RS9116W Wi-Fi AT Command Programming Reference Manual

Version 2.1

February 10, 2021

Table of Contents

About this Document

This document provides information on the RS9116 WiSeConnect architecture, boot-loader features, different command modes, and supported host interfaces.

It lists and describes AT commands to operate the RS9116W-WiSeConnect Module Family for Wi-Fi mode. Wi-Fi AT commands are provided along with the command parameters and their valid values with expected responses from the modules.

This document explains implementing application software on the Host to control and operate the RS9116W module.

Note:

This document should be used with WiSeConnect version 2.3.0.

1 Architecture Overview

1.1 Overview

RS9116 WiSeConnect module includes Wi-Fi, TCP/IP and BT 5 stacks embedded in its internal flash memory. This module requires a separate application processor which acts as a host.

Host can communicate with RS9116 module using one of the interfaces listed below.

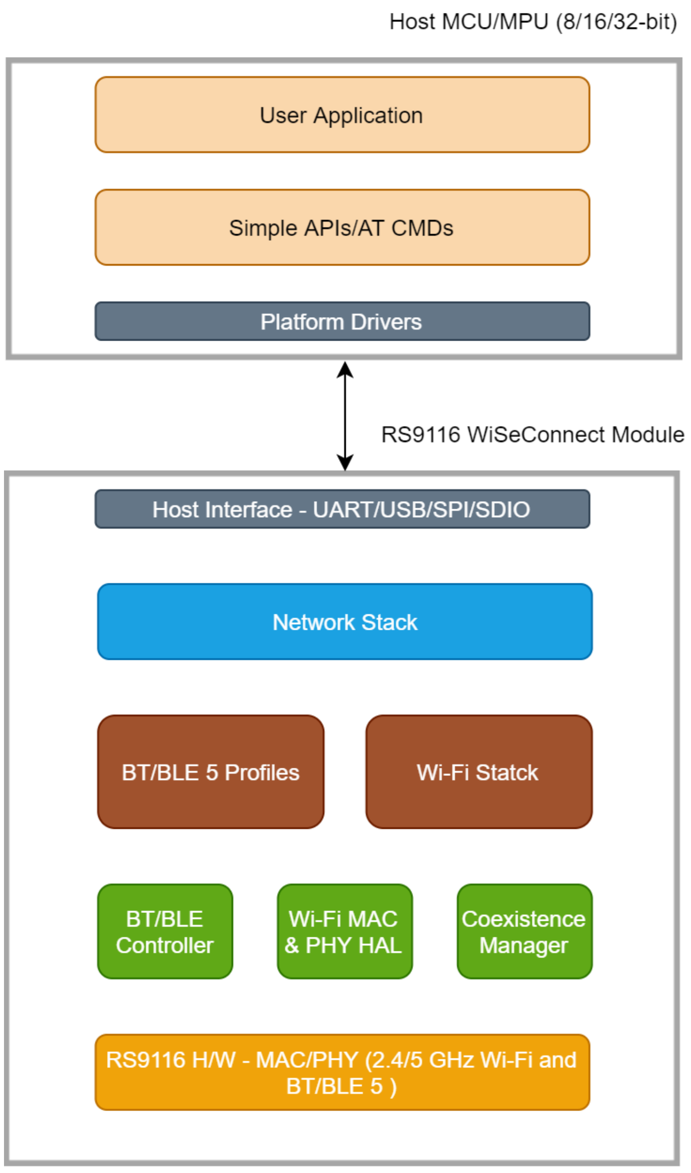


Figure 1: RS9116W Block Diagram

RS9116 supports following interfaces for interacting with host.

- SPI
- UART
- USB
- SDIO

Note:

USB and SDIO interfaces are currently not supported.

Host can use either Simple APIs or AT Commands for configuring the module.

There are two modes for host interaction:

- AT Mode (ASCII)
  In this mode, host should use AT Commands for interacting with RS9116 Module. This mode can be used only with UART or USB-CDC interfaces. Section **'AT Command Mode'** provides more details about this mode. Section **'WLAN Commands'** describes all commands and other required details.
- Binary Mode
  In this mode, host should use Binary Commands for interacting with RS9116 Module. This mode can be used with UART, SPI, USB and SDIO interfaces. Section **'Binary Command Mode**' provides more details about this mode. Section **'WLAN Commands'** describes all commands and other required details.

Host Interface block diagram is shown below,

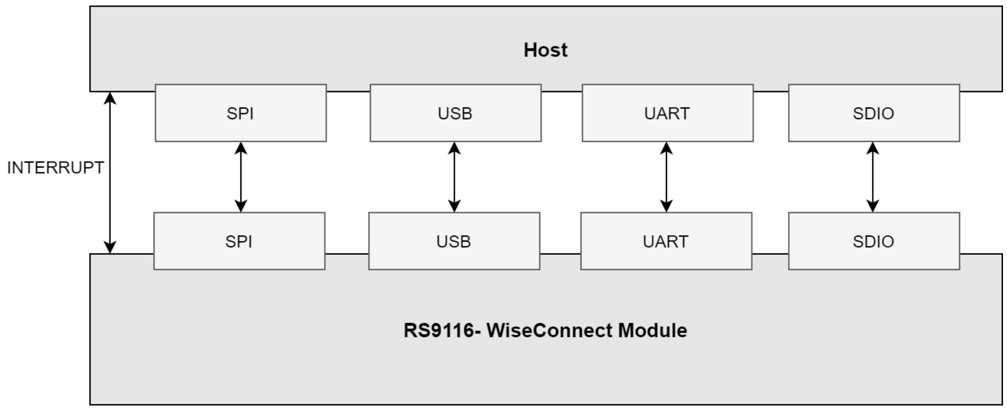


Figure 2: RS9116W Host Interface Block Diagram

1.2 WLAN Architecture

The following figure depicts the WLAN software architecture of the RS9116-WiSeConnect.

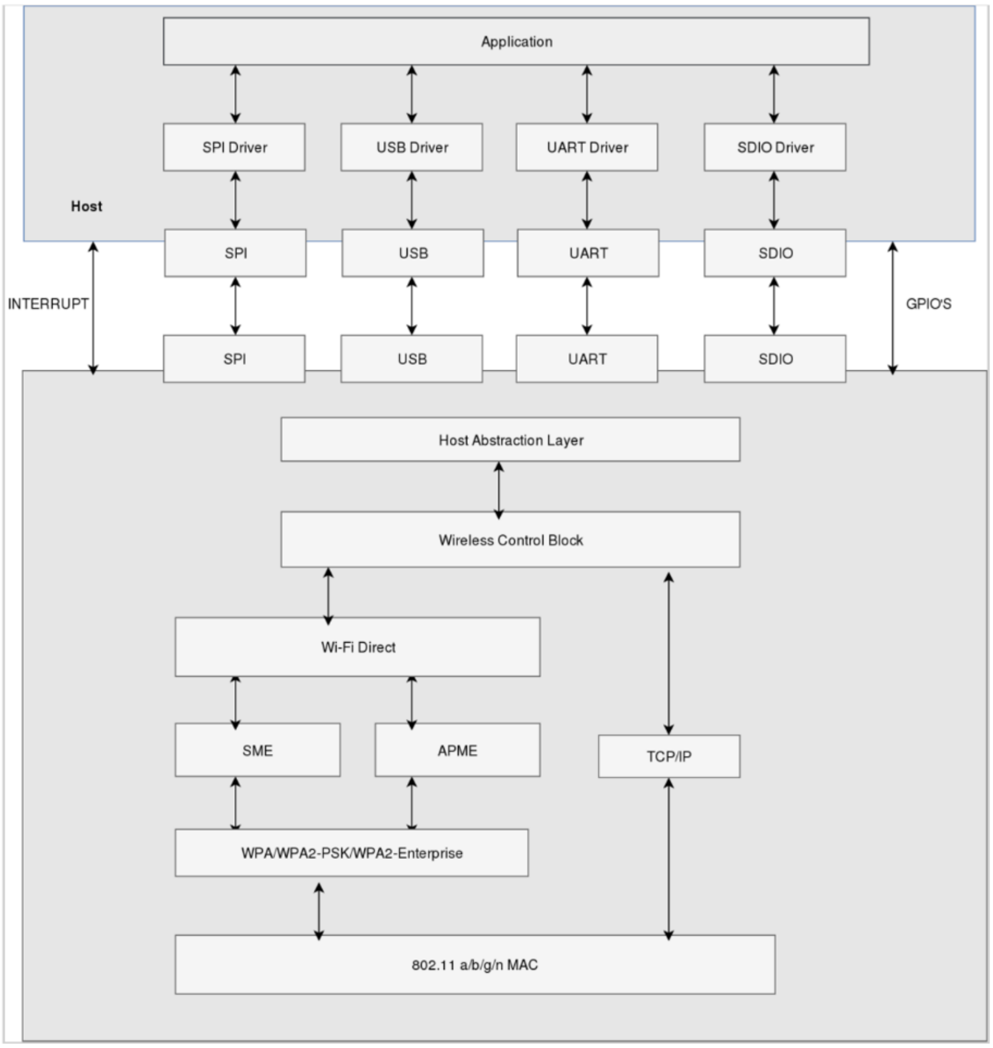


Figure 3: RS9116W Software Architecture

Application

The application layer invokes Simple APIs or AT Commands provided by SPI/ UART/USB/SDIO driver. The application developer can use these APIs to build wireless applications.

SPI

The SPI interface on the RS9116-WiSeConnect works in slave mode. It is a 4-wire interface. In addition to the SPI interface, the module provides additional interrupt pin to signal events to the host.
The interrupt is raised by the module in SPI mode for the following condition.

- When the module has data in its output buffer, it indicates host by raising active high signal on the interrupt pin.

The interrupt from module is active high and host must configure the interrupt in level trigger mode.

USB

The USB interface on the RS9116-WiSeConnect transmits/receives data to/from the Host in USB mode.

UART

The UART interface on the RS9116-WiSeConnect transmits/receives data to/from the Host in UART mode.

SDIO

The SDIO interface on the RS9116-WiSeConnect transmits/receives data to/from the Host in SDIO mode.

Host Abstraction Layer (HAL)

The HAL abstracts the lower layers in the Host interface to which the RS9116-WiSeConnect is connected. The HAL interacts with the Wireless Control Block layer for the processing of the frames obtained from or destined to the Host.

Wireless Control Block (WCB)

The data from/to the Host is classified as Wi-Fi specific frames or TCP/IP specific frames. The functionality of the WCB module depends on the type and direction of the frames.

Wi-Fi Control Frames (WCF)

The WCB interprets the Wi-Fi control information from the Host and interacts with the SME (Station Management Entity) or APME (Access Point Management Entity) based on operating mode of RS9116-WiSeConnect.
Configuration of the RS9116-WiSeConnect from the Host for Wi-Fi access is through AT commands or Binary commands.

TCP/IP Control Frames

TCP/IP networking protocol provides end-to-end connectivity specifying how data should be formatted, addressed, transmitted, routed and received at the destination.
If the packets received from the Host by WCB during transmission is interpreted as TCP/IP frames, then the WCB interacts with TCP/IP stack for further processing before passing the packets to MAC layer. Similarly, if the packets are received from the TCP/IP stack by WCB during reception then WCB processes by passing these packets to host.

Station Management Entity (SME)

The SME is the core layer which manages the Wi-Fi connectivity in Station mode. The SME maintains the state machine to detect the activity on the Wi-Fi network and indicates to the user accordingly. It interacts with the WPA supplicant if security is enabled in the Wi-Fi network.

Access point Management Entity (APME)

The APME is the core layer which manages the connectivity in Access Point group owner modes. The APME maintains the state machine to handle multiple clients connected to the module. It interacts with WPA supplicant if security is enabled in the Wi-Fi network.

WPA Supplicant

The main functionality of WPA supplicant is to support the key negotiation between Wi-Fi devices in a secure mode. This functionality depends on the mode in which RS9116-WiSeConnect operates in Station mode, Access point mode. The WPA supplicant is used to initiate the 802.1x/E Access Point authentication if WPA/WPA2-PSK is used as the security parameter. It also plays a major part in performing the 4-way handshake to derive the PTK in WPA/WPA2-PSK modes.

2 Bootloader

This section briefs about features supported by the Network and Security Processor's (NWP) bootloader.

Basic Features

- Load default firmware
- Load selected firmware
- Upgrade firmware from the host
- Selecting default images
- Enable / Disable host interaction bypass
- Support for multiple host interfaces (SDIO/SPI/UART/USB/USB-CDC)
- Firmware integrity check
- Upgrading keys
- JTAG selection

The RS9116W supports two boot loading modes. They are:

1. **Host interaction (Non-bypass) Mode:** In this mode, the host can interact with the bootloader and can give boot up options (commands) to configure different bootup operations. The host tells the module what operations it must perform based on the selections made by the user.
2. **Bypass mode:** In this mode, bootloader interactions are completely bypassed and uses the stored bootup configurations (which are selected in host interaction mode) & loads default firmware image in the module. This mode is recommended for final production of software to minimize the bootup time.

Host Interaction Mode

In Host Interaction mode, host interaction varies based upon host interface. Host interaction in SPI/USB and UART/USB-CDC are different. In UART & USB-CDC boot-up options are menu based. In SPI/USB, it uses command exchanges. The details are explained in the below section.

Host Interaction Mode in UART / USB-CDC

This section explains the host interaction mode in the UART/USB CDC mode.

Startup Operation

After powering up, host is required to carry out ABRD (Auto baud rate detection) operation. After successful ABRD, the module displays the menu of bootup options to host. The host needs to select the appropriate option.

Note:
On powerup, bootloader checks the integrity of the bootup options. If the integrity fails, it computes the integrity from backup. If integrity passes, it copies the backup to the actual location. If the integrity of the backup options also fails, the bootup options are reset/cleared. In either of the cases, bootloader bypass is disabled, or corresponding error messages are given to host. In case of integrity failure and when the backup integrity check passes, "LAST CONFIGURATION NOT SAVED" message is displayed. When backup integrity also fails, “BOOTUP OPTIONS CHECKSUM FAILED" is displayed before displaying the bootup options.

Hyper Terminal Configuration

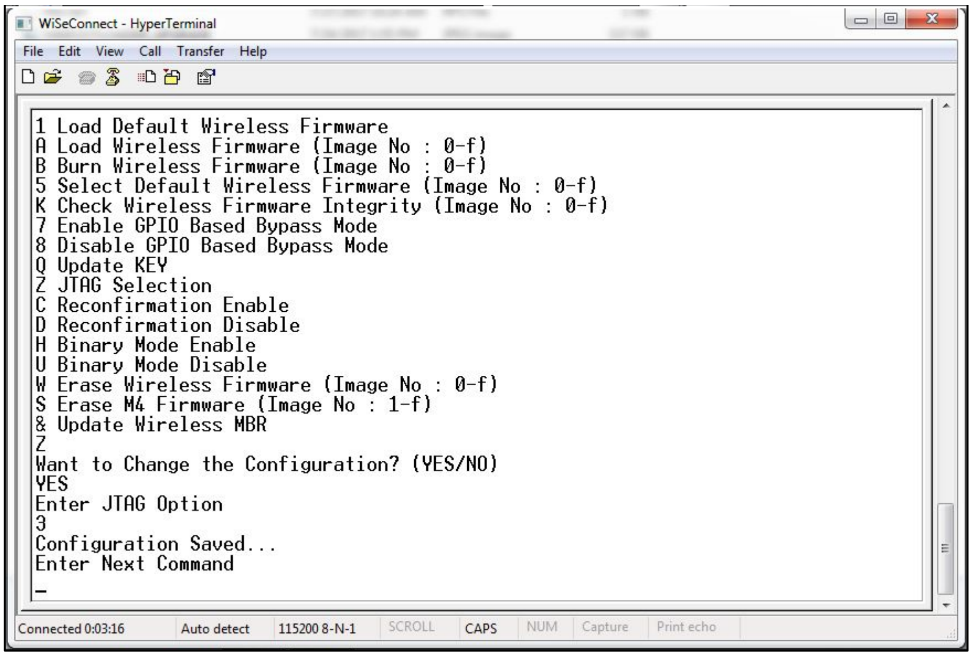
RS9116W uses the following UART interface configuration for communication:

Baud rate: The different baud rates supported by the module are: 9600 bps, 19200 bps, 38400 bps, 57600 bps, 115200 bps, 230400 bps, 460800 bps, 921600 bps.
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

Before the module is powered up, follow the sequence of steps given below:

- Open Hyper terminal and enter any name in the **"Name"** field. After this, click **"OK"** button. Here, **"WiSeConnect"** is entered as shown in the figure below.

Note:
Default baud rate of the module is 115200.

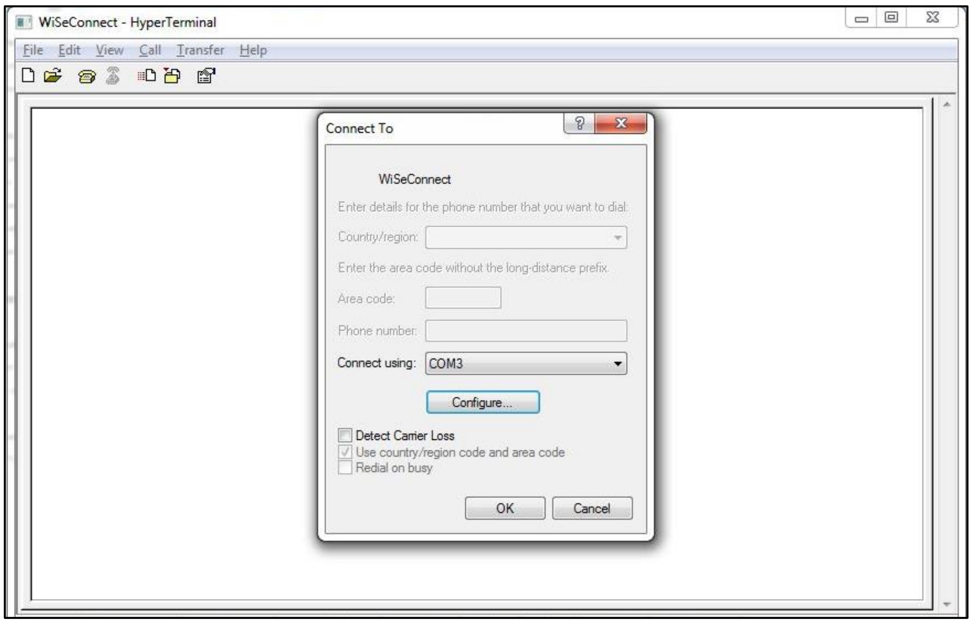

WiSeConnect - HyperTerminal
File Edit View Call Transfer Help

```
1 Load Default Wireless Firmware
A Load Wireless Firmware (Image No : 0-f)
B Burn Wireless Firmware (Image No : 0-f)
5 Select Default Wireless Firmware (Image No : 0-f)
K Check Wireless Firmware Integrity (Image No : 0-f)
7 Enable GPIO Based Bypass Mode
8 Disable GPIO Based Bypass Mode
Q Update KEY
Z JTAG Selection
C Reconfirmation Enable
D Reconfirmation Disable
H Binary Mode Enable
U Binary Mode Disable
W Erase Wireless Firmware (Image No : 0-f)
S Erase M4 Firmware (Image No : 1-f)
& Update Wireless MBR
Z
Want to Change the Configuration? (YES/NO)
YES
Enter JTAG Option
3
Configuration Saved...
Enter Next Command
_
```

Connected 0:03:16 | Auto detect | 115200 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

Figure 4: HyperTerminal Name Field Configuration

- After clicking **"OK"**, the following dialog box is displayed as shown in the figure below.


WiSeConnect - HyperTerminal
File Edit View Call Transfer Help
Connect To
Configure...
OK
Cancel

WiSeConnect

Enter details for the phone number that you want to dial:

Country/region:

Enter the area code without the long-distance prefix.

Area code:

Phone number:

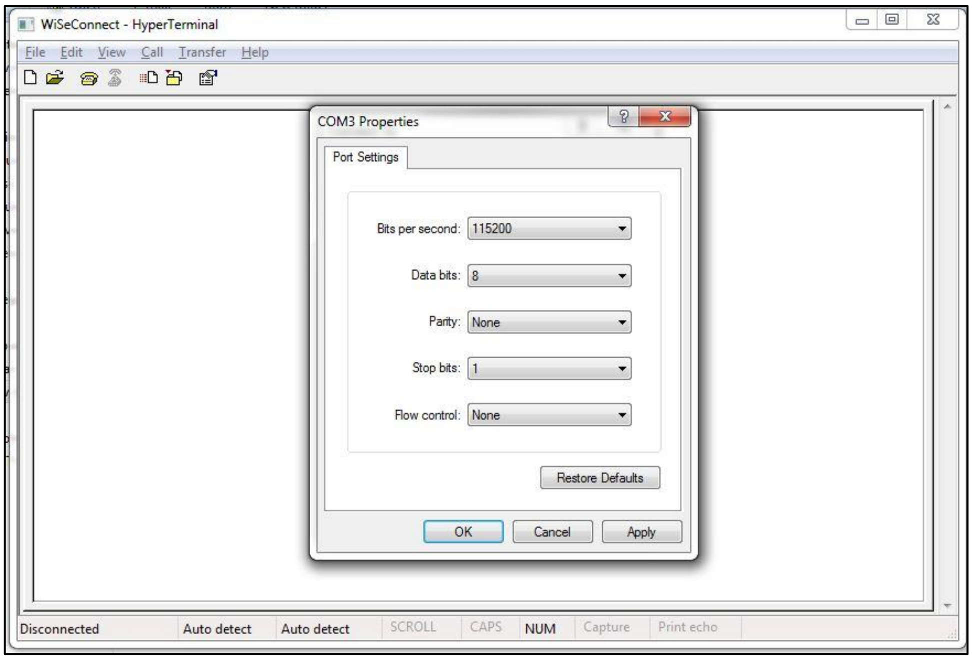
Connect using: COM3

☐ Detect Carrier Loss
☑ Use country/region code and area code
☐ Redial on busy

Figure 5: HyperTerminal COM Port Field Configuration

- In the **"Connect using"** field, select appropriate com port. In the figure above COM3 is selected. Click **"OK"** button.

- After clicking the **"OK"** button the following dialog box is displayed as shown in the figure below.



Bits per second: 115200
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

Figure 6: HyperTerminal Baud Rate Field Configuration

Set the following values for different fields in figure 5 as given below.

- Set baud rate to 115200 in "Bits per second" field.
- Set Data bits to 8 in "Data bits" field.
- Set Parity to None in "Parity" field.
- Set stop bits to 1 in "Stop bits" field.
- Set flow control to None in "Flow control" field.
- Click "OK" button after entering the data in all the fields.

Auto Baud Rate Detection (ABRD)

The RS9116W automatically detects the baud rate of the Host's UART interface by exchanging some bytes. The Host should configure the UART interface for the following parameters for ABRD detection.
RS9116W uses the following UART interface configuration for communication:

Baud Rate: The following baud rates are supported: 9600 bps, 19200 bps, 38400 bps, 57600 bps, 115200 bps, 230400 bps, 460800 bps, 921600 bps.
Data bits: 8
Stop bits: 1
Parity: None
Flow control: None

To perform ABRD on the RS9116W, the host must follow the procedure outlined below.

1. Configure the UART interface of the Host at desired baud rate.
2. Power on the RS9116W.
3. The Host, after releasing the module from reset, should wait for 20 ms for initial boot-up of the module to complete and then transmit 0x1C at the baud rate to which its UART interface is configured. After transmitting '0x1C' to the module, the Host should wait for the module to transmit 0x55 at the same baud rate.
4. If the '0x55' response is not received from the module, the host has to re-transmit 0x1C, after a delay of 200ms.
5. After finally receiving '0x55', the host should transmit '0x55' to the module. The module is now configured with the intended baud rate.

Note:
Performing ABRD in host interaction mode is must for USB CDC mode.

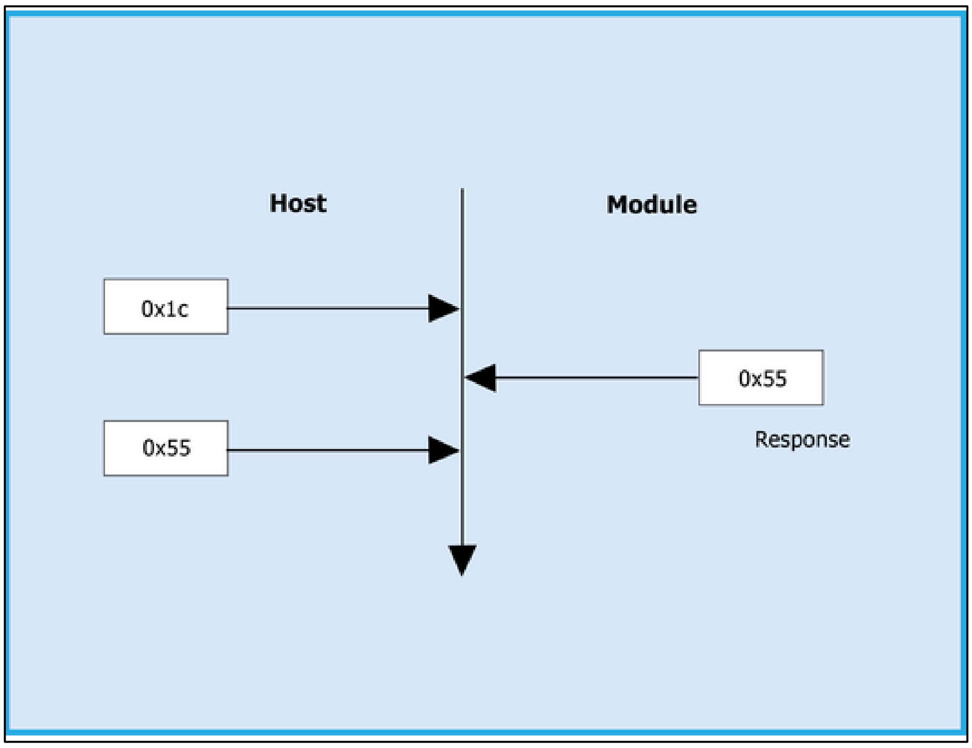


Figure 7: ABRD Exchange between Host and Module

Below are the bootup options, Firmware upgrade and Firmware loading procedures for WiSeConnect Product.

Start-Up Messages on Power-Up

After powering up the module, do the ABRD process by giving "shift + | " followed by 'U'. Pipe symbol is shown below, you will see a welcome message on the host, followed by boot up options:

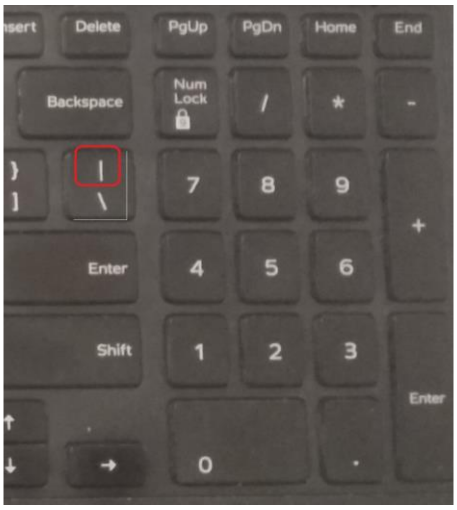
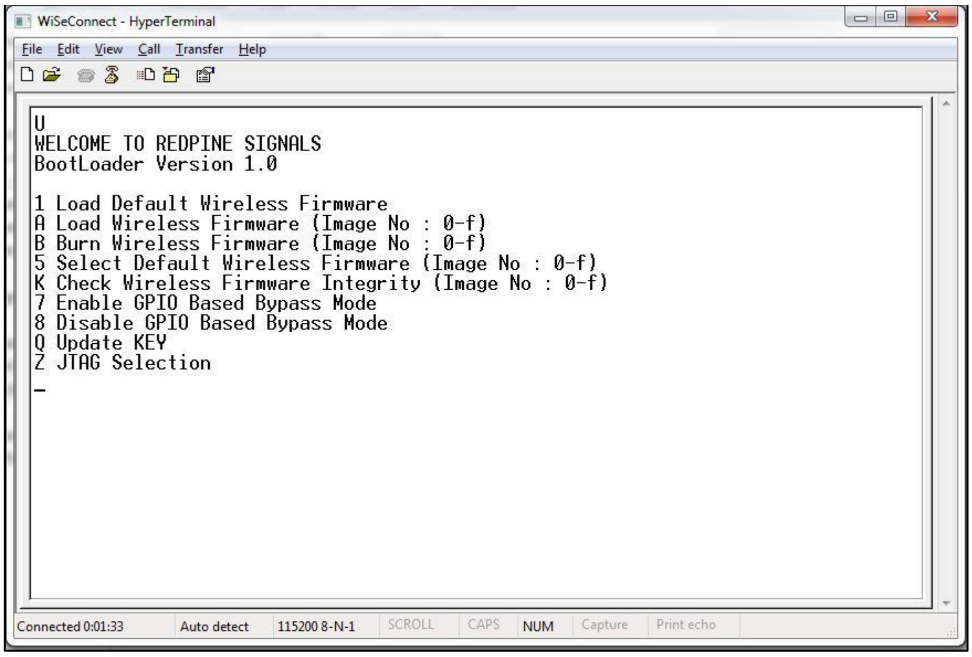


Figure 8: Pipe Symbol on Keyboard

Note:
Windows Hyper Terminal is used to demonstrate the boot up /up-gradation procedure.


WiSeConnect - HyperTerminal
File Edit View Call Transfer Help
Connected 0:01:33 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo

```
U
WELCOME TO REDPINE SIGNALS
BootLoader Version 1.0

1 Load Default Wireless Firmware
A Load Wireless Firmware (Image No : 0-f)
B Burn Wireless Firmware (Image No : 0-f)
5 Select Default Wireless Firmware (Image No : 0-f)
K Check Wireless Firmware Integrity (Image No : 0-f)
7 Enable GPIO Based Bypass Mode
8 Disable GPIO Based Bypass Mode
Q Update KEY
Z JTAG Selection
_
```

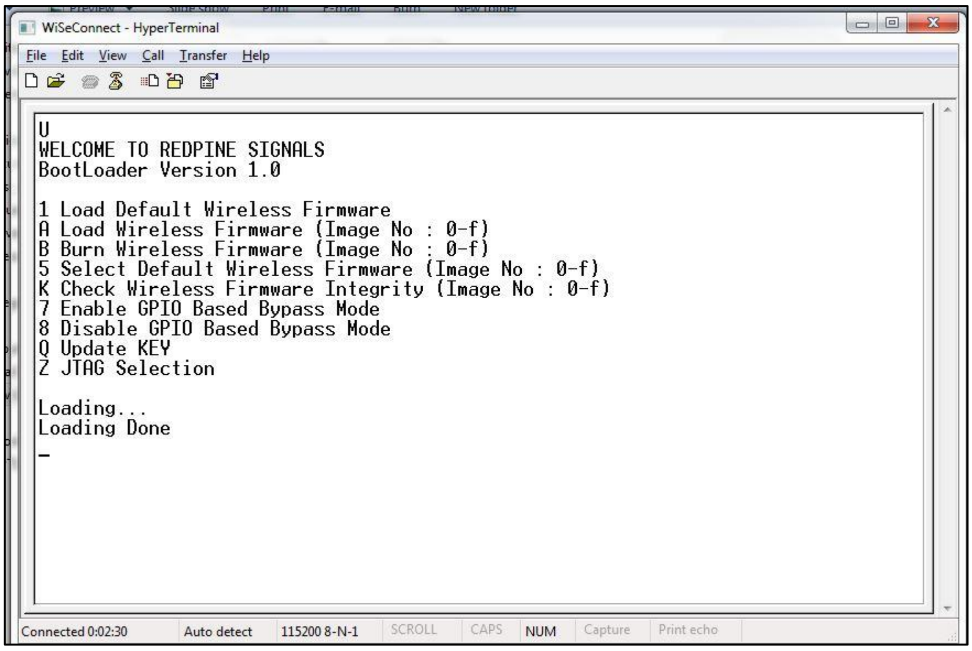
Figure 9: RS9116-WiSeConnect Module UART / USB-CDC Welcome Message

Loading the Default Wireless Firmware in the Module

To load the default firmware flashed onto the module, choose Option 1: "Load Default Wireless Firmware ".

Load Default Wireless Firmware

- After the welcome message is displayed as shown in the above figure, select option 1 "Load Default Wireless Firmware" for loading Image.


WiSeConnect - HyperTerminal
File Edit View Call Transfer Help
Connected 0:02:30 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo

```
U
WELCOME TO REDPINE SIGNALS
BootLoader Version 1.0

1 Load Default Wireless Firmware
A Load Wireless Firmware (Image No : 0-f)
B Burn Wireless Firmware (Image No : 0-f)
5 Select Default Wireless Firmware (Image No : 0-f)
K Check Wireless Firmware Integrity (Image No : 0-f)
7 Enable GPIO Based Bypass Mode
8 Disable GPIO Based Bypass Mode
Q Update KEY
Z JTAG Selection

Loading...
Loading Done
_
```

Figure 10: RS9116-WiSeConnect Module UART / USB-CDC Default Firmware Loaded

Note:
By default, the module will be configured in AT mode. If mode switch from AT plus command mode to binary mode is required, then user must give 'H' in the bootloader options.
The module lasts in the binary mode unless it changed to AT plus command mode and vice-versa.
To change from binary mode to AT mode, then user must give 'U' in the bootloader options.

Loading selected Wireless Firmware in the Module

To load the selected firmware (from flash) onto the module, choose Option A: "Load Wireless Firmware (Image No: 0-f)".

Load Wireless Firmware

- After the welcome message is displayed as shown in the above figure, select option A "Load Wireless Firmware (Image No: 0-f)" for loading Image.
- In response to the option A, Module asks to Enter Image No.
- Select the image number to be loaded from flash.
- After successfully loading the default firmware, "Loading Done" message is displayed.
- After firmware loading is completed, the module is ready to accept commands.

Note:

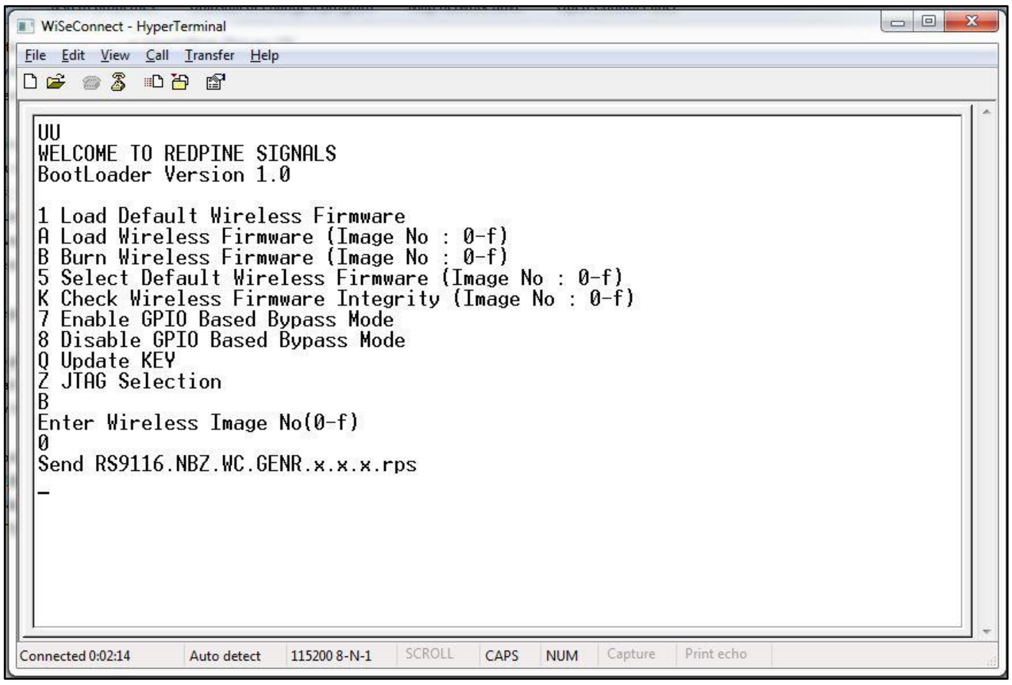
1. In order to use host bypass mode, the user must select one of the images as default image by selecting option 5 (Select Default Wireless Firmware).
2. In Host interaction mode, if no option is selected after bootup menu for 20 seconds then the bootloader will load selected Wireless default image.
3. If the valid firmware is not present, then a message prompts "Valid firmware not present".

Firmware Upgradation

After powering up the module, a welcome message is displayed.

Upgrade NWP firmware Image

- After the welcome message is displayed, select option B "Burn Wireless Firmware (Image No: 0-f)" to upgrade Wireless Image.
- The message "Enter Wireless Image No (0-f)" is displayed.
- Then select the Image no to be upgraded.
- The message "Send RS9116.NBZ.WC.GENR.x.x.x.rps" should appear as shown in the figure below.



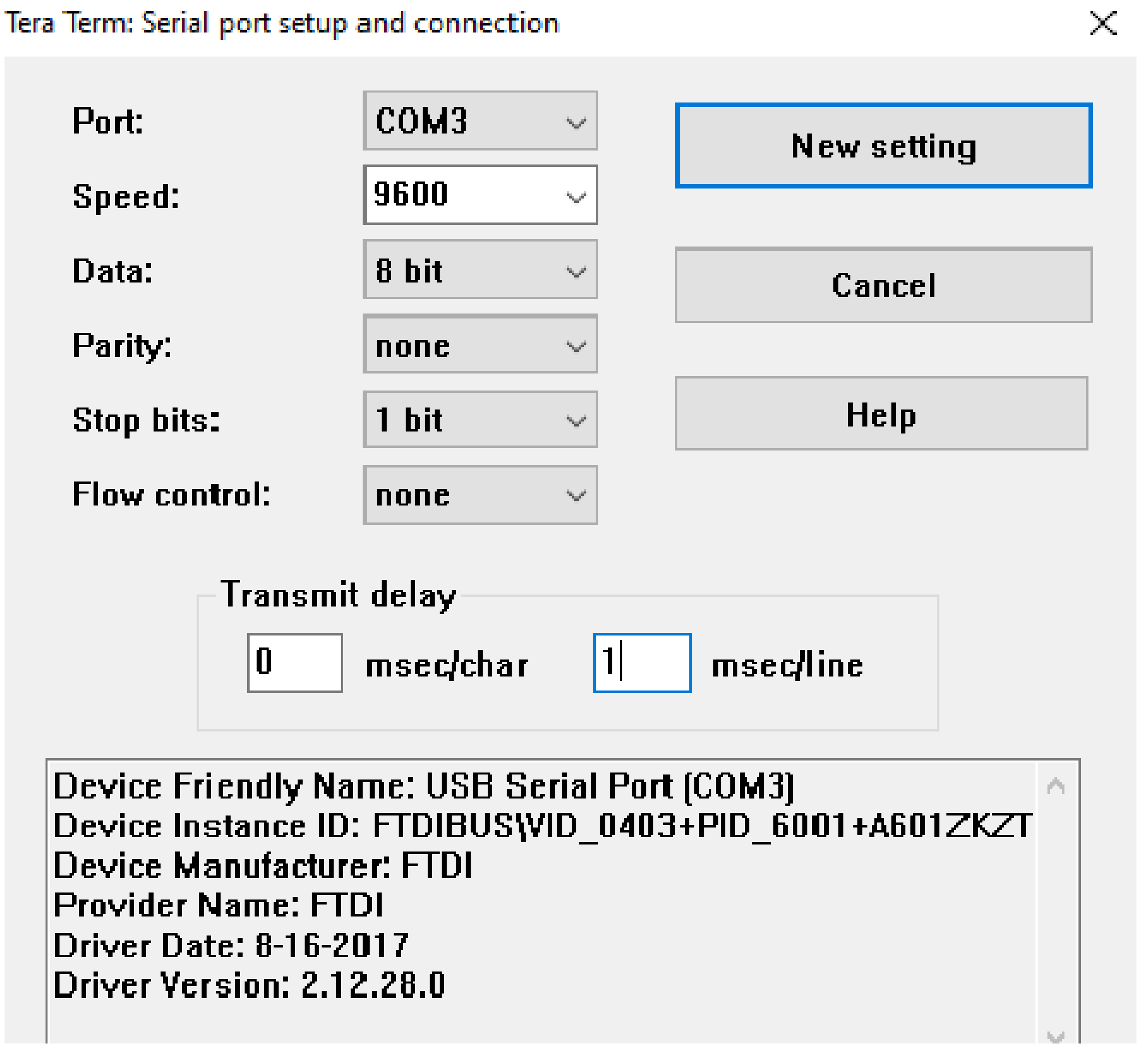
```
UU
WELCOME TO REDPINE SIGNALS
BootLoader Version 1.0

1 Load Default Wireless Firmware
A Load Wireless Firmware (Image No : 0-f)
B Burn Wireless Firmware (Image No : 0-f)
5 Select Default Wireless Firmware (Image No : 0-f)
K Check Wireless Firmware Integrity (Image No : 0-f)
7 Enable GPIO Based Bypass Mode
8 Disable GPIO Based Bypass Mode
Q Update KEY
Z JTAG Selection
B
Enter Wireless Image No(0-f)
0
Send RS9116.NBZ.WC.GENR.x.x.x.rps
_
```

Figure 11: RS9116-WiSeConnect Module Firmware Upgrade File Prompt Message

Note:

On faster machine using Tera Term it is recommended to configure transmit delay to avoid file transfer to freeze.
After detection of serial port, change transmit delay from 0 msec/line to 1 msec/line to get 'firmware upgradation successful' as shown in the figure below.



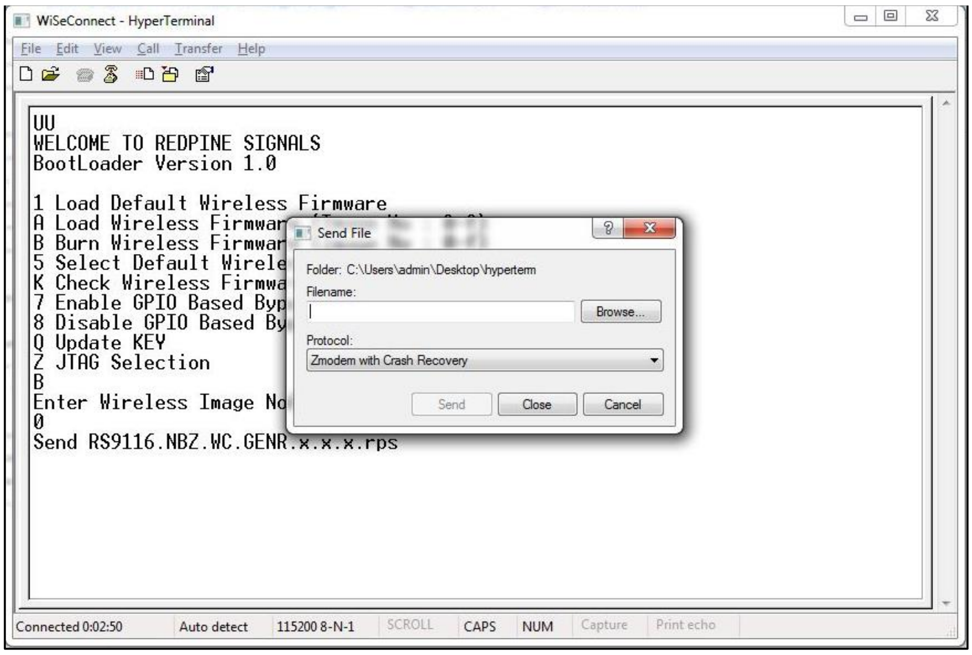
Port: COM3
Speed: 9600
Data: 8 bit
Parity: none
Stop bits: 1 bit
Flow control: none

Transmit delay: 0 msec/char, 1 msec/line

```
Device Friendly Name: USB Serial Port (COM3)
Device Instance ID: FTDIBUS\VID_0403+PID_6001+A601ZKZT
Device Manufacturer: FTDI
Provider Name: FTDI
Driver Date: 8-16-2017
Driver Version: 2.12.28.0
```

Figure 12: Serial Port Setup and Connection

- In the "File" menu of HyperTerminal, select the "send file" option. A dialog box will appear as shown in the figure below. Browse to the path where "RS9116.NBZ.WC.GENR.X.X.X.rps" is located and select Kermit as the protocol option. After this, click the "Send" button to transfer the file.
- If the valid firmware is not present, then a message prompts "Valid firmware not present".



```
UU
WELCOME TO REDPINE SIGNALS
BootLoader Version 1.0

1 Load Default Wireless Firmware
A Load Wireless Firmwar
B Burn Wireless Firmwar
5 Select Default Wirele
K Check Wireless Firmwa
7 Enable GPIO Based Byp
8 Disable GPIO Based By
Q Update KEY
Z JTAG Selection
B
Enter Wireless Image No
0
Send RS9116.NBZ.WC.GENR.x.x.x.rps
```

Send File

Folder: C:\Users\admin\Desktop\hyperterm

Filename:

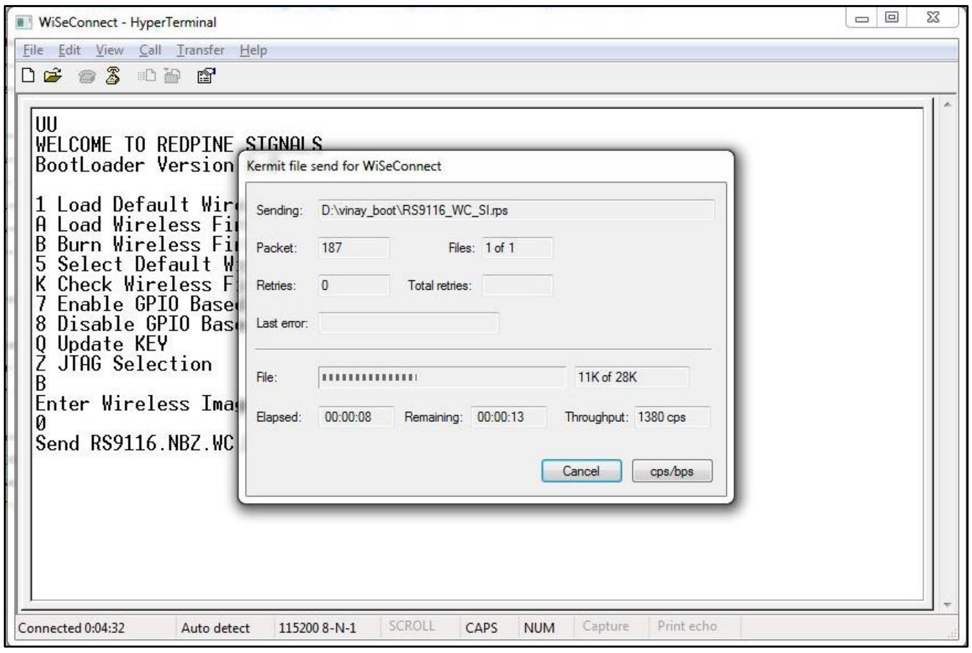
Protocol: Zmodem with Crash Recovery

Connected 0:02:50 | Auto detect | 115200 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

WiSeConnect - HyperTerminal
File Edit View Call Transfer Help
Browse...
Send
Close
Cancel

Figure 13: RS9116-WiSeConnect Module Firmware Upgrade File Selection Message

- The dialog box message is displayed while file transfer is in progress as shown in the figure below.



```
UU
WELCOME TO REDPINE SIGNALS
BootLoader Version

1 Load Default Wir
A Load Wireless Fi
B Burn Wireless Fi
5 Select Default W
K Check Wireless F
7 Enable GPIO Base
8 Disable GPIO Bas
Q Update KEY
Z JTAG Selection
B
Enter Wireless Ima
0
Send RS9116.NBZ.WC
```

Kermit file send for WiSeConnect

Sending: D:\vinay_boot\RS9116_WC_SI.rps

Packet: 187 Files: 1 of 1

Retries: 0 Total retries:

Last error:

File: 11K of 28K

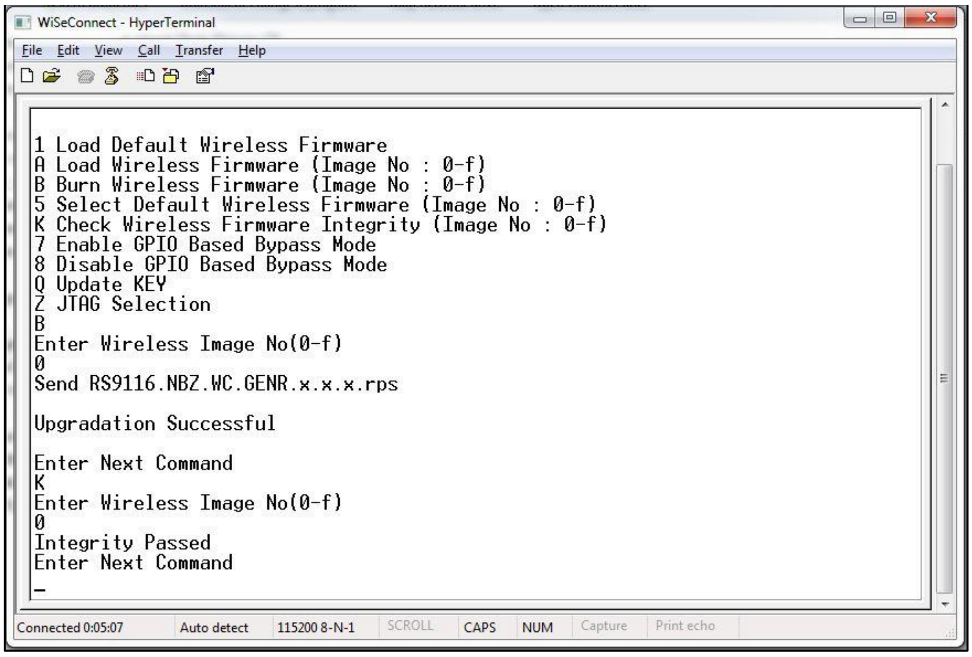
Elapsed: 00:00:08 Remaining: 00:00:13 Throughput: 1380 cps

Connected 0:04:32 | Auto detect | 115200 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

WiSeConnect - HyperTerminal
File Edit View Call Transfer Help
Cancel
cps/bps

Figure 14: RS9116-WiSeConnect Module Firmware Upgrade File Transfer Message

- After successfully completing the file transfer, module computes the integrity of the image and displays "Upgradation Failed, re-burn the image" in case of failure. It displays "Upgradation Failed and default image invalid, Bypass disabled" in case of both failure and corruption of the default image.
- In the case of success, module checks if the bootloader bypass is enabled and computes the integrity of the default image selected. If the integrity fails, it sends "Upgradation successful, Default image invalid, gpio bypass disabled." If integrity passes or GPIO bypass not enabled, it sends "Upgradation Successful" message on the terminal as shown in the figure below.


