**SVEUČILIŠTE U SPLITU**

**FAKULTET ELEKTROTEHNIKE,**

**STROJARSTVA I BRODOGRADNJE**

**IZVJEŠTAJ**

**Senzor za mjerenje razine vode**

**Bruno Milinović , Ivan Pućo**

Split, Srpanj 2020.

**Sadržaj:**

1.UVOD................................................................................................3

2.KORIŠTENA OPREMA …..............................................................4

2.1. ARDUINO UNO …......................................................................4

2.2. Nordic nRF24L01..........................................................................6

2.3. Senzor VL52L0X..........................................................................8

3. MQTT.............................................................................................10

4. DOCKER........................................................................................11

5. PRAKTIČNA IZVEDBA PROJEKTA I REZULTATI................12

5.1. Transmitter...................................................................................12

5.2. Receiver.......................................................................................15

5.3. Prikaz dobivenih rezultata...........................................................18

**1.Uvod**

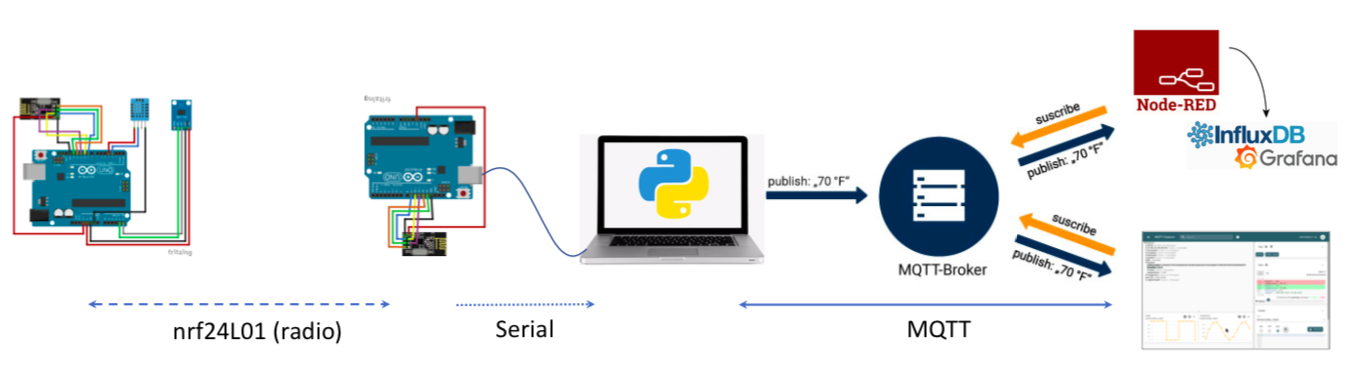
U današnje vrijeme se gospodarstvo ne može zamisliti bez mjerenja. Pravilan odabir može u znatnoj mjeri smanjiti investicijske troškove, povećati pouzdanost proizvodnih procesa i smanjiti troškove montaže i održavanja.

Cilj ovog projekta pod nazivom “Senzor za mjerenje razine vode” je navesti i opisati mjerne metode i uređaje, odnosno senzore koji se koriste pri mjerenju razine vode.

Projekt smo realizirali povezivanjem senzora i mikrokontrolera, te korištenjem nRF24L01+ radio modula. Senzor VL53L0X očitava vrijednosti razine vode (generira podatke),te ih šalje na Arduino UNO (mikrokontroler) koji služi kao predajnik. Predajnik putem radio modula nRF24L01+ šalje podatke mikrokontroleru koji služi kao prijamnik.

Putem MQTT brokera podatci se šalju na server, te zatim pomoću aplikacije docker koristili smo servise kao što su: Node-RED, InfluxDB i Grafana. Na temelju prikupljenih podataka u grafani je kreiran graf koji prikazuje vrijednosti razine vode.

Iako smo za realizaciju ovog projekta za napajanje koristili USB-port,u realnim uvjetima se za napajanje koristi baterija. Najveće ograničenje ovakvih sustava jest zbog ograničenog kapaciteta baterije. Taj problem smo riješili periodičkim paljenjem i gašenjem senzora, odnosno stavljanjem senzora u “sleepmode”.



