Mark as done

Introduction

The introduction for this first lab is long because I am introducing you to something I assume is completely new to you. In the future, they will be much shorter. We will be combining computer programming and microcontrollers to build small experimental systems to play with. You did some of this in Physics 1, and I think it is a fun way to learn about experimentation. We will be using what is called an Adafruit M4 Feather microcontroller to do lab work. It is similar to what you used in Physics 1.

1. You will not need to install any software as long as you use the lab computers for the microcontroller. The programming software is called "Mu".
2. The language used to program an M4 Feather is CircuitPython based on Python. I will provide you with working code and only ask you to make small modifications. The idea is to begin learning programming by seeing and editing examples.

Light Emitting Diodes (LEDs)

LEDs are small, powerful lights that are used in many different applications. We will work on blinking an LED, the "Hello World" of microcontrollers. That's right – it's as simple as turning a light on and off. It might not seem like much, but establishing this important baseline will give you a solid foundation as we work toward more complex experiments.

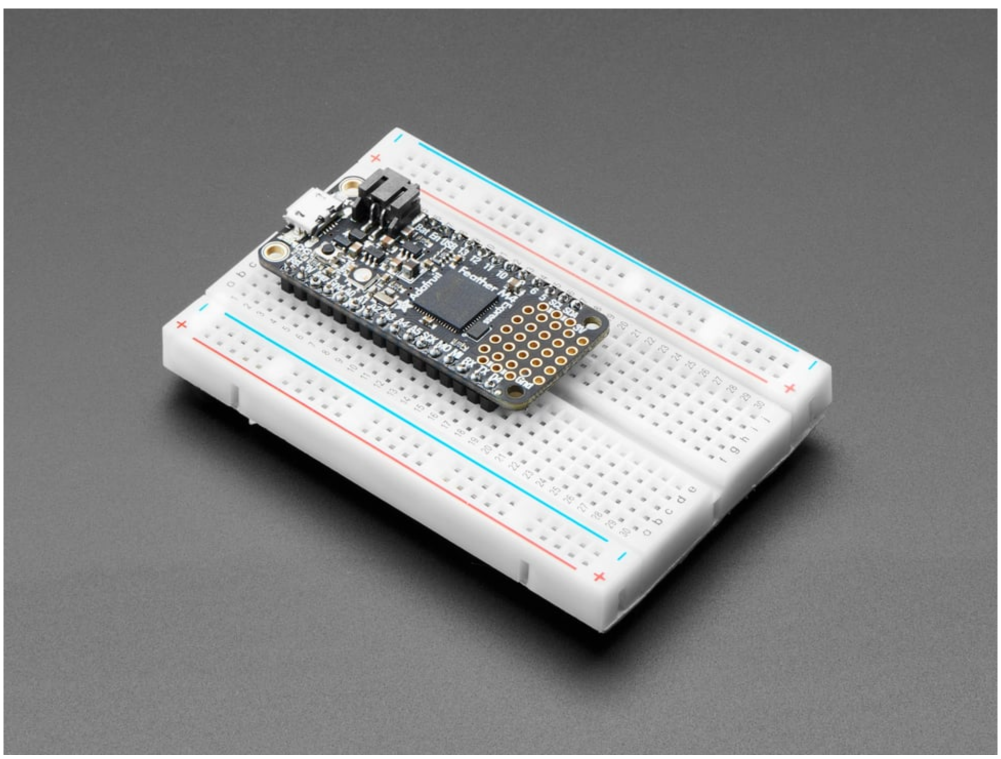
The basis of this lab is about *digital output* or ones and zeros as a "signal". As we progress through these lab activities, we will compare and contrast digital (ones and zeros) with analog (a continuous variable). We will also compare and contrast input vs. output (I/O).

Q: Why do you think powering an LED should be an output and not an input?

Q: Explain how powering an LED is digital ones and zeros in your own words.

After learning how to blink an LED, you will be asked to use the blinking light to communicate information using Morse Code. As part of this task, you will make investigations into what is called "timing resolution". In the context of LED Morse Code, this resolution is related to distinguishing blink lengths so that a message is effectively communicated.

Here is what your M4 Feather looks like.

