# Lab 2: Morse Code Decoder

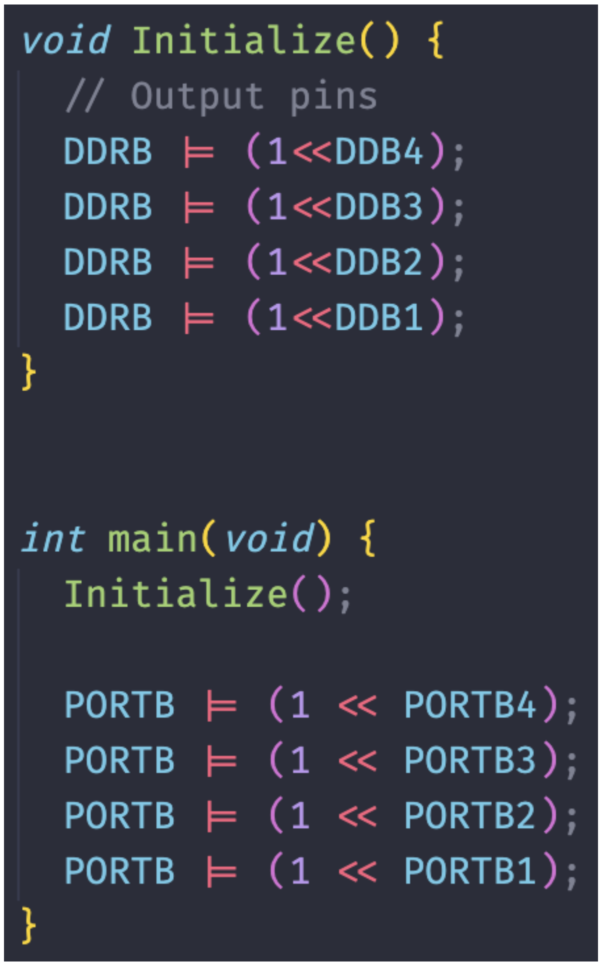
ESE3500: Embedded Systems & Microcontroller Laboratory
University of Pennsylvania

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.).

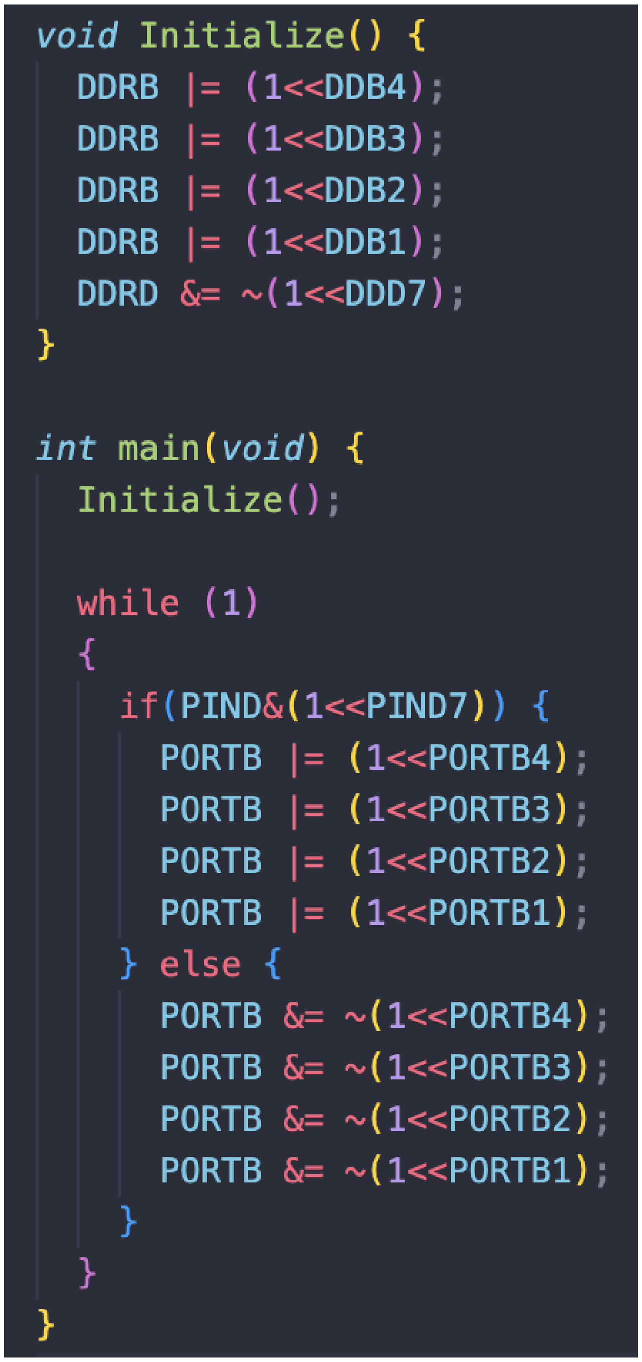
**Student Name: Alexander Kyimpopkin**
**Pennkey: alxkp**
**GitHub Repository: https://github.com/ese3500/lab-2-morse-alxkp**