WiSeConnect - HyperTerminal
File Edit View Call Transfer Help

```
1 Load Default Wireless Firmware
A Load Wireless Firmware (Image No : 0-f)
B Burn Wireless Firmware (Image No : 0-f)
5 Select Default Wireless Firmware (Image No : 0-f)
K Check Wireless Firmware Integrity (Image No : 0-f)
7 Enable GPIO Based Bypass Mode
8 Disable GPIO Based Bypass Mode
Q Update KEY
Z JTAG Selection
B
Enter Wireless Image No(0-f)
0
Send RS9116.NBZ.WC.GENR.x.x.x.rps

Upgradation Successful

Enter Next Command
K
Enter Wireless Image No(0-f)
0
Integrity Passed
Enter Next Command
_
```

Figure 15: RS9116-WiSeConnect Module Firmware Upgrade Completion Message

- At this point, the upgraded firmware Image is successfully flashed to the module.
- The user can again cross check the integrity of the Image by selecting the Option K "Check Wireless Firmware Integrity (Image No: 0-f)" for Wireless Image.
- Follow the steps mentioned in **Loading the Default Wireless Firmware in the Module** to load the firmware from flash, select Option 1 from the above figure.
- The module is ready to accept commands from the Host.

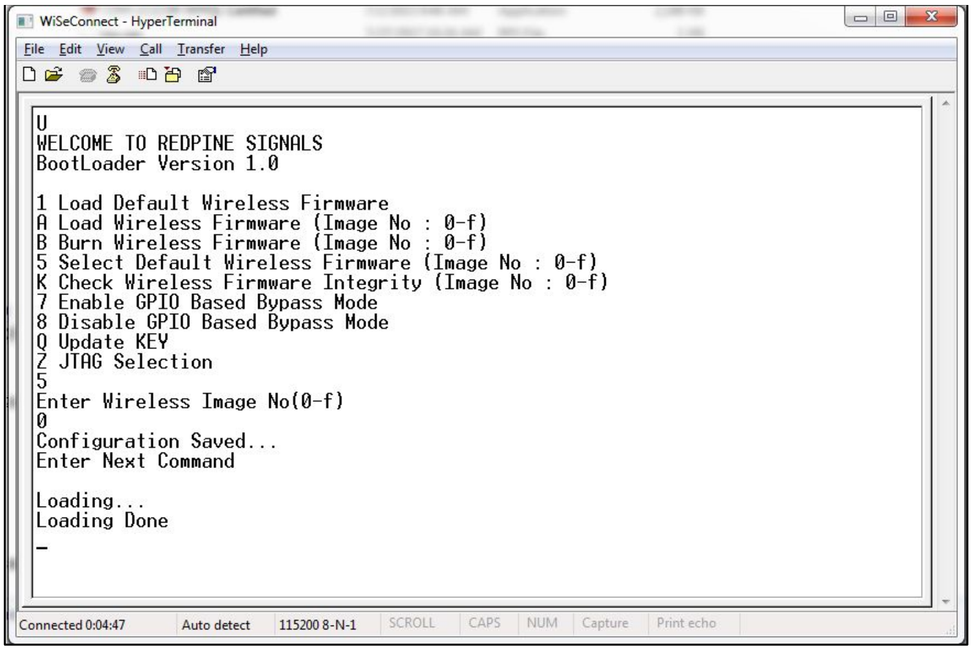
Bypass Mode in UART / USB-CDC

Making Default Wireless Firmware Selection

With this option, host can select the default firmware image to be loaded.

Selecting a valid Image as the Default Image

- After the welcome message is displayed, user can select option 5 "Select Default Wireless Firmware (Image No: 0-f)".
- The message "Enter Wireless Image No. (0-f)" is displayed.
- Then select the Image number
- It is better to check the Integrity of Image before selecting it as Default Image.
- When default image is selected, module checks for the validity of the image selected and displays "Configuration saved".


WiSeConnect - HyperTerminal
File Edit View Call Transfer Help
Connected 0:04:47 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo

```
U
WELCOME TO REDPINE SIGNALS
BootLoader Version 1.0

1 Load Default Wireless Firmware
A Load Wireless Firmware (Image No : 0-f)
B Burn Wireless Firmware (Image No : 0-f)
5 Select Default Wireless Firmware (Image No : 0-f)
K Check Wireless Firmware Integrity (Image No : 0-f)
7 Enable GPIO Based Bypass Mode
8 Disable GPIO Based Bypass Mode
Q Update KEY
Z JTAG Selection
5
Enter Wireless Image No(0-f)
0
Configuration Saved...
Enter Next Command

Loading...
Loading Done
_
```

Figure 16: Making Image no - 0 as the Default Image

Enable/Disable GPIO Based Bypass Option

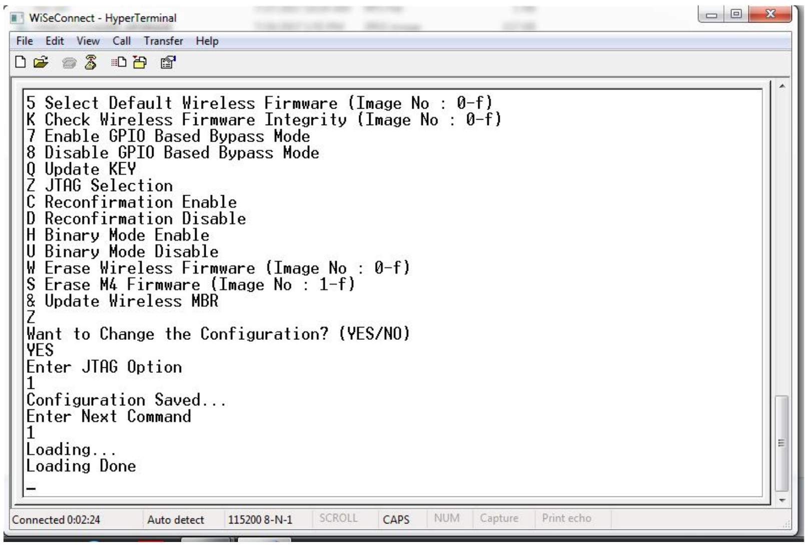
This option is for enabling or disabling the GPIO bootloader bypass mode.

Enabling the GPIO Based Bypass Mode

If user select option 7, GPIO based Bootloader bypass gets enabled. When this option is selected, module checks for the validity of the image selected and displays "Configuration saved" if valid and "Default image invalid" if valid default image is not present. Once enabled, from next bootup, Bootloader will latch the value of UULP_GPIO_2. If asserted, it will bypass the whole boot loading process and will load the default firmware image selected.

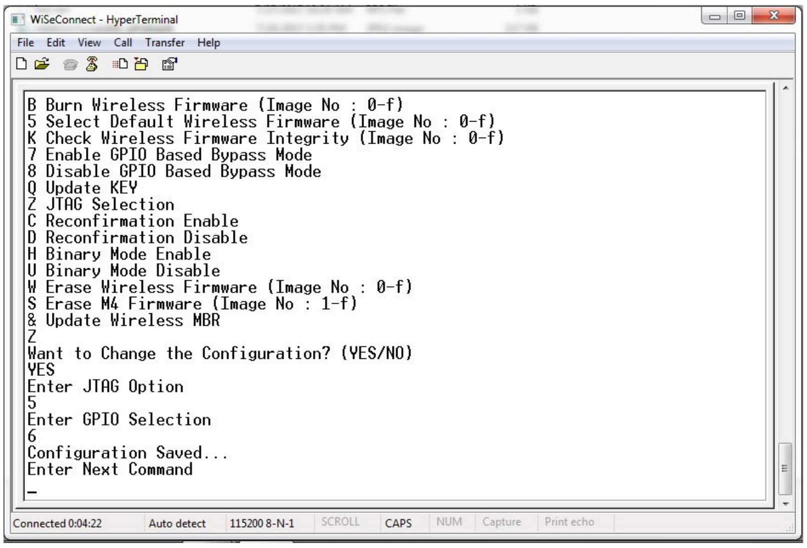
- After the welcome message is displayed, user can select option 5 "Select Default Wireless Firmware (Image No: 0-f)".
- The message "Enter Wireless Image No. (0-f)" is displayed.
- Then select the Image no.
- It is better to check the Integrity of Image before selecting it as Default Image.
- When default image is selected, module checks for the validity of the image selected and displays "Configuration saved".
- Then select option 7 to "Enable GPIO Based Bypass Mode"
- Module responds to select the host interface in Bypass mode (0 - UART, 1 - SDIO, 2 - SPI, 4 - USB, 5 - USB-CDC)
- Select the required interface.

If the default image is valid, then it enables GPIO Bypass mode, otherwise it will not enable the GPIO Bypass mode.


WiSeConnect - HyperTerminal
File Edit View Call Transfer Help
Connected 0:02:24 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo

```
5 Select Default Wireless Firmware (Image No : 0-f)
K Check Wireless Firmware Integrity (Image No : 0-f)
7 Enable GPIO Based Bypass Mode
8 Disable GPIO Based Bypass Mode
Q Update KEY
Z JTAG Selection
C Reconfirmation Enable
D Reconfirmation Disable
H Binary Mode Enable
U Binary Mode Disable
W Erase Wireless Firmware (Image No : 0-f)
S Erase M4 Firmware (Image No : 1-f)
& Update Wireless MBR
Z
Want to Change the Configuration? (YES/NO)
YES
Enter JTAG Option
1
Configuration Saved...
Enter Next Command
1
Loading...
Loading Done
_
```

Figure 17: Enabling the GPIO-based Bypass Mode: Valid Default Firmware


WiSeConnect - HyperTerminal
File Edit View Call Transfer Help
Connected 0:04:22 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo

```
B Burn Wireless Firmware (Image No : 0-f)
5 Select Default Wireless Firmware (Image No : 0-f)
K Check Wireless Firmware Integrity (Image No : 0-f)
7 Enable GPIO Based Bypass Mode
8 Disable GPIO Based Bypass Mode
Q Update KEY
Z JTAG Selection
C Reconfirmation Enable
D Reconfirmation Disable
H Binary Mode Enable
U Binary Mode Disable
W Erase Wireless Firmware (Image No : 0-f)
S Erase M4 Firmware (Image No : 1-f)
& Update Wireless MBR
Z
Want to Change the Configuration? (YES/NO)
YES
Enter JTAG Option
5
Enter GPIO Selection
6
Configuration Saved...
Enter Next Command
_
```

Figure 18: Enabling the GPIO-based Bypass Mode: Invalid Firmware

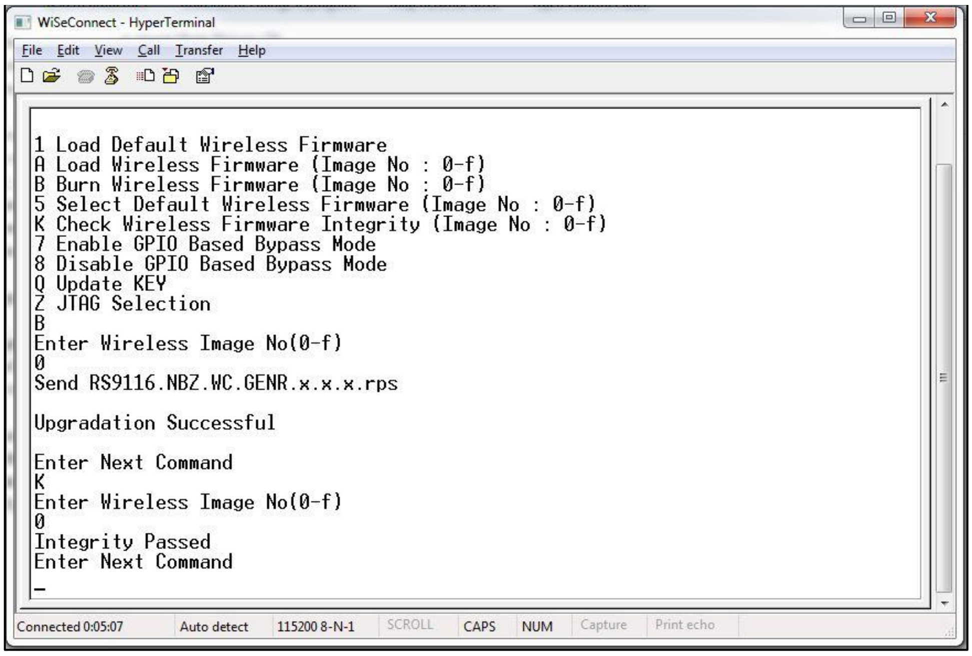
Disabling the GPIO Based Bypass Mode

- If the host selects option 8, GPIO based bypass gets disabled.

UULP_GPIO_2 needs to be de-asserted upon power up to move to host interaction mode, to select bootup options like disable Bypass mode or to change the default image.

Check Integrity of the Selected Image

This option enables the user to check whether the given image is valid or not. When this command is given, bootloader asks for the image for which integrity must be verified as shown in the figure below.


WiSeConnect - HyperTerminal
File Edit View Call Transfer Help
Connected 0:05:07 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo

```
1 Load Default Wireless Firmware
A Load Wireless Firmware (Image No : 0-f)
B Burn Wireless Firmware (Image No : 0-f)
5 Select Default Wireless Firmware (Image No : 0-f)
K Check Wireless Firmware Integrity (Image No : 0-f)
7 Enable GPIO Based Bypass Mode
8 Disable GPIO Based Bypass Mode
Q Update KEY
Z JTAG Selection
B
Enter Wireless Image No(0-f)
0
Send RS9116.NBZ.WC.GENR.x.x.x.rps

Upgradation Successful

Enter Next Command
K
Enter Wireless Image No(0-f)
0
Integrity Passed
Enter Next Command
_
```

Figure 19: Integrity Check Passed

Other Operations

This section contains additional, less frequently used bootloader options.

Update KEY

Note:
This feature is not enabled in the current release.

JTAG Selection

Note:
This feature is not enabled in the current release.

3 Host Interfaces

RS9116 WiSeConnect Module supports SPI, USB, UART and SDIO for interfacing to host. This section describes UART interface in detail including the supported features, protocols and commands.

Only UART and USB-CDC interfaces are supported in AT mode.

Note:

USB and SDIO interfaces are currently not supported.

3.1 UART Interface

This section describes RS9116-WiSeConnect UART interface, including the commands and processes to operate the module via UART.

UART on the RS9116-WiSeConnect is used as a host interface to configure the module to send data and to receive data.

Features

- Supports hardware (RTS/CTS) flow control.
- Supports following list of baud rates,
  - 9600 bps
  - 19200 bps
  - 38400 bps
  - 57600 bps
  - 115200 bps
  - 230400 bps
  - 460800 bps
  - 921600 bps

Note:

For baud rates greater than 115200, it is mandatory to enable UART hardware flow control.

Hardware Interface

RS9116W uses TTL serial UART at an operating voltage of 3.3V.
Host UART device must be configured with the following settings:

- Data bits - 8
- Stop bits - 1
- Parity - None
- Flow control - None

Software Protocol

AT+ command mode

This section explains the procedure that the host needs to follow in order to send Wi-Fi commands frames to the module and to receive responses from the module in AT+ command mode.

TX Operation

The Host uses TX operations:

1. To send management commands to the module from the Host.
2. To send actual data to the module which is to be transmitted onto the air.
3. If the host receives error code indicating packet dropped, the host must wait for a while and send the next command /data.
4. The host should send next data packet only if it receives "OK<number of bytes sent>" response for the previous one.

Rx Operation

The RS9116W responds with either an 'OK' or 'ERROR' string, for Management or Data frames along with a result or error code.

The module sends the response/received data to Host in a format as shown below:

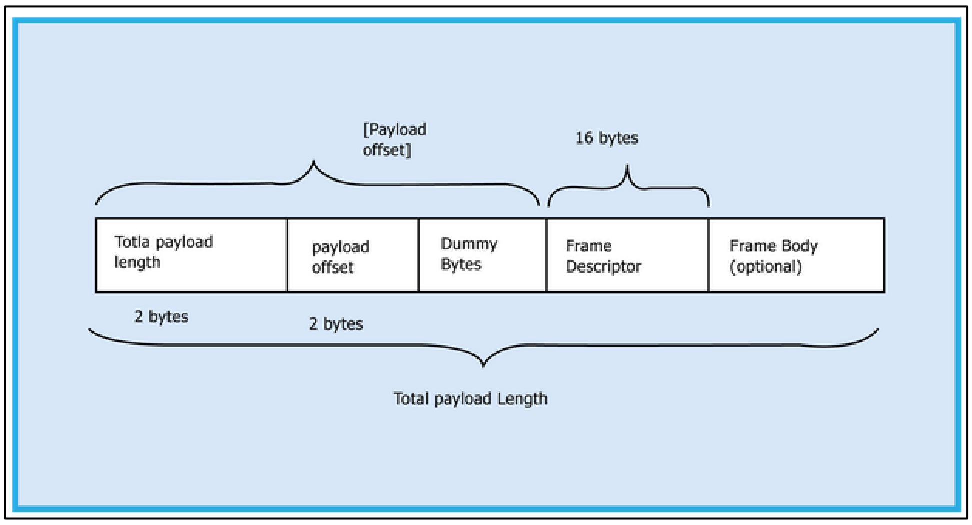


Figure 20: RX Frame Format

Note:
If Payload offset is 'x', 'x-4' dummy bytes will be added before Frame Descriptor.

The host should follow the steps below to read the frame from the Module:

Read 4 bytes using Frame read.

1. Decode Total payload length and payload offset.
2. Read remaining payload by sending Frame to read with (total payload length – 4 bytes), discard dummy bytes and then decode Frame descriptor and Frame Body.

4 AT Command Mode

The Wi-Fi AT command set represents the frames that are sent from the Host to operate the RS9116W-WiSeConnect. The command set resembles the standard AT command interface used for modems.

- All AT commands start with "at" and are terminated with a carriage return('\r') and a new line('\n') character.
- The AT command set for the RS9116W-WiSeConnect starts with "at+rsi_" followed by the name of the command and any relevant parameters.
- In some commands, a '?' character is used after the command to query data from the module.

Appendix C: Sample AT command sequences captures the sample flow of commands to configure the module in various functional modes.

Syntax of AT command:
at+rsi_<command_name>[=][parameters][?]\r\n
Example:
at+rsi_command=< parameter1 >,< parameter2 >,< parameter3 >\r\n
Each parameter should be separated by comma (,).

Note:

1. All commands are issued from Host to module as a sequence of ASCII characters. All return messages from module to Host consist of OK or ERROR strings, along with some return parameters. The return parameters may be ASCII or Hex on a case by case basis. ERROR is accompanied by <Error code>.
2. A command should NOT be issued by the Host before receiving the response of a previously issued command from the module.

RS9116W-WiSeConnect supports following host interfaces in AT Command mode:

- UART
- USB-CDC

The following protocols are supporting in AT Command mode

- WLAN
- BT/BLE

5 WLAN Commands

The following sections explains RS9116-WiSeConnect commands, their parameters, their responses and relevance in AT mode.

5.1 Set Operating Mode

Description:
This is the first command that needs to be sent from host after receiving card ready frame from RS9116W module. This command configures the module in different functional modes.

Command Format:

AT Mode:

```
at+rsi_opermode=
<oper_mode>,<feature_bit_map>,<tcp_ip_feature_bit_map>,<custom_feature_bit_map>,<ext_custom_feature_bit_map>,<
bt_feature_bit_map>,<ext_tcp_ip_feature_bit_map>,<ble_feature_bit_map>,<ble_coustom_ext_feature_bit_map>,<config_feature_bit_map>\r\n
```

Note:

- ext_custom_feature_bit_map gets enabled when BIT(31) is set to ‘1’ in custom_feature_bitmap

at+rsi_opermode=<oper_mode>,<feature_bit_map>,<tcp_ip_feature_bit_map>,<custom_feature_bitmap>,<ext_custom_feature_bit_map>\r\n

- ext_tcp_ip_feature_bit_map gets enabled when BIT(31) is set to ‘1’ in tcp_ip_feature_bit_map

at+rsi_opermode=<oper_mode>,<feature_bit_map>,<tcp_ip_feature_bit_map>,<custom_feature_bitmap>,<ext_tcp_ip_feature_bit_map>\r\n

- bt_feature_bit_map gets enabled when BIT(31) is set to ‘1’ in both custom_feature_bit_map and ext_custom_feature_bit_map

at+rsi_opermode=<oper_mode>,<feature_bit_map>,<tcp_ip_feature_bit_map>,<custom_feature_bitmap>,<ext_custom_feature_bit_map>,<bt_feature_bit_map>\r\n

- ble_feature_bit_map gets enabled when BIT(31) is set to 1 in custom_feature_bitmap, ext_custom_feature_bit_map and bt_feature_bit_map

at+rsi_opermode=<oper_mode>,<feature_bit_map>,<tcp_ip_feature_bit_map>,<custom_feature_bitmap>,<ext_custom_feature_bit_map>,<bt_feature_bit_map>,<ext_tcp_ip_feature_bit_map>,<ble_feature_bit_map>\r\n

- ble_custom_ext_feature_bit_map gets enable when BIT(31) is set to 1 in ble_custom_feature_bit_map

at+rsi_opermode=<oper_mode>,<feature_bit_map>,<tcp_ip_feature_bit_map>,<custom_feature_bitmap>,<ext_custom_feature_bit_map>,<bt_feature_bit_map>,<ext_tcp_ip_feature_bit_map>,<ble_feature_bit_map>,<ble_coustom_ext_feature_bit_map>\r\n

- config_feature_bit_map gets enabled when BIT(31) is set to '1' in both tcp_ip_feature_bit_map and ext_tcp_ip_feature_bit_map

at+rsi_opermode=<oper_mode>,<feature_bit_map>,<tcp_ip_feature_bit_map>,<custom_feature_bitmap>,<ext_custom_feature_bit_map>,<bt_feature_bit_map>,<ext_tcp_ip_feature_bit_map>,<ble_feature_bit_map>,<ble_coustom_ext_feature_bit_map>,<config_feature_bit_map>\r\n

Oper_mode:

Sets the mode of operation. oper_mode contains two parts <wifi_oper_mode, coex_mode>. Lower two bytes represents wifi_oper_mode and higher two bytes represents coex_modes.

```
oper_mode = ((wifi_oper_mode) | (coex_mode << 16))
```

Wi-Fi_oper_mode values:

- 0 - Wi-Fi Client Mode.
  The module works as a normal client that can connect to an Access Point with different security modes other than enterprise security.
- 2 – Enterprise Security Client Mode.
  The module works as a client that can connect to an Access Point with WPA/WPA2-Enterprise security.
- 6 – Access Point mode**.**
  In this mode, the module acts as an Access Point, depending on the inputs supplied for the command "Configure AP Mode". In Access Point mode, a Maximum of 16 clients can connect based on the bits set in custom feature bit map selection in opermode.
- 8 - PER Mode.
  This mode is used for calculating packet error rate and mostly used during RF certification tests.
- 9 – Concurrent mode.
  This mode is used to run module in concurrent mode. In concurrent mode, host can connect to an AP and can create AP simultaneously.

Note:

In concurrent mode,

1. AP MAC address's last byte will differ, and it will be one plus the station mode MAC last byte.
2. Maximum of 4 Station devices can be connected to Concurrent AP.
3. In TCP/IP non bypass mode, Broadcast / Multicast packet goes to first created interface (e.g. if Station mode connects first, the broadcast / multicast packet goes to the network that belongs to station mode).
4. IPV6 support is not supported in the current release.
5. Aggregation is not supported.

Possible coex modes:

Coex_mode	Description
1	WLAN
2	ZigBee*
3	WLAN + ZigBee*
4	Bluetooth
5	WLAN + Bluetooth
6	Bluetooth + Zigbee co-existence*
7	WLAN + Bluetooth + ZigBee co-existence*
8	Dual Mode (Bluetooth and BLE)
9	WLAN + Dual Mode
10	Dual Mode + ZigBee co-existence*
11	WLAN + Dual Mode + ZigBee co-existence*
12	BLE mode
13	WLAN + BLE
14	BLE and ZigBee co-existence*
15	WLAN + BLE + ZigBee co-existence*

Note:

1. Coex modes are **supported only in 384K memory** configuration
2. For WLAN Coex mode, AP mode scan is not allowed.
3. ZigBee is not currently supported. Will be supported in future releases.

If coex mode is enabled in opermode command, then BT / BLE protocol will start and give corresponding card ready in parallel with opermode command response (which will be handled by corresponding application). BT card ready frame is described in **RS9116W BT Classic AT Command Programming Reference Manual.pdf** (available at https://docs.silabs.com/rs9116) and BLE card ready frame is described in **RS9116W BLE AT Command Programming Reference Manual.pdf** (available at https://docs.silabs.com/rs9116).

To select coex mode, please refer to **RS9116_TCP_IP_Feature_Selection_vx.x.x.xlsx** given in "Docs" folder of the release package.

feature_bit_map: This bitmap is used to enable following WLAN features:

feature_bit_map	**Functionality**	**Bit set to 0**	**Bit set to 1**	**Note and Info**
feature_bit_map[0]	Open mode feature	Disable	Enable	Supported in client mode.
feature_bit_map[1]	PSK security	Disable	Enable	Supported in client mode.
feature_bit_map[2]	Aggregation	Disable	Enable	
feature_bit_map[3]	LP GPIO handshake	Disable	Enable	
feature_bit_map[4]	ULP GPIO handshake	Disable	Enable	
feature_bit_map[5]	Reserved	Reserved	Reserved	
feature_bit_map[6]	Reserved	Reserved	Reserved	
feature_bit_map[7]	WPS support	Enable	Disable	
feature_bit_map[8]	To support EAP-LEAP in WiFi + BT Coex mode	Disable	Enable	
feature_bit_map[9:31]	Reserved. Should set to be '0'			

Note:

feature_bit_map[0], feature_bit_map[1] are valid only in Wi-Fi Client mode.

tcp_ip_feature_bit_map: To enable TCP/IP related features.

tcp_ip_feature_bit_map	**Functionality**	**Bit set to 0**	**Bit set to 1**	**Note and Info**
tcp_ip_feature_bit_map[0]	TCP/IP bypass	Disable	Enable	
tcp_ip_feature_bit_map[1]	HTTP server	Disable	Enable	
tcp_ip_feature_bit_map[2]	DHCPv4 client	Disable	Enable	
tcp_ip_feature_bit_map[3]	DHCPv6 client	Disable	Enable	

tcp_ip_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
tcp_ip_feature_bit_map[4]	DHCPv4 server	Disable	Enable	
tcp_ip_feature_bit_map[5]	DHCPv6 server	Disable	Enable	
tcp_ip_feature_bit_map[6]	Dynamic update of web pages (JSON objects)	Disable	Enable	
tcp_ip_feature_bit_map[7]	HTTP client	Disable	Enable	
tcp_ip_feature_bit_map[8]	DNS client	Disable	Enable	
tcp_ip_feature_bit_map[9]	SNMP agent	Disable	Enable	
tcp_ip_feature_bit_map[10]	SSL	Disable	Enable	
tcp_ip_feature_bit_map[11]	PING from module (ICMP)	Disable	Enable	
tcp_ip_feature_bit_map[12]	HTTPS Server	Disable	Enable	Only supported in opermode 0.
tcp_ip_feature_bit_map[13]	Set to 0.			
tcp_ip_feature_bit_map[14]	To send configuration details to host on submitting configurations on wireless configuration page	Disable	Enable	
tcp_ip_feature_bit_map[15]	FTP client	Disable	Enable	
tcp_ip_feature_bit_map[16]	SNTP client	Disable	Enable	
tcp_ip_feature_bit_map[17]	IPv6 mode	Disable	Enable	IPv6 will also get enabled if DHCP v6 client/DHCP v6 server is enabled irrespective of tcp_ip_feature_bit_map[17].
tcp_ip_feature_bit_map[18]	RAW Socket feature	Disable	Enable	This feature is supported only in AP mode. TCP_BYPASS feature should be disabled for this feature to be supported. If any packet from host with frame type 0x1 is received by firmware, the packet will be sent on air without TCP/IP stack processing. ARP and broadcast packets (other than DHCP packets) which are coming on air will be sent to host.
tcp_ip_feature_bit_map[19]	To MDNS and DNS-SD	Disable	Enable	
tcp_ip_feature_bit_map[20]	SMTP client	Disable	Enable	
tcp_ip_feature_bit_map[21:24]	To select no of sockets			
tcp_ip_feature_bit_map[25]	Single SSL socket	Disable	Enable	
tcp_ip_feature_bit_map[26]	Private & Public certificate	Disable	Enable	If Secure handshake is with CA – certificate alone, then disable loading private and public keys and erase these certificates from the flash using load_cert API. Or if Secure handshake is needed for verification of Private and Public keys, then enable loading of private and public keys.

tcp_ip_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
tcp_ip_feature_bit_map[27]	SSL certificate on to the RAM	Disable	Enable	
tcp_ip_feature_bit_map[28]	TCP-IP data packet Dump on UART2	Disable	Enable	
tcp_ip_feature_bit_map[29]	POP3 client	Disable	Enable	
tcp_ip_feature_bit_map[30]	To enable OTAF (On the Air Firmware) up gradation.	Disable	Enable	
tcp_ip_feature_bit_map[31]	tcp_ip_ext_feature_bitmap validity	Disable	Enable	

Note:
Feature selection utility is provided in the release package. WiSeConnect device supports the selected features combination only if it is feasible as per **WiSeConnect_TCPIP_Feature_Selection_vx.x.x.xlsx**.

custom_feature_bit_map: This bitmap is used to enable following custom features

custom_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
custom_feature_bit_map[2]	If this bit is set to '1', the DHCP server behavior, when the module is in AP mode changes. The DHCP server, when it assigns IP addresses to the client nodes, does not send out a Gateway address, and sends only the assigned IP and Subnet values to the client. It is highly recommended to keep this value at '0' as the changed behavior is required in only very specialized use cases and not in normal AP functionality. The default value of this bit is '0'.			
custom_feature_bit_map[4]	To run NWP (Network Processor) at Higher clock frequency(160MHZ)	Disable	Enable	Need to set pll_mode to 1 in feature frame command.
custom_feature_bit_map[5]	Hidden SSID in AP mode	Disable	Enable	
custom_feature_bit_map[6]	DNS server IP address in DHCP offer response in AP mode.	Disable	Enable	
custom_feature_bit_map[8]	DFS channel passive scan support	Disable	Enable	it's mandatory to set region before scanning DFS channel.
custom_feature_bit_map[10]	Asynchronous messages to host to indicate the module state.	Disable	Enable	
custom_feature_bit_map[11]	Packet pending wake on wireless indication in UART mode	Disable	Enable	
custom_feature_bit_map[12]	Bypass AP Blacklist in client mode	Disable	Enable	By default, client maintains AP blacklist internally to avoid some access points. To bypass AP blacklist feature in client mode during roaming or rejoin, this

custom_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
				feature should be enabled.
custom_feature_bit_map[17]	To select between de-authentication or null data (with power management bit set) based roaming. Depending on selected method station, it will send de-auth or Null data to the connected AP when roaming from connected AP to the newly selected AP.	Disable	Enable	
custom_feature_bit_map[18]	Reserved			
custom_feature_bit_map[19]	Reserved			
custom_feature_bit_map[20]	Trigger Auto Configuration	Disable	Enable	
custom_feature_bit_map[22]	Used to enable per station power save packet buffer limit in AP mode. When enabled, only two packets per station will be buffered when station is in power save	Disable	Enable	
custom_feature_bit_map[23]	HTTP/HTTPs authentication	Disable	Enable	
custom_feature_bit_map[24]	To run NWP at Higher clock frequency(120MHZ)	Disable	Enable	Need to set pll_mode to 1 in feature frame command
custom_feature_bit_map[25]	HTTP server credentials to host in get configuration command	Disable	Enable	
custom_feature_bit_map[26]	To accept or reject new connection request when maximum clients are connected in case of LTCP.	Disable	Enable	By default, this bit value is zero. **When BIT[26] is zero:** For a LTCP socket when maximum clients are connected if a new connection request is received, then this connection request will not be rejected. Instead module will maintain this connection request in LTCP pending list. This request will be served when any of the connected client is disconnected. **When BIT[26] is set:** For a LTCP socket when maximum clients are connected if a new connection request is received, then this connection request will be rejected immediately. Module will not maintain this connection request in LTCP pending list.
custom_feature_bit_map[27]	Dual band roaming and vcsafd feature	Disable	Enable	

custom_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
custom_feature_bit_map[28]	Real time clock from host	Disable	Enable	
custom_feature_bit_map[29]	IAP support in BT mode	Disable	Enable	
custom_feature_bit_map[31]	ext_custom_feat_bitmap validity	Disable	Enable	

BIT[0:1], BIT[3:4], BIT[7], BIT[21], BIT[30]: Reserved, should be set to all '0'.

Note:

For UART / USB-CDC in AT mode:

When user does not give any tcp_ip_feature_bit_map value then default settings for client mode, Enterprise client mode.
HTTP server, DHCPv4 client, DHCPv6 client and JSON objects are enabled.

When user does not give any tcp_ip_feature_bit_map value then default settings for Access point mode are: HTTP server, DHCPv4 server, DHCPv6 server and JSON objects are enabled.

Parameters- feature_bit_map, tcp_ip_feature_bit_map and custom_feature_bit_map are optional in opermode command in UART mode for AT mode.
If user does not give these parameters then default configuration gets selected, as explained above, based upon the operating mode configured.

If opermode is 8 (PER mode is selected) - feature_bit_map, tcp_ip_feature_bit_map and custom_feature_bit_map can be ignored or not valid. Set to zero.

ext_custom_feature_bit_map:

This feature bitmap is an extension of custom feature bitmap and is valid only if BIT[31] of custom feature bitmap is set. This enables the following feature.

ext_custom_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
ext_custom_feature_bit_map[0]	Antenna diversity feature.	Disable	Enable	
ext_custom_feature_bit_map[1]	4096-bit RSA key support	Disable	Enable	This bit is required to set for 4096-bit RSA key support. If key size is 4096-bit, module will use software routine for exponentiation, so connection time will increase.
ext_custom_feature_bit_map[3]	SSL certificate with 4096-bit key support	Disable	Enable	
ext_custom_feature_bit_map[4]	This bit is applicable only in AP and concurrent AP mode. If this bit is set, the module will send broadcast data (if any) immediately without waiting for DTIM.	Disable (AP sends broadcast packet at DTIM only.)	Enable (Enable sending broadcast without waiting for DTIM)	If this bit is enabled, then connected client who is in power save may miss the packet.

ext_custom_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
ext_custom_feature_bit_map[5]	Pre authentication Support.	Disable	Enable	
ext_custom_feature_bit_map[6]	40MHZ Support	Disable	Enable	11J AP don't support 40MHZ **Note**: 40MHz is not supported.
ext_custom_feature_bit_map[9]	EXT_HTTP_SKIP_DEFAULT_LEADING_CHARACTER To skip default leading character ("\") in resource name			
ext_custom_feature_bit_map[11]	802.11R Over the Air Roaming Support	Disable	Enable	1.Resource Request Support is not Present. 2 If both BIT[11] and BIT[16] are not enabled then it will select as Legacy Roaming.
ext_custom_feature_bit_map[12]	802.11J support.	Disable	Enable	If this bit is enabled, set region command is mandatory with setting it to Japan region and band value must be 1 (5GHz).
ext_custom_feature_bit_map[13]	802.11W support.	Disable	Enable	
ext_custom_feature_bit_map[14]	TLS Multiple versions Support in SSL.	Disable	Enable	
ext_custom_feature_bit_map[15]	Support 16 Stations in AP Mode.			If this bit is enabled, then 16 stations can connect in AP mode otherwise Maximum of 8 stations can connect.
ext_custom_feature_bit_map[16]	802.11R Over the Distributed System Roaming Support			1. Resource Request Support is not Present. 2. If both BIT[11] and BIT[16] are not enabled then it will select as Legacy Roaming.
ext_custom_feature_bit_map[19]	Low power mode in GPIO based or M4 based ULP power save			
ext_custom_feature_bit_map[20:21]	Default memory configuration (RAM) is 192KB. User can set these bits to change the memory configuration as below: **MODE(KB)** | **BIT[20]** | **BIT[21]** 192 | 0 | 0			

ext_custom_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
	256 | 0 | 1 320 | 1 | 0 384 | 1 | 1			
ext_custom_feature_bit_map[22:23]	To configure Sleep clock source selection To select sleep clock for TA (Thread Arc processor), either crystal clock or RC clock **SELECTION** | **BIT[23]** | **BIT[22]** Use RC clock as sleep clock | 0 | 0 Use 32KHz clock from external XTAL OSCILLATOR | 0 | 1 Use 32KHz bypass clock on UULP_GPIO_3 | 1 | 0 Use 32KHz bypass clock on UULP_GPIO_4 | 1 | 1			If BIT[23] is set '1' and BIT[22] is set to '0' in ext_custom_feature_bit_map, then user has to use UULP_GPIO_0 for sleep indication to host.
ext_custom_feature_bit_map[24]	Reserved			
ext_custom_feature_bit_map[25]	To enable 1.8v supply for TA	Disable	Enable	
ext_custom_feature_bit_map[26]	To enable 3.3v supply for TA	Disable	Enable	
ext_custom_feature_bit_map[27]	To select UART port for NWP (Network Processor) debug prints	Disable (Enable debug prints on UART 2)	Enable (Enable debug prints on UART 1 and is applicable only if Host Interface is not UART)	By default, all the debug prints from NWP will be coming on UART2 if this bit is not enabled. UART 1 pins are mapped to the following pins w.r.t to NWP. User needs to ensure that these pins are not used in MCU applications in WiSeMCU mode to avoid conflicts of pins usage based on the requirement. This bit is valid only if BIT[28] in ext_custom_feature_bit_map is set to 0. UART 1 - TX: GPIO_9 RX: GPIO_8 UART 2 - TX: GPIO_6 RX: GPIO_10 There is no functionlaity on rx pins for debug prints.
ext_custom_feature_bit_map[28]	UART debug prints from NWP	Disable (Enable debug prints)	Enable (Disable debug	By default, this bit value is zero. When BIT[26] is zero: For a LTCP socket when

ext_custom_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
			prints on either UART1 or UART2)	maximum clients are connected if a new connection request is received, then this connection request will not be rejected. Instead module will maintain this connection request in LTCP pending list. This request will be served when any of the connected client is disconnected. When BIT[26] is set: For a LTCP socket when maximum clients are connected if a new connection request is received, then this connection request will be rejected immediately. Module will not maintain this connection request in LTCP pending list.
ext_custom_feature_bit_map[29:30]	Reserved			
ext_custom_feature_bit_map[31]	BT and BLE feature_bit_map validity	Disable	Enable	

ext_tcp_ip_feature_bit_map

ext_tcp_ip_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
ext_tcp_ip_feature_bit_map[1]	DHCP USER CLASS	Disable	Enable	
ext_tcp_ip_feature_bit_map[2]	HTTP server root path bypass	Disable	Enable	
ext_tcp_ip_feature_bit_map[3]	Correcting the ACK sequence number in the TCP packet retransmission path	Disable	Enable	Need to enable this bit if user wants to run the bi-directional data transfer.
ext_tcp_ip_feature_bit_map[4]	To enable TCP ACK division factor feature	Disable	Enable	
ext_tcp_ip_feature_bit_map[5]	To enable SSL server certificate validation by host.	Disable	Enable	
ext_tcp_ip_feature_bit_map[6]	Support for SSL 16K record size	Disable	Enable	
ext_tcp_ip_feature_bit_map[7]	To enable DNS_CLIENT_BYPASS by host	Disable	Enable	

ext_tcp_ip_feature_bit_map	**Functionality**	**Bit set to 0**	**Bit set to 1**	**Note and Info**
ext_tcp_ip_feature_bit_map[8]	To enable TCP window scaling feature	Disable	Enable	If user wants to use more than 64Kb window size. **tcp_rx_window_size_cap** in socket config command is used to increase the window size.
ext_tcp_ip_feature_bit_map[9]	To enable Dual Mode feature	Disable	Enable	Enabling this feature allows to use both bypass and non-bypass modes simultaneously.
ext_tcp_ip_feature_bit_map[10]	To enable Ethernet Wi-Fi bridge feature	Disable	Enable	
ext_tcp_ip_feature_bit_map[11]	To enable the dynamic coex memory	Disable	Enable	To enable or disable the coex and update TCP RX window accordingly.
ext_tcp_ip_feature_bit_map[12:15]	To select the no of selects	Disable	Enable	
ext_tcp_ip_feature_bit_map[16]	To enable socket wait close	Disable	Enable	If it is set socket will not be closed until the shutdown is called from host. **Note:** Refer to 'rsi_shutdown()' API in RS9116W SAPI Programming Reference Manual at https://docs.silabs.com/rs9116/ **Note:** It is recommended to enable this bit, when using TCP sockets.
ext_tcp_ip_feature_bit_map[17]	To enable MQTT feature	Disable	Enable	If user wants to use AT command for MQTT, should be enable this bit in the Opermode Command
ext_tcp_ip_feature_bit_map[18]	To enable HTTP OTAF Feature	Disable	Enable	To do firmware upgrade with http this bit should be enabled
ext_tcp_ip_feature_bit_map[19:29]	Reserved			
ext_tcp_ip_feature_bit_map[30]	To configure additional memory for SSL connection typically to a cloud server	Disable	Enable	If user connects to a cloud server using two SSL connections, then it is required to set this bit to avoid 0xD2 error
ext_tcp_ip_feature_bit_map[31]	config_feature_bit_map validity	Disable	Enable	

Note:

1.Set BIT(31) in tcp_ip_feature_bit_map & BIT(29) in ext_tcp_ip_feature_bit_map to open 3 SSL sockets.

2. Three SSL Feature is supported only in Opermode WLAN Only Mode.

bt_feature_bit_map

This bitmap is valid only if BIT[31] of extended custom feature bit map is set.

bt_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
bt_feature_bit_map[0-14]	Reserved			
bt_feature_bit_map[15]	HFP profile bit enable	Disable	Disable	
bt_feature_bit_map[16:19]	Reserved for future use			
bt_feature_bit_map[20:22]	number of slaves supported by BT			Maximum no of BT slaves: 1
bt_feature_bit_map[23]	A2DP profile bit enable	Disable	Enable	
bt_feature_bit_map[24]	A2DP profile role selection	Disable	Enable	
bt_feature_bit_map[25]	A2DP accelerated mode selection	Disable	Enable	
bt_feature_bit_map[26]	A2DP i2s mode selection	Disable	Enable	
bt_feature_bit_map[27:29]	Reserved			
bt_feature_bit_map[30]	RF Type selection	Disable	Enable	
bt_feature_bit_map[31]	Validate ble feature bit map	Disable	Enable	

ble_feature_bit_map

This bitmap is valid only if BIT[31] of bt custom feature bit map is set.

ble_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
ble_feature_bit_map[0:7]	BLE nbr of attributes,			Maximum No of Ble attributes = 80, Please refer **NOTE** given below for more info.
ble_feature_bit_map[8:11]	BLE Nbr of GATT services			maximum no services - 10 Please refer **NOTE** given below for more info.
ble_feature_bit_map[12:15]	BLE Nbr of slaves			Maximum No of Ble slaves = 8 Please refer **NOTE** given below for more info.
ble_feature_bit_map[16:23]	BLE tx power index			Give 31 as ble tx power index (e.g. 31<<16). This variable is used to select the ble tx power index value. The following are the possible values. Default Value for BLE Tx Power Index is 31 Range for the BLE Tx Power Index is 1 to 75 (0, 32 indices are invalid) 1 - 31 BLE -0DBM Mode 33 - 63 BLE- 10DBM Mode 64- 75 BLE - HP Mode.
ble_feature_bit_map[24:26]	BLE powersave options BLE_DUTY_CYCLING BIT(24) BLR_DUTY_CYCLING BIT(25)			

ble_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
	BLE_4X_PWR_SAVE_MODE BIT(26)			
ble_feature_bit_map[27:29]	Reserved			
ble_feature_bit_map[30]	RS9113 - RS9116 compatible features	Disable	Enable	
ble_feature_bit_map[31]	Reserved			

ble_custom_ext_feature_bit_map:

This bitmap is valid only if BIT[31] of ble custom feature bit map is set.

ble_coustom_ext_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
ble_coustom_ext_feature_bit_map[0:4]	BLE nbr of Connection Events			
ble_coustom_ext_feature_bit_map[5:12]	BLE nbr of record size in bytes (n)			E.g.: n*16:(n=60, default 1024 bytes(1K))
ble_coustom_ext_feature_bit_map[13]	GATT INIT			0 - GATTt Init in Firmware 1 - GATT Init in Host

Note:

If bit **bt_custom_feature_bit_map[31]** is set:

1. User can enter maximum of 8 BLE slaves.
2. Maximum of 10 services in total can exist out of which two services namely GAP and GATT are added by default. So, if this bitmap has value 10 user can add up to 8 services.
3. Maximum of 80 attributes in total can exist out of which ten attributes of GAP and GATT are added by default. So, if this bitmap has value 80 user can add up to 70 attributes.

