

# Izrada sustava automatske ograde

---

**Begović, Jakov**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:211:058582>

*Rights / Prava:* [Attribution-NonCommercial-ShareAlike 3.0 Unported / Imenovanje-Nekomercijalno-Dijeli pod istim uvjetima 3.0](#)

*Download date / Datum preuzimanja:* **2025-01-27**



*Repository / Repozitorij:*

[Faculty of Organization and Informatics - Digital Repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN

Jakov Begović

IZRADA SUSTAVA AUTOMATSKE OGRADE

ZAVRŠNI RAD

Varaždin, 2024.

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ź D I N**

**Jakov Begović**

**Matični broj: 00161526238**

**Studij: Informacijski sustavi**

**IZRADA SUSTAVA AUTOMATSKE OGRADE**

**ZAVRŠNI RAD**

**Mentor:**

Prof. dr. sc. Ivan Magdalenić

**Varaždin, lipanj 2024.**

*Jakov Begović*

### **Izjava o izvornosti**

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi*

---

Zahvaljujem svima koji su mi pomogli u prikupljanju informacija za i u izradi ovog rada.  
Posebice zahvaljujem svom mentoru prof. dr. sc. Ivanu Magdaleniću, dr. sc. Borisu Tomašu,  
asis. mag. inf. Lovri Posariću te gospodinu Đuri Mikiću.

## Sažetak

Područja Interneta stvari i automatizacije rastu u opsegu i svrsi iz godine u godinu. Njihova prisutnost se približava neizbježnoj. Od poljoprivrede potpomognute IoT mrežom, preko polu-automatiziranih tvornica do pametnih gradova. Ovaj rad istražuje dubinu i kompleksnost automatiziranog IoT sustava usmjerenog korisniku. Istražuje načine prikupljanja znanja i pristupe razvoju sustava automatske ograde. Sustav ispunjava 7 zahtjeva vezanih uz funkcionalnost i sigurnost. Kako bi se zahtjevi ispunili razvijena je elektronička shema, računalni program mikrokontrolera i računalni program Android mobilnog uređaja. Mikrokontroler upravlja sustavom i nadzire ga. Mobilni uređaj omogućuje korisničku komunikaciju s mikrokontrolerom i samim time upravljanje nad sustavom. Njihova komunikacija se vrši putem Bluetooth Low Energy protokola. Izazov automatiziranih sustava, kao što je prikazano radom, je robusnost, pouzdanost i sigurnost. Detalji mogu dovesti do njegovog kvara. Oni mogu biti previđeni u razvoju što doводи do nesigurnosti ili opasnosti. Samim time se postavlja pitanje: "kako razviti što sigurniji i robusniji sustav?"

**Ključne riječi:** računalno inženjerstvo, elektroničko inženjerstvo, Internet stvari, automatizacija, ugrađeni sustav, mobilni uređaj, Bluetooth Low Energy

# Sadržaj

<b>1. Uvod</b>	1
1.1. Opis zadatka	1
1.2. Značajnost teme	1
<b>2. Metode i tehnike rada</b>	2
2.1. Izvori znanja	2
2.2. Alati za dizajn računalnih programa	3
2.3. Alati za dizajn elektroničkih krugova	3
<b>3. Razrada teme</b>	4
3.1. Zahtjevi nad sustavom	4
3.2. pristupi ispunjenju zahtjeva	4
3.2.1. Mogućnost upravljanja brzinom motora	4
3.2.2. Mogućnost upravljanja smjerom okretanja motora	4
3.2.3. Mogućnost slanja naredbi koristeći mobilni uređaj	4
3.2.4. Mogućnost trajne pohrane podataka o pokretanju motora	5
3.2.5. Mogućnost prikupljanja i prikaza podataka o pokretanju motora	5
3.2.6. Mogućnost registriranja krajnje pozicije ograde	5
3.2.7. Mogućnost detekcije potencijalne kolizije	5
3.3. Modeliranje elektroničkih krugova	6
3.3.1. Shema sustava	6
3.3.2. Transformatori i ispravljači	7
3.3.3. Mosni ispravljač	8
3.3.3.1. Vrijednost rezultantnog napona	8
3.3.3.2. Filtriranje napona	8
3.3.3.3. Pražnjenje kondenzatora	9
3.3.4. Upravljač brzinom	10
3.3.4.1. PWM signali	10
3.3.4.2. MOSFET	11
3.3.4.3. Pozicija MOSFET-a u krugu	11
3.3.4.4. <i>Pull-down</i> otpornik	12
3.3.4.5. Podizanje naponske razine PWM signala	12
3.3.5. Upravljač smjerom	13
3.3.5.1. Bipolarni tranzistor	14
3.3.5.2. Princip rada NPN tranzistora	14
3.3.5.3. Struja kroz tranzistor	14

---

3.3.5.4.	<i>Pull-down</i> otpornik . . . . .	15
3.3.6.	Releji i prekidači . . . . .	15
3.3.6.1.	Releji . . . . .	15
3.3.6.2.	Poredna dioda . . . . .	16
3.3.6.3.	Prekidači . . . . .	16
3.3.6.4.	Naponsko djelilo . . . . .	16
3.3.7.	Motor . . . . .	19
3.3.7.1.	Tip motora . . . . .	19
3.3.7.2.	Snaga motora . . . . .	19
3.3.7.3.	Specifikacije korištenog motora . . . . .	20
3.3.7.4.	Promjena smjera rotacije . . . . .	20
3.3.8.	Odašiljač i prijemnik infracrvenih zraka . . . . .	20
3.3.9.	SD kartica . . . . .	21
3.3.9.1.	SPI protokol . . . . .	22
3.3.10.	Mikrokontroler . . . . .	22
3.4.	Modeliranje računalnog programa mikrokontrolera . . . . .	23
3.4.1.	Razvojna okolina . . . . .	23
3.4.2.	Struktura programa . . . . .	23
3.4.3.	Datoteka platformio.ini . . . . .	23
3.4.4.	Datoteka main.cpp . . . . .	24
3.4.4.1.	Prekidi . . . . .	24
3.4.5.	Datoteka BLEController.h . . . . .	25
3.4.5.1.	Komunikacijski protokol Bluetooth Low Energy . . . . .	25
3.4.5.2.	Postavljanje BLE servera . . . . .	25
3.4.5.3.	Funkcija povratnog poziva karakteristike piši . . . . .	26
3.4.5.4.	Funkcija povratnog poziva karakteristike čitaj . . . . .	26
3.4.6.	Datoteka SDCommunicator.h . . . . .	27
3.4.7.	Datoteka ElectronicsControl.h . . . . .	28
3.4.7.1.	Prekidi . . . . .	28
3.4.7.2.	Konstruktor . . . . .	28
3.4.7.3.	Postavljanje brzine i smjera . . . . .	29
3.4.7.4.	Trenutno zaustavljanje struje kroz motor . . . . .	29
3.4.8.	Datoteka WS2812BControl.h . . . . .	29
3.5.	Modeliranje računalnog programa Android uređaja . . . . .	30
3.5.1.	Razvojna okolina . . . . .	30
3.5.2.	Struktura programa . . . . .	30
3.5.3.	Tok korisnika . . . . .	31
3.5.4.	Načini ostvarenja funkcionalnosti . . . . .	37
3.5.4.1.	Traženje dopuštenja . . . . .	37
3.5.4.2.	Uključivanje Bluetootha . . . . .	37
3.5.4.3.	Skeniranje uređaja . . . . .	38
3.5.4.4.	Povezivanje na BLE server . . . . .	38
3.5.4.5.	Ispis karakteristika . . . . .	38



3.5.4.6. Primanje i ispisivanje podataka . . . . .	38
3.5.4.7. Određivanje brzine . . . . .	39
3.5.4.8. Prekid veze s uređajem . . . . .	40
<b>4. Zaključak . . . . .</b>	<b>41</b>
<b>Popis literature . . . . .</b>	<b>46</b>
<b>Popis slika . . . . .</b>	<b>47</b>

# 1. Uvod

## 1.1. Opis zadatka

Izrada sustava automatske ograde. Preko Dasduino mikrokontrolera se kontrolira električni motor. Naredbe za otvaranje i zatvaranje ograde se vrše putem mrežnih protokola. Mikrokontroler sprema podatke o otvaranju i zatvaranju ograde. Te podatke je moguće grafički prikazati preko mobilne aplikacije za Android uređaj. Rad sadrži opis korištenih tehnologija i prikaz i opis rješenja. Po potrebi sadrži elektroničke sheme, slike i popratni video.

U ovom se radu razrađuje sustav automatizirane ograde za ograde s kliznim vratima. Na ovoj se pretpostavci temelje svi pristupi pokretanju i sigurnosti.

U njemu se pokušava pristupiti razvoju automatizirane ograde iz perspektive tehnologija automatizacije i Interneta stvari. IoT tehnologije su tehnologije koje stapaju djelatnosti informatike, računalstva, elektronike i elektrotehnike. Samim time, ovaj će rad sadržavati praktični dio programiranja sustava automatske ograde te komunikacije i upravljanja s njim. Uz to, sadržavat će dizajn sustava za upravljanje električnim motorom. U taj sustav će biti ugrađen mikrokontroler koji će omogućavati upravljanje i komunikaciju sa sustavom.

## 1.2. Značajnost teme

21. stoljeće sa sobom nosi nove tehnologije, naročito na području informatike, računalstva i automatizacije. Značajnost tih tehnologija je sveprisutna - one prodiru kroz sve slojeve društva. Od poljoprivrede potpomognute IoT mrežom [1], preko polu-automatiziranih tvornica [2] do pametnih gradova [3].

Tehnologija automatiziranih ograda nalazi svoje korijene u 20. stoljeću [4]. Ona više nije ekskluzivna za većinu modernog društva [5]. Sve više i više kućanstava ju prihvaća i koristi. Ona je danas neizbježna za tvornice nižih i viših razina automatizacije, tehnološki obogaćena kućanstva, prometna čvorišta i slično.

## 2. Metode i tehnike rada

### 2.1. Izvori znanja

Ova tema pokriva područja elektroničkog inženjerstva i informatičkog inženjerstva. Sva-ko od navedenih zahtjeva drugačiji pristup prikupljanju informacija.

Prikupljanje informacija o informatičkom inženjerstvu sam vršio s pomoću izvora:

- Dokumentacija programskog jezika
- Dokumentacija razvojne okoline
- Tuđa rješenja i pristupi
- Objave na web stranicama GitHub, StackOverflow, YouTube
- Drugi izvori kreirani od strane informatičkih inženjera
- Konzultacije s chatbot-om i virtualnim asistentom ChatGPT-em
- Konzultacije s mag. inf. Lovrom Posarićem

Prikupljanje informacija o elektroničkom inženjerstvu sam vršio s pomoću izvora:

- Tehnička uputstva elektroničkih proizvoda
- Blogovi
- Objave na web stranicama YouTube, Electronics Stack Exchange
- Falstad simulator elektroničkih krugova [6]
- Materijali za predmet "Osnove elektrotehnike" Fakulteta elektrotehnike i računalstva
- Prva i druga knjiga "Osnove elektrotehnike" Viktora Pintera [7]
- Konzultacije s mentorom, profesorom dr. sc. Ivanom Magdalenićem
- Konzultacije s chatbot-om i virtualnim asistentom ChatGPT-em
- Knjiga "The Art of Electronics" Paula Horowitz i Winfielda Hilla [8]
- Konzultacije s radnicima u struci

## **2.2. Alati za dizajn računalnih programa**

Razvijena su dva računalna programa.

Računalni program razvijen za mikrokontroler Dasduino CONNECTPLUS napisan je u C++ programskom jeziku. Razvijen je s pomoću programa Visual Studio Code koristeći PlatformIO ekstenziju. U svrhu njegovog testiranja korištena je mobilna aplikacija nRF Connect tvrtke Nordic Semiconductor.

Računalni program razvijen za Android mobilni uređaj napisan je u Kotlin programskom jeziku. Razvijen je s pomoću programa Android Studio.

## **2.3. Alati za dizajn elektroničkih krugova**

Alati korišteni za dizajn elektroničkih krugova su Easy EDA u Professional verziji i Falstad simulator el. krugova [6].

## **3. Razrada teme**

### **3.1. Zahtjevi nad sustavom**

Zahtjevi nad sustavom su ograničeni opsegom ovog završnog rada te s vremenom koje je dodijeljeno za njegovu izradu. Rad prikazuje temeljne principe rada automatizirane ograde.

Zahtjevi nad sustavom u opsegu ovog završnog rada su:

1. Mogućnost upravljanja brzinom motora
2. Mogućnost upravljanja smjerom okretanja motora
3. Mogućnost upravljanja koristeći mobilni uređaj
4. Mogućnost trajne pohrane podataka o pokretanju motora
5. Mogućnost prikupljanja i prikaza podataka o pokretanju motora
6. Mogućnost registriranja krajnje pozicije ograde
7. Mogućnost detekcije potencijalne kolizije

### **3.2. Pristupi ispunjenju zahtjeva**

#### **3.2.1. Mogućnost upravljanja brzinom motora**

Brzina motora se može regulirati na različite načine. Oni ovise o tipu motora koji se koristi. U ovom radu se koristi univerzalni motor. Njegovom brzinom se upravlja s pomoću modulacije širine impulsa (engl. *pulse width modulation*, akr. PWM).

#### **3.2.2. Mogućnost upravljanja smjerom okretanja motora**

Kada se polaritet statora univerzalnog motora obrne, obrne se i smjer rotacije njegove osovine. Samim time, kako bi upravljanje smjerom okretanja motora bilo moguće, izrađen je sustav gdje se polaritet vodova iz upravljačke ploče lako izmijeni.

#### **3.2.3. Mogućnost slanja naredbi koristeći mobilni uređaj**

Mikrokontroler upravljačke ploče izabran je kako bi podržavao mogućnost komunikacije putem BLE protokola. Takav pristup komunikaciji izrazito je fleksibilan i lako proširiv u svojim značajkama. Za mobilni uređaj operacijskog sustava Android 13 razvijena je aplikacija za komunikaciju s mikrokontrolerom.

### **3.2.4. Mogućnost trajne pohrane podataka o pokretanju motora**

Mikrokontroler je spojen na modul za komunikaciju sa SD karticom. Ona predstavlja trajnu memoriju sustava.

### **3.2.5. Mogućnost prikupljanja i prikaza podataka o pokretanju motora**

U SD karticu se upisuju trenutci pokretanja motora. Na zahtjev se šalju mobilnom uređaju putem BLE protokola. Ispisuju se na sučelju mobilnog uređaja.

### **3.2.6. Mogućnost registriranja krajnje pozicije ograde**

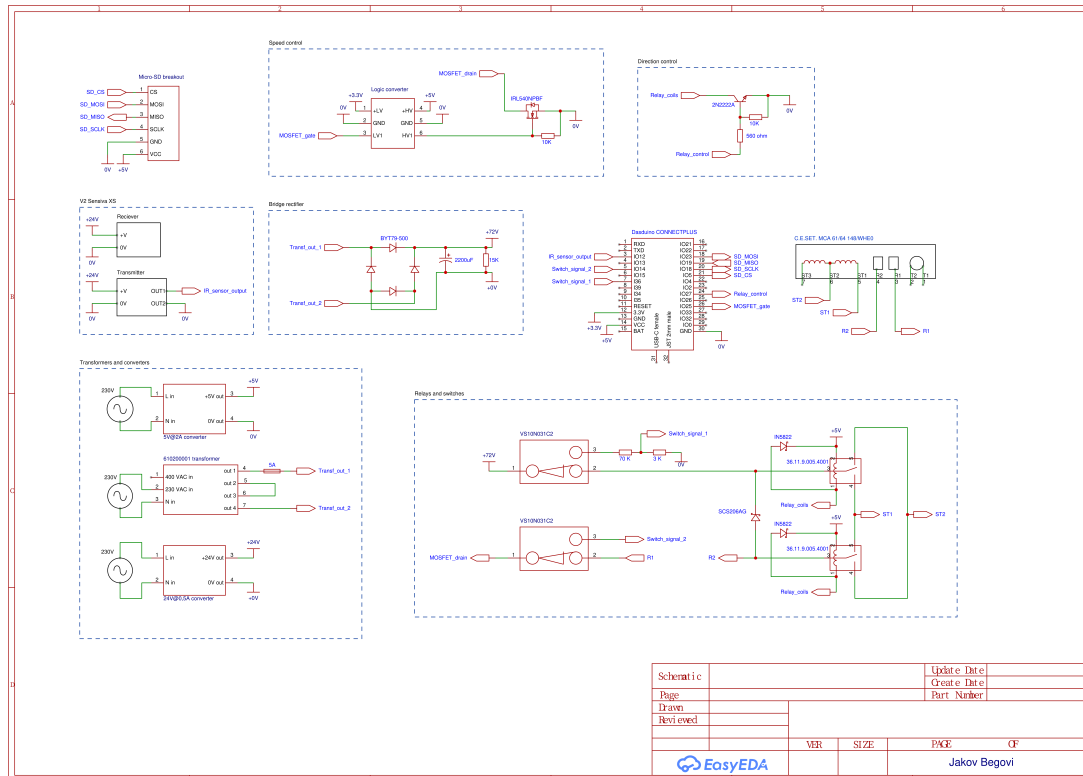
Umetnuta su 2 mehanička prekidača koji pri pritisku prekidaju protok struje kroz motor. Oni se ugrađuju na pozicije potpuno otvorenih i potpuno zatvorenih vrata ograde. Mikrokontroleru se šalje signal u trenutku pritiska. Signali prekidača su diferencirani.

### **3.2.7. Mogućnost detekcije potencijalne kolizije**

Ugrađen je odašiljač i prijemnik infracrvenih zraka dizajniran za korištenje na otvorenom. Oni se ugrađuju tako da infracrvene zrake strujaju paralelno uz klizna vrata ograde. Mikrokontroleru se šalje signal u trenutku prekida toka zraka.

