**Enviroment analysis**

(sistem intelligent de analiza si monitorizare a mediului de lucru)

# Descrierea ideii, justificarea atractivitatii, simplitate si fazabilitate:

Proiectul va avea in vedere monitorizarea nivelului de temperatura, umiditate si gaze a unui mediu de lucru in privinta alertarii asupra unor medii nepotrivite. Structura proiectului consta intr-o componenta care va avea montata senzorii de mediu si un buzzer de alerta.

Starile de functionare ale dispozitivului vor fi:

* Achizitie date
* Trimite date catre baza de date
* Compara date cu niste praguri limita
* Alertare prin buzzer
* Detectie persoane prin PIR

Dispozitivul odata alimentat acesta va prealua date din mediul inconjurator prin senzorii montati si va trimite datele catre o baza de date online, InfluxDB. Partea de monitorizare va fi creata pe o pagina web printr-un framework de Javascript. Pagina web va afisa datele sub forma de grafice si va afisa alerta atunci cand datele trec de praguri limita. Placuta Arduino, dupa ce trimite datele, in cazul depasirii pragurilor alese va activa buzzer-ul pentru a alerta oamenii.

Lucrarea va aduce atentie aspura senzorului de gaz, care v-a prelua date de mediu si le va evalua.

Table of Contents

[Descrierea ideii, justificarea atractivitatii, simplitate si fazabilitate: 1](#_Toc92637976)

[Consideratii teoretice: 3](#_Toc92637977)

[Functionare: 3](#_Toc92637978)

[Capitolele lucrarii: 3](#_Toc92637979)

[Obiectivele lucrarii de laborator ( ce pot invata studentii din aceasta lucrare ): 4](#_Toc92637980)

[Legatura cu disciplina SIO (Sisteme de Intrare-Iesire): 4](#_Toc92637981)

[Materiale didactice necesare: 4](#_Toc92637982)

[Prezentarea componentelor: 6](#_Toc92637983)

[1. MQ-2 6](#_Toc92637984)

[2. Led 6](#_Toc92637985)

[Partea hardware: 7](#_Toc92637986)

[Schema electrica – proiect 1: 7](#_Toc92637987)

[Schema electrica – proiect 2: 8](#_Toc92637988)

[Resurse interne din microcontroller pentru care se scrie un soft propriu: 9](#_Toc92637989)

[**Partea software:** 11](#_Toc92637990)

[Schema logica (proiect 1) Schema logica (proiect 2) 11](#_Toc92637991)

[Impartirea echipei: 12](#_Toc92637992)

[Componente necesare, pret si sursa de aprovizionare 13](#_Toc92637993)

[Anexa 13](#_Toc92637994)

[Cod proiect 1: 13](#_Toc92637995)

[Cod proiect 2: 21](#_Toc92637996)

# Consideratii teoretice:

Senzorul de gaz, [MQ-2](https://www.pololu.com/file/0J309/MQ2.pdf) (click pentru a intra pe datasheet) :



# Functionare:

* Senzorul, alimentat, isi va incalzii rezistenta
* Atunci cand un gaz este prezent in jurul senzorului, conductivitatea acestuia creste cu cresterea concentratiei de gaz.

Avantaje:

* Circuit simplu
* Ieftin si eficient
* Sensibilitate crescuta la GPL, propan si hydrogen

Utilizari obisnuite:

* Detector de gaz portabil
* Detector de gaz pentru locuinte

# Capitolele lucrarii:

1. ARDUINO, ce reprezinta? Beneficiul lucrului cu un microcontroller? De ce ARDUINO?
2. Documentare ARDUINO, pini GPIO, intrari/iesiri analogice/digitale.
3. Configurarea controllerului si a senzorilor. Crearea unui demo pentru a valida ideea.
4. Utilizarea modulului WIFI
5. Implementarea modulelor de interactiune cu senzorii.
6. Implementarea modulului de trimitere a datelor catre baza de date.
7. Evaluarea datelor achizitionate si activarea buzzerului in cazul unei probleme.

