Concurrent Mode Application

User guide

Version 0.2

May 2016

Redpine Signals, Inc.
2107 N. First Street, #540
San Jose, CA 95131.
Tel: (408) 748-3385
Fax: (408) 705-2019
Email: info@redpinesignals.com
Website: www.redpinesignals.com

About this Document

This document describes the process of bringing up the RS9113 based module in concurrent (both STA and AP) mode.

Disclaimer:

The information in this document pertains to information related to Redpine Signals, Inc. products. This information is provided as a service to our customers, and may be used for information purposes only. Redpine assumes no liabilities or responsibilities for errors or omissions in this document. This document may be changed at any time at Redpine's sole discretion without any prior notice to anyone. Redpine is not committed to updating this document in the future.

Table of Contents

Table of Figures

Table of Tables

No table of figures entries found.

1 Introduction

This project is applicable to all the WiSeConnect variants like WiSeConnect Plus, WiSeMCU and WyzBee. The term WiSeConnect refers to its appropriate variant.

1.1 Application Overview

1.1.1 Overview

This application demonstrates how to configure the device in both WiFi Station mode and Access point mode and how to transfer data on both modes.

In this Application, WiSeConnect device starts as Access Point and connects with Access point in station mode. After successful creation of Access Point and successful connection with Access point, Application opens TCP socket and transfers TCP data in station mode and device responds for the Ping request sent by connected station with Ping Reply in Access Point mode.

1.1.2 Sequence of Events

This Application explains user how to:

- Create WiSeConnect device as Soft Access point
- Connect with 3rd party Access Point in Station mode
- Open TCP server socket on configured port number on the device.
- Send TCP data to remote peer in station mode

1.2 Application Setup

The WiSeConnect in its many variants supports SPI and UART interfaces. Depending on the interface used, the required set up is as below:

1.2.1 SPI based Setup Requirements

- Windows PC1 with CooCox IDE
- Spansion (MB9BF568NBGL) micro controller

Note: If user does not have Spansion (MB9BF568NBGL) host platform, please go through the SPI-Porting guide **\sapis\docs\RS9113-WiSeConnect-SAPI-Porting-Guide-vx.x.pdf** for SAPIs porting to that particular platform.

- WiSeConnect device
- Access point
- Windows PC2
- A Mobile device as a Wi-Fi station (This example uses a windows Laptop)
- A TCP server application running on the Wi-Fi station (This example uses iperf for windows)

1.2.2 UART/USB-CDC based Setup Requirements

- Windows PC1 with Dev-C++ IDE
- WiSeConnect device
- Access point
- Windows PC2
- A Mobile device as a Wi-Fi station (This example uses a windows Laptop)

- A TCP server application running on the Wi-Fi station (This example uses iperf for windows)

