

# DIZAJN DALJINSKOG UPRAVLJAČA DASKE ZA SURFANJE

---

**Petrić, Ana**

**Undergraduate thesis / Završni rad**

**2023**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Split / Sveučilište u Splitu**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:228:372128>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-01-29**



*Repository / Repozitorij:*

[Repository of University Department of Professional Studies](#)



**SVEUČILIŠTE U SPLITU**  
**SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE**

Prijediplomski stručni studij Elektronika

**ANA PETRIĆ**

**ZAVRŠNI RAD**

**DIZAJN DALJINSKOG UPRAVLJAČA DASKE ZA  
SURFANJE**

Split, rujan 2023.

**SVEUČILIŠTE U SPLITU**  
**SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE**

Prijediplomski stručni studij Elektronika

**Predmet:** Elektronički sklopovi

**ZAVRŠNI RAD**

**Kandidat:** Ana Petrić

**Naslov rada:** Dizajn daljinskog upravljača daske za surfanje

**Mentor:** dr. sc. Tonko Kovačević, prof. struč. stud.

Split, rujan 2023.

# SADRŽAJ

SAŽETAK .....	1
1. UVOD.....	2
2. KORIŠTENE KOMPONENTE .....	4
2.1. DOIT ESP DEV KIT v1 .....	4
2.2. 1.8" TFT ekran.....	6
2.3. Hall senzori.....	8
2.3.1. Analogni A49E Hall senzor.....	9
2.3.2. Digitalni A3144 Hall senzor.....	11
3. KOMUNIKACIJSKI SUSTAV .....	14
3.1. Bluetooth komunikacija.....	15
3.1.1. Bluetooth Low Energy .....	15
3.1.2. Bluetooth Classic .....	17
3.2. SPI serijska komunikacija .....	22
3.3. Pulsno-širinska modulacija.....	23
4. DIZAJN SUSTAVA.....	26
4.1. Daljinski upravljač.....	27
4.2. Prijemnik u električnoj surf dasci .....	29
5. PROGRAMSKA PODRŠKA.....	31
5.1. Programska podrška daljinskog upravljača .....	31
5.2. Programska podrška prijemnika u električnoj surf dasci.....	35
5.3. Testiranje programa.....	38
6. 3D MODELIRANJE DALJINSKOG UPRAVLJAČA .....	41
7. ZAKLJUČAK.....	45
LITERATURA .....	46

POPIS SLIKA .....	48
POPIS TABLICA.....	50
PRILOG 1 – PROGRAMSKI KOD DALJINSKOG UPRAVLJAČA .....	51
PRILOG 2 – PROGRAMSKI KOD PRIJEMNIKA U SURF DASCİ .....	54

# SAŽETAK

## **Dizajn daljinskog upravljača daske za surfanje**

Ovaj završni rad obuhvaća projektiranje i dizajn daljinskog upravljača za električnu surf dasku korištenjem ESP32 mikrokontrolera, analognog i digitalnog Hall senzora i TFT 1.8" ekrana. Ključni aspekt rada zasniva se na uspostavljanju pouzdane bežične komunikacijske veze između daljinskog upravljača te prijemnika u električnoj surf dasci korištenjem Bluetooth Classic komunikacijskog protokola. Sve programske funkcionalnosti daljinskog upravljača i prijemnika električne surf daske razvijene su u Arduino IDE programskom sučelju, a 3D model napravljen je u programu FreeCAD.

**Ključne riječi:** ESP32, Bluetooth, TFT ekran, efoil

# SUMMARY

## **Surfboard remote controller design**

This thesis focuses on the design and development of an electric surfboard remote controller using an ESP32 microcontroller, analog and digital Hall sensors, and a 1.8" TFT display. The key aspect of the work is establishing a reliable wireless communication link between the remote control and the receiver in the electric surfboard using the Bluetooth Classic communication protocol. All software functions of the remote control as well as the receiver for the electric surfboard are developed in the Arduino IDE programming interface and a 3D model was created using FreeCAD.

**Keywords:** ESP32, Bluetooth, TFT display, efoil

## 1. UVOD

U okviru studentskoga projekta AdriaFoil kojemu je cilj razvoj električne surf daske razvijen je daljinski upravljač. Električna surf daska s hidrokrilom, poznata još kao i e-foil, inovativan je sportski uređaj koji omogućava vožnju po vodenim površinama uz pomoć električnog motora.

Cilj ovoga završnoga rada je projektiranje i razvoj dizajna daljinskog upravljača i povezivanja daljinskog upravljača s električnom surf daskom stabilnim komunikacijskim protokolom. Pored toga potrebno je osigurati kontrolu brzine ESC prijemnika električne surf daske pomoću daljinskog upravljača. U radu su korištene tri faze razvoja dizajna daljinskog upravljača i interakcije s prijemnikom električne surf daske.

### **Tri faze razvoja su:**

- Elektronička arhitektura daljinskog upravljača
- Programsko rješenje daljinskog upravljača i prijemnika u električnoj surf dasci
- Dizajn kućišta daljinskog upravljača

Prilikom razvoja arhitekture daljinskog upravljača, posebna pozornost usmjerena je na odabir odgovarajućih elektroničkih komponenti čije će specifikacije biti prilagođene okolini sustava, željenom rasponu vrijednosti, stabilnosti i zadovoljavajućem utrošku energije.

Programskim rješenjem se nastojala osigurati stabilna veza daljinskog upravljača i prijemnika u električnoj surf dasci korištenjem Bluetooth Classic komunikacijskoga protokola. Posebna pozornost usmjerena je na pouzdanost komunikacije daljinskog upravljača i prijemnika električne surf daske te na brzinu prijenosa podataka.

Cilj treće faze razvoja daljinskog upravljača je dizajn funkcionalnog prototipa kućišta daljinskog upravljača. Za potrebe 3D modeliranja daljinskog upravljača korišten je FreeCAD programski alat. Ispis 3D modela daljinskog upravljača napravljen je korištenjem FDM (engl. *Fused Deposition Modeling*) tehnologijom na UltiMaker S3 printeru, a priprema 3D modela za ispis je napravljena korištenjem UltiMaker Cura softverskog alata.

Na slici 1.1. je završeni prototip daljinskog upravljača koji je izrađen u sklopu ovog rada.



Slika 1.1. Završeni prototip daljinskog upravljača



## 2. KORIŠTENE KOMPONENTE

Komponente korištene u radu u daljinskom upravljaču su:

- Razvojna pločica DOIT ESP32 DEV KIT v1
- TFT 1.8" ekran
- A3144 digitalni Hall senzor
- A49E analogni Hall senzor.

U prijemniku električne surf daske korišteni su DOIT ESP32 DEV KIT v1 i ESC (engl. Electronic Speed Controller).

U sljedećim poglavljima detaljno su objašnjena svojstva i funkcionalnosti pojedine komponente korištene u radu.

### 2.1. DOIT ESP DEV KIT v1

DOIT ESP32 DEV KIT v1 je razvojna pločica koja se temelji na ESP32-WROOM-32 mikrokontroleru. ESP32-D0WDQ6 je čip ESP32-WROOM-32 mikrokontrolera dizajniran da bude skalabilan i adaptivan [1]. ESP32-D0WDQ6 posjeduje dvije CPU jezgre od kojih je u ovom radu korištena jedna jezgra [1]. Obje jezgre imaju mogućnost zasebne kontrole. Frekvencija takta procesora ESP32-D0WDQ6 podesiva je u rasponu od 80 MHz do 240 MHz. ESP32-WROOM-32 mikrokontroler sadrži kapacitivni senzor dodira, sučelje SD kartice te omogućava različite tipove serijske komunikacije kao što su: Ethernet, SPI velike brzine, UART, I2S i I2C [1].

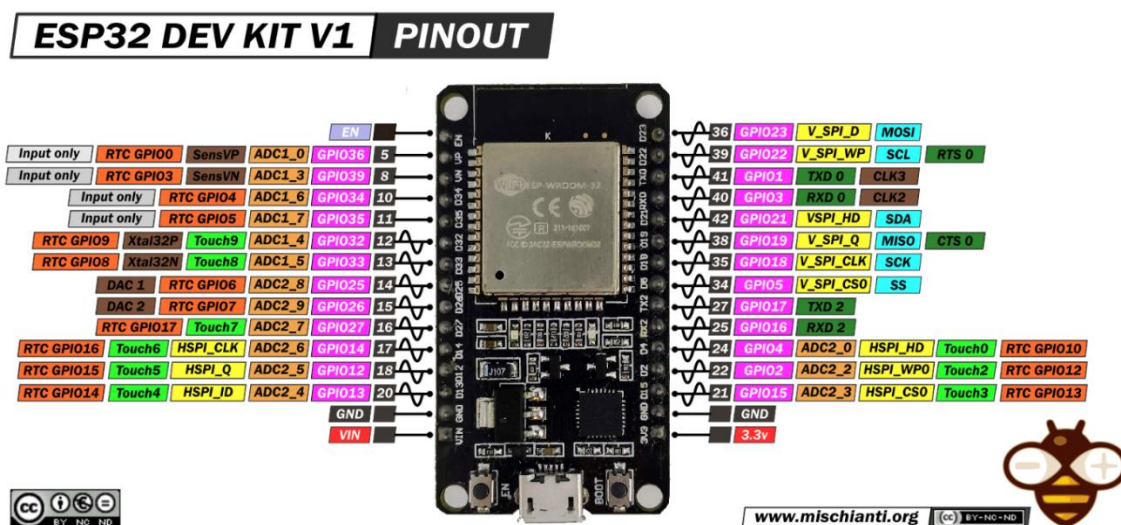
ESP32-WROOM-32 mikrokontroler posjeduje integrirani Bluetooth, BLE (engl. *Bluetooth Low Energy*) i Wi-Fi čime je omogućena njegova primjena u različitim kompleksnim aplikacijama od kojih je u radu korišten Bluetooth Classic komunikacijski protokol. Brzina prijenosa podataka podržana ESP32-WROOM-32 mikrokontrolerom je 1502 Mbit/s dok je izlazna snaga antene 20 dBm čime je osiguran najširi fizički raspon. Na temelju navedenih specifikacija,

ESP32-WROOM-32 mikrokontroler pronalazi primjene u industriji. Postoji mogućnost instalacije operativnog sustava freeRTOS. Također, moguće je bežično ažuriranje *firmwarea* kojim korisnici mogu nadograditi svoje proizvode i nakon puštanja u prodaju.

ESP32-WROOM-32 mikrokontroler uključuje:

- 18 kanala analognog-digitalnog pretvarača (ADC)
- 3 SPI sučelja
- 3 UART sučelja
- 2 I2C sučelja
- 16 kanala s pulsno-širinskim izlazom
- 2 digitalno-analognih pretvarača (DAC)
- 2 I2S sučelja
- 10 kapacitivno osjetljivih GPIO pinova

U radu je ESP32-WROOM-32 mikrokontroler korišten u sklopu razvojne pločice DOIT ESP32 DEV KIT v1 prikazanoj na slici 2.1.



Slika 2.1. ESP32 DEV KIT v1 pinout dijagram [2]

Analogno-digitalni pretvarač (engl. *Analog to Digital Converter*) i digitalno-analogni pretvarač (engl. *Digital to Analog Converter*) dodijeljeni su određenim „statističkim“ pinovima. Korisnik ima mogućnost odlučivanja dodjele UART, I2C, pulsno-širinske modulacije, SPI komunikacijskih protokola. U radu su korišteni GPIO pinovi za ADC kanal, pulsno-širinski pinovi i pinovi za SPI serijsku komunikaciju.

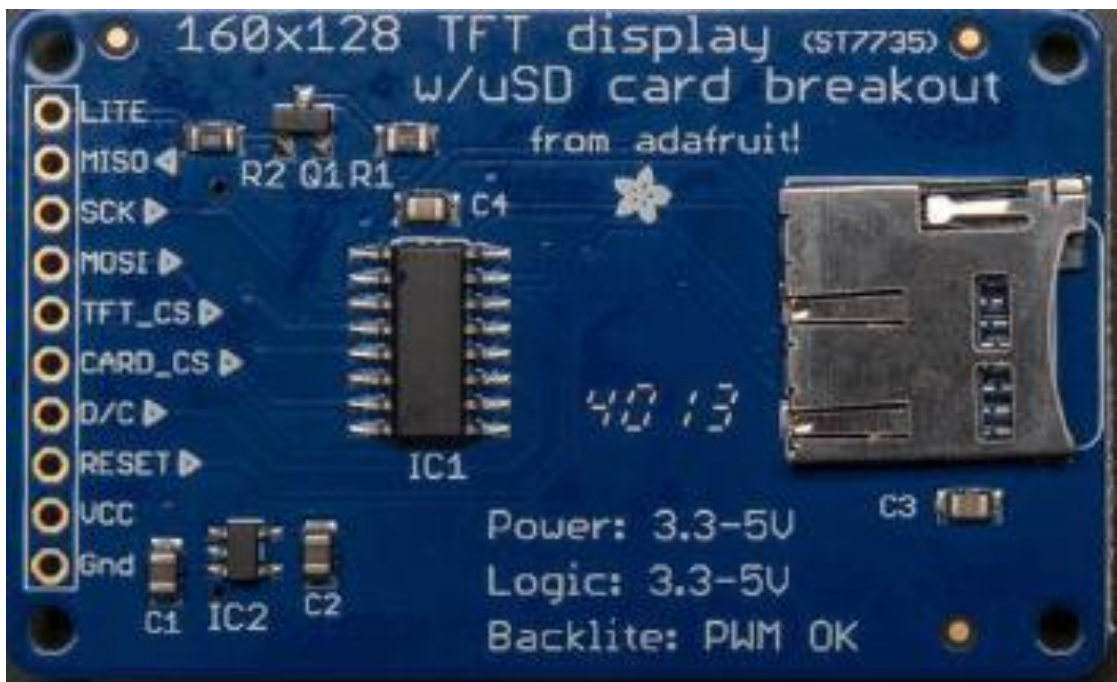
GPIO (engl. *General Input-Output Pins*) 34 do 39 mogu se koristiti samo kao ulazni pinovi. ESP32-WROOM-32 mikrokontroler ima 18x12 bitne analogno ulazne kanale koji se definirani na GPIO 32-39, GPIO 12-15, GPIO 25-27 te GPIO 0, GPIO 2, GPIO 4 pinovima [3]. 2x8 bitni DAC kanal koriste GPIO25 i GPIO26 pinove [3]. ESP32-WROOM-32 mikrokontroler ima 16 neovisnih kanala kojima se može konfigurirati pulsno-širinski modulacijski (engl. *Pulse Width Modulation*) signal različitih svojstava [3]. Samo pinovi koji se koriste za izlaz mogu konfigurirati pulsno-širinsku modulaciju [3]. Ulazni pinovi GPIO34-39 ne mogu generirati pulsno-širinsku modulaciju [3].

