EXPERIMENT 7

AIM – To interface Bluetooth with Arduino/Raspberry Pi and write a program to send sensor data to smart phone using Bluetooth.

THEORY - The objective of this experiment is to establish a communication link between a microcontroller (Arduino or Raspberry Pi) and a smartphone using Bluetooth technology. The primary goal is to collect sensor data from various sensors connected to the microcontroller and transmit this data to a smartphone application via Bluetooth.

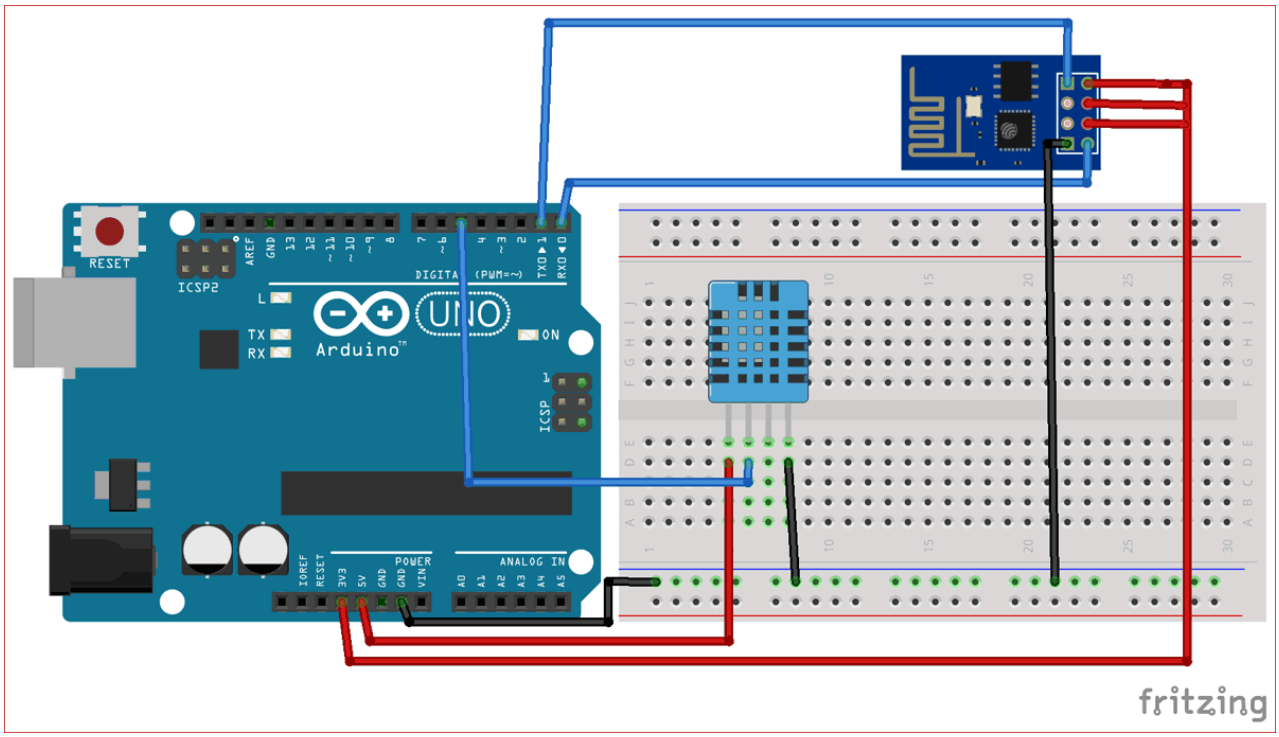
Applications

- **Smart Home Systems**: Enables remote monitoring and control of home conditions (e.g., temperature, humidity).
- **Health Monitoring**: Collects and transmits health-related data (e.g., heart rate) to a smartphone for analysis.
- **Environmental Monitoring**: Gathers data from sensors deployed in different locations, providing insights into environmental conditions.
- **Industrial IoT**: Facilitates real-time monitoring of industrial parameters, enhancing operational efficiency.

```
#define DHTPIN 2          // Pin where the DHT sensor is connected
#define DHTTYPE 11        // 11 for DHT11, 22 for DHT22
void setup() {
  Serial.begin(9600);   // Start the Serial communication
  pinMode(DHTPIN, INPUT_PULLUP); // Set the DHT pin as input with pull-up
}
void loop() {
  delay(2000);
  float humidity = readHumidity();
  float temperature = readTemperature();
  if (humidity >= 0 && temperature >= 0) {
    Serial.print("Humidity: ");
    Serial.print(humidity);
    Serial.print("%, Temperature: ");
    Serial.print(temperature);
    Serial.println("°C");
    Serial.print("H:");
