

BLE
Version 1.2

30 April 2018

Redpine Signals, Inc.

2107 North First Street, Suite #540, San Jose, California 95131,

United States of America.

Phone: +1-408-748-3385, Fax: +1-408-705-2019

Email: info@redpinesignals.com

Website: www.redpinesignals.com

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 © 2018 Redpine Signals, Inc. All rights reserved.

Table of Contents

1	BLE Only peripheral	5
1.1	Introduction	5
1.2	Example	5
1.2.1	Overview	5
1.2.2	Sequence of Events	5
1.3	Setup Diagrams	5
1.3.1	Setup Requirements	5
1.4	Configuration and Execution of the Application	6
1.5	Configuring the Application	6
1.6	Executing the Peripheral Application	7
2	BLE Only Peripheral Powersave	9
2.1	Introduction	9
2.2	Example	9
2.2.1	Overview	9
2.2.2	Sequence of Events	9
2.3	Setup Diagrams	9
2.3.1	Setup Requirements	10
2.4	Configuration and Execution of the Application	10
2.4.1	Configuring the Peripheral Application	10
2.4.2	Executing the Peripheral Application	12
3	BLE Test Mode Rx	15
3.1	Introduction	15
3.2	Example	15
3.2.1	Overview	15
3.2.2	Sequence of Events	15
3.3	Setup Diagrams	15
3.3.1	Setup Requirements	15
3.4	Configuration and Execution of the Application	16
3.5	Configuring the Application	16

3.6	Executing the Rx Test mode Application	18
4	BLE Test Mode Tx	20
4.1	Introduction	20
4.2	Example	20
4.2.1	Overview	20
4.2.2	Sequence of Events	20
4.3	Setup Diagrams	20
4.3.1	Setup Requirements	20
4.4	Configuration and Execution of the Application	21
4.4.1	Configuring the Active Tx Test Mode Application:.....	21
4.4.2	Executing the Tx Test mode Application	23

1 BLE Only peripheral

1.1 Introduction

This example is applicable to WiSeConnect[™] and WiSeMCU[™] parts. For simplicity, this document refers to WiSeConnect, but all discussion applies to both WiSeConnect and WiSeMCU parts. The feature(s) used in this example may or may not be available in your part. Refer to the product datasheet to verify the features available in your part.

1.2 Example

1.2.1 Overview

This application demonstrates that how to configure device in Advertising mode and in connected mode in simple BLE peripheral mode case.

1.2.2 Sequence of Events

This Application can be run in two different roles.

1. Advertising
2. Slave

Slave role explains user how to:

- Set local name to the device
- Get local name from our device
- Configure the device to advertise
- Connect to advertising using the mobile or the BLE Central (Master Device).
- Continue advertising even after disconnection with the peer.

Note:

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

1.3 Setup Diagrams

In WiseMCU ble only project redpine application will run in M4 processor RAM based. M4 itself acts as Host interface; here no external host interface is required.

1.3.1 Setup Requirements

- Windows PC with KIEL
- Redpine Module

- Smart phone (Android)/Tablet with LE Application.

Ex: Install Light Blue app for Tablet ipad mini / BLE Scanner app for android smart phone.

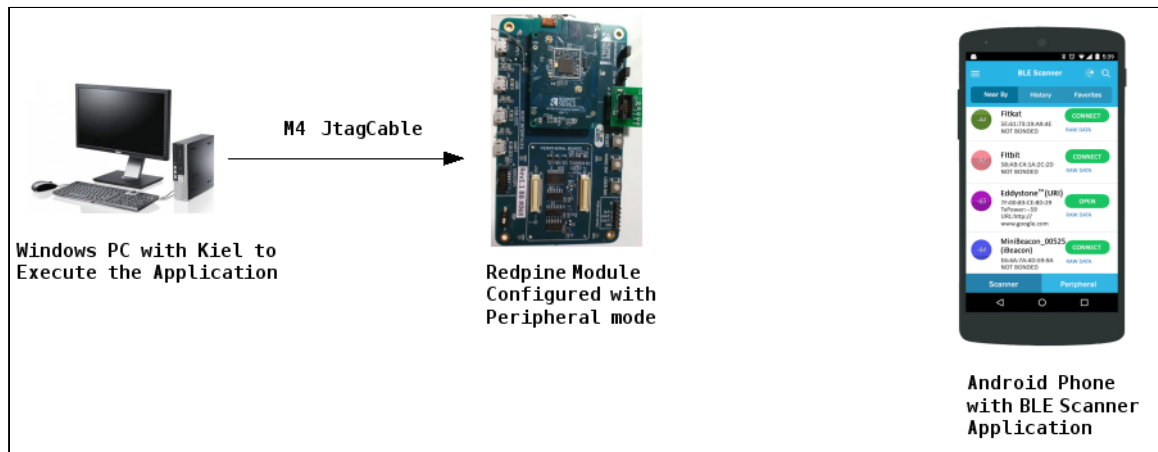


Figure 1: Setup Diagram for peripheral mode

1.4 Configuration and Execution of the Application

1.5 Configuring the Application

1. Open ***simple_peripheral.c*** file and update/modify following macros:

RSI_BLE_LOCAL_NAME refers the name of the Redpine device to appear during scanning by remote devices.

```
#define RSI_BLE_LOCAL_NAME "BLE_ONLY_IMAGE"
```

