

### Blink LED:

This was a quick and easy assignment. I was able to find out the address of the buttons and LED by looking at the VINT Hub.

```
1 package GettingStarted;
2
3 //Add Phidgets Library | You added a file called phidget22 when config
4 import com.phidget22.*;
5
6 public class BlinkLED {
7     //Handle Exceptions | Exceptions will happen in your code from time
8     public static void main(String[] args) throws Exception{
9
10         //Create | Here you've created a DigitalOutput object for your L
11         DigitalOutput redLED = new DigitalOutput();
12
13         //Address | This tells your program where to find the device you
14         redLED.setHubPort(1);
15         redLED.setIsHubPortDevice(true);
16
17         //Open | Open establishes a connection between your object and y
18         redLED.open(1000);
19
20         //Use your Phidgets | Here is where you can have some fun and us
21         while(true){
22             redLED.setState(true);
23             Thread.sleep(1000);
24             redLED.setState(false);
25             Thread.sleep(1000);
26         }
27     }
28 }
29
```

### Button Events:

This is when I learned about events and how to trigger them. This is the basis for the rest of the projects.

```

1 package GettingStarted;
2 import com.phidget22.*;
3 public class ButtonEvents {
4     //Handle Exceptions
5     public static void main(String[] args) throws Exception {
6
7         //Create
8         DigitalInput redButton = new DigitalInput();
9         DigitalInput greenButton = new DigitalInput();
10
11         //Address
12         redButton.setIsHubPortDevice(true);
13         redButton.setHubPort(0);
14         greenButton.setIsHubPortDevice(true);
15         greenButton.setHubPort(5);
16
17         //Event | Event code runs when data input from the sensor changes. The following
18         greenButton.addStateChangeListener(new DigitalInputStateChangeListener() {
19             public void onStateChange(DigitalInputStateChangeEvent f) {
20                 if (f.getState() == true) {
21                     System.out.println("Green button Pressed");
22                 } else {
23                     System.out.println("Green Button Not Pressed");
24                 }
25             }
26         });
27         redButton.addStateChangeListener(new DigitalInputStateChangeListener() {
28             public void onStateChange(DigitalInputStateChangeEvent e) {
29                 if (e.getState() == true) {
30                     System.out.println("Red button Pressed");
31                 } else {
32                     System.out.println("Red Button Not Pressed");
33                 }
34             }
35         });
36
37         //Open
38         redButton.open(1000);
39         greenButton.open(1000);
40
41         //Keep program running
42         while (true) {
43             Thread.sleep(10);
44         }
45     }
46 }
47
48
49
50

```

### Button LED Events:

In this assignment I had to recreate the tug of war assignment but using events. It was a bit hard at first to get the lights to work but eventually I got it to work.

```

1 package GettingStarted;
2 import com.phidget22.*;
3 public class ButtonLEDEvents {
4
5     //Add Phidgets Library
6
7
8     //Turn on/off LEDs with Global Variables
9     static boolean turnRedLEDon = false;
10    static boolean turnGreenLEDon = false;
11    static int greenpress = 0;
12    static int redpress = 0;
13    static int greenpress1 = 0;
14    static int redpress1 = 0;
15
16    //Handle Exceptions
17    public static void main(String[] args) throws Exception {
18
19        //Create
20        DigitalInput redButton = new DigitalInput();
21        DigitalInput greenButton = new DigitalInput();
22        DigitalOutput redLED = new DigitalOutput();
23        DigitalOutput greenLED = new DigitalOutput();
24
25        //Address
26        redButton.setHubPort(5);
27        redButton.setIsHubPortDevice(true);
28        greenButton.setHubPort(0);
29        greenButton.setIsHubPortDevice(true);
30        redLED.setHubPort(4);
31        redLED.setIsHubPortDevice(true);
32        greenLED.setHubPort(1);
33        greenLED.setIsHubPortDevice(true);
34
35
36
37        //Event | Event code runs when data input from the sensor changes. The following e
38        redButton.addStateChangeListener(new DigitalInputStateChangeListener() {
39            public void onStateChange(DigitalInputStateChangeEvent e) {
40                //Record button state to turn on/off the red LED
41                turnGreenLEDon = e.getState();
42                if (greenpress == greenpress1) {
43                    System.out.println("Green score: " + greenpress);
44                    greenpress++;
45                } else {
46                    greenpress1 = greenpress;
47                }
48            }
49        });
50    }
51 }

