

Wi-Fi Remote Control Switch

Presented by

Ma May Kar Yar Lay

Mg Wai Yar Aung

21.4.2023

Introduction

- In this project , we use mobile application and wi-fi network to learn how to remotely control LED on/off .
- We can control not only in the wi-fi area but also far away from the host via cloud method.

Objectives

- To understand how to remotely control home appliances (refers to LED in this project) via wi-fi module fundamentally.
- To accomplish above objective, we are using ESP8266 wi-fi module and installing software application on smart phones as user interface to control LED on/off via a wi-fi network.

Wi-Fi

Wi-Fi is a wireless networking technology that uses radio waves to provide wireless high-speed Internet access and allows devices such as computers (laptops and desktops), mobile devices (smart phones and wearables), and other equipment (printers and video cameras) to interface with the Internet.

It allows these devices and many more to exchange information with one another, creating a network.



Components

Hardware

- Arduino UNO
- ESP8266 Wi-Fi Module
- LEDs
- Resistors
- Jumpers
- Breadboard
- Smart Phone

Software

- Remote XY

Arduino Uno

Microcontroller: ATmega328P

Operating Voltage: 5V

Input Voltage (recommended): 7-12V

Inout Voltage (limit): 6-20V

Digital I/O Pins: 14 (of which 6 provide PWM output)

PWM Digital I/O Pins: 6

Analog Input Pins: 6

DC Current per I/O Pin: 20 mA

DC current for 3.3V Pin: 50 mA

Flash Memory: 32 KB (ATmega328P) of which 0.5 KB used by bootloader

SRAM: 2 KB (ATmega328P)

EEPROM: 1 KB (ATmega328P)

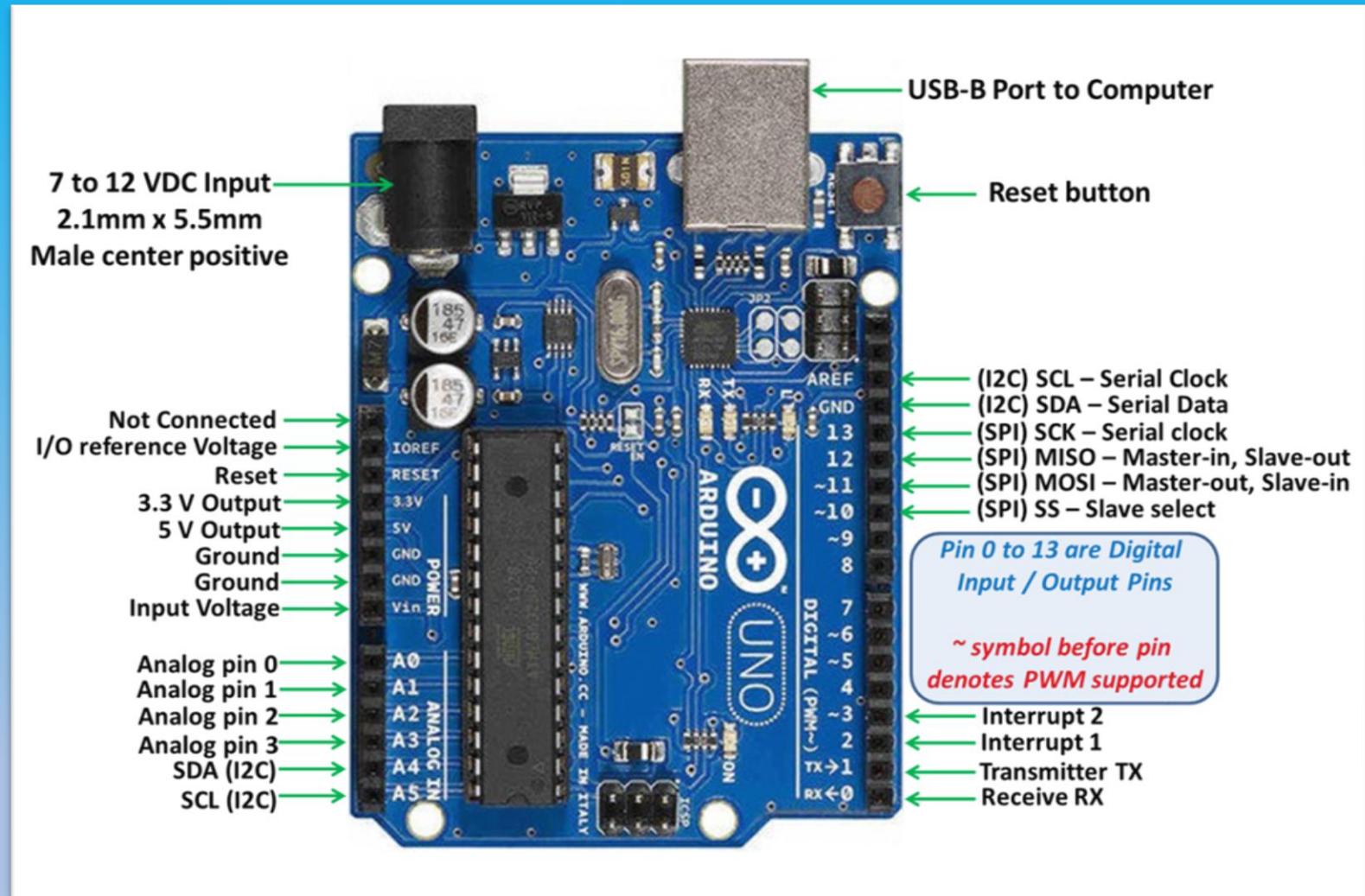
Clock Speed: 16 MHz

LED_BUILTIN: 13

Length: 68.6 mm

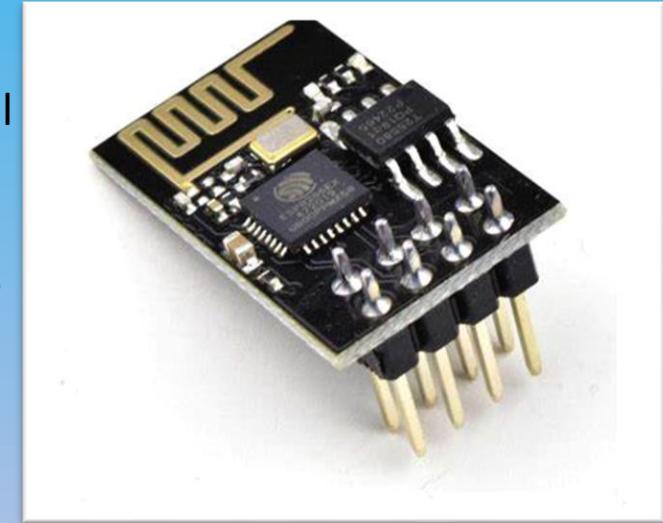
Width: 58.4 mm

Weight: 25 g



ESP8266-01 Wi-fi Module

- Wi-fi modules or wi-fi microcontrollers are used to send and receive data over Wi-Fi. They can also accept commands over the Wi-Fi. They are used for communications between devices. They are most commonly used in the field of Internet of Things.
- ESP8266 is the most widely used Wi-Fi module. It is a low-cost microchip with a full TCP/IP stack and microcontroller capability, produced by Espressif Systems.
- This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections.
- The layers of the TCP/IP model: network access, internet, transport, and application. Used together, these layers are a suite of protocols.
- It can also work in two modes: station mode, where it connects to an existing wi-fi hotspot, or access point mode, where it creates its own wi-fi network.



ESP8266-01 Wi-fi Module Specification

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2).
- General-purpose input/output (16 GPIO).
- Inter-Integrated Circuit (I²C) serial communication protocol.
- Analog-to-digital conversion (10-bit ADC).
- Serial Peripheral Interface (SPI) serial communication protocol.
- I²S (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO).
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2).
- Pulse-width modulation (PWM)

