

# Rapport de tache

Alexandre Pichot

Intervenants : C.Lebastard JL Salvat



## PLANT MANAGER



## Table des matières

<b>Introduction.....</b>	<b>3</b>
Explication .....	3
Architecture .....	4
<b>Taches et objectifs.....</b>	<b>5</b>
Gantt .....	5
Mes taches.....	5
Mes objectifs :.....	6
<b>Test de capteur et actionneurs unitaire.....</b>	<b>7</b>
Flotteur .....	7
Leds .....	8
Bouton poussoir.....	9
Capteur de CO2 .....	10
Pompe .....	11
Bandeau led .....	12
<b>Carte électronique .....</b>	<b>13</b>
Explication .....	13
Design spark.....	13
Carte Pompe .....	14
Carte Plante .....	15
Soudage et tests des 2 cartes .....	16
<b>Impression 3d .....</b>	<b>17</b>
Explication .....	17
Problème .....	18
Boitier pompe .....	18
Boitier Plante .....	20
<b>Conclusion .....</b>	<b>22</b>
<b>Annexe .....</b>	<b>23</b>
Source :.....	25

# Introduction

## Explication

Notre projet est un système d'arrosage automatique intelligent conçu pour surveiller et entretenir une plante tout en prenant en compte les conditions météorologiques en temps réel. Il est composé de deux boîtiers : un premier placé directement sur la plante pour mesurer l'humidité du sol ainsi que les paramètres environnementaux et un second dédié à la gestion de l'arrosage.

Le premier boîtier est équipé de capteurs permettant d'évaluer l'humidité de l'air, la température, la lumière ou encore l'humidité du sol. Ces données permettent d'adapter l'arrosage en fonction des conditions extérieures.

Le second boîtier, placé à proximité de la plante, contient les éléments nécessaires pour assurer l'irrigation. Une pompe à eau est activée lorsque l'humidité du sol est insuffisante et un capteur flotteur surveille le niveau d'eau du réservoir pour éviter tout dysfonctionnement. En cas de niveau d'eau trop bas, un buzzer et une LED préviennent l'utilisateur.

Le tout est piloté par un Raspberry Pi Zero 2W qui centralise les données des capteurs et contrôle l'arrosage en temps réel. Alimenté en 5V via un câble USB-C, il permet d'optimiser l'arrosage en fonction des conditions climatiques et des besoins réels de la plante. Ce système est particulièrement utile lorsque l'on part en vacances afin d'éviter que les plantes ne se dessèchent ou encore pour les personnes ayant tendance à oublier d'arroser régulièrement leurs plantes, garantissant ainsi un entretien automatisé et sans effort.

## Architecture

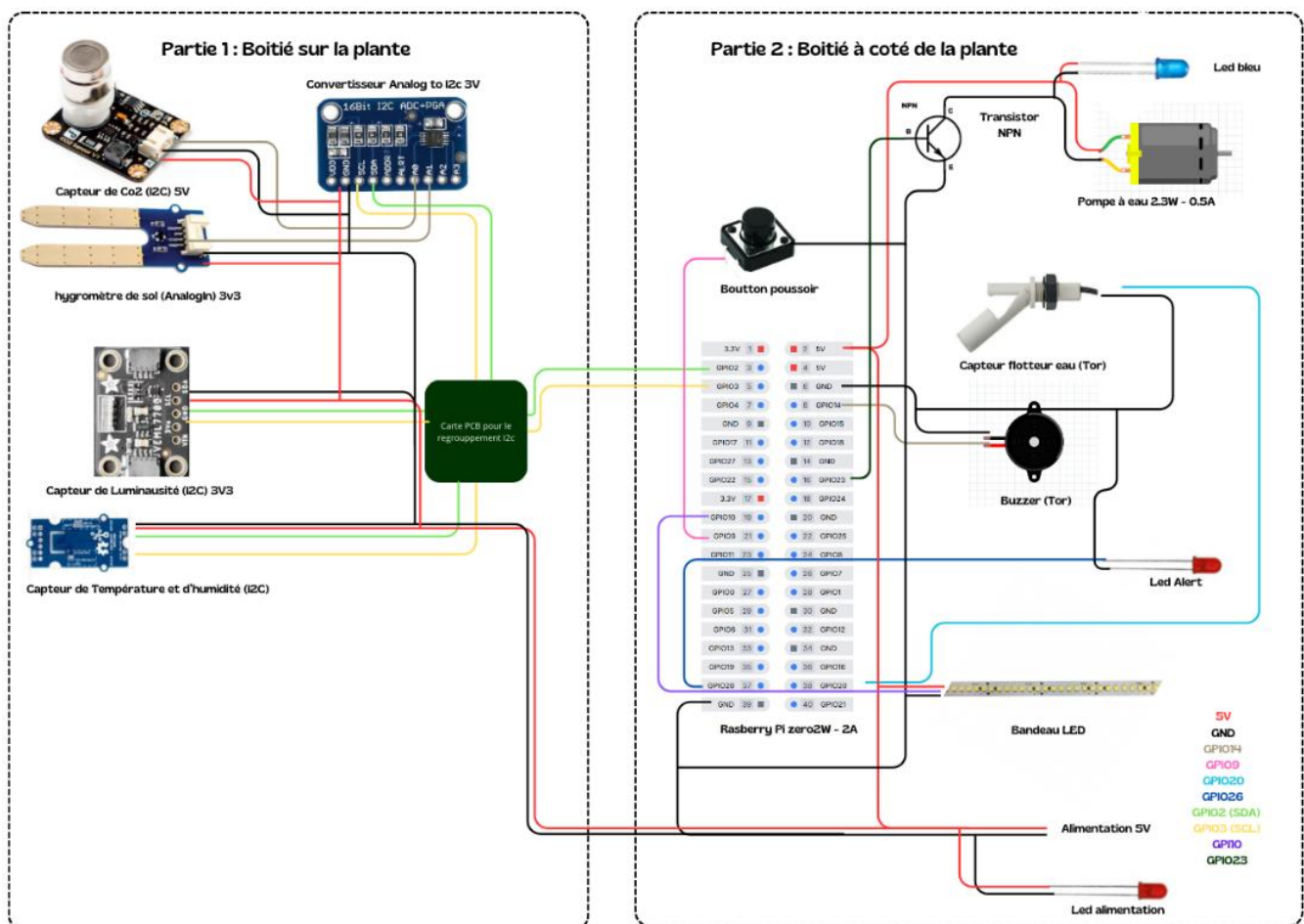
Dans cette architecture, il y aura deux boîtiers, chacun alimenté en 5V. J'ai fait ce choix car de nombreux éléments du projet ont une consommation énergétique importante, ce qui permet d'éviter l'utilisation d'un régulateur de tension.

Le premier boîtier, appelé "Boîtier Pompe", contient la Raspberry Pi Zero 2W ainsi que les actionneurs et capteurs tout ou rien, tels que les LEDs, le buzzer, le bouton poussoir et le flotteur. De plus, un bandeau LED sera utilisé pour afficher le taux d'humidité du sol, et une pompe de 2.3W sera intégrée pour l'arrosage.

Le second boîtier regroupe plusieurs capteurs environnementaux permettant de mesurer le CO<sub>2</sub>, la température, l'humidité de l'air, la luminosité et l'humidité du sol. Certains de ces capteurs étant analogiques, j'ai ajouté un convertisseur analogique-I2C afin de limiter le câblage à quatre fils entre les deux boîtiers : le 5V, le GND, ainsi que les deux fils du bus I2C (SDA et SCL).

Ce choix permet de simplifier le câblage et de faciliter la communication entre les deux boîtiers via le bus I2C.

