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

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

Introduction

01 ❑ Maintenir le gazon en bonne santé.

02 ❑ Amélioration des Performances et de l'Expérience de Jeu.

03 ❑ Assurer une gestion plus écologique.

04 ❑ Efficacité et Gestion des Ressources.

05 ❑ Adaptabilité et Adoption de Technologies.

Pourquoi automatiser le processus d'arrosage d'un terrain de football ?

Comment automatiser le processus d'arrosage d'un terrain de football ?

TIPE 2024 GHANAM

TERRAIN DE FOOTBALL

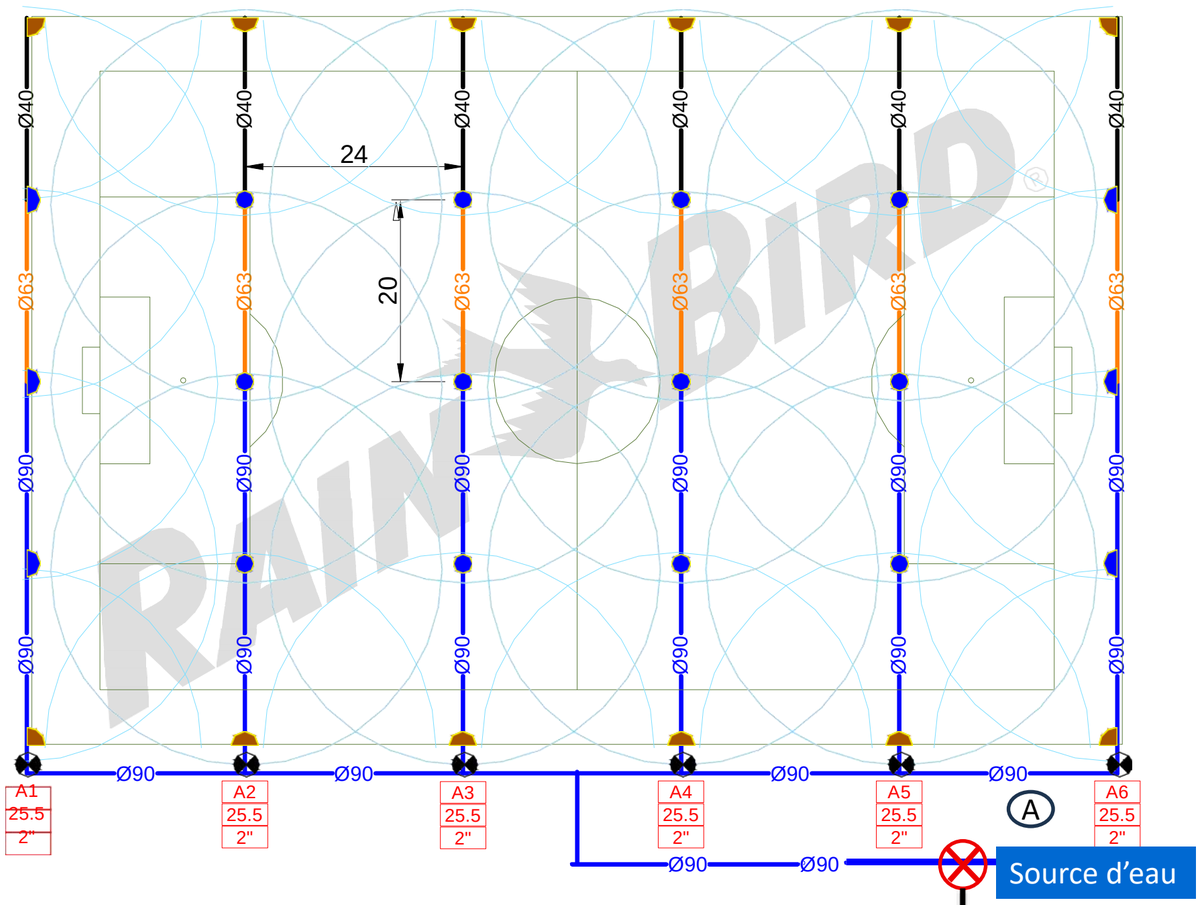
DIMENSIONS 120 x 80

105 + (2 x 7.5)

68 + (2 x 6.0)

ARROSEURS 8005

0 10 m



NOTES GENERALES

1. ETUDE REALISEE AVEC UN DEBIT MINIMUM DE 25.5 M³/H ET UNE PRESSION MINIMUN DE 6 BAR AU NIVEAU DE LA VANNE GENERALE DE SECTIONNEMENT .
2. CANALISATIONS EN POLYETHYLENE HAUTE DENSITE : PE63 PN10
3. IL EST POSSIBLE DE RAJOUTER DES RAMPES POUR L'ARROSAGE DE ZONES ANNEXES A CONDITION DE RESPECTER LES CAPACITES HYDRAULIQUES DU SYSTEME.
4. POSITION DES ARROSEURS A L'ECHELLE. POSITION SCHEMATIQUE DES CANALISATIONS.

LEGENDE

	Quantité
- ARROSEUR RAIN BIRD 8005 BUSE#22 PRESSION = 5 bar PORTEE = 22.4 m DEBIT = 5.84 m³/h	18
- ARROSEUR RAIN BIRD 8005 BUSE#16 PRESSION = 5 bar PORTEE = 20.4 m DEBIT = 3.99 m³/h	12
- VANNE GENERALE DE SECTIONNEMENT	1
- ELECTROVANNE RAIN BIRD	6
A PROGRAMMATEUR RAIN BIRD (6 STATIONS)	1

A1 / 25.5 / 2"

- PROGRAMMATEUR ET NUMERO DE STATION
- DEBIT DE LA RAMPE EN M³/H
- DIMENSION De L'ELECTROVANNE ELECTROVANNE INCORPOREE STATIONS

Un systéme d'arrosage automatique

Problèmatique:

Comment concevoir un système d'arrosage automatique abordable et simple pour un terrain de football, avec possibilité de contrôle à distance et autonomie énergétique ?

Objectifs

1- Analyse des Besoins via une Étude SysML

2- Étude des Composants

3- Conception du Système

4- Assurer l'Autonomie Énergétique

5- Commande à Distance par Smartphone

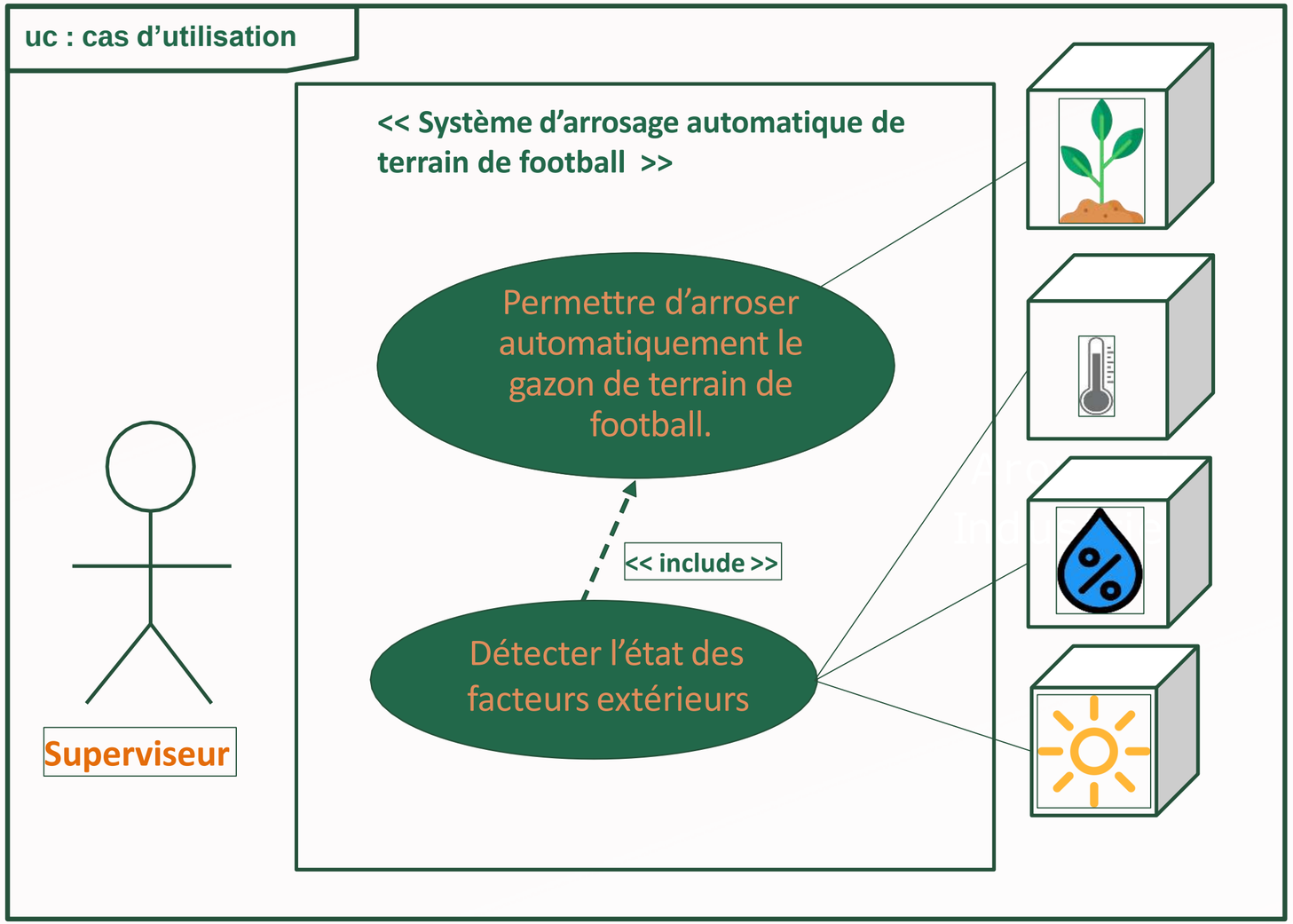
6- Étude Économique et Comparaison

7- Le profil final de notre système

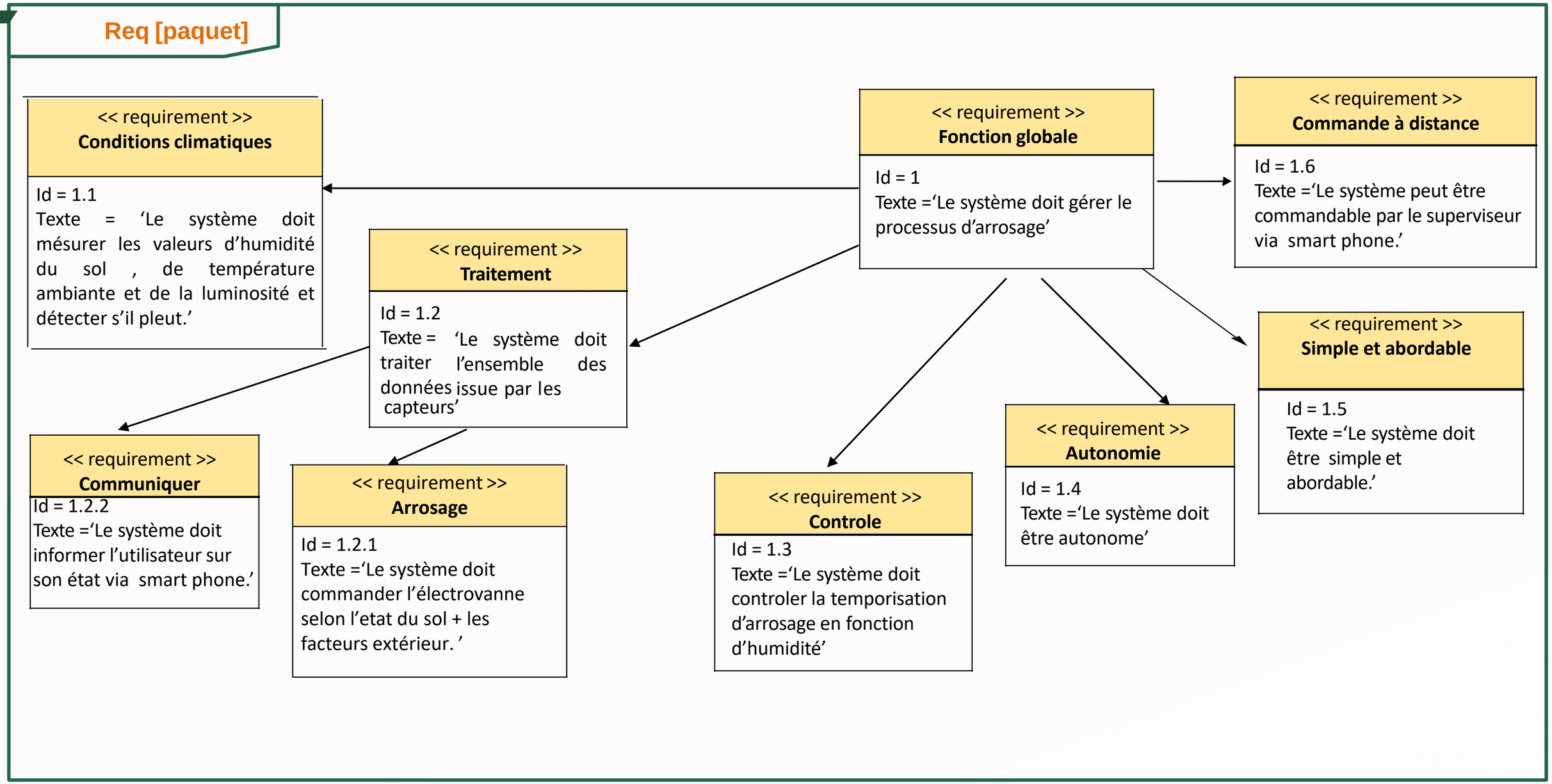
Objectif 1: Analyse des Besoins via une Étude SysML

- ❑ 1.1 Diagramme de cas d'utilisation
- ❑ 1.2 Diagramme des exigences
- ❑ 1.3 Fonctionnement global

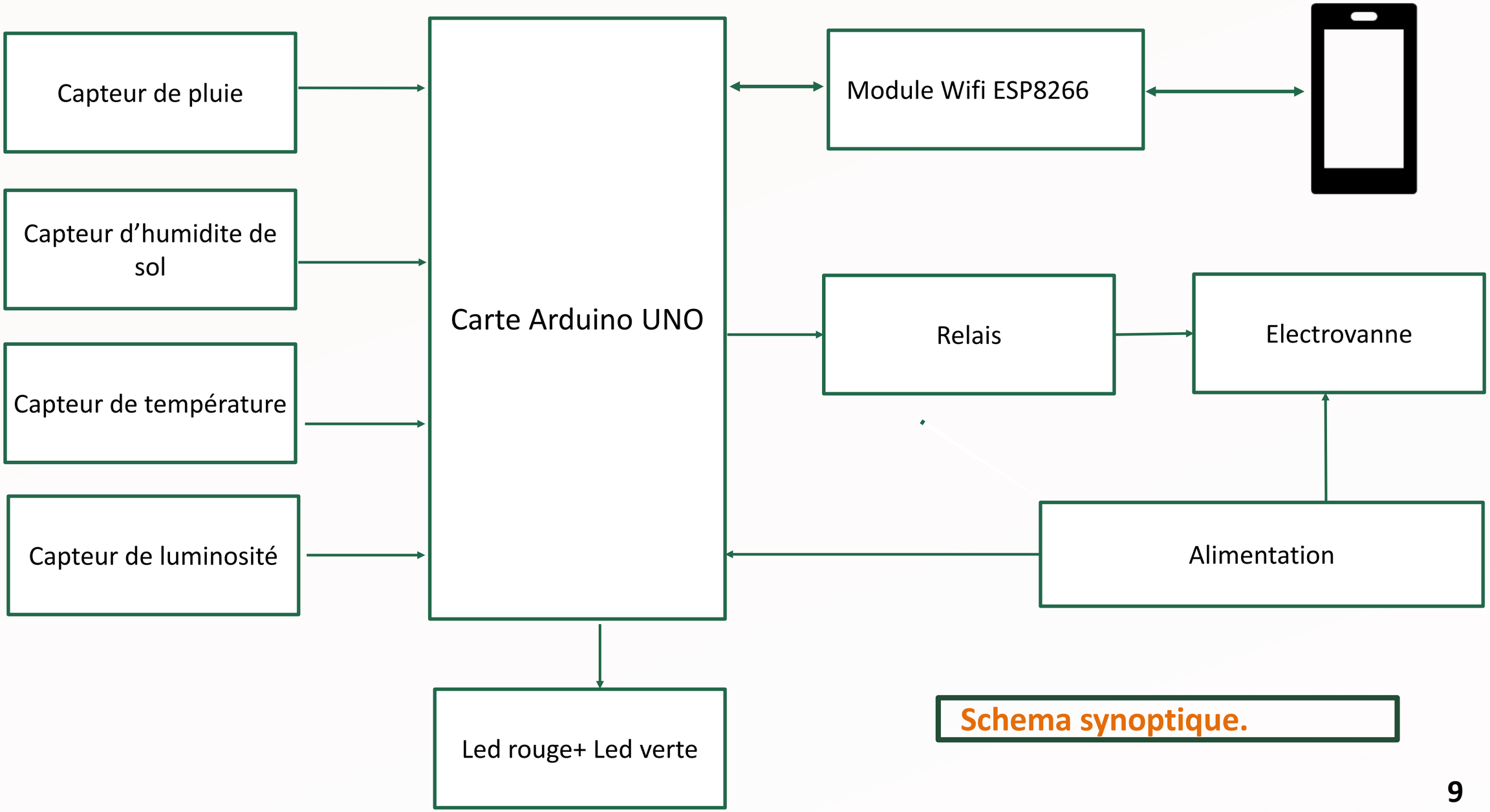
1.1 Diagramme de cas d’utilisation



1.2 Diagramme des exigences



1.3 Fonctionnement global



Schema synoptique.

