

02

FUNGI KIT

Programming Guide

Step 1: Opening Thonny

⇒ To program the microcontroller, you will be using a program called **Thonny**. Open the program by double clicking on the shortcut on the desktop or clicking on the taskbar icon to open it.

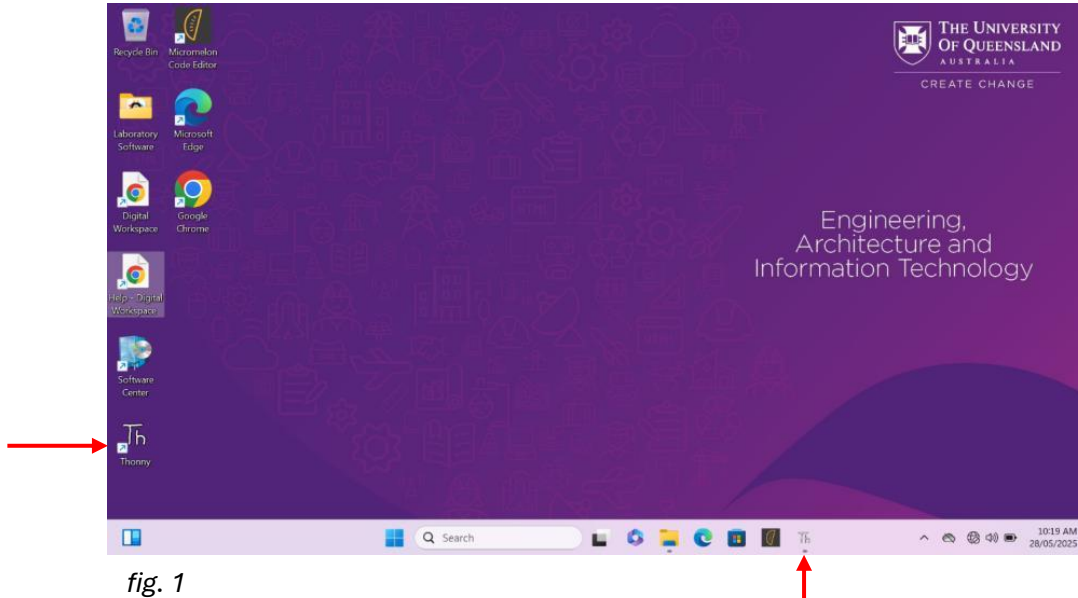


fig. 1

⇒ Once Thonny is open, go to the top toolbar and select **Run > Configure Interpreter...**

