Introduction to Robotics with the Raspberry Pi

WORKING WITH INFRARED SENSORS

Jeff Cicolani
Sunday, July 9th, 2017

This workshop is assuming a you are using Windows. If you are using a Mac or Linux machine, good luck… I mean, you will need to look up the proper instructions for your OS. Sorry, I'm a Windows guy, now with a little Linux, but Windows is my go to OS for general use.

Introduction

By this point in the series, you should have a working robot. In previous sessions we've covered everything you need to know to install and program your robot. We've covered motors, sensors, and communication between the Raspberry Pi and the Arduino. Part of these workshops included accessing and processing information from an ultrasonic rangefinder. From here on out, we will start introducing new sensors, processing algorithms, and delving deeper into Python and C for the Arduino.

In this workshop, we will be working with infrared (IR) sensors. We will be looking at different types of sensors and a couple of different packages you find for them. We will be using a series of IR sensors to detect the edge of a surface and to detect a line.

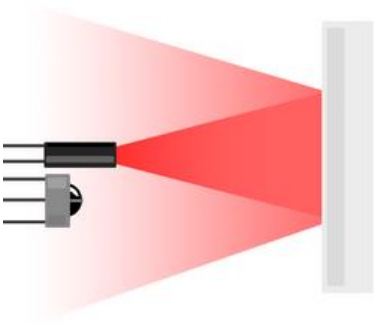
By the end of this workshop, you will be able to add edge sensing and line following applications to your robot.

Infrared Sensors

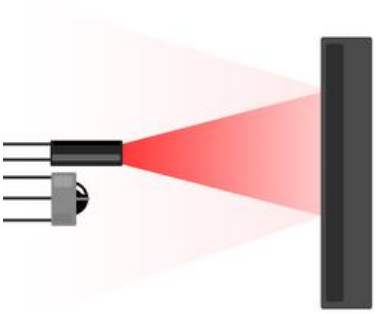
An infrared (IR) sensor is any sensor that uses a light detector, tuned for the IR spectrum, to detect an IR signal. Generally, the IR sensor is paired with IR emitting LED to provide the IR signal. The emissions from the LED are either measured for intensity or, simply, presence.

Types of IR Sensors

When measuring for intensity, the sensor is measuring how much IR radiation is hitting it. The IR source could be anything, but is generally from a paired IR LED. The LED is emitting at a specific brightness. The IR receiver will then return a signal relative to that intensity.



Lightly colored objects reflect more IR light



Darker colored object reflect less IR light

Some IR sensors work by detecting the difference between ambient IR radiation and the IR radiation emitted by objects within the sensors range. These sensors are frequently used for detecting heat or motion.

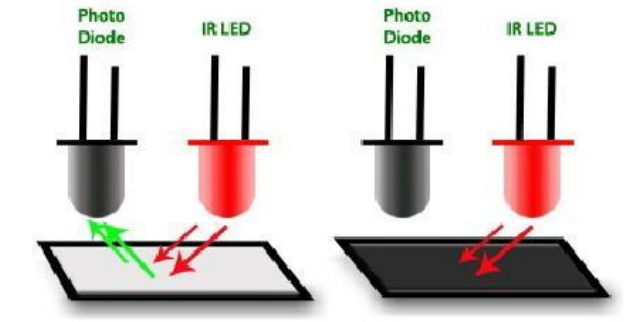
When the sensor is configured to detect the presence of an IR signal, it is doing just that. The sensor is detecting a threshold of IR intensity. The sensor returns a low signal until the threshold is exceeded, at which point it returns a high signal. These sensors are generally paired with an emitting LED, either in a reflected or direct configuration.

Reflectance Sensors

Reflectance sensors are any sensor designed to detect a signal reflected off a target. Ultrasonic rangefinders are reflectance sensors because they detect the wavelength of sound that is bounced off objects in front of them. IR reflectance sensors work in a similar fashion in that they read the intensity of IR radiation reflected off an object.

Line and Edge Detection

When a threshold is introduced to a reflectance sensor, it is generally used to detect whether the reflected signal is above or below a specific intensity. These sensors are good for line detection when the contrast between the surface and the line are high, for instance, a black line on a white table. When the sensor is over the white surface, most of the IR signal is returned to the sensor and the sensor will return a high value. When the sensor is over the dark line, less of the IR signal is returned, resulting in a low value.



High Value of reflectance/voltage Low Value of reflectance/voltage

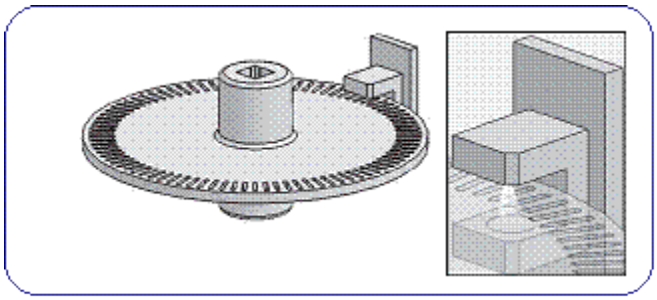
In much the same, way, the sensor can detect the edge of a surface. When the sensor is over the surface, the sensor receives more IR signal. When the sensor is over an edge, the signal is greatly reduced, resulting in a low value.

