



VIRTUAL COURSE
**BUILD YOUR OWN
DATA LOGGER**



WILDLABS.NET

[The conservation technology network]

FREAKLABS



MODULE 3-6

INTERRUPTS

LAB A:

PUSHBUTTON INTERRUPTS

Goals

- Learn about pushbuttons and the quirks we need to adjust for in code

Program a pushbutton to turn an LED on and off to learn how to:

- Initialize an interrupt
- Write an interrupt service routine (ISR)
- How to “handle” an interrupt event

Why Is This Important?

- Pushbuttons are a simple way to learn about interrupts
 - Give us full control of when our interrupt event happens (ie. button press)
 - Can easily test our code
 - Gives more confidence in the reliability of our application
- Pushbuttons are useful in all kinds of applications
 - User interface
 - Start/stop of an application, or specific functions
 - Change device settings / parameters

Pushbuttons



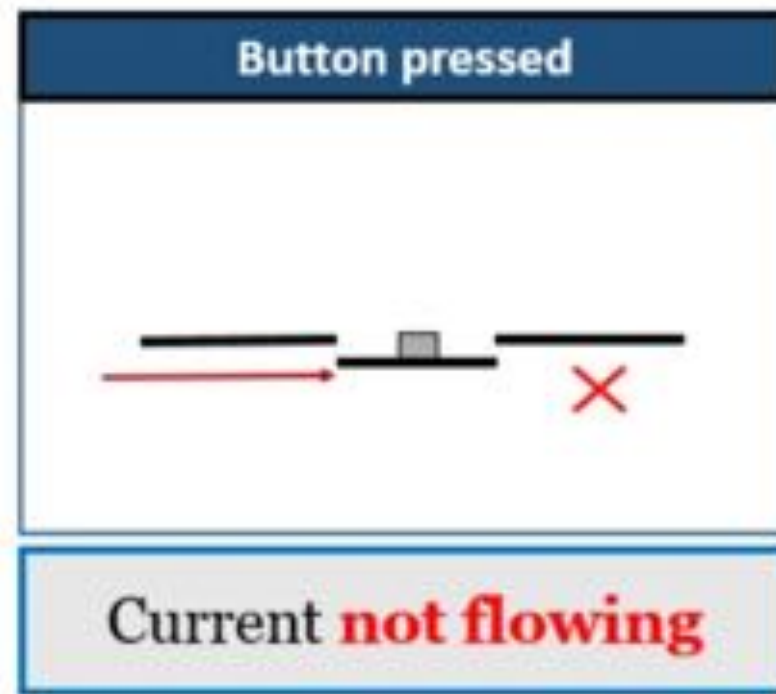
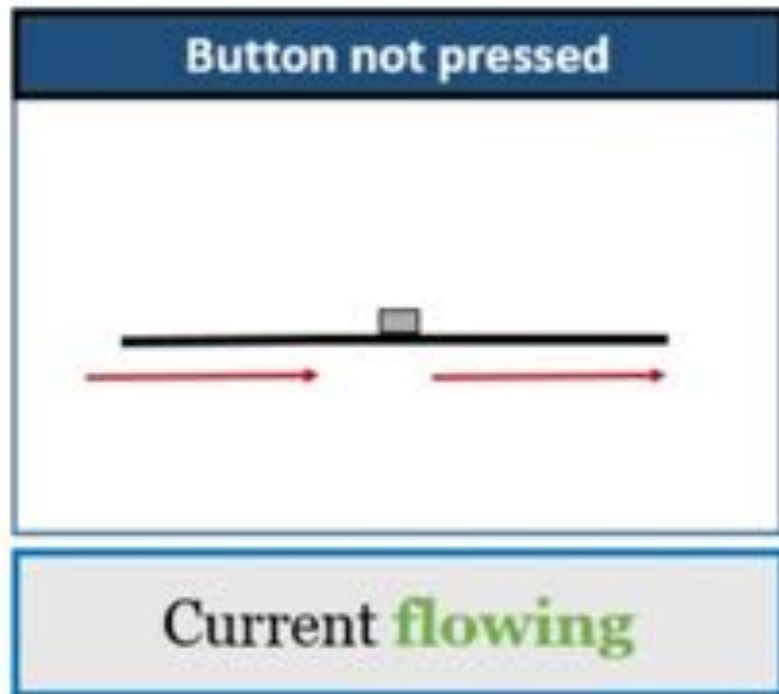
Pushbuttons – Momentary Switches



