BT SPP Master Application

User Guide
Version 0.2

May 2016

Redpine Signals, Inc.
2107 N. First Street, #680
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 BT master device and used for SPP chat application between two BT devices.

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 Master mode and establish SPP profile connection with remote slave device and data exchange between two devices using SPP profile.

In this Application, WiSeConnect device configures in Master mode and initiates basic connection with remote slave device. After successful basic connection, Application waits to accept SPP profile level connection from remote device. Once SPP connection success, Application will wait for data to receive from connected remote device. If remote device sends data to WiSeConnect device, WiSeConnect device receives the data and send back the same data to remote device using SPP profile.

1.1.2 Sequence of Events

This Application explains user how to:

- Configure WiSeConnect module to act as Master
- Connect the WiSeConnect module with the Slave
- Accept SPP level connection from the Smartphone
- Loop back the received messaged

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
- BTLE supported Smart phone with GATT client

Note: Install Light blue App for tablet for ipad mini and BLE scanner app for android smart phone.

1.2.2 UART/USB-CDC based Setup Requirements

- Windows PC with Dev-C++ IDE
- WiSeConnect device
- BTLE supported Smart phone with GATT client

Note: Install Light blue App for tablet for ipad mini and BLE scanner app for android smart phone.

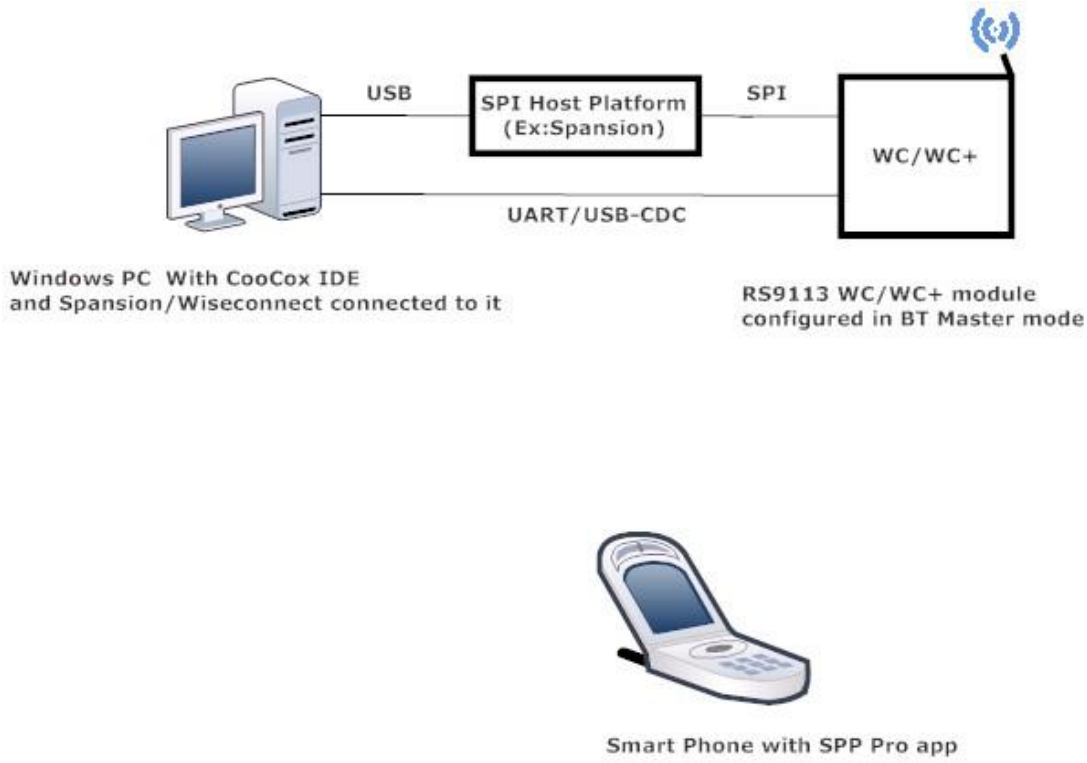


Figure 1: Setup Diagram

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

2.2 Configuring the Application

1. Open ***sapis/examples/bt/spp_master/rsi_spp_master.c*** file and update/modify following macros,

   RSI_BT_LOCAL_ANME refers name of the WiSeConnect device.

```
#define RSI_BT_LOCAL_NAME          "SPP_MASTER"
```

   PIN_CODE refers four bytes string required for pairing process.