# Obiectivele lucrarii de laborator ( ce pot invata studentii din aceasta lucrare ):

* Introducerea in mediul ARDUINO, microcontroller, pini, module ARDUINO
* Lucrul cu limbajul si mediu de programare ARDUINO IDE
* Achizitie de date de la senzorii montati
* Interactiunea cu un proiect real si util pentru perioada contemporana
* Concepte de fiabilitate si fezabilitate, cat si corectitudinea unui cod sursa pentru a evita diferite erori
* Efectuarea unui proiect amplu care presupune intelegerea schemelor bloc a fiecarui modul si imperecherea acestora intr-un mod util

# Legatura cu disciplina SIO (Sisteme de Intrare-Iesire):

Proiectul va prezenta utilizarea modulelor de intrare a doi senzori de precizie, DHT22, MQ-2, si un BUZZER, ca modul de iesire, de alerta care va fi actionat atunci cand datele preluate din mediu vor trece peste anumite praguri. Modulul WIFI va prelua datele si le va trimite catre o baza de date Online.

# Materiale didactice necesare:

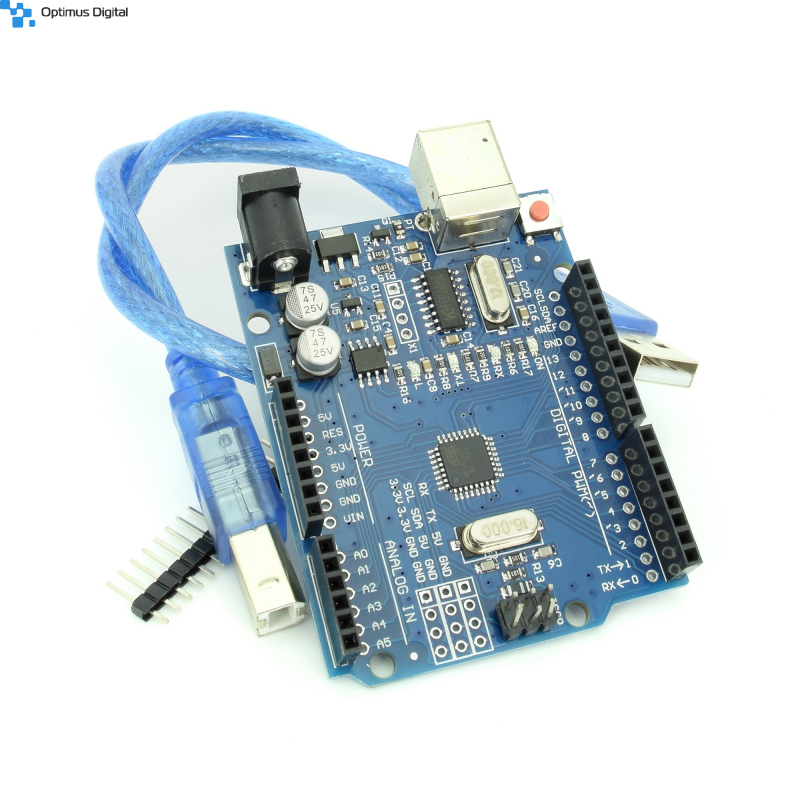
* Proiectul 1 (proiectul de masurarea mediului):
  + WeMOS D1 Mini
  + MQ-2
  + DHT-11
  + Buzzer
  + Pir

Proiectul 2 (proiectul studentilor):

* + Arduino Uno
  + MQ-2
  + 3 Led

Pentru realizarea acestui proiect se va utiliza ca mediu de dezvoltarea hardware placa Arduino Uno R3, iar pentru partea software, aplicatia Arduino IDE, in care vom scrie cod C/C++. Acest software este disponibil in urmatorul link: <https://www.arduino.cc/en/software>

Pentru inceput avem nevoie de placuta Arduino Uno si cablul de programare a acestuia.



