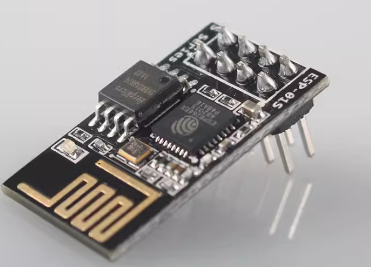
# Arduino UNO R4 Minima with EPS-01S WiFi Module for Wireless Communication

## Description

In this project, we will learn how to interface an Arduino Uno R4 Minima with an EPS-01S WiFi module to enable wireless communication over a WiFi network. We'll utilize the EPS-01S module to connect to a WiFi network and transfer information between the Arduino and other devices connected to the same network. This project is useful for applications such as remote monitoring, IoT (Internet of Things) projects, and wireless data transmission.



## How-To Guide

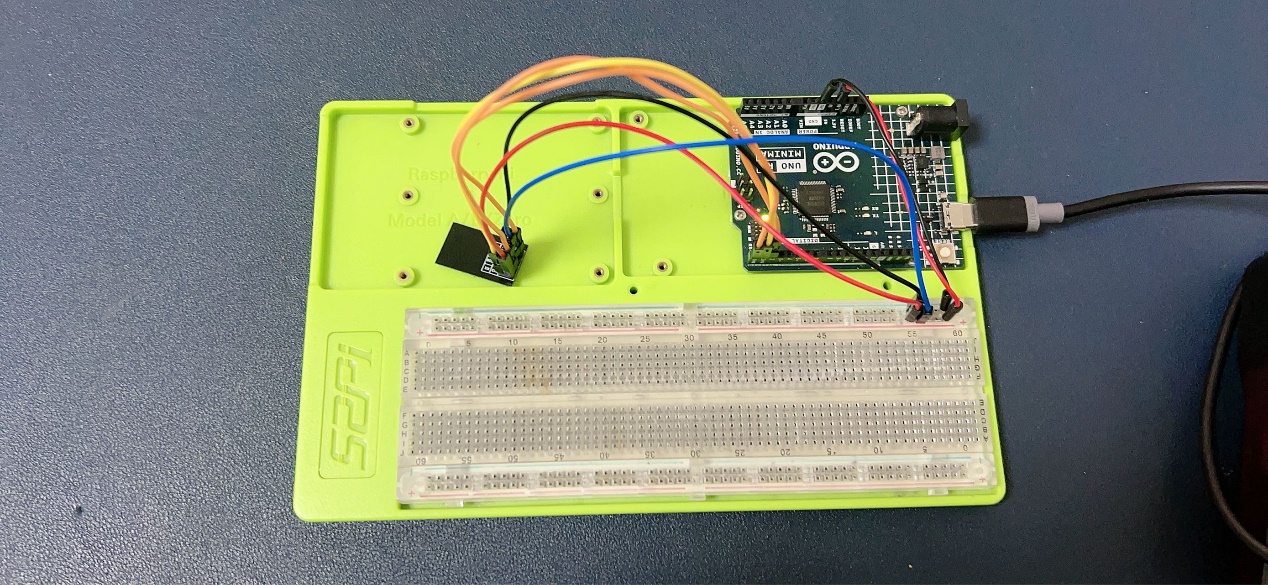
### 1. Gather Components

* 1 x Arduino Uno R4 Minima
* 1 x Breadboard
* 1 x EPS-01S WiFi Module
* 10 x Jumper wires

### 2. Assemble the Circuit

* Connect the EPS-01S WiFi module to the Arduino Uno R4 Minima using jumper wires.
* Ensure proper connections between the EPS-01S module and the Arduino Uno R4 Minima.

|  |  |
| --- | --- |
| Arduino UNO R4 Minima | ESP-01S Module |
| 3V3 | 3V3 |
| 3V3 | EN |
| D0 | TX |
| D1 | RX |
| GND | GND |
| D2 | RST |



