

# Primjena ESP8266 komunikacijskog modula u udaljenom upravljanju

---

**Puljko, Benjamin**

**Undergraduate thesis / Završni rad**

**2017**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:098909>

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

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

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni preddiplomski studij računarstva**

**PRIMJENA ESP8266 KOMUNIKACIJSKOG MODULA U  
UDALJENOM UPRAVLJANJU**

**Završni rad**

**Benjamin Puljko**

**Osijek, 2017.**

# SADRŽAJ

1. UVOD .....	1
1.1. Zadatak završnog rada.....	1
2. ESP8266 KOMUNIKACIJSKI MODUL U PRIMJENI UPRAVLJANJA ROBOTSKIM MOBILNIM SUSTAVOM .....	2
2.1. Sustavi mobilnosti kod mobilnih platformi .....	2
2.2. Udaljeno upravljanje mehaničkim mobilnim platformama .....	2
2.3. Bežične komunikacijske tehnologije i rješenja u 2.4GHz spektru .....	3
2.4. Idejno rješenje sustava upravljanja robotskim kolicima korištenjem ESP8266.....	3
3. REALIZACIJA SUSTAVA UPRAVLJANJA ROBOTSKIM KOLICIMA TEMELJENOM NA ESP8266.....	5
3.1. Komponente korištene u realizaciji sustava .....	5
3.1.1. Arduino.....	5
3.1.2. ESP8266 ESP-12F.....	9
3.1.3. HC-SR04 .....	10
3.1.4. DHT22.....	11
3.1.5. LM393 breakout .....	11
3.1.6. L298N.....	12
3.2. Funkcijski dijagram sustava .....	13
3.3. Električna shema sustava.....	14
3.4. Upravljački blok dijagram sustava .....	15
3.5. Način rada ESP8266 komunikacijskog modula .....	16
3.5.1. Sloj veze .....	16
3.5.2. Mrežni sloj.....	16
3.5.3. Transportni sloj.....	17
3.5.4. Sloj primjene .....	17
3.6. Programska rješenja sustava.....	18
3.6.1. HC-SR04 ultrazvučni senzor .....	18
3.6.2. LM393 senzor svjetlosti .....	20
3.6.3. DHT22 senzor .....	21
3.6.4. L298N H-most.....	22
3.7. Programiranje ESP8266 komunikacijskog modula .....	25
3.8. Izrada aplikacije .....	27

3.9.	Povezivanje aplikacije s ESP8266.....	28
3.10.	Izrada web stranice.....	29
4.	TESTIRANJE I REZULTATI .....	32
4.1.	Senzori.....	32
4.1.1.	DHT22.....	32
4.1.2.	HCSR-04 .....	33
4.1.3.	LM393 .....	35
4.2.	Robotska kolica .....	36
5.	ZAKLJUČAK .....	37
	LITERATURA.....	38
	SAŽETAK.....	39
	ABSTRACT .....	40
	ŽIVOTOPIS .....	41

# 1. UVOD

Tema ovog završnog rada je primjena ESP8266 komunikacijskog modula u udaljenom upravljanju. Primjer udaljenog upravljanja bit će robotska kolica kojima će se upravljati pomoću mobilnog uređaja. U ovom primjeru udaljenog upravljanja, ESP8266 komunikacijski modul koristiti će se samo za komunikaciju, dok će „mozak“ robotskih kolica biti Croduino(standardna hrvatska Arduino kompatibilna pločica).

Budući da će se programirati Croduino, najbolje rješenje je korištenje integriranog programskog okruženja IDE koji se razvio za Arduino i koje je dostupno za mnoge operacijske sustave. Ovaj IDE je razvijen u programskom jeziku Java te je izveden iz jednog prijašnjeg projekta za obradu programskih jezika te obradu užičenih platformi.

Primjene ESP8266 modula gotovo su beskonačne. Možemo automatizirati otvaranje garažnih vrata putem mobitela, uz detektor pokreta spojite ESP8266 da dobijemo e-mail svaki puta kada se izvan dopuštenog vremena netko kreće našim posjedom, uključimo klimu ili grijanje putem aplikacije, smanjimo svjetlinu svoje noćne lampe također putem aplikacije... Dakle gotovo sve što nam padne na pamet može se povezati s ESP8266 modulom.

## 1.1.Zadatak završnog rada

Zadatak završnog rada je primijeniti ESP8266 komunikacijski modul u udaljenom upravljanju. Za primjer udaljenog upravljanja koriste se robotska kolica kojima se upravlja pomoću mobilnog uređaja. ESP8266 komunikacijski modul spojen je preko Arduina koji kontrolira robotska kolica. Svo programiranje odrađeno je u Arduino IDE-u.

## **2. ESP8266 KOMUNIKACIJSKI MODUL U PRIMJENI UPRAVLJANJA ROBOTSKIM MOBILNIM SUSTAVOM**

### **2.1.Sustavi mobilnosti kod mobilnih platformi**

Većina postojećih mobilnih sustava može se svrstati u pet različitih podsustava: kretanje, sučelje čovjek-stroj, senzori, izvor napajanja i manipulator. Pri projektiranju mobilnih platformi postoje dva glavna zahtjeva: prvo, dizajniranje za okoliš i drugo, za zadatak. Okoliš se može povezati s poznatim zadacima. To možemo vidjeti analizirajući postojeće mobilne platforme. Prizemni mobilni sustavi uglavnom su na kotačima, gusjenicama ili nogama. Nedavno su podvodna vozila postigla veću pozornost. Na primjer, Jason Jr je ispisao povijest promatrajući potopljeni luksuzni brod, titanik. Sljedeći veliki val mobilnih platformi biti će u svemiru. Ovo će se razvijati kao posljedica NASA-ine posvećenosti svemirskoj postaji. Ti će se sustavi povezati s postojećim sustavima pa čak i biti samostalni sustavi slobodnog leta. Svemirske mobilne platforme slične su podvodnim s vrlo malo razlika. Njihova istovjetnost uključuje misije i stupnjeve slobode. Pitanja stabilnosti i komunikacije nerazdvojivi su kako u sustavima tako i u okolišu.

### **2.2.Udaljeno upravljanje mehaničkim mobilnim platformama**

Telerobotika je područje robotike koje se bavi udaljenim upravljanjem poluautonomnih robota, koristeći bežične mreže(kao wi-fi, bluetooth i sl.) ili povezane spojeve. To je kombinacija dviju velikih podpodručja, teleoperacije i teleprisutnosi. Teleoperacija označava rad sustava ili stroja na daljinu, dok teleprisutnost označava skup tehnologija koje omogućuju osobi da se osjeća kao da je prisutna, da daje izgled prisustva ili da ima učinak, putem telerobotike, na mjestu koje nije njihovo stvarno mjesto.

Telemanipulator ili teleoperator je uređaj daljinski upravljani od strane čovjeka. Ako takav uređaj ima sposobnost obavljanja samostalnog rada, naziva se telerobot. Ako je uređaj potpuno autonoman, naziva se robot. U jednostavnim slučajevima, naredbe kontrolnog operatera odgovaraju izravno radnjama u kontroliranom uređaju.

Sučelje za udaljeno upravljanje može biti jednostavno kao na primjer monitor-miš-tipkovnica sučelje. Telerobotika koju pokreću internetske veze često su takve vrste. Vrijedna izmjena za

takvu vrstu sučelja je joystick, koji pruža intuitivnu navigacijsku shemu za planarno kretanje robota.



*Slika 1. Telemanipulator za istraživanje potencijalno eksplozivnih uređaja*

### **2.3. Bežične komunikacijske tehnologije i rješenja u 2.4GHz spektru**

Živimo svoj život na 2.4 GHz. Ruteri, bežični telefoni, bluetooth slušalice, baby monitori i otvarači garaže svi žive na toj frekvenciji. Područje frekvencije oko 2.4 GHz rezervirano je za industrijsku, znanstvenu i medicinsku(ISM) uporabu. Pošto su ti uređaji svi na malom području oko 2.4 GHz, tada njihovi radio valovi neće interferirati s, recimo, mobilnim uređajima koji rade u 1.9 GHz spektru ili AM radiju, koji emitira između 535 kHz i 1.7 MHz. Sadašnji ISM standardi su osnovani 1985. g. i to baš na vrijeme. Telefoni su počeli gubiti kablove i u bliskoj budućnosti, širokopoljasne internetske veze postaju bežične. Svi ti uređaji trebali su frekvenciju koja ne treba licencu, a da je smještena između onih koje trebaju. Frekvencije koje nisu toliko visoke da bi žrtvovala prodiranje emitiranja(kroz zidove i sl.), a nisu ni toliko niske da im je potrebna antena od pola metra.

### **2.4. Idejno rješenje sustava upravljanja robotskim kolicima korištenjem ESP8266**

Robotska kolica biti će spojena na sljedeći način: S donje strane okvira robotskih kolica nalaziti će se baterija, H-most za upravljanje dc motorima te dc motori na koje su nataknuti plastični točkovi. Na sredini donje strane montiran je okretni pomoćni točak koji pomaže u stabilnosti te okretanju robota u stranu. S gornje strane okvira stavljen je protoboard na kojemu ćemo spajati sve žice. HC-SR04 senzor udaljenosti je postavljen s prednje strane robotskih kolica, dok su DHT22 i LM393 breakout senzori postavljeni na protoboard. Kao „mozak“ robotskih kolica koristiti će se Croduino Nova koji na sebi ima spojen ESP8266 modul i nije potrebno posebno povezivanje ESP-a i Croduina, što dodatno olakšava posao. Robotskim kolicima moći će se upravljati preko mobilne aplikacije za android sustave u koju ćete morati (nakon pokretanja) unijeti IP adresu robotskih kolica te će imati naredbe za kretanje naprijed, nazad, lijevo, desno i stop.



### **3. REALIZACIJA SUSTAVA UPRAVLJANJA ROBOTSKIM KOLICIMA TEMELJENOM NA ESP8266**

#### **3.1. Komponente korištene u realizaciji sustava**

##### **3.1.1. Arduino**

Arduino je univerzalni mikrokontroler zasnovan na Atmel tehnologiji i idealan je za razvoj upravljačke elektronike i robotike. Platforma je otvorenog koda temeljena na jednostavnoj razvojnoj pločici s ulazno/izlaznim konektorima i besplatnom programskom podrškom s jednostavnim korisničkim sučeljem. Programiranje uređaja se izvodi pomoću integriranog razvojnog okruženja, koje postoji za Windows, Mac i Linux operacijski sustav, u programskom jeziku sličnom C-u. Glavna je namjena cijelog sustava komuniciranje s različitim hardverom koji je na njega priključen. Arduino je razvijen u Italiji za poučavanje programiranja u C/C++ programskim jezicima. Napravljen je od open source hardverskih komponenti koje su spojene na 8-bitni Atmel AVR mikrokontroler ili 32-bitni ARM procesor. Svi se modeli mogu proširivati tzv. shield-ovima, gotovim PCB-ovima koji imaju različite funkcije.

