**Example 4: Using an Ultrasonic Sensor (Distance Measurement)**

**Reference:** "Programming the Raspberry Pi, Second Edition: Getting Started with Python" by Simon Monk

python

import RPi.GPIO as GPIO

import time

# Set GPIO mode

GPIO.setmode(GPIO.BCM)

# Define GPIO pins

TRIG = 23

ECHO = 24

# Set up the GPIO pins

GPIO.setup(TRIG, GPIO.OUT)

GPIO.setup(ECHO, GPIO.IN)

def get\_distance():

# Trigger the sensor

GPIO.output(TRIG, True)

time.sleep(0.00001)

GPIO.output(TRIG, False)

# Wait for the echo to start and stop

while GPIO.input(ECHO) == 0:

start\_time = time.time()

while GPIO.input(ECHO) == 1:

end\_time = time.time()

# Calculate distance

duration = end\_time - start\_time

distance = (duration \* 34300) / 2

return distance

try:

while True:

distance = get\_distance()

print(f'Distance: {distance:.2f} cm')

time.sleep(1)

except KeyboardInterrupt:

GPIO.cleanup()

**More details:** Refer to "Programming the Raspberry Pi, Second Edition: Getting Started with Python" for detailed explanations on using various sensors.

**Example 5: Using a Servo Motor (Actuator)**

**Reference:** "Python Programming for Arduino" by Pratik Desai

python

Copy code

import RPi.GPIO as GPIO

import time

# Set GPIO mode

GPIO.setmode(GPIO.BCM)

# Define GPIO pin

SERVO\_PIN = 18

# Set up the GPIO pin

GPIO.setup(SERVO\_PIN, GPIO.OUT)

# Set up PWM

pwm = GPIO.PWM(SERVO\_PIN, 50) # 50 Hz

pwm.start(0)

def set\_angle(angle):

duty\_cycle = angle / 18 + 2

GPIO.output(SERVO\_PIN, True)

pwm.ChangeDutyCycle(duty\_cycle)

time.sleep(1)

GPIO.output(SERVO\_PIN, False)

pwm.ChangeDutyCycle(0)

try:

while True:

set\_angle(90) # Set servo to 90 degrees

time.sleep(2)

set\_angle(0) # Set servo to 0 degrees

time.sleep(2)

except KeyboardInterrupt:

pwm.stop()

GPIO.cleanup()

**More details:** Refer to "Python Programming for Arduino" for detailed explanations on using actuators like servo motors.

**Example 6: Using a Light Sensor (Photoresistor)**

**Reference:** "Python Programming for Arduino" by Pratik Desai

python

Copy code

import RPi.GPIO as GPIO

import time

# Set GPIO mode

GPIO.setmode(GPIO.BCM)

# Define GPIO pin

LIGHT\_PIN = 18

# Set up the GPIO pin

GPIO.setup(LIGHT\_PIN, GPIO.IN)

def read\_light():

return GPIO.input(LIGHT\_PIN)

try:

while True:

light\_status = read\_light()

if light\_status == 0:

print("Light detected")

else:

print("No light detected")

time.sleep(1)

except KeyboardInterrupt:

GPIO.cleanup()

**More details:** Refer to "Python Programming for Arduino" for detailed explanations on using sensors like photoresistors.