Architecture global de la SAE v3



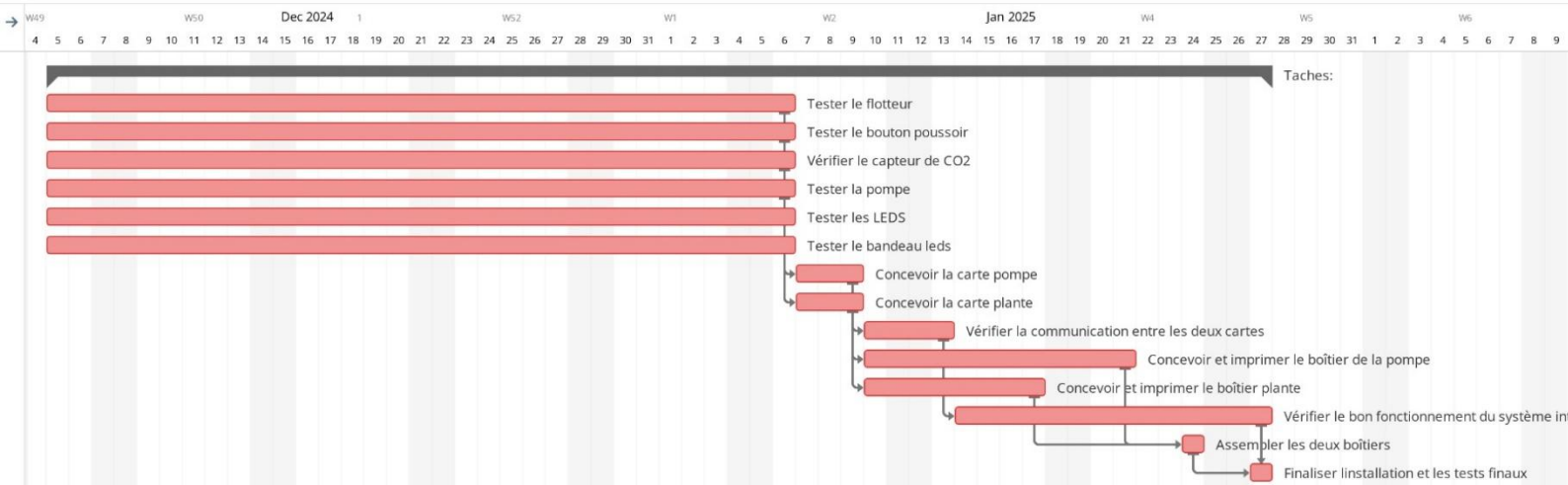
# Taches et objectifs

## Gantt

Taches Pichot

Read-only view, generated on 24 Feb 2025

Instagram



## Mes taches

Les taches que je dois réaliser :

- Tester le flotteur
- Tester le bouton poussoir
- Vérifier le capteur de CO<sub>2</sub>
- Tester la pompe
- Tester les LEDs
- Tester le bandeau LED
- Concevoir la carte pompe
- Concevoir la carte plante
- Vérifier la communication entre les deux cartes
- Concevoir et imprimer le boîtier de la pompe
- Concevoir et imprimer le boîtier plante
- Vérifier le bon fonctionnement du système intégré
- Assembler les deux boîtiers
- Finaliser l'installation et les tests finaux

## Mes objectifs :

Mes objectifs pour ce projet sont :

- Assurer la fiabilité des capteurs et actionneurs en testant leur bon fonctionnement avant l'intégration.
- Optimiser la conception des cartes électroniques en s'assurant qu'elles répondent aux besoins du projet tout en étant compactes et fonctionnelles
- Garantir une communication stable entre les cartes électroniques.
- Concevoir des boîtiers adaptés qui protègent les composants tout en facilitant leur installation et leur maintenance.
- Vérifier l'intégration complète du système pour garantir un fonctionnement fluide entre les capteurs, les actionneurs et l'électronique.
- Assembler et tester le système final afin de valider la conformité du projet et s'assurer qu'il fonctionne correctement dans son environnement réel.
- Rendre l'installation simple et accessible pour une utilisation facile, même pour une personne non technique.
- Faire des boîtiers esthétiques.



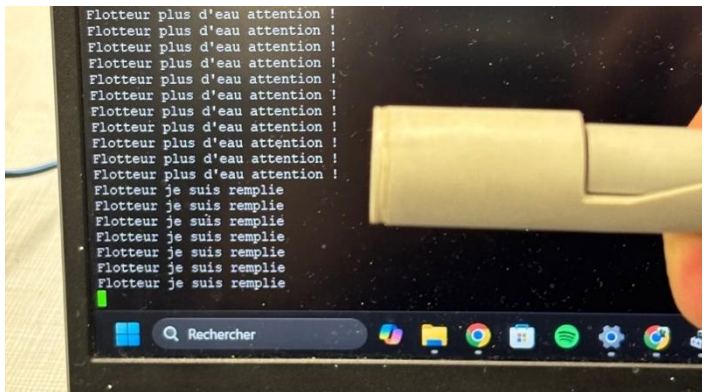
# Test de capteur et actionneurs unitaire

## Flotteur

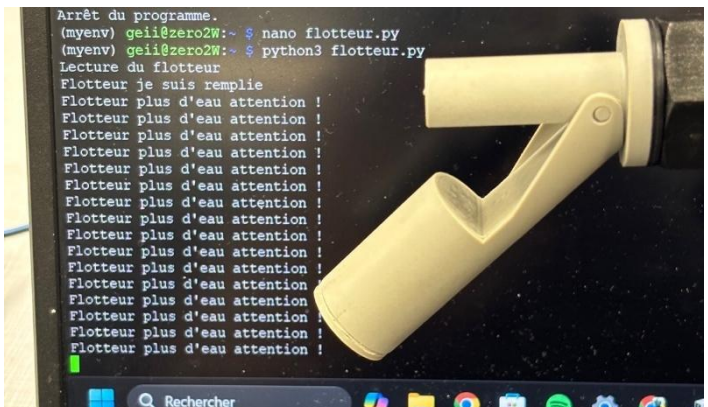
Le flotteur est un capteur tout ou rien (TOR), je l'ai donc branché directement sur une broche GPIO, avec son autre broche reliée au GND. Dans le code, j'ai activé une résistance de pull-up interne.

Le pull-up consiste à activer une résistance interne de la Raspberry Pi Zero 2W pour maintenir la broche GPIO à 3.3V par défaut.

- Lorsque le flotteur est fermé (en court-circuit), la broche est reliée à GND (0V), et la Raspberry Pi détecte un état bas (0).



- Lorsque le flotteur est ouvert, la broche n'est plus reliée à GND, elle reste donc à 3.3V, et la Raspberry Pi détecte un état haut (1).



## Leds

Pour le projet il y aura 3 leds :

- Led rouge pour les défauts (plus d'eau dans le réservoir, un capteur en défaut ou encore un problème de branchement avec le bus I2c)
- Led bleue quand la pompe est en marche
- Led rouge quand le boîtier est alimenté et que l'interrupteur est sur on

Pour le branchement des leds :

- Led rouge : la pat + sur un pin GPIO avec une résistance ohm et l'autre pat sur le gnd. On active le pull up interne. Et on crée appelle une fonction pour allumer la led pour 5s. J'ai mit une résistance de 68 ohm car  $R = \frac{3.3V - 2V}{0.02A}$
- Pour la led bleu elle sera directement en parallèle de la pompe pour que quand le transistor est fermé la led et la pompe marche en même donc pas besoin de code pour cette led mais il faudra quand même une résistance de 150 ohms car nous avons du 5v et plus du 3.3v donc  $R = \frac{5V - 2V}{0.02A}$
- Et pour la led d'alimentation elle sera en parallèle de l'alimentation 5v juste derrière de l'interrupteur ON/OFF et elle aura la même résistance de 150 ohms car notre alimentation est de 5V

```
geii@zero2W: ~  
GNU nano 7.2 led_random.py  
1 import RPi.GPIO as GPIO  
2 import time  
3  
4  
5 PIN = 26 # GPIO 26  
6 GPIO.setmode(GPIO.BCM)  
7 GPIO.setup(PIN, GPIO.OUT) # Définition de la  
8  
9 def led (duration=5):  
10     print("Led on")  
11     GPIO.output(PIN, GPIO.HIGH)  
12     time.sleep(duration)  
13     GPIO.output(PIN, GPIO.LOW)  
14     print("Led off")  
15  
16 try:  
17     led(5)  
18 finally:  
19     GPIO.cleanup()
```