### **2.2. 1.8" TFT ekran**

1.8" TFT ekran posjeduje 128x160 pixela u boji te pomoću driver-a ST7735R može prikazati punu 18-bitnu boju odnosno 262144 nijanse [4]. Ekran se nalazi na razvojnoj pločici na koju je priključen pomoću fleksibilnog flat kabela. Na razvojnoj pločici nalazi se i niskopropusni regulator napona od 3,3 V, *3/5 level shifter* koji se koristi sa 3,3 V ili 5 V napajanjem i logikom te držač za microSD karticu formata FAT16/FAT32 [4]. 1.8" TFT ekran koristi SPI serijsku komunikaciju. Sadrži pixel-adresirajući okvir za pohranu čime je omogućeno korištenje na bilo kojem mikrokontroleru [4]. Na slici 2.2. i slici 2.3. je prikazan TFT ekran s prednje i stražnje strane.



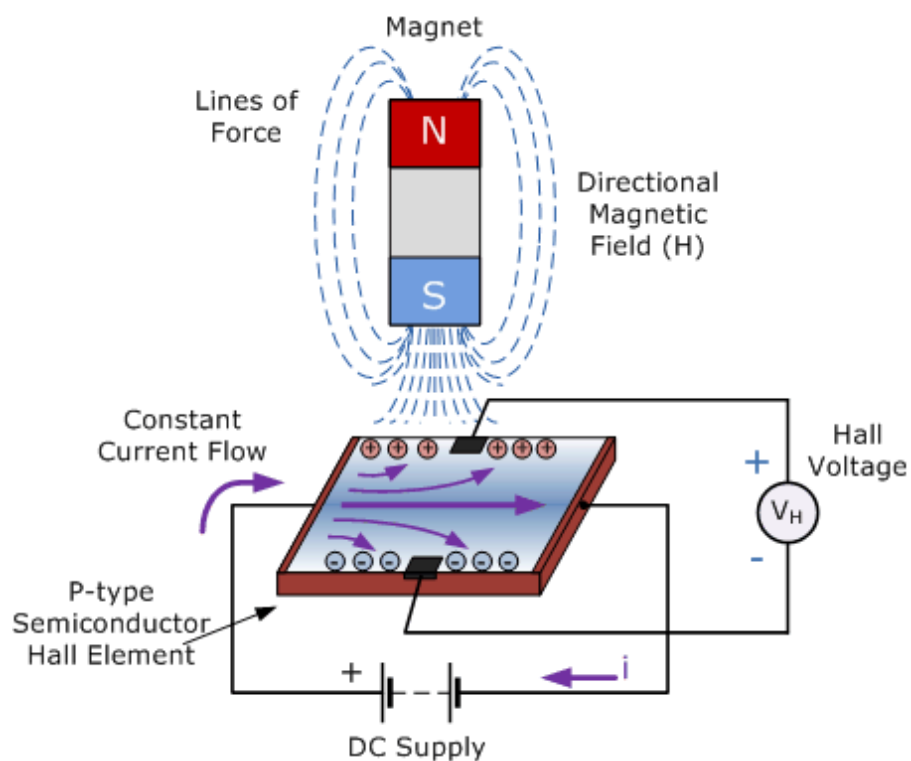
Slika 2.2. Prednji prikaz 1.8" TFT ekrana [4]



Slika 2.3. Stražnji prikaz 1.8" TFT ekrana [4]

### 2.3. Hall senzori

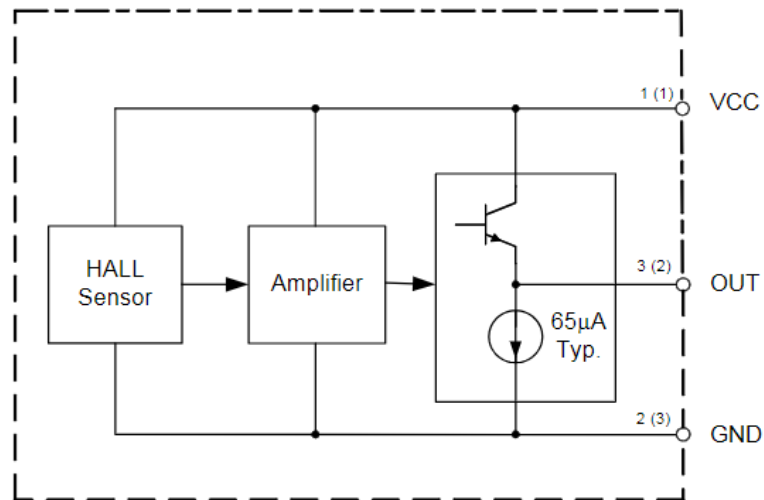
Hall senzor je elektronički uređaj koji se koristi za mjerenje magnetskog polja. To je moguće zahvaljujući Hall efektu prikazanom na slici 2.4. Hall efekt je posljedica električnog napona generiranog u vodiču kroz koji prolazi struja kada je taj vodič izložen magnetskom polju. Kada struja teče kroz vodič tada se elektroni kreću duž vodiča. Kada je vodič izložen magnetskom polju okomitom na smjer struje, elektroni će biti pod utjecajem Lorentzove sile koja će rezultirati pojavom električnog napona okomito na smjer struje i magnetskog polja. Ova pojava se naziva Hall efekt.



Slika 2.4. Način rada Hall effect senzora [5]

### 2.3.1. Analogni A49E Hall senzor

Analogni A49E Hall senzor je linearni senzor koji se aktivira pod utjecajem magnetskog polja kojeg stvara trajni magnet ili elektromagnet. Izlazni napon određen je naponom napajanja te jačinom magnetskoga polja. Zahvaljujući integriranoj arhitekturi senzora eliminiran je jak šum čime je izbjegnuto korištenje vanjskog filtriranja. Preciznim otpornicima postignuta je povećana temperaturna stabilnost i preciznost. Radni temperaturni raspon Hall A49E senzora je od  $-40^{\circ}\text{C}$  do  $85^{\circ}\text{C}$  što ga čini pogodnim za korištenje u komercijalnim, potrošačkim i industrijskim aplikacijama. Hall A49E senzor korišten u radu nalazi se u TO-92S kućištu [6]. Funkcijski blok dijagram Hall effect senzora A49E je prikazan na slici 2.5.



Slika 2.5. Funkcijski blok dijagram AH49E [6]

Osnovne značajke [6]:

- minijaturna izvedba
- potrošnja od 3.5 mA na 5 V
- jedan strujni izlaz
- linearan izlaz za fleksibilnost u dizajnu strujnog kruga
- nizak izlazni šum eliminira potrebu za filtriranjem
- temperaturni raspon od -40 °C do 85 °C
- reagira na pozitivni ili negativni Gaussov utjecaj

Moguće aplikacije [6]:

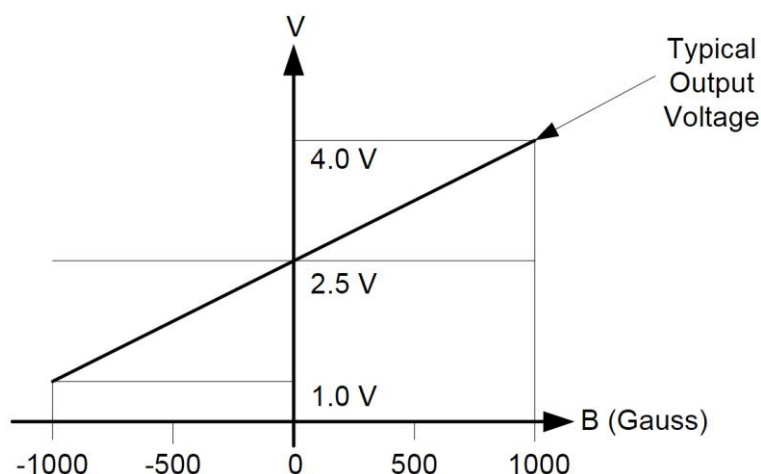
- senzor trenutne struje
- kontrola motora
- senzor pozicije
- čitanje magnetskih kodova
- detektor feromagnetskih metala
- senzor vibracija
- senzor razine tekućine
- senzor težine

U tablici 2.1. su prikazane osnovne karakteristike Hall senzora A49E.

Tablica 2.1. Karakteristike Hall senzora AH49E [6]

Parametar	Simbol	Vrijednost	Mjerna jedinica
Napon napajanja	$V_{CC}$	3.0...6.5	V
Struja izvora	$I_O$	10	mA
Radna temperatura	$T_A$	-40...+85	°C

U odsutnosti vanjskoga magnetskoga polja ( $B = 0$  T), stacionarni izlazni napon iznosi pola napona napajanja. Ako se južnim magnetnim polom prilazi prednjoj strani (strana s identifikacijskom oznakom) Hallovog povećat će se izlazni napon kruga, a ako se prilazi sjevernim magnetskim polom prednjoj strani Hallovog senzora smanjit će se izlazni napon kruga. Varijacije u razini napona su simetrične što se može iščitati iz prijenosne karakteristike prikazane na slici 2.6. Najveća magnetska osjetljivost postiže se pri napajanju od 6 V te posljedično tome nastaje povećana potrošnja i blagi gubitak simetrije izlaznoga signala. Takvi uvjeti nisu zadovoljavajući i preporučeni za korištenje osim ako je bitna veličina izlaznog signala. Izlazni signal može se povećati kapacitivnim spajanjem na dodatno pojačalo, osobito ako je prisutna velika promjena frekvencije magnetskoga polja.



Slika 2.6. Prijenosna karakteristika AH49E [6]

### 2.3.2. Digitalni A3144 Hall senzor

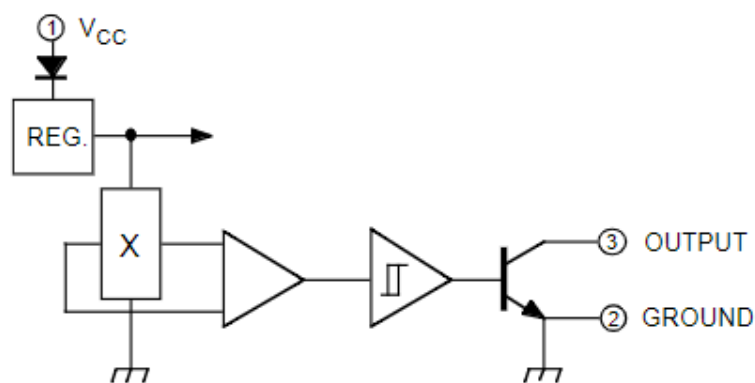
Digitalni A3144 Hall senzor je monolitni integrirani krug dizajniran za kontinuirani rad s ekstremnim temperaturama do  $+150$  °C te ga karakterizira stabilnost u uvjetima promjene temperature i napona napajanja [7]. Senzor se sastoji od regulatora napona za rad s napajanjem od 4,5 V do 24 V, diode za zaštitu od reverzibilne baterije, kvadratnoga generatora Hallovog-



napona, kompenzacijske sklopovske temperature, pojačala za male signale, Schmittova okidača i otvoreno-kolektorskog izlaza do 25 mA [7]. Karakteristike hall senzora A3144 prikazane su u tablici 2.2., a na slici 2.7. je prikazan funkcijski blok dijagram.

Tablica 2.2. Karakteristike Hall senzora A3144 [7]

Apsolutne maksimalne vrijednosti na temperaturi +25 °C		Mjerna jedinica
Napon izvora [V <sub>CC</sub> ]	28	V
Gustoća magnetskog toka [B]	∞	T
Konstantna struja izlaza [I <sub>OUT</sub> ]	25	mA
Radni temperaturni opseg [T <sub>A</sub> ]	-40...+85	°C



Slika 2.7. Funkcijski blok dijagram A3144 [7]

Značajke i prednosti [7]:

- izvanredna stabilnost temperature za automobilsku ili industrijsku primjenu
- rad na napajanju od 4,5 V do 24 V ... potrebno je samo neregulirano napajanje
- otvoreno-kolektorski izlaz 25 mA ... Kompatibilan s digitalnom logikom
- zaštita od reverzibilne baterije
- aktivacija s malim, komercijalno dostupnim permanentnim magnetima
- malo kućište
- otpornost na stres

Izlaz senzora pada na nulu kada magnetsko polje Hallovog senzora premaši prag za uključivanje, a kada magnetsko polje padne ispod praga za isključivanje izlazni napon prelazi u visoko stanje.

Djelovanje Hall A3144 senzora zasniva se na detekciji magnetskoga polja i promjeni izlaza Hall senzora. Ako magnetsko polje postigne ili premaši prag za uključivanje onda izlaz senzora naglo prelazi u nisku razinu. Ovo izlazno stanje se često koristi za označavanje prisutnosti magnetskog objekta. Ako magnetsko polje padne ispod praga za isključivanje, senzor će odgovoriti prebacivanjem izlaznog signala u visoku razinu. Ovaj signal obično ukazuje na to da je magnetski objekt udaljen ili da promjene u magnetskom polju više nisu prisutne.

### 3. KOMUNIKACIJSKI SUSTAV

Komunikacija sustava temelji se na jednosmjernoj bežičnoj komunikaciji između daljinskog upravljača ka prijemniku u električnoj surf dasci. Bežično se samo prenose vrijednosti od 0 do 255 s daljinskog upravljača ka prijemniku u električnoj surf dasci koje indirektno upravljaju brzinom daske.

Da bi se uspostavila zadovoljavajuća komunikacija između pošiljatelja i primatelja sustava, trebaju biti zadovoljeni faktori: daljina (snaga), brzina komunikacije, stabilnost veze i mala potrošnja energije. Daljina između predajnika i prijemnika posljedično uvjetuje potrošnju energije sustava, stoga je bitno postići zadovoljavajuću udaljenost kako bi se postigao što bolji proporcionalni odnos uložene snage i potrošnje energije.

Da bi se osigurala neometana komunikacija, bitno je osigurati zadovoljavajuću brzinu komunikacije, odnosno brzinu prijenosa podataka prijemnika ka predajniku čime se izbjegava pojava latencije te se tako postiže brže vrijeme odziva i sinkroniziranost sustava. Također je i stabilnost sustava bitan faktor u postizanju neometane komunikacije, a osigurava se prijenosom podataka bez zapreka, odabirom odgovarajućeg frekvencijskoga područja i implementiranjem pravodobne sigurnosti sustava. Da bi se zadovoljili navedeni faktori, u kreiranju sustava izabran je Bluetooth classic komunikacijski protokol verzije Bluetooth v4.2.BR/EDR (engl. *Basic Rate/Enhanced Data Rate*). Kao moguća razmatrana rješenja, u realizaciji komunikacije sustava su i Wi-Fi, BLE i nRF24L01.