Placuta se conecteaza la Laptop/PC prin cablu pentru a putea fi alimentata si pentru a o putea programa. Alimentarea insasi nu necesita conectarea la Laptop/PC, poate fi alimentat si printr-un alimentator de telefon cu socket USB.

# Prezentarea componentelor:

## MQ-2

* 1. Specificatii:
     1. Tensiune alimentare - 5V
     2. Rezistenta de load - 20KΩ
     3. Rezistenta senzorului - 33Ω ± 5%
     4. Concentratie detectabila – 200 – 10k ppm
  2. Poza



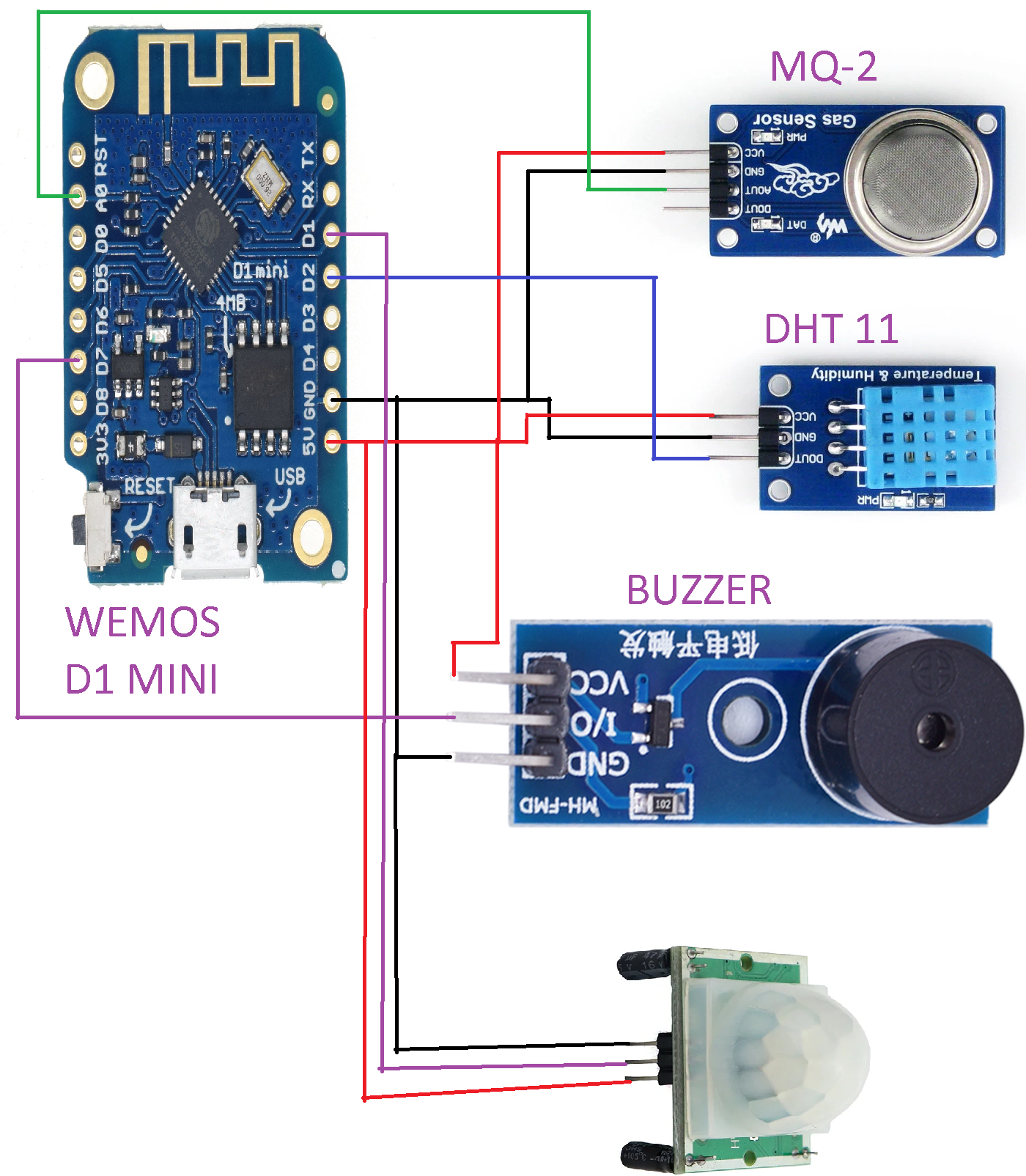
## Led

* 1. Specificatii
     1. Tensiune alimentare – 5 V
     2. Curent – 0.25 A
     3. Utilizarea unei rezistente de 220Ω pentru alimentare
  2. Poza

