Lab 2: Morse Code Decoder

ESE3500: 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!
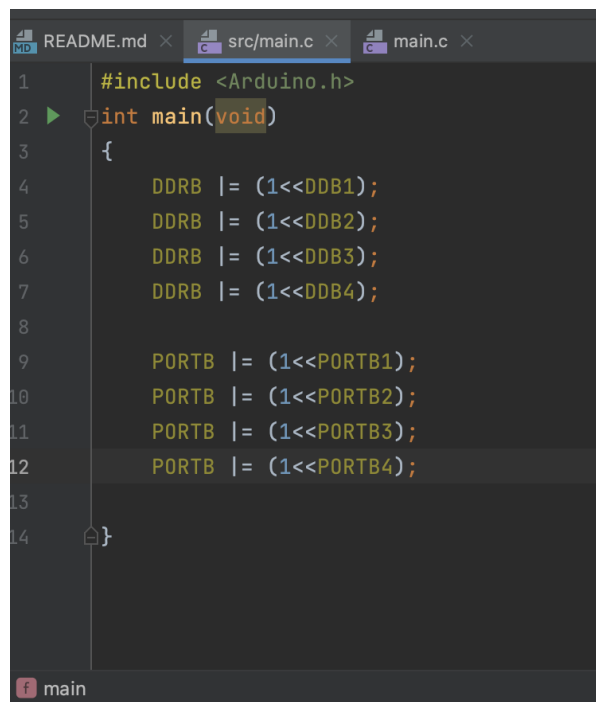
For all the questions that require a video, provide a link to the video (e.g. youtube, google drive, etc.).

Student Name: Roberto Milan
Pennkey: rmilan
GitHub Repository: https://github.com/ese3500/lab2-rmilan.git

1.

2.

```c
#include <Arduino.h>
int main(void)
{
    DDRB |= (1<<DDB1);
    DDRB |= (1<<DDB2);
    DDRB |= (1<<DDB3);
    DDRB |= (1<<DDB4);

    DDRD &= ~(1<<DDD7);

    while(1) {
        if (PIND & (1 << PIND7)) {
            PORTB |= (1 << PORTB1);
            PORTB |= (1 << PORTB2);
            PORTB |= (1 << PORTB3);
            PORTB |= (1 << PORTB4);
        } else {
            PORTB &= ~(1 << PORTB1);
            PORTB &= ~(1 << PORTB2);
            PORTB &= ~(1 << PORTB3);
            PORTB &= ~(1 << PORTB4);
        }
    }
}
```

3.

```c
int main(void)
{
    int count = 0;
    DDRB |= (1<<DDB1);
    DDRB |= (1<<DDB2);
    DDRB |= (1<<DDB3);
    DDRB |= (1<<DDB4);

    DDRD &= ~(1<<DDD7);

    while(1) {
        if (PIND & (1 << PIND7)){
            if(count == 0) {
                PORTB |= (1 << PORTB1);
                PORTB &= ~(1 << PORTB4);
                count++;
                _delay_ms( ms: 500);
                //wait till button is released??
            } else if (count == 1) {
                PORTB |= (1 << PORTB2);
                PORTB &= ~(1 << PORTB1);
                count++;
                _delay_ms( ms: 500);
            } else if (count == 2) {
                PORTB |= (1 << PORTB3);
                PORTB &= ~(1 << PORTB2);
                count++;
                _delay_ms( ms: 500);
            } else if (count == 3) {
                PORTB |= (1 << PORTB4);
                PORTB &= ~(1 << PORTB3);
                count = 0;
                _delay_ms( ms: 500);
            }
        }
    }
}
```

4.
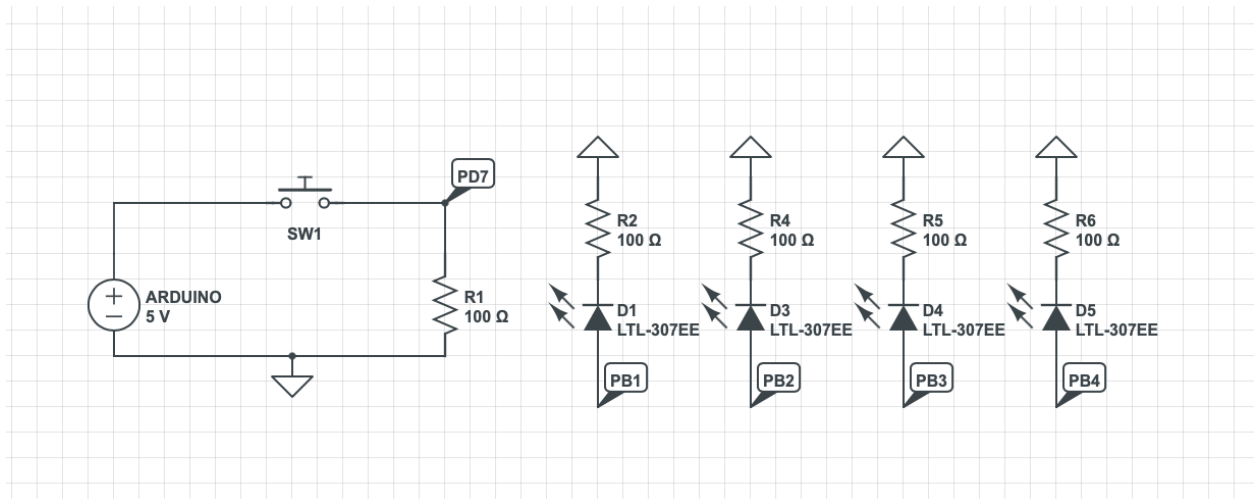
5.

One advantage is that the ISR can run once, when the input capture detects a rising edge, and then allow the rest of the code to run and when it detects a falling edge it will run again and allow the rest of the code to run. When using polling you have another pin stuck in a state waiting for the button to be released.
Polling in this situation would only need one if statement and not as structured code, also you are wasting an input capture pin on a menial task in this situation.

6.

30ms is 480000 ticks, 200ms is 3200000 ticks, and 400ms is 6400000 ticks.

7.

The prescaler allows us to decrease the number of ticks per second that the built-in clock or a timer that we are using does. This decreases the frequency from the set 16MHz and allows us to do specific actions (like create a square wave) based on the speed of our prescaled clock/timer.

8.

9. I used 480ms for the upper bound of the dash and as the threshold prior to deciding if there was a space.

Video

(sometimes the buttons debasing would screw up dot/dash and also there were times when the button was not pressed but a falling edge was not detected, I assume this has to do with the button but perhaps my code is jank, so the values got messed up since the code thought that the button was pressed)
(also space occurs when the button is pressed after 480ms so in the terminal, it will show the character for the prior presses directly before the space is shown)

10.

SOMEDAY I WILL RULE YOU ALL

11.

12.

13.