

Smart Garden

Mihnea-Andrei Balan 332CB

Introducere

Design-ul isi propune sa faciliteze cresterea indoor a plantelor. Acest lucru are la baza mentinerea umiditatii solului in parametri optimi si folosirea luminii artificiale in zilele innorate. In plus, o alta aplicatie (optionala) a sistemului este redarea sunetelor care se considera ca ajuta la cresterea plantelor.

Prezentarea pe scurt a proiectului vostru:

- ce face
- care este scopul lui
- care a fost ideea de la care ați pornit
- de ce credeți că este util pentru alții și pentru voi

Descriere generală

Actuatorii vor avea diferite dependente. De exemplu, planta va fi udata doar seara si dimineata. Iluminarea artificiala suplimentara va functiona doar pe timpul zilei, lasand o perioada de intuneric plantei. Stimulul auditiv va functiona zilnic doar la anumite ore. Pentru usurinta demonstratiei, sistemul va putea fi actionat si la comanda.



O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.

Exemplu de schemă bloc: <http://www.robs-projects.com/mp3proj/newplayer.html>

Hardware Design

Listă de piese:

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

Design-ul realizat in tinkercad este doar orientativ, neavand toate componentele necesare disponibile.



Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)
- diagrame de semnal
- rezultatele simulării

Software Design

Design-ul software va fi realizat in Arduino IDE.

Am utilizat SoftwareSerial si DFRobotDFPlayerMini.

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

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 = 4;
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);

  Serial.println(F("Bun venit!"));

  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);
}

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);
  delay(2000);
  digitalWrite(relayPin, HIGH);
}

buttonState = digitalRead(buttonPin);
if (buttonState == HIGH) {
  music_switch();
}
delay(100);

buttonState = digitalRead(buttonPin2);
if (buttonState == HIGH) {
  manual_watering();
}
}
```

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

Rezultate Obținute

https://youtu.be/_Y8xmQvLD8c

<https://youtube.com/shorts/vdCunu3Rg3Y?feature=share>

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

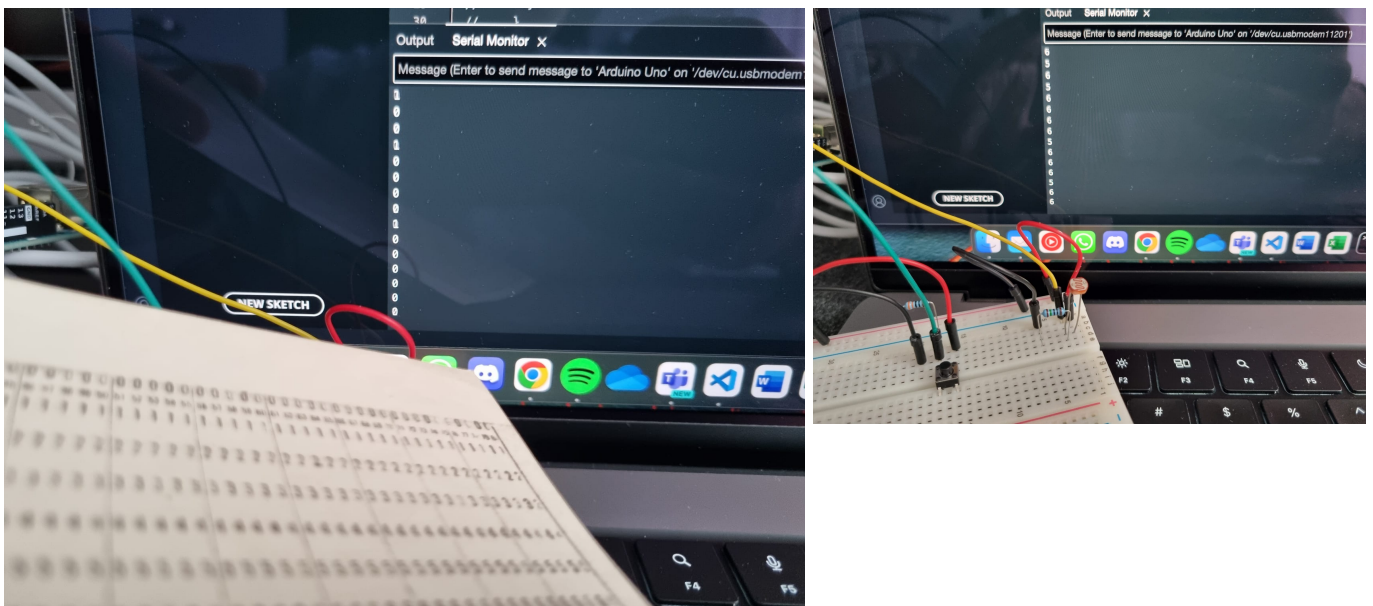
Download

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

17.05.2024 - imi mai lipsesc senzorul de umiditate si difuzorul. In poze se poate vedea ca valoarea citita depinde de cantitatea de lumina care ajunge la fotorezistor.

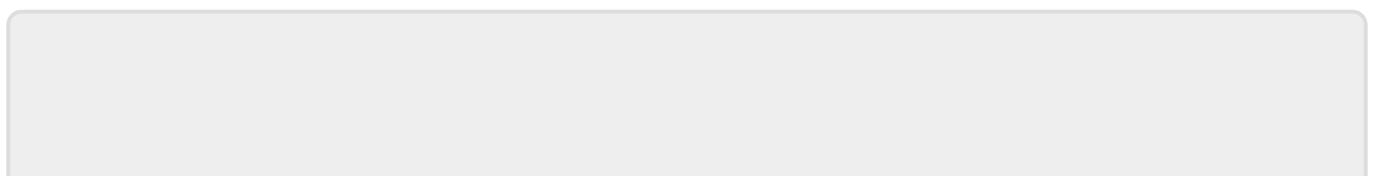


Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

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

[Export to PDF](#)



Last update: 2024/05/24
23:51

pm:prj2024:avaduva:mihnea_andrei.balan http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/mihnea_andrei.balan

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/mihnea_andrei.balan



Last update: **2024/05/24 23:51**