Introduction:

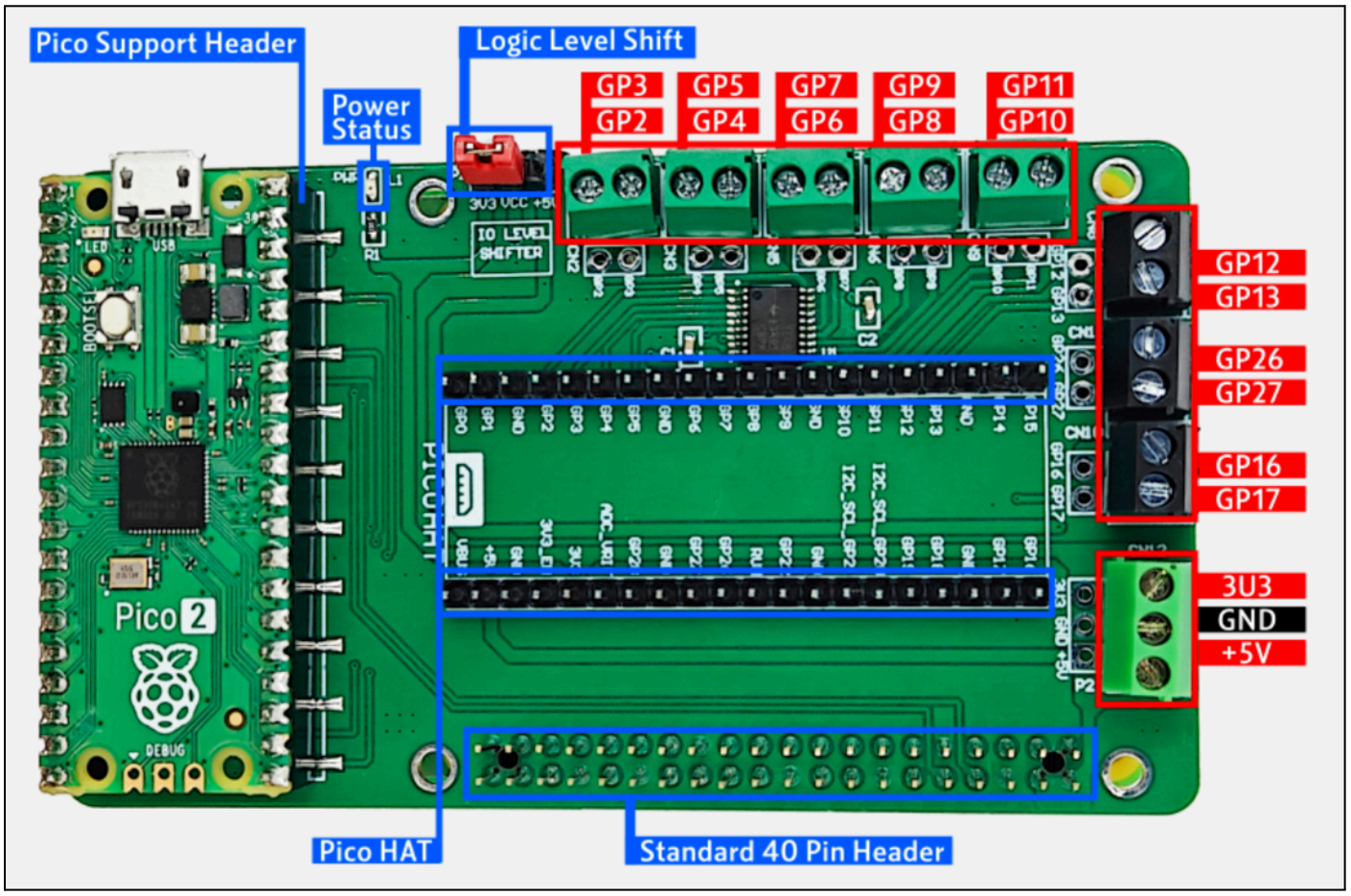
The PicoBridge IO HAT for Raspberry Pi is a versatile expansion board designed specifically for all Raspberry Pi models with a standard 40-pin GPIO layout. This HAT enhances your Raspberry Pi's capabilities, ensuring seamless integration for both beginners and advanced users.

When you attach the PicoBridge IO HAT for Raspberry Pi, you unlock extra GPIO pins, allowing you to stack additional Raspberry Pi HATs while maintaining access to critical GPIO functionalities.

Key Features:

- Standard 40 pin header to support Raspberry Pi Models
- Pico Header support for easy integration of Pico/Pico W/Pico 2
- Connector for Pico HAT interfacing
- Pico GPIOs breakout as screw terminal and 2.54" header
- Some GPIOs with 5V/3.3V Logic Level Shift options
- 5V and 3.3V Power supply breakout as 3.5mm screw terminal and 2.54" Header

Pinout Details:



Getting Started with HAT

Setup Raspberry Pi

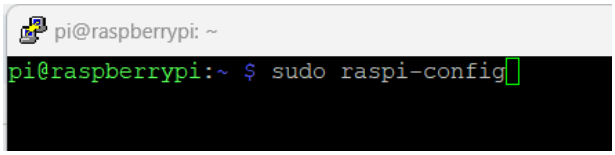
Visit official site of Raspberry Pi to get started =>

https://www.raspberrypi.com/documentation/computers/getting-started.html

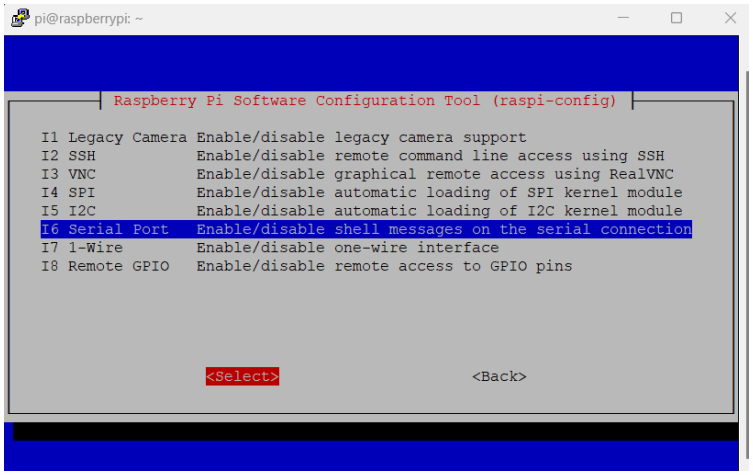
Follow the guide above to set up your Raspberry Pi with a suitable Operating System. Once done, connect Screen, keyboard and mouse.