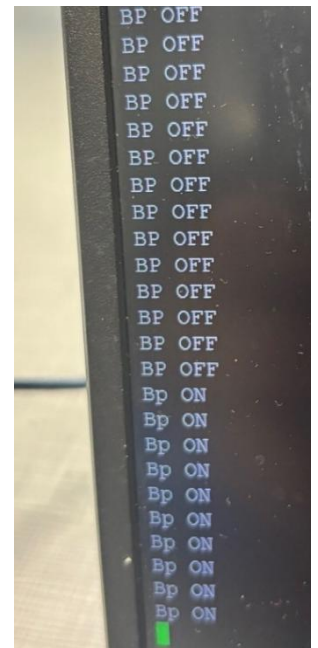
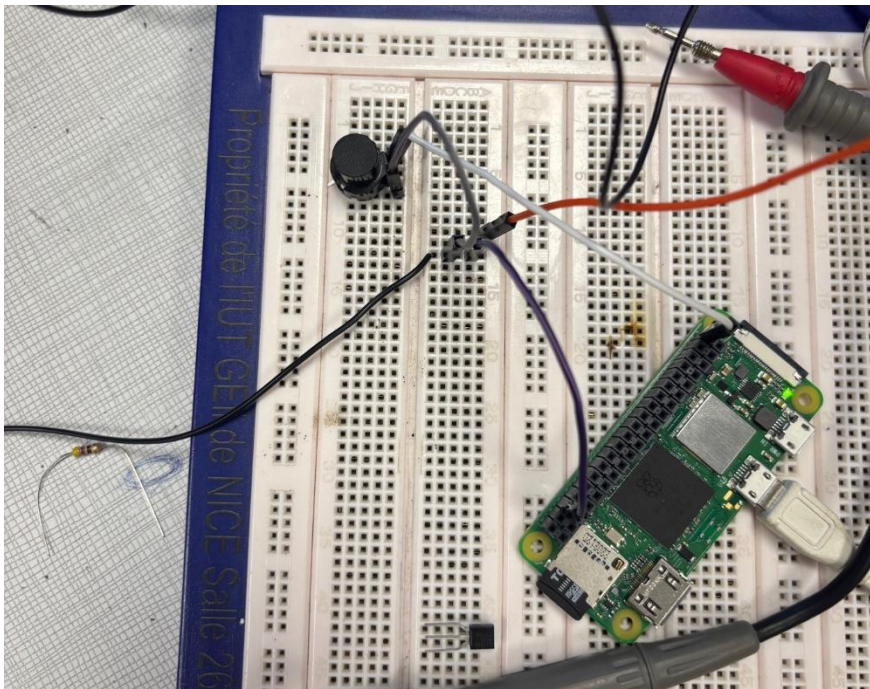


## Bouton poussoir

L'objectif de ce test est de vérifier le bon fonctionnement du bouton poussoir câblé en mode pull-up sur une broche GPIO de la Raspberry Pi Zero 2W. Le test consiste à s'assurer que le bouton est bien détecté par la Raspberry Pi que l'état du GPIO passe de 1 HIGH à 0 LOW lorsqu'on appuie sur le bouton et que le retour d'état est instantané.

Le bouton poussoir est connecté comme suit : une broche du bouton est reliée au GND, l'autre broche est connectée à la broche GPIO 12. Une résistance de pull-up interne est activée via le code pour maintenir le GPIO à 3.3V lorsqu'aucune action n'est effectuée.

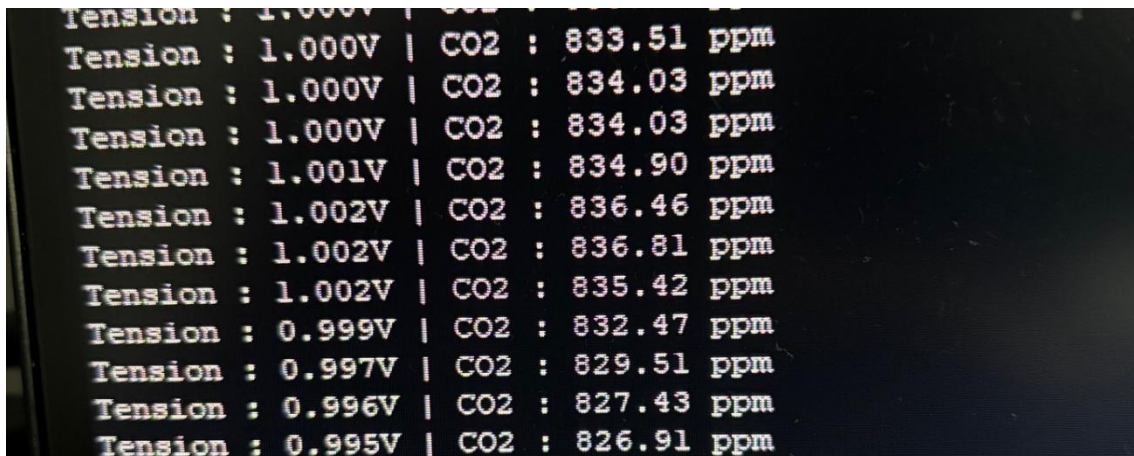
Le test est effectué à l'aide d'un code Python qui configure le GPIO 12 en entrée avec une résistance pull-up. L'affiche en temps réel l'état du bouton appuyé ou relâché et vérifie que le passage de HIGH 3.3V à LOW 0V est bien détecté.



## Capteur de CO2

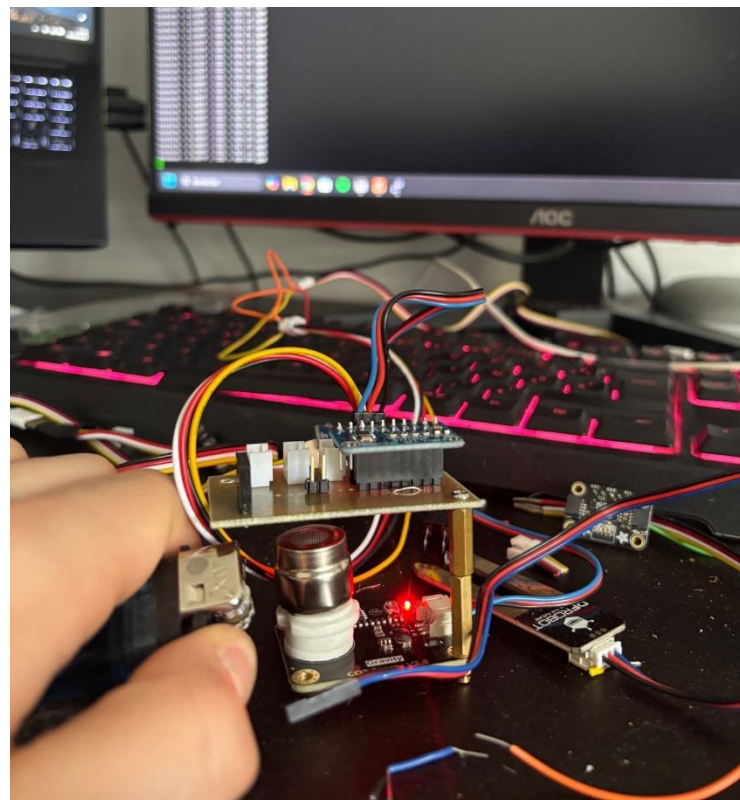
Le capteur utilisé dispose d'une sortie analogique ce qui nécessite l'utilisation d'un convertisseur analogique I2C pour permettre la lecture des valeurs via la Raspberry Pi Cette conversion est essentielle pour intégrer le capteur dans le système et assurer une transmission fiable des données au microcontrôleur

Le test consiste à vérifier plusieurs points clés la lecture correcte des données du capteur via le convertisseur analogique I2C l'activation de la LED d'alerte lorsque le seuil de CO2 est dépassé ainsi que la réactivité du système face aux variations du taux de CO2



Tension : 1.000V	CO2 : 833.51 ppm
Tension : 1.000V	CO2 : 834.03 ppm
Tension : 1.000V	CO2 : 834.03 ppm
Tension : 1.001V	CO2 : 834.90 ppm
Tension : 1.002V	CO2 : 836.46 ppm
Tension : 1.002V	CO2 : 836.81 ppm
Tension : 1.002V	CO2 : 835.42 ppm
Tension : 0.999V	CO2 : 832.47 ppm
Tension : 0.997V	CO2 : 829.51 ppm
Tension : 0.996V	CO2 : 827.43 ppm
Tension : 0.995V	CO2 : 826.91 ppm

