u FLASH mod sa trajanjem impulsa od 500ms i vremenom odgode od 500ms,
- Driver EN3 u STROBE mod sa *duty cycle*-om od 25% i periodom od 1000ms,
- Drivere EN5, EN6, EN7, EN8 i EN9 u OFF mod.

Uređaj je vratio istu poruku, što znači da je konfiguracija uspješno izvršena. Osmatranjem ponašanja rasvjetnog tijela uočeno je sljedeće:

- Segment 1 svijetli konstantno maksimalnim intenzitetom.
- Segment 2 ne svijetli jer je u stanju pripravnosti do zahtjeva za okidanje FLASH-a.
- Segment 3 svijetli periodično sa polovičnim intenzitetom, period svjetlosnih impulsa iznosi 1000 ms, a trajanje jednog impulsa 250ms.
- Segment 4 svijetli konstantno četvrtinom maksimalnog intenziteta.

Zaključuje se da su konfiguracije iz prvog i trećeg koraka zaista bile uspješne.

U četvrtom koraku poslana je poruka sa znakom funkcije 'M' koji nije softverski implementiran, što znači da bi modul trebao signalizirati grešku. Uređaj je vratio poruku ["##XM?K"], gdje znak '?' signalizira grešku, a znak 'K' daje informaciju o tome da se radi o nepoznatoj komandi.

U petom koraku poslana je poruka sa zahtjevom za trigerovanje FLASH drivera. Uređaj je vratio istu poruku, što znači da je trigerovanje uspješno izvršeno. Prispjećem zahtjeva za trigerovanje, segment 2 je nakon vremena odgode od 500ms, generisao svjetlosni impuls sa trajanjem od 500 ms i 75% maksimalnog intenziteta, što dodatno potvrđuje uspješnost poslane komande.

U šestom koraku poslana je poruka sa zahtjevom za resetovanje svih drivera. Svi segmenti rasvjetnog tijela su prestali da svijetle, a uređaj je vratio istu poruku, što znači da je resetovanje uspješno izvršeno.

Obzirom da testni scenario obuhvata sve implementirane funkcionalnosti modula za osvjetljenje, a sve stavke scenarija su uspješno izvršene, zaključuje se da je razvijeni modul u potpunosti funkcionalan i odgovara prepostavljenim specifikacijama koje su definisane na početku rada.

# Zaključak

U ovom radu prezentiran je jedan od načina realizacije kontrolabilnog modula za osvjetljenje. Sa ekonomskog aspekta, u poređenju sa komercijalnim uređajima, ponuđeno rješenje je neuporedivo jeftinije i omogućuje fleksibilniju manipulaciju izlaznim funkcionalnostima. Međutim sa aspekta performanse, realizovani uređaj je do određene mjere ograničen u odnosu na komercijalna rješenja. Ovo se ogleda u maksimalnim iznosima izlaznih struja koje ograničavaju snagu rasvjetnih tijela koja se povezuju na modul, kao i maksimalnoj brzini rada koja je esencijalni uslov za korištenje modula u inspekcijskim postavkama sa velikim brzinama uzorkovanja.

## Smjernice za budući rad

Za realizaciju modula, 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 i mogućnostima tajmerskih, kao i drugih relevantnih hardverskih modula implementiranih na kontroleru.

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

# **Prilozi**

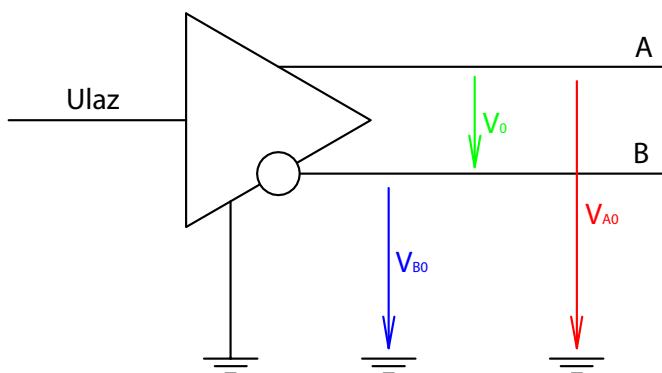
## Prilog A

# 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 elektromagnetični valovi). Zbog toga razlika između napona  $V_{A0}$  i  $V_{B0}$  ostaje ista, tj. ne dolazi do greške pri prenosu.

