

# WLAN-STATION BT TCPIP bypass bridge 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: <a href="mailto:info@redpinesignals.com">info@redpinesignals.com</a>
Website: <a href="mailto:www.redpinesignals.com">www.redpinesignals.com</a>



### **About this Document**

This document describes the process of bringing up the RS9113 based module as a WiFi STA+BT co-ex mode in TCP/IP bypass 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.

Copyright © 2015 Redpine Signals, Inc. All rights reserved.



### **Table of Contents**

1	Introdu	uction	4
		plication Overview	
	1.1.1	Overview	
	1.1.2	Sequence of Events	
	1.1.2.	1 WLAN Task	4
	1.1.2.		
	1.2 Ap	plication Setup	4
	1.2.1	SPI based Setup Requirements	
	1.2.2	UART/USB-CDC based Setup Requirements	
2	Configu	uration and Execution of the Application	
		ializing the Application	
	2.1.1	SPI Interface	
	2.1.2	UART/USB-CDC Interface	
	2.2 Cor	nfiguring the Application	
	2.2.1	Configuring the WLAN task	
	2.2.2	Configuring the BT Application	
	2.3 Exe	ecuting the Application	
		Table of Element	
		<u>Table of Figures</u>	
Fi	gure 1 Setu	p to demonstrate WLAN Station BT bridge tcpipbypass application	6
		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

The coex application demonstrates how information can be exchanged seamlessly using two wireless protocols (WLAN and BT) running in the same device.

In this coex application, WiSeConnect BT device connects with remote BT device (Smart Phone) and WiSeConnect WiFi interface connects with an Access Point in station mode and do data transfer in BT and WiFi interfaces.

The coex application has WLAN and BT tasks and acts as an interface between remote Smartphone BT device and remote PC which is connected to Access point. Smartphone interacts with BT task, while remote PC interacts with WLAN task. When Smartphone connects and sends message to WiSeConnect device, BT task accepts and sends to WLAN task, which in turn sends to remote PC which is connected to Access Point. Similarly, when remote PC sends message to WiSeConnect device, the message will be sent to Smartphone via BT task.

Thus messages can be seamlessly transferred between PC and Smartphone.

#### 1.1.2 Sequence of Events

#### 1.1.2.1 WLAN Task

This Application explains user how to:

- Configure device in station mode
- Connect WiSeConnect device to Access point
- Receive UDP data sent by connected station and forward to BT task
- Send data received by BT task to connected station using UDP protocol

#### 1.1.2.2 BT Task

This Application explains user how to:

- Configure device to SPP profile mode
- Configure device in discoverable and connectable mode
- Establish SPP profile level connection with remote smart phone
- Receive data sent by Smart phone and forward to WLAN task
- Send data received by WLAN task and send to Smart phone

#### 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-WiSeConnect-SAPI-Porting-Guide-vx.x.pdf for SAPIs porting to that particular platform.

- WiSeConnect device
- Access point
- Windows PC2 with UDP socket application

Note: Download UDP socket application from below link,

http://sourceforge.net/projects/sockettest/files/latest/download

• BT supported Smart phone with SPP application.

**Note:** Install SPP Pro Application for BT SPP application.

### 1.2.2 UART/USB-CDC based Setup Requirements

- Windows PC with Dev-C++ IDE
- WiSeConnect device
- Access point
- Windows PC2 with UDP socket application

Note: Download UDP socket application from below link,

http://sourceforge.net/projects/sockettest/files/latest/download

• BT supported Smart phone with SPP application.

