IoT

*Lab Manual*

**List of Experiments**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Programs** | **Page No** |
| 01 | Sense the Available Networks Using Arduino | 1 |
| 02 | Measure the Distance Using Ultrasonic Sensor and Make Led Blink Using Arduino | 5 |
| 03 | Detect the Vibration of an Object Using Arduino | 9 |
| 04 | Connect with the Available Wi-Fi Using Arduino | 13 |
| 05  06 | Sense a Finger When it is Placed on Board Using  Arduino  Temperature Notification Using Arduino | 17  21 |
| 07  08 | LDR to Vary the Light Intensity of LED Using  Arduino  MySQL Database Installation in Raspberry Pi | 25  29 |

SQL Queries by Fetching Data from Database in Raspberry Pi 33

09

Switch Light On and Off Based on the Input of

10

User Using Raspberry Pi 35

vii

**1. SENSE THE AVAILABLE NETWORKS USING ARDUINO**

**AIM:**

To write a program to sense the available networks using Arduino.

**COMPONENTS REQUIRED:**

1. WiFi Module or ESP 8266 Module.
2. Connecting cable or USB cable.

**ALGORITHM:**

STEP 1: Start the process.

STEP 2: Start ->Arduino IDE -1.8.8

STEP 3: Then enter the coding in Arduino Software. STEP 4: Compile the coding in Arduino Software.

STEP 5: Connect the USB cable to WiFi module.

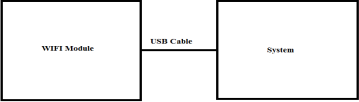
STEP 6: Select tools -> select board -> Module node Mch.0.9CE ESP

1.2 modules -> select port.

STEP 7: Upload the coding in ESP Module node Mch.0.9CE and open serial monitor to view the available networks.

STEP 8: Stop the process.

**BLOCK MODULE:**



**CODING:**

#include <ESP8266WiFi.h> void setup() { Serial.begin(115200); WiFi.mode(WIFI\_STA); WiFi.disconnect(); delay(100); Serial.println("Setup done");

}

void loop() { Serial.println("scan start"); int n = WiFi.scanNetworks(); Serial.println("scan done"); if (n == 0) {

Serial.println("no networks found");

} else { Serial.print(n);

Serial.println(" networks found"); for (int i = 0; i < n; ++i) { Serial.print(i + 1);

Serial.print(": "); Serial.print(WiFi.SSID(i)); Serial.print(" ("); Serial.print(WiFi.RSSI(i)); Serial.print(")");

Serial.println((WiFi.encryptionType(i) == ENC\_TYPE\_NONE) ? " " : "\*");

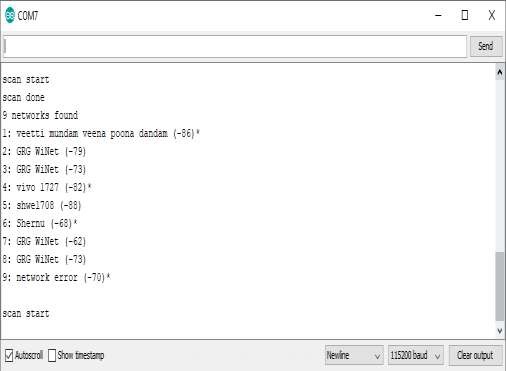
delay(10);

}

}

Serial.println(""); delay(5000);

}

**OUTPUT:**

**RESULT:**

Thus the output for sensing the available networks using Arduino has successfully executed.

**2. MEASURE THE DISTANCE USING ULTRASONIC SENSOR AND MAKE LED BLINK USING ARDUINO**

**AIM:**

To write a program to measure the distance using ultrasonic sensor and make LED blink using Arduino.

**COMPONENTS REQUIRED:**

1. Ultra sonic sensor.
2. Jumper wires.
3. Connecting cable or USB cable.

**ALGORITHM:**

STEP 1: Start the process.

STEP 2: Start ->Arduino IDE -1.8.8

STEP 3: Then enter the coding in Arduino Software. STEP 4: Compile the coding in Arduino Software.

