

# Sistem de irigatii automat

## Introducere

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

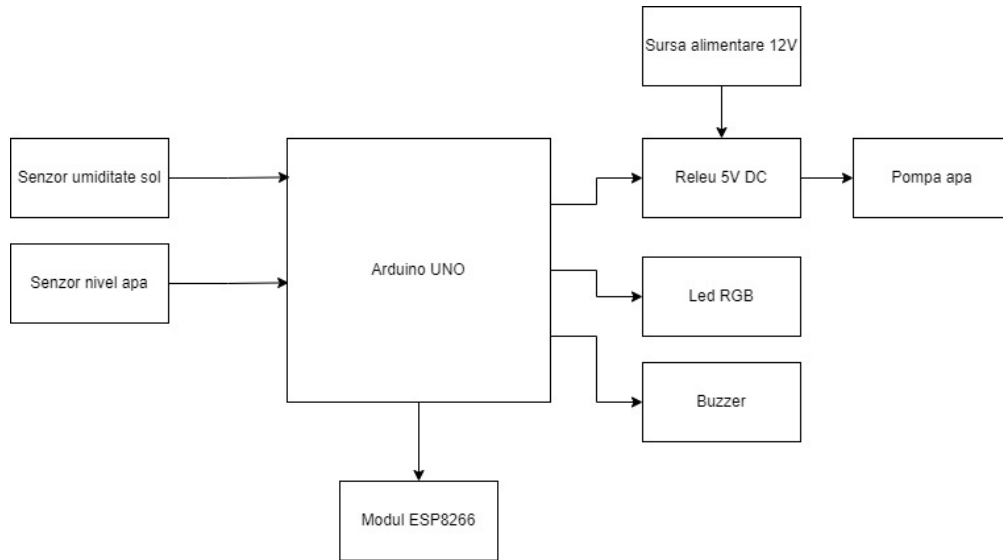
Proiectul consta intr-un sistem automat de irigatii, pentru a simplifica ingrijirea plantelor din casa. Acesta va masura umiditatea solului precum si nivelul de apa dintr-un vas pentru a verifica daca poate fi pornita o pompa pentru irigare. Ideea a pornit de la multitudinea de plante de casa care au fost neglijate. Sistemul va contine un buzzer pentru atentionare in cazul lipsei de apa pentru ca orice persoana sa poata constientiza cand este nevoie de apa. Pentru a urmari evolutia umiditatii solului sistemul va comunica de asemenea cu un server pentru a oferi o reprezentare grafica a acesteia. }

## Descriere generală

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>

Schema bloc:



Mod de functionare: Arduino citeste datele de pe senzorul de umiditate si de pe cel de nivel al apei, transmite catre ESP umiditatea solului, iar apoi in functie de valori este iluminat cu o anumite culoare un led RGB, porneste un buzzer si poate porni o pompa de 12V alimentata de la o priza.

