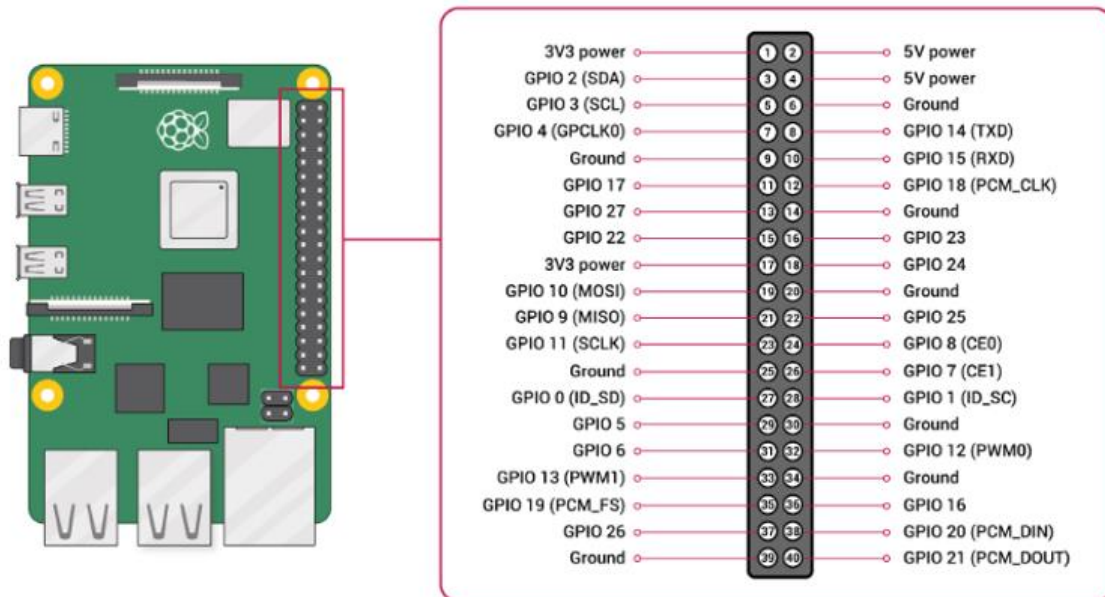


NAME: ANANYA PILLAI

1) RASPBERRY PI – ULTRASONIC SENSOR

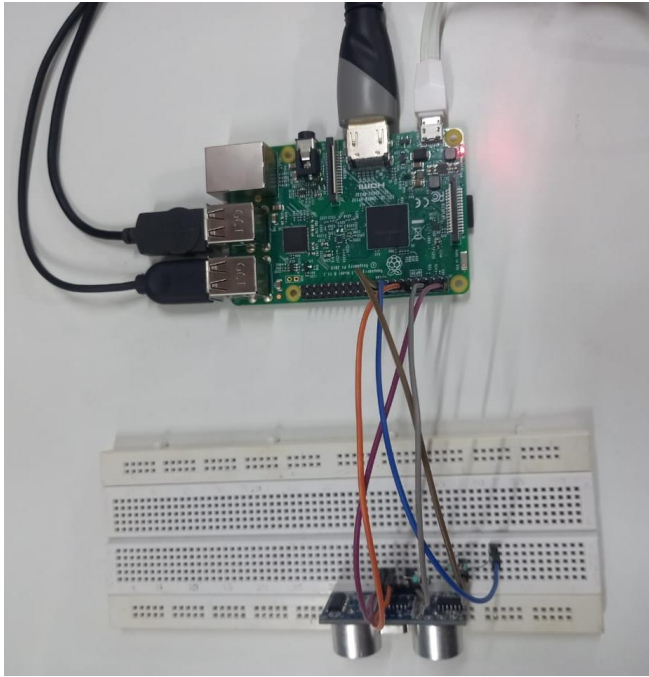


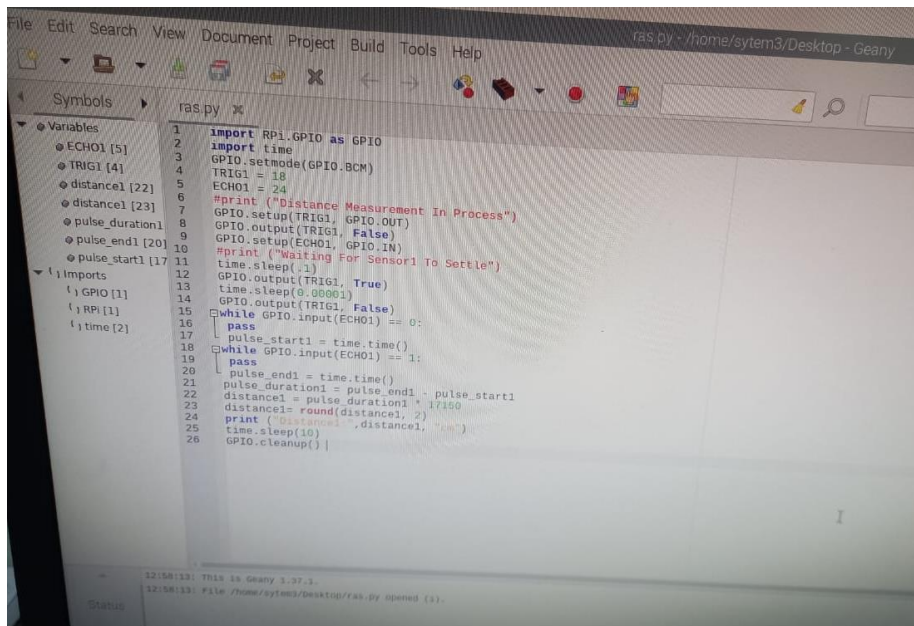
CODE:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
TRIG1 = 18
ECHO1 = 24
#print ("Distance Measurement In Process")
GPIO.setup(TRIG1, GPIO.OUT)
GPIO.output(TRIG1, False)
GPIO.setup(ECHO1, GPIO.IN)
#print ("Waiting For Sensor1 To Settle")
time.sleep(.1)
GPIO.output(TRIG1, True)
time.sleep(0.00001)
GPIO.output(TRIG1, False)
while GPIO.input(ECHO1) == 0:
```

```
    pass
    pulse_start1 = time.time()
while GPIO.input(ECHO1) == 1:
    pass
    pulse_end1 = time.time()
pulse_duration1 = pulse_end1 - pulse_start1
distance1 = pulse_duration1 * 17150
distance1= round(distance1, 2)
print ("Distance1:",distance1, "cm")
time.sleep(10)
GPIO.cleanup()