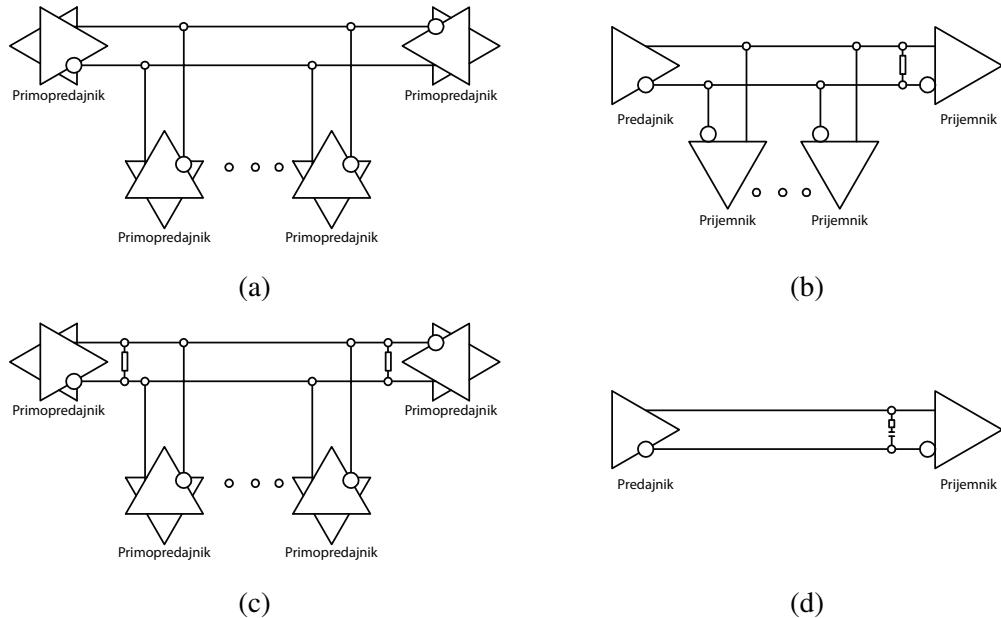


Slika A.1: Diferencijalni naponi RS485 predajnika

### A.2 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 [12].



**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  $120\Omega$ ),
- *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.

