

FloraCare

FOST Cooler pad de laptop

PS: Nu stiu exact ce s-a stricat la cooler, dar n-am reusit in ultimele zile sa gasesc un inlocuitor asa ca am divigat cu un proiect lucrat precedent care consider ca indeplineste in continuare conditiile.

Introducere

Proiectul acesta are ca scop automatizarea ingrijirii plantelor de interior, astfel incat utilizatorul sa nu mai fie nevoit sa verifice constant starea lor. Ideea a pornit de la nevoia de a avea grija de plante chiar si atunci cand nu esti acasa sau pur si simplu nu ai timp.

Sistemul e construit in jurul unei placi Arduino Uno si foloseste senzori care monitorizeaza umiditatea solului, temperatura si nivelul apei din rezervor. Cand solul se usuca, pompa porneste automat si uda planta. Daca apa din rezervor scade prea mult, un LED se aprinde ca semnal de avertizare. Am inclus si un difuzor care anunta finalizarea udarii, pentru un plus de feedback.

Pe langa toate acestea, exista si un ecran LCD care afiseaza in timp real valorile masurate. Optional, sistemul poate reda sunete despre care se spune ca ajuta plantele sa creasca, iar in zilele fara soare, o sursa de lumina artificiala asigura lumina necesara.

Ideea e simpla: sa ai plante sanatoase fara batai de cap, intr-un mod eficient si accesibil.

LOREM IPSUM

Descriere generală

Pentru un control mai precis asupra comportamentului sistemului, actuatorii vor functiona in intervale bine definite. Udarea plantei va avea loc doar dimineata si seara, pentru a respecta ciclul natural al solului si a evita udarea in exces. Iluminarea artificiala suplimentara va fi activata exclusiv pe durata zilei, permitand plantei sa aiba si perioade de intuneric necesare dezvoltarii. Stimulul auditiv va fi redat zilnic, dar doar in anumite intervale orare predefinite, astfel incat sa nu interfereze cu alte functii sau perioade de repaus.

Pentru flexibilitate si usurinta in demonstratie, toate functiile pot fi declansate si manual, la comanda, indiferent de momentul din zi.



Hardware Design

Listă de componente:

• Arduino Uno • Water Pump • Light Sensor • Speaker • Clock Module • Breadboard • Cables • LEDs

Mai jos se regaseste desgnul orientativ in Tinkercad:



Mai jos se ragaseste o poza hardware a proiectului:



De asemenea si un video demonstrativ:

<https://youtu.be/8lZL-eniPDA>

Software Design

Design-ul software va fi realizat in Arduino IDE.

Am utilizat SoftwareSerial si DFRobotDFPlayerMini drept biblioteci.

```
#include "Arduino.h"
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"
#include "LiquidCrystal_I2C.h"

// LCD pe adresa 0x27, cu 16 coloane si 2 randuri

LiquidCrystal_I2C lcd(0x27, 16, 2);

SoftwareSerial mySoftwareSerial(10, 11); // RX, TX
DFRobotDFPlayerMini myDFPlayer;

void printDetail(uint8_t type, int value);

int play_in_progress=0; // if = 1 then player is working

const int buttonPin = 2;
const int buttonPin2 = 7;
const int relayPin = 2;
int buttonState = 0;

// Intervalul de intrare
const int inputMin = 1;
const int inputMax = 1021;

// Intervalul de ieşire
const int outputMin = 1;
const int outputMax = 30;

void setup()
{
  mySoftwareSerial.begin(9600);
  Serial.begin(115200);
  pinMode(buttonPin, INPUT);
  pinMode(buttonPin2, INPUT);
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH);
}
```

```
    lcd.begin();
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sistem activ");

    myDFPlayer.begin(mySoftwareSerial);
}

void music_switch() {
    if(play_in_progress == 0) {
        myDFPlayer.play(1); //Play the first mp3
        play_in_progress = 1;
    }
    else {
        myDFPlayer.pause(); //stops the music
        play_in_progress = 0; // no longer playing
    }
}

void manual_watering() {

    digitalWrite(relayPin, LOW);
    while (buttonState == HIGH) {
        buttonState = digitalRead(buttonPin2);
    }

    digitalWrite(relayPin, HIGH);

    lcd.begin();
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sistem activ");
}

void loop()
{
    int soundVolume = ((analogRead(A1) - inputMin) * (outputMax - outputMin + 1) / (inputMax - inputMin + 1)) + outputMin;
    myDFPlayer.volume(soundVolume);

    int waterSensor = analogRead(A2);
    int lightSensor = analogRead(A0);

    Serial.print(F("Valoare senzor de lumina: "));
    Serial.println(lightSensor);
    Serial.print(F("Valoare senzor de umiditate: "));
    Serial.println(waterSensor);
}
```

```
if (waterSensor < 100 && lightSensor < 50) {  
    digitalWrite(relayPin, LOW);  
  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Udare automata");  
  
    delay(2000);  
    digitalWrite(relayPin, HIGH);  
}  
  
buttonState = digitalRead(buttonPin);  
if (buttonState == HIGH) {  
    music_switch();  
}  
delay(100);  
  
buttonState = digitalRead(buttonPin2);  
if (buttonState == HIGH) {  
    manual_watering();  
}  
}
```

