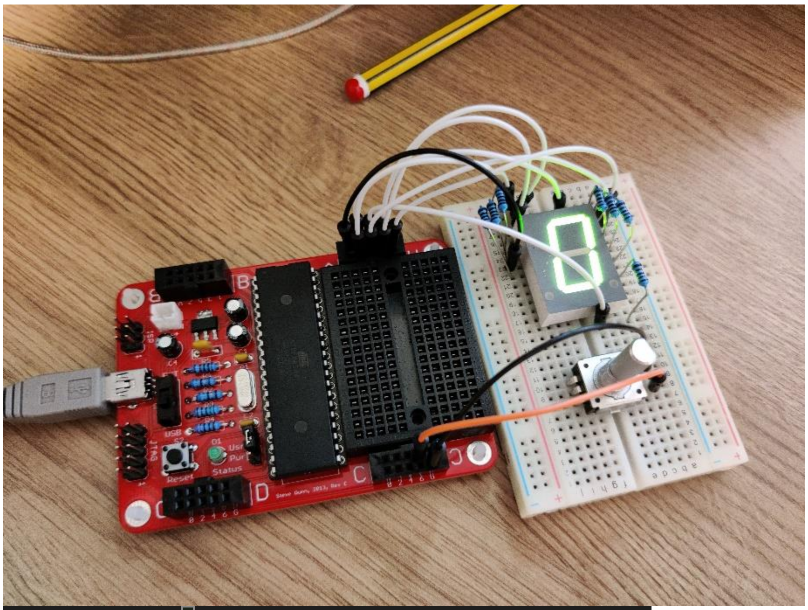
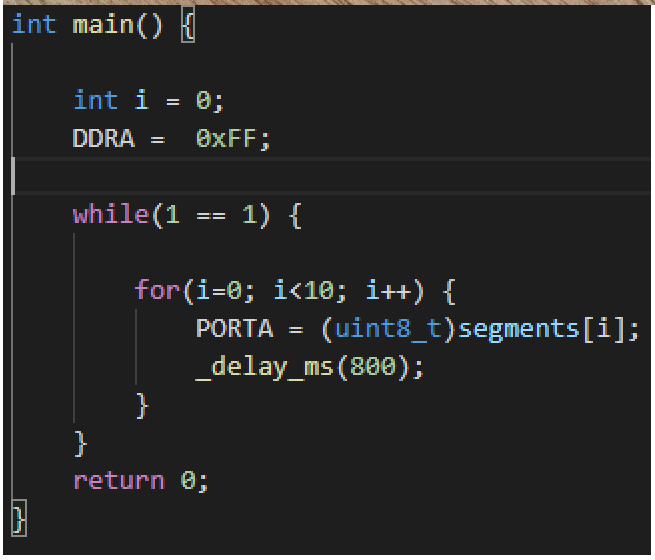
C6 – Digital Input and Output

