Intro to Robotics with Raspberry Pi!

Section 2. Physical Computing – Sensors and Data

Outline

Sensing the Environment with Raspberry Pi

Types Of Sensors

Different Types of Sensors

Denbot Kit Sensors: Ultrasonic and Optical

Controlling DC Motors

Powering Motors

Controlling Motor Speed with PWM

Measuring Motor Speed

Using Wheel Encoders

Measuring Rotations

Calculating Speed (Rotation and Translation)

Closed-Loop Speed Control

Maintaining a Constant Speed

Intro to Navigation

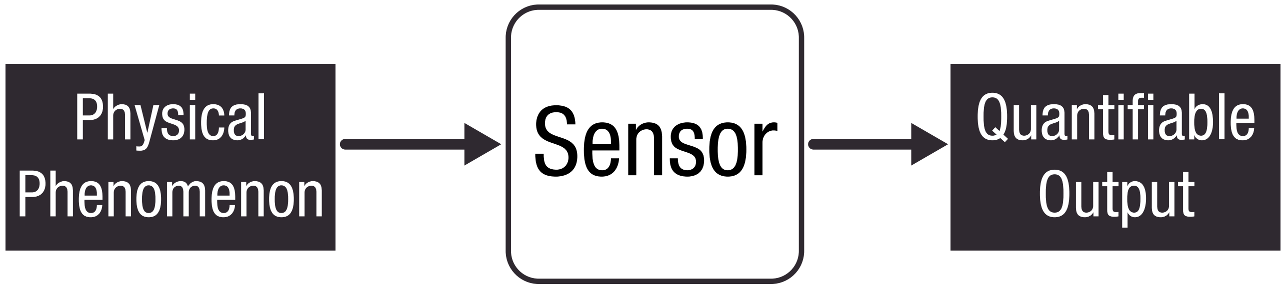
Controlling Direction

Differential Steering and Turning

Types of Sensors

What is a Sensor?

- An object whose purpose is to detect changes in its environment and to provide a corresponding output.



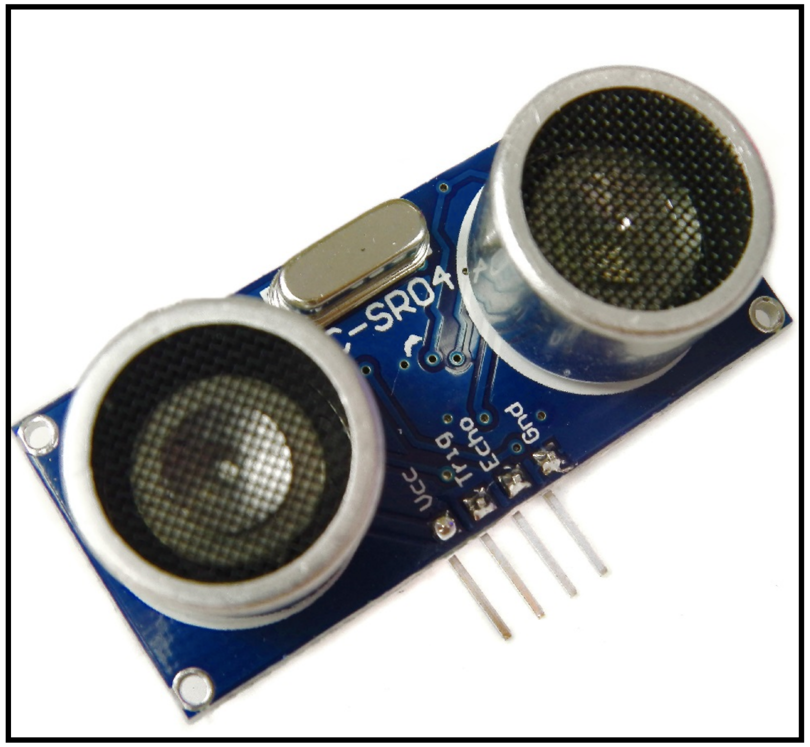
Types of Sensors

	Flow	Ultrasonic	
	Distance	Resistive	
Active	Position	Capacitive	Digital
Passive	Pressure	Inductive	Analog
	Temperature	Photosens.	
	Light	Electrochem.	

