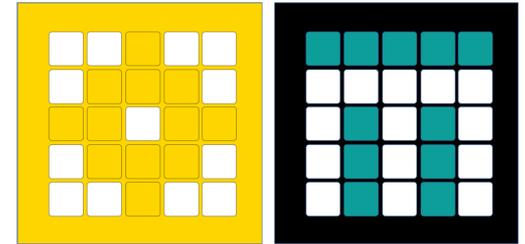


# PRIME LESSONS

By the Makers of EV3Lessons



## VIEWING SENSOR VALUES

BY SANJAY AND ARVIND SESHAN

This lesson uses SPIKE 3 software

# LESSON OBJECTIVES

Learn how to view sensor values on SPIKE 3

Learn how the sensor values can **differ** from the values returned by the SP3 Python API, and how to adjust for those differences.



# WHY DO YOU NEED SENSOR DATA?

Sensor data can be....

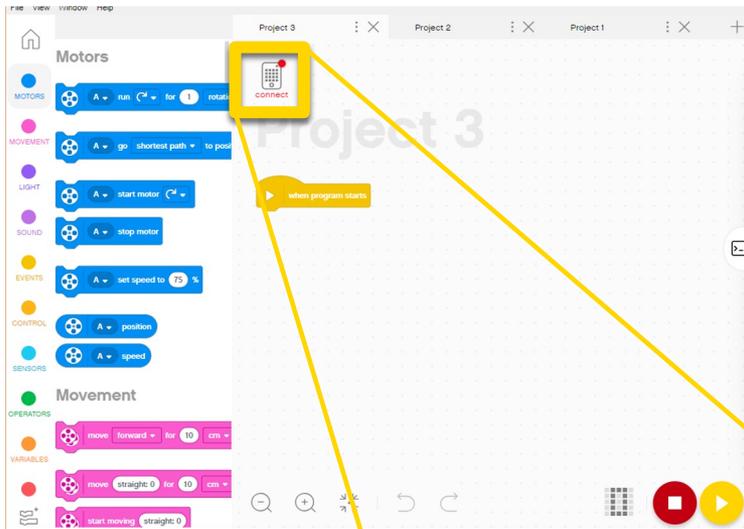
- Used to help program more easily (no more guess and check!!)

- Used to help program more accurately

- Used to debug code as well as build issues

SPIKE Prime has no screen, but you can still view sensor data using the Hub Dashboard

# YOU MUST BE CONNECTED TO HUB



To view any sensor data, you must first be connected to the Hub (SPIKE Prime or Robot Inventor)

