Introduction to IOT

LAB ACTIVITY – 2

Institute of Engineering and Technology

JK Lakshmipat University, Jaipur

SUBMITTED TO

Mr. Divanshu Jain

SUBMITTED BY

Garv Baheti

2020BTechCSE031

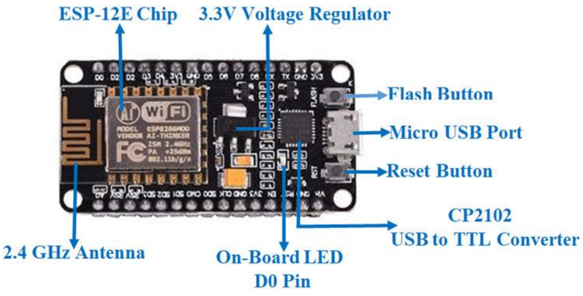
August 29, 2022

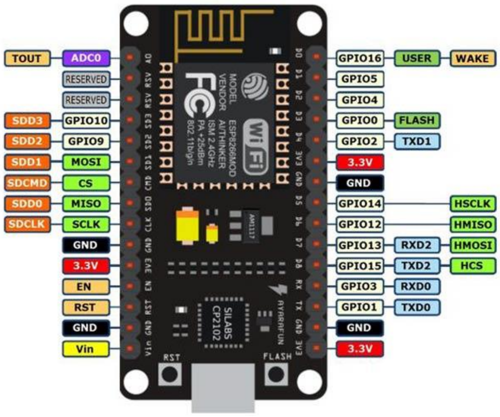
OBJECTIVE

1. To blink LED when the intensity of room changes using LDR
2. To turn on a buzzer/LED when an object is detected using IR sensor in analog mode.
3. To turn on a buzzer/LED when an object is detected using IR sensor in digital mode.
4. Calculate the distance of an object using Ultrasonic Sensor and control a led and show the value on serial monitor.

THEORY

NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The **NodeMCU ESP8266 development board** comes with the ESP-12E module containing the ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects. NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.





NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

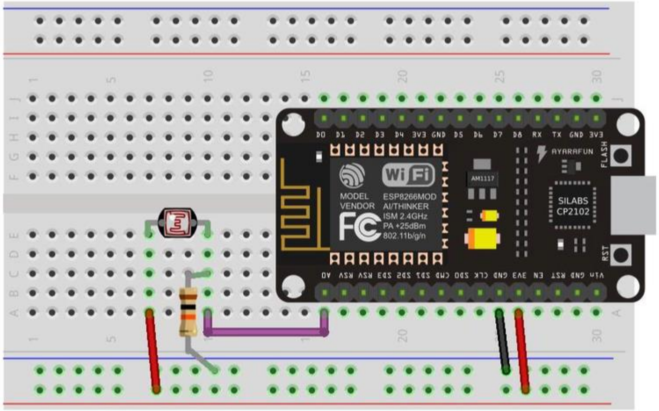
NodeMCU Development Board Pinout Configuration

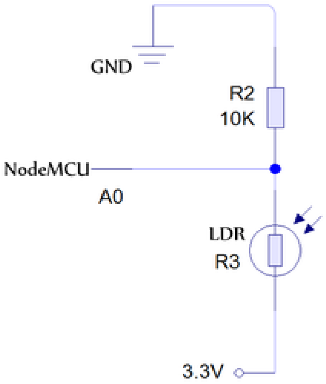
Pin Category	**Name**	**Description**
Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** Regulated 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	**EN, RST**	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

TO BLINK LED WHEN THE INTENSITY OF ROOM CHANGES USING LDR

REQUIREMENTS

- NodeMCU x 1
- Micro USB cable x 1
- PC x 1
- Software Arduino IDE(version 1.6.4+)
- LDR Sensor
- LED
- Resistor



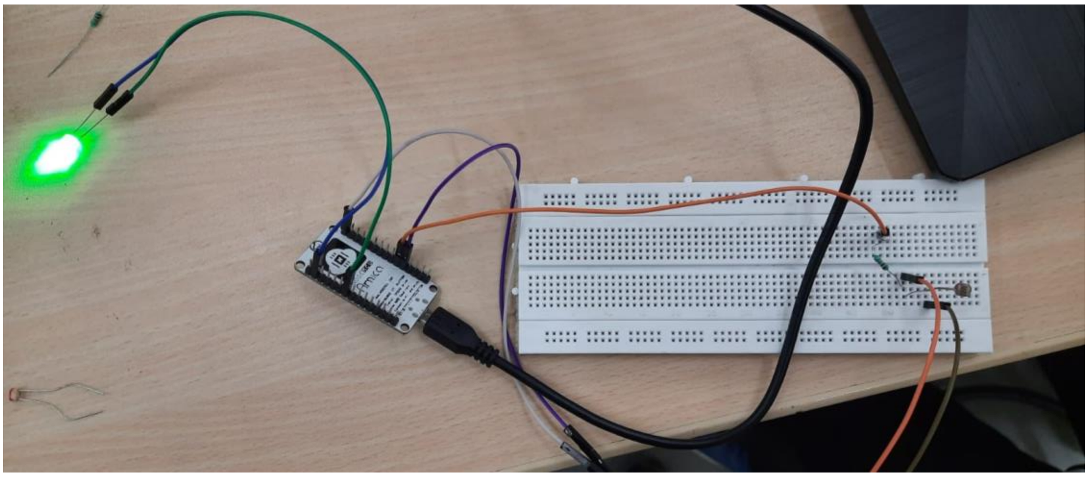


CODE

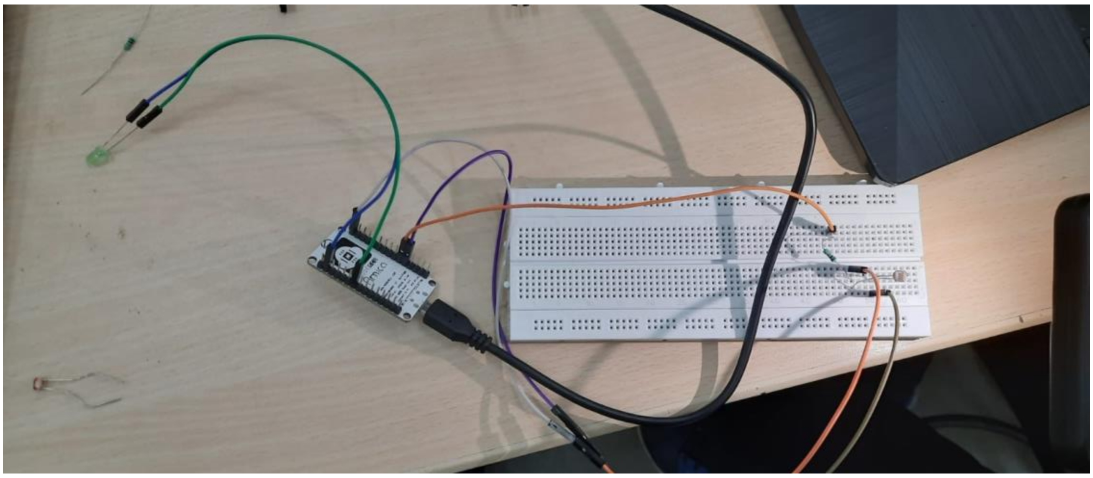
```
const int ledPin = 5;
const int ldrPin = A0;

void setup() {
Serial.begin(9600);
pinMode(ledPin, OUTPUT);
pinMode(ldrPin, INPUT);
}

void loop() {
int ldrStatus = analogRead(ldrPin);
Serial.print(ldrStatus);

  if (ldrStatus <=300) {
  digitalWrite(ledPin, HIGH);
  Serial.println("LDR is DARK, LED is ON");
  }
  else {
  digitalWrite(ledPin, LOW);
  Serial.println("LED is OFF");

  }
}
```

