

Senn Robotics Team

Module 05: Sensors

Infrared Sensors

This program is intended to teach you the basics of the IR (infrared) sensor and how to read the data that it produces. An infrared sensor essentially detects the amount of infrared energy reflected from an object, and it can be used to detect areas that are light or dark, such as black tape. If an area is light, more infrared energy is reflected, and the value is higher. If an area is dark, the less infrared energy is reflected, and the sensor produces a lower value.

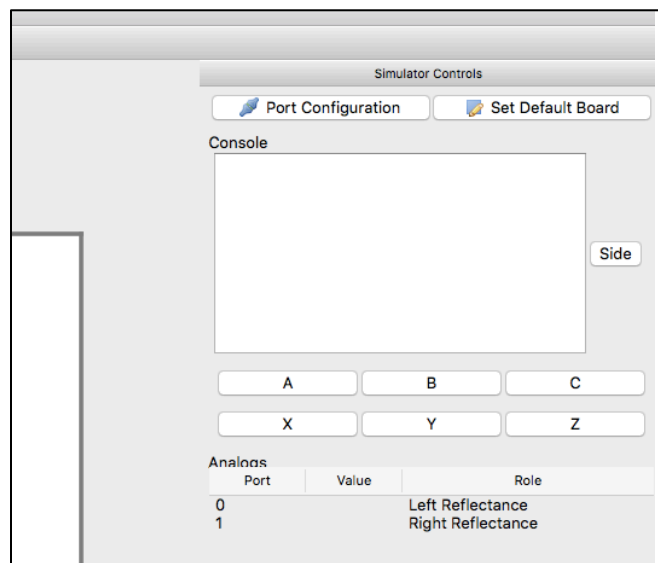
To detect the values coming from the infrared sensor, the `analog10()` function with the port number the sensor is attached to as the only argument/parameter. For example, if the IR sensor were connected to port 1, you would call the function using the function

```
analog10(1);
```

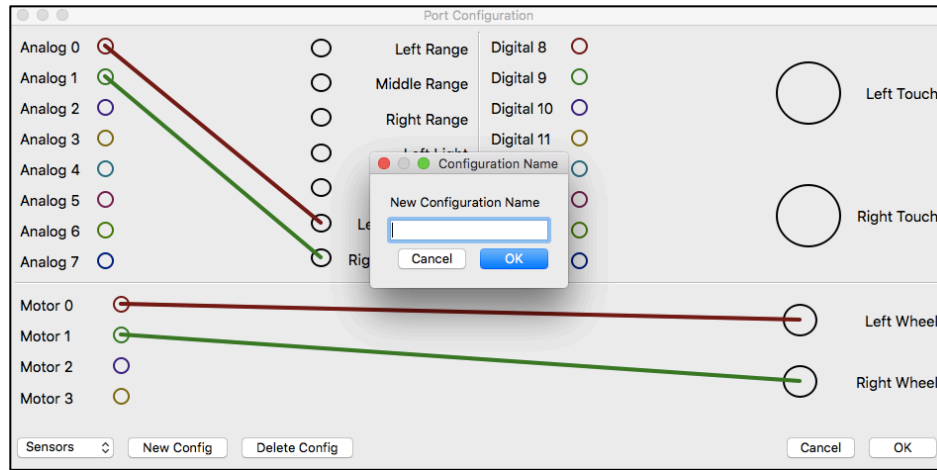
This function will return a value anywhere between 0 and 1023 ($1024 = 2^{10}$). As the value approaches 0, less infrared energy is detected. As the value approaches 1023, more infrared energy is detected. You will need to work with the engineers to determine what a good 'threshold' value is, but we will start in the middle at 512.

Setting Up the KISS Simulator

You need to create a new port configuration using the KISS simulator for the left and right reflectance sensors to run. Once you've launched the simulator, click **Port Configuration** in the upper right hand corner.



Then, click **New Config** in the lower left hand corner. Name the configuration **Sensor** and click **OK**. Connect the motors and analog sensors to the appropriate ports.



You can select this configuration in the future.

If-Statement

We can use a new logical structure called an "if-statement" to check whether what the sensor is testing. The basic structure is

```
if (condition to check) {
    statements to do if the condition is true
}
```

Here some example of conditions:

```
i > 4 (greater than)
i < 10 (less than)
i >= 512 (greater than or equal to)
i <= 512 (less than or equal to)
i == 1 (exactly equal to. BE CAREFUL TO USE DOUBLE == and not =)
i != 1 (not equal to)
```

Exercises

Using the example program, follow these steps. Consider answering questions 6 and 7 in your coding notebook.

1. Define variables for the power of the wheels so we don't have to keep typing in 75 each time.
2. Define a variable for the port number, so you don't have to type in the port number.
3. Ack! This code is terrible - there are no printf statements! Add them so you know what's happening as the robot executes the program.
4. Ack! This code is still terrible. There are very few comments. Add comments along the way so your coding team knows what's happening.
5. Modify the code so that the robot waits 10 seconds if it sees a dark color and 5 secs if it sees white.
6. What is the difference between the 'if' and the 'else if'?
7. Under what circumstances would the very last if/else case run? Why would you need this case?

Challenge Exercise

Write code that does the following:

If you position the robot in the upper left pink corner of the game board, it moves step by step to the black line. On the black line it turns and follows the black line to the end (of the black line). Use just a series of if-else statements.