Pour valider le test une source de CO2 a été simulée en approchant brièvement le gaz d'un briquet près du capteur L'augmentation du taux de CO2 a bien été détectée par le capteur et affichée sur la Raspberry Pi entraînant l'activation immédiate de la LED d'alerte Cette réaction confirme le bon fonctionnement du capteur du convertisseur et du système d'alerte



## Pompe

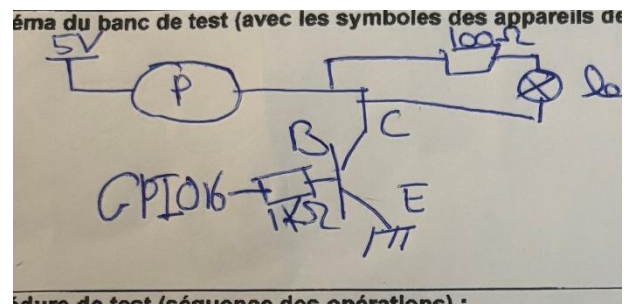
**QR50A/QR50B**  
**2.4W/4.8W**  
**USB DC5V**



L'objectif de ce test est de vérifier le bon fonctionnement de la pompe 5V, commandée via un transistor NPN 2N2222 connecté à une broche GPIO de la Raspberry Pi Zero 2W. Ce transistor est utilisé comme un interrupteur électronique, permettant d'activer ou de désactiver la pompe en fonction d'un signal envoyé par la Raspberry Pi. Le choix du 2N2222 est motivé par sa capacité à commuter jusqu'à 800mA, ce qui est suffisant pour une pompe de 2.3W sous 5V, soit environ 480mA. Il est également facilement activable avec un signal 3.3V, ce qui le rend parfaitement

compatible avec les GPIO de la Raspberry Pi. Contrairement à un relais, il ne consomme pas d'énergie en permanence et ne génère pas d'interférences électromagnétiques. De plus, les relais mécaniques ont une durée de vie plus courte et nécessitent souvent une alimentation séparée. L'utilisation d'un transistor de puissance comme un MOSFET ou un TIP120 n'était pas nécessaire, car la pompe a une consommation relativement faible, et un composant plus puissant aurait été surdimensionné. Le 2N2222 est donc un choix compact, économique et efficace pour cette application.

La pompe est connectée de la manière suivante : le collecteur du transistor est relié au pôle négatif de la pompe, l'émetteur est connecté à la masse (GND) et la base est reliée au GPIO 14 de la Raspberry Pi via une résistance de 1K $\Omega$ . Pour protéger le circuit contre les surtensions lors de l'arrêt de la pompe, une diode de roue libre 1N4007 est placée en parallèle avec la pompe, avec sa cathode vers le +5V et son anode vers le collecteur du transistor. Cette configuration garantit un fonctionnement stable et sécurisé de la pompe, tout en évitant les pics de tension qui pourraient endommager les composants électroniques.



Montage sur labdec



## Bandeau led

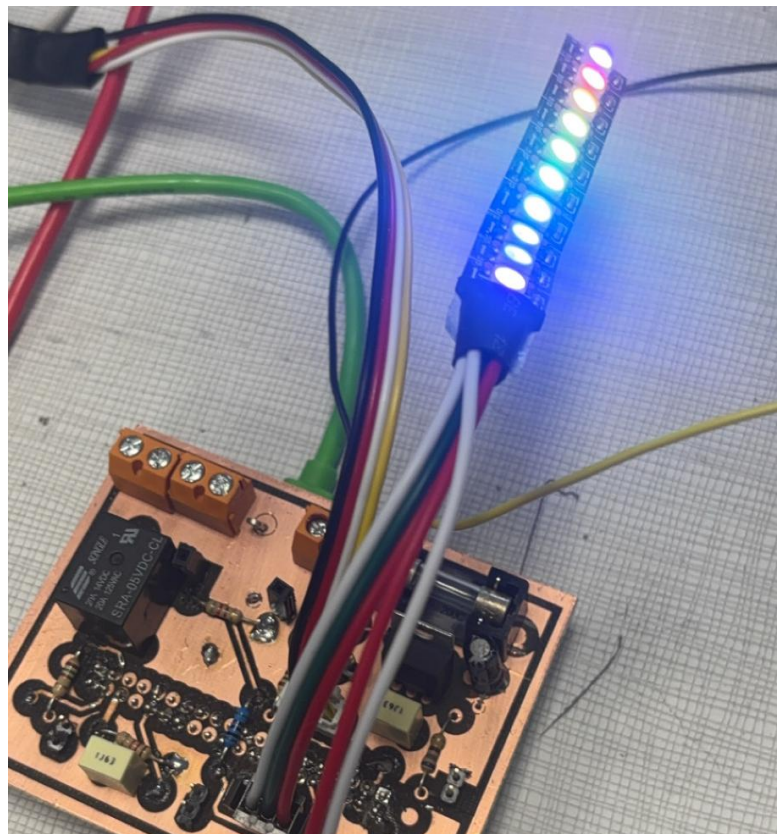
Le bandeau LED est utilisé pour afficher le niveau d'humidité du sol de manière visuelle. Il est composé de 10 LEDs, où chaque LED représente 10% d'humidité. Plus l'humidité est élevée, plus de LEDs s'allument, offrant ainsi une indication rapide et intuitive. Grâce à ce système, il est possible de surveiller l'état du sol en un coup d'œil, sans avoir besoin d'un écran ou d'un affichage numérique.

Le bandeau est alimenté en 5V et contrôlé via un GPIO de la Raspberry Pi Zero 2W. Une résistance est placée en série pour éviter les interférences, et un condensateur stabilise l'alimentation. Pour piloter les LEDs, la bibliothèque Adafruit\_WS2801 est utilisée. Cette librairie permet de contrôler chaque LED individuellement en envoyant des signaux numériques via un GPIO classique.

Le script Python utilisé initialise la connexion avec les LEDs en définissant le nombre total de pixels et en établissant la communication avec le bandeau. La ligne suivante configure le bandeau en définissant le nombre total de LEDs à 10 et en utilisant un bus SPI logiciel pour la communication

Grâce à au code donné par la bibliothèque, plusieurs effets lumineux sont appliqués pour afficher l'état du sol. Par exemple, la fonction `rainbow_cycle_successive()` affiche un dégradé de couleurs progressif, tandis que `rainbow_cycle()` fait défiler toutes les couleurs du spectre de manière fluide. La fonction `brightness_decrease()` permet de réduire progressivement l'intensité lumineuse des LEDs, et `blink_color()` génère un clignotement rapide d'une couleur spécifique. D'autres effets comme `appear_from_back()` donnent une impression d'allumage progressif des LEDs dans un sens précis.

L'utilisation de ces effets permet d'adapter l'affichage lumineux aux données du capteur d'humidité du sol. Par exemple, une variation du taux d'humidité entraîne un changement de couleur ou une animation spécifique sur le bandeau LED. Ce système offre un retour visuel efficace et dynamique, permettant une surveillance rapide et intuitive du niveau d'humidité du sol, sans nécessiter d'écran ou d'affichage numérique complexe.



# Carte électronique

## Explication

J'ai conçu la carte pompe pour centraliser la gestion de l'arrosage automatique tout en intégrant les différents composants nécessaires au fonctionnement du système. Elle permet de contrôler la pompe à eau, tout en protégeant l'ensemble du circuit grâce à un fusible de 2A contre les surtensions.

J'ai également intégré un bandeau LED qui affiche le niveau d'humidité du sol de manière visuelle, facilitant ainsi la lecture des données sans passer par un écran. Des LEDs indicatrices sont aussi présentes pour signaler l'état du système, comme l'activation de la pompe ou une alerte de niveau d'eau bas.

