

# Pin pong Telegram

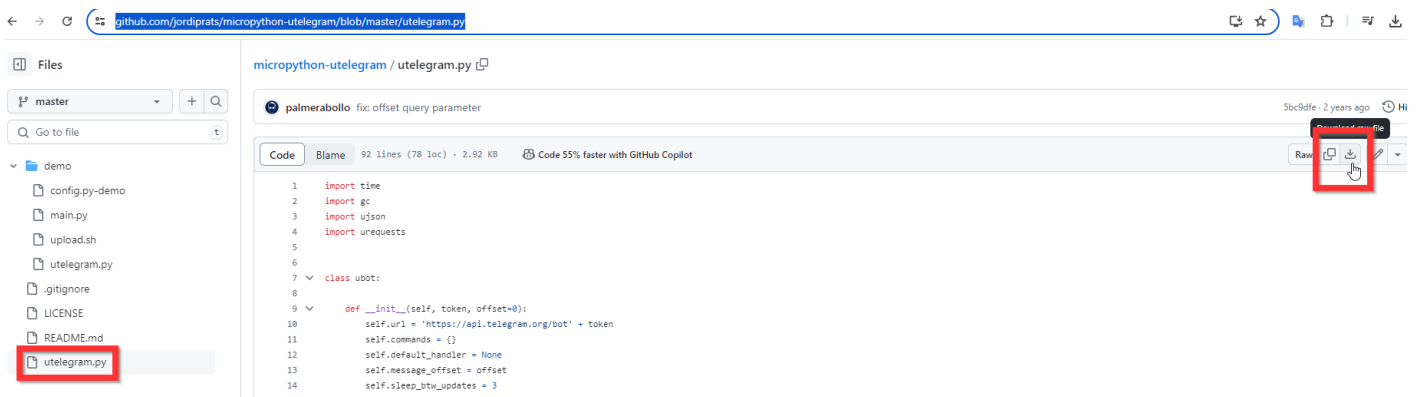
Como paso previo a enviar y recibir mensajes, vamos a realizar los pasos de este vídeo

