NAMAN SRIVASTAVA

AIM:

To measure distance using ultrasonic sensor with Arduino and transmit data wirelessly between two Raspberry Pi units to control LEDs based on distance.

PROCEDURE:

Arduino and Ultrasonic Sensor Setup:

- Connect Trig pin to pin 10 of Arduino
- Connect Echo pin to pin 11 of Arduino
- Connect GND pin to GND of Arduino
- Connect VCC pin to 5V of Arduino
- Upload code to Arduino using Arduino IDE

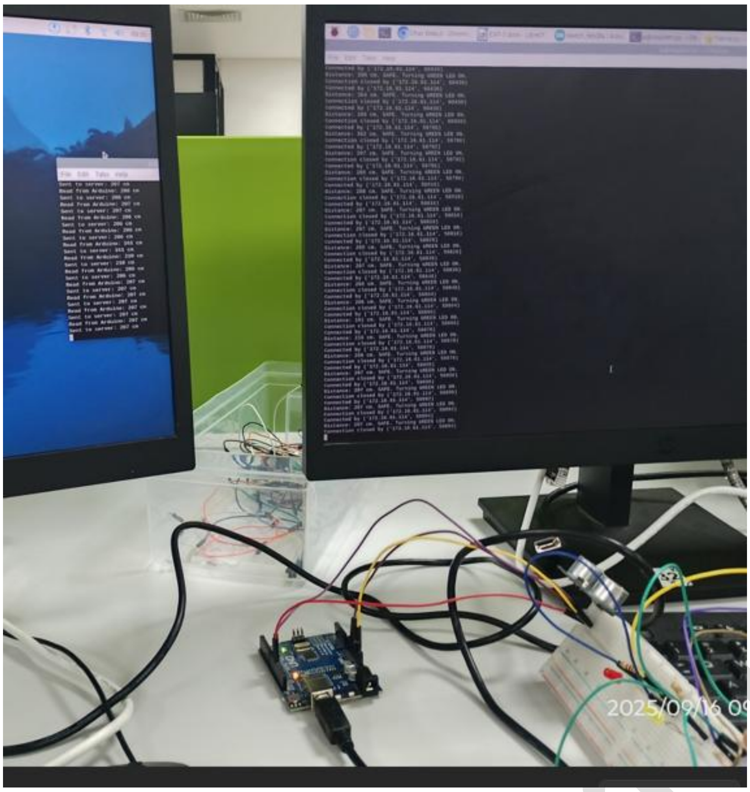
Raspberry Pi 1 (Client) Setup:

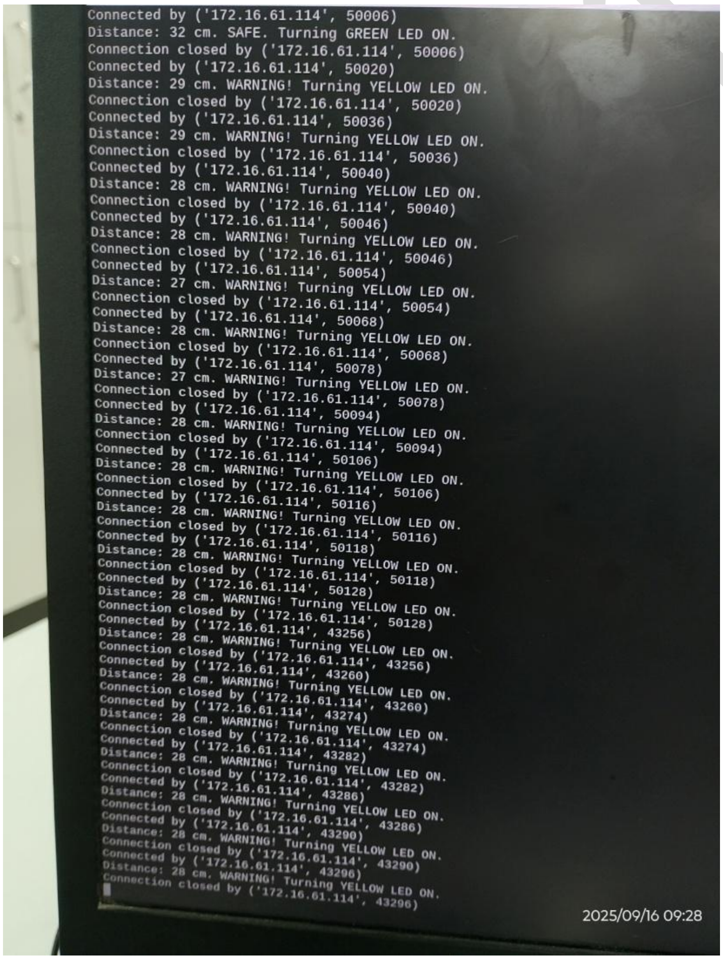
- Connect Arduino to RPi via USB cable
- Install Python serial library
- Read distance data from Arduino
- Send data wirelessly to Raspberry Pi 2

