Student name: Rammaka Aaron Iddamalgoda

Student ID: s223496576

SIT225: Data Capture Technologies

Link to video: https://deakin.au.panopto.com/Panopto/Pages/Viewer.aspx?id=a87af578-4c48-4a33-9e2b-b1fe0100c040

Activity 2.2: Working with sensor - HC-SR04

Hypothesis:

Moving the sensor to a path with decent foot traffic for 30 minutes will result in the following patterns in the sensor data:

- **Higher foot traffic** will correlate with a reduction in the distance values recorded by the sensor, as more people walk past the sensor.
- **Less foot traffic** will result in relatively constant distance values.

Observing the Data Changing Pattern:

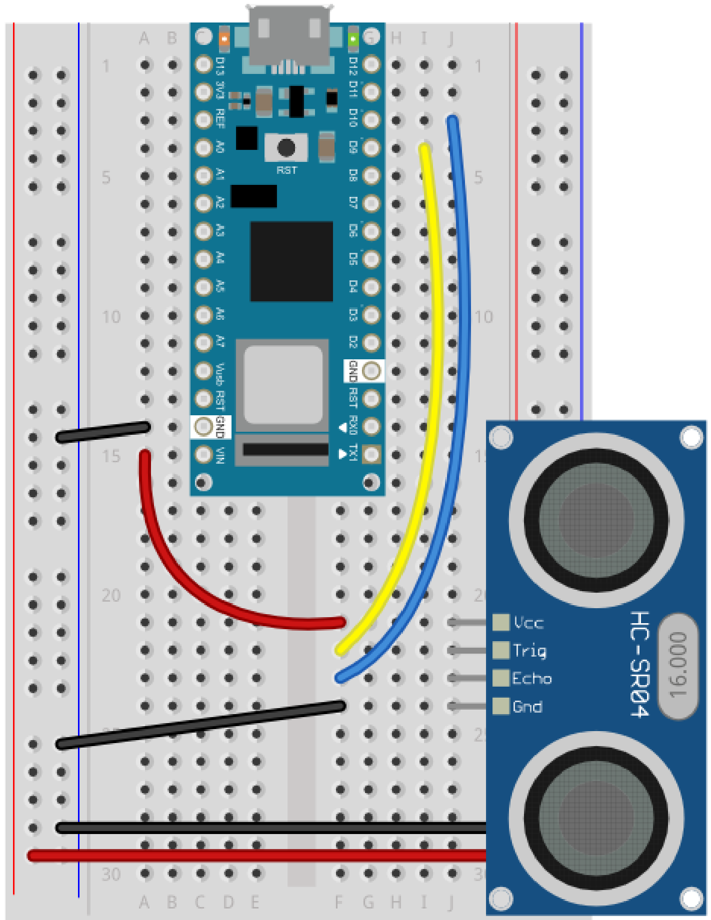
From the actual sensor data:

- When no one walked past, the distance was around **240 cm** (to the wall).
- Most people walked by the distance ranged between **150–200 cm**.
- Only a handful of people passed very close, the distance dropped below **50 cm**.

This suggests:

1. **Up and Down Trends**: Observed clear downward spikes when people walk closer to the sensor, followed by an upward trend as they move away.
2. **Repeating Patterns**: If there is frequent foot traffic, periodic downward spikes followed by a return to the baseline distance (around 240 cm).

Step	Action
1	Connect your HC-SR04 sensor to the Arduino board. Note that the pin layout in the image below may look different than the board you may have.

<table>
<tr><td></td><td>
</td></tr>
<tr><td>2</td><td>Connect your Arduino board to your computer using the USB cable.</td></tr>
<tr><td>3</td><td>Write an Arduino sketch const int trigger = 2;
const int echo = 3;

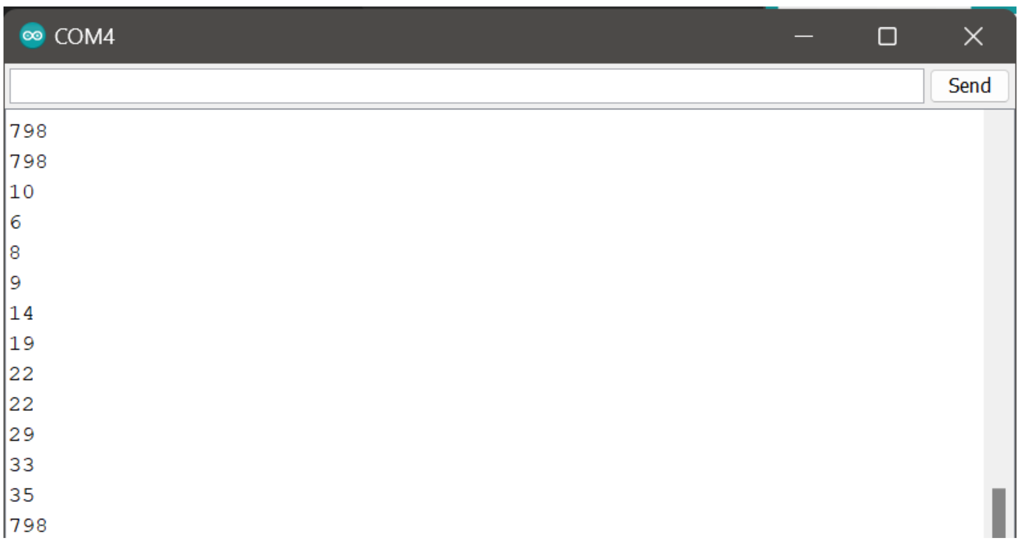
int getUltrasonicDistance() {
 long duration;
 int distance;

 digitalWrite(trigger, LOW);
 delayMicroseconds(5);

 digitalWrite(trigger, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigger, LOW);

 duration = pulseIn(echo, HIGH);
 distance = duration * 0.034 / 2;</td></tr>
</table>

	`` return distance; } void setup() { pinMode(trigger, OUTPUT); pinMode(echo, INPUT); Serial.begin(9600); } void loop() { Serial.print("Distance: "); Serial.println(getUltrasonicDistance()); delay(1000); } ``` Compile the code in Arduino IDE, deploy to the board and observe output in the Arduino IDE serial monitor. ``` py script import serial import time import csv # Setup the serial connection (adjust COM port as necessary) arduino = serial.Serial('COM3', 9600, timeout=1) time.sleep(2) # Allow time for Arduino to initialize # Open a CSV file to store the data with open('distance_data.csv', 'w', newline='') as file: writer = csv.writer(file) writer.writerow(["Timestamp", "Distance"]) # CSV headers while True: data = arduino.readline().decode('utf-8').strip() # Read from Arduino if data: timestamp = time.strftime("%Y%m%d%H%M%S") # Timestamp in required format writer.writerow([timestamp, data]) # Save to CSV print(f"{timestamp}, {data}") console time.sleep(1) ```

4	Question: Spec of SR04 is available here (https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf). Identify 2 critical aspects you should be careful about this sensor operation. Answer: **Voltage Requirement**: • The HC-SR04 operates at **5V**, but some Arduino boards (like my Arduino Nano 33 IoT) operate at **3.3V**. Therefore, ensuring that the sensor is connected to a compatible 5V supply is necessary. ☐ **Maximum Distance and Measurement Range**: • The HC-SR04 has a **maximum range of 4 meters**. Any object further than this distance will not be detected correctly, and the sensor may output inaccurate or zero values.
5	Question: displaying distance values while generating a periodic motion by moving your hand gradually back and forth towards the sensor. 

Activity 2.4: Plot data using Python Notebook