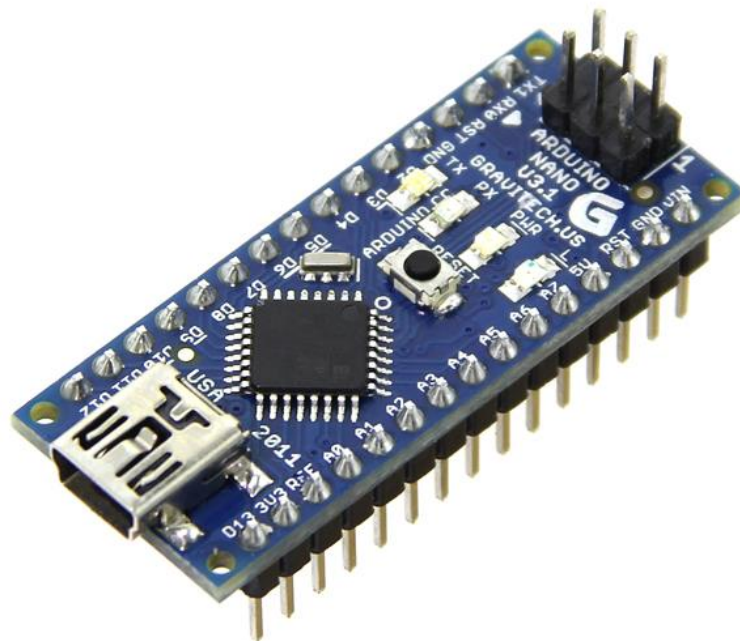
Arduino se programira putem Arduino Software-a, besplatnim IDE-om u kojem se piše C/C++ programski kod i prebacuje na Arduino. On radi na Windows platformi, MAC OS X-u i Linuxu, a može se doći i do izvornog koda. S IDE-om dolazi velik broj primjera koji mogu dobro poslužiti za upoznavanje s programiranjem.[1]

Arduino NANO je model sa sljedećom tehničkom specifikacijom:

- Mikroprocesor : ATmega328
- Frekvencija procesora : 16MHz
- Radni napon : 5V
- Ulazni napon (preporučeno): 7 -12V
- Digitalni I/O pinovi : 22 (od toga moguće 6 PWM izlaza)

- Analogni I/O pinovi : 8
- DC struja za I/O pinove: 40mA
- DC struja za 3,3V pin: 50mA
- Flash memorija: 32 kB (0,5kB rezervirano za bootloader)
- SRAM: 2 kB
- EEPROM: 1 kB
- VELIČINA : 18 x 45 mm

Arduino se razvio od sveučilišnog projekta kojemu je cilj bio pojednostaviti studentima izradu projekata potrebnih za učenje. Danas je Arduino platforma otvorenog tipa (open-source) načinjena za elektroničke projekte i "oživljavanje" okoline oko nas. Baziran je na jednostavnom hardveru i softveru tako da je dostupan svakome uz minimalnu prilagodbu. Namijenjen je svima, u profesionalne svrhe, a ponajprije kreativnima. [2]



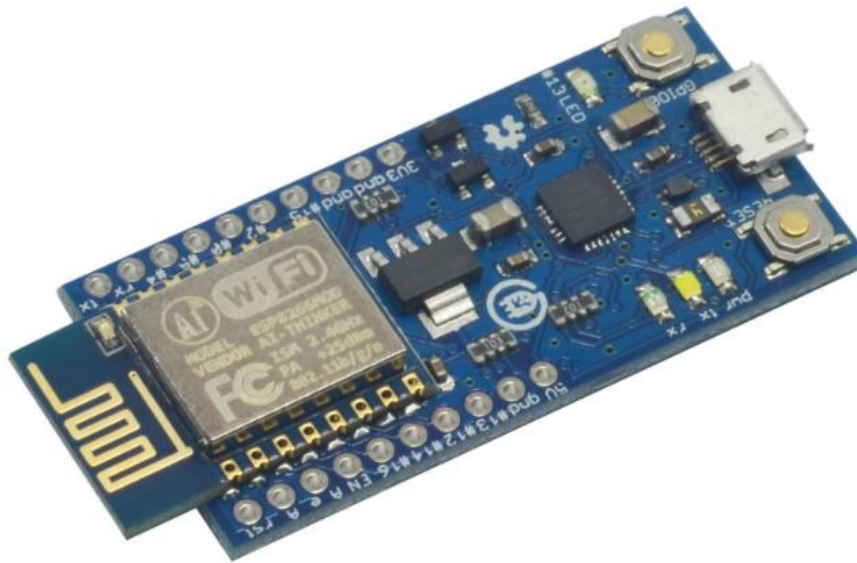
*Slika 2. Arduino Nano*

### **3.1.1.1. Croduino Nova**

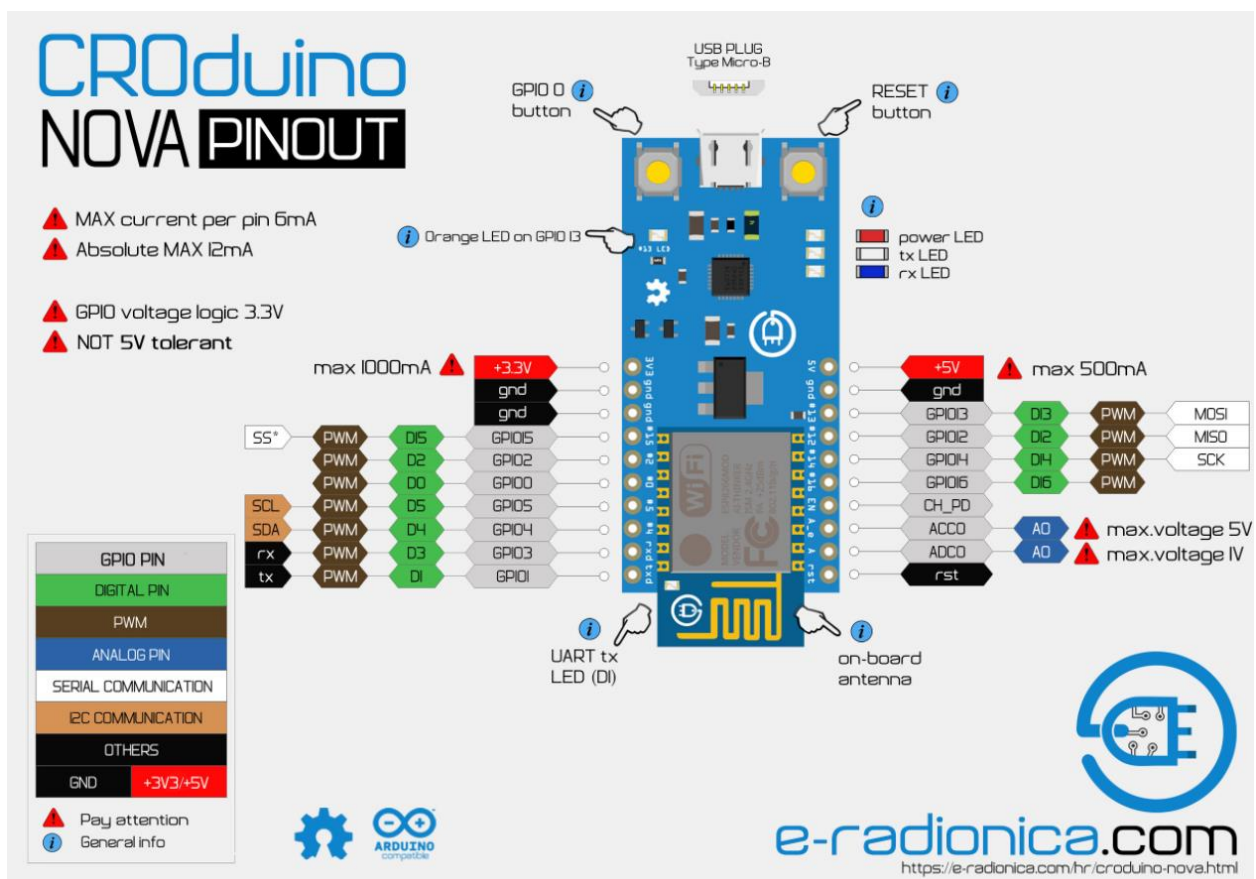
Budući da u izradi robotskih kolica koristimo Croduino, a ne Arduino Nano, moramo dodatno pojasniti što je to Croduino. Croduino Nova je mikrokontrolerska pločica s mogućnošću konekcije na WiFi mrežu, pa tako i na internet. Baziran je na već dobro poznatom ESP8266 kojega je inače samostalno vrlo komplicirano koristiti. Sa NOVOM, jednostavno ukopčate pločicu u USB i krenite programirati svoj idući projekt koji će biti povezan na internet(IoT). Mozak ove pločice je već spomenuti ESP8266 proizvođača Espressif. Dolazi s potpunim WiFijem(sa client i server strane) i integriranim TCP/IP stogom koji podržava DNS. Najvažnije od svega- programibilna je iz Arduina, čak nema ni potrebe za ručnim resetiranjem pločice. Radi na frekvenciji od 80MHz, ima čak 1MB prostora za kod(0.8MB dostupno, bootloader 0.2MB), te 82kB RAM memorije(49kB na korištenje). Pločica podržava I2C, SPI i serijsku komunikaciju, a ima 9 GPIO pinova od kojih svi podržavaju PWM i jedan analogni ulaz.[3]

Croduino Nova ima sljedeće karakteristike:

- ESP8266 u pakiranju ESP-12
- 80MHz, 1MB flash, 82kB RAM
- Programibilno iz Arduina
- 9 GPIO pinova od kojih svi podržavaju PWM
- 1 analogni ulaz(ADC), max napon 1V i 5V
- I2C, SPI, serijska komunikacija
- CP2102 USB na UART bridge s micro USB konektorom
- 3.3V regulator napona
- LED diode za: napajanje, tx, rx, GPIO13
- Hardverski reset prilikom uploada
- Dva pushbuttona: reset i GPIO0
- FCC i CE certifikati za ESP-12



Slika 3. Croduino Nova



Slika 4. Grafički pinout Croduino Nova

### 3.1.2. ESP8266 ESP-12F

ESP8266 je jeftin Wi-Fi čip s punim TCP/IP snopom i MCU(mikrokontroler jedinica) sposobnosti koju proizvodi kineski proizvođač, Espressif Systems. Čip je prvi put došao do zapadnih proizvođača u kolovozu 2014. g. s modulom ESP-01, kojeg je izradio proizvođač treće strane, AI-Thinker. Ovaj mali modul omogućuje mikrokontrolerima povezivanje s Wi-Fi mrežom i stvaranje jednostavnih TCP/IP veza koristeći naredbe Hayes-ovim stilom. Međutim, u to vrijeme nije bilo skoro nikakve dokumentacije na engleskom jeziku na čipu kao ni naredbi koje je prihvaćao. Niska cijena i činjenica da je na modulu bilo vrlo malo vanjskih komponenti koje su ukazivale da bi na kraju modul mogao biti vrlo jeftin, privukla je mnoge hakere da istraže modul, čip i softver na njemu, kao i prevođenje kineske dokumentacije. [4]

Neke karakteristike:

