Simple Central 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 BTLE central 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

1 Introduction 4
1.1 Application Overview 4
1.1.1 Overview 4
1.1.2 Sequence of Events 4
1.2 Application Setup 4
1.2.1 SPI based Setup Requirements 4
1.2.2 UART/USB-CDC based Setup Requirements 4
2 Configuration and Execution of the Application 6
2.1 Initializing the Application 6
2.1.1 SPI Interface 6
2.1.2 UART/USB-CDC Interface 6
2.2 Configuring the Application 6
2.3 Executing the Application 7

Table of Figures

Figure 1: Setup Diagram 5

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 connect with remote BLE device in BLE central mode.

1.1.2 Sequence of Events

This Application explains user how to:

- Connect with remote BTLE peripheral device.

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 peripheral device (This Application uses TI sensor tag for remote device)

1.2.2 UART/USB-CDC based Setup Requirements

- Windows PC with Dev-C++ IDE
- WiSeConnect device
- BTLE peripheral device (This Application uses TI sensor tag for remote device)

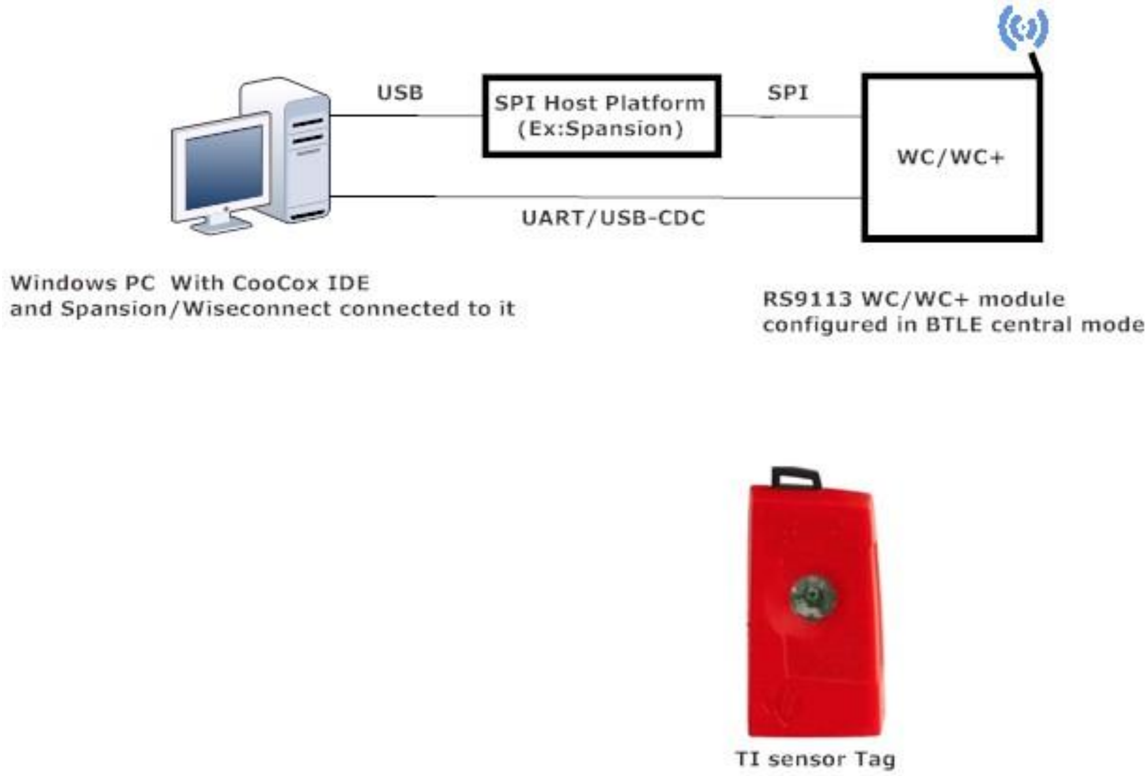

USB
SPI Host Platform (Ex:Spansion)
SPI
WC/WC+
UART/USB-CDC
Windows PC With CooCox IDE and Spansion/Wiseconnect connected to it
RS9113 WC/WC+ module configured in BTLE central mode
TI sensor Tag

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

2.2 Configuring the Application

1. Open ***sapis/examples/ble/simple_central/rsi_ble_central.c*** file and update/modify following macros,

RSI_BLE_DEV_ADDR_TYPE refers address type of the remote device to connect.

Valid configurations are LE_RANDOM_ADDRESS and LE_PUBLIC_ADDRESS

```
      #define RSI_BLE_DEV_ADDR_TYPE        LE_PUBLIC_ADDRESS
```

RSI_BLE_DEV_ADDR refers address of the remote device to connect.

```
      #define RSI_BLE_DEV_ADDR       “00:1A:7D:DA:71:13”
```

NO_OF_ADV_REPORTS refers Number of advertising reports to be held by application

```
      #define NO_OF_ADV_REPORTS        10
```

Following are the event numbers for advertising, connection and Disconnection events,

```
      #define RSI_APP_EVENT_ADV_REPORT          0
      #define RSI_APP_EVENT_CONNECTED           1
      #define RSI_APP_EVENT_DISCONNECTED        2
```

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_DISABLE
```

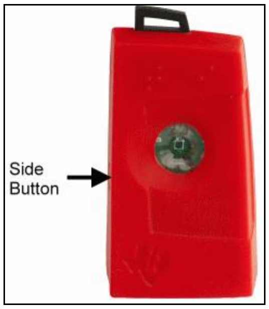
```
#define RSI_TCP_IP_FEATURE_BIT_MAP TCP_IP_FEAT_DHCPV4_CLIENT


#define  RSI_CUSTOM_FEATURE_BIT_MAP  0
#define  RSI_BAND                    RSI_BAND_2P4GHZ
```

2.3 Executing the Application

1. Configure the remote TI sensor tag in peripheral mode and put it in advertising mode.

   Press the side button to configure TI sensor tag in advertising mode



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

4. After the program gets executed, WiSeConnect device tries to connect with the remote device specified in **RSI_BLE_DEV_ADDR** macro.
5. Observe that the connection is established between the desired device and WiSeConnect device.