

Présentation du projet Jardin intelligent lot



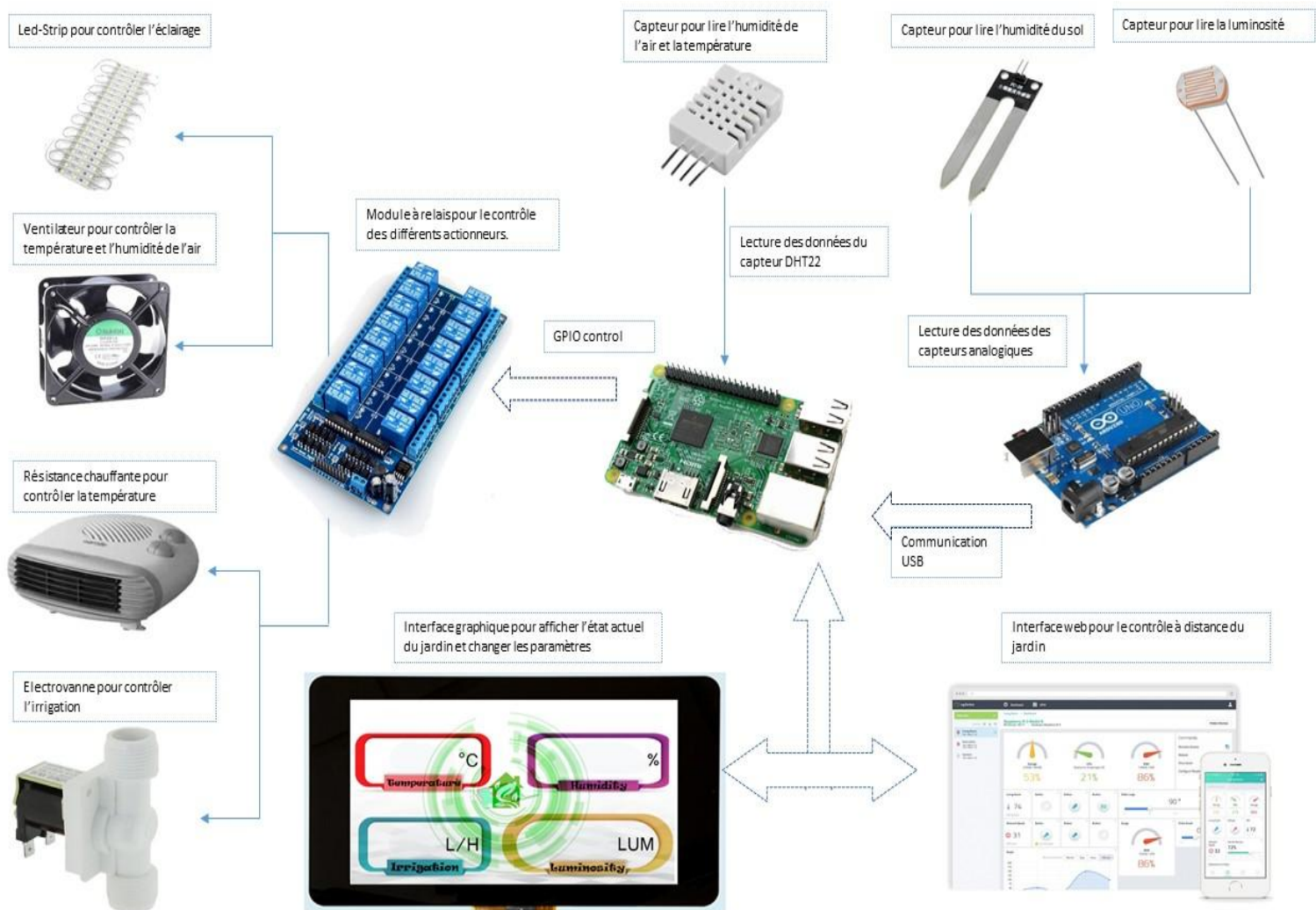
Elaboré par :