    Serial.print(humidity);
    Serial.print(", T:");
```

```
    Serial.println(temperature);
  } else {
    Serial.println("Failed to read from DHT sensor!");
  }         }
float readHumidity() {
  return readDHTData()[0]; // Return humidity
}
float readTemperature() {
  return readDHTData()[1]; // Return temperature
}
float* readDHTData() {
  static float data[2]; // Array to hold humidity and temperature
  uint8_t bits[5];      // Buffer to hold the 5 bytes of data
  uint8_t i = 0;
  uint8_t j = 0;
  uint8_t laststate = HIGH;
  uint32_t count = 0;
  pinMode(DHTPIN, OUTPUT);
  digitalWrite(DHTPIN, LOW);
  delay(18);
  digitalWrite(DHTPIN, HIGH);
  delayMicroseconds(40);
  pinMode(DHTPIN, INPUT);
  for (i = 0; i < 85; i++) {
    count = 0;
    while (digitalRead(DHTPIN) == laststate) {
      count++;
      delayMicroseconds(1);
      if (count == 255) break; // Timeout    }
    laststate = digitalRead(DHTPIN);
    if (count == 255) break; // Timeout
    if (i >= 4 && i % 2 == 0) {
      bits[j / 8] <<= 1;
      if (count > 50) bits[j / 8] |= 1; // If the pulse is longer than 50us, it's a 1
```

```
    j++;    } }
  data[0] = bits[0]; // Humidity
  data[1] = bits[2]; // Temperature
  if (bits[4] != (bits[0] + bits[2])) {
    data[0] = -1; // Invalid humidity
    data[1] = -1; // Invalid temperature  }
  return data;
}
```



RESULT – Successfully interfaced Bluetooth with Arduino to send sensor data to smart phone using Bluetooth.

EXPERIMENT 8

AIM – To interface Bluetooth with Arduino/Raspberry Pi and write a program to turn LED ON/OFF when ‘1’/’0’ is received from smart phone using Bluetooth.