3.1 Digital Output

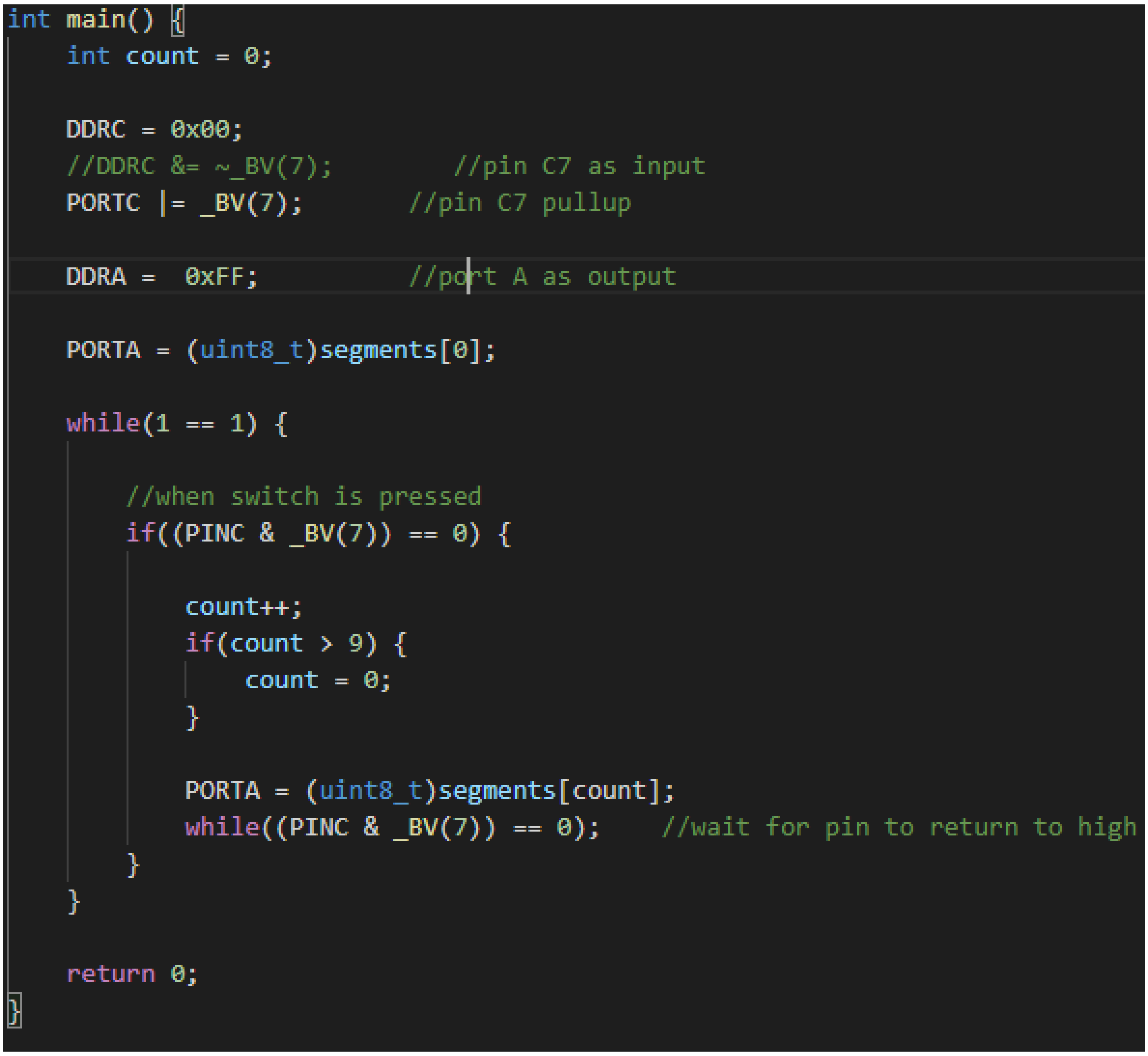




```
int main() {

    int i = 0;
    DDRA =  0xFF;

    while(1 == 1) {

        for(i=0; i<10; i++) {
            PORTA = (uint8_t)segments[i];
            _delay_ms(800);
        }
    }
    return 0;
}
```

3.2 Digital Input

As the switch is pressed, I notice that sometimes, especially when pressed fast the counter will skip one or two numbers. This is due to switch bounce which the microcontroller is registering as separate button presses.



