

# AgriSmart Mini pentru agricultură inteligentă

## Introducere

Proiectul realizat reprezintă un sistem inteligent ce este folosit pentru îngrijirea plantelor bazat pe senzori și microcontroler ATmega328P. Scopul principal al proiectului este supravegherea condițiilor de mediu necesare dezvoltării sănătoase a unei plante.

Sistemul monitorizează:

- umiditatea solului
- intensitatea luminii
- detectarea ploii
- data și ora măsurărilor

Valorile sunt afișate pe un display OLED și pot fi salvate pe un card MicroSD pentru analiză ulterioară.

Ideea proiectului a pornit de la necesitatea automatizării îngrijirii plantelor și a monitorizării condițiilor de mediu într-un mod simplu și accesibil.

Proiectul este util deoarece:

- ajută la îngrijirea corectă a plantelor
- oferă informații în timp real

## Descriere Generală

Sistemul este construit în jurul microcontrolerului ATmega328P, care colectează date de la senzori și controlează modulele de afișare și semnalizare.

## Schema Bloc



## Descriere generală

### Senzor umiditate sol

Măsoară nivelul de saturație cu apă a solului pe baza rezistenței/capacității electrice. Senzorul

generează o tensiune analogică variabilă. Microcontrolerul ATmega preia acest semnal printr-un pin configurat ca ADC (Convertor Analog-Digital), transformând tensiunea într-o valoare procentuală (0-100%). Această valoare dictează direct comportamentul modului de LED-uri, trimite procentajul către display-ul OLED pentru afișare și îl împachetează alături de celelalte date pentru a fi scris pe cardul MicroSD.

## **Senzor digital de lumină BH1750**

Măsoară intensitatea luminoasă ambientală. Spre deosebire de senzorul de sol, acesta are propriul ADC intern și comunică direct digital prin magistrala I2C. Microcontrolerul acționează ca Master, cerând periodic date de la senzor. Odată citită, valoarea este procesată de MCU și direcționată către două ieșiri: este formatată și trimisă vizual pe o pagină a display-ului OLED și este salvată pe MicroSD pentru istoric.

## **Modul senzor detectare ploaie**

Detectează prezența picăturilor de apă pe suprafața plăcuței de contact modificându-și conductivitatea. În momentul în care detectează apă, trimite un semnal către microcontroler. La declanșarea acestui semnal, MCU-ul execută o rutină de alertă: schimbă starea unui pin GPIO pentru a aprinde instantaneu LED-ul albastru, actualizează statusul pe display și înregistrează evenimentul împreună cu ora exactă oferită de RTC pe MicroSD.

## **Modul RTC DS3231**

Păstrează cu precizie data și ora curentă, fiind independent de resetările microcontrolerului datorită bateriei proprii (CR2032). Modulul este interogată constant de microcontroler prin protocolul I2C. RTC-ul nu interacționează direct cu alți senzori, dar informația furnizată de el este "lipită" (atașată ca timestamp) de către MCU la fiecare pachet de date citit de la senzorii de umiditate, ploaie și lumină. Fără acest modul, datele trimise către MicroSD nu ar avea context temporal. De asemenea, ora este afișată pe display.

## **Modul MicroSD**

Asigură stocarea nevolatilă a tuturor informațiilor colectate de sistem pentru a putea fi analizate ulterior (ex: sub forma unui grafic pe laptop). Comunică cu microcontrolerul prin protocolul SPI (MOSI, MISO, SCK, CS). Microcontrolerul adună datele de la toți senzorii de intrare (I2C și ADC), formează un șir de caractere (tip CSV: Dată, Oră, Umiditate, Lumină, Status Ploaie) și îl scrie periodic pe card.

## **Display OLED (I2C)**

Oferă interfața grafică, afișând parametrii în timp real. Este un dispozitiv exclusiv de ieșire. Microcontrolerul folosește magistrala I2C pentru texte și simboluri pe ecran. Conținutul afișat se schimbă dinamic pe baza datelor culese de la senzori și a input-ului primit de la buton. La apăsarea butonului, MCU-ul șterge ecranul curent și trimite datele pentru următoarea pagină de informații.

## **LED-uri indicatoare de stare și Buton**

Reprezintă elementele de interacțiune directă de bază (I/O) controlate prin pinii GPIO ai microcontrolerului. MCU-ul trimite semnale către LED-uri pentru a semnaliza vizual o anumită acțiune dedusă din logica senzorilor (Verde = Sol umed, Galben = Umiditate medie, Roșu = Necesită udare, Albastru = Plouă).