### 3. Write the Arduino Code

Open the Arduino IDE and create a new sketch. Then, use the following code:

void setup()

{

    Serial.begin(115200);

    Serial1.begin(115200);

    pinMode(2 /\*ESP RST\*/, OUTPUT);

    digitalWrite(2, LOW);

    delay(50);

    digitalWrite(2, HIGH);

    delay(50);

    while (Serial1.find("OK"))

    {

    }

    // 等模块就绪

    while (true)

    {

        delay(100);

        if (Serial1.available() > 0)

            if (Serial1.find("AT"))

                break;

        Serial1.println("AT");

    }

    Serial1.println("AT+CWMODE=1");

    delay(100);

    Serial1.println("AT+CWJAP=\"YOUR\_AP\_SSID\",\"YOUR\_AP\_PASSWORD\"");

    while (true)

    {

        delay(100);

        if (Serial1.available() > 0)

            if (Serial1.find("GOT IP"))

                break;

    }

    // fix busy p... due to poor network ... conn to ap wait too long

    delay(2000);

    Serial1.println("AT+CIPSTART=\"TCP\",\"192.168.3.8\",44655");

// Please change the IP address to your local sokit server’s IP address, and the 44655 is the port, change it if you like.

    while (true)

    {

        delay(100);

        if (Serial1.available() > 0)

            if (Serial1.find("OK"))

                break;

    }

    // 100% ok

    delay(100);

    Serial1.println("AT+CIPMODE=1");

    delay(100);

    Serial1.println("AT+CIPSEND"); // start to send

}

void loop()

{

    int sensorValue = 0;

    sensorValue = analogRead(A0); //read sensor, here we let the A0 empty, it will auto detect A0’s value, in real life you can connect the raindrop sensor, soil moisture sensor ‘s AO to it.

    delay(1000);

    Serial.println("go");

    Serial1.print("ADC=");

    Serial1.print(sensorValue);

}

### 4. Understand the Code

This code is for setting up and communicating with an ESP-01S module via the serial interface (UART) on an Arduino Uno R4 Minima. Let's break down each part of the **setup()** and **loop()** functions:

#### setup()

##### 1. Serial Communication Initialization

**Serial.begin(115200);**: Initializes serial communication with a baud rate of 115200 bits per second for communication with the Arduino IDE's serial monitor.

**Serial1.begin(115200);**: Initializes serial communication with another device (in this case, the ESP-01S module) connected to the Arduino's hardware serial port (TX1/RX1) at a baud rate of 115200 bits per second.

##### 2. ESP-01S Module Initialization

* pinMode(2 /\*ESP RST\*/, OUTPUT);: Sets pin 2 (ESP RST) as an output pin.
* digitalWrite(2, LOW);: Sets pin 2 (ESP RST) to a low state.
* delay(50);: Waits for 50 milliseconds.
* digitalWrite(2, HIGH);: Sets pin 2 (ESP RST) to a high state.
* delay(50);: Waits for another 50 milliseconds.

##### 3. Wait for ESP-01S Module Ready

The code waits until the ESP-01S module responds with "OK" over the serial interface.

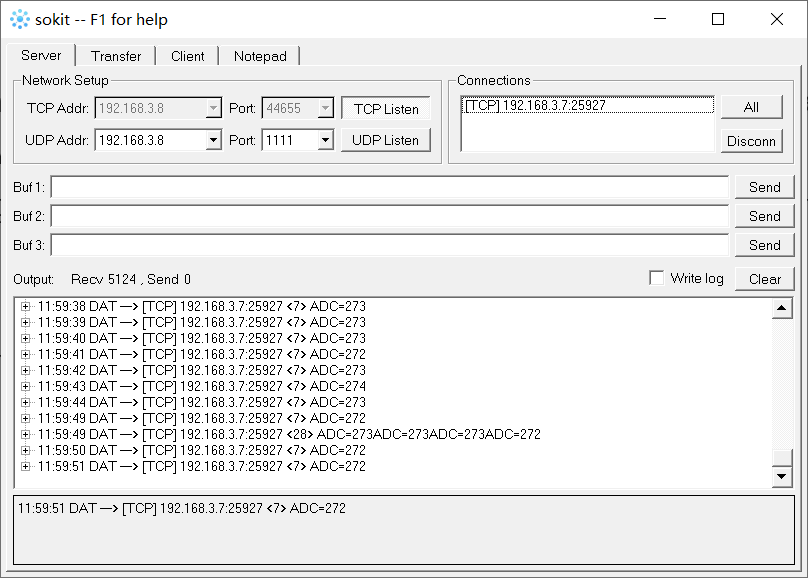
This is done in a loop using **Serial1.find("OK").**