Prednosti Wi-Fi komunikacije u odnosu na korišteni Bluetooth v4.2 su brži prijenos podataka, veća udaljenost prijenosa podataka i bolja sigurnost. Kao posljedicu Wi-Fi ima mnogo veću potrošnju energije u odnosu na Bluetooth v4.2[8].

Komunikacija s nRF24L01 modulom obuhvaća 125 različitih kanala gdje svaki kanal zauzima frekvencijsku širinu nešto manju od 1 MHz dok Bluetooth v4.2 obuhvaća 79 kanala širine 1 MHz [9]. Korištenje većeg broja kanala omogućava bolju otpornost na smetnje i interferencije, smanjenje odziva i optimizaciju energetske potrošnje. Komunikacija korištenjem nRF24L01 modula omogućava korištenje *Enhanced ShockBurst* protokola koji omogućava automatsko

rukovanje paketima (engl. *Automatic Packet Handling*) [9]. Ovim protokolom pojednostavljen je proces programiranja tako da je korisnik neobvezan voditi računa o primljenim paketima, potrebnoj retransmisiji ili slanju potvrde natrag pošiljatelju. Za korištenje nRF24L01 potrebna je dodatna elektronika što čini arhitekturu sustava složenijom, što može dovesti do nestabilnosti komunikacije u sustavu. nRF24L01 se može uzeti u razmatranje u sljedećim iteracijama temeljem značajka koje su zadovoljile uvjete komunikacije ovoga sustava.

### **3.1. Bluetooth komunikacija**

Bluetooth je bežični standard kratkoga dometa čija je uloga razmjena podataka između uređaja na kratkim udaljenostima. Bluetooth standardom ostvaruje se PAN mreža (engl. *Personal Area Network*) kojim se međusobno povezuju elektronički uređaji unutar radnoga prostora. Najčešće korišten način rada koristi snagu prijenosa od 2,5 mW čime se postiže mali domet do 10 m [10]. Bluetooth tehnologija koristi UHV (engl. *Ultra High Frequencies*) radio valove u ISM opsegu (engl. *Industrial, Science, Medical range*) od 2,408 GHz do 2,480 GHz [10]. Bluetooth protokol je obuhvaćen standardom IEE 802.15.1.

Standard Bluetooth v4.2 uključuje dvije varijante Bluetooth komunikacijskoga protokola: BLE i Bluetooth Classic [11]. U radu je korišten Bluetooth v4.2.BR/EDR komunikacijski protokol standarda Bluetooth v4.2 u kojem su postignute manje nadogradnje i poboljšanja u odnosu na Bluetooth 4.0 standard uključujući poboljšanu sigurnost i stabilnost veze. Bluetooth v4.2.BR/EDR komunikacijski protokol naziva se još i Bluetooth Classic protokol [10].

#### **3.1.1. Bluetooth Low Energy**

*Bluetooth Low Energy* - BLE je bežični komunikacijski protokol standardiziran kao Bluetooth 4.0. LE neovisan je i nekompatibilan s Bluetooth Classic komunikacijskim protokolom. Bluetooth LE i Bluetooth Classic komunikacijski protokol koegzistiraju na većini uređaja [10].

Mikrokontroler ESP-32-WROOM podržava Bluetooth v4.2.BR/EDR komunikacijski protokol i BLE [11].

Glavna prednost korištenja ovakve vrste komunikacijskog protokola, kojeg karakterizira niska potrošnja energije sustava od 0.01 W do 0.5 W, ostvarena je putem minimiziranja radio veze i slanja manjih količina podataka pri niskim brzinama prijenosa od 125 kbit/s, 500 kbit/s, 1 Mbit/s i 2 Mbit/s. Osim toga, važno je napomenuti da su Bluetooth LE čipovi i moduli manji i ekonomičniji od standardnih Bluetooth čipova, što smanjuje troškove i omogućava kreiranje kompaktnih uređaja. Velika prednost Bluetooth LE komunikacije ogleda se i u otvorenom pristupu dokumentaciji koja je dostupna na službenoj web-stranici.

Bluetooth LE nije korišten u realizaciji komunikacije sustava prvenstveno zbog nižeg kapaciteta prijenosa podataka u odnosu na Bluetooth Classic komunikaciju [10]. Vrijeme prijenosa podataka potrebnome Bluetooth LE je 3 ms dok za Bluetooth Classic iznosi 0.625 ms čime se Bluetooth Classic pokazao boljim rješenjem za vrstu sustava realiziranu u projektu. Vrijednost dometa kao bitnoga faktora u realizaciji sustava bila je od presudne važnosti kod odabira Bluetooth Classic koji omogućava veći domet od BLE [10]. Usporedba glavnih značajki ova dva protokola je prikazana u tablici 3.1.

Tablica 3.1. Usporedba Bluetooth Classic i Bluetooth LE protokola [10]

	<b>Bluetooth Classic</b>	<b>Bluetooth Low Energy</b>
<b>Frekvencijski pojas</b>	2.4 GHz ISM pojas (2.402 – 2.480 GHz)	2.4 GHz ISM pojas (2.402 – 2.480 GHz)
<b>Kanali</b>	79 kanala a razmakom od 1 MHz	40 kanala a razmakom od 2 MHz
<b>Modulacije</b>	GFSK, $\pi/4$ DQPSK, 8DPSK	GFSK
<b>Brzina prijenosa podataka</b>	1-3 Mbit/s	125 kbit/s, 500 kbit/s, 1 Mbit/s, 2 Mbit/s
<b>Komunikacijske topologije</b>	Point-to-Point (uključujući piconet)	Point-to-Point (uključujući piconet) Broadcast Mesh
<b>Minimalno vrijeme potrebno za slanje podataka</b>	0.625 ms	3 ms
<b>Potrošnja energije</b>	1 W	0.01 - 0.50 W
<b>Maksimalna potrošnja struje</b>	<30 mA	<15 mA

### 3.1.2. Bluetooth Classic

Bluetooth Classic ili Bluetooth v4.2.BR/EDR je bežični komunikacijski protokol koji se koristi za kratkodometnu komunikaciju između uređaja kao što su pametni telefoni, slušalice, tipkovnice i drugi. Bluetooth ima različite načine rada (engl. *mode of operation*) koji određuju kako se uređaji međusobno povezuju i komuniciraju [10]. Glavne specifikacije Bluetooth Classic protokola prikazane su u tablici 3.2.

Tablica 3.2. Karakteristike Bluetooth Classic protokola [10]

<b>Bluetooth Classic</b>	
<b>Frekvencijski pojas</b>	2.4 GHz ISM pojasa (2.402...2.480 GHz)
<b>Kanali</b>	79 kanala s razmakom od 1 MHz
<b>Uporaba kanala</b>	FHSS (engl. <i>Frequency-Hopping Spread Spectrum</i> )
<b>Modulacija</b>	GFSK, $\pi/4$ DQPSK, 8DPSK
<b>Brzina prijenosa podataka</b>	GFSK: 1 Mbit/s $\pi/4$ DQPSK: 2 Mbit/s 8DPSK: 3 Mbit/s
<b>Tx Power (engl. <i>Transsmition Power</i>)</b>	$\leq 100$ mW (+20 dBm)
<b>Rx Sensitivity (engl. <i>Receiver Sensitivity</i>)</b>	$\leq -70$ dBm
<b>Komunikacijske topologije</b>	Point-to-point (uključujući piconet)

Dva osnovna načina rada Bluetooth Classic komunikacije su *Basic Rate* (BR) i *Enhanced Data Rate* (EDR).

*Basic Rate* [12]:

- **Brzina prijenosa podataka:** BR mod podržava brzinu prijenosa od 1 megabit po sekundi (1 Mbit/s).
- **Modulacija:** Koristi se modulacija zvana *Gaussian Frequency Shift Keying* (GFSK). GFSK je tehnika modulacije koja se koristi za prenos digitalnih podataka putem analognih signala na način koji je otporan na smetnje.
- **Primjena:** BR mod se obično koristi za osnovne aplikacije Bluetooth komunikacije, kao što su glasovni pozivi, prijenos malih datoteka i jednostavne komunikacije između uređaja.

*Enhanced Data Rate* [12]:

- **Brzina prijenosa podataka:** EDR mod podržava veće brzine prijenosa podataka. Koristi dva različita načina modulacije:
- **$\pi/4$ -Differential Quadrature Phase Shift Keying ( $\pi/4$ -DQPSK):** Omogućava brzinu prijenosa od 2 megabita po sekundi (2 Mbit/s).
- **8-Differential Phase Shift Keying (8DPSK):** Omogućava brzinu prijenosa od 3 megabita po sekundi (3 Mbit/s).
- **Modulacija:** Ovi brži načini prijenosa koriste naprednije tehnike modulacije kako bi se postigle veće brzine prijenosa podataka.
- **Primjena:** EDR mod se koristi za zahtjevnije aplikacije Bluetooth komunikacije, kao što su prijenos visokokvalitetnog zvuka ili videozapisa, kao i za brži prijenos datoteka.

Snaga odašiljača Bluetooth uređaja podijeljena je u tri razreda [12]:

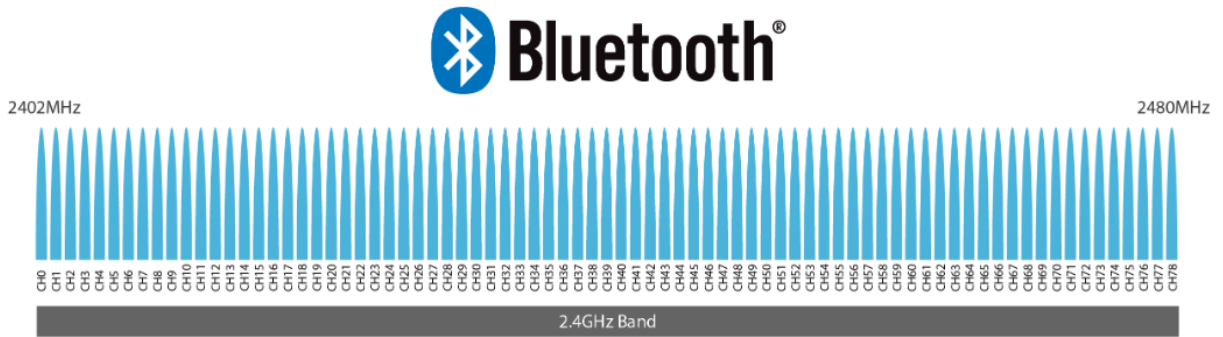
- Razred 1: +20 dBm
- Razred 2: +4 dBm
- Razred 3: +0 dBm

Dometa Bluetooth Classic komunikacijskoga protokola iznosi oko 100 m u idealnim uvjetima. U stvarnom okruženju je dometa mnogo kraći i ovisi o snazi odašiljača, dizajnu, korištenoj modulaciji i okolini.

Osnovna brzina prijenosa podataka je standardna BR radijska modulacija s 1 Mbit/s bežičnim prijenosom. Unaprijeđena brzina prijenosa podataka EDR uvedena je kako bi omogućile brzine od 2 Mbit/s potrebne za prijenos zvuka, videa i velikih količina podataka.

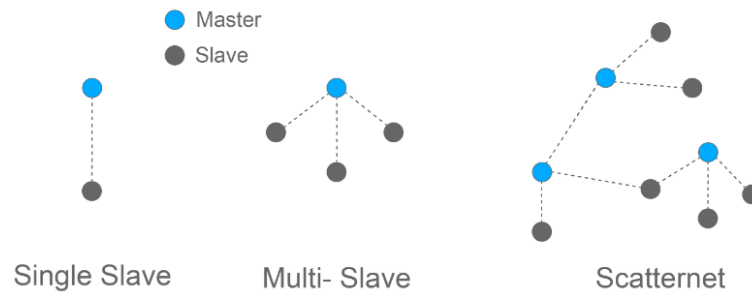
Frekvencijski opseg Bluetooth Classic od 2,4 GHz podijeljen je na 79 kanala, a svaki je kanal odijeljen za 1 MHz te su označeni brojevima od 0 do 78 što se može vidjeti na slici 3.1. Broj kanala i algoritam prilagodljivog frekvencijskog preskakanja pridonose sprječavanju smetnji uzrokovanih različitim izvorima.





Slika 3.1. Podjela na kanale frekvencijskog pojasa Bluetooth Classic-a [12]

Bluetooth Classic podržava tri oblika topologije: *single slave*, *multi-slave* i *scatternet* prikazane na slici 3.2.



Slika 3.2. Tri topologije Bluetooth Classic-a [12]

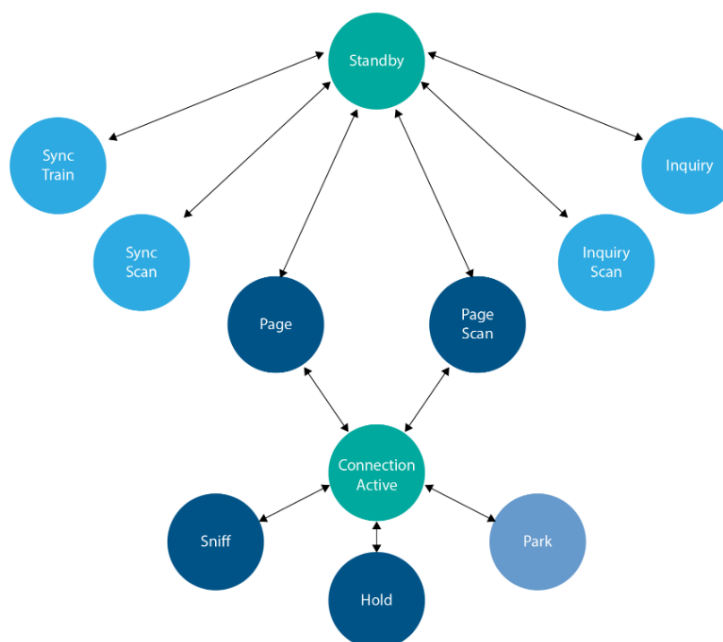
Koraci u uspostavljanju komunikacije između *master* i *slave* uređaja Bluetooth Classic komunikacije su [12]:

- **Standby:** Ovo je početno stanje u kojem su Bluetooth uređaji neaktivni i štede energiju.
- **Inquiry:** *Master* uređaj traži dostupne *slave* uređaje u blizini putem procesa nazvanog "inquiry". *Master* šalje zahtjev za informacijama o dostupnim uređajima u blizini.
- **Inquiry Scan:** *Slave* uređaji odgovaraju na "inquiry" zahtjeve tako što šalju informacije o sebi, kao što su adresa i usluge koje nude.
- **Page:** *Master* uređaj odabire određeni *slave* uređaj i započinje proces "page" kako bi uspostavio direktnu vezu sa *slave* uređajem. Ovaj korak uključuje razmjenu informacija za uspostavljanje veze.
- **Page Scan:** *Slave* uređaj prihvata "page" zahtjev od *master* uređaja i uspostavlja vezu.
- **Synchronization Train:** Ovaj korak se koristi za postizanje preciznog vremenskog usklađivanja između uređaja kako bi se osigurala točna komunikacija.
- **Synchronization Scan:** *Slave* uređaj prihvata zahtjev za sinkronizacijom i uspostavlja sinkroniziranu vezu.