RESULT

When No Light is detected



When Light is detected

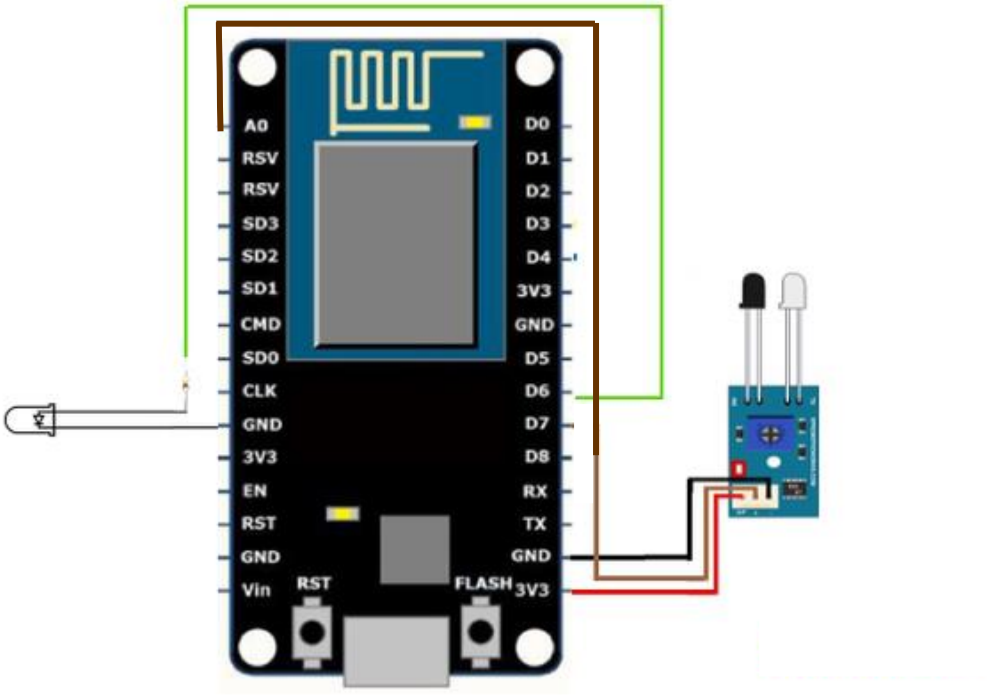


TO TURN ON A BUZZER/LED WHEN AN OBJECT IS DETECTED USING IR SENSOR IN ANALOG MODE.

REQUIREMENTS

- NodeMCU x 1
- Micro USB cable x 1
- PC x 1
- Software Arduino IDE (version 1.6.4+)
- LED
- IR SENSOR

CIRCUIT

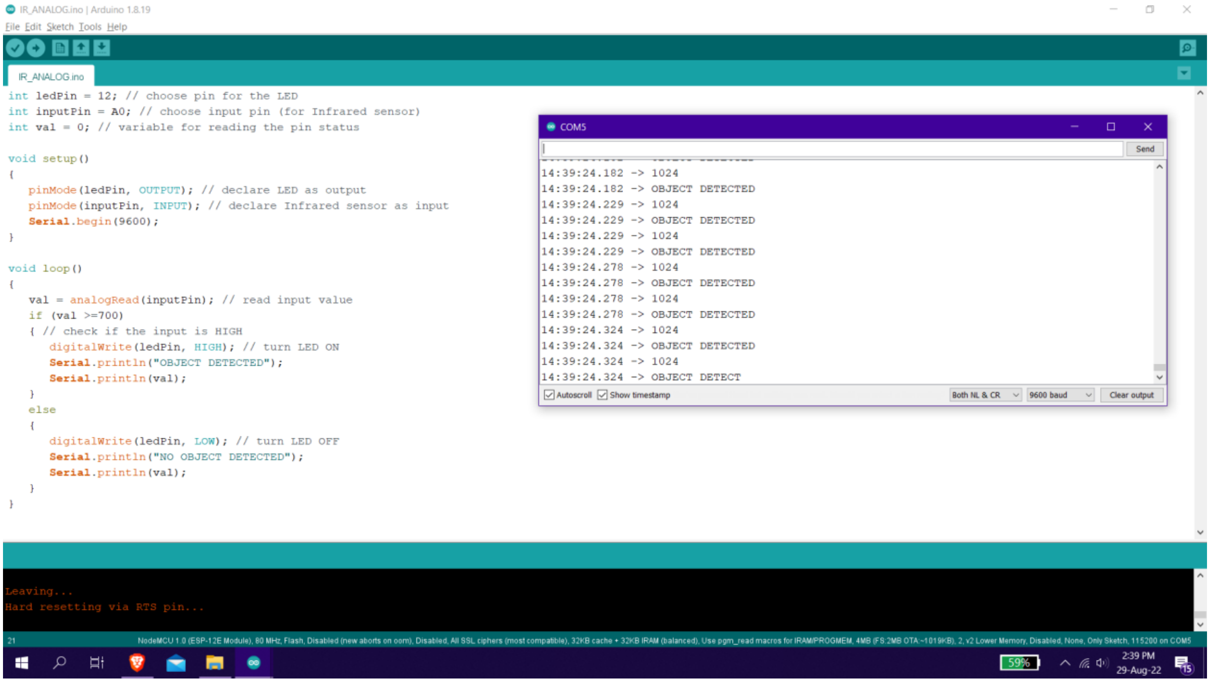


CODE

```
int ledPin = 12; // choose pin for the LED
int inputPin = A0; // choose input pin (for Infrared sensor)
int val = 0; // variable for reading the pin status

void setup()
{
   pinMode(ledPin, OUTPUT); // declare LED as output
   pinMode(inputPin, INPUT); // declare Infrared sensor as input
   Serial.begin(9600);
}

void loop()
{
   val = analogRead(inputPin); // read input value
   if (val >=700)
   { // check if the input is HIGH
      digitalWrite(ledPin, HIGH); // turn LED ON
      Serial.println("OBJECT DETECTED");
      Serial.println(val);
   }
   else
   {
      digitalWrite(ledPin, LOW); // turn LED OFF
      Serial.println("NO OBJECT DETECTED");
      Serial.println(val);
   }
}
```

RESULT

When Object is Detected(<15cm)

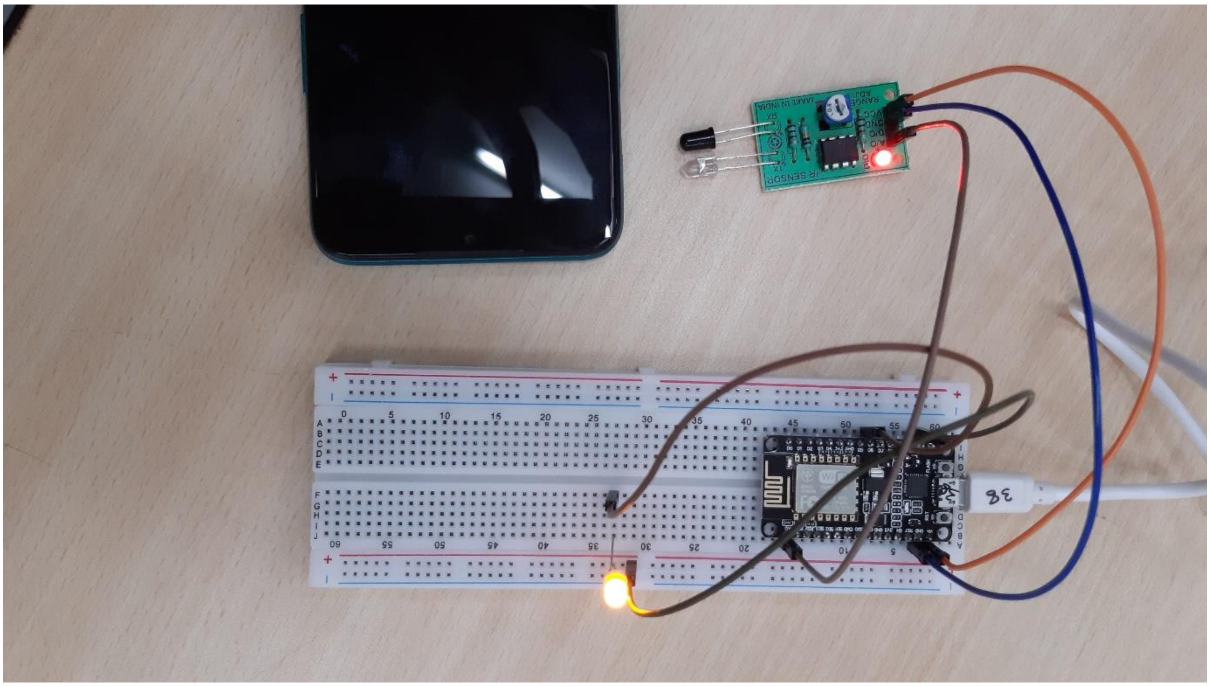

IR_ANALOG.ino | Arduino 1.8.19
File Edit Sketch Tools Help
IR_ANALOG.ino
COM5
Send
☑ Autoscroll ☑ Show timestamp
Both NL & CR
9600 baud
Clear output
59%
2:39 PM
29-Aug-22

```
int ledPin = 12; // choose pin for the LED
int inputPin = A0; // choose input pin (for Infrared sensor)
int val = 0; // variable for reading the pin status

void setup()
{
   pinMode(ledPin, OUTPUT); // declare LED as output
   pinMode(inputPin, INPUT); // declare Infrared sensor as input
   Serial.begin(9600);
}

void loop()
{
   val = analogRead(inputPin); // read input value
   if (val >=700)
   { // check if the input is HIGH
      digitalWrite(ledPin, HIGH); // turn LED ON
      Serial.println("OBJECT DETECTED");
      Serial.println(val);
   }
   else
   {
      digitalWrite(ledPin, LOW); // turn LED OFF
      Serial.println("NO OBJECT DETECTED");
      Serial.println(val);
   }
}
```