## 3.3. Modeliranje elektroničkih krugova

### 3.3.1. Shema sustava

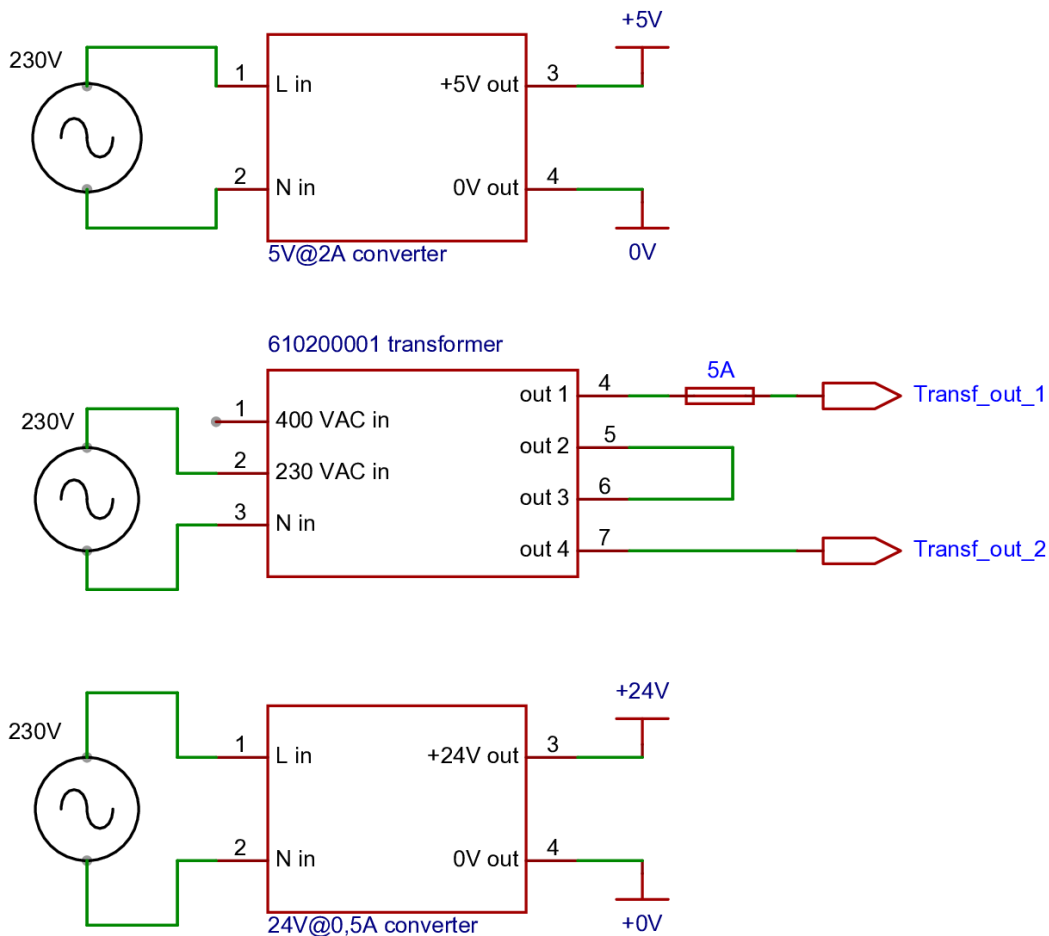


Slika 1: Shema sustava

Na kompletnoj shemi sustava (slika 1) moguće je vidjeti sve komponente i poveznice sustava. Kompletna shema je podijeljena u 9 cjelina:

1. Transformatori i ispravljači
2. Mosni ispravljač
3. Upravljač brzinom
4. Upravljač smjerom
5. Releji i prekidači
6. Motor
7. Odašiljač i prijemnik infracrvenih zraka
8. Čitač SD kartice
9. Mikrokontroler

### 3.3.2. Transformatori i ispravljači



Slika 2: Shema transformatora i ispravljača

Svaka od prikazanih komponenti ove cjeline namijenjena je za spajanje na gradsku mrežu: 230 V izmjenične struje. Njihov zadatak je ispravljanje 230 VAC u istosmjernu struju prilagođenih napona.

Ispravljač 5 voltnim izlazom koristi za napajanje mikrokontrolera, pokretanje MOSFETA, releja i čitača SD kartica. Svaki od ovih elemenata namijenjen je ili izabran kako bi funkcionirao upravo na ovoj naponskoj razini.

Važno je napomenuti kako pri povezivanju svih tih komponenti na isti 5 V izvor stvara pad napona za oko pola volta. Taj pad je neznačajan za ovaj sustav. Sve komponente pravilno funkcioniraju i u vidu tog slučaja.

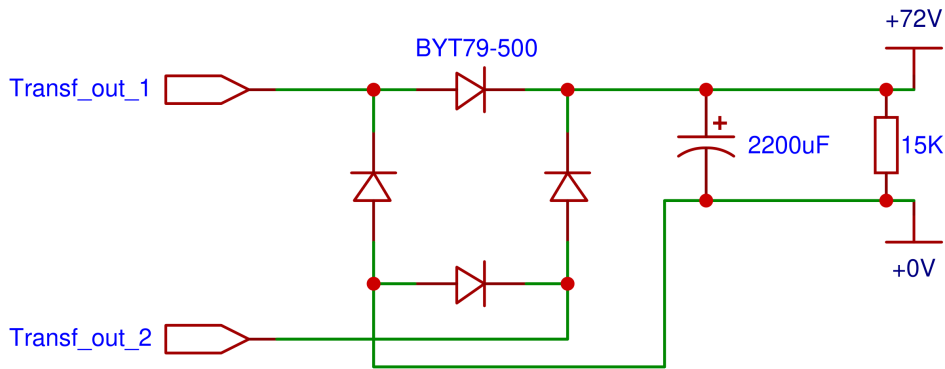
Ispravljač s 24 voltnim izlazom namijenjen je za napajanje isključivo odašiljača i prijemnika infracrvenih zraka. Također se mogao koristiti pretvarač istosmjerne struje.

Prema njegovom tehničkom uputstvu, korišteni transformator pretvara 230 V izmjenične struje u 48 V izmjenične struje s maksimalnom jačinom električne struje od 4,17 A [9]. Model ugrađen u stvarni sustav pruža 52 V izmjenične struje nakon transformacije. Kako bi se mo-



torom moglo upravljati PWM signalima potrebno je ispraviti izmjeničnu u istosmjernu struju. Mosni ispravljač je iskorišten u tu svrhu.

### 3.3.3. Mosni ispravljač



Slika 3: Shema mosnog ispravljača

Mosni ispravljač (engl. *bridge rectifier*) koristi se za ispravljanje izmjenične struje u istosmjernu [10]. Rezultat tog ispravljanja su valovi pozitivnog napona.

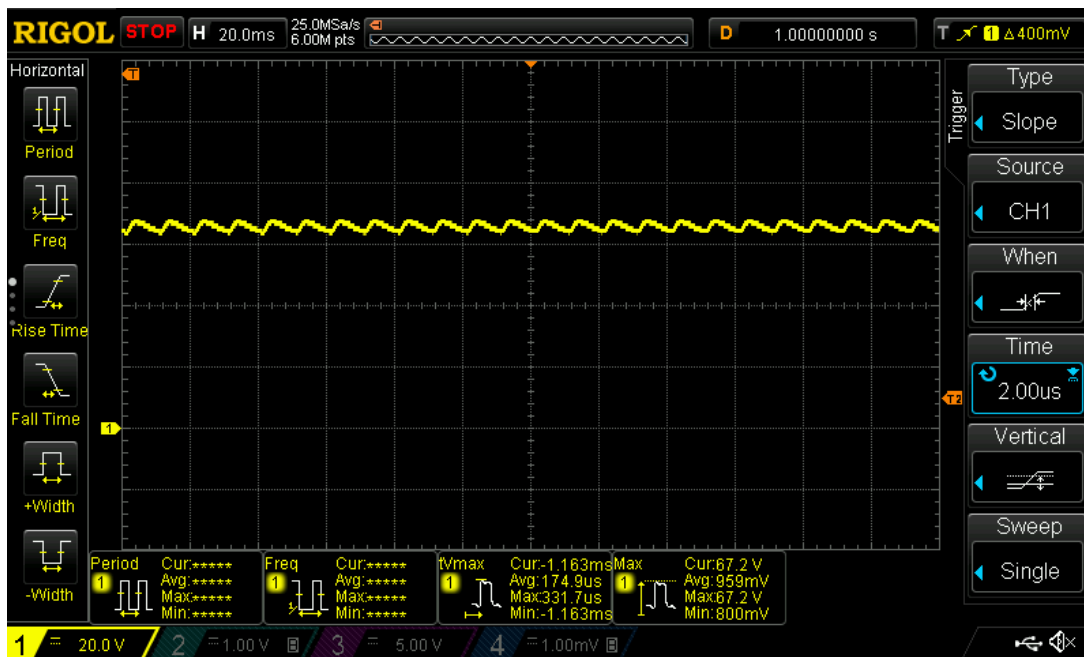
#### 3.3.3.1. Vrijednost resultantnog napona

Pretvaranje izmjenične u istosmjernu struju za sobom povlači i promjenu razine napona. Pretvaranje 230 V izmjenične struje gradske mreže u 52 V izmjenične struje znači da su amplitude valova pale s oko  $230 \times \sqrt{2} = 325V$  na  $52 \times \sqrt{2} = 73,5V$ . Možemo zaključiti da valovi pozitivnog napona iz ispravljača imaju amplitudu vrijednosti 73,5 V.

Treba napomenuti kako postoji određeni pad napona preko dioda koje ispravljaju struju. U ovom su krugu za ispravljanje korištene diode BYT79-500 proizvođača WeEn. Razina pada napona kroz diodu tog modela seže do 1,05 V [11]. U stvarnom sustavu iz ispravljača teče struja s amplitudom vrijednosti 72 V.

#### 3.3.3.2. Filtriranje napona

Na izlazu iz ispravljača se ugrađuje krug za filtriranje kako bi napon imao stalnu vrijednost [12]. Tu funkciju u ovom krugu vrši polarizirani kondenzator kapaciteta 2200 uF. Resultantna vrijednost napona nije potpuno konstantna, no njezine oscilacije su minimalne. Kod spajanja ispravljača na motor dobiva se pad napona kroz motor koji je prikazan na slici 4.



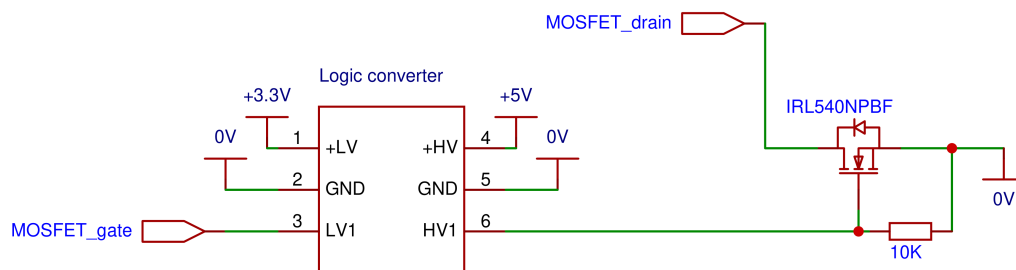
Slika 4: Pad napona iz mosnog ispravljača kroz motor

Vidljivo je da napon pada i raste u periodama od oko 10 milisekundi. Naime, pošto se transformira napon iz gradske mreže, čija je frekvencija 50 Hz, 50 puta u sekundi će izmjenični napon gradske mreže dostići svoju amplitudu. Frekvencija dostizanja amplitude će se udvostručiti nakon ispravljanja negativnog vala. Frekvencija tada postaje 100 Hz i vrijeme između dvije amplitude je  $1 \div 100 = 0,01$  sekunda. Odnosno, 10 milisekundi, kao što je i vidljivo na slici 4. Pad napona između dvije amplitude je minimalan i samim time neznačajan.

### 3.3.3.3. Pražnjenje kondenzatora

Uprava za sigurnost i zdravlje na radu Ministarstva rada SAD-a (engl. *Occupational Safety and Health Administration*, akr. OSHA) zahtjeva da dijelove električne opreme koji rade na 50 volta ili više treba zaštititi od slučajnog dodira korištenjem odobrenih kućišta ili na druge specificirane načine [13]. Naime, kondenzatori ovakvih kapaciteta i naponskih razina predstavljaju stvaran rizik od električnog šoka i štete ljudskom tijelu [14]. U tu je svrhu paralelno s kondenzatorom spojen otpornik od 15 K $\Omega$ . Njegova korist je pražnjenje kondenzatora nakon isključenja napona do transformatora. Vrijednost otpornika izabrana je kako bi se kondenzator u što kraćem vremenskom roku ispraznio te da otpornik troši što manje snage pri uobičajenom radu sustava. Prema testiranjima, razlika u naponu kroz motor sa i bez otpornika 15 K $\Omega$  je minimalna.

### 3.3.4. Upravljač brzinom



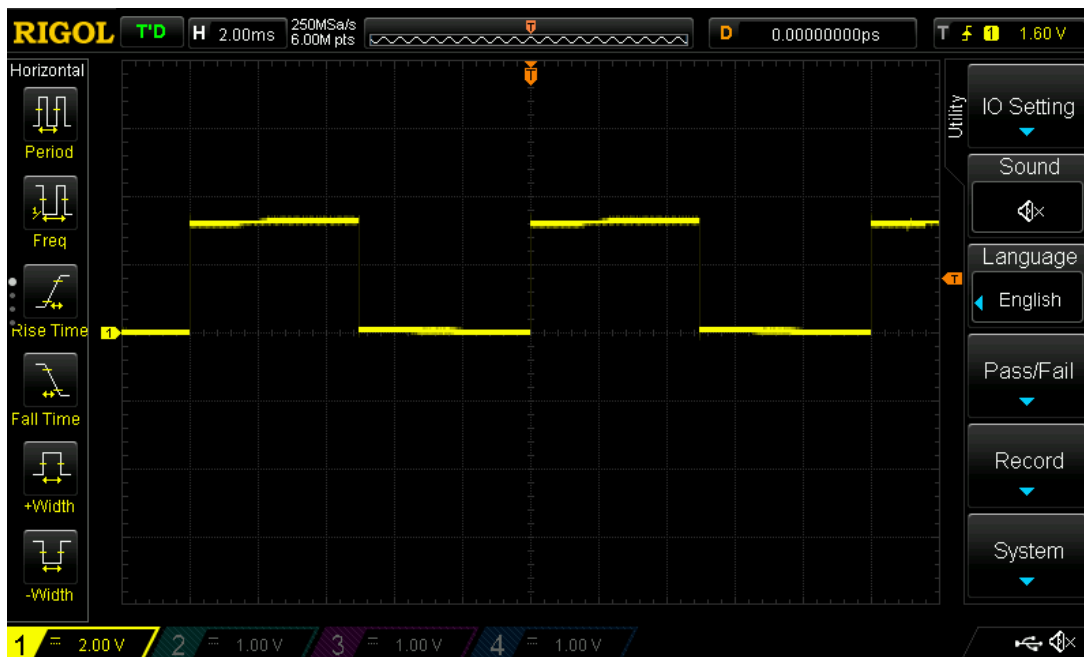
Slika 5: Shema upravljača brzinom

#### 3.3.4.1. PWM signali

Kao što je već navedeno, brzinom ovog motora se upravlja putem PWM signala. PWM signali funkcioniraju na principu promjene vremena trajanja pozitivnog napona vala kroz periodu. Promjenom postotka ispunjenosti periode mijenja se prosječni napon kroz trošilo. Pozitivni napon konstantne je veličine, dok ostatak vala nosi 0 V [15].

Na primjer, ako je radni ciklus (postotak ispunjenosti vala pozitivnom naponskom razinom, engl. *duty cycle*) 50%, kroz vrijeme će napon kroz trošilo u prosjeku biti 50% pozitivne naponske razine. Ako je radni ciklus 25%, kroz vrijeme će napon kroz trošilo u prosjeku biti 25% pozitivne naponske razine.

Na slici 6 prikazani su PWM signali iz korištenog mikrokontrolera. U ovoj konfiguraciji je perioda signala 10 milisekundi i radni ciklus je 50%. Pozitivna naponska razina je 3,3 V te ostatak vala iznosi 0 V.



Slika 6: PWM signali iz korištenog mikrokontrolera, radni ciklus 50%

### 3.3.4.2. MOSFET

Korištenje MOS-tranzistora omogućuje upravljanje krugovima velikih snaga s pomoću krugova nižih naponskih razina [16]. U ovom se sustavu s pomoću N-kanalnog MOSFET-a upravlja brzinom motora. N-kanalni MOSFET propušta struju između područja S (engl. *Source*) i D (engl. *Drain*) kada je napon između S i G (engl. *Gate*) područja iznad specificiranog. Taj  $U_{GS}$  specifičan je za pojedini tranzistor. MOSFET-i tzv. logičke razine (engl. *logic level*) propuštaju struju sve od niskih  $2 U_{GS}$ . U ovom je sustavu iskorišten N-kanalni MOSFET obogaćenog tipa modela IRL540NPbF proizvođača International Rectifier. Otpor između područja D i S mu je  $0,053 \Omega$  na niskih  $5 V U_{GS}$  [17].

Kada je MOSFET korišten na  $U_{GS}$  nižem od specificiranog on u krugu predstavlja otpornik [18]. U ovom se sustavu ne koristi u tu svrhu, nego u svrhu potpunog puštanja ili potpunog zaustavljanja struje kroz motor.

### 3.3.4.3. Pozicija MOSFET-a u krugu

N-kanalni MOSFET-i se u krugu pozicioniraju tako da im je S područje priključeno na 0 V, G na upravljač MOSFET-a i D na električni krug čijim se protokom struje upravlja. Takav se način spajanja naziva *low side*. Kada bi na D područje priključili izvor napona i na S područje električni krug tada bi se to nazivalo *high side* [19]. U takvoj bi konfiguraciji nakon propuštanja struje kroz MOSFET  $U_{SG}$  iznosio nešto manje od napona iz izvora.  $U_{GS}$  bi iznosio  $-1 \times U_{SG}$ . Takvo se stanje može spriječiti jedino s posebnim elektroničkim konstrukcijama. Bez njih bi takvo umrežavanje uništilo MOS-tranzistor i onemogućilo njegov rad.

