\

**TITLE:** “Silo Level Sensor”

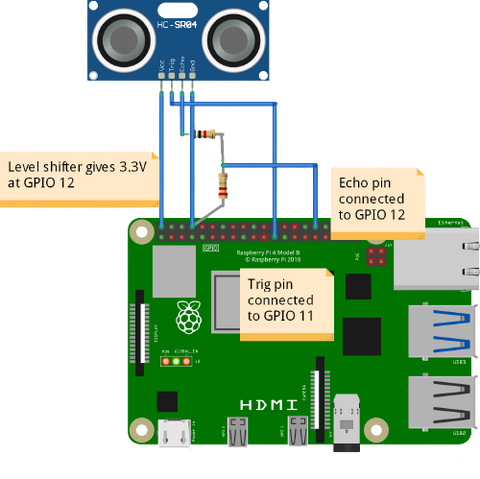
**OBJECTIVE:** “To optimize inventory management, prevent overfilling or underfilling of silos and minimize operational risks.”

**COMPONENTS REQUIRED:**

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Name of the Component** | **Quantity** |
| 1. | Raspberry PI | 1 |
| 2. | Ultrasonic Sensor | 1 |
| 3. | Jumper Cables | 4 |

**CONNECTIONS:**

1. The trigger pin of the ultrasonic sensor goes to the GPIO 11 (physical PIN is 23) of RPI via the level shifter.
2. The echo pin of the ultrasonic sensor goes to GPIO 12 (physical PIN 32) of RPI.
3. The 5V and GND pin of ultrasonic sensor is connected to 5V and GND of Raspberry Pi respectively.



**CODE:**

import RPi.GPIO as GPIO

import time

TRIG\_PIN = 23

ECHO\_PIN = 24

THRESHOLD\_DISTANCE = 100

GPIO.setmode(GPIO.BCM)

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

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

def measure\_distance():

    GPIO.output(TRIG\_PIN, False)

    time.sleep(0.1)

    GPIO.output(TRIG\_PIN, True)

    time.sleep(0.00001)

    GPIO.output(TRIG\_PIN, False)

    while GPIO.input(ECHO\_PIN) == 0:

        pulse\_start = time.time()

    while GPIO.input(ECHO\_PIN) == 1:

        pulse\_end = time.time()

    pulse\_duration = pulse\_end - pulse\_start

    distance = pulse\_duration \* 17150

    distance = round(distance, 2)

    return distance

try:

    while True:

        distance = measure\_distance()

        print("Measured Distance:", distance, "cm")

        if distance > THRESHOLD\_DISTANCE:

            print("Grain level is below the threshold. Take action!")

        time.sleep(1)

except KeyboardInterrupt:

    print("Exiting program")

finally:

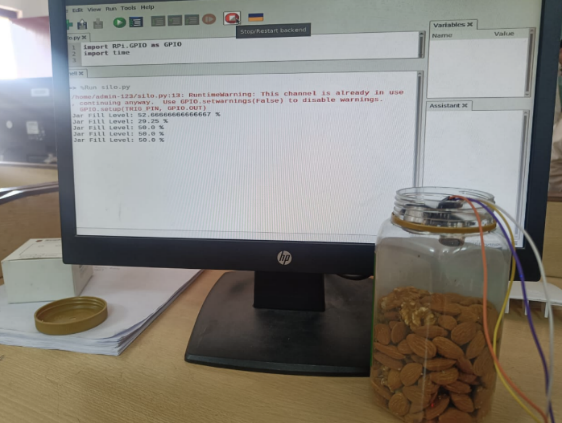
    GPIO.cleanup()

**CODE EXPLANATION:**

1. **Measure Distance Function:** The measure\_distance() function sends a trigger pulse to the ultrasonic sensor, measures the time it takes for the echo pulse to return, and converts it to distance in centimeters.
2. **Main Loop:**
   1. The script enters a continuous loop to repeatedly measure the distance using the **measure\_distance()** function.
   2. It prints the measured distance
   3. If the measured distance exceeds the threshold distance set earlier, it prints a message indicating that the grain level is below the threshold and suggests taking action.

**WORKING:**

1. In air, sound travels at a speed of 343 metres per second.
2. An ultrasonic distance sensor sends out pulses of ultrasound which are inaudible to humans, and detects the echo that is sent back when the sound bounces off a nearby object.
3. It then uses the speed of sound to calculate the distance from the object.
4. The ultrasonic sensor detects the distance between the top layer of the silo and gives the value to the raspberry pi.
5. The code then divides the value of the distance of the top layer and the height of the silo multiplied by 100 which gives the percentage of silo that is filled with the grain.

**SCREENSHOT OF WORKING CIRCUIT:**

**OBSERVATION TABLE:**

|  |  |  |
| --- | --- | --- |
| **Level Filled Up to** | **Sensor Value** | **Percentage shown** |
| 32 cm | 31-32 cm | 96.00 - 99.99 percent |
| 26 cm | 25-26 cm | 78.125 - 81.25 percent |
| 20 cm | 19-21 cm | 59.375 - 65.625 percent |

**RESULT ANALYSIS:**

1. When the jar was fully filled, the value shown by the Raspberry PI was 96-99 percent.
2. When the jar was emptied to 80 percent, value shown by the Raspberry PI was 78-81 percent.
3. When the jar was emptied further, value shown by the Raspberry PI was 59-66 percent.

**CONCLUSION:**

Ultrasonic sensor is used to detect the silo volume without having to be there in person with a 96 percent accuracy.