Lab 5: Reading a Simple Switch

Objective: Using timing to consistently read a switch/button.

Description of Lab 5:

It should be noted that a simple button is actually two pieces of metal that are either touching or are separated. When the switch is pressed, the two pieces of metal are forced together. The contact of the metal is often not smooth. In fact, during the act of closing or opening the contact a switch will often "bounce" resulting in the kind of voltage waveform as shown in Figure 5-1.

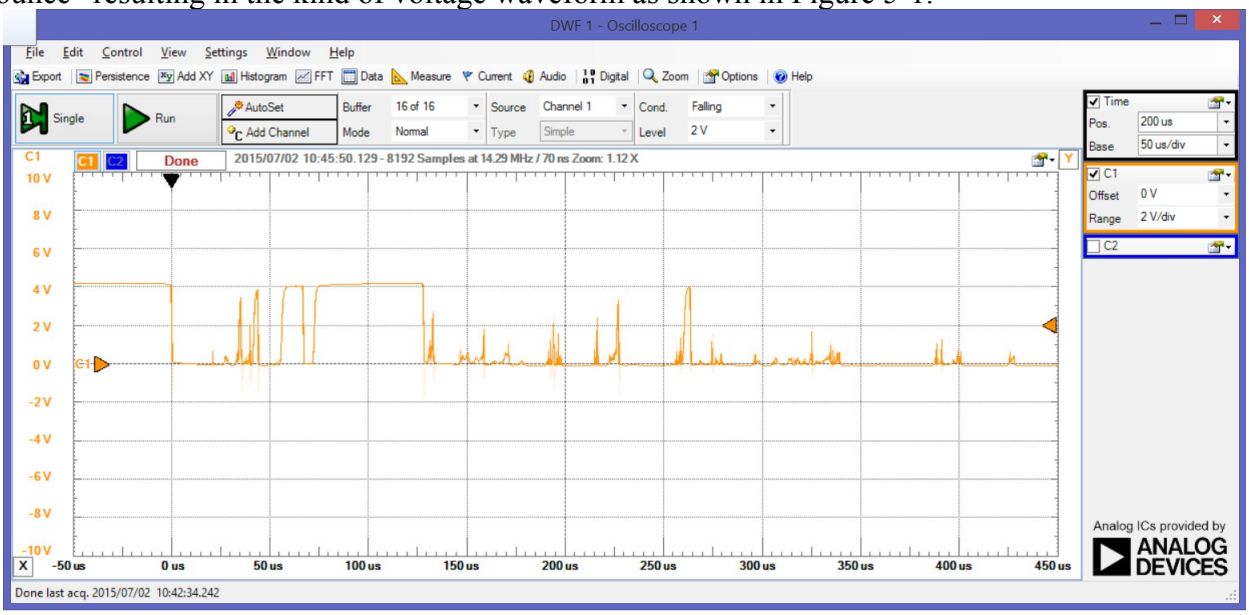

Figure 5-1. Voltage Waveform at Switch Closing.

One way to address the switch bounce, is to add a circuit that will not allow the voltage to change so quickly. However the extra circuitry will cost in board space, production time and expense, and a simple software fix can address this. The software is easily implemented as a state machine, the State Machine Diagram of which is shown in Figure 5-2. ***As a final note, the input line is low when the button is pressed.***

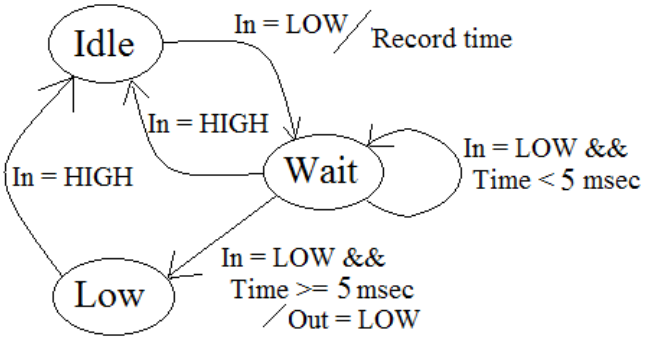

Figure 5-2. State Transition Diagram (STD) for Switch Debounce Process.

ECE 241 **Lab 5** **Fall 2017**
Dates Sept. 26 to Oct 2, 2017 **Points: 20**

Program Structure:
declare ButtonState; // Global variable holding status of button
unsigned long Time; // Holds time to debounce Switch

int ButtonNextState(input) // function that is to be called in loop to service the switch
switch based on ButtonState
Idle: // State where nothing has been happening.
if input is low,
Time = millis(); // Record time of high to low transition.
set ButtonState to Wait // Move to Wait state.
Turn on LED (Pin 13 high)
Wait: // Button has gone low and we are waiting for it to remain low for 5 milliseconds
if input is high, // If button has gone high,
set ButtonState to Idle // Reset back to Idle.
else if (millis()-time >= 5) // if 5 milliseconds has passed.
set ButtonState to Low // Move to low state.
Turn off LED (Pin 13 low.
return 1 // indicating that button has been pressed.
Low: // Button is low and has been so for 5 milliseconds.
if input is high // Once button released.
set ButtonState to Idle:
end of switch
return 0 // By default return 0 indicating nothing is happening.

SetUp: // Function run at the start of the program.
Set pin as input. // Check Schematic in Lab 4 for pin number.
Set ButtonState to Idle. // Initialize state

Loop: // Function continuously called
// Check status of button.
if(ButtonNextState(digitalRead(input))
Send serial message indicating button press // Indicate button has been pressed.
end of if

Lab Assignment:

Prelab: Write the program described in the program section. You are required to use the function call approach described above. You will need to read the button in later labs.

Lab 5:
1) Download and demonstrate the program in the prelab. Your demonstrations should be done such that you verify that the message is only sent when the button is initial pressed.
2) Using the Analog Discovery 2, capture the timing of the debounce operation that will show on the LED (Pin 13).

Question: What was the time you observed on pin 13, and how close to 5 milliseconds was it?