Stanja *slave* uređaja u Bluetooth Classic komunikaciji su [12]:

- **Aktivan (engl. Active):** Ovo je osnovni način rada kada je uređaj aktivno uključen i spreman za komunikaciju. U ovom *mode-u* uređaj može primiti i slati podatke.
- **Mirovanje (engl. Park):** Stanje mirovanja način rada u kojem se uređaj privremeno isključuje iz aktivne komunikacije kako bi štedio energiju. Uređaj ostaje povezan, ali ne komunicira aktivno. Ovo se često koristi u situacijama gdje je potrebno smanjiti potrošnju energije.
- **Održavanje (engl. Hold):** Mod održavanja omogućava uređaju da privremeno zaustavi komunikaciju bez potrebe za ponovnim uspostavljanjem veze. To je korisno za privremeno zaustavljanje komunikacije, ali zadržavanje veze.
- **Praćenje (engl. Sniff):** Uređaj može prijeći u režim praćenja kako bi smanjio potrošnju energije dok je još uvijek povezan. U ovom *mode-u* uređaj povremeno provjerava ima li podataka za prijenos, što omogućava uštedu energije.

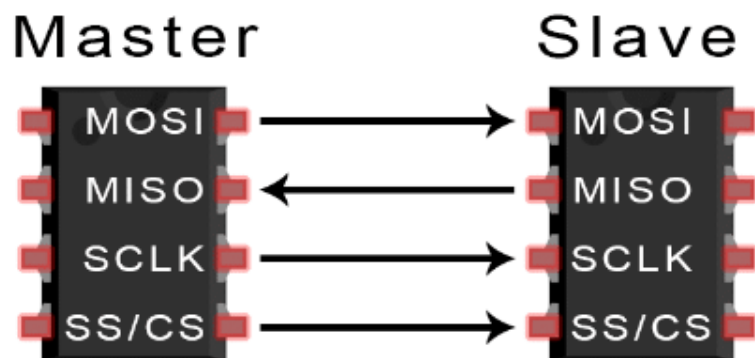
Dijagram stanja i potrebnih koraka je prikazan na slici 3.3.



Slika 3.3. Dijagram stanja i koraka uspostavljanja Bluetooth Classic veze [12]

### 3.2. SPI serijska komunikacija

Serijsko periferno sučelje (engl. *Serial Peripheral Interface*) odnosno SPI je komunikacijski protokol koji koristi četiri linije za komunikaciju između *master* uređaja i jednog ili više *slave* uređaja [13]. Četiri linije koje koristi su MOSI (engl. *Master Out Slave In*), MISO (engl. *Master In Slave Out*), SCLK (engl. *Serial Clock*) i SS (engl. *Slave Select*) [13] koje su prikazane na slici 3.4. Podržava brzinu prijenosa podataka do 10 Mbit/s [13]. SPI komunikacija je sinkrona te *master* određuje takt sabirnice između sebe i jednog ili više *slave* uređaja.

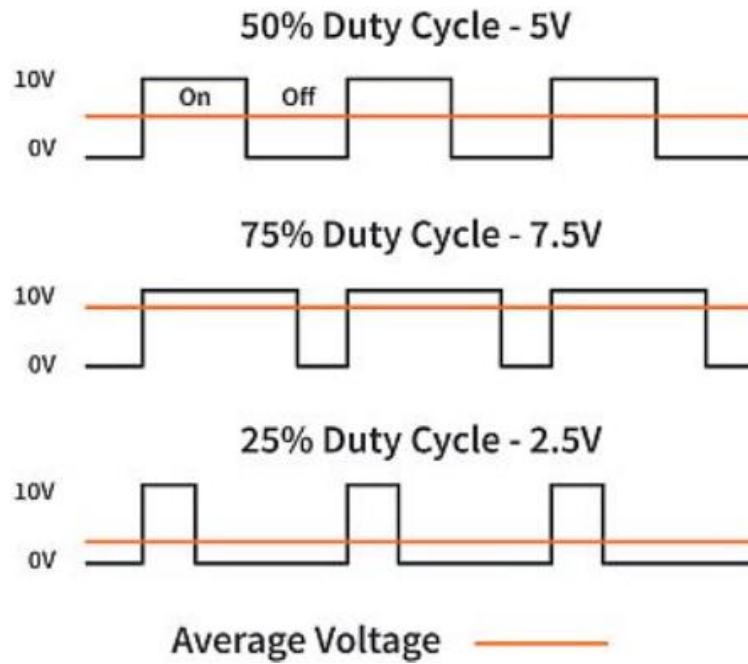


Slika 3.4. SPI *master - slave* komunikacija [13]

### 3.3. Pulsno-širinska modulacija

Pulsno-širinska modulacija ili PWM je metoda kojom se pomoću digitalnog signala upravlja analognim uređajima. Digitalnom signalu fiksne amplitude mijenja se omjer između vremena uključenosti i isključenosti tako da se srednja vrijednost napona i struje koja se upućuje trošilu kontrolira prekidačem koji vrši brzu izmjenu između 0 % i 100 % [14] kao što je vidljivo na slici 3.5. Veća će snaga biti isporučena trošilu što je prekidač više uključen. Glavna prednost pulsno-širinske modulacije temelji se na činjenici da je gubitak snage u prekidačima vrlo nizak. Ako je prekidač isključen praktički nema struje, a kada je prekidač uključen i snaga se prenosi na trošilu može se reći da gotovo nema pada napona na prekidaču. U oba slučaja gubitak snage je blizu nule.

Visoka logička razina predstavlja „vrijeme uključenosti“ (engl. *on-time*), a za njegovu prezentaciju koristimo koncept radnoga ciklusa (engl. *duty cycle*) [14]. Radni ciklus opisuje omjer vremena tijekom kojeg je digitalni signal uključen unutar određenoga vremenskoga intervala.



Slika 3.5. Tri različita radna ciklusa pulsno-širinske modulacije [14]

Radni ciklus određen je kao:

$$D = \frac{T_{ON}}{Period} * 100 \quad (2.1)$$

gdje je:

D – radni ciklus izražen u postocima

$T_{ON}$  – vrijeme trajanja signala u stanju „ON“

Period – ukupno vrijeme trajanja ciklusa ( $T_{ON}+T_{OFF}$ )

Srednja vrijednost napona signala određena je kao:

$$V_{avg} = \frac{D}{100} * V_{MAX} \quad (2.2)$$

gdje je:

$V_{avg}$  – srednja vrijednost napona signala

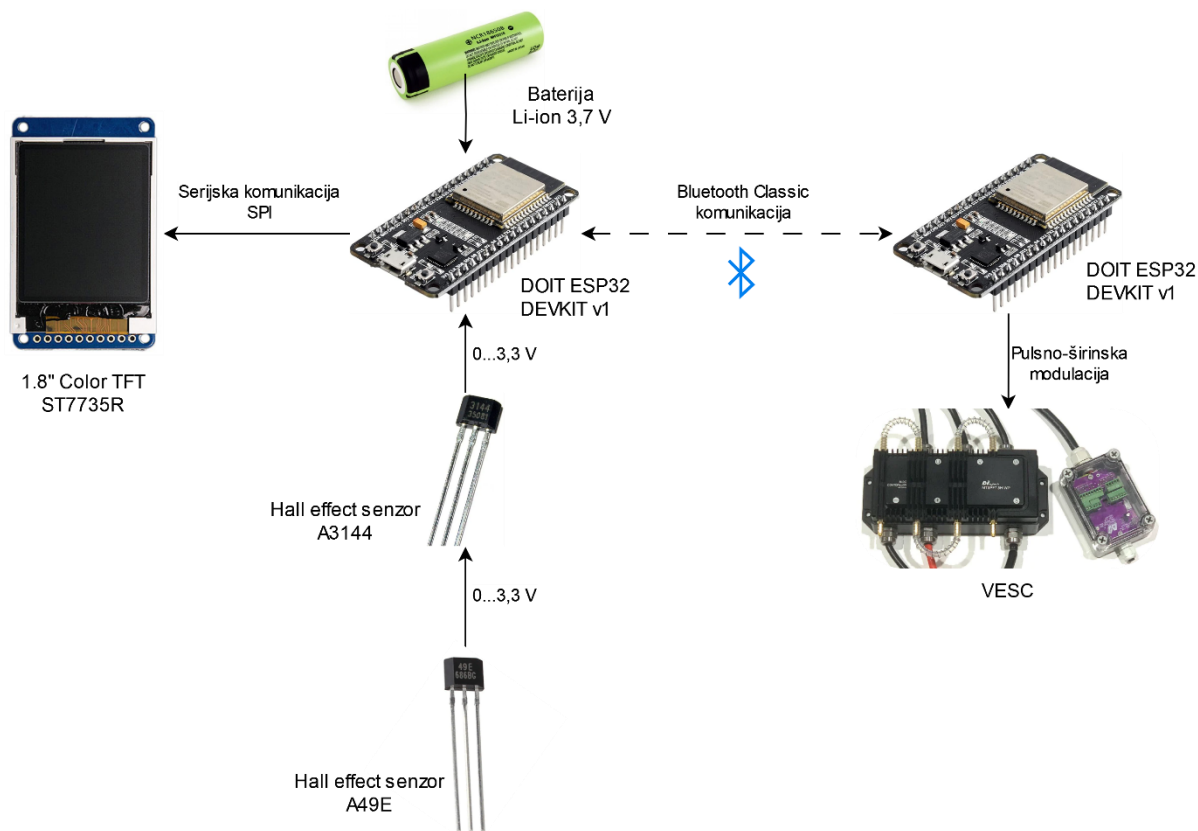
$D$  – radni ciklus u postocima

$V_{MAX}$  – maksimalni napon signala

Pulsno-širinska modulacija u radu je primijenjena u kontroli motora pomoću elektroničkog kontrolera brzine (engl. *Electronic Speed Controller*), tj. ESC-a. ESC očitava srednju vrijednost napona na ulazu koja se stvara kao posljedica pulsno-širinske modulacije na izlazu iz ESP32.

## 4. DIZAJN SUSTAVA

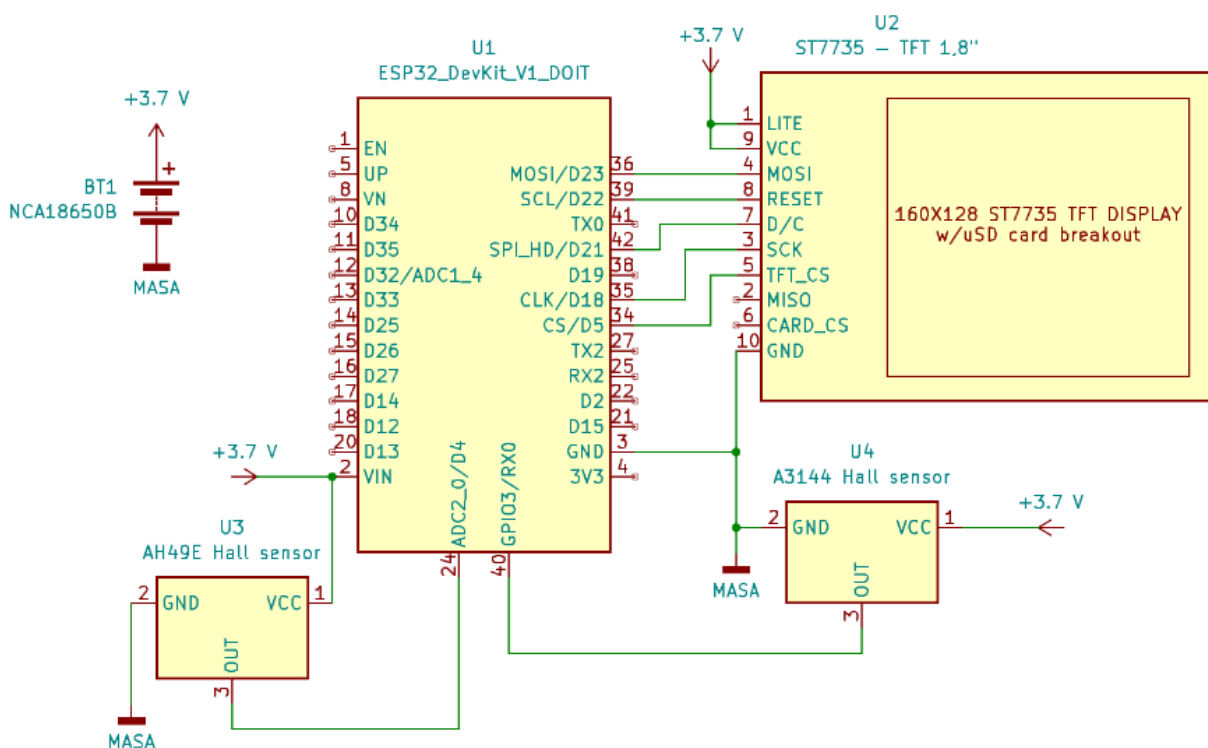
U ovom poglavlju obrađena je arhitektura sveobuhvatnoga sustava koja uključuje daljinski upravljač i prijemnik električne surf daske. Detaljno su objašnjeni razlozi korištenja pojedine komponente u arhitekturi sustava uz odgovarajuće shematske prikaze. Sve scheme su izrađene korištenjem programskog alata KiCAD. Na slici 4.1. je prikazana blok shema sustava.