If bit **bt_custom_feature_bit_map** is not set:

1. Default number of BLE slaves supported is 3.
2. Maximum of 5 services in total can exist out of which two services namely GAP and GATT are added by default. So, user can add up to 3 services.
3. Maximum of 20 attributes in total can exist out of which ten attributes of GAP and GATT are added by default. So, user can add up to 10 attributes.

config_feature_bit_map:

This bitmap is valid only if BIT[31] of ext_tcp_ip_feature_bit_map is set.

config_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
config_feature_bit_map[0]	To select wakeup indication to host.	Disable	Enable	

config_feature_bit_map	Functionality							Bit set to 0	Bit set to 1	Note and Info
	If it is disabled UULP_GPIO_3 is used as a wakeup indication to host. If it is enabled UULP_GPIO_0 is used as a wakeup indication to host.									
config_feature_bit_map[5:2]	**BIT(5)**	**BIT(4)**	**BIT(3)**	**BIT(2)**	**BT Voltage**	**Wi-Fi Voltage**	**Description**			These bits are used for dynamic voltage selection
	0	0	0	0			Default configuration. Voltages are based on BIT(25) in ext_custom_feature_bit_map, if this bit is set to 1 means 1.8v selected else 3.3v selected			
	0	0	0	1	3.1 V	3.1V	Fixed voltage operation 3.1V for BT and Wi-Fi.			
	0	0	1	0	1.85 V	1.85V	Fixed voltage operation 1.85V for BT and Wi-Fi.			
	0	0	1	1	1.85 V	3.1V	Protocol based PA Voltage switching between BT and Wi-Fi – Option1			
	0	1	0	0	1.55 V	3.1V	Protocol based PA Voltage switching between BT and Wi-Fi – Option2			
	0	1	0	1	1.55 V/1.85V	1.85V /3.1V	NOTE: It's not supported Yet. Adaptive PA voltage switching based on Protocol and Wireless channel conditions BT 1.85V when far from headset – otherwise 1.55V Wi-Fi 1.85V when near AP – otherwise 3.1V			

config_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
config_feature_bit_map[9:6]	BIT(6) / BIT(7) / BIT(8) / BIT(9) / Description: 0 0 0 0 — 0 means INTERNAL_PMU 0 0 0 1 — EXTERNAL_PMU_GOOD_TIME_100USEC 0 0 1 0 — EXTERNAL_PMU_GOOD_TIME_200USEC 0 0 1 1 — EXTERNAL_PMU_GOOD_TIME_300USEC 0 1 0 0 — EXTERNAL_PMU_GOOD_TIME_400USEC 0 1 0 1 — EXTERNAL_PMU_GOOD_TIME_500USEC 0 1 1 0 — EXTERNAL_PMU_GOOD_TIME_600USEC 0 1 1 1 — EXTERNAL_PMU_GOOD_TIME_700USEC 1 0 0 0 — EXTERNAL_PMU_GOOD_TIME_800USEC 1 0 0 1 — EXTERNAL_PMU_GOOD_TIME_900USEC 1 0 1 0 — EXTERNAL_PMU_GOOD_TIME_1000USEC 1 0 1 1 — EXTERNAL_PMU_GOOD_TIME_1100USEC 1 1 0 0 — EXTERNAL_PMU_GOOD_TIME_1200USEC 1 1 0 1 — EXTERNAL_PMU_GOOD_TIME_1300USEC 1 1 1 0 — EXTERNAL_PMU_GOOD_TIME_1400USEC 1 1 1 1 — EXTERNAL_PMU_GOOD_TIME_1500USEC			These bits are used to select external PMU good time. 1 to 15 means 100usec to 1500usec (in100usec granularity)
config_feature_bit_map[11:10]	BIT(11) / BIT(10) / Description: 0 1 — if this set External LDO Enable 1 1 — If this set External LDO is 1.0v, Else External LDO is 1.1v (This field valid only if External LDO is enabled i.e. 10bit is set)			These bits are used for External LDO selection **External PMU:** 1.In case of External PMU, User has to set EXTERNAL_PMU_GOOD_TIME_CONFIGURATION value to external PMU good time, if

Sub-table for config_feature_bit_map[9:6]:

BIT(6)	BIT(7)	BIT(8)	BIT(9)	Description
0	0	0	0	0 means INTERNAL_PMU
0	0	0	1	EXTERNAL_PMU_GOOD_TIME_100USEC
0	0	1	0	EXTERNAL_PMU_GOOD_TIME_200USEC
0	0	1	1	EXTERNAL_PMU_GOOD_TIME_300USEC
0	1	0	0	EXTERNAL_PMU_GOOD_TIME_400USEC
0	1	0	1	EXTERNAL_PMU_GOOD_TIME_500USEC
0	1	1	0	EXTERNAL_PMU_GOOD_TIME_600USEC
0	1	1	1	EXTERNAL_PMU_GOOD_TIME_700USEC
1	0	0	0	EXTERNAL_PMU_GOOD_TIME_800USEC
1	0	0	1	EXTERNAL_PMU_GOOD_TIME_900USEC
1	0	1	0	EXTERNAL_PMU_GOOD_TIME_1000USEC
1	0	1	1	EXTERNAL_PMU_GOOD_TIME_1100USEC
1	1	0	0	EXTERNAL_PMU_GOOD_TIME_1200USEC
1	1	0	1	EXTERNAL_PMU_GOOD_TIME_1300USEC
1	1	1	0	EXTERNAL_PMU_GOOD_TIME_1400USEC
1	1	1	1	EXTERNAL_PMU_GOOD_TIME_1500USEC

Sub-table for config_feature_bit_map[11:10]:

BIT(11)	BIT(10)	Description
0	1	if this set External LDO Enable
1	1	If this set External LDO is 1.0v, Else External LDO is 1.1v (This field valid only if External LDO is enabled i.e. 10bit is set)

config_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
				this is zero then it indicates using Internal PMU. 2. In case of External PMU 1.0v or 1.05v, User has to set both the bits config_feature_bit_map[11] & config_feature_bit_map[10].
config_feature_bit_map[13:12]	<table><tr><th>BIT(13)</th><th>BIT (12)</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>UULP_GPIO_0 as sleep indication to host.</td></tr><tr><td>0</td><td>1</td><td>ULP_GPIO_6 as sleep indication to host.</td></tr><tr><td>1</td><td>1</td><td>ULP_GPIO_5 as sleep indication to host</td></tr></table>			If these bits are not set, then by default UULP_GPIO_0 will be used.
config_feature_bit_map[14]	To select TLS 1.0 version	Disable	Enable	
config_feature_bit_map[15]	To select TLS 1.2 version	Disable	Enable	
config_feature_bit_map[16]	Active high or low interrupt mode selection for wake on wireless operation If it is disabled active low interrupt is used in wake on wireless operation. If it is enabled active high interrupt is used in wake on wireless operation.	Disable	Enable	
config_feature_bit_map[23:17]	Reserved			
config_feature_bit_map[24:25]	Configurability options for 40MHz XTAL good time in µs<table><tr><th>BIT(25)</th><th>BIT(24)</th><th>Good time</th></tr><tr><td>0</td><td>0</td><td>1000</td></tr><tr><td>0</td><td>1</td><td>2000</td></tr><tr><td>1</td><td>0</td><td>3000</td></tr><tr><td>1</td><td>1</td><td>600</td></tr></table>			These bits are used to select XTAL good time. These changes are available from Release 2.3.0 onwards. Releases prior to 2.3.0 these config_feature_bit map[31:17] are reserved. Its only applicable for customers using chip and not the module. Please contact support for more details. Default value is 1000 µs.

config_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
config_feature_bit_map[31:26]	Reserved for LMAC			

Note:

32KHz external clock connection and power save pins

As per Silicon lab's datasheet update in May 2019, the 32KHz external clock and the power save pins connections have changed. To keep SW compatibility between initial design (i.e. first EVKs developed by Silicon Labs) as well as new designs, there are currently 2 options for connecting the 32KHz external clock and the power save pins:

Option 1:

External 32KHz clock connection pins	: XTAL_32KHZ_P & XTAL_32KHZ_N
Power Save connection pins	: HOST_BYP_ULP_WAKEUP & UULP_VBAT_GPIO_3

Option 2:

External 32KHz clock connection pin	: UULP_VBAT_GPIO_3
Power Save connection pins	: HOST_BYP_ULP_WAKEUP & UULP_VBAT_GPIO_0

As per Silicon Labs datasheet updated in May'2019, Option 2 must be used for External 32KHz external clock and Power save connections in new designs.

Response:

AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0021,0x0025,0xFF73,0x002C,0xFF6E,0xFF6F,0xFF70,0xFFC5.

Example:

AT Mode:
When only oper_mode is given in command:

at+rsi_opermode=1\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6F 0x70 0x65 0x72 0x6D 0x6F 0x64 0x65 0x3D 0x31 0x0D 0x0A

Response:

OK\r\n
0x4F 0x4B 0x0D 0x0A

When other parameters along with mode_val is given in opermode command:
at+rsi_opermode=1,1,2,0\r\n

0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6F 0x70 0x65 0x72 0x6D 0x6F 0x64 0x65 0x3D 0x31 0x2C 0x31 0x2C 0x32 0x2C 0x30 0x0D 0x0A
By giving above command module is configured in WFD mode with HTTP server enabled in open mode.

Note:

Enabling Aggregation bit (feature_bit_map[2]) and Low power mode bit (ext_custom_feature_bit_map[19]) in Opermode will result in Wi-Fi data not working. So, they cannot be enabled at the same time.

5.2 Band

Description:
This command configures the band in which the module must be configured. This command must be issued after opermode command.

Command Format:

AT Mode:
at+rsi_band=< bandVal >\r\n

Command Parameters:

The valid values for the parameter for this command are as follows:
bandVal(1 byte):

Mode	Functionality
0	2.4 GHz
1	5 GHz
2	Dual band (2.4 Ghz and 5 Ghz)

Note:

1. Dual band is supported in station mode.

Note:

802.11J support only 5 GHz bandVal

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0005, 0x0021, 0x0025, 0x002C, 0x003c.

Relevance:
This command is relevant in all operating modes.

Example:
AT Mode:
at+rsi_band=0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x62 0x61 0x6E 0x64 0x3D 0x30 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.3 WLAN Config

Description:
This command configures WLAN parameters.

Command Format:

AT Mode:
at+rsi_config=< config_type >,< config_val >\r\n

Command Parameters:

The valid values for the parameter for this command are as follows:

The valid values for the parameter for this command are as follows:
Config_type[2 bytes]: 1 - To configure RTS threshold value
Config_value[2 bytes]: [256-2346] - Range of rts threshold value

Note:

Only RTS_THRESHOLD is supported.

Note:

rsi_config command must be given after rsi_init command.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0063, 0x0064.

Relevance:
This command is relevant in all operating modes.

Example:
AT Mode:
at+rsi_config=1,256\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x63 0x6F 0x6E 0x66 0x69
0x67 0x3D 0x31 0x2C 0x32 0x35 0x36 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.4 Set MAC Address

Description:
This command sets the module's MAC address. This command has to be issued before band command.

Command:
AT Mode:
at+rsi_setmac=< macAddr >\r\n

Command Parameters:

macAddr[6 bytes] – Mac address to be set for module.

Note:

In concurrent mode, given MAC is applied to station mode and AP mode MAC address last byte will differ from station mode MAC.

AP mode MAC address last byte will be one plus the station mode MAC address last byte given by host.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0021, 0x0025, 0x002C.

Relevance:
This command is relevant in all operating modes.

Example:
AT Mode:
at+rsi_setmac=001122334455\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x73 0x65 0x74 0x6D 0x61 0x63 0x3D 0x30 0x30 0x31 0x31 0x32 0x32 0x33 0x33 0x34 0x34 0x35 0x35 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.5 Init

Description:
This command programs the module's Baseband and RF components and returns the MAC address of the module to the host. This command has to be issued after band command.

Command:

AT Mode:
at+rsi_init\r\n

Command Parameters:
No parameters

Response:

AT Mode:

Result Code	Description
OK<macAddress>	macAddress (6 bytes, Hex)
ERROR<Error code>	Failure

Response Parameters:

macAddress[6 byte]: The MAC address of the module.

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C,0x0002.

Relevance:
This command is relevant in all operating modes

Example:
AT Mode:
at+rsi_init\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x69 0x6E 0x69 0x74 0x0D 0x0A

Response:
FOR NON-CONCURRENT MODE:
OK<MAC_Address>\r\n
OK 0x00 0x23 0xA7 0x13 0x14 0x15\r\n
0x4F 0x4B 0x00 0x23 0xA7 0x13 0x14 0x15 0x0D 0x0A

FOR CONCURRENT MODE:

Response:
AT Mode:

Result Code	Description
OK <macAddress1> <macAddress2>	macAddress(6 bytes Hex) macAddress(6 bytes Hex)
ERROR<Error code>	Failure

Response Parameters:

macAddress[6 bytes]: The MAC address for two interfaces of the module.

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C,0x0002.

Relevance:
This command is relevant in all operating modes

Example:
AT Mode:
at+rsi_init\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x69 0x6E 0x69 0x74 0x0D 0x0A

Response:
OK<MAC_Address1><MAC_Address2>\r\n
OK 0x00 0x23 0xA7 0x13 0x14 0x15 0x00 0x23 0xA7 0x13 0x14 0x16\r\n
0x4F 0x4B 0x00 0x23 0xA7 0x13 0x14 0x15 0x00 0x23 0xA7 0x13 0x14 0x16 0x0D 0x0A

5.6 User Configurable Gain Table

Description:
This command is used to configure region based gain table to the module from user. This method is used for overwriting default region based gain tables that are present in firmware. This command must be issued immediate to init command. Customer can load all the three gain tables (i.e., 2.4GHz-20Mhz, 5GHz-20Mhz, 5GHz-40Mhz) one after other by changing band and bandwidth values.
Note:

Internally firmware maintains two tables : Worldwide table & Region based table. Worldwide table is populated by firmware with Max power values that chip can transmit that meets target specs like EVM. Region based table has default gain value set.

1)When certifying with user antenna, Region must be set to Worldwide and sweep the power from 0 to 21dBm. Arrive at max power level that is passing certification especially band-edge.

2)These FCC/ETSI/TELEC/KCC Max power level should be loaded in end-to-end mode via WLAN User Gain table. This must be called done every boot-up since this information is not saved inside flash. Region based user gain table sent by application is copied onto Region based table .SoC uses this table in FCC/ETSI/TELEC/KCC to limit power and not to violate allowed limits.

For Worldwide region firmware uses Worldwide table for Tx. For other regions(FCC/ETSI/TELEC/KCC), Firmware uses min value out of Worldwide & Region based table for Tx. Also there will be part to part variation across chips and offsets are estimated during manufacturing flow which will be applied as correction factor during normal mode of operation.

This frame must be used by customers who has done FCC/ETSI/TELEC/KCC certification with their own antenna. All other customers should not use this. Inappropriate use of this frame may result in violation of FCC/ETSI/TELEC/KCC or any certifications and Silicon Labs is not liable for that.

Command:
AT Mode:
at+rsi_gain_table=<Band>,<Bandwidth>,<payload_len>,<payload>\r\n

Command Parameters:

Band[1 byte]:

Mode	Functionality
1	2.4 GHz
2	5 GHz

Bandwidth:

Mode	Functionality
0	20 MHz
1	40 MHz

Note:

In 2.4 GHz band, 40Mhz is not supported.

Payload_len:

1) Max table size in 2.4Ghz is 128 bytes

2) Max table size in 5Ghz is 64bytes.

Payload: pass channel gain values for different regions in the mentioned format.

Gain Table Payload Format:

1. Gain table Format for 2.4G Band: (Each entry of the table is 1 byte)

   In 2Ghz, Max Gain/Power obtained from certification should be doubled and loaded.

```
<TABLE NAME>[] = {
          <NO.of Regions>,
           <REGION NAME 1>, <CHANNEL_CODE_2G>,
        <CHANNEL NUMBER 1>, <2 * MAX POWER FOR b RATE>, <2 * MAX POWER FOR g RATE>, <2 * MAX POWER FOR n RATE>,
        <CHANNEL NUMBER 2>, <2 * MAX POWER FOR b RATE>, <2 * MAX POWER FOR g RATE>, <2 * MAX POWER FOR n RATE>,
        .
        .
        .
        .
        .
        <CHANNEL NUMBER m-1>, <2 * MAX POWER FOR b RATE>, <2 * MAX POWER FOR g RATE>, <2 * MAX POWER FOR n RATE>,
<CHANNEL NUMBER m>, <2 * MAX POWER FOR b RATE>, <2 * MAX POWER FOR g RATE>, <2 * MAX POWER FOR n RATE>,

<REGION NAME 2>, <CHANNEL_CODE_2G>,

<CHANNEL NUMBER 1>, <2 * MAX POWER FOR b RATE>, <2 * MAX POWER FOR g RATE>, <2 * MAX POWER FOR n RATE>,

<CHANNEL NUMBER 2>, <2 * MAX POWER FOR b RATE>, <2 * MAX POWER FOR g RATE>, <2 * MAX POWER FOR n RATE>,

.

.

.

.

<CHANNEL NUMBER m-1>, <2 * MAX POWER FOR b RATE>, <2 * MAX POWER FOR g RATE>, <2 * MAX POWER FOR n RATE>,

<CHANNEL NUMBER m>, <2 * MAX POWER FOR b RATE>, <2 * MAX POWER FOR g RATE>, <2 * MAX POWER FOR n RATE>,


};
```

Gain table Format for 5G Band: (Each entry of the table is 1 byte)

In 5Ghz, Max Gain/Power obtained from certification should be loaded.

```
<TABLE NAME>[] = {

<NO.of Regions>,

   <REGION NAME 1>, <CHANNEL_CODE_5G>,

      <CHANNEL NUMBER IN BAND 1 IF ANY>, <MAX POWER FOR 11a RATE>, <MAX POWER FOR n RATE>,

      <BAND_NUMBER 1>, <MAX POWER FOR 11a RATE>, <MAX POWER FOR n RATE>,

      <CHANNEL NUMBER IN BAND 2 IF ANY>, <MAX POWER FOR 11a RATE>, <MAX POWER FOR n RATE>,

      <BAND_NUMBER 2>, <MAX POWER FOR 11a RATE>, <MAX POWER FOR n RATE>,

      <CHANNEL NUMBER IN BAND 3 IF ANY>, <MAX POWER FOR 11a RATE>, <MAX POWER FOR n RATE>,

      <BAND_NUMBER 3>, <MAX POWER FOR 11a RATE>, <MAX POWER FOR n RATE>,

      <CHANNEL NUMBER IN BAND 4 IF ANY>, <MAX POWER FOR 11a RATE>, <MAX POWER FOR n RATE>,

      <BAND_NUMBER 4>, <MAX POWER FOR 11a RATE>, <MAX POWER FOR n RATE>,

   .

   .

   .

   <REGION NAME y>, <CHANNEL_CODE_5G>,

};
```

2. Supported Region names:

FCC, ETSI,TELEC, KCC

The following are the regions and their values to be passed instead of macros in the example.

Region	Macro Value
FCC	0
ETSI	1
TELEC	2
KCC	4

3.<CHANNEL_CODE_2G> is an 8 bit value which is encoded as:

If Tx powers of all the channels are same, then use CHANNEL_CODE_2G as 17. In this case, mention channel number as 255

If Tx power is not same for all channels, then indicate CHANNEL_CODE_2G as no-of channels. And specify tx power values for all the channels indicated.

4. <CHANNEL_CODE_5G> is a 8 bit value encoded as number of rows in a region for 5G band.

a. 5G is divided into 4 sub bands:

band 1: channel number <= 48

band 2: channel number > 48 and channel number <= 64

band 3: channel number > 64 and channel number <= 144

band 4: channel number > 144

b. If any channel in a band has different set of power values, specify the channel number followed by power values.

c. If all the channels in a band 1 has same power values, specify the band number as 1 followed by power value.

d. If all the channels in a band 2 has same power values, specify the band number as 2 followed by power value.

e. If all the channels in a band 3 has same power values, specify the band number as 3 followed by power value.

f. If all the channels in a band 4 has same power values, specify the band number as 4 followed by power value.

Note:

Length of the payload should match with payload_len parameter value.

Response:

AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0021, 0x003E.

Relevance:
This command is relevant in wlan operating modes.

Example:
AT Mode:
at+rsi_gain_table=0,1,128,<payload>\r\n

Example payload formats:

Examples:

For 2.4Ghz Band in 20Mhz bandwidth:

{3, //NUM_OF_REGIONS

```
    FCC, 13, //NUM_OF_CHANNELS
//   rate,  11b, 11g, 11n
        1,  34,  20,  20,
        2,  34,  28,  28,
        3,  34,  32,  32,
        4,  34,  36,  36,
        5,  34,  38,  38,
        6,  34,  40,  40,
        7,  34,  38,  38,
        8,  34,  36,  36,
        9,  34,  32,  32,
        10, 34,  32,  32,
        11, 34,  24,  24,
        12, 34,  16,  24,
        13, 34,  12,  12,
    TELEC, 17,
         255, 20,  16, 16,
    KCC, 17,
         255, 26,  20, 20,
}; //}}}
```

For 5Ghz band in 20Mhz bandwidth:

```
{2,
FCC, 6,
    1,  9, 10, //band 1
    2,  8,  9, //band 2
  100,  4,  4, //band 3
    3,  6,  8, //band 3
  149,  3,  3, //band 4
    4,  6,  7, //band 4
TELEC, 4,
  1, 9, 10, //band 1
  2, 8, 10, //band 2
  3, 6,  8, //band 3
  4, 6,  7, //band 4
};
```

For 5Ghz band in 40Mhz bandwidth:

```
{2,
FCC, 8,
    1,  9, 10, //band 1
   62,  8,  9, //band 2
    2,  8,  9, //band 2
  102,  4,  4, //band 3
  134,  6,  8, //band 3
    3,  6,  8, //band 3
  151,  3,  3, //band 4
    4,  6,  7, //band 4
TELEC, 4,
   1, 9, 10, //band 1
   2, 8, 10, //band 2
   3, 6,  8, //band 3
   4, 6,  7, //band 4
};
```

Customers using Certified MARS antenna should use the below mentioned gain table structures

```
/******2GHz-20MHz*****/
{3,//NUM_OF_REGIONS
    FCC, 0xD,//NUM_OF_CHANNELS
//   rate,  11b, 11g, 11n
        1,  28,  32,  30,
        2,  28,  32,  30,
        3,  28,  32,  30,
        4,  30,  28,  34,
        5,  30,  28,  34,
        6,  30,  28,  34,
        7,  30,  28,  34,
        8,  30,  28,  34,
        9,  28,  30,  30,
        10, 28,  30,  30,
        11, 28,  30,  30,
        12, 28,  30,  30,
        13, 28,  30,  30,
    TELEC,0x11, //NA
         255, 20,  16, 16,
    KCC, 0x11   //NA,
         255, 26,  20, 20
};

/******5GHz-20MHz*****/
{2,
FCC, 0x6,
    1, 12, 12, //band 1
    2, 11, 11, //band 2
  100, 10, 12, //band 3
    3, 13, 13, //band 3
  140, 10, 11, //band 4
    4, 13, 13, //band 4
TELEC, 0x4, //NA
  1, 9, 10, //band 1
  2, 8, 10, //band 2
  3, 6,  8, //band 3
  4, 6,  7, //band 4
};

/******5GHz-40MHz*****/
{2,
FCC, 0x8,
    1,  9,  9, //band 1
   62,  8,  8, //band 2
    2,  9,  9, //band 2
  102,  9,  9, //band 3
  134, 12, 12, //band 3
    3, 10, 10, //band 3
  151, 11, 11, //band 4
```

```
      4, 11, 11, //band 4
   TELEC, 0x4, //NA
      1, 9, 10, //band 1
      2, 8, 10, //band 2
      3, 6,  8, //band 3
      4, 6,  7, //band 4
   };
```

Response:

OK\r\n
0x4F 0x4B 0x0D 0x0A

Note:

In AT command mode, User configurable gain table command will simply return OK as a response irrespective of payload content.

5.7 PER Mode

Description:

This command configures the PER (Packet Error Rate) Mode in RS9116-WiSeConnect. This command should be issued after *Init* command.

Command Format:

AT Mode:
at+rsi_per=<per_mode_enable>,<power>,<rate>,<length>,<mode>,<channel>,<rate_flags>,<aggr_enable>,<no_of_pkts>,<delay>\r\n

Command Parameters:
per_mode_enable[2 byte]: To enable or disable PER Mode.
0 – disable
1 – enable

power[2 byte]: To set Tx power in dbm.

rate[4 byte]: To set transmit data rate.

Value of rate	Data Rate (Mbps)
0	1
2	2
4	5.5
6	11
139	6
143	9
138	12
142	18
137	24
141	36

Value of rate	Data Rate (Mbps)
136	48
140	54
256	MCS0
257	MCS1
258	MCS2
259	MCS3
260	MCS4
261	MCS5
262	MCS6
263	MCS7

PER Mode Data Rates

length[2 bytes]: To configure length of the TX packet. Valid values are in the range of 24 to 1500 bytes in the burst mode and range of 24 to 260 bytes in the continuous mode

mode[2 byte]: transmit mode

Mode	Functionality
0	Burst Mode
1	Continuous Mode
2	Continuous wave Mode (non modulation) in DC mode
3	Continuous wave Mode (non modulation) in single tone mode (center frequency -2.5MHz)
4	Continuous wave Mode (non modulation) in single tone mode (center frequency +5MHz)

Note:

Before starting Continuous Wave mode, it is required to start Continuous mode with power and channel values which is intended to be used in Continuous Wave mode i.e.

1. Start Continuous mode with intended power value and channel values
   - Pass any valid values for rate and length
2. Stop Continuous mode
3. Start Continuous Wave mode.

channel[2 bytes]: For setting the channel number in 2.4 GHz/5GHz .

The following tables map the channel number to the actual radio frequency in the 2.4 GHz spectrum.

Channel Numbers (2.4GHz)	Center frequencies for 20MHz channel width (MHz)
1	2412
2	2417
3	2422
4	2427
5	2432
6	2437

Channel Numbers (2.4GHz)	Center frequencies for 20MHz channel width (MHz)
7	2442
8	2447
9	2452
10	2457
11	2462
12	2467
13	2472
14	2484

Channel Number and Frequencies for 20MHz Channel Width in 2.4GHz

Note:

To support PER mode in 12,13,14 channels, set region command has to be given by the host before PER command.

Channel numbers in 5 GHz range from 36 to 165. The following table map the channel number to the actual radio frequency in the 5 GHz spectrum for 20MHz channel bandwidth.

Channel Numbers (5GHz)	Center Frequencies for 20MHz channel width (MHz)
36	5180
40	5200
44	5220
48	5240
52	5260
56	5280
60	5300
64	5320
100	5500
104	5520
108	5540
112	5560
116	5580
120	5600
124	5620
128	5640
132	5660
136	5680
140	5700

Channel Numbers (5GHz)	Center Frequencies for 20MHz channel width (MHz)
144	5720
149	5745
153	5765
157	5785
161	5805
165	5825

Channel Number and Frequencies for 20MHz Channel Width in 5GHz

The following tables map the channel number to the actual radio frequency in the 4.9 GHz spectrum for 802.11J.

Channel Numbers (4.9GHz)	Center frequencies for 20MHz channel width (MHz)
184	4920
188	4940
192	4960
196	4980
8	5040
12	5060
16	5080

Note:

802.11J features are valid only when Japan region is set. For other regions even if 802.11J is enabled it has no effect.

rate_flags[2 bytes]: Rate flags contain short GI, Greenfield and channel width values. Various fields in rate flags are divided as specified below:

Fields	Short GI	Greenfield	Channel Width	Reserved
Bits:	0	1	2-4	5-15

Rate Flags

To enable short GI – set rate flags value as '1'
To enable Greenfield – set rate flags value as '2'

Note:

Short GI is not supported in this release.

Channel width should be set to zero to set 20MHz channel width.
Reserved1: reserved bytes. This field can be ignored. Set to '0.'

aggr_enable[2 byte]: This flag is for enabling or disabling aggregation support.

Note:
Aggregation feature is supported only in burst mode. This field will be ignored in case of continuous mode.

Reserved2: reserved bytes. This field can be ignored. Set '0'

no_of_pkts[2 byte]: This field is used to set the number of packets to be sent in burst mode. If the value given is 'n' then 'n' number of packets will be sent on air, after that transmission will be stopped. If this field is given as zero (0) then packets will be sent continuously until user stops the transmission. This field will be ignored in case of continuous mode

delay[4 byte]: This field is used to set the delay between the packets in burst mode. Delay should be given in microseconds. i.e. if the value is given as 'n' then a delay of 'n' microseconds will be added for every transmitted packet in the burst mode.
If this field is set to zero (0) then packets will be sent continuously without any delay. This field will be ignored in case of continuous mode.

Note:

Only per_mode_enable , power, rate, length , mode, channel, rate_flags fields are valid. Remaining fields are not supported.

Response:

AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x000A,0x0021, 0x0025, 0x002C.

Relevance:
This command is relevant when the module is configured in operating mode 8.

Example:
AT Mode:

To start transmit:
at+rsi_per=1,18,139,30,0,1,0,0,0,0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x70 0x65 0x72 0x3D 0x31 0x2C 0x31 0x38 0x2C 0x31 0x33 0x39 0x2C 0x33 0x30 0x2C 0x30 0x2C 0x31 0x2C 0x30 0x2C 0x30 0x2C 0x30 0x2C 0x30 0x0D 0x0A

To stop transmit:
at+rsi_per=0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x70 0x65 0x72 0x3D 0x30 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

The following table providing the mapping between <power> and output power in dBm.

						Q7	M7DB	M4SB	W3	M15SB							
					Module_Offset	2	3	2	2	3							
WorldWide Region																	
All Power values are in dBm																	
2.4GHz	20MHz								5GHz	20MHz				M7DB			

			Max Power @ Antenna =Max Power @ Chip_out-Module Offset							Max Power table				Max Power table - Module Offset			
	Data Rate	Max Power @ Chip_out (As per dotc)	Q7	M7DB	M4SB	W3	M15SB		Data Rate	BAND-1	BAND-2	BAND-3	BAND-4	BAND-1	BAND-2	BAND-3	BAND-4
	1	20	18	17	18	18	17		6	14	13	12	12	11	10	9	9
	2	20	18	17	18	18	17		9	14	13	12	12	11	10	9	9
	5.5	20	18	17	18	18	17		12	14	13	12	12	11	10	9	9
	11	20	18	17	18	18	17		18	14	13	12	12	11	10	9	9
	6	21	19	18	19	19	18		24	14	13	12	12	11	10	9	9
	9	21	19	18	19	19	18		36	13	12	10	10	10	9	7	7
	12	21	19	18	19	19	18		48	11	10	8	8	8	7	5	5
	18	20	18	17	18	18	17		54	9	8	7	6	6	5	4	3
	24	19	17	16	17	17	16		MCS0	14	13	12	12	11	10	9	9
	36	17	15	14	15	15	14		MCS1	14	13	12	12	11	10	9	9
	48	16	14	13	14	14	13		MCS2	14	13	12	12	11	10	9	9
	54	16	14	13	14	14	13		MCS3	13	13	12	12	10	10	9	9
	MCS0	20	18	17	18	18	17		MCS4	11	12	10	10	8	9	7	7
	MCS1	20	18	17	18	18	17		MCS5	10	10	8	8	7	7	5	5
	MCS2	20	18	17	18	18	17		MCS6	8	8	6	6	5	5	3	3
	MCS3	19	17	16	17	17	16		MCS7	7	6	5	3	4	3	2	0
	MCS4	17	15	14	15	15	14										
	MCS5	16	14	13	14	14	13										
	MCS6	16	14	13	14	14	13										

	MCS 7	14	12	11	12	12	11										
2.4GHz	40MHz								5GHz	40MHz				M7DB			
			Max Power @ Antenna =Max Power @ Chip_out-Module Offset							Max Power table				Max Power table - Module Offset			
	Data Rate	Max Power @ Chip_out	Q7	M7DB	M4SB	W3	M15SB		Data Rate	BAND-1	BAND-2	BAND-3	BAND-4	BAND-1	BAND-2	BAND-3	BAND-4
	MCS 0	8	6	5	6	6	5		MCS0	10	10	9	9	7	7	6	6
	MCS 1	8	6	5	6	6	5		MCS1	10	10	9	9	7	7	6	6
	MCS 2	8	6	5	6	6	5		MCS2	10	10	9	9	7	7	6	6
	MCS 3	8	6	5	6	6	5		MCS3	10	10	9	9	7	7	6	6
	MCS 4	8	6	5	6	6	5		MCS4	10	10	9	9	7	7	6	6
	MCS 5	8	6	5	6	6	5		MCS5	10	10	9	9	7	7	6	6
	MCS 6	8	6	5	6	6	5		MCS6	9	8	7	7	6	5	4	4
	MCS 7	8	6	5	6	6	5		MCS7	8	6	5	5	5	3	2	2

5.8 Configure AP Mode

Description

This command is used to set the configuration information for AP mode. This command must be issued after *init* command.

Payload

AT Mode:

at+rsi_apconf = <channel_no>,<ssid>,<security_type>,< encryp_mode>, <psk>,<beacon_interval>,<dtim_period>,< max_sta_support>,<ap_keepalive_type>,<ap_keepalive_period>\r\n

Parameters

channel_no[2 bytes]: The channel in which the AP would operate. Refer the **PER Mode** section.

A value of zero enables the Auto channel selection feature.
0 - Auto channel selection.

Auto Channel Selection is used to enable interfaces to automatically figure out which **channel** configuration to use. It sets the channel which has low traffic.

Note:

DFS channels are not supported in ap mode.

ssid[32 bytes]: SSID of the AP to be created

Note:

1. To support comma (,) in SSID, user needs to give SSID with in double quotes ("<SSID>").

   Ex: “MY, NETWORK”
2. Not supported double quotes (") in SSID.

security_type[1 byte]: Security type.

Mode	Functionality
0	Open
1	WPA
2	WPA2

encryp_mode[1 byte]: Encryption type.

Mode	Functionality
0	Open
1	TKIP
2	CCMP

psk[64 bytes]: PSK of the AP in security mode. If the AP is in Open mode, this parameter can be set to '0'.

Note:

Minimum and maximum length of PSK is 8 bytes and 63 bytes respectively.

beacon_interval[2 bytes]: Beacon interval of the AP in milliseconds. Allowed values are integers from 100 to 1000 which are multiples of 100.

dtim_period[2 bytes]: DTIM period. Allowed values are from 1 to 255

ap_keepalive_type[1 byte]: This is the bitmap to enable AP keep alive functionality and to select the keep alive type.

BIT[0]: To enable/disable keep alive functionality.
1 - To enable keep alive functionality.
0 - To disable keep alive functionality.

BIT[1]: To select AP keep alive method.
1 - To enable null data based keep alive functionality.
0 - default keep alive functionality (i.e. disconnect the station if there are no wireless exchanges from station with in ap_keepalive_period).

ap_keepalive_period[1 byte]: This is the period after which AP will disconnect the station if there are no wireless exchanges from station to AP. Keep alive period is calculated in terms of 32 multiples of beacon interval(i.e. if there are no wireless transfers from station to AP with in (32*beacon_interval*keep_alive_period) milli seconds time period, station will be disconnected).If null data based method is selected, AP checks the connectivity of station by sending null data packet. If station does not ack the packet, that station will be disconnected after 4 retries.

max_sta_support[2 bytes]: Number of clients supported. This value should be less than or equal to the value given in custom feature select bit map[BIT[13:16]] of the **Set Operating Mode command**. If value is not set in custom feature select bit map[BIT[13:16]] of the **Set Operating Mode command** then maximum supported stations are 4.

Note:

In RS9116W there is Support of connecting 16 clients to the created AP by setting custom feature bitmap [Bit[13-16]] and also extended custom feature bitmap [Bit[15]] by using the equation mentioned below:

Number of stations = (Stations Obtained by setting Bit[13-16] + 1) * 2

For example if host want 16 clients support in AP then need to set following bits BIT[13], BIT[14] and BIT[15] (leave BIT[16] as 0) in custom feature bitmap and also BIT[15] in extended custom feature bitmap then number of stations will become (7+1) * 2 = 16. If you are configuring more than 16 stations, then it will throw an error.

If you are not setting extended custom feature bitmap, then it can configure maximum of 8 stations. This is Backward Compatibility this case also if you are configuring more than 8 stations then it will throw an error.

Response
AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes

Possible error codes are 0x0021,0x0025,0x002C,0x0026,0x004C,0x0028,0x001A,0x000A,0x001D

Relevance
This command is relevant when the module is configured in Operating Mode 6.

Example:
AT Mode:
Configured AP with channel num =11, ssid = ap_ssid, open mode, beacon interval = 100, DTIM count =3 and max stations support =3.
at+rsi_apconf=11,ap_ssid,0,0,0,100,3,3\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x61 0x70 0x63 0x6F 0x6E 0x66 0x3D 0x31 0x31 0x2C 0x72 0x65 0x64 0x70 0x69 0x6E 0x65 0x2C 0x30 0x2C 0x30 0x2C 0x30 0x2C 0x31 0x30 0x30 0x2C 0x33 0x2C 0x33 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.9 WPS PIN Method

Description:
This command configures the WPS PIN method to be used in RS9116-WiSeConnect.This command should be issued before join command.

Command Format:
AT Mode:
at+rsi_wps_method=<wps_method>,<generate_pin>,<wps_pin>\r\n

Command Parameters:

wps_method[2 bytes]: WPS method type Should set to '1' for PIN method.

generate_pin[2 bytes]: This parameter specifies whether to validate entered pin or generate pin .This parameter is valid only if wps_method is 1.
0-Use entered pin in wps_pin field.
1-pin generation

If generate_pin is 0, module will validate the given 8-digit wps_pin. If pin given is less than 8 digit or if pin is wrong, then module will give error.

wps_pin[8 bytes]: wps_pin is of 8-digit pin. Module validates and uses this pin only in case of when wps_method is pin method and generate_pin is 0.

Response:

Note:

Response contains following payload only if PIN method is selected. In case of PUSH method response does not contains any payload.

AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Response Parameters:

wps_pin: The WPS PIN will be used by the module to connect with WPS AP.

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C, 0x0037, 0x0038.

Relevance:
This command is relevant when the module is configured in Operating Mode 0 and 6.

Example:

AT Mode:

When PIN of length 8 is given

at+rsi_wps_method=1,0,12345678\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x77 0x70 0x73 0x5F 0x6D 0x65 0x74 0x68 0x6F 0x64 0x3D 0x31 0x2C 0x30 0x2C 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x0D 0x0A

Response:

OK 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08\r\n
0x4F 0x4B 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x0D 0x0A

When PIN of length less than 8 is given

at+rsi_wps_method=1,1,1234\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x77 0x70 0x73 0x5F 0x6D 0x65 0x74 0x68 0x6F 0x64 0x3D 0x31 0x2C 0x30 0x2C 0x31 0x32 0x33 0x34 0x0D 0x0A

5.10 Scan

Description:
This command scans for Access Points and gives the scan results to the host. The scan results are sorted in decreasing order of signal strength (RSSI value). The scanned access point with highest signal strength will be the first in the list. This command has to be issued after *init* command and before *Join* command.

Command Format:
AT Mode:
at+rsi_scan=<channel>,<ssid>,<channel_bit_map_2_4>,<channel_bit_map_5>\r\n
or
at+rsi_scan=<channel>,<ssid>,<scan_feature_bitmap>\r\n

Command Parameters:
Channel[4 byte]: Channel Number on which scan has to be done. If this value is 0, the module scans in all the channels in the band that is selected through the band command. The values of this parameter are listed in table below To select DFS channels user need to set custom feature bit in opermode command.

Note:

1. If chan_num is 0 and channel bit maps (selective scan) are provided, then module will scan only the channels specified in bitmaps instead of scanning all channels.
2. In case of 5GHz, module performs passive scan in DFS channels only when BIT[8] is set in custom feature bit map in **Set Operating Mode** command.
3. scan feature bitmap
   BIT(0) (QUICK SCAN feature) -It is valid only if channel number and ssid is given.
   BIT(1) (SCAN RESULTS TO HOST) - when it is enabled additional scan results are given to host. After getting scan results, host has to issue another scan request by disabling this bit in scan feature bitmap before issuing join command.
4. If channel bitmap is specified, Module will scan only channels which are valid in selected region

Table 1: Channels in 2.4 GHz Mode

Channel Number	**chan_num parameter**
All channels	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14

Note:

Scanning in 12,13,14 channels is allowed based on the region selected in **Set Region** command.

Table 2: Channels in 5GHz Mode

Channel Number	**chan_num parameter**
All channels	0
36	36
40	40

Channel Number	chan_num parameter
44	44
48	48
52	52
56	56
60	60
64	64
100	100
104	104
108	108
112	112
116	116
132	132
136	136
140	140
144	144
149	149
153	153
157	157
161	161
165	165

Channel Number(4.9GHz)	chan_num parameter
All channels	0
184	184
188	188
192	192
196	196
8	8
12	12
16	16

ssid[32 byte]: Optional Input. For scanning a hidden Access Point, its SSID can be provided as part of the SCAN command. The maximum number of scanned networks reported to the host is 11. If not used, null characters should be supplied to fill the structure.

Note:

1. To support comma (,) in SSID, user needs to give SSID with in double quotes ("<SSID>").

   Ex: “MY, NETWORK”

1. Not supported double quotes (") in SSID.

Reserved: Set to '0's.
scan_feature_bitmap[1 byte]: Scan feature bitmap

BIT[0]: To enable/disable quick scan feature.
1 - To enable quick scan feature.
0 - To disable quick scan feature.

BIT[1]-BIT[7]: Reserved.
channel_bit_map_2_4[2 bytes]: channel bitmap for scanning in set of selective channels in 2.4Ghz.

Channel_bit_map_5[4 bytes]: channel bitmap for scanning a set of selective channels in 5Ghz.

Note:

For 11J channel bit map need to give in Channel_bit_map_5.

Table 3: Channel Number to Bitmap Mapping in 2.4GHz

Channel Number	Channel bit position in bitmap
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10
12	11
13	12
14	13

Table 4: Channel Number to Bitmap Mapping in 5GHz

Channel Number	chan_num parameter
36	0
40	1
44	2
48	3
52	4

Channel Number	chan_num parameter
56	5
60	6
64	7
100	8
104	9
108	10
112	11
116	12
120	13
124	14
128	15
132	16
136	17
140	18
144	19
149	20
153	21
157	22
161	23
165	24

Channel Number(4.9GHz)	Channel bit position in bitmap
8	0
12	1
16	2
184	3
188	4
192	5
196	6

AT Mode:

Result Code	Description
OK< scanCount >< padding > < rfChannel >< securityMode >< rssiVal >< uNetworkType >< ssid >< bssid >< reserved > …….up to the number of scanned nodes	Successful execution of the command.
ERROR<Error code>	Failure

Response Parameters:

scancount(4 bytes): Number of Access Points scanned

padding(4 bytes): padding bytes which can be ignored.

rfChannel(1 byte): Channel Number of the scanned Access Point

security Mode(1 byte):

Mode	Functionality
0	Open
1	WPA
2	WPA2
3	WEP
4	WPA Enterprise
5	WPA2 Enterprise

rssival(1 byte): RSSI of the scanned Access Point

uNetworkType(1 byte): Network type of the scanned Access Point
1– Infrastructure mode

ssid(32 bytes): SSID of the scanned Access Point

bssid(6 bytes): MAC address of the scanned Access Point.

Reserved(2 bytes): Reserved bytes.

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C, 0x0014, 0x0002, 0x0003, 0x0024, 0x001A, 0x0015, 0x000A, 0x0026.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 2, 6.

Example:
AT Mode:
To scan all the networks in all channels
at+rsi_scan=0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x73 0x63 0x61 0x6E 0x3D 0x30 0x0D 0x0A

To scan a specific network "Test_AP" in a specific channel 6
at+rsi_scan=6,Test_AP\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x73 0x63 0x61 0x3D 0x36 0x2C 0x54 0x65 0x73 0x74 0x5F 0x41 0x50 0x0D 0x0A

Response:
If two networks are found with the SSID "ap_ssid_net1" and "ap_ssid_net2", in channels 6 and 10, with measured RSSI of -20dBm and -14dBm respectively, the return value is
O K < scanCount =2> < padding > < rfChannel =0x0A> < securityMode =0x02> < rssiVal =14> < uNetworkType =0x01> < ssid =ap_ssid_net2> < bssid =0x00 0x23 0xA7 0x1F 0x1F 0x15> < reserved >< rfChannel =0x06> < securityMode =0x00> < rssiVal =20> < uNetworkType =0x01> < ssid =ap_ssid_net1> < bssid =0x00 0x23 0xA7 0x1F 0x1F 0x14> < reserved > \r\n
0x4F 0x4B 0x02 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0A 0x02 0x0D 0x01 0x52 0x65 0x64 0x70 0x69 0x6E 0x65 0x5F 0x6E 0x74 0x32 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x23 0xA7 0x1F 0x1F 0x15 0x00 0x00 0x06 0x00 0x14 0x01 0x52 0x65 0x64 0x70 0x69 0x6E 0x65 0x5F 0x6E 0x74 0x31 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x23 0xA7 0x1F 0x1F 0x14 0x00 0x00 0x0D 0x0A

5.11 Join

Description:
This command is used for following:

1. Associate to an access point (operating mode = 0 or 2)

2. Create an Access Point (operating mode 6)
3. To enable WPS PUSH method in Access point mode

Command Format:
AT Mode:
at+rsi_join=<ssid>,<dataRate>,<powerLevel>,<Security_mode>,<join_feature_bitmap>,<listen_interval>,<vap_id>,<join_bssid>\r\n

Note:
1. To support comma (,) in SSID, user needs to give SSID with in double quotes ("<SSID>").
   i. Ex: “MY, NETWORK”
2. Not supported double quotes (") in SSID.

Command Parameters:

reserved1: Reserved. Set to '0'

Security_mode[1 byte]: This variable is used to define the security mode of the Access point to which module is supposed to connect.

Possible values:

Mode	Functionality
0	Connect only to AP in open mode
1	Connect to AP in WPA mode
2	Connect to AP in WPA2 mode
3	Connect to AP in WEP open mode
4	Connect to AP in EAP WPA mode
5	Connect to AP in EAP WPA2 mode

Note:
1. Security_mode parameter is valid only if opermode is 0 or 2.
2. psk is required for security mode 1,2,6. Otherwise module returns Join failure with error 0x16.
3. In Enterprise mode (Security_mode 4,5), module will derive the PSK using EAP exchanges with Authentication server.
4. Module strictly obey security mode specified in Join command, not depends on psk.
5. In opermode 6, Once Access point is created host can enable WPS PUSH method by giving JOIN command (with same parameters which were used to create Access point) again.
6. WPS method is not supported in Coex mode.

dataRate[1 byte]: Transmission data rate. Physical rate at which data has to be transmitted.

Table 5: Transmission Data Rates

Data Rate (Mbps)	Value of Data Rate
Auto-rate	0
1	1

Data Rate (Mbps)	Value of Data Rate
2	2
5.5	3
11	4
6	5
9	6
12	7
18	8
24	9
36	10
48	11
54	12
MCS0	13
MCS1	14
MCS2	15
MCS3	16
MCS4	17
MCS5	18
MCS6	19
MCS7	20

powerLevel[1 byte]: This fixes the Transmit Power level of the module. This value can be set as follows:

At 2.4GHz

Mode	Functionality
0	Low power (7+/-1) dBm
1	Medium power (10 +/-1) dBm
2	High power (18 + /- 2) dBm

At 5 GHz

Mode	Functionality
0	Low power (5+/-1) dBm
1	Medium power (7 +/-1) dBm
2	High power (12 + /- 2) dBm

psk: Passphrase used in WPA/WPA2-PSK security mode.

In open mode, WEP mode, Enterprise Security. this should be filled with NULL characters.

Ssid: When the module is in Operating modes 0 or 2, this parameter is the SSID of the Access Point (assuming WPS is not enabled in the Access Point).
When the module is in operating modes 0 or 2 and wants to connect to an access point in WPS mode then the value of this parameter is NULL.

When an Access Point needs to be created, this parameter should be the same as the parameter ssid in the command "Configure AP mode".

Note:

1. To support comma (,) in SSID, user needs to give SSID with in double quotes ("<SSID>").

   Ex: “MY, NETWORK”

2. Not supported double quotes (") in SSID.

Reserved2: Reserved, set to '0'

ssid_len: Actual length of the SSID

join_feature_bitmap[1 byte]:

join_feature_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
join_feature_bit_map[0]	b/g only mode in station mode	Disable	Enable	
join_feature_bit_map[1]	listen interval from join command	Disable	Enable	
join_feature_bit_map[2]	quick join feature	Disable	Enable	
join_feature_bit_map[3]	CCXV2	Disable	Enable	
join_feature_bit_map[4]	AP based on BSSID	Disable	Enable	
join_feature_bit_map[5]	Management Frame Protection Capable only (802.11W)	Disable	Enable	
join_feature_bit_map[6]	Management Frame Protection required (802.11W)			BIT[5] and BIT[6] valid when 11W(BIT[13] in ext custom feature bitmap) enabled, if both bits are not set it will disable PMF.
join_feature_bit_map[7]	listen interval from power save command	Disable	Enable	

listen_interval[4 bytes]: This is valid only if BIT(1) in join_feature_bit_map is set. This value is given in Time units (1024 microsecond). This parameter is used to configure maximum sleep duration in power save.

