Rapport de stage de deuxième année 2023/2024 :

Développement d'un banc d'essai pour étudier les caractéristiques mécaniques de la plante basilic et capucine

Etudiant : BIN SHARIFFUDIN Danish Aiman

Tuteur Entreprise : Loïc TADRIST

Tuteur Universitaire : Eric JEANCELME

Durée de stage : 15/04/2024 – 21/06/2024

Lieu de stage : Laboratoire ISM

Adresse : 413 Avenue Gaston Berger, 13090, Aix-en-Provence

Site : IUT Aix-Marseille

Département : Génie Mécanique et Productique

Adresse : 413 Avenue Gaston Berger, 13090, Aix-en-Provence

FICHE D'IDENTITE ET DE CONFIDENTIALITE

Année 2024

STAGIAIRE : BIN SHARIFFUDIN Danish Aiman

TITRE : Stage en mécatronique

RESUME : Développer un banc d'essai pour étudier les caracteristiques de plantes.

Nombre de pages : 36 **Nombre de références bibliographiques :** 6

ENTREPRISE : Laboratoire ISM

Président directeur général : Jean Marc LINARES

Nombre d'employés :

Domaine d'activité : Mécanique et biomécanique

Adresse : 413 Av. Gaston Berger, 13090, Aix-en-Provence

Téléphone : **Télécopie :**

Tuteur industriel : Loïc TADRIST

Fonction : Professeur à l'IUT AIX-MARSEILLE

Mail : loic.tadrist@univ-amu.fr **Téléphone :** +33 4 13 94 63 93

PARTIE A REMPLIR PAR LE TUTEUR INDUSTRIEL

Accessibilité de ce rapport (entourer la mention correcte) :

LIBRE **CONFIDENTIEL pendant ..... an(s)**

Date : 18/06/2024 **Nom du stagiaire :** BIN SHARIFFUDIN **Signature :**

REMERCIEMENTS

Tout d'abord, j'aimerais exprimer ma gratitude à M. Loïc TADRIST qui m'a accueilli et proposé un stage au sein de laboratoire Institut des Sciences du Mouvement (ISM).

Je remercie également les chercheurs, Paul LACORRE et Louison FIORE qui ont participé au bon déroulement du stage et m'ont enseigné plein de nouvelles choses que je n'ai jamais fait lors de mes deux années à l'IUT.

Je tiens aussi à remercier l'ensemble de l'équipe du laboratoire ISM qui m'a beaucoup aidé à résoudre de nombreux problèmes auxquels j'ai été confronté pendant le stage.

Je souhaite adresser mes remerciements à M. Nicolas SANCHEZ, responsable des stages à l'IUT qui m'a assisté lors de mes recherches de stage et donné l'autorisation pour réaliser le stage au sein de l'IUT.

Finalement, je n'oublierai jamais mon tuteur universitaire M. Eric JEANCELME et les intervenants pédagogiques du département Génie Mécanique et Productique (GMP) à l'IUT Aix-Marseille.

SOMMAIRE

1. INTRODUCTION

Lors de mon stage, j'ai eu l'opportunité de travailler au sein de l'Institut des Sciences du Mouvement (ISM) Etienne-Jules Marey, une unité mixte de recherche (UMR 7287) associée à Aix-Marseille Université. L'ISM, basé à Marseille, se distingue par son approche polyvalente de l'analyse du mouvement humain et animal, intégrant des domaines tels que la biomécanique et la robotique bio-inspirée.

Dans le cadre de mes études de BUT Génie Mécanique et Productique à l'IUT d'Aix-En-Provence, je suis censé faire un stage de deuxième année de 10 semaines afin de valider le semestre 4 et obtenir mon diplôme. À la suite de la difficulté de trouver un stage en entreprise et après une discussion avec M. Nicolas SANCHEZ, je l'ai réalisé à l'IUT pour une durée de 8 semaines.

J'ai eu de la chance de trouver un stage proposé par M. Loïc TADRIST. Le sujet de stage porte sur le développement d'un banc d'essais pour étudier les caractéristiques mécaniques des plantes basilics et capucine. Je ferai le stage donc dans le laboratoire ISM.

L'opportunité de traiter ce sujet m'a offert une expérience unique et enrichissante. En effet, j'ai pu explorer le domaine de la programmation de manière plus approfondie. Grâce aux tâches réalisées lors de ce stage, j'ai pu consolider et élargir les connaissances acquises durant mes deux années de formation, tant en programmation qu'en mécatronique.

Ce rapport de stage présente une analyse détaillée de mes activités au sein de l'ISM. Il met en lumière les diverses facettes de la recherche sur le mouvement et les technologies avancées utilisées pour mener à bien ces études. De plus, il illustre l'importance des collaborations interdisciplinaires et des partenariats industriels dans le développement de nouvelles technologies et solutions pratiques.

2. PRESENTATION DE L'ENTREPRISE

2.1. LES EQUIPES DE RECHERCES

Le laboratoire est composé de 3 groupes de recherches. Chaque équipe a ses propres fonctions et missions dans lesquelles elle se spécialise. Voici les équipes :

a) L'équipe BioMécanique/bioIngénierie (BMI)
Cette equipe se focalise sur
b) L'équipe Dynamiques Comportementales & Cognition (DynamiCC)
c) L'équipe Systèmes Bio Inspirés (SBI)

Parmi les trois groupes, je fais partie de l'équipe de SBI. Cette équipe a pour vocation de repousser les frontières de la connaissance scientifique sur le vivant afin d'inspirer l'innovation en ingénierie. Elle s'articule également sur deux axes de recherches, biorobotique et mécanismes bio-inspirés. Mon sujet de stage est basé sur l'axe mécanismes bio-inspirés qui est dirigé par M. Julien CHAVES-JACOB.

Cet axe de recherche se focalise sur l'élaboration de nouveaux algorithmes de conception destinés à l'industrie du futur, spécifiquement dans les domaines de la mécanique et de la biomécanique.

Les travaux de recherche de l'axe Mécanismes bio-inspirés sont orientés autour de trois thématiques de recherche :

- Structure de pièces bio-inspirée des os longs,
- Liaisons mécaniques bio-inspirées de la morphogenèse,
- Actionneurs et dissipateurs bio-Inspirés.

2.2. IMPLANTATION DU LABORATOIRE

Le laboratoire ISM se trouve à Avenue de Luminy F-13288, Marseille. D'ailleurs, il porte une antenne à Aix-en-Provence. Elle est précisément s'est intégrée à l'IUT Aix-Marseille dans le département de Génie Mécanique et Productique (GMP).

Figure 1 : L'IUT Aix-Marseille

3. SUJET DU STAGE

Sachant que le stage est établi dans un laboratoire de ISM, M. Loïc TADRIST m'a proposé un sujet étant basée sur mécatronique, programmation et la biologie. Le sujet est donc l'étude de développement d'un banc d'essai pour étudier les caractéristiques de la plante basilic et capucine. Elles ont été choisies parce que les plantes ne peuvent refleurir après flétrissement qu'après arrosage.

3.1. BETE A CORNE

Durant le stage, j'ai travaillé sur la mise en œuvre de l'expérience et récupérer les photos et les données des plantes. Ceci a pour objectif de faciliter la suite de travail qui vont être fait par des chercheurs ou d'autres stagiaires.

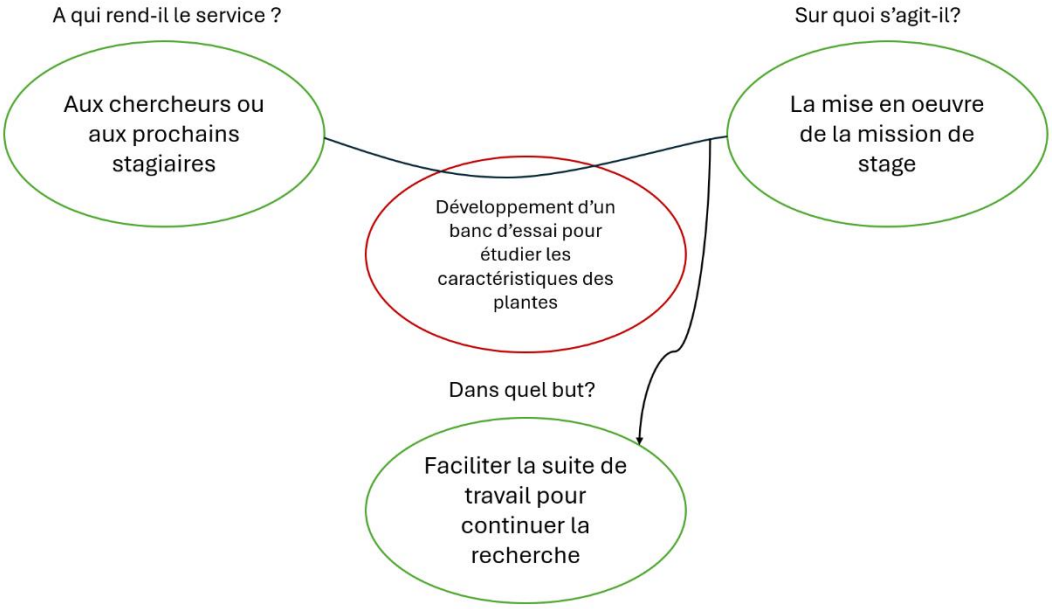


Figure 2 : Diagramme de bête à corne

3.2. CAHIER DES CHARGES

Afin de définir les buts et les objectifs de ce stage, j'ai réalisé un diagramme de pieuvre me permettant représenter le travail que j'ai fait lors du stage avec son environnement. Ce diagramme met en lumière les fonctions de service d'un produit.

Ainsi, j'ai décidé à mettre ce diagramme dans ce rapport car il pourrait m'aider à construire un tableau de cahier des charges. Ce qui est également utile pour la planification et organisation du travail. Grâce à ça, j'étais capable à prioriser mes tâches et avoir une vraie idée du produit final de ma mission de stage.

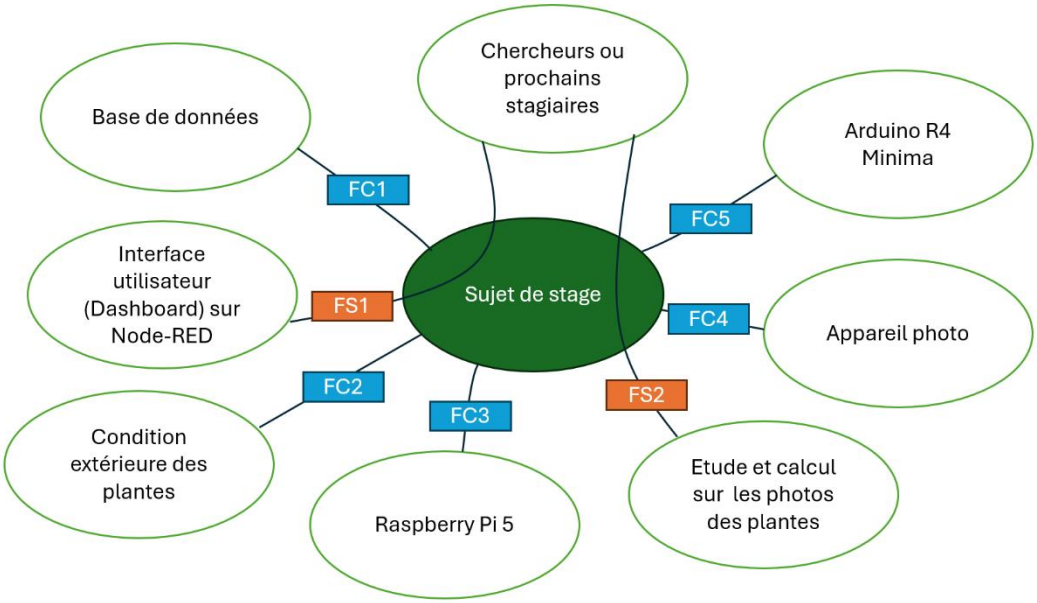


Figure 3 : Diagramme de pieuvre

Le diagramme se divise sous deux parties, FS et FC. FS signifie la fonction services tant que FC est la fonction contrainte. La fonction service permet de définir les services auxquels ils pourraient être utile pour le client ou dans notre cas, les chercheurs et les prochains stagiaires. D'un autre côté, la fonction contrainte représente les exigences auxquelles le produit doit se conformer.

