

Garderobă Inteligentă

Introducere

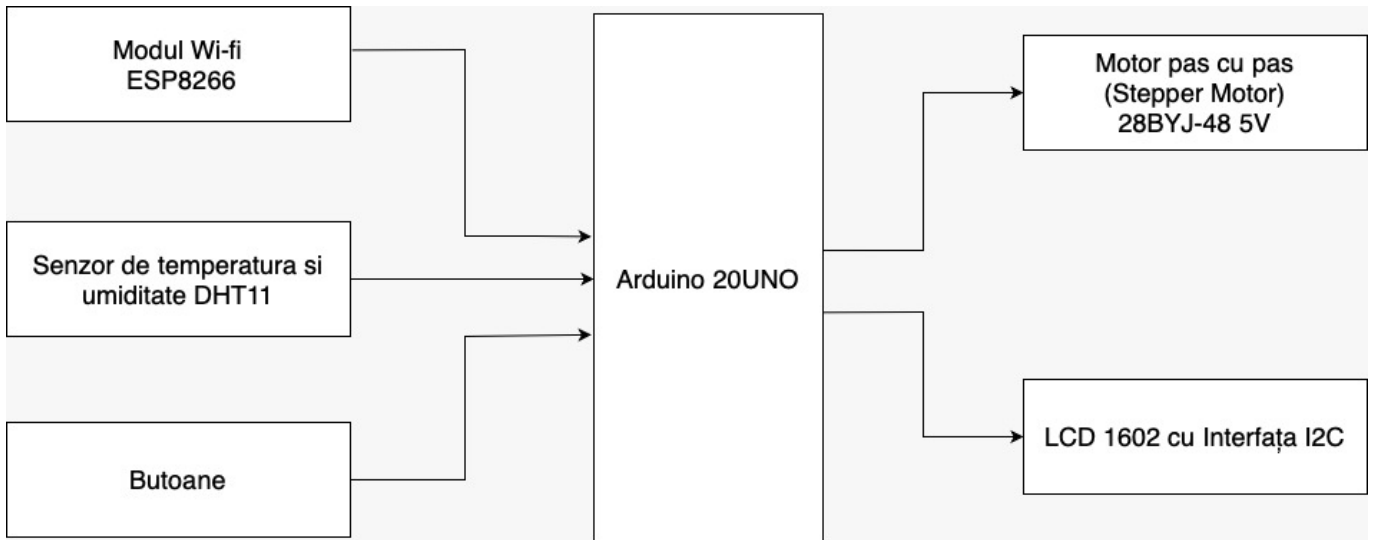
Acest proiect are ca principal obiectiv automatizarea unei garderobe uzuale concentrându-se pe transformarea acesteia într-o experiență vestimentară inteligentă și personalizată.

Acesta vă oferă posibilitatea de a accesa o selecție personalizată de articole vestimentare, adaptate atât la sezonul selectat, cât și la preferințele dumneavoastră, punând la dispoziție garderoba acestuia. Proiectul oferă o abordare simplificată pentru a găsi articolele dorite, facilitându-vă procesul de căutare.

Descriere generală

Garderoba inteligentă este o soluție inovatoare care combină tehnologia cu organizarea eficientă a hainelor și accesoriilor. Aceasta constă într-un sistem automatizat de depozitare și gestionare a articolelor vestimentare. Proiectul dispune de un motor stepper ce permite rotația suportului de haine la 360°. Cadrul este împărțit în 4 secțiuni, fiecare având un unghi de 90°. Motorul este pus în mișcare cu ajutorul celor 4 butoane, ele reprezentând câte un anotimp care corespunde fiecărei secțiuni a suportului.

