# Lab 9: ESP8266 WiFi Module Projects

COEN-4720 Embedded Systems

*Cris Ababei*

*Dept. of Electrical and Computer Engineering, Marquette University*

## 1. Objective

The objective of this lab is to use an ESP8266 WiFi module connected to the Nucleo board to control the green LED wirelessly via a smartphone (Android) app or via a computer program on a host PC.

*NOTES:*

- *This is a more challenging lab. Please allocate enough time to work on it as early as possible!*
- *The lab assumes that you are familiar with Android app development; however, you can replace the Android portion with Apple iOS app development or just a computer program (in your programming language of choice) on your laptop.*

## 2. ESP8266 WiFi Module

The ESP8266 module [1] is a TTL "Serial to Wireless Internet" device (can be used as a UART-at-9600-baudrate to WiFi adapter). It is a very basic microcontroller unit with built in WiFi and can connect to 802.11b/g/n networks on the 2.4GHz band. The ESP8266 behaves like any other WiFi client. If your router can handle 128 laptops or 128 cell phones, it can also handle 128 ESP8266s.

While simple, it is an impressive, low-cost WiFi module suitable for adding WiFi functionality to an existing microcontroller project via a UART serial connection. The module can even be reprogrammed to act as a standalone WiFi connected device - just add power! The feature list is impressive and includes:

- 802.11 b/g/n protocol
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack

This module requires a 3.3V supply for VCC, and 3.3V logic. It doesn't have 5V to 3.3V logic level shifting. It is not 5V tolerant. Do not connect for example RX or TX of a "5V Arduino" to it as that may destroy this module. You should use a logic level converter (such as this one [2]) or a "3.3V Arduino". The 3.3V supply on the "Arduino Uno" for example has inadequate current capability to power this module. You must provide a separate, higher 3.3V supply (about 300mA or more). Nonetheless, we could still wire the ESP8266 to an Arduino in several different ways [3]. Though, using a level converter and a different power supply for the module would be safer.

When connecting it to the Nucleo board, make sure you use 3V3 as power supply for the module.

## 3. Example 1: Control LED on Nucleo board via ESP8266 WiFi Module

In this example, we send with an Android app the controls "1" or "2" to the Nucleo board. This is done via WiFi. Control messages are received by the Nucleo board via the ESP8266 module that is connected to the board. The complete project of this example is in the files provided with this lab. A simplified block diagram is shown in Figure 1 below.

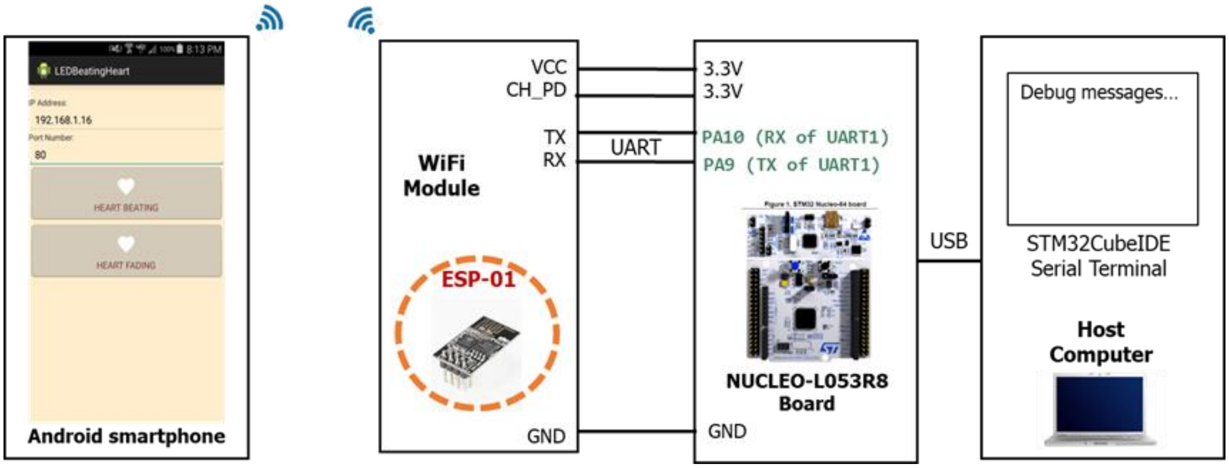


Figure 1: Block diagram of Example 1.

You should create a new project from scratch for this example. When creating the project and getting to the CubeMX step, enable: the SPI1 (to connect the LCD display), UART1 (to connect the WiFi module), the GPIOs for D/C, CS, and RST pins of the LCD display, and the Clock to use HSI. All these have been done in the labs before, so you should know how to do it at this time.

