

# Skyler MacDougall

## Homework 2: Due 1/29/2020

1. Rewrite Lab 1, page 15, using C and register writes. ArduinoC is allowed for the use of Serial outputs and delays.

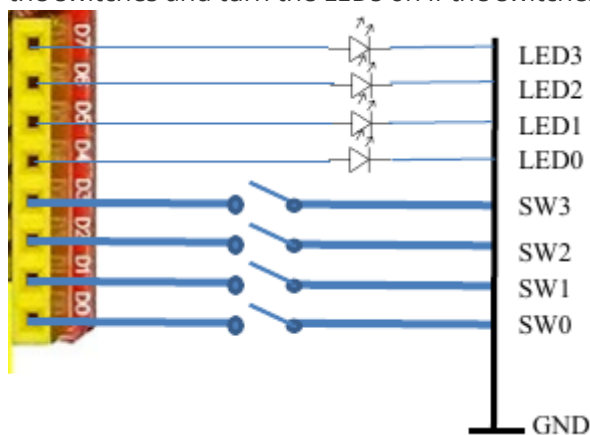
```
1 //Lab_1_hello_arduino
2 //#define LED_PIN 13
3 char inChar;
4 boolean isFreshInChar;
5
6 void setup() {
7     DDRB |= 0x20; //pinMode(LED_PIN , OUTPUT);
8     PORTB &= 0xDF; //digitalWrite(LED_PIN , LOW);
9
10    // Set serial monitor console termination for 'No line ending'
11    Serial.begin(9600);
12    Serial.println("Lab 1: hello arduino v0.3\n");
13    delay(5000);
14 }
15
16 void loop() {
17     isFreshInChar = false;
18     if (Serial.available()) {
19         inChar = Serial.read();
20         Serial.print("Serial input detected: ");
21         Serial.println(inChar);
22         isFreshInChar = true;
23     }
24
25     if (inChar == 'n') PORTB |= 0x20; //digitalWrite(LED_PIN , HIGH);
26     if (inChar == 'f') PORTB &= 0xDF; //digitalWrite(LED_PIN , LOW);
27
28     if (inChar == 'b') { // blink with 25% duty cycle
29         PORTB |= 0x20; //digitalWrite(LED_PIN , HIGH);
30         delay(250);
31         PORTB &= 0xDF; //digitalWrite(LED_PIN , LOW);
32         delay(750);
33     }
34
35     // Discover 't' persistence bug by observing high rate LED blink
36     if (inChar == 't') { // toggle
37         //digitalWrite(LED_PIN , !digitalRead(LED_PIN ));
38         if (PINB & 0x20){
39             PORTB &= 0xDF;
40         }
41         else{
42             PORTB |= 0x20;
43         }
44     }
```

```

45
46 // Add state change detection to get proper toggle action.
47 if (inChar == 'T') { // toggle
48     if (isFreshInChar){//digitalWrite(LED_PIN , !digitalRead(LED_PIN
49     ));
49     if (PINB & 0x20){
50         PORTB &= 0xDF;
51     }
52     else{
53         PORTB |= 0x20;
54     }
55     }
56 }
57
58 } // loop()

```

2. Assuming the below, write a program using register reads and writes, read the state of the switches and turn the LEDs on if the switches are closed.



```

1 void setup(){
2     DDRB = 0xF0;
3     PORTB = 0x0F;
4 }
5
6 void loop(){
7     //LED0
8     if(PINB & 0x01){
9         PORTB |= 0x10;
10    }
11    else{
12        PORTB &= 0xEF;
13    }
14    //LED1
15    if(PINB & 0x02){
16        PORTB |= 0x20;
17    }
18    else{
19        PORTB &= 0xCF;
20    }
21    //LED2
22    if(PINB & 0x04){
23        PORTB |= 0x40;
24    }
25    else{
26        PORTB &= 0xBF;

```

```

27     }
28     //LED3
29     if(PINB & 0x08){
30         PORTB |= 0x80;
31     }
32     else{
33         PORTB &= 0x7F;
34     }
35 }

```

3. Explain the steps necessary when changing the state of a port from input to output, or vice versa.

```

1 //The following is a step by step for how to go from in to out.
2 PORTD = 0x00;
3 DDRD  = 0xFF;
4 //The following is a step by step for how to go from out to in
5 DDRD  = 0x00;
6 PORTD = 0xFF;

```

The steps stated above are for full ports. The process is similar for single pin assignment.