- Wireless standard 802.11 b/g/n
- Radni napon: 3.0 - 3.6V (preporuča se 3.3V)
- Veličina : 16 x 24 x 3 mm
- Raspon frekvencije: 2.4 GHz - 2.5 GHz
- Radna temperatura: -40 °C ~ 125 °C



Slika 5. ESP8266 ESP-12F komunikacijski modul

### 3.1.3. HC-SR04

HC-SR04 ultrazvučni je senzor s kojime možete izmjeriti udaljenost između njega samog i predmeta koji se nalazi ispred senzora. Senzor emitira ultrazvučni val frekvencije 40 kHz koji prolazi kroz zrak i ako se na njegovom putu nađe neka prepreka, val se odbije nazad do modula. Uzimajući u obzir vrijeme puta vala te brzinu zvuka, možemo izračunati udaljenost. Modul ima 4 pina: gnd, vcc, trig i echo. Gnd je nula, vcc je plus koji mora biti spojen na 5V, dok su trig i echo pinovi spojeni na digitalne I/O pinove arduina. [5][6]

Karakteristike senzora:

- Domet: 2cm - 4m
- Preciznost: 3mm
- Kut mjerenja: 15°
- Napon: 5V
- Struja: <2mA



*Slika 6. HC-SR04 ultrazvučni senzor*

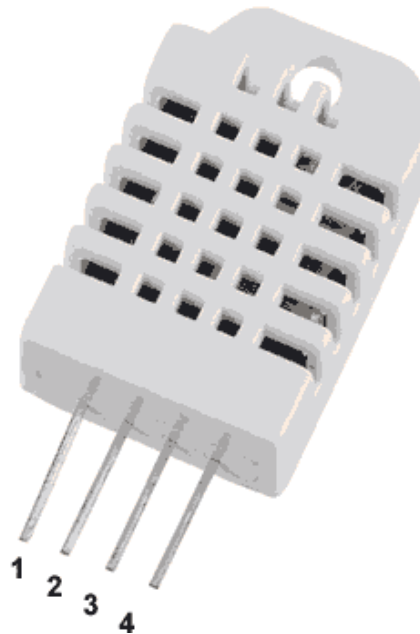
### 3.1.4. DHT22

DHT22(poznat i pod nazivom AM2302) je jednostavni senzor temperature i vlage s preciznijim i točnijim očitanjima od njegovog prethodnika DHT11. Koristi kapacitivni senzor vlage i termički otpornik za mjerenje zraka koji ga okružuje i vraća digitalni signal na data pin. Spajanje je vrlo jednostavno, prvi pin lijevo je 3-5V napajanje, drugi je data pin te sasvim desno je pin gnd(nula). Treći pin je pin nc(not connected) koji ne koristimo. [7]

Karakteristike senzora:

- Radni napon: 3.3V-6V DC
- Mjerno područje: Vlaga 0-100% / Temperatura -40°C – 125°C
- Preciznost mjerenja: Vlažnost  $\pm 2\%$  , Temperatura  $\pm 0.5\text{ }^{\circ}\text{C}$

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



Slika 7. DHT22 pinovi

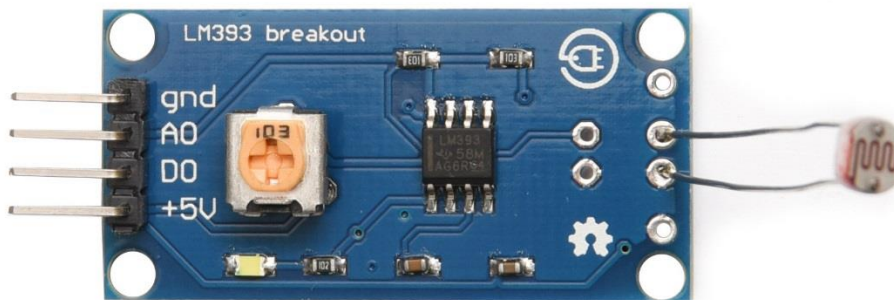
### 3.1.5. LM393 breakout

Na ovom modulu nalazi se komponenta koja je sposobna detektirati svjetlost koristeći fotootpornik. Pomoću ugrađenog komparatora na svojim izlazima nam daje ili

digitalni(HIGH/LOW) ili analogni(0V-5V) signal, ovisno o jačini svjetlosti. Spajamo ili digitalni ili analogni pin, ovisno o tome kakav ispis jačine svjetlosti želimo. [8]

Karakteristike:

- Radni napon: 3V – 5V
- Izlaz: Digitalni i analogni
- Dimenzije: 40 x 20 mm
- Na pločici se nalazi LM393 komparator



*Slika 8. LM393 breakout*

### 3.1.6. L298N

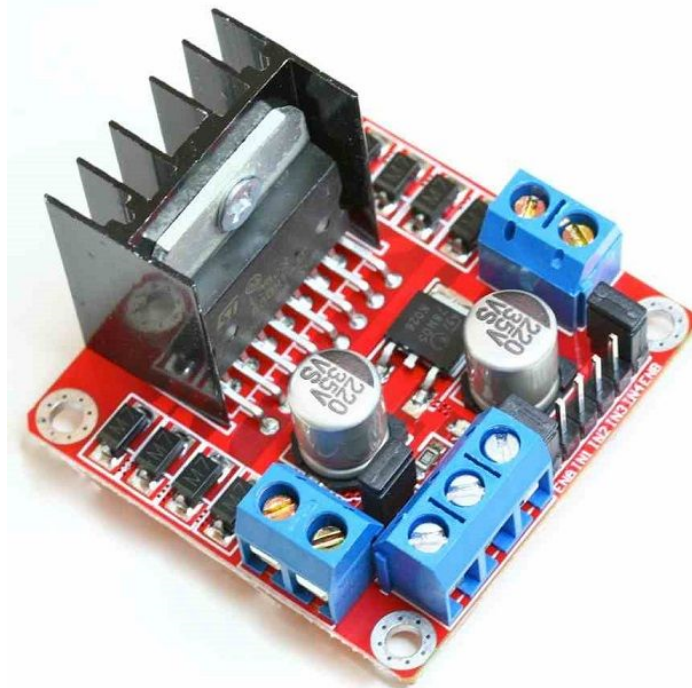
L298N je H most koji omogućava zamjenu polariteta na svojim izlazima pa tako omogućava kontrolu vrtnje motora u oba smjera na vrlo jednostavan način. Također može upravljati i drugim uređajima koji za određenu potrebu zahtijevaju zamjenu polariteta na svojim ulaznim krajevima. [9]

Karakteristike:

- Max snaga: 25W
- Napon: 5-35V(za motor), 5V za logiku



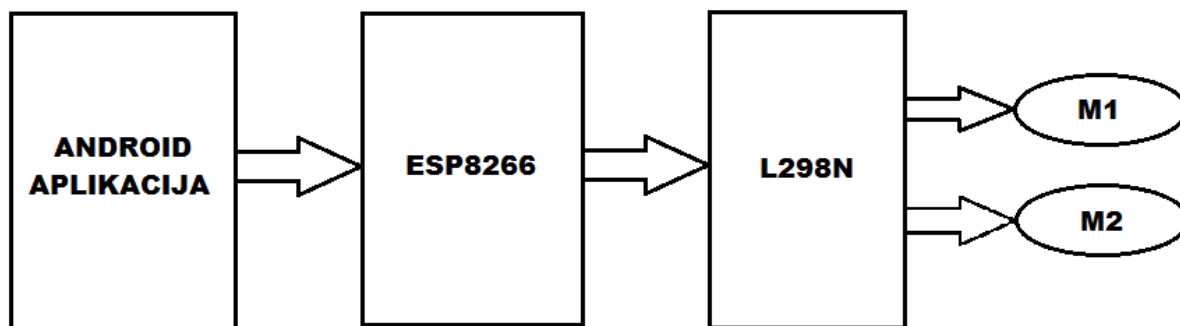
- Struja: 2A
- Dimenzije: 43 x 43 x 23 mm



*Slika 9. H-most L298N*

### **3.2.Funkcijski dijagram sustava**

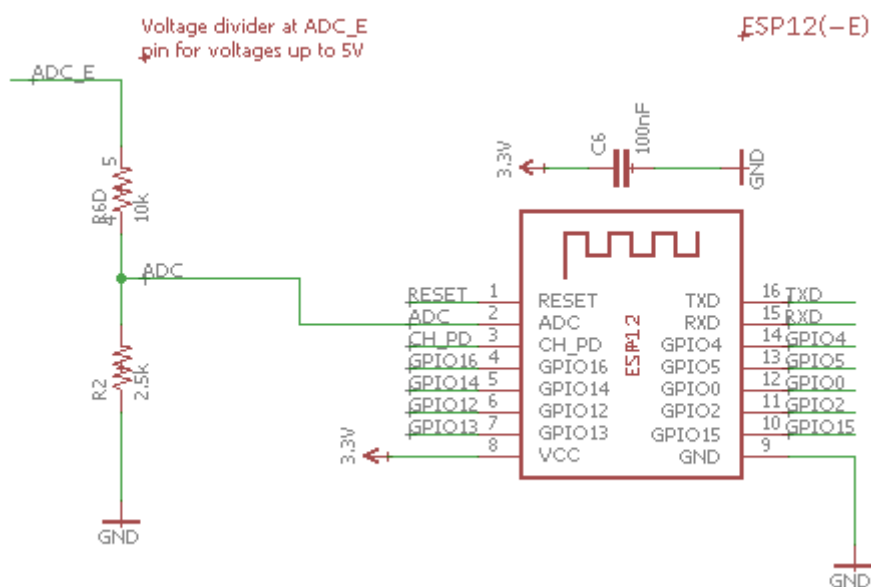
Rad ovog sustava poprilično je jednostavan. Šaljemo naredbu putem android aplikacije. Ukoliko želimo da robot ide naprijed, pritisnemo strjelicu prema gore i dok ju držimo pritisnutu, robot ide naprijed. Čim tipku pustimo, robot stane. Aplikacija šalje zahtjev ESP-u te on u svom programu provjerava koja je naredba zatražena. Prema zadanoj naredbi šalje u programu definirane vrijednosti L298N mostu(high ili low) te one pokreću motore 1 i 2 u određenom smjeru. Funkcijski dijagram možemo vidjeti na slici ispod.



