

# Environmental Monitoring System

**Autor:** [Popescu Lucian-Ioan](#)

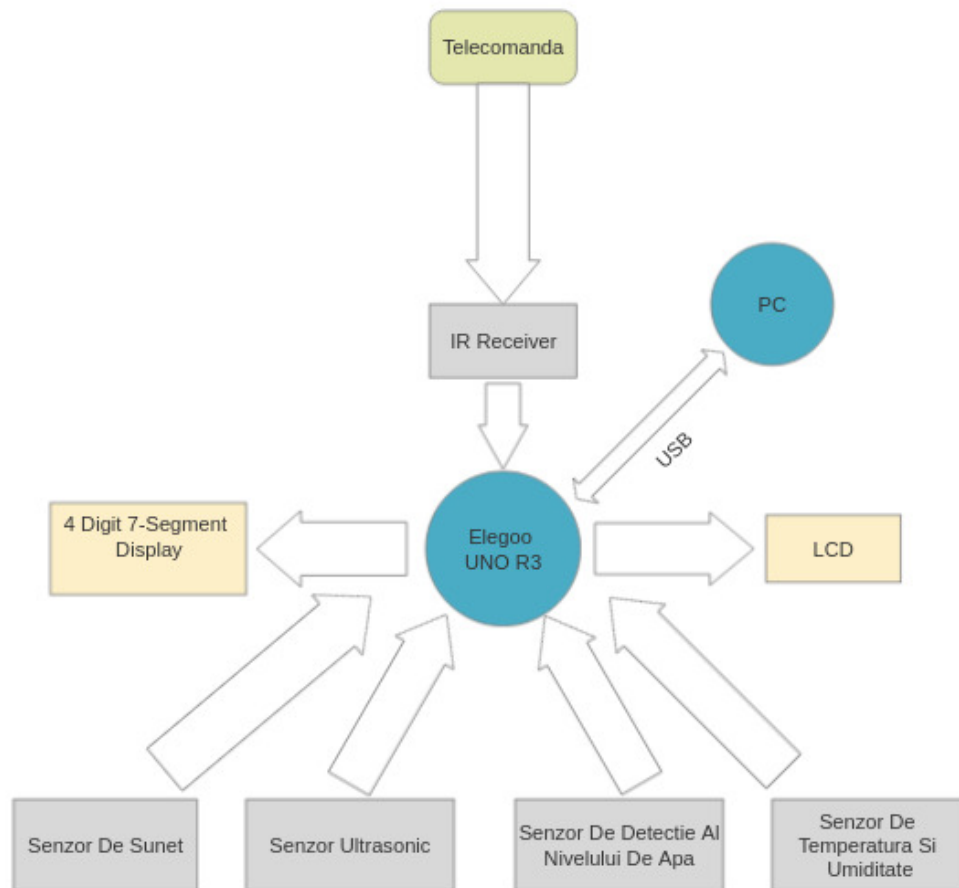
**Grupa:** 336CB

## Introducere

Sistem de colectare a datelor de la senzori si afisare a lor pe LCD. Selectarea datelor afisate se face prin intermediul unei telecomenzi. In plus exista un LCD suplimentar care va afisa uptime-ul.

Ideea proiectului a plecat de la dorinta de a agrega si multiplexa cat mai multi senzori pe Arduino pentru a culege si lucra cu datele din mediul inconjurator. Proiectul e folositor deoarece el poate fi modelat astfel incat sa fie folosit in viata de zi cu zi pentru a colecta date legate de temperatura, umiditate, puritatea aerului intr-o casa sau masina smart.