Even you can follow guide here if interested to setup Raspberry Pi headless to use without keyboard & Screen => https://drive.google.com/file/d/1KP6EZQLhoLEB4NboW8sT6slN3HPN_gxO/view?usp=sharing

Once setup done then you are required to enable the serial interface of Raspberry Pi to use with Pico serial communication. To do this, run the command **sudo raspi-config**.



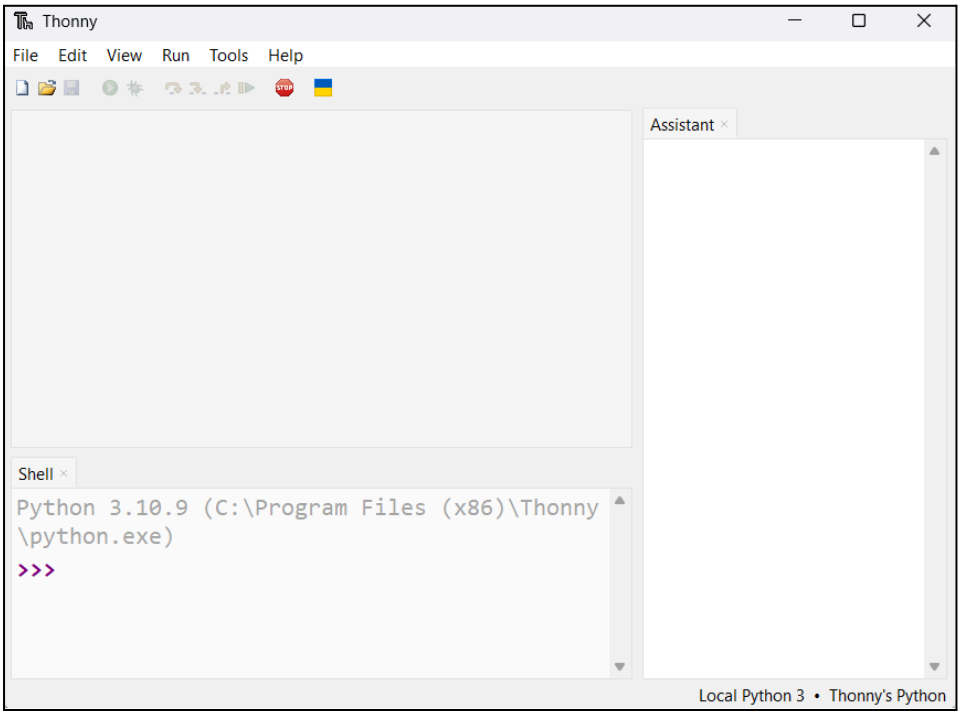
This opens the Raspberry Pi user interface (UI), where you can configure different Raspberry Pi settings.



Here you **disable Shell** over Serial and **enable the Serial port** as it might cause a collision between the interfaces. Reboot once after setting.

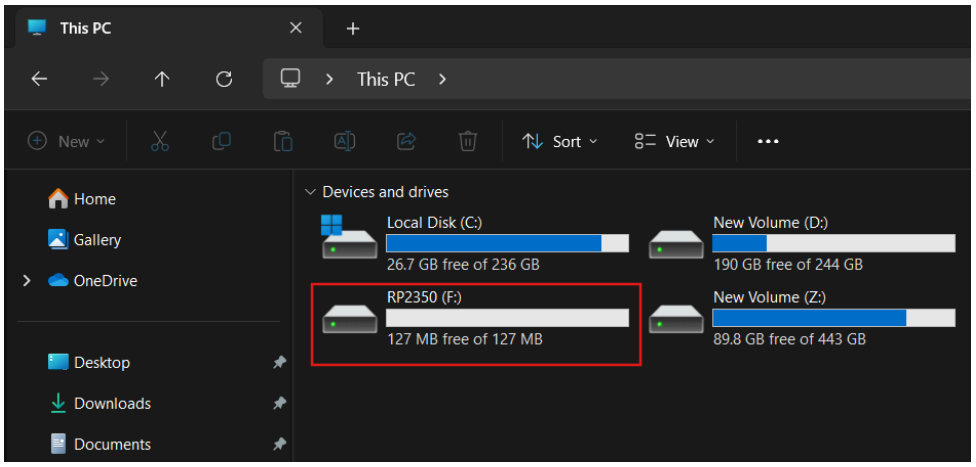
Setup Raspberry Pi Pico/PicoW/Pico 2

Step1: Download and install Thonny IDE from official link => https://thonny.org/

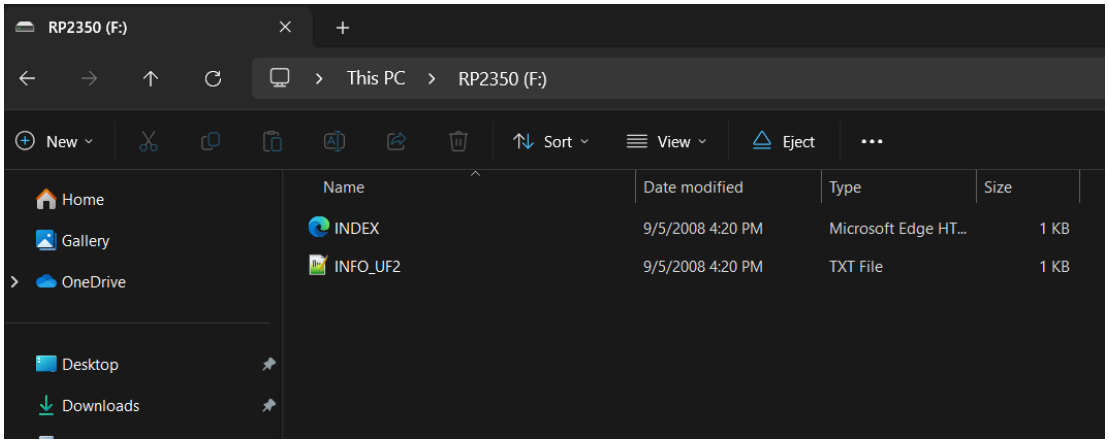


Step 2: MicroPython installation to Raspberry Pi Pico

If already installed, skip this step and proceed next else connect Pico with PC/laptop. Once you connect it will list as USB drive RP2040 (for Pico/PicoW) and RP2350 (Pico 2)

