UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA MIKRORAČUNARSKU ELEKTRONIKU

# Protivpožarni alarm

Projekat iz predmeta Računarska Elektronika

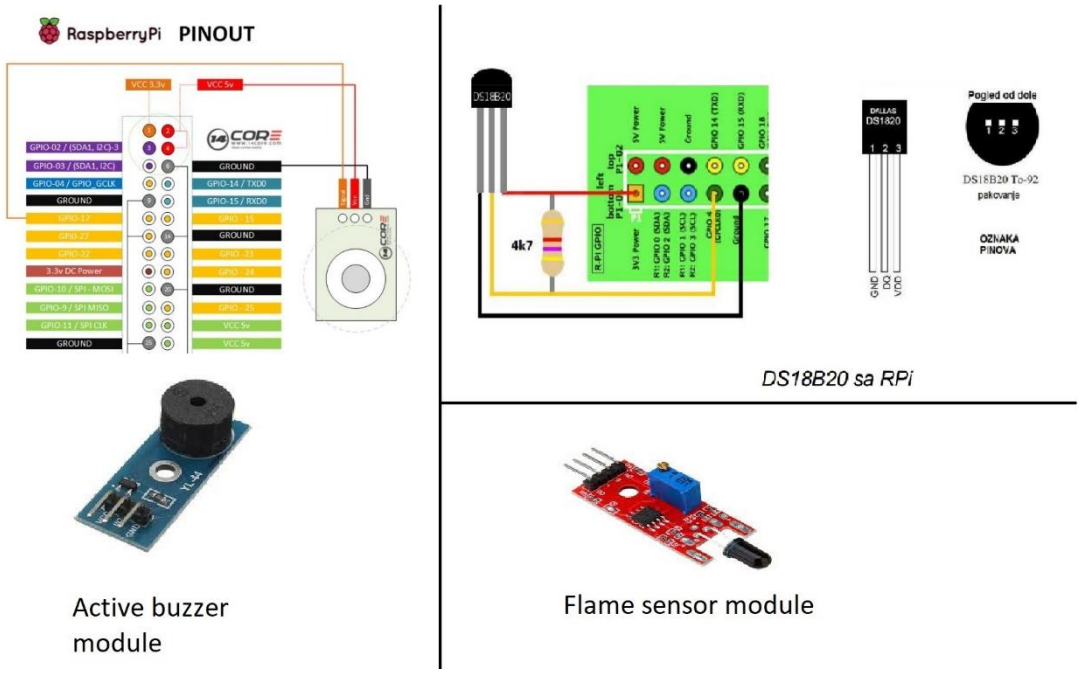
Student: Aleksa Mitrovčan EE106/2016
Mentor: Prof. Dr. Ivan Mezei

# 1. Opis zadatka

Cilj projekta je da se uz pomoću *RaspberryPi* uređaja vrši nadzor temperature i potencijalne pojave plamena ili požara, koje *RaspberryPi* može lako da detektuje pomoću temperaturnog senzora i flame senzora (IR Reciever). Ukoliko dodje do detekcije požara, active buzzer module će svojim zvukom obavestiti korisnika o mogućoj opasnosti.

