WLAN-AP BT bridge TCPIP bypass 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 a WiFi AP+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.

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

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 BTLE device (Smart Phone) and WiSeConnect WiFi interface starts as an Access Point and allow stations (Windows laptop) to connect it.

The coex application has WLAN and BT tasks and acts as an interface between remote Smartphone BTLE device and connected WiFi Station. Smartphone interacts with BT task, while connected station 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 connected station. Similarly, when 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:

- Create device as an Access Point
- Connect stations to WiSeConnect 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
- Windows Laptop (STA) 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
- Windows Laptop (STA) 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.

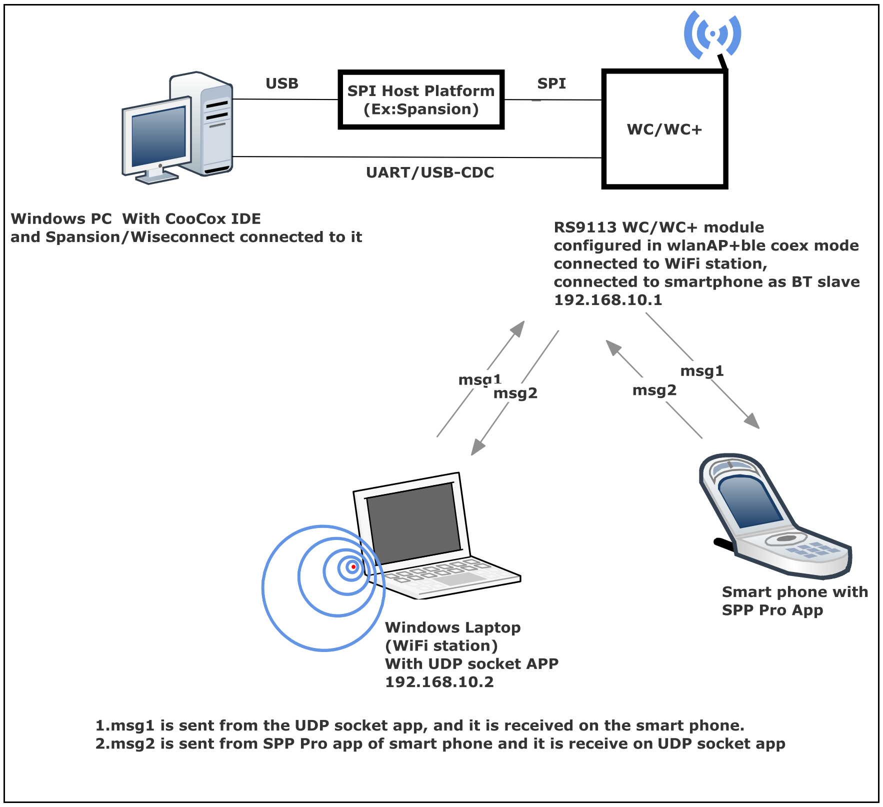

USB
SPI Host Platform
(Ex:Spansion)
SPI
WC/WC+
UART/USB-CDC
Windows PC With CooCox IDE
and Spansion/Wiseconnect connected to it
RS9113 WC/WC+ module
configured in wlanAP+ble coex mode
connected to WiFi station,
connected to smartphone as BT slave
192.168.10.1
msg1
msg2
msg1
msg2
Windows Laptop
(WiFi station)
With UDP socket APP
192.168.10.2
Smart phone with
SPP Pro App
1.msg1 is sent from the UDP socket app, and it is received on the smart phone.
2.msg2 is sent from SPP Pro app of smart phone and it is receive on UDP socket app

Figure 1 Setup to demonstrate WLAN AP 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_ap_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_ap_bt_bridge_tcpipbypass*** example in Dev-C++ IDE

2.2 Configuring the Application

2.2.1 Configuring the WLAN task

1. Open **sapis/examples/wlan_bt/wlan_ap_bt_bridge_tcpipbypass /rsi_wlan_ap_app.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 in 2.4GHz are 1 to 11 and 5GHZ are 36 to 48 and 149 to 165. 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***.