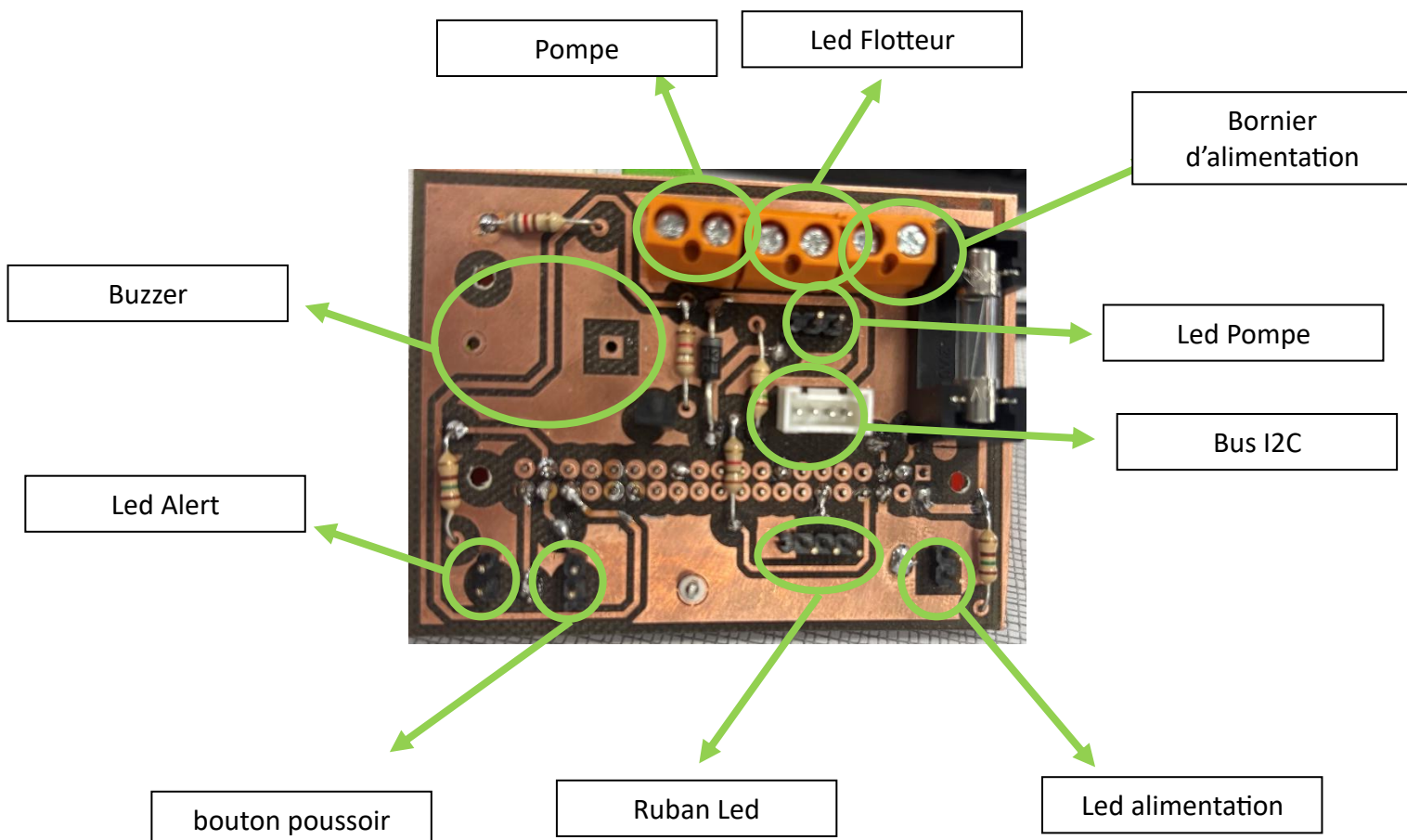
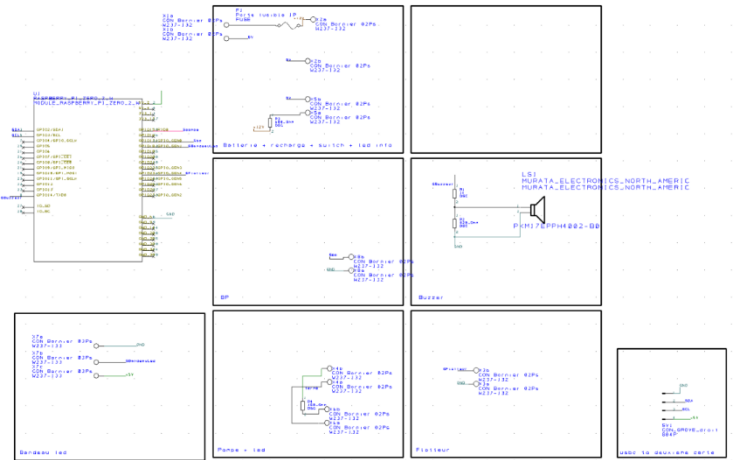
Un bouton poussoir (BP) permet d'interagir directement avec le système, par exemple pour forcer un arrosage manuel ou réinitialiser certaines fonctionnalités. Enfin, un connecteur Grove assure la communication avec le bus I2C, simplifiant ainsi le câblage et facilitant l'ajout de nouveaux capteurs. En regroupant tous ces éléments sur une seule carte, j'ai optimisé la fiabilité, réduit l'encombrement et facilité l'assemblage du système.

## Design spark

DesignSpark est une suite de logiciels gratuits développée par RS Components pour la conception électronique et mécanique. Elle comprend des outils pour créer des circuits imprimés (PCB), modéliser en 3D et réaliser des schémas électriques. Facile à utiliser, elle est adaptée aussi bien aux débutants qu'aux professionnels. Son interface intuitive permet un prototypage rapide et une compatibilité avec les formats standard. Totalement gratuite, elle est idéale pour des projets comme la conception de PCB, la modélisation 3D ou la domotique. Grâce à ses fonctionnalités complètes, DesignSpark est une alternative efficace aux logiciels payants.

## Carte Pompe

La carte est alimentée en 5V via un bornier, où je pourrai également brancher un câble USB-C. Pour protéger le circuit contre les surtensions, j'ai ajouté un fusible de 2A. Le buzzer fonctionnant en 1.5V, j'ai conçu un pont diviseur de tension pour l'adapter correctement. Les différentes connexions sont reliées aux broches GPIO de la Raspberry Pi. Un transistor 2N2222 est utilisé pour commander la pompe, et un connecteur Grove permet la communication via le bus I2C. Étant donné que mon PC a été volé, j'ai perdu mes cartes conçues sur DesignSpark. Heureusement, j'ai retrouvé un ancien schématic de la carte. Je vous partage donc la deuxième version de la carte.





## Carte Plante

La carte plante est conçue pour centraliser les mesures environnementales essentielles au bon développement de la plante. Elle intègre plusieurs capteurs, à la fois I2C et analogiques, permettant de surveiller en temps réel les conditions climatiques et l'humidité du sol. Ces données sont ensuite exploitées par la Raspberry Pi Zero 2W.