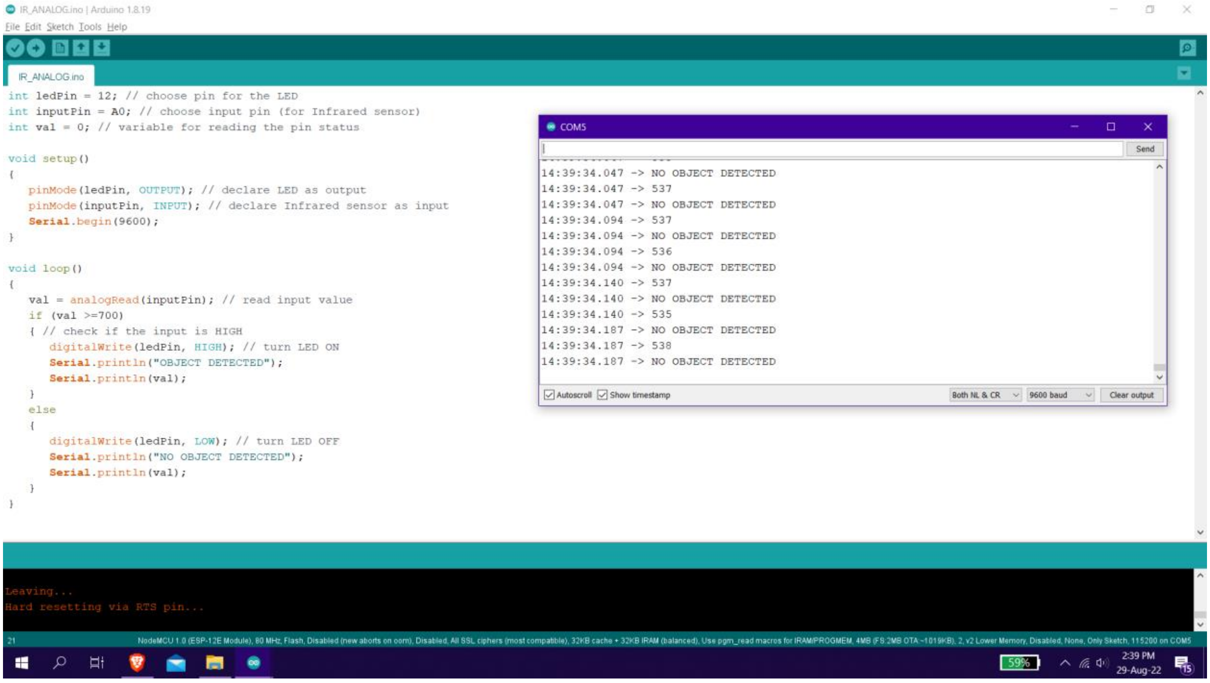
```
14:39:24.182 -> 1024
14:39:24.182 -> OBJECT DETECTED
14:39:24.229 -> 1024
14:39:24.229 -> OBJECT DETECTED
14:39:24.229 -> 1024
14:39:24.229 -> OBJECT DETECTED
14:39:24.278 -> 1024
14:39:24.278 -> OBJECT DETECTED
14:39:24.278 -> 1024
14:39:24.278 -> OBJECT DETECTED
14:39:24.324 -> 1024
14:39:24.324 -> OBJECT DETECTED
14:39:24.324 -> 1024
14:39:24.324 -> OBJECT DETECT
```

```
Leaving...
Hard resetting via RTS pin...
```

21 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5



When Object is Not Detected(>15cm)



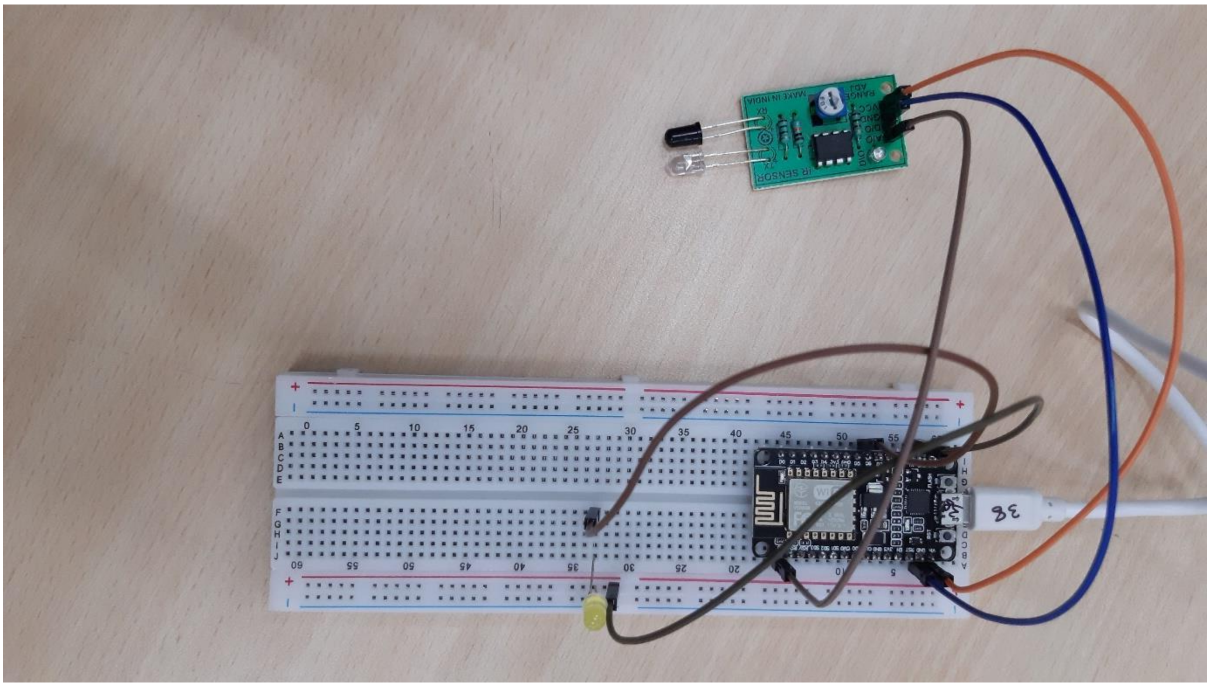
```
int ledPin = 12; // choose pin for the LED
int inputPin = A0; // choose input pin (for Infrared sensor)
int val = 0; // variable for reading the pin status

void setup()
{
   pinMode(ledPin, OUTPUT); // declare LED as output
   pinMode(inputPin, INPUT); // declare Infrared sensor as input
   Serial.begin(9600);
}

void loop()
{
   val = analogRead(inputPin); // read input value
   if (val >=700)
   { // check if the input is HIGH
      digitalWrite(ledPin, HIGH); // turn LED ON
      Serial.println("OBJECT DETECTED");
      Serial.println(val);
   }
   else
   {
      digitalWrite(ledPin, LOW); // turn LED OFF
      Serial.println("NO OBJECT DETECTED");
      Serial.println(val);
   }
}
```

```
14:39:34.047 -> NO OBJECT DETECTED
14:39:34.047 -> 537
14:39:34.047 -> NO OBJECT DETECTED
14:39:34.094 -> 537
14:39:34.094 -> NO OBJECT DETECTED
14:39:34.094 -> 536
14:39:34.094 -> NO OBJECT DETECTED
14:39:34.140 -> 537
14:39:34.140 -> NO OBJECT DETECTED
14:39:34.140 -> 535
14:39:34.187 -> NO OBJECT DETECTED
14:39:34.187 -> 538
14:39:34.187 -> NO OBJECT DETECTED
```

