

Table of contents

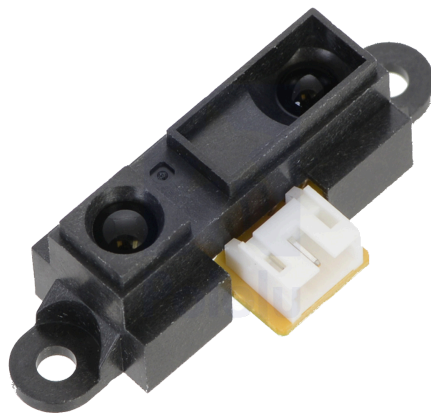
Distance Sensor Documentation for Botball	2
---	---

Distance Sensor Documentation for Botball

Technical Details

Overview

The distance sensor used in the Botball kit (2024) is the GP2Y0A21 Sharp distance sensor (<https://www.pololu.com/product/136>), a popular sensor for robotics projects. This sensor is ideal for detecting objects in front of the robot and can be used to avoid obstacles.



GP2Y0A21 Sharp distance sensor

Specifications

- **Range:** 10 cm to 80 cm
- **Output:** Analog voltage proportional to the distance of the object
- **Operating Principle:** Emits infrared light and measures the amount of light reflected back

Sensor Characteristics

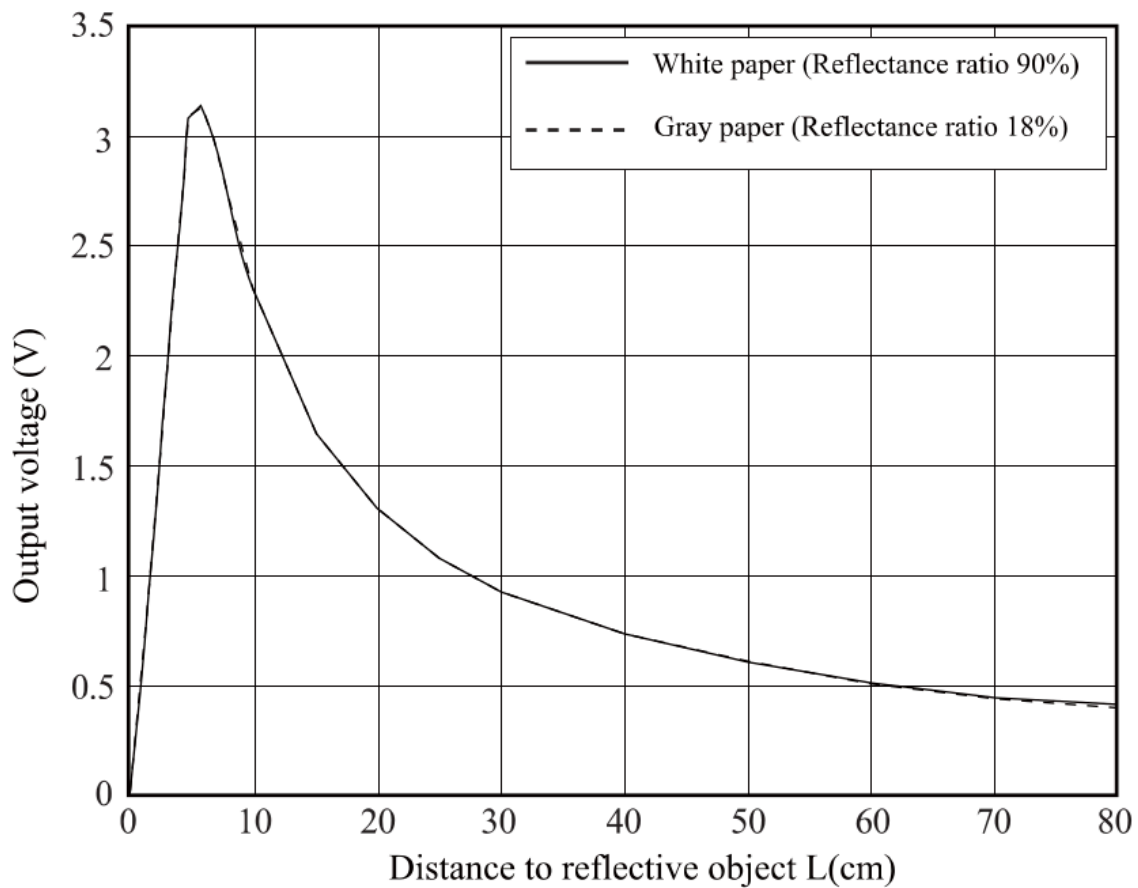
- **Range Sensitivity:** The sensor's output is an analog voltage that varies with the

distance of the detected object.

- **Color Sensitivity:** The sensor is not sensitive to the color of the object.
- **Placement Sensitivity:** The sensor's performance is affected by the angle at which the emitted light hits the object.

Voltage-Distance Relationship

The output voltage of the sensor decreases as the distance increases. However, this relationship is non-linear, and the sensor becomes less accurate as the distance increases.



Output Voltage vs Distance

- **Close Range (10 cm):** Approximately 2.25V
- **Far Range (80 cm):** Approximately 0.4V

- **Non-Detectable Range:** Objects closer than 10 cm result in a high output (~2900 analog value on the Wombat)

Conversion Formulas

To convert the analog values read by the Wombat controller to distances, we need to derive the appropriate formulas. Here's how you can perform similar experiments to understand and verify these constants.

