

SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I
BRODOGRADNJE

IZVJEŠTAJ
SENZOR POKRETA

Ante Turalija, Anđela Marić

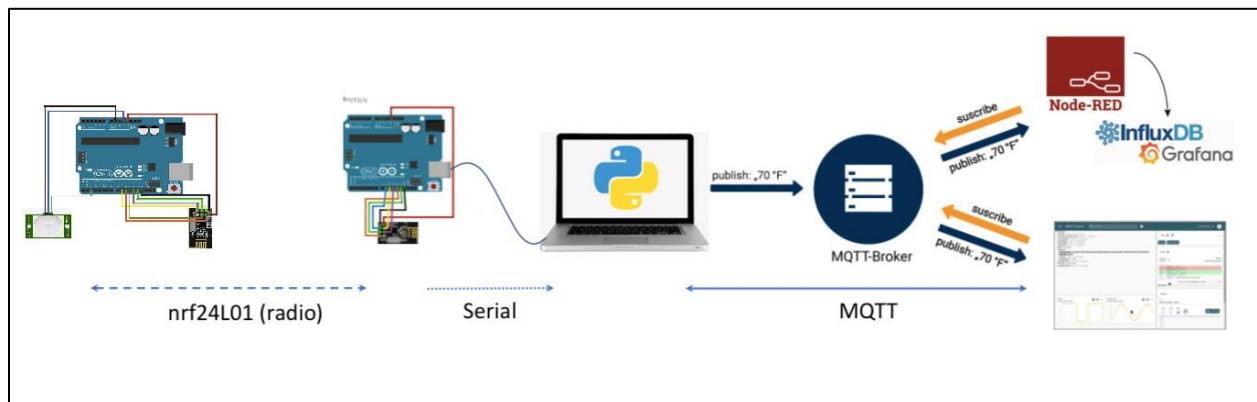
Split, svibanj, 2020

SADRŽAJ:

1. UVOD.....	1
2. KORIŠTENJA OPREMA.....	2
2.1. Arduino Uno	2
2.2. Nordic nRF24L01+	4
2.3. Senzor pokreta	5
3. PREDAJNIK	6
4. PRIJAMNIK	10
5. MQTT KOMUNIKACIJA I GRAFANA	13
6. RAZULTATI I ZAKLJUČCI.....	14
7. TESTIRANJE nRF24L01+	17
8. LITERATURA	18
9. SAŽETAK	19

1. UVOD

U ovom projektu realizirana je senzorska tehnologijom za detekciju pokreta. Povezivanjem senzora i mikrokontrolera te korištenjem nRF24L01+ radio modula stvoren je jedan funkcionalni paket prikladan za uporabu. Senzor pokreta generira podatke koji se prikupljaju na mikrokontroleru koji služi kao predajnik. Kao što je vidljivo na slici, koja prikazuje arhitekturu cijelog sustava, predajnik putem radio modula šalje podatke mikrokontroleru koji služi kao prijamnik. Pokretanjem python skripte podaci se putem MQTT brokera šalju na server. Također je korišten Docker, aplikacija koja upotrebljava sljedeće servise: Node-RED, InfluxDB, Chronograf i Grafanu. Kreirati smo nekoliko grafova u Grafani koja prikazuje vrijednosti pokreta iz InfluxDB baze podataka. Za ostvarenje ove izvedbe potrebna su dva mikrokontrolera Arduino Uno, PIR senzor pokreta, dva nRF24L01 uređaja te odgovarajuća hardverska i softverska podrška.