1. The screenshot is below:



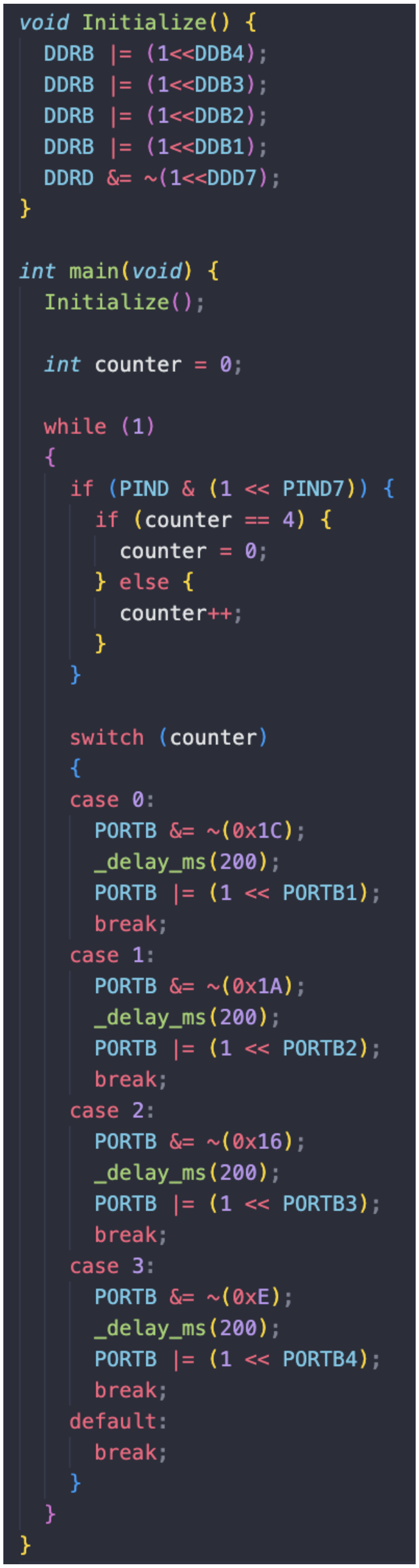
```c
void Initialize() {
  // Output pins
  DDRB |= (1<<DDB4);
  DDRB |= (1<<DDB3);
  DDRB |= (1<<DDB2);
  DDRB |= (1<<DDB1);
}


int main(void) {
  Initialize();

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

2. The screenshot is below:



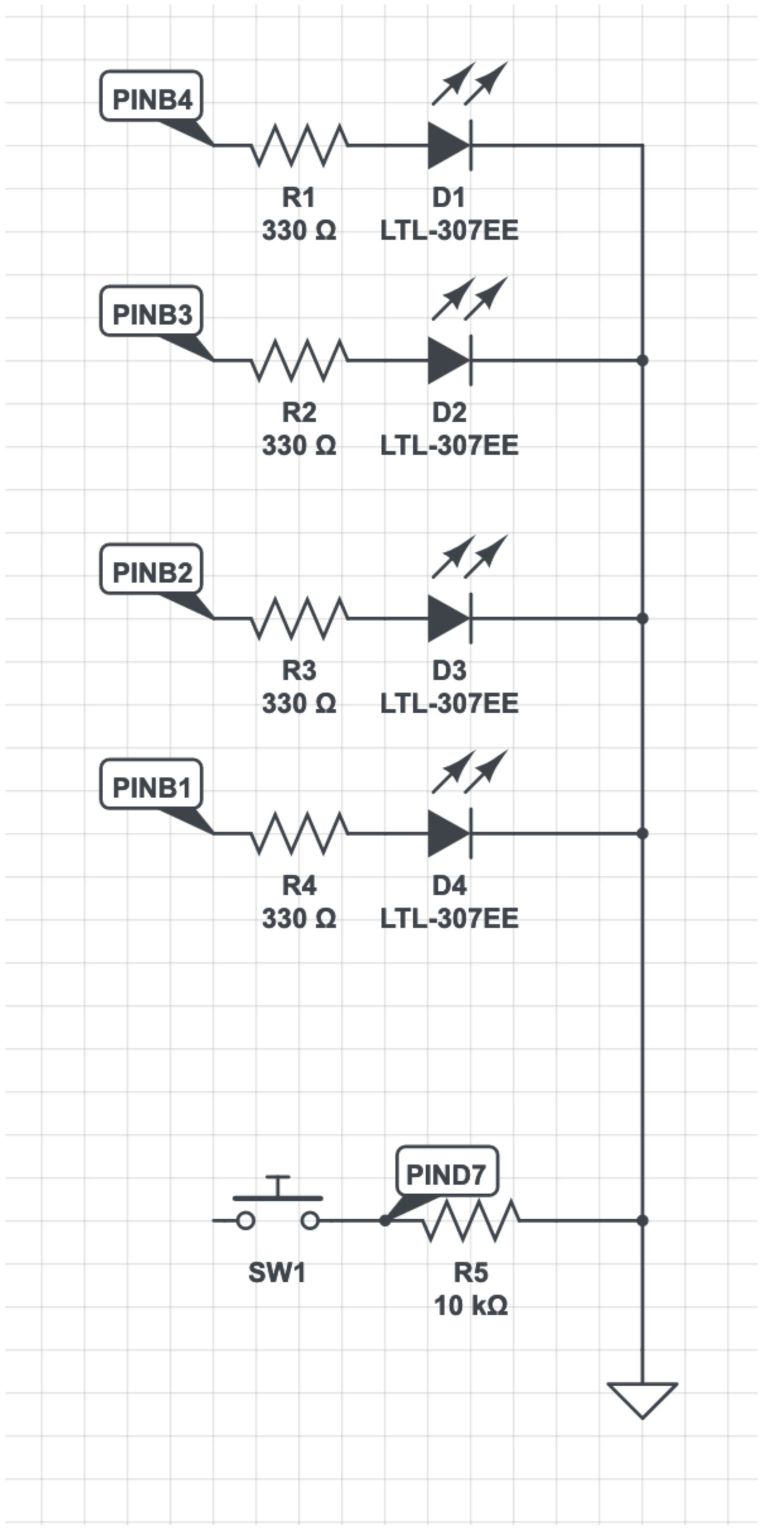
```c
void Initialize() {
  DDRB |= (1<<DDB4);
  DDRB |= (1<<DDB3);
  DDRB |= (1<<DDB2);
  DDRB |= (1<<DDB1);
  DDRD &= ~(1<<DDD7);
}