ESP8266-01 Wi-fi Module Pin Description

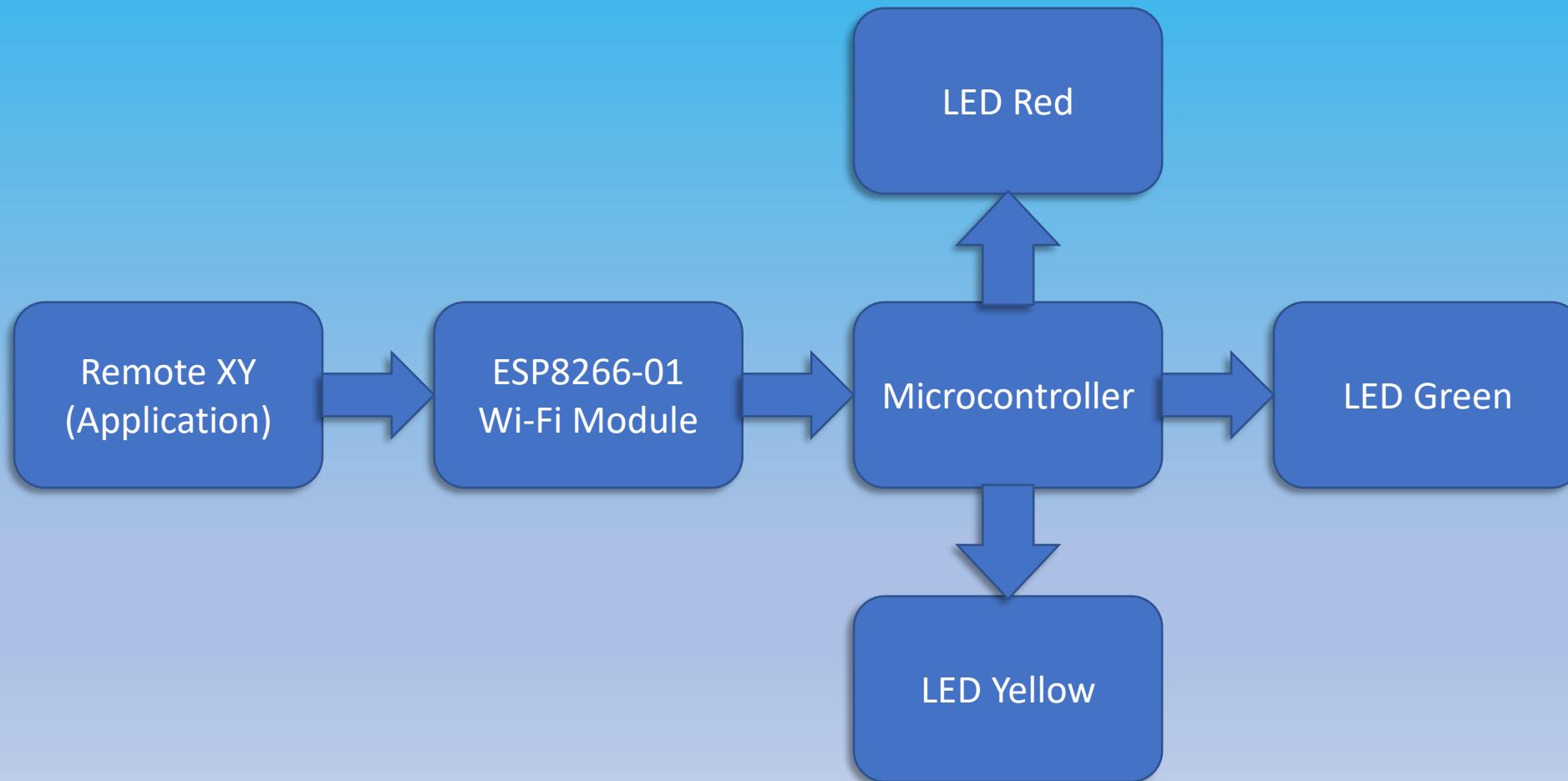


- VCC – Power Pin(+3.3V;can handle up to 3.6V)
- GND: – Ground Pin
- RST: – Reset Pin
- EN(CH_PD): – Enable Pin.
- TX: – Serial Transmit Pin of UART.
- RX: – Serial Receive Pin of UART.
- GPIO0 & GPIO2: – General Purpose I/O Pins. It also known as TX/RX pins are used for Programming the module or for serial I/O purpose.

ESP8266-01 Wi-fi Module Application

- (1) Internet of Things
- (2) Network consumer equipment
- (3) Measurement
- (4) Building automation
- (5) Home automation
- (6) Smart home gateway
- (7) Intelligent lighting
- (8) Smart Plugs and Lights
- (9) Baby Monitor
- (10) Sensor Network
- (11) Industrial Control

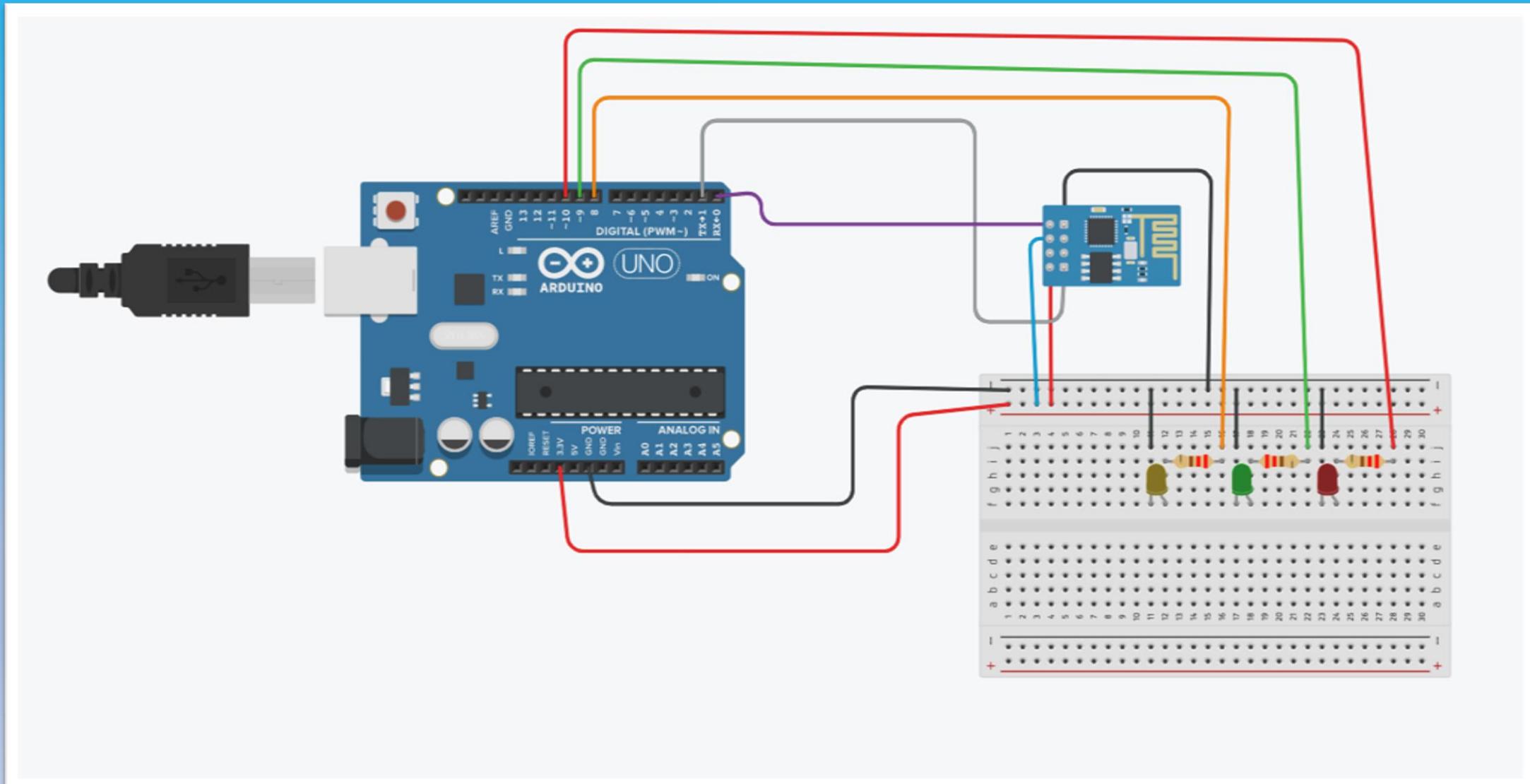
Block Diagram



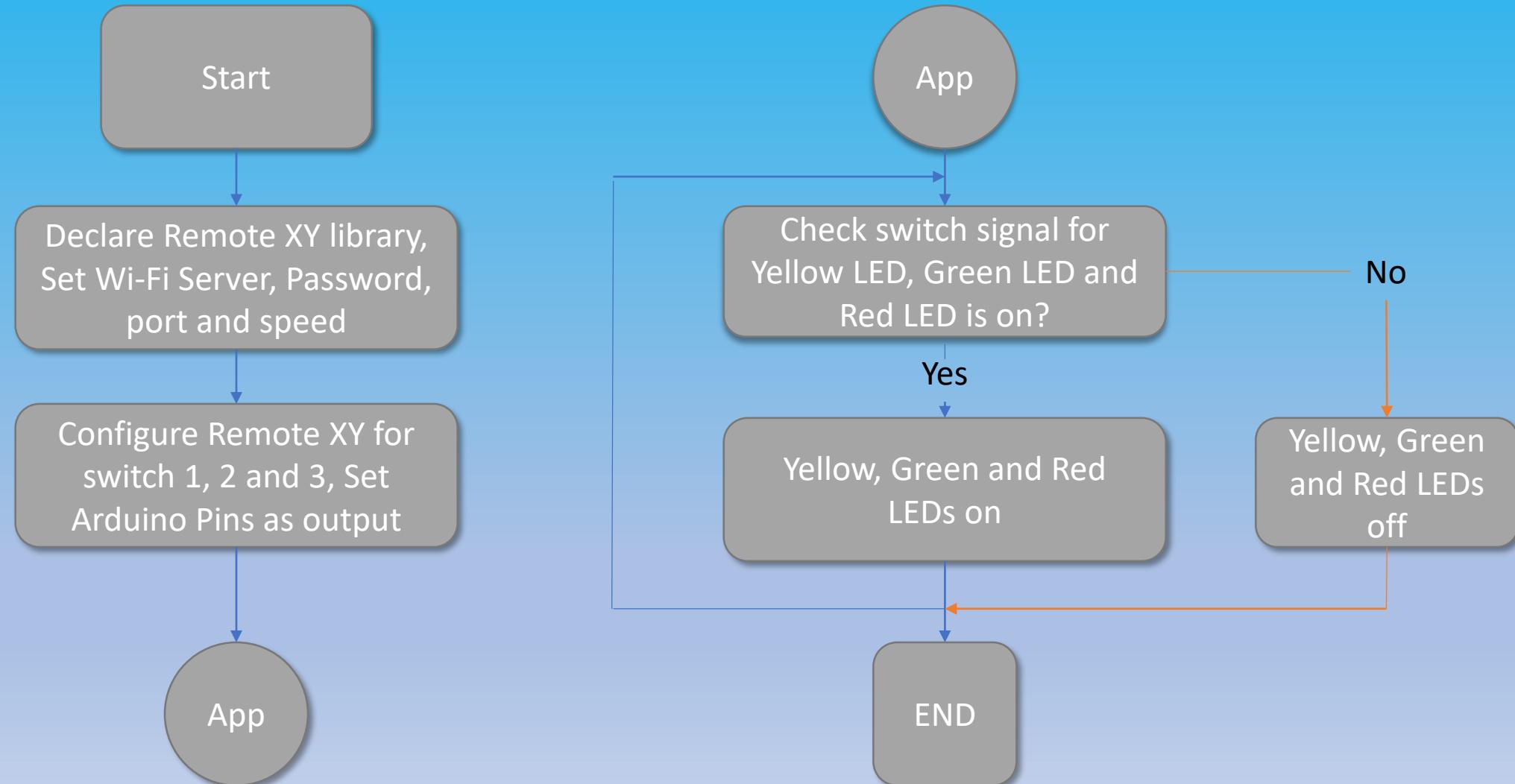
Pin Connection

- LED1 (yellow) → 8
 - LED2(green) → 9
 - LED3(red) → 10
 - Tx (ESP8266) → 0 (Rx: Arduino)
 - Rx (ESP8266) → 1 (Tx: Arduino)
 - VCC + EN → 3.3 V (Arduino)
 - GND → GND (Arduino)
- Disconnect the Tx and Rx pins of Wi-Fi module while uploading the sketch to Arduino.

