

CPGE

Les Classes Préparatoires aux Grandes Ecoles
d'ingénieurs

Centre: Lycée Mohammed V – BENI MELLAL

esith
Constructeur de compétences



Royaume du Maroc

Ministère de l'Enseignement Supérieur,
de la Recherche Scientifique et de l'Innovation

Le système d'arrosage automatique pour un terrain de football

TIPE 2024

Documents annexes

- **Nom et Prénom:** GHANAM Fatima-ezzahra
- **Numéro d'inscription au CNC:** BM030T
- **Encadré par:** M. EZ-ZAHORI Hassan et M. QABBOUCH Amine

Annexe 1:

test_capteur_humidite_sol_capacitif.ino

```
1 //Test de capteur d'humidite de sol capacitif V1.2
2 // Cree par Fatima-Ezzahra GHNAM ,N°CNC :BM030T
3 void setup() {
4   Serial.begin(9600); // Initialisation de la communication série
5 }
6
7 void loop() {
8   int sensorValue = analogRead(A0); // Lecture de la valeur analogique du capteur
9   float percentageHumidity = map(sensorValue, 0, 1023, 0, 100); // Conversion en pourcentage
10
11   Serial.print("Humidité du sol : ");
12   Serial.print(percentageHumidity);
13   Serial.println("%");
14
15   delay(30000); // Attente d'une 30 seconde avant la prochaine lecture
16 }
```

Figure 14 : Programme Arduino pour mesurer l'humidité du sol à l'aide du capteur d'humidité capacitif (V1.2)

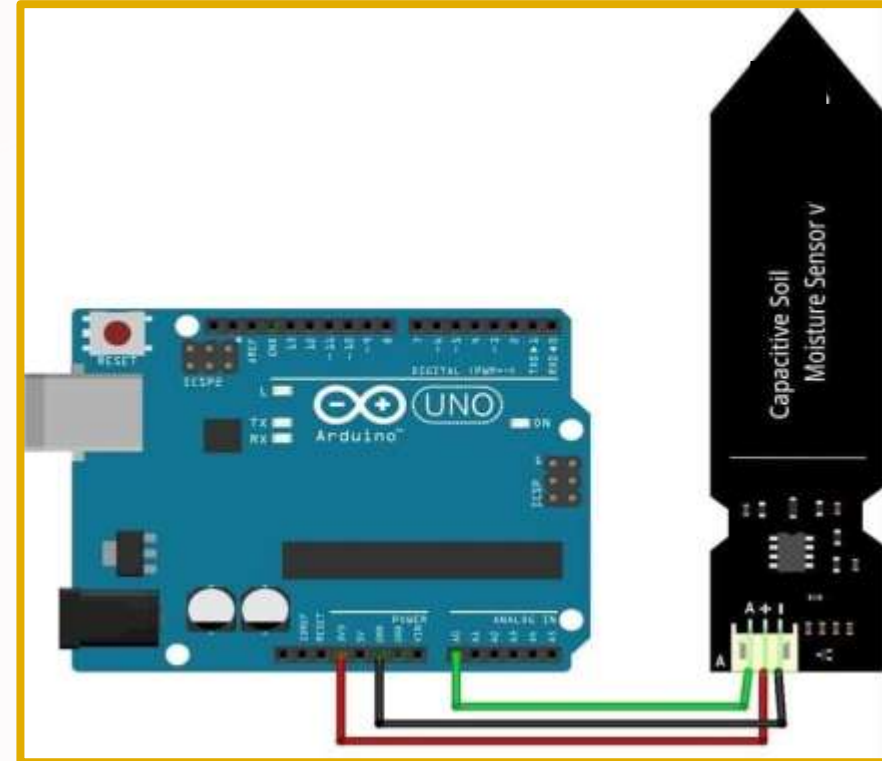


Figure 15 : Schéma de montage du circuit

Annexe 2:

```
Test_capteur_temperature_LM35.ino
1 // Programme pour tester le capteur de température LM35
2 // Cree par Fatima_Ezzahra GHANAM ,N°CNC:BM030T
3 const int sensorPin = A0; // Broche du capteur LM35
4
5 float temperature; // Variable pour la température
6
7 void setup() {
8     Serial.begin(9600); // Initialiser la communication série
9 }
10
11 void loop() {
12     int sensorValue = analogRead(sensorPin); // Lire la valeur analogique
13     float voltage = sensorValue * (5.0 / 1023.0); // Convertir en tension
14     temperature = voltage * 100; // Convertir en température
15
16     Serial.print("Température: "); // Afficher la température
17     Serial.print(temperature);
18     Serial.println(" °C");
19
20     delay(1000); // Attendre 1 seconde
21 }
```