Slika 4.1. Blok shema sustava

## 4.1. Daljinski upravljač

Daljinski upravljač ima ulogu pošiljatelja koji slanjem analognih vrijednosti analognog Hallovog senzora A49E upravlja radom ESC-a na strani prijemnika električne surf daske. Arhitektura daljinskog upravljača je jednostavna te ju čine razvojna pločica DOIT ESP32 DEV KIT v1, Hallov senzor A49E, Hallov senzor A3144 i TFT 1.8" ekran. Detaljna elektronička shema izrađena u KiCAD programu je prikazana na slici 4.2.



Slika 4.2. Elektronička shema daljinskog upravljača

ESP32 mikrokontroler upravlja funkcionalnošću komponenti daljinskog upravljača. Prednost njegova korištenja očituje se u mogućnosti korištenja WiFi i Bluetooth komunikacije koje su podržane na ovom mikrorračunalu. Važno je napomenuti da su prednosti ESP32 mikrokontrolera, zbog kojih predstavlja idealno rješenje u arhitekturi daljinskog upravljača,



snažan procesor radne frekvencije od 80 MHz do 240 MHz, veliki broj GPIO pinova, mogućnost programiranja u različitim programskim sučeljima kao što su Arduino IDE, MicroPython i drugi.

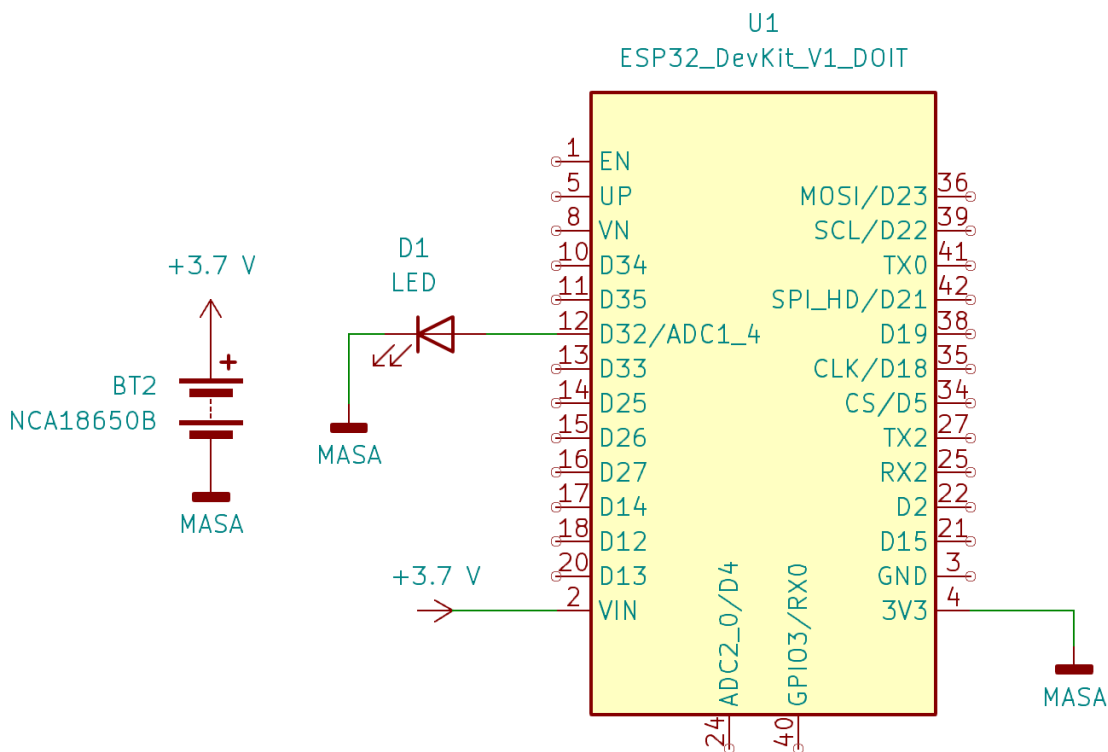
U radu analogni Hall senzor A49E ima ulogu potencijometra. Zadaća Hall A49E senzora je očitavanje magnetskog polja te slanje analognih vrijednosti ka ESP32 mikrokontroleru koji ih dalje obrađuje. Analogne vrijednosti Hall A49E senzora očitavaju se u prisutnosti magnetskog polja tako da senzor reagira na prisutnost sjevernog i južnog magnetskog pola. Pogodnost korištenja Hallovog senzora A49E u arhitekturi daljinskog upravljača u odnosu na obični potencijometar je svojstvo beskontaktnosti čime je omogućena veća fleksibilnost daljinskog upravljača. Za razliku od potencijometra Hall A49E senzor se može konfigurirati da radi na manjem kutu, ali u punom rasponu vrijednosti. Također prednost u odnosu na potencijometar je i vodootpornost budući da Hall A49E senzor ne mora imati fizički kontakt s magnetom.

Digitalni Hall senzor A3144 omogućuje sigurno upravljanje daljinskim upravljačem kako bi se spriječile sve slučajne radnje i zaustavili svi opasni događaji. Hall A3144 senzor reagira na prisutnost i odsutnost magnetskoga polja pri čemu daje digitalni izlaz 0 ili 1. Prilikom odsutnosti magnetskog polja Hall A3144 senzor automatski prekida rad daljinskog upravljača pri tome proslijeđujući vrijednost 0 prijemniku prema električnoj surf dasci čime se zaustavlja rad ESC-a. Ponovnom aktivacijom Hall A3144 senzora daljinski upravljač nastavlja slati odgovarajuće vrijednosti. Na temelju svojstava vodootpornosti, beskontaktnosti i velike brzine osjetljivosti, Hall A3144 senzor predstavlja najbolje rješenje u izboru prekidača za navedeno elektroničko rješenje daljinskog upravljača.

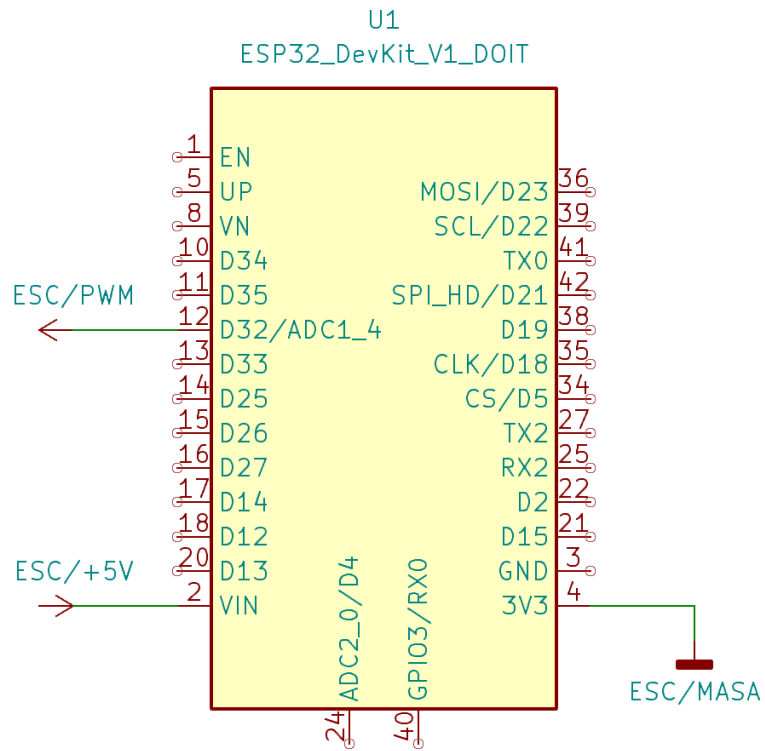
TFT 1.8" ekran korišten je za prikaz brojevnih vrijednosti koje se šalju Bluetooth komunikacijskim protokolom prijemniku električne surf daske. Na ovaj način korisniku je omogućen vizualni pregled brzine upravljanja električnom daskom pomoću daljinskog upravljača. TFT LCD 1.8" ima 128x160 piksela u boji. Ekranom upravlja chip ST7735R TFT s kojim se može prikazati puna 18-bitna boja odnosno 262144 nijanse boja. Veličina zaslona TFT 1.8" ekrana i mogućnost korištenja SPI serijske komunikacije čija je brzina prijenosa podataka u opsegu od 10 do 20 Mbit/s bile su ključne za odabir odgovarajućeg ekrana.

## 4.2. Prijemnik u električnoj surf dasci

Arhitektura prijemnika u električnoj surf dasci realizirana je u dva koraka. Prvo je sastavljen prototip sa svjetlećom diodom na DAC izlazu, a u izvedenom projektu je DAC spojen na ESC. Prototipna verzija se sastoji od ESP32 mikrokontrolera i svjetleće diode, a svrha ovog modela je dokaz koncepta rada sustava. Izvedeni projekt je *de facto* isti prototipu, samo je umjesto svjetleće diode ESP32 spojen na ESC. Na slici 4.3. je prikazana shema testnog modela prijemnika, a na slici 4.4. je prikazana shema prijemnika spojenog na ESC.



Slika 4.3. Shema prototipa prijemnika



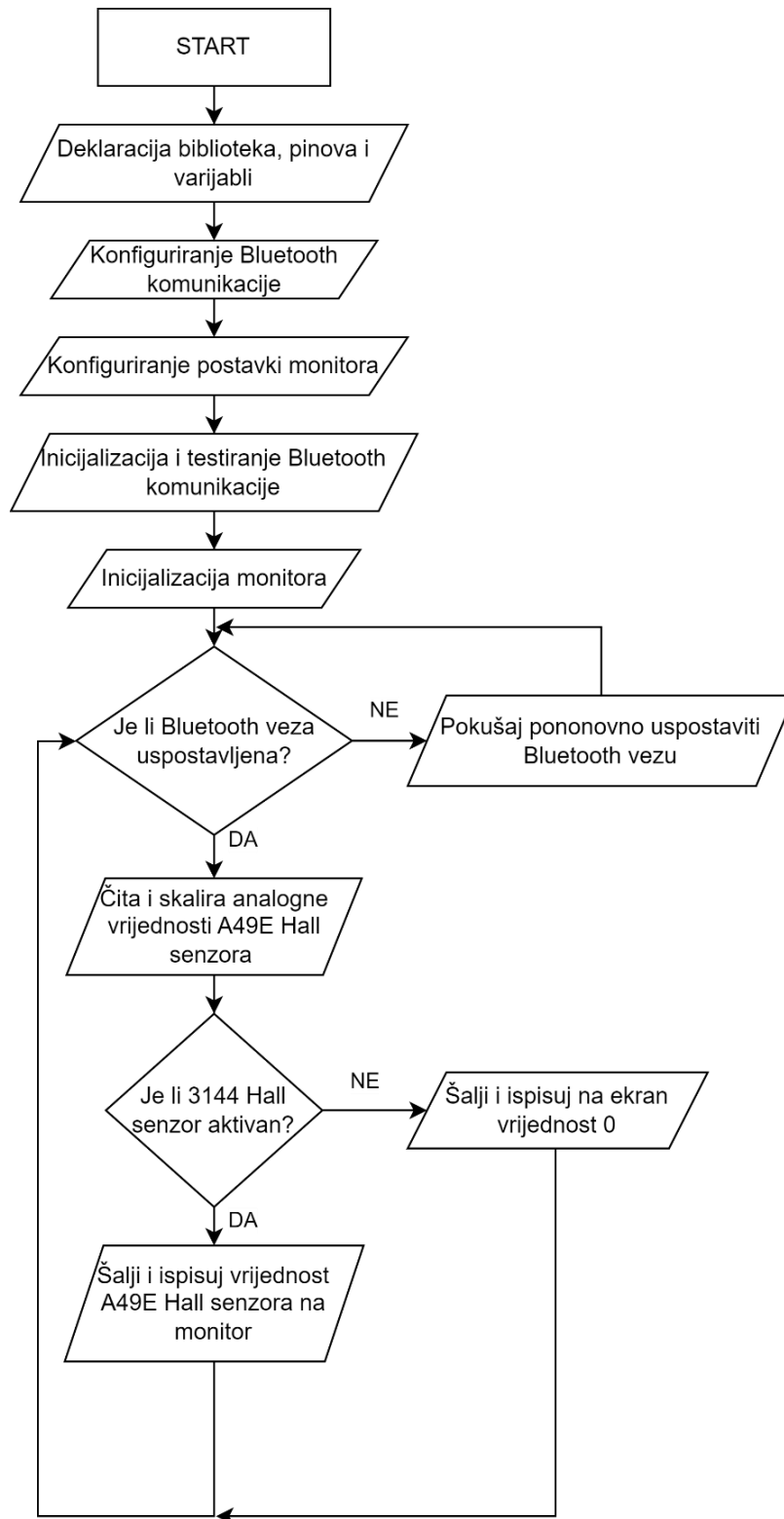
Slika 4.4. Shema prijemnika spojenog na ESC

## **5. PROGRAMSKA PODRŠKA**

Programska podrška sustava napisana je u C programskom jeziku u Arduino IDE programskom sučelju. U ovome poglavlju detaljno je objašnjena programska podrška daljinskog upravljača i prijemnika u električnoj surf dasci.

### **5.1. Programska podrška daljinskog upravljača**

Na slici 5.1. je prikazan dijagram toka daljinskog upravljača. Uz pomoć blok dijagrama toka programa se može primjetiti da su prvo deklarirane potrebne biblioteke, pinovi i varijable. Tada se konfigurira Bluetooth komunikacija i postavlja monitor te se onda inicijaliziraju. Beskonačna petlja na početku ispitiva stanje Bluetooth veze te ako nije uspostavljena pokušava ju obnoviti. Kada je Bluetooth veza uspostavljena program učitava i skalira analogne vrijednosti hall senzora A49E te ispitiva stanje „osigurača“. Ako je osigurač aktivan analogne vrijednosti se bežično šalju prijemniku, a kada se „osigurač“ deaktivira prijemniku se šalje vrijednost 0.