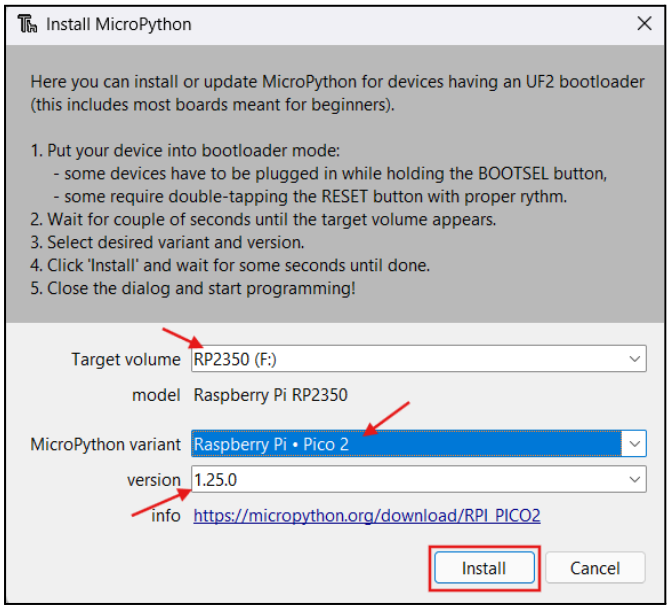


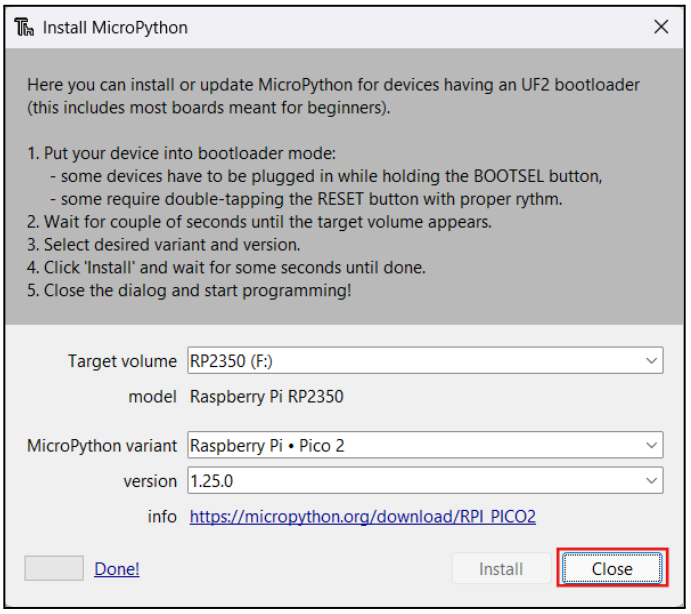
Default files inside,



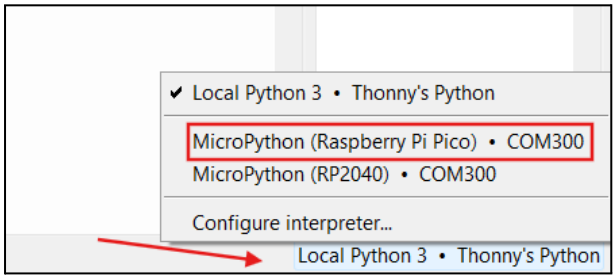
From Thonny IDE at right bottom corner click and install micropython,

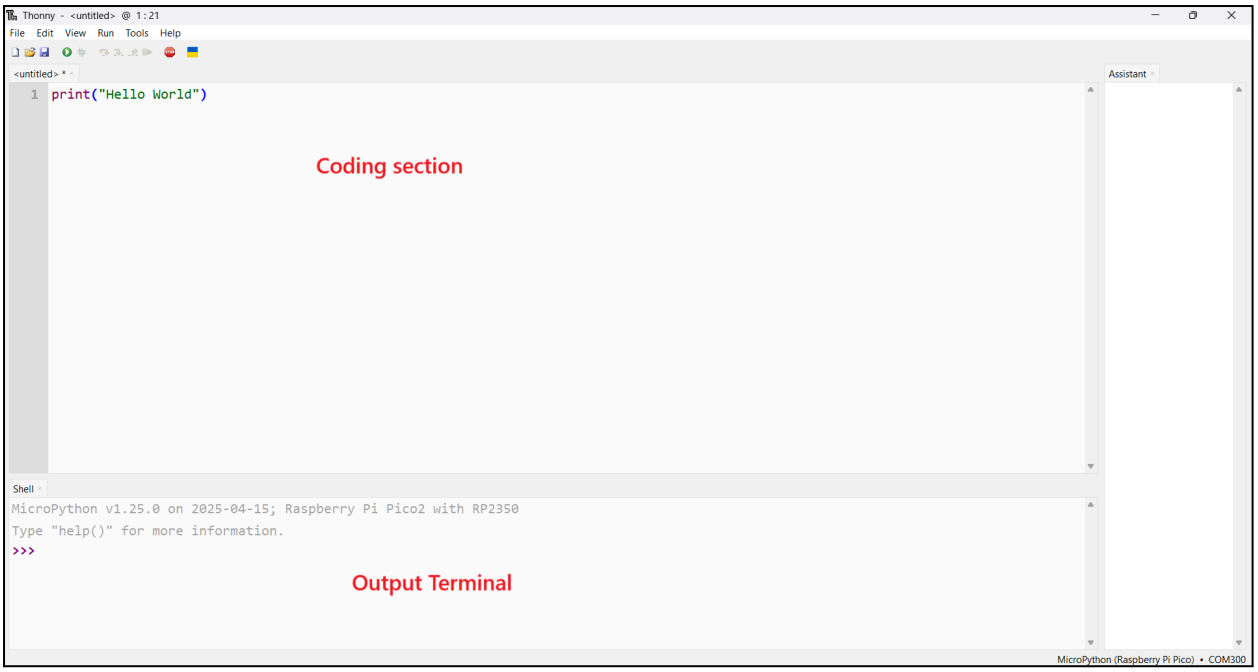
Depending on your pico variant select options and install,





