

Mini Stație Meteo cu afișaj digital

Introducere

Acest proiect constă într-o stație meteorologică inteligentă, capabilă să măsoare și să proceseze în timp real temperatura, umiditatea relativă, presiunea atmosferică și nivelul de iluminare ambientală. Datele sunt afișate pe un ecran LCD prin protocolul I2C. Sistemul este unul interactiv și adaptiv: utilizatorul poate naviga între pagini de monitorizare, poate schimba unitățile de măsură (C/F/K, mmHg/hPa, m/ft) și poate configura pragurile de alertă printr-un meniu dedicat. Experiența este completată de un sistem de feedback vizual bazat pe tranziții de culoare (LED RGB) și un sistem acustic dinamic (Buzzer).

Scopul proiectului: Crearea unui instrument de precizie pentru monitorizarea microclimatului, oferind utilizatorului nu doar date brute, ci și informații procesate (prognoză, punct de rouă, altitudine) și alerte personalizabile.

Ideea de la care am pornit: Proiectul a evoluat de la un simplu experiment cu semnale PWM la un sistem complex de monitorizare a mediului. Am păstrat componenta de feedback reactiv, dar am transformat-o într-o interfață de comunicare non-verbală cu utilizatorul: culorile LED-ului nu sunt doar decorative, ci reprezintă o hartă termică a confortului.

Utilitate: Stația este esențială pentru optimizarea spațiului de lucru, prevenind disconfortul termic sau oboseala oculară (prin monitorizarea luxilor). Pentru dezvoltatori, codul reprezintă un studiu de caz în optimizarea resurselor AVR: utilizarea EEPROM pentru salvarea setărilor, implementarea algoritmului Magnus-Tetens pentru punctul de rouă și filtrarea digitală a semnalelor barometrice.

Laboratoare folosite: ADC (Lumină), I2C (Senzori & LCD), PWM și Timere (RGB & Buzzer), Întreruperi (Butoane).



Demo video: youtube.com

Descriere generală

Arhitectura software a sistemului este construită în jurul unei bucle de monitorizare non-blocking care gestionează achiziția de date de la senzori digitali de înaltă precizie (AHT20 și BMP280). Sistemul este versatil, operând în două moduri principale: Modul Monitorizare (afișarea datelor pe 3 pagini distincte) și Modul Meniu (configurarea pragurilor de operare).

1. Interfața și Navigarea (Control prin Întreruperi) Utilizatorul interacționează cu stația prin trei butoane gestionate prin întreruperi (INT1 și PCINT2), asigurând un răspuns instantaneu al interfeței:

Navigare Pagini: Comutarea între temperatura/umiditate, presiune/prognoză și altitudine/lux.

Meniu Configurare: Prin apăsarea lungă a butonului principal, se intră într-o interfață ce permite setarea pragurilor T_{min} , T_{max} , H_{low} , H_{high} și a offset-ului de altitudine. Aceste valori sunt salvate în EEPROM, fiind persistente la oprirea alimentării.

Conversie Unități: Schimbarea în timp real a scărilor de măsură (Celsius, Fahrenheit, Kelvin etc.).

2. Feedback Automatizat și Logică de Decizie

A. Feedback Acustica Dinamic (Buzzer PWM) Sistemul integrează un buzzer pasiv controlat prin Timer 0 în mod Fast PWM. Spre deosebire de o alarmă on/off, intensitatea sunetului este proporțională cu gravitatea abaterii termice.

Gestiune Inteligentă: Alerta sonoră este inhibată automat în condiții de lumină scăzută (Mod Noapte, sub 10 Lux) sau manual prin funcția Mute, prevenind deranjul fonic inutil.

B. Feedback Optic Avansat (Tranziții RGB & Indicatori Umiditate) Sistemul utilizează o logică de semnalizare stratificată pentru a oferi o citire rapidă a confortului ambiental:

Indicator Termic (LED RGB): Implementează o tranziție fluidă de culori folosind funcția map() pe 3 canale PWM.

Albastru: Temperatură sub pragul T_{min} .

Verde (Optim): Când temperatura tinde spre valoarea T_{opt} .