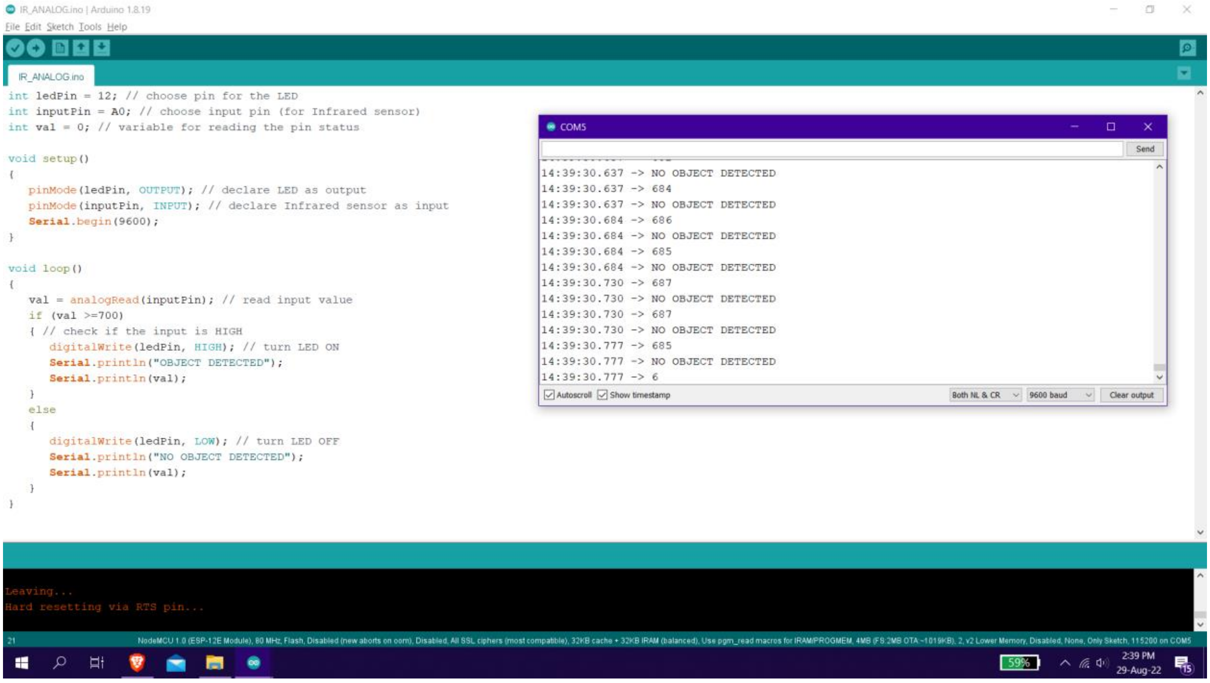
☑ Autoscroll ☑ Show timestamp Both NL & CR 9600 baud Clear output

```
Leaving...
Hard resetting via RTS pin...
```

21 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5



When Object is not close enough



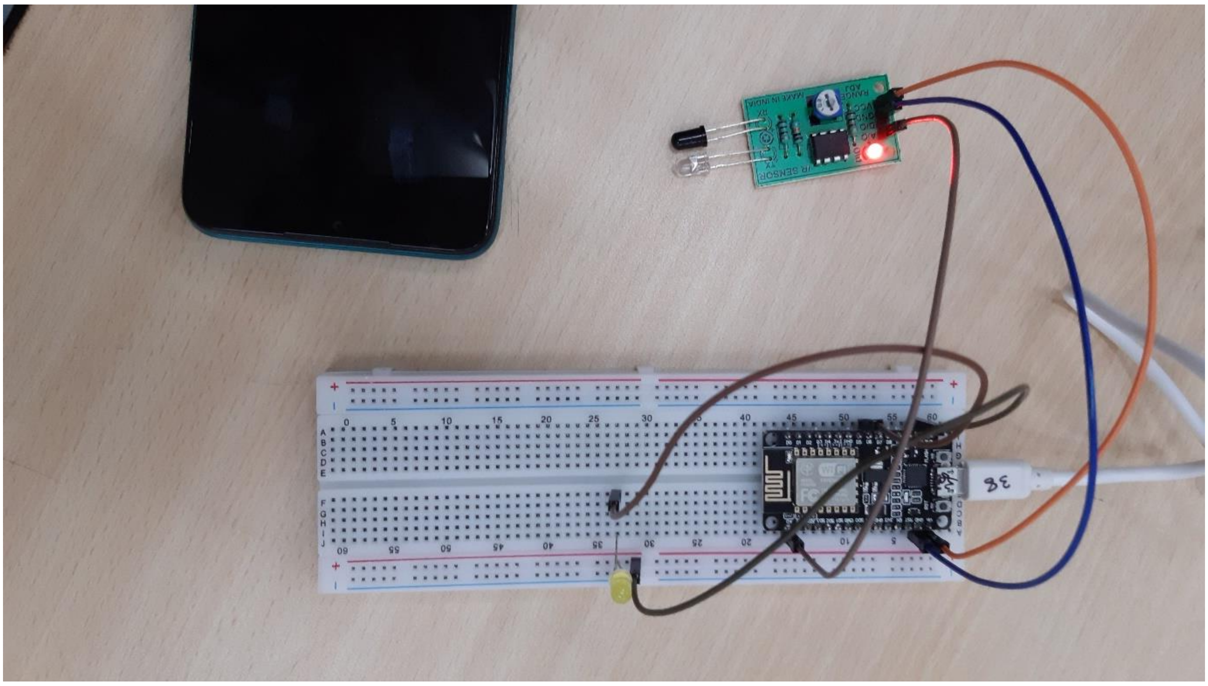
```
int ledPin = 12; // choose pin for the LED
int inputPin = A0; // choose input pin (for Infrared sensor)
int val = 0; // variable for reading the pin status

void setup()
{
   pinMode(ledPin, OUTPUT); // declare LED as output
   pinMode(inputPin, INPUT); // declare Infrared sensor as input
   Serial.begin(9600);
}

void loop()
{
   val = analogRead(inputPin); // read input value
   if (val >=700)
   { // check if the input is HIGH
      digitalWrite(ledPin, HIGH); // turn LED ON
      Serial.println("OBJECT DETECTED");
      Serial.println(val);
   }
   else
   {
      digitalWrite(ledPin, LOW); // turn LED OFF
      Serial.println("NO OBJECT DETECTED");
      Serial.println(val);
   }
}
```

```
14:39:30.637 -> NO OBJECT DETECTED
14:39:30.637 -> 684
14:39:30.637 -> NO OBJECT DETECTED
14:39:30.684 -> 686
14:39:30.684 -> NO OBJECT DETECTED
14:39:30.684 -> 685
14:39:30.684 -> NO OBJECT DETECTED
14:39:30.730 -> 687
14:39:30.730 -> NO OBJECT DETECTED
14:39:30.730 -> 687
14:39:30.730 -> NO OBJECT DETECTED
14:39:30.777 -> 685
14:39:30.777 -> NO OBJECT DETECTED
14:39:30.777 -> 6
```

☑ Autoscroll ☑ Show timestamp Both NL & CR 9600 baud

```
Leaving...
Hard resetting via RTS pin...
```