RSI_SEL_ANTENNA refers to the antenna which is to be used by Redpine module.
If user using internal antenna then set,

```
#define RSI_SEL_ANTENNA RSI_SEL_INTERNAL_ANTENNA
```

If user using external antenna (U.FL connector) then set,

```
#define RSI_SEL_ANTENNA RSI_SEL_EXTERNAL_ANTENNA
```

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

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

```
#define BT_GLOBAL_BUFF_LEN            15000
```

2. Open **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_EXT_CUSTOM_FEATURE_BIT_MAP
#define RSI_BAND                      RSI_BAND_2P4GHZ      MODE_256K
```

3. Open **rsi_ble_config.h** file and update/modify following macros,

```
#define RSI_BLE_PWR_INX
8
#define RSI_BLE_PWR_SAVE_OPTIONS
#define RSI_DUTY_CYCLING              0
```

Note:

rsi_wlan_config.h and rsi_ble_config.h files are already set with desired configuration in respective example folders user need not change for each example.

1.6 Executing the Peripheral Application

1. After the program gets executed, Redpine module will be in advertising state configured power save profile.
2. Open a LE App in the Smartphone and do the scan.
3. Initiate connection from the mobile App.

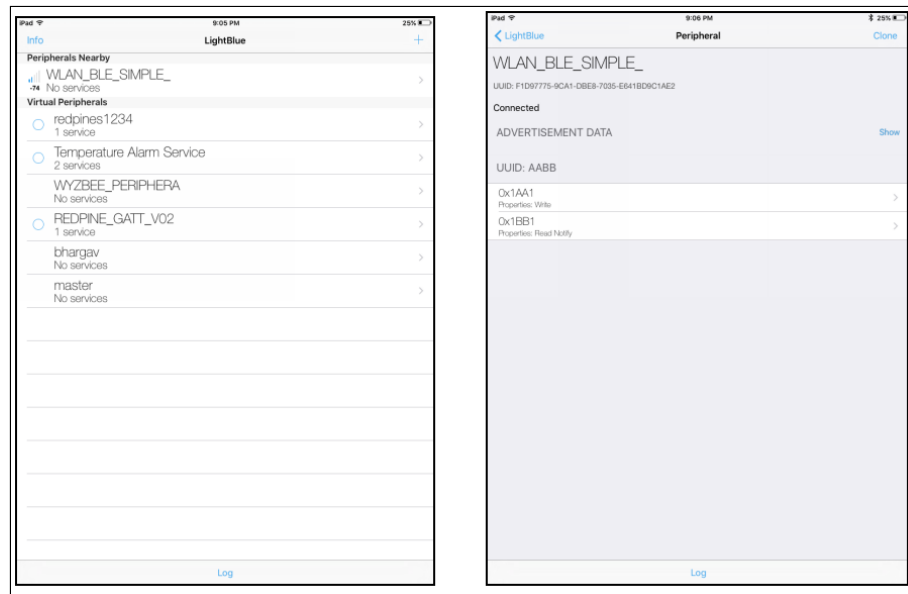


Figure 2: Scanning for BLE devices and connecting to BLE_ONLY_SIMPLE device.

4. Observe that the connection is established between Smartphone or dongle and Redpine module.
5. Verify whether module advertising again after successful disconnection with the master device.

2 BLE Only Peripheral Powersave

2.1 Introduction

This example is applicable to WiSeConnect[™] and WiSeMCU[™] parts. For simplicity, this document refers to WiSeConnect, but all discussion applies to both WiSeConnect and WiSeMCU parts. The feature(s) used in this example may or may not be available on your part. Refer to the product datasheet to verify the features available on your part.

2.2 Example

2.2.1 Overview

This application demonstrates how to configure a device in power save in Advertising mode and in connected mode in simple BLE peripheral mode with power save.

2.2.2 Sequence of Events

This Application can be run in three different roles.

1. Slave role
2. Advertising role

Slave role explains to the user how to:

- Set the local name to the device
- Get local name from our device
- Configure the device to advertise
- Configure the module in power save mode
- Analyze power save functionality when WiSeMCU device in Advertise mode and in the connected state as Slave using Agilent power analyzer.
- Continue advertising with power save even after disconnection with the peer.

Note:

Install Light blue App for the tablet iPad mini and BLE scanner app for an android smart phone.

2.3 Setup Diagrams

In WiseMCU ble only project redpine application will run on M4 processor RAM based. M4 itself acts as Host interface; here no external host interface is required.

2.3.1 Setup Requirements

- Windows PC with KIEL
- Redpine Module
- Smartphone (Android)/Tablet with LE Application.

Ex: Install Light Blue app for Tablet iPad mini / BLE Scanner app for an Android smartphone.

- Agilent Power Analyzer.

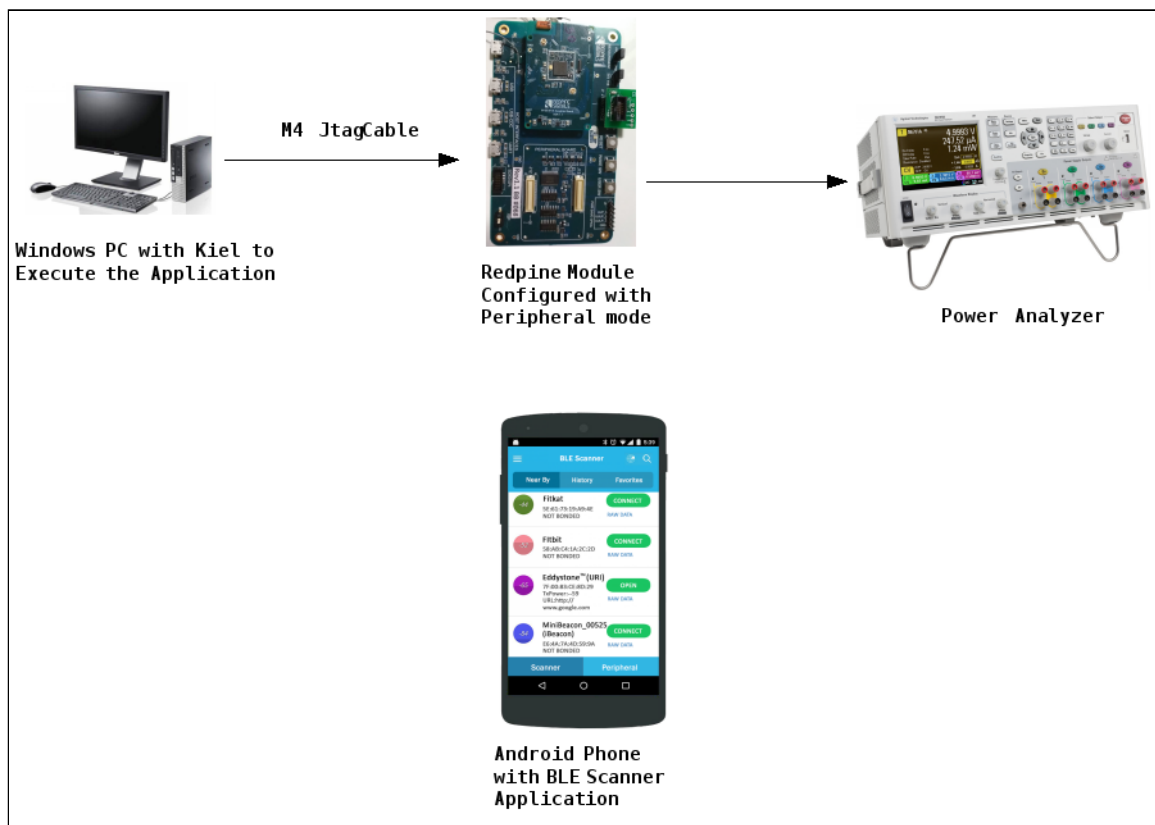


Figure 1: Setup Diagram for peripheral mode

2.4 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.

2.4.1 Configuring the Peripheral Application

1. Open ***simple_peripheral_pwr_save.c*** file and update/modify following macros:

RSI_BLE_LOCAL_NAME refers the name of the Redpine device to appear during scanning by remote devices.

