

# Sistem de irigare - Zanficu Madalina 333CA

## Introducere

Proiectul asigura plantei tale un nivel propice de umiditate.

Acest sistem de irigare minimalist vine în sprijinul plantelor de interior uitate de proprietarii plecați în vacanță. Planta se va bucura de un mediu propice dezvoltării iar tu vei sta fără griji fiind notificat constant de starea plantei tale.

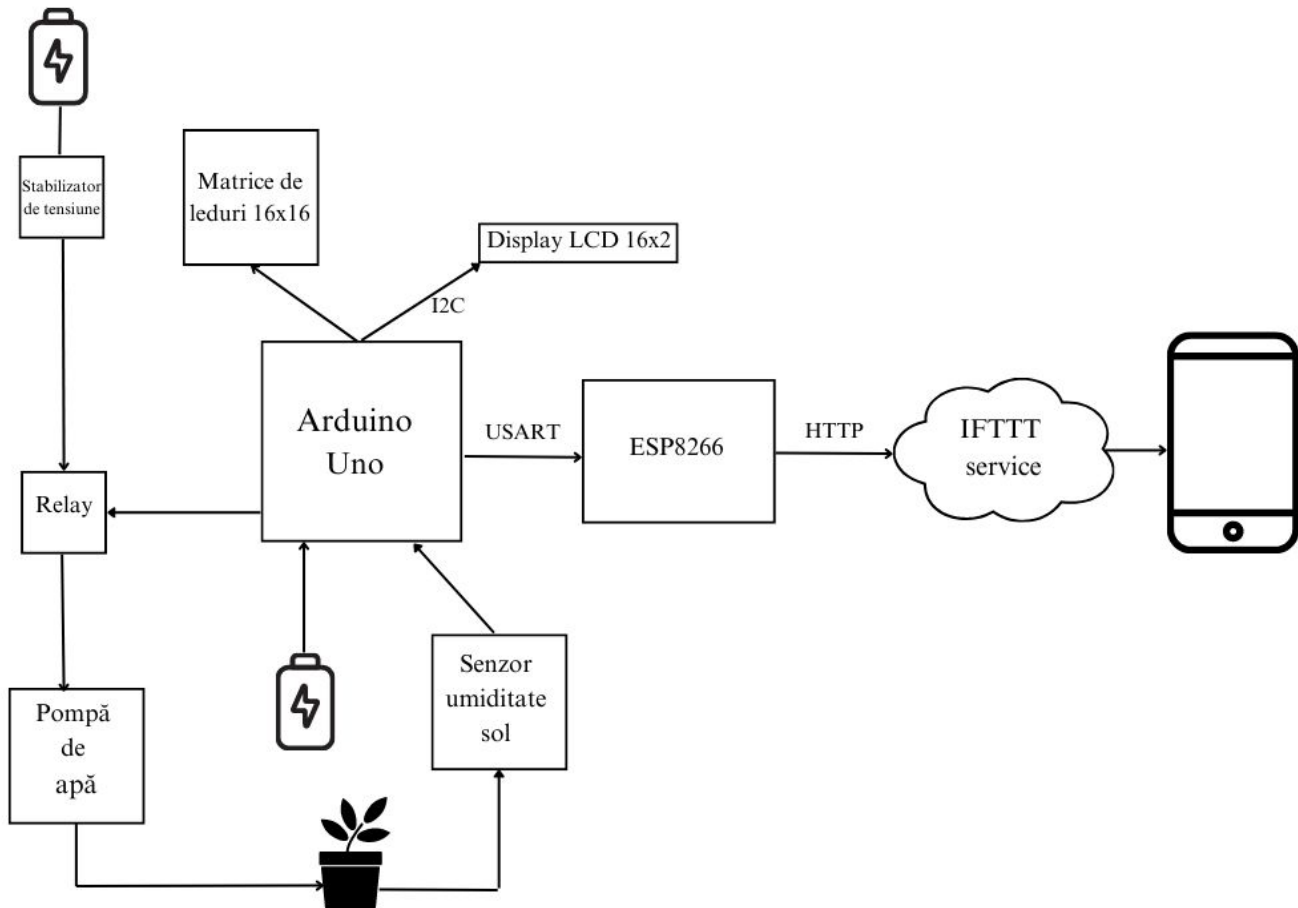
În caz ca esti acasa, vei putea vedea nivelul de umiditate pe un ecran LCD și te vei putea bucura de o animatie pe o matrice led în timp ce planta este udata cu ajutorul unei pompe.

## Descriere generală

În mod normal plantele trebuie udate în funcție categoric: iubitoare de apa sau rezistente la seceta. Pentru a simplifica procesul de măsurare a nivelului de umiditate, **Arduino** va lua măsurători de la **senzorul de umiditate** într-o perioada standard de 1 ora. Daca nivelul de umiditate este sub o valoare de threshold, Arduino va comanda **pompa de apa** pentru a uda planta.