#### 3.3.4.4. *Pull-down* otpornik

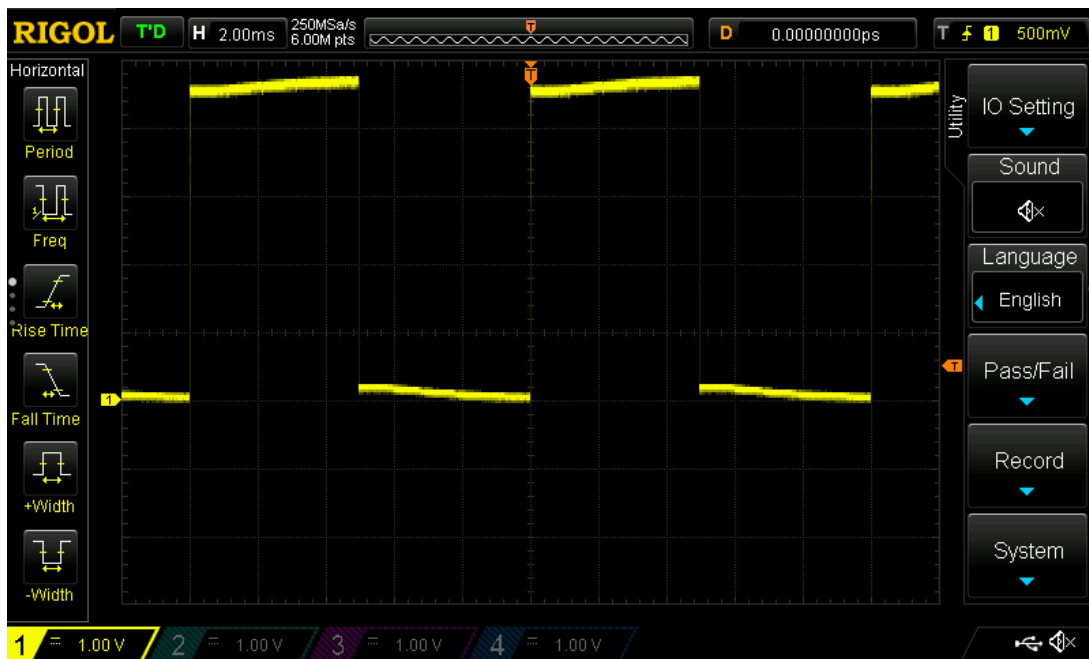
Kako bi osigurali da  $U_{GS}$  iznosi 0 V kada se do G područja ne pušta napon dodan je tzv. *pull-down* otpornik. Na taj se način osigurava nepropuštanje struje kroz motor kada je naponska razina PWM signala 0 V [8, str. 200]. Vrijednost otpornika odabrana je kako ne bi smetala funkcioniranju sustava kada je naponska razina PWM signala pozitivna.

#### 3.3.4.5. Podizanje naponske razine PWM signala

Raspon naponske razine mikrokontrolera korištenog u ovom radu seže do 3.3 V. To nije razina za koju je spomenuti MOSFET određen te je stoga potrebno podići napon signala poslanih od mikrokontrolera do MOSFET-a.

Veliki izazov u podizanju napona tih signala je čistoća signala. Trebao bi biti nepromjenjiv kada se digne na određenu razinu. Također, vrijeme prijelaska sa stanja bez napona na stanje maksimalnog napona bi trebalo biti minimalno. Razlog tome je što su N-kanalni MOSFET-i dizajnirani za korištenje na specficiranoj  $U_{GS}$  naponskoj razini. Ako se ne koriste na toj razini  $U_{GS}$  bi trebao iznositi 0 V. Korištenje MOSFET-a na nekoj drugoj  $U_{GS}$  razini rezultira pretjeranim zagrijavanjem. Sa zagrijavanjem MOSFET-a dolazi i povećanje njegove otpornosti. Povećanje otpornosti ponovno vodi do povećanja zagrijavanja i kreira se petlja pozitivne povratne sprege [18]. Takvi sustavi rezultiraju u eksponencijalnim promjenama stanja. U ovom kontekstu eksponencijalno se povećavaju toplina i otpornost MOS-tranzistora.

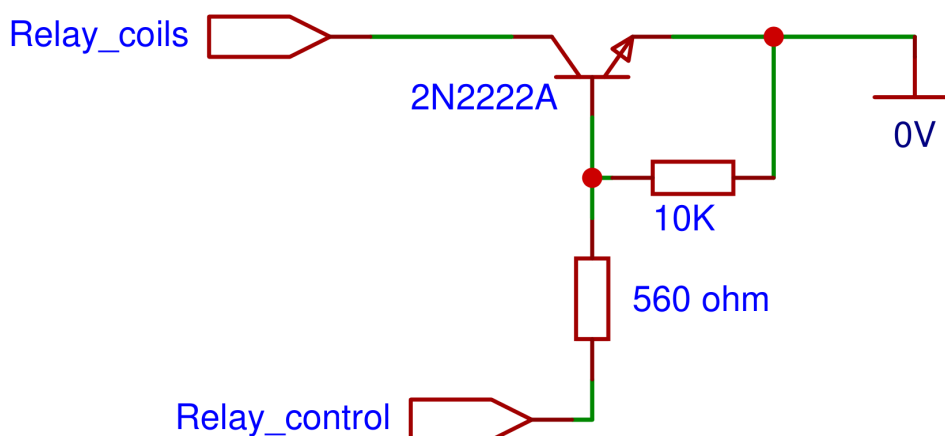
Kako bi signal bio što čišći iskorišten je logički pretvarač (engl. *logic converter*) koji ima sposobnost pretvaranja logičkih signala nižih naponskih razina u logičke signale viših naponskih razina. Korišten je logički pretvarač nalik onom proizvedenom od strane SparkFun-a. On rješava oba problema kod podizanja logičkih razina za MOSFET-e: vrijeme podizanja i spuštanja signala od 0 V do maksimalnog napona je minimalno i napon ostaje konstantan nakon podizanja. Kod korištenog pretvarača potrebno je specificirati napon niže i više razine tako da se na *LV* i *HV* ulaze puste željeni naponi. Tada se na *LV1* ulaz puštaju signali koji se podižu na *HV* razinu na izlazu *HV1*.



Slika 7: PWM signali iz logičkog pretvarača, radni ciklus 50%

Kao što je vidljivo na slici 7 signal iz logičkog pretvarača minimalno oscilira na pozitivnim naponskim razinama i na 0 V. Također, pretvorba signala s 3,3 V iz mikrokontrolera na 5 V je gotovo trenutna. Stvarni napon iz logičkog pretvarača je 4,5 V. Razlog tome je već spomenuti pad napona iz 5-voltnog napajanja.

### 3.3.5. Upravljač smjerom



Slika 8: Shema upravljača smjerom

### 3.3.5.1. Bipolarni tranzistor

Bipolarni tranzistor je tro-terminalni elektronički uređaj. Ta tri terminala se nazivaju baza, kolektor i emiter. Strujom kroz bazu se upravlja puno većom strujom između kolektora i emitera. Dolazi u 2 varijante: NPN i PNP [20]. U ovom će radu biti opisan rad NPN tipa. To je tip tranzistora koji je korišten u sklopu upravljača smjerom.

### 3.3.5.2. Princip rada NPN tranzistora

Kod NPN tranzistora struja može teći od baze do emitera i ne može od baze do kolektora. Struja do baze i do kolektora struja do emitera.  $I_C$  (struja kroz kolektor) iznosi  $\beta \times I_B$ . Beta se naziva transportnim faktorom baze. Ta vrijednost može biti od 50 do 250, ovisno o vanjskim faktorima. Navedena jednadžba predstavlja korist tranzistora: strujom kroz bazu se upravlja puno većom strujom kroz kolektor [8, str. 72].

Tranzistori se mogu koristiti kao prekidači. Korist u tom pristupu je jednostavnost i funkcionalnost implementacije. U upravljaču smjerom se koristi kao prekidač struje kroz zavojnice releja. Kada se kroz bazu pustiti struja struja teče kroz zavojnice, otvarajući releje. Kada kroz bazu ne teče struja struja ne teče niti kroz zavojnice, zatvarajući releje.

### 3.3.5.3. Struja kroz tranzistor

U ovom radu je korišten tranzistor 2N2222A proizvođača Diotec Semiconductor. To je bipolarni tranzistor NPN tipa. Kada kroz bazu ne teče struja on ne pušta struju od kolektora do emitera. Kada se želi pustiti struja kroz tranzistor na bazu se pušta napon od 3,3 V preko otpornika vrijednosti 560  $\Omega$ .

Maksimalan dopušten napon između emitera i baze  $U_{EB}$  je 6 V. Maksimalna dopuštena struja do kolektora tog tranzistora je 600 mA [21]. Uobičajeni pad napona između baze i emitera kod silicijskih tranzistora je oko 0,7 V [22]. Pad napona kroz otpornik je oko  $3,3V - 0,7V = 2,6V$ . Struja kroz otpornik je stoga  $2,6V \div 560\Omega = 4,6mA$ .

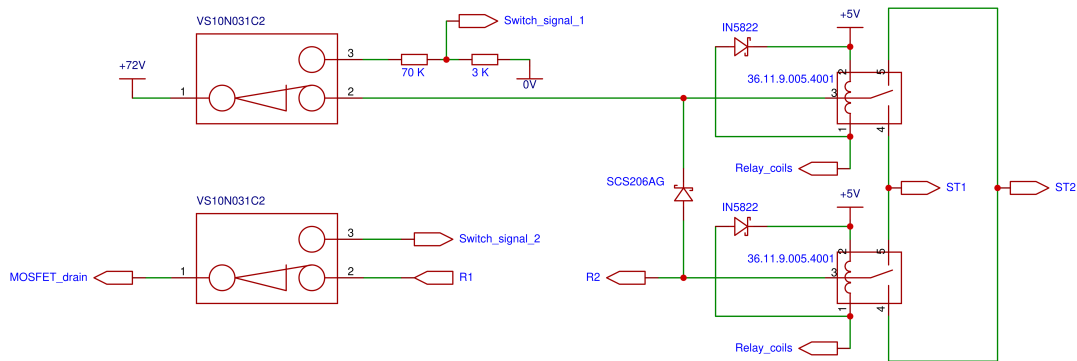
Na kolektor su u paralelnom spoju spojene dvije zavojnice. Obje zavojnice su spojene na izvor napona od 5 V na jednom terminalu i na kolektor na drugom. Pretpostavit ćemo da će u stvarnom slučaju izvor napona iznositi 4,5 V zbog pada napona iz izvora. Prema njegovom tehničkom uputstvu, kako bi se relej aktivirao potrebno je primijeniti 0,36 W snage [23]. Samim time, struja kroz jednu zavojnicu treba biti minimalno  $P = V \times I \rightarrow 0,36W \div 4,5V = 0,08A$  kako bi se 1 relej aktivirao. Struja kroz kolektor bi tada iznosila 0,16 A, odnosno 160 miliampera. Ta vrijednost je daleko niža od maksimalne dopuštene.

Kako struja kroz bazu tranzistora ne bi ograničavala struju kroz zavojnice transportni faktor baze bi trebao iznositi  $160mA \div 4,6mA = 34,8$ . Tranzistor 2N2222A ima transportni faktor baze  $\beta$  od minimalno 100 kod  $I_C = 150mA$  i  $U_{CE} = 10V$ . Karakteristike za ovu specifičnu situaciju nisu dane, no testiranje je pokazalo da ovakva postava omogućuje aktivaciju releja.

### 3.3.5.4. Pull-down otpornik

Kao i kod MOSFET-a i u ovoj konfiguraciji je iskorišten *pull-down* otpornik kako bi se osigurala pravilna naponska razina na bazi kada ne postoji signal od mikrokontrolera.

### 3.3.6. Releji i prekidači



Slika 9: Shema releja i prekidača

#### 3.3.6.1. Releji

Releji korišteni u ovom radu su tipa SPDT (akr. od engl. *single pole double throw*). SPDT prekidači su prekidači koji sadrže 1 ulaz i 2 izlaza. Ulaz se kod takvih prekidača naziva COM (akr. od engl. *common*). U jednom trenutku je ulaz priključen na jedan od izlaza. Izlaz na kojeg je relej priključen u neaktiviranom stanju se naziva NC (akr. od engl. *normally closed*). Aktivacijom releja COM se odvađa od NC izlaza i spaja s drugim izlazom. Izlaz s kojim je tada povezan se naziva NO (akr. od engl. *normally open*). Aktivacija releja vrši se energiziranjem njegove zavojnice specificiranom snagom.

Na elektroničkoj shemi vidljivi su ulazi i izlazi iz releja. Ulazi 3 predstavljaju COM, izlazi 5 NC i izlazi 4 NO. Kako bi se relej aktivirao potrebno je pustiti struju kroz njegove zavojnice. To se vrši kroz ulaz 2. Struja teče iz ulaza 2 kroz zavojnicu i izlazi na izlaz 1.

Korist releja u ovom sustavu je promjena polariteta zavojnice statora korištenog motora. Kao što će kasnije biti opisano, tako se mijenja smjer rotacije osovine motora.

Relej korišten u ovom sustavu je oznake 36.11.9.005.4001 proizvođača Finder. Kao što njegove tehničke upute navode, zavojnica ovog releja namijenjena je za energiziranje na 5 V [23]. Zavojnice su u paralelnom spoju kako bi se osigurao dovod 5-voltnog napona. Napon bi se podijelio po zavojnicama kada bi one bile serijski povezane i tada ne bi bio dovoljno visok da aktivira relej.



### 3.3.6.2. Poredna dioda

Poredna dioda je "dioda koja je spojena paralelno induktivnom trošilu. Osigurava put el. struji trošila kada se ona ne može zatvoriti nekom drugom komponentom u elektroničkom učinkom pretvaraču." [24] Njezina potreba se nalazi u činjenici da, kad se putem prekidača prekine tok struje kroz induktivno trošilo, nastane visok skok negativnog napona između terminala induktivnog trošila [25]. To stanje opisuje Lenzov zakon. On navodi kako je "smjer induciranog napona uvijek takav da se od toga napona stvorena struja svojim magnetskim učinkom protivi promjeni magnetskog toka  $d\Phi$  zbog kojega je došlo do induciranja napona" [7, str. 243]. Odnosno, induktivno trošilo ruši svoje elektromagnetsko polje stvarajući vrlo visok napon kako bi održao količinu struje koja teče njime konstantnom. Taj napon je obrnutog polariteta od polariteta induktivnog trošila u električnom krugu [25].

Rezultat induktivnog napona je stvaranje velikih pozitivnih napona na terminalu *low side* povezanog prekidača. U slučaju releja taj prekidač je bipolarni NPN tranzistor i terminal na kojem se stvara velika količina napona je kolektor. U slučaju motora taj prekidač je N-kanalni MOSFET i terminal D područja.

Kako bi se izbjeglo stvaranje štete kod tih električnih komponenti u mrežu se ugrađuju poredne diode. Njihova polarnost je obrnuta polarnosti naponskog izvora. Pri toku struje kroz induktivno trošilo dioda ne provodi struju. U trenutku prekida tok struje od strane prekidača dioda otvara put za tok struje induciranog napona. Na taj način inducirani napon neće rasti do razmjera opasnih za elektroničke prekidače ili druge elektroničke uređaje povezane s njegovim izvorom [25].

U svrhe porednih dioda se koriste diode tipa Schottkey. Pad napona kroz njih je oko 0,2 V, što je izrazito malo. Također, brzo prekidaju tok struje pri ponovnom energiziranju induktivnog trošila [26].

U ovom su radu su za relej korištene diode modela IN5822 proizvođača First Components International. Za motor je korištena dioda modela SCS206AG proizvođača Rohm Semiconductor. Obje diode su tipa Schottkey.

### 3.3.6.3. Prekidači

Prikazani prekidači su mehanički. Njihova aktivacija zahtjeva pritisak na polugu prekidača. Oni su SPDT tipa. Ulazi 1 predstavljaju COM, izlazi 2 NC i izlazi 3 NO. Modela su proizvođača Highly Electric. Njihova korist je ispunjenje zahtjeva sustava broj 6: mogućnost registriranja krajnje pozicije ograde. Kada pomična ograda dostigne svoju krajnju poziciju prekidač se aktivira i šalje signal mikrokontroleru. Pomične ograde imaju dvije krajnje pozicije: otvorenu i zatvorenu. Iz tog su razloga implementirana 2 prekidača.

### 3.3.6.4. Naponsko djelilo

Naponsko djelilo (engl. *voltage divider*) je jednostavni sklop koji pretvara visoki napon u niži [27]. Naponsko djelilo predstavlja jednostavan i učinkovit način smanjenja napona. Cijelo

djelilo se sastoji od dva serijski spojena otpornika. Formula za izračun napona kroz drugi otpornik  $R_2$  glasi:

$$U_{out} = U_{in} \times \frac{R_2}{R_1 + R_2}$$

Korišteni mikrokontroler može kroz svoje digitalne ulaze detektirati od -0,3 do 0,83 V za nizak napon i od 2,48 do 3,6 V za visok napon [28, str. 12]. To znači da bi signal trebao biti između 2,5 i 3,6 V. Srednji broj između te dvije vrijednosti je 3,05 V te je to i ciljani napon preko drugog otpornika u naponskom djelilu. Za struju kroz djelilo uzmimo proizvoljnu vrijednost od 1 mA. Sada možemo izračunati vrijednosti  $R_1$  i  $R_2$ :

$$R = U \div I \rightarrow R_{uk} = \frac{72V}{1 \times 10^{-3}A} = 72000\Omega = \mathbf{72K\Omega}$$

$$3,05V = 72V \times \frac{R_2}{72K\Omega} \rightarrow R_2 = \frac{3,05V \times 72K\Omega}{72V} = \mathbf{3050\Omega}$$

$$3050\Omega + R_1 = 72K\Omega \rightarrow R_1 = \mathbf{68920\Omega}$$

U svrhu zaokruživanja bit će uzete vrijednosti 70 K $\Omega$  i 3 K $\Omega$ . Tada će struja kroz naponsko djelilo iznositi:

$$I = U \div R \rightarrow I = \frac{72V}{73K\Omega} = \mathbf{0,99mA}$$

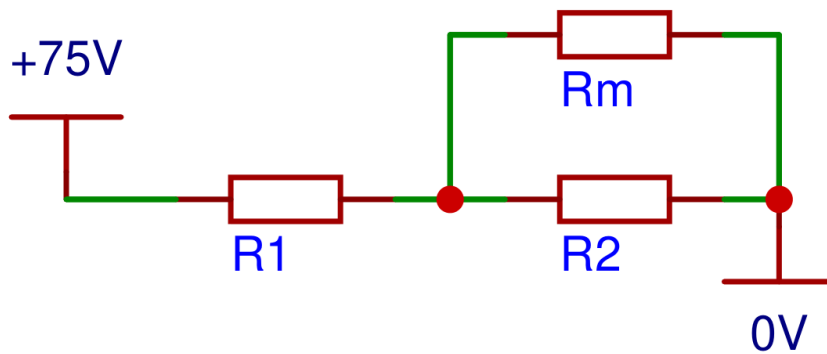
A napon kroz  $R_2$ :

$$U_{out} = 72V \times \frac{3K\Omega}{73K\Omega} = \mathbf{2,95V}$$

Dodatno, u krug je pridružen mikrokontroler putem ulaza I36. Struja koja teče kroz ulaz mikrokontrolera pri primijenjenih 3,3 V je najviše 50 nA [28, str. 12]. Samim time možemo zaključiti da je otpor kroz ulaz mikrokontrolera minimalno:

$$R = U \div I \rightarrow R_m = \frac{3,3V}{50 \times 10^{-9}A} \rightarrow R_m = \mathbf{66000000\Omega = 66M\Omega}$$

Sada shemu naponskog djelila možemo koncipirati kao onu prikazanu na slici 10.