```
#define RSI_BLE_LOCAL_NAME          "BLE_ONLY_IMAGE"
```

RSI_SEL_ANTENNA refers to the antenna which is to be used by Redpine module.
If the user using internal antenna then set,

```
#define RSI_SEL_ANTENNA            RSI_SEL_INTERNAL_ANTENNA
```

If the user using an external antenna (U.FL connector) then set,

```
#define RSI_SEL_ANTENNA            RSI_SEL_EXTERNAL_ANTENNA
```

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

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

```
#define BT_GLOBAL_BUFF_LEN          15000
```

2. Open **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_EXT_CUSTOM_FEATURE_BIT_MAP  
#define RSI_BAND                      RSI_BAND_2P4GHZ  
MODE_256K
```

3. Open **rsi_ble_config.h** file and update/modify following macros,

```
#define RSI_BLE_PWR_INX  
8  
#define RSI_BLE_PWR_SAVE_OPTIONS 0  
#define RSI_DUTY_CYCLING 0
```

Note:

rsi_wlan_config.h and rsi_ble_config.h files are already set with the desired configuration in respective example folders user need not change for each example.

2.4.2 Executing the Peripheral Application

1. After the program gets executed, Redpine module will be in advertising state configured power save profile.
2. Open an LE App in the smartphone and do the scan.
3. WiSeMCU device will go to sleep and wakes up for every advertising interval and goes back to sleep after advertising. Please refer the given below image for power save cycle in advertising mode.