Roșu: Temperatură peste pragul T_{max} .

Culori intermediare: Sistemul generează nuanțe de tranziție (ex. galben/portocaliu) pe măsură ce temperatura urcă spre pragul critic.

Indicator Umiditate (LED-uri Discrete): Trei ieșiri digitale dedicate semnalizează starea aerului:

Albastru (Low): Umiditate sub pragul de confort (aer uscat).

Verde (OK): Intervalul ideal de confort (setat prin meniu).

Roșu (High): Umiditate excesivă.

3. Procesare Digitală și Calcule Complexe Filtrare Barometrică: Presiunea este trecută printr-un filtru de medie mobilă (smooth_pressure) pentru a elimina fluctuațiile cauzate de curenții de aer.

Prognoză și Altitudine: Sistemul calculează prognoza bazată pe tendința barometrică și altitudinea relativă folosind formula barometrică, compensată cu datele de calibrare citite direct din senzorul BMP280.

Punct de Rouă: Calculat matematic pentru a avertiza utilizatorul asupra condițiilor de condens.

Hardware Design

1. Stadiul actual al implementării

Proiectul este funcțional la nivel hardware, toate componentele fiind conectate și testate pe breadboard. Firmware-ul scris în C/C++ (Arduino framework) comunică cu senzorii prin I2C, citește luminozitatea prin ADC, afișează datele pe LCD și oferă feedback vizual (LED RGB) și sonor (buzzer PWM).

Piesa	Link	Tip Utilizare
ATMEGA328 microcontroler	Placa dezvoltare Arduino UNO R3 ATMEGA328P	Baza dezvoltare
Modul senzor AHT20 si BMP280	Modul senzor temperatura, umiditate, presiune	I2C
Modul senzor de lumina ambientala TEMENT6000	Modul senzor lumina ambientala	ADC
LCD cu convertor I2C	LCD cu convertor I2C	I2C
Buzzer pasiv 3.3V sau 3V	Buzzer pasiv	PWM
Modul cu buton	Modul cu buton	GPIO, Intreruperi
Led RGB cu catod comun	Leg RGB cu catod comun	GPIO (posibila modificare spre PWM)

Pentru a evita posibilele riscuri de scurt circuit pentru fiecare led legat am folosit in serie cate o rezistenta de 470Ω, iar pentru butoane o rezistenta de pull-up de 10kΩ, iar pentru o conectare facila a diverselor piese componente am folosit un breadboard de 800 de puncte si fire de legatura mama-tata si tata-tata alaturi de pini.

3. Descrierea detaliată a pinilor folosiți

Magistrala I2C — A4 (SDA) și A5 (SCL)

Pinii A4/A5 si SDA/SCL sunt pini hardware I2C ai ATmega328P, ceea ce permit comunicarea simultană cu trei dispozitive pe aceeași magistrală:

Dispozitiv	Adresă I2C
LCD 16x2 cu convertor I2C	0x27
Senzor AHT20 (temperatură/umiditate)	0x38
Senzor BMP280 (presiune)	0x77

Adresele distincte permit coexistența fără conflicte și fără consum suplimentar de pini digitali.

Butoane — D2, D3, D4

Pin	Funcție	Tip întrerupere	Rol
D2	Navigare / Meniu (apăsare lungă 2s)	GPIO polling	Schimbare pagină sau intrare/ieșire meniu setări
D3	Plus / Mute	INT1 — front descendent (ISC11=1)	Incrementare valori în meniu sau activare/dezactivare sunet
D4	Minus / Schimbare unități	PCINT20 — Pin Change Interrupt	Decrementare valori în meniu sau comutare unități de măsură

D3 este conectat la **INT1** (întrerupere externă dedicată) pentru reacție imediată. D4 folosește **PCINT2** deoarece INT0 este rezervat pentru D2. Fiecare buton are rezistență pull-up de 10kΩ în

circuit și pull-up intern activat în firmware:

```
PORTD |= (1 << PORTD2) | (1 << PORTD3) | (1 << PORTD4);
```