Slika 10: Shema naponskog djelila

Promjena napona kao posljedica prisutnosti mikrokontrolera je minimalna. Kada je vrijednost  $R_m$  više od 10 puta veća od vrijednosti  $R_2$  napon kroz  $R_m$  je 4% manji nego kad  $R_m$  ne bi bio prisutan u krugu [29]. Promjena otpora kroz  $R_2$  je, koristeći formulu za ukupan otpor paralelno spojenih otpornika, najviše:

$$R_2 || R_m = \left( \frac{1}{R_2} + \frac{1}{R_m} \right)^{-1} = \frac{R_2 \times R_m}{R_2 + R_m} = \frac{3K\Omega \times 66M\Omega}{3K\Omega + 66M\Omega} = 2999,86\Omega$$

$$\frac{3000 - 2999,86}{3000} = 0,00467\%$$

Napon kroz  $R_2$ , i samim time  $R_m$ , je tada promijenjen za najviše:

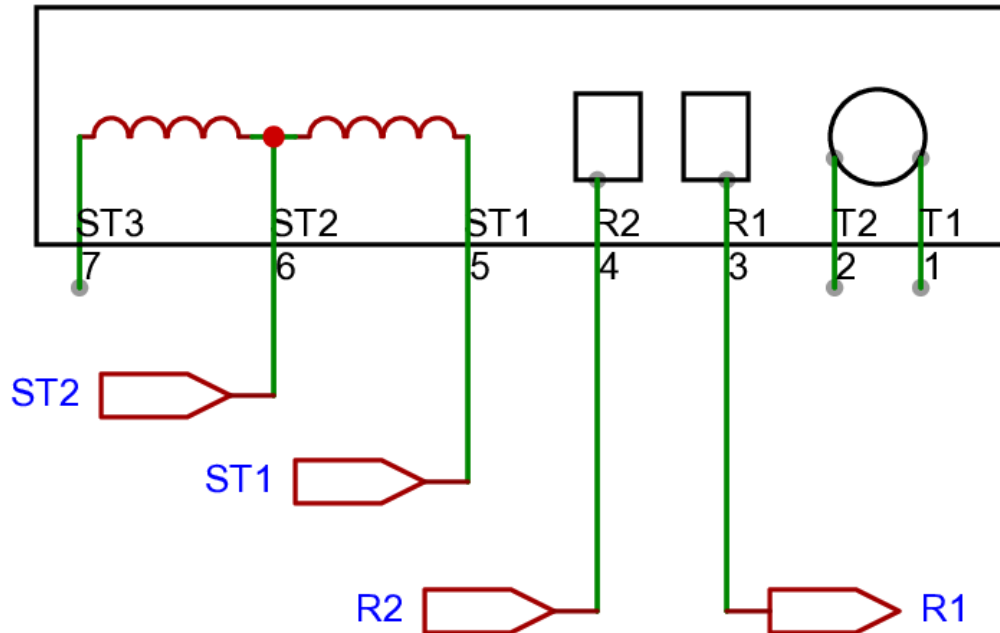
$$U_{out} = 72V \times \frac{2999,86\Omega}{72999,86\Omega} = 2,9589V$$

$$\frac{2,9589 - 2,95}{2,9589} = 0,296\%$$

Zaključno, prisutnost mikrokontrolera u naponskom djelilu nema relevantan utjecan na rezultatni napon.

### 3.3.7. Motor

C.E.SET. MCA 61/64 148/WHE0



Slika 11: Shema motora

#### 3.3.7.1. Tip motora

Odabrani motor mora moći biti pokretan istosmjernom strujom kako bi kontrola s pomoću PWM signala bila moguća. Zbog dostupnosti je odabran univerzalni motor. Univerzalni motori mogu biti pokretani i izmjeničnom i istosmjernom strujom. Oni imaju snažan početni okretni moment što ih čini posebno korisnim u kontekstu pokretanja ograde [30].

Korišteni motor sadrži 7 priključaka: 3 za stator, 2 za rotor i 2 za tahogenerator. Priključci statora označeno su sa ST, rotora R i tahometra T. Kako bi se motor pokrenuo stator i rotor se spajaju u seriju. Onda se kroz motor pušta napon. Kako bi se obrnuo smjer rotacije motora potrebno je promijeniti polaritet statora s obzirom na rotor [31, str. 877].

#### 3.3.7.2. Snaga motora

Potrebnu jačinu motora uvjetuje težina pomične ograde [32]. Optimizirajući za učinkovitost, snaga motora za pokretanje pomične klizne ograde korištene u stambene svrhe bi trebala biti od 400 do 500 W [33]. Na tržištu automatskih pokretača kliznih ograda za stambene svrhe snage motora iznose oko 280 W. U stanju su pokretati ograde do 635 kg teške i 12 metara široke [34] [35]. Automatski pokretači za ograde težine do 1 tone i širine do 15,5 m sadrže motore snage do 750 W [32].

### 3.3.7.3. Specifikacije korištenog motora

Motor korišten u ovom radu je modela C.E.SET. MCA 61/64 148/WHE0. Kao što je navedeno na motoru, njegova snaga je 380 W i namijenjen je korištenju na 230 V napona. Pri tim uvjetima jačina električne struje kroz motor je 1,75 A. Kao što je već navedeno, ovaj motor se pokreće 72-voltnim naponom istosmjerne struje.

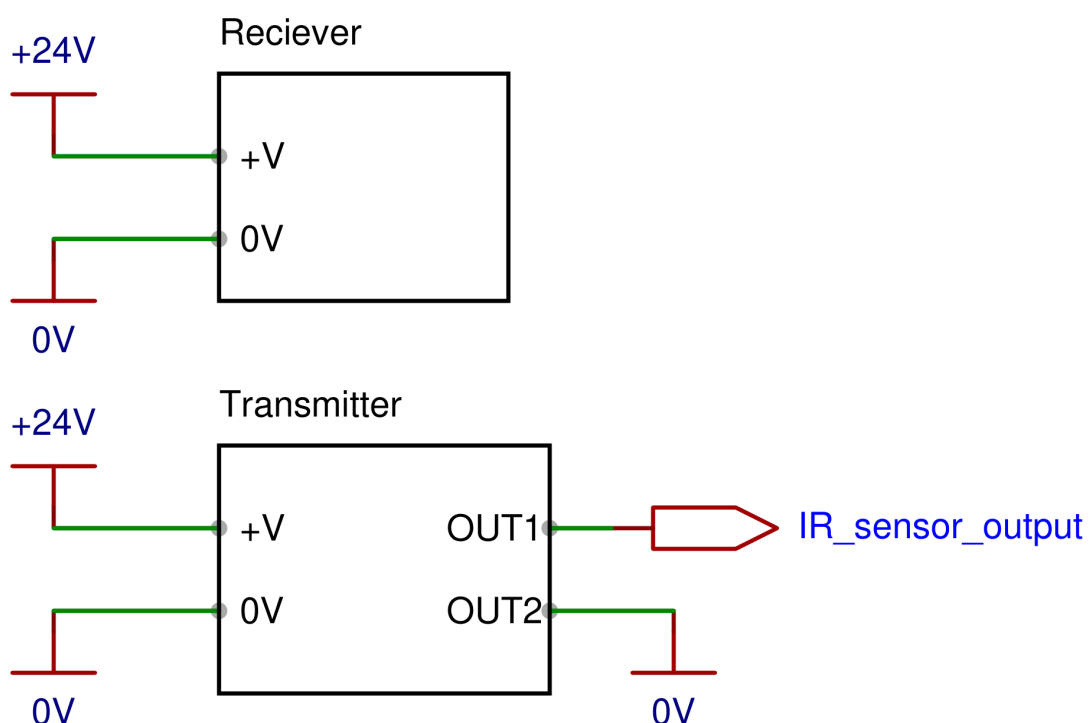
### 3.3.7.4. Promjena smjera rotacije

Kao što je već navedeno, promjena smjera rotacije univerzalnog motora se vrši promjenom polariteta statora s obzirom na rotor. U uobičajenom radu na ulaz ST2 se pušta 72 V napona istosmjerne struje. Ona teče kroz zavojnicu statora, izlazi kroz ST1, ulazi kroz R2 i izlazi kroz R1.

Aktivacija releja obrće polaritet statora. Na ulaz ST1 se pušta 72 V napona istosmjerne struje. Ona teče kroz zavojnicu statora, izlazi kroz ST2, ulazi kroz R2 i izlazi kroz R1.

Prije promjene polariteta je potrebno zaustaviti motor kako se on ne bi ošteti.

### 3.3.8. Odašiljač i prijemnik infracrvenih zraka



Slika 12: Shema odašiljača i prijmnika infracrvenih zraka

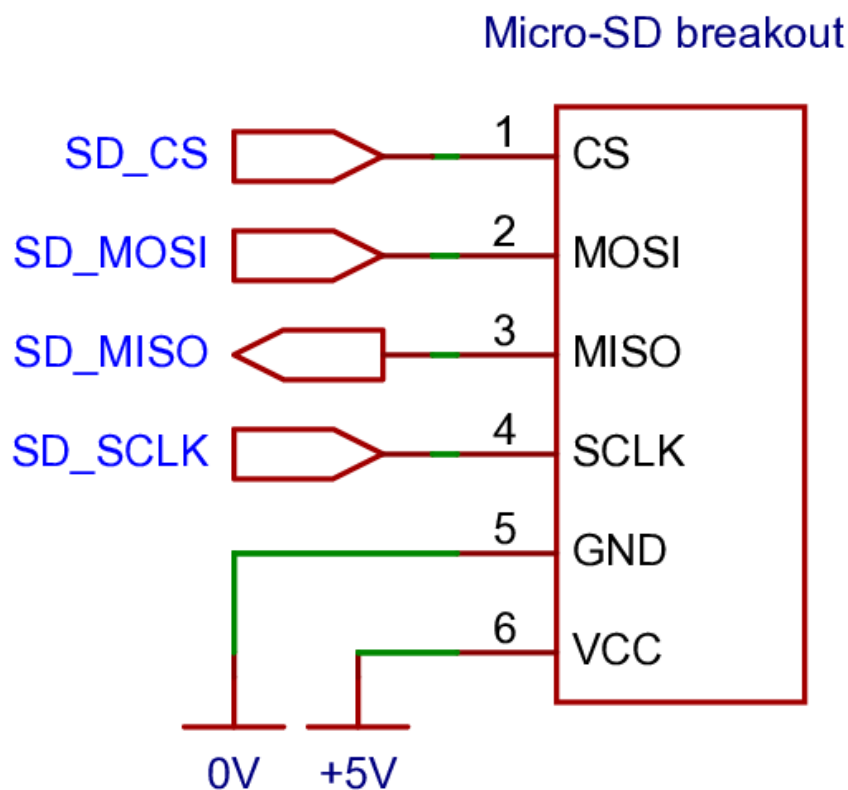
Kako bi zahtjev 7 bio ispunjen u sustav je ugrađen odašiljač i prijemnik infracrvenih zraka. Pri prekidu infracrvene zrake mikrokontroleru se šalje signal. Struja kroz motor se

zaustavlja pri primitku tog signala.

Model korištenog para odašiljača i prijelnika je Sensiva XS proizvođača V2. Kao što je navedeno na Internetnoj stranici proizvođača, danje svjetlo ne ometa njegov rad [36]. Priključen je na napajanje od 24 V istosmjerne struje.

Korišteni model ima 2 načina rada: izlazi OUT1 i OUT2 su pri prisutnom signalu povezani ili su nepovezani. Spoj na izlazu J1 određuje način rada [37]. U ovom je radu konfiguriran kako bi pri prisutnom signalu ulazi bili nepovezani. Kada se signal prekine ulazi se povezuju i šalje se signal mikrokontroleru.

### 3.3.9. SD kartica



Slika 13: Shema pločice za spajanje SD kartice

Pločica za spajanje SD kartice korištena u ovom radu je modela microSD breakout proizvođača Soldered Electronics. Kartica se koristi za dugotrajno pohranjivanje podataka prikupljenih od strane mikrokontrolera. Na njoj se nalazi datoteka ekstenzije .txt u koju mikrokontroler upisuje vremena pokretanja ograde. Na zahtjev mobilnog uređaja mikrokontroler čita zadnjih 10 unosa. Komunikacija između mikrokontrolera i SD kartice se vrši putem SPI protokola.

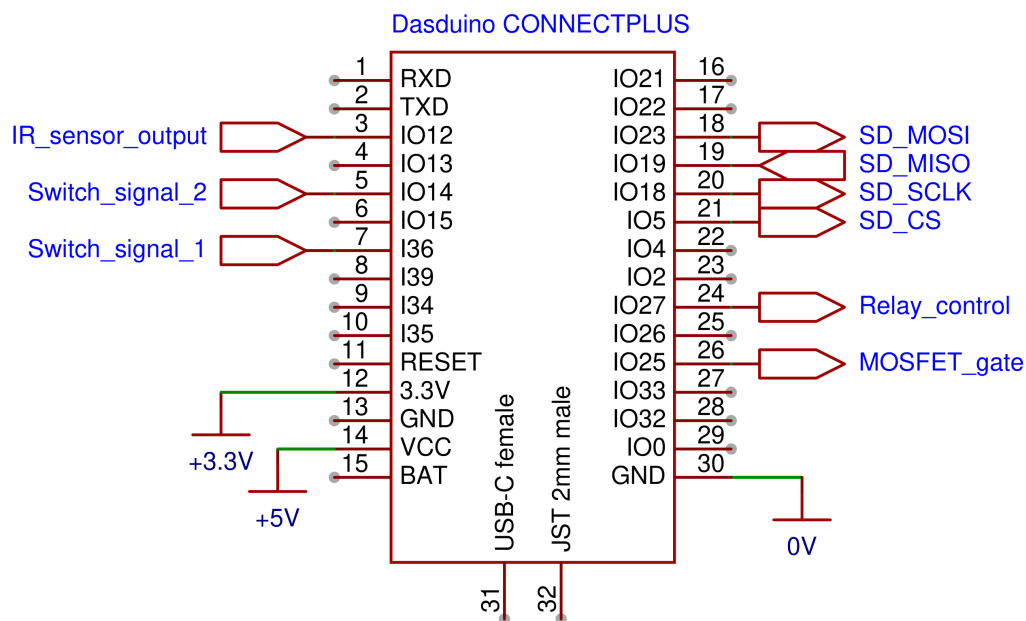
### 3.3.9.1. SPI protokol

SPI protokol omogućava komunikaciju između *master* i *slave* uređaja. *Master* uređaj upravlja komunikacijom dok *slave* odgovara na naredbe. Postoje 4 poveznice između uređaja u SPI protokolu:

- *Master-In Slave-Out* (akr. MISO) kojim *master* dobiva podatke
- *Master-Out Slave-In* (akr. MOSI) kojim *slave* dobiva naredbe
- *Clock* (akr. SCLK) kojeg generira *master*, njime se određuju taktovi prijenosa podataka
- *Chip Select* (akr. CS) kojim *master* određuje je li *slave* aktivan

[38, str. 172]

### 3.3.10. Mikrokontroler



Slika 14: Shema mikrokontrolera

Korištena razvojna pločica je Dasduino CONNECTPLUS proizvođača Soldered Electronics. Mikrokontroler koji je u njoj sadržan je modela ESP32-WROVER-E proizvođača Espressif. Mikrokontroler upravlja radom sustava, nadzire njegovo stanje i komunicira s mobilnim uređajem.

ESP32-WROVER-E modul sadrži 4 MB integrirane flash memorije i 8 MB integriranog PSRAM-a. Također sadrži ugrađenu antenu i podršku za komunikaciju putem WiFi-a, Bluetootha i BLE-a. Mikrokontroler sadrži 2 procesorske jezgre kojima se može individualno upravljati [28, str. 1]. Korišteni operacijski sustav je freeRTOS [28, str. 2].

Razvojna pločica omogućava komunikaciju s mikrokontrolerom putem USB-C priključka, ugrađuje naponski regulator, ugrađuje svjetleću diodu i slično [39]. Na VCC ulaz razvojne pločice priključen je naponski izvor od 5 V. Ugrađeni naponski regulator modela TPS7A2633DRVR proizvođača Texas Instruments prilagođava uvedeni napon na 3,3 V, za kojeg je ESP32 i namijenjen [40]. Ulazni napon kod regulatora može sezati do 18 V te samim time 5 V neće oštetiti uređaj [41].

## 3.4. Modeliranje računalnog programa mikrokontrolera

### 3.4.1. Razvojna okolina

Program je razvijen s pomoću programa Visual Studio Code i ekstenzije PlatformIO. Razvijen je u programskom jeziku C++. Kao što se navodi na njihovoj službenoj Internetskoj stranici, PlatformIO je "višeploatformski, višearhitekturni, višestruko okvirni, profesionalni alat za inženjere ugrađenih sustava i programere koji razvijaju aplikacije za ugrađene proizvode." [42] Projekt se konfigurira s pomoću platformio.ini datoteke. Putem nje se odabire vrsta mikrokontrolera, platforma za koju je on razvijen, struktura programa, uključene vanjske biblioteke i slično.

### 3.4.2. Struktura programa

Program se sastoji od 5 datoteka:

- platformio.ini
- main.cpp
- BLEController.h
- SDCommunicator.h
- ElectronicsControl.h
- WS2812BControl.h

i popratnih biblioteka. Korištene biblioteke su dio Arduino core integracije za ESP32. Osim toga, korištena je *FastLED* biblioteka autora Daniela Garcie u verziji 3.6.0.

### 3.4.3. Datoteka platformio.ini

Kao što je već navedeno ova datoteka služi konfiguraciji projekta. Sadržaj ove datoteke ovog programa je:

```
[env:esp-wrover-kit]
platform = espressif32
board = esp-wrover-kit
```



```

framework = arduino

lib_deps = fastled/FastLED@^3.6.0

monitor_speed = 115200
monitor_filters = esp32_exception_decoder

```

Ovdje je definirano korištenje *FastLED* biblioteke. Korištena je u datoteci `WS2812BControl.h`. Korist zastavice `esp32_exception_decoder` je pravilno dekodiranje poruka iznimke ESP32 mikroprocesora. Zastavica `monitor_speed` određuje brzinu prijenosa podataka kod komunikacije putem USB priključka.