## Descriere generală

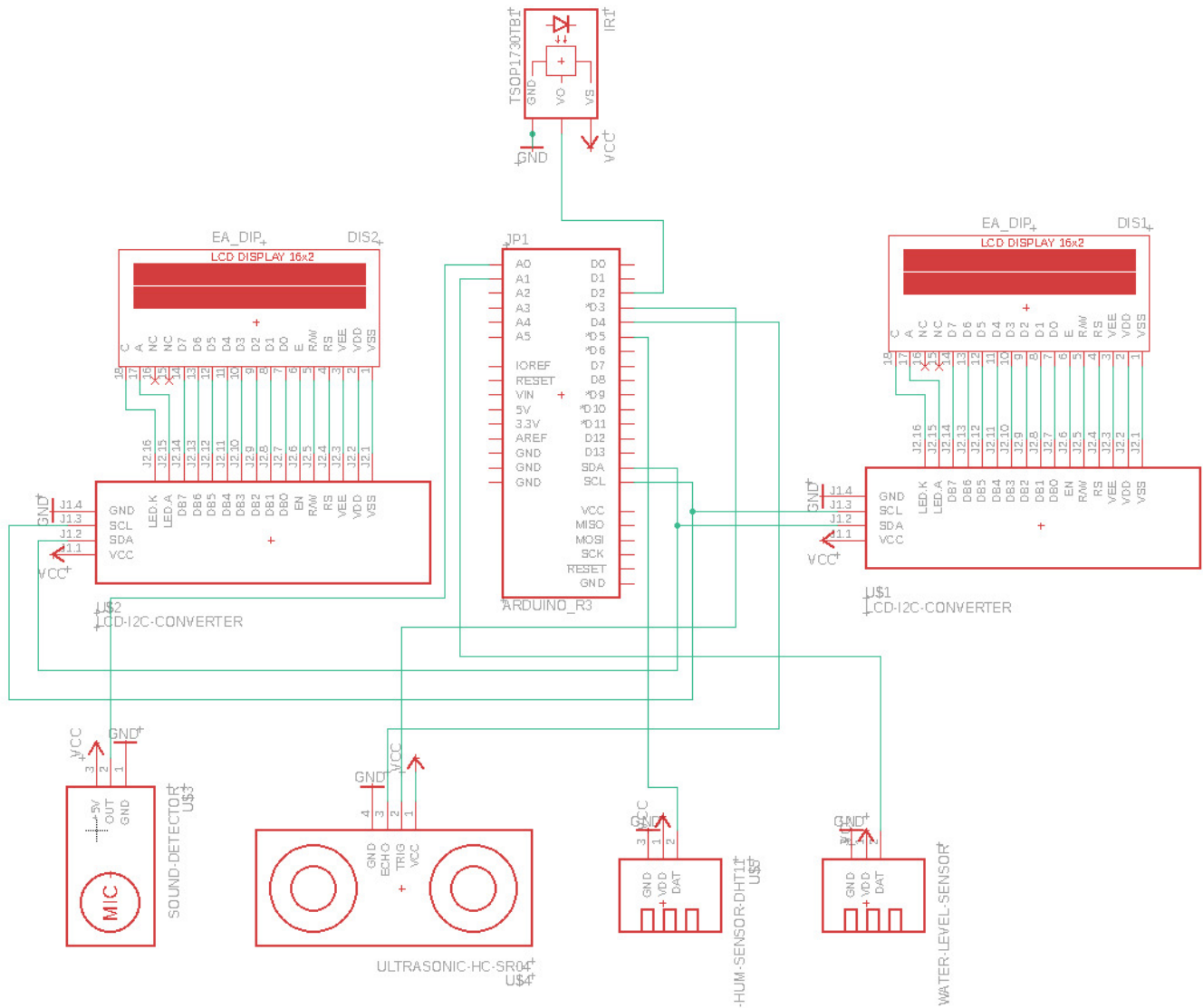


In loc de 4x7 display voi folosi tot un LED 1602 I2C deoarece pinii de pe placa nu sunt suficienti.