PWM — D5, D6, D9, D10

Pin	Registru	Periferic	Justificare
D5 (PD5)	Timer0, OC0B	Buzzer pasiv	Intensitatea sunetului proporțională cu severitatea temperaturii
D6 (PD6)	Timer0, OC0A	LED Albastru (canal B al RGB)	Timer0 partajat cu buzzerul, economisind un timer
D9 (PB1)	Timer1, OC1A	LED Verde (canal G al RGB)	Timer1 pe 16 biți — rezoluție mai bună pentru tranziții de culoare
D10 (PB2)	Timer1, OC1B	LED Roșu (canal R al RGB)	Timer1 pe 16 biți, canal B

Intensitatea buzzerului se calculează astfel:

```
float severity = (t >= settings.t_max) ? (t - settings.t_max) : (settings.t_min - t);
if (severity > 10.0f) severity = 10.0f;
OCR0B = (uint8_t)((severity / 10.0f) * 255.0f);
```

LED-uri discrete umiditate — D11, D12, D13 (PB3, PB4, PB5)

Pini GPIO simpli pe portul B, comutați direct prin registrul PORTB. Nu necesită PWM, doar stări on/off în funcție de pragurile `h_low` și `h_high`:

Pin	Culoare LED	Condiție de activare
D11 (PB3)	Albastru	$h < \text{settings.h_low}$ — umiditate scăzută
D12 (PB4)	Verde	$h_low \leq h \leq h_high$ — umiditate optimă
D13 (PB5)	Roșu	$h > \text{settings.h_high}$ — umiditate ridicată

ADC — A3 Senzorul TEMENT6000 produce o tensiune analogică proporțională cu luminozitatea.

```
ADMUX = (1 << REFS0) | (1 << MUX1) | (1 << MUX0); // Ref AVcc, canal ADC3
ADCSRA = (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0); //
Prescaler 128
```

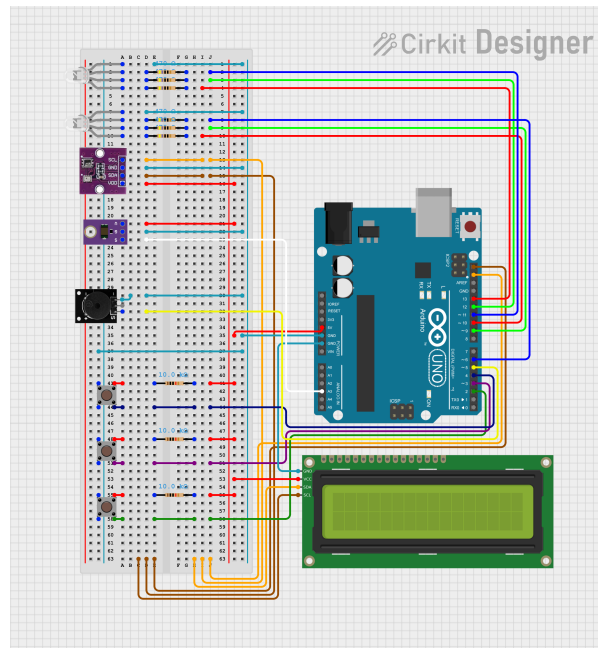
Valoarea citită este convertită în lux:

```
float lux = (float)ADC * 0.9765f;
```

Luminozitatea sub 10 lux activează modul nocturn (`night_mute`), care dezactivează automat buzzerul și afișează indicatorul [N] pe LCD.

4. Schema electrică

Schema a fost realizată în **Circuit Designer** și prezintă toate componentele conectate pe breadboard-ul de 800 de puncte.



Magistrala I2C leagă în paralel modulul AHT20+BMP280 și LCD-ul 16x2 prin pinii GND, VCC, SDA, SCL.

Butoanele sunt conectate cu rezistențe pull-up de 10kΩ între VCC și pinul butonului, celălalt capăt la GND.

LED-urile au rezistențe de 470Ω în serie pe fiecare anod, limitând curentul la ~10mA la 5V.

Buzzerul pasiv este conectat direct la D5 cu GND comun.

TEMT6000 are ieșirea analogică conectată la A3.

