AP UDP Echo 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 as an UDP echo application in 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 UDP socket for Echo service in AP TCP/IP bypass mode.

1.1.2 Sequence of Events

This Application explains user how to:

- Create WiSeConnect device as Soft Access point in TCP/IP bypass mode
- Assign static IP to WiSeConnect soft Access point
- Open UDP socket for Echo service
- Connect WiFi Station to WiSeConnect Access point
- Send UDP datagram from Connected station to WiSeConnect Access point
- Send UDP echo by transmitting same received data from WiSeConnect device to connected station

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 PC 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-WiSeConnsect-SAPI-Porting-Guide-vx.x.pdf** for SAPIs porting to that particular platform.

- WiSeConnect device
- A Mobile device as a Wi-Fi station (This example uses a windows Laptop)
- A UDP application running on the Wi-Fi station (This example uses SocketTest application for windows)

Note: Download UDP Socket Application from below link**:**
http://sourceforge.net/projects/sockettest/files/latest/download

1.2.2 UART/USB-CDC based Setup Requirements

- Windows PC with Dev-C++ IDE
- WiSeConnect device
- A Mobile device as a Wi-Fi station (This example uses a windows Laptop)
- A UDP application running on the Wi-Fi station (This example uses SocketTest application for windows)

Note: Download UDP Socket Application from below link:
http://sourceforge.net/projects/sockettest/files/latest/download

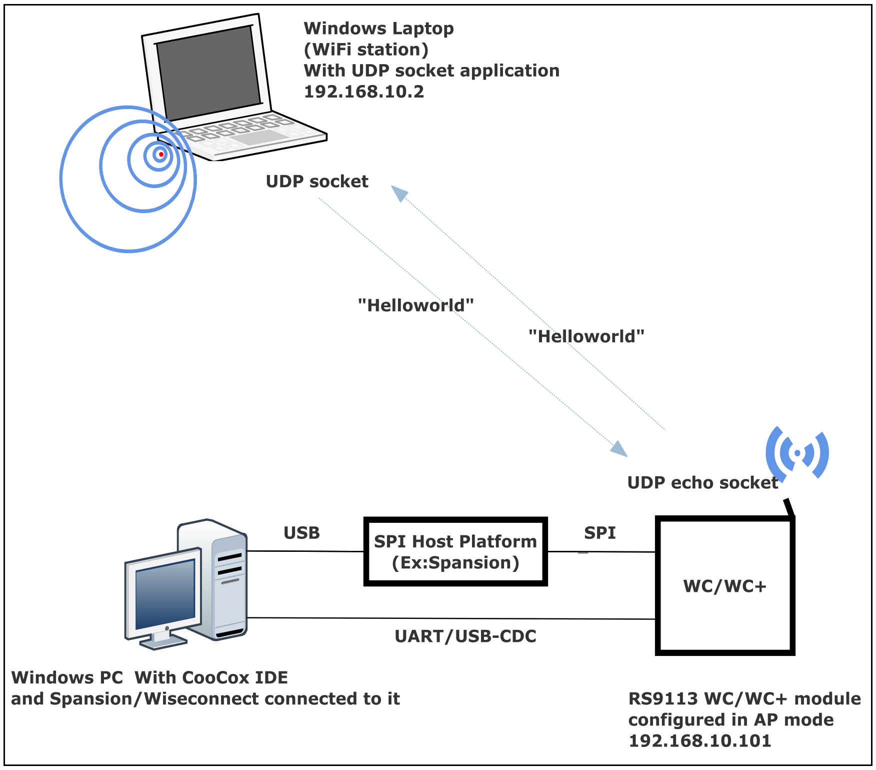


Figure 1: Setup diagram

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 ***ap_udp_echo*** 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 ***ap_udp_echo*** example in Dev-C++ IDE

2.2 Configuring the Application

1. Open ***sapis/examples/ap_udp_echo/rsi_ap_udp_echo_tcpipbypass.c*** file and update/modify following macros

SSID refers to the name of the Access point.

```
      #define SSID                    “REDPINE_AP”
```

CHANNEL_NO refers to the channel in which AP would be started

```
      #define CHANNEL_NO              11
```

Note: Valid values for CHANNEL_NO are 1 to 11 in 2.4GHz band and 36 to 48 & 149 to 165 in 5GHz band. 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.

