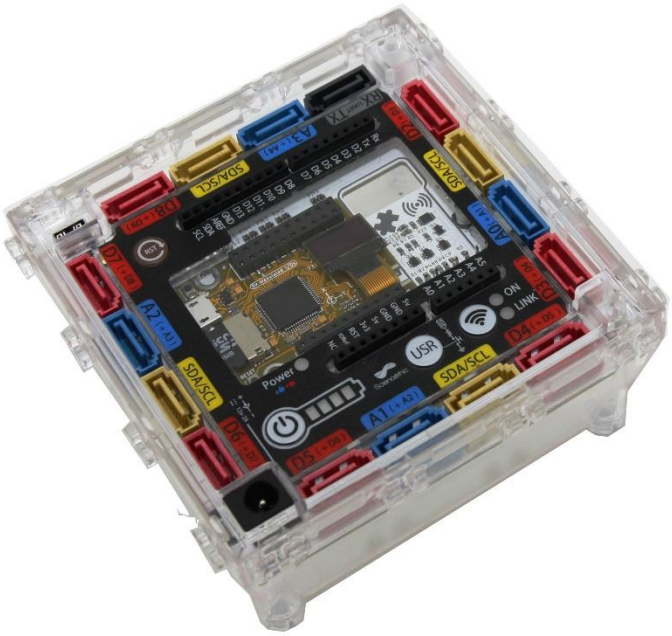
# Guide d'utilisation de Plug'Uino® Py sous Windows



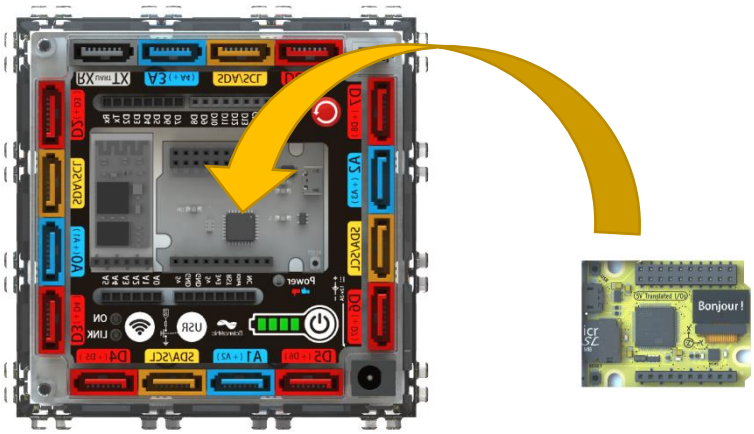

Sciencéthic

NOVEMBRE 2019

# Sommaire

# Présentation de Plug'Uino® Py

L'interface Plug'Uino® Py est composée du boitier de connexion SATA et d'une carte amovible Plug'Py.



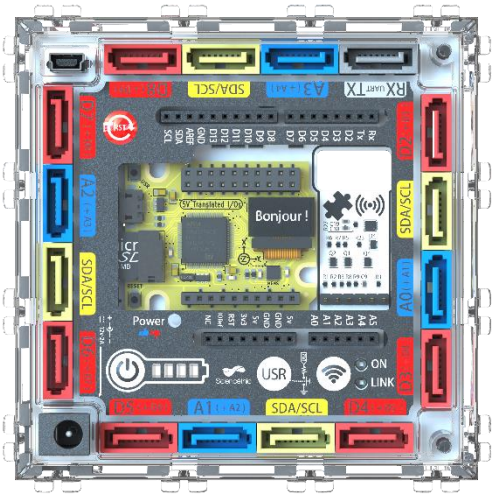

*Plug'Uino®* *Plug'Py*

**Plug'Uino® Py**

# Description du matériel

## 1. Le boîtier de connexion SATA

Le boîtier de connexion SATA est un boîtier transparent incassable en polycarbonate, il peut contenir :
- soit la carte de microcontrôleur Plug'Uno (aux fonctions identiques à Arduino Uno Rev.3)
- soit la carte de microcontrôleur Plug'Py fonctionnant sous MicroPython.
Ces deux dernières cartes sont amovibles et peuvent être interchangées.




**Plug'Uino® Py**

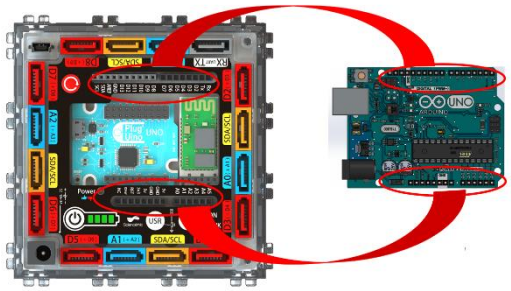

**Plug'Uino® Uno**

(100% compatible ArduinoTM Uno Rev3)

Ce boîtier donne accès aux entrées sorties E/S, de la carte microcontrôleur déjà installée.
Ces E/S sont accessibles par deux types de connecteurs : SIL et SATA

### Connecteurs SIL

Les connecteurs SIL donnent accès aux broches des entrées / sorties des cartes microcontrôleur Plug'Py ou Plug'Uno.
L'implémentation des broches des connecteurs SIL est exactement comme sur une carte Arduino™Uno

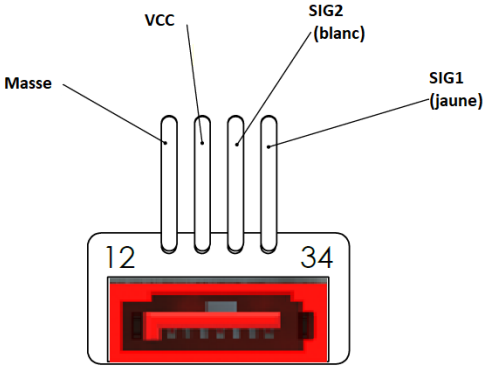


*Implantation exactement identique des signaux entre Plug'Uino® Uno ou Plug'Uino® Py et Arduino™ Uno*

## Connecteurs SATA

Les connecteurs SATA ont un code de couleur :


Rouge : numérique
Bleu :analogique
Jaune : bus de données $I^2C$
Gris : UART



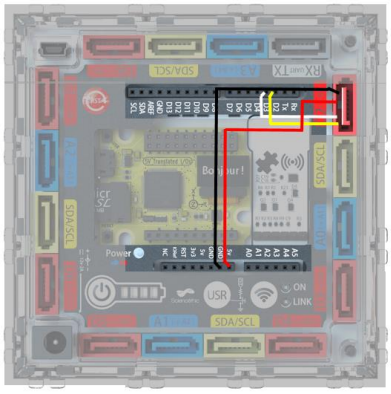

Ces connecteurs permettent de relier en 1 seul « clic » au microcontrôleur les différents fils (Masse (GND), 5 V (Vcc) et Signaux (SIG1 et SIG2) nécessaires aux fonctionnement des capteurs et actionneurs. Ils simplifient et fiabilisent la connectique.

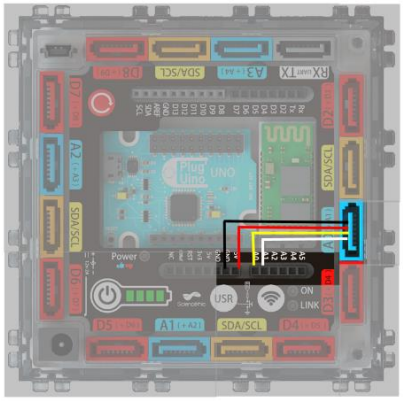
Certains capteurs ou actionneurs peuvent utiliser SIG1 et SIG2 ce qui explique la notation ci-dessous sur les connecteurs :
Le connecteur D2 (+D3) :

**D2** (+D3)
D2 est relié à SIG1
D3 est relié à SIG2

D3 ne sera utilisé que si le module connecté nécessite deux E/S digitales

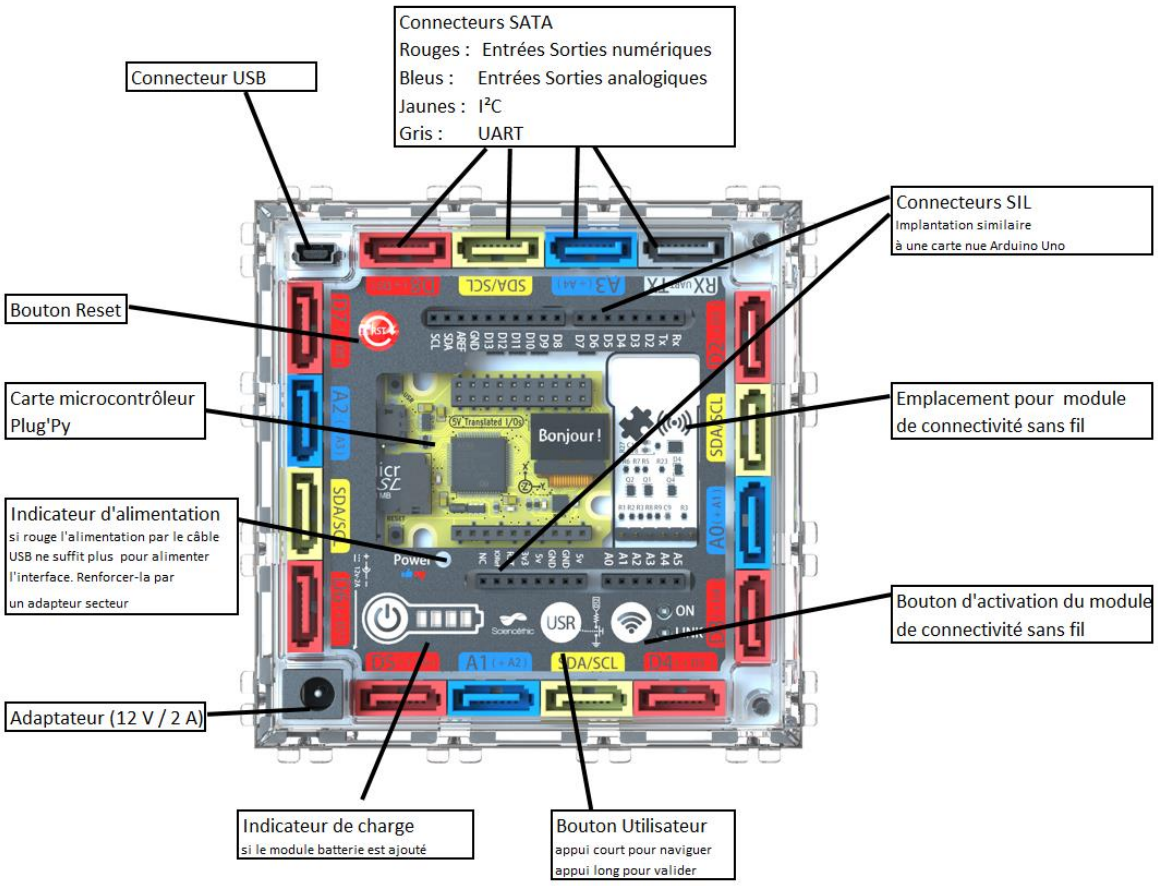
Les signaux accessibles par un connecteur SATA sont également accessibles sur les connecteurs SIL comme le montrent les deux illustrations suivantes :





