MODUL ZA DALJINSKO UPRAVLJANJE ROBOTIMA

Fabijanić, Kristijan

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Karlovac University of Applied Sciences / Veleučilište u Karlovcu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:128:341519

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-20**

VELEUČILIŠTE U KARLOVCU
Karlovac University of Applied Sciences

Repository / Repozitorij:

Repository of Karlovac University of Applied Sciences - Institutional Repository

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

VELEUČILIŠTE U KARLOVCU
STROJARSKI ODJEL
STRUČNI STUDIJ MEHATRONIKE

KRISTIJAN FABIJANIĆ

MODUL ZA DALJINSKO UPRAVLJANJE ROBOTIMA

ZAVRŠNI RAD

KARLOVAC, 2022.

VELEUČILIŠTE U KARLOVCU
STROJARSKI ODJEL
STRUČNI STUDIJ MEHATRONIKE

KRISTIJAN FABIJANIĆ

MODUL ZA DALJINSKO UPRAVLJANJE ROBOTIMA

ZAVRŠNI RAD

MENTOR:

dr.sc. DENIS KOTARSKI

KARLOVAC, 2022.

KARLOVAC UNIVERSITY OF APPLIED SCIENCES
MECHANICAL ENGINEERING DEPARTMENT
STUDY OF MECHATRONICS

KRISTIJAN FABIJANIĆ

MODULE FOR REMOTE CONTROL OF ROBOTS

UNDERGRADUATE THESIS

MENTOR:

dr.sc. DENIS KOTARSKI

KARLOVAC, 2022.

VELEUČILIŠTE U KARLOVCU Karlovac University of Applied Sciences	Klasa: 602-07/__-01/____
ZADATAK ZAVRŠNOG / DIPLOMSKOG RADA	Datum:

* Ime i prezime	Kristijan Fabijanić		
OIB / JMBG			
Adresa			
Tel. / Mob./e-mail			
Matični broj studenta			
JMBAG			
Studij(staviti znak **X** ispred odgovarajućeg studija)	X preddiplomski	specijalistički diplomski	
Naziv studija	Stručni studij mehatronike		
Godina upisa			
Datum podnošenja molbe			
Vlastoručni potpis studenta/studentice			

.

* Naslov teme na hrvatskom:Modul za daljinsko upravljanje robotima
* Naslov teme na engleskom:Module for remote control of robots
Opis zadatka: U završnom radu razmatra se projektiranje, izvedba i programiranje modula za daljinsko upravljanje robotima koji se temelji na primjeni arduino upravljačke jedinice s pripadnim komponentama. U okviru radu potrebno je: • opisati robotske sustave i načine daljinskog upravljanja, • odabrati komponente i projektirati funkcionalnu cjelinu modula daljinskog upravljanja, • napisati programski kod koji se može koristiti za upravljanje različitih kategorija robota, • testirati modul za slučaj upravljanja mobilnim robotom s diferencijalnom i svesmjernom konfiguracijom pogona.

Mentor:	Predsjednik Ispitnog povjerenstva:

PREDGOVOR

Izjavljujem da sam ovaj rad izradio samostalno koristeći se stečenim znanjima tijekom obrazovanja, te uz navedenu stručnu literaturu.

Posebno se zahvaljujem svojoj obitelji koja mi je omogućila školovanje te bila podrška u istom.

Posebno hvala profesoru i mentoru Denisu Kotarskom na svim stručnim savjetima i pomoći, te strpljenju oko izrade završnog rada.

Kristijan Fabijanić

Karlovac, travanj 2022. _____

SAŽETAK

Roboti su sve više prisutni u svim područjima industrije, raznih djelatnosti, u obrazovanju, za zabavu i slobodno vrijeme, a također i kao uređaji za kućanstvo. U ovom završnom radu tema je modul za daljinsko upravljanje robotima, odnosno projektiranje i izrada jednog opće primjenjivog uređaja koji će moći upravljati raznim inačicama robota putem Bluetooth tehnologije za komuniciranje daljinskim bežičnim putem. U ovom radu upravljačka jedinica je Arduino mikrokontroler koji koristi HC-05 Bluetooth modul za komunikaciju sa željenim robotom. Upravljački program je izveden na način da se postigne jednostavan i pouzdan način povezivanja i komunikacije upravljača sa željenim robotom te mogućnost odabira i promjene istog.

Ključne riječi: Daljinsko upravljanje, edukacijski roboti, HC-05 Bluetooth modul, bežična komunikacija, projektiranje sustava

SUMMARY

Robots are increasingly present in all areas of industry, various business activities, education, entertainment, and free-time activities, and also as household appliances. The topic of this paper is a module for remote control of robots, which includes the design and development of a generally applicable device that will be able to control various versions of robots via Bluetooth technology for remote wireless communication. In this paper, the control unit is an Arduino microcontroller that uses the HC-05 Bluetooth module to communicate with the desired robot. Program code is written in such a way that it is possible to achieve a simple and reliable way of connecting and communicating with the controller and the desired robot and the ability to select and change it.

Keywords: Remote control, educational robots, HC-05 Bluetooth module, wireless communication, system design

SADRŽAJ

1 UVOD 1

1.1 Robotski sustavi 1

1.2 Daljinsko upravljanje 2

2 ARDUINO PROGRAMSKO I RAZVOJNO SUČELJE 3

2.1 Arduino IDE 3

2.2 Arduino Nano razvojna pločica 4

3 PROJEKTIRANJE I IZVEDBA SKLOPOVSKOG SUSTAVA 7

3.1 Odabir komponenti sklopovskog sustava upravljača 7

3.2 HC-05 Bluetooth modul 8

3.2.1 Modifikacija HC-05 modula 9

3.2.2 Shema spajanja HC-05 pločice na Arduino Nano 9

3.2.3 Shema spajanja HC-05 prilikom konfiguriranja putem Serial Monitor-a 12

3.3 Digitalni i analogni ulazi 12

3.4 LCD display 13

4 IZVEDBA PROGRAMSKOG SUSTAVA 15

4.1 Konfiguriranje HC-05 modulakoristeći AT+ naredbe 15

4.2 Korisničko sučelje modula za daljinsko upravljanje robotima 16

4.3 Programsko rješenje BT upravljača 17

5 ZAKLJUČAK 40

LITERATURA 41

PRILOZI 42

POPIS SLIKA

Slika 1 : Edukacijski robot s gusjenicama 1
Slika 2 : KUKA OmniMov 2
Slika 3 : Arduino IDE, prostor za pisanje programskog koda 3
Slika 4 : Arduino IDE, Serial Monitor 4
Slika 5 : Arduino Nano razvojna pločica 4
Slika 6 : Eksperimentalna pločica 5
Slika 7 : Raspored pinova Arduino Nano razvojne pločice 6
Slika 8 : Simbolički prikaz odabranih komponenata 7
Slika 9 : HC-05 modul na pločici 8
Slika 10 : Modifikacija HC-05 pločice 9
Slika 11 : Shema spajanja HC-05 pločice 11
Slika 12 : Shema spajanja HC-05 pločice za konfiguriranje 12
Slika 13 : Električna shema spajanja ulaza 13
Slika 14 : 1.3 inčni LCD display 14
Slika 15 : Shema spajanja LCD display-a na Arduino Nano 14
Slika 16 : Primjer unosa AT naredbe putem Serial Monitor-a 16
Slika 17 : Izgled komunikacijskog moda na LCD-u 17
Slika 18 : Izgled konfiguracijskog moda na LCD-u 17

POPIS TABLICA

Tablica 1 : Tehničke specifikacije Arduino Nano razvojne pločice.....5

Tablica 2 : Važnije AT+ naredbe.....15

Tablica 3 : Važnije AT+ naredbe, nastavak Tablice 2.....16

1. UVOD

Robotika je primijenjena tehnička znanost koja predstavlja spoj strojeva i računalne tehnike. Neka područja koje robotika pokriva su projektiranje strojeva, teorija upravljanja i regulacije, područje elektronike, računalnog programiranja, područje umjetne inteligencije, ljudski faktor i teoriju proizvodnje. Robotika se prvenstveno bavi istraživanjem i projektiranjem strojeva koji mogu zamijeniti čovjeka u izvršavanju zadataka, kao što su razni oblici fizičkog rada, donošenja odluka te obavljanja rada koji može biti opasan za ljudsko zdravlje. [1]

1.1. Robotski sustav

Robotski sustav se sastoji od više podsustava poput pogonskog(aktuatorskog), energetskog, senzorskog(mjernog), upravljačkog te mehaničkog podsustava. Pogonski podsustav sustav može biti elektromotorni, pneumatski, hidraulički. Pogonski podsustav uključuje izvršne elemente poput elektromotora, pneumatske, hidrauličke cilindre, prijenosnike snage i gibanja te popratne sklopove za upravljanje tokom energije. U energetski podsustav robota spadaju izvori napajanja. Senzorski podsustav robota uključuje razne senzore koji daju povratnu informaciju o stanju robota upravljačkom podsustavu. Upravljački podsustav obrađuje podatke dobivene od senzorskog podsustava, izračunava i šalje upravljačke signale pogonskom podsustavu za pokretanje aktuatora. Mehanički podsustav robota uključuje konstrukciju robota, mehanizme za pretvorbu energije pogonskog podsustava u potrebno gibanja, te ostale mehanizme. [2]

Slika 1 : Edukacijski robot s gusjenicama [3]

Slika 2 : KUKA OmniMove [4]

1.2. Daljinsko upravljanje

Daljinsko upravljanje je vođenje objekta upravljanja na daljinu, gdje udaljenost između upravljačkog sustava i upravljanog objekta nije strogo definirana. Prijenos upravljačkog signala na daljinu može se ostvariti preko elektromagnetskih valova s ožičenjem ili bez ožičenja. Sustav daljinskog upravljanja se sastoji od upravljačkog člana, odašiljača, prijenosne veze, prijemnika i izvršnog člana. Regulaciju upravljanja može vršiti čovjek ili senzorski podsustav. [5]

Neki od načina slanja upravljačkog signala bežičnom vezom su:

IR diode – informacije se šalje putem infracrvene svjetlosti. Infracrvena svjetlost se nalazi van vidljivog spektra svjetlosti. Nedostatak ovakvog upravljanja je što odašiljač i prijemnik zahtijevaju međusobnu vidljivost. Ovaj način upravljanje se najčešće može pronaći kod daljinskog upravljanja televizora, klime i ostalih sličnih uređaja.

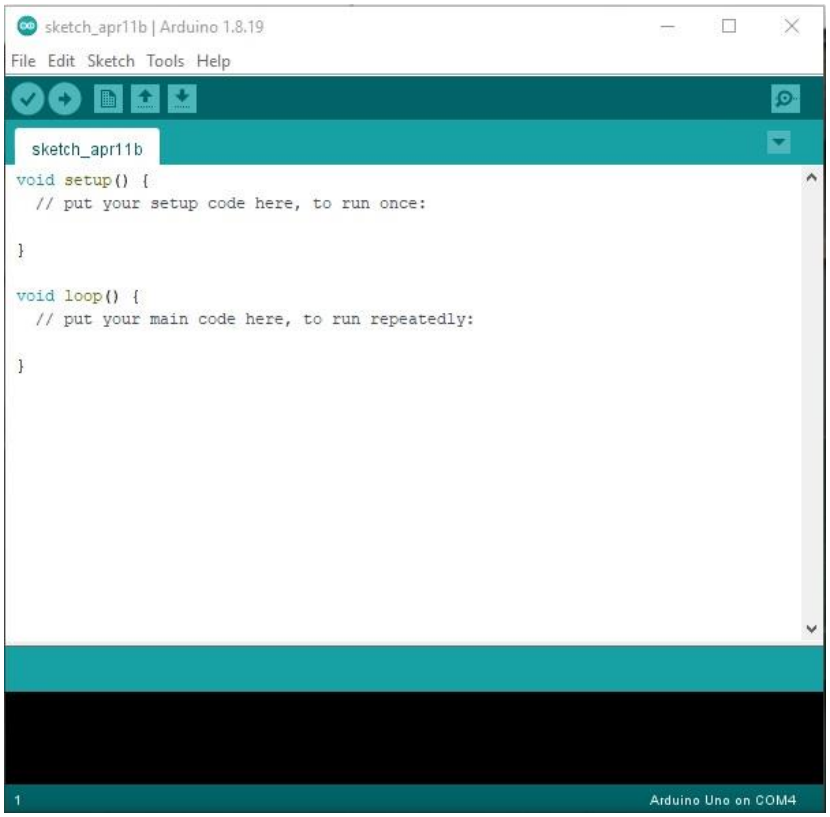
Radiovalovima – informacije se šalju zrakom koristeći elektromagnetske valove kao nositelje informacija. Uređaji i protokoli koji rade na osnovu radiovalova a koriste se za daljinsko upravljanje rade na frekvencijama između 0.3 i 3 GHz. Također se mogu koristeći radiovalove koristiti i tehnologije specijalizirane za prijenos veće količine podataka u što kraćem vremenu poput WiFi i Bluetooth tehnologija koje rade na frekvenciji od 2.4 GHz.