STEP 5: In Arduino board, connect VCC to power supply 5V and connect to ground as in

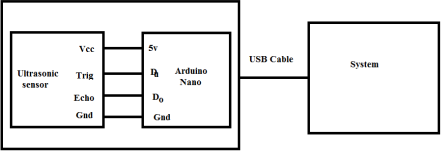
PIN gnd and connect trig to trigpio =9, connect echo to echopin=10 using jumper wires.

STEP 6: Connect the Arduino board with USB cable to the system. STEP 7: Select tools -> select board ->Arduino Nano -> select processor -> AT Mega 328 p and the select port.

STEP 8: Upload the coding in Arduino board and now for the LED to blink.

STEP 9: Then, the output will be displayed in the serial monitor. STEP 10: Stop the process.

**BLOCK MODULE:**



**CODING:**

const int trigPin = 9; const int techoPin = 10; long duration;

int distance; void setup()

{

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output pinMode(echoPin, INPUT); // Sets the echoPin as an Input Serial.begin(9600); // Starts the serial communication

}

void loop()

{

digitalWrite(trigPin, LOW);// Clears the trigPin delayMicroseconds(2);

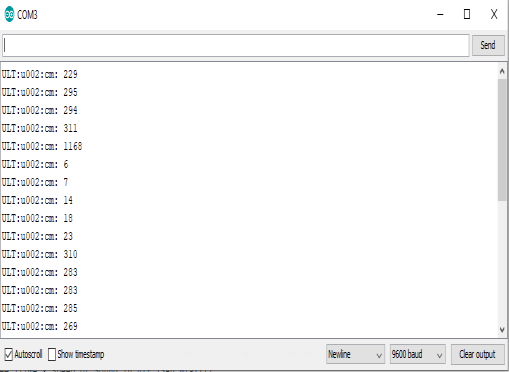
digitalWrite(trigPin, HIGH);// Sets the trigPin on HIGH state for 10 micro seconds

delayMicroseconds(10); digitalWrite(trigPin, LOW); duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.034 / 2; // distance= (Time x Speed of Sound in Air (340 m/s))/2

Serial.println(distance); delay(1000);

}

**OUTPUT:**

**RESULT:**

Thus the output for measuring the distance using ultrasonic sensor and LED blink using Arduino has successfully executed.

**3. DETECT THE VIBRATION OF AN OBJECT USING ARDUINO**

**AIM:**

To write a program to detects the vibration of an object with sensor using Arduino.

**COMPONENTS REQUIRED:**

1. Vibration sensor
2. Jumper wires
3. USB cable

**ALGORITHM:**

STEP 1: Start the process. STEP 2: StartArduino.1.8.8.

STEP 3: Then enter the coding in Arduino software.

STEP 4: In Arduino board, connect vcc to power supply 5V and connect do to analog pin A0 and connect gnd to ground gnd using jumper wires.

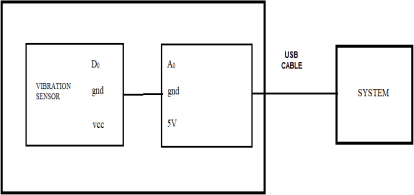
STEP 5: Connect the arduino board with the USB cable to the system.

STEP 6: Select toolsSelect boardArduino Nano gndSelect processor AT mega 823p and then select the port.

STEP 7: Upload the coding to the Arduino board.

STEP 8: Then the output will be displayed in the serial monitor. STEP 9: Stop the process.

