

Sistem pentru detecția precipitațiilor și monitorizarea acestora

Introducere

Prezentarea pe scurt a proiectului:

Scopul proiectului meu este de a realiza un sistem de detecție a ploii, care poate memora cantitatea de apă care se află pe senzorul de precipitații la un interval de timp și care poate afișa cantitatea curentă detectată.

Atunci când se apasă un buton, se va face comutarea de la valoarea curentă a precipitațiilor la valorile salvate în memorie.

Descriere generală

Schema bloc a proiectului:



Sistemul, la pornirea inițială, va afișa cantitatea de precipitații indicată de senzor la momentul curent de timp.

La apăsarea unui buton(B1), se va comuta la afișarea cantităților stocate în memorie împreună cu ziua și ora acelei înregistrări. Se poate naviga prin înregistrări cu două butoane(înainte(B2) și înapoi(B3)).

Când se dorește revenirea la valorile curent citite, se apasă, din nou, pe butonul (B1).

Hardware Design

Lista de componente pe care le voi utiliza:

- 1 x [Placa de dezvoltare compatibila ARDUINO UNO R3](#)
- 1 x [Ecran LCD I2C](#)
- 1 x [Placa de stocare MicroSD](#)
- 1 x [Modul RTC DS1302](#)
- 3 x [Buton Mini în 4 pini](#)
- 1 x [Modul senzor ploaie](#)

- 1 x [Breadboard 830 puncte](#)
- 1 x [Buzzer Pasiv 5V](#)

Schema electrică:



Pinii folosiți în realizarea cablajului sunt următorii:

- Pentru ecranul LCD, folosind protocolul I2C, legat pinul SDA al ecranului la pinul A4 al plăcuței și pinul SCL al ecranului la pinul A5 al plăcuței, fiind pinii standard pentru acest tip de comunicare.
- Pentru modulul RTC DS1302, pinul de CLK la pinul digital 6 al plăcii, pinul de DAT la pinul digital 7 al plăcii și pinul RST la pinul digital 2 al plăcuței.
- Pentru senzorul de ploaie, m-am folosit de pinul analogic al acestuia, A0, pe care l-am legat la pinul A0 al plăcuței.
- Pentru cititorul de card MicroSD, am folosit legarea standard pentru comunicarea peste SPI și anume pinul MOSI la pinul digital 11 al plăcuței, pinul MISO la pinul digital 12 al plăcuței, pinul SCK la pinul digital 13 al plăcuței și pinul pentru CS la pinul digital 10 al plăcuței.
- Buzzer-ul a fost conectat la masă și la pinul digital 9 al plăcuței.
- Cele 3 butoane le-am conectat la masă și la pinii digitali 3, 4 și, respectiv, 5 ai plăcuței.

Software Design

Pentru realizarea componentei software am folosit mediul de dezvoltare Arduino IDE.

Bibliotecile terțe utilizate sunt:

- `virtuabotixRTC.h` pentru setarea și preluarea informațiilor de la modulul RTC DS1302.
- `pitches.h` pentru o serie de note folosite în sunetul redat pe Buzzer.

Execuția codului este una destul de clară care urmează pașii:

- Se declară două variabile, `virtualbotixRTC myRTC(6, 7, 2)` și `LiquidCrystal_I2C lcd(0x27, 16, 2)` de care va depinde interacțiunea cu modulul RTC și cu ecranul lcd.
- Se declară variabile ajutătoare, printre care variabile care rețin data și ora de stocat pe cardul SD la un moment dat, tipul de precipitații detectate la un moment de timp ("Dry", "Droplets", "Light Rain", "Moderate Rain", "Heavy Rain" sau "Extreme Rain") sau cele două șiruri care rețin notele muzicale respectiv durata acestora.
- Se inițializează interfața Serială, se setează pinii pentru Chip Select(card SD), B1, B2, B3, buzzer și, respectiv, senzorul de ploaie, se inițializează ecranul, cardul SD și fișierul de stocare al informațiilor.
- În funcția `loop` se verifică dacă ar trebui să înceapă melodia(dacă ultimul status a fost "Heavy Rain" sau "Extreme Rain"), respectiv dacă ar trebui să continue melodia. La apăsarea butonului B1, se comută din modul `Timp Real (Live Feed)` în modul `Memorie (Memory)` sau invers și se setează variabilele necesare fiecărui mod. La comutare între moduri se va afișa, pentru o secundă "Read From Memory"(comutare mod `Timp Real` → `Memorie`) sau "Live Feed"(comutare mod `Memorie` → `Timp Real`).
- Pentru afișarea datelor, în funcție de modul curent se vor afișa fie informații în timp real după structura `DD/MM HH:MM RAIN_STATUS` în cazul modului `Timp Real`, fie se vor afișa informații salvate în memorie prin care se poate naviga folosind butonul B2 pentru a merge înainte și butonul B3 pentru a merge înapoi în cazul modului `Memorie`.

