



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA MIKRORAČUNARSKU
ELEKTRONIKU



**Merenje osunčanosti i vlažnosti zemljišta
kod biljaka uz pomoć senzora**
Projekat iz predmeta Računarska Elektronika

Student: Mladen Ilić EE116/2016
Mentor: Prof. Dr. Ivan Mezei

1. Opis zadatka

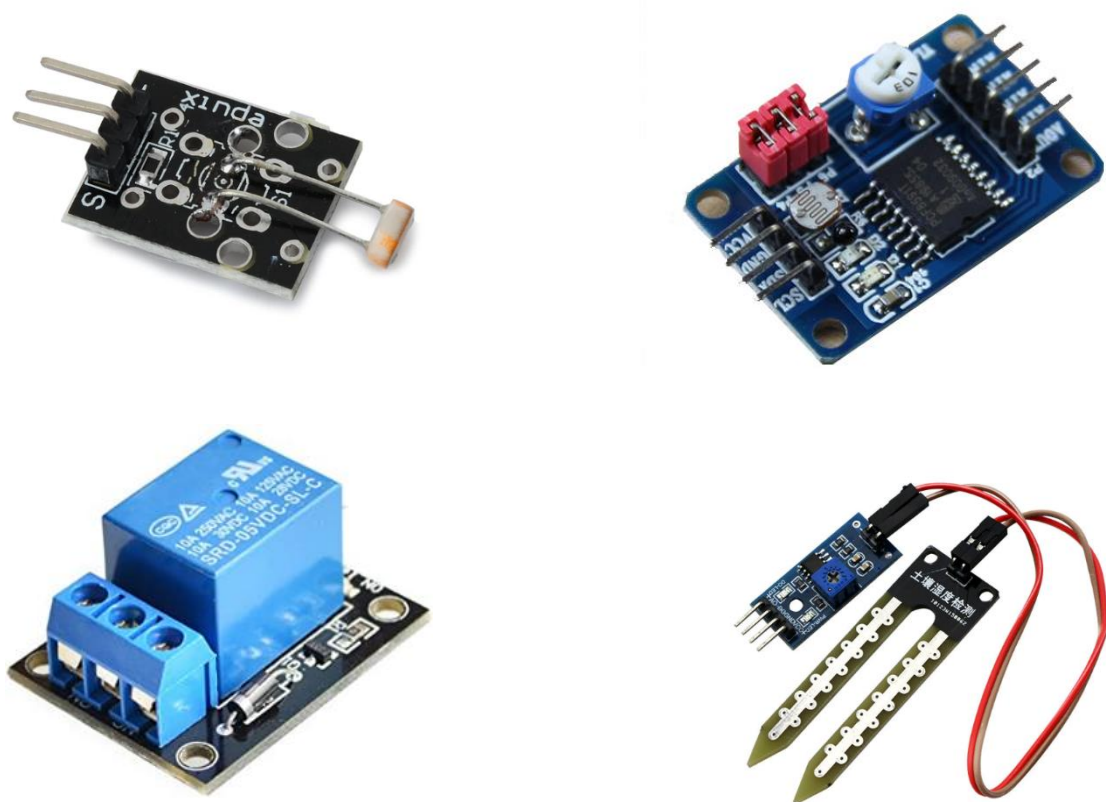
Cilj projekta je da se uz pomoć *RaspberryPi* meri vlažnost zemljišta i osunčanost biljke kao i zalivanje iste, korišćenjem YL-40 pločice, senzora za vlažnost (FC-28), senzora za osvetljenje (Foto senzor) i 5V releja na koji je povezan elektromagnetni ventil. Tačnije rečeno, potrebno je omogućiti uvid i kontrolu stanja biljke preko GUI aplikacije.

2. Opis konkretne realizacije

Na početku je potrebno povezati senzore sa *RaspberryPi*. Prvi modul koji se povezuje je YL-40 kako bi se obezbedila AD koverzija. YL-40 ima 9 pinova od kojih se koriste 6 (VCC, GND, SDA, SCL, AIN1, AIN2) i povezuju se na GPIO (*General Purpose Input Output*) pinove na RPi-u. Pločica YL-40 se napaja na 3V3.