5. Dovezi de funcționare

Schema din Cirkuit Designer confirmă conectarea corectă a tuturor componentelor. Dovada funcționării magistralei I2C este dată de prezența celor trei dispozitive cu adrese distincte (0x27, 0x38, 0x77) pe aceleași linii SDA/SCL, fără conflicte.

La alimentarea circuitului, secvența de boot din firmware afișează mesajul de bun venit pe LCD timp de 1,5 secunde, confirmând funcționarea comunicației I2C:

```
lcd_print("  Statie Meteo  ");
lcd_send(0xC0, LCD_CMD);
lcd_print("Nicolaescu Alex");
delay(1500);
```



Software Design

1. Mediu de dezvoltare

[GITHUB](#) Proiectul a fost dezvoltat utilizând **PlatformIO** (în cadrul VS Code) și framework-ul **Arduino**, dar cu o abordare orientată către programarea la nivel de regiștri specifică **AVR-GCC**. Această combinație a permis utilizarea unor funcții utilitare de sistem (precum `millis()`), păstrând în același timp controlul total asupra perifericelor microcontrolerului ATmega328P.

2. Librării și surse 3rd-party

Pentru a maximiza eficiența și a reduce amprenta de memorie, codul **nu utilizează biblioteci comerciale grele** (precum cele de la Adafruit sau SparkFun). În schimb, s-au folosit:

- **avr/interrupt.h:** Pentru gestionarea vectorilor de întrerupere (butoane).
- **avr/eeprom.h:** Pentru stocarea permanentă a pragurilor de alertă în mod eficient prin `eeprom_read_block` și `eeprom_update_block`.
- **math.h:** Pentru implementarea formulelor logaritmice (punct de rouă) și a puterilor (altitudine).
- **Driver I2C Custom:** Implementare proprie pentru protocolul Two-Wire Interface (TWI) pentru comunicarea cu senzorii și LCD-ul, bazată pe manipularea directă a regiștrilor TWDR, TWCR, TWSR.

3. Algoritmi și structuri de date

- **Filtru de Medie Mobilă (Moving Average):** Implementat în funcția `smooth_pressure` folosind un buffer circular de 10 eșantioane pentru a stabiliza citirile barometrice și a evita fluctuațiile cauzate de zgomot.
- **Algoritmul Magnus-Tetens:** Utilizat pentru calcularea punctului de rouă pe baza temperaturii și umidității.
- **Formula Barometrică:** Calculul altitudinii relative folosind raportul dintre presiunea curentă și o presiune de referință (P_0).
- **Structura DeviceSettings:** O structură de date care grupează toate pragurile de mediu și un Magic Number (0x44) pentru validarea integrității datelor citite din EEPROM la startup.
- **Mapare PWM Proporțională:** Un algoritm care calculează severitatea abaterii termice și ajustează factorul de umplere (Duty Cycle) pentru buzzer și culorile LED-ului RGB.

4. Surse și funcții implementate

Codul este modularizat pe funcționalități cheie, cele mai importante fiind:

Gestionarea Hardware (Low-Level)

- `i2c_init()`, `i2c_start()`, `i2c_write()`, `i2c_read()`: Setul de funcții care gestionează magistrala I2C la frecvența de 100kHz.
- `lcd_send()`, `lcd_init()`: Driverul pentru ecranul LCD 1602, care traduce comenzile în nibbles (4 biți) pentru interfața I2C a modulului PCF8574.
- **Timere (PWM):** Configurare manuală a `Timer0` (pentru Buzzer și Blue LED) și `Timer1` (pentru Red și Green LED) în modul Fast PWM.

Achiziție și Procesare Date

- `bmp_read_calibration()`: Funcție critică ce citește coeficienții unici de calibrare din memoria senzorului BMP280.
- `bmp_compensate_temp()` & `bmp_compensate_pressure()`: Implementarea calculelor matematice pe 32 și 64 de biți conform specificațiilor Bosch pentru obținerea datelor brute.
- `update_leds(t, h)`: Logica de control a culorilor, care decide tranziția între albastru-verde-roșu în funcție de confortul termic.

