Simple Chat 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 peripheral mode with GATT server running it.

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 GATT server in BLE peripheral mode and explains how to do read&write operations with GATT server from connected remote device using GATT client.

In this Application, GATT server configures with Custom service with write and readable characteristic UUIDs. When connected remote device writes data to writable characteristic UUID, WiseConnect device receives the data which is received on writable characteristic UUID and writes the same data to readable characteristic UUID and sends notifications to the connected device (or) remote device can read the same data using read characteristic UUID.

1.1.2 Sequence of Events

This Application explains user how to:

- Create Simple chat service
- Make the device to advertise
- Connect from remote BTLE device
- Receive the message from the connected peer/Smartphone
- Loop back the received message

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.

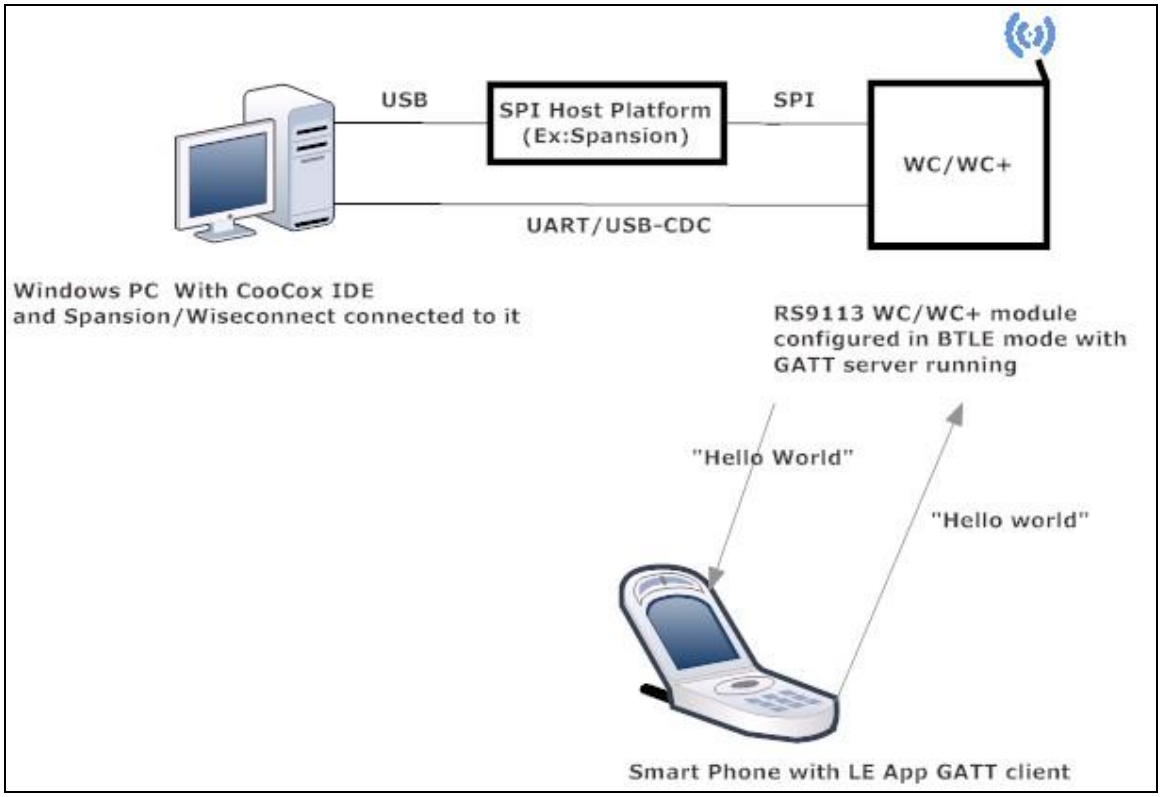


Figure 1: Setup Diagram

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 ***simple_chat*** 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_chat*** example in Dev-C++ IDE

2.2 Configuring the Application

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

RSI_BLE_NEW_SERVICE_UUID refers to the attribute value of the newly created service.

```
#define RSI_BLE_NEW_SERVICE_UUID       0xAABB
```

RSI_BLE_ATTRIBUTE_1_UUID refers to the attribute type of the first attribute under this service (**RSI_BLE_NEW_SERVICE_UUID**).

```
#define RSI_BLE_ATTRIBUTE_1_UUID     0x1AA1
```

RSI_BLE_ATTRIBUTE_2_UUID refers to the attribute type of the second attribute under this service (**RSI_BLE_NEW_SERVICE_UUID**).

```
#define RSI_BLE_ATTRIBUTE_2_UUID     0x1BB1
```