```

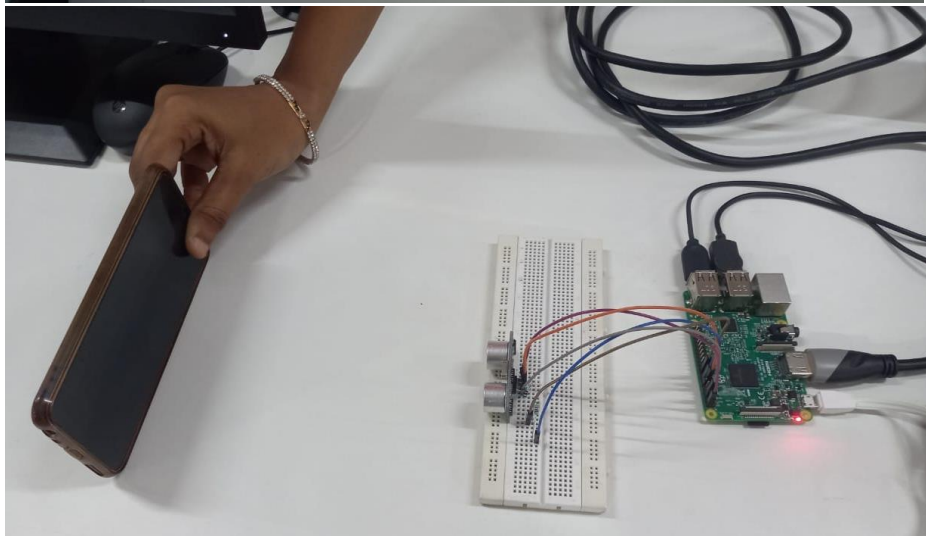
CONNECTIONS AND SETUP:



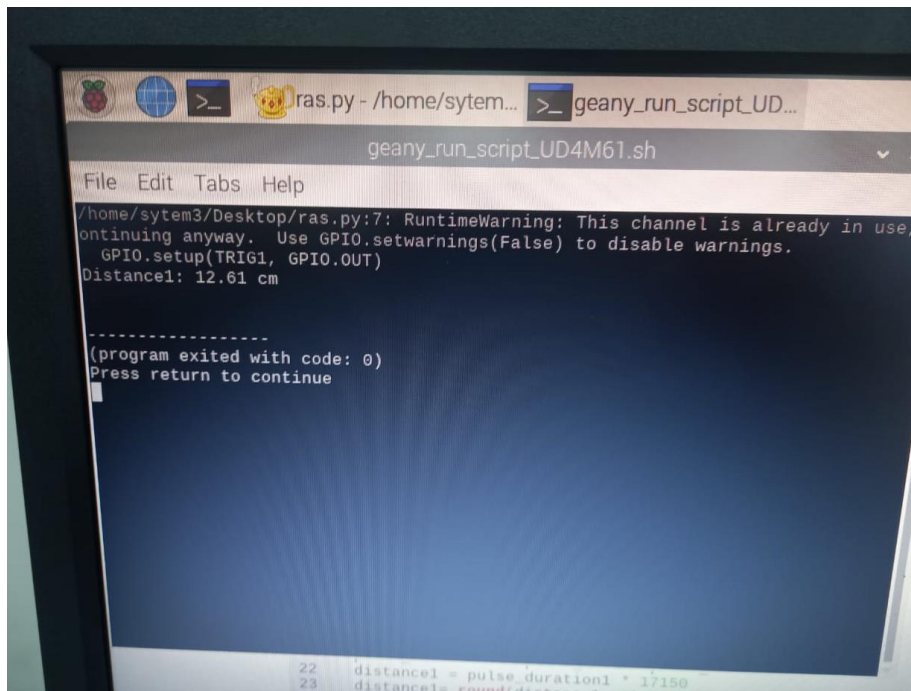


The screenshot shows a Geany IDE window with a Python script named `ras.py`. The script is designed to interface with an ultrasonic sensor (HC-SR04) on a Raspberry Pi. It imports the `RPi.GPIO` and `time` modules. The sensor's trigger pin is set to GPIO 18 and the echo pin to GPIO 24. The script prints "Distance Measurement In Process" and "Waiting For Sensor1 To Settle" before sending a 10µs pulse to the trigger pin. It then enters a loop where it waits for the echo pin to go high, measures the pulse duration, and calculates the distance using the formula $\text{distance} = \text{pulse_duration} \times 17150$. The distance is printed in centimeters, and the script sleeps for 10 seconds before cleaning up the GPIO pins.

