

PROJET DE FIN D'ÉTUDES

# Smart Garden

Entreprise d'accueil : Ecole Nationale d'Ingénieurs de Tunis (ENIT)

Réalisé par : Nour TRABELSI  
Ahmed Aziz GADDOUR

Encadré par :  
Encadreur ENIT : M. Férid KOURDA  
Encadreur ISET : M. Faouzi FAZZENI

# Plan

## POINTS À DISCUTER :

- Cadre de projet
- Choix du matériel et logiciels
- Présentation du travail réalisé
  - La Partie Hardware
  - La Partie Software
- Conclusion

# Présentation De Projet

Dans le département Génie électrique à l'ENIT est situé notre jardin intelligent. « Smart Garden », complément du projet « Smart House »

A pour but d'offrir un endroit de travail agréable en plein air.

Projet PAQ Promesse Collabora financé par la banque mondiale. Intitulé : « Smart House » sous la responsabilité de M. Le professeur Férid KOURDA.

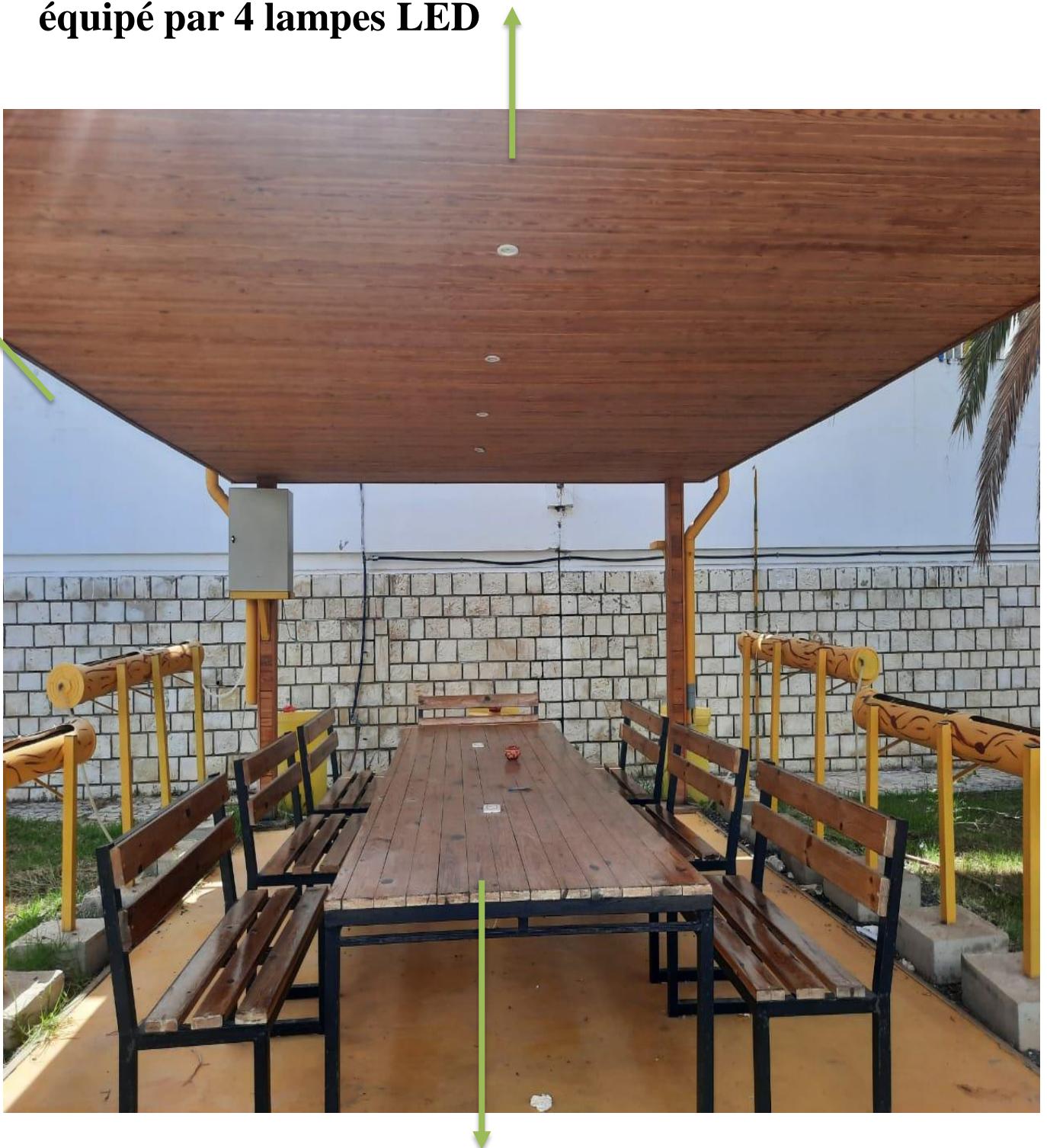


# Mise en situation

L'espace contient :

une grande table avec des bancs fixes en bois situés sous un toit équipé par 4 lampes LED

Sur le toit,  
un panneau  
voltaïque  
est installé

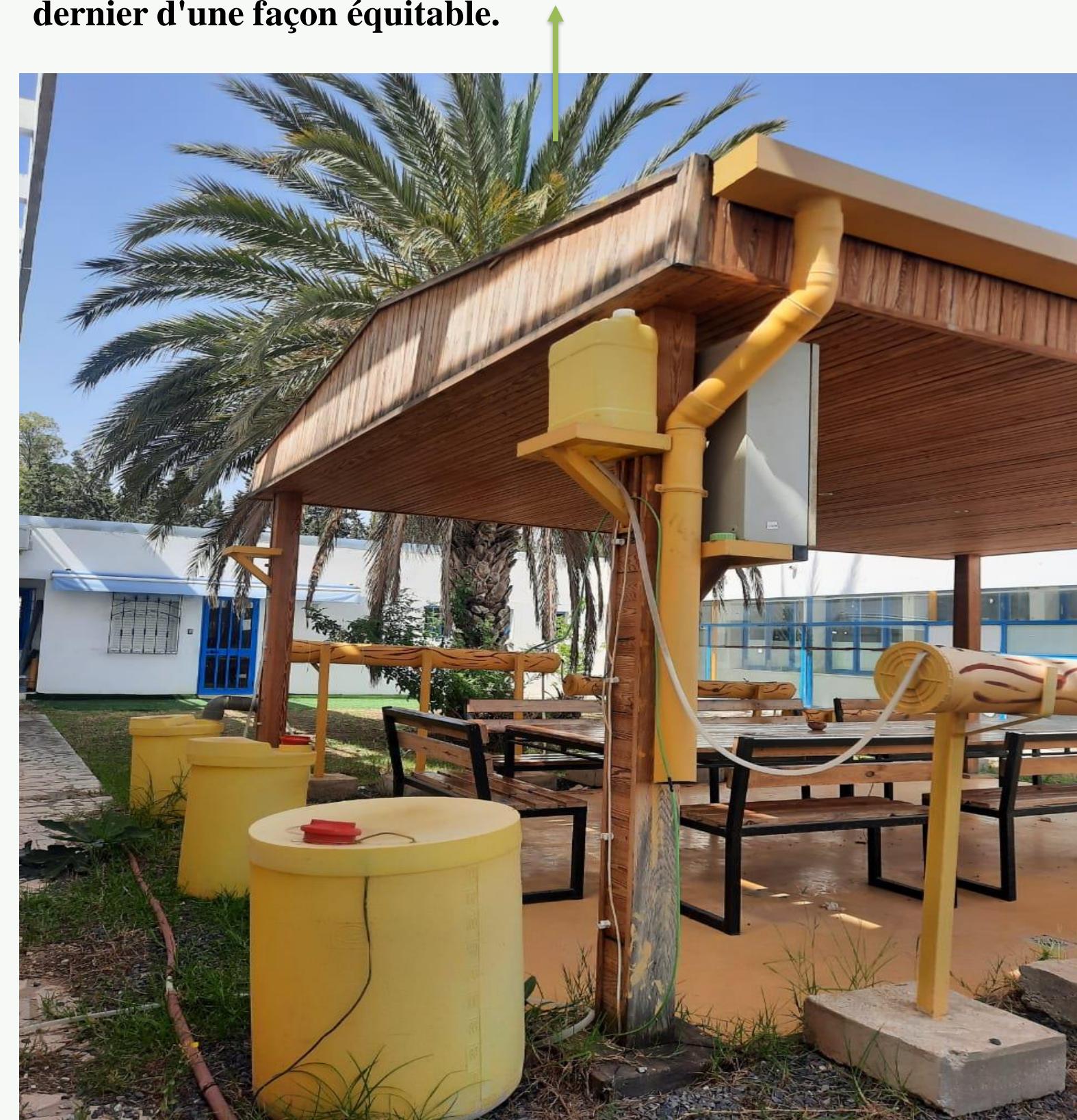


Sur la table, 2 boutons poussoirs sont fixés.

A côté de ces boutons, nous avons prévu la mise en place de deux ports universels

Le fonctionnement de l'arrosage est réalisé à l'aide des tubes liés à des bidons de 20 L qui rassemblent l'eau de pluie.

La répartition de l'eau se fait par des trous situés tout au long de ce dernier d'une façon équitable.



# Problématiques

- 01 Le problème d'alimentation électrique
- 02 Le problème d'arrosage
- 03 Le problème d'éclairage
- 04 Le problème de supervision et de commande à distance

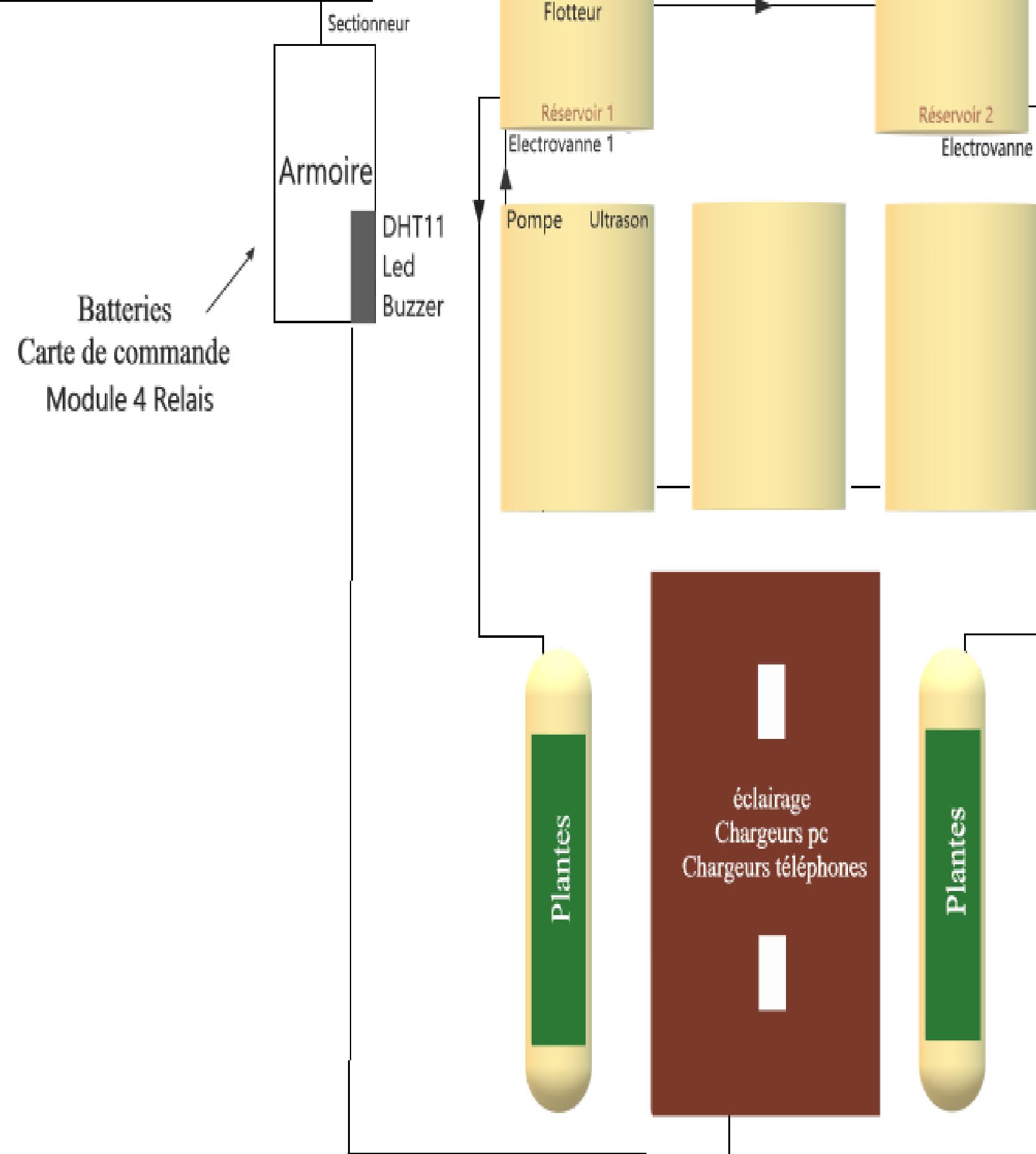
## Les étapes du projet

- 01 La charge de batterie par panneaux photovoltaïque avec une BMS (battery management system).**
- 02 L'éclairage de l'espace de travail qui sera muni d'une minuterie.**
- 03 La charge des téléphones et PC par batteries.**
- 04 La gestion de la récolte d'eau en vue de l'arrosage automatique durant toute l'année du jardin aménagé.**
- 05 Réalisation d'une carte de commande à base d'un microcontrôleur ESP32 afin d'assurer la communication à distance WIFI en vue de :**
  - La récolte des données**
  - La commande à distance des actionneurs**
  - La visualisation de l'historique des données**

# Choix du matériel et logiciels



panneau voltaïque



# Choix du matériel et logiciels

Notre système du jardin intelligent est divisé en 4 parties :

- **Alimentation du système** : Des batteries alimentées par des panneaux photovoltaïques pour fournir l'énergie électrique à notre système.
- **Armoire de commande** : l'emplacement de la carte commande gérant l'irrigation automatique et l'éclairage de l'espace et le BMS.
- **Cartes des chargeurs** : Pour adapter la tension d'alimentation de notre système (tensions des batteries +12/+24) à la tension des chargeurs PCS (+20V) et des chargeurs des téléphones portables (+5V).
- **Application** : Superviser le système et commander l'éclairage et l'irrigation.

# Choix du matériel

## Périphériques d'alimentation

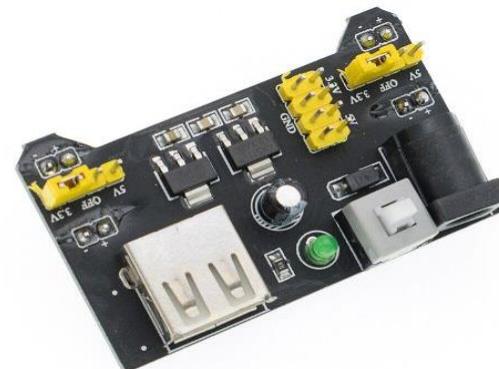
Panneaux Photovoltaïque



Batteries Plomb-Acide



Module d'alimentation 3,3V/5V



## Les Cartes électroniques

Chargeur PC régulateur 20V + Circuit de la protection contre la haute température.



Chargeur Téléphone régulateur 5V .

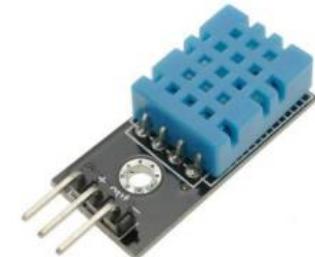


Carte de Commande



## Les capteurs

Capteur de température et humidité ambiante



Capteur de niveau (capteur à ultrasons)

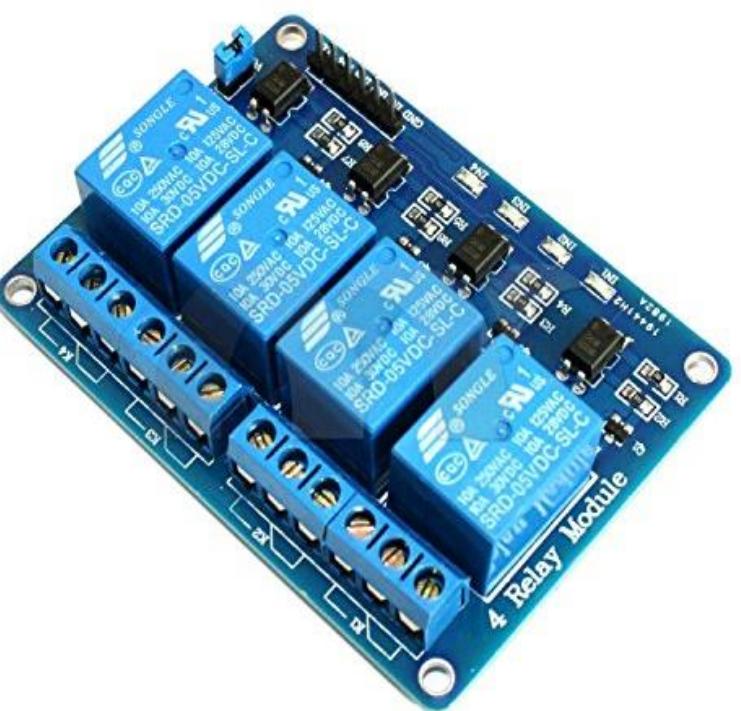


Détecteur de niveau (Flotteur)



## Les pré-actionneurs

Module relais



## Les actionneurs

Les Lampes



Mini pompe d'eau



Électrovanne motorisée



# L'unité de commande

ESP WROOM-32D

C'est un (Soc) system on a chip,  
doté de communications sans fil Wifi et Bluetooth.



# Les Logiciels

01

**Proteus ISIS**



02

**Arduino IDE**



03

**App Inventor**



04

**SolidWorks**



05

**Firebase**



# Bilan du projet

## Calcul énergétique journalière

Composant	Référence	Quantité	I(mA)	V(v)	P(W)
Microcontrôleur	ESP32	1	260	3,3	0.858
Capteur de température	DHT11	1	2,5	3,3 - 5	1.25
Capteur Ultrason	HOSR04	1	=2	3,3 – 5	0,01
Capteur Flotteur	X	1	X	3.3	0.001
Pompe	X	1	370	12	4,44
Module de relais 4 canaux	Relais	4	320	5	1,85
Servo moteur	MG995	2	400	4.8 - 6	2
Lampes 1	X	2	1200	12	14,4
Lampes 2	X	2	1200	12	14,4

- Les ordinateurs portables consomment classiquement des puissances électriques **de 50 à 100 W**
- le chargeur du téléphone consomme une puissance de **10 W**
- Donc énergie Total est : **Ej = 1143 w/j**

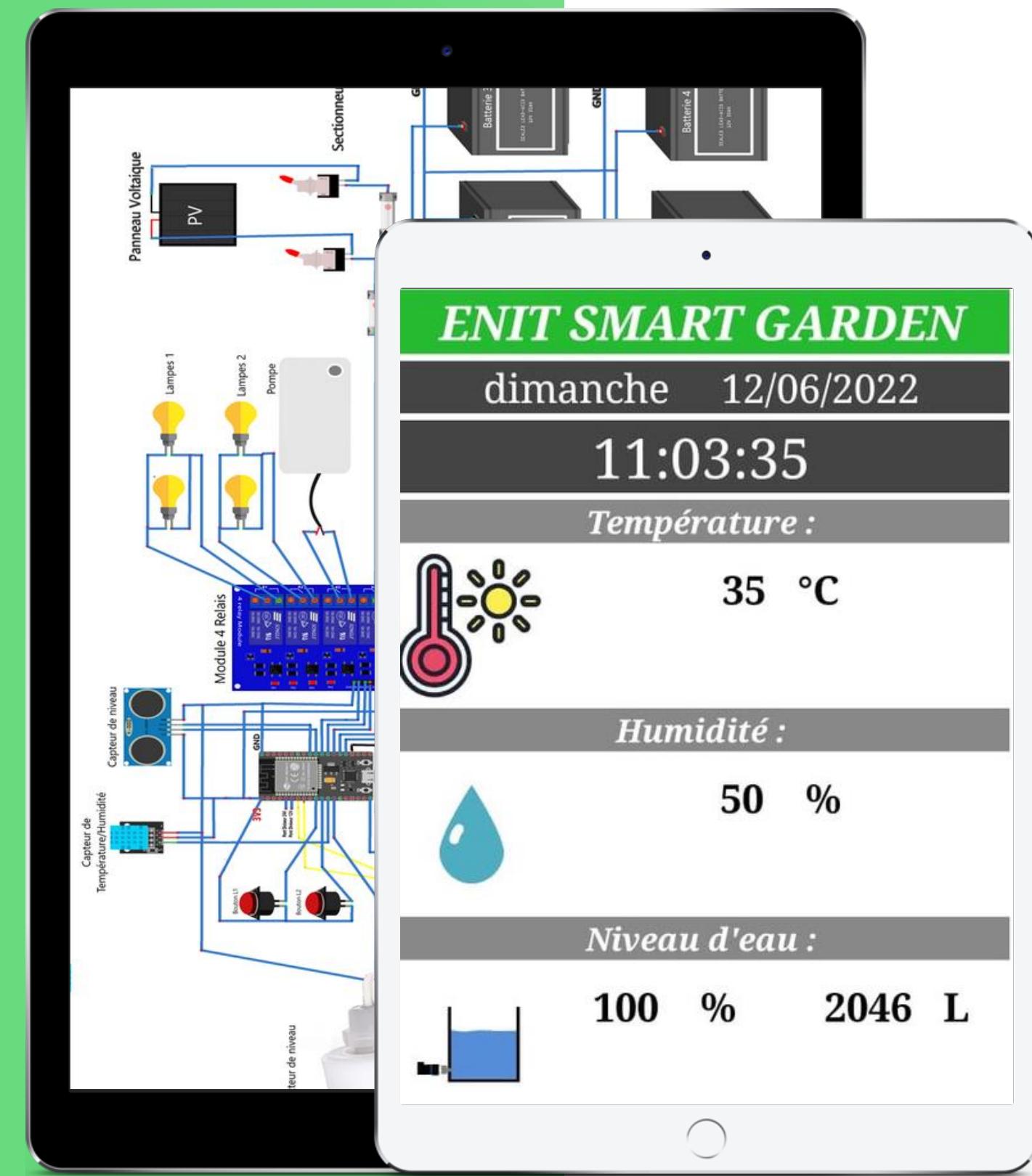
# Bilan économique

## Cout approximatif

- Les cartes réalisées ne sont pas disponibles dans les marchés tunisiens mais le plus ajouté par rapport les cartes prêtées en ligne dans les sites internationaux est la sécurité de circuit optimale.**
- Notre solution d'electrovanne motorisée du servomoteur est plus fiable que l'electrovanne prêtée et coute moins cher.**
- Le cout total de projet est : 1400D**