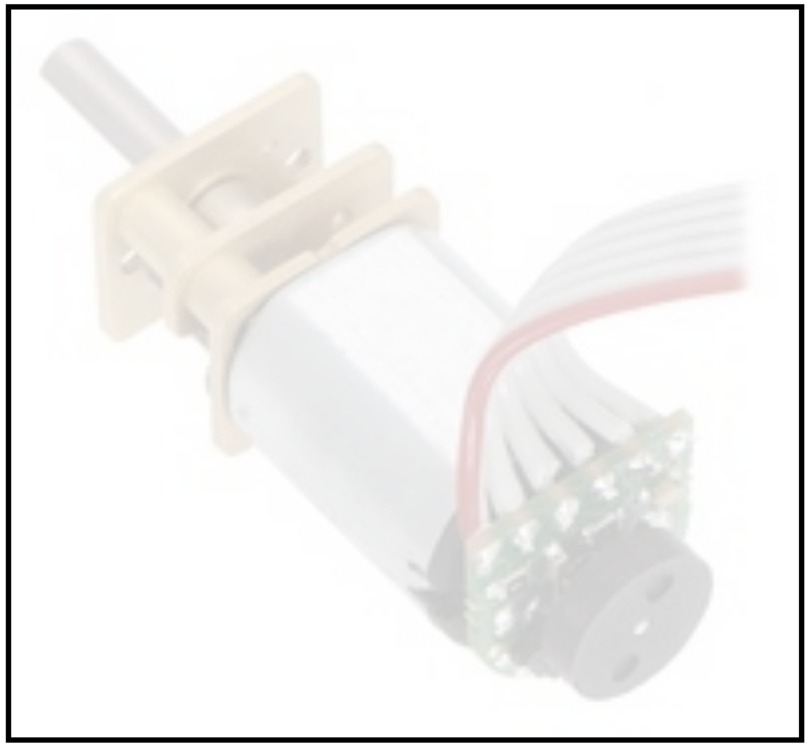
- **By power/energy supply requirement**
- **By parameter measured**
- **By principle of operation**
- **By output signal type**