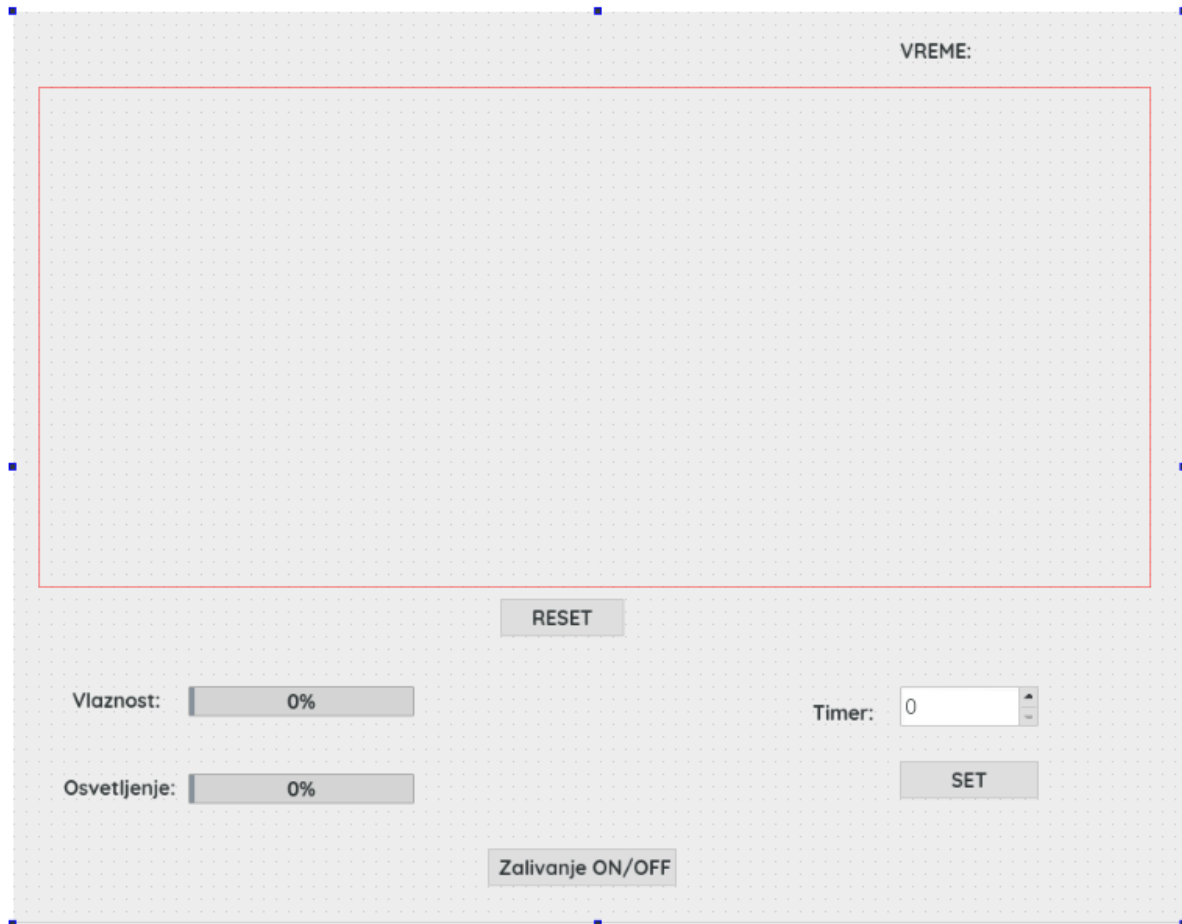
Sledeći senzor koji povezujemo je foto senzor koji ima 3 pina, čiji pin za signal spajamo na AIN2 pin YL-40 modula, a pinove VCC (3V3) i GND spajamo na Rpi. Na sličan način povezujemo i senzor za vlagu s tim da VCC spajamo na 5V, a signalni pin se povezuje na AIN1 pin YL-40 modula.

