Lab 2: Morse Code Decoder

ESE350: Embedded Systems & Microcontroller Laboratory
University of Pennsylvania

In this document, you'll fill out your responses to the questions listed in the Lab 2 Manual. Please fill out your name and link your Github repository below to begin. Be sure that your code on the repo is up-to-date before submission!

For all the questions that require a video, provide a link to the video (e.g. youtube, google drive, etc.).

Student Name: Jacob Bendell
Pennkey: jbendell, 73815023
GitHub Repository:

1. Blinking 4 LEDs

```
4
5   void Initialize()
6   {
7       DDRB |= (1 << DDB1);
8       DDRB |= (1 << DDB2);
9       DDRB |= (1 << DDB3);
10      DDRB |= (1 << DDB4);
11  }
12
13  int main(void)
14  {
15      Initialize();
16
17      /* Replace with your application code */
18      while (1)
19      {
20          PORTB |= (1<<PORTB1);
21          PORTB |= (1<<PORTB2);
22          PORTB |= (1<<PORTB3);
23          PORTB |= (1<<PORTB4);
24          _delay_ms(10000);
25          PORTB &= ~(1<<PORTB1);
26          PORTB &= ~(1<<PORTB2);
27          PORTB &= ~(1<<PORTB3);
28          PORTB &= ~(1<<PORTB4);
29          _delay_ms(10000);
30      }
31  }
32
33
```

2. LED w/ a switch

```
int main(void)
{
    Initialize();

    /* Replace with your application code */
    while (1)
    {
        if (PIND & (1 << PIND7))
        {
            PORTB |= (1 << PORTB1);
        }
        else
        {
            PORTB &= ~(1 << PORTB1);
        }
    }
}
```

3. Sequencing LEDs

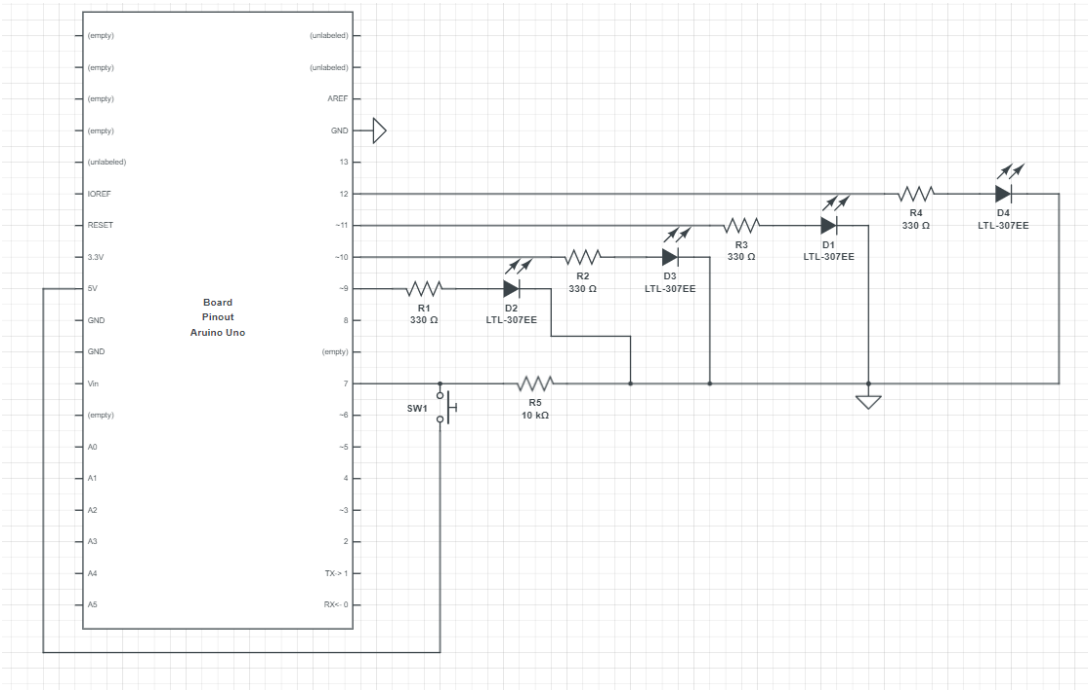
```
int main(void)
{
    Initialize();
    //turn all off
    PORTB &= ~(1<<PORTB1);
    PORTB &= ~(1<<PORTB2);
    PORTB &= ~(1<<PORTB3);
    PORTB &= ~(1<<PORTB4);

    int light = 1;

    while (1)
    {
        if (light == 1) {
            PORTB |= (1 << PORTB1);
            //turn the rest off
            PORTB &= ~(1<<PORTB2);
            PORTB &= ~(1<<PORTB3);
            PORTB &= ~(1<<PORTB4);
        } else if (light == 2) {
            PORTB |= (1<<PORTB2);
            //turn the rest off
            PORTB &= ~(1<<PORTB1);
            PORTB &= ~(1<<PORTB3);
            PORTB &= ~(1<<PORTB4);
```

```
40              PORTB &= ~(1<<PORTB1);
41              PORTB &= ~(1<<PORTB3);
42              PORTB &= ~(1<<PORTB4);
43          } else if (light == 3) {
44              PORTB |= (1<<PORTB3);
45              //turn the rest off
46              PORTB &= ~(1<<PORTB2);
47              PORTB &= ~(1<<PORTB1);
48              PORTB &= ~(1<<PORTB4);
49          } else {
50              PORTB |= (1<<PORTB4);
51              //turn the rest off
52              PORTB &= ~(1<<PORTB2);
53              PORTB &= ~(1<<PORTB3);
54              PORTB &= ~(1<<PORTB1);
55          }
56
57          if (PIND & (1 << PIND7))
58          {
59              light = light + 1;
60              if (light == 5) {
61                  light = 1;
62                  }
63              _delay_ms(10000);
64          }
65
66      }
67  }
68          /*
69          PORTB |= (1<<PORTB1);
```

4. Sequencing LED Schematic



5. An advantage of using interrupts is that the CPU is only used when the interrupt is triggered unlike with polling where an infinite loop is run to constantly look for an event. A disadvantage of interrupts is that they have a slower response time compared to a tight polling loop. The loop's only job is to be looking for the event

trigger while the interrupt allows the CPU to be performing other tasks not associated with the event.

6. Number of ticks 16 MHz clock
    - 30 ms → 16,000,000 * .03 = 480000
    - 200 ms → 16,000,000 * .2 = 3200000
    - 400 ms → 16,000,000 * .4 = 6400000
7. A prescaler effectively allows us to slow down the clock. The MCU is able to be programmed to increment every 1, 8, 64, 256, and 1024 ticks. The System Clock and the Timer Clocks can be prescaled by dividing the prescaling values. Because we are limited to 8 bit and 16 bit registers, if low frequency events are to be measured, a prescaler is necessary to not cause the register to overflow.
8. Demo submitted in the form
9. Translate: Someday I will rule you all
10. n/a