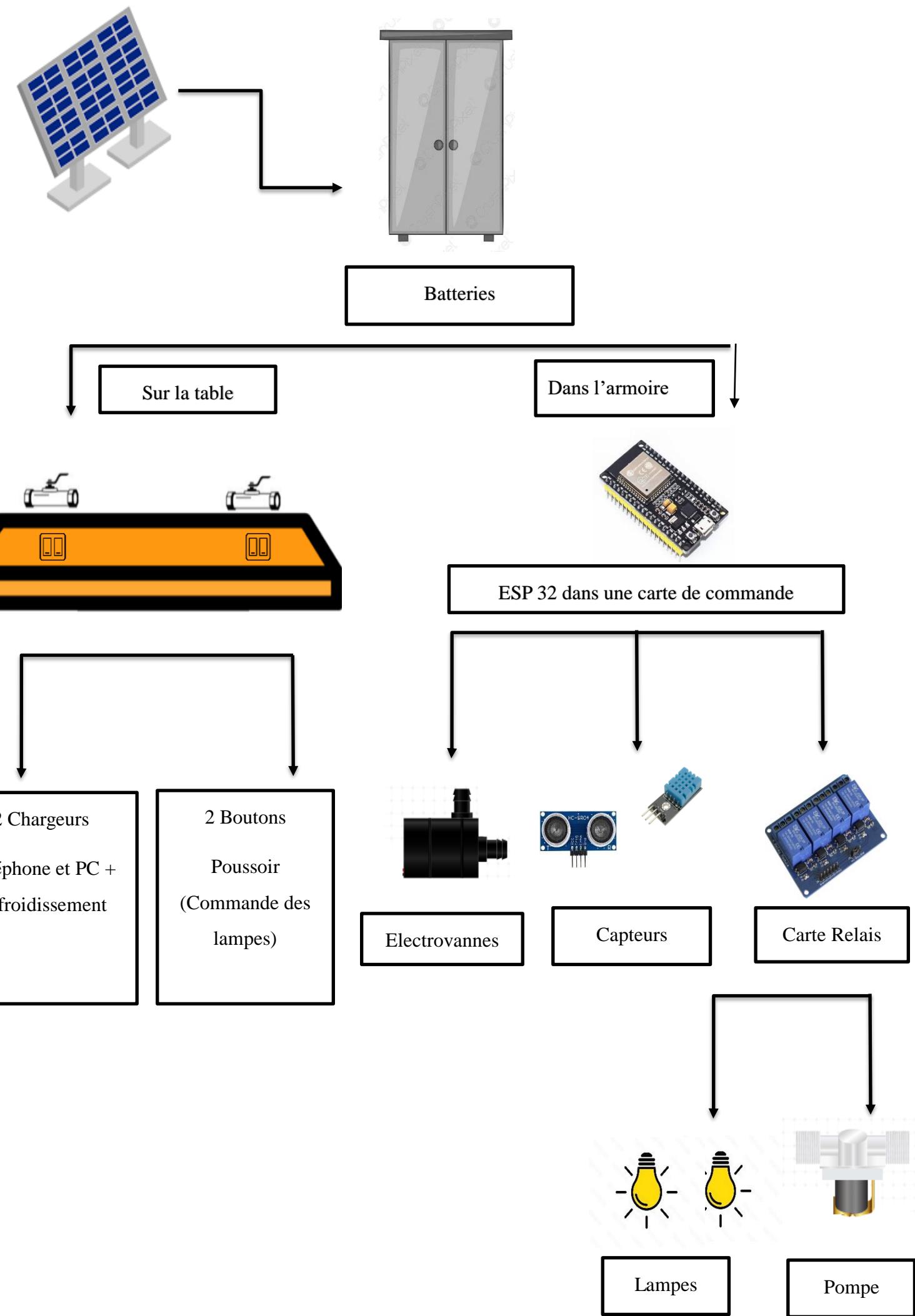
	Les objets	Cout	Quantité
Carte Electronique	Chargeur 5V	13D	2
	Chargeur 20V + Fan	24D	2
	Carte commande	X	1
Matérielles électroniques	Panneau	300D	2
	Batteries	70D	4
	Lampes	15D	4
	Relais	18D	1
	Microcontrôleur	45D	1
	Capteur ultrason	10D	1
	Capteur température	9D	1
	Capteur flotteur	20D	1
	Pompe	80D	1
	Servomoteur	40D	2
Matérielles	Les tuyaux, sert-câble	80D	X
	Tube PVC, Vannes		

# Présentation de la solution proposée



## La partie Hardware

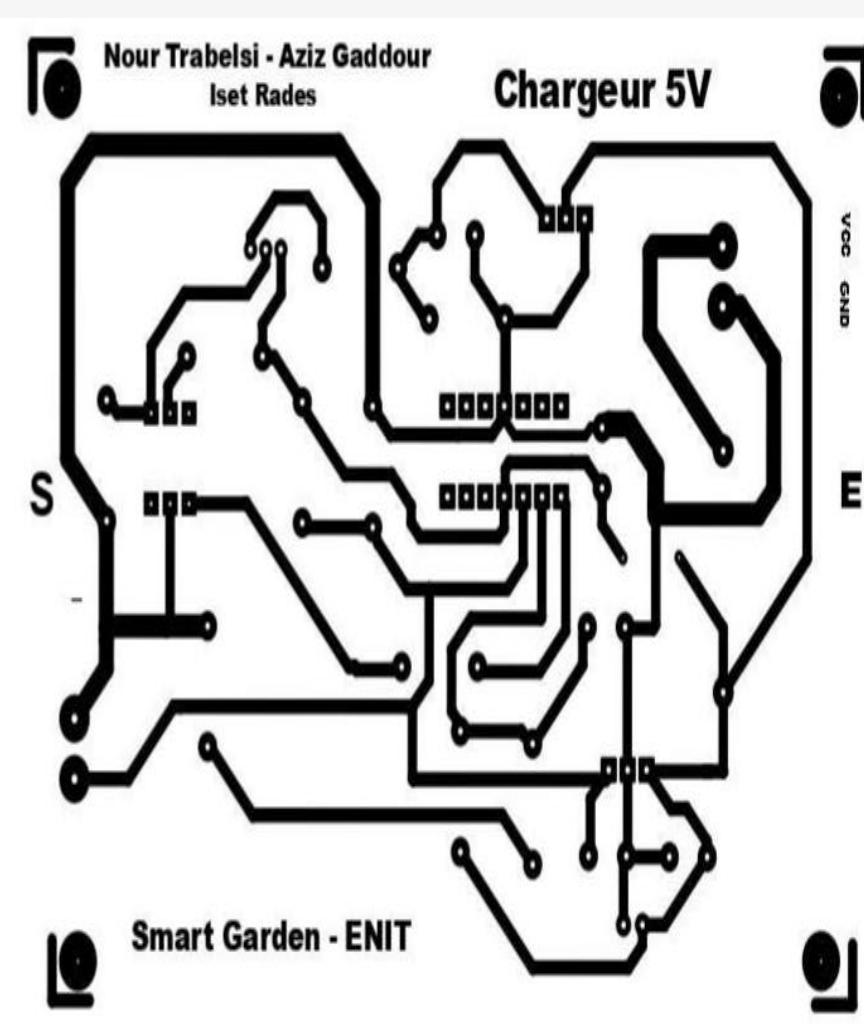
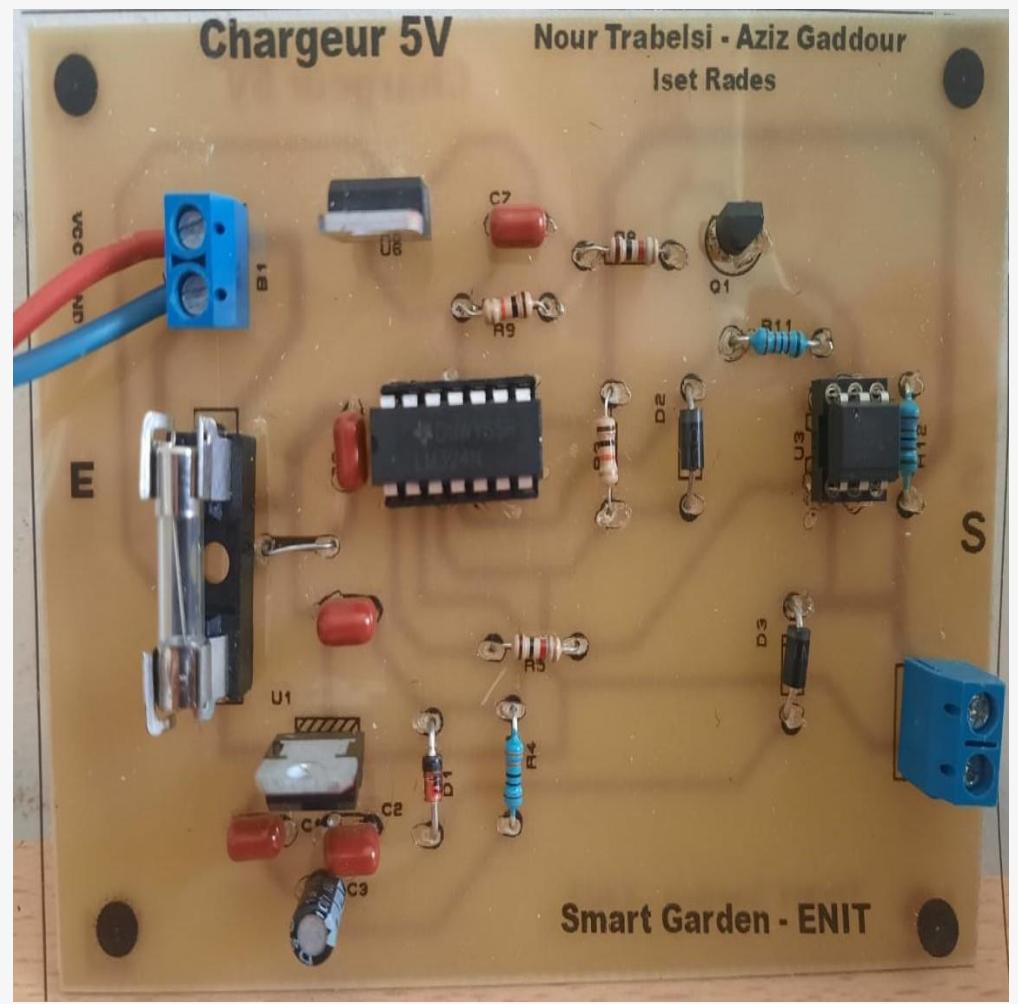
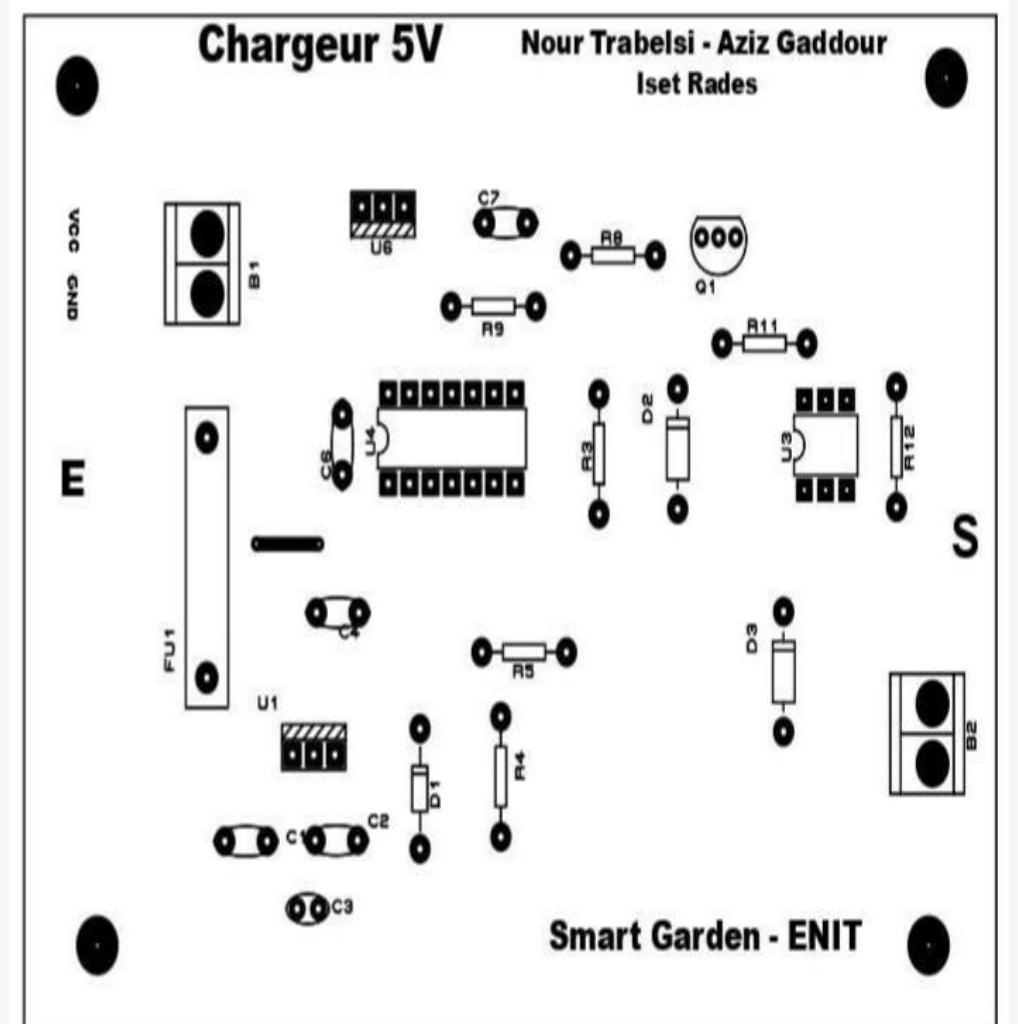
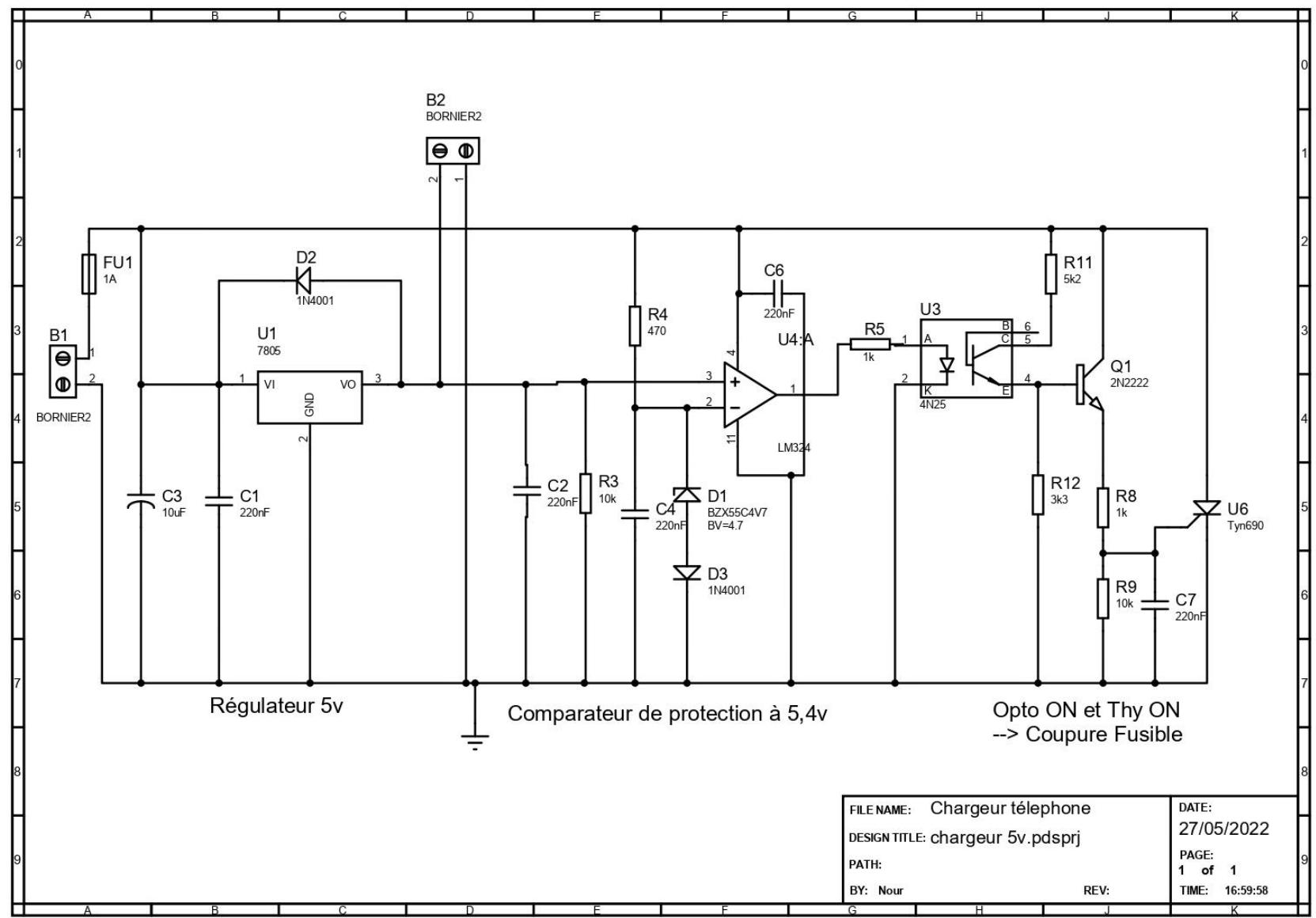
## La partie Software