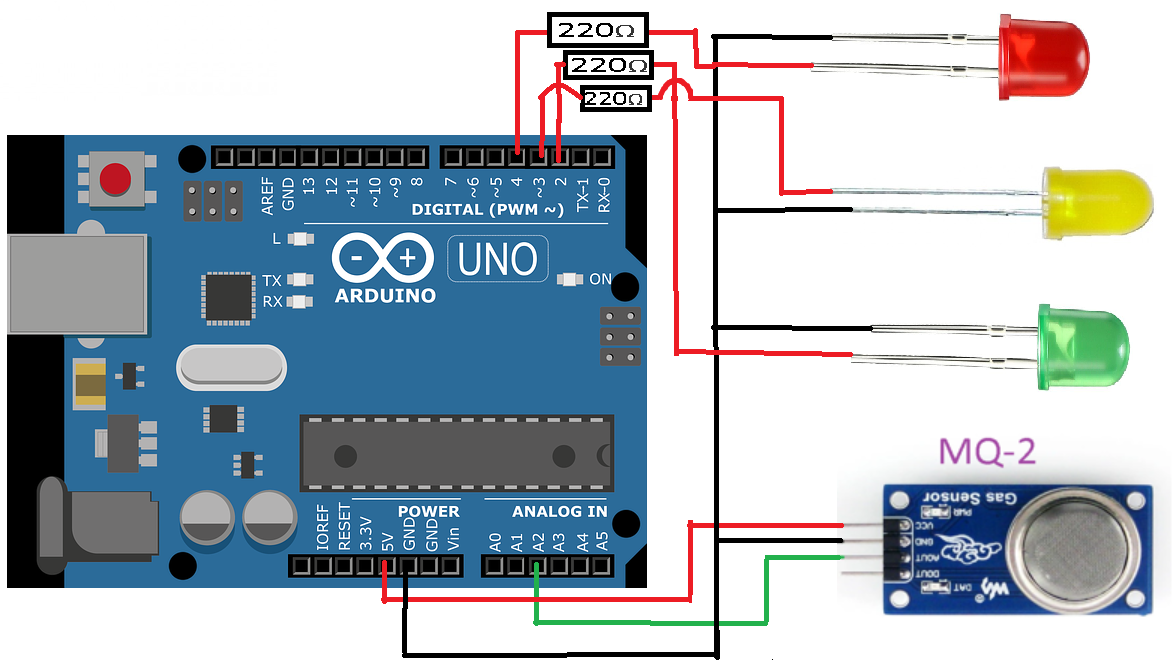


# Partea hardware:

## Schema electrica – proiect 1:

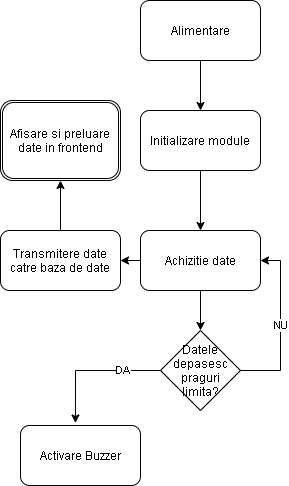


## Schema electrica – proiect 2:

****

# **Scheme logice:**

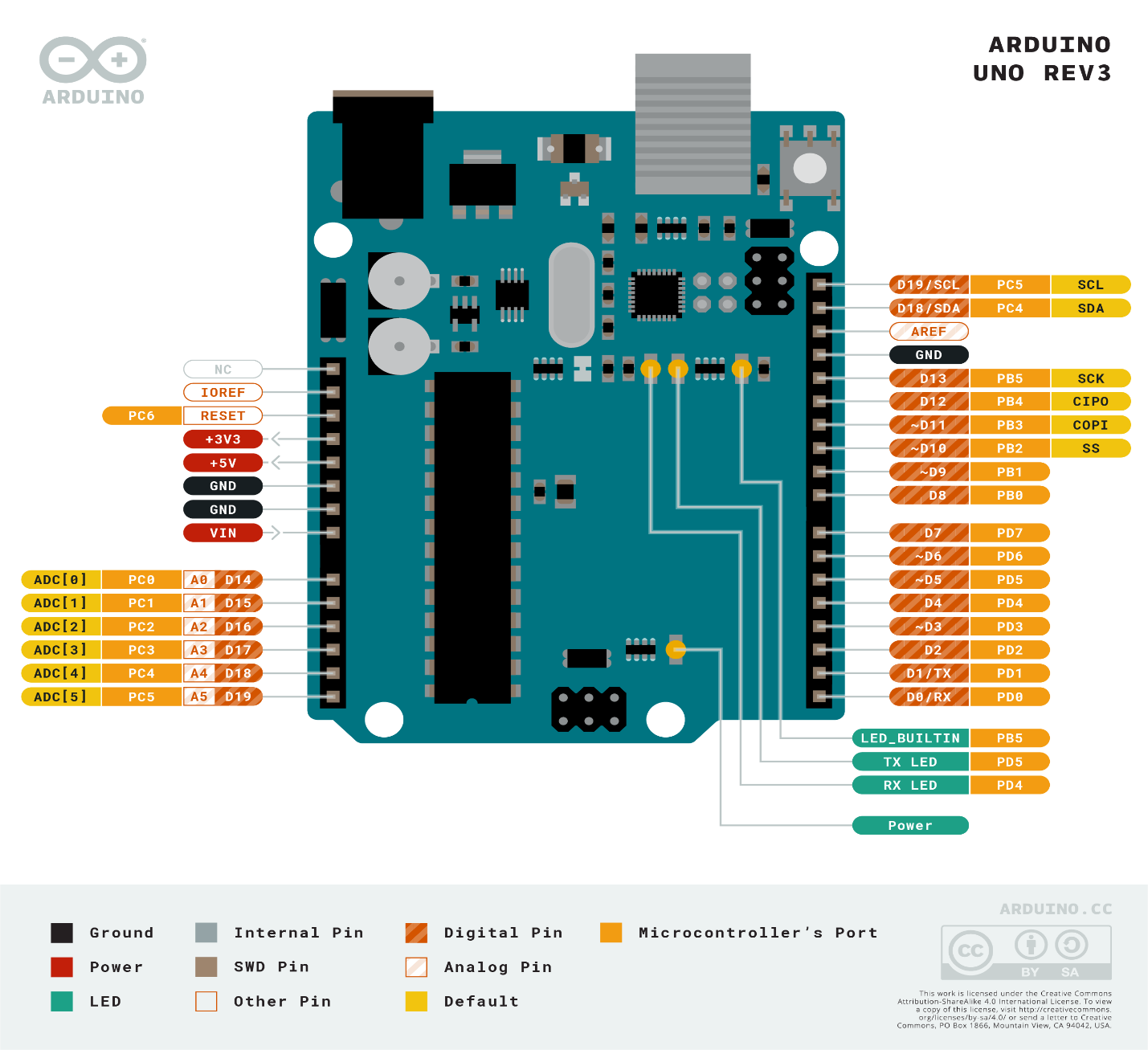
# Schema logica (proiect 1)Schema logica (proiect 2)

****

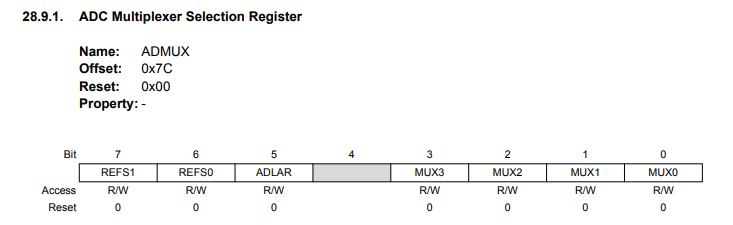
# Resurse interne din microcontroller pentru care se scrie un soft propriu:

Citirea dateor de la un pin analogic care va fi conectat la senzorul MQ-2.

