PERIYAR MANIAMMAI INSTITUTE OF SCIENCE & TECHNOLOGY

PERIYAR MANIAMMAI
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University)
Established Under Sec. 3 of UGC Act, 1956 • NAAC Accredited
think • innovate • transform

RECORD NOTE BOOK

COURSE NAME	:
COURSE CODE	:
STUDENT NAME	:
REGISTER NUMBER	:
BRANCH	:
YEAR	:
SEMESTER	:

PERIYAR MANIAMMAI INSTITUTE OF SCIENCE & TECHNOLOGY

PERIYAR MANIAMMAI
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University)
Established Under Sec. 3 of UGC Act, 1956 • NAAC Accredited
think • innovate • transform

CERTIFICATE

This is to certify that Mr. _____

a student of _____

bearing the register number _____ has been completed the recordnotebook

for the course _____

as per the curriculum during the academic year 2024 – 2025. The work submitted In this notebook has been verified and found to be satisfactory.

Year: III Semester : VI

Laboratory/Course in-Charge
(With date)
Name:

Head of the Department
(with date and seal)

Sumbitted for the Practical Examination held on _____

Internal Examiner
(with Date)
Name:

External Examiner
(with Date)
Name:

INDEX

Exp : 01 LED Blinking by Interface with Arduino

Aim:

To interface LED with Arduino UNO

Software Requirement

Arduino IDE

Hardware Requirement

- Arduino UNO

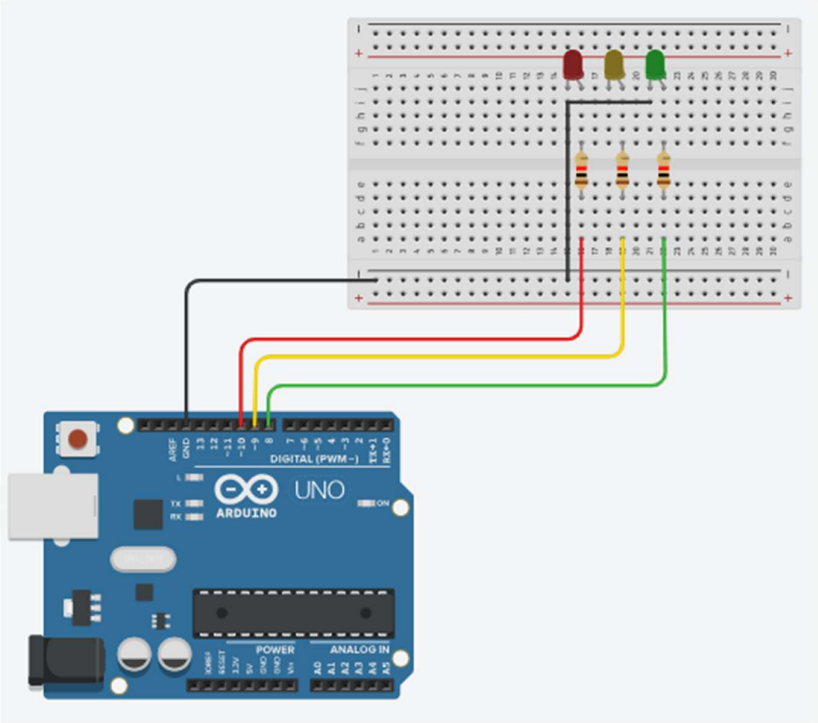
Source Code

```
#define MOTOR_IN1 8
#define MOTOR_IN2 9
#define MOTOR_IN3 10
void setup() {
   pinMode(MOTOR_IN1, OUTPUT);
   pinMode(MOTOR_IN2, OUTPUT);
   pinMode(MOTOR_IN3, OUTPUT);
}
void loop() {
 // Turn on LED 1
 digitalWrite(MOTOR_IN1, HIGH);
 delay(500);
 digitalWrite(MOTOR_IN1, LOW);

 // Turn on LED 2
 digitalWrite(MOTOR_IN2, HIGH);
 delay(500);
 digitalWrite(MOTOR_IN2, LOW);

 // Turn on LED 3
 digitalWrite(MOTOR_IN3, HIGH);
 delay(500);
 digitalWrite(MOTOR_IN3, LOW);
}
```

Circuit Diagram:



Result:

Thus,the interface LED with arduino UNO has been successfully and verified.

Exp : 02 Interface and Control DC Motor with Arduino

Aim:

Interface & control DC motor with Arduino uno.

Software Requirement:

- **Arduino UNO**

Hardware Requirement

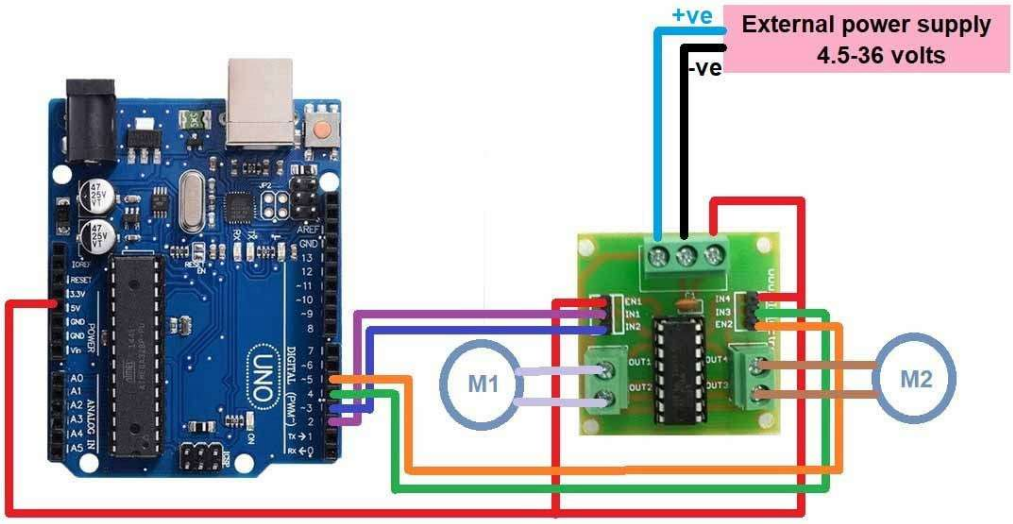
- **Arduino IDE**

Source Code:

```
#define MOTOR_IN1 8
#define MOTOR_IN2 9
void setup() {
  pinMode(MOTOR_IN1, OUTPUT);
  pinMode(MOTOR_IN2, OUTPUT);
}
void loop() {
  // Rotate Right (Forward)
  digitalWrite(MOTOR_IN1, HIGH);
  digitalWrite(MOTOR_IN2, LOW);
  delay(3000); // rotate for 3 seconds
  // Stop
  digitalWrite(MOTOR_IN1, LOW);
  digitalWrite(MOTOR_IN2, LOW);
  delay(1000); // pause 1 second

  // Rotate Left (Reverse)
  digitalWrite(MOTOR_IN1, LOW);
  digitalWrite(MOTOR_IN2, HIGH);
  delay(3000); // rotate for 3 seconds
  // Stop
  digitalWrite(MOTOR_IN1, LOW);
  digitalWrite(MOTOR_IN2, LOW);
  delay(1000); // pause 1 second
}
```

Circuit Diagram:



Result:

Thus the Interface and Control DC Motor with Arduino has been successfully verified.