Note:

To ensure data for module is buffered for sufficient time in access point,

1. Listen interval in association request is incremented by 6 if user configured interval is greater than 11.
2. If user configured listen interval less than or equal to 11, by default module will send listen interval 16 in association request.

But during power save module goes to sleep for user defined listen interval only.

Vap_id[1 byte]: Possible values are 0 and 1.
When 0 – Module will try to connect to scanned AP.
When 1 – Module will create AP.

join_bssid[6 bytes]: This contains BSSID of selected AP. This is valid only if

join_feature_bitmap BIT[4] is set otherwise module will ignore the value.

Note:

1. vap_id will be considered only in concurrent mode.
2. In concurrent mode, if connected station network is same as default dhcp server network then dhcp server will not start but join command for AP creation will give success message to host.

5.12 Request Timeout

Description:
This command is used to set various timeouts. Currently this command can be used to set the authentication and association request timeouts.

Command Format:

AT Mode:
at+rsi_timeout=<timeout_bitmap>,<timeout _value>\r\n

Command Parameters:

timeout_bitmap[4 bytes]:

timeout_bitmap	Functionality
timeout_bitmap[0]	sets timeout for association and authentication request. timeout_value: timeout value in ms (default 300ms).
timeout_bitmap[1]	Sets each channel active scan time in ms (default 100ms)
timeout_bitmap[2]	Sets the WLAN keep alive time in seconds (default value is 90s)

Note:

For Setting WLAN Keep alive timeout need to give time out command before init. If timeout is given as '0'(zero). Keep alive functionality will be disabled.

Example:

AT Mode:
To set authentication and association request timeout of 1.5 seconds
at+rsi_timeout=1,1500\r\n

5.13 Re-Join

Description:
The module automatically tries to re-join if it loses connection to the network it was associated with. If the re-join is successful, then the WLAN link is re-established. During the time the module is trying to re-join, if the Host sends any command, the module does not accept it and throws an error code 37 in status code. The module aborts the re-join after a fixed number of re-tries (maximum number of retries for rejoin is 20 by default). If this happens, an asynchronous message is sent to the Host with an error code 25. User can configure the rejoin parameters using rejoin command.

Note:

When Re-join fails module will close all prior opened TCP/IP sockets.

Command Format:
AT Mode:
at+rsi_rejoin_params=< rsi_max_try >,< rsi_scan_interval >,
< rsi_beacon_missed_count >,< rsi_first_time_retry_enabled >\r\n

Command Parameters:
rsi_max_try[4 bytes]: This represents the number of attempt for join before giving up the error.

> **Note**:
>
> If number of rejoin attempts is 0 then module will try infinitely for rejoin.

rsi_scan_interval[4 bytes]: This is the time interval in seconds for the subsequent retry.
rsi_beacon_missed_count[4 bytes]: This is the beacon missed count that module used to declare module connection status. If module found continuous beacon missed is greater than or equal to this value then it will declare connection as disconnected and will start rejoin process again.
rsi_first_time_retry_enable[4 bytes]: If this is set to 1 then module will retry to connect if first join attempt fails. Number of attempts and scan interval may be configured by rsi_max_try and rsi_scan_interval respectively.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0025, 0x002C.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 2.

Example:

AT Mode:
N/A

Response:

Asynchronous responses from module:

Following message to indicate that module is in process of rejoin, so unable to process requested command.
ERROR<Error code=37>\r\n
0x45 0x52 0x52 0x4F 0x52 0x25 0x00 0x0D 0x0A

Following message to indicate rejoin failure to host.
ERROR<Error code=25>\r\n
0x45 0x52 0x52 0x4F 0x52 0x19 0x00 0x0D 0x0A

5.14 WMM PS

Description:
This command is used to enable WMM PS configurations. This command should be issued before join command and before power save command.

Command Format:

AT Mode:
at+rsi_wmm_config=< wmm_ps_enable >< wmm_ps_type >< wmm_ps_wakeup_interval >< wmm_ps_uapsd_bitmap >\r\n

Command Parameters:

wmm_ps_enable[2 bytes]: To enable or disable WMM PS
0 - disable
1 - enable

wmm_ps_type[2 bytes]: WMM PS type
0 - Tx Based
1 - Periodic

wmm_ps_wakeup_interval[4 bytes]: Wakeup interval in milli seconds.

wmm_ps_uapsd_bitmap[1 byte]: Bitmap , 0 to 15 possible values.

Note:

Individual ACs shouldn't be configured for WMM power save. If user wants to enable UAPSD, then UAPSD bitmap should be 0xF (need to enable all AC's).

wmm_ps_uapsd_bitmap	Funtionality
wmm_ps_uapsd_bitmap[0]	Access category: voice
wmm_ps_uapsd_bitmap[1]	Access category: video
wmm_ps_uapsd_bitmap[2]	Access category: Background
wmm_ps_uapsd_bitmap[3]	Access category: Best effort U-APSD
wmm_ps_uapsd_bitmap[4:7]	All set to '0'. Don't care bits.

Parameters wmm_ps_type, wakeup_interval, wmm_ps_uapsd_bitmap will be used for WMM-PS if Power save is enabled and psp_type given as UAPSD.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Example:
AT Mode:
at+rsi_wmm_config=1,1,0,10\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x77 0x6D 0x6D 0x5F 0x70 0x73 0x3D 0x31 0x2C 0x31 0x2C 0x30 0x2C 0x31 0x30
0x0D 0x0A

Response:
OK\r\n
…………..
0x4F 0x4B 0x0D 0x0A

5.15 Set Sleep Timer

Description:
This command configures the sleep timer mode of the module to go into sleep during power save operation. The command can be issued any time in case of power save mode 9. If this command is not issued, then by default module takes 3 seconds as sleep timer.

Command Format:
AT Mode:
at+rsi_sleeptimer=< TimeVal >\r\n

Command Parameters:

TimeVal[2 bytes]: Sleep Timer value in seconds.
Minimum value is 1, and maximum value is 2100.

Response:

AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0025, 0x002C.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 2.

5.16 Power Mode

Description:
This command configures the power save mode of the module. Power save is disabled by default. The command can be issued any time after the Join command in case of power save mode 1, 2 and 3.
And after Init command before join command in case of power save mode 8 and 9.

Note:

1. RS9116-WiSeConnect doesn't support power save modes while operating in AP or group owner mode.
2. Power save modes 2 and 8 are not supported in USB interface.
3. In SPI interface when ULP mode is enabled, after wakeup from sleep, host has to initialize SPI interface of the module.
4. To use number of dtim skip feature, listen interval should be disable from join command, listen_interval_dtim param in at+rsi_pwmode command shoud be 1(dtim alligned).

Command Format:
AT Mode:
at+rsi_pwmode=< powerVal >,< ulp_mode_enable >,
<listen_interval_dtim>,<PSP_type>,<monitor_interval><num_of_dtim_skip>,<listen_interval>\r\n

Command Parameters:
powerVal[1 byte]:

Mode	Functionality
0	Mode 0: Disable power save mode
1	Power save Mode 1
2	Power save Mode 2
3	Power save Mode 3
8	Power save Mode 8
9	Power save Mode 9

ulp_mode_enable[1 byte]:

Mode	Functionality
0	Mode 0: Disable power save mode
1	Ultra low power mode with RAM retention. Valid for powerVal modes 2,3,8 and 9.
2	Ultra low power mode without RAM retention. Valid for powerVal modes 8 and 9.

listen_interval_dtim[1 byte]:

According to set or reset of this param, the module computes the desired sleep duration based on listen interval (from join command) and its wakeup align with Beacon or DTIM Beacon (based on this parameter).
0 - module wakes up before nearest Beacon that does not exceed the specified listen interval time.
1 - module wakes up before nearest DTIM Beacon that does not exceed the specified listen interval time.

psp_type[1 byte]: This parameter shows Power Save Procedure type used. Following is the values for the PSP_type.
0 – Max Power save procedure.
1 – Fast power save procedure.
2 – UAPSD power save

Note:

1. When fast psp is enabled, module will disable power save for monitor interval of time for each data packet received or sent.
2. UAPSD power save is valid only if wmm is enabled through wmm ps command

Monitor_interval[2 bytes]: This is time in ms to keep module in wakeup state for each Tx or Rx traffic sent or received respectively. Default value for this is 50 ms.

num_of_dtim_skip[1 byte]:

This parameter is to skip the number of dtim. if its value is n then our module will wake up at (n+1)th dtim at each wakeup cycle.

To use this feature, ensure following condition,
BIT(1) is reset in the join_feature_bitmap in join command

listen_interval[2 bytes]: This is valid only if BIT(7) in join_feature_bit_map is set. This value is given in Time units (1024 microsecond). This parameter is used to configure

maximum sleep duration in power save and should be less than the listen interval configured in join command.

Note:

If the User wants to change the Listen_interval dynamically,then user needs to disable Power save and enable power save again with a new listen_interval.

Response:

AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0021, 0x0025, 0x002C, 0xFFF8,0x0015,0x0026,0x0052

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 2.

5.16.1 Power Save Operation

The behavior of the module differs according to the power save mode it is configured.

The following terminology can be used in the below section in order to describe the functionality.

Protocol	Non-Connected State	Connected State
WLAN	This mode is significant when module is not connected with any AP	This mode is significant when module is in associated state with AP
BT Classic	This mode is significant when module is in Idle (standby) state.	This mode is significant when module is in Connected sniff mode, Discoverable mode (ISCAN) and Connectable mode (PSCAN)
BLE	This mode is significant when module is in Idle (standby) state.	This mode is significant when module is in Advertising state, Scan state or Connected state.

Note:

1. In case of WLAN, wake up period will be calculated based on DTIM interval.
2. In case of BT-Classic, wake up period will be calculated based on inquiry scan interval in discoverable mode, page scan interval in connectable mode and sniff interval in connected mode.
3. In case of BLE, wake up period will be calculated based on advertise interval in advertising state, scan interval in scanning state and connection interval in connected state.
4. If incase BT/BLE wakeup period is lesser than the WLAN wakeup period, the module will wake up and servs BT/BLE and go back to the sleep again.

5.16.2 Power Save Mode 1

Once the module is configured to power save mode 1, it wakes itself up periodically based upon the DTIM interval configured in connected AP. In power mode 1, only the RF of the module is in power save while SOC continues to work normally. After successful execution of command, confirmation is received in response. This command has to be given only when module is in connected state (with the AP).

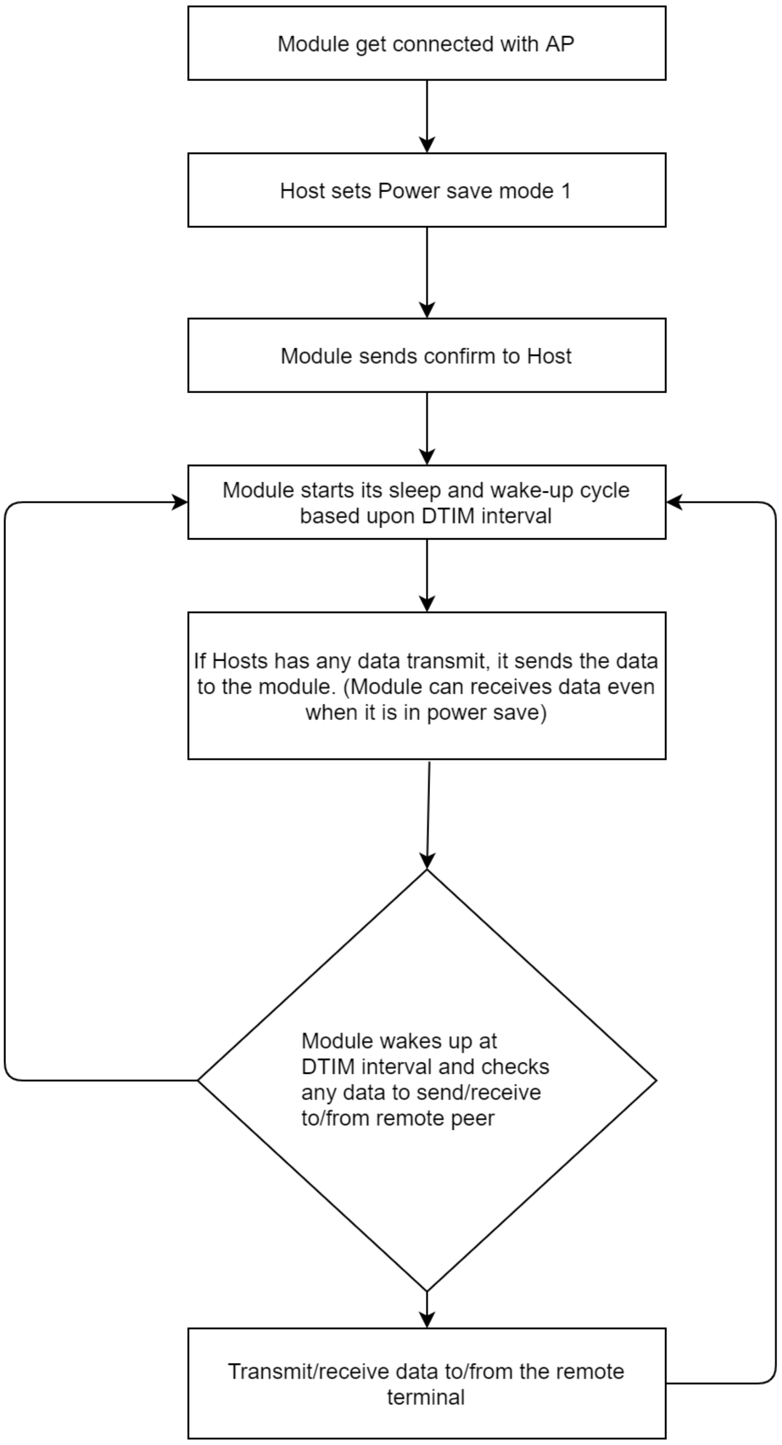

Module get connected with AP
Host sets Power save mode 1
Module sends confirm to Host
Module starts its sleep and wake-up cycle based upon DTIM interval
If Hosts has any data transmit, it sends the data to the module. (Module can receives data even when it is in power save)
Module wakes up at DTIM interval and checks any data to send/receive to/from remote peer
Transmit/receive data to/from the remote terminal

Figure 21: Power Save Mode 1

After having configured the module to power save mode, the Host can issue subsequent commands. In power save mode 1 the module can receive data from host at any point of time, but it can send/receive the data to/from remote terminal only when it is awake at DTIM intervals.

5.16.3 Power Save Mode 2

Once the module is configured to power save mode 2, it can be woken up either by the Host or periodically during its sleep-wake up cycle based upon the DTIM interval.
Power mode 2 is GPIO based. In ULP mode, feature_bit_map[4]has to be set in opermode command. In this mode, Whenever host want to send data to module, gives wakeup indication to module by setting ULP_GPIO_5 high in case of LP or UULP_GPIO_2 in case of ULP(which make module to wake up from power save). After wakeup, if the module is ready for data transfer, it sends wakeup indication to host. Host required to wait until module give wakeup indication before sending any data to module.
After completion of data transfer host can give sleep permission to module by resetting ULP_GPIO_5 in case of LP or

UULP_GPIO_2 in case of ULP. After recognizing sleep permission from host, module give confirmation to host by resetting UULP_GPIO_3 and again gets back to its sleep-wakeup cycle.
Module can send received packets or responses to host at any instant of time. No handshake is required on Rx path.

Note:

1. By default UULP_GPIO_3 is used for wakeup indication to host.
2. If BIT(0) is set in config_feature_bit_map then UULP_GPIO_0 is used for wakeup indication to host.

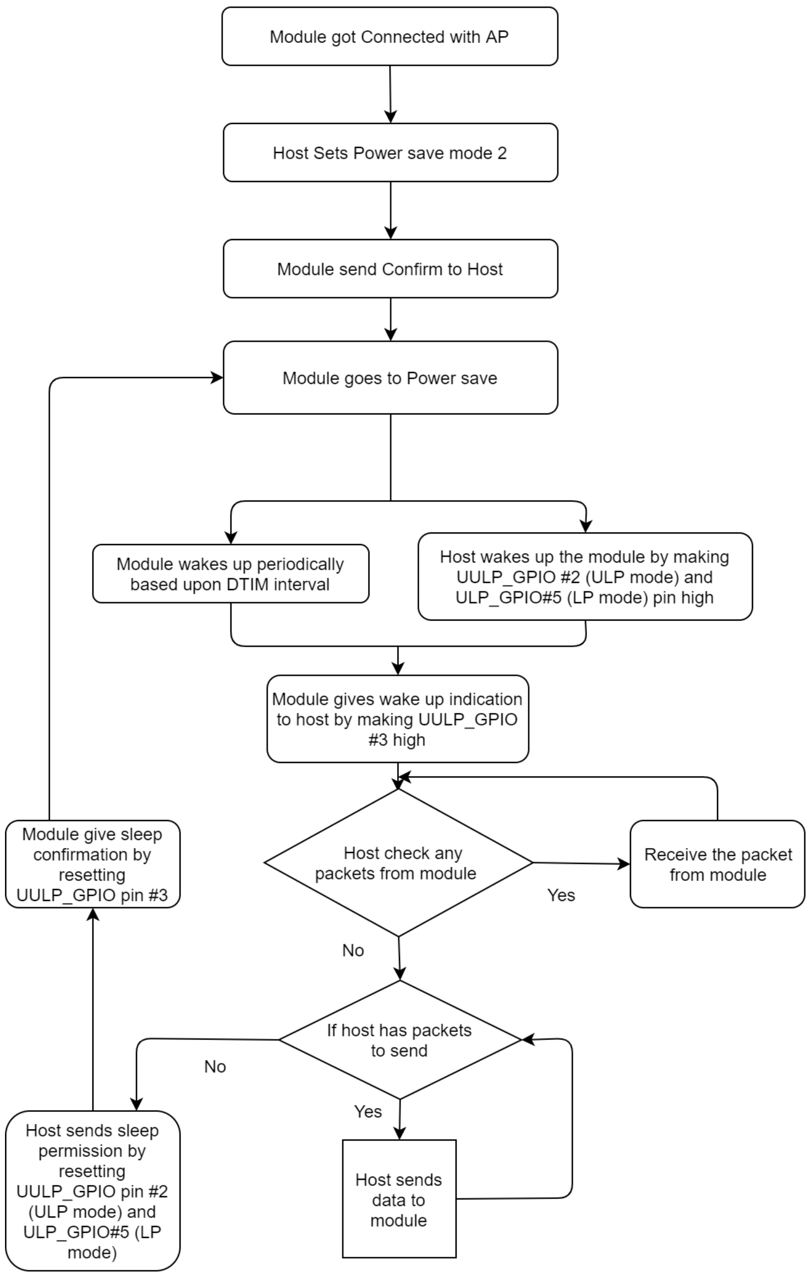

Module got Connected with AP
Host Sets Power save mode 2
Module send Confirm to Host
Module goes to Power save
Module wakes up periodically based upon DTIM interval
Host wakes up the module by making UULP_GPIO #2 (ULP mode) and ULP_GPIO#5 (LP mode) pin high
Module gives wake up indication to host by making UULP_GPIO #3 high
Module give sleep confirmation by resetting UULP_GPIO pin #3
Host check any packets from module
Yes
Receive the packet from module
No
If host has packets to send
No
Yes
Host sends sleep permission by resetting UULP_GPIO pin #2 (ULP mode) and ULP_GPIO#5 (LP mode)
Host sends data to module

Figure 22: Power Save Mode 2

5.16.4 Power Save Mode 3

Power Mode 3 is message based power save. In Power Mode 3 like Power mode 2 both radio and SOC of RS9116-WiSeConnect are in power save mode. This mode is significant when module is in associated state with AP. Module wakes up periodically upon every DTIM and gives wakeup message ("WKP") to host. Module cannot be woken up asynchronously. Every time module intends to go to sleep it sends a sleep request message ("SLP") to the host and expects host to send the ack message. Host either send ack ("ACK") or any other pending message. But once ack is sent, Host should not send any other message unless next wakeup message from module is received.

Module shall not go into complete power-save state if ack is not received from host for given sleep message. Module can send received packets or responses to host at any instant of time. No handshake is required on Rx path.

Table 6: Message from Module in Power Save Mode

AT Mode
"WKP"
"SLP"

Table 7: Message from host in Power Save Mode

AT Mode
"ACK"

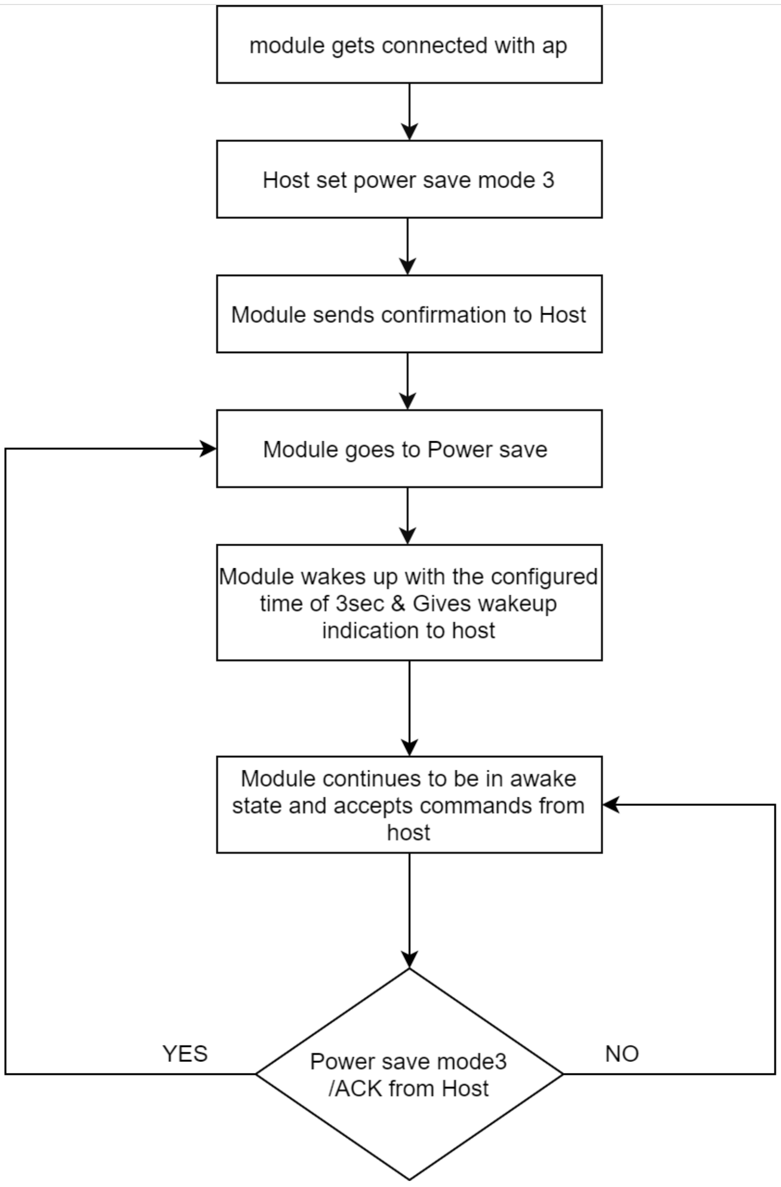

module gets connected with ap
Host set power save mode 3
Module sends confirmation to Host
Module goes to Power save
Module wakes up with the configured time of 3sec & Gives wakeup indication to host
Module continues to be in awake state and accepts commands from host
Power save mode3 /ACK from Host
YES
NO

Figure 23: Power Save Mode 3

5.16.5 Power Save Mode 8

This command must be issued after Init command.
In Power Mode 8 both RF and SOC of RS9116-WiSeConnect are in complete power save mode. This mode is significant when module is not connected with any AP. Power mode 8 is GPIO based. In ULP mode ,

feature_bit_map[4]has to be set in opermode command.
In case of LP (when ulp_mode_enable is '0') host can wakeup the module from power save by making ULP_GPIO_5 high.
In case of ULP (when ulp_mode_enable is '1' or '2') host can wakeup the module from power save by making UULP_GPIO_2 high.
When ulp_mode_enable is set to '0' or '1', once the module gets wakeup it continues to be in wakeup state until it gets power mode 8 commands from host.
When ulp_mode_enable is set to '2', after waking up from sleep module sends following message to host when RAM retention is not enabled. After receiving this message host needs to start giving commands from beginning(opemode) as module's state is not retained.

Table 8: Message from Module in ULP Mode 2

AT Mode
"WKP FRM SLEEP"

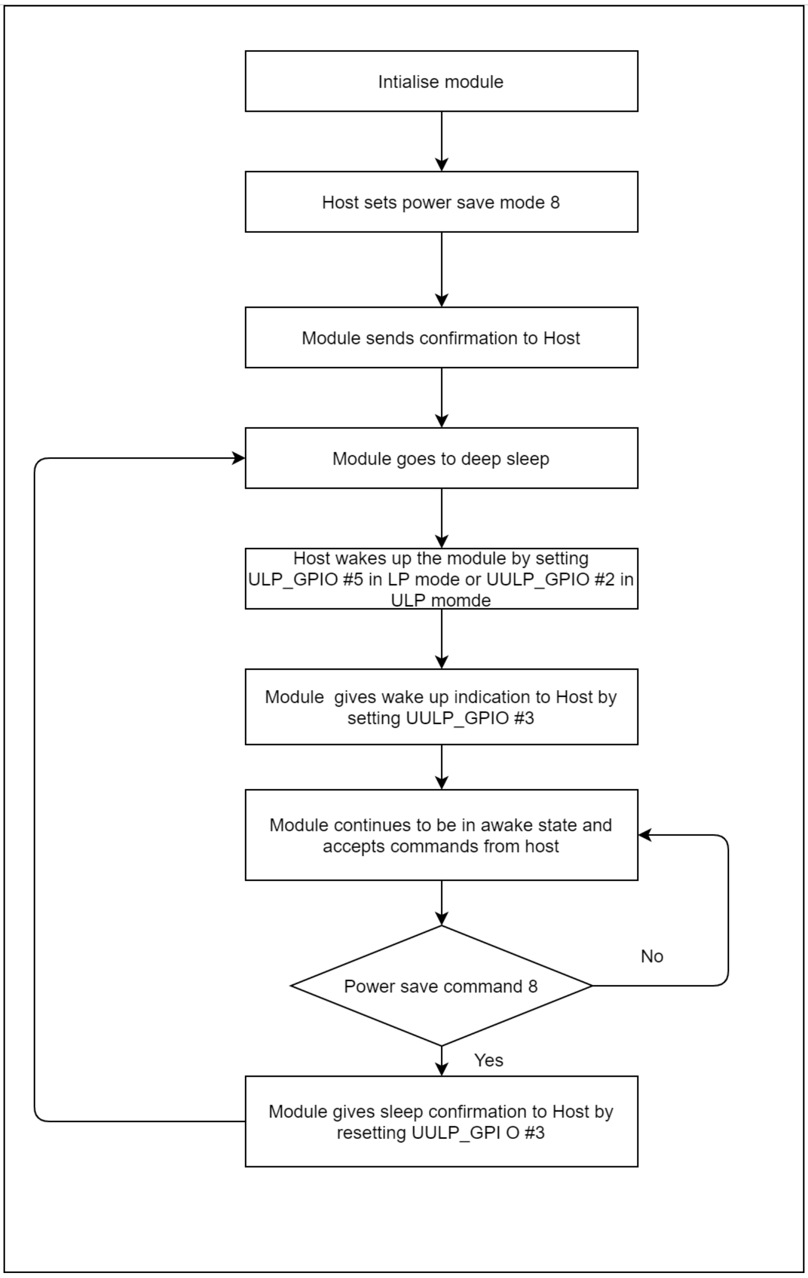

Intialise module
Host sets power save mode 8
Module sends confirmation to Host
Module goes to deep sleep
Host wakes up the module by setting ULP_GPIO #5 in LP mode or UULP_GPIO #2 in ULP momde
Module gives wake up indication to Host by setting UULP_GPIO #3
Module continues to be in awake state and accepts commands from host
Power save command 8
No
Yes
Module gives sleep confirmation to Host by resetting UULP_GPI O #3

Figure 24: Power Save Mode 8

5.16.6 Power Save Mode 9

In Power Mode 9 both Radio and SOC of RS9116-WiSeConnect are in complete power save mode. This mode is significant when module is not connected with any AP. Once power mode 9 command is given, the module goes to sleep immediately and wakes up after sleep duration configurable by host by set sleep timer command. If host does

not set any default time, then the module wakes up in 3sec by default. Upon wakeup module sends a wakeup message to the host and expects host to give ack before it goes into next sleep cycle. Host either send ack or any other messages but once ACK is sent no other packet should be sent before receiving next wakeup message.
When ulp_mode_enable is set to '2', after waking up from sleep, the module sends following message to host when RAM retention is not enabled. After receiving this message, host needs to start giving commands from beginning (opemode) as module's state is not retained.

AT Mode
"WKP FRM SLEEP"

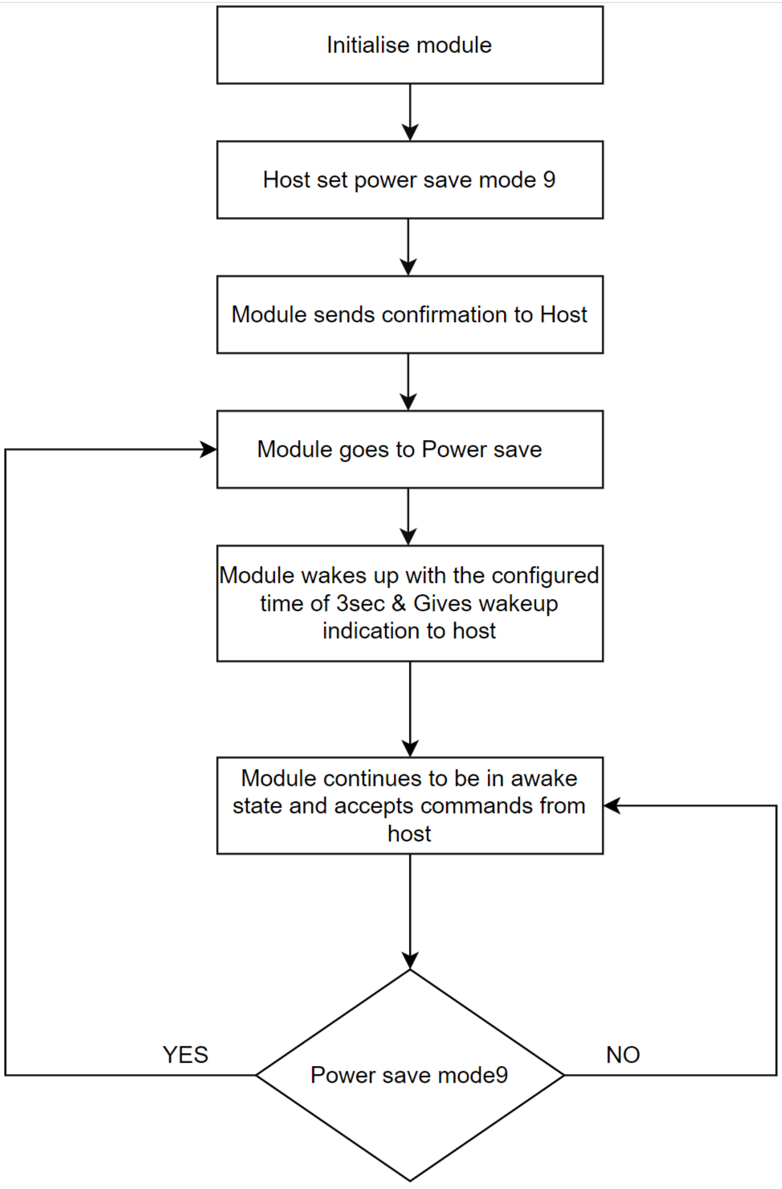

Initialise module
Host set power save mode 9
Module sends confirmation to Host
Module goes to Power save
Module wakes up with the configured time of 3sec & Gives wakeup indication to host
Module continues to be in awake state and accepts commands from host
Power save mode9
YES
NO

Figure 25: Power Save Mode 9

5.17 PSK/PMK

Description:
The command is used to set the PSK (Pre shared key) to join to WPA/WPA2-PSK enabled APs. Using this command user can also pass the PMK (PAIRWISE MASTER KEY) as a parameter and can also generate PMK by providing PSK and SSID of connecting AP.
User can directly give PMK from host to reduce the connection time. This command should be issued after init and before join command, if module needs to connect to an secure Access point. This command can be ignored if the AP is in Open mode.

Command Format:

AT Mode:

at+rsi_psk=<TYPE>,<psk_or_pmk >,<ap_ssid >\r\n

Command Parameters:

TYPE[1 byte]: possible values of this field are 1, 2, 3.
1 - indicate pre_shared_key is provided in psk_or_pmk field,
2 - indicate pairwise_master_key is provided in psk_or_pmk field,
3 - indicate generate pairwise master key from given pre shared key and SSID to which module wants to connect.

Note:

AT command mode TYPE 4 and 5 is added to support ',' in PSK.

TYPE 4 – indicate length based PSK
TYPE 5 – indicate generation of PMK from length based PSK and SSID

To support above type new parameter introduced in the command and command format

for the same is
at+rsi_psk=<TYPE>,<length_of_psk>,<psk_or_pmk >,<ap_ssid >\r\n

where,
length_of_psk (1 byte): gives the length of psk which can includes ',' for example at+rsi_psk=4,10,123,456789

Where,
"123,456789" is psk with 10 bytes length

psk_or_pmk[64 bytes]: In this field expected parameters are pre shared key of the access point to which module wants to associate or pair wise master key. Length of this field is 64 Bytes. In case of PMK only 32 bytes are valid, In case of PSK length can vary (8 to 63).

Note:

PMK is of 32 Bytes. In AT plus command mode, 32 bytes of PMK is given in hex format (64 characters) in the command.

For Example: If PMK is of array, PMK[32] = {0x71, 0x72, 0x01, 0x0A, 0x16, 0x17, 0x07,0x90, 0x71, 0x72, 0x01, 0x0A, 0x16, 0x17, 0x07,0x90, 0x71, 0x72, 0x01, 0x0A, 0x16, 0x17, 0x07,0x90, 0x71, 0x72, 0x01, 0x0A, 0x16, 0x17, 0x07,0x90};, then the command should be given as
at+rsi_psk=2,7172010A161707907172010A161707907172010A161707907172010A16170790\r\n

ap_ssid[32 bytes]: This field contains the SSID of the access point, this field will be valid only if TYPE value is 3.

Note:

If user generates PMK using TYPE 3 (i.e. by providing psk and ssid) then module uses generated PMK for connection establishment and there is no need to give pre shared key or pair wise master key again.

Response:
Response contains following payload only if TYPE value is 3. In case of TYPE 1 & 2 response does not contain any payload.

AT Mode:

Result Code	Description
OK	Successful execution of the command. If TYPE value is '1'or '2'.
OK< pmk >	Successful execution of the command. If TYPE value is '3'.
ERROR<Error code>	Failure

Response Parameters:
Pair wise master key of 32 bytes is given to host if TYPE is 3.

Relevance:
This command is relevant in operating mode 0.

Possible Error Codes:
Possible error codes for this command are 0x0021, 0x0025,0x0026,0x0028,0x002C,0x0039, 0x003a, 0x003b .

Example:

AT Mode:
To join a WPA2-PSK security enabled network with key "12345ABCDE", the command is
at+rsi_psk=1,12345ABCDE\r\n
0x61 0x740x2B0x720x730x690x5F 0x70 0x73 0x6B 0x3D 0x31 0x2c 0x31 0x32 0x33 0x34 0x35 0x41 0x42 0x43 0x44 0x45 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A
To join a WPA2-PSK security enable network with pairwise_master_key "ABCDEFABCDEFABCDEF12345678901234ABCDEFABCDEFABCDEF12345678901234" ,the command is
at+rsi_psk=2, ABCDEFABCDEFABCDEF12345678901234ABCDEFABCDEFABCDEF12345678901234,\r\n

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A
To generate pairwise_master_key for the pre_shared_key "12345678" and SSID "wise_ap", the command is
at+rsi_psk=3,12345678,wise_ap\r\n

Response:
OK<pairwise_master_key>\r\n
0x4F 0x4B <32bytes of pairwise_master_key> 0x0D 0x0A

5.18 Set WEP Keys

Description:
This command configures the WEP key in the module to connect to an AP with WEP security. This command should be issued before join.

Command Format:

AT Mode:
at+rsi_wepkey=< index >,< key1 >,< key2 >,< key3 >,< key4 >\r\n

Command Parameters:

index[2 bytes]: used to select key index configured in AP
0-Key 1 will be used.
1-Key 2 will be used.
2-Key 3 will be used.
3-Key 4 will be used.
Key/Key1/Key2/Key3/Key4: Actual keys. The module supports WEP hex mode only.
The key to be supplied to the AP should be of 10 characters (for 64 bit WEP mode) or 26 characters (for 128 bit WEP mode), and only the following characters are allowed for the key: A,B,C,D,E,F,a,b,c,d,e,f,0,1,2,3,4,5,6,7,8,9

Response:

AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0021, 0x0025, 0x002C,0x002D

Relevance:
This command is relevant when the module is configured in Operating Mode 0.

Example:

AT Mode:
Give write up for the below key entry
at+rsi_wepkey=0,ABCDE12345,ABCDE12346,ABCDE12347, ABCDE12348\r\n
If the user wants to enter only one valid key
at+rsi_wepkey=0,ABCDE12345,0,0,0\r\n
If the user wants to enter only one valid key
at+rsi_wepkey=2,0,0,ABCDE12345,0\r\n

Response:

OK\r\n
0x4F 0x4B 0x0D 0x0A

5.19 Set WEP Authentication Mode

Description:
This command configures the authentication mode for WEP in the module, if the AP is in WEP security mode. This command is supported only in AT Mode.

Command Format:
at+rsi_authmode=auth_mode\r\n

Command Parameters:
auth_mode: set to '0' for open WEP authentication

Note:

WEP shared mode is not supported in RS9116_WC_GENR_0_x_x release.

Response:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure, Possible error codes are 0x0021, 0x0025, 0x002C, 0xFFF8,0x002D

Relevance:

This command is relevant when the module is configured in Operating Mode 0

Example:
at+rsi_authmode=0\r\n
0x61 0x74 0x2B0x72 0x73 0x69 0x5F 0x61 0x75 0x74 0x68 0x6D 0x6F 0x64 0x65 0x3D 0x30 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.20 Set EAP Configuration

Description:
This command is used to configure the EAP parameters for connecting to an Enterprise Security enabled Access Point. The supported EAP types are EAP-TLS, EAP-TTLS, EAP-PEAP,EAP-FAST and EAP-LEAP.

Note:

EAP-GTC is not supported for EAP-FAST.
This command can be sent any time after at+rsi_init and before at+rsi_join in enterprise security mode.

Command Format:
AT Mode:
at+rsi_eap =< eapMethod >,< innerMethod >,< userIdentity >,< password >,< okc >,<private_key_password>\r\n

Command Parameters:

eapMethod[32 bytes]: Should be one of among TLS, TTLS, FAST, PEAP or LEAP. It should be ASCII character string.

innerMethod[32 bytes]: This field is valid only in TTLS/PEAP. In case of TTLS/PEAP supported inner methods are MSCHAP/MSCHAPV2. In case of TLS/FAST/LEAP this field is not valid, and it should be fixed to MSCHAPV2.

Here MSCHAP/MSCHAPV2 are ASCII character strings.

userIdentity[64 bytes]: User ID which is configured in the user configuration file of the radius sever.

Password[128 bytes]: Password which is configured in the user configuration file of the Radius Server for that User Identity.

Okc[4 bytes]: To enable or disable opportunistic key caching(OKC)
0 – disable
1 – enable

When this is enabled, module will use cached PMKID to get MSK (Master Session Key) which is need for generating PMK which is needed for 4-way handshake.

private_key_password[82 bytes]: This is password for encrypted private key given to the module. Module will use this password during decryption of encrypted private key.

Password length should not be more than 80 bytes

Response:

AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure,

Possible error codes:

Possible error codes are 0x0021, 0x0025, 0x002C.

Relevance:
This command is relevant when the module is configured in Operating Mode 2.

5.21 Set Certificate

Description:
This command is used to load/erase SSL (certificate and private keys) and enterprise security (EAP-TLS or EAP-FAST) certificates. Certificates should be loaded before using SSL/EAP. This command should be sent before join command for enterprise security mode and before socket creation for SSL sockets. Certificates will be loaded in non-volatile memory of the module, so that loading certificate is required to be done only once.

Note:

This command should be sent only after opermode command.

Command Format:

AT Mode:
at+rsi_cert =< CertType >,< total_len >,< KeyPwd >,< Certificate >\r\n

Command Parameters:

total_len[2 bytes]: Certificate's total length in bytes.

Note:

1. For Enterprise security, maximum cert_len should be less than 12280 bytes. For SSL Certificates, the max length is 12280 bytes and for the Private Keys, it is 4088 bytes.
2. Module shares same SSL certificates for all supported SSL sockets.
3. For enterprise, user can load certificates in two ways
   - User can provide wifiuser.pem which contains 4 certificates in a given fixed order of private key, client certificate 1, client certificate 2, CA certificate with CertType as 1.
   - User can load individual EAP certificates private key, public key, and CA certificates with CertType as 17,33 and 49 respectively. Maximum certificate length for each individual certificate is 4088 bytes
4. Certificate Loading into Flash is only allowed upto init state. After init state e.g. scan/join etc. certificate loading into flash is not allowed.
5. Certificate Loading into RAM is allowed in connected state also provided same type of socket already does not exist. For example, loading client cert is only allowed if there does not exist any client socket. Same is for server certificates too

Note:

Recommended to use loading of single certificate method (wifiuser.pem)

Note:

By default, SSL certificates will be loaded onto flash. Set BIT(27) in tcp_ip_feature_bit_map to load SSl certificate onto RAM.

cert_type[1 byte]	**Type of certificate**
1	EAP client certificate
2	FAST PAC file
3	SSL Client Certificate
4	SSL Client Private Key
5	SSL CA Certificate
6	SSL Server Certificate
7	SSL Server Private Key
17	EAP private key
33	EAP public key
49	EAP CA certificate

In case of AT mode Host need to send whole certificate at a time.

keyPwd(128 bytes): Reserved.

certificate: This is the data of the actual certificate.

Certificate erase:
For erasing certificate,
total_len, KeyPwd, Certificate fields should be set to '0' in AT Mode.
cert_type should be set to type of the certificate to erase as mentioned above.

Response:

AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0015,0x0021, 0x0025,0x0026, 0x002C.

Relevance:
This command is relevant when the module is configured in Operating Mode 0,2

Example:

AT Mode:
It may not be possible to issue this command in Hyper-terminal because the content of a certificate file needs to be supplied as one of the inputs of the command. This can be done by other means, such as using a Python script. A sample Python excerpt is shown below, where wifiuser.pem is the names of the certificate file:

```
def set_cert():

print "Set certificate\n"

f3 = open('e:\\certificates\wifiuser.pem', 'r+')

str = f3.read()

num =len (str)

print 'Certificate len', num

out='at+rsi_cert=1,6522,password,'str'\r\n'

print 'Given command'

sp.write(out)
```

For loading certificate:

at+rsi_cert=1,10,12345678,@$cd5%ghij\r\n

0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x63 0x65 0x72 0x74 0x3D 0x31 0x2C 0x31 0x30 0x2C 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x2C 0x40 0x24 0x63 0x64 0x35 0x25 0x67 0x68 0x69 0x6A 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A
For erasing certificate:
at+rsi_cert=1,0,0,0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x63 0x65 0x72 0x74 0x3D 0x31 0x2C 0x30 0x2C 0x30 0x2C 0x30 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.22 Set Certificate with Indices

Description:
This command is used to load/erase SSL (certificate and private keys) and enterprise security (EAP-TLS or EAP-FAST) certificates. Certificates should be loaded before using SSL/EAP. This command should be sent before join command for enterprise security mode and before socket creation for SSL sockets. Certificates will be loaded in non-volatile memory of the module, so that loading certificate is required to be done only once.

Note:

This command is used for WI-FI alone mode in case of loading 3 SSL certificates to FLASH.

Note:

This command should be sent only after opermode command.

Command Format:
AT Mode:
at+rsi_cert =< CertType >,< total_len >,<CertInx>,< KeyPwd >,< Certificate >\r\n

Command Parameters:
total_len[2 bytes]: Certificate's total length in bytes.

Note:

1. For Enterprise security, maximum cert_len should be less than 12280 bytes. For SSL Certificates, the max length is 12280 bytes and for the Private Keys, it is 4088 bytes.
2. Module shares same SSL certificates for all supported SSL sockets.
3. For enterprise, user can load certificates in two ways

   - User can provide wifiuser.pem which contains 4 certificates in a given fixed order of private key, client certificate 1, client certificate 2, CA certificate with CertType as 1.

   - User can load individual EAP certificates private key, public key, and CA certificates with CertType as 17,33 and 49 respectively. Maximum certificate length for each individual certificate is 4088 bytes.
4. Certificate Loading into Flash is only allowed upto init state. After init state e.g. scan/join etc certificate loading into flash is not allowed.
5. Certificate Loading into RAM is allowed in connected state also provided same type of socket already does not exist. For example, loading client cert is only allowed if there does not exist any client socket. Same is for server certificates too.

Note:

Recommended to use loading of single certificate method (wifiuser.pem)

Note:

1. Set BIT(27) in tcp_ip_feature_bit_map to load SSl certificate onto RAM. By default, SSL certificates will be loaded onto flash.
2. Set BIT(31) in tcp_ip_feature_bit_map & BIT(29) in ext_tcp_ip_feature_bit_map to open 3 SSL Client sockets.

cert_type[1 byte]	Type of certificate
1	EAP Client certificate
2	FAST PAC file
3	SSL Client Certificate

cert_type[1 byte]	Type of certificate
4	SSL Client Private Key
5	SSL CA Certificate
6	SSL Server Certificate
7	SSL Server Private Key
17	EAP private key
33	EAP public key
49	EAP CA certificate

In case of AT mode Host need to send whole certificate at a time.

CertInx (1 byte): 1. Module can hold two sets of SSL certificate into RAM. This field is used to provide the index of the certificates and possible values are 0 and 1.

2. Module can hold three sets of SSL certificate onto Flash. This field is used to provide the index of the certificates and possible values are 0, 1 and 2.

Note:
Currently RS9116 can either load certificates onto RAM or FLASH, but not both at a time.

keyPwd(127 bytes): Reserved.

certificate: This is the data of the actual certificate.

Certificate erase:
For erasing certificate,

total_len, KeyPwd , Certificate fields should be set to '0' in AT Mode.
cert_type should be set to type of the certificate to erase as mentioned above

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0015,0x0021, 0x0025,0x0026, 0x002C, 0x005D, 0x005E, 0x005F.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 2.

Example:

AT Mode:
It may not be possible to issue this command in Hyper-terminal, because the content of a certificate file needs to be supplied as one of the inputs of the command. This can be done by other means, such as using a Python script. A sample Python excerpt is shown below, where wifiuser.pem is the name of the certificate file to load certificate with index 0:

```
def set_cert():
print "Set certificate\n"
f3 = open('e:\\certificates\wifiuser.pem', 'r+')
str = f3.read()
num =len (str)
print 'Certificate len', num
out='at+rsi_cert_inx=1,6522,0,password,'str'\r\n'
```

print 'Given command'
sp.write(out)

For loading certificate with index 0:
at+rsi_cert_inx=1,10,0,12345678,@$cd5%ghij\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x63 0x65 0x72 0x74 0x5F 0x69 0x6E 0x78 0x3D 0x31 0x2C 0x31 0x30 0x2C 0x30 0x2C 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x2C 0x40 0x24 0x63 0x64 0x35 0x25 0x67 0x68 0x69 0x6A 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A
For erasing certificate with index 0:
at+rsi_cert_inx=1,0,0,0,0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x63 0x65 0x72 0x74 x5F 0x69 0x6E 0x78 0x3D 0x31 0x2C 0x30 0x2C 0x30 0x2C 0x30 0x2C 0x30 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.23 Disassociate

Description:
This command is issued to request the module to disassociate (disconnect) from an Access Point. The Host can then issue a fresh set of Init, Scan and Join commands to connect to a different Access Point or the same Access Point with a different set of connection parameters. This command can also be used to stop the module from continuing an on-going rejoin operation. Additionally, this command is used when the module is in AP mode, to remove clients from its list of connected nodes.

