

# Using the Arduino Status Monitor for Debugging and User Interaction

(G Payne – 2016)

## Overview:

The Status Monitor is a tool to interact with the Arduino textually. You can have the Arduino send you messages about the values in variables or text messages about where the sketch is at a point in time. You can also send text values (numbers, letters etc.) TO the Arduino and then have the sketch use those values to control the flow of the program.



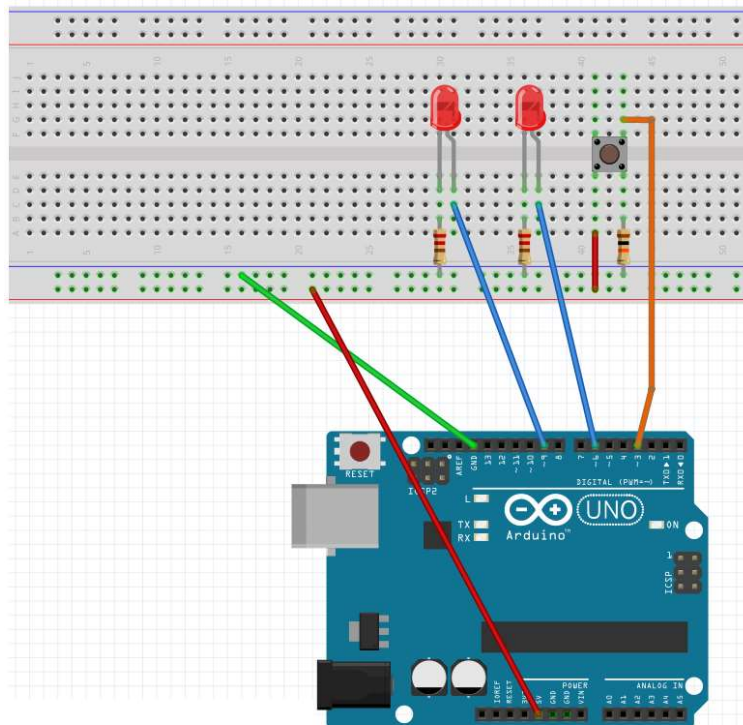
We're going to make a sketch that can light up one or both LEDs based on values you type in the Status Monitor AND it will send a message TO the Monitor to indicate if you've pressed the button.

## Parts Needed:

- Arduino board and breadboard
- 1x momentary contact switch
- 1x 10K resistor
- 2x LEDs,
- 2x 220R or 470R resistors,
- Connecting wires

## Demo:

Connect up the LEDs and button on the breadboard and connect it to the Arduino as shown.



Pin 3 – Button  
Pin 6 – LED 1 +  
Pin 9 – LED 2 +

GND pin goes to  
NEGATIVE rail on  
breadboard

220R or 470R resistors  
on LEDs' NEGATIVE sides

10K resistor on  
Button NEGATIVE side

+5V supply pin goes to  
POSITIVE rail on  
breadboard

In the Arduino IDE, start a new sketch and enter the code below. Save it as 'SerialMonitorDemo'. Upload it to the Arduino board.

```
// Serial Monitor demonstration 2016 - Gordon Payne
// Demonstrates how to send and receive data between the Arduino and the
// connected computer using the Serial Monitor screen
// and serial communications

int btn = 3; // button attached to pin 3
int led1 = 6; // LED 1 on pin 6
int led2 = 9; // LED 2 on pin 9
char reading; // value of character received from Serial Monitor
int btnState = 0; // start with button NOT pressed
long lastClk = millis(); // use for debouncing button
// ie. disregards multiple button pin reads

void setup() {
  pinMode(btn, INPUT);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  Serial.begin(9600); // open communications between Arduino and Serial
  // monitor
  // at 9600 bits-per-second
  digitalWrite(led1, LOW); // make sure both LEDs are OFF
  digitalWrite(led2, LOW);
}

void loop() {
  if(Serial.available()){ // if there is data from the Serial monitor
    //available (ie. a character sent)
    reading = Serial.read(); // read a byte of data

    switch(reading){
    case '1': // if a '1' received
      digitalWrite(led1, HIGH); // turn LED1 ON
      Serial.println("LED 1 ON");
      break;
    case '2': // if a '2' received
      digitalWrite(led2, HIGH); // turn LED2 ON
      Serial.println("LED 2 ON");
      break;
    case '3': // if a '3' received, cycle LEDs
      Serial.println("Cycle LEDs");
      for(int i = 1; i <= 4; i++){ // cycle each LED ON/OFF 4 times
        digitalWrite(led1, HIGH);
        delay(80);
        digitalWrite(led1, LOW);
        delay(80);
        digitalWrite(led2, HIGH);
        delay(80);
        digitalWrite(led2, LOW);
        delay(80);
      }
      break;
    }
  }
}
```

```

case '4': // if a '4' received, turn BOTH LEDs ON
  Serial.println("Cycle LEDs");
  for(int i = 1;i<=4;i++){ // cycle each LED ON/OFF 4 times
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
  }
  break;
case '0': // if a '0' received, turn BOTH LEDs OFF
  digitalWrite(led1,LOW);
  digitalWrite(led2,LOW);
  Serial.println("LEDs OFF");
  break;
default: // disregard all other data sent from Serial Monitor
  break;
}
}
btnState = digitalRead(btn); // get the current state of the button pin
if(millis()-lastClk >150){ // if it's been at least 150 milliseconds since
  //last button pin check
  // filters out multiple button pin reads
  // (like debouncing)
  if (btnState == HIGH){// if button is pressed
    Serial.println("clicked!");
    lastClk = millis(); // reset lastClk to the current time to be ready
    //for next click
  }
}
Serial.print(btnState);
Serial.print("\t"); // '\t' is a TAB feed. Spaces out values across the line
Serial.print("Last button clicked time:");
Serial.print("\t");
Serial.println(lastClk);
delay(150); // slow down sketch for easier viewing on screen
}

```

When you open the Serial Monitor from the Tools menu and enter 1,2,3,4 or 0, or press the button, you should get output like below:

```

COM6
0 Last button clicked time: 7283
0 Last button clicked time: 7283
0 Last button clicked time: 7283
0 Last button clicked time: 7283
LED 1 ON
0 Last button clicked time: 7283
LED 2 ON
0 Last button clicked time: 7283
Cycle LEDs
0 Last button clicked time: 7283
LEDs OFF
0 Last button clicked time: 7283
0 Last button clicked time: 7283
clicked!
1 Last button clicked time: 28576
0 Last button clicked time: 28576
Autoscroll No line ending 9600 baud

```

**Now go out and MAKE SOMETHING AMAZING!!!!**