```

```

51
52 //Event | Event code runs when data input from the sensor changes. The following event is
53 greenButton.addStateChangeListener(new DigitalInputStateChangeListener() {
54     public void onStateChange(DigitalInputStateChangeEvent e) {
55         //Record button state to turn on/off the green LED
56         turnRedLEDOn = e.getState();
57         if (redpress == redpress1) {
58
59             System.out.println("Red score: " + redpress);
60             redpress++;
61         } else {
62             redpress1 = redpress;
63         }
64     }
65 });
66
67 //Open
68 redLED.open(1000);
69 greenLED.open(1000);
70 redButton.open(1000);
71 greenButton.open(1000);
72
73 //Use your Phidgets | In the button events you recorded the Button State. Here we will use
74 while(true) {
75     //turn red LED on based on red button input
76     redLED.setState(turnRedLEDOn);
77     //turn green LED on based on green button input
78     greenLED.setState(turnGreenLEDOn);
79     //sleep for 150 milliseconds
80     Thread.sleep(10);
81     if (redpress >= 10) {
82         greenLED.setState(turnGreenLEDOn);
83         redLED.setState(turnRedLEDOn);
84         System.out.println("Red Wins");
85         System.exit(0);
86     } else if (greenpress >= 10) {
87         greenLED.setState(turnGreenLEDOn = true);
88         redLED.setState(turnRedLEDOn = true);
89         System.out.println("Green Wins");
90         System.exit(0);
91     }
92 }
93 }
94 }
95 }
96
97

```

### Closing Phidgets:

This taught me the function of closing and opening a fidget and how that could be useful.

```

1 package GettingStarted;
2
3 import com.phidget22.*;
4
5 public class ClosingPhidgets {
6     public static void main(String[] args) throws Exception{
7
8         //Create
9         TemperatureSensor temperatureSensor = new TemperatureSensor();
10
11         //Open
12         temperatureSensor.open(1000);
13
14         //Use your Phidgets
15         System.out.println("Temperature: " + temperatureSensor.getTemperature() + " °C" );
16
17         //Close your Phidgets
18         temperatureSensor.close();
19         temperatureSensor.open(1000);
20         System.out.println("Temperature: " + temperatureSensor.getTemperature() + " °C" );
21     }
22 }
23
24
25

```

### Hot or Cold:

In this project it was already hot enough in the room so i had to change the minimum value in order to make sure the red led worked for this.

```

1 package GettingStarted;
2
3 import com.phidget22.DigitalInput;
4
5
6 public class HotOrCold {
7
8
9     public static void main(String[] args) throws Exception{
10         DigitalInput redButton = new DigitalInput();
11         DigitalOutput redLED = new DigitalOutput();
12         DigitalInput greenButton = new DigitalInput();
13         DigitalOutput greenLED = new DigitalOutput();
14
15         //Address | Address your four objects which lets your program know where to find them.
16         redButton.setHubPort(0);
17         redButton.setIsHubPortDevice(true);
18         redLED.setHubPort(1);
19         redLED.setIsHubPortDevice(true);
20         greenButton.setHubPort(5);
21         greenButton.setIsHubPortDevice(true);
22         greenLED.setHubPort(4);
23         greenLED.setIsHubPortDevice(true);
24
25         //Open | Connect your program to your physical devices.
26         redButton.open(1000);
27         redLED.open(1000);
28         greenButton.open(1000);
29         greenLED.open(1000);
30         //Create | Here you have created a TemperatureSensor object. TemperatureSensor is a class in you
31         TemperatureSensor temperatureSensor = new TemperatureSensor();
32
33         //Open | Open establishes a connection between your object and your physical Phidget. You provid
34         temperatureSensor.open(1000);
35
36         //Use your Phidgets | This code will print the temperature every 150ms
37         while (true) {
38             System.out.println("Temperature: " + temperatureSensor.getTemperature() + " °C");
39             if (temperatureSensor.getTemperature() >= 20 && temperatureSensor.getTemperature() <= 24){
40                 redLED.setState(false);
41                 greenLED.setState(true);
42             } else {
43                 greenLED.setState(false);
44                 redLED.setState(true);
45             }
46             Thread.sleep(150);
47         }
48     }
49 }

