

# SISTEM DE MONITORIZARE SENZORI UTILIZAND ANDROID

Alexandru MOLDOVAN, Iosif ZAHARIA

Facultatea de Inginerie Hunedoara, Universitatea Politehnica Timișoara, România

**Rezumat:** Lucrarea prezintă un sistem de achiziție în timp real a informațiilor provenite de la un grup de senzori conectați la o placă de dezvoltare Arduino. Datele măsurate se transmit la distanță prin intermediul unui modul bluetooth către un telefon mobil. Aplicația instalată pe telefonul mobil permite vizualizarea variației informațiilor pe grafice.

**Cuvinte cheie:** Arduino, senzor lumină, senzor gaz, senzor umiditate și temperatură, bluetooth.

## Introducere

Evoluția tehnologiei din ultimul timp permite extinderea și dezvoltarea aplicațiilor pe dispozitive mobile care permite comunicarea wireless cu alte echipamente. Aplicația de față permite utilizatorului să monitorizeze în timp real următorii parametri: temperatură, umiditate, nivel concentrație gaz, iluminare. Aplicația de achiziție este realizată în Arduino și încărcată în microcontrolerul plăcii de dezvoltare. Pentru monitorizarea acestor parametri se folosește o aplicație customizată pe telefonul mobil.

## 1. Descrierea părții hardware a aplicației

Schema bloc a aplicației hardware este prezentată în Figura 1. În continuare sunt prezentate caracteristicile tehnice și modul de funcționare a componentelor aplicației. Senzorii de temperatură, umiditate, gaz și iluminare transmit datele wireless unei aplicații pe Android utilizând un modul bluetooth.

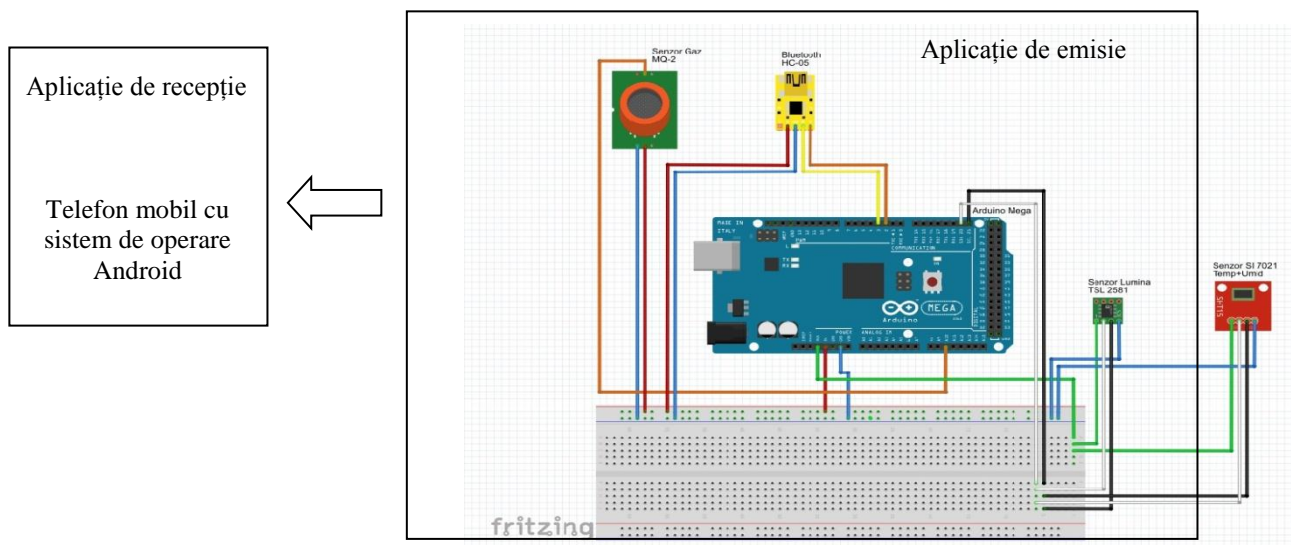


Figura 1. Schema bloc a aplicației hardware.

### 1.1. Placa de dezvoltare Arduino

Arduino permite funcționarea aplicațiilor în timp real datorită microcontrolerului conectat pe placă. Placa de dezvoltare conține pini analogici de intrare și pini digitali ce pot fi configurați de intrare/ieșire. Placa de dezvoltare Arduino Mega dispune de pini specifici magistralelor seriale I2C și UART.



Figura 2. Placă de dezvoltare Arduino Mega.

### 1.2. Senzor gaz MQ2

Modulul prezentat în Figura 3 este folosit pentru a detecta scurgerile de gaze în încăperi mici sau mari și reprezintă o metodă de precauție pentru incendii sau pentru intoxicații.