Nivelul de umiditate este afișat pe **ecranul LCD** folosind interfata seriala I2C. Matricea de led-uri va afisa o animatie. De fiecare data cand planta este udata, cu ajutorul **modului ESP32** si a protocoalelor USART și HTTP, se vor trimite notificari pe telefonul proprietarului.

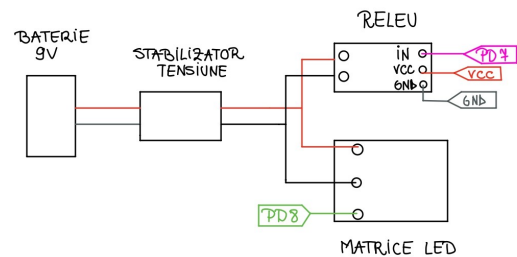
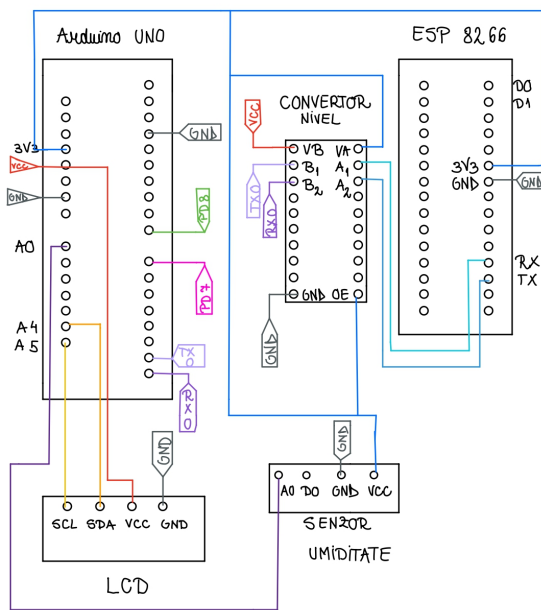
Pentru alimentarea pacutei **Arduino** voi folosi o baterie de 9V si un stabilizator de tensiune pentru a regla tensiunea la 5V.



## Hardware Design

Componente hardware:

- Arduino Uno
- NodeMcu IoT Wireless cu Wifi ESP8266
- Ecran LCD 2 x 16
- Matrice LED 16 x 16
- Pompă de apă
- Baterie 9 V
- Solid State Relay
- Stabilizator de tensiune
- Senzor de umiditate
- Modul convertor nivel logic



## Software Design

### Arduino Uno

Biblioteci folosite:

- LiquidCrystal\_I2C.h → LCD display
- FastLED.h → Matrice led-uri

#### 1. Timer

Sistemul de irigare are la bază un timer ce numără până la 3600 de secunde (1h), pentru a putea verifica în mod constant nivelul de umiditate al solului.

Am folosit **Timerul 1** de pe Arduino în modul de funcționare CTC cu un prescaler de 1024. Ținând cont de acești parametri, timer-ul meu va declanșa o întrerupere atunci când controlul TCNT0 va ajunge la valoarea maximă de pe 16 biti, și îmi va contoriza practic mai puțin de 4 secunde. Pentru a putea număra până la 3600 de secunde am nevoie de 860 astfel de numărări. Astfel îmi incrementez variabila timer\_counter, iar când aceasta va ajunge la numărul de overflows necesare, variabila check\_plant este setată pe true pentru a face verificările necesare.

```
ISR(TIMER1_COMPA_vect) {
    timer_counter++;
    if (timer_counter == overflows) {
        check_plant = true;
    }
}
```

```

    timer_counter = 0;
}
}

void timer_setup() {
    // Dezactivează întreruperile la nivel global
    cli();

    // Setează Timer1 pentru modul de funcționare CTC
    TCCR1A = 0;

    // Setează prescaler la 1024
    TCCR1B = (1 << WGM12) | (1 << CS12) | (1 << CS10);

    // Setează valoarea OCR1A la valoarea maxima posibilă pe 16 biti
    OCR1A = 65535;

    // Activare întrerupere TIMER1_COMPA
    TIMSK1 |= (1 << OCIE1A);

    // Activare întreruperi la nivel global
    sei();
}

```

## 2. Logica de actionare a pompei

În funcția principală - loop, toate acțiunile de verificare a plantei sunt declanșate de **variabila check\_plant**. Astfel se citește valoarea de la senzorul de umiditate, și în funcție de aceasta, arduino comandă releul setând valoarea HIGH pe pinul aferent.

```

if (check_plant == true) {

    // Se citește valoarea de pe senzor (valoare între 0 - 1024)
    soil_moisture = analogRead(sensor_read);

    // Se calculează procentul de umiditate în logica inversă
    humidity_level = map(soil_moisture, 1024, 0, 0, 100);

    check_plant = false;
    show_on_lcd(humidity_level);
    command_relay(humidity_level);
    communication_ESP(humidity_level);
}