<https://www.youtube.com/watch?v=eZkb9omr-sA>

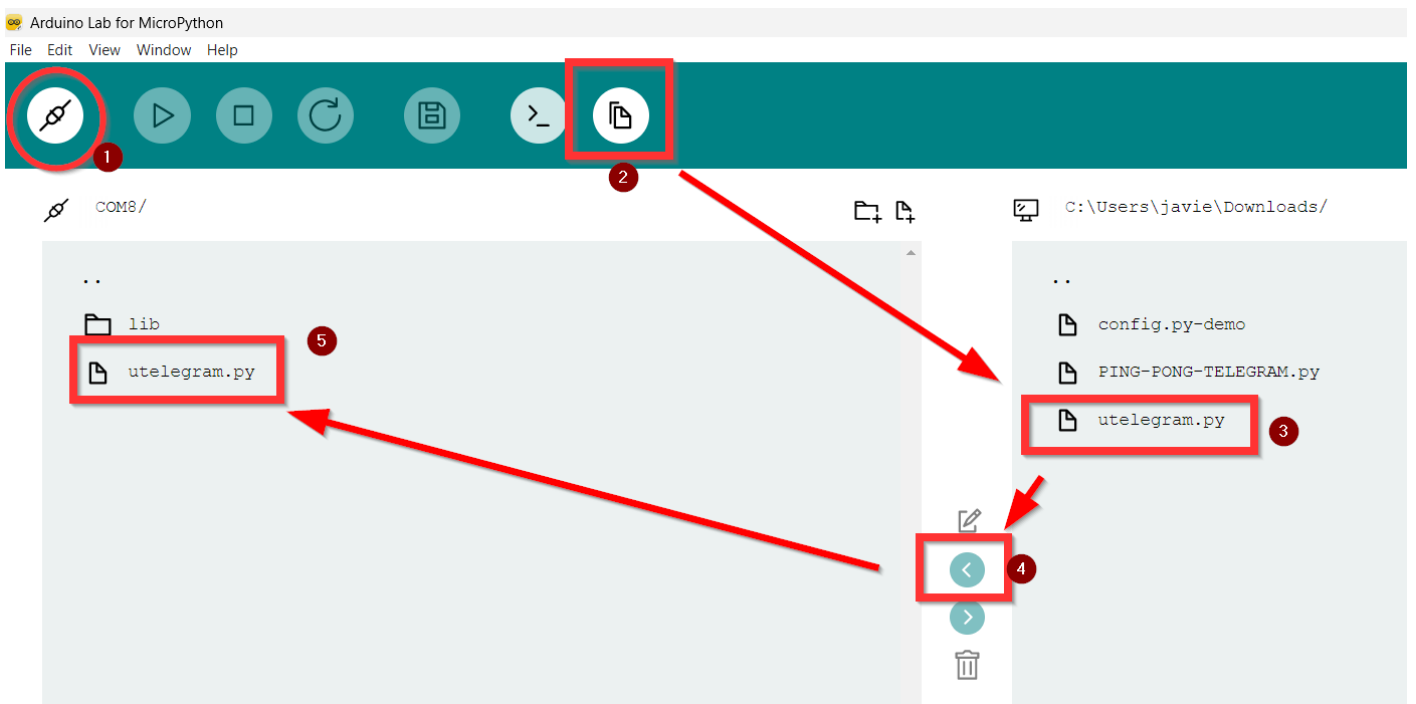
## Paso 1: Librería uTelegram.py

Del repositorio de Jordi Prats

<https://github.com/jordiprats/micropython-utelegram/blob/master/utelegram.py>



Y la cargamos dentro de nuestro ESP32, ejecutamos Arduino Lab for MicroPython, conectamos, vamos al gestor de archivos y lo llevamos dentro del ESP32 Alvik

