



ROBOFUN.RO

LECȚIA XXII

CURS GRATUIT

ARDUINO SI ROBOTICĂ

Alcoolmetru cu Arduino

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs
CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

ESTI LIBER SA DISTRIBUI ACEST DOCUMENT PRIN ORICE MIJLOACE CONSIDERI (EMAIL, PUBLICARE PE WEBSITE / BLOG, PRINTARE, SAU ORICE ALT MIJLOC), ATAT TIMP CAT NU ADUCI NICI UN FEL DE MODIFICARI ACESTUIA. CODUL SURSA DIN ACEST DOCUMENT POATE FI UTILIZAT IN ORICE FEL DE SCOP, DE NATURA COMERCIALA SAU NU, FARA NICI UN FEL DE LIMITARI.

Alcoolmetru realizat cu Arduino Leonardo

Utilizand un Arduino, un senzor MQ-3, un LCD shield si trei leduri brick de culori diferite se poate construi un alcoolmetru hobby foarte rapid.

Senzorul MQ-3 are o sensibilitate ridicata, un timp de raspuns rapid si este sensibil la vaporii de alcool care provin de la o sursa din vecinata. Se alimenteaza in curent continuu de la o sursa de tensiune de 5V si scoate la iesire un semnal analogic raportat la concentratia alcoolului.

Datele privind nivelul de alcool in aerul expirat vor fi afisate pe un LCD avand 2 linii si 16 coloane. Ledurile vor indica 3 nivele prestabiliute de concentratie astfel pentru o concentratie mica se va aprinde doar ledul verde, pentru o concentratie moderata se vor aprinde ledurile verde si galben iar pentru o concentratie mare se vor aprinde toate cele 3 leduri.

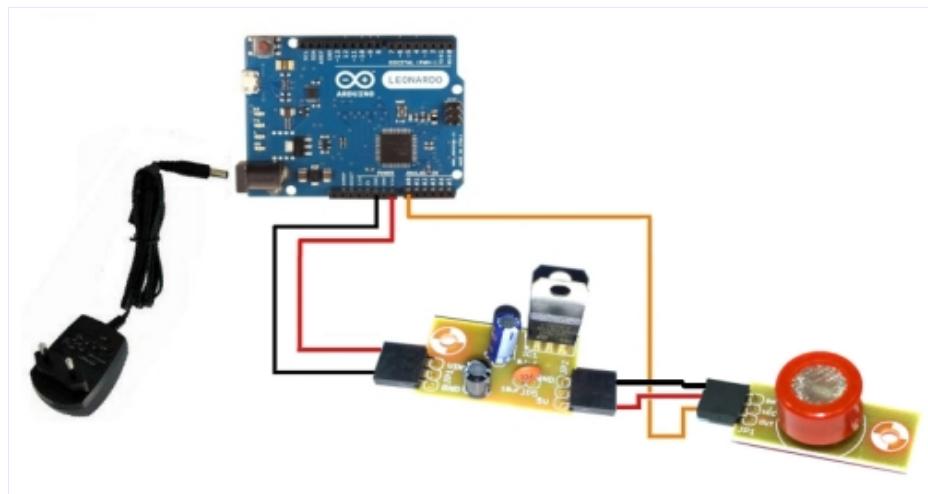
Trebuie sa iti spun de la inceput sa nu te astepti sa obtii o concentratie in procente a nivelului de alcool, acest lucru este destul de complicat de realizat, si necesita conditii de calibrare intr-un laborator specializat. NU folosi acest alcoolmetru pentru situatii in care poti pune vietii in pericol ! Si NU conduce sub influenta bauturilor alcoolice !

OK, acum ca am trecut de partea avertizarilor obligatorii, sa trecem la treaba.

