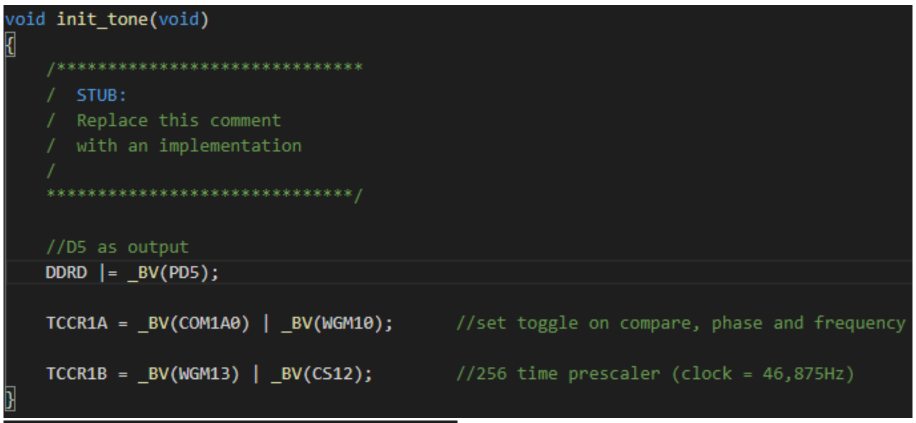
C8 – Timers/Counters and Pulse-Width Modulation

Preparation

2.1.1.Tone Oscillator

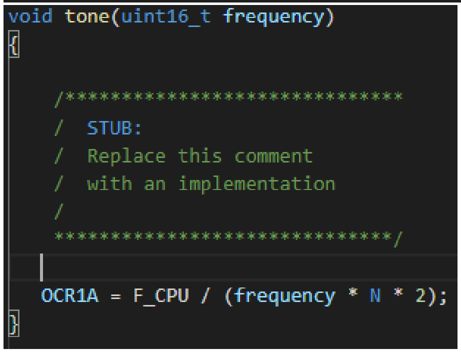
1. Timer 1 is a good choice since it is a 16 bit timer compared to the others which are 8 bit timers. This is better since it allows us to set a greater range of frequencies as there are 16 bits of resolution, the same range of frequency could be done with an 8 bit timer you just have to change the clock divide.
2. By setting the COM1A0 bit, this sets the mode to toggle on compare. Then setting the WGM10 bit this well set the PWM mode to Phase and Frequency correct. This can be done in avr C with the bitwise or and the _BV function.
3. The prescaler should be 8.
4. 

```
void init_tone(void)
{
    /*****************************
    /  STUB:
    /  Replace this comment
    /  with an implementation
    /
    *****************************/

    //D5 as output
    DDRD |= _BV(PD5);

    TCCR1A = _BV(COM1A0) | _BV(WGM10);      //set toggle on compare, phase and frequency

    TCCR1B = _BV(WGM13) | _BV(CS12);        //256 time prescaler (clock = 46,875Hz)
}
```

5. 

```
void tone(uint16_t frequency)
{

    /*****************************
    /  STUB:
    /  Replace this comment
    /  with an implementation
    /
    *****************************/

    OCR1A = F_CPU / (frequency * N * 2);
}
```

2.1.2 PWM Oscillator and Amplitude Control

1. I chose Timer 2 since one of the outputs of Timer 0 is shared with the SPI which is used to program the board, so using Timer 2 avoids possible future programming problems.
2. F = 12,000,000 / (2 * 1 * 2^8) = 23,437.5Hz

3. The Timer 2 should be put into Phase Correct mode by setting the WGM20 and WGM22 bits. Then we set it to toggle mode using bit COM2A0 and set no clock prescale using bit CS20.

2.2 Pin Assignment

Pin Name	Description
PD5	Oscillator / Timer 1
PD6 / PD7	Modulation / Timer 2
PD4	Trigger