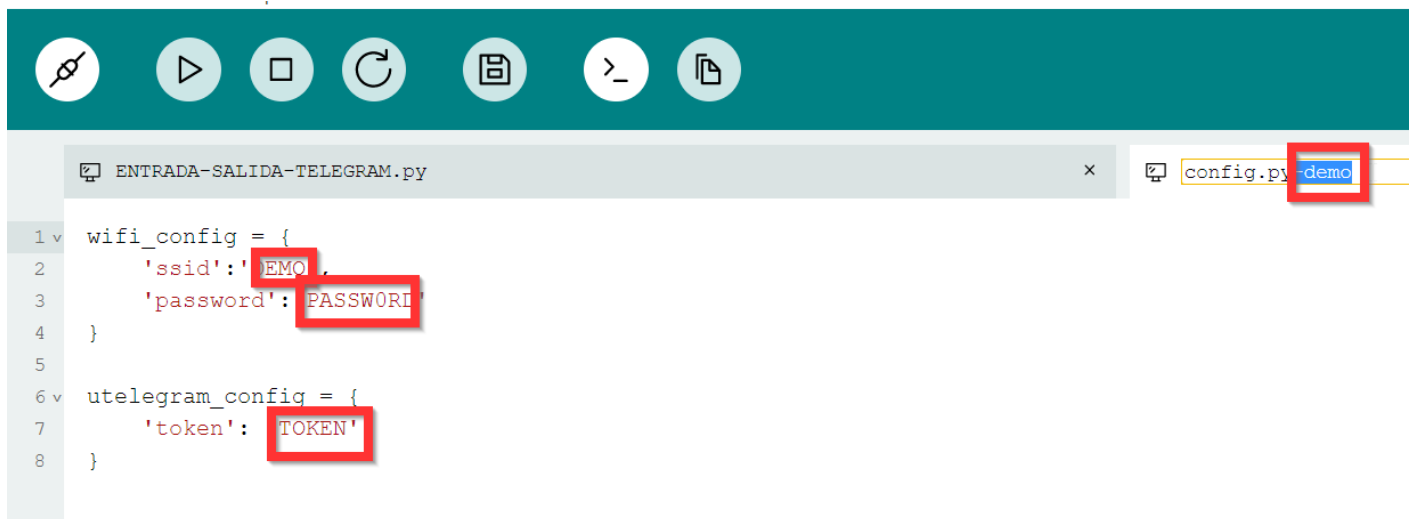


## Paso 2 Archivo config.py

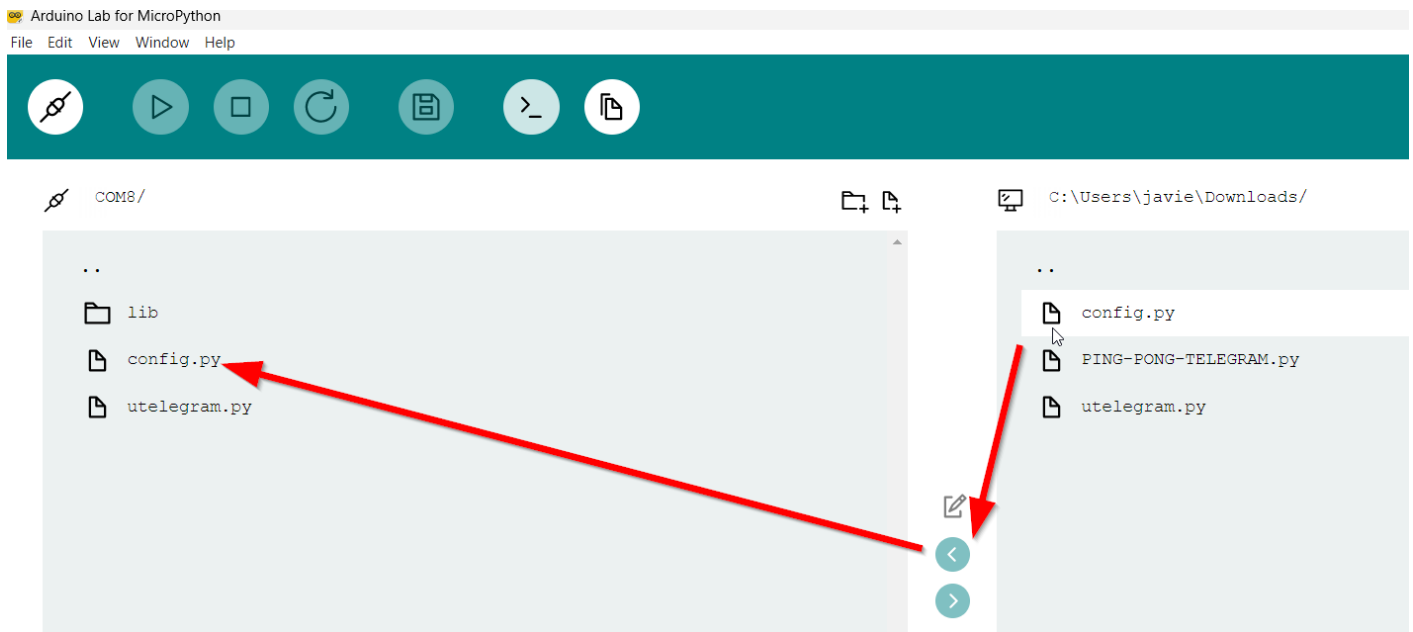
El archivo config.py no es más que el archivo que contiene la wifi y el token, se puede descargar de <https://github.com/jordiprats/micropython-utelegram/blob/master/demo/config.py-demo> o también se puede copiar y pegar de aquí mismo

```
wifi_config = {  
    'ssid':'DEMO',  
    'password':'PASSWORD'  
}  
  
utelegram_config = {  
    'token': 'TOKEN'  
}
```

Ponemos los valores de nuestra wifi SSID, PASSWORD y TOKEN y borramos del nombre el -demo y lo dejamos como config.py



y como antes, lo pasamos al ESP32 Alvik



Se podría poner esa información en el código del programa principal main.py tal y como el programa de la página <https://libros.catedu.es/books/arduino-alvik/page/mensajes-a-telegram>