2. ARDUINO PROGRAMSKO I RAZVOJNO SUČELJE

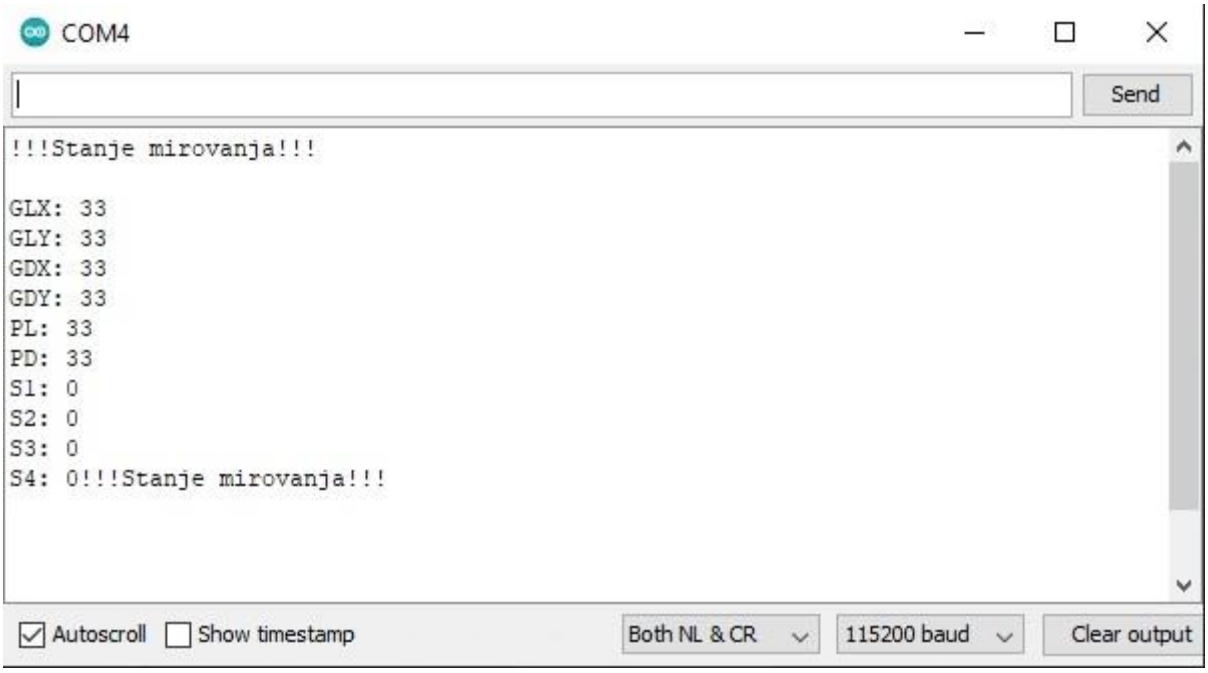
Arduino je platforma otvorenog tipa(eng. open-source) temeljena na hardware-u i software-u kojeg je jednostavno koristiti. Arduino pločice mogu čitati razne ulaze, npr. svijetlo, temperaturu na senzoru, pritisak tipkala, te zatim te informacije obraditi i proslijediti na izlaz i ostvariti radnju poput pokretanja elektromotora, uključivanja LED diode, ispis na ekranu. To što će mikrokontroler učiniti ovisi o napisanom programskom kodu koji se piše u Arduino programskom okruženju (Arduino IDE) koje je slično C programskom jeziku. [6]

2.1. Arduino IDE

Arduino IDE (IDE – Integrated Development Environment) je programsko okruženje koje sadrži uređivač teksta za pisanje programskog koda, područje za poruke, programsku traku s padajućim izbornicima za pojedine cjeline. Navedeno programsko okruženje povezuje se s hardware-om te koristeći compiler pretvara napisani programski kod u binarni oblik te ga upisuje u mikrokontroler pločice. Jedan od važnijih alata prilikom korištenja Arduino IDE-a je Serial Monitor. Na prozoru Serial Monitor-a se mogu ispisivati povratne informacije putem serijske veze Arduino pločice i računala.

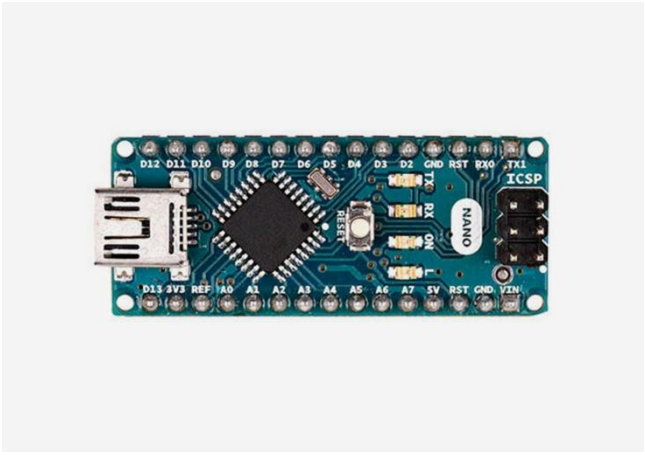


Slika 3 : Arduino IDE, prostor za pisanje programskog koda



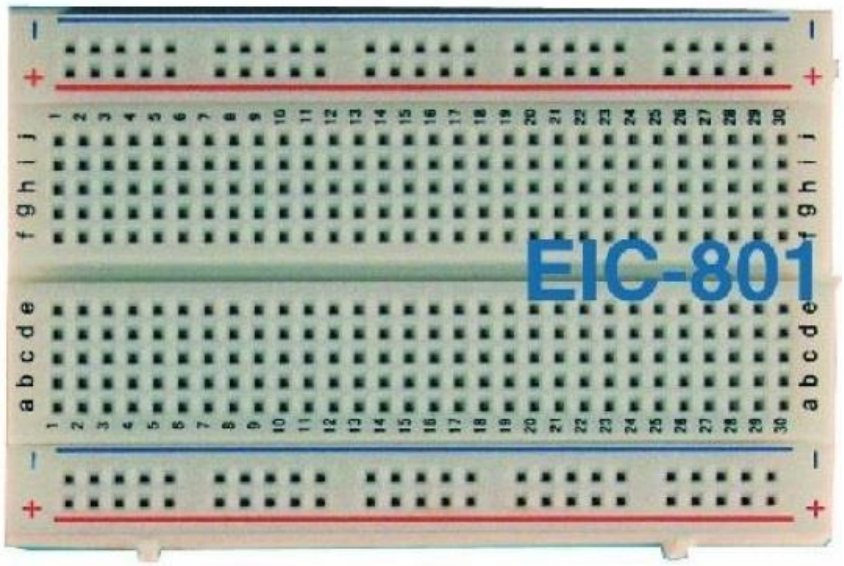
Slika 4 : Arduino IDE, Serial Monitor

2.2. Arduino Nano razvojna pločica



Slika 5 : Arduino Nano razvojna pločica [6]

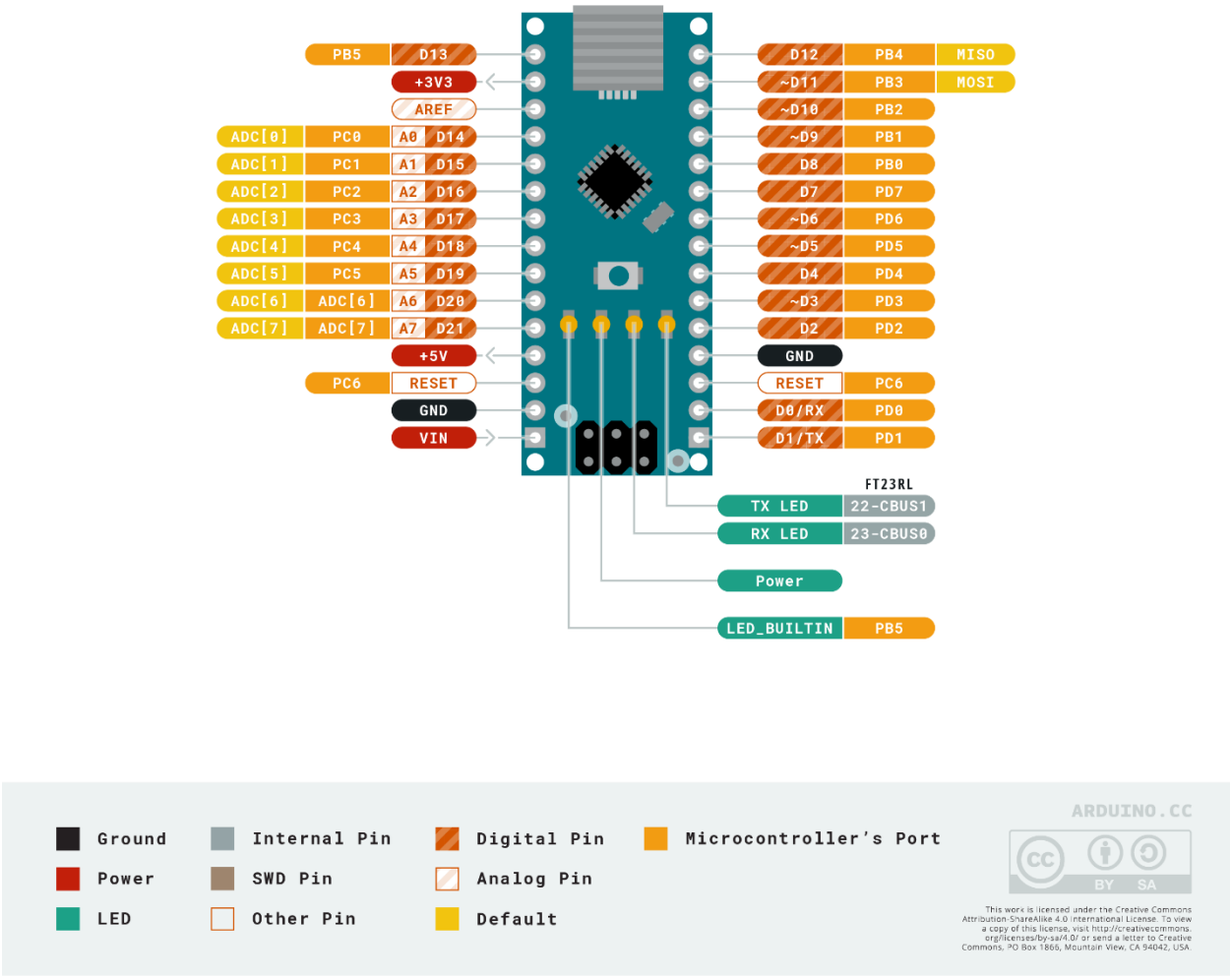
Arduino Nano je pločica malih dimenzija, prikladna za korištenje na eksperimentalnoj pločici za potrebe razvoja prototipova. Pločica se napaja preko USB Mini-B kabela, te koristi isti za potrebe unosa programa. Također se može napajati preko pinova namijenjenih za napajanje.



Slika 6 : Eksperimentalna pločica [3]

Mikrokontroler	Atmega328
Radni napon	5 V
Flash memorija	32 KB, od kojih se 2 KB koristi za bootloader
SRAM	2 KB
EEPROM	1 KB
Radna frekvencija	16 MHz
Broj analognih ulaznih/izlaznih pinova	8
Broj digitalnih ulaznih/izlaznih pinova	22
Broj pinova sposobnih za PWM signal	6
Ulazni napon	7 - 12 V
Maksimalna dozvoljena struja na ulaznim/izlaznim pinovima	20 mA
Dimenzije pločice	18 x 45 mm

Tablica 1 : Tehničke specifikacije Arduino Nano razvojne pločice [6]



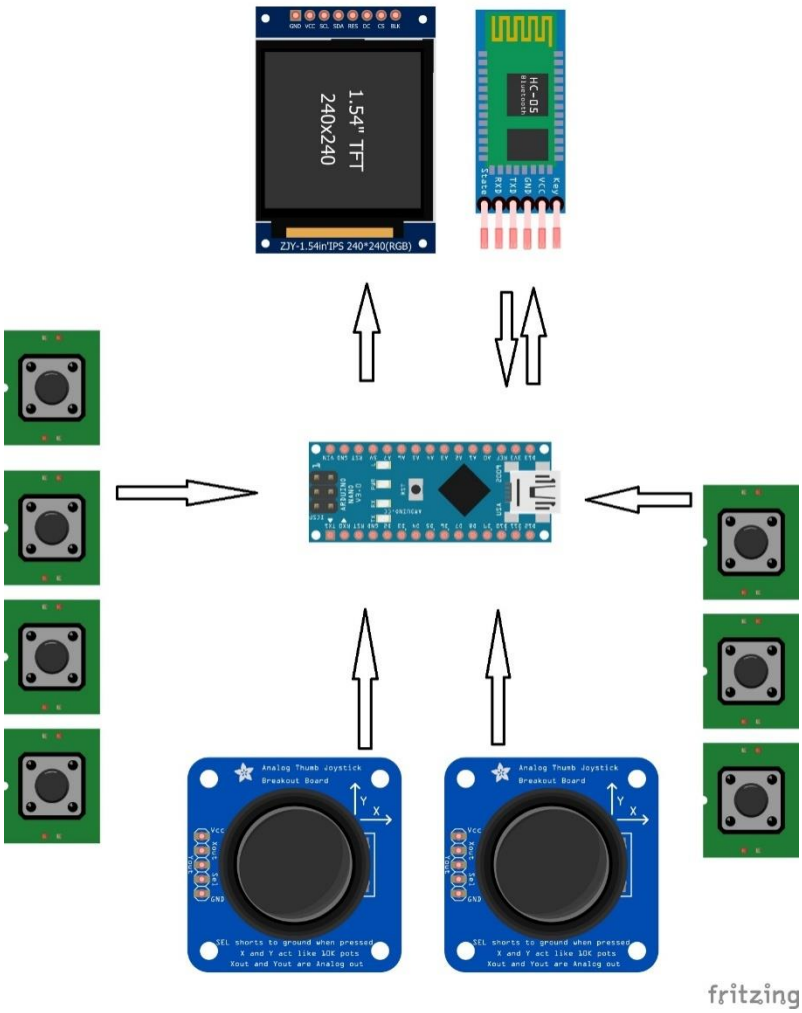
Slika 7 : Raspored pinova Arduino Nano razvojne pločice [6]

Arduino nano pločica ima 22 I/O pinova (ulaznih/izlaznih pinova), te se svih 22 može koristiti kao digitalni I/O pinovi. Od tih 22 se može 8 koristiti kao analogni pinovi(D14-D21), 2 pina se mogu koristiti za hardware-sku serijsku komunikaciju(D0, D1), te 3 pina za SPI komunikaciju(D11 – MOSI, D12 – MISO, D13 – SCK). Pločica sadrži i pinove za napajanje perifernih komponenata s dostupnim naponima 5 V i 3.3 V (maksimalna dozvoljena struja 50 mA). Sadrži i pinove za dodavanje vanjskog tipkala za reset te pinova za spajanje mase perifernih komponenata. Pinovi koji mogu generirati PWM (Pulse Width Modulation) signal su D3, D5, D6, D9, D10, D11.