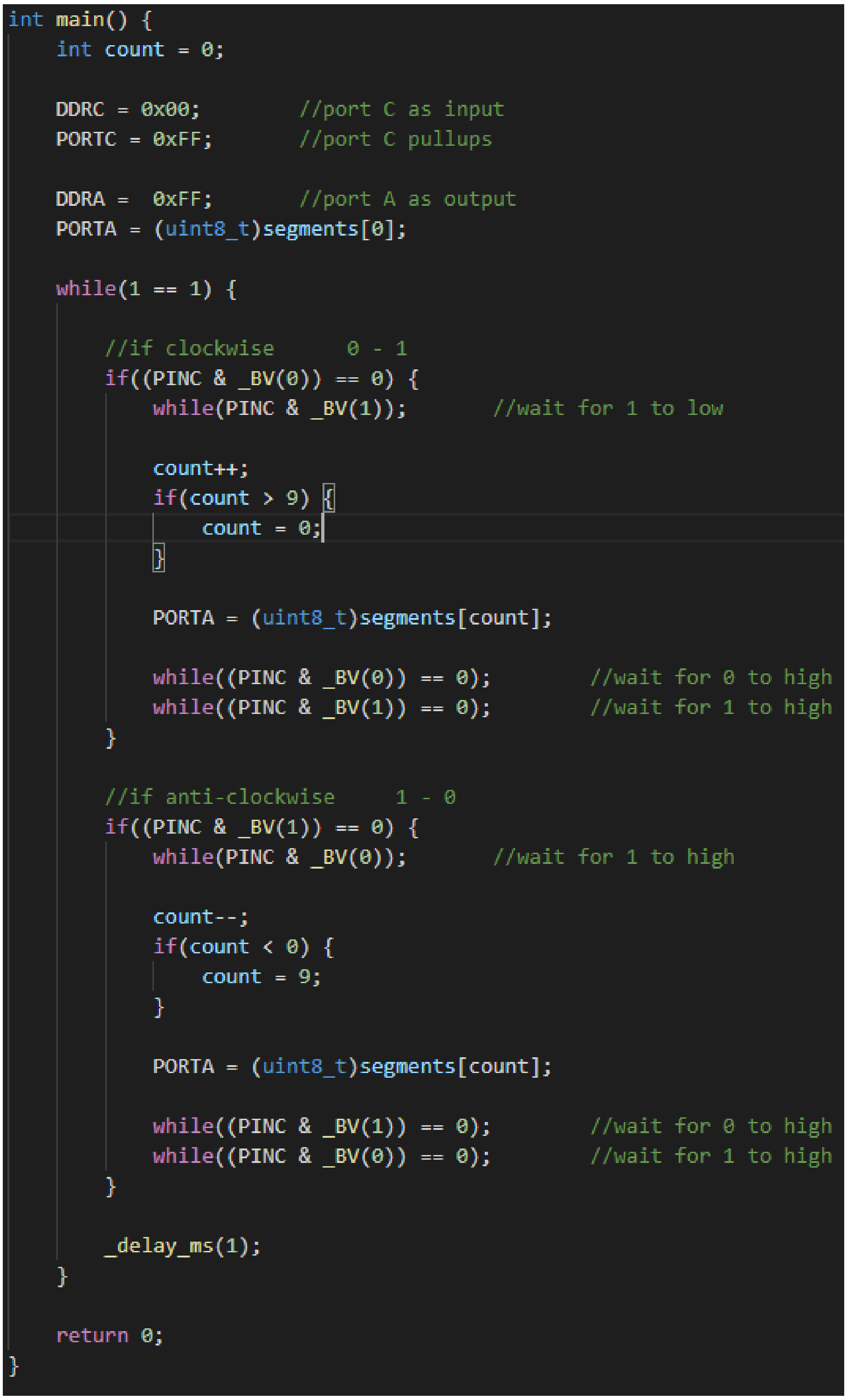
```
int main() {
    int count = 0;

    DDRC = 0x00;
    //DDRC &= ~_BV(7);        //pin C7 as input
    PORTC |= _BV(7);       //pin C7 pullup

    DDRA =  0xFF;          //port A as output

    PORTA = (uint8_t)segments[0];

    while(1 == 1) {

        //when switch is pressed
        if((PINC & _BV(7)) == 0) {

            count++;
            if(count > 9) {
                count = 0;
            }

            PORTA = (uint8_t)segments[count];
            while((PINC & _BV(7)) == 0);    //wait for pin to return to high
        }
    }

    return 0;
}
```

3.3 De-bouncing

To try and alleviate switch bounce I have put a low pass RC filter in series with the output of the switch. This consists of a resistor in series followed by a capacitor to ground which then goes to the pin.

3.4 Rotary Encoder

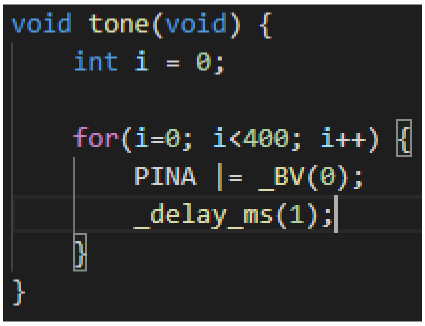
You shouldn’t need to have any additional debouncing for the rotary encoder since in my main loop I added a small delay of 1ms which should eliminate any debounce so the microcontroller doesn’t detect it. The only limitation is if the encoder is turn two notches and one of them is not detected it may register as a decrement instead of an increment and visa versa.



```
int main() {
    int count = 0;

    DDRC = 0x00;        //port C as input
    PORTC = 0xFF;       //port C pullups

    DDRA =  0xFF;       //port A as output
    PORTA = (uint8_t)segments[0];

    while(1 == 1) {

        //if clockwise      0 - 1
        if((PINC & _BV(0)) == 0) {
            while(PINC & _BV(1));       //wait for 1 to low

            count++;
            if(count > 9) {
                count = 0;
            }

            PORTA = (uint8_t)segments[count];

            while((PINC & _BV(0)) == 0);        //wait for 0 to high
            while((PINC & _BV(1)) == 0);        //wait for 1 to high
        }

        //if anti-clockwise     1 - 0
        if((PINC & _BV(1)) == 0) {
            while(PINC & _BV(0));       //wait for 1 to high

            count--;
            if(count < 0) {
                count = 9;
            }

            PORTA = (uint8_t)segments[count];

            while((PINC & _BV(1)) == 0);        //wait for 0 to high
            while((PINC & _BV(0)) == 0);        //wait for 1 to high
        }

        _delay_ms(1);
    }

    return 0;
}
```

3.5 Simple Sound

For this part I modified the previous program so that it plays a sound right after the counter is incremented and decremented. I did this using a function I created called tone() which will play a 500Hz tone for 400ms.



```
void tone(void) {
    int i = 0;

    for(i=0; i<400; i++) {
        PINA |= _BV(0);
        _delay_ms(1);
    }
}
```