Figure 2: Connectable Advertising with power save

4. Initiate connection from the mobile App.

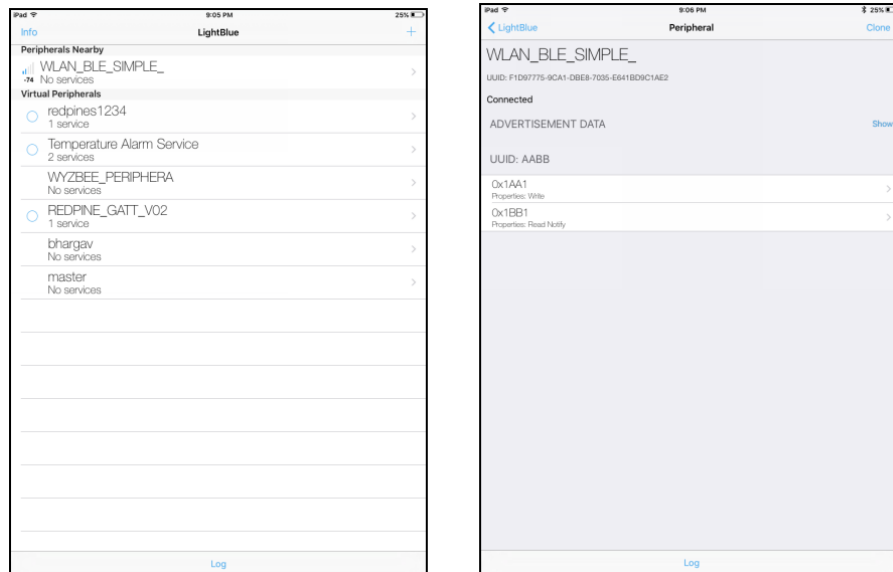


Figure 3: Scanning for BLE devices and connecting to BLE_ONLY_SIMPLE device

5. Observe that the connection is established between Smartphone or dongle and Redpine module.
6. After successful connection, Module goes to sleep and wakes up for every connection interval. Please check the image below for power save cycle after connection.

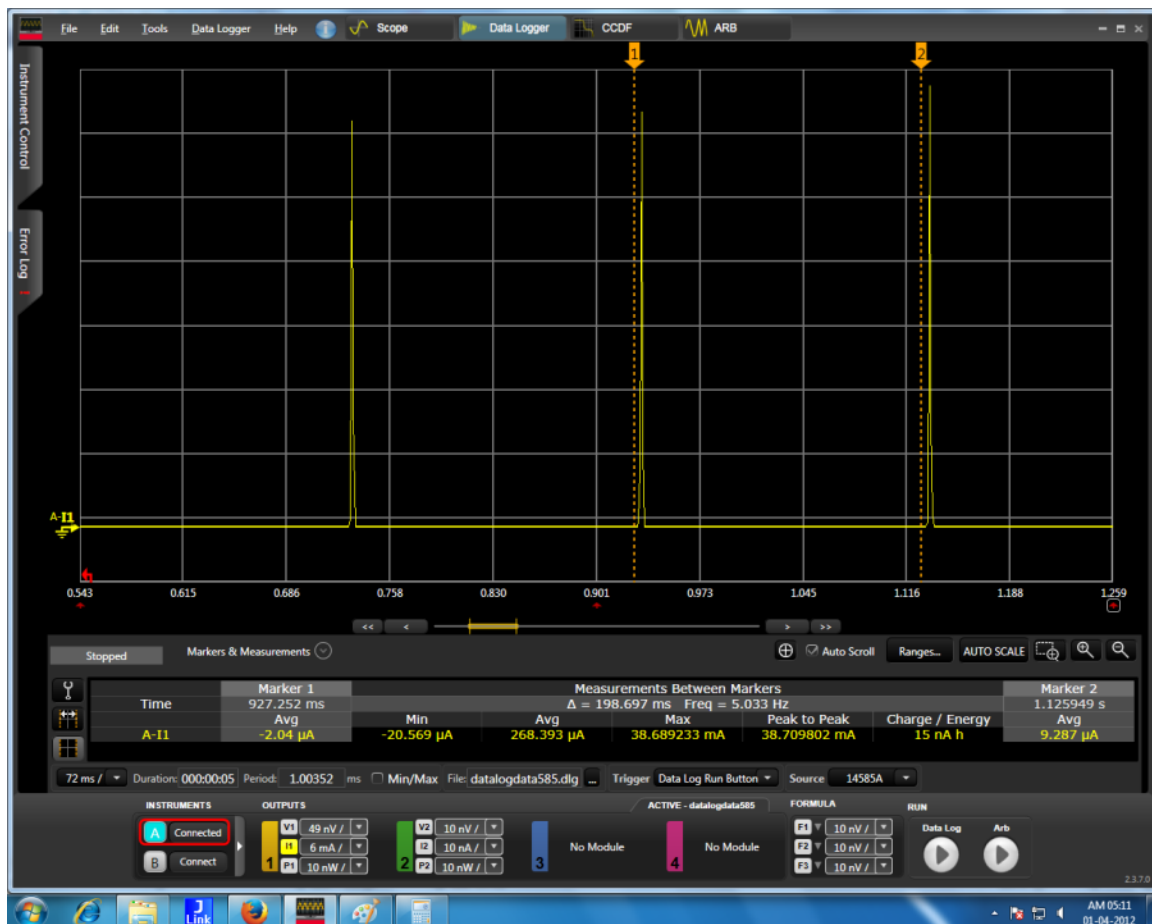


Figure 4: Connected Slave with Powersave

Note1:

The default configuration of connection interval of Master device (smartphone) is 18ms. So, WiSeMCU device will wake up for every 18ms sec and goes back to sleep after connection interval.

Note2:

Above power save profile image capture when it is in an idle state after successful connection. So, the user may not get same profile as shown above image. It will vary based on the traffic.

7. Verify whether Module goes to power save & advertising again After Successful Disconnection with Master device.

Note:

Install Light blue App for tablet iPad mini and BLE scanner app for an android smartphone.

