CSCI 250 Python Computing: Building a Sensor System
TR 12:30-1:45 and 2:00-3:15 – MZ022 – Spring 2018
Lab 1: Blink with Extensions … Multiple LEDs and Button Press
Wendy Fisher

Corresponding Learning Outcome:

- Develop and run basic Python functions and programs in the Linux environment to collect data from sensors using the Raspberry Pi Hardware (e.g., optical, acoustic, acceleration, magnetic field).

  During this lesson, students will learn how to:
  - Assemble the ribbon cable, pi wedge, and understand the basics of using the breadboard
  - Interface with the General-Purpose Input/Output (GPIO) pin to control a LED
  - Update and run python code to blink the LED
  - Experiment with different resistance to control the brightness of the LED

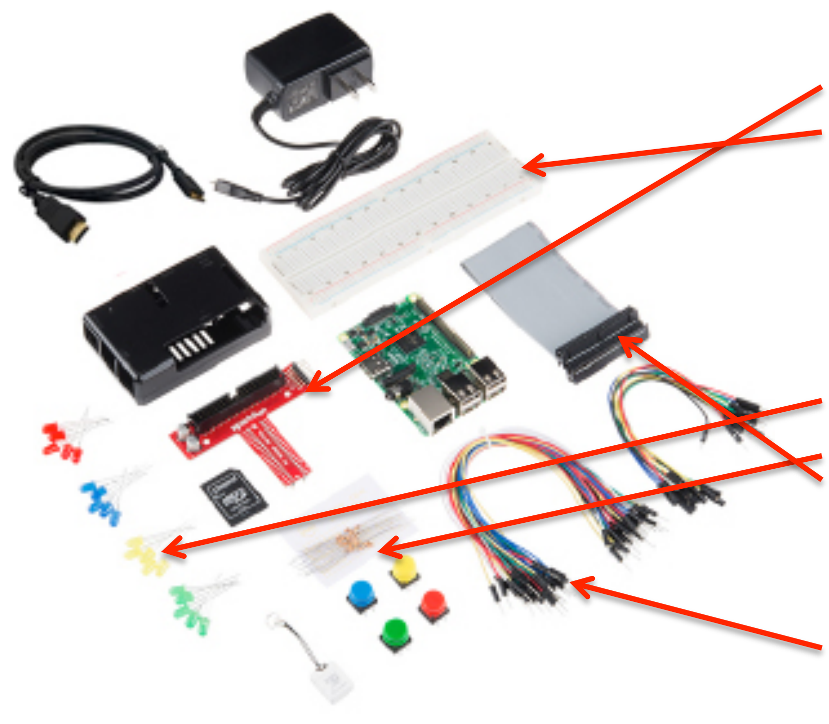
Reference the Sparkfun website Raspberry Pi 3: https://www.sparkfun.com/products/13825

We want to make sure the entire class is on the same page and has a solid foundation, so we will walk through this together. The next four labs will become progressively more difficult and require more independent programming and resourcefulness of looking at specification sheets for the sensors, etc.

Let's begin:

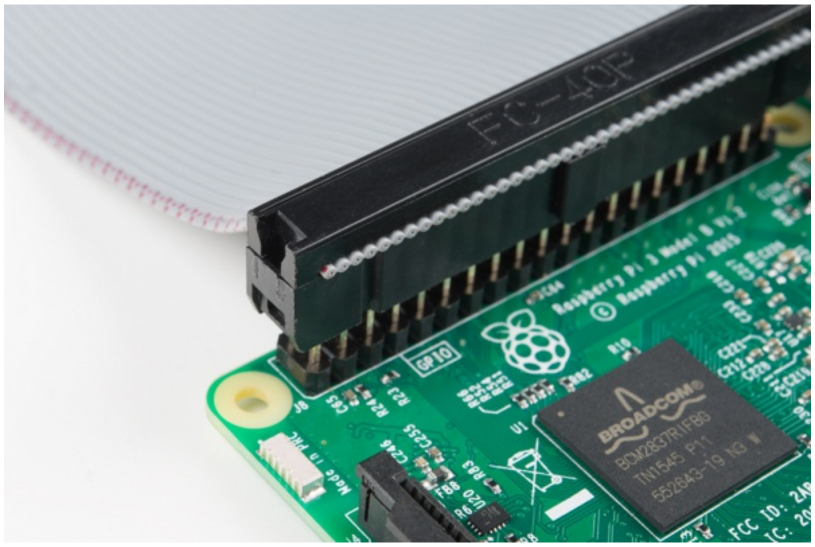
1. In addition to the equipment you need to get your Raspberry Pi up and running, collect the following equipment from your box. This is all you will need for the main part of this lesson (if you do the extra credit, you determine what is needed).