Click on the small Hub icon in a Project in SPIKE Prime or Robot Inventor



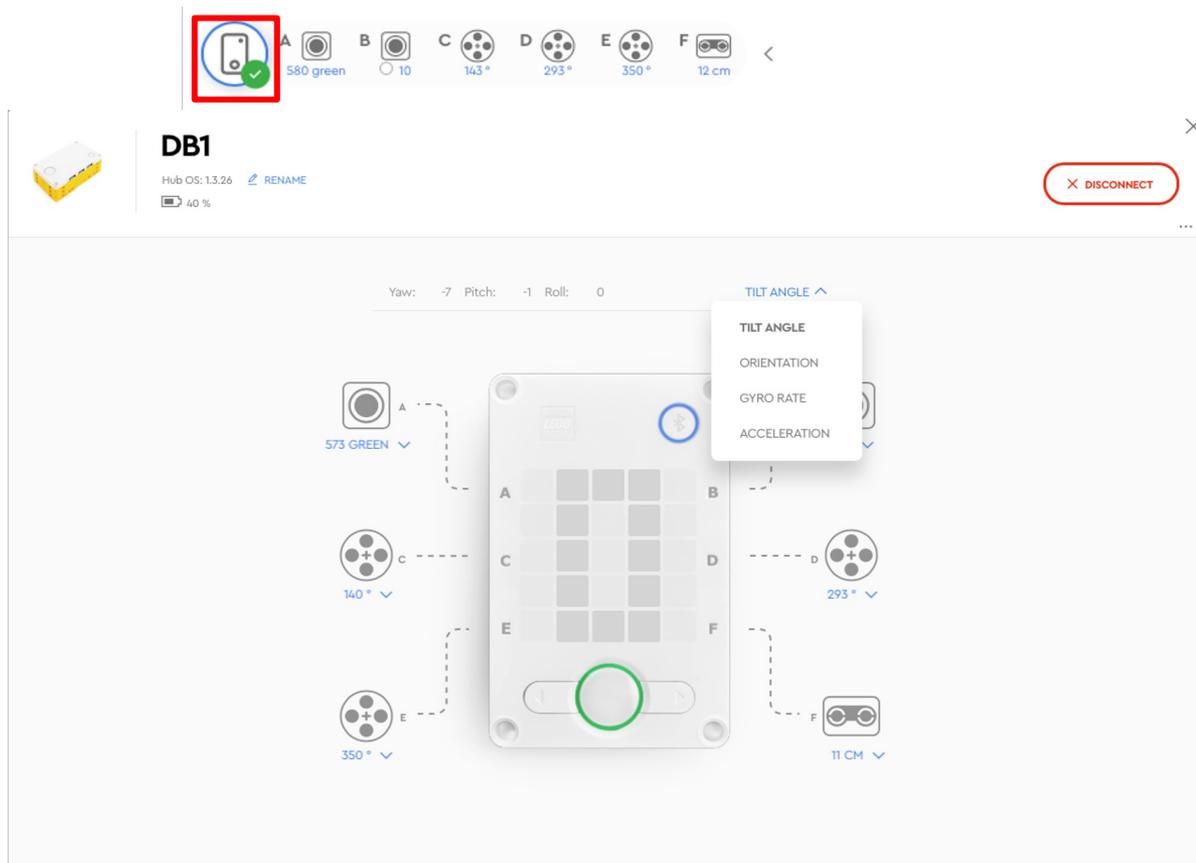
# SENSOR VALUES ON PROJECT PAGE

Once connected to the Hub, Sensor and Motor Values are visible on the top of each Project page in SPIKE Prime



# SENSOR VALUES ON DASHBOARD

Clicking on the Hub Icon in either SPIKE Prime takes you to the Hub Dashboard where you can see more sensor data



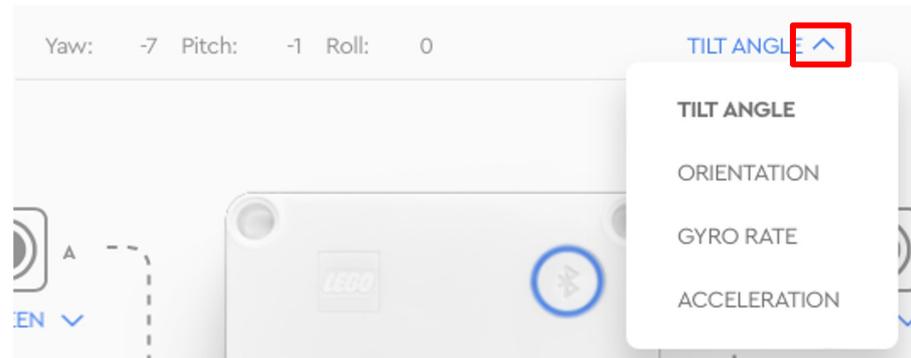
# MORE SENSOR DATA ON THE DASHBOARD

From the Hub Dashboard, you can also see a lot of additional information about each sensor/motor

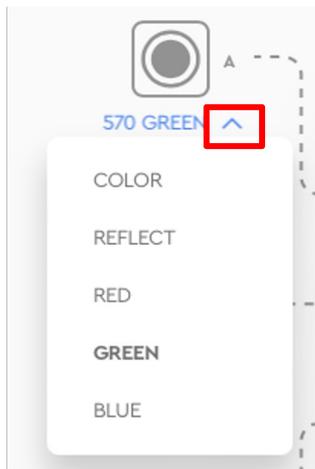
Select the mode using the down arrow

You can also see values for the built-in gyro sensor

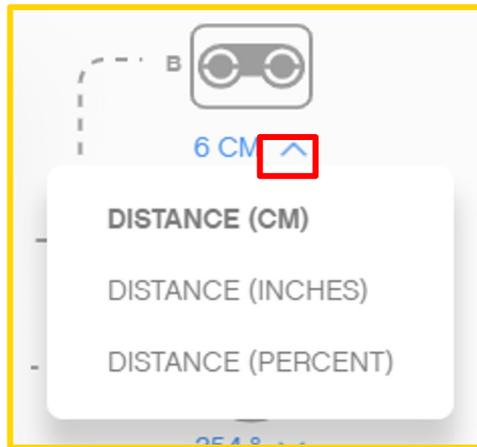
## Gyro Sensor



## Color Sensor



## Distance Sensor

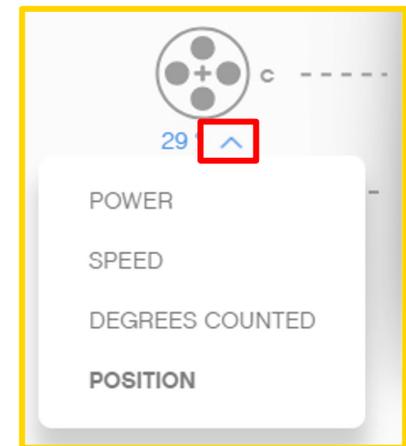


## Force Sensor



Note: The FORCE is 0 N i.e., 0 Newtons. Not "ON"

## Motor



# SENSOR DATA IS POWERFUL

Sensor data can help you program better and also help you debug your code

As you go through the rest of our lessons, you will use this feature often.

As you complete each challenge, think about how knowing sensor data might help you.

**NOTE:** Unlike Blocks, Python API for SP3 v3.4 returns values that can be **different** from what you see in the Dashboard. Some are outright confusing.