RSI_BLE_MAX_DATA_LEN refers to the Maximum length of the attribute data.

```
#define RSI_BLE_MAX_DATA_LEN      20
```

RSI_BLE_APP_SIMPLE_CHAT refers name of the WiSeConnect device to appear during scanning by remote devices.

```
#define RSI_BLE_ APP_SIMPLE_CHAT    "BLE_SIMPLE_CHAT"
```

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

RSI_BLE_CHAR_SERV_UUID refers to the attribute type of the characteristics to be added in a service.

```
#define RSI_BLE_CHAR_SERV_UUID         0x2803
```

RSI_BLE_CLIENT_CHAR_UUID refers to the attribute type of the client characteristics descriptor to be added in a service.

```
#define RSI_BLE_CLIENT_CHAR _UUID     0x2902
```

RSI_BLE_ATT_PROPERTY_READ is used to set the READ property to an attribute value.

```
#define RSI_BLE_ATT_PROPERTY_READ      0x02
```

RSI_BLE_ATT_PROPERTY_WRITE is used to set the WRITE property to an attribute value.

```
#define RSI_BLE_ATT_PROPERTY_WRITE      0x08
```

RSI_BLE_ATT_PROPERTY_NOTIFY is used to set the NOTIFY property to an attribute value.

```
#define RSI_BLE_ATT_PROPERTY_NOTIFY      0x10
```

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. **SPI Interface**

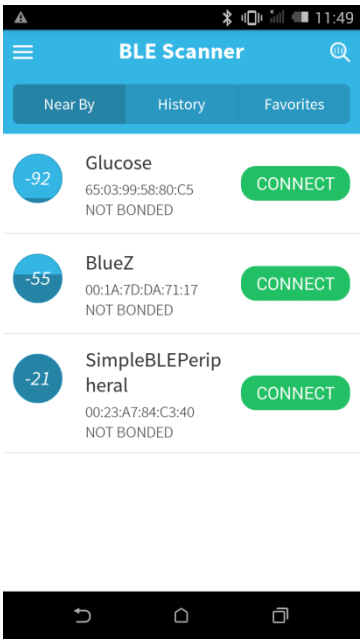
   If User using SPI interface, Please refer the document ***sapis/platforms/spansion_MB9BF568NBGL/RS9113-WiSeConnect_SAPIS_Spansion_Project_User_guide.pdf*** for executing the ***simple_chat*** example in CooCox IDE.