Denbot Kit Sensors

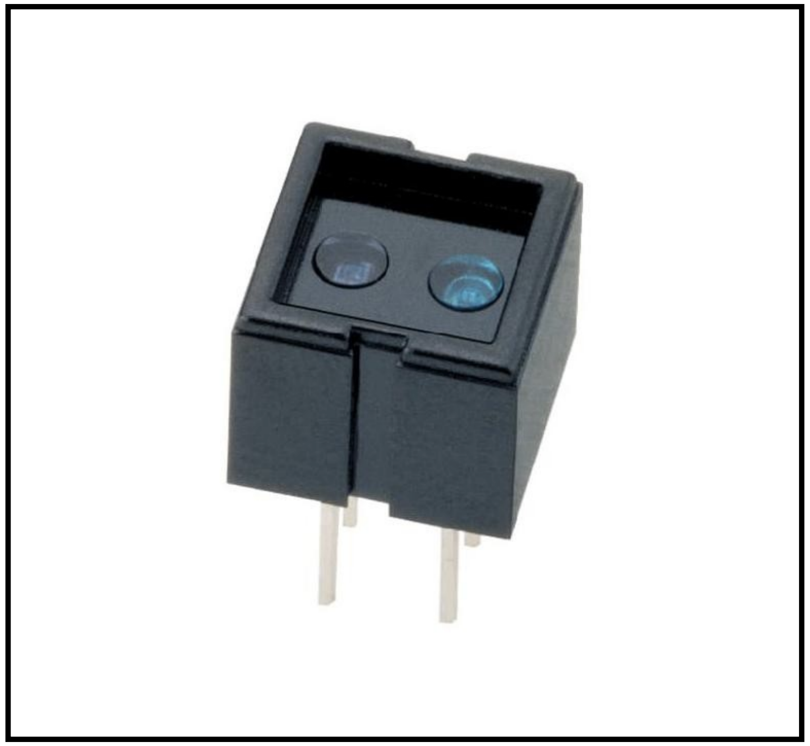
- **Distance (ultrasonic)**

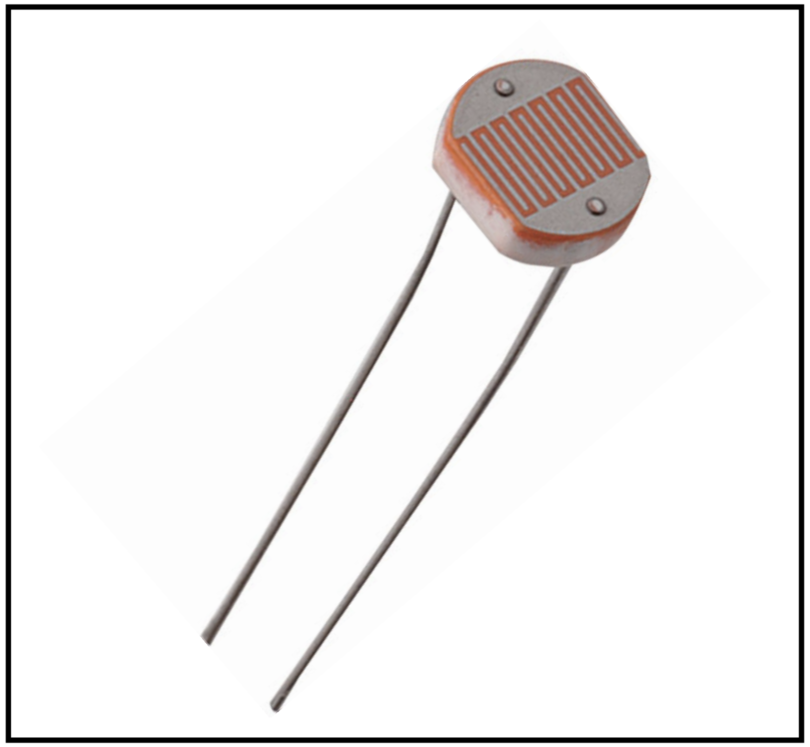


- **Speed (magnetic)**



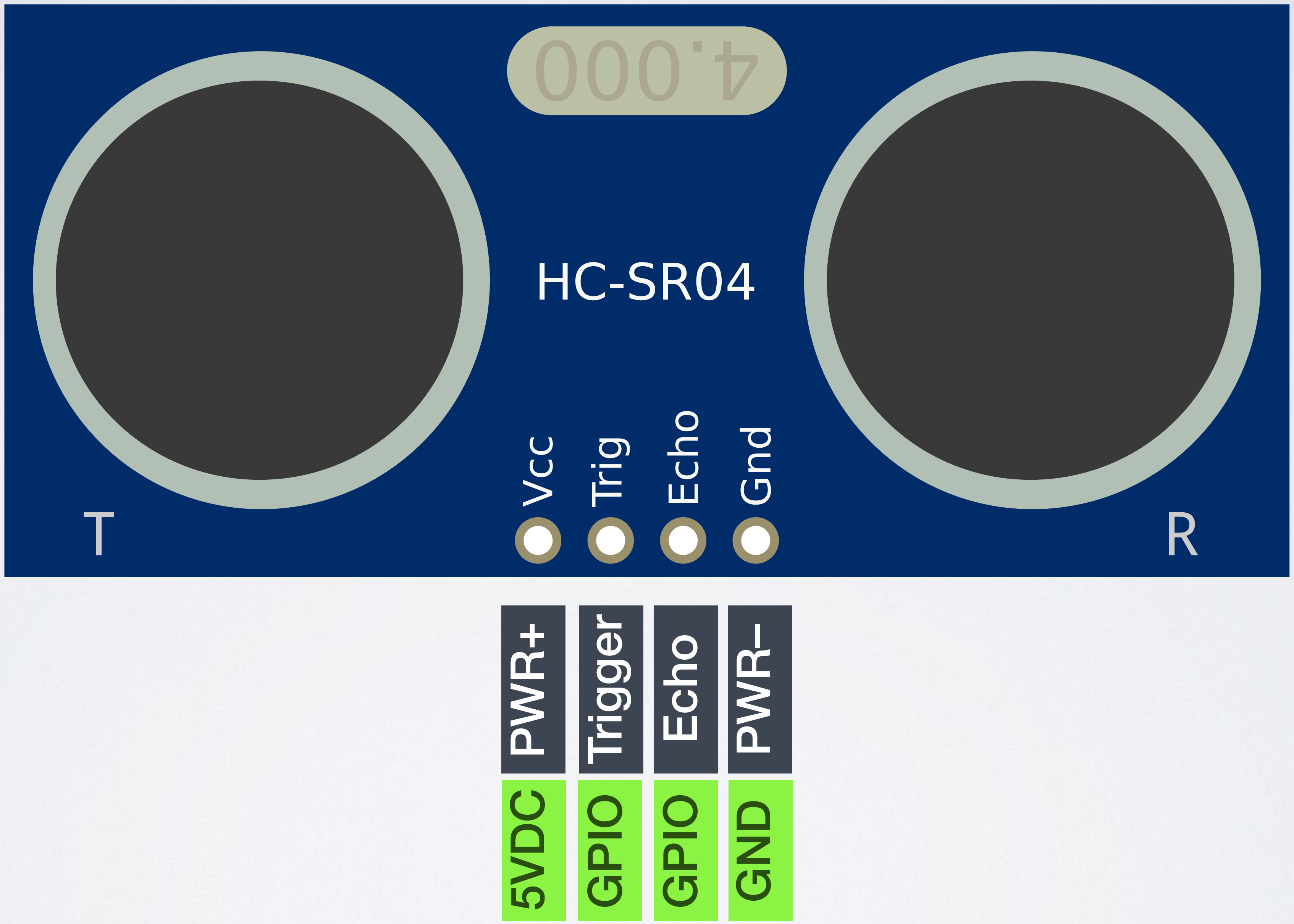
- **Light (photosensitive, resistive)**





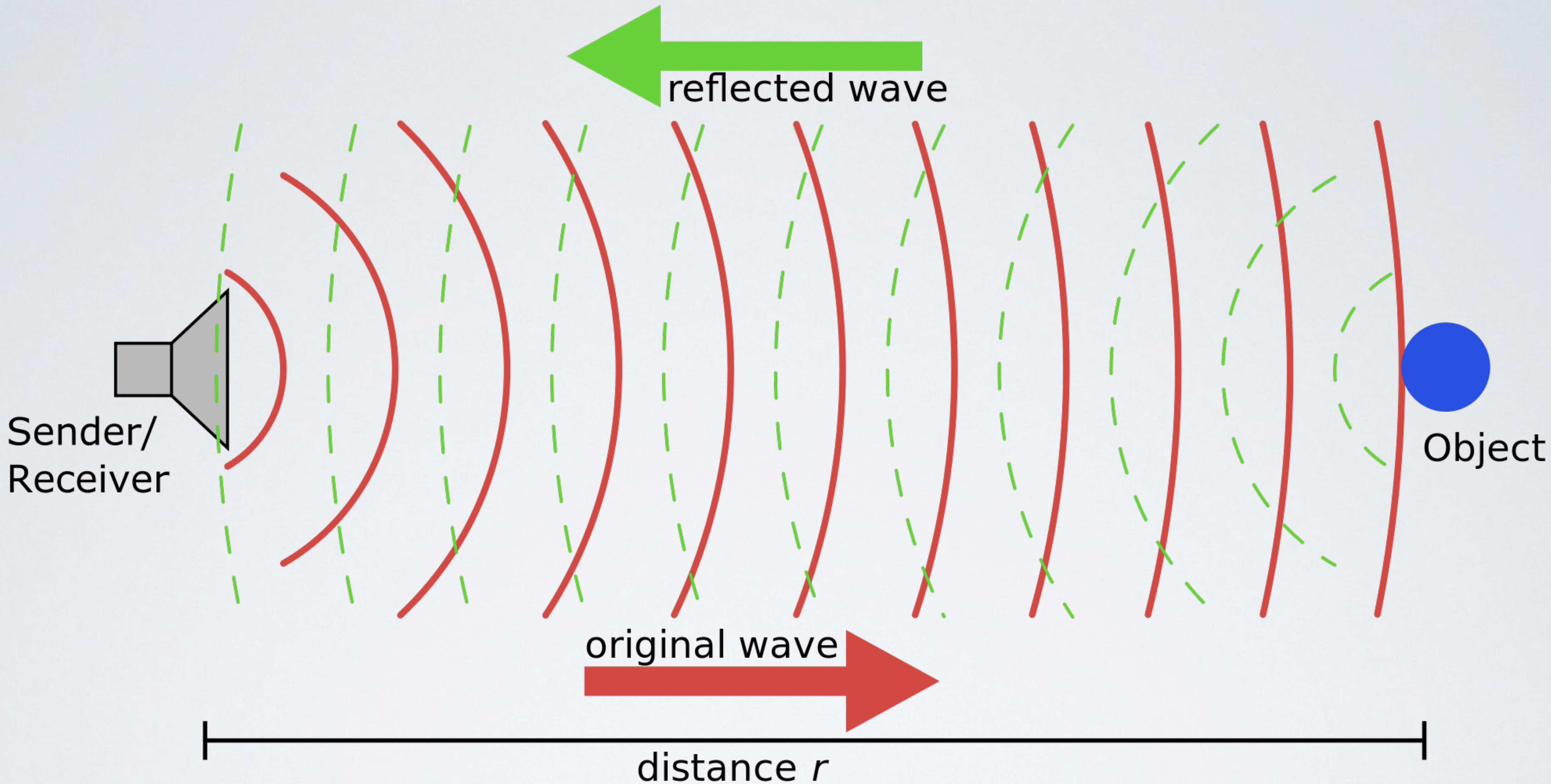
Ultrasonic Sensor Operation

Ultrasonic Sensor Pinout