Objectif 2: Étude des Composants

2.1. Sélection et test des capteurs :

- 2.1.1. Capteur d'humidité
- 2.1.2. Capteur de température
- 2.1.3. Capteur de pluie
- 2.1.4. Capteur de luminosité (LDR)

2.2. Unité de commande d'électrovanne :

- 2. 2.1.Caractéristiques de l'électrovanne,relais et génerateur de tension variable
- 2.2.2. Choix du raccordement entre tuyaux entrée/sortie avec l'électrovanne
- 2.2.3. 2.2.3. test de fonctionnement de l'électrovanne par bouton poussoir
- 2.2.4 Schéma de câblage de la partie commande de l'Électrovanne par Arduino

2.4. Avertisement visuel

2.1. Sélection et test des capteurs :

2.1.1. Capteur d'humidité:

2.1.1.1 Choix du capteur d'humidité :

Figure 1 : Capteur d'humidité résistif

Caractéristiques

Alimentation électrique :
3,3 V ou 5 V DC
Signal de tension de sortie:
0~4,2 V DC
Courant : 35 mA

Corrosif

Figure 2 : Capteur d'humidité capacitif

Caractéristiques

Alimentation électrique :
3,3 V ou 5,5 V DC
Signal de tension de sortie:
0~3V DC
Courant : 5 mA

Non Corrosif

Donc le choix du capteur est fixé sur le **capteur d'humidité capacitive**

2.1. Sélection et test des capteurs :

2.1.1. Capteur d'humidité:

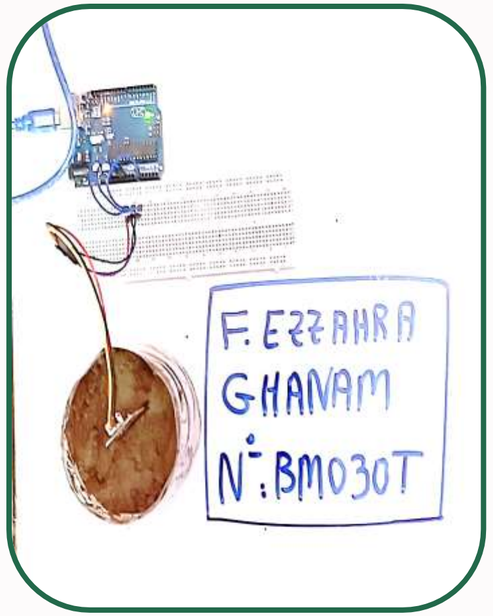
2.1.1.2 Test du capteur d'humidité :

Programme Arduino en annexe 1

- **Sol Sec**
- **0% à 20%**

Humidité du sol : 0.00%

- **Sol Légèrement Humide**
- **20% à 40%**

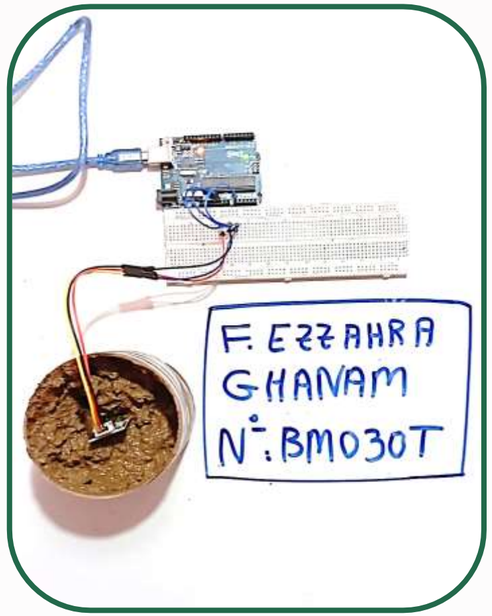


Humidité du sol : 36.00%

- **Sol Humide**
- **40% à 60%**

Humidité du sol : 51.00%

- **Sol Très Humide**
- **60% à 80%**



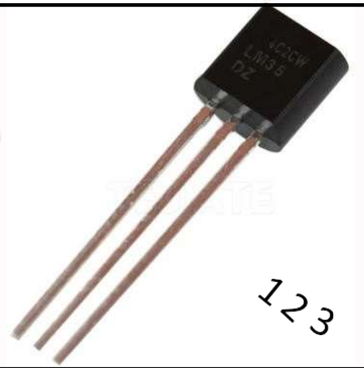
Humidité du sol : 68.00%

- **Sol Détrempé**
- **80% à 100%**

Humidité du sol : 87.00%

2.1.2. Capteur de température:

2.1.2.1.Choix du capteur de température :

	DHT11	LM35
	1: Vcc 2: GND 3: OUT	1: Vcc 2: GND 3: OUT 
Tension d'alimentation	3 à 5 V	4 V à 30 V
Courant de fonctionnement	0.5 mA	60µA
Écart de température	0 ~ +50°C	-55°C à + 155°C
Précision assurée	+/- 2°C	±1°C de -55°C à +150°C

- ✓ **Disponibilité au laboratoire**
- ✓ **Compatibilité avec la carte Arduino**

Donc le choix du capteur est fixé sur le **capteur de temperature LM35**

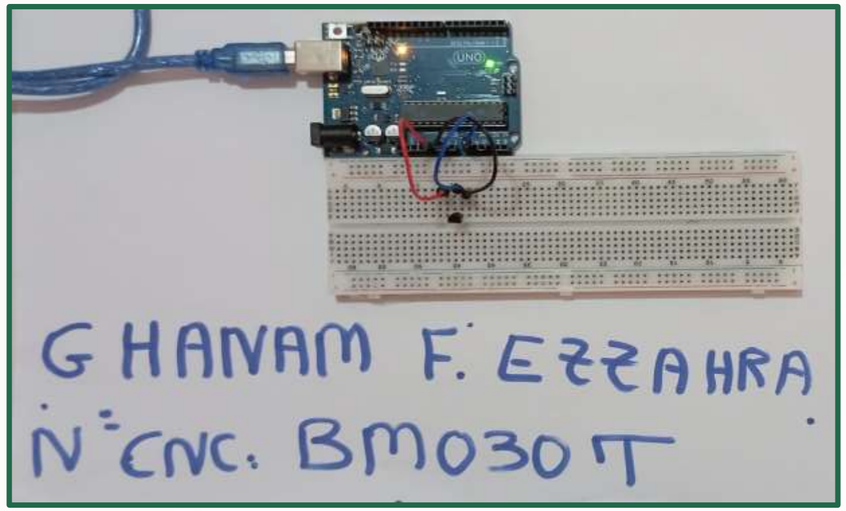
2.1. Sélection et test des capteurs :

2.1.2. Capteur de température:

2.1.2.2.Test du capteur de température :

Programme Arduino en annexe 2

Température ambiante



programme_mesure_temperature_capteur_LM35.ino

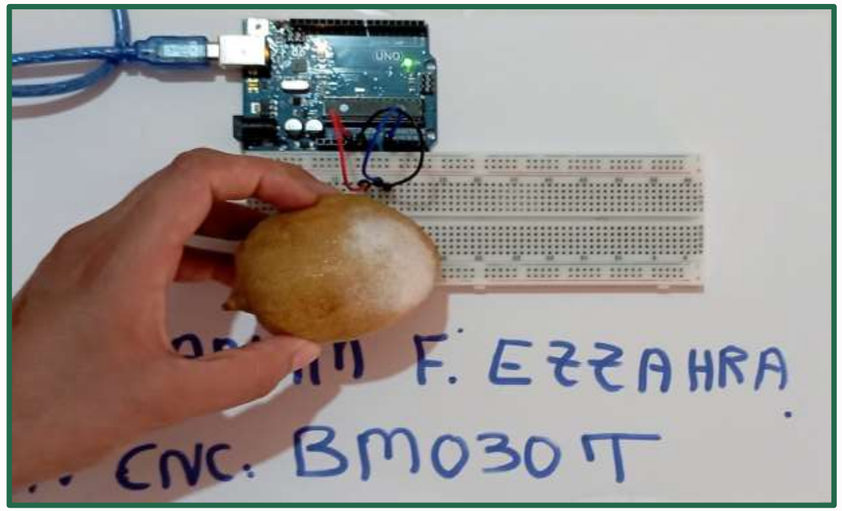
```
1  //Programme pour mesurer l température avec le capteur LM35
2  //Créé par Fatima-ezzhra GHANAM N°CNC BM030T
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Uno' on 'COM4')

```
temperature : 29.33 °C
```

Température froide



programme_mesure_temperature_capteur_LM35.ino

```
1  //Programme pour mesurer l température avec le capteur LM35
2  //Créé par Fatima-ezzhra GHANAM N°CNC BM030T
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Uno' on 'COM4')

```
temperature : 1.96 °C
```

Température chaude

programme_mesure_temperature_capteur_LM35.ino

```
1  //Programme pour mesurer l température avec le capteur LM35
2  //Créé par Fatima-ezzhra GHANAM N°CNC BM030T
```

Output Serial Monitor ×

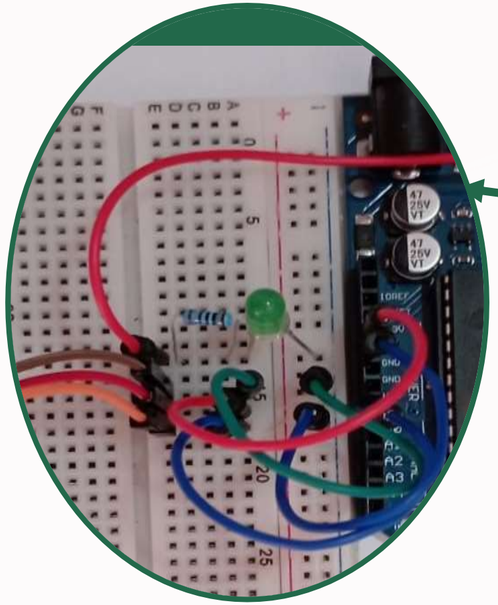
Message (Enter to send message to 'Arduino Uno' on 'COM4')

```
temperature : 52.30 °C
```

2.1.3. Capteur de pluie:

2.1.3.1.Test du capteur de pluie :

Programme Arduino en annexe 3



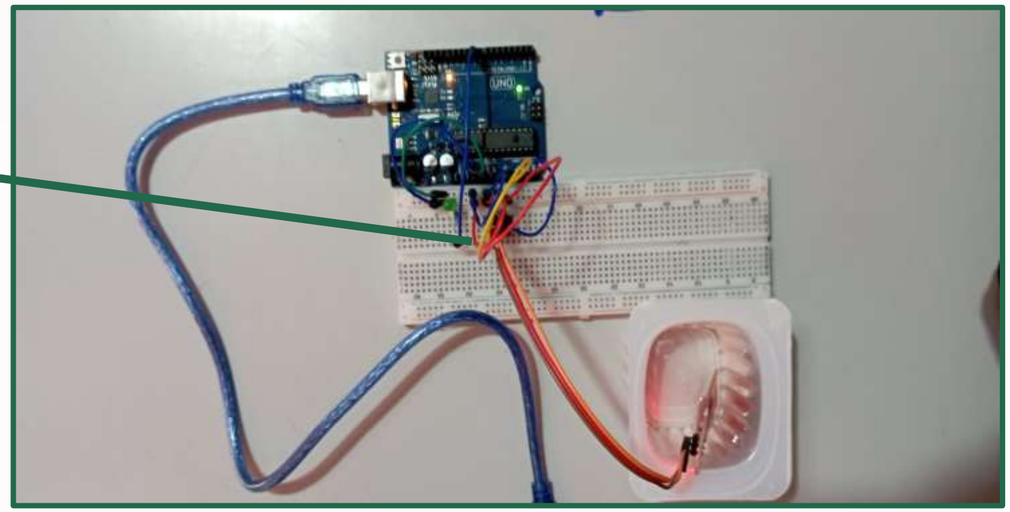


Figure 3 : Test du capteur de pluie émergé dans l'air.

