Discover Wi-Fi User Manual

Release：V1.4 Date：12th Dec 2013

Revision history

Rev	Date	Description	By
1.0	20130614	Initial version	Huangyin
1.1	20130620	Modifying some instruction	Huangyin
1.2	20130805	Review and Modifications	Ankur Tomar
1.3	20130816	Review and Modifications	Ankur Tomar
1.4	20131204	Adding http and https function	Huangyin

Contents

Section 1 Introduction

The Discover Wi-Fi is a low power, self-contained, certified Wi-Fi network controller module that provides simple serial-to-Wi-Fi connectivity to the internet and enables wireless connectivity to the STM32F4DISCOVERY kit (a very flexible development kit based on STM32F4 high performance microcontroller from STMicroelectronics). The Discover Wi-Fi board connects to the STM32F4DISCOVERY kit using a serial host interface [UART OR SPI]; it can also be used as a standalone Wi-Fi station or network controller. It can be used to enable wireless connectivity to the simplest products with minimal engineering resources.

1.1 Discover Wi-Fi

The Discover Wi-Fi board, a product designed by Embest; is based on Muratas' SN8200 Wi-Fi Network Controller module. The board design provides an easier connection to the STM32F4 Discovery kit and supports more overall software features through UART. Software demos are provided, including EZ Web Wizard solution, to help give users a quick and easy transition to wireless connectivity.

Board Features:

- 2.4GHz IEEE 802.11b/g/n
- Supports AP/STA Dual mode
- Built-in TCP/IP Stack, HTTP, DHCP, DNS, and Web Server
- Supports WPA/WPA2 PSK security
- Wi-Fi chipset: Broadcom BCM43362
- MCU: ST Microelectronics STM32 ARM Cortex-M3
- Host Interfaces: UART, SPI Interface & Standalone
- Other Interfaces: GPIO, ADC, DAC, I2C
- JTAG Interface for Debugging
- Power Options
    - 5V Power Jack
    - Mini USB Plug

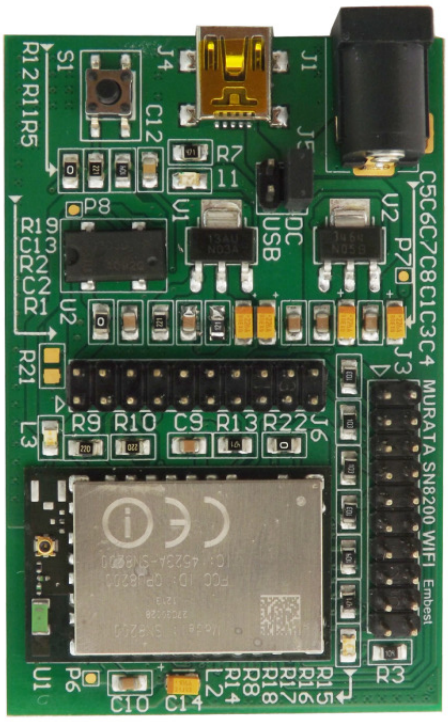


Figure 1-1 Discover Wi-Fi

1.2 STM32F4DISCOVERY Kit

The STM32F4DISCOVERY is a low-cost and easy-to-use development kit designed to allow quick evaluation and expediate development with an STM32F4 high-performance microcontroller. It is based on an STM32F407VGT6 and includes an ST-LINK/V2 embedded debug tool interface, ST MEMS digital accelerometer, ST MEMS digital microphone, audio DAC with integrated class D speaker driver, LEDs, pushbuttons and a USB OTG micro-AB connector. For more information please refer to the STMicroelectronics official URL: www.st.com/stm32f4-discovery.

You can purchase this kit from elment14, Order Codes: Farnell/element14 - 2009276, Newark - 87T3791

Figure 1-2 STM32F4DISCOVERY Kit

Features:

- STM32F407VGT6 microcontroller featuring 32-bit ARM Cortex-M4F core, 1 MB Flash, 192 KB RAM in an LQFP100 package
- On-board ST-LINK/V2 with selection mode switch to use the kit as a standalone STLINK/V2 (with SWD connector for programming and debugging)
- Board power supply: through USB bus or from an external 5 V supply voltage
- External application power supply: 3V & 5V
- LIS302DL, ST MEMS motion sensor, 3-axis digital output accelerometer
- MP45DT02, ST MEMS audio sensor, omni-directional digital microphone
- CS43L22, audio DAC with integrated class D speaker driver
- Eight LEDs:
  - LD1 (red/green) for USB communication
  - LD2 (red) for 3.3 V power
  - 4 user LEDs; LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)
  - 2 USB OTG LEDs LD7 (green) VBus and LD8 (red) over-current
- Two push buttons (user and reset)
- USB OTG FS with micro-AB connector
- Extension header for all LQFP100 I/Os for quick connection to prototyping board and easy probing.

Section 2 Hardware Features

2.1 Board Physical Dimensions

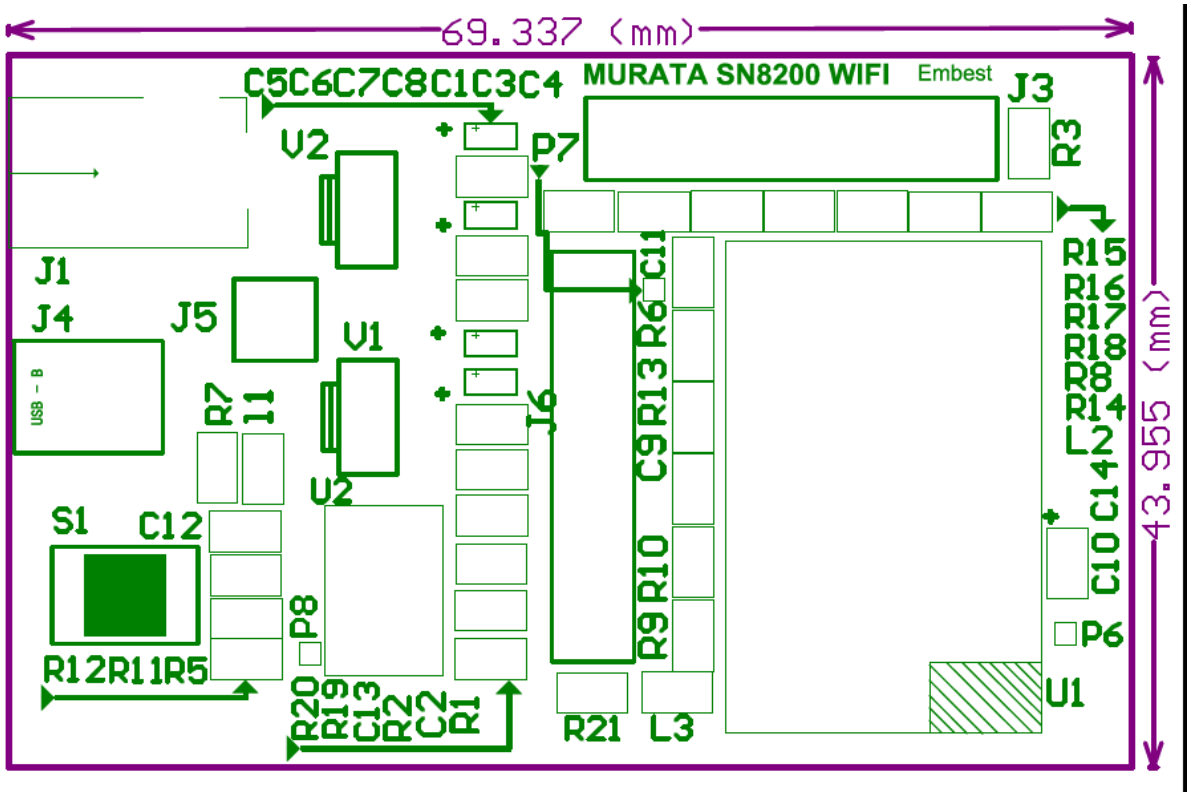


Figure 2-1 DISCOVER WI-FI PCB

- Size: 69mmx44mm
- Board layers: 4
- Board thickness: 1.6mm

2.2 Board Electrical Characteristics

- Power: 5V, 2A; or Mini-USB power.
- Operating Temperature: 0~70°C.
- Power Consumption: around 2.5 W.

2.3 Board Technical Description

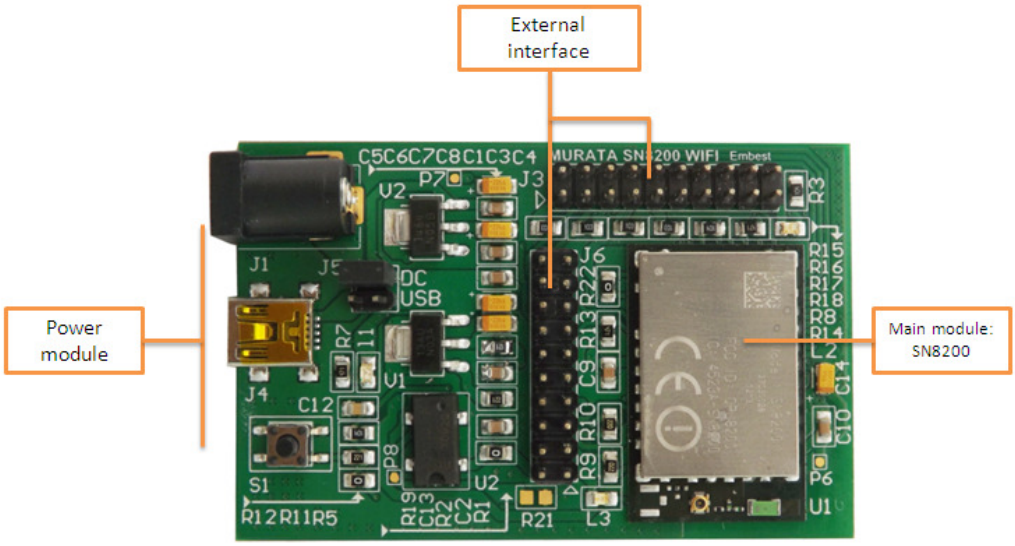


Figure 2-2 Discover Wi-Fi Hardware

2.3.1 Block Description

- Wi-Fi Module: SN8200 Wi-Fi module
- Power Section (J1, J4): The board is powered by Mini-USB or 5V, 2A DC.
- Switch and LEDs: One reset switch and two signal LEDs.
- External Interface
    - JTAG interface (J3): Standard 20 pin interface, used for Module Firmware Loading.
    - User interface (J6): External interface for users.

2.3.2 Wi-Fi Module - SN8200

Features

- 2.4GHz IEEE 802.11b/g/n Radio Technology
- Wi-Fi Chip - Broadcom BCM43362
- MCU - STM32 ARM Cortex-M3
- Dimensions: 30.5 x 19.4 x 2.8 mm
- Package: LGA
- On-Board Antenna
- Max Receive Sensitivity: -96dbm @ b mode/11Mbps
- Transmit Power: +18 dBm
- Host Interfaces: UART & SPI
- Other Interfaces: ADC, DAC, I2C, GPIO
- Operating Temperature Range: -30ºC to 85ºC
- ROHS Compliant
- FCC/IC certified, CE compliant
- PN 88-00151-00
- EVK/SDK P/N 88-00151-85

Competitive Advantages

- CE Certified
- TCXO/XTAL that supports extended product life
- Wide Link Budget (up to 113 dB)
- Easy software integration
    - AP/STA dual mode
    - Built-in Wi-Fi security support for WPA-PSK, WPA2-PSK
    - Built-in TCP/IP stack
    - Built-in DHCP, DNS
    - Built-in HTTP server for AP mode
    - Simple integration interface – Serial Network Interface (SNIC) support socket interface

SN8200 Block Diagram

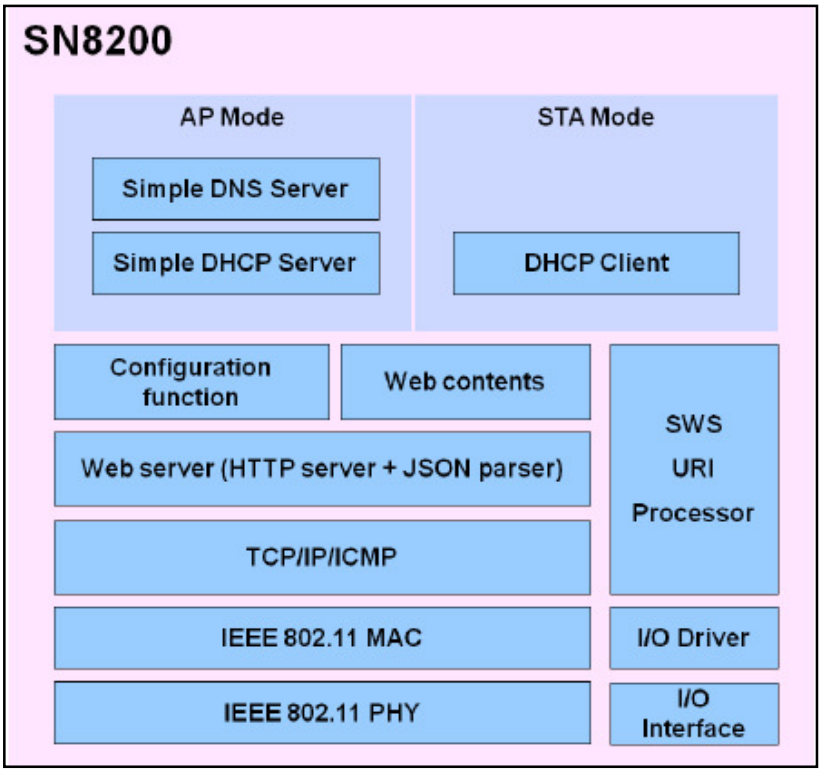


Figure 2-4 SN8200 Diagram

2.3.3 External Interface – Pin Detail

Table2-1 External Interface Pin Functions

	PINS	Function	PINS	Function
J3	1	VCC-MCU	2	VCC-MCU
	3	SPI_MISO/JRTST	4	GND
	5	JTMS	6	GND
	7	JTCK	8	GND
	9	-	10	GND
	11	SPI_SCK/JTDO	12	GND
	13	RST_N_SW	14	GND
	15	-	16	GND
	17	-	18	GND
	19	-	20	GND
J6	1	VCC-MCU	2	VCC-MCU
	3	UART_RX	4	SPI1_MOSI
	5	UART_TX	6	SPI1_SCK
	7	UART_RTX	8	SPI1_NSS
	9	UART_CTS	10	SPI1_MISO
	11	GND	12	GND
	13	LED-net	14	SPI3_SCK/JTDO
	15	RST_N_SW	16	SPI3_MOSI/LED-ready
	17	Interrupt	18	SPI3_MISO/JRTST
	19	WAKER UP	20	SPI3_NSS/JTDI