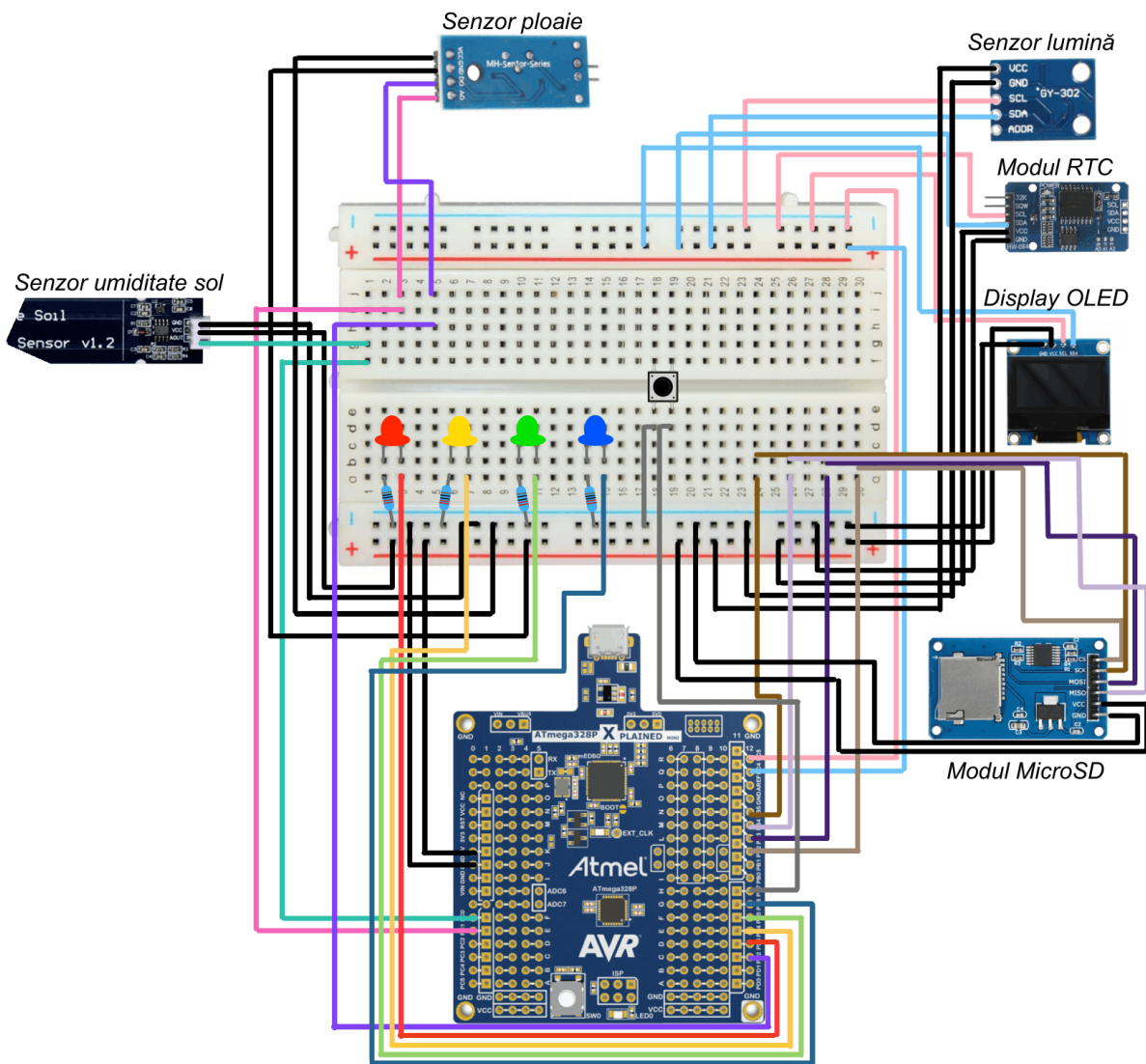
# Hardware Design

## Listă de componente

- ATmega328P
- Display OLED I2C
- Senzor umiditate sol
- Senzor lumină BH1750
- Modul RTC DS3231
- Modul MicroSD
- Modul senzor ploaie
- LED-uri (roșu, verde, galben, albastru)
- Buton
- Rezistori și condensatori
- Breadboard 400 puncte
- Fire dupont

## Schema electrică

Am realizat schema electrică de tip Fritzing deoarece în alte aplicații nu am găsit toate componentele.



### Alimentare

PIN J202	Semnal	Merge la
----------	--------	----------

5	+5V	Rail + nr. 1 breadboard
6	GND	Rail - nr. 1 breadboard

Toate modulele sunt alimentate de la +5V și GND. Alimentarea este distribuită prin rail-urile breadboard-ului.

### Magistrala I2C (OLED + RTC + BH1750)

Pe magistrala I2C sunt conectate OLED-ul, senzorul BH1750 și modulul RTC DS3231.

