EXP NO: **DATE:**

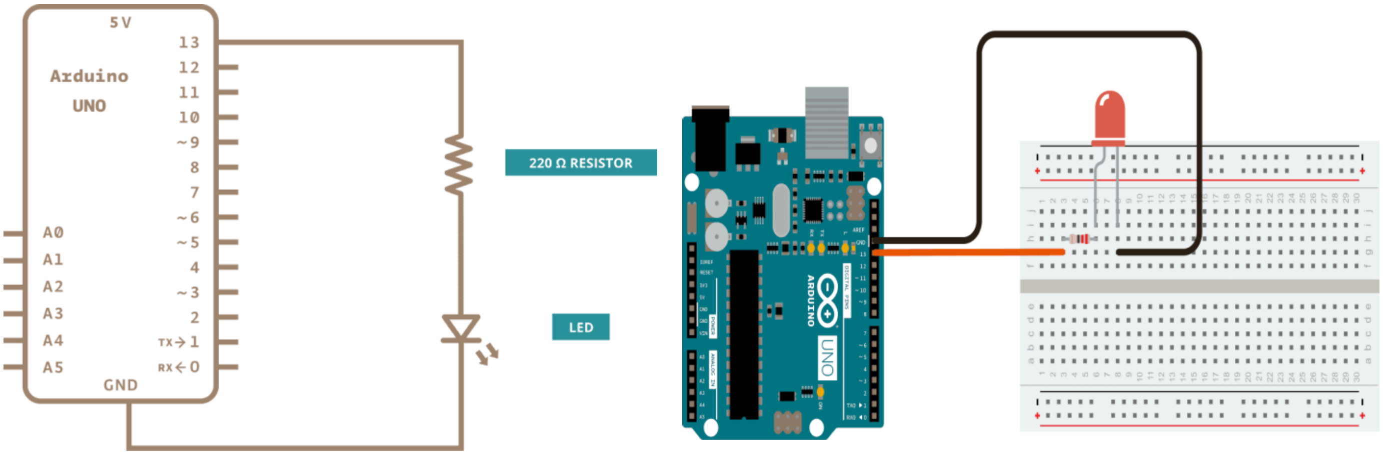
AIM:

To Blink the on-board LED of Arduino uno continuously.

APPARATUS REQUIRED:

- Arduino Development Board
- Arduino IDE Software

SCHEMATIC:



PROGRAM:

```
void setup()
{
  pinMode (LED_BUILTIN, OUTPUT);  // initialize digital pin LED_BUILTIN as an output.
}
void loop()  // the loop function runs over and over again forever
{
  digitalWrite (LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);        // wait for a second
  digitalWrite (LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);     // wait for a second

}
```

Result: Thus blink the LED of Arduino Uno has been written and executed successfully.

EXP NO: **DATE:**

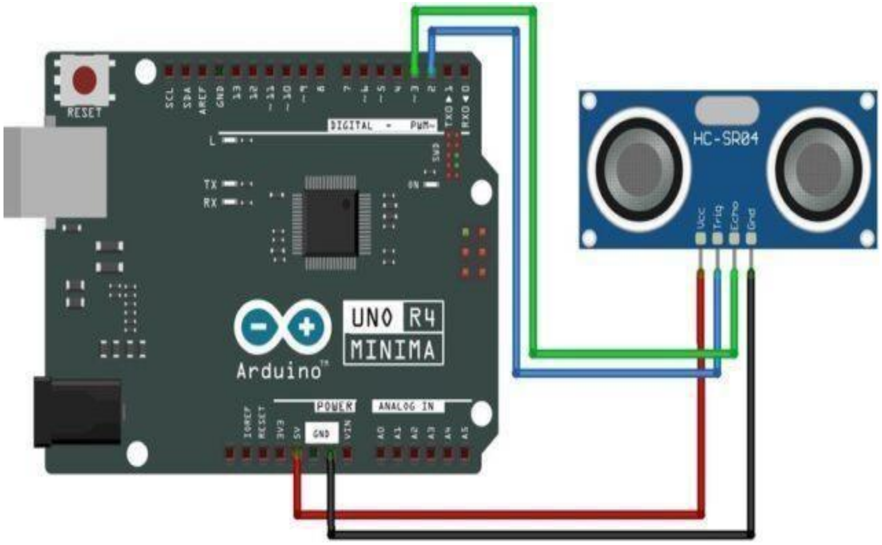
AIM:

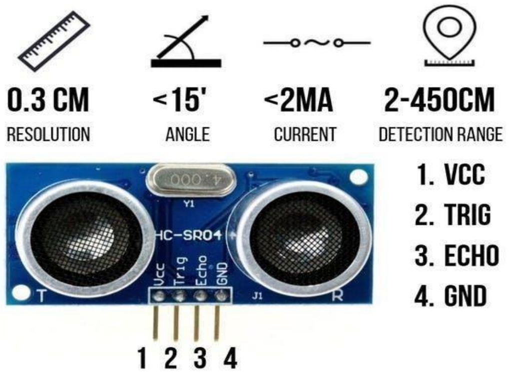
To identify the obstacle and the measure the distance of the obstacle using the ultrasonic sensor and display the distance in serial monitor.