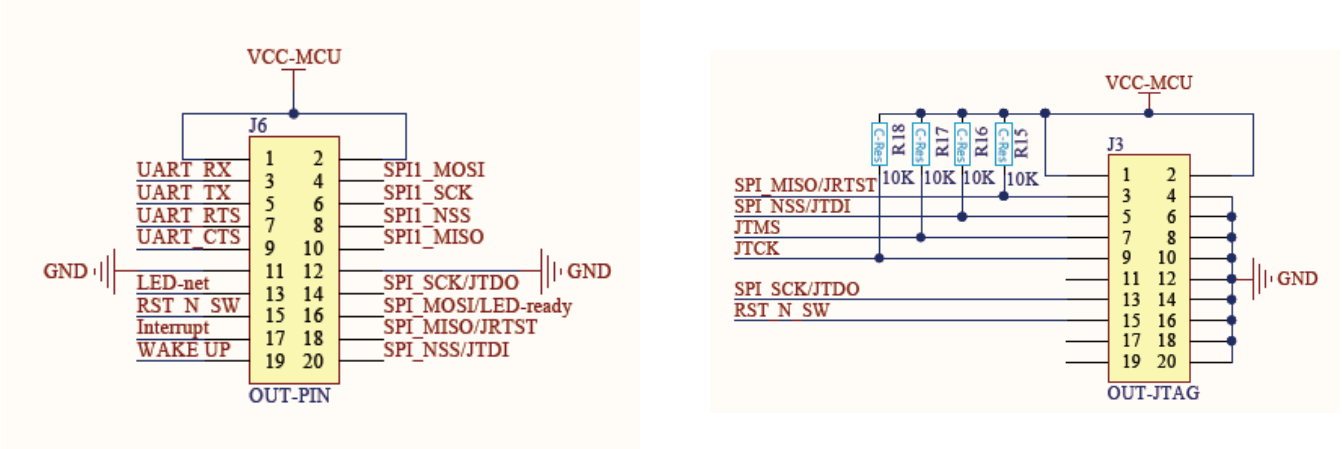


Figure 2-3 External Interface Schematic

Section 3 Quick Start (Standalone Mode)

3.1 Powering ON

The Discover Wi-Fi board can be powered using a 5V-2A DC power adapter OR Mini USB power supply, please set jumper J5 appropriately in accordance with the chosen power supply, DC or USB. Once the board is powered 'ON', by default it will go into AP mode.

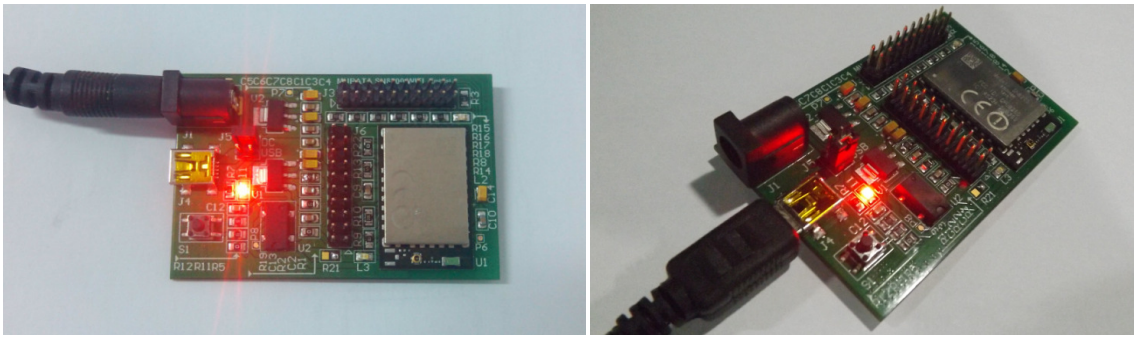


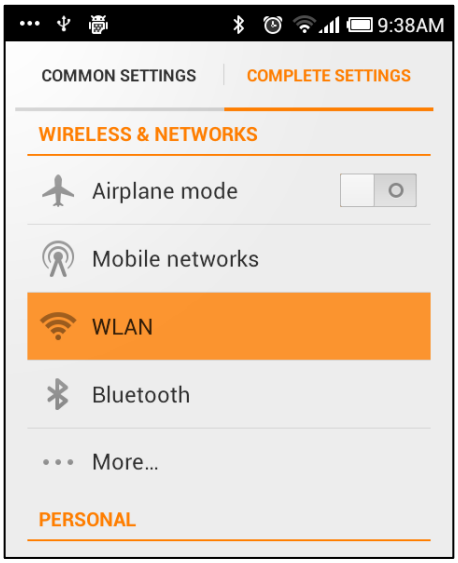
Figure 3-1 Power Connection (Left: DC Power; Right: USB Power)

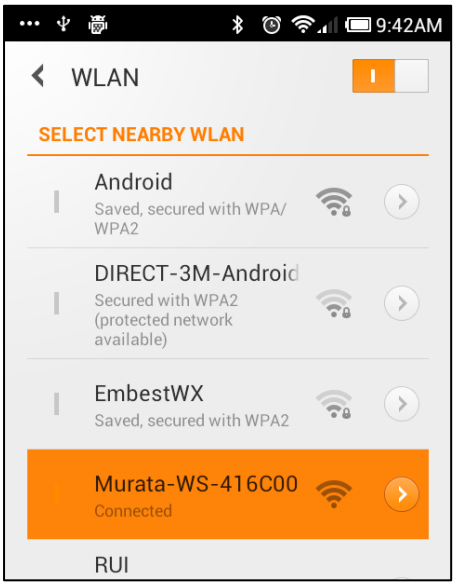
3.2 First Start-up

First we need a computer/laptop or Smartphone or other equipment with Wi-Fi capability. Here we are using a smartphone as an example.

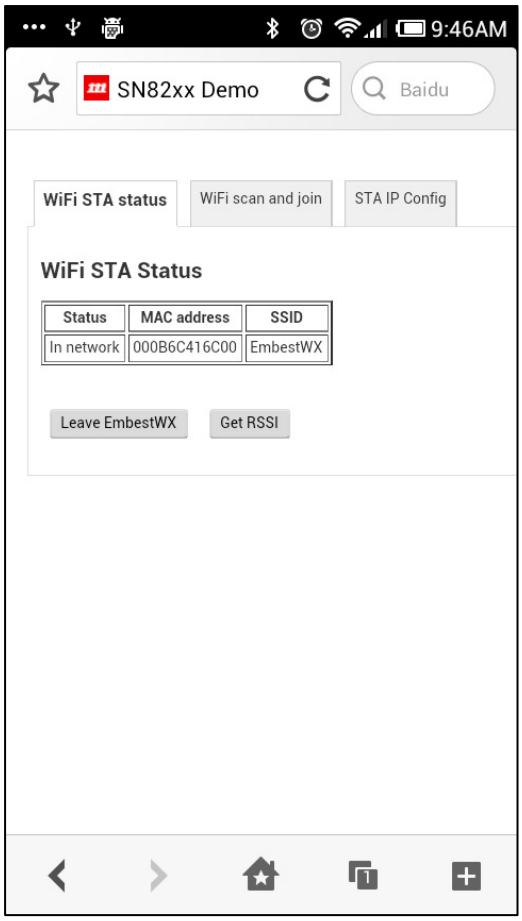
- ✓ Step1: Open your WLAN Settings

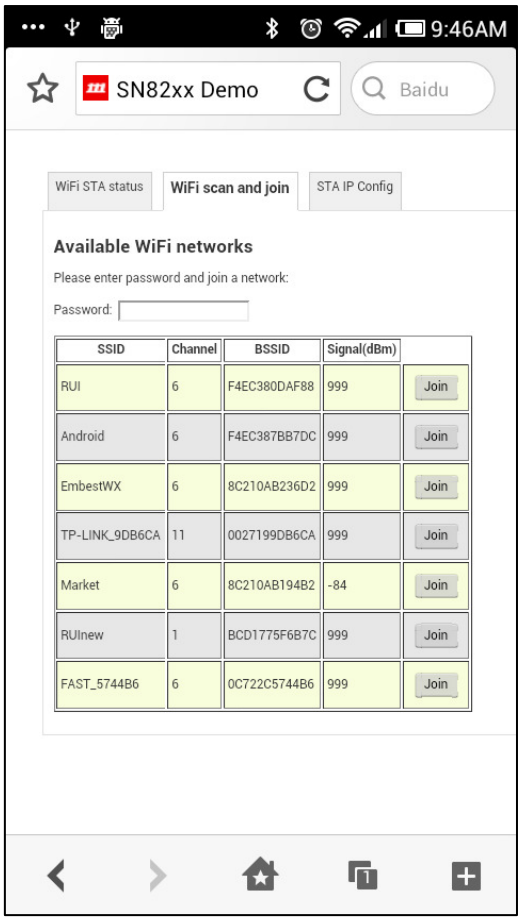
- ✓ Step2: You'll find the "Murata Wi-Fi wireless AP", because the Discover Wi-Fi module is running in AP mode by default. Now "Join" the network.

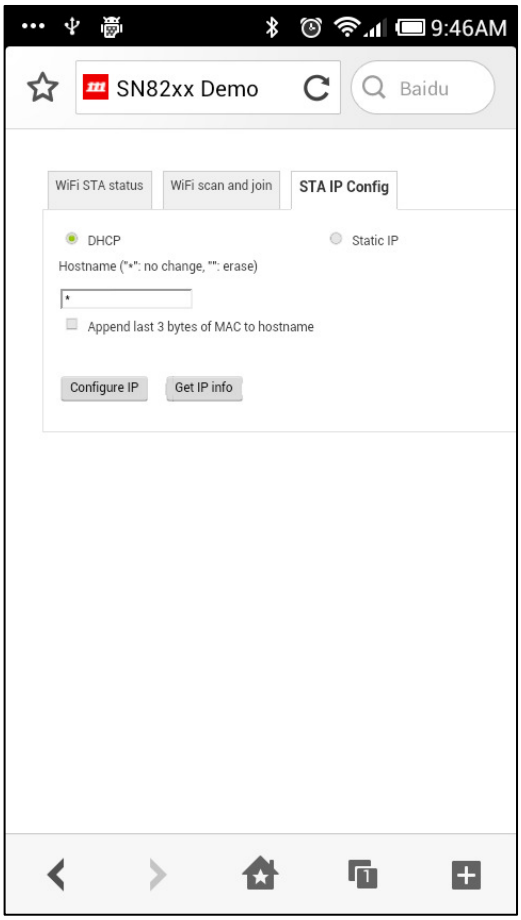




✓ Step3: Go to the mobile browser of your choice, and visit “SN8200.com”.


9:46AM
SN82xx Demo
Baidu
WiFi STA status
WiFi scan and join
STA IP Config
WiFi STA Status
Status
MAC address
SSID
In network
000B6C416C00
EmbestWX
Leave EmbestWX
Get RSSI


9:46AM
SN82xx Demo
Baidu
WiFi STA status
WiFi scan and join
STA IP Config
Available WiFi networks
Please enter password and join a network:
Password:
SSID
Channel
BSSID
Signal(dBm)
RUI 6 F4EC380DAF88 999 Join
Android 6 F4EC387BB7DC 999 Join
EmbestWX 6 8C210AB236D2 999 Join
TP-LINK_9DB6CA 11 0027199DB6CA 999 Join
Market 6 8C210AB194B2 -84 Join
RUInew 1 BCD1775F6B7C 999 Join
FAST_5744B6 6 0C722C5744B6 999 Join


9:46AM
SN82xx Demo
Baidu
WiFi STA status
WiFi scan and join
STA IP Config
DHCP
Static IP
Hostname ("*": no change, "": erase)
*
Append last 3 bytes of MAC to hostname
Configure IP
Get IP info

Section 4 Working with STM32F4DISCOVERY

The Discover Wi-Fi module provides UART and SPI host interfaces, Embest has provided a number of test code examples to help the end user to control the Discover Wi-Fi expansion board from the STM32F4DISCOVERY Kit using the UART interface (for SPI interfaces users can develop their own solution using the Murata SPI software solution, for more information please refer to Discover Wi-Fi/SN8200 Reference Material/SNIC-SPI-01-2B091.exe). Below is the list of developed main functions for the UART solution:

0	Get Wi-Fi Status
1	Wi-Fi Scan
2	Join Wi-Fi
3	Get IP
4	TCP Client
5	TCP Server
6	Send From Stock
7	Disconnect Wi-Fi
8	AP ON/OFF
9	UDP Client
a	UDP Server
b	Wi-Fi OFF
c	Wi-Fi ON
d	HTTP get req
e	HTTP post req
f	HTTP post Jsonreq
g	HTTP chunked post req
h	HTTPS get req
i	TLS client
j	TLS server(HTTPS server)
m	Show Menu
q	Press 'q' to Quit

4.1 System Setup

4.1.1 Hardware Setup

It's better to have two available boards (EVK1 and EVK2) to complete all the tests. Here EVK2 will mainly be used as and *HTTP* or *HTTPS* server. If you do not have another Discover Wi-Fi, it is possible to setup your local server to finish the test yourself.

EVK1 Setup

- ✓ First connect the STM32F4DISCOVERY kit to the Discover Wi-Fi module using the provided DuPont cables. For this example we will be using the UART interface between the STM32F4Discovery kit and the Discover Wi-Fi module. Please refer to Figure 4-2
  (or refer to the schematic <WI-FI_SN8200_schematic.pdf>).

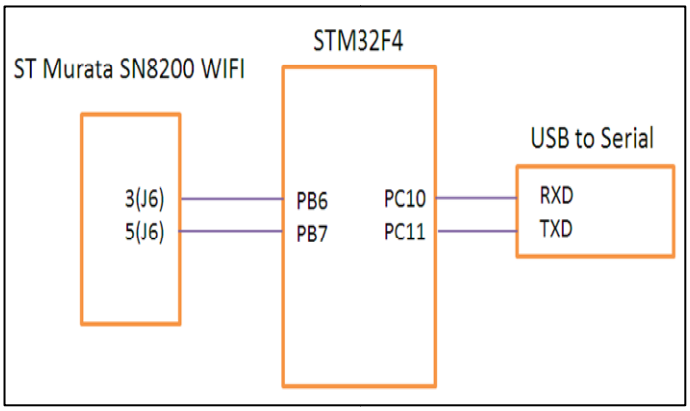


Figure 4-1 Connection

- ✓ We also need a Hyper-terminal connection between a PC and the STM32F4DISCOVERY kit using RS232, for which we will use a USB to serial converter (or you need a TTL to RS232 logic converter if a USB to serial converter is not available).