int main(void) {
  Initialize();

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

3. The screenshot is below



```
void Initialize() {
  DDRB |= (1<<DDB4);
  DDRB |= (1<<DDB3);
  DDRB |= (1<<DDB2);
  DDRB |= (1<<DDB1);
  DDRD &= ~(1<<DDD7);
}

int main(void) {
  Initialize();

  int counter = 0;

  while (1)
  {
    if (PIND & (1 << PIND7)) {
      if (counter == 4) {
        counter = 0;
      } else {
        counter++;
      }
    }

    switch (counter)
    {
    case 0:
      PORTB &= ~(0x1C);
      _delay_ms(200);
      PORTB |= (1 << PORTB1);
      break;
    case 1:
      PORTB &= ~(0x1A);
      _delay_ms(200);
      PORTB |= (1 << PORTB2);
      break;
    case 2:
      PORTB &= ~(0x16);
      _delay_ms(200);
      PORTB |= (1 << PORTB3);
      break;
    case 3:
      PORTB &= ~(0xE);
      _delay_ms(200);
      PORTB |= (1 << PORTB4);
      break;
    default:
      break;
    }
  }
}
```

4. A screenshot of our schematic is below:



5. An advantage to using interrupts is that they are able to quite literally interrupt other code running, so they will almost (barring a higher priority interrupt firing simultaneously) be able to activate, leading to a very fast and quite predictable

response time for the LED. One of the downsides to using interrupts, however, is that they do not have predictable timing for when they will fire, unlike a polling loop, which will have very predictable execution timing.

6. For a 16MHz clock, we have 16x10^6 ticks per second by definition, and thus we calculate our timings for 30ms, 200ms, and 400ms as follows:
    a. 30ms: 16x10^6 ticks/second * 1 second/1000 ms * 30 ms = 480000 ticks
    b. 200ms: 16x10^6 ticks/second * 1 second/1000 ms * 200 ms = 320000 ticks
    c. 400ms: 16x10^6 ticks/second * 1 second/1000 ms * 400 ms = 640000 ticks
7. A prescaler allows a timer whose clock counter is stored in a 16 bit value to be used with a wider range of times, since it effectively scales down the tick rate by its factor: for instance, we used a clock prescaler of 1024 to slow down the effective clock rate of a 16MHz clock to 15.625kHz, making it usable for applications which do not need to run as fast as the crystal, and without needing to allocate additional memory to track overflows of the 16bit counter.
8. We left the default timings for the 30ms and 200ms values intact for dot and dash respectively, but changed the timing for the pause to be 2.4seconds instead of 400ms, since we were not able to type out morse code particularly quickly. A link is attached to a video showing the first three letters of our name. (https://drive.google.com/file/d/1z151oXgaqd5Bqd5OLEbgV_os6ezA5nWF/view?usp=share_link)
9. The message he tapped out was “Some day I will rule you all”
10. N/A