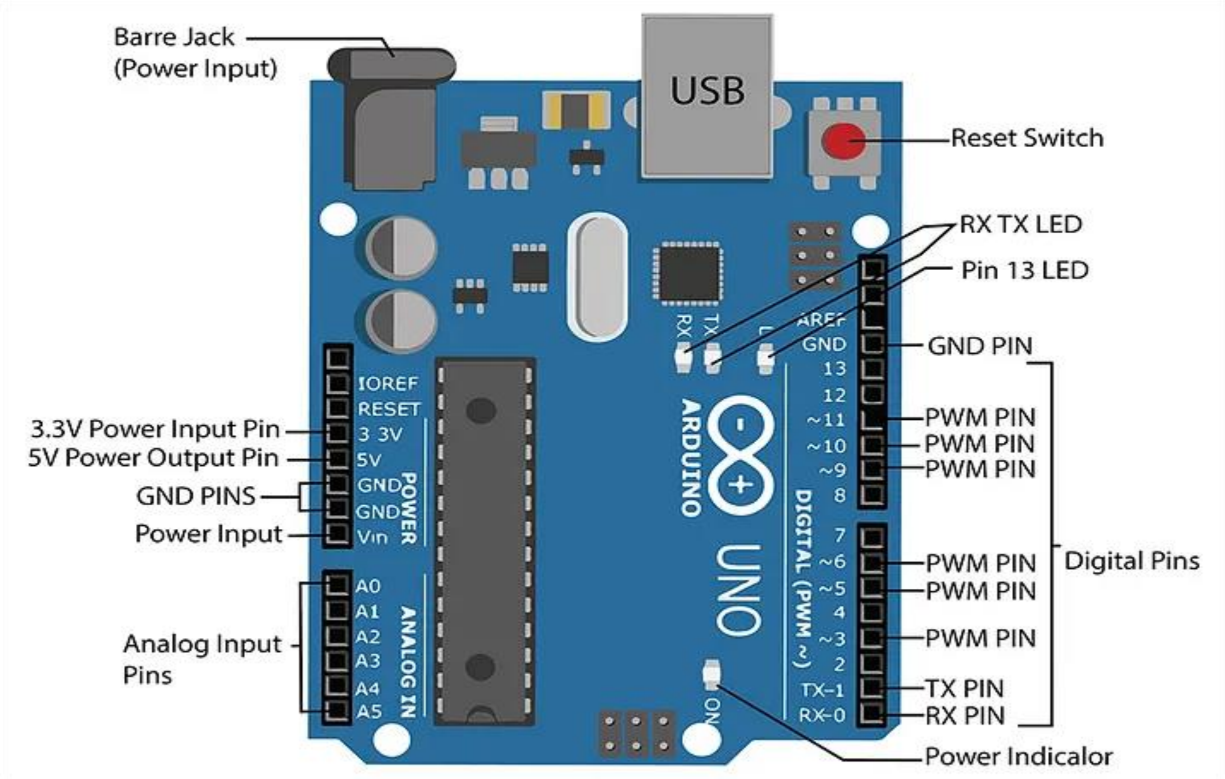
Experiment 1

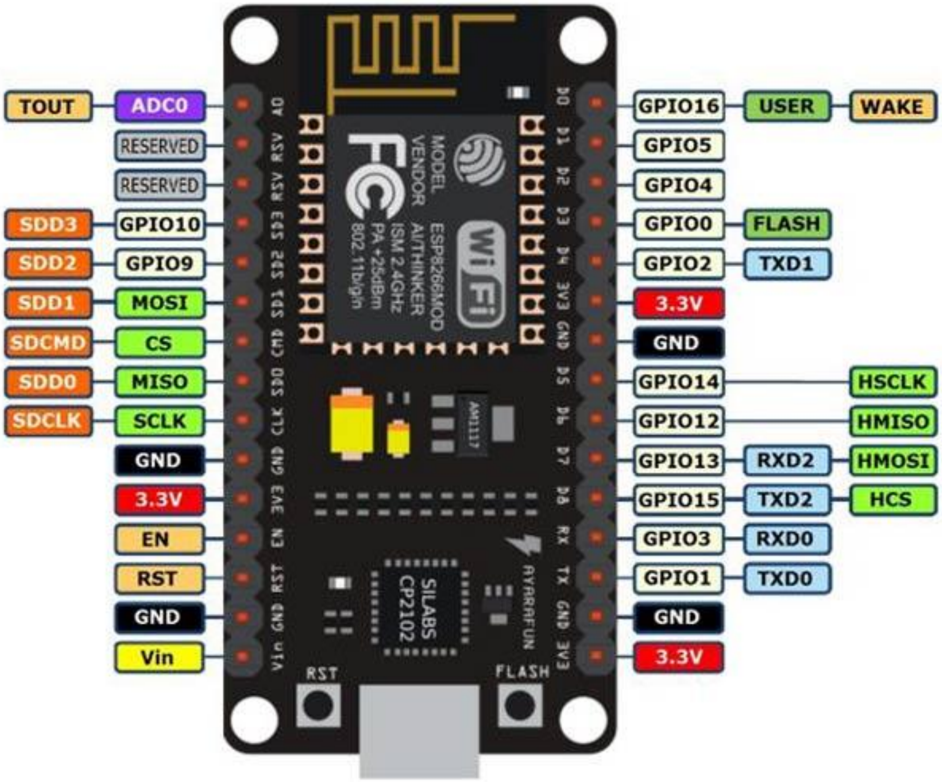
Name of the Experment:-LED blinking using ArduinoUno.



Arduino Uno R3 Pin Diagram

```
void setup() {
  // put your setup code here, to run once:
pinMode(2,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
digitalWrite(2,HIGH);
  delay (2000);
digitalWrite(2,LOW);
  delay (2000);
}
```

Name of the Experment:-LED blinking using Node MCU ESP8266



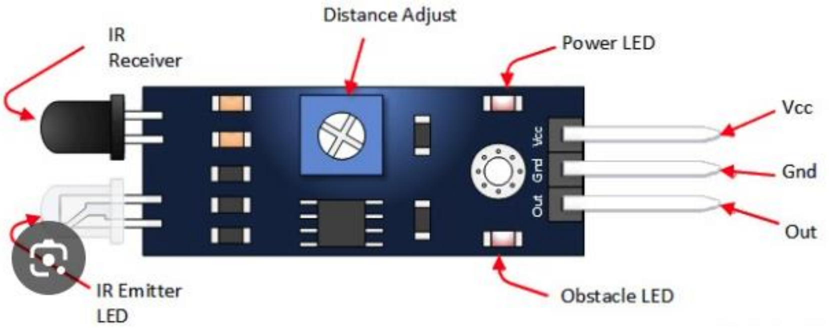
Node MCU ESP8266 Pin Diagram

Code:

```
void setup() {
  // put your setup code here, to run once:
pinMode(16,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
digitalWrite(16,HIGH);
  delay (2000);
digitalWrite(16,LOW);
  delay (2000);
}
```

Experiment 2

Name of the Experment:-IR Sensor Interfacing using Arduino Uno.



Code:

```
void setup() {
  pinMode(2,INPUT);
pinMode(3,OUTPUT);
}

void loop() {
if(digitalRead(2)==0)
{
digitalWrite(3,HIGH);
}
else {
digitalWrite(3,LOW);
}
}
```

