



Microcontrôleurs - Module PyPortal

[Mise à jour le 24/2/2021]



- **Ressources**

- [Adafruit PyPortal - CircuitPython Powered Internet Display](#)
- [Adafruit PyPortal - IoT for CircuitPython \(overview\)](#)
- [Documentation CircuitPython](#)

1. Généralités

Le **PyPortal** a été développé pour l'“**Internet des objets**”. Il permet de créer des interfaces graphiques à écran tactile en Python ou Arduino.



2. Principales caractéristiques

- **Microcontrôleur** : ATSAM51J20
 - Cortex-M4 à 120 MHz
 - Alimentation : 3,3V
 - Mémoire flash : 1Mo
 - Port USB natif
- **Coprocasseur** : Wi-Fi Espressif ESP32 (TLS/SSL intégré)
- **Mémoire externe** : 8Mo QSPI flash
- **Affichage** : TFT couleur 3,2 pouces 320 x 240 avec écran tactile résistif
- **Stockage** :
 - Emplacement pour carte microSD
- **Entrées/Sorties**

- Connectique pour I2C
- Connectique pour 2 entrées analogiques ou numériques
- **Divers**
 - Haut parleur
 - Capteur de lumière
 - Capteur de température (ADT7410 résolution 16bits (0,0078°C))
 - Néopixel

3. Configuration du circuit PyPortalPython

- Source sur [adafruit](#) pour une configuration minimum.

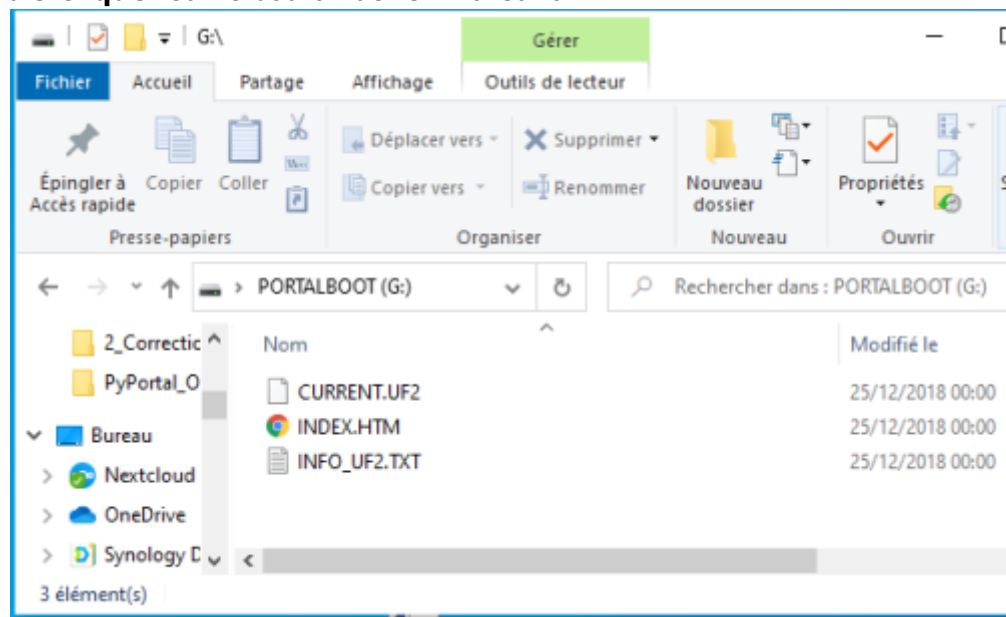
2. Mise en oeuvre

- **Mise à jour**

1. Bootloader

Source : [Update the UF2 Bootloader](#)

a. **Double-cliquer** sur le bouton de réinitialisation.



b. [Télécharger](#) le chargeur de démarrage .UF2 et le faire glisser dans la fenêtre PORTALBOOT.

2. Firmware : [Installez CircuitPython](#)

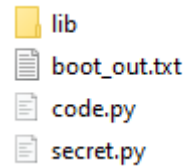
a. [Télécharger](#) la dernière version de circuitPython pour PyPortal et la faire glisser dans la fenêtre PORTALBOOT.



- **IDE : Mu** (Recommandé)

- **REPL** : activé en entrant **ctrl-c** puis une touche quelconque dans la console ⇒ **>>**.
 - **help("modules")** liste tous les modules (bibliothèques) sur la carte.

- **import board + dir(board)** pour lister les broches de la carte.



- **Organisation** minimum sur la carte PyPortal
 - **Fichiers** à créer
 - **code.py** s'exécute par défaut
 - **secret.py** pour sauvegarder le SSID et le mdp du Wifi, compte Adafruit IO, etc.
 - **Sous répertoire**
 - **lib** pour rassembler les bibliothèques nécessaires à l'application. Celles-ci seront copiées à partir du paquet (**bundle**) correspondant à la version de CircuitPython
- **Test**
- [MicroPython](#)
- [Arduino](#)

*.py

```
# Faire clignoter la LED ACT
import board # Accès au matériel de la carte
import digitalio # Accès aux entrées/sorties
import time # Temporisations

led = digitalio.DigitalInOut(board.D13)
led.direction = digitalio.Direction.OUTPUT

while True:
    led.value = True
    time.sleep(0.1)
    led.value = False
    time.sleep(0.1)
```

A faire

4. Bibliothèques

- [Téléchargement et installation](#) du bundle de bibliothèques.

5. Exemples

- **Extraits de code** illustrant les fonctionnalités de la carte (capteur, carte µSD, etc.) : [Tutoriel PyPortal CircuitPy Tutorial \(AdaBox 011\)](#)

From:

<https://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<https://webge.fr/dokuwiki/doku.php?id=microc:uc:pyportal&rev=1692434355>

Last update: **2023/08/19 10:39**