SESECURITY_TYPE refers to the type of security .Access point supports Open, WPA, WPA2 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_WPA2
```

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 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
```

REMOTE_IP_ADDRESS refers remote peer IP address to send data received by BT device.

IP address of Wilress STA which is connected to WiSeConnect Access Point.

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:

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 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            DISABLE
#define  RSI_FEATURE_BIT_MAP        FEAT_SECURITY_PSK
#define  RSI_TCP_IP_BYPASS          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 task

1. Open ***sapis/examples/wlan_bt/wlan_ap_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. 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
```

2.3 Executing the Application

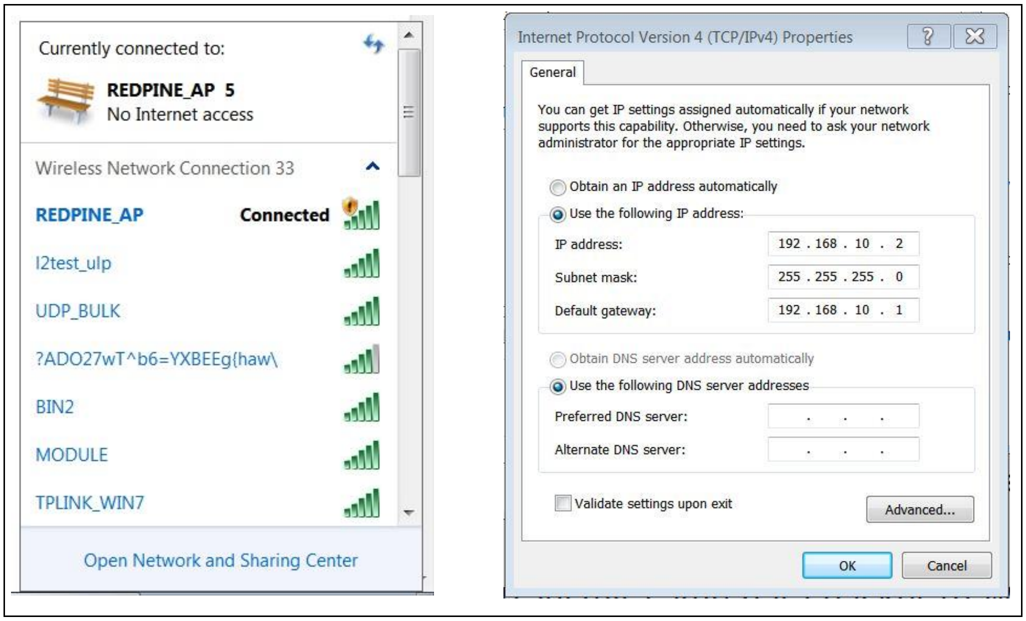
1. **SPI Interface**

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

3. After the program gets executed, WiSeConnect BT is in Discoverable state and WLAN will create as an Access Point and opens UDP socket on port number **`DEVICE_PORT`**.