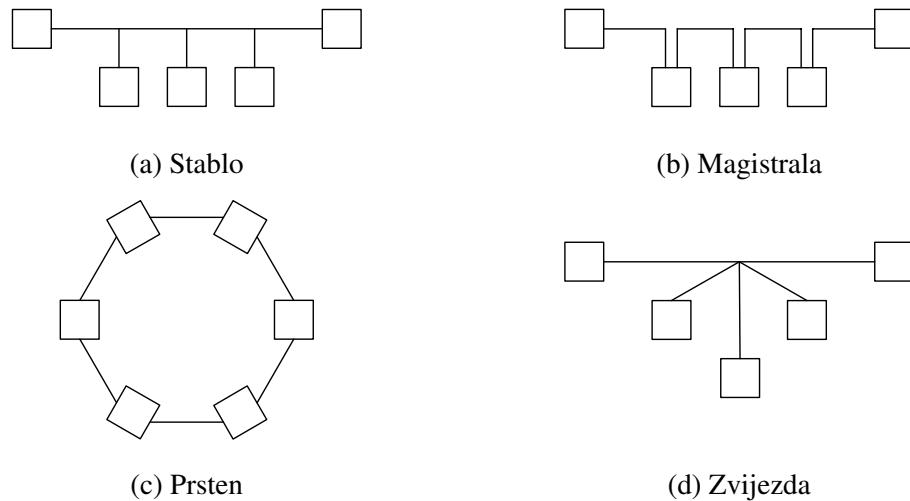
### A.3 Topologija mreže

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

U nastavku je dat programski kod koji realizira softverske funkcionalnosti modula za osvjetljenje.

```
1  /*************************************************************************/
2  /****** ZAVRSNI RAD *****/
3  /**** Dizajn kontrolabilnog modula za osvjetljenje ***/
4  /****** Mentor: Doc. dr Emir Sokic *****/
5  /****** Student: Hasan Grosic *****/
6  /****** *****/
7
8  /* Mapiranje pinova */
9  /*****
10 * SINK kanali
11 -----
12 * EN1 | PD2
13 * EN2 | PD4
14 * EN3 | PB4
15 * EN4 | PB5
16 * EN5 | PC0
17 * EN6 | PC1
18 * EN7 | PC2
19 * EN8 | PC4
20 * EN9 | PC5
21 *****
22 * SOURCE kanali
23 -----
24 * PWM1 | PD3
25 * PWM2 | PD5
26 * PWM3 | PD6
27 * PWM4 | PB2
28 * PWM5 | PB3
29 *****
30
31 /****** VARIJABLE *****/
32
33 byte dolazna_poruka[40]; // Ulazni buffer za serijske podatke
34 int duzina_dolazne_poruke; // Broj bajta u ulaznom bufferu
35 byte odlazna_poruka[40]; // Izlazni bufer za serijske podatke
36 int duzina_odlazne_poruke; // Broj bajta u izlaznom bufferu
37 byte adresa; // Adresa uredjaja
```

```

38 byte intenziteti[5]; // Intenziteti SOURCE kanala
39 byte modovi[9]; // Modovi SINK kanala (A - OFF; B - ON; C - FLASH; D -
      STROBE)
40 int trajanje_impulsa[9]; // Trajanje impulsa na SINK kanalima u FLASH/
      STROBE modu
41 int odgoda_period[9]; // Vremena odgode impulsa/periodi bljeskanja na SINK
      kanalima u FLASH/STROBE modu
42 // Ukoliko je FLASH mod onda je u pitanju vrijeme odgode (delay) impulsa u
      ms, a ako je STROBE mod onda se radi o periodu bljeskanja u ms
43 boolean trigeri[9]; // Flagovi za trigerovanje impulsa, setuju se u
      trigerujFlash(), a koriste se u ISR()
44 int brojaci_ms[9]; // Brojaci koji služe za brojanje ms u FLASH/STROBE modu
      , koriste se u ISR()
45 int osmine = 0; // Brojac osmina (125us), 8 osmina daje 1ms, koristi se u
      ISR()
46 const byte broadcast = 0x55; // Adresa za BROADCAST
47 boolean slanje_flag = true; // Flag koji se resetuje ukoliko se ne poklapa
      adresa ili ako je broadcast ili ako je duzina dolazne poruke manja od 6,
      koristi se u loop()
48 boolean greska_flag = false; // Flag za signalizaciju greske
49 boolean procesiranje_flag = true; // Flag koji se resetuje ukoliko nema
      potrebe obradjivati poruku jer je kratka
50 ****
51 **** METODE ****
52 ****
53
54 void greska(byte sifra) // Signalizacija greske
55 {
56     odlazna_poruka[4] = '?';
57     odlazna_poruka[5] = sifra;
58     duzina_odlazne_poruke = 6;
59 }
60
61 void setujBit(volatile uint8_t &registar, uint8_t redni_broj) {
62     registar |= 1 << redni_broj;
63 }
64
65 void resetujBit(volatile uint8_t &registar, uint8_t redni_broj) {
66     registar &= ~(1 << redni_broj);
67 }
68
69 byte ocitajBit(const volatile uint8_t &registar, uint8_t redni_broj)
70 {
71     if (registar & (1 << redni_broj))
72         return '1';
73     return '0';
74 }
75
76 void adresiraj() // Ucitavanje adrese sa modula za adresiranje
77 {
78     adresa = 0x00; // Resetovanje adrese
79     setujBit(DDRB, PB0); // Inicijalizacija ADRCLK kao izlaz
80     setujBit(DDRB, PB1); // Inicijalizacija ADRREAD kao izlaz
81     // Paralelno ucitavanje u shift registar
82     resetujBit(PORTB, PB1); // Setovanje ADRREAD
83     delay(1);
84     setujBit(PORTB, PB1); // Resetovanje ADRREAD
85     delay(1);