The connections to the WiFi module will be as follows (UART1 connections are as they were for the Bluetooth module):

**Nucleo Board -> WiFi Module**

| | |
|---|---|
| PA9 (TX of UART1) | -> RX of WiFi module |
| PA10 (RX of UART1) | -> TX of WiFi module |
| GND from Nucleo | -> GND of WiFi module |
| 3.3V from Nucleo | -> VCC of WiFi module |
| 3.3V from Nucleo | -> **CH_PD** of WiFi module |

For reference, Figure 2 shows the connections to the ESP8266 module.

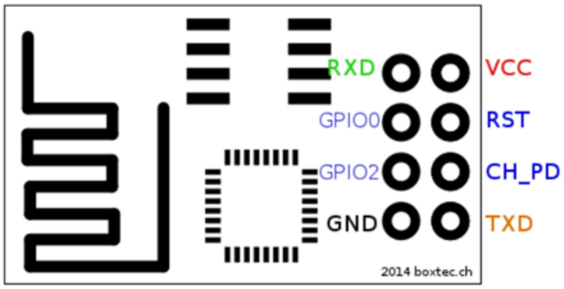


Figure 2: Connections to ESP8266 WiFi module.

After you wire up everything, the experimental setup should look like that in Figure 3.

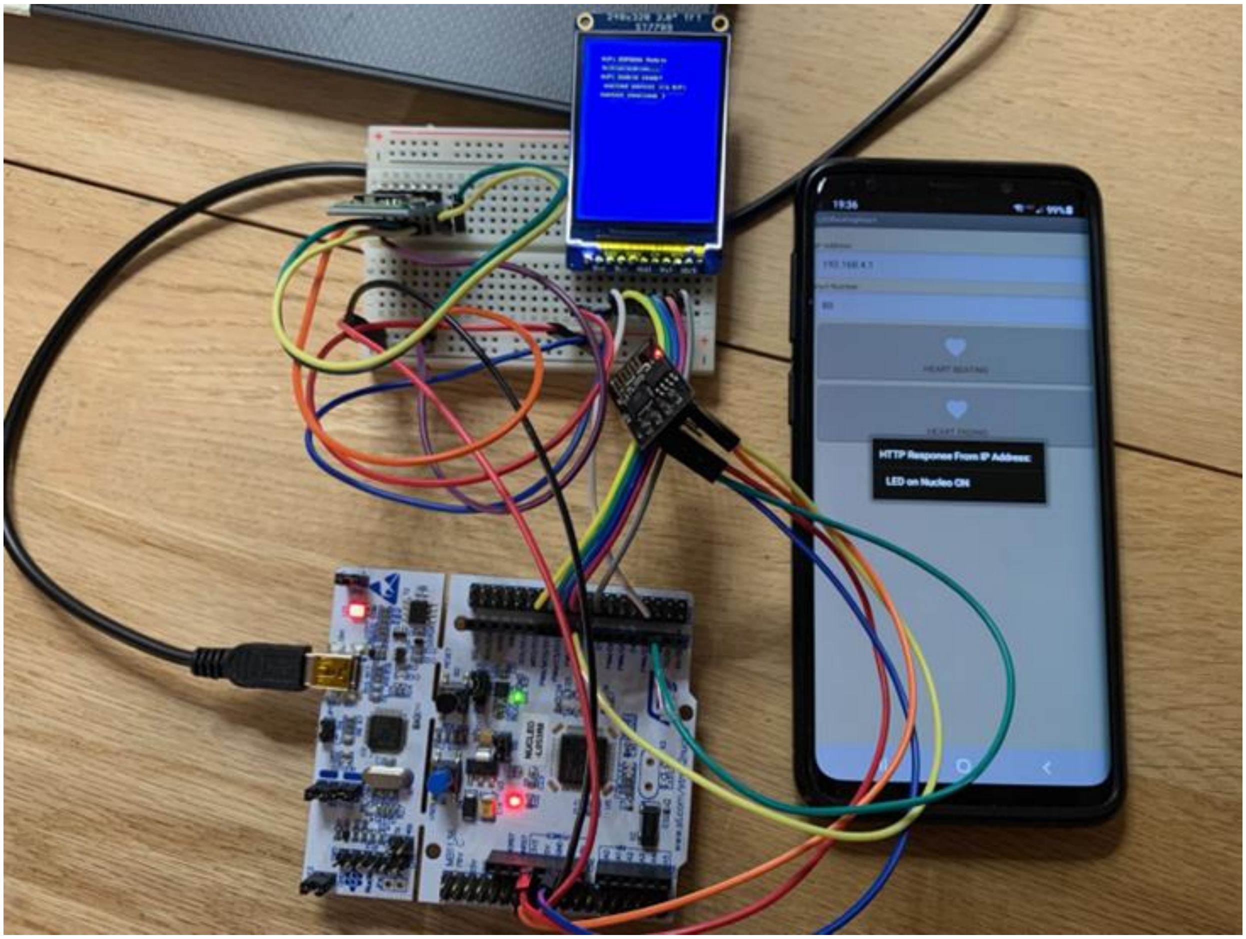
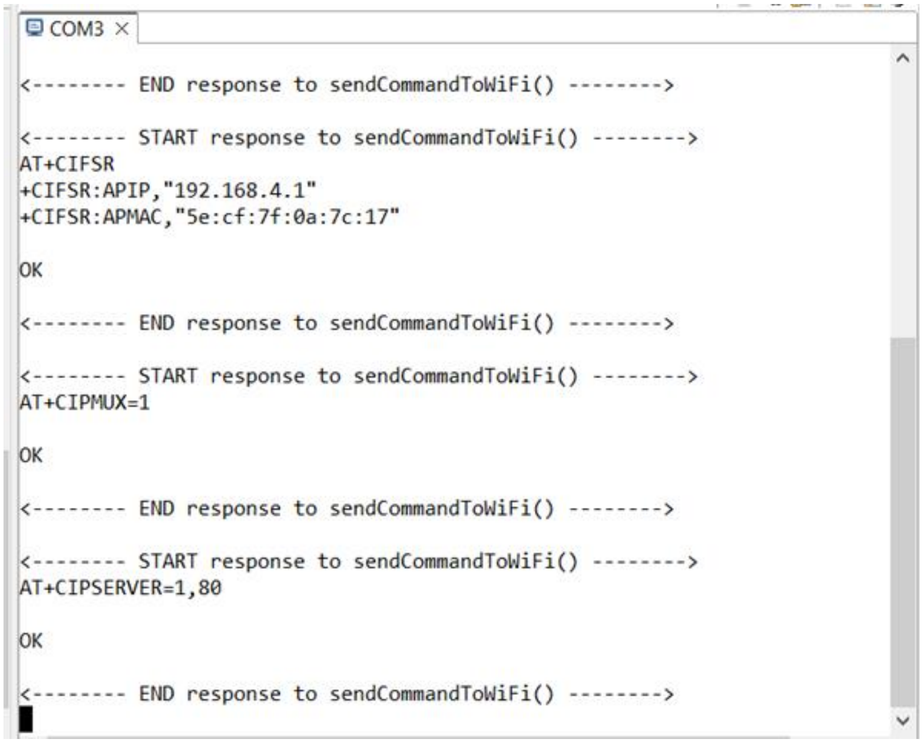