Prezentarea pinilor Arduino.



Partea de pe placa care ne va interesa va fi partea cu pinii analogici.



Registrul ADMUX va avea nevoie de combinatie de biti de pe pozitiile 0-3 pentru a semnala ce porturi analogice sunt legate la convertorul Analog-Digital din cadrul microprocesorului.

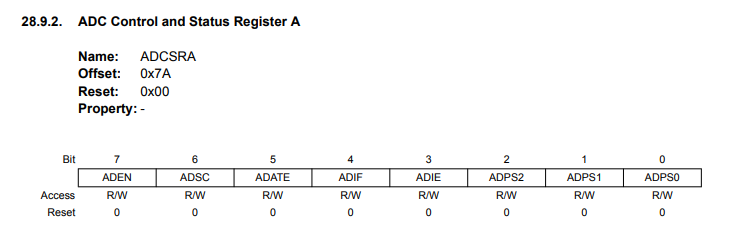
Avand in vedere ca senzorul MQ-2 este legat la pinul A2, ADMUX va primii valoarea 1 pe pozitia 1.

ADMUX |= (1<<MUX1);

De asemenea, este nevoie sa punem si valoarea 1 pe pozitia 6 pentru a active tensiunea referentiala interna pentru procesarea semnalului analogic.

ADMUX |= (1<<REFS0);

In urmatoarea parte ne vom concentra asupra registrului ADCSRA.



Acest registru va fi manipulat in felul urmator:

1. Pe pozitia 3 se va pune bitul 1 pentru a activa intreruperea externa de la finalul conversiei din analogic in digital. ADCSRA |= (1<<3);
2. Pe pozitiile 0-2 se vor pune valoarea 1 pentru a seta prescaler-ul pentru a scadea frecventa ceasului de input pentru a citii si convertii correct. ADCSRA |= 0b00000111;
3. In final se va pune valoarea 1 pe pozitia 7 pentru a activa convertorul AD.   
   ADCSRA |= (1<<7);

# Impartirea echipei:

-Popa Lorena: Crearea mediului de vizualizare a datelor

-Dinu Vladut: Conectare senzori si punerea in functiune a sistemului din cadrul proiectului 1 + Documentatie

-Botsch Ovidiu: Trimiterea datelor catre baza de date

-Cismaru Florin: Crearea serviciilor de preluare a datelor din baza de date

-Cozmescu Daniel: Conectare senzori si punerea in functiune a sistemului din cadrul proiectului 2 + implementarea senzorului PIR in cadrul proiectului 1

# Componente necesare, pret si sursa de aprovizionare pentru proiectul 1

|  |  |  |
| --- | --- | --- |
| SENZOR | PRET | LINK |
| Senzor de temperature si umiditate DHT11 | 6.87 Lei | <https://www.optimusdigital.ro/en/temperature-sensors/99-dht11-temperature-sensor-module.html> |
| Modul Senzor Gaz MQ-2 | 12.49 Lei | https://www.optimusdigital.ro/ro/senzori-de-gaze/107-modul-senzor-gas-mq-2.html?search\_query=Mq2&results=1 |
| Buzzer 5/3.3 V | 1.95 Lei | https://www.optimusdigital.ro/ro/audio-buzzere/634-buzzer-pasiv-de-5-v.html?search\_query=buzzer&results=55 |
| WeMos D1 Mini | 33.32 Lei | <https://www.emag.ro/placa-de-dezvoltare-wemos-nodemcu-d1-mini-esp8266-ai022-s62/pd/DN83KQBBM/> |
| Total | 54.63 Lei |

