

RESEARCH ON THE DEVELOPMENT AND IMPLEMENTATION OF AN ALGORITHM FOR THE AUTOMATION OF AN IRRIGATION SYSTEM

ȘERBAN Valentin

Facultatea: Inginerie Industrială și Robotică, Specializarea: Informatică Aplicată în Inginerie Industrială,
Anul de studii: IV, e-mail: valentin.serban1110@gmail.com

Conducător științific: Ș.l.dr.ing. **Ioan-Cristian TARBĂ**

SUMMARY: The paper presents the results of research on the development and implementation of an algorithm for automating an irrigation system in order to properly irrigate crops based on real-time data from their ecosystem and those specific to the type of plant. The system develops the following advantages: correct irrigation of the crop; eliminating the need for the human factor in this process; the possibility to verify data such as temperature, humidity and current status of the remote system; possibility to be controlled remotely. The technological process of the system: the system records the data from a temperature and a humidity sensor in the ground, they are sent to an IoT server where they are compared with those specific to the type of plant selected from the application, are displayed in real time in the mobile application, and depending on the compared values, signals will be given to the relay to open or close the water supply.

KEY WORDS: research, development, implementation, irrigation, automation

1. Introducere

Irigația este un ansamblu de lucrări și operații prin care se aduce și se administrează artificial apă pe un teren cu vegetație.

Sistemul de irigații reprezintă un ansamblu de măsuri tehnice și agrotehnice care fie ajută la sporirea recoltelor agricole, fie la întreținerea obiectivelor de arhitectură peisagistică, fie pentru refacerea vegetației pe terenurile modificate. [1]

Sistemul de irigații clasic este format din o pompă care preia apa din sursă, un furtun pentru transferul apei către plante și stropitori sau picurători pentru răspândirea acesteia în mod egal pe întreaga suprafață acoperită. Punerea în funcțiune a acestui ansamblu este manuală, iar scoaterea din funcțiune la fel, prin acționarea unui robinet acest lucru fiind un consumator de timp. De asemenea cantitatea de apă nu este optimă pentru tipul de plante irigat și nu este mereu exact în momentul în care plantele au nevoie.

2. Stadiul actual

În vederea automatizării unui sistem de irigații s-a făcut un număr considerabil de prototipuri și există multe produse care vizează această piață.

Majoritatea produselor disponibile au ca și funcții setarea orelor și a zilelor în care să pornească sistemul de irigații și timpul de funcționare.

Au fost realizate prototipuri de sistem de automatizare al unui sistem de irigații care permit citirea temperaturii, care pornesc sistemul de irigații în funcție de umiditatea din sol, dar și care pot fi controlate printr-o aplicație mobilă. Însă niciunul dintre acestea nu poate îndeplini toate funcțiile simultan și niciunul nu are disponibilă o bază de date din care se pot selecta tipurile de plante.

3. Dezvoltarea metodologiei privind dezvoltarea și implementarea unui algoritm pentru automatizarea unui sistem de irigații

Algoritmul are ca scop transformarea semnalelor electrice citite de către senzorii de umiditate și temperatură în valori reale, crearea unei baze de date și adăugarea valorilor specifice tipurilor de plante în aceasta, transportul utilizând un MQTT și compararea datelor înregistrate cu cele din baza de date, afisarea valorilor într-o aplicație mobilă, posibilitatea de control al sistemului prin intermediul aplicației, compararea și verificarea tuturor variabilelor înainte de modificarea stării sistemului de irigații și trimiterea semnalului către rețeaua care acționează sistemul.

Sistemul a fost realizat fizic pentru a putea fi testată dezvoltarea și implementarea algoritmului pentru automatizarea sistemului de irigații.

Dezvoltarea aplicației se face pe o placă de dezvoltare de tip ESP-WROOM-32 deoarece aceasta are integrat un modul Wi-Fi care este necesar la conectarea sistemului prin IoT.