Slika 5.1. Dijagram toka programa daljinskog upravljača

Prvi korak kod realizacije programske podrške daljinskog upravljača je deklaracija potrebnih biblioteka. To su biblioteke Bluetooth Classic komunikacije, TFT ekrana i SPI komunikacije.

```
#include "BluetoothSerial.h" // biblioteka za Bluetooth komunikaciju
#include <Adafruit_GFX.h> // biblioteka za grafiku ekrana
#include <Adafruit_ST7735.h> // biblioteka za TFT 1.8" ekran
#include <SPI.h> // biblioteka za SPI serijsku komunikaciju
```

Konfiguracija Bluetooth Classic komunikacije postiže se provjerom konstanti `CONFIG_BT_ENABLED` i `CONFIG_BLUEDROID_ENABLED`. Ako je konfiguracija onemogućena tada će se generirati greška porukom `#error`.

```
// Konfiguracija Bluetooth komunikacije
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to enable it
#endif
```

Također, konfigurira se i Bluetooth SPP profil te ako je konfiguracija onemogućena tada se generira greška porukom `#error`.

```
// Konfiguracija SPP profila Bluetooth komunikacije
#if !defined(CONFIG_BT_SPP_ENABLED)
#error Serial Bluetooth not available or not enabled. It is only available for
the ESP32 chip.
#endif
```

`SerialBT` je objekt kojim je omogućena komunikacija putem Bluetootha funkcijom `SerialBT.begin(myName, true)` koja pokreće Bluetooth vezu s imenom uređaja `myName`. Drugi argument funkcije `true` označava da se mikrokontroler postavlja u način rada pošiljatelja te se uspostavlja veza s primateljem.

```
SerialBT.begin(myName, true); // postavljanje uređaja u master mode rada
```

Funkcijom `connectToSlave()` provjerava se Bluetooth veza. Ako Bluetooth veza nije uspostavljena funkcijom `SerialBT.connect()`, omogućena je ponovna uspostava veze s primateljem.

```

void connectToSlave() { // funkcije za povezivanje
  if (!SerialBT.connected()) { // ako uređaji nisu povezani, uđi u petlju
    Serial.printf("Pokusaj povezivanja sa slave BT device imena \"%s\"...\n",
                  slaveName.c_str());
    isConnected = SerialBT.connect(slaveName); // pokušaj ponovnog
                                                povezivanja sa primateljem
  }
}

```

U beskonačnoj petlji, nakon što je uspostavljena stabilna Bluetooth veza između pošiljatelja i primatelja, skaliraju se vrijednosti analognog A49E Hall senzora. Vrijednosti ADC mikroracunala ESP32 su  $2^{12}$  bita odnosno u rasponu od 0 do 4095 te su skalirane u rasponu vrijednosti DAC  $2^8$  bita odnosno od 0 do 255.

Razlozi skaliranja vrijednosti ADC vrijednosti su karakteristika DAC-a čiji raspon vrijednosti iznosi  $2^8$  bita kao i smanjenje vremena odziva (engl. *Latency*) čime je ubrzana komunikacija između pošiljatelja i primatelja. Naredbom `analogRead(AN_Pot)` čitaju se analogne vrijednosti A49E Hall senzora koje se potom pohranjuju u varijablu `analogUlaz`. Skalirane vrijednosti varijable `analogUlaz` pohranjuju se u varijablu `analogIzlaz`.

```

float analogUlaz = analogRead(AN_Pot); // očitavanje analognih
vrijednosti Hall senzora
int analogIzlaz = analogUlaz / 4095 * 255; // skaliranje vrijednosti od 0
do 255

```

U beskonačnoj petlji vrši se provjera stanja digitalnog A3144 Hall senzora. Ako je Hall A3144 senzor aktivan tada se funkcijom `SerialBT.println(String(analogIzlaz))` šalju analogne vrijednosti A49E Hall senzora Bluetooth vezom primatelju.

```

if (digitalRead(hallPin) == LOW) { // ako je osigurač upaljen, uđi u petlju
  SerialBT.println(String(analogIzlaz)); // slanje skaliranih vrijednosti
  (...)
}

```

Funkcijom `tft.print()` ispisuju se analogne vrijednosti na TFT 1.8" ekran.

```

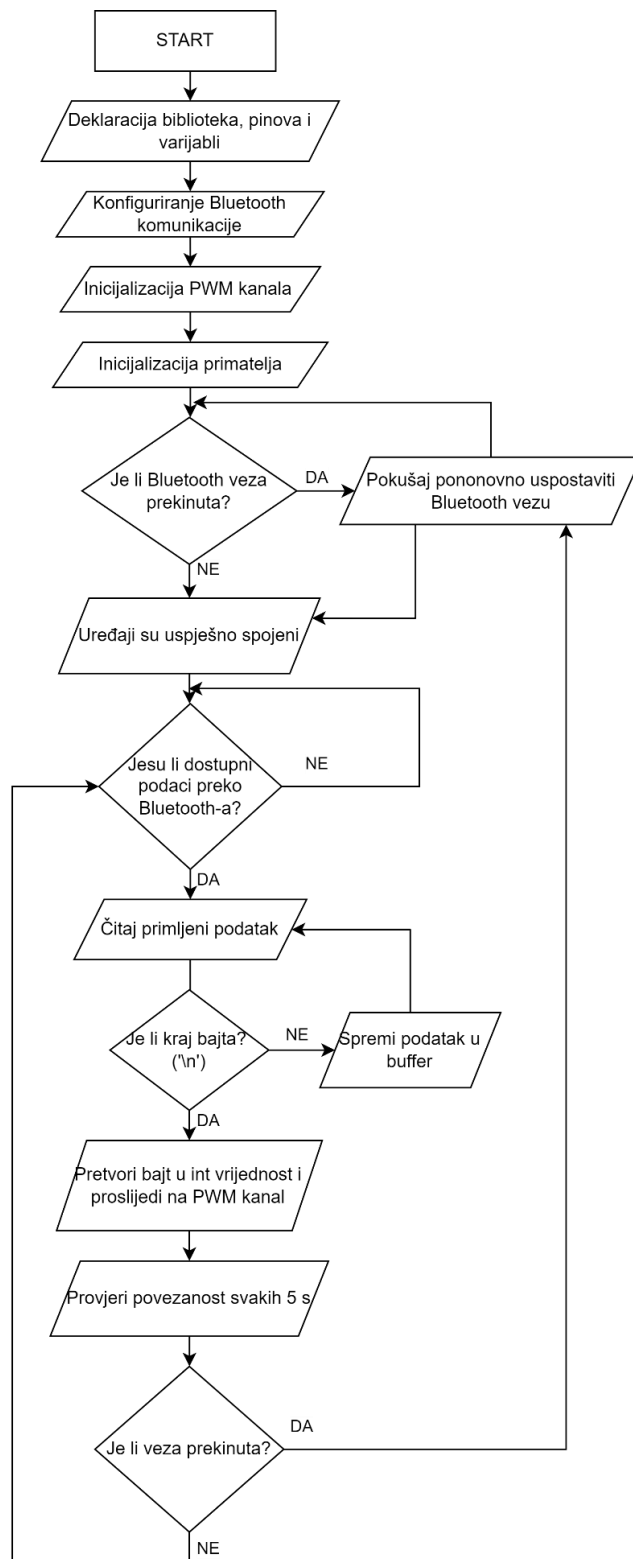
tft.print(analogIzlaz);

```

## **5.2. Programska podrška prijemnika u električnoj surf dasci**

Na slici 5.2. je prikazan dijagram toka prijemnika u električnoj surf dasci. Blok dijagram toka prikazuje kako se prvo deklariraju potrebne biblioteke, pinovi i varijable. Tada se konfigurira Bluetooth komunikacija, inicijalizira pulsno-širinski kanal i Bluetooth veza. Beskonačna petlja na početku ispitiva stanje Bluetooth veze te ako je nema pokušava je uspostaviti. Kada je veza uspostavljena program ispitiva dostupnost podataka koje daljinski upravljač šalje. Kada primi podatke, pretvara ih iz bajta u int tip podataka te prosljeđuje na pulsno-širinski kanal. Stanje Bluetooth veze se provjerava svakih 5, ukoliko je veza prekinuta program pokušava vezu ponovno uspostaviti.





Slika 5.2. Dijagram toka programa prijemnika

Prvi korak u realizaciji programske podrške prijemnika u električnoj surf dasci je deklaracija Bluetooth Classic biblioteke.

```
#include "BluetoothSerial.h" // biblioteka za Bluetooth komunikaciju
```

U `void setup()` petlji prijemnika inicijalizira se kanal za pulsno-širinsku modulaciju koja se koristi za kontrolu ESC-a. Naredbom `ledcAttachPin()` izlaz ESC-a povezuje se na kanal pulsno-širinske modulacije dok naredba `ledcSetup()` konfigurira pulsno-širinsku modulaciju određenog kanala, frekvencije i rezolucije.

```
ledcAttachPin(ESC, PWM1_Ch); // inicijalizacija kanala
                             // za ESC pin
ledcSetup(PWM1_Ch, PWM1_Freq, PWM1_Res); // inicijalizacija
                                         // pulsno-širinskog kanala
```

Funkcijom `SerialBT.begin()` pokreće se Bluetooth komunikacija te prijemnik počinje uparivanje s drugim Bluetooth uređajem.

```
SerialBT.begin(myName); // postavljanje uređaja u slave mode rada
```

U beskonačnoj petlji provjerava se dostupnost podataka funkcijom `SerialBT.available()`. Funkcijom `SerialBT.read()` čitaju se primljeni bajtovi te se potom spremaju u varijablu `RxByte` tipa `char`.

```
void loop() {
  if (SerialBT.available()) { // provjera dostupnosti podataka
                              // s Bluetooth veze
    char RxByte = SerialBT.read(); // čitanje podataka od pošiljatelja
    (...)
  }
}
```

Potom se vrši usporedba primljenih bajtova s *new line characterom* (“\n”). Kada se pojavi *new line character* (“\n”) program broj spremljen u `RxByte` pretvara u cijelobrojnu vrijednost integer, funkcijom `RxBuffer.toInt()`.

```

if (RxByte == '\n') { // ako je registriran '\n',
                      // uđi u petlju
    int receivedValue = RxBuffer.toInt(); // pretvaranje byte vrijednosti
                                          // u int tip podatka
    (...)
} else { // ako nije registriran '\n',
         // uđi u petlju
    RxBuffer += RxByte; // spremanje podataka u RxBuffer
}

```

Naredbom `ledcWrite()` primljene vrijednosti šalju se na kanal pulsno-širinske modulacije.

```

ledcWrite(PWM1_Ch, receivedValue); // pošalji vrijednost na
                                   // pulsno-širinski kanal

```

Nakon primljene vrijednosti od tri uzastopna bajta vrši se provjera povezanosti svakih 5 sekundi te ako je veza izgubljena, funkcijom `tryReconnect()` omogućava se ponovno uspostavljanje Bluetooth veze.

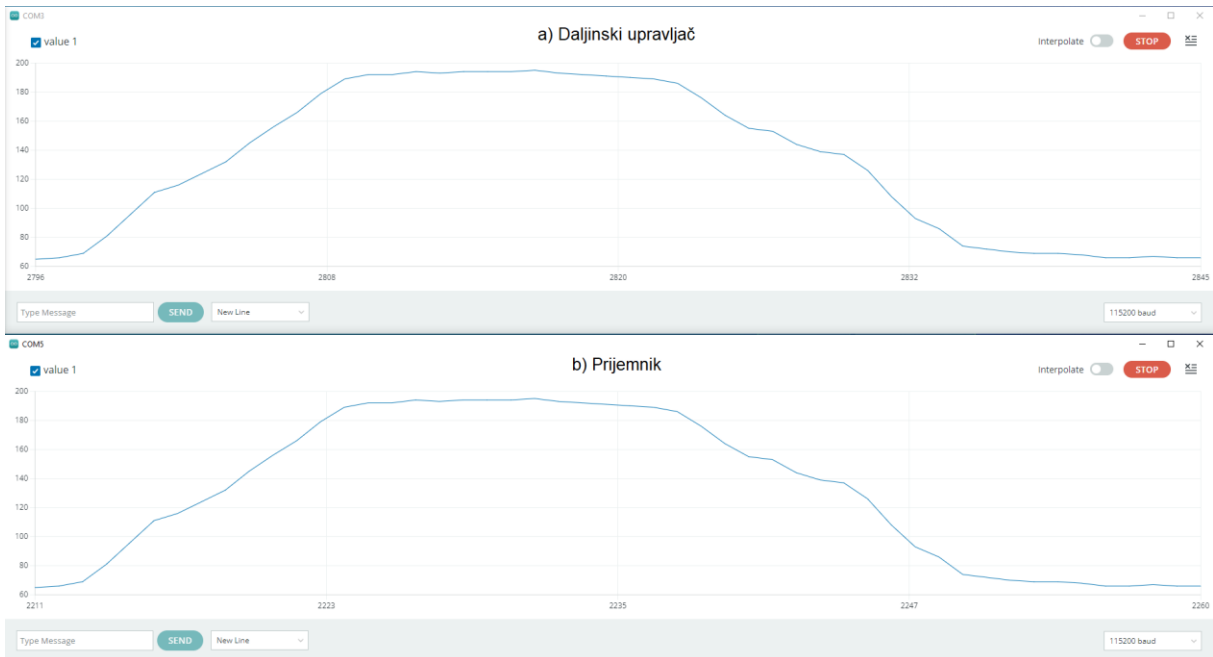
```

if (millis() % 5000 == 0) { // izvršava se svakih 5 s
    tryReconnect(); // poziva funkciju za obnavljanje veze
}

```

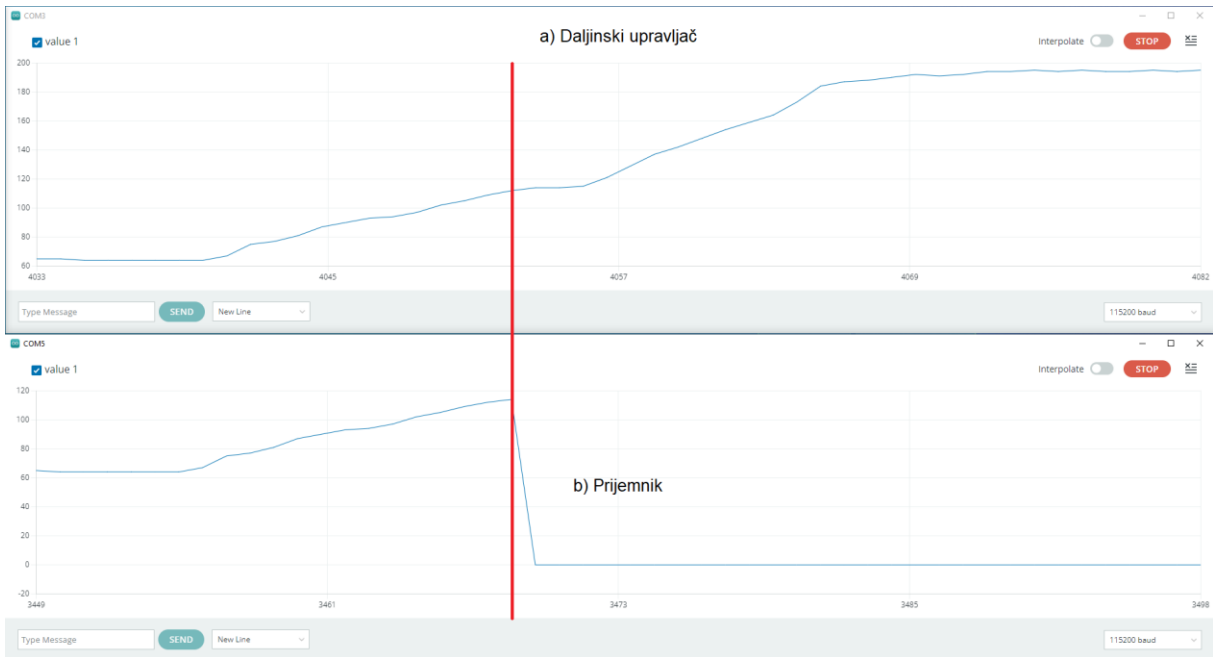
### 5.3. Testiranje programa

Programski kod je testiran na brzinu, prekid veze i sigurnost. Korištenjem Serial Plottera u programskom sučelju Arduino IDE testirana je brzina programa. Na jedan COM port je spojen ESP32 daljinskog upravljača, a na drugi COM port je spojen ESP32 prijemnika. Vrijednosti očitane s Hall senzora u punom rasponu vrijednosti od 0 do 255 unutar daljinskog upravljača se šalju gotovo instantno što se može vidjeti na slici 5.3.