IR Sensor Interfacing using NodeMCU ESP8266

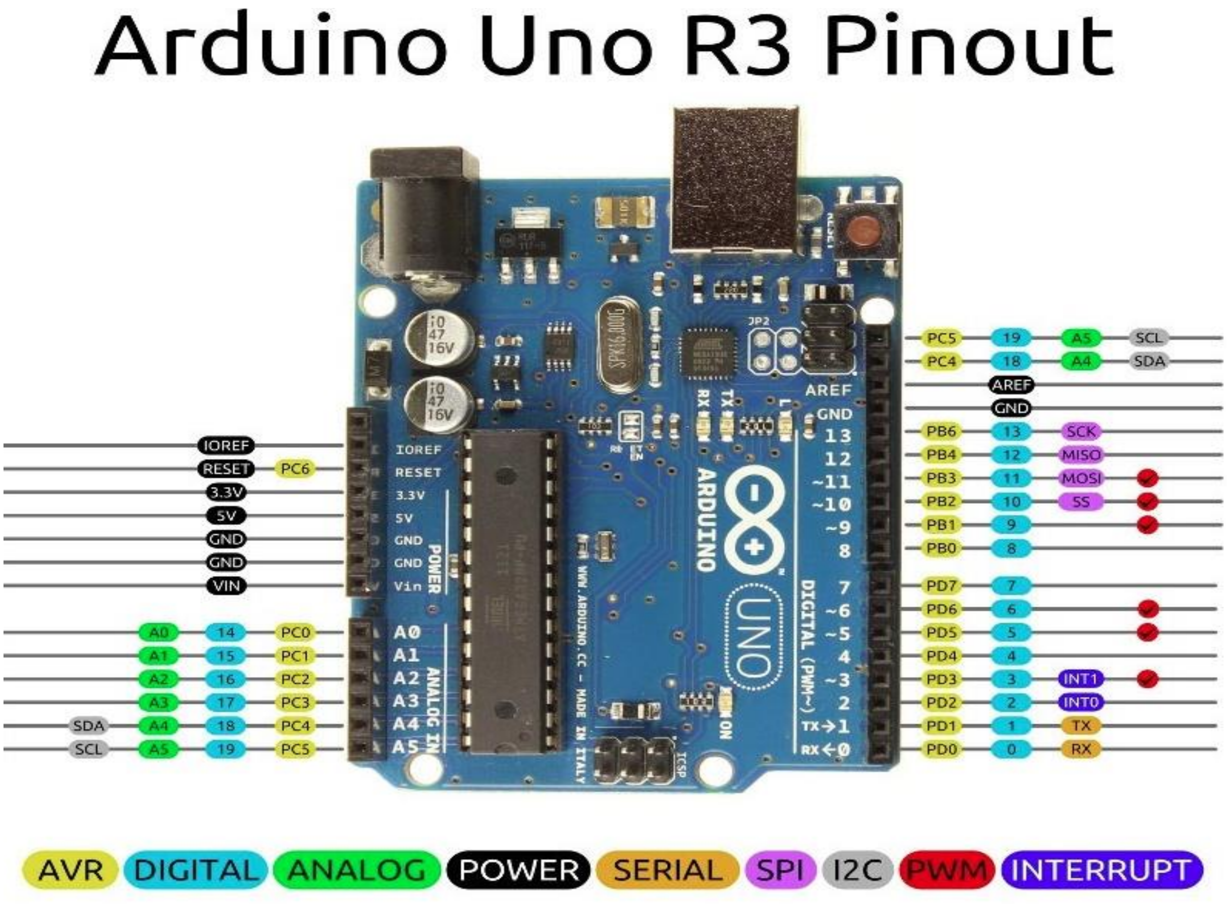
Code:

```
void setup() {
pinMode(16,INPUT);
pinMode(5,OUTPUT);
}

void loop() {
if(digitalRead(16)==0)
{
digitalWrite(5,HIGH);
}
else {
```

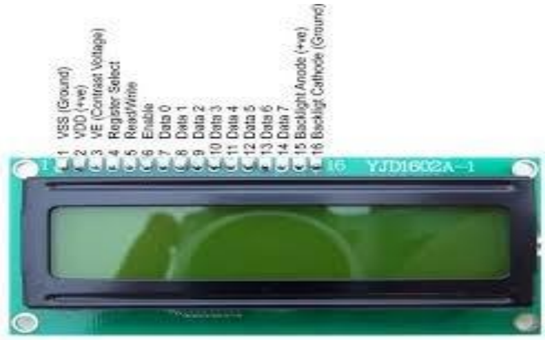
```
digitalWrite(5,LOW);
}
}
```

Experiment 3

Name of the Experment:-LCD & PIR Sensor interfacing with Arduino UNO.