### 3.4.4. Datoteka main.cpp

Datoteka `main.cpp` sadrži funkcije `setup()` i `loop()`. Kao što je već navedeno, ESP32 je kompatibilan s Arduino core funkcionalnostima [43]. Ta integracija omogućuje korištenje ovakve strukture. Funkcija `setup()` se izvršava nakon resetiranja pločice. Funkcija `loop()` se izvršava neprestano za vrijeme njezina rada.

`setup()` funkcija inicijalizira i postavlja BLE server, inicijalizira čitač SD kartice, vraća postavke elektronike na njihove zadane i povezuje funkcije prekida s relevantnim ulazima.

Osim za otklanjanje grešaka `loop()` funkcija u ovom programu nema korist. Sve su funkcionalnosti osposobljene kao reakcije na promjene stanja ulaza u mikrokontroler ili poruka putem BLE komunikacije.

#### 3.4.4.1. Prekidi

U Arduino core funkcionalnosti spadaju i funkcije za upravljanje prekidima. Korist prekida je trenutno reagiranje na promjenu stanja ulaza mikrokontrolera. U trenu specificirane promjene specificiranog ulaza se poziva specificirana funkcija. Poziv funkcije `attachInterrupt( uint8_t pin, std::function< void( void ) > intRoutine, int mode )` osposobljava prekid. `pin` je ulaz koji se nadgleda, `function` funkcija koja se poziva i `mode` promjena koja uzrokuje prekid [44]. U ovom su programu definirana 3 prekida:

```

pinMode( irInterruptPIN, INPUT_PULLUP );
attachInterrupt( digitalPinToInterrupt( irInterruptPIN ), irInterrupt, CHANGE );

pinMode( switch1InterruptPIN, INPUT_PULLDOWN );
attachInterrupt( digitalPinToInterrupt( switch1InterruptPIN ), switch1Interrupt,
CHANGE );

pinMode( switch2InterruptPIN, INPUT_PULLUP );
attachInterrupt( digitalPinToInterrupt( switch2InterruptPIN ), switch2Interrupt,
CHANGE );

```

Svaki od njih se okine na promjenu digitalnog stanja njihovih ulaza. Funkcije `pinMode( uint8_t pin, uint8_t mode )` konfiguriraju ulaze i izlaze u mikrokontroler. Ovdje se povezuju na 3,3 V preko 45 K $\Omega$  otpornika za `INPUT_PULLUP` i na 0 V preko 45 K $\Omega$  otpornika za

`INPUT_PULLDOWN` [28, str. 13]. Korist tih otpornika je uspostavljanje naponske razine ulaza i simultano izbjegavanje ometanja njihovog rada [45]. Strukture funkcija pozvanih u trenutku prekida su opisane u sklopu datoteke `ElectronicsControl.h`.

### 3.4.5. Datoteka `BLEController.h`

Datoteka `BLEController.h` sadrži varijable vezane uz komunikacijski protokol BLE, klase i metode koje osposobljavaju tu komunikaciju te funkciju za inicijalizaciju BLE servera.

#### 3.4.5.1. Komunikacijski protokol Bluetooth Low Energy

Bluetooth Low Energy (akr. BLE) je bežični komunikacijski protokol dizajniran za slučajeve upotrebe koje zahtijevaju trošenje vrlo malih količina energije. Podržava *point-to-point* komunikaciju kao i *mesh* mreže. BLE također pruža uslugu lociranja [46].

U komunikaciji između ESP32 mikroprocesora i mobilnog Android uređaja bivši služi svrhu servera, a potonji klijenta. Ta se vrsta komunikacije naziva *point-to-point*. Klijent se povezuje na server. U tzv. GATT (akr. od engl. *Generic Attribute*) tablici servera su sadržani profili, servisi i karakteristike. Server i klijent komuniciraju koristeći karakteristike GATT tablice [47]. Svaki servis i karakteristika imaju univerzalno jedinstven identifikator (akr. UUID od engl. *universally unique identifier*). Klijentska aplikacija pomoću tih identifikatora određuje karakteristiku koju želi koristiti.

Karakteristike posjeduju vrijednosti, opise i svojstva. Svojstva karakteristika ESP32 BLE servera mogu biti čitaj (engl. *read*), piši (engl. *write*), obavijesti (engl. *notify*), emitiraj (engl. *broadcast*), naznači (engl. *indicate*) i piši bez odgovora (engl. *write no response*) [48]. Svojstvo piši se koristi za prijenos podataka s klijenta na server. Svojstvo čitaj se koristi za prijenos podataka sa servera na klijent. Vrijednosti karakteristika su bitovni nizovi koji se izmjenjuju.

#### 3.4.5.2. Postavljanje BLE servera

Za rukovanje BLE protokolom pomoću ESP32 mikroprocesora koristi se ESP32 BLE biblioteka [49].

Postavljanje BLE servera vrši se određivanjem njegovog imena, funkcija povratnih poziva (engl. *callback*) servera, postavljanja karakteristika, određivanja funkcija povratnih poziva za karakteristike, pokretanja servisa i pokretanja servera u načinu rada oglašavanje (engl. *advertise*).

Pokazivači na klase s funkcijama povratnih poziva servera se definiraju pomoću metode `void BLEServer::setCallbacks( BLEServerCallbacks *pCallbacks )`. Ona prima argument pokazivača na objekt klase koja nasljeđuje `BLEServerCallbacks` klasu. Ona je definirana u zaglavnoj datoteci `BLEServer.cpp`. Funkcije povratnih poziva karakteristika se uspostavljaju pomoću metode `void BLECharacteristic::setCallbacks( BLECharacteristicCallbacks* pCallbacks )`. Ona prima argument pokazivača na

objekt klase koja nasljeđuje `BLECharacteristicCallbacks` klasu. Ona je definirana u zaglavnoj datoteci `BLECharacteristic.cpp`.

Kod ovog su servera definirane dvije karakteristike. Jedna ima svojstvo čitaj, a druga piši. Korist svojstva piši je određivanje brzine motora putem Android uređaja. Korist svojstva čitaj je prijenos podataka o korištenju automatizirane ograde na mobilni uređaj.

### 3.4.5.3. Funkcija povratnog poziva karakteristike piši

Metoda `void MyWriteCallback::onWrite( BLECharacteristic* pCharacteristic, esp_ble_gatts_cb_param_t* param )` se poziva u trenutku primanja niza bitova od klijentske aplikacije. Ona služi promjeni PWM signala do motora. Vrijednosti se primaju u intervalu od -100 do 100. One se šalju kao niz bitova tipa podatka `int8_t`. One predstavljaju postotke ispunjenosti PWM vala. Pri primitku vrijednosti od mobilnog uređaja provjerava se nalazi li se ona u definiranom intervalu. Ako se ne nalazi vrijednost se zanemaruje.

Ako se vrijednost nalazi u tom intervalu provjerava se dopuštenje promjene PWM signala. Stanje na ulazima prekida definiraju dopuštanja. To rade putem varijabli tipa `bool`. U slučaju dopuštenja okretanja motora u zadanom smjeru poziva se metoda `void ElectronicsControl::setSpeedNDir( int8_t speedValue )` koja je definirana u datoteci zaglavljiva `ElectronicsControl.h`. Nakon poziva te metode mijenja se intenzitet i boja svjetleće diode ugrađene u Dasduino CONNECTPLUS razvojnu pločicu. Ta promjena se vrši putem objekta klase `WS2812BControl` definirane u istoimenoj datoteci.

### 3.4.5.4. Funkcija povratnog poziva karakteristike čitaj

Metoda `void MyReadCallback::onRead( BLECharacteristic* pCharacteristic )` se poziva u trenutku zahtjeva klijenta za podacima o korištenju automatizirane ograde. Pri primitku zahtjeva se u niz znakovnih nizova unosi zadnjih 10 unosa u SD karticu. Nizovi znakova se ne šalju u istom paketu. Kada god se ova karakteristika poziva putem mobilnog uređaja poziva se 11 puta za redom. Tako se osigurava da se dopuštena veličina BLE paketa ne premaši. Vrijednost za slanje se definira pomoću funkcije `void BLECharacteristic::setValue( std::string value )`.

U ovoj konfiguraciji se u tekstualnu datoteku na SD kartici upisuje vrijeme pokretanja ograde. Vrijeme se upisuje tako da se zapiše vrijednost koju vraća funkcija `millis()`. Kako bi se unosi mogli razlikovati nakon svakog se u datoteku zapisuje znak za novi red. Funkcija `millis()` vraća broj milisekundi od zadnjeg pokretanja mikrokontrolera. Tip vrijednosti koju vraća je `unsigned long` [50]. Ta vrijednost seže do 4,294,967,295 [51]. 4,294,967,295 milisekundi to predstavlja 49,7 dana. Nakon tog vremena se vrijednost koju funkcija `millis()` vraća resetira na 0.

Kako bi opseg mogućeg vremena predstavljanja bio veći potrebno je koristiti modul koji vraća trenutno vrijeme. Princip zapisivanja u i čitanja iz datoteke SD kartice bi bio identičan.

### 3.4.6. Datoteka SDCommunicator.h

Datoteka `SDCommunicator.h` sadrži istoimenu klasu. Njezina korist je pokretanje komunikacije sa SD karticom, upisivanje u i čitanje iz njezinih datoteka. U ovom radu se koristi samo datoteka imena `log.txt`.

Metoda `void SDCommunicator::appendToFile( const char *fileName, String line )` koristi dodavanju novog reda u datoteku specificiranu ulaznim parametrom imena `fileName`. Datoteka za upis se otvara s pomoću metode `File FS::open( const sString& path, const char* mode, const bool create )`. Način otvaranja `mode` može biti `FILE_READ`, `FILE_WRITE` i `FILE_APPEND`. U ovom se slučaju koristi `FILE_APPEND`. Pri upisu u datoteku `FILE_READ` briše dosadašnji sadržaj. Nakon pozivanja metode `size_t Print::print( const String &s )` datoteka se zatvara s pozivom metoda `void File::close()`.

Metoda `String* SDCommunicator::readLastNlines( const char *fileName, int nOfLines )` služi čitanju zadnjih `nOfLines` linija iz datoteke naziva `fileName`. Rezultati se upisuju u niz `readStrings` koji sadrži `nOfLines` elemenata. Povratna vrijednost je pokazivač na taj niz. Kako bi se iz datoteke pročitali zadnjih `nOfLines` unosa kursor čitača se pomiče na kraj datoteke naredbom `fileRead.seek( 0, SeekEnd )`. Petlja koja čita zadnjih `nOfLines` unosa je definirana kao:

```
1 while ( fileRead.position() > 0 && newlineCount < nOfLines )
2 {
3     fileRead.seek( fileRead.position() - 2 );
4
5     char c = fileRead.read();
6
7     if ( c != '\n' )
8     {
9         line = String( c ) + line;
10    }
11    else
12    {
13        readStrings[ nOfLines - 1 - newlineCount ] = line;
14
15        newlineCount++;
16        line = "";
17    }
18 }
```

Uvjet prekida petlje je da pozicija kursora nije došla na početak datoteke i da je broj znakova novih linija manji od broja vrijednosti koje se žele pročitati. Naredba u liniji 3 pomiče kursor na poziciju 1 bajt iza trenutne pozicije. Odnosno, jedan znak unatrag. U liniji 5 se taj znak upisuje u varijablu `c`. U slučaju da taj znak ne predstavlja znak za novi red on se dodaje nizu znakova `line`. U suprotnom se znakovni niz `line` unosi u prosljeđeni niz `readStrings`. Unosi u datoteku se čitaju od zadnjeg prema prvom. Odnosno, u obrnutom redoslijedu. Kako bi se pravilno upisali potrebno ih je upisivati u niz krenuvši od posljednje pozicije. Indeks posljednje pozicije u nizu je `nOfLines - 1`. Kako bi umanjivali tu vrijednost potrebno je od nje

oduzeti broj već pročitanih linija. Nakon izlaza iz petlje se datoteka zatvara i vraća se pokazivač na niz `readStrings`.

### 3.4.7. Datoteka `ElectronicsControl.h`

Datoteka `ElectronicsControl.h` sadrži prekide, zastavice kojima oni upravljaju i klasu `ElectronicsControl`.

#### 3.4.7.1. Prekidi

Prekidi su definirani prema obrascu:

```
void IRAM_ATTR <naziv_funkcije>()
{
    <zastavica> = ( digitalRead( <povezani_ulaz> ) == <prisutnost_signala> );

    if( !<zastavica> )
    {
        electronicsControl.stopMotor();
    }
}
```

Oni nose imena prikazana u kodu poglavlja 3.4.4.1. Kod prekida generiranih od mehaničkih prekidača u grananje je također dodan uvjet koji provjerava u koju stranu se onemogućava vrtnja motora. Vrtanja motora se kod senzora infracrvenih zraka zaustavlja u oba smjera. Zastavice kojima se mijenja vrijednost se koriste pri provjeri dopuštenog smjera vrtnje motora, kao što je navedeno u poglavlju 3.4.5.3. Ti se prekidi pozivaju na svaku promjenu digitalnog signala u definirane ulaze. Razlog tome je potreba promjene zastavice nakon što signal nestane.

Jedan od izazova kod definiranja strukture prekida je ograničenje na vrijeme trajanje rutine prekida. Iako se ono može mijenjati, zadano vrijeme je oko 300 mikrosekundi, odnosno oko 0,3 milisekunde [52, str. 12]. Ako se to vrijeme premaši mikrokontroler će se resetirati. Testiranje je pokazalo kako korišteni kod ne premašuje dopušteno vrijeme.

#### 3.4.7.2. Konstruktor

U konstruktoru klase se prekida struja do releja kako bi se osigurala njihova zadana pozicija i postavljaju se uvjeti za slanje PWM signala. Za slanje PWM signala je korištena biblioteka *LED Control*. Pošto je teret kojim se upravlja motor druga opcija za biblioteku je *Motor Control Pulse Width Modulator* [53]. Ona omogućuje niže frekvencije PWM signala, dok kod korištene one sežu do najmanje 80 Hz. Korist nižih frekvencija je rjeđa promjena signala. Tako smanjuje cijena promjene signala. Ta cijena može biti ubrzano zagrijavanje MOSFET-a, povećana količina struje kroz motor i slično.

### 3.4.7.3. Postavljanje brzine i smjera

Metoda `void ElectronicsControl::setSpeedNDir( int8_t speedValue )` se koristi za postavljanje brzine i smjera. Prosljeđena vrijednost je garantirano u intervalu od -100 do 100, kao što je objašnjeno u poglavlju 3.4.5.3.

Ako je prosljeđena vrijednost istog predznaka kao i dosadašnja vrijednost brzine pozicija releja se ne mijenja. Provjera predznaka se vrši naredbom `return ( num1 >= 0 ) == ( num2 >= 0 )` u metodi `bool ElectronicsControl::signsAreSame( int num1, int num2 )`.

Struja kroz motor se zaustavlja prije promjene napona na bazu tranzistora. Razlog tomu je činjenica da kontakti releja nisu pogodni za velike istosmjerne struje i napone pri promjeni toka struje [54]. Nakon postavljanja smjera se određuje brzina motora naredbom `ledcWrite( pwmMosfetChannel, valToDuty( speedValue ) )`.

Metoda `int ElectronicsControl::valToDuty( int speedValue )` služi prevađanjem intervala vrijednosti od -100 do 100 u interval vrijednosti od 0 do 255. Razlog tome je što je za definiranje radnog ciklusa PWM signala rezervirano 8 bitova. Vrijednost koja se tada može prosljeđiti je u već spomenutom intervalu. Kod ove metode je:

```
int valToDuty( int8_t speedValue )
{
    return int( 2.551 * abs( speedValue ) );
}
```

Prvi korak je promjena intervala od -100 do 100 u interval od 0 do 100. Tome koristi metoda `int abs( int )` biblioteke `stdlib.h`. Drugi korak je pretvorba intervala od 0 do 100 u interval od 0 do 255. Pri pokušaju zapisa decimalne vrijednosti 2,55 u IEEE 754 definiranom formatu od 64 bita stvarna unesena vrijednost je 2.54999 [55]. Rezultat množenja te vrijednosti s 100 i pretvorba te vrijednosti u cijeli broj daje 254. To znači da je za tu vrijednost radni ciklus signala ispunjen s 99,61%. Zbog toga je potrebno množiti s 2,551.

### 3.4.7.4. Trenutno zaustavljanje struje kroz motor

Trenutno zaustavljanje struje kroz motor se vrši putem metode `void ElectronicsControl::stopMotor()`. Ona sadrži dvije naredbe: poziv funkcije za postavljanje radnog ciklusa PWM signala na 0% i zabilježavanje promjene u varijablu. Ova metoda je posebno korisna jer poziv metode `setSpeedNDir` zahtjeva puno više procesorskog vremena. Kao što je već navedeno, rutine prekida su ograničene s vremenom i stoga je ova metoda nužna kako ne bi došlo do resetiranja mikroprocesora.

## 3.4.8. Datoteka WS2812BControl.h

Datoteka `WS2812BControl.h` sadrži istoimenu klasu. Ona definira metode za promjenu boje i svjetline svjetleće diode ugrađene u Dasduino CONNECTPLUS razvojnu pločicu. Model te diode je `WS2812BControl` [40]. Kako bi postigla te funkcije koristi `FastLED` biblioteku. Boja

te diode se mijenja u funkciji povratnog poziva karakteristike piši.

Za upravljanje ovom diodom se koristi takozvani WS2812B protokol [56]. Njegova kompleksnost je sakrivena putem korištene biblioteke.

U konstruktoru klase se u listu `CRGB leds` dodaje ugrađena dioda. To se vrši naredbom `FastLED.addLeds< WS2812B, 32, RGB >( leds, 1 )`.

U metodi `void setColor( CRGB color )` se mijenja boja diode pozivom metoda `leds[ 0 ] = color` i `FastLED.show()`.

Ona je preopterećena metodom `void setColor( CRGB color, uint8_t intensity )`. Kod te metode se poziva naredba `FastLED.setBrightness( intensity )` nakon čega se poziva istoimena metoda naredbom `setColor( color )`.