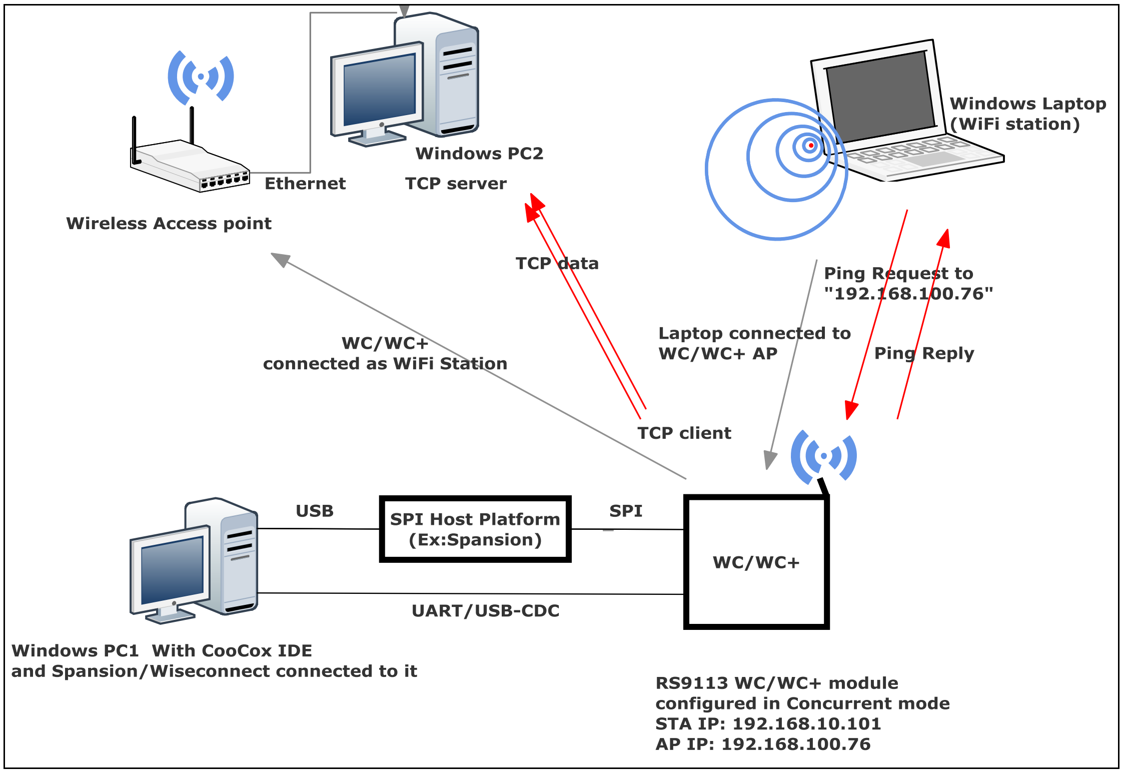


Figure 1: Setup Diagram

2 Configuration and Execution of the Application

The example application is available in the Release at **{Release $}/host/sapis/examples.** These examples will have to be initialized, configured and executed to test the application. The initialization varies based on the interface but configuration and execution are the common.

2.1 Initializing the Application

2.1.1 SPI Interface

If User using SPI interface, Please refer the document ***sapis/platforms/spansion_MB9BF568NBGL/RS9113-WiSeConnect_SAPIS_Spansion_Project_User_guide.pdf*** for opening the ***concurrent_mode*** example in CooCox IDE.

2.1.2 UART/USB-CDC Interface

If User using UART interface, Please refer the document ***sapis/platforms/windows_uart/RS9113-WiSeConnect_SAPIS_Windows_Project_UserGuide.pdf*** for opening the ***concurrent_mode*** example in Dev-C++ IDE

2.2 Configuring the Application

1. Open ***sapis/examples/concurrent_mode/rsi_concurrent_mode.c*** file and update/modify following macros :

SSID refers to the name of the Access point.

```
#define SSID                        “REDPINE”
```

STA_SECURITY_TYPE refers to the type of security. In concurrent mode STA supports Open, WPA, WPA2 securities.

Valid configurations are:

RSI_OPEN - For OPEN security mode

RSI_WPA - For WPA security mode

RSI_WPA2 - For WPA2 security mode

```
#define STA_SECURITY_TYPE         RSI_OPEN
```

STA_PSK refers to the STA secret key to connect with the secured Access Point.

```
#define STA_PSK                       “”
```

AP_SSID refers to the name of the WiSeConnect Access point would be created.

```
#define AP_SSID                       “REDPINE_AP”
```

AP_CHANNEL_NO refers to the channel in which AP would be started

```
#define AP_CHANNEL_NO                11
```

Note1: Valid values for **CHANNEL_NO** are 1 to 11 in 2.4GHz band and 36 to 48 & 149 to 165 in 5GHz. In this example default configured band is 2.4GHz. So, if user wants to use 5GHz band then user has to set **RSI_BAND** macro to 5GHz band in ***sapis/include/rsi_wlan_config.h*** file.

Note2: In concurrent mode, STA and AP should be present in same channel. So, configure the AP_CHANNEL_NO to same channel in which Wireless Access point (to which WiSeConnect STA connects) exist.