- `dynamic_buzzer(t, lux)`: Funcția de siguranță care modulează sunetul și verifică starea de Mute (manuală sau nocturnă).

Interfață și Întreruperi

- `ISR(INT1_vect)`: Gestionează butonul de “Plus / Mute”.
- `ISR(PCINT2_vect)`: Gestionează butonul de “Minus / Schimbare Unități”.
- **Logică Meniu**: Implementată în `loop()`, gestionează starea de Long Press (2 secunde) pe pinul D2 pentru a comuta între modul de afișare și cel de editare a pragurilor salvate în EEPROM.

Fluxul Principal (loop)

Programul urmează o structură de tip mașină de stări:

1. Verifică starea butoanelor (prin întreruperi).
2. Achiziționează date de la AHT20, BMP280 și ADC (LDR).
3. Procesează datele (filtrare, conversii).
4. Actualizează ieșirile PWM (LED-uri, Buzzer).
5. Actualizează afișajul LCD în funcție de pagina selectată sau meniul activ.

5. Stadiul actual al implementării software

Implementarea firmware-ului pentru ministația meteo este **completă și complet funcțională**. Toate modulele software descrise sunt integrate într-o mașină de stări stabilă, de tip non-blocking. Achiziția de la senzori, algoritmul de filtrare a datelor, calculul punctului de rouă, controlul PWM proporțional pentru LED/Buzzer, interfața de navigare prin pagini și meniul interactiv cu salvare în EEPROM rulează asincron, fără blocaje ale buclei principale.

6. Elementul de noutate al proiectului

Spre deosebire de stațiile meteo convenționale de nivel hobby care doar afișează valori brute, acest proiect aduce ca elemente de noutate:

- **Feedback-ul acustic și vizual proporțional**: Intensitatea buzzer-ului și tranzițiile cromatice ale LED-ului RGB nu sunt binare (ON/OFF), ci sunt mapate matematic utilizând gradul de severitate al abaterii termice față de limitele de confort.
- **Sistemul Smart Mute adaptiv**: Dispozitivul integrează o funcție autonomă de diminuare a deranjului fonic; dacă senzorul TEMT6000 indică o valoare sub 10 lx (**Mod Noapte**), buzzerul se dezactivează automat, prioritatea trecând pe indicarea optică.

- **Meniu de calibrare hardware persistent:** Permite calibrarea offset-ului de altitudine direct din butoane, fără necesitatea rescrierii firmware-ului.

7. Justificarea utilizării funcționalităților din laborator

Laborator	Funcționalitate în Proiect	Justificare Tehnică
GPIO	Direcția pinilor și pull-up	Configurația directă a regiștrilor DDRB, DDRD și PORTD pentru controlul LED-urilor discrete de umiditate și activarea pull-up-urilor interne pentru butoane.
Înteruperi	INT1 și PCINT2	Folosite pentru detectarea apăsării butoanelor de pe pinii D3 și D4. Asigură comutarea unităților și modificarea pragurilor în mod asincron, cu debouncing software de 250ms.
I2C (TWI)	Comunicație master-slave	Implementarea mașinii de stări pentru controlul magistralei hardware (regiștrii TWCR, TWDR, TWSR) la 100kHz pentru citirea senzorilor și scrierea datelor pe LCD-ul cu expander PCF8574.
PWM & Timere	Fast PWM pe 8 biți	Utilizarea Timer0 (canalele A și B pentru Blue LED și Buzzer) și Timer1 (canalele A și B pentru Red și Green LED) în mod Fast PWM pentru controlul intensității și al culorilor.
ADC	Citire analogică	Configurat pe pinul A3 (regiștrii ADMUX, ADCSRA) cu prescaler de 128 pentru conversia semnalului generat de TMT6000 în unități de luxi.