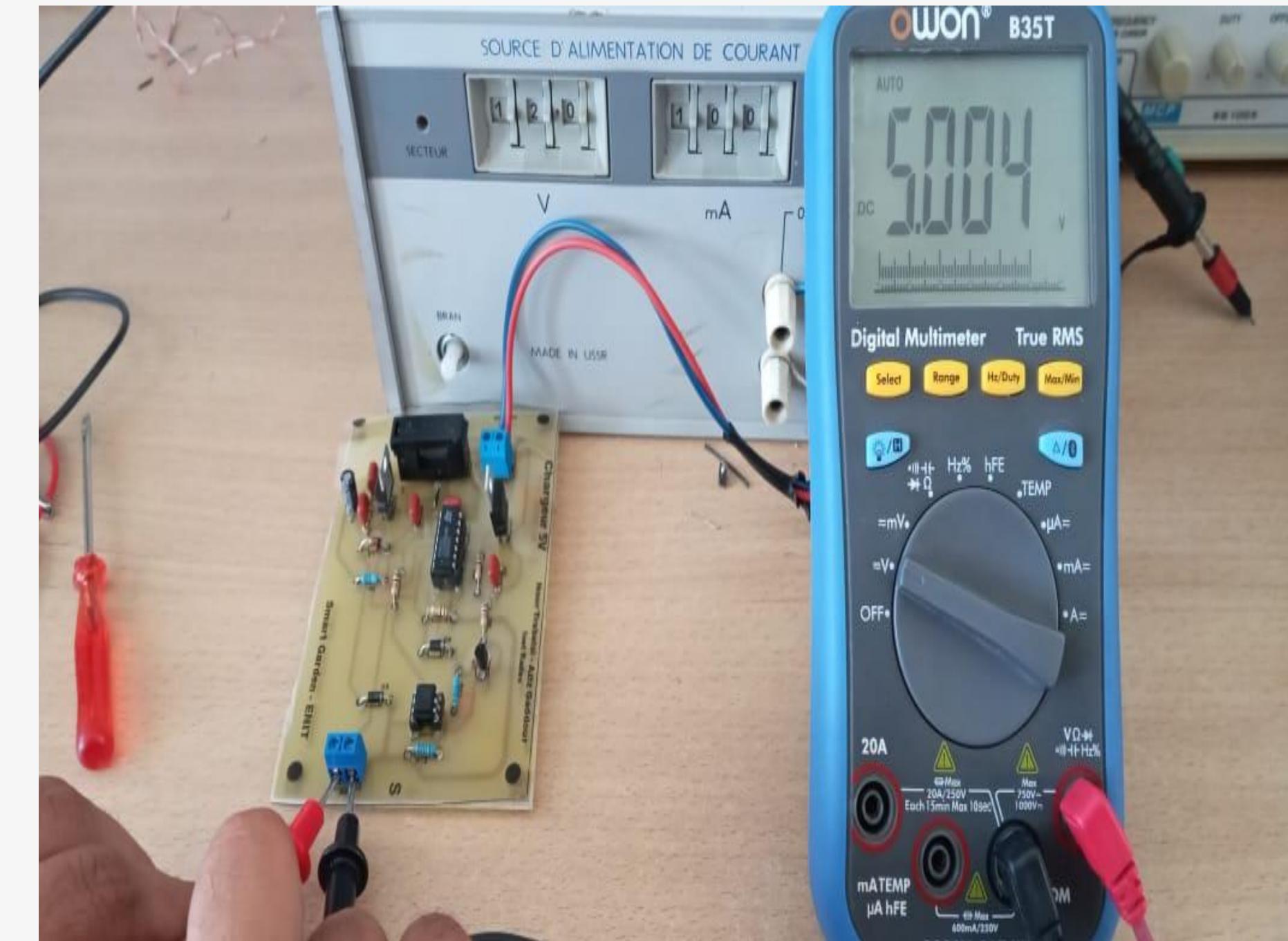
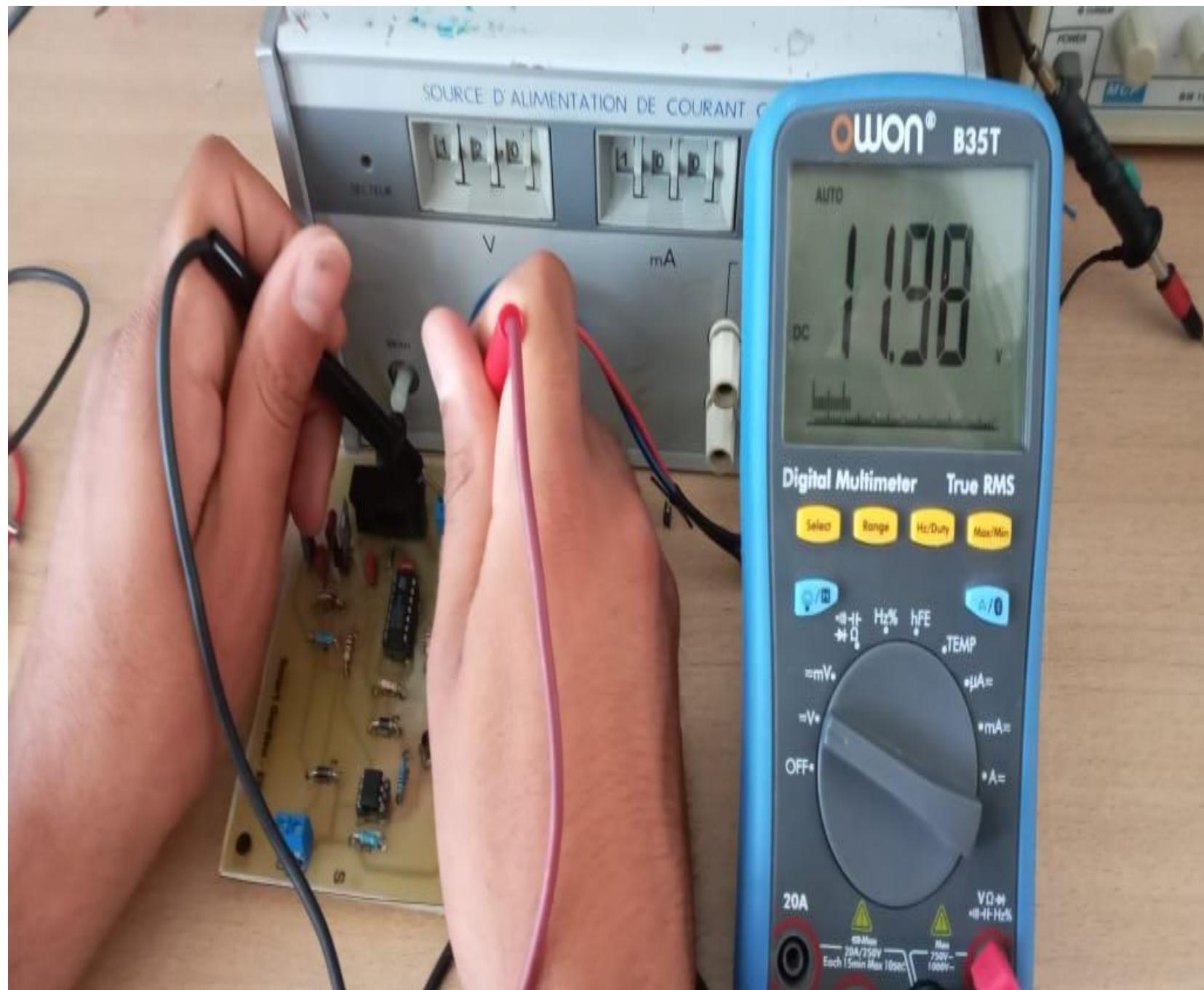
# La partie hardware

## Conception des cartes électroniques

- Conception de la carte chargeur 5V



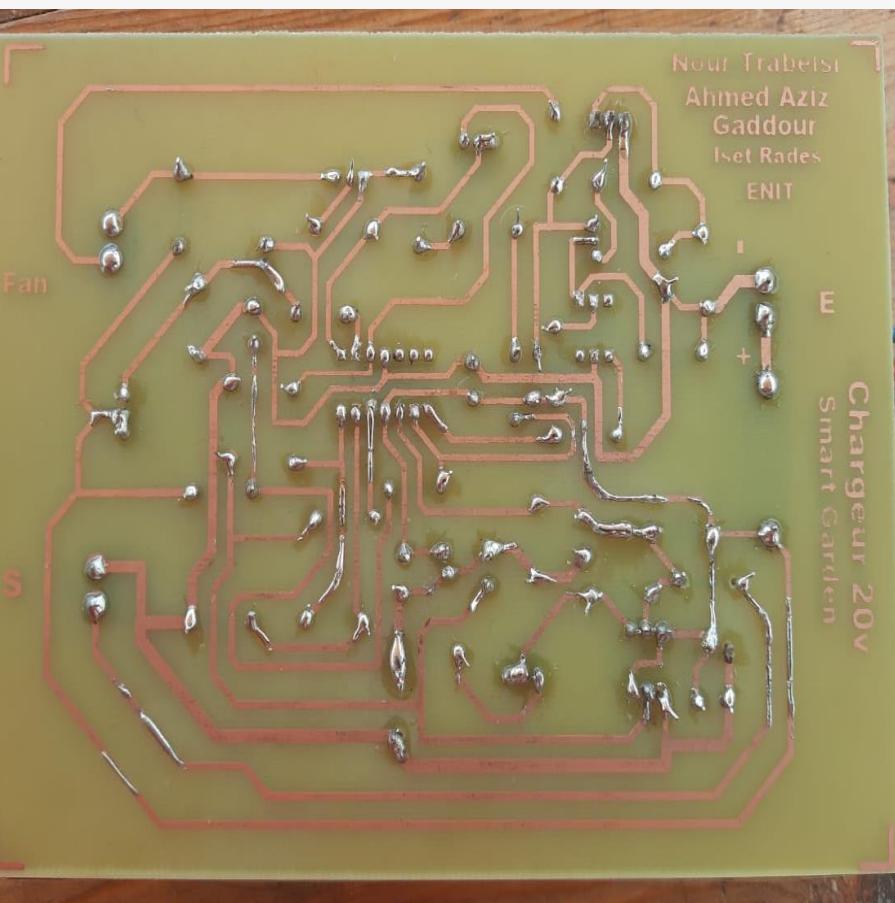
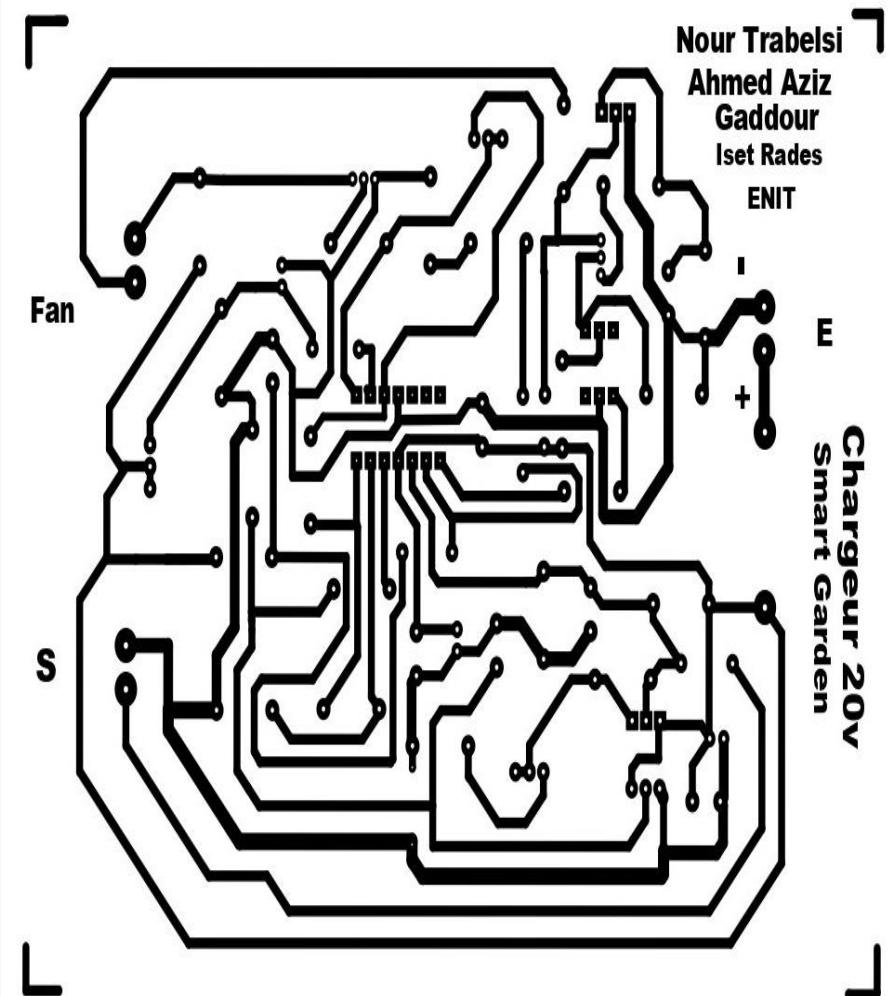
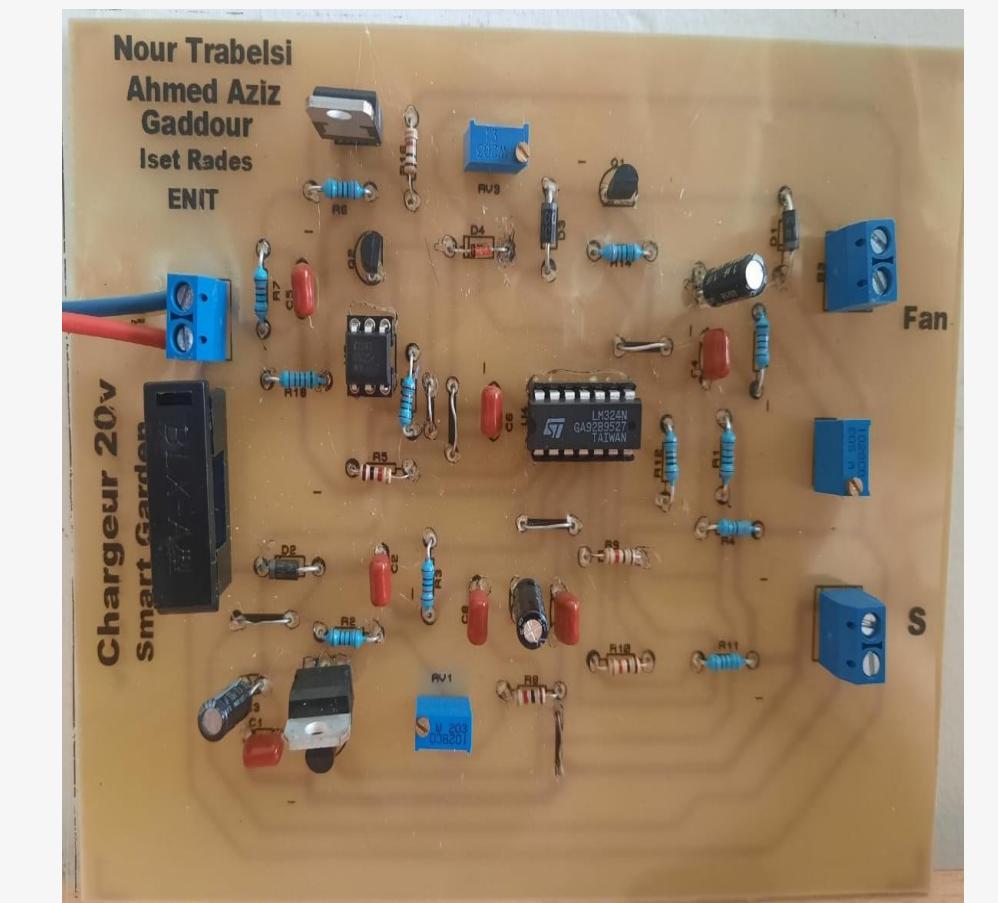
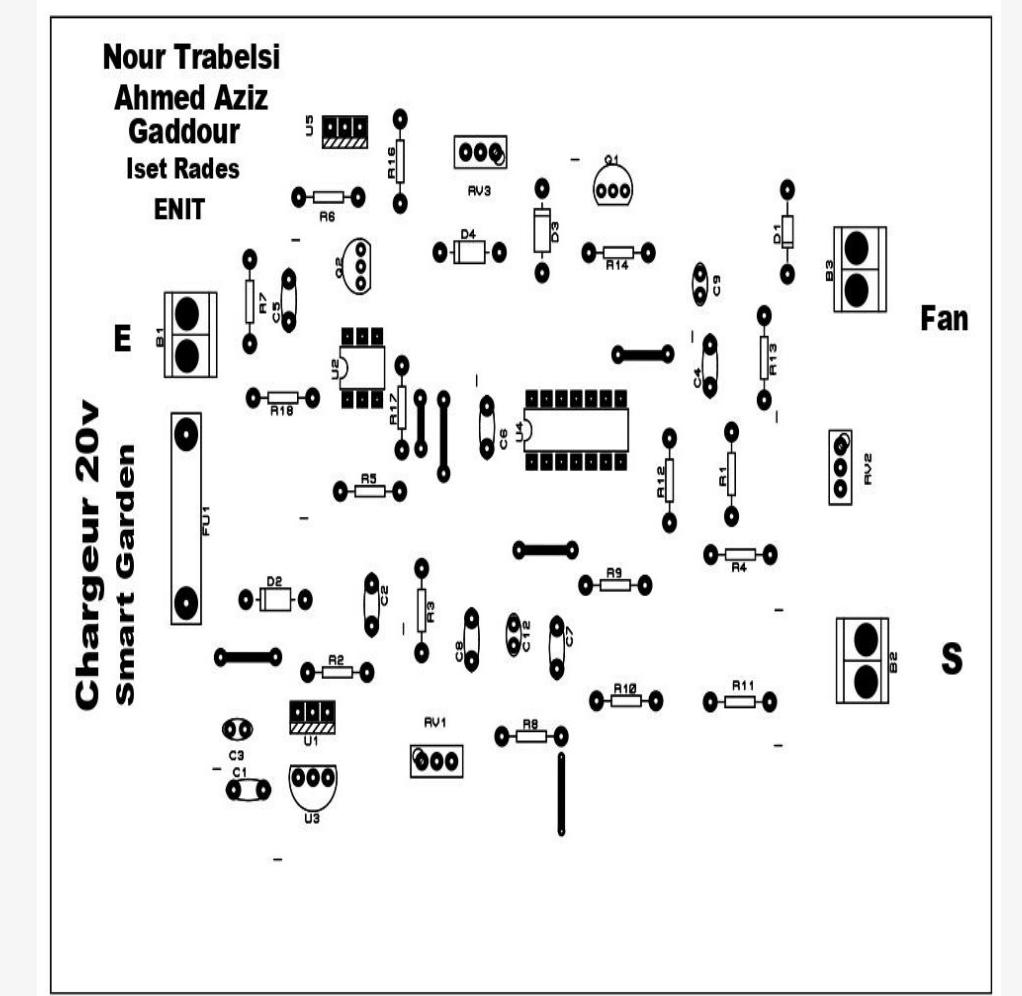
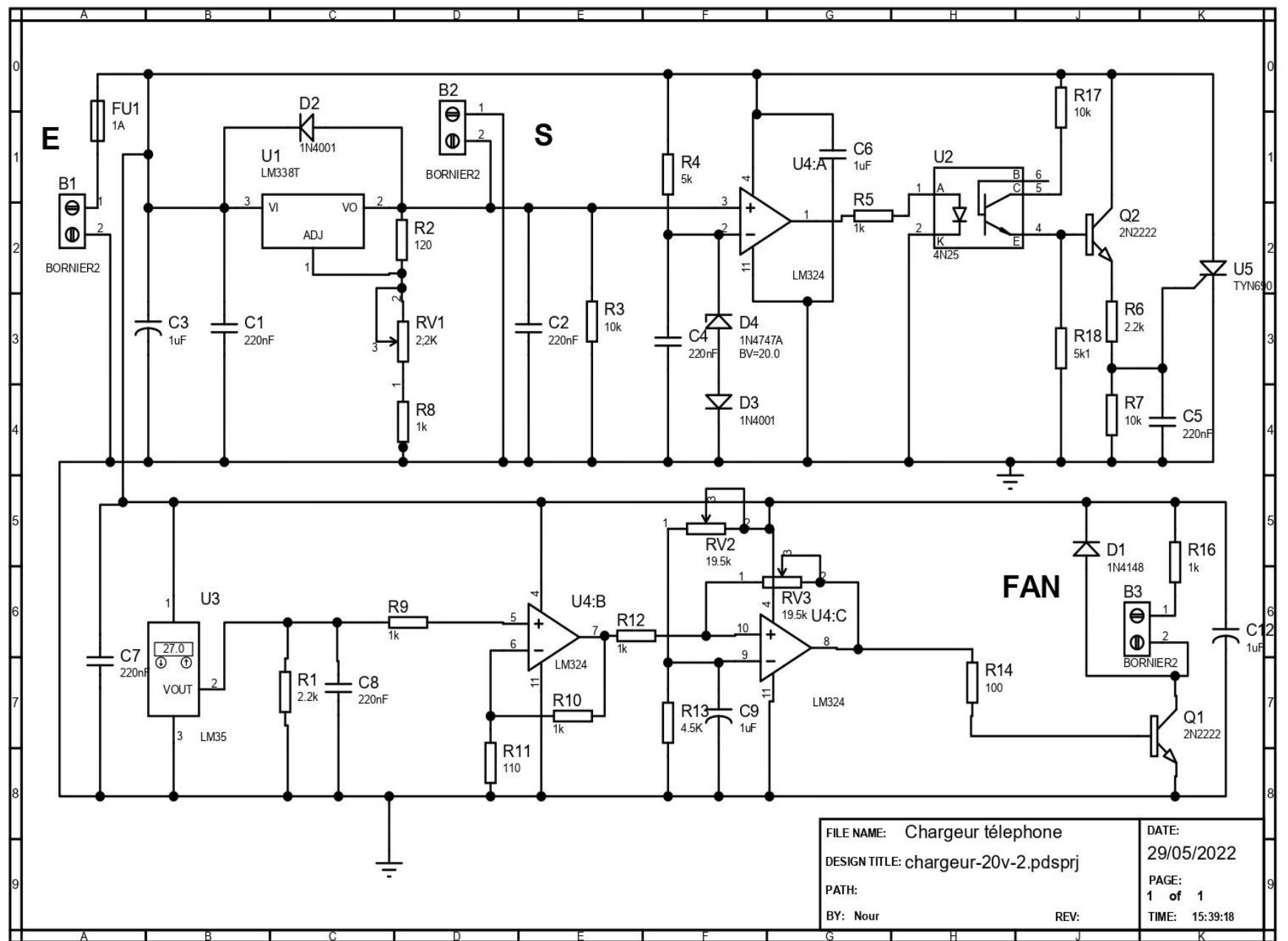
## ● La carte chargeur 5V