Pin ATmega328P	Merge la
PC4	Rail + nr. 2 breadboard
PC5	Rail - nr. 2 breadboard

### Display OLED I2C

Pin OLED	Pin ATmega328P
VCC	+5V
GND	GND
SDA	PC4
SCL	PC5

### Senzor BH1750

Pin BH1750	Pin ATmega328P
VCC	+5V
GND	GND
SDA	PC4
SCL	PC5

### Modul RTC DS3231

Pin DS3231	Pin ATmega328P
VCC	+5V
GND	GND
SDA	PC4
SCL	PC5

Pinul PC4 este folosit pentru linia de date SDA, iar pinul PC5 este folosit pentru linia de ceas SCL.

Motivul folosirii acestor pini este că, pe microcontrollerul ATmega328P, PC4 și PC5 sunt pinii hardware dedicați pentru comunicația I2C. Astfel, toate modulele I2C pot fi conectate în paralel pe aceleași două linii, fiecare având o adresă diferită.

## Modul MicroSD – SPI

Pin MicroSD	Pin ATmega328P
CS	PB2
MOSI	PB3
MISO	PB4
SCK	PB5
VCC	+5V
GND	GND

Modulul MicroSD folosește interfața SPI. Această interfață are patru linii principale:

- CS — selectarea modulului;
- MOSI — date trimise de la microcontroller către card;
- MISO — date trimise de la card către microcontroller;
- SCK — semnalul de ceas SPI.

Pinii PB3, PB4 și PB5 sunt folosiți deoarece sunt pinii hardware SPI ai microcontrollerului. PB2 este folosit ca pin CS pentru selectarea cardului SD în timpul comunicației.

## Senzor umiditate sol

Pin senzor	Pin ATmega328P
AO	PC0
VCC	+5V
GND	GND

Ieșirea analogică AO a senzorului este conectată la PC0. Acest pin poate funcționa ca intrare analogică, permițând citirea unei valori proporționale cu umiditatea solului.

PC0 este ales deoarece este unul dintre pinii analogici disponibili ai microcontrollerului.

## Senzor ploaie

Pin senzor	Pin ATmega328P
AO	PC1
DO	PD2
VCC	+5V
GND	GND

Senzorul de ploaie folosește două ieșiri:

- AO conectat la PC1, pentru citirea analogică a nivelului de apă;
- DO conectat la PD2, pentru citirea digitală a pragului de detecție.

PC1 este folosit deoarece poate funcționa ca intrare analogică. PD2 este folosit pentru semnal digital

și poate fi util și dacă se dorește folosirea unei întreruperi externe.

## LED-uri

LED	Pin ATmega328P
Albastru	PD3
Verde	PD4
Galben	PD5
Roșu	PD6

LED-urile sunt conectate la pini digitali ai microcontrollerului. Fiecare LED este conectat și printr-o rezistență de limitare a curentului de 220 Ω.

## Buton

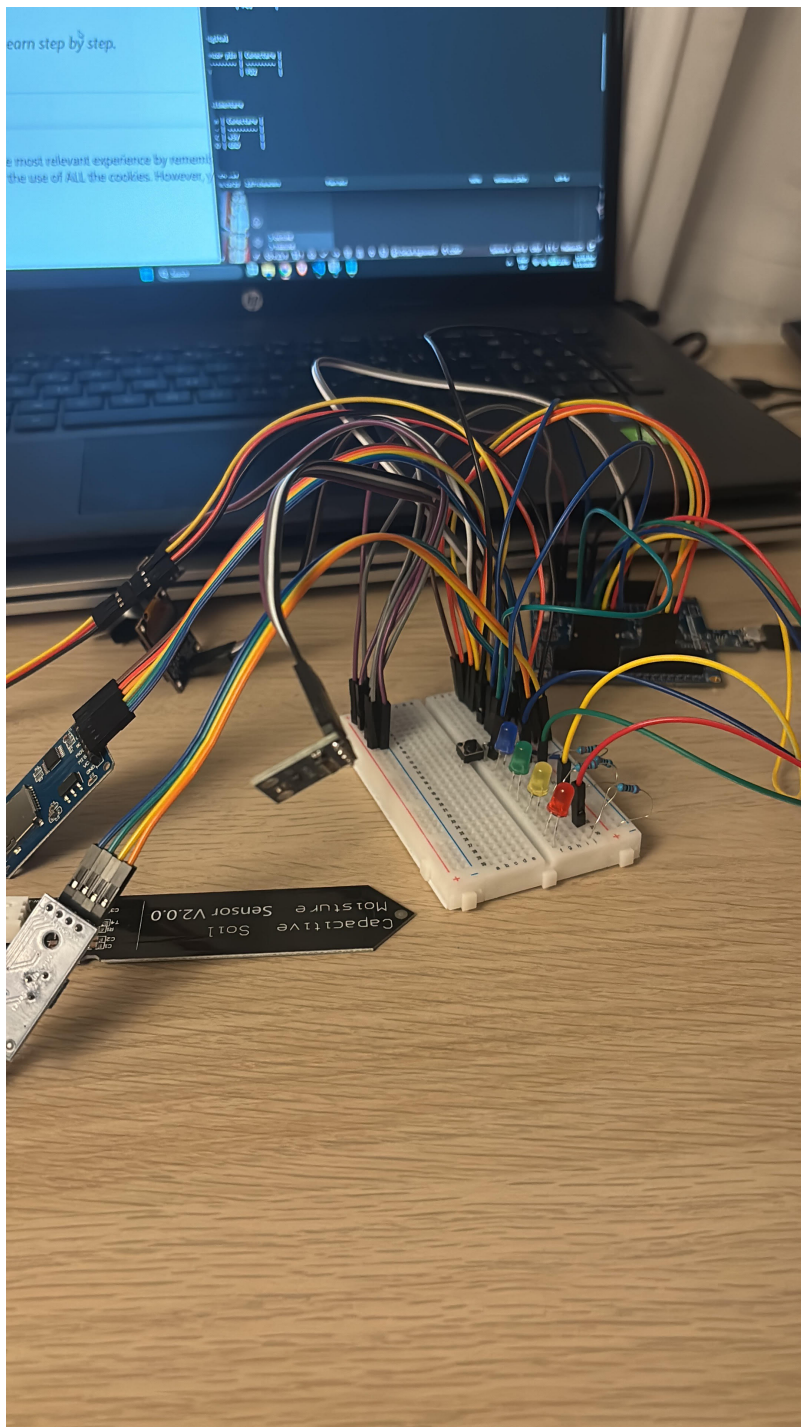
Componentă	Pin ATmega328P
Buton	PD7 și GND