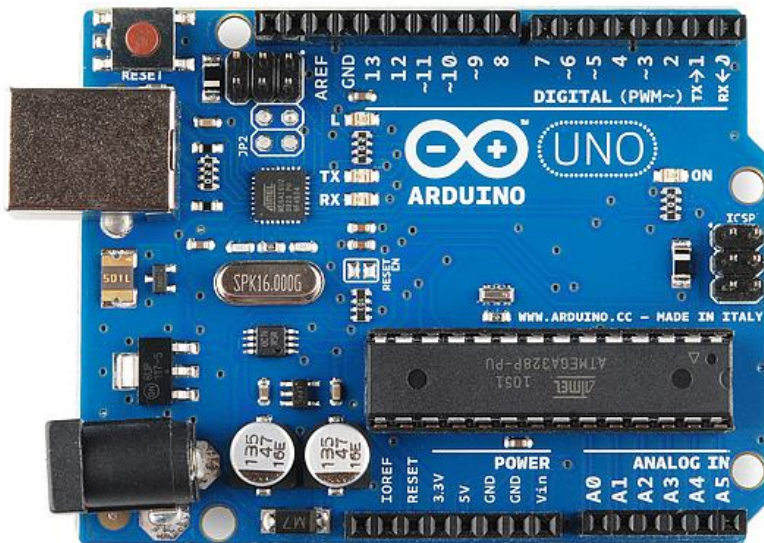


Arhitektura cijelog sustava

2. KORIŠTENA OPREMA

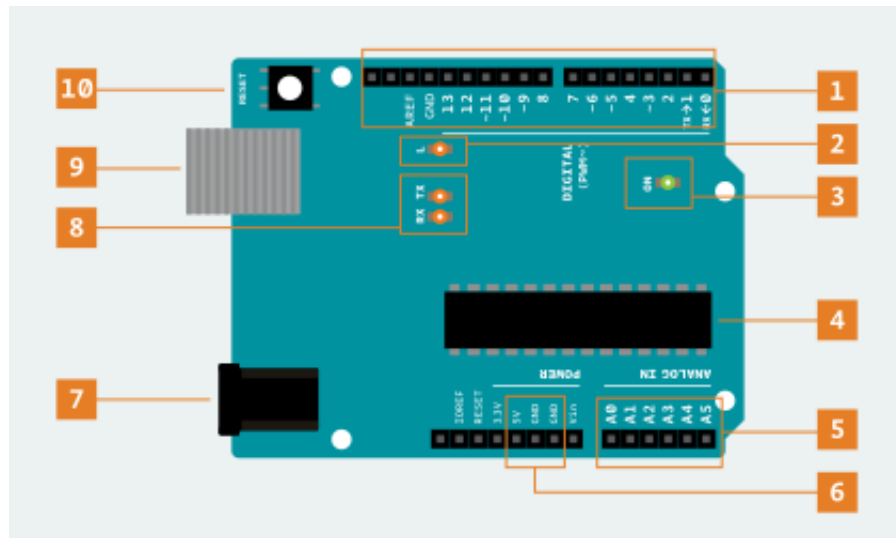
2.1 Arduino Uno

Arduino mikrokontroler je open source platforma, koja omogućava spajanje elektronskih komponenti i senzora te upravljanje s njima pomoću programa upload-ovanog na Arduino ploču putem USB kabela.



Slika 2.1. Uređaj Arduino Uno

Na Arduino je moguće spajati senzore, LED, IR diode, releje, display-e, tranzistore, otpornike i ostale aktivne i pasivne elektronske komponente i tako kreirati interaktivne hardware-software kreacije. Nakon završenog kreiranja i upload-a programa nije potrebna veza Arduino ploče i računala za programiranje. Napajanje je moguće vršiti i putem USB kabela sa USB porta računala ili iz adaptera za USB izlazom napona 5V.

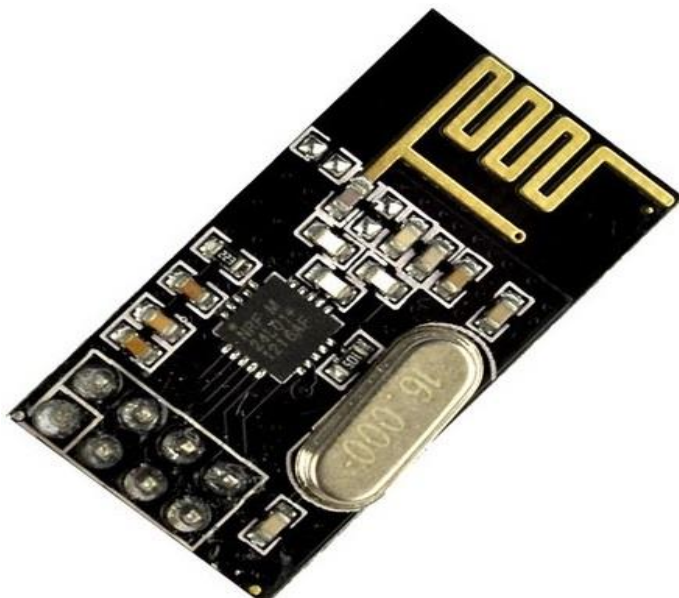


Slika 2.2. Prikaz osnovnih dijelova Arduino ploče

1. Digitalni pinovi. Digitalne pinove koriste funkcije kao što su: `digitalRead()`, `digitalWrite()`,
2. `analogRead()` i `analogWrite()`.
3. LED na pinu 13.
4. LED indikator napajanja. Indikator postojanja napajanja ploče.
5. ATmega mikrokontroler. Mozak Arduino ploče.
6. Analogni ulazi. Na ovim pinovima koristiti funkciju `analogRead()`.
7. GND i 5V pinovi. Ove pinove koristimo za osiguravanje +5V i mase za uređaje koje spajamo na Arduino ploču.
8. Konektor za napajanje preko kojega se Arduino ploča napaja kada nije priključena na USB port. Prihvaća napon od 7V do 12V (plus u sredini konektora).
9. TX (slanje) i RX (primanje) LED diode. Služe za indicaciju komunikacije između Arduino ploče i računala.
10. USB port. Koristi se za napajanje Arduino ploče, i za serijsku komunikaciju pomoću `println()` funkcije.
11. Reset tipka. Reset Atmega mikrokontrolera

2.2. Nordic nRF24L01+

Nordic nRF24L01+ je primopredajnik kojega karakterizira niska potrošnja (eng. *Ultra Low Power* - *ULP*). Maksimalna brzina komunikacije podataka koja se može postići je do 2Mbps na frekvenciji od 2,4GHz ISM (eng. *Industrial, scientific and medical*). Ovi primopredajnici koriste 2,4 GHz nelicencirani pojas poput mnogih WiFi routera, Bluetooth, nekih bežičnih telefona i slično čiji je raspon od 2,400 do 2,525 GHz. Širina nRF24L01 kanala je 1 MHz što ukupno omogućava komunikaciju na 125 nepreklapajućih kanala (0-124).