*Arhitektura cijelog sustava*

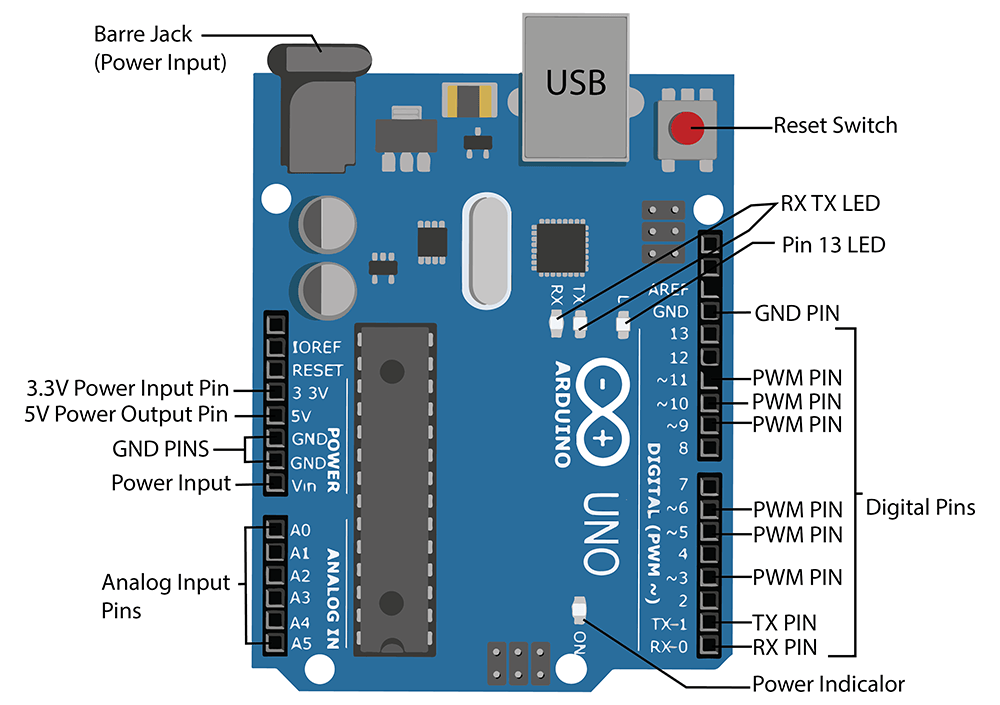
**2. KORIŠTENA OPREMA**

**2.1. Arduino Uno**

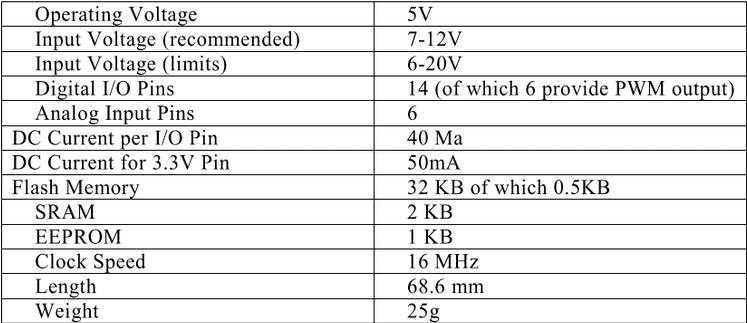
Arduino mikrokontroler je open source platforma, bazirana na “easy-to-use” hardwerskog i softwerskog dijela. Na Arduino je moguće spajati senzore, LED, IR diode, releje, display-e, tranzistore, otpornike i ostale aktivne i pasivne elektronske komponente i tako kreirati interaktivne hardware-software kreacije. Samo programiranje pločice ne zahtjeva previše “programskog znanja” kao kod ostalih mikrokontroskih sustava. Bez obzira na operacijski sustav, arduino uno se može programirat iz Windowsa, Mac-a, Linuxa, ….

Napajanje je moguće vršiti i putem USB kabla sa USB porta računala ili iz adaptera za

USB izlaz napona 5V. Arduino Uno pločica temelji se na mikročipu ATmega328.



*Slika 2.1 Arduino Uno( specifikacija pinova)*



*Slika 2.2 Arduino Uno specifikacije*

**Opće funkcije pinova**:

VIN – Ulazni napona na ploču kada se koristi vanjski napon napajanja.

5 V – Ovaj pin izvodi regularnih 5 V od regulatora na pločici.

3V3- Ovaj pin izvodi regularnih 3,3 V od regulatora na pločici.

GND- uzemljenje.

RESET - Reset Atmega mikrokontrolera

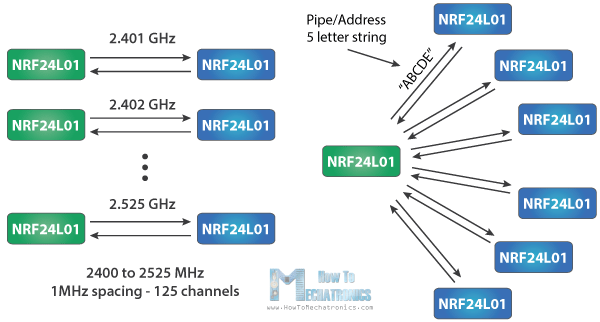
Digitalni Pinovi - (0 – 13)

Analogni pinovi- (A0 – A5)