Exp : 03 Interface Ultrasonic Sensor with Arduino and Write a Code to Calculate Object Distance

Aim:

To interface ultrasonic sensor with Arduino uno &write a code to calculate object distance.

Software Requirement:

- Arduino IDE

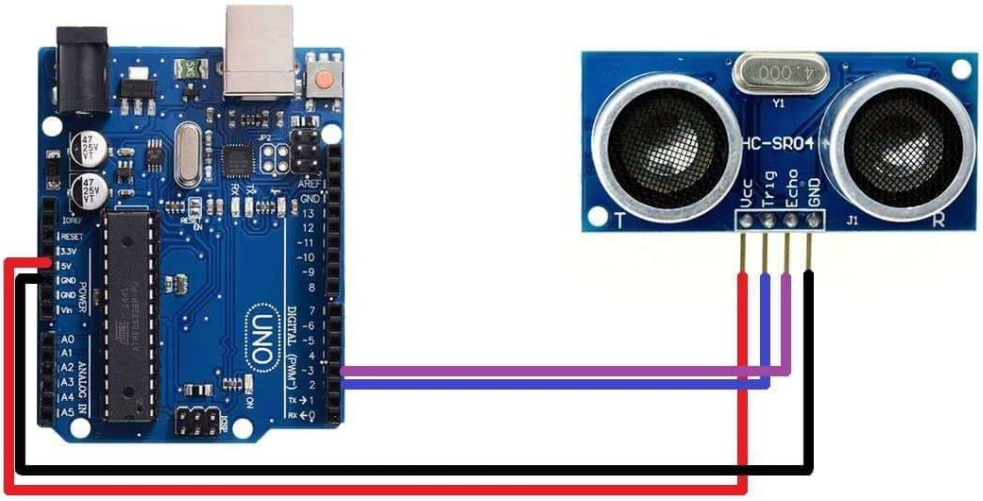
Hardware Requirement:

- Ardunio UNO

Source Code:

```
#define TRIG_PIN 6
#define ECHO_PIN 7
void setup() {
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  Serial.begin(9600);
}
void loop() {
  long distance = getDistance();
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500); // Delay for readability
}
long getDistance() {
 // Trigger the sensor
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
 // Read echo duration
  long duration = pulseIn(ECHO_PIN, HIGH);
 // Calculate distance in cm
  long distance = duration * 0.034 / 2;
  return distance;
}
```

Circuit Diagram:



Result:

Thus,the Interface Ultrasonic Sensor with Arduino and Write a Code to Calculate the Object Distance was successfully verified

Exp : 04 Raspberry Pi Setup and Working with Linux Commands

Aim:

To setup a raspberry pi os & working with linux commands

Software Requirement:

- Raspberry Pi Imager.
- SD Card Formater
- Raspbian OS

Hardware Requiement:

- Rashberry pi 4
- SD Card
- Card Reader

Steps For OS Installation:

Step 1: Gather Required Components

- Raspberry Pi board (e.g., Pi 4)
- microSD card (16GB+ recommended)
- Power supply (5V 3A USB-C for Pi 4)
- Monitor, keyboard, and mouse (or use SSH later)
- HDMI cable
- Internet connection (Ethernet or Wi-Fi)

Step 2: Flash the OS (Raspberry Pi OS)

1. Download Raspberry Pi Imager: https://www.raspberrypi.com/software/
2. Insert your microSD card into your computer.
3. Launch the Imager → Choose OS → Select *Raspberry Pi OS (64-bit)*.
4. Choose Storage → Select your microSD card.
5. Click Write. Wait for it to complete.

Step 3: First Boot

1. Insert the microSD card into the Raspberry Pi.
2. Connect monitor, keyboard, and mouse.
3. Power it on — it will boot into the OS.
4. Complete the setup wizard (Wi-Fi, region, update).

Working with Linux Commands on Raspberry Pi

Basic File and Directory Commands

```
ls
cd /path
pwd
mkdir name
rm file
rm -r dir
cp a b
mv a b
```

System & Package Management

```
sudo apt update
sudo apt upgrade
sudo apt install name
sudo reboot
sudo shutdown now
```

Network Commands

```
ifconfig
ping google.com
hostname -I
```

Python and Script Execution

```
python3 script.py
chmod +x file.sh
./file.sh
```

Result:

Thus,the rasbian os is successfully installed&working with basic linux command on Rasbperry pi successfully.

Exp : 05 LED Blinking by Interface with Raspberry Pi

Aim:
To integrate LED blinking with Rasbperry Pi 4.

Software Requirement:

- Thonny IDE

Hardware Requirement:

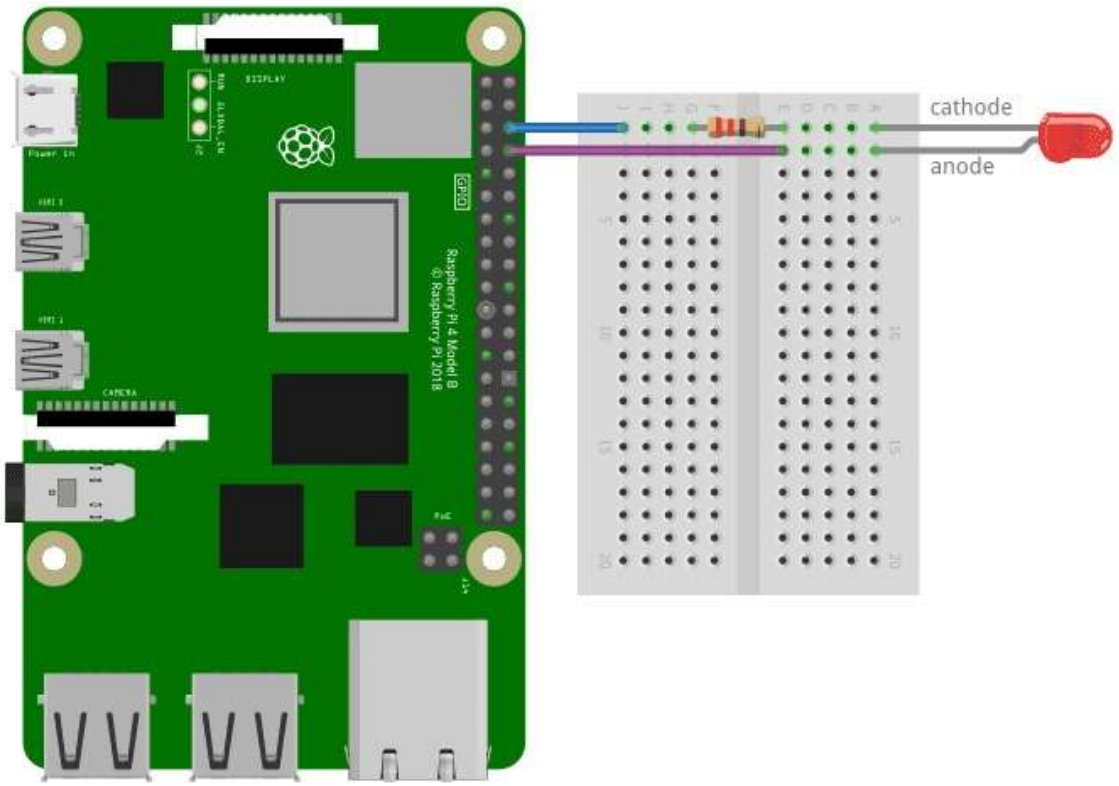
- Rasbperry Pi 4
- LED
- Breadboard

Source Code:

```
import RPi.GPIO as GPIO
import time
# Setup GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(3, GPIO.OUT)
while True:
    print("LIGHT_ON")
    GPIO.output(3, GPIO.HIGH)
    time.sleep(1)
    print("LIGHT_OFF")
    GPIO.output(3, GPIO.LOW)
    time.sleep(1)

GPIO.cleanup
```

Circuit Diagram:



Result:

Thus,the integrate LED with raspberry pi has been successfully & verified.

Exp : 06 Interface and Control Servo Motor with Raspberry Pi

Aim:

To interface & control servo motor with raspberry pi.

Software Requirements:

- Thonny IDE

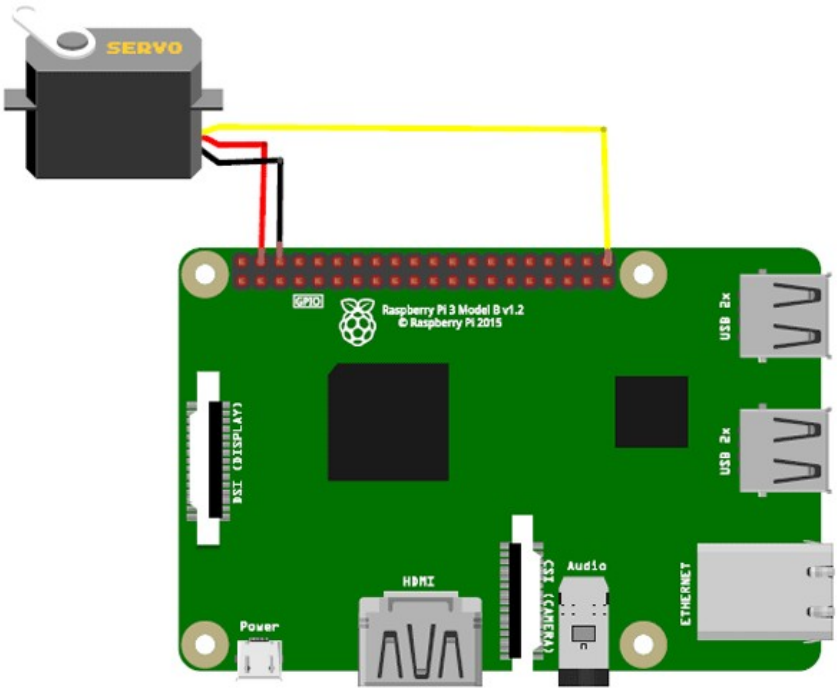
Hardware Requirements:

- Rasbperry Pi 4
- Servomotor
- Breadboard

Source Code:

```
import RPi.GPIO as GPIO
import time
#ervoPIN=11
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(11, GPIO.OUT)
p=GPIO.PWM(11,100)
p.start(0)
while True:
    p.ChangeDutyCycle(0)
    time.sleep(0.5)
    p.ChangeDutyCycle(10)
    time.sleep(0.5)
    p.ChangeDutyCycle(15)
    time.sleep(0.5)
    p.ChangeDutyCycle(20)
    time.sleep(0.5)
    p.ChangeDutyCycle(25)
    time.sleep(0.5)
    p.ChangeDutyCycle(30)
    time.sleep(0.5)
    p.ChangeDutyCycle(5)
    time.sleep(0.5)
    p.ChangeDutyCycle(0)
    time.sleep(0.5)
p.stop()
GPIO.cleanup()
```

Circuit Diagram:



Result:

Thus,the integrate & control servo motor in raspberry pi 4 has been successfully and verified.

Exp : 07 Interface Ultrasonic Sensor with Arduino and Write a Code to Calculate the Object

Aim:

To interface ultrasonic sensor with Arduino & calculate the object using raspberry pi

Software Requirement:

- Thonny IDE

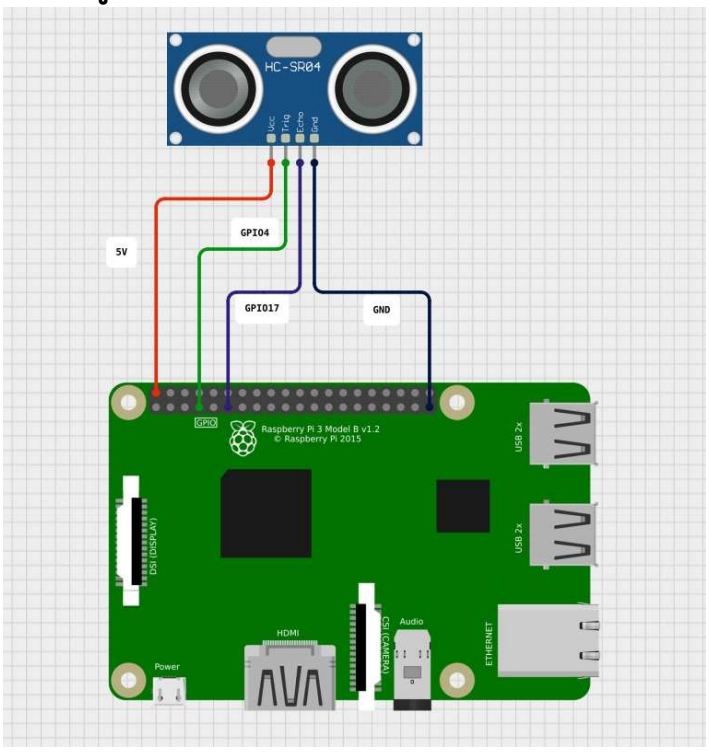
Hardware Requirement:

- Rasbperry Pi 4
- Ultrasonic Sensor
- Breadboard

Source Code:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
TRIG = 31
ECHO = 18
i=0
GPIO.setup(TRIG,GPIO.OUT)#this for output from sensor
GPIO.setup(ECHO,GPIO.IN)#this for inout to sensor
GPIO.setup(TRIG,LOW)
print("calibring:")
time.sleep(2)
print("Place Object")
try:
    while True:
        GPIO.output(TRIG,HIGH)
        time.sleep(0.00001)
        GPIO.output(TRIG,LOW)
        while GPIO.input(ECHO)==0:
            pulse_start=time.time()
        while GPIO.input(ECHO)==1:
            pulese_end=time.time()
        pulse_duration=pulse_end-pulse_start
        distance=pulse_duration*17150
        distance=round(distance+1.15,2)
        if distance <=30 and distance >=5:
            print("distance:",distance,"cm")
            print("object is near")
            i=1
        if distance >30 and i==1:
            print("place the object....")
            i=0
            time.sleep(2)
except KeyboardInterrupt:
        print("Measurement stopped by user")
finally:
    GPIO.cleanup()
```

Circuit Diagram:



Output:

```
calibring:
Place Object
distance: 15.8 cm
object is near
distance: 14.9 cm
object is near
distance: 15.1 cm
object is near
place the object....
distance: 27.4 cm
object is near
place the object....
distance: 38.7 cm
place the object....
distance: 11.4 cm
object is near
distance: 8.2 cm
object is near
place the object....
Measurement stopped by user
```

Result:

Thus the integrate ultrasonic sensor with raspberry pi & calulate the distance using sensor scuessfully.

Exp : 08 Basic Working with Database using Using SQLite

Aim:
To create database in raspberry and work with basic in SQLite.

Software Requirement:

- Thonny IDE
- DB Browser

Hardware Requirement:

- **Raspberry Pi 4**

Source Code:

```
import sqlite3
import datetime
conn=sqlite3.connect("Temperature.db")
cursor=conn.cursor()
cursor.execute("'CREATE TABLE TempData(Temperature FLOAT,Time FLOAT,Status VARCHAR(10))"')

# cursor.execute("DROP TABLE IF EXISTS TempData")

currecntdatetime=datetime.datetime.now()

cursor.execute("INSERT INTO TempData(Temperature,Time,Status) VALUES(96.2,1.26,'Normal')")
cursor.execute("INSERT INTO TempData(Temperature,Time,Status) VALUES(95.4,1.30,'Normal')")
cursor.execute("INSERT INTO TempData(Temperature,Time,Status) VALUES(100.2,1.34,'Abnormal')")
cursor.execute("INSERT INTO TempData(Temperature,Time,Status) VALUES(101.2,1.40,'Abnormal')")
cursor.execute("INSERT INTO TempData(Temperature,Time,Status) VALUES(92.2,1.16,'Normal')")

var = cursor.execute("SELECT Temperature FROM TempData")
conn.commit()
conn.close()
```

Output:

	Temperature	Time	Status
	Filter	Filter	Filter
1	96.2	1.26	Normal
2	95.4	1.3	Normal
3	100.2	1.34	Abnormal
4	101.2	1.4	Abnormal
5	92.2	1.16	Normal

Result:

Thus the working with SQLite in raspberry pi and the database created successfully.

Exp : 09 Collecting Data from Sensor and Store it in Database using SQLite & Rasbpi

Aim:

To collect data from sensor & store it in databse using SQLite &raspberry pi

Software Requirement:

- Thonny IDE
- DB Browser

Hardware Requirement:

- Rasbperry Pi

Source Code:

```
import RPi.GPIO as GPIO
import time
import sqlite3
from datetime import datetime
GPIO.setmode(GPIO.BOARD)
TRIG = 31
ECHO = 18
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
GPIO.output(TRIG, GPIO.LOW)
print("Calibrating...")
time.sleep(2)
print("Place Object")
def log_distance(distance):
    conn = sqlite3.connect('sensordata.db')
    c = conn.cursor()
    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    c.execute("INSERT INTO distances (timestamp, distance) VALUES (?, ?)", (timestamp, distance))
    conn.commit()
    conn.close()
try:
    while True:
        GPIO.output(TRIG, GPIO.HIGH)
        time.sleep(0.00001)
        GPIO.output(TRIG, GPIO.LOW)
        while GPIO.input(ECHO) == 0:
            pulse_start = time.time()
        while GPIO.input(ECHO) == 1:
            pulse_end = time.time()
        pulse_duration = pulse_end - pulse_start
        distance = round(pulse_duration * 17150 + 1.15, 2)
```