The next few slides describe the differences and how to adjust for them

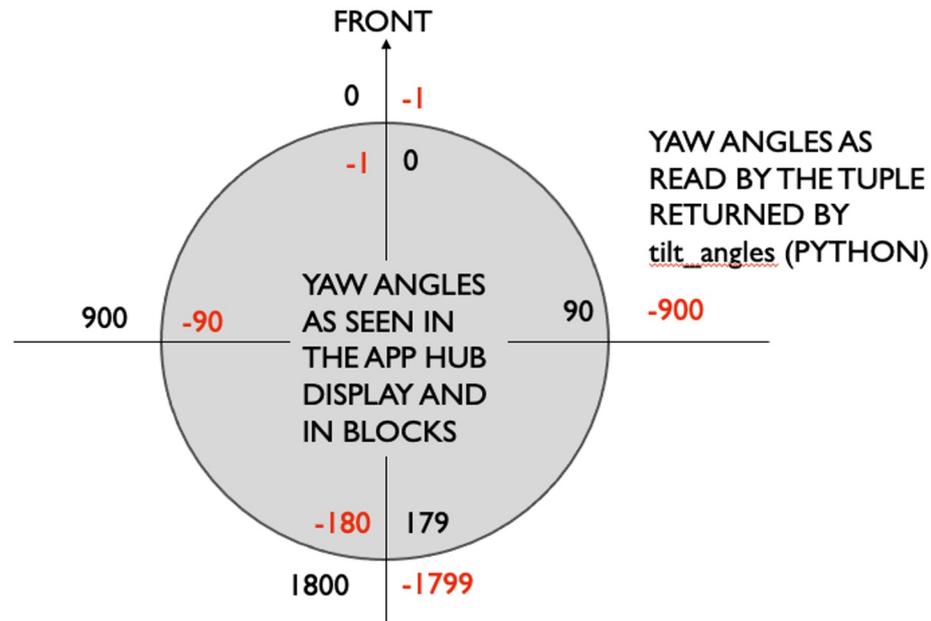
Example programs are found in the Sensor lessons.

# GYRO: DASHBOARD VS API

The gyro sensor in the dashboard shows yaw angle as 0 to 179 clockwise, and -1 to -180 counterclockwise.

The Python API `motion_sensor.tilt_angles(port)[0]` returns -1 to -1799 clockwise, and 0 to 1800 counterclockwise, with a sharp transition at the halfway mark.

Multiply the API reading by -0.1 to get the same value as in the dashboard.



# COLOR SENSOR: DASHBOARD VS API

The color sensor API `color_sensor.rgbi(port)` returns four values, red, green, blue and an overall intensity.

The Knowledge Base as of SP3 v 3.4 does not have documentation on these values

The r,g,b values can be seen on the Dashboard and range from 0-800 by observation

The intensity is not shown on the Dashboard. By testing in a program, the values appear to go from 0 - 1024.

Color and Reflection appear to work consistently with the Dashboard.

# DISTANCE SENSOR: DASHBOARD VS API

When the distance sensor does not sense any object, i.e nothing is within its sensor range, the dashboard shows the maximum value of 200cm

The distance sensor API `distance_sensor.distance(port)` will return a value of -1 under the same condition.

The Dashboard shows the distance in CM or IN. This is the same as Blocks, but the Python API returns the distance in **mm**.

To check if an object is closer than **10cm**, you have to check that the value is less than 100 **and** greater than 0. Remember the API returns values in **mm**.

Simply checking “less than 100” is not enough as -1 is also less than 100!

# FORCE SENSOR: DASHBOARD VS API

The Dashboard has an option to show the force in:

■ Newtons (FORCE option): Values range from 0 - 10

deciNewtons (FORCE RAW option): Values range from 0 - 100

The Python API `force_sensor.force(port)` returns the result in deciNewtons. This is equal to the Dashboard FORCE RAW

The Python API `force_sensor.raw(port)` returns a value from ~0-700. it is not clearly documented in the KB. It is not represented anywhere on the Dashboard.

# MOTOR: DASHBOARD VS API

The Dashboard shows motor speed as a percentage. This is the same as in Blocks

The Python APIs that **input** velocity defines absolute range values based on the type of motor:

Small motor (essential): -660 to 660

Medium motor: -1110 to 1110

Large motor: -1050 to 1050

The Python API `motor.velocity(port)` that **outputs** the current velocity appears to return a **percentage**, which is the same as the speed Dashboard value.

# WHEN CAN SENSOR DATA CAN HELP?

**Reduce Guess-and-Check:** I want my robot to turn a particular amount, but I am not sure how much to turn without guessing.

**Debug Code:** The robot does not follow the green line like I programmed it to do. Why not? What color does the robot think that green line is?

**Check Builds:** I built my robot with the Force sensor a little bit inside the robot. I am not sure that the Force sensor is getting pressed enough. How can I make sure the sensor is getting pressed?

**Test Sensors:** I told my robot to stop when the Distance sensor is 20cm away. But it seems to stop earlier. Is the sensor working correctly? How can I see what the Distance sensor sees?

# CREDITS

This lesson was created by Sanjay Seshan and Arvind Seshan for Prime Lessons

Additional contributions by FLL Share & Learn community members.

More lessons are available at [www.primelessons.org](http://www.primelessons.org)



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).