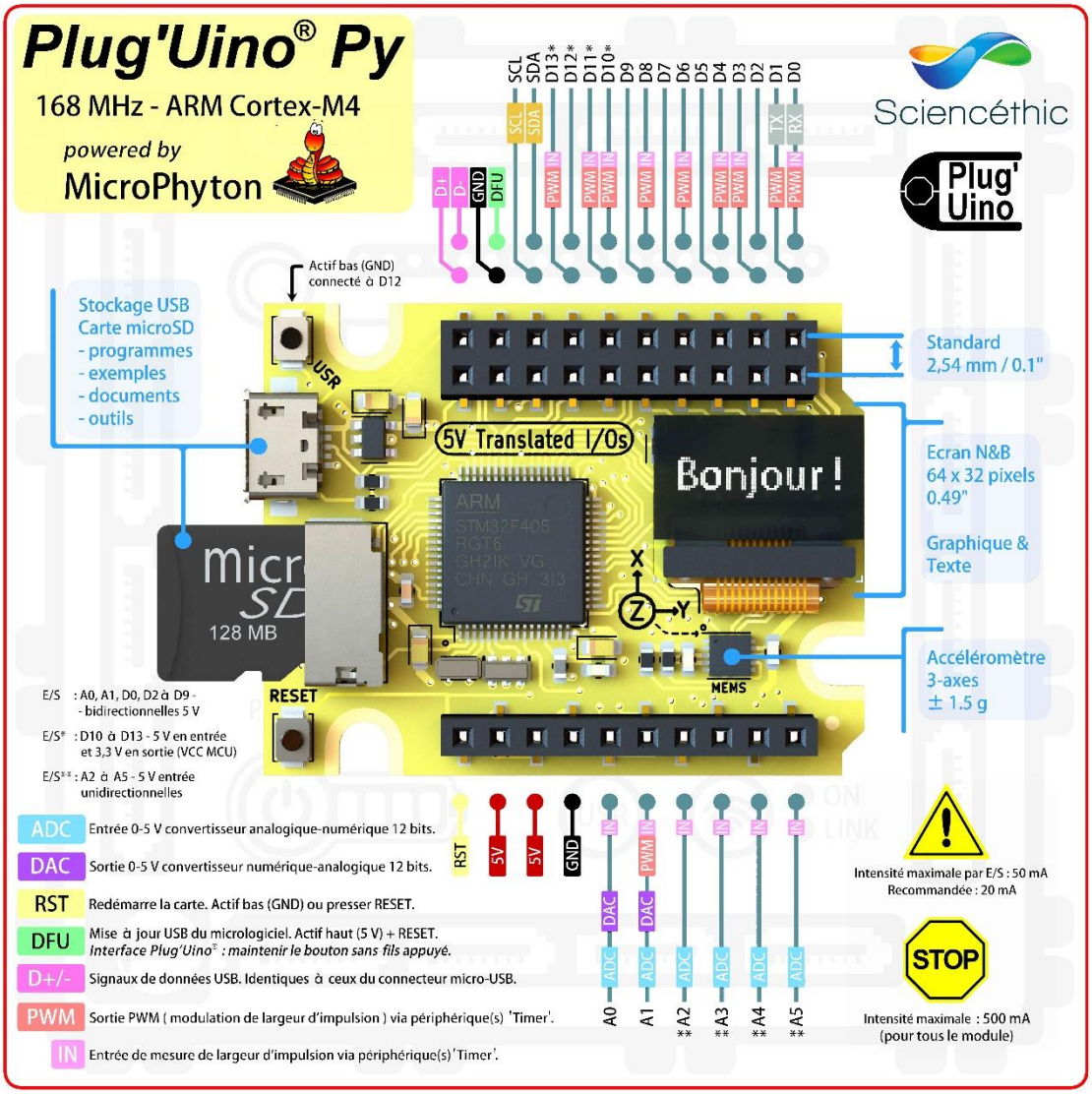
Enfin le boîtier de connexion SATA peut accepter deux modules supplémentaires :
- Un module de connectivité sans fil comme par exemple Bluetooth
- Un module batterie

Plug'Uino® Py peut devenir ainsi un objet communicant et autonome par rapport à l'énergie et par rapport à l'ordinateur, tablette ….



## 2. La carte Plug'Py

La carte Plug'Py fonctionne sous MicoPython, elle contient entre autres un lecteur de carte SD, un afficheur 64x32 pixels, un accéléromètre…
Elle est directement programmable en langage Python.



# Utilisation de Plug'Uino® Py

L'interface Plug'Uino® Py offre trois modes de fonctionnement :

Mode Interface – Mode Embarqué – Mode REPL

Le tableau ci-dessous illustre ces modes de fonctionnement et leur relation à l'ordinateur

| Plug'Uino® Py | Où le script s'écrit? | Où le script se lance? | Où le script s'exécute? | Où s'affichent les résultats? |
|---|---|---|---|---|
| Mode objet embarqué |  |  |  |  et/ou  |
| Mode Terminal REPL |  |  et/ou  |  |  et/ou  |
| Mode Interface |  |  |  et  |  et/ou  |

Le mode interface est le mode indiqué pour l'enseignement de la programmation en travaux pratiques de Physique-Chimie et de SVT.

Le mode objet embarqué est le mode pour créer des objets technologiques.

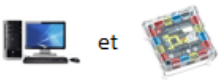
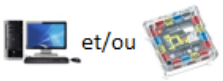
MicroPython fonctionne comme un petit système d'exploitation qui exécute des programmes utilisateur et fournit un interpréteur de commandes (REPL).

Ce que nous appelons ‘’Mode Terminal REPL’’, c'est l'utilisation du mode REPL de MicroPython dans une fenêtre-terminal : port série (cf. page 19) ou page web (cf. page 23).

# Mode Interface

## Mode Interface :

Ce mode consiste à exécuter un script à partir de l'ordinateur qui enverra à l'interface des instructions Python et recevra en retour ses réponses.
Le programme sur le PC se charge alors de traiter les données, comme par exemple faire un tracé de courbes.
Ce mode nécessite d'établir une connexion entre l'interface et l'ordinateur. Cette connexion se fait à l'aide d'un port USB configuré comme un port série COMxx (xx étant le numéro alloué par l'ordinateur à l'interface)

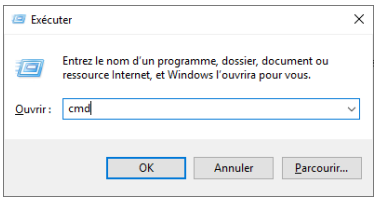
| Plug'Uino® Py | Où le script s'écrit? | Où le script se lance? | Où le script s'exécute? | Où s'affichent les résultats? |
|---|---|---|---|---|
| Mode Interface |  |  |  |  |

Pour le bon fonctionnement de ce mode, il est nécessaire de vérifier que :

- la librairie **PySerial** est installée dans votre ordinateur.
- la présence de la librairie libPlugPy.py (cf. point 2-)

1- Installation de la librairie **PySerial**

PySerial est une librairie du langage Python qui permet d'accéder au port série.
Pour l'installer :
Taper simultanément sur les touches Windows+R
taper cmd et valider




Ensuite taper :
pip install pyserial
Taper sur entrée
Le message :''Requirement already satisfied in c:\....
indique la bonne installation de la librairie.
Cette installation n'est nécessaire qu'une seule fois.

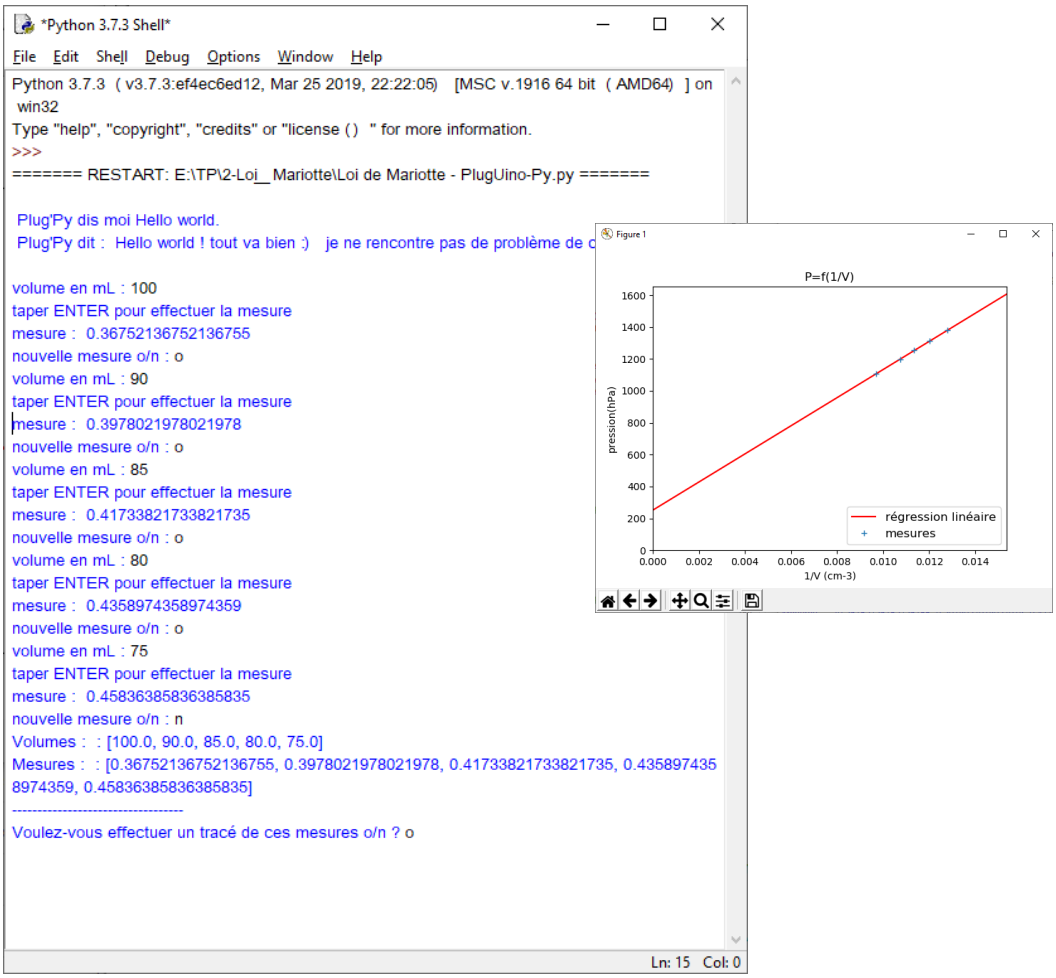
2- La librairie **libPlugPy.py** doit se trouver dans le même répertoire que le script à exécuter.
La librairie libPlugPy.py se charge de traduire et de :
- connecter de façon logicielle l'interface à vos scripts Python
- mettre en forme et transmettre les requêtes qui lui sont faites
- lire les résultats d'exécution et les retourner au script de l'utilisateur

**Exemple d'utilisation du mode interface**
exemple Mariotte :
ouvrir un éditeur Python, IDLE par exemple.
Ouvrir le script ‘’**Loi de Mariotte .py’’** qui se trouve sur la carte SD
E:\TP\2-Loi_Mariotte