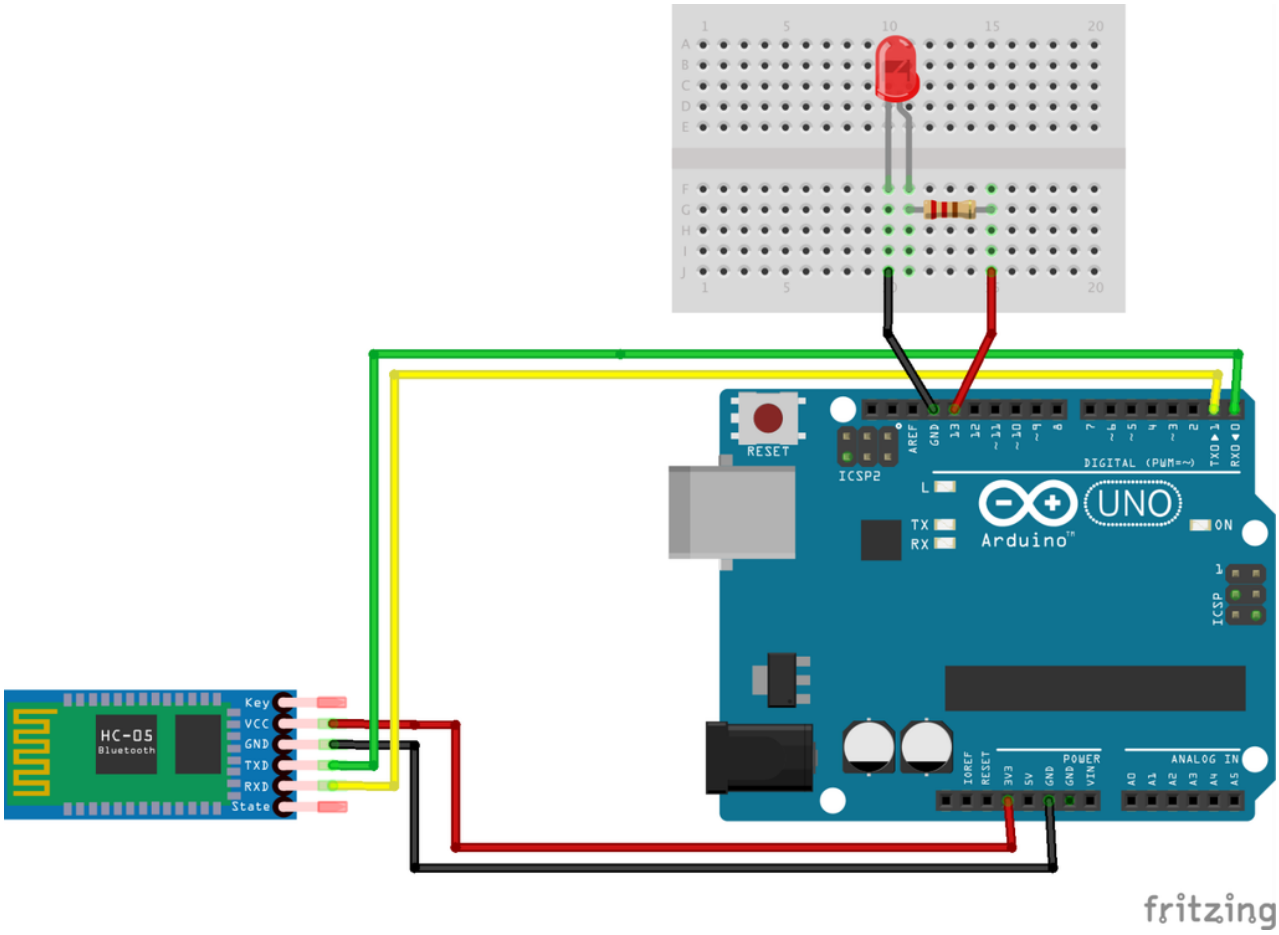
THEORY - This experiment investigates the integration of Bluetooth technology with Arduino or Raspberry Pi to facilitate wireless communication for controlling electronic devices. By utilizing a Bluetooth module (such as HC-05 or HC-06), we establish a connection between a smartphone and the microcontroller, allowing the transfer of data over short distances.

The primary objective is to control an LED based on commands sent from a smartphone application. When a user sends a '1' via Bluetooth, the LED will turn ON, and when a '0' is received, the LED will turn OFF. This interaction demonstrates fundamental concepts of IoT and wireless communications, highlighting how various devices can interact seamlessly.

CODE –

```
char junk;
String inputString="";
void setup()                  // run once, when the sketch starts
{
 Serial.begin(9600);           // set the baud rate to 9600, same should be of your Serial Monitor
 pinMode(13, OUTPUT);
}
void loop()
{
  if(Serial.available()){
  while(Serial.available())
    {
      char inChar = (char)Serial.read(); //read the input
      inputString += inChar;        //make a string of the characters coming on serial
    }
    Serial.println(inputString);
    while (Serial.available() > 0)
    { junk = Serial.read() ; }      // clear the serial buffer
    if(inputString == "a"){         //in case of 'a' turn the LED on
      digitalWrite(13, HIGH);
    }else if(inputString == "b"){   //incase of 'b' turn the LED off
      digitalWrite(13, LOW);
    }
```

```
    inputString = "";
  }
}
```



RESULT – Successfully interfaced Bluetooth with Arduino to turn LED ON/OFF when ‘1’/’0’ is received from smart phone using Bluetooth.