

Ex. No. 7**CONTROL YOUR HOME POWER OUTLET FROM ANYWHERE
USING RASPBERRY PI, ZIGBEE AND ARDUINO****Aim:**

Control a home power outlet remotely using Raspberry Pi, Zigbee, and Arduino.

Hardware Components:

- Raspberry Pi (with Raspbian OS)
- Zigbee module (e.g., XBee)
- Arduino (with Zigbee module)
- Power outlet (connected to a relay module for switching)
- Jumper wires
- Power supply for Raspberry Pi and Arduino
- Zigbee coordinator (e.g., another XBee module or a Zigbee USB dongle for Raspberry Pi)

Procedure:**1.Setup Zigbee Communication:**

- Connect the Zigbee module to the Arduino using jumper wires.
- Use the Arduino IDE to upload a program to the Arduino that establishes communication with the Zigbee module. This program should define functions to send and receive commands.

2.Connect Zigbee Coordinator to Raspberry Pi:

- If using a Zigbee USB dongle for Raspberry Pi, plug it into one of the USB ports.
- If using another Zigbee module as a coordinator, connect it to the Raspberry Pi using the appropriate interface (e.g., UART).

3.Configure Zigbee Network:

1. Set up a Zigbee network with the coordinator and end devices (Arduino modules). Ensure that they can communicate with each other.

4.Power Outlet Control:

1. Connect the relay module to the Arduino, ensuring proper wiring and safety precautions.
2. Write a program for the Arduino that interprets Zigbee commands received and controls the relay to turn the power outlet on/off.

5. Raspberry Pi Control:

1. Write a Python program on the Raspberry Pi using a Zigbee library (e.g., Zigpy) to send commands to the Zigbee network.
2. Implement a simple user interface or command-line interface on the Raspberry Pi for the user

to control the power outlet remotely.

6.Integration:

1. Establish communication between the Raspberry Pi and Arduino by defining a protocol for sending/receiving commands.
2. Test the setup by sending commands from the Raspberry Pi to the Arduino and observing the corresponding changes in the power outlet state.

Coding:

ARDUINO CODE:

```
#include <SoftwareSerial.h>
SoftwareSerial zigbeeSerial (2, 3); // RX, TX pins on Arduino
void setup () {
  zigbeeSerial.begin (9600);
  pinMode (4, OUTPUT); // Relay control pin
}
void loop () {
  if (zigbeeSerial.available() > 0) {
    char command = zigbeeSerial.read();

    if (command == '1') {
      digitalWrite (4, HIGH); // Turn on the relay
    } else if (command == '0') {
      digitalWrite (4, LOW); // Turn off the relay
    }
  }
}
```

RASPBERRY PI (PYTHON) CODE:

```
import zigpy
from zigpy.quirks import Bus
from zigpy.zdo import types
from zigpy.zcl.clusters.general import OnOff
```

Initialize Zigbee communication

```
zigbee = zigpy. application. Controller (application=None, auto_form=True)
zigbee. startup(auto_form=True)
bus = Bus(zigbee)
```

Find and bind to the power outlet device (replace 0x1234 with actual device IEEE address)

```
power_outlet = zigbee.get_device(0x1234)
```

Function to control the power outlet

```
def control_power_outlet(state):
    power_outlet. on off. send (OnOff. cluster_id, state)
```

Example: Turn on the power outlet

```
control_power_outlet (1)
```

Example: Turn off the power outlet

```
control_power_outlet (0)
```

Pre-Lab Questions:

1. How can wireless communication be utilized to control electrical devices remotely?
2. What factors influence the choice of a communication protocol for home automation?
3. Why is it important to use both Raspberry Pi and Arduino in an IoT-based control system?
4. How does a microcontroller interact with a wireless module to transmit control signals?
5. What role does Zigbee play in home automation applications?

Post-Lab Questions:

1. In what way did the communication between Zigbee and Raspberry Pi affect overall system performance?
2. How can this system be modified to support multiple power outlets instead of a single one?
3. What observations can be made regarding the latency of the control signals in real-world usage?
4. How does the system handle connectivity issues, and what improvements can be suggested?
5. How could artificial intelligence be incorporated into this setup to enhance automation?

RESULTS:

The integration of Raspberry Pi, Zigbee, and Arduino enables efficient remote control of a home power outlet, providing a versatile and practical smart home automation.