AP_SECURITY_TYPE refers to the security type of the WiSeConnect Access Point. Access point supports OPEN, WPA-PSK, WPA2-PSK security modes.

Valid configuration is:

RSI_OPEN - For OPEN security mode

RSI_WPA - For WPA security mode

RSI_WPA2 - For WPA2 security mode

```
#define AP_SECURITY_TYPE          RSI_WPA2
```

AP_ENCRYPTION_TYPE refers to the type of Encryption method .Access point supports OPEN, TKIP, CCMP methods.

Valid configuration is:

RSI_CCMP - For CCMP encryption

RSI_TKIP - For TKIP encryption

RSI_NONE - For open encryption

```
#define AP_ENCRYPTION_TYPE        RSI_CCMP
```

AP_PSK refers to the secret key if the Access point to be configured in WPA/WPA2-PSK security modes.

```
#define AP_PSK                   “1234567890”
```

BEACON_INTERVAL refers to the time delay between two consecutive beacons in milliseconds in AP mode. Allowed values are integers from 100 to 1000 which are multiples of 100.

```
#define BEACON_INTERVAL        100
```

DTIM_INTERVAL refers DTIM interval of the Access Point. Allowed values are from 1 to 255.

```
#define DTIM_INTERVAL          4
```

DEVICE_PORT port refers internal TCP client port number

```
#define DEVICE_PORT              5001
```

REMOTE_PORT port refers remote TCP server port number which is opened in Windows PC2.

```
#define REMOTE_PORT              5001
```

SERVER_IP_ADDRESS refers remote peer (Windows PC2) IP address to connect with TCP server socket.

IP address should be in long format and in little endian byte order.

Example: To configure "192.168.0.100" as remote IP address, update the macro **SERVER_IP_ADDRESS** as **0x6400A8C0**.

```
      #define SERVER_IP_ADDRESS      0x6400A8C0
```

NUMEBR_OF_PACKETS refers how many packets to send from TCP client to TCP server

```
      #define NUMBER_OF_PACKETS     1000
```

To configure IP address in STA mode

DHCP_MODE refers whether IP address configured through DHCP or STATIC in STA mode

```
      #define DHCP_MODE   1
```

Note: If the user wants to configure STA IP address through DHCP then skip configuring the following **DEVICE_IP, GATEWAY** and **NETMASK** macros.

(Or)

If user wants to configure STA IP address through STATIC then set DHCP_MODE macro to "0" and configure following DEVICE_IP, GATEWAY and NETMASK macros.

IP address to be configured to the device in STA mode should be in long format and in little endian byte order.

Example: To configure "192.168.0.10" as IP address, update the macro **DEVICE_IP** as **0x010AA8C0**.

```
      #define  DEVICE_IP          0X0A00A8C0
```

IP address of the gateway should also be in long format and in little endian byte order.

Example: To configure "192.168.0.1" as Gateway, update the macro GATEWAY as **0x0100A8C0**

```
      #define  GATEWAY            0x0100A8C0
```

IP address of the network mask should also be in long format and in little endian byte order.

Example: To configure "255.255.255.0" as network mask, update the macro **NETMASK** as **0x00FFFFFF**

```
      #define   NETMASK            0x00FFFFFF
```

Note1: In this application, we are not providing the facility to configure the Access Point's IP parameters. Default IP address of the WiSeConnect Access point is "192.168.100.76"

Note2: In concurrent mode, IP networks of WiSeConnect STA and WiSeConnect Access Point should be different. So, Please configure Wireless Access Point IP network (Ex: 192.168.0.1) to other than WiSeConnect Access point IP network.