**Note:** Install SPP Pro Application for BT SPP application.



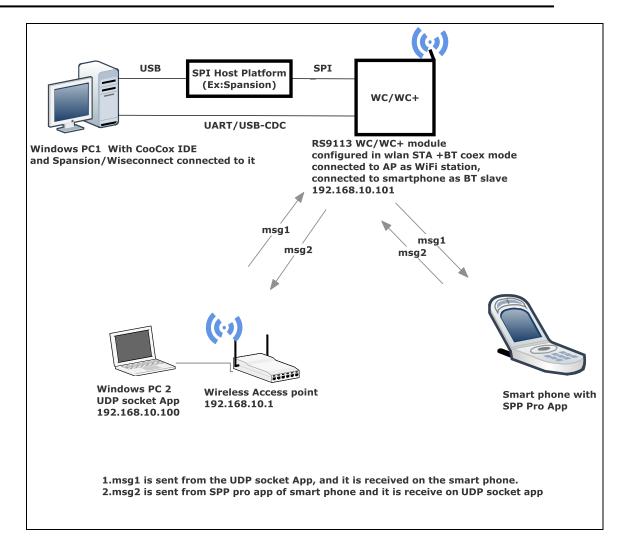


Figure 1 Setup to demonstrate WLAN Station BT bridge tcpipbypass application



### 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 wlan\_bt\_bridge\_tcpipbypass 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 *wlan\_bt\_bridge\_tcpipbypass* example in Dev-C++ IDE

### 2.2 Configuring the Application

#### 2.2.1 Configuring the WLAN task

 Open sapis/examples/wlan\_bt/wlan\_bt\_bridge\_tcpipbypass /rsi\_wlan\_app.c file and update/modify following macros:

**SSID** refers to the name of the Access point.

#define SSID "<ap name>"

**SECURITY\_TYPE** refers to the type of security. In this application STA supports Open, WPA-PSK, WPA2-PSK securities.

Valid configuration is:

RSI OPEN - For OPEN security mode

RSI WPA - For WPA security mode

RSI WPA2 - For WPA2 security mode

#define SECURITY TYPE RSI OPEN

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

#define PSK "<psk>"

**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

**REMOTE\_IP\_ADDRESS** refers remote peer IP address to send data received by BT device.



IP address of Windows PC2 which is connected to Access Point through LAN.

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

Example: To configure "192.168.10.2" as IP address, update the macro **REMOTE IP ADDRESS** as **0x020AA8C0**.

#define REMOTE IP ADDRESS

0x020AA8C0

To configure IP address:

**DHCP MODE** refers whether IP address configured through DHCP or STATIC

#define DHCP MODE 0

#### Note:

Configure DHCP\_MODE macro to 0 and assign IP address through static as co-ex application running in TCP/IP bypass mode.

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

Example: To configure "192.168.10.10" as IP address, update the macro **DEVICE\_IP** as **0x0A0AA8C0**.

#define DEVICE IP

0X0A0AA8C0

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  $0 \times 0 = 0$ 

#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

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 OPEN

#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

### 2.2.2 Configuring the BT Application

 Open sapis/examples/wlan\_bt/wlan\_bt\_bridge\_tcpipbypass /rsi\_bt\_app.c file and update/modify following macros,

**RSI\_BT\_LOCAL\_ANME** refers name of the WiSeConnect device to appear during scanning by remote devices.

#define RSI BT LOCAL NAME

"SPP SLAVE"



**PIN CODE** refers four bytes string required for pairing process.

#define PIN CODE "4321"

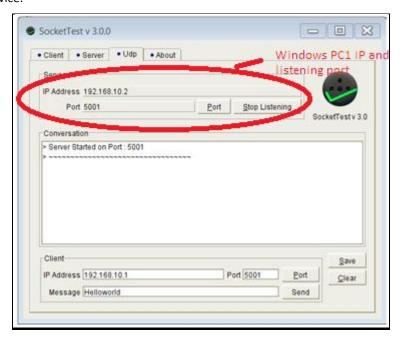
Following are the **non-configurable** macros in the application.

**BT\_GLOBAL\_BUFF\_LEN** refers Number of bytes required by the application and the driver

#define BT GLOBAL BUFF LEN 10000

### 2.3 Executing the Application

- Configure the Access point in OPEN/WPA-PSK/WPA2-PSK mode to connect WiSeConnect device in STA mode.
- Install UDP socket application in Windows PC1 and open UDP server socket on port number REMOTE\_PORT (Ex: 5001) to receive the data sent by BT remote device.