SECURITY_TYPE refers to the type of security .Access point 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 SECURITY_TYPE              RSI_WPA2
```

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

Valid configurations are:

RSI_CCMP - For CCMP encryption

RSI_TKIP - For TKIP encryption

RSI_NONE - For open encryption

```
        #define ENCRYPTION_TYPE      RSI_CCMP
```

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

```
        #define PSK                  “1234567890”
```

BEACON_INTERVAL refers to the time delay between two consecutive beacons in milliseconds. 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 UDP server port number

```
        #define DEVICE_PORT            5001
```

REMOTE_PORT port refers remote UDP server port number

```
        #define REMOTE_PORT            5001
```

GLOBAL_BUFF_LEN refers Application memory length which is required by the driver

```
        #define GLOBAL_BUFF_LEN      8000
```

To configure IP address

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

Example: To configure “192.168.10.1” as IP address, update the macro **DEVICE_IP** as **0x010AA8C0**.

```
        #define  DEVICE_IP           0X010AA8C0
```

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

Example: To configure “192.168.10.1” as Gateway, update the macro GATEWAY as **0x010AA8C0**

```
        #define  GATEWAY             0x010AA8C0
```

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
```

Note: In AP mode, configure the same IP address for both **DEVICE_IP** and **GATEWAY** macros.

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

```
#define  CONCURRENT_MODE              RSI_DISABLE
#define  RSI_FEATURE_BIT_MAP          FEAT_SECURITY_PSK
#define  RSI_TCP_IP_BYPASS            RSI_ENABLE
#define  RSI_TCP_IP_FEATURE_BIT_MAP TCP_IP_FEAT_BYPASS
#define  RSI_CUSTOM_FEATURE_BIT_MAP   0
#define  RSI_BAND                     RSI_BAND_2P4GHZ
```

2.3 Executing the Application

1. **SPI Interface**

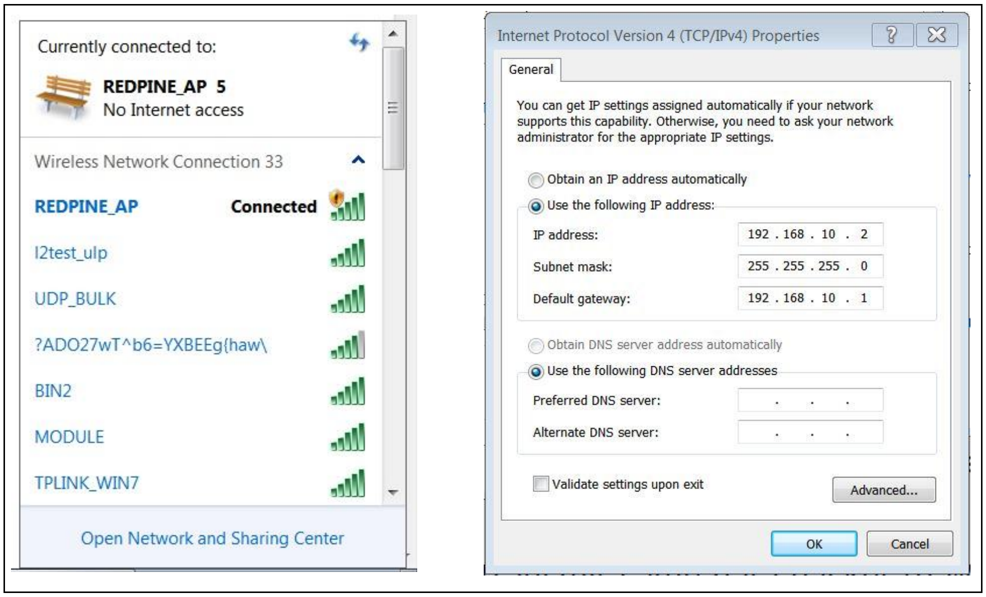
   If User using SPI interface, Please refer the document ***sapis/platforms/spansion_MB9BF568NBGL/RS9113-WiSeConnect_SAPIS_Spansion_Project_User_guide.pdf*** for executing the ***ap_udp_echo*** example in CooCox IDE.

