

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2025

**IZVEDBA DALJINSKI UPRAVLJANOG MOBILNOG
ROBOTA**

Luka Žugaj

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2025

**IZVEDBA DALJINSKI UPRAVLJANOG MOBILNOG
ROBOTA**

Luka Žugaj

Zagreb, lipanj 2020.

DIPLOMSKI ZADATAK br. 2025

Pristupnik: **Luka Žugaj (0036494176)**
Studij: Elektrotehnika i informacijska tehnologija
Profil: Elektroničko i računalno inženjerstvo
Mentor: prof. dr. sc. Davor Petrinović

Zadatak: **Izvedba daljinski upravljano mobilnog robota**

Opis zadatka:

U okviru diplomskog rada potrebno je razviti i realizirati inteligentnog mobilnog robota, s pogonom na sva četiri kotača, upravljano putem daljinskog sustava temeljenog na Bluetooth tehnologiji veze i korištenjem GPRS modula. GPRS modul mora osigurati mogućnost slanja podataka s mobilnog robota serveru na zahtjev u svrhu vizualizacije traženih podataka putem internetskog preglednika. Pored dvosmjerne GPRS veze modul treba sadržavati i GPS prijemnik za lokalizaciju robota. Pored informacije o poziciji, senzori robota trebaju obuhvaćati ultrazvučni senzor udaljenosti do prepreke i optičke enkodere za praćenje vrtnje mora i gibanja robota. Upravljanje sustava realizirati odgovarajućim mikrokontrolerom. Realizirati programsku potporu za cijeli sustav. Provesti integraciju svih komponenti, njihovo testiranje, kao i provjeru rada cjelokupnog sustava. Za dodatne informacije obratite se mentoru.

Rok za predaju rada: 30. lipnja 2020.

Sadržaj

Uvod	1
1. Arhitektura sustava i programska podrška	2
1.1. Arduino IDE i programski jezici C/C++	3
1.2. Blockly	4
1.3. Google Spreadsheet uređivač	5
2. Fizičke komponente sustava	6
2.1. Arduino Mega 2560	6
2.2. Moduli serijske komunikacije	8
2.2.1. SIM800L EVB	8
2.2.2. Bluetooth HC-06	10
2.2.3. GPS U-BLOX NEO-7M	12
2.3. Senzori	14
2.3.1. Senzor brzine LM393	14
2.3.2. Ultrazvučni senzor HC-SR04	16
2.3.3. DC Motor Driver Dual H-Bridge L298N	18
2.4. Napajanje sustava	20
3. Implementacija i povezivanje sustava mobilnog robota	21
3.1. Opis problema i motivacija	21
3.2. Načelo rada sustava	22
3.2.1. Sustav u stanju upravljanja mobilnom aplikacijom	22
3.2.2. Slanje podataka na zahtjev korisnika	27
4. Slanje podataka na Internet	32
4.1. PushingBox servis	34
4.2. Pohrana podataka u Google Spreadsheet	35
5. Izrada Android aplikacije za upravljanje robotom	40

6. Mogućnosti nadogradnje sustava.....	46
Zaključak	48
Literatura	49
Sažetak.....	50
Summary.....	51

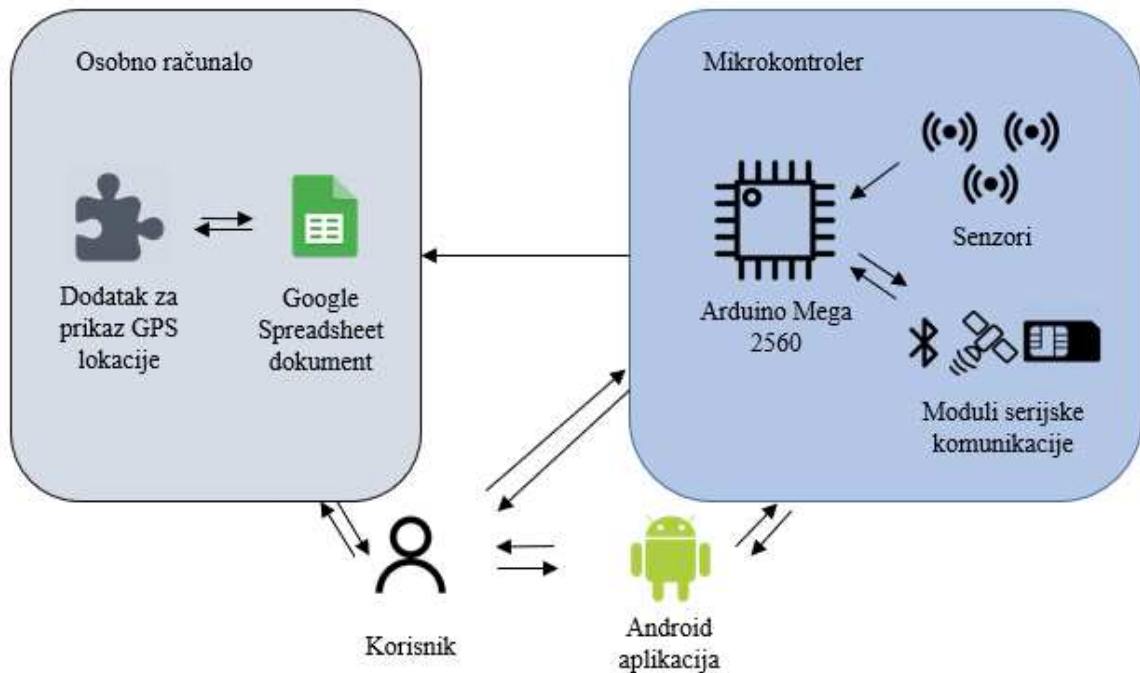
Uvod

Zadnjih nekoliko godina u svijetu se pojavila velika potreba za umrežavanjem digitalnih sustava i različitih elektroničkih uređaja putem interneta. Globalna ideja čovječanstva je maksimizirati automatizaciju različitih procesa i tako uštedjeti na potrebnim resursima i vremenu. Takav način komunikacije i povezivanja naziva se Internet stvari i predstavlja budućnost kojom će sustavi međusobno razmjenjivati informacije. Obrtnici i poduzetnici se usmjeravaju na poboljšanje korisničkog iskustva svojih korisnika, a primjena Interneta stvari upravo je zamišljena da grupira podatke iz raznih sustava i prikaže ih na najjednostavniji način krajnjem kupcu. To su najčešće grafičke i statističke analize koje čovjek lako pamti i koristi u svakodnevnom radu.

Cilj ovog rada je dizajnirati IoT (eng. *Internet of Things*) sustav s takvim karakteristikama u obliku mobilnog robota na daljinsko upravljanje te omogućiti korisniku lako i intuitivno upravljanje te pregled dostupnih podataka. Sustav je upravljan s mikrokontrolerom Arduino Mega 2560 na kojeg su povezani bluetooth modul HC-06, SIM modul SIM800L EVB te GPS modul U-BLOX NEO-7M koji ostvaruju serijsku komunikaciju s Arduino modulom. Robot također sadrži nekoliko senzora koji uključuju podatke za analizu i rad sustava kao što su brzina gibanja robota, pređena udaljenost robota i udaljenost od prepreke. Svi podaci se dostavljaju preko GSM veze na internet u jednakim vremenskim intervalima u obliku Google Spreadsheet forme. Tako podaci postaju lako dostupni za daljnju obradu i analizu. Jedna od prepoznatljivih značajki Interneta stvari je imati mogućnost upravljanja sustavima na dlanu, najčešće preko mobilnih telefona, stoga je u svrhu ovog rada dizajnirana Android aplikacija kojom se omogućuje upravljanje mobilnog robota.

1. Arhitektura sustava i programska podrška

Izgled i rad sustava koji je predstavljen izradom mobilnog robota može se podijeliti na nekoliko cjelina. Kako bi se na pregledan način objasnio rad cjelokupnog sustava, na sljedećoj slici (Slika 1.1) prikazane su komponente koje ga sačinjavaju. Uz komponente sustava, strelicama je naznačen i smjer komunikacije između različitih slojeva sustava.



Slika 1.1 Prikaz arhitekture sustava

Glavna sastavnica sustava je mikrokontroler koji povezuje očitavanja različitih senzora u jednu cjelinu i ujedno komunicira s ostalim modulima serijske komunikacije. U mikrokontroleru se odvija obrada podataka i slanje istih na internet. S druge strane nalazi se korisnik, koji na dva načina može doći do željenih informacija. Prvi način je preko Google Spreadsheet dokumenta gdje su pohranjeni podaci sa senzora i vrijeme očitavanja, a drugi način je na zahtjev prema upravljačkoj jedinici. U idućim poglavljima je detaljno obrađeno kako funkcionira ta dvosmjerna komunikacija, koji se podaci prenose i na koji način.

Za potrebe projekta potrebna su različita programska okruženja. Mikrokontroler korišten u radu je Arduino Mega 2560. Stoga je programski kod za glavnu komponentu pisan u programskom okruženju Arduino IDE koji se temelji na programskim jezicima C/C++. Mobilni robot je upravljan Android aplikacijom te je za njenu izradu korištena platforma App Inventor koja je rađena na softveru zvanom Blockly. Posljednja programska cjelina

sustava je Google Spreadsheet. S obzirom na to da se podaci prenose u Spreadsheet dokument, nužno je definirati programski kod koji prihvaća parametre i očitavanja senzora te ih prikazuje unutar tablice u zadanom obliku. Skripta je pisana u JavaScript kodu, a u sljedećim poglavljima je detaljnije opisan svaki od programskih jezika navedenih programskih cjelina.

1.1. Arduino IDE i programski jezici C/C++

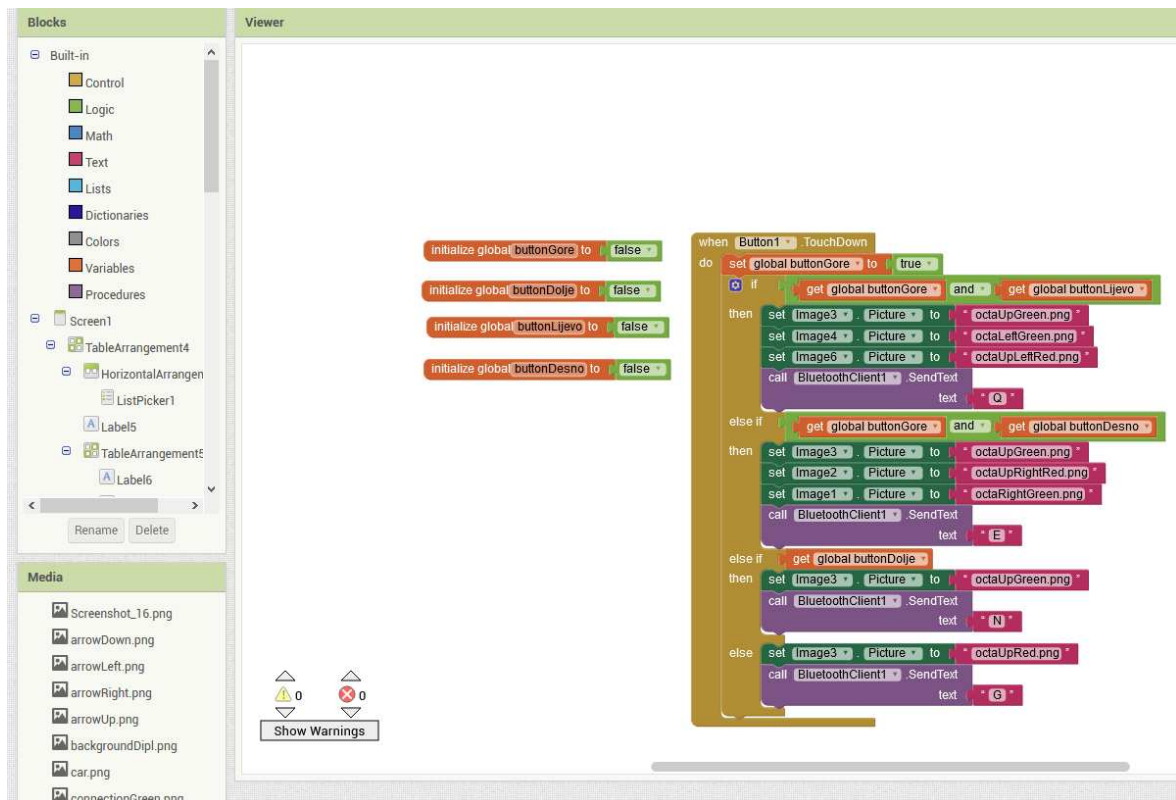
Programsko okruženje Arduino IDE često se naziva podskupom programskih jezika C i C++ jer se sastoji od skupa funkcija temeljenih na tim jezicima. Kod koji je napisan prilikom izvršenja prolazi kroz manje izmjene, kao što je na primjer automatsko generiranje prototipa funkcije, a zatim se izravno prosljeđuje u C/C++ prevodilac [1]. Naredbe u Arduino IDE okruženju podržavaju funkcije C/C++ strukture. Zbog razloga što su C i C++ poznati kao jezici opće namjene, korisnicima Arduino IDE platforme jednostavno je koristiti gotove funkcije i biblioteke iz postojećih programskih jezika. Za većinu senzora, koji imaju kompleksniju izvedbu ili sadrže čip u sebi, već postoje određene biblioteke dostupne na internetu. Integracija takvih biblioteka u programski kod je vrlo lako izvediva. Arduino IDE je specifičan po svojoj strukturi zbog procesora koji je ugrađen unutar Arduino mikrokontrolera. Naime, procesor ne sadrži operacijski sustav ni mogućnost da pokreće više programa od jednog istovremeno. Zbog toga se u kodu definira beskonačna petlja koja može izvršavati samo jedan program odjednom. Pored beskonačne petlje, postoji „setup“ petlja koja se izvršava samo jednom i postavlja određene varijable u početno stanje. U slučaju greške, Arduino se ponovno pokreće i kod se iznova izvršava uključujući „setup“ petlju. Na sljedećoj slici je prikazan Arduino IDE softver s jednostavnim primjerom koji se koristi za testiranje mikrokontrolera i provjere njegovog rada.



Slika 1.2 Arduino IDE sučelje

1.2. Blockly

Blockly se često naziva bibliotekom JavaScripta, ali predstavlja softver otvorenog koda koji omogućava pisanje koda pomoću vizualnih alata temeljenih na blokovima. Razvijen je od strane Google-a i programera na sveučilištu Massachusetts Institute of Technology. Ova tehnologija se koristi u mnoštvu projekata, od kojih je većina edukacijske namjene kao što su: Microsoft MakeCode, micro:bit, App Inventor, Gameblox itd. [2]. Blockly je dizajniran kako bi se jednostavno integrirao u neki web servis. Jedna od karakteristika Blockly-a jest da je potpuno ovisan o klijentskoj strani, odnosno ne zahtijeva podršku od aplikacija treće strane. Korisnik ima mogućnost definiranja grafičkog sučelja preko metode povlačenja i ispuštanja (eng. *drag and drop*). Tako je lako upravljati radnom površinom, brisati, dodavati ili modificirati programske blokove. Sustav koji implementira Blockly dozvoljava korisniku da kombinira tehnike pisanja koda, npr. korištenje programskih blokova ili pisanje čistog programskog koda u nekom drugom uređivaču koda. Ovaj rad je napravljen koristeći MIT App Inventor, koji kombinira te tehnike. Sučelje tog sustava prikazano je na sljedećoj slici.



Slika 1.3 MIT App Inventor sučelje

1.3. Google Spreadsheet uređivač

U sklopu ovog rada za integraciju projekta i slanje podataka na internet u realnom vremenu koristi se Google Spreadsheet. Ovaj dio Google platforme omogućava korisniku jednostavnu analizu i skladištenje podataka na intuitivnom grafičkom sučelju. Osim toga što je ovaj alat najčešće korišten kao pomoć za pohranu statičkih podataka, sadrži dodatak koji omogućava pisanje programskog koda i dinamički prikaz informacija u tablici. Kod je pisan u programskom jeziku JavaScript. Ovakav tip integracije najčešće se koristi kada se želi pristupiti tablici preko nekog drugog aplikacijskog programabilnog sučelja (eng. *Application programming interface*) [3]. Iako su na ovaj način informacije dostupne na internetu, pristup dokumentu mogu imati samo odabrani korisnici. Sigurnost podataka je na visokoj razini ako se radi na projektu kućne izrade, no za neke kompleksnije primjene svakako je primjenjivo koristiti enkripcijske ključeve i token za pristup tablicama. S obzirom na to da se u radu koristi modul koji ima pristup na internet, neophodno je povezati i uskladiti rad modula s programskim kodom pisanim u Google Spreadsheet uređivaču. U kodu su navedeni parametri koji se primaju i način popunjavanja tablice, sve kako bi korisniku dohvat podataka bio što lakši, a prikaz i slanje istih automatiziran.

2. Fizičke komponente sustava

Osim programske podrške, koja predstavlja logiku cijelog sustava, mikrokontroler, senzori i ostali moduli serijske komunikacije čine opipljivi dio modela robota. Uz komponente koje zahtijevaju električnu povezanost putem žica i napajanja, konstrukcija mobilnog robota sastoji se od raznih plastičnih dijelova koji su pomoću vijaka učvršćeni na šasiju. Robot je napravljen u obliku automobila na daljinsko upravljanje, a na šasiji se nalaze sve komponente sustava, uključujući i napajanje u obliku baterija. Stoga je potrebno naći optimalan raspored svih dijelova. Kako bi se omogućio ispravan rad modula poput GPS i SIM modula, potrebno ih je smjestiti na dio gdje će razina signala biti najveća. Također, senzore treba učvrstiti da se prilikom kretanja robota ne pomiču.