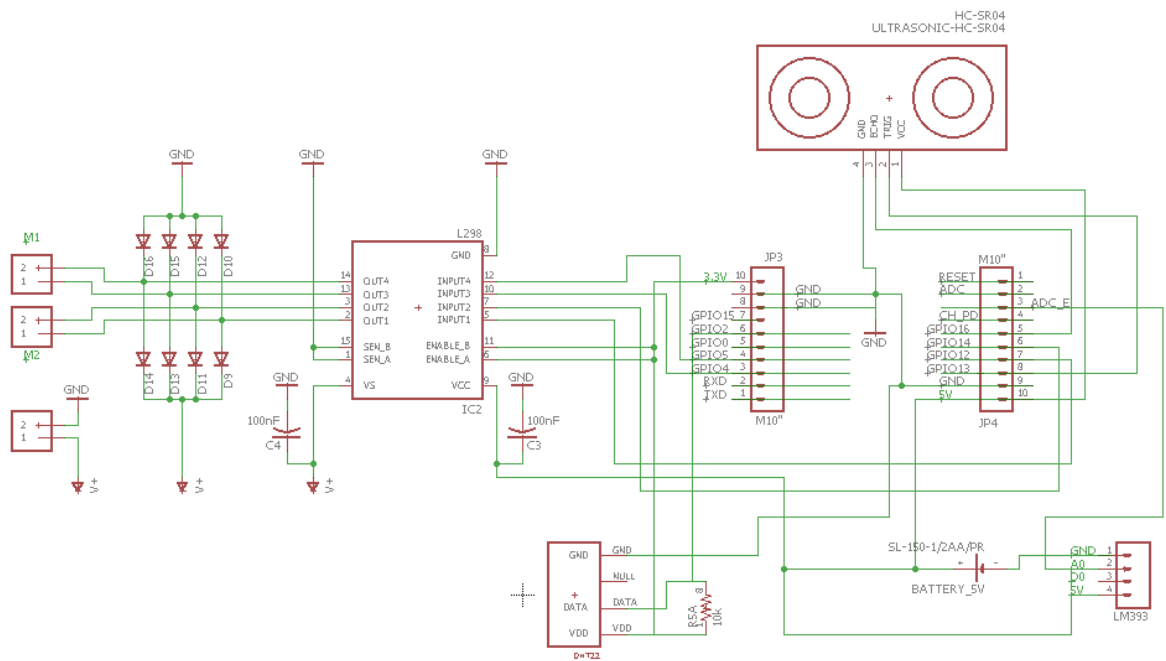
Slika 10. Funkcijski dijagram sustava

### 3.3. Električna shema sustava

Spajanje svih elemenata je bilo čisto shematski, s time da se moralo paziti koliki je napon dozvoljen na određenim modulima. Za pokretanje ESP-a maksimalni dopušten napon je 3.3V, no pošto smo spajali s Croduino Novom, ona je dizajnirana da može primiti napajanje od 5V. Standardni ESP8266 komunikacijski modul ima nešto manje pinova nego na Croduino Novi, no dobro nam je došlo zbog senzora. Shema je napravljena u programu Eagle.

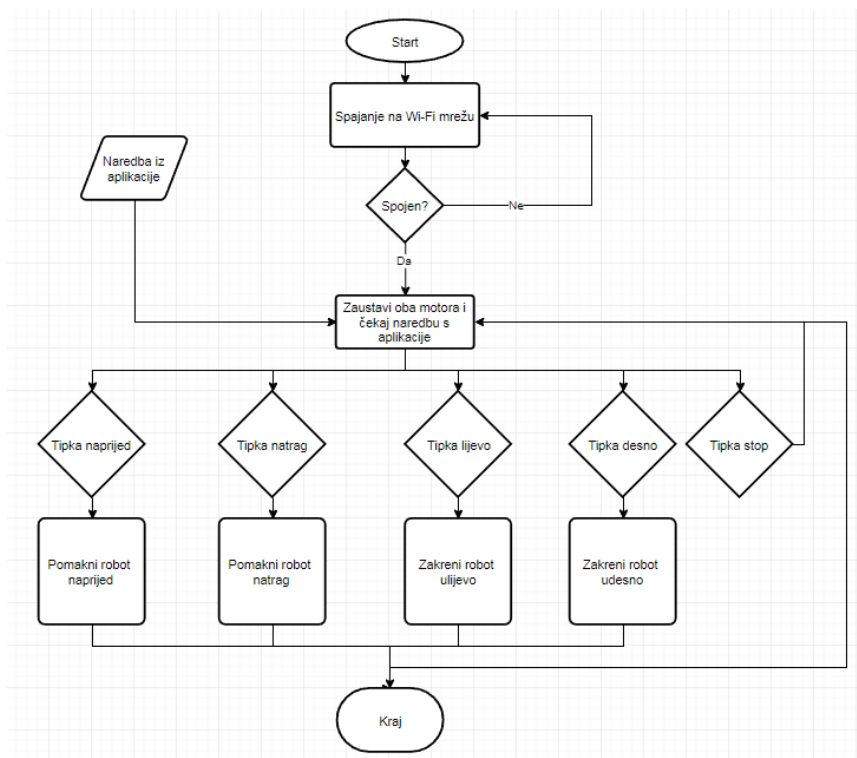


Slika 11. Pinout shema ESP12 modula



Slika 12. Električna shema sustava

### 3.4. Upravljački blok dijagram sustava



Slika 13. Blok dijagram upravljanja sustavom

### 3.5. Način rada ESP8266 komunikacijskog modula

Sustav koji ljudi nazivaju *internet* nije samo jedan protokol. To je cijeli niz slojeva protokola, koji se zove TCP/IP stog. [10]

Layer	Protocols
Application	HTTP, FTP, mDNS, WebSocket, OSC ...
Transport	TCP, UDP
Internet	IP
Link	Ethernet, Wi-Fi ...

*Slika 14. Slojevi i protokoli u njima*

#### 3.5.1. Sloj veze

Sloj veze sadrži fizičku vezu između dva uređaja. Na primjer ethernet kabla ili Wi-Fi vezu. Ovo je sloj najbliži hardveru. Da bi spojili ESP8266 na mrežu, moramo stvoriti Wi-Fi link. To možemo učiniti na dva načina: spojiti ESP8266 na neku bežičnu mrežnu točku ili da ESP8266 bude pristupna točka i bežične stanice se mogu spojiti na njega. [10]

#### 3.5.2. Mrežni sloj

Iako su uređaji fizički spojeni, ne mogu još komunicirati jer ne znaju gdje poslati poruku. Tu u priču ulazi internet protokol(IP). DHCP server(Dynamic Host Configuration Protocol Server) osigurava da su te adrese jedinstvene. To znači da možete poslati poruku na određenu adresu. Postoje dvije vrste internet protokola: IPv4 i IPv6. IPv6 je poboljšana verzija IPv4 protokola i ima puno više adresa nego IPv4. IP adresa sastoji se od 4 broja, na primjer 192.168.1.5. bi bila neka IP adresa. Zapravo se sastoji od dva dijela, prva tri broja su adresa lokalne mreže, dok je zadnji broj specifičan za uređaj. Koristeći IP adresu možemo pronaći ESP8266 i slati mu naredbe. [10]

### **3.5.3. Transportni sloj**

Različiti uređaji u mreži daju sve od sebe da dostave IP pakete na adrese koje su im zadane, međutim, nije neuobičajeno da se paket izgubi i nikada ne dođe. IP ne može garantirati da su paketi došli u istom redoslijedu u kojemu su slani. To znači da ne možemo „pouzdana“ slati poruka samo koristeći sloj veze i mrežni sloj, jer ne znamo kada i hoće li paket doći i hoće li biti ispravan. Potreban nam je i treći sloj: transportni sloj.

Postoje uglavnom dva protokola koja čine ovaj treći sloj: TCP(Transmission Control Protocol) i UDP(User Datagram Protocol). TCP osigurava da su svi paketi primljeni, da su paketi poslagani po redu te da su oštećeni paketi ponovo poslani. UDP u drugu ruku ne garantira da će svaki paket doći do svoje destinacije, međutim provjerava greške i kada ju pronade, jednostavno uništi paket, bez ponovnog slanja. To znači da nije pouzdan kao TCP, ali je brži i ima puno manje kašnjenje jer ne zahtjeva otvorenu vezu za slanje poruka kao TCP. [10]

### **3.5.4. Sloj primjene**

Sada kada imamo pouzdanu komunikaciju koristeći TCP, još uvijek imamo jedan problem. Zamislimo to ovako: šaljemo pismo, TCP garantira da će pismo biti dostavljeno, ali primatelj ne razumije jezik kojim je pisano i ne zna što bi radio s njime. Zato nam je potreban i četvrti sloj protokola, da dva programa mogu međusobno komunicirati.

HTTP(HyperText Transfer Protocol) je protokol koji koriste i web serveri i klijenti da bi komunicirali. Na primjer kada upisemo <http://www.google.com> u adresnu traku web tražilice, ono šalje HTTP GET zahtjev google web serveru. Server razumije HTTP zahtjev i šalje stranicu kao odgovor. Većina modernih stranica koriste zaštićenu verziju HTTP-a, zvanu HTTPS. Ta zaštićena veza enkriptira podatke zbog sigurnosnih razloga. [10]

### 3.6. Programska rješenja sustava

#### 3.6.1. HC-SR04 ultrazvučni senzor

HC-SR04 ultrazvučni senzor radi na principu da emitira ultrazvučni val frekvencije 40 000 Hertza koji prolazi zrakom i ako se na putu nađe neki predmet ili prepreka odbiti će se nazad u senzor. Uzmemo li u obzir vrijeme puta zvučnog vala i brzinu zvuka, možemo izračunati udaljenost predmeta(prepreke) od modula. Ovaj modul ima 4 pina: ground, Vcc, Trig i Echo. Ground i Vcc pinovi moraju biti spojeni na ground i 5V pinove na Croduino pločici, a Trig i Echo pinovi na bilo koji digitalni I/O pin na Croduinu. Na ovim robotskim kolicima Trig i Echo pinovi su spojeni na GPIO13 i GPIO16 digitalne I/O pinove na Croduinu.



*Slika 15. HC-SR04 senzor na robotskim kolicima*

Sada kada smo spojili senzor potrebno ga je isprogramirati. Arduino kod za ultrazvučni senzor je poprilično jednostavan te su u kodu uz pojedine dijelove zakomentirane bilješke za lakše praćenje. Prije setup funkcije definiramo pinove na koje spajamo senzor te definiramo varijable u koje ćemo spremati vrijednosti izračuna. U setup funkciji odredimo koji pin je output, a koji input te započnemo serijsku komunikaciju. U loop funkciji pišemo kod koji će se stalno ponavljati. Pomoću pulseIn() funkcije šaljemo zvučni val i spremamo ga u varijablu vrijeme, u

varijablu udaljenost spremamo izračun puta koji je taj val prešao te na kraju ispisujemo(nije nužno) vrijednost udaljenosti na serial monitor. Primjer koda možemo vidjeti na priloženoj slici.

```
// definiramo pinove
const int trigPin = 9;
const int echoPin = 10;

// definiramo varijable koje računamo
long vrijeme;
int udaljenost;

void setup() {
  pinMode(trigPin, OUTPUT); // Postavimo trigPin na output
  pinMode(echoPin, INPUT); // Postavimo echoPin na input
  Serial.begin(9600); // Počinjemo serijsku komunikaciju
}
void loop() {
  // Resetiramo trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Postavljamo trigPin na HIGH 10 mikrosekundi
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Očitava echoPin, vraća vrijeme puta zvučnog vala u mikrosekundama
  vrijeme = pulseIn(echoPin, HIGH);
  // Izračunavamo udaljenost
  udaljenost= vrijeme*0.034/2;
  // Ispisuje udaljenost na serial monitor
  Serial.print("Udaljenost: ");
  Serial.println(udaljenost);
}
```