Circuit Diagram



Flow Chart



Software Implementation

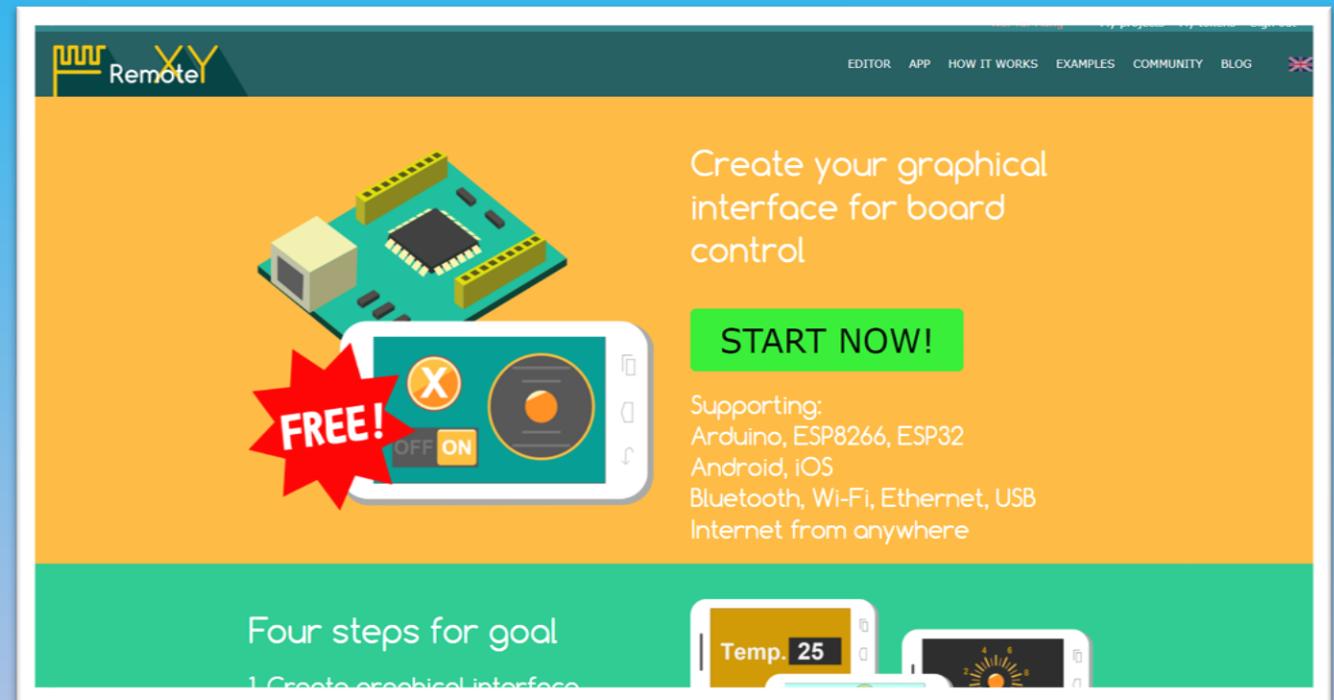
Wi-Fi Access Point

The software setting is shown step by step as follows.

1.Go to the website "Remote XY".

<https://remotexy.com>

2.Then click " start now".



3. You will see home page of Remote XY. Then, go to configuration.

The screenshot shows the Remote XY web application interface. At the top, there is a navigation bar with links for "My projects", "My tokens", "Sign out", and language selection ("en"). Below the navigation bar, there are tabs for "APP", "HOW IT WORKS", "EXAMPLES", "COMMUNITY", and "BLOG". On the far right, there is a search icon, a refresh icon, a user profile icon, and a plus sign icon.

The main area is titled "New project". On the left, there is a sidebar titled "Elements" with sections for "Controls", "Sensors", "Indication", "Decoration", and "Label".

- Controls:**
 - Button (orange circle)
 - Push switch (green circle with "ON" text)
 - Switch (grey rectangle with "OFF" and "ON" text)
 - Select (three circular buttons)
 - Slider (vertical orange bar)
 - Joystick (orange circle with two smaller circles)
- Sensors:**
 - RGB color (color wheel)
 - Edit field (text input with "edit" and "X" icons)
- Indication:**
 - None
- Decoration:**
 - Label (orange square)
 - Frame (orange square)
 - Page (two overlapping green squares)
- Label:**
 - Label (orange square)
 - Frame (orange square)
 - Page (two overlapping green squares)

In the center, there is a large smartphone-shaped canvas with a grid background. A green button labeled "Get source code" is located in the top right corner of the canvas area. The text "Drag-and-drop elements here from the left area" is displayed inside the smartphone shape.

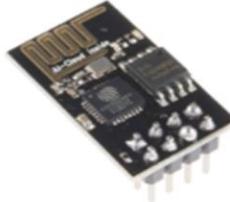
Select as follows. After that, click “Apply”.

Connection Board Module IDE

Wi-Fi Arduino UNO ESP8266 Wi-Fi module ARDUINO

Wi-Fi access point Arduino IDE

Compatible modules



ESP8266 Wi-Fi module

Available modules



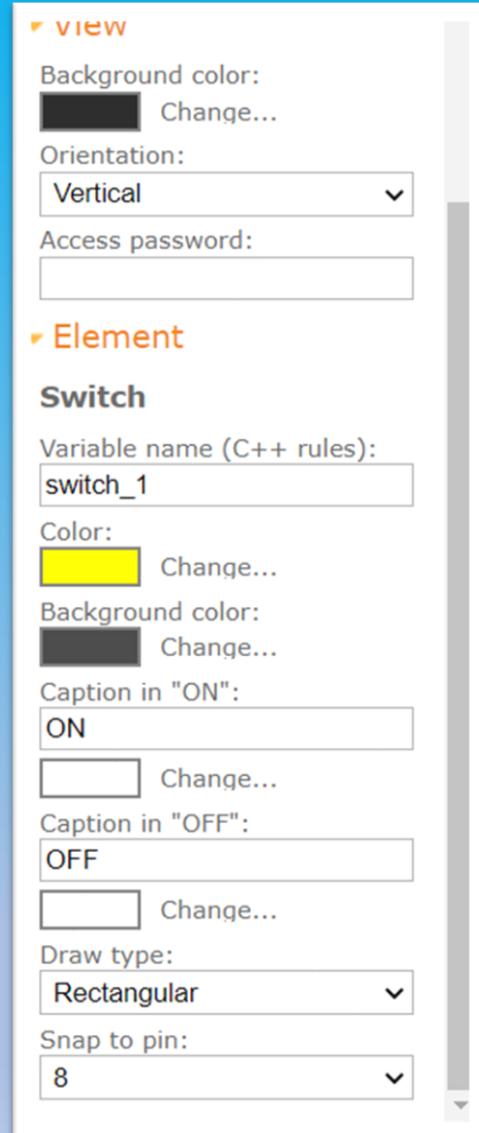
Apply Cancel



4.

In module interface, set serial port to 0(RX) and 1(TX), baud rate to 115200.

Name your wi-fi and set password. Port can be set as your preference.



5. In View, you can decorate. If you set access password, you will need to enter password when connecting to wi-fi on Remote XY app.

6. In Controls, take switch or push button. In this project, switch is used and name the elements, LED Yellow, LED Green and LED Red.

7. Click the switch and set pin in “Snap to pin”. In this project, set the pins according to the circuit diagram.

LED yellow is set to pin 8.

LED green is set to pin 9.

LED red is set to pin 10.

8. This is the final result. After that, don't forget to save the project. Then, click " Get source code".

The screenshot shows the RemoteXY mobile application interface. On the left, there's a sidebar titled "Elements" containing categories for Controls (Button, Push switch, Switch, Select, Slider, Joystick, RGB color, Edit field), Sensors (Compass, Accelerometer, Orientation), Indication, Decoration (Label, Frame, Page), and a "Wifi Acess Point" section with icons for file operations. In the center, a smartphone screen displays three sets of controls labeled "LED Yellow", "LED Green", and "LED Red". Each set includes a text label, a small image of a switch, and an "OFF" button followed by an "ON" button. On the right, there's a "Properties" panel with sections for Configuration (Wi-Fi access point, Arduino Uno, ESP8266 Wi-Fi module, Arduino IDE), Module interface (Connection interface: Hardware Serial, Serial port: Serial, pins 0(RX) and 1(TX), Speed (baud rate): 115200), and Wi-Fi access point settings (Name (SSID): RemoteXY, Open point checked, Password (8 or more chars): 12345678, Port:). A "Get source code" button is located at the top right of the central screen area.

- The Remote XY will generate arduino code automatically.
- You can download the code or just simply copy the code.
- Download library which needs to be installed before running the code.