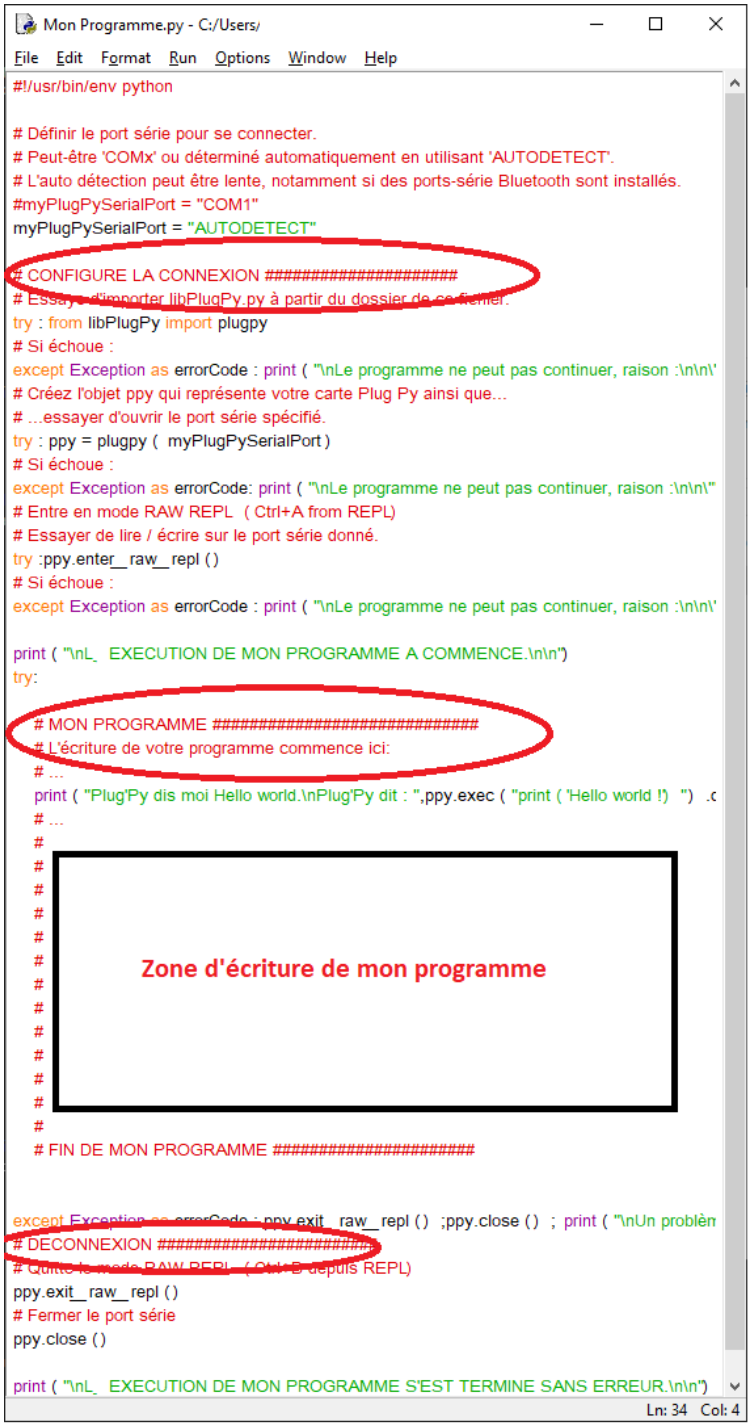
L'écriture de ce script est structurée en 4 sections :

1- Configuration de la connexion
2- Mesures
3- Fermeture du port
4- Tracé du graphique

Sélectionner le Menu Run,
sélectionner Python Shell : cela ouvre une fenêtre Python Shell
ensuite valider ‘’Run Module’’




Noter que pour écrire un programme, il est conseillé d’utiliser une trame d’écriture que l’on trouve dans le script ‘’Mon Programme.py’’
Il se trouve dans E:\Misc&Tools\Remote Control
(E étant l’unité dans laquelle se trouve la carte SD de Plug’Uino® Py)
Il est structuré en trois sections :
1- configuration de la connexion et importation des librairies
2- La zone d’écriture du programme
3- La déconnexion du port COM utilisé


Mon Programme.py - C:/Users/
File Edit Format Run Options Window Help
#!/usr/bin/env python
# Définir le port série pour se connecter.
# Peut-être 'COMx' ou déterminé automatiquement en utilisant 'AUTODETECT'.
# L'auto détection peut être lente, notamment si des ports-série Bluetooth sont installés.
#myPlugPySerialPort = "COM1"
myPlugPySerialPort = "AUTODETECT"
# CONFIGURE LA CONNEXION ####################
# Essaye d'importer libPlugPy.py à partir du dossier de ce fichier.
try : from libPlugPy import plugpy
# Si échoue :
except Exception as errorCode : print ( "\nLe programme ne peut pas continuer, raison :\n\n\'
# Créez l'objet ppy qui représente votre carte Plug Py ainsi que...
# ...essayer d'ouvrir le port série spécifié.
try : ppy = plugpy ( myPlugPySerialPort )
# Si échoue :
except Exception as errorCode: print ( "\nLe programme ne peut pas continuer, raison :\n\n\"
# Entre en mode RAW REPL ( Ctrl+A from REPL)
# Essayer de lire / écrire sur le port série donné.
try :ppy.enter_raw_repl ()
# Si échoue :
except Exception as errorCode : print ( "\nLe programme ne peut pas continuer, raison :\n\n\'
print ( "\nL_ EXECUTION DE MON PROGRAMME A COMMENCE.\n\n")
try:
# MON PROGRAMME ############################
# L'écriture de votre programme commence ici:
# ...
print ( "Plug'Py dis moi Hello world.\nPlug'Py dit : ",ppy.exec ( "print ( 'Hello world !') ") .c
# ...
#
#
#
#
#
#
#
#
#
#
#
#
#
#
Zone d'écriture de mon programme
# FIN DE MON PROGRAMME ####################
except Exception as errorCode : ppy.exit_raw_repl () ;ppy.close () ; print ( "\nUn problèn
# DECONNEXION ####################
# Quitte le mode RAW REPL ( Ctrl+B depuis REPL)
ppy.exit_raw_repl ()
# Fermer le port série
ppy.close ()
print ( "\nL_ EXECUTION DE MON PROGRAMME S'EST TERMINE SANS ERREUR.\n\n")
Ln: 34 Col: 4

# Mode Objet embarqué

## Mode ‘Objet embarqué’ :

Grâce aux périphériques embarqués, carte SD, bouton utilisateur USR et à afficheur, Plug’Uino® Py ne nécessite pas d’ordinateur pour exécuter des applications.

| Plug'Uino® Py | Où le script s'écrit? | Où le script se lance? | Où le script s'exécute? | Où s'affichent les résultats? |
|---|---|---|---|---|
| Mode objet embarqué |  |  |  |  et/ou  |

Vous trouverez de multiples programmes sur la carte SD dans le répertoire \Samples\Gui

Pour y accéder:

Appui simultané sur les boutons RST + USR
Le menu ‘’Boot Options’’ apparaît sur l’écran.

Pour défiler le menu : appui court sur le bouton USR,
Pour sélectionner : Appui long sur le bouton USR,
Sélectionner : ‘’Startup file’’ (appui long sur USR),
Sélectionner ‘’Samples’’,
Sélectionner ‘’GUI’’,

Dans ce répertoire, vous trouverez 4 scripts installés:

- AccelGraph : Trace le graphique de la réponse de l’accéléromètre intégré en fonction du temps

- AdcBarGraph : Un graphique à barre représentant le niveau de tension analogique de l’entrée A0

- micro TRex : un jeu ☺

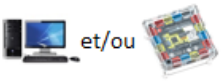
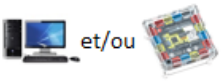
- SigGenerator : Génère un signal carré, triangle ou sinusoïdal qui sera disponible sur D0

# Mode Terminal REPL

**R** Read
**E** Evaluate
**P** Print
**L** Loop

## Mode Terminal REPL:

REPL (Read-Eval-Print-Loop), c'est le mode interpréteur interactif de MicroPython qui peut exécuter des lignes d'un programme instruction après instruction et de façon interactive.
Pour ce mode, il est nécessaire d'établir une liaison entre l'ordinateur et la carte microcontrôleur.
Cette liaison peut se faire :
- soit par un port série
- soit par la connectivité offerte par WebUSB qui permet de connecter Plug'Py (la carte microcontrôleur de Plug'Uino® Py) à une page Web.

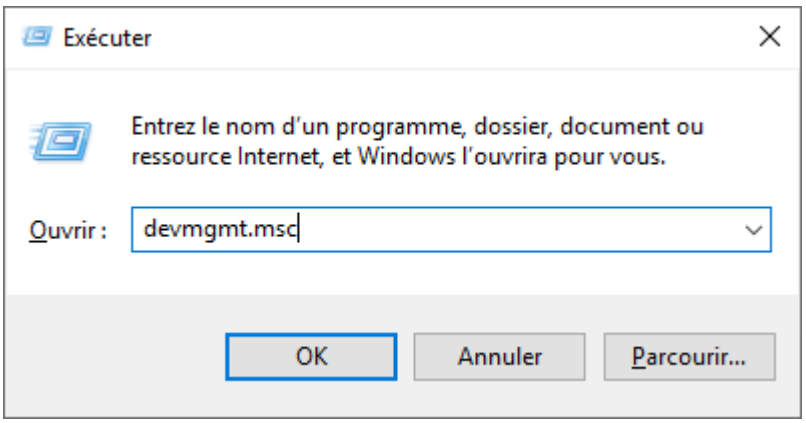
| Plug'Uino® Py | Où le script s'écrit? | Où le script se lance? | Où le script s'exécute? | Où s'affichent les résultats? |
|---|---|---|---|---|
| Mode Fenêtre REPL |  |   |  |   |

### Mode Terminal REPL par port Série

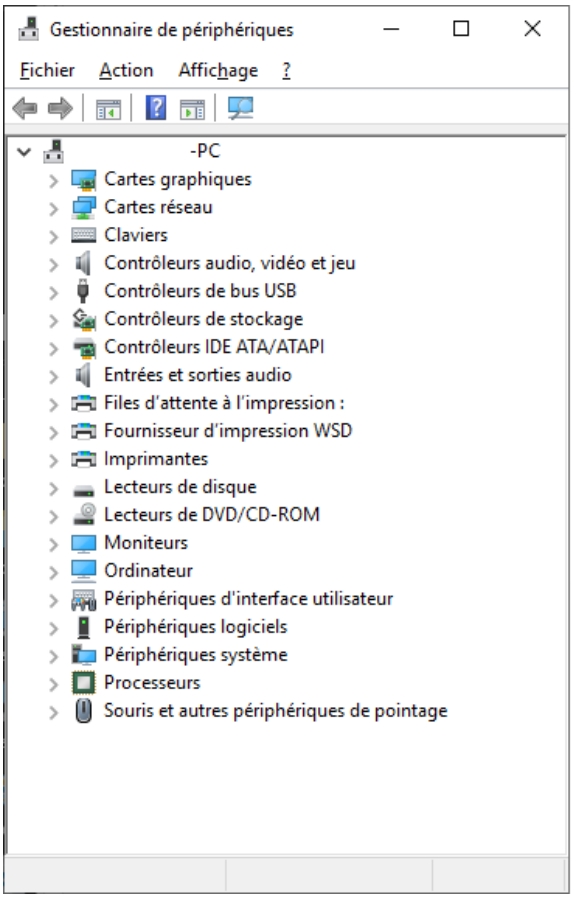
Grâce à un programme Terminal de port-série comme PuTTY (téléchargeable sur PuTTy.org) on peut saisir des instructions à exécuter par Plug'Uino® Py et au retour lire les résultats