1. This is the normal LCD Panel with 16 pins; it is driven by digital IO pins of the Arduino UNO.



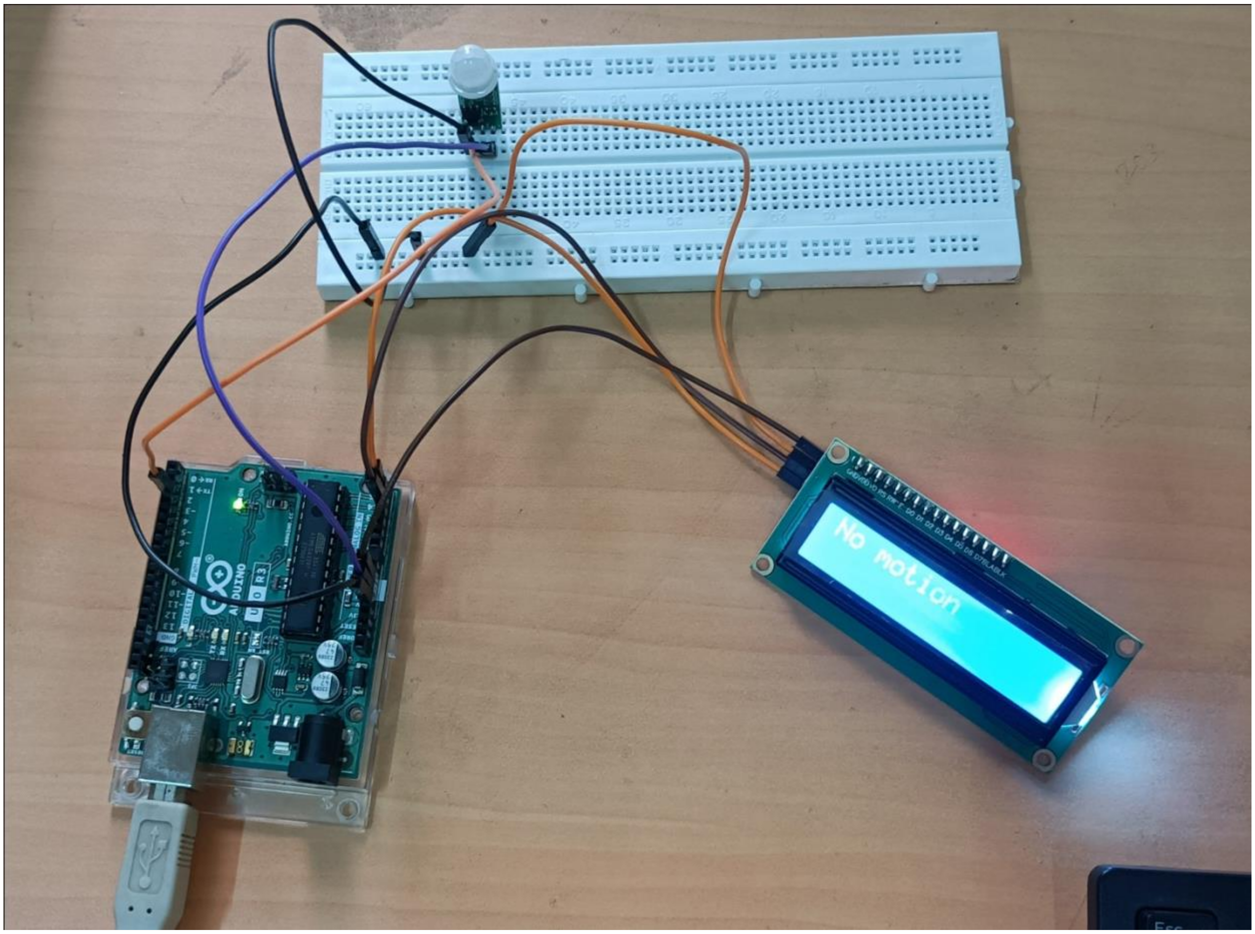
2. The LCD of Fig-1 can be operated using I2C Bus of Arduino UNO after installing an I2C Interface Board (the little one at the top of LCD) with the LCD (Fig-2, 3).