Figure 16 : Programme Arduino pour mesurer la température à l'aide du capteur (LM35)

Annexe 3:

programme_capteur_de_pluie.ino

```
1 // Performance du capteur de pluie
2 // Programme écrit par Fatima-ezzahra GHANAM
3 const int waterSens = A0; // Définir la broche du capteur de pluie
4 const int led = 9; // Définir la broche de la LED (pin 9)
5 int waterVal; // Définir la variable pour la valeur du capteur de pluie
6
7 void setup() {
8     pinMode(led, OUTPUT); // Configurer la LED comme sortie
9     pinMode(waterSens, INPUT); // Configurer le capteur de pluie comme entrée
10    Serial.begin(9600); // Démarrer la communication série à 9600 bauds
11 }
12
13 void loop() {
14     waterVal = analogRead(waterSens); // Lire la valeur du capteur de pluie
15     Serial.print("Niveau d'eau : "); // Afficher le texte "Niveau d'eau :"
16     Serial.println(waterVal); // Afficher la valeur du capteur de pluie sur le moniteur série
17
18     if (waterVal <= 350) { // Si la valeur est inférieure ou égale à 350
19         Serial.println("Il ne pleut pas"); // Afficher "Il ne pleut pas" sur le moniteur série
20         digitalWrite(led, HIGH); // Allumer la LED
21     } else { // Sinon
22         Serial.println("Il pleut"); // Afficher "Il pleut" sur le moniteur série
23         digitalWrite(led, LOW); // Éteindre la LED
24     }
25     delay(30000); // Attendre 30 secondes avant la prochaine lecture
26 }
```

Figure 18 : Programme Arduino pour détecter s'il pleut .

Annexe 4:

Programme python pour le tracage de ULDR en fonction de l'éclairement_GHANAM

```
import matplotlib.pyplot as plt
Eclairement=[0,10,100,200,300,400,500]
Uldr=[4.1,3.18,0.98,0.59,0.44,0.35,0.29]
plt.plot(Eclairement,Uldr)
plt.xlabel("Eclairement en (LUX)")
plt.ylabel('Uldr en (V)')
plt.title("tracage de Uldr en fonction de l'éclairement by GHANAM")
plt.show()
```

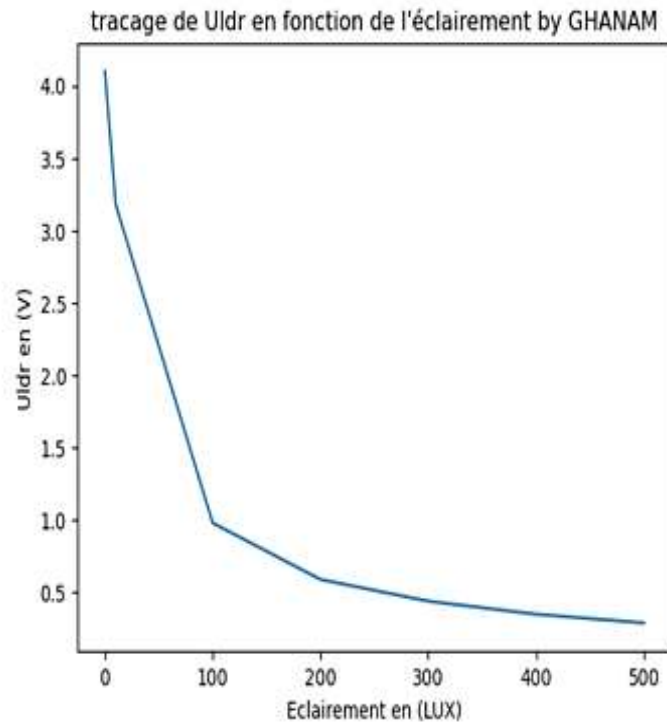


Figure 19 : Programme Python pour le traçage de U_{LDR} en fonction de l'éclairement