Meliane Dhia Eddine

Résumé

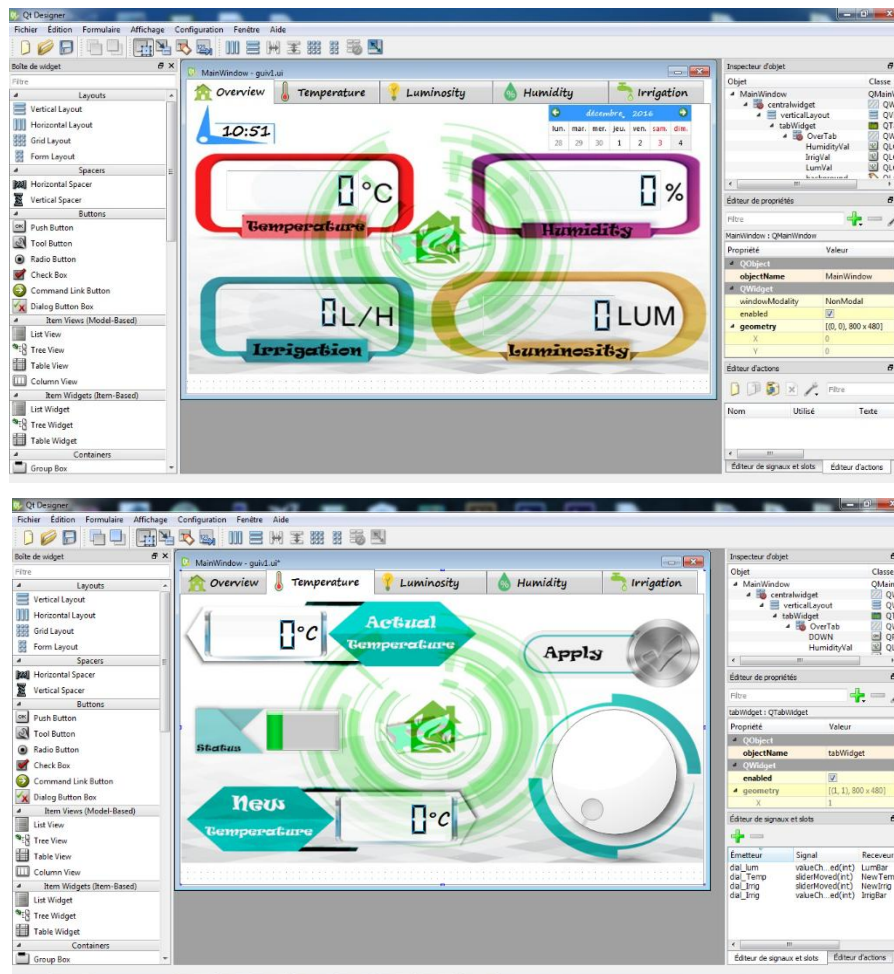
Le but de ce projet est de réaliser un jardin d'intérieur connecté à l'internet et équipé d'une interface graphique permettant de contrôler ses paramètres.