2.1. Arduino Mega 2560

Glavna upravljačka jedinica cijelog sklopa je razvojna pločica Arduino Mega 2560. Kako i samo ime govori, radi se o elektroničkoj komponenti koja svoj rad temelji na otvorenoj softverskoj i računalnoj platformi Arduino. Najjednostavnije rečeno, Mega 2560 je alat za upravljanje elektroničkim sklopovima i njihovom logikom. Zahvaljujući jednostavnom rukovanju ovim alatom, moguće je izraditi projekt kućne izrade koji obavlja gotovo iste funkcije kao i komercijalni sustav, čija je cijena nekoliko puta viša. Arduino Mega 2560 je nasljednik istoimenog modela Arduino Mega, čija je glavna komponenta 8-bitni mikrokontroler ATmega2560. Zbog svoje pristupačnosti, niske cijene i visokih performansi, iznimno je popularan kod zadataka gdje nije potrebno izvršavanje više funkcija odjednom. To je ujedno i najveća mana ovog mikrokontrolera, no zbog visoke frekvencije rada pločice Arduino Mega 2560, u ovom radu je pokazano kako je moguće izvršavati nekoliko zadataka gotovo istovremeno, bez značajnijih programskih kašnjenja. Zbog opsega ovog projekta, broja modula i senzora koji su potrebni za njegovo ostvarenje, ne koriste se manje verzije kao što su Arduino UNO ili Micro, ponajviše zbog manjka digitalnih pinova i vrata za serijsku komunikaciju. U sljedećoj tablici prikazana je tehnička karakteristika preuzete sa službene dokumentacije s portala Arduino [1].

Tablica 2-1 Karakteristika Arduino Mega 2560

Mikrokontroler	ATmega2560
Radni napon	5 V
Preporučeni ulazni napon	7-12 V
Granice ulaznog napona	6-20 V
Digitalni ulazno/izlazni pinovi	54 (od kojih 15 daju PWM izlaz)
Analogni ulazni pinovi	16
Istosmjerna struja po ulaznom/izlaznom pinu	20 mA
Istosmjerna struja za 3.3 V pin	50 mA
Brza memorija	256 KB (od kojih 8 KB koristi bootloader)
SRAM	8 KB
EEPROM	4 KB
Radna frekvencija	16 MHz
Ugrađen LED pin	13
Duljina	101.52 mm
Širina	53.3 mm
Težina	37 g

Programska podrška za rad sustava napravljena je u platformi Arduino IDE, koja je detaljnija opisana u poglavlju 1.1 ovog rada. Osim navedene specifikacije, za funkcionalnost ovog rada upotrijebljeni su moduli koji koriste prekidne funkcije mikrokontrolera za svoj rad. Ovaj model Arduina sadrži 6 takvih pinova, od kojih su 4 korištena za serijsku komunikaciju pa za senzore preostaju svega 2 prekidna pina. S obzirom na to da se za izradu mobilnog robota koriste 3 modula za serijsku komunikaciju i 2 senzora prekidne namjene, odabirom Arduino Mega 2560 su zadovoljeni početni zahtjevi sustava. Jedan od razloga zbog kojeg je ova razvojna pločica tako popularna je i njeno jednostavno testiranje i procedura rješavanja problema, ako je to potrebno. U platformi Arduino IDE se mogu pratiti izvršeni programski isječci i njihovi rezultati, ako se definiraju pomoćne varijable i logika koja ispisuje krajnji ishod, a sve što je potrebno jest povezati razvojnu pločicu USB kabelom s računalom.

2.2. Moduli serijske komunikacije

Za izvedbu mobilnog robota na daljinsko upravljanje, koji šalje podatke na internet kao što je GPS lokacija, potrebna su 3 modula serijske komunikacije. Daljinsko upravljanje se odvija preko bluetooth veze modulom HC-06, dok se GPS veza ostvaruje preko U-BLOX NEO-7M modula. Svi obrađeni podaci se dodjeljuju specifičnim varijablama u programskom kodu mikrokontrolera i šalju preko GPRS veze na internet pomoću SIM800L EVB modula. Podaci se asinkrono prenose od modula prema Arduino i obrnuto. Za razliku od I2C/SPI/USB komunikacije, kod serijske ili asinkrone komunikacije ne postoji predefiniрани okidni signal određenog trajanja, nego se brzina uzorkovanja definira brzinom prijenosa (eng. *baud rate*) u Arduino IDE programskom sučelju. Bitno je da prijemnik i odašiljač, koji šalju i primaju informaciju, budu podešeni s istom brzinom prijenosa jer će u suprotnom prijemnik primati beskorisne i netočne podatke, što će uzrokovati neispravan rad modula. Osim podešavanja ispravne brzine prijenosa bitova, treba obratiti pažnju na povezivanje modula. Svaki modul, koji ima mogućnost serijske komunikacije, sadrži dva pina: prijemni i odašiljački. Tako prilikom spajanja dva modula, prijemni pin jednog modula mora biti povezan s odašiljačkim pinom drugog. Ista stvar je primjenjiva za drugi pin. Prijenos informacije započinje startnim bitom „0“, zatim se pričekava 1.5 perioda vremena uzorkovanja bita te se kreće s prijenosom. Vrijeme uzorkovanja bita se može izračunati iz brzine prijenosa, koja je najčešće 9600 bita po sekundi. Stoga je vrijeme uzorkovanja jednako obrnuto proporcionalnoj vrijednosti brzine prijenosa, to jest 104.16 μ s. Primjera radi, ako šaljemo jedno slovo od 8 bitova, za slanje cijele informacije će biti potrebno 10 bitova uključujući start i stop bit. Stop bit je utvrđen kao i start bit, osim što mu je vrijednost bita jednaka „1“. Konačno, prienos jednog bita vremenski iznosi 1041.6 μ s.

2.2.1. SIM800L EVB

Danas postoje brojne kopije i varijante Arduina proizvedene od drugih proizvođača, koje imaju integrirane čipove za GSM/GPRS vezu pa čak i bluetooth komunikaciju. Zbog ograničenja takvih razvojnih pločica u smislu broja pinova, ali i zbog vrlo visoke cijene, za izradu rada su odabrane zasebne komponente. Jedna od kompleksnijih komponenti ovog sustava je SIM800L EVB. Modul sadrži čip tvrtke SimCom, koji se koristi i na gotovo svim drugim pločicama koje koriste GSM/GPRS komunikaciju. Za rad modula potrebna je važeća SIM kartica koja podržava 2G vezu. U slučaju da je umetnuta kartica mobilne mreže

operatera koji ne podržava 2G, modul neće ispravno raditi. Prednost ovog modula u odnosu na ostale jest da ima ugrađeno automatsko podešavanje brzine prijenosa simbola. U slučaju da se na Arduinu podesi specifična vrijednost, ovaj modul će se prilagoditi brzini koju Arduino pruža. Neke od ostalih funkcija SIM800L su sljedeće, koje su navedene u dokumentaciji [4]:

- Slanje i primanje SMS poruka
- Podrška quad-band GSM850, EGSM900, DCS1800 i PCS1900S
- Umetanje mikro SIM kartice i podrška za mobilne mreže koje imaju mogućnost korištenja 2G mreža
- Primanje i slanje poziva
- Slanje i primanje GPRS podataka (preko protokola HTTP, TCP/IP)
- Koristi AT set naredbi za upravljanje



Slika 2.1 SIM800L EVB modul

Na gornjoj slici je prikaz modula s prednje i stražnje strane. Komponenta na slici označena s J5 predstavlja utor za eksternu antenu, koju je preporučljivo montirati jer se time postiže jači signal i stabilniji rad komponente. Modul je poželjno ugraditi na mjesto gdje će signal biti jači. Najčešće je to položaj na izbočenom dijelu robota dalje od drugih elektroničkih komponenti. Sklop mora biti napajan s približno 5V, inače dolazi do njegovog nepoželjnog rada koje se manifestira s čestim gašenjem i ponovnim pokretanjem. Opće je poznato da prilikom ostvarenja GSM ili GPRS veze dolazi do velikog propuštanja struje kroz modul, stoga je potrebno osigurati napajanje modula koje osigurava rad na zadanim specifikacijama. Potrošnju kod različitih načina rada je moguće vidjeti u sljedećoj tablici preuzetoj iz dokumentacije tvrtke SimCom [4].

Tablica 2-2 Strujna karakteristika SIM800L EVB modula

Načini rada	Frekvencija	Potrošnja struje
<i>Power down</i>		60 μ A
<i>Sleep</i>		1 mA
<i>Stand by</i>		18 mA
<i>Call</i>	GSM850	199 mA
	EGSM900	216 mA
	DCS1800	146 mA
	PCS1900	131 mA
GPRS		453 mA
Vršna vrijednost struje kod prijenosa podataka		2 A

U usporedbi s ostalim sensorima i modulima, struja koja prolazi kroz SIM800L EVB nekoliko je puta veća. Zato se savjetuje koristiti vanjsko napajanje koje će napajati isključivo ovu komponentu. Baterije od litija su prikladne za ovu namjenu jer je njihov napon koje mogu dati približno 4 volta, a kombinacijom dvije baterije u seriji osigurava se potrebna visina napona, kao i kapacitet baterija koji omogućuje da sklop radi dulje vrijeme bez potrebe za zamjenom baterije. Često se u kombinaciji s dvije litij baterije koristi i sklop za regulaciju napona kako bi napon na SIM modulu bio stabilnih 5 volta, inače bi sklop odmah prestao raditi ako bi se napajanje od dvije serijski spojene baterije izravno spojilo na modul.

2.2.2. Bluetooth HC-06

Za daljinsko upravljanje mobilnog robota koristi se bluetooth modul HC-06. Najčešće se koristi u projektima za daljinsko upravljanje ili kod slanja manjih skupova podataka. Ono što ograničava njegovu uporabu je domet u kojem tehnologija radi. Domet ovisi o karakteristikama prostora u kojem je HC-06 prisutan, ali najčešće se radi o dometu od 10 do 15 metara. Modul pripada klasi 2 čija je maksimalna snaga 2.5 milivata. Također, koristi specifikaciju Bluetooth 2.0 koja omogućava prijenos podataka u brzini do 3Mbit/s. Opće je poznato da koristi frekvenciju od 2.402 do 2.480 gigaherca, ali mnogi ne znaju da je zbog emitiranja signala niske snage od jednog milivata omogućen nesmetan rad modula jer se tako sprječavaju moguća međudjelovanja s drugim izvorima iste frekvencije. Kako bi se modul integrirao u projekt mobilnog robota, nužno je pripaziti na brzinu slanja podataka jer se radi o robotu koji reagira na trenutnu pobudu uzrokovanu od strane korisnika. Za daljinsko

upravljanje koristi se Android aplikacija, koja u pozadini šalje naredbe preko Bluetootha. Stoga se za veću brzinu prijenosa podataka za određenu naredbu šalje samo jedan znak, u protivnom bi interakcija korisnika s aplikacijom bila narušena, jer bi robot prepoznao naredbu tek nakon nekog vremena kašnjenja. Kašnjenje je također moguće smanjiti optimizacijom programskog koda sa strane Android aplikacije. HC-06 je za razliku od svog prethodnika HC-05, *slave only*, što omogućava upravljanje preko aplikacije. Slično kao i kod SIM800L EVB, konfiguracija modula je moguća preko AT upravljačkih naredbi. Tako se na primjer može podesiti brzina prijenosa simbola (eng. *baud rate*). Ista brzina mora biti definirana u programskom kodu Arduino mikrokontrolera. Iako je rukovanje s ovom Bluetooth komponentom vrlo jednostavno, treba obratiti pozornost na njenu tehničku mogućnost i specifikaciju [5]:

- Bluetooth protokol: Bluetooth V2.0 standard
- Razina snage: Class2(+6 dBm)
- Frekvencijski pojas: 2.40—2.48 GHz, ISM Band
- Osjetljivost prijemnika: -85 dBm
- Tip modulacije: Gauss frequency Shift Keying
- Raspon radnog napona:+3.3V to +6V
- Raspon radne temperature: -20°C to +55°C
- Radna struja: 40 mA

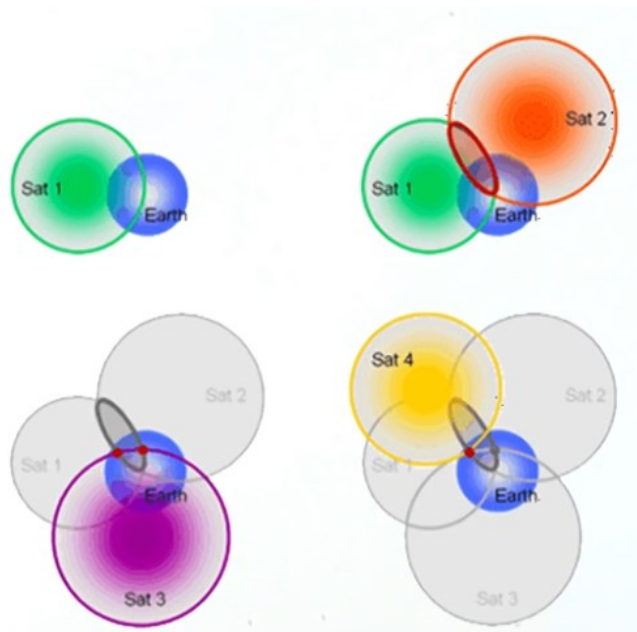


Slika 2.2 Bluetooth HC-06 modul

2.2.3. GPS U-BLOX NEO-7M

Danas vlasnici automobila, ali i drugih skupocjenih mobilnih stvari, žele u bilo kojem trenutku imati kontrolu nad njima. Zato se industrija zvana Internet stvari smatra sljedećim velikim tržištem u koje poduzetnici, ali i svi drugi koje zanima tehnologija, žele uložiti svoj novac kako bi stekli tu željenu kontrolu. Postoje brojne informacije koje uređaji odašilju prema korisnicima u sklopu obrade podataka. Jedna od takvih informacija je GPS položaj nekog objekta. Svi moderniji mobilni telefoni i prenosivi elektronski uređaji danas imaju ugrađen čip, koji omogućuje vlasniku praćenje preko GPS lokacije. Za izradu mobilnog robota u ovom radu u tu svrhu se koristi modul U-BLOX NEO-7M. Ovaj modul, kao što ime govori, sadrži NEO-7M čip napravljen od tvrtke U-BLOX. Osim toga, na pločici se nalazi ugrađena visokofrekvencijska antena, IPX sučelje za povezivanje dodatnih eksternih antena i punjiva sigurnosna baterija, koja čuva informacije u slučaju prekida napajanja. Preporučljivo je koristiti softver U-center prije nego se modul spoji s Arduinom jer se ovako mogu podesiti postavke kako bi modul ispravno radio prilikom serijske komunikacije s Arduinom. S druge strane, radi se o komunikaciji modula sa satelitima postavljenim u Zemljinoj orbiti, stoga je potrebno nekoliko minuta kako bi modul pronašao najprikladnije satelite za određivanje lokacije. U GPS sustavu, koji pokriva Zemlju, sudjeluju minimalno 24 satelita. Za rad modula i izračun 2D pozicije, odnosno geografske širine i duljine, GPS prijammnik u modulu mora primiti informacije od minimalno 3 satelita. Ako se primaju informacije s više od 3 satelita, moguće je odrediti i visinu lokacije. Sateliti odašilju barem 2 signala niske snage prema Zemlji i ti signali bez problema prolaze kroz oblake, plastiku, stakla i slične tvari. Ono što je problematično, signal se gubi ako nailazi na krute tvari kao što su betonske zgrade, drveća ili planine. Zato je za izradu mobilnog robota potrebno odabrati najizloženije mjesto kako bi signal neometano došao na modul. Prilikom povezivanja modula s računalom, on automatski započinje s procesom primanja podataka od satelita. GPS signal sadrži 3 tipa podataka koja se šalju. Jedan od njih je broj, tj. identifikator GPS satelita, koji pokazuje koji od satelita odašilje signale prema modulu. Drugi podatak sadrži informaciju o vremenu i datumu, kao i stanju satelita koji šalje signal. Posljednji podatak odaslan prema modulu uključuje informaciju koja govori modulu gdje bi se koji satelit trebao nalaziti u određeno vrijeme u tom danu. Ti podaci se obrađuju te se uspoređuje vrijeme koje je poslao satelit, s vremenom kojeg Arduino ili računalo prikazuju preko USB sučelja. Tako se na temelju razlike vremena može utvrditi koliko je satelit udaljen od lokacije koja se određuje. Kombinacijom više satelita odjednom, mogu se vidjeti presjeci

orbita koje daju sateliti i na temelju toga odrediti točna lokacija. Ovaj princip je prikazan na sljedećoj slici gdje je vidljivo kako se lokacija može dobiti presjekom minimalno 3 orbite satelita. Ako 3 satelita šalju signal, lokacija nije sasvim poznata jer postoje dvije mogućnosti. Unatoč tome, ako je modul dovoljno izložen signalu, signal će sigurno doći od barem 4 satelita kako bi se odredila ispravna lokacija.