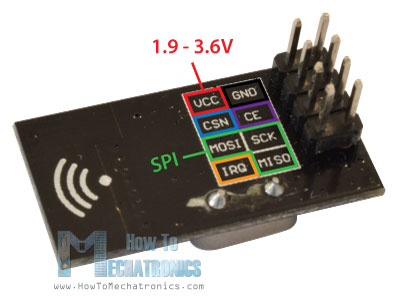
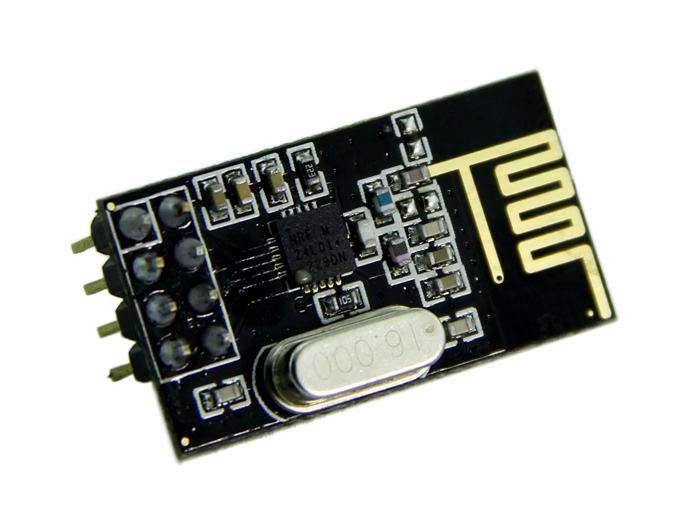
Svaki od 14 digitalnih i 6 analognih pinova može se koristiti kao ulaz ili izlaz ,te ih je moguće upravljat funkcijama pinMode(), digitalWrite() i digitalRead(). Digitalni pin se koristi pri naponu od 5 V. Svaki pin može dati ili primiti maksimalno 40 mA, te ima unutarnji pull-up otpornik 20-50 kOhma. U Arduinu digitalni signal se može nalaziti u dva stanja: LOW ili HIGH. Kad je pin spojen na 5 V smatra se HIGH, a kad je spojen na 0 V ili GND, smatra se LOW. Analogni naponi mogu imati bilo koju vrijednost napona između 0 i 5 V.

**2.2. Nordic nRF24L01+**

Nordic nRF24L01+ je primopredajnik kojega karakterizira niska potrošnja (eng. *Ultra Low Power - ULP*). Koristi frekvencijski opseg od 2,4GHz i može radit sa brzinom prijenosa od 250 kbps do 2 Mbps. Nordic nRF24L01+ modul može koristiti do 125 različitih kanala što omogućava stvaranje mreže od 125 neovisno funkcionalnih modula. Svaki kanal može imat do 6 adresa ili svaka jedinica može istovremeno komunicirat sa šest drugih jedinica.



*Slika 2.3 Prikaz nordic nRF24L01 mreže*



*Slika 2.4 Nordic nRF24L01*

Tri zelena pina sa slike 2.3 (lijevo) služe za ostvarivanje SPI komunikaciju, te trebaju biti povezani na SPI pinove arduina.

Važna svojstva:

* Frekvencijski pojas 2.4GHz
* 125 radio kanala
* Brzina transmisije (250kbps – 2Mbps)
* Napajanje: 1,9 V - 3,6 V
* Mala potrošnja snage
* Osjetljivost pri prijemu: <90 dB
* Domet prijenosa: 250 m na otvorenom
* Dimenzije: 15 x 29 mm

***2.3 Senzor VL53L0X***

*VL53L0X senzor predstavlja novu generaciju “Time of flight” laserskih modula. Ovaj senzor koristi se za mjerenje udaljenosti mete unutar dometa od 2 m. VL53L0X senzor sastoji se od 940 nm VCSEL odašiljača.VCSEL emitira laser (okomito) koji je potpuno nevidljiv ljudskom oku, te infracrveni filtri omogućuju dulji domet i veću otpornost na okolna svjetla.*



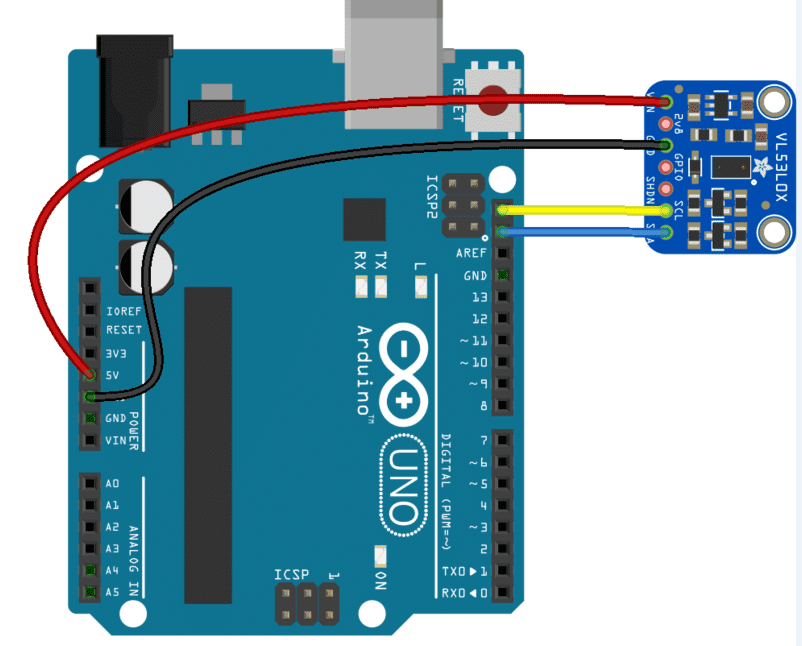
*Slika 2.5 Senzor VL53L0X*

*Svojstva:*

* *Napon napajanja 2.6 V do 5,5 V*
* *Struja napajanja: 10 mA*
* *Komunikacija: IC2 protokol*
* *VCSEL laser 940 nm*
* *Ugrađen linearni regulator napona 2,8V*
* *Domet 2m*