3 BLE Test Mode Rx

3.1 Introduction

This example is applicable to WiSeConnect™ and WiSeMCU™ parts. For simplicity, this document refers to WiSeConnect, but all the discussion applies to both WiSeConnect and WiSeMCU parts. The feature(s) used in this example may or may not be available on your part. Refer to the product datasheet to verify the features available on your part.

3.2 Example

3.2.1 Overview

This application demonstrates how to configure a device for Test modes for active Rx cases.

3.2.2 Sequence of Events

This Application can be run in the role of Active Rx test modes.
Test mode explains to the user how to:

- Configure RX test mode & Keep M4 in sleep

3.3 Setup Diagrams

In WiseMCU ble only project redpine application will run on M4 processor RAM based. M4 itself acts as Host interface; here no external host interface is required.

3.3.1 Setup Requirements

- Windows PC with KIEL
- Redpine Module
- Third Party Dongle Which Supports the BLE Test Mode.

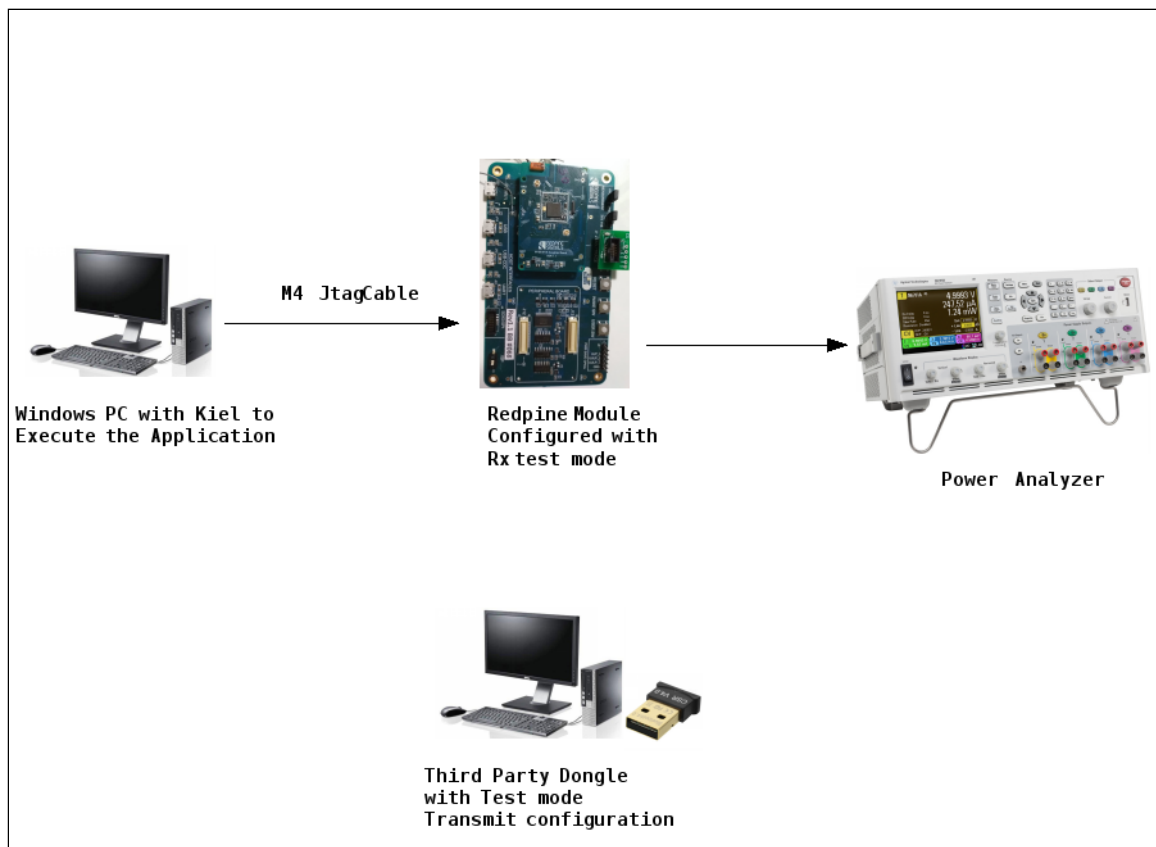


Figure 1: Setup Diagram for Rx test mode

3.4 Configuration and Execution of the Application

3.5 Configuring the Application

1. Open **ble_active_tx_pwr.c** file and update/modify following macros:

RSI_SEL_ANTENNA refers to the antenna which is to be used by Redpine module.
If the user using internal antenna then set,

```
#define RSI_SEL_ANTENNA RSI_SEL_INTERNAL_ANTENNA
```

If user using external antenna (U.FL connector) then set,

```
#define RSI_SEL_ANTENNA RSI_SEL_EXTERNAL_ANTENNA
```


RSI_BLE_RX_CHANNEL: Channel number in which packets will receive. Remote device also should transmit on the same channel as our module. (Range : 0x00 – 0x27).

```
#define RSI_BLE_RX_CHANNEL 0x10
```

RSI_BLE_TX_PHY is the phy rate of our module. Possibilities for this macro are

```
#define RSI_BLE_1MBPS          0x1
#define RSI_BLE_2MBPS          0x2
#define RSI_BLE_CODED_PHY      0x3
```