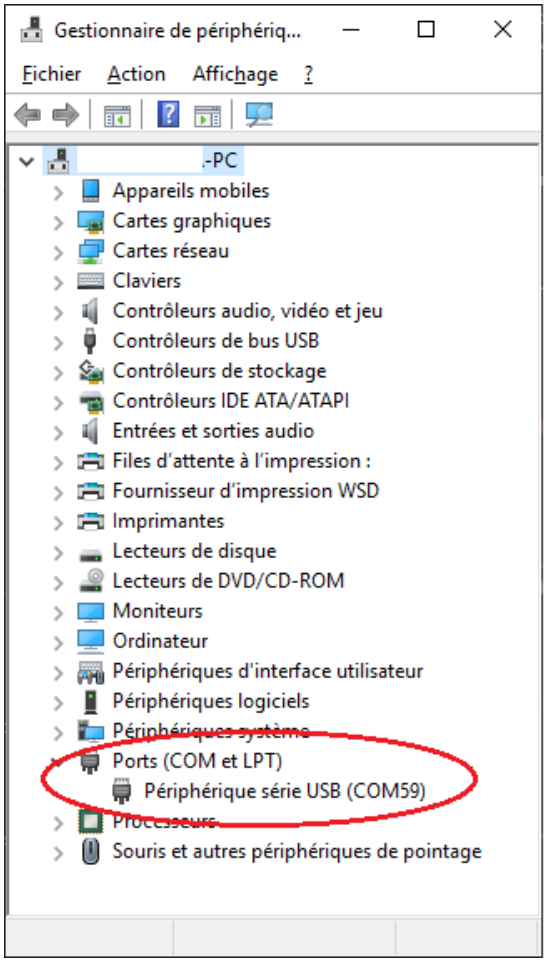
Mais préalablement, il est nécessaire de connaître le numéro du port COM qui reconnaît l'interface Plug'Uino® Py.

Sous Windows 10 :
Touche Windows + R
Ouvrir ‘’devmgmt.msc’’



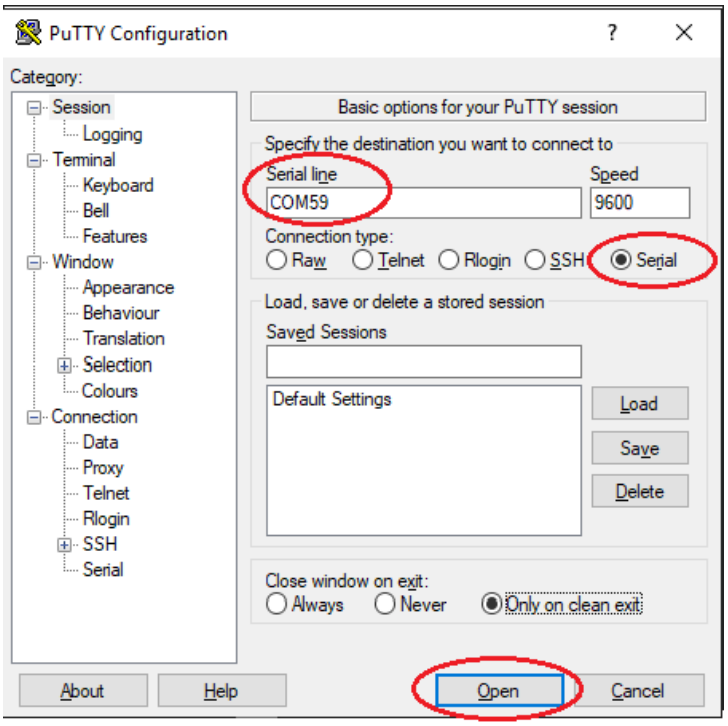

Valider, l'écran suivant est affiché :



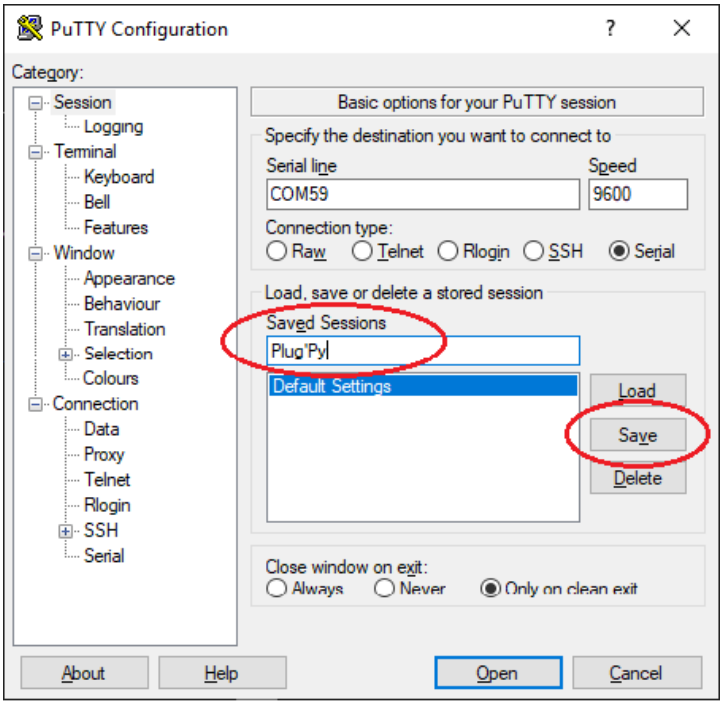




En branchant l'interface, une ligne supplémentaire sera affichée Ports (COM et LPT) et vous indique le COM utilisé, dans notre illustration c'est le COM59.

Revenons sur PuTTy qui établit la connexion entre le PC et l'interface, après le téléchargement à l'adresse PuTTy.org, on exécute l'application et on renseigne comme suit:

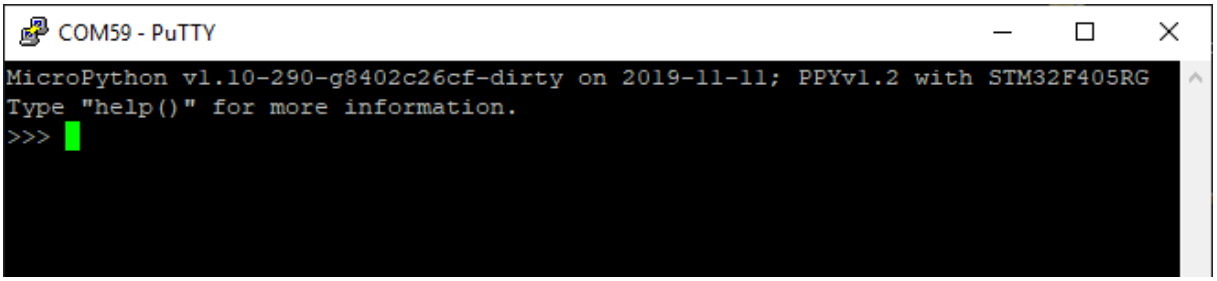



Il est à noter que l'on peut sauvegarder la session, par exemple

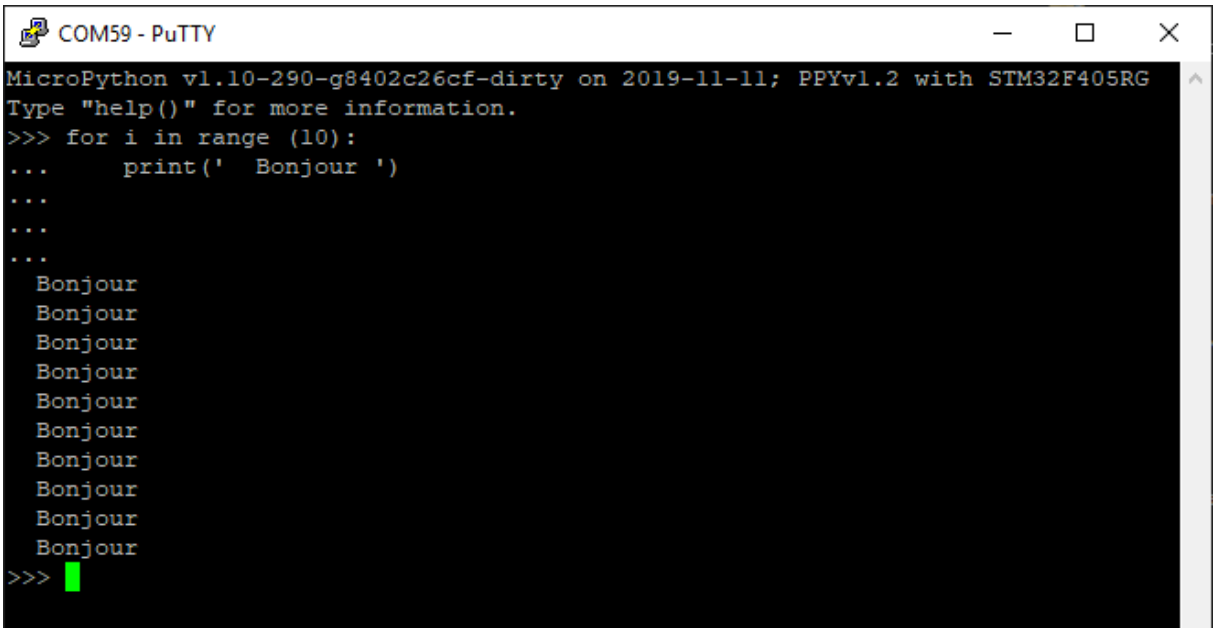



Pour les connexions suivantes, il suffit de sélectionner la session Plug'Py et valider par Open.

Une fenêtre du terminal série s'ouvre pour le mode REPL:



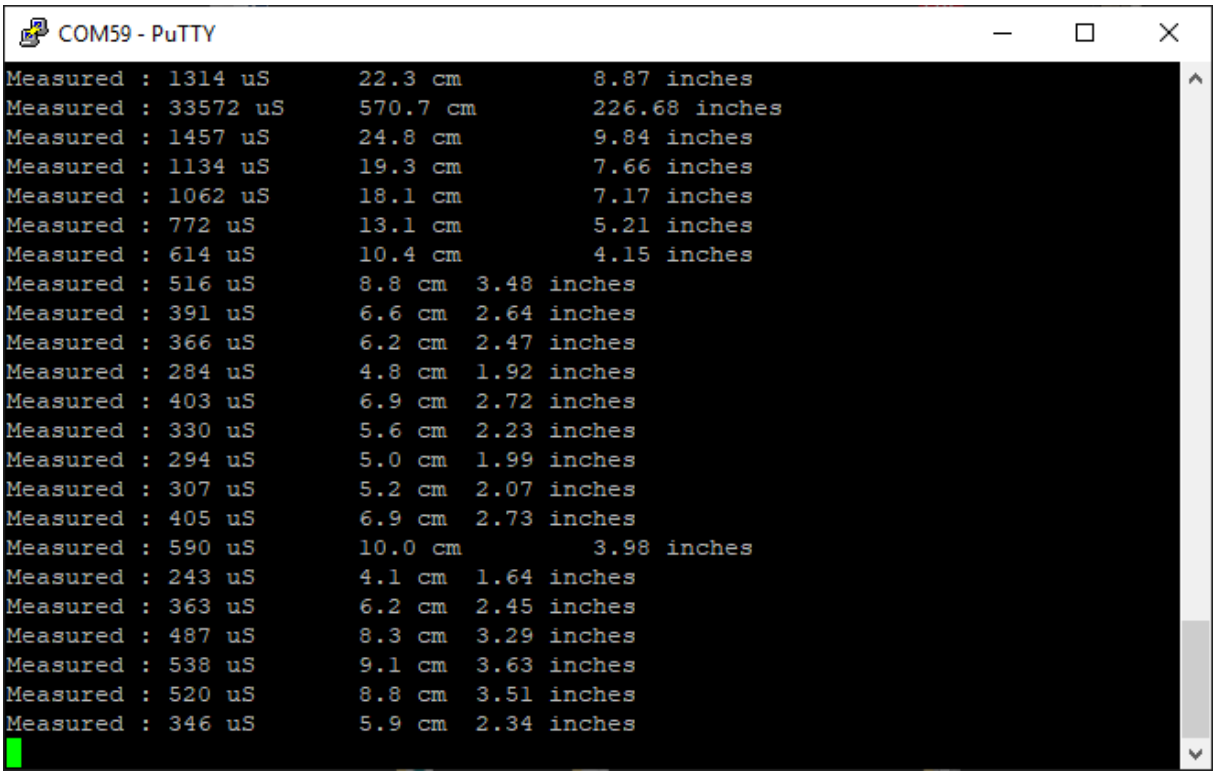

On peut demander à la carte microntrôleur MicroPython de traiter les instructions suivantes : écrire Bonjour 10 fois. Le résultat est renvoyé à la fenêtre du terminal REPL.