APPARATUS REQUIRED:

- Arduino Development Board
- Arduino IDE Software
- Ultrasonic sensor - HC-SR04 Module
- BreadBorad
- Jumper wires

SCHEMATIC:





PROGRAM:

```
#define echoPin 2 // attach pin D2 Arduino to pin Echo of HC-SR04

#define trigPin 3 //attach pin D3 Arduino to pin Trig of HC-SR04

// defines variables

long duration; // variable for the duration of sound wave travel

int distance; // variable for the distance measurement
```

```
void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Serial Monitor
  Serial.println("with Arduino UNO R3");
}
void loop() {
  digitalWrite(trigPin, LOW);   // Clears the trigPin condition
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);   // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  delayMicroseconds(10);
  digitalWrite(trigPin,  LOW);
  duration = pulseIn(echoPin, HIGH);   // Reads the echoPin, returns the sound wave travel time in microseconds
  // Calculating the distance
  distance = duration * 0.034 / 2;  // Speed of sound wave divided by 2 (go and back)
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
}
```

Result :

Thus the program for measuring the distance to an obstacle was successfully written and executed, with the distance measured in centimeters and displayed on the serial monitor using an ultrasonic sensor.

EXP NO: **DATE:**

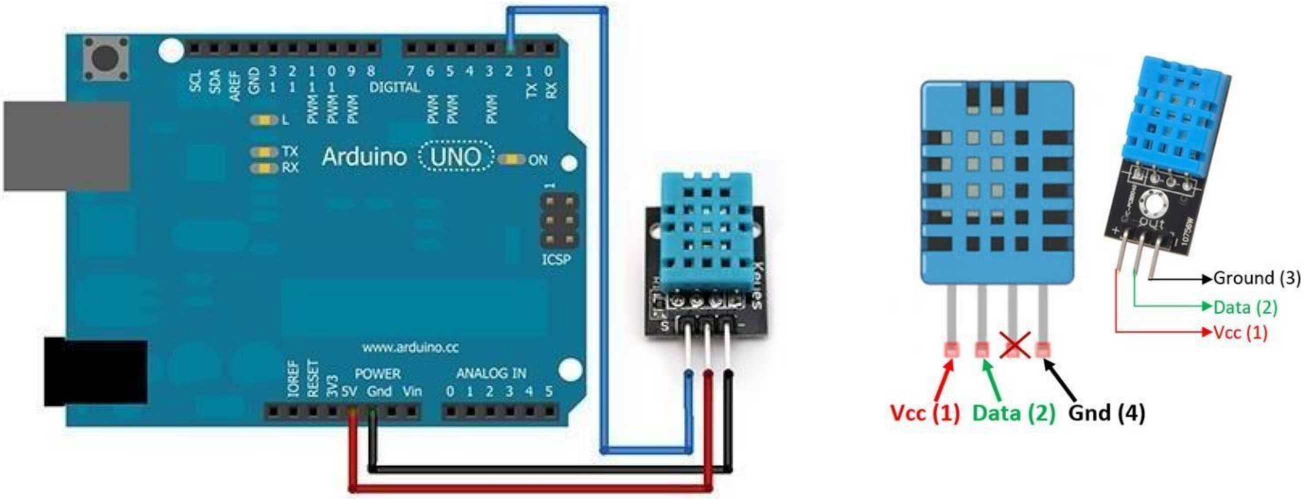
AIM:

To find the humidity and temperature in the atmosphere using humidity and temperature sensor and display the results in serial monitor.

APPARATUS REQUIRED:

- Arduino Development Board
- Arduino IDE Software
- Humidity and temperature sensor – DHT11
- BreadBorad
- Jumper wires

SCHEMATIC:



PROGRAM:

```
#include <DHT11.h>

DHT11 dht11(2);

void setup() {

   Serial.begin(9600);

   }
```

```
void loop() {
  int temperature = 0;
  int humidity = 0;
  int result = dht11.readTemperatureHumidity(temperature, humidity);
  if (result == 0) {
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.print(" °C\tHumidity: ");
    Serial.print(humidity);
    Serial.println(" %");
  }
  else {
    Serial.println(DHT11::getErrorString(result));
  }
}
```

Result:

Thus, the program to find the humidity and temperature in the atmosphere using humidity and temperature sensor has been written and executed successfully and the temperature and humidity values displayed in serial monitor.

EXP NO: **DATE:**

AIM:

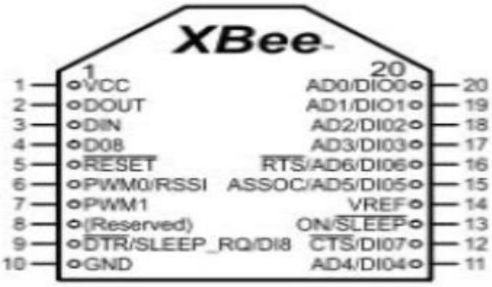
To write the program to establishes communication between the Arduino and Zigbee module.