Principe de fonctionnement

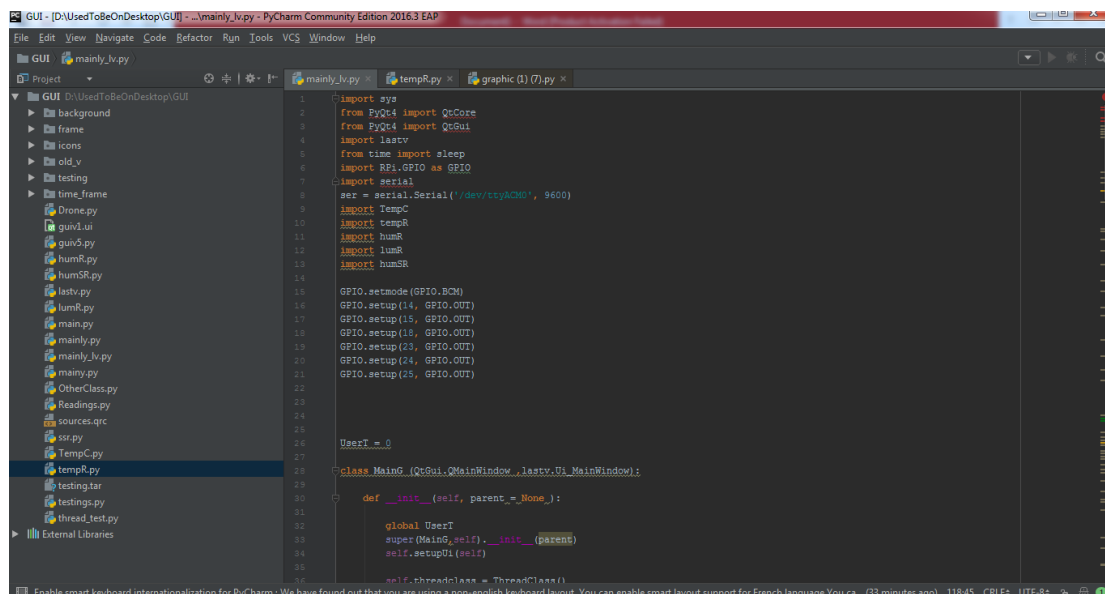


Réalisation du projet

1. Design de l'interface graphique sur Qt designer :



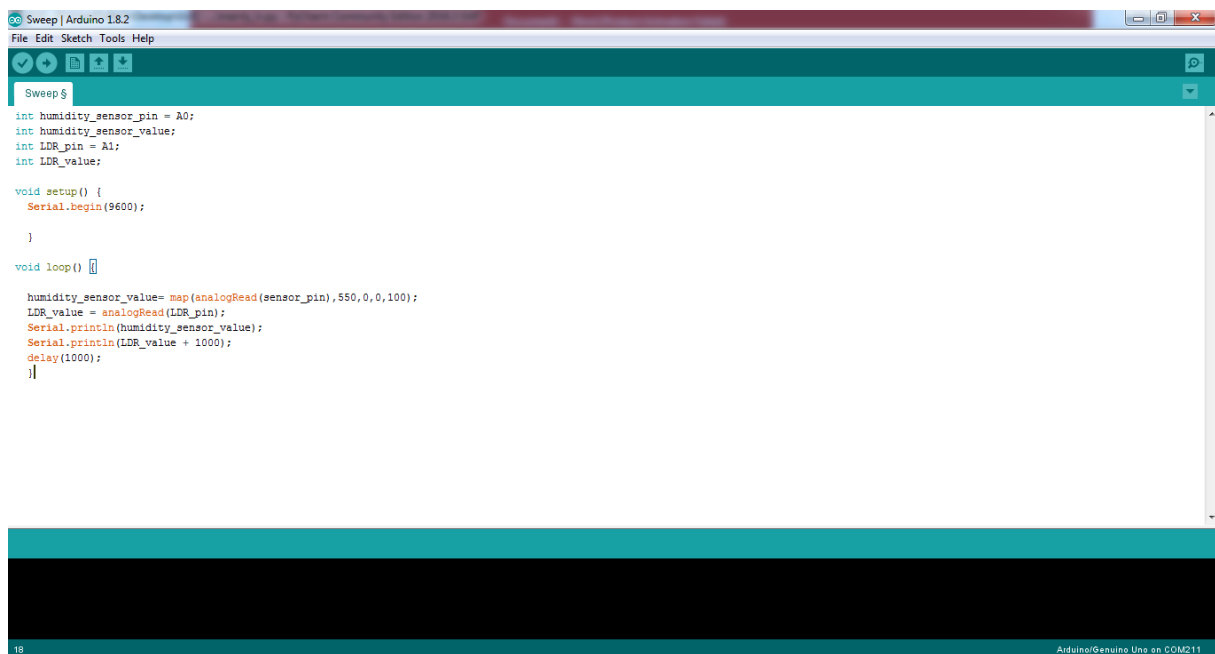
2. Développement du code en python en utilisant l'IDE pycharm :