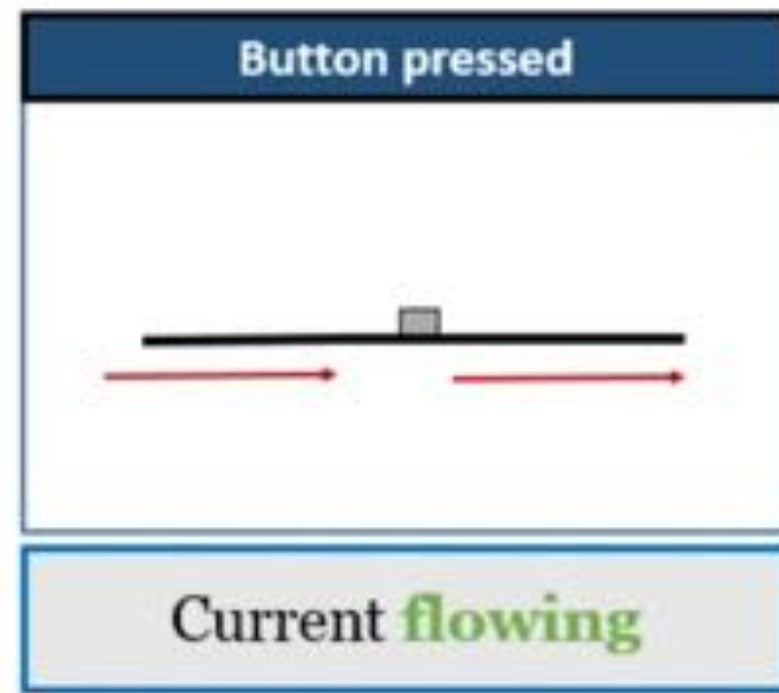
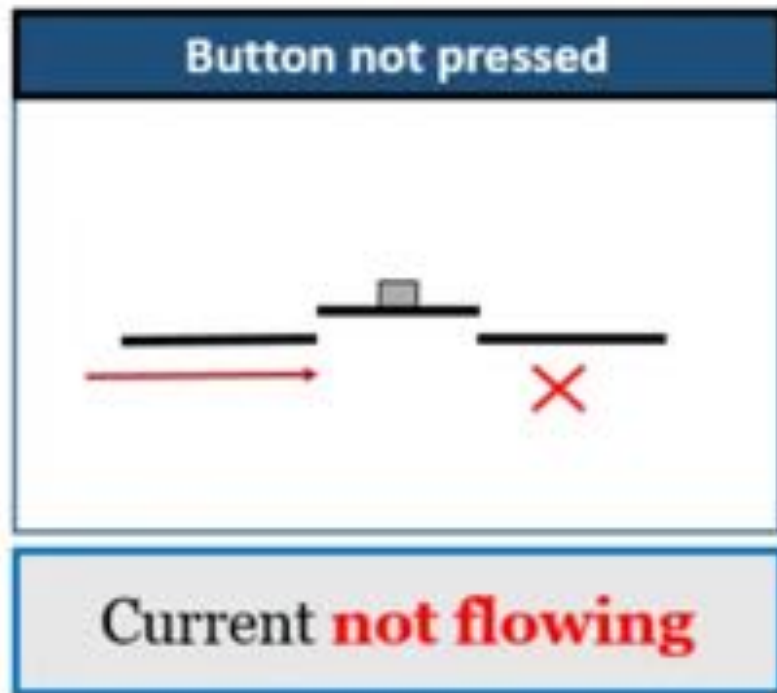
VS