2. Open ***sapis/include/rsi_wlan_config.h*** file and update/modify following macros :

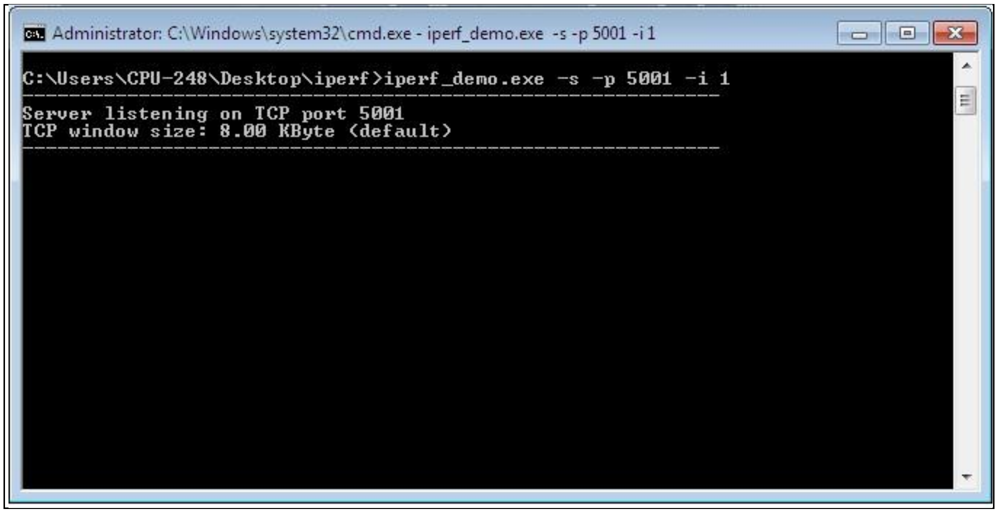
```
#define CONCURRENT_MODE                RSI_ENABLE
#define RSI_FEATURE_BIT_MAP            FEAT_SECURITY_PSK
```

```
#define RSI_TCP_IP_BYPASS           RSI_DISABLE
#define RSI_TCP_IP_FEATURE_BIT_MAP (TCP_IP_FEAT_DHCPV4_SERVER
                                  |TCP_IP_FEAT_DHCPV4_CLIENT)
#define  RSI_CUSTOM_FEATURE_BIT_MAP  0
#define  RSI_BAND                   RSI_BAND_2P4GHZ
```

2.3 Executing the Application

1. Configure the Access point in OPEN/WPA-PSK/WPA2-PSK mode to connect WiSeConnect device in STA mode.
2. Open TCP server application using iperf application in Windows PC2 which is connected to Access point through LAN.

   `Iperf_demo.exe -s -p <SERVER_PORT> -i 1`


Administrator: C:\Windows\system32\cmd.exe - iperf_demo.exe -s -p 5001 -i 1

```
C:\Users\CPU-248\Desktop\iperf>iperf_demo.exe -s -p 5001 -i 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
```