```
        if 5 <= distance <= 100:  # Adjust max distance as needed
            print("Distance:", distance, "cm")
            log_distance(distance)
        time.sleep(1)  # Adjust sampling rate
except KeyboardInterrupt:
    print("Measurement stopped by user")
finally:
    GPIO.cleanup()
```

Output:

timestamp	distance
Filter	Filter
2025-05-03 10:00:01	18.46
2025-05-03 10:00:02	22.73
2025-05-03 10:00:03	15.39
2025-05-03 10:00:04	98.25
2025-05-03 10:00:05	12.81
2025-05-03 10:00:06	19.63
2025-05-03 10:00:07	24.1
2025-05-03 10:00:08	31.95
2025-05-03 10:00:09	17.42
2025-05-03 10:00:10	45.7
2025-05-03 10:00:11	5.2
2025-05-03 10:00:12	38.61
2025-05-03 10:00:13	11.48
2025-05-03 10:00:14	49.99
2025-05-03 10:00:15	29.0
2025-05-03 10:00:16	9.86
2025-05-03 10:00:17	21.55
2025-05-03 10:00:18	33.87
2025-05-03 10:00:19	14.44
2025-05-03 10:00:20	27.77

Result:

Thus the collecting data from sensor & store it in databse using SQLite &raspberry pi successfully.

Exp : 10 Node Red Installation and Flow Creation in Node Red

Aim:
To install node red installation & flow creation in node red