On peut aussi commander l'exécution d'un script qui est sur la carte SD et de voir les résultats retournés dans la fenêtre du terminal REPL.

Exemple : excéutons le script USranger.py, ce script se trouve dans le répertoire Samples\USRanger pour y accéder :
Appui simultané sur les boutons RST + USR
Sélectionner Startup file (rappel appui long sélectionner appui court pour défiler)
Sélectionner Samples (Appuis courts successifs pour atteindre ‘’Samples’’ ensuite appui long pour l'ouvrir)
Sélectionner \USRanger et exécuter USRanger

Les résultats seront affichés




Il est à noter que l'on peut commander un script se trouvant dans la carte SD à partir de la fenêtre du terminal série.

Par exemple, exécuter successivement le script OledBitmap ensuite le script OledShapes, pour cela :

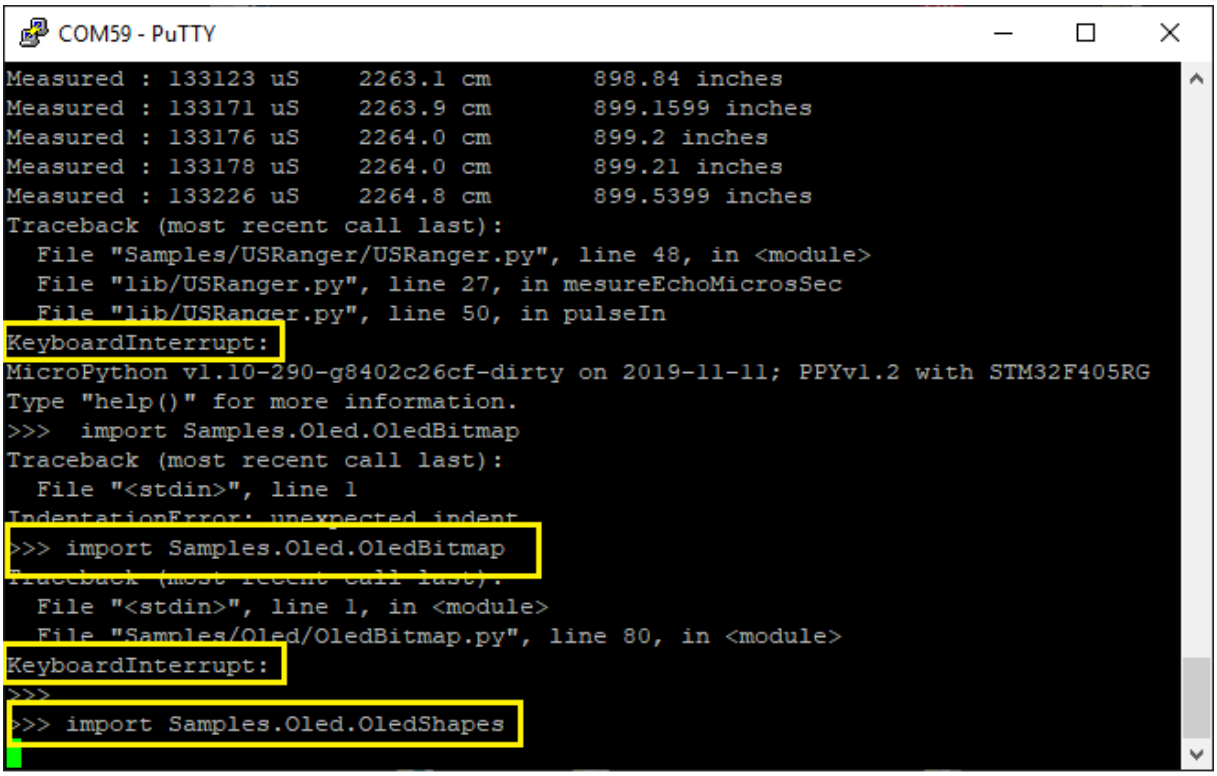
CTRL+C interrompt l'exécution du script en cours
ensuite on écrit :

```
import Samples.Oled.OledBitmap
```

on observe le résultat sur l'écran Oled de l'interface

CTRL+C interrompt l'exécution du script OledBitmap

```
import Samples.OledShapes
```




On observe l'écran Oled de l'interface.

Il est à noter que les commandes spéciales REPL sont :

CTRL+C interompt l'exécution du script en cours

CTRL+D Redémare le système (pour importer une ressource plus qu'une fois)

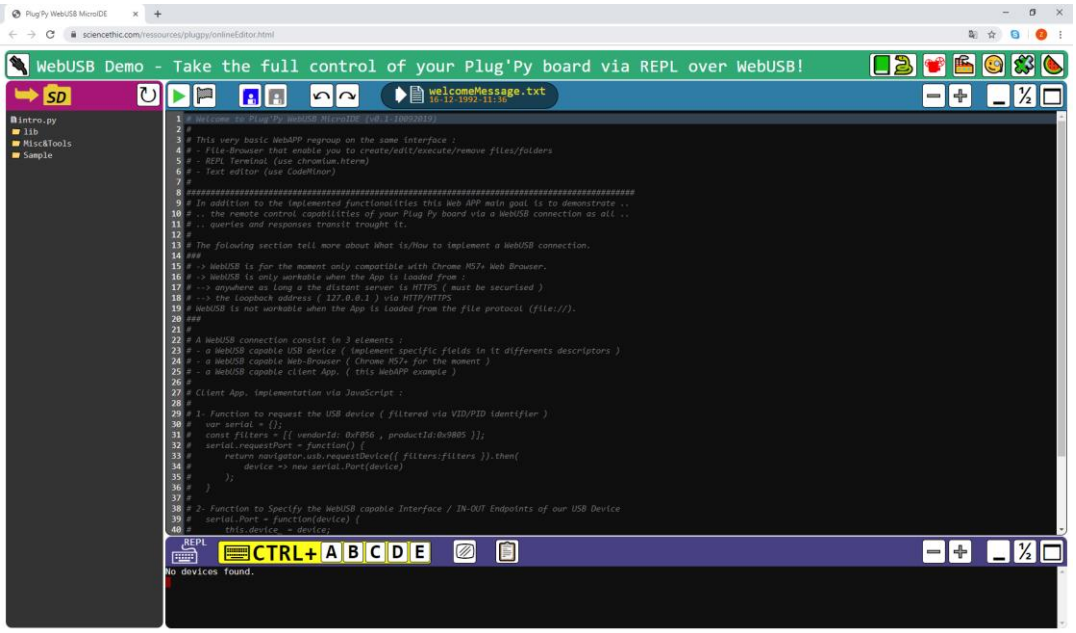
CTRL+E Mode d'écriture du presse-papier (Coller)

CTRL+A Mode REPL brut

CTRL+B Quitte le mode REPL brut

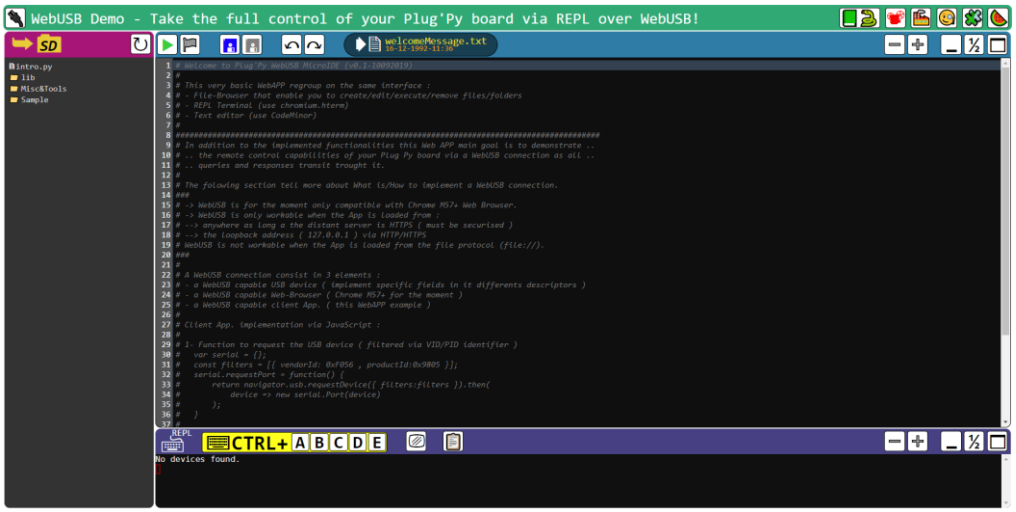
**Mode Terminal REPL par WebUSB**

L'application Web "Python Editor" utilise la connectivité WebUSB qui permet de connecter Plug'Py à une page Web.
L'application Web "Python Editor" offre une interface de type terminal qui asservit l'interpréteur REPL de Plug'Py.




Cette application peut être chargée depuis sciencethic.com.