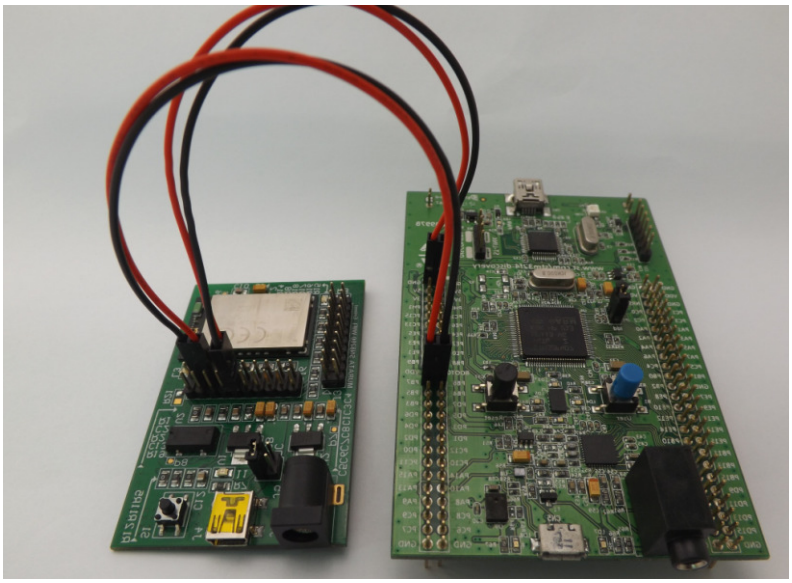
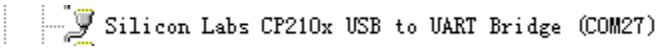


Figure 4-2 Physical Connection (USB to serial not included)

- ✓ Connect the USB end of the USB-RS232 converter to the computer/PC and see if it's installed and detected as a COM port on the computer/PC as below (the number following "COM" will vary according to your computer/PC configuration, here it's COM11):



- ✓ Now setup a Hyper-terminal communication on your computer/PC using the settings below:

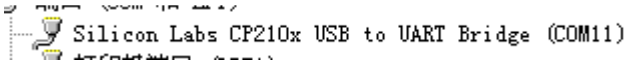
  - Port: COMx (accordingly)
  - Bits: 115200
  - Data bits: 8
  - Parity Check: none;
  - Stop: 1
  - Data flow control: none

Note: *Recommended computer/PC configuration:*

- *2.0GHz (or higher) CPU*
- *512MB RAM*
- *USB interfaces*
- *A serial interface*
- *Windows XP or above operating system*
- *Pre-installed KEIL IDEv4.70, or please follow the below steps to install KEIL IDE.*
- *Install the **Setup_server.exe** (or **testserver.exe**) and the **Setup_client.exe** (or **testclient.exe**) can be founder under **"Discover-wifi V2/software/"***

EVK2 Setup

You need to connect the PC and a second ST Murata Wi-Fi module via UART (refer to the schematic <*WI-FI_SN8200_schematic.pdf*> or the footprint). You need a TTL to RS232 transfer module, and here we will use a USB to serial module to setup serial communication. Connect the module to your PC using the USB to serial cable, a serial port should be detected on your PC which can be checked under the PC's Device Manager (COMxx) as below (here its detected as COM11):



Then open your Hyper-terminal on the PC, and setup as per the below settings:

- Port: COMxx
- Bits: 115200
- Data bits: 8
- Parity Check: None;
- Stop: 1
- Data flow control: None

4.1.2 Turn the System Power ON

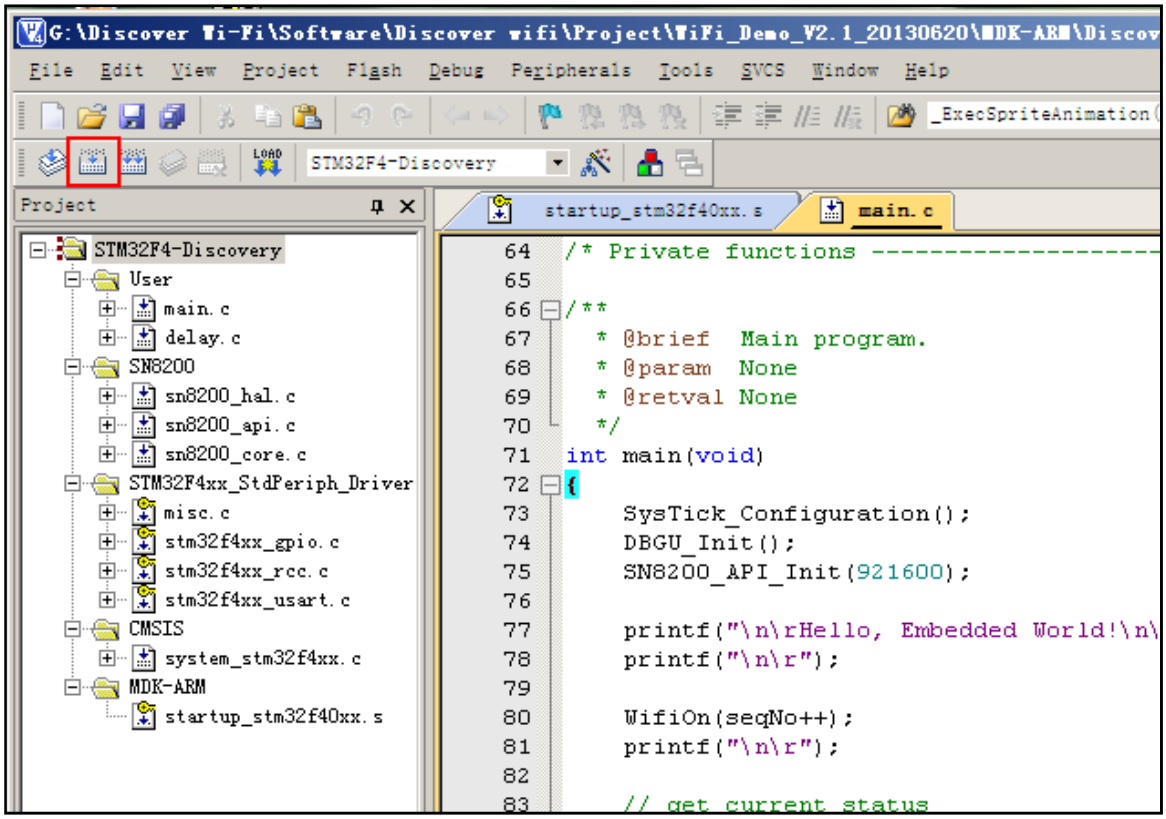
Turn the system power ON for the STM32F4DISCOVERY kit and Discover Wi-Fi board.

- STM32F4DISCOVERY Kit: Connect the MicroUSB cable between STM32F4DISCOVERY Kit USB port (CN5) and computer/PC USB port.
- Discover Wi-Fi Board: You can use either MiniUSB cable or 5V@ 2A DC to power ON the Wi-Fi module, please refer to **Section 3.1**.

4.2 Software Setup

✓ First open the Sample Project in KEIL MDK ARM IDE (location: software\ST-Discovery-Wifi\Project\WiFi_Demo_V2.2_20130620\MDK-ARM).

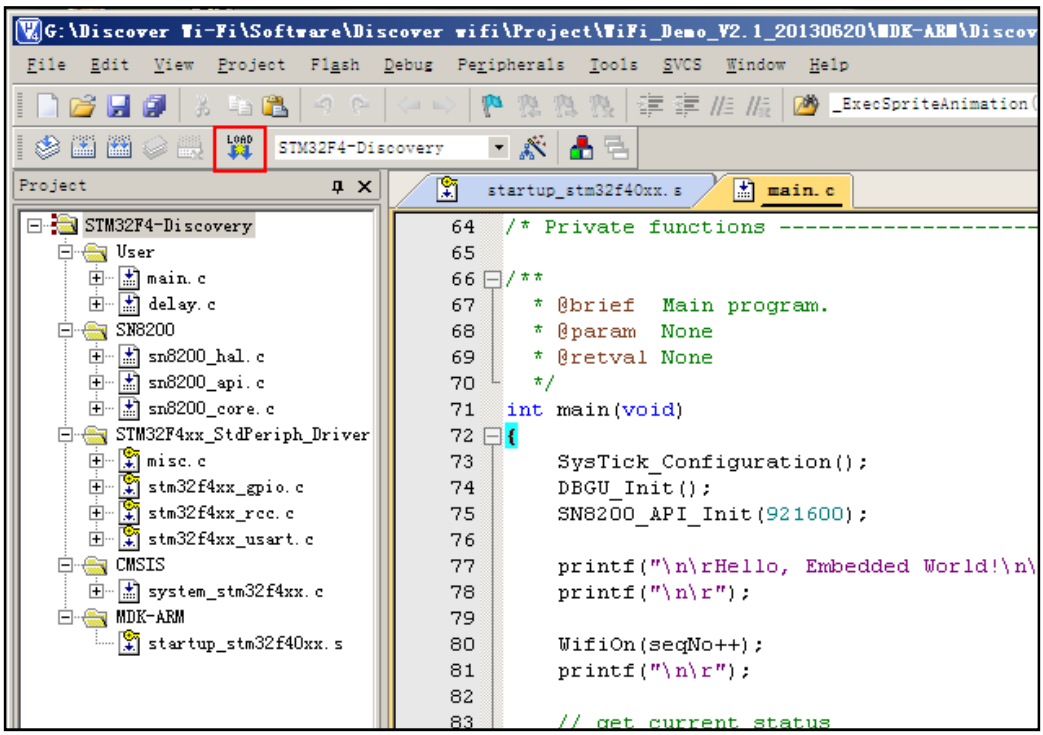
✓ Now build the project by clicking the "Build" icon (highlighted in the image below) in IDE or by pressing the "F7" function key.



✓ Make sure the project is built successfully without any errors.

```
linking...
Program Size: Code=16200 RO-data=632 RW-data=192 ZI-data=31696
FromELF: creating hex file...
".\Output\DiscoverWIFI.axf" - 0 Error(s), 0 Warning(s).
```

✓ Once the build has completed successfully, download the code into the board by clicking the "Download" icon (highlighted in the image below) and wait for the download to finish.



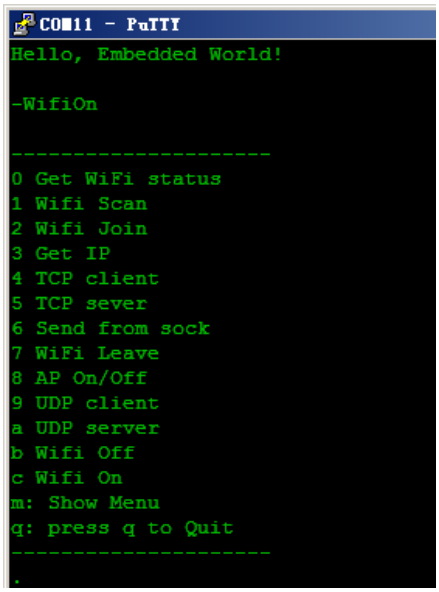
✓ Once the code is downloaded into the board it will be verified and you should see the below screen.

```
Load "G:\\Discover wifi\\Project\\WiFi_Demo_V2.1_20130620\\MDK-ARM\\Output\\DiscoverWIFI.axf"
Erase Done.
Programming Done.
Verify OK.
```

4.3 Running Test Functions

Note: This demo demonstrates all the available testing functions, so please follow as per the provided instructions below:

Once the sample code is downloaded into the board (as described in Section 4.2), please RESET the STM32F4DISCOVERY Kit. Upon RESET you should see the below message on your Hyper-Terminal screen:



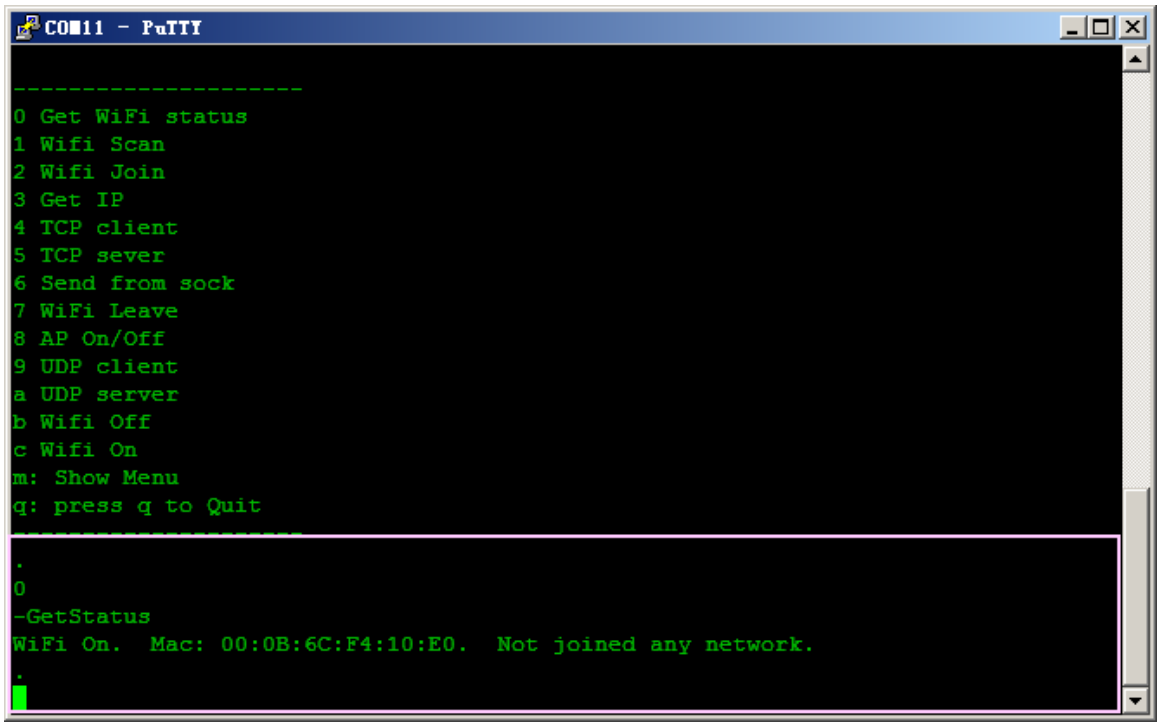
The compete testing process is divided into two sections; STA and AP. You can press "m" anytime to go back to the menu.

```
0 Get WiFi status
1 Wifi Scan
2 Wifi Join
3 Get IP
4 TCP client
5 TCP sever
6 Send from sock
7 WiFi Leave
8 AP On/Off
9 UDP client
a UDP server
b Wifi Off
c Wifi On
d HTTP get req
e HTTP post req
f HTTP post Json req
g HTTP chunked post req
h HTTPS get req
i TLS client
j TLS server (HTTPS server)
m: Show Menu
q: press q to Quit
```