*Senzor VL53L0X se koristi u raznim aplikacijama kao što su:*

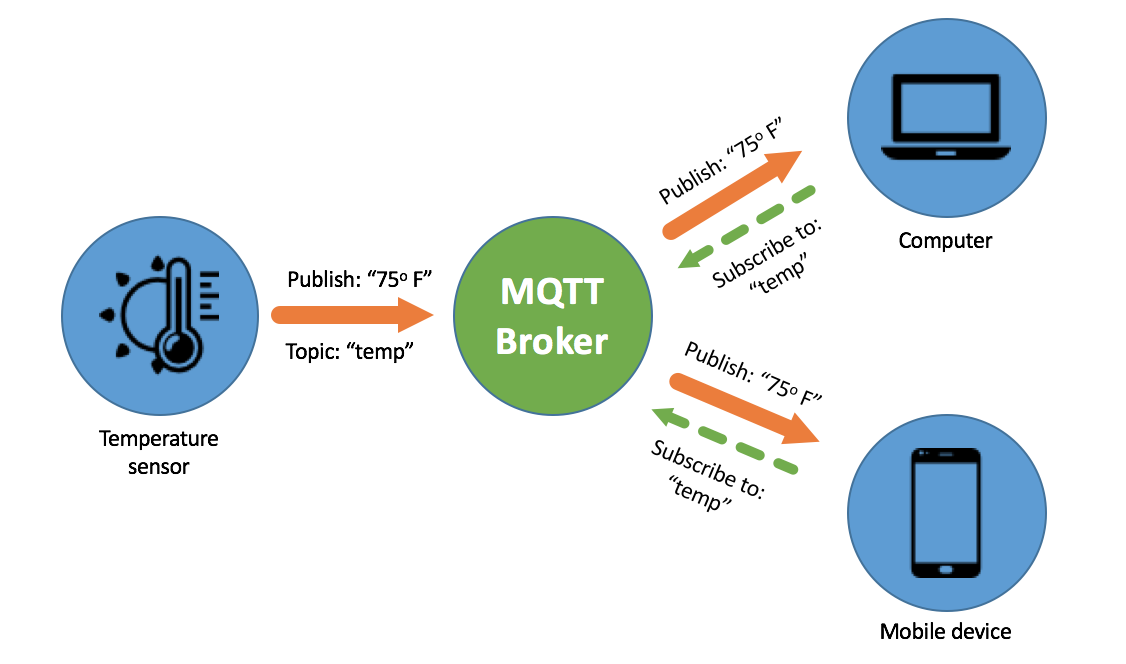
* *Detekcija ruku kod sušilica za ruke, dozatorima sapuna, slavinama itd...*
* *Koristi se za autofokus kod fotoaparata u uvjetima slabe osvjetljenosti*
* *Za razne primjene u telefonima*
* *Otkrivanje prepreka u robotici*



*Slika 2.6 Shema spajanja Arduina i Senzora*

***3.MQTT***

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.



*Slika 3.1 Primjer MQTT protokola*

MQTT koristi teme (topic) kako bi odredio koja poruka ide kojem preplatniku.Tema je hijerarhijski strukturiran niz koji se moze koristiti za filtriranje i usmjeravanje podataka. Publisher određuje neku temu i objavljuje podatke pod tom temom. Subscriber se preplaćuje na tu temu i prima te podatke. Uloga brokera je da dostavi podatke subscriberu.

**4. DOCKER**

Docker je softver koji je moguće koristiti na Linux i Windows operativnim sustavima. To je alat dizajniran za kreiranje, implementaciju i pokretanje aplikacija koristeći kontejnere. Prvenstvena namjena je za programere koji za svoj posao mogu odabrati željenu platformu te na njoj programirati bez brige o operativnom sustavu na kojem će se njihova aplikacija pokrenuti. Mi ćemo u ovom radu detaljnije proučiti njegove primjene u IoT-u, prvenstveno implementaciju i administraciju IoT uređaja.