Voici l'explication plus détaillé pour chaque fonction :

FS1 : Les chercheurs doivent pouvoir contrôler les actions dans les flux en utilisant l'interface utilisateur sur Node-RED.

FS2 : Les prochains stagiaires permettent d'avancer sur ce sujet au futur en faisant les études et les calculs sur les photos des plantes.

FC1 : Les données des capteurs peuvent être stockées grâce à la base de données de Raspberry Pi 5 et un disque dur.

FC2 : Les lectures des données peuvent être variées avec la mise en œuvre de la condition extérieure étant manipulable.

FC3 : La séance des photos permet de s'automatiser avec la programmation faite sur Raspberry Pi 5.

FC4 : La condition des plantes et les études sur elles peuvent être observée grâce aux appareils photo installés.

FC5 : L'acquisition des données serait possible et facile avec les capteurs qui sont programmés par l'Arduino R4 Minima.

	Fonction	Critère	Niveau	Flexibilité
FS1	Permettre aux chercheurs d'utiliser l'interface Node-RED.	Programmation des flux sur Node-RED.	Impératif	F0
FS2	Permettre aux prochains stagiaires de faire les études et calculs sur les plantes.	La mise en œuvre de plantes dans la boîte de lumière.	Souhaitable	F1
FC1	Doit pouvoir stocker les données dans la base de données	Utilisation de Raspberry Pi 5 et le disque dur.	Impératif	F0
FC2	Doit être installé pour donner une condition étant manipulable.	Installation de ventilateur avec deux tubes qui se collent.	Souhaitable	F2
FC3	Doit pouvoir automatiser les travaux	Programmation en python pour la séance des photos.	Impératif	F0
FC4	Doit pouvoir prendre des photos de plantes	Installation de l'appareil photo.	Impératif	F0
FC5	Doit pouvoir connecter avec les capteurs pour collecter les données.	Utilisation de l'Arduino R4 Minima	Impératif	F0

Figure 4 : Tableau de cahier des charges

D'après le tableau de cahier des charges fait à la page précédente, j'ai deux fonctions qui sont souhaitable. Ceci a du fait que ces deux fonctions sont moins importantes pour avancer la mise en œuvre du banc d'essai. A noter que, c'est un nouveau sujet qui venait de commencer. Ainsi, j'ai dû commencer depuis rien.

La flexibilité met en lumière la difficulté à le réaliser et l'importance de fonction dans le travail. Selon le tableau, nous pouvons déduire que chaque tâche est primordiale pour l'avancement de stage. Cela peut augmenter les risques qui pourraient retarder le travail.

3.3. SWOT

Afin de les résoudre, j'ai construit un diagramme de SWOT (AFOM en français) qui est utile pour définir les risques interne et externe qui peuvent se produire.

	Helpful	Harmful
Internal Origin	***STRENGTHS*** • Matériels • Interdisciplinarité • Bases de la programmation	***WEAKNESS*** • Manque d'expériences • Connaissances de nouveaux langages de programmation
External Origin	***OPPORTUNITY*** • Recherche et Innovation • Nouvelles compétences • Partenariats industriels	***THREATS*** • Durée de stage • Performance de certains logiciels

Figure 5 : Le diagramme de SWOT

En conclusion, ce stage présente des opportunités significatives pour l'innovation et la recherche, mais aussi des défis techniques importants à surmonter. Une stratégie bien planifiée et des collaborations étroites avec des partenaires industriels seront cruciales pour le succès du projet.

3.4. LE JALONNEMENT DE STAGE

Afin de présenter le flux du travail que j'ai fait lors du stage, j'ai réalisé un jalon. Cet outil de gestion permet de faire sortir les objectifs que j'ai besoin de compléter et chronologiser les tâches. Voici le jalon qui pourrait évoquer en bref les démarches qui sont réalisés :

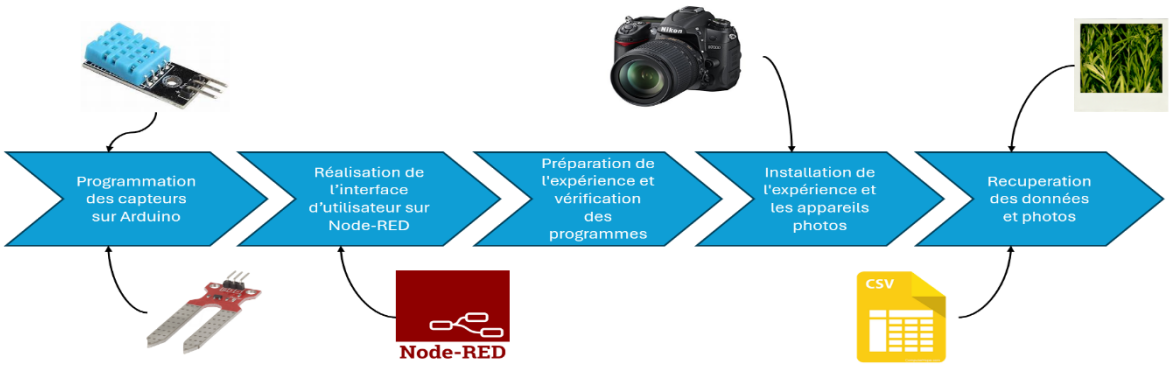


Figure 6 : Le jalonnement de démarches faits

4. RAPPORT TECHNIQUE

4.1. MISE EN ŒUVRE de L'EXPERIENCE

Afin de réaliser l'expérience, la mise en œuvre a été faite. Durant mon stage, j'ai utilisé deux types de microcontrôleur s'appelant Raspberry Pi et Arduino R4 Minima. Ceux-ci me permettaient de travailler avec des capteurs qui vont me servir à avoir les données d'atmosphérique ainsi que les données des plantes.

4.1.1. Arduino R4 Minima

Le microcontrôleur Arduino R4 Minima est vraiment utile pour mon projet d'étudier les caractéristiques mécaniques. Voici les raisons pourquoi il m'a facilité mes taches pendant le stage :

Figure 7 : Arduino R4 Minima

a) Comptabilité avec beaucoup type de capteur

R4 Minima est le modèle d'Arduino le plus récent. Ainsi, il est vraiment compatible avec les capteurs y compris DHT 11 et MAX31865.

b) Facile à programmer

Le logiciel programmation d'Arduino utilise un environnement base sur le langage de C/C++ ce qui est facile et simple à utiliser. Il existe aussi une variété de bibliothèque des capteurs me permettant à avoir une référence du codage sur les exemples donnés.

c) Flexibilité

R4 Minima possède plusieurs ports d'extensions comme le I2C. Sur le microcontrôleur, il y'a un port de « SCL » et « SDA » ils même. Normalement, il faudrait connecter avec A4 et A5 (pour le modèle de Nano) en tant qu'un remplacement de SCL et SDA. Grace a ça, je pourrais connecter plusieurs capteurs dans le même microcontrôleur sans besoin d'en ajouter encore un.

d) Capacités de mémoire améliorées

Comparé aux versions précédentes, l'Arduino R4 Minima possède un microcontrôleur plus performant avec une fréquence d'horloge plus élevée et plus de mémoire. Cela peut être bénéfique pour le traitement des données en temps réel et pour exécuter des algorithmes plus complexes.

Figure 8 : Raspberry Pi 5

Raspberry Pi 5

D'un autre côté, le Raspberry Pi 5 dans mon projet a pour but d'afficher l'interface de données sur un autre écran. Voici comment ce microcontrôleur m'a aidé le travail :

a) Connectivité améliorée

Le microcontrôleur est équipé avec deux ports de USB 3.0, offrant des vitesses de transfert de données plus rapides.

b) Capacités graphiques

Avec un GPU performant, il peut gérer des taches graphiques et des images de haute qualité. Cela pourrait me faciliter d'analyser les plantes pour identifier les anomalies.

c) Ports GPIO

Le Raspberry Pi 5 dispose de nombreux ports de GPIO (General Purpose input/output) jouant à peu près le même rôle comme l'Arduino R4 Minima. Le langage de la programmation est python ce qui est utilisé par la majorité des programmeurs a l'instant.

d) Flexibilité

Ce microcontrôleur me permet d'utiliser pleines de fonctionnalités sans besoin d'utiliser un capteur. Par exemple, afin d'avoir l'horodatage sur Arduino, il me faut connecter un outil qui s'appelle « RTC Module ». A l'inverse, je n'en ai pas besoin un avec Raspberry Pi 5. Je pourrais coder la fonction d'horodatage toute suite sur python ou Node-Red.

Pour résumé, les deux microcontrôleurs sont très utiles pour l'avancement de ce stage. Chacun a ses avantages et ses inconvénients. Ce qui est similaire de ces deux capteurs, c'est leur communauté de support étant robuste. Je peux facilement trouver les tutoriels et les solutions dans leur forum.

4.2. LES CAPTEURS UTILISES

Les capteurs sont utilisés afin d'analyser les données. Dans cette partie, je vais expliquer les codages et les câblages qui ont été faits. Grâce aux tutoriels, j'ai facilement trouvé les exemples du codage et du câblage des capteurs sur « GitHub ». Voici les capteurs que j'ai utilisés étant connectés avec Arduino R4 Minima ainsi que Raspberry Pi 5 :

4.2.1. DHT 11

Le capteur DHT 11 est utilisé pour détecter la température et l'humidité de l'atmosphère. Avec 1 seconde de temps de réponse, ce capteur me permet d'avoir une donnée rapide.

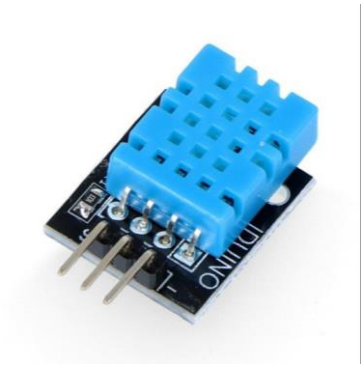


Figure 9 : DHT 11

Câblages de DHT 11

- (+) → 3.3V/5V
- (-) → GND (Ground/Terre)
- S (Supply) → D2 (n'importe quel « Digital Pin »)

Programmation du DHT 11

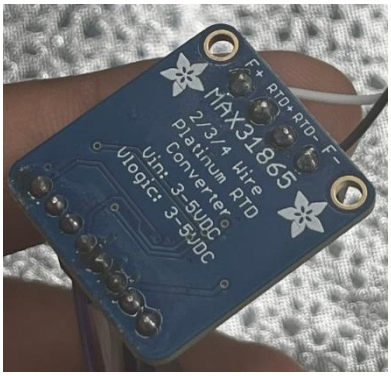
```
#include <dht.h>  // Inclure la bibliothèque du capteur DHT

DHT dht(2, DHT11);  // Definir le digital pin et le type de capteur

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

void loop() {
  delay(10)
  float t = dht.readTemperature(); // Lire les donnees de la température
  delay(10)
  float h = dht.readHumidity(); // Lire les donnees de l'humidité

  Serial.print("Temperature = ");  // Afficher le label pour la température
  Serial.println(t);  // Afficher la température en degrés Celsius

  Serial.print("Humidity = ");  // Afficher le label pour l'humidité
  Serial.println(h);  // Afficher l'humidité en pourcentage

  delay(2000);  // Attendre 2 secondes avant la prochaine lecture
}
```

Figure 10 : Programmation de DHT 11

4.2.2. MAX 31865

Figure 11 : MAX 31865

MAX 31865 est un convertisseur RTD (Resistance Temperature Detector) vers numérique. Il peut fonctionner avec le capteur PT 100 ou PT 1000. MAX 31865 facilite l'interfaçage en convertissant la valeur de résistance du capteur en valeur numérique de la température. Avant de brancher à l'Arduino ou au PT 100, il faut souder les connections sur le convertisseur. Il y a aussi un endroit spécifique à souder afin de connecter 2 fils du PT100.