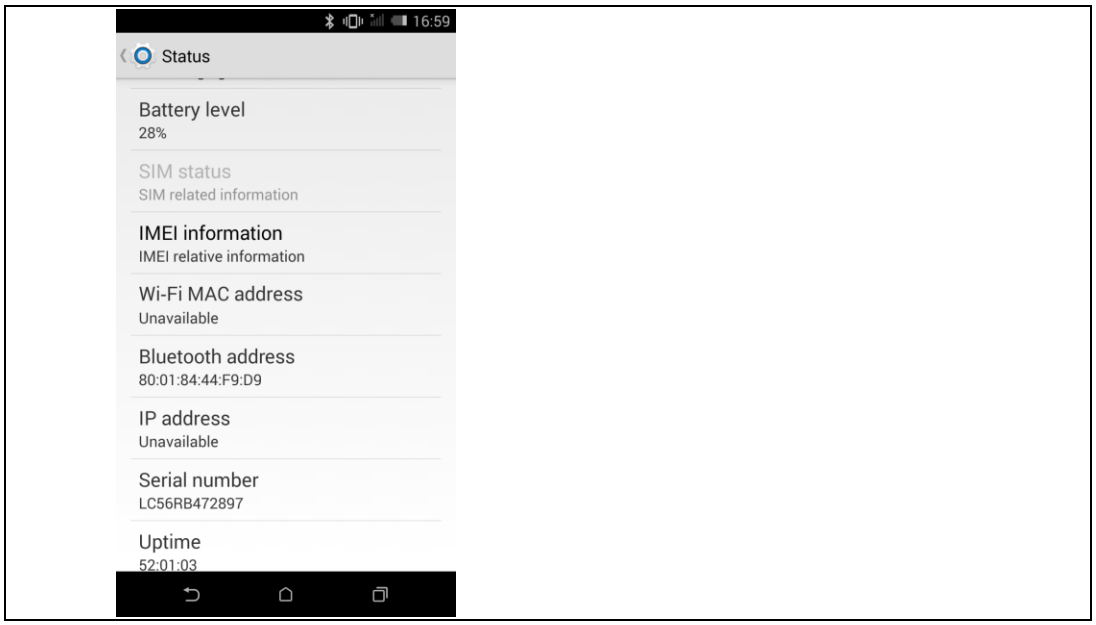
```
#define PIN_CODE          "4321"
```

   REMOTE_BD_ADDR refers Remote device BD address to connect.

   Provide the Smart phone BD address,

```
#define REMOTE_BD_ADDR             "00:1B:DC:07:2C:F0"
```

Note: In smart phone User Can check the BD address of Bluetooth device in the following location,
Settings/Aboutphone/status/Bluetooth Address



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

BT_GLOBAL_BUFF_LEN refers the number of bytes required by the application and the driver

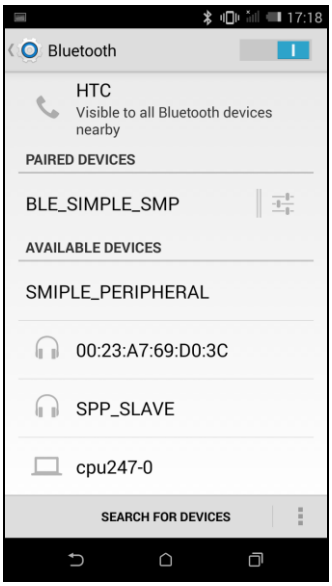
```
#define BT_GLOBAL_BUFF_LEN        10000
```

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. Power on Bluetooth in smart phone and put it in visible mode to all Bluetooth devices.