Slika 2.2. Nordic nRF24L01

Nekoliko važnih karakteristika:

- Napajanje: 1.9 - 3.6 V
- Radni napon ulaza IO: 0 - 3.3 V
- Brzina prijenosa: +7 dB
- Osjetljivost pri prijemu: <90 dB
- Domet prijenosa: 250 m na otvorenom
- Dimenzije: 15 x 29 mm

2.3. Senzor pokreta

Pasivni infracrveni senzor (PIR senzor) je izrađen od piroelektričnog senzora koji mjeri infracrveno svjetlo koje zrači iz objekata u vidnom polju. Svi objekti s temperaturom iznad apsolutne nule emitiraju toplinsku energiju u obliku zračenja. Budući da ga odlikuju multifunkcionalnost i niska cijena, prihvaćen je u mnoštvu projekata te od strane hardverske zajednice otvorenog koda za projekte povezane s Arduinoom ili nekom drugom platformom



Slika 2.3. PIR senzor

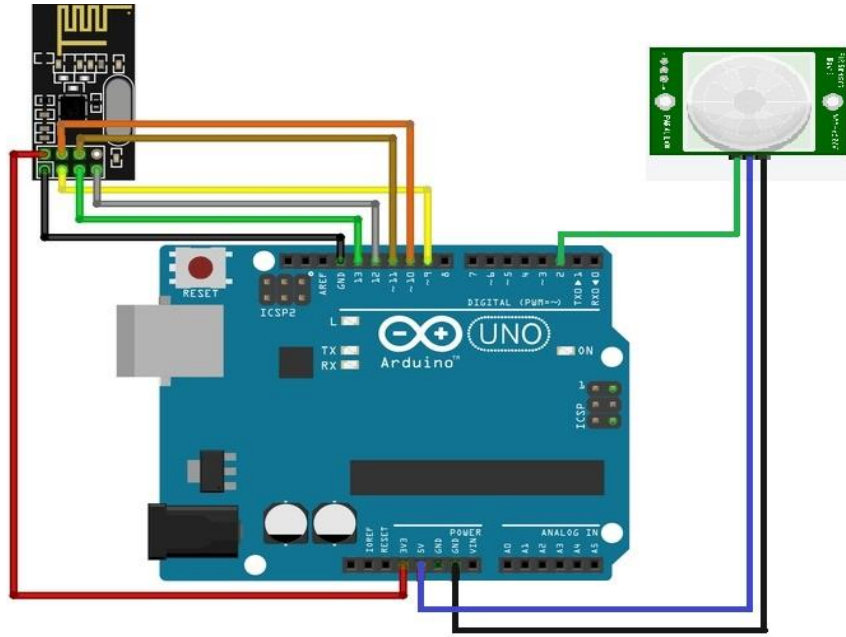
PIR senzori detektiraju opće kretanje, ali ne daju podatke tko se ili što kreta. PIR senzori se obično nazivaju jednostavno "PIR", ili ponekad "PID", za "pasivni infracrveni detektor". Pojam pasivni odnosi se na činjenicu da PIR uređaji ne zrače energiju za potrebe detekcije. Djeluju u cijelosti otkrivanjem infracrvenog zračenja (zračenja topline) koje emitiraju ili odbijaju objekti. Radi pri naponu od 3.3 V.

3. PREDAJNIK

Predajnik je uređaj čiji je uloga prikupljanje podataka, obrada te na kraju slanje tih podataka drugom uređaju (u ovom slučaju drugi uređaj je također Arduino Uno). Dva ili više predajnika ne mogu istovremeno slati poruku na istom kanalu jer će doći do ometanja, te poruka neće biti poslana.



Slika 3.1. Predajnik



Slika 3.2. Način spajanja predajnika

U ovom projektu realizirali smo da se predajnik nalazi u stanju spavanja dok je senzor pokreta spojen na napajanje. U trenutku kada se detektira pokret, senzor trigira interrupt te budi Arduino koji šalje poruku prijamniku. Istu stvar smo napravili i za nRF24L01 uređaj. Svrha ovoga je naravno bila ušteda energije kod baterijskog napajanja.

Funkcija Hibernate:

```
void SENSORS::Hibernate()
{
    attachInterrupt(PIRsensorInterrupt,wakeUpNow, CHANGE);

    delay(50);
    LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF);
    delay(50);
    detachInterrupt(PIRsensorInterrupt);
}
```

RF powerDown i RF powerUp:

```
void RADIO::RF_powerDown()  
{  
    radio.powerDown();  
}  
void RADIO::RF_powerUp()  
{  
    radio.powerUp();  
}
```

U ovom projektu također je izrađena State Machine na način da je prvo stvorena enumeracija svakog stanja mašine korištenjem typedef enumeracije. U ovom projektu postoje tri stanja:

```
typedef enum  
{  
    READ_SERIAL,  
    RADIO_TX,  
    SLEEP,  
}
```

Od ova tri stanja dva se se naizmjenično pokreću, a to su RADIO TX i SLEEP, te je na taj način arduino isključen sve dok se ne detektira pokret što trigira interrupt i budi arduino. Na ovaj način se ostvaruje ušteda energije.