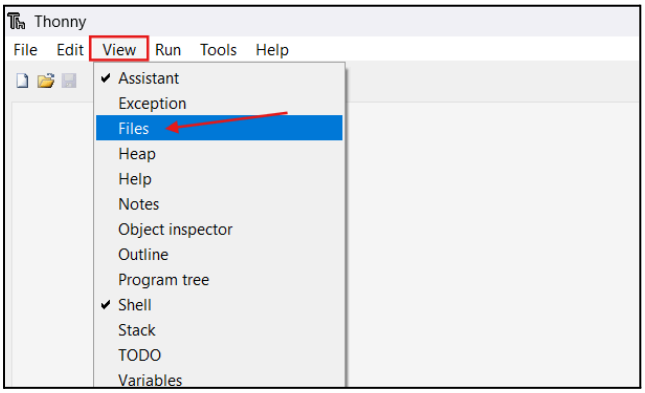
Now when you again click you will get MicroPython device with suitable COM port, select as shown

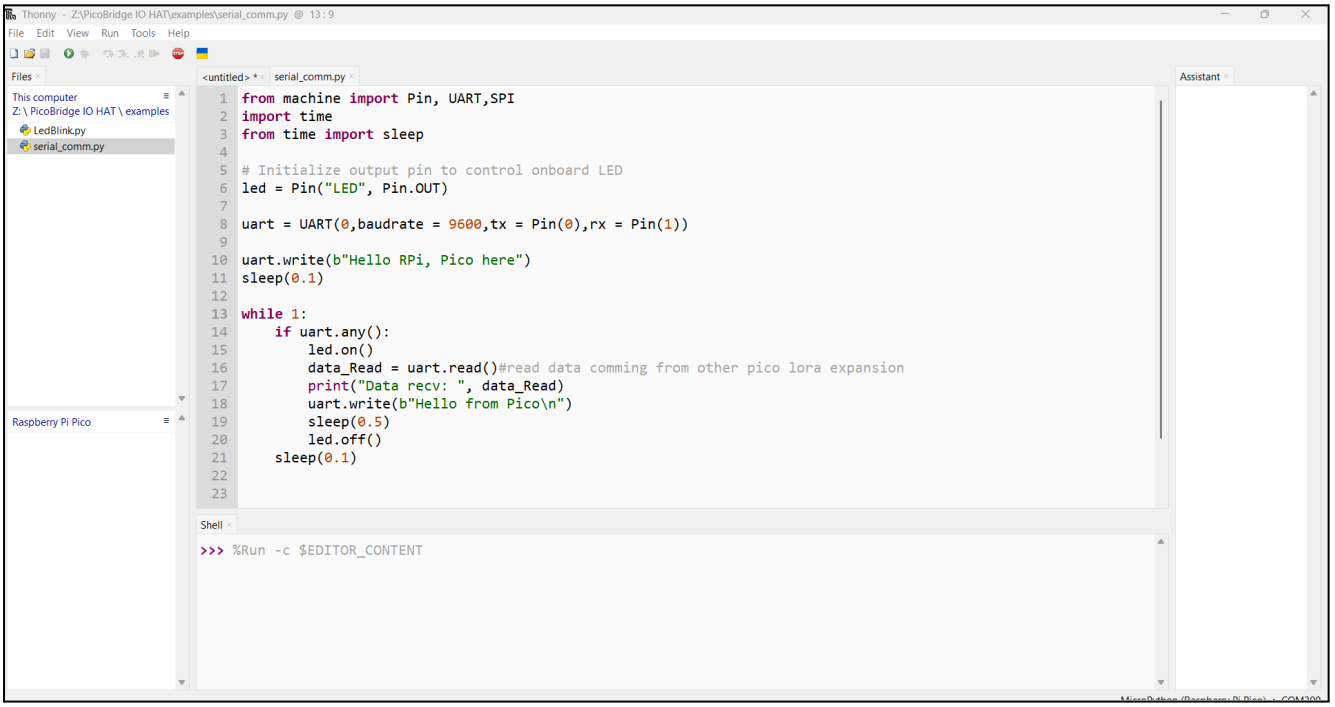




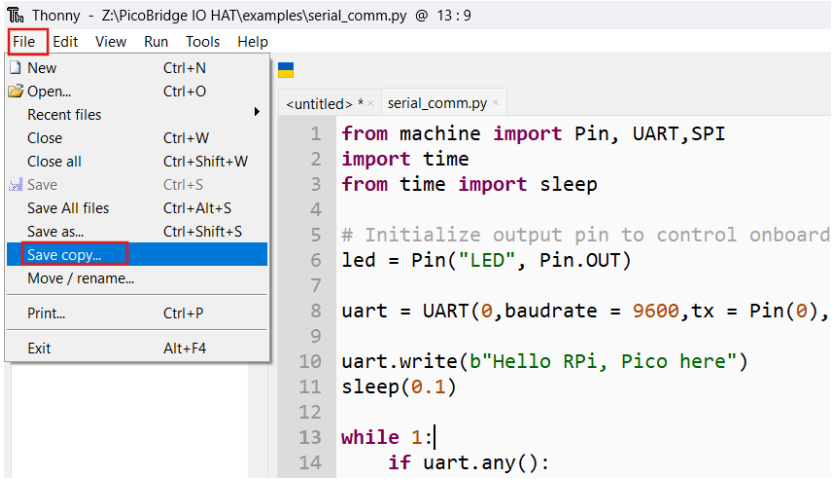
Step 3: Write Serial communication code with Raspberry Pi on Raspberry Pi Pico side

Select **View > Files** in Thonny IDE, better for folder file views of hardware and PC useful for easy transfer.





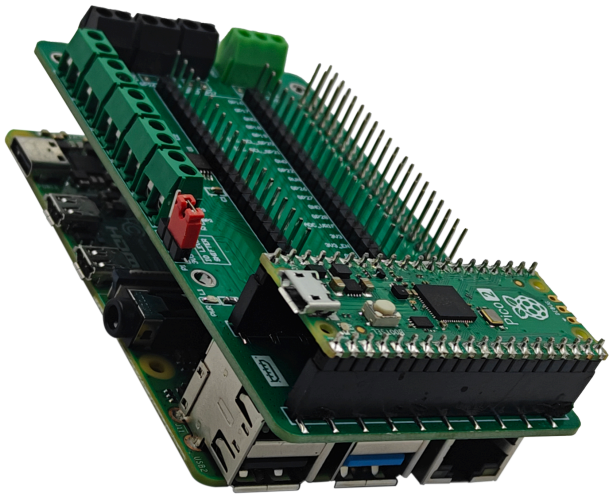
Write and save code to Raspberry Pi Pico as main.py for standalone execution,



Now, done on Pico side. Next will see coding on Raspberry Pi side and running demo.