Slika 5.3. Prijenos vrijednosti a) Daljinski upravljač i b) Prijemnik

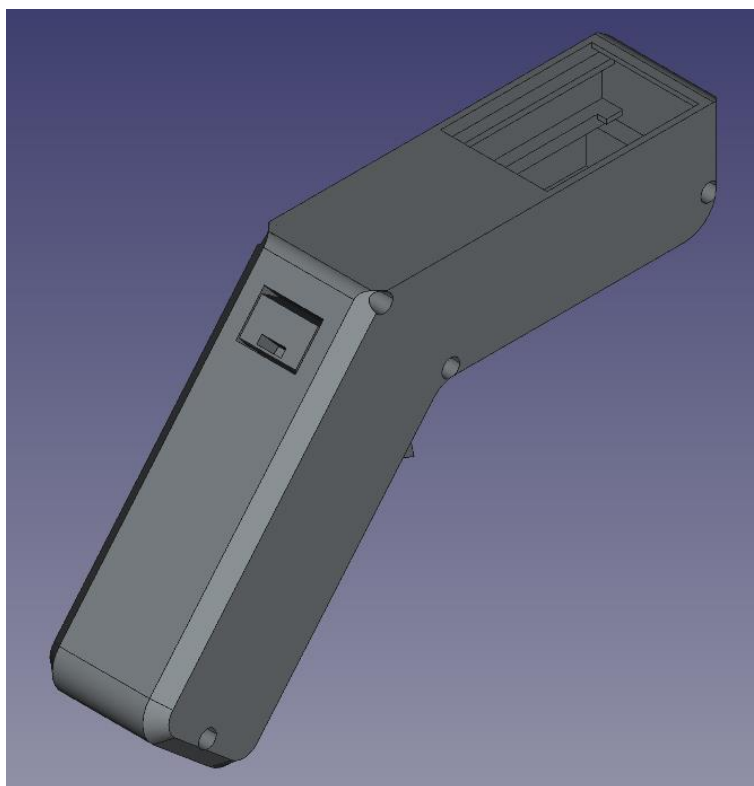
Prekid veze je simuliran uzastopnim gašenjem i paljenjem daljinskog upravljača dok je prijemnik stalno upaljen. Tijekom testiranja pouzdasnosti sustava su se daljinski upravljač i prijemnik svaki put spojili što su idealni rezultati. Sigurnosni aspekt daljinskog upravljača leži u „osiguraču“ odnosno digitalnom hall senzoru A3144. Kada senzor ne detektira magnetsko polje mora biti onemogućen prijenos analognih vrijednosti hall senzora A49E odnosno treba se slati vrijednost 0. Upravo je to i rezultat testiranja što se može vidjeti na slici 5.4. S crvenom vertikalnom linijom označen je trenutak kada je „osigurač“ prekinut te na prijemnoj strani od tog trenutka primjećujemo vrijednosti 0 unatoč aktivnom analognom hall senzoru A49E.



Slika 5.4. Prijenos vrijednosti a) Daljinski upravljač, b) Prijemnik i „osigurač“

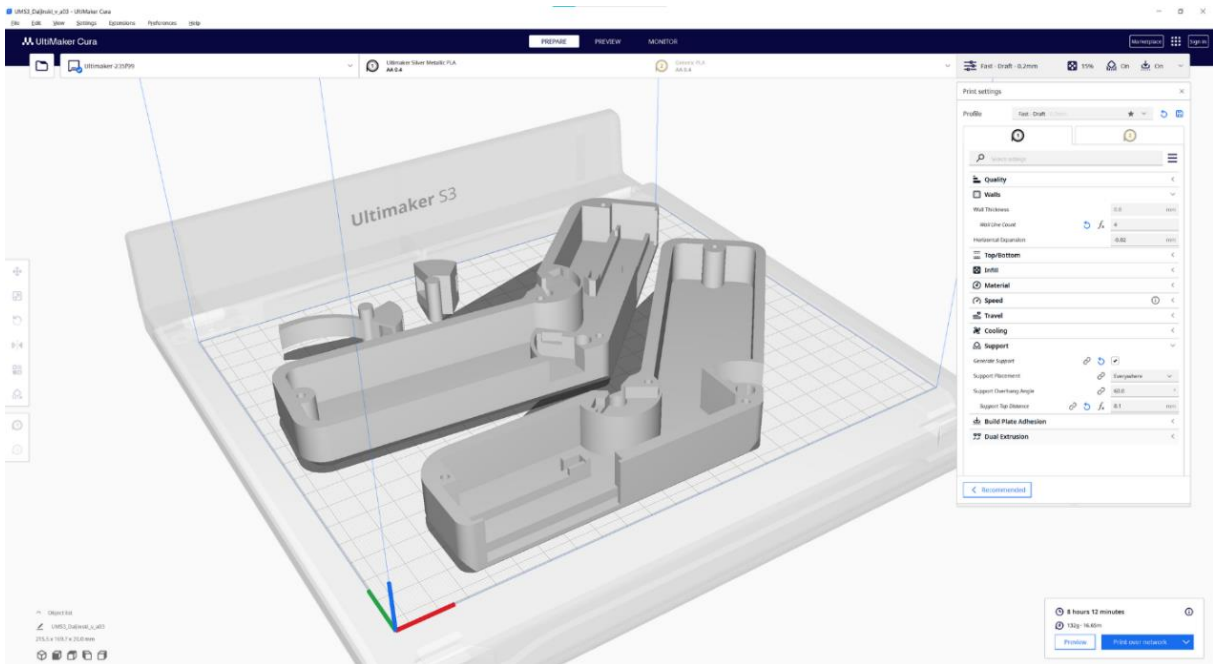
## 6. 3D MODELIRANJE DALJINSKOG UPRAVLJAČA

Treća faza razvoja daljinskog upravljača je izrada prototipa kućišta daljinskog upravljača, što je ključni korak u procesu dizajna i proizvodnje ovog uređaja. Za postizanje ovog cilja, korišten je softverski alat FreeCAD za izradu 3D modela daljinskog upravljača. Na slici 6.1. je prikazan 3D model daljinskog upravljača unutar FreeCAD progama.

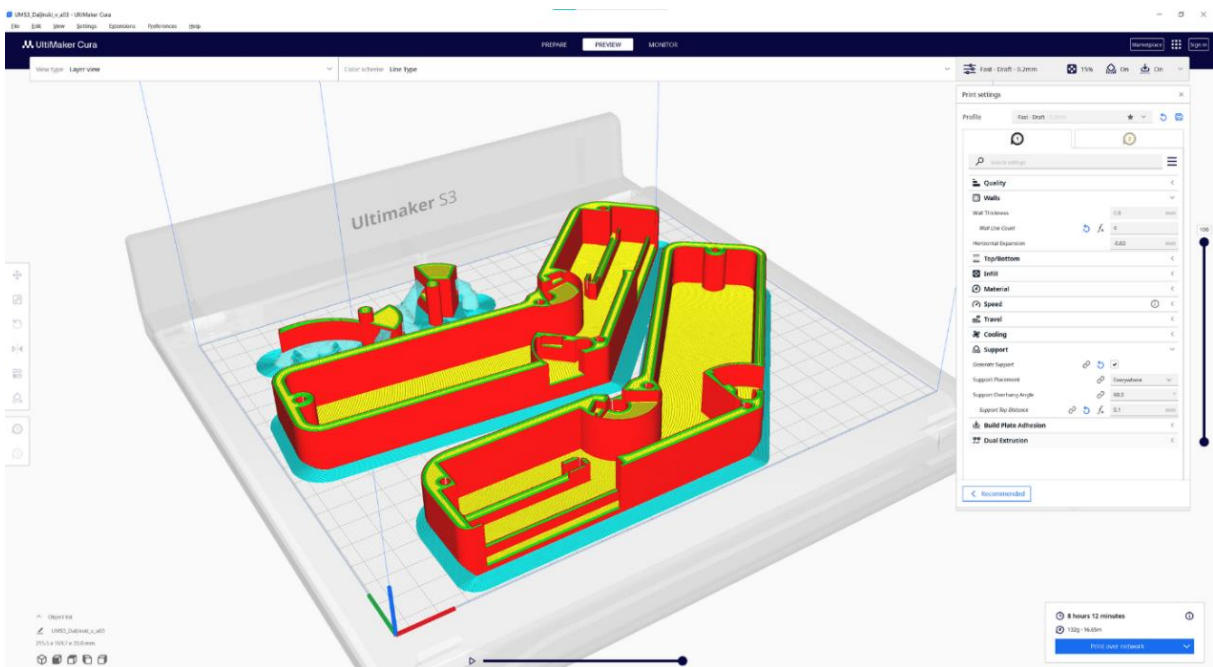


Slika 6.1. 3D model daljinskog upravljača

Priprema 3D modela za 3D ispis je napravljena u programu UltiMaker Cura. Unutar programa UltiMaker Cura su konfigurirani parametri ispisa kao što su brzina ispisa, gustoća slojeva i podrška „visećim“ dijelovima kako bi se osigurao čvrst, precizan i funkcionalan ispis modela. Programsko sučelje UltiMaker Cura-e se može vidjeti na slikama 6.2. i 6.3.



Slika 6.2. UltiMaker Cura prilikom pripreme modela za printanje



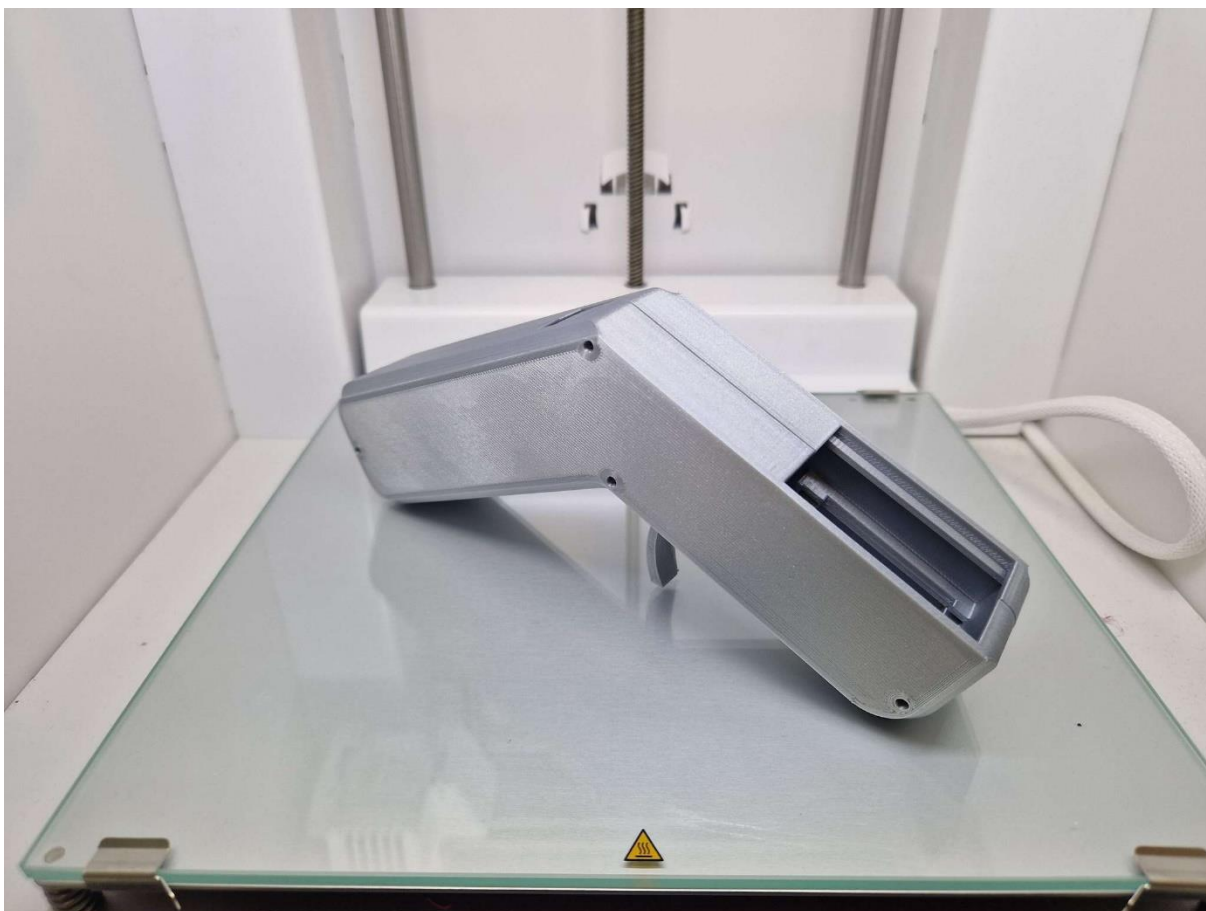
Slika 6.3. UltiMaker Cura prilikom pregleda modela prije ispisa

Za izradu fizičkog prototipa korištena je tehnologija FDM, odnosno 3D printer UltiMaker S3. Na slikama 6.4. i 6.5. se vide modeli nakon završetka printanja te sastavljeni model.