```

```
86 // Ucitavanje adrese u mikrokontroler
87 for (int i = 0; i < 8; i++) {
88     // Citanje adrese
89     if (ocitajBit(PIND, PD7) == '1') // Ocitavanje SERIAL_OUT
90         setujBit(adresa, i);
91     // Clock za shiftanje
92     setujBit(PORTB, PB0); // Setovanje ADRCLK
93     delay(1);
94     resetujBit(PORTB, PB0); // Resetovanje ADRCLK
95     delay(1);
96 }
97 }

98
99 void inicijalizirajPWM() // Inicijalizacija timera za PWM kanale
100 {
101     // TIMER 0 - Fast PWM - Set on Compare Match, clear at BOTTOM (inverting
102     // mode)
103     TCCR0A = 0b11110011;
104     TCCR0B = 0b11000010;
105     TCNT0 = 0x00;
106     // TIMER 1 - Fast PWM (8 bit) - Set on Compare Match, clear at BOTTOM (
107     // inverting mode)
108     TCCR1A = 0b00110001;
109     TCCR1B = 0b00001010;
110     TCCR1C = 0b11000000;
111     TCNT1 = 0x0000;
112     // TIMER 0 - Fast PWM - Set on Compare Match, clear at BOTTOM (inverting
113     // mode)
114     TCCR2A = 0b11110011;
115     TCCR2B = 0b11000010;
116     TCNT2 = 0x00;
117     // Postavljanje Duty Cycle-ova za SOURCE kanale na nulu
118     // Komplement se radi zbog inverznog PWM-a
119     OCR2B = 255; // PWM1
120     OCR0B = 255; // PWM2
121     OCR0A = 255; // PWM3
122     OCR1B = 255; // PWM4
123     OCR2A = 255; // PWM5
124 }

125 void inicijalizirajPrekide() // Inicijalizacija vremenskih prekida uz
126     // podesavanja TIMER2 su obavljena u inicijalizirajPWM()
127 {
128     sei(); // Globalno omogucavanje prekida
129     setujBit(TIMSK2, TOIE2); // TOIE2, omogucavanje OVERFLOW prekida na
130     // TIMER2
131 }

132 void inicijalizirajSerial() // Inicijalizacija serijske komunikacije
133 {
134     Serial.begin(115200);
135     setujBit(DDRC, PC3);
136     resetujBit(PORTC, PC3); // Postavljanje uređaja u prijemni mod
137 }

138 void inicijalizirajIzlaze() // Inicijaliziraj izlazne kanale
139 {
140     // Postavljanje pinova u OUTPUT mod
```

```

139     setujBit(DDRD, PD2); // EN1
140     setujBit(DDRD, PD3); // PWM1
141     setujBit(DDRD, PD4); // EN2
142     setujBit(DDRD, PD5); // PWM2
143     setujBit(DDRD, PD6); // PWM3
144     setujBit(DDRB, PB2); // PWM4
145     setujBit(DDRB, PB3); // PWM5
146     setujBit(DDRB, PB4); // EN3
147     setujBit(DDRB, PB5); // EN4
148     setujBit(DDRC, PC0); // EN5
149     setujBit(DDRC, PC1); // EN6
150     setujBit(DDRC, PC2); // EN7
151     setujBit(DDRC, PC4); // EN8
152     setujBit(DDRC, PC5); // EN9
153     // Iskljucivanje svih kanala
154     resetujKanale();
155 }
156
157 void citajSerial() // Citanje serijskih podataka
158 {
159     duzina_dolazne_poruke = 0;
160     int i = 0;
161     while (i < 40)
162     {
163         if (Serial.available())
164         {
165             dolazna_poruka[i] = Serial.read();
166             if (dolazna_poruka[i] == '\n' && dolazna_poruka[i - 1] == '\r' &&
167                 dolazna_poruka[i - 2] == '\r') // Provjera terminadora CR + CR
168                 + LF
169             {
170                 i++;
171                 break;
172             }
173             i++;
174         }
175     }
176     duzina_dolazne_poruke = i;
177     if (duzina_dolazne_poruke < 7) // Ako je poruka kraca od 4 (3 karaktera
178     // su terminatori)
179     {
180         slanje_flag = false;
181         procesiranje_flag = false;
182     }
183 }
184
185 void pisiSerial() // Pisanje serijskih podataka
186 {
187     setujBit(PORTC, PC3); // Postavljanje uredjaja u predajni mod
188     odlazna_poruka[duzina_odlazne_poruke++] = '\r'; // Dodavanje CR
189     // terminatora
190     odlazna_poruka[duzina_odlazne_poruke++] = '\r'; // Dodavanje CR
191     // terminatora
192     odlazna_poruka[duzina_odlazne_poruke++] = '\n'; // Dodavanje LF
193     // terminatora
194     for (int i = 0; i < duzina_odlazne_poruke; i++) {
195         Serial.write(odlazna_poruka[i]); // Pisanje bajta
196         while (ocitajBit(UCSR0A, TXC0) == '0'); // Cekanje dok se bajt ne