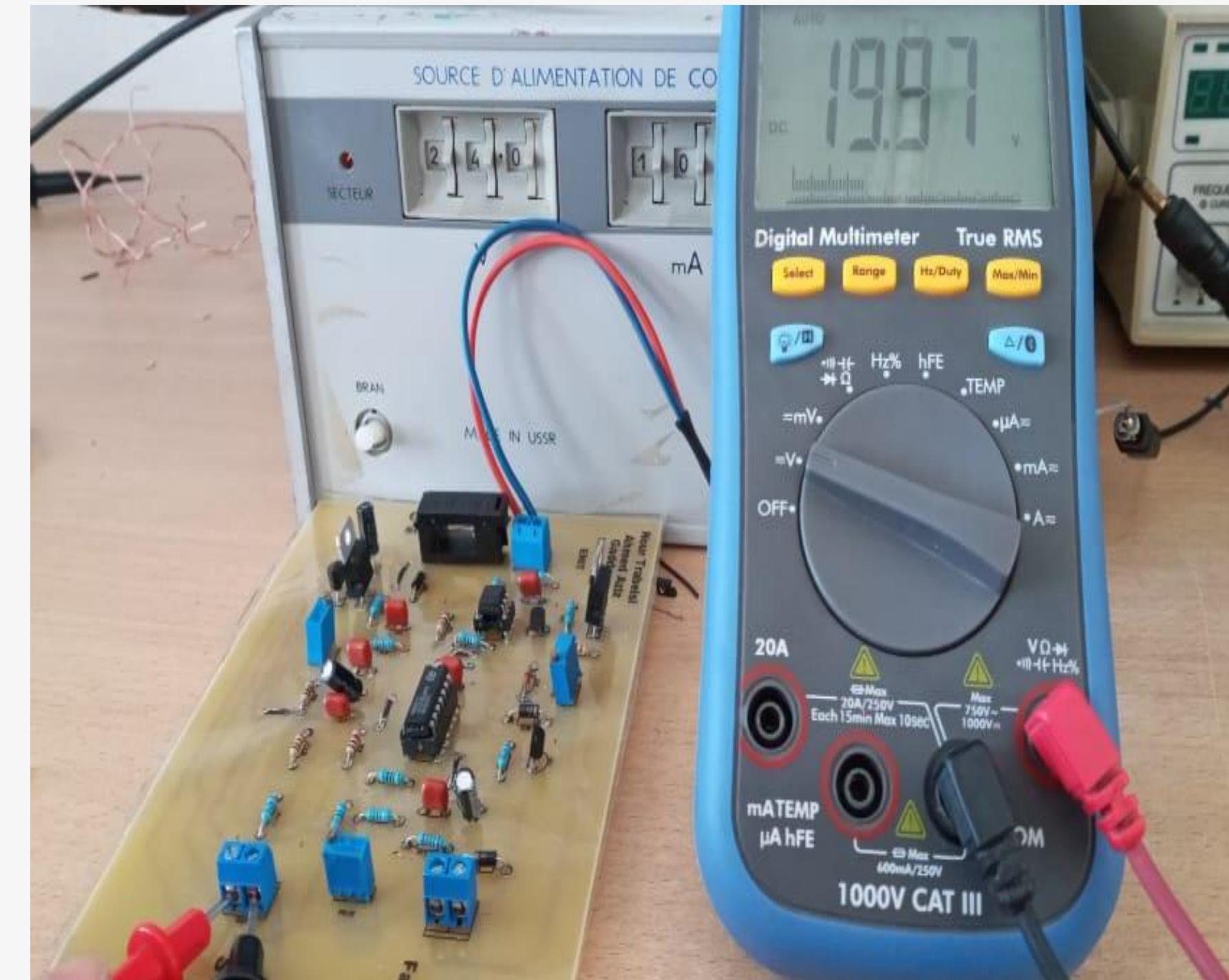
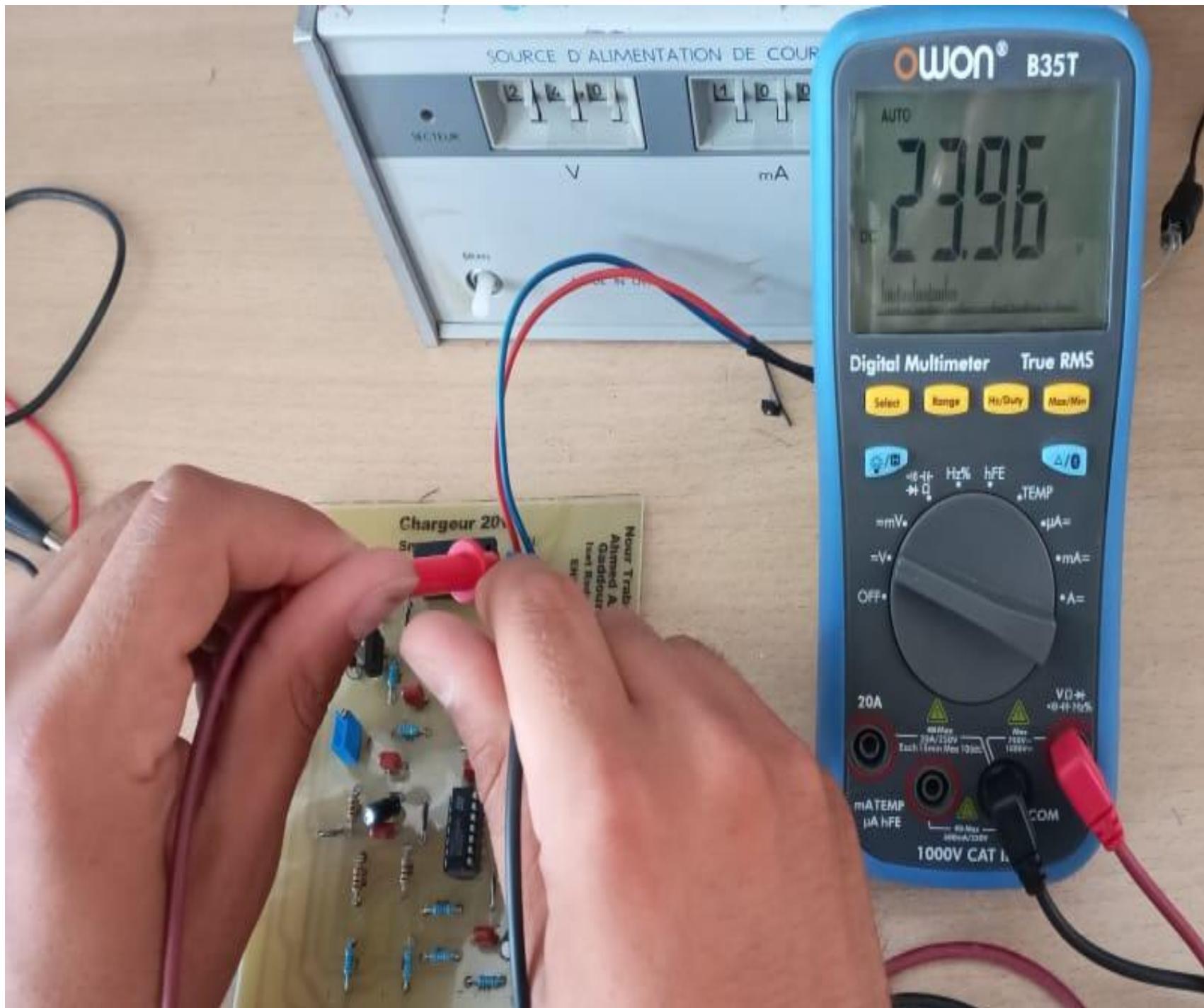
# La partie hardware

## Conception des cartes électroniques

- Conception de la carte chargeur 20V



## ● La carte chargeur 20V

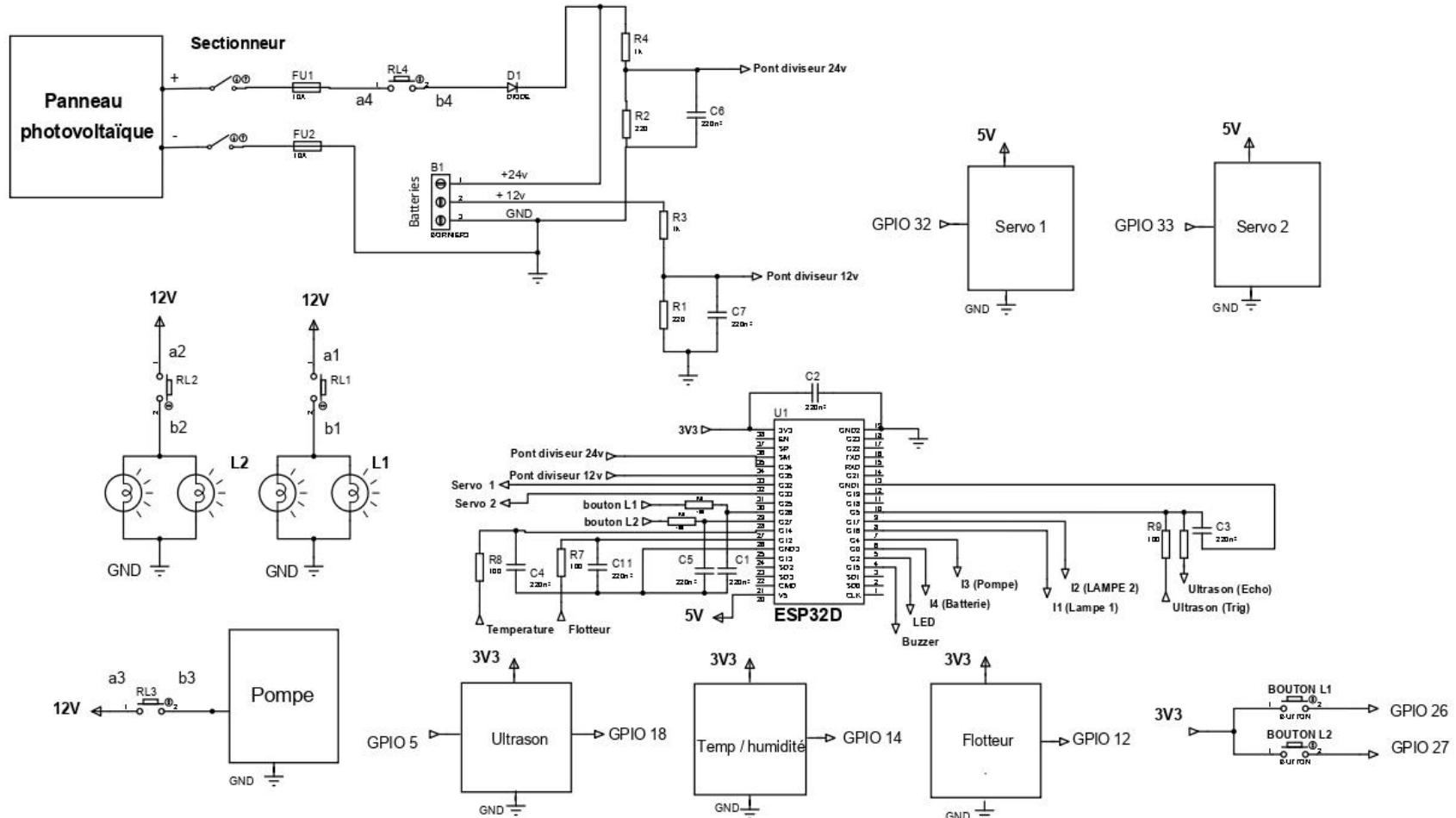


# La partie hardware

## Conception des cartes électroniques

- Conception de la carte commande

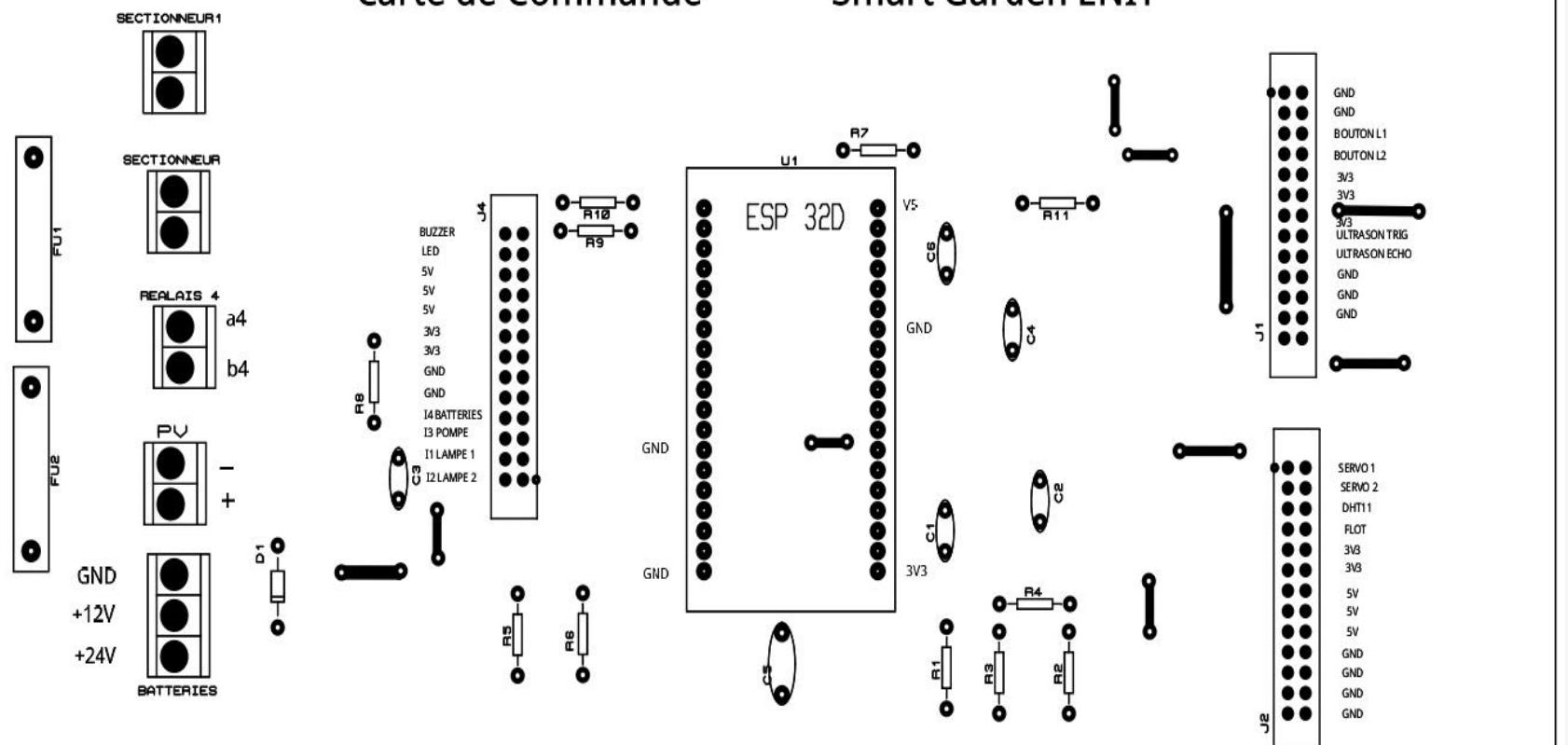
**Synoptique Carte de Commande**



Tous les entrées doivent être capatisées (capacité de découplage = 220nF)