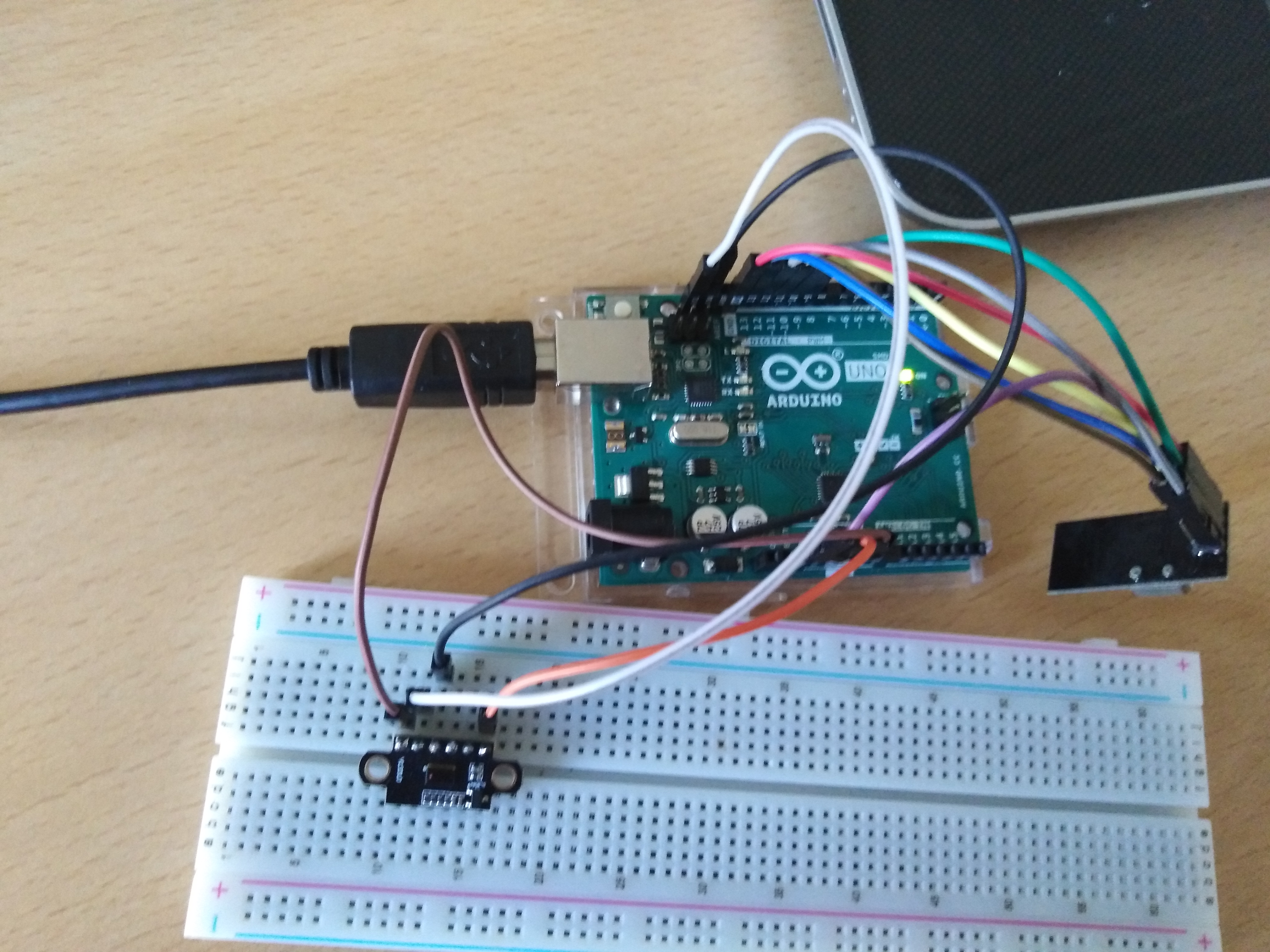
Multi-container Docker aplikacija koristi sljedeće servise;

* Node-Red flow-based razvojni alat za vizualno programiranje (port : 1880)
* Influx DB – time series baza podataka (port : 8086)
* Chronograf – admin korisničko sučelje za influxDB (port: 127.0.0.1:8888)
* Grafana – vizualizacijsko korisničko sučelje za influxDB (port: 3000)

**5.PRAKTIČNA REALIZACIJA PROJEKTA I REZULTATI**

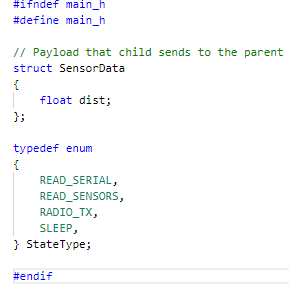
**5.1 Transmitter**

Za realizaciju transmittera koristili smo senzor VL53L0X, Arduino Uno pločicu i radio modul nRF24L01.Senzor očitava podatke u postavljenoj okolini , te ih preko radio modula šalje prema receiveru. Na slici 5.1 prikazan je realiziran transmitter u našem slučaju.



*Slika 5.1 transmittera*

*Programski kod za transmitter je podjeljen dvije biblioteke, jedan za čitanje senzora i jedan za slanje/primanje podataka.Da bismo radili sa senzorom potebno je instalirat određene bilbioteke unutar kojih se nalaze određeni kodovi. Senzor VL53L0X ispisuje vrijednosti udaljenosti razine vode od senzora u određenim vremenskim intervalima. Mjerna jedinica za udaljenost je u mm (milimetrima). U State machine smo stvorili 4 stanja u kojima se arduino može nalazit .* Svrha ovoga je naravno bila ušteda energije kod baterijskog napajanja.