```
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BCM)
4 TRIG1 = 18
5 ECHO1 = 24
6 #print ("Distance Measurement In Process")
7 GPIO.setup(TRIG1, GPIO.OUT)
8 GPIO.output(TRIG1, False)
9 GPIO.setup(ECHO1, GPIO.IN)
10 #print ("Waiting For Sensor1 To Settle")
11 time.sleep(1)
12 GPIO.output(TRIG1, True)
13 time.sleep(0.00001)
14 GPIO.output(TRIG1, False)
15 while GPIO.input(ECHO1) == 0:
16     pass
17     pulse_start1 = time.time()
18 while GPIO.input(ECHO1) == 1:
19     pass
20     pulse_end1 = time.time()
21     pulse_duration1 = pulse_end1 - pulse_start1
22     distance1 = pulse_duration1 * 17150
23     distance1 = round(distance1, 2)
24     print ("Distance1: ", distance1, "cm")
25     time.sleep(10)
26 GPIO.cleanup()
```

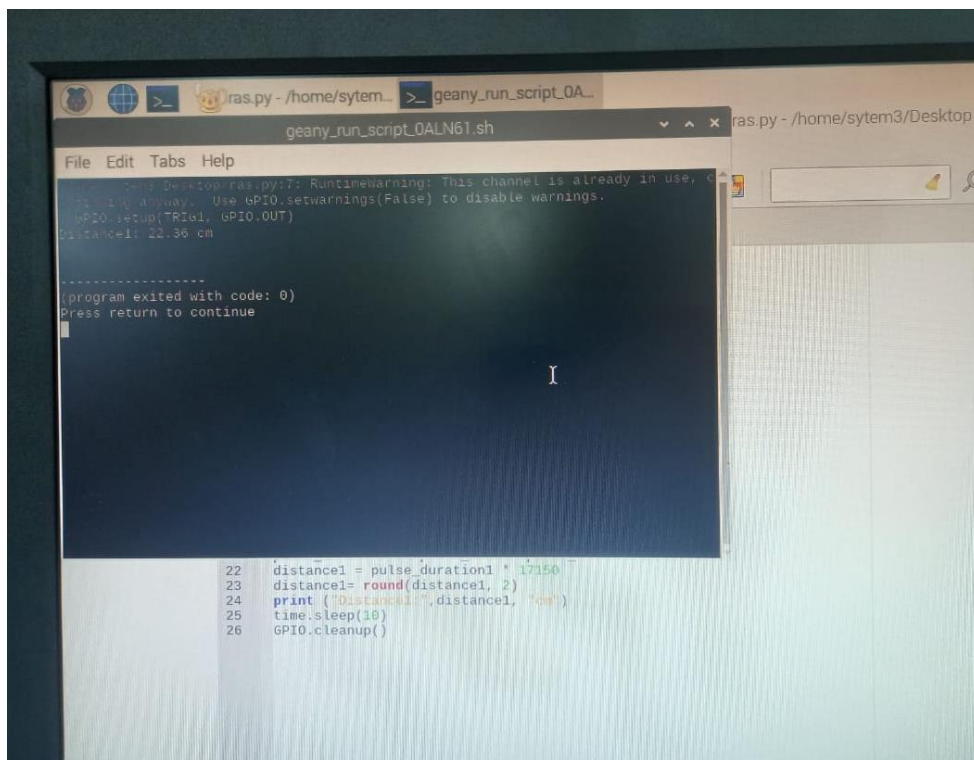


OUTPUT:



```
geany_run_script_UD4M61.sh
File Edit Tabs Help
/home/sytem3/Desktop/ras.py:7: RuntimeWarning: This channel is already in use,
continuing anyway. Use GPIO.setwarnings(False) to disable warnings.
GPIO.setup(TRIG1, GPIO.OUT)
Distance1: 12.61 cm

-----
(program exited with code: 0)
Press return to continue
```



```
geany_run_script_OALN61.sh
File Edit Tabs Help
/home/sytem3/Desktop/ras.py:7: RuntimeWarning: This channel is already in use,
continuing anyway. Use GPIO.setwarnings(False) to disable warnings.
GPIO.setup(TRIG1, GPIO.OUT)
Distance1: 22.36 cm

-----
(program exited with code: 0)
Press return to continue
```

2) NODE MC

CODE:

```
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
//----- Enter you Wi-Fi Details-----//
char ssid[] = "vivo 1904"; //SSID
char pass[] = "resh7604"; // Password
//-----//

const int trigger = 16;
const int echo = 5;
long T;
float distanceCM;
WiFiClient client;
unsigned long myChannelField = 2204739; // Channel ID
const int ChannelField = 1; // Which channel to write data
const char * myWriteAPIKey = " LRFT3R8SP6RDUPV1"; // Your write API Key

void setup()
{
  Serial.begin(9600);
  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);
}

void loop()
{
  if (WiFi.status() != WL_CONNECTED)
  {
```

```
Serial.print("Attempting to connect to SSID: ");
Serial.println(ssid);
while (WiFi.status() != WL_CONNECTED)
{
  WiFi.begin(ssid, pass);
  Serial.print(".");
  delay(5000);
}
Serial.println("\nConnected.");
}
digitalWrite(trigger, LOW);
delay(1);
digitalWrite(trigger, HIGH);
delayMicroseconds(10);
digitalWrite(trigger, LOW);
T = pulseIn(echo, HIGH);
distanceCM = T * 0.034;
distanceCM = distanceCM / 2;
Serial.print("Distance in cm: ");
Serial.println(distanceCM);
ThingSpeak.writeField(myChannelField, ChannelField, distanceCM,
myWriteAPIKey);
delay(1000);}
```

OUTPUT:

```
sketch_jun13a.ino
1  #include "ThingSpeak.h"
2  #include <ESP8266WiFi.h>
3
4  //----- Enter you Wi-Fi Details-----//
5  char ssid[] = "vivo 1904"; //SSID
6  char pass[] = "resh7604"; // Password
7  //-----//
8
9  const int trigger = 16;
10 const int echo = 5;
11 long T;
12 float distanceCM;
13 WiFiClient client;
14
15 unsigned long myChannelField = 2204739; // Channel ID
16 const int ChannelField = 1; // Which channel to write data
17 const char * myWriteAPIKey = "LRFT3R8SP6RDUPV1"; // Your write API Key
18
19 void setup()
20 {
21   Serial.begin(9600);
22   pinMode(trigger, OUTPUT);
23   pinMode(echo, INPUT);
24   WiFi.mode(WIFI_STA);
25   ThingSpeak.begin(client);
26 }
27 void loop()
28 {
29   if (WiFi.status() != WL_CONNECTED)
30   {
31     Serial.print("Attempting to connect to SSID: ");
32     Serial.println(ssid);
33     while (WiFi.status() != WL_CONNECTED)
34     {
35       WiFi.begin(ssid, pass);
```