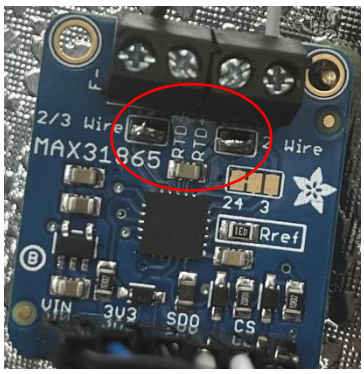


Figure 12 et Figure 13 : Les endroits à souder

Câblage de MAX 31865

- VIN → 5V
- GND → GND
- CS → D10
- SDI → D11
- SDO → D12
- SCK → D13

Programmation du convertisseur MAX 31865

Afin de lire les données du MAX 31865, j'ai utilisé la bibliothèque qui s'appelle « Adafruit_MAX31865.h ». Le code commence par définir le pin de CS étant D10. Cela permet à utiliser des broches par défaut du SPI (SDI(D11), SDO(D12) et SCK(D13)). Ainsi, je n'ai pas besoin de définir tous les pins.

La valeur de RREF est la valeur référentielle pour calculer la résistance. Dans le « setup », j'ai mis « thermo.begin(MAX31865_2WIRE) » car seulement deux fils du capteur PT100 sont connectés avec le convertisseur. La valeur numérique de température est calculée en utilisant la valeur de résistance.

Max_31865.ino

```ino
#include <Adafruit_MAX31865.h>
// Initialisation de l'objet MAX31865
Adafruit_MAX31865 thermo = Adafruit_MAX31865(10);
#define RREF 430.0
#define RNOMINAL 100.0 //La résistance à 0 degré celsius

void setup() {
  Serial.begin(9600);
  // Initialiser la communication SPI
  Serial.println("Adafruit MAX31865 PT100 Sensor Test!");
  thermo.begin(MAX31865_2WIRE);  // Modifier selon le type de connexion : 2WIRE, 3WIRE, 4WIRE
}

void loop() {
  // Lire la résistance du RTD
    uint16_t rtd = thermo.readRTD();
    float ratio = rtd;
    ratio /= 32768;
    ratio = ratio;
    float res = RREF*ratio;
  Serial.print("Resistance: ");
  Serial.println(res);
  // Calculer la température en utilisant la formule standard pour PT100/PT1000
  float temp = thermo.temperature(RNOMINAL, RREF);
  Serial.print("Temperature: ");
  Serial.println(temp);
  delay(1000);  // Pause de 1 seconde entre chaque lecture
}
```

Figure 14 : Programmation de MAX31865

4.2.3. Capteur d'humidité du sol

Ce capteur me servit à collecter les données de l'humidité du sol. En plus, il a pour vocation de me donner une indication pour l'arrosage. J'ai installé quatre capteurs afin de les analyser simultanément. Les quatre pots qui ont été installé sont deux pots basilics, un grand pot de basilic et un grand pot de capucine.

Câblage du 4 capteurs d'humidité du sol

- (+) → 5V
- (-) → GND
- (S) → A0, A1, A2, A3

A noter qu'il ne faut pas utiliser A4 et A5 car ils sont les pins de SCL et SDA.

Programmation du capteur

Dans cette programmation, les données du capteur sont en résistance (en ohm). Ainsi, les données doivent être converties aux percentage. La valeur maximum de résistance définie est 700 Ω. Elle n'est pas exactement la valeur max mais elle est la valeur que nous ne pouvons pas atteindre pourtant la plus proche. Nous pourrions donc obtenir une valeur d'estimation de l'humidité.

```
#define AOUT_PIN A0 // Definir les digital pin pour chaque capteur
#define AOUT_PIN A1
#define AOUT_PIN A2
#define AOUT_PIN A3
void setup() {
  Serial.begin(9600);
}
void loop() {
    int h_pot1 = analogRead(AOUT_PIN1); // Lire les donnees du capteur
    int h_pot2 = analogRead(AOUT_PIN2);
    int h_pot3 = analogRead(AOUT_PIN3);
    int h_pot4 = analogRead(AOUT_PIN4);

    int ph1 = ((h_pot1*100)/700); // Convertir la valeur de l'humidite en resistance au percentage
    int ph2 = ((h_pot2*100)/700);// (700 est la valeur max de la resistance dans notre cas normalement)
    int ph3 = ((h_pot3*100)/700);
    int ph4 = ((h_pot4*100)/700);

    Serial("humidity 1 = "); // Afficher les donnees
    Serial.println(ph1);
    Serial("humidity 2 = "); // Afficher les donnees
    Serial.println(ph2);
    Serial("humidity 3 = "); // Afficher les donnees
    Serial.println(ph3);
    Serial("humidity 4 = "); // Afficher les donnees
    Serial.println(ph4);
    delay(2000); // Attendre 2s avant la prochaine lecture
}
```

Figure 15 : Programmation du capteur de l'humidité du sol

Figure 16 : M5 K Mètre

4.2.4. M5 K mètre (Psychromètre)

M5 K mètre est un module de M5Stack permettant de mesurer la température à l'aide d'un thermocouple. Il est conçu à s'interfacer facilement et le système est basé par le microcontrôleur ESP32.

Figure 17 : Les deux tubes soudés

Deux de ces modules ont été installés afin de donner une condition externe. Ils sont installés avec deux tube soudés entre eux et un ventilateur d'un ordinateur est mis sur les deux tubes. Les deux tubes sont percés et un des deux a été scié en carré pour mettre un coton mouillé.

Câblage du module

- (+) → 5V
- (-) → GND
- SCL → SCL
- SDA → SDA

Programmation du module

A noter que dans cette programmation, il faut changer l'adresse pour le deuxième capteur pour qu'il soit unique. L'adresse du capteur par défaut est « 0x66 ». Nous pouvons le convertir en utilisant « sensor2.ChangeAddr ». Nous avons besoin de compiler pour seulement une fois et puis le supprimer et réécrire.

```
#include <M5_KMeter.h> // Bibliotheque de M5 Kmetre
M5_KMeter sensor; // Definir les variables des capteurs utilises
M5_KMeter sensor2;
void setup() {
  Serial.begin(9600);
  Wire.begin();
  Sensor.begin(&Wire, 0x66); // Definir l'adresse de I2C
  sensor2.begin(&Wire, 0x60);
}

void loop() {
    sensor.update();
    float tk = sensor.getTemperature(); // Lire les donnees
    float internaltemp = sensor.getInternalTemp();
    sensor2.update();
    float tk2 = sensor2.getTemperature(); // Lire les donnees
    float internaltemp2 = sensor2.getInternalTemp();
    float tempk = tk; // Afficher les donnees
    float tempk2 = tk2;
    delay(2000); // Attendre 2s avant la prochaine lecture
}
```

Figure 18 : Programmation de M5 K Mètre

4.2.5. Ventilateur

Le ventilateur du pc n'est pas un capteur mais il est connecté avec une alimentation de 12V et un relais pour qu'il puisse fonctionner avec un bouton sur le Dashboard de Node-RED. J'avais besoin d'utiliser l'alimentation de 12V car l'Arduino R4 Minima ne fournit que 5V au maximum. J'ai utilisé également une prise coupée pour la connecter avec l'alimentation.

Figure 8 : Ventilateur de PC

Figure 19 : Alimentation 12V

Figure 20 : Prise coupée

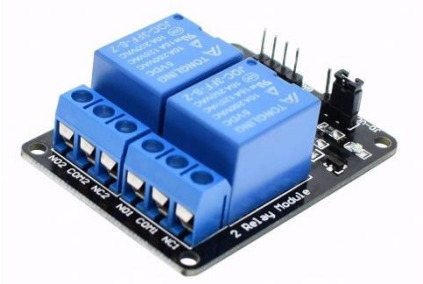


Figure 21 : Le relais

Afin d'associer le ventilateur avec les deux tubes mentionnes, j'ai fait une conception à CATIA V5 pour l'impression 3D. Voici la conception qui a été faite :

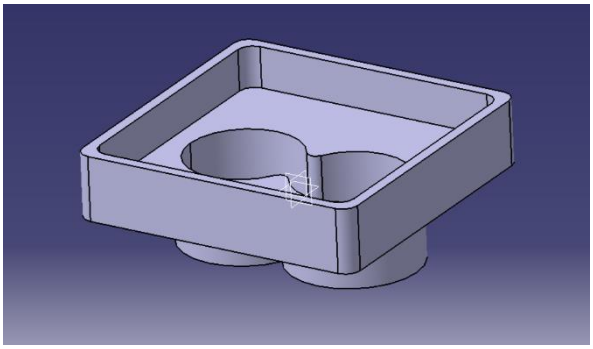


Figure 22 et Figure 23 : La conception et le produit final de l'impression 3D pour le ventilateur et les tubes

Câblage du ventilateur, prise coupée, l'alimentation 12V et le relais

a) Alimentation 12V → Ventilateur
- V- → Fil noir

b) Prise coupée → Alimentation 12V
- La terre (marron) → La terre
- La phase (vert et jaune)→ La phase
- Le neutre (bleu) → Le neutre

c) Relais → Alimentation 12V
- COM → V+

d) Relais → Ventilateur
- NO → Fil rouge

e) Relais → Arduino
- GND → GND
- IN+ → D8 (n'importe quel digital pin)
- VCC → 5V

Programmation du ventilateur connecté avec le relais

```
const int relayPin = 8; // Definir le digital pin
void setup() {
  Serial.begin(9600);
  pinMode(relayPin, OUTPUT); // Definir le digital pin avec son etat initial
  digitalWrite(relayPin, LOW); // Le ventilateur est éteint

}

void loop() {
  if (Serial.available() > 0)
    {
    char command = Serial.read();
      if (command == '1') // Si le bouton dans le dashboard est cliqué sur "Allumer le ventilateur"
      {
        digitalWrite(relayPin, HIGH); // Le ventilateur s'allume
      }
      else if (command == '0') // Si le bouton dans le dashboard est cliqué sur "Eteindre le ventilateur"
        {
        digitalWrite(relayPin, LOW); // Le ventilateur s'éteindre
        }
}
```

Figure 24 : Programmation du ventilateur avec le relais

Figure 25 : Un exemple de la boite de lumière

4.2.6. Boîte de lumière

La boîte de lumière a pour objectif de faciliter les séances photos des plantes. Elle fonctionne avec une condition du temps définie sur Node-RED. Comme le ventilateur, les deux câbles de la boîte sont connectés sur une multiprise et la multiprise est connectée sur un relais. Puis finalement le relais est connecté avec une prise coupée.

Afin d'accélérer le travail, j'ai utilisé le microcontrôleur Raspberry Pi 5 pour donner la condition du temps. Avec l'Arduino, j'aurais besoin un outil qui s'appelle « RTC (Real time clock) Module ». Cela va coûter de l'argent et le temps. Ainsi, le codage a été fait avec Node-RED et la boîte fonctionne avec un bouton si c'est sans la condition du temps.

Câblage de la boîte de lumière, relais, prise coupée et multiprise

a) Prise coupée → Relais
   - La terre → NO

b) Multiprise → Relais
   - La terre → COM

c) Relais → Raspberry Pi 5
   - VCC → 5V
   - IN → GPIO 17
   - GND → GND

4.3. Node-RED

Node-RED est un outil de programmation visuelle développé par IBM pour assembler des flux de travail et connecter des appareils de manière simple. Il est particulièrement utilisé dans le domaine de l'Internet des objets (IoT), mais il peut aussi être utilisé pour automatiser des tâches et intégrer divers systèmes comme le projet que je réalise.

Dans mon cas, Node-RED est utilisé pour afficher les données des capteurs sous forme de graphique et de jauge. Il est aussi possible de mettre les dessins au format « .svg » dans l'interface utilisateur afin de créer un jumeau digital (Digital Twin). Cela a pour but de donner une interface plus interactive et représentative. En outre, le Node-RED fournit plusieurs palettes d'outil, notamment la palette « Dashboard » qui permet d'ajouter un bouton (par exemple pour activer le nœud de fonction).