3. PROJEKTIRANJE I IZVEDBA SKLOPOVSKOG SUSTAVA

3.1. Odabir komponenti sklopovskog sustava upravljaca

Modul za daljinsko upravljanje robotima se općenito sastoji od nekih ulaznih signala koji se uz eventualnu obradu šalju putem daljinske komunikacije prijemniku robota. U ovom projektu je odabran HC-05 prijemno-predajni Bluetooth modul kao komponenta koja će izvršavati bežičnu daljinsku komunikaciju koristeći Bluetooth (u nastavku teksta BT) protokol, za analogne ulaze su odabrane dvije takozvane „gljive“ te 6 tipkala u svrhu digitalnih ulaza, a za upravljačku jedinicu je odabran Arduino Nano . Radi prikaza potrebnih informacija je dodan i 1.3 inčni display. Navedene komponente su odabrane zbog svoje prihvatljive cijene, relativno dobre dostupnosti dokumentacije te svoje međusobne kompatibilnosti.



Slika 8 : Simbolički prikaz odabranih komponenata

3.2. HC-05 BlueTooth modul

HC-05 je modul koji podržava serijsku BT komunikaciju, podržava brzinu BT komunikacije do 3 Mbps. Modul podržava serijsko UART sučelje s programski podesivom brzinom komunikacije [7]. Najčešće dostupna varijanta HC-05 modula dolazi u varijanti s pločicom sa 6 priključnih pinova [8]:

1 : EN – pin na kojeg se isključivo dovođenjem logičke nule isključuje rad modula
2 : Vcc – pin za pozitivan pol napaja, minimalan dozvoljen napon 3.6 V,
maksimalan dozvoljeni napon 6 V
3 : Gnd – pin za masu napajanja
4 : Tx – pin na koji se serijski ispisuju primljene informacije preko BT veze
5 : Rx – pin na koji se šalju informacije kako bi se poslale preko BT veze
6 : State – pin koji je povezan s LED diodom koja indicira stanje BT modula

Slika 9 : HC-05 modul na pločici [9]

Pinovi pločice, odnosno pinovi samog modula (na slici 2 pločica zelene boje) rade na logičkim razinama napona od 3.3 V. Izravnim priključenjem standardnog napona Arduina u razini od 5 V može dovesti do oštećenja HC-05 modula.

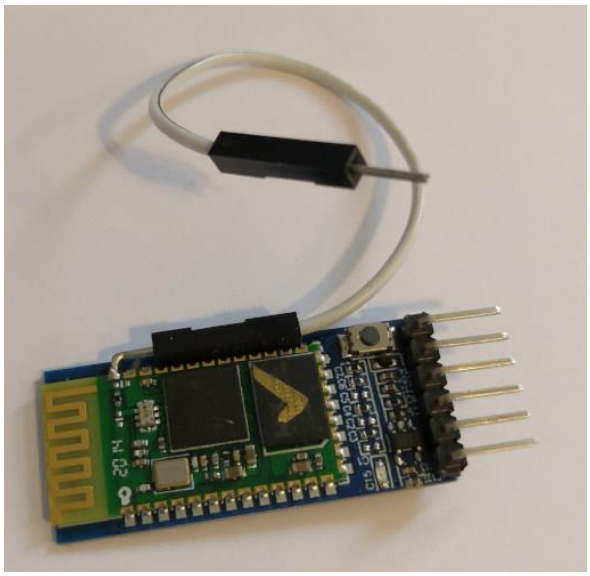
Pločica također sadrži LED diodu koja trepereći određenim uzorkom prikazuje u kojem se stanju rada modul nalazi [8]:

- Paljenje LED diode periodički svake 2 sekunde kada se modul nalazi u modu za AT+ naredbe na brzini serijske komunikacije 38400 bit-ova u sekundi
- Uzastopno treperenje diode označava kako se BT modul nije još povezao sa drugim uređajem
- Periodički svake 4 sekunde dvostruko uzastopno paljenje i gašenje LED diode označava kako je uspostavljena BT veza

Pločica sadrži tipkalo koje nakon pritiska povezuje 3.3 V s regulatora napona na pločici sa PIO11 (Programmable Input Output) [7], odnosno Pin34 modula, koji dovođenjem logičke jedinice omogućava slanje AT+ naredbi putem serijske komunikacije.

3.2.1. Modifikacija pločice HC-05 modula

AT+ naredbe su izrazito važne i nužne za konfiguriranje modula, te za manipulaciju BT veze (uparivanje s drugim BT uređajem, naredbe za prekid veze, traženje dostupnih uređaja). Zbog slabe praktičnosti tvornički predviđenog načina slanja AT+ naredbi gdje je nužno držati tipkalo pritisnutim u trenutku slanja i izvršenja AT+ naredbe je izvršena modifikacija pločice u obliku dodavanju novog pin-a. Modifikacija je izvršena tako da se kabel s prikladnim konektorom zalemio na Pin34 HC-05 modula. Tako je omogućeno izvršavanje AT+ naredbi bez fizičkog kontakta tipkala.



Slika 10 : Modifikacija HC-05 pločice

3.2.2. Shema spajanja HC-05 pločice na Arduino Nano

HC-05 je spojen na napon napajanja 5V, EN (Key) i Pin 34 su preko svojih otpornih naponskih djelila povezani na digitalne pinove Arduina. Otporna naponska djelila su postavljena zato što Arduino kao izlazni napon daje 5V, a HC-05 za ispravan rad zahtijeva logičke razine na 3.3V, te zbog malog iznosa potrebne struje potrošača je otporno naponsko djelilo bilo najjednostavnije i najpovoljnije rješenje.

Ohmov zakon govori kako je struja proporcionalna naponu, a obrnuto proporcionalna otporu:

$$I = \frac{U}{R}$$

U serijskom spoju, kroz otpornike teče ista struja, te uvrštavanjem Ohmovog zakona dobiva se jednadžba:

$$\frac{U}{R_1 + R_2} = \frac{U_2}{R_2}$$

Sređivanjem izraza se dobiva:

$$R_2 = \frac{U_2 R_1}{U - U_2}$$

S otporom trošila većim od 250 Ω je zadovoljen uvjet maksimalne dopuštene struje od 20 mA koju može pružiti Arduino Nano na izlazu pin-a. Radi manje parazitske struje bira se veća vrijednost od minimalne otpora naponskog djelila R_1 = 1 kΩ.

Uvrštavanjem vrijednosti napona U = 5 V, U_2 = 3.3 V u izraz i na osnovu odabira R_1 = 1 kΩ se izračunava vrijednost drugog otpora naponskog djelila:

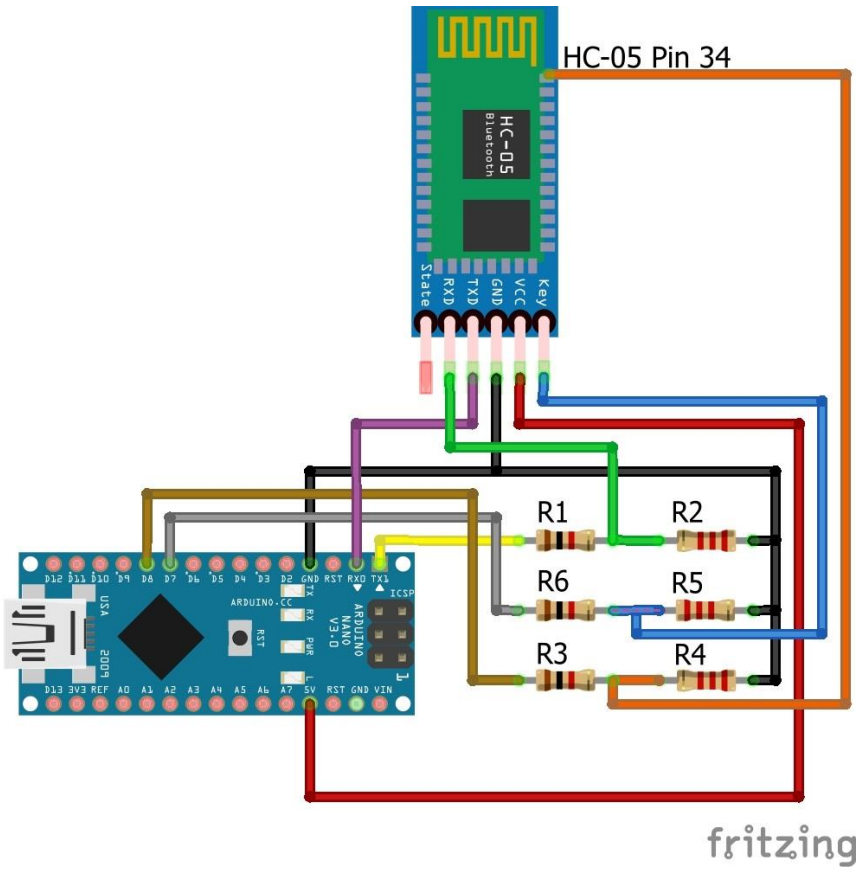
$$R_2 = \frac{3.3\ \mathrm{V} * \ 1\ \mathrm{k\Omega}}{5\mathrm{V} - 3.3\ V} = 1.9412\ \mathrm{k\Omega}$$

Zbog standardne dostupnosti komponenata se odabire otpor vrijednosti R_2 = 2.2 kΩ, koji će dati neznatno veći pad napona, ali i dalje unutar dozvoljenih granica napona HC-05 modula.

Navedena naponska djelila su potrebna za pinove Pin34 (dodani pin za AT+ naredbe), EN, Rx. Za Tx pin nije potrebno naponsko djelilo, odnosno nepotrebno je jer Tx pin služi za slanje 3.3 V signala iz HC-05 modula prema Arduino Nano. Rx pin Arduina prihvaća i logičke razine signala pri 3.3 V, iako je primarno za signale na razini 5 V. Također Tx HC-05 pločice je zaštićen diodom od protoka struje u suprotnom smjeru.

Na Arduino Nano pločici se za korištenje HC-05 pločice koriste Tx i Rx hardware serijskog porta Arduina. Iako je moguće koristiti druge digitalne pinove Arduina u svrhu serijske komunikacije pomoću software-ski emuliranog serijskog porta, koristi se hardware-ski serijski port radi smanjivanja broja korištenih pinova Arduina. No prilikom programiranja Arduina treba s pažnjom pristupiti tome jer se hardware-ski port koristi za upisivanje programa u mikrokontroler preko USB sabirnice, te s krivim postupanjem može doći do ne mogućnosti upisivanja programa zbog zauzetosti serijske sabirnice.

Također se Tx i Rx koristi za Serial Monitor Arduino IDE-a, te ako se želi ispisati nešto na Serial Monitoru, to isto se šalje preko BT komunikacije.

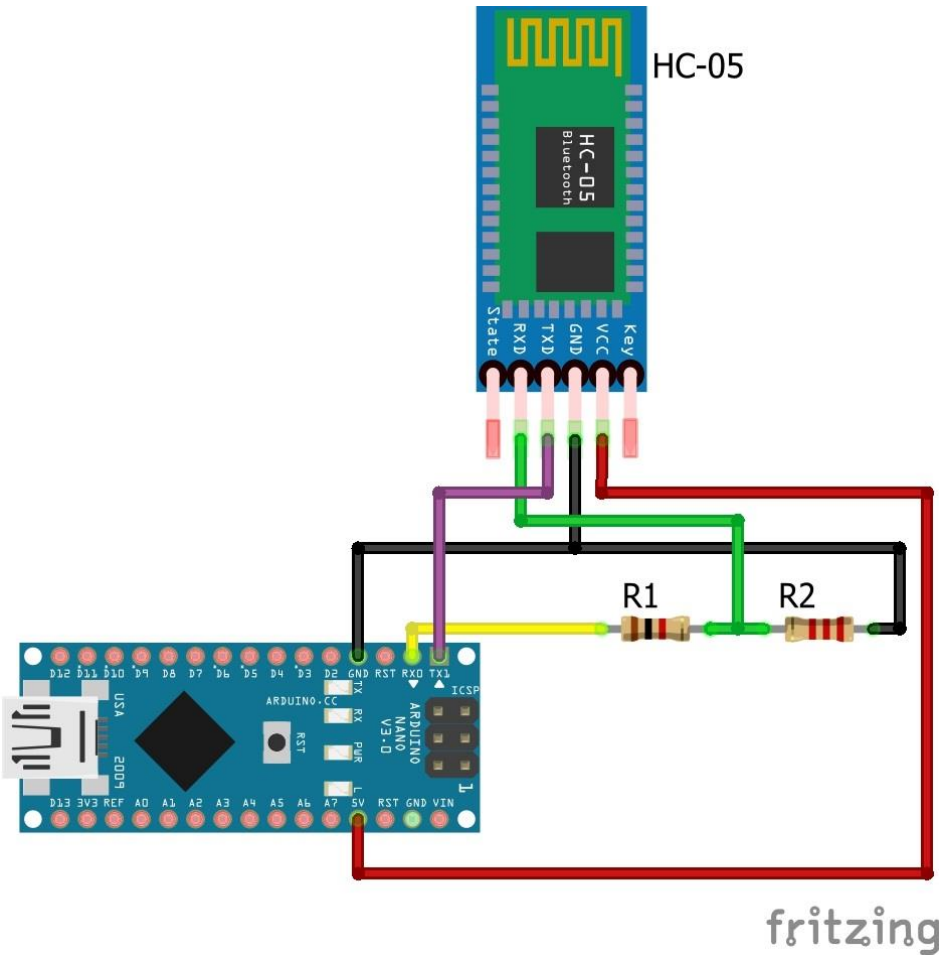


Slika 11 : Shema spajanja HC-05 pločice

Dva pina Arduino Nano se koristi u svrhu upravljanja HC-05 modulom. Jedan pin za promjenu načina rada između BT komunikacijskog moda i moda za upisivanje AT+ naredbi (D8 pin), te pin za paljenje, odnosno isključivanje HC-05 modula (D7 pin).