##### 4. Configure ESP-01S Module

* Sets the ESP-01S module to station mode (**AT+CWMODE=1**).
* Connects to a specified Wi-Fi network with the SSID "YOUR\_NETWORK\_SSID" and password "YOUR\_NETWORK\_PASSWORD" (AT+CWJAP=" YOUR\_NETWORK\_SSID "," YOUR\_NETWORK\_PASSWORD ").
* Waits until the ESP-01S module obtains an IP address ("GOT IP").

##### 5. Establish TCP Connection

* Establishes a TCP connection to a remote server with the IP address "192.168.3.8" and port 44655 (**AT+CIPSTART="TCP","192.168.3.8",44655**).
* Please change the IP address to your local TCP server, you can create it by using sokit tool.



* Waits until the connection is successfully established.

##### 6. Configure TCP Connection

* Sets the TCP connection mode to transparent mode (**AT+CIPMODE=1**).
* Sends the "**AT+CIPSEND**" command to indicate that data will be sent.

#### loop()

##### 1. Read Sensor Data

* Reads the analog value from pin **A0**, which presumably is connected to a sensor.
* Stores the sensor value in the variable **sensorValue**.
* Send Sensor Data:
* Prints "**ADC=**" followed by the sensor value to the ESP-01S module via **Serial1**.
* This data will likely be sent over the established TCP connection to the remote server.