**BLOCK DIAGRAM:**



**CODING:**

Int ledPin = 13; Int vib=A0; void setup()

{

pinMode(ledPin, OUTPUT);

pinMode(vib, INPUT); //set EP input for measurement Serial.begin(9600); //init serial 9600

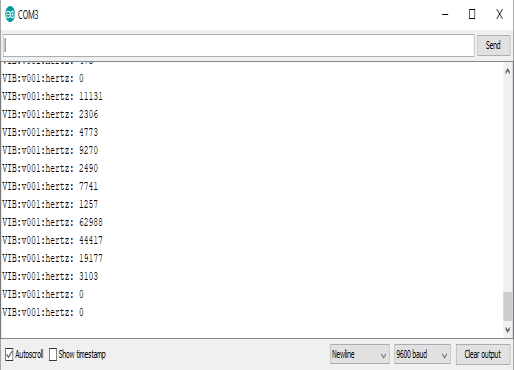
}

void loop()

{

long measurement=pulseIn (vib, HIGH); delayMicroseconds(50); Serial.print("VIB:v001:hertz: " ); Serial.println(measurement);

}

**OUTPUT:**

**RESULT:**

Thus the output for detecting the vibrations of an object with vibration sensor using Arduino has been successfully executed.

1. **CONNECT WITH THE AVAILABLE WI-FI USING ARDUINO AIM:**

To write a program to connect with the available Wi-Fi using

Arduino

**COMPONENTS REQUIRED:**

1. ESP 8266 module or Wi-Fi module
2. Connecting cables or USB cables

**ALGORITHM:**

STEP1: Start the process.

STEP2: StartArduino IDE 1.8.8.

STEP3: Include the file directory ESP 8266 in Arduino.

STEP4: Then enter the coding to Wi-Fi module or ESP 8266 module. STEP5: Then enter the coding in Arduino software.

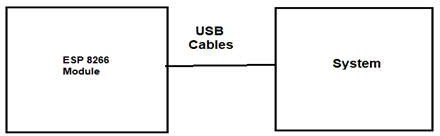
STEP6: Connect the USB cable to the Wi-Fi module and the Arduino connected system with available network.

STEP7: Select toolsSelect boardNode MCU 0.9C ESP-12 module and then SelectPort.

STEP8: Upload the coding to ESP 8266 module and open serial monitor to View the available network connects IP address.

STEP9: Stop the process.

**BLOCK DIAGRAM:**



**CODING:**

#include <ESP8266WiFi.h> // Include the Wi-Fi library

const char\* ssid = "Error"; // The SSID (name) of the Wi-Fi network you want to connect to

const char\* password = "networkerror"; // The password of the Wi-

Fi network void setup() {

Serial.begin(115200); // Start the Serial communication to send messages to the computer

delay(10); Serial.println('\n');

WiFi.begin(ssid, password); // Connect to the network Serial.print("Connecting to ");

Serial.print(ssid); Serial.print(“...") int i = 0;

while (WiFi.status() != WL\_CONNECTED) { // Wait for the Wi-Fi to connect

delay(1000);

Serial.print(++i); Serial.print(' ');

}

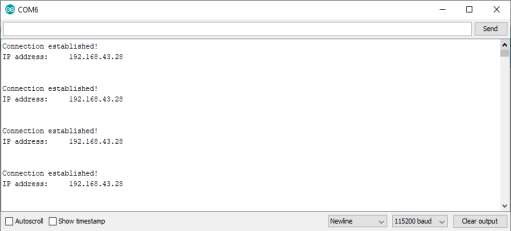
void loop() { Serial.println('\n');

Serial.println("Connection established!"); Serial.print("IP address:\t");

Serial.println(WiFi.localIP()); // Send the IP address of the ESP8266 to the computer

}

}

**OUTPUT:**

**RESULT:**

Thus the output for connecting with the available Wi-Fi using Arduino has been successfully executed.

**5. SENSE A FINGER WHEN IT IS PLACED ON BOARD USING ARDUINO**

**AIM:**

To write a program to sense a finger when it is placed on the board Arduino.

**COMPONENTS REQUIRED:**

1. Touch Sensor
2. Jumper wire
3. USB cable

**ALGORITHM:**

STEP 1: Start the process. STEP 2: Start Arduino 1.8.8

STEP 3: Then enter the coding in arduino software. STEP 4: Compile the coding in the arduino software.

