# ReflectionLog: Use Buttons and LEDs

Youdis

```
//Add Phidgets Library
import com.phidget22.*;
public class UseButtonsandLEDs {
//Handle Exceptions
public static void main(String[] args) throws Exc

    //Create
    DigitalInput redButton = new DigitalInput();
    DigitalOutput redLED = new DigitalOutput();
    DigitalInput greenButton = new DigitalInput()
    DigitalOutput greenLED = new DigitalOutput();
    int buttonCount = 0;
    Boolean buttonPressed = true;
    //Address
    redButton.setHubPort(0);
    redButton.setIsHubPortDevice(true);
    redLED.setHubPort(1);
    redLED.setIsHubPortDevice(true);
    greenButton.setHubPort(5);
    greenButton.setIsHubPortDevice(true);
    greenLED.setHubPort(4);
    greenLED.setIsHubPortDevice(true);

    //Open
    redButton.open(1000);
    redLED.open(1000);
    greenButton.open(1000);
    greenLED.open(1000);
```

Adding buttonCount variable to track how many times the buttons are clicked and a debounce variable to prevent the button counter going up repeatedly when a button is being held down..

```
//Use your Phidgets
while(true){

    if( redButton.getState()){
        greenLED.setState(true);
    } else {
        greenLED.setState(false);
    }

    if(greenButton.getState()){
        redLED.setState(true);
    } else {
        redLED.setState(false);
    }
```

Gets state of both buttons(true if button is pressed, false if not) if a button is pressed then the opposite colour's Led will light up if not then that Led will be turned off.

```
}


if((greenButton.getState() || redButton.getState()) && !buttonPressed) {
    buttonCount++;
 System.out.println("Button has been pressed " + buttonCount + " times");
 buttonPressed = true;
}
else if((!greenButton.getState() && !redButton.getState())) {
    buttonPressed = false;
}


Thread.sleep(100);
```

If either button is pressed and the buttonPressed debounce variable is false then it will add one to the button count variable, will print how many times the button has been pressed by printing the button count variable, and turn the buttonPressed variable to true. If the first condition isn't true then it will go to the second one which is if both buttons aren't pressed then the buttonPressed variable will turn to false letting the first condition be able to be true again in another iteration of the loop. Also changed thread.Sleep from 150ms to 100ms to speed up how fast the loop is iterated.