```

```

    napise
191 }
192 duzina_odlazne_poruke = 3; // Resetovanje duzine odlazne poruke
193 resetujBit(PORTC, PC3); // Vracanje uredjaja u prijemni mod
194 }

195
196 void konfigurirajSourceove() // Konfiguracija intenziteta SOURCE kanala
197 {
198     if (duzina_dolazne_poruke != 16)
199     {
200         greska('P'); // Neocekivan broj parametara
201         return;
202     }
203     for (int i = 0; i < 5; i++)
204         intenziteti[i] = dolazna_poruka[2*i+4];
205     // Postavljanje Duty Cycle-ova za SOURCE kanale
206     // Komplement se radi zbog inverznog PWM-a
207     OCR2B = 255 - intenziteti[0]; // PWM1
208     OCR0B = 255 - intenziteti[1]; // PWM2
209     OCR0A = 255 - intenziteti[2]; // PWM3
210     OCR1B = 255 - intenziteti[3]; // PWM4
211     OCR2A = 255 - intenziteti[4]; // PWM5

212
213     // Potvrda prijema poruke
214     for (int i = 0; i < duzina_dolazne_poruke - 3; i++) // Kopiranje dolazne
215         odlaznu_poruka[u odlaznu
216         {
217             odlazna_poruka[i] = dolazna_poruka[i];
218         }
219         duzina_odlazne_poruke = duzina_dolazne_poruke - 3; // Funkcija
220             pisSerial() dodaje terminator, zato minus 3
221 }

222
223 void konfigurirajSinkove() // Konfiguracija modova i parametara SINK kanala
224 {
225     if (duzina_dolazne_poruke != 34)
226     {
227         greska('P'); // Neocekivan broj parametara
228         return;
229     }
230     for (int i = 0; i < 9; i++)
231     {
232         if (dolazna_poruka[4 + 3 * i] >= 'A' && dolazna_poruka[4 + 3 * i] <=
233             'D') {
234             modovi[i] = dolazna_poruka[4 + 3 * i]; // Postavka moda
235             if (dolazna_poruka[4 + 3 * i] == 'C') { // Flash
236                 odgoda_period[i] = map(dolazna_poruka[6 + 3 * i], 0, 255, 0,
237                     1000); // Odgoda impulsa [0 - 1000] ms
238                 trajanje_impulsa[i] = map(dolazna_poruka[5 + 3 * i], 0, 255, 1,
239                     1000); // Trajanje impulsa [1 - 1000] ms
240             }
241             else if (dolazna_poruka[4 + 3 * i] == 'D') { // Strobe
242                 odgoda_period[i] = map(dolazna_poruka[6 + 3 * i], 0, 255, 10,
243                     1000); // Period impulsa [10 - 1000] ms
244                 trajanje_impulsa[i] = map(dolazna_poruka[5 + 3 * i], 0, 255,
245                     0.1 * odgoda_period[i], 0.9 * odgoda_period[i]); // Duty
246                     Cycle [10 - 90] %
247             }
248     }

```

```

240         trigeri[i] = false;
241         brojaci_ms[i] = 0;
242     }
243     else {
244         greska('K');
245         return;
246     }
247 }
248
249 for (int i = 0; i < 9; i++)
250 {
251     if (modovi[i] == 'A' || modovi[i] == 'C') // Ako je OFF ili FLASH mod
252         resetuje se izlaz (setuje pin zbog inverzne logike)
253     {
254         switch (i)
255         {
256             case 0: // EN1
257                 setujBit(PORTD, PD2);
258                 break;
259             case 1: // EN2
260                 setujBit(PORTD, PD4);
261                 break;
262             case 2: // EN3
263                 setujBit(PORTB, PB4);
264                 break;
265             case 3: // EN4
266                 setujBit(PORTB, PB5);
267                 break;
268             case 4: // EN5
269                 setujBit(PORTC, PC0);
270                 break;
271             case 5: // EN6
272                 setujBit(PORTC, PC1);
273                 break;
274             case 6: // EN7
275                 setujBit(PORTC, PC2);
276                 break;
277             case 7: // EN8
278                 setujBit(PORTC, PC4);
279                 break;
280             case 8: // EN9
281                 setujBit(PORTC, PC5);
282                 break;
283         }
284     else if (modovi[i] == 'B') // Ako je ON mod setuje se izlaz (resetuje
285         pin zbog inverzne logike)
286     {
287         switch (i)
288         {
289             case 0: // EN1
290                 resetujBit(PORTD, PD2);
291                 break;
292             case 1: // EN2
293                 resetujBit(PORTD, PD4);
294                 break;
295             case 2: // EN3
296                 resetujBit(PORTB, PB4);

```

```

296         break;
297     case 3: // EN4
298         resetujBit(PORTB, PB5);
299         break;
300     case 4: // EN5
301         resetujBit(PORTC, PC0);
302         break;
303     case 5: // EN6
304         resetujBit(PORTC, PC1);
305         break;
306     case 6: // EN7
307         resetujBit(PORTC, PC2);
308         break;
309     case 7: // EN8
310         resetujBit(PORTC, PC4);
311         break;
312     case 8: // EN9
313         resetujBit(PORTC, PC5);
314         break;
315     }
316   }
317 }

318 // Potvrda prijema poruke
319 for (int i = 0; i < duzina_dolazne_poruke - 3; i++) // Kopiranje dolazne
320   poruke u odlaznu
321 {
322   odlazna_poruka[i] = dolazna_poruka[i];
323 }
324 duzina_odlazne_poruke = duzina_dolazne_poruke - 3; // Funkcija
325   pisiSerial() dodaje terminator, zato umanjivanje za 3
326 }

327 ISR(TIMER2_OVF_vect) // Prekidna rutina koja se pokrece svakih 125us, pa se
328   koristi za vremensko tempiranje FLASH i STROBE kanala
329 {
330   osmine++;
331   if (osmine == 8) // Da li je proslo osam osmina ili 1ms
332   {
333     osmine = 0;
334     for (int i = 0; i < 9; i++)
335     {
336       // TRIGGER - Ako je kanal u FLASH modu kojem jos uvijek aktivan
337       // nalog za impuls
338       if (trigeri[i])
339       {
340         if (brojaci_ms[i] < odgoda_period[i]) // Dok ne istekne vrijeme
341           odgode (delay) resetuje se izlaz (setuje pin zbog inverzne
342           logike)
343         {
344           switch (i)
345           {
346             case 0: // EN1
347               setujBit(PORTD, PD2);
348               break;
349             case 1: // EN2
350               setujBit(PORTD, PD4);
351               break;