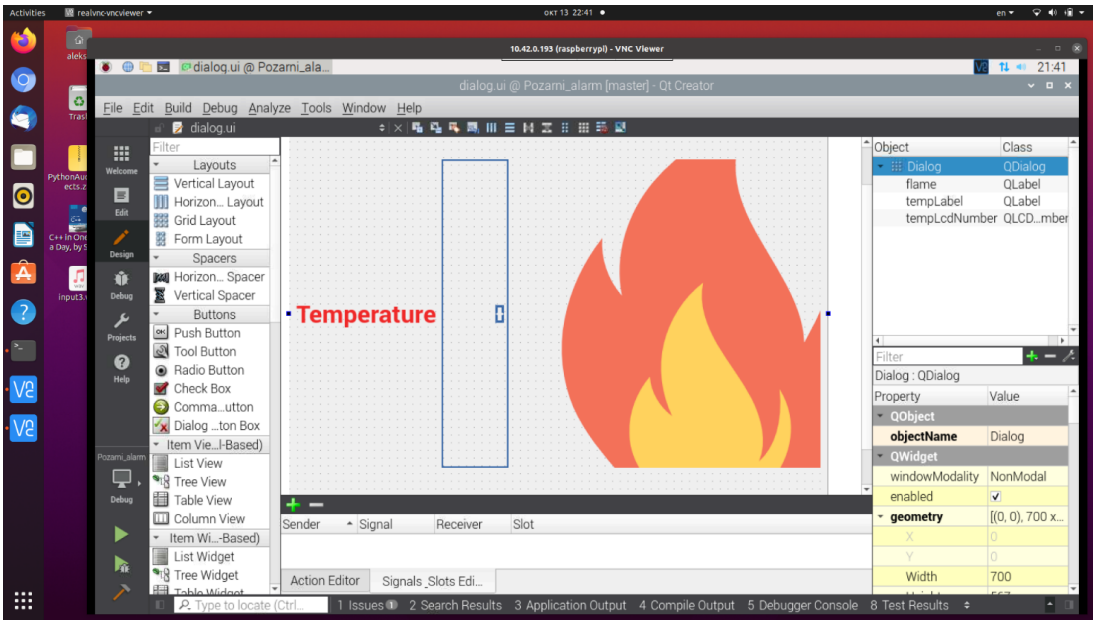
# 2. Opis konkretne realizacije

Na početku je potrebno povezati senzore sa *RaspberryPi* . Prvi modul koji se povezuje je temeperaturni sensor DS18B20, uputstva za povezivanje su na slici ispod. Zatim flame sensor module (IR Reciever) : 1.VC -> 3.3-5V ; 2.GND -> GND ; 3.DO(Digital outpu) ->GPIO pin. Na kraju povezujemo active buzzer module čije je povezivanje slično flame senzoru (slika ispod).




Slika 1. Senzori

Nakon povezivanja prelazi se na pravljenje GUI interfejsa i koda za projekat. Aplikacija sadrži LcdNumber display gde se prikazuje vrednost temperaturnog senzora, labela Tempperature i PixMap u kojem se nalazi slika vatre koja će iskočiti samo u slučaju kada je vatra detektovana.




Slika 2. Izgled GUI-a

Na samom početku, nakon kreiranja projekta potrebno je uključiti sve potrebne biblioteke koje će se koristiti u projektu. Prvo što treba da se uradi jeste dodavanje -lwiringPi biblioteke.

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

Slika 3. Dodavanje lwiringPi u .pro fajl

Ova linija koda služi za dodavanje *wiringPi* biblioteke tj. putanje ka toj biblioteci.

Sledeći korak jeste pisanje .hpp fajla i ubacivanje svih neophodnih biblioteka u projekat.

```
#ifndef DIALOG_H
#define DIALOG_H

#include <QDialog>
#include <QApplication>
#include <QWidget>
#include <wiringPi.h>
#include <stdio.h>
#include <QtGlobal>
#include <stdlib.h>
#include <QDebug>
#include <QFile>
#include <stdlib.h>
#include <iostream>
#include <string>
#include <QTimer>
#include <QMessageBox>
#include <QString>
#include <QTextStream>
#include <QImage>
#include <QPixmap>
#include <QLabel>

using namespace std;




namespace Ui {
class Dialog;
}

class Dialog : public QDialog
{
    Q_OBJECT

public:
    explicit Dialog(QWidget *parent = nullpt
    ~Dialog();
    FILE *ft;




public slots:
  double getTemp(void);
  void buzz();


private:
    Ui::Dialog *ui;
    QString tekst;
    QTimer *myTimer1;
    QTimer *myTimer2;




};

#endif // DIALOG_H
```

Slika 4. .hpp fajl

U *.hpp* fajlu su deklarisane funkcije:

1. **double getTemp(void) –** funkcija koja čita vrednost temperaturnog senzora
2. **void buzz()** – aktivira active buzzer module.

