УНИВЕРЗИТЕТ
У НОВОМ САДУ

ФАКУЛТЕТ
ТЕХНИЧКИХ НАУКА

Трг Доситеја Обрадовића 6, 21000 Нови Сад, Југославија
Деканат: 021 350-413; 021 450-810; Централа: 021 350-122
Рачуноводство: 021 58-220; Студентска служба: 021 350-763
Телефакс: 021 58-133; e-mail: ftndean@uns.ns.ac.yu

RAČUNARSKA ELEKTRONIKA

NAZIV PROJEKTA:

Weather Station

TEKST ZADATKA:

Implementirati stanicu za očitavanje osnovnih vremenskih paramtetara u okolini pomoću RaspberryPi pločice i odgovarajućih senzora. Takođe je potrebno kreirati i odgovarajući user interface poštujući osnovne principe istog.

Mentor:
Prof. Ivan Mezei

Student:
Petar Stamenković EE18/2019

U Novom Sadu, 11.07.2023.

SADRŽAJ

1. UVOD

Projektna tema je Weather Station(Stanica za očitavanje vremenskih paramtera) čiji je cilj da pomoću RaspberryPi pločice, određenih senzora i user interface-a omogući korisniku da uvidi trenutno stanje u njegovoj prostoriji.

2. KORIŠĆENE KOMPONENTE

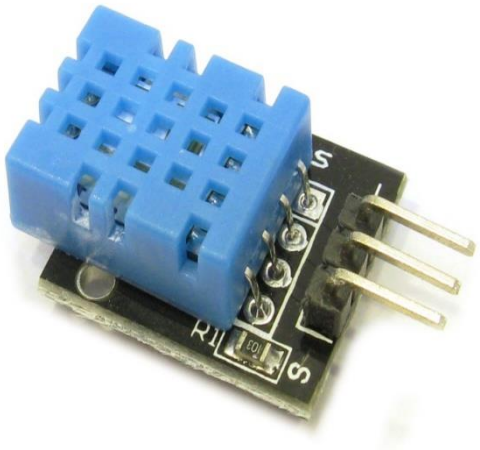
2.1. RasperryPi

Za realizaciju projekta korišćen je RPi model 4.

2.2. DHT11 Senzor

DHT11 je senzor pomoću kojeg je moguće očitati temperaturu i vlažnost vazduha u prostoriji. Za projekat je korišćen modul sa 3 pina:

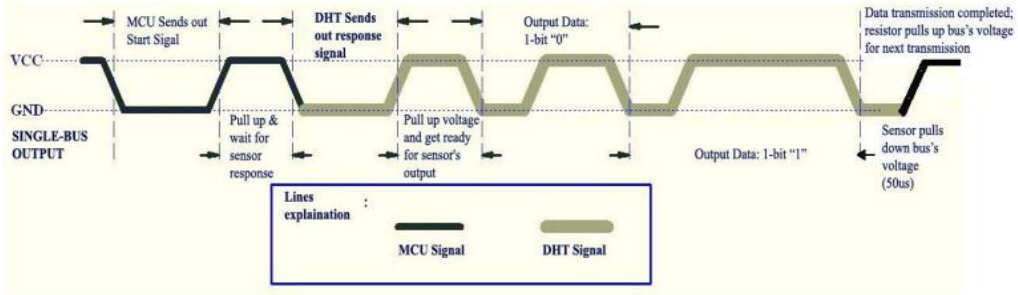
1. VDD – Napajanje od 3V do 5.5V
2. GND – Uzemljenje
3. Data – Port odakle se informacije izvlače serijski



Slika1 : DHT11 Senzor

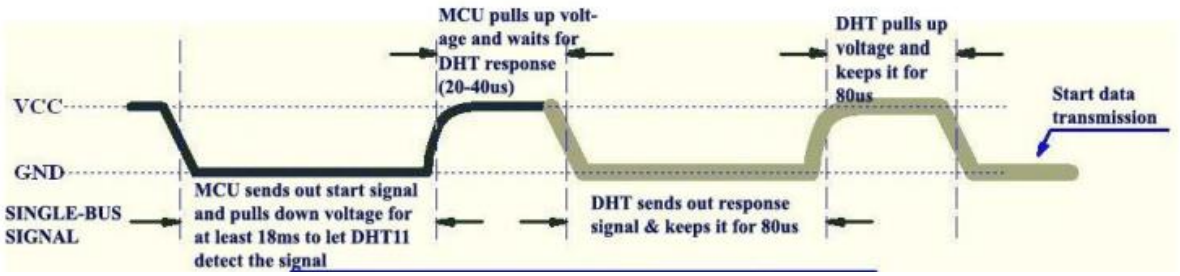
Temperaturni opseg je od 0 do 50 stepeni Celzijusa a opseg vlažnosti vazduha je od 20% do 90%, uz preciznost ±1% za obe veličine. Obe veličine su 16-bitne(8 bita za ceo podatak i 8 bita za decimalni deo podatka) a postoji i 5 blok od 8 bita koji služi za proveru validnosti podatka(biće prikazano) dakle ukupno 40 bita.