4.3.1 STA Test Functions

Basic Functions

✓ First press "0", the STM32F4 will show its Wi-Fi status on Hyper-Terminal, as below:



```
0
-GetStatus
WiFi On.  Mac: 00:0B:6C:F4:10:E0.  Not joined any network.
.
```

✓ Now by pressing "1" (Wi-Fi- Scan) you can scan all the available wireless networks. The terminal will display the scanned wireless networks.

```
1
-WifiScan
.
.
.
.
SSID:                liaops CH:  6 RSSI: 206 Sec: 4
SSID:               Embest CH:  6 RSSI: 231 Sec: 4
SSID:  INTEST_Meeting-Room CH:  6 RSSI: 170 Sec: 4
SSID:        juanpi_com_hr CH:  9 RSSI: 231 Sec: 4
SSID:                 CMCC CH: 11 RSSI: 186 Sec: 0
SSID:           CMCC-AUTO CH: 11 RSSI: 185 Sec: 4
SSID:       juanpi_com_pro CH:  6 RSSI: 231 Sec: 4
SSID:           brainaire CH:  6 RSSI: 231 Sec: 6
SSID:           CMCC-AUTO CH:  6 RSSI: 165 Sec: 4
SSID:                 CMCC CH:  6 RSSI: 164 Sec: 0
SSID:       juanpi_com_pub CH:  1 RSSI: 231 Sec: 6
.
```

✓ Then pressing "2" (Wi-Fi Join) to join the selected Wi-Fi network, here we will choose the Embest network and select between WPA or WPA 2 security by pressing "2" or "4" and then enter the security key to join the network.

```
2
-WifiDisconn
Enter SSID:
Embest
Enter Security Mode (e.g., 0 for open, 2 for WPA TKIP, 4 for WPA2 AES):
4
Enter Security Key:
embest999
```

✓ Once the network is joined, terminal will display the successful network joined status.

```
-WifiJoin
.
Network UP
.
Join success
.
-SnicInit
-SnicIPConfig
.
IPConfig OK
.
```

✓ Now you can again verify the Wi-Fi connection status by pressing "0". It will show the joined wireless network.

```
0
-GetStatus
WiFi On.  Mac: 00:0B:6C:F4:10:E0.  Joined SSID: Embest
.
```

✓ To discover the assigned IP address and STA mode, press "3", the assigned IP address in this case is: 192.168.2.125

```
3
-SnicInit

Interface Type? (0: STA  1: AP)
0
```

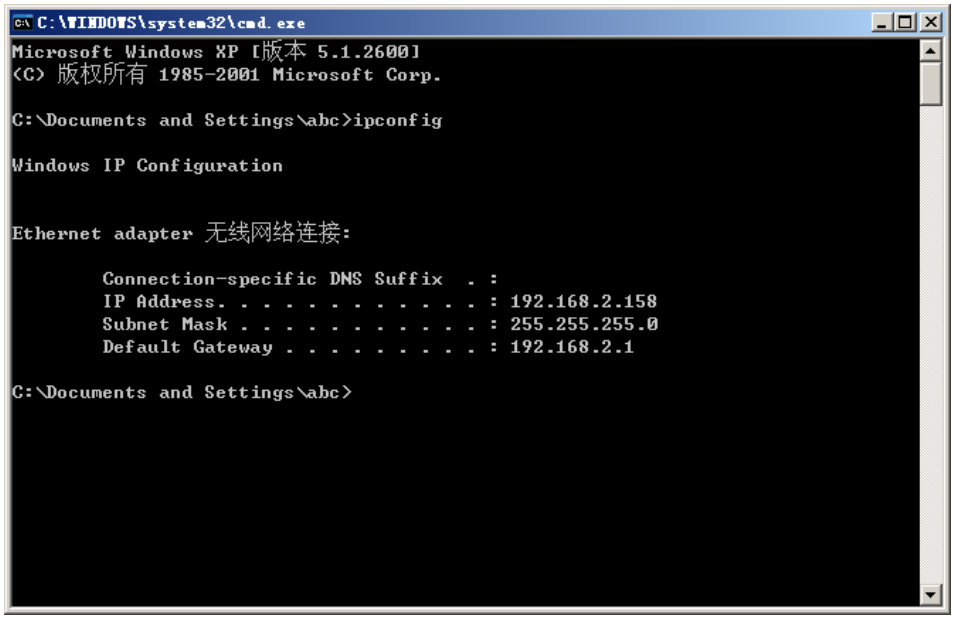
```
3
-SnicInit

Interface Type? (0: STA  1: AP)
0
-SnicGetDhcp
.
IP assigned as 192.168.2.125
.
```

TCP Testing

STM32F4DISCOVERY Kit>>Discover Wi-Fi ----------Works as a CLIENT

Computer/PC ----------Works as a SERVER

✓ Connect your computer/PC to the same wireless network (Embest in this case), and get your IP address: 192.168.2.158



✓ Now setup a TCP client on the STM32F4-WI-FI and connect to the TCP server created on computer/PC. Please follow the below steps to setup the SERVER and the CLIENT.

- On the PC, first run the testserver application: testserver.exe. (location: Discover Wi-Fi\Software)

```
G:\Discover Wi-Fi\Software>testserver.exe 192.168.2.158 1234
----------------------
0 TCP client
1 TCP server
2 Send from sock
3 UDP server
4 UDP send to soft AP


q Quit
----------------------
```

```
G:\Program Files\Murata\testserver>testserver.exe 192.168.2.158 1234
----------------------
0 TCP client
1 TCP server
2 Send from sock
3 UDP server
4 UDP send to soft AP


q Quit
----------------------
```

- Press ”1” on the PC to set the PC as a TCP SERVER.

```
----------------------
Trying to create a socket, and accept incoming connections.
Socket 0 listening on port 0x4d2.
----------------------
0 TCP client
1 TCP server
2 Send from sock
3 UDP server
4 UDP send to soft AP


q Quit
----------------------
```

- Now on Hyper-terminal, set the STM32F4-WI-FI module as a TCP CLIENT by pressing ”4” from the main Menu. Hyper-terminal will display a message of opening Socket 4. Now enter the SERVER IP address (computer/PC) and the SERVER port number:

```
4
-tcpCreateSocket
Socket 4 opened
.
Enter server IP to connect:
192.168.2.158
```

```
Enter server port number:
1234
```

- Now the socket connection if UP:

```
-tcpConnectToServer
.
Socket connection UP
.
```

- The PC (test server) will display a message as below:

```
Connection accepted
Socket 1 will be used to send data.
```

- The connection has been created, and we can now use Socket 1 to send data.
- On the STM32F4-Wi-Fi module CLIENT, press ”6”. Here we can set Socket 4 to send data:

```
6
Enter socket number to send from:
4
```

- Choose the default ”0”

```
6
Enter socket number to send from:
Content Option? (0: Default  1: User specific)
0
```

```
-sendFromSock
pkt 1, 128 bytes sent
.
```

- 128bytes will be sent as default, and the SERVER will display that the data has been received.

```
Recv: 128
```

- Let’s change the direction, now the SERVER will send the data to the CLIENT.
- Press”2” on the SERVER window:

```
Connection accepted
Socket 1 will be used to send data.
Recv: 128
Trying to send 200 bytes from socket 1.
Bytes sent: 200
----------------------
0 TCP client
1 TCP server
2 Send from sock
3 UDP server
4 UDP send to soft AP


q Quit
----------------------
```

- The SERVER has sent 200 bytes of data as default, and the CLIENT displays a notice indicating that it has received the data.

```
200 bytes received from socket 4
.
```

Note: Similar steps can be done in the opposite direction.

1. Create a TCP SERVER on the STM32 WI-FI module
2. Create a TCP CLIENT on PC and connect to the TCP server on STM32
3. Send data back and forth.

4.3.2 AP Test Functions

Basic AP Function

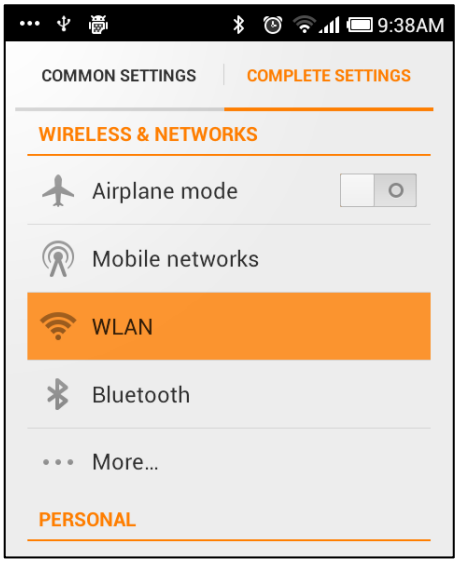
✓ The Discover Wi-Fi module is in AP mode as default, but you can press ”8” to change its state.

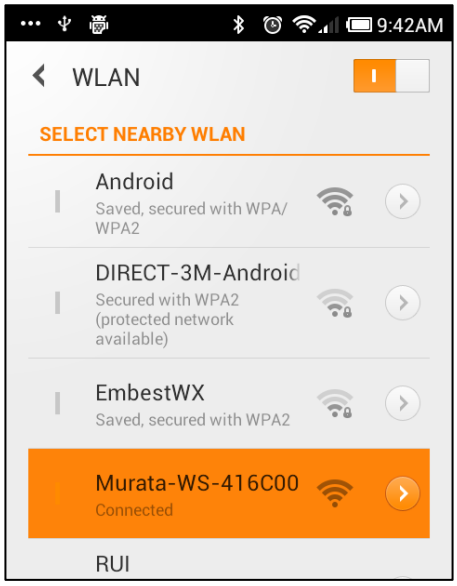
```
8
-AP status
AP is OFF
.
8
-AP status
AP is ON
.
```

✓ The ON/OFF state of AP mode can be tested by using another Wi-Fi device, here we are using a smartphone to do so:

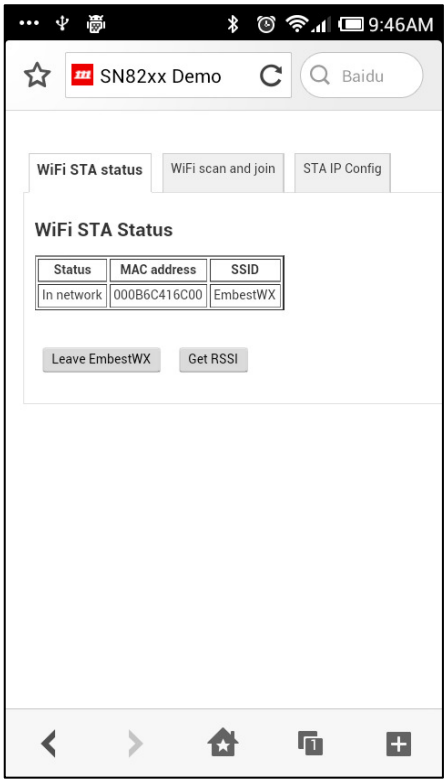
- Step1: Open your WLAN Settings

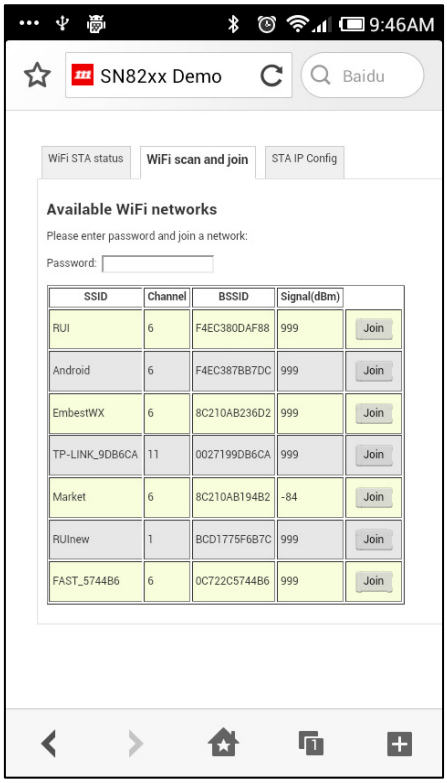
- Step2: You'll find the "Murata Wi-Fi wireless AP", because the Discover Wi-Fi module is running in AP mode by default. Now "Join" the network.

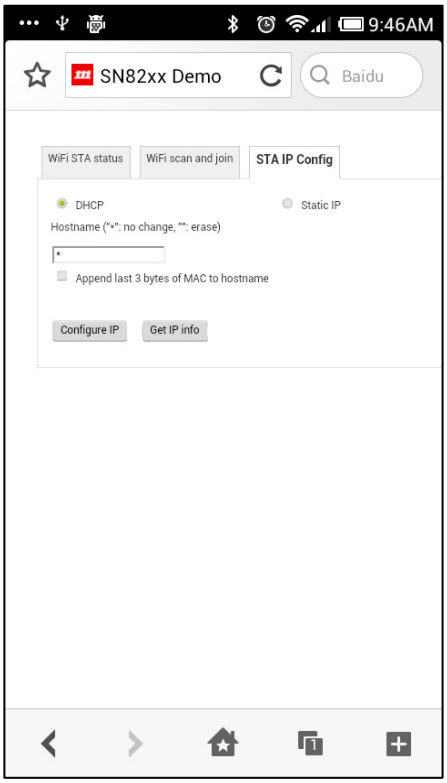




- Step3: Go to the mobile browser of your choice, and visit SN8200.com".