8. Scheletul proiectului și interacțiunea funcționalităților

Sistemul este structurat sub forma unei arhitecturi conduse de evenimente combinată cu o mașină de stări ciclică. Interacțiunea se realizează astfel:

- **Gestiunea intrărilor (Evenimente):** Apăsarea butoanelor generează înteruperi care modifică variabilele globale volatile (`temp_unit`, `display_page`, `manual_mute`) sau modifică pragurile în mod direct dacă `in_menu` este activ.
- **Mașina de stări din loop:** Dacă `in_menu` este activ, sistemul oprește citirea senzorilor și afișează interfața de editare. La ieșire, se apelează automat `save_to_eeprom()`. Dacă `in_menu` este fals, se inițiază secvențial citirea I2C. Datele de presiune sunt trimise către filtrul `smooth_pressure()`, iar temperatura și umiditatea sunt folosite ca argumente pentru `calculate_dew_point()`.
- **Controlul ieșirilor:** Datele procesate sunt trimise către `update_leds()` și `dynamic_buzzer()` care modifică regiștrii de comparare PWM (`OCR0A`, `OCR0B`, `OCR1A`, `OCR1B`), în timp ce ecranul LCD este actualizat cu string-urile formate prin `dtostrf()`.

Validarea funcționării: Sistemul a fost validat prin monitorizarea datelor trimise în paralel pe un port serial virtual, cât și prin introducerea stației în medii cu parametri controlați (sursă de căldură pentru testarea tranziției RGB în roșu și pornirea buzzerului, respectiv acoperirea senzorului LDR pentru validarea modului de noapte).

9. Calibrarea elementelor de senzorică

Pentru a asigura acuratețea datelor afișate și a compensa erorile de poziționare sau toleranțele componentelor hardware, s-au implementat trei niveluri de calibrare:

- **Calibrare software fixă (Offsets):** S-au definit constantele `AHT20_T_OFFSET = -0.8f` (pentru a corecta căldura radiată de componentele din jur pe breadboard) și `BMP280_P_OFFSET = 600.0f` (pentru alinierea cu presiunea stației de referință).
- **Compensare dinamică din Datasheet (BMP280):** În funcția `bmp_read_calibration()`, la pornirea sistemului sunt citiți din memoria nevolatilă a senzorului cei 11 coeficienți unici de calibrare din fabrică (`dig_T1-T3`, `dig_P1-P9`). Funcțiile `bmp_compensate_temp()` și `bmp_compensate_pressure()` aplică ecuațiile de compensare pe 32 și 64 de biți conform specificațiilor Bosch.
- **Calibrare interactivă (Alitudine):** Deoarece presiunea la nivelul mării se modifică zilnic, s-a introdus în meniu parametrul `altitude_offset`, permițând utilizatorului să introducă altitudinea exactă a locației curente pentru calibrarea punctului zero al algoritmului barometric.

10. Optimizări realizate

- **I2C și LCD:** Înlocuirea funcțiilor din biblioteca standard Arduino cu funcții custom bazate pe manipulare de biți și așteptare pe flag-ul hardware TWINT. **De ce:** Reduce dimensiunea codului compilat (Flash) cu aproximativ 4KB și mărește viteza de reîmprospătare a ecranului.
- **Funcțiile de afișare:** Înlocuirea funcției standard `sprintf` (care conține un overhead imens pentru parsarea tipurilor float) cu funcția de sistem mult mai rapidă `dtostrf()` pentru conversia valorilor float în siruri de caractere. **De ce:** Microcontrolerul nu are unitate de calcul în virgulă flotantă (FPU) hardware; optimizarea reduce semnificativ timpul de execuție al buclei.
- **Scrierea în EEPROM:** Utilizarea funcției `eeprom_update_block` în loc de `eeprom_write_block`. **De ce:** Funcția de update verifică dacă noua valoare diferă de cea deja salvată și efectuează scrierea doar dacă este necesar, protejând durata de viață a celulelor EEPROM (limitate la 100.000 de cicluri de scriere).

11. Demo Video Proiect

Deoarece site-ul este public, prezentarea video și demonstrarea funcționalităților complete au fost încărcate în platforma securizată de stocare a facultății.

