FUNGI KIT

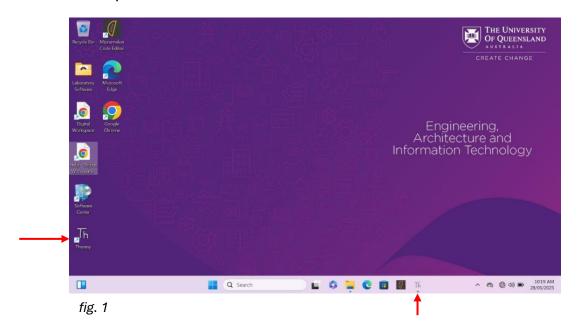
Programming Guide





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.



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

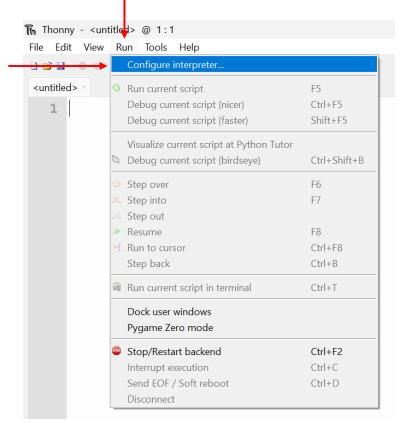


fig. 2



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 >.

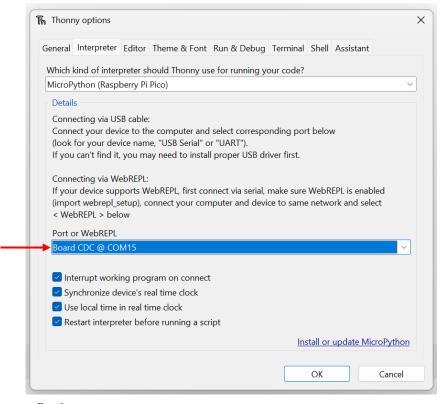


fig. 3

- ⇔ 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.

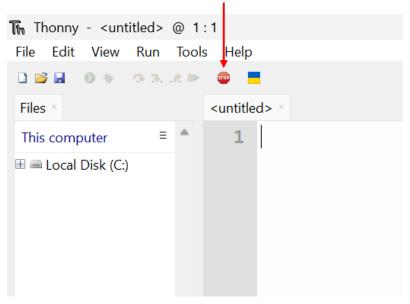
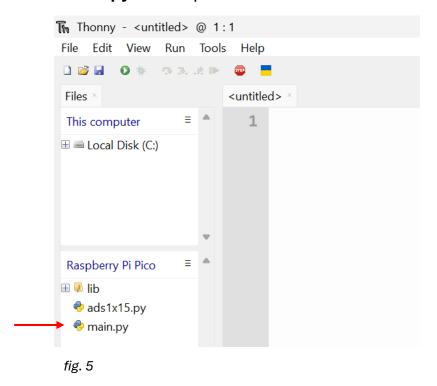


fig. 4



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.
- number 2 Double click on the main.py file to open it.



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

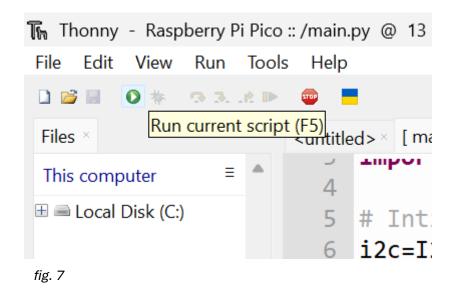
```
<untitled> × [ main.py ] ×
                                                                                      <untitled> × [ main.py ] ×
  1 # Import libraries
                                                                                        32 # 6. Main loop (runs forever)
  2 from machine import I2C, Pin
  3 from ads1x15 import ADS1115
                                                                                        33 while True:
                                                                                                 # Read from each sensor channel (0, 1, 2, 3)
# Fill in the blands below with a number for each channel
  4 import time
 # 1. Set up the I2C collection
# (This connects your lico to the sensor)
# The SDA pin used in the wiring guile is pin 0
# The SCL pin used in the wiring guile is pin 1
                                                                                                  A0 = adc.read(4, ____
                                                                                                  A1 = adc.read(4,
                                                                                        38
                                                                                                  A2 = adc.read(4,
                                                                                                  A3 = adc.read(4,
 10 i2c = I2C(0, sda=Pin(___), scl=Pin(___))
                                                                                        40
                                                                                                  # Convert the raw readings to voltage
                                                                                        41
 12 # 2. Set up the ADC (the sensor that reads voltages)
                                                                                                  # We need to convert the variables above into readable data
# Fill in the blanks below with the variables above to
# convert the raw valves to a voltage
                                                                                        42
     # The address for the ADC is
                                                                                        43
 14 adc = ADS1115(i2c, address=___, gain=0)
                                                                                        44
                                                                                        45
                                                                                                  A0_v = adc.raw_to_v(\underline{\hspace{1cm}})
16 # 3. Set up the LED
17 # The LED iv on pin 25
                                                                                        46
                                                                                                  A1_v = adc.raw_to_v(_
                                                                                       47
                                                                                                  A2_v = adc.raw_to_v(
 18 led = Pin(____, Pin.OUT)
                                                                                                  A3_v = adc.raw_to_v(
19
```

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.



🗳 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.

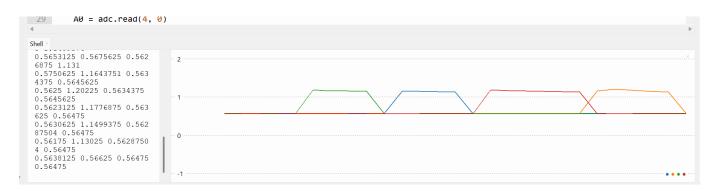


fig. 8



Troubleshooting

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

```
Shell ×

>>> %Run -c $EDITOR_CONTEN 
T

MPY: soft reboot
Traceback (most recent ca
11 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 ×

>>> %Run -c $EDITOR_CONTEN
   T

MPY: soft reboot
Traceback (most recent ca
11 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.