Source code of project: Wifi Acess Point

- 1. Download the source code** of the program, open it in the Arduino IDE.
- 2. Install [RemoteXY library](#)** for Arduino IDE.
- 3. Compile the source code and upload it to the Arduino board using the Arduino IDE.**
- 4. Correctly connect the [ESP8266 Wi-Fi module](#) to the Arduino board. ESP8266 Firmware AT_v0.40 or up.**
- 5. Install the mobile app [RemoteXY ver.4.11.1](#) for smartphone/tablet.**
- 6. Connect to Arduino using mobile app.**

project.ino [Download code](#) [Download library](#)

```
/*
-- Wifi Acess Point --

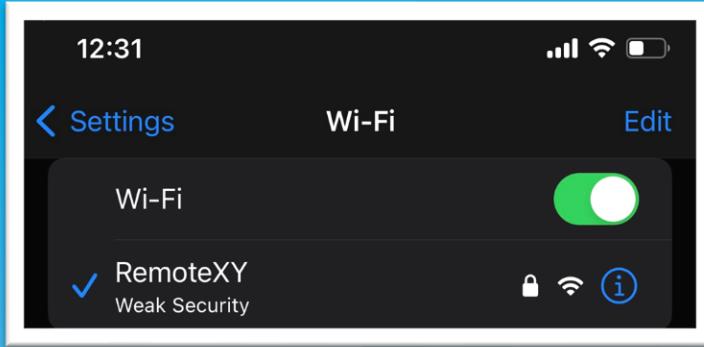
This source code of graphical user interface
has been generated automatically by RemoteXY editor.
To compile this code using RemoteXY library 3.1.8 or later version
download by link http://remotexy.com/en/library/
To connect using RemoteXY mobile app by link http://remotexy.com/en/download/
- for ANDROID 4.11.1 or later version;
- for iOS 1.9.1 or later version;
```

➤ Disconnect the Tx and Rx pins of Wi-Fi module while uploading the sketch to Arduino.

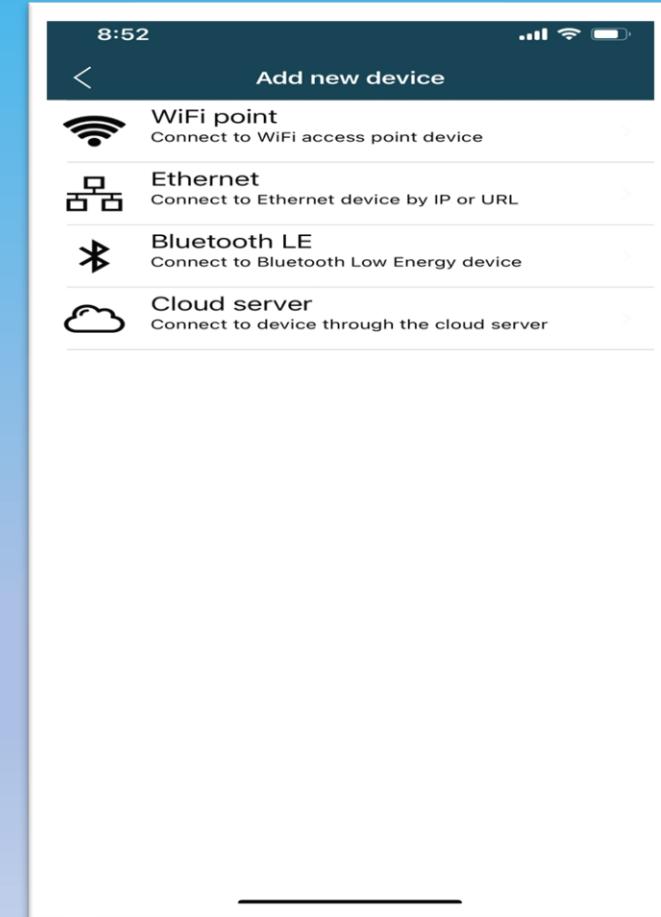
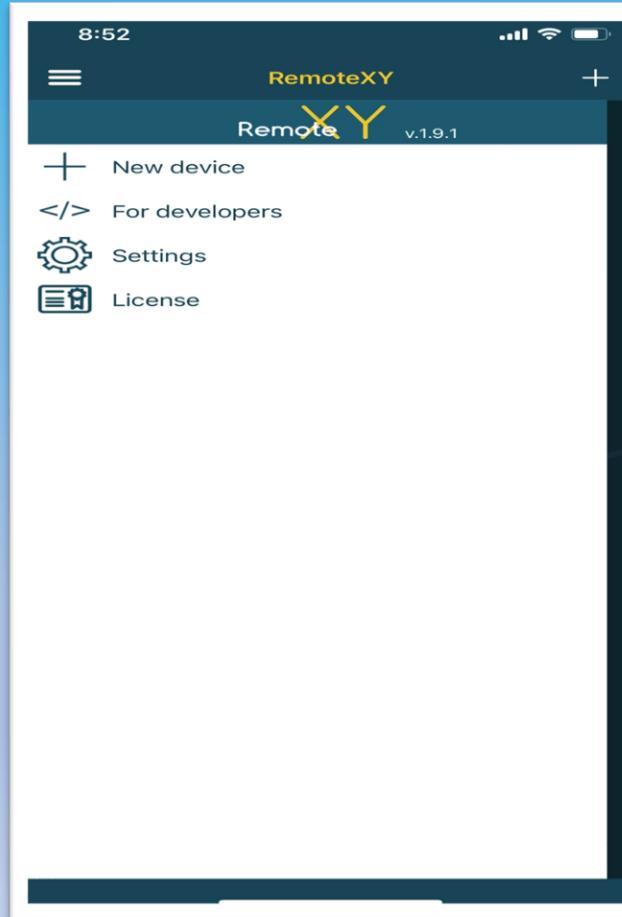
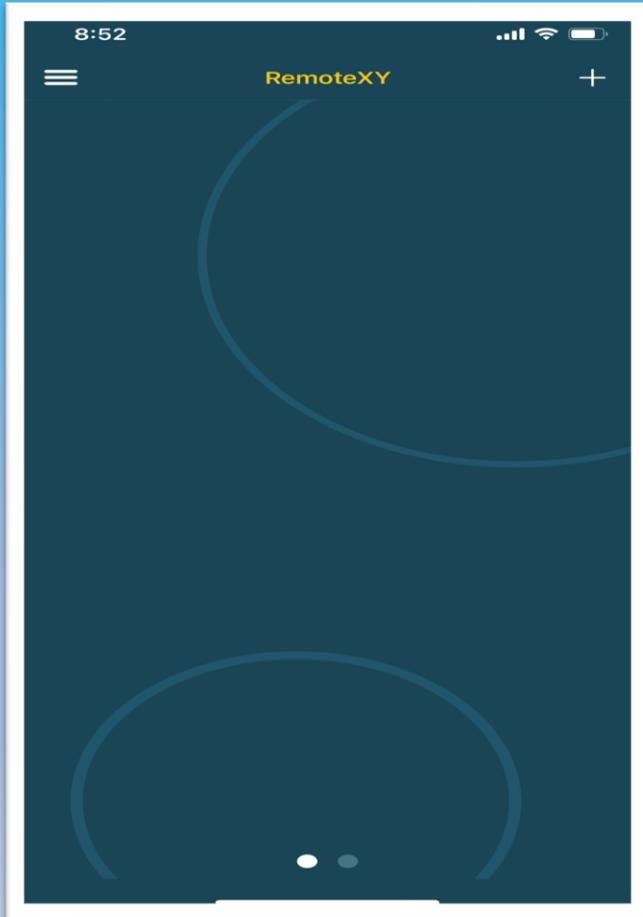
Download Remote XY: Arduino control app from Google play store or App store to your mobile phone.