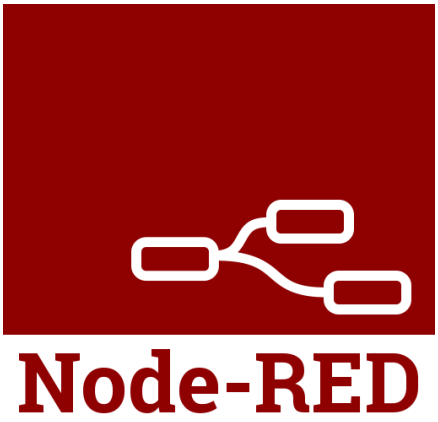


Figure 26 : Icône de Node-RED

Ce n'est pas que ça, il sert aussi à générer un fichier des données automatiquement selon la condition donnée. Donc, nous pouvons garder les données précieuses et les télécharger sans besoin d'appuyer sur un bouton ou d'interagir avec l'interface. Ensuite, il est aussi utile pour exécuter un programme de langage Python ce qui est important pour la suite de travail.

4.3.1. FLUX ARDUINO R4 MINIMA

4.3.1.1. Afficher les données sous forme graphique et jauge

Voici le flux pour l'acquisition des données :

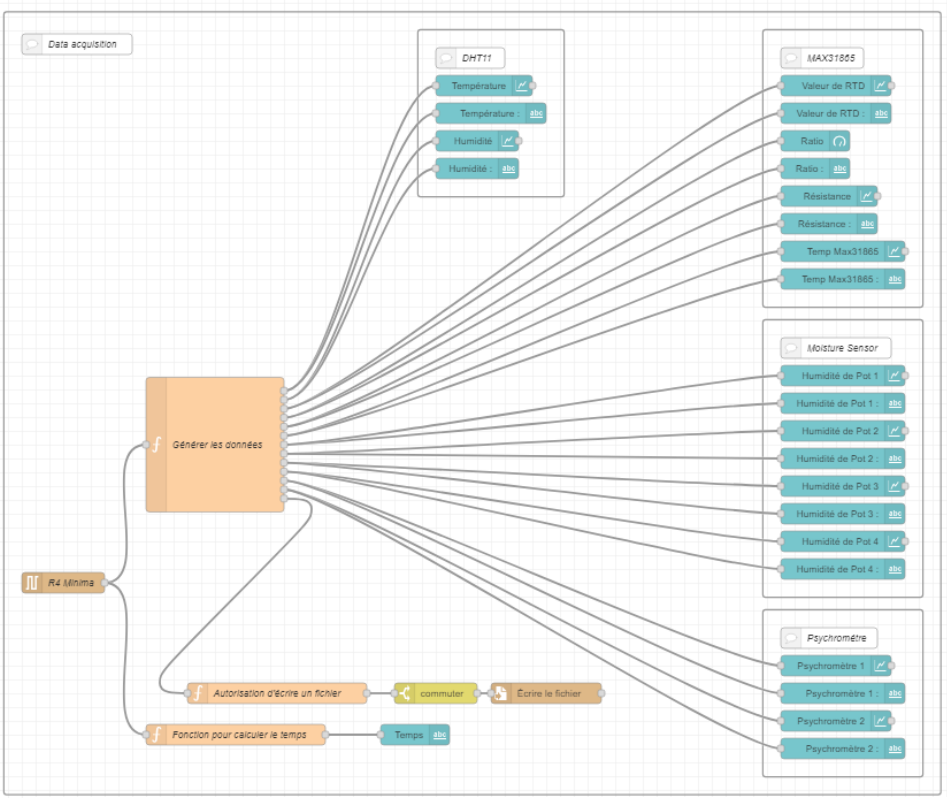


Figure 27 : Flux de l'acquisition des données

Le nœud « R4 Minima » permet de connecter le microcontrôleur l'Arduino R4 Minima avec Node-RED. Puis, le nœud fonction « Générer les données » est mis pour transférer les données aux nœuds de « Dashboard ».

```
msg.fileName=global.get('FileName')
const m = msg.payload.split(',');
const t = {payload:parseFloat(m[0])};
const h = {payload:parseFloat(m[1])};
const rtd = {payload:parseFloat(m[2])};
const ratio = {payload:parseFloat(m[3])};
const res = {payload:parseFloat(m[4])};
const tm = {payload:parseFloat(m[5])};
const hpot1 = {payload:parseFloat(m[6])};
const hpot2 = { payload: parseFloat(m[7]) };
const hpot3 = { payload: parseFloat(m[8]) };
const hpot4 = { payload: parseFloat(m[9]) };
const tempk = {payload: parseFloat(m[10])};
const tempk2 = {payload:parseFloat(m[11])};
msg.temperature = parseFloat(m[0])
msg.humidity = parseFloat(m[1])
msg.rtd = parseFloat(m[2])
msg.ratio = parseFloat(m[3])
msg.resistance = parseFloat(m[4])
msg.temperaturemx31865 = parseFloat(m[5])
msg.humidity_pot1 = parseFloat(m[6])
msg.humidity_pot2 = parseFloat(m[7])
msg.humidity_pot3 = parseFloat(m[8])
msg.humidity_pot4 = parseFloat(m[9])
msg.temperature_k = parseFloat(m[10])
msg.temperature_k2 = parseFloat(m[11])
var currentTimeUTC = new Date();
var timezoneOffset = 60;
if (currentTimeUTC.getMonth() >= 2 && currentTimeUTC.getMonth() < 10) {
    timezoneOffset = 120;
}
var currentTimeFrance = new Date(currentTimeUTC.getTime() + timezoneOffset * 60000);
var timestamp = currentTimeFrance.toISOString();
msg.payload = {
    timestamp: timestamp
}
```

```
37
38  flow.set('Temperature_Ext', m[0]);
39  flow.set('Humidity_Ext', m[1]);
40  flow.set('Ratio', m[3]);
41  flow.set('Temperature_pot', m[5]);
42  flow.set('Humidity_pot1', m[6]);
43  flow.set('Humidity_pot2', m[7]);
44  flow.set('Humidity_pot3', m[8]);
45  flow.set('Humidity_pot4', m[9]);
46  flow.set('Temperature_k', m[10]);
47  flow.set('Temperature_k2', m[11]);
48  flow.set('m',m);
49
50
51  msg.payload = 'Horodatage , ' + timestamp + ', Temperature , ' + m[0] + ", Humidite ," + m[1] + ", Valeur de RTD ," + m[2] + ", Ratio ," + m[3] + ", Resistance ," + m[4] + ", Temp max3
52  return [t,h,rtd,ratio,res,tm,hpot1,hpot2,hpot3,hpot4,tempk,tempk2,msg];
53
```

Figure 28 et Figure 29 : Programmation de "Générer les données"

Dans cette programmation, « m » est utilisé en tant qu'une variable qui garde les données sous une forme de tableau. Le « global.get » permet de définir la variable que nous voulons garder tant que « global.set » permet de reprendre la variable définie dans la fonction sans besoin de la redéfinir.

Après ça, le nœud « Autorisation d'écrire un fichier » est employé pour donner l'autorisation pour créer un fichier de « .csv » quand le bouton sur le dashboard est cliqué.

```
1
2  msg.writeEnabled=false
3  if (flow.get('fileGenerationEnabled')==!undefined){
4  if (flow.get("fileGenerationEnabled")==true) {
5  msg.writeEnabled=true;}}
6
7  return msg;
```

Figure 30 : Flux de "Autorisation pour écrire un fichier"

Finalement, le nœud « Fonction pour calculer le temps » sert à afficher l'horodatage dans le fichier afin de faciliter les lectures de données.

```
1   var currentTimeUTC = new Date();
2   var timezoneOffset = 60;
3   if (currentTimeUTC.getMonth() >= 2 && currentTimeUTC.getMonth() < 10) {
4       timezoneOffset = 120;
5   }
6   var currentTimeFrance = new Date(currentTimeUTC.getTime() + timezoneOffset * 60000);
7   var timestamp = currentTimeFrance.toISOString();
8   msg.payload = {
9       timestamp: timestamp
10  }
11
12  msg.payload = timestamp
13
14  return msg;
```

Figure 31 : Flux de «Fonction pour calculer le temps"

4.3.1.2. Créer un fichier

Voici le flux pour créer le fichier des données :

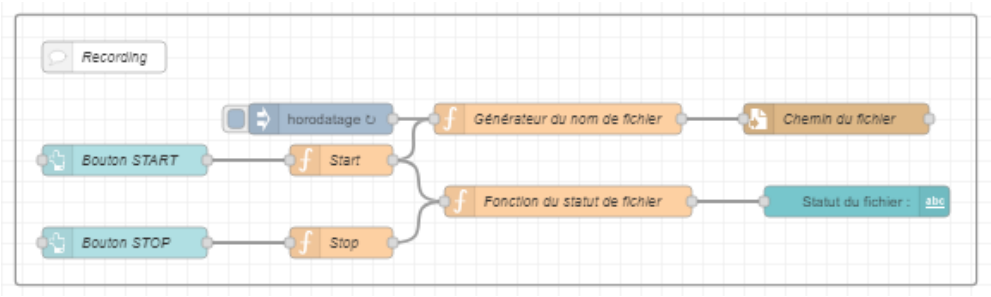


Figure 32 : Flux de "Creer un fichier"

Les nœuds de boutons « START » et « STOP » sont employé pour donner une chaine de caractères (start et stop) pour activer les nœuds fonctions « START » et « STOP ».

```
flow.set("fileGenerationEnabled", true);
return msg;
```

Figure 33 : Fonction "START"

```
flow.set("fileGenerationEnabled", false);
return msg;
```

Figure 34 : Fonction "STOP"

Voici la fonction si le bouton « START » est cliqué :

```
if (flow.get('fileGenerationEnabled')==!undefined){
if (flow.get("fileGenerationEnabled")==true) {
    let output = [];

        for (var i=1;i<2;i++) {


            let now = new Date();
            now.setTime(now.getTime() - 1000*60*60*24*i);
            let yyyy = now.getFullYear();
            let mm = now.getMonth() < 9 ? "0" + (now.getMonth() + 1) : (now.getMonth() + 1);
            let dd  = now.getDate() < 10 ? "0" + (now.getDate() + 1) : (now.getDate() + 1);
            let hh = now.getHours() < 10 ? "0" + now.getHours() : now.getHours();
            let mmm  = now.getMinutes() < 10 ? "0" + now.getMinutes() : now.getMinutes();
            let ss  = now.getSeconds() < 10 ? "0" + now.getSeconds() : now.getSeconds();

            let newfile = { "topic": "archive", "payload": ""};
            newfile.fname = "DONNEES"+"_"+ dd + "." + mm + "." + yyyy +'_'+hh+"."+mmm+"."+ss+ ".csv";
            newfile.path = "C:\\Users\\Danish Aiman\\OneDrive\\Documents\\GMP\\stage s4\\DATA\\";
            newfile.fullName = newfile.path+newfile.fname;
            global.set('FileName', newfile.fullName) //global set
            newfile.payload = 'cd '+newfile.path + ' && ' + 'echo test DHT 11 and MAX 31865 > ' + newfile.fname;
            output.push(newfile);

        }
}
}
return output;
```

Figure 35 : Fonction "Générateur du nom de fichier"

Dans la fonction, il faut définir le chemin de fichier c'est-à-dire où nous voulons sauvegarder le fichier des données. Elle permet également de définir le nom du fichier avec l'horodatage.

4.3.1.3. Jumeau Digital

Dans ce jumeau digital, je mets les dessins des plantes au format « .svg » pour donner l'indication de l'arrosage. Pour chaque fichier de dessins, ouvrir le fichier sous forme de texte et le mets dans le nœud qui s'appelle « SVG Graphics ».