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

```
30
31  double Dialog::getTemp(void){
32
33      FILE *ft;
34      char tekst[100];
35      ft=fopen("/sys/bus/w1/devices/28-00000a41dec3/w1_slave","r");
36
37      int i=0;
38      for(i=0;i<22;i++) //samo temperatura
39      fscanf(ft,"%s", tekst);
40      fclose(ft);
41      for(i=0;i<10;i++) tekst[i]=tekst[i+2];
42
43      double temp=atoi(tekst); //prebaci u double
44      double tem=temp/1000;
45
46      qDebug() << tem;
47      ui->tempLcdNumber->display(tem);
48
49
50      return tem;
51
52
53      }
54
```

Slika 6. getTemp()

Funkcija getTemp() čita iz putanje "/sys/bus/w1/devices/28-00000a41dec3/w1_slave" tekstualnu datoteku i iz nje izvlači samo vrednost temperature, zatim prebacuje tu vrednost iz karaktera u cifru (double) pomoću atoi funkcije (43. linija koda). Pošto je u datoteci temperatura zabelezena kao petocifreni broj npr. (21453) što je ustvari 21.453°C, tu vrednost delimo sa 1000 da bi dobili realnu vrednost temperature i tu konačnu vrednost ispisujemo na LcdNumber displej.

```
52
53  void Dialog::buzz() {
54
55      pinMode(3, OUTPUT);
56      int j;
57      if(getTemp()>25.0) {
58
59          if(wiringPiISR(FlamePin, INT_EDGE_FALLING, &myISR)) {
60                  printf("setup interrupt failed !\n");
61
62          }
63           QPixmap pix(":/slike/flame.png");
64           ui->flame->setPixmap(pix.scaled(200,200,Qt::KeepAspectRatio));
65          for(j = 0;j<20;j++)
66          {
67              digitalWrite(3, HIGH);
68              delay(100);
69              digitalWrite(3, LOW);
70              delay(100);
71          }
72
73
74
75      }
76
77  }
```

Slika 7. Void buzz()

Buzz() reaguje samo ukoliko je temperatura veca od odredjene vrednosti (u ovom slučaju iz bezbednosnih razloga je to samo 25.0) i ukoliko IR Reciever (Flame sensor) detektuje vatru. Ukoliko detektuje, iskočiće slika vatre (Qpixmap), a buzzer ce da pišti svaki put na 1000ms.

```cpp
#include "dialog.h"
#include "ui_dialog.h"
#include <string>
#include <iostream>

using namespace std;




Dialog::Dialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::Dialog)
{
    ui->setupUi(this);

    myTimer1 = new QTimer(this);
    connect(myTimer1, SIGNAL(timeout()), this, SLOT(getTemp()));    //povezi tajmer sa getTemp() funkcijom
    myTimer1->start(1000);                                          //tajmer svake sekunde
    myTimer2 = new QTimer(this);
    connect(myTimer2, SIGNAL(timeout()), this, SLOT(buzz()));       //povezi tajmer sa buzzer modulom
    myTimer2->start(1000);
    pinMode(3, OUTPUT);                                             //buzzer
    pinMode(0, OUTPUT);                                             //flame pin
    QPixmap pix(":/slike/flame.png");                               //slika


```

Slika 8. Dialog

U Dialog-u smo inicijalizovali tajmere koje smo povezali sa getTemp() i buzz() funkcijama, tako da se tajmer izvršava svake sekunde. Inicijalizovali smo pinove za flame sensor module i buzz module, takodje i sliku vatre koju smo prethodno dodali u Resource Files i u folder gde se nalazi projekat.

Unutar ***main.cpp*** se nalazi *wiringPiSetup()* koja inicijalizuje *wiringPi biblioteku.* Takođe smo fiksirali veličinu prozora na kom se ispisuje temperatura.