State Mašina (main.cpp):

#include <Arduino.h>

#include <LowPower.h>

#include "main.h"

#include "Sensors.h"

#include "Radio\_nRF.h"

SENSORS sensor;

RADIO radioNRF;

SensorData dataToSend;

StateType state = READ\_SERIAL;

void setup()

{

Serial.begin(9600);

radioNRF.nRF\_init();

sensor.vl53l0x\_init();

}

void loop()

{

Serial.println("S");

switch (state)

{

case READ\_SERIAL:

state = READ\_SENSORS;

break;

case READ\_SENSORS:

dataToSend.dist = sensor.readDistance();

state = RADIO\_TX;

break;

case RADIO\_TX:

radioNRF.RF\_send(dataToSend);

state = SLEEP;

break;

case SLEEP:

delay(50);

LowPower.powerDown(SLEEP\_8S, ADC\_OFF, BOD\_OFF);

delay(50);

state = READ\_SERIAL;

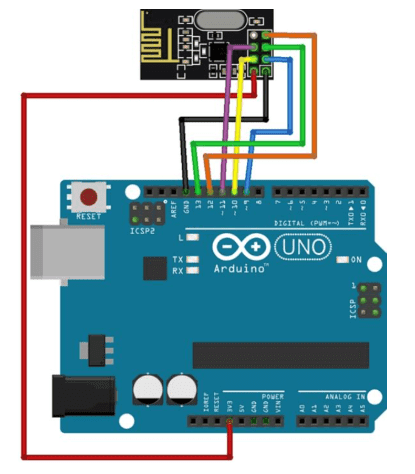
break;

default:

break; }

**5.2 RECEIVER**

Receiver smo realizirali spajanjem arduino uno pločice i radio modula nRF24L01. Receiver prima podatke poslane od strane transmittera.



*Slika 5.2 Shema spajanja Receivera Slika 5.3 Realizacija receivera*

*Programski kod za receiver se sastoji od phyton skripte i .cpp file-a za čitanje primljenih podataka.Pomoću python skripte šaljemo podatke na mqtt server, te time uspostavljamo serijsku komunikaciju . U python skripti definirani su podatci o brokeru, temi, klijentu.*

*Primjer korištenog koda u python skripti:*

import serial

import paho.mqtt.client as mqtt

BROKER = "mqtt.eclipse.org"

CLIENTID = "MQTTExample"

TOPIC\_ONE = "A507/sensors/distance"

COMPORT = "COM10"

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, 9600, timeout=5)

while True:

message = ser.readline()

print (message)

if b'P:' in message:

string, distance= message.split(b' ')

distance, rest = distance.split(b'!')

print(distance.decode('utf-8'))

mqttc.publish(TOPIC\_ONE, payload=distance.decode('utf-8'), qos=QOS, retain=False)

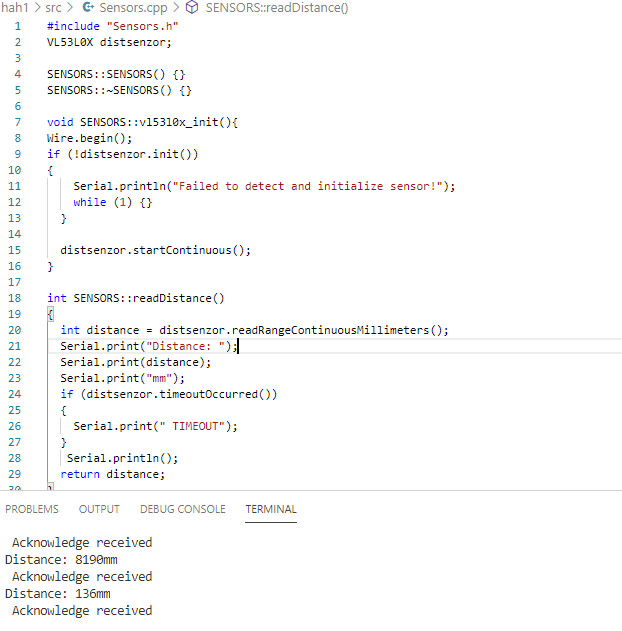
time.sleep(0.01)