*Slika 16. Primjer koda za HC-SR04 senzor*

Kako bismo pojednostavili kod i ubrzali rad robota, skratili smo pisanje koristeći biblioteku (library) za ultrazvučni senzor. Na slici je prikazan dio koda koji mjeri udaljenost pomoću senzora.

```

#include <Ultrasonic.h>

Ultrasonic ultrasonic(13, 16);

int udaljenost;
void setup() {

  Serial.begin(115200);
  Serial.println();
}
void loop() {

  Serial.print("Udaljenost: ");
  Serial.println(ultrasonic.distanceRead());
  Serial.println(" cm");
  delay(500);

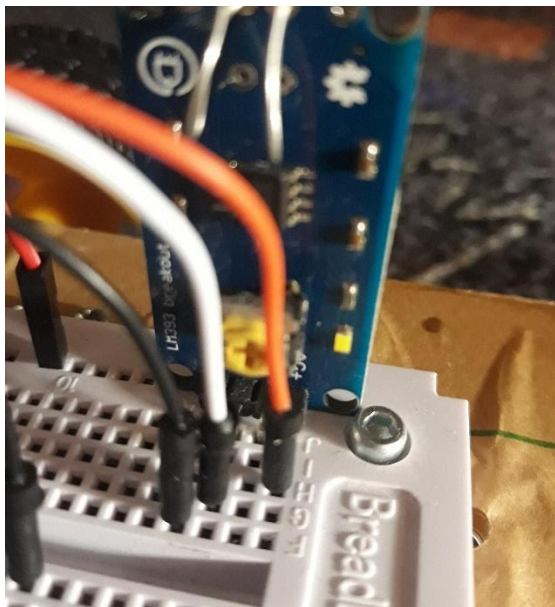
}

```

*Slika 17. Kod za HC-SR04 senzor*

### 3.6.2. LM393 senzor svjetlosti

Gnd pin spojen je na gnd od Croduina, 5v pin spojen je na 5V Croduina te je analogni pin LM393 breakouta spojen na A0 pin na Croduino Nova pločici.



*Slika 18. LM393 breakout spojen na protoboard*

Arduino kod za LM393 je također jednostavan kao i za ultrazvučni senzor. Štoviše, za korištenje ovog senzora nije nam potrebno korištenje biblioteke. Definiramo analogni pin te u loop funkciji očitavamo napon na senzoru.



```

#define ANALOG_SENSOR_PIN A0

int LightAnalogValue;

void setup() {

  Serial.begin(115200);
  Serial.println();
}

void loop() {

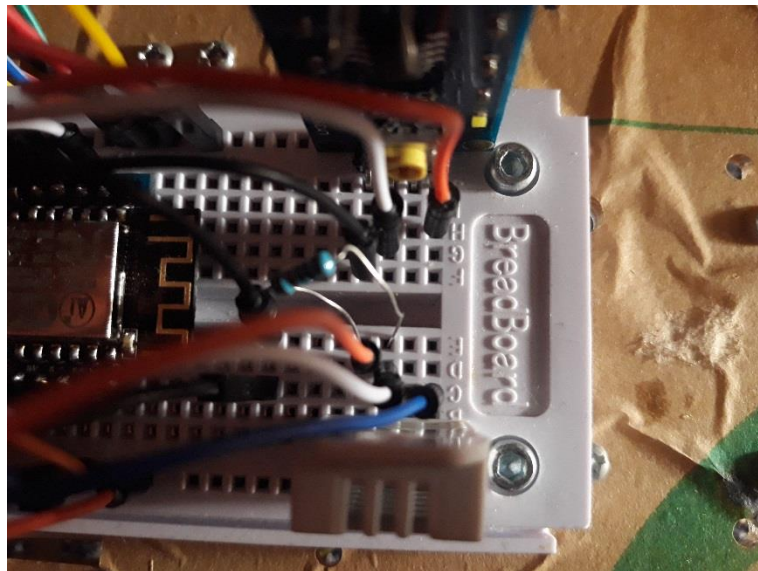
  LightAnalogValue = analogRead(ANALOG_SENSOR_PIN); //Očitaj napon sa senzora
  Serial.print("Količina svjetlosti: "); //Ispisuje vrijednost količine svjetlosti
  Serial.println(LightAnalogValue,DEC); // (manja vrijednost = jača svjetlina)
  delay(1000);
}

```

*Slika 19. Kod za LM393 breakout*

### 3.6.3. DHT22 senzor

Prvi pin DHT22 senzora spojimo na 3.3V jer koristimo ESP8266 koji treba 3.3V napajanje. Drugi pin je data pin. Njega smo spojili na GPIO2 pin Croduina. Treći pin ne spajamo, a četvrti pin je ground koji spajamo na ground Croduina. Potrebno je postaviti pull-up otpornik između prvog i drugog pina kao što je prikazano na slici.



*Slika 20. DHT22 spojen na protoboard*

Kod za DHT22 senzor je preuzet iz example predložka koji se nalazi u DHT biblioteci. Cjelokupan kod možemo vidjeti na sljedećoj slici:

```

#include "DHT.h"

#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  Serial.println("DHTxx test!");

  dht.begin();
}

void loop() {

  delay(2000); //Senzor je spor pa stavljamo delay 2 sekunde

  float h = dht.readHumidity(); //očitaj vlažnost zraka
  float t = dht.readTemperature(); //očitaj temperaturu

  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Vlažnost: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperatura: ");
  Serial.print(t);
  Serial.print(" °C ");

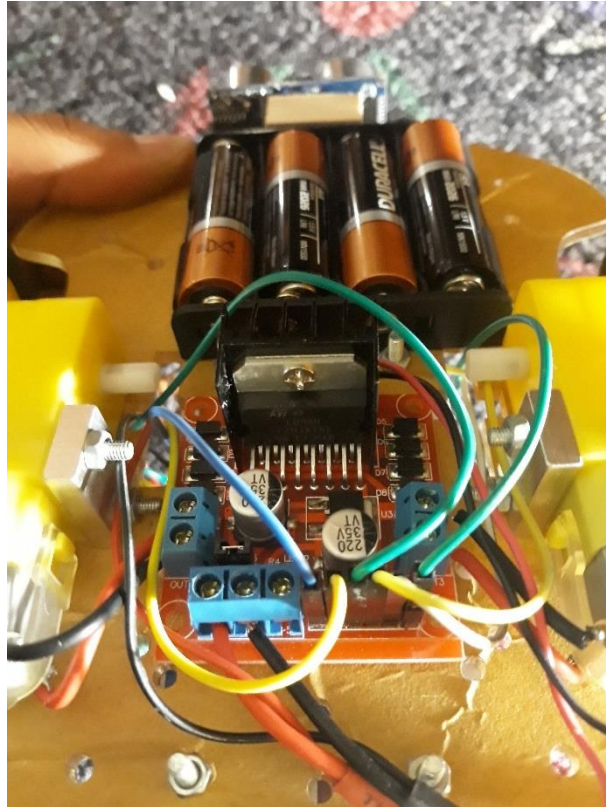
}

```

*Slika 21. Kod za DHT22 senzor*

### 3.6.4. L298N H-most

Spajanje H-mosta je poprilično jednostavno. S lijeve i desne strane nalaze se po dva ulaza(plus i minus) za dva DC motora. Od donje tri stopice, lijeva je za napajanje dok je srednja ground. Desna je 5V output koji nam nije potreban. Desno dolje na H-mostu se nalaze pinovi koji su nam potrebni za pokretanje i smjer vrtnje motora. Prvi pin s lijeve strane je Enable pin za prvi motor, dok je prvi pin s desne strane Enable pin za drugi motor te su oni spojeni na . Između se nalaze 4 pina koje spajamo na Croduino digitalne I/O pinove.



*Slika 22. L298N H-most*

U kodu definiramo pinove preko kojih određujemo smjer kretanja te pravimo funkcije kretanja zbog lakšeg snalaženja u kodu. Loop funkcija je vezana uz izradu aplikacije za kretanje robotskih kolica pa će ona biti objašnjena u daljnjem tekstu.

```

void stop_all_now()
{
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(12, LOW);
    digitalWrite(14, LOW);
}
void left_f(void)
{
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
}
void right_f(void)
{
    digitalWrite(14, LOW);
    digitalWrite(12, HIGH);
}
void left_b(void)
{
    digitalWrite(5, LOW);
    digitalWrite(4, HIGH);
}
void right_b(void)
{
    digitalWrite(14, HIGH);
    digitalWrite(12, LOW);
}
}

```

*Slika 23. Kod za rotaciju motora*

```

void naprijed(void)
{
    left_f();
    right_f();
}
void natrag(void)
{
    left_b();
    right_b();
}
void lijevo(void)
{
    left_b();
    right_f();
}
void desno(void)
{
    left_f();
    right_b();
}
void setup(){
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(14, OUTPUT);
    Serial.begin(115200);
}
}

```

*Slika 24. Funkcije za lakšu obradu koda*

### 3.7. Programiranje ESP8266 komunikacijskog modula

Da bi se moglo upravljati robotskim kolicima preko ESP8266 komunikacijskog modula potrebno je prvo spojiti se na WiFi mrežu te dobiti AP(Access Point) preko kojeg onda možemo kontrolirati sve što je spojeno na naš ESP8266. Jedini problem na koji se naiđe je taj da ESP ne može istovremeno i biti spojen na WiFi mrežu i imati svoju adresu na kojoj možemo jednostavnim HTML jezikom napraviti neko sučelje. Stoga sam napravio program koji se spaja na moju mobilnu WiFi mrežu te mi vrati IP adresu preko kojeg mogu koristiti ESP8266 modul.

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

// SSID i lozinka WiFi mreže na koju se spajamo
const char *ssid = "AndroidAP";
const char *password = "";

WiFiClient client;

WiFiServer server(80);

String data = "";

String checkClient (void)
{
  while(!client.available()) delay(1);
  String s = client.readStringUntil('\r');
  s.remove(0, 5);
  s.remove(s.length()-9,9);
  //return request;
}

void connectWiFi()
{
  Serial.println("Connecting to WIFI");
  WiFi.begin(ssid, password);
  while (!(WiFi.status() == WL_CONNECTED))
  {
    delay(300);
    Serial.print("..");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address is : ");
  Serial.print(WiFi.localIP());
}
```

Slika 25. Spajanje na WiFi mrežu