```

## LED Brightness:

It was fun using the loop to increase and decrease the brightness. This seems like it will be a useful function for making an alarm.

```

1 package GettingStarted;
2
3
4 //Add Phidgets Library
5 import com.phidget22.*;
6
7 public class LEDBrightness {
8     //Handle Exceptions
9     public static void main(String[] args) throws Exception{
10
11         //Create
12         DigitalOutput redLED = new DigitalOutput();
13
14         //Address
15         redLED.setHubPort(1);
16         redLED.setIsHubPortDevice(true);
17
18         //Open
19         redLED.open(1000);
20         double i = 0;
21         boolean rev = false;
22         //Use your Phidgets with Duty Cycle | Duty Cycle controls 1
23         while(true) {
24             if (rev == false){
25                 i++;
26             } else if (i > 0){
27                 i--;
28             } else {
29                 System.exit(0);
30             }
31             if (i >= 10) {
32                 rev = true;
33             }
34             double d = i / 10;
35             redLED.setDutyCycle(d);
36             Thread.sleep(100);
37         }
38     }
39 }
40 }
41
42

```

### Read button:

This taught me the importance of button inputs and how that can be useful

```

1 package GettingStarted;
2
3
4 //Add Phidgets Library | You added a file called phidget22 when configuring
5 import com.phidget22.*;
6
7 public class ReadButton {
8     //Handle Exceptions | Exceptions will happen in your code from time to ti
9     public static void main(String[] args) throws Exception{
10         boolean button = false;
11         //Create | Here you've created a DigitalInput object for your button.
12         DigitalInput redButton = new DigitalInput();
13
14         //Address | This tells your program where to find the device you want
15         redButton.setHubPort(5);
16         redButton.setIsHubPortDevice(true);
17
18         //Open | Open establishes a connection between your object and your p
19         redButton.open(1000);
20
21         //Use your Phidgets | Here is where you use your Phidgets! This code
22         while(true){
23             if (redButton.getState() == button){
24                 System.out.println("Button State: " + redButton.getState());
25                 Thread.sleep(150);
26                 button = !button;
27             }
28         }
29     }
30 }
31 }
32
33

```

### Read Humidity:

It was hard work getting the humidity high enough to not be too low but eventually I got it to work.



```

1 package GettingStarted;
2
3
4 //Add Phidgets Library | You added a file called phidget22 when configuring your project.
5 import com.phidget22.*;
6
7 public class ReadHumidity {
8     public static void main(String[] args) throws Exception{
9
10         //Create | Here you've created a HumiditySensor and a TemperatureSensor object for y
11         HumiditySensor humiditySensor = new HumiditySensor();
12         TemperatureSensor temperatureSensor = new TemperatureSensor();
13
14         //Open | Open establishes a connection between your object and your physical Phidget
15         humiditySensor.open(1000);
16         temperatureSensor.open(1000);
17
18         //Use your Phidgets | This code will print humidity and temperature read by the sens
19         while(true){
20             if (humiditySensor.getHumidity() >= 30) {
21                 System.out.print("Humidity: " + humiditySensor.getHumidity() + " %RH, ");
22             } else {
23                 System.out.print("Humidity: Humidity is low,");
24             }
25             if (temperatureSensor.getTemperature() >= 21) {
26                 System.out.println(" Temperature: " + temperatureSensor.getTemperature() + "
27             } else {
28                 System.out.println(" Temperature: Room is too cold " );
29             }
30             Thread.sleep(150);
31         }
32     }
33 }
34

```

## Read Temperature:

It was fun using the code to convert celsius to fahrenheit but the fahrenheit sometimes had .99999 so I used decimal format to round it down.

```

1 package GettingStarted;
2
3 import java.text.DecimalFormat;
4
5 public class ReadTempature {
6     private static final DecimalFormat df = new DecimalFormat("0.00");
7     public static void main(String[] args) throws Exception{
8
9         //Create | Here you have created a TemperatureSensor object. TemperatureSensor is a class in your Phidgets library
10        TemperatureSensor temperatureSensor = new TemperatureSensor();
11
12        //Open | Open establishes a connection between your object and your physical Phidget. You provide a timeout value o
13        temperatureSensor.open(1000);
14
15        //Use your Phidgets | This code will print the temperature every 150ms
16        while (true) {
17            System.out.println("Temperature: " + temperatureSensor.getTemperature() + " °C" );
18            System.out.println("Temperature: " + df.format((temperatureSensor.getTemperature() * 1.8) + 32)) + " °F" );
19            Thread.sleep(150);
20        }
21    }
22 }
23
24
25
26
27

