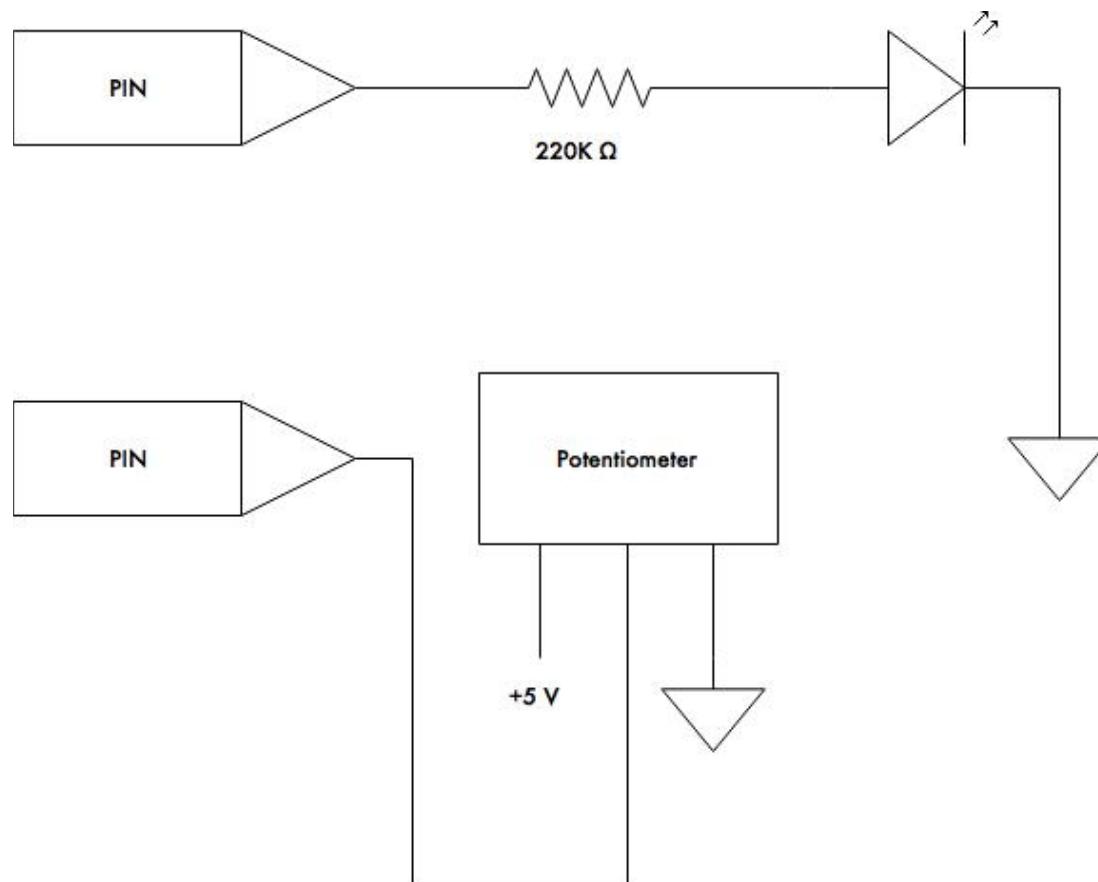


E2: Integrated Builds

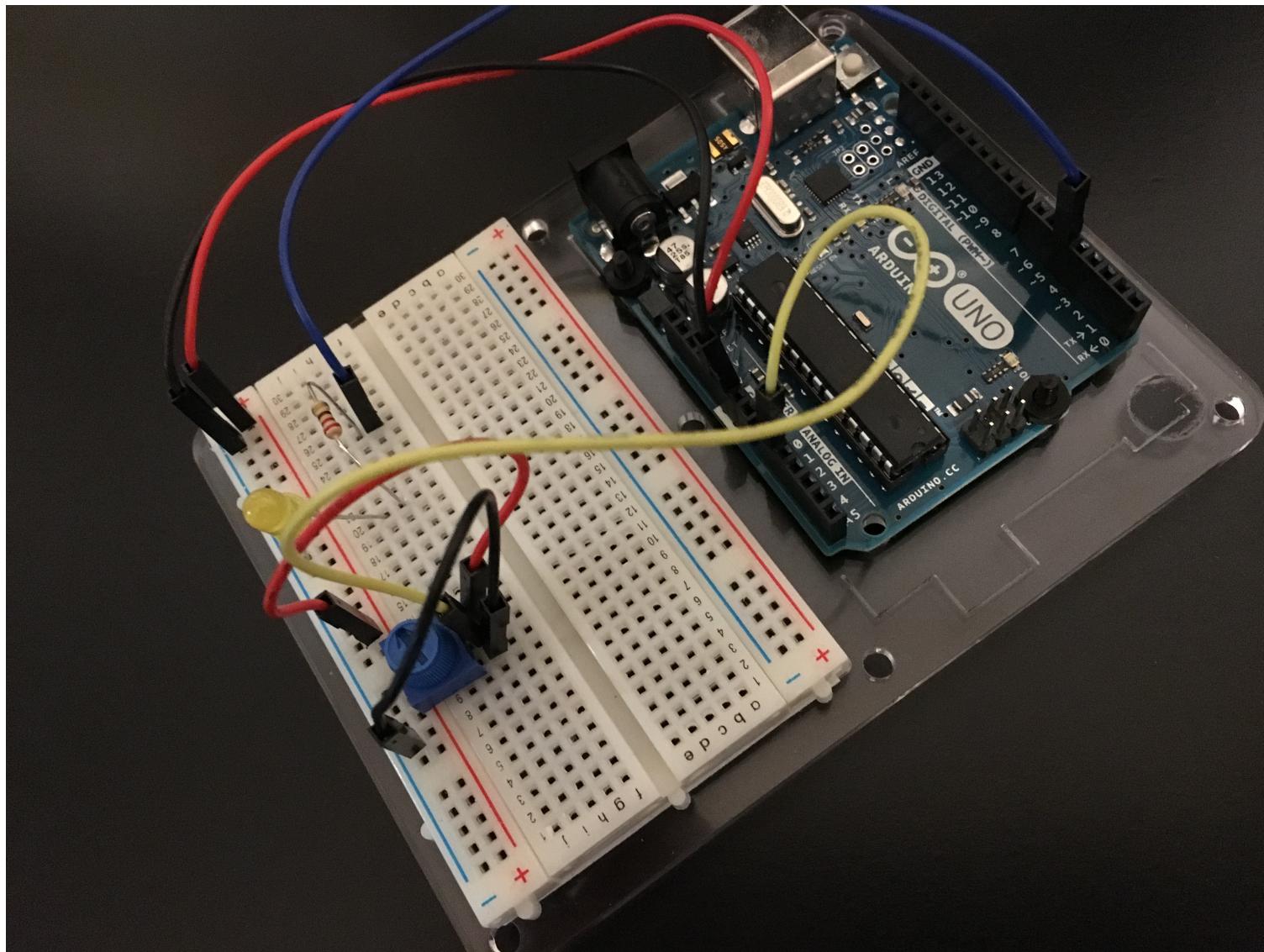
Light Dimmer Switch

Connect a potentiometer and an LED to the Arduino. Control the brightness of the LED based on the turning of the dial.

CIRCUIT SCHEMATIC



BUILD PHOTO - Light Dimmer Switch



BUILD PHOTOS - Light Dimmer Switch

SKETCH CODE - Light Dimmer Switch

```
int potentiometerPin = A0; // connect potentiometer to analog pin
int LEDPWMPin = 5; // set pin (any labeled ~) as PWM pin to write to

void setup()
{
    Serial.begin(9600); // initialize serial monitor
}

void loop()
{
    int potentiometerValue = analogRead(potentiometerPin);

    // read potentiometer value as input, and divide by 4 (analogRead values range from
    // 0-1023 and analogWrite from 0-255), write to LED via PWM pin
    analogWrite(LEDPWMPin, potentiometerValue / 4);
    Serial.println(potentiometerValue); // set serial monitor output
    delay(500); // delay to make change in values easier to read
}
```

SERIAL MONITOR OUTPUT - Light Dimmer Switch

620
629
642
628
641
629
642
282
96
44
24
25
28
28
33
20
403
577
682
1001
1023
1023
1023
1023
1022
763
733
359

