

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.

La valeur 2 permet de sélectionner le module grove servo-moteur

La valeur saisie sur L'application permet de modifier l'angle Du servo-moteur

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 fait 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

The screenshot displays the App Inventor visual programming interface with the following code blocks:

- Initialisation:**
 - Initialise global `COMMENT` à `" "`
 - Initialise global `device` à `" "`
 - Initialise global `AndroidVersion` à `9`
 - Initialise global `donnees` à `" "`
 - Initialise global `serviceUUID` à `"6E400001-B5A3-F393-E0A9-E50E24DCCA9E"`
 - Initialise global `characteristicUUID_TXD` à `"6E400002-B5A3-F393-E0A9-E50E24DCCA9E"`
 - Initialise global `characteristicUUID_RXD` à `"6E400003-B5A3-F393-E0A9-E50E24DCCA9E"`
- Scan Click:**
 - Appeler `BluetoothLE1` . `StartScanning`
 - Mettre `etat` . `Texte` à `" état : Scan en cours "`
 - Mettre `Liste_BLE` . `Visible` à `vrai`
- Stop_scan Click:**
 - Appeler `BluetoothLE1` . `StopScanning`
 - Mettre `etat` . `Texte` à `" état : stop scan "`
- BluetoothLE1 DeviceFound:**
 - Mettre `Liste_BLE` . `Éléments de la chaîne` à `BluetoothLE1` . `DeviceList`
- Connecter Click:**
 - Mettre `global device` à `choisir liste élément` liste `Liste_BLE` . `Sélection`
 - index `1`
 - Mettre `etat` . `Texte` à `" état : Connexion en cours "`
 - Appeler `BluetoothLE1` . `Se connecter`
 - index `Liste_BLE` . `Indice sélection`

```

quand BluetoothLE1 .Connected
faire
    mettre etat . Texte à joint " état : connecter "
    obtenir global device
    mettre Liste_BLE . Visible à faux
    appeler BluetoothLE1 .RegisterForStrings
        serviceUuid obtenir global serviceUUID
        characteristicUuid obtenir global characteristicUUID_RXD
        utf16 faux

quand Déconnecter .Clic
faire
    appeler BluetoothLE1 .Déconnecter

quand BluetoothLE1 .StringsReceived
serviceUuid characteristicUuid stringValues
faire
    mettre global donnees à obtenir stringValues
    mettre text_aff . Texte à obtenir stringValues
    mettre global COMMENT à " Try to avoid UUID issue "
    mettre text_aff1 . Texte à choisir liste élément liste diviser texte choisir liste élément liste obtenir global donnees
    index 1
    mettre text_aff2 . Texte à choisir liste élément liste diviser texte choisir liste élément liste obtenir global donnees
    index 2
    mettre text_aff3 . Texte à choisir liste élément liste diviser texte choisir liste élément liste obtenir global donnees
    index 3

quand BluetoothLE1 .Disconnected
faire
    mettre etat . Texte à " état : déconnecter "

quand Screen1 .PermissionGranted
permissionName
faire
    si obtenir permissionName == Permission Bluetooth
    alors appeler Screen1 .AskForPermission permissionName Permission FineLocation
    sinon si obtenir permissionName == Permission FineLocation
    alors appeler Screen1 .AskForPermission permissionName Permission BluetoothScan
    sinon si obtenir permissionName == Permission BluetoothScan
    alors appeler Screen1 .AskForPermission permissionName Permission BluetoothConnect

quand Screen1 .Initialise
faire
    mettre etat . Texte à joint " Ensure both Devices have Bluetooth switched on!\n "
    " Ensure the Phone has Location switched on! "
    mettre global AndroidVersion à GetApiLevel1 . AndroidVersion
    si obtenir global AndroidVersion >= 12
    alors appeler Screen1 .AskForPermission permissionName Permission Bluetooth

quand PB_moteur .Clic
faire
    appeler BluetoothLE1 .WriteStrings
        serviceUuid obtenir global serviceUUID
        characteristicUuid obtenir global characteristicUUID_TXD
        utf16 faux
        values " 3.0 "

quand PB_servo .Clic
faire
    appeler BluetoothLE1 .WriteStrings
        serviceUuid obtenir global serviceUUID
        characteristicUuid obtenir global characteristicUUID_TXD
        utf16 faux
        values join items using separator " " list créer une liste 2 val_servo . Texte

quand PB_capteurs .Clic
faire
    appeler BluetoothLE1 .WriteStrings
        serviceUuid obtenir global serviceUUID
        characteristicUuid obtenir global characteristicUUID_TXD
        utf16 faux
        values " 1.0 "
    
```

Présentation graphique de l'application AppInventor