```

```

348     case 2: // EN3
349         setujBit(PORTB, PB4);
350         break;
351     case 3: // EN4
352         setujBit(PORTB, PB5);
353         break;
354     case 4: // EN5
355         setujBit(PORTC, PC0);
356         break;
357     case 5: // EN6
358         setujBit(PORTC, PC1);
359         break;
360     case 6: // EN7
361         setujBit(PORTC, PC2);
362         break;
363     case 7: // EN8
364         setujBit(PORTC, PC4);
365         break;
366     case 8: // EN9
367         setujBit(PORTC, PC5);
368         break;
369     }
370 }
371 else if (brojaci_ms[i] < odgoda_period[i] + trajanje_impulsa[i]
372     ] // Dok traje impuls setuje se izlaz (resetuje pin zbog
373     inverzne logike)
374 {
375     switch (i)
376     {
377         case 0: // EN1
378             resetujBit(PORTD, PD2);
379             break;
380         case 1: // EN2
381             resetujBit(PORTD, PD4);
382             break;
383         case 2: // EN3
384             resetujBit(PORTB, PB4);
385             break;
386         case 3: // EN4
387             resetujBit(PORTB, PB5);
388             break;
389         case 4: // EN5
390             resetujBit(PORTC, PC0);
391             break;
392         case 5: // EN6
393             resetujBit(PORTC, PC1);
394             break;
395         case 6: // EN7
396             resetujBit(PORTC, PC2);
397             break;
398         case 7: // EN8
399             resetujBit(PORTC, PC4);
400             break;
401         case 8: // EN9
402             resetujBit(PORTC, PC5);
403             break;
404     }
405 }
```

```

404     else
405     {
406         switch (i)
407         {
408             case 0: // EN1
409                 setujBit(PORTD, PD2);
410                 break;
411             case 1: // EN2
412                 setujBit(PORTD, PD4);
413                 break;
414             case 2: // EN3
415                 setujBit(PORTB, PB4);
416                 break;
417             case 3: // EN4
418                 setujBit(PORTB, PB5);
419                 break;
420             case 4: // EN5
421                 setujBit(PORTC, PC0);
422                 break;
423             case 5: // EN6
424                 setujBit(PORTC, PC1);
425                 break;
426             case 6: // EN7
427                 setujBit(PORTC, PC2);
428                 break;
429             case 7: // EN8
430                 setujBit(PORTC, PC4);
431                 break;
432             case 8: // EN9
433                 setujBit(PORTC, PC5);
434                 break;
435         }
436         trigeri[i] = false; // Flag za triger se resetuje jer je
437                         zavrsio impuls
438         brojaci_ms[i] = 0; // Brojac se resetuje jer je zavrsio
439                         impuls
440     }
441
442     // STROBE - Ako je kanal u STROBE modu
443     else if (modovi[i] == 'D')
444     {
445         if (brojaci_ms[i] < trajanje_imulta[i]) // Dok traje impuls
446             setuje se izlaz (resetuje pin zbog inverzne logike)
447         {
448             switch (i)
449             {
450                 case 0: // EN1
451                     resetujBit(PORTD, PD2);
452                     break;
453                 case 1: // EN2
454                     resetujBit(PORTD, PD4);
455                     break;
456                 case 2: // EN3
457                     resetujBit(PORTB, PB4);
458                     break;
459                 case 3: // EN4