State machine

```
void loop()
{
  switch (state)
  {
    case READ_SERIAL:

      state = SLEEP;
      break;

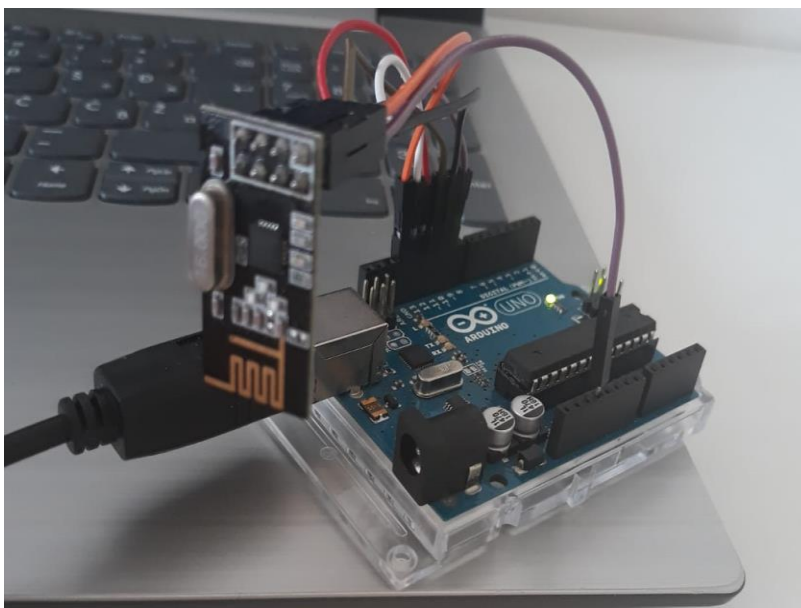
    case SLEEP:
      Serial.println(F("spavanje"));
      delay(100);
      radioNRF.RF_powerDown();
      sensor.Hibernate();
      radioNRF.RF_powerUp();
      dataToSend.pokret_rukom=1;
      delay(100);
      Serial.println(F("buđenje"));
      state=RADIO_TX;
      break;

    case RADIO_TX:
      Serial.println(F("slanje"));
      Serial.println(dataToSend.pokret_rukom);
      delay(100);
      radioNRF.RF_send(dataToSend);
      state = SLEEP;
      break;

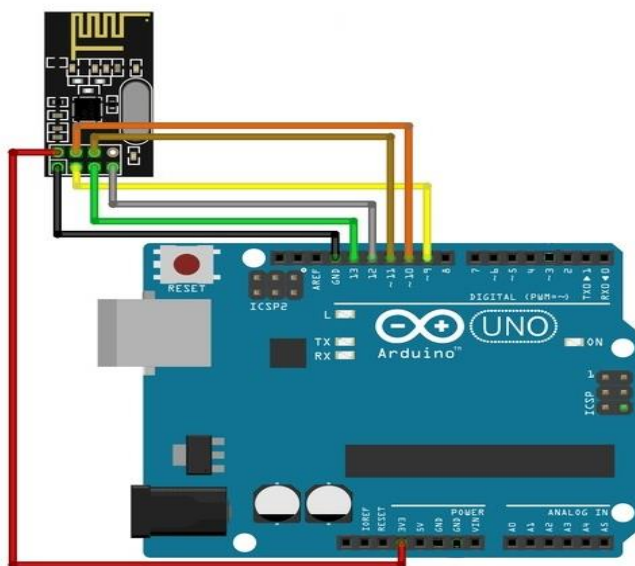
  }
}
```

4. PRIJAMNIK

Uloga prijamnika je da stalno osluškuje i iščekuje poruku. Kada poruka stigne do prijamnika, pokretanjem python skripte putem MQTT brokera se ispisuje na server. U našem slučaju poruka koju dobiva prijamnik je 0 ili 1, ovisno je li se dogodio pokret ili ne.



Slika 4.1. Prijamnik



Slika 4.2. Način spajanja prijamnika

PYTHON je programski jezik opće namjene, interpretiran i visoke razine koji je stvoren 1990. godine. Python dopušta programerima korištenje nekoliko stilova programiranja. Objektno orijentirano, strukturno i aspektno orijentirano programiranje stilovi su dopušteni korištenjem Pythona te ova fleksibilnost čini Python programski jezik sve popularnijim.

U našem projektu python skriptu smo iskoristili za prijenos podataka sa prijavnika na mqtt server, odnosno za uspostavu „online“ komunikacije.