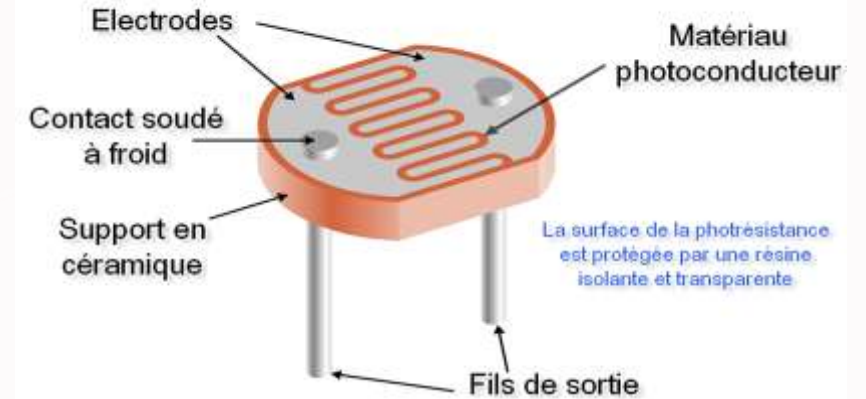


Figure 20 : La composition du capteur de luminosité (LDR)

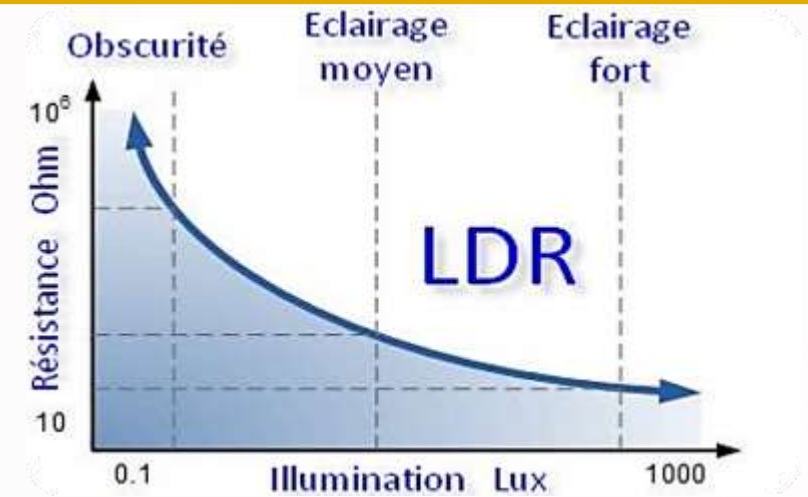


Figure 21 : La variation de résistance en fonction de l'éclairement pour le capteur (LDR)

Annexe 4:

programme_luminosite_capteur_LDR.ino

```
1 //Programme pour mesurer la luminosite avec un capteur LDR
2 //Crée par Fatima-Ezzahra GHANAM N°CNC: BM030T
3
4 const int ldrPin = A0; // Broche analogique à la quelle est connecté le capteur LDR
5
6 void setup() {
7   Serial.begin(9600); //Initialise la communication série
8 }
9
10 void loop() {
11   int lecteurLDR = analogRead(ldrPin); //Lecture de la valeur du capteur
12   Serial.print("Luminosite : ");
13   Serial.println(lecteurLDR);
14
15   delay(1000); //Attente d'une seconde avant de prendre une nouvelle mesure
16 }
```

Figure 22 : Programme Arduino pour mesurer la luminosite à l'aide du capteur (LDR)