##### 3. Delay

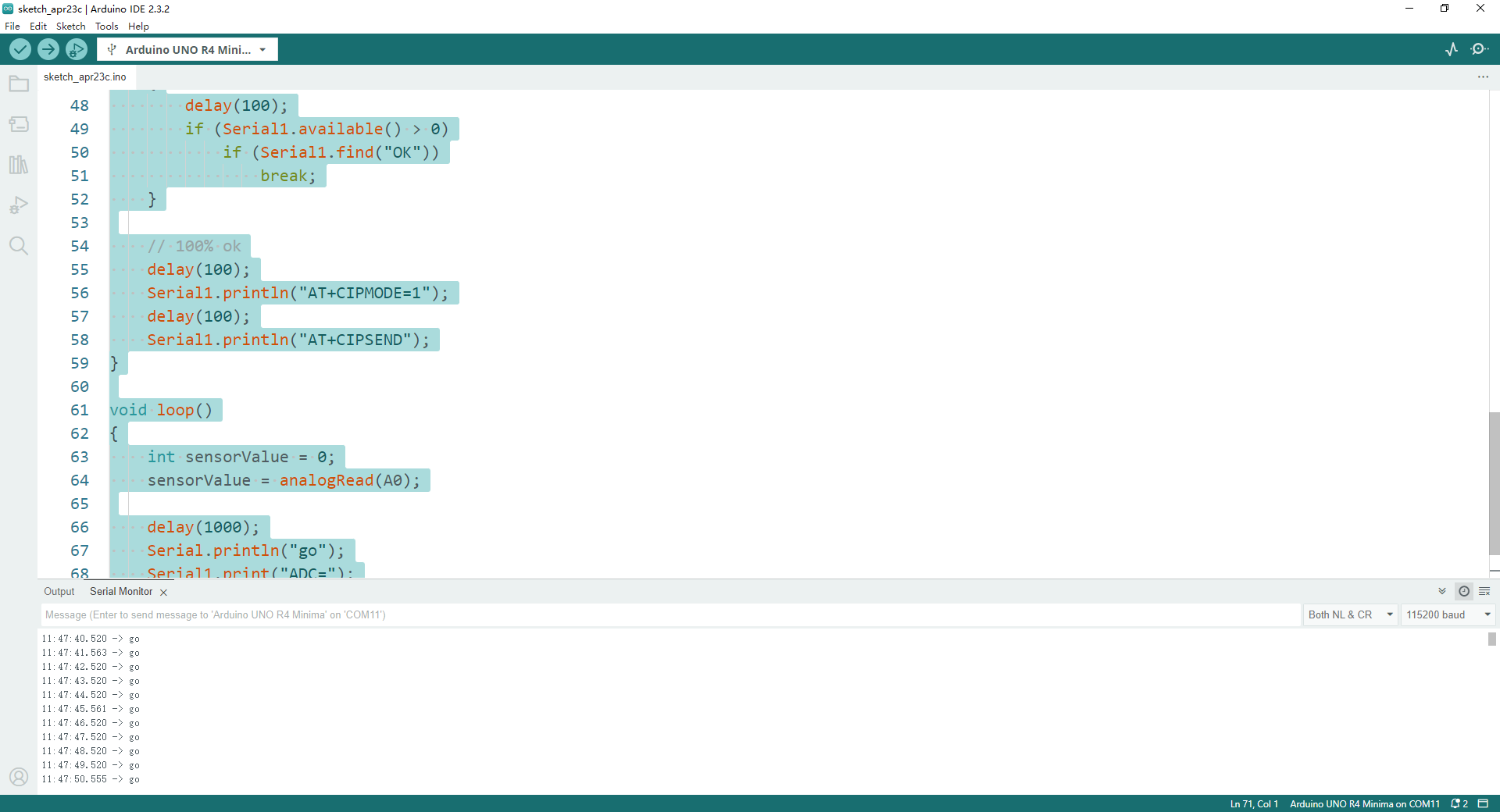
* Pauses execution for 1000 milliseconds (1 second) before repeating the loop.

#### Summary

This code initializes communication with the ESP-01S module, connects it to a Wi-Fi network, establishes a TCP connection to a remote server, reads sensor data, and sends it over the TCP connection. It continuously loops, reading and sending sensor data at regular intervals.

