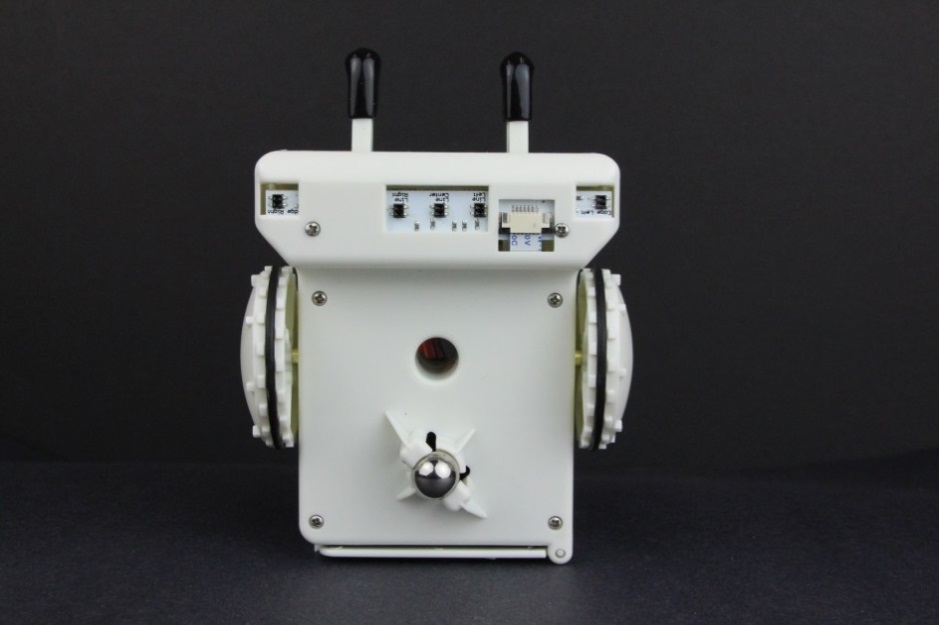
Five infrared (IR) sensors are located underneath Sparki. These can be used to detect (and, ultimately, to follow) lines and edges. Can you use these sensors to stay within a target area? Can you use them to follow a line?



**Background:** Each of Sparki’s IR sensors has

* A light-emitting diode (LED) that shines infrared light on the surface below Sparki; and
* A phototransistor that measures how much of that infrared light is reflected back from the surface.

The sensor outputs a value that is based on how much light is detected.

* If you place Sparki on a highly reflective surface, the sensors will return very high values (1000+); in other words, a **higher value reading** means that the sensors have sensed **more reflected light**.
* If you place Sparki on a dull, black surface the sensors will return very low values (perhaps less than 200); in other words, a **lower value reading** means that the sensors have sensed **less reflected light**.

*Note:* Because you can also use the infrared sensors to determine whether Sparki is about to go over the edge of a surface (since without a surface underneath Sparki to bounce the infrared light back, the sensor will not see much infrared light), the outside infrared sensors are labeled “edge sensors”.

**Tutorial:** Using a black marker, draw a large square with a thick border (at least 1cm) on a piece of white paper or foam core. Sparki can detect this line by sensing the light reflected from the surface and determining whether it is “high” (corresponding to the white paper) or “low” (corresponding to the black line). To define “high” and “low” we must first identify the threshold value for the black line (*i.e.*, the sensor reading that indicates that Sparki detects the line.) The flowchart for this code is:

Output to Serial Monitor

Read IR Sensors

The code for this flowchart is:

#include <Sparki.h> // loads the Sparki library of code

void **setup**(){

Serial.begin(9600); // required so that Sparki can output the measured

// reflected light values to the Serial Output Monitor

}

void **loop**() {

int edgeLeft = sparki.edgeLeft(); // measure the left edge IR sensor

int lineLeft = sparki.lineLeft(); // measure the left IR sensor

int lineCenter = sparki.lineCenter(); // measure the center IR sensor

int lineRight = sparki.lineRight(); // measure the right IR sensor

int edgeRight = sparki.edgeRight(); // measure the right edge IR sensor

Serial.println(edgeLeft);

Serial.println(lineLeft);

Serial.println(lineCenter);

Serial.println(lineRight);

Serial.println(edgeRight);

delay (3000); //make sure to use a sufficient delay so you have time to

// read your sensor outputs

}

Copy the code in the previous box, paste it into Sparki’s IDE, and upload it to Sparki. Answer the following questions:

* What range of values is returned when Sparki’s sensors are over the black line? \_\_\_\_\_\_\_\_\_\_
* What range of values is returned when Sparki’s sensors are over the white paper? \_\_\_\_\_\_\_\_\_
* What threshold value will you use in your program to determine whether Sparki detects the line? \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now you are ready to solve the two mini-challenges of this handout.

**Step 1:** In your engineering notebook, create a flowchart for the program that will accomplish the mini-challenge. You may want to build on the flowchart provided in the tutorial.

**Mini Challenge #1:** Create a program that enables Sparki to locate all four sides of the square that you drew for the tutorial. Have Sparki give a signal each time it finds a side.

**Step 2:** Write the program that corresponds to your flowchart. Note that:

* You will need to understand how to make Sparki move forward and turn.
* You will need to use conditional programming (if/then).
* You will need elements from the sample code in the tutorial above, including #include <Sparki.h> and the variables.

**Step 3:** Upload your program to Sparki and demonstrate that your code works.

**Mini Challenge #2:** Create a program that enables Sparki to follow a line of your choice. Assume that Sparki starts on the line.

**Step 1:** Using a black marker, draw a line thick line (at least 1cm think) on a piece of white paper or foam core.

**Step 2:** In your engineering notebook, create a flowchart for the program that will accomplish the mini-challenge. You may want to build on the flowchart provided in the tutorial.

**Step 3:** Write the program that corresponds to your flowchart. Note that:

* You will need to understand how to make Sparki move forward and turn.
* You will need to use conditional programming (if/then)
* You will need elements from the sample code in the tutorial above, including #include <Sparki.h> and the variables.

**Step 4:** Upload your program to Sparki and demonstrate that your code works.