

# 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: Jared Lebos**

**Pennkey: 79827692**

**GitHub Repository: [https://github.com/lebj09/lab2\\_79827692/tree/main](https://github.com/lebj09/lab2_79827692/tree/main)**

```
#include <avr/io.h>
int main (void)
{
    DDRB |= (1<<DDB1); // output pin
    DDRB |= (1<<DDB2); // output pin
    DDRB |= (1<<DDB3); // output pin
    DDRB |= (1<<DDB4); // output pin

    PORTB |= (1<<PORTB0); // internal pull up
    DDRB &= ~(1<<DDB0); // input pin

    PORTB |= (1<<PORTB4);
    PORTB |= (1<<PORTB3);
    PORTB |= (1<<PORTB2);
    PORTB |= (1<<PORTB1);

    while(1){

    }
}
```

1.

```

* GccApplication3.c
*
* Created: 2/12/2022 12:44:05 AM
* Author : Jared
*/
#include <avr/io.h>
int main (void)
{

    DDRB |= (1<<DDB1); // output pin

    PORTD |= (1<<PORTD7); // internal pull up
    DDRD &= ~(1<<DDD7); // input pin

    PORTB |= (1<<PORTB1);

    while(1){
if(PIND &(1<<PIND7)){
    // turn LED on
    PORTB |= (1<<PORTB1); // PORTB5
    // _delay_ms(1000);
}
else{
    // turn LED off
    PORTB &= ~(1<<PORTB1); // PORTB5
    // _delay_ms(1000);
}

    }
}

```

2.

```

//
#include <avr/io.h>
#include <util/delay.h>

int count = 1;
int prev = 0;

int main (void)
{
    DDRB |= (1<<DDRB1); // output pin
    DDRB |= (1<<DDRB2); // output pin
    DDRB |= (1<<DDRB3); // output pin
    DDRB |= (1<<DDRB4); // output pin

    PORTD |= (1<<PORTD7); // internal pull up
    DDRD &= ~(1<<DDRD7); // input pin

    while(1){

        if(PIND &(1<<PIND7)){prev=0;}
        // nested if to reset counter

        else{
            if(prev==0){

                _delay_ms(100); //debounce

                count = count +1 ; //increment the count

                if( count > 4){count = 1;} // nested if to reset counter
                prev=1;
            }

        }

        if (count == 1){
            PORTB |= (1<<PORTB1);
            PORTB &= ~(1<<PORTB2); // PORTB
            PORTB &= ~(1<<PORTB3); // PORTB
            PORTB &= ~(1<<PORTB4); // PORTB
        }

        else if (count == 2){
            PORTB &= ~(1<<PORTB1);
            PORTB |= (1<<PORTB2); // PORTB
            PORTB &= ~(1<<PORTB3); // PORTB
            PORTB &= ~(1<<PORTB4); // PORTB
        }

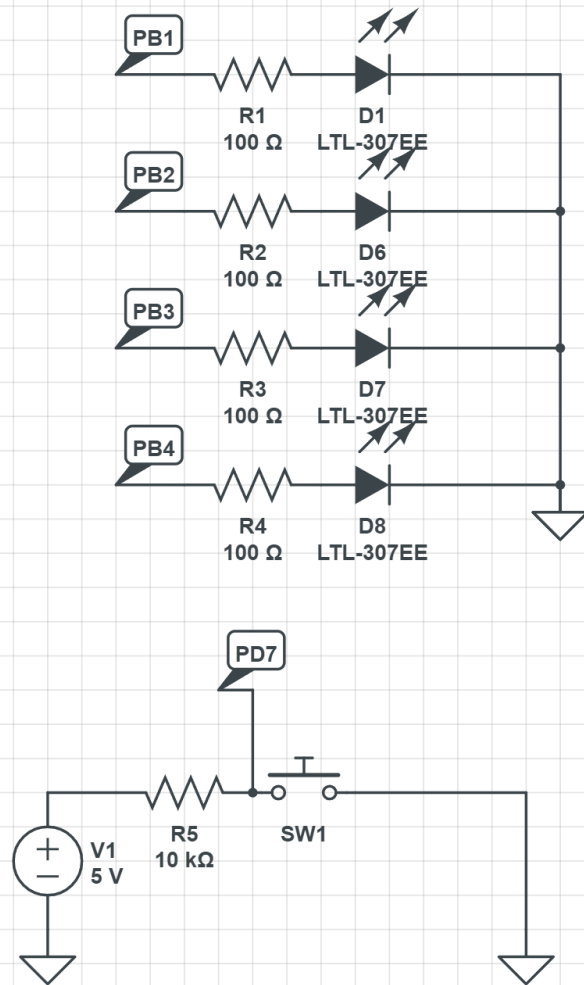
        else if (count == 3){
            PORTB &= ~(1<<PORTB2);
            PORTB |= (1<<PORTB3); // PORTB
            PORTB &= ~(1<<PORTB1); // PORTB
            PORTB &= ~(1<<PORTB4); // PORTB
        }

        else if (count == 4){
            PORTB &= ~(1<<PORTB2);
            PORTB |= (1<<PORTB4); // PORTB
            PORTB &= ~(1<<PORTB3); // PORTB
            PORTB &= ~(1<<PORTB1); // PORTB
        }

        else{
            PORTB &= ~(1<<PORTB2);
            PORTB &= ~(1<<PORTB4); // PORTB
            PORTB &= ~(1<<PORTB3); // PORTB
            PORTB &= ~(1<<PORTB1); // PORTB
        }

    }
}

```



4.

5. Interrupts allow for the processor to do other things while still being able to catch an input.

6. 30ms = 480 000, 200ms = 3200 000, 400ms = 6400 000

Prescaler allows us to 'lower' the timer frequency by a certain factor therefore making each tick take a longer time.