Difficultés et solutions :

- J'ai essayé de diviser le code en modules, par exemple la lecture de température et l'envoi vers la base de données, le contrôle du chauffage ou refroidissement.
- Dans le code principal j'ai utilisé le Multithreading, pour me permettre d'assurer l'exécution de différentes tâches simultanément comme par exemple le contrôle de l'éclairage et le chauffage.
- L'interface graphique se plante parfois, après quelques lectures j'ai conclu que le Qthread , contient quelques problèmes de gestion de mémoire.

3. Développement du code sur l'IDE de Arduino :



```
Sweep | Arduino 1.8.2
File Edit Sketch Tools Help

Sweep $

int humidity_sensor_pin = A0;
int humidity_sensor_value;
int LDR_pin = A1;
int LDR_value;

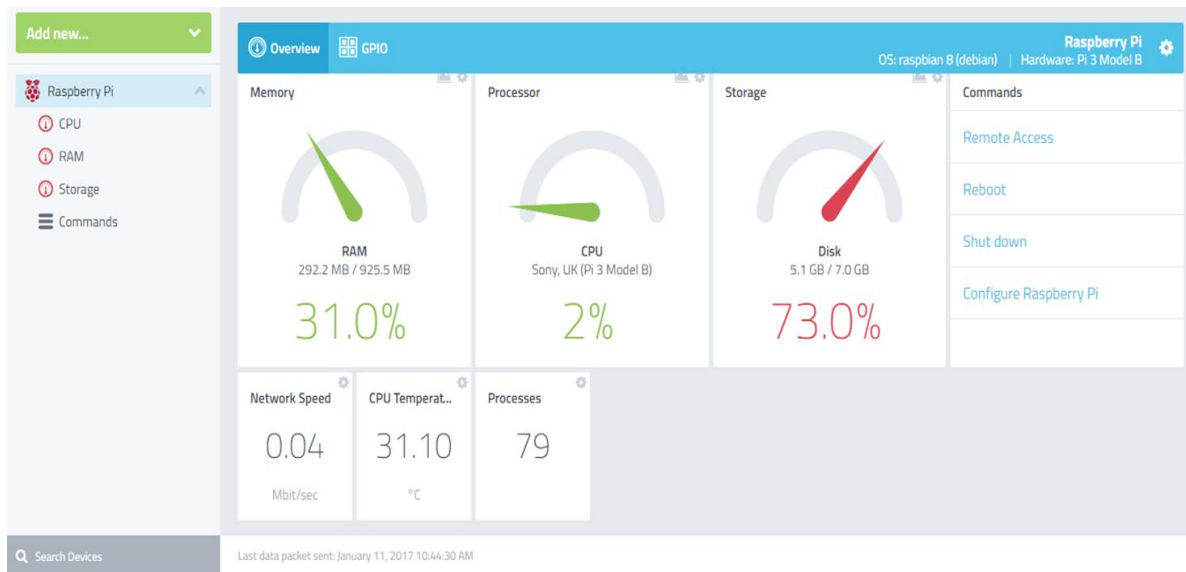
void setup() {
  Serial.begin(9600);
}

void loop() {
  humidity_sensor_value= map(analogRead(sensor_pin),550,0,0,100);
  LDR_value = analogRead(LDR_pin);
  Serial.println(humidity_sensor_value);
  Serial.println(LDR_value + 1000);
  delay(1000);
}
```

Difficultés et solutions :

- La communication entre la Raspberry pi et l'Arduino est à travers l'USB.
- La communication USB entre la Raspberry et l'Arduino se plante parfois et affiche un message d'erreur (" Buffer Overflow "), pour remédier à ça, j'ai ajouté un "delay " dans le sketch de l'Arduino pour limiter la fréquence d'envoi.