Collecting the Data

1. Connect the distance sensor to the wombat
2. Set up the multimeter to measure the voltage output from the sensor
3. Place the sensor at a fixed position and measure various distances (10 cm to 80 cm) using a ruler or measuring tape
4. Record the analog value read by the Wombat for each distance
5. Measure the corresponding voltage output using the multimeter
6. Repeat measurements for accuracy and record the data

Deriving the Formula

Using the collected data:

1. Plot the analog values against the voltages and fit a linear regression to find the conversion constant
2. Plot the voltages against the distances and fit a curve to find the best-fit equation

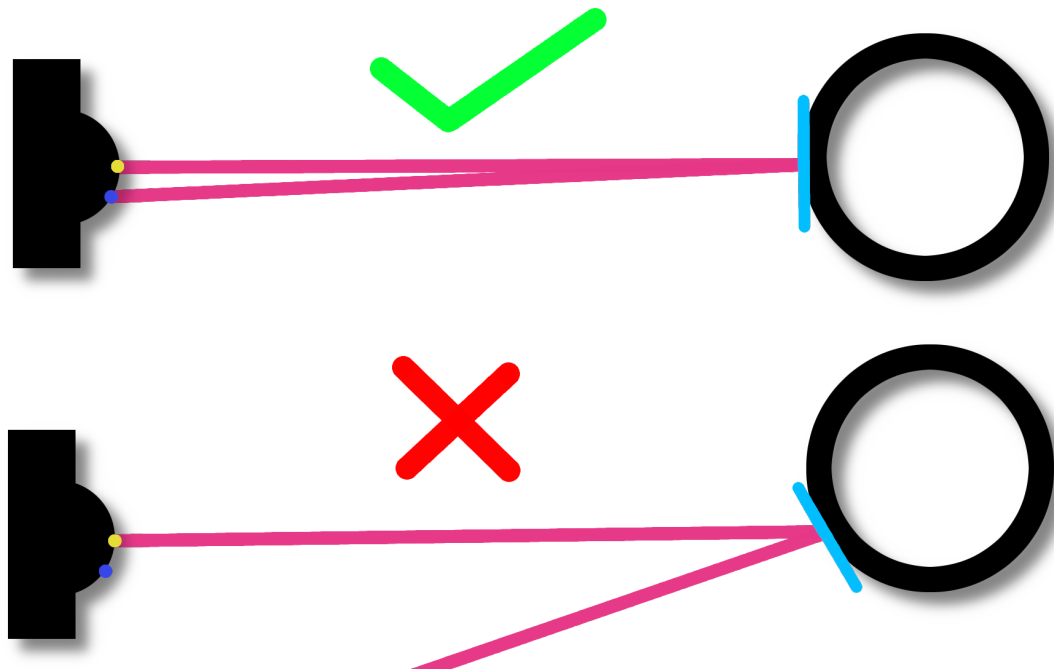
Formulas

- **Analog to Voltage Conversion:** $\text{volt} = \frac{\text{analog}}{1251.215}$
- **Voltage to Distance Conversion:** $\text{distance} = \frac{36.6 - 7 * \text{volt}}{0.2 + \text{volt}}$

Mounting Advice

Importance of Placement

Proper placement of the sensor is crucial to ensure accurate detection. The sensor must be aligned perpendicular to the object to avoid detection errors due to angular reflections.



Distance Sensor Placement Diagram

Placement Guidelines

1. **Perpendicular Alignment:** Ensure the sensor is perpendicular to the surface of the object.
2. **Height Consideration:** The sensor should be at a height where it can effectively detect objects, especially cylindrical objects like PVC pipes.
3. **Avoid Angled Reflections:** If the emitted light hits the object at an angle, the sensor may fail to detect it.

In the provided diagram:

- The upper drawing shows the correct placement where the sensor is aligned to detect the PVC pipe.
- The lower drawing shows an incorrect placement where the sensor is too low and cannot properly detect the object.

Sensor Maintenance

- **Keep the Lens Clean:** Dust, water, or oil can deteriorate the sensor's performance. Ensure the lens is clean.

Environment Considerations

- **Direct Light Exposure:** Avoid exposing the detector to direct sunlight or tungsten lamp light, as it may affect distance measurement accuracy.
- **Reflective Surfaces:** The sensor may not measure the distance accurately if a mirror reflector is used. Adjust the sensor's mounting angle if necessary.
- **Boundary Lines and Moving Directions:** To reduce measurement deviation, ensure the sensor is aligned parallel to any boundary lines and perpendicular to the moving direction of the reflective object.

Programming the Sensor

Reading Sensor Values

One must know, it's not necessary to use the formulas to convert the sensor values to distances. You can directly use the analog values to detect objects within a certain range, but from experience, it's better to convert the values to distances for easier interpretation.

Practical Use cases

Here is an example of how you might read and convert the sensor values in pseudocode:

```
function readDistanceSensor():  
    analogValue = analog(sensorPin)  
    voltage = analogValue / 1251.215  
    distance = (36.6 - 7 * voltage) / (0.2 + voltage)  
    return distance
```

Using the Sensor in Robot Programs

Obstacle Avoidance Example

To use the sensor for obstacle avoidance, you can program your robot to change direction if an object is detected within a certain distance.

```
function driveToObstacle():  
    while True:  
        distance = readDistanceSensor()  
        if distance < 12:  
            stopRobot()  
        else:  
            moveForward()
```

Further Technical Details

For more technical details, download the of the sensor

Authors

Tobias Madlberger

Contact Us



Please report any issues, usability improvements, either by creating an issue in the GitHub repository (<https://github.com/kipr/libwallaby/issues>) or by writing a message on the official botball discord