Butonul este conectat între PD7 și GND.

## Sumar pini

MCU pin	Funcție
PC0	Soil sensor AO
PC1	Rain AO
PC4	SDA
PC5	SCL
PB2	SD CS
PB3	SD MOSI
PB4	SD MISO
PB5	SD SCK
PD2	Rain DO
PD3	LED albastru
PD4	LED verde
PD5	LED galben
PD6	LED roșu
PD7	Buton

## Componente conectate



[Video funcționare display OLED](#)

[Video funcționare led-uri](#)

## Software Design

**Stadiul actual al implementării software**

În stadiul actual, proiectul are implementată partea de monitorizare a condițiilor de mediu pentru o plantă. Sistemul folosește mai multe periferice hardware: ADC, GPIO, USART, I2C/TWI și SPI.

Funcționalitățile implementate până în acest moment sunt:

1. citirea senzorului de umiditate a solului prin ADC;
2. citirea senzorului de ploaie, atât pe ieșirea analogică AO, cât și pe ieșirea digitală DO;
3. citirea intensității luminoase de la senzorul digital BH1750 prin I2C;
4. citirea datei și orei de la modulul RTC DS3231 prin I2C;
5. afișarea informațiilor pe display-ul OLED SSD1306;
6. schimbarea paginilor afișate pe OLED folosind un buton;
7. semnalizarea stării sistemului folosind LED-uri;
8. transmiterea valorilor prin USART către Serial Monitor pentru debug;
9. inițializarea comunicației SPI pentru modulul MicroSD;
10. integrarea logicii de scriere în format CSV pe cardul MicroSD, folosind Petit FatFs.

Sistemul monitorizează umiditatea solului, intensitatea luminii, detectarea ploii și data/ora măsurătorilor, iar aceste valori sunt afișate pe OLED și pot fi salvate pe cardul MicroSD pentru analiză ulterioară.

Pentru MicroSD, sistemul folosește SPI pe pinii hardware ai ATmega328P: **CS = PB2, MOSI = PB3, MISO = PB4, SCK = PB5**. Datele sunt formate sub formă de linie CSV:

date	time	soil_percent	soil_raw	lux	rain	rain_raw
23/05/2026	14:30:00	48	560	34	0	967

Pentru a nu bloca funcționarea butonului și a display-ului, salvarea pe card nu se face la fiecare iterație a buclei principale, ci periodic. În varianta actuală, Serial Monitor afișează datele mai rar, iar logarea în `log.csv` este gândită să se realizeze o dată pe oră, folosind timpul real citit de la RTC.

## Biblioteci utilizate și motivarea alegerii lor

În proiect am ales să folosesc o abordare bazată pe registre AVR și module software proprii, fără biblioteci Arduino mari, precum Adafruit SSD1306 sau Wire. Motivul principal este faptul că proiectul urmărește folosirea directă a perifericelor microcontrollerului ATmega328P, într-un mod apropiat de laboratoarele de PM.