Pentru citirea temperaturii folosește un senzor de temperatură de tip DS18B20 rezistent la apă, iar pentru acesta au fost realizate o serie de încercări de realizare al unui program de citire a datelor. Pentru a funcționa acest tip de senzor sunt necesare două librării în Arduino IDE, acestea sunt „OneWire library by Paul Stoffregen” și „DallasTemperature library by Miles Burton”. Problema întâmpinată în timpul realizării programului a fost aceea că anumite funcții ale librărilor nu puteau fi compilate în codul specific plăcii de dezvoltare folosite.

Pentru citirea umidității se folosește un senzor de umiditate în sol împreună cu un traductor de tip LM393, pentru acesta au fost realizate o serie de încercări în vederea realizării programului, dar problema întâmpinată a fost aceea că senzorul înregistrează umiditatea scăzută ca valoare mare, iar pe cea ridicată ca și valoare mică. În acest sens a fost aplicată o formulă pentru inversarea valorilor și transformarea acestora în procente de umiditate relativă.

Ecuția pentru transformarea valorilor citite de senzorul de umiditate în sol cu ajutorul traductorului LM393 în procentul de umiditate relativă:

$$RH = \frac{4095 - x}{4095 - lmin} * 100 \quad [\%] \quad (1)$$

După transformarea valorilor în procente de umiditate relativă au fost atribuite limite de valori pentru determinarea tipului de sol în care se află senzorul de umiditate. Tabelul 1.

Tabelul 1 reprezintă modificările statusului tipului de sol în funcție de valorile înregistrate de către senzorul de umiditate în sol, iar mai apoi convertite în procent de umiditate relativă.