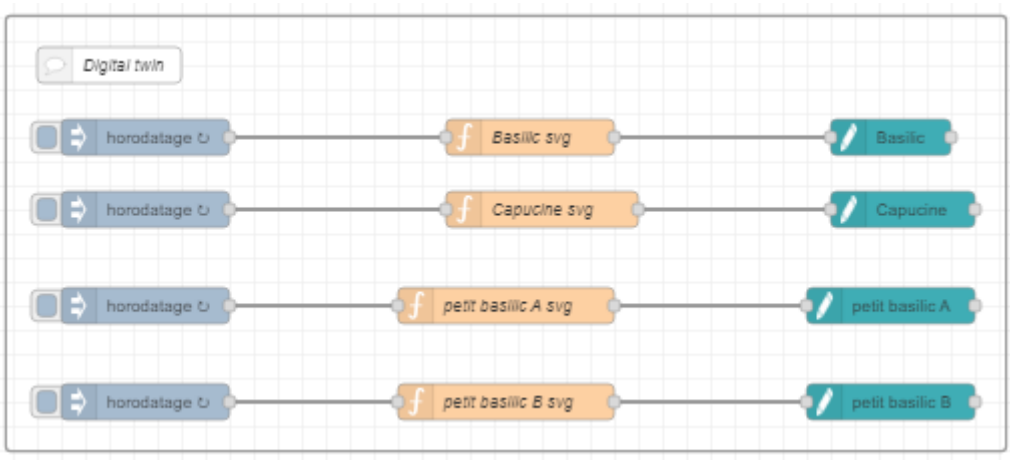


Figure 36 : Flux de " Jumeau Digital"

Dans chaque nœud, j'ai mis une fonction qui permet d'afficher les conditions internes et externes ainsi que l'indicateur pour l'arrosage des plantes.

```
msg.payload=[];

msg.payload.push({
        "command": "update_text",
        "selector": "#tspan4",
        "textContent": "Humidité:"+flow.get('Humidity_Ext')
})
msg.payload.push({
        "command": "update_text",
        "selector": "#tspan3",
        "textContent": "Température:"+flow.get('Temperature_Ext')
})
msg.payload.push({
        "command": "update_text",
        "selector": "#tspan5",
        "textContent": "Température du sol:"+flow.get('Temperature_pot')
})
msg.payload.push({
        "command": "update_text",
        "selector": "#tspan6",
        "textContent": "Humidité du sol:"+ flow.get('Humidity_pot1')
})

const hpot1 = flow.get('Humidity_pot1')

if (hpot1<75) {
msg.payload.push ({
"command": "update_style",
"selector": "#path47",
"attributeName": "fill",
"attributeValue": "red"})
}
else
        {
                msg.payload.push({
                        "command": "update_style",
                        "selector": "#path47",
                        "attributeName": "fill",
                        "attributeValue": "green"
                })
        }
return msg;
```

Figure 37 : Un exemple de la fonction de dessin basilique

A noter que le « #path » et le « #tspan » sont différents pour chaque dessin. Ainsi, il faut bien vérifier dans le code source du dessin.

4.3.1.4. Ventilateur

Voici le flux pour allumer le ventilateur :

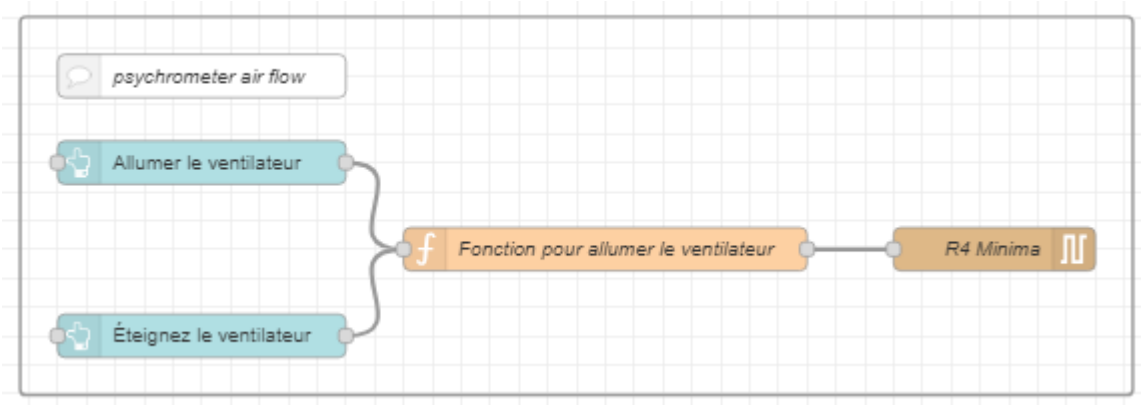


Figure 38 : Flux de "Ventilateur"

Dans le nœud « Fonction pour allumer le ventilateur », il envoie le « msg.payload » sous forme booléen (1 ou 0) pour allumer le ventilateur.

```
1    return msg;
```

Figure 39 : La fonction pour allumer le ventilateur

« return msg » signifie que le nœud de fonction va transférer le « msg » au nœud suivant.

4.3.2. FLUX RASPBERRY PI 5

Ensuite, nous allons passer au deuxième flux qui servit à exécuter le programme de python et allumer la boîte de la lumière. Après avoir déployé le flux, les appareils photo vont prendre les photos des plantes et la lumière reste s'allumer. Néanmoins s'il fait nuit, la lumière s'allume et s'éteindre avec la condition du temps définie. Voici le flux :

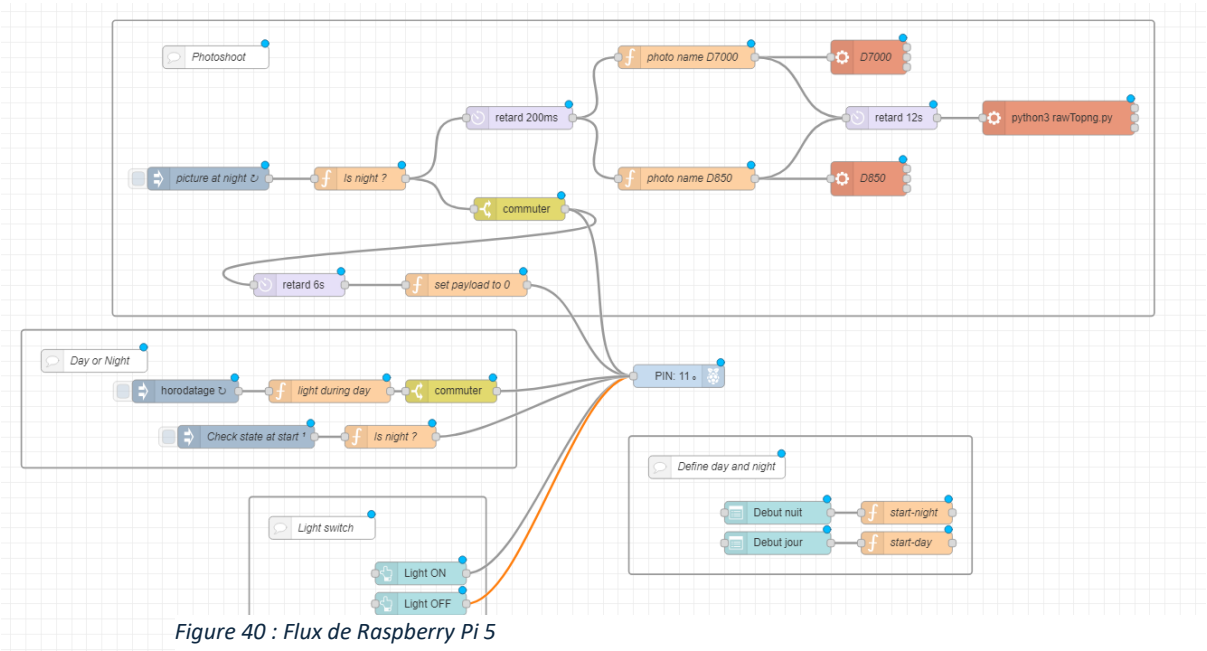


Figure 40 : Flux de Raspberry Pi 5

4.3.2.1. Définir le début du jour et de la nuit

Dans ce flux, l'utilisateur permet de définir le début de la nuit qui va déclencher la condition du temps de la totalité du flux. Pour que le flux puisse fonctionner toujours et que les valeurs restent les mêmes, j'ai utilisé la fonction de « ContextStorage » (voir l'annexe).

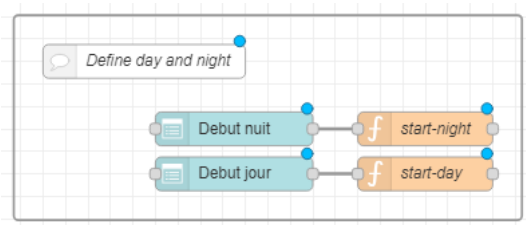


Figure 41 : Flux pour définir le début du jour et de la nuit

```
1   flow.set('start-night', msg.payload)
2   return msg;
```

```
1   flow.set('start-day', msg.payload)
2   return msg;
```

4.3.2.2. Interrupteur de la boîte de lumière

La boîte de lumière peut fonctionner avec des boutons. Le nœud est connecté avec le nœud du microcontrôleur Raspberry Pi 5 qui s'appelle « PIN 11 » étant aussi le pin de GPIO 17.

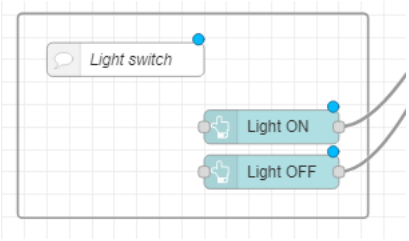


Figure 42 : Le flux de l'interrupteur de la boite de lumière

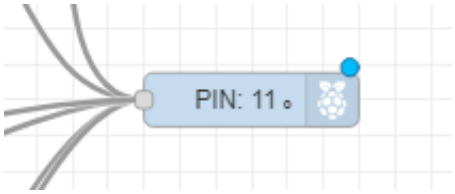


Figure 43 : Le nœud du microcontrôleur

4.3.2.3. Nuit ou Jour ?

Ce flux permet de transférer la vérification de « payload » au flux de « Photoshoot ». Il aide aussi à savoir s'il fait nuit ou pas.

```
var currentdate = new Date();
var start_night=flow.get('start-night')
var start_day=flow.get('start-day')
msg.actuate=0
if (currentdate.getHours()==start_night){
    if ((currentdate.getMinutes() ==0) && (currentdate.getSeconds() < 2)) {
    msg.activate=1
    msg.payload=0}
}
if (currentdate.getHours() == start_day) {
    if ((currentdate.getMinutes() == 0) && (currentdate.getSeconds() < 2)) {
        msg.activate = 1
        msg.payload = 1
    }
}

return msg;
```

Figure 9 : La fonction de "light during day"

```
var currentdate = new Date();
var start_night=flow.get('start-night')
var start_day=flow.get('start-day')

if ((currentdate.getHours()>start_night) || (currentdate.getHours()<start_day)){
    msg.night=true
    msg.payload=0}
    else {
        msg.night=false
        msg.payload=1}
return msg;
```

Figure 44 : La fonction de "is night ?"

4.3.2.4. La prise des photos

Ce flux est le flux le plus important car il récupère la valeur de « payload » et exécuter les programmes de python. Il commence par une fonction de « Is night ? » afin de savoir s'il fait déjà nuit ou pas. S'il fait nuit, la boîte de lumière va s'allumer pendant 6 secondes avant de s'éteindre. Puis, il se rallume pour la séance des photos pendant la nuit.

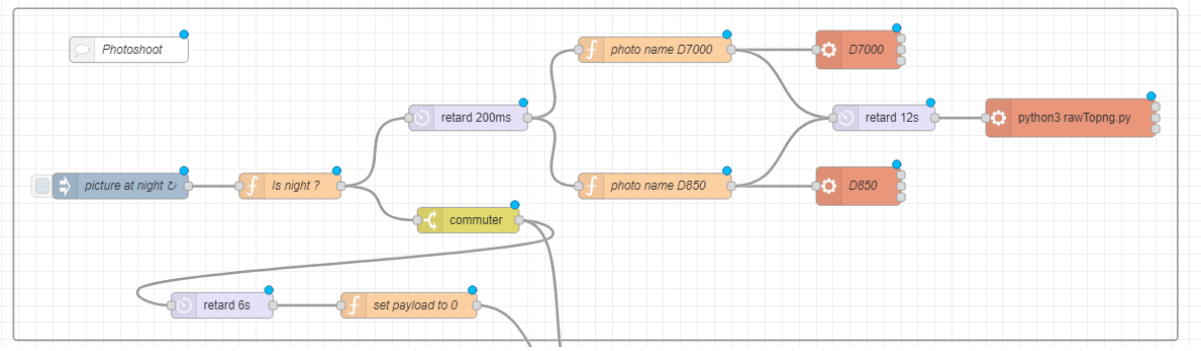


Figure 10 : Le flux pour la séance des photos

Ensuite, je vais expliquer la sous partie du flux étant vraiment important afin de prendre les photos. Les appareils photos attendent 0,2 secondes avant de commencer la prochaine prise de photo. Les appareils photo utilisés sont Nikon D7000 et Nikon D850.

