ESCUELA DE INGENIERÍA ELÉCTRICA

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

Guía 1

Introducción Raspberry Pi Pico

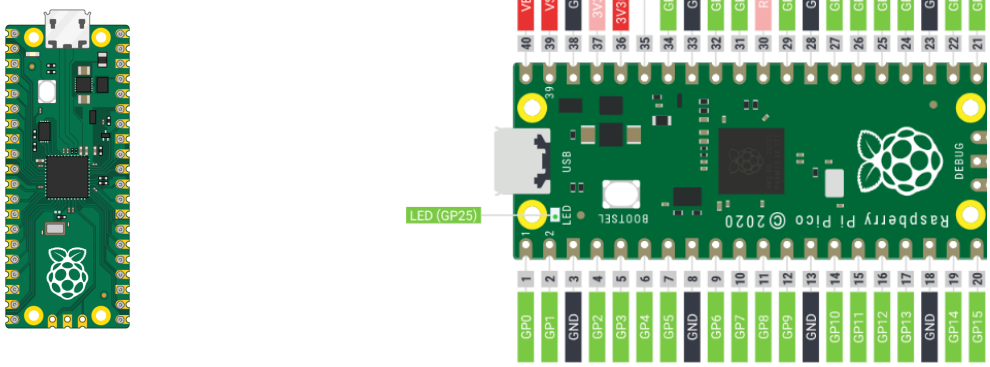
Curso: EIE390 - Robótica e Inteligencia Artificial
Profesor: Guillermo Cid Ampuero

Objetivos

- Objetivo 1: Configurar Raspberry Pi Pico para su uso en el Laboratorio.
- Objetivo 2: Configurar Thonny para programar en MicroPython.
- Objetivo 3: Progamar en MicroPython.

1 Introducción

La Raspberry Pi Pico es una placa de desarrollo de microcontrolador de bajo costo y alto rendimiento, diseñada por la Raspberry Pi Foundation.



(a) Raspberry Pi Pico (b) Pin Out Rpi Pico

Figura 1: Componentes del laboratorio

En esta oportunidad, para programar la RPI Pico, utilizaremos MicroPython.
Características:

- El chip microcontrolador RP2040 diseñado por Raspberry Pi en el Reino Unido.
- Procesador ARM Cortex M0+ de bajo consumo.
- 26 pines GPIO multifunción.
- 264kB de SRAM, y 2MB de memoria Flash a bordo. 2×SPI, 2×I2C, 2×UART, 3×12-bit ADC, 16×canales PWM controlables.
- Modo de funcionamiento de baja potencia (sleep) y modos de inactividad.

1.1 MicroPython

MicroPython es una implementación completa de Python3 pero para microcontroladores, como la RPI Pico.
Características principales

- Python completo: Implementa la sintaxis central de Python 3.4
- Optimizado para microcontroladores: Requiere mínimos recursos (256KB de código y 16KB de RAM)
- REPL interactivo: Permite probar código directamente en el dispositivo
- Amplia compatibilidad: Funciona en ESP32, ESP8266, STM32, Raspberry Pi Pico, entre otros
- Bibliotecas específicas: Incluye módulos para interactuar con hardware (GPIO, ADC, PWM, etc.)

Para poder cargar nuestros códigos en la RPI Pico, utilizaremos **Thonny**. Thonny es un entorno de desarrollo integrado gratuito y de código abierto para Python.

2 Desarrollo

2.1 Configuración MicroPython y RPI Pico

Lo primero que debemos realizar para poder utilizar nuestro microcontrolador con Thonny, es configurar la RPI Pico. Para eso debemos seguir los siguientes pasos.

Paso 1

Conectamos el Cable MicroUsb a nuestra RPI Pico y presionando el botón de *BOOTSEL* conectamos el microcontrolador a nuestro computador.

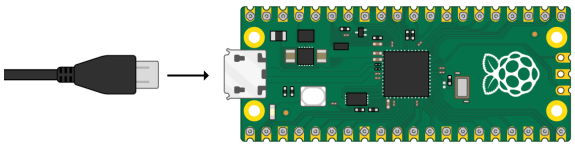
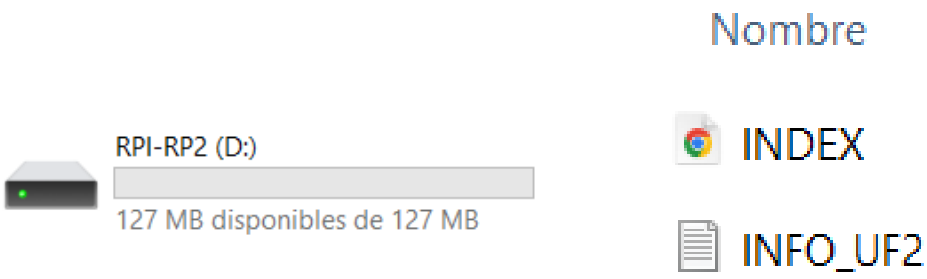