# Anexa

## Cod proiect 1:

#include <DHT.h>

// #include "dht11.h"

#include <ESP8266WiFiMulti.h>

ESP8266WiFiMulti wifiMulti;

#include <InfluxDbClient.h>

#include <InfluxDbCloud.h>

#define INFLUXDB\_URL "https://eu-central-1-1.aws.cloud2.influxdata.com"

// InfluxDB v2 server or cloud API token (Use: InfluxDB UI -> Data -> API Tokens -> <select token>)

#define INFLUXDB\_TOKEN "55trP1i-mBCyoUNGlbLwiRuz-dZ4HTB368xsU\_j-JoxKEbxSTxhUhcATze2TResaeP1dCiSegd3yH1QgVgdYCw=="

// InfluxDB v2 organization id (Use: InfluxDB UI -> User -> About -> Common Ids )

#define INFLUXDB\_ORG "arduinosio2021@gmail.com"

// InfluxDB v2 bucket name (Use: InfluxDB UI -> Data -> Buckets)

#define INFLUXDB\_BUCKET "ArduinoESP"

#define TZ\_INFO "CET-1CEST,M3.5.0,M10.5.0/3"

#define DEVICE "ESP8266"

// Replace with your network credentials

const char\* ssid = "Kid";

const char\* password = "aaaabbbb";

// InfluxDB client instance with preconfigured InfluxCloud certificate

InfluxDBClient client(INFLUXDB\_URL, INFLUXDB\_ORG, INFLUXDB\_BUCKET, INFLUXDB\_TOKEN, InfluxDbCloud2CACert);

// Data point

Point sensor("esp8266s");

#define MQ\_PIN (A0)

#define RL\_VALUE (5)

#define RO\_CLEAN\_AIR\_FACTOR (9.83)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Software Related Macros\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define CALIBARAION\_SAMPLE\_TIMES (50)

#define CALIBRATION\_SAMPLE\_INTERVAL (500)

#define READ\_SAMPLE\_INTERVAL (50)

#define READ\_SAMPLE\_TIMES (5)

#define PIR D1

#define DHTPIN D2 // what pin we're connected to

#define MQ2 A0 // what pin we're connected to

// Uncomment whatever type you're using!

#define DHTTYPE DHT12 // DHT 11

//#define DHTTYPE DHT22 // DHT 22 (AM2302)

//#define DHTTYPE DHT21 // DHT 21 (AM2301)

const int buzzer = 13;

int Ro = 10;

DHT dht(DHTPIN, DHTTYPE);

//DHT\_Module dht2(DHTPIN);

void setup() {

Serial.begin(115200);

WiFi.mode(WIFI\_STA);

wifiMulti.addAP(ssid, password);

Serial.print("Connecting to wifi");

while (wifiMulti.run() != WL\_CONNECTED) {

Serial.print(".");

delay(100);

}

Serial.println();

pinMode(PIR,INPUT);

Serial.print("PIR Installed\n");

Serial.println("DHTxx test!");

pinMode(MQ2, INPUT);

Serial.print("Calibrating...\n");

//Calibrating the sensor. Please make sure the sensor is in clean air

Serial.print("Calibration is done...\n");

Serial.print("Ro=");

Serial.print(Ro);

Serial.print("kohm");

Serial.print("\n");

Serial.print("\n");

dht.begin();

if (client.validateConnection()) {

Serial.print("Connected to InfluxDB: ");

Serial.println(client.getServerUrl());

} else {

Serial.print("InfluxDB connection failed: ");

Serial.println(client.getLastErrorMessage());

}

sensor.addTag("device", DEVICE);

}

void loop() {

// Wait a few secondsbetween measurements.

delay(1500);

sensor.clearFields();

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

int val = digitalRead(PIR); // read sensor value

if (val == HIGH) { // check if the sensor is HIGH

Serial.println("presence detected"); // turn LED ON

sensor.addField("PIR", 1);