setup() – initializeaza comunicarea seriala, seteaza pinii si porneste modulul audio DFPlayer.

music_switch() – comuta intre redarea si oprirea muzicii in functie de starea curenta.

manual_watering() – activeaza pompa atata timp cat butonul de udare manuala este apasat.

loop() – ruleaza continuu, citeste senzorii, ajusteaza volumul, face udarea automata si gestioneaza actiunile butoanelor.

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Irigarea s-a realizat automat in functie de umiditatea solului, un senzor analogic declansand pornirea unei pompe de apa printr-un releu, la sesizarea lipsei de lumina data de la fotorezistenta.

Dupa udare, utilizatorul a fost notificat sonor printr-o melodie redată de un buzzer (+ vizual pe LCD).

Sistemul a functionat stabil, fara blocaje, cu afisari corecte si control fiabil al releului, optimizat printr-o logica eficienta de initializare.

Concluzii

Proiectul propus ofera o solutie practica si eficienta pentru monitorizarea si udarea automata a

plantelor, fara ca utilizatorul sa intervina zilnic. Prin folosirea senzorilor si a componentelor de afisare si control, sistemul poate gestiona parametri importanti precum umiditatea solului, temperatura si nivelul apei.

Proiectul este usor de refacut, adaptabil si poate fi extins. Desi a fost gandit pentru plante de apartament, structura modulara permite folosirea lui si in aplicatii mai complexe, inclusiv in sisteme IoT.

Download

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

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele 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

Puteţi avea şi o secţiune 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** şi **Resurse Hardware**.

[Export to PDF](#)

- Pentru modulul MP3: <https://www.instructables.com/Tutorial-of-MP3-TF-16P/> Cat si datasheetul: <https://www.electronicoscaldas.com/datasheet/DFR0299-DFPlayer-Mini-Manual.pdf?srsId=AfmBOopRyqEZILLECGWZad9qkzdOQHsyO3ZfRmh8OgPTLugA3A6b7Q1f>

- Releu datasheet: <https://www.electronicoscaldas.com/datasheet/DFR0299-DFPlayer-Mini-Manual.pdf?srsId=AfmBOopRyqEZILLECGWZad9qkzdOQHsyO3ZfRmh8OgPTLugA3A6b7Q1f>

- ALPHA TEST: displayul LCD: https://www.optimusdigital.ro/ro/optoelectronice-lcd-uri/62-lcd-1602-cu-interfata-i2c-si-backlight-galben-verde.html?search_query=lcd+&results=215

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2025/ccristi/alex.moldoveanu0510>

Last update: **2025/05/30 07:10**