Kod 5V releja VCC pin je povezan na 5V, GND je spojen sa GND, a pin signala je spojen na GPIO 21.



Slika 1. Senzori

Nakon povezivanja prelazi se na pravljenje GUI interfejsa i kucanja koda za projekat. Aplikacija sadrži tri **push button**-a(RESET, SET, Zalivanje ON/OFF), dva **progress bar**-a (vlažnost, osunčanost), jedan **spinbox** (timer), i jedan **vertical layout** (grafik) i nekoliko **label**-a od kojih jedna prikazuje trenutno vreme.



Slika 2. Izgled GUI-a

Pre pocetka kodiranja potrebno je uključiti sve potrebne biblioteke koje će se koristiti u projektu. Prvo što treba da se uradi jeste dodavanje par linija unutar *.pro* fajla bez kojih ne može da se radi.

```
QT += charts
LIBS += -L/usr/local/include -lwiringPi
```

Slika 3. Linije koje se dodaju u *.pro* fajl

Prva linija služi za dodavanje *wiringPi* biblioteke tj. putanje ka toj biblioteci. Druga linija se dodaje kako bi se omogućio rad sa grafikom.

Nakon *.pro* fajla prelazimo na pisanje *.hpp* fajla. Tu je važno uključiti odgovarajuće biblioteke kao i deklarisanje funkcija i promenljivih koje ćemo koristiti.

```
#ifndef DIALOG_H
#define DIALOG_H

#include <QtCharts/QLineSeries>
#include <QWidget>
#include <wiringPi.h>
#include <wiringPiI2C.h>
#include <stdio.h>
#include <QtGlobal>
#include <QDialog>
#include <stdlib.h>
#include <QtCharts/QChartView>
#include <QTimer>
#include <QTime>
#include <QDateTime>
#include <QProgressBar>
#include <QSpinBox>
#include <QDebug>

using namespace QtCharts;

namespace Ui {
class Dialog;
}

class Dialog : public QDialog
{
    Q_OBJECT

public:
    explicit Dialog(QWidget *parent = nullptr);
    const char PCF8591 = 0x48; //adresa
    int fd, adcSvetlo, adcVlaga;
    int svetlo, vlaga, brojac, minut, dozvola_za_brojac=0;
    int val, counter=0;
    int set_brojaca = 0;

    QLineSeries *series;

    ~Dialog();

private slots:
    void on_pushButton_clicked(bool checked);
    void on_pushButton_2_clicked();
    void senzori();
    void prikazVremena();
    void on_pushButton_3_clicked();
    void count();

private:
    Ui::Dialog *ui;
    QTimer *timer1;
};

#endif // DIALOG_H
```

Slika 4. *.hpp* fajl

U *.hpp* fajlu je deklarirano šest funkcija:

1. **void on_pushButton_clicked(bool checked)** – omogućava navodnjavanje nakon klika na dugme.
2. **void on_pushButton_2_clicked()** – daje mogućnost resetovanja grafika.
3. **void senzori()** – reguliše tj. prikuplja podatke iz senzora i prikazuje ih preko *progress bar*-a.
4. **void prikazVremena()** – prikazuje trenutno vreme.
5. **void on_pushButton_3_clicked()** – postavlja uneto vreme u tajmer za navodnjavanje.
6. **void count()** – otvara elektromagnetni ventil nakon što tajmer istekne.

