

## PROJET UF DEV : PLANTE CONNECTEE

CUARTIELLA THEO

DEBLAECKER JEREMY

## Table des matières

PROJET UF DEV : PLANTE CONNECTEE .....	1
Présentation de l'équipe : .....	3
Contexte et enjeux : .....	3
Description de la solution : .....	3
Fonctionnalités : .....	3
Environnement et méthode de travail : .....	4
Réalisation de la solution : .....	4
Dépôt GitHub : .....	5

### Présentation de l'équipe :

Notre équipe est composée de deux membres, Théo Cuartiella et Jérémy Deblaecker. En tant qu'équipe nous ferons en sorte de travailler en symbiose pour pouvoir effectuer notre travail le plus efficacement possible. Nos compétences nous permettront de travailler avec du langage C et du Python pour développer l'application. Nous disposons également de connaissance en SQL pour développer une base de données.

### Contexte et enjeux :

L'objectif avec cette plante connectée est de faciliter le plus possible la vie des propriétaires de plante, peu de personnes savent avec précision qu'elle quantité d'eau ou la chaleur nécessaire pour entretenir leurs végétaux. Notre société a changé, les gens sont plus connectés et plus occupés qu'avant, c'est pourquoi une plante connectée paraît nécessaire pour s'en occuper au mieux. Notre but est de pouvoir créer une plante connectée que nous pourrions gérer depuis notre application. L'application permettra notamment d'automatiser le plus possible notre plante pour passer le moins de temps possible à s'en occuper. Pour se faire nous utiliserons des capteurs pour suivre l'humidité et la température de la plante. Ces capteurs permettront d'alerter le propriétaire que la plante est devenue trop sèche ou que la température est trop (ou pas assez) élevée.

### Description de la solution :

Pour faire notre objet connecté nous avons dû utiliser plusieurs hardware : le module ESP8266 qui dispose d'une interface WiFi idéale pour les objets connectés, le DHT11 pour capter la température et l'humidité de l'air ainsi qu'un capteur d'humidité du sol.

Afin d'utiliser l'objet connecté nous devons créer un code sur Arduino. Une fois ces données récupérées par Arduino nous les envoyons sur Adafruit par wifi grâce au protocole MQTT qui permet de publier nos données.

Une fois les données reçues par Adafruit nous les envoyons sur notre application web (protocole http). Sur cette application l'utilisateur pourra consulter les données émises par l'objet connecté et pourra les comparer à celle requise pour sa plante. L'utilisateur pourra également choisir la plante qu'il désire consulter afin de connaître ses informations.

### Fonctionnalités :

Notre solution a donc comme fonctionnalités :

- Une application qui permettra de suivre l'état de la plante : si elle a trop chaud ou trop froid ou si elle devient trop sèche ou si elle devient trop humide. L'utilisateur pourra renseigner dans l'application l'espèce de sa plante ce qui lui donnera accès à la quantité de d'eau nécessaire pour le développement de la plante ainsi que la température idéale pour la plante ainsi que la l'exposition au soleil nécessaire à la plante.
- Un objet connecté qui contient :

- Un capteur de température qui aura pour but de suivre l'évolution de la température de la plante ce qui permettra au client de savoir s'il doit déplacer sa plante dans un endroit plus chaud au plus froid. De plus l'application permettra de donner des conseils sur l'endroit le plus adapté en fonction de l'espèce de la plante. Cette solution s'appuiera sur la mise en place de l'application. La mise en place de ce capteur est de son développement logiciel devrait prendre plusieurs semaines. La mise en place du capteur est la priorité absolue car sans les mesures l'application ne servirait à rien.
- Un capteur d'humidité, qui aura comme bénéfice de pouvoir renseigner à l'utilisateur la quantité d'eau que la plante a besoin pour s'épanouir. Le capteur renseignera sur l'application si la plante a besoin d'eau en fonction de l'espèce de la plante. La mise en place de ce capteur est de son développement logiciel devrait prendre plusieurs semaines. La mise en place du capteur est la priorité absolue car sans les mesures l'application ne servirait à rien.

### Environnement et méthode de travail :

Pour réaliser ce projet dans les temps impartis nous allons utiliser divers outils pour nous organiser et partager nos documents :

- Nous utiliserons GitHub pour travailler en même temps sur le même code.
- Pour partager nos documents (Word, Gantt...) entre nous, nous utiliserons un One Drive.
- Pour nous organiser plus rapidement sur des tâches plus courtes nous utiliserons également Trello.

### Réalisation de la solution :

Pour composer notre projet nous avons dû relier nos composants, nous les avons donc reliés comme ceci :

- Connecter le pin D4 de l'Esp 8266 au pin DATA du DHT11.
- Connecter le pin 3V de l'Esp 8266 au pin VCC du DHT11.
- Connecter le pin G de l'Esp 8266 au pin GND du DHT11
- Connecter le pin D1 de l'Esp 8266 au pin D0 du capteur d'humidité.
- Connecter le pin G de l'Esp 8266 au pin GND du capteur d'humidité.
- Connecter le pin 3V de l'Esp 8266 au pin VCC du capteur d'humidité.
- Connecter le pin A0 de l'Esp 8266 au pin A0 du capteur d'humidité.

Nous avons également créé un code Arduino (voir le code Arduino joint) qui permet de relever les données émises par les capteurs. Ses données sont ensuite envoyées vers Adafruit grâce au protocole MQTT. Pour que ses données soit transmises sur Adafruit nous avons créés trois « feeds » qui permettent de recevoir les données (un feed pour chaque donnée transmise), et un Dashboard qui rassemble ses informations.

Pour faciliter l'utilisation du client nous avons ensuite nous avons développés une application web, qui permet à l'utilisateur de voir les données émises par l'objet connecté ainsi que les données optimales pour la pousse de la plante. L'utilisateur pourra également ajouter

L'utilisateur peut depuis cette application ajouter (ou modifier) des plantes. Pour se faire il lui suffit de renseigner les informations nécessaires (nom de la plante, humidité optimale, température optimal...).

Dépôt GitHub :

[https://github.com/jeremydeblaecker/PROJET\\_UF\\_DEV](https://github.com/jeremydeblaecker/PROJET_UF_DEV)