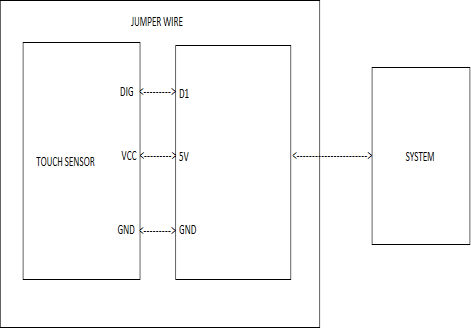
STEP 5: In arduino board, connect VCC to power supply 5v and connect SIG to Electrical signal DT and connect to ground and wing jumper wires.

STEP 6: Connect the arduino board with USB cable to the system. STEP 7: Select tools Select processor board and port.

STEP 8: Upload the coding to arduino board. Then the output will be displayed in the serial monitor.

STEP 9: Stop the process.

**BLOCK DIAGRAM:**



**CODING:**

int Led = 13 ; // define LED Interface

int buttonpin = 7; // define Metal Touch Sensor Interface int val ; // define numeric variables val

void setup ()

{

Serial.begin(9600);

pinMode (Led, OUTPUT) ; // define LED as output interface pinMode (buttonpin, INPUT) ; // define metal touch sensor output interface

}

void loop ()

{

val = digitalRead (buttonpin) ;

//Serial.println(val);

if (val == 1) // When the metal touch sensor detects a signal, LED flashes

{

digitalWrite (Led, HIGH); Serial.println(val); delay(1000);

}

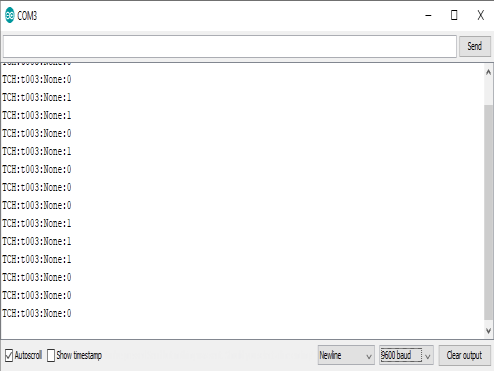
else

{

digitalWrite(Led,LOW); Serial.println(val); delay(1000);

}

}

**OUTPUT:**

**RESULT:**

Thus the output for sensing a finger when it is placed in board Arduino has been successfully executed.

**6. TEMPERATURE NOTIFICATION USING ARDUINO**

**AIM:**

To write a program to get temperature notification using Arduino.

**COMPONENTS REQUIRED:**

1. Temperature and humidity sensor.
2. Jumper wires
3. Connectivity cable or USB cable.

**ALGORITHM:**

STEP 1: Start the process. STEP 2: Start Arduino 1.8.8

STEP 3: Include the DHT library to the Arduino software. STEP 4: Then enter the coding in Arduino software.

STEP 5: Complete the coding in Arduino.

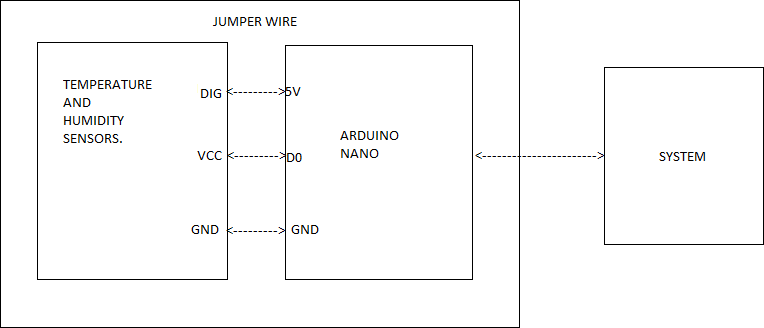
STEP 6: In Arduino board connect VCC to the power supply 5V and connect SIG to digital signal DT and connect SND to ground GND using jumper wires.

STEP 7: Connect the arduino board with USB cable to the system. STEP 8: Select tools  Selected.

STEP 9: Upload the coding to arduino board. Then the output will be displayed in the serial monitor.

STEP 10: Stop the process.

