Aim: Configuration of XBee S2C and LoRa Devices to Create a Wireless Sensor Network (WSN)

Theory:

A **Wireless Sensor Network (WSN)** is a network of spatially distributed sensor nodes that communicate wirelessly to collect, transmit, and sometimes process environmental or physical data (like temperature, humidity, pressure, etc.). The primary goal is to monitor and report conditions to a central node or gateway for further analysis or action. In IoT applications, WSNs are essential for enabling communication in environments where wired infrastructure is not feasible.

Two popular wireless technologies used in building WSNs are **XBee (Zigbee protocol-based)** and **LoRa (Long Range Radio)**. Both offer unique advantages in terms of power consumption, range, and data rate, making them suitable for different types of IoT deployments.

XBee S2C (Zigbee Protocol)

XBee modules (especially the S2C variant) implement the **Zigbee protocol**, based on the IEEE 802.15.4 standard. Zigbee supports **mesh, star, and tree topologies**, which allows for flexible and self-healing networks. Each XBee module can be configured as one of the following:

- **Coordinator**: Initializes and manages the network.
- **Router**: Passes data between nodes and extends coverage.
- **End Device**: Sends/receives data but relies on routers or the coordinator for routing.

Zigbee is known for:

- **Low power consumption**, suitable for battery-powered sensors
- **Short to medium range** (up to ~100 meters)
- **Moderate data rate** (~250 kbps)
- **Mesh support**, enhancing network reliability

XBee modules are configured using **XCTU software**, where parameters like PAN ID, device role, and destination addresses are set.

Steps: Configuring XBee S2C using XCTU:

1. **Connect XBee to PC** using a USB explorer or XBee USB adapter.
2. **Open XCTU** and add the device (via COM port).
3. **Update firmware** to latest Zigbee TH (Transparent or API mode).
4. **Assign PAN ID** – a unique network ID (e.g., 1234) to all devices.
5. **Set roles**:
    a. **Coordinator**:
        i. CE = 1 (Coordinator Enable)
        ii. AP = 1 (API mode or 0 for transparent)
    b. **Router/End Device**:
        i. CE = 0
        ii. AP = 1 or 0
6. **DH & DL** (Destination High/Low) should be set to the **SH & SL** (Serial High/Low) of the target device.
7. **Write settings** and disconnect. Repeat for each device.

Testing:

- Use serial monitors on both ends.
- Send data (e.g., sensor value) from one Arduino using Serial.print() and read it on the other.

LoRa (Long Range Radio)

LoRa is a proprietary wireless communication technology developed by Semtech that enables **long-range, low-power** data communication. Unlike Zigbee, LoRa typically operates in a **star topology**, where sensor nodes send data to a central gateway or receiver.

Key features of LoRa include:

- **Long range** (up to 10 km in open areas)
- **Very low power consumption**
- **Low data rate** (~0.3–50 kbps)
- **Excellent penetration** in urban and rural environments

LoRa uses **spread spectrum modulation**, which makes it resilient to interference and ideal for outdoor or industrial environments. It is well-suited for applications like:

- Remote weather stations
- Agricultural monitoring
- Smart city infrastructure

LoRa modules (e.g., SX1278 or RFM95) communicate via **SPI** with microcontrollers like Arduino or ESP32 and are programmed using libraries like LoRa.h.

Steps: LoRa Configuration using Arduino:

1. **Connect LoRa module to Arduino** via SPI:
    a. MISO → D12, MOSI → D11, SCK → D13, NSS → D10, RST → D9, DIO0 → D2
2. **Install LoRa library**:
    a. Use **Sandeep Mistry's LoRa library** in Arduino IDE.
3. **Sender Code Sample**
4. **Receiver Code Sample**