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 executing the ***ap_udp_echo*** example in Dev-C++ IDE

3. After the program gets executed, WiSeConnect Device will be configured as Access Point.

4. Connect a WiFi station (Windows Laptop) to WiSeConnect AP (Ex: "REDPINE_AP" is the AP name) and assign a static IP in the same Network of 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

Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address: 192 . 168 . 10 . 2
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 192 . 168 . 10 . 1

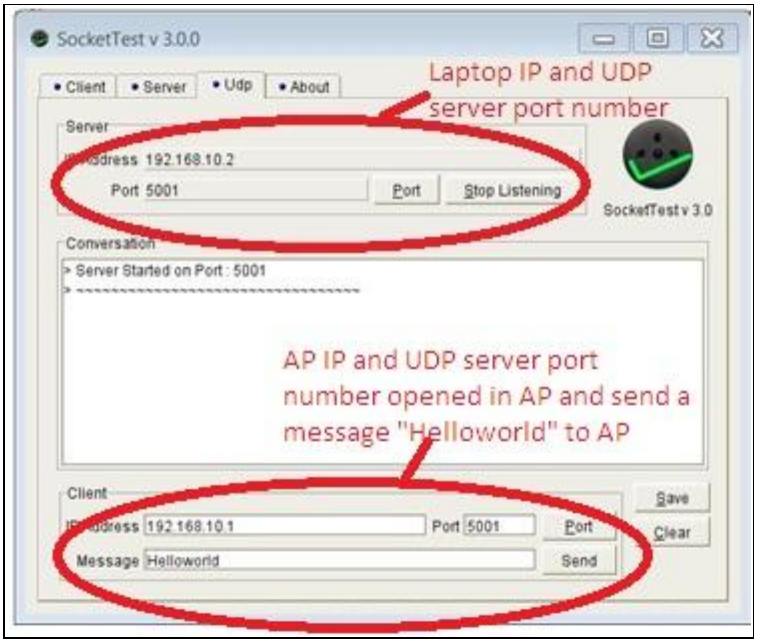
○ Obtain DNS server address automatically
● Use the following DNS server addresses

Preferred DNS server: . . .
Alternate DNS server: . . .

□ Validate settings upon exit

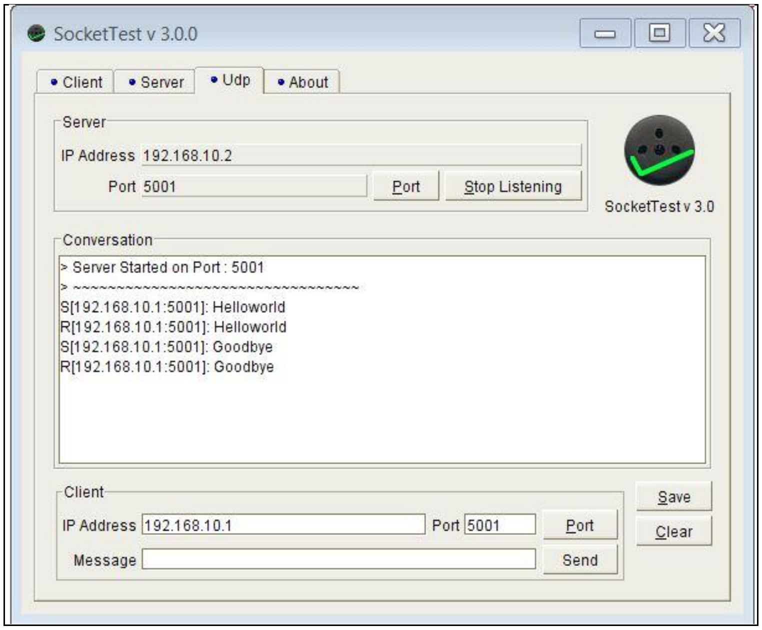
5. At remote side device (WiFi Station), open SocketTest application to open UDP server socket and client socket. As per the given below image, Open UDP server socket on port

number **REMOTE_PORT** to receive data sent by AP and open UDP client socket with port number **DEVICE_PORT** to send UDP data to AP.



```
> Server Started on Port : 5001
> ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

6. Send “Helloworld” and “Goodbye” messages from UDP client to UDP server opened in AP and same messages will send back by AP to the UDP server opened on WiFi Station. Please refer the given below image for messages sent by WiFi Station and AP.



```
> Server Started on Port : 5001
> ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
S[192.168.10.1:5001]: Helloworld
R[192.168.10.1:5001]: Helloworld
S[192.168.10.1:5001]: Goodbye
R[192.168.10.1:5001]: Goodbye
```