## 3.5. Modeliranje računalnog programa Android uređaja

Računalni program Android uređaja temeljen je na programu razvijenom od strane Chee Yi Onga za blog objavljen na Punch Trough Internetskoj stranici [57]. Za ovaj je rad prilagođen kako bi sadržavao karakteristike potrebne korisniku automatizirane ograde. On predstavlja temelj za aplikaciju koja objedinjuje sve moguće korisničke zahtjeve.

### 3.5.1. Razvojna okolina

Program je razvijen s pomoću programa Android Studio. Razvijen je u programskom jeziku Kotlin. Razvijen je za Android verziju 13 i API razinu 33.

### 3.5.2. Struktura programa

Program se sastoji od 9 Kotlin datoteka:

- MainActivity.kt
- BleOperationsActivity.kt
- CharacteristicAdapter.kt
- PermissionsHelper.kt
- ScanResultAdapter.kt
- BleExtensions.kt
- BleOperationType.kt
- ConnectionEventListener.kt
- ConnectionManager.kt

Također sadrži 5 *layout* datoteka:

- activity\_main.xml
- row\_scan\_result.xml
- activity\_ble\_operations.xml
- row\_characteristic.xml
- edittext\_decimal\_payload.xml

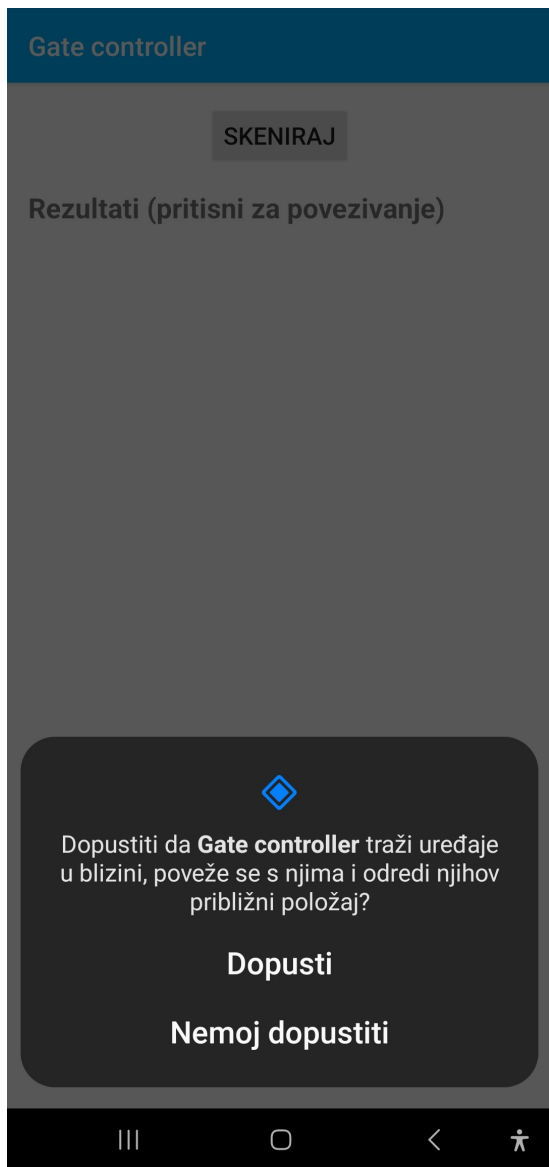
Uz to definira određene resurse i vrijednosti, posebnu ikonu i slično.

### **3.5.3. Tok korisnika**

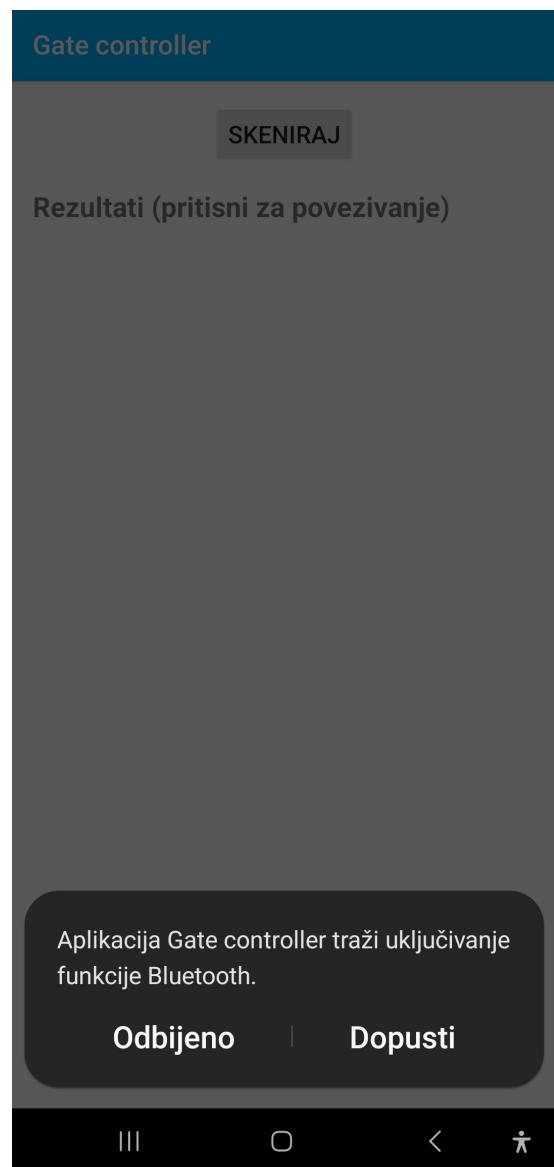
Tok korisnika (eng. User Flow) predstavlja "vizualni pregled toka kojim se korisnik kreće kako bi ispunio određeni zadatak u aplikaciji" [58].

Pri prvom otvaranju aplikacije od korisnika se traže dopuštanja za BLE. Nakon toga se traži uključivanje Bluetootha ako on već nije uključen.





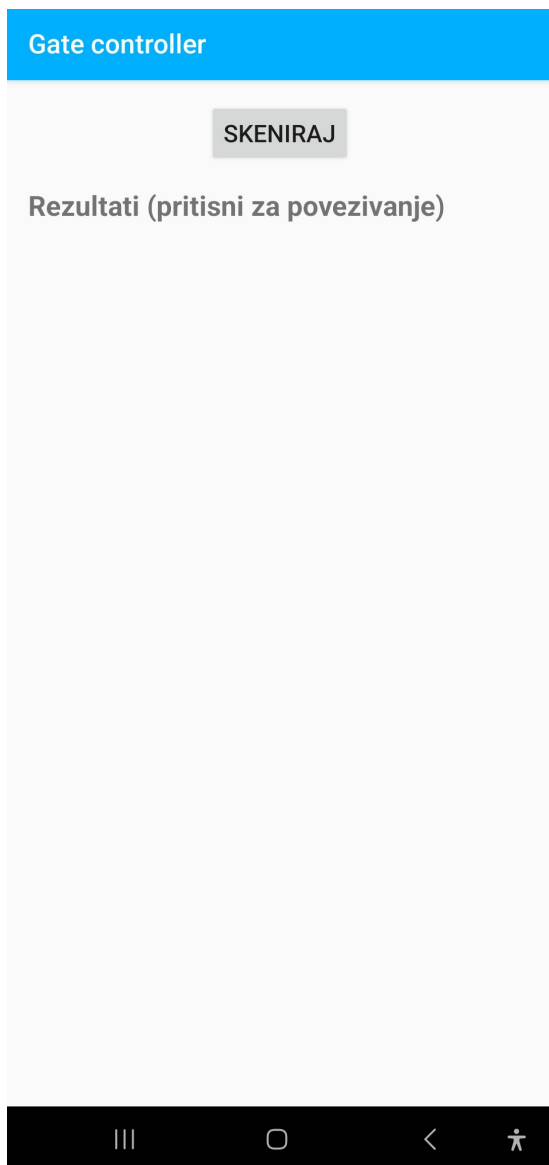
(a) Dopuštenja za BLE



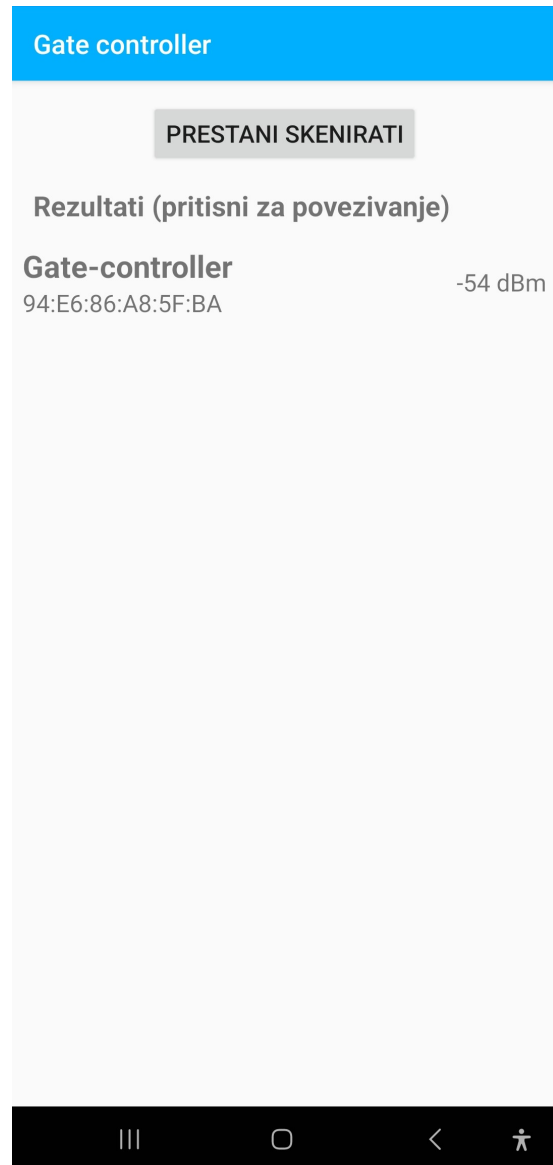
(b) Uključivanje Bluetootha

Slika 15: Ekрани 1 i 2

Nakon davanja dopuštenja korisnik treba pritisnuti gumb označen s tekstem "skeniraj" kako bi otkrio BLE server mikrokontrolera.



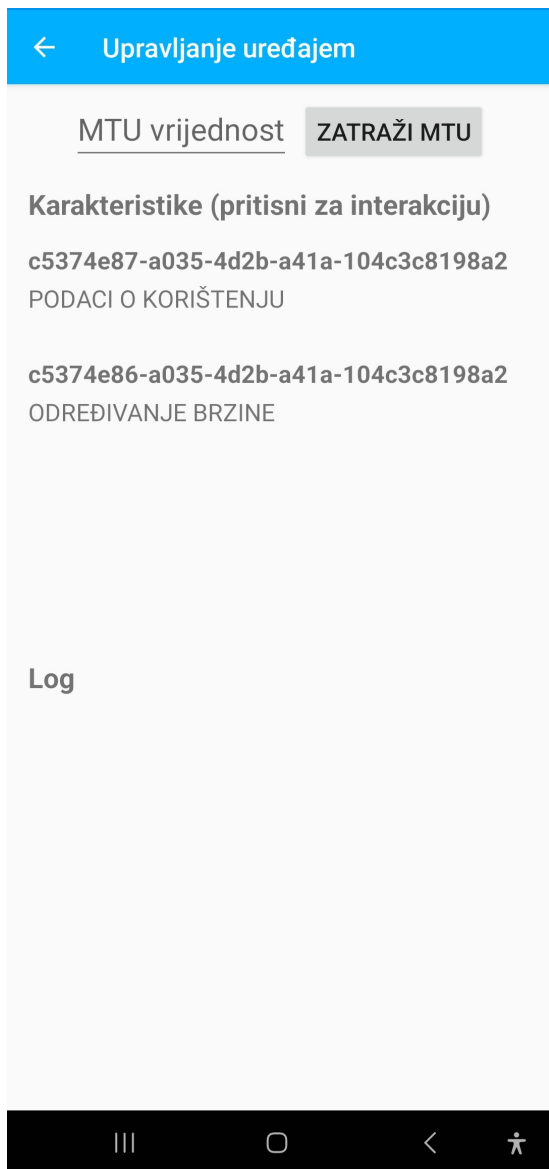
(a) Prije skeniranja



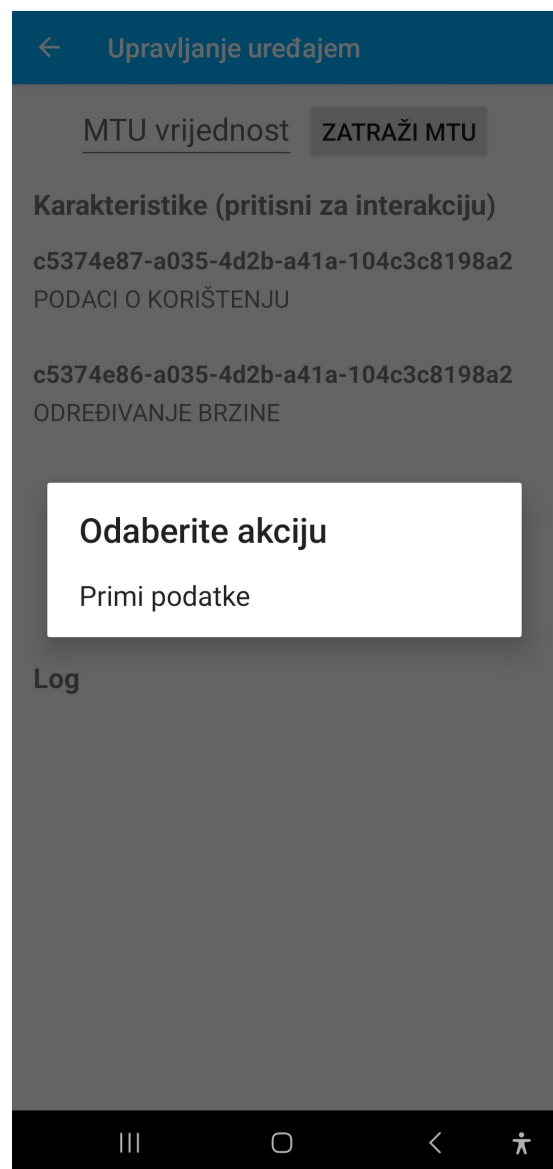
(b) Otkriven mikrokontroler

Slika 16: Ekran 3 i 4

Nakon pritiska na prikazanu opciju "Gate-controller" otvara se ekran s dvije karakteristike: "podaci o korištenju" i "određivanje brzine". Pri pritisku na "podaci o korištenju" otvara se skočni prozor za potvrđivanje radnje.



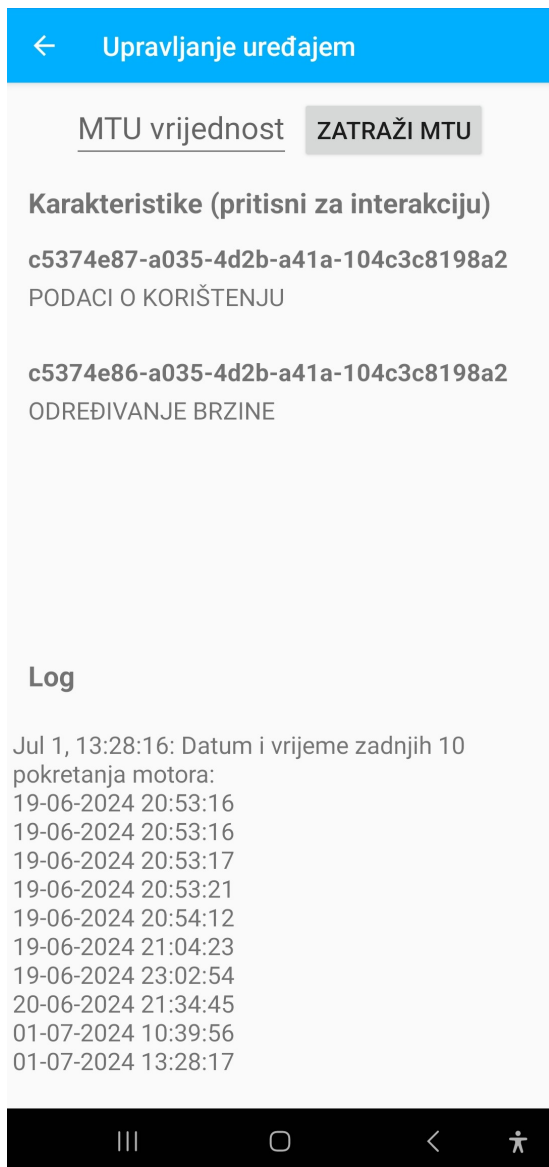
(a) Ponudene karakteristike



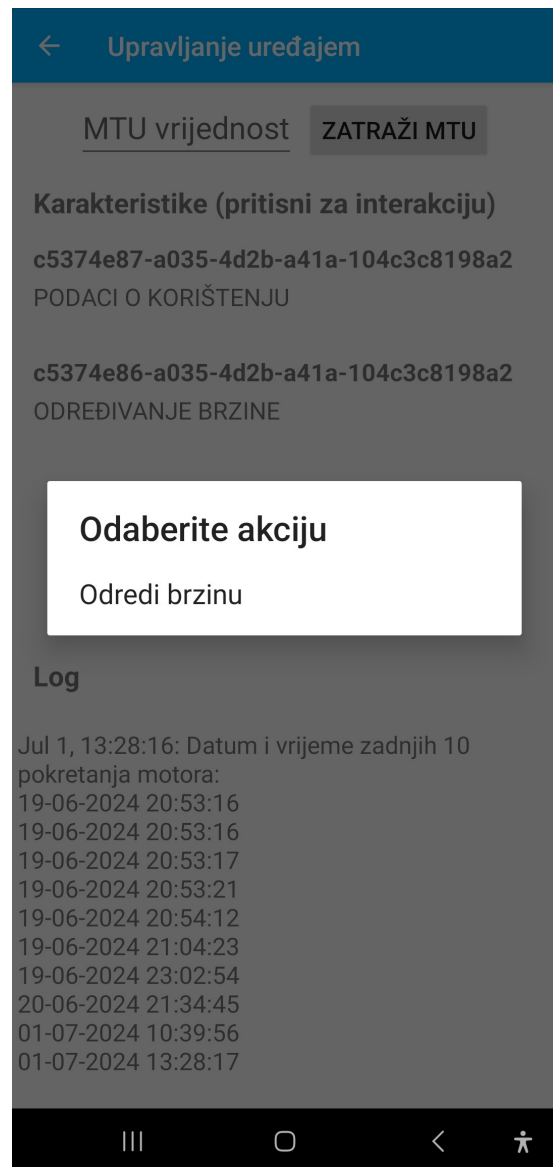
(b) Potvrda radnje "podaci o korištenju"

Slika 17: Ekрани 5 i 6

Nakon primanja vrijednosti o korištenju one se ispisuju u prozor na donjem dijelu ekrana. Ispisuju se u formatu "dd-MM-yyyy HH:mm:ss". Nakon pritiska na opciju "određivanje brzine" otvara se skočni prozor za potvrdu radnje.