Figure 4 : Test du capteur de pluie émergé dans l'eau.

programme-capteur-pluie.ino

```
// Programme Arduino pour tester les performances du capteur de pluie
// Performance du capteur de pluie
// Programme écrit par Fatima-ezzahra GHANAM
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Uno' on 'COM4')

```
Niveau d'eau : 0
Il ne pleut pas
```

programme-capteur-pluie.ino

```
// Programme Arduino pour tester les performances du capteur de pluie
// Performance du capteur de pluie
// Programme écrit par Fatima-ezzahra GHANAM
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Uno' on 'COM4')

```
Niveau d'eau : 757
Il pleut
```

2.1.4. Capteur de luminosité (LDR):

2.1.4.1. Mesure de la lumonisité à l'aide du capteur LDR :

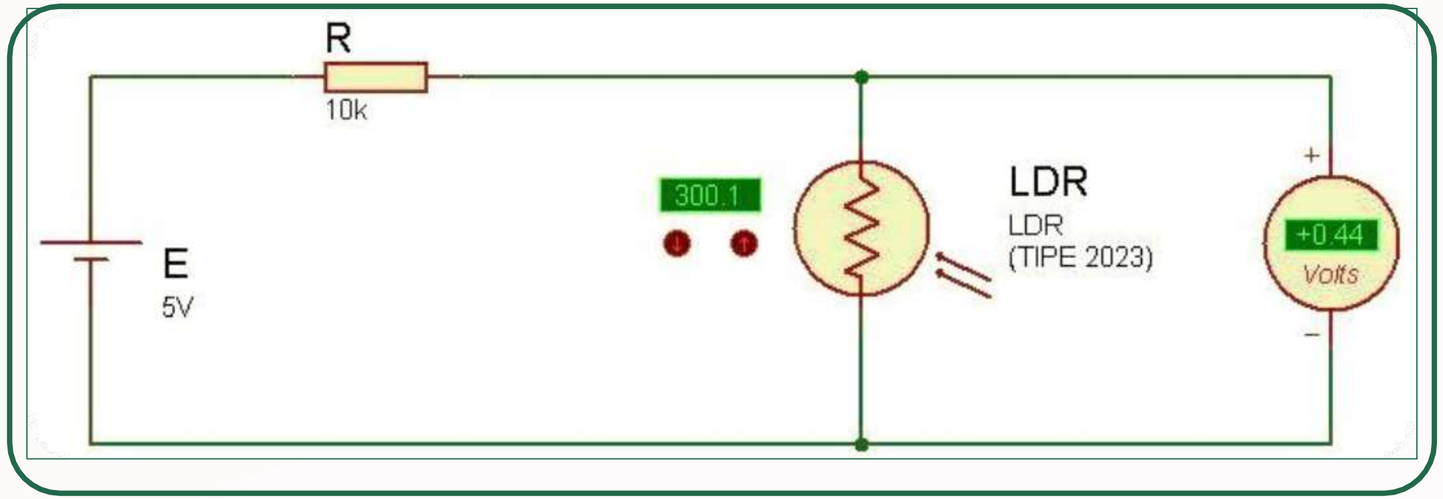


Figure 5 : Montage de conditionnement du capteur LDR sur le logociel (Proteus 8)

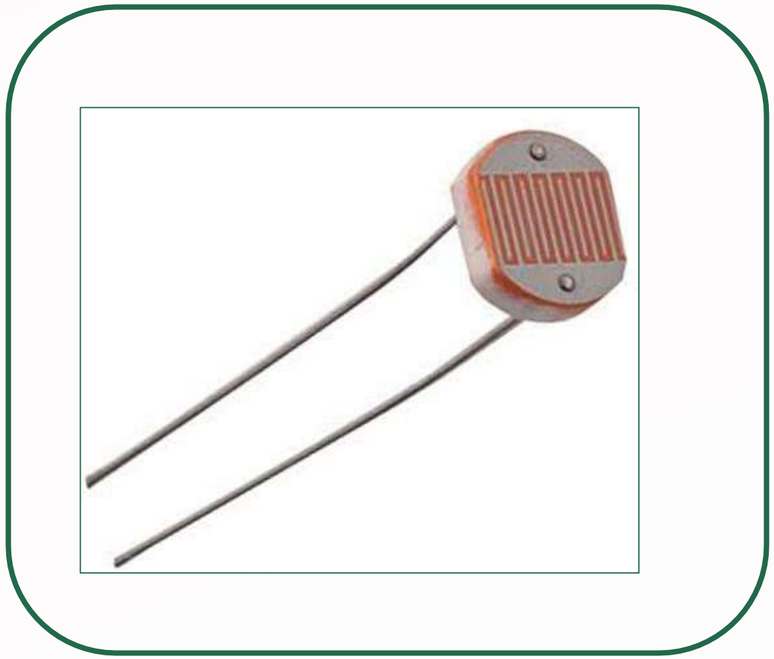


Figure 6 : Capteur de luminosité LDR (Light Dependent Resistor)

E : est la tension aux bornes du générateur (5V)

U_R : est la tension aux bornes de la résistance (R= 10kΩ)

U_{LDR} : est la tension aux bornes de la photorésistance (capteur)

Les caractéristiques de LDR en annexe **4**

2.1.4. Capteur de luminosité (LDR):

2.1.4.2. Détermination de $ULDR$ en fonction de l'éclairement :

Le programme de traçage à l'aide du langage **Python** se trouve en annexe 4.

Eclairement en (Lux)	U_{LDR} en (V)
0	4,1
10	3,18
100	0,98
200	0,59
300	0,44
400	0,35
500	0,29

Figure 7 : U_{LDR} en fonction Eclairement d'après : (*www.passionelectronique.fr)*

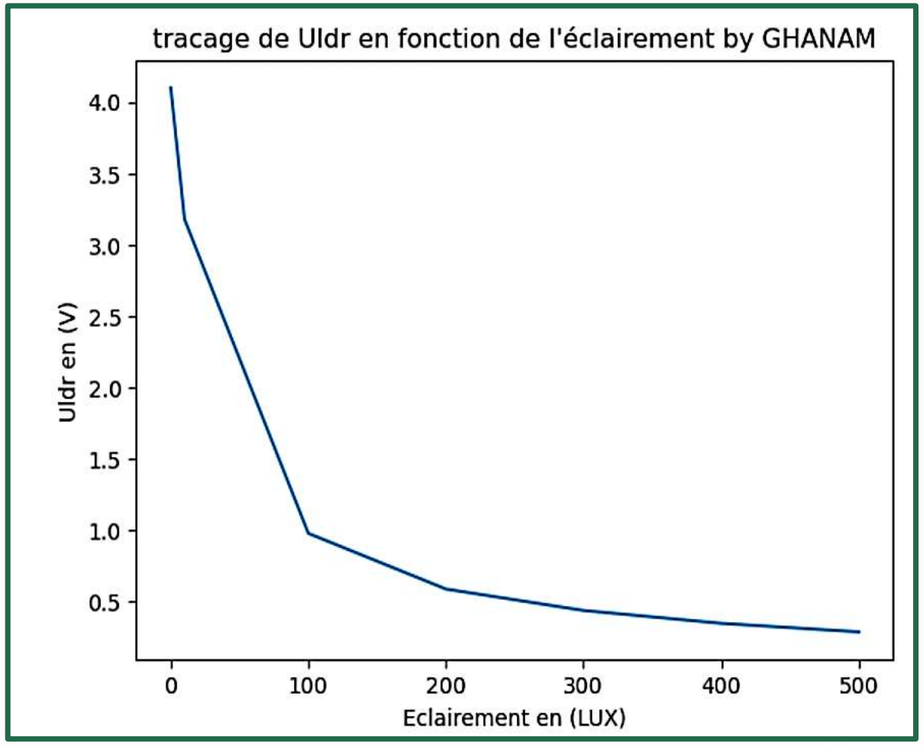


Figure 8 : l'évolution de la tension U_{LDR} en fonction de l'éclairement

Plus l'intensité lumineuse augmente plus la tension aux bornes de la photorésistance diminue.

2.1.4. Capteur de luminosité (LDR):

2.1.4.3. Test du capteur de luminosité (LDR): :

Programme Arduino en annexe 4

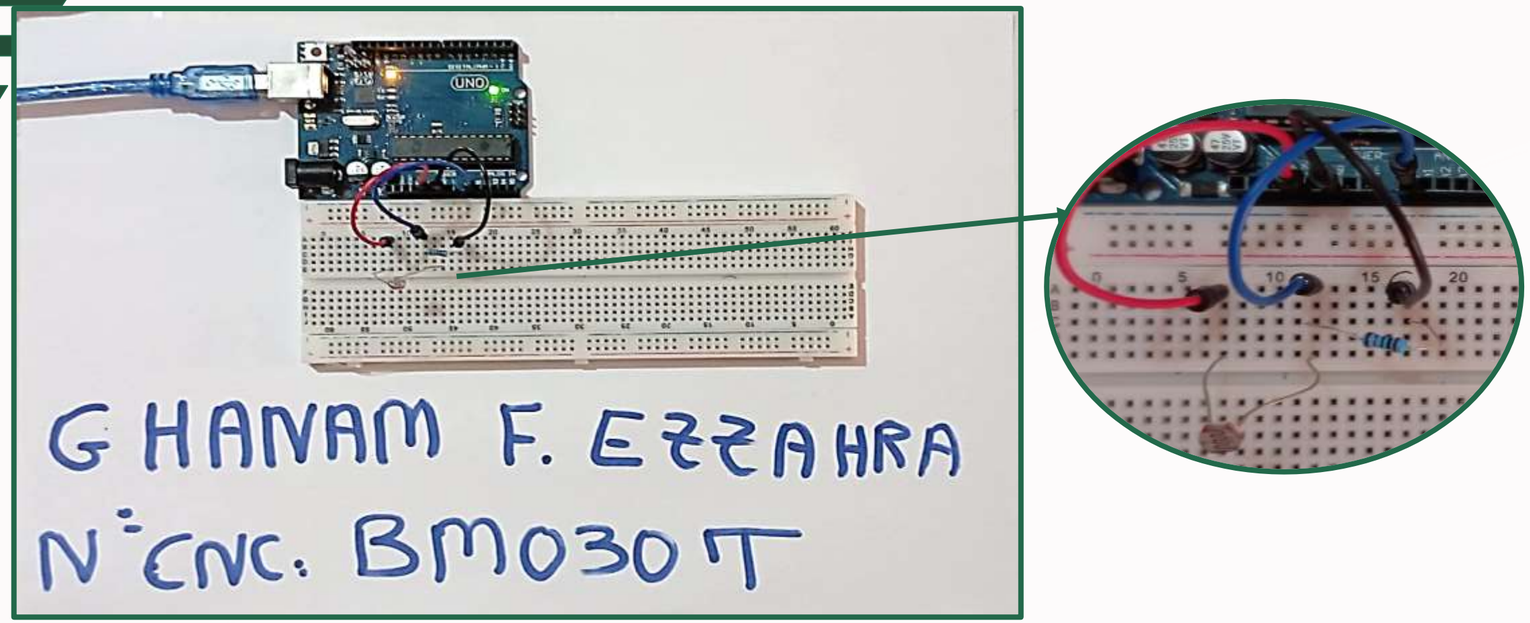


Figure 9 : Test du capteur de luminosité (LDR) en pleine soleil.

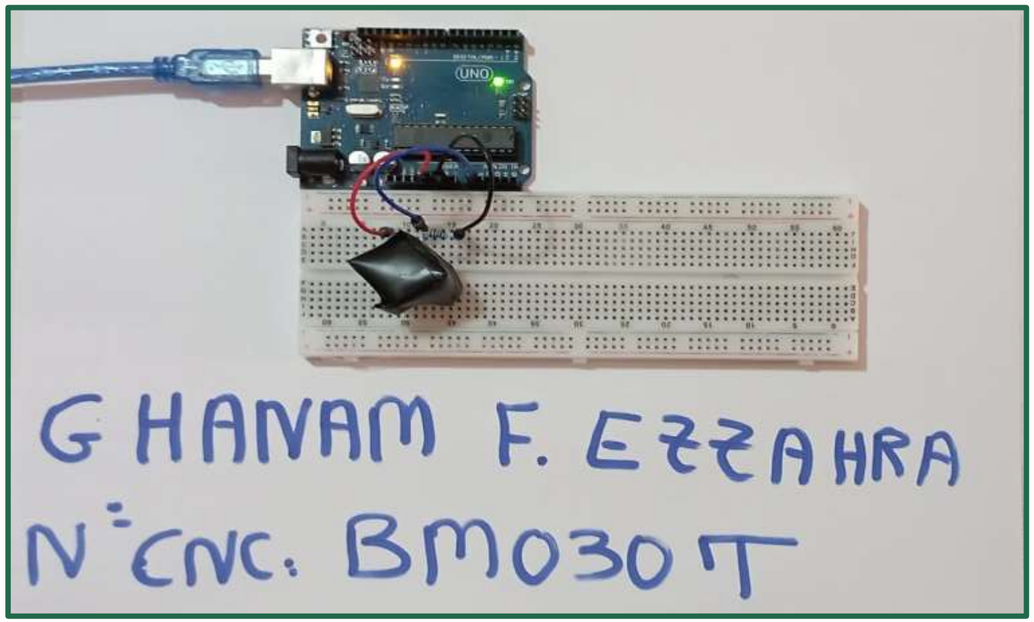


Figure 10 : Test du capteur de luminosité (LDR) au nocture

programme_luminosite_capteur_LDR.ino

```
//Programme pour mesurer la luminosité avec un capteur LDR
//Crée par Fatima-ezzahra GHANAM
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Uno' on 'COM4')

```
Luminosite : 638
```

programme_luminosite_capteur_LDR.ino

```
//Programme pour mesurer la luminosité avec un capteur LDR
//Crée par Fatima-ezzahra GHANAM
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Uno' on 'COM4')

```
Luminosite : 11
```

2.2. Unité de commande d'électrovanne :

2. 2.1.Caractéristiques de l'électrovanne,relais et génerateur de tension variable.

DC Power Supply ALF1501D

Relais

Electrovanne

Caractéristiques:

- **Description**:

•Alimentation symétrique avec 0 Volt commun aux deux sorties
•Sorties flottantes sur douilles de sécurité de 4mm
•Tension de sortie symétrique : ajustable de ±10 à ±15V

•Sortie caractéristiques:

- ±10 V ; 1.5A ±12V; 1.25A
- ±15 V ; 1 A

Caractéristiques:

- **JQC3F-5V DC-C Arduino**
- **Mode de déclenchement:** niveau haut
- Module relais 5V
- Compatible avec carte Arduino
- Signal de commande TTL 5V – 12V
- Peut contrôler le signal DC ou AC, également la charge 220V AC
- Contact normalement ouvert / fermé

Caractéristiques:

- **Tension:** DC 12VDC
- **Puissance maximale:** 14.4W
- **Max Courant:** 1.2A Max (pour 12V)
- **Courant nominal :** 1A
- **Entrée et sortie:** tuyau de 1/2 pouce (diamètre(extérieur)
- **un minimum de Pression**: 0.02- 0.8MPa
- une pression est nécessaire pour l'activer (0.02Mpa).
- **Température maximale du fluide:** 100 °C
- **Mode d'opération**: normalement fermé
- **Type de soupape:** diaphragme
- **Utilisation:** eau et fluides à faible viscosité.

2.2. Unité de commande d'électrovanne :

2.2.2. Choix du raccordement entre tuyaux entrée/sortie avec l'électrovanne

Critère	Colliers de serrage	Raccords spécefiques + Teflon plomberie bleu +
Avantages	Facile à installer, peu coûteux	Connexion étanche, réduit les fuites, durable, fiable
Inconvénients	Risque de fuites, entretien régulier	Coût initial élevé, installation technique
Recommandation	Option économique	Option recommandée pour étanchéité et fiabilité ✓

2.2. Unité de commande d'électrovanne :

2.2.3. test de fonctionnement de l'électrovanne par bouton poussoir

Figure 12 : Fonctionnement de l'electrovanne lorque l'alimentation est désactivée.

Figure 11 : Montage expérimental

Figure 13 : Fonctionnement de l'electrovanne lorque l'alimentation est activée.