21 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5

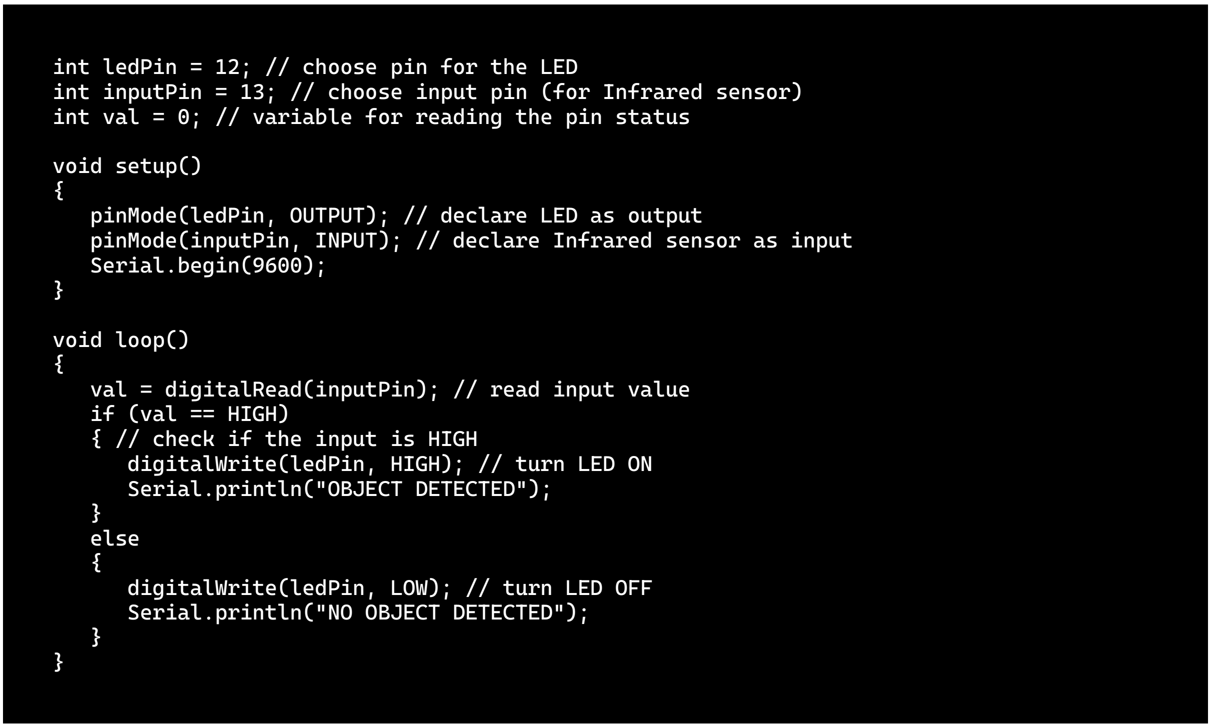

IR SENSOR
RANGE ADJ
MADE IN INDIA
38

TO TURN ON A BUZZER/LED WHEN AN OBJECT IS DETECTED USING IR SENSOR IN DIGITAL MODE.

REQUIREMENTS

- NodeMCU x 1
- Micro USB cable x 1
- PC x 1
- Software Arduino IDE(version 1.6.4+)
- IR SENSOR
- LED

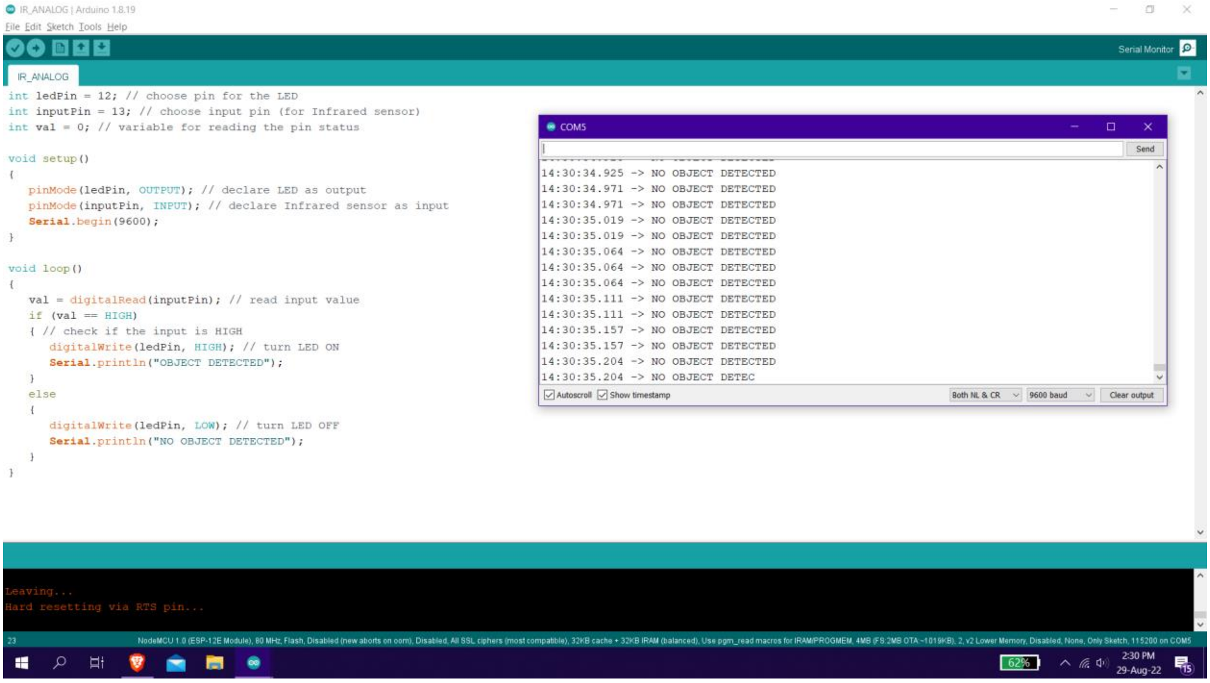
CODE



```
int ledPin = 12; // choose pin for the LED
int inputPin = 13; // choose input pin (for Infrared sensor)
int val = 0; // variable for reading the pin status

void setup()
{
   pinMode(ledPin, OUTPUT); // declare LED as output
   pinMode(inputPin, INPUT); // declare Infrared sensor as input
   Serial.begin(9600);
}

void loop()
{
   val = digitalRead(inputPin); // read input value
   if (val == HIGH)
   { // check if the input is HIGH
      digitalWrite(ledPin, HIGH); // turn LED ON
      Serial.println("OBJECT DETECTED");
   }
   else
   {
      digitalWrite(ledPin, LOW); // turn LED OFF
      Serial.println("NO OBJECT DETECTED");
   }
}
```

RESULT

When No Object is detected



```
int ledPin = 12; // choose pin for the LED
int inputPin = 13; // choose input pin (for Infrared sensor)
int val = 0; // variable for reading the pin status

void setup()
{
   pinMode(ledPin, OUTPUT); // declare LED as output
   pinMode(inputPin, INPUT); // declare Infrared sensor as input
   Serial.begin(9600);
}

void loop()
{
   val = digitalRead(inputPin); // read input value
   if (val == HIGH)
   { // check if the input is HIGH
      digitalWrite(ledPin, HIGH); // turn LED ON
      Serial.println("OBJECT DETECTED");
   }
   else
   {
      digitalWrite(ledPin, LOW); // turn LED OFF
      Serial.println("NO OBJECT DETECTED");
   }
}
```

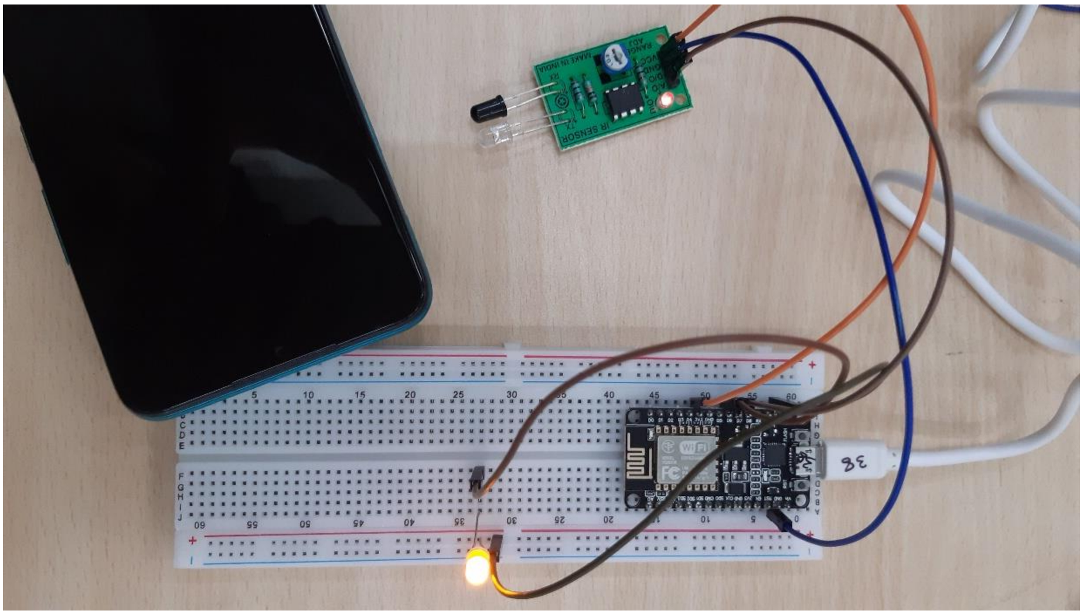
```
14:30:34.925 -> NO OBJECT DETECTED
14:30:34.971 -> NO OBJECT DETECTED
14:30:34.971 -> NO OBJECT DETECTED
14:30:35.019 -> NO OBJECT DETECTED
14:30:35.019 -> NO OBJECT DETECTED
14:30:35.064 -> NO OBJECT DETECTED
14:30:35.064 -> NO OBJECT DETECTED
14:30:35.064 -> NO OBJECT DETECTED
14:30:35.111 -> NO OBJECT DETECTED
14:30:35.111 -> NO OBJECT DETECTED
14:30:35.157 -> NO OBJECT DETECTED
14:30:35.157 -> NO OBJECT DETECTED
14:30:35.204 -> NO OBJECT DETECTED
14:30:35.204 -> NO OBJECT DETEC
```