```

## Smart Phidget Events:

This was just a combination of the temperature and humidity assignment but instead using events.

```
1 package GettingStarted;
2
3
4 //Add Phidgets Library
5 import com.phidget22.*;
6
7 public class SmartPhidgetEvents {
8
9     public static void main(String[] args) throws Exception {
10
11         //Create
12         HumiditySensor humiditySensor = new HumiditySensor();
13         TemperatureSensor temperatureSensor = new TemperatureSensor();
14
15         //Humidity Event | Event code runs when data input from the sensor changes. The following event is a
16         humiditySensor.addHumidityChangeListener(new HumiditySensorHumidityChangeListener() {
17             public void onHumidityChange(HumiditySensorHumidityChangeEvent e) {
18                 System.out.println("Humidity: " + e.getHumidity() + "%RH");
19             }
20         });
21
22         //Temperature Event | Event code runs when data input from the sensor changes. The following event i
23         temperatureSensor.addTemperatureChangeListener(new TemperatureSensorTemperatureChangeListener() {
24             public void onTemperatureChange(TemperatureSensorTemperatureChangeEvent e) {
25                 if (e.getTemperature() >= 21) {
26                     System.out.println("Temperature: " + e.getTemperature() + "°C");
27                 } else {
28                     System.out.println("Room is too cold");
29                 }
30             }
31         });
32
33     };
34
35     //Open
36     humiditySensor.open(1000);
37     temperatureSensor.open(1000);
38
39     //Keep program running
40     while (true) {
41         Thread.sleep(150);
42     }
43 }
44 }
45 }
```

## Thermostat:

This was a fun and challenging assignment. I had to really use my brain to make the thermostat that works.

```

1 package GettingStarted;
2
3 import com.phidget22.DigitalInput;
4
5
6
7 public class Thermostat {
8
9
10 public static void main(String[] args) throws Exception{
11     DigitalInput redButton = new DigitalInput();
12     DigitalOutput redLED = new DigitalOutput();
13     DigitalInput greenButton = new DigitalInput();
14     DigitalOutput greenLED = new DigitalOutput();
15
16     //Address | Address your four objects which lets your program know where to find them.
17     redButton.setHubPort(0);
18     redButton.setIsHubPortDevice(true);
19     redLED.setHubPort(1);
20     redLED.setIsHubPortDevice(true);
21     greenButton.setHubPort(5);
22     greenButton.setIsHubPortDevice(true);
23     greenLED.setHubPort(4);
24     greenLED.setIsHubPortDevice(true);
25
26     //Open | Connect your program to your physical devices.
27     redButton.open(1000);
28     redLED.open(1000);
29     greenButton.open(1000);
30     greenLED.open(1000);
31     //Create | Here you have created a TemperatureSensor object. TemperatureSensor is a class in your Phidgets library that
32     TemperatureSensor temperatureSensor = new TemperatureSensor();
33
34     //Open | Open establishes a connection between your object and your physical Phidget. You provide a timeout value of 100
35     temperatureSensor.open(1000);
36     int settemp = 21;
37     int seconds = 0;
38     int pressed = settemp;
39     //Use your Phidgets | This code will print the temperature every 150ms
40     while (true) {
41         if (seconds >= 100) {
42             System.out.println("Temperature: " + temperatureSensor.getTemperature() + " °C");
43             System.out.println("Set tempature is " + settemp);
44             seconds = 0;
45         }
46
47         if (temperatureSensor.getTemperature() >= (settemp - 2) && temperatureSensor.getTemperature() <= (settemp + 2)){
48             redLED.setState(false);
49             greenLED.setState(true);
50
51         } else {
52             greenLED.setState(false);
53             redLED.setState(true);
54         }
55
56         if( greenButton.getState()){
57             if (pressed == settemp) {
58                 settemp++;
59             }
60         } else {
61             if (greenButton.getState() == false) {
62                 pressed = settemp;
63             }
64         }
65
66         if(redButton.getState()){
67             if (pressed == settemp) {
68                 settemp--;
69             }
70         } else {
71             pressed = settemp;
72         }
73         Thread.sleep(100);
74         seconds++;
75     }
76 }
77 }
78