9:46AM
SN82xx Demo
Baidu
WiFi STA status
WiFi scan and join
STA IP Config
WiFi STA Status
Status
MAC address
SSID
In network
000B6C416C00
EmbestWX
Leave EmbestWX
Get RSSI


9:46AM
SN82xx Demo
Baidu
WiFi STA status
WiFi scan and join
STA IP Config
Available WiFi networks
Please enter password and join a network:
Password:
SSID
Channel
BSSID
Signal(dBm)
RUI 6 F4EC380DAF88 999 Join
Android 6 F4EC387BB7DC 999 Join
EmbestWX 6 8C210AB236D2 999 Join
TP-LINK_9DB6CA 11 0027199DB6CA 999 Join
Market 6 8C210AB194B2 -84 Join
RUInew 1 BCD1775F6B7C 999 Join
FAST_5744B6 6 0C722C5744B6 999 Join


9:46AM
SN82xx Demo
Baidu
WiFi STA status
WiFi scan and join
STA IP Config
DHCP
Static IP
Hostname ("*": no change, "": erase)
*
Append last 3 bytes of MAC to hostname
Configure IP
Get IP info

UDP Testing

STM32F4DISCOVERY Kit>>Discover Wi-Fi ---------- Work as CLIENT

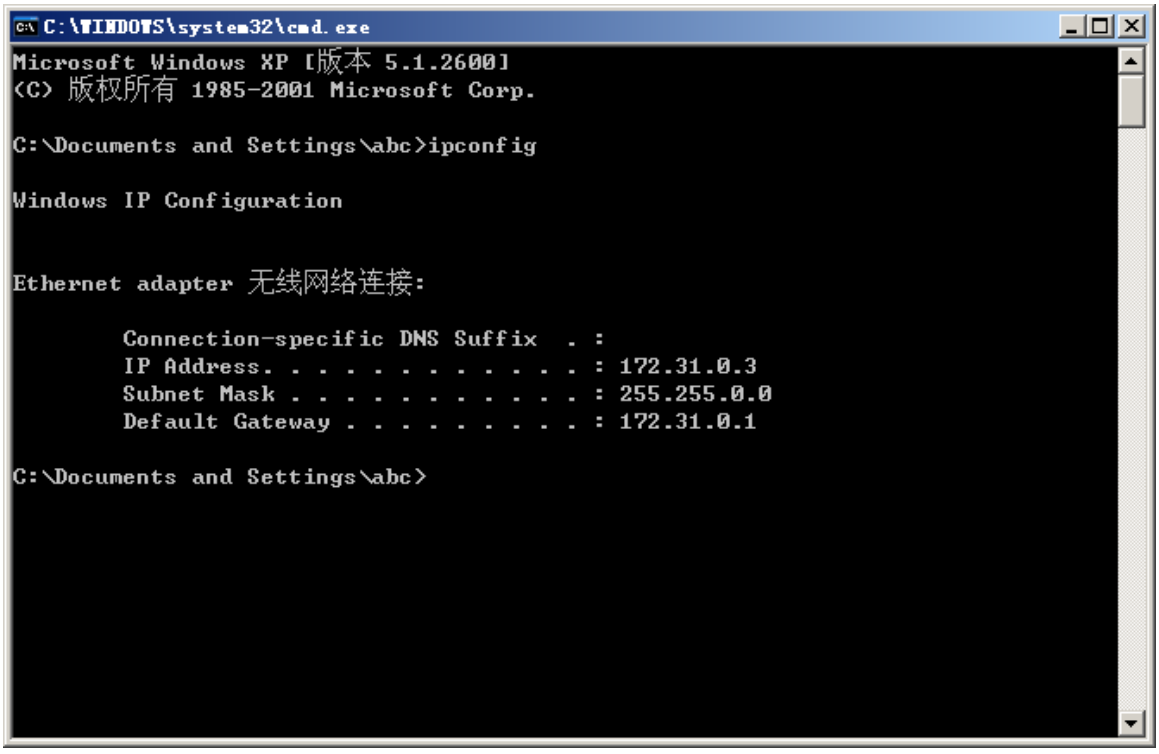
Computer/PC ----------Work as SERVER

✓ Make sure the Discover Wi-Fi module is in AP mode, now press ”3” to get the assigned IP address:

```
3
-SnicInit

Interface Type? (0: STA  1: AP)
1
```

```
Interface Type? (0: STA  1: AP)
1
-SnicGetDhcp
.
IP assigned as 172.31.0.1
.
```

✓ Connect the computer/PC to the Murata Wi-Fi AP, and get the IP address: 172.31.0.3 in this case.



✓ On the PC, first run the testserver application: **Setup_server.exe** (or testserver.exe) and setup the computer/PC as a UDP SERVER by selecting option 3 from the menu:

```
G:\Discover Wi-Fi\Software>testserver.exe 172.31.0.3 2222
----------------------
0 TCP client
1 TCP server
2 Send from sock
3 UDP server
4 UDP send to soft AP


q Quit
----------------------
```

```
G:\Program Files\Murata\testserver>testserver.exe 172.31.0.3 2222
----------------------
0 TCP client
1 TCP server
2 Send from sock
3 UDP server
4 UDP send to soft AP


q Quit
----------------------
```

```
----------------------
Create a UDP socket and start recv on port 0x10e1.
----------------------
```

- ✓ Setup the STM32F4 Wi-Fi as UDP CLIENT by selecting 9 from the main Menu on Hyper-terminal, you will be displayed with the SERVER IP address and port number.

```
Enter server IP to connect:
172.31.0.3
Enter server port number:
2222
```

- ✓ After that, the CLIENT will send 10 UDP packets to the SERVER:

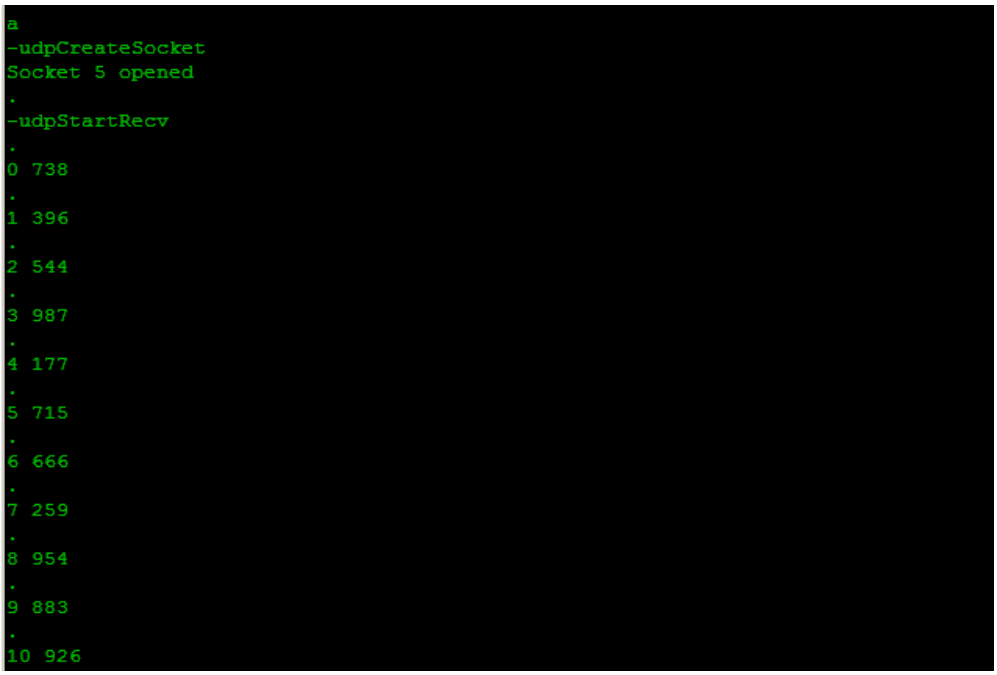
```
Send 10
-udpSendFromSock
.
0 1
-udpSendFromSock
.
1 2
-udpSendFromSock
.
2 3
-udpSendFromSock
.
3 4
-udpSendFromSock
.
4 5
-udpSendFromSock
.
5 6
-udpSendFromSock
.
6 7
-udpSendFromSock
.
7 8
-udpSendFromSock
.
8 9
-udpSendFromSock
.
9 10
-closeSocket
Socket 5 closed
.
```

✓ The SERVER will acknowledge by displaying received as below:

```
0
Recv: 1
1
Recv: 2
2
Recv: 3
3
Recv: 4
4
Recv: 5
5
Recv: 6
6
Recv: 7
7
Recv: 8
8
Recv: 9
9
Recv: 10
```

✓ Now press ”a” on the CLIENT (STM32F4 Wi-Fi) side, and press ”4” on the SERVER (computer/PC) side to send 100 UDP packets to testclient.

- ✓ Hyper-terminal will display the receving confirmation of 100UDP packets on CLIENT side.



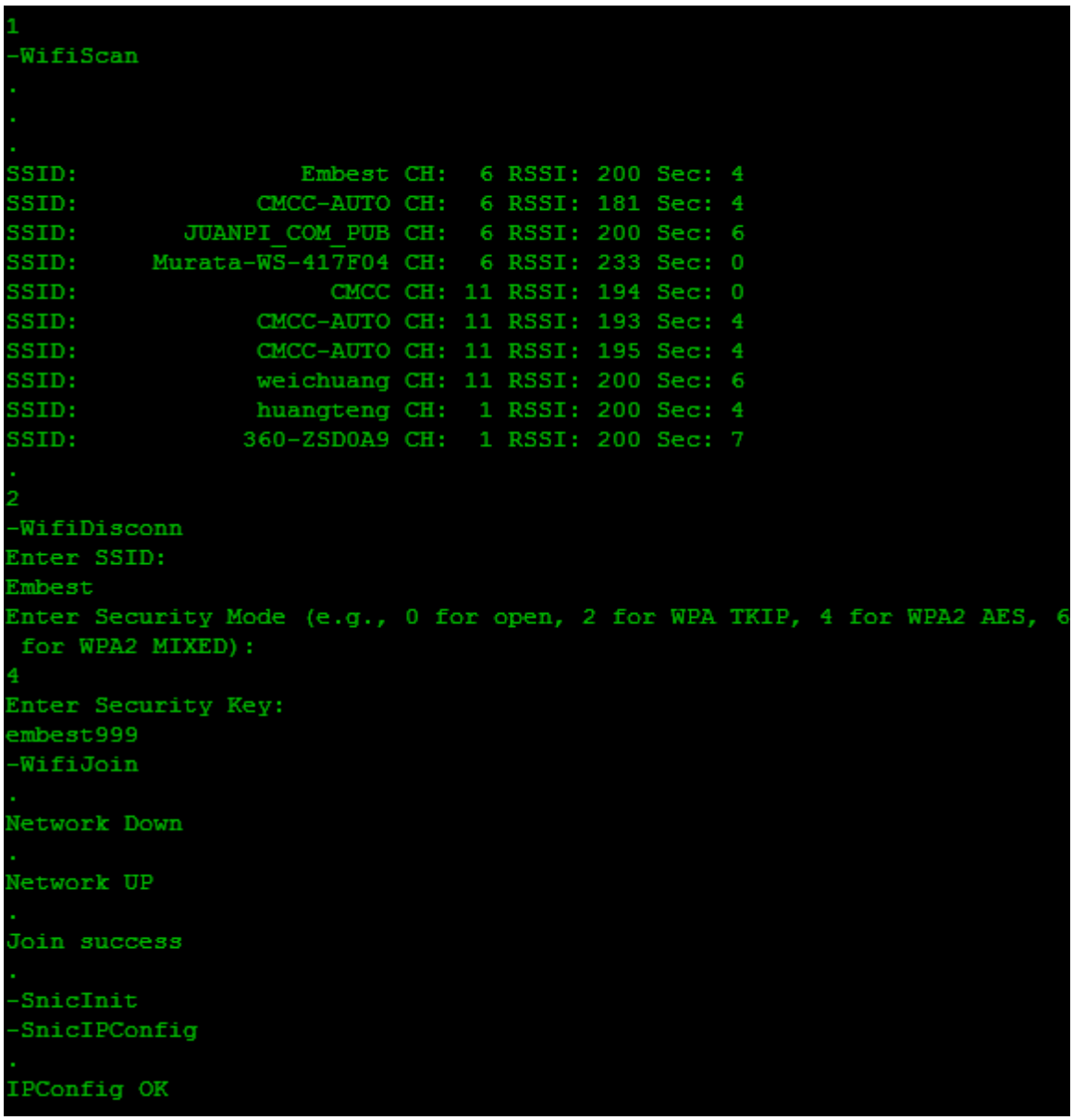
✓ Once the tests are finished you can press 'q' to terminate the program and the terminal will show as below:

4.4 HTTP Extended Function

Note: Here we used EVK1 and EVK2 to do the extended function tests. EVK2 is mainly used as an HTTP or HTTPS server, if you do not have another Discover Wi-Fi, you can setup your own server to finish the test yourself.

4.4.1 HTTP Get

If the network has internet access, any valid domain name can be used. And you only require EVK1 to do this test. First connect the EVK1 to the available wireless network. You can refer to ection 4.3.1 for the basic connection operation. Here we use the "Embest" network and connect successfully as below:



Press “d”, and input the server name. You can input any server address, here we have used www.murata-ws.com as an example.

```
d
Enter server name:  www.murata-ws.com
www.murata-ws.com
```

```
d
Enter server name:  www.murata-ws.com
www.murata-ws.com
Enter URI after the server name: ([CR] to accept .)
/
```

The terminal will then display the websites source code as below.

```
HTTP RSP code: 400, seq#: 4, Content Length: 332, Type: text/html; charset=
a: no
Content:
        <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
                                                    <html><head>
                                                              <tit
title>
     </head><body>
```

…

```
                    </body>
                          </html>

.

HTTP RSP indication, seq#: 6, content length: 0, More data: no
Content:
```

4.4.2 HTTP Post JSON Object

This test is to demonstrate communication with a SN82xx soft AP via JSON object which requires two SN82xx EVKs.

EVK2 starts as a soft AP (SSID=Murata-WS-417171) and EVK1’s STA interface joins the soft AP. EVK1 then queries for EVK2’s STA status.

