

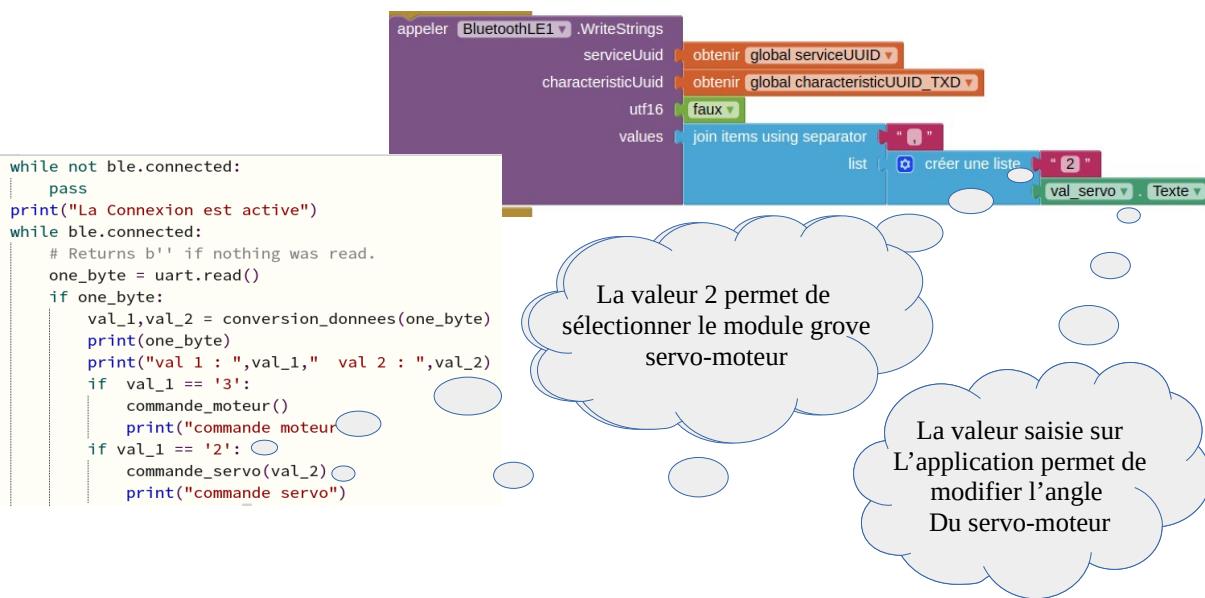
# Projet de communication en Bluetooth Ble de la carte Circuitpython avec AppInventor

Aide de Christophe John Ward, Utilisateur avancé

Pascal Mathieu, enseignant en lycée dans la spé SI et NSI

Vous trouverez ci-dessous le programme en python pour la carte metro M4 Airlift Lite et le code de l'application AppInventor

La communication se fait essentiellement avec des chaînes de caractères dans les deux sens.



## Programme Python

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT
```

```
# Basic example for using the BLE Connect UART
# To use, start this program, and start the Adafruit Bluefruit LE Connect app.
# Connect, and then select UART. Any text received FROM the connected device
# will be displayed. Periodically, text is sent TO the connected device.
```

```
# Programme de commande lecture et Ecriture par le Bluetooth Ble avec AppInventor
# Attention il faut utiliser le logiciel Bluefruit LE Connect sur votre téléphone ou tablette
# pour obtenir les valeurs de service UUID et carateristic UUID
# Circuitpython 7.3.3 sur une metro M4 avec Airlift lite
# Modifier le 20/03/2023 par Pascal Mathieu version 1.1
```

import board

```
import time
import pwmio
from adafruit_motor import servo
from adafruit_ble import BLERadio
from adafruit_ble.advertising.standard import ProvideServicesAdvertisement
from adafruit_ble.services.nordic import UARTService

from adafruit_airlift.esp32 import ESP32
esp32 = ESP32() # DEFAULT

adapter = esp32.start_bluetooth()

ble = BLERadio(adapter)
uart = UARTService()
advertisement = ProvideServicesAdvertisement(uart)

