



PREMIERE GENERALE

NUMERIQUE ET SCIENCES INFORMATIQUES

MINI-PROJET : SYSTEME DE SURVEILLANCE
POUR PLANTES

1 – FLOWER POWER

Quand arroser ses plantes ? Ont-elles besoin d'engrais, de lumière, de chaleur ? Le « Flower Power » fabriqué par Parrot permet de répondre à ces questions.

Planté dans un pot cet objet permet de surveiller les 4 facteurs essentiels à leur bien-être : **luminosité**, **température de l'air**, **teneur en engrais** et **humidité du substrat**.

Quatre capteurs intégrés recueillent des données toutes les 15 minutes qui sont transmises vers des appareils Bluetooth pour signaler en temps réel, via une application mobile les besoins de la plante.

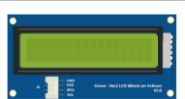


2 – MINI-PROJET

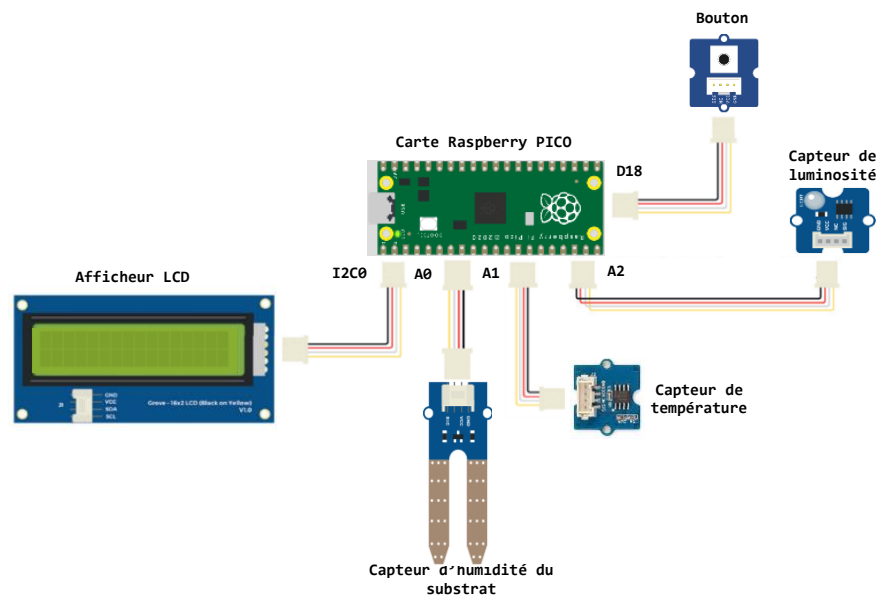
2.1 – Présentation du mini-projet

Le mini-projet consiste à mettre en œuvre un système qui permette la mesure et l'affichage de la valeur de trois facteurs essentiels au bien-être des plantes : **luminosité**, **température de l'air** et **humidité du substrat**.

2.2 – Matériel disponible

Carte Raspberry PICO.	
Capteur d'humidité du substrat grove.	
Capteur de température grove.	
Capteur de luminosité	
Afficheur LCD I2C Grove	

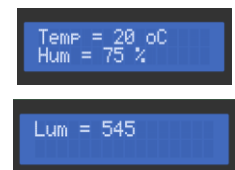
2.3 – Câblage des composants



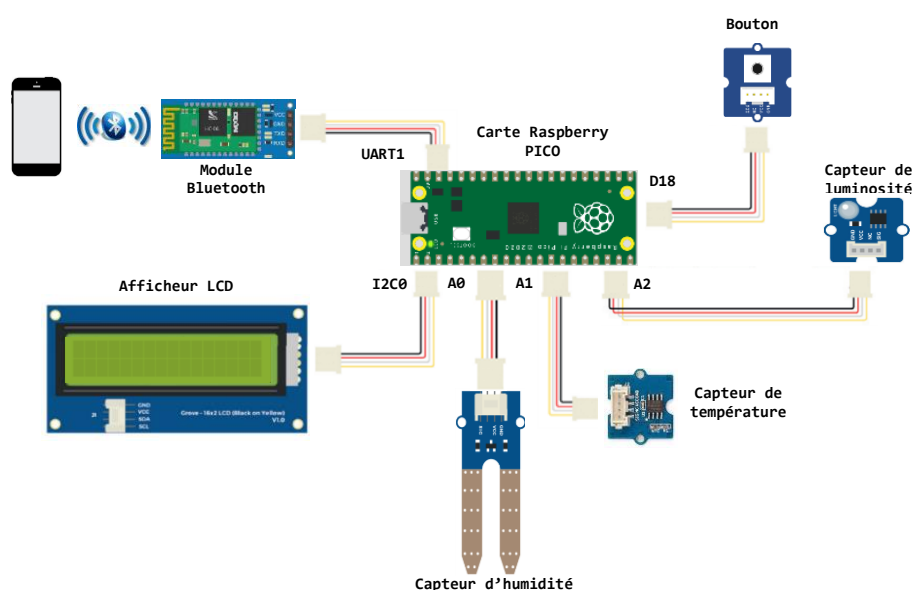
2.3 – Cahier des charges

- Le système devient fonctionnel après un appui sur le BP. Un nouvel appui désactive le système de surveillance.