Figura 2: Conectamos nuestra RPI Pico

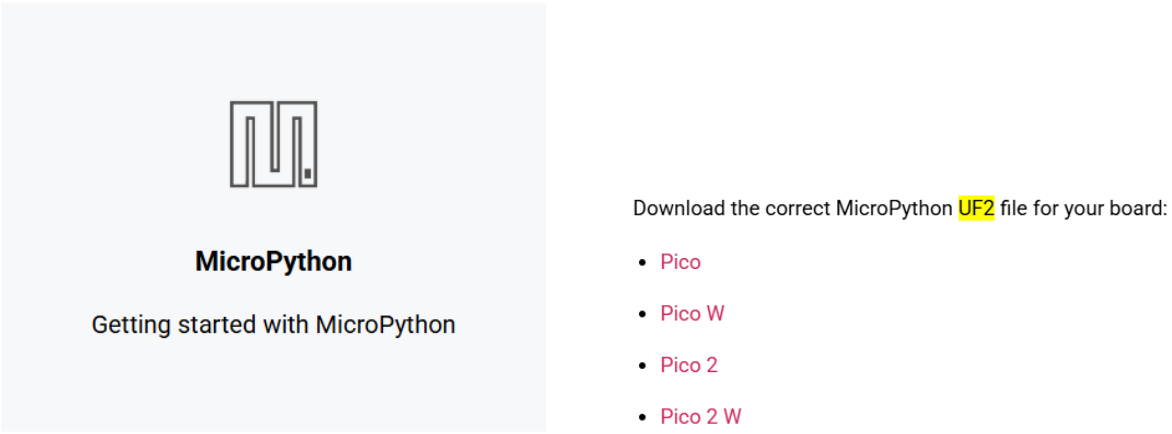
Inicialmente tu computador la reconocerá como un Disco Externo, dentro de este "disco.^encontraremos un documento *INDEX* el cual nos llevará a la pagina de Raspberry Pi.



(a) Primera conexión de RPI Pico a nuestro PC (b) index.html

Figura 3: Conexión RPI Pico al PC

Dentro de la pagina seleccionamos **MicroPython** y luego descargamos el archivo UF2 que corresponda a nuestro microcontrolador, en nuestro caso, descargamos el UF2 de la *PICO W*.



(a) MicroPython (b) UF2 de Pico-W

Figura 4: Descargar MicroPython

El archivo descargado lo arrastramos al "disco"que nos apareció al conectar la RPI Pico , y desde ese instante **la RPI Pico se reiniciará sola y comenzará a comer MicroPython en la RPI Pico.** Luego si revisamos nuestro PC, ya no nos aparecerá como un "disco"si no que ahora desde nuestro *Administrador de Dispositivos* podremos observar que se reconoce como un **Dispositivo Serie USB** con su *COM* respectivo.

