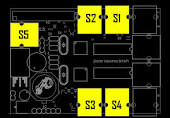
**Introduction To Sensors.**

You have built cars that preform predefined task regardless of outside circumstances, which you most likely have found to be a little frustrating. It is now time to start importing real world data into your programs, in order to alter the way your program functions. To do this you will be using wonderful things called sensors. Sensors gather outside analog data such as sound, light, color, pressure, distance and turns it into digital data that can be imported into programs to affect the way they work.

Introduction to BrickPi sensors.

If you remember when the BrickPi was first introduced, it was mention that it had 4 motor ports and 5 sensor ports. Well now it is time to use the sensor ports, below you will find a diagram disclosing the locations of the ports.

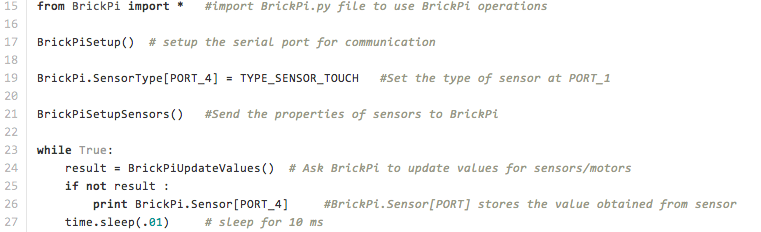


For the most part you will use just ports 1-4, port 5 is reserved for other types of sensors that require more advance coding.

We have 5 sensors that you will be able to use in your robots, and in this lab you will learn how to use each one of them, they are: Sound, Light, Ultrasonic, Touch and Color. However before jumping into using the sensors, there are a few common traits that are the same regardless of what sensors you are using.

Common Code For Sensors (except color sensors because they are special).

Take a look at the code below. This code is the standard example code for running a touch sensor. Now with this code we will do what we have done in the past and break down the important commands to us.



**from BrickPi import \*** and **BrickPiSetup()** : Just like the motors these need to be at the start of every program you write for the BrickPi.

**BrickPi.SensorType[PORT\_1] = TYPE\_SENSOR\_... :** So this is similar to when you assign a motor to a port, however there is one key difference. So just like the motors you can change out the “1” for any integer between 1-4 (NOT 5). The big difference between the sensors and the motors is the last part of the command. You must insert which type of sensor you are using where the “…” is. The 5 commands are below.

**TOUCH:** is of course for touch sensors

**ULTRASONIC\_CONT :** is for ultrasonic

**RAW :** is for the light sensor and the sound. This command will give you the raw analog data value form the sensor. It is up to you to write code that can analyze what this data means. To do so you will need to know how to program ranges in python.

**BrickPiSetupSensors() :** This just like the motors, you must make sure that it comes after you have assigned all your sensors and motors to their respective ports.

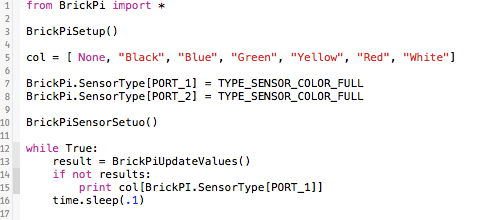
**print BrickPi.Sensor[PORT\_1] :** will print the data that is gathered from your sensors.

So now that you have the a basic understanding of how the sensor code work, I suggest you take the time to test out all the sensors except the color, which we will look at next.

Using The “Special” Color Sensors.

The color sensor has a few bugs still in the software that hopefully have been fix by the time you are doing this lab. The instructor will inform you on rather or not they have been fixed. This is one of the reasons why we are looking at it by itself.

Take a look at the example code for the color sensors provided by Dexter ind.



This code is very similar to the Touch sensor with a few very important exceptions, in lines 5 and 15. Line 15 is necessary to use the color sensor, because it provides the information to interpret the raw data form the sensor. Line 24, is only slightly different. Before the BrickPi.Sensor[PORT\_1] you must include the col[….] , this directs the raw data to the interpreting command. Other then these two command the programing is the same as the other sensors. However, there are a few hardware bugs.

The main bug is that 1 color sensor takes up two ports. In other words in order for 1 color sensor to work you must assign it to two ports. Take the above code for example, in lines 7 & 8 I assign color sensors to ports 1 & 2. The result of the above code is a working color sensor in port 1 and only port 1. The draw back of this bug is that you can’t assign port 2 to any other sensor; it must remain empty in order for the color sensor to work. This affectively limits the number of color sensors you can have down to 2 and they would be your only sensors. One of the sensors would be in port 1 the other in port 3.

Now that you know how to work the color sensors, it is time for you to write a program that takes readings form two color sensors simultaneously.

**Lab 8** Put The Up, part 2.

Remember that little Lego gun that you made earlier; let’s now turn it into an auto sentry. You are to modify it so that it will shoot whenever someone walks in front of it, and only when someone is in front of it.

**Lab 9** Clap on, Clap off.

For this lab you are going to add sensors to one of your cars in order that external things can affect the code and determine what it does. First you are to add a touch sensor to your car that when pressed will stop the car. Next you will add a sound sensor, so that when a loud sound (like clapping) will cause it to go in reverse and when you clap for a second time it will return to forward. The lab report for this lab is mostly the software so make sure you include that and the rest is going to be judge on rather or not it works.

**Lab 10** Follow Me.

You need to design and build a robot that can follow a vertical piece of cardboard around the room. The cardboard can move in a straight line or take turns; therefore you must program your robot to respond to these changes in parameters. You may use any sensors you wish.

**Lab 11** The Track.

Build a robot that can follow the black track around the room. Make sure that it can go in both directions. You may use any sensors you wish.