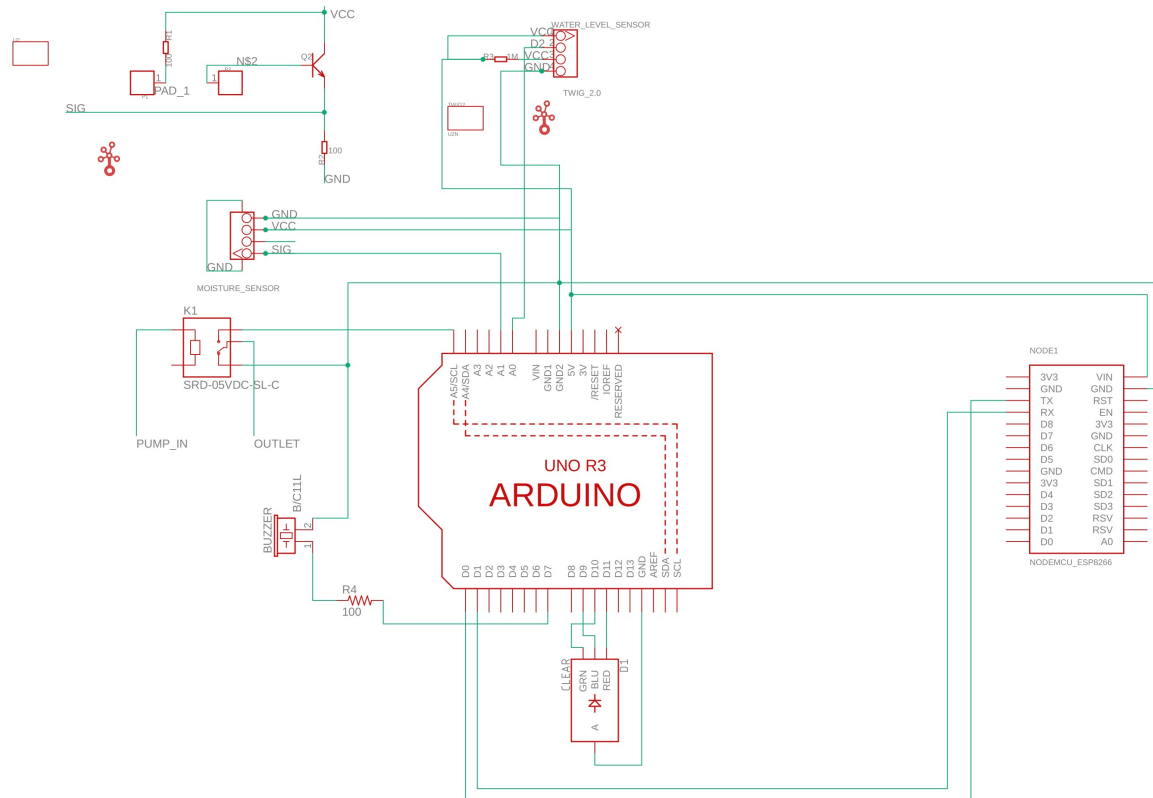
## Hardware Design

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

Componentele utilizate in proiect:

- Arduino UNO
- Pompa apa
- Modul NodeMCU Lua WiFi, V3, ESP8266
- Modul releu 5V DC
- Senzor umiditate sol
- Senzor nivel apa
- Led RGB
- Buzzer
- Sursa alimentare 12V



Pini alesi:

- Analogi: A0, A1, A5
- Digitali: PD0, PD1, PD7, PB1, PB2, PB3

Motivul pentru care au fost alesi in acest fel:

- A0, A1 pentru a avea ambii pini de input apropiati, iar A5 la distanta pentru a fi usor de tinut minte cum trebuie setati.
- PD0, PD1 deoarece sunt pinii de Rx, Tx si sunt folositi pentru comunicarea cu ESP8266
- PB1, PB2, PB3 Pini capabili de PWM utilizati pentru RGB
- PD7 ales aleatoriu dintre pinii ramasi liberi