**BLOCK DIAGRAM:**



**CODING:**

#include <dht.h>

#define dht\_apin A0 // Analog Pin sensor is connected to dht DHT;

void setup()

{

pinMode(A0,INPUT); Serial.begin(9600); delay(500);

Serial.println("DHT11 Humidity & temperature Sensor\n\n"); delay(1000);

}

void loop()

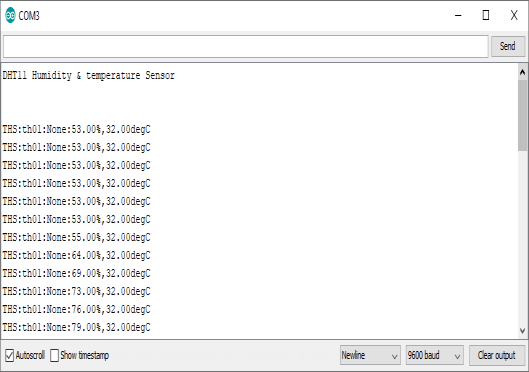
{

DHT.read11(dht\_apin); Serial.print("THS:th01:None:"); Serial.print(DHT.humidity); Serial.print("%,");

//Serial.print("temperature = "); Serial.print(DHT.temperature); Serial.println("degC");

delay(2000);//Wait 5 seconds before accessing sensor again.

}

**OUTPUT:**

**RESULT:**

Thus the output to get temperature notification using Arduino has successfully executed.

* 1. **LDR TO VARY THE LIGHT INTENSITY OF LED USING ARDUINO**

**AIM:**

To write a program for LDR to vary the light intensity of LED using Arduino.

**ALGORITHM:**

STEP1: Start the program.

STEP2: Start →Arduino 1.88[IDE].

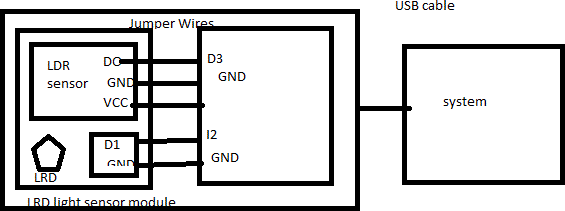
STEP3: Enter the coding in Arduino software. STEP4: Compile the coding in the Arduino software.

STEP5: From LDR light sensor module, connect VCC to power supply 5V and connect to digital pin D3 and connect GND to ground gnd using jumper wires to arduino board.

STEP6: For LED, connect D to digital pin D2 and connect GND to ground GND using jumper wires to arduino board.

STEP7: Show the variance of lights intensity in LED we use LDR light sensor module.

STEP8: Stop the process.



**CODING:**

const int ldr\_pin = 3; const int led\_pin = 2; void setup() { pinMode(ldr\_pin, INPUT);

pinMode(led\_pin, OUTPUT); Serial.begin(9600);

}

void loop() {

if ( digitalRead( ldr\_pin ) == 1) { digitalWrite(led\_pin, HIGH);

}

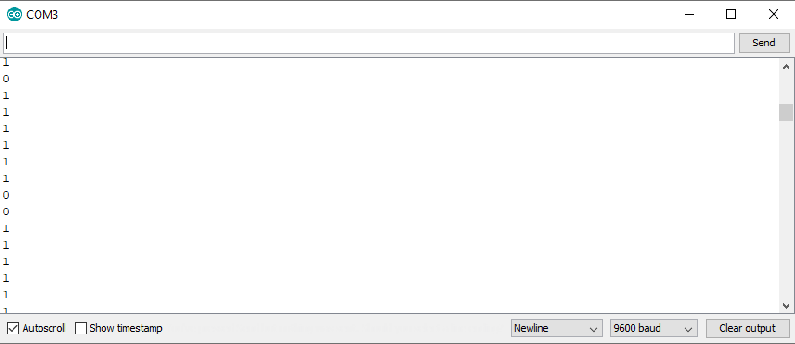
else {

digitalWrite(led\_pin , LOW);