3.2.3. Shema spajanja HC-05 prilikom konfiguriranja putem Serial Monitor-a

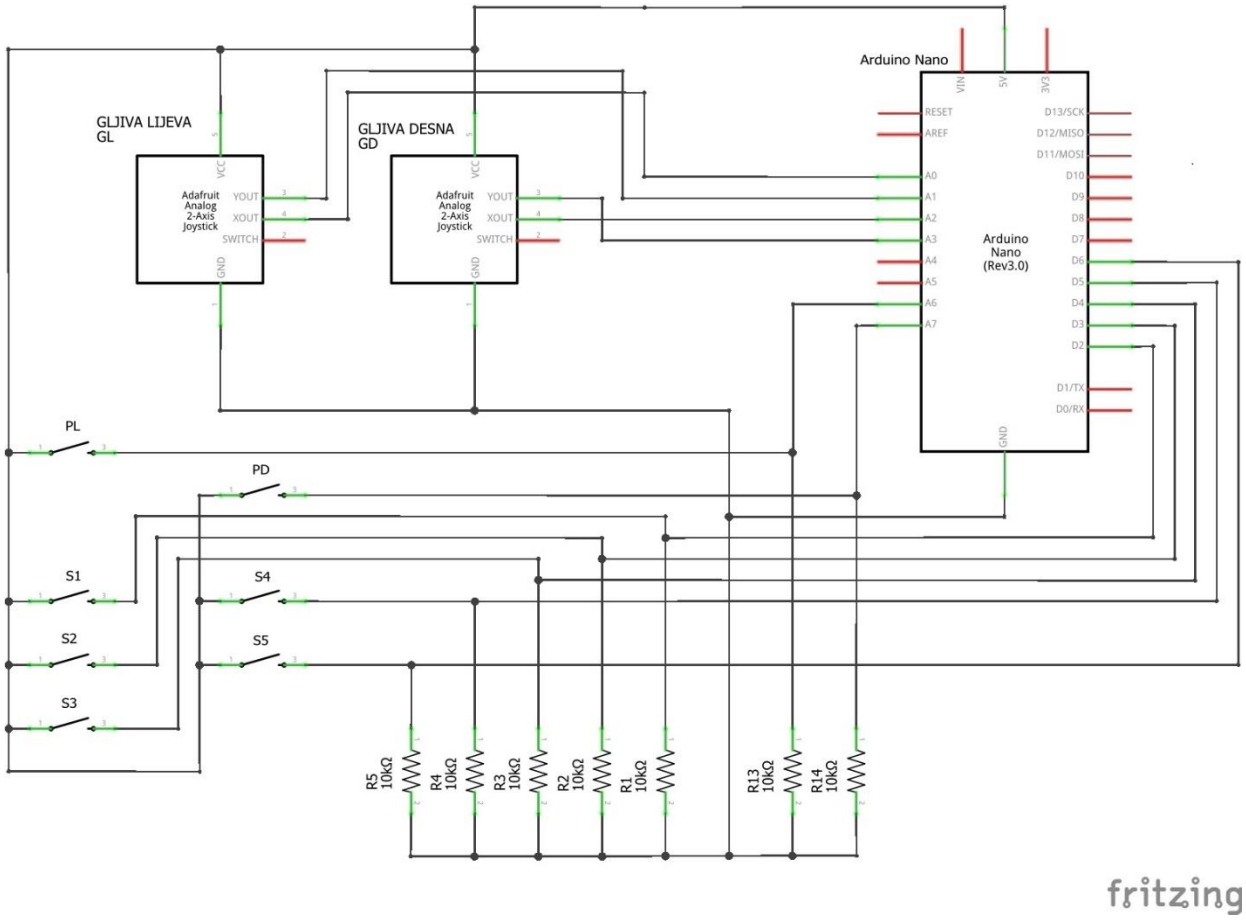
Za konfiguriranje HC-05 modula koristeći AT+ naredbe, tako da se one upisuju ručno preko Serial Monitor-a Arduino IDE-a, potrebno je spojiti Rx pin HC-05 pločice preko naponskog djelila na Rx pin Arduino Nano-a, a Tx pin HC-05 pločice na Tx Arduina. Na taj se način spaja jer informacije poslane iz Serial Monitora preko pretvarača dolaze do Rx pina Arduina (informacija se prima), a budući da informacije koje se šalju u HC-05, se šalju preko Rx pina HC-05 pločice, potrebno je povezati Rx pin HC-05 pločice na Rx pin Arduina gdje se primarno šalje poruka sa Serial Monitor-a, može se reći da se na određeni način presreće potrebna informacija.



Slika 12 : Shema spajanja HC-05 pločice za konfiguriranje

3.3. Digitalni i analogni ulazi

Za ulaze kojima će se upravljati robotom putem BT veze koristi se 6 tipkala te dvije takozvane „gljive“. Na sva tipkala s izlazne strane prema masi su spojeni pull-down otpornici otpora od 10 kΩ. Pull-down otpornici se koriste kako bi se u otpuštenom stanju tipkala osigurala povezanost ulaza Arduina s masom, odnosno logičkom nulom. Tipkalo S5 je projektirano za promjenu prikaza na 1.3 inčnom ekranu i ostalim podešavanjima.



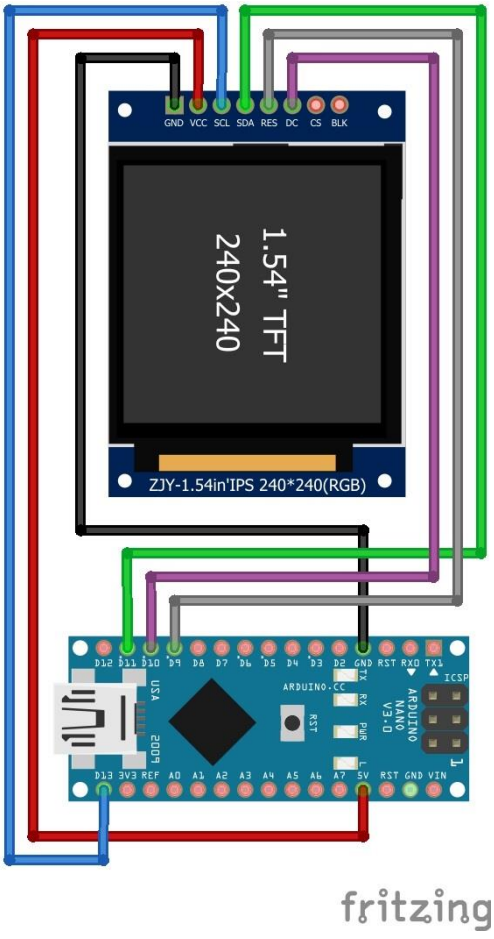
Slika 13 : Električna shema spajanja ulaza

3.4. LCD display

Svrha LCD display-a u ovom radu je prikaz informacija o vrijednosti ulaza, informacijama o povezanosti BT modula, ispisu dostupnih uređaja, prikaz odabira istih. Korišteni display ima rezoluciju 240 x 240 piksela na ekranu dijagonale 1.3 inča. LCD dobiva informacije preko serijske UART sabirnice, pritom zauzima 6 digitalnih pinova Arduina. Iako display ima 7 pinova, korištenjem prilagođenog Arduino library-a [9] je moguće koristiti samo 6, umjesto svih 7 pinova što pomaže pri smanjenju ukupne upotrebe količine ulaza. Iako je u dostupnim informacijama [9] o display-u navedeno da se koristi 3.3 V napajanje, testiranjem je utvrđeno da je potrebno 5 V napajanje jer pri napajanju od 3.3 V prikaz na display-u je zamućen te dolazi do titranja pozadinskog osvjetljenja.

Slika 14 : 1.3 inčni LCD display [10]

Ekran se spaja tako da se Gnd ekrana priključuje na Gnd Arduina, Vcc priključuje na 5V napajanje (iako je na samoj pločici ekrana navedeno 3.3 V), SDA na D13 (SCK), SDA na D11 (MOSI), RES i DC mogu se spojiti na bilo koji digitalan pin, a odabrani su RES na D9, DC Na D10. BLK se ne spaja.



Slika 15 : Shema spajanja LCD display-a na Arduino Nano

4. IZVEDBA PROGRAMSKOG SUSTAVA

4.1. Konfiguriranje HC-05 modula koristeći AT+ naredbe

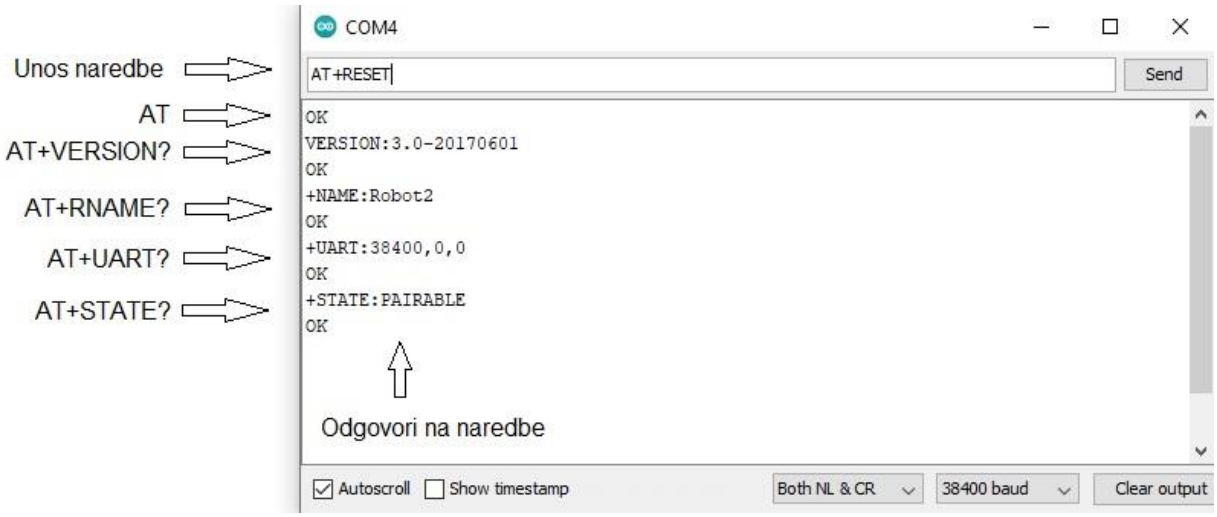
Naredba	**Opis naredbe**
AT	Provjera ispravnosti serijske veze, odgovara s „OK“
AT+RESET	Služi za ponovno pokretanje HC-05 modula, odgovara s „OK“
AT+VERSION?	Pruža informaciju o verziji Firmware-a HC-05 modula, odgovara s „VERSION:3.0-20170601“
AT+ORGL	Za postavljanje HC-05 modula na tvorničke postavke, odgovara s „OK“
AT+ADDR?	Pruža informaciju o vlastitoj adresi HC-05 modula, odgovara u obliku „+ADDR:1234:56:ABCDEF“
AT+NAME? AT+NAME=Robot1	Pruža informaciju o trenutno postavljenom nazivu uređaja ili postavlja naziv na zadano
AT+RNAME?Adresa	Dohvaća naziv zadanog dostupnog uređaja na osnovu adrese. Adresa mora biti oblika 0000,00,000000
AT+PSWD? AT+PSWD=1234	Pruža informaciju o BT lozinci ili postavlja lozinku na zadanu
AT+UART? AT+UART=BaudRate,StopBit,Paritet	Pruža informacije o serijskoj komunikacije ili ju postavlja na zadane vrijednosti
AT+BIND=Adresa	Veže HC-05 s zadanom adresom
AT+STATE?	Vraća informaciju o trenutnom statusu povezanosti HC-05 modula
AT+INQ	Nakon poziva funkcije dobiva informacije o dostupnosti drugih uređaja. Problematična naredba jer se izvršava u beskonačnost

Tablica 2 :Važnije AT+ naredbe [7]

AT+PAIR=Adresa,TimeOut	Uparuje HC-05 modul s uređajem na navednoj adresi. Nakon uparivanja se BT uređaji automatski povezuju
AT+DISC	Isključuje uspostavljenu vezu među BT uređajima

Tablica 3 : Važnije AT+ naredbe, nastavak Tablice 2 [7]

Putem Serial Monitora potrebno je inicijalno podesiti HC-05 modul na željeni način rada. Preimenovanje modula iz HC-05 u Robot2 je izvršeno unosom naredbe AT+NAME=Robot2, te je brzina serijske komunikacije promijenjena sa 9600 bitova u sekundi na 38400 bitova u sekundi naredbom AT+UART=38400,0,0 . Ukoliko nije izvršena modifikacija na pločici HC-05 kako je opisano u poglavlju 3.2.1. potrebno je držati tipkalo na pločici pritisnutim za vrijeme upisa naredbe.