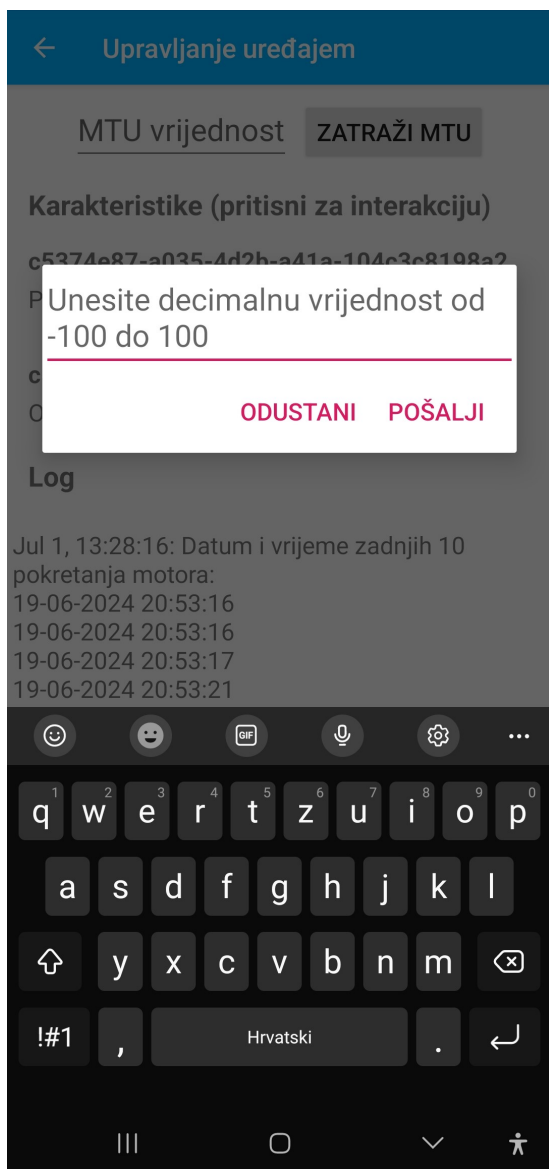
(a) Primljene vrijednosti



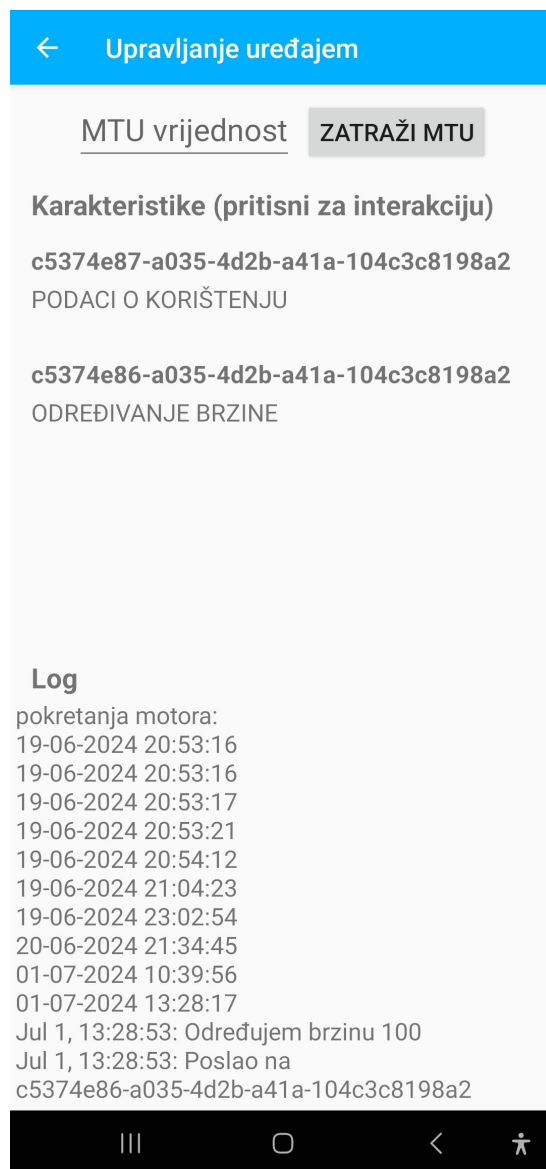
(b) Potvrda radnje "određivanje brzine"

Slika 18: Ekрани 7 i 8

Nakon potvrde radnje "određivanje vrijednosti" korisniku se predstavi okvir za unos vrijednosti. U njegovoj pozadini su mu definirane instrukcije: "Unesite decimalnu vrijednost od -100 do 100". Nakon što se vrijednost unese i nakon što se pritisne gumb "pošalji" na donjem dijelu ekrana se ispiše potvrdna poruka.



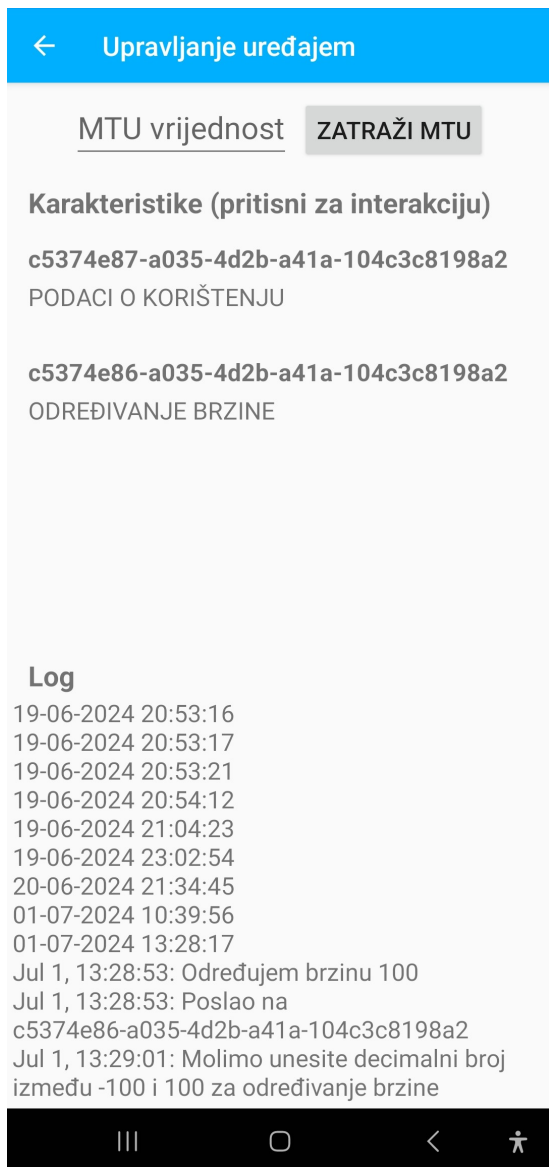
(a) Unos vrijednosti



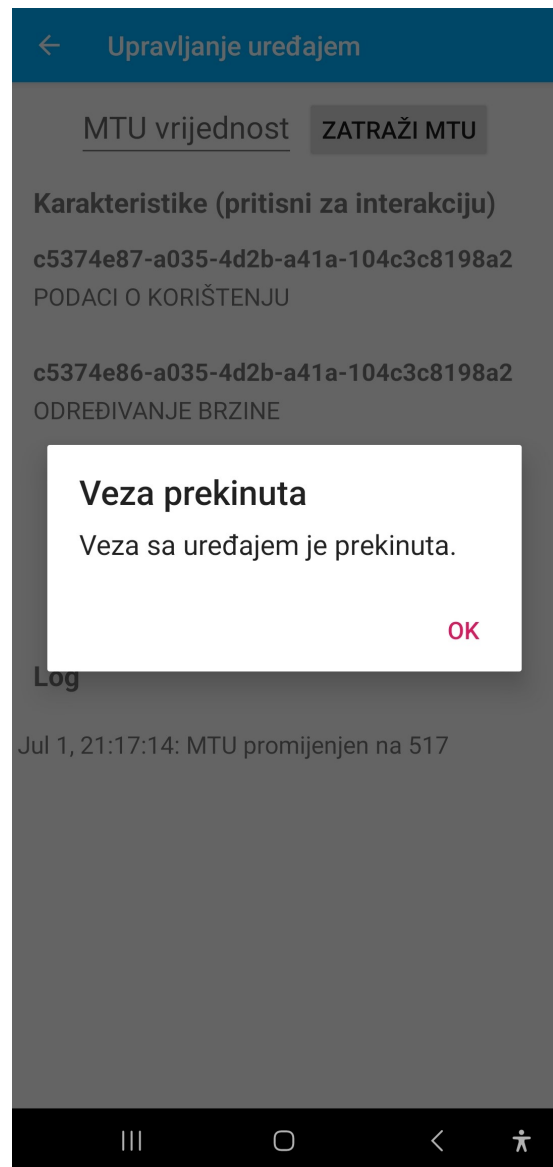
(b) Potvrдна poruka određivanja vrijednosti

Slika 19: Ekрани 9 i 10

U slučaju unosa vrijednosti van intervala od -100 do 100 korisniku se u donjem dijelu ekrana ispisuje poruka greške. Korisnika se obavještava putem skočnog prozora u slučaju prekida veze. Prekid veze može doći i od strane servera i klijenta. Klijent može prekinuti vezu pritiskom na strelicu u gornjem lijevom kutu ekrana.



(a) Poruka greške određivanja vrijednosti



(b) Obavještanje korisnika o prekidu veze

Slika 20: Ekрани 11 i 12

### 3.5.4. Načini ostvarenja funkcionalnosti

#### 3.5.4.1. Traženje dopuštenja

Kao što je navedeno na Internetskoj stranici za Android razvoj, za mogućnosti skeniranja i povezivanja na BLE uređaje potrebna su dopuštenja `BLUETOOTH_SCAN` i `BLUETOOTH_CONNECT` [59].

#### 3.5.4.2. Uključivanje Bluetootha

Zahtjev za uključanjem Bluetootha se podnosi samo ako su relevantna dopuštenja predana. On se generira putem naredbe unutar skripte `MainActivity.kt`:

```
Intent (BluetoothAdapter.ACTION_REQUEST_ENABLE).apply {
    bluetoothEnablingResult.launch (this)
}
```

### 3.5.4.3. Skeniranje uređaja

Ako su dopuštenja za Bluetooth dane onda se brišu dosadašnji rezultati skeniranja i naredbom `bleScanner.startScan( null, scanSettings, scanCallback )` započinje novo. `scanCallback` predstavlja funkciju povratnog poziva za otkriven uređaj. U ovoj se aplikaciji rezultati filtriraju kako bi se prikazali samo uređaji s imenom "Gate-controller".

### 3.5.4.4. Povezivanje na BLE server

U trenutku pritiska na željeni otkriveni uređaj poziva se naredba `result.device.ConnectionManager.connect( this, this@MainActivity ).Unutar metode ConnectionManager::connect( device: BluetoothDevice, context: Context) se poziva naredba enqueueOperation( Connect( device, context.applicationContext ) ).`

U trenutku uspostave veze izvršava se naredba pokretanja aktivnosti `BleOperationsActivity:`

```
Intent( this@MainActivity, BleOperationsActivity::class.java ).also {
    it.putExtra( BluetoothDevice.EXTRA_DEVICE, gatt.device )
    startActivity( it )
}
```

### 3.5.4.5. Ispis karakteristika

Na ekran aktivnosti `BleOperationsActivity` se ispisuju karakteristike povezanog uređaja. Kako bi se prikazale samo relevantne karakteristike one se filtriraju pomoću naredbe:

```
filteredCA.items = ca.items.filter {
    it.uuid == UUID.fromString("c5374e87-a035-4d2b-a41a-104c3c8198a2")
    || it.uuid == UUID.fromString("c5374e86-a035-4d2b-a41a-104c3c8198a2") }
```

### 3.5.4.6. Primanje i ispisivanje podataka

Kod pritiska na karakteristiku "podaci o korištenju" šalju se 11 uzastopnih zahtjeva za podacima. U trenutnoj konfiguraciji prvih 10 vrijednosti predstavlja broj milisekundi od pokretanja mikrokontrolera. Zadnji predstavlja broj milisekundi proteklih od uključivanja mikrokontrolera do ovog trenutka.

```
1 onCharacteristicRead = { _, characteristic, value ->
2     readValues.add(value.toAsciiString())
3
4     if (amountRead < 10) {
```

```

5         ConnectionManager.readCharacteristic(device, characteristic)
6         amountRead += 1
7     } else {
8         amountRead = 0
9
10        val referenceMillis = readValues[ readValues.size - 1 ].trim().toLong()
11        Timber.i("ReferenceMillis=_$_referenceMillis")
12        val now = Instant.now()
13
14        for(ind in 0..(readValues.size - 2)){
15            val millisDiff = referenceMillis - readValues[ind].trim().toLong()
16
17            val earlierTime = now.minusMillis(millisDiff)
18
19            val formatter: DateTimeFormatter = DateTimeFormatter.ofPattern("dd-MM-
                yyyy_HH:mm:ss").withZone(
20                ZoneId.systemDefault() )
21            shortLog(formatter.format(earlierTime).toString());
22        }
23    }
24 }

```

Nakon prikupljanja svih vrijednosti odredi se koliko je vremena proteklo od trenutnog do svakog od zabilježenih vremena otvaranja. To je predstavljeno u liniji 15. Od trenutnog vremena se oduzima taj broj milisekundi. To se dobiveno vrijeme formatira s pomoću naredbe linije 19. Na kraju se poziva metoda `private fun shortLog( message: String )` klase `BleOperationsActivity` istoimene skripte. Ona unosi taj znakovni niz u tekstualni obrazac na dnu ekrana.

### 3.5.4.7. Određivanje brzine

Pritiskom na karakteristiku "određivanje brzine" otvara se obrazac za unos decimalne vrijednosti. Obrazac onemogućava unos znakova osim decimalnih brojeva i znaka za minus. U trenutku pritiska gumba "pošalji" izvršava se sljedeći kod:

```

if ( isNotBlank() && isEmpty() && isInRange( decimalField.text.toString() ) )
{
    val bytes = decimalToBytes()

    log( "Odreujem_brzinu_$_{decimalField.text.toString()}" )
    ConnectionManager.writeCharacteristic( device, characteristic, bytes )
} else {
    log( "Molimo_unesite_decimalni_broj_dizmeu_-100_i_100_za_dodreivanje_brzine" )
}

```

Funkcija `private fun log( message: String )` radi istu funkciju kao i `shortLog()`, samo što uz tekst ispisuje i trenutni datum i vrijeme.



### 3.5.4.8. Prekid veze s uređajem

Skočni prozor s informacijom o prekidu veze se prikazuje putem sljedeće naredbe:

```
AlertDialog.Builder(this@BleOperationsActivity)
    .setTitle("Veza_prekinuta")
    .setMessage("Veza_s_uredajem_je_prekinuta.")
    .setPositiveButton("OK") { _, _ -> onBackPressed() }
    .show()
```

## 4. Zaključak

Područje Interneta stvari obuhvaća elektroničko inženjerstvo, računalno inženjerstvo, umrežene sustave, ugrađene sustave, računalstvo u oblaku, znanost o podacima, umjetnu inteligenciju i mnogo više. Širina tog područja zahtjeva raznorazne vještine od strane inženjera ili njegovog tima. Razvoj mobilne aplikacije za komunikaciju s mikrokontrolerom ne može započeti dok format poruka koje razmjenjuju nije poznat. Format poruka nije poznat dok se ne utvrde podaci koje mikrokontroler prikuplja. Podaci koje mikrokontroler prikuplja ovise o daljnjim okolnostima. Snaga motora utječe na jednostavnost njegovog ugrađivanja. Pristup njegovom napajanju utječe na njegovu primjenjivost. Način njegova napajanja utječe na način njegovog kontroliranja, što ponovno utječe na izgled cijele elektroničke mreže. Od prvog koraka do zadnjeg, područje Interneta stvari obuhvaća najmanje detalje implementacije. A ti detalji često imaju velik utjecaj na izgled sustava.

Najveći problem s kojim se područje razvoja ovakvih sustava suoči je vrijeme. Kako se sustav može poboljšati? Kako može biti robusniji? Kako ga se može bolje testirati? Kako se može umanjiti vjerojatnost greške? Kako se može osigurati dulji vijek trajanja? Sve su ovo pitanja na koje je moguće odgovoriti s istim odgovorom - produljenjem vremena dodijeljenog za njegov razvitak. I dokle god to vrijeme nije beskonačno razvijeni sustavi će biti nesavršeni, do neke razine manjkavi. Takvi sustavi predstavljaju teret, ako ne i nesigurnost ili čak opasnost. A uvijek postoji detalj koji nije bio razmotren ili detalj koji nije pokriven. I zato su ažuriranja preko mreže toliko važan pojam u programiranju ugrađenih sustava. Jer uvijek postoji to nešto. Ali pitanje je: koliko to nešto može biti opasno? Što ako zakaže infracrveni senzor rezultirajući se ozljedom čovjeka koji prolazi kroz vrata ograde? Što ako upravljačka ploča zakaže i motor krene nekontrolirano ubrzavati? Što ako zloćudni haker probije u sustav i dobije kontrolu nad njime? Ponovno, postoji rizik sa svakim sustavom. A u području Interneta stvari taj je sustav golem i teško je kontrolirati svaku varijablu.

U svakom slučaju, područje Interneta stvari i automatizacije može donijeti veliko bogatstvo čovječanstvu. Ako se prema njemu pristupi oprezno i odgovorno. Ako ne, sami sustavi bi mogli postati teret, nesigurnost ili čak opasnost.