Carte de Commande

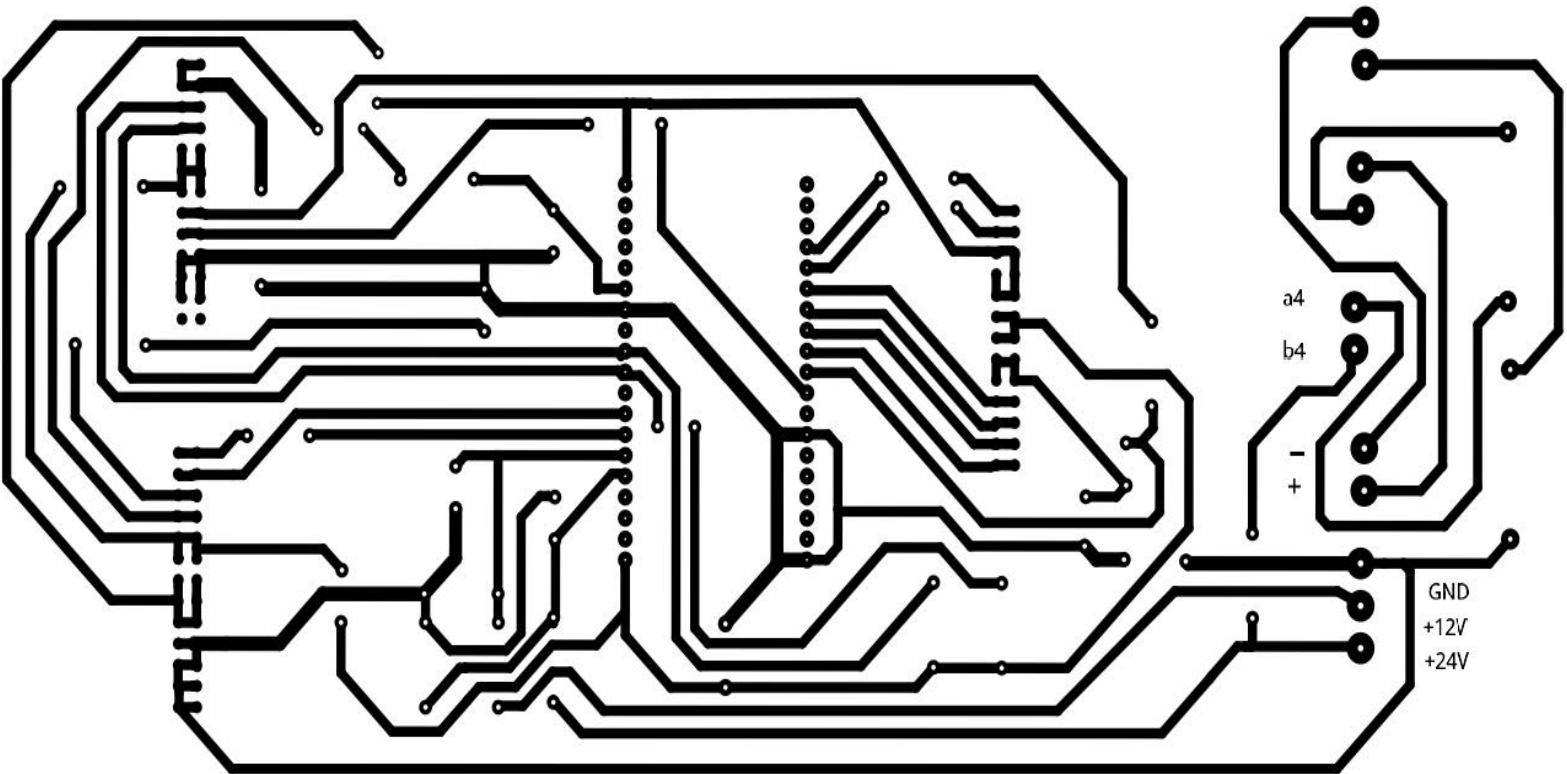
Smart Garden ENIT



Nour Trabelsi - Ahmed Aziz Gaddour - ISET RADES

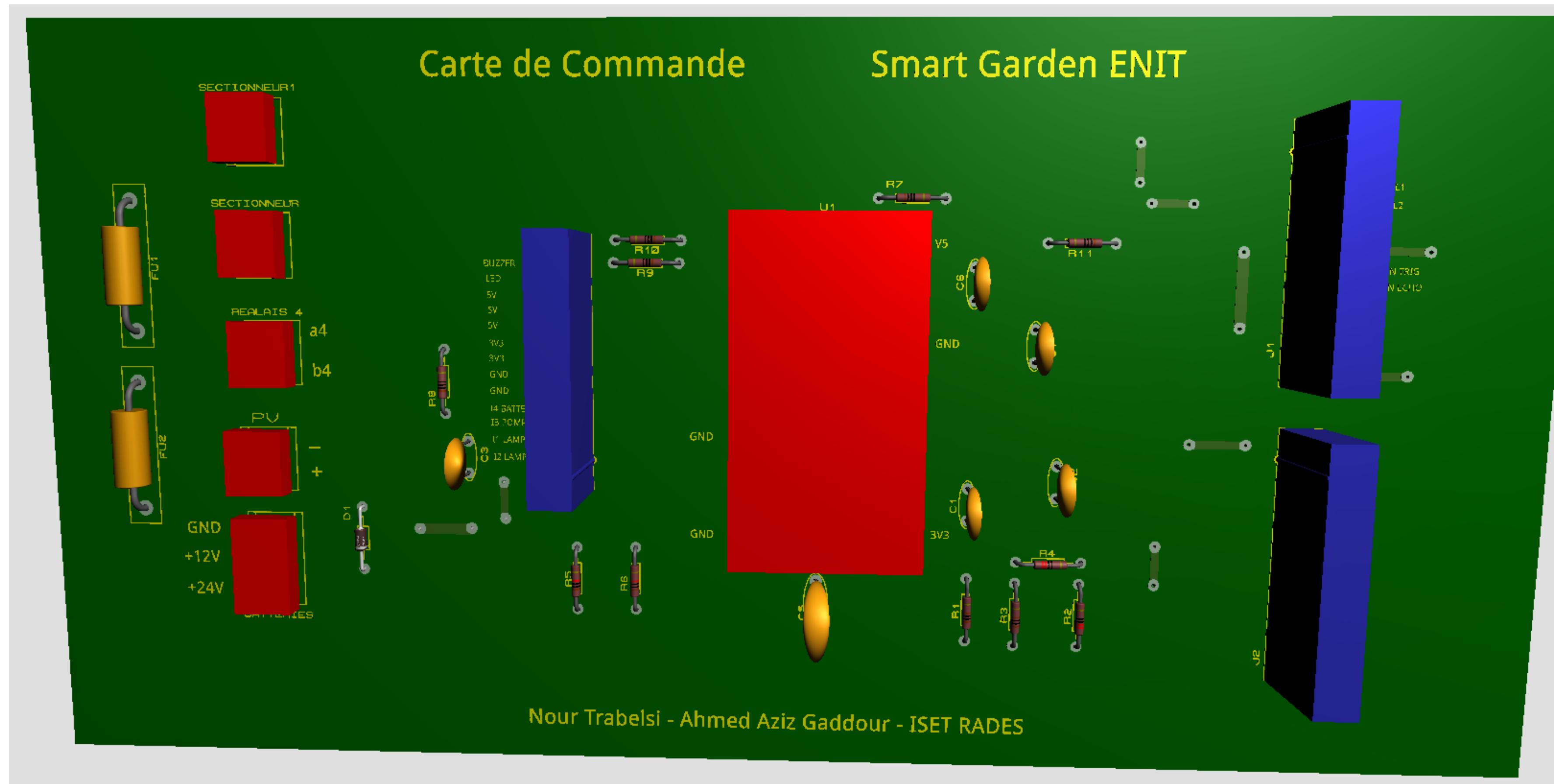
Smart Garden ENIT

Carte de Commande



Nour Trabelsi - Ahmed Aziz Gaddour - ISET RADES

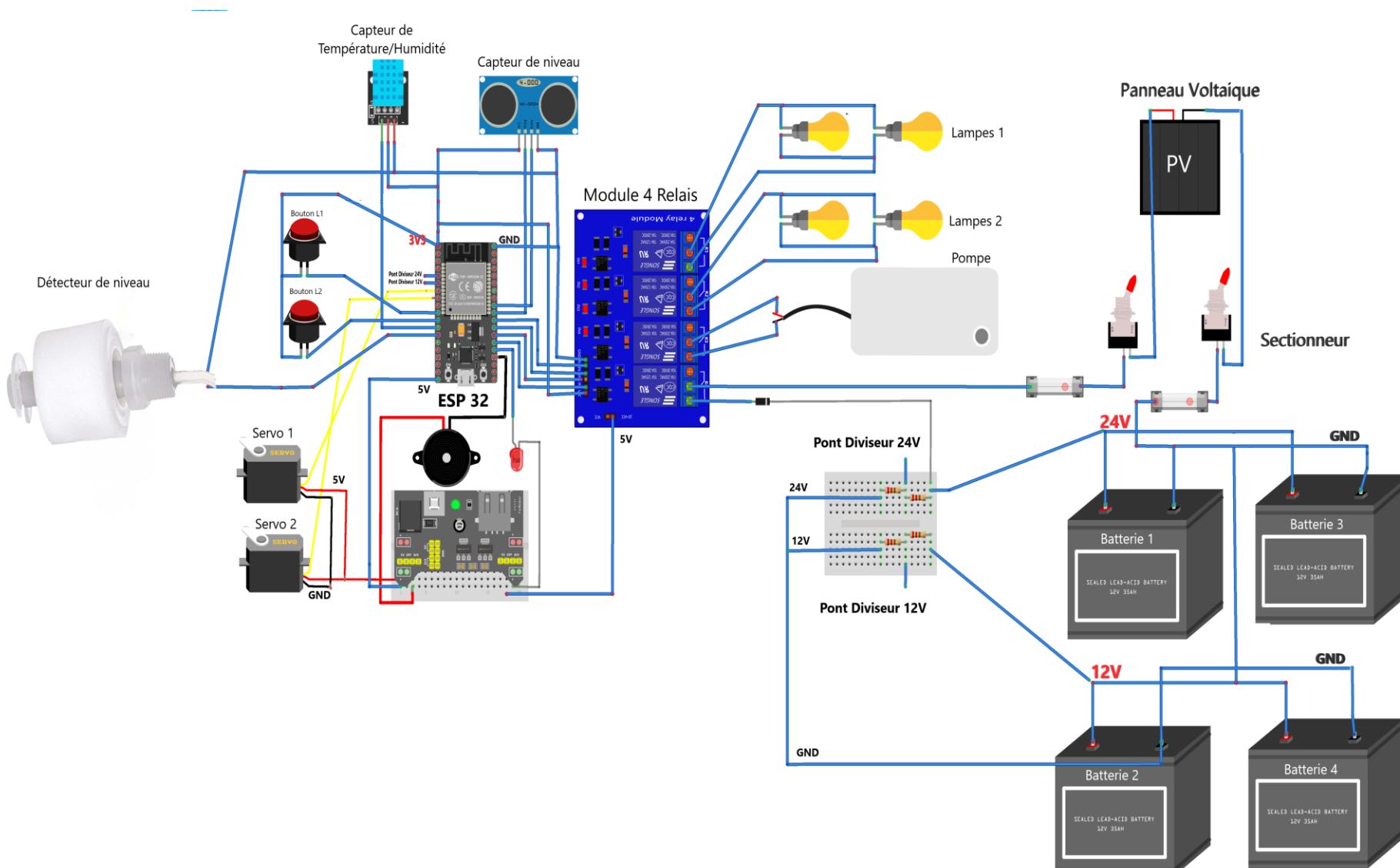
## ● la carte de commande



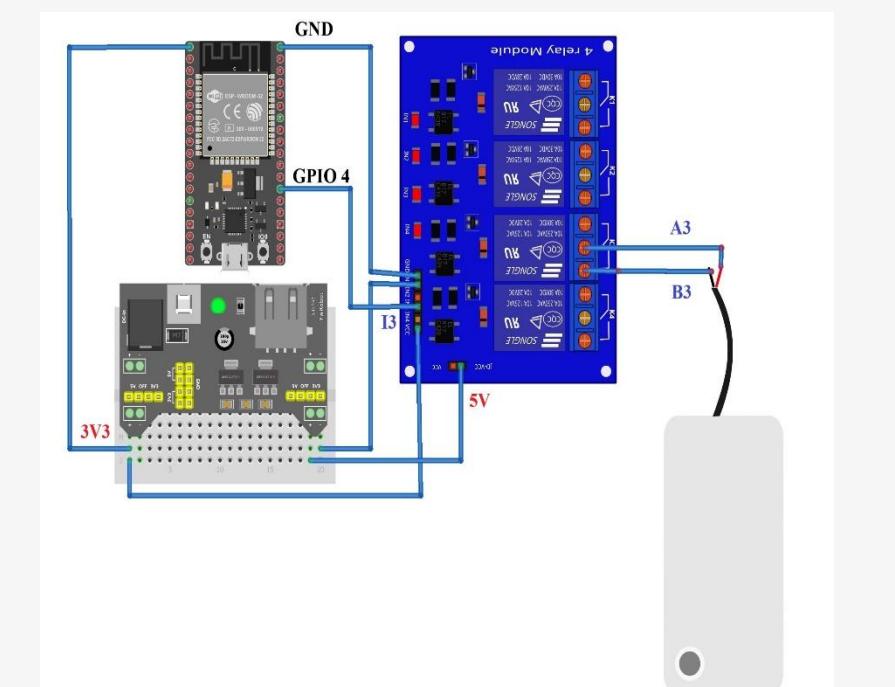
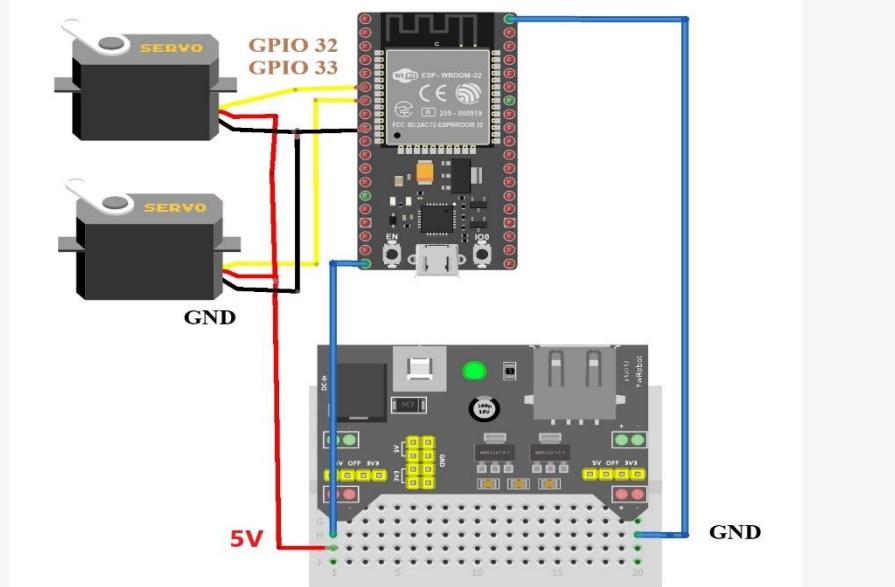
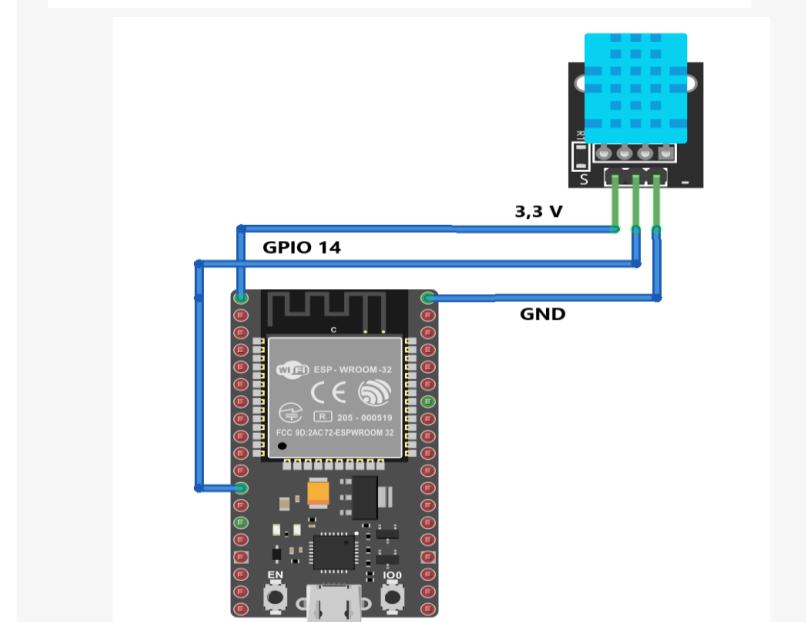
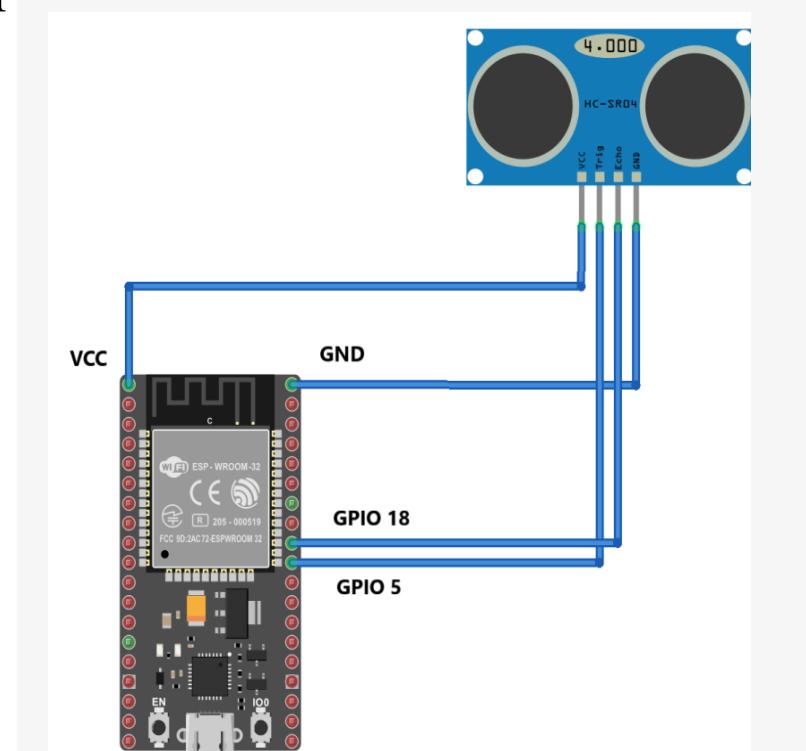
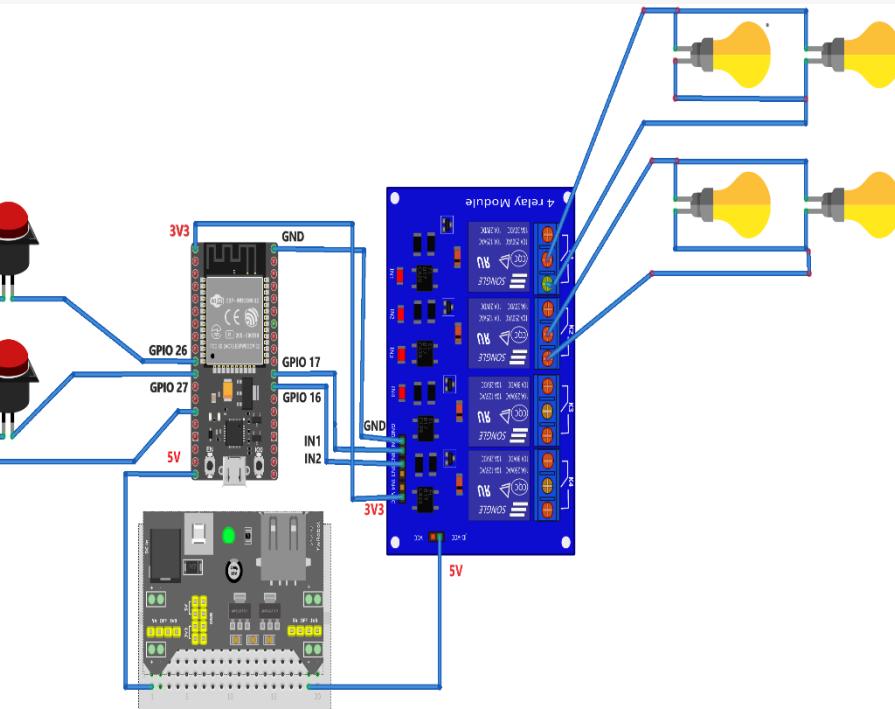
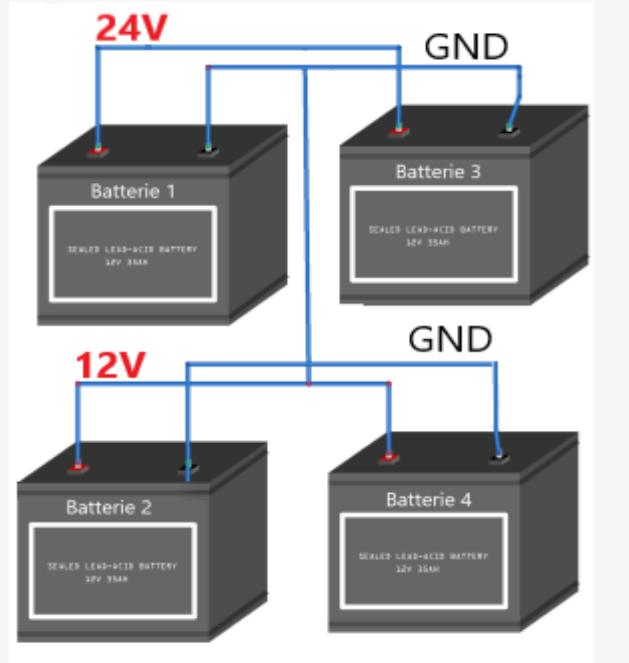
# La partie hardware

## Câblage des composants

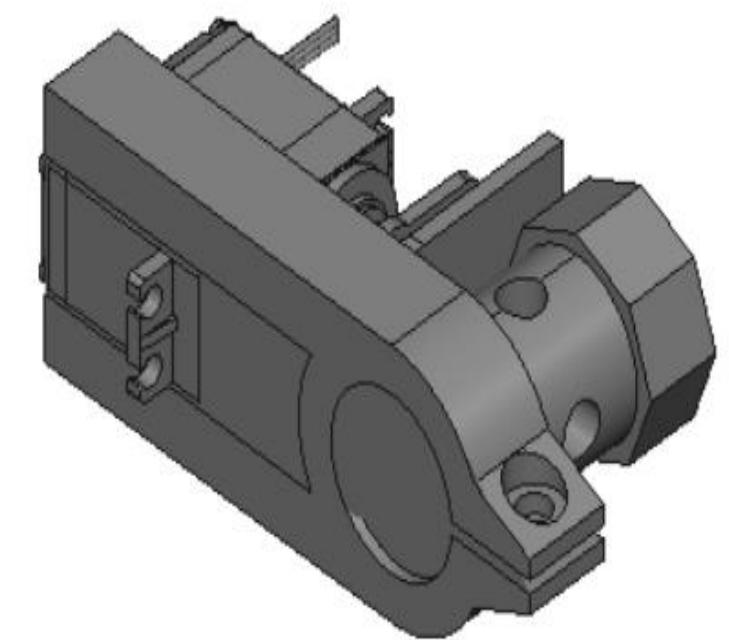
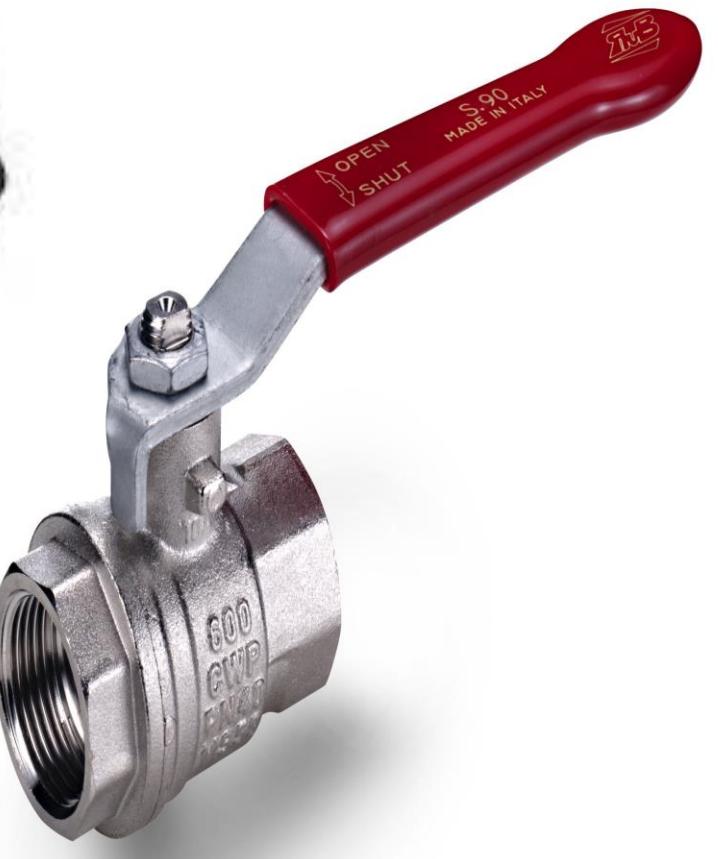
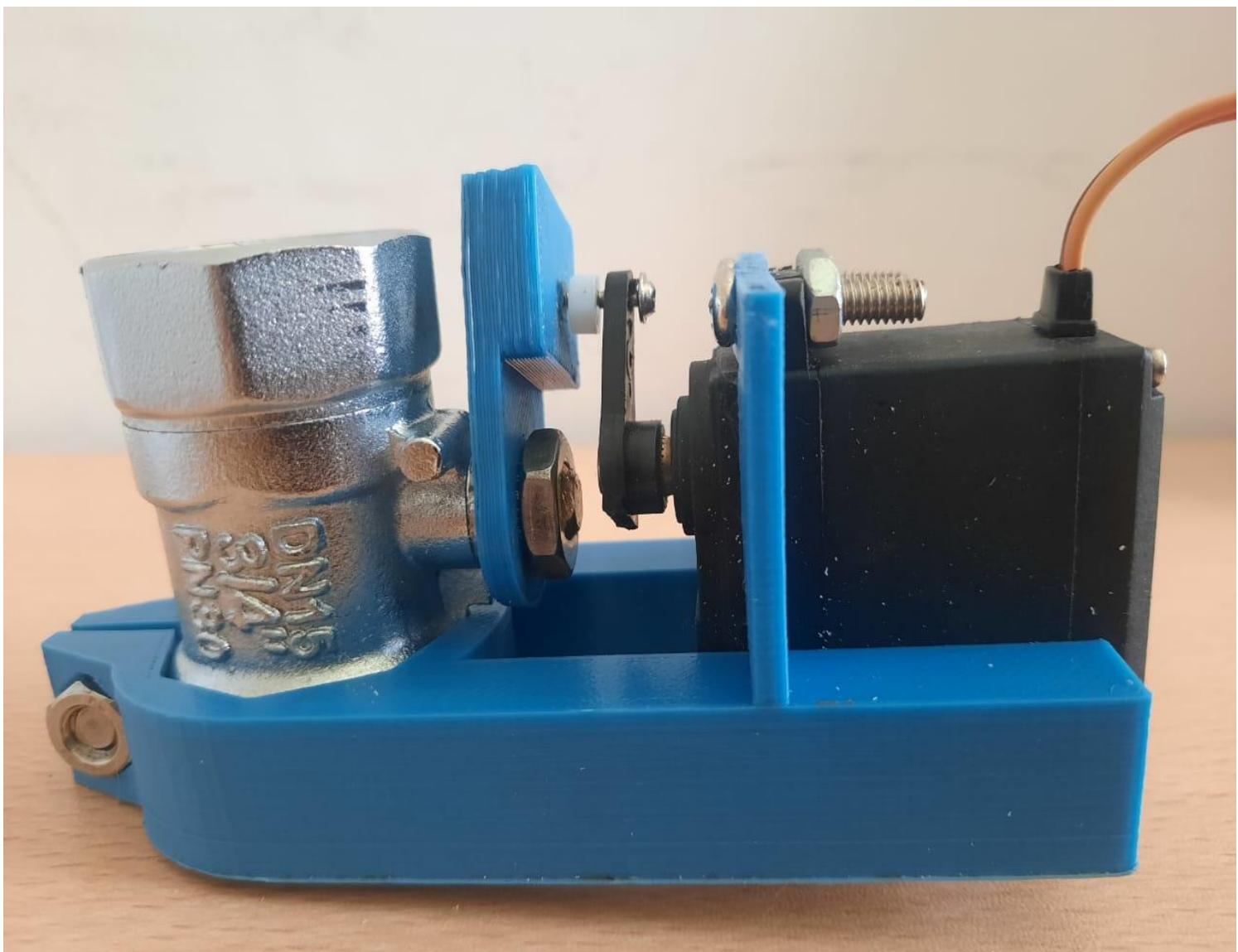
- Câblage de l'ESP 32, Relais, les actionneurs, les capteurs et l'alimentation



