IDD-Fa18-Lab1: Blink!

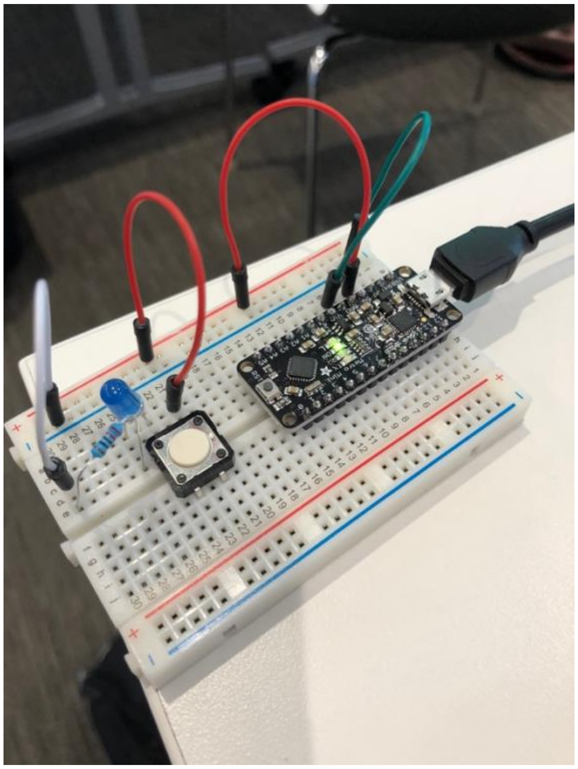
A lab report by Jeremy D. Walker

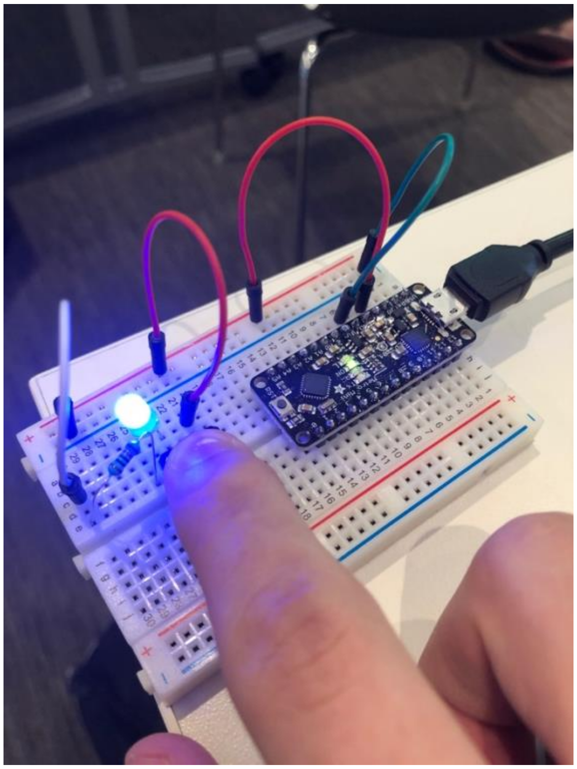
> Include your responses to the bold questions on your own fork of the lab activities. Include snippets of code that explain what you did. Deliverables are due next Tuesday. Post your lab reports as `README.md` pages on your GitHub, and post a link to that on your main class hub page.

We've copied the questions from the lab here. Answer them below!

Part A. Set Up a Breadboard

[insert a photo of your breadboard setup here]





Part B. Manually Blink a LED

a. What color stripes are on a 220 Ohm resistor?

For 5-band resistor; bands are (from left to right):
Red, Red, Black, Black, Brown

b. What do you have to do to light your LED?

Pressing the button with the circuit configuration shown connects power across the button from the power rail to the LED

Part C. Blink a LED using Arduino

1. Blink the on-board LED

a. What line(s) of code do you need to change to make the LED blink (like, at all)?

The only lines that need to be changed are if you want to use a GPIO other than digital output pin 13, otherwise the sample program will blink the LED by connecting the circuit as shown. (This configuration also allows the user to "overwrite" the program by manually pressing the button to turn on the LED as before)

b. What line(s) of code do you need to change to change the rate of blinking?

Lines 34 and 36 (the two delay lines) correspond to the duration of time (in ms) for the LED to be turned on and off

c. What circuit element would you want to add to protect the board and external LED?

You could add a diode to aid in circuit protection

d. At what delay can you no longer *perceive* the LED blinking? How can you prove to yourself that it is, in fact, still blinking?

At around 10ms, I can no longer perceive the LED blinking. You could insert a counter to determine the number of times that the LED turns on (or off).

e. Modify the code to make your LED blink your way. Save your new blink code to your lab 1 repository, with a link on the README.md.

I made the LED do one "fast blink" and one "slow blink" with the following code:

```
      void loop() {
      digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the
      voltage level)
      delay(2000);                       // wait for a second
      digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making
the voltage LOW
      delay(2000);                       // wait for a second

    digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the
voltage level)
      delay(500);                       // wait for a second
      digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the
voltage LOW
      delay(500);                       // wait for a second
}
```

2. Blink your LED

Make a video of your LED blinking, and add it to your lab submission.

https://youtu.be/U2TD6qERsfI

Custom blinking:

https://youtu.be/5X9OhGHZDks

Part D. Manually fade an LED

a. Are you able to get the LED to glow the whole turning range of the potentiometer? Why or why not?