```

Releul este un intermediar între Arduino și pompa deoarece Arduino nu poate oferi un amperaj suficient de mare pentru a comanda pompa (100-200mA), astfel releul va conduce acest amperaj către pompa direct din baterie.

```

void command_relay(int humidity_level) {
    int threshold = 80;

```

```
if (humidity_level < threshold) {
    digitalWrite(relay_write, HIGH);
    delay(3000);
    digitalWrite(relay_write, LOW);
}
}
```

### 3. Afișarea pe LCD, comunicarea cu ESP8266

Pentru afișarea datelor pe ecranul LCD, folosesc cunoștințe din **laboratorul 6 - I2C** iar pentru comunicarea cu modulul ESP8266 utilizez comunicarea seriala din **laboratorul 1 - USART**.

```
void communication_ESP(int humidity_level) {
    Serial.println(String(humidity_level));
}
```

### 4. Matrice de LED-uri

Pentru realizarea unei animații pe matricea de led-uri, mi-am creat un header `matrix_animation.h` și am folosit biblioteca `FastLED` pentru setarea led-urilor.

Pentru a crea o tranziție placută am folosit 15 matrici stocate în memoria Flash, motivul fiind spațiul insuficient. În programul principal realizez set-up matricei și apelez funcția `matrix_animation`.

## ESP 8266

Biblioteci folosite:

- `ESP8266WiFi.h` → Conectivitate la internet
- `ESP8266HTTPClient.h` → WEB request

Conexiunea la internet este stabilită în funcția de set-up a programului principal.

```
void WiFi_connection() {
    const char *ssid = "";
    const char *password = "";

    WiFi.begin(ssid, password);

    // Metoda blocanta pentru conectare la internet
    while(WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    // Se aprinde LED ul verde la conectare
    digitalWrite(RED, LOW);
    digitalWrite(GREEN, HIGH);

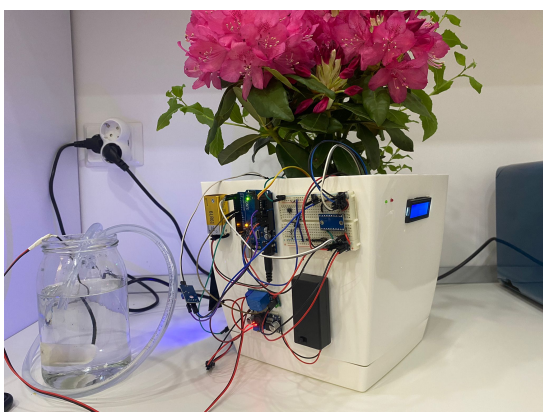
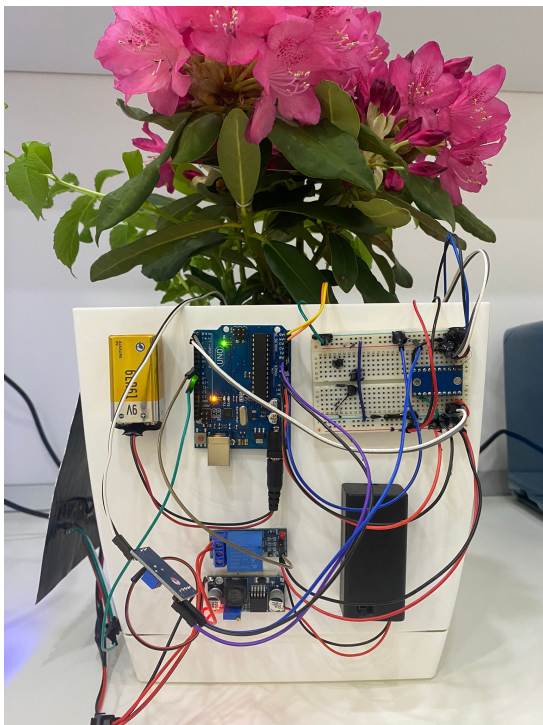
    Serial.println("WiFi Connected!");
}
```

În funcția principală se așteaptă un mesaj pe portul serial de la Arduino Uno.