fritzing



# Électrovanne motorisée

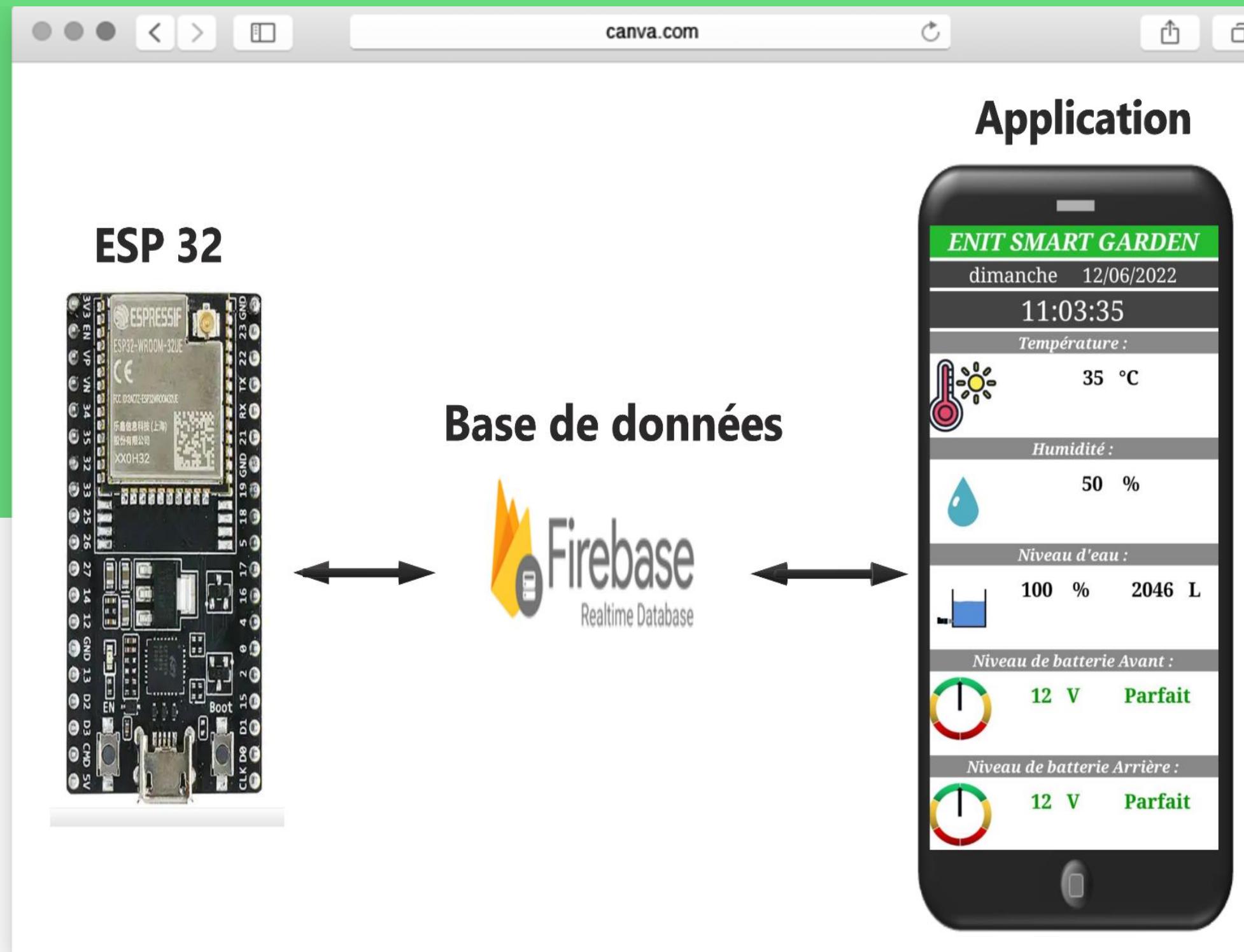


## La partie Software

### Architecture de la plateforme

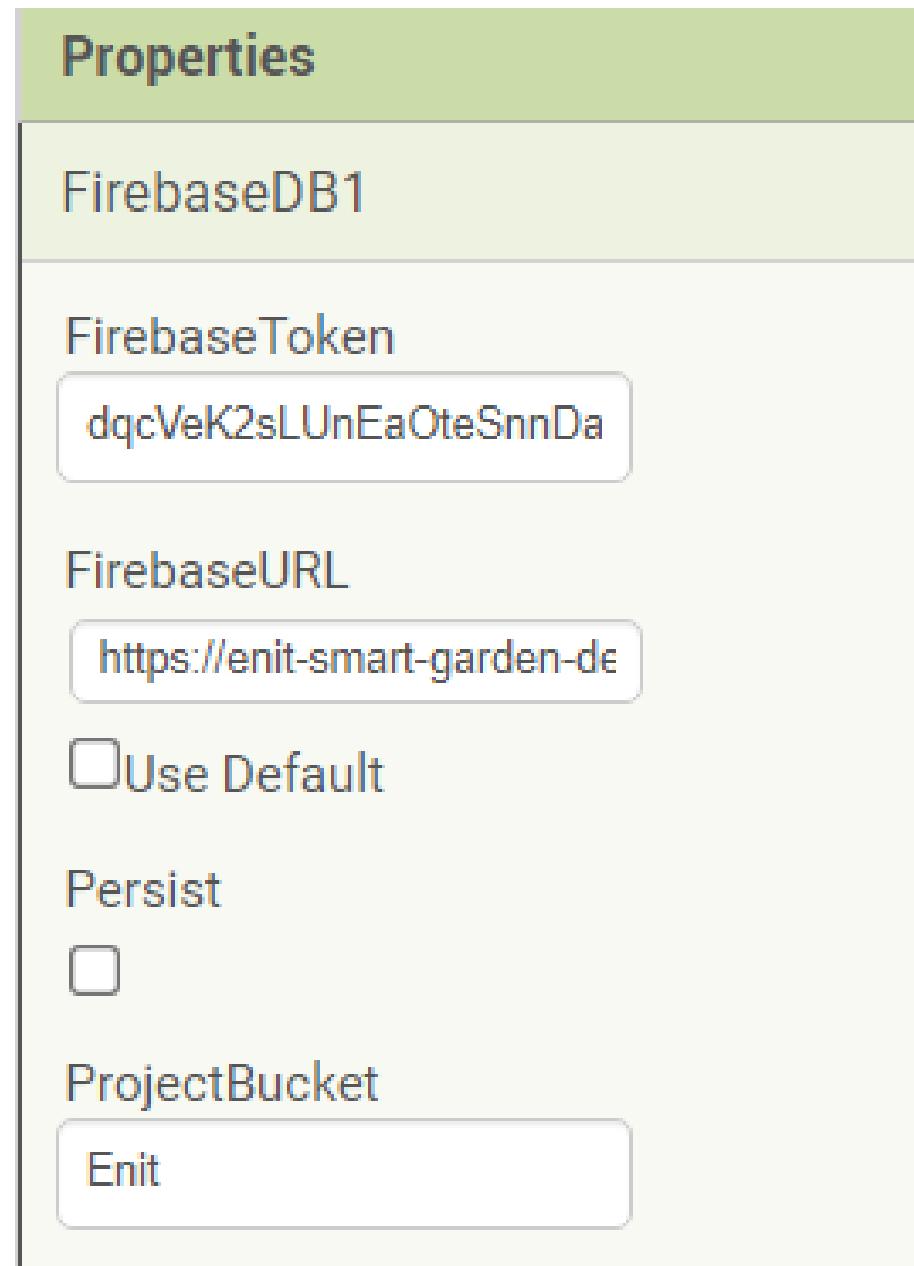
L'application mobile récupère les données mesurées du serveur firebase, qui également reçoit ces mesures du microcontrôleur.

En outre , l'application mobile envoi des commandes à partir de la même firebase qui seront exécutées par le microcontrôleur.



# La partie Software

## La Configuration de Firebase



The screenshot shows the Firebase Realtime Database dashboard for the project "enit smart garden".

**Realtime Database** tab is selected. The interface includes:

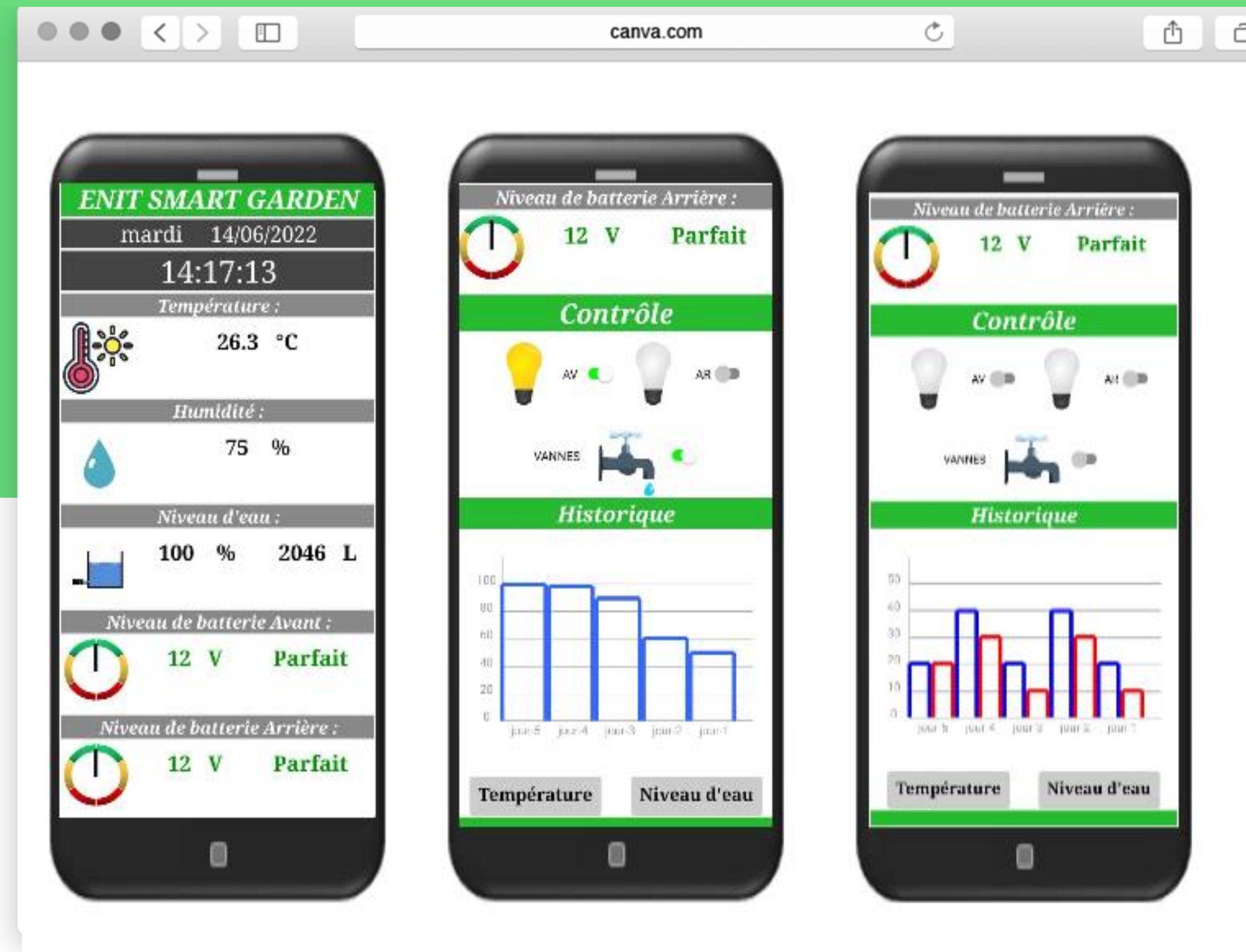
- A sidebar with tabs: Données, Règles, Sauvegardes, Utilisation.
- A banner: "Protégez vos ressources Realtime Database des utilisateurs malintentionnés".
- A URL input field: "https://enit-smart-garden-default-rtbd.firebaseio.com".
- A tree view of the database structure:
  - https://enit-smart-garden-default-rtbd.firebaseio.com/
  - Enit
    - Bat: 12.5
    - Humidity: 50
    - Level: 100
    - LumAr: 0
    - LumAv: 1
    - Robinet: 0
    - Temp: 35
    - jour1a: 10
- An "Emplacement de la base de données : États-Unis (us-central1)" message.
- A central area with controls and history:
  - Contrôle**: Includes a battery level indicator (12.5 V), light control (AV and AR buttons), and valve control (VANNES).
  - Historique**: A bar chart showing historical data for "jour1a" over the last 5 days.
  - Température** and **Niveau d'eau** displays.
- Two separate boxes on the right:
  - "Enit SMART GARDEN" with values: LumAr: 0, LumAv: 1, Robinet: 1.
  - "Historique" showing historical data for "jour1a" over the last 5 days.

## L'application Android « ENIT SMART GARDEN »

Notre solution est basée sur la plateforme app inventer.

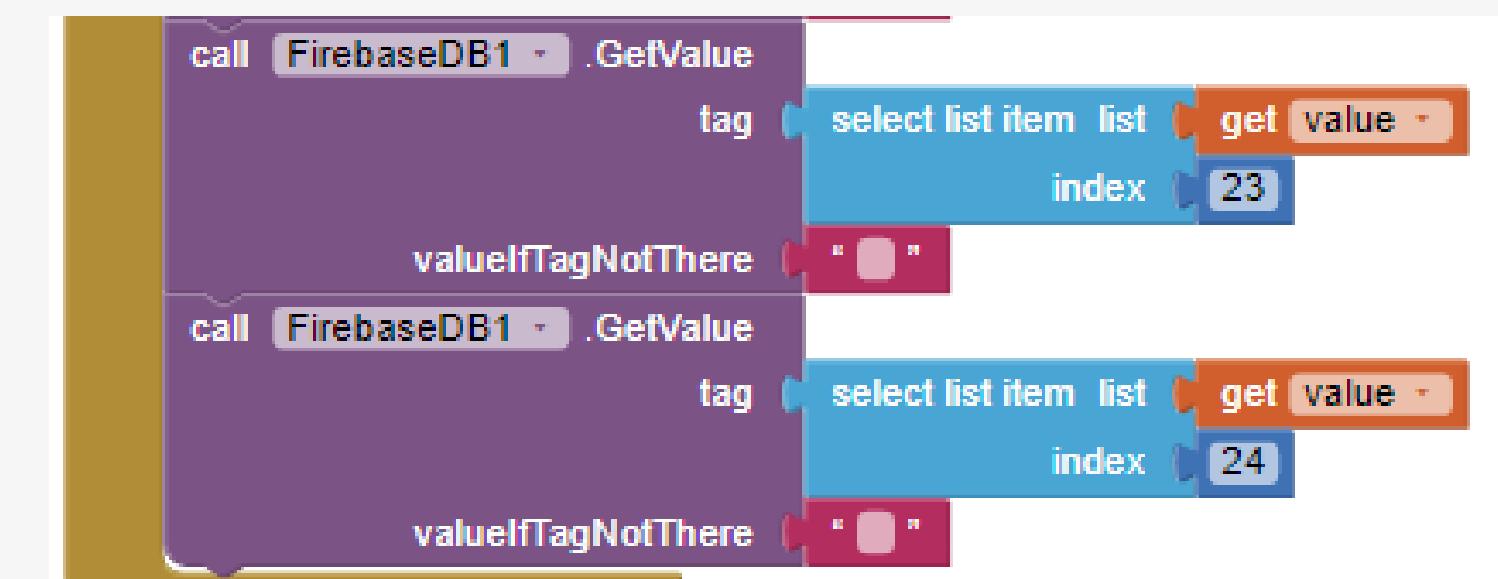
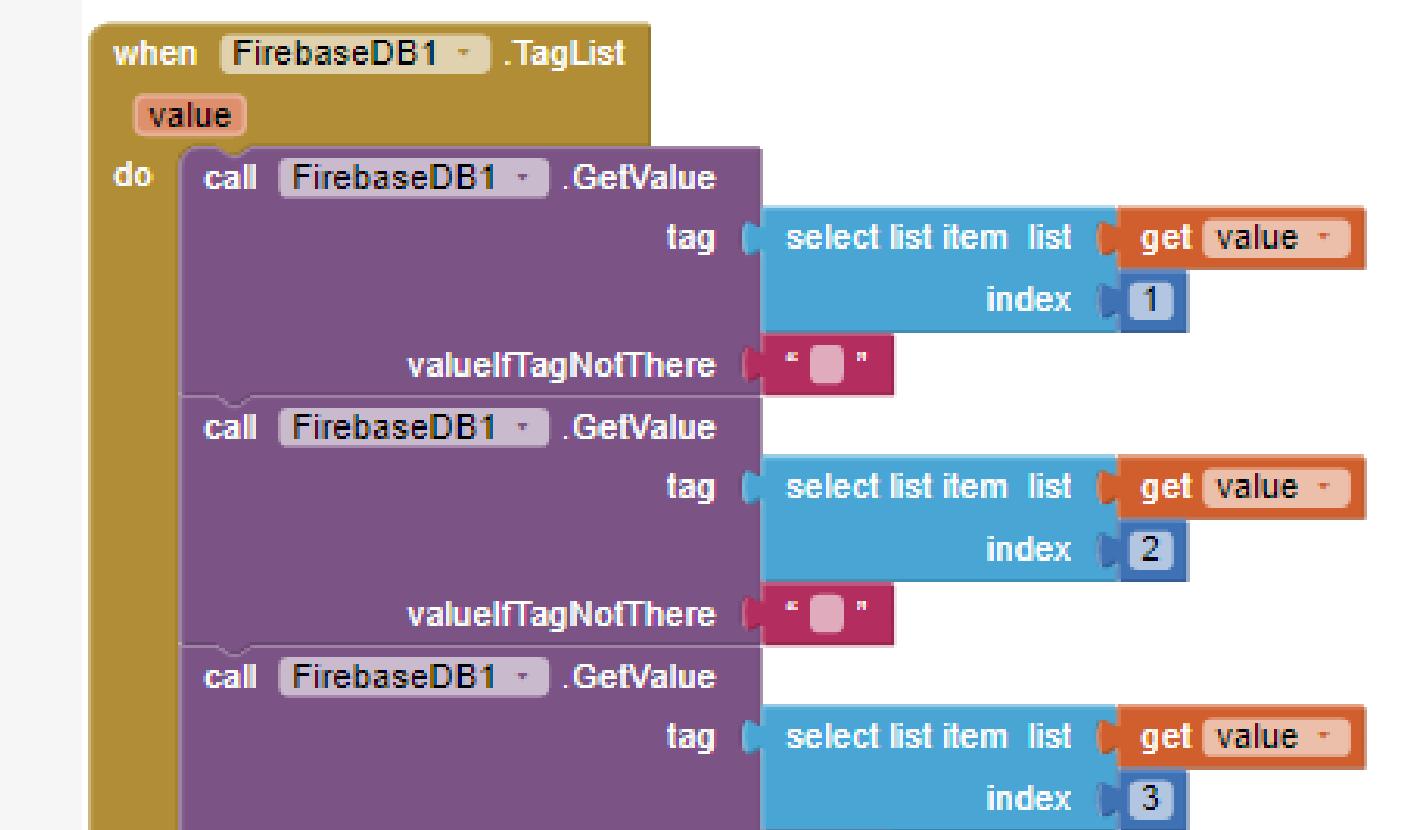
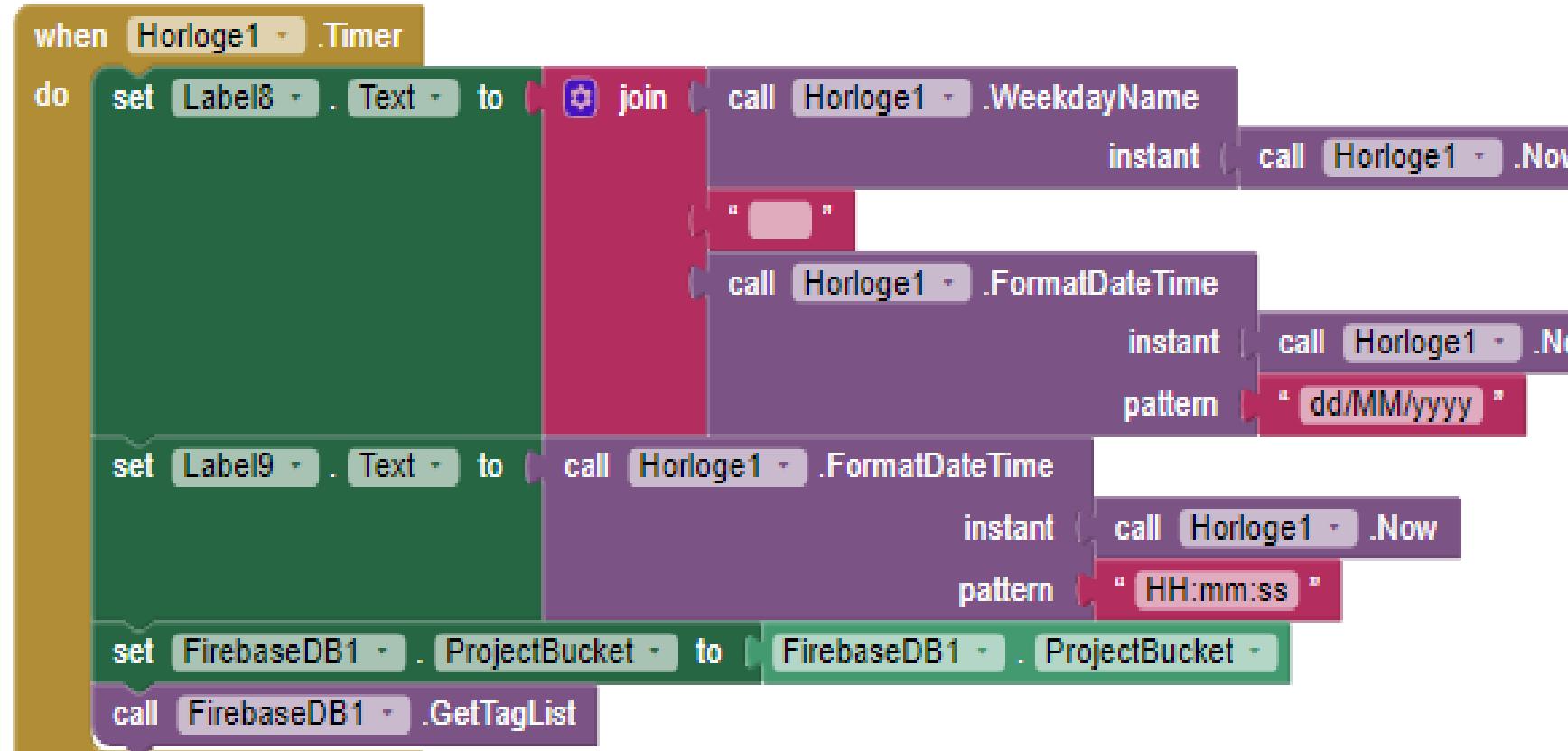
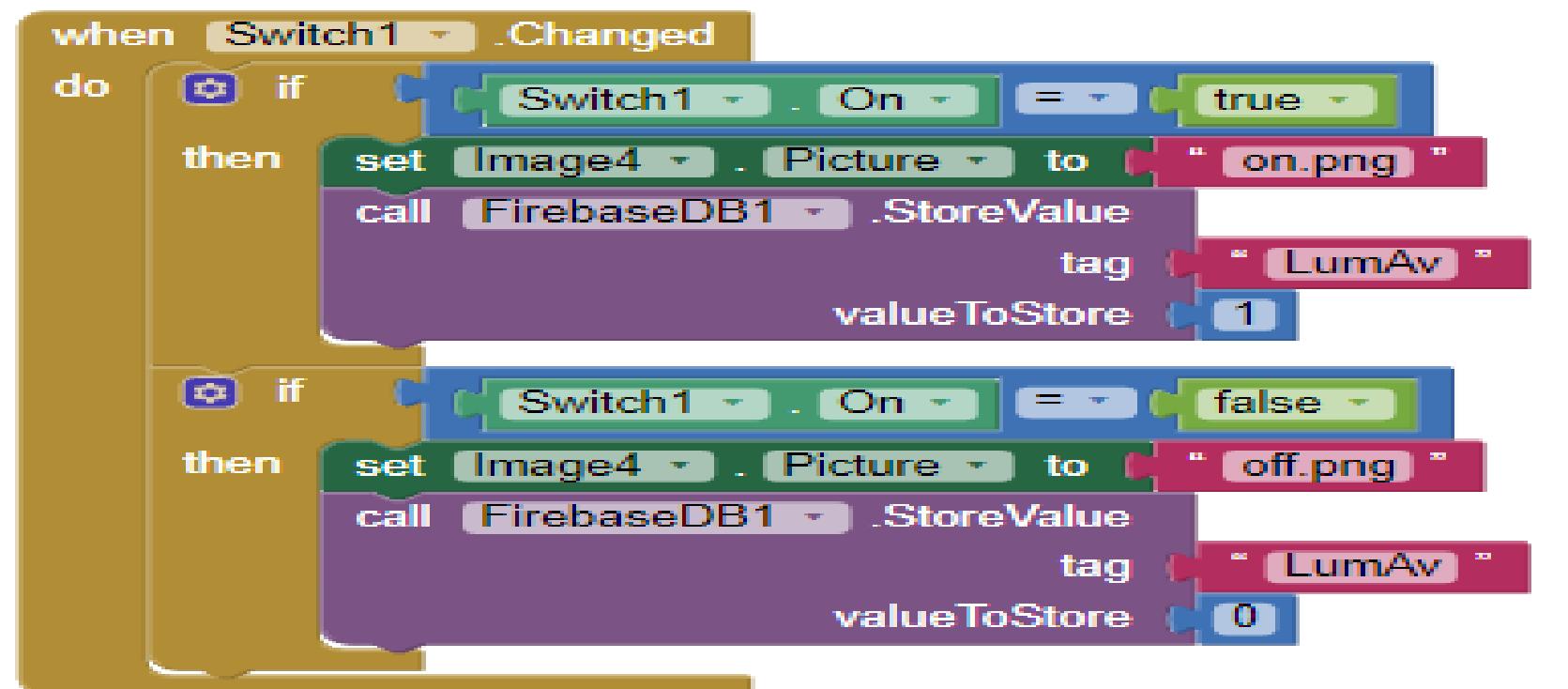
Notre application est composée de 3 parties :