Schema bloc

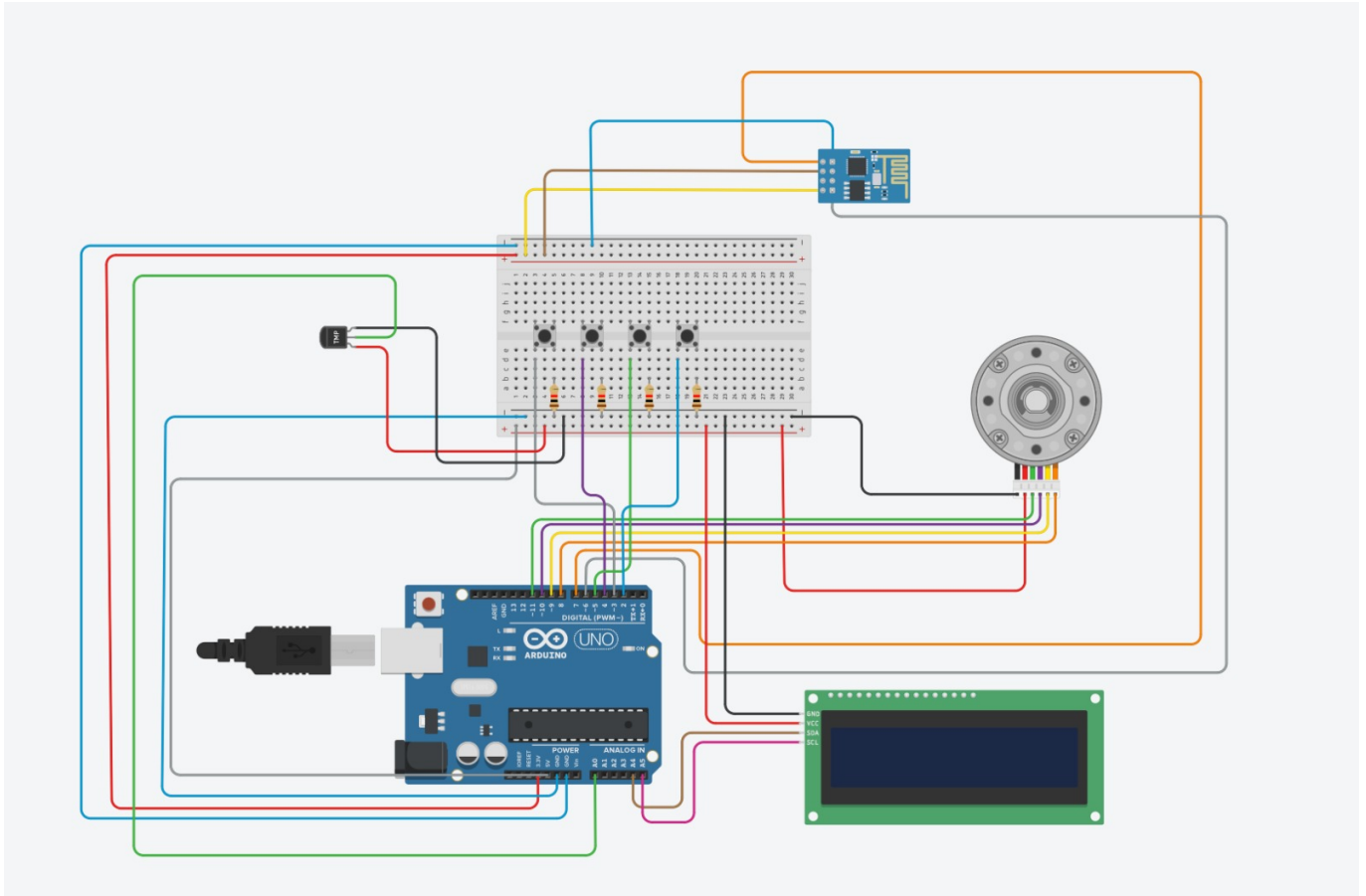


Hardware Design

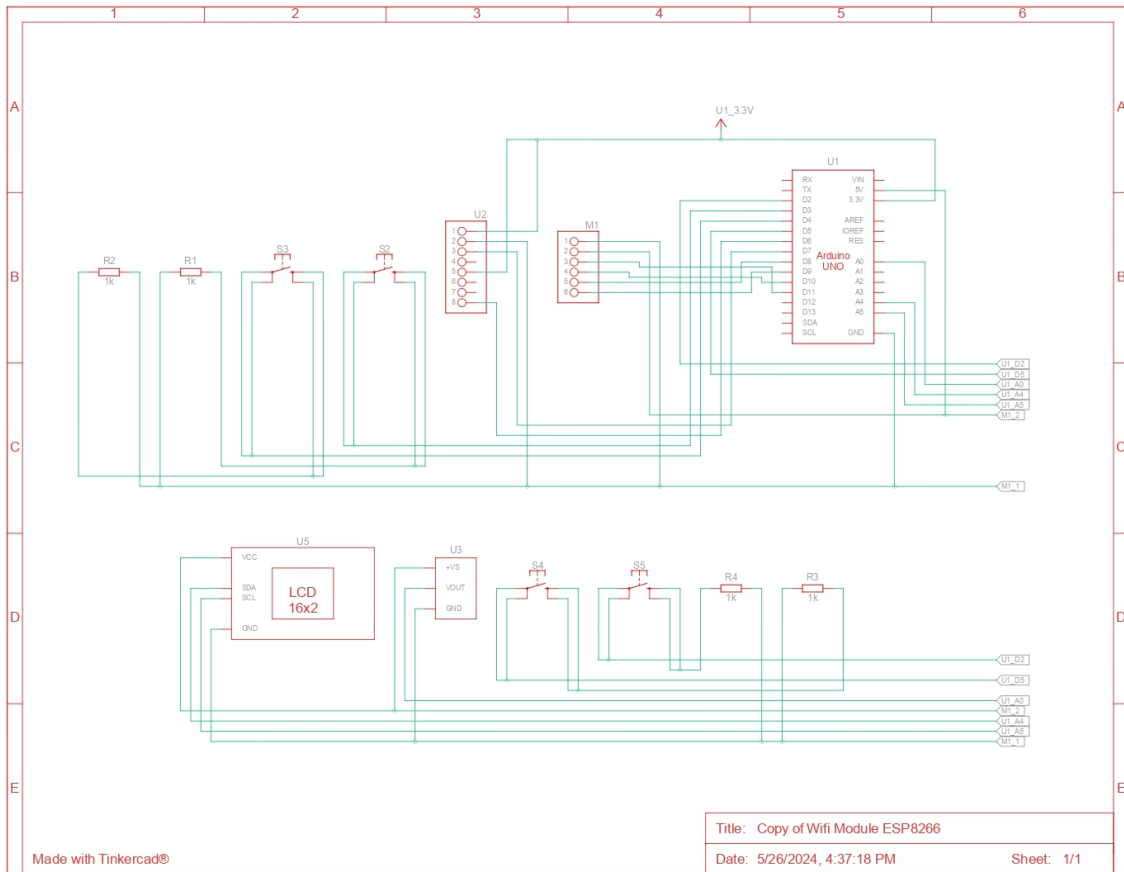
Materiale utilizate și rolul lor

- Placă de dezvoltare compatibilă cu Arduino UNO (Atmega328p și CH340) și cablu de 50 cm.
 - Rol: Permite utilizatorilor să controleze și să monitorizeze dispozitivul
- Motor pas cu pas (Stepper Motor) 28BYJ-48 5V
 - Rol: Rotește suportul în funcție de anotimpul dorit
- Butoane cu capac rotund (roșu, negru, albastru, alb)
 - Rol: Selectează anotimpul dorit din perspectiva utilizatorului
- Rezistențe
 - Rol: Protejarea componentelor electrice, evitând supraîncărcarea lor
- Breadboard HQ 400 puncte
 - Rol: Platformă flexibilă pentru prototipare rapidă
- Modul Wi-fi ESP8266
 - Rol: Conectează plăcuța la internet și preia datele necesare în timp real(temperatura)
- LCD 1602 cu Interfața I2C
 - Rol: Afișează informațiile preluate de pe internet(temperatura)