Bibliotecile și fișierele utilizate sunt:

Bibliotecă / modul	Rol în proiect
<avr/io.h>	acces direct la registrele microcontrollerului
<util/delay.h>	întârzieri scurte pentru inițializare și debounce
<stdio.h>	formatarea mesajelor cu `printf` / `snprintf`
<stdint.h>	tipuri de date fixe: `uint8_t`, `uint16_t`
<string.h>	calculul lungimii liniilor CSV cu `strlen`
usart.c / usart.h	comunicație serială pentru debug
twi.c / twi.h	comunicație I2C/TWI
ssd1306.c / ssd1306.h	driver propriu pentru display-ul OLED

adc.c / adc.h	citirea senzorilor analogici
bh1750.c / bh1750.h	citirea senzorului de lumină
rtc_ds3231.c / rtc_ds3231.h	citirea și setarea RTC-ului
spi_sd.c / spi_sd.h	comunicație SPI cu modulul MicroSD
sd.c / sd.h	nivel low-level pentru cardul SD
pff.c / pff.h	Petit FatFs pentru lucrul cu fișierul `log.csv`

Pentru OLED am folosit un driver propriu, deoarece proiectul afișează doar text și nu are nevoie de funcții grafice complexe. Astfel, codul este mai simplu și consumă mai puțină memorie.

Pentru MicroSD am ales **Petit FatFs**, deoarece este o variantă mai mică de filesystem FAT, potrivită pentru microcontrollere cu memorie limitată. Aceasta permite lucrul cu fișiere de tip `.csv`, care pot fi apoi deschise pe laptop în Excel sau orice editor text.

## Elementul de noutate al proiectului

Elementul de noutate al proiectului constă în integrarea mai multor senzori și interfețe de comunicație într-un sistem compact de monitorizare agricolă inteligentă.

Proiectul nu se limitează la un singur senzor, ci combină:

1. citire analogică pentru umiditatea solului;
2. citire analogică și digitală pentru ploaie;
3. citire digitală I2C pentru lumină;
4. timestamp real de la RTC;
5. afișare locală pe OLED;
6. semnalizare rapidă prin LED-uri;
7. salvare istorică pe MicroSD în format CSV.

Noutatea practică este faptul că sistemul oferă atât monitorizare în timp real, cât și posibilitatea de analiză ulterioară a datelor. Valorile salvate pe cardul MicroSD pot fi folosite ulterior pentru grafice, comparații sau observarea evoluției condițiilor de mediu.

Astfel, proiectul funcționează ca un mic sistem de data logging pentru agricultură inteligentă.

## Funcționalități din laborator utilizate în proiect

Proiectul folosește concepte din mai multe laboratoare PM.

### GPIO

GPIO este folosit pentru LED-uri și buton. LED-urile sunt conectate pe pinii **PD3-PD6**, iar butonul este

conectat pe **PD7** la GND, folosind pull-up intern. Conexiunile hardware pentru LED-uri și buton sunt definite și în documentația proiectului.

În cod, pinii LED sunt configurați ca ieșiri:

```
DDRD |= (1 << LED_BLUE) |  
      (1 << LED_GREEN) |  
      (1 << LED_YELLOW) |  
      (1 << LED_RED);
```

Butonul este configurat ca intrare cu pull-up intern:

```
DDRD &= ~(1 << BUTTON_PIN);  
PORTD |= (1 << BUTTON_PIN);
```

Acest lucru permite conectarea butonului direct între PD7 și GND, fără rezistență externă.

## ADC

ADC-ul este folosit pentru citirea senzorului de umiditate a solului și a ieșirii analogice de la senzorul de ploaie.

- PC0 → senzor umiditate sol AO
- PC1 → senzor ploaie AO

Valorile ADC sunt transformate în procente pentru umiditatea solului. De exemplu, o valoare ADC mare indică de obicei sol uscat, iar o valoare ADC mică indică sol umed.

## I2C / TWI

Magistrala I2C este folosită pentru trei module:

1. OLED SSD1306
2. BH1750
3. DS3231

Aceste module sunt conectate în paralel pe aceleași două fire: PC4 → SDA, PC5 → SCL.

## SPI

SPI este folosit pentru modulul MicroSD. Modulul MicroSD utilizează liniile: PB2 → CS, PB3 → MOSI, PB4 → MISO, PB5 → SCK.