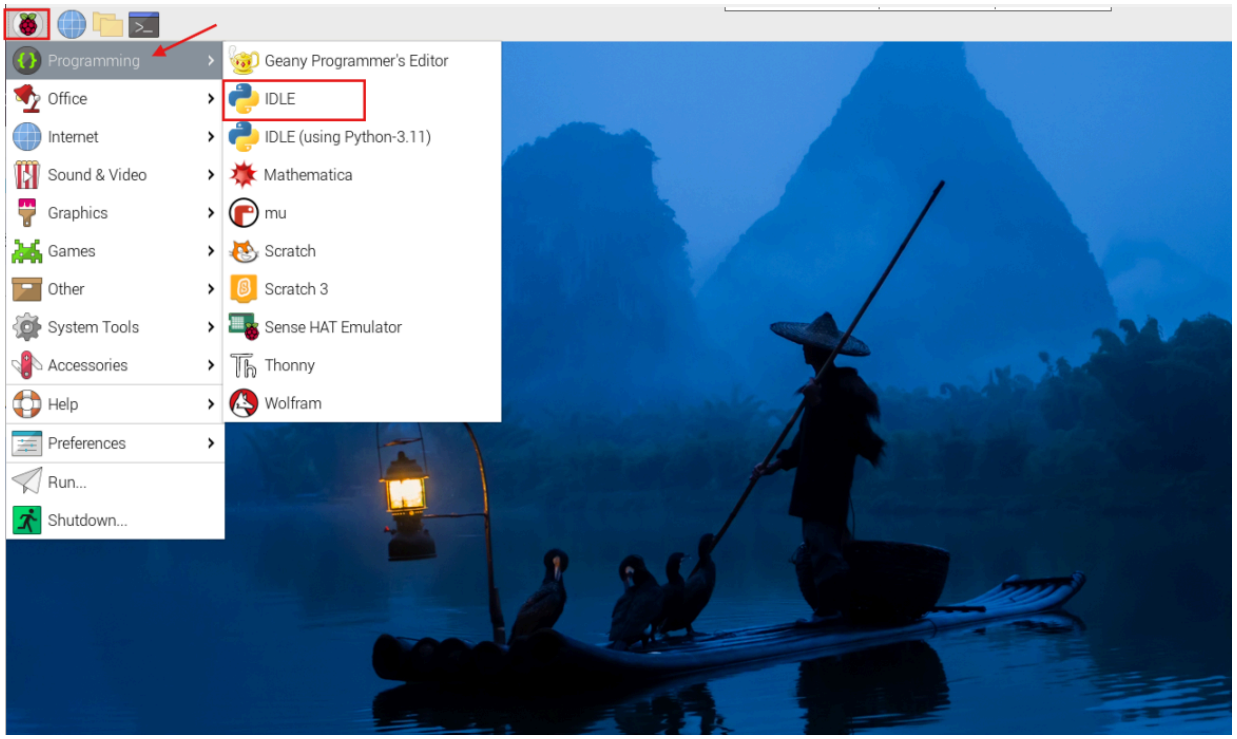
Serial Communication between Raspberry Pi and Pico

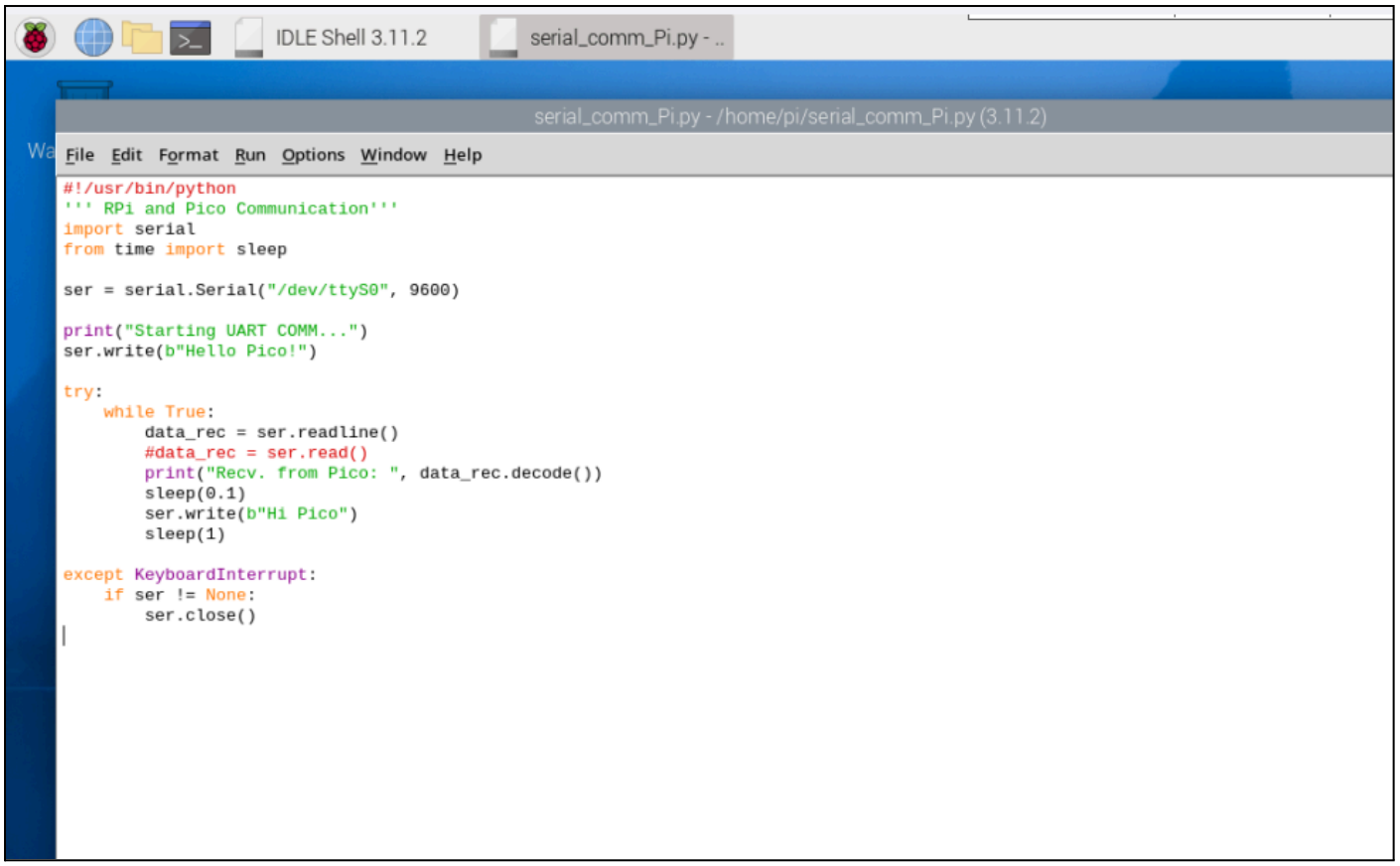
Connect PicoBridge HAT on your Raspberry Pi using 40 pin header option as shown,