```

```

459         resetujBit (PORTB, PB5);
460         break;
461     case 4: // EN5
462         resetujBit (PORTC, PC0);
463         break;
464     case 5: // EN6
465         resetujBit (PORTC, PC1);
466         break;
467     case 6: // EN7
468         resetujBit (PORTC, PC2);
469         break;
470     case 7: // EN8
471         resetujBit (PORTC, PC4);
472         break;
473     case 8: // EN9
474         resetujBit (PORTC, PC5);
475         break;
476     }
477 }
478 else // Dok traje ostatak perioda resetuje se izlaz (setuje pin
        zbog inverzne logike)
479 {
480     switch (i)
481     {
482     case 0: // EN1
483         setujBit (PORTD, PD2);
484         break;
485     case 1: // EN2
486         setujBit (PORTD, PD4);
487         break;
488     case 2: // EN3
489         setujBit (PORTB, PB4);
490         break;
491     case 3: // EN4
492         setujBit (PORTB, PB5);
493         break;
494     case 4: // EN5
495         setujBit (PORTC, PC0);
496         break;
497     case 5: // EN6
498         setujBit (PORTC, PC1);
499         break;
500     case 6: // EN7
501         setujBit (PORTC, PC2);
502         break;
503     case 7: // EN8
504         setujBit (PORTC, PC4);
505         break;
506     case 8: // EN9
507         setujBit (PORTC, PC5);
508         break;
509     }
510 }
511 brojaci_ms[i]++;
512 if (brojaci_ms[i] == odgoda_period[i]) // Brojac se resetuje
        kad odbroji period
513     brojaci_ms[i] = 0;
514 }
```

```

515     }
516 }
517 }
518
519 void trigerujFlash() // Postavljanje flagova za okidanje kanala u FLASH
520     modu (TRIGGER)
521 {
522     // Stvarno okidanje se vrsti unutar prekidne rutine ISR
523     for (int i = 0; i < 9; i++)
524     {
525         if (modovi[i] == 'C')
526             trigeri[i] = true;
527         else
528             trigeri[i] = false;
529     }
530
531     // Potvrda prijema poruke
532     for (int i = 0; i < duzina_dolazne_poruke - 3; i++) // Kopiranje dolazne
533         poruke u odlaznu
534     {
535         odlazna_poruka[i] = dolazna_poruka[i];
536     }
537     duzina_odlazne_poruke = duzina_dolazne_poruke - 3; // Funkcija
538         pisiSerial() dodaje terminator, zato umanjivanje za 3
539 }
540
541 void resetujKanale() // Iskljucivanje svih kanala (RESET)
542 {
543     // Postavljanje Duty Cycle-ova za SOURCE kanale na nulu
544     // Komplement se radi zbog invertiranog PWM-a
545     OCR2B = 255; // PWM1
546     OCR0B = 255; // PWM2
547     OCR0A = 255; // PWM3
548     OCR1B = 255; // PWM4
549     OCR2A = 255; // PWM5
550
551     // Postavljanje modova SINK kanala na OFF i resetovanje
552     for (int i = 0; i < 9; i++)
553     {
554         modovi[i] = 'A';
555         trajanje_impulsa[i] = 0;
556         odgoda_period[i] = 0;
557     }
558     // Uklanjanje aktivnih trigera
559     for (int i = 0; i < 9; i++)
560         trigeri[i] = false;
561     // Resetovanje SINK kanala (setovanje pinova zbog inverzne logike)
562     setujBit(PORTD, PD2); // EN1
563     setujBit(PORTD, PD4); // EN2
564     setujBit(PORTB, PB4); // EN3
565     setujBit(PORTB, PB5); // EN4
566     setujBit(PORTC, PC0); // EN5
567     setujBit(PORTC, PC1); // EN6
568     setujBit(PORTC, PC2); // EN7
569     setujBit(PORTC, PC4); // EN8
570     setujBit(PORTC, PC5); // EN9
571
572     // Potvrda prijema poruke

```

```

570     for (int i = 0; i < duzina_dolazne_poruke - 3; i++) // Kopiranje dolazne
571         poruke u odlaznu
572     {
573         odlazna_poruka[i] = dolazna_poruka[i];
574     }
575     duzina_odlazne_poruke = duzina_dolazne_poruke - 3; // Funkcija
576         pisiSerial() dodaje terminator, zato umanjivanje za 3
577 }
578
579 void procesirajPoruku() // Interpretacija dolazne poruke
580 {
581     odlazna_poruka[3] = dolazna_poruka[3];
582     duzina_odlazne_poruke++;
583     // Provjera pocetka i kraja poruke
584     if (dolazna_poruka[0] != '#' || dolazna_poruka[1] != '#' ||
585         dolazna_poruka[duzina_dolazne_poruke - 3] != '\r' || dolazna_poruka[
586         duzina_dolazne_poruke - 2] != '\r' || dolazna_poruka[
587         duzina_dolazne_poruke - 1] != '\n')
588     {
589         greska('F');
590         return;
591     }
592     // Provjera adresu
593     if (dolazna_poruka[2] != adresa)
594     {
595         {
596             slanje_flag = false;
597             if (dolazna_poruka[2] != broadcast) // Ako je broadcast nastavlja
598                 se ostatak procesa, ako ne zanemaruje se
599             {
600                 return;
601             }
602         }
603     }
604     // Provjera zadane funkcije
605     switch (dolazna_poruka[3])
606     {
607         case 'A': // Konfiguracija intenziteta SOURCE kanala
608             konfigurirajSourceove();
609             break;
610         case 'B': // Konfiguracija modova i parametara SINK kanala
611             konfigurirajSinkove();
612             break;
613         case 'T': // Okidanje (TRIGGER)
614             trigerujFlash();
615             break;
616         case 'R': // Iskljucivanje svih kanala (RESET)
617             resetujKanale();
618             break;
619         default: // Nepoznata komanda
620             greska('K');
621             break;
622     }
623 }
624
625 void setup()
626 {
627     // Ucitavanje adrese