# Popis literature

- [1] P. Rajak, A. Ganguly, S. Adhikary i S. Bhattacharya, „Internet of Things and smart sensors in agriculture: Scopes and challenges,” *Journal of Agriculture and Food Research*, sv. 14, str. 100776, 2023., ISSN: 2666-1543. DOI: <https://doi.org/10.1016/j.jafr.2023.100776>. adresa: <https://www.sciencedirect.com/science/article/pii/S2666154323002831>.
- [2] *Gigafactory Berlin-Brandenburg*, en. adresa: <https://www.tesla.com/giga-berlin> (pogledano 24. 6. 2024.).
- [3] *Smart City Observatory 2024 - IMD business school for management and leadership courses*, en, travanj 2024. adresa: [https://www.imd.org/wp-content/uploads/2024/04/20240412-SmartCityIndex-2024-Full-Report\\_4.pdf](https://www.imd.org/wp-content/uploads/2024/04/20240412-SmartCityIndex-2024-Full-Report_4.pdf) (pogledano 1. 7. 2024.).
- [4] *10 inventions that began life as science fiction - The New Economy*. adresa: <https://www.theneweconomy.com/technology/10-inventions-that-began-life-as-science-fiction> (pogledano 24. 6. 2024.).
- [5] *Automatic Gate Opening System Market Forecast To 2031*. adresa: <https://www.businessresearchinsights.com/market-reports/automatic-gate-opening-system-market-106035> (pogledano 24. 6. 2024.).
- [6] *Circuit Simulator Applet*. adresa: <https://www.falstad.com/circuit/> (pogledano 24. 6. 2024.).
- [7] V. Pinter, *Osnove elektrotehnike*, en. Tehnička knjiga, 1994., Google-Books-ID: rpAtGwA-ACAAJ, ISBN: 978-953-172-006-9.
- [8] P. Horowitz i W. Hill, *The Art of Electronics*, 3rd. Cambridge: Cambridge University Press, 2015., ISBN: 978-0-521-80926-9.
- [9] *TR21-SERIES-EN-datasheet*. adresa: <https://www.tme.eu/Document/9bb0130deec8de834650-TR21-SERIES-EN.pdf> (pogledano 24. 6. 2024.).
- [10] *ispravljač, mosni - Tehnički leksikon*. adresa: <https://tehnicki.lzmk.hr/clanak/ispravljac-mosni> (pogledano 24. 6. 2024.).
- [11] *BYT79-500 | WeEn*. adresa: <https://www.ween-semi.com/en/product/BYT79-500> (pogledano 25. 6. 2024.).

- [12] *Bridge Rectifier With Capacitor Filter: Circuit Diagram and Explain Step by Step | Voltage Lab*, en-US, Section: Electronics, travanj 2022. adresa: <https://www.voltagegellab.com/bridge-rectifier-with-capacitor-filter/> (pogledano 25. 6. 2024.).
- [13] *Guarding requirements for 50 volts or more DC. | Occupational Safety and Health Administration*. adresa: <https://www.osha.gov/laws-regs/standardinterpretations/2015-09-04> (pogledano 25. 6. 2024.).
- [14] *Capacitor & Capacitance Experiments for Science Labs & Science Fair Projects*. adresa: [https://www.juliantrubin.com/encyclopedia/electronics/capacitor.html#Hazards\\_and\\_safety](https://www.juliantrubin.com/encyclopedia/electronics/capacitor.html#Hazards_and_safety) (pogledano 25. 6. 2024.).
- [15] L. Posarić, *Pulse width modulation (PWM)*, nastavni materijali na kolegiju Programiranje ugrađenih sustava [Moodle], Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, Hrvatska, 2024.
- [16] *The Virtues of PWM Combined With a Power MOSFET - News*, en. adresa: <https://www.allaboutcircuits.com/news/the-virtues-of-a-pwm-combined-with-power-mosfet/> (pogledano 25. 6. 2024.).
- [17] I. T. AG, *IRL540N datasheet - Infineon Technologies*, en. adresa: [https://www.infineon.com/dgdl/Infineon-IRL540N-DataSheet-v01\\_01-EN.pdf?fileId=5546d462533600a40153565fc2a62567](https://www.infineon.com/dgdl/Infineon-IRL540N-DataSheet-v01_01-EN.pdf?fileId=5546d462533600a40153565fc2a62567) (pogledano 27. 6. 2024.).
- [18] *How to Read a Power MOSFET Datasheet - Jason Sachs*. adresa: <https://www.embeddedrelated.com/showarticle/809.php> (pogledano 25. 6. 2024.).
- [19] *High Side and Low Side Switches*. adresa: <https://www.rohm.com/electronics-basics/ipds/high-side-and-low-side-drive> (pogledano 26. 6. 2024.).
- [20] *tranzistor, bipolarni - Tehnički leksikon*. adresa: <https://tehnicki.lzmk.hr/clanak/tranzistor-bipolarni> (pogledano 26. 6. 2024.).
- [21] *2N2222A - Bipolar Transistors Datasheet*, en. adresa: <https://diotec.com/request/datasheet/2n2222a.pdf> (pogledano 26. 6. 2024.).
- [22] *Transistor Operating Details*. adresa: <http://hyperphysics.phy-astr.gsu.edu/hbase/Solids/basemit.html> (pogledano 26. 6. 2024.).
- [23] *36.11.9.005.4001 - General Purpose Relay, 36 Series, Power, SPDT, 5 VDC, 10 A*, en. adresa: <https://www.farnell.com/datasheets/2626622.pdf> (pogledano 26. 6. 2024.).
- [24] *dioda, poredna - Tehnički leksikon*. adresa: <https://tehnicki.lzmk.hr/clanak/dioda-poredna> (pogledano 26. 6. 2024.).
- [25] *How Does a Flyback Diode Work?* en, lipanj 2021. adresa: <https://www.circuitbread.com/ee-faq/how-does-a-flyback-diode-work> (pogledano 26. 6. 2024.).
- [26] *Freewheeling Diode or Flyback Diode Circuit Working And Its Functions*. adresa: <https://www.elprocus.com/freewheeling-or-flyback-diode-circuit-working-functions/> (pogledano 26. 6. 2024.).
- [27] *Voltage Dividers - SparkFun Learn*. adresa: <https://learn.sparkfun.com/tutorials/voltage-dividers/all> (pogledano 26. 6. 2024.).

- [28] *ESP32-WROVER-E datasheet*, en, 2023. adresa: [https://www.espressif.com/sites/default/files/documentation/esp32-wrover-e\\_esp32-wrover-ie\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wrover-e_esp32-wrover-ie_datasheet_en.pdf) (pogledano 26. 6. 2024.).
- [29] *Voltage divider (article) | Circuit analysis | Khan Academy*. adresa: <https://www.khanacademy.org/science/electrical-engineering/ee-circuit-analysis-topic/ee-resistor-circuits/a/ee-voltage-divider> (pogledano 27. 6. 2024.).
- [30] electrical4uonline, *Universal Motor (Complete Guide)*, en-us, kolovoz 2020. adresa: <https://www.electrical4uonline.com/universal-motor/> (pogledano 27. 6. 2024.).
- [31] S. Herman, *Delmar's Standard Textbook of Electricity* (MindTap Course List Series). Cengage Learning, 2016., ISBN: 9781285852706. adresa: <https://books.google.hr/books?id=E80FCwAAQBAJ>.
- [32] *SL3000UL Slide Gate Operator*, en. adresa: <https://www.liftmaster.com/sl3000ul-slide-gate-operator/p/SL3000UL> (pogledano 27. 6. 2024.).
- [33] *Electric Motors for Gates: Powerful and Reliable Options*. adresa: <https://allsecurityequipment.com/blogs/blog/electric-motors-for-gates> (pogledano 27. 6. 2024.).
- [34] T. Carrick, *The Best Automatic Gate Openers of 2024*, en-US, ožujak 2021. adresa: <https://www.bobvila.com/articles/best-automatic-gate-opener/> (pogledano 27. 6. 2024.).
- [35] *Amazon.com: CO-Z 1400lb Sliding Gate Opener*. adresa: <https://www.amazon.com/dp/B0197V30D2/?tag=best-automatic-gate-opener-20> (pogledano 27. 6. 2024.).
- [36] *V2 Sensiva XS*. adresa: <https://v2home.com/en/product/miniaturized-wall-mounted-and-pillar-mounted-photocells-synchronised-and-self-aligning/> (pogledano 27. 6. 2024.).
- [37] *V2 Sensiva XS datasheet*. adresa: [https://v2home.com/wp-content/uploads/2018/06/SENSIVA-XS\\_IL374\\_10.09.2018.pdf](https://v2home.com/wp-content/uploads/2018/06/SENSIVA-XS_IL374_10.09.2018.pdf) (pogledano 27. 6. 2024.).
- [38] E. White, *Making Embedded Systems: Design Patterns for Great Software*, en. "O'Reilly Media, Inc.", studeni 2011., ISBN: 978-1-4493-0214-6.
- [39] *Dasduino CONNECTPLUS*. adresa: <https://soldered.com/hr/proizvod/dasduino-connectplus/> (pogledano 27. 6. 2024.).
- [40] *SolderedElectronics/Dasduino-CONNECTPLUS-hardware-design*, original-date: 2022-10-28T09:15:05Z, svibanj 2024. adresa: <https://github.com/SolderedElectronics/Dasduino-CONNECTPLUS-hardware-design/blob/main/OUTPUTS/V1.1.1/Dasduino%20CONNECTPLUS%20Schematics.pdf> (pogledano 27. 6. 2024.).
- [41] *TPS7A26 datasheet*. adresa: <https://www.ti.com/lit/ds/symlink/tps7a26.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-ww&ts=1719509532683> (pogledano 27. 6. 2024.).
- [42] *What is PlatformIO? — PlatformIO latest documentation*. adresa: <https://docs.platformio.org/en/latest/what-is-platformio.html> (pogledano 28. 6. 2024.).

- [43] *Welcome to ESP32 Arduino Core's documentation - - — Arduino ESP32 latest documentation.* adresa: <https://docs.espressif.com/projects/arduino-esp32/en/latest/index.html> (pogledano 28. 6. 2024.).
- [44] *attachInterrupt() - Arduino Reference.* adresa: <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/> (pogledano 28. 6. 2024.).
- [45] *Pull-Down Resistor: What is it (And How Does it Work)? | Electrical4U.* adresa: <https://www.electrical4u.com/pull-down-resistor/> (pogledano 25. 6. 2024.).
- [46] *Bluetooth Technology Overview | Bluetooth® Technology Website.* adresa: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/> (pogledano 28. 6. 2024.).
- [47] *Bluetooth low energy Characteristics, a beginner's tutorial.* adresa: <https://devzone.nordicsemi.com/guides/short-range-guides/b/bluetooth-low-energy/posts/ble-characteristics-a-beginners-tutorial> (pogledano 1. 7. 2024.).
- [48] *ESP32\_BLE\_Arduino/src/BLECharacteristic.h at master · nkolban/ESP32\_BLE\_Arduino · GitHub.* adresa: [https://github.com/nkolban/ESP32\\_BLE\\_Arduino/blob/master/src/BLECharacteristic.h](https://github.com/nkolban/ESP32_BLE_Arduino/blob/master/src/BLECharacteristic.h) (pogledano 1. 7. 2024.).
- [49] *ESP32 BLE Arduino - Arduino Reference.* adresa: <https://www.arduino.cc/reference/en/libraries/esp32-ble-arduino/> (pogledano 1. 7. 2024.).
- [50] *millis() - Arduino Reference.* adresa: <https://www.arduino.cc/reference/en/language/functions/time/millis/> (pogledano 1. 7. 2024.).
- [51] *unsigned long - Arduino Reference.* adresa: <https://www.arduino.cc/reference/en/language/variables/data-types/unsignedlong/> (pogledano 1. 7. 2024.).
- [52] L. Posarić, *Inputs and Interrupts*, nastavni materijali na kolegiju Programiranje ugrađenih sustava [Moodle], Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, Hrvatska, 2024.
- [53] *Motor Control Pulse Width Modulator (MCPWM) - ESP32 - — ESP-IDF Programming Guide latest documentation.* adresa: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/mcpwm.html> (pogledano 1. 7. 2024.).
- [54] *Understand Relay Specifications to Get the Most Out of Your Switching System.* adresa: <https://www.pickeringtest.com/en-us/kb/hardware-topics/relay-reliability/switching-and-relay-specifications> (pogledano 1. 7. 2024.).
- [55] *Base Convert: IEEE 754 Floating Point.* adresa: <https://baseconvert.com/ieee-754-floating-point> (pogledano 1. 7. 2024.).
- [56] *WS2812 LED: Addressable LED Protocol Tutorial | Arrow.com.* adresa: <https://www.arrow.com/en/research-and-events/articles/protocol-for-the-ws2812b-programmable-led> (pogledano 1. 7. 2024.).
- [57] C. Y. Ong, *The Ultimate Guide to Android Bluetooth Low Energy*, en-US, travanj 2024. adresa: <https://punchthrough.com/android-ble-guide/> (pogledano 1. 7. 2024.).

- [58] D. Plantak-Vukovac, *PROCES DIZAJNA KORISNIČKOG ISKUSTVA - 2. faza: DIZAJN; Konceptualni dizajn interakcija*, nastavni materijali na kolegiju Razvoj interaktivnih sustava [Moodle], Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, Hrvatska, 2024.
- [59] *Bluetooth permissions | Connectivity | Android Developers*. adresa: <https://developer.android.com/develop/connectivity/bluetooth/bt-permissions#declare-android12-or-higher> (pogledano 1. 7. 2024.).
- [60] *krug, električni - Tehnički leksikon*. adresa: <https://tehnicki.lzmk.hr/clanak/krug-elektricni> (pogledano 25. 6. 2024.).
- [61] *mreža, električna - Tehnički leksikon*. adresa: <https://tehnicki.lzmk.hr/clanak/mreza-elektricna> (pogledano 25. 6. 2024.).
- [62] *transformator - Hrvatska enciklopedija*. adresa: <https://www.enciklopedija.hr/clanak/transformator> (pogledano 25. 6. 2024.).
- [63] *ispravljač - Hrvatska enciklopedija*. adresa: <https://www.enciklopedija.hr/clanak/ispravljac> (pogledano 24. 6. 2024.).
- [64] *pretvarač, istosmjerni - Tehnički leksikon*. adresa: <https://tehnicki.lzmk.hr/clanak/pretvarac-istosmjerni> (pogledano 24. 6. 2024.).
- [65] *modulacija širine impulsa - Tehnički leksikon*. adresa: <https://tehnicki.lzmk.hr/clanak/modulacija-sirine-impulsa> (pogledano 25. 6. 2024.).
- [66] *MOSFET - Tehnički leksikon*. adresa: <https://tehnicki.lzmk.hr/clanak/mosfet> (pogledano 25. 6. 2024.).
- [67] *releji - Tehnički leksikon*. adresa: <https://tehnicki.lzmk.hr/clanak/relej> (pogledano 26. 6. 2024.).
- [68] *tahogenerator - Tehnički leksikon*. adresa: <https://tehnicki.lzmk.hr/clanak/tahogenerator> (pogledano 27. 6. 2024.).

# Popis slika

1.	Shema sustava . . . . .	6
2.	Shema transformatora i ispravljača . . . . .	7
3.	Shema mosnog ispravljača . . . . .	8
4.	Pad napona iz mosnog ispravljača kroz motor . . . . .	9
5.	Shema upravljača brzinom . . . . .	10
6.	PWM signali iz korištenog mikrokontrolera, radni ciklus 50% . . . . .	11
7.	PWM signali iz logičkog pretvarača, radni ciklus 50% . . . . .	13
8.	Shema upravljača smjerom . . . . .	13
9.	Shema releja i prekidača . . . . .	15
10.	Shema naponskog djelila . . . . .	18
11.	Shema motora . . . . .	19
12.	Shema odašiljača i prijemnika infracrvenih zraka . . . . .	20
13.	Shema pločice za spajanje SD kartice . . . . .	21
14.	Shema mikrokontrolera . . . . .	22
15.	Ekрани 1 i 2 . . . . .	32
16.	Ekрани 3 i 4 . . . . .	33
17.	Ekрани 5 i 6 . . . . .	34
18.	Ekрани 7 i 8 . . . . .	35
19.	Ekрани 9 i 10 . . . . .	36
20.	Ekрани 11 i 12 . . . . .	37



# Prilog - elektronički izrazi i definicije

1. Električni krug je "jednostavna el. mreža koja sadržava barem jedan zatvoreni strujni put." [60]
2. Električna mreža je "struktura međusobno povezanih idealiziranih el. elemenata, od kojih je svaki karakteriziran određenim funkcionalnim odnosom definicijski ovisnih varijabla. Model je stvarne el. mreže sastavljene za obavljanje određene zadaće, koja je skup el. naprava međusobno spojenih el. vodičima preko priključnica." [61]
3. Transformator je "električni uređaj bez pokretnih dijelova koji međuinduktivno povezuje dva električna strujna kruga izmjenične struje i izmjeničnu električnu struju zadanoga električnoga napona pretvara u izmjeničnu električnu struju višega ili nižega električnoga napona." [62]
4. Ispravljač je "električni uređaj za napajanje trošila pretvaranjem izmjenične električne struje u istosmjernu." [63]
5. Mosni ispravljač je "ispravljač s pretvaračkim sklopom u Graetzovu mosnom spoju. Jednofazni je mosni spoj sastavljen od dvaju dvofaznih ispravljačkih spojeva sa srednjom točkom, a trofazni od dvaju trofaznih spojeva sa srednjom točkom." [10]
6. Pretvarač istosmjerne struje je "učinski pretvarač za istosmjernu pretvorbu, koji spaja dva istosmjerna sustava različitih naponskih razina." [64]
7. Modulacija širine impulsa (engl. *pulse width modulation*, akr. PWM) je "mijenjanje širine impulsa visokofrekvencijskoga pravokutnoga signala modulacijskim signalom, koji nosi informaciju o željenom osnovnom harmoniku struje ili napona na trošilu." [65]
8. MOSFET, MOS-tranzistor (akr. od engl. *Metal-Oxide-Semiconductor FET*) je "unipolarni tranzistor upravljani naponom. Strujom kroz kanal upravlja se naponima  $U_{DS}$  i  $U_{GS}$ . Napon praga  $U_{GS0}$  napon je pri kojem nastaje kanal i struja počinje teći." [66]
9. *Pull-down* otpornik je otpornik koji "se koristi kako bi se osiguralo da signal u digitalnom krugu ostane u niskom stanju kada nema drugih utjecaja." [45]
10. Releji je "uređaj koji el. strujom u jednom krugu uključuje ili isključuje el. struju u drugom krugu. Tako se razmjerno malom snagom u upravljačkom krugu mogu uključivati ili isključivati znatno veće snage u drugom krugu." [67]

11. Tahogenerator je "generator istosmjerne ili izmjenične struje male snage za mjerenje brzine vrtnje osovine. Napon mu je razmjern brzini vrtnje." [68]