2.2. Unité de commande d'électrovanne :

2.2.4 Schéma de câblage de la partie commande de l'Électrovanne par Arduino

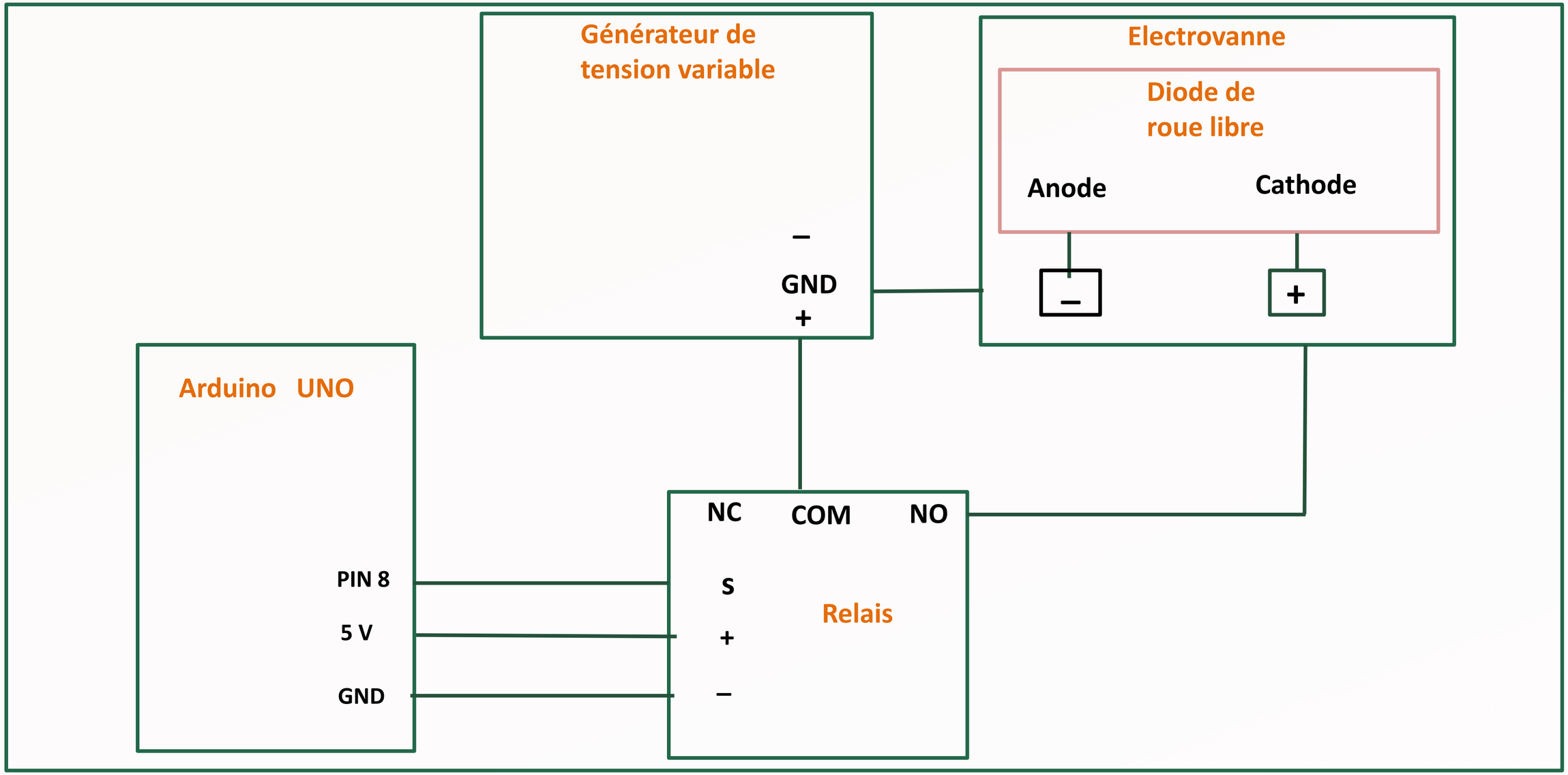


Schéma synoptique

2.4. Avertisement visuel

Outil / Caractéristique	Avantages	Inconvénients
MIT App Inventor	Contrôle à distance, interface conviviale.	Nécessite un smartphone, dépendant du Bluetooth.
Afficheur LCD 16*2	Affichage clair, faible consommation d'énergie	Limité aux chiffres, pas d'interaction directe.
LED Verte et Rouge	Indication visuelle claire, facile à intégrer.	Pas de détails des capteurs, aucune interaction.
Moniteur Série (Arduino IDE)	Utile pour le débogage, affiche les valeurs.	Requiert un PC connecté, usage limité sur site.

Objectif 3 :Conception du Système

3.1 Conditionnement d’un arrosage optimal

3.2 Algorithme du système

3.3 Montage expérimental

3.4 Tests de validation

3 .5 Conclusion

3.1 Conditionnement d'un arrosage optimal :

Humidité
[20% - 40%]

Luminosité < 500 lux

Température
[20°C – 35°C]

Pas de pluie
Niveau d'eau <350

Arrosage optimale du Gazon de Permudes
(Gazon hybride : réduction de 30%)

3.2 Algorithme du système

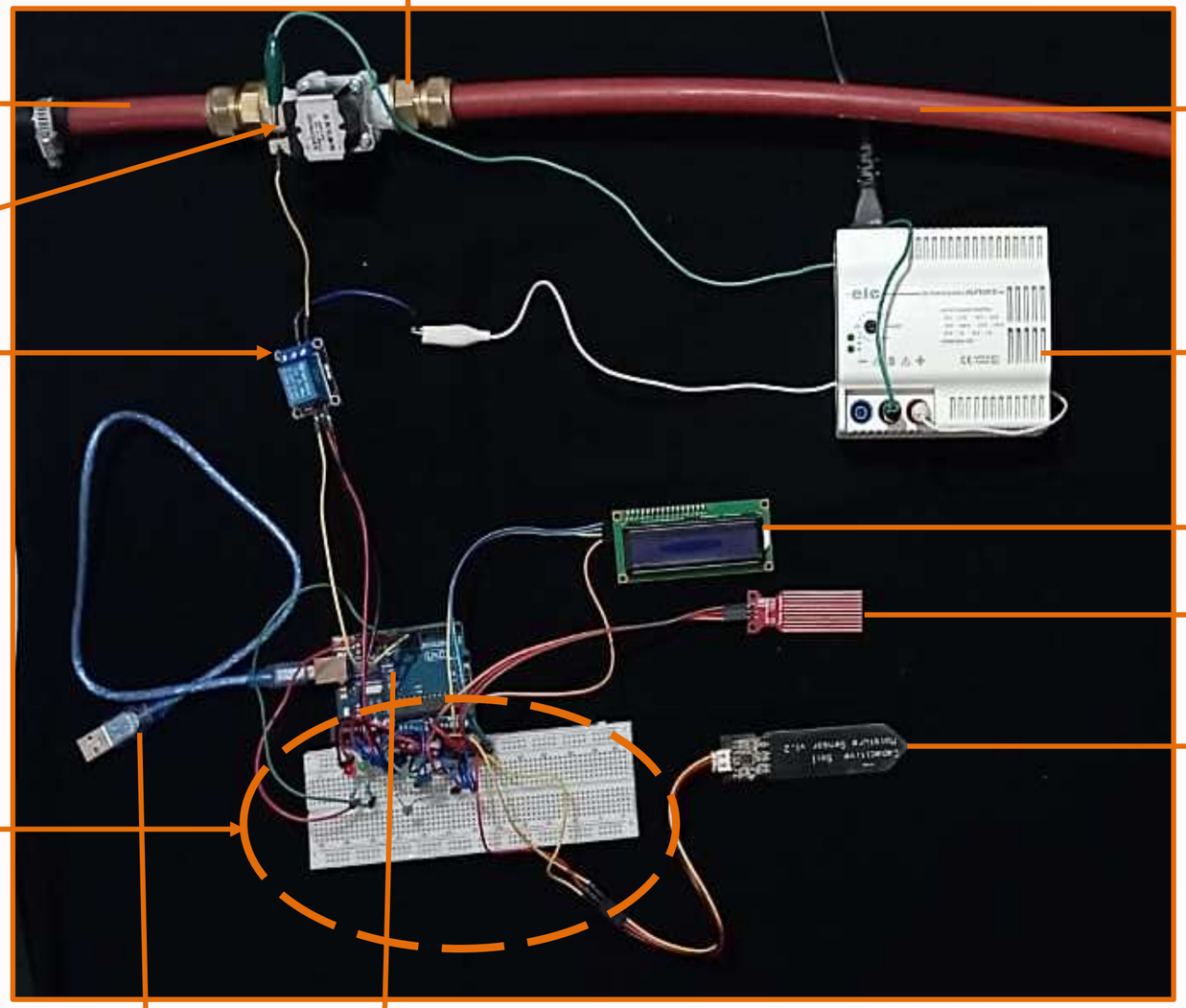
1. Initialisation

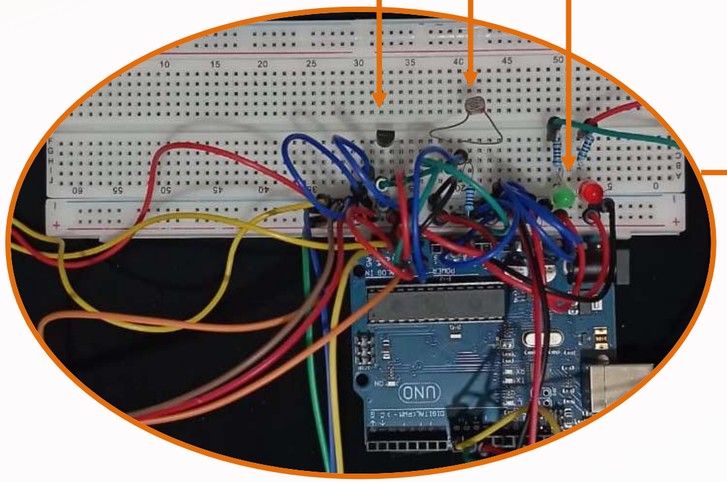
2. Boucle principale

1. Lire les valeurs des capteurs (humidité, température, luminosité, pluie)
2. Afficher les valeurs des capteurs
3. Vérifier les conditions :
   - Luminosité < 500
   - Température entre 20°C et 35°C
   - Pas de pluie
   - Humidité < 20%
4. Si toutes les conditions sont remplies :
   - Afficher "Arrosage GHANAM en cours "
   - Activer l'électrovanne
   - Allumer la LED verte
   - Éteindre la LED rouge
   - Continuer l'arrosage jusqu'à ce que l'humidité atteigne 40%
   - Si l'humidité atteint 40% :
     - Désactiver l'électrovanne
     - Éteindre la LED verte
     - Allumer la LED rouge
     - Afficher "Arrosage GHANAM terminé"
5. Sinon :
   - Afficher "Arrosage GHANAM desactivée"
   - Afficher les raisons pour les quelles l'arrosage est annulé
6. Attendre 2 heures
7. Recommencer la boucle

3.3 Montage experimental:

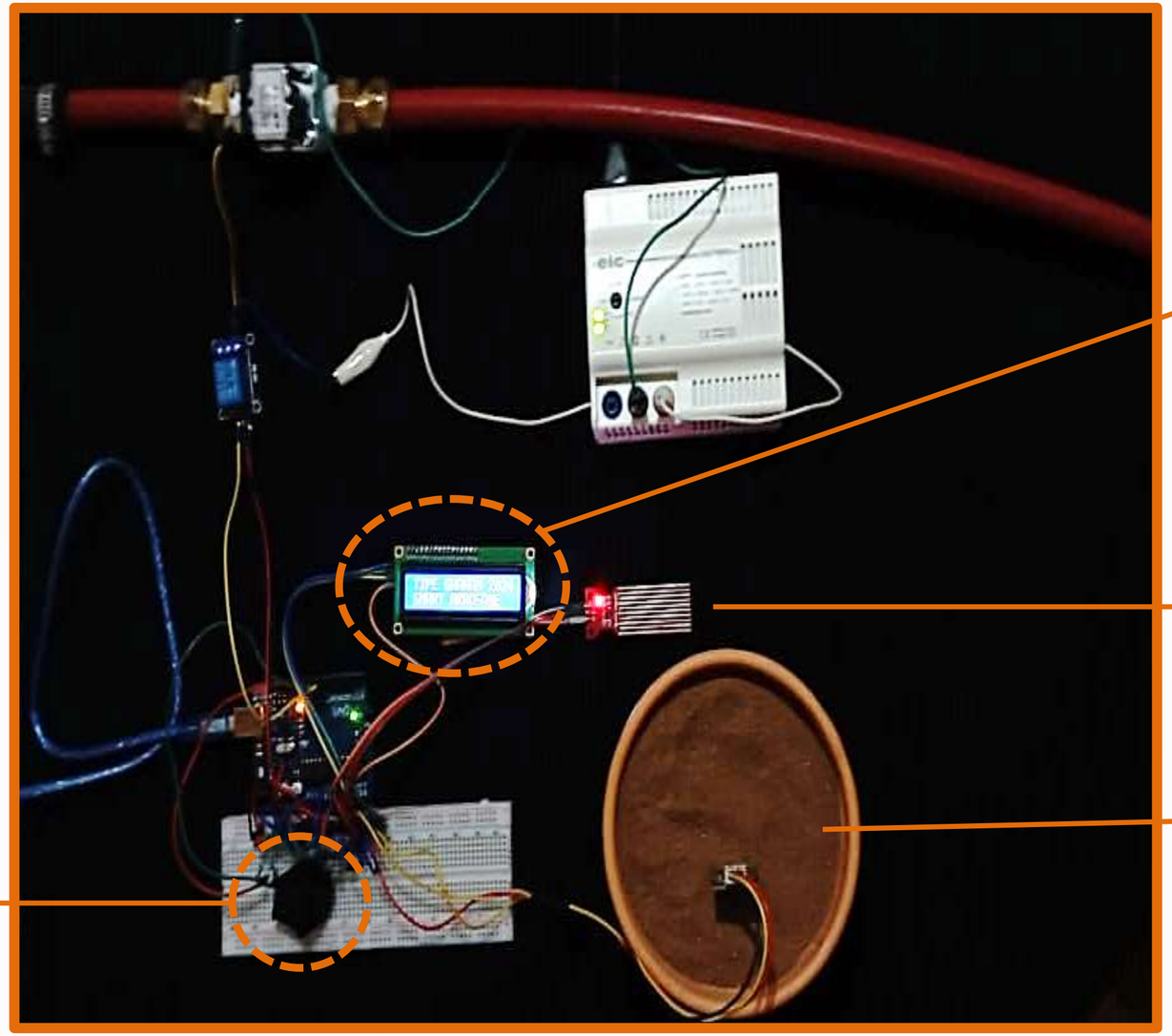
Programme Arduino en annexe 5





3 .4 Tests de validation

Scénario 1: Nuit sans lune sans pluie



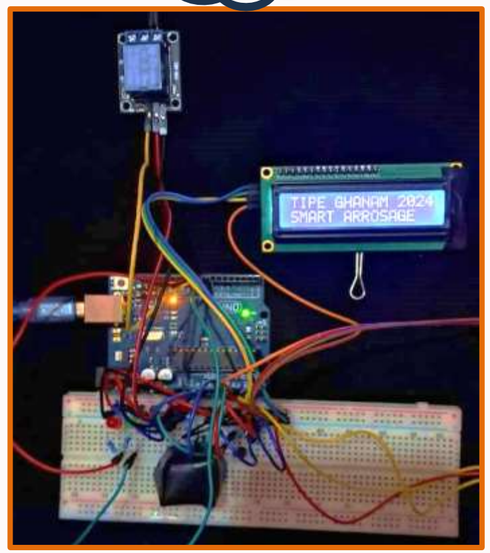
Pas de pluie

Sol sec

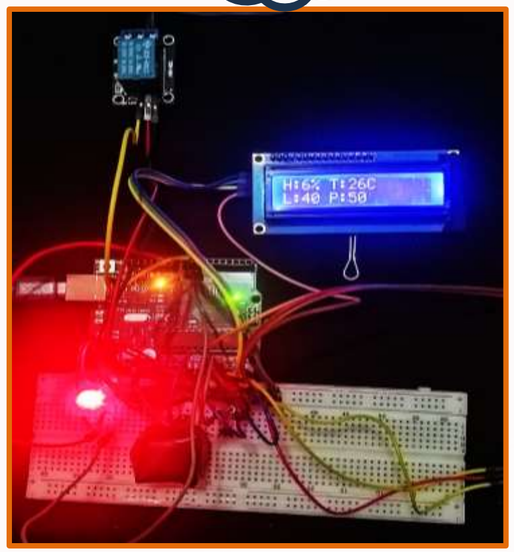
Une couverture opaque
a été utilisée
pour simuler
une nuit noire.