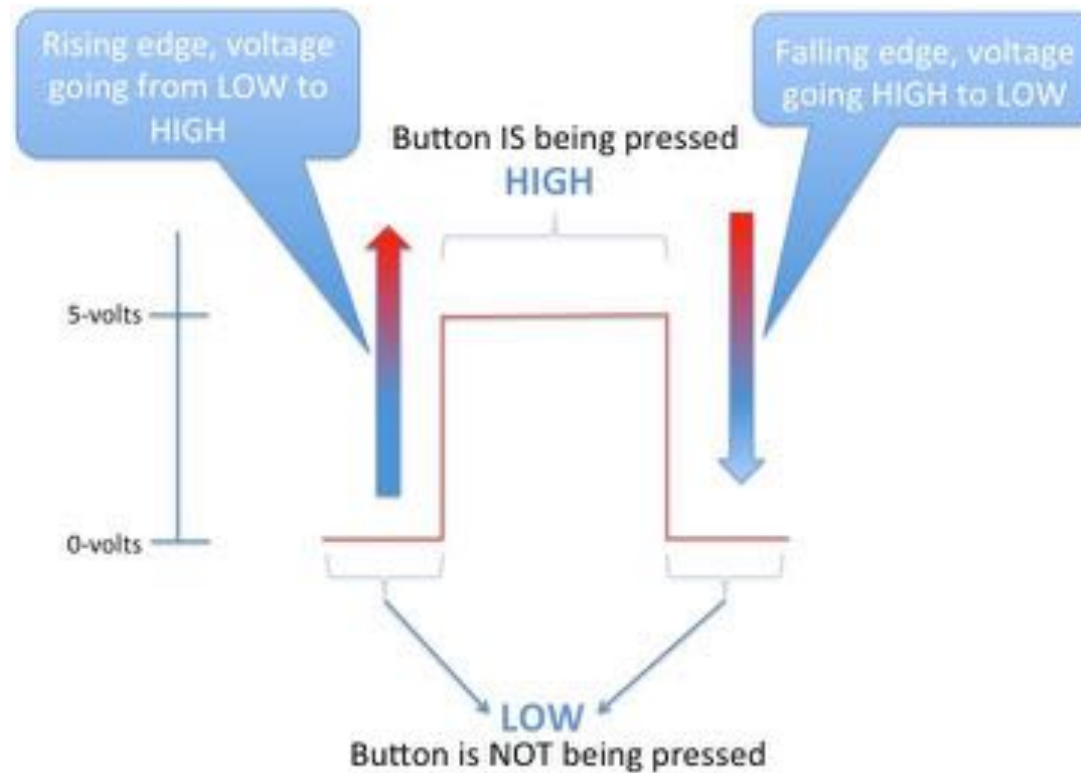
Pushbuttons - Active Low



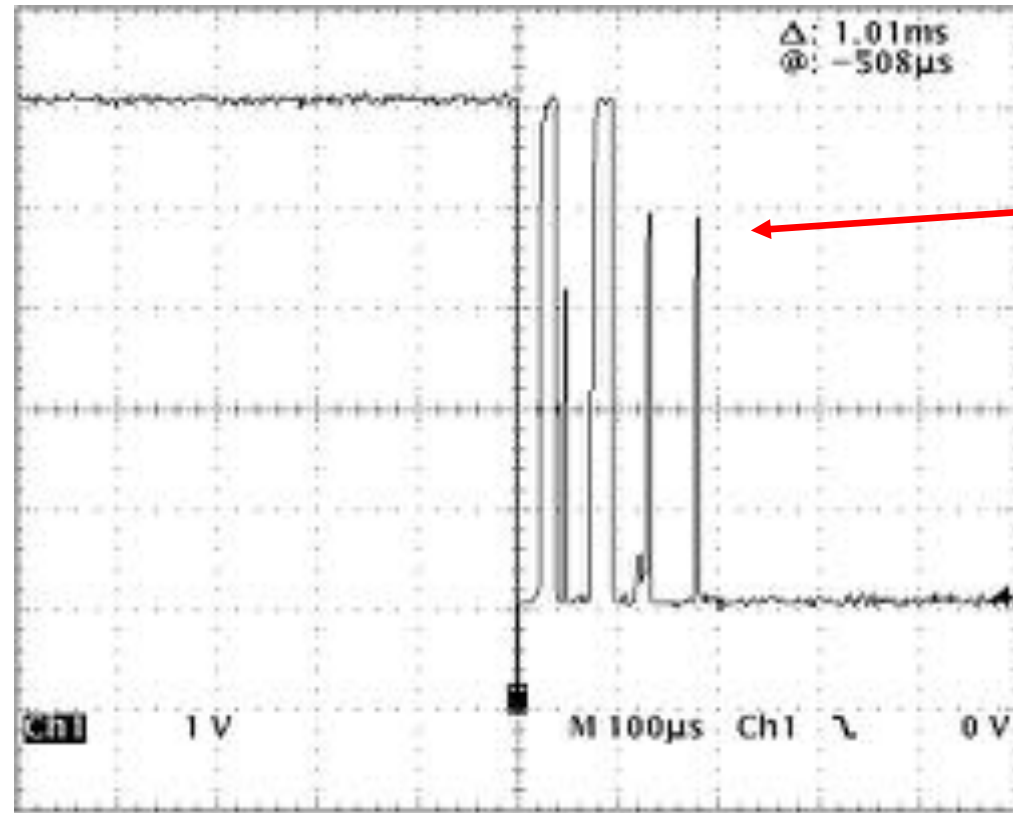
Pushbuttons - Active High



Pushbuttons – Transitions or Edges



Pushbuttons – Debounce Delay



micro connections

IDLE

BOUNCE

PUSHED

Wild Logger Interrupt Mapping

Used in attachInterrupt()

Used to access interrupt pin's properties



Interrupt Source	Interrupt Number	Pin Number
Real Time Clock	0	2
PIR Motion Sensor/Aux Intp	1	3
Pushbutton o	2	6

What Do I Need to Know?

- **attachInterrupt(interruptNum, isrFunction, type)**
 - interruptNum = interrupt number
 - isrFunction = the name of our interrupt service routine function that handles the interrupt
 - type:
 - FALLING = falling edge
 - RISING = rising edge