## USART

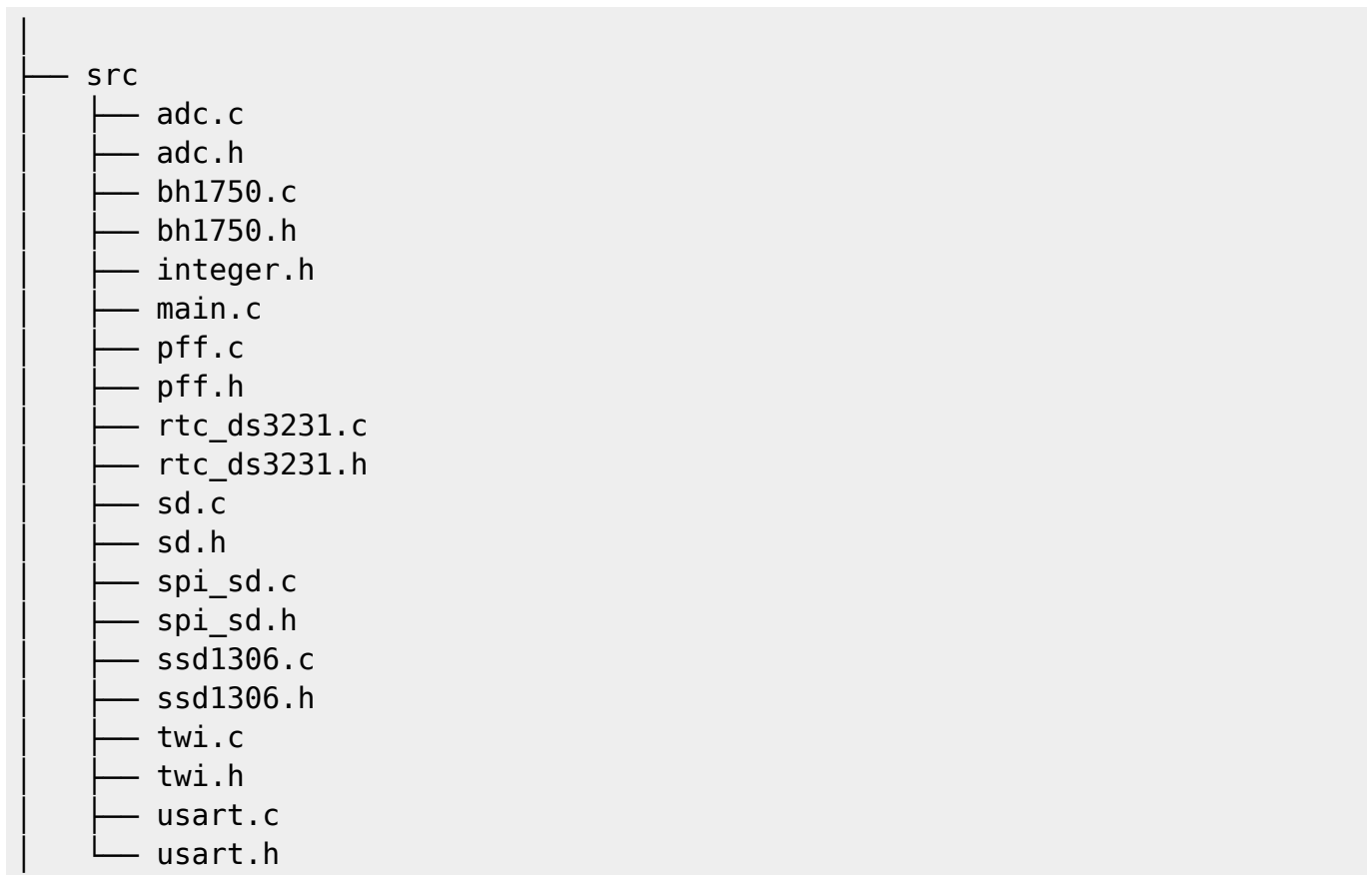
USART este folosit pentru debug prin Serial Monitor. Prin terminal sunt afișate informații precum:

- inițializarea sistemului
- scanarea I2C
- valorile senzorilor
- data și ora RTC
- statusul cardului SD
- mesajele de confirmare pentru logare CSV

## Arhitectura software

Codul proiectului este organizat modular, fiecare funcționalitate importantă având propriul fișier `.c`` și `.h``.

Structura software este:



Rolul fișierului ``main.c`` este de a coordona toate modulele. Acesta:

1. inițializează USART;
2. configurează GPIO;
3. inițializează ADC;
4. pornește magistrala I2C;

5. inițializează SPI;
6. montează cardul SD;
7. inițializează OLED-ul;
8. inițializează senzorul BH1750;
9. citește periodic senzorii;
10. actualizează LED-urile;
11. verifică butonul;
12. afișează pagina curentă pe OLED;
13. trimite date prin Serial Monitor;
14. salvează periodic date în log.csv.

Fluxul principal din `while(1)` este: citire sol → citire ploaie → citire lumină → citire RTC → actualizare LED-uri → verificare buton → afișare OLED → afișare în Serial Monitor → logare periodică pe MicroSD

Interacțiunea dintre module este:

- Sensor sol → ADC → procent umiditate → OLED + LED-uri + CSV
- Sensor ploaie → ADC/GPIO → status ploaie → OLED + LED albastru + CSV
- BH1750 → I2C → lux → OLED + CSV
- DS3231 → I2C → dată/oră → OLED + timestamp CSV
- Buton → GPIO → schimbare pagină OLED
- MicroSD → SPI + Petit FatFs → salvare log.csv
- USART → debug

## Validarea funcționalităților

Validarea proiectului s-a făcut incremental, pentru a identifica mai ușor eventualele probleme.