   Note: it is good practice to discharge yourself before handing electronics; you can easily do this by touching a grounded object.

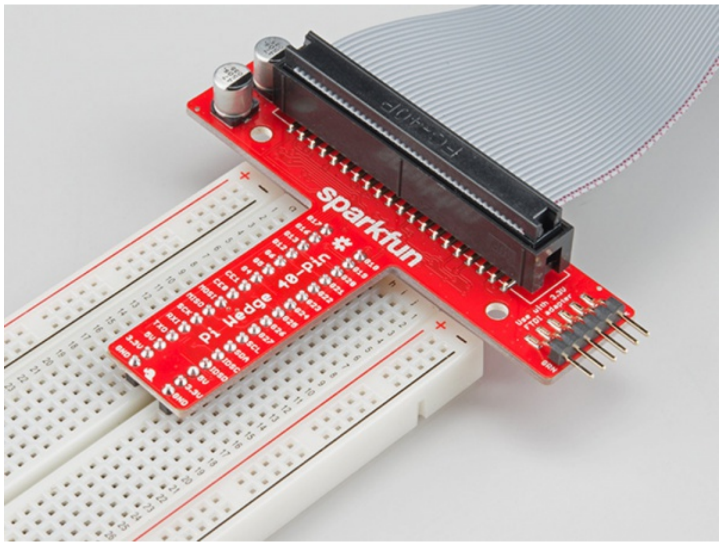


- Raspberry Pi 3
- SparkFun Pi Wedge (Preassembled)
- Breadboard - Full-Size (Bare)
- Pi Tin for the Raspberry Pi - Black
- 16GB microSD (Preloaded with OS)
- microSD USB Reader
- Red, Blue, Yellow, Green Buttons
- Red, Blue, Yellow, and Green LEDs
- Resistors 330 Ohm 1/6 Watt PTH
- GPIO Ribbon Cable - 40-pin, 6”
- Wall Adapter Power Supply
- Jumper Wires Premium 6" M/F – 10
- Jumper Wires Standard 7" M/M - 30
- HDMI Cable

2. Installation of the ribbon cable to the Raspberry Pi 40-pin GPIO. Holding the cable while lining up the red stripe (represents pin 1) on the cable with the end of the microSD card, push straight down until the cable is seated firmly.

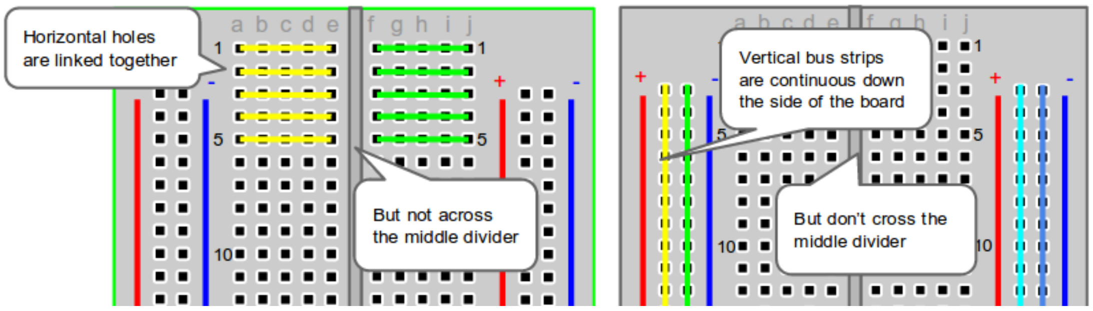


3. Attaching the ribbon cable to the Pi Wedge. Line up the red stripe (represents pin 1) on ribbon cable with the side of the Pi Wedge with the FTDI adapter (Note: we do not use the FTDI in our class). Then connect the Pi Wedge to the breadboard such that each side of the pins split the middle divider.



