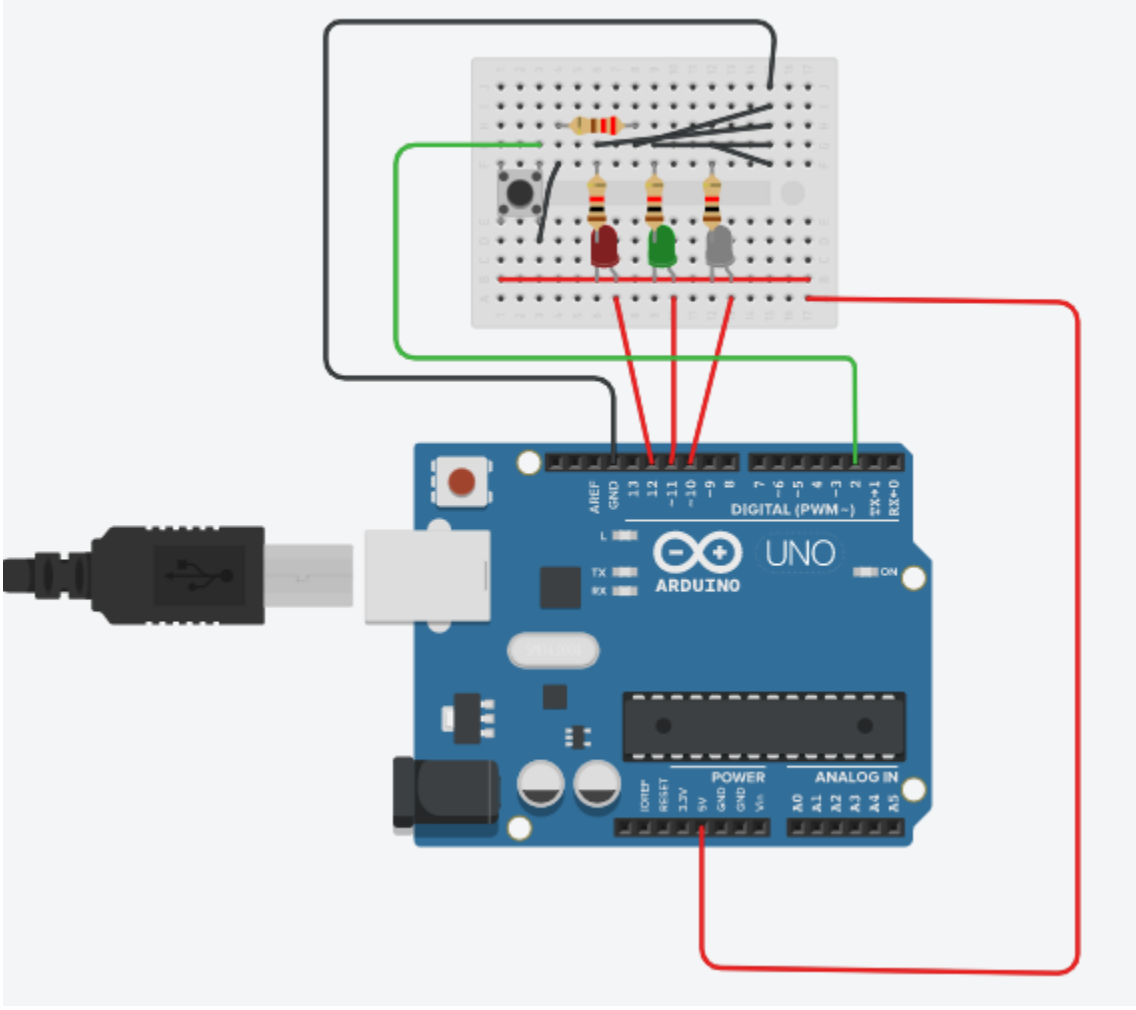
Hermoine Potions

The implementation for this task was very straightforward using the timer.h library. We have 4 tasks that need to be registered and they are all triggered by a button press.

The circuit:



It’s a very simple circuit where we have 3 LEDS connected to the Arduino along with one button. Each component is connected to a current limiting resistor and then to ground. The button is connected to an interrupt pin as that will be very important in the code.

The code:

```
9   Timer<5> timer; // creating timer
```

First thins was creating the timer object that can store up to 5 concurrent tasks.

```
11  void start_routine(){ // method to start everything
12    timer.cancel(); // cancel all to start from the beginning
13    digitalWrite(RED, HIGH); // turns on red led
14    timer.in(15*60*1000, redOff); // turn redd led off in 15 minutes
15    timer.every(2*60*1000, blinkGreen); // blink green led every 2 minutes
16    timer.in(5*60*1000, blinkWhite); // blink white led in 5 minutes
17    timer.in(8*60*1000, blinkWhite); // blink white led in 8 minutes
18  }
```

The start routine function is where everything is connected. Once the button is pressed, an interrupt occurs, and the timer object cancels all previous tasks. It then adds the 4 tasks that are given as follows: turn on the red LED and turn it off in 15 minutes, blink green led every 2 minutes, blink white led at 5 minutes and 8 minutes of the routine.

```
46  void setup()
47  {
48    pinMode(WHITE, OUTPUT);
49    pinMode(GREEN, OUTPUT);
50    pinMode(RED, OUTPUT);
51    pinMode(BTN, INPUT_PULLUP);
52    attachInterrupt(digitalPinToInterrupt(BTN), start_routine, FALLING);
53  }
```

All there is in the setup is the pin modes for the LEDs as well as the interrupt for the button.

```
55  void loop()
56  {
57    timer.tick();
58  }
```

The only thing in the loop is the timer.Tick() function that updates the timer.