REFLECTION - Light Dimmer Switch

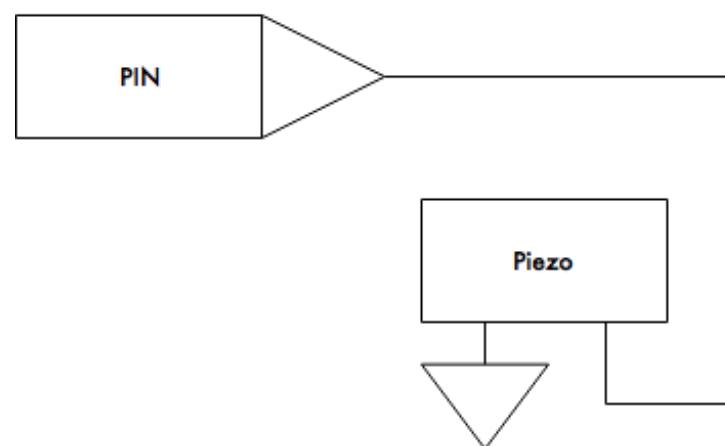
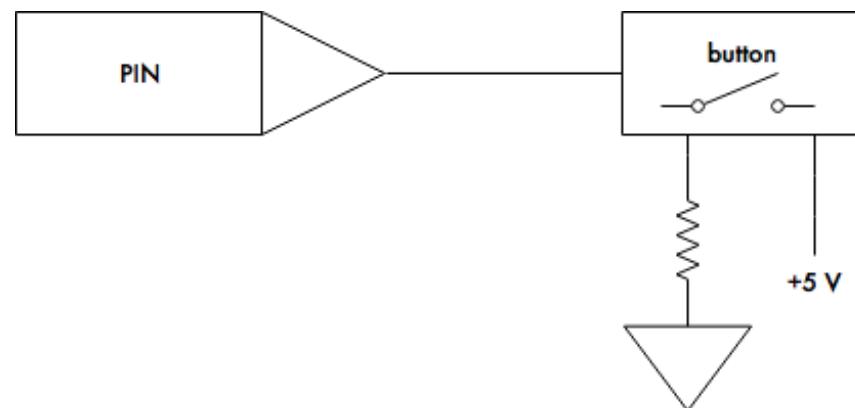
This was the build I ended up creating for my wildcard last week, so I just worked to try to recreate it and commit the steps to memory; it was good practice! I feel like I'm becoming more adept with setting up the builds. I did learn to be more careful about putting my resistors away in the proper bags when I ended up with a pretty dim LED on one pass.

I also experimented with the Fade tutorial using PWM to control the LED without the potentiometer input, which was interesting.