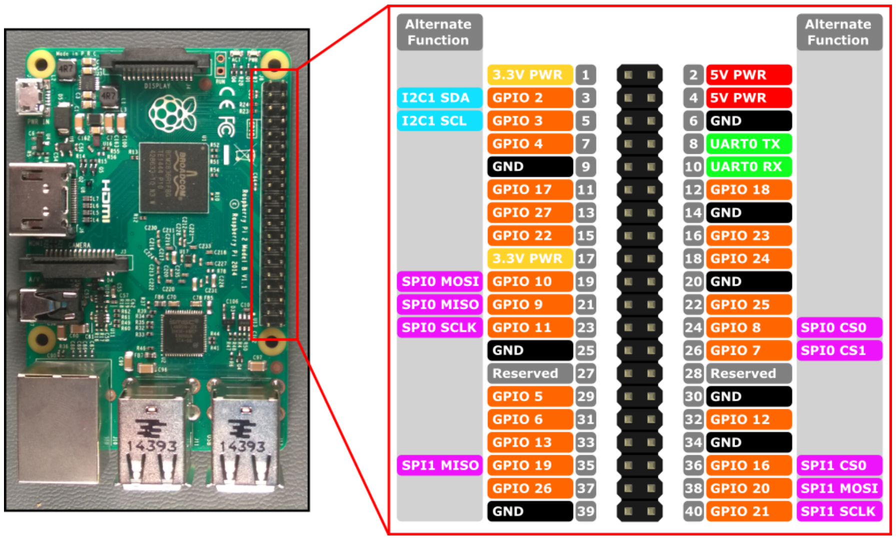
NOTE: Once the ribbon cable is installed, the top of the Raspberry Pi case will not fit back on, so your Raspberry Pi now needs to be stored carefully and recall if you continually remove the cable, you risk breaking it (yes, just like I have done).

4. Understanding the breadboard rails. Recall in class, we discussed the basics of how the breadboard works. You will need to keep this in mind as we start adding wires to our circuit.

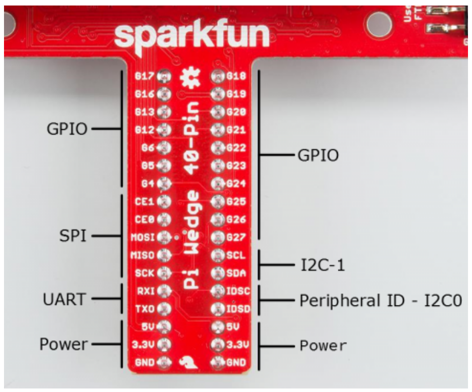

Horizontal holes are linked together
But not across the middle divider
Vertical bus strips are continuous down the side of the board
But don't cross the middle divider

Note: For more information, you can reference this great tutorial on the Sparkfun site: https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard

5. GPIO layout – the following image shows the mapping for the 40-pin GPIO header on the Raspberry Pi. Since we are using the Pi Wedge, the second image shows the new pin layout we will use. For the main part of this lesson, we are only working with GND and one of the GPIO pins (e.g., G25).



Alternate Function					Alternate Function
	3.3V PWR	1	2	5V PWR	
I2C1 SDA	GPIO 2	3	4	5V PWR	
I2C1 SCL	GPIO 3	5	6	GND	
	GPIO 4	7	8	UART0 TX	
	GND	9	10	UART0 RX	
	GPIO 17	11	12	GPIO 18	
	GPIO 27	13	14	GND	
	GPIO 22	15	16	GPIO 23	
	3.3V PWR	17	18	GPIO 24	
SPI0 MOSI	GPIO 10	19	20	GND	
SPI0 MISO	GPIO 9	21	22	GPIO 25	
SPI0 SCLK	GPIO 11	23	24	GPIO 8	SPI0 CS0
	GND	25	26	GPIO 7	SPI0 CS1
	Reserved	27	28	Reserved	
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
SPI1 MISO	GPIO 19	35	36	GPIO 16	SPI1 CS0
	GPIO 26	37	38	GPIO 20	SPI1 MOSI
	GND	39	40	GPIO 21	SPI1 SCLK



Note: We can also reference the Sparkfun site for more GPIO details: https://learn.sparkfun.com/tutorials/raspberry-gpio

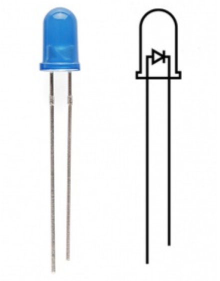
6. Wires – using the M/M Jumper Wires, it is time to connect the circuit, for example:

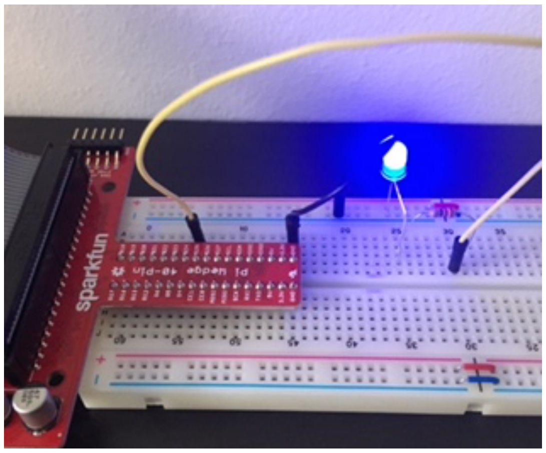
   Connect a black wire from the GND pin on the Pi Wedge to the negative (-) rail on the breadboard.

   Connect a yellow wire into the G25 pin on the Pi Wedge (we will connect the other end next).

   Note: You can use any color wires, the above is the colors I use for ground and just another color wire for the sensors (see photo in #7).

7. Time to add in the LED – recall, we discussed how current only flows one way through our LEDs that came in the kit … they will not work if they are in backwards (the positive leg is the longer one and the grounded leg is shorter). You will need to put the negative leg in the grounded rail (or add another black wire) … and then put the positive leg in a horizontal section of the breadboard (see photo). We will finish connecting next.





   Note: To learn more … and see how Light-Emitting Diodes (LEDs) are the "bacon of electronics":
   https://learn.sparkfun.com/tutorials/light-emitting-diodes-leds_ga=1.44566884.1582143202.1473712935

8. Before we put power to the LED, we will add resistance to control the flow of current to our LED. The Raspberry Pi kit has a bag of 330 Ohm resistors that will work just fine for our experiments. Add one to your circuit between the positive leg of the LED and where we will connect the other end of the yellow wire; just like the above photo.

   FOOD FOR THOUGHT: What do you think would happen with different resistance?
   *** To test your hypothesis, experiment with switching out the 330 Ohm resistor for the 1.0M Ohm resistor that came with your sensor kit after you get the initial setup working and then with none.*

330 Ohm Resistor

1.0M Ohm Resistor

   Note: If you want more information about the color bands on the resistors, Sparkfun has a tutorial:

   https://learn.sparkfun.com/tutorials/resistors/decoding-resistor-markings

Wow – way to go! Your first circuit (or at least the first one in this class!) – now let's code it up:
Again, we want to make sure the entire class is on the same page and has a solid foundation, so we will walk through this together. The next four labs will become progressively more difficult and require more independent programming. The following sample code will serve as a baseline and your task requires:

1. Open your IDE, create a new python file, type the following sample code, and save it as blink.py (or something you deem worthy). You should be able to run the code and have your LED blink, yea!!

```
#Author:
#Date:
#Description:

#import library to control the general purpose I/O pins and use short name
import RPi.GPIO as GPIO

#import the time library to use the sleep command
import time

#set the pin mode to use the numbers from the board
GPIO.setmode(GPIO.BCM)

#set the pin number 25 to be an output pin, direction
GPIO.setup(25, GPIO.OUT)

#blink - pull high, True, 3.3V OR low, False, 0 Volts
#NOTE: you can use control C to kill your program for or see slides for hints on a cleaner way to exit
while True:
    GPIO.output(25, True)
    time.sleep(1)
    GPIO.output(25, False)
    time.sleep(1)
```

2. Next: you need to update the code:
   a. Fill in the header information with your name, the date, and a description of the lab.
   b. Add two variables: one for the pin number and one for the number of seconds to sleep and replace the hard-coded numbers with your new variables in the code.
   c. Add a comment above your new variables as to why this is good programming practice/what advantage we gain when using variables instead of just the numbers.
   d. Run your code again to make sure it still blinks the LED.
   e. Experiment with different sleep times, different resistors, and re-run code, etc.

When completed, if you are still in class, let us know if you want to show us a demo of your circuit!

Submission Details – upload the following on Canvas:
(50 pts) Picture of your completed circuit with LED lit.
(50 pts) Python Code (blink.py) with changes defined in #2a-c above.

Creative Extensions … it is up to you to figure out how to add the following for possible extra credit, you can reference sparfun.com or raspberrypi.org and/or search on the Internet… have fun and be creative!
   a. (5 pts) Picture/code for: Adding another LED (or multiple).
   b. (10 pts) Picture/code for: Adding a button press to control one or more of the LEDs.