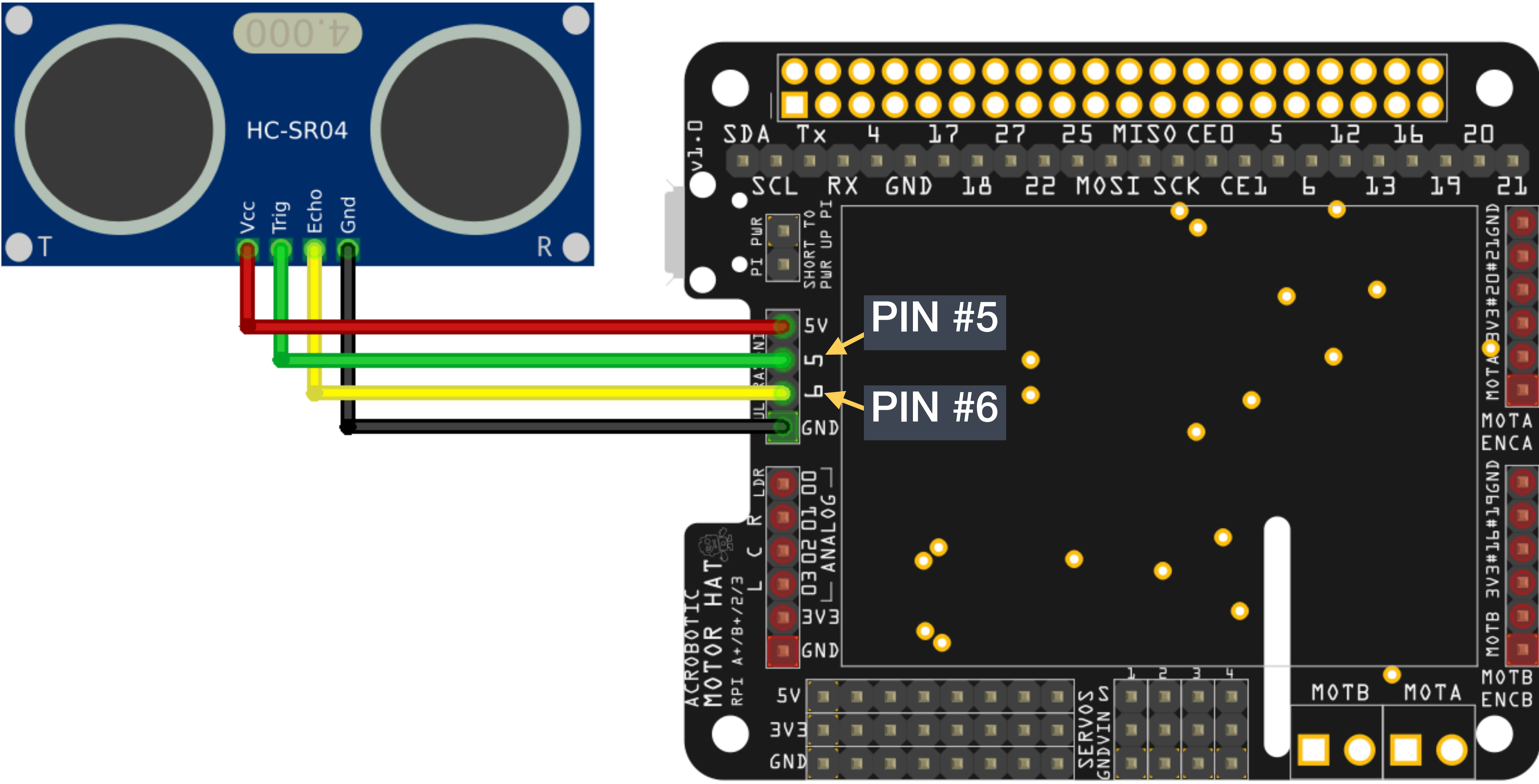


- To communicate with the Ultrasonic Sensor we need two power and two data signals.

Ultrasonic Sensor Principle Of Operation



Measuring Distance with Ultrasound



- Pin order on the HC-SR04 can change. Make a note of which pin (#5 or #6) is connected to **Echo** and **Trigger**.

Measuring Distance with Ultrasound

- As always, let's use **<u>Python</u>** on our Raspberry Pi's Operating System to get data from the ultrasonic sensor.
- Connect to your Raspberry Pi via SSH or VNC.
- From a Terminal window, navigate to the **sensors** directory.

```
cd ~/Makerden/sensors
```

- Fire up an interactive Python session.

```
python
```

Measuring Distance with Ultrasound

- Let's use the **<u>HCSR04 module</u>** to generate a trigger signal and measure its **time of flight**!

```
>>> from HCSR04 import Ultrasonic
```

```
>>> sensor_obj = Ultrasonic(pin_echo=6, pin_trigger=5)
```

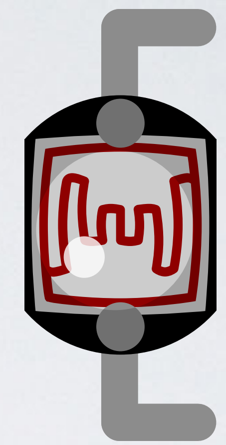
```
>>> t = sensor_obj.getPingTime()
```

