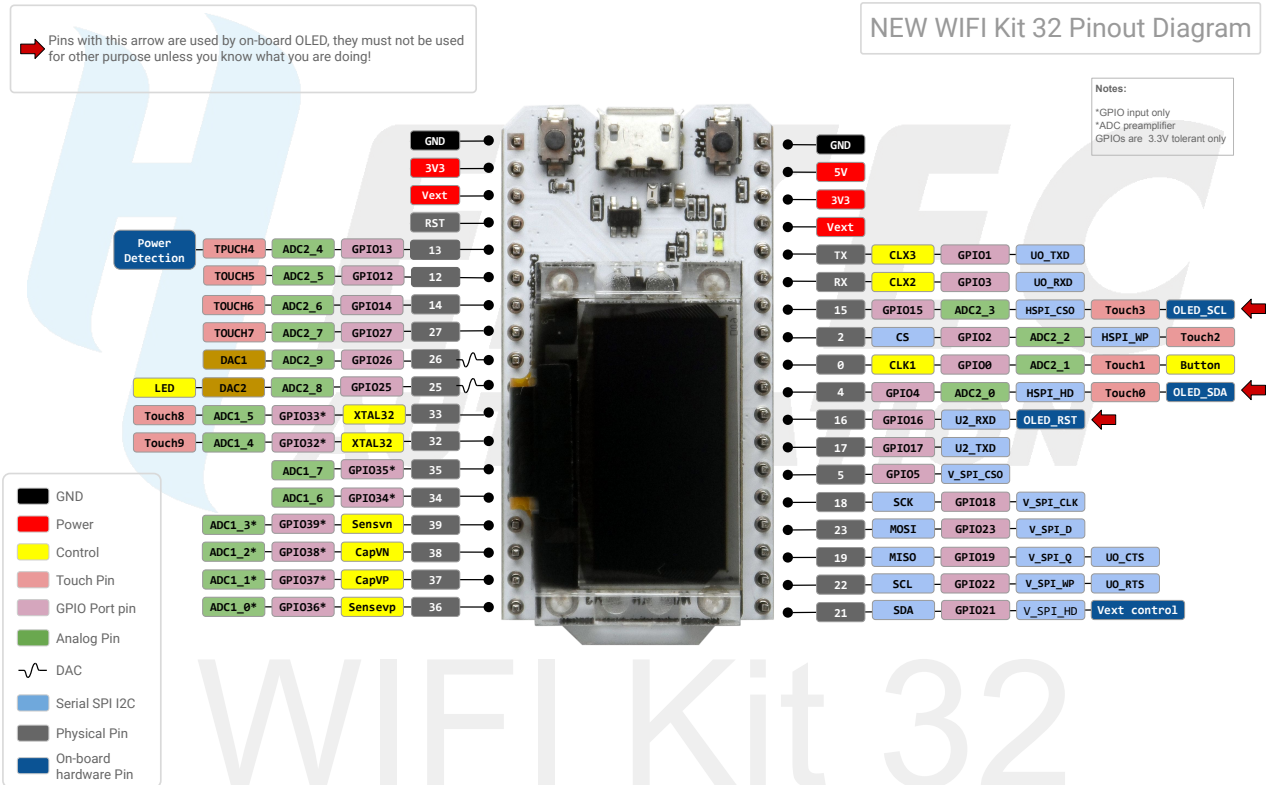


Utilisation de composants avec μPython

■ ■ ■ Schéma d'interconnexion



https://resource.heltec.cn/download/WiFi_Kit_32/WIFI_Kit_32_pinoutDiagram_V2.pdf

■ ■ ■ Exploitation du bouton « PROG »

```

from machine import Pin
from utime import sleep

button = Pin( , Pin.IN)
led = Pin( , Pin.OUT)

while 1:
    if button.value() == 0:
        print("bouton appuye")
        led.value(1)
        sleep(0.01)
        led.value(0)
    
```

Identifiez la broche connectée à la led ② et au bouton ① sur le schéma de d'interconnexion.

Travail :

Écrivez un programme qui envoie un message « bouton appuyé » par MQTT sur le topic "ESP32/bouton" lorsque l'on appuie sur le bouton.

Exploitation de l'écran OLED

Vous récupérerez la bibliothèque pilotant l'écran OLED de type SSD1306 sur : <https://github.com/micropython/micropython/blob/master/drivers/display/ssd1306.py>

Vous testerez le programme suivant en vérifiant le brochage de l'écran : RESET, bus I2C :

```
import ssd1306
from machine import Pin, SoftI2C as I2C
import time
import network

wlan = network.WLAN(network.STA_IF) # create station interface
wlan.active(True) # activate the interface
wlan.isconnected() # check if the station is connected to an AP
time.sleep_ms(500)
if not wlan.isconnected():
    print('connecting to network...')
    wlan.connect('IoT', '12344321') # connect to an AP
    time.sleep_ms(500)
    while not wlan.isconnected():
        pass
print('network config:', wlan.ifconfig())

# Heltec LoRa 32 with OLED Display
oled_width = 128
oled_height = 64
# OLED reset pin
i2c_rst = Pin(16, Pin.OUT)
# Initialize the OLED display
i2c_rst.value(0)
time.sleep_ms(5)
i2c_rst.value(1) # must be held high after initialization
# Setup the I2C lines
i2c_scl = Pin(15, Pin.OUT, Pin.PULL_UP)
i2c_sda = Pin(4, Pin.OUT, Pin.PULL_UP)
# Create the bus object
i2c = I2C(scl=i2c_scl, sda=i2c_sda)
# Create the display object
oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)
oled.fill(0)
oled.text(wlan.ifconfig()[0], 0, 0)
oled.text('INSA CVL', 0, 25)
oled.text('RnF', 0, 55)

oled.line(0, 0, 50, 25, 1)
oled.show()
```

Travail :

Écrivez un programme qui affiche sur cet écran OLED un message reçu par MQTT sur le topic "ESP32/OLED".

Exploitation du «*transceiver*» LoRa

Vous récupérerez la bibliothèque disponible à <https://github.com/martynwheeler/u-lora>.

Travail :

- Vous comparerez les fonctionnalités de cette bibliothèque avec celle-ci : <https://github.com/fantasticdonkey/uLoRa>
- Vous regarderez la configuration LoRa que l'on peut sélectionner sur le composant, en termes de «*bandwidth*», «*code rate*», «*spreading factor*» et puissance d'émission.
- Vous adapterez la bibliothèque à votre carte de développement : brochages et bus SPI. Comment peut-on se rendre compte si cela fonctionne ?
- Vous testerez l'envoi de message par LoRa et si possible la réception.