Practical 1

Q1 To write a program to sense the available networks using Arduino.

Here's how you can set up a basic simulation for WiFi scanning with an ESP8266 or ESP32 in Proteus, assuming you have the appropriate Proteus library installed for the module.

Requirements

1. ESP8266 or ESP32 model library in Proteus.

2. Proteus 8 Professional or later.

3. Arduino IDE to write and compile the code.

Steps to Simulate WiFi Network Scanning Using ESP8266/ESP32 in Proteus

1. Install ESP8266/ESP32 Library in Proteus

- If you haven't already, you'll need to download the ESP8266 or ESP32 library for Proteus. These are available from online sources like [The Engineering Projects](https://www.theengineeringprojects.com/).

- After downloading, add the `.LIB` and `.IDX` files to your Proteus library folder (`C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\LIBRARY`).

2. Create the Circuit in Proteus

- Open Proteus and create a new project.

- Add the ESP8266 or ESP32 module to your schematic.

- Connect VCC and GND to a 3.3V power source.

- Add a virtual terminal to display the serial output, which will show detected networks.

- Connect the TX pin of the ESP8266/ESP32 to the RX pin of the virtual terminal, and RX to TX.

3. Write the Arduino Code

Here's the Arduino code to scan for available WiFi networks:

```
#include <ESP8266WiFi.h>  // Use <WiFi.h> if you're using ESP32


void setup() {
  Serial.begin(115200);          // Initialize Serial Monitor at baud rate of 115200
  WiFi.mode(WIFI_STA);           // Set WiFi to station mode to scan networks
  delay(100);                    // Short delay to stabilize


  Serial.println("Scanning for WiFi networks...");
}


void loop() {
  int n = WiFi.scanNetworks();    // Start WiFi scan


  Serial.println("Scan complete.");
  if (n == 0) {
    Serial.println("No networks found.");
  } else {
    Serial.print(n);
    Serial.println(" networks found:");
    for (int i = 0; i < n; ++i) {
      // Print SSID and RSSI (signal strength) of each network
      Serial.print(i + 1);
      Serial.print(": ");
      Serial.print(WiFi.SSID(i));
      Serial.print(" (");
      Serial.print(WiFi.RSSI(i));
      Serial.println(" dBm)");
      delay(10);
    }
  }
  delay(10000);  // Wait 10 seconds before the next scan
}
```

4. Compile and Upload the Code

1. Open the Arduino IDE.

2. Set up the ESP8266 or ESP32 board by going to Tools > Board and selecting the correct board (e.g., Generic ESP8266 Module or ESP32 Dev Module).

3. Paste the code above into the Arduino IDE and compile it.

4. The Arduino IDE will generate a `.hex` file for the program. You can find this file by enabling Show verbose output during: compilation under File > Preferences.

5. Load the `.hex` File in Proteus

1. In Proteus, double-click on the ESP8266 or ESP32 module to open its properties.

2. Load the `.hex` file generated by the Arduino IDE into the Program File field.

3. Click OK to apply the changes.

6. Run the Simulation

1. Start the simulation in Proteus.

2. Open the Virtual Terminal to view the serial output.

3. The output should show messages indicating the scan process, the number of networks found, and details about each network, such as its SSID (network name) and RSSI (signal strength).

Explanation of the Code

- WiFi.mode(WIFI_STA): Sets the ESP8266/ESP32 module to station mode, allowing it to search for nearby WiFi networks.

- WiFi.scanNetworks(): Initiates a scan for available networks and returns the number of networks found.

- WiFi.SSID(i): Retrieves the SSID (name) of the ith network.

- WiFi.RSSI(i): Retrieves the RSSI (signal strength) of the ith network.