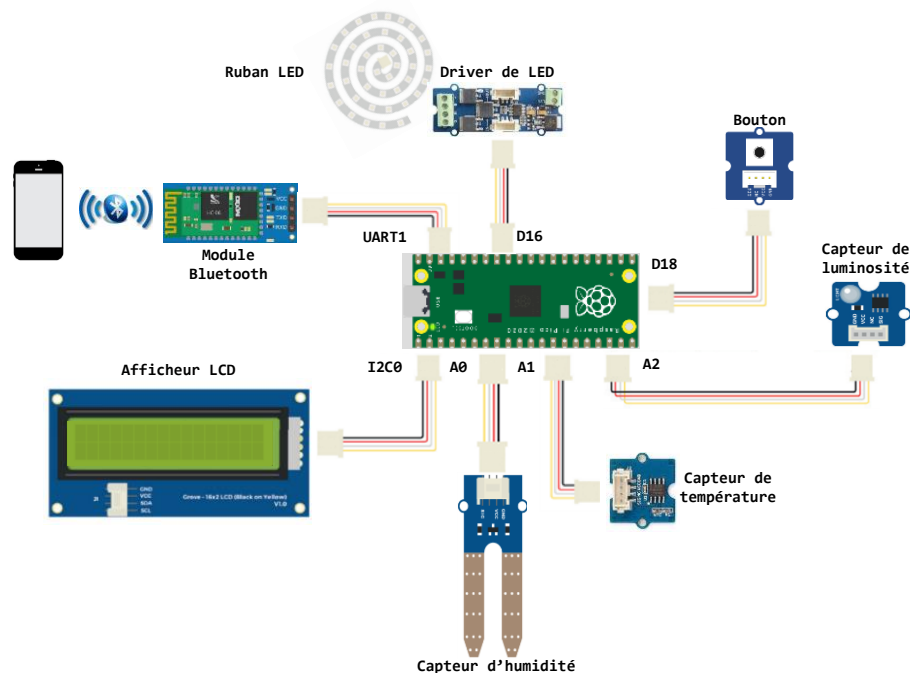
- On affichera alternativement (en boucle) pendant 5s les valeurs de la température et du taux d'humidité du substrat puis la valeur de la luminosité.



- Pour aller plus loin : Afficher les différentes mesures sur une application mobile. La transmission des données se fera à l'aide du module Bluetooth HC06.



- Pour aller plus loin : Commander l'allumage d'un éclairage à LED si la luminosité est inférieure à un seuil prédéfini. Cet allumage pourra également être commandé via l'application mobile.



3 – COMPOSANTS

3.1 – Capteur d'humidité du sol grove

Le capteur d'humidité grove est un détecteur d'humidité résistif qui peut déterminer s'il y a de l'humidité autour du capteur. Ce capteur est doté de deux sondes qui permettent au courant de passer à travers le sol, puis d'obtenir des valeurs de résistance pour mesurer la teneur en humidité du sol.



Caractéristiques	Valeurs	
Valeur de sortie capteur en sol sec	Min	0
	Max	300
Valeur de sortie capteur en sol humide	Min	300
	Max	700
Valeur de sortie capteur dans l'eau	Min	700
	Max	950

Ce capteur fournit une tension analogique fonction de l'humidité du sol. Pour récupérer cette tension il faut créer un objet **ADC** en précisant la broche analogique utilisée puis utiliser la méthode **read_u16()**.

Par exemple pour récupérer la valeur **N** correspondant à la tension sur la voie **ADC0** il faut exécuter les instructions ci-contre.

```
from machine import ADC
adc = ADC(0) # Utilisation de la voie ADC0
N = adc.read_u16()
```

Brochage



Broche	Nom	Câble	Rôle
1	SIG	Jaune	Sortie analogique
2	NC	Blanc	Non connectée
3	VCC	Rouge	Alimentation
4	GND	Noir	Masse (0V°)

3.2 – Capteur de luminosité grove

Ce capteur de lumière permet de détecter la présence de lumière. La tension de sortie analogique évolue de 0 à 3,3V suivant l'intensité lumineuse mesurée. Il intègre une LDR (Light Dependent Resistor) dont la résistance va varier en fonction de l'intensité lumineuse.

La valeur mesurée n'est qu'une valeur approximative qui ne représente pas l'intensité lumineuse exacte.



Ce capteur fournit une tension analogique fonction de la luminosité. Pour récupérer cette tension il faut créer un objet **ADC** en précisant la broche analogique utilisée puis utiliser la méthode `read_u16()`.

Brochage



Broche	Nom	Câble	Rôle
1	SIG	Jaune	Sortie analogique
2	NC	Blanc	Non connectée
3	VCC	Rouge	Alimentation
4	GND	Noir	Masse (0V°)

3.3 – Capteur de température analogique grove

Ce capteur de température délivre un signal analogique entre 0 à 5 Vcc en fonction de la température mesurée. Ce capteur utilise une thermistance pour mesurer la température ambiante. La résistance d'une thermistance augmente lorsque la température ambiante diminue. La plage de températures mesurables par ce capteur est de -40 à 125 °C et la précision est de ±1,5 °C.