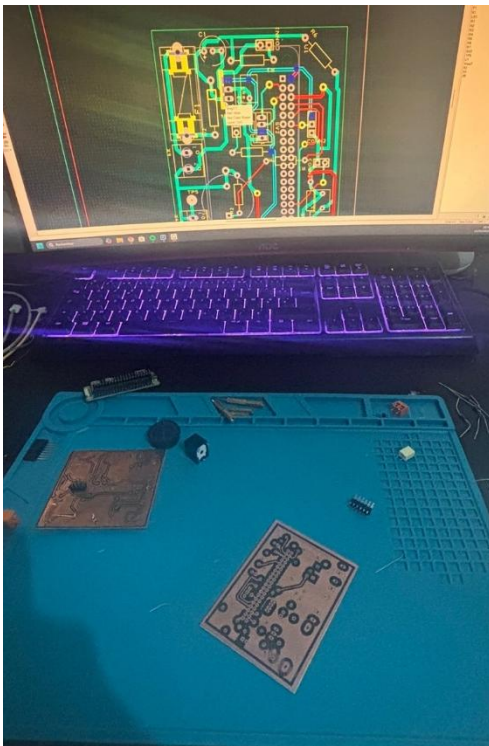
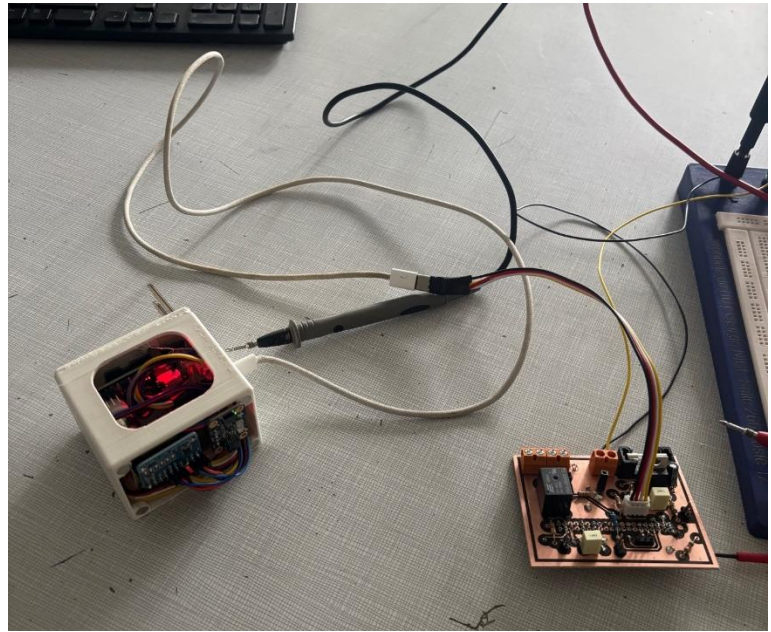
La carte comporte plusieurs capteurs I2C, qui utilisent seulement quatre fils : SDA, SCL, 5V et GND. Ces fils sont acheminés via un connecteur USB-C, simplifiant ainsi le câblage et assurant une connexion fiable entre la carte plante et la Raspberry Pi. Parmi ces capteurs, on retrouve un capteur de température et d'humidité, un capteur de CO2 pour surveiller la qualité de l'air, et un capteur de luminosité pour analyser l'exposition lumineuse de la plante. Ces mesures permettent d'adapter l'environnement et de mieux comprendre les besoins de la plante.

Pour la gestion des capteurs analogiques, la carte est équipée d'un convertisseur analogique-numérique ADS, qui permet à la Raspberry Pi d'interpréter ces données. On retrouve un hygromètre capacitif et le capteur de CO2



## Soudage et tests des 2 cartes

Une fois la carte plante et la carte pompe soudées, j'ai procédé à des tests pour vérifier leur bon fonctionnement. J'ai d'abord testé si les deux cartes communiquaient correctement entre elles, en s'assurant que les données circulaient bien via le bus I2C. Ensuite, j'ai vérifié le bon fonctionnement des capteurs, en m'assurant qu'ils répondaient correctement aux sollicitations et transmettaient des valeurs cohérentes. Ces tests ont permis de confirmer que le système était opérationnel et que l'intégration des composants était réussie.

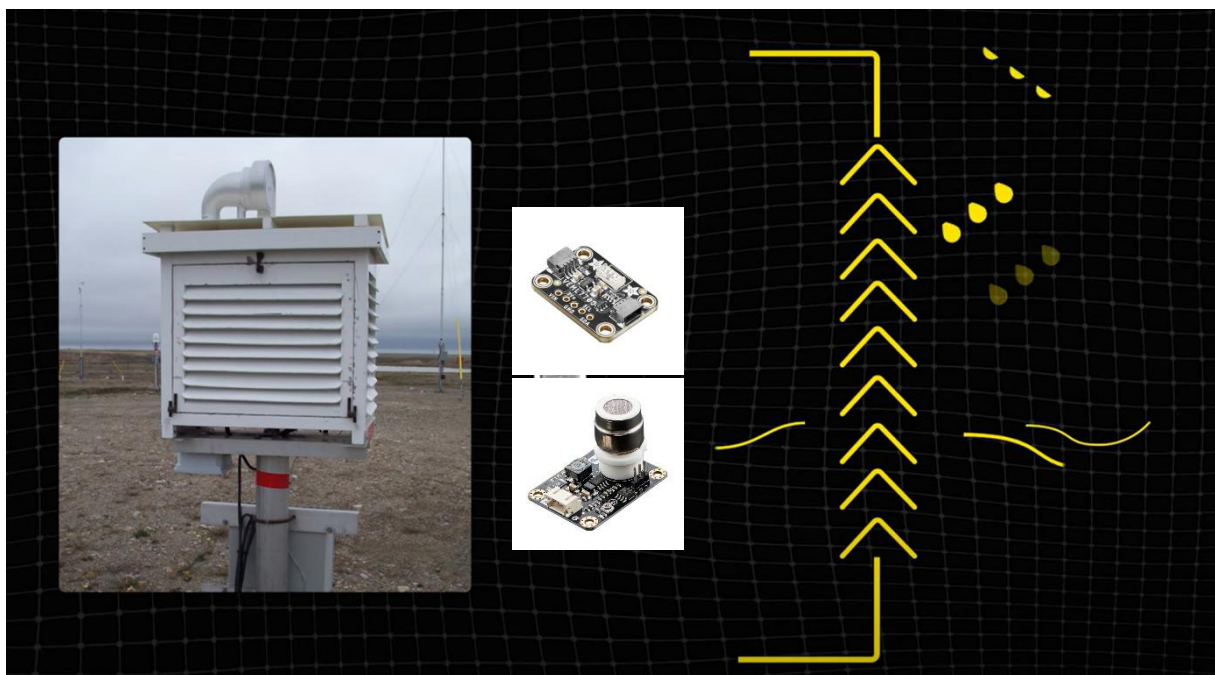


J'ai soudé deux cartes électroniques, chacune avec une conception différente pour répondre aux besoins du projet. La première carte est une simple face avec des vias pour relier certaines pistes entre les couches. J'ai commencé par souder les composants traversants en insérant leurs broches dans les trous avant d'appliquer la soudure sur la face opposée. Les vias ont été réalisés en étamant les trous pour assurer une bonne conductivité entre les connexions. La seconde carte, en double face, a posé plus de difficultés. J'ai rencontré des problèmes techniques, notamment des erreurs d'alignement et des défauts de gravure, ce qui m'a obligé à réimprimer la carte trois fois au FabLab avant d'obtenir un résultat satisfaisant. À travers ce processus, j'ai appris à utiliser la graveuse laser, ce qui m'a permis d'améliorer la précision des tracés et l'optimisation des pistes sur la carte. Après l'assemblage, j'ai vérifié chaque connexion avec un multimètre pour m'assurer de la continuité des pistes et détecter d'éventuels courts-circuits.

# Impression 3d

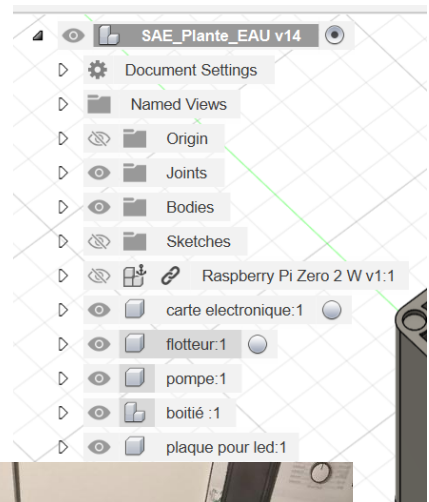
## Explication

Le premier boîtier (Boîtié plante) sera placé à l'extérieur planté soit dans la terre du jardin soit la terre d'un pote de fleur ou encore une jardinière je me suis inspiré de l'invention de Thomas Stevenson, qui a conçu une station météorologique équipée de petites lames en forme de dents permettant à l'air de circuler tout en protégeant les capteurs de la pluie. Il a également peint la structure en blanc afin de réfléchir le rayonnement solaire et ainsi limiter l'accumulation de chaleur à l'intérieur. J'ai repris ce principe pour optimiser la protection et le bon fonctionnement des capteurs dans mon projet.



Le second boîtier sera quant à lui placé à l'abri de la pluie. Les deux boîtiers seront reliés par un câble USB-C, dont la longueur pourra être étendue sans problème, tant qu'elle ne dépasse pas 3 mètres, afin d'assurer une communication stable entre le bus I2C.