Command Format:
AT Mode:
at+rsi_disassoc=< mode_flag >,< client_mac_addr >\r\n

Command Parameters:

mode_flag[2 bytes]:

Mode	Functionality
0	Module is in client mode. The second parameter mac_addr is ignored when mode is 0.
1	Module is in AP mode.

client_mac_addr[6 bytes]: MAC address of the client to disconnect. Used when the module is in AP mode

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:
Possible error codes are 0x0006, 0x0013, 0x0021, 0x002C, 0x0015.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

Note:

After issuing disconnect command, any power save enabled by that time will be disabled. User can reissue the power save command after initializing the module again.

Example:

AT Mode:
Module is in client mode and is connected to an AP. It wants to formally disconnect from the AP.
at+rsi_disassoc=0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x64 0x69 0x73 0x61 0x73 0x73 0x6F 0x63 0x3D 0x30 0x0D 0x0A
Module is in AP mode and 3 clients are connected to it. One of the clients, with MAC 0x01 0x02 0x03 0x040 0x05 0x06 , needs to be disconnected by the AP.
at+rsi_disassoc=1,010203040506\r\n
0x61 0x740x2B0x720x730x690x5F 0x64 0x69 0x73 0x61 0x73 0x73 0x6F 0x63 0x3D 0x31 0x2C 0x30 0x31 0x30 0x32 0x30 0x33 0x30 0x34 0x30 0x35 0x30 0x36 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.24 Set IP Parameters

Description:
This command configures the IP address, subnet mask and default gateway for the module.

Command Format:

AT Mode:
at+rsi_ipconf=< dhcpMode >, < ipaddr >,< netmask >,< gateway >, < hostname >, <vap id>,< fqdnFlag>\r\n

Command Parameters:

dhcpMode[1 byte]: Used to configure TCP/IP stack in manual or DHCP modes.
0– Manual
1– DHCP enabled
3 - To enable DHCP and to send host name in DHCP discover

5 - To enable Option 71 with static IP

Note:

In AP mode only DHCP manual mode (static IP) is valid. sending host name is valid only in DHCP enable mode.

4 -To enable FQDN Option with static ip (Option 81 with static IP)
7 -To enable DHCP, hostname, DHCP Client FQDN option (Option 81 with Dynamic IP)
9 - To support DHCP unicast Offer from server

ipAddr[4 bytes]: IP address in 4 bytes hex format. This can be 0's in the case of DHCP.

netmask[4 bytes]: Subnet mask in 4 bytes hex format. This can be 0's in the case of DHCP.

Gateway[4 bytes]: Gateway in 4 bytes hex format. This can be 0's in the case of DHCP.

hostname[31 bytes]: Host name for DHCP Client. This can be null, when DHCP mode is not enabled. This field is valid only when dhcpMode value is 3. Maximum Hostname length is valid up to 31 bytes including NULL.

Vap_id[1 byte]: Possible values are 0 and 1.
When 1 - Used start DHCP server in concurrent AP mode (should be given before AP creation i.e. join).
When 0 – Used to assign static IP for client mode.

Note:

vap_id will be considered only in concurrent mode and when dhcp mode is manual.

fqdnFlag[4 b ytes]:
0 - DNS Client should update TYPE_A(host name) record and TYPE_PTR records.
1 - DHCP Server will update both TYPE_A and TYPE_PTR records

Response:
AT Mode:

Result Code	Description
OK< macAddr >< ipaddr >< netmask >< gateway >	Successful execution of the command
ERROR<Error code>	Failure

Response Parameters:

macAddr(6 bytes): MAC Address
ipAddr(4 bytes) : Assigned IP address
netmask(4 bytes): Assigned subnet address
gateway(4 bytes): Assigned gateway address

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C, 0xFFFC,0xFF74, 0xFF9C, 0xFF9D.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

Example:
AT Mode:
To configure in manual mode, with 192.168.1.3, 255.255.255.0 and 192.168.1.1 as the IP address, subnet mask and gateway the command is
at+rsi_ipconf=0,192.168.1.3,255.255.255.0,192.168.1.1\r\n
0x610x740x2B0x720x730x690x5F0x690x700x63
0x6F0x6E0x660x3D 0x300x2C0x310x390x320x2E
0x310x360x380x2E0x310x2E0x330x2C 0x320x35
0x350x2E0x320x350x350x2E0x320x350x350x2E
0x300x2C 0x310x390x320x2E0x310x360x380x2E
0x310x2E0x310x0D 0x0A

Response:
OK<MAC_Address><IP_Address><Subnet_Mask><Gateway>\r\n
0x4F 0x4B 0x01 0x02 0x03 0x04 0x05 0x06 0xC0 0xA8 0x01 0x03 0xFF 0xFF 0xFF 0x00 0xC0 0xA8 0x01 0x01 0x0D0x0A
To configure the IP in DHCP enabled mode, the command is
at+rsi_ipconf=1,0,0,0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x69 0x70 0x63 0x6F 0x6E 0x66 0x3D 0x31 0x2C 0x30 0x2C 0x30 0x2C 0x30 0x0D 0x0A

Response:
OK<MAC_Address><IP_Address><Subnet_Mask><Gateway>\r\n
0x4F 0x4B 0x01 0x02 0x03 0x04 0x05 0x06 0xC0 0xA8 0x01 0x03 0xFF0xFF0xFF0x000xC00xA80x01 0x01 0x0D 0x0A
To configure the IP in DHCP enabled mode with hostname, the command is
at+rsi_ipconf=3,dhcp_client\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x69 0x70 0x63 0x6F 0x6E 0x66 0x3D 0x31 0x2C 0x30 0x2C 0x30 0x2C 0x30 0x0D 0x0A

Response:
OK<MAC_Address><IP_Address><Subnet_Mask><Gateway>\r\n
0x4F 0x4B 0x01 0x02 0x03 0x04 0x05 0x06 0xC0 0xA8 0x01 0x03 0xFF0xFF0xFF0x000xC00xA80x01 0x01 0x0D 0x0A

5.25 IP Change Notification

Description:
This notification is received when module gets a different IP compared to modules old IP address, after DHCP renewal. Module indicates this IP change to host by an asynchronous frame.

Command Format:
N/A

Command Parameters:
N/A

Response:
AT Mode:

Result Code	Description
AT+RSI_IPCONF< macAddr >< ipaddr >< netmask >< gateway >	

Response Parameters:

macAddr(6 bytes): MAC Address
ipAddr(4 bytes): Assigned IP address
netmask(4 bytes): Assigned subnet address
gateway(4 bytes): Assigned gateway address

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

Example:
AT Mode:
AT+RSI_IPCONF< macAddr >< ipaddr >< netmask >< gateway >\r\n
0x41 0x54 0x2B 0x52 0x53 0x49 0x5F 0x49 0x50 0x43 0x4F 0x4E 0x46 0x3D 0x01 0x02 0x03 0x04 0x05 0x06 0xC0 0xA8 0x01 0x03 0xFF0xFF0xFF0x000xC00xA8
0x01 0x01 0x0D 0x0A

5.26 Set IPv6 Parameters

Description:
This command configures the IPv6 address, prefix length and default router for the module.

Command Format:
AT Mode:
at+rsi_ipconf6=< mode >,< prefixLength >,< ipaddr6 >, < gateway6 >\r\n

Command Parameters:

mode[2 bytes]:
Used to configure TCP/IP stack in manual or DHCPv6 modes.
0–Manual
1–DHCPv6

prefixLength[2 bytes]: Prefix length of the IPv6 address.

ipaddr6[16 bytes]: IPv6 address. This can be 0's in the case of DHCPv6.

gateway6[16 bytes]: Default router's IPv6 address. This should be Null in the case of DHCPv6.

IPV6 address is generally of the form - octet of four hexadecimal digits separated by "colons".
e.g. 2001:0db8:1:0:0:0:0:123 (supported format)
2001:db8:1::123 (not supported format)

Double colons are used in place of continuous zeroes in IPV6 address, to minimize the IPV6 address, but in RS9116-WiSeConnect double colons are not supported. In ipconf6 command you have to supply all the eight groups of four hexadecimal digits.

Response:
AT Mode:

Result Code	Description
OK< prefixLength >< ipaddr6>< defaultgw6>	Successful execution of the command.
ERROR<Error code>	Failure

Response Parameters:
prefixLength(2 bytes): Prefix length of IPv6 address
ipaddr6(16 bytes): Assigned IPv6 address
defaultgw6(16 bytes): Assigned default_router address

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C, 0xFFFC, 0xFF9C,0xFF74, 0xFF9D.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

Example:
AT Mode:
To configure in manual mode, with 2001:db8:1:0:0:0:0:123 as the IPV6 address and with 2001:db8:1:0:0:0:0:100 as router IPV6 address, the command is :
at+rsi_ipconf6=0,64,2001:DB8:1:0:0:0:0:123,2001:DB8:1:0:0:0:0:100\r\n
0x610x740x2B0x720x730x690x5F0x690x700x63 0x6F0x6E0x660x36 0x3D 0x300x2C0x360x340x32 0x300x30 0x31 0x3A0x440x420x380x3A0x310x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x31 0x32 0x330x2C0x32 0x300x30 0x31 0x3A0x440x420x38 0x3A0x310x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x31 0x00 0x000x0D 0x0A

Response:
OK< prefixLength >< ipaddr6 >< gateway6>\r\n
0x4F 0x4B 0x40 0x00 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x23 0x01 0x00 0x00 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x0D 0x0A
To configure the IPV6 in DHCPV6 enabled mode, the command is
at+rsi_ipconf6=1\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x69 0x70 0x63 0x6F 0x6E 0x66 0x36 0x3D 0x31 0x0D 0x0A

Response:
OK< prefixLength >< ipaddr6 >< gateway6>\r\n
0x4F 0x4B 0x40 0x00 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x36 0x01 0x00 0x00 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x0D 0x0A
The IPv6 address assigned to module by DHCPv6 server is 2001:DB8:1:0:0:0:0:136, default prefix 64 is used and router ipv6 address is 2001:DB8:1:0:0:0:0:100.

5.27 SNMP

Description:
This command configures the SNMP agent in the module. This command can be issued only after set IP parameters or set IPv6 parameters command. This is the first command for using SNMP feature

Command Format:

AT Mode:
at+rsi_snmp_enable=< snmpEnable >\r\n

Command Parameters:
snmpEnable[1 byte]:
To enable SNMP agent in module
0 - SNMP disable
1 - SNMP enable

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command.
ERROR<Error code>	Failure

Response Parameters:

N/A

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C,0xFF74,0x0015,0x100,0xFF82.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

Example:
AT Mode:
To enable SNMP in module the command is
at+rsi_snmp_enable=1\r\n

0x610x740x2B0x720x730x690x5F0x73 0x6E 0x6D 0x70 0x5F 0x65 0x6E 0x61 0x62 0x6C 0x65 0x3D 0x31 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.28 SNMP Set

Description:
This is an asynchronous message which module gives to host whenever it receives snmp set message from the remote SNMP server, to set the value corresponding to the object id. This message can only be received when module is configured as an SNMP agent.

Command Format:

AT Mode:
N/A

Command Parameters:
N/A

Response:
AT Mode:

Result Code	Description
AT_RSI_SNMP_SET=< object_id >,< length >,< value >	Asynchronous message sent by remote snmp server and received by the module.

Response Parameters:

object_id (128 bytes): object id for which the snmp server wants to set.
length (4bytes): length of value of object id.
LSB is returned first.
value (200bytes): value of object id to be set. Out of these 200 bytes, user has to consider only length number of bytes.

Note:

Value contains maximum of 144 bytes in case of IPV6.

Possible error codes:
N/A

Relevance:
This message is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

Example:

AT Mode:

AT+RSI_SNMP_SET=1.3.6.1.2.1.1.1.0,4,home\r\n
0x41 0x54 0x2B 0x52 0x53 0x49 0x5F 0x53 0x4E 0x4D 0x50 0x5F 0x53 0x45 0x54 0x3D 0x31 0x2E 0x33 0x2E 0x36 0x2E 0x31 0x2E 0x32 0x2E 0x31 0x2E 0X31 0x2E 0x31 0x2E 0x30 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0x2C 0x00 0x00 0x00 0x04 0x2C 0X68 0X6F 0X6D 0X65 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00

0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X00 0X0D 0X0A

5.29 SNMP Get Response

Description:
This command is given in response to the get request received by the SNMP agent (module) and issued by the SNMP server. Whenever snmp server sends a snmp get request ,module indicates this to host by an asynchronous message. Then module has to issue this command for sending response to corresponding received snmp get request.

Command Format:
AT Mode:
at+rsi_snmp_get_rsp=< type >,< value >,< OID >\r\n

Command Parameters:

type[1 byte]: type of object requested.

Table 9 SNMP Object Types and Codes

SNMP Object Type	Object code
SNMP_ANS1_COUNTER	0x41
SNMP_ANS1_COUNTER64	0x46
SNMP_ANS1_END_OF_MIB_VIEW	0x82
SNMP_ANS1_GAUGE	0x42
SNMP_ANS1_OBJECT_ID	0x6
SNMP_ANS1_INTEGER	0x2
SNMP_ANS1_IP_ADDRESS	0x40
SNMP_ANS1_IPV6_ADDRESS	0x44
SNMP_ANS1_NO_SUCH_INSTANCE	0x81
SNMP_ANS1_NO_SUCH_OBJECT	0x80
SNMP_ANS1_OCTET_STRING	0x4
SNMP_ANS1_TIME_TICS	0x43

value[200 bytes]: value passed in response corresponding to the type of object requested.

Note:

1. Size of the value should be 200 bytes. For example, if the value is “Silicon” (length of the value is 7 bytes) then remaining 193 bytes should be filled with NULL.
2. OID should be given followed by value, without having any comma (,) separator between value and OID.
3. If sending data type SNMP_ANS1_COUNTER64 is more than 4bytes (1 word = 4bytes), then each word should be reversed (as it is reversed in case of counter 32) in hex form and sent. Only then expected response is received in server side. e.g.- 1122334455667788 sends it as 4433221188776655

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command.

Result Code	Description
ERROR<Error code>	Failure

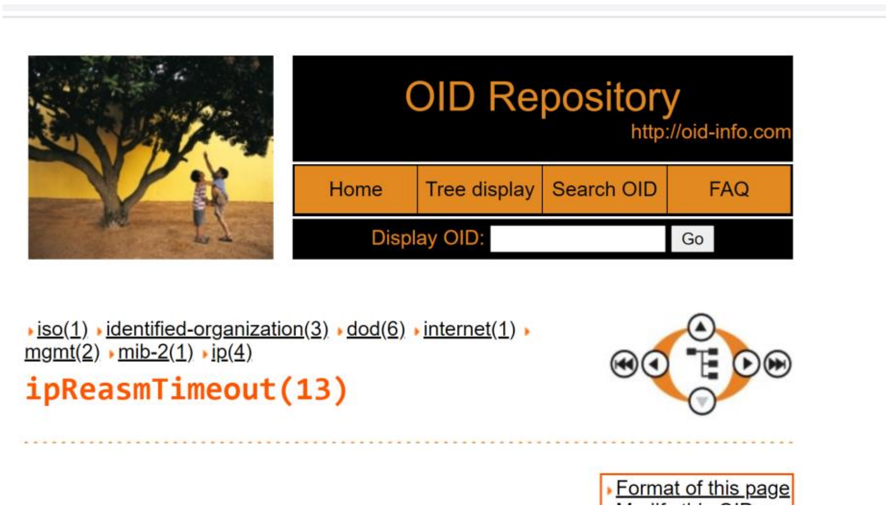
Response Parameters:
N/A

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

To get the Data types of individual OID's, should follow the path **http://www.oid-info.com/**

Copy the OID except for last digit, paste it on **Display OID. I**t gives Data type, description and information about OID's.



The list of example OIDs are:

OID	Data Type	Description
1.3.6.1.2.1.1.1.0	String	A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software.
1.3.6.1.2.1.4.3.0	Counter32	The total number of input datagrams received from interfaces, including those received in error.
1.3.6.1.2.1.4.4.0	Counter32	The number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc..
1.3.6.1.2.1.4.5.0	Counter32	The number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E). For entities which are not IP routers and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address..
1.3.6.1.2.1.4.6.0	Counter32	The number of input datagrams for which this entity was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination. In entities which do not act as IP routers, this counter will include only those packets which were Source- Routed via this entity, and the Source-Route option processing was successful.

OID	Data Type	Description
1.3.6.1.2.1.4.7.0	Counter32	The number of locally addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.
1.3.6.1.2.1.4.9.0	Counter32	The total number of input datagrams successfully delivered to IP user-protocols (including ICMP).
1.3.6.1.2.1.4.10.0	Counter32	The total number of IP datagrams which local IP user- protocols (including ICMP) supplied to IP in requests for transmission. Note that this counter does not include any datagrams counted in ipForwDatagrams..
1.3.6.1.2.1.4.11.0	Counter32	The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer space). Note that this counter would include datagrams counted in ipForwDatagrams if any such packets met this (discretionary) discard criterion.
1.3.6.1.2.1.4.12.0	Counter32	The number of IP datagrams discarded because no route could be found to transmit them to their destination. Note that this counter includes any packets counted in ipForwDatagrams which meet this `no-route' criterion. Note that this includes any datagrams which a host cannot route because all of its default routers are down.
1.3.6.1.2.1.4.13.0	Integer	The maximum number of seconds that received fragments are held while they are awaiting reassembly at this entity.
1.3.6.1.2.1.4.14.0	Counter32	Number of Internet Protocol (IP) fragments received which needed to be reassembled at this entity.
1.3.6.1.2.1.4.15.0	Counter32	Number of Internet Protocol (IP) datagrams successfully re-assembled.
1.3.6.1.2.1.4.16.0	Counter32	Number of failures detected by the IP reassembly algorithm (for whatever reason: timed out, errors, etc.). Note that this is not necessarily a count of discarded IP fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.
1.3.6.1. 2.1.4.17.0	Counter32	Number of Internet Protocol (IP) datagrams that have been successfully fragmented at this entity.
1.3.6.1.2.1.4.18.0	Counter32	The number of Internet Protocol (IP) datagrams that have been discarded because they needed to be fragmented at this entity but could not be, e.g. because their Don't Fragment flag was set.
1.3.6.1.2.1.4.19.0	Counter32	Number of Internet Protocol (IP) datagram fragments that have been generated as a result of fragmentation at this entity.
1.3.6.1.2.1.5.1.0	Counter32	The total number of ICMP messages which the entity received. Note that this counter includes all those counted by icmpInErrors.
1.3.6.1.2.1.5.2.0	Counter32	The number of ICMP messages which the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.).
1.3.6.1.2.1.5.8.0	Counter32	The number of ICMP Echo (request) messages received.
1.3.6.1.2.1.5.9.0	Counter32	The number of ICMP Echo Reply messages received.
1.3.6.1.2.1.5.14.0	Counter32	The total number of ICMP messages which this entity attempted to send. Note that this counter includes all those counted by icmpOutErrors.
1.3.6.1.2.1.5.15.0	Counter32	Number of ICMP messages which this entity did not send due to problems discovered within ICMP such as a lack of buffers. This value should not include errors discovered outside the ICMP layer such as the inability of IP to route the resultant datagram. In some implementations there may be no types of error which contribute to this counter's value.
1.3.6.1.2.1.5.21.0	Counter32	The number of ICMP Echo (request) messages sent.
1.3.6.1.2.1.5.22.0	Counter32	The number of ICMP Echo Reply messages sent.

OID	Data Type	Description
1.3.6.1.2.1.6.5.0	Counter32	The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.
1.3.6.1.2.1.6.6.0	Counter32	The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.
1.3.6.1.2.1.6.7.0	Counter32	The number of times TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.
1.3.6.1.2.1.6.8.0	Counter32	The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.
1.3.6.1.2.1.6.10.0	Counter32	The total number of segments received, including those received in error. This count includes segments received on currently established connections.
1.3.6.1.2.1.6.11.0	Counter32	The total number of segments sent, including those on current connections but excluding those containing only retransmitted octets.
1.3.6.1. 2.1.6.12.0	Counter32	The total number of segments retransmitted - that is, the number of TCP segments transmitted containing one or more previously transmitted octets.
1.3.6.1.2.1.6.14.0	Counter32	The total number of segments received in error (e.g., bad TCP checksums).
1.3.6.1. 2.1.6.15.0	Counter32	The number of TCP segments sent containing the RST flag.
1.3.6.1.2.1.7.1.0	Counter32	The total number of UDP datagrams delivered to UDP users.
1.3.6.1.2.1.7.4.0	Counter32	The total number of UDP datagrams sent from this entity.
1.3.6.1.2.1.7.3.0	Counter32	The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

Example:

AT Mode:
To respond for string type of SNMP object requested by the remote SNMP server "AT+RSI_SNMP_GET=1.3.6.1.2.1.1.1.0" the command is :
at+rsi_snmp_get_rsp=4,abcd\r\n
0x61 0x74 0x2B 0x72 0x730x69 0x5F 0x73 0x6E 0x6D 0x70 0x5F 0x67 0x65 0x74 0x5F 0x72 0x73 0x70 0x3D 0x34 0x2C 0x61 0x62 0x63 0x64 0x0D 0x0A
Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

For, OID of type INTEGER, COUNTER, COUNTER64, GAUGE, TIMER_TICKS input shall be given in hexa-equivalent form.
To respond for INTEGER type of SNMP object requested by the remote SNMP server “AT+RSI_SNMP_GET=1.3.6.1.2.1.1.3.0” the command is :

for Integer value 6:
HEX EQUIVALENT:
61 74 2B 72 73 69 5F 73 6E 6D 70 5F 67 65 74 5F 72 73 70 3D 32 2C 06 0D 0A

For Integer value 10:
HEX EQUIVALENT:
61 74 2B 72 73 69 5F 73 6E 6D 70 5F 67 65 74 5F 72 73 70 3D 32 2C 0A 0D 0A

For Integer value 276:
HEX EQUIVALENT:
61 74 2B 72 73 69 5F 73 6E 6D 70 5F 67 65 74 5F 72 73 70 3D 32 2C 14 01 0D 0A

For, OID of type IP_ADDRESS, if the response ip-address want to give is 192.168.10.163 the command shall be given in HEX form as shown below:
0x61 0x74 0x2b 0x72 0x73 0x69 0x5f 0x73 0x6e 0x6d 0x70 0x5f

0x67 0x65 0x74 0x5f 0x72 0x73 0x70 0x3d 0x36 0x34 0x2c 0xa3
0x0a 0xa8 0xc0 0x0d 0x0a
For, OID of type IPV6_ADDRESS, if the response ipv6-address want to give is 2001:db8:0:1:0:0:0:123

5.30 SNMP Get Next Response

Description:
This command is given in response to the get request received by the SNMP agent (module) and issued by the SNMP server. Whenever snmp server sends a snmp get_next request, module indicates this to host by an asynchronous message. Then module has to issue this command for sending response to corresponding received snmp get next request.

Command Format:
AT Mode:
at+rsi_snmp_getnext_rsp=< type >,< value >,<next objid in MIB table>\r\n

Command Parameters:
type[1 byte] : type of object requested.

Table 10 SNMP Object Types and Codes

SNMP Object Type	Object code
SNMP_ANS1_COUNTER	0x41
SNMP_ANS1_COUNTER64	0x46
SNMP_ANS1_END_OF_MIB_VIEW	0x82
SNMP_ANS1_GAUGE	0x42
SNMP_ANS1_OBJECT_ID	0x6
SNMP_ANS1_INTEGER	0x2
SNMP_ANS1_IP_ADDRESS	0x40
SNMP_ANS1_IPV6_ADDRESS	0x44
SNMP_ANS1_NO_SUCH_INSTANCE	0x81
SNMP_ANS1_NO_SUCH_OBJECT	0x80
SNMP_ANS1_OCTET_STRING	0x4
SNMP_ANS1_TIME_TICS	0x43

value[200 bytes]: value passed in response corresponding to the type of object requested.

Note:

1. Size of the value should be 200 bytes. For example, if the value is "Silicon" (length of the value is 7 bytes) then remaining 193 bytes should be filled with NULL.
2. OID should be given followed by value, without having any comma (,) separator between value and OID.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command.
ERROR<Error code>	Failure

Response Parameters:

N/A

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

Example:
AT Mode:
To respond for string type of SNMP object requested by the remote SNMP server "AT+RSI_SNMP_GETNEXT=1.3.6.1.2.1.1.1.0" the command is :

```
at+rsi_snmp_getnext_rsp=4,ap_ssid\r\n<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>•<NUL>•<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•<NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>•<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•<NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• <NUL>•<NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>• <NUL>•
<NUL>• <NUL>• 1.3.6.1.2.1.1.4.0\r\n
```

```
0x61  0x74  0x2B  0x72  0x73  0x69  0x5F  0x73  0x6E  0x6D  0x70  0x5F
0x67  0x65  0x74  0x6E  0x65  0x78  0x74  0x5F  0x72  0x73  0x70  0x3D
0x34  0x2C 0x72  0x65  0x64  0x70  0x69  0x6E  0x65  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x00  0x31  0x2E
0x33  0x2E  0x36  0x2E  0x31  0x2E  0x32  0x2E  0x31  0x2E  0x31  0x2E
0x34  0x2E  0x30  0x0D  0x0A
```

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.31 SNMP Trap

Description:
This command is issued by the SNMP agent (running in module), to notify the management station of significant events. This command must be issued whenever user wants to send snmp trap to snmp server.

Command Format:
AT Mode:
at+rsi_snmp_trap=< snmp_version >,< ip_version >,< ipv4/ipv6 address >,< community >,< trap_type >,<trap_oid>,< elapsed_time>,<object_list_count>,<snmp_buf>\r\n

Command Parameters:
snmp_version[4 bytes]: snmp version (1/2/3). Currently only version 2 is supported for SNMP trap.

ip_version[4 bytes]: IP version (4 or 6)

destIPaddr.ipv4_address[4 bytes]: ipv4 address based upon the ip_version 4 selected. Module ignores remaining 12 bytes in case of ip version 4.

destIPaddr.ipv6_address[16 bytes]: ipv6 address based upon the ip_version 6 selected.

community[32 bytes]: community name

trap_type[1 byte]: type of trap

trap_type	type of trap
0	(coldStart)
1	(warmStart)
2	(linkDown)
3	(linkUp)
4	(authenticationFailure)
5	(egpNeighborLoss)
6	(User specific trap type)

trap_oid[51 bytes]: trap oid of the user specific trap .elapsed time: Total device time elapsed .
obj_list_count[1 byte]: Obeject variables list count
snmp_buf[1024 bytes]: snmp buf contains the array of snmp trap object variable structures(SNMP_TRAP_OBJECT).

- User has to fill snmp_buf with number of SNMP_TRAP_OBJECT (obj_list_count) structure.
- Based on snmp_object_data_type, User need to fill SNMP_TRAP_OBJECT structure.
- To use Octet string object data type, user needs to fill length of the string in snmp_object_octet_string_size parameter.

Based on string size user has to fill octet string. If string length is zero, Octet string has to be filled with NULL.

Note:
1. Maximum supported length for snmp Buffer is 1024.
2. Maximum supported object list count is 10.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command.
ERROR<Error code>	Failure

Response Parameters:

N/A

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C, 0xFF82,0x0100,0x0104,.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

> **Note**:
>
> For trap_type value 0 – 5 trap_oid parameter has to be NULL character.

Example:
AT Mode:

For ipv4:

1. Command usage for Integer Object data type:

```
at+rsi_snmp_trap=2,4,192.168.0.178,public,1,,1234,1,1.3.2.2.2.1.1.2\000\000\000\000\000\n\000\000\000\016\32
0\334G\001\000\000\000x\350\360G\n\000\000\000\002\000\000\000\020\000\000\000\f\000\000\000\004\000\00
0\000\270\r\001 \000\000\001\000\000\000\000\000#\001, '\000' <repeats 117 times>,\r\n
```

```
0x61 0x74 0x2b 0x72 0x73 0x69 0x5f 0x73 0x6e 0x6d 0x70 0x5f 0x74 0x72 0x61 0x70 0x3d 0x32 0x2c 0x34 0x2c
0x31 0x39 0x32 0x2e 0x31 0x36 0x38 0x2e 0x00 0x2e 0x31 0x37 0x38 0x2c 0x70 0x75 0x62 0x6c 0x69 0x63
0x2c 0x31 0x2c 0x2c 0x31 0x32 0x33 0x34 0x2c 0x31 0x2c 0x31 0x2e 0x33 0x2e 0x32 0x2e 0x32 0x2e 0x32
0x2e 0x31 0x2e 0x31 0x2e 0x32 0x00 0x00 0x00 0x00 0x00 0x0a 0x00 0x00 0x00 0x0e 0xd0 0xdc 0x47 0x01
0x00 0x00 0x00 0x78 0xe8 0xf0 0x47 0x0a 0x00 0x00 0x00 0x02 0x00 0x00 0x00 0x10 0x00 0x00 0x00 0x0c
0x00 0x00 0x00 0x04 0x00 0x00 0x00 0xb8 0x0d 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x23
0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0D 0x0A
```

2. Command usage for Octect string object adata type:

```
at+rsi_snmp_trap=2,4,192.168.0.178,public,1,,1234,1,1.4.4.4.4.4.4.3\000\000\000\000\000\n\000\000\000\016\32
0\334G\001\000\000\000x\350\360G\n\000\000\000\004", '\000' <repeats 11 times>, "\004\000\000\000\270\r\001
\000\000\001\000\000\000\000\000#\001\000\000\031\000\000\000Sample SNMP Trap String!!, '\000' <repeats
111 times>,\r\n
0x61 0x74 0x2b 0x72 0x73 0x69 0x5f 0x73 0x6e 0x6d 0x70 0x5f 0x74 0x72 0x61 0x70 0x3d 0x32 0x2c 0x34 0x2c
0x31 0x39 0x32 0x2e 0x31 0x36 0x38 0x2e 0x00 0x2e 0x31 0x37 0x38 0x2c 0x70 0x75 0x62 0x6c 0x69 0x63
0x2c 0x31 0x2c 0x2c 0x31 0x32 0x33 0x34 0x2c 0x31 0x2c 0x31 0x2e 0x34 0x2e 0x34 0x2e 0x34 0x2e 0x34
0x2e 0x34 0x2e 0x34 0x2e 0x33 0x00 0x00 0x00 0x00 0x00 0x0a 0x00 0x00 0x00 0x0e 0xd0 0xdc 0x47 0x01
0x00 0x00 0x00 0x78 0xe8 0xf0 0x47 0x0a 0x00 0x00 0x00 0x04 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x04 0x00 0x00 0x00 0xb8 0x0d 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x23
0x01 0x00 0x00 0x19 0x00 0x00 0x00 0x53 0x61 0x6d 0x70 0x6c 0x65 0x20 0x53 0x4e 0x4d 0x50 0x20 0x54
0x72 0x61 0x70 0x20 0x53 0x74 0x72 0x69 0x6e 0x67 0x21 0x21 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

```
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00 0x00
0x00 0x00 0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0D 0x0A
```

3. For ipv6:

```
at+rsi_snmp_trap=2,6,2001:db8:1:0:0:0:0:123,public,1,,1234,1,1.4.4.4.4.4.4.3\000\000\000\000\000\n\000\000\00
0\016\320\334G\001\000\000\000x\350\360G\n\000\000\000\004", '\000' <repeats 11 times>,
"\004\000\000\000\270\r\001 \000\000\001\000\000\000\000\000#\001\000\000\031\000\000\000Sample SNMP
Trap String!!, '\000' <repeats 111 times>,\r\n
0x61 0x74 0x2b 0x72 0x73 0x69 0x5f 0x73 0x6e 0x6d 0x70 0x5f 0x74 0x72 0x61 0x70 0x3d 0x32 0x2c 0x36 0x2c
0x32 0x30 0x30 0x30 0x31 0x3A 0x44 0x42 0x38 0x3A 0x31 0x3A 0x31 0x32 0x33 0x2c 0x2c 0x70 0x75 0x62
0x6c 0x69 0x63 0x2c 0x31 0x2c 0x2c 0x31 0x32 0x33 0x34 0x2c 0x31 0x2c 0x31 0x2e 0x34 0x2e 0x34 0x2e
0x34 0x2e 0x34 0x2e 0x34 0x2e 0x34 0x2e 0x33 0x00 0x00 0x00 0x00 0x00 0x0a 0x00 0x00 0x00 0x0e 0xd0
0xdc 0x47 0x01 0x00 0x00 0x00 0x78 0xe8 0xf0 0x47 0x0a 0x00 0x00 0x00 0x04 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x04 0x00 0x00 0x00 0xb8 0x0d 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00
0x00 0x00 0x23 0x01 0x00 0x00 0x19 0x00 0x00 0x00 0x53 0x61 0x6d 0x70 0x6c 0x65 0x20 0x53 0x4e 0x4d
0x50 0x20 0x54 0x72 0x61 0x70 0x20 0x53 0x74 0x72 0x69 0x6e 0x67 0x21 0x21 0x00 0x00 0x00 0x00 0x00
```