Slika 16 : Primjer unosa AT naredbe putem Serial Monitor-a

4.2. Korisničko sučelje modula za daljinsko upravljanje robotima

Korisničko sučelje modula za daljinsko upravljanje prikazuje se na LCD-u. Prijelaz iz jednog mod-a u drugi se obavlja tipkalom S5. U komunikacijskom modu na ekranu se u realnom vremenu ispisuje stanje HC-05 modula(STATE: PAIRABLE, STATE: PAIRED, STATE: CONNECTING, STATE: CONNECTED, odnosno očitan odgovor naredbe AT+STATE?), nakon povezivanja ime, odnosno naziv spojenog BT uređaja, te ovisno o varijabli select nazivi i vrijednosti odabranih ulaza.

Slika 17 : Izgled komunikacijskog moda na LCD-u

U konfiguracijskom modu se ispisuju nazivi dostupnih uređaja ili adresa ako nije uspješno očitan naziv, te se pomoću tipkala S3 i S4 pomiče kursor za odabir, a potvrda odabranog uređaja se vrši pritiskom na tipkalo S5. U zaglavlju oba moda se nalaze nazivi ulaza kojima se u tom modu izvršavaju promjene. Kada je prihvaćen ulaz s tipkala S5 u oba moda, pozadina S5 na LCD-u mijenja boju u bijelu, a znakovi u crnu boju kako bi se dalo do znanja da je ulaz prihvaćen.