Senzorul are o sensibilitate ridicată și principalele gaze pe care le vizează sunt GPL-ul, izobutan, propan, metan, alcool, hidrogen și fum.

Senzorul dispune de un comparator, astfel ca se pot citi date analogice în timp real sau se poate afla dacă concentrația de gaz a depășit o anumită limită.

#### Specificatii tehnice:

- Tensiune: 5V;
- Curent: 150mA;
- Rezistența heater: 33R;
- Temperatura de lucru: -20°C - 50°C;
- Output digital sau analogic.

### 1.3. Senzor de temperatură și umiditate Sparkfun SI7021

În vederea măsurării temperaturii ambiante și a umidității se utilizează senzorul Sparkfun SI7021 (Figura 4) care comunică cu sistemul prin intermediul magistralei seriale I2C. Pentru achiziția senzorului se folosește o bibliotecă specifică apelată în codul sursă al aplicației.

Principalele caracteristici ale magistralei I2C sunt:

- este o magistrală serială, bidirecțională, alcătuită doar din 2 linii: SDA (Serial Data) și SCL (Serial Clock);
- un modul compatibil I<sup>2</sup>C are înglobată o interfață care îi permite conectarea directă la cele 2 linii ale interfeței; astfel, la nivelul magistralei nu vor mai fi necesare nici un fel de circuite pentru funcționarea corectă a sistemului;
- fiecare modul conectat la magistrală este adresabil prin program printr-o singură adresă;
- modulele pot fi transmițătoare sau receptoare;

### 1.4. Senzor de iluminare

Aplicația are un senzor de iluminare (Fig. 5) care folosește circuitul integrat TSL2581. Dispozitivul comunică cu sistemul prin magistrala I2C.

### 1.5. Modulul de comunicație bluetooth HC-05

Modulul HC-05 Bluetooth (Fig. 6) e proiectat pentru configurarea fără fir a conexiunii. Modulul Bluetooth cu port serial este complet calificat cu tehnologia Bluetooth V2.0 + EDR (viteză îmbunătățită a datelor) de 3Mbps și de modulație cu transmițător radio de 2,4 GHz și bandă de bază.

Acest modul comunică prin intermediul interfeței USART la o rată baud de 9600. De asemenea se pot configura valorile predefinite, folosind comenzi specifice.