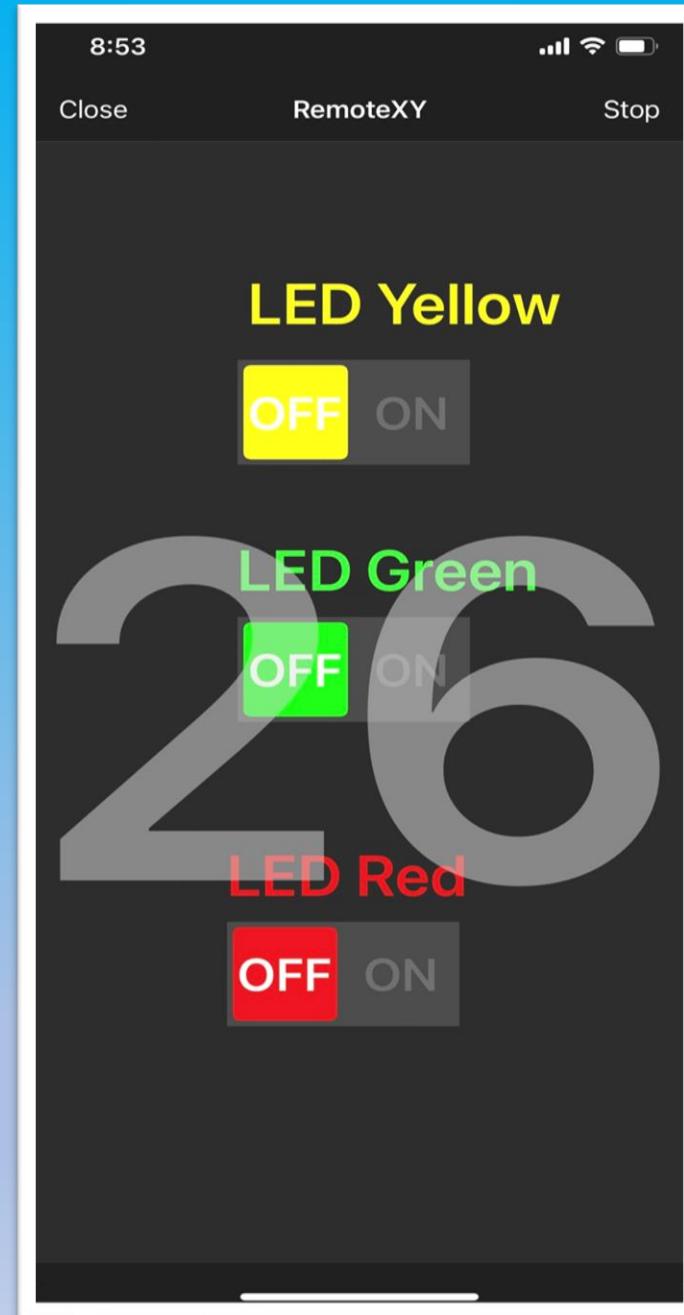
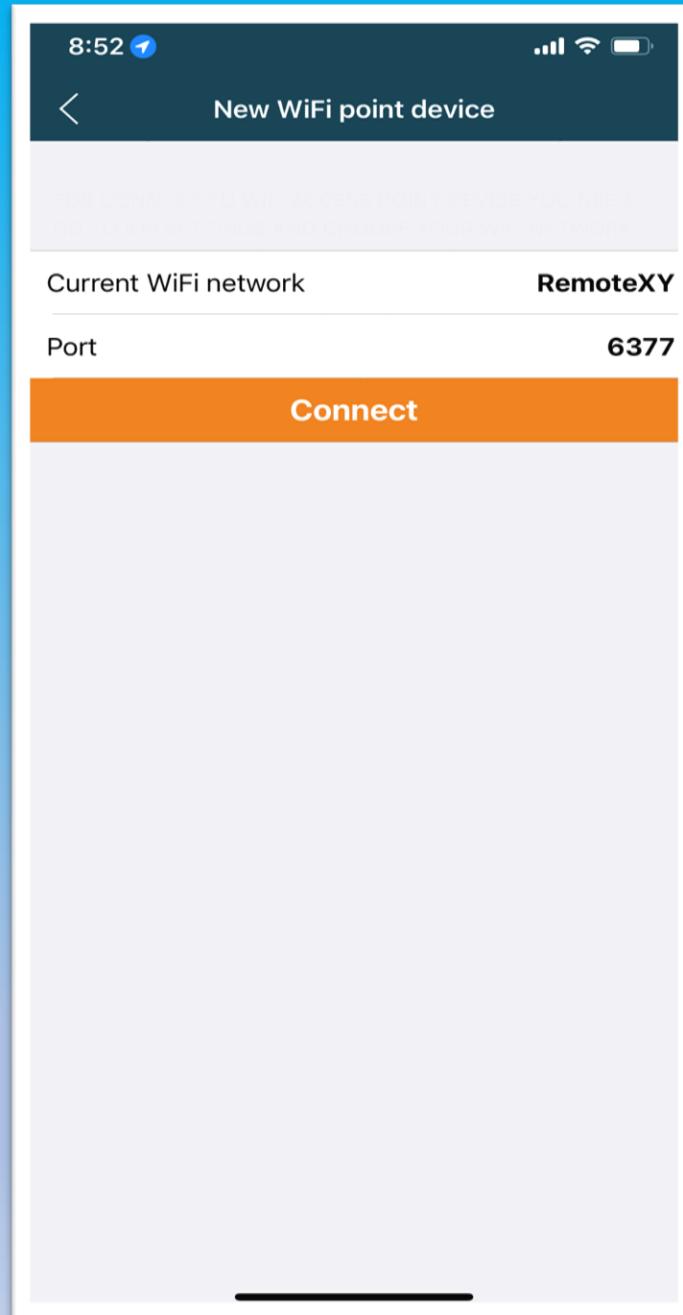
<https://apps.apple.com/app/id1168130280>

[https://play.google.com/store/apps/details?id=com.shevauto.r
emotexy.free](https://play.google.com/store/apps/details?id=com.shevauto.r emotexy.free)

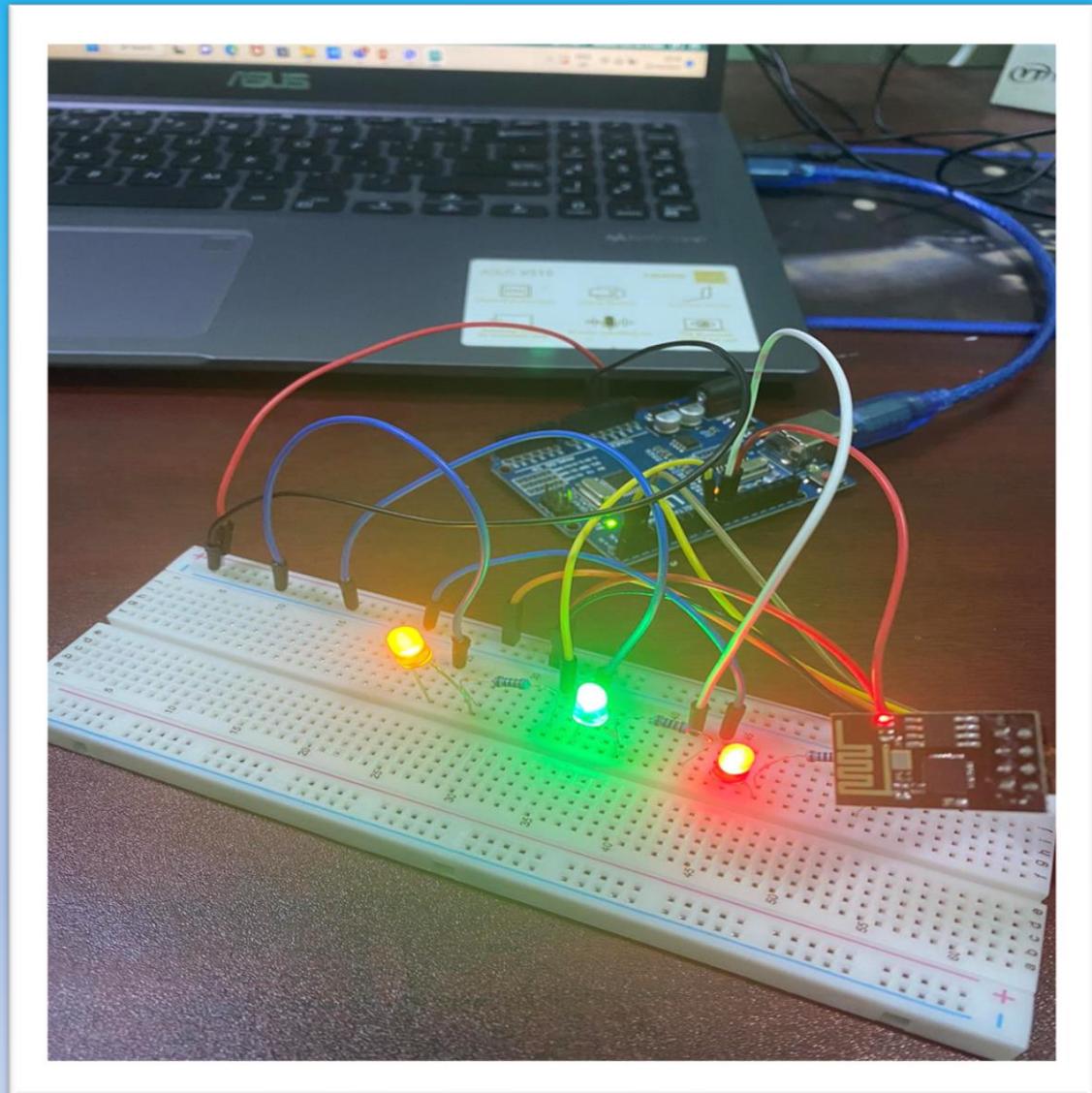


- Connect your phone to Remote XY in the Wi-Fi setting.
- Open your Remote XY app and add the new device.
- Add new device > Wi-Fi point > Remote XY > Enter password and connect.





Tests and Results



Program Sketch

Wifi_access_point.ino

```
1 #define REMOTEXY_MODE_ESP8266_HARDSERIAL_POINT
2
3 #include <RemoteXY.h>
4
5 // RemoteXY connection settings
6 #define REMOTEXY_SERIAL Serial
7 #define REMOTEXY_SERIAL_SPEED 115200
8 #define REMOTEXY_WIFI_SSID "RemoteXY"
9 #define REMOTEXY_WIFI_PASSWORD "12345678"
10 #define REMOTEXY_SERVER_PORT 6377
11
12 // RemoteXY configurate
13 #pragma pack(push, 1)
14 uint8_t RemoteXY_CONF[] = // 111 bytes
15 { 255,3,0,0,0,104,0,16,8,1,2,0,20,14,22,11,94,26,31,31,
16 | 79,78,0,79,70,70,0,2,0,20,41,22,11,135,26,31,31,79,78,0,
17 | 79,70,70,0,2,0,19,73,22,11,1,26,31,31,79,78,0,79,70,70,
18 | 0,129,0,19,65,24,6,1,76,69,68,32,82,101,100,0,129,0,20,33,
19 | 30,6,136,76,69,68,32,71,114,101,101,110,0,129,0,21,5,18,6,94,
20 | 76,69,68,32,89,101,108,108,111,119,0 };
21
22 // this structure defines all the variables and events of your control interface
23 struct {
24
25 | // input variables
26 | uint8_t switch_1; // =1 if switch ON and =0 if OFF
27 | uint8_t switch_2; // =1 if switch ON and =0 if OFF
28 | uint8_t switch_3; // =1 if switch ON and =0 if OFF
29 |
30 | // other variable
31 | uint8_t connect_flag; // =1 if wire connected, else =0
32
33 }
```

```
34 } RemoteXY;
35 #pragma pack(pop)
36
37 //////////////////////////////////////////////////////////////////
38 // END RemoteXY include //////////////////////////////////
39 //////////////////////////////////////////////////////////////////
40
41 #define PIN_SWITCH_1 8
42 #define PIN_SWITCH_2 9
43 #define PIN_SWITCH_3 10
44
45
46 void setup()
47 {
48     RemoteXY_Init ();
49
50     pinMode (PIN_SWITCH_1, OUTPUT);
51     pinMode (PIN_SWITCH_2, OUTPUT);
52     pinMode (PIN_SWITCH_3, OUTPUT);
53
54     // TODO you setup code
55
56 }
57
58 void loop()
59 {
60     RemoteXY_Handler ();
61
62     digitalWrite(PIN_SWITCH_1, (RemoteXY.switch_1==0)?LOW:HIGH);
63     digitalWrite(PIN_SWITCH_2, (RemoteXY.switch_2==0)?LOW:HIGH);
64     digitalWrite(PIN_SWITCH_3, (RemoteXY.switch_3==0)?LOW:HIGH);
65
66     // TODO you loop code
67 }
```