### 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 wlan\_bt\_bridge\_tcpipbypass 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 *wlan\_bt\_bridge\_tcpipbypass* example in Dev-C++ IDE

3.

4. After the program gets executed, WiSeConnect BT is in Discoverable state and WLAN will establish connection with wireless access point. After successful



connection with wireless access point WiSeconnect device opens UDP socket on port number **DEVICE PORT**.

5. Open Bluetooth SPP pro app on mobile and do scan until WiSeConnect device (Ex: "SPP\_SLAVE") present in scan list.

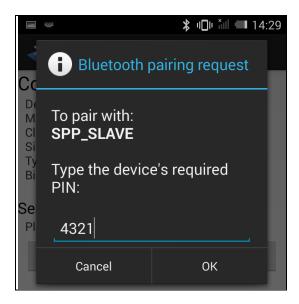


6. After successful scan, select the device and initiate pairing to WiSeConnect device.

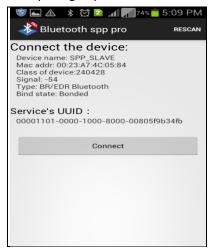


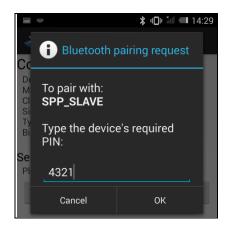
7. After initiating paring, Pairing request will pop-up at smart phone side and issue secret key which is given at WiSeConnect device (PIN\_CODE ) side.





8. After successful pair, initiate SPP connection to WiSeConnect module and give secret key for the received pairing request at remote device side.



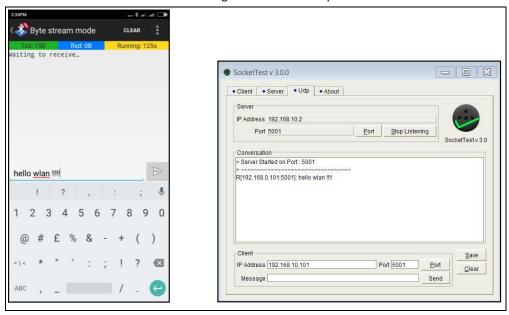




9. After successful SPP connection, select "Byte stream mode" to send and receive the data.



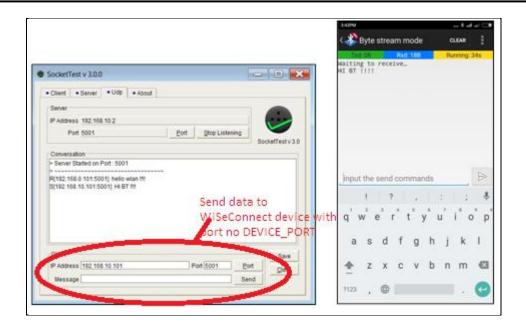
10. Send a message (Ex: "Hello wlan") from SPP pro APP to WiSeConnect device. WiSeConnect device forwards the received message from BT remote device to Windows PC1 which is connected to Access Point over WiFi protocol using UDP socket. User can observe the message on UDP socket opened on Windows PC1.



**Note:** rsi\_bt\_app\_send\_to\_wlan() function defined in rsi\_wlan\_app.c to send message from BT task to WLAN task

11. Now, send a message (Ex: "Hi BT") from Windows PC1 to WiSeConnect device using UDP socket with destination IP DEVICE\_IP and port number DEVICE\_PORT. WiSeConnect device forwards the received message from Windows PC1 to remote BT device which is connected to WiSeConnect BT device over BT protocol. User can observe the message on SPP pro app.





**Note:** rsi\_wlan\_app\_send\_to\_bt() function defined in rsi\_bt\_app.c to send message from WLAN task to BT task