```

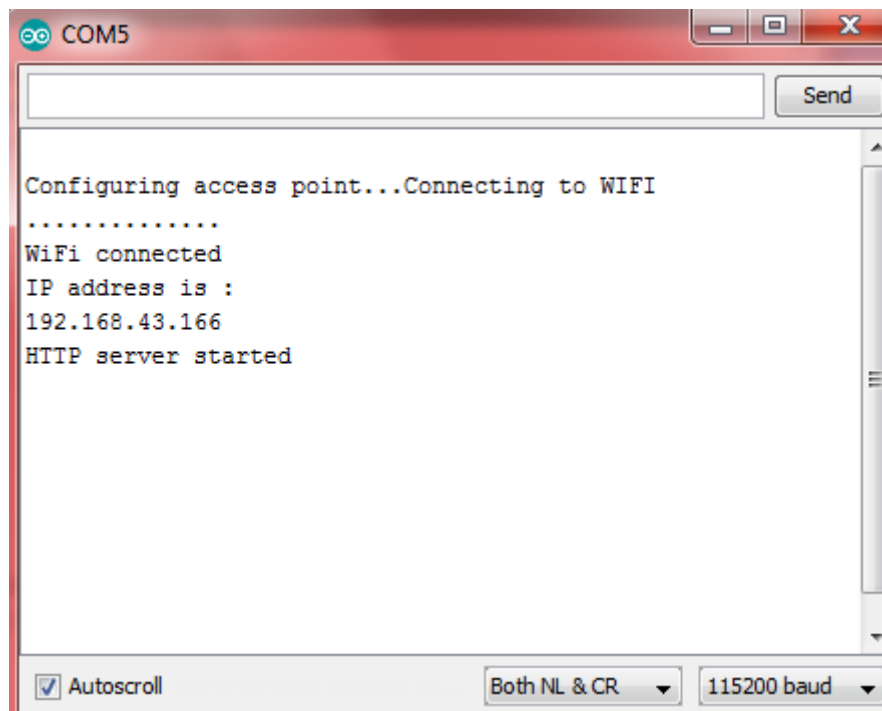
void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.print("Configuring access point...");
  connectWiFi();

  server.begin();
  dht.begin();
  Serial.println("HTTP server started");
}
void loop() {

  client = server.available();
  if (!client) return;
  while(!client.available()) delay(1);
  String s = client.readStringUntil('\r');
  s.remove(0, 5);
  s.remove(s.length()-9, 9);
}

```

*Slika 26. Spajanje na WiFi mrežu*

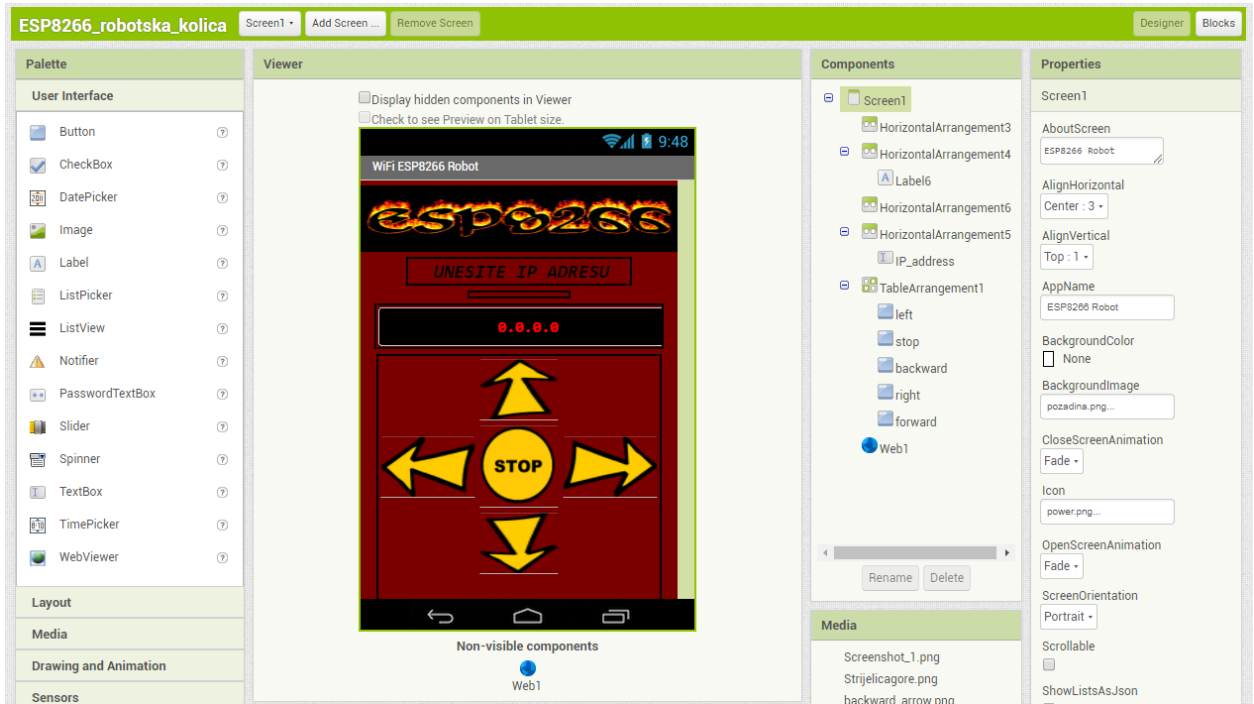


*Slika 27. Spajanje na WiFi*

Sa slike je vidljivo da se ESP8266 uredno spojio na WiFi mrežu i dao nam IP adresu.

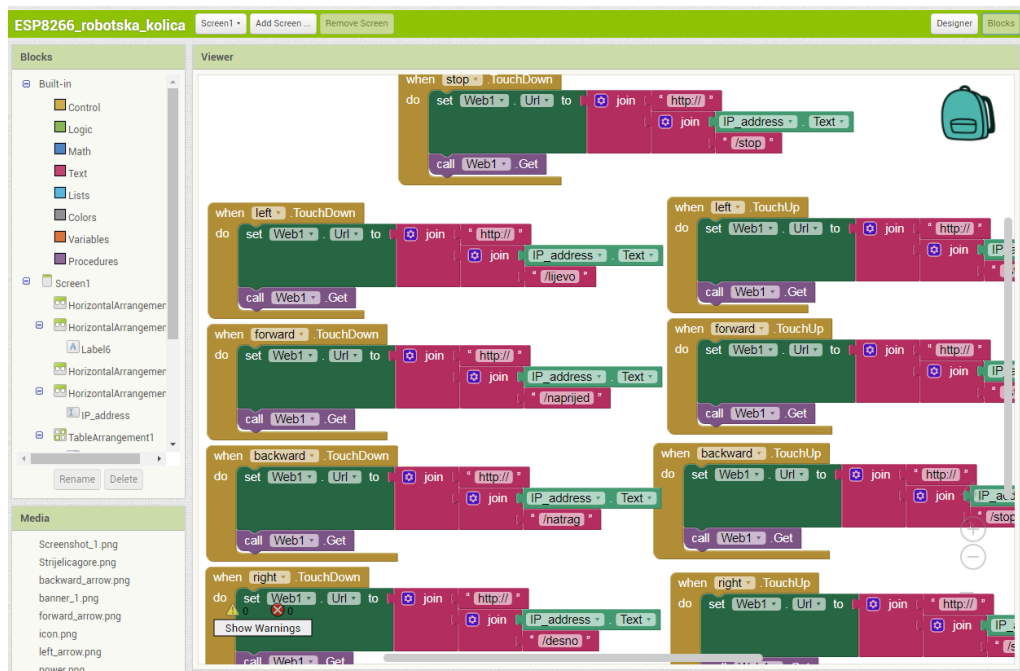
### 3.8. Izrada aplikacije

Za izradu aplikacije koristio sam MIT App Inventor 2 koji ima odlično sučelje za ljude koji nemaju mnogo iskustva s kreiranjem mobilnih aplikacija ili programiranjem općenito. Sučelje možemo vidjeti na slici ispod.



Slika 28. MIT App Inventor 2 sučelje

Nakon određivanja izgleda aplikacije, potrebno je pojedinim gumbima dati funkcije da bi ona pravilno radila. To se odrađuje u „Blocks“ dijelu MIT App Inventora. U ovome dijelu izrade aplikacije povezujete gumbe s njihovim funkcijama koristeći drag and drop način. Primjerice: Kada je gumb „forward“ pritisnut, on obavlja funkciju naprijed preko IP adrese ESP-a. Ukoliko gumb forward nije pritisnut, robotska kolica stoje na mjestu. Isto tako za gumbe lijevo, desno te natrag.



Slika 29. Izgled blokova funkcija

### 3.9. Povezivanje aplikacije s ESP8266

Unošenjem IP adrese u aplikaciju i pritiskom na određeni gumb, obavljamo određenu funkciju. Potrebno je primiti podatak s aplikacije te prosljediti naredbu do funkcije unutar ESP-a. Kod se nalazi na priloženoj slici.

```

if (s == "naprijed"){
    naprijed();
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("");
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    client.println("naprijed");
    client.println("</html>");
    client.stop();
    delay(1);
}
else if (s == "natrag"){
    natrag();
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("");
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    client.println("natrag");
    client.println("</html>");
    client.stop();
    delay(1);
}

```

Slika 30. Dio koda koji odgovara na naredbu aplikacije



```

else if (s == "lijevo"){
    lijevo();
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("");
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    client.println("lijevo");
    client.println("</html>");
    client.stop();
    delay(1);
}
else if (s == "desno"){
    desno();
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("");
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    client.println("desno");
    client.println("</html>");
    client.stop();
    delay(1);
}
else if (s == "stop"){
    stop_all_now();
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("");
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    client.println("stop");
    client.println("</html>");
    client.stop();
    delay(1);
}
}

```

*Slika 31. Dio koda koji odgovara na naredbu aplikacije*

### 3.10. Izrada web stranice

Kao što smo rekli u jednom od prethodnih odlomaka, ESP ne može istovremeno imati AP i biti spojen na neku WiFi mrežu. Stoga smo odvojeno napravili jednostavnu html stranicu koja se osvježi svake dvije sekunde i prikazuje vrijednosti svih senzora spojenih na ESP8266. Kontrola robota i očitavanje vrijednosti senzora ne može biti realizirano skupa osim da imamo bluetooth modul kojim bi slali podatke. Kod za izradu stranice možemo vidjeti na slici ispod.

```

void handleRoot() {
LDRReading = analogRead(LDRPin);
udaljenost = ultrasonic.distanceRead();
float h = dht.readHumidity();
float t = dht.readTemperature();
String html = "<!DOCTYPE html> <html> <head><meta na
html += LDRReading;
html += "</div><h1>Udaljenost(cm)</h1><div id=\"coun
html += udaljenost;
html += "</div><h1>Temperatura(°C)</h1><div id=\"cou
html += t;
html += "</div><h1>Vlaznost(%)</h1><div id=\"countNu
html += h;
html += "</div> </div> <div id=\"updateView\"> <label

