



---

# PLANTE CONNECTEE DOSSIER TECHNIQUE

---

Fougerolle Gabriel, Requena Lucas et Tabbara Louna



## Table des matières

Capteurs & Raspberry PI .....	3
Base de données .....	4
Application (web-app) .....	5
Synthèse .....	6
Annexe 1.....	7
Algorithme.....	7
Programme.....	7
Annexe 2.....	8
MCD.....	8

## Capteurs & Raspberry Pi

Pour la réalisation de ce projet, il était nécessaire de trouver un moyen de mesurer la température de la plante, l'humidité de sa terre et la lumière reçue. Après réflexion, nous avons choisi le matériel suivant :

- Une **Raspberry 3B+** (nano-ordinateur)
- Des **capteurs GROVE** :
  - Un capteur **photosensible**
  - Un capteur d'**humidité**
  - Un capteur de **température**

La Raspberry étant équipée d'un shield, le branchement des capteurs se fait donc sur ce dernier, directement sur trois entrées analogiques.

Afin de pouvoir programmer les capteurs et gérer les envois à la base de données, nous avons installé un système d'exploitation sur la Raspberry : Raspbian, gratuit, basé sur GNU/Linux. Comme Grove propose des programmes réalisés en python pour chacun de leurs capteurs, nous y avons installé python et la bibliothèque de Grove avec tous leurs programmes. Après les avoir lus, nous avons combiné ces programmes pour réaliser le programme et son algorithme en *Annexe 1*.

## Base de données

Afin d'enregistrer toutes les données des plantes, il était nécessaire de réaliser une base de données. Nous avons choisi une base en SQL, pour des raisons de connaissances et de simplicité (MySQL/MariaDB). Nous avons donc réalisé cette base de données à l'aide du MCD de l'*Annexe 2*.

## Application (web-app)

Nous avons aussi besoin d'une application, une interface depuis laquelle un utilisateur pourrait avoir accès à sa plante, voir ses informations, ajouter de nouvelles plantes etc. Nous avons décidé de réaliser une web-app, utilisant HTML, CSS, Javascript en Front-end et PHP en Back-end.

La web-app inclue donc les fonctionnalités suivantes :

- Inscription et connexion
- Ajout de plantes au compte
- Gestion des plantes
- Ajout de commentaires sur les types de plantes
- Votes pour les commentaires

Pour des raisons explicites de simplicité et de prix, nous avons également hébergé cette web-app sur la Raspberry. Pour cela nous avons installé apache2, php, mysql. Ainsi, nous avons pu placer le contenu de la web-app dans le répertoire `/var/www/html` et y accéder depuis l'IP de la Raspberry sur un réseau wifi privé (Partage de connexion du téléphone par exemple). De plus, nous avons créé un utilisateur pour la base de données.

## Synthèse

La web-app est prévue pour être déployée en ligne et hébergée en ligne, de même que la base de données. La Raspberry sera en réalité un petit appareil miniaturisé équipé des trois capteurs, envoyant des données avec identifiant permettant de reconnaître la plante. L'utilisateur pourrait ainsi ajouter chaque plante en se connectant à cet appareil en même temps.

## Annexe 1

### Algorithme

Connexion à la base de données  
Définition des entrées analogiques  
**humidité** = récupérer la valeur  
**luminosité** = récupérer la valeur  
**temperature** = récupérer la valeur  
Envoyer sur la base de données les trois valeurs pour un id précis (celui de la plante)

### Programme

```
#!/usr/bin/python

import time
import mysql.connector
from grove.factory import Factory
from grove.grove_light_sensor_v1_2 import GroveLightSensor
from grove.grove_moisture_sensor import GroveMoistureSensor

# connection temperature slot 4
PINT = 4
# connection lumiere slot 2
PINl = 0
# connection humidity slot 6
PINh = 6

mydb = mysql.connector.connect(
    host="localhost",
    user="toor",
    password="toor",
    database="theconnectedflower"
)
mycursor = mydb.cursor()
if mydb is not None:
    temp1 = Factory.getTemper("NTC-ADC", PINT)
    temp = temp1.temperature
    hum = GroveMoistureSensor(PINh)
    humi = hum/7
    lum = GroveLightSensor(PINl)
    humi = hum.moisture
    lumi = lum.light

    up_temp = "UPDATE plants SET temperature = %s WHERE user_id = %s AND plant_id = %s"
    up_hum = "UPDATE plants SET humidity = %s WHERE user_id = %s AND plant_id = %s"
    up_lum = "UPDATE plants SET light = %s WHERE user_id = %s AND plant_id = %s"
    mycursor.execute(up_temp, (temp, 1, 1))
    mycursor.execute(up_hum, (humi, 1, 1))
    mycursor.execute(up_lum, (lumi, 1, 1))
    mydb.commit()
```

## Annexe 2

### MCD