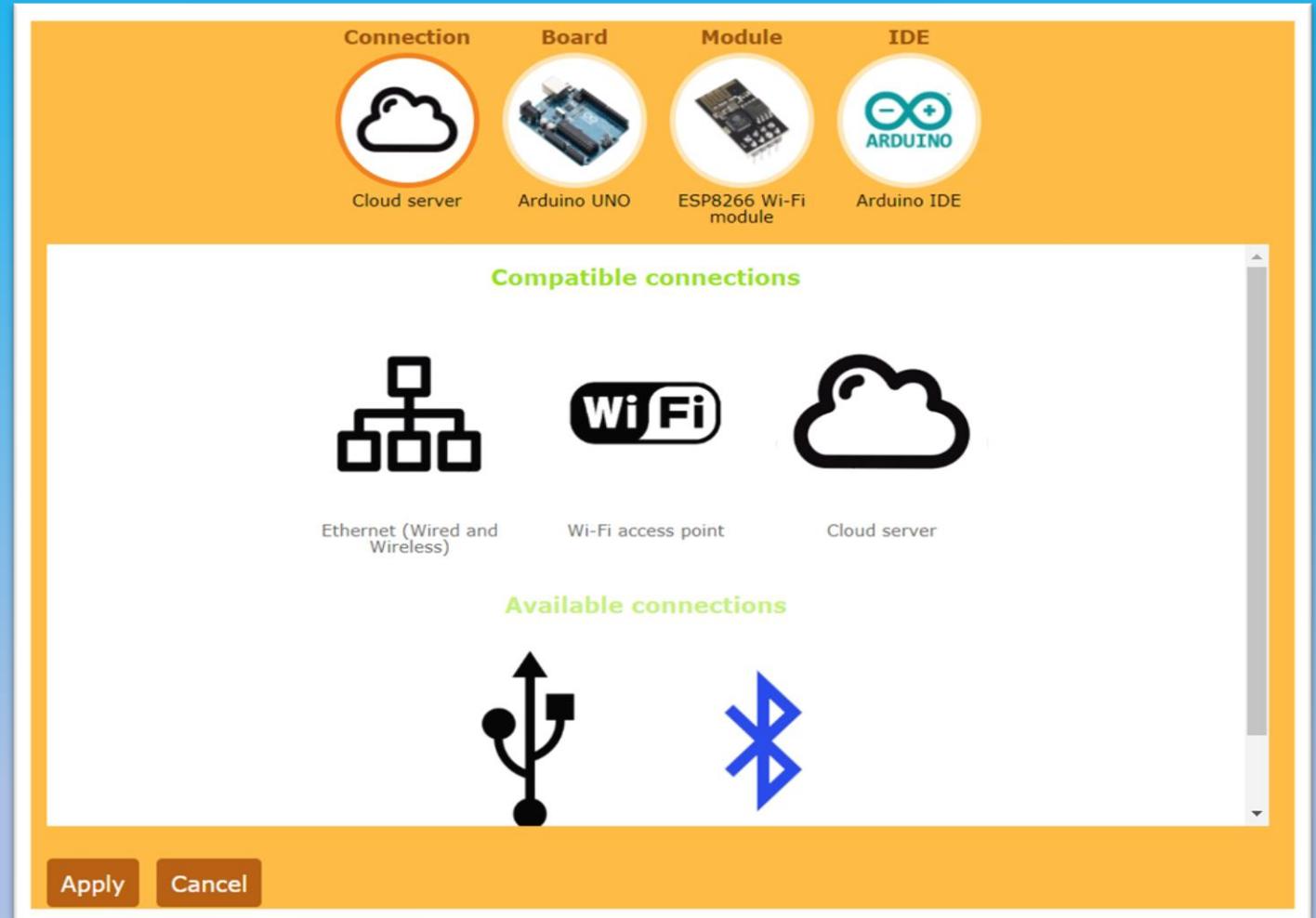
```
39 //////////////////////////////////////////////////////////////////
40
41 #define PIN_SWITCH_1 8
42 #define PIN_SWITCH_2 9
43 #define PIN_SWITCH_3 10
44
45
46 void setup()
47 {
48     RemoteXY_Init ();
49
50     pinMode (PIN_SWITCH_1, OUTPUT);
51     pinMode (PIN_SWITCH_2, OUTPUT);
52     pinMode (PIN_SWITCH_3, OUTPUT);
53
54     // TODO you setup code
55
56 }
57
58 void loop()
59 {
60     RemoteXY_Handler ();
61
62     digitalWrite(PIN_SWITCH_1, (RemoteXY.switch_1==0)?LOW:HIGH);
63     digitalWrite(PIN_SWITCH_2, (RemoteXY.switch_2==0)?LOW:HIGH);
64     digitalWrite(PIN_SWITCH_3, (RemoteXY.switch_3==0)?LOW:HIGH);
65
66     // TODO you loop code
67     // use the RemoteXY structure for data transfer
68     // do not call delay()
69
70
71 }
```

Cloud Server

For cloud server,

Step 1 and 2 are the same as above.

Then, choose cloud server of “connection” section in step 3.



4.

In module interface, set serial port to 0(RX) and 1(TX), baud rate to 115200. Name your wi-fi and set password you would like to connect. Then, create a token to build a cloud server.

Module interface

Connection interface:
Hardware Serial

Serial port:
Serial, pins 0(RX) and 1(TX)

Speed (baud rate):
115200

Wi-Fi connection:
Name (SSID):
WYA
Password:
25692511

Cloud server:
Token:
maykaryanlay

My tokens

Server:
cloud.remotexy.com
Port:
6376
Token:
f2fdxfc5ab449478aeebf532a1

View

Properties

Connection interface:
Hardware Serial

Serial port:
Serial, pins 0(RX) and 1(TX)

Speed (baud rate):
115200

Wi-Fi connection:
Name (SSID):
WYA
Password:
25692511

Cloud server:
Token:
maykaryanlay

My tokens

Server:
cloud.remotexy.com
Port:
6376
Token:
f2fdxfc5ab449478aeebf532a1

View

Background color: Change...
Orientation:
Vertical
Access password:
1234

Sign in

Email:
maykaryanlay1@gmail.com

Password:
.....

Forgot password?