- Un espace qui contient des valeurs à afficher
- Un espace qui contient des boutons de commande
- Un historique qui contient l'évolution de température et du niveau d'eau



# La partie Software

## Développement de l'application Android ENIT Smart Garden



Facebook | YouTube (130) Volkar | enit smart g | MIT App Inv | MIT App Inv | firebase typ | Récupératio | Types de dc | console.firebaseio.google.com/u/0/project/enit-smart-garden/database/enit-smart-garden-default-rtbd/data

Facebook FREE Techno Sampl... (9) Tutorial - How t... برشلونة Lancôme De Lancô... Coffret La Vie est B... (17) Comment auto... Sylenth1 3.041 C Autres favoris

# enit smart garden

## Realtime Database

Données Règles Sauvegardes Utilisation

Protégez vos ressources Realtime Database des utilisations abusives telles que la fraude à la facturation.

<https://enit-smart-garden-default-rtbd.firebaseio.com>

Bat_Ar: 0
Bat_Av: 0
Bat_Gen: 0
Humidity: 32
Level: 100
LumAr: "0"
LumAv: "0"
Robinet: "0"
Temp: 28
jour1a: 10
jour1b: 20

Emplacement de la base de données : États-Unis (us-central1)

The interface includes a top banner with two circular progress bars and a large blue button labeled 'Contrôle'. Below this are two lightbulb icons with sliders labeled 'AV' and 'AR'. A section labeled 'VANNES' shows a valve icon with a slider. A green 'Historique' bar at the bottom right contains a blue bar chart with data points for the last five days.

Créer

- Authentication
- App Check
- Firestore Database
- Realtime Database
- Extensions
- Storage
- Hosting
- Functions
- Machine Learning

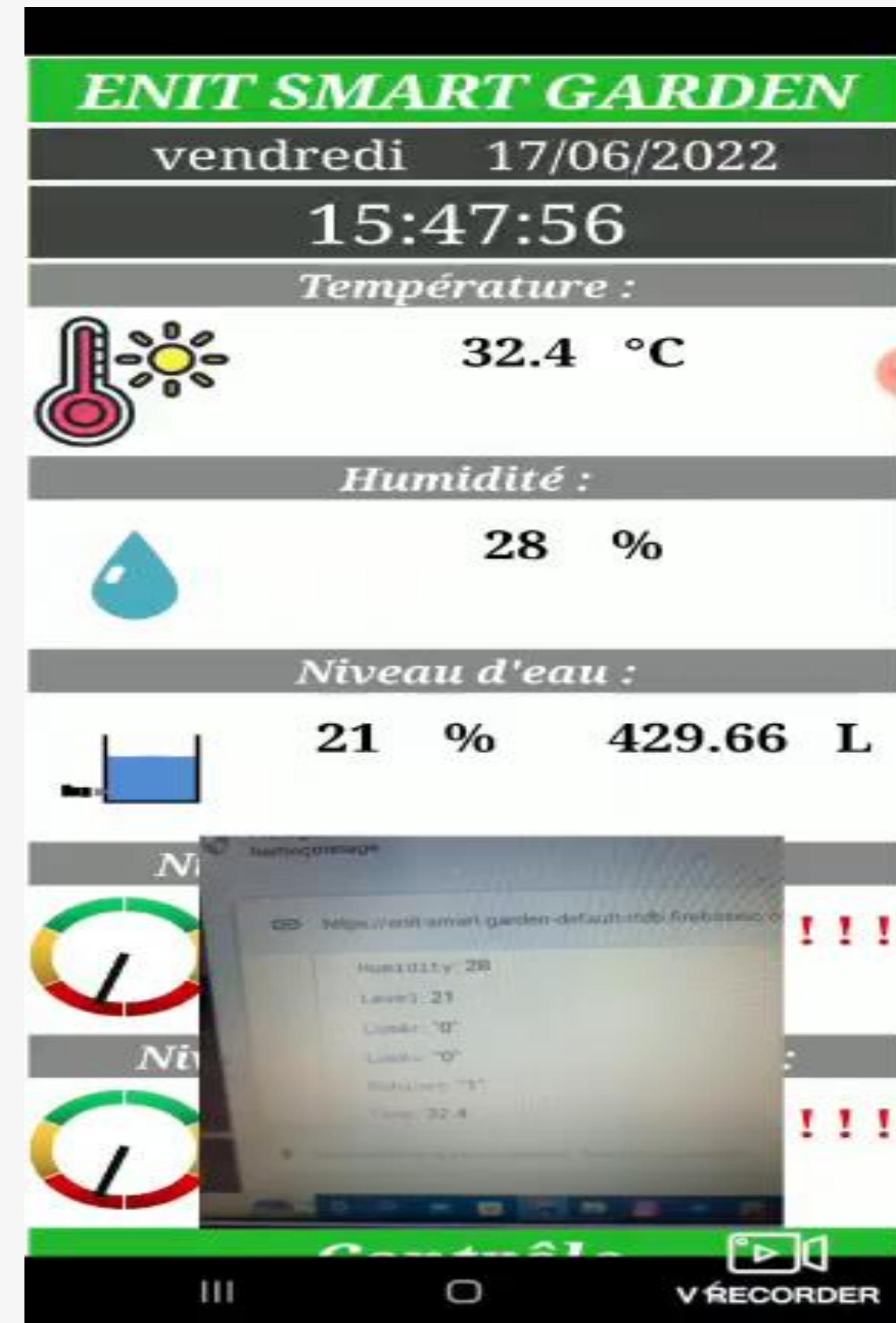
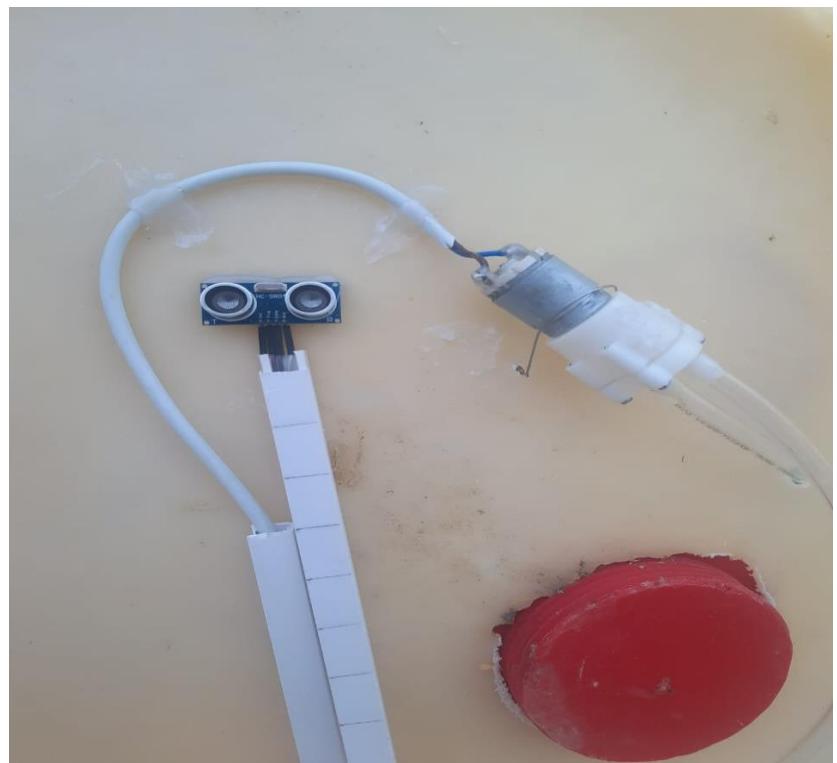
Publier et surveiller

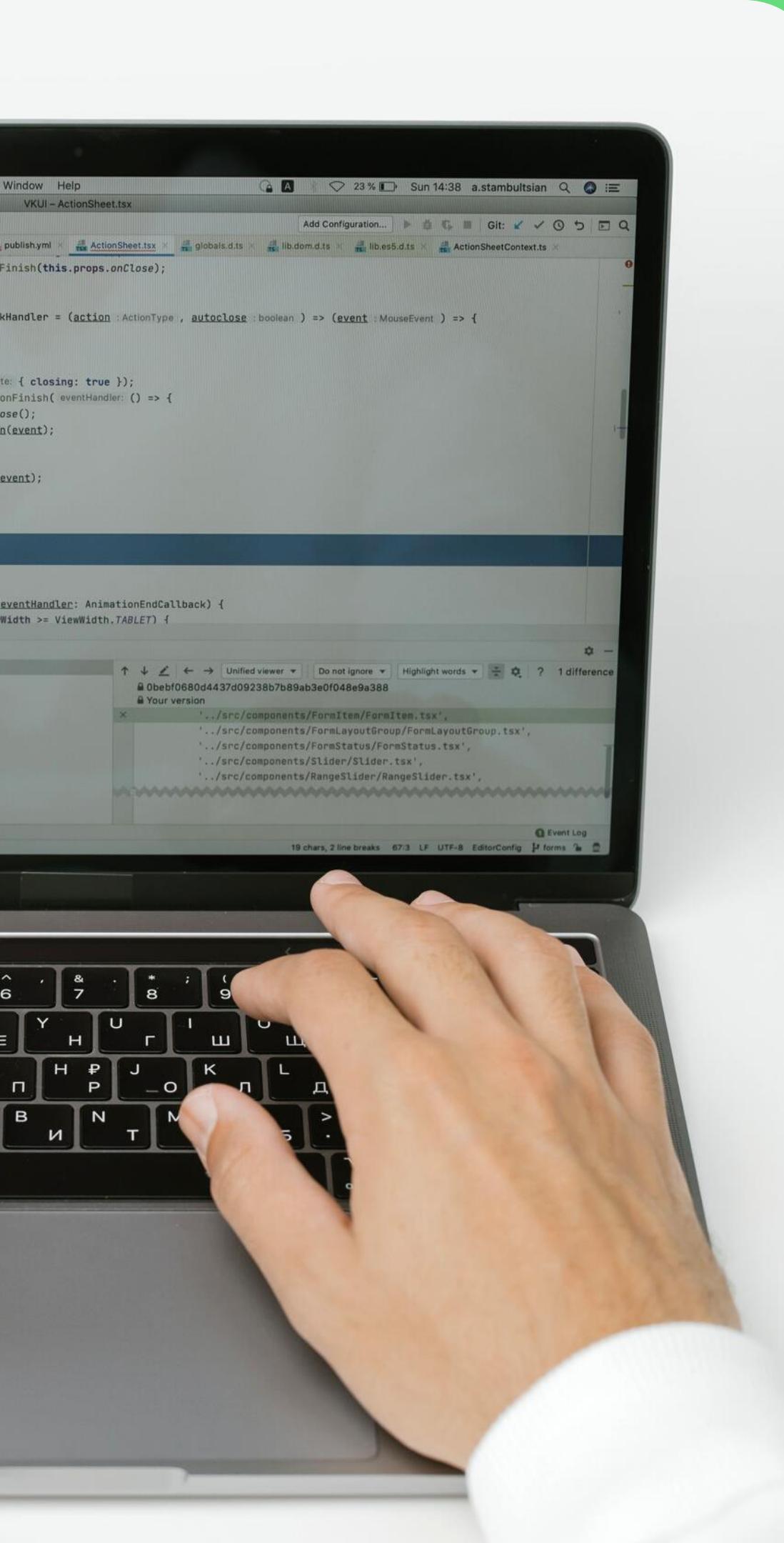
Analytics

Spark Sans frais 0 \$/mois Changer de formule

27/37

# Visualisation de température, Humidité et niveau d'eau





# Programmation de l'ESP-32

## Connexion a un point d'accès WIFI et Firebase

```
//Connexion a un point d'accès WIFI et Firebase
#include <WiFi.h>
#include <FirebaseESP32.h>
#define FIREBASE_HOST "https://enit-smart-garden-default.firebaseio.com"
#define WIFI_SSID "ENITWirless" // Change the name of your WIFI
#define WIFI_PASSWORD "ENITSTUDENT" // Change the password of your WIFI
#define FIREBASE_Authorization_key "dqcVeK2sLUnEaOteSnnDalrl8eBwYfK7RWbcfUvi"
FirebaseData firebaseData;
FirebaseJson json;

Serial.begin(115200);
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("Connecting...");
while (WiFi.status() != WL_CONNECTED)
{
    Serial.print(".");
    delay(300);
}
Serial.println();
Serial.print("IP Address: ");
Serial.println(WiFi.localIP());
Serial.println();

Firebase.begin(FIREBASE_HOST, FIREBASE_Authorization_key);
```

## Déclarations des variables

```
#define DHTPIN 14
const int trigPin = 5;
const int echoPin = 18;
#define RELAIS_POMPE 4
#define RELAIS_LAMPES_AV 17
#define RELAIS_LAMPES_AR 16
#define RELAIS_GENERAL 0
#define BUTTON_AV 26
#define BUTTON_AR 27
#define BAT_AV 35
#define BAT_GEN 34
#define Led 2
#define Buzzer 15
#define flotteur 12

myservo1.attach(32);
myservo2.attach(33);
temp.begin();
pinMode(BUTTON_AV, INPUT);
pinMode(BUTTON_AR, INPUT);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(flotteur, INPUT);
pinMode(BAT_GEN, INPUT);
pinMode(BAT_AV, INPUT);
pinMode(RELAIS_POMPE, OUTPUT);
pinMode(RELAIS_LAMPES_AV, OUTPUT);
pinMode(RELAIS_LAMPES_AR, OUTPUT);
pinMode(RELAIS_GENERAL, OUTPUT);
pinMode(BUZZER, OUTPUT);

dht.begin();
```

## Obtention du date et heure

```
//Obtention du date et heure
#include <NTPClient.h>
#include <WiFiMulti.h>
#include <WiFiUdp.h>
WiFiUDP ntpUDP;
NTPClient temps(ntpUDP, "fr.pool.ntp.org", 3600, 60000);

//Temps et date
temps.update();
epochTime = temps.getEpochTime();
heure = temps.getHours();
minutes = temps.getMinutes();
seconde = temps.getSeconds();
struct tm *ptm = gmtime ((time_t *)&epochTime);
day = ptm->tm_mday;
Serial.print("Month day: ");
Serial.println(day);

mois = ptm->tm_mon+1;
Serial.print("Month: ");
Serial.println(mois);

mois_chaine = months[mois-1];
Serial.print("Month name: ");
Serial.println(mois_chaine);

annee = ptm->tm_year+1900;
Serial.print("Year: ");
Serial.println(annee);
}
```

Mois	Nombre d'irrigation
Janvier	Une fois par semaine
Février	Une fois par semaine
Mars	Deux fois par semaine
Avril	Deux fois par semaine
Mai	Trois fois par semaine
Juin	Trois fois par semaine
Juillet	Quatre fois par semaine
Août	Trois fois par semaine
Septembre	Une fois par semaine
Octobre	Une fois par semaine
Novembre	Une fois par semaine
Décembre	Une fois par semaine