Python skripta

```
import serial
import paho.mqtt.client as mqtt

BROKER = "mqtt.eclipse.org"
CLIENTID = "MQTTEExample"
TOPIC_ONE = "A507/sensors/atural/pokret"

COMPORT = "COM5"
QOS = 1

import time

flag_connected = 0

def on_connect(client, userdata, flags, rc):
    logging.debug("Connected result code "+str(rc))
    client.loop_stop()

def on_disconnect(client, userdata, rc):
    logging.debug("Disconnected result code "+str(rc))
    client.loop_stop()

def on_publish(client,userdata,result):
    print("data published \n")
    pass

mqttc = mqtt.Client(CLIENTID)
mqttc.on_connect = on_connect
mqttc.on_disconnect = on_disconnect
mqttc.on_publish = on_publish
mqttc.connect(BROKER)
mqttc.loop_start()
```

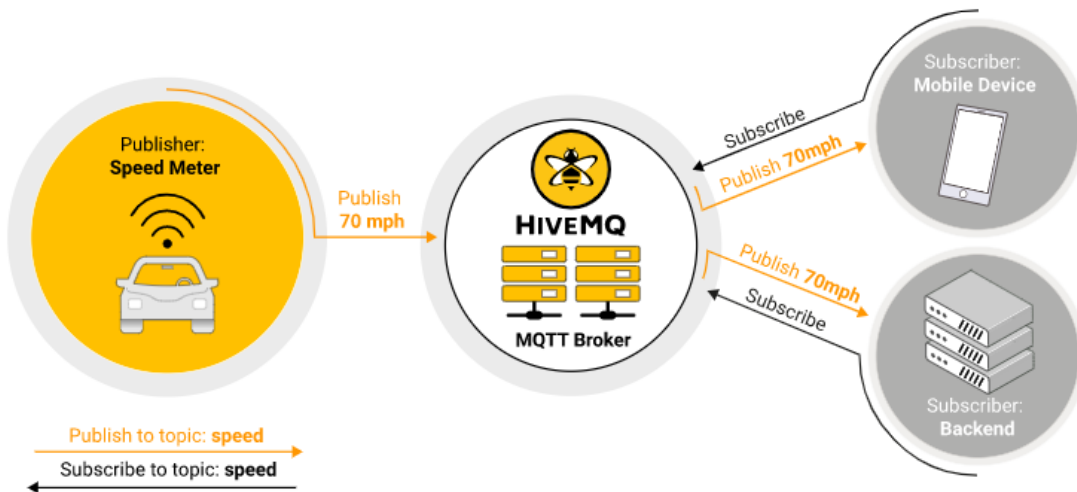
```

ser = serial.Serial(COMPORT, 115200, timeout=5)
while True:
    message = ser.readline()
    print (message)
    if b'P:' in message:
        string, pokret= message.split(b' ')
        pokret, rest = pokret.split(b'!')
        print(pokret.decode('utf-8'))
        mqttc.publish(TOPIC_ONE, payload=pokret.decode('utf-
8'), qos=QOS, retain=False)
        time.sleep(0.01)
mqttc.disconnect()
time.sleep(1)

```

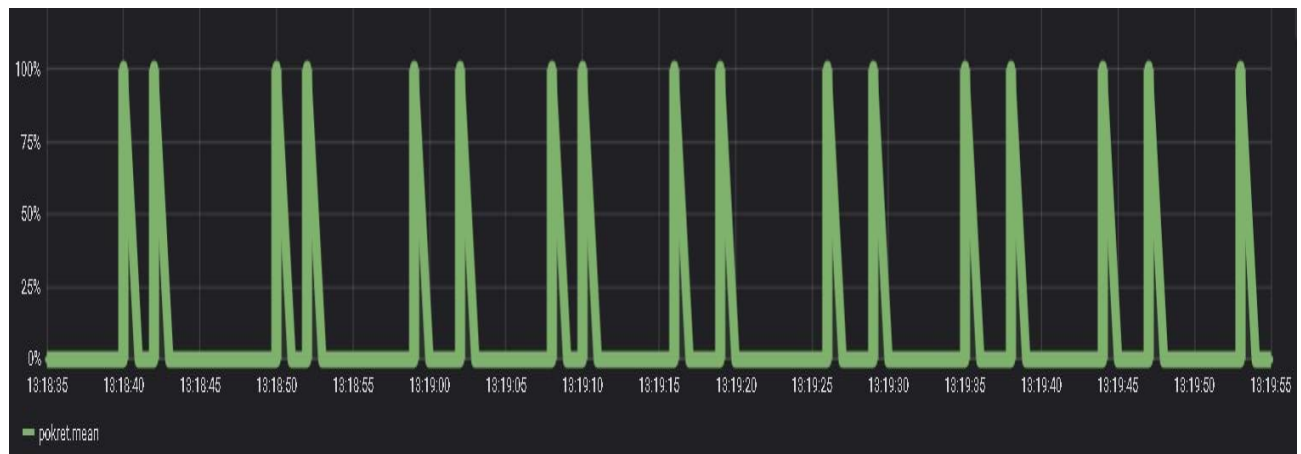
5. MQTT KOMUNIKACIJA I GRAFANA

MQTT je protokol za komunikaciju koji je razvijen u IBMu. Radi na principu objavi/pretplati te je kreiran za kao jednostavan protokol za upotrebu. Dizajniran je kako bi se minimizirala propusnost mreže i zahtjevi resursa uređaja. Jedna od bitnih značajki protokola su osiguranje pouzdanosti i određenog stupnja isporuke poruke. Zbog toga je postao jako rasprostranjen u primjeni i popularan. Najviše se koristi u IoT (eng. *Internet of Things* – Internet stvari) uređajima i M2M (eng. *Machine to Machine*) komunikaciji gdje je potreban protokol koji ne povećava potrošnju energije. Radi na vrhu TCP/IP protokola.