 - CHANGE = any edge
- **void isrFunction()**
 - Boilerplate function for interrupt service routine
- **volatile int varName**
 - variables used inside ISRs need to be declared as volatile along with their data type
 - This tells the compiler not to delete the variable when optimizing our code

Our Code

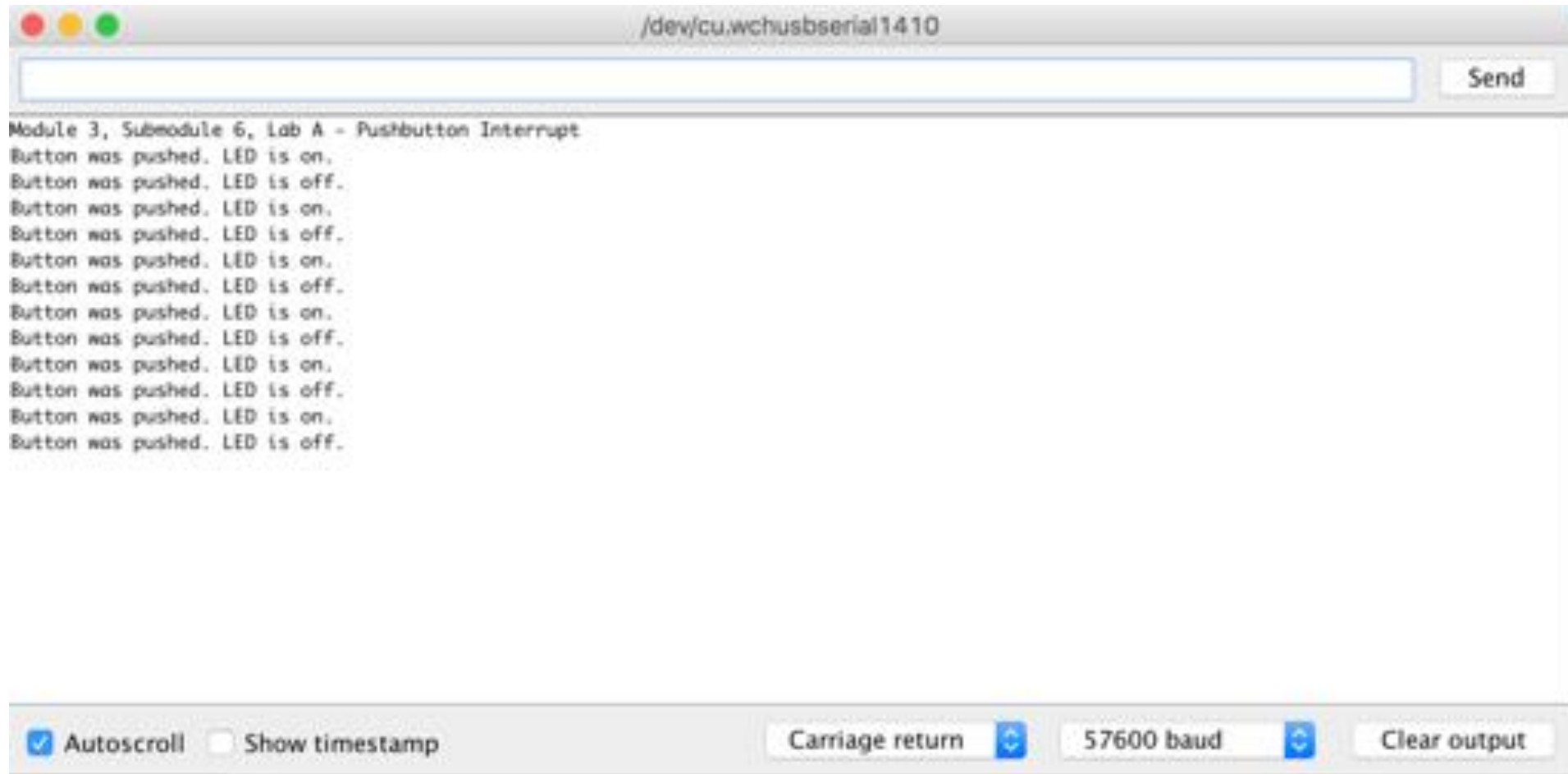
```
Lab3-6-6a-ButtonInp
1 #define DEBOUNCE_TIME 50
2
3 int intpButton = 2;
4 int pinLed = 4;
5 volatile int flagButton;
6
7 void setup()
8 {
9     pinMode(pinLed, OUTPUT);
10    digitalWrite(pinLed, LOW);
11
12    Serial.begin(57600);
13    Serial.println("Module 3, Submodule 6, Lab 6a - Pushbutton Interrupt");
14
15    attachInterrupt(intpButton, isrButton, FALLING);
16 }
```