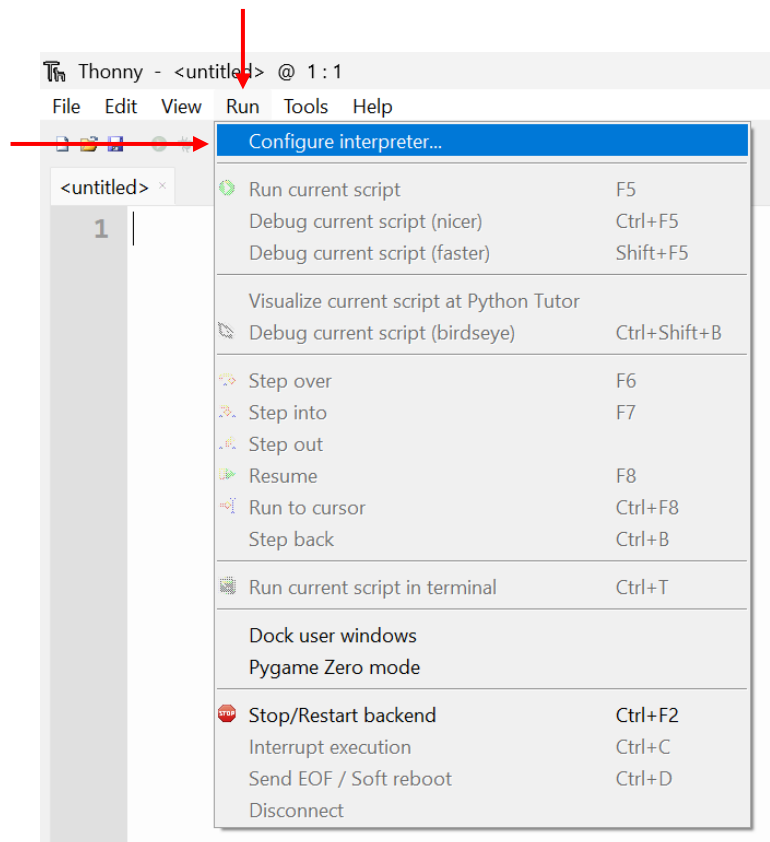


fig. 2

Step 2: Connecting to the Raspberry Pi Pico

🔗 In the second dropdown, select **Board CDC @ COM** (note: your number may be different).

⚠ If you do not see **Board CDC @ COM**, select **< Try to detect port automatically >**.

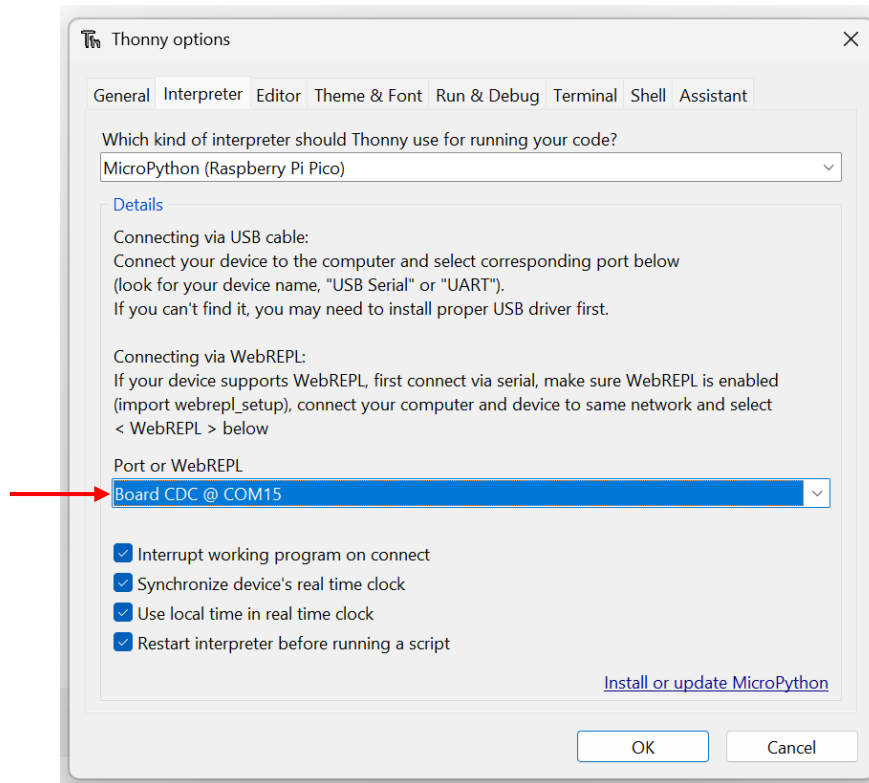


fig. 3

🔗 Click **OK**.

🔗 Then press the red **Stop** button (red stop sign) at the top left of the **Thonny** window. This will tell **Thonny** to find the Raspberry Pi Pico.