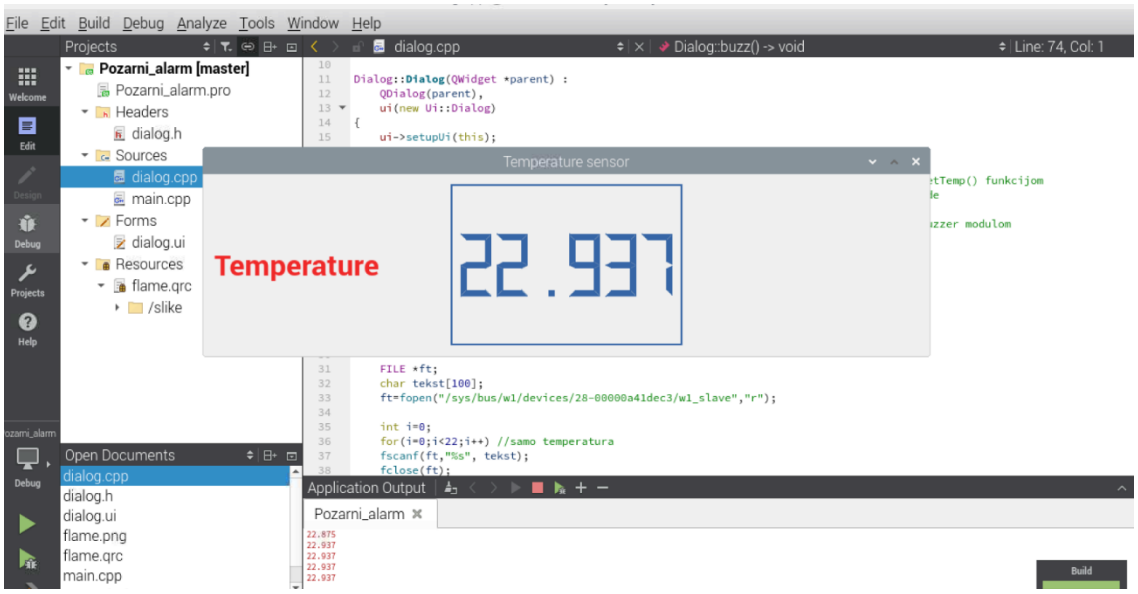
main.cpp

```cpp
#include "dialog.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    wiringPiSetup();
    Dialog w;
    w.setFixedSize(800, 200);
    w.setWindowTitle("Temperature sensor");
    w.show();

    return a.exec();
}

```

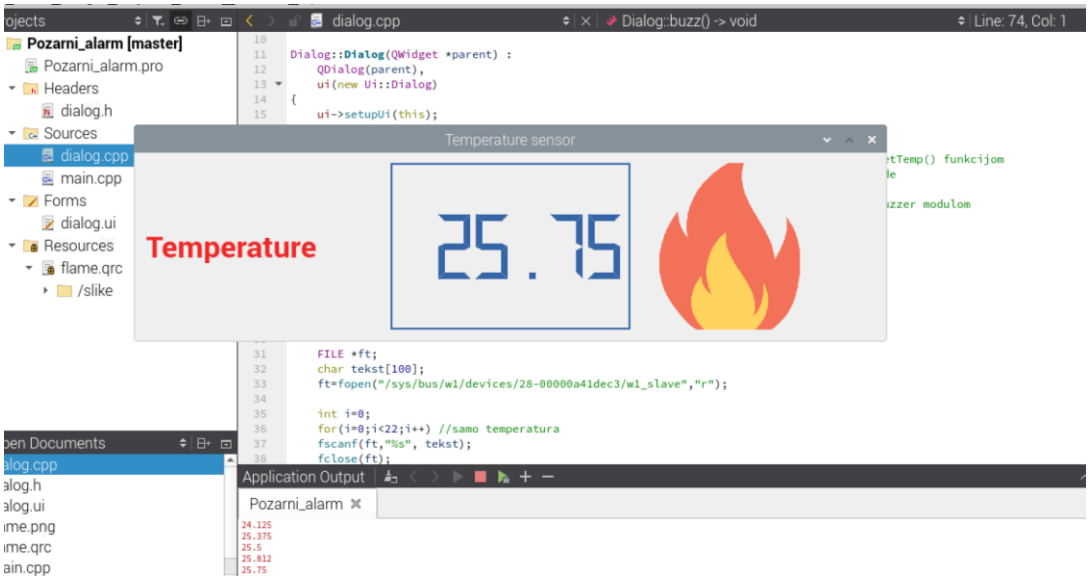
Slika 9. Izgled *main.cpp* fajla

# 3. Zaključak

Aplikacija je uspešno realizovana, ispod se nalaze slike kada nije detektovana vatra i kada jeste, u priloženom snimku se može čuti i pištanje buzzer module-a:




Slika 10. Merenje temperature (nije detektovana vatra)




Slika 11. Detektovana je vatra preko Flame sensora i temperatura je naglo porasla