Introduction

About 15 years ago a new term was coined, *physical computing*. Let me give you the Wikipedia introduction:

> **Physical computing** involves interactive systems that can sense and respond to the world around them. While this definition is broad enough to encompass systems such as smart automotive traffic control systems or factory automation processes, it is not commonly used to describe them. In a broader sense, physical computing is a creative framework for understanding human beings' relationship to the digital world. In practical use, the term most often describes handmade art, design or DIY hobby projects that use sensors and microcontrollers to translate analog input to a software system, and/or control electro-mechanical devices such as motors, servos, lighting or other hardware.

The scientific terminology for this is *interfacing* or *instrumenting* – connecting a computing device to sensors (input) and controllers (output.) If you think about doing experiments you can break the process into changing parameters, and measuring what happens.

Today's lab starts you down the road to designing and building experiments.

Getting Started with Interfacing

Start working through Ch. 4 *Physical computing with Raspberry Pi Pico*. Here are some notes.

- **Importing**. Python, like most programming languages, is a general purpose language that people use for a wide variety of purposes. So the basic language leaves out commands that are specialized. Like most languages, there is a way to extend the languages for specific purposes. In python this is done by *importing modules*. All other languages have a way to do this. For example, in c++ libraries are add by the `include` statement.

  It is crucial for you to learn how importing works in python, and what modules are available for you to import. For example, to use common mathematical functions you import the `math` module. How do you find out what is in a module? After you import a module you can use the `help` function. For the `math` example, if you type `help(math)` you will get a list of what is in the math module: things like `pi`, `sqrt`, `sin`, `cos`, `exp`, and more. I use this all of the time You should learn to do it too!

  To get a list of all of the available builtin modules type `help('modules')`. You will learn how to use many of the modules in this course.
- **The LED**. The PicoW has a few difference from the Pico the book is written for. Instead of

  ```
  led_onboard = machine.Pin(25, machine.Pin.OUT)
  ```

  Use

  ```
  led_onboard = machine.Pin(LED, machine.Pin.OUT)
  ```
- **Challenge 1** (pg. 49) *Longer Light-Up*. Use the led_onboard.value() function to make the LED stay on longer, then shorter. Finally change your code so the LED is on half as long as it is off.
- **Challenge 2** (pg. 53) *Multiple LED's*. Wire up three external LED's of different colors. Program them to turn on and off successively and continue that pattern.
- **Challenge 3** (pg. 57) *Building It Up*. Work through the challenge, then wire an external button and two LED's, and program things so you can turn each LED or both at the same time, or turn both off by pushing the button.

Chapter 5 – Traffic Light Controller

Note: The topics of threads, introduced on page 62, is actually an advanced topic. You have to be careful using threads because getting them to communicate with each other is a trick and subtle issue.

- **Challenge 4** (pg. 66) Add a second button that operates like the first so a pedestrian on either side of the road can trigger the walk.

Chapter 6 – Reaction Game

This chapter introduces interrupt routines, a very important topic in interfacing a microcontroller to the real world. You will use interrupts in other projects later in the course.

Work through and build the Reaction Time games in this chapter. This is a challenging activity! Follow directions carefully.

- **Challenge 5** (pg. 74) Make the modifications suggested in the *Challenge* box.
- **Challenge 6** (pg. 79) Make the three person modification and print out how fast the first person's reaction time was.