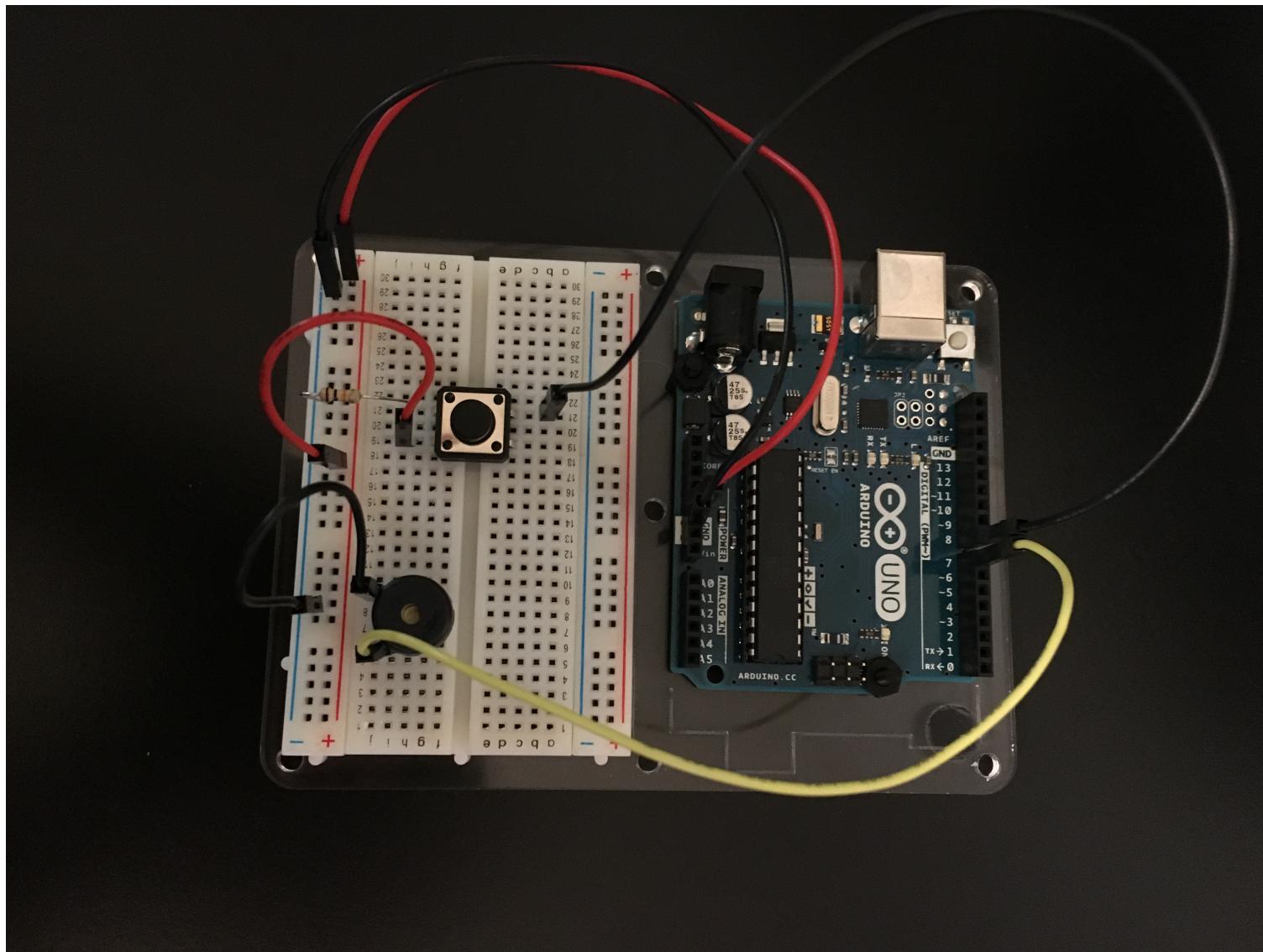
Doorbell

Connect a button and a piezo buzzer to the Arduino, so that the buzzer emits sound when the button is pressed.