Elle peut également être téléchargeable et exécutable localement (documentation de mise en œuvre est fournie dans le téléchargement)

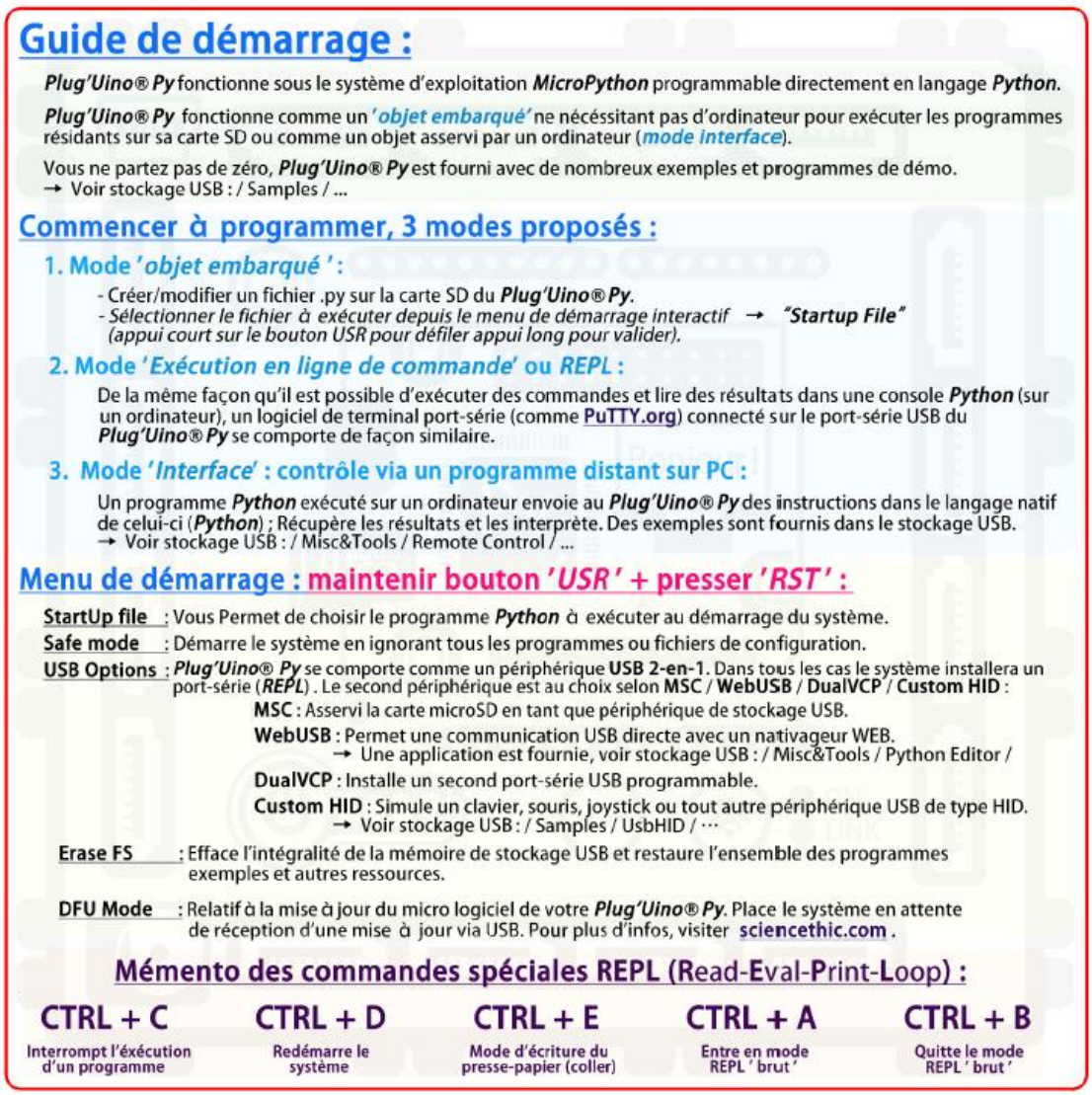


# Annexes

- Résumé du guide démarrage
- Le contenu de la carte SD
- Les modes USB de Plug'Uino® Py
- Traitement de quelques messages d'erreur

## 1- Résumé du guide de démarrage

Le document ci-dessous, résume tout ce qui est nécessaire à savoir pour le démarrage de la carte Plug'Py



# Guide de démarrage :

***Plug'Uino® Py*** fonctionne sous le système d'exploitation ***MicroPython*** programmable directement en langage ***Python***.

***Plug'Uino® Py*** fonctionne comme un *'objet embarqué'* ne nécéssitant pas d'ordinateur pour exécuter les programmes résidants sur sa carte SD ou comme un objet asservi par un ordinateur (*mode interface*).

Vous ne partez pas de zéro, ***Plug'Uino® Py*** est fourni avec de nombreux exemples et programmes de démo.
→ Voir stockage USB : / Samples / ...

## Commencer à programmer, 3 modes proposés :

### 1. Mode *'objet embarqué'* :

- Créer/modifier un fichier .py sur la carte SD du ***Plug'Uino® Py***.
- *Sélectionner le fichier à exécuter depuis le menu de démarrage interactif → "Startup File"*
  *(appui court sur le bouton USR pour défiler appui long pour valider).*

### 2. Mode *'Exécution en ligne de commande'* ou *REPL* :

De la même façon qu'il est possible d'exécuter des commandes et lire des résultats dans une console ***Python*** (sur un ordinateur), un logiciel de terminal port-série (comme PuTTY.org) connecté sur le port-série USB du ***Plug'Uino® Py*** se comporte de façon similaire.

### 3. Mode *'Interface'* : contrôle via un programme distant sur PC :

Un programme ***Python*** exécuté sur un ordinateur envoie au ***Plug'Uino® Py*** des instructions dans le langage natif de celui-ci (***Python***) ; Récupère les résultats et les interprète. Des exemples sont fournis dans le stockage USB.
→ Voir stockage USB : / Misc&Tools / Remote Control / ...

## Menu de démarrage : maintenir bouton *'USR'* + presser *'RST'* :

**StartUp file** : Vous Permet de choisir le programme ***Python*** à exécuter au démarrage du système.

**Safe mode** : Démarre le système en ignorant tous les programmes ou fichiers de configuration.

**USB Options** : ***Plug'Uino® Py*** se comporte comme un périphérique **USB 2-en-1**. Dans tous les cas le système installera un port-série (***REPL***) . Le second périphérique est au choix selon **MSC** / **WebUSB** / **DualVCP** / **Custom HID** :

- **MSC** : Asservi la carte microSD en tant que périphérique de stockage USB.
- **WebUSB** : Permet une communication USB directe avec un nativageur WEB.
  → Une application est fournie, voir stockage USB : / Misc&Tools / Python Editor /
- **DualVCP** : Installe un second port-série USB programmable.
- **Custom HID** : Simule un clavier, souris, joystick ou tout autre périphérique USB de type HID.
  → Voir stockage USB : / Samples / UsbHID / ···

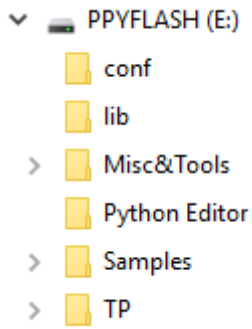
**Erase FS** : Efface l'intégralité de la mémoire de stockage USB et restaure l'ensemble des programmes exemples et autres ressources.

**DFU Mode** : Relatif à la mise à jour du micro logiciel de votre ***Plug'Uino® Py***. Place le système en attente de réception d'une mise à jour via USB. Pour plus d'infos, visiter sciencethic.com .

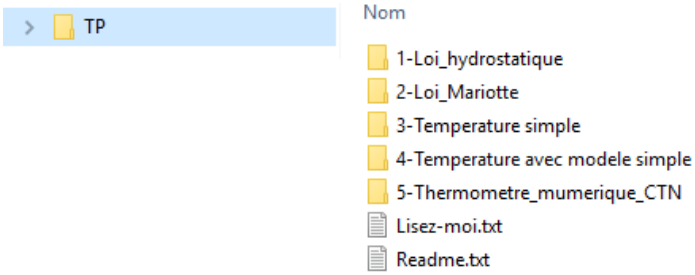
## Mémento des commandes spéciales REPL (Read-Eval-Print-Loop) :

| CTRL + C | CTRL + D | CTRL + E | CTRL + A | CTRL + B |
|---|---|---|---|---|
| Interrompt l'éxécution d'un programme | Redémarre le système | Mode d'écriture du presse-papier (coller) | Entre en mode REPL ' brut ' | Quitte le mode REPL ' brut ' |

## 2- Le contenu de la carte SD



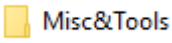

Le dossier appelé TP contient 5 exemples de TP.
Ces TP utilisent l'interface en ''mode Interface''





Le dossier Conf contient les fichiers nécessaires à la configuration


Contient les librairies pour le LCD et l'émetteur récepteur Ultra son




Contient des outils, il a 3 sous dossiers

- **Python Editor** (WebUSB) Contient le lien pour ouvrir l'application WebUSB

- **USB Drivers** contient les pilotes pour le port Série VCP et pour le port utilisé par WebUSB

- **Remote Control** contient 3 programmes montrant la prise de contrôle à distance de la carte microcontrôleur
ils sont nommés pc_example_adc_1, 2 ou 3.
Dans ce dossier, vous trouverez également le dossier Template dans lequel ''Mon Programme.py'' est donné comme modèle d'écriture des programmes.


Samples

Ce dossier est riche en applications (échantillons) qui vous permettront de découvrir la totalité des ressources électroniques de la carte Plug'Py

Accel

Contient 3 exemples pour programmer l'accéléromètre intégré

Adc

Contient 3 exemples pour programmer l'ADC (CAN)

Dac

Contient 3 exemples pour programmer le DAC (CNA)

| | |
|---|---|
| File | Contient un exemple d'utilisation d'un fichier externe au programme utilisable par pour Créer/lire/ écrire/ ajouter des données |
| Gui | Contient 4 programmes utilisant l'interface graphique de la carte Plug'Py, vous disposez notamment d'un générateur de signal |
| I2C | Contient un programme qui permet de lire ou d'écrire sur le bus I²C |
| LCD16x2RGB | Contient un programme permettant d'utiliser le LCD |
| Oled | Contient 4 programmes exeples d'utilisation l'écran Oled. Décrouvrez notamment OledShapes.py ! |
| Pin | Contient 2 programmes qui permettent de lire /écrire sur des entrées /sorties et un troisième pour utiliser les interruptions générées par le bouton USR |
| SmartLED | Contient 2 programmes utilisant la matrice LED RGB 5x5 Réf. 651 050 et la LED RGB Réf. 6510028 |
| Switch | Contient 3 programmes permettant d'utiliser le bouton USR |
| Timer | Contient 4 programmes permettant d'exploiter le timer |
| Uart | Contient deux exemples d'utilisation du mode UART (Universal Asynchronous Receiver Transmitter) |
| UsbHID | Contient 3 exemples de programmation de logiciel HID (human Interface Device) pour un Joystic, un clavier et une souris.<br>Ces programmes sont utilisables uniquement quand le mode USB HID a été sélectionné. |
| UsbVCP | Contient 2 programmes pour l'utilisation du périphérique USB Progammes uniquement disponibles en mode Dual VCP |
| USRanger | Contient un programme utilisant l'émetteur récepteur Ultra sons Réf. 651 049 |

## 3- Le port USB de Plug'Uino® Py vu par l'ordinateur

Plug'Uino® Py est vue par l'ordinateur comme un périphérique USB composite (USB 2 en 1) ç-à-d vue comme deux périphériques à la fois, parmi les quatre suivants:

| | |
|---|---|
| **VCP** | : Virtual COM Port, port série USB |
| **MSC** | : Mass Storage Class, stockage USB (carte microSD) |
| **WebUSB** | : communication avec une page Web |
| **HID** | : Humain Interface Device. Ce mode permet de programmer l'action d'un clavier, d'une souris, d'un Joystick…. sans que l'objet ne soit physiquement présent.<br>Exemple d'application :<br>si nous souhaitons générer un fichier de données d'un capteur avec une mise en forme qui aurait nécessité la génération de caractères spéciaux comme retour de chariot, tabulation…<br>Une des possibilités de ce mode HID est de permettre de simuler un clavier par l'envoi de ces caractères comme si un clavier réel l'avait fait après chaque prise de mesure.<br>Voir les scripts dans :<br>E:\Samples\UsbHID<br>Joyestick.py , Keyboard.py ou Mouse.py |

Parmi les 6 combinaisons qui permettent d'associer 2 modes parmi 4, seules quatre combinaisons sont implémentées.
On trouve ainsi le mode Port Série dans chacun des 4 modes suivants :

| | |
|---|---|
| 'VCP+MSC' | Port série et Carte SD |
| 'VCP+WEBUSB' | Port série et WebUSB |
| 'VCP+VCP' | Deux ports série |
| 'VCP+HID' | Port série et HID |

Pour choisir un mode sur l'interface directement :
Après sa mise sous tension:
Appui sur RST +USR simultanément
choisir USB Option (appui court sur USR pour défiler)
Valider (appui long sur USR)
et on choisit le mode voulu.
A noter que le mode VCP est d'emblée choisi, l'utilisateur peut choisir le second mode.