### Validarea OLED-ului

Prima etapă a fost testarea display-ului OLED prin afișarea unor mesaje simple:

```
HELLO
OLED OK
ATMEGA328P
```

#### [Video validare](#)

După confirmarea afișării corecte, au fost adăugate paginile cu date reale de la senzori.

### Validarea I2C

Pentru validarea magistralei I2C am folosit o funcție de scanare, care afișează în Serial Monitor adresele dispozitivelor detectate. Prin această metodă am putut verifica dacă OLED-ul, BH1750 și DS3231 sunt conectate corect.

## Validarea ADC

Senzorul de umiditate a fost testat prin citirea valorii brute ADC în două situații:

- senzor în aer / sol uscat
- senzor în sol umed / apă

Valoarea brută a fost afișată în Serial Monitor și pe OLED, iar apoi transformată într-un procent.

## Validarea senzorului de ploaie

Pentru senzorul de ploaie am verificat atât ieșirea analogică, cât și ieșirea digitală. Ieșirea digitală DO controlează logica pentru LED-ul albastru, iar ieșirea analogică AO este afișată și poate fi salvată în CSV.

## Validarea RTC

RTC-ul DS3231 a fost validat prin citirea și afișarea orei pe OLED și în Serial Monitor. Inițial, modulul pornea de la data 01/01/2000, ceea ce a indicat că ora nu era setată sau că modulul nu avea baterie de backup. Pentru rezolvare, am adăugat o funcție `rtc_set_time()`, folosită o singură dată pentru setarea datei și orei.

## Validarea MicroSD

Pentru MicroSD, validarea se face în două etape:

1. inițializarea SPI și montarea cardului;
2. deschiderea fișierului `log.csv` și scrierea unei linii CSV.

În Serial Monitor sunt afișate mesaje precum: SD mounted OK, log.csv opened OK, SD log OK.

După rulare, cardul poate fi scos și verificat pe laptop, unde fișierul `log.csv` poate fi deschis pentru a confirma salvarea datelor.

## Demo video

[Partea 1](#) [Partea 2](#)

Demo-ul include următoarele etape:

1. pornirea sistemului; (1)
2. afișarea mesajelor de inițializare în Serial Monitor; (1)
3. afișarea primei pagini pe OLED cu umiditatea solului; (1)
4. apăsarea butonului și schimbarea paginilor OLED; (1)
5. afișarea datei și orei de la RTC;
6. demonstrarea LED-urilor pentru starea solului; (1)(2)
7. acoperirea sau iluminarea senzorului BH1750 pentru modificarea valorii lux; (2)
8. umezirea senzorului de ploaie pentru aprinderea LED-ului albastru; (2)

În acest demo este prezentat sistemul AgriSmart Mini. La pornire, microcontrollerul inițializează comunicațiile USART, I2C și SPI. Display-ul OLED afișează valorile citite de la senzori, iar butonul permite schimbarea paginilor. LED-urile oferă feedback vizual pentru starea solului și pentru detectarea ploii. Senzorul BH1750 măsoară intensitatea luminii, iar RTC-ul DS3231 oferă data și ora curentă. Periodic, datele sunt formate ca linie CSV și salvate pe cardul MicroSD pentru analiză ulterioară.

## Calibrarea senzorilor

### Calibrarea senzorului de umiditate a solului \*

Senzorul de umiditate a solului oferă o valoare analogică, citită prin ADC. Deoarece senzorul nu oferă direct un procent, am realizat o calibrare pe baza valorilor brute.

Am măsurat valoarea ADC în două situații: sol foarte umed / senzor în apă → valoare ADC mică, sol uscat / senzor în aer → valoare ADC mare.

Pe baza acestor valori am ales două praguri:

```
const uint16_t wet = 300;  
const uint16_t dry = 800;
```

Apoi valoarea ADC este transformată într-un procent:

100% → sol foarte umed

0% → sol foarte uscat

Această transformare este necesară pentru ca utilizatorul să poată interpreta mai ușor starea plantei.

### Calibrarea senzorului de ploaie

Senzorul de ploaie are două ieșiri: AO → valoare analogică pentru nivelul de apă, DO → semnal digital de detecție peste un prag

Pragul ieșirii digitale a fost reglat din potențiometrul modulului. Am testat senzorul în stare uscată și apoi cu picături de apă pe suprafața de detecție. Scopul a fost ca ieșirea digitală să se activeze doar atunci când există apă pe senzor.

## Calibrarea BH1750

BH1750 este un senzor digital de lumină, cu ADC intern, deci nu a necesitat calibrare analogică. Validarea lui s-a realizat prin testarea în mai multe condiții:

- senzor acoperit → valoare lux mică
- lumină ambientală → valoare medie
- lumină puternică / lanternă → valoare mare

Valoarea citită este afișată direct în lux.