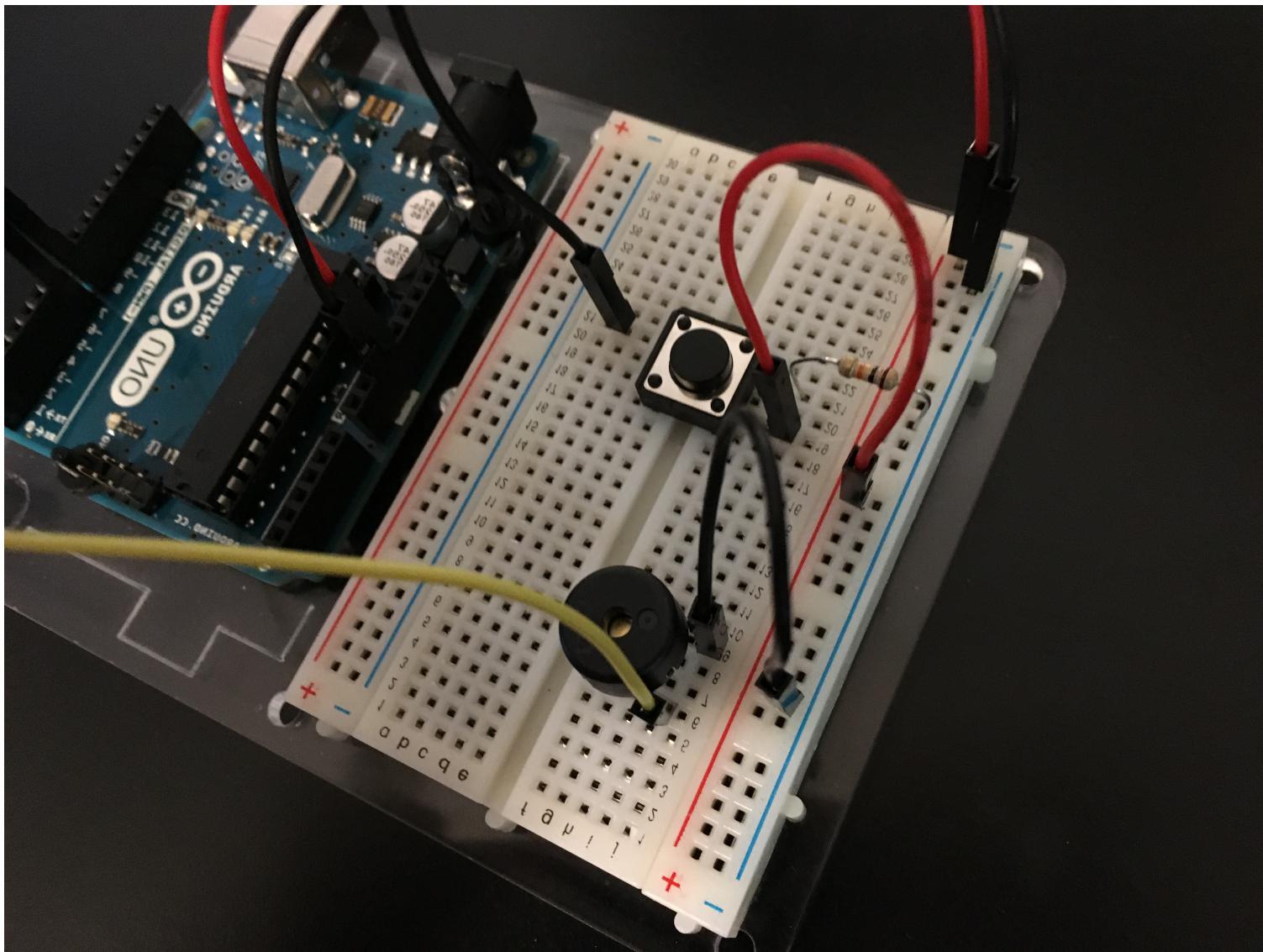
CIRCUIT SCHEMATIC



BUILD PHOTOS - Doorbell



BUILD PHOTOS - Doorbell



SKETCH CODE - Doorbell

```
#include <Button.h> // This uses the button library created by Andrew Davidson. http://blogs.uw.edu/fizzlab/technology/libraries
#include "pitches.h" // This uses the pitches found on Arduino.cc https://www.arduino.cc/en/Tutorial/ToneMelody?from=Tutorial.Tone

int doorbellPin = 8;
int ringPin = 7;
Button doorbell (doorbellPin); // button is connected to doorbellPin

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600); // initialize serial monitor
    pinMode(LED_BUILTIN, OUTPUT); // set built-in LED to output for debugging
    pinMode(doorbellPin, INPUT);
    pinMode(ringPin, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    int action = doorbell.checkButtonAction();

    if (action == Button::CLICKED) {
        Serial.println("Someone rang the doorbell!");
        tone(ringPin, NOTE_C8, 500); // play C7 as defined in pitches file
        delay(500);
        tone(ringPin, NOTE_C8, 500); // play C7 as defined in pitches file
        delay(500);
        tone(ringPin, NOTE_C8, 800); // play C7 as defined in pitches file
        delay(1000);
        tone(ringPin, NOTE_C8, 500); // play C7 as defined in pitches file
    }
}
```

```
delay(300);
tone(ringPin, NOTE_B7, 500); // play B7 as defined in pitches file
delay(500);
tone(ringPin, NOTE_A7, 500); // play A7 as defined in pitches file
delay(300);
tone(ringPin, NOTE_B7, 800); // play B7 as defined in pitches file
delay(300);
tone(ringPin, NOTE_C8, 500); // play C7 as defined in pitches file
delay(500);
tone(ringPin, NOTE_D8, 800); // play D7 as defined in pitches file
digitalWrite(LED_BUILTIN, HIGH); // turn on LED for debugging
}
else if (action == Button::HELD_CLICKED) {
Serial.println("Someone REALLY wants you to come to the door!");
tone(ringPin, 1047, 500); // play tone 1047 Hz
delay(100);
tone(ringPin, 1175, 500); // play tone 1175 Hz
delay(100);
tone(ringPin, 1047, 500); // play tone 1047 Hz
delay(100);
tone(ringPin, 1047, 500); // play tone 1047 Hz
delay(100);
tone(ringPin, 1175, 500); // play tone 1175 Hz
delay(100);
tone(ringPin, 1047, 500); // play tone 1047 Hz
digitalWrite(LED_BUILTIN, HIGH); // turn on LED
}
else {
Serial.println("Are you lonely?");
digitalWrite(LED_BUILTIN, LOW); // turn off LED
}
}
```

SERIAL MONITOR OUTPUT - Doorbell

```
Are you lonely? // no button press
Are you lonely?
Someone REALLY wants you to come to the door! // held press
Someone REALLY wants you to come to the door!
Are you lonely?
Are you lonely?
Are you lonely?
Someone rang the doorbell! // single quick press
Someone rang the doorbell! // single quick press
Are you lonely?
Someone rang the doorbell! // single quick press
Are you lonely?
```

Someone REALLY wants you to come to the door! // held press

Are you lonely?

REFLECTION - Doorbell

This build was fun; I spent more time experimenting with patterns and tones than setting up the actual functionality. Initially, I set up the piezo to emit sound at specific frequencies when the button was directly pressed, but later I went back and leveraged the button library and added different behaviors for a quick press and holding the button down, both using a pitch mapping definition file and direct frequencies as examples. I also got a bit cheeky with my serial monitor output.