

Sensors Write Up

An Introduction to EV3 Sensors

This guide serves as a basic introduction to EV3 sensors and some common ways to use them.

I. Basics of Using a Sensor

A sensor is something that measures some specific value, whether it's distance, reflected light intensity, angle, etc. However, the blocks below [In Figure 1] will **not** save whatever value(s) they read. You would have to add more programming blocks to make it useful. Fortunately, Blocks that incorporate sensors in useful ways already exist. They can be found in Figure 2.

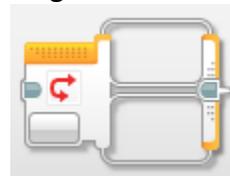


Figure 1

The two programming blocks which are crucial for using sensors:



Loops



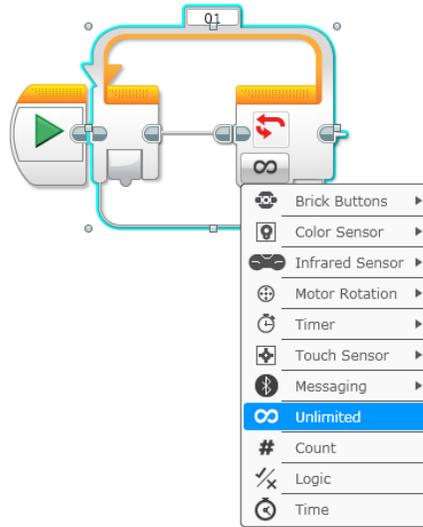
Switches

Figure 2

a. Loops

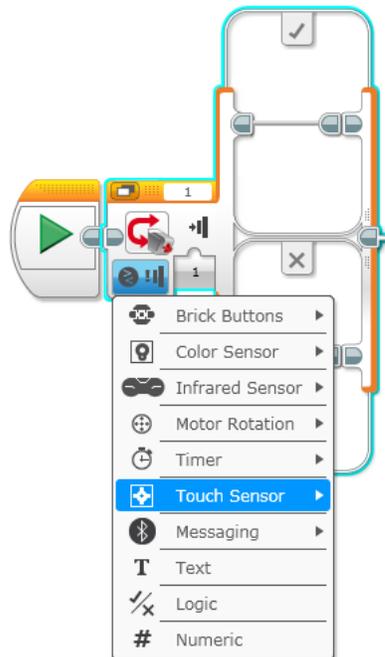
An important thing to remember with EV3 programming is that unless the code is inside a loop, it will only run once. So to get repetitive actions, a loop is required. You can put any piece of code inside a loop, including another loop.

The most important aspect of a loop is knowing the condition that determines when the loop ends, as shown in the image below. This is where the sensor comes into play. What you are trying to accomplish will change what you want the condition to be—this is something to experiment with.



b. Switches

Switches are useful when you would like a robot to switch between doing Action 1 and Action 2 depending on the situation; it becomes a fork in the road of your code. Like loops, the part that is important with sensors is the switch condition which can be a variety of things. But be careful, a switch on its own will only run once. If you'd want it to run multiple times, you'd need to put it inside a loop.

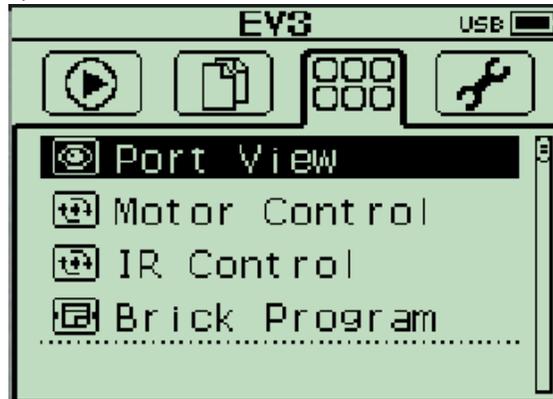


c. Port View

In using switches or loops with a sensor, you need to know what values to use as the different conditions for either exiting a loop or switching between actions in a switch. You

can either guess and check these values or you can use port view—a way to see exactly what the values are for something for any sensor/motor.

You can access port view directly on your EV3 from the main screen by going to the 3rd tab and clicking “port view”, as shown below:



Now you can use port view to get the exact real-time readings of your connected sensors.

II. Types of Sensors

a. Light Sensor

The light sensor on an EV3 can measure three things: color, reflected light, and ambient light. If you are measuring color, the sensor outputs a value (0 – 7) that corresponds to a specific color. Measuring reflected light outputs a value (0 – 100) depending on the intensity of the light that is reflected. The ambient light works the same way as the reflected light, except it just measures the surrounding light.

When using the sensor in a loop or switch, the change condition would be a specific value that the sensor can read. If it is below a value, it will do Action A, if it is above it will do Action B or break the loop depending on if you use a switch or a loop.

The light sensor can be a bit tricky sometimes, if the robot goes a bit too slowly, it can get stuck oscillating between two values and the code won't progress as expected. If the robot moves too quickly, it can sometimes not read a value fast enough and actually miss an exact point. So be sure to keep this in mind when programming and designing your robot.

b. Gyro Sensor

The gyro sensor is a very useful sensor. It can measure the exact degree it turns or the rate at which it is turning. An illustration for how useful it is: if you'd like to make 90 degree turns every time but the route is rocky, the gyro sensor can make sure a robot turns 90 every time, regardless if it catches or skips. Something to keep in mind, the gyro can do either positive or negative values, so use port view to determine the values you need to use in your program

The gyro sensor, if you'd like it to make individual 90 degree turns (90, 90, 90 and not 90, 180, 270) you would need to reset it, so it has a new zero value. There is a simple block for this, however, the actual resetting takes a full 1 – 2 seconds and your robot needs to be completely still while this happens so it has an accurate zero value.

The gyro sensor is a new sensor and is not free of glitches though. Every once in a while something goes a bit haywire internally with the sensor and the sensor will think it is

being rotated rapidly and won't stop. So it will start outputting very high sensor values and you'll find your program will all of a sudden stop working for no reason. The fix for this is simply plugging the gyro sensor out and plugging it back in; this resets the gyro sensor and resolves the glitch.

c. Ultrasonic Sensor

The ultrasonic sensor is a fairly straight forward sensor. The sensor can measure the distance to the nearest object in either centimeters or inches. The sensor does tend to fail when objects are too close or too far away. The recommended range is one cm to 250 cm.

d. Touch Sensor

The touch sensor is a simpler one as well. It has 3 states that you can check for: unpressed, pressed, pressed and then released. So depending on what you would like your robot to do, you can use one of these states in your programming.

e. Motors

Motors aren't sensors, but they do deserve a small section as they can be used like a sensor with loops. Motors can be told go reverse or forward. You can either do this with controlling what direction the motor spins in or with making the power positive or negative.

With moving motors forward you can do so either with seconds, rotations, or degrees. Seconds will turn the motors for X amount of seconds, it won't check to see that the motors have turned the same amount or have actually turned at all, they will just run the code for that amount of time. Rotations will make sure that the motor(s) have turned the correct amount before continuing with the code. If one or more of the motors can't physically complete the rotation the program will get stuck and cannot continue onwards until the rotation is completed. Degrees behaves exactly like rotation, it just uses degrees as the measurement instead.