Componentele necesare:

- Arduino UNO http://www.robofun.ro/arduino/arduino_uno_v3
- LCD shield 16x2 <http://www.robofun.ro/shield-lcd-16x2>
- 3 LED-uri brick http://www.robofun.ro/index.php?route=product/search&keyword=led%20brick&category_id=0
- Stabilizator 5V brick <http://www.robofun.ro/stabilizator-5v>
- Senzor alcool brick <http://www.robofun.ro/senzor-alcool-brick>
- Sursa de alimentare externa <http://www.robofun.ro/alimentator-extern-arduino>

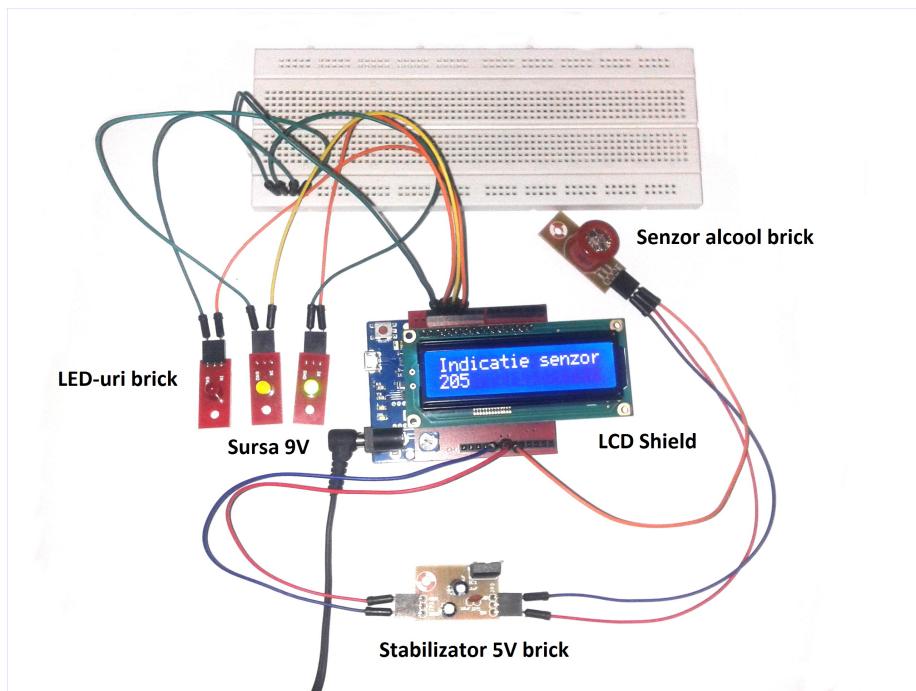
Schema de conectare: stabilizator 5V + senzor alcool brick



Conecteaza shield-ul LCD la Arduino (prin infigere) iar urmareste tabelul de mai jos pentru restul conexiunilor:

Stabilizator 5V brick Vin	Arduino Vin
Stabilizator 5V brick Gnd	Arduino Gnd
Stabilizator 5V brick Gnd	Senzor alcool brick Gnd
Led verde brick IN	Arduino digital 13
Led galben brick IN	Arduino digital 12
Led rosu brick IN	Arduino digital 11
Senzor alcool brick OUT	Arduino analog A0
Led verde brick Gnd	Arduino Gnd
Led galben brick Gnd	Arduino Gnd
Led rosu brick Gnd	Arduino Gnd

In final, vei obtine urmatorul montaj.



Cum functioneaza?

Platforma Arduino se alimenteaza de la o sursa de tensiune externa. Iesirea VIN (care este conectata prin placa Arduino la intrarea sursei de tensiune externa) a platformei se conecteaza la intrarea stabilizatorului 5V brick. Stabilizatorul primeste la intrare tensiunea de 7.5 V – 9 V si scoate la iesire 5V, alimentand senzorul MQ-3. Este necesar sa folosesti un stabilizator extern pentru ca senzorul de alcool MQ-3 consuma un curent relativ ridicat (contine in interior o rezistenta care se incinge in timpul functionarii), asa ca daca l-ai conecta direct la pinul 5V Arduino exista sansa sa arzi placa Arduino.