4. From Windows Laptop (STA), do scan and connect to WiSeConnect Access point (Ex: “REDPINE_AP”) and assign static IP address to Laptop which is same as **`REMOTE_IP_ADDRESS`** configured in rsi_wlan_ap_app.c file.



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:

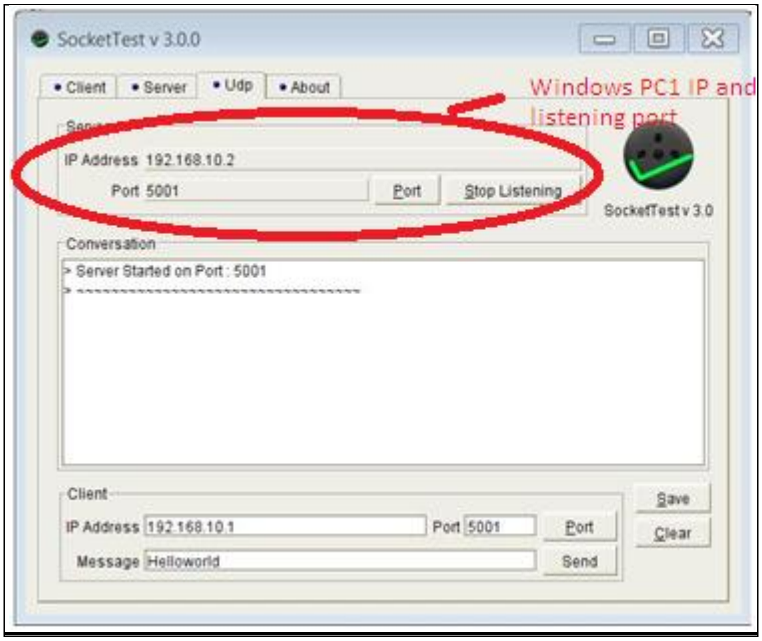
IP address:	192 . 168 . 10 . 2
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	192 . 168 . 10 . 1

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

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

□ Validate settings upon exit

5. Install UDP socket application in Windows laptop and open UDP server socket on port number **`REMOTE_PORT (Ex: 5001)`** to receive the data sent by BT remote device.



Windows PC1 IP and listening port

IP Address 192.168.10.2

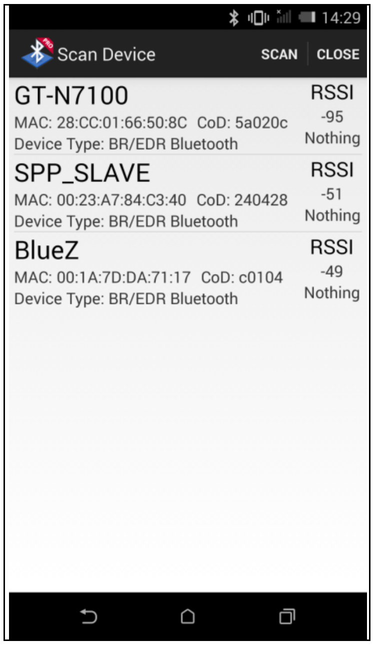
Port 5001

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

IP Address 192.168.10.1 Port 5001

Message Helloworld

6. Open Bluetooth SPP pro app on mobile and do the scan until WiSeConnect device (Ex: “SPP_SLAVE”) gets present in scan list.

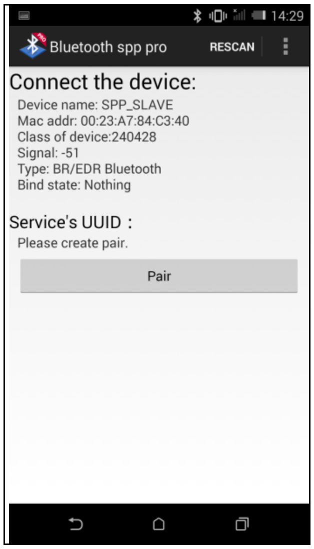


GT-N7100
MAC: 28:CC:01:66:50:8C CoD: 5a020c
Device Type: BR/EDR Bluetooth
RSSI -95 Nothing

SPP_SLAVE
MAC: 00:23:A7:84:C3:40 CoD: 240428
Device Type: BR/EDR Bluetooth
RSSI -51 Nothing

BlueZ
MAC: 00:1A:7D:DA:71:17 CoD: c0104
Device Type: BR/EDR Bluetooth
RSSI -49 Nothing

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