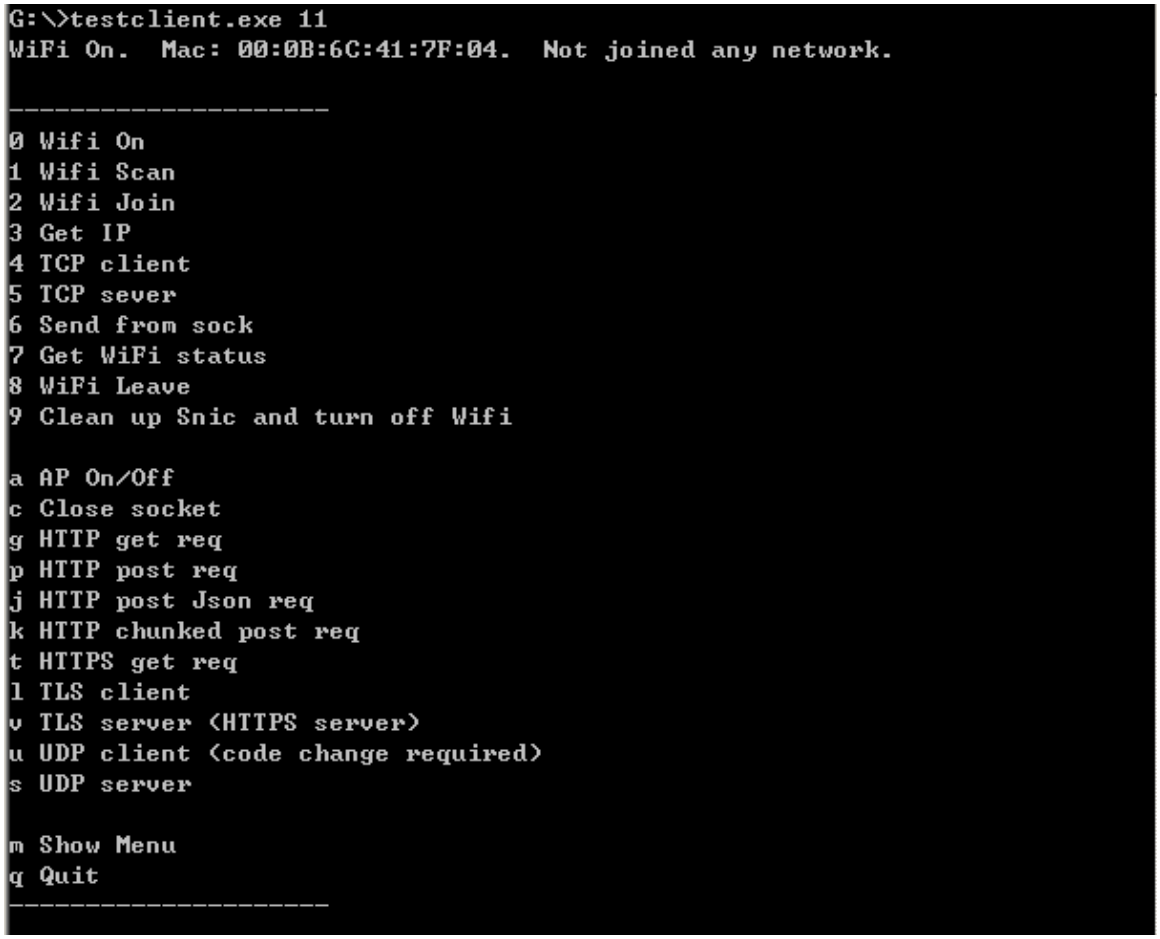
On EVK2’s host PC(1) Windows command prompt, run following:

Testclient <COM port>

In our case, the command is

```
G:\Program Files\Murata\Testclient>testclient.exe 11
```

The following menu should be displayed



Then make the EVK2 join the wireless network (in this case “Embest”).

```
SSID:               Embest CH:  6 RSSI: -56 Sec: 4
SSID:                 CMCC CH: 11 RSSI: -56 Sec: 0
SSID:           CMCC-AUTO CH: 11 RSSI: -56 Sec: 4
SSID:           CMCC-AUTO CH: 11 RSSI: -57 Sec: 4
SSID:      JUANPI_COM_PUB CH:  6 RSSI: -56 Sec: 6
SSID:                 CMCC CH:  6 RSSI: -74 Sec: 0
SSID:           CMCC-AUTO CH:  6 RSSI: -74 Sec: 4
SSID:      TP-LINK_3F25A8 CH:  6 RSSI: -56 Sec: 4
SSID:  INTEST_Meeting-Room CH:  6 RSSI: -75 Sec: 4
SSID:          HPYJ'S WIFI CH:  6 RSSI: -56 Sec: 4
SSID:           weichuang CH: 11 RSSI: -56 Sec: 6
SSID:       JUANPI_COM_HR CH: 11 RSSI: -68 Sec: 4
SSID:          360-ZS846F CH:  1 RSSI: -56 Sec: 4
SSID:          360-ZSD0A9 CH:  1 RSSI: -56 Sec: 7
SSID:          360-ZS8978 CH:  1 RSSI: -56 Sec: 7
SSID:          360-ZSBD5E CH:  1 RSSI: -56 Sec: 7
Enter SSID:
Embest
Enter Security Mode (e.g., 0 for open, 2 for WPA TKIP, 4 for WPA2 AES):
4
Enter Security Key:
embest999
-Network UP
-Join success
IPConfig OK
WiFi On.  Mac: 00:0B:6C:41:7F:04.  Joined SSID: Embest
```

Press the "a" key on the PC, and choose the AP mode.

```
-AP status
AP is OFF
-AP status
AP is ON
```

Press "1" to scan for WiFi networks on EVK1. You can find the EVK1's AP SSID.

```
-WifiScan
.
.
.
.
.
SSID:     Murata-WS-417F04 CH:  6 RSSI: 238 Sec: 0
SSID:              Embest CH:  6 RSSI: 200 Sec: 4
SSID:       JUANPI_COM_PUB CH:  6 RSSI: 200 Sec: 6
SSID:                CMCC CH: 11 RSSI: 196 Sec: 0
SSID:           CMCC-AUTO CH: 11 RSSI: 195 Sec: 4
SSID:           CMCC-AUTO CH: 11 RSSI: 195 Sec: 4
SSID:       TP-LINK_3F25A8 CH:  6 RSSI: 200 Sec: 4
SSID:          HPYJ'S WIFI CH:  6 RSSI: 200 Sec: 4
SSID:        JUANPI_COM_HR CH: 11 RSSI: 192 Sec: 4
SSID:           weichuang CH: 11 RSSI: 200 Sec: 6
SSID:           360-ZSBD5E CH:  1 RSSI: 200 Sec: 7
SSID:           huangteng CH:  1 RSSI: 200 Sec: 4
SSID:           360-ZS8978 CH:  1 RSSI: 200 Sec: 7
SSID:           hikvision CH:  6 RSSI: 200 Sec: 6
SSID:  INTEST_Meeting-Room CH:  6 RSSI: 181 Sec: 4
SSID:                CMCC CH:  6 RSSI: 183 Sec: 0
SSID:           CMCC-AUTO CH:  6 RSSI: 182 Sec: 4
.
```

Then join it.

```
2
-WifiDisconn
Enter SSID:
Murata-WS-417F04
Enter Security Mode (e.g., 0 for open, 2 for WPA TKIP, 4 for WPA2 AES, 6
 for WPA2 MIXED):
0
-WifiJoin
.
Network Down
.
Network UP
.
Join success
.
-SnicInit
-SnicIPConfig
.
IPConfig OK
.
```

Get the EVK1’s IP address:

```
3
-SnicInit

Interface Type? (0: STA  1: AP)
0
-SnicGetDhcp
.
IP assigned as 172.31.0.3
.
```

Then input “f” to test the http post JSON object.

```
f
Make sure STA is connected to SN8200 soft AP.
Enter server name: ([CR] to accept sn8200.com)
```

Input the server name ”sn8200.com” when required.

```
f
Make sure STA is connected to SN8200 soft AP.
Enter server name: ([CR] to accept sn8200.com)
sn8200.com
Enter URI after the server name: ([CR] to accept /sws/wifi/stat)
```

Then input “/sws/wifi/stat” when prompted.

```
Enter URI after the server name: ([CR] to accept /sws/wifi/stat)
/sws/wifi/stat
```

Then input ”Params=%7B%22if%22%3A%22sta%22%7D&callback=jsonp1363359950547” when asked. (you can copy and paste this into the terminal)

```
Enter content to POST: ([CR] to accept Params=%7B%22if%22%3A%22sta%22%7D
&callback=jsonp1363359950547)
Params=%7B%22if%22%3A%22sta%22%7D&callback=jsonp1363359950547
```

The information is then displayed as below.

```
HTTP RSP code: 200, seq#: 13, Content Length: 64, Type: application/json
, More data: no
Content:
        jsonp1363359950547({"s":2,"mac":"000B6C417F04","ssid":"Embest"}
)
```

The response contains a JSON object showing EVK2’s STA interface is connected to an AP named “Embest”.

4.4.3 Http Post/Http Chunked Post

Http post

If the network has internet access, any valid domain name can be used. In this example, EVK2 is setup as a web server, and it contains an index.html file. (If you do not have EVK2, you can setup a local web server yourself)

Firstly make the EVK1 and EVK2 join the same wireless network.

```
SSID:                Embest CH:  6 RSSI: -56 Sec: 4
SSID:                  CMCC CH: 11 RSSI: -56 Sec: 0
SSID:             CMCC-AUTO CH: 11 RSSI: -56 Sec: 4
SSID:             CMCC-AUTO CH: 11 RSSI: -57 Sec: 4
SSID:        JUANPI_COM_PUB CH:  6 RSSI: -56 Sec: 6
SSID:                  CMCC CH:  6 RSSI: -74 Sec: 0
SSID:             CMCC-AUTO CH:  6 RSSI: -74 Sec: 4
SSID:        TP-LINK_3F25A8 CH:  6 RSSI: -56 Sec: 4
SSID:  INTEST_Meeting-Room CH:  6 RSSI: -75 Sec: 4
SSID:           HPYJ'S WIFI CH:  6 RSSI: -56 Sec: 4
SSID:             weichuang CH: 11 RSSI: -56 Sec: 6
SSID:         JUANPI_COM_HR CH: 11 RSSI: -68 Sec: 4
SSID:            360-ZS846F CH:  1 RSSI: -56 Sec: 4
SSID:            360-ZSD0A9 CH:  1 RSSI: -56 Sec: 7
SSID:            360-ZS8978 CH:  1 RSSI: -56 Sec: 7
SSID:            360-ZSBD5E CH:  1 RSSI: -56 Sec: 7
Enter SSID:
Embest
Enter Security Mode (e.g., 0 for open, 2 for WPA TKIP, 4 for WPA2 AES):
4
Enter Security Key:
embest999
-Network UP
-Join success
IPConfig OK
WiFi On.  Mac: 00:0B:6C:41:7F:04.  Joined SSID: Embest
```

```
1
-WifiScan
.
.
.
SSID:               Embest CH:  6 RSSI: 200 Sec: 4
SSID:           CMCC-AUTO CH:  6 RSSI: 181 Sec: 4
SSID:       JUANPI_COM_PUB CH:  6 RSSI: 200 Sec: 6
SSID:    Murata-WS-417F04 CH:  6 RSSI: 233 Sec: 0
SSID:                CMCC CH: 11 RSSI: 194 Sec: 0
SSID:           CMCC-AUTO CH: 11 RSSI: 193 Sec: 4
SSID:           CMCC-AUTO CH: 11 RSSI: 195 Sec: 4
SSID:           weichuang CH: 11 RSSI: 200 Sec: 6
SSID:           huangteng CH:  1 RSSI: 200 Sec: 4
SSID:          360-ZSD0A9 CH:  1 RSSI: 200 Sec: 7
.
2
-WifiDisconn
Enter SSID:
Embest
Enter Security Mode (e.g., 0 for open, 2 for WPA TKIP, 4 for WPA2 AES, 6
 for WPA2 MIXED):
4
Enter Security Key:
embest999
-WifiJoin
.
Network Down
.
Network UP
.
Join success
.
-SnicInit
-SnicIPConfig
.
IPConfig OK
```

On EVK2, press ”3” to get its IP address.

```
Interface Type? (0: STA  1: AP)
0
IP assigned as 192.168.2.121
```

Press”5”, and input the port number 80. Then EVK1 will be configured as an Http server:

```
Enter server port number to set: (CR] to accept 80)
80
Socket 4 opened
```

Turn to EVK1, and input “e” to connect to the server. Input the server’s IP address.

```
e
Enter server name: ([CR] to accept www.murata-ws.com)
192.168.2.121
```

```
Enter server name: ([CR] to accept www.murata-ws.com)
192.168.2.121
Enter URI after the server name: ([CR] to accept /)
/
```

Input "abcd"

```
Enter server name: ([CR] to accept www.murata-ws.com)
192.168.2.121
Enter URI after the server name: ([CR] to accept /)
/
Enter content to POST:
abcd
```

Then the terminal will show the server's content:

```
HTTP RSP code: 200, seq#: 7, Content Length: 47, Type: text/html, More data
: no
Content:
        <html><body>Hello from SN8200 # 0</body></html>
```

And the server (EVK2) will display the client's information:

```
Accepted connection from 192.168.2.122
Connection socket: 5
112 bytes received from socket 5
POST / HTTP/1.1
Host: 192.168.2.121
Content-Length: 4
Content-Type: text/html
Accept-Language: en-US

abcdpkt 2, 134 bytes sent
```

Http chunked post

This example shows how to post bigger data with chunk encoding. After you finished the Http Post, you can:

Choose "g" on EVK2 to test the http chunked post, and input the server's IP address

```
g
Enter server name (or the peer testclient IP, peer testclient should start
TCP server on port 80): ([CR] to accept 192.168.10.100)
192.168.2.121
```

It will then display the server's content:

```
HTTP RSP code: 0, seq#: 25
.

HTTP more RSP code: 200, seq#: 26, Content Length: 47, Type: text/html,
More data: no
Content:
        <html><body>Hello from SN8200 # 0</body></html>
```

```
Accepted connection from 192.168.2.138
Connection socket: 5
248 bytes received from socket 5
POST /rest/thermostatGetTime HTTP/1.1
Host: 192.168.2.140
Content-Type: application/x-www-form-urlencoded
Transfer-Encoding: chunked
Accept: text/html,application/xml
Accept-Language: en-US

2C
mcu_serial_number_hex=00112233445566778899AA
pkt 3, 134 bytes sent
46 bytes received from socket 5
28
&username=MyUsername&password=MyPassword
5 bytes received from socket 5
0

Socket 5 closed
```

4.4.4 Https Server/Client

Firstly make the EVK1 and EVK2 join the same wireless network.

```
1
-WifiScan
.
.
.
SSID:               Embest CH:  6 RSSI: 200 Sec: 4
SSID:           CMCC-AUTO CH:  6 RSSI: 181 Sec: 4
SSID:       JUANPI_COM_PUB CH:  6 RSSI: 200 Sec: 6
SSID:    Murata-WS-417F04 CH:  6 RSSI: 233 Sec: 0
SSID:                CMCC CH: 11 RSSI: 194 Sec: 0
SSID:           CMCC-AUTO CH: 11 RSSI: 193 Sec: 4
SSID:           CMCC-AUTO CH: 11 RSSI: 195 Sec: 4
SSID:           weichuang CH: 11 RSSI: 200 Sec: 6
SSID:           huangteng CH:  1 RSSI: 200 Sec: 4
SSID:          360-ZSD0A9 CH:  1 RSSI: 200 Sec: 7
.
2
-WifiDisconn
Enter SSID:
Embest
Enter Security Mode (e.g., 0 for open, 2 for WPA TKIP, 4 for WPA2 AES, 6
 for WPA2 MIXED):
4
Enter Security Key:
embest999
-WifiJoin
.
Network Down
.
Network UP
.
Join success
.
-SnicInit
-SnicIPConfig
.
IPConfig OK
```

https get

This example shows how to send a HTTPS get request on a locally setup, test HTTPS server.