Nakon deklarisanja funkcija unutar *.hpp* fajla, potrebno je te iste funkcije definisati u *.cpp* fajlu.

```
1  #include "dialog.h"
2  #include "ui_dialog.h"
3
4
5  Dialog::Dialog(QWidget *parent) :
6      QDialog(parent),
7      ui(new Ui::Dialog)
8  {
9      ui->setupUi(this);
10
11      QTimer *timer = new QTimer(this);
12      connect(timer, &QTimer::timeout, this, &Dialog::senzori);
13      connect(timer, &QTimer::timeout, this, &Dialog::prikazVremena);
14      timer -> start(1000);
15
16      timer1 = new QTimer(this);
17      connect(timer1, &QTimer::timeout, this, &Dialog::count);
18
19      pinMode(21, OUTPUT);
20
21      series = new QLineSeries();
22
23      QChart *chart = new QChart();
24      chart->legend()->hide();
25      chart->addSeries(series);
26      chart->createDefaultAxes();
27
28      chart->axisX()->setRange(0, 60); //sekunde
29      chart->axisY()->setRange(0, 100); //procenat vlage
30
31      chart->setTitle("Osuncanost biljke u toku 1 minuta");
32
33      QChartView *chartView = new QChartView(chart);
34      chartView->setRenderHint(QPainter::Antialiasing);
35
36      ui -> verticalLayout -> addWidget(chartView);
37
38      brojac = 0;
39 }
```

Slika 5. Definisanje tajmera, izlaznog pina i kreiranje grafika

Kod definisanje funkcije *void senzori()* potrebno je uključiti *wiringPiI2C.h* biblioteku jer je za pristup i očitavanje podataka sa gore navednih senzora primenjeno povezivanje preko I2C magistrale. Nakon očitavanja adc vrednosti (0-255), izvršeno je skaliranje tih vrednosti na skalu od 0 do 100%, a zatim su te vrednosti ispisane na *progress bar*-u. Takođe, paralelno sa tim se iscrtava i grafik koji prikazuje procenat osunčanosti u toku jednog minuta.

```

40
41 Dialog::~Dialog()
42 {
43     delete ui;
44 }
45
46 void Dialog::senzori() {
47     fd = wiringPiI2CSetup(PCF8591);
48
49     wiringPiI2CReadReg8(fd, PCF8591 + 2);
50     adcSvetlo = wiringPiI2CReadReg8(fd, PCF8591 + 2);
51     svetlo = 100 - (adcSvetlo*100/255);
52
53     wiringPiI2CReadReg8(fd, PCF8591 + 1);
54     adcVlaga = wiringPiI2CReadReg8(fd, PCF8591 + 1);
55
56
57     if (adcVlaga > 240) {
58         vlaga = 0;
59     } else if (adcVlaga > 235){
60         vlaga = 10;
61     } else if (adcVlaga > 230){
62         vlaga = 20;
63     } else if (adcVlaga > 225){
64         vlaga = 30;
65     } else if (adcVlaga > 220){
66         vlaga = 40;
67     } else if (adcVlaga > 215){
68         vlaga = 50;
69     } else if (adcVlaga > 210){
70         vlaga = 60;
71     } else if (adcVlaga > 205){
72         vlaga = 70;
73     } else if (adcVlaga > 200){
74         vlaga = 80;
75     } else if (adcVlaga > 195){
76         vlaga = 90;
77     } else {
78         vlaga = 100;
79     }
80
81     ui->progressBar->setValue(vlaga);
82
83     ui->progressBar_2->setValue(svetlo);
84
85     if(brojac == 0) {
86         series->clear();
87     }
88
89     series -> append(brojac, svetlo);
90
91     brojac =(brojac+1)%60;
92 }

```

Slika 6. Definicija funkcije *void senzori()*

Što se tiče definisanja *void on_pushButton_clicked(bool checked)*, potrebno je napomenuti da je parameter (*bool checked*) odabran da bi to dugme moglo da vrši dve radnje tj. da otvara i da zatvara ventil za vodu.

Funkcija *void on_pushButton_2_clicked ()* prebriše grafik pozivanjem metode *clear()* i brojač (1 minut) postavi na 0.

Void prikazVremena() je funkcija koja u gornjem desnom uglu GUI-a prikazuje trenutno vreme korišćenjem funkcije *currentTime()*.