Sign in Create account

Create new token Close

Board name:
maykaryanlay

Cloud server:
cloud.remotexy.com

Create Cancel

5. Copy the token.

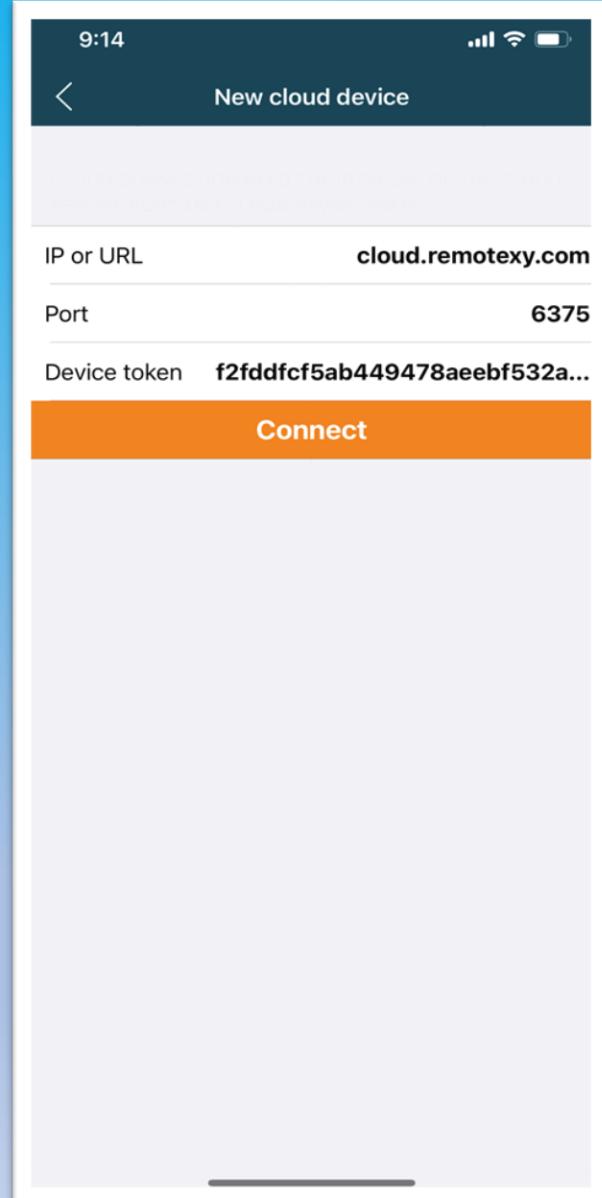
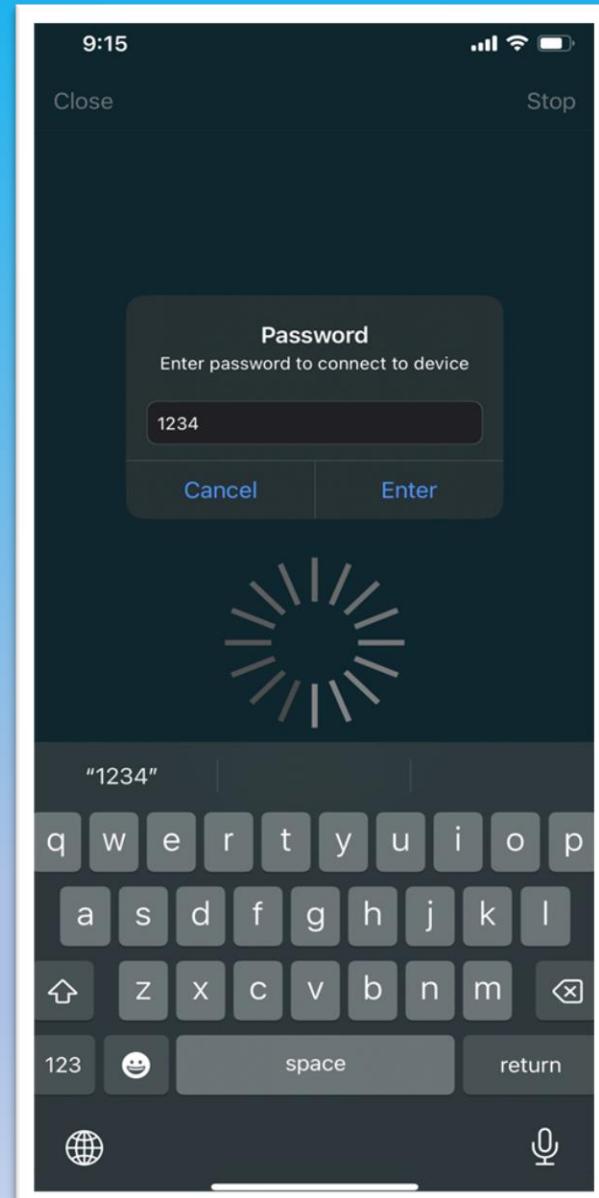
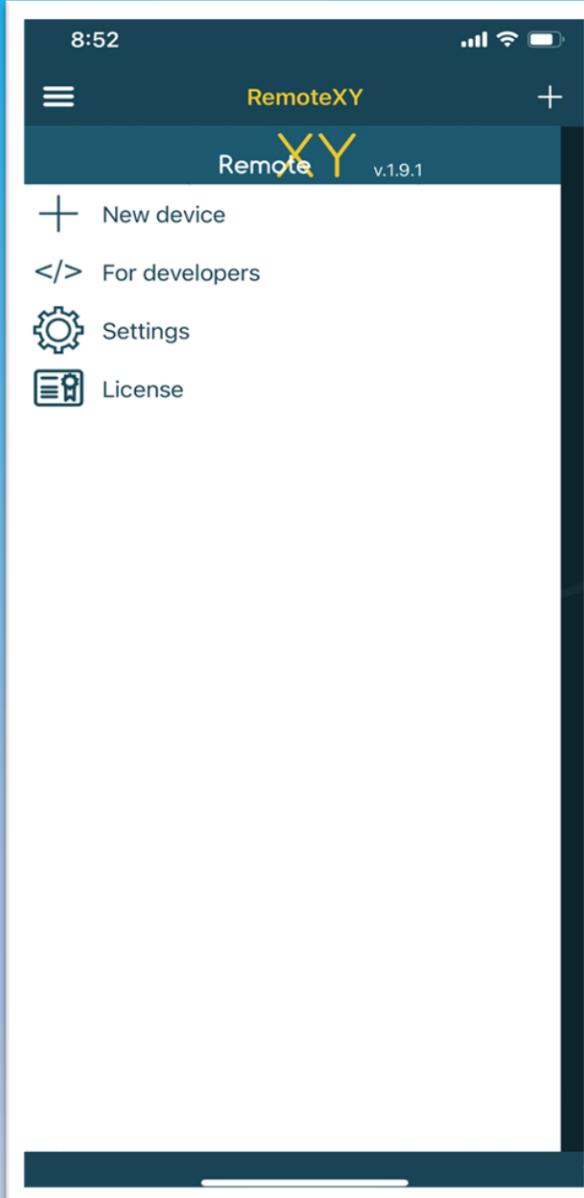
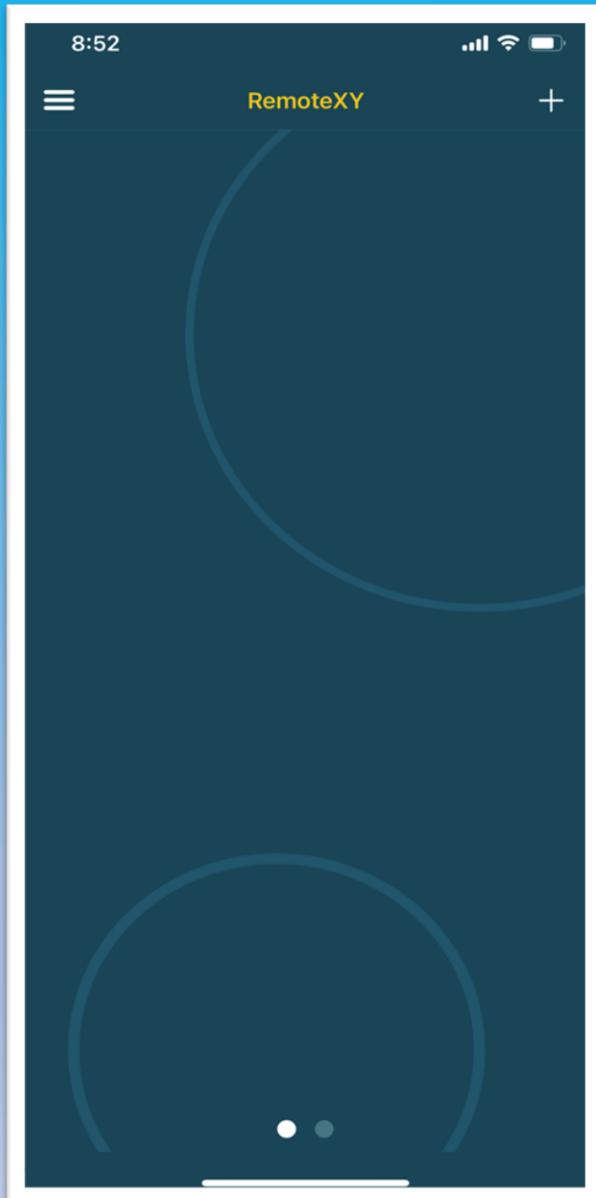
My cloud tokens

Connecting via a cloud server allows you to control the board from anywhere in the world. To connect the board on a cloud server you need a token. One board needs one token.

[Create new token](#)

Nº	Board name	Token	Device state	Server	Device port	App port	Actions
1	maykaryanlay	1e35410c594ca81a56146776cb78e91c	disconnected	cloud.remotexy.com	6376	6375	Edit Delete

Do as follows and fill your cloud server in IP or URL, port number in port and your token in Device token.



Then, get source codes and run it in your Arduino.

The screenshot shows the RemoteXY website interface. At the top, there is a dark green header bar with the RemoteXY logo on the left, user account information ('Wai Yar Aung *** My projects My tokens Sign out') in the center, and navigation links ('EDITOR APP HOW IT WORKS EXAMPLES COMMUNITY BLOG') and a UK flag icon on the right. Below the header, the main content area has a white background. A title 'Source code of project: Cloud Server' is displayed in bold black font. Below the title, a numbered list of six steps is enclosed in a green-bordered box:

1. Download the source code of the program, open it in the Arduino IDE.
2. Install **RemoteXY library** for Arduino IDE.
3. Compile the source code and upload it to the Arduino board using the Arduino IDE.
4. Correctly connect the **ESP8266 Wi-Fi module** to the Arduino board. ESP8266 Firmware AT_v0.40 or up.
5. Install the mobile app **RemoteXY ver.4.11.1** for smartphone/tablet.
6. Connect to Arduino using mobile app.

Below the list, there are three links: 'project.ino' (in blue), 'Download code' (in orange), and 'Download library' (in orange). A large code editor window is present at the bottom, containing the following C++ code:/*
-- Cloud Server --

This source code of graphical user interface
has been generated automatically by RemoteXY editor.
To compile this code using RemoteXY library 3.1.8 or later version
download by link <http://remotexy.com/en/library/>
To connect using RemoteXY mobile app by link <http://remotexy.com/en/download/>
- for ANDROID 4.11.1 or later version;
- for iOS 1.9.1 or later version;