Pour choisir un mode par programmation :
Ouvrir usbOption.py dans le dossier conf sur la carte SD E:\conf\usbOption.py
Dans lequel on lit :

```
from ppy import usb_mode
usb_mode('VCP+MSC')
```

Il suffit de modifier ‘VCP+MSC’ dans
usb_mode('VCP+MSC')

Par l’une des propositions

# 'VCP+WEBUSB'
# 'VCP+VCP'
# 'VCP+HID'

Le système affecte un numéro de port à chaque mode.
En changeant de mode, n’oubliez pas de revérifier le numéro du COM alloué.

## 4- Traitement de quelques messages d'erreurs

### A- Erreur de librairie

En exécutant le début du script
1 "" Sciencéthic mesures pour loi de Mariotte
2 lecture sur une broche analogique .....
3 capteur branché sur la broche analogique A1, LED branchée sur la broche digitale D2"""
4
5 # Make sure keep libPlugPy.py aside
6 from libPlugPy import plugpy

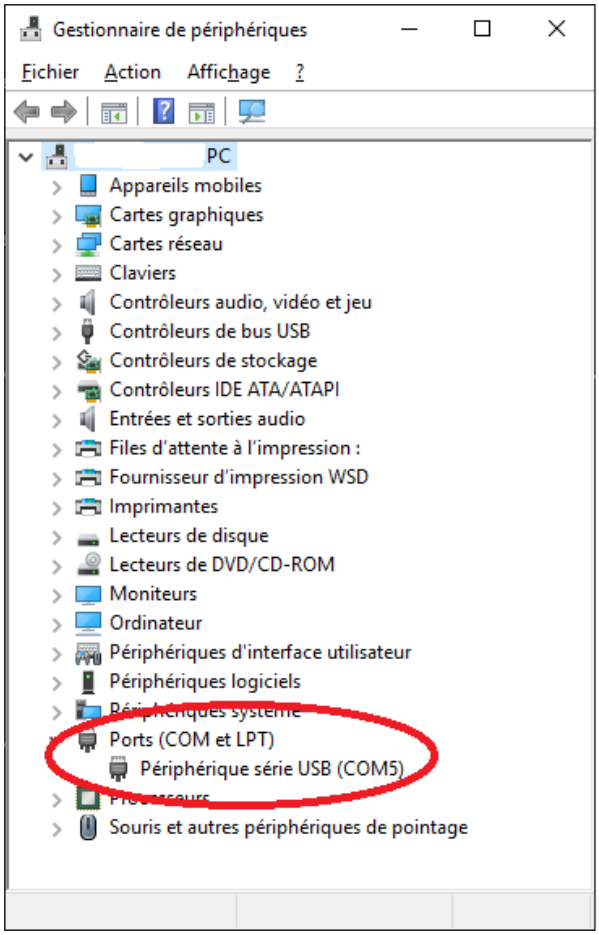
Traceback (most recent call last):
File "E:\TP\Mariotte_law_PlugPy.py", line 6, in <module>
from libPlugPy import plugpy
ModuleNotFoundError: No module named 'libPlugPy'

**Correction** :
Assurer-vous que la librairie se trouve dans la racine du dossier où se trouve le script à exécuter.

### B- Erreur de port COMxx sous WINDOWS 10

Traceback (most recent call last):
File "E:\..... line 17, in <module>
ppy = plugpy( myPlugPySerialPort )
"E:\..... line 147, in __init__
raise plugpyError('failed to access ' + device)
libPlugPy.plugpyError: failed to access COM8

Pour corriger , il faut vérifier le numéro du port alloué à Plug'Uino® Py
Sous Windows 10 :
Touche Windows + R
Exécuter ‘’devmgmt.msc’’




**Correction**:

Remplacer dans le script Com8 par Com5

## C- Erreur de port COMxx sous WINDOWS 7 (Installation de pilote)

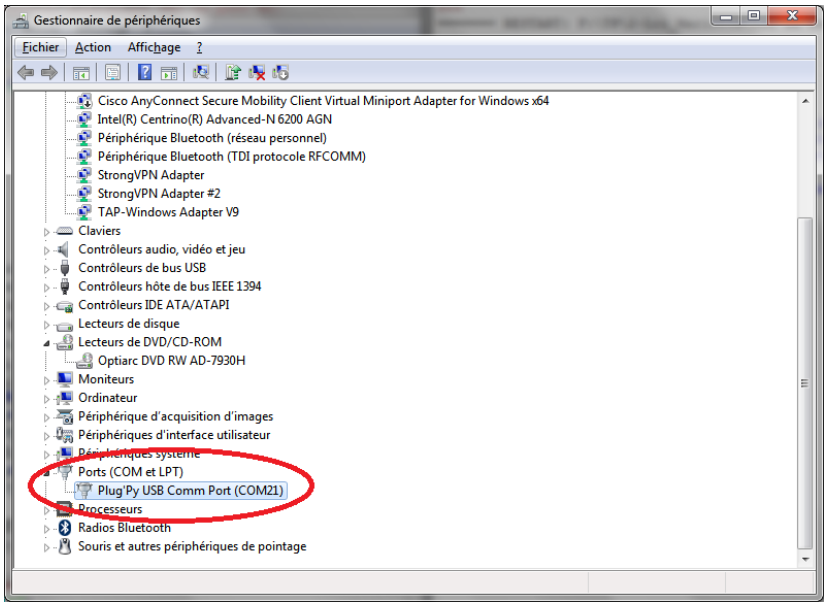
### Erreur de port Windows 7

L'erreur de port COM

Traceback (most recent call last):
File "E:\..... line 17, in <module>
ppy = plugpy( myPlugPySerialPort )
"E:\..... line 147, in __init__
raise plugpyError('failed to access ' + device)
libPlugPy.plugpyError: failed to access COM8

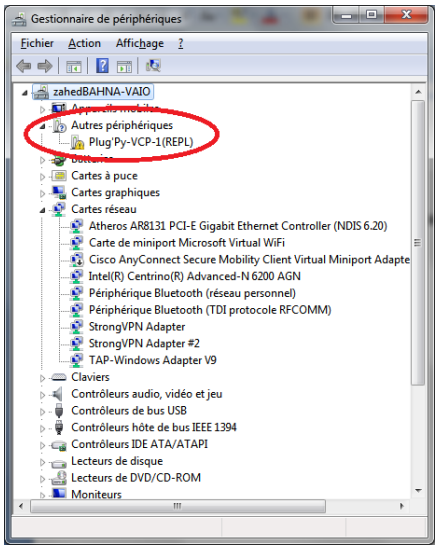
se traite de la même façon que sur Windows 10, à savoir

Touche Windows + R
Exécuter ''devmgmt.msc''

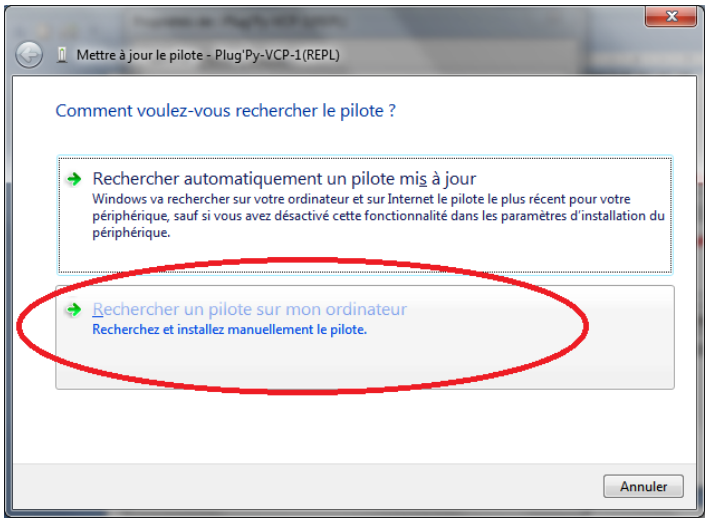



Mais si le pilote n'a jamais été installé, vous n'allez pas voir le Plug'Py dans les Ports (COM et LPT).
Il faut passer donc par l'installation du port

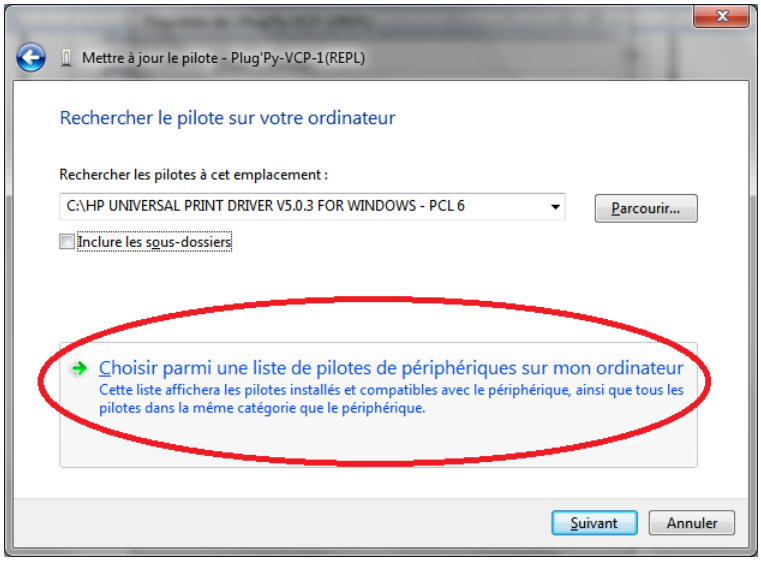
Clic droit sur Autres Périphériques et Plug’Py-VCP-1 (REPL)




Sélectionner « Recherchez et installez manuellement le pilote ».