Slika 6.4. Radna ploča printera nakon završetka printanja





Slika 6.5. Očišćeni i sastavljeni modeli čine cjelinu daljinskog upravljača

## 7. ZAKLJUČAK

Ovaj rad donosi razvoj funkcionalnog prototipa daljinskog upravljača koristeći ESP32-WROOM-32 mikrokontroler, analogni i digitalni Hall-effect senzor i TFT 1.8" ekran. Za ostvarivanje komunikacije između daljinskog upravljača i prijemnika na električnoj surf dasci koristio se Bluetooth Classic komunikacijski protokol, dok je komunikacija između ST7735R TFT ekrana i mikrokontrolera ESP32-WROOM-32 ostvarena korištenjem SPI komunikacijskog protokola. Programska podrška za daljinski upravljač i prijemnik na električnoj surf dasci napisana je korištenjem Arduino IDE softverskog sučelja, a kućište daljinskog upravljača modelirano je u FreeCAD programu te je potom 3D isprintano na UltiMaker S3 3D printeru korištenjem FDM tehnologije.

Ispitivanje prototipa pokazalo je obećavajuće rezultate, ali prostora za daljnjim unapređenjem ne manjka, kako bi sustav postao praktično primjenjiv. Sustav se može unaprijediti na sljedeće načine: optimizacijom programskog koda kako bi se povećala pouzdanost i stabilnosti Bluetooth komunikacije, primjenom sigurnosnih protokola radi sprječavanja neovlaštene kontrole nad surf daskom, smanjenjem potrošnje energije kako bi se produžio vijek baterija, uspostavom dvosmjerne komunikacije za praćenje statusa baterije, unaprjeđenjem korisničkog sučelja dodavanjem više informacija, izbornika i novih opcija.

## LITERATURA

- [1] Espressif Systems Ltd., ESP32-WROOM-32 Datasheet, Shanghai, 2016.
- [2] Mischianti, R., ESP32 DEV KIT v1 high resolution pinout and specs, 2021., <https://mischianti.org/2021/02/17/doit-esp32-dev-kit-v1-high-resolution-pinout-and-specs/> [21.3.2023.]
- [3] Santos, S., ESP32 Pinout Reference: Which GPIO pins should you use?, 2018., <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/> [22.3.2023.]
- [4] Fried, L., 1.8" TFT Display Breakout and Shield Datasheet, 2012.
- [5] Wojciechowicz, T., Tech 101 From Symmetry Electronics: Hall Effect Sensors, 2017., <https://www.symmetryelectronics.com/blog/tech-101-from-symmetry-electronics-hall-effect-sensors/> [3.4.2023.]
- [6] BCD Semiconductor Manufacturing Ltd., Linear hall-effect IC AH49E Datasheet, 2010.
- [7] Allegro MicroSystems Inc., Sensitive hall-effect switches for high-temperature operation A3144 Datasheet, 1993.
- [8] Wi-Fi Standards Explained, 2021., <https://www.geeksforgeeks.org/wi-fi-standards-explained/> [27.8.2023.]
- [9] How nRF24L01+ Wireless Module Works & Interface with Arduino, 2018., <https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/> [27.8.2023.]
- [10] Bluetooth Technology Overview, 2023., <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/> [17.5.2023.]
- [11] Santos, R., ESP32 Bluetooth Classic with Arduino IDE – Getting Started, 2019., <https://randomnerdtutorials.com/esp32-bluetooth-classic-arduino-ide/> [8.6.2023.]

- [12] Introduction to Bluetooth Classic, 2020., <https://www.argenox.com/library/bluetooth-classic/introduction-to-bluetooth-classic/> [10.6.2023.]
- [13] Campbell, S., Basics of the SPI communication protocol, 2016., <https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/> [16.8.2023.]
- [14] Upadhyay, J., What is a PWM signal?, 2022., <https://www.circuitbread.com/ee-faq/what-is-a-pwm-signal> [8.6.2023.]

## POPIS SLIKA

Slika 1.1. Završeni prototip daljinskog upravljača .....	3
Slika 2.1. ESP32 DEV KIT v1 pinout dijagram [2] .....	5
Slika 2.2. Prednji prikaz 1.8" TFT ekrana [4] .....	7
Slika 2.3. Stražnji prikaz 1.8" TFT ekrana [4] .....	7
Slika 2.4. Način rada Hall effect senzora [5].....	8
Slika 2.5. Funkcijski blok dijagram AH49E [6].....	9
Slika 2.6. Prijenosna karakteristika AH49E [6] .....	11
Slika 2.7. Funkcijski blok dijagram A3144 [7] .....	12
Slika 3.1. Podjela na kanale frekvencijskog pojasa Bluetooth Classic-a [12].....	20
Slika 3.2. Tri topologije Bluetooth Classic-a [12].....	20
Slika 3.3. Dijagram stanja i koraka uspostavljanja Bluetooth Classic veze [12] .....	22
Slika 3.4. SPI <i>master - slave</i> komunikacija [13] .....	23
Slika 3.5. Tri različita radna ciklusa pulsno-širinske modulacije [14] .....	24
Slika 4.1. Blok shema sustava .....	26
Slika 4.2. Elektronička shema daljinskog upravljača .....	27
Slika 4.3. Shema prototipa prijemnika .....	29
Slika 4.4. Shema prijemnika spojenog na ESC .....	30
Slika 5.1. Dijagram toka programa daljinskog upravljača .....	32

Slika 5.2. Dijagram toka programa prijemnika .....	36
Slika 5.3. Prijenos vrijednosti a) Daljinski upravljač i b) Prijemnik.....	39
Slika 5.4. Prijenos vrijednosti a) Daljinski upravljač, b) Prijemnik i „osigurač“ .....	40
Slika 6.1. 3D model daljinskog upravljača.....	41
Slika 6.2. UltiMaker Cura prilikom pripreme modela za printanje.....	42
Slika 6.3. UltiMaker Cura prilikom pregleda modela prije ispisa.....	42
Slika 6.4. Radna ploča printera nakon završetka printanja.....	43
Slika 6.5. Očišćeni i sastavljeni modeli čine cjelinu daljinskog upravljača .....	44

## **POPIS TABLICA**

Tablica 2.1. Karakteristike Hall senzora AH49E [6].....	10
Tablica 2.2. Karakteristike Hall senzora A3144 [7].....	12
Tablica 3.1. Usporedba Bluetooth Classic i Bluetooth LE protokola [10].....	17
Tablica 3.2. Karakteristike Bluetooth Classic protokola [10] .....	18

## PRILOG 1 – PROGRAMSKI KOD DALJINSKOG UPRAVLJAČA

```
/*
  Završni rad
  Dizajn daljinskog upravljača za električnu surf dasku

  Programski kod daljinskog upravljača (predajnik)

  Izradila: Ana Petrić
  Mentor: Tonko Kovačević
*/
#include "BluetoothSerial.h" // biblioteka za Bluetooth komunikaciju
#include <Adafruit_GFX.h> // biblioteka za grafiku ekrana
#include <Adafruit_ST7735.h> // biblioteka za TFT 1.8" ekran
#include <SPI.h> // biblioteka za SPI serijsku komunikaciju

// Konfiguracija Bluetooth komunikacije
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to enable it
#endif

// Konfiguracija SPP profila Bluetooth komunikacije
#if !defined(CONFIG_BT_SPP_ENABLED)
#error Serial Bluetooth not available or not enabled. It is only available for
the ESP32 chip.
#endif

#define AN_Pot 4 // deklaracija analognog pina Hall senzora
const int hallPin = 3; // deklaracija digitalnog pina Hall senzora
int hallState = 0; // deklaracija varijable stanja Hall senzora

BluetoothSerial SerialBT; // deklaracija Bluetooth SerialBT objekta
String slaveName = "ESP32-BT-Slave"; // deklaracija primatelja
String myName = "ESP32-BT-Master"; // deklaracija pošiljatelja
bool isConnected = false; // deklaracija varijable za povezivanje
#define TFT_CS 5 // deklaracija chip select pina ekrana
#define TFT_RST 22 // deklaracija reset pina ekrana
#define TFT_DC 21 // deklaracija data command pina ekrana
// konfiguracija Adafruit_ST7735 ekrana
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

void connectToSlave() { // funkcije za povezivanje
  if (!SerialBT.connected()) { // ako uređaji nisu povezani, uđi u petlju
    Serial.printf("Pokusaj povezivanja sa slave BT device imena \"%s\"...\n",
      slaveName.c_str());
    isConnected = SerialBT.connect(slaveName); // pokušaj ponovnog
      povezivanja sa primateljem
  }
}
```



```

void setup() {
  Serial.begin(115200);
  SerialBT.begin(myName, true); // postavljanje uređaja u master mode rada
  Serial.printf("Uređaj \"%s\" prelazi u način rada master, uključite slave
                uređaj !\n", myName.c_str());
  connectToSlave(); // poziva funkciju za uspostavljanje veze
  if (SerialBT.connected()) { // provjera uspostavljene veze
    isConnected = true; // potvrđi uspostavljenu vezu
    Serial.println("Uređaj je uspješno povezan!");
  } else {
    isConnected = false; // deklaracija varijable za povezivanje
    Serial.println("Spajanje nije uspješno.");
  }
  tft.initR(ST77XX_WHITE);
}

void loop() {
  if (!SerialBT.connected() && !isConnected) { // provjera veze
    connectToSlave(); // poziva funkciju
                    // za uspostavljanje veze
  }
  float analogUlaz = analogRead(AN_Pot); // očitavanje analognih
                                        // vrijednosti Hall senzora
  int analogIzlaz = analogUlaz / 4095 * 255; // skaliranje vrijednosti
                                             // od 0 do 255
  if (digitalRead(hallPin) == LOW) { // ako je osigurač upaljen,
                                     // uđi u petlju
    SerialBT.println(String(analogIzlaz)); // slanje skaliranih vrijednosti

    tft.fillScreen(ST77XX_CYAN); // postavljanje pozadine TFT ekrana

    tft.setCursor(12, 10); // pozicioniranje teksta (X, Y)
    tft.setTextSize(2); // definiranje veličine teksta
    tft.setTextColor(ST77XX_RED); // definiranje boje teksta
    tft.print("AdriaFoil"); // ispis teksta na TFT ekran

    tft.setCursor(20, 60);
    tft.setTextSize(5);
    tft.setTextColor(ST77XX_WHITE);
    tft.print(analogIzlaz);

    tft.setCursor(37, 130);
    tft.setTextSize(1);
    tft.setTextColor(ST77XX_BLUE);
    tft.print("Ana Petric");

    tft.setCursor(20, 145);
    tft.print("Tonko Kovacevic");
  }
}

```

```
else { // Ako osigurač nije pritisnut, uđi u petlju
SerialBT.println(String(0)); // Slanje vrijednosti 0

tft.fillScreen(ST77XX_YELLOW);

tft.setCursor(20, 60);
tft.setTextSize(2);
tft.setTextColor(ST77XX_RED);
tft.print("Pritisni");

tft.setCursor(15, 90);
tft.print("osigurac!");
}
Serial.println(String(analogIzlaz)); // pošalji na Serial Monitor
delay(90); // odgoda za 90 ms
}
```

## PRILOG 2 – PROGRAMSKI KOD PRIJEMNIKA U SURF DASCI

```
/*
  Završni rad
  Dizajn daljinskog upravljača za električnu surf dasku

  Programski kod pošiljatelja u surf dasci

  Izradila: Ana Petrić
  Mentor: Tonko Kovačević
*/
#include "BluetoothSerial.h" // biblioteka za Bluetooth komunikaciju

#define ESC 32 // deklaracija analognog pina za ESC
#define PWM1_Ch 0 // deklaracija pulsno-širinskog kanala
#define PWM1_Res 12 // deklaracija pulsno-širinske rezolucije
#define PWM1_Freq 1000 // deklaracija pulsno-širinske frekvencije

// Konfiguracija Bluetooth komunikacije
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to enable it
#endif

// Konfiguracija SPP profila Bluetooth komunikacije
#if !defined(CONFIG_BT_SPP_ENABLED)
#error Serial Bluetooth not available or not enabled. It is only available for
the ESP32 chip.
#endif

BluetoothSerial SerialBT; // deklaracija Bluetooth SerialBT objekta
String RxBuffer = ""; // deklaracija buffer-a
String masterName = "ESP32-BT-Master"; // deklaracija primatelja
String myName = "ESP32-BT-Slave"; // deklaracija pošiljatelja
bool isConnected = false; // deklaracija varijable za povezivanje

void tryReconnect() { // funkcija za ponovno uspostavljanje veze
  if (!SerialBT.connected()) { // ako uređaji nisu povezani, uđi u petlju
    Serial.println("Pokušaj ponovnoga uspostavljanja veze...");
    isConnected = SerialBT.connect(masterName); // ponovnog povezivanje
    // s masterom
  }
  if (isConnected) { // ako su uređaji povezani, uđi u petlju
    Serial.println("Ponovno uspostavljanje veze je uspješno!");
  } else { // ako se nikako ne žele spojiti, uđi u petlju
    Serial.println("Neuspjelo spajanje. Provjeriti domet i konfiguraciju
na drugom uređaju");
  }
}
}
```

```

void setup() {
  ledcAttachPin(ESC, PWM1_Ch);           // inicijalizacija kanala
                                         // za ESC pin
  ledcSetup(PWM1_Ch, PWM1_Freq, PWM1_Res); // inicijalizacija
                                         // pulsno-širinskog kanala
  Serial.begin(115200);
  SerialBT.begin(myName); // postavljanje uređaja u slave mode rada
  Serial.println("Uređaj je pokrenut te se može povezati Bluetooth vezom!");
}

void loop() {
  if (SerialBT.available()) {           // provjera dostupnosti podataka
                                         // s Bluetooth veze
    char RxByte = SerialBT.read();      // čitanje podataka od pošiljatelja
    if (RxByte == '\n') {               // ako je registriran '\n', uđi u petlju
      int receivedValue = RxBuffer.toInt(); // pretvaranje byte vrijednosti
                                         // u int tip podatka
      Serial.println(receivedValue);     // ispis primljene vrijednosti
                                         // na plotter
      ledcWrite(PWM1_Ch, receivedValue); // pošalji vrijednost na
                                         // pulsno-širinski kanal
      RxBuffer = "";                    // isprazni buffer
    } else {                             // ako nije registriran '\n',
                                         // uđi u petlju
      RxBuffer += RxByte;               // spremanje podataka u RxBuffer
    }
  }

  if (millis() % 5000 == 0) { // izvršava se svakih 5 s
    tryReconnect();           // poziva funkciju za obnavljanje veze
  }
}

```