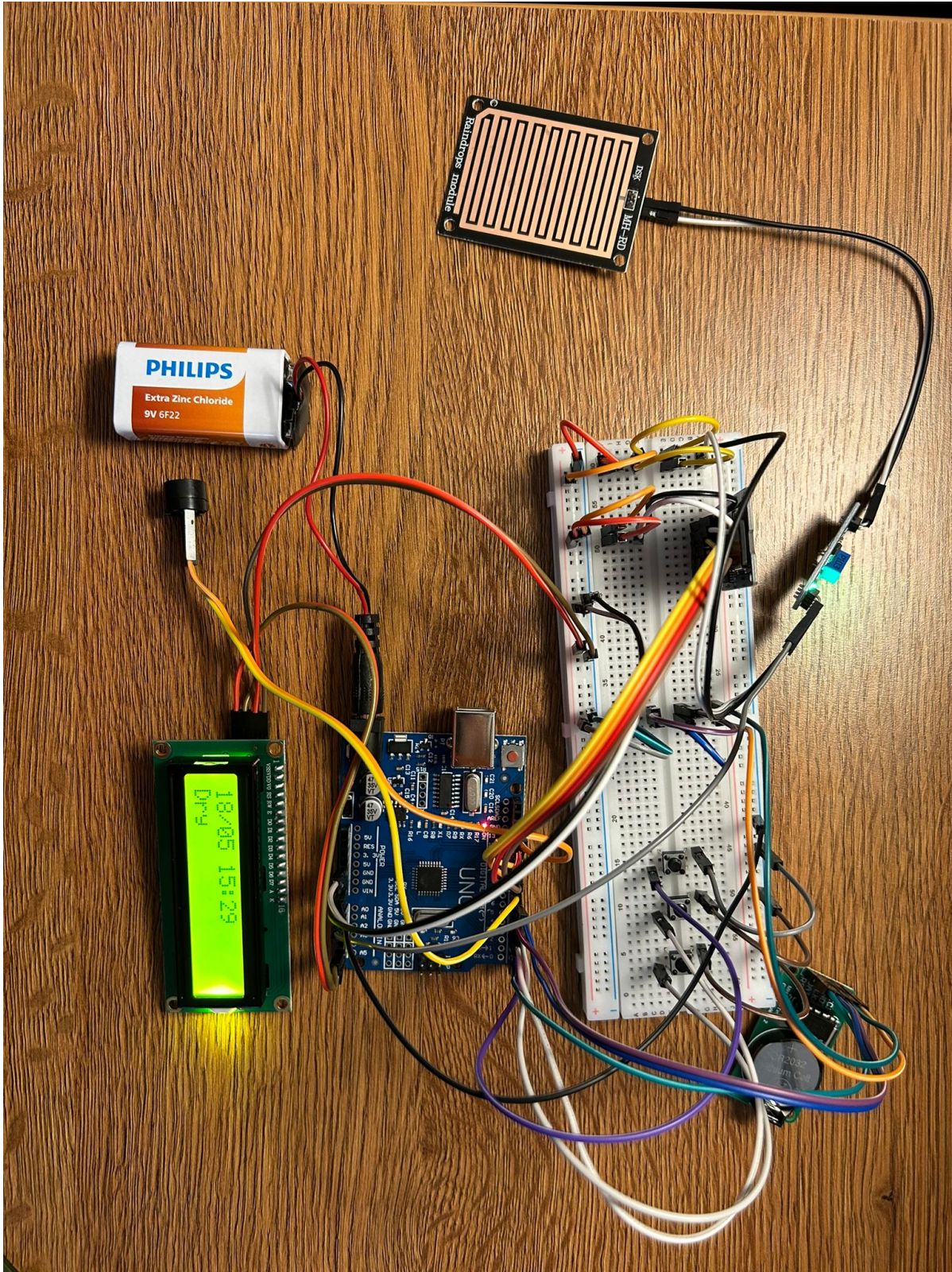
- Navigarea în memoria cardului SD(`void readAndDisplayData()`) se face folosind poziția de fișier. Această parte este ușurată datorită formatului ușor de parsat în care stochez informația("DD/MM HH:MM,RAIN_QTY RAIN_STATUS,").
- Pentru determinarea stării precipitațiilor(`char *getRainCategory(int intensity)`), preiau informația de la senzorul de ploaie pe portul analog A0 și o mapez, pe rând, în intervale de intensitate.
- La fiecare pornire a sistemului, informațiile anterior stocate se șterg.
- Noțiunile folosite din laboratoare sunt comunicarea prin protocolul I2C pentru interacțiunea cu ecranul lcd; comunicarea prin protocolul SPI pentru interacțiunea cu cardul micro SD; GPIO pentru interacțiunea cu butoanele; PWM pentru a genera sunet din Buzzer; ADC pentru a converti informația analogică de la senzorul de ploaie.
- Calibrarea senzorilor am realizat-o pe măsură ce am realizat testarea acestora pentru a obține valorile așteptate(calibrarea a fost necesară doar pentru senzorul de ploaie pentru a fi mai puțin sensibil la apă).