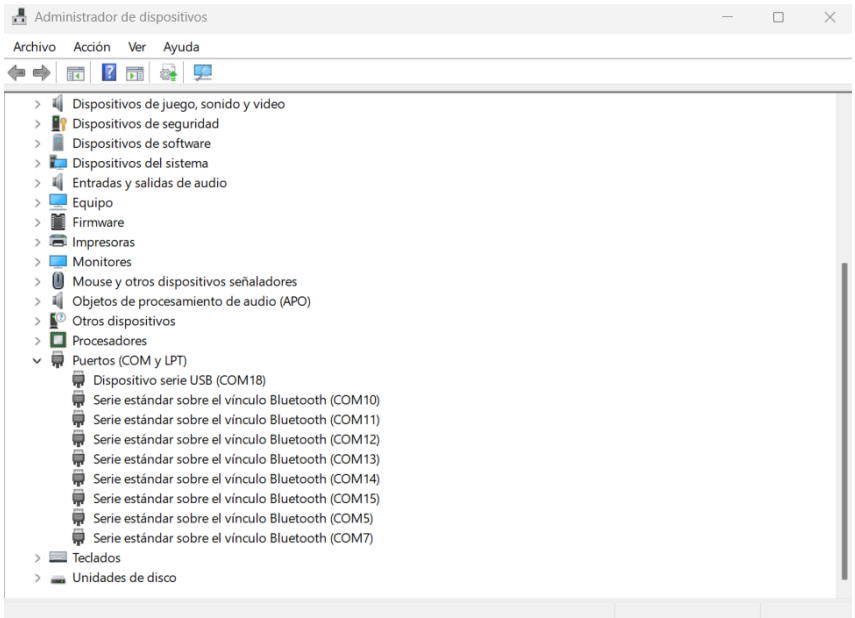


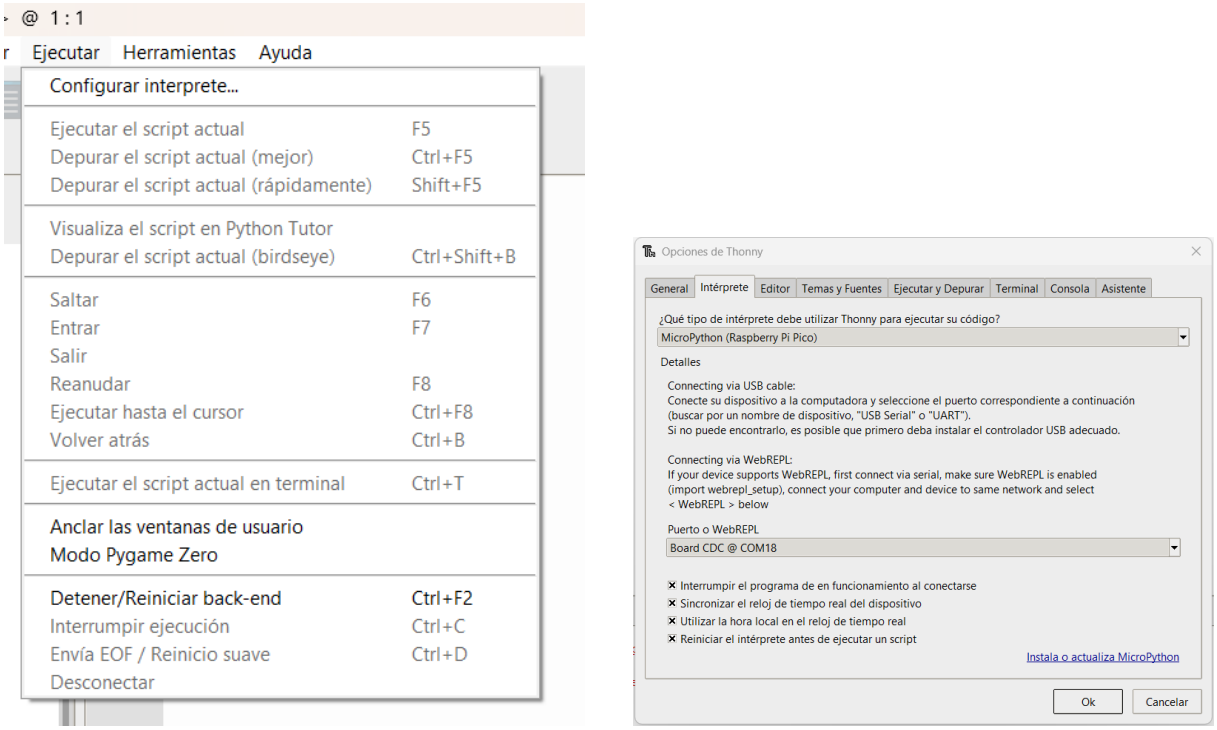
Figura 5: RPI Pico como dispositivo USB Serial con su respectivo COM

Paso 2 Ya que tenemos nuestra Rpi Pico configurada, pasamos a descargar **Thonny** (Link de descarga de Thonny)

Paso 3 Una vez descargado el Programa Thonny, debemos configurarlo para poder utilizarlo con la RPI Pico. Para eso, realizamos el siguiente paso a paso:

- Seleccionamos la Pestaña Ejecutar.
- Seleccionamos *Configurar Interprete*

- Seleccionamos **MicroPython (Raspberry Pi Pico)**
- Seleccionamos el Puerto COM al cual esta conectada nuestra Raspberrry Pi Pico.



(a) Pestaña Ejecutar en Thonny (b) Seleccionar Interprete y Puerto

Figura 6: Configurar Thonny

Paso 4 Una vez configura tanto la RPI Pico como Thonny, procedemos a escribir nuestro primer código de prueba, este permitirá parpadear el LED que viene incorporado en la RPI Pico.

```
1 from time import sleep
2 from machine import Pin
3
4 led = Pin(25, Pin.OUT)
5
6 while True:
7     led.toggle()
8     sleep(1)
```

Luego para poder guardar nuestro código, tecleamos **CTRL + S** y nos dará la opción de guardar nuestro código en nuestro PC o en la RPI Pico, en este caso, seleccionamos la RPI Pico y lo guardamos como **main.py**, al igual que los códigos de Python. Luego para ejecutar presionamos el botón verde y podremos observar como el LED de la RPI Pico parpadea.