```

## Tug of War:

This was an assignment where you could actually make a game which was very fun. There was a problem with a delay in counting each button press but I just reduced the sleep time and it started working perfectly.

```

1 package GettingStarted;
2
3 import com.phidget22.*;
4
5 public class TugOfWar {
6     public static void main(String[] args) throws Exception{
7
8         //Create | Create objects for your buttons and LEDs.
9         DigitalInput redButton = new DigitalInput();
10        DigitalOutput redLED = new DigitalOutput();
11        DigitalInput greenButton = new DigitalInput();
12        DigitalOutput greenLED = new DigitalOutput();
13
14        //Address | Address your four objects which lets your program know where to find them.
15        redButton.setHubPort(0);
16        redButton.setIsHubPortDevice(true);
17        redLED.setHubPort(1);
18        redLED.setIsHubPortDevice(true);
19        greenButton.setHubPort(5);
20        greenButton.setIsHubPortDevice(true);
21        greenLED.setHubPort(4);
22        greenLED.setIsHubPortDevice(true);
23
24        //Open | Connect your program to your physical devices.
25        redButton.open(1000);
26        redLED.open(1000);
27        greenButton.open(1000);
28        greenLED.open(1000);
29        int presses1 = 0;
30        int presses2 = 0;
31        int pressed1 = 0;
32        int pressed2 = 0;
33        //Use your Phidgets | This code will turn on the LED when the matching button is pressed and t
34        while(true){
35
36            if( greenButton.getState()){
37                if (pressed1 == presses1) {
38                    presses1++;
39                    System.out.println("Green Presses: " +presses1);
40                    greenLED.setState(true);
41                }
42
43            } else {
44                if (greenButton.getState() == false) {
45                    greenLED.setState(false);
46                    pressed1 = presses1;
47                }
48            }
49        }
50

```

```

51         if(redButton.getState()){
52             if (pressed2 == presses2) {
53                 presses2++;
54                 System.out.println("Red Presses: " + presses2);
55                 redLED.setState(true);
56             }
57         } else {
58             redLED.setState(false);
59             pressed2 = presses2;
60         }
61         if (presses1 >= 10) {
62             System.out.println("Green Wins");
63             System.exit(0);
64         }
65         if (presses2 >= 10) {
66             System.out.println("Red Wins");
67             System.exit(0);
68         }
69         Thread.sleep(10);
70     }
71 }
72 }
73 }
74 }
75 }

```

## Use Buttons And LEDs

This was a combination of previous button and LED assignments and it came together perfectly.

```

1 package GettingStarted;
2
3 //Add Phidgets Library | You added a file called phidget22 when configuring your project. Import gives
4 import com.phidget22.*;
5
6 public class UseButtonsaAndLEDs {
7 //Handle Exceptions | Exceptions will happen in your code from time to time. These are caused by unexpe
8 public static void main(String[] args) throws Exception{
9
10 //Create | Create objects for your buttons and LEDs.
11 DigitalInput redButton = new DigitalInput();
12 DigitalOutput redLED = new DigitalOutput();
13 DigitalInput greenButton = new DigitalInput();
14 DigitalOutput greenLED = new DigitalOutput();
15
16 //Address | Address your four objects which lets your program know where to find them.
17 redButton.setHubPort(0);
18 redButton.setIsHubPortDevice(true);
19 redLED.setHubPort(1);
20 redLED.setIsHubPortDevice(true);
21 greenButton.setHubPort(5);
22 greenButton.setIsHubPortDevice(true);
23 greenLED.setHubPort(4);
24 greenLED.setIsHubPortDevice(true);
25
26 //Open | Connect your program to your physical devices.
27 redButton.open(1000);
28 redLED.open(1000);
29 greenButton.open(1000);
30 greenLED.open(1000);
31 int presses = 0;
32 int pressed = 0;
33 //Use your Phidgets | This code will turn on the LED when the matching button is pressed and turns
34 while(true){
35
36     if( redButton.getState()){
37         if (pressed == presses) {
38             presses++;
39             System.out.println(presses);
40             greenLED.setState(true);
41         }
42
43     } else {
44         if (greenButton.getState() == false) {
45             greenLED.setState(false);
46             pressed = presses;
47         }
48     }
49 }
50
51 if(greenButton.getState()){
52     if (pressed == presses) {
53         presses++;
54         System.out.println(presses);
55         redLED.setState(true);
56     }
57 } else {
58     if (redButton.getState() == false) {
59         redLED.setState(false);
60         pressed = presses;
61     }
62 }
63
64 }
65
66 Thread.sleep(150);
67 }
68 }
69 }
70

```