Ce capteur fournit une tension analogique fonction de la température. Pour récupérer cette tension il faut créer un objet **ADC** en précisant la broche analogique utilisée puis utiliser la méthode `read_u16()`.

La valeur de la thermistance R_T est déterminée à partir de la valeur N (entier compris entre 0 et 65535) obtenue lors de la lecture de la tension en sortie du capteur :

$$R_T = R_0 \left(\frac{65535}{N} - 1 \right) \quad \text{avec } R_0 = 100 \text{ k}\Omega$$

La valeur de la température est alors obtenue à partir de la relation :

$$T = \frac{1}{\frac{\log\left(\frac{R_T}{R_0}\right)}{B} + \frac{1}{298.15}} - 273.15 \quad \text{avec } B = 4275$$

Brochage



Broche	Nom	Câble	Rôle
1	SIG	Jaune	Sortie analogique
2	NC	Blanc	Non connectée
3	VCC	Rouge	Alimentation
4	GND	Noir	Masse (0V°)

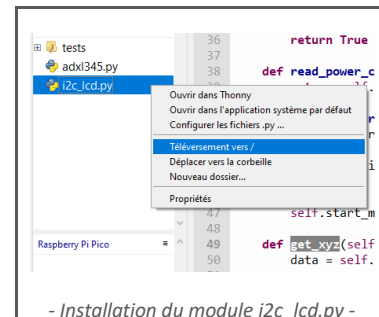
3.4 – Afficheur LCD

L'afficheur LCD doit être connecté sur le bus I2C de la carte Raspberry PICO. Il permet un affichage de 2 lignes de 16 caractères blancs sur un rétroéclairage bleu.



Le module, **i2c_lcd** permet d'utiliser cet afficheur.

Pour pouvoir utiliser ce module il faut le téléverser dans la carte Raspberry PICO. Pour cela rechercher, dans la zone « Fichiers » le fichier **i2c_lcd.py** puis avec le bouton droit de la souris cliquer sur « Téléversement vers/ ».



- Installation du module i2c_lcd.py -

L'afficheur communique avec une liaison I2C. Il est donc nécessaire de créer un objet **I2C** en spécifiant les broches utilisées pour l'interface I2C et éventuellement la fréquence de la liaison. Il faut ensuite créer un objet **lcd** qui nous permettra de gérer l'afficheur.

```
from machine import Pin, I2C
from i2c_lcd import lcd

i2c = I2C(0, scl = Pin(9), sda = Pin(8), freq = 400000)
display = lcd(i2c)
```

Les principales méthodes de ce module sont :

Methode	Rôle	Exemple
lcd()	Création d'un objet de la classe lcd permettant de piloter l'afficheur	display = lcd()
write_char(c)	Affichage du caractère c	display.write_char('T')
write(chaine)	Affichage la chaine de caractères chaine	display.write('Temp')
cursor(s)	Curseur visible si s = 1 (par défaut) et invisible si s = 0	display.cursor(0)
setCursor(x, y)	Placement du curseur sur la colonne x et la ligne y	display.setCursor(0, 1)
display(s)	Affichage visible si s = 1 (par défaut) et invisible si s = 0	display.display(0)
clear()	Permet d'effacer l'affichage	display.clear()
home()	Placement du curseur sur la colonne 0 et la ligne 0	display.home()

Brochage



Broche	Nom	Câble	Rôle
1	SCL	Jaune	Ligne Horloge liaison I2C
2	SDA	Blanc	Ligne Données liaison I2C
3	VCC	Rouge	Alimentation
4	GND	Noir	Masse (0V°)

3.5 – Module bluetooth HC06

Le module Bluetooth HC06 doit être connecté sur le bus UART de la carte Raspberry PICO. Pour le projet, c'est l'interface UART1 qui sera utilisée.



Les interfaces UART sont gérées par la classe UART de la librairie machine. Il faut créer un objet de type UART, sur lequel seront appliqués les différentes méthodes, en spécifiant le numéro de l'interface, la vitesse de communication (paramètre **baudrate**) et le numéro des broches **Rx** et **Tx**.

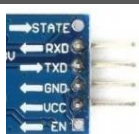
```
from machine import Pin, UART

bluetooth = UART(1, baudrate = 9600, tx = Pin(4), rx = Pin(5))
```

Les principales méthodes de cette classe sont :

Methode	Rôle	Exemple
read()	Lecture de la donnée reçue sur la ligne Rx.	donnee = bluetooth.read()
write(donnee)	Envoi de la chaîne de caractères stockée dans donnees sur la ligne Tx.	bluetooth.write('Temp')

Brochage



Broche	Nom	Rôle
1	RXD	Ligne Réception à connecter sur broche Tx Raspberry
2	TXD	Ligne Emission à connecter sur broche Rx Raspberry
3	GND	Masse (0V)
4	VCC	Alimentation