Slika 18 : Izgled konfiguracijskog moda na LCD-u

4.3. Programsko rješenje BT upravljača

```
/*Program je pisan za verziju firmware-a 3.0-20170601 HC-05 modula*/
/*BlueTooth, skraćeno BT*/
```

```
/*Upotrijebljeno radi uštede jednog pina LCD-a:
// Fast ST7789 IPS 240x240 SPI display library
// (c) 2019-20 by Pawel A. Hernik*/

/*LCD EKRAN*/
#define TFT_DC    10 //definiranje pinova koji se koriste za 1.3 incni LCD
#define TFT_RST   9
#define SCR_WD   240 //definiranje rezolucije LCD-a
#define SCR_HT   240
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Arduino_ST7789_Fast.h>
Arduino_ST7789 lcd = Arduino_ST7789(TFT_DC, TFT_RST);

/*Varijable za privremenu pohranu očitanja kod serijske komunikacije*/
char c[30];
int i;
int ef = 0; /*error flag, 0-ispravna veza, 1-neispravna veza*/
int con = 0; //indikator povezanosti s drugim modulom
int conctrl = 0; //pomocni indikator povezanosti s drugim modulom
String BTadresa[6];//za pohranu dostupnih adresa
String BTnaziv[6];//za pohranu naziva dostupnih uređaja
String BTstate;//stanje BT veze(odgovor od AT+STATE?)
String pomocnistring = "";
char pomocnichar;
int stringbrojac = 0; //Koristi se prilikom očitanja dobivene BT informacije
int strtimpoc = 0;
int strtimkrj = 0;
int strtim = 0;
int pingcon = 0;//indikator povezanosti na osnovi provjere putem PING-a
int indeks = 0;//pokazuje na indeks polja koje sadrži trenutno odabranu
adresu
int pingtim = 0;//timer za PING
char buffer[5];   /*Array za privremenu pohranu broja u obliku stringa*/

/*za timer*/
unsigned long StartTime = 0;
unsigned long CurrentTime = 0;
unsigned long ElapsedTime = 0;

/*za mod-ove, načine rada(0 = standardna BT komunikacija-automatski se spaja
na zadnje spojeni uređaj, 1 = mod za traženje i odabir dostupnih BT
uređaja)*/
int mod = 0;
int modpr = -1;//postavljeno -1 kako bi se uspostavila inicijalna promje moda
int S5pr = 0;//prijasnje stanje tipkala S5

//Select : za željeni odabir ulaza zbrajaju se heksadekadske vrijednosti
dodijeljene ulazima
//0x400 0x200 0x100 0x80  0x40  0x20  0x8 0x4 0x2 0x1
//GLX   GLY   GDX   GDY   PL    PD    S1  S2  S3  S4
int select = 0x07FF;/*Za Default očitanje, svi ulazi se očitavaju*/

//Pridruživanje varijable fizičkom ulazu
/*analogni ulazi*/
int GLX = A0;
int GLY = A1;
int GDX = A2;
int GDY = A3;
int PL = A6;
int PD = A7;
/*tipkala ulazi*/
int S1 = 2;
```

```
int S2 = 3;
int S3 = 4;
int S4 = 5;
int S5 = 6;

/*varijable za pohranu vrijednosti ulaza,navedeno inicijalno stanje*/
int GLXval = 0;
int GLYval = 0;
int GDXval = 0;
int GDYval = 0;
int PLval = 0;
int PDval = 0;
int S1val = 0;
int S2val = 0;
int S3val = 0;
int S4val = 0;
int S5val = 0;
int GLXvalpr = -1;
int GLYvalpr = -1;
int GDXvalpr = -1;
int GDYvalpr = -1;
int PLvalpr = -1;
int PDvalpr = -1;
int S1valpr = -1;
int S2valpr = -1;
int S3valpr = -1;
int S4valpr = -1;
int S5valpr = -1;

/*************************/
/***Funkcija za čekanje***/
/*************************/
int cekaj(int a) { //Funkcija koja se koristi umjesto delay() funkcije, jer
je s porastom količine programskog koda delay() postao nestabilniji
  int poc;
  int kraj;
  int vrijeme = 0;
  poc = millis();//zapiši trenutno vrijeme u poc
  while (vrijeme <= a) {//zadržava se u petlji dok ne prođe zadano vrijeme
    kraj = millis();
    vrijeme = kraj - poc;
  }
}

/*******************************/
/***Moja funkcija za ocitanje***/
/*******************************/
void citaj(int a) {//Funkcija koja u argument prima "select"
  /*očitavanje potenciometra*/
  if ( (a & 0x0400) != 0) {//provođenje AND operacije nad pojedinačnim
bitovima heksadekadskog broja
    GLXval = analogRead(GLX);
    GLXval = (unsigned int)GLXval >> 4;/*Pomaknuti bitove za 4 mjesta u desno
i nadopisati nule na početak radi smanjenja rezolucije, koristi se radi
smanjenja osjetljivosti ulaza pri mirovanju*/
  }
  if ( (a & 0x0200) != 0) {
    GLYval = analogRead(GLY);
    GLYval = (unsigned int)GLYval >> 4;
  }
  if ( (a & 0x0100) != 0) {
    GDXval = analogRead(GDX);
    GDXval = (unsigned int)GDXval >> 4;
```

```
  }
  if ( (a & 0x0080) != 0) {
    GDYval = analogRead(GDY);
    GDYval = (unsigned int)GDYval >> 4;
  }
  if ( (a & 0x0040) != 0) {
    PLval = analogRead(PL);
    PLval = (unsigned int)PLval >> 4;
  }
  if ( (a & 0x0020) != 0) {
    PDval = analogRead(PD);
    PDval = (unsigned int)PDval >> 4;
  }
  /*ocitavanje tipkala*/
  if ( (a & 0x0008) != 0) {
    S1val = digitalRead(S1);
  }
  if ( (a & 0x0004) != 0) {
    S2val = digitalRead(S2);
  }
  if ( (a & 0x0002) != 0) {
    S3val = digitalRead(S3);
  }
  if ( (a & 0x0001) != 0) {
    S4val = digitalRead(S4);
  }
  S5val = digitalRead(S5);

  if (S5val == 1) {//Služi za prikaz na LCD-u prilikom pritiska tipkala S5
    lcd.setTextColor(BLACK, WHITE);//crna slova, bijela pozadina teksta
    lcd.setCursor(176, 0);//postavi cursor na LCD-u na zadano mjesto
    lcd.setTextSize(1);//velicina teksta 1
    if (mod == 0) {//ispis u komunikacijskom modu
      lcd.print(F("mod "));
    }
    if (mod == 1 ) {//ispis u konfiguracijskom modu
      lcd.print(F(" ok "));
    }
    lcd.setTextSize(2);
    lcd.print(F("S5"));//Pobijeli pozadinu S5 u zaglavlju LCD-a
    while (digitalRead(S5) == 1);//pričekaj dok se tipkalo ne pusti
    cekaj(10);
    if (mod == 0) {//ovisno o trenutnom modu promijeni vrijednsot varijable
mod
      mod = 1;
    }/*Ulazak u BT konifg*/
    else if (mod == 1) {
      mod = 0;
    }/*Ulazak u BT kom mod*/
  }
  lcd.setTextColor(WHITE, BLACK);//vrati boju teksta na bijelu s crnom
pozadinom
}

/*****************************/
/***Moja funkcija za prikaz***/
/*****************************/
void lcdprikaz(char *niz) { //Funkcija koja ispisuje zadani niz znak po znak
s očitavanjem između ispisa znakova, kako bi se poboljšala brzini očitanja
  while (*niz) {
    lcd.print(*niz);
    niz++;
```

```
    citaj(select);//nakon ispisa jednog znaka(potrebno otprilike 2 ms za
ispis jednog znaka na LCD) obavi čitanje ulaza
  }
}

/*********************************************/
/***Moja funkcija za ispis fiksnog dijela info-a***/
/*********************************************/
void UlaziInfoS(int a, int b) { //Funkcija služi tome da se statički dio
ispisa na ekranu ispiše samo jednom, kako bi se uštedjelo na vremenu
odvijanja cijele petlje programa
//Za argument a prima mod, za argument b prima select
  int pom1 = 40;//varijabla za pomicanje cursora
  lcd.fillScreen(BLACK);//ispuni cijeli ekran crnom bojom kako bi se
prebrisao prijašnji sadržaj

  if (a == 0) {//ispis zaglavlja na LCD-u za komunikacijski mod
    lcd.setTextColor(WHITE, BLACK);
    lcd.setCursor(176, 0);
    lcd.setTextSize(1);
    lcd.print(F("mod "));
    lcd.setTextSize(2);
    lcd.print(F("S5"));
    lcd.setCursor(0, 16);
    lcd.setTextSize(3);
    lcd.setTextColor(GREEN, LGREY);
    lcd.print(F("BT komunik.  "));
    lcd.setTextSize(2);
    pom1 = pom1 + 32; /*Ostavi mjesto za ispis statusa HC-05*/
  }
  if (a == 1) {//ispis zaglavlja za konfiguracijski mod
    lcd.setTextColor(WHITE, BLACK);
    lcd.setTextSize(2);
    lcd.setCursor(0, 0);
    lcd.print(F("-"));
    lcd.print(F("S3 "));
    lcd.print(F("+"));
    lcd.print(F("S4"));
    lcd.setCursor(80, 0);
    lcd.print(F("|"));
    lcd.setCursor(160, 0);
    lcd.print(F("|"));
    lcd.setCursor(176, 0);
    lcd.setTextSize(1);
    lcd.print(F(" ok "));
    lcd.setTextSize(2);
    lcd.print(F("S5"));
    lcd.setCursor(0, 16);
    lcd.setTextSize(3);
    lcd.setTextColor(RED, DGREY);
    lcd.print(F("BT konfig.   "));
  }

  lcd.setTextSize(2);
  //ispis pojedinih naziva ulaza ovisno o select-u
  if ( (b & 0x0600) != 0) {
    lcd.setCursor(0, pom1);
    lcd.setTextColor(YELLOW, BLACK);
    lcd.print(F("GL "));
    lcd.setTextColor(WHITE, BLACK);
    if ( (b & 0x0400) != 0) {
      lcd.print(F("X:"));
    }
```

```
    if ( (b & 0x0200) != 0) {
      lcd.setCursor(144, pom1);
      lcd.print(F("Y:"));
    }
    pom1 = pom1 + 16;
  }
  if ( (b & 0x0180) != 0) {
    lcd.setCursor(0, pom1);
    lcd.setTextColor(YELLOW, BLACK);
    lcd.print(F("GD "));
    lcd.setTextColor(WHITE, BLACK);
    if ( (b & 0x0100) != 0) {
      lcd.print(F("X:"));
    }
    if ( (b & 0x0080) != 0) {
      lcd.setCursor(144, pom1);
      lcd.print(F("Y:"));
    }
    pom1 = pom1 + 16;
  }
  lcd.setTextColor(YELLOW, BLACK);
  if ( (b & 0x0040) != 0) {
    lcd.setCursor(0, pom1);
    lcd.setTextColor(YELLOW, BLACK);
    lcd.print(F("PL"));
    pom1 = pom1 + 16;
  }
  if ( (b & 0x0020) != 0) {
    lcd.setCursor(0, pom1);
    lcd.setTextColor(YELLOW, BLACK);
    lcd.print(F("PD"));
    pom1 = pom1 + 16;
  }
  if ( (b & 0x0008) != 0) {
    lcd.setCursor(0, pom1);
    lcd.setTextColor(YELLOW, BLACK);
    lcd.print(F("S1"));
    pom1 = pom1 + 16;
  }
  if ( (b & 0x0004) != 0) {
    lcd.setCursor(0, pom1);
    lcd.setTextColor(YELLOW, BLACK);
    lcd.print(F("S2"));
    pom1 = pom1 + 16;
  }
  if ( (b & 0x0002) != 0) {
    lcd.setCursor(0, pom1);
    lcd.setTextColor(YELLOW, BLACK);
    lcd.print(F("S3"));
    pom1 = pom1 + 16;
  }
  if ( (b & 0x0001) != 0) {
    lcd.setCursor(0, pom1);
    lcd.setTextColor(YELLOW, BLACK);
    lcd.print(F("S4"));
    pom1 = pom1 + 16;
  }
  lcd.setTextColor(WHITE, BLACK);
  GLXvalpr = -1;   /*Za ispravan ispis vrijednosti ulaza nakon promjene mod-
a(reset ulaza)*/
  GLYvalpr = -1;
  GDXvalpr = -1;
  GDYvalpr = -1;
```

```
  PLvalpr = -1;
  PDvalpr = -1;
  S1valpr = -1;
  S2valpr = -1;
  S3valpr = -1;
  S4valpr = -1;
  S5valpr = -1;
}

/************************************/
/***Moja funkcija za promjenu mod-a***/
/************************************/
void PromjenaModa(int a) { //Funkcija koja nakon detektiranja promjene
varijable mod izvršava inicijalne radnje za promjenu mod-a
  char pom[64];
  int j;
  if (a == 0) { /*Potrebne radnje za BT komunikacijski mod*/
    BTadresa[0] = BTadresa[indeks];//Postavi odabranu adresu
    UlaziInfoS(a, select);//ispiši nazive korištenih varijabli
    cekaj(150);
    cekaj(1000);
    indeks = 0;
    digitalWrite(7, LOW);//isključi HC-05 modul, zatim isključi pin34
    digitalWrite(8, LOW);/*Kako bi se spriječilo stavljanje HC-05 u
konfiguracijski mod 2 na baud rate 38400*/
    cekaj(500);
    digitalWrite(7, HIGH);//uključi HC-05 modul
    cekaj(500);
    con = 0;//reset indikatora i zastavica
    ef = 0;
  }
  if (a == 1) {//Inicijalne radnje za konfiguracijski mod
    UlaziInfoS(a, 0x0000);
    digitalWrite(7, LOW);//Ponovno pokretanje HC-05 modula
    cekaj(150);
    digitalWrite(7, HIGH);
    cekaj(1000);
    digitalWrite(8, LOW);
    cekaj(500);
    digitalWrite(8, HIGH);//Pin34 postavlja se na jedinicu za pisanje AT
naredbi
    cekaj(500);
    lcd.setTextSize(2);
    lcd.setCursor(20, 40);
    lcd.setTextColor(WHITE, BLACK);
    lcd.print("...Trazenje...");
    cekaj(20);
    PretraziBT();//Poziv funkcije za traženje dostupnih BT uređaja
    if (modpr != mod) {//Ukoliko se pritisne tipkalo S5 prekida se procedura
i mijenja se mod
      return;
    }
    cekaj(100);
    ImenaBT();
    if (modpr != mod) {//Ukoliko se pritisne tipkalo S5 prekida se procedura
i mijenja se mod
      return;
    }
    cekaj(100);
    cekaj(100);
    digitalWrite(7, LOW);//Nakon obavljene pretrage se modul isključuje
    digitalWrite(8, LOW);
    indeks = 1;//Inicijalna vrijednost indeksa na osnovu pokazivača na LCD-u
```

```
    cekaj(100);
  }
}

/*****************************************/
/***Moje funkcije za ispis varijabli info-a***/
/*****************************************/
//Pozivaju se po potrebi
void UlaziInfoGLX(int a) { //Funkcije za ispis pojednih vrijednosti odabranih
ulaza
  /*GLXval*/
  lcd.setCursor(60, a);
  /*   lcd.setTextColor(WHITE,BLACK);   /* -->Promijeni boju slova u bijelu i
boju pozadine slova u crnu*/
  sprintf(buffer, "%d", GLXval);   /*  -->Pretvori vrijednost integer-a u
string kako bi se mogao ispisati*/
  lcdprikaz(buffer);               /* -->Ispiši string, 0dnosno vrijednost
integera GLXval*/
  lcdprikaz("  ");                 /* -->Prebriši iduca dva polja od
zaostalih znakova*/
  cekaj(10);
}

void UlaziInfoGLY(int a) {
  /*GLYval*/
  lcd.setCursor(168, a);
  sprintf(buffer, "%d", GLYval);
  lcdprikaz(buffer);
  lcdprikaz("  ");
  cekaj(10);
}

void UlaziInfoGDX(int a) {
  /*GDXval*/
  lcd.setCursor(60, a);
  sprintf(buffer, "%d", GDXval);
  lcdprikaz(buffer);
  lcdprikaz("  ");
  cekaj(10);
}

void UlaziInfoGDY(int a) {
  /*GDYval*/
  lcd.setCursor(168, a);
  sprintf(buffer, "%d", GDYval);
  lcdprikaz(buffer);
  lcdprikaz("  ");
  cekaj(10);
}

void UlaziInfoPL(int a) {
  /*PLval*/
  lcd.setCursor(36, a);
  sprintf(buffer, "%d", PLval);
  lcdprikaz(buffer);
  lcdprikaz("  ");
  cekaj(10);
}

void UlaziInfoPD(int a) {
  /*PDval*/
  lcd.setCursor(36, a);
  sprintf(buffer, "%d", PDval);
```

```
  lcdprikaz(buffer);
  lcdprikaz("  ");
  cekaj(10);
}

void UlaziInfoS1(int a) {
  /*S1val*/
  lcd.setCursor(36, a);
  sprintf(buffer, "%d", S1val);
  lcdprikaz(buffer);
  lcdprikaz("  ");
  cekaj(10);
}

void UlaziInfoS2(int a) {
  /*S2val*/
  lcd.setCursor(36, a);
  sprintf(buffer, "%d", S2val);
  lcdprikaz(buffer);
  lcdprikaz("  ");
  cekaj(10);
}

void UlaziInfoS3(int a) {
  /*S3val*/
  lcd.setCursor(36, a);
  sprintf(buffer, "%d", S3val);
  lcdprikaz(buffer);
  lcdprikaz("  ");
  cekaj(10);
}

void UlaziInfoS4(int a) {
  /*S4val*/
  lcd.setCursor(36, a);
  sprintf(buffer, "%d", S4val);
  lcdprikaz(buffer);
  lcdprikaz("  ");
  cekaj(10);
}

void UlaziInfoS5(int a) {
  /*S5val*/
  lcd.setCursor(36, a);
  sprintf(buffer, "%d", S5val);
  lcdprikaz(buffer);
  lcdprikaz("  ");
  cekaj(10);
}

/*************************************************************************
****/
/***Moja funkcija za detektiranje promjene i ispis vrijednosti varijabli
ulaza***/
/*************************************************************************
****/
void UlaziDin(int a, int b) { //Funkcija koja će detektirati promjenu
vrijednosti ulaza, te ispisati odabrane
  int pom1 = 24;
  lcd.setTextSize(2);
  lcd.setTextColor(WHITE, BLACK);
  if (mod == 1) {//Radnje u konfiguracijskom modu, pomicanje pokazivača > na
LCD-u
```

```
    if ((!(BTadresa[indeks].equals(""))) && (indeks < 6)) {//Ako postoji
spremljeni podatak
      lcd.setCursor(0, (pom1 + (indeks * 16)));
      lcd.print(F(">"));
      if (S3val == 1) {//Pomicanje pokazivača > za jedno mjesto gore
korištenjem tipke S3
        while (digitalRead(S3) == 1);
        cekaj(10);
        if (indeks > 1) {
          lcd.setCursor(0, (pom1 + (indeks * 16)));
          lcd.print(F(" "));
          indeks--;
        }
      }
      if (S4val == 1) {//Pomicanje pokazivača > za jedno mjesto dolje
korištenjem tipke S4
        while (digitalRead(S4) == 1);
        cekaj(10);
        if (indeks < 5) {
          lcd.setCursor(0, (pom1 + (indeks * 16)));
          lcd.print(F(" "));
          indeks++;
        }
      }
    }
    else {//vrati pokazivač na početno mjesto
      indeks = 1;
    }
  }
  pom1 = 56;//pozicioniranje LCD cursora
  if ( (b & 0x0400) != 0) {
    pom1 = pom1 + 16;//pomakni cursor u novi red
    if (GLXval != GLXvalpr) { /*detektiranje promjene vrijednosti ulaza*/
      GLXvalpr = GLXval;//došlo je do promjene, resetiraj vrijednsot
      UlaziInfoGLX(pom1);
    }
  }
  if ( (b & 0x0200) != 0) {
    if (GLYval != GLYvalpr) {
      GLYvalpr = GLYval;
      UlaziInfoGLY(pom1);
    }
  }
  if ( (b & 0x0100) != 0) {
    pom1 = pom1 + 16;//pomakni cursor u novi red
    if (GDXval != GDXvalpr) {
      GDXvalpr = GDXval;
      UlaziInfoGDX(pom1);
    }
  }
  if ( (b & 0x0080) != 0) {
    if (GDYval != GDYvalpr) {
      GDYvalpr = GDYval;
      UlaziInfoGDY(pom1);
    }
  }
  if ( (b & 0x0040) != 0) {
    pom1 = pom1 + 16;
    if (PLval != PLvalpr) {
      PLvalpr = PLval;
      UlaziInfoPL(pom1);
    }
  }
```

```
  if ( (b & 0x0020) != 0) {
    pom1 = pom1 + 16;
    if (PDval != PDvalpr) {
      PDvalpr = PDval;
      UlaziInfoPD(pom1);
    }
  }
  if ( (b & 0x0008) != 0) {
    pom1 = pom1 + 16;
    if (S1val != S1valpr) {
      S1valpr = S1val;
      UlaziInfoS1(pom1);
    }
  }
  if ( (b & 0x0004) != 0) {
    pom1 = pom1 + 16;
    if (S2val != S2valpr) {
      S2valpr = S2val;
      UlaziInfoS2(pom1);
    }
  }
  if ( (b & 0x0002) != 0) {
    pom1 = pom1 + 16;
    if (S3val != S3valpr) {
      S3valpr = S3val;
      UlaziInfoS3(pom1);
    }
  }
  if ( (b & 0x0001) != 0) {
    pom1 = pom1 + 16;
    if (S4val != S4valpr) {
      S4valpr = S4val;
      UlaziInfoS4(pom1);
    }
  }
}

/*****************************************************************/
/***Moja funkcija za dohvat zadnje korištene adrese i imena***/
/*****************************************************************/
void ZadnjiBT() { //Funkcija koja dohvaća iz HC-05 pomoću AT+MRAD naredbe
zadnju korištenu adresu
  citaj(select);
  if (modpr != mod) {//Prekid funkcije pomoću S5 tipkala
    return;
  }
  indeks = 0;//na indeksu 0 polja stringa adrese se sprema trenutno korištena
adresa
  String pomstring = "";//Isprazni string
  digitalWrite(7, HIGH);//Uključivanje modula
  cekaj(1000);
  digitalWrite(8, HIGH);//Uključivanje pin34
  cekaj(100);
  if (BTadresa[indeks].equals("")) {
    SerialAT(F("AT+MRAD"));//Upiši naredbu koristeći funkciju SerialAT, Macro
naredba F() se koristi za štednju prostora radne memorije, sadržaj se čita iz
programske memorije
    citaj(select);
    if (modpr != mod) {//Prekid funkcije pomoću S5 tipkala
      return;
    }
  }
  BTnaziv[indeks] = "";//Isprazni string naziva
```

```
  cekaj(10);
  pomstring.concat(F("AT+RNAME?"));//Nadopiši u pomstring AT+RNAME?
  pomstring.concat(BTadresa[indeks]);//Nadopiši adresu nakon AT+RNAME?
  SerialAT(pomstring);//Sastavljeni pomstring predaj funkciji SerialAT
  citaj(select);
  if (modpr != mod) {//Prekid funkcije pomoću S5 tipkala
    return;
  }
  if (ef != 1) {//Ako je ispravno očitano ime ispiša ga
    lcd.setCursor(0, 56);
    lcd.print("IME:");
    lcd.print(BTnaziv[indeks]);
  }
}

/*****************************************************************/
/***Moja funkcija za povezivanje BT modula sa drugim uređajem***/
/*****************************************************************/
void PoveziBT() {
  citaj(select);
  if (modpr != mod) {//Prekid funkcije pomoću S5 tipkala
    return;
  }
  String pomstring = F("");
  digitalWrite(7, HIGH);
  cekaj(1000);
  digitalWrite(8, HIGH);
  cekaj(100);
  pomstring.concat(F("AT+BIND="));
  pomstring.concat(BTadresa[indeks]);
  SerialAT(pomstring);
  citaj(select);
  if (modpr != mod) {//Prekid funkcije pomoću S5 tipkala
    return;
  }
  pomstring = "";
  pomstring.concat(F("AT+PAIR="));
  pomstring.concat(BTadresa[indeks]);
  pomstring.concat(",5");
  SerialAT(pomstring);
  if (ef != 1) {//Ako nije došlo do error-a stavi indikator povezanosti u
povezanu vrijednost
    con = 1;
  }
}

/*****************************************************/
/***Moja funkcija za traženje dostupnih BT uređaja***/
/*****************************************************/
void PretraziBT() {//Funkcija za traženje drugih uređaja
  int brpetlj = 0;//brojač petlji
  String pomstring = F("");//čišćenje pomstring-a
  digitalWrite(7, HIGH);//Postavljanje HC-05 u AT+ mod
  cekaj(1000);
  digitalWrite(8, HIGH);
  cekaj(100);
  for (i = 1; i < 6; i++) {
    BTadresa[i] = F(""); /*Ukloni prijašnje spremljene adrese*/
  }
  for (brpetlj = 0; brpetlj < 3; brpetlj++) {//3 puta izvrši naredbu AT+INQ
    citaj(select);
    if (modpr != mod) {//Prekid funkcije pomoću S5 tipkala
      return;
```

```
    }
    digitalWrite(8, HIGH);
    cekaj(100);
    SerialAT(F("AT+INQ"));
    citaj(select);
    if (modpr != mod) {//Prekid funkcije pomoću S5 tipkala
      return;
    }
    cekaj(500);
  }
  StartTime = millis();
  ElapsedTime = 0;
}

/*********************************************************************/
/***Moja funkcija za dohvaćanje imena uređaja pomoću njihove adrese***/
/*********************************************************************/
void ImenaBT() {//Funkcija za dohvat imena uređaja na osnovu dobivenih adresa
  int pom1 = 40;
  int count = 1;//brojač pokušaja dohvata naziva
  lcd.setTextSize(2);
  lcd.setCursor(20, pom1);
  lcd.setTextColor(WHITE, BLACK);
  citaj(select);
  if (modpr != mod) {//Prekid funkcije pomoću S5 tipkala
    return;
  }
  String pomstring = F("");
  for (i = 1; i < 6; i++) {
    BTnaziv[i] = F(""); /*Ukloni prijašnje spremljene nazive*/
  }
  digitalWrite(8, LOW);//Restart HC-05 modula
  cekaj(500);
  digitalWrite(7, LOW);
  cekaj(500);
  digitalWrite(7, HIGH);
  cekaj(1000);
  digitalWrite(8, HIGH);
  cekaj(100);
  indeks = 1;
  while ((!(BTadresa[indeks].equals(""))) && (indeks < 6)) {//ako postoji
spremljena adresa