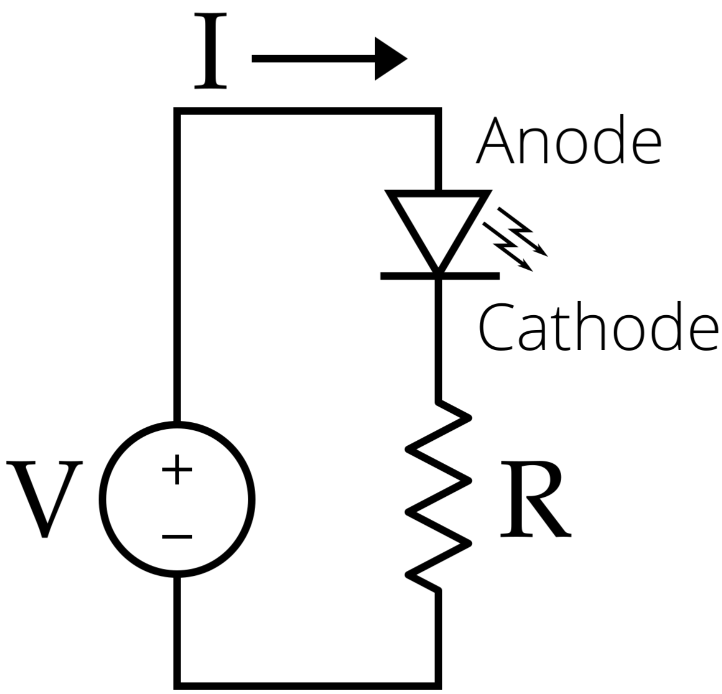


The different input and output (I/O) are labeled. For example, on the upper side of the image next to a coin, there is SDA, SCL, 5, 6, 9, 10, etc. The labels that are integers are digital I/O pins. On the lower side are labels D4, TX, RX, etc. The pins labeled A5, A4, A3, etc. are analog I/O. We will be examining the differences between digital vs. analog and input vs. output over the next few weeks.

Procedure

Circuit Setup

We need to wire a circuit to power the LED. Below are images of how you will wire the LED to the microcontroller digital pin 6 and what the circuit diagram looks like in the more general sense. We will be doing quite a bit of circuit analysis later in the term.



A few things to notice about the circuit. The LED has a longer and a shorter leg because LEDs are directional. The longer leg should be connected to the more positive voltage. There is a resistor in the circuit to keep the circuit from drawing more current than can be supplied by the USB port of a computer. We will be learning more about this through Ohm's Law, $V = IR$, later in the term.

Programming the microcontroller uses a language developed specifically for these microcontrollers. It is similar to the language Python . You will be provided programs for each lab and only asked to modify the programs to change their behavior. We will use a free CircuitPython code editor called Mu.

Virtually all microcontroller programs have the following structure.

- Include libraries (pre-made functions for doing things)
- Declare variables (stuff you want to store and reuse)
- Setup
  - Declare digital and analog inputs and outputs
- Loop function
  - The stuff that gets repeated

You can add comments that the program ignores if you put a `#` symbol before the comment. You can also use """Comment"" for comments that span more than a single line. Commenting is a good practice so that you know what you were thinking when you did wrote the code. I will show examples as we move along.

```
"""Name of the program"""
import math
x = 1 # variable
while True: #loop forever
     y = x + y # y is a counter
```

Below is the program written to begin the first lab.

```
"""Lab 1 - Blinky
based on CircuitPython Essentials Digital Out example"""

import time
import board
from digitalio import DigitalInOut, Direction, Pull

# LED setup.
led = DigitalInOut(board.D6)
led.direction = Direction.OUTPUT

while True:
    led.value = True
    # add a print statement so you can see when it is on and off in the Serial monitor
    time.sleep(1.0)
    led.value = False
    time.sleep(1.0)
```

The first couple of lines are comments. I recommend you add comments to the other lines to get practice. I will state what each line does. Your job is to comment your code.

1. Comment to name the program
2. Comment to name the program
3. Import a library for timing
4. Import a library to talk to the microcontroller
5. Import a library for digital I/O
6. Comment
7. Define the LED as a variable to store digital I/O connected to pin 6
8. Define the LED pin as an output
9. Begin an infinite loop
    - turn the LED on
    - comment - a print statement looks like
      ```
      print("Hello World")
      ```
    - wait for 1 second
    - turn the LED off
    - wait for one second
10. **What unit of time is the delay function using?**

Assignment

Save this program to the Feather as "code.py" and verify that it functions. The microcontroller should appear on the computer as a drive called CIRCUITPY.

How can you test it to make sure that changes to the program make changes to the behavior once the changed program is saved?

Recently three sailors were rescued from a South Asian island after SOS was spotted in the sand.

If it were night, they may have wanted to use lights. Program your LED to write SOS using morse code. The Morse Code table below shows how to encode based on dots (short signal) and dashes (long signal). The spacing between dots, dashes, letters, and words is important as noted on the table.

image source: https://en.wikipedia.org/wiki/Morse_code

Write a procedure for creating this message in your notebook and answer the following questions.

Questions to address:

What are the shortest dash/dot you can use and still have your message read? Try coding on your LED the signal SOS and decrease the dot/dash length of time until you cannot keep up with reading the message. You may want to create variables dot and dash so you only need to change dot, and dash is automatically set as 3 * dot.

How long between each SOS message should you wait so that it is obvious a new message is starting? Be sure this is implemented into your code. Ask your professor or TA to read the message.

Notice not all letters are three units. How does this affect your construction of a message? What is the international Morse Code convention? Comment on why this convention might have been created.

Send a sentence-long (3-5 word) message and make sure your instructor/TA recognize the message?

When you have finished, be sure your lab notebook is completely updated with drawings, code, answers to questions, and any other information you would need to repeat this experiment without my instructions. Please include copies of your code for SOS and your sentence. These should be commented so any other programmer knows what they do.

Last modified: Wednesday, August 16, 2023, 10:47 AM