MLI = pwmio.PWMOut(board.D3, duty_cycle=65635, frequency=50)
servo_1 = servo.Servo(MLI)
servo_1.angle = 0

def conversion_donnees(valeur):    # Terminée
    """ A partir d'une information transmise en Ble par appinventor
    soit one_byte = b'2,180\x00' la fonction renvoie deux valeurs
    liste_val[0] = 2 et liste_val[1] = 180
    """
    donnees = []
    #one_byte = b'2,180\x00'
    chaine = str(valeur)
    liste_donnees = chaine.split(",") # séparation de la chaîne [b', '2,180\x00', ""]
    donnees = str(liste_donnees[1]) # récupération de la chaîne du milieu
    val = donnees[:-4] # suppression des termes \x00
    liste_val = val.split(",")
    val_1 = liste_val[0]
    val_2 = liste_val[1]
    return val_1,val_2 # séparation en deux valeurs l'une pour le module grove et l'autre pour la commande

def commande_moteur():    # A faire
    pass

def commande_servo(angle): # Terminée
    servo_1.angle = int(angle)

def lecture_capteurs():    # A faire
    return "45,45,3,70"    # valeur provisoire pour test

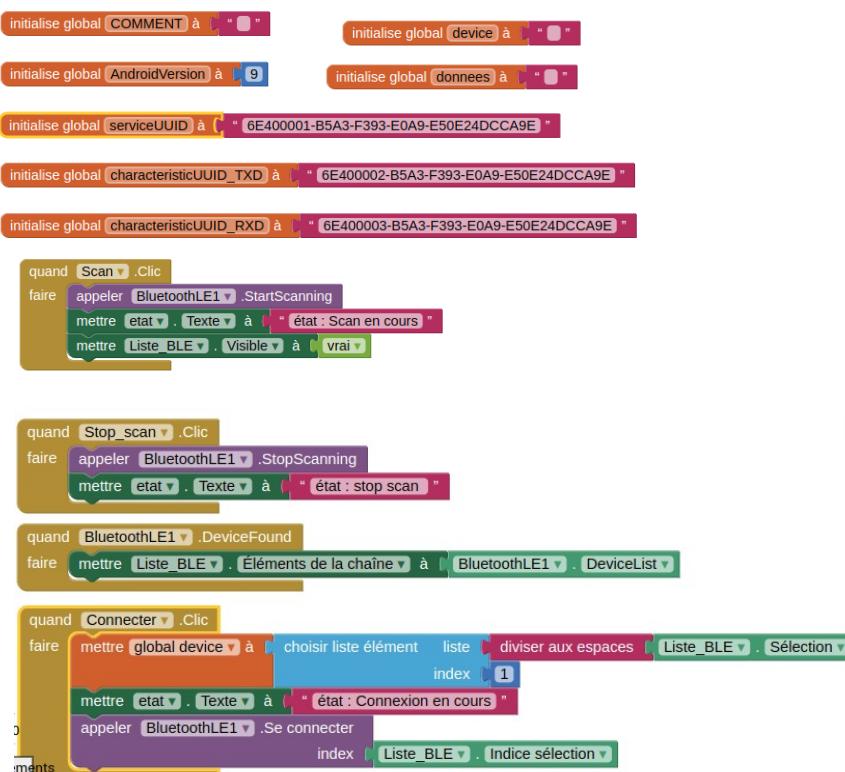
while True:
```

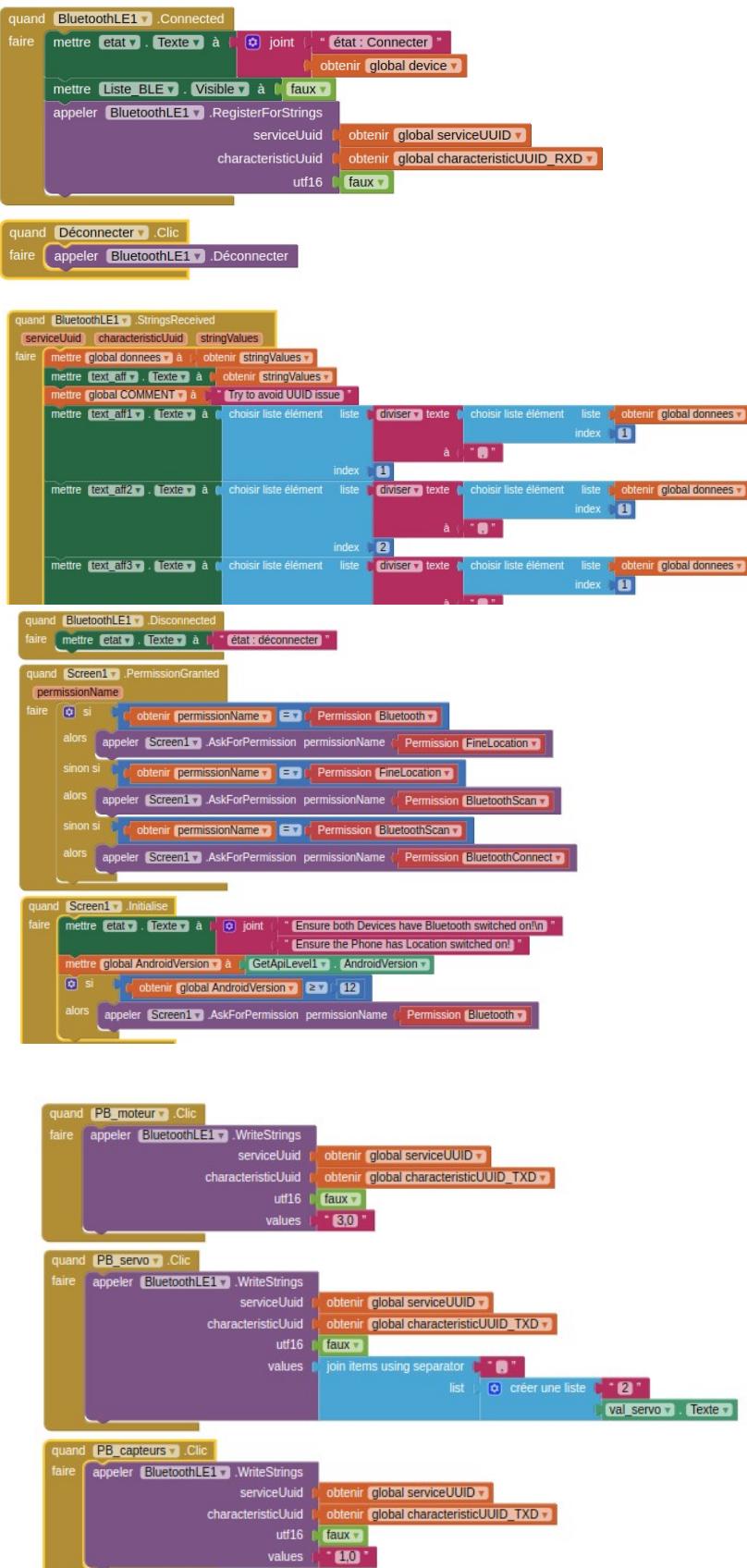
```

ble.start_advertising(advertisement)
print("Attente de connexion")
while not ble.connected:
    pass
print("La Connexion est active")
while ble.connected:
    # Returns b" if nothing was read.
    one_byte = uart.read()
    if one_byte:
        val_1,val_2 = conversion_donnees(one_byte)
        print(one_byte)
        print("val 1 : ",val_1," val 2 : ",val_2)
        if val_1 == '3':
            commande_moteur()
            print("commande moteur")
        if val_1 == '2':
            commande_servo(val_2)
            print("commande servo")
        if val_1 == '1' :
            donnees = lecture_capteurs()
            print("lecture des capteurs : ",donnees)
            uart.write(donnees)
            time.sleep(1)
            uart.write("12,198,56")
            time.sleep(1)

```

## Programme AppInventor





## Présentation graphique de l'application AppInventor