Software Requirement:

- **Node Red**

Hardware Requirement:

- **Rasbperry Pi**

Steps:

Step 1: Install Node-RED on Raspberry Pi

bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)

Step 2: Enable and Start Node-RED

sudo systemctl enable nodered.service
node-red-start

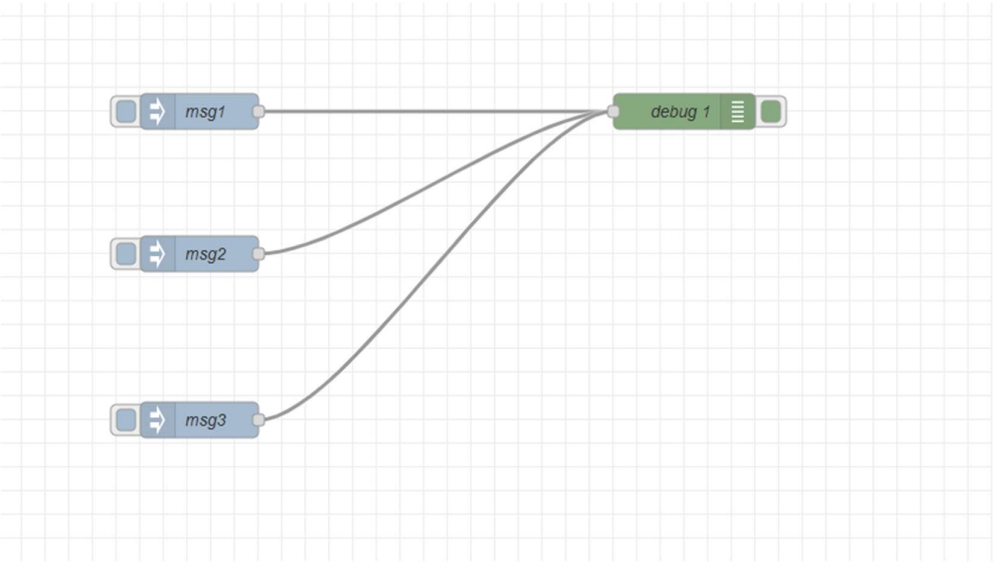
Step 3: Check the status of Node-Red

node-red-stop
node-red-start

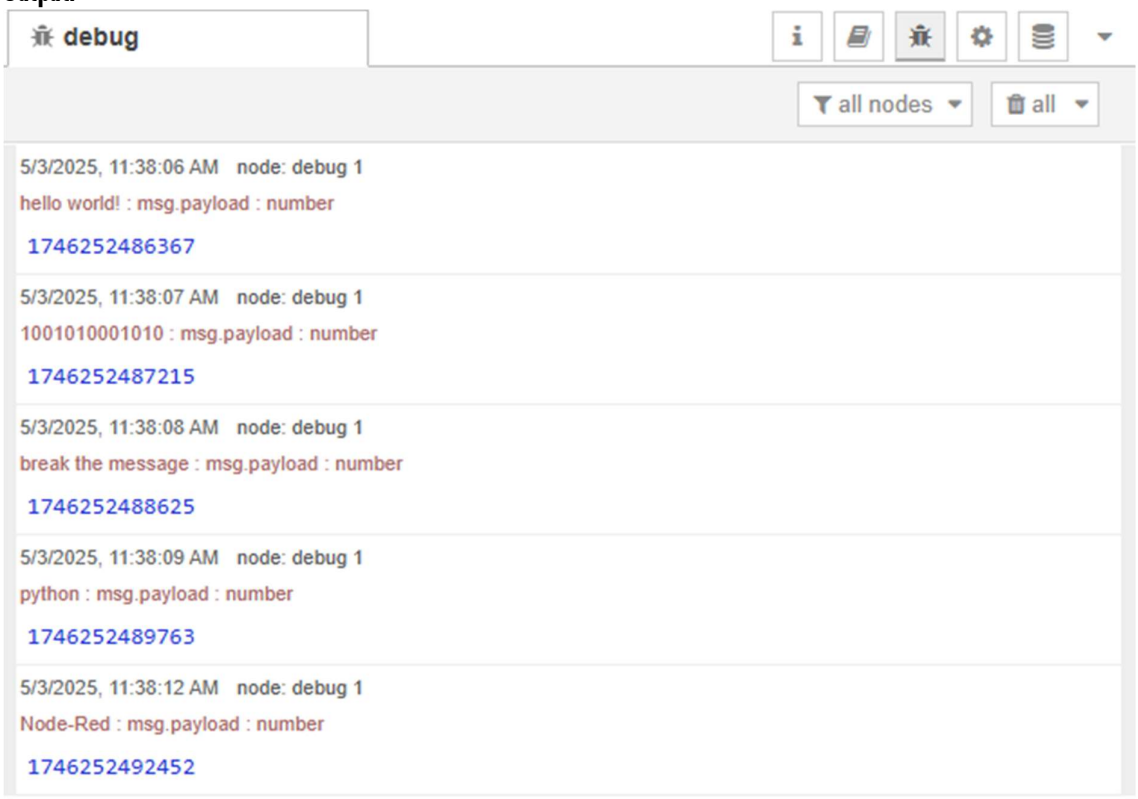
Step 4: Access Node-RED

http://127.0.0.1:1880/

Flow Diagram:



Output:



Result:

Thus,the Node Red Installation and Flow Creation in Node Red has been successfully.

Exp : 11 LED Control Using NodeRed via Rasbpi

Aim:
To control LED using NodeRed in Rasbperry Pi

Software Requiremnt:

- Node-Red

Hardware Requirement:

- Rasbperry Pi
- LED
- Breadboard

Steps:

Step 1: Open Node-RED:

http://<raspberry_pi_ip>:1880

Step 2: Drag these nodes:

- Inject Node (set payload to "on")
- Inject Node (set payload to "off")
- Exec Node

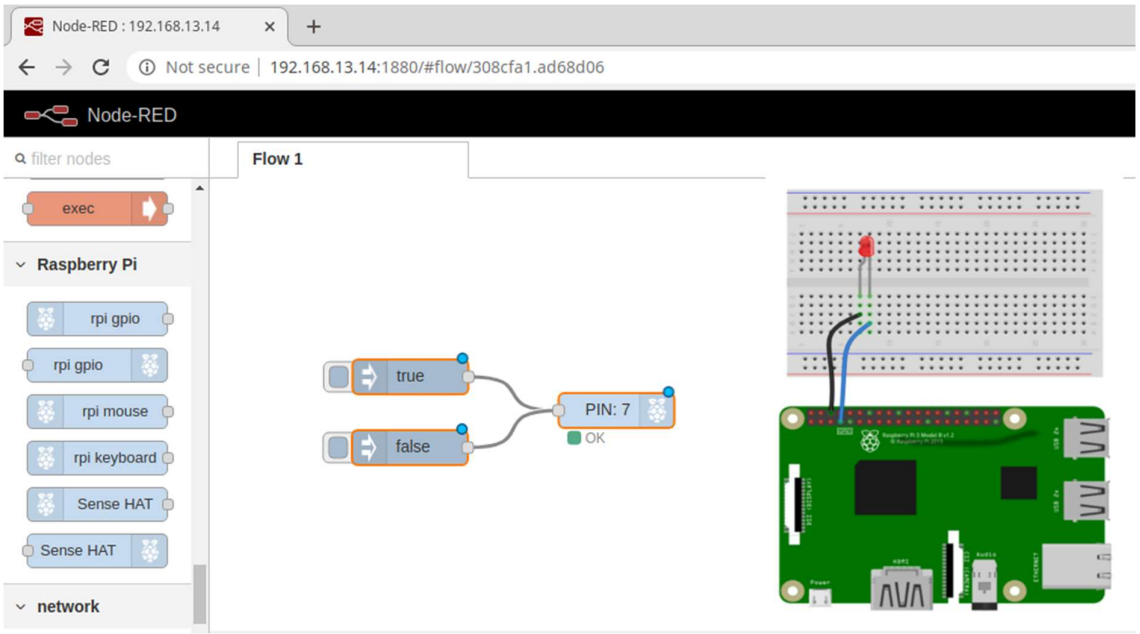
Step 3: Deploy Work Flow