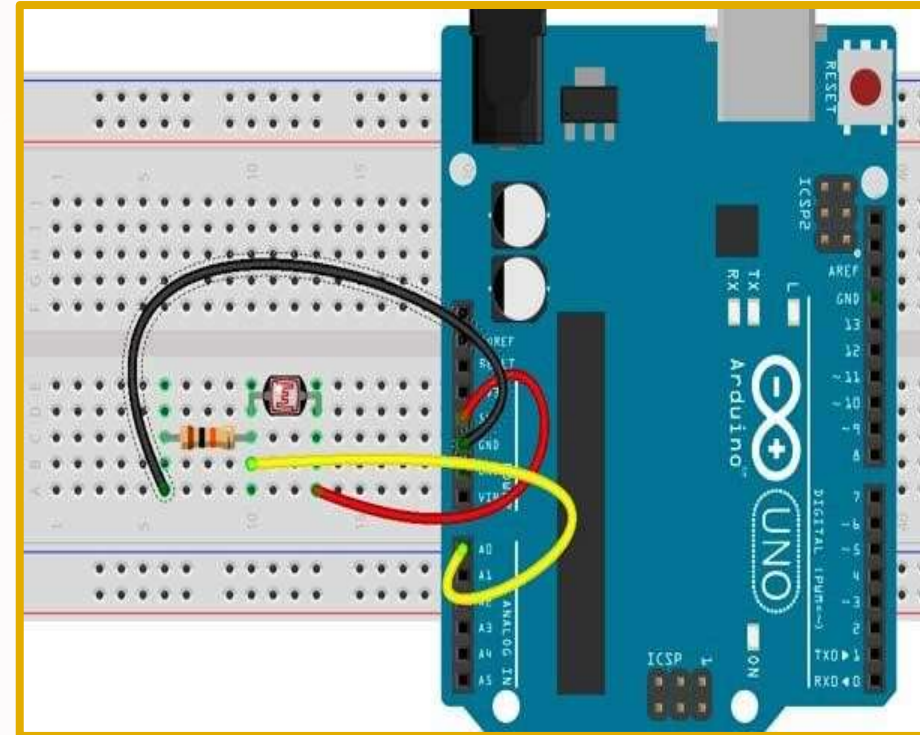


Figure 23 : Schéma de montage du circuit

Annexe 5 :

systeme_arrosage_automatique_terrain_foot_9.ino

```
1 // Système d'arrosage automatique de terrain de foot.
2 // Créé par Fatima_Ezzahra GHANAM N°CNC: BM030T
3 #include <Wire.h>
4 #include <LiquidCrystal_I2C.h>
5
6 // Initialisation de l'afficheur LCD I2C (adresse, colonnes, lignes)
7 LiquidCrystal_I2C lcd(0x27, 16, 2);
8
9 // Broche de données pour le capteur d'humidité capacitif
10 const int humiditePin = A3;
11
12 // Broche de données pour le capteur de température LM35
13 const int temperaturePin = A1;
14
15 // Broche de données pour le capteur de luminosité LDR
16 const int luminositePin = A0;
17
18 // Broche de contrôle pour le relais de l'électrovanne
19 const int relaisPin = 8;
20
21 // Broche de données pour le capteur de pluie
22 const int pluiePin = A2;
23
24 // Broches pour les LED
25 const int ledVertePin = 9;
26 const int ledRougePin = 10;
```

Figure 24 : Programme Arduino du système d'arrosage automatique du terrain de football

Annexe 5 :

```
27
28 void setup() {
29     lcd.begin(16, 2); // Initialisation de l'écran LCD
30     lcd.init();       // Initialisation supplémentaire de l'écran LCD
31     lcd.backlight();  // Allumer le rétroéclairage de l'écran LCD
32
33     // Affichage initial pendant 30 secondes
34     lcd.setCursor(0, 0);
35     lcd.print("TIPE GHANAM 2024");
36     lcd.setCursor(0, 1);
37     lcd.print("SMART ARROSAGE ");
38     delay(30000); // Attendre 30 secondes
39
40     // Effacer l'écran après l'affichage initial
41     lcd.clear();
42
43     // Configuration des broches
44     pinMode(humiditePin, INPUT);
45     pinMode(temperaturePin, INPUT);
46     pinMode(luminositePin, INPUT);
47     pinMode(pluiePin, INPUT);
48     pinMode(relaisPin, OUTPUT);
49     pinMode(ledVertePin, OUTPUT);
50     pinMode(ledRougePin, OUTPUT);
51
52     // Initialisation des états par défaut
53     digitalWrite(relaisPin, LOW);
54     digitalWrite(ledVertePin, LOW);
55     digitalWrite(ledRougePin, HIGH);
56 }
57
```

Figure 25 : Programme Arduino du système d'arrosage automatique du terrain de football (suite).

Annexe 5 :

```
58 void loop() {  
59     // Lecture des valeurs des capteurs  
60     int humidite = map(analogRead(humiditePin), 0, 1023, 0, 100); // Conversion en pourcentage  
61     int temperature = analogRead(temperaturePin) * (5.0 / 1023.0) * 100;  
62     int luminosite = analogRead(luminositePin);  
63     int pluie = analogRead(pluiePin);  
64  
65     // Affichage des valeurs des capteurs sur l'écran LCD  
66     lcd.clear();  
67     lcd.setCursor(0, 0);  
68     lcd.print("H:");  
69     lcd.print(humidite);  
70     lcd.print("% T:");  
71     lcd.print(temperature);  
72     lcd.print("C");  
73     lcd.setCursor(0, 1);  
74     lcd.print("L:");  
75     lcd.print(luminosite);  
76     lcd.print(" P:");  
77     lcd.print(pluie);  
78  
79     delay(30000); // Attendre 30 secondes entre chaque mesure  
80  
81     // Conditions pour l'activation de l'arrosage  
82     bool conditionsArrosage = (luminosite < 500) && (temperature >= 20 && temperature <= 35) && (pluie < 350) && (humidite < 20);
```