Slika 2.3 Određivanje lokacije preko broja satelita

Osim podataka poput geografske duljine i širine, moguće je odrediti druge parametre na temelju udaljenosti između dva očitavanja, kao na primjer pređenu udaljenost ili brzinu objekta ako je u gibanju. Uz ove podatke, modul ima još nekoliko funkcionalnosti [6]:

- Preko serijskog ulaza moguće je podesiti parametre i spremiti ih u EEPROM memoriju
- SMA konektor podržava spajanje različitih vrsta antena
- Kompatibilan sa 3.3/5 V naponskim razinama
- Sadrži ugrađenu keramičku antenu za primanje slabih GPS signala
- Integrirano USB sučelje za lako testiranje i ispitivanje sklopa
- Brzina prijenosa simbola podesiva od 9600 do 115200 simbola u sekundi
- Uključena rezervna baterija koja omogućava brzo paljenje ako dođe do kratkotrajnog nestanka napajanja
- Omogućeno korištenje GPS, GLONASS, QZSS i SBAS sustava



Slika 2.4 GPS U-BLOX NEO-7M

2.3. Senzori

Osim modula serijske komunikacije, za implementaciju mobilnog robota potrebni su i senzori čiji je rad djelomično jednostavniji. Radi se o jednosmjernoj komunikaciji između senzora i mikrokontrolera, gdje senzor zahtjeva napajanje da bi ispravno radio. Postoje analogni i digitalni senzori, odnosno senzori koji daju digitalni izlaz „0“ ili „1“ ili kontinuiranu vrijednost izvan tih okvira. U ovom radu su korištena dva senzora s digitalnom izlazom: ultrazvučni senzor HC-SR04 i optički senzor LM393. Ono što odlikuje većinu senzora je vrlo mala struja koja je potrebna za njihov kontinuiran rad.

2.3.1. Senzor brzine LM393

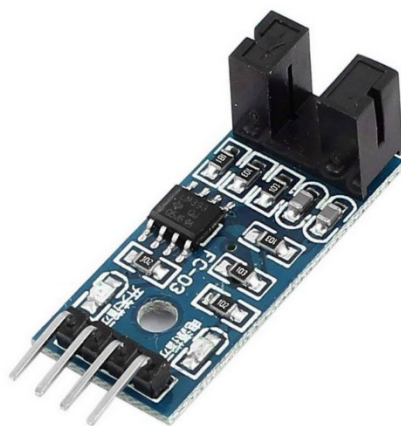
Ideja izrade mobilnog robota je prenijeti što je moguće više informacija korisniku. Zbog toga što je robot izrađen u obliku automobila na daljinsko upravljanje, postoji mogućnost za mjerenje njegove brzine [13]. Na kotače automobila postavljeni su pretvornici pomaka napravljeni od plastike. Princip rada ovog senzora je vrlo jednostavan: optički senzor sadrži prostor između kojeg se namjesti pretvornik pomaka ili enkoder. Okretanjem motora okreće se i enkoder. Plastični enkoder ima određen broj rupa po kojem optički senzor prepoznaje kada je plastični dio prošao kroz senzor ili kada je „svjetliji“ dio prošao kroz senzor. Svakim prolaskom svjetla kroz senzor stvara se digitalni puls na izlaznom digitalnom pinu senzora. Taj puls zatim prelazi s niske razine i 0 volti na visoku razinu, tj. 5 volti. Ono što mikrokontroler mora odraditi u ovom procesu je izbrojati koliko se brzo ti impulsi

ponavljaju. Na temelju broja impulsa u određenom vremenu i zahvaljujući tome što je poznato koliko točno rupa sadrži enkoder, može se ustanoviti koliko se brzo okreće motor u određenom vremenskom intervalu. Također, ako se izračuna radijus kotača, može se izračunati koliko udaljenost pređe pojedini kotač u jednom okretu. Stoga je preporučljivo postaviti barem dva senzora na dva različita kotača kako bi mjerenje bilo što točnije. Kao i s ostalim modulima i sensorima, treba obratiti pažnju na tehničku specifikaciju komponente, koja je prikazana u sljedećoj tablici na temelju podataka iz dokumentacije [7].

Tablica 2-3 Tehnička karakteristika LM393

Komparator napona	LM393
Radni napon	3.3-5 V
Širina utora za osjetilo pomaka	5 mm
Izlazni oblik pulsa	Digitalni (0 ili 1)
Aplikacija	Mjerenje brzine pomaka
Dimenzije	32x14x7 mm
Struja potrebna za rad	15 mA

Uz navedenu specifikaciju, senzor ima dva ugrađena LED indikatora, koja olakšavaju analizu rada komponente. Jedan svijetli kontinuirano crvenom bojom ako je prisutno napajanje u zadanom okviru prema karakteristici, dok drugi svijetli zelenom bojom ako dolazi do prolaska svijetlih dijelova kroz senzor. U tom trenutku se aktivira prekidna funkcija na mikrokontroleru i brojilo prekida se uveća za 1. Programski kod je potrebno podesiti ovisno o tome primjenjuje li se napon od 3.3 ili od 5 volti, jer su očitavanja proporcionalna dovedenom naponu.



Slika 2.5 Senzor brzine LM393

2.3.2. Ultrazvučni senzor HC-SR04

Ultrazvučni senzori se često koriste kod dizajna i izrade mobilnih robota. Glavna namjena im je određivanje udaljenosti od najbliže prepreke, a zatim se na temelju te informacije definira logika u programskom kodu mikrokontrolera. Kao i mnogi drugi senzori, ultrazvučni senzor svoj rad zasniva na fizikalnim zakonima. U prvom koraku procesa određivanja udaljenosti od prepreke, šalje se signal visoke frekvencije u rasponu od 25 do 50 kHz. Nakon što se signal poslao, senzor čeka da se signal reflektira od nekog objekta i vrati nazad. Modul u pozadini mjeri razliku vremena između trenutka kada je signal poslan i trenutka kada se signal vratio. Na temelju ta dva vremena, fizikalno se određuje udaljenost od objekta. Za referentnu brzinu gibanja ultrazvučnog signala uzima se vrijednost od 343 m/s, koja je približno jednaka brzini gibanja zvuka. Ova vrijednost vrijedi za sobnu temperaturu od 20 Celzijevih stupnjeva. Stoga se na temperaturama koje odstupaju sobnoj očekuju greške od vrijednosti nekoliko posto. Zbog jednostavnog načina rada, ovakvi senzori nisu skupi te koštaju svega nekoliko desetaka kuna. Postoje i drugi modeli senzora, koji automatski prilagođavaju parametre poput brzine zvuka u različitim uvjetima jer imaju ugrađen senzor za mjerenje temperature. Sličan princip rada koriste radari, ali za razliku od njih, ovi senzori bolje reagiraju ako se signal reflektira od određenih materijala, kao što je na primjer plastika. S druge strane, ako je objekt neravan, signal će se najčešće odbiti u izmijenjenom smjeru i neće se vratiti senzoru. Ako se to dogodi, senzor će pretpostaviti da je prepreka predaleko, a mikrokontroler će za informaciju o udaljenosti od prepreke dobiti maksimalnu vrijednost, što je netočan podatak. Jedan od razloga zašto se nerijetko koristi ultrazvučni senzor je zbog činjenice da nekad razvijeni infracrveni senzori, koji su se koristili za sličnu svrhu, lošije rade ako se signal odbija od materijala crne boje ili ako signal prolazi kroz dim. Za potrebe ovog rada koristi se jedan ovakav modul. Modul se često integrira u projekt sa servomotorom. To koriste autonomni roboti, koji se samostalno kreću bez potrebe za daljinskim upravljanjem. U tom slučaju, ako robot naiđe na prepreku koja se nalazi na graničnoj udaljenosti, uključuje se servomotor koji okreće senzor i pamti se vrijednost udaljenosti prilikom tog okreta. Mikrokontroler raspolaže tim informacijama i donosi odluku u koju stranu će robot skrenuti. U ovom projektu se ne koristi servomotor nego se informacije dobivaju od jednog senzora koji promatra udaljenost od prepreke s prednje strane mobilnog robota.

U sljedećoj tablici prikazana je specifikacija ovog senzora koja jasno ukazuje za koju vrstu projekta je preporučljivo koristiti HC-SR04 [8].

Tablica 2-4 Karakteristika HC-SR04 modula

Radni napon	3.3-5V
Struja mirovanja	<2 mA
Radna struja	15 mA
Kut prepoznavanja primljenog signala u odnosu na odaslani	<15 stupnjeva
Raspon detekcije udaljenosti	2 – 400 cm
Rezolucija udaljenosti	0.3 cm
Frekvencija odaslanoog signala	40 kHz
Trajanje okidnog impulsa signala	10 μ s
Dimenzija	45x20x15 mm

Rad ovog senzora mora biti vrlo brz jer se želi postići da robot na vrijeme procjeni kada je prepreka blizu i da se zatim zaustavi. Budući da se signali šalju vrlo brzo, u trajanju signala reda veličine 10 μ s, može se očekivati da se unutar jedne sekunde dogodi više desetaka očitavanja senzora, ovisno o optimizaciji koda. Tako je osigurano pravovremeno zaustavljanje robota. Na sljedećoj slici je prikazano kako izgleda ultrazvučni senzor HC-SR04.



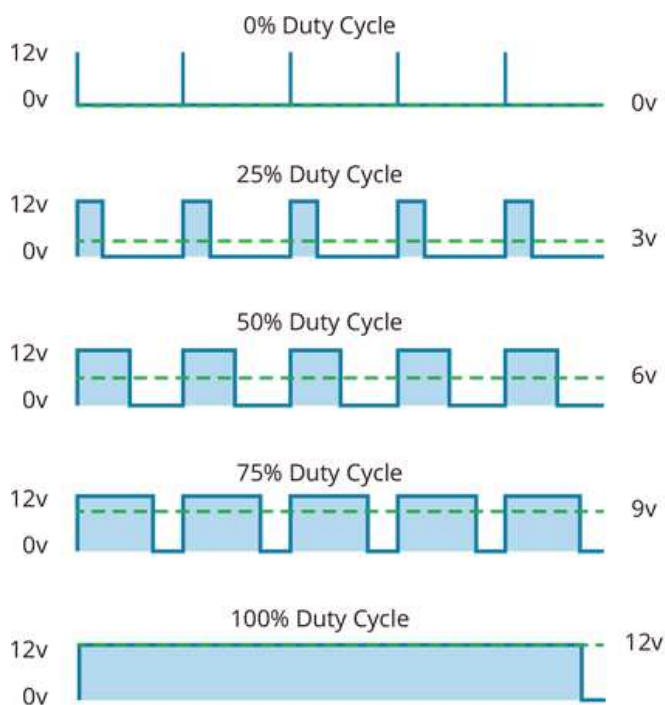
Slika 2.6 Ultrazvučni senzor HC-SR04

Osim pinova za napajanje i uzemljenje, senzor sadrži još dva pina, *Trig* i *Echo*. Pomoću *Trig* pina se šalje ultrazvučni val prema prepreci. Naredbu i programsku logiku koliko često se

šalje signal određuje mikrokontroler. *Echo* pin generira impuls kada se odaslani signal vrati. Trajanje tog impulsa je jednako vremenu koje je trebalo odaslanom signalu da postane detektiran.

2.3.3. DC Motor Driver Dual H-Bridge L298N

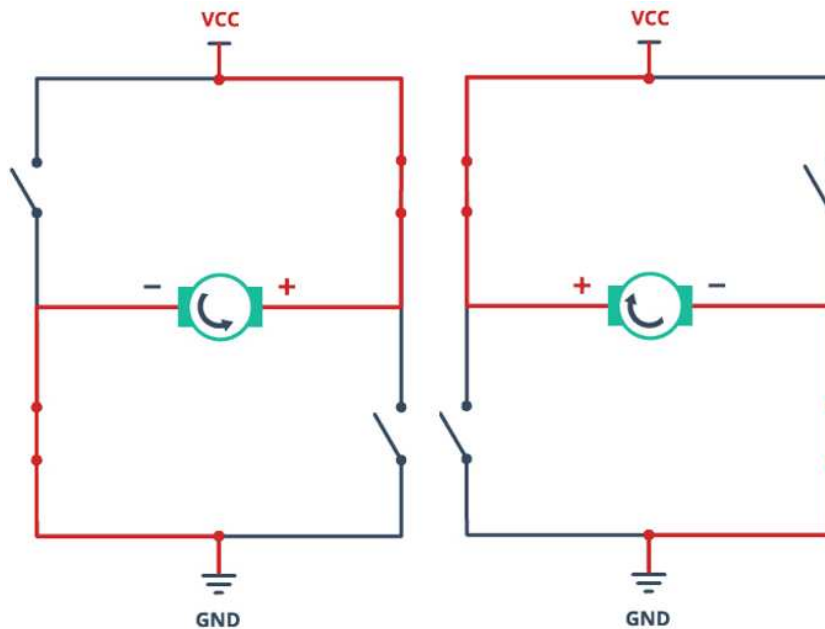
Uz module serijske komunikacije i senzora koji šalju različite informaciju prema glavnoj upravljačkoj jedinici, za upravljanje mobilnog robota potreban je modul koji će osigurati pravilno upravljanje motorima. Za tu svrhu se koristi modul L298N. Ova komponenta predstavlja sučelje između motora i mikrokontrolera te omogućava mikrokontroleru da, na temelju definiranog programskog koda, upravlja motorima. Upravljanje se postiže uključivanjem i isključivanjem motora, promjenom brzine vrtnje i promjenom smjera kretanje motora. Modul kombinira tehnike PWM (eng. *Pulse width modulation*) i H-Bridge. PWM tehnika dozvoljava podešavanje trenutne brzine motora koristeći srednju vrijednost napona. Princip rada PWM je prikazan na sljedećoj slici:



Slika 2.7 PWM tehnika u L298N modulu

Kao što je prikazano na slici, u slučaju male brzine motora, signal koji se šalje prema motoru će većinu vremena biti u niskoj razini. Primjerice, za slučaj u kojemu je maksimalna vrijednost napajanja motora 3 volta, signal će biti u visokoj razini svega 25% perioda signala. U jednom takvom periodu signala, modul računa srednju vrijednost napona i tako ograničava brzinu motora. Druga tehnika je H-Bridge koja kontrolira smjer kretanja motora. Modul

sadrži četiri sklopke, a njihovim zatvaranjem i otvaranjem moguće je obrnuti polaritet napona kojim su motori napajani. Naredbom koja je poslana od strane mikrokontrolera kontrolira se smjer kretanja, a ako se on promjeni, događa se promjena u načinu rada sklopki kao što je prikazano slikom:



Slika 2.8 H-Bridge način rada

Dual H-Bridge L298N dopušta upravljanje s dva motora istovremeno pa su za izradu ovog rada korištena dva takva modula jer se robotom upravlja s 4 motora. Specifikacija u sljedećoj tablici [9] opisuje karakteristiku modula i koji je minimalan napajanje potrebno dovesti da bi sklop ispravno radio.

Tablica 2-5 Karakteristika Dual H-Bridge L298N

Napon za rad modula	5 V
Napon za upravljanje motorima	5-35 V
Maksimalna struja upravljanja	2 A
Raspon struje za logičko upravljanje	0-36 mA
Maksimalna izlazna snaga	25W
Dimenzija	43x43x26 mm

Za napajanje ove komponente potreban je izvor koji će pružati potrebnu struju od 2A, a to je moguće postići korištenjem dviju litijevih baterija nazivnog napona od 3.7V.

2.4. Napajanje sustava

Napajanje je vrlo bitan dio cijelog sustava jer u slučaju odabira nedovoljnog napajanja, sustav neće pravilno raditi. Većina senzora i modula koji su korišteni u ovom projektu, ne zahtijevaju veliku struju za rad. Modul kojemu je potrebna najveće struja je SIM800L EVB, čija maksimalna struja može doseći čak 2 ampera. Uz njega, za kretanje robota koriste se 4 istosmjerna motora, koja se mogu upravljati strujom od 0.2 do 2 ampera. Stoga se ove komponente moraju izravno napajati iz baterija, a ne preko pinova iz mikrokontrolera. Arduino zahtijeva napajanje od minimalno 7V, dok L298N modul za upravljanje motora treba barem 5V za rad. Na temelju tih potreba, za napajanje sustava koriste se 18650 litijeve baterije nazivnog napona od 3.7V. Ove baterije, kada su u potpunosti napunjene, daju napon od 4.2V. Spajanjem dviju takvih baterija u seriju postižu se uvjeti za ispravan rad sustava. U radu će se koristiti ukupno četiri ovakve baterije, dvije koje će napajati mikrokontroler i L298N modul, a ostale dvije su potrebne za rad SIM800L modula. Na sljedećoj slici je prikazana baterija korištena u projektu te je u tablici opisana tehnička karakteristika.