Raspberry Pi 2 (Server) Setup:

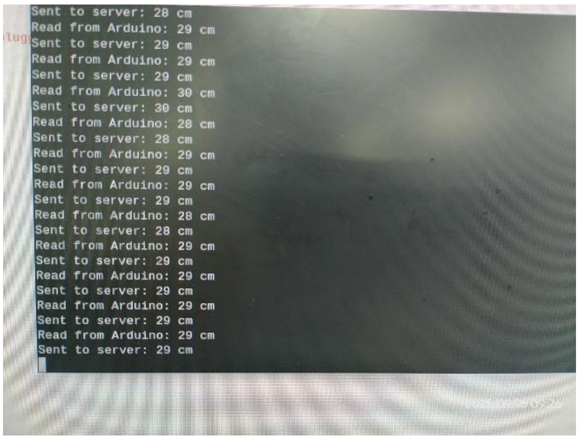
- Connect LEDs to GPIO pins with resistors
- Create socket server to receive data
- Control LEDs based on distance values

Photo

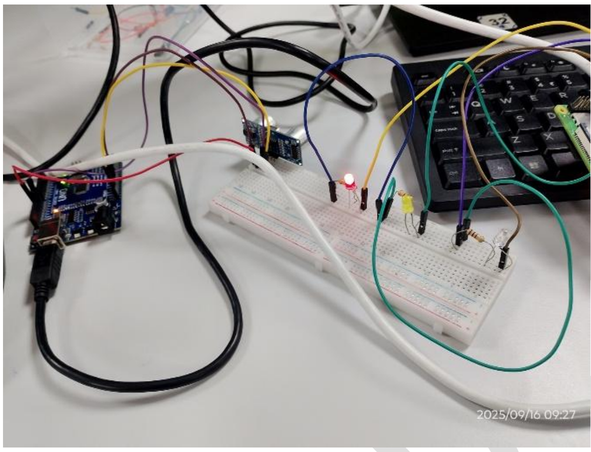




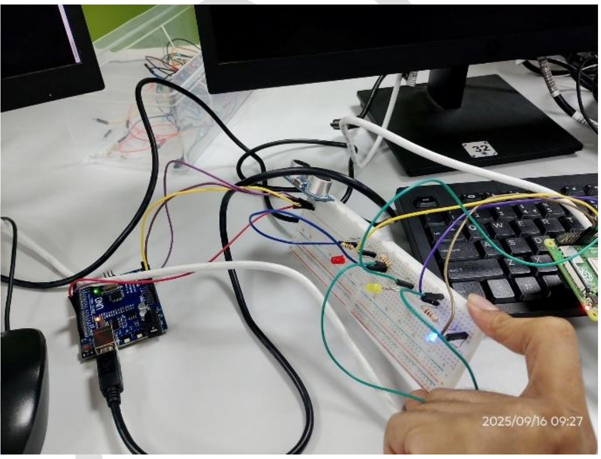
```
Connected by ('172.16.61.114', 50006)
Distance: 32 cm. SAFE. Turning GREEN LED ON.
Connection closed by ('172.16.61.114', 50006)
Connected by ('172.16.61.114', 50020)
Distance: 29 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50020)
Connected by ('172.16.61.114', 50036)
Distance: 29 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50036)
Connected by ('172.16.61.114', 50040)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50040)
Connected by ('172.16.61.114', 50046)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50046)
Connected by ('172.16.61.114', 50054)
Distance: 27 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50054)
Connected by ('172.16.61.114', 50068)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50068)
Connected by ('172.16.61.114', 50078)
Distance: 27 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50078)
Connected by ('172.16.61.114', 50094)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50094)
Connected by ('172.16.61.114', 50106)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50106)
Connected by ('172.16.61.114', 50116)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50116)
Connected by ('172.16.61.114', 50118)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50118)
Connected by ('172.16.61.114', 50128)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 50128)
Connected by ('172.16.61.114', 43256)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 43256)
Connected by ('172.16.61.114', 43260)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 43260)
Connected by ('172.16.61.114', 43274)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 43274)
Connected by ('172.16.61.114', 43282)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 43282)
Connected by ('172.16.61.114', 43286)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 43286)
Connected by ('172.16.61.114', 43290)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 43290)
Connected by ('172.16.61.114', 43296)
Distance: 28 cm. WARNING! Turning YELLOW LED ON.
Connection closed by ('172.16.61.114', 43296)
```