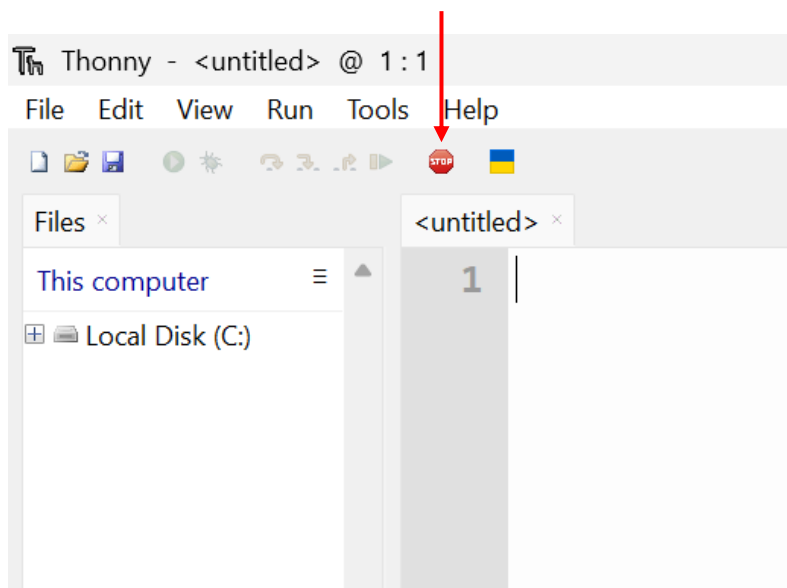


fig. 4

Step 3: Opening the Code on the Raspberry Pi Pico

☞ You should now see a section in the **Files** panel on the left named **Raspberry Pi Pico**.

☞ Double click on the **main.py** file to open it.

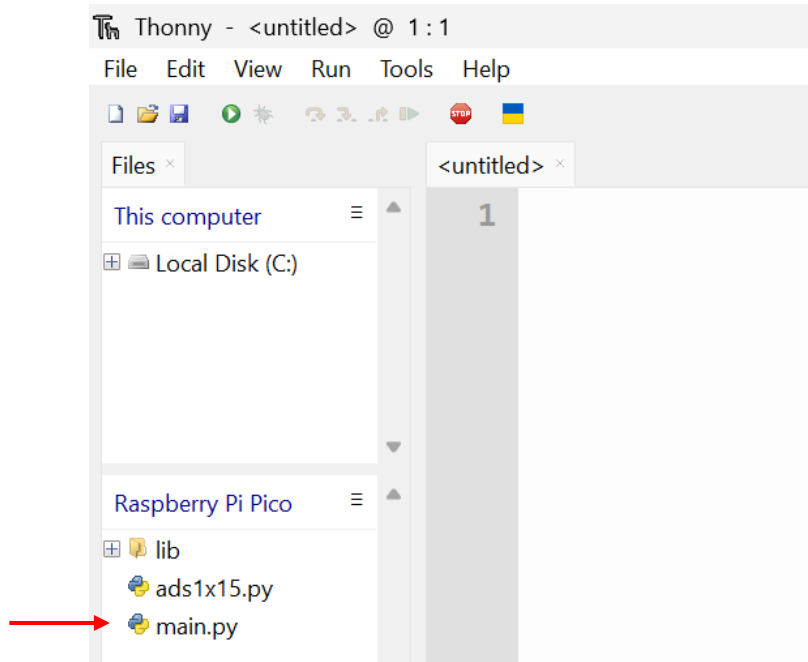


fig. 5

Step 4: Filling in the Code

☞ Now, look through the code and fill in the missing blanks (where you see '___')