In this example, EVK2 is setup as a web server, and it contains an index.html file. (If you do not have EVK2, you can setup a local web server yourself)The HTTPS server contains an index.html file.

Press ”3” to get its IP address:

```
Interface Type? (0: STA  1: AP)
0
IP assigned as 192.168.2.121
```

Press ”v” to configure the EVK1 as a HTTPS server:

```
Enter server port number to set: ([CR] to accept 443)
443
Socket 5 opened
```

Turn to EVK2, and input "e" to connect to the server. Input the server's IP address.

```
h
Enter server name: ([CR] to accept 192.168.2.121)
192.168.2.121
```

```
h
Enter server name: ([CR] to accept 192.168.2.121)
192.168.2.121
Enter URI after the server name: ([CR] to accept /)
/
```

The terminal will then show the server's content:

```
HTTP RSP code: 200, seq#: 16, Content Length: 47, Type: text/html, More dat
a: no
Content:
        <html><body>Hello from SN8200 # 3</body></html>
```

And the EVK2 will display the HTTPS client's information:

```
Accepted connection from 192.168.2.122
Connection socket: 6
39 bytes received from socket 6
GET / HTTP/1.1
Host: 192.168.2.121

pkt 4, 134 bytes sent
Socket 6 closed
```

TLS client test

This example shows how to create a TLS socket, connect to a TLS server socket; send and receive data. When connecting to port 443, it is similar to the HTTPS get above. An HTTPS server is running on EVK2, with index.html contained.
Input "i" on EVK1, to connect to the HTTP server, and choose port 443:

```
i
-tcpCreateSocket
Socket 4 opened
.
Enter server IP to connect:
192.168.2.121
```

```
Enter server port number:
443
```

Then the terminal will show the connect information:

```
-tcpConnectToServer
.
Socket connection UP
.
-sendFromSock
pkt 1, 58 bytes sent
.
134 bytes received from socket 4
HTTP/1.1 200 OK
Content-Type: text/html
Connection: close
Content-Length: 47

<html><body>Hello from SN8200 # 4</body></html>

-closeSocket
Socket 4 closed
.
Socket closed
```

And the EVK2 will show the client's information:

```
Accepted connection from 192.168.2.122
Connection socket: 6
58 bytes received from socket 6
GET / HTTP/1.1
Host: 192.168.2.125
Accept: text/html

pkt 5, 134 bytes sent
Socket 6 closed
```

TLS server test

This example shows how to create a bound TLS socket, and listen on a port. When listening to port 443, it is actually a HTTPS server.
Press "3" to get the EVK1's IP address:

```
Interface Type? (0: STA  1: AP)
0
-SnicGetDhcp
.
IP assigned as 192.168.2.122
```

Then press "j" to configure the EVK1 as a TLS server. Set the port number as 443

```
j
Enter server port number to set:
443
```

```
Enter server port number to set:
443
-tcpCreateSocket
Socket 4 opened
.
-tcpCreateConnection
```

Then test the TLS server.

Press "l" on EVK2, and input the server's IP address and port number. Then the EVK1 will be configured as a TLS client

```
Enter server IP to connect: <[CR] to accept 192.168.10.101>
192.168.2.122
Enter server port number: <[CR] to accept 443>
443
```

Then the server will display the client's information:

```
Accepted connection from 192.168.2.112
Connection socket: 5
.
128 bytes received from socket 5
GET /favicon.ico HTTP/1.1
Host: 192.168.2.122
Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 5.1) Ap-sendFromSock
128 bytes received from socket 5
GET /favicon.ico HTTP/1.1
Host: 192.168.2.122
Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 5.1) Ap-sendFromSock
128 bytes received from socket 5
GET /favicon.ico HTTP/1.1
Host: 192.168.2.122
Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 5.1) Ap-sendFromSock
128 bytes received from socket 5
GET /favicon.ico HTTP/1.1
Host: 192.168.2.122
Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 5.1) Ap-sendFromSock
128 bytes received from socket 5
GET /favicon.ico HTTP/1.1
Host: 192.168.2.122
Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 5.1) Ap-sendFromSock
128 bytes received from socket 5
GET /favicon.ico HTTP/1.1
Host: 192.168.2.122
Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 5.1) Ap-sendFromSock
```

Section 5 Webserver Customization

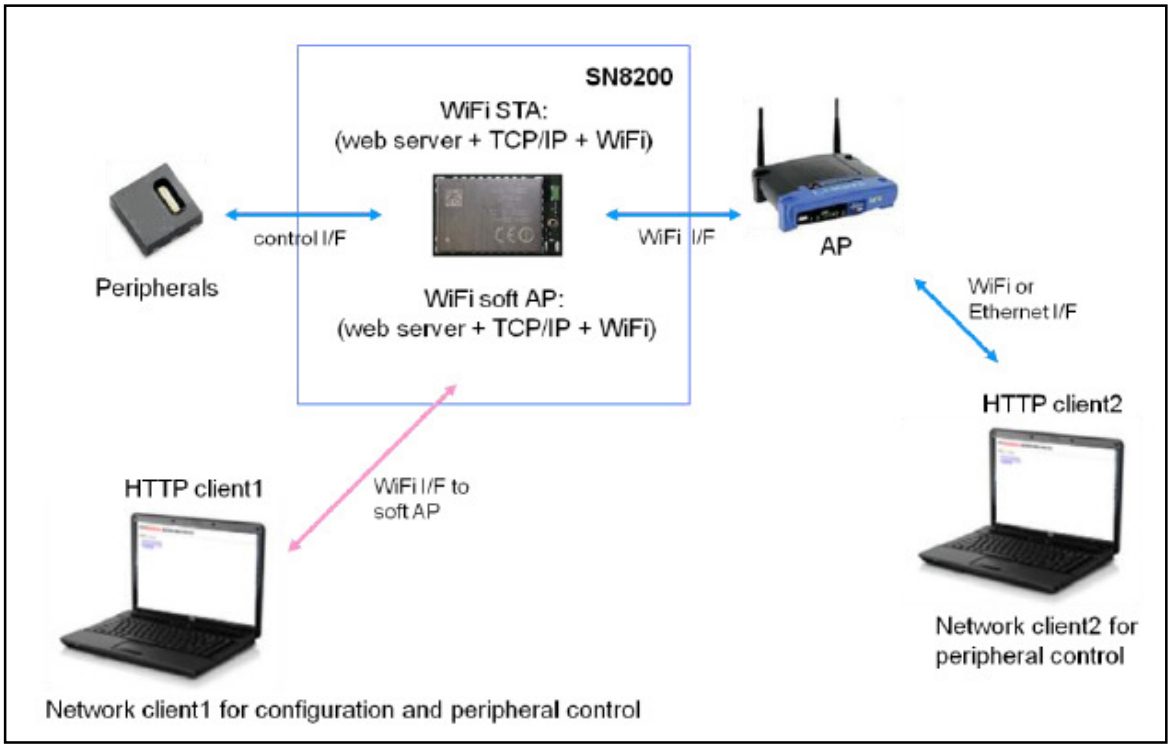


Figure 4-4 EZ Web Wizzard Solution

Developers can develop their own firmware including webserver contents based on Murata EZ Web Wizzard Solution, Murata EZ Web Wizzard (EWW) software supports easy custom web-based control to save cost on additional host microcontrollers. For more information please refer to the URL below:

http://www.murata-ws.com/sn8200.htm

Section 6 Safety Instructions

Please note that the Discover Wi-Fi board is supplied without any casing/box, all the components are exposed. Therefore, extra precautions must be taken for ESD (electrostatic discharge) to make sure that there is no static interference when using this board. Appropriate ESD protections must be taken and wearing electrostatic discharge protection equipment is recommended such as an anti-static wristband.

ESD damage can range from subtle performance degradation to complete device failure. Precision IC's may be more susceptible to damage because very small parametric changes could cause the device to fail its defined specifications.

Warning:

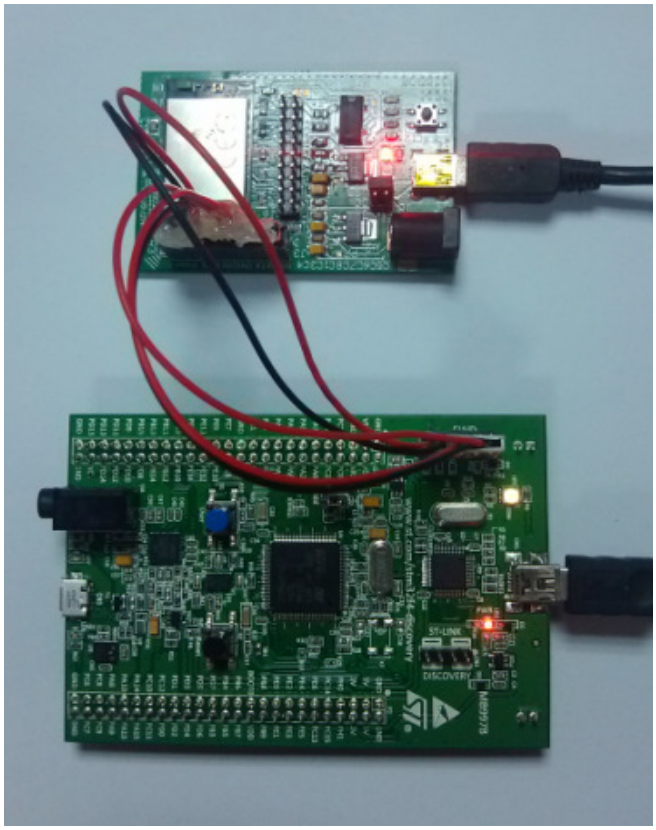
This is a class B product, this product may cause radio interference in which case users may be required to take adequate measures.

Appendix Firmware Update

1 Hardware connection

- Power on the STM32DISCOVERY board and DISCOVERY Wi-Fi board. (using the same PC)
- Jumper settings: open the STLINK jumpers on the STM32FDISCOVERY Board.
- ST-link connects the Wi-Fi board JTAG surface.

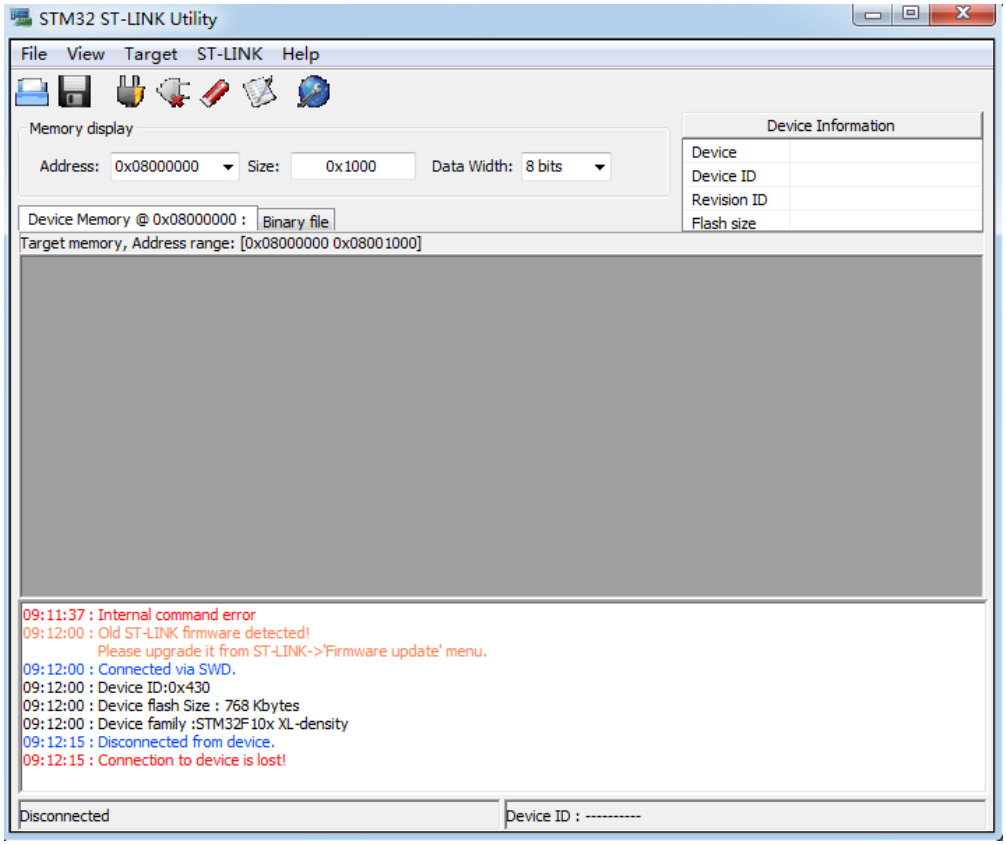
	CN2(STM32F4DISCOVERY SWD)	J3(DISCOVERY Wi-Fi)
GND	1	20
JTCK	2	9
GND	3	18
JTMS	4	7
NRST	5	15
JTD0	6	13



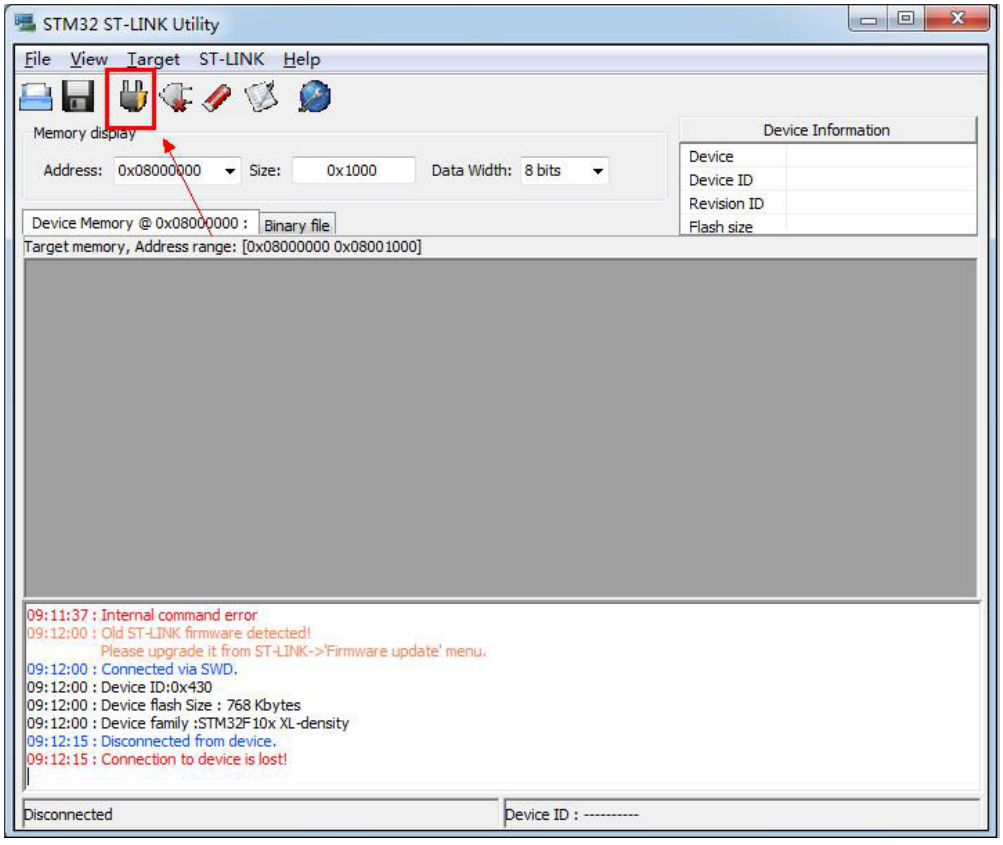
2 Running STM32 ST-LINK Utility

- Install the STM32 ST-LINK Utility_v2.4.0 (has been tested on windows XP 32 bit PC)