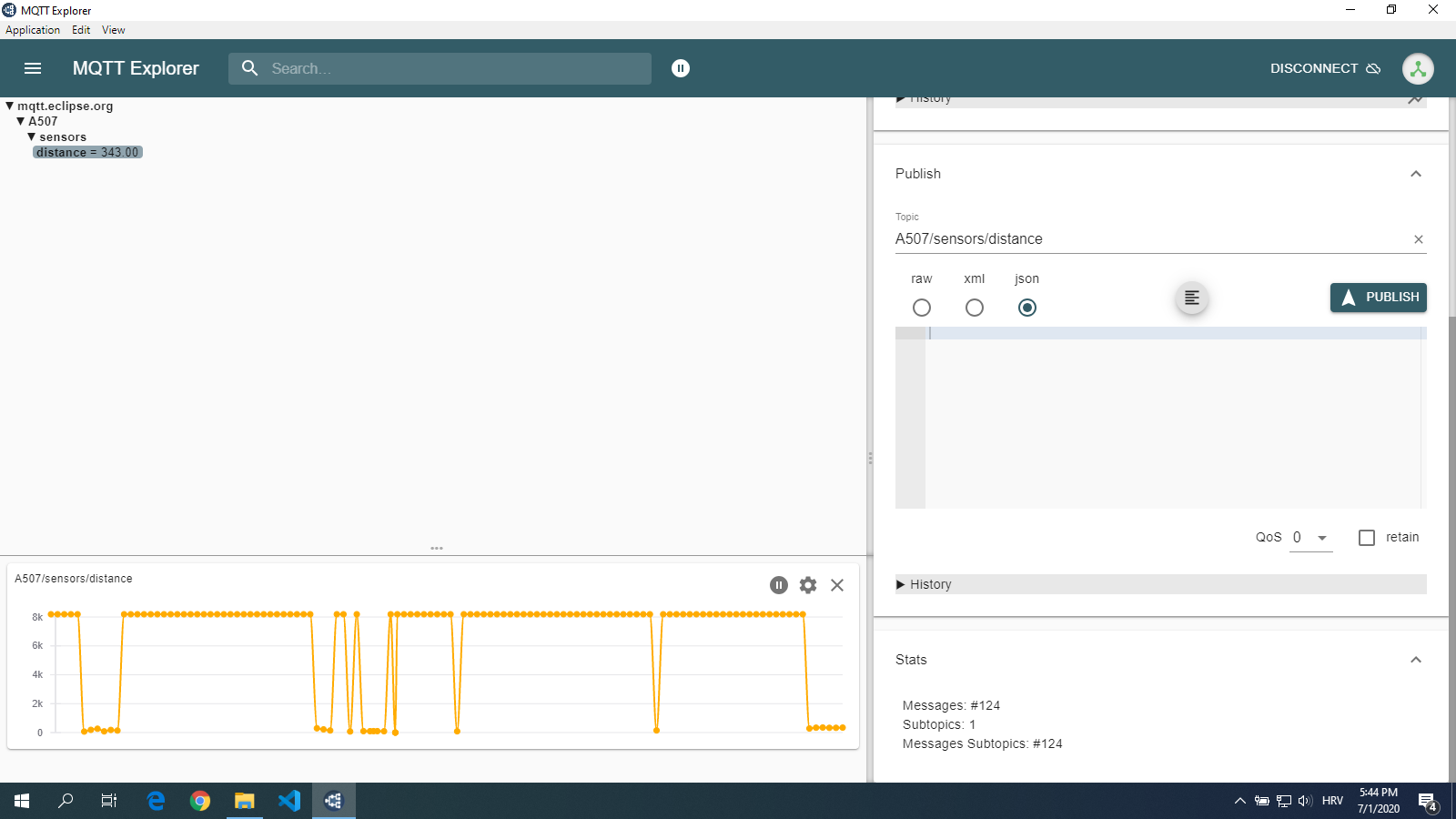
mqttc.disconnect()

time.sleep(1)

**5.3 PRIKAZ DOBIVENIH REZULTATA**

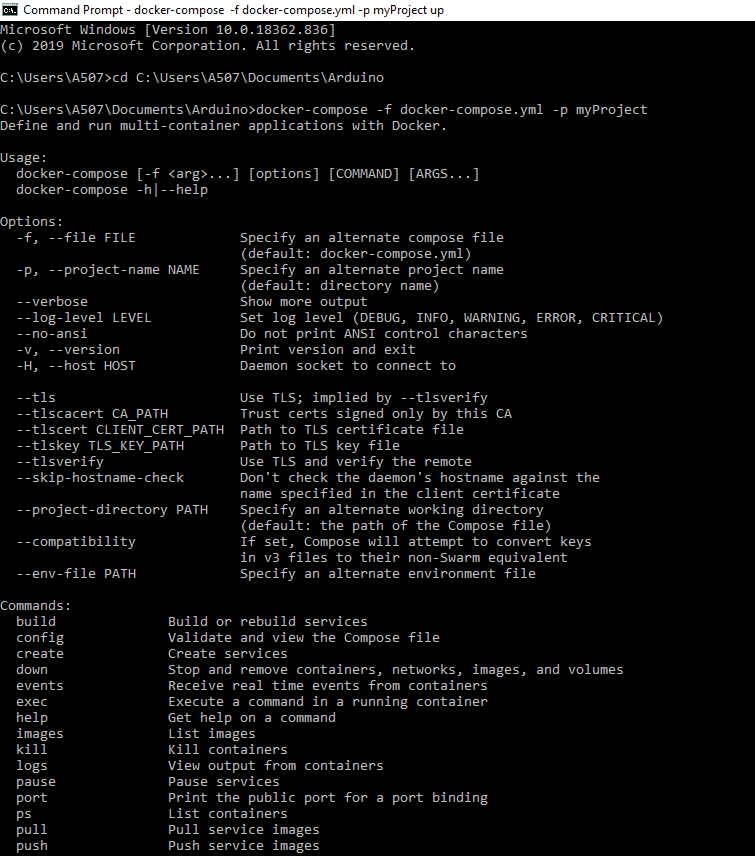
U Visual studio code-u pokrenemo transmitter koji počinje očitavat podatke razine vode , te ih slati . Pokretanjem python skripte u terminalu se mogu vidjeti vrijednosti razine vode.