Codul sursa

```
#include <LiquidCrystal.h>
#define nivel1 40
#define nivel2 60
#define nivel3 80

int analog_in = A0;
int led_verde = 13;
int led_galben = 12;
int led_rosu = 11;
int analog_reading = 0;
int percent_reading = 0;

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

void setup() {
    pinMode(led_verde, OUTPUT);
    pinMode(led_galben, OUTPUT);
    pinMode(led_rosu, OUTPUT);

    lcd.begin(16, 2);
    lcd.print("Alcoolmetru");
    delay(1000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Indicatie senzor");
}
```

```

void loop() {
    lcd.setCursor(0,1);
    analog_reading = analogRead(analog_in);
    lcd.print(analog_reading);
    percent_reading=map(analog_reading, 0, 1023, 0, 100);

    if (percent_reading < nivel1){
        digitalWrite(led_verde,HIGH);
        digitalWrite(led_galben,LOW);
        digitalWrite(led_rosu,LOW);
    } else if (percent_reading < nivel2){
        digitalWrite(led_verde,HIGH);
        digitalWrite(led_galben,HIGH);
        digitalWrite(led_rosu,LOW);
    } else if (percent_reading < nivel3){
        digitalWrite(led_verde,HIGH);
        digitalWrite(led_galben,HIGH);
        digitalWrite(led_rosu,HIGH);
    }
}

```

Programul incepe cu directiva preprocesor aferenta afisajului:

```
#include<LiquidCrystal.h>
```

Se definesc 3 constante de nivel avand valorile aferente:

```
#define nivel1 40
#define nivel2 60
#define nivel3 80
```

Urmatoarele instructiuni declara pinul aferent intrarii care provine de la senzor, ledurile conectate la platforma, variabila ce va stoca valoarea concentratiei si variabila ce va stoca procentul de concentratie.

```

int analog_in = A0;
int led_verde = 13;
int led_galben = 12;
int led_rosu = 11;
int analog_reading = 0;
int percent_reading = 0;
```

Se initializeaza afisajul LCD:

```
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
```

In rutina setup() pinii care controleaza ledurile sunt declarati ca fiind pini de iesire :

```
pinMode(led_verde, OUTPUT);
pinMode(led_galben, OUTPUT);
pinMode(led_rosu, OUTPUT);
```

Instructiunea urmatoare seteaza numarul de linii si de coloane pentru LCD, respectiv 16 coloane si 2 linii:

```
lcd.begin(16, 2);
```

Urmatorul set de instructiuni afiseaza pe display text.

```
lcd.print("Alcoolmetru");
delay(1000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Indicatie senzor");
```

Se continua cu executia buclei loop(). Instructiunea schimba pozitia cursorului la coloana 0 si linia 1 deoarece acolo se va afisa informatia analogica furnizata de senzor.

```
lcd.setCursor(0,1);
```

Functia analogRead(analog_in) returneaza in analog_reading valoarea citita de la senzor.

```
analog_reading = analogRead(analog_in);
```

Instructiunea afiseaza pe LCD continutul variabilei *analog_reading*:

```
lcd.print(analog_reading);
```

Functia map schimba domeniul intervalului 0 – 1023 in 0 – 100 iar rezultatul se stocheaza in *percent_reading*:

```
percent_reading=map(analog_reading, 0, 1023, 0, 100);
```

Urmatorul bloc de instructiuni testeaza daca variabila *percent_reading* a depasit un anumit nivel si aprinde ledurile astfel:

```
if (percent_reading < nivel1){ // daca percent_reading este mai mic  
//decat nivel1  
    digitalWrite(led_verde,HIGH); // aprinde ledul verde  
    digitalWrite(led_galben,LOW); // stinge celelalte doua leduri  
    digitalWrite(led_rosu,LOW);  
} else if (percent_reading < nivel2){ // daca prima conditie nu a  
//fost satisfacuta se testeaza a doua  
    digitalWrite(led_verde,HIGH); // se aprind primele doua leduri  
    digitalWrite(led_galben,HIGH);  
    digitalWrite(led_rosu,LOW); // cel de-al treilea ramane stins,  
//suntem in concentratie medie  
} else if (percent_reading < nivel3){ // daca s-a depasit nivelul 3  
//de concentratie  
    digitalWrite(led_verde,HIGH); // se aprind toate ledurile.  
    digitalWrite(led_galben,HIGH);  
    digitalWrite(led_rosu,HIGH);  
}
```