Explicații conținut video:

1. **Inițializare:** Se observă afișarea ecranului de bun venit timp de 1.5 secunde.
2. **Navigare:** Se prezintă apăsarea scurtă a butonului D2 pentru a trece prin cele 3 pagini de date.
3. **Schimbarea unităților:** Se demonstrează apăsarea butonului D4 care schimbă unitățile (C/F/K, respectiv mmHg/hPa).
4. **Meniu de setări:** Se evidențiază apăsarea lungă (2s) pe D2, modificarea pragului `T_max` folosind butoanele D3 și D4, urmată de salvarea automată în EEPROM.
5. **Declanșarea alertelor:** Se apropie o sursă de căldură de senzor; se observă tranziția fluidă a

LED-ului RGB din verde în roșu și intensificarea sunetului PWM pe măsură ce temperatura depășește pragul limită. Apăsarea butonului Mute oprește imediat alerta acustică (apare indicatorul [M]).

Rezultate Obținute

În urma implementării și testării stației meteo, s-au obținut următoarele rezultate ce confirmă atingerea obiectivelor propuse:

1. Monitorizarea Mediului cu Înaltă Precizie Sistemul a demonstrat o capacitate de achiziție de date stabilă, integrând cu succes senzori cu protocoale diferite (I2C și ADC):

* **Acuratețe Termică:** Prin senzorul AHT20, sistemul oferă citiri cu o precizie ridicată. Calculul punctului de rouă (Dew Point) este realizat în timp real prin formula Magnus-Tetens, oferind informații despre riscul de condens. * **Stabilitate Barometrică:** Implementarea filtrului de medie mobilă (Moving Average) în funcția `smooth_pressure` a eliminat fluctuațiile citirilor BMP280, rezultând o afișare stabilă a presiunii. * **Senzitivitate Luminoasă:** Fotorezistorul permite maparea nivelului de luxi, facilitând funcția de **Mute Automat** pe timp de noapte pentru a proteja confortul utilizatorului.

2. Interfață Utilizator Adaptivă Afișajul LCD 16x2 a fost optimizat pentru a prezenta datele într-o formă structurată pe **trei pagini dinamice**, navigabile prin butoane:

Pagina	Parametri Afișați	Unități de Măsură
0	Temperatură, Umiditate, Punct Rouă	Celsius, Fahrenheit, Kelvin / %
1	Presiune Atmosferică, Prognoză Meteo	mmHg, hPa
2	Altitudine Relativă, Luminozitate	metri, picioare (ft) / lux

Comutarea unităților de măsură și a paginilor se realizează instantaneu prin utilizarea întreruperilor externe (**INT1** și **PCINT2**).

3. Managementul Setărilor și Persistența Datelor Un rezultat major este implementarea meniului de configurare interactiv:

* **Editare în timp real:** Utilizatorul poate modifica pragurile de alertă (T min/max, H low/high) direct din interfața hardware. * **Stocare Permanentă:** Valorile sunt salvate în **EEPROM**, fiind păstrate chiar și după întreruperea alimentării. * **Siguranță:** Validarea datelor prin `Magic Number` previne erorile de sistem la prima inițializare a memoriei.

4. Feedback Senzorial Inteligent

Feedback Vizual (Sistem LED) S-a implementat o logică de semnalizare optică stratificată:

* **Indicator Termic (RGB):** Tranziție fluidă de culori (Albastru → Verde → Roșu) gestionată prin PWM, oferind o diagnoză vizuală imediată a confortului termic. * **Indicator Umiditate (Discrete):** Trei LED-uri dedicate semnalizează starea aerului (Uscat, Optim sau Umed) conform pragurilor setate de utilizator.

Feedback Acustic (Buzzer PWM) Modulul audio acționează ca un sistem de alertă inteligent:

* **Dinamică Sonoră:** Sunetul generat prin Timer 0 își schimbă intensitatea în funcție de severitatea abaterii de temperatură. * **Inhibare Automată:** Alerta este dezactivată automat în condiții de întuneric (sub 10 lx) sau manual prin funcția Manual Mute (stare indicată pe ecran prin simbolul [M]).

5. Optimizarea Resurselor (Firmware) Codul sursă a fost scris fără a depinde de biblioteci externe masive, utilizând manipularea directă a regiștrilor pentru protocolul I2C și timerele PWM. Rezultatul este un firmware compact, rapid și eficient, adaptat limitărilor de memorie ale microcontrolerului ATmega328P.