*Slika 5.4 Očitane vrijednosti transmittera*

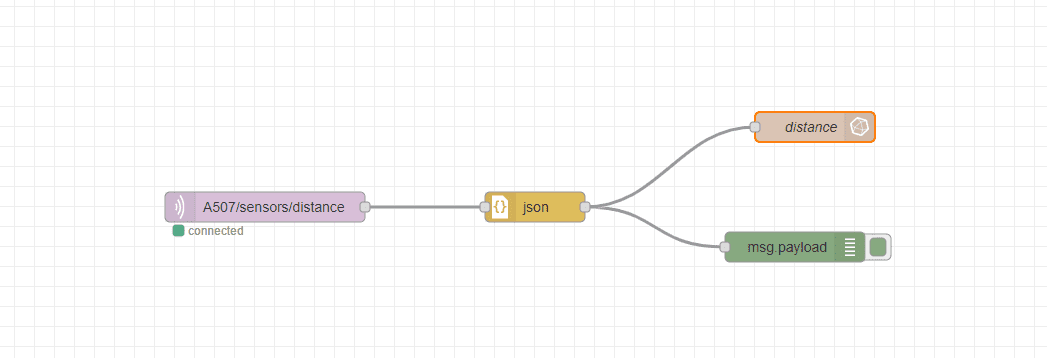


*Slika 5.5 Prikaz podataka na MQTT-u*

Kako bismo koristili Grafanu za vizualizaciju podataka ,potrebno je pokrenuti docker aplikaciju.



Slika 5.6 Prikaz pokretanja docker aplikacije

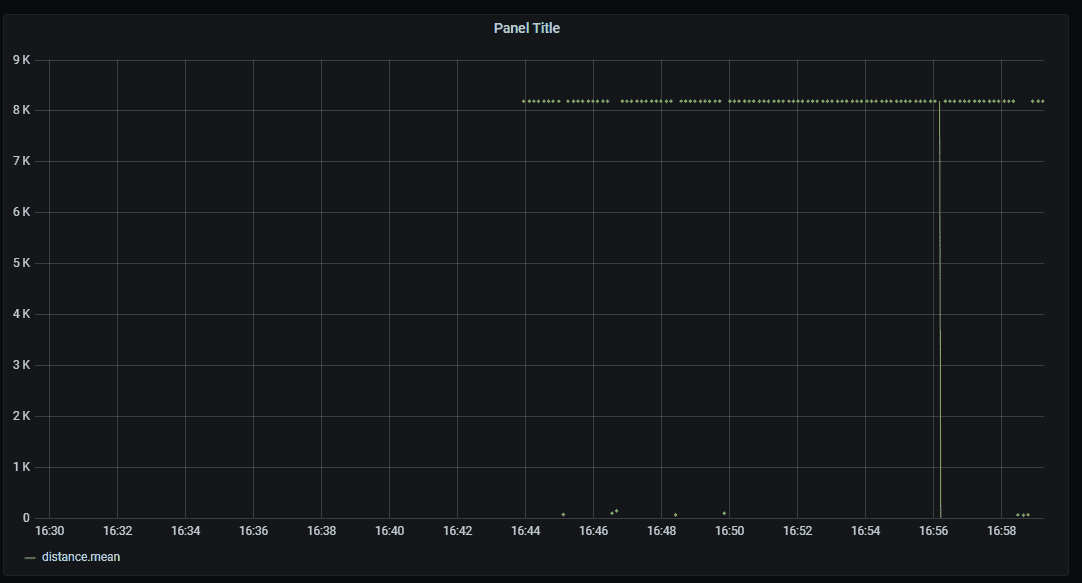


*Slika 5.7 Node Red flowchart*

Nakon pokretanja docker aplikacije i spajanja Node Reda na MQTT broker , tada smo u mogućnosti koristit vizualni prikaz dobivenih podataka pomoću grafane.



*Slika 5.8 Grafički prikaz razine vode uz vanjski utjecaj*



*Slika 5.9 Grafički prikaz razine vode bez vanjskih utjecaja*

Na slici 5.8 možemo uočiti skokovite promjene u vrijednosti razine vode zbog izloženosti posude s vodom vanjskim utjecajima, te promjene senzor VL53L0X očitava te ih šalje receiveru. Na slici 5.9 Posudu s vodom smo izolirali od vanjskih utjecaja ,te razina vode se ne mijenja, što možemo očitat iz grafa.

Iz dobivenih rezultata u projektu možemo zaključiti da senzor VL53L0X ima dobar omjer kvalitete i Cijene.Prije prvog korištenja potrebna je kratka kalibracija, nakon koje je senzor spreman za uporebu. Prilikom upotrebe uočili smo manu senzora VL53L0X, odnosno senzor ne može očitavat promjene u dometu manjem od 58mm.

**LITERATURA**

[1] <http://mqtt.org/faq>

[2] <https://www.arduino.cc/en/Guide/ArduinoUno>

[3] <https://components101.com/sensors/vl53l0x-lidar-distance-sensor>

[4] <https://www.st.com/resource/en/datasheet/vl53l0x.pdf>

[5] <https://docs.docker.com/get-started/overview/>

[6] <https://github.com/toperkov/WiSe-2019-20/tree/master/instructions>

[7] <https://github.com/toperkov/WiSe-2019-20/blob/master/instructions/lab-9.md>

[8] <https://nodered.org/>