Serijska komunikacija između Rpi i DHT11 se zasniva na tome da senzor čeka start signal od Rpi i kada ga dobije u vidu responsa vraća pomenutih 40bita i prelazi u running-mode. Kada se podaci prihvate DHT11 se vraća u low-power-consumption mod(slika 2).



Slika 2: Početak komunikacije

Magistrala je na visokom nivou(VCC) i pri početku komunikacije, Rpi će je spustiti na barem 18ms kako bi DHT detektovao signal njemu nadređenog, a zatim će se magistrala vratiti na visok nivo i čekati tu između 20us i 40us odgovor od senzora. (Slika 3)



Slika 3: Vremenski parametri komunikacije

```
// DHT11 Communication requests
// Pull pin down for 18ms (Due to specification)
pinMode(DHTPIN,OUTPUT);
digitalWrite(DHTPIN,LOW);
delay(18);
//Prepare to read after 18ms
digitalWrite(DHTPIN,HIGH);
delayMicroseconds(40);
pinMode(DHTPIN,INPUT);
```

Slika 4: Realizacija komunikacije u .cpp fajlu

U kodu, response koji senzor vraća je prikazan preko dht_dat niza gde su članovi dht_dat[0] i dht_dat[1] za vlažnost, dht_dat[2] i dht_dat[3] za temperaturu i dht_dat[4] član koji proverava validnost sabiranjem prethodnih.

```
// Checking the check-sum last byte and extracting data
if((j>=40) && (dht_dat[4] = (dht_dat[0] + dht_dat[1] + dht_dat[2] + dht_dat[3]) & 0xFF))
{
    // Read Humidity
    float humidity = (float)((dht_dat[0] << 8) + dht_dat[1]) / 10;
    if(humidity > 100)
    {
        humidity = dht_dat[0];
    }
    // Read Temperature
    float temperature = (float)(((dht_dat[2] & 0x7F) << 8) + dht_dat[3]) / 10;
    if(temperature > 125)
    {
        temperature = dht_dat[2];
    }
    if(dht_dat[2] & 0x80)
    {
        temperature = -temperature;
    }
```

Slika5:Ekstrakcija podataka uz proveru valdinosti

2.3. Rain/Water detection senzor

Za detekciju kiše korišćen je standardni senzor sa bakarnim žicama(često korišćen uz arduino). Moguće ga je koristiti još i za merenje dubine i količine tečnosti međutim za potrebe ovog projekta detekcija je jedina bitna. Pinologija je sledeća :

1. + (VDD) – Napajanje od 3V do 5V
2. – (GND) – Uzemljenje
3. S (Signal) – Očitavanje vrednosti senzora

Slika 6 : Water level sensor

Njegova upotreba u kodu je trivijalna i koristi se samo funkcija *digitalRead* sa pina koji smo mi odredili. Ukoliko je vrednost HIGH kiša je detektovana, dok LOW govori suprtono.

```
pinMode(WATER_SENSOR,INPUT);
int wat_value = digitalRead(WATER_SENSOR);
// This if branch is just a testing branch, it should be deleted in the final version
if(wat_value == HIGH)
{
    //std::cout << "Water detected" << std::endl;
    ui-> label_4->setText("Rain is detected");
}
else
{
    //std::cout << "No water detected" << std::endl;
    ui-> label_4->setText("No rain is detected");
}
```

Slika 7 : Segment koda za detekciju kiše

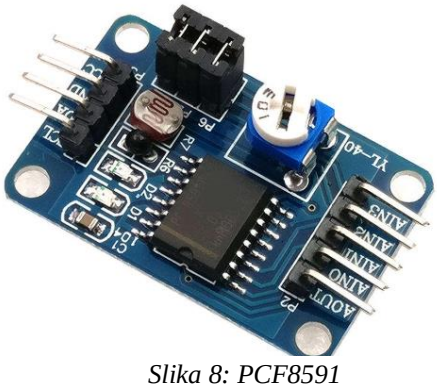
2.4. PCF8591 (YL-40) Modul

Detekcija svetlosti je rađena pomoću PCF8591 modula koji radi A/D konverziju i sadrži na sebi jedan foto-otpornik. Analogni napon koji ide sa foto-otpornika se pretvara u digitalan i on se isčitava pomoću funkcije *wiringPiI2CReadReg8*.