Résultat Scénario 1: Nuit sans lune sans pluie

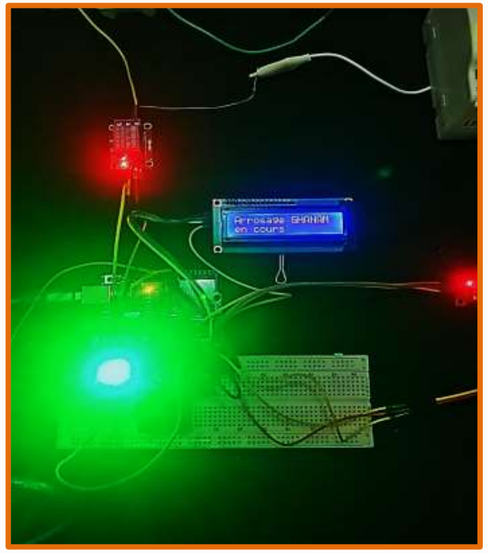
1



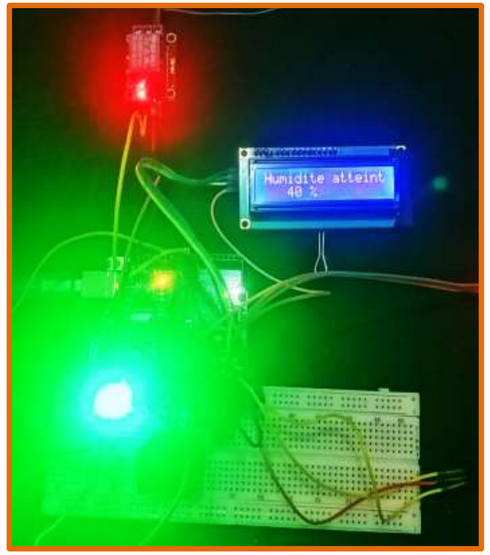
2



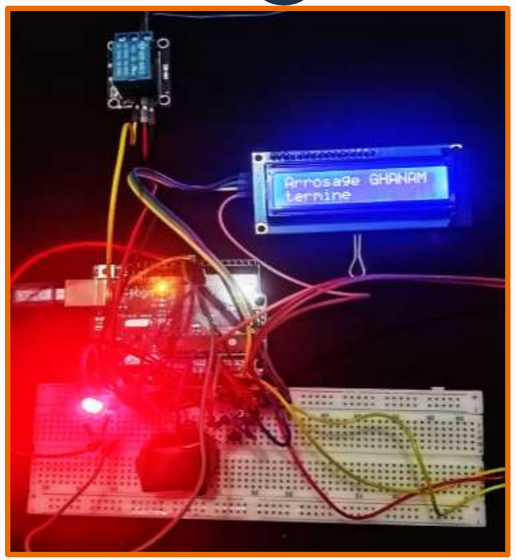
3



4



5



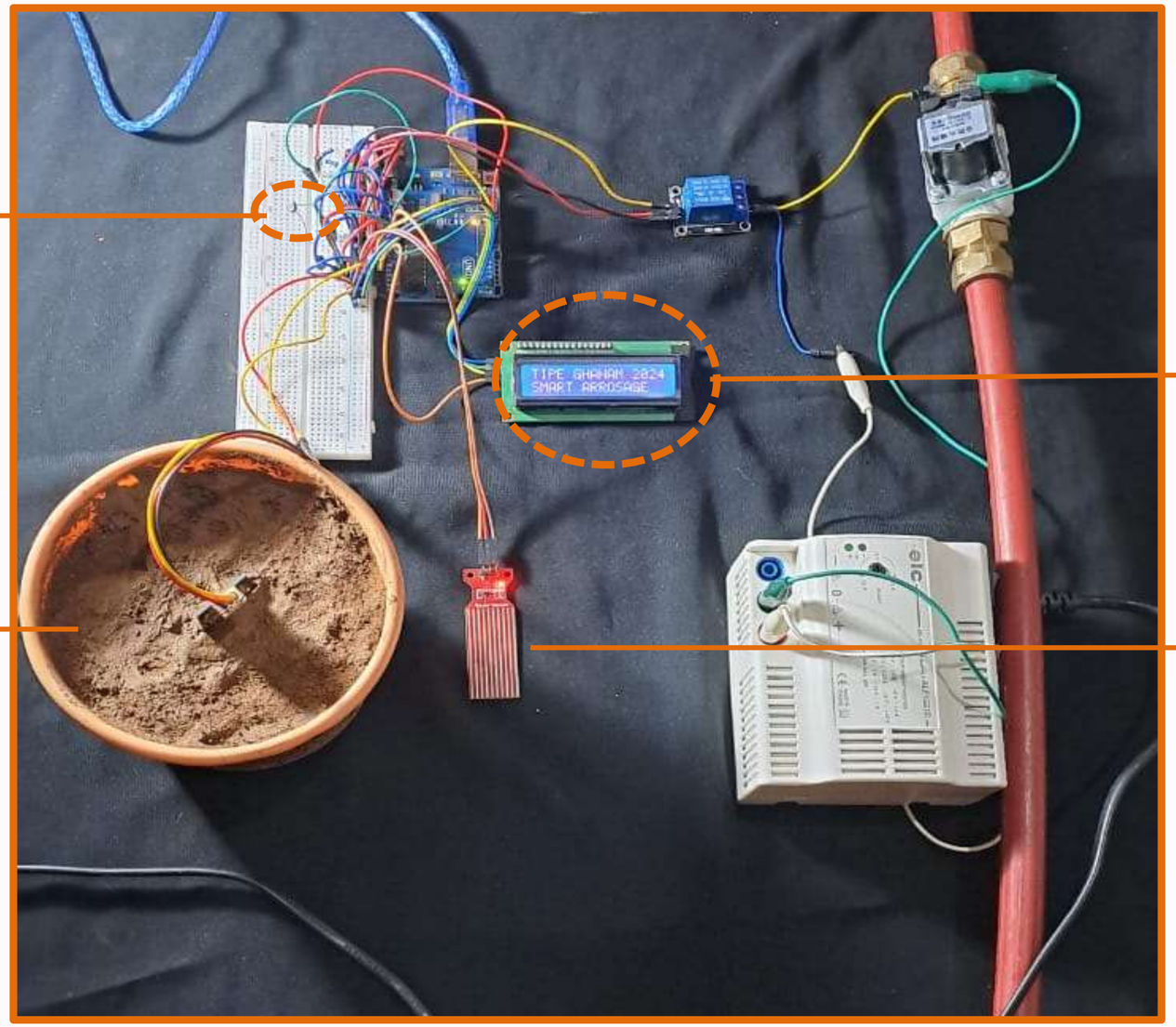
Test bien validé

3 .4 Tests de validation

Scénario 2: Journée chaude

Trop de luminosité

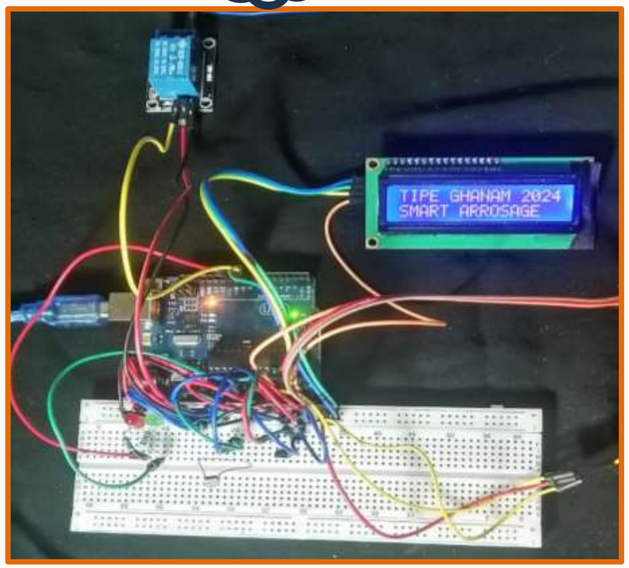
Sol Légèrement Humide



Pas de pluie

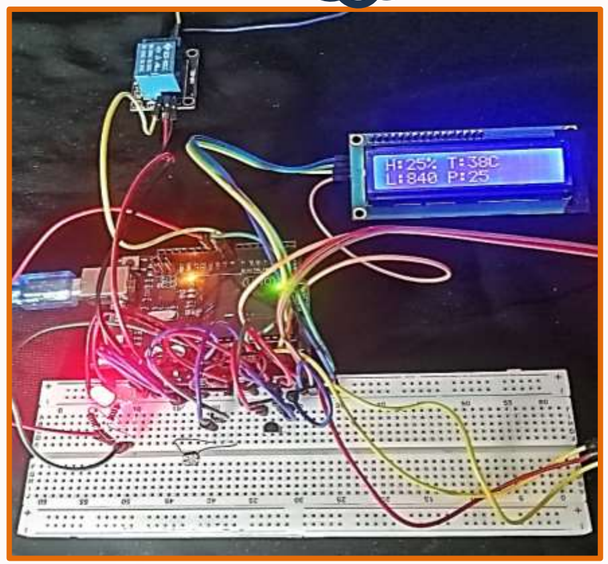
Resultat Scénario 2: Journée chaude

1



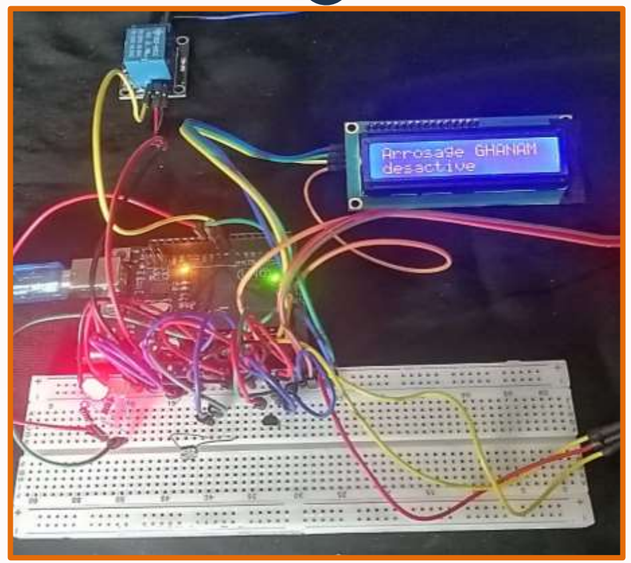
TIPE GHANAM 2024
SMART ARROSAGE

2



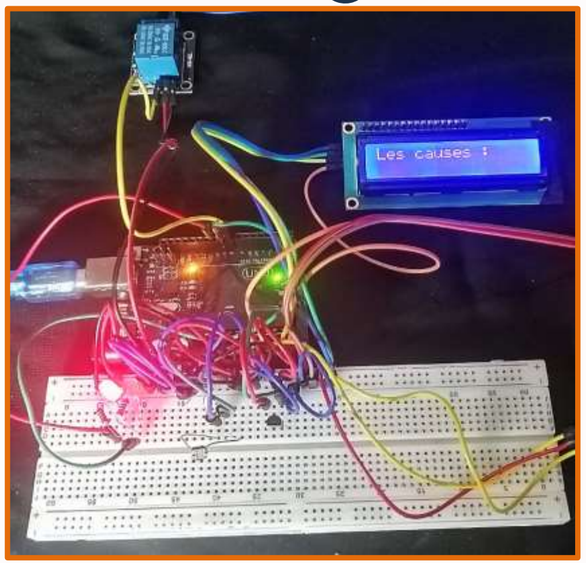
H:25% T:38C
L:840 P:25

3



Arrosage GHANAM
desactive

4

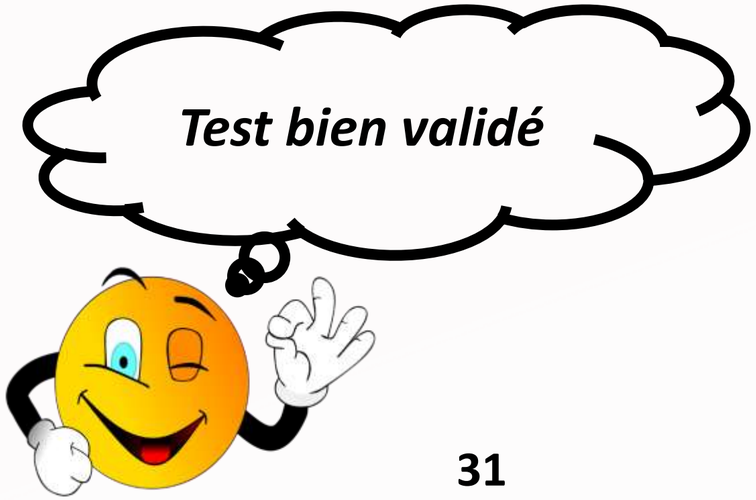


Les causes :