Flow Creation:

Select GPIO.OUT PIN NUMBER

3.3V Power - 1	2 - 5V Power
SDA1 - GPIO02 - 3	4 - 5V Power
SCL1 - GPIO03 - 5	6 - Ground
GPIO04 - 7	8 - GPIO14 - TxD
Ground - 9	10 - GPIO15 - RxD
GPIO17 - 11	12 - GPIO18
GPIO27 - 13	14 - Ground
GPIO22 - 15	16 - GPIO23
3.3V Power - 17	18 - GPIO24
MOSI - GPIO10 - 19	20 - Ground
MISO - GPIO09 - 21	22 - GPIO25
SCLK - GPIO11 - 23	24 - GPIO8 - CE0
Ground - 25	26 - GPIO7 - CE1
SD - 27	28 - SC
GPIO05 - 29	30 - Ground
GPIO06 - 31	32 - GPIO12
GPIO13 - 33	34 - Ground
GPIO19 - 35	36 - GPIO16
GPIO26 - 37	38 - GPIO20
Ground - 39	40 - GPIO21

Flow Diagram:



Result:

Thus the control LED with Node-Red in Rasbperry Pi was successfully.

Exp : 12 Data Acquisition from Sensor and Visualize in Node Red

Aim:

To get data from sensor and visualize in Node red using raspberry pi

Software Requirement:

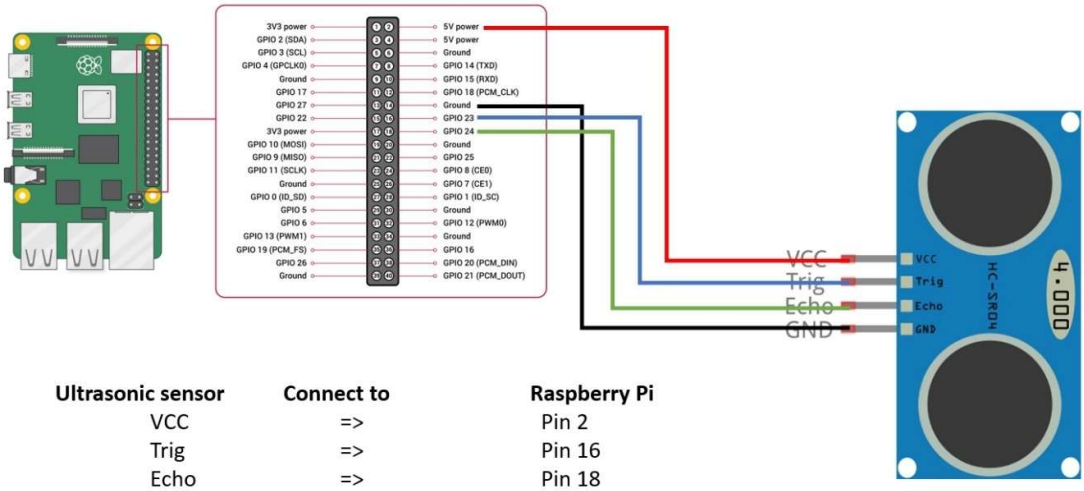
- Node-Red

Hardware Requirement

- **Rasbperry Pi 4**

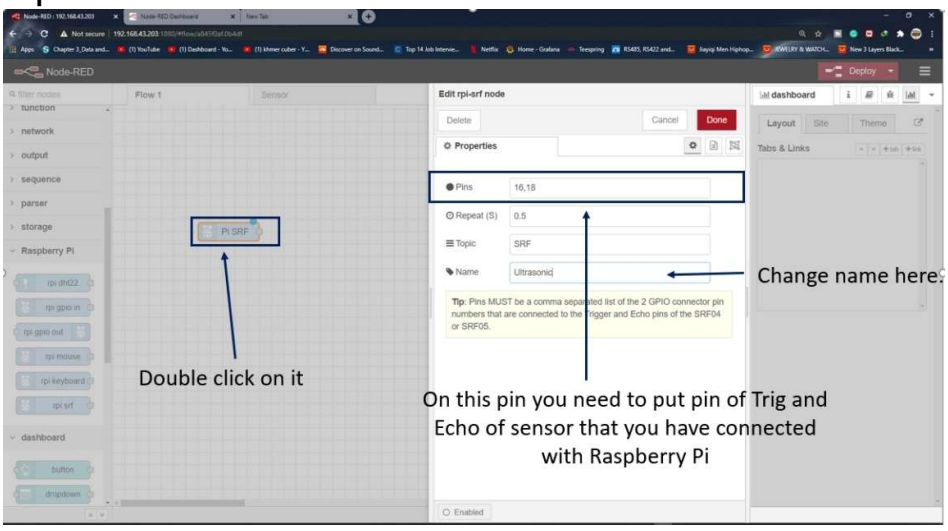
Steps:

Step 1 : Conect Ultrasonic Sensor with raspberry Pi 4



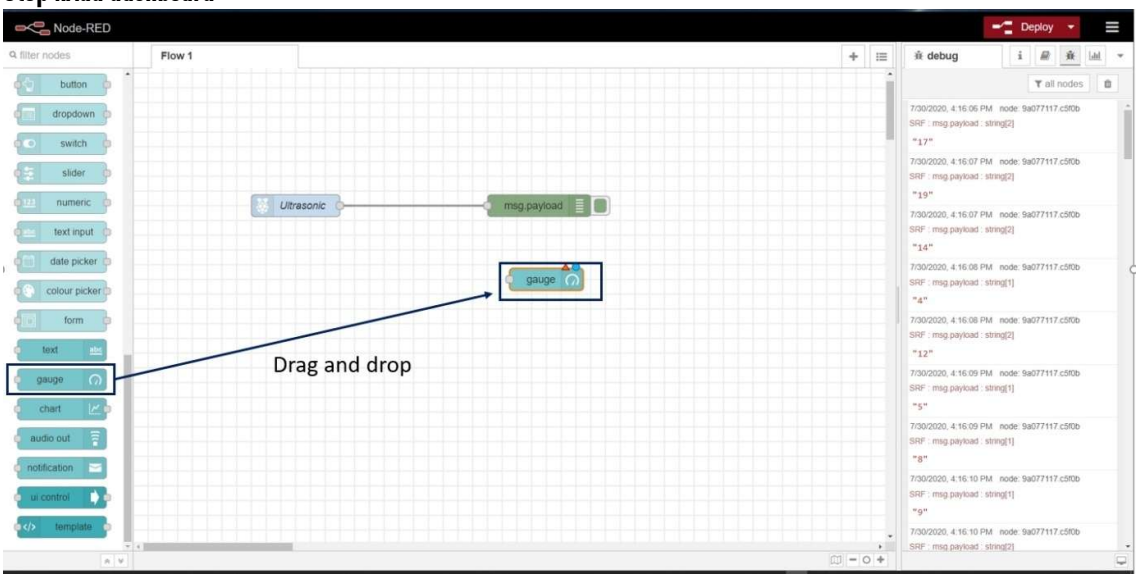