Continues on next page ...

Our Code

```
3.6.2-LabA §
18
19 void loop()
20 {
21   if (flagButton == 1)
22   {
23     int ledVal = digitalRead(pinLed);
24
25     delay(DEBOUNCE_TIME);
26     flagButton = 0;
27
28     if (ledVal == 0)
29     {
30       Serial.println("Button was pushed. LED is on.");
31       digitalWrite(pinLed, HIGH);
32
33     }
34     else
35     {
36       Serial.println("Button was pushed. LED is off.");
37       digitalWrite(pinLed, LOW);
38     }
39   }
40 }
41
42 }
43
44 void isrButton()
45 {
46   flagButton = 1;
47 }
48
```

Output



The screenshot shows a serial terminal window titled "/dev/cu.wchusbserial1410". The window contains a text area with the following output:

```
Module 3, Submodule 6, Lab A - Pushbutton Interrupt  
Button was pushed. LED is on.  
Button was pushed. LED is off.  
Button was pushed. LED is on.  
Button was pushed. LED is off.  
Button was pushed. LED is on.  
Button was pushed. LED is off.  
Button was pushed. LED is on.  
Button was pushed. LED is off.  
Button was pushed. LED is on.  
Button was pushed. LED is off.  
Button was pushed. LED is on.  
Button was pushed. LED is off.
```

At the bottom of the window, there are several controls: a checked "Autoscroll" checkbox, an unchecked "Show timestamp" checkbox, a "Carriage return" dropdown menu, a "57600 baud" dropdown menu, and a "Clear output" button.

Output –Debounce Delay Value = 0

```
Module 3, Submodule 6, Lab 6a - Pushbutton Interrupt
Delay = 0 seconds.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
```