```
19 void setup()
20 {
21   Serial.begin(9600);
22   pinMode(trigger, OUTPUT);
23   pinMode(echo, INPUT);
24   WiFi.mode(WIFI_STA);
25   ThingSpeak.begin(client);
26 }
27 void loop()
28 {
29   if (WiFi.status() != WL_CONNECTED)
30   {
31     Serial.print("Attempting to connect to SSID: ");
32     Serial.println(ssid);
33     while (WiFi.status() != WL_CONNECTED)
34     {
35       WiFi.begin(ssid, pass);
36       Serial.print(".");
37       delay(5000);
38     }
39     Serial.println("\nConnected.");
40   }
41   digitalWrite(trigger, LOW);
42   delay(1);
43   digitalWrite(trigger, HIGH);
44   delayMicroseconds(10);
45   digitalWrite(trigger, LOW);
46   T = pulseIn(echo, HIGH);
47   distanceCM = T * 0.034;
48   distanceCM = distanceCM / 2;
49   Serial.print("Distance in cm: ");
50   Serial.println(distanceCM);
51   ThingSpeak.writeField(myChannelField, ChannelField, distanceCM, myWriteAPIKey);
52   delay(1000);
53 }
```



```
19 void setup()
20 {
21   Serial.begin(9600);

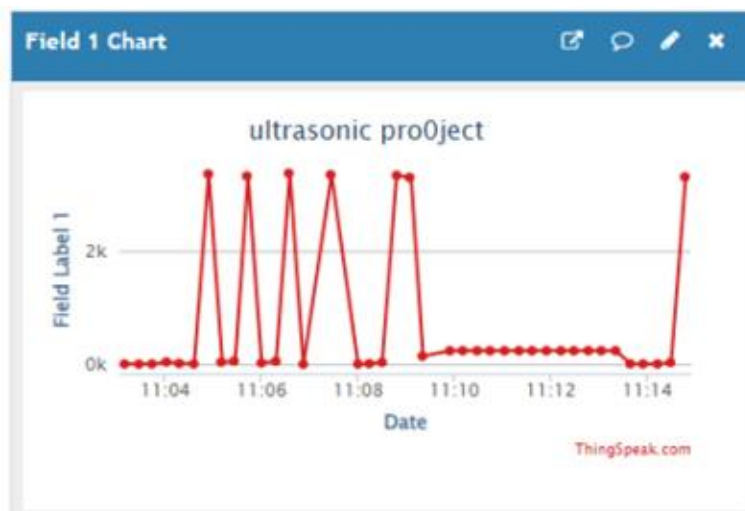
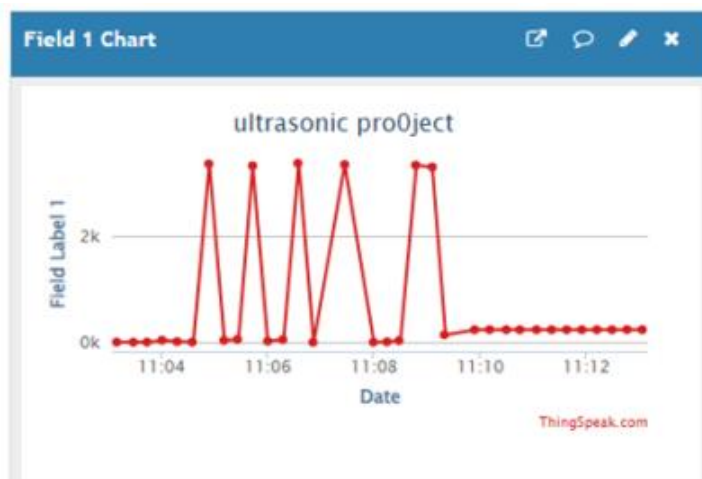
```

Output Serial Monitor x

Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM12')

New Line 9600 baud

.....Attempting to connect to SSID: vivo 1904
.
Connected.
Distance in cm: 5.63
Distance in cm: 3402.74
Distance in cm: 5.13
Distance in cm: 9.83
Distance in cm: 3346.77
Distance in cm: 10.63
Distance in cm: 2.57
Distance in cm: 3.25
Distance in cm: 6.17
Distance in cm: 3.69
Distance in cm: 3.45
Distance in cm: 4.22
Distance in cm: 2.52
Distance in cm: 5.86
Distance in cm: 3.98
Distance in cm: 5.47
Distance in cm: 15.01
Distance in cm: 6.97
Distance in cm: 4.49
Distance in cm: 42.08
Distance in cm: 9.81
Distance in cm: 18.14
Distance in cm: 15.62
Distance in cm: 15.62
Distance in cm: 15.64
Distance in cm: 15.62
Distance in cm: 39.07
Distance in cm: 3347.22
Distance in cm: 3.98



THANK YOU !