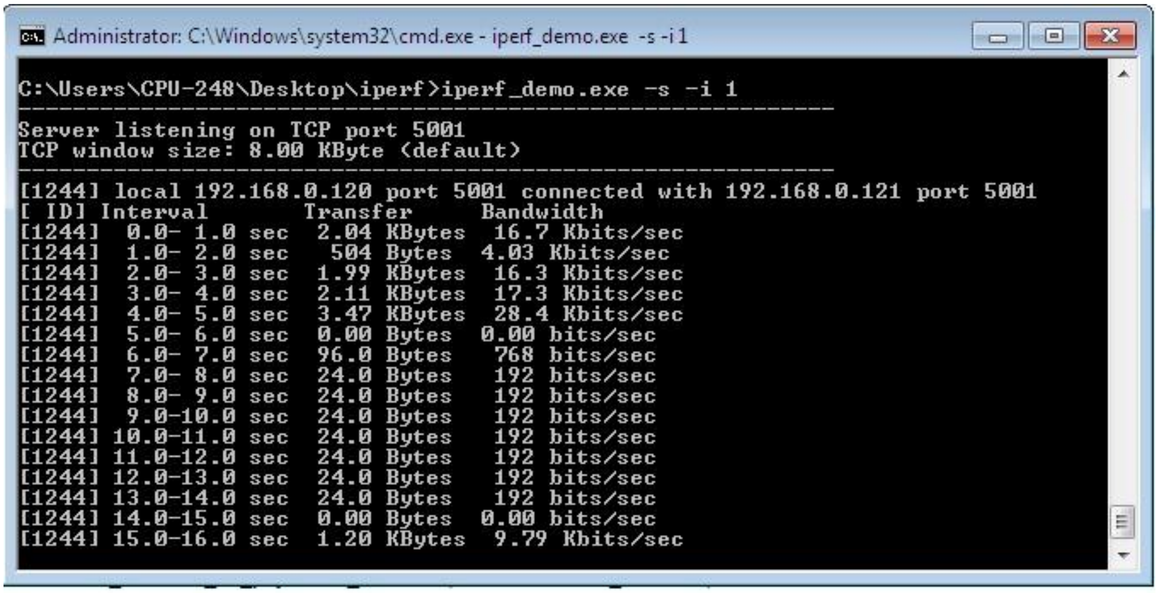
3. SPI Interface

If User using SPI interface, Please refer the document ***sapis/platforms/spansion_MB9BF568NBGL/RS9113-WiSeConnect_SAPIS_Spansion_Project_User_guide.pdf*** for executing the ***concurrent_mode*** example in CooCox IDE.

4. **UART/USB-CDC Interface**

If User using UART interface, Please refer the document ***sapis/platforms/ windows_uart/RS9113-WiSeConnect_SAPIS_Windows_Project_UserGuide.pdf*** for executing the ***concurrent_mode*** example in Dev-C++ IDE

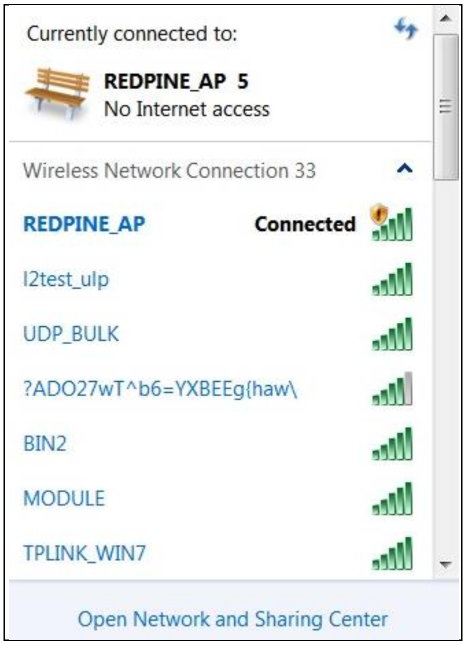
5. After the program gets executed, WiSeConnect Device will be connected to Access point and start as an Access point having the configuration same as that in the application.
6. After successful connection in STA mode, STA connects to TCP server socket opened on Windows PC2 using TCP client socket and sends configured **`NUMBER_OF_PACKETS`** to remote TCP server. Please refer the given below image for reception of TCP data on TCP server.



Administrator: C:\Windows\system32\cmd.exe - iperf_demo.exe -s -i1

```
C:\Users\CPU-248\Desktop\iperf>iperf_demo.exe -s -i 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1244] local 192.168.0.120 port 5001 connected with 192.168.0.121 port 5001
[ ID] Interval       Transfer     Bandwidth
[1244]  0.0- 1.0 sec  2.04 KBytes  16.7 Kbits/sec
[1244]  1.0- 2.0 sec   504 Bytes  4.03 Kbits/sec
[1244]  2.0- 3.0 sec  1.99 KBytes  16.3 Kbits/sec
[1244]  3.0- 4.0 sec  2.11 KBytes  17.3 Kbits/sec
[1244]  4.0- 5.0 sec  3.47 KBytes  28.4 Kbits/sec
[1244]  5.0- 6.0 sec  0.00 Bytes  0.00 bits/sec
[1244]  6.0- 7.0 sec  96.0 Bytes   768 bits/sec
[1244]  7.0- 8.0 sec  24.0 Bytes   192 bits/sec
[1244]  8.0- 9.0 sec  24.0 Bytes   192 bits/sec
[1244]  9.0-10.0 sec  24.0 Bytes   192 bits/sec
[1244] 10.0-11.0 sec  24.0 Bytes   192 bits/sec
[1244] 11.0-12.0 sec  24.0 Bytes   192 bits/sec
[1244] 12.0-13.0 sec  24.0 Bytes   192 bits/sec
[1244] 13.0-14.0 sec  24.0 Bytes   192 bits/sec
[1244] 14.0-15.0 sec  0.00 Bytes  0.00 bits/sec
[1244] 15.0-16.0 sec  1.20 KBytes  9.79 Kbits/sec
```