Figure-2:

Figure-3:

Code:

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
void setup() {
lcd.init();
lcd.backlight();
pinMode(2,INPUT);
}

void loop() {
  if(digitalRead(2)==1)
  {
lcd.print("Motion");
  }
  else {
lcd.print("No motion");
  }
delay(100);
lcd.clear();
}
```



Experiment No 4

Name of the Experiment:- LCD and Ultrasonic sensor interfacing with Arduino UNO.

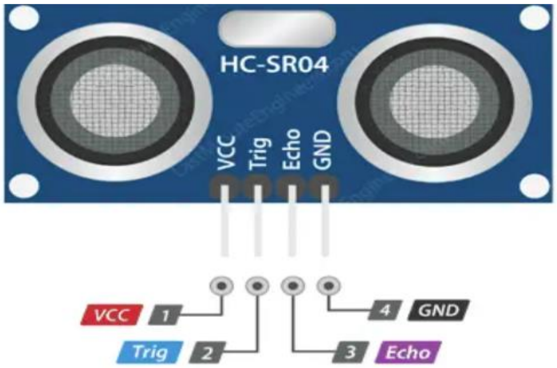


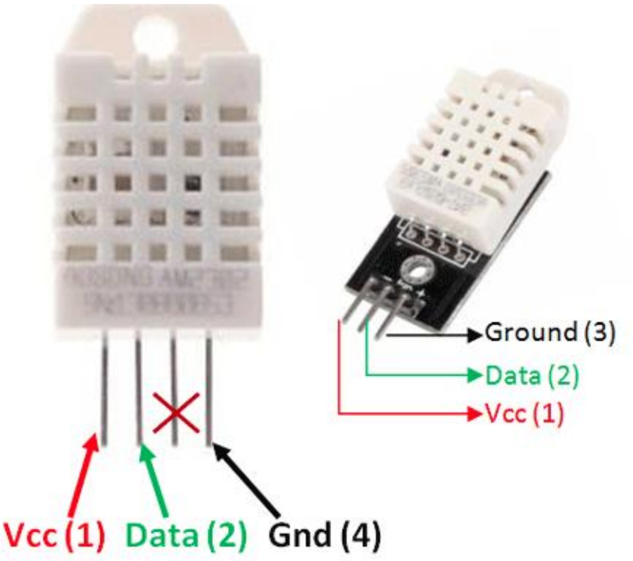
Fig.

```
#include <LiquidCrystal_I2C.h>
#include <NewPing.h>
```

```
NewPing ultra(2, 3, 500);
LiquidCrystal_I2C lcd(0x27, 16, 2);
int d;

void setup() {
  lcd.init();
  lcd.backlight();
}

void loop() {
  d = ultra.ping_cm();
  lcd.setCursor(0, 0);
  lcd.print("Distance: ");
  lcd.setCursor(10, 0);
  lcd.print(d);
  lcd.setCursor(13, 0);
  lcd.print("c.m.");
delay(100);
lcd.clear();
}
```

Experiment No 5

Name of the Experiment:-Temperature and humidity display using DHT11/22 sensor and Arduino Uno.

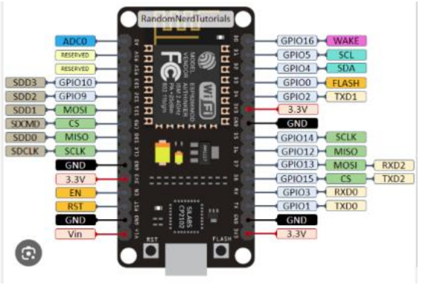