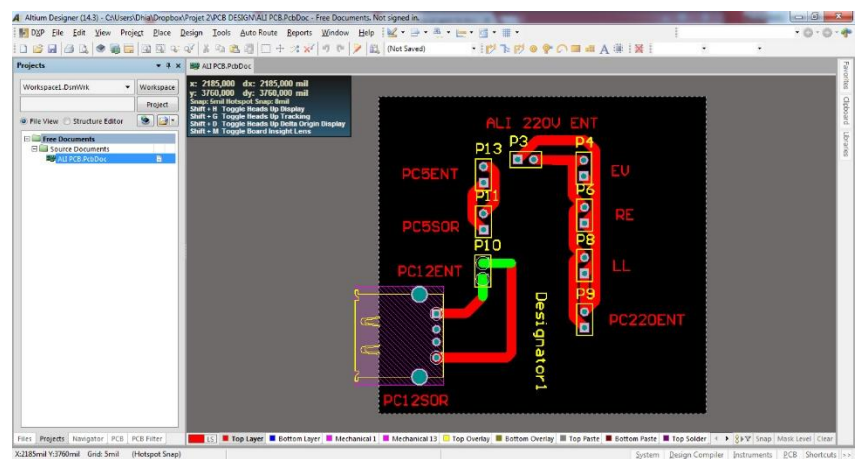
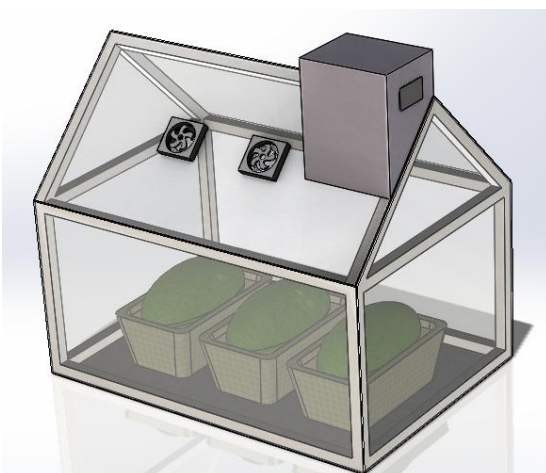
4. Utilisation de la plateforme Cayenne



Difficultés et solutions :

- Pour la partie de l'IoT j'ai tout d'abord créé un serveur WAMP local, dans mon Pc, étant dans le même réseau que la Raspberry pi , celle-ci lui envoie les données, mais cette solution s'est avérée limitée puisqu'on ne peut accéder au serveur que si on est connecté dans le même réseau local, d'où l'idée d'utiliser la plateforme IoT Cayenne .

5. Design des Cartes électriques pour faciliter la connexion des différents composants, et la conception du jardin sur SolidWorks.



Conclusion

Ce projet était l'un des projets les plus challengent que j'ai fait, en effet dans ce projet j'ai appris à utiliser la Raspberry pi, à utiliser Qt designer pour faire des interfaces graphiques et utiliser la technique de l'Iot.

J'ai bien voulu continuer à améliorer ce projet, mais faute de temps limité, j'ai été obligé d'arrêter le développement de ce projet et commencer un autre projet "Reverse engineering d'un drone".

Tutos et ressources consultés durant la réalisation du projet :

<https://diyhacking.com/best-raspberry-pi-iot-project/>

<https://www.raspberrypi.org/documentation/usage/python/>

<http://pyqt.sourceforge.net/Docs/PyQt4/designer.html>

<https://circuitdigest.com/microcontroller-projects/arduino-raspberry-pi-interfacing>

Enfin, je souhaite vous montrer mon atelier personnel, auquel je suis en train de travailler, sur le projet des mini-drones et leur intégration dans l'industrie 4.0.