☑ Autoscroll ☑ Show timestamp Both NL & CR 9600 baud

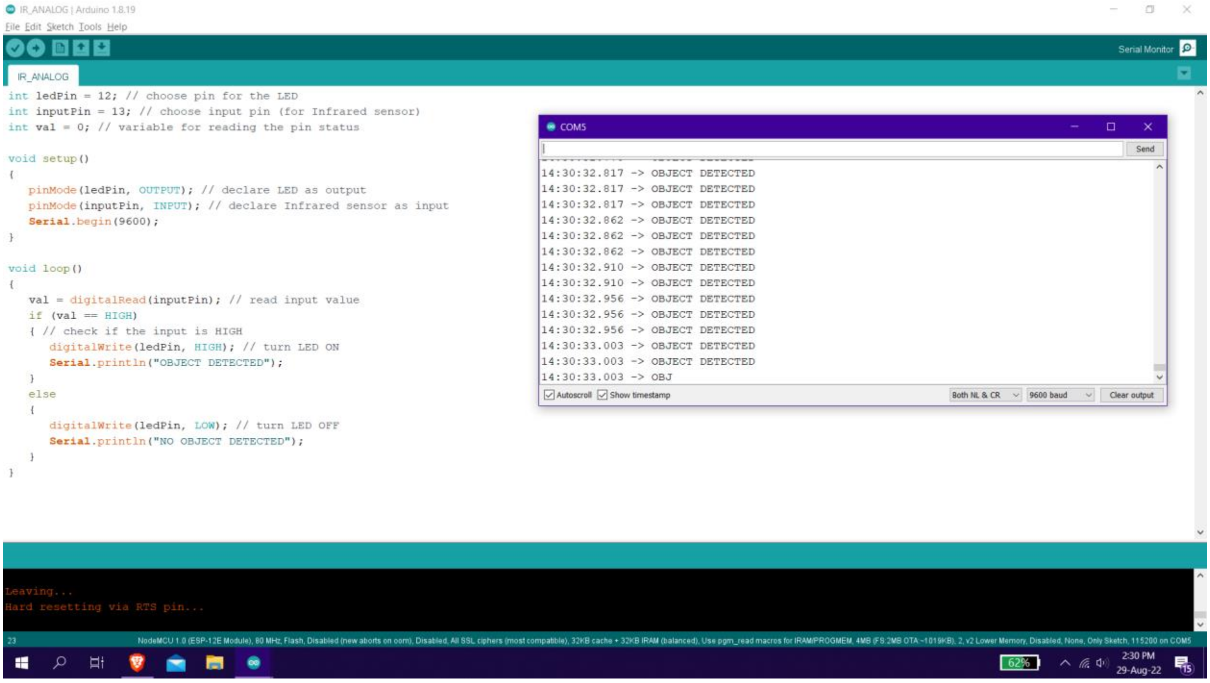
```
Leaving...
Hard resetting via RTS pin...
```

23 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5

62% 2:30 PM 29-Aug-22



When Object is Not detected



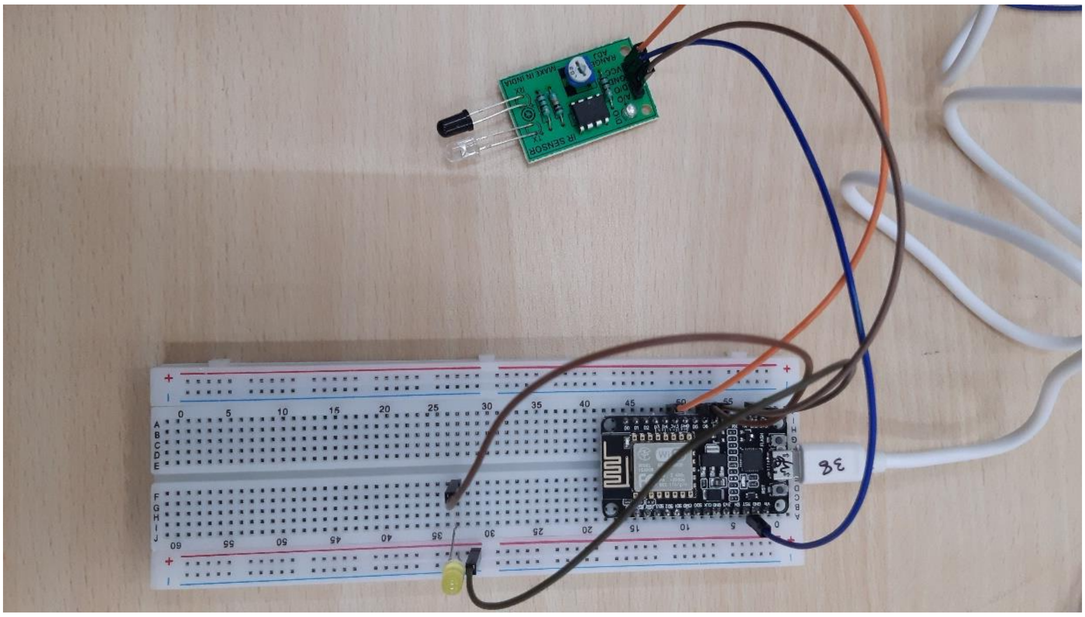
```
int ledPin = 12; // choose pin for the LED
int inputPin = 13; // choose input pin (for Infrared sensor)
int val = 0; // variable for reading the pin status

void setup()
{
   pinMode(ledPin, OUTPUT); // declare LED as output
   pinMode(inputPin, INPUT); // declare Infrared sensor as input
   Serial.begin(9600);
}

void loop()
{
   val = digitalRead(inputPin); // read input value
   if (val == HIGH)
   { // check if the input is HIGH
      digitalWrite(ledPin, HIGH); // turn LED ON
      Serial.println("OBJECT DETECTED");
   }
   else
   {
      digitalWrite(ledPin, LOW); // turn LED OFF
      Serial.println("NO OBJECT DETECTED");
   }
}
```

```
14:30:32.817 -> OBJECT DETECTED
14:30:32.817 -> OBJECT DETECTED
14:30:32.817 -> OBJECT DETECTED
14:30:32.862 -> OBJECT DETECTED
14:30:32.862 -> OBJECT DETECTED
14:30:32.862 -> OBJECT DETECTED
14:30:32.910 -> OBJECT DETECTED
14:30:32.910 -> OBJECT DETECTED
14:30:32.956 -> OBJECT DETECTED
14:30:32.956 -> OBJECT DETECTED
14:30:32.956 -> OBJECT DETECTED
14:30:33.003 -> OBJECT DETECTED
14:30:33.003 -> OBJECT DETECTED
14:30:33.003 -> OBJ
```

☑ Autoscroll ☑ Show timestamp Both NL & CR 9600 baud Clear output

```
Leaving...
Hard resetting via RTS pin...
```