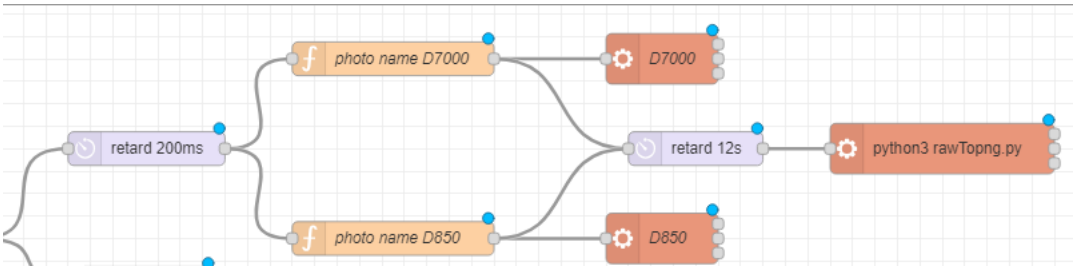


Figure 46 : La sous partie du flux

Dans la fonction de « photo name D7000 et D850 », se trouve une programmation pour définir le nom du fichier des photos afin d'identifier les photos.

```
1  msg.payload= "D7000_"+Date.now()+'.nef';
2  return msg;
```

Figure 47 : Fonction de "photo name D7000"

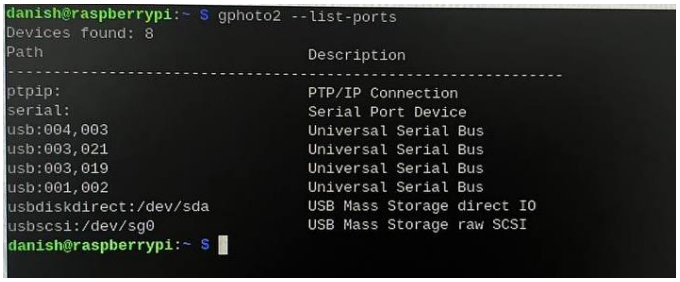
```
1  msg.payload= "D850_"+Date.now()+'.nef';
2  return msg;
```

Figure 48 : La fonction de "photo name D850"

« Date.now() » signifie les millisecondes et à noter que les photos vont être sauvegardées au format « .nef ».

4.4. GPHOTO2

Pour les nœuds de « D7000 » et « D850 », ils sont définis par une commande étant faite par le terminal. Il faut tout d'abord télécharger un outil qui s'appelle « gphoto2. Une fois qu'il est téléchargé, saisissez la commande « gphoto2 --list-ports » dans le terminal. Cela permet d'afficher les appareils photos qui ont connecte avec le microcontrôleur. Nous allons voir donc ce résultat par exemple :



Ensuite, afin d'identifier le nom de port porte par les appareils photos, il faut saisir la commande « gphoto2 --auto-detect ». Il va afficher les noms des ports comme ci-dessous :

Puis, il existe une commande pour prendre une photo sur le terminal. J'ai utilisé la commande « gphoto2 --capture-image-and-download ». Grace a cette commande l'appareil photo va prendre une photo et la télécharger mais sans un nom propre pour le fichier. Alors, puisque la fonction de nom du fichier a été défini sur Node-RED, nous pouvons juste ajouter « --filename » à la fin de la commande. Cependant, nous avons deux appareils photos. Donc, il est important de définir quel appareil photo que nous voulons utilise. Avec la commande de « --auto-detect », nous pouvons juste prendre le nom du port et le mettre au début de la commande.

Comme par exemple, « --port usb : 003,019 » pour le Nikon D7000 et « --port usb : 003,021 » pour le Nikon D850.

Pour finir, nous pouvons mettre cette commande par exemple :

« cd ~/EXP_Photos/ && gphoto2 --port usb:001,004 --capture-image-and-download –filename »

« EXP_Photos » signifie le chemin ou nous voulons sauvegarder les photos. La commande est mise dans le nœud « exec » pour la lancer.

4.5. RawToPng.py

Les photos prises seront au format « .nef (raw) » pour pouvoir changer les paramètres des photos. Cependant, les photos seront grandes en mémoire. Ainsi, il faut convertir le format au « .jpeg » ou « .png ». Afin de le réaliser, j'ai fait une programmation de python pour qu'il puisse être exécuter avec le nœud de « exec ». Voici la programmation :

RawToPng.py

```
1  import rawpy
2  from imageio import imsave
3  import sys
4  import os
5
6  base_folder = "/home/danish/EXP_Photos/"
7
8  def convert(filename, extension=".nef"):
9       basename = base_folder + filename
10      print(basename)
11      with rawpy.imread(basename + extension) as raw:
12           rgb = raw.postprocess()
13      imsave(basename[:-4] + '.jpeg', rgb)
14      os.remove(basename) # delete RAW
15
16 convert(sys.argv[1], "")
```

Figure 49 : Le script python de la conversion raw au jpeg

Les photos en «.nef » sont primordiales pour changer la luminosité des photos et avoir plus de pixel en ajoutant cette ligne :

« rgb = raw.postprocess(gamma= (2.222, 20), no_auto_bright=True, output_bps=16) » (à la ligne 12)

Cette ligne de commande va diminuer les pixels qui se produisent dans les photos.

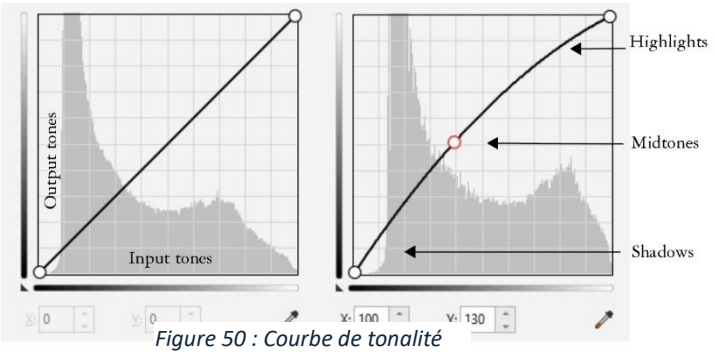


Figure 50 : Courbe de tonalité

La figure ci-dessus montre que nous pouvons augmenter la luminosité en modifiant la ligne sur le graphique. Si la coordonnée de l'abscisse sur la partie noire est bougée à la partie blanche de l'ordonnée, nous aurons une photo plus claire avec plus de pixel.

Cependant, il n'est pas activé car les photos prises sont déjà propres.

5. INSTALLATION DE L'EXPERIENCE

Avant d'installer l'expérience, il faut bien vérifier les programmations faites ainsi que les capteurs utilisés. Afin de soigner les câblages de capteur, j'ai fait quelques impressions 3D pour certain capteur et relais.

Une fois que c'est fait, j'ai transféré l'expérience de laboratoire ISM à la salle 219 à l'IUT Aix-Marseille. Il est nécessaire que la salle soit vide afin de diminuer la vibration qui peut bouger l'appareil photo et changer la mise aux pointes.

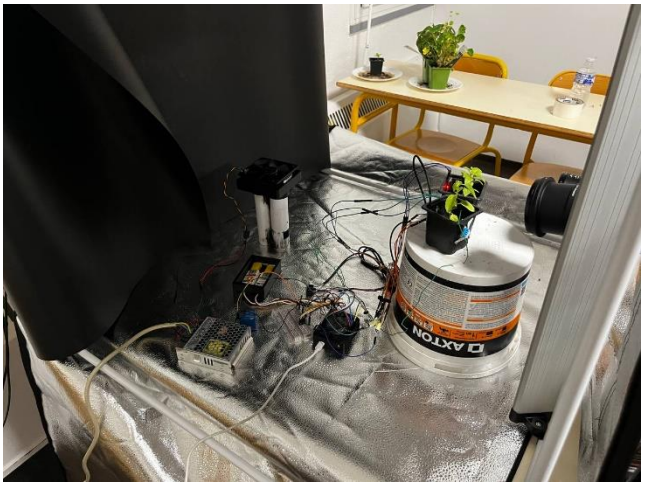



Figure 51 et Figure 52 : Installation de l'expérience

Les appareils photo se focalise sur les pétioles et certaines feuilles doivent être coupées pour ne pas bloquer les pétioles. Pour pouvoir faciliter les mesures des pétioles, un papier avec des petit carrés mesurant 5x5 mm sera mis derrière les plantes pour savoir l'échelle

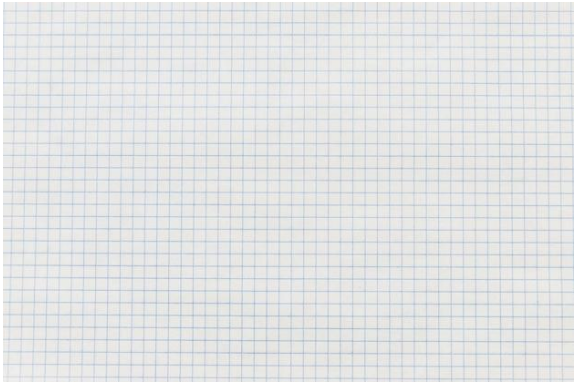


Figure 53 : Un exemple de papier avec des petits carrés

6. PROBLEMES RECONTRES

Lors de mes huit semaines de stage, je me rencontrais beaucoup de problème qui a retardé le travail. Voici les trois problèmes que j'ai eu :

1. Durée de stage

La durée de stage est courte pour finir la totalité du sujet. Avec seulement huit semaines de stage, je ne pourrais pas commencer les calculs et les mesures sur les plantes. J'ai commencé l'expérience deux semaines avant la date finale de stage. Ainsi, j'ai passé beaucoup de temps sur l'installation de l'appareil photo et les vérifications des photos. En plus, un des appareils photos manque une bague a longe. Donc, il faut passer une commande et finalement la séance de photo a commencé la semaine suivante.

2. Manque d'expérience

En tant qu'un étudiant en génie mécanique, j'avais moins de l'expérience en mécatronique. Cependant, je connais un peu les bases de la programmation grâce aux expériences au lycée. Parfois, j'ai passé beaucoup de temps sur une programmation ou un flux de Node-RED. Grâce aux stagiaires dans le laboratoire et M. Loïc TADRIST, les problèmes sont résolus. Au début du stage, je n'étais pas à l'aise avec les nouveaux logiciels et je ne pouvais pas travailler en autonomie pour des tâches qui étaient difficiles pour moi.

3. Performance de logiciel

Lors de ce stage, j'ai travaillé sur la programmation d'Arduino et le Node-RED en utilisant mon ordinateur. Sachant qu'il est moins puissant, j'ai eu toujours de problème sur mon ordinateur qui bug à cause du manque de mémoire RAM moindre.

7. MISSION HORS DU SUJET DE STAGE

Lors de l'absence de 5 jours de mon tuteur, M. Loïc TADRIST, j'ai fait une conception d'un banc essai à CATIA V5 demandé par un de ses stagiaire, Lucas DUBOIS. Son objectif est pour qu'il puisse présenter son banc d'essai au tout le monde de manière claire et représentative.