Cablajul proiectului



Schema electrică

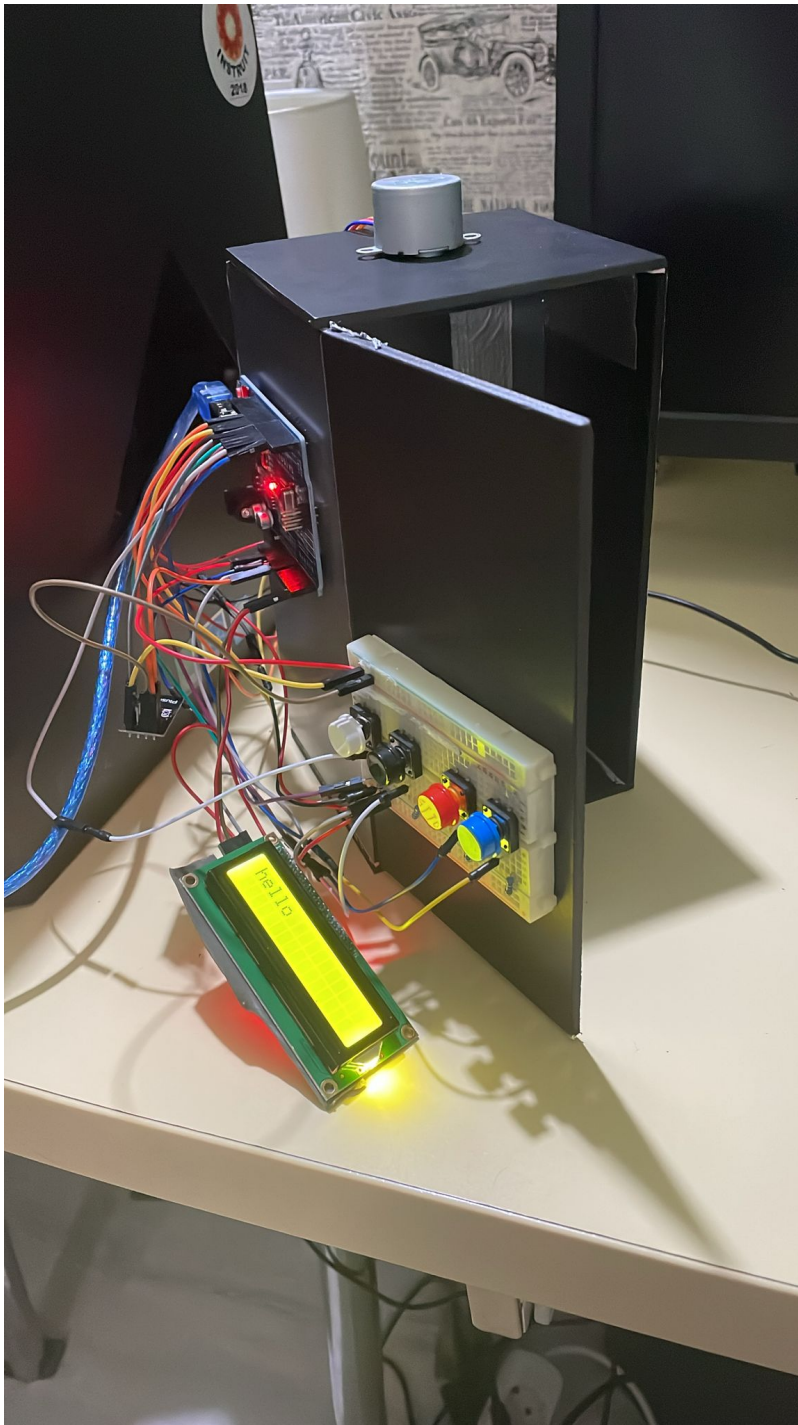


Funcționarea unei componente

În următorul clip este demonstrată funcționarea motorului pas cu pas (Stepper Motor) care se rotește în funcție de ce buton apăsăm, fiecare buton reprezentând un anotimp. În sectorul de cerc din dreptul garderobei se află rezultatul nostru pentru fiecare dintre butoane.

https://drive.google.com/file/d/1uHs1_MRQsfwcs1cf6xWDZbA51dKv51c/view?usp=sharing

Iar în această poză se vede funcționalitatea LCD-ului care momentan afișează doar "hello", dar ar trebui ca după conectarea cu modulul Wi-fi să afișeze temperatura de afară.



Conectarea pinilor

- Arduino Uno
 - Pinul GND : Acesta este folosit pentru a conecta masa la breadboard și la alte componente pentru a completa circuitul.
 - Pinul 5V: Acest pin furnizează 5V către breadboard și alte componente, asigurând alimentarea necesară.
- Butoane
 - Pini Digitali (D2, D3, D4, D5):
 - Conectate la pinii digitali D2, D3, D4 și D5.
- Modul Wi-Fi

- Pinii VCC, CH_PD, RST: Conectati la 3.3V
- Pinul GND: Conectat la GND.
- Pinul RX: Conectat la RX de pe Arduino.
- Pinul TX: Conectat la TX de pe Arduino.
- Motorul Pas cu Pas
 - Pini Digitali (D8, D9, D10, D11): conectati la Stepper Motor
- Display LCD
 - SDA este conectat la A4
 - SCL la A5
 - VCC la 5V
 - GND la GND
- DHT11 - Senzor de temperatură și umiditate
 - Pinul VCC la 5V
 - Pinul 2 la D6
 - Pinul 4 la GND

Concepte folosite din laboratoare

1. I2C : Am folosit acest protocol pentru a comunica cu LCD 16×2 cu interfață I2C
2. GPIO : Butoanele sunt conectate la pinii digitali GPIO ai Arduino-ului pentru a citi starea acestora (HIGH sau LOW).
3. ADC : Senzorul de temperatură și umiditate DHT11 este conectat la pinul analogic A0 de pe Arduino, folosind funcționalitatea ADC pentru a citi datele analogice de la senzor.

Software Design

Biblioteci folosite

Wire.h și LiquidCrystal_I2C.h

- Wire.h este biblioteca standard pentru comunicarea I2C pe Arduino.
- LiquidCrystal_I2C.h permite utilizarea unui ecran LCD 16×2 cu interfață I2C. Oferă o interfață simplă pentru controlul ecranului

Stepper.h

- Biblioteca Stepper este utilizata pentru schimbarea poziției sezoniere în funcție de apăsarea butoanelor.