☞ Read the comments above each for hints on what to input

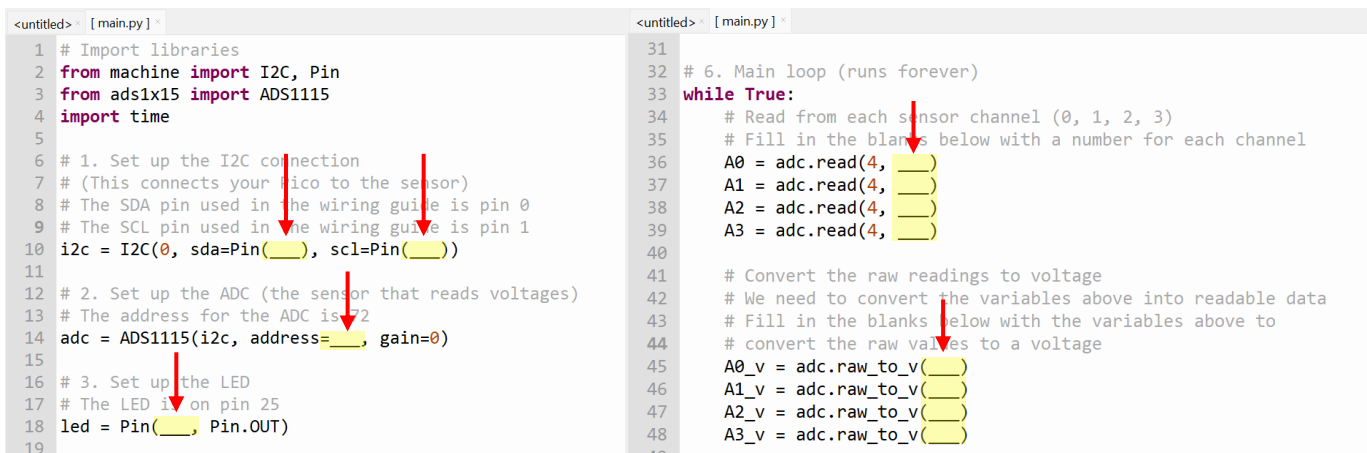


fig. 6

Step 5: Running and Testing the Program

🔗 Click the **Green Play Button** in the top left corner of **Thonny** to start the program.

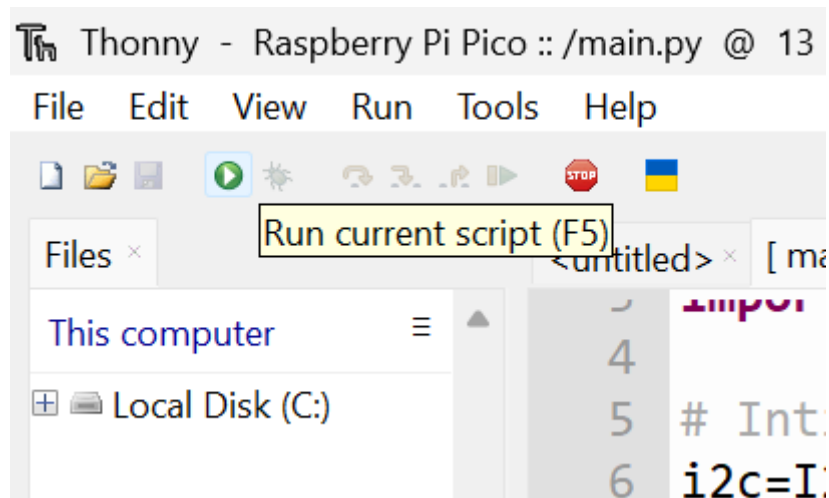


fig. 7

🔗 If successful, you will see:

- Readings print to the **Shell** at the bottom **Thonny**.
- A graph being plotted in real time.
- The LED blinking once every second
- A file created in the left **Files** tab called “**data.csv**”.

Try touching each wire from the ADC with your fingers and observe how it affects the data shown in the plot.