Ultrasonic sensor	Connect to	Raspberry Pi
VCC	=>	Pin 2
Trig	=>	Pin 16
Echo	=>	Pin 18
GND	=>	Pin 14

Step 2:



Step3:Add debug and connect

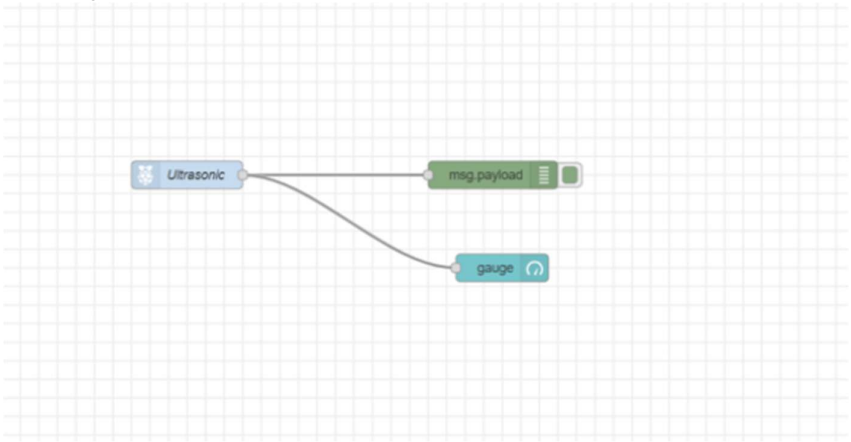
Step4:Add dashboard



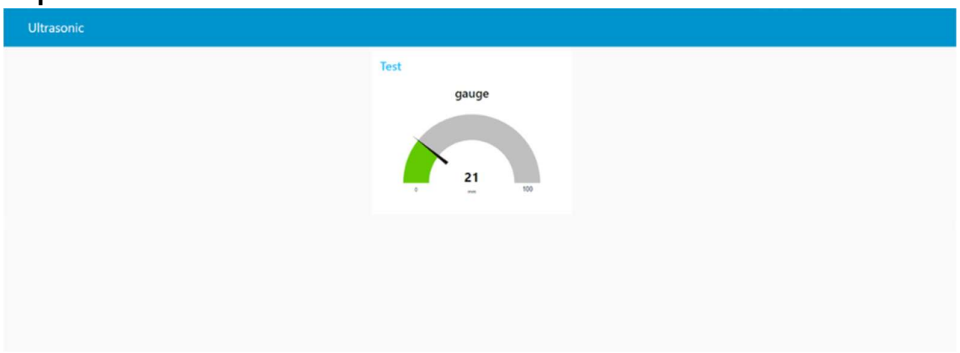
Double Click on gauge and create group and add into this.

Step5: Deploy the Flow

Flow Diagram:



Output:



Result:

Thus,the data aqusition from sensor and visualize in node-red using raspberry pi was successfully.

Exp : 13 Creating a Dashboard in Node Red.

Aim:

To create dashboard in node red using raspberry pi

Software Requirement:

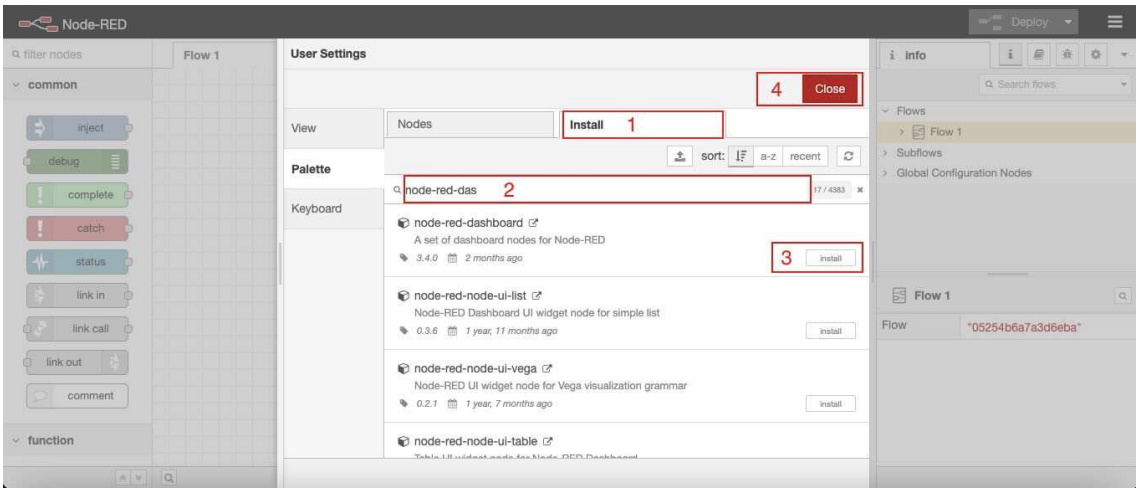
- Node-Red

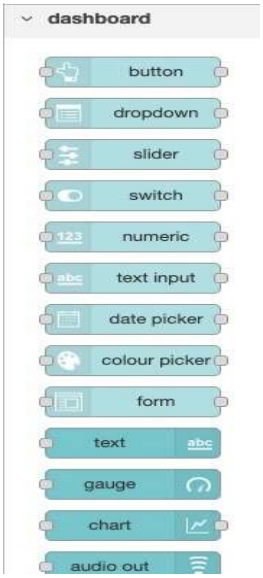
Hardware Requirement:

- **Rasbperry pi**

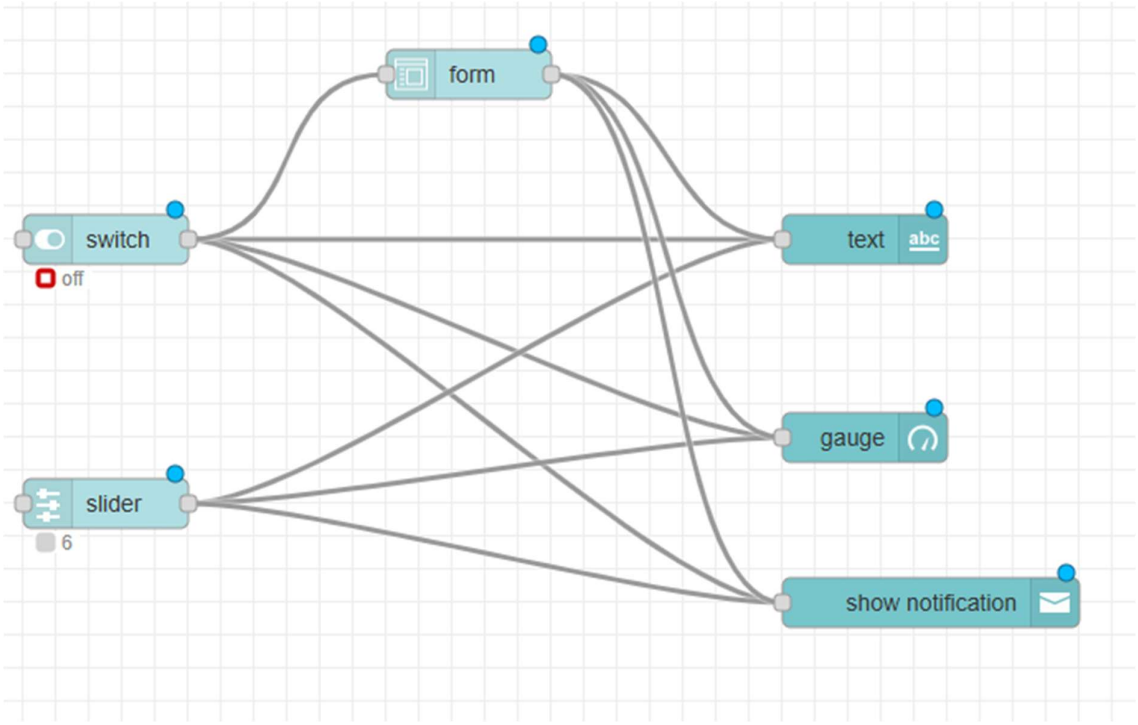
Steps:

Step 1:Install Prerequiresites





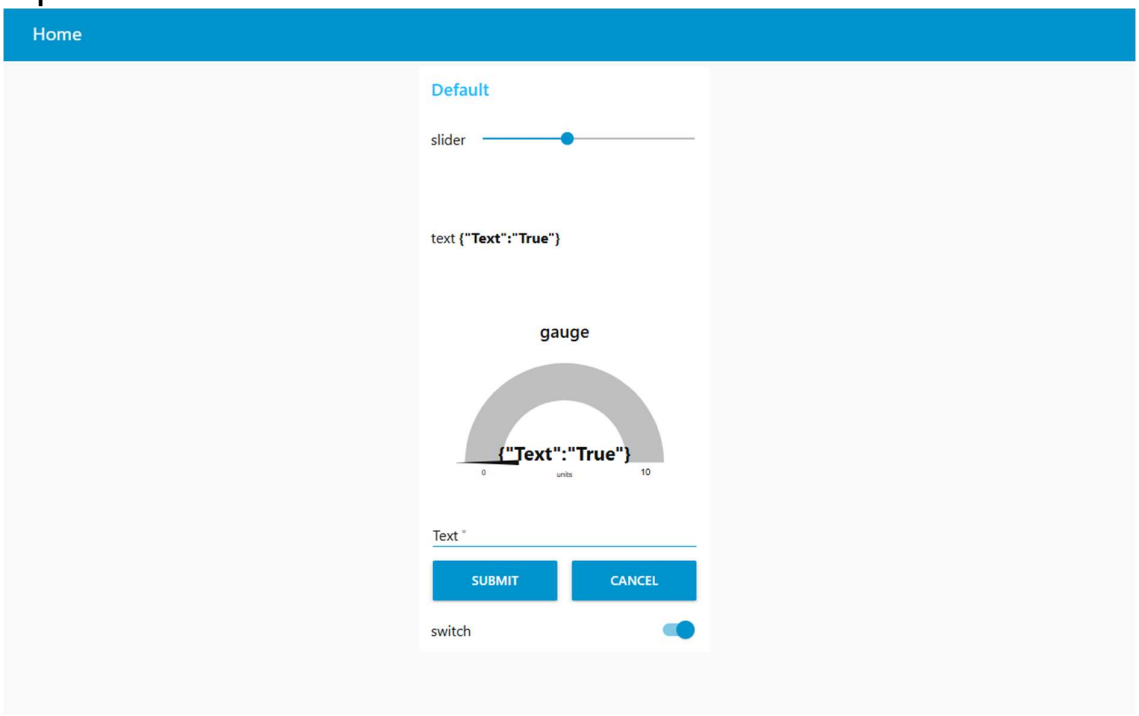
Step2 : Drag & Drop Need



Step 3: Deploy The Flow:

Step4:View Dashboard: http://127.0.0.1:1880/ui

Output:



Resullt:

Thus,creating the dashboard in node red using raspberry pi has been successfully.