```
0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00

0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x000x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x000x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0D 0x0A
```

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.32 Open Socket

Description:
This command opens a TCP/UDP/SSL/Websocket client socket, a listening TCP/UDP/SSL socket.

Note:

1. A maximum of 10 sockets can be opened. Range of socket handles are between 1 to 10.
2. Module supports maximum of 2 SSL sockets. These can be either client or server sockets. When HTTPS server is enabled then user can open only 1 SSL socket.
3. Module supports maximum of 8 non SSL Web sockets and 2 SSL Web sockets.
4. After MQTT connection, the maximum LTCP can support is only 9.
5. Each SSL/Web socket will occupy one TCP socket. SSL is supported only in Wi-Fi Client mode
6. If SSL is enabled, module uses same SSL version until module reboots.
7. If SSL is enabled, module uses same CA certificate for both SSL server socket and SSL Client socket until module reboots.

Note:

Above case is valid only when BIT(27) is not set in tcp_ip_feature_bit_map (module will use SSL certificates from FLASH).

If BIT(27) (SSL certificate on to the RAM feature) is set in tcp_ip_feature_bit_map, module only supports either SSL server socket or SSL Client socket.

Command Format:
AT Mode:
To open TCP/SSL/Web socket over IPv4:
at+rsi_tcp=< destIPaddr >, < destSocket >, < moduleSocket >, < tos >, < ssl_ws_enable >, < ssl_ciphers >, < webs_resource_name >, < webs_host_name >, <tcp_retry_count>, < socket_bitmap >, <rx_window_size>, <tcp_keepalive_timeout>, < vap_id >, <cert_inx>\r\n

To open TCP/SSL/Web socket over IPv6:
at+rsi_tcp6=< destIPaddr >, < destSocket >, < moduleSocket >, < tos >, < ssl_ws_enable >, <ssl_ciphers >, < webs_resource_name >, < webs_host_name >, <tcp_retry_count>, < socket_bitmap >, <rx_window_size>, <tcp_keepalive_timeout>, < vap_id >\r\n

To open TCP/SSL server socket over IPv4:
at+rsi_ltcp=< moduleSocket >, < max_count >, < tos >, < ssl_ws_enable >, <ssl_ciphers >, <tcp_retry_count>, < socket_bitmap >, <rx_window_size>, <tcp_keepalive_timeout>, < vap_id >, <cert_inx>\r\n

To open TCP/SSL server socket over IPv6:
at+rsi_ltcp6=< moduleSocket >, < max_count >, < tos >, < ssl_ws_enable >, < ssl_ciphers >, <tcp_retry_count>, < socket_bitmap ><rx_window_size>, <tcp_keepalive_timeout>, < vap_id >\r\n

To open LUDP socket over IPv4:
at+rsi_ludp=< moduleSocket >,< tos >,< socket_bitmap >,< vap_id >\r\n

To open LUDP socket over IPv6:
at+rsi_ludp6=< moduleSocket >, < tos >, < socket_bitmap >, < vap_id >\r\n

It is recommended to enable BIT(16) in 'ext_tcp_ip_feature_bit_map' in 'at+rsi_opermode' when using TCP sockets.

Command Parameters:

moduleSocket (2 bytes): Port number of the socket in the module. Value ranges from 1024 to 49151.

Note:

User can give any Port number from the above range except 30000, as that port number is reserved for specific feature.

destSocket (2 bytes): destination port. Value ranges from 1024 to 49151. Ignored when TCP server or Listening UDP sockets are to be opened.

Note:

User can give any Port number from the above range except 30000, as that port number is reserved for specific feature.

destIPaddr.ipv4_address (4 bytes): IP Address of the target server. Ignored when TCP server or Listening UDP sockets are to be opened. If ip_version is 4 then only first four bytes of the ipv4_address is filled rest twelve bytes will be 0.

destIPaddr.ipv6_address (16 bytes): IPv6 Address of the target server. Ignored when TCP server or Listening UDP sockets are to be opened. All 16 bytes are filled if ip_version is 6.

max_count (2 bytes): Maximum number of clients which can be connected in case of LTCP.

Note:

1. Module supports maximum 2 SSL sockets, so max_count should be less than or equal to 2 in case of LTCP. If max_count is 2 then host can create only one SSL based LTCP socket with two clients support.
2. This field '**max_count**' can be ignored if the socket type is other than 2 (TCP server).

tos (4 bytes): type of service field. Possible values are 0-7.

Table 11 TOS Values

TOS Value	Description
0	Best Effort
1	Priority
2	Immediate
3	Flash-mainly used for voice signaling
4	Flash Override
5	Critical-mainly used for voice RTP

TOS Value	Description
6	Internet
7	Network

ssl_bitmap (1 byte): This field is used to enable following:

Possible values:
0 – To open TCP socket.
BIT(0) - To open SSL Client socket.
BIT(1) - To open Web socket client.
BIT(2) - To open SSL socket with TLS 1.0 version.
BIT(3) - To open SSL socket with TLS 1.2 version.
BIT(7) – To open High performance TCP RX socket.

Note:
To Support SSL Socket with Multiple TLS Versions need to set extended custom feature bitmap i.e BIT[14].

Examples:
ssl_ws_enable value should be

- '0' to open normal TCP socket
- '1' to open SSL over TCP socket. By default, module will open SSL socket which support both TLS 1.0 and TLS 1.2.
- '2' to open web socket client over TCP socket
- '3' to open web socket over SSL TCP socket
- '5' to open SSL over TCP socket with TLS 1.0 version

- '9' to open SSL over TCP socket with TLS 1.2 version
- '7' to open web socket client over SSL TCP socket with TLS 1.0 version
- '11' to open web socket client over SSL TCP socket with TLS 1.2 version
- '128' to open High performance TCP socket
- '129' to open High performance TCP socket over SSL
- '130' to open High performance web socket TCP client socket
- '131' to open High performance web socket TCP client socket over SSL

ssl_bitmap (1 byte): 1-byte bitmap used to select the various cipher modes. This field is optional, and the possible values are listed below:

Bit Position	Value	Name
BIT(1)	2	TLS_RSA_WITH_AES_256_CBC_SHA256
BIT(2)	4	TLS_RSA_WITH_AES_128_CBC_SHA256
BIT(3)	8	TLS_RSA_WITH_AES_256_CBC_SHA
BIT(4)	16	TLS_RSA_WITH_AES_128_CBC_SHA
BIT(5)	32	TLS_RSA_WITH_AES_128_CCM_8
BIT(6)	64	TLS_RSA_WITH_AES_256_CCM_8

These values can be OR'ed together to select multiple ciphers. To select all the ciphers, either all bits can be set or alternatively, 0 can be passed.

Other than SSL bitmap cipher modes few default cipher modes are also supported.

Default cipher modes:

TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBE_SHA

TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA

TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA

TLS_DHE_RSA_WITH_AES_128_CBC_SHA

TLS_DHE_RSA_WITH_AES_256_CBC_SHA

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

Curve IDs supported:

Curve Id	Description	Support
15	secp160k1	Yes
16	secp160r1	Yes
17	secp160r2	Yes
18	secp192k1	Yes
19	secp192r1	Yes
20	secp224k1	Yes
21	secp224r1	Yes
22	secp256k1	Yes
23	secp256r1	Yes
24	secp384r1	Yes
25	secp521r1	Yes
26	brainpoolP256r1	Yes
27	brainpoolP384r1	Yes
28	brainpoolP512r1	Yes

webs_resource_name (51 bytes): Web socket resource name.

A string of 50 characters is the maximum possible input to this variable webs_host_name: Web socket host name.
A string of 50 characters is the maximum possible input to this variable

tcp_retry_count (1 byte): To configure tcp retransmissions count

socket_bitmap (1 byte): To configure socket bit map

socket_bitmap	Functionality
socket_bitmap[0]	synchronous data read **Info:** Module sends data to host only after receiving data read request from host.
socket_bitmap[1]	TCP socket **Info:** To open a listening TCP socket and to accept client connection, host need to provide accept command.
socket_bitmap[2]	TCP ACK indication **Info:** When this bit is enabled module gives an TCP ACK indication(Frame type 0xAB) to host after receiving TCP ACK from remote peer. Host has to send next data packet to module only after receiving this TCP ACK indication
socket_bitmap[3]	If this bit is set module handles small sized received packets effectively. **Info:** Recommended to set for the sockets which receive small size packets
socket_bitmap[4]	TCP RX window size **Info:** When this bit is enabled, module opens socket with RX window size based on the value provided in rx_window_size field

BIT(0) : Set to enable synchronous data read
Module sends data to host only after receiving data read request from host.

BIT(1) : To open a listening TCP socket, on which to accept client connection host need to provide accept command.
BIT(2): Set to enable TCP ACK indication. This bit is valid for TCP/SSL Client, TCP/SSL Server (Listening TCP) and Web Sockets.

When this bit is enabled module gives an TCP ACK indication (Frame type 0xAB) to host, when it receives TCP ACK from remote peer. Host has to send next data packet to module only after receiving this TCP ACK indication.

BIT(3): If this bit is set module handles small size received packets effectively.
Recommended to set for the sockets which receives small size packets .

BIT(4): Set to configure TCP RX window size. This bit is valid for TCP/SSL Client, TCP/SSL Server (Listening TCP).
When this bit is enabled module opens socket with RX window size based on the value provided in rx_window_size field

rx_window_size (1 byte): This field is used to configure the RX window size for the TCP socket. Possible values are 1 to 15 based on the availability of memory.

tcp_keepalive_timeout (2 bytes): This field is used to configure TCP keep alive initial timeout in seconds.

cert_inx (1 byte): This field is used for the Certificate index

Response:
AT Mode:
For TCP/SSL/Web socket over IPv4/IPv6:

Result Code	Description
OK< ip_version >< socketType >< socketDescriptor >< moduleSocket >< ipv4_addr /ipv6_addr > < mss >< window_size >	Successful execution of the command.
ERROR<Error code>	Failure

For TCP/SSL server socket over IPv4 / IPv6:

or
For LUDP socket over IPv4 / IPv6:

Result Code	Description
OK< ip_version >< socketType >< socketDescriptor >< moduleSocket >< ipv4_addr /ipv6_addr >	Successful execution of the command.
ERROR<Error code>	Failure

Response Parameters:

ip_version(2 bytes): IP version used, either 4 or 6.

socketType(2 bytes): Type of the created socket.
0–TCP/SSL Client
2–TCP/SSL Server (Listening TCP)
4–Listening UDP

socketDescriptor(2 bytes): Created socket's descriptor or handle, starts from 1. socketDescriptor ranges from 1 to 10. The first socket opened will have a socket descriptor of 1, the second socket will have 2 and so on.

moduleSocket(2 bytes): Port number of the socket in the module.

moduleIPaddr.ipv4_address(16 bytes): The IPv4 address of the module. Only first four bytes of ipv4_address is filled rest 12 bytes are '0' in case of IPv4.

moduleIPaddr.ipv6_address(16 bytes): The IPv6 address of the module in case of IPv6.

mss(2 bytes): maximum segment size of the remote peer. In case of Ludp/Ltcp this field will not present.

window_size(4 bytes): Window size of the remote peer. In case of Ludp/Ltcp this field will not present.

Possible error codes:

Possible error codes are 0xBB46, 0xBB22, 0xBB23, 0xBB33, 0xBB34, 0xBB35, 0xBB36, 0xBB45, 0xBB46, 0x0015, 0x0021, 0x0025, 0x002C, 0xFF74, 0xBBD3, 0xBBD2, 0xBBD1, 0xFF80.

Relevance:

This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

Example:

AT Mode:
To TCP socket over IPv4:
at+rsi_tcp=192.168.40.10,8000,1234,0,1,0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x74 0x63 0x70 0x3D 0x31 0x39 0x32 0x2E 0x31 0x36 0x38 0x2E 0x34 0x30 0x2E 0x31 0x30 0x2C 0x38 0x30 0x30 0x30 0x2C 0x31 0x32 0x33 0x34 0x2C 0x30 0x2C 0x31 0x2C 0x30 0x0D 0x0A

Response:
OK< ip_version =0x04 0x00>< socketType =0x0000 >< socketDescriptor =0x0001>< moduleSocket =0x4d2>< ipv4_addr= 0xC0 0xA8 0x28 0x120x00(12 times)> < mss =0xB4 0x05>< window_size =0x00 0x00 0x01 0x0>\r\n
0x4F 0x4B 0x04 0x00 0x00 0x00 0x01 0x00 0xd2 0x04 0xC0 0xA8 0x28 0x12 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0xB4 0x05 0x00 0x00 0x01 0x00 0x0D 0x0A
To open Web socket over IPv4:
at+rsi_tcp=174.129.224.73,80,1234,0,2,0,echo.websocket.org\r\n

0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x74 0x63 0x70 0x3D 0x31 0x37 0x34 0x2E 0x31 0x32 0x39 0x2E 0x32 0x32 0x34 0x2E 0x37 0x33 0x2C 0x38 0x30 0x2C 0x35 0x30 0x30 0x30 0x2C 0x30 0x2C 0x32 0x2C 0x30 0x2C 0x2C 0x65 0x63 0x68 0x6F 0x2E 0x77 0x65 0x62 0x73 0x6F 0x63 0x6B 0x65 0x74 0x2E 0x6F 0x72 0x67 0x3C 0x43 0x52 0x3E 0x3C 0x4C 0x46 0x3E

Response:
OK< ip_version =0x04 0x00>< socketType =0x0000 >< socketDescriptor =0x0001>< moduleSocket =0x4d2>< ipv4_addr= 0xC0 0xA8 0x28 0x120x00(12 times)> < mss =0xB4 0x05>< window_size =0x00 0x00 0x01 0x0>\r\n
0x4F 0x4B 0x04 0x00 0x00 0x00 0x01 0x00 0xd2 0x04 0xC0 0xA8 0x28 0x12 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xB4 0x05 0x00 0x00 0x01 0x00 0x0D 0x0A
To open Web socket over IPv6:
at+rsi_tcp6=2001:DB8:1:0:0:0:0:153,8000,1234,2,1,0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x74 0x63 0x70 0x36 0x3D 0x32 0x30 0x30 0x31 0x3A 0x44 0x42 0x38 0x3A 0x31 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x31 0x35 0x33 0x2C 0x38 0x30 0x30 0x30 0x2C 0x31 0x32 0x33 0x34 0x2C 0x32 0x2C 0x31 0x2C 0x30 0x0D 0x0A

Response:
OK< ip_version =0x06 0x00>< socketType =0x0000 >< socketDescriptor =0x0001>< moduleSocket =0x4d2>< ipv6_addr= addr 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x53 0x01 0x00 0x00 > < mss =0xB4 0x05>< window_size =0x00 0x00 0x01 0x00>\r\n
0x4F 0x4B 0x06 0x00 0x00 0x00 0x01 0x00 0xd2 0x04 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x53 0x01 0x00 0x00 0xB4 0x05 0x00 0x00 0x01 0x00 0x0D 0x0A
To open LTCP socket over IPv4 :
at+rsi_ltcp=1234,5,7,1,0r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6C 0x74 0x63 0x70 0x3D 0x31 0x32 0x33 0x34 0x2C 0x35 0x2C 0x37 0x2C 0x31 0x2C 0x30 0x0D 0x0A

Response:
OK<ip_version><socket_type><socket_handle><Lport><module_ipv4addr>\r\n
OK< ip_version =0x04 0x00>< socketType =0x0002 >< socketDescriptor =0x0001>< moduleSocket =0x4d2>< ipv4_addr= 0xC0 0xA8 0x12 0xD2 0x00(12 times) > \r\n
0x4F 0x4B 0x02 0x00 0x01 0x00 0xd2 0x04 0xC0 0xA8 0x28 0x12 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0D 0x0A
To open LTCP socket over IPv6:
at+rsi_ltcp6=1234,5,7,1,0r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6C 0x74 0x63 0x70 0x3D 0x31 0x32 0x33 0x34 0x2C 0x35 0x2C 0x37 0x2C 0x31 0x2C 0x30 0x0D 0x0A

Response:
OK<ip_version><socket_type> <socket_handle><Lport> <module_ipv6_addr>\r\n
OK< ip_version =0x06 0x00>< socketType =0x0002 >< socketDescriptor =0x0001>< moduleSocket =0x4d2>< ipv6_addr= 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x36 0x01 0x00 0x00 > \r\n
0x4F 0x4B 0x06 0x00 0x02 0x00 0x01 0x00 0xd2 0x04 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x36 0x01 0x00 0x00 0x0D 0x0A
To open LUDP socket over IPv4:
at+rsi_ludp=1234,7\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6C 0x75 0x64 0x70 0x3D 0x31 0x32 0x33 0x34 0x2C 0x7 0x0D 0x0A

Response:
OK< ip_version =0x04 0x00>< socketType =0x0004 >< socketDescriptor =0x0001>< moduleSocket =0x4d2>< ipv4_addr= 0xC0 0xA8 0x28 0x12 0x00(12 times) > \r\n
0x4F 0x4B 0x04 0x00 0x04 0x00 0x01 0x00 0xd2 0x04 0xC0 0xA8 0x28 0x12 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0D 0x0A
To open LUDP socket over IPv6:
at+rsi_ludp6=1234,5\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6C 0x75 0x64 0x70 0x36 0x3D 0x31 0x32 0x33 0x34 0x2C 0x35 0x0D 0x0A

Response:
OK< ip_version =0x06 0x00>< socketType =0x0004 >< socketDescriptor =0x0001>< moduleSocket =0x4d2>< ipv6_addr= 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x36 0x01 0x00 0x00 > \r\n
0x4F 0x4B 0x06 0x00 0x04 0x00 0x01 0x00 0xd2 0x04 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x36 0x01 0x00 0x00 0x0D 0x0A

5.33 TCP Socket Connection Established

Description:
If a server TCP socket is opened in the module, the socket remains in listening state till the time the remote terminal opens and binds a corresponding client TCP socket. Once the socket binding is done, the module sends an asynchronous message to the Host to indicate that its server socket is now connected to a client socket.

Note:
If SSL is enabled module will use same SSL version until module reboots.

Command Format:
AT Mode:
N/A

Command Parameters:
N/A

Response:
AT Mode:

Result Code	Description
AT+RSI_LTCP_CONNECT=< ip_version >< socket >,<fromPortNum >,< ipv4_address / ipv6_address >< mss >< window_size >< SrcPortNum >	Asynchronous message from modue to host on tcp connection establishment .

Response Parameter:

ip_version(2 bytes): IP version used, either 4 or 6.

Sock_id(2 bytes): Socket descriptor of the server TCP socket.

fromPortNum(2 bytes): Port number of the remote socket

dst_ip_address .ipv4_address(16 bytes): Remote IPv4 address. Only first four bytes of ipv4_address are filled, rest 12 bytes are zero.

dst_ip_address.ipv6_address(16 bytes): Remote IPv6 address

mss(2 bytes): Maximum segment size of remote peer.

window_size(4 bytes): Window size of the remote peer.

SrcPortNum(2 bytes): Source Port Number to which client is connected.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

5.34 Query a Listening Socket's Active Connection Status

Description:
This command is issued when a listening/server TCP/SSL socket has been opened in the module, to know whether the socket got connected to a client socket.

Command Format:

AT Mode:
at+rsi_ctcp=< socketDescriptor >\r\n

Response:
AT Mode:

Result Code	Description
OK< socketDescriptor >< ip_version >< ipv4_address / ipv6_address >< destPort >	Successful Execution of Command.

Result Code	Description
ERROR <Error>	Failure

Response Parameters:
ip_version(2 bytes): IP version used, either 4 or 6.

socketDescriptor(2 bytes): Socket handle for an already open listening TCP/SSL socket in the module.

destIPaddr(16 bytes): Destination IPv4/IPv6 address of the remote peer whose socket is connected. Last 12 bytes will be filled with '0' in case of IPv6.

dPort (2 bytes): Port number of the remote peer client which is connected

Possible Error codes:
Possible error codes are 0x0021, 0x0025, 0x002C, 0xFF86,0xFFFA,0xFF82.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1 , 2 or 6.

Example:
AT Mode:

Response:
OK< socketDescriptor =7>< ipv4_address =192.168.40.10>
< destPort =8001> \r\n
0x4F 0x4B 0x07 0x00 0xC0 0xA8 0x28 0x0A 0x41 0x1F 0x0D 0x0A

5.35 Close Socket

Description:
This command closes a TCP/LTCP/UDP/SSL/Web socket in the module.

Command Format:

AT Mode:
at+rsi_cls=< socketDescriptor >,< port_number >\r\n

Parameters:

socketDescriptor (2 bytes): Socket descriptor of the socket to be closed.

port_number (2 bytes): This field is valid only for LTCP socket.

When this filed is mentioned module closes all LTCP connections which are opened with provided port number.

Note:

1. This field will be ignored in case of closing UDP/TCP client sockets.
2. In order to use port based socket close to close all LTCP sockets, user has to set socket_handle as zero.
3. For web socket: If close command is given socket will be closed without waiting for server to initiate web socket close.

In order get graceful closure, host has to issue data send command with an opcode of 0x8 and fin bit set and with an dummy data. This dummy data will be ignored by server side web socket(Example: at+rsi_snd=1,0,0,136,\r\n in AT mode).

In this case module sends web socket close frame to server. On receiving this frame server initiates web socket close. After successful socket closes exchanges, module gives remote terminate indication to host.

Response:

AT Mode:

Result Code	Description
OK< socketDsc >< bytesSent >	Successful execution of command

Result Code	Description
ERROR<Error code>	Failure

Note:

In the case of TCP socket, when a remote peer closes the socket connection, the module sends the "AT+RSI_CLOSE<socket_handle><number of bytes sent>\r\n" message to the Host. This is an asynchronous message sent from module to host and not the response of a specific command. Socket_handle is sent in 2 bytes, number of bytes sent is 4 bytes in hex. The least significant byte is returned first. AT+RSI_CLOSE is returned in uppercase and ASCII format.

Response Parameters:

socketDsc(2 bytes): Socket descriptor of the socket closed.
sentBytescnt(4 bytes):Number of bytes sent successfully on that socket.

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C, 0xFF86,0xBB35,0xBB27,0xBB42

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

Example:

AT Mode:
To close the socket with handle 1, the command is
at+rsi_cls=1\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x63 0x6C 0x73 0x3D 0x31 0x0D 0x0A

Response:
OK<socket_handle><number of bytes sent>\r\n
0x4F 0x4B 0x01 0x00 0x01 0x02 0x03 0x04 0x0D 0x0A

5.36 Send Data

This section explains how to send data from the host to the module.

5.36.1 Send Data in AT mode

Description:
This command is used to send data from the host to the module, which is transmitted over a wireless media in AT mode.

Note:

1. Maximum data that can be sent over TCP/LTCP socket is 1460 Bytes
2. Maximum data that can be sent over LUDP socket is 1472 Bytes
3. Maximum data that can be sent over Web socket is 1450 Bytes
4. Maximum data that can be sent over TCP-SSL/LTCP-SSL is 1370 Bytes
5. Maximum data that can be sent over Web socket over SSL is 1362 Bytes
6. Maximum data that can be sent over TCP/LTCP socket is 1440 Bytes in IPv6 mode.
7. Maximum data that can be sent over LUDP socket is 1452 Bytes in IPv6 mode

Command Format:

To send data over IPv4:

at+rsi_snd=< socketDescriptor >,< sendBufLen >, < destIPaddr >,< destPort >, < sendDataBuf >\r\n

To send data over IPv6:

at+rsi_snd6=< socketDescriptor >,< sendBufLen >, < destIPaddr >,< destPort >, < sendDataBuf >\r\n

Command Parameters:

socketDescriptor (2 byte): Socket handle of the socket over which data is to be sent.

sendBufLen (4 bytes): Length of the data that is getting transmitted. Wrong parameter may cause module to hang in some cases. In case of Web socket correct length is mandatory.

destIPaddr (4 bytes): Destination IPv4/IPv6 Address. Should be '0' in case of data transmitting on a TCP/LTCP/SSL socket.

destPort (2 bytes): Destination Port. Should be '0' in case of data transmitting on a TCP/LTCP/SSL socket.

In case of Web sockets, this field represents web socket information.
In web socket information seventh bit indicates the FIN packet and bit [3:0] indicates the opcode (type of the packet to be included in web socket header).

OPCODE should be as follows (Refer RFC 6455):

Mode	Functionality
0	Continuation frame
1	Text frame
2	Binary frame
[3-7]	Reserved for further non-control frames
8	Connection close frame
9	Ping frame
10	Pong frame
[B-F]	Reserved for further control frames
FIN Bit	0: More web socket frames to be followed. 1: Final frame web socket message.

sendDataBuf (1400 bytes): Actual data to be sent to the specified socket.

Response:

Result Code	Description
OK<length> (or) OK<socket id ><length>	2 bytes length (2 bytes hex), length of data sent. 1-byte socket id, 2 bytes length (2 bytes hex), length of data sent.
ERROR<Error code>	Failure On a failure while sending the data on the TCP socket, if the error code indicates "TCP connection closed", then the module closes the socket. Possible error codes are 0x0030,0xFFFE,0xFF7E,0xFFF8,0x003F,0xFFF7.

When enabled the socket BIT(2) (opensocket)

Note:

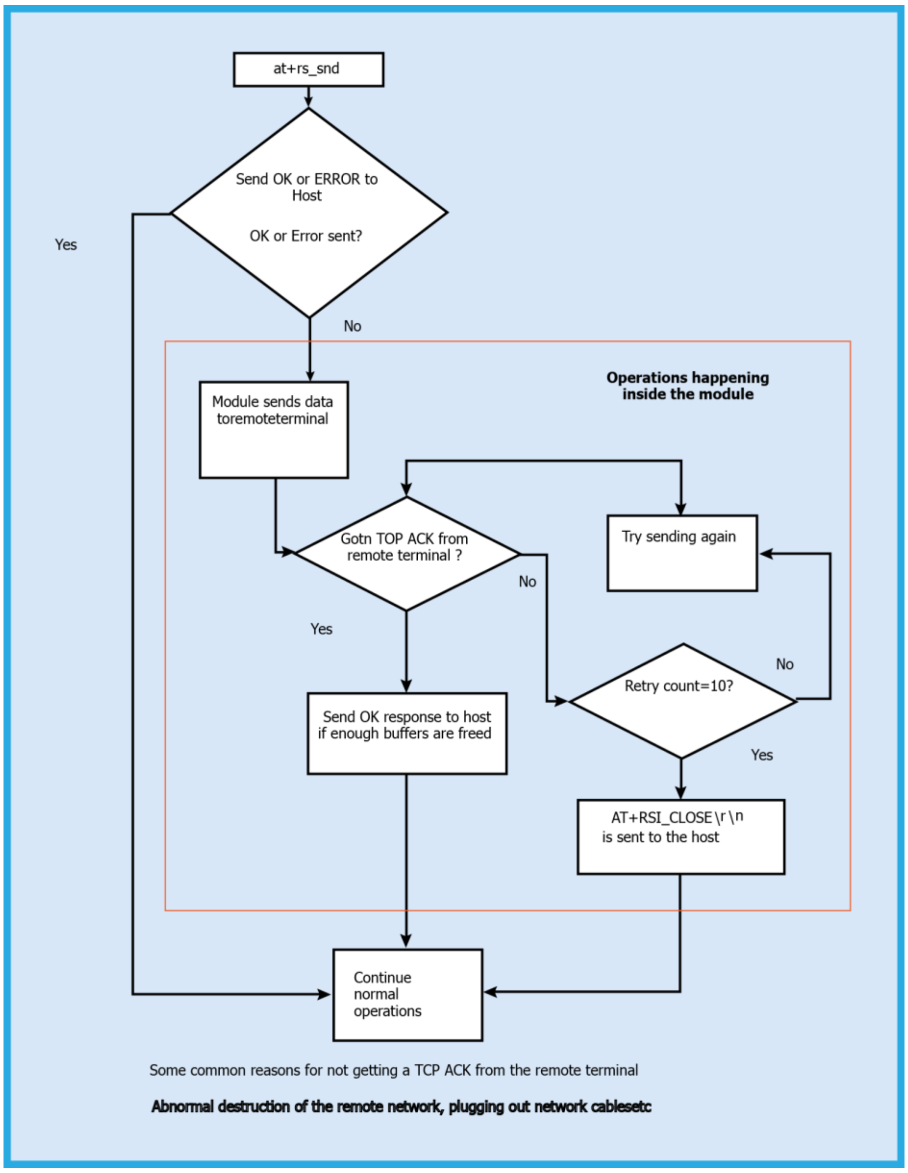
1. When enabled the socket bit map(2) (open socket), the send response gives 3bytes. User needs to consider the following for “snd” command if TCP_ACK(bit[2]) is enabled. User will get an immediate “OK<socket id ><length>\r\n” response for “snd” command. This indicates the “snd” command transaction happened successfully at the host interface level. This doesn’t mean that the packet is successfully

transmitted to the remote peer. Module responds with “OK<socket id ><length>\r\n” and takes the next “snd” command till it has buffers to buffer those packets.

2. In case of SSL socket the response of send command gives length of data (includes SSL data) on the TCP socket.

Note:

1. The parameter sendDataBuf cont.
2. The parameter sendDataBuf contains the actual data not the ACII representations of the data. User need to consider following for "snd" command in case of UART mode. User will get an immediate "OK<length>\r\n" response for "snd" command. This indicates the "snd" command transaction happened successfully at the host interface level. This doesn't mean that the packet is successfully transmitted to the remote peer. Module responds with "OK<length>\r\n" and takes the next "snd" command till it has buffers to buffer those packets. User need to take care that the data_len value that is given in "snd" command should be same as the number of bytes that are getting transmitted with "snd" command. In TCP/IP bypass only < sendDataBuf > parameter is valid and remaining parameters are dummy user can send 0.


at+rs_snd
Send OK or ERROR to Host
OK or Error sent?
Yes
No
Operations happening inside the module
Module sends data toremoteterminal
Gotn TOP ACK from remote terminal ?
Try sending again
No
Yes
Retry count=10?
No
Send OK response to host if enough buffers are freed
Yes
AT+RSI_CLOSE\r\n is sent to the host
Continue normal operations
Some common reasons for not getting a TCP ACK from the remote terminal
Abnormal destruction of the remote network, plugging out network cablesetc

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Note: on Byte Stuffing:

The '\r\n' character sequence (0x0D, 0x0A in hex) is used to indicate the termination of an AT command. If the actual data to be sent from Host comprises of \r\n characters in sequence, the host should replace this set of characters with (0xDB) and (0xDC). If (0xDB) itself is part of the data then (0xDB 0xDD) has to be sent. If (0xDB 0xDC) itself is part of the data, then (0xDB 0xDD 0xDC) has to be sent. If either 0xDD or 0xDC is not sent after 0xDB, then an error (-9) is sent.

Example 1 : If 0x41 0x42 0x43 0x0D 0x0A is the actual data stream that needs to be sent then the command is
at+rsi_snd <hn> <sz=5> <Dip> <Dport> < 0x41> <0x42> <0x43> <0xDB> <0xDC> <0x0D> <0x0A>

Example 2 : If 0x41 0x42 0x43 0x0D 0x0A 0x31 0x32 is the actual data stream that needs to be sent then the command is
at+rsi_snd <hn> <sz=7> <Dip> <Dport> < 0x41> <0x42> <0x43> <0xDB> <0xDC> <0x31> <0x32> <0x0D> <0x0A>

Example 3 : If 0x41 0x42 0x43 0xDB 0x31 0x32 is the actual data stream that needs to be sent then the command is
at+rsi_snd <hn> <sz=7> <Dip> <Dport> < 0x41> <0x42> <0x43> <0xDB> <0xDD> <0x31> <0x32> <0x0D> <0x0A>

Example 4 : If 0x41 0x42 0x43 0xDB 0xDC 0x31 0x32 is the actual data that needs to be transmitted, then the command is
at+rsi_snd <hn> <sz=8> <Dip> <Dport> <0x41> <0x42> <0x43> <0xDB><0xDD><0xDC> <0x31><0x32> <0x0D> <0x0A>

Example 5 : If 0x41 0x42 0x43 0x0D 0x0A 0xDB 0x31 0x32 is the actual data that needs to be transmitted, then the command is
at+rsi_snd <hn> <sz=9> <Dip> <Dport> <0x41> <0x42> <0x43> <0xDB><0xDC> <0xDB> <0xDD> <0x31><0x32> <0x0D> <0x0A>

Example 6 : If 0x41 0x42 0x43 0x0D 0x0A 0xDB 0xDC 0x31 0x32 is the actual data that needs to be transmitted, then the command is
at+rsi_snd <hn> <sz=10> <Dip> <Dport> <0x41> <0x42> <0x43> <0xDB><0xDC> <0xDB> <0xDD> <0xDC> <0x31><0x32> <0x0D> <0x0A>
at+rsi_snd is the only command that requires byte stuffing to be done by the Host before sending to the module. There are NO other commands (from Host to module) that require byte stuffing. There are NO responses (from module to Host) that are byte stuffed by module before giving to Host.

Example: Data Stuffing

Command:
To send a data stream 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A over a TCP socket
at+rsi_snd=1,10,0,0, 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A\r\n
0x61 0x74 0x2B 0x72 0x73 0x690x5F0x730x6E0x64 0x3D 0x31 0x2C 0x31 0x30 0x2C 0x300x2C0x300x2C 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0D 0x0A

To send a data stream 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A over a UDP socket to a destination IP 192.168.1.20 and destination port 8001
at+rsi_snd=1,10,192.168.1.20,8001, 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A\r\n
0x61 0x74 0x2B 0x72 0x730x690x5F0x730x6E0x64 0x3D 0x31 0x2C 0x31 0x30 0x2C 0x310x39 0x32 0x2E 0x31 0x36 0x38 0x2E 0x31 0x2E 0x32 0x30 0x2C 0x38 0x30 0x30 0x31 0x2C 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0D 0x0A

To send a stream "abcdefghij" over a Multicast socket
at+rsi_snd=1,10,239.0.0.0,1900,abcdefghij\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x73 0x6E 0x64 0x3D 0x31 0x2C 0x31 0x30 0x2C 0x32 0x33 0x39 0x2E 0x30 0x2E 0x30 0x2E 0x30 0x2C 0x31 0x39 0x30 0x30 0x2C0x61 0x62 0x63 0x64 0x65 0x66 0x67 0x68 0x69 0x6A 0x0D 0x0A

Response:
For 250 bytes sent, the response is

OK 250\r\n
0x4F 0x4B 0xFA 0x00 0x0D 0x0A

When TCP_ACK(bit[2]) is enabled the send response gives 3bytes,1byte for socket id and 2bytes represents the length.
at+rsi_snd=1,10,0,0,0123456789\r\n

Response:

OK
0x02\0x00
0x4F 0x4B 0x02 0x0A 0x00 0x0D 0x0A
When enable socket bit map of tcp socket the response of send command over ssl

at+rsi_snd=1,10,0,0,hellohello\r\n

Response:
0x4F 0x4B 0x01 0x45 0x00 0x0D 0x0A

Possible error codes:
Possible error codes are -2, 63.

Relevance:
This command is relevant when module is configured in operating mode 0,1,2 or 6.

5.37 Read Data

This section explains how to read socket data from the module to host.

Note:
To use this feature, BIT(0) should be set in socketbitmap while opening a socket.

5.37.1 Read Data in AT mode

Description:
This command is used to request number of bytes received on given socket. Module will give requested number of bytes received on particular socket (synchronous) only if this command is given from host. If requested numbers of bytes are greater than bytes available on given socket module will return only available number of bytes to host. If no data available on given socket module will wait till data received on given socket to serve this command.

Command Format:
To read data
at+rsi_read=< socketDescriptor >,< no of bytes >,<timeout in ms>\r\n

Command Parameters:
socketDescriptor (1 byte): Socket handle of the socket over which data is to be received.
no of bytes (4 bytes): Length of the data that has to be read from the module.
time out in ms (2 bytes): Read timeout in milliseconds to configure read data time out.

Response:

Result Code	Description
AT+RSI_READ<ip_version > < recvSocket >< recvBufLen >< ipv4_address / ipv6_address > < src_port >< recvDataBuf >	**ip_version (2 bytes, hex):** IP version of the data received. 4 – for IPv4 6 – for IPv6 **recvSocket (2 bytes, hex)**: socket handle of the socket over which the data is received. The least significant byte is returned first. If Web socket has been enabled, upper byte holds the web socket info. Seventh bit indicates the FIN packet and bit 3:0 gives the opcode information from web socket header.

Result Code	Description
	recvBufLen (2 bytes, hex): Number of bytes received. The least significant byte is sent first. For example, 900 bytes (0x0384) would be sent as <0x84> <0x03> **ipv4_address / ipv6_address (16 bytes, hex):** Source IPv4/IPv6 address. i.e. IP address from which data is received. This field is not present in the message if the data is received over a TCP socket. Only first 4 bytes are filled rest 12 bytes are zero for IPv4 address. **src_port (2 bytes, hex):** Source port. i.e. port number from which data is received. This field is not present in the message if the data is received over a TCP socket. **recvDataBuf:** Actual received data stream. A maximum of 1472 bytes can be received in case of UDP and 1460 bytes in case of TCP, in this field. When the module sends data to the Host, byte stuffing is NOT done by the module. The size parameter should be used to know how many bytes of valid data is expected.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Example:
Command:
at+rsi_read=1,4,100\r\n

Response:

For IPV4:
If 'abcd' is sent from remote terminal to module, on an UDP socket with handle 1, from source ipv4 address 192.168.1.1 to destination ipv4 address 192.168.1.2(module address) and source port 8001, the module sends the following response to the host.
AT+RSI_READ 4 1 4 192 168 1 1 8001 abcd \r\n
0x41 0x54 0x2B 0x52 0x53 0x49 0x5F 0x52 0x45 0x41 0x44 0x04 0x00 0x01 0x00 0x04 0x00 0xC0 0xA8 0x01 0x01 0x41 0x1F 0x61 0x62 0x63 0x64 0x0D 0x0A
If 'abcd' is sent from remote terminal to module, on a TCP socket with handle 1, the module sends the following response to the host.
AT+RSI_READ 4 1 4 abcd \r\n
0x41 0x54 0x2B 0x52 0x53 0x49 0x5F 0x52 0x45 0x41 0x44 0x04 0x00 0x01 0x00 0x04 0x00 0x61 0x62 0x63 0x64 0x0D 0x0A

For IPV6:
If 'abcd' is sent from remote terminal to module, on an UDP socket with handle 1, from source ipv6 address 2001:db8:1:0:0:0:0:153to sdestinationipv6 address 2001:db8:1:0:0:0:0:154 (module address)and source port 8001, the module sends the following response to the host.
AT+RSI_READ 6 1 4 2001:db8:1:0:0:0:0:153 8001 abcd \r\n
0x41 0x54 0x2B 0x52 0x53 0x49 0x5F 0x52 0x45 0x41 0x44 0x06 0x00 0x01 0x00 0x04 0x00 0x32 0x30 0x30 0x31 0x3A 0x44 0x42 0x38 0x3A 0x31 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x31 0x35 0x33 0x41 0x1F 0x61 0x62 0x63 0x64 0x0D 0x0A

Note:
The data delivered to the Host on receiving data on a TCP socket does not include the source IP address and source port (Sip and Sport).

5.37.2 Receive Data on Socket

This section explains how module sends received data to Host.

Note:
Module support maximum SSL record size is 8K. If it is more than 8K module will close the socket.

5.37.3 Receive Data on Socket in AT mode

Description:
The module delivers the data obtained on a socket to the Host with this message. This is an asynchronous response. It is sent from the module to the host when the module receives data from a remote terminal. SSL data is also received in a similar fashion.

Command Parameters:
N/A

Response:

Result Code	Description
AT+RSI_READ<ip_version > < recvSocket >< recvBufLen >< ipv4_address / ipv6_address > < src_port >< recvDataBuf >	**ip_version (2 bytes, hex):** IP version of the data received. 4 – for IPv4 6 – for IPv6 **recvSocket (2 bytes, hex)**: socket handle of the socket over which the data is received. The least significant byte is returned first. If Web socket has been enabled, upper byte holds the web socket info. Seventh bit indicates the FIN packet and bit 3:0 gives the opcode information from web socket header. **recvBufLen (2 bytes, hex):** Number of bytes received. The least significant byte is sent first. For example, 900 bytes (0x0384) would be sent as <0x84> <0x03> **ipv4_address / ipv6_address (16 bytes, hex):** Source IPv4/IPv6 address. i.e IP address from which data is received. This field is not present in the message if the data is received over a TCP socket. Only first 4 bytes are filled rest 12 bytes are zero for IPv4 address. **src_port (2 bytes, hex):** Source port. i.e port number from which data is received. This field is not present in the message if the data is received over a TCP socket. **recvDataBuf:** Actual received data stream. A maximum of 1472 bytes can be received in case of UDP and 1460 bytes in case of TCP, in this field. When the module sends data to the Host, byte stuffing is NOT done by the module. The size parameter should be used to know how many bytes of valid data is expected.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Example:
Command:
N/A

Response:
For IPV4:
If 'abcd' is sent from remote terminal to module, on an UDP socket with handle 1, from source ipv4 address 192.168.1.1 to destination ipv4 address 192.168.1.2(module address) and source port 8001, the module sends the following response to the host.
AT+RSI_READ 4 1 4 192 168 1 1 8001 abcd \r\n
0x41 0x54 0x2B 0x52 0x53 0x49 0x5F 0x52 0x45 0x41 0x44 0x04 0x00 0x01 0x00 0x04 0x00 0xC0 0xA8 0x01 0x01 0x41 0x1F 0x61 0x62 0x63 0x64 0x0D 0x0A
If 'abcd' is sent from remote terminal to module, on a TCP socket with handle 1, the module sends the following response to the host.
AT+RSI_READ 4 1 4 abcd \r\n
0x41 0x54 0x2B 0x52 0x53 0x49 0x5F 0x52 0x45 0x41 0x44 0x04 0x00 0x01 0x00 0x04 0x00 0x61 0x62 0x63 0x64 0x0D 0x0A

For IPV6:
If 'abcd' is sent from remote terminal to module, on an UDP socket with handle 1, from source ipv6 address 2001:db8:1:0:0:0:0:153to sdestinationipv6 address 2001:db8:1:0:0:0:0:154 (module address)and source port 8001, the module sends the following response to the host.
AT+RSI_READ 6 1 4 2001:db8:1:0:0:0:0:153 8001 abcd \r\n
0x41 0x54 0x2B 0x52 0x53 0x49 0x5F 0x52 0x45 0x41 0x44 0x06 0x00 0x01 0x00 0x04 0x00 0x32 0x30 0x30 0x31 0x3A 0x44 0x42 0x38 0x3A 0x31 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x31 0x35 0x33 0x41 0x1F 0x61 0x62 0x63 0x64 0x0D 0x0A

Note:
The data delivered to the Host on receiving data on a TCP socket does not include the source IP address and source port (Sip and Sport).

5.38 Wireless Firmware Upgrade

Description:
This command is sent as a response to the wireless firmware upgradation request.
When user clicks on Upgrade button on the wireless up gradation page, module sends an asynchronous message ("AT+RSI_FWUPREQ" in AT mode) to host. Upon receiving this message, host must send wireless firmware upgrade request message if upgrade is required. Host can ignore if upgrade is not required.

Command Format:
AT Mode:
at+rsi_fwupok\r\n

Command Parameters:
N/A

Response:
AT Mode:
There is no response for this command.
After successful upgradation, firmware gives a success indication with an asynchronous message as "AT+RSI_FWUPSUCCESS\r\n". Also "Firmware up gradation successful" pop-up window appears on the browser.
On firmware upgrade failure or host not responding for firmware upgrade request, module gives an error message on pop-up window: "module not responding" on the browser.

Response Parameters:
N/A

Possible Error Codes:
Possible error codes are 0x0021,0x0025,0x002C, 0x0034.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Note:
Wireless firmware upgradation is supported only for the latest versions of firefox, google chrome.

5.39 Background Scan (BG scan)

Description:
This command is to scan Access Points, when module is in connected state. The scan results are sorted in decreasing order of signal strength (RSSI value). The scanned access point with highest signal strength will be the first in the list.
Upon issuing this command, RS9116 WiSeConnect validates the Channel bit map issued through the scan command, to ensure that background scan is performed only on those channels.

Command Format:
AT Mode:
at+rsi_bgscan= < bgscan_enable >,< enable_instant_bgscan >,< bgscan_threshold >,< rssi_tolerance_threshold >, < bgscan_periodicity >, < active_scan_duration >,< passive_scan_duration >,< multi_probe >\r\n

Command Parameters:
bgscan_enable (2 bytes):
To enable/Disable bgscan
0 – Disable
1 - Enable

enable instant_bgscan (2 bytes):
If this is set to 1 then module will send probe request immediately on air and bgscan results will be given to host.

Note:
If host requires BGScan results then instant_bg_enable has to be set to 1.

bgscan_threshold (2 bytes):
This is the threshold in dBm to trigger the bgscan. After bgscan_periodicity, if connected AP RSSI falls below this value then bgscan will be triggered.

rssi_tolerance_threshold (2 bytes):
This is difference of last RSSI of connected AP and current RSSI of connected AP. Here last RSSI is the RSSI calculated at the last beacon received and current RSSI is the RSSI calculated at current beacon received.
If this difference is more than rssi_tolerance_threshold then bgscan will be triggered irrespective of periodicity.

bgscan_periodicity (2 bytes):
This is time period in seconds to trigger bgscan if RSSI of connected AP is above (assuming RSSI is positive value) the given bgscan_threshold.

active_scan_duration (2 bytes):
This is active scan duration per channel in milli seconds.

passive_scan_duration (2 bytes):
This is passive scan duration per DFS channel in 5GHz in milli seconds.

multi_probe (1 byte):
If set to one then module will send two probe request one with specific SSID provided during join command and other with NULL ssid (to scan all the access points).

Note:

1. Channel to scan in background scan is taken from Channel bit maps of scan command, e.g. channel_bit_map_2_4 and channel_bit_map_5.
2. active_scan_duration should be less than DTIM period to avoid multicast packet loss.
3. Number of channels supported in bgscan are 24 only.

Response:
AT Mode:
If instant_bg_enable is disabled:
OK\r\n
If instant_bg_enable is enabled:

Result Code	Description
OK< scanCount >< padding > < rfChannel >< securityMode >< rssiVal >< uNetworkType >< ssid >< bssid >< reserved > …….up to the number of scanned nodes	Successful execution of the command.
ERROR<Error code>	Failure

Response Parameters:
Scancount(4 byte): Number of Access Points scanned
padding(4 byte): reserved bytes.
rfChannel(1 byte): Channel Number of the scanned Access Point
securityMode(1 byte):
0–Open
1–WPA
2–WPA2
3-WEP
4–WPA Enterprise
5–WPA2 Enterprise
rssival(1 byte): RSSI of the scanned Access Point
uNetworkType(1 byte):
Network type of the scanned Access Point
1– Infrastructure mode
ssid(32 bytes): SSID of the scanned Access Point
Bssid(6 bytes): MAC address of the scanned Access Point
Reserved2(2 byte): Reserved bytes.

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C, 0x004A

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 2.

Note:

If host does not provide channel bit map and scan channel number in scan command, module will scan all channels in 2.4 GHz and all non-DFS channels in 5GHz.

Example:
AT Mode:
When instant bgscan is enabled, host will get OK response followed by the response of BG scan. Module will do back ground scanning in the configured channel given in the channel bit map or scan all channels if bitmap is not provided (all non DFS channels in 5GHz)and send the scanned result to host.
at+rsi_bgscan=1,1,10,4,10,15,20,1\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x62 0x67 0x73 0x63 0x61 0x6E 0x3D 0x31 0x2C 0x31 0x2C 0x31 0x30 0x2C 0x34 0x2C 0x31 0x30 0x2C 0x31 0x35 0x2C 0x32 0x30 0x2C 0x31 0x0D 0x0A

Response:
If two networks are found with the SSID "ap_ssid_net1" and "ap_ssid_net2", in channels 6 and 11, with measured RSSI of -20dBm and -14dBm respectively, the return value is
O K < scanCount =2> < padding > < rfChannel =0x0A> < securityMode =0x02> < rssiVal =14> < uNetworkType =0x01> < ssid =ap_ssid_net2> < bssid =0x00 0x23 0xA7 0x1F 0x1F 0x15> < reserved >< rfChannel =0x06> < securityMode =0x00> < rssiVal =20> < uNetworkType =0x01> < ssid =ap_ssid_net1> < bssid =0x00 0x23 0xA7 0x1F 0x1F 0x14> < reserved > \r\n
0x4F 0x4B 0x02 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0A 0x02 0x0D 0x01 0x52 0x65 0x64 0x70 0x69 0x6E 0x65 0x5F 0x6E 0x74 0x32 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x23 0xA7 0x1F 0x1F 0x15 0x00 0x00 0x06 0x00 0x14 0x01 0x52 0x65 0x64 0x70 0x69 0x6E 0x65 0x5F 0x6E 0x74 0x31 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x23 0xA7 0x1F 0x1F 0x14 0x00 0x00 0x0D 0x0A

When instant bgscan is disabled and When connected AP RSSI falls below -10dBm (e.g. -15, -12 etc) then bgscan will be triggered after 10 seconds period get over. But host will get only OK message here.
at+rsi_bgscan=1,0,10,4,10,15,20,1\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x62 0x67 0x73 0x63 0x61 0x6E 0x3D 0x31 0x2C 0x31 0x2C 0x31 0x30 0x2C 0x34 0x2C 0x31 0x30 0x2C 0x31 0x35 0x2C 0x32 0x30 0x2C 0x31 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.40 Roam Parameters

Description:
This command is used to enable roaming and set roaming parameters. This command can be issued any time after init command, but this command will come into action only after bgscan.

Command Format:
AT Mode:
at+rsi_roam_params= < roam_enable >,< roam_threshold >,<roam_hysteresis>\r\n

Command Parameters:
roam_enable (4 bytes):
To Enable/Disable roaming.
0 – Disable
1 - Enable

roam_threshold (4 bytes):
If connected AP RSSI falls below this then module will search for new AP from background scanned list.

roam_hysteresis (4 bytes):
If module found new AP with same configuration (SSID, Security etc) and if (connected_AP_RSSI – Selected_AP_RSSI) is greater than roam_hysteresis, then it will try to roam to the new selected AP.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of command.
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0021, 0x0025, 0x002C, 0x0026.

Relevance:
This command is relevant when the module is configured in Operating Mode 0,2.

Example:
AT Mode:
at+rsi_roam_params=1,5,2\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x72 0x6F 0x61 0x6D 0x5F 0x70 0x61 0x72 0x61 0x6D 0x73 0x3D 0x31 0x2C 0x35 0x2C 0x32 0x0D 0x0A

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

Note:

WLAN - RS9116 support Wi-Fi L2 roaming.

5.41 HT Caps

Description:

This command is used to enable HT (high throughput) Caps (capabilities) in module when operating in AP mode. This command must be issued after ap configuration parameters command.

Command Format:
AT Mode:
at+rsi_ht_caps=< mode_11n_enable > ,< ht_caps_bitmap >\r\n

Command Parameters:
mode_11n_enable (2 bytes):
Enable/Disable 11n capabilities in AP Mode.
1 - Enable
0 - Disable

ht_caps_bit_map (2 bytes):
Bit map corresponding to high throughput capabilities.

ht_caps_bit_map	Functionality	Bit set to 0	Bit set to 1	Note and Info
ht_caps_bit_map[10:15]	All set to '0'			
ht_caps_bit_map[8]	Rx STBC support	Disable	Enable	
ht_caps_bit_map[6:7]	Set to '0'			
ht_caps_bit_map[5]	short GI for 20Mhz support	Disable (short GI for 20Mhz support disabled)	Enable (short GI for 20Mhz support enabled)	
ht_caps_bit_map[4]	Green field support	Disable (Green field support disabled)	Enable (Green field support enabled)	
ht_caps_bit_map[2:3]	Set to '0'			
ht_caps_bit_map[1]	Channel Width Support	Disable(Channel Width Support disabled)	Enable(Channel Width Support enabled)	
ht_caps_bit_map[0]	Set to '0'			

Response:

AT Mode:

Result Code	Description
OK	Successful execution of command.
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0021, 0x0025, 0x002C,0x004D.

Relevance:
This command is relevant when the module is configured in Operating Mode 6.

Example:
AT Mode:
at+rsi_ht_caps=1,2\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x68 0x74 0x5F 0x63 0x61 0x70 0x73 0x3D 0x31 0x2C 0x32 0x0D 0x0A
Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.42 DNS Server

Description:
This command is used to provide the DNS server's IP address to the module. This command should be issued before the "DNS Resolution" command and after the "Set IP Parameter" command.

Command Format:
AT Mode:
For IPv4:
at+rsi_dnsserver=< DNSMode >,< primary_dns_ip.ipv4_address >,< secondary_dns_ip.ipv4_address >\r\n
For IPv6:
at+rsi_dnsserver6=< DNSMode >,< primary_dns_ip.ipv6_address >,< secondary_dns_ip.ipv6_address >\r\n

Command Parameters:
DNSMode (8 bytes):
1 - The module can obtain DNS Server IP address during the command "Set IP Params" if the DHCP server in the Access Point supports it. In such a case, value of '1' should be used if the module wants to read the DNS Server IP obtained by the module
0 - Value of '0' should be used if the user wants to specify a primary and secondary DNS server address.

primary_dns_ip.ipv4_address (4 bytes):
This is the IPv4 address of the Primary DNS server to which the DNS Resolution query will be sent. Should be set to '0' if DNSMode =1. Only first 4 bytes of ipv4 address are filled, rest 12 bytes are zero.

primary_dns_ip.ipv6_address (32 bytes):
This is the IPv6 address of the Primary DNS server to which the DNS Resolution query will be sent. Should be set to '0' if DNSMode =1.

secondary_dns_ip.ipv4_address (4 bytes):
This is the IPv4 address of the Secondary DNS server to which the DNS Resolution query will be sent.
If DNSMode =1 or if the user does not want to specify a secondary DNS Server IP address, this parameter should be set to '0'. Only first 4 bytes of ipv4 address are filled, rest 12 bytes are zero.

secondary_dns_ip .ipv6_address (32 bytes):
This is the IPv6 address of the Secondary DNS server to which the DNS Resolution query will be sent.
If DNSMode =1 or if the user does not want to specify a secondary DNS Server IP address, this parameter should be set to '0'

Response:
AT Mode:
For IPv4:

Result Code	Description
OK< primary_dns_ip.ipv4_address >< secondary_dns_ip.ipv4_address >	Successful execution of command.
ERROR<Error code>	Failure.

For IPv6:

Result Code	Description
OK< primary_dns_ip.ipv6_address >< secondary_dns_ip.ipv6_address >	Successful execution of command.
ERROR<Error code>	Failure.

Response Parameters:
primary_dns_ip.ipv4_address(16 bytes): IPv4 address of the primary DNS server. Only first 4 bytes of IPv4 address is filled, rest 12 bytes are zero.
primary_dns_ip.ipv6_address(16 bytes): IPv6 address of the primary DNS server
secondary_dns_ip.ipv4_address (16 bytes): IP address of the secondary DNS server. Only first 4 bytes of IPv4 address is filled, rest 12 bytes are zero.
secondary_dns_ip.ipv6_address (16 bytes): IPv6 address of the secondary DNS server.

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002C, 0xFFF8,0xFF74,0xBBA8,0xBBB2,0xBBAF,0xBB17,0xBBB3

Relevance:
This command is relevant in Operating Modes 0, 1, 2, 6.

Example:
AT Mode:
For IPv4:
at+rsi_dnsserver=1,0,0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x64 0x6E 0x73 0x73 0x65 0x72 0x76 0x65 0x72 0x3D 0x31 0x2C 0x30 0x2C 0x30 0x0D 0x0A

Response:
OK<primary=1.2.3.4><secondary=5.6.7.8>\r\n
0x4F 0x4B 0x01 0x02 0x03 0x04 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x05 0x06 0x07 0x08 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0D 0x0A

Command:
at+rsi_dnsserver=0,8.8.8.8,0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x64 0x6E 0x73 0x73 0x65 0x72 0x76 0x65 0x72 0x3D 0x30 0x2C 0x38 0x2E 0x38 0x2E 0x38 0x2E 0x38 0x2C 0x30 0x0D 0x0A

Response:
OK< primary=8.8.8.8><secondary =0>\r\n
0x4F 0x4B 0x08 0x08 0x08 0x08 0x00 0x00 0x00 0x00 0x0D 0x0A

For IPv6:
at+rsi_dnsserver6=1,0,0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x64 0x6E 0x73 0x73 0x65 0x72 0x76 0x65 0x72 0x36 0x3D 0x31 0x2C 0x30 0x2C 0x30 0x0D 0x0A

Response:
OK< primary v6 address=2001:db8:1:0:0:0:0:2>< secondary v6 address = 2001:db8:1:0:0:0:0:3>\r\n
0x4F 0x4B 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x02 0x00 0x00 0x00 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x03 0x00 0x00 0x00 0x0D 0x0A

Command:
at+rsi_dnsserver6=0,2001:DB8:1:0:0:0:0:5,0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x64 0x6E 0x73 0x73 0x65 0x72 0x76 0x65 0x72 0x36 0x3D 0x30 0x2C 0x32 0x30 0x30 0x31 0x3A 0x44 0x42 0x38 0x3A 0x31 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x35 0x2C 0x30 0x0D 0x0A

Response:
OK< primary v6 address = 2001:DB8:1:0:0:0:0:5 ><secondary v6 address=0>\r\n
0x4F 0x4B 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x05 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0D 0x0A

5.43 DNS Resolution

Description:
This command is to obtain the IP address of the specified domain name.

Command Format:
AT Mode:
For IPv4:
at+rsi_dnsget=< aDomainName >,< uDNSServerNumber >\r\n

For IPv6:
at+rsi_dnsget6=< aDomainName >,< uDNSServerNumber >\r\n

Command Parameters:

aDomainName (90 bytes):
This is the domain name of the target website. A maximum of 90 characters is allowed.

uDNSServerNumber (2 bytes):
Reserved.

Response:
AT Mode:

For IPv4:

Result Code	Description
OK< ip_version >< uIPCount >< aIPaddr.ipv4_address >..repeats for 10 times	Successful execution of command
ERROR<Error code>	Failure.

For IPv6:

Result Code	Description
OK< ip_version >< uIPCount >< aIPaddr.ipv6_address >..repeats for 10 times	Successful execution of command
ERROR<Error code>	Failure.

Response Parameters:

ip_version(2 bytes):
IP version used , either 4 or 6.

uIPCount(2 bytes):
Number of IP addresses resolvedaIPaddr.ipv4_address(16 bytes): Individual IPv4 addresses, up to a maximum of 10. Only first 4 bytes are filled in ipv4 address, rest 12 bytes are zero.

aIPaddr.ipv6_address(16 bytes):
Individual IPv6 addresses, up to a maximum of 10.

Note:
Maximum timeout for DNS resolution is 120 seconds.

Possible error codes:
Possible error codes are 0x0015,0x0021, 0x0025, 0x002C,0xFFBB,0xBBA1,0xBBAA,0xBBA3,0xBBA4,0xBBAC

Relevance:
This command is relevant in Operating Modes 0, 1, 2 and 6.

Example:
AT Mode:
For IPv4:
at+rsi_dnsget=www.silabs.com,1\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x64 0x6E 0x73 0x67 0x 65 0x74 0x3D 0x77 0x77 0x77 0x2E 0x72 0x65 0x64 0x70 0x69 0x6E 0x65 0x73 0x69 0x67 0x6E 0x61 0x6C 0x73 0x2E 0x63 0x6F 0x6D 0x2C 0x31 0x0D 0x0A

Response:
OK<num_IPAddr=1><IPAddr1=201.168.1.100>\r\n
0x4F 0x4B 0x01 0x00 0xC9 0xA8 0x01 0x64 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0D 0x0A

For IPv6:
at+rsi_dnsget6=www.silabs.com,1\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x64 0x6E 0x73 0x67 0x 65 0x74 0x36 0x3D 0x77 0x77 0x77 0x2E 0x72 0x65 0x64 0x70 0x69 0x6E 0x65 0x73 0x69 0x67 0x6E 0x61 0x6C 0x73 0x2E 0x63 0x6F 0x6D 0x2C 0x31 0x0D 0x0A

Response:
OK<num_IPAddr=1><IPv6Addr1=2001:DB8:1:0:0:0:0:0:6>\r\n
0x4F 0x4B 0x01 0x00 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x06 0x00 0x00 0x00 0x0D 0x0A

5.44 DNS UPDATE

Description:
This command is to update client host name (type A record) in DNS server.

Command Format:

AT Mode:
at+rsi_dnsupdate=< ipversion >,< aZoneName >,< aHostName >,<uDNSServerNumber >,< ttl >\r\n

Command Parameters:

ip_version:
IP version used 4 (No support for ip version 6).

aZoneName:
This is the zone name of the Domain.
A maximum of 31 characters is allowed.

aHostName:
This is the host name of the Domain.
A maximum of 31 characters is allowed.

uDNSServerNumber (2 bytes):
DNS Server number.

ttl:
Time to Live, Time in seconds for which hostname should be active.

Response:

AT Mode:

Result Code	Description
OK	Successful execution of command
ERROR<Error code> .	Failure

Command Parameters:

aZoneName:
This is the zone name of the domain.
A maximum of 31 characters is allowed.

aHostName:
This is the host name of the domain.
A maximum of 31 characters is allowed.

uDNSServerNumber(2 bytes):
DNS Server number.

ttl(2 bytes):
Time To Live, Time in seconds for which service should be active.

Possible error codes:
Possible error codes are 0x0015,0x0021, 0x0025,0x002C,0xFFBB,0xBBA1,0xBBAA,0xBBA3,0xBBA4,0xBBAC

Relevance:
This command is relevant in Operating Modes 0, 1, 2 and 6.

Example:

AT Mode:
at+rsi_dnsupdate=4,RPS,silabs,1,53 \r\n

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.45 HTTP Get

Description:
This command is used to transmit HTTP GET request from the module to a remote HTTP server.
A subsequent HTTP GET request can be issued only after receiving the response of the previously issued HTTP GET request. Module acts as a HTTP client when this command is issued.

Command Format:
AT Mode:

For IPv4:
at+rsi_httpget=< https_enable >,< http_port >,< User_name >,< Password >,< Host_name >,< Ip_address >,< url >,< Extended_header >\r\n

For IPv6:
at+rsi_httpget6=< https_enable >,< http_port >,< User_name >,< Password >,< Host_name >,< Ip_address >,< url >,< Extended_header >\r\n

Command Parameters:
https_enable:

Set BIT(0) to enable HTTPS.
Set BIT(1) to enable NULL delimiter for HTTP buffer instead of comma
Set BIT(2) to use SSL TLS 1.0 version if HTTPS is enabled.
Set BIT(3) to use SSL TLS 1.2 version if HTTPS is enabled.
Set BIT(4) to use SSL TLS 1.1 version if HTTPS is enabled.

Note:
If SSL is enabled by default, it will use SSL TLS 1.0 and TLS 1.2 version. BIT(2) and BIT(3) are valid only when HTTPS is enabled.

Note:
If SSL is enabled module will use same SSL version until module reboots.

Set BIT(5) to enable http_post data feature

Set BIT(6) to enable HTTP version 1.1.

Set BIT(7) to enable user defined http_content type in Flags.

http_port:
HTTP server port number.
If this is not mentioned, default port numbers 80(IPV4),8080(IPV6) will be used.

User_name:
User_name for HTTP/HTTPS server authentication. Default username is 'admin'.

Password:
Password for HTTP/HTTPS server authentication. Default password is 'admin'.

Host_name:
Host name of the HTTP/HTTPS server.

Ip_address:
IPv4/IPv6 address of HTTP/HTTPS server.

url: requested URL.

Extended_header:
The Purpose of this is to append user configurable header fields to the default HTTP/HTTPS header. To write extended header through 'at' command, user must use 'Data stuffing' mentioned separately in this document as well mentioning here also in context of extended header. extended header can have multiple header fields each ended by \r\n(oxd oxa) but here \r\n is our delimiter for whole 'at' command so use data stuffing and replace all \r\n(0xd 0xa) by 0xdb oxdc besides delimiter(\r\n).follow example given below.

default http header contains the following:

Default http header include:

1.Contentent - Type: It describes about the type of the file which user want to send like html, txt, gif etc**..**

2.Content - Length: It tells about the length of the text which User is using.

The above two mentioned fields are created in the header by the firmware.

Note:

1. Maximum supported length for User_name, Password together is 278 bytes.
2. Maximum supported length for Buffer is (872- (length of User_name+length of Password)) bytes excluding delimiters.
3. If username, password, hostname and extended http headers are not required, user should send empty string separated by delimiter.
4. If content of any field contains comma (,) then NULL delimiter should be used.
5. Host needs to do byte stuffing in extended header field. Please refer the note **Data Stuffing.**

For example,
Https_enable = BIT(0)
http_port = 443
Username : username
Password : password
Hostname: www.google.com
IP = 192.168.40.50
URL=/index.html
Extended HTTP Header = ContentType: **html**ÛÜ

Response:
Module may give http response in multiple chunks for a single HTTP GET request.

AT Mode:

Result Code	Description
AT+RSI_HTTPRSP=< more ><status_code>< offset >< data_len >< data>	After the module sends out the HTTP GET request to the remote server, it may take some time for the response to come back. The response from the remote server is sent out to the Host from the module in the following form: AT+RSI_HTTPRSP=<More><Status_Code><Data Offset><Data Length><Data> The string AT+RSI_HTTPRSP is in uppercase ASCII.
ERROR<Error_code>	Failure

Response Parameters:
More(2 bytes):
This indicates whether more HTTP data for the HTTP GET request is pending or not.
0–More data is pending. Further interrupts may be raised by the module till all the data is transferred to the Host.
1– End of HTTP data.

Status code(2 bytes):
Provided the HTTP status code as received in the response patcket such as 200, 201, 404 etc. A status_code equal to 0 indicates that there was no HTTP header in the received packet, probalby a continuation of the frame body received in the previous chunk.

Offset(4 bytes): Always contains '0'.

data_len(4 bytes): data length in current chunk.

Data(Maximum 1400 bytes): Actual http data.

Possible error codes:
Possible error codes for this command are 0x0015,0x0021, 0x0025, 0x002C, 0xFF74, 0xBBF0

Relevance:
This command is relevant when the module is configured in Operating Mode 0,1, 2 and 6.

Example:
AT Mode:

For IPv4:
Https_enable : 0
http_port : 80
Username : username
Password : password
Host_name : www.google.com
IP Address: 192.168.40.86
URL: /index.html
Extended HTTP Header: ContentType: **html**ÛÜ
the below command is used
at+rsi_httpget=0,80,username,password,hostname,192.168.40.86,/index.html,ContentType: **html**ÛÜ\r\n

For IPv6:
Https_enable : = 0, to disable https
Http_port :8080
Username : username
Password : password
Hostname :www.google.com
IP Address: 2001:DB8:1:0:0:0:0:4
URL: /index.html
Extended HTTP Hedaer: ContentType:**html**ÛÜ
the below command is used
at+rsi_httpget6=0,8080,username,password,hostname,2001:DB8:1:0:0:0:0:4,/index.html,ContentType: **html**ÛÜ\r\n

5.46 HTTP Post

Description:
This command is used to transmit HTTP POST request to a remote HTTP server.
A subsequent HTTP POST request can be issued only the response to a previously issued HTTP POST request is received. Module acts as a HTTP client when this command is issued.

Command Format:
AT Mode:
For IPv4:
at+rsi_httppost=< https_enable >,< http_port >,< User_name >,< Password >,< Host_name >,< Ip_address >,< url >,< Extended_header >,< Data/Data_length >\r\n

For IPv6:
at+rsi_httppost6=< https_enable >,< http_port >,< User_name >,< Password >,< Host_name >,< Ip_address >,< url >,< Extended_header >,< Data/Data_length >\r\n

Command Parameters:
https_enable:
Set BIT(0) to enable HTTPS.
Set BIT(1) to enable NULL delimiter for HTTP buffer instead of comma
Set BIT(2) to use SSL TLS 1.0 version.
Set BIT(3) to use SSL TLS 1.2 version.
Set BIT(4) to use SSL TLS 1.1 version.

Note:
If SSL is enabled by default, it will use SSL TLS 1.0 and TLS 1.2 version.
BIT(2) and BIT(3) are valid only when HTTPS is enabled.

Note:
If SSL is enabled, module will use same SSL version until module reboots.

Set BIT(5) to enable HTTP post data feature.
Set BIT(6) to enable HTTP version 1.1.

Set BIT(7) to enable user defined http_content type in Flags.

http_port:
HTTP server port number.
If this is not mentioned default port number 80 will be used.

User_name: User_name for HTTP/HTTPS server authentication. Default username is 'redpine'.

Password: Password for HTTP/HTTPS server authentication. Default password is 'admin'.

Host_name: Host name of the HTTP/HTTPS server.

Ip_address: IPv4/IPv6 address of HTTP/HTTPS server.

url: requested URL.

Extended_header: The purpose of this is to append user configurable header fields to the default HTTP/HTTPS header. To write extended header through 'at' command, user must use 'Data stuffing' mentioned separately in this document as well as in context of extended header mentioned here. Extended header can have multiple header fields each ended by \r\n(oxd oxa). But here \r\n is our delimiter for whole 'at' command, so use data stuffing and replace all \r\n(0xd 0xa) by 0xdb oxdc besides delimiter(\r\n)

Data: Post data to be sent.

Data/Data_length: Post data to be sent/ Total length of the http data.

Note:

1. When BIT(6) is enabled in https_enable feature bitmap, hostname is mandatory (To support HTTP version 1.1).
2. When BIT(5) is enabled in https_enable feature bitmap, instead of Data, host need to give total HTTP data length.
3. Maximum supported length for User_name, Password together is 278 bytes.
4. Maximum supported length for Buffer is (872- (length of User_name + length of Password)) bytes excluding delimiters.
5. If username, password, hostname and extended http headers are not required, user should send empty string separated by delimiter.
6. If content of any field contains comma (,) then NULL delimiter should be used.

Example 1:
Https_enable = 0
Http_port = 80
Username = username
Password = password
Hostname = www.google.com
IP = 192.168.40.50
URL=/index.html
Extended HTTP Header = ContentType:**html**ÛÜ
Data=<data>

Example 2 (Set BIT(5) in https_enable field):
Https_enable = 32
Http_port = 80
Username = username
Password = password
Hostname = posttestserver.com
IP = 64.90.48.15
URL=/post.php
Extended HTTP Header = ContentType: **html**ÛÜ
Data_length= 1800

Response:
Module may give http response in multiple chunks for a single HTTP POST request.

AT Mode:

Result Code	Description
AT+RSI_HTTPRSP=< more ><status_code>< offset >< data_len >< data >	After the module sends out the HTTP POST request to the remote server, it might take some time to receive the response. The response from the remote server is sent out to the Host from the module in the following form: AT+RSI_HTTPRSP=< more ><status_code>< offset >< data_len >< data > The string AT+RSI_HTTPRSP is in uppercase ASCII.
ERROR<Error_code>	Failure

Response Parameters:
More(2 bytes): This indicates whether more HTTP data for the HTTP POST request is pending or not.
0–More data is pending. Further interrupts may be raised by the module till all the data is transferred to the Host.
1– End of HTTP data.
2– HTTP post request success response.

Status code(2 bytes):
Provided the HTTP status code as received in the response patcket such as 200, 201, 404 etc. A status_code equal to 0 indicates that there was no HTTP header in the received packet, probably a continuation of the frame body received in the previous chunk.

Offset(4 bytes): Always contains '0'.

data_len(4 bytes): data length in current chunk.

Data(Maximum 1400 bytes): Actual http data.

Possible error codes:
Possible error codes for this command are 0x0015,0x0021, 0x0025, 0x002C, 0xFF74, 0xBBF0

Relevance:
This command is relevant when the module is configured in Operating Mode 0,1,2 and 6 .

Example 1:
AT Mode:
For IPv4:
at+rsi_httppost=1,443,username,password,hostname,192.168.40.86,/index.html,
ContentType:**html**ÛÜ,<data=abcd>\r\n

For IPv6:
at+rsi_httppost6=0,443,username,password,hostname,2001:DB8:1:0:0:0:0:4,/index.html,
ContentType:**html**ÛÜ,<data=abcd>\r\n

Example 2:

AT Mode:

For IPv4:
at+rsi_httppost=32,80,username,password,posttestserver.com,64.90.48.15,/index.html,
ContentType: **html**ÛÜ, 100\r\n

at+rsi_httppost=65,443,username,password,posttestserver.com,64.90.48.15,/index.html,
ContentType: **html**ÛÜ, 100\r\n

at+rsi_httppost=97,443,username,password,posttestserver.com,64.90.48.15,/index.html,
ContentType: **html**ÛÜ, 100\r\n

For IPv6:
at+rsi_httppost6=32,80,username,password,hostname,2001:DB8:1:0:0:0:0:4,/index.html,
ContentType: **html**ÛÜ, 100\r\n

at+rsi_httppost6=65,443,username,password,hostname,2001:DB8:1:0:0:0:0:4,/index.html,
ContentType: **html**ÛÜ, 100\r\n

at+rsi_httppost6=97,443,username,password,hostname,2001:DB8:1:0:0:0:0:4,/index.html,
ContentType: **html**ÛÜ, 100\r\n

5.47 HTTP Post Data

Description:
This command is used to transmit HTTP POST request to a remote HTTP server. A subsequent HTTP POST request can be issued only when the response to a previously issued HTTP POST request is received. Module acts as a HTTP client when this command is issued.

Command Format:

AT Mode:
at+rsi_httppost_data=< current_chunk_length >,< Data>\r\n

current_chunk_length: Length of the current data.

Data: HTTP data.

Note:
httppost_data command is valid only when BIT(5) is enabled in the https_enable feature bitmap in HTTP POST command.

Response:

Module may give http response in multiple chunks for a single HTTP POST request.

AT Mode:

Result Code	Description
AT+RSI_HTTPPOSTRSP= <more ><status_code>< offset >< data_len >< data >	After the module sends out the HTTP POST request to the remote server, it might take some time to receive the response. The response from the remote server is sent out to the Host from the module in the following form: AT+RSI_HTTPRSP=<More><Status_Code><Data Offset><Data Length><Data> The string AT+RSI_HTTPRSP is in uppercase ASCII.
ERROR<Error_code>	Failure

Response Parameters:

More(2 bytes): This indicates whether more HTTP data for the HTTP POST request is pending or not .

4- More data is pending from host
5-End of HTTP data content length
8–More data is pending. Further interrupts may be raised by the module till all the data is transferred to the Host.
9– End of HTTP data from server.

Status code(2 bytes):
Provided the HTTP status code as received in the response patcket such as 200, 201, 404 etc. A status_code equal to 0 indicates that there was no HTTP header in the received packet, probably a continuation of the frame body received in the previous chunk.

Offset(4 bytes): Always contains „0?.

data_len(4 bytes): data length in current chunk.

Data(Maximum 1400 bytes): Actual http data.

Possible error codes:
Possible error codes for this command are as follows,
0x0015,0x0021, 0x0025, 0x002C, 0xFF74, 0xBBF0, 0XBB38, 0xBBEF, 0xBB3E, 0xBB38, 0xBBE7.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

5.48 HTTP Put

Description:
This command is used to transmit HTTP PUT request to a remote HTTP server. Module acts as a HTTP client when this command is issued.
This section explains different commands to use HTTP client PUT.
This command should be given only after **Set IP Parameters** command.
Following table explains list of HTTP PUT commands and their description.

HTTP PUT Command	Description
PUT_CREATE	Creates HTTP client thread and HTTP client socket. This should be the first command to use the HTTP client PUT
PUT_START	Connects to the specified HTTP server and creates the specified resource.
PUT_PACKET	To send the resource data packet
PUT_DELETE	To delete the HTTP client thread and socket

- **PUT_CREATE** should be called as a first command to use HTTP PUT.
- Once put start is successful, **PUT_PACKET** should be called to send the resource data packet for the previously create resource.
- Once put create is successful, **PUT_START** should be called to create the specified resource on the specified HTTP server.
- Call **PUT_DELETE** to delete HTTP Client thread and socket.

AT Mode:
HTTP PUT client has different command types. Based on the command type, following parameters change accordingly.
at+rsi_httpput=<command_type>,<remaining parameters>\r\n
Following are available command types.

HTTP PUT command	Command Type	Command Format
PUT create	**1**	at+rsi_httpput=1\r\n
PUT start	**2**	at+rsi_httpput=2,<ip_version>,<https_enable>,<port number>,<total contentlength>,<http buffer>\r\n Ip_version: IP version to use. 4 – For IPv4 6 – For IPv6 **ip_address:** Ipv4/Ipv6 address of the HTTP server. **port number:** HTTP server port number **total_length:** Total content length of the HTTP PUT data **http_buffer:** http buffer contains the username, password, host name, ip address, URL address, extended http header.
PUT PACKET	**3**	at+rsi_httpput=3,<current length>,<http buffer>\r\n **current length :** length of the current http put content chunk **http_buffer:** http buffer contains the put data
PUT DELETE	**4**	at+rsi_httpput=4\r\n

Note:
1. Maximum supported PUT buffer length is 900 bytes.
2. Maximum timeout for http put start command response is 20 Seconds.
3. Maximum timeout for http put packet command response is 10 Seconds.

https_enable:
Set BIT(0) to enable HTTPS.
Set BIT(1) to enable NULL delimiter for HTTP buffer instead of comma
Set BIT(2) to use SSL TLS 1.0 version.
Set BIT(3) to use SSL TLS 1.2 version.
Set BIT(4) to use SSL TLS 1.1 version

Note:
If SSL is enabled by default, it uses SSL TLS 1.0 and TLS 1.2 versions.
BIT(2) and BIT(3) are valid only when HTTPS is enabled.

Note:
If SSL is enabled, module uses same SSL version until module reboots.

Set BIT(5) to enable http_post data feature

Set BIT(6) to enable HTTP version 1.1.

Set BIT(7) to enable user defined http_content_type in flags.

port_number:
HTTP server port number.
If this is not mentioned default port number 80 will be used.

http_put_buffer:
This field contains following values in the order of
< User_name >,< Password >,< Host_name >,< Ip_address >,< url >, < Extended_header >.
The parameter Buffer is a character buffer.

If BIT(1) is not set in https_enable field
< User_name >,< Password >,< Host_name >,< Ip_address >,< url >, < Extended_header >, < Extended_header > fields should be delimited with comma(,)

If BIT(1) is set in https_enable field
< User_name >'\0'< Password >'\0'< Host_name >'\0'< Ip_address >'\0'<URL>'\0' < Extended_header > fields should be delimited with NULL('\0')

content_length: Total length of the resource content.

User_name: User_name for HTTP/HTTPS server authentication. Default user name is 'redpine'.

Password: Password for HTTP/HTTPS server authentication. Default password is 'admin'.

Host_name: Host name of the HTTP/HTTPS server.

Ip_address: IPv4/IPv6 address of HTTP/HTTPS server.

url: requested URL.

Extended_header: The purpose of this is to append user configurable header fields to the default HTTP/HTTPS header. To write extended header through 'at' command, user must use 'Data stuffing' mentioned separately in this document as well as in context of extended header mentioned here. Extended header can have multiple header fields each ended by \r\n(oxd oxa). But here \r\n is our delimiter for whole 'at' command, so use data stuffing and replace all \r\n(0xd 0xa) by 0xdb oxdc besides delimiter(\r\n)

current_length: Current content chunk length

data: resource data to be send (This parameter is valid only for PUT PACKET command).

Note:

1. Maximum supported length for User_name, Password together is 278 bytes.
2. Maximum supported length for Buffer is (872 S- (length of User_name+length of Password)) bytes excluding delimiters.

3. If username, password, hostname and extended http headers are not required, user should send empty string separated by delimiters.
4. If content of any field contains comma (,) then NULL delimiter should be used.
5. When BIT(6) is enabled in https_enable feature bitmap, hostname is mandatory (To support HTTP version 1.1).

AT Mode:

Result Code	Description
AT+RSI_HTTPRSP= < command_type>< end_of_file > offset>< data_len >< data >	After the module sends out the HTTP PUT request to the remote server, it may take some time for the response to come back. The response from the remote server is sent out to the Host from the module in the following form: AT+RSI_HTTPRSP=<command_type><end_of_file > Offset><DataLength><Data> The string AT+RSI_HTTPRSP is in uppercase ASCII.
ERROR<Error_code>	Failure

Response Parameters:

command_type:
"5", HTTP Client PUT packet command type during the response from the server.

end_of_file (Valid for HTTP PUT PKT):
End of file or HTTP resource content.

− 8 - More data pending from server
− 9 - End of HTTP file from server/resource content

Offset(4 bytes): Always contains '0'.

Data_length:
data_len data length in current chunk

Data:
(Maximum 1024 bytes): Actual http data from server.

Possible Error Codes:
Possible Error codes for this command are 0x0021, 0x0015, 0x0025, 0xBB38.

Relevance:
This command is relevant in Operating Modes 0, 1, 2, 6.

Example:
AT Mode:
HTTP client PUT Service:
at+rsi_httpput=1\r\n
at+rsi_httpput=2,4,0,80,19,username,password,192.168.1.100,192.168.1.100,/,ContentType: **html**ÛÜ\r\n
at+rsi_httpput=3,19,<html><body></html>\r\n
at+rsi_httpput=4\r\n

5.49 DFS Client (802.11h)

Description:
DFS client implementation is internal to the module. In 5 GHz band, the module does only passive scan in DFS channels. If the Access Point detects radar signals, it indicates the module (client) to switch to a different channel by using the "channel switch frame". The module performs channel switch as per the AP's channel switch parameters. There is no command required to enable this feature, it is enabled by default.

5.50 Query Firmware Version

Description:
This command is used to query the version of the firmware loaded in the module.

Command Format:
AT Mode:
at+rsi_fwversion?\r\n

Response:
AT Mode:

Result Code	Description
OK< fwversion ><\0>	Successful execution of command. All values returned in ASCII.
ERROR<Error code>	Failure.

Response Parameters:
Fwversion: Firmware version.

Possible error codes:
Possible error codes for this command are 0x002c.

Relevance:
This command is relevant in all modes.

Example:
AT Mode:
at+rsi_fwversion?\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x66 0x77 0x76 0x65 0x72 0x73 0x69 0x6F 0x6E 0x3F 0x0D 0x0A
Response:
OK 1.1.0\0\r\n
0x4F 0x4B 0x31 0x2E 0x31 0x2E 0x30 0X00 0x0D 0x0A

Note:

The Format of Firmware version response will be changed from RS9116.NB0.WC.GENR.OSI.2.x releases.

Response:

OK1610.1.2.24.0014 or OK1610.2.0.0.0014 (for 2.x)

5.51 Query RSSI Value

Description:
This command is used to retrieve the RSSI value for Access Point to which the module is connected.

Command Format:
AT Mode:
at+rsi_rssi?\r\n

Response:
AT Mode:

Result Code	Description
OK< rssiVal >	Successful execution of command
ERROR<Error code>	Failure.

Response Parameters:

rssiVal(2 bytes) : RSSI value (-n dBm) of the Access Point to which the module is connected. It returns absolute value of the RSSI. For example, if the RSSI is -20dBm, then 20 is returned.

Possible error codes:
Possible error codes for this command are 0x0021, 0x0025, 0x002C.

Relevance:
This command is relevant when the module is configured in Operating Mode 0 and 2.

Note:
Closer the RSSI value to '0', stronger the signal strength.

Example:
AT Mode:
at+rsi_rssi?\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x72 0x73 0x73 0x69 0x3F 0x0D 0x0A

Response:
For a RSSI of -20dBm, the return string is
OK < rssiVal =-20> \r\n
0x4F 0x4B 0x14 0x0D 0x0A

5.52 Query MAC Address

Description:
This command is used to query the MAC address of the module. This command can be issued anytime after init command.

Command Format:
AT Mode:
at+rsi_mac?\r\n

Response:
For None Concurrent Mode:
AT Mode:

Result Code	Description
OK< macAddress >	Successful execution of command.
ERROR<Error code>	Failure.

Response Parameters:
macAddress(6 bytes): MAC address of the module.

Possible error codes:
Possible error codes for this command are 0x002c .

Relevance:
This command is relevant in all operating modes.

Example:
AT Mode:
at+rsi_mac?\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6D 0x61 0x63 0x3F 0x0D 0x0A

Response:
If the MAC ID is 0x00 0x23 0xA7 0x1B 0x8D 0x31, then the response is
OK< macAddress >\r\n
0x4F 0x4B 0x00 0x23 0xA7 0x1B 0x8D 0x31 0x0D 0x0A

For Concurrent Mode:
Response:
AT Mode:

Result Code	Description
OK<macAddress 1> <macAddress2>	Successful execution of command.
ERROR<Error code>	Failure.

Response Parameters:
macAddress1(6 bytes), macAddress2(6 bytes): MAC address of the module for two interfaces.

Possible error codes:
Possible error codes for this command are 0x002c .

Relevance:
This command is relevant in all operating modes.

Example:
AT Mode:
at+rsi_mac?\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6D 0x61 0x63 0x3F 0x0D 0x0A

Response:
OK< macAddress1 >< macAddress2 >\r\n
0x4F 0x4B 0x00 0x23 0xA7 0x1B 0x8D 0x31 0x00 0x23 0xA7 0x1B 0x8D 0x32 0x0D 0x0A

5.53 Query Network Parameters

Description:
This command is used to retrieve the WLAN and IP configuration parameters. This command should be sent only after the connection to an Access Point is successful.

Command Format:
AT Mode:
at+rsi_nwparams?\r\n

Response:
AT Mode:

Result Code	Description
OK< wlan_state >< Chn_no >< psk >< mac_addr >< ssid >< connType >< sec_type >< dhcpMode >< ipaddr >< subnetMask >< gateway >< num_open_socks >< prefix_length >< ipv6addr >< defaultgw6>< tcp_stack_used >[< sock_id >< socket_type >< sPort >< dPort >< destIPaddr.ipv4_address/ destIPaddr.ipv6_address >] upto Max sockets supported.	Successful execution of command
ERROR<Error Code>	Failure.

Response Parameters:
wlan_state(1 byte):
This indicates whether the module is connected to an Access Point or not.
0 – Not Connected
1 – Connected

Chn_no(1 byte):
Channel number of the AP to which the module joined

Psk(64 bytes):
Pre-shared key used

Mac_Addr(6 bytes):
MAC address of the module

ssid(32 bytes):
This value is the SSID of the Access Point to which the module is connected

ConnType(2 byte):
0x0001 – Infrastructure

Sec_type(1 byte):
Security mode of the AP to which the module joined.
0– Open mode
1– WPA security
2– WPA2 security
3- WEP
4– WPA-Enterprise
5– WPA2-Enterprise

Ipaddr(4 bytes):
This is the IP Address of the module.

SubnetMask(4 bytes):
This is the Subnet Mask of the module

Gateway(4 bytes):
This is the Gateway Address of the module.

dhcpMode(1 byte):
This value indicates whether the module is configured for DHCP or Manual IP configuration for both IPv4 and IPv6.

dhcp_mode (0 bit)-
0 – Static IP configuration for IPv4
1 – Dynamic IP configuration for IPv4

dhcp_mode (1 bit)-
0 – Static IP configuration for IPv6
1 – Dynamic IP configuration for IPv6

Num_open_socket(2 bytes):
This value indicates the number of sockets currently opened

Prefix_length(2 bytes):
Prefix length of IPv6 address.

Ipv6addr(16 bytes):
This is the IPv6 Address of the module

Defaultgw6(16 bytes):
This is the IPv6 address of the default router

tcp_stack_used(1 byte):
TCP/IP stack used
1-IPv4
2-IPv6
4-Both IPv4 and IPv6 (dual stack).

Sock_id(2 bytes):
Socket handle of an existing socket

Socket_type(2 bytes):
Type of socket

0–TCP/SSL/websocket Client
2– TCP/SSL Server (Listening TCP)
4– Listening UDP

Sport(2 bytes):
Port number of the socket in the module.

Dport(2 bytes):
Destination port number of the remote peer.

destIPaddr.ipv4_address(16 bytes):
IPv4 address of the remote terminal.
Only first 4 bytes of ipv4 address gets filled, remaining 12 bytes are zero.

destIPaddr.ipv6_address(16 bytes):
IPv6 address of the remote terminal

If the Set IP Params command was not sent to the module before Query Network Parameters command, the module returns a default values for following fields which should be ignored are outlined below:
dhcpMode, ipaddr, subnetMask, gateway, num_open_socks, prefix_length, ipv6addr, defaultgw6, tcp_stack_used, socket_info

Possible error codes:
Possible error codes for this command are 0x0021, 0x002C.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2.

Example:
AT Mode:

at+rsi_nwparams?\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6E 0x77 0x70 0x61 0x6D 0x73 0x3F 0x0D 0x0A

Response:
Response when nwparams command was given before ipconfig command so ipparameters are not set in this response, and sockets are not opened yet. Here SSID is 'cisco' and gateway is '192.168.100.76'.

Note:
Command will not get PSK, it will get PMK to reduce connection time with AP from the next boot up. In the nwparams response structure you can find 32 bytes of PMK remaining are filled with Zeros.

OK< wlan_state >< Chn_no >< psk >< mac_addr >< ssid >< connType >< sec_type >< dhcpMode >< ipaddr >< subnetMask >< gateway >< num_open_socks >< prefix_length >< ipv6addr >< defaultgw6>< tcp_stack_used >[< sock_id >< socket_type >< sPort >< dPort >< destIPaddr.ipv4_address/ destIPaddr.ipv6_address >]
OK< wlan_state =0x01>< Chn_no =0x06>< psk =0x00(repeats 63 times)>< mac_addr =0x00 0x23 0xA7 0x16 0x16 0x16>< ssid =0x63 0x69 0x73 0x63 0x6F 0x00<repeats 27 times>< connType =0x01 0x00>< sec_type =0x00 > < dhcpMode =0x01>< ipaddr =0x00 0x00 0x00 0x00 >< subnetMask =0xFF 0xFF 0xFF 0x00 0>< gateway =0xC0 0xA8 0x64 0x4C>< num_open_socks =0x00 0x00>< prefix_length =0x00 0x00>< ipv6addr =0x00(16times)>< defaultgw6=0x00(16 times) > [<sock_id =0x00 0x00>< socket_type =0x00 0x00>< sPort =0x00 0x00>< dPort =0x00 0x00>< destIPaddr.ipv4_address/ destIPaddr.ipv6_address =0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00>]<repeats 11 times > 0x0D 0x0A.

5.54 Soft Reset

Description:
This command acts as a hard reset to the module. The module will reset all information regarding the WLAN connection and IP configuration after receiving this command. The Host has to start right from the beginning, from Auto Baud Rate Detection (ABRD) for device detection and issuing the first command "Set Operating Mode". This command is valid only in case of UART and USB-CDC interface.

Note:
No response comes for this command. All the frame exchanges will follow as in a normal bootup sequence.

Command Format:
AT Mode:
at+rsi_reset\r\n

Command Parameters:
N/A

Response:
None

Response Parameters:
N/A

Possible error codes:
N/A

Relevance:
This command is relevant in all operating modes.

Example:
AT Mode:
at+rsi_reset\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x72 0x65 0x73 0x65 0x74 0x0D 0x0A

Response:
None

5.55 Set/Reset Multicast Filter

Description:
This command is to set/reset the multicast MAC address bitmap to filter multicast packets. This command should be given after init command.

Payload:
AT Mode:
at+rsi_multicast_filter=< uMcastBitMapFrame >\r\n

Payload Parameters:
uMcastBitMapFrame:
There are two bytes in the payload which represent 2 parts. Lower byte represent the command type (cmd as mentioned below) and higher byte is the hash value (6 Bits) generated from the desired multicast MAC address (48 Bits) using hash function.

uMcastBitMapFrame	Functionality
uMcastBitMapFrame[0:1]	These 2 bits represents the command type. Possible values are as follows, 1. RSI_ MULTICAST_MAC_ADD_BIT (To set particular bit in multicast bitmap) 2. RSI_ MULTICAST_MAC_CLEAR_BIT (To reset particular bit in multicast bitmap) 3. RSI_ MULTICAST_MAC_CLEAR_ALL (To clear all the bits in multicast bitmap) 4. RSI_ MULTICAST_MAC_SET_ALL (To set all the bits in multicast bitmap)
uMcastBitMapFrame[2:7]	reserved
uMcastBitMapFrame[8:13]	6bit hash value generated from the hash algorithm which corresponds to the multicast MAC address is used to set/reset corresponding bit in multicast filter bitmap. This field is valid only if 0 or 1 is selected in command type (uMcastBitMapFrame[0:1])
uMcastBitMapFrame[14:15]	reserved

Response:

AT Mode:

Result Code	Description
OK	Successful execution of command.
ERROR<Error code>	Failure.

Response Parameters:

N/A

Possible error codes:
Possible error codes are 0x0021, 0x0025, 0x002c.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

Note:
The Hash function is 8 CRC generator with the 2 MSB's ignored.

5.56 Join or Leave Multicast Group

Description:
This command is used to join or leave a multicast group.

Command Format:
AT Mode:
For IPv4:
at+rsi_multicast=< req_type >,< group_address. ipv4_address >\r\n

For IPv6:
at+rsi_multicast6=< req_type >,< group_address. Ipv6_address >\r\n

Command Parameters:

req_type:
Request type i.e. join request or leave request
0 - Leave multicast group
1 - Join multicast group

group_address.ipv4_address/ group_address.ipv6_address:
IPv4/IPv6 address of multicast group. Last 12 bytes are set to '0' in case of IPv4.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of command.
ERROR<Error code>	Failure.

Response Parameters:

N/A

Possible error codes:
Possible error codes 0x0021, 0x0025, 0x002C, 0xBB21,0xBB4c,0xBB17,0xBB55.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 and 6.

Example:
AT Mode:
For IPv4:
To join a multicast group:
at+rsi_multicast=1,239.0.0.0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6D 0x75 0x6C 0x74 0x69 0x63 0x61 0x73 0x74 0x3D 0x31 0x2C 0x32 0x33 0x39 0x2E 0x30 0x2E 0x30 0x2E 0x30 0x0D 0x0A

Response:
OK \r\n
0x4F 0x4B 0x0D 0x0A
To leave a multicast group:
at+rsi_multicast=0,239.0.0.0\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6D 0x75 0x6C 0x74 0x69 0x63 0x61 0x73 0x74 0x3D 0x30 0x2C 0x32 0x33 0x39 0x2E 0x30 0x2E 0x30 0x2E 0x30 0x0D 0x0A

Response:
OK \r\n
0x4F 0x4B 0x0D 0x0A
For IPv6:
To join multicast ipv6 group:
at+rsi_multicast6=1,FF0E:0:0:0:0:0:0:1\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6D 0x75 0x6C 0x74 0x69 0x63 0x61 0x73 0x74 0x3D 0x31 0x2C 0x46 0x46 0x30 0x45 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x31 0x0D 0x0A

Response:
OK \r\n
0x4F 0x4B 0x0D 0x0A
To leave multicast ipv6 group:
at+rsi_multicast6=0,FF0E:0:0:0:0:0:0:1\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x6D 0x75 0x6C 0x74 0x69 0x63 0x61 0x73 0x74 0x36 0x3D 0x30 0x2C 0x46 0x46 0x30 0x45 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x30 0x3A 0x31 0x0D 0x0A

Response:
OK \r\n
0x4F 0x4B 0x0D 0x0A

Note:
if MDNS feature is enabled, multicast group is not supported.

5.57 Ping from Module

Description:
This command is used to send the ping request to the target IP address.

Command Format:
AT Mode:
at+rsi_ping=< ip_version >,< ping_address. ipv4_address/ ping_address. Ipv6_address >,< ping_size >, <timeout_ms>\r\n

Command Parameters:
ip_version:
IP version of the ping request.
4 - For IPV4
6 - For IPV6.

ping_size:
ping data size to send. Maximum supported is 300 bytes.

Ping_address.ipv4_address /Ping_address.ipv6_address:
Destination IPv4/IPv6 address.

Response:
AT Mode:

Result Code	Description
OK< ip_version>< ping_size > < ping_address. ipv4_address/ ping_address. Ipv6_address>	Successful execution of command
ERROR<Error code>	Failure

Response Parameters:
ip_version(2 bytes):
IP version of the ping reply.

ping_size (2 bytes):
Contains the length of the data which is present in the ping reply.

ping_address. ipv4_address/ping_address.ipv6_address:
Ipv4/IPv6 address of the ping reply. Last 12 bytes are set to '0' in case of IPv4.

timeout_ms:

ping request command response timeout in milli seconds.

Note:
Default ping request command response timeout is 1 second.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Possible Error Codes:
Possible error codes are 0x0025, 0x002C,0x002F, 0xBB29, 0xFF74,0x0015,0xBB21,0xBB4B,0xBB55.

Example:
AT Mode:

For IPv4:
at+rsi_ping=4,192.168.1.100,10\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x70 0x69 0x6E 0x67 0x3D 0x34 0x2C 0x31 0x39 0x32 0x2E 0x31 0x36 0x38 0x2E 0x31 0x2E 0x31 0x30 0x30 0x2C 0x31 0x30 0x0D 0x0A

Response:
0x4F 0x4B 0x04 0x00 0x0A 0x00 0xC0 0xA8 0x01 0x64 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0D 0x0A

For IPv6:
at+rsi_ping=6,2001.db8.1.0.0.0.0.123,10\r\n
0x61 0x74 0x2B 0x72 0x73 0x69 0x5F 0x70 0x69 0x6E 0x67 0x3D 0x36 0x2C 0x32 0x30 0x30 0x31 0x2E 0x64 0x62 0x38 0x2E 0x31 0x2E 0x30 0x2E 0x30 0x2E 0x30 0x2E 0x30 0x2E 0x31 0x32 0x33 0x2C 0x31 0x30 0x0D 0x0A

Response:
0x4F 0x4B 0x06 0x00 0x0A 0x00 0xB8 0x0D 0x01 0x20 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x23 0x01 0x00 0x00 0x0D 0x0A

5.58 Loading the Webpage

RS9116W allows two kinds of Webpages to be stored: static and dynamic. Static pages allow plain html, css and JavaScript; whereas dynamic pages allow JSON data to be associated with the static Webpages. This JSON data can be stored, retrieved and updated independently. The Webpages can fetch this data and dynamically fill form fields. These fields can be modified from the host side or the browser.
The host can store up to 10 Webpages, each up to 4K in size. The size of a page can exceed the 4K limitation, but this will result in lesser number of files that can be stored. For example, user could store one 12K file, and seven 4K files. Similarly, there can be 10 JSON data objects with each NOT exceeding 512 Bytes in any circumstances. These objects can only be stored if they have an associated webpage

5.58.1 Loading the static webpage

Description:
This command is used to load a static webpage, to store a static webpage and to overwrite an existing static webpage. This command should be issued before join command.
If webpage total length is more than MAX_WEBPAGE_SEND_SIZE, then host has to send webpage in multiple chunks.

Command Format:
AT Mode:
at+rsi_webpage= < filename >,< total_len >,< current_len >,< has_json_data >,< webpage >\r\n

Command Parameters:
filename:
name of the file to load the webpage.

total_len:
Total Length of the webpage

current_len:
Length of the current webpage chunk has_json_data
1 - if file has associated json data
0 - if no associated data. In this case webpage is static page.

webpage:
The HTML content chunk.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of command.
ERROR<Error code>	Failure.

Response Parameters:

There is no response payload.

Possible Error codes:
Possible error codes are 0x0021, 0x0015,0x0025, 0x00C1, 0x00C2, 0x00C3 , 0x00C5, 0x00C6,0x00C8

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Example:
The host can also overwrite an existing page, this is achieved by issuing this same command with the same filename. No explicit erase is required. However, the size of the new file cannot exceed the size of the old file rounded upto the 4K chunk boundary. Precisely, if an old file used 2 chunks of 4K (i.e. up to 8K in size), then the new file can't exceed 8K in size. To overcome this limitation, the host should erase the existing file and then write a new file which can exceed the size of the old file provided the device has enough space available.

AT Mode:
at+rsi_webpage=sample.html,2783,1024,0,<html><head><title></title>[1024-chars]\r\n

Response:
OK\r\n
E.g.:
at+rsi_webpage=page1.html,1024,1024,0,<html><head><title></title>[1024-chars]\r\n

5.58.2 Loading the dynamic webpage (Create JSON)

Description:
This command is used to load the associate json data with the static WebPages.

Command Format:
AT Mode:
at+rsi_jsoncreate= < filename >,< total_len >,< current_len >,< has_json_data >,< webpage >\r\n

Command Parameters:
filename:
The webpage file with which this JSON data is associated.

total_length:
total length of the JSON

current_length:
length of current JSON chunk

json_data:
This is the JSON Object that stores the data of the webpage.

Note:
Maximum supported JSON length is 512Check length bytes.

Response:
AT Mode:

Result Code	Description
OK	Success.
ERROR<Error code>	Failure.

Response Parameters:

There is no response payload.

Possible Error codes:
Possible error codes are 0x0015, 0x0021, 0x0025,0x002C, 0x00B1,0x00B2,0x00B3,0x00B4,0x00B5,0x00B6.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Example:
AT Mode:
Webpage should already be present in modules flash with bool json set to 1 before issuing this command.
at+rsi_jsoncreate=sample.html,60,60,{"temp":27, "accx":2.4, "accy":2.6, "accz":1.1, "enabled":true}\r\n

Response:
OK\r\n
Webpage should already be present in modules flash with bool json set to 1 before issuing this command.

Command:
at+rsi_jsoncreate=sample.html,60,60,{"temp":27, "accx":2.4, "accy":2.6, "accz":1.1, "enabled":true}\r\n

Response:
OK\r\n

5.59 Clearing the Webpage

These commands are used to erase the webpage and information related to webpage from the flash.

5.59.1 Erasing the webpage

Description:
This command is used to erase the webpage file from the FLASH. This command should be issued before join command. The erase command should be used specifying the filename, this will free up the number of 4K chunks the existing file was using, for writing new files.

Command Format:
AT Mode:
at+rsi_erasefile= < filename> \r\n

Command Parameters:
filename:
name of the webpage file that has to be erased.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of command.
ERROR<Error code>	Failure.

Response Parameters:

There is no response payload.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Possible Error codes:
Possible error codes are 0x0021, 0x0025,0x002C, 0x00C4

Example:
AT Mode:
at+rsi_erasefile = sample.html\r\n

Response:
OK\r\n

5.59.2 Erasing the JSON Data

Description:
This command is used to erase the JSON data file associated with a webpage in the FLASH.

Command Format:
AT Mode:
at+rsi_erasejson = <filename> \r\n

Command Parameters:
filename:
name of the webpage file of which JSON data must be erased.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of command.
ERROR<Error code>	Failure.

Response Parameters:

There is no response payload.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Possible Error Codes:
Possible error codes are 0x0021, 0x0025, 0x002C, 0x00B4.

Example:
AT Mode:
at+rsi_erasejson=sample.html\r\n

Response:
OK\r\n

5.59.3 Clear all the Webpages

Description:
This command is used to erase all the Webpages in the file system.

Command Format:
AT Mode:
at+rsi_clearfiles= < clear > \r\n

Command Parameters:
clear:
set '1' to clear files.

Response:
AT Mode:

Result Code	Description
OK	Successful execution of command.
ERROR<Error code>	Failure.

Response Parameters:

There is no response payload.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Possible Error Codes:
Possible error codes are 0x0021, 0x0025, 0x002C.

Example:
AT Mode:
Command:
at+rsi_clearfiles=1\r\n

Response:
OK\r\n

5.60 Loading of Store Configuration Page

RS9116W allows two kinds of Webpages to be stored: static and dynamic. Static pages allow plain html, css and JavaScript; whereas dynamic pages allow JSON data to be associated with the static Webpages. Store configuration page is a static web page.
To load store configuration page Host need to use "store_config.html" as a URL name and no need of loading the associated JSON data (Dynamic web page).
The size of the page should not exceed the 40K .
The field present in the webpage can be disable by the host. But host can't change the webpage fields structure. Host can change the "LOGO" and "TITLE" which are present in the store configuration page.
For loading Store configuration page refer section: **Loading the webpage.**

5.61 Web Page Bypass

Description:
This command is used to send URL response.
When unavailable page request is received, module will give asynchronous indication to host for requesting web page. Host has to respond with URL RESPONSE frame.
When module receives the query then it checks that if it already has the page in its memory. If yes, it sends out the page to the remote terminal and the query is serviced. If not, it sends the asynchronous URL REQUEST.
Asynchronous Message is as follows.

AT Mode:
AT+RSI_URLREQ< url_length> < url_name >< request_type > <post_content_length >< post_data >\r\n
After receiving this asynchronous frame from module, host has to respond with URL RESPONSE frame with the following payload.

Note:
Please note that whenever you are giving url request from a browser to the module, .html should be given.

url_length(1 byte):
This is the number of characters in the requested URL.

url_name (40 bytes):
This is actual URL name.

request_type(1 byte):
type of the request received.
0 – HTTP GET request
1 – HTTP POST request

post_content_length(2 bytes):
length of the post content

post_data(512 bytes):
HTTP POST received.
post_content_length, post_data fields are valid if request_type is HTTP POST.
These fields can be ignored in case of request_type is HTTP GET.

Command Format:

AT Mode:
at+rsi_urlrsp=< total_len >,< more_chunks >,< webpage >\r\n
The Host , after receiving this asynchronous message from the module, should fetch the page from its memory and give it back to the module with the message
at+rsi_urlrsp=< total_len >,< more_chunks >,< webpage >\r\n

Command Parameters:
total_len(4 bytes):
This is the total number of characters in the page.
If the queried web page is not found, the Host should send '0' for this parameter.

more_chunks(1 byte):
'0'- There are no more segments coming from the Host after this segment
'1'- There is one more segment coming from the Host after this Segment

webpage(1400 bytes):
This is the actual source code of the current segment

Response:
AT Mode:

Result Code	Description
OK	Successful execution of command.
ERROR<Error code>	Failure.

Response Parameters:
There is no response payload.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 1, 2 or 6.

Possible Error Codes:
Possible error codes are 0x0015,0x0021,0x0025,0x002C.

Example:
AT Mode:
Example 1:

If the web page source code is of 3000 characters, the Host should send it through 3 segments, the first two of 1400 bytes, and the last one of 200 bytes as shown:
AT+RSI_URLRSP=< total_len =3000>,< more_chunks =1>, < webpage =code of the 1st segment>\r\n
AT+RSI_URLRSP=< total_len =3000>,< more_chunks =1>, < webpage =code of the 2nd segment>\r\n
AT+RSI_URLRSP=< total_len =3000>,< more_chunks =0>, < webpage =code of the 3rd segment>\r\n

Example 2:

If the queried web page is not found in the Host, then host should send
AT+RSI_URLRSP=0\r\n

5.62 Set Region

Description:
This command used to configure the device to operate according to the regulations of its operating country. This command should be immediately followed by init command.

Command Format:
AT Mode:
at+rsi_setregion=<setregion_code_from_user_cmd>,<region_code>/r/n

Command Parameters:
setregion_code_from_user_cmd(1 byte):
Enable/Disable set region code from user.
1 - Enable - Use the region information from user command
0 – Disable - Use the region information from beacon (country Ie)

Note:
1. For world mode domain, setting the region information from beacon is not supported.

Region_code(1 byte):

0/1-US domain(Default)
2-Europe domain
3-Japan Domain
4-World mode domain

Response:
AT Mode:

Result Code	Description
OK< region_code >	Successful execution
ERROR	Failure.

Response Parameters:

region_code(1 bytes):
0x01(US domain)
0x02(Europe domain)
0x03(Japan Domain)
0x04(Other than above three domains)

Note:

1. In dual band mode, if country element extracted from beacon in 2.4Ghz band and 5Ghz band are not matching, error is thrown and region is set to US
2. Refer to the tables below for the region supported and domain rules followed by Silicon module

Relevance:
This command is relevant when the module is configured in Operating Mode 0,1,2,8 modes.
The tabulated rules are followed for the regions supported by the Silicon module.

Table 12 Regulations followed for US domain

Rule No	band	First Channel	Number of Channels	Last Channel	Maximum power in dBm	scan type
1	2.4GHz	1	11	11	27	Active
2	5GHz	36	4	48	16	Active
3	5GHz	52	4	64	23	Passive
4	5GHz	100	5	116	23	Passive
5	5GHz	132	4	144	23	Passive
6	5GHz	149	5	165	29	Active

Table 13 Regulations followed for Europe domain

Rule No	band	First Channel	Number of Channels	Last Channel	Maximum power in dBm	scan type
1	2.4GHz	1	13	13	20	Active
2	5GHz	36	4	48	23	Active
3	5GHz	52	4	64	23	Passive
4	5GHz	100	11	140	30	Passive