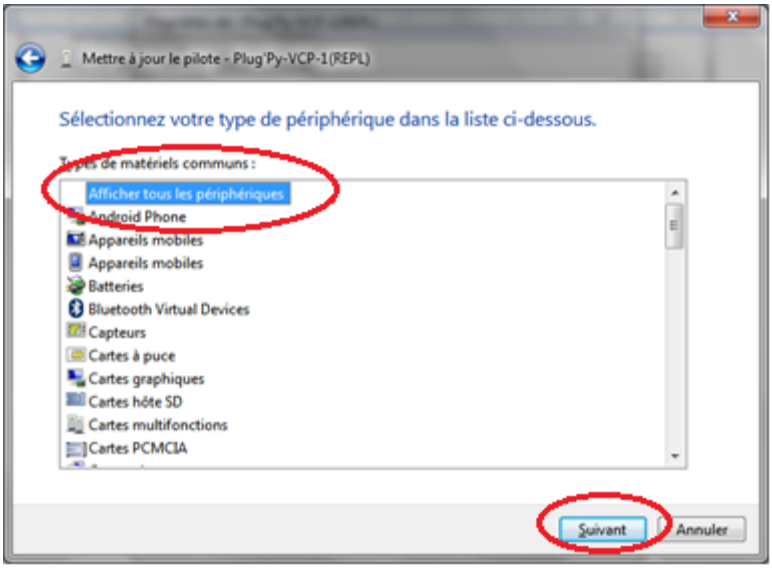


Cliquer Choisi parmi une liste…. Et suivant

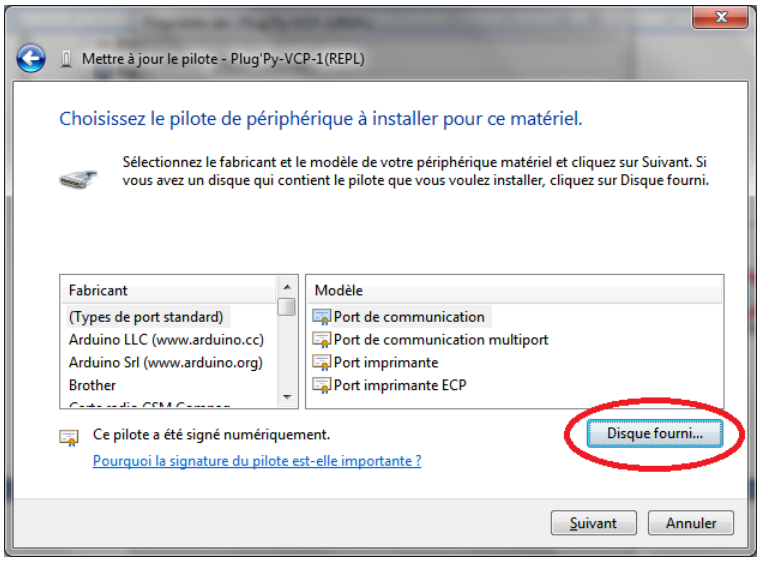



une liste apparaît

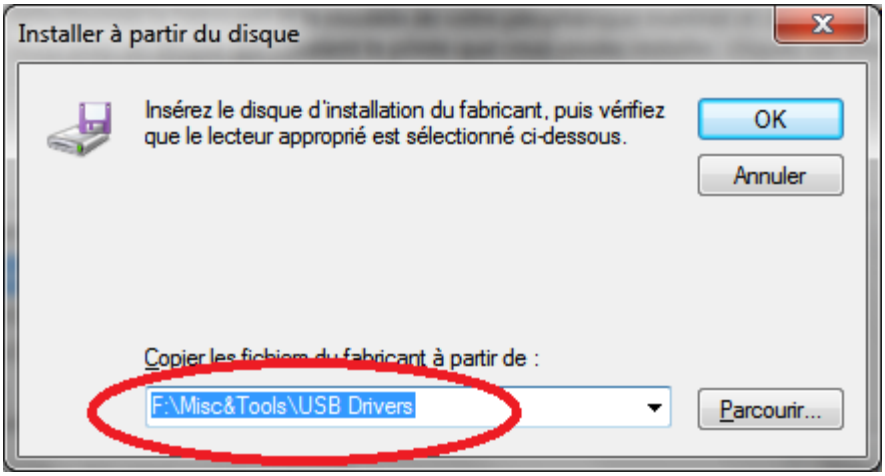
Maintenir la sélection sur ‘’Afficher tous les périphériques’’
Cliquer sur suivant.



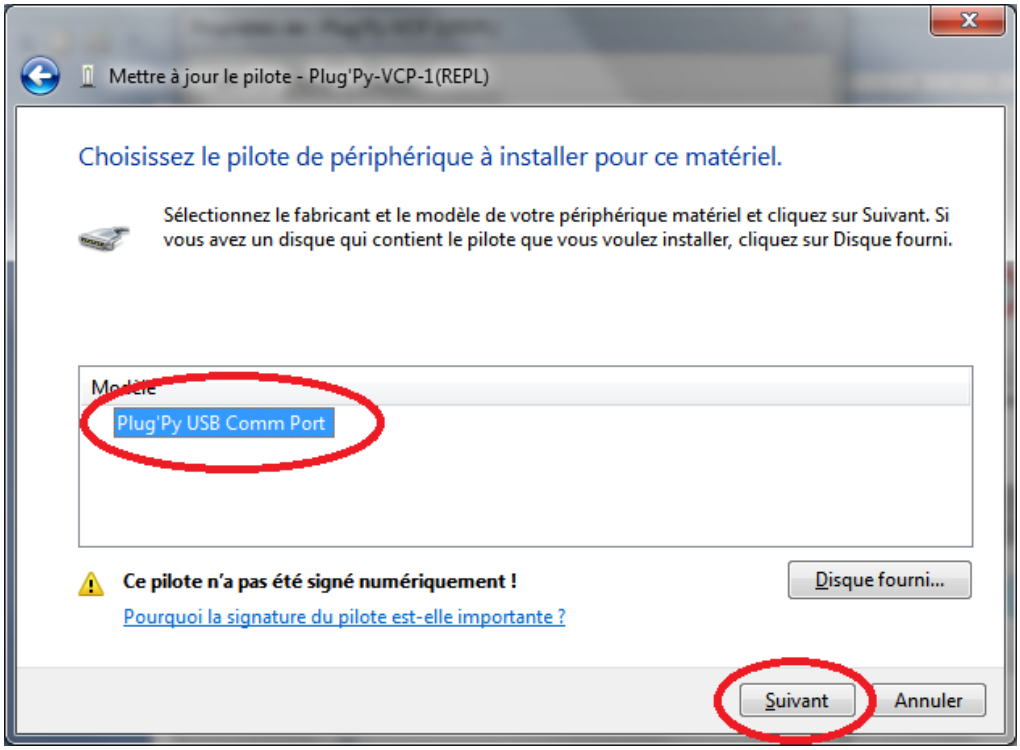
Cliquer sur ‘’Disque fourni’’



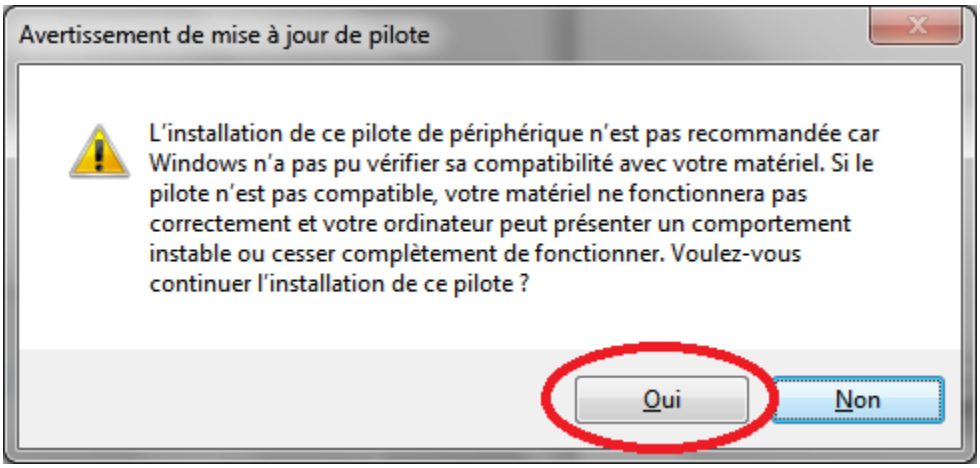

Donner le chemin Lettre de l’unité de stockage suivi de :\Misc&Tools\USD Drivers
Dans notre cas, la lettre ‘’F’’ est attribuée à l’unité de stockage



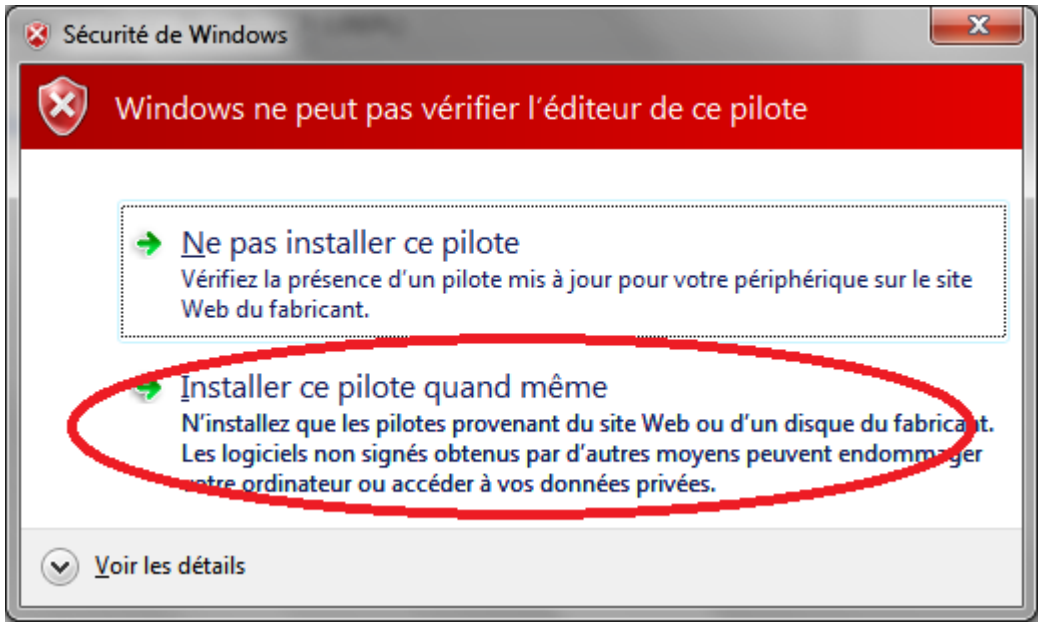
Sélectionner Plug’Py USB Comm Port et cliquer sur suivant



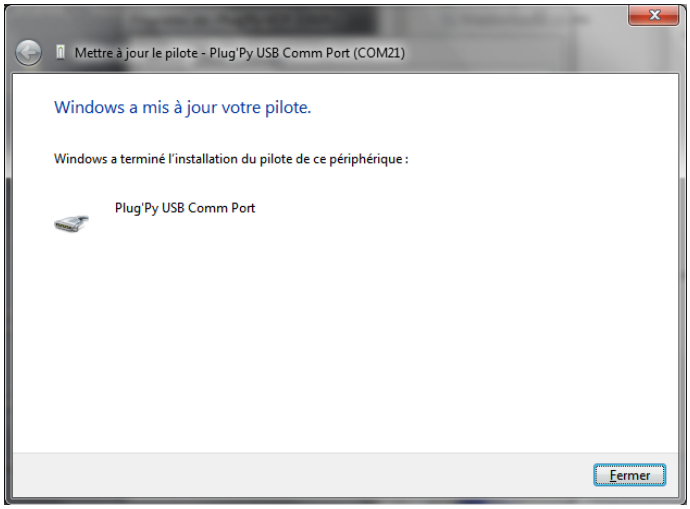

Un message d’avertissement apparaît
Cliquer sur oui



Un message de sécurité de Windows s'affiche.
Cliquer sur Installer ce pilote quand même




Après installation un message vous avertit de la bonne installation du pilote et vous indique le numéro de COM alloué.




A noter que dans la rubrique Ports (COM et LPT), Plug'Py est désormais reconnue et un numéro de port est désormais attribué.

Le pilote est bien installé, noter le numéro du COM accordé.