7. Connect WiFi STA (from laptop) to WiSeConnect Access Point (Ex: AP name configured as “REDPINE_AP”).



Currently connected to:

REDPINE_AP 5
No Internet access

Wireless Network Connection 33

REDPINE_AP Connected

l2test_ulp

UDP_BULK

?ADO27wT^b6=YXBEEg{haw\

BIN2

MODULE

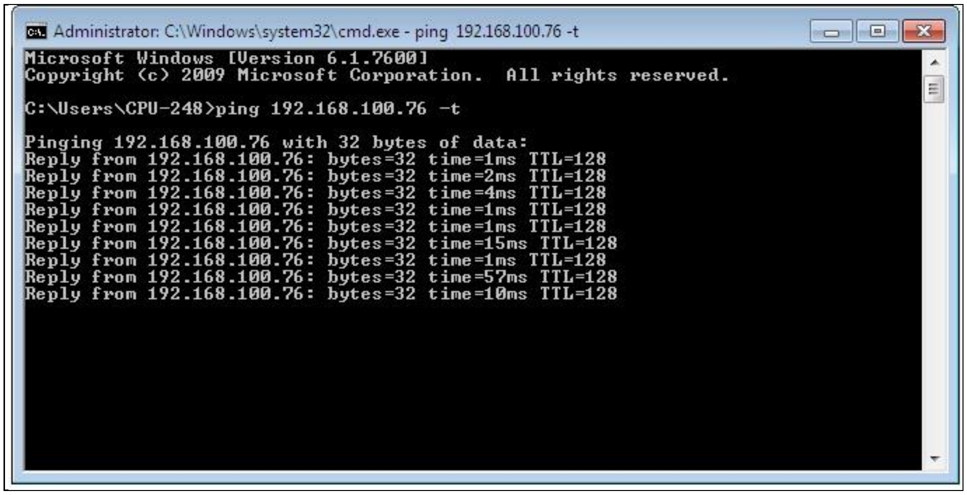
TPLINK_WIN7

Open Network and Sharing Center

8. After successful connection initiate ping from WiFi STA (Laptop) to WiSeConnect Access point IP address “192.168.100.76”.

```
ping 192.168.100.76 -t
```

9. WiSeConnect Access Point gives Ping reply for the received Ping Request. Please refer the given below image for ping success,


Administrator: C:\Windows\system32\cmd.exe - ping 192.168.100.76 -t

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\CPU-248>ping 192.168.100.76 -t

Pinging 192.168.100.76 with 32 bytes of data:
Reply from 192.168.100.76: bytes=32 time=1ms TTL=128
Reply from 192.168.100.76: bytes=32 time=2ms TTL=128
Reply from 192.168.100.76: bytes=32 time=4ms TTL=128
Reply from 192.168.100.76: bytes=32 time=1ms TTL=128
Reply from 192.168.100.76: bytes=32 time=1ms TTL=128
Reply from 192.168.100.76: bytes=32 time=15ms TTL=128
Reply from 192.168.100.76: bytes=32 time=1ms TTL=128
Reply from 192.168.100.76: bytes=32 time=57ms TTL=128
Reply from 192.168.100.76: bytes=32 time=10ms TTL=128
```