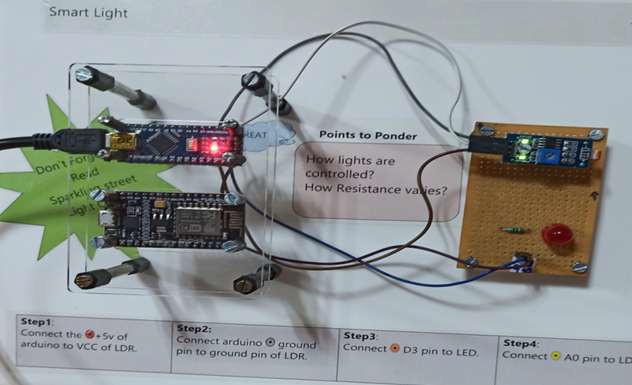
}

Serial.println(digitalRead( ldr\_pin )); delay(100);

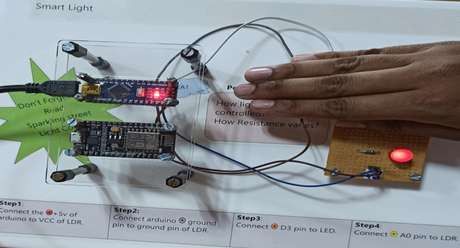
}

**OUTPUT:**

**LED OUTPUT:**

LED OFF

LED ON



**RESULT:**

Thus the output for LDR to vary the light intensity of LED using Arduino has successfully executed.

* 1. **MYSQL DATABASE INSTALLATION IN RASPBERRY PI AIM:**

To write a program to install MySQL database in Raspberry pi.

**COMPONENTS REQUIRED:**

1. Raspberrypi
2. HDMI
3. Micro USB power input

**ALGORITHM:**

STEP1: Start the process

STEP2: Connect micro USB power input to Raspberry pi.

STEP3: Connect HDMI to the system to act as monitor for Raspberry pi.

STEP4: Connect USB port to mouse and keyboard.

STEP5: then enter the coding in terminal for installing MySQL to Raspberry pi.

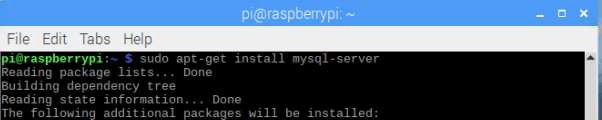
STEP6: stop the process.

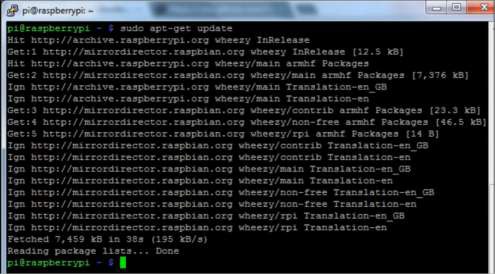
**CODING:**

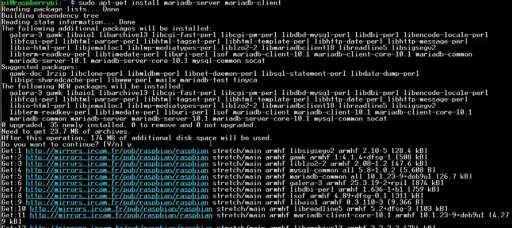
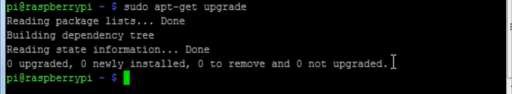
sudo apt-get install mysql-server sudo apt update

sudo apt upgrade

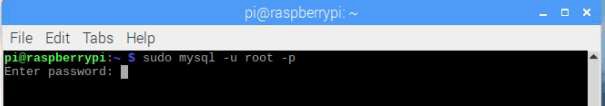
sudo apt install mariadb-server sudomysql\_secure\_installation sudomysql -u root –p

**OUTPUT:**









**RESULT:**

Thus the output to install MySQL database in Raspberry pi has successfully executed.

**9. SQL QUERIES BY FETCHING DATA FROM DATABASE IN RASPBERRY PI**

**AIM:**

To write a program to work with basic MySQL queries by fetching data from database in Raspberry pi.