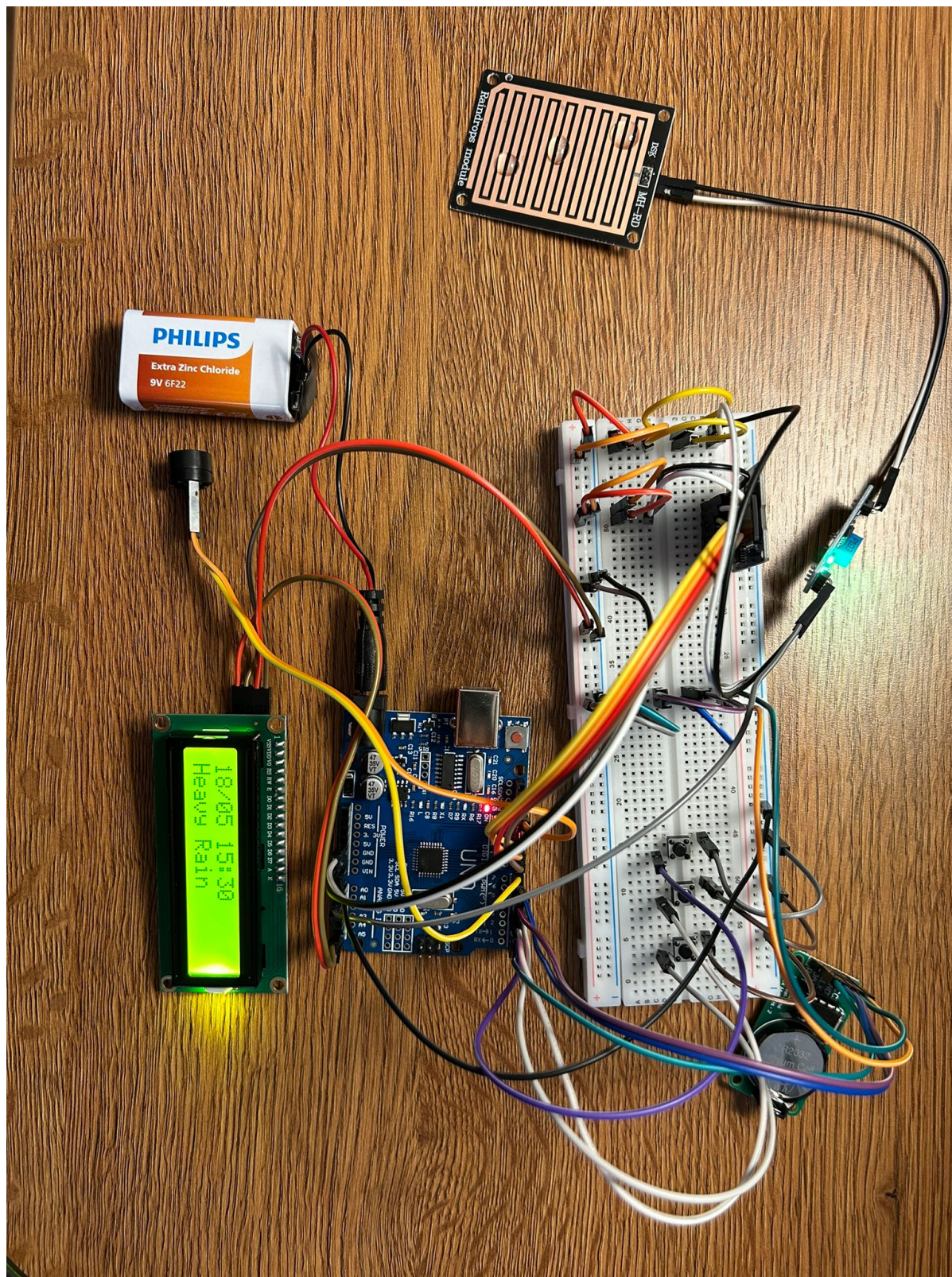
Surse folosite:

- [Bibliotecă și exemple modul DS1302](#)
- [Conectare modul card SD și utilizare](#)

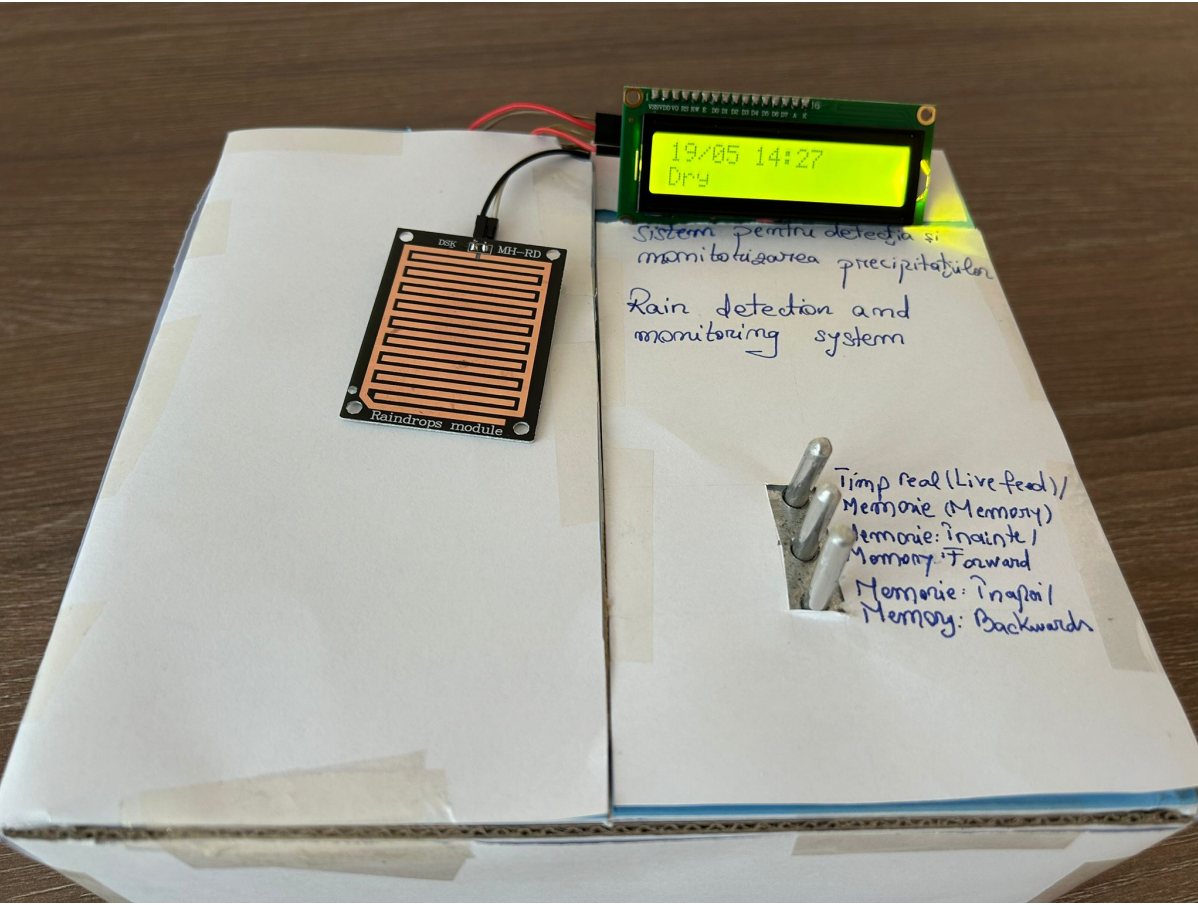
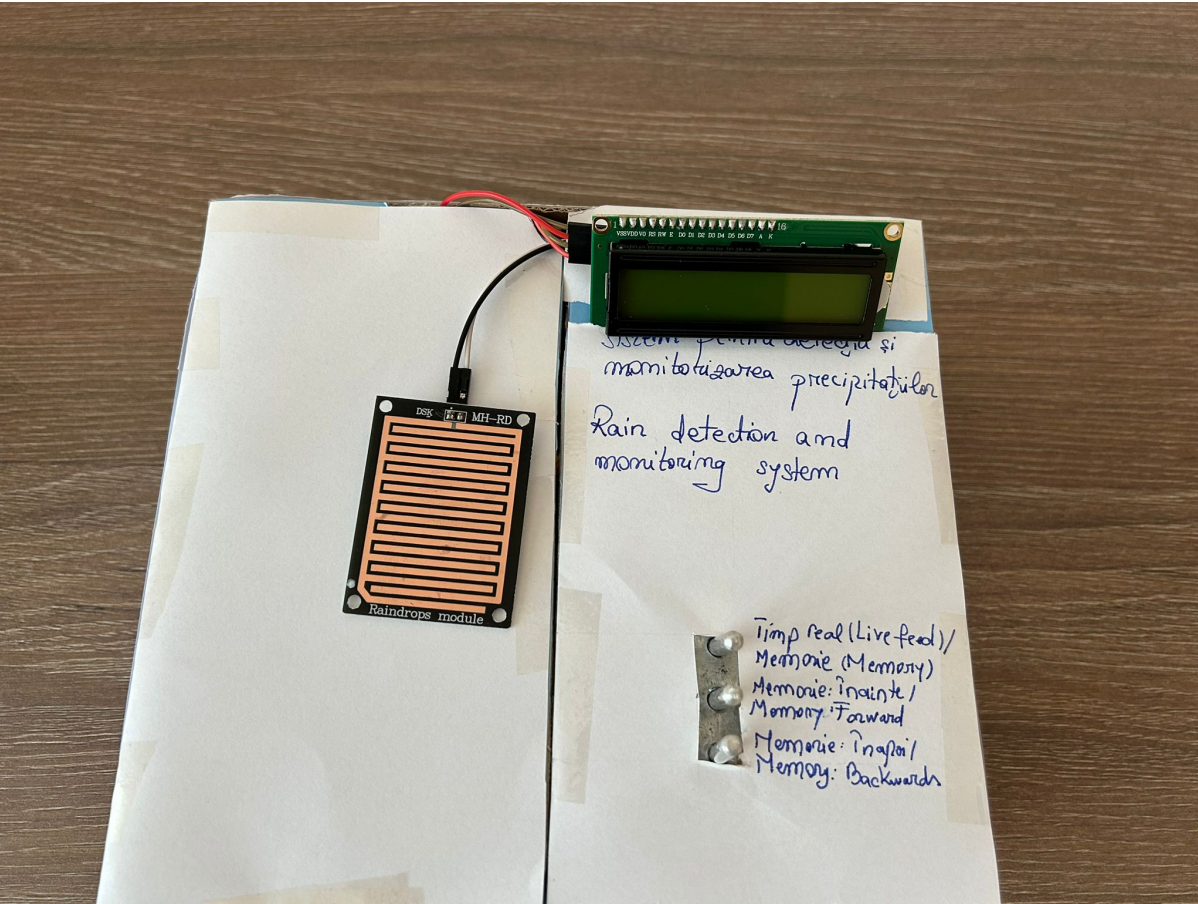
Rezultate Obținute