Cause 2:
Temp hors plage

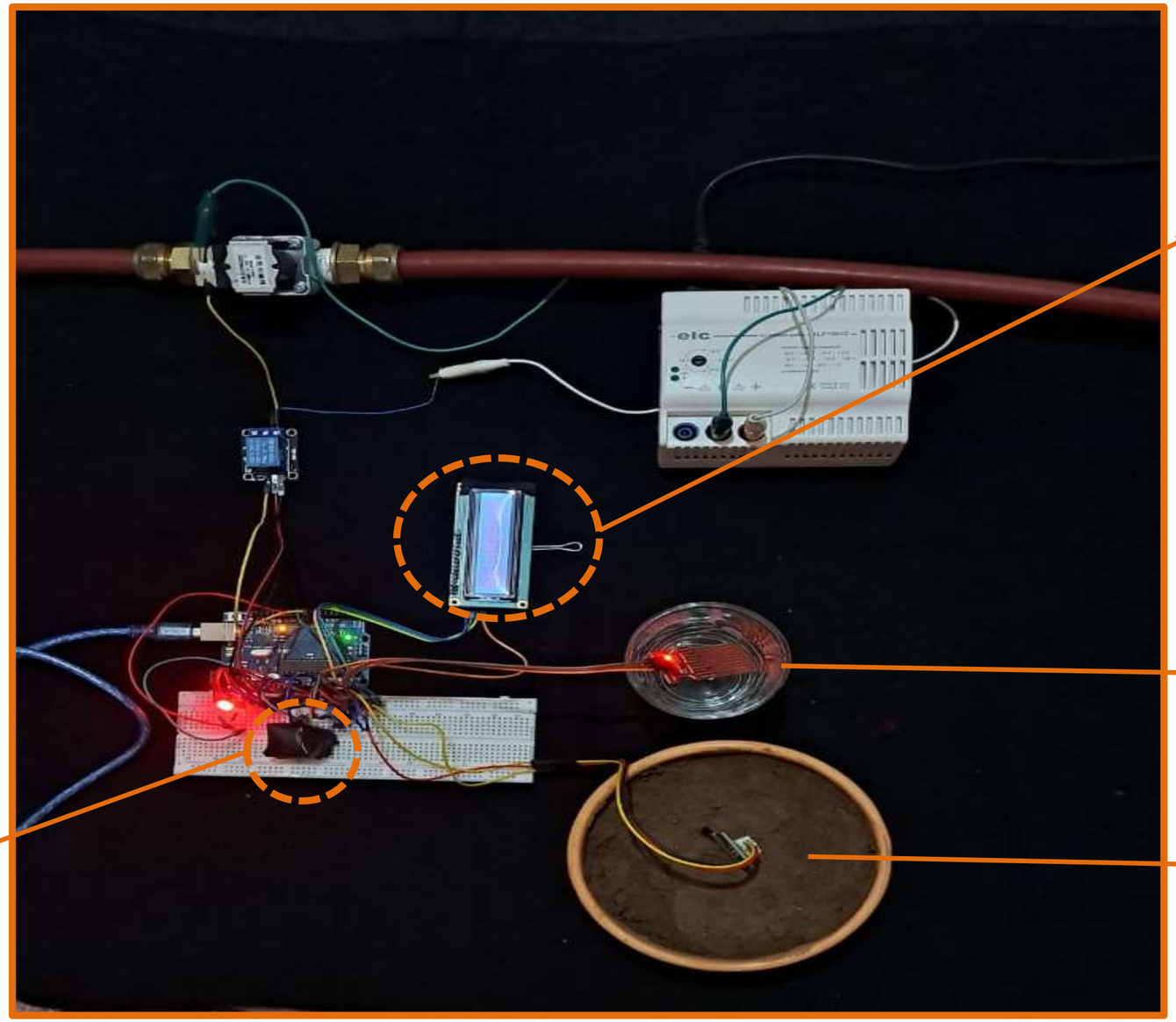
Cause 3:
Trop de lumi

Cause 4:
Humidite OK



3 .4 Tests de validation

Scénario 3 : Nuit pluvieuse



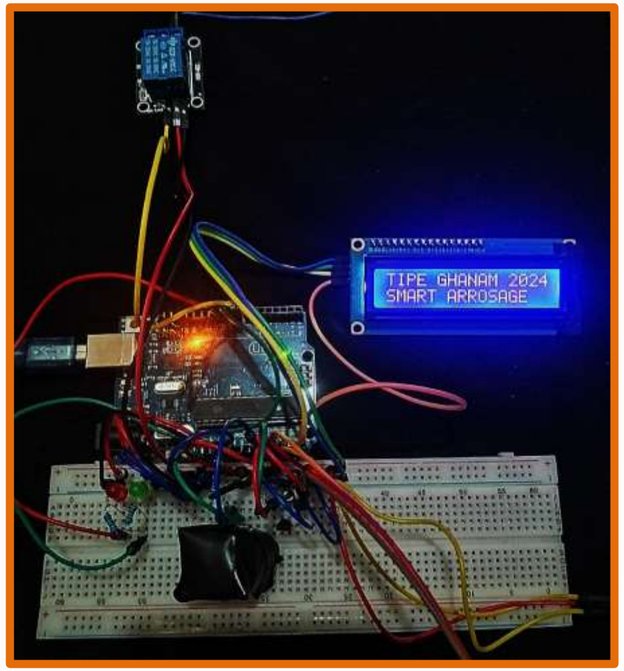
Il pleut

Sol un peu humide

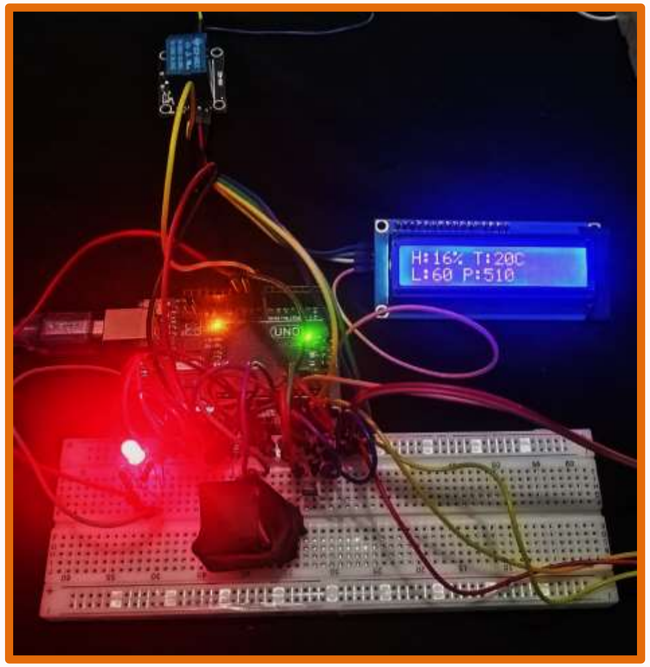
Une couverture opaque
a été utilisée
pour simuler
la nuit .

Résultat Scénario 3 : Nuit pluvieuse

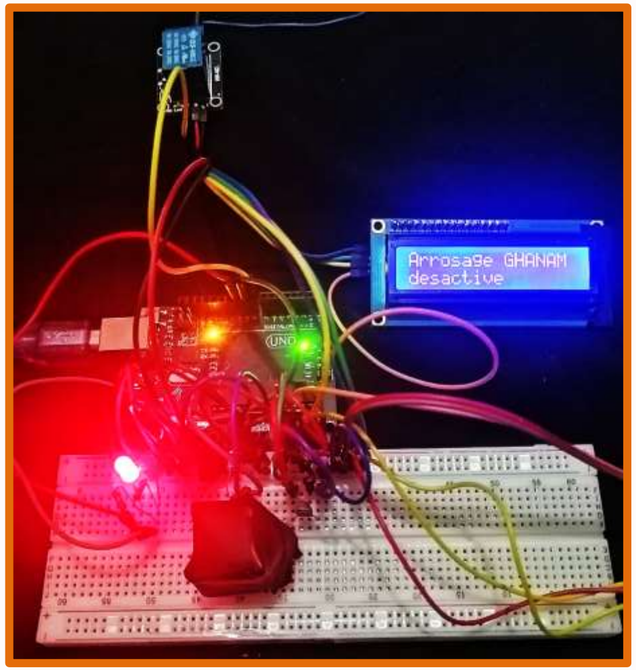
1



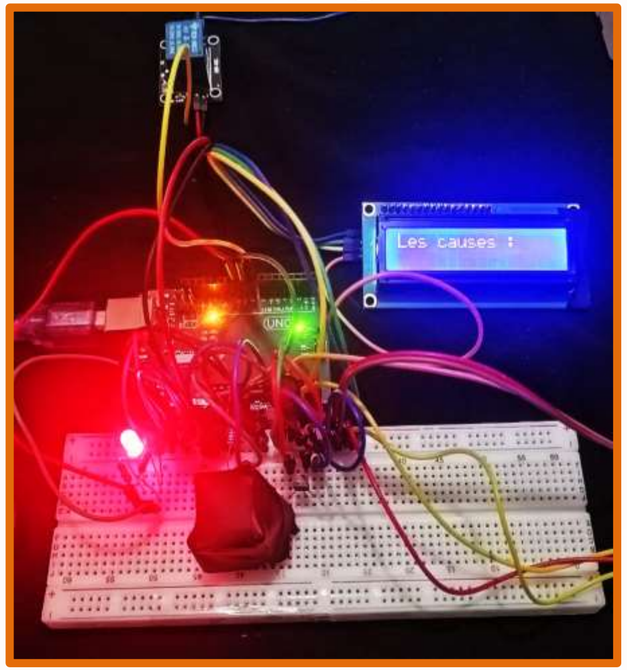
2



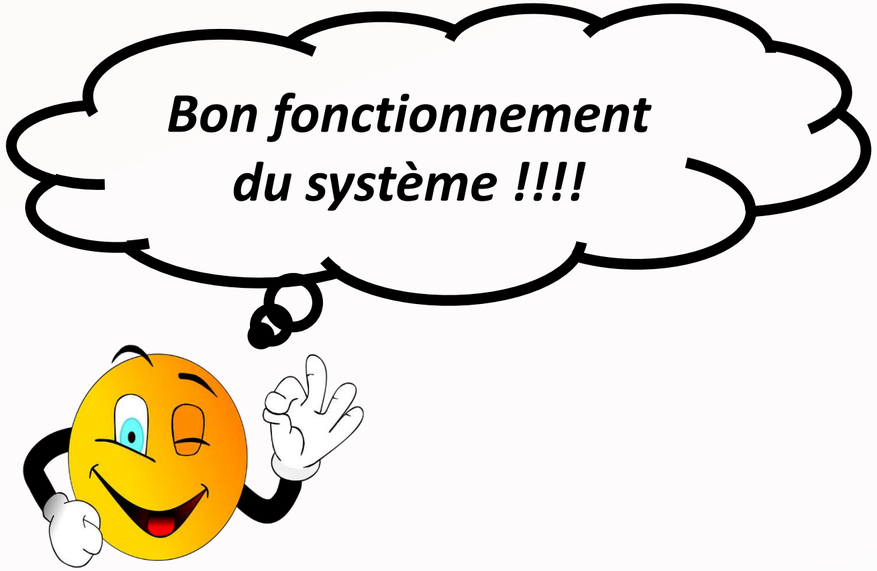
3



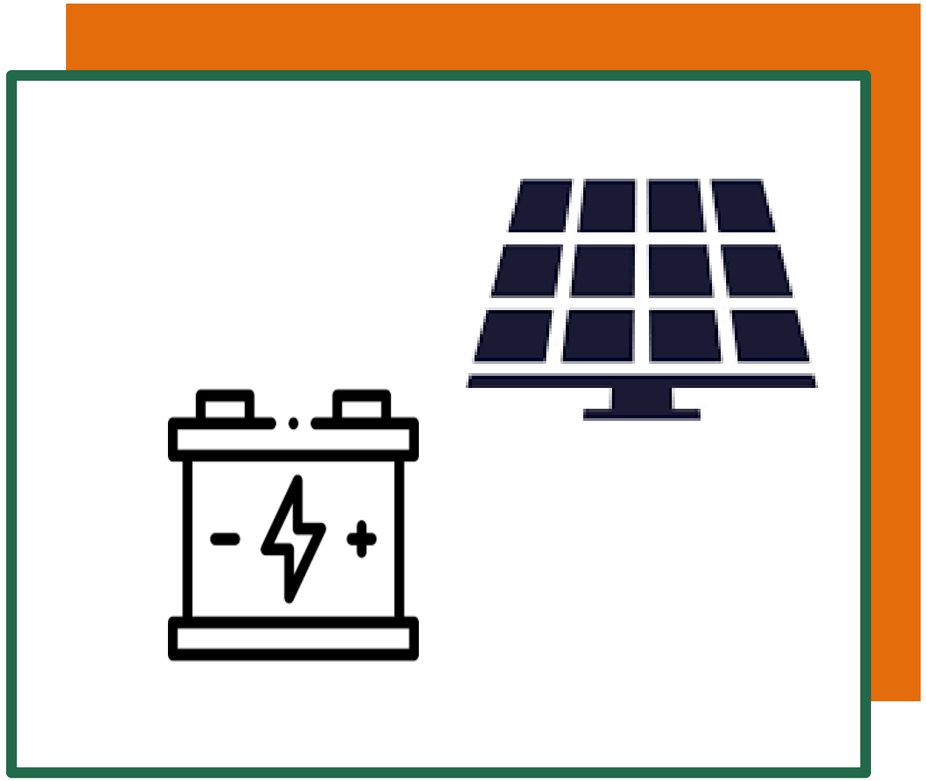
4



3 .4 Conclusion



Objectif 4 :Assurer l'Autonomie Énergétique du Système



4.1. Choix entre la pile et la batterie

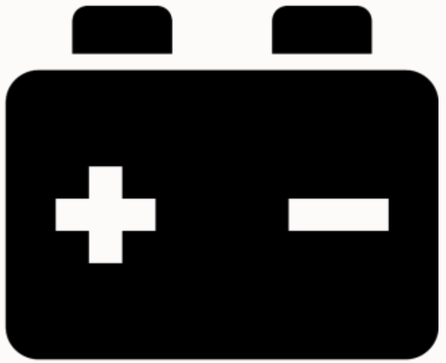
4.2 Branchement d’ensemble {Arduino UNO +l’Electrovanne +Batterie+Panneau solaire }

4.3Dimensionnement de la Batterie et du Panneau Solaire

4.4 Recommandation pour le choix de la Batterie et du Panneau Solaire

4.5 Conclusion

4.1. Choix entre la pile et la batterie



VS

Pourquoi une batterie et non pas une pile?

- Rechargeable
- Capacité plus élevée
- Meilleure longévité

- Non rechargeable
- Moins durable
- Moins économique à long terme

4.2 Branchement d'ensemble {Arduino UNO +l'Electrovanne +Batterie+Panneau solaire }

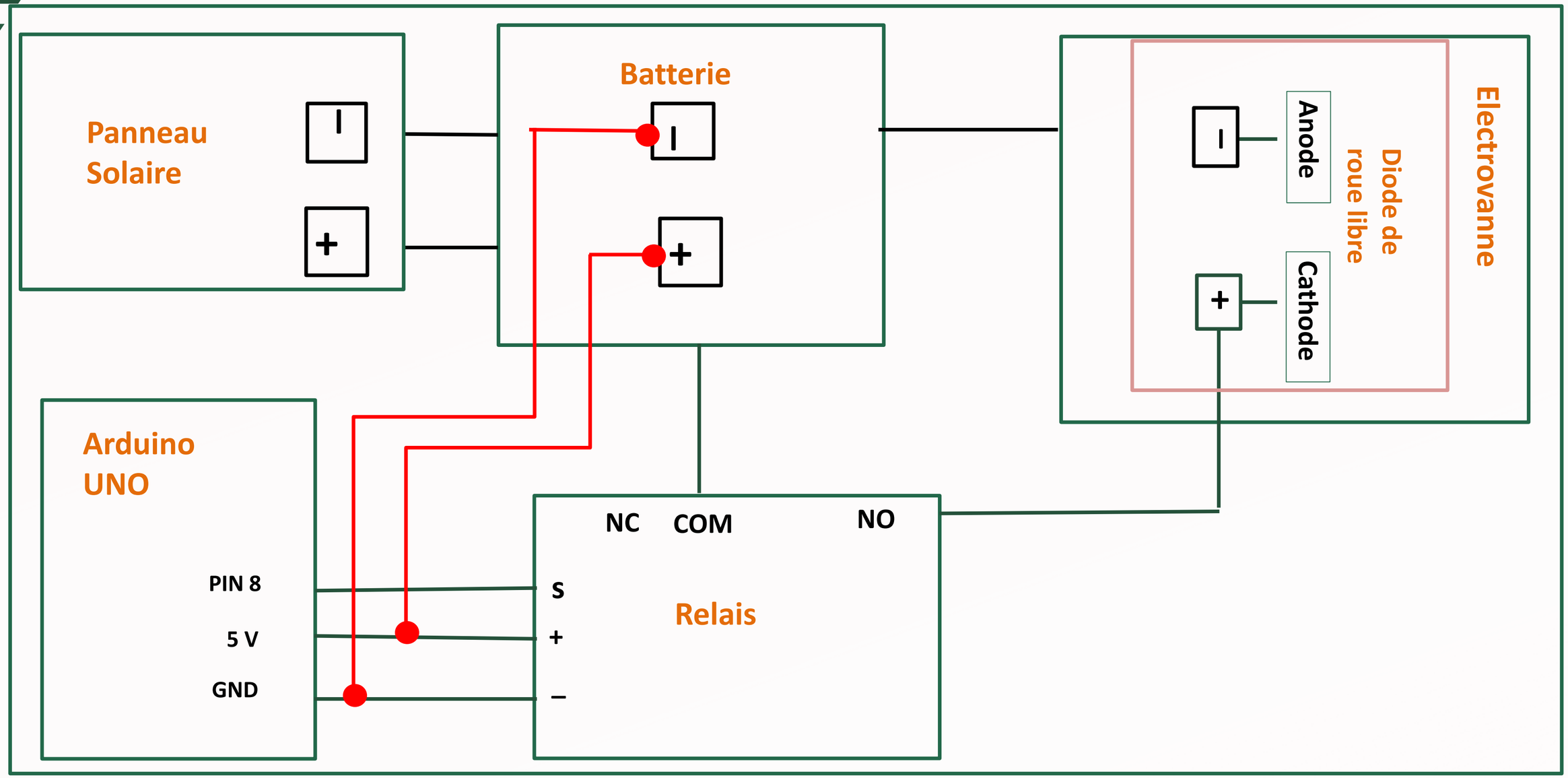


Schéma synoptique

4.3 Dimensionnement de la Batterie et du Panneau Solaire

1.Composants à alimenter

- **Carte Arduino UNO (alimentée par 12V)**:
  - Tension d'alimentation recommandée : 7-12V
- **Électrovanne**:
  - Tension: 12VDC
  - Puissance: 14.4W
  - Courant nominal: 1A

2.Consommation d'énergie

- **Arduino UNO**: Environ 50 mA à 12V
- **Consommation journalière:**

  Mode actif:

  - ❑ Tests (11 fois/jour, 3 minutes/test): 11 * 3 min * 50 mA = 1650 mA * min = 27.5 mA * h
  - ❑ Arrosage (1 heure/jour): 1h * 50 mA = 50 mA * h
  - **Total mode actif**: 27.5 mA * h + 50 mA * h = 77.5 mA * h

  Mode repos:

  - (24h - 1.55h) * 10 mA = 22.45h * 10 mA = 224.5 mA * h
  - **Total Arduino UNO**: 77.5 mA * h + 224.5 mA * h = 302 mA * h = 0.302 Ah/jour à 12V
- **Électrovanne**:
  - 1A à 12V
  - **Consommation journalière**: 1A * 1h = 1 Ah/jour

4.3 Dimensionnement de la Batterie et du Panneau Solaire

3.Calcul de la capacité de la batterie

•**Arduino UNO**: 0.302 Ah/jour à 12V
•**Électrovanne**: 1 Ah/jour à 12V
•**Consommation totale quotidienne**: 1 Ah (électrovanne) + 0.302 Ah (Arduino) ≈ 1.302 Ah

4.Sélection de la batterie

•**Batterie recommandée**: 12V, 2Ah
- **Autonomie**: 2Ah / 1.302Ah ≈ 1.5 jours (avec marge de sécurité)

5.Dimensionnement des panneaux solaires

•**Énergie nécessaire**: 1.302 Ah * 12V ≈ 15.62 Wh/jour
•**Panneau solaire recommandé**: 20W pour garantir la recharge complète en une journée

6.Conclusion

Pour garantir l'autonomie du système d'arrosage automatique pour une journée :
•**Choix de la batterie**: Batterie 12V, 2Ah, pour une autonomie d'environ 1.5 jours
•**Panneau solaire**: 20W pour assurer une recharge adéquate en une journée

Cette solution est économique, durable et efficace pour ce projet.

