

MicroPython

A blue decorative shape in the bottom right corner of the slide, consisting of a series of parallel lines that create a sense of depth and movement.

1 - Qui suis-je ?

1.1 - Qui suis-je ?

Vincent Poulailleau

<https://github.com/vpoulailleau>

vpoulailleau@gmail.com

<https://www.linkedin.com/in/vpoulailleau/>

<https://www.lecalamar.fr/>

L'essentiel : Qui suis-je ?

Vincent Poulailleau

<https://www.linkedin.com/in/vpoulailleau/>

<https://www.lecalamar.fr/>

3 - Installation de MicroPython

3.1 - Installation des outils d'installation !

Installez Python 3.7 ou plus

Sous Linux : `sudo apt install python3.8` ou équivalent

Sous Windows 10 : passez par le store d'applications

Sinon : www.python.org/downloads

3.2 - Installation des outils d'installation !

Téléchargez le programme de flashage

<https://github.com/micropython/micropython/blob/master/tools/pydfu.py>

Par exemple :

<https://github.com/micropython/micropython/blob/cdaec0dcaffbfcb58e190738cf6e9d34541464f0/tools/pydfu.py>

3.3 - Installation des outils d'installation !

Créez un environnement virtuel, avec un shell

```
cd le_dossier_qui_va_bien  
python3.8 -m venv venv
```

Activez l'environnement virtuel

```
source venv/bin/activate sous Linux  
.\venv\Scripts\Activate.ps1 sous Windows
```

(venv) est ajouté au début du prompt

```
pip install pyusb==1.1.1
```


3.4 - Installation de MicroPython

Sur STM32F4 Discovery

<https://www.micropython.org/download#other>

Prenez la dernière version stable

Ou, à vos risques et périls, la version en cours de développement

3.5 - Installation de MicroPython

Débranchez la STM32F4 Discovery

Mettez un jumper entre VDD et BOOT0

Piquez le jumper J1 de la STM32F4
Discovery Shield (en dessous du bornier
CAN)

VDD et BOOT0 sont sur le header de droite
de la STM32F4 Discovery

3.6 - Installation de MicroPython

Branchez un câble mini-USB en haut de la STM32F4-Discovery

Pour l'alimentation électrique

Ou câblez la pin PA9 à la pin 5V

Branchez un câble micro-USB en bas de la STM32F4-Discovery

Pour le transfert du fichier (flashage)

3.7 - Installation de MicroPython

Dans le terminal avec l'environnement virtuel activé

```
python pydfu.py --list
```

Liste des périphériques en mode DFU

```
Bus 1 Device 009: ID 0483:df11
```

```
Memory Layout
```

```
0x8000000 4 pages of 16K bytes
```

```
0x8010000 1 pages of 64K bytes
```

```
0x8020000 7 pages of 128K bytes
```

3.8 - Installation de MicroPython

```
python pydfu.py --upload STM32F4DISC-20210222-v1.14.dfu
```

À adapter avec le nom du fichier téléchargé

Flashage

```
File: STM32F4DISC-20210222-v1.14.dfu
  b'DfuSe' v1, image size: 332349, targets: 1
  b'Target' 0, alt setting: 0, name: "ST...", size: 332064, elements: 2
    0, address: 0x08000000, size: 14592
    1, address: 0x08020000, size: 317456
  usb: 0483:df11, device: 0x0000, dfu: 0x011a, b'UFD', 16, 0xee0d55e3
Writing memory...
0x08000000  14592 [=====] 100%
0x08020000 317456 [=====] 100%
Exiting DFU...
Finished
```

3.9 - Installation de MicroPython

Ouvrez la nouvelle clé USB (MicroPython est vu comme une clé)

Ouvrez README.txt

3.10 - Installation de MicroPython

boot.py

```
# boot.py -- run on boot-up  
  
# can run arbitrary Python, but best to keep it minimal  
  
import machine  
  
import pyb  
  
pyb.country('FR')  
  
pyb.main('main.py') # main script to run after this one
```

Et main.py vidé

3.11 - ATTENTION

Éjectez correctement la clé USB

Sinon, risque de corruption du système de fichier

Ré-installation de MicroPython

Potentielle perte du travail en cours

3.12 - Installation de MicroPython

Remettez le jumper à sa position initiale

Appuyez sur le bouton poussoir noir (Reset)

Démarrez un terminal MicroPython

`screen /dev/ttyACM0` sous Linux

TeraTerm sous Windows

3.13 - Installation de MicroPython

Appuyez sur entrée

Tapez « 3 + 4 » puis entrée

MicroPython répond 7 !

Tapez « help() » puis entrée

3.14 - REPL

Le mode actuel est REPL

Lancé quand main.py a fini son exécution

Read **E**val **P**rint **L**oop, demande une commande, affiche le résultat

Permet de tester en temps réel

Pas de compilation, chargement, debug, recompilation...

Bienvenu dans le nouveau monde de l'informatique embarquée !

L'essentiel : Installation de MicroPython

`help()` pour avoir de l'aide

REPL : *R*ead *E*val *P*rint *L*oop