RPi komunicira sa modulom pomoću I2C komunikacije pomoću wiringPi biblioteke. U terminalu, komandom i2cdetect potrebno je pronaći adresu modula jer se njom inicijalizuje isti u kodu(slika 9.)

```
fd = wiringPiI2CSetup(0x48); // Check the address for my case
```

Slika 9 : Linija inicijalizacije i2c komunikacije



Slika 8: PCF8591

2.5. Ostalo

Pored navedenih komponenti korišćeni su još i :

1. Protobord
2. Kratkospojnici
3. Potrebni kablovi za RPi

3. FUNKCIJE U KODU

Kod je particionisan u nekoliko funkcija radi preglednosti i smešten je skoro u potpunosti u dialog.cpp fajl, dok su u dialog.h fajlu uključene neke biblioteke i deklarisane funckije i elemetni GUI-ja.

3.1. Dialog klasa

Ova klasa predstavlja segment koda gde se inicijalizuje elementi gui-ja, i2c komunikacija, wiringPi biblioteka kao i tajmeri koji pozivaju određene slotove(funkcije). Ona klasa takođe ima i svoj predefinisani destruktor.

3.2. Temperature_humidity_read

Baš kao što joj ime kaže, u ovoj funkciji se podešava senzor za temperaturu i vlažnost (DHT11), ekstrahuju se podaci i ispisuju na labele i QChart koji ćemo kasnije spomenuti. Ovde se takođe računa prosečna temperatura i vlažnost tokom proteklog vremena. Funkcija se poziva istekom timera1 na svaku sekundu.

3.3. Forecast_read

U ovoj funkciji kombinujemo podatke od senzora za kišu i svetlost i u zavisnosti od njihove kombinacije prosleđujemo određene podatke i informacije na labele. Takođe, pojavljuje se i slika koja prikazuje adekvatno vreme u gornjem desnom ćošku. Funkcija se poziva istekom timera2 na svaku sekundu.

3.4. Clear_chart

Ova funkcija se poziva pritiskom na *Clear chart* dugme i ona resetuje određene labele kao i QChart.

3.5. Start_temperature_humidity_timer

Ova funkcija je povezana sa *Temperature and humidity* dugmetom i startuje timer1 i time i rad senzora DHT11.

3.6. Start_forecast_timer

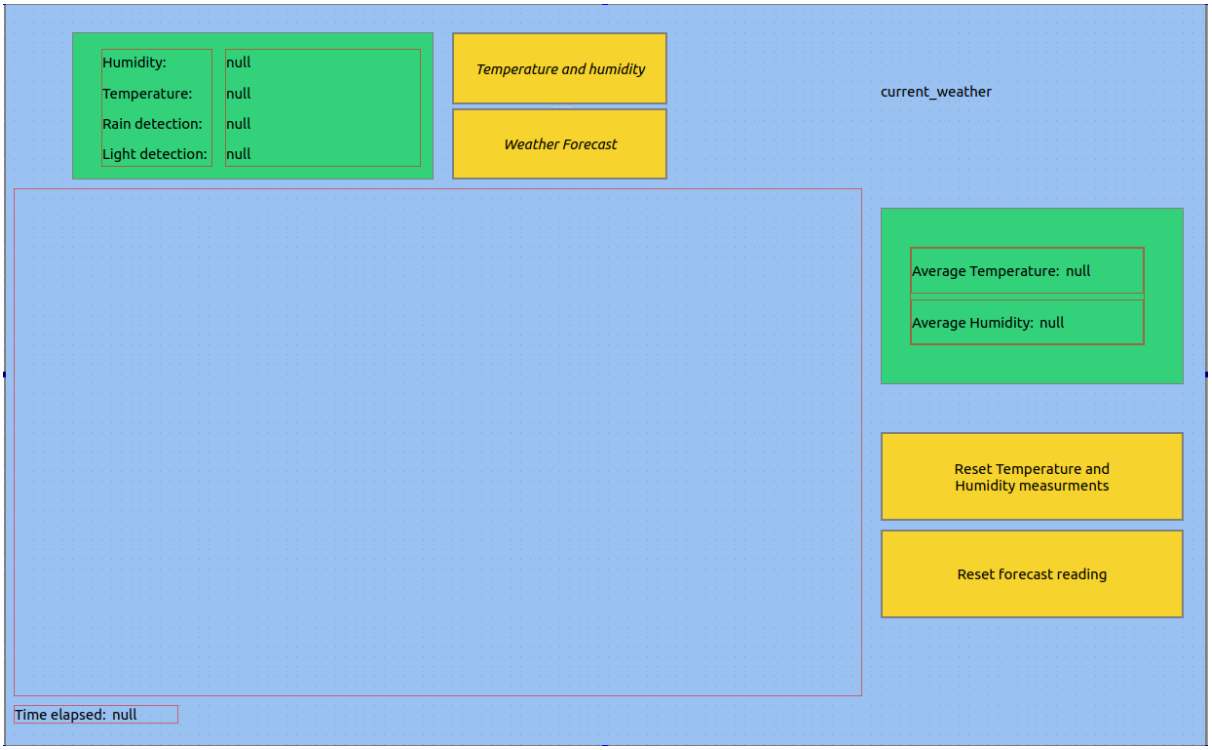
Ova funckija je povezana sa *Weather Forecast* dugmetom i startuje timer2 i time i iscrtavanje slika i ispis na labele za kišu i svetlost.