4.4 Recommandation pour le choix de la Batterie et du Panneau Solaire

1.Recommandation pour le choix d'une Batterie 12V 2Ah

Modèle	Caractéristiques	Prix	Où Acheter	Justification
ExpertPower 12V 2Ah	Durable, maintenance réduite	~20 €	Amazon	Bon rapport qualité/prix
CTEK 12V 2Ah	Léger, longue durée de vie	~35 €	Amazon	Haute performance et durabilité
GBA 12V 2.0Ah Professional	Robuste, bonne performance	~25 €	eBay	Idéal pour usage professionnel

2.Recommandation pour le choix d'une Panneaux Solaires 20W

Modèle	Caractéristiques	Prix	Où Acheter	Justification
Renogy 20W 12V	Monocristallin, 18% efficacité	~30 €	Amazon	Bon rapport qualité/prix
ALLPOWERS 20W 12V	Monocristallin, haute performance	~32 €	Amazon	Compact et léger
Sunway Solar 20W 12V	Monocristallin, durable	~35 €	eBay	Bon pour une recharge rapide

4.4 Conclusion

- ✓ Autonomie garantie
- ✓ Durabilité et efficacité optimales
- ✓ Budget raisonnable

- ✓ Une Batterie ExpertPower 12V 2Ah
- ✓ Un Panneau Solaire Renogy 20W 12V

Objectif 5: Implémentation de la Commande à Distance par Smartphone.

5.1. Développement de l'application mobile

5.2. Test de la commande à distance

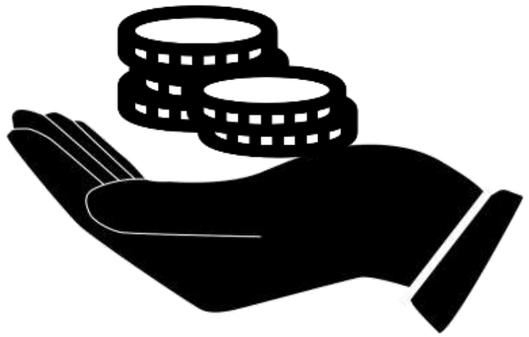
Objectif 5: Implémentation de la Commande à Distance par Smartphone.

En raison de contraintes de temps, je ne peux pas encore inclure la commande à distance. Néanmoins, je la garde en perspective pour les futures améliorations du projet.

Objectif 6 : Étude Économique et Comparaison

6.1. Évaluation du coût total

6.2. Comparaison avec l'arrosage manuel



6.1. Côut de notre système d'arrosage automatique

Le composant	Le prix	Le site d'achats
Capteur de pluie	15 DH	Micro Planet Maroc
Capteur de température LM35	12 DH	Micro Planet Maroc
Capteur de Luminosité LDR	6 DH	Micro Planet Maroc
Capteur d'humidité de sol capacitive V1.2	35 DH	Moussa soft Maroc
Carte Arduino UNO R3 + Cable USB	120 DH	Moussa soft Maroc
Bread Board	35 DH	Moussa soft Maroc
Fils de connexion	7 DH	Moussa soft Maroc
Led Rouge +Led Verte	2 DH	Moussa soft Maroc
Module Wifi ESP8266	70 DH	Moussa soft Maroc
Boite étanche à couvercle encastré 240*180*85	18 DH	Electro Delivery

Le composant	Le prix	Le site d'achats
Afficheur lcd 16*2	35 DH	Micro Planet Mroc
Relais JQC3F-5V DC-C Arduino	30 DH	A2itronic
Electrovanne 12 V DC	90 DH	Moussa soft Maroc
Diode de roue libre	3 DH	Moussa soft
Raccordement spécifique *2	20 DH	Bricoma à BENI MELLAL
Teflan	2 DH	Bricoma à BENI MELLAL
1 m Tuyau d'eau	10 DH	Bricoma à BENI MELLAL
Batterie ExpertPower 12V 2Ah	200 DH	Amazon
Panneau Solaire Renogy 20W 12V	300 DH	Amazon

Le prix totale	1010 DH

6.2. Comparaison avec l'arrosage manuel

1. Consommation d'Eau

Dimension du Terrain : 120m x 80m = 9600 m^2

Consommation d'Eau par m^2 (approximation pour l'irrigation) :

- **Normal** : 5 L/m^2/jour
- **Automatique** : 2.5 L/m^2/jour (réduction de 50% grâce à l'efficacité)

Consommation d'Eau Totale :

- **Normal** : 9600 m^2 x 5 L/m^2/jour x 365 jours = 17,520,000 L/an = 17,520 m^3/an
- **Automatique** : 9600 m^2 x 2.5 L/m^2/jour x 365 jours = 8,760,000 L/an = 8,760 m^3/an

Coût de l'Eau : 5 DH/m^3

Coût Annuel de l'Eau :

- **Normal** : 17,520 m^3 x 5 DH/m^3 = 87,600 DH/an
- **Automatique** : 8,760 m^3 x 5 DH/m^3 = 43,800 DH/an

❑ **Économie d'Eau :** 87,600 DH - 43,800 DH = 43,800 DH/an

2. Énergie

Système d'Arrosage Normal :

- **Consommation Électrique** : environ 1000 DH/an

Système Automatique :

- **Énergie Solaire** : 0 DH/an

❑ **Économie d'Énergie :** 1000 DH/an

6.2. Comparaison avec l'arrosage manuel

3. Main-d'œuvre

Système d'Arrosage Normal :

- **Coût de Main-d'œuvre** : environ 2000 DH/an

Système Automatique :

- **Automatisation** : 0 DH/an

❑ **Économie de Main-d'œuvre :** 2000 DH/an

4. Gazon Hybride

Gazon Hybride :

- **Réduction de la consommation d'eau de 30%**
- **Consommation d'Eau** : 8,760 m^3/an x 0.7 = 6,132 m^3/an
- **Coût Annuel de l'Eau avec Gazon Hybride** : 6,132 m^3 x 5 DH/m^3 = 30,660 DH/an

❑ **Économie d'Eau avec Gazon Hybride :** 43,800 DH - 30,660 DH = 13,140 DH/an

5. Coût Initial du Système Automatique :

Prix total : 1010DH

6.Résumé des Calculs

✓ **Économies Annuelles Totales :**

- **Énergie** : 1000 DH/an
- **Main-d'œuvre** : 2000 DH/an
- **Eau** : 43,800 DH - 30,660 DH = 13,140 DH/an
- **Total** : 1000 + 2000 + 13,140 = 16,140 DH/an

✓ **Coût Initial du Système Automatique :** 1010 DH

✓ **Retour sur Investissement Rapide (ROI) :** 1010 DH / 16,140 DH ≈ 0.06 ans (moins de 1 mois)

6.3 Conclusion

Comparaison des Coûts Annuels (DH)

Aspect	Automatique	Normal	Économies
Énergie	0	1000	1000
Main-d'œuvre	0	2000	2000
Eau	30 660	87 600	56,940
Total	30,660	90,600	59,940

Efficacité économique=(Côuts normaux /Économies réalisés)×100

Efficacité économique=(90 600 /59 940)×100

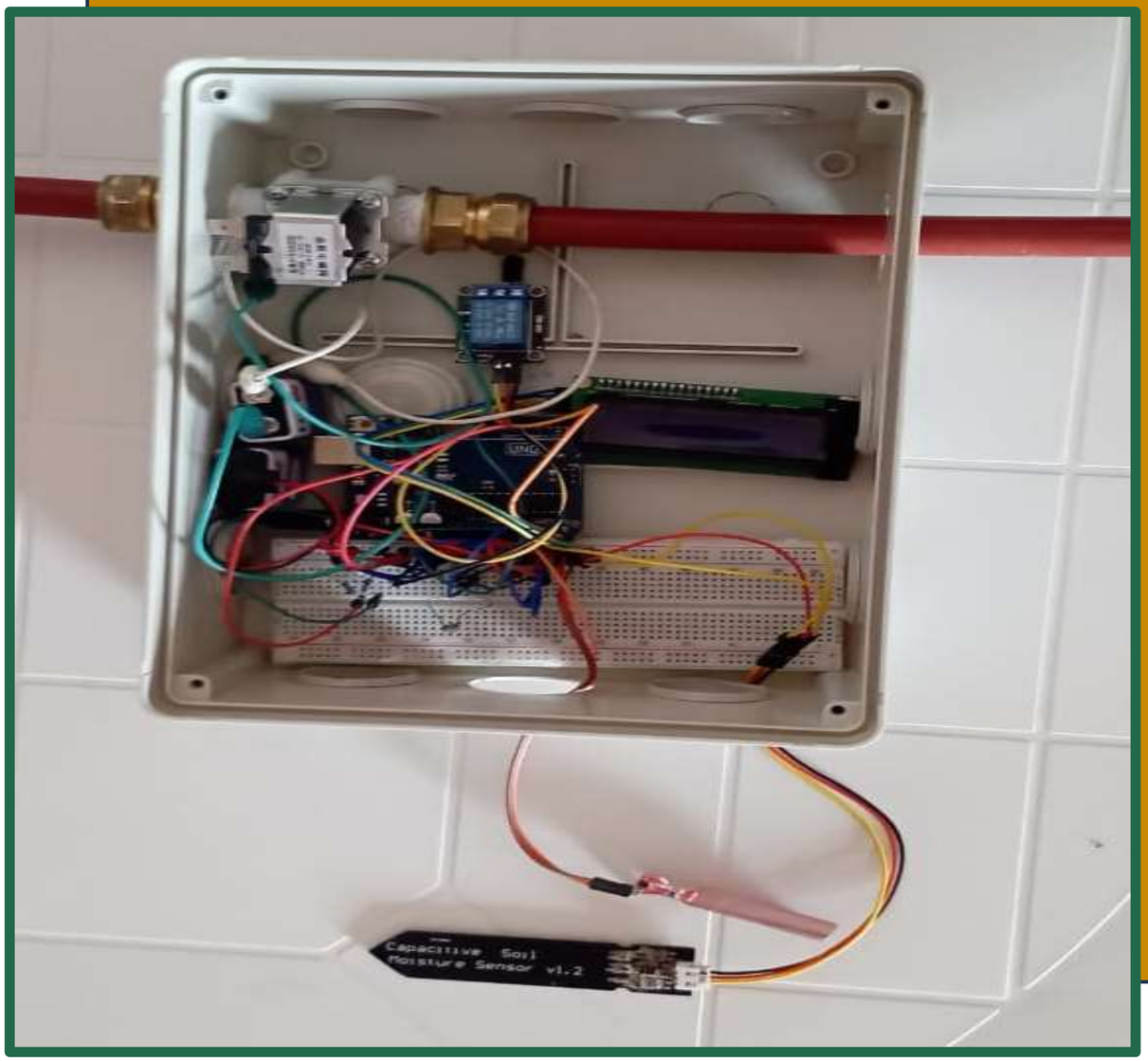
Efficacité économique≈ 66.16%

66.16%

Adopter notre système, c'est choisir l'efficacité et la durabilité pour un avenir plus vert et économique !!!!!

Ghanam
SMART WATERING

Objectif 7: Le profil final de notre système.



Conclusion

Compatible ✓

Automatique ✓

simple ✓

Ghanam
SMART WATERING

Autonomie énergétique ✓

Abordable ✓

Ecologique ✓

Commande à distance ✗

Perspectives d'Amélioration

Annexe 1:

test_capteur_humidite_sol_capacitif.ino

```
//Test de capteur d'humidite de sol capacitif V1.2
// Cree par Fatima-Ezzahra GHNAM ,N°CNC :BM030T
 void setup() {
  Serial.begin(9600); // Initialisation de la communication série
}

void loop() {
  int sensorValue = analogRead(A0); // Lecture de la valeur analogique du capteur
  float percentageHumidity = map(sensorValue, 0, 1023, 0, 100); // Conversion en pourcentage

  Serial.print("Humidité du sol : ");
  Serial.print(percentageHumidity);
  Serial.println("%");

  delay(30000); // Attente d'une 30 seconde avant la prochaine lecture
}
```

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

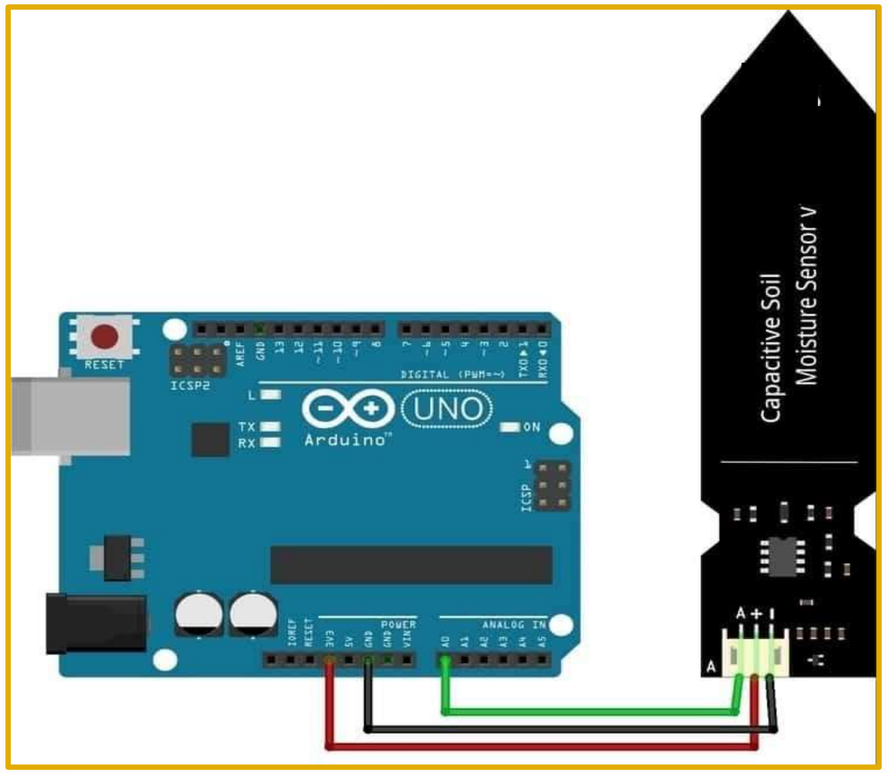


Figure 15 : Schéma de montage du circuit

Annexe 2:

Test_capteur_temperature_LM35.ino

```
// Programme pour tester le capteur de température LM35
// Cree par Fatima_Ezzahra GHANAM ,N°CNC:BM030T
const int sensorPin = A0; // Broche du capteur LM35

float temperature; // Variable pour la température

void setup() {
  Serial.begin(9600); // Initialiser la communication série
}

void loop() {
  int sensorValue = analogRead(sensorPin); // Lire la valeur analogique
  float voltage = sensorValue * (5.0 / 1023.0); // Convertir en tension
  temperature = voltage * 100; // Convertir en température

  Serial.print("Température: "); // Afficher la température
  Serial.print(temperature);
  Serial.println(" °C");

  delay(1000); // Attendre 1 seconde
}
```

