Introduction

This course focuses on the Raspberry Pi Pico W microcontroller as the main devices that you will use to control experiments, and take data. In the first few labs you will work through the book *Get started with MicroPython on Raspberry Pi Pico, The Official Raspberry Pi Pico Guide*. I will call this book *Getting Started*. It is available free as a PDF. I have made some edits to the book, so please download and use the PDF on the course web site.

You will actually use the version of the microcontroller that has WiFi capability. This is called the *Pico W*. I will abbreviate this as *PicoW*.

Today you will work through the first three chapters

1. Soldering pins to your PicoW and mounting it on a breadboard. Chapter 1.
2. Installing MicroPython (latest version) on your PicoW. Chapter 1.
3. Start programming your PicoW using MicroPython. Chapter 2.
4.

Activity 1 – Soldering the Headers to the PicoW

Note: you may have a PicoW with the headers already connected. Read the pages anyway.

- First read pp. 8 – 11, the one page insert, then pp. 12 – 16.
- When you are finished, show the instructor the PicoW soldered to pins. Figure 1-14 and the paragraph below shows how soldering can go wrong and how to correct it.
- Mount it in a breadboard with row 1 of the breadboard and pin 1 on the PicoW in the same row. The USB connector should be sticking out at the of the breadboard.

Activity 2 – Programming Your PicoW with MicroPython

- Download the latest version of µPy (MicroPython) from `https://micropython.org/download/RPI_PICO_W/`
  - ► It will be the first link under **Firmware** *Releases*. The name will be something like **`v1.23.0 (2024-06-02).uf2`** / [Release notes] (latest)
- Plug the PicoW into your laptop with a USB Micro cable and follow the directions on pg. 17.
- The next step, in Chapter 2, is installing the *Thonny* IDE (Integrated Development Environment.) It is a small stand-alone editor that allows you to read and write files to the PicoW.
- Write your first µPy program on the PicoW. It will print the message *Hello, World!*.

The Challenges

Throughout the book there are *Challenges* you have to complete.

Important Tip: You *always* want to save your most recent code *both* on the *MicroPython* device *and* on your laptop! So, here are my suggestions:

1. Make the file names descriptive, for example **`Hello.py`**, or **`Challenge1.py`**.
2. Save the first time on the PicoW.
3. Make a folder on your laptop for your code. Make a subfolder named **`Lab01`**. Save the files for Lab01 in this folder.
4. Make sure you save the file with the same name in your Lab01 folder, “This computer.”

Challenge 1 (pg. 29): Loop the Loop. Try printing the numbers from 1 to 9 in a 3x3 square using a loop inside a loop. If you want more than one print statement on the same line, add , **`end=’ ‘`** at the end of the print statement.

Challenge 2 (pg. 32): Add more. Ask the user for a number, then print out too high or too low, or end the game if they guess 5. Hint: To get an integer from the user you need to change the input line to

```
num_guessed = int(input(“Enter a number: “)
```

The `int` function converts the text to a number.

Challenge 3 Complex numbers. Yes, these little μPy microcontrollers know complex numbers! You enter a complex number, for example, as `2-3j`, which has a real part of `2` and a complex part of `-3`. To enter a pure complex number, put a `j` at the end, like `3.14j`. Write a program that inputs two complex numbers and prints their sum, product, and magnitude (try `print(abs(1+1j)`.)

Challenge 4 (pg. 43) Give the color codes for 100 Ω, 5.6 kΩ, 330 kΩ, and 1.5 MΩ resistors. Find the value and precision of the following resistors: brn-blk-yel, ora-ora-red-gold, gry-whi-brn-silver.