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

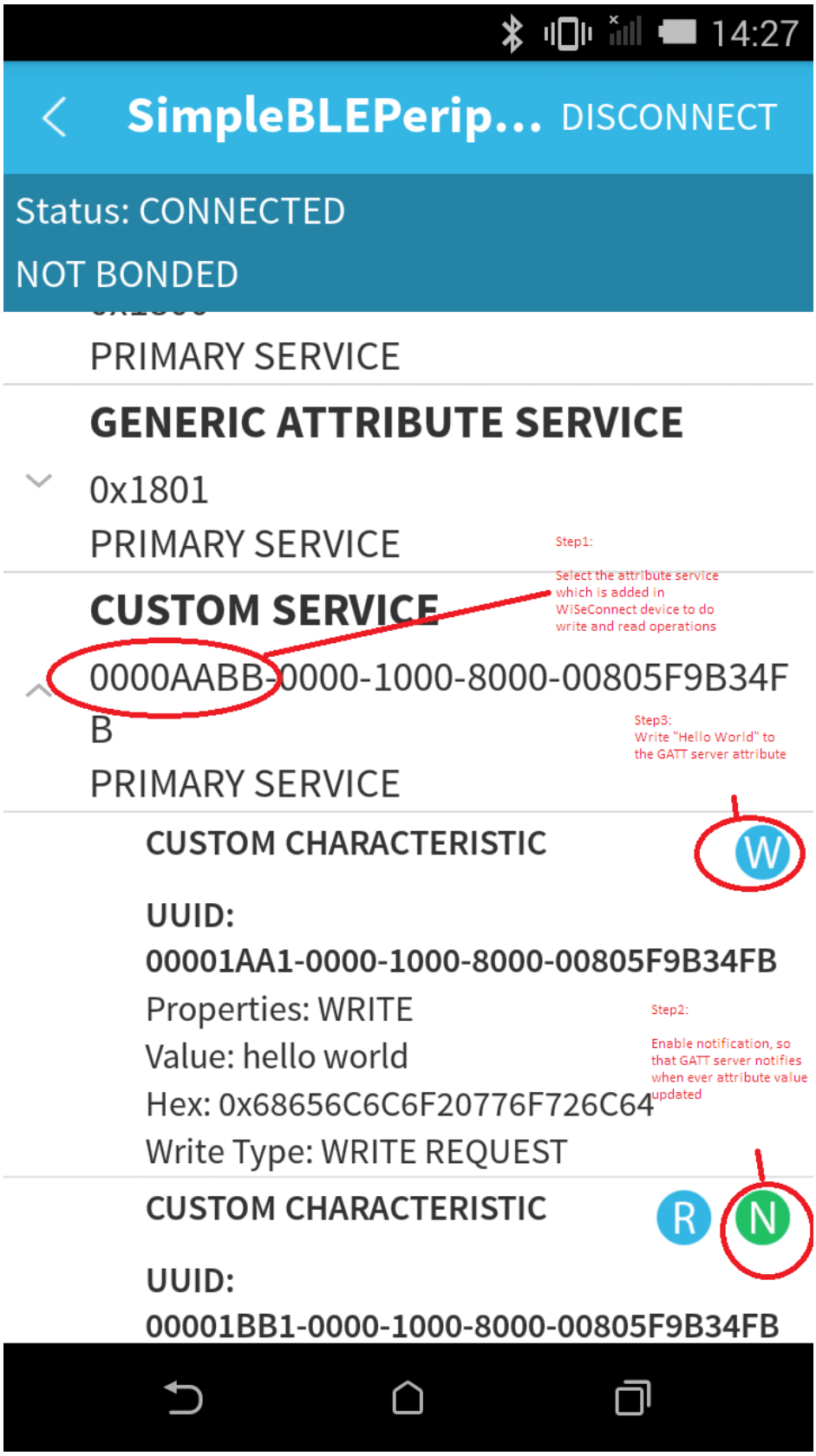
   If User using UART interface, Please refer the document ***sapis/platforms/windows_uart/RS9113-WiSeConnect_SAPIS_Windows_Project_UserGuide.pdf*** for executing the ***simple_chat*** example in Dev-C++ IDE

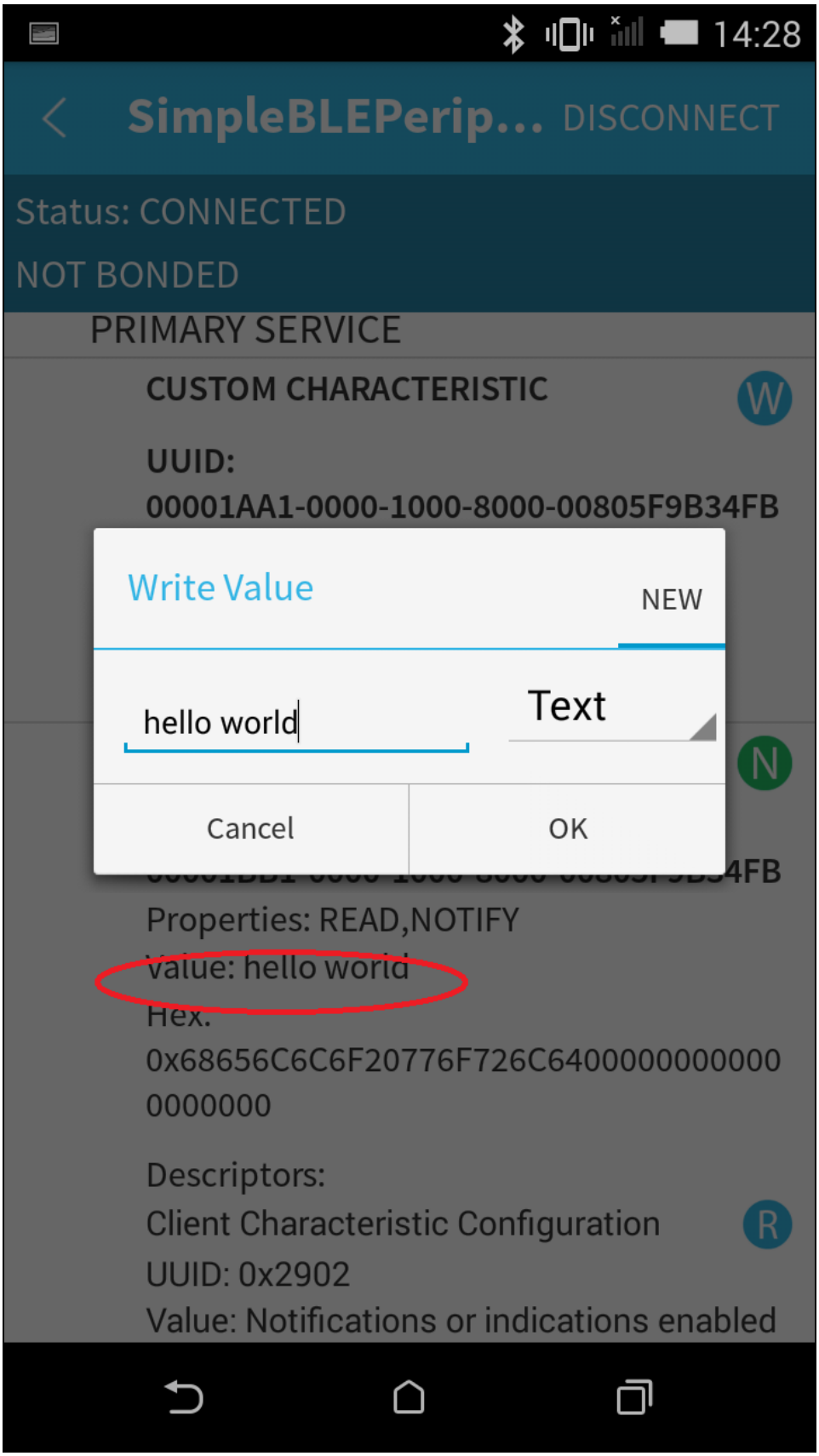
3. After the program gets executed, WiSeConnect module will be in Advertising state.
4. Open a LE App in the Smartphone and do the scan.