avrduide: 6414 bytes of flash verified

Output –Debounce Delay Value = 100 ms

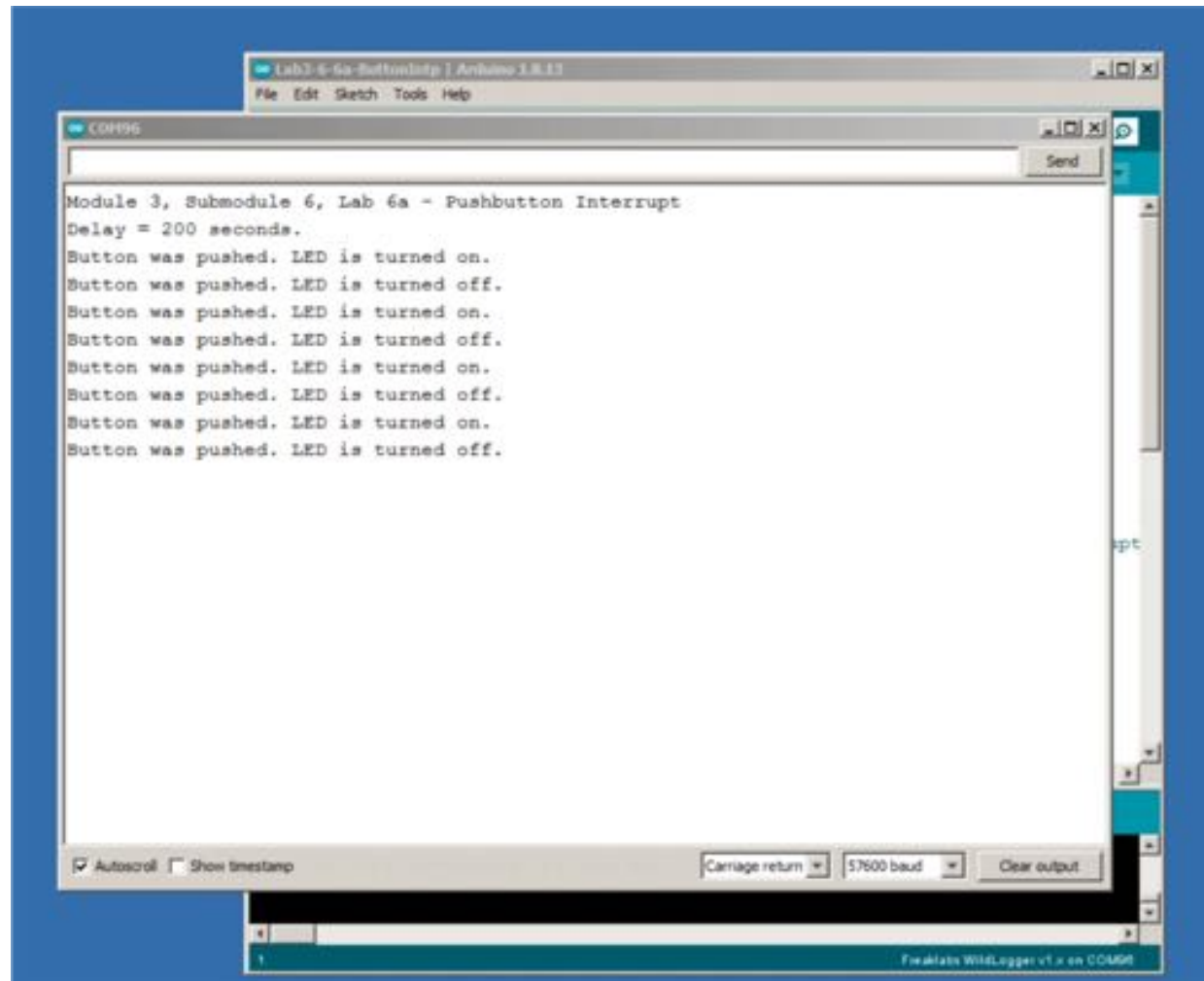
```
lab3-6-6a-ButtonIntp [Arduino 1.8.13]
File Edit Sketch Tools Help

COM9

Module 3, Submodule 6, Lab 6a - Pushbutton Interrupt
Delay = 100 seconds.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.
Button was pushed. LED is turned on.
Button was pushed. LED is turned off.

Autoscroll Show timestamps Carriage return 57600 baud Clear output
FreeLab WinLogger v1.0 on COM9
```

Output –Debounce Delay Value = 200 ms



The screenshot shows the serial monitor window of an Arduino IDE. The window title is "Lab3-6-6a-ButtonIntep | Arduino 1.8.13". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The main text area contains the following output:

```
Module 3, Submodule 6, Lab 6a - Pushbutton Interrupt  
Delay = 200 seconds.  
Button was pushed. LED is turned on.  
Button was pushed. LED is turned off.  
Button was pushed. LED is turned on.  
Button was pushed. LED is turned off.  
Button was pushed. LED is turned on.  
Button was pushed. LED is turned off.  
Button was pushed. LED is turned on.  
Button was pushed. LED is turned off.
```

At the bottom of the window, there are several controls: a checked "Autoscroll" checkbox, an unchecked "Show timestamp" checkbox, a "Carriage return" dropdown menu, a "57600 baud" dropdown menu, and a "Clear output" button. The status bar at the bottom right indicates "Freaklabs WinLogger v1.x on COM6".



COMING UP

Lab 6B:

PIR Motion Sensor Interrupts