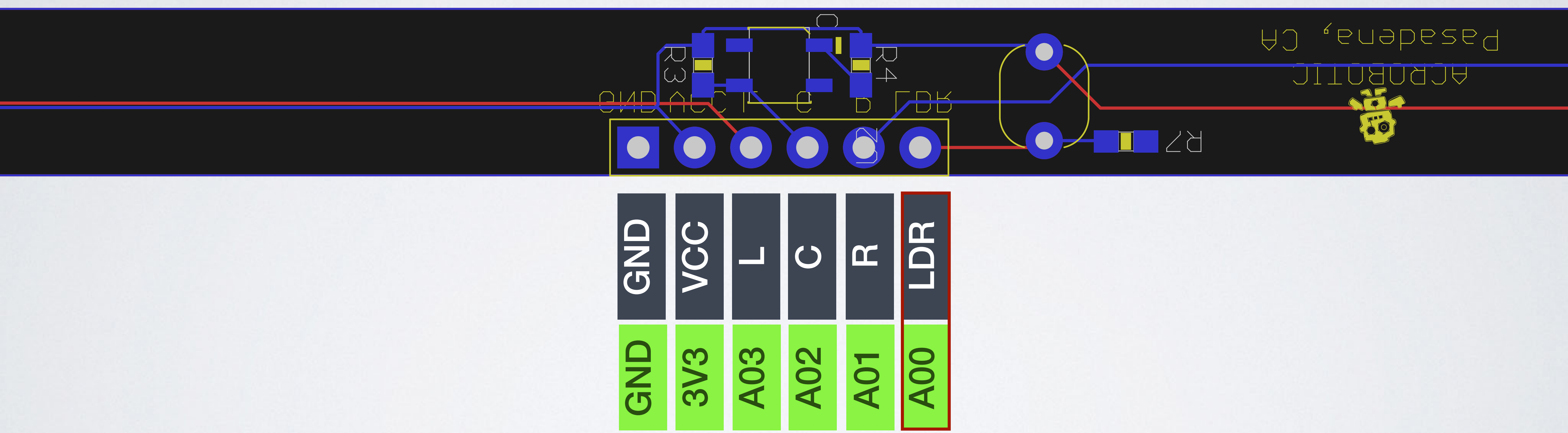
```
>>> print("Time of flight is %f milliseconds") % (t*1000)
```

- Write a script called **time_of_flight.py** where the 'ping time' is printed at **1Hz**. *Place objects in front of sensor to try and see it change.*

Photoresistive Sensor Operation

Light-Dependent Resistor (LDR)





- The 10K Ohm LDR located underneath the Line-Sensor board, and connected to a 10K Ohm (fixed) resistor as a voltage divider.

Measuring Brightness with Photoresistivity

- Let's use the **ADS1115 module** to read (digital) values between –32,768 and 32,767 corresponding to analog voltages output by the LDR!

```
>>> from ADS1115 import ADS115
```

```
>>> adc_obj = ADS1115()
```

```
>>> channel_0 = adc_obj.read_adc(0) #'LDR' sensor
```

```
>>> print("Channel 0 is reading %d") % (channel_0) #{-32,768~32,767}
```

- Write a script called **ldr_sensor.py** where the 'channel value' is printed at **1Hz**. *Cast shadows over sensor to try and see it change.*

Photoreflective Sensor Operation

Measuring Brightness with Photoreflectivity

- Let's use the **ADS1115 module** to read (digital) values between –32,768 and 32,767 corresponding to analog voltages output by the QRE1113!

```
>>> from ADS1115 import ADS115
```

```
>>> adc_obj = ADS1115()
```

```
>>> channel_3 = adc_obj.read_adc(3) #'L' (left) sensor
```

```
>>> print("Channel 3 is reading %d") % (channel_3) #{-32,768~32,767}
```

- Write a script called **light_sensor.py** where the 'channel value' is printed at **1Hz**. *Cast shadows over sensor to try and see it change.*