Rangefinders

Without the threshold, the IR sensor will return a range of signal. When coupled with an analog to digital converter, this range can be used to determine the approximate range to an object. Unlike ultrasonic rangefinders, IR rangefinders are designed to detect specific ranges, so it is important to match the sensor to the application.

Interrupt Sensors

Interrupt sensors are used to detect the presence of an IR signal. These are usually paired with an emitting diode and configured in such a way that allows an object to pass between the emitter and detector. When the object is present and blocking the emitter, the receiver returns a low signal. When the object is not present, and the receiver is allowed to detect the emitter, the signal is high.



These sensors are frequently used in devices known as encoders. An encoder will generally consist of a disc or tape with translucent and transparent sections. As the disc or tape moves past the sensor, the signal continuously goes from high to low. A microcontroller, or other electronics, can then use this alternating signal to count the pulses. Because the number of transparent sections are known, the movement can be calculated with high confidence. Some encoders use a number of sensors to provide more precise information about the movement, including direction.

PIR Motion Detectors

Another, very common, sensor is known as a PIR motion detector. PIR stands for Passive Infrared. These sensors will have a faceted lens that reflects and refracts the IR radiation emitted or reflected by an object onto IR sensors within. When a change is detected on these sensors, a high signal is produced.

These sensors are what controls the automatic doors at your local grocery store or operate the automatic lights in your home or office.

Working with IR Sensors

Four Sensor Module

The four-sensor module is just that. It consists of four IR sensors connected to a control board. The control board had six pins that will connect to your microcontroller; VCC, ground, and one for each sensor. Each sensor also has a potentiometer which controls the sensitivity. The board's electronics use the potentiometer to set the threshold for each sensor, so the output is either high or low. These sensors are specifically designed to be used for line following applications.

Connecting the Four Sensor Module

To set up the sensor you will need to connect each of the IR modules to the sensor board. Make sure the pins on the module is connected to the proper pins on the board. Next, you'll need to connect the six pins from the board to your Arduino. To do this, use the female to female jumpers.

1. Connect the VCC to 5v
2. Connect GND to GND
3. Connect the remaining pins from the board to pins 3, 4, 5, and 6.

Code

The code for using this module is pretty straight forward. All we are looking for is the digital pin state. When it returns 1, the reflectance exceeds the threshold. When it returns 0, the threshold is not met.

```
int ir1 = 0;
int ir2 = 0;
int ir3 = 0;
int ir4 = 0;

const int ir1pin = 3;
const int ir2pin = 4;
const int ir3pin = 5;
const int ir4pin = 6;

void setup() {
        pinMode(ir1pin, INPUT);
        pinMode(ir2pin, INPUT);
        pinMode(ir3pin, INPUT);
        pinMode(ir4pin, INPUT);

        Serial.begin(9600);
}

void loop() {
        ir1 = digitalRead(ir1pin);
        ir2 = digitalRead(ir2pin);
        ir3 = digitalRead(ir3pin);
        ir4 = digitalRead(ir4pin);

        Serial.println(String(ir1) + ',' +
```

```
                String(ir2) + ',' +
                String(ir3) + ',' +
                String(ir4))
}
```

The most challenging part of this project is getting the threshold set correctly. It is generally easiest to do this once the board and the sensors are mounted, just make sure you have access to the potentiometers to adjust the sensitivity. Use a piece of electrical tape on a blank, white piece of paper.

1. Move the paper such that the tape is beneath one of your sensors.
2. Adjust the sensor until it indicates it detects the paper. If you are using a module that has an indicator LED for each IR sensor, the LED will be on when the return is high and off when low.
3. Slowly adjust the potentiometer the other direction until the sensor indicates a low state.
4. Move the paper so the line is no longer beneath the sensor.
    a. If the sensor goes high, then your threshold for that sensor is set.
    b. If the sensor remains low, you will need to repeat these steps for that sensor until it the sensor is low for the line and high when the line is not present.
5. Repeat these steps for each sensor.

Explore

Now that you have the sensor calibrated and returning a 0 or 1 to your Arduino, build some logic that does something with this information. Ideally you'll be sending this information back to the Raspberry Pi and performing your logic there.

Hint: you are already returning the data to the Raspberry Pi if you are using Serial.print().

Bigger hint: This is the code I use to read the sensor data from the Arduino in Python:

```
import serial

ser = serial.Serial('/dev/ttyACM0',9600)

try:
        while True:
                serial_line = ser.readline()
                myList = serial_line.split(",")

                for i in range(len(myList)):
                        if myList[i][-4:] == '/n/r':
                                myList[i] = myList[1].rstrip()
                        print('sensor'+str(i+1)+'='+myList[i])

except KeyboardInterrupt:
        ser.close()
```

There are a couple tricks in that code that we will be exploring in upcoming workshops. In particular, we are using a "try" block. This is used for error handling. Here, we are simply using it for executing a line of code

that closes the serial port when we use ctrl+c to end the program. We are also using a Python list to hold the parsed information from the serial line. Again, we'll be exploring this in a later workshop.

In the next workshop, we will be learning about proportional, integral, and derivative (PID) control, which is one of the most common control algorithms. This is the last piece of the puzzle to make a simple line following robot.

After this next workshop we'll be spending a lot of time with Python. Earlier I blasted you with a lot of information, not really covering anything in detail. We will be revisiting those topics, spending a lot more time with each, and introducing some other critical pieces.