Poslednje dve funkcije rade “zajedno” tj. nakon pozivanja *void on_pushButton_3_clicked()* funkcije pokreće se tajmer koji kad istekne aktivira funkciju *void count()*.

```
93
94 = void Dialog::on_pushButton_clicked(bool checked)
95 {
96 =     if(checked == true) {
97         digitalWrite(21, 1);
98     } else if (checked == false) {
99         digitalWrite(21, 0);
100     }
101 }
102
103 = void Dialog::on_pushButton_2_clicked()
104 {
105     series -> clear();
106     brojac = 0;
107 }
108
109 = void Dialog::prikazVremena()
110 {
111     QTime time = QTime::currentTime();
112     QString text = time.toString("hh : mm : ss");
113     ui->label_5->setText(text);
114 }
115
116 = void Dialog::on_pushButton_3_clicked()
117 {
118     minut = ui->spinBox->value();
119     val = minut*60000;
120     timer1 ->start(val);
121 }
122
123 = void Dialog::count()
124 {
125     digitalWrite(21, 1);
126     delay(2000);
127     digitalWrite(21, 0);
128 }
```

Slika 7. Definisanje funkcija

Unutar **main.cpp** fajla se samo doda doda linija `wiringPiSetup()` koja inicijalizuje `wiringPi` biblioteku.

```
1  #include "dialog.h"
2  #include <QApplication>
3
4  int main(int argc, char *argv[])
5  {
6      wiringPiSetup();
7
8      QApplication a(argc, argv);
9      Dialog w;
10     w.show();
11
12     return a.exec();
13 }
14
```

Slika 8. Izgled `main.cpp` fajla

Neophodno je objasniti i rad senzora koji su primenjeni prilikom izrade ovog projekta.

Senzor za merenje vlažnosti zemljišta se sastoji od dve sonde koje mere količinu vode tako što te dve sonde omogućavaju struji da prođe kroz zemlju, što daje vrednost otpora kako bi se izmerila vlažnost. Kada je voda prisutna, zemlja će provoditi više struje što znači da će otpornost biti manja. Suva zemlja slabo provodi struju, pa će samim tim otpornost biti veća i zemlja će zbog toga provoditi manje struje. Ovaj modul takođe sadrži potencijometar, koji postavlja *threshold* vrednost. Ta vrednost će se uporediti sa LM393 komparatorom. Izlazna LED će svetleti i smanjivaće se u skladu sa *threshold*-om.

Glavni deo foto senzora je fotootpornik čiji će otpor početi da se menja kad postoji neka svetlost tj. kad je intenzitet svetlosti sve jači. Što je veći intenzitet svetla, to je manja otpornost i obrnuto. Foto senzor koji je korišćen u ovom projektu (Slika 1.) je povezan na AIN2 ulaz YL-40 modula da bi vrednosti sa senzora bile kovertovane u digitalne kako bi ih RPi registrovao.

5V relay module funkcioniše na veoma jednostavan način, tako što ima 3 ulaza na terminalima (NC – normally close, NO – normally open, COM – Common) od kojih je u ovom projektu korišćen NO ulaz na koji se dovodi napajanje od 24V (jer elektromagnetni ventil radi na 24V) i COM sa kog se vodi + kraj ventila. Pritiskom na *pushButton* šalje se signal ka releju i dolazi do preklapanja NO i COM kontakata i tada ventil počinje da se napaja.