SoftwareSerial.h

- Biblioteca SoftwareSerial este utilizata pentru comunicarea serială cu modulul ESP8266.

ArduinoJson.h

- Biblioteca ArduinoJson este utilizată pentru parsarea și manipularea datelor JSON. Permite parsarea răspunsurilor JSON primite de la API-ul meteo.

DHT.h

- Biblioteca DHT este specifică pentru senzorii de temperatură și umiditate DHT11. Gestionează comunicarea cu senzorul DHT11 pentru a citi valorile de temperatură și umiditate.

Variabilele si rolul lor

Senzor DHT

- #define DHTPIN A0: Pinul la care este conectat senzorul DHT11.
- #define DHTTYPE DHT11: Tipul senzorului (DHT11).
- DHT dht(DHTPIN, DHTTYPE);: Inițializează senzorul DHT11 pe pinul specificat.

Conexiune Wi-Fi

- const char* ssid = "Nume internet";: SSID-ul rețelei Wi-Fi.
- const char* password = "Parola internet";: Parola rețelei Wi-Fi.
- const char* apiKey = "cdadacc4dcebbd54cd11e8bdbb787111";: Cheia API pentru accesarea serviciului meteo.

LCD

- LiquidCrystal_I2C lcd(0x27, 16, 2);: Inițializează un LCD cu interfață I2C pe adresa 0x27, cu dimensiuni 16x2.

Motor Stepper

- int motorSpeed = 15;: Viteza motorului stepper (RPM).
- int pin1 = 8;, int pin2 = 9;, int pin3 = 10;, int pin4 = 11;: Pini de control ai motorului stepper.
- Stepper myStepper(2048, pin1, pin3, pin2, pin4);: Inițializează motorul stepper cu 2048 pași pe rotație și pini de control specificați.

Butoane

- int buttonAutumn = 4;, int buttonSummer = 5;, int buttonSpring = 2;, int buttonWinter = 3;: Pini pentru butoanele de control al sezonului.
- int readingAutumn = 1;, int readingSummer = 1;, int readingSpring = 1;, int readingWinter = 1;: Valorile citite de la butoane.

Alte Variabile

- int currentSeason = 0;: Sezonul curent (0 - Primăvară, 1 - Vară, 2 - Toamnă, 3 - Iarnă).
- int opus = 1024;, int dreapta = 512;, int stanga = -512;: Valori pentru pașii necesari pentru motorul stepper în funcție de direcție.
- SoftwareSerial espSerial(0, 1);: Inițializează un port serial software pe pini 0 și 1 pentru comunicarea cu ESP8266.

Functii si rolul lor

setup()

- Inițializează comunicarea serială pentru debugging (`Serial.begin(115200);`).
- Inițializează comunicarea serială software pentru ESP8266 (`espSerial.begin(115200);`).
- Inițializează LCD-ul (`lcd.init(); lcd.backlight();`).
- Configurează pinii pentru butoane (`pinMode(buttonAutumn, INPUT_PULLUP);`).
- Setează viteza motorului stepper (`myStepper.setSpeed(motorSpeed);`).
- Inițializează senzorul DHT (`dht.begin();`).
- Conectează la rețeaua Wi-Fi apelând `connectToWiFi()`.

loop()

- Citește valorile de temperatură și umiditate de la senzorul DHT.
- Afișează temperatura și umiditatea pe LCD.
- Citește starea butoanelor.
- Controlează motorul stepper în funcție de butonul apăsat și sezonul curent.

connectToWiFi()

- Trimite comenzi AT către ESP8266 pentru a seta modul de operare și a se conecta la rețeaua Wi-Fi.
- Verifică răspunsul pentru a asigura conexiunea Wi-Fi reușită.

fetchAndDisplayTemperature()

- Trimite comenzi AT către ESP8266 pentru a inițializa o conexiune TCP către API-ul meteo.
- Trimite o cerere GET pentru a obține datele meteo.
- Parsează răspunsul JSON pentru a extrage temperatura.
- Afișează temperatura pe LCD.

Rezultate Obținute

Proiectul este finalizat cu succes.

Codul pentru garderoba inteligenta se afla la acest link :

<https://github.com/10bianca/Garderoba-Inteligenta/blob/main/test.ino>

Demonstratia pentru functionarea intregului proiect este in acest videoclip:

Concluzii

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**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

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

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/bianca.tanase1007>



Last update: **2024/05/27 06:16**