Slika 2.9 18650 litijeva baterija

Tablica 2-6 Karakteristika litijeve 18650 baterije

Kapacitet	2900 mAh
Maksimalna izlazna struja	10 A
Napon	3.7 V (4.2 V kada je u potpunosti napunjena)
Masa	48 g
Tip baterije	NMC, 18650
Punjiva	da

3. Implementacija i povezivanje sustava mobilnog robota

3.1. Opis problema i motivacija

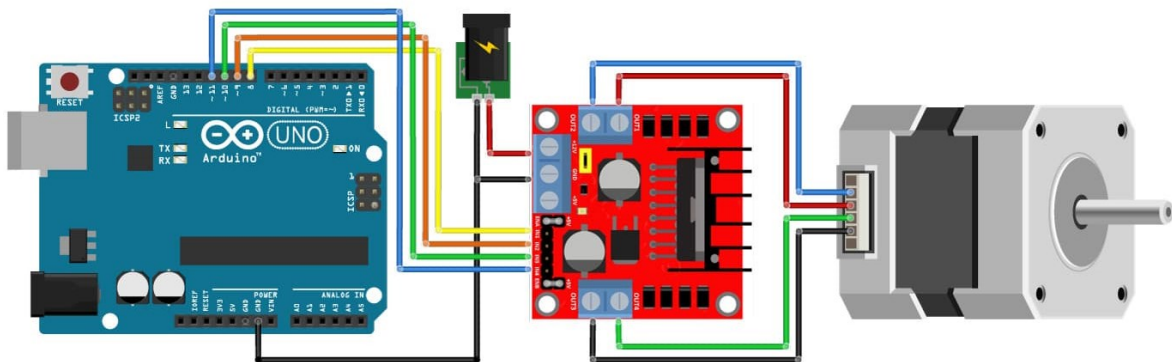
U 21. stoljeću dolazi do velike segmentacije u obavljanju svakodnevnih zadataka i to dovodi do specijalizacije mnogih poslova u ljudskom životu. Danas gotovo svaka tvrtka, koja ima trend rasta zaposlenih, formira nove odjele i dovodi ljude specijalizirane za odrađivanje određenih zadataka. Ono što je jedan od najvećih izazova sličnih velikih i rastućih sustava, jest održavanje stabilnosti u procesima svakodnevnog rada. Brojna istraživanja su pokazala da je s vremenom sve više rutinskih poslova koji oduzimaju vrijeme, a to vrijeme se može iskoristiti za razvoj novih ideja i procesa. Motivacija izrade ovog projekta je upravo ušteda vremena za ponavljajućim procesima i automatizacija prikaza podataka na što efikasniji način za krajnjeg korisnika [12]. Iako se ovaj rad temelji na izradi mobilnog robota na daljinsko upravljanje, ideja rada je pokazati automatizaciju nekoliko procesa istovremeno primjenjujući najnovije tehnologije iz industrije *Internet of Things*. Ovaj projekt, osim što pokazuje na koji način je moguće ostvariti automatizaciju, oslikava i nekoliko stvarnih slučajeva današnjice. Primjerice, sve je više tvrtki koje u svom posjedu imaju vozila kao što su kamioni, automobili, motocikli i slično. Teško je ostvariti ili zaposliti nekog da prati sve parametre nekih vozila koja su u pokretu u određenom trenutku, kao na primjer: GPS lokaciju, prosječnu brzinu automobila, ukupnu pređenu udaljenost u određenom vremenskom intervalu. Jedan od ciljeva ovog rada je redovito slanje sličnih podataka u bazu, koja se nalazi na internetu. Koristeći takve podatke moguće je generirati statistiku, grafove i tablične prikaze koji mogu dati uvid krajnjem korisniku u moguću potrošnju i tako olakšati planiranje i utrošak budžeta u nekom vremenskom periodu. Procjenjuje se da je do 2020. godine umreženo više od 30 milijardi raznih uređaja i senzora čiji se podatci mogu iskoristiti za razne analize [11]. Nova istraživanja pokazuju da će se u sljedećih 5 godina taj ukupan broj umreženih uređaja udvostručiti na preko 70 milijardi uređaja. Danas je gotovo nemoguće pronaći kompaniju koja ne ulaže u digitalnu transformaciju i koja nije barem jednim dijelom svog poslovnog modela uključena u tehnologiju Interneta stvari [10].

3.2. Načelo rada sustava

Rad sustava moguće je podijeliti u dvije cjeline. Prva cjelina predstavlja upravljanje robotom preko mobilne aplikacije bluetooth vezom, a druga rad robota u mirovanju ili u slučaju zahtjeva korisnika. Oba načina rada imaju većinu istih funkcionalnosti, ali je u sljedećim poglavljima detaljno objašnjeno kako sustav funkcionira u određenim fazama.

3.2.1. Sustav u stanju upravljanja mobilnom aplikacijom

Kako bi mobilni robot postao pokretljiv, potrebno je povezati i implementirati logiku koja osigurava kontrolu gibanja motora i dijelova zaduženih za njegov rad. Za tu svrhu, osim mikroupravljača Arduino Mega 2560, koriste se dva H-Bridge L298N modula i četiri istosmjerna motora koja osiguravaju mobilnost robota. Shema fizičkog spoja ovih modula je prikazana na sljedećoj slici:



Slika 3.1 Shema spajanja motora s Arduinoom

L298N modul sadrži digitalne pinove koji osiguravaju upravljanje motora, točnije promjenu smjera vrtnje motora i kontrolu njihove brzine. Navedena definicija mora se programski izvesti u sučelju Arduino IDE na sljedeći način:

```
#define ENABLE_PD 5
#define L298N_PD_1 2
#define L298N_PD_2 3
```

Svaki od četiri motora je definiran na prikazan način. Prva definicija prikazuje kojem pinu mikrokontrolera pripada upravljačka naredba, koja određuje brzinu gibanja prednjeg desnog motora. Ostale dvije linije koda opisuju kojem pinu odgovaraju postavke za smjer okretanja motora. L298N se regulira na sljedećem principu: ako je jedan od ova dva pina u stanju „HIGH“, tada će se motor kretati u određenom smjeru. Kada se zamijene stanja ta dva pina, odnosno kada jedan pređe u stanje „LOW“, a drugi u stanje „HIGH“, tada se motor giba u

suprotnom smjeru. Konačno, ako su oba pina u stanju „LOW“, motor se zaustavlja. Analogno tome, definiraju se pinovi za preostala tri kotača koja sudjeluju u kretanju mobilnog robota. Osim povezivanja pinova sa svakim od motora, neophodno je definirati svih osam mogućih smjerova kretanja vozila u programskom kodu. Primjerice, za kretanje vozila prema naprijed, napravljena je funkcija koja se poziva u trenutku kada korisnik u mobilnoj aplikaciji pritisne gumb koji uzrokuje kretanju prema naprijed. Ono što se dogodi opisano je sljedećim programskim isječkom:

```
void naprijedRavno() {
    digitalWrite(L298N_PD_1, HIGH);
    digitalWrite(L298N_PD_2, LOW);
    analogWrite(ENABLE_PD, brzinaAuta);

    digitalWrite(L298N_SR_1, LOW);
    digitalWrite(L298N_SR_2, HIGH);
    analogWrite(ENABLE_SD, brzinaAuta);

    digitalWrite(L298N_PL_1, LOW);
    digitalWrite(L298N_PL_2, HIGH);
    analogWrite(ENABLE_PL, brzinaAuta);

    digitalWrite(L298N_SL_1, HIGH);
    digitalWrite(L298N_SL_2, LOW);
    analogWrite(ENABLE_SL, brzinaAuta);
}
```

Kod 3.1 Funkcija za kretanje motora u jednom smjeru

Ova funkcija se sastoji od četiri manja dijela, a svaki od njih je pridružen jednom motoru. Ovisno kako su spojene žice svakog od motora, tako su definirane i varijable u kodu. Bitno je primijetiti da je u ovom slučaju uvijek jedna varijabla za kretanje u „HIGH“, a druga u „LOW“ stanju, što ukazuje da su sva četiri motora aktivna ako se robot giba prema naprijed. Lako je povezati da se na isti način mogu definirati ostalih 7 smjerova kretanja. Uvođenjem svih 8 smjerova kretanja umjesto osnovnih 4, postiže se fleksibilniji sklop i omogućuje se lakše upravljanje vozilom preko mobilne aplikacije jer se tako dočarava stvarno kretanje robota umjesto tzv. tenkovskog gibanja. Iako se kotači ne mogu okretati lijevo ili desno, pravovremenim paljenjem i gašenjem određenih motora moguće je postići efekt kosog kretanja.

Osim motora potrebnih za upravljanje vozilom i programskog dijela, koji definira različite smjerove kretanja, nužno je odrediti poveznicu između upravljanja preko mobilne aplikacije i implementacije programske logike u Arduino mikrokontroleru. Komunikacija između ta dva dijela sustava postiže se preko bluetooth modula HC-06. U ovom poglavlju je opisana komunikacija modula s mikrokontrolerom Arduino Mega 2560, a u sljedećem poglavlju je detaljno objašnjen postupak izrade mobilne aplikacije u platformi App Inventor. Mobilni robot koristi Android aplikaciju preko koje korisnik, pritiskom na gumb, mijenja smjer i brzinu kretanja. Korištenje aplikacije nije moguće bez ostvarene bluetooth veze. Prilikom ostvarenja veze, svakim pritiskom gumba s mobilnog telefona se šalje znak koji pripada određenoj tipki. Bluetooth modul prihvaća taj znak i serijskom komunikacijom predaje vrijednost znaka prema mikrokontroleru da bi on znao koja je sljedeća akcija koju treba obaviti. Za definiciju varijabli bluetooth modula potrebne su sljedeće varijable:

```
int BT_komanda;  
int brzinaAuta = 150;    // raspon od 50 - 255.  
int brzinaKoefficient = 4;
```

Varijabla `BT_komanda` predstavlja znak koji je poslan u trenutku kada korisnik aplikacije pritisne neki gumb. Ostale dvije navedene varijable se koriste za slanje podatka o brzini vrtnje motora prema L298N modulu, koji zatim podešava odgovarajući napon za napajanje motora. Varijabla `brzinaKoefficient` koristi se u slučaju kada se robot kreće ukoso jer se u tom slučaju želi postići da se neki motori kreću manjom brzinom nego ostali, kako bi se ostvario željeni efekt. S programske strane, potrebno je osigurati ispravnu serijsku komunikaciju između HC-06 modula i Arduina:

```
void setup() {  
    Serial1.begin(9600); // bluetooth modul  
    Serial3.begin(9600); // sim800 modul  
    Serial2.begin(9600); // GPS modul  
}
```

Ovime se utvrđuje brzina prijenosa simbola između Arduina i određenog modula. Većina komponenata s mogućnošću serijske komunikacije ima zadane i ugrađene postavke za prijenos bitova informacije pri brzini od 9600 bit/s. Ako brzina prijenosa bitova nije ista u programskom kodu i u čipu modula serijske komunikacije, doći će do pogrešnog prijenosa znaka i određene funkcionalnosti neće ispravno raditi. Nakon što su definirani pinovi za komunikaciju i brzina prijenosa simbola, treba specificirati izvršnu beskonačnu petlju koju

će mikrokontroler obavljati. Petlja se neprestano izvršava bez zastoja u kodu. Primjer koji prikazuje izgled petlje je prikazan u sljedećem kodu:

```
void loop() {
  if (Serial1.available() > 0) {
    BT_komanda = Serial1.read();
    stopRobot(); //inicijalizacija motora
    switch (BT_komanda) {
      case 'G':naprijedRavno();break;
      case 'N':nazadRavno();break;
      case 'L':lijevo();break;
      case 'D':desno();break;
      case 'E':naprijedDesno();break;
      case 'Q':naprijedLijevo();break;
      case 'C':nazadDesno();break;
      case 'Y':nazadLijevo();break;
      case '0':brzinaAuta = 100;break;
      case '4':brzinaAuta = 160;break;
      case '6':brzinaAuta = 190;break;
      case '8':brzinaAuta = 220;break;
      case '9':brzinaAuta = 255;break;
      case 'S':stopRobot();break;
    }
  }
}
```

Kod 3.2 Izvršna petlja za daljinsko upravljanje motora

Iz ovog programskog isječka je moguće vidjeti da će se uvjet izvršiti samo ako je neki podatak, odnosno gumb u aplikaciji, stvarno pritisnut. Ako to nije slučaj, vrijednost `Serial1.available` će biti -1 te se petlja preskače. Osim tog uvjeta, robot je moguće unaprijediti tako da mu se ugradi ultrazvučni senzor HC-SR04. Senzor je vrlo jednostavno koristiti i povezati s mikrokontrolerom. Potrebne varijable za rad senzora su opisane na sljedeći način:

```
#include <NewPing.h>
#define trig_pin A1
#define echo_pin A2
#define max_udaljenostPrepreke 200
int udaljenostOdPrepreke = 0;
NewPing sonar(trig_pin, echo_pin, max_udaljenostPrepreke);
```

Varijable `trig_pin` i `echo_pin` su nužne za odašiljanje i primanje reflektiranog ultrazvučnog signala, a detaljnija uloga ovih dviju pinova objašnjena je u 2.3.2 poglavlju, koje opisuje rad ovog senzora. Senzor u sustavu mobilnog robota ima zaštitnu ulogu. Ako se robot previše približi prepreci, robot će automatski prekinuti gibanje. To se postiže neprestanim očitanjem udaljenosti od prepreke koje je opisano u beskonačnoj petlji koju izvodi mikrokontroler.

```
udaljenostOdPrepreke = readPing();
    if(udaljenostOdPrepreke<=40){
        if ((BT_komanda == 'G') || (BT_komanda == 'Q') ||
(BT_komanda == 'E')){
            stopRobot();
        }
    }
```

`readPing` je vrlo jednostavna funkcija koja poziva drugu funkciju integriranu u biblioteku `NewPing.h`. Ovo je primjer njene strukture:

```
int readPing(){
    int cm = sonar.ping_cm();
    if (cm==0 || cm>250){
        cm=250;
    }
    return cm;
}
```

Kod 3.3 Funkcija za određivanje udaljenosti od prepreke

Zbog optimizacije ove biblioteke, prilikom poziva funkcije unutar nje, ne dolazi do kašnjenja u kodu, što je iznimno bitno kod projekta kao što je ovaj jer je važno da sustav pravovremeno reagira na opasnosti. Kao što je prikazano u gornjem programskom kodu, robot provjerava postoji li prepreka na udaljenosti manjoj od 40 centimetara. Ako postoji, korisniku je ograničeno kretanje vozila u 3 smjera koja uključuju kretanje prema naprijed. Stoga je naveden uvjet gdje se ispituju 3 mogućnosti znaka, koji je poslan nakon što je korisnik pritisnuo na određeni gumb. Korisnik aplikacije ima mogućnost kretanja u preostalih 5 smjerova, a ako pritisne gumb jednog od 3 zabranjena smjera, robot se neće pokrenuti. Ultrazvučni senzor ima ograničenje u udaljenosti prepreke koje može uočiti. Preporučljivo ga je koristiti na udaljenostima manjim od 200 centimetara. Ako se dogodi da se robot nađe na vrlo maloj udaljenosti od prepreke, moguće je da senzor vrati vrijednost 0. Identična situacija vrijedi ako je prepreka predaleko, tada vrijednosti postaju četveroznamenkaste.

Stoga se zbog ovog ograničenja definira uvjet u `readPing` funkciji koji eliminira pojavu takvih vrijednosti.

3.2.2. Slanje podataka na zahtjev korisnika

Iako korisnik ima uvid u podatke sustava preko Google Spreadsheet-a, nekada se nije prikladno povezivati na Internet kako bi se došlo do informacija. Za tu svrhu, u sklopu ovog projekta, napravljena je mogućnost dobivanja podataka na zahtjev korisnika preko SMS poruke. Svima je poznato da je SMS najrasprostranjeniji komunikacijski kanal i medij današnjice, unatoč tome što ljudi sve više koriste druge digitalne usluge kao što su WhatsApp ili Viber. Osobina SMS-a koju ni jedan kanal nema jest da će poruka gotovo 100% biti dostavljena, za razliku od ostalih aplikacija kojima je potrebna veza s internetom. Uostalom, za takve aplikacije je potrebno preuzeti aplikaciju kako bi mobitel uopće mogao primiti poruku. Ne bi li sustav i prijenos podataka bili što sigurniji, pristup podacima može imati samo onaj korisnik koji ima pohranjen broj telefona koji se koristi u SIM800L modulu. Također, sustav će poslati podatke o sustavu samo ako je na zadani broj poslana ispravna ključna riječ. SIM800L je modul serijske komunikacije, što znači da stalno razmjenjuje nekakve podatke s mikrokontrolerom kojim je upravljana. U trenutku kada je na mobilni broj SIM kartice poslana poruka s ispravnom ključnom riječi, modul javlja mikrokontroleru da mu je pristigla poruka. Sustav je podešen tako da je SIM modul u GPRS načinu rada, odnosno šalje podatke u bazu na internet sve dok ne primi zahtjev od korisnika u obliku SMS poruke. Tada modul prekida GPRS vezu i pokreće novu GSM vezu. Kada korisnik pošalje poruku na ispravan broj, modul šalje znak mikrokontroleru da je poruka stigla, a sljedeća naredba je zatim vidljiva u serijskom monitoru, koji prikazuje koji se podaci šalju, a koji primaju preko mikrokontrolera.

```
>> +CMTI: "SM",1
```

CMTI naredba obavještava sustav da je primljena SMS poruka i da je identifikator poruke u ovom slučaju 1. Određene SIM kartice imaju ograničen prostor za pohranu poruka. Za većinu kartica koje podržavaju 2G, minimalan broj poruka koje se mogu pohraniti je 10. Stoga je nužno brisati primljene poruke nakon što je analiza sadržaja gotova. To se postiže sljedećom naredbom:

```
Serial3.print("AT+CMGD=1");
```

Ideja je automatizirati proces primanja poruke, promijeniti vezu iz GPRS u GSM i obratno, te poduzeti željene akcije za slanje podataka. Kako bi se pročitao sadržaj poruke, koristi se

naredba AT+CMGR, a kao odgovor na tu naredbu modul šalje podatke o poruci uključujući sadržaj poruke, broj pošiljatelja i vrijeme primitka:

```
Serial3.print("AT+CMGR=1");
>>+CMGR: "REC UNREAD","+38597123456","",11/06/20,15:30:42"
Test poruka
```

Slanje povratne SMS poruke moguće je jedino ako je poslana prava ključna riječ, zato je potrebno filtrirati sadržaj CMGR naredbe kako bi se taj sadržaj pohranio u varijablu i napravila usporedba s ključnom riječi. To je moguće postići definiranjem pomoćne varijable, koja će čitati ono što je SIM800L poslao prema Arduino. U slučaju CMGR naredbe, filtriranje je moguće napraviti na temelju toga što su sve informacije odvojene zarezom. Tako se može definirati da se pomoćna varijabla popunjava znakovima sve dok se ne pojavi znak zarez, a zatim se analizira niz do tad primljenih znakova.