### Paso 3 Programa principal main.py

El programa lo podemos descargar de <https://github.com/jordiprats/micropython-utelegram/blob/master/demo/main.py>

o de aquí mismo

tal cual, no hay que poner nuestro ssid, ni password ni token pues lo "lee" de config.py

```
from config import utelegram_config
from config import wifi_config

import utelegram
import network
import utime

debug = True

sta_if = network.WLAN(network.STA_IF)
sta_if.active(True)
```

```

sta_if.scan()

sta_if.connect(wifi_config['ssid'], wifi_config['password'])


if debug: print('WAITING FOR NETWORK - sleep 20')
utime.sleep(20)


def get_message(message):
    bot.send(message['message']['chat']['id'], message['message']['text'].upper())


def reply_ping(message):
    print(message)
    bot.send(message['message']['chat']['id'], 'pong')

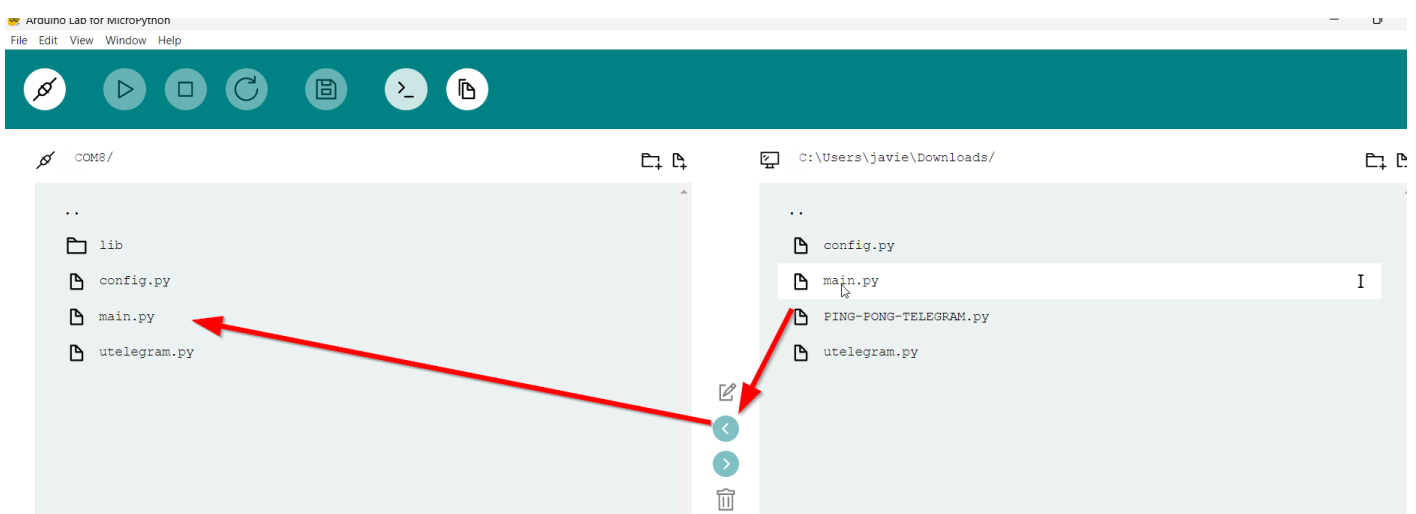

if sta_if.isconnected():
    bot = utelegram.ubot(utelegram_config['token'])
    bot.register('/ping', reply_ping)
    bot.set_default_handler(get_message)

    print('BOT LISTENING')
    bot.listen()
else:
    print('NOT CONNECTED - aborting')

