

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: Erica Feehery

Pennkey: efeehery

GitHub Repository: <https://github.com/eese3500/lab-2-morse-feehe21>

1.

```
//PB1
DDRB |= (1<<DDB1); //set to output pin
PORTB |= (1<<PORTB1); // set to high

//PB2
DDRB |= (1<<DDB2); //set to output pin
PORTB |= (1<<PORTB2); // set to high

//PB3
DDRB |= (1<<DDB3); //set to output pin
PORTB |= (1<<PORTB3); // set to high

//PB4
DDRB |= (1<<DDB4); //set to output pin
PORTB |= (1<<PORTB4); // set to high
```

2.

```

int main(void)
{
    //PB1
    DDRB |= (1<<DDB1); //set to output pin
    PORTB |= (1<<PORTB1); // set to high

    //PB2
    DDRB |= (1<<DDB2); //set to output pin
    PORTB |= (1<<PORTB2); // set to high

    //PB3
    DDRB |= (1<<DDB3); //set to output pin
    PORTB |= (1<<PORTB3); // set to high

    //PB4
    DDRB |= (1<<DDB4); //set to output pin
    PORTB |= (1<<PORTB4); // set to high

    //PD7
    DDRD &= ~(1<<DDD7); //set to input pin

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

```

3.

```

int main(void)
{
    char c = 0;
    bool prev_state = 0;
    //PB1
    DDRB |= (1<<DDB1); //set to output pin
    PORTB &= ~(1<<PORTB1); // set to low

    //PB2
    DDRB |= (1<<DDB2); //set to output pin
    PORTB &= ~(1<<PORTB2); // set to low

    //PB3
    DDRB |= (1<<DDB3); //set to output pin
    PORTB &= ~(1<<PORTB3); // set to low

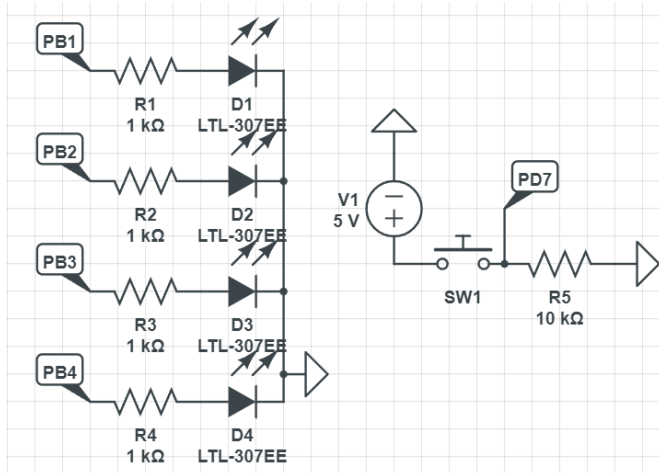
    //PB4
    DDRB |= (1<<DDB4); //set to output pin
    PORTB &= ~(1<<PORTB4); // set to low

    //PD7
    DDRD &= ~(1<<DDD7); //set to input pin
    ...

    while (1)
    {
        bool current_state = PIND & (1<<PIND7);
        if (current_state && !prev_state) {
            c++;
        }
        prev_state = current_state;
        if (c >= 4) {
            c -= 4;
        }
        if (c == 0) {
            PORTB |= (1<<PORTB1); // set to high
        } else {
            PORTB &= ~(1<<PORTB1); // set to low
        }
        if (c == 1) {
            PORTB |= (1<<PORTB2); // set to high
        } else {
            PORTB &= ~(1<<PORTB2); // set to low
        }
        if (c == 2) {
            PORTB |= (1<<PORTB3); // set to high
        } else {
            PORTB &= ~(1<<PORTB3); // set to low
        }
        if (c == 3) {
            PORTB |= (1<<PORTB4); // set to high
        } else {
            PORTB &= ~(1<<PORTB4); // set to low
        }
    }
}

```

4.



5. Give an advantage and a disadvantage to using interrupts over polling for this task.
 - a. An advantage of using interrupts for this task is that it uses less CPU power to check the condition.
 - b. A disadvantage of using interrupts for this task is that it is more difficult to conceptualize and trace the order of execution.
6. For a 16MHz clock, how many “ticks” are in 30ms, 200ms, and 400ms?
 - a. 30ms: $30 * 10^{-3} * 16 * 10^6 = 480,000$ ticks
 - b. 200ms: $200 * 10^{-3} * 16 * 10^6 = 3,200,000$ ticks
 - c. 400ms: $400 * 10^{-3} * 16 * 10^6 = 6,400,000$ ticks
7. Describe how a prescaler allows us to work with a wider range of frequencies on our microcontroller.
 - a. A prescaler adjusts the clock such that we receive ticks at a reduced frequency. It essentially “groups” the ticks at its original frequency together, such that we receive a tick once per some specified number of the original clock ticks. Thus, the prescaler allows us to work with frequencies lower than the clock’s frequency.
8. <https://drive.google.com/file/d/1qxWD2GHYIOXUKRWiTODXV-sWJPQF0MEN/view?usp=sharing>
 - a. Better quality: <https://youtu.be/1tmgyLxrAaE>
9. SOMEDAY ILL RULE YOU ALL
10. <https://youtu.be/MZqwwMEJIJY>