S programske strane, ako dođe do dolazne SMS poruke, potrebno je napraviti nekoliko radnji kako bi se korisniku poslao povratni SMS s podacima senzora. Te radnje su slijedne te ih je najlakše opisati sljedećim stanjima u mikrokontroleru:

```
enum _parseState {
    STANJE_DETEKCIJA_TIP_PORUKE,

    STANJE_AT_ZANEMARI_KOMANDA,

    STANJE_CMTI_STORE_TIP,
    STANJE_CMTI_CITANJE_ID,

    STANJE_CMGR_CITANJE_STATUS,
    STANJE_CMGR_CITANJE_MSISDN,
    STANJE_CMGR_CITANJE_IGNORE,
    STANJE_CMGR_CITANJE_DATUM,
    STANJE_CMGR_CITANJE_SADRZAJ,

    STANJE_CUSD_CITANJE_STATUS,
    STANJE_CUSD_CITANJE_SADRZAJ,
    STANJE_CUSD_CITANJE_IGNORE
};
```

Kod 3.4 Stanja potrebna za slanje SMS poruke

Kada korisnik pošalje SMS, sustav se nalazi u početnom stanju pod nazivom STANJE_DETEKCIJA_TIP_PORUKE, gdje se utvrđuje tip naredbe koju SIM800L šalje

prema mikrokontroleru. Tada se čitaju prva 3 znaka, kako bi se ustanovilo je li poslana AT+ naredba, ako je, radi se o naredbi koju će sustav ignorirati i pomoćna varijabla će se isprazniti i ponovno popunjavati. U slučaju kada je primljena poruka, šalje se naredba +CMTI koju sustav prepoznaje čitajući znak po znak. Mikrokontroler prelazi u stanje STANJE_CMTI_STORE_TIP pa zatim u STANJE_CMTI_CITANJE_ID, gdje se utvrđuje primljeni tip poruke i identifikacijska oznaka poruke kako bi se oznaka s tom oznakom kasnije mogla pročitati i usporediti s ključnom riječi. Nakon što se odredio redni broj poruke, mikrokontroler poziva CMGR naredbu i postavlja mikrokontroler u početno stanje. Tada se ponavlja sličan ciklus, sustav prepoznaje naredbu i prelazi u stanje STANJE_CMGR_CITANJE_STATUS. U kodu je vidljivo da je zadnje stanje CMGR faze čitanje sadržaja poruke te se u toj fazi konačno pohranjuje sadržaj poruke u određenu varijablu. Varijabla se šalje u funkciju koja provjerava podudara li se tekst poruke s ključnom riječi:

```
void parseSMS_sadrzaj() {
    char* ptr = buffer;
    while ( strlen(ptr) >= 2 ) {
        if ( ptr[0] == 'r' ) {
            if ( ptr[1] == '1' ){
                Serial3.print("AT+CUSD=1,\"*101#\r");
            }
            else {
                resetBuffer();
            }
        }
    }
}
```

Kod 3.5 Provjera teksta dolazne poruke i ključne riječi

Iz programskog koda se može vidjeti da je definirana riječ „r1“ te se u tom slučaju poziva CUSD naredba. Ova naredba je korištena u radu kako bi korisnik, uz sve podatke sa senzora i ostalih modula, dobio informaciju o stanju računa SIM kartice umetnute u SIM800L modul. To je još jedna korisna informacija ako korisnik koristi pretplatu s unaprijed plaćenom uslugom, tzv. *prepaid* model. Tada sustav opet prelazi u početno stanje te zatim u 3 navedena CUSD stanja definirana u kodu Kod 3.4. Informacija o sadržaju poruke se zatim sprema u dodatnu pomoćnu varijablu te se u zadnjem koraku procesa definira SMS poruka koja će biti poslana korisniku:

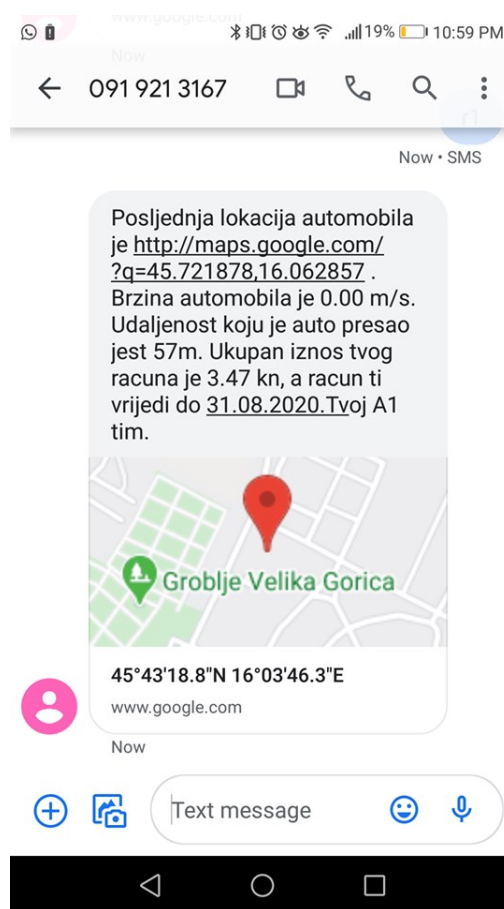
```

Serial3.print("Posljednja lokacija automobila je ");
Serial3.print("http://maps.google.com/?q=");
Serial3.print(testStringGPSlat);
Serial3.print(",");
Serial3.print(testStringGPSlng);
Serial3.print(" . Brzina automobila je ");
Serial3.print(brzinaPrednjegKotaca);
Serial3.print(" m/s. Udaljenost koju je auto presao
jest ");
Serial3.print(udaljenostPrednjegKotaca);
Serial3.print("m.");
Serial3.print(StanjeRacuna);
Serial3.print(".");
Serial3.println((char)26);

```

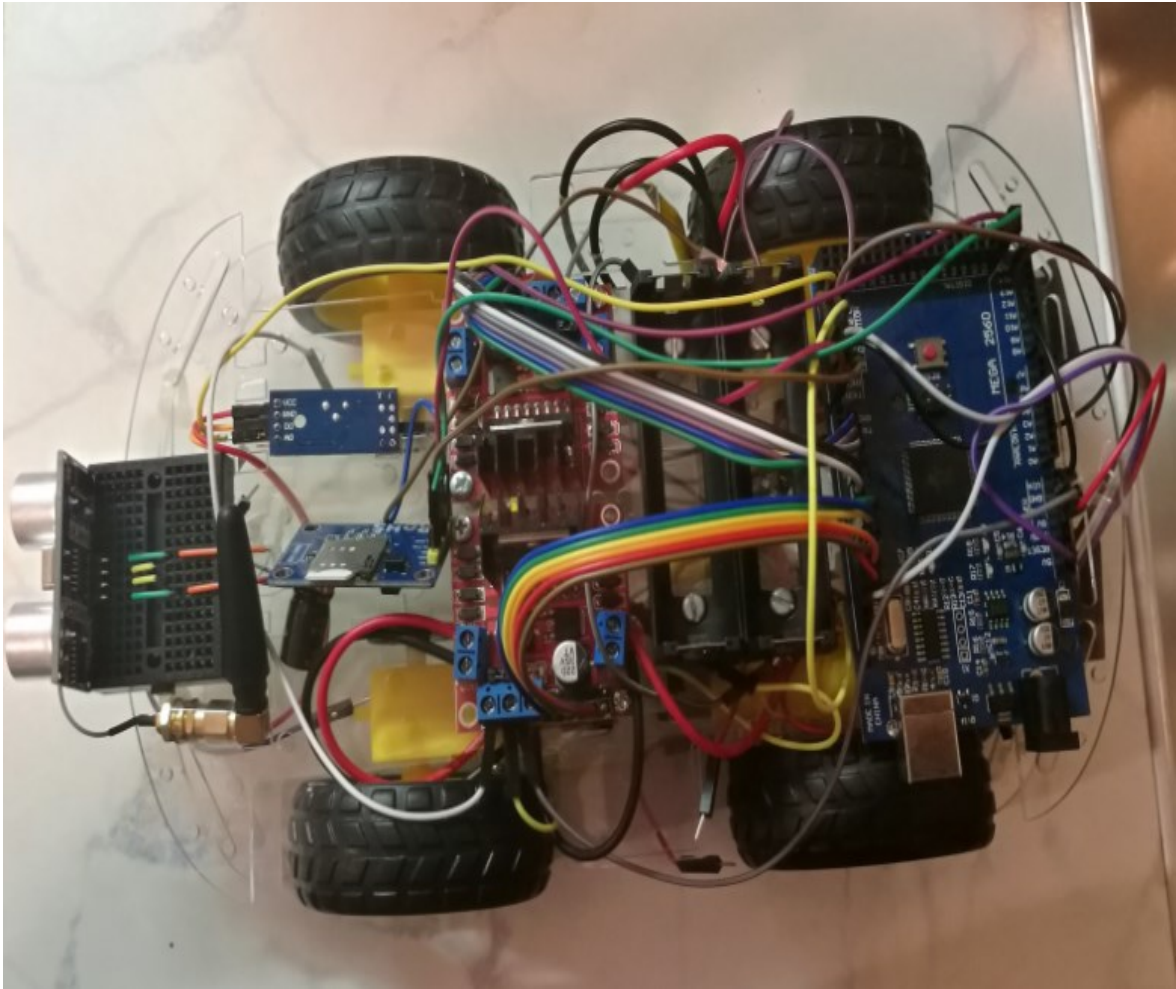
Kod 3.6 Struktura SMS poruke koja se šalje korisniku

Nakon izvršenog koda, sustav se ponovno vraća u GPRS način rada kao što je slikom Kod 4.1 u idućem poglavlju. Na sljedećoj slici je prikazana SMS poruka s navedenim podacima:



Slika 3.2 SMS poruka primljena na zahtjev korisnika

Ova mobilna aplikacija za razmjenu poruka automatski prikazuje pregled poveznice koja usmjerava korisnika na Google Maps. Klikom na poveznicu otvara se aplikacija Google Maps, koja je ugrađena u svaki noviji mobilni telefon. Na mapi će biti prikazan marker s posljednjom lokacijom robota, a za dodatne informacije i prijašnje podatke o sensorima i lokaciji robota korisnik ima opciju otvoriti Google Spreadsheet bazu putem internet preglednika.



Slika 3.3 Prikaz prototipa sustava

4. Slanje podataka na Internet

Kao i kod većine sustava koji implementiraju karakteristike IoT-a, ideja je raspolagati podacima i na temelju njih doprinijeti automatizaciji procesa, koja pojednostavljuje svakodnevni život korisnika. Skladištenje podataka se postiže korištenjem SIM800L EVB modula. Modul ima mogućnost GPRS komunikacije, kojom se informacije od senzora šalju izravno na internet u Google Spreadsheet dokument. Osim GPRS veze, modul može ostvariti i GSM vezu, koja omogućava slanje SMS poruka. Takav način rada je također objašnjen u ovom radu. SIM800L modul, kao i HC-06 bluetooth modul, koristi serijsku komunikaciju kao način razmjene informacije s Arduinoom. Programiranje modula je mnogo kompleksnije u odnosu na bluetooth modul, jer zahtjeva korištenje niza naredbi kako bi se ostvarila uspješna GPRS komunikacija. Sljedeći set naredbi je nužan za ostvarenje GPRS komunikacije:

```
Serial3.print("AT+SAPBR=3,1,\"Contype\",\"GPRS\"\r");
Serial3.print("AT+SAPBR=3,1,\"APN\",\"gprs0.vipnet.hr\"\r");
Serial3.print("AT+SAPBR=0,1\r");
Serial3.print("AT+SAPBR=1,1\r");
Serial3.print("AT+SAPBR=2,1\r");
Serial3.print("AT+HTTPINIT\r");
Serial3.print("AT+HTTPPARA=\"CID\",1\r");
Serial3.print("AT+HTTPPARA=\"URL\",\"http://api.pushingbox.com/pushingbox?devid=v03F8A1472B771AC&latlng=37.91\r");
Serial3.println("AT+HTTPACTION=0\r");
```

Kod 4.1 Postavljanje SIM800L modula u GPRS način rada

Kako bi se modulu omogućilo korištenje GPRS veze, prvo je potrebno podesiti Bearer Profile, odnosno vrstu autentikacije SIM kartice prema mreži operatera kojoj pripada. To radi naredba AT+SAPBR. Nizom SAPBR naredbi se prekida postojeća GPRS veza te postavlja nova sa zadanim postavkama operatera. Za korištenje ovog modula smije se koristiti isključivo SIM kartica čija mreža podržava 2G mrežnu vezu. Stoga je za ovaj sustav korištena kartica A1, tj. nekadašnje VIP mreže, pa je sukladno s time definiran prikladni naziv pristupne točke (eng. *Access point name*). Naredbom AT+HTTPINIT se inicijalizira zahtjev prema HTTP (eng. *HyperText Transfer Protocol*) protokolu kojeg podržava neka web lokacija. Zahtjev prema HTTP lokaciji nastavlja se uporabom AT+HTTPPARA komande, kojom se podešavaju parametri za interakciju s web lokacijom. S obzirom na to

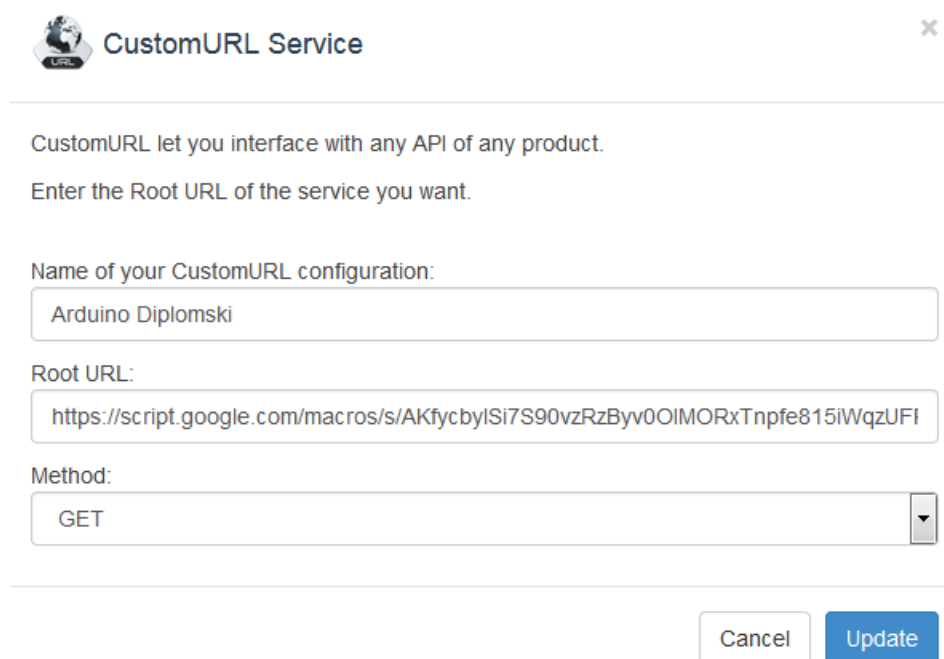
da je ovaj modul godinama na tržištu, ugrađeni program koji sadrži, dozvoljava isključivo komunikaciju s HTTP web lokacijama. Od 2020., gotovo sve internet lokacije moraju zadovoljiti HTTPS standarde, koje imaju dodatan sloj sigurnosti u odnosu na „nesigurni“ HTTP protokol. Kako ovo predstavlja ograničenje, potrebno je naći servis koji pretvara HTTP zahtjeve u HTTPS i tako šalje podatke na sigurnu lokaciju. Za tu svrhu koristi se PushingBox, čija je uloga opisana u idućem potpoglavlju. Konačno, naredbom AT+HTTPACTION se izvršava GPRS veza i šalju potrebni parametri. Navedene naredbe treba implementirati u logiku mikrokontrolera i uzeti u obzir kašnjenja pri obradi svake od naredbe. Bitno je testirati trajanje svake naredbe i ustanoviti kojoj treba najviše vremena da se izvede. Ako se dvije uzastopne naredbe prebrzo izvedu, druga naredba će se zaista realizirati, a prva zanemariti, što će dovesti do neispravnog rada elementa, tj. podatci se neće uspješno poslati na internet. Arduino Mega 2560 ne može obrađivati više od jedne naredbe istovremeno pa je nužno izbjegavati često korištenu funkciju `delay()` jer ona unosi kašnjenje u radu cijelog sustava. Kada se pozove navedena funkcija, sve ostale naredbe su „na čekanju“. Umjesto `delay()` funkcije, koristi se funkcija `millis()`, koja vraća parametar u obliku ukupnog vremena koje je prošlo od trenutka paljenja sustava. Tako se postiže da se naredbe neprekidno izvršavaju bez kašnjenja te da se za specifične naredbe provjeri je li prošlo određeno vrijeme za njihovo obavljanje. Ako nije prošlo zadano vrijeme, uvjet se preskače i izvodi se sljedeća linija koda. Primjer ovakve implementacije je vidljiv u sljedećem isječku koda:

```
if(millis() - previousMillisSIM800_6 >= intervalSIM800){
  previousMillisSIM800_7 = previousMillisSIM800_6 +
  ATcommandDelay;
  Serial3.print("AT+HTTTPARA=\"CID\",1\r");
```

U ovom je primjeru potrebno osigurati vrijeme u iznosu `ATcommandDelay` kako bi se naredba `AT+HTTTPARA=\"CID\",1\r` uspješno izvela, odnosno da SIM800L modul vrati odgovor prema Arduino od prošle odrađene AT naredbe. Iako se čini da mikrokontroler paralelno obavlja više zadataka, to se ne događa, no zahvaljujući visokoj frekvenciji rada Arduina od 16 MHz, moguće je postići prividno obavljanje nekoliko tisuća naredbi u sekundi.