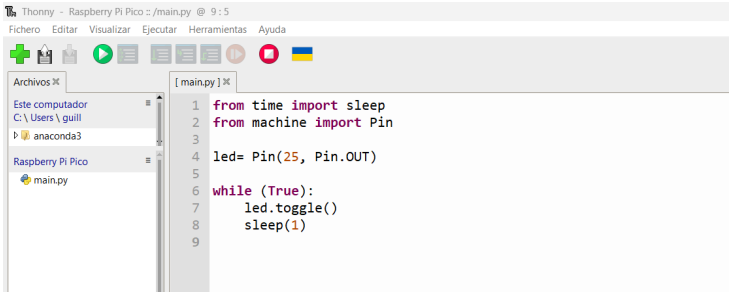


Figura 7: Iniciar Código en RPI Pico

2.2 IMPORTANTE

Importante 1 Cuando se desconecta la RPI Pico y se vuelve a conectar, el microcontrolador siempre buscará el archivo que se llame **main.py** y será el que ejecutara de manera inmediata. Si , ustedes al código anterior lo llamaron con otro nombre y conectan y desconectan la RPI Pico, el código no se ejecutará de manera inmediata.

Importante 2

Cuando ustedes conecten su RPI Pico al PC, en la parte inferior debe les aparecer el Puerto y el nombre de la placa, de esta manera se sabe que Thonny ya reconoció la placa y están trabajando en ella.Además, otra manera de confirmar, es que en la sección de archivos aparecen los códigos de su RPI Pico y no archivos de su PC.

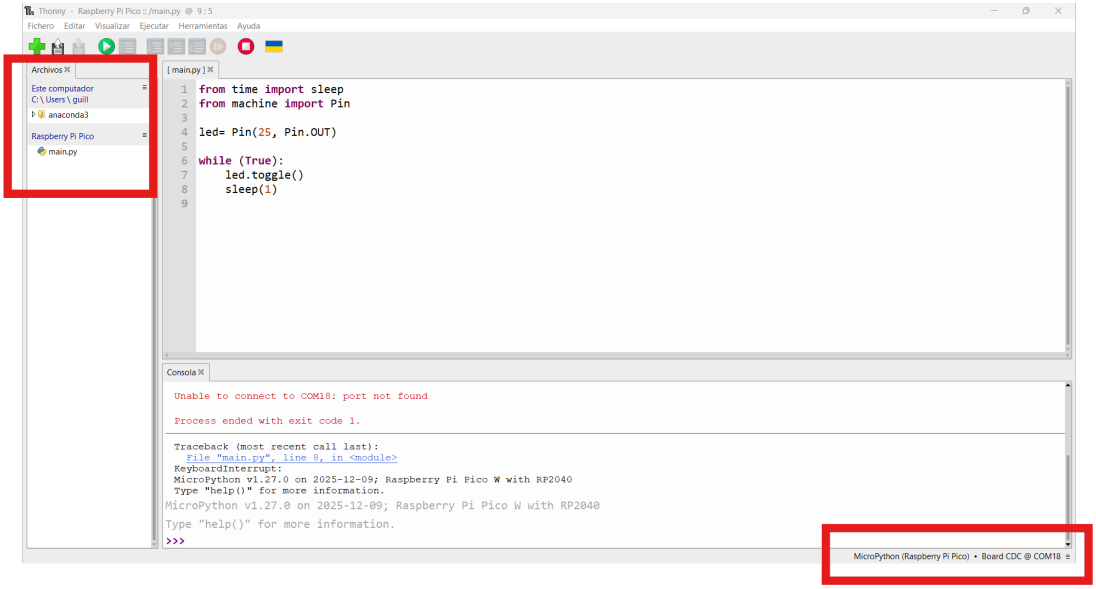


Figura 8: Raspberry Pi Pico conectada de manera correcta

3 Sigamos Programando

Como pudieron ver , MicroPython es muy similar a trabajar con Python, tenemos librerías y una sintaxis bastante familiar. Por lo que vamos a ir trabajando con algunos códigos sencillos para ir adentrándose en MicroPython, los cuales se encuentran en el siguiente repositorio de GitHub.

Repositorio de GitHub

1. Ciclo For y Condicionales con MicroPython.
2. Leer Sensor de Temperatura Interna
3. Uso de Funciones y archivos externos
4. Recibir comandos por terminal
5. Código Final solicitando temperaturas
6. Código Final solicitando temperaturas usando main
7. Código Final con lecturas en tiempo real

4 Referencias

1 Link 1

2 Link 2

3 Link 3