J'ai utilisé Fusion 360 pour modéliser l'ensemble de mes pièces 3D. Pour les deux boîtiers, j'ai également modélisé les composants qu'ils contiennent afin d'avoir une vue globale précise de leur agencement et de mieux anticiper l'intégration des différents éléments.



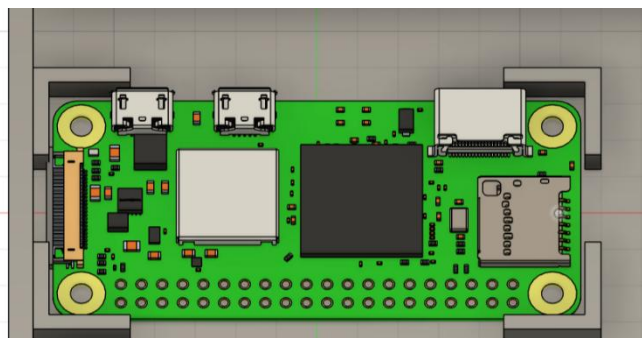
## Problème

J'ai rencontré un problème avec mon imprimante 3D, qui s'est bouchée, empêchant toute extrusion de filament. Après plusieurs tentatives de nettoyage sans succès, j'ai dû démonter l'extrudeur pour identifier la source du blocage. La buse étant trop obstruée, j'ai décidé d'en acheter une nouvelle et de la remplacer. Depuis ce changement, l'imprimante fonctionne beaucoup mieux, l'extrusion est fluide et les impressions sont plus propres et régulières.



## Boîtier pompe

Dans le boîtier pompe, plusieurs éléments seront intégrés, notamment la pompe, l'alimentation et l'IHM. J'ai modélisé les différents éléments et même pour la raspberry pi zero 2W j'ai trouvé un modèle 3d sur internet.





La pompe sera fixée à l'aide de vis, avec des emplacements prévus pour des entretoises M2 afin d'assurer une installation stable.

L'IHM comprendra un bandeau LED RGB à forte densité de LEDs. Pour obtenir un effet diffus et homogène, un cache bandeau LED de 1 mm d'épaisseur a été collé, permettant de laisser passer la lumière tout en l'adoucissant.

Des ouvertures ont été prévues pour différents composants :

- Le bouton poussoir
- Les différentes LED d'indication
- L'embout femelle USB-C
- Interrupteur ON/OFF
- Un emplacement pour le flotteur, qui sera immergé dans l'eau
- La pompe

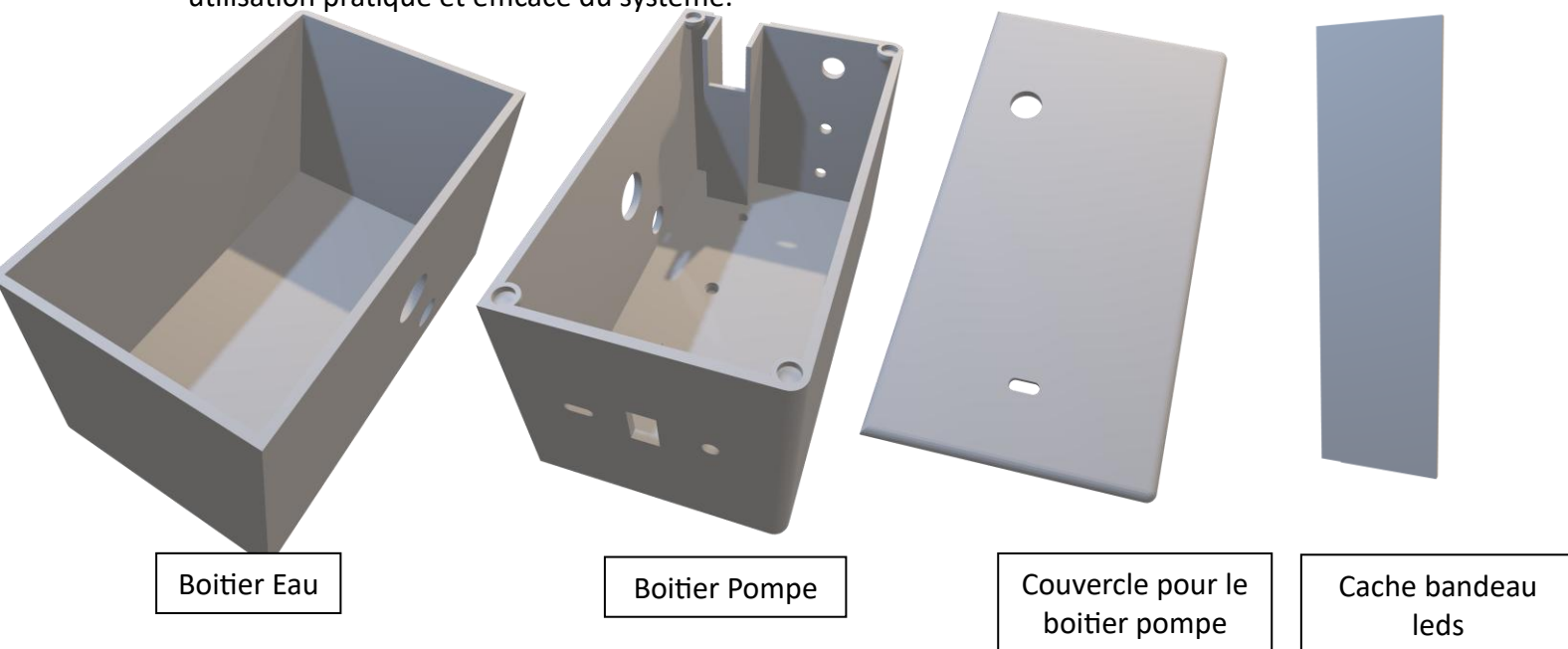
Le boîtier d'eau fait en TPU 95A sera fixé au boîtier de la pompe à l'aide d'une colle époxy bi-composant, la colle et le TPU est spécialement conçue pour une utilisation immergée. Cela permettra à la pompe de prélever directement l'eau dans le boîtier d'eau, tandis que le flotteur y restera immergé pour détecter le niveau d'eau.

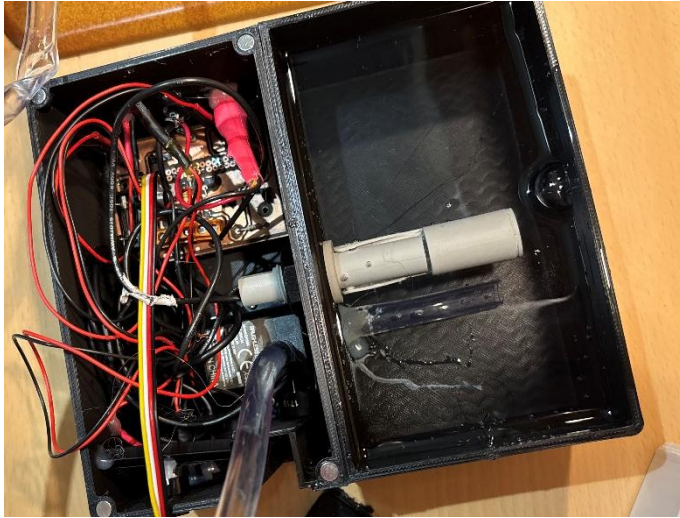


Le boîtier sera fermé par un couvercle aimanté, avec des trous prévus pour :

- L'évacuation de l'eau via le tuyau à eau
- L'embout femelle USB-C pour le bus I2C

Grâce à cette conception, il est assez simple de remplir la cuve à eau, garantissant une utilisation pratique et efficace du système.