Figure 26 : Programme Arduino du système d'arrosage automatique du terrain de football **(suite).**

Annexe 5 :

```
83
84 if (conditionsArrosage) {
85     // Affichage de l'état de l'arrosage en cours
86     lcd.clear();
87     lcd.setCursor(0, 0);
88     lcd.print("Arrosage GHANAM");
89     lcd.setCursor(0, 1);
90     lcd.print("en cours");
91
92     // Activation de l'électrovanne et des LED vertes
93     digitalWrite(relaisPin, HIGH);
94     digitalWrite(ledVertePin, HIGH);
95     digitalWrite(ledRougePin, LOW);
96
97     // Attendre que l'humidité atteigne 40%
98     while (map(analogRead(humiditePin), 0, 1023, 0, 100) < 40) {
99         lcd.clear();
100         lcd.setCursor(0, 0);
101         lcd.print("Humidite atteint");
102         lcd.setCursor(0, 1);
103         lcd.print(" 40 % ");
104         delay(30000);
105     }
106
```

```
107     // Désactivation de l'électrovanne et des LED vertes après l'arrosage
108     digitalWrite(relaisPin, LOW);
109     digitalWrite(ledVertePin, LOW);
110     digitalWrite(ledRougePin, HIGH);
111
112     // Affichage de l'état de l'arrosage terminé
113     lcd.clear();
114     lcd.setCursor(0, 0);
115     lcd.print("Arrosage GHANAM");
116     lcd.setCursor(0, 1);
117     lcd.print("termine");
118
119     delay(30000); // Attendre 30 secondes
120 } else {
121     // Affichage de l'arrosage désactivé et des causes
122     lcd.clear();
123     lcd.setCursor(0, 0);
124     lcd.print("Arrosage GHANAM");
125     lcd.setCursor(0, 1);
126     lcd.print("desactive");
127
128     delay(30000); // Attendre 30 secondes
129     lcd.clear();
130     lcd.setCursor(0, 0);
131     lcd.print("Les causes :");
132     delay(10000); // Attendre 10 secondes

```

Figure 27 : Programme Arduino du système d'arrosage automatique du terrain de football (suite).

Annexe 5 :

```
133
134 // Affichage des différentes causes de désactivation
135 if (pluie >= 350) {
136     lcd.clear();
137     lcd.setCursor(0, 0);
138     lcd.print("Cause 1:");
139     lcd.setCursor(0, 1);
140     lcd.print("Il pleut");
141     delay(10000); // Attendre 10 secondes
142 }
143
144 if (temperature < 20 || temperature > 35) {
145     lcd.clear();
146     lcd.setCursor(0, 0);
147     lcd.print("Cause 2:");
148     lcd.setCursor(0, 1);
149     lcd.print("Temp hors plage");
150     delay(10000); // Attendre 10 secondes
151 }
152
153 if (luminosite >= 500) {
154     lcd.clear();
155     lcd.setCursor(0, 0);
156     lcd.print("Cause 3:");
157     lcd.setCursor(0, 1);
158     lcd.print("Trop de lumi");
159     delay(10000); // Attendre 10 secondes
160 }
```

```
161
162     if (humidite >= 20) {
163         lcd.clear();
164         lcd.setCursor(0, 0);
165         lcd.print("Cause 4:");
166         lcd.setCursor(0, 1);
167         lcd.print("Humidite OK");
168         delay(10000); // Attendre 10 secondes
169     }
170 }
171 //Répéter le test toutes les 2 heures optimise les conditions d'arrosage
172 // et assure au moins 4 cycles d'arrosage chaque nuit.
173 delay(7200000);
174 }
```

Figure 28 : Programme Arduino du système d'arrosage automatique du terrain de football **(FIN)**.



Merci pour vous



Nom: **GHANAM**

Prénom: **Fatima_Ezzahra**

Numero CNC: **BM030T**