Power your Raspberry Pi with PicoBridge HAT together.

Open any python IDE in raspberry Pi and write serial communication code on pi as shown below,





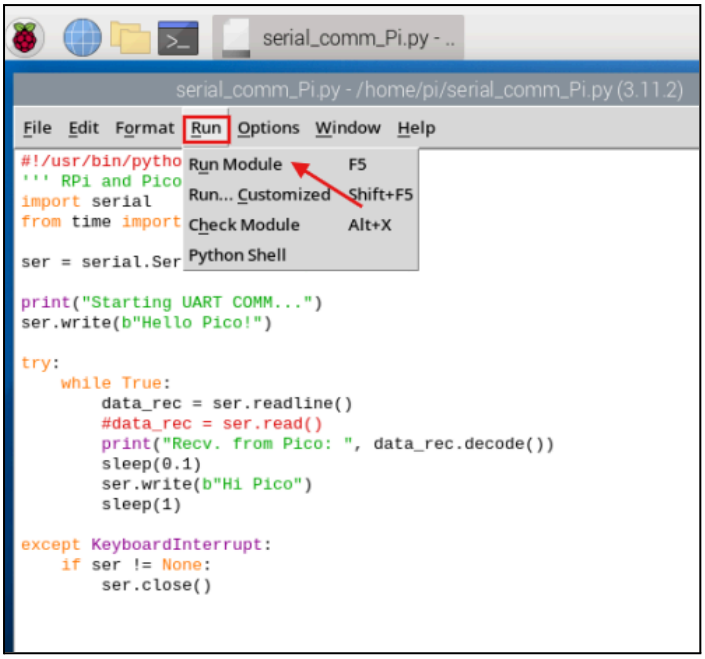
Pico UART0(GP0/TX and GP1/RX) and Raspberry Pi default UART used.

Note:

1) UART **'/dev/ttyS0'** for RPi 4 and previous version

2) UART **'/dev/ttyAMA0'** for RPi 5

Pico code already running with and Run Raspberry Pi code,



Raspberry Pi receive message from Pico and when Pico receive message from Pi it will blink onboard led,

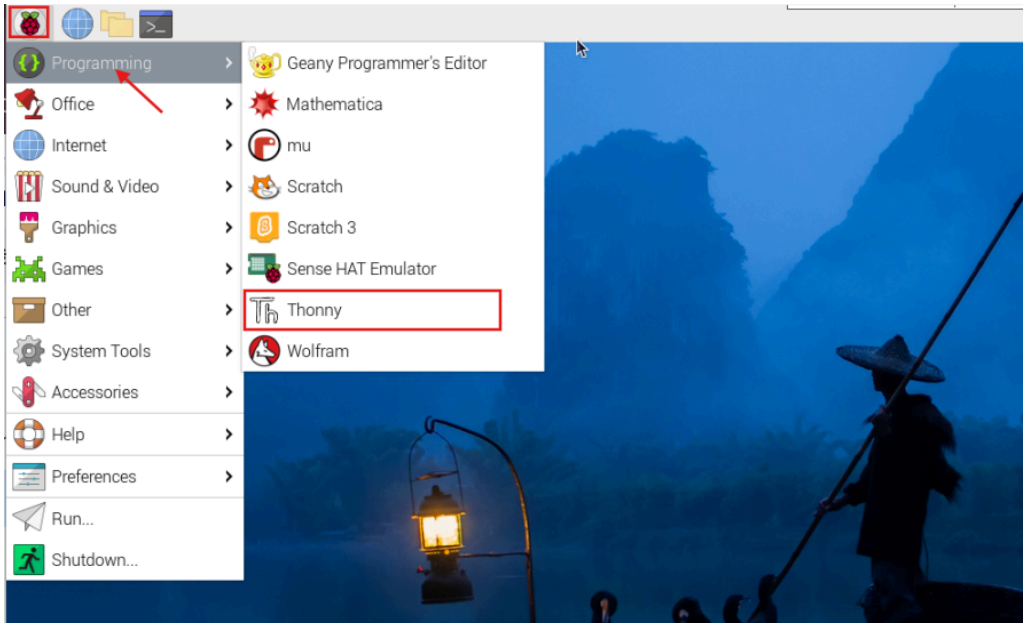
serial_comm_Pi.py - /home/pi/serial_comm_Pi.py (3.11.2)

```
#!/usr/bin/python
''' RPi and Pico Communication'''
import serial
from time import sleep

ser = serial.Serial("/dev/ttyS0", 9600)

print("Starting UART COMM...")
ser.write(b"Hello Pico!")

try:
    while True:
        data_rec = ser.readline()
        #data_rec = ser.read()
        print("Recv. from Pico: ", data_rec.decode())
        sleep(0.1)
        ser.write(b"Hi Pico")
        sleep(1)

except KeyboardInterrupt:
    if ser != None:
        ser.close()
```