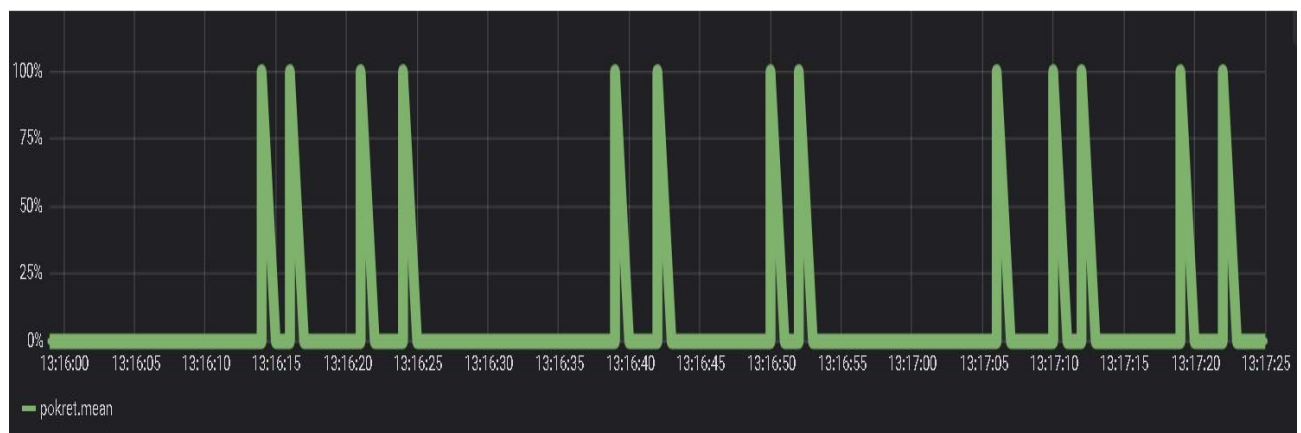


GRAFANA – jedan od servisa koji upotrebljava aplikacija Docker. Uz grafanu postoje i sljedeći servisi: Node-RED, InfluxDB te Chronograf. Napravili smo nekoliko grafova u Grafani koji prikazuju vrijednosti pokreta iz InfluxDB baze podataka.