```
#include <LiquidCrystal_I2C.h>

#include <DHT.h>

DHT dht(2,DHT22);
```

```
LiquidCrystal_I2C lcd(0x27,16,2);
float t,h;
void setup() {
  // put your setup code here, to run once:
lcd.init();
lcd.backlight();
dht.begin();
}
void loop() {
  // put your main code here, to run repeatedly:
t=dht.readTemperature();
h=dht.readHumidity();
lcd.setCursor(0,0);
lcd.print("Temp: ");
lcd.setCursor(6,0);
lcd.print(t);
lcd.setCursor(11,0);
lcd.print("*C");
lcd.setCursor(0,1);
lcd.print("Humid: ");
lcd.setCursor(7,1);
lcd.print(h);
lcd.setCursor(12,1);
lcd.print("%");
delay(100);
lcd.clear();
}
```

Experiment No 6

Name of the Experiment: -Text message display using Bluetooth module & Node MCU ESP8266.



```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
String t;
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
lcd.init();
lcd.backlight();
}

void loop() {
  // put your main code here, to run repeatedly:
if(Serial.available())
{
  t=Serial.readString();
}
lcd.print(t);
delay(100);
lcd.clear();
}
```

Experiment No 7

7(a) Name of the Experiment: -Counter design using Microbit.

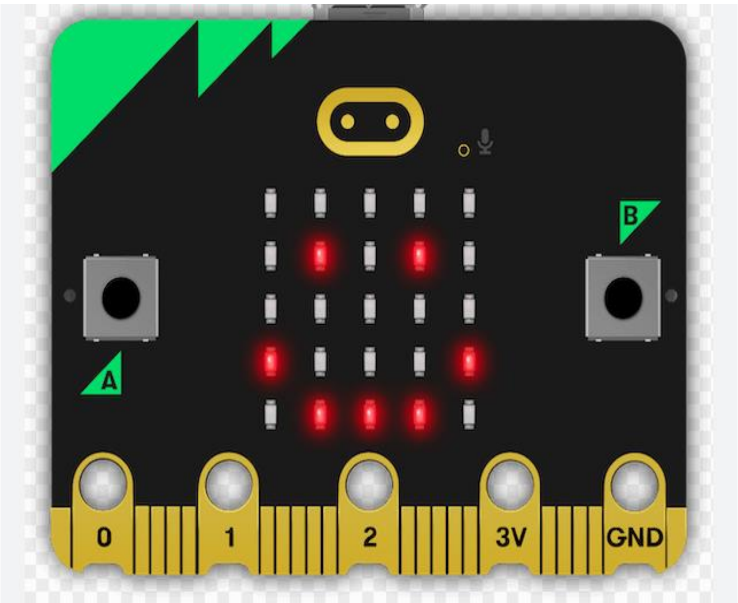


Fig.

Radio and Bluetooth Antenna: -Micro:bit can communicate with other micro:bits by radio, and with other devices using Bluetooth.

Processor and Temperature Sensor:-The micro:bit's processor is its brain, fetching, decoding and carrying out your instructions. It also contains a temperature sensor so you can measure how warm or cold your environment is.

Compass:-Find magnetic North or measure the strength of magnetic fields using the micro:bit's compass. It can measure magnetic fields in three dimensions, so you can use this for science experiments or for making simple door or window alarms.

Acclerometer:-The micro:bit's accelerometer measures forces in 3 dimensions, including gravity, so your projects can tell which way up your micro:bit is. You can use it for science experiments, add shake inputs to games or make simple alarms that alert you when someone moves your things.

Pins:- Connect your micro:bit to make headphones, simple switches and electronics, sense touch and more. The pins can power simple accessories like colourful lights, motors and robots.