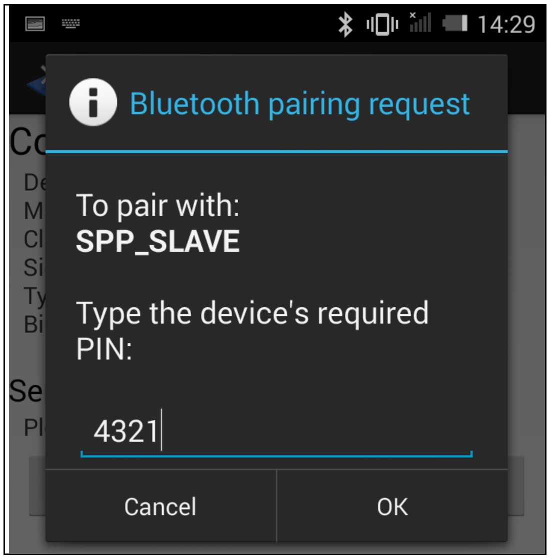
Connect the device:

Device name: SPP_SLAVE
Mac addr: 00:23:A7:84:C3:40
Class of device:240428
Signal: -51
Type: BR/EDR Bluetooth
Bind state: Nothing

Service's UUID :

Please create pair.

8. 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.



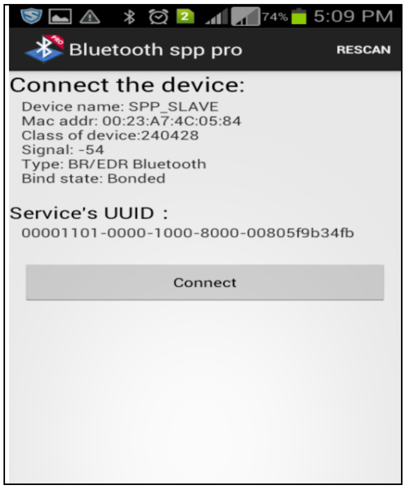
Bluetooth pairing request

To pair with:
SPP_SLAVE

Type the device's required PIN:

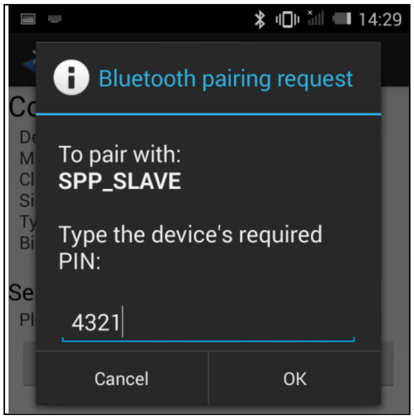
4321

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



Connect the device:
Device name: SPP_SLAVE
Mac addr: 00:23:A7:4C:05:84
Class of device:240428
Signal: -54
Type: BR/EDR Bluetooth
Bind state: Bonded

Service's UUID :
00001101-0000-1000-8000-00805f9b34fb

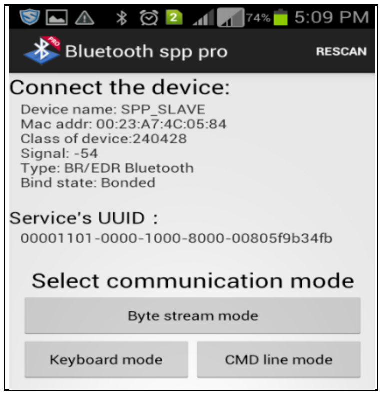


To pair with:
SPP_SLAVE

Type the device's required PIN:

4321

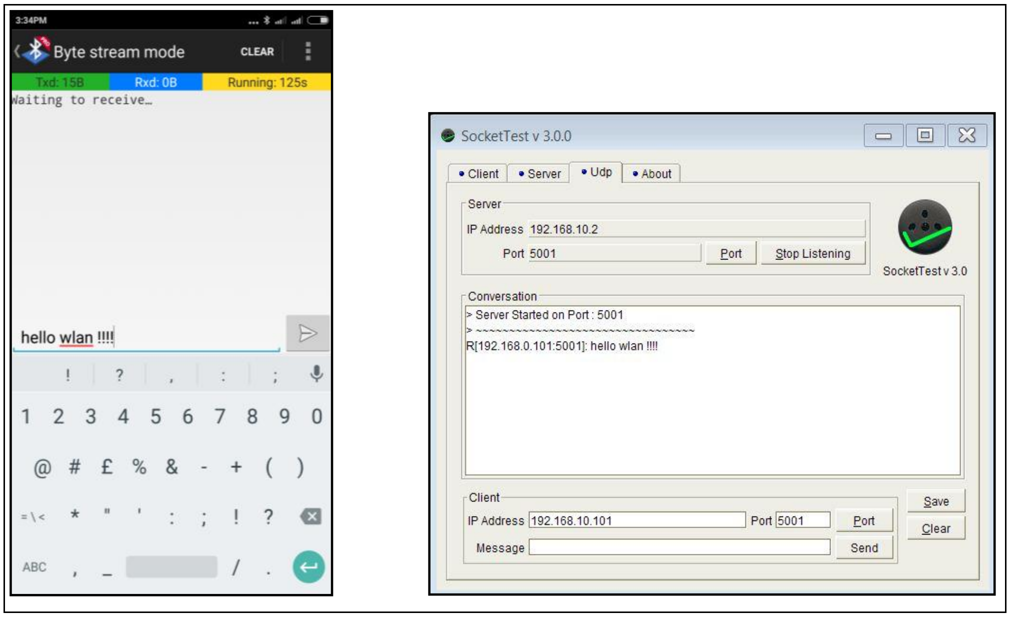
10. After successful SPP connection, select "Byte stream mode" to send and receive the data.



Connect the device:
Device name: SPP_SLAVE
Mac addr: 00:23:A7:4C:05:84
Class of device:240428
Signal: -54
Type: BR/EDR Bluetooth
Bind state: Bonded

Service's UUID :
00001101-0000-1000-8000-00805f9b34fb

11. 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 laptop which is connected as a STA over WiFi protocol using UDP socket. User can observe the message on UDP socket opened on Windows laptop.



