

Initiation à l'ESP32

D.Moreaux

18 février 2026

1 Préparation

Les modules ESP32-C3 avec écran OLED peuvent être programmés de différentes façons. On peut utiliser la suite de développement de Espressif (l'entreprise qui fabrique les ESP32), la suite Arduino (avec les bibliothèques prévues pour Arduino) ou on peut y installer MicroPython¹.

Pour cette initiation, MicroPython a déjà été installé. On le trouve (ainsi qu'une documentation complète) sur le site <https://micropython.org>. Pour l'installer, on télécharge la version correspondant à la carte ou au processeur que l'on va utiliser et on utilise l'outil ESPTOOL pour écrire MicroPython.

```
esptool erase_flash
esptool write_flash 0 ES032_GENERIC_C3-....bin
```

Pour se connecter au périphérique en micropython, on utilisera l'éditeur de fichiers Thonny.

2 Led clignotante

Pour faire clignoter une LED, on aura besoin de la bibliothèque `machine` pour accéder aux broches et de la bibliothèque `time` pour mesurer le temps entre deux changements.

```
from machine import Pin
import time
```

La broche 8 est connectée à une LED, on l'utilisera donc pour le clignotement

1. Ou CircuitPython, autre environnement utilisant le langage Python mais non compatible avec MicroPython

```

from machine import Pin
import time

# Défini la broche P en sortie sur le GPIO8
p=Pin(8,mode=Pin.OUT)

# Répéter sans fin
while True:
    p.value(1)
    time.sleep_ms(500)
    p.value(0)
    time.sleep_ms(500)

```

Le code ci-dessus peut être sauvé dans un fichier sur le ESP32 et peut ensuite être exécuté en utilisant Thonny.

3 Ecran OLED

Pour utiliser l'écran, il faut d'abord copier le fichier ssd1306.py sur l'ESP32. Il s'agit d'une bibliothèque permettant d'accéder à l'écran.

L'écran communique à l'aide d'un système appelé I2C, il faudra donc d'abord configurer l'écran

```

from machine import Pin, SoftI2C
import ssd1306

i2c = SoftI2C(scl=Pin(6), sda=Pin(5))
# taille écran : 72x40
oled = ssd1306.SSD1306_I2C(72,40,i2c)

# Effacer l'écran
oled.fill(0)

# Afficher un texte
oled.text("HW",0,0)

# Mettre l'écran à jour
oled.show()

```

4 Compteur

Pour effectuer un compteur, on utilisera l'écran et un délai entre deux mises à jours. On importera donc `time` pour pouvoir faire un `sleep`.

Une variable `cnt` contiendra la valeur du compteur (il faudra la transformer en texte pour l'afficher)

```
from machine import Pin, SoftI2C
import time
import ssd1306

i2c = SoftI2C(scl=Pin(6), sda=Pin(5))
# taille écran : 72x40
oled = ssd1306.SSD1306_I2C(72,40,i2c)

# Le compteur commence à 0
cnt = 0

while True:
    oled.fill(0)
    # on augmente le compteur de 1
    cnt = cnt + 1
    # On affiche le compteur (str permet de transformer la valeur en texte)
    oled.text("val" + str(cnt),0,0)
    oled.show()
    # et on fait une pause avant de recommencer
    time.sleep_ms(200)
```

5 Mesure de température

On peut mesurer les températures à l'aide d'un composant DS18B20. Ce composant est très courant et bon marché.

Une petite platine permet de relier un DS18B20 à la broche 2 de l'ESP32.

```
from machine import Pin, SoftI2C
import time
import ssd1306
# Les deux bibliothèques pour le capteur de température.
from ds18x20 import *
from onewire import *
```

```

i2c = SoftI2C(scl=Pin(6), sda=Pin(5))
# taille écran : 72x40
oled = ssd1306.SSD1306_I2C(72,40,i2c)

# On crée l'accès au capteur
ds = DS18X20( OneWire( Pin(2)))
# Et on détecte le premier de la liste (il peut y en avoir plusieurs)
ds_id = ds.scan()[0]

while True:
    # On lance la conversion (cela peut prendre du temps)
    ds.convert_temp()
    time.sleep_ms(750)
    # On résume le résultat
    temp = ds.read_temp(ds_id)

    oled.fill(0)
    oled.text("T=" + str(temp),0,0)
    oled.show()

```

6 Aller plus loin

Le ESP32 permet bien d'autres choses. En effet, il supporte le Wifi et le Bluetooth. Des bibliothèques permettent de le faire agir comme un serveur web (comme Micropyserver) ou gérer d'autres types de matériel.

Le modèle utilisé ici est un module bon marché, d'autres permettent de disposer de plus de broches d'entrée/sortie. Il supporte aussi le protocole MQTT qui permet de transférer des données vers un serveur central ou d'écouter certains "messages" envoyés au serveur pour agir en conséquences.

7 Remarques

Avec les modules ESP32-C3 OLED, le réseau se montre un peu capricieux :

- Il est nécessaire de configurer la puissance d'émission (sinon, elle est de 0) à l'aide de `wlan.config(txpower=5)`.
- L'antenne est de mauvaise qualité (ou plutôt très mal connexion. Il est y possible d'installer une antenne (simple fil de cuivre) qui augmente fortement la portée.