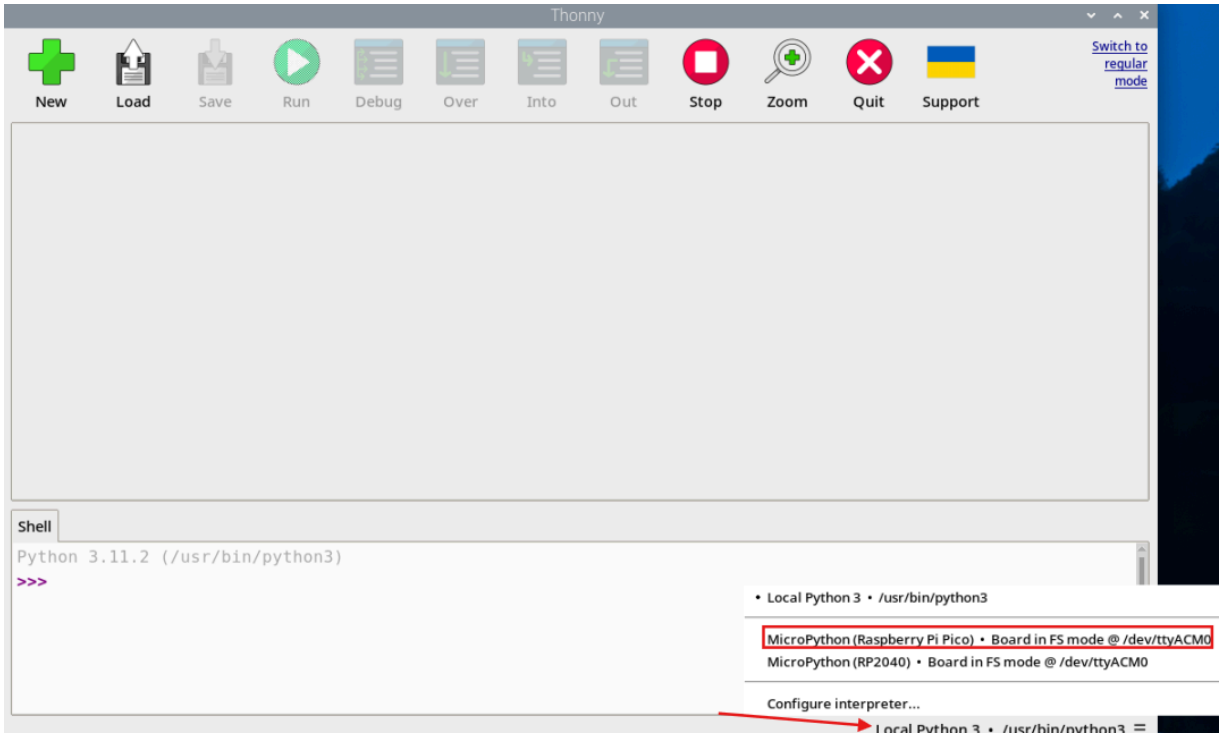
IDLE Shell 3.11.2

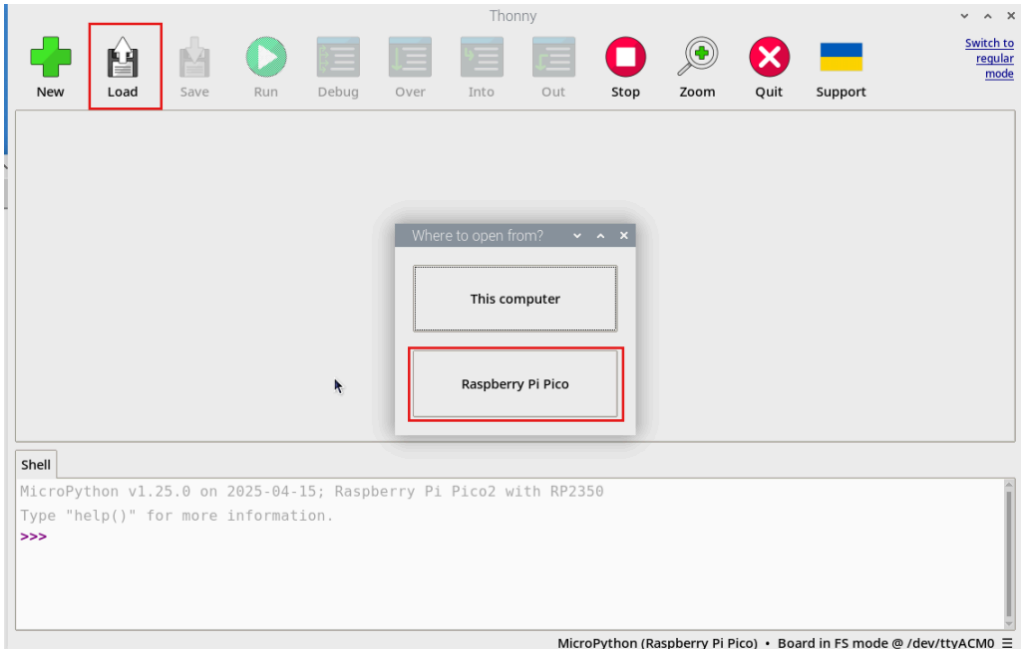
```
Python 3.11.2 (main, Apr 28 2025, 14:11:48) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====================== RESTART: /home/pi/serial_comm_Pi.py ======================
Starting UART COMM...
Recv. from Pico:  Hello from Pico

Recv. from Pico:  Hello from Pico

Recv. from Pico:  Hello from Pico

Recv. from Pico:  Hello from Pico

Recv. from Pico:  Hello from Pico

Recv. from Pico:  Hello from Pico

Recv. from Pico:  Hello from Pico

Recv. from Pico:  Hello from Pico

Recv. from Pico:  Hello from Pico

Recv. from Pico:  Hello from Pico

Recv. from Pico:  Hello from Pico
```

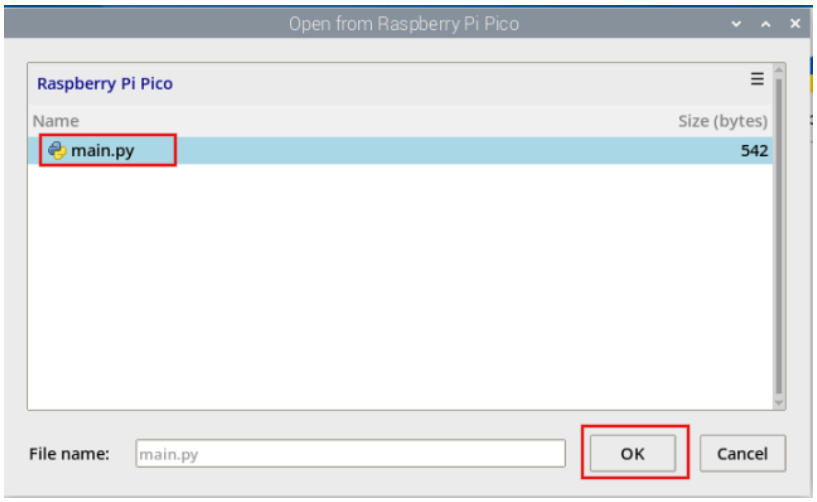
You can directly use the Raspberry Pi itself for previous Pico setup because the latest pi OS has Thonny as shown below and connect Pico with Raspberry pi using micro cable,