Figure 3: Experimental setup for WiFI lab.

During operation debug messages are sent via UART2 to the serial terminal on the host PC. For example, the last portion of the messages printed in the serial terminal of the CubeIDE when the function:

```
wifi_esp8266_init();
```

is run inside main() is shown in Figure 4.



```
<-------- END response to sendCommandToWiFi() -------->

<-------- START response to sendCommandToWiFi() -------->
AT+CIFSR
+CIFSR:APIP,"192.168.4.1"
+CIFSR:APMAC,"5e:cf:7f:0a:7c:17"

OK

<-------- END response to sendCommandToWiFi() -------->

<-------- START response to sendCommandToWiFi() -------->
AT+CIPMUX=1

OK

<-------- END response to sendCommandToWiFi() -------->

<-------- START response to sendCommandToWiFi() -------->
AT+CIPSERVER=1,80

OK

<-------- END response to sendCommandToWiFi() -------->
```

Figure 4: Debug messages are printed in the Serial terminal of the STM32CubeIDE.

At this time, you should take time to read all the code of this example and familiarize yourself with it. You should also read the implementation of the Android app, if you plan to use it. This example will be demonstrated in class.

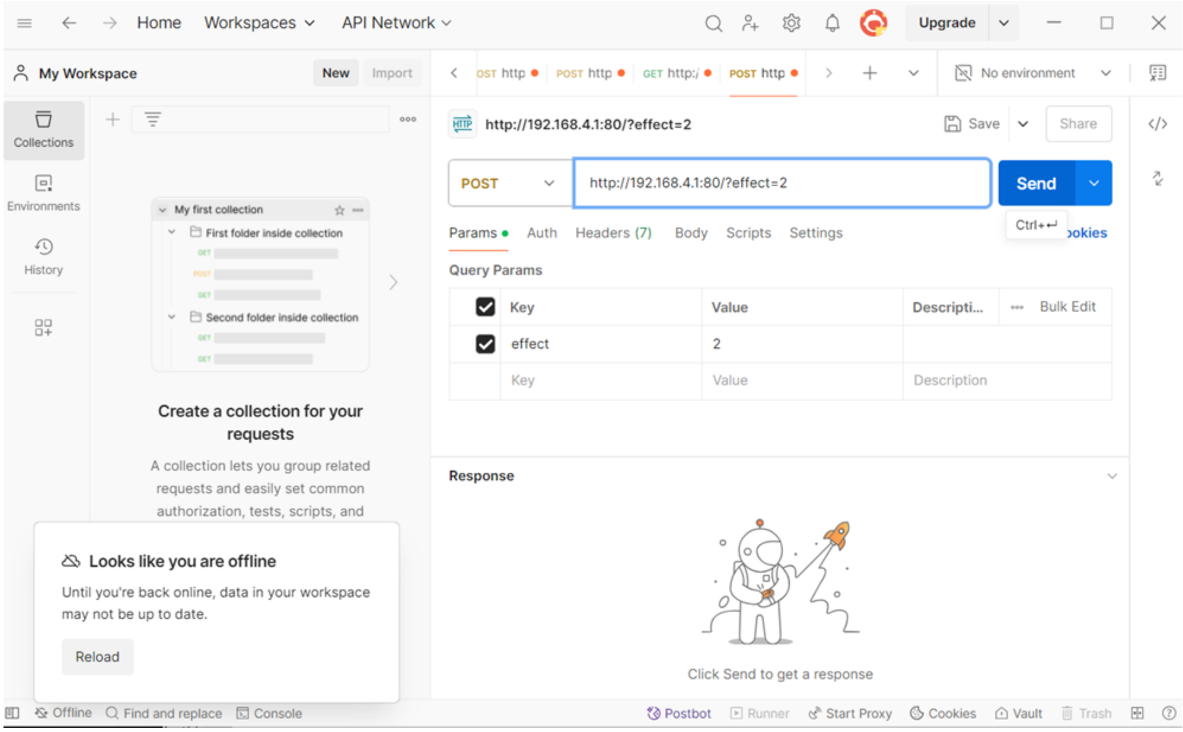
NOTE: This example is an adaptation for the Nucleo board of an Arduino project, which basically uses the WiFi module connected to the Arduino board to control an LED array also connected to the Arduino. It was demonstrated by Cris during the lecture. The control commands are sent to the Arduino from an Android smartphone. This project is described in detail and its source codes are publicly available here (including the complete source code of the Android app!): http://www.dejazzer.com/eece4920/index.html

## 4. Lab assignment

Modify the project from example 1 to send a third value (e.g., ”3”) for the control signal sent from the smartphone. When that value is received, the main program on your Nucleo board should change the color of the background of the LCD display to a randomly selected color from a set of eight different colors of your choice.

**NOTE:** Regarding the Android app: you are not required to use an Android app. You can use your own Apple app that you should develop on your own or you should implement your own computer program (in any programming language you want!). **Take this as an opportunity to learn smartphone app development!**

Alternatively, you can use any other existing smartphone or laptop application to send the control signal with three different control values to the ESP8266 module connected to the Nucleo board. You will have to search and find such apps on your own. An example of such an app to run on a PC is **postman** (https://www.getpostman.com/downloads/), which can be used as shown in Figure 5 below. Also, a webpage that presents a tutorial on postman is https://www.guru99.com/postman-tutorial.html#1



http://192.168.4.1:80/?effect=2

POST http://192.168.4.1:80/?effect=2

Query Params

| ☑ | Key | Value | Descripti... | ••• Bulk Edit |
|---|---|---|---|---|
| ☑ | effect | 2 | | |
| | Key | Value | Description | |

Response

Click Send to get a response

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and

Looks like you are offline

Until you're back online, data in your workspace may not be up to date.

Reload

Figure 5: Example of using postman application.

## 5. References

[1] ESP8266 WiFi Module at Adafruit, https://www.adafruit.com/products/2282
[2] SparkFun Logic Level Converter - Bi-Directional, https://www.sparkfun.com/products/12009
[3] ESP8266 wiring schemas, http://yaab-arduino.blogspot.co.uk/2015/03/esp8266-wiring-schemas.html