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      15000
```

RSI_BLE_TX_PHY is the phy rate of our module. Possibilities for this macro are

```
#define RSI_BLE_1MBPS          0x1
#define RSI_BLE_2MBPS          0x2
#define RSI_BLE_CODED_PHY      0x3
```

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      15000
```

2. Open **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_EXT_CUSTOM_FEATURE_BIT_MAP      MODE_256K
#define RSI_BAND                  RSI_BAND_2P4GHZ
```

3. Open ***rsi_ble_config.h*** file and update/modify following macros,

```
#define RSI_BLE_PWR_INX  
8  
#define RSI_BLE_PWR_SAVE_OPTIONS 0  
#define RSI_DUTY_CYCLING 0
```

Note:

rsi_wlan_config.h and rsi_ble_config.h files are already set with the desired configuration in respective example folders user need not change for each example.

3.6 Executing the Rx Test mode Application

1. After the program gets executed, Redpine module will be in Receive state with configured M4 power save profile.
2. Take Third party dongle and run TX test mode with the following command on Linux.
hcidtool -i hci0 cmd 0x08 0x001E 0x10 0x20 0x01
3. Verify whether packets are received to Redpine module before running following transmit stop command.
hcidtool -I hci0 cmd 0x08 0x001F
4. Please refer the given below image in Active Rx test mode.

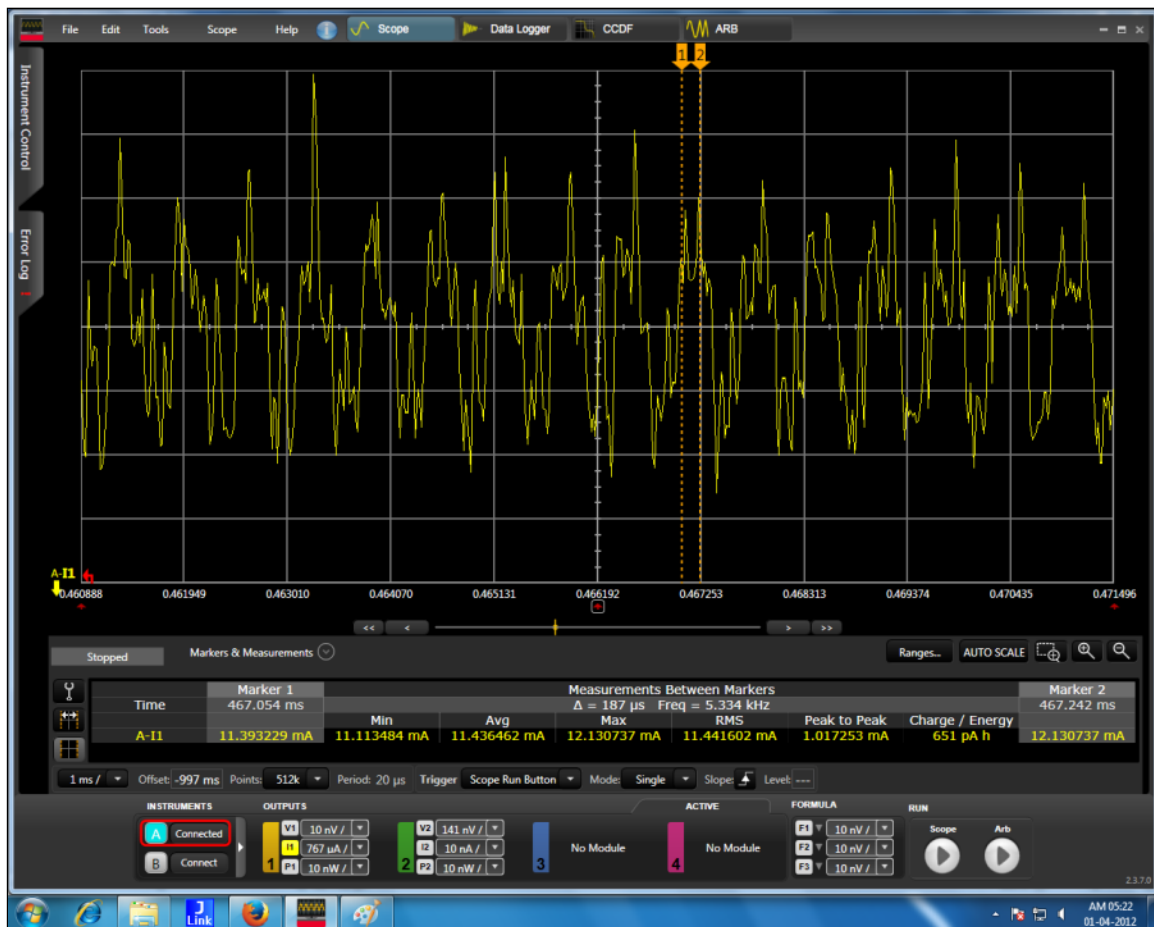


Figure 2: Active Rx test mode