3.7. clear_forecast

Ova funkcija resetuje timer2 i merenja što se tiče kiše i svetlosti. Sliku postavlja u reset stanje. Funkcija se poziva pritiskom na dugme *Reset forecast reading*.

4. USER INTERFACE (GUI)

Veoma bitan deo projekta jeste i adekvatno korisničko okruženje pa će u kratkim crtama i ono biti objašnjeno. U nastavku su navedeni najosnovniji elementi koji su korišćeni :



Slika 9 : Izgled korisničkog interfejsa

4.1. QChart

Za lepši prikaz temperature i vlažnosti korišćen je QChart koji je podešen da ima dve Y ose na kojima će se prikazivati pomenute veličine. Ubačena je legenda u chart, dok su *temp* i *hum* QlineSeries instance u koje se smeštaju naši podaci. Pomenuto dugme *Clear chart* resetuje QChart, *temp i hum*. Chart je smešten u *vericalLayout* koji je uokviren tankom crvenom linijom na slici 9.

4.2. PushButtons

Standardni dugmići tipa PushButtons su korišćeni za pomenute funkcije i označeni su žutom bojom na slici 9.

4.3. QPixMap

Ova klasa je korišćena u funkcji *forecast_read* gde se u zavisnosti od paramtera metodom *setPixmap* na labelu nalepljuje određena slika iz foldera *Icons* koji dolazi u okviru projekta.

4.4. Lables

Najviše korišćeni elementi koji služe za ispis podataka. U zelenom okviru na slici 9 , koriste se u kombinaciji sa *vericalLayout i horizontalLayout*.

4.5. QMessageBox

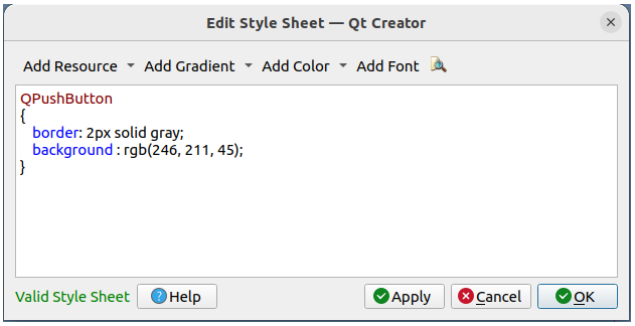
Ova klasa je korišćena za neka standardna obaveštenja u kodu poput : merenja su pokrenta ili merenja su zaustavaljena kao i za neka upozorenja poput : previsoka temperatura ili moguća kiša.

4.6. StyleSheets i boje

Korišćena je kombinacija boja koja pomaže korisniku da brzo shvati da se radi o vremenskoj stanici.

- Plava – Simbolizuje nebo
- Žuta – Simbolizuje sunce
- Zelena – Simbolizuje travu i prirodu.

Jedno podešavanje *styleSheet-a* je prikazano na slici 10.



Slika 10: Podešavanje boje i okvira jednog dugmeta

5. ZAKLJUČAK

Na osnovu znanja iz dosadašnjih predmeta koji su obuhvatali tehnike programiranja kao i predemta "*Računarska Elektronika*", projekat je uspešno realizovan. Korišćena je RaspberryPi pločica u kombinaciji sa par senzora i projektovan je odgovarajući korisnički interfejs koji poštuje osnovne principe projektovanja. U prilogu sa ovom dokumentacijom je i kratak demonstrativni video koji pokazuje korektno ponašanje sistema, kao i sav kod koji je korišćen pri izradi.