Fig. 3



Fig. 4



Fig. 5



Fig. 6

## 2. Descrierea părții software a aplicației (partea de emisie)

Codul sursă al aplicației de emisie este prezentat în continuare.

```
// BT module Arduino
// GND ----- GND
// VCC ----- 5V
// TX-0 ----- pin2
// RX-1 ----- pin3

// Iluminare TSL2581
#include <Wire.h>
#include "TSL2581.h"
#include <SparkFun_S17021_Breakout_Library.h>
#include "SparkFun_S17021_Breakout_Library.h"
WaveShare_TSL2581 tsl = WaveShare_TSL2581();
int TSL2581_INT = 13;

// Bluetooth HC-06
#include <SoftwareSerial.h>
//SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);
#define bluetooth Serial3

// Gaz MQ-2
int alarma_gaz = 4; //alarma la pinul 4
int gaz = A0; //intrare semnal analogic gaz

// Temp+hum
float humidity = 0;
float tempf = 0;
float tempC = 0;
Weather sensor;

// Transmisie bluetooth Roboremo
int on = 0; //buton oprit / pornit
char cmd[100];
int cmdIndex;

boolean cmdStartsWith(char *st) {
  for(int i=0; ; i++) {
    if(st[i]==0) return true;
    if(cmd[i]==0) return false;
    if(cmd[i]!=st[i]) return false;
  }
  return false;
}

void exeCmd() {
  if( cmdStartsWith("on") ) on = 1;
  if( cmdStartsWith("off") ) on = 0;
}

void read_id(void)
{
  int id;
  int a;
  id = tsl.TSL2581_Read_ID();
  a = id & 0xf0; //The lower four bits are the silicon version number
  if (!(a == 144)) //ID = 90H = 144D
  {
    Serial.println("false ");
  }
  else {
    Serial.print("I2C DEV is working ,id = ");
    Serial.println(id);
    delay(500);
  }
}

void Read_gpio_interrupt(uint16_t mindata, uint16_t maxdata)
{
  tsl.SET_Interrupt_Threshold(mindata, maxdata);
  int val = digitalRead(TSL2581_INT);
  if (val == 1)
  {
    //Serial.print("interrupt = 1 \n");
  }
  else {
    // Serial.print("interrupt = 0 \n");
    tsl.Reload_register();
  }
}

void setup(void)
{
  delay(500); // wait for bluetooth module to start
  Serial.begin(9600);
  bluetooth.begin(9600);

  //Initialize the I2C sensors and ping them
  sensor.begin();
  pinMode(alarma_gaz, OUTPUT);

  cmdIndex = 0;
  Serial.begin(9600);
  Wire.begin(); //I2c config
  pinMode(TSL2581_INT, INPUT); // sets the digital pin 7 as input
  pinMode(alarma_gaz, OUTPUT);

  pinMode(alarma_gaz, OUTPUT);
  pinMode(gaz, INPUT);

  read_id();

  /* Setup the sensor power on */
  tsl.TSL2581_power_on();
  delay(2000);
  // /* Setup the sensor gain and integration time */
  tsl.TSL2581_config();
}

void loop(void)
{
  while(bluetooth.available() ) {

    char c = (char)bluetooth.read();

    if(c=='\n') {
      cmd[cmdIndex] = 0;
      exeCmd(); // execute the command
      cmdIndex = 0; // reset the cmdIndex
    }
    else {
      cmd[cmdIndex] = c;
      if(cmdIndex<99) cmdIndex++;
    }
  }

  if(on) {

    int gaz = analogRead(A0);
    if (gaz > 400)
    {
      digitalWrite(alarma_gaz, HIGH);
    }
    else
    {
      digitalWrite(alarma_gaz, LOW);
    }
    humidity = sensor.getRH();
    tempf = sensor.getTempF();
    tempC = (tempf - 32) / 1.8;

    int nivel_gaz = analogRead(gaz);

    int nivel_gaz = analogRead(gaz);

    unsigned long Lux;
    tsl.TSL2581_Read_Channel();
    Lux = tsl.calculateLux(2, NOM_INTEG_CYCLE);

    Read_gpio_interrupt(2000, 50000);
    delay(50);

    if (nivel_gaz > 400)
    {
      digitalWrite(alarma_gaz, HIGH);
    }
    else
    {
      digitalWrite(alarma_gaz, LOW);
    }
    //delay(500);
    String st = (String)"v " + gaz + "\n";
    Serial.println( st );
    bluetooth.println( st );

    String temperatura = (String)"t " + tempC + "\n";
    Serial.print("Temp:");
    Serial.print( tempC);
    Serial.print("C ");
    bluetooth.println( temperatura );

    String umiditate = (String)"h " + humidity + "\n";
    Serial.print("Umiditate:");
    Serial.print( humidity );
    Serial.println("%");
    bluetooth.println( umiditate );

    String ilum = (String)"i " + Lux + "\n";
    Serial.print("Iluminare:");
    Serial.print( Lux );
    Serial.println("Lux");
    bluetooth.println( ilum );
  }
}
```

## 3. Descrierea părții software a aplicației (partea de recepție)

Roboremo este o aplicație personalizabilă pentru comandă și control wireless. Ea se poate instala pe orice dispozitiv mobil cu sistem de operare Android pentru comunicații wireless. Conexiunea se poate face prin Bluetooth, WiFi, USB, Internet mobil.

Se poate utiliza pentru a controla de exemplu o mașină, un avion, o navă, o dronă, de asemenea, pentru proiecte de automatizare acasă.

Cu ajutorul Roboremo se pot construi 8 interfețe grafice separate adăugând butoane, glisoare, LED-uri, grafice, accelerometre, etc.

Aici Roboremo (Figura 7) este personalizat pentru a recepționa în timp real pe grafice variația parametrilor concentrație gaz, iluminare, temperatură și umiditate.

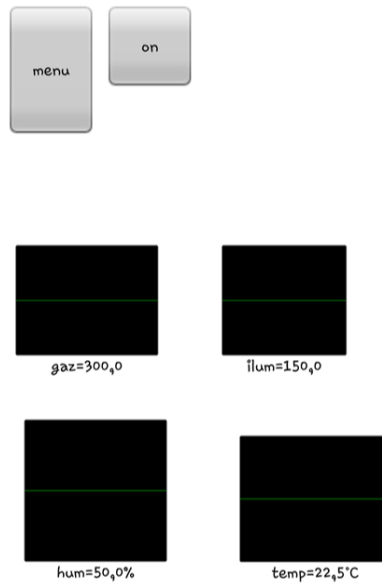


Fig. 7. Interfața aplicației Roboremo.

## Concluzii

Aplicația este extrem de utilă pentru monitorizarea în timp real a unor parametrii proveniți de la senzori. Monitorizarea pe dispozitiv mobil este extrem de utilă utilizatorului.

## Bibliografie

1. <https://create.arduino.cc/projecthub/Aritro/smoke-detection-using-mq-2-gas-sensor-79c54a>
2. <https://learn.adafruit.com/adafruit-si7021-temperature-plus-humidity-sensor/arduino-code>
3. <https://www.mouser.se/ds/2/588/TSL2581%20TSL2583-A-553251.pdf>
4. <http://ww1.microchip.com/downloads/en/devicedoc/usart.pdf>