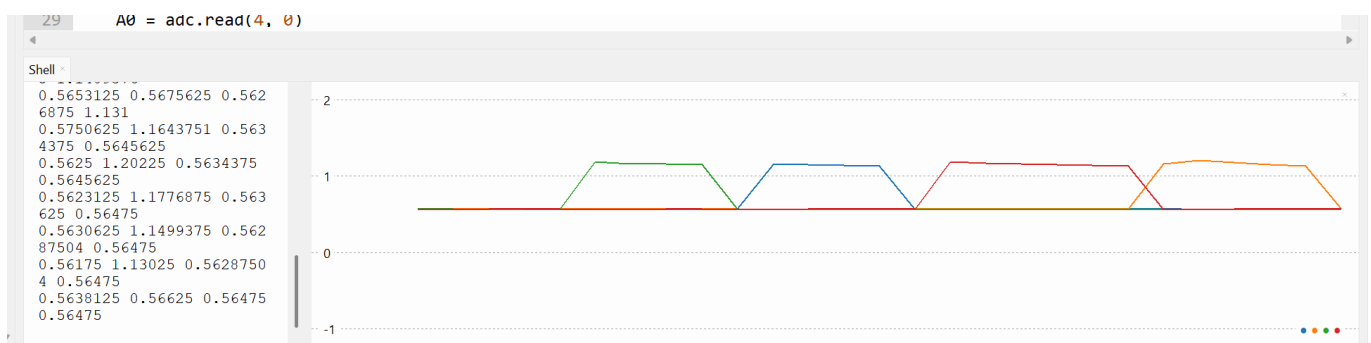
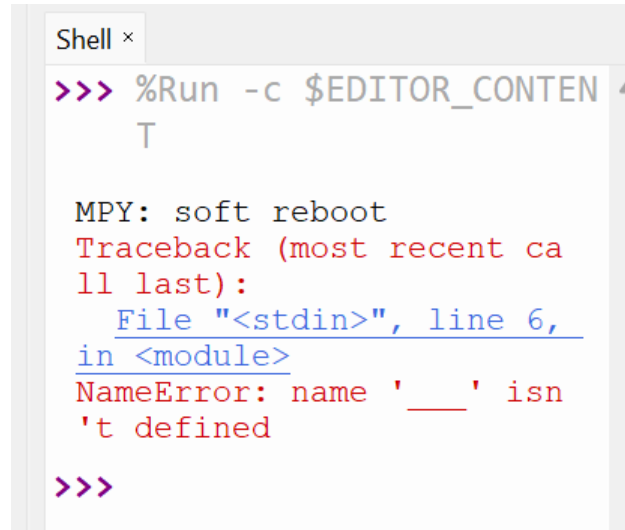


fig. 8

Troubleshooting

⚠ If you see the error below in your **Shell**, it means you missed one of the blanks to fill in.



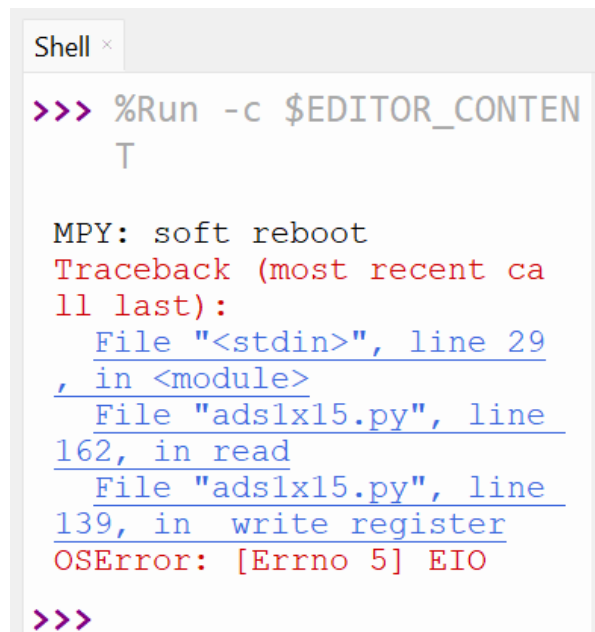
```
Shell x
>>> %Run -c $EDITOR_CONTENT
T

MPY: soft reboot
Traceback (most recent call last):
  File "<stdin>", line 6,
    in <module>
NameError: name '___' isn't defined

>>>
```

fig. 9

⚠ If you see this error below in your **Shell**, double check your wiring. This indicates the Raspberry Pi Pico cannot read from your ADC.



```
Shell x
>>> %Run -c $EDITOR_CONTENT
T

MPY: soft reboot
Traceback (most recent call last):
  File "<stdin>", line 29,
    in <module>
  File "adslx15.py", line 162, in read
  File "adslx15.py", line 139, in write_register
OSError: [Errno 5] EIO

>>>
```

fig. 10

⚠ If you are still having problems, ask for help from a tutor.