Concluzii

Proiectul stației meteo a reprezentat o oportunitate excelentă de a integra concepte fundamentale de electronică și programare embedded într-un dispozitiv util și compact.

În urma realizării acestui sistem, pot fi desprinse următoarele idei principale:

Eficiența Programării Low-Level: Utilizarea regiștrilor și a protocolului I2C implementat manual a demonstrat că se pot obține performanțe ridicate și un consum minim de memorie (Flash/RAM) fără a depinde de biblioteci comerciale complexe.


Interactivitatea și Flexibilitatea: Implementarea meniului în EEPROM a transformat un simplu cititor de senzori într-un dispozitiv adaptiv, capabil să răspundă nevoilor specifice ale utilizatorului prin calibrare și setarea pragurilor de alertă.

Importanța Prelucrării Datelor: Filtrarea semnalelor (prin media mobilă) și utilizarea formulelor climatologice (Magnus-Tetens) au arătat că precizia unei stații meteo nu depinde doar de senzori, ci și de algoritmi de procesare a datelor brute.

Feedback-ul Multimodal: Combinarea alertelor vizuale (RGB) cu cele sonore (PWM) și condiționarea acestora de factori externi (Lumină/Lux) oferă o experiență de utilizare sigură și non-intruzivă.

În concluzie, stația meteo realizată este un sistem robust și versatil, care reușește să centralizeze monitorizarea complexă a mediului ambiant într-o interfață simplă și eficientă, fiind o bază solidă pentru extinderi viitoare (ex: conectivitate IoT sau logarea datelor pe card SD).

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).

Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin.**

[anicolaeescu2602_download.zip](#)

Jurnal

Această secțiune documentează etapele parcurse în dezvoltarea proiectului, evidențiind principalele sarcini și modul în care au fost soluționate problemele tehnice întâmpinate.

Data	Etapă / Activitate	Descriere și Rezultate
20.04.2026	Documentare și Setup	Analiza senzorilor AHT20 și BMP280. Stabilirea schemei de pini pe ATmega328P.
25.04.2026	Driveri I2C Low-Level	Implementarea manuală a protocolului TWI. Testarea comunicației cu LCD-ul (adresa 0x27).
02.05.2026	Achiziție Date Sensori	Citirea datelor brute de la AHT20. Implementarea logică pentru controlul LED-ului RGB.
05.05.2026	Calibrare și Filtrare	Integrarea BMP280. Implementarea filtrului de medie mobilă pentru presiune (10 eșantioane).
08.05.2026	Interfață și Meniu	Implementarea logică pentru pagini multiple pe LCD și meniul de setări praguri.
10.05.2026	Gestiune Memorie	Integrarea logică EEPROM pentru salvarea permanentă a pragurilor setate de utilizator.
11.05.2026	Alerte și Mod Noapte	Implementarea PWM dinamic pentru Buzzer și a logicii de silențios bazată pe LDR (Lux).
12.05.2026	Finalizare și Wiki	Curățarea codului, realizarea Makefile-ului și finalizarea documentației pe Wiki.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

Resurse Hardware (Datasheets)

Documentația oficială utilizată pentru înțelegerea hărții regiștrilor și a protocoalelor de comunicare:

Atmel ATmega328P: [Datasheet complet \(Architecture, Registers, Timers\)](#).

Bosch BMP280: [Digital Pressure Sensor Datasheet](#) (utilizat pentru formulele de compensare și calibrare).

Cursurile si laboratoarele din cadrul cursului de Proiectarea cu Microprocesoare

Resurse Software

Uneltele și mediile de programare care au facilitat dezvoltarea proiectului:

PlatformIO Core: [Documentație oficială](#) pentru managementul proiectului și al build-ului.

AVR Libc Home Page: [Standard C library for AVR](#) (referință pentru `<avr/interrupt.h>` și `<avr/eeprom.h>`).

Formula Magnus-Tetens: [Dew Point Calculation Methodology](#) (calculul punctului de rouă).

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2026/atoader/anicoloesu2602>



Last update: **2026/05/25 08:39**