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

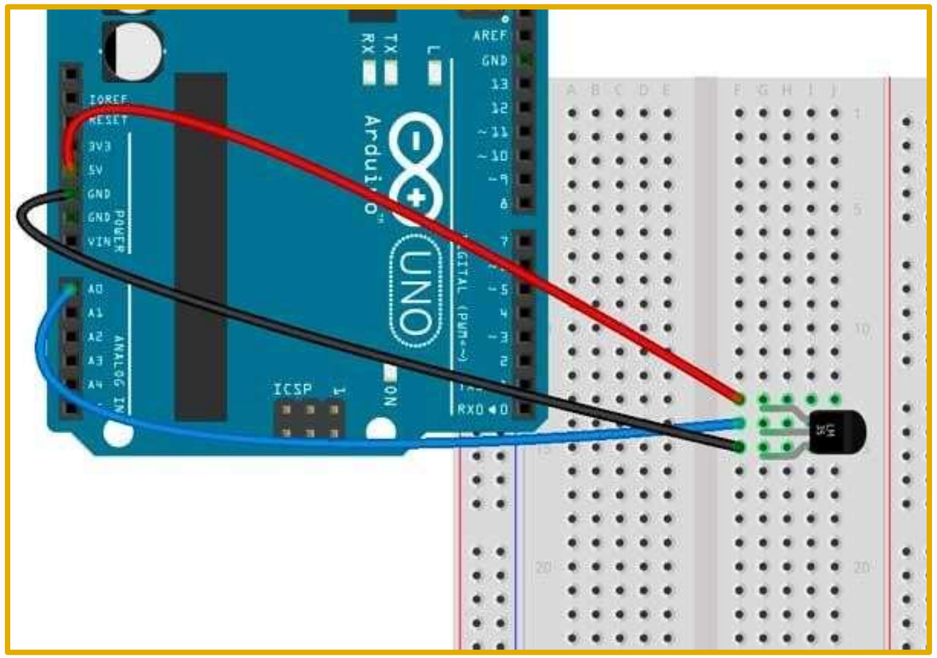


Figure 17 : Schéma de montage du circuit

Annexe 3:

programme_capteur_de_pluie.ino

```
// Performance du capteur de pluie
// Programme écrit par Fatima-ezzahra GHANAM
const int waterSens = A0; // Définir la broche du capteur de pluie
const int led = 9; // Définir la broche de la LED (pin 9)
int waterVal; // Définir la variable pour la valeur du capteur de pluie

void setup() {
  pinMode(led, OUTPUT); // Configurer la LED comme sortie
  pinMode(waterSens, INPUT); // Configurer le capteur de pluie comme entrée
  Serial.begin(9600); // Démarrer la communication série à 9600 bauds
}

void loop() {
  waterVal = analogRead(waterSens); // Lire la valeur du capteur de pluie
  Serial.print("Niveau d'eau : "); // Afficher le texte "Niveau d'eau :"
  Serial.println(waterVal); // Afficher la valeur du capteur de pluie sur le moniteur série

  if (waterVal <= 350) { // Si la valeur est inférieure ou égale à 350
    Serial.println("Il ne pleut pas"); // Afficher "Il ne pleut pas" sur le moniteur série
    digitalWrite(led, HIGH); // Allumer la LED
  } else { // Sinon
    Serial.println("Il pleut"); // Afficher "Il pleut" sur le moniteur série
    digitalWrite(led, LOW); // Éteindre la LED
  }
  delay(30000); // Attendre 30 secondes avant la prochaine lecture
}
```

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

Annexe 4:

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

```python
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()
```

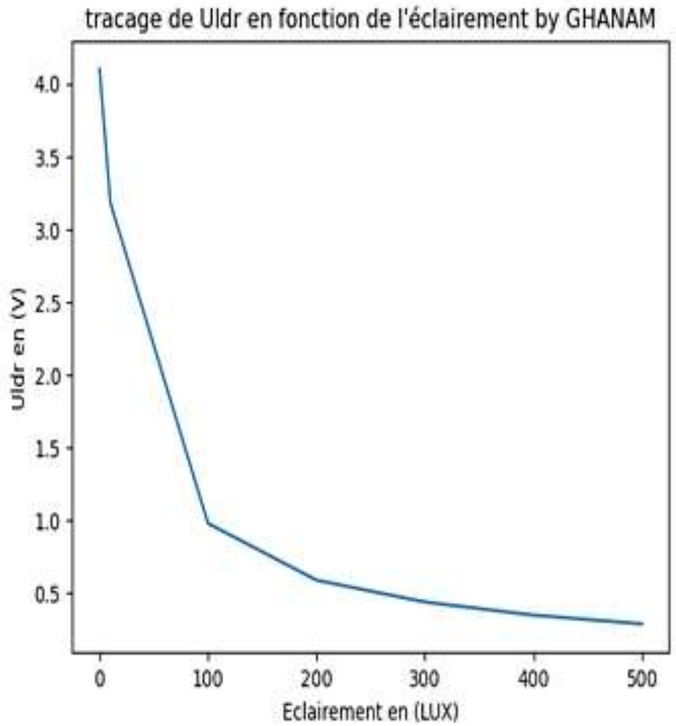


Figure 19 : Programme Python pour le traçage de U_{LDR} en fonction de l'éclairement

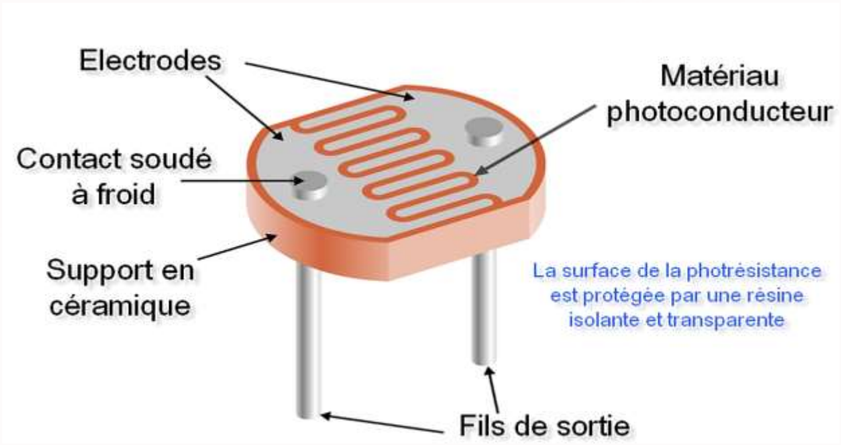


Figure 20 : La composition du capteur de luminosité (LDR)

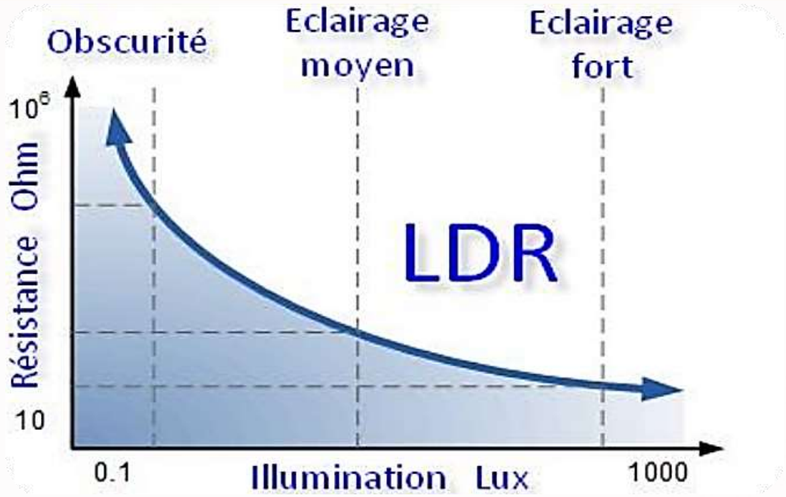


Figure 21 : La variation de résistance en fonction de l'éclairement pour le capteur (LDR)

Annexe 4:

programme_luminosite_capteur_LDR.ino

```
//Programme pour mesurer la luminosité avec un capteur LDR
//Crée par Fatima-Ezzahra GHANAM N°CNC: BM030T

const int ldrPin = A0; // Broche analogique à la quelle est connecté le capteur LDR

void setup() {
  Serial.begin(9600); //Initialise la communication série
}

void loop() {
  int lecteurLDR = analogRead(ldrPin);//Lecture de la valeur du capteur
  Serial.print("Luminosite : ");
  Serial.println(lecteurLDR);

  delay(1000);//Attente d'une seconde avant de prendre une nouvelle mesure
}
```

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

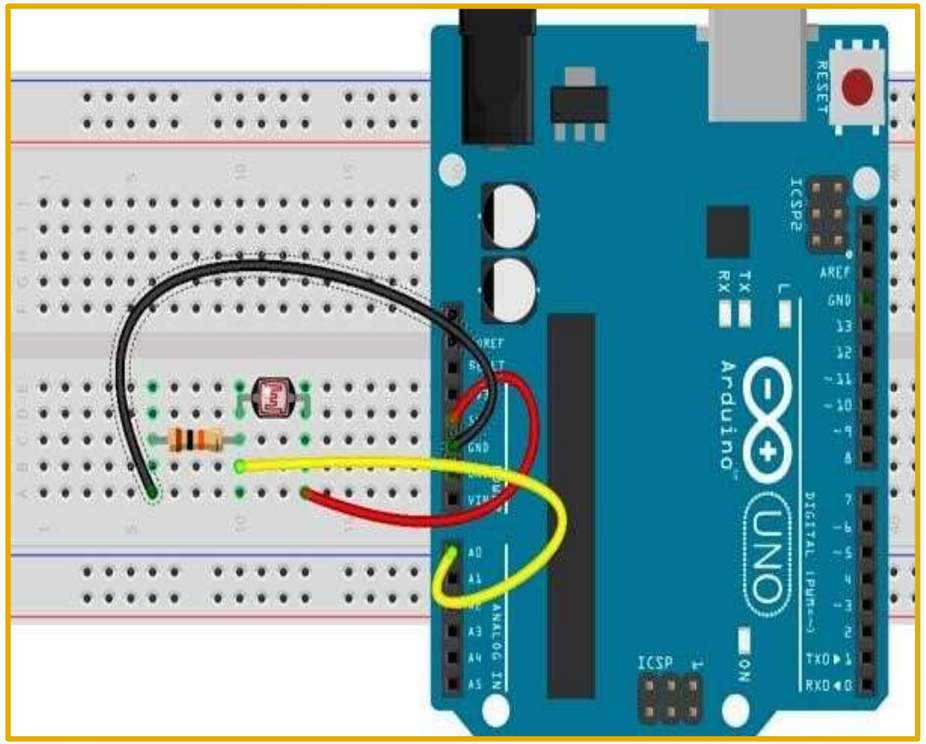


Figure 23 : Schéma de montage du circuit

Annexe 5 :

systeme_arrosage_automatique_terrain_foot_9.ino

```
// Système d'arrosage automatique de terrain de foot.
// Créé par Fatima_Ezzahra GHANAM N°CNC: BM030T
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Initialisation de l'afficheur LCD I2C (adresse, colonnes, lignes)
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Broche de données pour le capteur d'humidité capacitif
const int humiditePin = A3;

// Broche de données pour le capteur de température LM35
const int temperaturePin = A1;

// Broche de données pour le capteur de luminosité LDR
const int luminositePin = A0;

// Broche de contrôle pour le relais de l'électrovanne
const int relaisPin = 8;

// Broche de données pour le capteur de pluie
const int pluiePin = A2;

// Broches pour les LED
const int ledVertePin = 9;
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 :

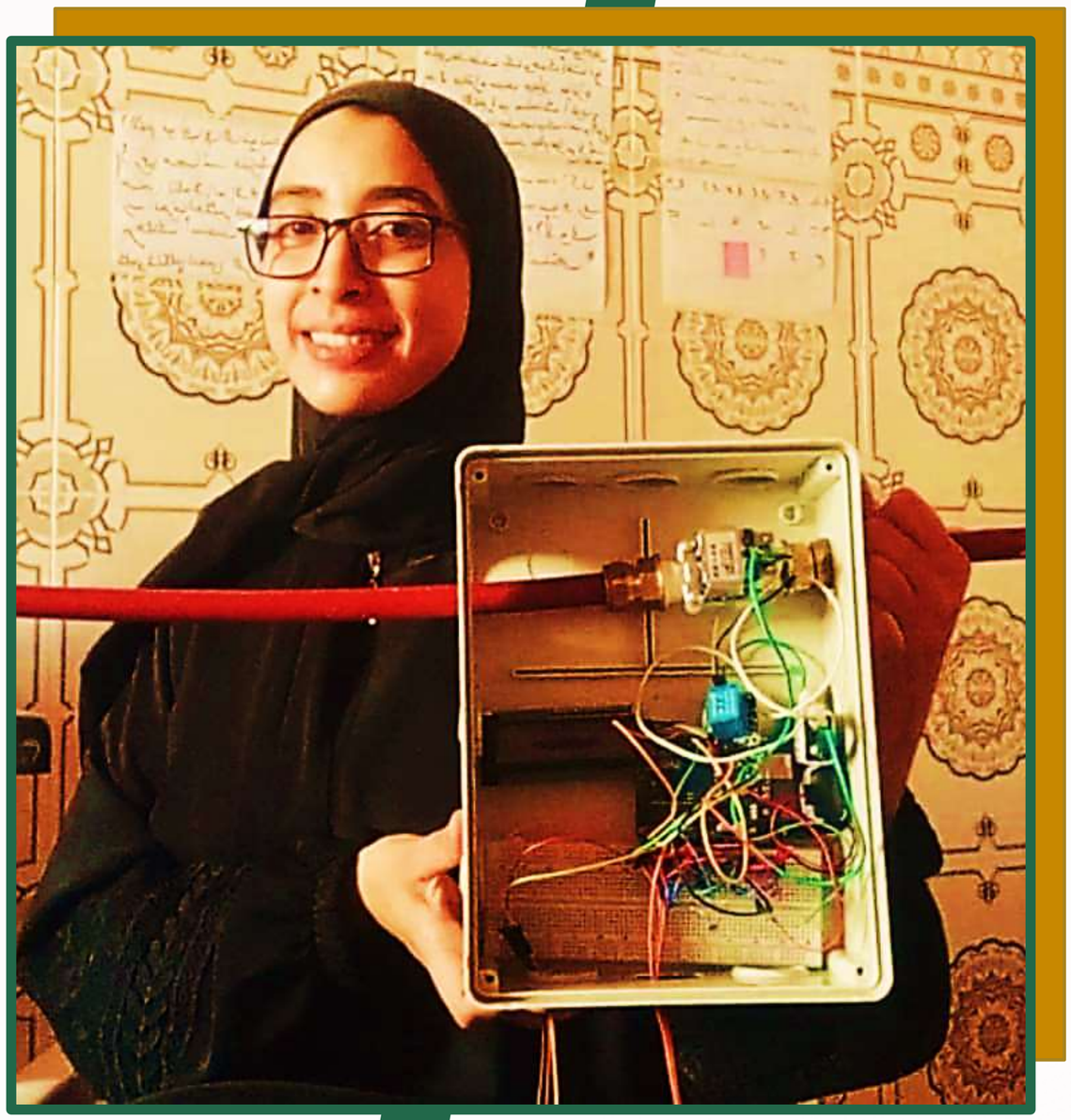
```
133
134     // Affichage des différentes causes de désactivation
135     if (pluie >= 350) {
136       lcd.clear();
137       lcd.setCursor(0, 0);
138       lcd.print("Cause 1:");
139       lcd.setCursor(0, 1);
140       lcd.print("Il pleut");
141       delay(10000); // Attendre 10 secondes
142     }
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).**

Ghanam
SMART WATERING

Merci pour vous



Nom:GHANAM
Prénom:Fatima_Ezzahra
Numero CNC: BM030T