APPARATUS REQUIRED:

- Arduino Development Board
- Arduino IDE Software
- XBee Pro
- BreadBorad
- Jumper wires

ZIGBEE MODULE:

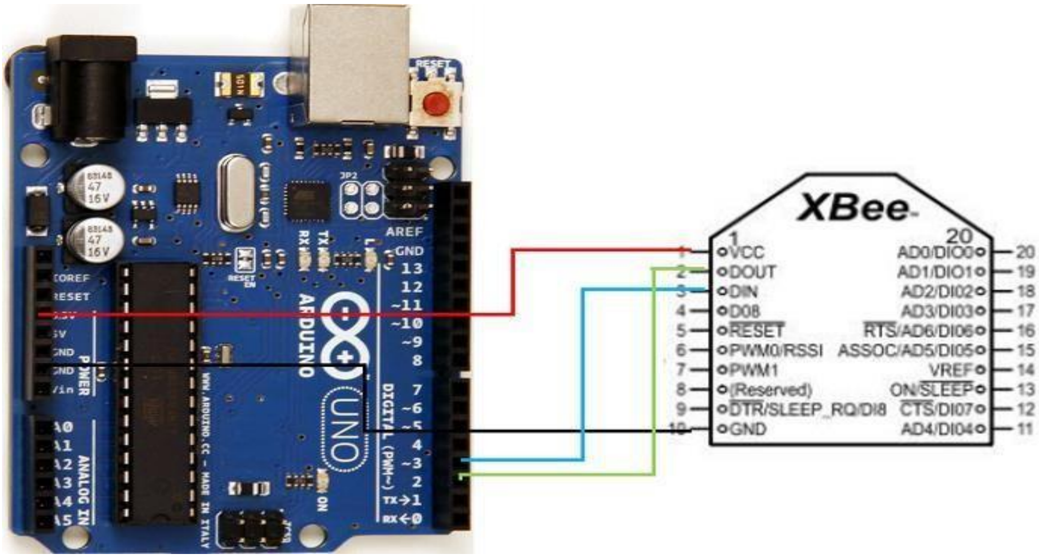
Zigbee is a wireless communication protocol targeted for battery powered devices (it has both low power and low cost). It generally operates in the 2.4GHz range and supports data ranges from 20 to 250 kbits/s.



Digi International developed the XBee modules, a family of wireless communication devices. These modules support various wireless communication protocols, including Zigbee, Wi-Fi, and cellular, and they can communicate over UART (Universal Asynchronous Receiver-Transmitter) interfaces.

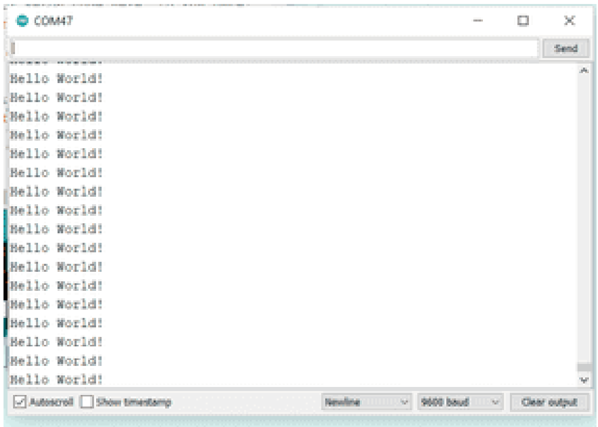
These modules utilize Zigbee communication, which is a low-power wireless communication protocol commonly used in home automation, industrial control, and sensor networks. XBee Zigbee modules are capable of forming mesh networks, making them suitable for applications where reliability and long-range communication are essential.

SCHEMATIC:



PROGRAM:

```
#include <SoftwareSerial.h>
SoftwareSerial zigbeeSerial(2, 3); // RX (pin 1), TX (pin 2)
void setup()
{
  // Initialize the Hardware Serial interface (usually connected to your computer)
  Serial.begin(9600); // Use the appropriate baud rate
  // Initialize the Software Serial interface (for the Zigbee module)
  zigbeeSerial.begin(9600); // Use the same baud rate as your Zigbee module
}
void loop()
{
  // Read data from Zigbee
  if (zigbeeSerial.available()) {
    char receivedChar = zigbeeSerial.read();
    Serial.print("Received: ");
    Serial.println(receivedChar); // Print received data to the Serial Monitor
  }
  // Send data to Zigbee
  zigbeeSerial.print("Hello, Zigbee!"); // Send a message to the Zigbee module
  delay(1000); // Add a delay to control message transmission rate
}
```



```
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
```

☑ Autoscroll □ Show timestamp Newline 9600 baud Clear output

Result :
Thus, the program to establish communication between the Arduino and a Zigbee module was successfully written and executed, with the outputs displayed in both the Serial monitor.