4 BLE Test Mode Tx

4.1 Introduction

This example is applicable to WiSeConnect™ and WiSeMCU™ parts. For simplicity, this document refers to WiSeConnect, but all discussion applies to both WiSeConnect and WiSeMCU parts. The feature(s) used in this example may or may not be available on your part. Refer to the product datasheet to verify the features available on your part.

4.2 Example

4.2.1 Overview

This application demonstrates how to configure a device for Test modes for active Tx cases.

4.2.2 Sequence of Events

This Application can be run in the role of Active Tx test modes.
Test mode explains to the user how to:

- Configure TX test mode & Keep M4 in sleep

4.3 Setup Diagrams

In WiseMCU ble only project redpine application will run on M4 processor RAM based. M4 itself acts as Host interface; here no external host interface is required.

4.3.1 Setup Requirements

- Windows PC with KIEL
- Redpine Module
- Third Party Dongle Which Supports the BLE Test Mode.

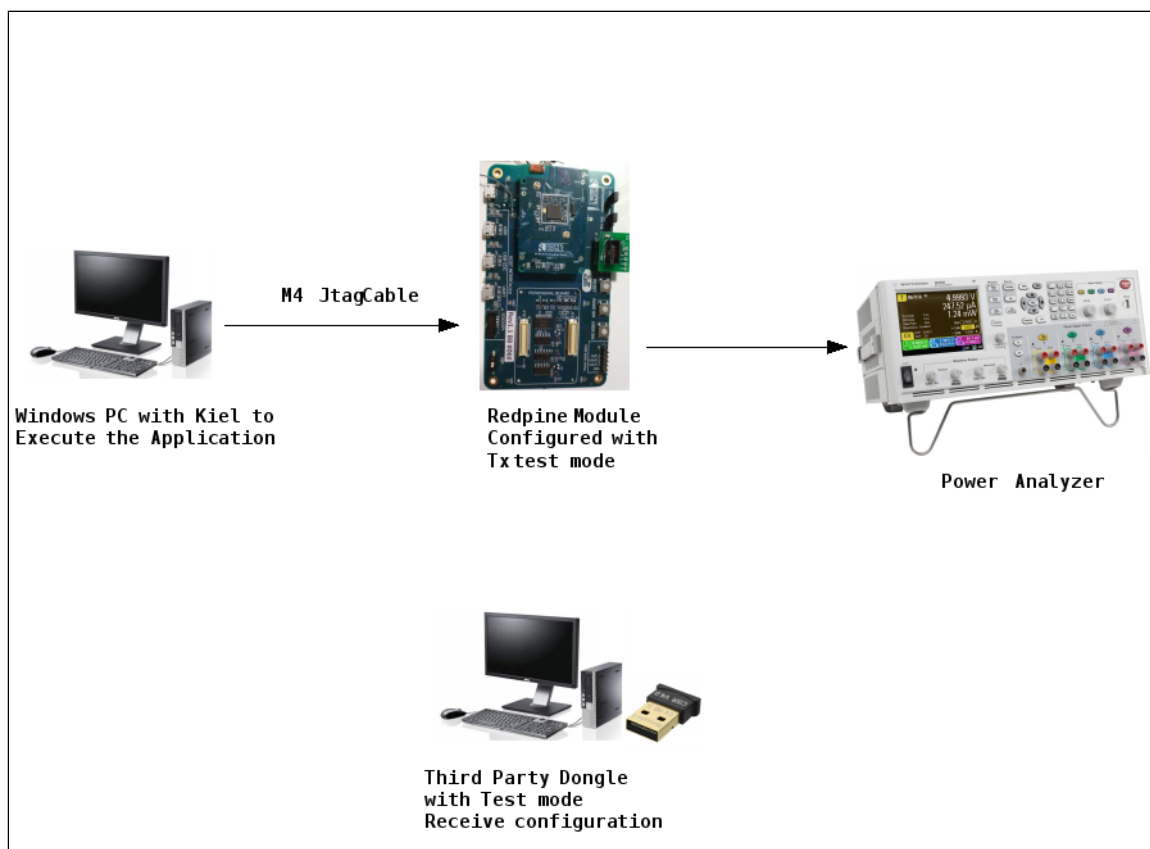


Figure 1: Setup Diagram for Tx Test mode

4.4 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.

4.4.1 Configuring the Active Tx Test Mode Application:

1. Open **ble_active_tx_pwr.c** file and update/modify following macros:

RSI_SEL_ANTENNA refers to the antenna which is to be used by Redpine module.
If the user using internal antenna then set,