Program Sketch

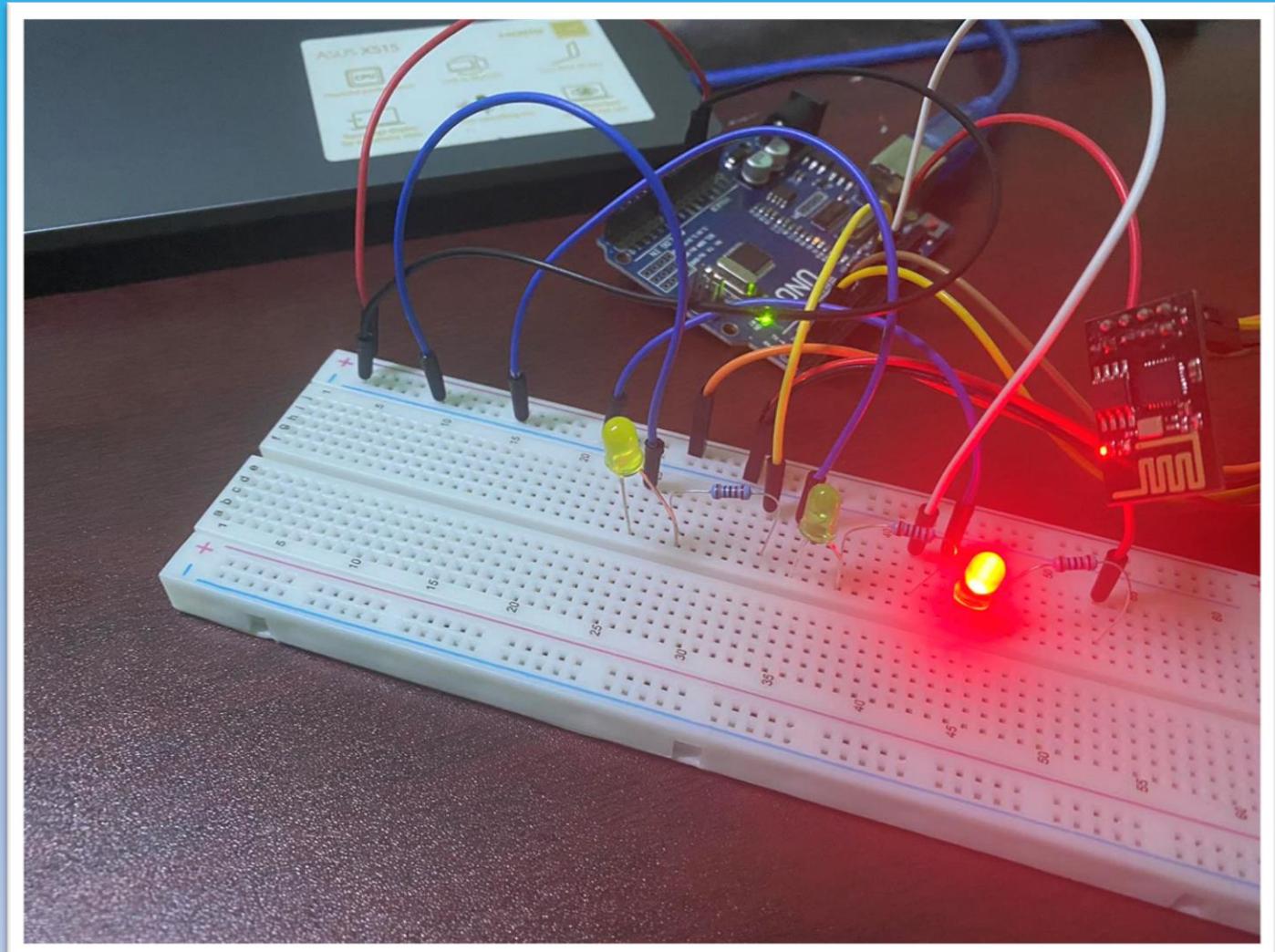
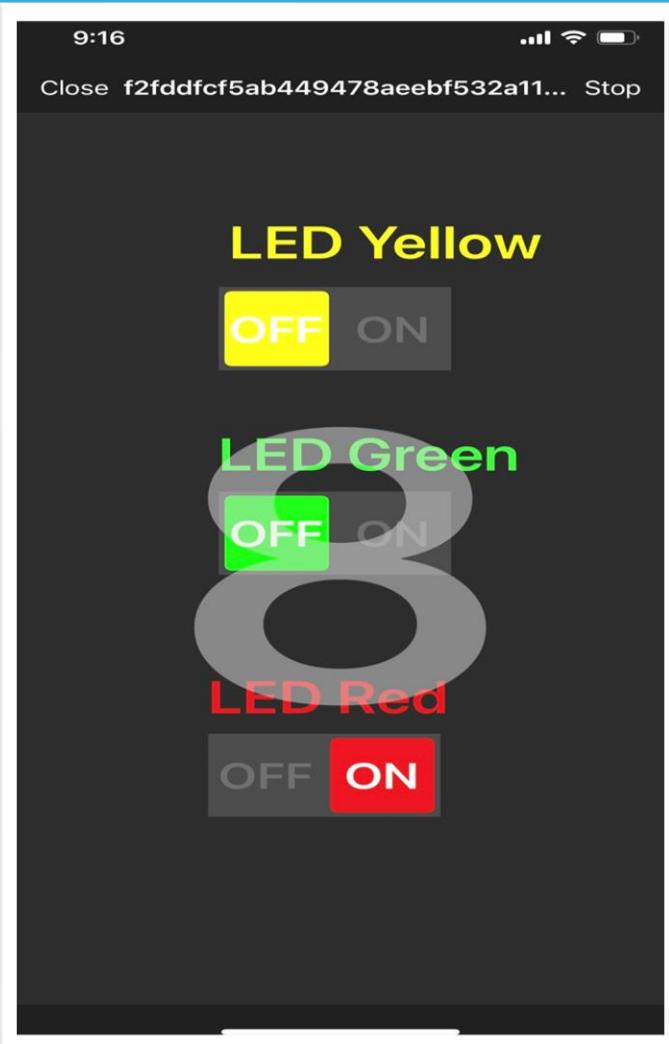
cloud_server.ino

```
1 #define REMOTEZY_MODE_ESP8266_HARDSERIAL_CLOUD
2
3 #include <RemoteZY.h>
4
5 // RemoteZY connection settings
6 #define REMOTEZY_SERIAL Serial
7 #define REMOTEZY_SERIAL_SPEED 115200
8 #define REMOTEZY_WIFI_SSID "WYA"
9 #define REMOTEZY_WIFI_PASSWORD "25692511"
10 #define REMOTEZY_CLOUD_SERVER "cloud.remotezy.com"
11 #define REMOTEZY_CLOUD_PORT 6376
12 #define REMOTEZY_CLOUD_TOKEN "f2ffdfcf5ab449478aeebf532a11cc58"
13 #define REMOTEZY_ACCESS_PASSWORD "1234"
14
15 // RemoteZY configure
16 #pragma pack(push, 1)
17 uint8_t RemoteZY_CONF[] = // 111 bytes
18 { 255,3,0,0,0,104,0,16,8,1,2,0,20,14,22,11,94,26,31,31,
19   79,78,0,79,70,70,0,2,0,20,41,22,11,135,26,31,31,79,78,0,
20   79,70,70,0,2,0,19,73,22,11,1,26,31,31,79,78,0,79,70,70,
21   0,129,0,19,65,24,6,1,76,69,68,32,82,101,100,0,129,0,20,33,
22   30,6,136,76,69,68,32,71,114,101,101,110,0,129,0,21,5,18,6,94,
23   76,69,68,32,89,101,108,108,111,119,0 };
24
25 // this structure defines all the variables and events of your control interface
26 struct {
27
28   // input variables
29   uint8_t switch_1; // =1 if switch ON and =0 if OFF
30   uint8_t switch_2; // =1 if switch ON and =0 if OFF
31   uint8_t switch_3; // =1 if switch ON and =0 if OFF
32
33 }
```

```
34   | // other variable
35   uint8_t connect_flag; // =1 if wire connected, else =0
36
37 } RemoteZY;
38 #pragma pack(pop)
39
40 ///////////////////////////////////////////////////
41 // END RemoteZY include
42 /////////////////////////////////////////////////
43
44 #define PIN_SWITCH_1 8
45 #define PIN_SWITCH_2 9
46 #define PIN_SWITCH_3 10
47
48
49 void setup()
50 {
51   RemoteZY_Init ();
52
53   pinMode (PIN_SWITCH_1, OUTPUT);
54   pinMode (PIN_SWITCH_2, OUTPUT);
55   pinMode (PIN_SWITCH_3, OUTPUT);
56
57   // TODO you setup code
58
59 }
60
61 void loop()
62 {
63   RemoteZY_Handler ();
64
65   digitalWrite(PIN_SWITCH_1, (RemoteZY.switch_1==0)?LOW:HIGH);
66   digitalWrite(PIN_SWITCH_2, (RemoteZY.switch_2==0)?LOW:HIGH);
```

```
42 //////////////////////////////////////////////////////////////////
43
44 #define PIN_SWITCH_1 8
45 #define PIN_SWITCH_2 9
46 #define PIN_SWITCH_3 10
47
48
49 void setup()
50 {
51     RemoteXY_Init ();
52
53     pinMode (PIN_SWITCH_1, OUTPUT);
54     pinMode (PIN_SWITCH_2, OUTPUT);
55     pinMode (PIN_SWITCH_3, OUTPUT);
56
57     // TODO you setup code
58
59 }
60
61 void loop()
62 {
63     RemoteXY_Handler ();
64
65     digitalWrite(PIN_SWITCH_1, (RemoteXY.switch_1==0)?LOW:HIGH);
66     digitalWrite(PIN_SWITCH_2, (RemoteXY.switch_2==0)?LOW:HIGH);
67     digitalWrite(PIN_SWITCH_3, (RemoteXY.switch_3==0)?LOW:HIGH);
68
69     // TODO you loop code
70     // use the RemoteXY structure for data transfer
71     // do not call delay()
72
73
74 }
```

Tests and Results



Discussion

- When connecting the ESP8266 wi-fi module for too long, it gets hot.
- Wi-fi module can disconnect sometime, uploading the code one more time can solve this problem.
- In this project, Remote XY is used but there are many other websites and platforms to approach ESP8266 wi-fi module.
- There are numerous breakout boards available based on ESP8266 Wi-fi Module like ESP8266 Node MCU V3. Because of its compact size, its most importantly used in autonomous project.
- 2.4GHz frequency can reach up to 150 feet (46 m) indoors and 300 feet (92 m) outdoors. The 2.4GHz band uses longer waves, which makes it better suited for longer ranges or transmission through walls and other solid objects.

References

<https://remotexy.com/>

<https://www.electronicwings.com/sensors-modules/esp8266-wifi-module>

https://erc-bpgc.github.io/handbook/electronics/Modules/wifi_module/

Thank you.