    pomstring.concat(F("AT+RNAME?"));
    pomstring.concat(BTadresa[indeks]);
    SerialAT(pomstring);
    lcd.setCursor(20, pom1);
    lcd.setTextColor(WHITE, BLACK);
    lcd.print("...Trazenje...");
    citaj(select);
    if (modpr != mod) {//Prekid funkcije pomoću S5 tipkala
      return;
    }
    cekaj(10);
    pomstring = F("");
    if (ef != 1) {//ako je uspješno pročitan naziv
      lcd.setCursor(0, pom1);
      for (i = 0; i < 21; i++) {//očisti red na LCD-u
        lcd.print(" ");
      }
      cekaj(40);
      lcd.setCursor(20, pom1);
      lcd.setTextColor(YELLOW, BLACK);
      lcd.print(indeks);
```

```
      lcd.print("# ");
      lcd.setTextColor(WHITE, BLACK);
      lcd.print(BTnaziv[indeks]);
      indeks++;
      count = 0;//resetiraj brojač za iduću adresu
      pom1 = pom1 + 16;
    }
    if (ef == 1) {
      if (count >= 3) {//nakon 3 neuspjela pokušaja dohvaćanja naziva, ispiši
adresu umjesto naziva
        lcd.setCursor(0, pom1);
        for (i = 0; i < 21; i++) {
          lcd.print(" ");
        }
        lcd.setCursor(20, pom1);
        lcd.setTextColor(YELLOW, BLACK);
        lcd.print(indeks);
        lcd.print("# ");
        lcd.setTextColor(WHITE, BLACK);
        lcd.print(BTadresa[indeks]);
        indeks++;
        pom1 = pom1 + 16;
        count = 0;//resetiraj brojač za iduću adresu
      }
    }
    count++;//Broji neuspjele pokušaje
    i = 0;
  }
  cekaj(10);
  digitalWrite(8, LOW);//Isključi pin34
  cekaj(100);
}

/***********************************************/
/***Funkcija za slanje AT+ naredbi HC-05 modul***/
/***********************************************/
void SerialAT(String niz) {//Funkcija koja izvršava potrebne radnje ovisno o
dobivenoj riječi
  int pomtp;//timer početak
  int pomtk;//timer kraj
  int pomvr = 0;//timer za izračun
  int j = 0;
  int k = 0;
  int br = 0;
  int brdvot = 0;//brojač dvotočke, za manipulaciju s oblikom adrese
  int brzar = 0;//brojač zareza, za manipulaciju s oblikom adrese
  int brinq = 0;//brojač koliko puta se izvršila AT+INQ naredbe
  char pom[64];//pomoćna polja za manipulaciju dobivenih podataka
  char pom2[16];
  String rijec = F("");
  String rname = F("AT+RNAME?");
  String pair = F("AT+PAIR=");
  String bind = F("AT+BIND=");
  String pet = F(",5");
  i = 0;
  ef = 0;//resetiraj Error flag
  while (Serial.available() > 0) { /*Ako je nešto u serijskom bufferu ostalo,
očisti da ne smeta u radu*/
    cekaj(2);
    Serial.read();//Jedini dostupan mehanizam za pražnjenje serijskog buffera
  }
  cekaj(20);
  Serial.print(niz);//serijski ispiši dobivenu naredbu
```

```
  Serial.print(F("\r\n"));//potrebni znakovi za ispravno izvršenje naredbe(\r
- carriage return \n - novi red)

  pomtp = millis();//pokreni pomoćni timer
  //Provjera ispravnog unosa i odgovora naredbi koje imaju duži odziv, ako je
potrebno duže od 10 s, došlo je do greške
  if (niz.equals(rname + BTadresa[indeks]) || niz.equals(pair +
BTadresa[indeks] + pet) || niz.equals(bind + BTadresa[indeks]) ||
niz.equals(F("AT+INQ"))) { /*Potrebna jos jedna petlja zbog dugog odziva na
funkciju*/
    lcd.setCursor(0, 56);
    while ((Serial.available() == 0) && (pomvr < 10000)) {//Sve dok se ne
pojavi nešto u serijskom bufferu za čitanje čekaj, ako protekne više od 10 s,
javi da je došlo do greške
      pomtk = millis();
      pomvr = pomtk - pomtp;
      if (pomvr >= 10000) {
        ef = 1;
        con = 0;
      }
    }
    cekaj(15);
    if (ef == 0) {//ako je uspješno došao odgovor sa HC-05 na AT+ naredbu,
odradi potrebno ovisno o naredbi
      if (!(niz.equals(F("AT+INQ")))) {//Ako je unešena bilo koja druga
naredba osim AT+INQ
        StartTime = millis();//pokreni timer
        while (Serial.available() > 0) {
          cekaj(1);/*Pričekaj radi stabilnosti*/
          pom[i] = Serial.read();//Pospremi odgovor u pom[i]
          Serial.print(pom[i]);/*Koristi se samo za debugging putem Serial
Monitor-a*/
          i++;
        }
      }
      if (niz.equals(F("AT+INQ"))) {//Ako je unešena AT+INQ
        StartTime = millis();//pokreni timer
        ElapsedTime = 0;//resetiraj proteklo vrijeme timera
        while ((ElapsedTime < 10000) && (brinq <= 3)) {
          while ((Serial.available() > 0) && (ElapsedTime < 10000)) {
            cekaj(1);/*Pricekaj radi stabilnosti*/
            pom[i] = Serial.read();//Pospremi odgovor u pom[i]
            if (pom[i] == ':') {//Broji dvotočke u dobivenom odgovoru
              brdvot++;
            }
            if (pom[i] == ',') {//Broji zareze u dobivenom odgovoru
              brzar++;
            }
            if (((pom[i] < 71) || (pom[i] > 122)) && (pom[i] != '+')) {
//ispiši i spremi samo određene znakove(pogledati u ASCII tablicu)
              Serial.print(pom[i]);/*Koristi se samo za debugging putem
Serial Monitor-a*/
              i++;//Ako je neki znak van zadanog opsega, u idućem prolazu
petlje će se u pom[i] prebrisati, u suprotnom neće
            }
            if (pom[i - 1] == '\n') {//brojač ispisanih redova odgovora
naredbe AT+INQ
              brinq++;
              if ((brdvot == 3) && (brzar >= 1)) {//kada je ispisana adresa u
potpunosti
                rijec = "";/*Poništi prijašnju adresu*/
                k = 1;
```

```
                //Ispravan oblik adrese za daljnje korištenje je npr.
00AB,01,012345, a ne AB:1:12345,FF,FF kako bi AT+INQ dala u odgovoru, gdje je
nakon zareza navedena klasa uređaja i snaga signala
                while (pom[k] != ':') { /*Broji količinu znamenaka do prvog
":"*/
                  br++;
                  k++;
                }
                for (k = 0; k < (4 - br); k++) { /*Nadopiši koliko ih
nedostaje da bi bilo 4 znamenaka*/
                  rijec.concat(F("0"));
                }
                j = 1;
                while (pom[j] != ':') { /*Napiši prvi konkretni dio adrese do
":"*/
                  rijec.concat(pom[j]);
                  j++;
                }
                rijec.concat(F(","));/*Napiši ","*/
                j++;
                k = j;
                br = 0;
                while (pom[k] != ':') { /*Broji količinu znamenaka od prvog
do drugog ":"*/
                  br++;
                  k++;
                }
                for (k = 0; k < (2 - br); k++) { /*Nadopiši koliko nula
nedostaje da bi bile 2 znamenke*/
                  rijec.concat(F("0"));
                }
                while (pom[j] != ':') { /*Napiši drugi konkretni dio adrese
do ":"*/
                  rijec.concat(pom[j]);
                  j++;
                }
                rijec.concat(F(","));/*Napiši ","*/
                j++;
                k = j;
                br = 0;
                while (pom[k] != ',') { /*izbroji koliko ih nedostaje do 6
znamenki*/
                  k++;
                  br++;
                }
                for (k = 0; k < 6 - br; k++) {
                  rijec.concat(F("0"));//Nadopiši po potrebi nule
                }
                for (k = j; k < j + br; k++) {
                  rijec.concat(pom[k]);
                }
                i = 1;
                if (i <= 5) {
                  while ((BTadresa[i].length() != 0)) {//Pronađi slobodno
mjesto za pospremu adrese
                    if (BTadresa[i].equals(rijec)) {
                      i = 6;//postavi uvjet za prekid petlje kada više nema
slobodnih mjesta
                    }
                    else {
                      i++;
                    }
                  }
```

```
                }
                if (i <= 5) {
                  BTadresa[i] = rijec;//Pospremi dobivenu adresu na slobodno
mjesto
                }

                i = 0;//Reset varijabli za dohvat nove adrese
                br = 0;
                brdvot = 0;
                brzar = 0;
              }
            }

          }
          CurrentTime = millis();
          ElapsedTime = CurrentTime - StartTime;

        }
        if (brinq >= 4) {//Procedura za prekid AT+INQ naredbe nakon ispisanih
4 reda adresa kako bi se spriječio buffer overflow. Ponekad je moguće da se
ispiše i 5. red adrese zbog sporog serijskog čitanja
          cekaj(1000);
          digitalWrite(8, LOW);//AT+INQ se prekida postavljanjem Pin34 na
logičku nulu
          cekaj(2000);
          digitalWrite(8, HIGH);
          cekaj(500);
          Serial.print(F("AT+DISC\r\n"));//HC-05 se pri korištenoj verziji
firmware-a automatski spaja na zadnju adresu dobivenu iz AT+INQ naredbe.
Potrebno provesti naredbu za prekid veze
          cekaj(500);
          digitalWrite(8, LOW);
          cekaj(100);
        }
      }

      if (niz.equals(rname + BTadresa[indeks])) {//Procedura za
AT+RNAME?adresa, odnosno za dohvat imena
        for (j = 0; j < i - 2; j++) {
          rijec.concat(pom[j]);//Dodaj u string rijec znak po znak pročitani
odgovor, izuzev zadnja 2 očitana znaka \r\n
        }
        if (rijec.equals(F("FAIL"))) {//ukoliko je očitana riječ FAIL
          ef = 1;//došlo je do greške
          con = 0;//nije uspostavljena veza
          BTnaziv[indeks] = "";
          BTnaziv[indeks] = rijec;
          for (j = 0; j < 16; j++) {
            BTnaziv[indeks].concat(" ");//priprema za ispis na LCD
          }
        }
        else {//Ako je uspješno očitan naziv
          BTnaziv[indeks] = "";//očisti mjesto za pohranu
          cekaj(10);
          for (j = 7; j < i - 6; j++) {//Ukloni prvih 7 znakova +RNAME: i
zadnjih 6 znakova odgovora \r\nOK\r\n
            BTnaziv[indeks].concat(pom[j]);//Na taj način se sprema čisti
naziv
          }
        }
      }
    }
    else {//Ako nije veza uspostavljena
```

```
      Serial.println(F("!Adresa nedostupna!"));
      digitalWrite(8, LOW);
      cekaj(100);
      con = 0;
      conctrl = 0;
    }
  }
  else {//Za AT+ naredbe koje imaju brži odziv
    while ((Serial.available() == 0) && (pomvr < 50)) {//Očekivani početak
odziva unutar 50ms
      pomtk = millis();
      pomvr = pomtk - pomtp;
      if (pomvr >= 50) {
        ef = 1;//Ako nije u 50 ms došlo do odziva, došlo je do greške
      }
    }
    lcd.setCursor(0, 40);
    cekaj(15);/*Pricekaj radi stabilnosti, duljina ovisna o duljini riječi*/
    if (ef == 0) {
      while (Serial.available() > 0) {
        cekaj(1);/*Pričekaj radi stabilnosti*/
        pom[i] = Serial.read();//Čitaj odgovor i pospremi u pom[i]
        Serial.print(pom[i]);/*Koristi se samo za debugging putem Serial
Monitor-a*/
        i++;
      }
      if (niz.equals(F("AT+MRAD"))) { /*Za pročitati adresu zadnje spajanog
modula/spojenog trenutno*/
        BTadresa[indeks] = F("");/*Poništi prijašnju adresu*/
        k = 6;
        //Manipulacija s adresom kao i kod AT+INQ naredbe
        while (pom[k] != ':') { /*Broji količinu znamenaka do prvog ":"*/
          br++;
          k++;
        }
        for (k = 0; k < (4 - br); k++) { /*Nadopiši koliko ih fali da bi ih
bilo 4 znamenaka*/
          BTadresa[indeks].concat(F("0"));
        }
        j = 6;
        while (pom[j] != ':') { /*Napiši prvi konkretni dio adrese do ":"*/
          BTadresa[indeks].concat(pom[j]);
          j++;
        }
        BTadresa[indeks].concat(F(","));/*Napiši ","*/
        j++;
        k = j;
        br = 0;
        while (pom[k] != ':') { /*Broji količinu znamenaka od prvog do drugog
":"*/
          br++;
          k++;
        }
        for (k = 0; k < (2 - br); k++) { /*Nadopiši koliko ih fali da bi bilo
2 znamenke*/
          BTadresa[indeks].concat(F("0"));
        }
        while (pom[j] != ':') { /*Napiši drugi konkretni dio adrese do ":"*/
          BTadresa[indeks].concat(pom[j]);
          j++;
        }
        BTadresa[indeks].concat(F(","));/*Napiši ","*/
        j++;
```

```
        br = (i - 6) - j;
        for (k = 0; k < (6 - br); k++) { /*Nadopiši koliko ih fali da bi bilo
6 znamenki*/
          BTadresa[indeks].concat(F("0"));
        }
        for (k = j; k < i - 6; k++) {
          BTadresa[indeks].concat(pom[k]);
        }
      }
      if (niz.equals(F("AT+STATE?"))) {//Za stanje HC-05 modula
        BTstate = F("");
        j = 1;
        while (pom[j] != '\r') {//\r znak se upotrebljava kao limit ispisa
          lcd.print(pom[j]);//ispiši stanje HC-05 modula na LCD-u
          BTstate.concat(pom[j]);
          j++;
        }
        for (j = i - 6; j < 21; j++) {
          lcd.print(F(" "));//Prebriši zaostale znakove na LCD-U od
prijašnjeg stanja
        }
      }
    }
    else {
      lcd.print(F("Modul nije aktivan!"));//Ukoliko ne postoji serijska veze
između Arduina i HC-05
      Serial.println(F("Modul nije aktivan!"));
      ef = 1;
      con = 0;
      conctrl = 0;
    }
  }
}

/*********************************************************************/
/***Moja Funkcija za slanje odabranih ulaza putem BT komunikacije***/
/*********************************************************************/
void salji(){//Ovisno o select šalje putem BT veze ImeVarijable+Vrijednost+\n
  int pingtimkr = 0;
  int pingtimvr = 0;
  pingtimkr = millis();
  pingtimvr = pingtimkr - pingtim;
  if (pingtimvr > 100) {
    Serial.print(F("~PING~\n"));//Svakih 100 ms pošalji PING
    cekaj(20);
    pingtim = millis();
  }
  if ( (select & 0x0400) != 0) {
    Serial.print(F("~GLX~"));//Naziv ulaza, ~ se koristi za raspoznavanje
naziva ulaza
    Serial.print(GLXval);//Pošalji vrijednost ulaza
    Serial.print(F("\n"));//Pošalji znak \n kao oznaku kraja pojedinog ulaza
    cekaj(8);//Za ispravnu serijsku komunikaciju na Baudrate 38400 je
potrebno minimalno 0.42 ms po znaku, što u ovom slučaju je minimalnih 3.36
ms. No budući da timer ne radi idealno, potrebno je dodati još neku
sigurnosnu razliku u kojoj će se izvršiti naredbe na robotu. Ako se ne
poštiva čekanje dolazi do overflow-a serijskog buffer-a predajnika, kao i
prijemnika
  }
  if ( (select & 0x0200) != 0) {
    Serial.print(F("~GLY~"));
    Serial.print(GLYval);
    Serial.print(F("\n"));
```

```
    cekaj(8);
  }
  if ( (select & 0x0100) != 0) {
    Serial.print(F("~GDX~"));
    Serial.print(GDXval);
    Serial.print(F("\n"));
    cekaj(8);
  }
  if ( (select & 0x0080) != 0) {
    Serial.print(F("~GDY~"));
    Serial.print(GDYval);
    Serial.print(F("\n"));
    cekaj(8);
  }
  if ( (select & 0x0040) != 0) {
    Serial.print(F("~PL~"));
    Serial.print(PLval);
    Serial.print(F("\n"));
    cekaj(8);
  }
  if ( (select & 0x0020) != 0) {
    Serial.print(F("~PD~"));
    Serial.print(PDval);
    Serial.print(F("\n"));
    cekaj(8);
  }
  if ( (select & 0x0008) != 0) {
    Serial.print(F("~S1~"));
    Serial.print(S1val);
    Serial.print(F("\n"));
    cekaj(8);
  }
  if ( (select & 0x0004) != 0) {
    Serial.print(F("~S2~"));
    Serial.print(S2val);
    Serial.print(F("\n"));
    cekaj(8);
  }
  if ( (select & 0x0002) != 0) {
    Serial.print(F("~S3~"));
    Serial.print(S3val);
    Serial.print(F("\n"));
    cekaj(8);
  }
  if ( (select & 0x0001) != 0) {
    Serial.print(F("~S4~"));
    Serial.print(S4val);
    Serial.print(F("\n"));
    cekaj(8);
  }
}

void setup() {//inicijalne postavke Arduina pri pokretanju
  Serial.begin(38400);//Inicijaliziraj serijsku komunikaciju na Baudrate
38400 bitova u sekundi
  lcd.init(SCR_WD, SCR_HT);//Inicijaliziraj LCD
  lcd.fillScreen(BLACK);//Prekrij LCD crnom bojom
  pinMode(7, OUTPUT);//Definiranje pinova kao Output za ispravan rad
  pinMode(8, OUTPUT);
  digitalWrite(7, LOW);//Inicijalno isključi HC-05 modul
  digitalWrite(8, LOW);
  BTadresa[0] = F("");//Očisti mjesto za zadnju adresu
```

```
  Serial.println(F("***Pokrenuto na pocetku***\n"));//Ispiši na Serial
Monitor-u da je pokrenut program na početku

  StartTime = millis();//Pokreni timer
}

void loop() { //glavna petlja programa
  if (modpr != mod) {//Detektiranje promjene mod-a, izvrši inicijalne radnje
za promjenu moda
    modpr = mod;
    cekaj(20);
    select = 0x07FF;
    PromjenaModa(mod);
  }

  if (mod == 0) { /*Potrebne radnje za BT komunikaciju mod*/
    citaj(select);//Očitaj ulaze
    UlaziDin(mod, select);//Ispiši ulaze ovisno o promjeni
    if (conctrl == 1) {//Ako je uspostavljena veza, šalji podatke
      salji();
    }
    if (Serial.available() > 0) {//Čitanje povratnih informacija s Robota
      while (Serial.available() > 0) {
        pomocnichar = Serial.read();
        cekaj(1);
        if (pomocnichar == '~') {//Odvajanje ključnih riječi pomoću ~
          stringbrojac++;
        }
        else if (stringbrojac == 1) {
          pomocnistring.concat(pomocnichar);//očitaj ključnu riječ
        }
        if (stringbrojac == 2) {//kada se očita drugi ~ riječ je poslana
          if (pomocnistring.equals(F("PING"))) {//Ako je dobiven PING ispiši
na Serial Monitor-u TestP

            Serial.println(F("TestP"));
            cekaj(8);
          }

          StartTime = millis();
          pingcon = 1;//Veza postoji jer je dobivena povratna informacija
          pomocnistring = F("");
          stringbrojac = 0;
          break;//Kako bi se spriječili potencijalni problemi očitavanjem
nepotrebnih ili neispravnih informacija
        }
      }
    }
    CurrentTime = millis();
    ElapsedTime = CurrentTime - StartTime;
    if (ElapsedTime > 6000) {//Ako unutar 6 sekundi nije primljen PING,
postavi zastavicu upozorenja pingcon
      pingcon = 0;
    }
    if (pingcon == 0) {//Ako nema PING-a, provjeriti naredbom AT+STATE stanje
povezanosti HC-05 modula
      digitalWrite(8, HIGH);
      cekaj(500);
      SerialAT(F("AT+STATE?"));/*Provjera povezanosti HC-05 s Arduino-m, ako
nije povezan ef se prebacuje u 1 te se poduzima daljnja procedura*/
      if ((BTstate.equals(F("STATE:PAIRABLE"))) && conctrl == 1) {
        conctrl = 0;//Veza nije uspostavljena
      }
```

```
      cekaj(10);
      digitalWrite(8, LOW);
      cekaj(100);
    }
    if (ef == 1) {//Za slučaj greške
      digitalWrite(7, LOW);
      digitalWrite(8, LOW);
      cekaj(500);
      digitalWrite(7, HIGH);
      cekaj(1000);
      digitalWrite(8, HIGH);/*Ako je došlo do greške, pokušati restart HC-05
ako je ušao u mod na baud rate 38400*/
      cekaj(500);
      ef = 0;
      con = 0;
      conctrl = 0;
    }
    if (ef != 1) {//Kada greška ne postoji
      if (conctrl == 0) {//Ako nije povezano
        digitalWrite(8, HIGH);
        cekaj(500);
        SerialAT(F("AT+STATE?"));
        cekaj(10);
        if (BTstate.equals(F("STATE:PAIRABLE"))) {//Ako je spreman za
povezivanje
          conctrl = 0;
          con = 0;
          cekaj(10);
          ZadnjiBT();//Dohvati adresu i ime zadnje povezanog
          cekaj(10);
          if (ef != 1) {
            PoveziBT();//Povezi s zadanim na BTadresa[0]
          }
        }
        if (BTstate.equals(F("STATE:CONNECTED"))) {//Kada je uspješno
povezano
          conctrl = 1;
          con = 1;
          digitalWrite(8, LOW);
          cekaj(500);
          while (Serial.available() > 0) {
            Serial.read();
          }
          Serial.print("~SEL~");//Pročitaj poslani Select s robota
          cekaj(50);
          stringbrojac = 0;
          strtimpoc = millis();
          strtim = 0;
          while ((stringbrojac < 2) && (strtim < 500)) {//Usporiti dok se ne
pročita sve potrebno
            while (Serial.available() > 0) {
              pomocnichar = Serial.read();
              cekaj(5);
              if (pomocnichar == '~') {
                stringbrojac++;
                cekaj(5);
              }
              else if (stringbrojac == 1) {
                pomocnistring.concat(pomocnichar);
                cekaj(5);
              }
            }
            strtimkrj = millis();
```

```
            strtim = strtimkrj - strtimpoc;
          }
          Serial.println(stringbrojac);//Za Debugging putem Serial Monitora
          Serial.println(pomocnistring);
          stringbrojac = 0;
          if (!(pomocnistring.equals(F("")))) {//Ako je nešto poslano
            select = pomocnistring.toInt();//pohrani dobivenu informaciju u
select
            cekaj(5);
            pomocnistring = F("");
          }
          UlaziInfoS(mod, select);//Ponovno ispiši na LCD samo odabrane
nazive
          cekaj(200);
          lcd.setCursor(0, 40);
          lcd.print(F("STATE:CONNECTED"));//ispiši na LCD stanje povezanosti
          lcd.setCursor(0, 56);
          lcd.print(F("IME:"));//ispiši na LCD naziv spojenog uređaja
          lcd.print(BTnaziv[indeks]);
          cekaj(100);
          pingcon = 1;
          StartTime = millis();
        }
      }
    }
  }

  else if (mod == 1) {//Kontinuirane radnje u konfiguracijskom modu
    UlaziDin(mod, 0x0000);
    citaj(select);
  }

  CurrentTime = millis();
  ElapsedTime = CurrentTime - StartTime;
}
```