}

else {

Serial.println("Nothing to show");

sensor.addField("PIR", 0);

}

float h = dht.readHumidity();

// Read temperature as Celsius (the default)

float t = dht.readTemperature();

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t)) {

Serial.println("Nu se poate citi de la DHT!");

}

//uint16\_t h2, t2;

//if(dht2.read(t2,h2)){

// Serial.println();

// Serial.print(t2);

// Serial.print(" ");

// Serial.print(h2);

// Serial.println();

// }

// else{

// Serial.println("Modulul nu functioneste pupjos");

// }

// Compute heat index in Fahrenheit (the default)

//float hif = dht.computeHeatIndex(f, h);

// Compute heat index in Celsius (isFahreheit = false)

//float hic = dht.computeHeatIndex(t, h, false);

float mq2\_read = analogRead(MQ2);

Serial.print("Humidity: ");

Serial.print(h);

Serial.print(" %\t");

Serial.print("Temperature: ");

Serial.print(t);

Serial.println(" \*C ");

Serial.print("Gas: ");

Serial.print(mq2\_read);

Serial.print(" ppm");

Serial.print("\n");

if(mq2\_read > float(1000) || t > float(50)){

tone(buzzer, 1000); // Send 1KHz sound signal...

delay(500); // ...for 1 sec

noTone(buzzer); // Stop sound...

sensor.addField("Buzzer", 1);

}

else{

sensor.addField("Buzzer", 0);

}

sensor.addField("Temp", t);

sensor.addField("Humidity", h);

sensor.addField("GAZ", mq2\_read);

// Accurate time is necessary for certificate validation and writing in batches

// For the fastest time sync find NTP servers in your area: https://www.pool.ntp.org/zone/

// Syncing progress and the time will be printed to Serial.

timeSync(TZ\_INFO, "pool.ntp.org", "time.nis.gov");

// Print what are we exactly writing

Serial.print("Writing: ");

Serial.println(sensor.toLineProtocol());

// Check server connection

if (!client.writePoint(sensor)) {

Serial.print("InfluxDB write failed: ");

Serial.println(client.getLastErrorMessage());

}

}

## Cod proiect 2:

int greenPin=2;

int yellowPin=3;

int redPin=4;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

setup\_adc();

pinMode(greenPin,OUTPUT);

pinMode(yellowPin,OUTPUT);

pinMode(redPin,OUTPUT);

}

void loop()

{

delay(1000);

ADCSRA |= (1<<ADSC);//comand citirea valorii potentiometrului

}

void setup\_adc()

{

ADMUX |= (1<<REFS0);//activez tensiunea interna de 5 volti ca prag superior pentru procesarea semnalului analogic

ADMUX |= (1<<MUX1) ;// ADMUX |=(1<<1) ; voi citi semnalul analogic de pe pinul A2

ADCSRA |= (1<<3); //activez intreruperea de la finalul conversiei ADC

ADCSRA |= 0b00000111;//setez valoarea prescalerului la 128

ADCSRA |= (1<<7);//activez convertorul ADC

}

ISR(ADC\_vect)

{

Serial.println("--------------------------------------------------------------");

Serial.println(ADC);

if(ADC < 470)

{

Serial.println("LOW GAS LEVEL IN AIR");

digitalWrite(greenPin,HIGH);

digitalWrite(yellowPin,LOW);

digitalWrite(redPin,LOW); }

else if(ADC>=470 &&ADC<=480)

{

Serial.println("MEDIUM GAS LEVEL IN AIR");

digitalWrite(greenPin,LOW);

digitalWrite(yellowPin,HIGH);

digitalWrite(redPin,LOW); }

else if(ADC>480)

{

Serial.println("HIGH GAS LEVEL IN AIR");

digitalWrite(greenPin,LOW);

digitalWrite(yellowPin,LOW);

digitalWrite(redPin,HIGH); }

Serial.println("--------------------------------------------------------------");

}