Waiting to receive…

hello wlan !!!!

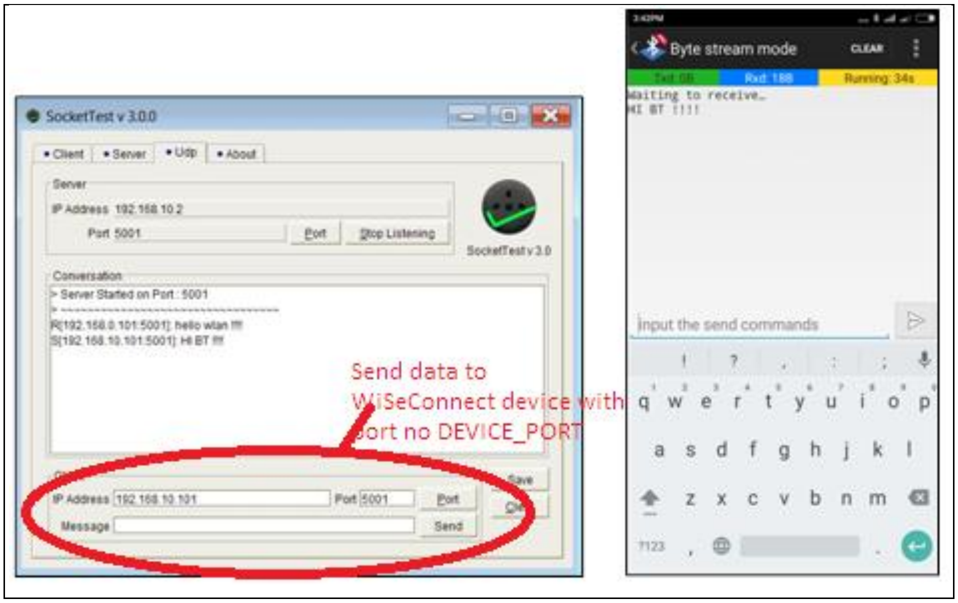
Server
IP Address 192.168.10.2
Port 5001

Conversation
```
> Server Started on Port : 5001
> ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
R[192.168.0.101:5001]: hello wlan !!!!
```

Client
IP Address 192.168.10.101
Port 5001
Message

Note: rsi_bt_app_send_to_wlan() function defined in rsi_wlan_ap_app.c to send message from BT task to WLAN task

Now, send a message (Ex: “Hi BT”) from Windows laptop to WiSeConnect device using UDP socket with destination IP **DEVICE_IP** and port number **DEVICE_PORT**. WiSeConnect device forwards the received message from Windows laptop to remote BT device which is connected to WiSeConnect BT device over BT protocol. User can observe the message on SPP pro app.



Server
IP Address 192.168.10.2
Port 5001

Conversation
```
> Server Started on Port : 5001
> ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
R[192.168.0.101:5001]: hello wlan !!!!
S[192.168.10.101:5001]: Hi BT !!!
```

IP Address 192.168.10.101
Port 5001
Message

Send data to WiSeConnect device with port no DEVICE_PORT

Waiting to receive…
HI BT !!!!

Note: rsi_wlan_app_send_to_bt() function defined in ***rsi_bt _app.c*** is to send message from WLAN task to BT task