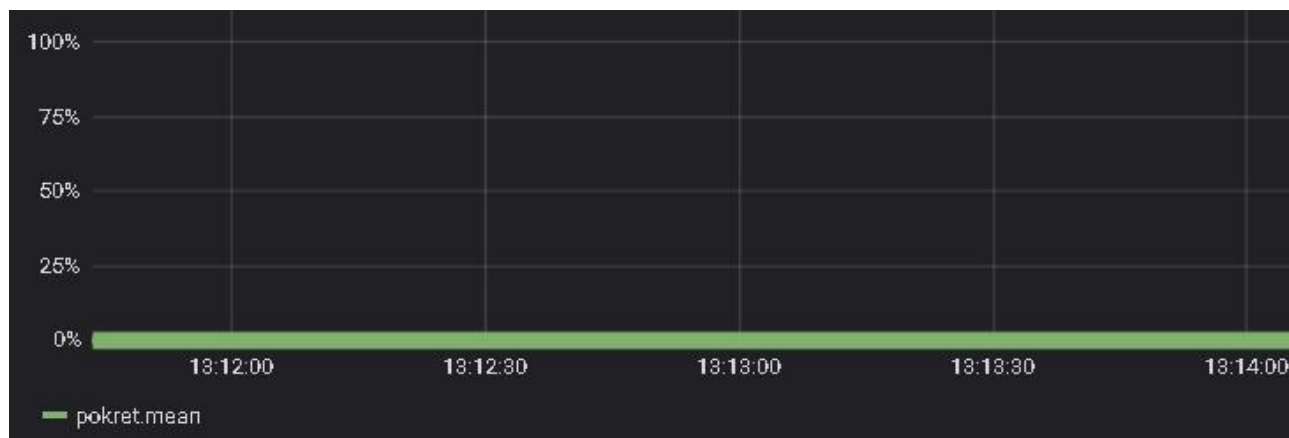
6. RAZULTATI I ZAKLJUČCI



Slika 6.3. Grafički prikaz pokreta osobe koja je udaljena 1 metar od senzora

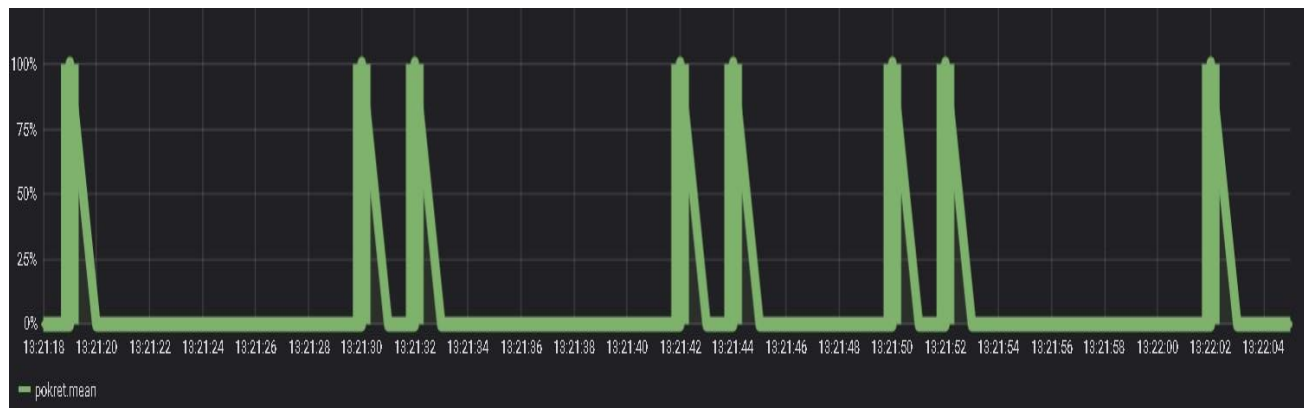


Slika 6.4. Grafički prikaz pokreta osobe koja je udaljena 2 metra od senzora

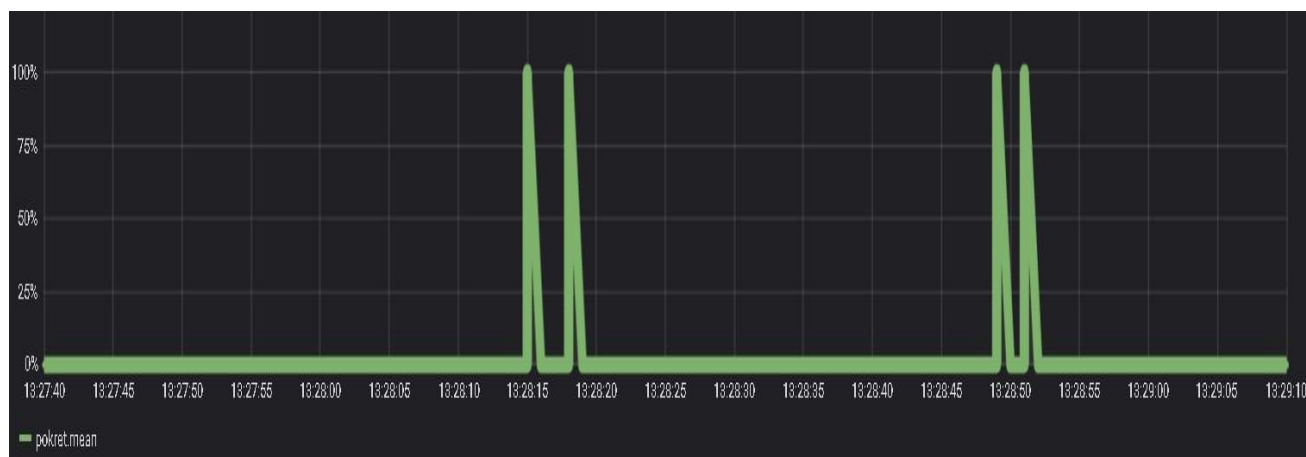


Slika 6.5. Grafički prikaz pokreta osobe koja je udaljena 3 metra od senzora

Promatrajući prethodna tri grafa možemo zaključiti da senzor jednako dobro reagira na pokrete koje su udaljeni od njega 1 metar i 1.5 metara. Dok na pokrete na 3 metra ne reagira uopće. Rezultati su zabilježeni na temperaturi od 23 °C.



Slika 6.6. Grafički prikaz pokreta kojeg generira senzor prilikom bacanje lopte udaljene 2 metra od senzora