```
from microbit import *

value=0
```

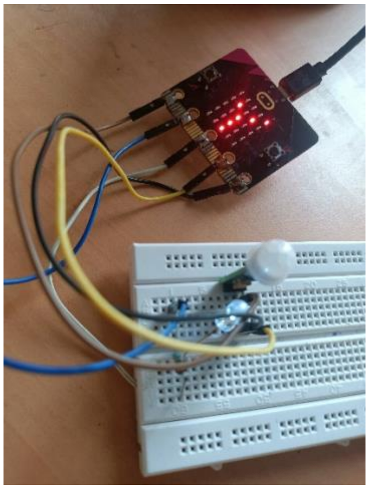
```
while True:
    if(value < 10):
        display.show(value)
    else:
        display.scroll(value)
    if(value < 50):
        value += 1
    else:
        value = 0
    sleep(1000)
```

7(b) **Name of the Experiment: -**Updown counter design using button of Microbit.

```
# Imports go at the top
from microbit import *


# Code in a 'while True:' loop repeats forever
val=0
while True:
    if(button_a.was_pressed()):
        val += 1
         display.show(val)
    elif(button_b.was_pressed()):
        val -= 1
        if(val<10):
           display.show(val)
        else:
           display.scroll(val)
    sleep(1000)
```

Experiment No 8

8. Name of the Experiment: -PIR sensor interfacing with Microbit.



Code:-

```
# Imports go at the top
from microbit import *



# Code in a 'while True:' loop repeats forever
while True:
    if(pin0.read_digital()==1):
        pin1.write_digital(1)
        display.show("Y")
    else:
        pin1.write_digital(0)
        display.show("N")
```

Experiment No 9

Name of the Experiment: -Wireless data communication using Microbit.

Transmission Code:-

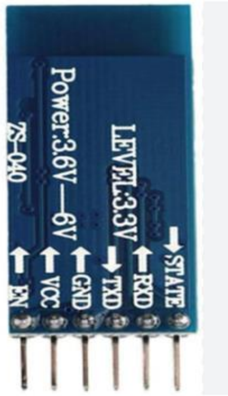
```
# Imports go at the top
from microbit import *
import radio
radio.config(group=33)
radio.on()
# Code in a 'while True:' loop repeats forever
while True:
    radio.send('HELLOICFAI')
```

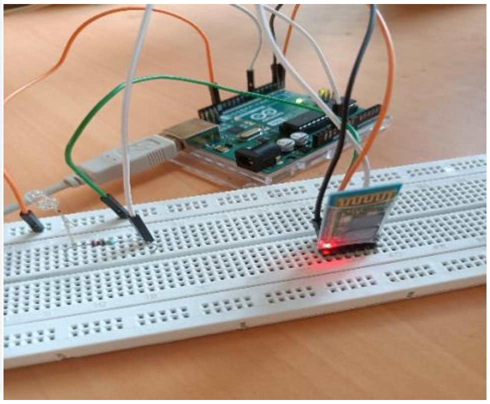
Receiving Code:-

```
# Imports go at the top
from microbit import *
import radio
radio.config(group=33)
radio.on()
# Code in a 'while True:' loop repeats forever
while True:
        msg=str(radio.receive())
        display.scroll(msg)
```

Experiment No 10

Name of the Experiment: -LED blinking using Bluetooth based mobile application.





```
String st;
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
pinMode(7,OUTPUT);
}


void loop() {
  // put your main code here, to run repeatedly:
if(Serial.available())
{
  st=Serial.readString();
}
if(st=="on")
{
  digitalWrite(7,HIGH);
}
else if(st=="off")
{
  digitalWrite(7,LOW);
}
}
```

Experiment No 11

Name of the Experiment: -LED blinking using ESP32 (inbuilt Bluetooth).

```
#include <BTAddress.h>
#include <BTAdvertisedDevice.h>
#include <BTScan.h>
#include <BluetoothSerial.h>

BluetoothSerial blt;
char st;
void setup() {
  blt.begin("ESP32 DEVKIT V1");
  pinMode(2,OUTPUT);
}

void loop() {
  if(blt.available()){
    st=blt.read();
  }
  if(st=='h'){
    digitalWrite(2,HIGH);
  }
  else if(st=='l'){
    digitalWrite(2,LOW);
  }
}
```

Experiment No 12

Name of the Experiment: -Temperature and humidity display in mobile app through ESP32.