    server.send(200, "text/html", html);
}

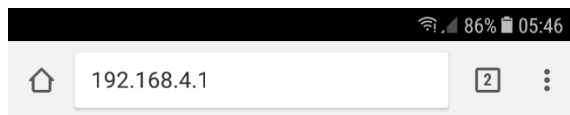
void getData() {

    LDRReading = analogRead(LDRPin);
    udaljenost = ultrasonic.distanceRead();
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    String text2 = "{\"data\":[";
    text2 += "{\"dataValue\":\":";
    text2 += LDRReading;
    text2 += "\",\"";
    text2 += "{\"dataValue\":\":";
    text2 += t;
    text2 += "\",\"";
    text2 += "{\"dataValue\":\":";
    text2 += h;
    text2 += "\",\"";
    text2 += "{\"dataValue\":\":";
    text2 += udaljenost;
    text2 += "\",\"";
    text2 += "]}";
    server.send(200, "text/html", text2);
}

```

*Slika 32. Izrada HTML stranice*

Izgled web stranice možemo vidjeti na slici koja je uslikana mobitelom jer stranici mogu pristupiti samo osobe lokalno spojene na WiFi mrežu(AP) ESP-a. Stranica se osvježava svake dvije sekunde.



Senzori

## Svjetlina

193

## Udaljenost(cm)

16

## Temperatura(°C)

22.10

## Vlaznost(%)

70.00

*Slika 33. Izgled HTML stranice*

## 4. TESTIRANJE I REZULTATI

### 4.1.Senzori

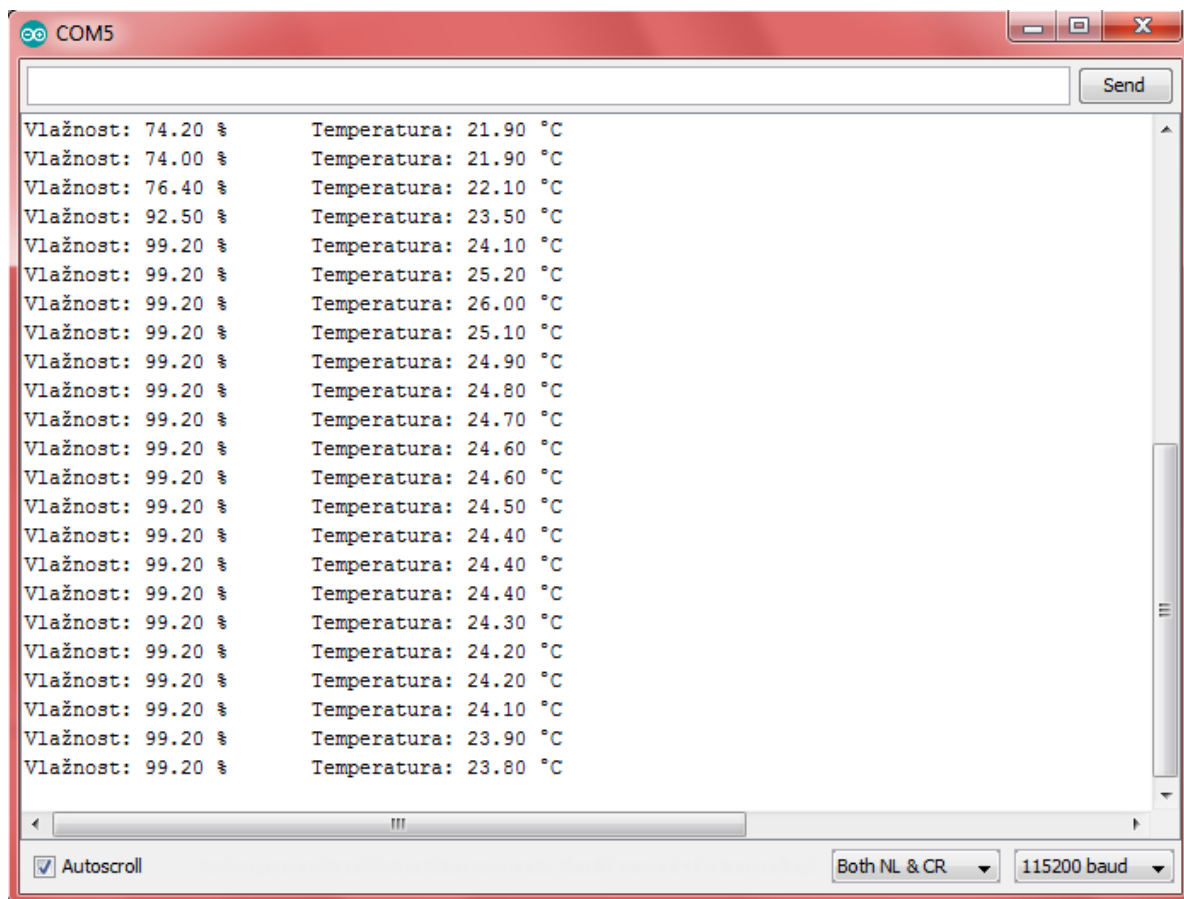
#### 4.1.1. DHT22

Senzor vlage i temperature DHT22 uspoređen je s *Auriol temperature station* digitalnim termometrom da bi vidjeli kolika je razlika između ta dva očitavanja temperatura te da bi na temelju toga mogli izračunati i vidjeti pogrešku senzora. Na slici ispod vidimo očitavanje *Auriol* digitalnog termometra (gornja vrijednost je temperatura kod senzora koji se nalazio kod robotskih kolica, dok je donja vrijednost u samom uređaju).



*Slika 34. Auriol temperature station*

Sa slike je vidljivo da je očitana temperatura u sobi 21.2 °C. Usporedimo to s DHT22 senzorskim očitanjem na slici ispod.



Slika 35. DHT22 senzorska očitavanja

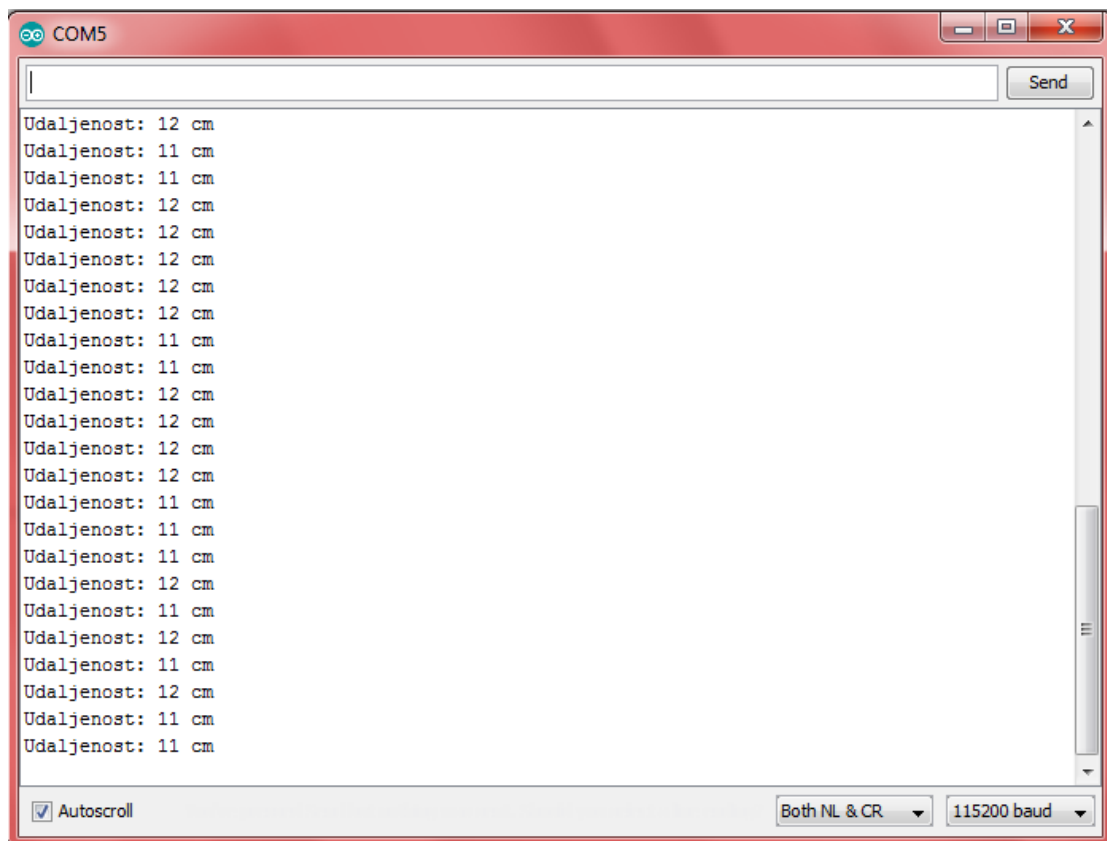
Sa slike je vidljivo da je početna temperatura bila 21.9 °C, te sam, da bi provjerio radi li ispravno senzor, dahnuo u senzor te mu time povećao vlagu i temperaturu. Senzor šalje vrijednost temperature i vlage svake dvije sekunde. Vidimo da je maksimalna temperatura koju je senzor postigao 26 °C i da nakon toga pada, no vlaga ne opada tako brzo. Iz toga zaključujemo da jeza mjerenje točne vlage potrebno pričekati određeno vrijeme ne bi li dobili što precizniji rezultat. U specifikacijama za senzor temperature piše da je točnost očitavanja temperature  $\pm 0.5$  °C što skoro odgovara našem primjeru. Međutim, moramo uzeti u obzir da i referentni primjerak digitalnog termometra također ima određeno odstupanje pa možemo reći da je DHT22 senzor poprilično točan što se tiče očitavanja temperature, dok za vlagu nismo imali referentni uređaj za usporediti.

#### 4.1.2. HCSR-04

Senzor je postavljen na 11 cm razmaka od svoje prepreke, što je prikazano na slici ispod.



*Slika 36. HCSR-04 senzor na udaljenosti od 11cm*

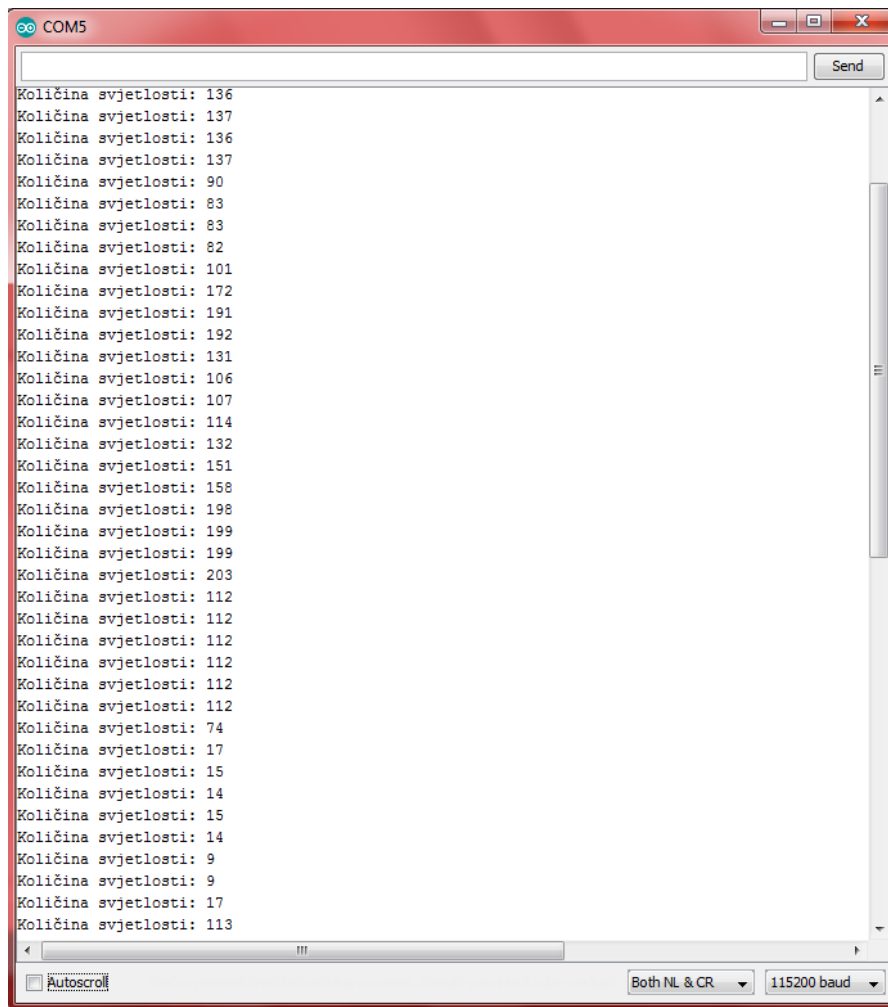


### *Slika 37. Mjerenje HCSR-04 senzora udaljenosti*

Sa slike možemo vidjeti da je senzor mijenjao vrijednosti između 12 i 11 cm te bi mogli na temelju toga naslutiti da senzor nije pouzdan. Međutim, kod je definiran za izražavanje udaljenosti u centimetrima, a ne u milimetrima što bi bilo puno točnije, no radi ovog projekta to nije potrebno. Također, na slici gdje je stiropor udaljen od senzora 11 cm, vidimo da nije točno 11 cm, već je malo više(zbog vidljivosti broja jedanaest) pa razumijemo zašto senzor vraća vrijednosti 11 i 12. Preciznost HCSR-04 bi trebala biti 3mm, što(u neku ruku) možemo potvrditi ovim testiranjem.

#### **4.1.3. LM393**

Senzor količine svjetlosti testiran je preko serial monitora. Na senzoru se nalazi komparator pomoću kojeg računamo količinu svjetlosti koja dolazi na senzor. Spojen je na analogni pin pa vraća vrijednosti između 9 i 203. Što veća vrijednost, manje je svjetla i obrnuto. U ovom je primjeru korištena svjetiljka za osvjetljavanje te prikriivanje senzora rukom.

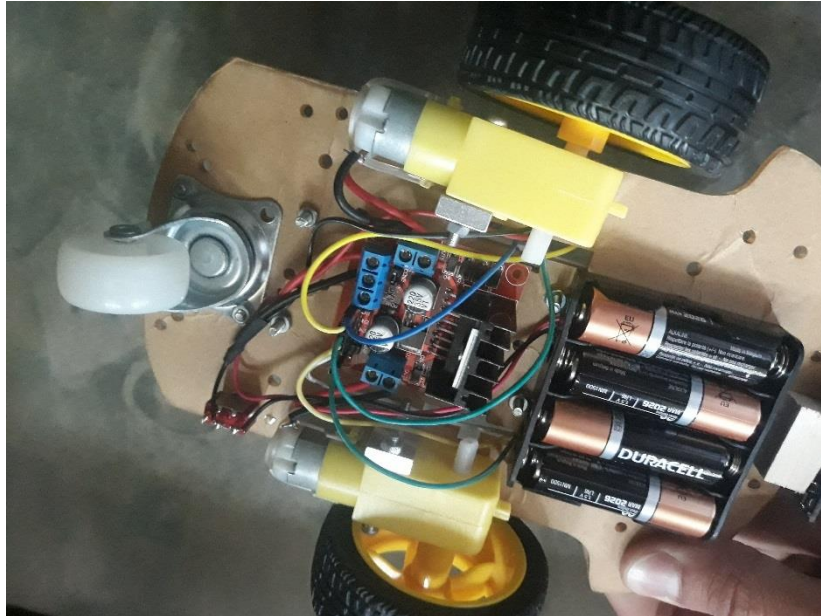


Slika 38. LM393 mjerenje svjetlosti

## 4.2. Robotska kolica

Testiranje robotskih kolica odvojeno je od senzora, jer kada ispunimo uvjete za senzore u glavnoj *loop* funkciji, gdje nam DHT senzor treba 1-2 sekunde delay, automatski utječemo i na odaziv naredbe s aplikacije. Točnije, robotska kolica izvršavaju naredbu koja im je poslana te, prije nego što mogu izvršiti novu naredbu, mora proći dvije sekunde. To znači da ako npr. pritisnemo naredbu *naprijed*, robotska kolica će se dvije sekunde kretati naprijed prije nego stanu. To je upravo zbog tog delaya.





*Slika 39. Sustav mobilnosti robotskih kolica*

Kretanje robota nije savršeno. Kao prvo, pri kretanju naprijed te naglom zaustavljanju, robot se propne zbog male težine(isto tako naglo kretanje prema natrag), pa bi dobro bilo postaviti nekakav mali olovni uteg na kraj robota koji se diže u zrak. Nadalje, robot ne ide „ravno naprijed“ niti „ravno natrag“ kada mu damo naredbu da ide naprijed/natrag. Ovisno o poziciji pomoćnog(okretnog) točkića robot je na udaljenost do 1m pokazivao greške i do 20cm u stranu, dok je najmanja pogreška bila samo 5cm Također, funkcije robota su programirane da budu samo naprijed, nazad, lijevo, desno te nije ukomponirana funkcija da robot istovremeno ide naprijed te da možemo skrenuti lijevo ili desno. Kao što vidimo, robot se još jako puno može nadograditi, no glavna funkcija je obavljena.

## **5. ZAKLJUČAK**

S obzirom da tehnologija u današnje vrijeme iznimno brzo napreduje i sve smo više povezani internetom, ljudi si olakšavaju i ubrzavaju pojedine poslove i zadatke.