## Calibrarea RTC DS3231

RTC-ul DS3231 a necesitat setarea inițială a datei și orei. Am implementat o funcție de setare a timpului, care scrie valorile în registrele RTC în format BCD. După setarea orei, această funcție trebuie comentată pentru a nu reseta timpul la fiecare pornire.

Bateria modulului RTC are rolul de a păstra ora atunci când sistemul nu este alimentat.

## Optimizări realizate

### 1. Evitarea bibliotecilor mari

Nu am folosit biblioteci externe mari pentru OLED sau I2C. În schimb, am folosit drivere proprii, mai simple și mai potrivite pentru nevoile proiectului. Astfel, codul este mai ușor de înțeles și consumă mai puțină memorie.

### 1. Folosirea unei magistrale I2C comune

OLED-ul, BH1750 și RTC-ul folosesc aceeași magistrală I2C. Astfel, trei module diferite sunt conectate folosind doar două fire de comunicație: SDA și SCL.

### 1. Afișarea pe pagini OLED

Display-ul OLED are spațiu limitat, deci informațiile sunt împărțite în mai multe pagini. Această abordare face interfața mai clară și permite afișarea mai multor tipuri de date fără aglomerarea ecranului.

### 1. Verificarea frecvență a butonului

Într-o variantă inițială, delay-ul principal era prea mare, ceea ce făcea ca butonul să fie verificat rar. Am optimizat bucla principală prin folosirea unui delay mic, astfel încât butonul să rămână responsiv.

### 1. Răirea mesajelor din Serial Monitor

Afișarea în Serial Monitor nu se face la fiecare iterație a buclei, deoarece ar produce foarte multe mesaje și ar îngreuna urmărirea informațiilor. În varianta optimizată, datele sunt afișate mai rar, de exemplu o dată pe minut.

### 1. Răirea scrierilor pe MicroSD

Scrierea pe cardul MicroSD nu se face continuu. În proiect, logarea este gândită să se facă periodic, de exemplu o dată pe oră. Această optimizare reduce numărul de scrieri pe card și face fișierul CSV mai ușor de analizat.

### 1. Folosirea RTC-ului pentru timestamp real

În loc să folosesc doar delay-uri pentru măsurarea timpului, logarea este asociată cu ora reală primită de la RTC. Astfel, fiecare linie din CSV are context temporal clar.

## Rezultate obținute

În urma implementării, sistemul poate citi valorile de la senzorii principali, poate afișa datele pe OLED, poate semnaliza starea plantei prin LED-uri și poate transmite informații prin USART pentru debug.

Datele citite au format de tipul:

```
SOIL=48% RAW=560 LUX=34 RAIN=0 RAIN_RAW=967 TIME=14:30:00 DATE=23/05/2026
```

În fișierul CSV, datele sunt salvate sub forma:

date	time	soil_percent	soil_raw	lux	rain	rain_raw
23/05/2026	14:30:00	48	560	34	0	967

Acest format permite analiza ulterioară a valorilor pe laptop.

## Concluzii

Proiectul **AgriSmart Mini** demonstrează integrarea mai multor periferice hardware ale microcontrollerului ATmega328P într-un sistem de monitorizare agricolă inteligentă. Sistemul combină senzori analogici, senzori digitali, afișare locală, feedback vizual și stocare pe MicroSD.

Prin folosirea ADC, I2C, SPI, GPIO și USART, proiectul acoperă o parte importantă din funcționalitățile studiate în laborator. Implementarea modulară permite extinderea ulterioară a sistemului.

## Jurnal de dezvoltare

- 1 Mai - Stabilirea ideii proiectului și alegerea componentelor.
- 10 Mai - Realizarea schemei hardware și a conexiunilor principale.
- 15 Mai - Testarea OLED-ului și a comunicației I2C.
- 18 Mai - Testarea senzorului de umiditate și a senzorului de ploaie.
- 19 Mai - Integrarea BH1750 și RTC DS3231.
- 20 Mai - Implementarea paginilor OLED, LED-urilor și butonului.
- 22 Mai - Integrarea SPI și a logicii MicroSD pentru salvare CSV.

- 23 Mai - Finalizarea documentației software și pregătirea demo-ului.

## Bibliografie

1. Documentația oficială ATmega328P / registre AVR.
2. Laboratoarele PM: GPIO, UART, ADC, SPI, I2C.
3. Documentația senzorului BH1750.
4. Documentația modulului RTC DS3231.
5. Documentația modulului OLED SSD1306.
6. Documentația Petit FatFs.

[Export to PDF](#)

## GitHub Repository

<https://github.com/Miruna-Brindusesc/hw-agrismart-mini>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/alexandru.jipa2803/miruna.brindusescu>



Last update: **2026/05/23 16:52**