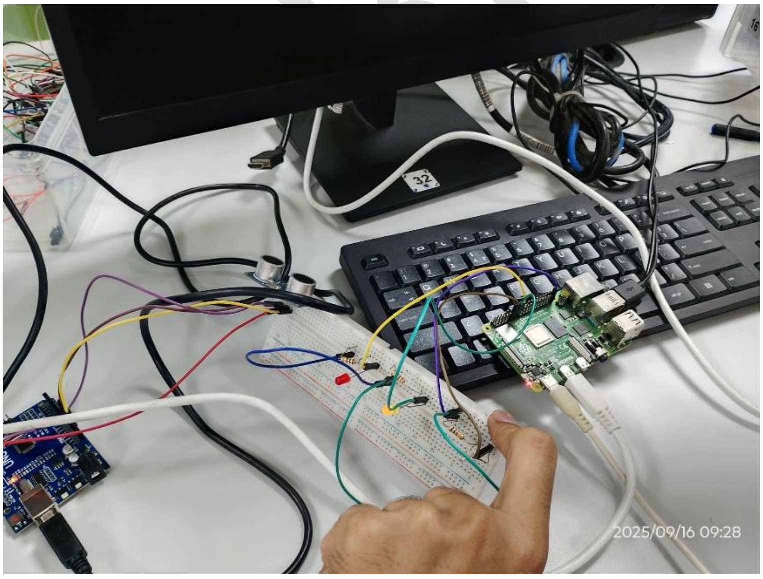
```
Sent to server: 28 cm
Read from Arduino: 29 cm
Sent to server: 29 cm
Read from Arduino: 29 cm
Sent to server: 29 cm
Read from Arduino: 30 cm
Sent to server: 30 cm
Read from Arduino: 28 cm
Sent to server: 28 cm
Read from Arduino: 29 cm
Sent to server: 29 cm
Read from Arduino: 29 cm
Sent to server: 29 cm
Read from Arduino: 28 cm
Sent to server: 28 cm
Read from Arduino: 29 cm
Sent to server: 29 cm
Read from Arduino: 29 cm
Sent to server: 29 cm
Read from Arduino: 29 cm
Sent to server: 29 cm
Read from Arduino: 29 cm
Sent to server: 29 cm
```



2025/09/16 09:27



2025/09/16 09:27



2025/09/16 09:28

ARDUINO CODE:

```
#define TRIG_PIN 10
#define ECHO_PIN 11

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

void loop() {
  long duration, distance;

  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  duration = pulseIn(ECHO_PIN, HIGH);
  distance = duration * 0.034 / 2;

  Serial.println(distance);
  delay(500);
}
```

RASPBERRY PI CLIENT CODE:

python

```
import serial
import socket
```

```
arduino = serial.Serial('/dev/ttyUSB0', 9600)
client_socket = socket.socket()
client_socket.connect(('192.168.1.100', 8888))

while True:
    if arduino.in_waiting > 0:
        distance = arduino.readline().decode().strip()
        client_socket.send(distance.encode())
```

RASPBERRY PI SERVER CODE:

python

```
import socket
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setup([18, 19, 20], GPIO.OUT)

server_socket = socket.socket()
server_socket.bind(('0.0.0.0', 8888))
server_socket.listen(1)

client, addr = server_socket.accept()

while True:
    data = client.recv(1024).decode()
    distance = int(data)

    GPIO.output([18, 19, 20], GPIO.LOW)

    if distance < 10:
        GPIO.output(18, GPIO.HIGH)  # Red LED
    elif distance < 30:
        GPIO.output(19, GPIO.HIGH)  # Yellow LED
    else:
        GPIO.output(20, GPIO.HIGH)  # Green LED
```

RESULT:

Successfully implemented client-server system using Arduino for distance measurement and Raspberry Pi units for wireless communication and LED control based on proximity detection.