**Pi2Go Programming: Sensors**



**AIM:** After completing this worksheet you should be able to describe the sensors on the Pi2Go simulation and use Python to get information from them.

**You Need:** To complete this worksheet you need to have to have a virtual Pi2Go simulator (see WS1), and understand how to control it from the IDLE command line (see WS2).

**If the simulator isn’t already running:**

Start it (see WS1) and select the Pi2Go robot and default\_world.xml

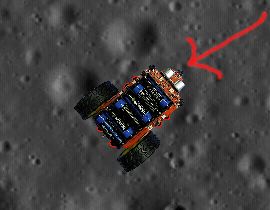
Now open a new IDLE window.

A Pi2Go robot has four different kinds of sensors:

1. **Ultrasonic:** It has an ultrasonic sensor which uses sound and echoes to measure distances.
2. **Infrared:** It has several infrared sensors that measure how much infrared light is reflected. These can be used both to measure distances (like with the ultrasonic sensor) and to measure colour – for instance a black surface reflects less infra-red light than a white surface.
3. **Light:** The Pi2Go also has several light sensors which detect the intensity of light shining on them.
4. **Switch:** Finally the Pi2Go has a switch which can detect whether it is pressed or not.

These sensors are all simulated on your virtual Pi2Go and you can use Python to get values from the sensors.

**The Ultrasonic Sensor**



The Ultrasonic Sensor

Locate the ultrasonic sensor on your virtual Pi2Go. There is one command you can use to get information from the ultrasonic sensor. This is:

>>> pi2go.getDistance()

pi2go.getDistance() returns the distance in cm to the nearest reflecting object. A value of zero means it does not detect an object

Type pi2go.getDistance()



**Question 1:** What value do you get?

Let us put something in front of the virtual Pi2Go.

Type E anywhere in the Simulator Window to open the Objects window and place one of the blocks in front of the Pi2Go’s distance sensor.

Try getting the distance using the Python command.

**Question 2:** What value do you get?

Now try moving the object closer to the sensor using the mouse to click and drag it.

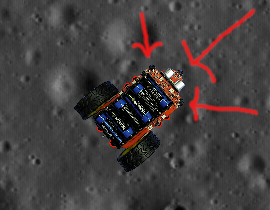


**Question 3:** What value do you get?

**Infrared Sensors**

The Pi2Go’s infrared sensors are organised into two groups: sensors for detecting obstacles and sensors for detecting lines.

There are three sensors for detecting obstacles grouped at the front of the robot:



The locations of the InfraRed Obstacle Sensors

There are four commands for getting information from the obstacle sensors:

>>> pi2go.irLeft()

returns 1 (meaning **True**) if the Left IR Obstacle sensor detects an obstacle

and 0 (meaning **False**) otherwise

>>> pi2go.irRight()

returns 1 (meaning **True**) if the Right IR Obstacle sensor detects an obstacle

and 0 (meaning **False**)otherwise

>>> pi2go.irCentre()

returns 1 (meaning **True**) if the Centre IR Obstacle sensor detects an obstacle

and 0 (meaning **False**) otherwise

>>> pi2go.irAll()

returns 1 (meaning **True**) if any of the Obstacle sensors detect an obstacle

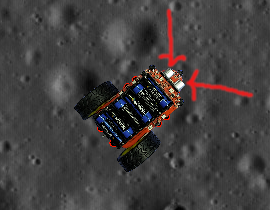
and 0 (meaning **False**) otherwise

Using the Objects Window place blocks so that pi2go.irAll() returns 1.

**Question 4:** How many blocks did you need?

The Pi2Go’s other infrared sensors are used to detect how dark the colour of the surface the robot stands on is. These are referred to as “line sensors” since their primary use is for detecting black lines on white surfaces.

There are two line sensors.



The Line Sensors

There are two command for getting information from the line sensors:

>>> pi2go.irLeftLine()

returns the state of Left IR Line sensor

>>> pi2go.irRightLine()

returns the state of Right IR Line sensor

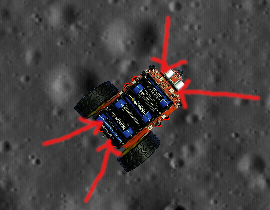
In the Objects Window select the Black Square and place it under the front of the robot.

**Question 5:** What do the python commands return if the Pi2Go is over the black square?

**Question 6:** What do they return if the Pi2Go is over the background?

**Light Sensors**

There are four light sensors: two at the front and two at the back.



Front and back light sensors.

There are five commands for getting data from the Pi2Go light sensors.

>>> pi2go.getLight(0)

returns the value 0..1023 for the the sensor numbered 0.

Try getting values for sensors 1, 2 and 3 as well.

>>> pi2go.getLightFL()

returns the value 0..1023 for Front-Left light sensor

>>> pi2go.getLightFR()

returns the value 0..1023 for Front-Right light sensor

>>> pi2go.getLightBL()

returns the value 0..1023 for Back-Left light sensor

>>> pi2go.getLightBR()

returns the value 0..1023 for Back-Right light sensor

Put a light source into the world.

Check the values returned by one of the sensors when

1. the light is in front of the sensor shining towards it,
2. the light is in front of the sensor shining away from it,
3. the light is on the other side of the robot from the sensor shining towards it

**Question 7:** What value is returned?

Case A Case B Case C

**Switch**

The switch is on the side of the real Pi2Go.

In the Pi2Go simulator, you will find it displayed as a button at the bottom of the virtual world window.



The Pi2Go Switch

There is one command for getting data from the Pi2Go Switch:

>>> pi2go.getSwitch();

returns True when the switch is pressed and False when it isn’t.

**Try using the switch now.**

**Remember:** When you have finished working with your robot type:

**pi2go.cleanup()**



University of Liverpool, 2019

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