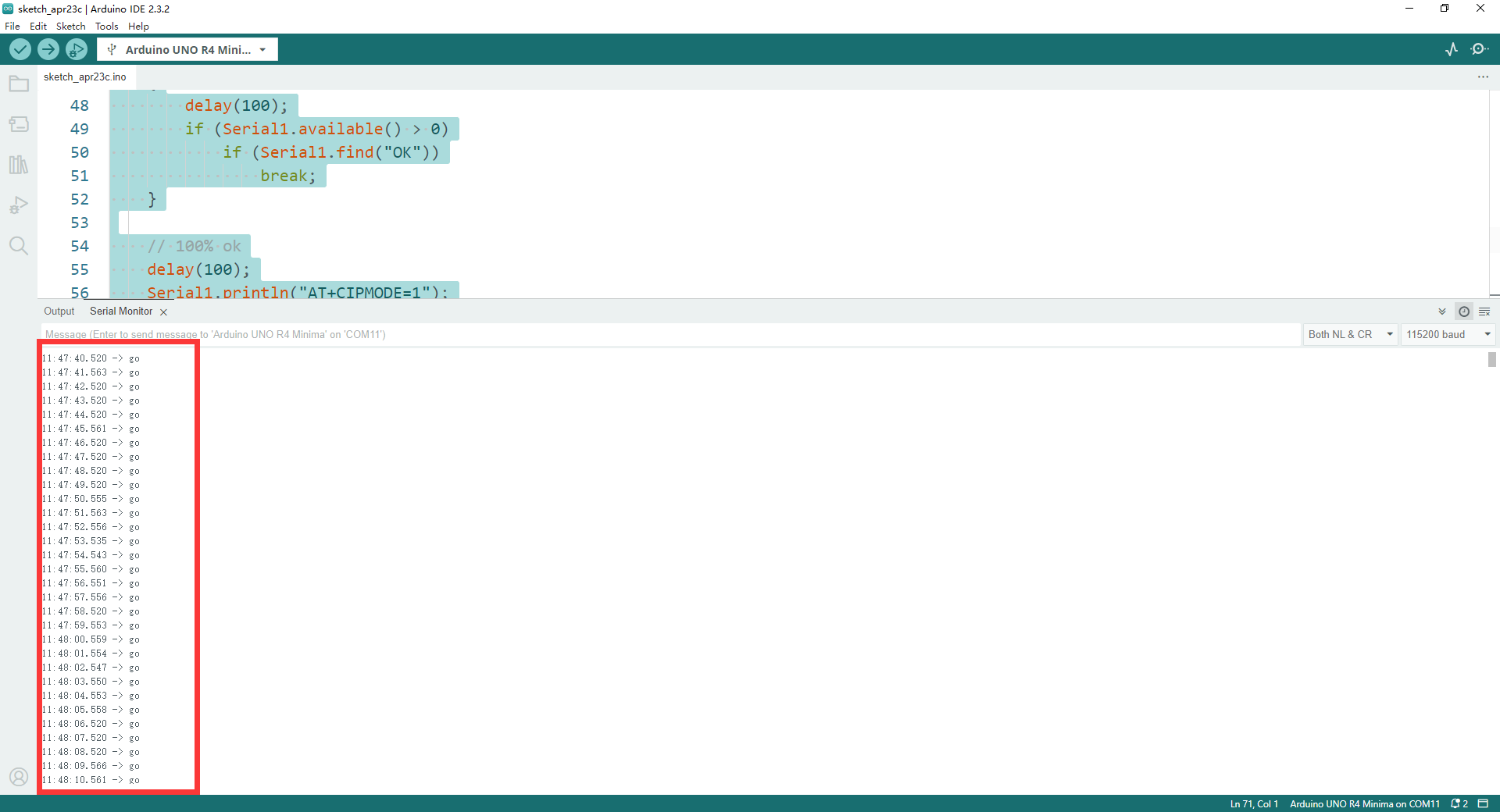
### 5. Test the Project

Upload the code to your Arduino Uno R4 Minima.



Open the serial monitor and ensure that the baud rate is set to 9600 bauds.

Monitor the serial output to verify the connection status with the WiFi network.



### 6. Experiment and Learn

Modify the code to include additional functionality such as sending sensor data or receiving commands over WiFi.

Explore advanced AT commands supported by the EPS-01S module for more advanced features.

Implement error-checking mechanisms to handle cases where the module fails to connect to the WiFi network.

By following these steps, you can successfully interface an EPS-01S WiFi module with the Arduino Uno R4 Minima to enable wireless communication over a WiFi network, facilitating various IoT and remote monitoring applications.

### 7. AT Commands

* ESP-01S Client

AT+CWMODE=1 // Enable STA mode

AT+CWJAP\_DEF=”your\_ssid”,”your\_ap\_password” // Connect to WiFi

AT+CIPSTART=”TCP”,”server\_ip\_address”,server\_port // Connect to Server

AT+CIPSEND // Start to transmit

* ESP-01S Server

AT+CWMODE=1 // Enable STA mode

AT+CWJAP\_DEF=”your\_ssid”,”your\_ap\_password” // Connect to WiFi

AT+CIPMUX=1 // Enable Multiple connection

AT+CIPSERVER=1,5000 // Enable server

AT+CIFSR // Check MAC address and IP address

AT+CIPSEND // Enable transmission

* ESP-01S AP Mode

AT+CWMODE=2 // Enable AP mode

AT+CWSAP=”ESP01S”,”123456”,11,3 // Enable ESP01S as WiFi, SSID name is ESP01S, password is 123456.

AT+CIPSEND // Enable transmission