```

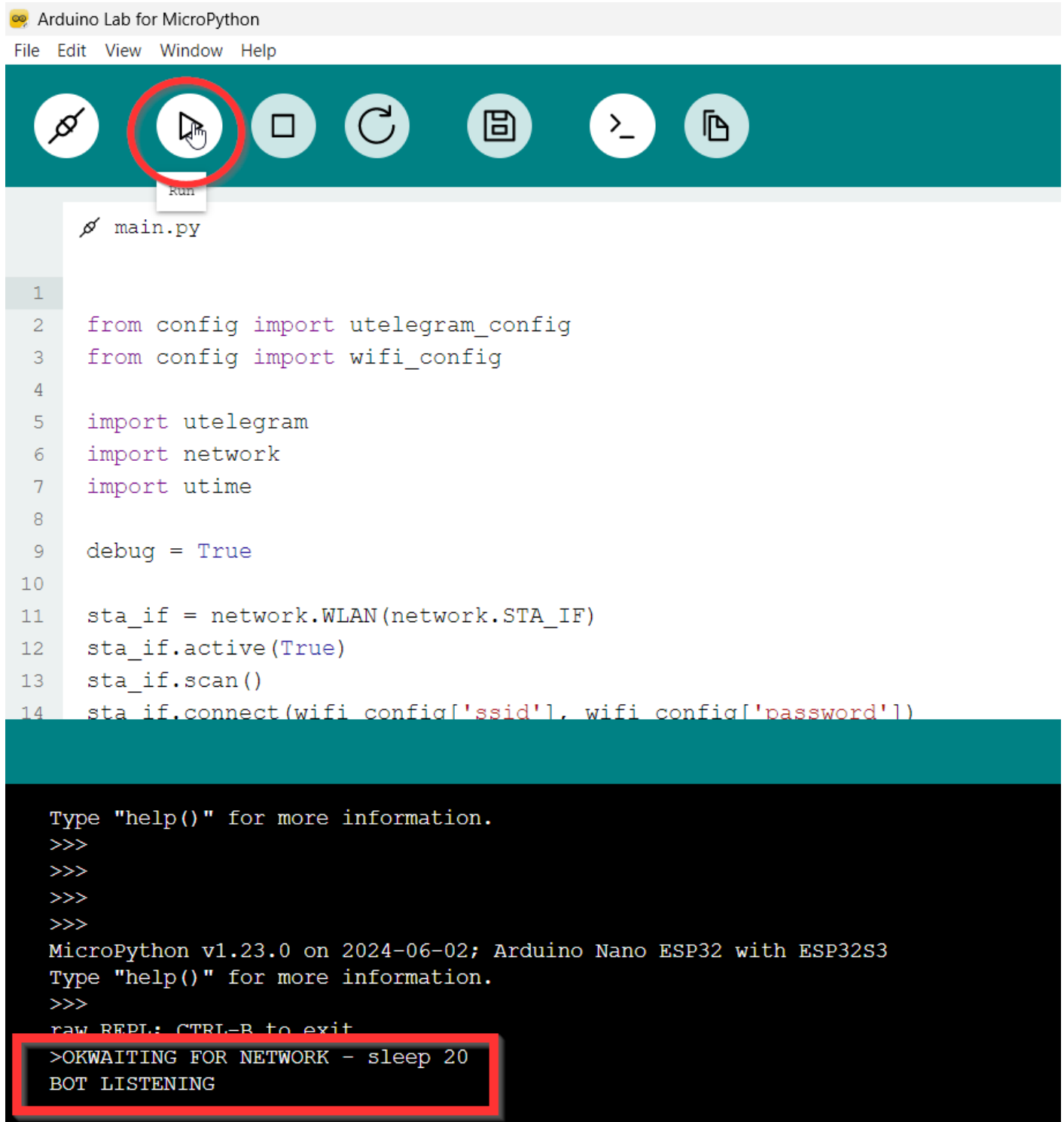
❏ No sé por qué hay que esperar 20 segundos en `utime.sleep(20)` ❏ ¿sospecho que necesita tiempo para estar preparado para "escuchar"

Y lo llevamos al ESP32



## Ejecución

Pulsamos el main.py del ESP32 (no hace falta encender Alvik pues todas las instrucciones son sólo del ESP32), ESPERAR 20 SEGUNDOS hasta que aparezca BOT LISTENING



Nos vamos a Telegram al usuario del bot que hemos creado, le tecleamos **/ping** y contesta el ESP32 **pong**



<https://www.youtube.com/embed/eZkb9omr-sA>

---

Revision #4  
Created 6 July 2024 22:07:45 by Javier Quintana  
Updated 7 July 2024 21:33:56 by Javier Quintana