4.1. PushingBox servis

PushingBox je usluga u oblaku (eng. cloud service), koja izvršava radnje temeljene na API (eng. *Application program interface*) zahtjevu kojeg primi [14]. Sve što je potrebno za njeno besplatno korištenje jest imati izrađen Google email račun. Pomoću njega se moguće prijaviti u platformu i izraditi svoj API scenarij. Za prikaz podataka u radu koristi se Google Spreadsheet, a korištenje PushingBox-a omogućava prijenos zahtjeva s HTTP domene, na kojoj se ovaj servis nalazi, u HTTPS domenu koju koristi Google Spreadsheet platforma. Primjer definiranja zadatka u PushingBox-u je prikazan na donjoj slici:



CustomURL Service

CustomURL let you interface with any API of any product.

Enter the Root URL of the service you want.

Name of your CustomURL configuration:

Arduino Diplomski

Root URL:

https://script.google.com/macros/s/AKfycbyISi7S90vzRzByv0OIMORxTnpfe815iWqzUFI

Method:

GET

Cancel Update

Slika 4.1 Izrada scenarija u PushingBox platformi

U trenutku kada SIM800L modul pozove poveznicu koja sadrži PushingBox domenu, izvršava se scenarij na slici. Naime, GET metodom se poziva skripta definirana unutar Google Spreadsheet dokumenta. Poveznica prikazana na slici sadrži identifikacijsku oznaku Spreadsheet-a te se tako pristupa točno definiranom dokumentu. Kada SIM modul obavi AT naredbu koja poziva poveznicu na PushingBox servisu, pokreće se skripta unutar Googleove platforme koja prihvaća određene parametre poput brzine vozila, geografske širine i duljine. Ti podatci se zatim raspoređuju u predefinirane stupce i redove unutar dokumenta. Identifikacijsku oznaku također sadrži i scenarij na slici, koji je potrebno definirati u PushingBox platformi na sljedeći način:

Test Arduino Sheet Rename

Deactivate

Test Scenario

DeviceID: v03f8a1472b771ac



Arduino Diplomi **Data:** ?latlng=\$latlng\$&brzina=\$brzina\$&predUdaljenost=\$predUdaljenost\$&preprekaUdaljenost=\$preprekaUdaljenost\$ Edit Delete

Back

Add an Action

Slika 4.2 Definiranje identifikacijske oznake i parametara unutar poveznice zahtjeva

Na slici su prikazani svi parametri koji su dio poveznice te se oni prenose izravno u Google dokument. Scenarij slanja podataka je moguće testirati preko testnog okruženja u PushingBox platformi bez korištenja SIM modula.

4.2. Pohrana podataka u Google Spreadsheet

Svi podatci o senzorima se pohranjuju u formi Google dokumenta u obliku tablice. U prethodnom poglavlju je objašnjeno što je sve potrebno da se podatci prenesu sa sustava mobilnog robota do dokumenta na internetu koji se nalazi na HTTPS lokaciji. Zahvaljujući korištenjem Googleove platforme, upis i brisanje podataka se može automatski izvoditi. Kako bi se to omogućilo, mora se definirati programska skripta koja će se izvršiti svaki put kada ju pozove neki vanjski servis poput PushingBox-a. Pushingbox služi kao prijenosnik informacije, a te informacije je potrebno prepoznati i analizirati u dokumentu. To se definira u uređivaču skripte koji je programiran u programskom jeziku JavaScript i prikazan idućim programskim kodom:

```

for (var informacija in e.parameter) {
    Logger.log('In for loop, informacija='+informacija);
    var value = stripQuotes(e.parameter[informacija]);
    switch (informacija) {
        case 'latlng':
            rowData[1] = value;
            break;
        case 'brzina':
            rowData[2] = value;
            break;
        case 'predUdaljenost':
            rowData[3] = value;
            break;
        case 'preprekaUdaljenost':
            rowData[4] = value;
            break;
        default:
            result = "Nepostojeca informacija";
    }
}

```

Kod 4.2 Analiza parametara u Google Spreadsheet skripti

U kodu se može vidjeti da se prenose 4 parametra sa sustava mobilnog robota u dokument. Na slici Slika 4.3 navedeni su identični parametri u istom redosljedu. To je bitno jer Google skripta prima parametre u unaprijed definiranom rasporedu. Skripta također filtrira parametre jednog po jednog i zapisuje ih u uzastopne stupce u tablici kako bi informacije bile pregledne korisniku. Kod je vrlo lako modificirati u slučaju da se treba dodati ili ukloniti određeni parametar. Od značajnih podataka, koji daju senzori i ostali korišteni moduli u projektu, koriste se sljedeće informacije za ispis u dokumentu: geografska širina i duljina, brzina motora robota, ukupna pređena udaljenost robota i udaljenost robota od prepreke. Sve vrijednosti su trenutne, odnosno ne računaju se srednje vrijednosti više podataka kako bi se ispisala konačna vrijednost u dokumentu. Ono što omogućava predefiniran redosljed podataka jest automatsko generiranje grafova na temelju dobivenih podataka. Tako se korisniku najjednostavnije predočavaju korisne informacije te se odmah primjećuju određene anomalije ako dođe do njih. Uz podatke sa senzora, u dokumentu je definirano vrijeme svakog upisa podatka i to vrijeme je vidljivo u dokumentu u prvom stupcu. Na sljedećoj slici je prikaz podataka unutar Google Spreadsheet dokumenta.

Diplomski rad - Luka Žugaj ☆ Spremljeno na Disk

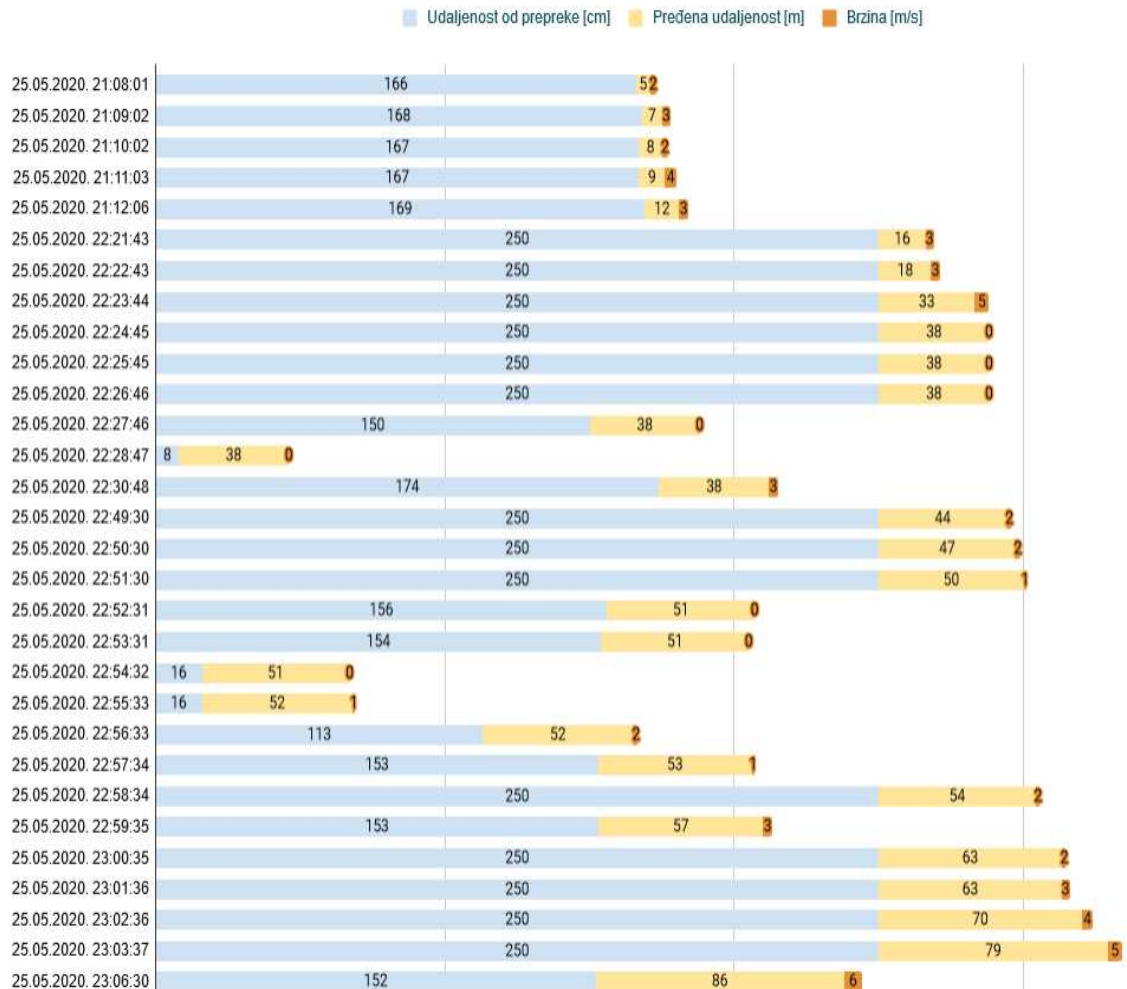
Datoteka Uredi Prikaži Umetni Oblik Podaci Alati Programski dodaci Pomoć Dokument je zadnji pu

100% Kn % .0 .00 123 Zadano (Ar... 10 B I S A

	A	B	C	D	E
1660	Vrijeme	GPS lokacija	Brzina [m/s]	Pređena udaljenost [m]	Udaljenost od prepreke [cm]
1661	25.05.2020. 21:08:01	45.721806,16.063437	2	5	166
1662	25.05.2020. 21:09:02	45.721828,16.063320	3	7	168
1663	25.05.2020. 21:10:02	45.721809,16.063364	2	8	167
1664	25.05.2020. 21:11:03	45.721798,16.063175	4	9	167
1665	25.05.2020. 21:12:06	45.721828,16.063206	3	12	169
1666	25.05.2020. 22:21:43	,	3	16	250
1667	25.05.2020. 22:22:43	,	3	18	250
1668	25.05.2020. 22:23:44	45.721798,16.063208	5	33	250
1669	25.05.2020. 22:24:45	45.721775,16.063288	0	38	250
1670	25.05.2020. 22:25:45	45.721809,16.063183	0	38	250
1671	25.05.2020. 22:26:46	45.721840,16.063093	0	38	250
1672	25.05.2020. 22:27:46	45.721832,16.063084	0	38	150
1673	25.05.2020. 22:28:47	45.721855,16.063007	0	38	8
1674	25.05.2020. 22:30:48	45.721825,16.063068	3	38	174
1675	25.05.2020. 22:49:30	45.721916,16.062675	2	44	250
1676	25.05.2020. 22:50:30	45.721867,16.062918	2	47	250
1677	25.05.2020. 22:51:30	45.721939,16.062609	1	50	250
1678	25.05.2020. 22:52:31	45.721939,16.062609	0	51	156
1679	25.05.2020. 22:53:31	45.721939,16.062609	0	51	154
1680	25.05.2020. 22:54:32	45.721939,16.062609	0	51	16
1681	25.05.2020. 22:55:33	45.721939,16.062609	1	52	16
1682	25.05.2020. 22:56:33	45.721939,16.062609	2	52	113
1683	25.05.2020. 22:57:34	45.721939,16.062609	1	53	153
1684	25.05.2020. 22:58:34	45.721981,16.062384	2	54	250
1685	25.05.2020. 22:59:35	45.721844,16.063053	3	57	153
1686	25.05.2020. 23:00:35	45.721813,16.063213	2	63	250
1687	25.05.2020. 23:01:36	45.721855,16.063011	3	63	250
1688	25.05.2020. 23:02:36	45.721821,16.063171	4	70	250
1689	25.05.2020. 23:03:37	45.721832,16.063057	5	79	250
1690	25.05.2020. 23:06:30	45.721901,16.062687	6	86	152
1691					

Slika 4.3 Prikaz podataka unutar Google Spreadsheets dokumenta

Na gornjoj slici je primjer ispisa svih podataka koji su uzeti u obzir. Moguće je da dođe prekida GPS veze u određenom trenutku jer je veza veoma osjetljiva pa se u tom slučaju ne ispisuje vrijednost GPS lokacije. Za brzinu robota se koristi trenutna brzina očitana u vremenu definiranom na slici. Osim pisanih podataka, moguće je automatsko generiranje grafikona, koji se osvježavaju nakon svakog novog upisa u dokument. Kod velikih količina podataka, kao što je primjer u ovom projektu, preporučljivo je koristiti slikovne metode predočenja podataka jer je dokazano da ljudi brže uočavaju informacije u obliku slike nego teksta. Primjer grafikona je predstavljen na idućoj slici, a korisnik dokumenta može promijeniti oblik prikaza u bilo kojem trenutku.



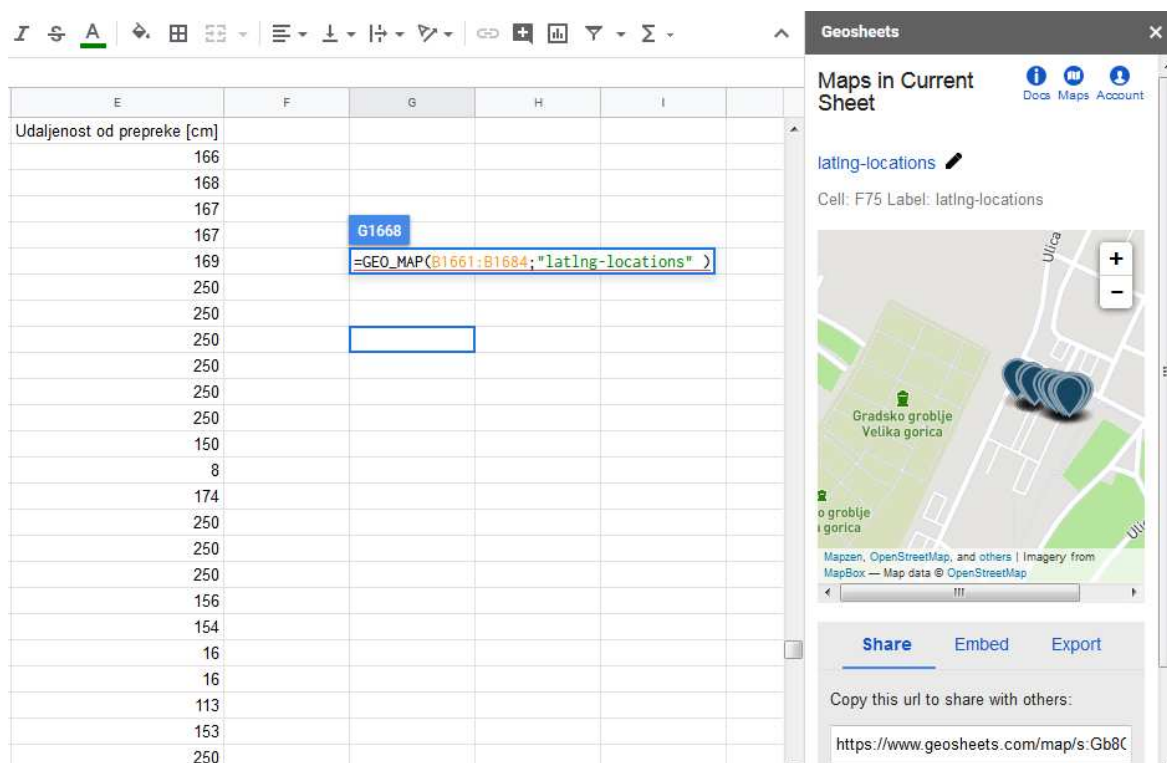
Slika 4.4 Prikaz podataka sustava u obliku grafikona

Iako je veliki broj podataka uključen u analizu, krajnji korisnik može u par klikova promijeniti raspon podataka i tako uključiti vremensko razdoblje koje mu odgovara. Također, očitavanja i slanje obavijesti putem interneta se u ovom slučaju šalju svake minute zbog lakšeg testiranja i prikaza, ali se za komercijalne potrebe to vrijeme može povećati.