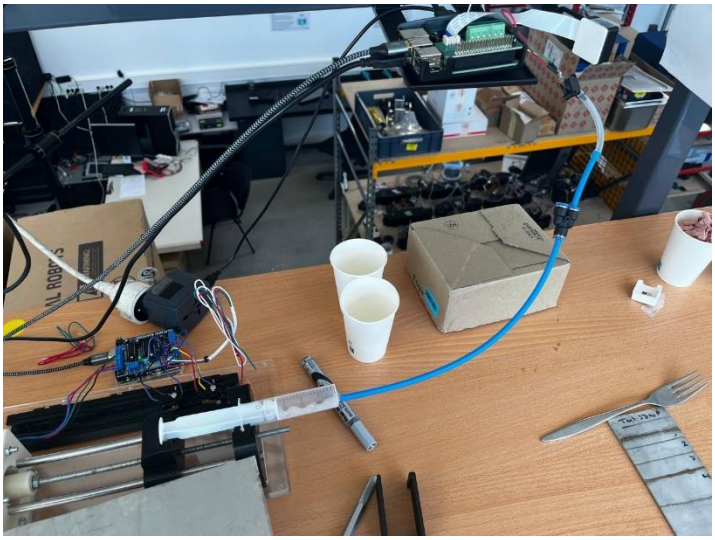


Figure 54 : Le banc d'essai de Lucas DUBOIS

Pour les modélisations de certaines pièces compliquées comme le microcontrôleur et les capteurs, j'ai récupéré les pièces grâce à la communauté qui s'appelle « GrabCad ». En bref, c'est un site permet de partager les fichiers « cad » ou « step » qui est compatible avec CATIA V5. (Voir Figure … de l'annexe pour la conception du banc d'essai)

GRABCAD

Figure 55 : Icone de GrabCad

Lors de la réalisation de conception j'ai appris un nouvel outil à CATIA V5 permettant de modéliser les fils et les tuyaux. L'outil s'appelle « Electrical Harness Assembly »

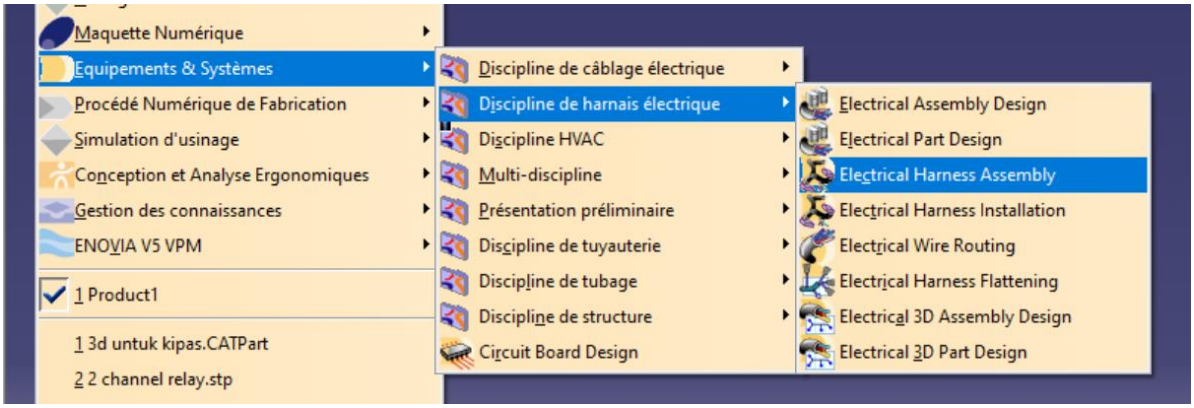


Figure 56: Le chemin pour trouver l'outil

8. CONCLUSION

Ce stage dédié au développement d'un banc d'essai pour l'étude des caractéristiques mécaniques des plantes a permis d'atteindre mes objectifs professionnels et personnels et d'acquérir des connaissances précieuses.

Le projet a commencé par la partie programmation des capteurs incluant les capteurs que je n'ai jamais programmée durant mes études en génie mécanique. Il suit ensuite avec la création de l'interface d'utilisateur sur un nouveau logiciel que je trouve très utile pour de nombreux projets qui nécessitent une surveillance et une collecte de données. Finalement, l'installation de l'expérience me permet d'apprendre le langage programmation de python. Pour la suite de stage, il reste à faire les études et les mesures des plantes pour mieux comprendre les propriétés de matériaux végétaux.

D'un point de vue personnel, ce stage a été une expérience enrichissante qui m'a permis de développer des compétences techniques en mécatronique, ainsi que de consolider mes connaissances en mécatronique. Travailler en collaboration avec l'équipe de recherche a également renforcé mes capacités de travail en équipe et de gestion de projet.

D'un point de vue professionnel, ce stage m'a permis de développer des compétences qui peuvent améliorer mon employabilité et élargir mes études dans le domaine mécanique et mécatronique, tout en me fournissant une expérience pratique précieuse et en renforçant ma capacité à travailler sur des projets complexes. Ces acquis sont directement en lien avec mon projet avant de finir les études étant de devenir un étudiant polyvalent sur le domaine d'études.

Pour conclure, mes huit semaines au laboratoire ISM m'a attribué non seulement de participer de faire partie dans l'équipe mais aussi d'acquérir les compétences précieuses que je pourrais utiliser pour mes études à l'avenir. Les perspectives incluent l'amélioration continue du banc d'essai offrant ainsi des opportunités prometteuses pour la recherche et le développement dans le domaine mécanique et mécatronique.

9. SITOGRAPHIE

Présentation de l'entreprise : https://ism.univ-amu.fr/fr

Forum de l'Arduino : https://forum.arduino.cc/

Forum de Raspberry Pi : https://forums.raspberrypi.com/

gphoto2 : http://gphoto.org/

Rawpy : https://pypi.org/project/rawpy/

GrabCad: https://grabcad.com/library

10. ANNEXE

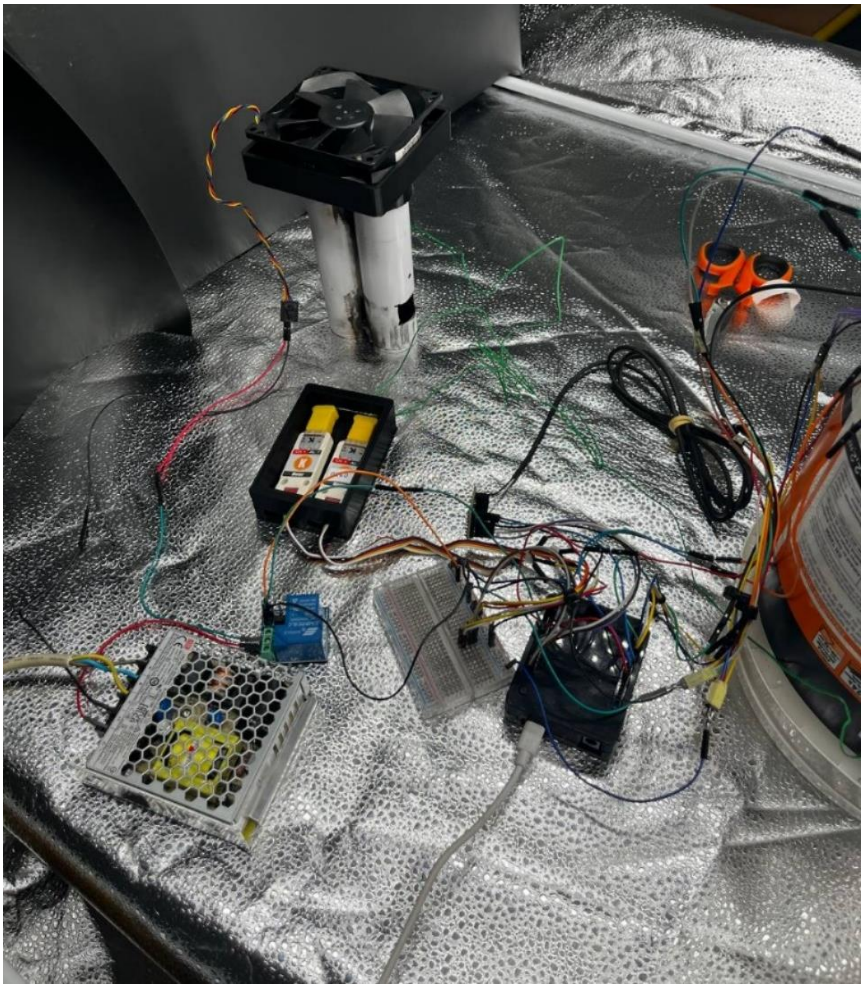


Figure 57 : Le banc d'essai complet

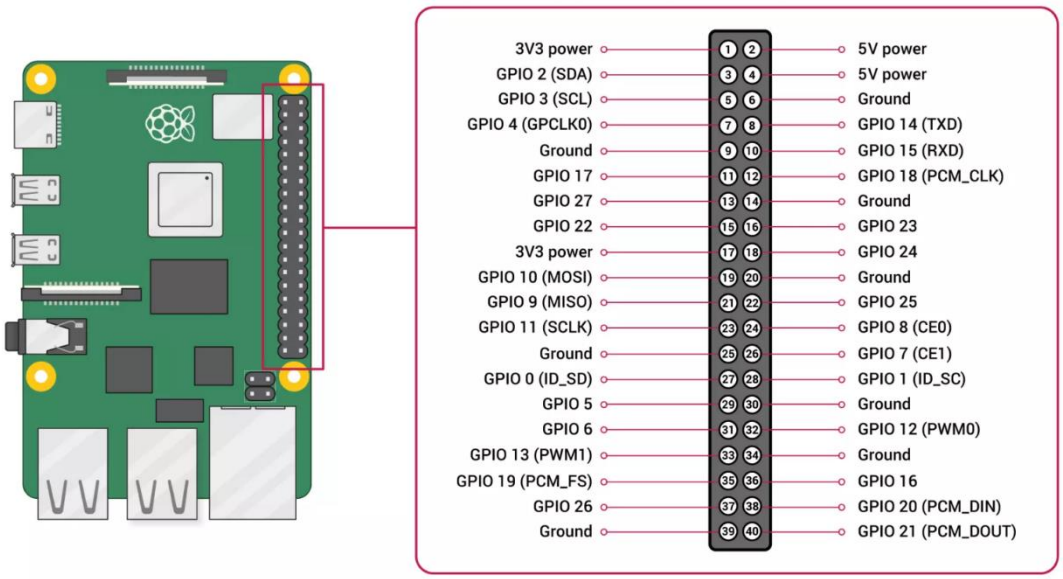
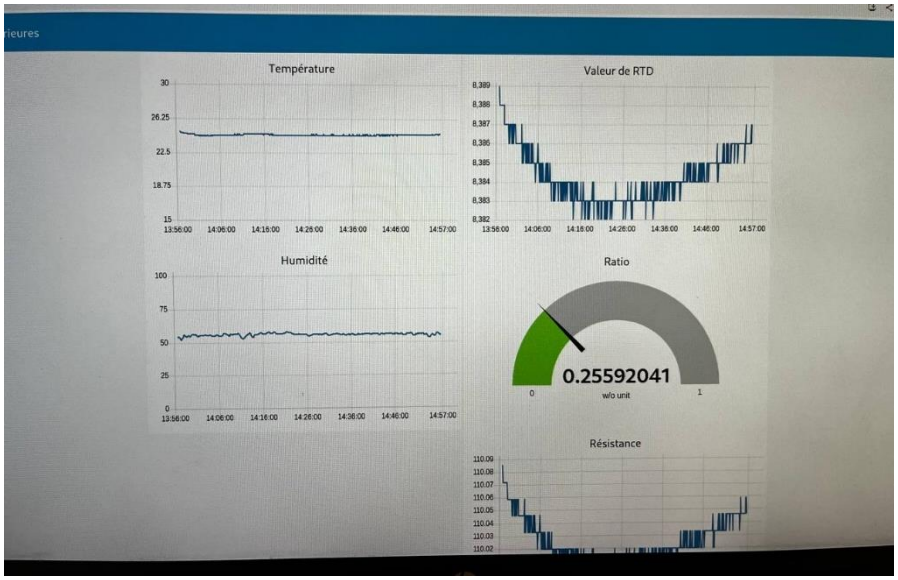
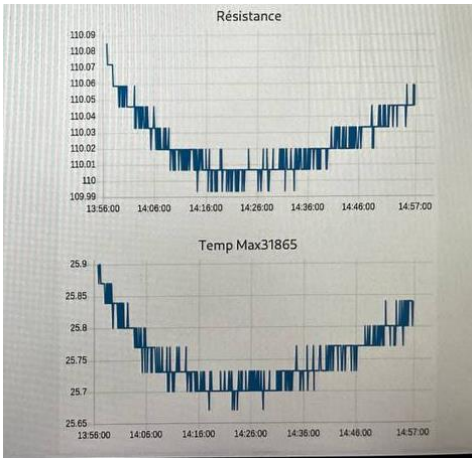