## Hardware Design

Componente necesare:

- Elegoo UNO R3
- 2x LCD1602
- Telecomanda
- IR Receiver
- DHT11 Senzor De Temperatura Si Umiditate
- Senzor De Sunet
- Senzor Ultrasonic
- Senzor De Detectie Al Nivelului De Apa
- PC



## Software Design

Codul sursa se poate gasi in urmatorul repo: <https://github.com/lucic71/EMS>

La fiecare moment de timp, in functia loop, se afiseaza pe LCD-ul 1 meniul utilizatorului. Pe LCD-ul 2 e afiseaza uptime-ul in secunde. Sistemul asteapta de la utilizator o comanda, iar atunci cand o primeste se decide ce meniu va afisa pe ecran.

Functiile statice definite in fisierul .ino comunica cu senzorii sistemului pentru a colecta si afisa mai apoi date pe LCD.

A fost nevoie sa hardcodez tastele venite de la telecomanda, pentru asta am folosit urmatoarele define-uri:

```
#define KEYONE 0XF30CFF00
#define KEYTWO 0XE718FF00
#define KEYTHREE 0XA15EFF00
#define KEYFOUR 0XF708FF00
#define KEYEQ 0XE619FF00
```

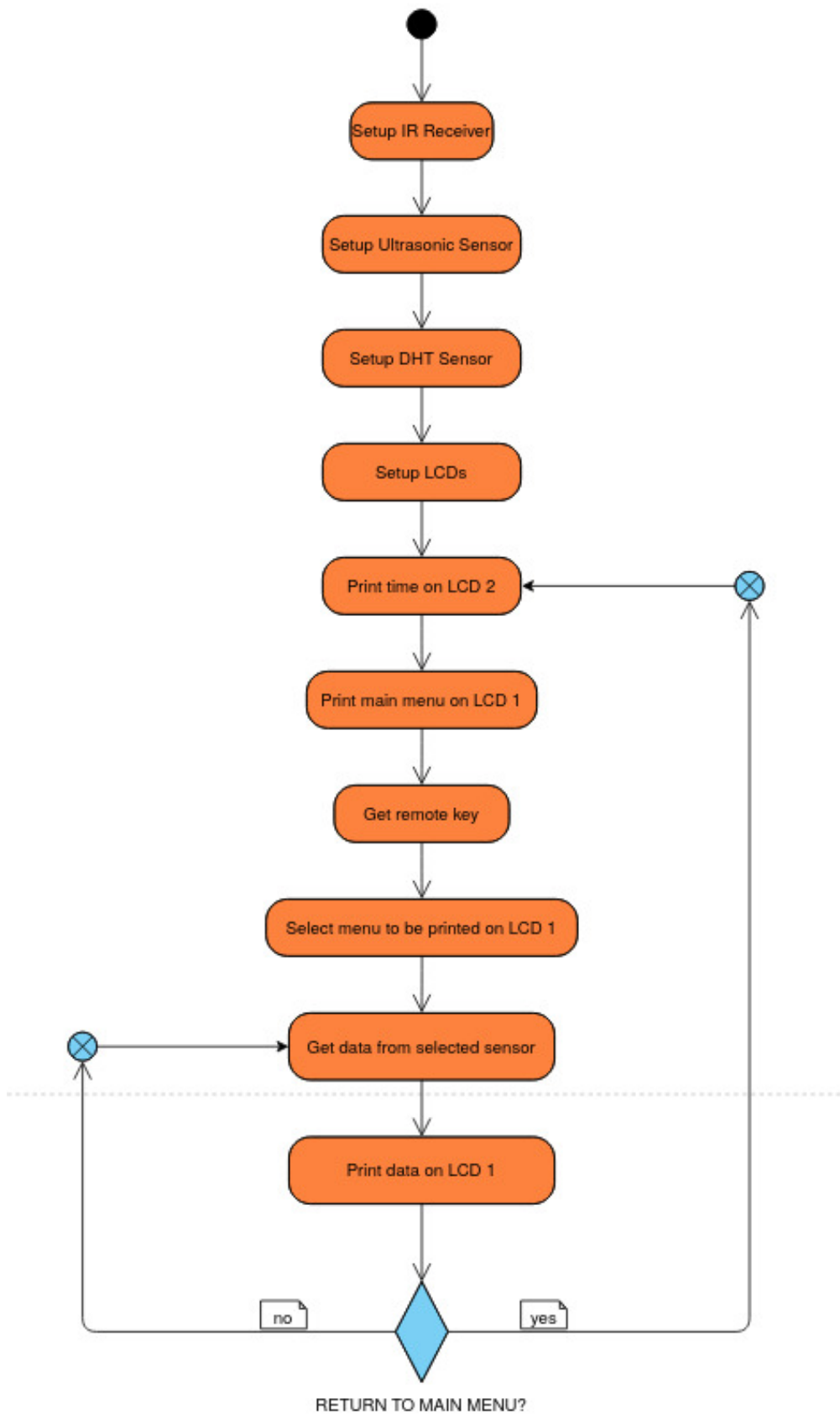
```
#define KEYUNDEF 0xFFFFFFFF
```

Pentru implementare am folosit urmatoarele biblioteci:

- [IRremote](#), pentru comunicare cu telecomanda
- [DHT](#), pentru comunicarea cu senzorul de temperatura
- [LiquidCrystal\\_I2C](#), pentru comunicarea cu ecranele LCD

Ca mediu de dezvoltare am folosit Arduino IDE.

[Diagrama de activitate:](#)



## Rezultate obtinute

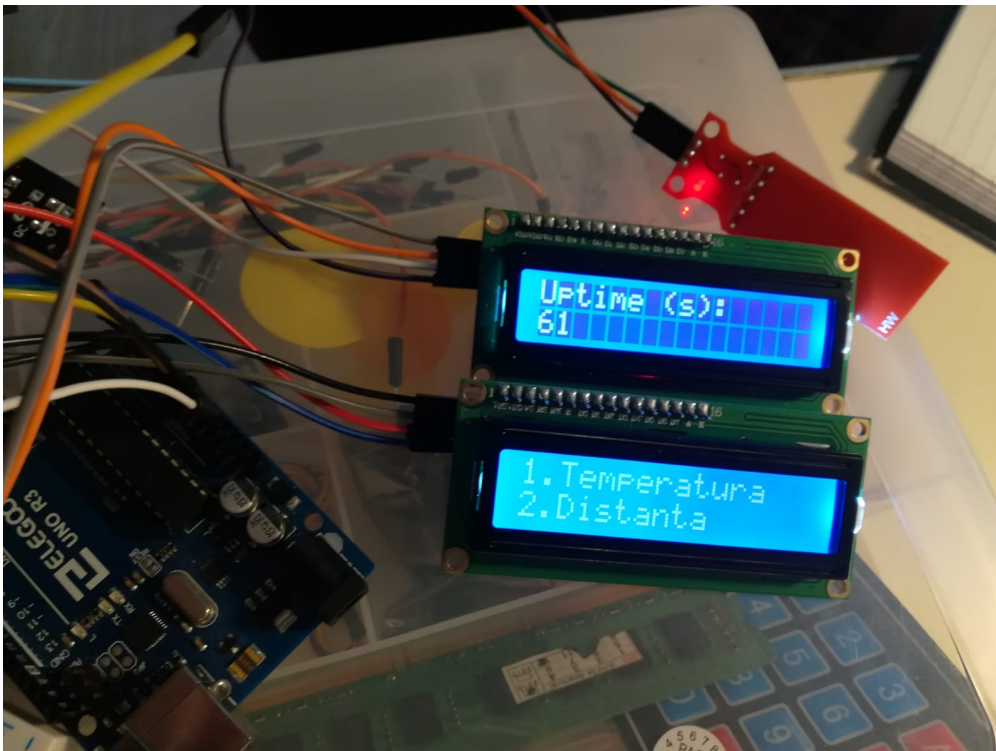
Meniurile de pe LCD care se afiseaza utilizatorului sunt:

1. Temperatura
2. Distanta
3. Nivel sunet
4. Nivel apa

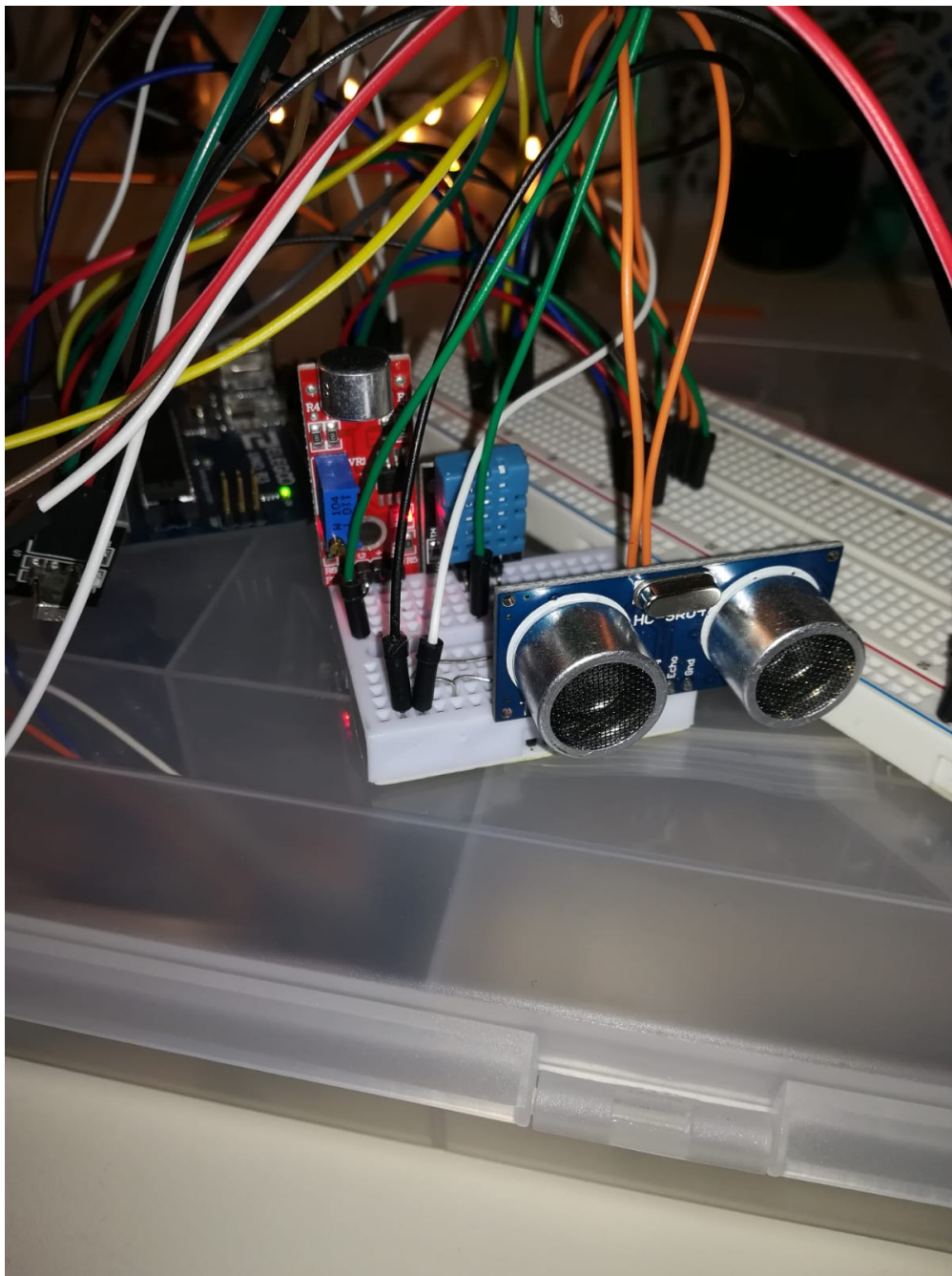
Pentru a selecta un meniu se apasa tasta 1, 2, 3, respectiv 4 de pe telecomanda. Pentru a reveni in meniul principal se apasa tasta EQ.

