

```

/* Griffin Mack
11713813

Lab2 - Introduction to Digital Input/Output

*For this lab, user input is taken from an external switch.
Depending on the input,
*the Arduino onboard LED will blink, along with an external LED that
is 180 out of phase.
*
*Additionally, each change of the switch increments a counter that
is displayed in decimal,
*and in hexadecimal on the serial display. This count's LSB will
also be displayed on an
*external 7 segment display.
*
* pinout for LED & 7-segment display      7-segment
display diagram
* arduino      7-segment
*NOTE: pin 1(o) will
*   pin 1      o
only display if current
*   pin 2      a
value is >9 (decimal)
*   pin 3      b
*   pin 4      c
*   pin 5      d
*   pin 6      e
0
*   pin 7      f
*   pin 8      g
*   pin 9      o
*   pin 10     LED(external)
*/
#define LED_PIN 10
#define SWITCH_PIN 11

void setup() {
  /* setup for Serial Monitor and digital I/O pins
  */
  Serial.begin(9600);
  initialize serial:
  pinMode(SWITCH_PIN, INPUT);
  initialize digital pin 11 as input(switch)
  pinMode(LED_BUILTIN, OUTPUT);
  initialize digital pin LED_BUILTIN as output.
  for(int i = 2; i < 11; i++) {
    pinMode(i, OUTPUT);
  }
  initialize digital pins 2-10 as output.
  //
  //
  //
  //

```

```

    }
}

void loop() {
    /* start of program (main loop found below)
    */

    String patterns[16] = {"11111100", "01100000", "11011010",
"11110010",      //array storing patterns for 0-F on 7-segment display
                        "01100110", "10110110", "10111110",
"11100000",      //      EX) Displaying 0:  0 1 2 3 4 5 6 7
                        "11111110", "11110110", "11101111",
"00111111",      //                        a b c d e f g h
                        "10011101", "01111011", "10011111",
"10001111"};      //                        1 1 1 1 1 1 0 0

    // variables used to blink LED's
    unsigned long previous_time = 0;
    unsigned long current_time = 0;
    const long interval = 1000;
    int LED_state = LOW;

    // variables used to monitor switch position
    int switch_state = LOW;
    int previous_switch_state = LOW;
    int switch_counter = 0;

    displayNumber(patterns[0]);                                //
    initializing 7-segment display
    switch_state = digitalRead(SWITCH_PIN);                    //
    initializing switch
    previous_switch_state = switch_state;                       //
    initializing switch previous state

    Serial.println("Toggle switch to begin counting!");        //
    inform the user how to use the counter

    while(1) {
        // update the count if the switch is moved
        if(switch_state != previous_switch_state) {
            switch_counter ++;                                  //
        }
        increment the counter
        displayNumber(patterns[switch_counter % 16]);          //
        display the counter on 7-segment
        serialPrint(switch_counter);                            //
        display the counter on serial monitor
    }
    // if the switch is HIGH, the LED's should be blinking
    if(switch_state == HIGH) {
        current_time = millis();                                //
    }
}

```

```

outputs how long the program has been running
    if (current_time - previous_time >= interval) {    //
check if it has been more than 1s since blinking
    previous_time = current_time;                    //
store time for next pass
    LED_state = blinkLED(LED_state);                //
blink LED and update LED state
    }
}
else{
    digitalWrite(LED_BUILTIN, LOW);                //
reset the builtin LED to low
    digitalWrite(LED_PIN, LOW);                    //
reset the external LED to low

    }
    previous_switch_state = switch_state;            //
store the state of the switch to compare during next run
    switch_state = digitalRead(SWITCH_PIN);
}
}

void displayNumber(String pattern) {
    /* uses the pattern provided to write HIGH or LOW to the
    respective 7-segment pins
    * INPUT:  pattern to print
    * OUTPUT: number displayed on 7-segment display
    */
    for(int i = 0; i < 8; i++) {
        if(pattern[i] == '1') digitalWrite(i+2, HIGH);
        else digitalWrite(i+2, LOW);
    }
}

void serialPrint(int switch_counter) {
    /* prints the current switch counter to the Serial Monitor
    * INPUT:  current counter values
    * OUTPUT: "current count.. decimal: 1    hex: 0x1"
    */
    Serial.print("current count.. decimal: ");
    Serial.print(switch_counter);                    //
display switch_counter in decimal format
    Serial.print("    hex: 0x");
    Serial.println(switch_counter % 16, HEX);          //
display switch_counter in hex (only 0-F)
}

int blinkLED(int LED_state) {
    /* takes current LED state and reverses it(blinks)
    * INPUT:  current LED state (onboard LED)

```

```
* OUTPUT: onboard LED and external LED flipped state
* RETURN: new LED state
*/
if (LED_state == LOW) {
    LED_state = HIGH;
    toggle LED LOW->HIGH
}
else {
    LED_state = LOW;
    toggle LED HIGH->LOW
}
digitalWrite(LED_BUILTIN, LED_state);
digitalWrite(LED_PIN, !LED_state);
external LED is always opposite of onboard LED
return LED_state;
return new LED state
}

//
//
//
//
```