Figure 58 : Les pins du microcontrôleur Raspberry Pi 5





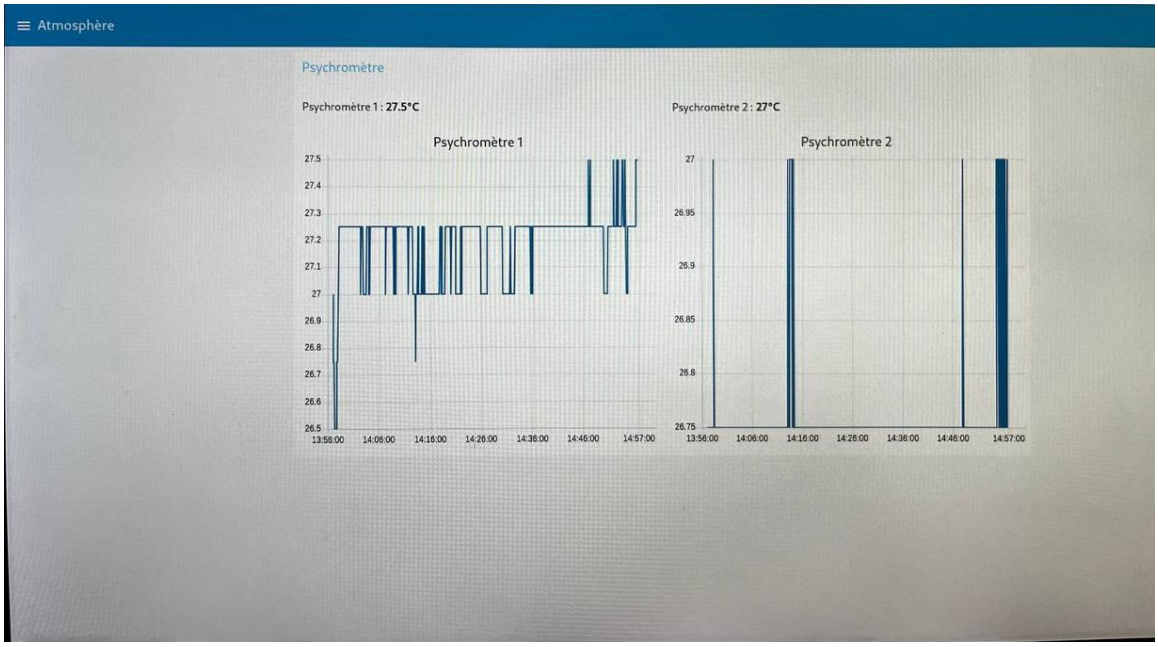
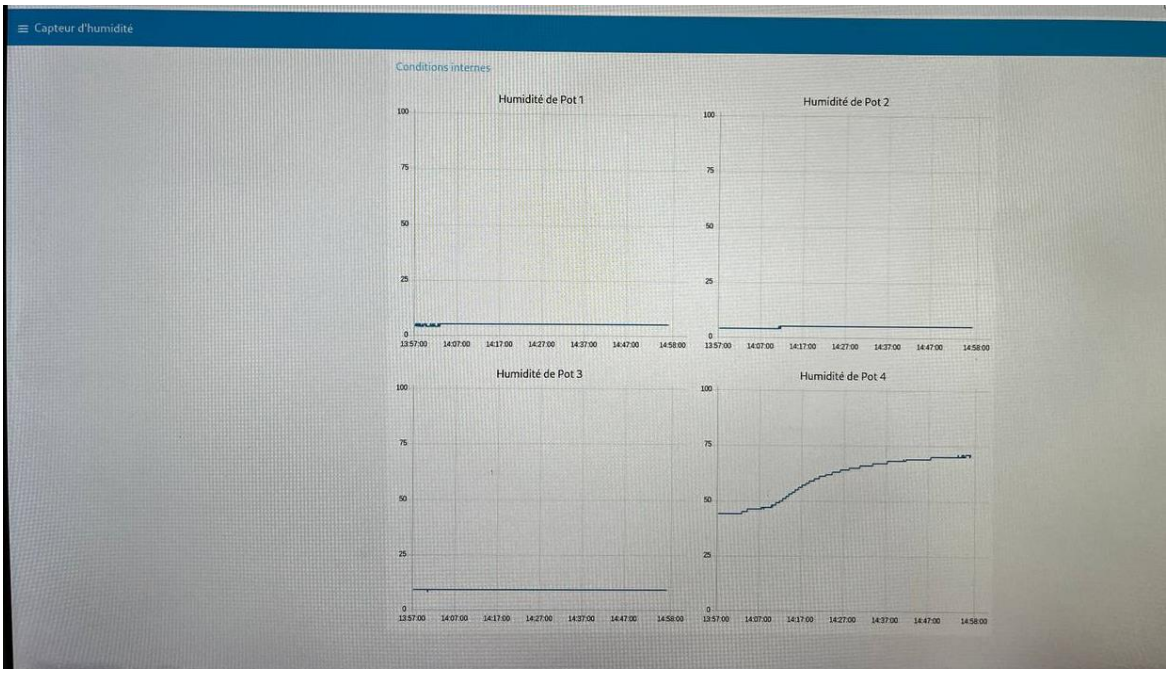


Figure 59,60,61 et 62 : Le dashboard de Node-RED



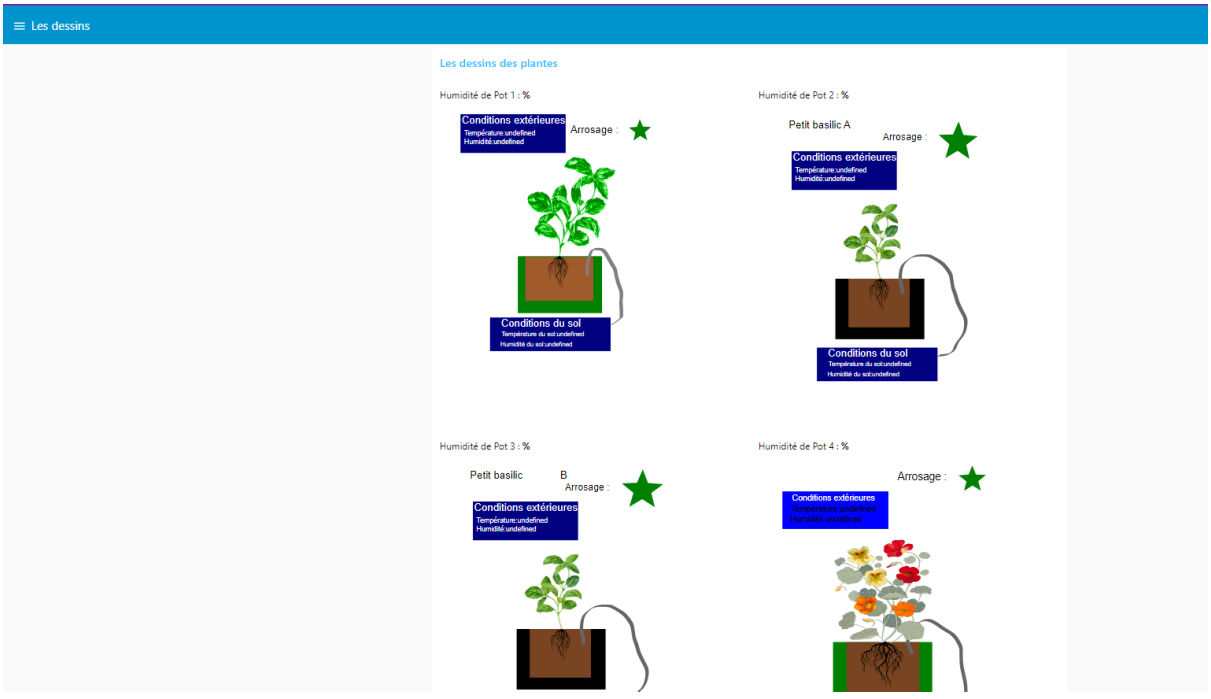


Figure 63 et 64 : Le dashboard de Node-RED

Figure 65 : Un exemple de photo prise par Nikon D850

Figure 66 :Un exemple de photo prise par Nikon D7000

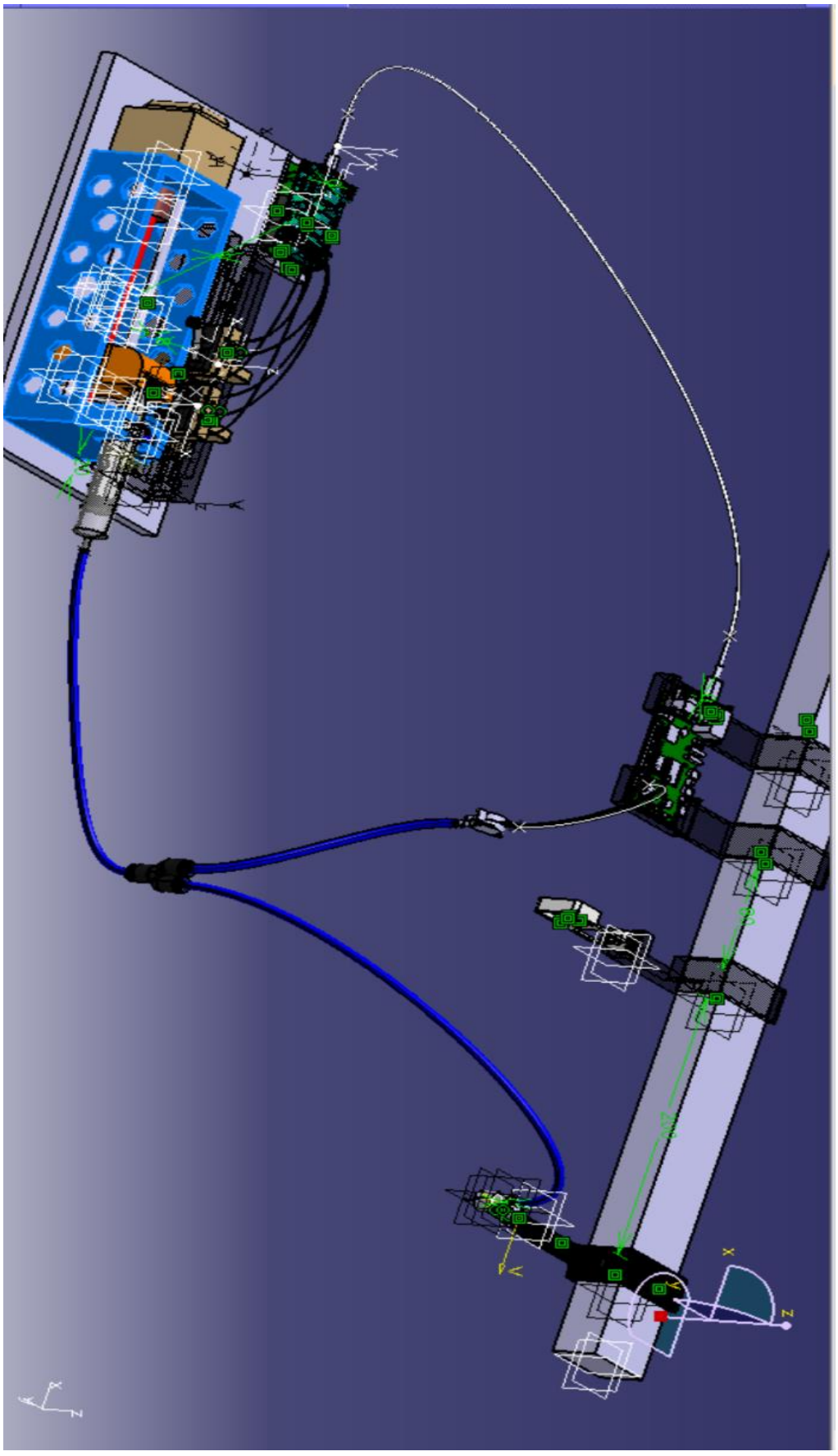


Figure 67 : La conception du banc d'essai de Lucas DUBOIS