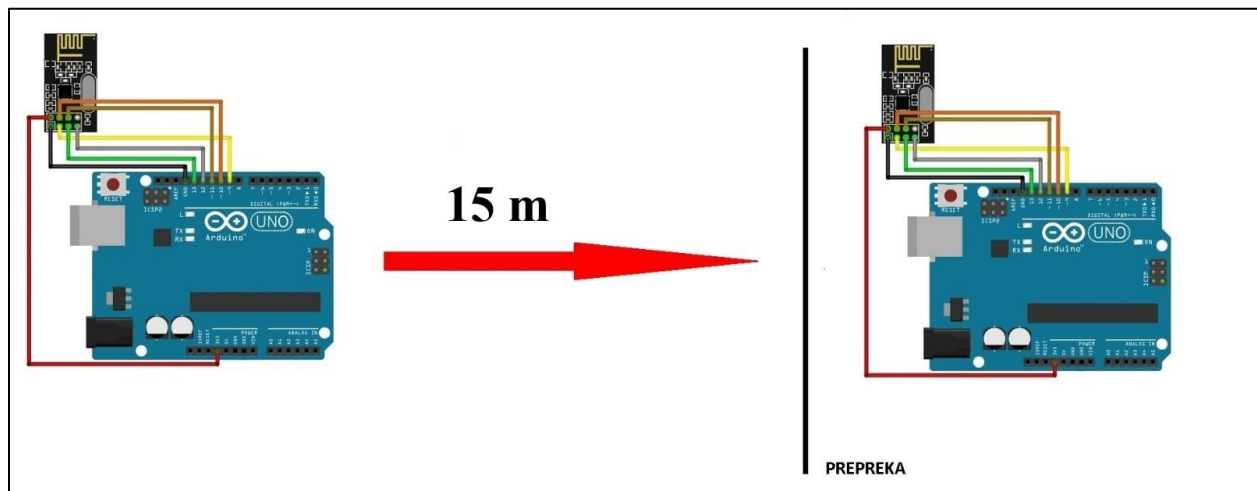


Slika 6.7. Grafički prikaz sporih pokreta koje generira senzor

PIR senzori su male veličine, jeftin, male potrošnje i vrlo lako razumljivi, što je vrlo popularno. Kao i svi senzori imaju dobre i loše strane. Prije uporabe samog senzora morali smo ga prethodno kalibrirati jer nije uopće reagirao na pokret. Kalibrirali smo ga tako što smo podesili viljak za osjetljivost. Viljak za podešavanje vremenskog intervala nismo podešavali. Iz prethodnih grafova možemo zaključiti da senzor jednako dobro reagira na pokrete koji su udaljeni od njega 1 metar i 1.5 metara, dok na pokrete na 3 metra ne reagira uopće. Također na slabe pokrete ne reagira nikako ili jako slabo. Prilikom bacanja lopte na 1.5 metara udaljenosti od senzora očitavaju se pokreti. Senzor nije baš pouzdan jer u određenim trenucima ne reagira, iako bi trebao, ali s obzirom na svoju jako nisku cijenu, usluge koje pruža su na prihvatljivoj razini. Svi rezultati su zabilježeni na temperaturi između 23 °C i 26 °C.

7. TESTIRANJE nRF24L01+

U sklopu ovog projekta također smo testirali pouzdanost radio komunikacije s obzirom na udaljenost predajnika i prijamnika. Test smo proveli na način da smo isprogramirali radio da šalje podatke tijekom određenog vremena, te smo onda pobrojali izgubljene pakete u odnosu na broj poslanih. Prvi test smo napravili na udaljenosti od 15 metara gdje su svi paketi uspješno poslani. Nakon toga smo jedan nrf24l01 odnijeli u drugu prostoriju koja je odvojena zidom debljine 30cm i uočili smo drastično smanjeni broj poslanih pakete. Poslali smo 205 paketa od kojih je samo 58 uspješno stiglo do prijamnika, odnosno 147 je paketa izgubljeno u radio komunikaciji ili ako izrazimo u postotcima 71% paketa nije uspješno stiglo do prijamnika. Ono što možemo zaključiti nakon ovog testa je da radio komunikacija odlično radi kada između uređaja ne postoje nikakva fizička prepreka, te da svaka prepreka ometa slanje podataka. Što je prepreka veća to je neuspješnija komunikacija.



Slika 7.1 Prikaz slanja podataka između dva radio uređaja na udaljenosti od 15 metara sa preprekom

8. LITERATURA

<https://hr.wikipedia.org/wiki/Arduino>

<https://www.arduino.cc/>

http://wiki.sunfounder.cc/index.php?title=NRF24L01_Wireless_Transceiver_Module

https://en.wikipedia.org/wiki/Motion_detector

[https://hr.wikipedia.org/wiki/Python_\(programski_jezik\)](https://hr.wikipedia.org/wiki/Python_(programski_jezik))

9. SAŽETAK

Arduino mikrokontroler je platforma, koja omogućava spajanje elektronskih komponenti i senzora te upravljanje s njima pomoću programa učitano na Arduino ploču putem USB kabela. Arduino platforma je jako jeftina te na vrlo jednostavan način povezuje računala sa fizičkim svijetom. Nordic nRF24L01+ je primopredajnik kojega karakterizira niska potrošnja. Nordic primopredajnici koriste 2,4 GHz nelicencirani pojas poput WiFi routera, Bluetooth-a te nekih bežičnih telefona čiji je raspon od 2,400 do 2,525 GHz. Širina nRF24L01 kanala je 1 MHz što omogućava komunikaciju na 125 nepreklapajućih kanala (0-124). Senzor pokreta je uređaj koji može otkriti kretanje ljudi ili predmeta. PIR senzori se obično nazivaju jednostavno "PIR", ili ponekad "PID", za "pasivni infracrveni detektor. Kako je sposoban pretvoriti gibanje koje osjeti u električne signale, senzor ili emitira podražaje ili dobiva signale iz samog pokretnog predmeta. PIR senzor je male veličine, niske potrošnje, jeftin, jednostavan je za korištenje te odlično reagira na pokrete koji se nalaze unutar dometa do 1.5 metara. Neki nedostaci ovog senzora su nepouzdanost jer se nakon određenog vremenskog intervala sam isključi, nereagiranje na pokret udaljene više od 3 metra, te slabo detektiranje sporih pokreta.

Cjelokupni kod ovoga projekta se može naći na jednom od sljedeća dva linka:

- 1) https://github.com/atural00/izvjestaji_WiSe_2019_20/tree/master/Projekt_senzor%20pokreta
- 2) https://github.com/amaric99/izvjestajiWiSe2019_20/tree/master/projekt_senzor_pokreta