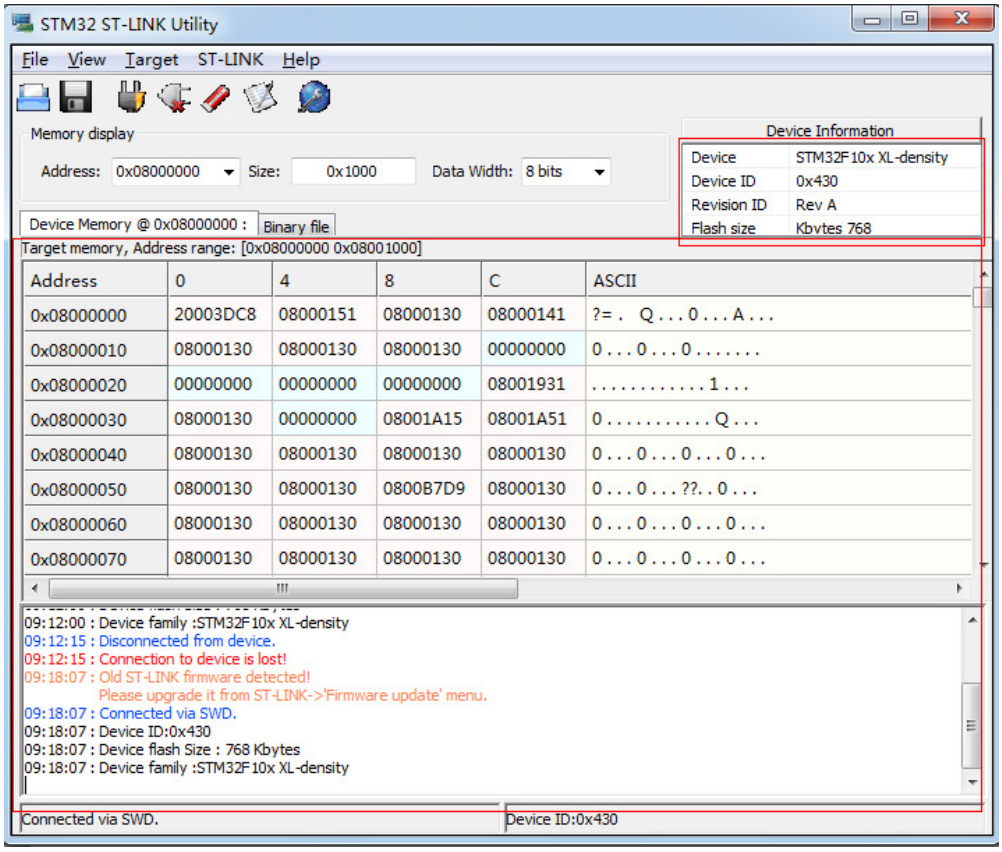
- Open STM32 ST-LINK Utility.



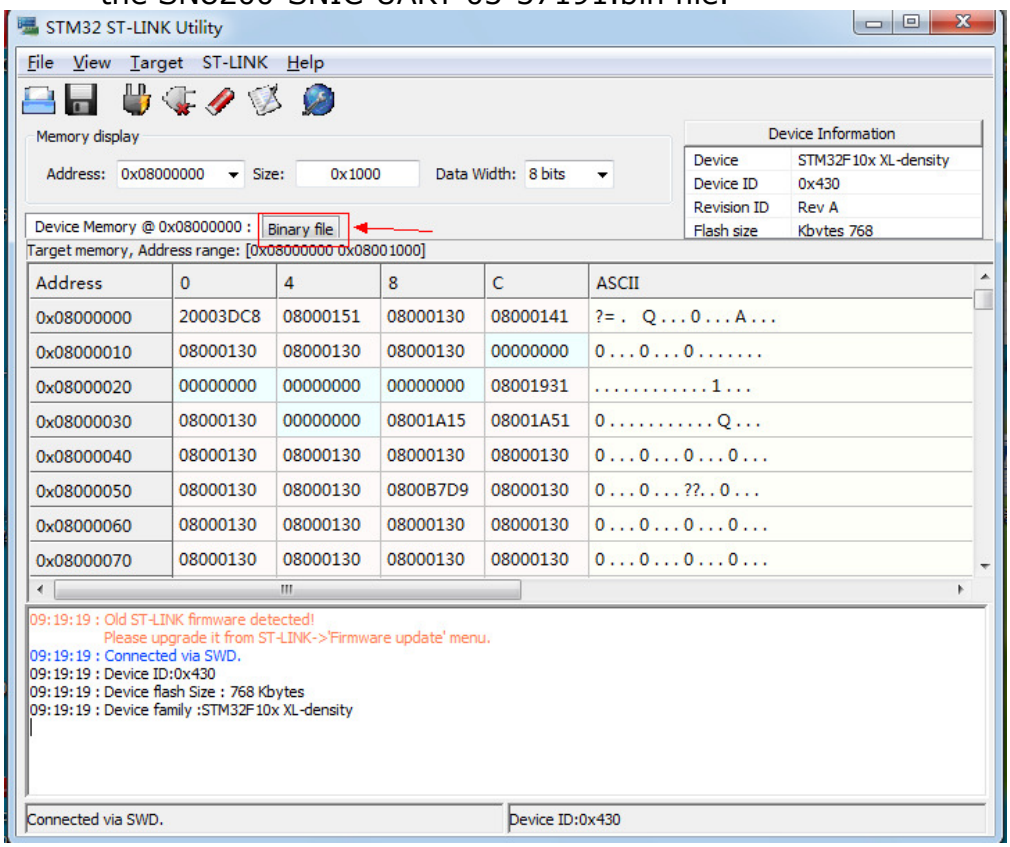
- Connect the board



- Click the icon highlighted in the image above and if properly connected the following will be displayed.

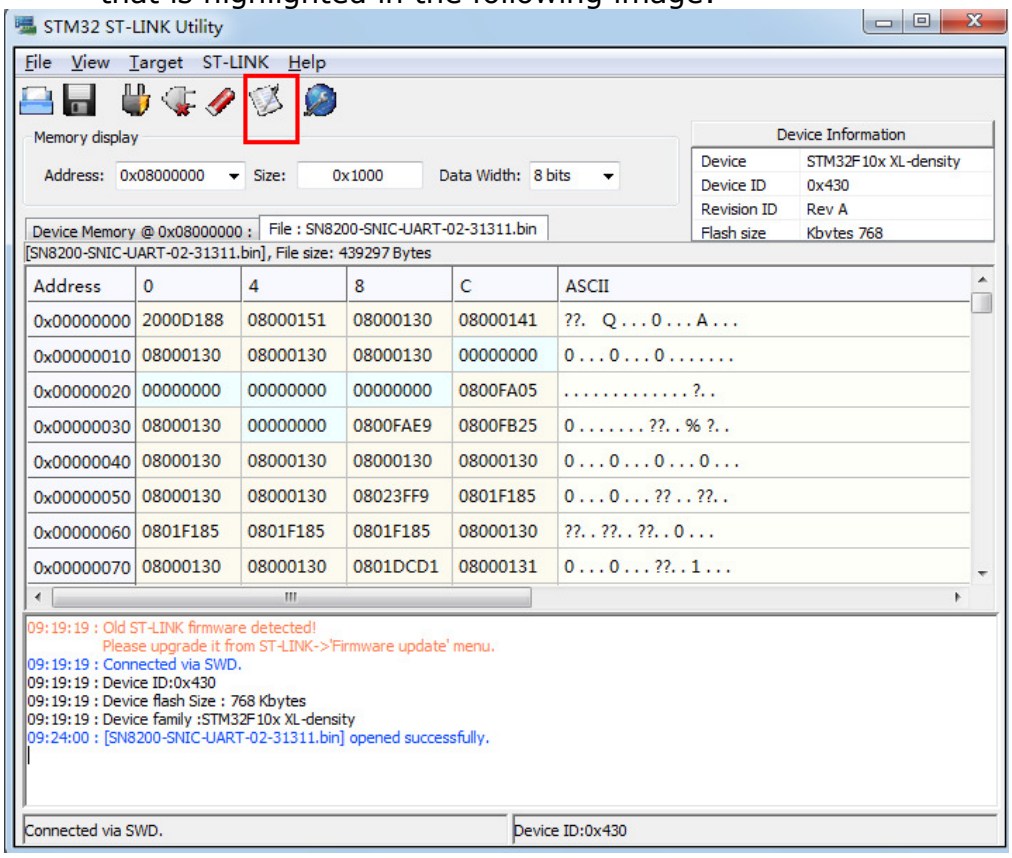


- Load the .bin file. Click the icon as shown in the following image and load the SN8200-SNIC-UART-03-37191.bin file.

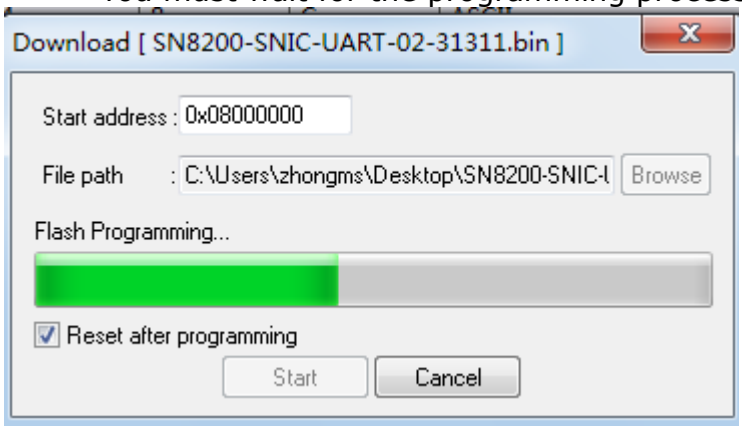


- Once loading is complete you can program the device by clicking the icon

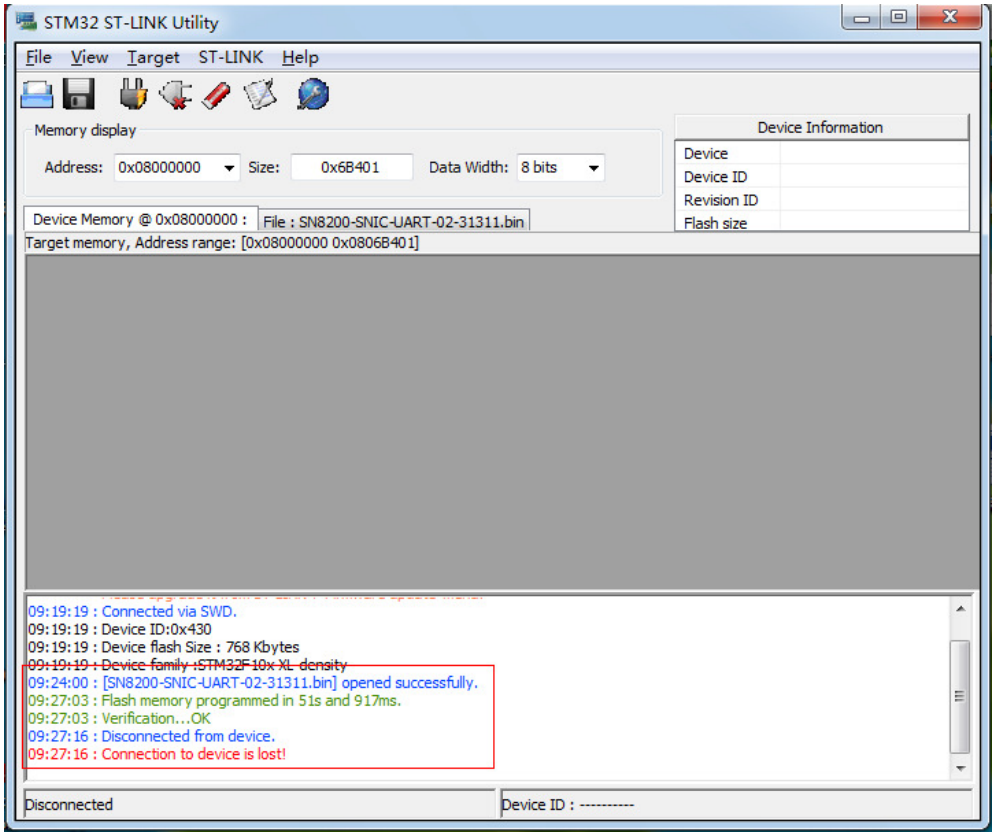
that is highlighted in the following image:



- You must wait for the programming process to complete



- The information highlighted below will be displayed upon successful completion of the process.


STM32 ST-LINK Utility
File View Target ST-LINK Help
Memory display
Address: 0x08000000 Size: 0x6B401 Data Width: 8 bits
Device Information
Device
Device ID
Revision ID
Flash size
Device Memory @ 0x08000000 :
File : SN8200-SNIC-UART-02-31311.bin
Target memory, Address range: [0x08000000 0x0806B401]
09:19:19 : Connected via SWD.
09:19:19 : Device ID:0x430
09:19:19 : Device flash Size : 768 Kbytes
09:19:19 : Device family :STM32F10x XL density
09:24:00 : [SN8200-SNIC-UART-02-31311.bin] opened successfully.
09:27:03 : Flash memory programmed in 51s and 917ms.
09:27:03 : Verification...OK
09:27:16 : Disconnected from device.
09:27:16 : Connection to device is lost!
Disconnected
Device ID : ----------