## Système d'irrigation

```
void Arrosage()
{
    Temps();
    //Arrosage automatique
    if ((mois == 1 && day == 2) || (mois == 1 && day == 9) || (mois == 1 && day == 16)
    || (mois == 1 && day == 23) || (mois == 1 && day == 30) )
    {
        if(heure == 0 && minutes ==0 && (seconde == 0 || seconde == 1 || seconde == 2 || seconde == 3))
        {
            for (pos = 0; pos <= 90; pos += 1)
            {
                myservo1.write(pos);
                myservo2.write(pos);
                delay(15);
            }
        }
        if(heure == 0 && minutes ==15 && (seconde == 0 || seconde == 1 || seconde == 2 || seconde == 3))
        {
            for (pos = 90; pos >= 0; pos --)
            {
                myservo1.write(pos);
                myservo2.write(pos);
                delay(15);
            }
        }
    }
}
```

## Système de gestion des batteries

```
void Battery()
{
    Bat_Av = (digitalRead(BAT_AV)/4095)*37.5;
    Bat_Gen = (digitalRead(BAT_AV)/4095)*37.5;
    Bat_Ar = Bat_Gen - Bat_Av ;
    Firebase.setFloat(firebaseData, "/Enit/Bat_Av", Bat_Av);
    Firebase.setFloat(firebaseData, "/Enit/Bat_Ar", Bat_Ar);
    Firebase.setFloat(firebaseData, "/Enit/Bat_Gen", Bat_Gen);

    Serial.println(Bat_Av);

    // Alarme et coupure recharge par panneau
    while (Bat_Av > 12.5)
    {
        Serial.println("programme lumiere");
        digitalWrite(RELAIS_GENERAL, LOW);
        digitalWrite(BUZZER, HIGH);
        digitalWrite(RELAIS_LAMPES_AV, HIGH);
        digitalWrite(RELAIS_LAMPES_AR, HIGH);
        Temperature();
        Flotteur();
        Niveau();
        Acquisition();
        Vanne();
        Arrosage();
        Allumer1();
        Allumer2();
    }
}
```

## Système de Gestion du réservoir 1

```
void Niveau()
{
    //Niveau eau ultrason
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Reads the echoPin, returns the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);

    // Calculate the distance
    distanceCm = duration * SOUND_SPEED/2;
    Serial.print("distance: ");
    Serial.println(distanceCm);
    distancePercent = 100 - distanceCm/70*100;
    Serial.println(distancePercent);
    Firebase.setFloat(firebaseData, "/Enit/Level", (int)distancePercent);
}
```

## Commande de la pompe

```
void Flotteur ()
{
    etat_FLOT = digitalRead(flotteur);
    if (etat_FLOT == 1)
    {
        digitalWrite(RELAIS_POMPE, HIGH);
    }
    else
    {
        digitalWrite(RELAIS_POMPE, LOW);
    }
}
```

## Système d'éclairage du jardin

```
void Allumer1()
{
    if (Firebase.RTDB.getInt(&firebaseData, F("/Enit/LumAv")))
    {
        String(firebaseData.toInt<>()).c_str();
        LumAv = firebaseData.intData();
        Serial.print("lumAv = ");
        Serial.println(LumAv);

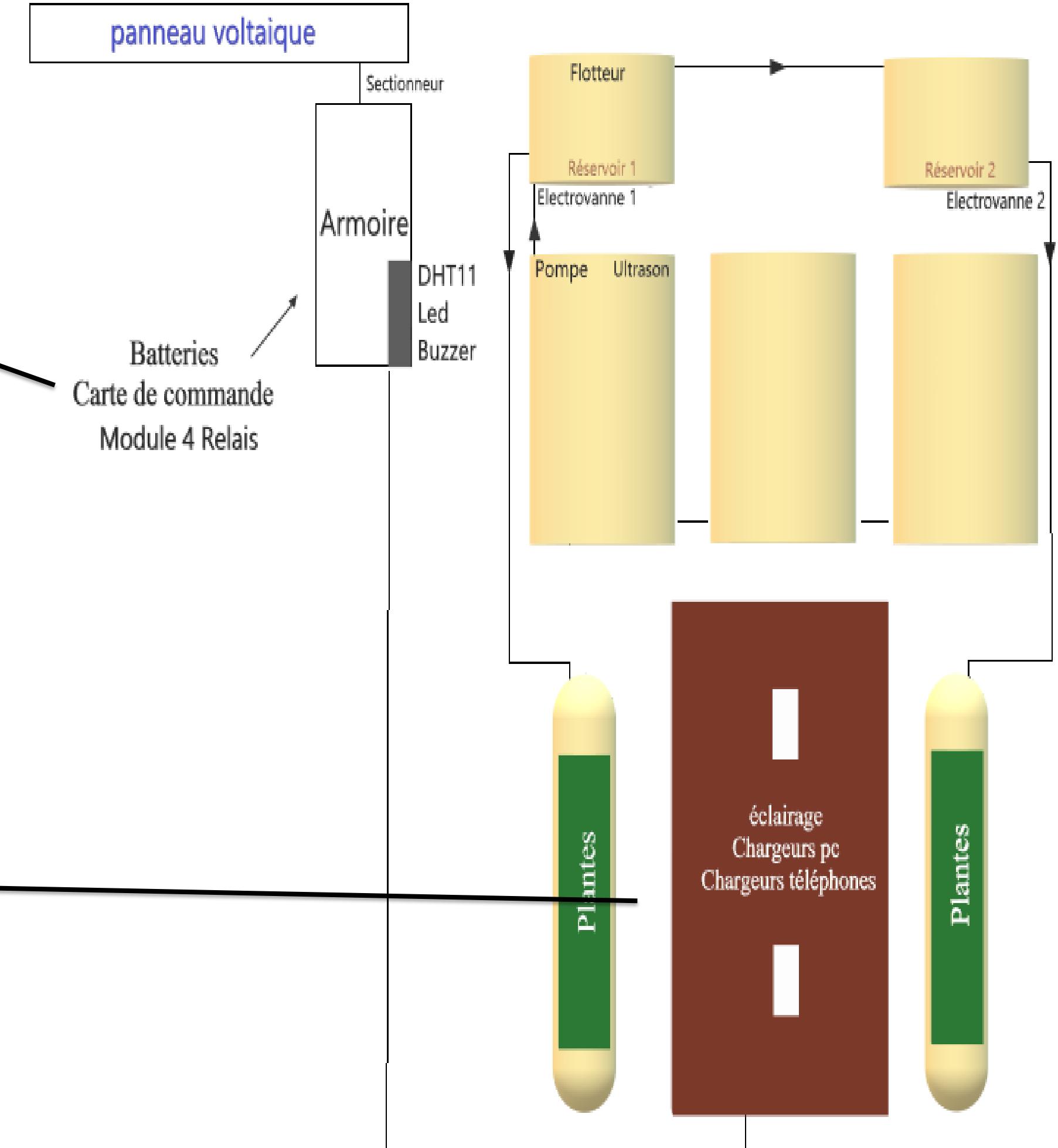
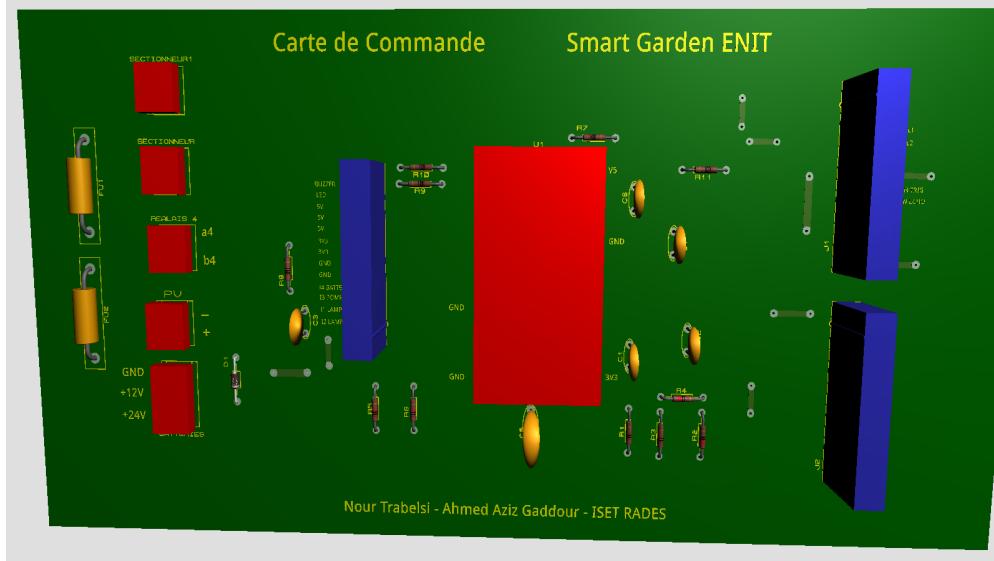
    }
    //Lecture des boutons posés sur la table
    ButtonAv = digitalRead(BUTTON_AV);
    //Allumer les lampes
    if ((ButtonAv == HIGH) || (LumAv == 1))
    {
        Serial.println("programme lumiere av");
        Serial.println("button 1 activé");
        temps_depart = millis();
        digitalWrite(RELAIS_LAMPES_AV, HIGH);
        while ((millis() - temps_depart) <= 600)
        {
            Temperature();
            Flotteur();
            Niveau();
            Acquisition();
            Vanne();
            Arrosage();
            Allumer2();
            Battery();
        }
        digitalWrite(RELAIS_LAMPES_AV, LOW);
        Firebase.setInt(firebaseData, "/Enit/LumAv", 0);
    }
}
```

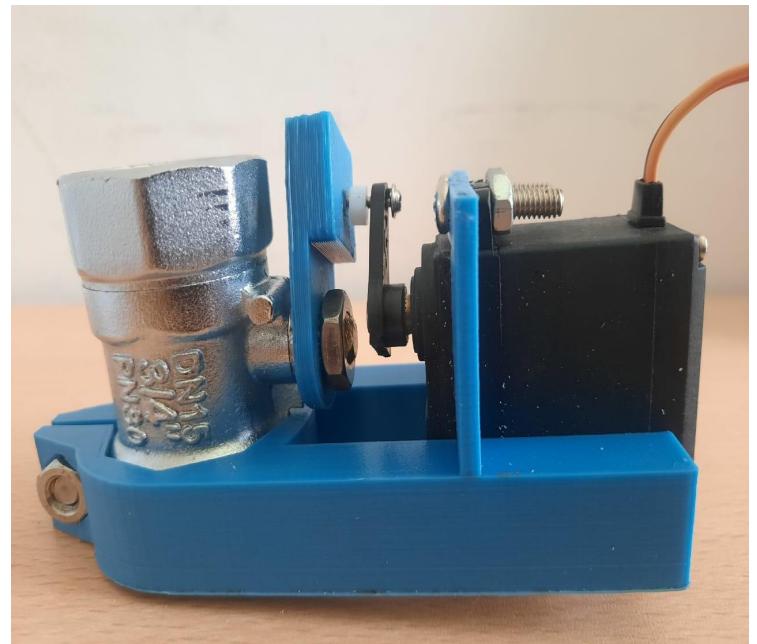
## Supervision de la température et d'humidité

```
void Temperature()
{
    // temperature
    float hum = dht.readHumidity();
    float temp = dht.readTemperature();
    if (isnan(hum) || isnan(temp) )
    {
        Serial.println(F("Failed to read from DHT sensor!"));
        return;
    }

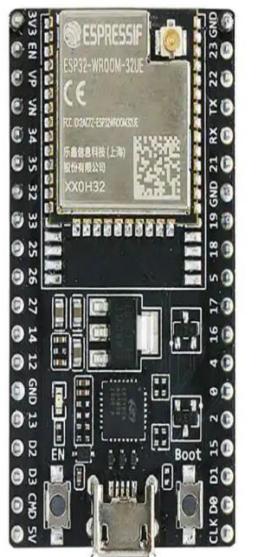
    Serial.print("Temperature: ");
    Serial.print(temp);
    Serial.print("°C");
    Serial.print(" Humidity: ");
    Serial.print(hum);
    Serial.print("%");
    Serial.println();

    Firebase.setFloat(firebaseData, "/Enit/Temp", temp);
    Firebase.setFloat(firebaseData, "/Enit/Humidity", hum);
}
```





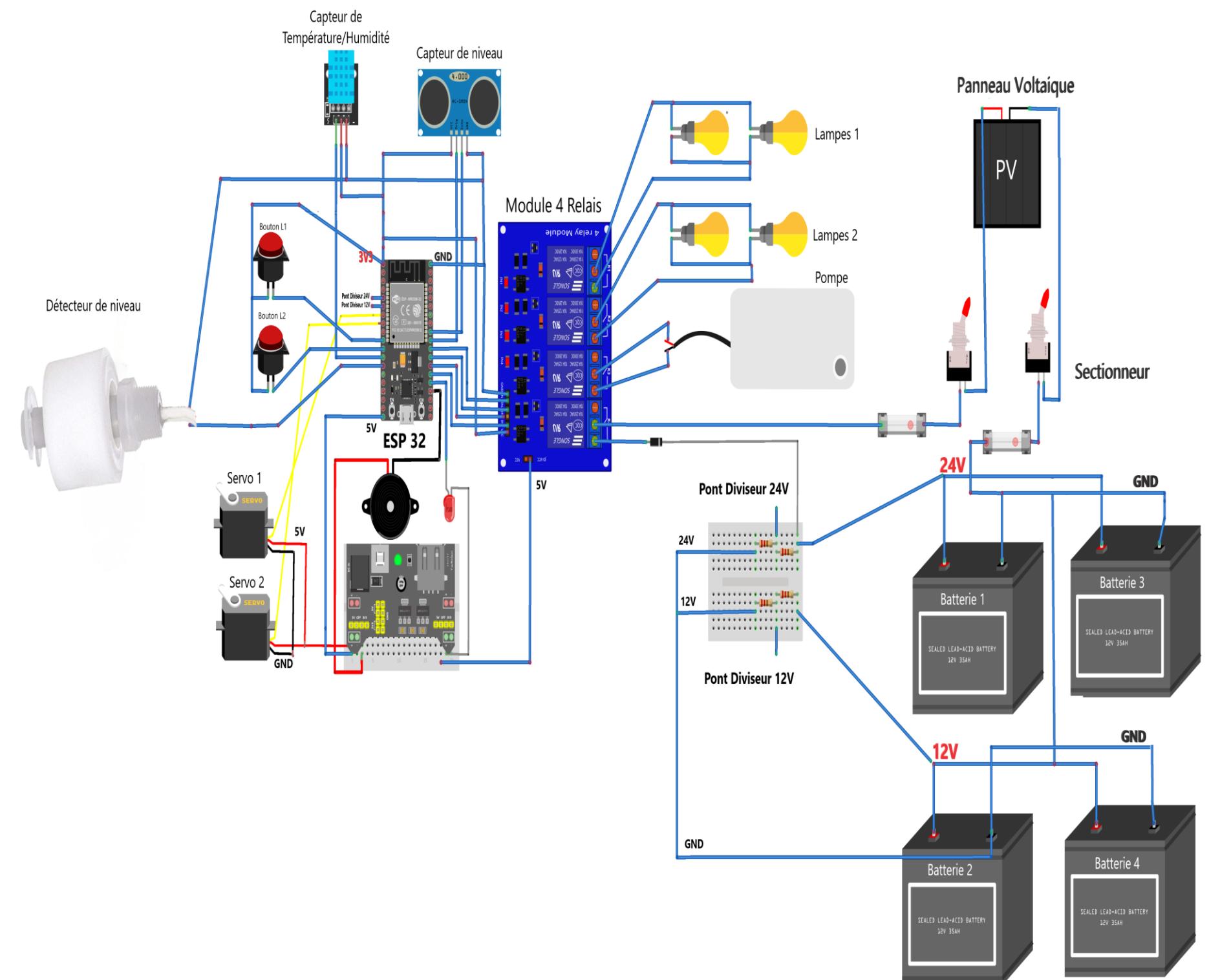
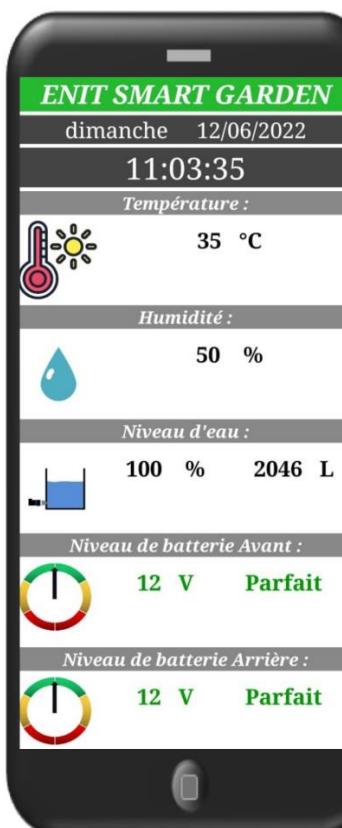
**ESP 32**



**Base de données**



## Application



fritzing

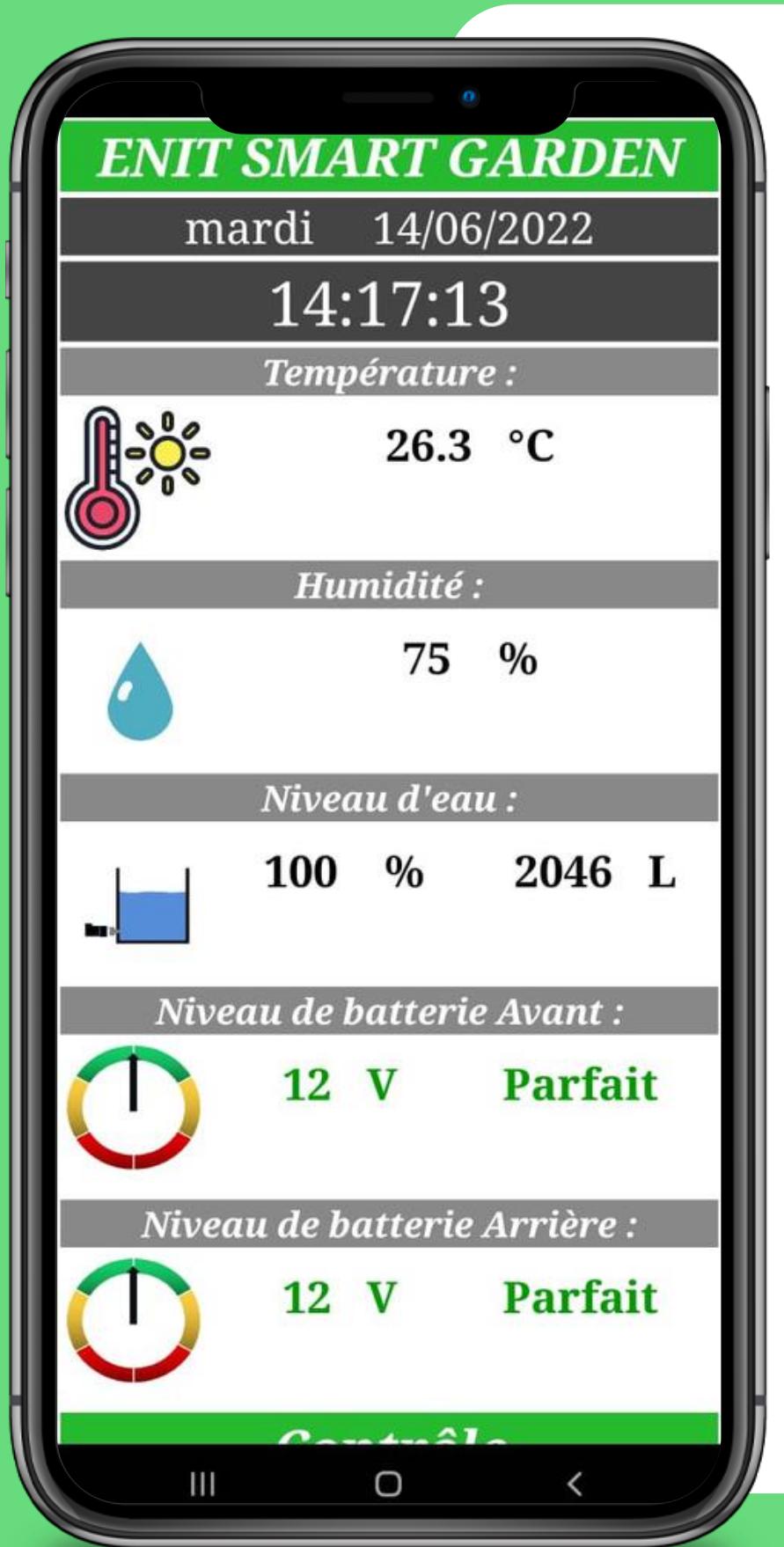
# Conclusion

**Ce projet de fin d'année nous a permis de mettre en pratique les connaissances théoriques et pratiques acquises durant notre formation à ISET RADES.**

**Dans ce projet nous avons eu la possibilité d'acquérir et d'approfondir des compétences en matière de la conception et la réalisation des cartes électroniques, la conception mécanique en 3D , la programmation des microcontrôleurs , le développement des applications mobiles et la communication en temps réel .**

**L'environnement du laboratoire dans lequel, nous faisons notre stage était très intéressant.**

**Ainsi le directeur et tous les membres de l'équipe que nous remercions se côtoient et sont aisément accessibles.**



Merci Pour  
Votre Attention