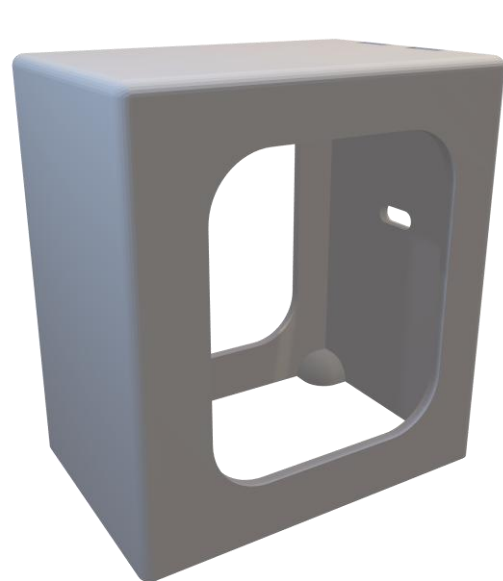
Voici le rendu final  
des deux boîtiers  
assembler

## Boitier Plante

Dans ce boîtier j'ai tous les capteurs i2c

Je l'imprime en PLA blanc pour éviter que la chaleur déforme les valeurs des capteurs

Et j'ai fait des trous pour insérer des fenêtres pour que le vent circule dans la boîte et que la pluie ne rentre pas cette boîte va directement dans le pot de plante.



Boitier plante



Couvercle boîtier  
plante



Fenêtre





Rendu imprimer  
sans les fenêtres

# Conclusion

Ce projet d'arrosage automatique intelligent, réalisé en binôme, nous a permis de concevoir un système autonome et optimisé pour surveiller et entretenir une plante. Grâce aux capteurs I2C et analogiques, la carte plante collecte des données sur l'humidité du sol, la température, la luminosité et le CO<sub>2</sub>, tandis que la carte pompe contrôle l'arrosage de manière efficace.

Avec un coût total de 163 €, ce système est fonctionnel et évolutif. Il pourrait être amélioré en réglant la PWM de la pompe pour ajuster le débit d'eau, ou encore en imprimant un embout sur le tuyau afin de transformer le flux en gouttelettes, évitant ainsi un arrosage trop brutal. Ces optimisations permettraient d'améliorer l'efficacité et la précision du système en fonction des besoins de la plante.

Ce projet a également permis d'approfondir mes compétences en électronique, programmation et modélisation 3D, en travaillant sur la conception de circuits imprimés, la gestion des capteurs et la fabrication des boîtiers via l'impression 3D. Il a nécessité une approche pluridisciplinaire, mêlant conception matérielle et développement logiciel, ainsi qu'une bonne gestion du travail en binôme pour assurer l'aboutissement du projet. Cette expérience a été enrichissante.

éléments	quantité	prix unitaire	prix tot
USB type c	1	3.88 €	3.88 €
ADS 1115 ADC	1	3.89 €	3.89 €
Ruban Led	1	7.25 €	7.25 €
PLA blanche	0.3	17.99 €	5.40 €
TPU 95 noir	0.2	15.99 €	3.20 €
PLA noir	0.4	17.99 €	7.20 €
pompe	1	5.79 €	5.79 €
cole epoxi bi composants	1	13.00 €	13.00 €
flotteur	1	19.86 €	19.86 €
capteur de Co2	1	58.90 €	58.90 €
capteur de luminosité	1	4.00 €	4.00 €
Entretoise M2	0.2	3.00 €	0.60 €
Raspberry Pi zero 2W	1	20.00 €	20.00 €
Cable USB-C to USB-C	1	10.00 €	10.00 €
		TOTAL :	162.96 €

# Annexe

(Lien git du projet : <https://github.com/Pichot06/SAEPlante>)

## Procédure de test pour pompe

Institut  
Universitaire  
De Technologie  
Nice - Côte d'Azur

(12/01)

**Fiche de mesures**

Référence de la mesure (nom du projet, numéro de fiche, date, nom de l'auteur) :  
SAE Plante

Module ou Éléments sous test :  
Pompe avec transistor 2N2222

Objectif du test :  
activer le transistor avec une PIN de la raspberry

Schéma du banc de test (avec les symboles des appareils de mesure) :

Procédure de test (séquence des opérations) :  
mettre a 0 et a 1 la pin

Résultats de la mesure (oscillogramme ou tableau de mesures ou caractéristique) :

0	1
led éteinte pompe "	led marche pompe "

Commentaires des mesures/ Conclusion  
Marche !

Prix avec les liens :

éléments	quantité	prix unitaire	prix tot	lien
USB type c	1	3.88 €	3.88 €	<a href="https://www.aliexpress.com/p/order/detail.html?spm=a2g0o.order_list.o">https://www.aliexpress.com/p/order/detail.html?spm=a2g0o.order_list.o</a>
ADS 1115 ADC	1	3.89 €	3.89 €	<a href="https://www.aliexpress.com/p/order/detail.html?spm=a2g0o.order_list.o">https://www.aliexpress.com/p/order/detail.html?spm=a2g0o.order_list.o</a>
Ruban Led	1	7.25 €	7.25 €	<a href="https://www.aliexpress.com/p/order/detail.html?spm=a2g0o.order_list.o">https://www.aliexpress.com/p/order/detail.html?spm=a2g0o.order_list.o</a>
PLA blanche	0.3	17.99 €	5.40 €	<a href="https://www.amazon.fr/eSUN-Filament-Pr%C3%A9cision-Dimensionnell3cuUTVQM0QG0Z8lZNXpKCyz972r4bNoQEVzQSxANM_NYYa5ObGJGmIZLEKG6Gssm5aCb2IP8gU4t7yztCQdZJCyLfDV1Ug5Uc90_bKmTYYdLw7l69&amp;sr=1-1">https://www.amazon.fr/eSUN-Filament-Pr%C3%A9cision-Dimensionnell3cuUTVQM0QG0Z8lZNXpKCyz972r4bNoQEVzQSxANM_NYYa5ObGJGmIZLEKG6Gssm5aCb2IP8gU4t7yztCQdZJCyLfDV1Ug5Uc90_bKmTYYdLw7l69&amp;sr=1-1</a>
TPU 95 noir	0.2	15.99 €	3.20 €	TPU Flexible Filament 1.75mm, JAYO Imprimante 3D Filament TPU Shore 95
PLA noir	0.4	17.99 €	7.20 €	<a href="https://www.aliexpress.com/p/order/detail.html?spm=a2g0o.order_list.o">https://www.aliexpress.com/p/order/detail.html?spm=a2g0o.order_list.o</a>
pompe	1	5.79 €	5.79 €	<a href="https://www.aliexpress.com/p/order/detail.html?spm=a2g0o.order_list.o">https://www.aliexpress.com/p/order/detail.html?spm=a2g0o.order_list.o</a>
cole epoxi bi composants	1	13.00 €	13.00 €	Amazon
flotteur	1	19.86 €	19.86 €	<a href="#">944 pour 'flotteur'   RS (rs-online.com)</a>
capteur de Co2	1	58.90 €	58.90 €	<a href="https://fr.rs-online.com/web/p/kits-de-developpement-pour-capteur/201">https://fr.rs-online.com/web/p/kits-de-developpement-pour-capteur/201</a>
capteur de luminosité	1	4.00 €	4.00 €	<a href="https://fr.aliexpress.com/item/1005005981771097.html?spm=a2g0o.productDetail&amp;pdp_ext_f=%7B%22order%22%3A%223%22%2C%22eval%22%3A%22url%3Asearch%7Cquery_from%3A">https://fr.aliexpress.com/item/1005005981771097.html?spm=a2g0o.productDetail&amp;pdp_ext_f=%7B%22order%22%3A%223%22%2C%22eval%22%3A%22url%3Asearch%7Cquery_from%3A</a>
Entretoise M2	0.2	3.00 €	0.60 €	Aliexpress
Raspberry Pi zero 2W	1	20.00 €	20.00 €	<a href="https://fr.aliexpress.com/item/1005005792181612.html?spm=a2g0o.productDetail&amp;pdp_ext_f=%7B%22order%22%3A%223165%22%2C%22eval%22%3A%22url%3Asearch%7Cquery_from%3A">https://fr.aliexpress.com/item/1005005792181612.html?spm=a2g0o.productDetail&amp;pdp_ext_f=%7B%22order%22%3A%223165%22%2C%22eval%22%3A%22url%3Asearch%7Cquery_from%3A</a>
Cable USB-C to USB-C	1	10.00 €	10.00 €	
		TOTAL :	162.96 €	

Source :

[https://www.youtube.com/watch?v=x5MdpQb1MA0&t=1719s&ab\\_channel=Abr%C3%A8ge](https://www.youtube.com/watch?v=x5MdpQb1MA0&t=1719s&ab_channel=Abr%C3%A8ge)