## Software Design

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

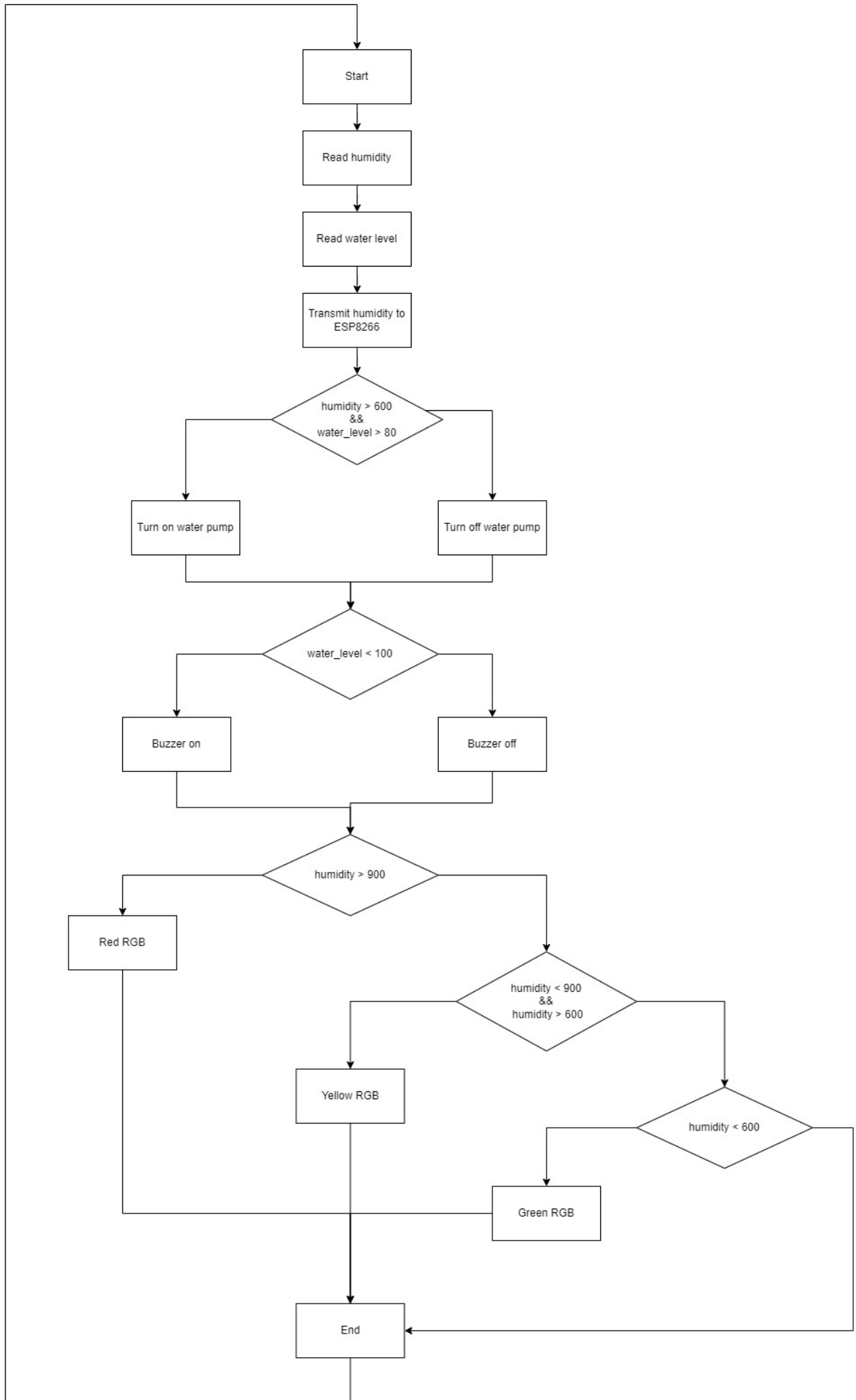
Mediu de dezvoltare:

- Arduino IDE

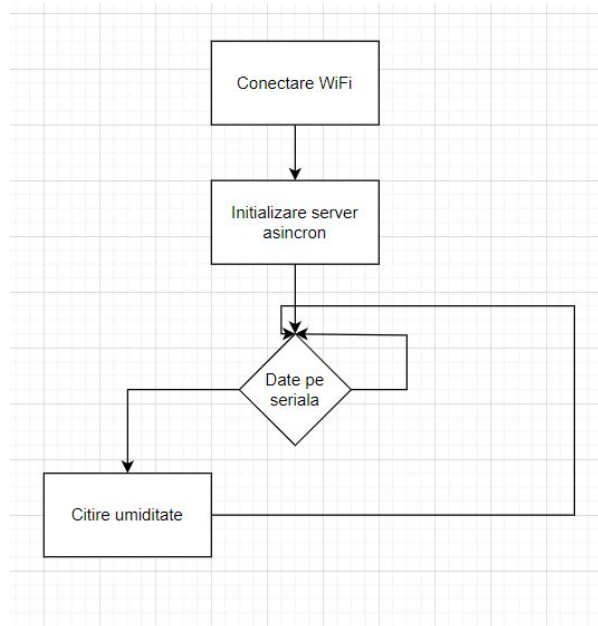
Librarii si surse 3rd-party:

- ESPAsyncTCP
- ESPAsyncWebServer
- <https://code.highcharts.com/highcharts.js> javascript pentru grafic

Software Arduino:



## Software ESP8266:

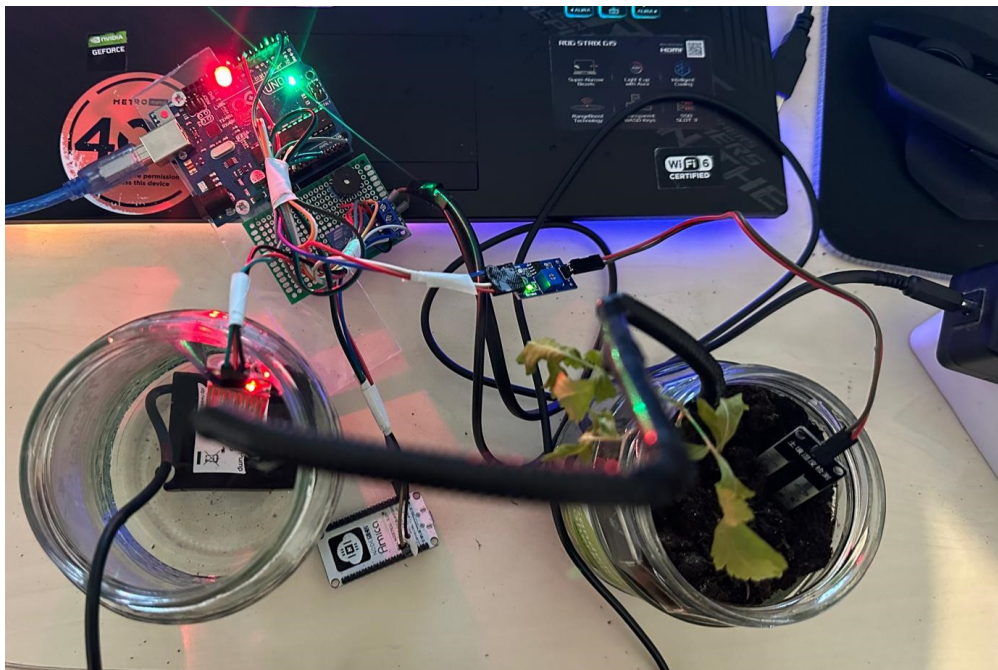


## Rezultate Obținute

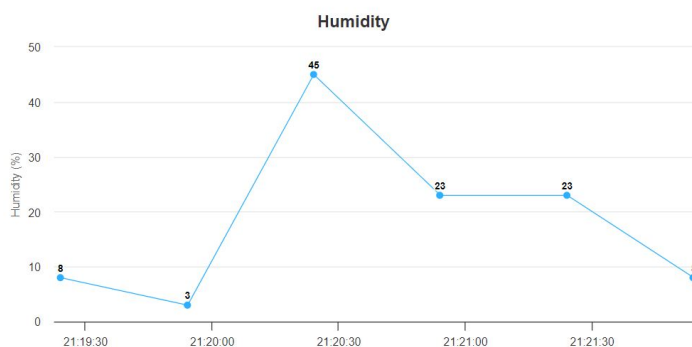
Care au fost rezultatele obținute în urma realizării proiectului vostru.

Am reusit sa obtin un aparat functional care imi va permite sa am o planta fara a uita sa o ud, lucrul foarte util este buzzer-ul care ma va atentiona cand mai trebuie adaugata apa in sistem. De asemenea graficul expus prin intermediul modului ESP imi va da o intelegere mai buna a cat de repede se usuca solul si cat de des ar trebui udat.





Humidity chart




## Concluzii

A fost un proiect interesant. Lucrul complicat a fost de a face debugging in cazul in care ceva nu functiona corespunzator, problema putand fiind data fie de software fie de hardware. Per total experienta a fost una utila deoarece mi-a oferit o perspectiva asupra utilizarii unui microcontroller pentru automatizarea unor sarcini cotidiene.

## 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**.

[sistemirigatiiautomat\\_svemerrobin cristian\\_333cb.zip](#)

## Jurnal

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

- 30 aprilie - finalizare documentatie intiala
- 10 mai - finalizare hardware
- 21 mai - finalizare software
- 29 mai - modificare documentatie pentru finalul proiectului

## Bibliografie/Resurse

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

Resurse software:

- <https://randomnerdtutorials.com/esp8266-weather-forecaster/>

Resurse hardware:

- <https://github.com/mcauser/Grove-Eagle>
- <https://www.snapeda.com/parts/SRD-05VDC-SL-C/Songle%20Relay/view-part/?welcome=home>

[Export to PDF](#)

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

Permanent link:  
[http://ocw.cs.pub.ro/courses/pm/prj2023/abirlica/sistem\\_irigatii\\_automat](http://ocw.cs.pub.ro/courses/pm/prj2023/abirlica/sistem_irigatii_automat) 

Last update: **2023/05/29 19:33**