No, the LED only glows during transitions

Part E. Fade an LED using Arduino

a. What do you have to modify to make the code control the circuit you've built on your breadboard?

The pin has to be changed from GPIO 9 to GPIO 11

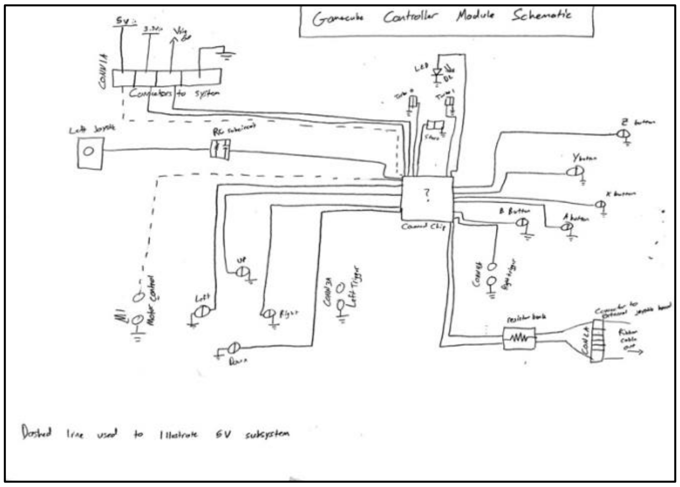
https://youtu.be/efsDOxT8WgE

b. What is analogWrite()? How is that different than digitalWrite()?

Analog write uses PWM to emulate an analog signal vs digital write which just assigns a value of high or low to the pin

Part F. FRANKENLIGHT!!!

1. Take apart your electronic device, and draw a schematic of what is inside.



a. Is there computation in your device? Where is it? What do you think is happening inside the "computer?"

I think there is a basic chip that determines which buttons were pressed and communicates this over a 3.3v logical wire

b. Are there sensors on your device? How do they work? How is the sensed information conveyed to other portions of the device?

There are numerous buttons

c. How is the device powered? Is there any transformation or regulation of the power? How is that done? What voltages are used throughout the system?

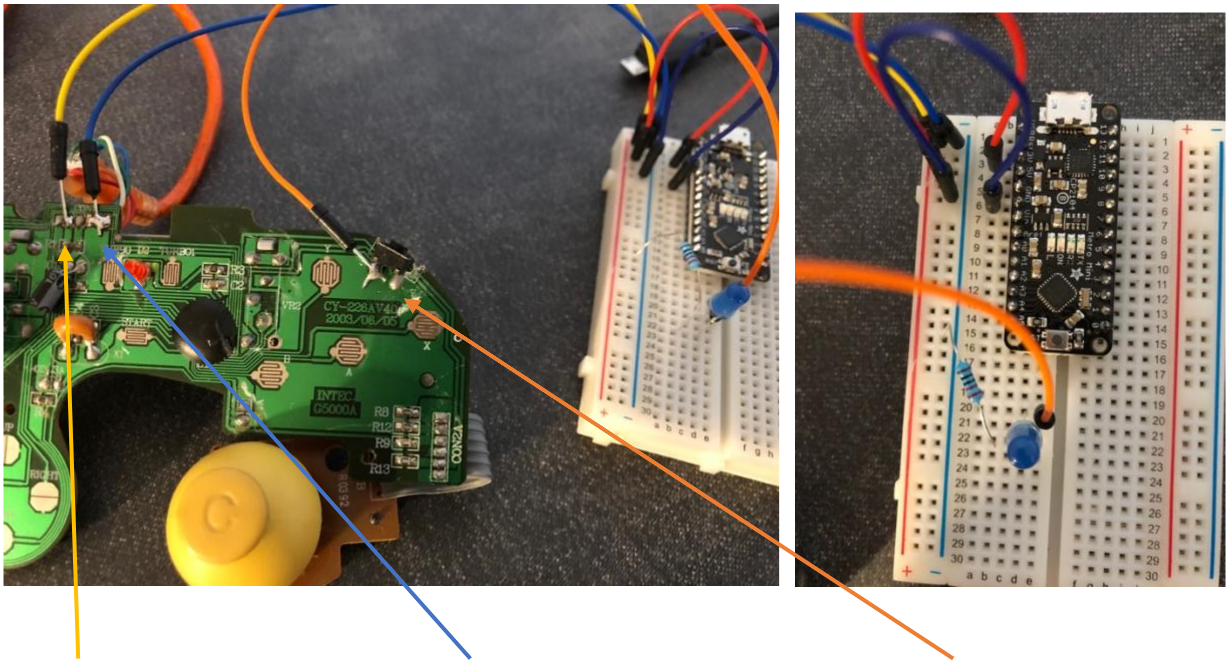
It appears there is both a 5v and 3.3v power source. 3.3 is used for data transmission and main power while 5v powers a “rumble motor”

d. Is information stored in your device? Where? How?

I don’t think there is any on board memory as all data is transmitted to the main system

2. Using your schematic, figure out where a good point would be to hijack your device and implant an LED.

I decided to power the controller via the Arduino board we have and connect an LED with pull down resistor to the z-button on the top right of the board. This will allow the LED to pulse when the button is pressed simulating the data that would be transferred if the controller were connected as normal. The LED was a little dim because 3.3 is used for powering the button as opposed to the 5v used above in this lab, but otherwise the circuit worked.



3.3 Vin

GND

Z-Button Signal

3. Build your light!

https://youtu.be/60oWOBbOzTI