**COMPONENTS REQUIRED:**

1. Raspberry pi
2. HDMI
3. Micro USB power input

**ALGORITHM:**

STEP1: Start the process.

STEP2: Connect micro USB power input to Raspberry pi.

STEP3: Connect HDMI to the system to act as monitor for Raspberry pi.

STEP4: Connect USB port 2.0 to mouse and keyboard.

STEP5: When enter the coding in the terminal to update and upgrade package using commands.

STEP6: Create database in MySQL and basic SQL queries by fetching data from database by using insert, update and delete queries.

STEP7: Stop the process.

**CODING:**

sudomysql -u root –p

CREATE DATABASE exampledb;

CREATE USER 'exampleuser'@'localhost' IDENTIFIED BY 'pimylifeup';

CREATE TABLE Books(Id INTEGER PRIMARY KEY, Title VARCHAR(100),

Author VARCHAR(60));

INSERT INTO Books(Title, Author) VALUES (1,‘War and Peace’,

‘Leo Tolstoy’);

SELECT \* FROM Books;

UPDATE Books SET Author='Lev Nikolayevich Tolstoy' WHERE Id=1;

DELETE FROM Books2 WHERE Id=1;

**OUTPUT:**

| Id | Title | Author |

+ + + +

+ + + +

+ + + +

| Id | Title | Author |

+ + + +

| 1 | War and Peace | Leo Tolstoy |

+ + + +

+ + + +

| Id | Title | Author |

+ + + +

| 1 | War and Peace | Lev Nikolayevich Tolstoy |

+ + + +

| Id | Title | Author |

+ + + +

+ + + +

**RESULT:**

The output to fetch data from database using SQL queries in Raspberry pi has successfully executed.

**10. SWITCH LIGHT ON AND OFF BASED ON THE INPUT OF USER USING RASPBERRY PI**

**AIM:**

To write a program to switch light on when the input is 1 and switch the light off when the input is 0 using Raspberry pi.

**COMPONENTS REQUIRED:**

1. Raspberry pi
2. Breadboard
3. Jumperwires
4. Resistor
5. LED

**ALGORITHM:**

STEP1: Start the process.

STEP2: Connect micro USB power input to Raspberry pi

STEP3: Connect HDMI to the system to act as monitor for Raspberry pi.

STEP4: Connect USB port 2.0 to mouse and keyboard.

STEP5: Enter the coding in the terminal for installing python and GPTO.

STEP6: Open notepa →enter coding →save as →file extension python or py.

STEP7: Copy file location → open terminal → paste file location in terminal → press enter.

STEP8: In the terminal window to get output enter 0 or 1, to switch light ON when the input is 1 and switch light OFF when the input is 0 in breadboard using Raspberry pi.

STEP9: Stop the process.

**CODING:**

sudo apt-get install python-pip sudo apt-get install python-dev sudo pip install RPi.GPIO

sudo –i #python

importRPi.GPIO as GPIO import time GPIO.setmode(GPIO.BCM) GPIO.setwarnings(False) GPIO.setup(18,GPIO.OUT)

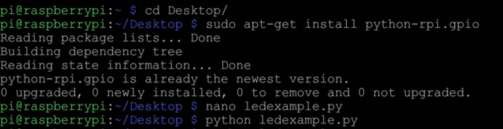
ip=int(input("enter the value: ")) ifip==1:

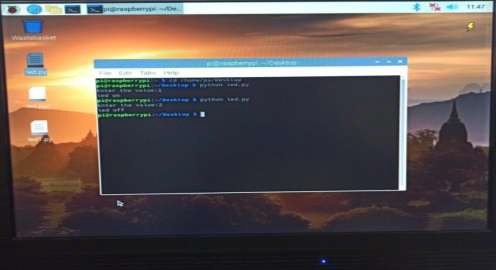
print "LED on" GPIO.output(18,GPIO.HIGH)

time.sleep(1) elifip==0:

print "LED off" GPIO.output(18,GPIO.LOW)

time.sleep(1)

**OUTPUT:**



**RESULT:**

Thus the output to switch light ON**/**OFF using Raspberry pi has been successfully executed.