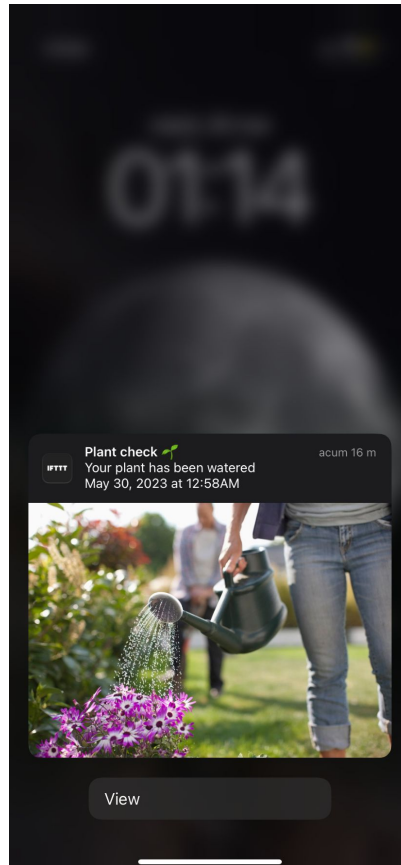
Când modulul ESP8266 a primit mesajul, este apelată funcția `send_notification` în care se construiește un **request WEB de tipul POST** către **serverul de IFTTT**. Serverul trimite notificarea în aplicația din telefon.

## Rezultate Obținute

### Asamblare circuit



### Notificare



## Concluzii

Mi-a placut sa lucrez la proiect si sunt multumita de rezultatul final. Am reusit sa stric o placuta Arduino dar din fericire am avut timp sa gasesc alta in timp util 🗑️

Cu siguranta o sa folosesc proiectul pe viitor, insa voi face cateva modificari pe partea de contorizarea a timpului si voi adauga un modul RTC. Acest proiect este un punct de start, si cred ca as putea duce acest proiect la un alt nivel avand deja cateva idei de improvements pe partea de user-experience.

## Download

Link github: <https://github.com/madalinazanficu/Smart-Pot-PM>

Demo: <https://drive.google.com/drive/folders/1XQAmDZ7C1NcX6RibIn-EicSvMRG--suO?usp=sharing>

## Jurnal

13 aprilie - alegerea proiectului

24 aprilie - achizitionarea pieselor

3 mai - crearea paginii de documentatie

17 mai - milestone hardware

25 mai - milestone software

28 mai - finalizare pagină documentație

## Bibliografie/Resurse

### Resurse hardware

- <https://cleste.ro/modul-cu-senzor-umiditate-sol.html>
- <https://cleste.ro/cautare?controller=search&s=pompa>
- <https://cleste.ro/modul-cu-stabilizator-de-tensiune-.html>
- <https://cleste.ro/modul-releu-1-canal.html>
- <https://cleste.ro/modul-convertoar-nivel-logic-i2c-bidirec-tional-8-bi-i-txs0108e.html>
- [https://www.sigmanortec.ro/Placa-dezvoltare-Lua-Wifi-NodeMcu-CH340G-ESP8266-p134883103?gclid=CjwKCAjwvdajBhBEEiwAeMh1U1LnenacW\\_b7hPtDAUt7c7c5FgNnzULs3LgBiXTFDtxf\\_IYTTW-5AxCzCsQAvD\\_BwE](https://www.sigmanortec.ro/Placa-dezvoltare-Lua-Wifi-NodeMcu-CH340G-ESP8266-p134883103?gclid=CjwKCAjwvdajBhBEEiwAeMh1U1LnenacW_b7hPtDAUt7c7c5FgNnzULs3LgBiXTFDtxf_IYTTW-5AxCzCsQAvD_BwE)
- <https://www.sigmanortec.ro/LCD-1602-p125700685>
- <https://hobbymarket.ro/platforme-dezvoltare/placa-dezvoltare-compatibila-arduino-uno-r3-p-1774.html>
- [https://www.sigmanortec.ro/matrice-led-adresabile-16x16-flexibila-ws2812b-5v?gclid=CjwKCAjwvdajBhBEEiwAeMh1U3XH5q4PV31RcHiIW4TJWt2WFdyQKuScNevIFyOeLzZeCxQfURijyRoCqtsQAvD\\_BwE](https://www.sigmanortec.ro/matrice-led-adresabile-16x16-flexibila-ws2812b-5v?gclid=CjwKCAjwvdajBhBEEiwAeMh1U3XH5q4PV31RcHiIW4TJWt2WFdyQKuScNevIFyOeLzZeCxQfURijyRoCqtsQAvD_BwE)

### Resurse software

- [https://lastminuteengineers.com/soil-moisture-sensor-arduino-tutorial/#google\\_vignette](https://lastminuteengineers.com/soil-moisture-sensor-arduino-tutorial/#google_vignette)
- [https://www.youtube.com/watch?v=iwKE\\_HWU-6M&t=273s](https://www.youtube.com/watch?v=iwKE_HWU-6M&t=273s)
- <https://www.instructables.com/16x16-RGB-LED-Panel-Arduino-Projects/>
- <https://docs.arduino.cc/hardware/uno-rev3>
- <https://circuitdigest.com/microcontroller-projects/getting-started-with-nodemcu-esp12>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/adarmaz/irrigationsystem>



Last update: **2023/05/30 12:27**