```
#define RSI_SEL_ANTENNA          RSI_SEL_INTERNAL_ANTENNA
```

If the user using an external antenna (U.FL connector) then set,

```
#define RSI_SEL_ANTENNA RSI_SEL_EXTERNAL_ANTENNA
```

RSI_BLE_TX_CHANNEL: Channel number in which packets will transfer. Remote device also should receive in the same channel as our module. (Range : 0x00 – 0x27).

```
#define RSI_BLE_TX_CHANNEL 0x10
```

RSI_BLE_TX_PHY is the phy rate of our module. Possibilities for this macro are

```
#define RSI_BLE_1MBPS          0x1
#define RSI_BLE_2MBPS          0x2
#define RSI_BLE_125KBPS        0x3
#define RSI_BLE_500KBPS        0x4
```

RSI_BLE_TX_PAYLOAD_LEN: Length in bytes of payload data in each packet. Maximum payload length is 32 bytes.

```
#define RSI_BLE_PAYLOAD_LEN      20
```

RSI_BLE_TX_PAYLOAD_TYPE is the BLE payload type. Following are the possible macros

```
#define PRBS9_SEQ              0x0
#define FOUR_ONES_FOUR_ZEROS  0x1
#define ALT_ONES_AND_ZEROS     0x2
#define PRBS15_SEQ             0x3
#define ALL_ONES               0x4
#define ALL_ZEROS              0x5
#define FOUR_ZEROS_FOUR_ONES   0x6
#define ALT_ZERO_ALT_ONE       0x7
```

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      15000
```

2. Open **rsi_wlan_config.h** file and update/modify the 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_EXT_CUSTOM_FEATURE_BIT_MAP                MODE_256K
#define RSI_BAND                      RSI_BAND_2P4GHZ
```

3. Open **rsi_ble_config.h** file and update/modify following macros,

```
#define RSI_BLE_PWR_INX
8
#define RSI_BLE_PWR_SAVE_OPTIONS                0
#define RSI_DUTY_CYCLING                      0
```

Note:

rsi_wlan_config.h and rsi_ble_config.h files are already set with the desired configuration in respective example folders. The users are advised not change further.

4.4.2 Executing the Tx Test mode Application

1. After the program gets executed, Redpine module will be in Transmit state with configured M4 power save profile.
2. Take third party dongle and run RX test mode with the following command on Linux:
hcidtool -i hci0 cmd 0x08 0x001D 0x10 0x00
3. Verify whether the packets are received to the third-party dongle or not after running the following receive stop command:
hcidtool -I hci0 cmd 0x08 0x001F
4. Please refer to the below image in Active Tx test mode.
hcidtool -i hci0 cmd 0x08 0x001D 0x10 0x00

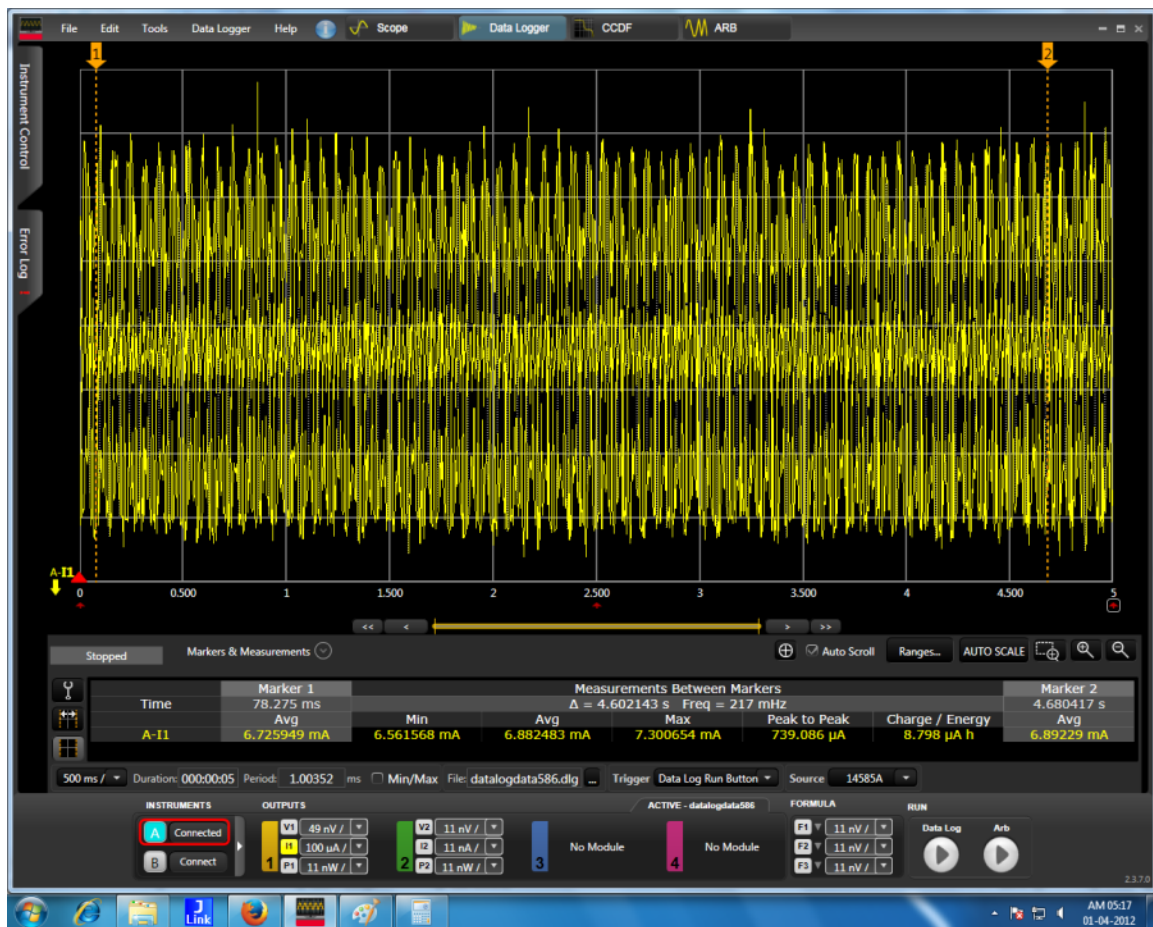


Figure 2: Active Tx test mode