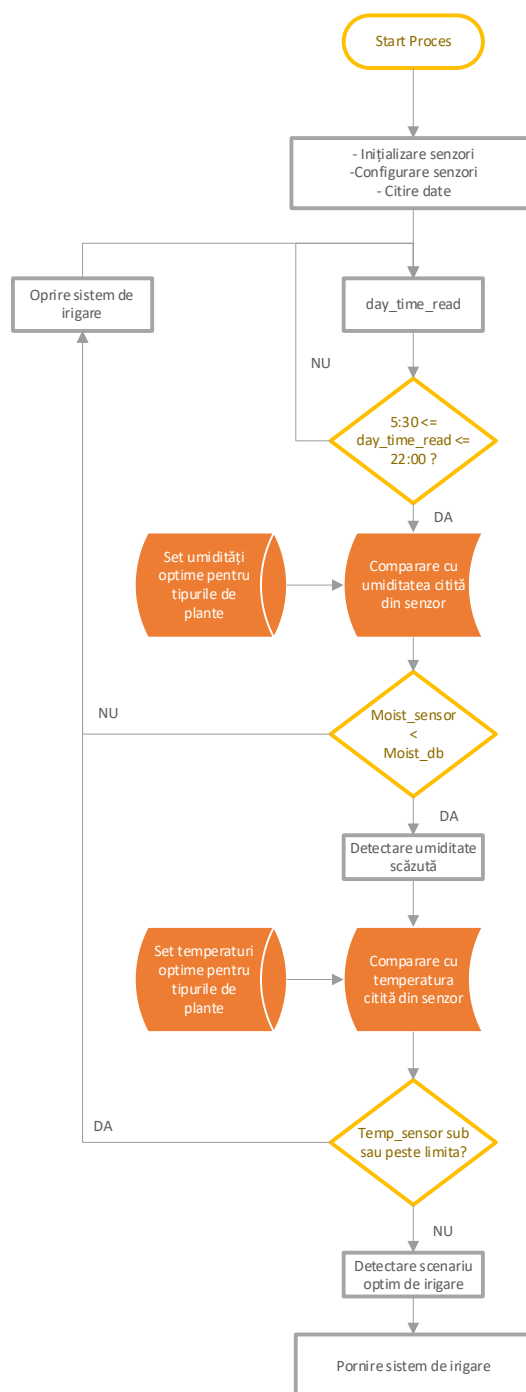


Fig. 1. Schema logică

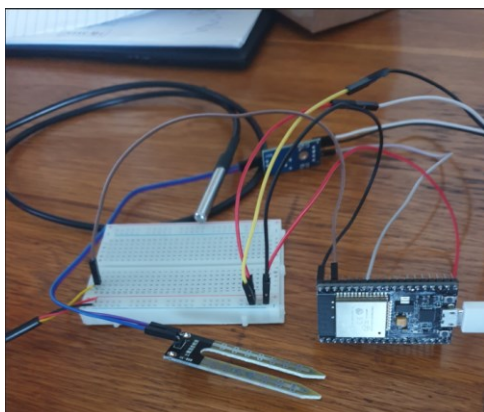


Fig. 2. Prototipul sistemului

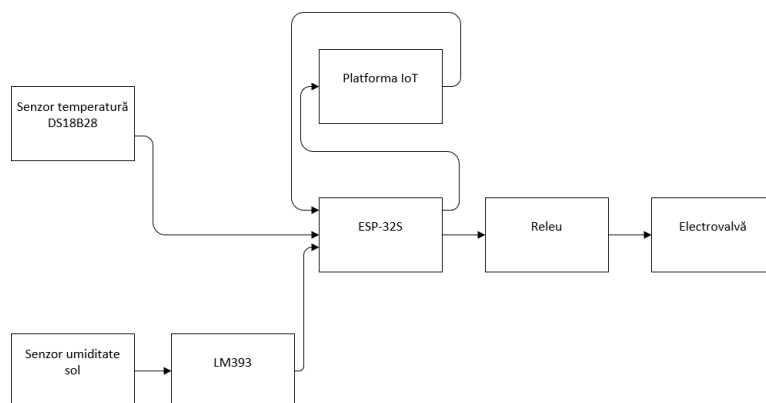


Fig 3. Schema hardware a sistemului

Tabelul 1. Modificările tipului de sol

Nr. Crt.	Tipul solului	Limite [%]	
		Minim	Maxim
1	Uscat	0	30
2	Umed	31	70
3	În apă	71	100

Codul pentru senzorul de umiditate:

```

/*
# 0 - 30 uscat
# 31 - 70 umed
# 71 - 100 in apa
*/
# define ledPin 15
# define sensorPin 32
float trigger = 30;
void setup()
{
    Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, LOW);
}
void loop()
{
    Serial.print("Umiditate:");
    Serial.println((4095*100-analogRead(sensorPin)*100)/(4095-500));
    delay(10000);
}

```

[2]

Se folosește un server de MQTT pentru crearea unei baze de date ce conține parametrii optimi tipurilor de plante, pentru transportarea și compararea parametrilor înregistrați de către senzori și pentru trimiterea semnalului către releul ce acționează electrovalva.

Pentru partea de stocare, transport și comparare a datelor se folosește platforma Blynk care permite crearea bazelor de date, realizării unei interfețe cu utilizatorul, stocarea datelor și transmiterea datelor între aplicație, server și partea hardware prin intermediul Wi-Fi.

Codul sursă pentru integrarea platformei Blynk:

```

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

```

```

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

// This function will be called every time Slider Widget
// in Blynk app writes values to the Virtual Pin 1
BLYNK_WRITE(V1)
{
  int pinValue = param.asInt(); // assigning incoming value from pin V1
to a variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  Serial.print("V1 Slider value is: ");
  Serial.println(pinValue);
}

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
}

void loop()
{
  Blynk.run();
}

```

4. Concluzii

Algoritmul de automatizare al unui sistem de irigații cercetat însumează funcțiile prototipurilor existente, iar în plus față de acestea prin intermediul MQTT pot fi adăugate un număr nelimitat de valori prestabilite pentru fiecare tip de plantă în baza de date, iar sistemul compară și verifică datele automat.

5. Bibliografie

- [1]. Definition of irrigation | Dictionary.com". www.dictionary.com. Retrieved 2021-01-21.
- [2]. *** <https://www.instructables.com/How-to-Make-Automatic-Irrigation-System-Using-Ardu/>
- [3]. ***<https://examples.blynk.cc/?board=ESP32&shield=ESP32%20WiFi&example=GettingStarted>

6. Notății

Următoarele simboluri sunt utilizate în cadrul lucrării:

RH = umiditate relativă

x = valoarea măsurată de senzorul de umiditate

lmin = limita minima măsurată