```

```
622     adresiraj();
623     // Inicijalizacija uredjaja
624     inicijalizirajPWM();
625     inicijalizirajSerial();
626     inicijalizirajIzlaze();
627     inicijalizirajPrekide();
628     // Inicijalizacija zaglavlja odlazne poruke
629     odlazna_poruka[0] = '#';
630     odlazna_poruka[1] = '#';
631     odlazna_poruka[2] = adresa;
632     duzina_odlazne_poruke = 3;
633     // Resetovanje flagova
634     greska_flag = false;
635     slanje_flag = true;
636     procesiranje_flag = true;
637 }
638
639 void loop()
{
640     //UCSR0C |= (1 << UMSEL00) | (1 << UMSEL01);
641     if (Serial.available()) {
642         citajSerial();
643         procesirajPoruku();
644         if (slanje_flag)
645             pisiSerial();
646         duzina_odlazne_poruke = 3;
647         // Resetovanje flagova
648         slanje_flag = true;
649         procesiranje_flag = true;
650         greska_flag = false;
651     }
652 }
```

# Literatura

- [1] ATmega328/P 8-bit AVR Microcontroller, Atmel Corporation, 2016, datum pristupa: 17. januar 2017., dostupno na: <http://www.atmel.com/>
- [2] Torras, C., Computer Vision: Theory and Industrial Applications. Springer-Verlag, 1992.
- [3] Sun, D. W., Computer Vision Technology in the Food and Beverage Industries. Elsevier Science, 2012.
- [4] Demant, C., Streicher-Abel, B., Garnica, C., Industrial Image Processing: Visual Quality Control in Manufacturing. Springer Berlin Heidelberg, 2013.
- [5] Martin, D., A Practical Guide to Machine Vision Lighting, Advanced Illumination, 2012, datum pristupa: 4. januar 2017., dostupno na: <http://www.advancedillumination.com/>
- [6] Triniti Technology: Intelligent lighting for Machine Vision, Gardasoft Vision Ltd., 2016, datum pristupa: 6. januar 2017., dostupno na: <http://www.gardasoft.com/>
- [7] ULC-2, Universal LED Controller, Metaphase Technologies, Inc., 2016, datum pristupa: 6. januar 2017., dostupno na: <http://www.metaphase-tech.com/>
- [8] DCS-100E, Single Output Controller/3 Channels per Output, Advanced Illumination Inc., 2015, datum pristupa: 6. januar 2017., dostupno na: <http://www.advancedillumination.com/>
- [9] Kreso, S., Huseinbegović, S., Projektovanje mikroprocesorskih sistema. Elektrotehnički fakultet Sarajevo, 2012.
- [10] Arduino Uno Reference Design, Rev. 3, Arduino, 2016, datum pristupa: 17. januar 2017., dostupno na: [https://www.arduino.cc/en/uploads/Main/Arduino\\_Unc\\_Rev3-schematic.pdf](https://www.arduino.cc/en/uploads/Main/Arduino_Unc_Rev3-schematic.pdf)
- [11] Stojanović, Z., Arhitektura, protokoli i servisi interneta. Evropski univerzitet Brčko distrikta, 2015.
- [12] Perrin, B., The Art and Science of RS-485, Circuit Cellar, 1999.
- [13] MAX481/MAX483/MAX485/MAX487/MAX491/MAX1487 Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers, Maxim Integrated, 2014, datum pristupa: 9. mart 2017., dostupno na: <http://www.maximintegrated.com./>
- [14] SNx4HC165 8-Bit Parallel-Load Shift Registers, Texas Instruments, 2017, datum pristupa: 9. mart 2017., dostupno na: <http://www.ti.com/>
- [15] Schubert, E., Light-Emitting Diodes. Cambridge University Press, 2006.

- [16] Bergin, T., Cusack, J., DeSmet, K., Advantages of LED Lighting in Vision Inspection Systems, QuadTech, Inc., 2010.
- [17] Specification for White LED NSDW570GS-K1, NICHIA Corporation, 2012, datum pristupa: 18. maj 2017., dostupno na: <http://www.nichia.com/>
- [18] Halbritter, H., Jäger, C., Weber, R., Schwind, M., Möllmer, F., "High-speed led driver for ns-pulse switching of high-current leds", IEEE Photonics Technology Letters, Vol. 26, No. 18, Sept. 2014, str. 1871-1873.
- [19] Vo, J., AN-903 A Comparison of Differential Termination Techniques, National Semiconductor, 1993.