23 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5

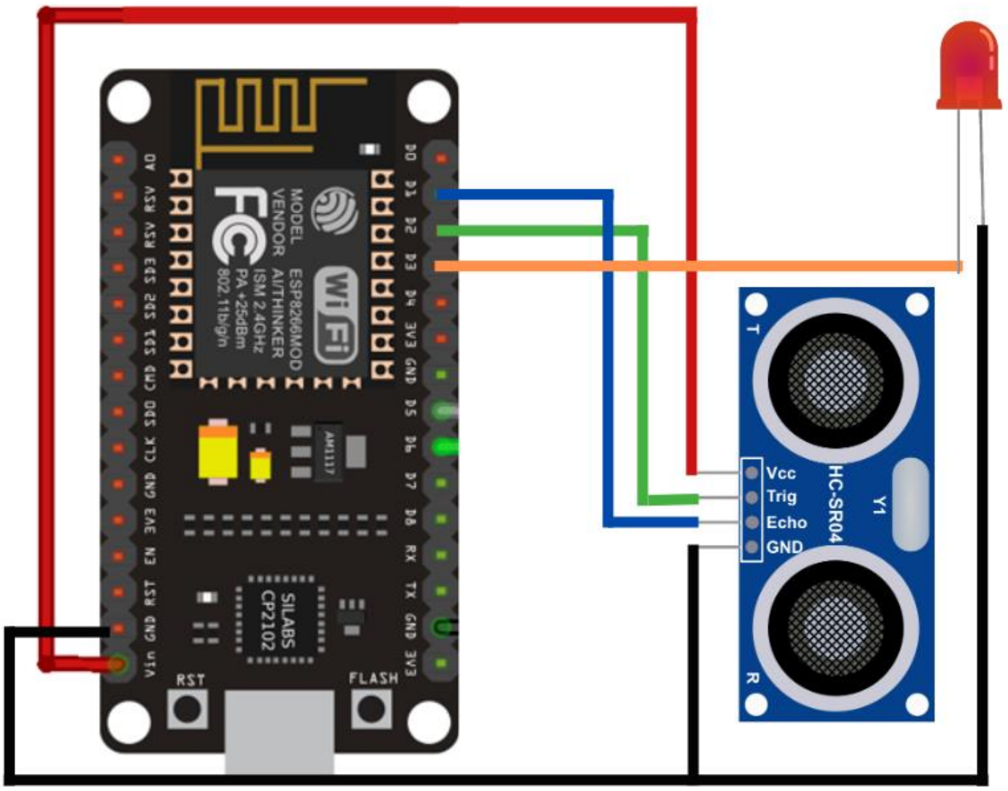


CALCULATE THE DISTANCE OF AN OBJECT USING ULTRASONIC SENSOR AND CONTROL A LED AND SHOW THE VALUE ON SERIAL MONITOR

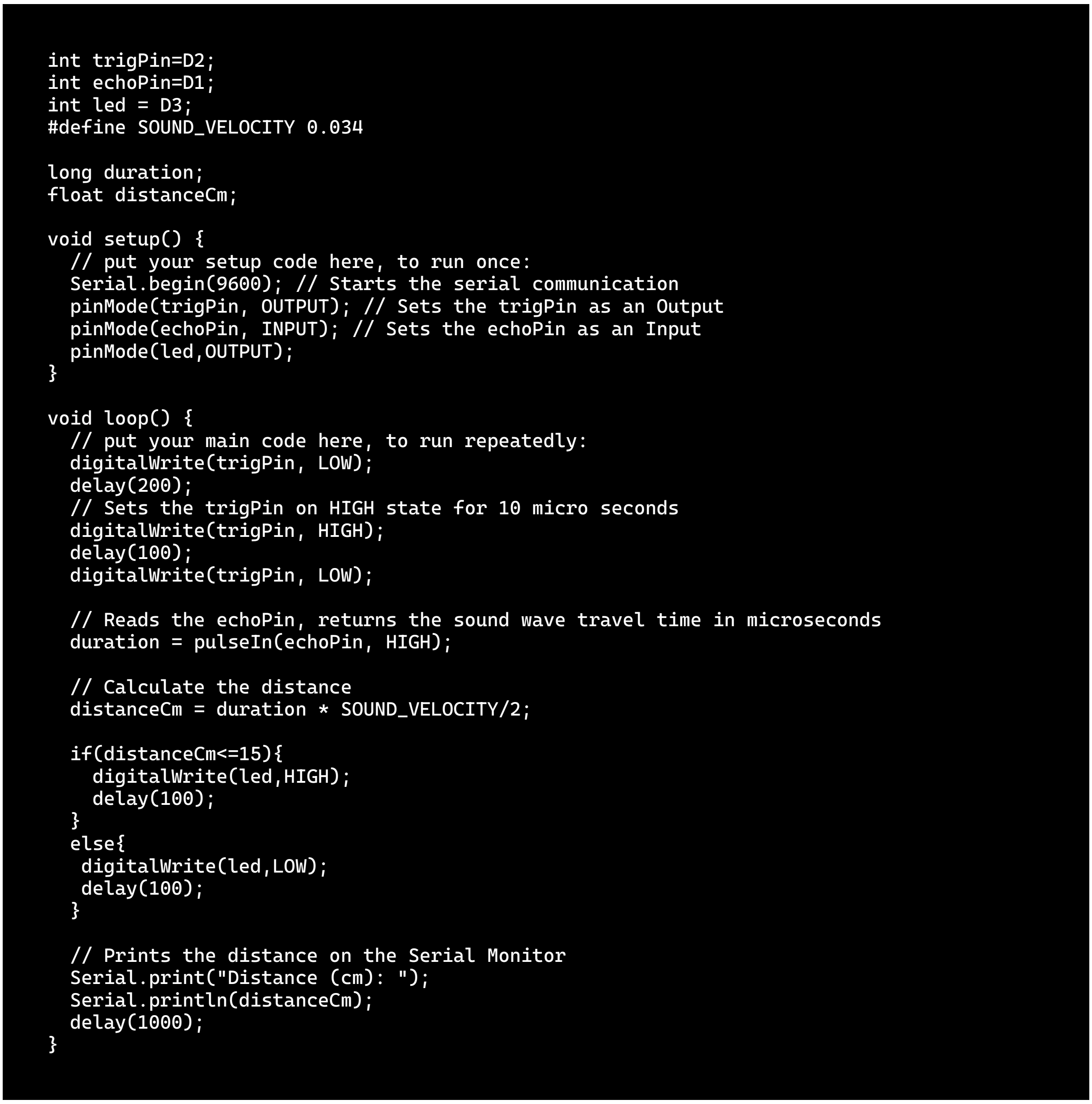
REQUIREMENTS

- NodeMCU x 1
- Micro USB cable x 1
- PC x 1
- Software Arduino IDE (version 1.6.4+)
- Ultrasonic Sensor
- LED

CIRCUIT

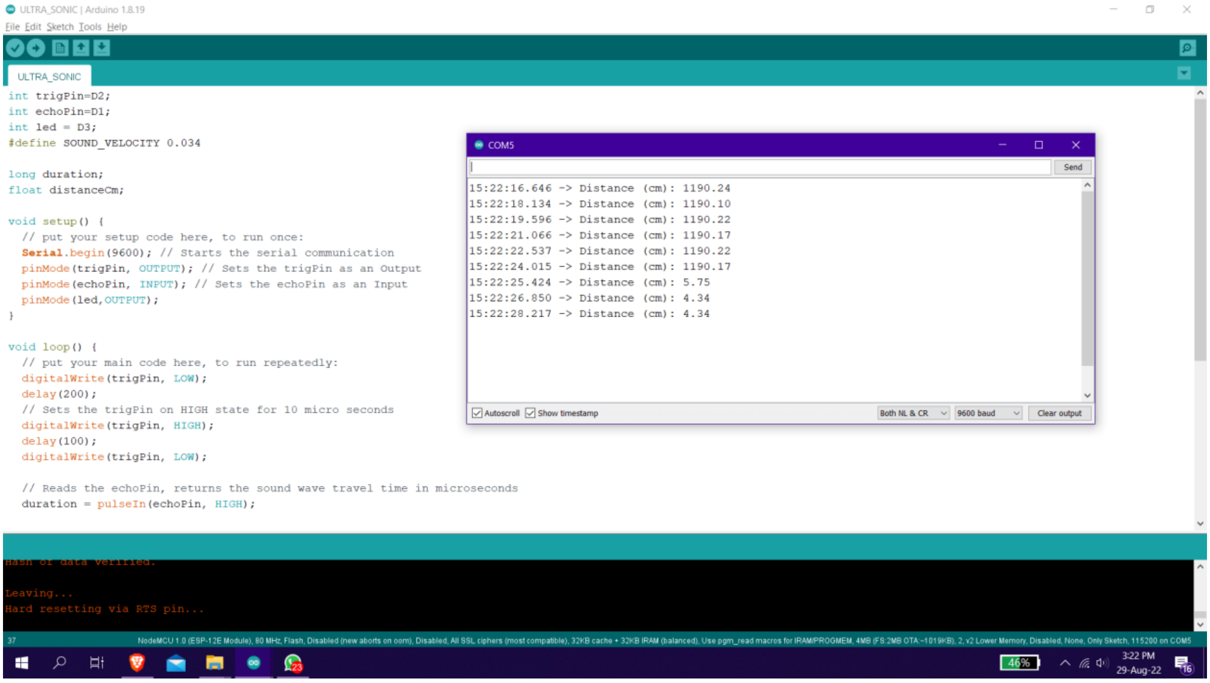


CODE



```
int trigPin=D2;
int echoPin=D1;
int led = D3;
#define SOUND_VELOCITY 0.034

long duration;
float distanceCm;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Starts the serial communication
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(led,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(trigPin, LOW);
  delay(200);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delay(100);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculate the distance
  distanceCm = duration * SOUND_VELOCITY/2;

  if(distanceCm<=15){
    digitalWrite(led,HIGH);
    delay(100);
  }
  else{
   digitalWrite(led,LOW);
   delay(100);
  }

  // Prints the distance on the Serial Monitor
  Serial.print("Distance (cm): ");
  Serial.println(distanceCm);
  delay(1000);
}
```