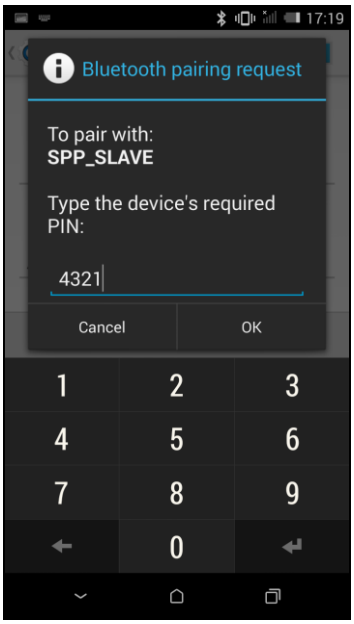
2. **SPI Interface**

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

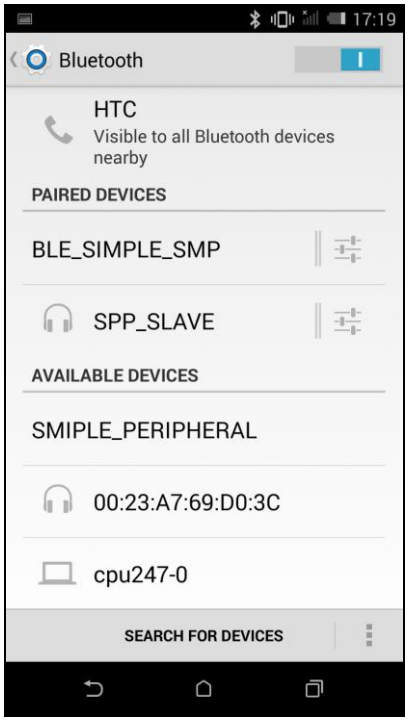
3. **UART/USB-CDC Interface**

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

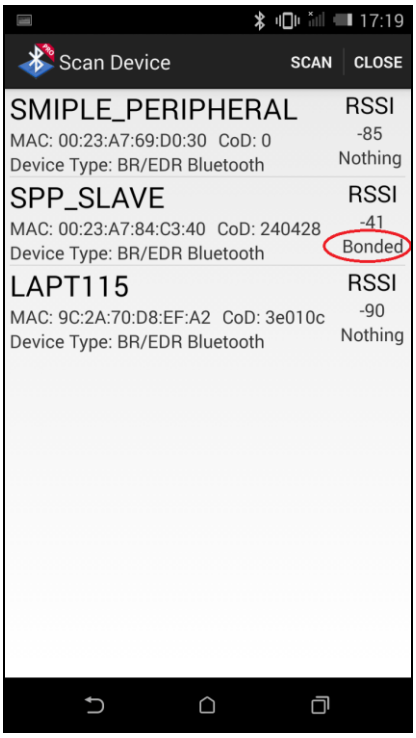
4. After the program gets executed, WiSeConnect module initiates basic connection with the remote device (Smart phone). User has to provide **`PIN_CODE`** at remote device for successful connectivity. Please find below images for connection at remote device.



5. After successful connection, In smart phone WiSeConnect device lists under Paired devices.

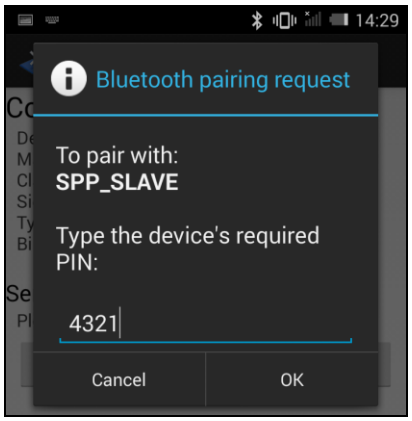


6. After successful connection, Open Bluetooth SPP pro app on mobile and do the scan until WiSeConnect device (Ex: “SPP_SLAVE”) present in scan list. After successful scan, WiSeConnect device can be found as already bonded as we have already completed basic pairing from WiSeConnect device.



7. After successful scan, select the device and initiate SPP connection with the WiSeConnect device.

8. At remote device, Bluetooth pairing request will pop-up for SPP connection success. providing secret key (PIN_CODE) for SPP connection success.



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

10. Send some data (Ex: “redpine signals”) from remote device to WiSeConenct device and same data will send back from WiSeConnect device to remote device. Please refer the given below image for sending and receiving data from remote device.