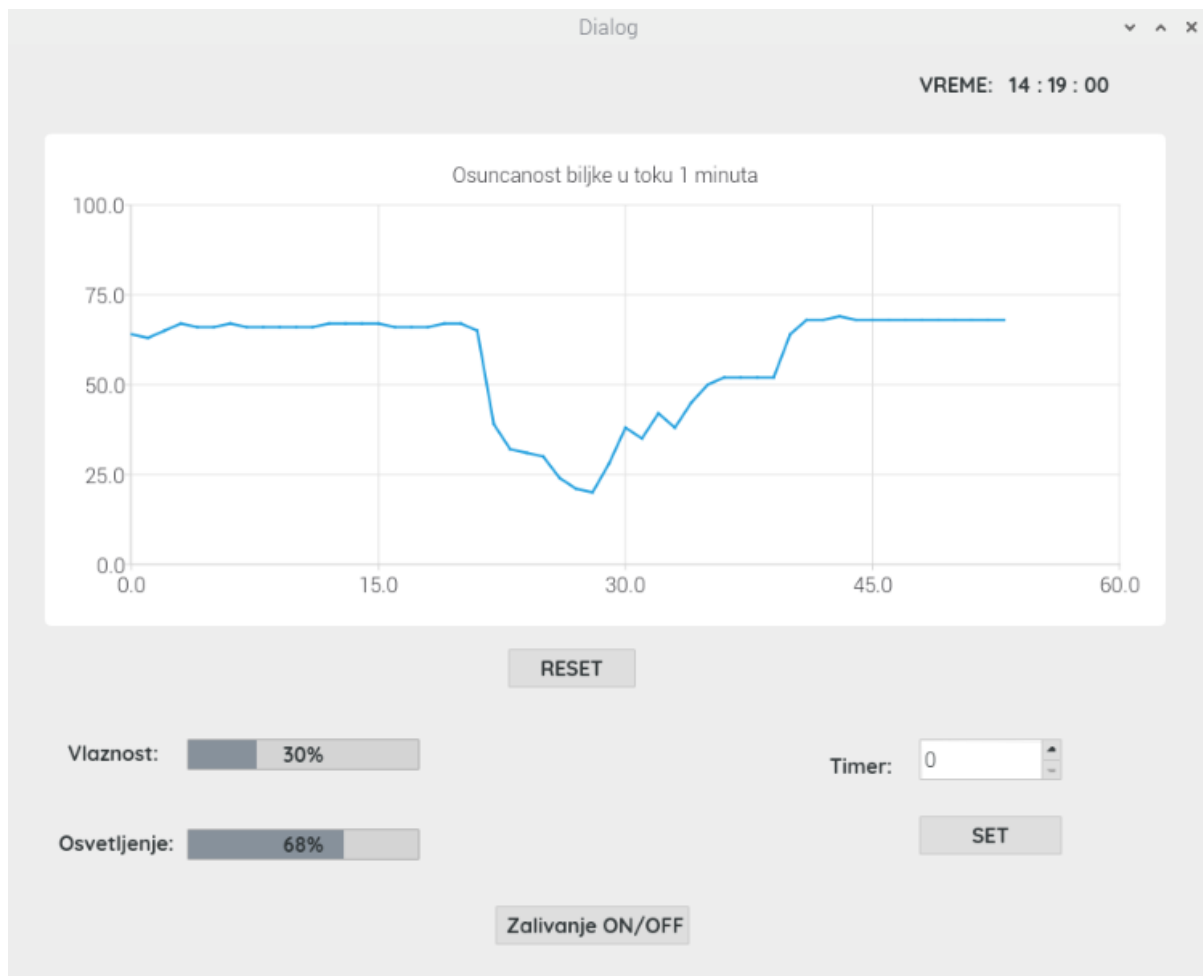
YL-40 pločica funkcioniše tako što je potrebno PCF8591 kolo povezati na I2C magistralu RPi-a. To znači da je potrebno povezati pinove SCL i SDA na odgovarajuće pinove na RPi-u, kao i VCC na 3,3V i GND na GND. Funkcija koju ovaj modul obavlja u ovom projektu je AD konverzija. Takođe treba napomenuti da ova pločica na sebi sadrži potencijometar, fotootpornik, temperaturni senzor itd...

3. Zaključak

Projekat je realizovan uspešno i ovakav koncept (konkretno ne ovaj projekat) može naći primenu i u većim sistemima za melioraciju. Za kraj su ostale slike rada kompletnog projekta.



Slika 9. Primena projekta



Slika 10. Izgled aplikacije