Mai jos se gasesc cateva poze cu sistemul rezultat si un video demo.

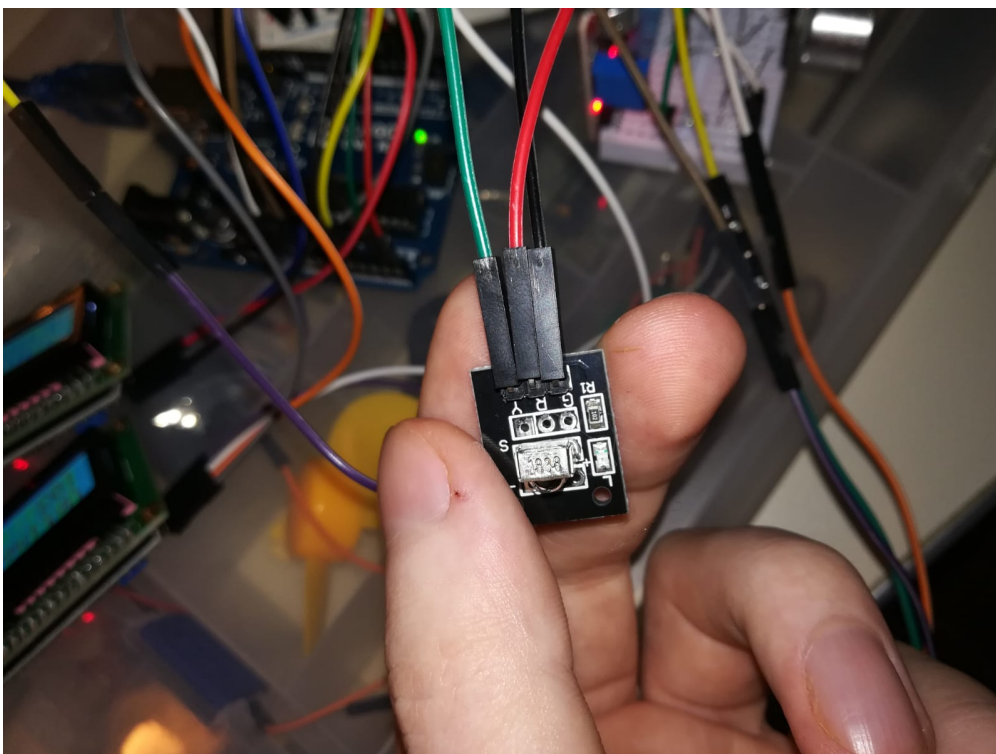
Cele doua LCD-uri:



Senzorii de sunet, temperatura si ultrasonic conectati pe breadboard:



IR receiver-ul folosit pentru comunicarea cu telecomanda:



Pentru ca am folosit 2 LCD-uri cu I2C a fost nevoie ca pentru unul dintre ei sa modific adresa. Pentru asta am legat pinii de adresa A0 ca mai jos:



Demo: <https://www.youtube.com/watch?v=A8L2GOVxvNM>

## Concluzii

Am reusit sa implementez toata functionalitatea pe care mi-am propus-o.

Cea mai dificila parte a fost legata de interconectarea componentelor prin intermediul placutei. Initial mi-am propus sa folosesc un 4 digit 7 segment display pentru afisarea uptime-ului, dar mi-am dat seama ca nu-mi ajung pini de pe placuta. In final am decis sa folosesc doua LCD-uri 16x2 cu I2C pentru ca folosesc doar 4 pini de pe placuta.

Implementarea software a mers foarte bine deoarece am gasit rapid biblioteci pentru componentele mele. Ce mi-a ramas de facut a fost sa creez logica pentru interactiunea cu utilizatorul.

## Bibliografie/Resurse

<https://www.arduino.cc/reference/en/libraries/>

[https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2021/abirlica/environmental\\_monitoring\\_system](http://ocw.cs.pub.ro/courses/pm/prj2021/abirlica/environmental_monitoring_system) 

Last update: **2021/05/30 17:17**