5. ZAKLJUČAK

Zadatak završnog rada bio je projektiranje, izvedba i programiranje modula za daljinsko upravljanje robotima koji se temelji na primjeni Arduino upravljačke jedinice s pripadnim komponentama. Za lakše razumijevanje eksperimentalnog dijela opisani su općenito robotski sustavi, načini daljinskog upravljanja, Arduino razvojno sučelje i okruženje, korištene komponente sa shemama spajanja istih.

Prilikom izvedbe i testiranja sustava bilo je problema zbog nedostatka dostupne dokumentacije za verziju 3.0-20170601 HC-05 Bluetooth modula koja se u načinu rada razlikuje od verzije 2.0-20100601 za koju je dostupna dokumentacija. Prilikom pisanja i testiranja programskog koda najviše problema je bila s vremenskim usklađivanjem serijske komunikacije između Arduino Nano i HC-05 modula gdje je zbog ograničenosti brzine serijske komunikacije bilo potrebno povećavati čekanja nakon pojedinog serijskog ispisa kako nebi došlo do overflow-a serijskog buffer-a, a za upravljač je poželjno čim manje potrebno vrijeme izvođenje petlje. Postignuto je izvođenje ciklusa petlje, odnosno vrijeme u kojem se ispišu i pošalju sve vrijednosti ulaza robotu za 150 ms u slučaju kada se koriste svi ulazi, što je granična vrijednost ugodnog upravljanja. Poboljšanja u brzini izvođenja programa se mogu postići boljom optimizacijom programa ili odabirom boljeg hardware-a.

LITERATURA

[1]https://people.etf.unsa.ba/~jvelagic/laras/dok/Robotika_uvod.pdf
Datum pristupa: 11.4.2022.

[2]Osnove robotike: Predavanje 2, VUKA, Denis Kotarski

[3]https://www.chipoteka.hr/

Datum pristupa: 11.4.2022.

[4]https://www.kuka.com

Datum pristupa: 11.4.2022.

[5]https://enciklopedija.hr/Natuknica.aspx?ID=13776

Datum pristupa: 11.4.2022.

[6]https://www.arduino.cc/

Datum pristupa: 11.4.2022.

[7]https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf
Datum pristupa: 30.3.2022.

[8]https://components101.com/wireless/hc-05-bluetooth-module
Datum pristupa: 30.3.2022.

[9]https://github.com/cbm80amiga/Arduino_ST7789_Fast
Datum pristupa: 1.4.2022.

[10]https://imgaz.staticbg.com/images/oaupload/banggood/images/81/65/d5f793d7-9ee1-4881-8234-9e188f969133.jpg
Datum pristupa: 1.4.2022.

PRILOZI

1. CD-R