RESULT

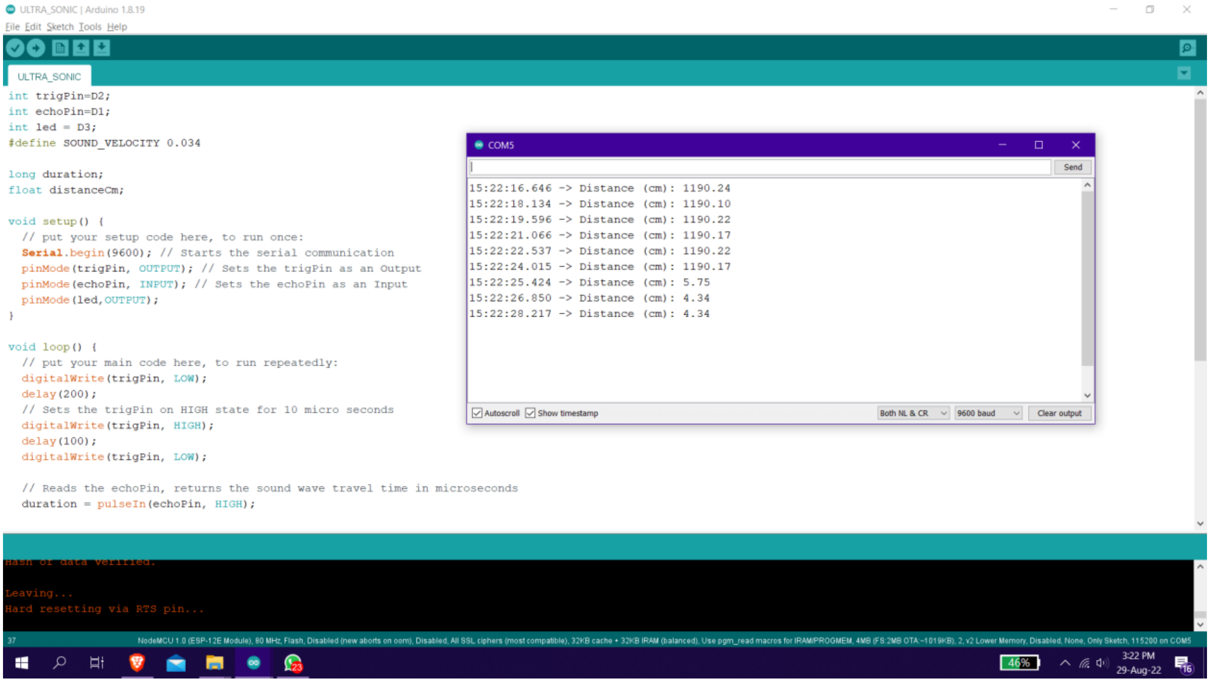

ULTRA_SONIC | Arduino 1.8.19
File Edit Sketch Tools Help
ULTRA_SONIC
COM5
Send
☑ Autoscroll ☑ Show timestamp
Both NL & CR
9600 baud
Clear output
3:22 PM
29-Aug-22
46%

```
int trigPin=D2;
int echoPin=D1;
int led = D3;
#define SOUND_VELOCITY 0.034

long duration;
float distanceCm;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Starts the serial communication
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(led,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(trigPin, LOW);
  delay(200);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delay(100);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
```

```
15:22:16.646 -> Distance (cm): 1190.24
15:22:18.134 -> Distance (cm): 1190.10
15:22:19.596 -> Distance (cm): 1190.22
15:22:21.066 -> Distance (cm): 1190.17
15:22:22.537 -> Distance (cm): 1190.22
15:22:24.015 -> Distance (cm): 1190.17
15:22:25.424 -> Distance (cm): 5.75
15:22:26.850 -> Distance (cm): 4.34
15:22:28.217 -> Distance (cm): 4.34
```

```
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

37 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5

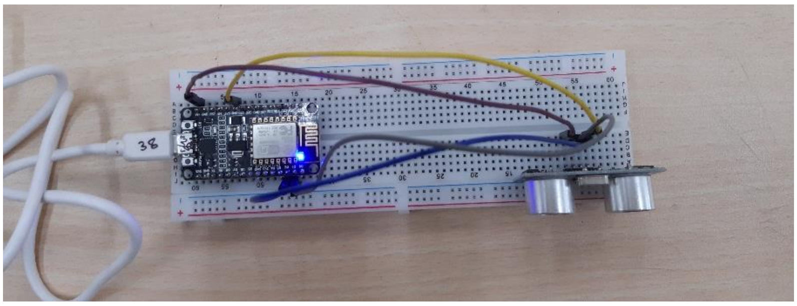

ULTRA_SONIC | Arduino 1.8.19
File Edit Sketch Tools Help
ULTRA_SONIC
COM5
Send
☑ Autoscroll ☑ Show timestamp
Both NL & CR
9600 baud
Clear output
3:22 PM
29-Aug-22
46%

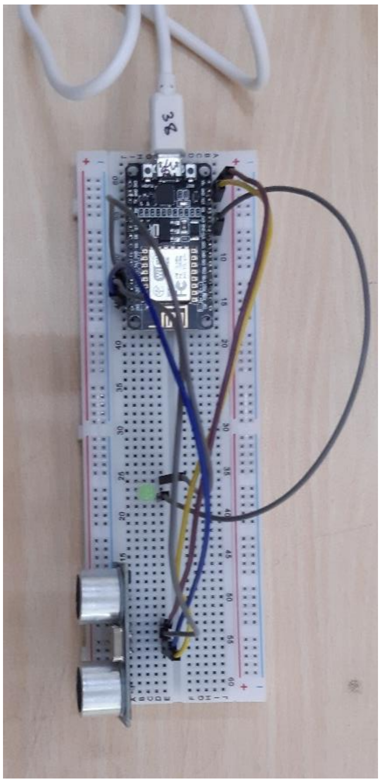
```
int trigPin=D2;
int echoPin=D1;
int led = D3;
#define SOUND_VELOCITY 0.034

long duration;
float distanceCm;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Starts the serial communication
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(led,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(trigPin, LOW);
  delay(200);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delay(100);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
```

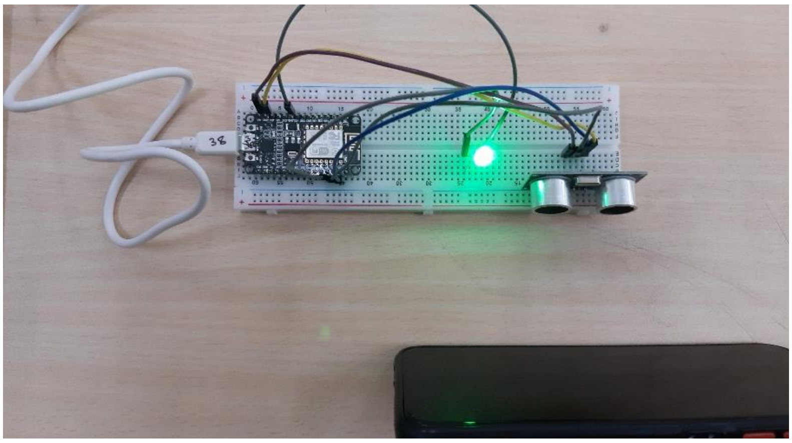
```
15:22:16.646 -> Distance (cm): 1190.24
15:22:18.134 -> Distance (cm): 1190.10
15:22:19.596 -> Distance (cm): 1190.22
15:22:21.066 -> Distance (cm): 1190.17
15:22:22.537 -> Distance (cm): 1190.22
15:22:24.015 -> Distance (cm): 1190.17
15:22:25.424 -> Distance (cm): 5.75
15:22:26.850 -> Distance (cm): 4.34
15:22:28.217 -> Distance (cm): 4.34
```

```
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

37 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM5


38


38


38