

Sistem de irigare automat

Cuprins

1	Introducere	2
2	Hardware	2
2.1	Placa Arduino	2
2.2	Senzor de umiditate a solului	2
2.3	Pompa de apa si modulul releu	3
2.4	Senzor de nivel al apei	3
2.5	Buzzer	3
2.6	Senzor DHT11	3
2.7	Senzor de lumina si bec	3
3	Software	3
3.1	Declararea componentelor	3
3.2	Controlul pompei	4
3.3	Functia setup()	4
3.4	Logica principala (loop)	4
3.5	Controlul iluminarii	5
3.6	Semnalizare sonora	5
3.7	Afisare date	5
4	Testare	5
4.1	Testare hardware	5
4.2	Testare software	5
4.3	Testare functionala	6

1 Introducere

Acest proiect are ca scop realizarea unui **sistem de irigare automat**, bazat pe o placa de dezvoltare Arduino, care permite udarea plantelor in mod autonom, fara interventia continua a utilizatorului.

Problema abordata este reprezentata de irigarea necontrolata a plantelor, care poate duce fie la uscarea solului, fie la supraudare. De asemenea, proiectul urmareste prevenirea functionarii pompei in lipsa apei si semnalizarea situatiilor critice.

Solutia propusa consta intr-un sistem embedded care utilizeaza senzori pentru masurarea umiditatii solului, nivelului apei, temperaturii si luminozitatii, si care controleaza automat o pompa de apa, un buzzer si un bec de iluminare.

La nivel high-level, proiectul indeplineste urmatoarele cerinte:

- monitorizarea umiditatii solului;
- controlul automat al pompei de apa;
- protectie la nivel scazut al apei din rezervor;
- semnalizare sonora pentru situatii critice;
- control automat al iluminarii in functie de luminozitate;
- afisarea parametrilor prin interfata seriala.

2 Hardware

Aceasta sectiune descrie componentele hardware utilizate, rolul acestora si modul in care sunt integrate in sistem.

2.1 Placa Arduino

Placa Arduino reprezinta unitatea centrala de control. Aceasta citeste datele de la senzori, executa logica de decizie si controleaza actuatorile (pompa, buzzer, bec).

2.2 Senzor de umiditate a solului

Senzorul de umiditate a solului este conectat la un pin analogic si furnizeaza o valoare proportionala cu nivelul de umiditate din sol. Aceasta valoare este utilizata pentru a determina daca solul este uscat sau umed.

2.3 Pompa de apa si modulul releu

Pompa de apa este controlata prin intermediul unui modul releu. Releul este comandat de Arduino printr-un pin digital si permite comutarea pompei fara a suprasolicita placa de dezvoltare.

2.4 Senzor de nivel al apei

Senzorul de nivel al apei este utilizat pentru a masura cantitatea de apa disponibila in rezervor. Acesta ofera o valoare analogica, iar daca aceasta scade sub un prag stabilit, pompa este dezactivata automat.

2.5 Buzzer

Buzzer-ul este utilizat pentru semnalizarea sonora a nivelului scazut de apa. In acest mod, utilizatorul este avertizat imediat asupra unei situatii critice.

2.6 Senzor DHT11

Senzorul DHT11 masoara temperatura si umiditatea aerului. Aceste valori sunt citite periodic si afisate pentru monitorizare.

2.7 Senzor de lumina si bec

Senzorul de lumina HW-072 functioneaza in mod digital si semnalizeaza daca mediul este luminos sau intunecat. In functie de acest semnal, Arduino controleaza automat un bec sau LED, care se aprinde atunci cand este intuneric.

3 Software

Partea software este implementata in limbajul C/C++ folosind Arduino IDE. Codul este structurat pentru a asigura claritate, modularitate si functionare stabila.

3.1 Declararea componentelor

In prima parte a codului sunt declarate bibliotecile, pinii si pragurile utilizate:

```
1 #include <DHT.h>
2
3 const int soilPin = A0;
4 const int relayPin = 7;
5 int DRY_THRESHOLD = 400;
```

```

6
7 #define DHTPIN 8
8 #define DHTTYPE DHT11
9 DHT dht(DHTPIN, DHTTYPE);

```

Aceste declaratii definesc conexiunile hardware si pragurile de decizie ale sistemului.

3.2 Controlul pompei

Pentru controlul pompei se foloseste o functie dedicata, care tine cont de tipul releului utilizat:

```

1 void setPump(bool on) {
2     digitalWrite(relayPin, on ? HIGH : LOW);
3 }

```

Aceasta functie permite pornirea si oprirea pompei intr-un mod clar si reutilizabil.

3.3 Functia setup()

Functia `setup()` initializeaza pinii, senzorii si interfata seriala:

```

1 void setup() {
2     Serial.begin(9600);
3     pinMode(relayPin, OUTPUT);
4     pinMode(buzzerPin, OUTPUT);
5     pinMode(lightDigitalPin, INPUT_PULLUP);
6     pinMode(ledPin, OUTPUT);
7     dht.begin();
8 }

```

Aceasta etapa este esentiala pentru pregatirea sistemului inainte de rularea efectiva.

3.4 Logica principala (loop)

In bucla principala sunt cititi toti senzorii si se aplica logica de control:

```

1 bool soilDry = (soilValue < DRY_THRESHOLD);
2 bool waterLow = (waterLevelValue < WATER_LOW_THRESHOLD);
3 bool pumpShouldBeOn = soilDry && !waterLow;
4 setPump(pumpShouldBeOn);

```

Pompa porneste doar daca solul este uscat si exista suficienta apa in rezervor.

3.5 Controlul iluminarii

Iluminarea este controlata automat in functie de semnalul digital al senzorului de lumina:

```
1 bool isDark = (lightVal == HIGH);  
2 digitalWrite(ledPin, isDark ? HIGH : LOW);
```

3.6 Semnalizare sonora

Pentru nivel scazut al apei se activeaza buzzer-ul:

```
1 if (waterLow) {  
2     tone(buzzerPin, 1000);  
3 } else {  
4     noTone(buzzerPin);  
5 }
```

3.7 Afisare date

Toate informatiile sunt afisate prin interfata seriala pentru monitorizare si debug.

4 Testare

4.1 Testare hardware

Componentele au fost testate individual:

- verificarea valorilor analogice ale senzorilor;
- comutarea releului;
- activarea buzzer-ului;
- functionarea becului in functie de lumina.

4.2 Testare software

Codul a fost testat prin simularea mai multor scenarii:

- sol uscat cu rezervor plin;
- sol umed;
- nivel scazut de apa;
- functionare continua.

4.3 Testare functionala

Sistemul functioneaza complet automat: pompa porneste si se opreste corect, buzzer-ul avertizeaza situatiile critice, iar iluminarea este controlata in functie de luminozitate.

Concluzie

Proiectul demonstreaza integrarea corecta dintre hardware si software si reprezinta o solutie functionala pentru irigarea automata a plantelor. Sistemul poate fi extins cu usurinta prin adaugarea de conectivitate wireless sau interfete grafice.