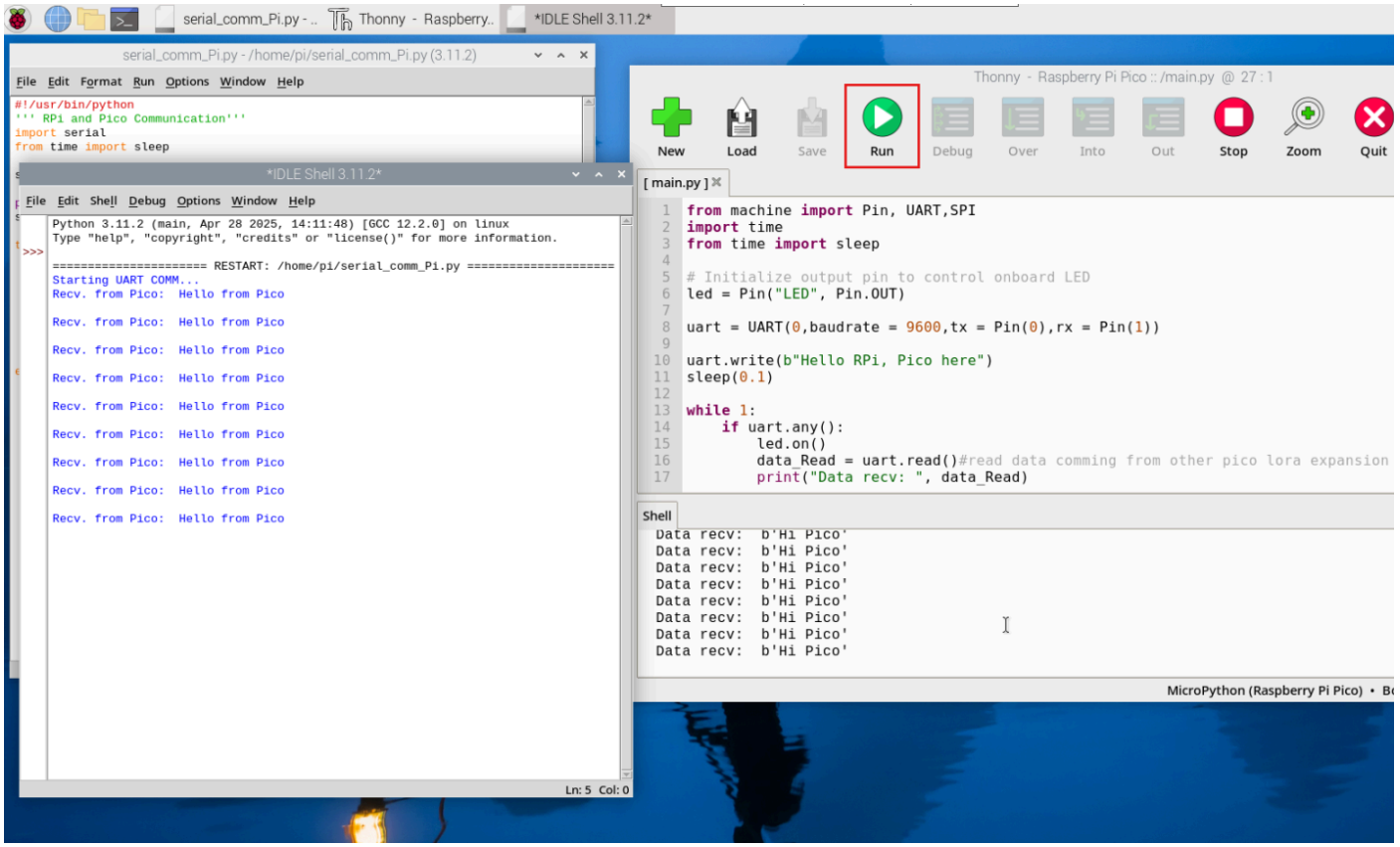
To visualize output of Pico,







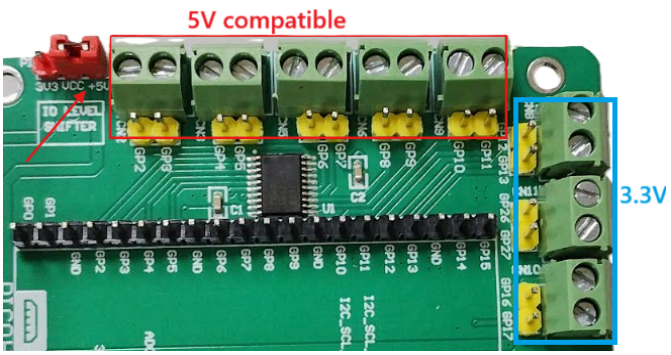
Once code open click green run play button,

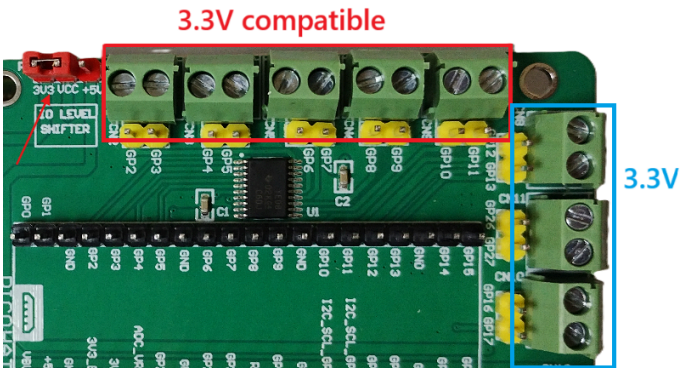


Level Shift Feature

PicoBridge has facility to level shift for some GPIOs Breakout to make it compatible and integrate easily with hardware sensors or MCU having logic 3.3V or 5V.

Just change the jumper at the respective voltage level to change the logic of GPIO as shown, but on the right side GPIOs always default 3.3V logic.





Resources:

- [Getting Started with Pico]
- [Getting Started with Raspberry Pi]