Osnovno znanje iz programiranja dovoljno je za programiranje u Arduino IDE-u koje je poprilično jednostavno i samoobjašnjivo programsko okruženje, a prednost je također i veličina zajednice koja je svakim danom sve veća. ESP8266 omogućuje nam povezivanje na internet, a to nam omogućuje IoT(Internet of Things). To je pojam koji označava fizičke, opipljive

elektroničke naprave koje su povezane na internet. IoT je vrlo popularan i postaje sve popularniji.

Jednostavnost programiranja omogućuje ljudima osmišljavanje raznih primjena kako Arduina tako i ESP8266 modula. Primjerice možete napraviti senzor pokreta pa da vam pošalje mail čim netko aktivira taj senzor. Također možete napraviti malu meteorološku postaju gdje vam senzor vraća vrijednost vlage i temperature, a vi si to prosljedite ili na mobitel ili na nekakav display. Možete upravljati vašim domom s mobitela. Podignite rolete, prigušite svjetlo ili upalite klimu samo pritiskom na gumb na mobitelu.

Znanje stečeno tijekom izrade ovog završnog rada vrlo je korisno i lako se može primijeniti na rješavanje sličnih problema ili općenito komunikacijom između hardvera i softvera. Što se tiče poboljšanja aplikacije, mogla bi se web stranica napraviti u CSS-u da izgleda ljepše te bi se žice koje se nalaze na robotskim kolicima mogle zaštititi nekakvom kutijom.

## **LITERATURA**

[1] Wikipedia, Arduino

<https://en.wikipedia.org/wiki/Arduino> , pristup: 23.06.2017.

[2] Arduino, Arduino Nano

<https://store.arduino.cc/arduino-nano> , pristup: 23.06.2017.

[3] E-radionica, Croduino Nova

<https://e-radionica.com/hr/croduino-nova.html> , pristup: 23.06.2017.

[4] Electrodragon, ESP8266 ESP-12F

[http://www.electrodragon.com/w/ESP-12F\\_ESP8266\\_Wifi\\_Board#Product\\_Brief](http://www.electrodragon.com/w/ESP-12F_ESP8266_Wifi_Board#Product_Brief) , pristup: 24.06.2017.

[5] Howtomechatronics, HC-SR04

<http://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/> , pristup: 24.06.2017.

[6] E-radionica, HC-SR04

<https://e-radionica.com/hr/ultrazvucni-modul-hc-sr04.html> , pristup: 24.06.2017.

[7] Electroschematics, DHT22

<http://www.electroschematics.com/11293/am2302-dht22-datasheet/> , pristup: 24.06.2017.

[8] E-radionica, LM393 breakout

<https://e-radionica.com/hr/senzor-svjetlosti-s-lm393-e-radionica-made.html> , pristup: 24.06.2017.

[9] E-radionica, Dual H-most L298N

<https://e-radionica.com/hr/dc-motor-driver-dual-h-bridge-l298n.html> , pristup: 24.06.2017.

[10] ESP8266 protokoli

<https://tttapa.github.io/ESP8266/Chap05%20-%20Network%20Protocols.html> , pristup: 22.9.2017.

## SAŽETAK

**Naslov:** Primjena ESP8266 komunikacijskog modula u udaljenom upravljanju

Cilj ovog završnog rada je izraditi robotska kolica kojima će se upravljati preko mobilne aplikacije koja se povezuje s ESP8266 komunikacijskim modulom. Na početku rada ukratko su opisane teorijske podloge i sustavi u kojima se koristi ESP8266. Također su opisani i drugi sustavi temeljeni na bežičnoj mreži. Proučeni su nacrti i načini spajanja pojedinih senzora i

komponentata te su priključeni na Croduino(mikrokontroler). Nadalje, programiran je ESP8266 da bi mogao biti povezan na internet te je napravljena mobilna aplikacija korištenjem MIT App Inventora. Pokretanjem aplikacije i unošenjem IP adrese koju generira ESP8266 komunikacijski modul, možemo upravljati robotskim kolicima pritiskom na određeni gumb(naprijed, nazad, lijevo, desno). Također se ESP8266 može isprogramirati da ima svoj access point(AP) te se na njemu može napraviti jednostavna HTML stranica koja može prikazivati razne podatke.

**Ključne riječi:** Arduino, Croduino, ESP8266, robot, IoT

## **ABSTRACT**

**Title:** Application of ESP8266 communication module in remote control

The goal of this project was to make a robot which will be controlled remotely with a mobile application which is connected to the ESP8266 communication module. At the beginning of the project, theoretical background and systems in which ESP8266 is used are briefly described. Also, systems which use a WiFi connection are described. The sketches and ways of connecting individual sensors and components were studied and connected to Croduino(microkontroler).

Furthermore, ESP8266 was programmed to be connected to the Internet using WiFi and a mobile application was created using the MIT App Inventor. By running the application and by entering the IP address generated by the ESP8266 communications module, we can manage a robot by pressing a certain button(forward, reverse, left, right). The ESP8266 can also be programmed to have its own access point and we can then create a simple HTML page that can display various data.

**Ključne riječi:** Arduino, Croduino, ESP8266, robot, IoT

## **ŽIVOTOPIS**

Benjamin Puljko rođen je 30. rujna 1995. godine u Osijeku. Stanuje u Veliškovcima na adresi Zrinsko-Frankopanska 8. Godine 2002. započinje osnovnoškolsko obrazovanje u OŠ Ivana Kukuljevića Veliškovci, gdje se iskazuje na natjecanjima iz matematike i fizike. Nakon završene osnovne škole, upisuje opću gimnaziju u Valpovu, gdje tijekom srednjoškolskog obrazovanja također sudjeluje na brojnim natjecanjima iz matematike, fizike, biologije i kemije. Prvu, drugu i

čtvrtu godinu srednje škole završava s odličnim, dok treću završava s vrlo dobrim uspjehom.  
2014. godine upisuje sveučilišni studij računarstva na Elektrotehničkom fakultetu u Osijeku.