5. In the App, WiSeConnect module device will appear with the name configured in the macro **RSI_BLE_APP_SIMPLE_CHAT (Ex: "BLE_SIMPLE_CHAT")** or sometimes observed as WiSeConenct device as internal name "**SimpleBLEPeripheral**".



6. Initiate connection from the App.
7. After successful connection, LE scanner displays the supported services of WiSeConnect module.
8. Select the attribute service which is added **`RSI_BLE_NEW_SERVICE_UUID`**
   (Ex: 0xAABB).
9. After selecting the service do **Write and Read** operations with GATT server.
10. Enable notifications for the read attribute **`RSI_BLE_ATTRIBUTE_2_UUID`**
    (Ex: 0x1BB1). So that GATT server notifies when value updated in that particular attribute.
11. Write data (Ex: “Hello World”) to attribute **`RSI_BLE_ATTRIBUTE_1_UUID`** (Ex: 0x1AA1). So that GATT server notifies when value updated in that particular attribute.
12. WiSeConnect module receives the data sent by remote device and same data writes into the attribute **`RSI_BLE_ATTRIBUTE_2_UUID`** (Ex: 0x1BB1) and will notifies the GATT client (remote device).
13. Please refer the given below images for write and read operations from remote device GATT client.


14:27
SimpleBLEPerip... DISCONNECT
Status: CONNECTED
NOT BONDED
PRIMARY SERVICE
GENERIC ATTRIBUTE SERVICE
0x1801
PRIMARY SERVICE
Step1:
Select the attribute service which is added in WISeConnect device to do write and read operations
CUSTOM SERVICE
0000AABB-0000-1000-8000-00805F9B34FB
Step3:
Write "Hello World" to the GATT server attribute
PRIMARY SERVICE
CUSTOM CHARACTERISTIC
W
UUID:
00001AA1-0000-1000-8000-00805F9B34FB
Properties: WRITE
Value: hello world
Hex: 0x68656C6C6F20776F726C64
Write Type: WRITE REQUEST
Step2:
Enable notification, so that GATT server notifies when ever attribute value updated
CUSTOM CHARACTERISTIC
R
N
UUID:
00001BB1-0000-1000-8000-00805F9B34FB


14:28
SimpleBLEPerip...
DISCONNECT
Status: CONNECTED
NOT BONDED
PRIMARY SERVICE
CUSTOM CHARACTERISTIC
W
UUID:
00001AA1-0000-1000-8000-00805F9B34FB
Write Value
NEW
hello world
Text
Cancel
OK
N
Properties: READ,NOTIFY
Value: hello world
Hex:
0x68656C6C6F20776F726C6400000000000000000000
Descriptors:
Client Characteristic Configuration
R
UUID: 0x2902
Value: Notifications or indications enabled