Stadiul proiectului după conectarea tuturor componentelor și calibrarea senzorului de ploaie:





Stadiul proiectului după finalizarea componentei software și pregătirea acestuia în cutia specială:



Concluzii

Acest proiect a fost unul interesant de realizat, în principal datorită faptului că a trebuit să realizăm singuri totul (de la alegerea pieselor și conectarea acestora la alegerea bibliotecilor corespunzătoare și proiectarea software) care trece pe oricine prin toate etapele (și toate stările de agonie și, respectiv extaz).

Problemele majore întâmpinate au fost la utilizarea excesivă a memoriei dinamice a plăcuței ceea ce ducea la coruperea datelor scrise pe cardul micro SD, în primă instanță, iar mai apoi la imposibilitatea comunicării cu acesta și chiar distorsionarea informațiilor afișate pe display.

Validarea funcționării conforme a proiectului am realizat-o pe bază experimentală până când am atins rezultatul dorit: un proiect stabil cu un comportament determinist. [Link Demo](#)

Download

[Link download cod sursă](#)

Jurnal

07.05.2025 - Am plasat comanda pentru piese.

08.05.2025/09.05.2025 - Au ajuns piesele.

09.05.2025 - Am comandat câteva fire în plus și unele de dimensiuni mai mari.

10.05.2025/14.05.2025 - Au ajuns restul de piese.

13.05.2025 - Primă conectare a pieselor, doar alimentare la curent.

16.05.2025 - Montaj complet piese

17.05.2025 - Testare funcționalități piese și începere implementare software.

19.05.2025 - Finalizare implementare software și realizare cutie de depozitare.

Bibliografie/Resurse

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/avaduva/denis.vladulescu>



Last update: **2025/05/19 13:44**