Jedan od vrlo korisnih podataka je GPS lokacija nekog sustava u pokretu. Ono što daje dodatnu vrijednost ove informacije je njen prikaz na karti. Osim navedenih prikaza geografske širine i duljine, u ovom radu se ti podatci izravno ucrtavaju na kartu pomoću dodatka Geosheets. Ovaj dodatak je moguće dodati u svaki Google Spreadsheet dokument. Prilikom upisa informacije o lokaciji, interaktivna karta označava lokaciju na temelju primljene geografske širine i duljine koje se nalaze u određenom stupcu. Geosheets koristi nekoliko funkcija koje se aktiviraju pri upisu novog podatka u ćeliju dokumenta. Sve što je potrebno jest definirati sljedeću funkciju u jednu od slobodnih ćelija u dokumentu:

```
=GEO_MAP(B1650:B1700;"latlng-locations")
```

Nakon što se definira raspon podataka, prikazuje se interaktivna mapa na desnoj strani dokumenta:



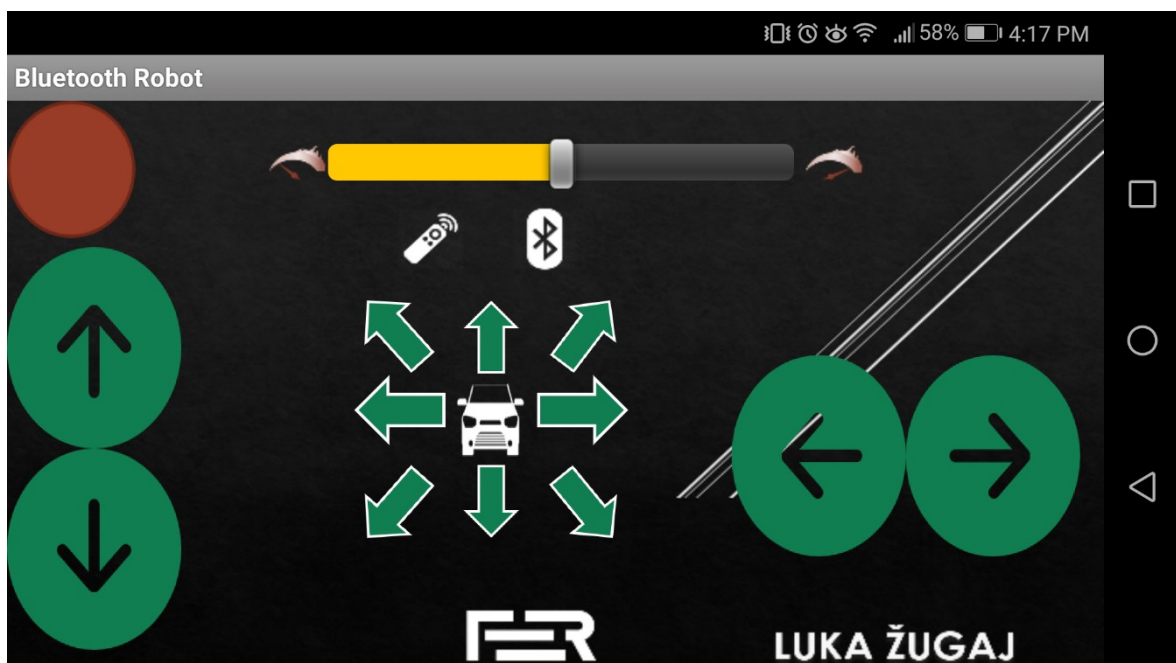
Slika 4.5 Prikaz GPS lokacije u dokumentu

Ovaj dodatak nudi nove mogućnosti, npr. ako su očitane lokacije blizu jedna druge, moguće je da se više lokacija grupira u jednu i da se navede broj ukupnih približnih lokacija. Tako se lako može uočiti da se robot zadržao dulje vrijeme na jednom mjestu. Podatke o lokaciji je moguće prenijeti u neki drugi sustav, koji na temelju vremena upisa podataka može povezati svaku lokaciju i na temelju toga stvoriti prikaz puta robota. Sa slike je vidljivo da GPS modul ne daje uvijek istu lokaciju, što je uzrokovano manjkom signala u određenom trenutku očitavanja. Unatoč tome, greška je vrlo mala, u okviru 30 metara, što je vrlo prihvatljivo za modul u cjenovnom opsegu do 50 kuna.

5. Izrada Android aplikacije za upravljanje robotom

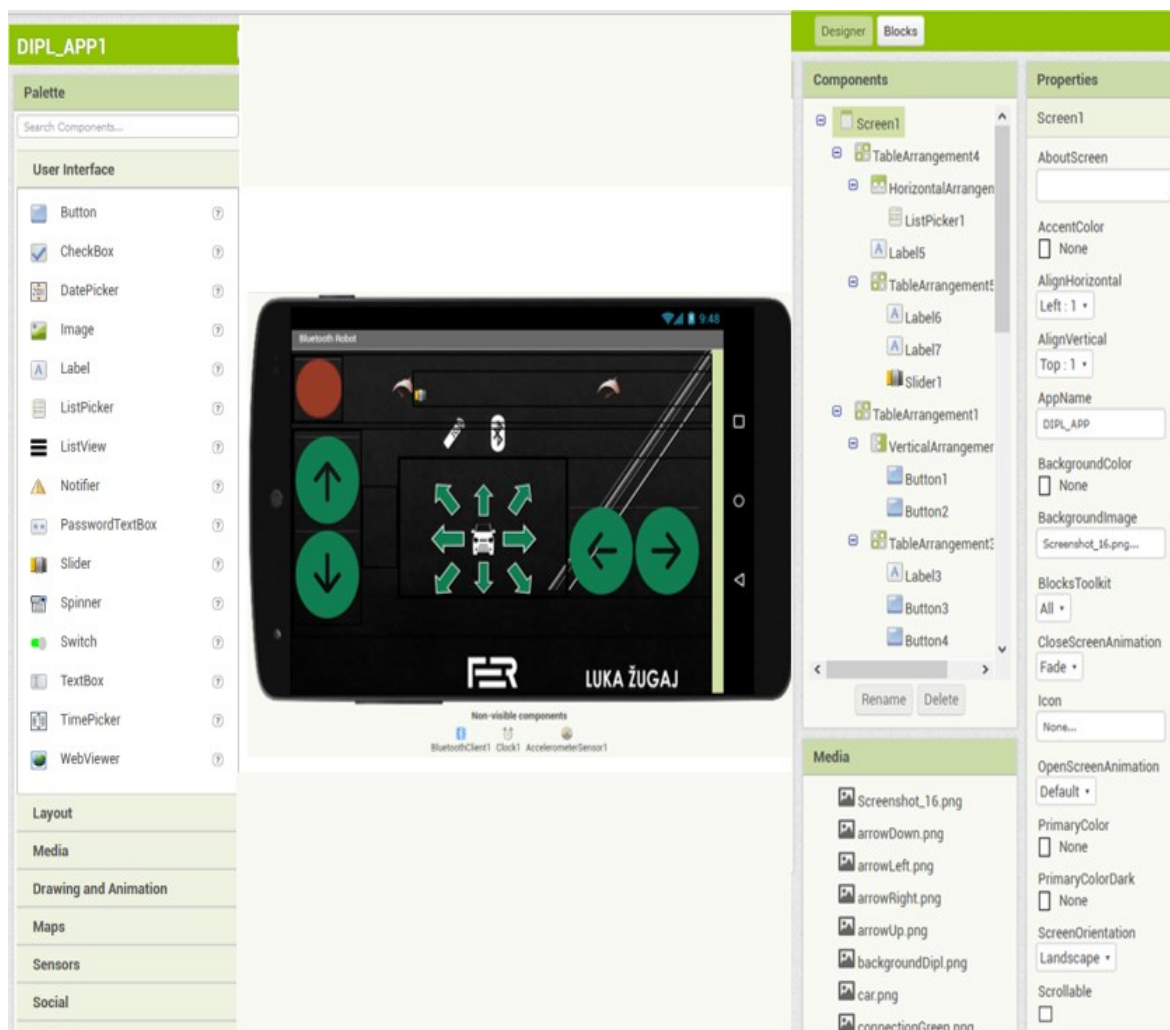
Programska platforma kojom je napravljena aplikacija za upravljanje mobilnog robota je App Inventor, a temelji se na programskom jeziku Blockly, koji je detaljnije opisan u poglavlju 1.2. Naredbe se šalju Bluetooth vezom pritiskom na gumbe unutar aplikacije, koji zatim šalju informaciju bluetooth modulu povezanom na Arduino mikrokontroler. Takav način komunikacije je nužno podesiti s aplikacijske strane, ali i sa strane hardvera, odnosno mikrokontrolera. Način na koji je podešena hardverska strana objašnjen je u poglavlju 3.2.1, gdje je definiran rad motora kada korisnik pritisne određenu opciju u aplikaciji.

App Inventor se sastoji od dva dijela. Prvi dio je orijentiran na dizajn i izgled aplikacije, definiciju gumba, pozadine, slika unutar aplikacija i slično. Drugi dio opisuje programsku logiku potrebnu za funkcioniranje aplikacije, tj. opisuje ulogu svakog gumba i slike u trenutku pritiskivanja, kao i logiku povezivanja aplikacije preko bluetooth veze [15]. Ova platforma nudi opciju da se funkcionalnost i dizajn aplikacije naprave u programskom jeziku JavaScript te da se zatim taj kod učita. Tako je olakšan proces izmjene aplikacije bez potrebe za definiranjem dodatnih blokova, već je moguće dodati nekoliko linija koda da bi se ostvario isti efekt. Na sljedećoj slici je prikazan konačan izgled aplikacije:



Slika 5.1 Izgled aplikacije

Za izradu nove aplikacije, potrebno je napraviti novi projekt te definirati ime projekta. Nakon toga se otvara početna stranica na kojoj je moguće definirati izgled aplikacije. Prikaz početne stranice vidljiv je na idućoj slici:



Slika 5.2 Sučelje platforme App Inventor

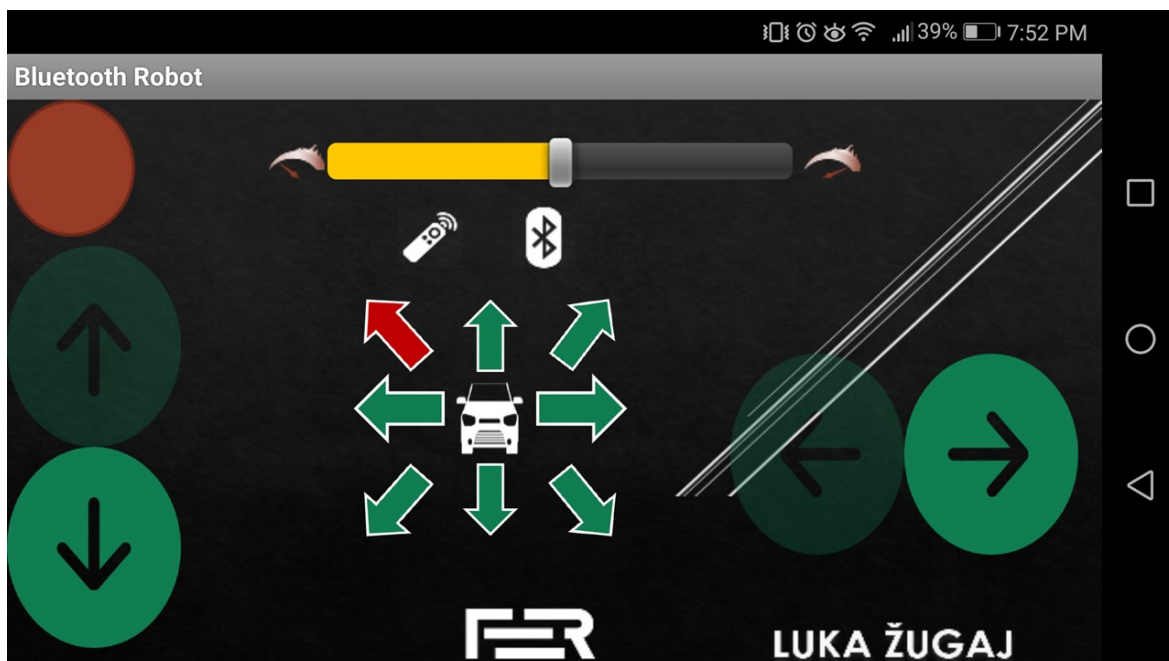
S lijeve strane slike prikazan je izbornik u kojem su definirani različiti elementi, koji se mogu koristiti u izradi aplikacije. U ovoj aplikaciji se koriste neki od tih elemenata, a prikaz korištenih elementa nalazi se s desne strane. Za navigaciju robota koriste se gumbi, tj. *Button* element, a za podešavanje brzine motora upotrebljava se klizač, tj. *Slider*. *Slider* se kontrolira povlačenjem prsta po ekranu. Inicijalna vrijednost brzine motora podešena je na vrijednost od 150 okreta u minuti, što je približno srednjoj vrijednosti koju motor može postići. Zbog tog razloga, klizač je podešen na sredini skale. Za jednostavnije upravljanje robotom, aplikacija koristi horizontalnu orijentaciju koju je moguće podesiti u postavkama u krajnjem desnom dijelu slike. Slike i ostale medije koji se koriste u dizajnu aplikacije treba učitati na sustav i prikladno ih imenovati. Primjerice, gumb je moguće definirati na dva načina,

pisanjem programskog koda ili izravnim podešavanjem u postavkama prikazanim na prethodnoj slici. Gumb koji označava kretanje prema lijevo označen je s oznakom *Button3* i vidi se da pripada podskupu *TableArrangement3*, koji pripada podskupu *TableArrangement1*. Programskim kodom je to definirano ovako:

```
TableArrangement3 = new TableArrangement(TableArrangement1);
TableArrangement3.Column(3);
TableArrangement3.Width(220);
TableArrangement3.Row(0);
Button3 = new Button(TableArrangement3);
Button3.BackgroundColor(0x00FFFFFF);
Button3.Column(0);
Button3.Height(110);
Button3.Width(95);
Button3.Image("arrowLeft.png");
Button3.Row(1);
```

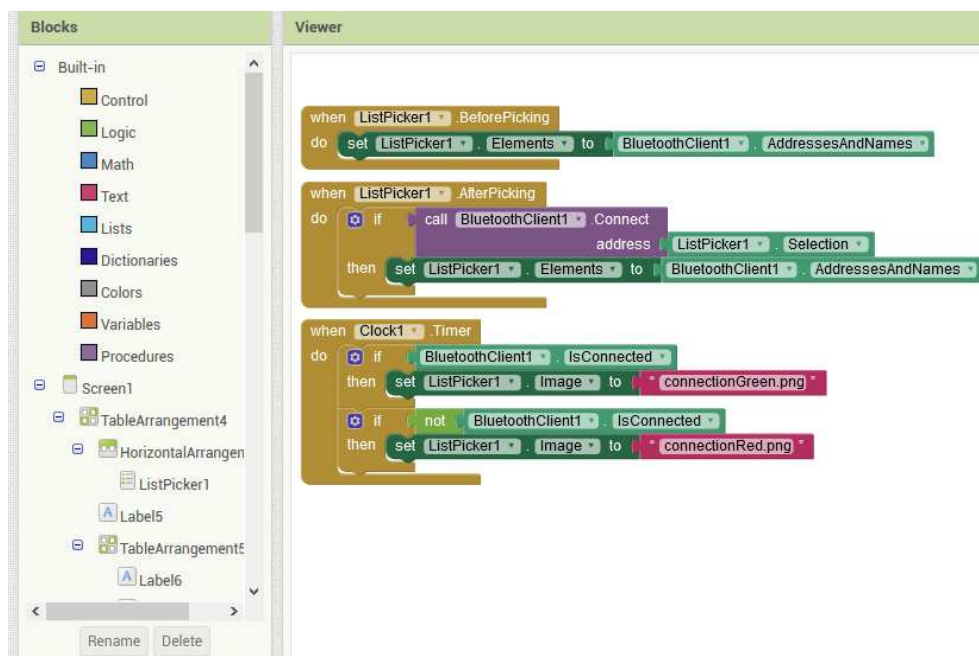
Kod 5.1 Definicija strukture gumba u programskom jeziku JavaScript

Osim navedene strukture, potrebno je uključiti prikladne biblioteke potrebne za ispravan rad korištenih funkcija. Podešavanjem ovih parametara lako se može izmijeniti pozicija, širina i visina nekog elementa. Na isti način se definiraju svi ostali elementi. Za upravljanje sustavom koriste se 4 gumba, klizač za podešavanje brzine motora i gumb za povezivanje bluetooth modula s mobilnom aplikacijom. Ostali elementi su korišteni zbog ljepšeg izgleda aplikacije. Osim interaktivnih elemenata aplikacije poput gumbova i klizača, u aplikaciji je moguće koristiti senzore koji su već ugrađeni u mobilnom telefonu. Tako se za upravljanje koristi bluetooth, no mogu se iskoristiti i ostali senzori mobitela poput akcelerometra. Primjena tog senzora navedena je kao moguća nadogradnja i opisana je u idućem poglavlju. Kako bi svi ti elementi dobili svoju funkciju, nužno je razviti programsku logiku koja će povezati mobilnu aplikaciju sa mikrokontrolerom. To se postiže u *Blocks* sučelju App Inventor platforme. Za ovu aplikaciju je bitno definirati bluetooth vezu te ponašanje sustava kada je bilo koji od gumba pritisnut. Nadalje, za upravljanje robotom se koriste 8 smjerova kretanja. S obzirom na to da su definirana 4 gumba, ostalih 4 smjera se ostvaruju kada korisnik pritisne na dva gumba istovremeno. Primjerice, za kretanje prema naprijed lijevo, tj. ukoso, nužno je istovremeno pritisnuti gumb prema gore i prema lijevo. Na slici Slika 5.1 je prikazan izgled aplikacije i svih 8 smjerova kretanja. Kada korisnik pritisne na neku tipku, odgovarajući smjer kretanja će biti naznačen crvenom bojom. Ponašanje je moguće vidjeti na sljedećem primjeru:



Slika 5.3 Ponašanje aplikacije kod pritiska dva gumba istovremeno

App Inventor se temelji na programskom jeziku JavaScript, ali je napravljen tako da korisnik platforme svoj kod piše „slaganjem blokova“. Ova značajka pospješuje korisničko iskustvo i daje korisniku slobodu da poveže elemente u kompleksnu cjelinu bez velikog znanja u području programiranja. Za postizanje bluetooth veze, napravljen je blok:



Slika 5.4 Definiranje bluetooth veze

Osim korištenja blokova, ista funkcionalnost se postiže sljedećim kodom, koji zatim može biti umetnut u platformu kako bi se prikazali blokovi kao na prethodnoj slici.

```

public void ListPicker1BeforePicking() {
    ListPicker1.Elements(YailList.makeList(BluetoothClient1
        .AddressesAndNames()));
}
public void ListPicker1AfterPicking() {
    if(BluetoothClient1.Connect(ListPicker1.Selection())) {
        ListPicker1.Elements(YailList.makeList(BluetoothClient1
            .AddressesAndNames()));
    }
}
public void Clock1Timer() {
    if(!BluetoothClient1.IsConnected()) {
        ListPicker1.Image("connectionRed.png");
    }
    if(BluetoothClient1.IsConnected()) {
        ListPicker1.Image("connectionGreen.png");
    }
}
}

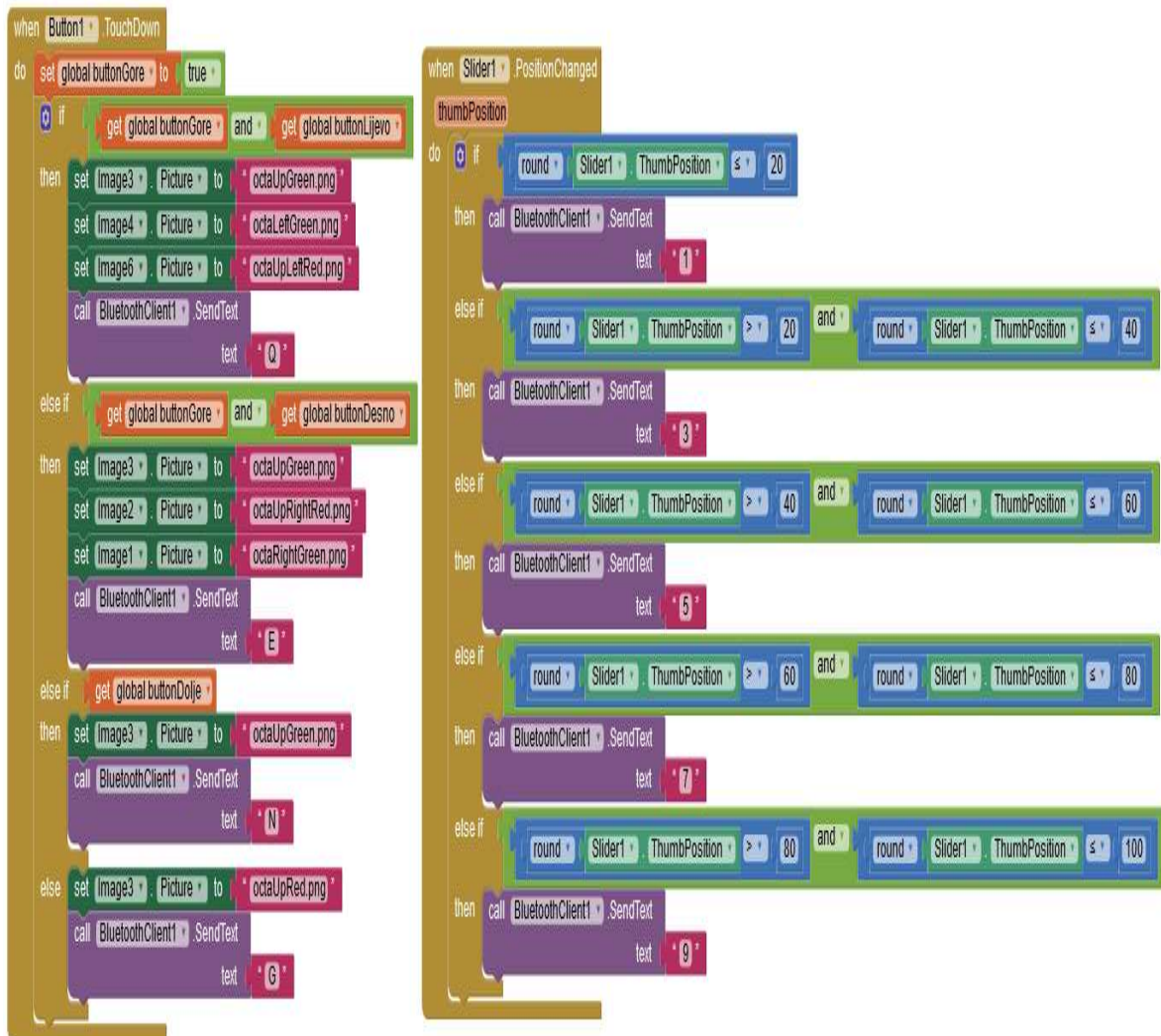
```

Kod 5.2 Bluetooth veza ostvarena mobilnom aplikacijom

Ako korisnik želi upravljati robotom, mora se povezati bluetooth vezom s mikrokontrolerom sustava. To se postiže pritiskom na crveni gumb u gornjem lijevom kutu slike Slika 5.3, gdje se zatim odabire HC-06 modul za spajanje. Nakon toga, gumb mijenja boju iz crvene u zelenu, što označava da je spajanje uspješno.

Na lijevoj strani slike Slika 5.4 prikazan je izbornik unutar kojeg je prikazana logika koja se može koristiti, kao i opis svake od funkcije, što uvelike olakšava proces dizajniranja aplikacije. Od ostalih funkcionalnosti aplikacije, na sličan način se definiraju ostali elementi u sustavu. Iako je sve pojednostavljeno za što lakšu integraciju, to donosi i neke svoje nedostatke. Blokovi i funkcije koje su ugrađene u platformu su predefinirane, što znači da nije moguće definirati novu funkciju koja ne koristi konfiguraciju App Inventor platforme. Zato se korištenje ove platforme preporučuje za projekte koji nisu namijenjeni masovnoj proizvodnji i za komercijalnu svrhu, nego za projekte kućne automatizacije i izrade. Kako bi neki gumb ispravno radio, tj. da se prilikom pritiska nekog gumba pokaže trenutni smjer kretanja, potrebno je provjeriti jesu li pritisnute još neke tipke u isto vrijeme. Na taj se način može zaključiti jesu li pritisnute dvije tipke istovremeno ili samo jedna. To znači da se prilikom opisivanja logike moraju pokriti svi slučajevi koji imaju smisla. Na primjer, ako korisnik pritisne tipku prema naprijed, aplikacija mora provjeriti jesu li uz tu tipku pritisnute tipka prema lijevo ili prema desno. Razlog tomu je što se robot može kretati ukoso pa se

moraju provjeriti i ostali smjerovi kretanja. Istu stvar je potrebno provjeriti kada korisnik prestane s pritiskom neke tipke. Iako izgleda jednostavno, za svaku tipku su potrebna dva velika bloka sa svim pokrivenim kombinacijama. Jedna od takvih kombinacija je prikazana na idućoj slici, a konfiguracija za ostale tipke je napravljena na identičnoj logici. Stoga, za 8 tipki je potrebno 16 takvih blokova da bi se pokrili svi slučajevi. Osim gumba za kretanje, još je potrebno definirati klizač kojim se šalju naredbe za upravljanje brzinom motora te je na isti način prikazana struktura:



Slika 5.5 Logika gumba za upravljanje smjera robota i klizača za upravljanje brzinom motora

6. Mogućnosti nadogradnje sustava

Metode koje su korištene u ovom radu povezuju nekoliko različitih sustava kao što su Arduino, Google Spreadsheet, internet servisi poput PushingBox-a i programske podrške kao što je App Inventor. Ono što omogućava integracija svih ovih sustava u jedan jest automatizacija procesa, digitalizaciju podataka i odmak od rutinskih poslova. Sve su ovo karakteristike tehnologije Interneta stvari. To je nova velika grana industrije koja ima bezbroj mogućnosti za nadogradnju. U svrhu ovog rada napravljen je jedan takav sustav koji kombinira nekoliko vrsta komunikacije, kao što su GPS, GPRS i bluetooth. Unatoč velikoj kompleksnosti sustava, uvijek je moguće optimizirati postojeće procese, ali i implementirati nove koji će dodatno poboljšati rad cijelog sustava.

Budući da je u današnje vrijeme velika pozornost usmjerena ka razmjeni podataka putem interneta, treba dodatno pripaziti na sigurnost podataka koji se prenose. U primjeru ovog sustava, korisnik ima uvid u bazu koja se nalazi na HTTPS lokaciji. Unatoč tome, problem je što se zahtjev koji šalje SIM modul prvo šalje na HTTP servis, koji zatim preusmjerava zahtjev preko sigurnijeg protokola. Postoje varijacije SIM modula koje dozvoljavaju ovakve zahtjeve, a jedan od načina je da se na modul ugradi nova verzija softvera, koja je ugrađena u SIM800L modul, tzv. *firmware*. Postoje razni podaci na internetu, ali nažalost nema javnog i službenog izvora za preuzimanje takvog softvera. Stoga se u ovom radu nije koristila HTTPS metoda, iako je preporučljiva kako bi podatci ostali sigurni.

Osim sigurnog prijenosa podataka, sustav je dizajniran tako da korisnik mora poslati ključnu riječ putem SMS-a na određen broj telefona kako bi dobio trenutne informacije o sustavu u obliku SMS poruke. Umjesto toga, mogao bi se dodati sloj sigurnosti u vidu provjere broja mobitela onoga tko šalje poruku. Tako bi SIM modul nakon primitka poruke automatski provjerio s kojeg je broja došla poruka, a ako taj broj nije u bazi dozvoljenih brojeva, poruka bi se automatski obrisala iz memorije sklopa i ništa se ne bi dogodilo. Ovo je moguće postići dodatnim filtriranjem u programskoj logici Arduino mikrokontrolera.

S programske strane, odnosno Android aplikacije koja je izrađena za ovaj projekt, također je moguće unaprijediti sustav. Iako se trenutno za upravljanje mobilnim robotom koriste gumbi za odabir željenog smjera kretanja, App Inventor platforma omogućava korištenje akcelerometra za upravljanje robota. Ovaj se senzor nalazi u svim modernim telefonima. Senzor daje podatke o nagibu mobitela u tri osi. Na temelju tih osi mogu se dobiti koordinate

položaja mobitela. S tim podacima, aplikacija bi imala uvid kako je zakrenut mobitel i tako bi se umjesto korištenja gumbova moglo upravljati robotom. Pored toga, App Inventor podržava slanje podataka preko interneta u bazu koju korisnik može definirati. U slučaju da je projekt veći i da sadrži nekoliko robota koji bi istovremeno slali podatke u bazu, na ovaj način bi se omogućilo da svi korisnici, koji imaju preuzetu Android aplikaciju, dobiju uvid u podatke o svim robotima. To je primjereno za veće komercijalne sustave koji raspolažu kritičnim informacijama, a koji su nadgledani od više osoba.

Skladištenje podataka ovog sustava odvija se preko platforme Google Spreadsheet. Ova platforma je otvorenog tipa, što znači da svako s osnovnim znanjem programiranja u jeziku JavaScript ima mogućnost modifikacije nad dokumentom. Tako se u ovom projektu automatizirao prikaz određenih podataka u obliku tablice, grafikona i dodataka poput GeoSheets-a. Međutim, proces je moguće dodatno upotpuniti i prilagoditi kako bi se dobilo bolje korisničko iskustvo. Sustav je napravljen tako da različite podatke sprema u određene stupce. Ovako se uvijek ista vrsta podataka nalazi u specifičnom stupcu tablice. Svakim upisom podataka, skripta koja je programirana u Google Spreadsheetu, umeće novi red. Iako se za malu količinu podataka ovaj način čini prikladan, u slučaju svakodnevnog upisa podataka u intervalu od jedne minute, tablica bi vrlo brzo postala velik izvor informacija koje je teško filtrirati i analizirati. Prema tome, za poboljšanje prikaza podataka nužno je isprogramirati skriptu tako da uspoređuje datum zadnjeg upisanog podatka s datumom novog podatka. U slučaju da se datumi razlikuju, potrebno je napraviti novu stranicu unutar dokumenta i dodijeliti joj ime u obliku novog datuma. Tako bi korisnik imao uvid o podacima po danima.

Zaključak

Sustav mobilnog robota primjenjuje nekoliko komunikacijskih tehnologija koje služe za povezivanje uređaja i za njihovu međusobnu komunikaciju. Korištenjem interneta, u tom procesu razmjene podataka, želi se postići jedinstveno korisničko iskustvo kako bi korisnik u bilo kojem trenutku imao uvid u stanje sustava kojeg posjeduje. U ovom radu je korišteno nekoliko senzora i modula serijske komunikacije, koji su kontrolirani upravljačkom jedinicom Arduino Mega 2560. Prateći nove metode razmjene informacija, pomoću tehnologije *Internet of Things*, izrađen je sustav koji analizira podatke sustava i pohranjuje ih na sigurnu lokaciju na internetu. Iako mikrokontroler, koji je korišten u radu, nema mogućnost paralelnog izvršavanja procesa i naredbi istovremeno, optimizacijom programske logike je postignut slični efekt. Brojni komercijalni sustavi nude korisnicima nekoliko mogućnosti za kontrolu i nadzor. Stoga je u ovom projektu omogućeno slanje podataka na zahtjev korisnika putem SMS-a, pored toga što se podatci šalju automatizirano u jednakim vremenskim intervalima na web sučelje. Tako je ostvaren *customer centric* pristup kojemu teže tvrtke kako bi učvrstili odnos s postojećim klijentima, ali i ostvarili odnos s novim. Prema dobivenim rezultatima iz ovog rada, može se zaključiti da je moguće izraditi složen sustav koji kombinira nekoliko sustava, uključujući slanje podataka na internet i daljinsko upravljanje preko mobilne aplikacije, uz nisku cijenu izrade.

Literatura

- [1] Arduino, *Getting Started with Arduino MEGA2560*, poveznica: <https://www.arduino.cc/en/Guide/ArduinoMega2560>; pristupljeno 5. lipnja 2020.
- [2] Google, *Blockly reference*, poveznica: <https://developers.google.com/blockly>; pristupljeno 20. lipnja 2020.
- [3] Google, *Extending Google Sheets*, poveznica: <https://developers.google.com/apps-script/guides/sheets>; pristupljeno 21. lipnja 2020.
- [4] SimCom, *SIM800L Hardware Design*, (2013, kolovoz), poveznica: https://img.filipeflop.com/files/download/Datasheet_SIM800L.pdf; pristupljeno 25. lipnja 2020.
- [5] Components101, *HC-06 Bluetooth Module*, (2018, studeni), poveznica: <https://components101.com/wireless/hc-06-bluetooth-module-pinout-datasheet>; pristupljeno 22. lipnja 2020.
- [6] U-blox, *NEO M-8*, (2020, travanj), poveznica: https://www.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_%28UBX-15031086%29.pdf; pristupljeno 25. svibnja 2020.
- [7] Rajguru Eletronics, *Lm393 Motor Speed Measuring Sensor Module For Arduino*, poveznica: <https://5.imimg.com/data5/VQ/DC/MY-1833510/lm393-motor-speed-measuring-sensor-module-for-arduino.pdf>; pristupljeno 15. lipnja 2020.
- [8] SparkFun Electronics, *Ultrasonic Ranging Module HC-SR04*, poveznica: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>; pristupljeno 25. svibnja 2020.
- [9] Handson Technology, *L298N Dual H-Bridge Motor Driver*, poveznica: <http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf>; pristupljeno 25. svibnja 2020.
- [10] Thamer, A. *The Future of the Internet of Things, Int'l Journal of Computing, Communications & Instrumentation Engg. (IJCCIE) Vol. 4, Issue 1 (2017)*, str. 56-60
- [11] Statista, *Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025*, (2016, studeni), poveznica: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>; pristupljeno 27. lipnja 2020.
- [12] Krasniqi, K., Hajrizi, E. *Use of IoT Technology to Drive the Automotive Industry from Connected to Full Autonomous Vehicles*, 2016., str. 269-274.
- [13] Khan, M. A., Khan, S. F. *IoT based framework for Vehicle Over-speed detection*, 2018.
- [14] PushingBox, *API*, poveznica: <https://www.pushingbox.com/api.php>; pristupljeno 27. lipnja 2020.
- [15] MIT App Inventor, *The MIT App Inventor Library: Documentation & Support*, poveznica: <https://appinventor.mit.edu/explore/library>; pristupljeno 23. lipnja 2020.

Sažetak

Izvedba daljinski upravljano mobilnog robota

U ovom radu je opisana arhitektura mobilnog robota, način rada svake komponente i njihova integracija u složen sustav koji primjenjuje tehnologiju *Internet of Things*. Velika pozornost je posvećena načinu prikaza podataka prema korisniku, što je ostvareno slanjem informacija preko interneta ili putem SMS-a. Također, objašnjena je programska logika koja omogućava automatiziran rad sustava i daljinsko upravljanje preko mobilne aplikacije. Zahvaljujući korištenim alatima i web servisima, koji se temelje na otvorenom kodu, lako je modificirati i prilagoditi funkcije potrebne za izradu robota. Rezultat rada je sustav koji povezuje nekoliko platformi u jednu, što daje brojne mogućnosti korisniku za daljnjom nadogradnjom, a to je jedna od odlika IoT sustava.

Ključne riječi: IoT, daljinsko upravljanje, mobilni robot, senzori, automatizacija, serijska komunikacija

Summary

Realization of Remotely Controlled Mobile Robot

This paper describes the architecture of a mobile robot, the way each component works and their integration into a complex system that uses the Internet of Things technology. Special attention is paid to the way data is displayed to the user, which is achieved by sending information over the internet or via SMS. Also, the program logic is shown and described, which enables automated system operation and remote control via a mobile application. With the help of tools and web services used in this project, which are based on open source technologies, it is easy to modify and adapt functions needed to create a robot. The result is a system that connects several platforms into one, which gives the user many opportunities for further upgrade, and this is one of the features of the IoT system.

Keywords: IoT, remote control, mobile robot, sensors, automatization, Arduino, serial communication