```
#include <DHT.h>
#include <BTAddress.h>
#include <BTAdvertisedDevice.h>
#include <BTScan.h>
#include <BluetoothSerial.h>
BluetoothSerial blt;
DHT dht(2,DHT11);
float t,h;
char c;
void setup() {
  blt.begin("ESP32 DEVKIT V1");
  dht.begin();
}
void loop() {
  if(blt.available()){
    c=blt.read();
  }
  t=dht.readTemperature();
  h=dht.readHumidity();
  if(c=='1'){
    blt.print(t);
    blt.print("*C");
  }
  else if(c=='2'){
    blt.print(h);
    blt.print("%");
  }
  delay(1000);
  blt.flush();
}
```

Experiment No 13

Name of the Experiment: - Touch sensor based message sending through Microbit.

Transmitter Part: -

```
# Imports go at the top
from microbit import *
import radio
radio.config(group=43)
radio.on()

# Code in a 'while True:' loop repeats forever
while True:
    if pin_logo.is_touched():
        x="Welcome to ICFAI University"
        radio.send(x)
        display.scroll(x)
```

Receiver Part: -

```
# Imports go at the top
from microbit import *
```

```
import radio
radio.config(group=43)
radio.on()



# Code in a 'while True:' loop repeats forever
while True:
    st=str(radio.receive())
    display.scroll(st)
    sleep(100)
```

Experiment No 14

Name of the Experiment: - Motion detection and alert system using ESP32 and mobile app.

```
#include <BTAddress.h>
#include <BTAdvertisedDevice.h>
#include <BTScan.h>
#include <BluetoothSerial.h>
BluetoothSerial BT;
void setup() {
  // put your setup code here, to run once:
BT.begin("ESP32 DEVKIT V1");
pinMode(2,INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
if(digitalRead(2)==1)
{
  BT.print("1");
}
else
{
  BT.print("0");
}
delay(1000);
}
```

Experiment No 15

Name of the Experiment: - LED blinking through WiFi(ESP 8266)

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
// Replace with your network credentials

const char *ssid = "DRB";

const char *password = "43210000";
// Create an instance of the ESP8266WebServer class

ESP8266WebServer server(80);
#define LED_BUILTIN 2

// Pin connected to the LED

const int ledPin = LED_BUILTIN;
```

```
 // Change the pin if necessary
void setup() {

  // Set the LED pin as an output

  pinMode(ledPin, OUTPUT);
  // Connect to Wi-Fi

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {

    delay(1000);

    Serial.println("Connecting to WiFi...");

  }

  Serial.println("Connected to WiFi");



  // Define the HTTP routes

  server.on("/", HTTP_GET, handleRoot);

  server.on("/on", HTTP_GET, handleOn);

  server.on("/off", HTTP_GET, handleOff);
  // Start server
  server.begin();
  Serial.println("HTTP server started");
}

void loop() {
  // Handle client requests
  server.handleClient();
}

void handleRoot() {
  // HTML web page to control the LED with styling
  String htmlPage = "<html><head><style>";
  htmlPage += "body {font-family: Arial, sans-serif; text-align: center;}";
  htmlPage += "h1 {color: #333333;}";
  htmlPage += "a {display: inline-block; padding: 10px 20px; margin: 10px; text-decoration: none; font-size: 18px; color: #ffffff; background-color: #4CAF50; border-radius: 5px;}";

  htmlPage += "</style></head><body>";

  htmlPage += "<h1>NodeMCU LED Control</h1>";

  htmlPage += "<form action='/on'><input type='submit' value='Turn On'></form>";

  htmlPage += "<form action='/off'><input type='submit' value='Turn Off'></form>";

  htmlPage += "</body></html>";

  server.send(200, "text/html", htmlPage);

}



void handleOn() {

  // Turn on the LED

  digitalWrite(ledPin, HIGH);

  server.send(200, "text/plain", "LED turned on");

}
```

```
void handleOff() {
  // Turn off the LED
  digitalWrite(ledPin, LOW);
  server.send(200, "text/plain", "LED turned off");
}
```