Ex. No. 6   **INTERFACE WITH ZIGBEE AND TRANSMIT SENSOR DATA**
TO OTHER NODE

AIM:

To set up a sensor node to read data from a temperature sensor and transmit it wirelessly using Zigbee.

REQUIREDCOMPONENTS:

- Arduino board (e.g., Arduino Uno)
- Zigbee module (e.g., XBee Series 2)
- Temperature sensor (e.g., DHT11 or DHT22)
- Jumper wires

EXPERIMENTAL SETUP:

1.**Connect Temperature Sensor:**
    Connect the temperature sensor to the Arduino following its datasheet.
Usually, connect the VCC pin to 5V, GND pin to GND, and DATA pin to a digital pin on the Arduino (e.g., pin 2).
2.**Connect Zigbee Module:**
- Connect the Zigbee module to the Arduino.
- Connect VCC to 5V, GND to GND, TX to RX, and RX to TX.
- Ensure proper wiring based on your Zigbee module's specifications.
3.**Upload Sensor Node Code:**

Write a code to read data from the temperature sensor and transmit it using Zigbee.
Modify the code to use the appropriate digital pin for the sensor and configure the Zigbee module.

```
#include <DHT.h>
#define DHT_TYPE DHT11
#define DHT_PIN 2

void setup ()
{
  // Initialize Zigbee module and sensor
  // Setup Serial communication for Zigbee
}

void loop() {
  float temperature = read Temperature ();
  send Temperature Data(temperature);
  delay (2000);  // Adjust delay as needed
}

float read Temperature () {
  // Implement temperature reading logic here using DHT library
}
```

```
void send Temperature Data (float temp) {
 // Implement Zigbee transmission logic here
}
```

4. Power the Sensor Node:
- Power the Arduino using an appropriate power source.
- Set up a coordinator node to receive and display the wirelessly transmitted sensor data.

Part 2: Coordinator Node Setup

Materials:
- Arduino board (e.g., Arduino Uno)
- Zigbee module (e.g., XBee Series 2)
- Computer with Arduino IDE for monitoring

PROCEDURE:

1.Connect Zigbee Module:
  Connect the Zigbee module to the Arduino as explained in the sensor node setup.

2.Upload Coordinator Node Code:
  Write a code for the coordinator node to receive and display the transmitted data.

Here's a simple example:

```
void setup () {
 // Initialize Zigbee module
 // Setup Serial communication for Zigbee
 Serial. Begin (9600);
}

void loop () {
 if (Serial.available() > 0) {
  char data = Serial.read();
  Serial.print("Received: ");
  Serial.println(data);
 }
}
```

3.Connect to Computer:
  Connect the coordinator node Arduino to a computer using a USB cable.
4.Monitor Data:
  Open the Serial Monitor in the Arduino IDE to monitor the received data.
5.Power the Coordinator Node:
  Power the Arduino using the USB cable connected to the computer.

6. Verification:
- Power both the sensor node and the coordinator node.
- Observe the sensor node transmitting data every few seconds.
- Verify that the coordinator node receives and displays the transmitted data on the Serial Monitor.

Pre-Lab Questions:

1. What is Zigbee, and how does it compare to other wireless communication protocols like Wi-Fi and Bluetooth?
2. How does Zigbee handle addressing and data transmission between nodes?
3. What are the roles of the Coordinator, Router, and End Device in a Zigbee network?
4. Describe the interfacing requirements between a microcontroller and a Zigbee module.
5. Explain the power consumption characteristics of Zigbee and its suitability for IoT applications.

Post-Lab Questions:

1. Write down the steps involved in configuring and initializing the Zigbee module for data transmission.
2. What challenges did you face while interfacing Zigbee with the microcontroller, and how did you overcome them?
3. Compare the real-time performance of your Zigbee-based communication with other wireless alternatives.
4. What modifications can be made to improve the efficiency of the Zigbee-based sensor data transmission system?
5. Write any debugging techniques which is used when troubleshooting connectivity issues.

RESULTS:

Thus the set up of a sensor node successfully read data from a temperature sensor and transmit it wirelessly using Zigbee.