Table 14 Regulations followed for Japan domain

Rule No	band	First Channel	Number of channels	Last Channel	Maximum power in dBm	scan type
1	2.4GHz	1	14	14	20	Active
2	5GHz	36	4	48	20	Active
3	5GHz	52	4	64	20	Passive
4	5GHz	100	11	140	30	Passive

Note:
Though the transmit power levels w.r.t to region are greater than 20dBm, Silicon module is supporting maximum of 18dBm.

Possible Error Codes:
Possible error codes for this command are 0x0021, 0x0025, 0x002C, 0xFF82, 0x00CC, 0x00C7, 0x00CD,0x00CE

Example:

AT Mode:

Command:
Setting Japan region.
at+rsi_setregion=1,3\r\n

Command:
Setting world mode region domain.
at+rsi_setregion=1,4\r\n

Response:
OK\r\n

5.63 Set Region of Access Point

Description:
This command is used to set the region domain of the module in Access point mode. This command helps device to self-configure and operate according to the regulations of its operating country and includes parameters like country name, channel quantity and maximum transmission level. These parameters are added in Country information element in the beacons and probe responses. This command should be immediately followed by init command.

Command Format:
AT Mode:
at+rsi_setregion_ap=<setregion_code_from_user_cmd>,< country_code>,<no_of_rules>,[<first_channel>,<Number of channel>,< max_tx_power>,< First channel >,<Number of channels>,<max_tx_power>],…………no of rules times/r/n

Command Parameters:
setregion_code_from_user_cmd(1 byte):
set region code from user command enable/disable
1- Enable-Get the region information from user command
0- Disable-Get the region information based on region code from module's internal memory

Note:
Country code: Country code is of 3 bytes.Country code is case sensitive and should be in *Upper case*.If the first parameter is 1,the second parameter should be one of the these 'US','EU','JP' country codes.

Note:
If the country code is of 2 characters, 3rd character should be <space>.

Note:
The below parameters are considered only if the first parameter is 1.

Number of rules(4 byte):
Number of rules for the given domain

First channel(1 byte):
Start channel for the nth rule

Number of channels(1 byte):
number of channels holding the same rule from the start channel

Maximum transmit power(1 byte):
Maximum transmit power used in that set of channels.

Response:
AT Mode:

Result Code	Description
OK	Successful execution
ERROR	Failure.

Relevance:

This command is relevant when the module is configured in Operating Mode 6**.**

Note:

1. AP configuration in DFS channels (52 to 140) is not supported
2. Though the transmit power levels w.r.t., to region are greater than 20dBm, Silicon module is supporting maximum of 18dBm

Possible Error Codes:
Possible error codes for this command are 0x0021,0x0025,0x002C, 0x00ca, 0x00cb,0x00cc,0xFF71,0xFF82

Example:
AT Mode:

Example 1:
at+rsi_setregion_ap=1,US<space>,2,1,4,23,5,7,30\r\n
Explanation:
From the above command, consider the first rule 1,4,23
First channel is given as 1, and no of channels is 4.this means channels 1 to 4 (1,2,3,4)holds the maximum Tx power 23dBm
Consider the second rule 5,7,30
First channel is given as 5, and no of channels is 7.this means channels 5 to 11 (5,6,7,8,9,10,11)holds the maximum Tx power 30dBm

Note:
The Country code given in the command reflects as it is in the beacon frame.

Example 2:
Command:
at+rsi_setregion_ap=0,US<space>\r\n

Response:
OK\r\n

Note:
Refer to the tables in the Set Region section for the region supported and domain rules followed by Silicon module

5.64 PER Statistics of the Module

Description:
This command is used to get the Transmit(TX) & Receive(RX) packets statistics. When this command is given by the host by enabling this feature with valid channel number, module gives the statistics to host for every second until this feature is disabled. This command can be given after init command.

Command Format:
AT Mode:
at+rsi_per_stats=< per_stats_enable >,< per_stats_channel >\r\n

Command Parameters:
per_stats_enable(2 bytes):
Enable /disable per status feature

0 – Enable
1 – Disable
per_stats_channel(2 bytes):
valid channel number in which user wants to get the stats. Channel number is not required if the first parameter is 1(Disable)

Response:
AT Mode:

OK	Successful execution
ERROR	Failure.

Once PER stats is enabled, module sends asynchronous message for every second. Following list of fields will be given to host.

Result Code	Description
AT+RSI_PER_STATS=<tx_pkts >,<reserved_1>,<tx_retries >,<crc_pass>,<crc_fail>,<cca_stk>,<cca_not_stk>,<pkt_abort>,<fls_rx_start>,<cca_idle>,<reserved_2>,<rx_retries>,<reserved_3>,<cal_rssi>,< reserved_4>,<xretries>,<max_cons_pkts_dropped>,< reserved_5><bss_broadcast_pkts>,<bss_multicast_pkts>,<bss_filter_matched_multicast_pkts>,<eof_pkt_drop_count>,<mask_pkt_drop_count>,<no_of_acks_sent>,<pkt_rcvd_with_48M>,<pkt_rcvd_with_24M>,<pkt_rcvd_with_12M>,<pkt_rcvd_with_6M>,<pkt_rcvd_with_54M>,<pkt_rcvd_with_36M>,<pkt_rcvd_with_18M>,<pkt_rcvd_with_9M>,<pkt_rcvd_with_11M>,<pkt_rcvd_with_5M>,<pkt_rcvd_with_2M>,<pkt_rcvd_with_1M>,<pkt_rcvd_with_mcs0>,<pkt_rcvd_with_mcs1>,<pkt_rcvd_with_mcs2>,<pkt_rcvd_with_mcs3>,<pkt_rcvd_with_mcs4>,<pkt_rcvd_with_mcs5>,<pkt_rcvd_with_mcs6>,<pkt_rcvd_with_mcs7>	

Response Parameters:

tx_pkts(2 bytes):
Number of TX packets transmitted

reserved_1(2 bytes):
Reserved

tx_retries(2 bytes):
Number of TX retries happened

crc_pass(2 bytes):
Number of RX packets that passed CRC

crc_fail(2 bytes):
Number of RX packets that failed CRC

cca_stk(2 bytes):
Number of times cca got stuck

cca_not_stk(2 bytes):
Number of times cca didn't get stuck

pkt_abort(2 bytes):
Number of times RX packet aborts happened

fls_rx_start(2 bytes):
Number of false rx starts. If valid WLAN packet is received and is dropped due to some reasons.

cca_idle(2 bytes):
CCA idle time

reserved_2(2 bytes):
Reserved

rx_retries(2 bytes):
Number of RX retries happened

reserved_3(2 bytes):
Reserved

cal_rssi(2 bytes):
The calculated RSSI value of recently received RX packet

reserved_4(4 bytes):
Reserved

xretries(2 bytes):
Number of TX Packets dropped after maximum retries

max_cons_pkts_dropped(2 bytes):
Number of consecutive packets dropped after maximum retries

reserved_5(2 bytes):
Reserved

bss_broadcast_pkts(2 bytes):
BSSID matched broadcast packets count.

bss_multicast_pkts(2 bytes):
BSSID matched multicast packets count.

bss_filter_matched_multicast_pkts(2 bytes):
BSSID and multicast filter matched packets count. The filtering is based on the parameters given in multicast filter command Set/Reset Multicast filter. If multicast filter is not set then this count is equal to bss_multicast_pkts count.

Note:

1. In PER mode (opermode 8) following stats related to RX packets (crc_pass, crc_fail, cca_stk, cca_not_stk, pkt_abort, fls_rx_start, cca_idle) are only valid, remaining fields can be ignored.
2. The multicast stats are valid only in associated state in client mode and are invalid in non-associated state. In associated state, the stats are for packets which are destinated for the module's MAC address only. And in non-associated state, the stats are for all the packets received, irrespective of the destination MAC address.
3. The paramters valid in other than PER mode(opermode 0,1,2,6) mode : tx_pkts, tx_retries cal_rssi, xretries, crc_pass, max_cons_pkts_dropped, crc_fail, cca_stk, cca_not_stk, pkt_abort, fls_rx_start, cca_idle, bss_broadcast_pkts, bss_multicast_pkts, bss_filter_matched_multicsast_pkts.

Relevance:
This command is valid in all operating modes.

Possible Error Codes:
Possible error codes for this command are 0x0021, 0x0025, 0x002c, 0x000A.

5.65 Remote Socket Closure

Description:
This is an asynchronous message which will be given to host in the following cases.
1. When the remote peer closes connected TCP/SSL/Web socket.
2. When module is not able to send data over socket because of unavailability of remote peer.
3. When remote peer disappears without intimation then module closes socket after TCP keep alive time (~20 minutes) and sends this message to host.

Command Format:
N/A

Command Parameters:
N/A

Response:
AT Mode:
AT+RSI_CLOSE< socketDsc >< bytesSent >\r\n

Response Parameter:
socketDsc(2 bytes):
Socket descriptor of the socket which is closed.

bytesSent(2 bytes):
Number of bytes sent successfully on that socket

Possible error codes:
No possible error code as it is asynchronous message from module to host.

Relevance:
This command is relevant when the module is configured in Operating Mode 0,1, 2 and 6.

5.66 Bytes Transmitted Count on Socket

Description:
This command is used to get the number of bytes transmitted successfully by the module on a given socket.

Command Format:
AT Mode:
at+rsi_bytes_sent_count=< sock_handle >\r\n

Command Parameters:

sock_handle: socket handle on which number of bytes have been successfully transmitted.

Response:
AT Mode:

OK<sock_handle><SentBytescnt >	Success.
ERROR	Failure.

Response Parameters:

socket_handle(2 bytes):
Socket handle on which number of bytes have been successfully transmitted.

SentBytescnt(4 bytes): Number of bytes sent successfully on the given socket handle

Relevance:
This command is relevant when the module is configured in Operating Mode 0,1, 2 and 6.

Possible Error Codes:
Possible error codes for this command are 0xFF86, 0XFFFA, 0xFF82, 0x002C, 0x0025, 0x0021.

5.67 Debug Prints on UART

Description:
This command is used for debug prints on UART interfaces 1 and 2. Host can get 4 types of debug prints based on the assertion level and assertion type.

To select UART 1 or 2 interfaces in bit(27) of ext_custom_feature_bit_map

PIN Configuration:

UART 1 : From Host GPIO_9 is for Tx and GPIO_8 is for Rx
UART 2 : From Host GPIO_6 is for Tx and GPIO_10 is for Rx

Command Format:
AT Mode:
at+rsi_debug=< assertion_type >,< assertion_level >\r\n

Command Parameters:
assertion_type(4 byte):
Possible values are 0 to 15.
assertion_level(4 byte):
Possible values 0 to 15. 1 being least level and 15 being highest level of debug prints. 0 is to disable all the prints.

Note:

1. If debug prints are enabled once, to disable the debug prints host is supposed to give the same command with assertion type and assertion level as 0.
2. Baud rate for UART 2 on host application side should be 460800.

Response:
AT Mode:

OK	Successful execution
ERROR	Failure.

Possible Error Codes:

Possible error codes for this command are 0x0021, 0x0025, 0x002C, 0xFFF8

Relevance:
This command can be given at any time.

5.68 Asynchronous Message for Connection State Notification

Description:
Asynchronous message are used to indicate module state to host. These message are enabled by setting 10th bit in customer feature select bitmap in opermode command, please refer opermode command.

Command Format:
N/A

Response:
AT Mode:

AT+RSI_STATE-I< TimeStamp >,<StateCode>,<reason_code>,< rsi_channel>,<rsi_rssi>,<rsi_bssid>\r\n	This type of asynchronous message is given by the module when it is in scanning state.
AT+RSI_STATE-II< TimeStamp >,<StateCode>,<reason_code>,< rsi_channel>,<rsi_rssi>,<rsi_bssid>\r\n	This kind of message is given by the module once the scan results are observed and decided to join or not to join/Rejoin to AP.
AT+RSI_STATE-III< TimeStamp >,<StateCode>,<reason_code>,< rsi_channel>,<rsi_rssi>,<rsi_bssid>\r\n	Once the association or disassociation is done, module will give final state asynchronous message

Response Parameters:

TimeStamp(4 bytes):This is value of counter at the time of message. This counter is continuously incrementing by one per 100ms time.

stateCode(1 byte): This field indicates state of the module. state code contains two parts (upper nibble/lower nibble).

Upper nibble represents the state of rejoin process. Following are the possible values of StateCode represented by the upper nibble of state code.

Scan Trigger State (state-I): Indicates the reason for scan triggered
0x00-Startup (Initial Roam)
0x10-Beacon Loss (Failover Roam)
0x20-De-authentication (AP induced Roam / Disconnect from supplicant)

Scan Result/Decision State(state – II): Indicates a state change based on scan result
0x50-Current AP is best
0x60-Better AP found
0x70-No AP found

Final Connection State (state – III): Indicates the connection state change
0x80-Associated
0x90-Unassociated

Following are the possible values of StateCode represented by the lower nibble of state code

Indicates reason for state change
0x00-No reason specified
0x01-Authentication denial
0x02-Association denial
0x03-AP not present
0x05-WPA2 key exchange failed

reason_code(1 byte):This is used to get the reason code from firmware point of view. Following are the possible reason code for the failure.
0x00-No reason specified
0x01-Authentication denial
0x02-Association denial
0x10-Beacon Loss (Failover Roam)
0x20-De-authentication (AP induced Roam/Deauth from supplicant)
0x07-PSK not configured
0x09-Roaming not enabled

rsi_channel(1 byte):

Following represents the meaning of AP channel at the given stage.
State-I: channel of association or Invalid if it is startup.
State-II: channel of next association if module finds better AP in bgscan result.
State-III: Channel at the time of association.

If value of rsi_channel is 0, it means channel information is not available.
rsi_rssi(1 byte): Following represents the meaning of AP RSSI at the given stage.
State-I: RSSI of AP at the time of trigger.
State-II: RSSI of next association.
State-III: RSSI at the time of final association.

If value of rsi_rssi is 100, it means RSSI information is not available.
rsi_bssid(6 bytes): Following represents the meaning of AP MAC at the given stage.
State-I: MAC of AP at the time of scan trigger.
State-II: MAC of next association.
State-III: MAC at the time of association.

Note:

If the value of AP MAC is 00:00:00:00:00:00, it means MAC information is not available.

Response:
N/A

Relevance:
This command is relevant in oper modes 0,1,2 and 6.

Possible Error Codes:
N/A

Note:

By default, this feature is disabled. To enable this feature host has to set the custom bit 0x400 in opermode command.

5.69 TSF Synchronization Packet

Description:

This command is used to send a frame for timing synchronization. This command is valid only if module is connected with the specified peer. If the specified peer is not connected, module will return error code (0x0078) to host.
Command Format in UART mode:
at+rsi_sync_pkt

Usage:
at+rsi_sync_pkt=<MAC_address>,<payload>\r\n

Command Parameters:
MAC_address(6 bytes):
MAC address of receiver

payload(100 bytes):
payload to be sent to the receiver

Response:

Transmitter End:
In response to above command, module returns TSF timer value captured after sending the packet on air at transmitter end.
OK<TSF Value (8 bytes)>
Example: 4f 4b 06 2b 25 02 00 00 00 00 0d 0a

Receiver End:
On the receiver end, an asynchronous message with senders MAC address, TSF timer value followed by payload is forward to host.
AT+RSI_TSF_SYNC_PKT_RECVD=<Mac Addr(6bytes)> <reserved (2bytes)> <TSF Value(8bytes) ><payload>

Command Format in SPI mode:

Description:

Host will issue the command for sending TSF synchronization packet with frame type (0xC0). At reception of this command, Module will prepare a management frame (Action frame) attaching the payload. The module then captures the timestamp after sending the packet on air and sends back the response frame to host. On the receiver side, the module filters the packet based on Action frame type, attaches the timestamp and forwards the asynchronous packet to the host.

Request Frame Type:
RSI_REQ_HOST_TSF_SYNC_PKT 0xC0

5.70 Station Connect/Disconnect Indication In AP Mode

Description:
Asychronous message are used to indicate host in AP mode when the station is connected (frame type 0xC2)/disconnected (frame type 0xC3).

Command Format:
N/A

Response:
AT Mode:

AT+RSI_CLIENT_STATION_CONNECTED= < MAC_address >	MAC address of station connected

AT+RSI_CLIENT_STATION_DISCONNECTED=< MAC_address >	MAC address of station disconnected

Response Parameters:

MAC_address(6 bytes): MAC address of station connected/disconnected

Relevance:
This command is valid when opermode is 1 or 6.

Possible Error Codes:
N/A

5.71 Transparent Mode Command

Description:
This command is used to Enter/Start transparent mode, parameters for transparent mode are to be provided in this command. On reception of this command, module tries to start transparent mode and replies with "AT+RSI_TMODE " message with status.

Command Format:
AT Mode:
at+rsi_ trans_mode_params=<packetization Length>, <Escape character>, <gap time>, <frame time>, <escape time>, <IP version>, <socket type>, <local port>, <Destination port>, <IP Address>, <Max_count>, <Type of service>, <SSL Parameters>, <SSL Ciphers>\r\n

Command Parameters:

Packetization Length:
Possible values are 10 to 1024 Escape character: Any special character

Gap Time(in milliseconds):
Varies from 0 to 65533

Frame time(in milliseconds):TLS
varies from 1 to 65534(should be greater than gap time)

Escape Time(in milliseconds):
Varies from 2 to 65535(should be greater than frame time)

IP Version:
Possible values are 4 or 6
4 - IPV4
6 - IPV6

Socket Type:
Possible values are
0 – TCP
2 – LTCP
4 – LUDP

Local Port number:
Local port number

Destination Port number:
Destination port number

IP Address:
Server IP address if socket type is LUDP/LTCP

Max_count:
Maximum no. of clients in LUDP/LTCP, fixed to 1 in transparent mode.

Type of Service:
Type of service, varies from 0 to 8

SSL parameters:
This field is used to enable SSL for selected socket.

Possible values:
0 – To open TCP socket.
1 - To open SSL client socket.
5 - To open SSL socket with TLS 1.0 version.
9 - To open SSL socket with TLS 1.2 version.

ssl_ciphers : to select various cipher modes, possible values
2- TLS_RSA_WITH_AES_256_CBC_SHA256
4 - TLS_RSA_WITH_AES_128_CBC_SHA256
8 - TLS_RSA_WITH_AES_256_CBC_SHA
16 - TLS_RSA_WITH_AES_128_CBC_SHA
32 - TLS_RSA_WITH_AES_128_CCM_8
64 - TLS_RSA_WITH_AES_256_CCM_8

Response:
AT Mode:

Result Code	Description
AT+RSI_TMODE0	Successfully entered into transparent mode
AT+RSI_TMODE1	Graceful exit from Transparent mode (by giving escape sequence from host after escape time)
AT+RSI_TMODE2	Exited from transparent mode due to WiFi Disconnected.
AT+RSI_TMODE3	Exited from transparent mode due to TCP Remote terminate from Peer.
AT+RSI_TMODE4	Exited from transparent mode due to TCP retries over terminated TCP connection.
AT+RSI_TMODE5	Did not enter transparent mode due to invalid transparent mode parameters.
AT+RSI_TMODE6	Did not enter transparent mode due to module doesn't have IP.
AT+RSI_TMODE7	Did not enter transparent mode as could not create requested socket.
Error Codes	Failure. Possible error codes for this command are 0xFF87, 0x0021,0x0025,0x002C,0xFFF8.

Relevance:
This command can be given only after successful connection with AP, module should have a valid IP and there should be no prior sockets opened.

Note:

This command is only valid in AT mode using UART interface.

5.72 UART Hardware Flow control

Description:
This command is used to Enable/Disable the hardware flow control feature.
This command is valid only in case of UART host interface.

Command Format:
AT Mode:
at+rsi_uart_hwflowctrl=<uart_hw_flowcontrol_enable_pinset>\r\n

This command has been deprecated.

or

at+rsi_hfc=<uart_hw_flowcontrol_enable_pinset>\r\n

Note:
This command is only valid in AT mode using UART interface. User can use any of the two commands, but it is recommended to use

at+rsi_hfc=<uart_hw_flowcontrol_enable_pinset>\r\n for enabling hardware flow control.

Command Parameters:
uart_hw_flowcontrol_enable(1 byte):
Enable or Disable UART hardware flow control.
1/2 -Enable and Pin set to be used to for RTS/CTS purpose.
0 - Disable

If uart_hw_flowcontrol_enable parameter is 0, uart flow control is disabled. If the parameter is given as 1 or 2, it means uart hardware flow control is enabled and Pin set to be used

If parameter is given as 1: Pin set used for RTS/CTS functionality will be

UART_CTS	:	GPIO - 11
UART_RTS	:	GPIO - 7

If parameter is given as 2: Pin set used for RTS/CTS functionality will be

UART_CTS	:	GPIO - 15
UART_RTS	:	GPIO - 12

Response:
AT Mode:

OK	Successful execution
ERROR	Failure

Response Parameters:

uart_hw_flow_control_enabled(1 byte) : If uart flow control is enabled, then response parameter will 1 else 0.

Possible Error Codes:
Possible error codes for this command are 0x004E,0x002C.

Note:

Hardware flow control must be enabled in the module before enabling it in the host.

5.73 Socket Configuration Parameters

Description:
This command is used to set the socket configuration parameters. User is recommended to use this command(optional). Based on the socket configuration, module will use available buffers effectively. This command should be given after IP configuration command and before any socket creation.

Command Format:
AT Mode:
at+rsi_socket_config=<total_sockets>,<total_tcp_sockets>,<total_udp_sockets>,<total_tcp_tx_only_sockets>,<total_tcp_rx_only_sockets>,<total_udp_tx_only_sockets>,<total_udp_rx_only_sockets>,<total_tcp_rx_high_performance_sockets>,<tcp_rx_window_size_cap>,<tcp_ack_window_division_factor>\r\n

Command Parameters:

total_sockets(1 byte):
Desired total number of sockets to open.

total_tcp_sockets(1 byte):
Desired total number of TCP sockets to open.

total_udp_sockets(1 byte):
Desired total number of UDP sockets to open.

tcp_tx_only_sockets(1 byte):
Desired total number of TCP sockets to open which are used only for data transmission.

tcp_rx_only_sockets(1 byte):
Desired total number of TCP sockets to open which are used only for data reception.

udp_tx_only_sockets(1 byte):
Desired total number of UDP sockets to open which are used only for data transmission.

udp_rx_only_sockets(1 byte):
Desired total number of UDP sockets to open which are used only for data reception.

tcp_rx_high_performance_sockets(1 byte):
Desired total number of high performance TCP sockets to open. High performance sockets can be allocated with more buffers based on the buffers availability. This option is valid only for TCP data receive sockets. Socket can be opened as high performance by setting high performance bit in socket create command.

tcp_rx_window_size_cap(1 byte):
Desired to increase the tcp rx window size

Following conditions has to be met:

1. total_sockets <= Maximum allowed sockets (10)
2. (total_tcp_sockets + total_udp_sockets) <= total_sockets
3. (total_tcp_tx_only_sockets + total_tcp_rx_only_sockets) <= total_tcp_sockets
4. (total_udp_tx_only_sockets + total_udp_rx_only_sockets) <= total_udp_sockets
5. total_tcp_rx_high_performance_sockets <= total_tcp_rx_only_sockets

tcp_ack_div_factor(1 byte):
In case of high latency networks in order to give TCP ACK with respective to the window size. Default value is 2.

Possible values: 2 - tcp_rx_window_size_cap

Response:
AT Mode:

OK	Successful execution
ERROR	Failure.

Relevance:

This command is valid when opermode is 0,1,2 or 6.

Possible Error Codes:
Possible error codes for this command are x0021,0x0025,0x002C,0xFF6D.

Example
AT Mode:
at+rsi_socket_config=4,2,2,1,1,1,1,1,10\r\n

5.74 RF Current Mode Configuration

Description:
This command is used to configure modules RF in different current/power consumption modes. This command should be given only before init command.

Command Format:
AT Mode:
at+rsi_rf_current_mode=< rf_rx_curr_mode >,< rf_tx_curr_mode >,< rf_tx_dbm >\r\n

Command Parameters:
rf_rx_curr_mode(1 byte):

Current/Power Mode in which modules RF-Receive(RX) should be programmed.
0 – High Current/Power mode
1 – Medium Current/Power mode
2 – Low Current/Power mode

rf_tx_curr_mode(1 byte):
Current/Power Mode in which modules RF-Transmit(TX) should be programmed.
0 – High Current/Power mode
1 – Medium Current/Power mode
2 – Low Current/Power mode

rf_tx_dbm(2 byte):
This two bytes are reserved. Set to '0'.

Response:
AT Mode:

OK	Successful execution
ERROR	Failure.

Response Parameters:

N/A

Relevance:
This command is valid when opermode is 0,1,2 or 6.

Possible Error Codes:
Possible error codes for this command are 0x0021, 0x0025, 0x002C, 0x0051.

Example:
AT Mode:
at+rsi_rf_current_mode=0,0,0\r\n : Program Modules RF TX and RX in high power mode
at+rsi_rf_current_mode=1,1,0\r\n : Program Modules RF TX and RX in medium power mode
at+rsi_rf_current_mode=2,1,0\r\n : Program Modules RF-TX in medium power mode and RF-RX in low power mode**.**

5.75 Trigger Auto Configuration

Description:
This command is used to trigger the Stored Auto Configuration. This command should be given only after Card Ready response.

Command Format:
AT Mode:
at+rsi_trigger_auto_config\r\n

Response:

AT Mode:

None	Successful execution
ERROR	Failure.

Response Parameters:

N/A

Relevance:
This command is valid when opermode is 0,1,2 or 6.

Possible Error Codes:
Possible Error codes are 0x0021,0xFF36,0xFF74, 0xFF35

Note:

1. To avail this feature(Wait On Host) user need to set BIT(20) in Custom feature bit map in opermode command.
2. This feature is valid Only when store configuration feature is enabled.
3. AT mode: If this feature is enabled, after "Loading Done" message "AT+RSI_TRIGGER_AUTO_CONFIG " will come, so that user can give either at+rsi_trigger_auto_config command to trigger the auto configuration, (or) user can continue with the Opermode command.

5.76 HTTP Abort

Description:
This command is used to abort the HTTP/HTTPS GET/POST

Pre-Condition:
This command should be given only after Ipconf command.

Command Format:
AT Mode:
at+rsi_http_abort\r\n

Response:
AT Mode:

OK	Successful execution
ERROR	Failure.

Response Parameters:
N/A

Relevance:
This command is valid when opermode is 0,1,2 or 6.

Possible Error Codes:
Possible Error codes for this command are 0x0021, 0x0025, 0x002C.

5.77 HTTP Server Credentials from Host

Description:
This command is used to set the HTTP Server Credentials.

Pre-Condition:
This command should be given only after Opermode command.

Command Format:
AT Mode:
at+rsi_credentials=<username>,<password>\r\n

username(31 byte): Username for HTTP Server. Default user name is 'redpine'.
password(31 byte): Password for HTTP Server. Default password is 'admin'.

Response:
AT Mode:

OK	Successful execution
ERROR	Failure.

Response Parameters:

N/A

Relevance:
This command is valid when opermode is 0,1,2 or 6.

Possible Error Codes:
Possible Error codes for this command are 0x0021, 0x0025, 0x00F1,0x0015.

5.78 FTP Client

Description:
This section explains different commands to use FTP client.
This command should be given only after **Set IP Parameters** command.
Following table explains list of ftp commands and their description.

FTP Command	Description
Create	Creates FTP objects. This should be the first command for accessing FTP.
Connect	Connects to FTP server.
Make Directory	Creates directory in a specified path.
Delete Directory	Deletes directory in a specified path
Change Working Directory	Changes working directory to a specified path.
Directory List	Lists directory contents in a specified path.
File Read	Reads the file
File Write	Open file to write
File Write Content	Writes content into file which is opened using File Write command. File content can be written in multiple chunks using this command.
File Delete	Deleted file
File Rename	Renames file
Disconnect	Disconnects from FTP server. Once disconnect is done user can connect again using connect command.
Destroy	Destroys FTP objects. Once destroy is given user can’t use FTP unless it is created again.

- **Create** should be called as a first command to use FTP.
- Once create is successful **connect** should be called to connect to a FTP server.
- After connection is successful host can issue remaining commands.
- After FTP operations host has to give **disconnect** command to disconnect from FTP server.
- Once disconnect is done, host can again connect to the FTP server using **connect** command.
- To destroy FTP objects host has to give **destroy** command. Once destroy is given user can’t use FTP unless it is created again using **create** command.

Command Format:
AT Mode:
FTP client has different command types. Based on the command type next parameters will change.
at+rsi_ftp=<command_type>,<remaining parameters >\r\n
Following are available command types.

FTP command	Command Type	Command Format
Create	**1**	at+rsi_ftp=1\r\n
Connect	**2**	at+rsi_ftp=2,<ip_version>, <IPaddress>,<username>,<password>,<server_port>\r\n **ip_version(1 byte):**

FTP command	Command Type	Command Format
		IP version to use. 4 – For IPv4 6 – For IPv6 **IP address(IPv4 : 4 byte,IP6 : 16 byte):** IPv4/IPv6 address for FTP server to connect. **username(31 byte):** username of FTP server **password(31 byte):** password of FTP server **server_port(4 byte):** FTP server port number
Make Directory	**3**	at+rsi_ftp=3,<Directory_path>\r\n **Directory_path(51 byte):** Path of the directory to make.
Delete Directory	**4**	at+rsi_ftp=4,<Directory_path>\r\n **Directory_path(51 byte):** Path of the directory to delete.
Change Working Directory	**5**	at+rsi_ftp=5,<Directory_path>\r\n **Directory_path(51 byte):** Change of directory path.
Directory List	**6**	at+rsi_ftp=6,<Directory_path>\r\n **Directory_path(51 byte):** path of the directory for list.
File Read	**7**	at+rsi_ftp=7,<file_name>\r\n **file_name(51 byte):** name of the file to read.
File Write	**8**	at+rsi_ftp=8,<file_name>\r\n **file_name(51 byte):** name of the file to write.
File Write content	**9**	at+rsi_ftp_file_content=<end_of_file>,<file_content>\r\n **end_of_file(1 byte):** Represents whether end of file is reached or not. 0 – More data is coming to write into file. 1 – Current chunk is the last chunk and no more data is coming. **File_content(1400 byte):** Content of the file to write.
File Delete	**10**	at+rsi_ftp=10,<file_name>\r\n **file_name(51 byte):** Name of the file to delete.
File Rename	**11**	at+rsi_ftp=11,<file_name>,<new_file_name>\r\n **file_name(51 byte):** Old name of the file. **new_file_name(51 byte):** New file name.
Disconnect	**12**	at+rsi_ftp=12\r\n
Destroy	**13**	at+rsi_ftp=13\r\n
Passive mode	14	at+rsi_ftp=14\r\n
Active mode	15	at+rsi_ftp=15\r\n

command_type(1 byte):
Type of the FTP command.
This parameter is valid for all commands.

Remaining parameters:

Ip_version(1 byte):
IP version to use. This parameter is valid for **connect** command.
4 – For IPv4
6 – For IPv6

server_ip_address.ipv4_address(4 byte):
IPv4 address of the FTP server. This parameter is valid for **connect** command.

server_ip_address.ipv6_address(16 byte):
IPv6 address of the FTP server. This parameter is valid for **connect** command.

username(31 byte):
username for the FTP server. This parameter is valid for **connect** command.

password(31 byte):
Password for the FTP server. This parameter is valid for **connect** command.

server_port(4 byte):
FTP server port number. This parameter is valid for **connect** command.

path(51 byte):
path of the directory/file.
This parameter is valid for **Make Directory**, **Delete Directory**, **Change Working Directory**, **Directory List**, **File Read**, **File Write**, **File rename**, **File Delete** commands.

new_file_name(51 byte):
New file name.
This parameter is valid for **File Rename** command.

end_of_file(1 byte):
Represents whether end of file is reached or not. This parameter is valid for **File Write Content** command.
0 – More data is coming to write into file.
1 – Current chunk is the last chunk and no more data is coming.

file_content():
Content of the file to write. This parameter is valid for **File Write Content** command.

Response:
AT Mode:

FTP command	Command Response
Directory List	AT+RSI_FTP_DIR_LIST=<command_type><more><length><data>\r\n **command_type(1 byte):** This filed contains value '6'. **More(1 byte):** Represents whether more response is pending from module or not. 1 – More response is pending 0 – End of response **Length(2 bytes):** Length of current chunk response **Data(variable bytes):** Content of the directory list
File Read	AT+RSI_FTP_FILE=<command_type><more><length><data\r\n **command_type(1 byte):** This filed contains value '7'. **More(1 byte):** Represents whether more response is pending from module or not. 1 – More response is pending 0 – End of response **Length(2 bytes)**: Length of current chunk response

FTP command	Command Response
	Data(Variable bytes): Content of the file
For all other commands	OK<command_type>\r\n **command_type(1 byte):** Type of the FTP command.

Response Parameters:
command_type(1 byte): Type of the FTP command.
More(1 byte): Represents whether more response is pending from module or not.
1 – More response is pending
0 – End of response
Length(2 bytes): Length of current chunk response
Data(Variable bytes): Response data

Relevance:
This command is valid when opermode is 0,1,2 or 6.

Possible Error Codes:
Possible Error codes for this command are 0x0021, 0x0015, 0xFF6B, 0xBB01, 0xBB50, 0xBBD3, 0xBBD4, 0xBBD5, 0xBBD6, 0xBBD9, 0xBBDA, 0xBBDB, 0xBBDC, 0xBBDD, 0xBBDE.

Example:

AT Mode:

1.FTP File Read
```
at+rsi_ftp=1\r\n
at+rsi_ftp=2,4,192.168.0.150,admin,test123,201\r\n
at+rsi_ftp=7,file_read1.txt\r\n
at+rsi_ftp=12\r\n
at+rsi_ftp=2,4,192.168.0.150,admin,test123,201\r\n
at+rsi_ftp=7,file_read2.txt\r\n
at+rsi_ftp=12\r\n
at+rsi_ftp=13\r\n
```

2.FTP File Write
```
at+rsi_ftp=1\r\n
at+rsi_ftp=2,4,192.168.0.150,admin,test123,201\r\n
at+rsi_ftp=8,file_write1.txt\r\n
at+rsi_ftp_file_content=0,This is start of sample data\r\n
at+rsi_ftp_file_content=1,This is end of sample data\r\n
at+rsi_ftp=12\r\n
at+rsi_ftp=13\r\n
```

5.79 SNTP Client

Description:
This section explains different commands to use SNTP client.
This command should be given only after Set IP Parameters command.
Following table explains list of sntp commands and their description.

SNTP Command	Description
Create	Creates SNTP objects. This should be the first command to get time updates from the SNTP server
Get Time	To Get the Current time in seconds
Get Time-Date	To Get the Current time in Time-Date format
Get Server Address	To Get the SNTP server Details.
Get Server Info	To Get the SNTP server Details.

SNTP Command	Description
Delete	To Delete the SNTP client.

- **Create** should be called as a first command to use SNTP.
- Once create is successful **Get Server Address** should be called to get details of the SNTP server.
- Call **Get Time** to get the time in seconds from SNTP server.
- Call **Get Time Date** to get the Time Date format from SNTP server.

Command Format:
AT Mode:
SNTP client has different command types. Based on the command type next parameters will change.
at+rsi_sntp=<command_type>,<remaining parameters>\r\n
Following are available command types.

SNTP command	Command Type	Command Format
Create	**1**	at+rsi_sntp=1,<ip_version>,<IPaddress><sntp method>\r\n **ip_version(1 byte):** IP version to use. 4 – For IPv4 6 – For IPv6 **IP address(IPv4 : 4 byte /IPv6 : 16 byte):** IPv4/IPv6 address for SNTP server to connect. **sntp method(1 byte):** SNTP method to use. 1 – For BroadCast Method 2 – For UniCast Method
Get Time	**2**	at+rsi_sntp=2\r\n
Get Time Date	**3**	at+rsi_sntp=3\r\n
Get Server Address	**4**	at+rsi_sntp=4\r\n
Delete	**5**	at+rsi_sntp=5\r\n
Get Server info	**6**	at+rsi_sntp=6\r\n

command_type(1 byte):
Type of the SNTP command. This parameter is valid for all commands.

Ip_version(1 byte):
IP version to use. This parameter is valid for **create** command.
4 – For IPv4
6 – For IPv6

server_ip_address.ipv4_address(4 byte):
IPv4 address of the SNTP server. This parameter is valid for **create** command.

server_ip_address.ipv6_address(16 byte):
IPv6 address of the SNTP server. This parameter is valid for **create** command.

sntp method(1 byte):
Mode of the SNTP client to run
1 – For Broadcast
2 – For Unicast

Response:
AT Mode:

SNTP command	Command Response
Create	Ok<1>\r\n

SNTP command	Command Response
Get Time	OK<Time in seconds>\r\n
Get Time Date	OK<Time in Ddat-Time format>\r\n
Get Server Address	OK<Ip version><Ip address><sntp method>\r\n Ip_version(1 byte): Ip version of the SNTP server. Ip_address(IPv4 : 4 byte /IPv6 : 16 byte): Ip address of the SNTP server. sntp_method(1 byte): sntp method of the server.
Invalid SNTP server response	AT+RSI_INVALID_SNTP_SERVER=<ip_version><ip_address><sntp_method>\r\n Ip_version(1 byte): Ip version of the SNTP server. Ip_address(IPv4 : 4 byte /IPv6 : 16 byte): Ip address of the SNTP server. sntp_method(1 byte): sntp method of the server.
For remaining commands	OK<Command_type>\r\n Command_type: 1byte. Type of the SNTP command

Response Parameters:
Command_type(1 byte): Type of the SNTP command.
ip_version(1 byte): IP version of the SNTP server.
server_ip_address(IPv4 : 4 byte /IPv6 : 16 byte): IP address of the SNTP server.
sntp method(1 byte): sntp method of the SNTP server

Relevance:
This command is valid when opermode is 0,1,2 or 6.

Possible Error Codes:
Possible Error codes for this command are 0x0021, 0x0015, 0x0074,0xBB10.

Example:
AT Mode:
1.SNTP client:
at+rsi_sntp=1,4,192.168.0.100,2\r\n
at+rsi_sntp=2\r\n
at+rsi_sntp=3\r\n
at+rsi_sntp=4\r\n
at+rsi_sntp=5\r\n

5.80 MDNS and DNS-SD

Description:
This section explains different commands to use MDNS and DNS-SD.
This command should be given only after Set IP Parameters command.
Following table explains list of MDNS and DNS-SD commands and their description.

MDNSD Command	Command Type	Description
Init	1	Creates MDNS Daemon. This should be the first command to initialize.
Register Service	3	To add a service/start service discovery.
Deinit	6	To Stop MDNS responder in module

- **Init** should be called as a first command to use MDNS/DNS-SD.
- Once Init is successful, Add a service using Register Service.
- Reset more bit in Register Service command to indicate module to start MDNS/DNS-SD service.
- To stop MDNS/DNS-SD service use Deinit command.

Command Format:
AT Mode:
MDNS/DNS-SD client has different command types. Based on the command type following parameters will change accordingly.

at+rsi_mdns=<command_type>,<remaining parameters>\r\n
Following are available command types.

MDNSD command	Command Type	Command Format
Init	**1**	at+rsi_mdns=1,<ip_version>,<ttl>,<buffer>\r\n ip_version(1 byte): IP version to use. 4 – For IPv4 6 – For IPv6 ttl(2 byte): Time To Live, Time in seconds for which service should be active. buffer(1000 byte): Host name which is to used as host name in Type-A record.
Register Service	**3**	at+rsi_mdns=3,<port_number>,<ttl>,<more>,<buffer>\r\n port_number(2 byte): Port number on which service which should be added. ttl(2 byte): Time To Live, Time in seconds for which service should be active. more(1 byte): This byte should be set to '1' when there are more services to add. 0 – This is last service, starts MDNS service. 1 – Still more services will be added. buffer(1000 byte): This field contains strings separated by null character(ascii : 0x00(Hex)) Buffer contains 3 string fields separated by NULL, fields are 1. Name to be added in Type-PTR record 2. Name to be added in Type-SRV record(Service name) 3. Text field to be added in Type-TXT record.
Deinit	**6**	at+rsi_mdns=6\r\n

command_type(1 byte):
Type of the MDNSD command.
This parameter is valid for all commands.

ip_version(1 byte):
IP version to use.
This parameter is valid for **Init** command.
4 – For IPv4
6 – For IPv6

ttl(2 byte):
Time To Live, this field is valid for Init command(type - 1), register service command(type - 3).

more(1 byte):
This field is only valid for Register service command.

Response:
AT Mode:

MDNS command	Command Response
Any MDNS Command	OK<Command Type>\r\n

Response Parameters:
Command_type: Type of the MDNSD command.

Relevance:
This command is valid when opermode is 0,1,2 or 6.

Possible Error Codes:
Possible Error codes for this command are 0x0021, 0x0015, 0x0074,0xFF2B.

Example:
AT Mode:
MDNSD Add Service:
at+rsi_mdns=1,4,600,http-wsc_obe.local.\r\n

at+rsi_mdns=3,80,600,0,_http._tcp.local.**NULL** wsc_obe._http._tcp.local**NULL**text_field\r\n
at+rsi_mdns=6\r\n

1. Currently registering only one service is supported
2. IPv4 is only supported for MDNS/DNS-SD service
3. NULL – Is a NULL character with ASCII value '0x00'(HEX)
4. If module joins multicast group,MDNS is not supported

5.81 SMTP Client

Description:
This section explains different commands to use SMTP client. This command should be given only after Set IP Parameters command.
Following table explains list of smtp commands and their description.

SMTP Command	Description
Create	Creates SMTP related thread, this should be the first command to use the SMTP client
Init	To initialize the SMTP client with username, password, from mail address and domain name
Send	To send the mail to recpient
Deinit	To delete the SMTP client

- **Create** should be called as a first command to use SMTP.
- Once create is successful **Init** should be called to initialize the SMTP client with username, password, from address and domain name of the SNMP agent.
- Call **Send** to send the SMTP client mail to the SMTP server agent.
- Call **Deinit** to delete the SMTP client.

Command Format:
AT Mode:
SMTP client has different command types. Based on the command type following parameters will change accordingly.
at+rsi_smtp=<command_type>,<remaining parameters>\r\n
Following are available command types.

SMTP command	Command Type	Command Format
Create	**1**	at+rsi_smtp=1\r\n Creates the SMTP client
Init	**2**	at+rsi_smtp=2,<ip_version>,<ip address>,<auth_type>,<server port>,<smtp buffer>\r\n ip_version(1 byte): IP version to use. 4 – For IPv4 6 – For IPv6 ip_address(IPv4 : 4 byte /IPv6 : 16 byte): Ipv4/Ipv6 address of the SMTP server agent. auth_type(1 byte): Authentication type of the SMTP server. 1. For Auth-Login type 2. For Auth_plain type server_port(4 byte): SMTP server agent port number smtp_buffer(1024 byte):smtp buffer contains server's username, password, from mail address, smtp server local domain name.
Mail send	**3**	at+rsi_smtp=3,<smtp_feature>,<mail body_length>,<smtp buffer>\r\n smtp_feature(1 byte): smtp feature bitmap BIT(0): Low priority mail BIT(1): Normal priority mail

SMTP command	Command Type	Command Format
		BIT(2): High priority mail BIT(3): Smtp extended header feature smtp_mail_body_length(2 byte):Length of the mail body smtp_buffer(1024 byte): smtp buffer contains recepient mail address, mail subject line, mail body, smtp extended header.
Deinit	**4**	at+rsi_smtp=4\r\n To delete the smtp client

command_type(1 byte): Type of the SMTP command. This parameter is valid for all commands.

ip_version(1 byte):
IP version to use. This parameter is valid for **Init** command.
4 – For IPv4
6 – For IPv6

ipv4_address(4 byte) :
Ipv4 address of the SMTP agent.

ipv6_address(16 byte):
Ipv6 address of the SMTP agent.

auth_type(1 byte):
Authentication method used by the SMTP server agent.
1 for AUTH_PLAIN
3 for AUTH_LOGIN

server_port(4 byte):
SMTP server agent port number.

smtp_feature(1 byte):
SMTP client feature bit map

BIT(0):
MAIL_PRIORITY_LOW

BIT(1):
MAIL_PRIORITY_NORMAL

BIT(2):
MAIL_PRIORITY_HIGH

BIT(3):
To Enable SMTP_EXTENDED_HEADER feature

smtp_client_mail_body_length(2 byte):
Length of the SMTP client mail body.

smtp_buffer(1024 byte):
smtp_buffer contains remaining parameters based on command type.

Note:

1. Maximum supported length for username, password, domain name, recipient mail address, from mail address is 100 bytes each excluding NULL character
2. Maximum supported length for recipient mail address, mail subject line, mail body together is 1024 bytes including NULL characters (3 bytes)
3. Maximum supported length for mail subject line is 750 bytes
4. Maximum supported length for mail body is 950 bytes

Response Parameters:
Command_type: Type of the SMTP command.

Relevance:
This command is valid when opermode is 0,1,2 or 6.

Possible Error Codes:
Possible Error codes for this command are 0x0021, 0x000e, 0xBBAA, 0xBBAD, 0x002C, 0x0015, 0xBBA5, 0xBB21, 0x003E, 0xBBB2, 0x003E, 0xBBA5, 0xBBA3, 0xBA0, 0xBBA1, 0xBBA2,OxBBA4, OxBBA6, 0xBBA7, 0xBBA8, 0xBBA9, 0xBBAA, 0xBBAB, 0xBBAC, 0xBBAD, 0xBBAE, 0xBBAF, 0xBBB0, 0xBBB1, 0x0025 ,0xFF74, 0xBBF0.

Example:
AT Mode:
SMTP client mail send Service:
at+rsi_smtp=1\r\n
at+rsi_smtp=2,4,192.168.0.100,3,25,
username**NULL**password**NULL** Silicon1@Siliconlabs.com**NULL**mail.Siliconlabs.com**NULL**\r\n
at+rsi_smtp=3,3,4,Silicon2@Siliconlabs.com**NULL**subjectline**NULL**body\r\n
at+rsi_smtp=4\r\n

5.82 POP3 Client

Description:
This section explains different commands to use POP3 client. This command should be given only after Set IP Parameters command.
Following table explains list of pop3 commands and their description.

POP3 Command	Description
Session Create	Creates POP3 client Daemon, this should be the first command to use the POP3 client. This command initiates POP3 client to establish a connection with POP3 server and then gets authenticated to it.
Get mail stats	To get the total number of mails count and size
Get mail list	To get the size of the mail for the passed index
Retrieve mail	To retrive the mail content for the passed mail index
Mark mail	To mark a mail as deleted for the passed mail index
Unamark mail	To unmark(reset) all the marked(deleted) mails in the current session.
get server status	To get the pop3 server status
session delete	To delete pop3 client session

- **Create** should be called as a first command to use POP3.
- Once create is successful call **get mail** stats to get the mail statistics i.e. number of mails present in server and the size of total mails
- Call get **mail list** to get the size of the mail for the passed index
- Call **Retrieve mail** to get the mail content for the passed mail index
- Call **Mark mail** to delete a particular mail in the POP3 server, by passing the mail index
- Call **Unmark mail** to reset all the mails in the current session
- Call **get server status** to get the pop3 server status
- Call **session delete** to delete the pop3 client session

Note:
Retrieve mail command should be issued only after get mail list command, Providing index of the intended mail.

Note:
Maximum allowed username and password length is 101 (Including NULL character)

Command Format:
AT Mode:
POP3 client has different command types. Based on the command type following parameters will change accordingly.
at+rsi_pop3=<command_type>,<remaining parameters>\r\n
Following are available command types.

POP3 command	Command Type	Command Format
Session Create	**1**	at+rsi_pop3=1,<ip_version>,>,<ipaddress>,<serverport,<auth_type>,<username>,<password>\r\n ip_version(1 byte): IP version to use. 4 – For IPv4 6 – For IPv6 ip_address(IPv4 : 4 byte /IPv6 : 16 byte): Ipv4/Ipv6 address of the POP3 server. server_port(2 byte): POP3 server port number auth_type(1 byte): Authentication type of the POP3 server. For Auth-Login type For Auth_plain type username(101 byte): username for POP3 server authentication. password(101 byte): password for POP3 server authentication.
Mail stats	**2**	at+rsi_pop3=2\r\n To get the mail stats from the POP3 server.
Mail list	**3**	at+rsi_pop3=3,<mail index>\r\n mail index(2 byte) : Index of the particular mail which is used to get the size of the mail.
Retrieve mail	**4**	at+rsi_pop3=4,<mail index>\r\n mail index(2 byte) : Index of the particular mail which is used in the mail list command.
Mark mail	**5**	at+rsi_pop3=5,<mail index>\r\n mail index(2 byte) : Index of the particular mail which is used for deletion by passing the index
Unmark mail	**6**	at+rsi_pop3=6\r\n To reset all the marked mails in the current session
server status	**7**	at+rsi_pop3=7\r\n To get the POP3 server status
session delete	**8**	at+rsi_pop3=8\r\n To delete the POP3 client session

command_type: Type of the POP3 command. This parameter is valid for all commands.

ip_version(1 byte): IP version to use. This parameter is valid for **session create** command.
4 – For IPv4
6 – For IPv6

ipv4_address(4 bytes):
Ipv4 address of the POP3 server.

ipv6_address(16 bytes):
Ipv6 address of the POP3 server.

auth_type(1 byte):
Authentication method used by the SMTP server agent.

server_port_number(2 byte):
POP3 server agent port number.

username(101 byte):
username for POP3 server authentication

password(101 byte):
password for POP3 server authentication

pop3_client_mail_index(2 byte):
POP3 mail index number

Command_type(1 byte):

Type of the POP3 command.

mail_count(2 byte):
Total mails count/ mail index

size(4 byte):
Total size of the mails/ size of the particular mail

more(4 bytes):
This indicates whether more POP3 data for the retrieve command is pending or not.
0–More data is pending. Further interrupts may be raised by the module till all the data is transferred to the Host.
1– End of POP3 mail content.

length(2 byte):
Length of the current pop3 mail content chunk

data(1000 byte):
POP3 client mail content buffer

Relevance:
This command is valid when opermode is 0,1,2 or 6.

Possible Error Codes:
Possible Error codes for this command are 0x0021, 0x0015, 0x0025,0xFF74,0xBB87,0xBBFF,BBC5.

Example:
AT Mode:
POP3 client mail receive Service:
at+rsi_pop3=1,4,192.168.0.100,3,25,testuser,test123\r\n
at+rsi_pop3=2\r\n
at+rsi_pop3=3,100\r\n
at+rsi_pop3=4,100\r\n
at+rsi_pop3=5,100\r\n
at+rsi_pop3=6\r\n
at+rsi_pop3=7\r\n
at+rsi_pop3=8\r\n

5.83 IAP Init

Description:
This command initializes the interface between RS9116W and IAP(iPod Accesory protocol) co processor before starting the Apple Authentication process.

Command:
AT Mode:
N/A

Command Parameters:
No parameters

Response:
AT Mode:
N/A

Relevance:
This command is relevant in AP mode and Station mode

Note:
This command is required only if IAP co-processor is mounted on RS9116W chip with I2C interface. This command is not required if IAP chip is mounted on the host MCU. For more information, contact Silicon Labs.

5.84 Load MFI IE

Description:
This command is used to load the MFI Information Element in the beacon generated by the WAC (Wireless accessory configuration) server.

Command:
AT Mode:
N/A

Command Parameters:
ie_len(1 byte): Length of the MFI information element need to be loaded in the beacon of RS9116W Accesory.
ie(200 byte): array of MFI information element.

Response:
AT Mode:
N/A

Relevance:
This command is relevant in AP mode

Possible Error Codes:
Possible Error codes for this command are 0x0021, 0x0015, 0x002C.

5.85 Over the Air Firmware Upgradation

Description:
This command upgrades the firmware on the module over a TCP socket. It is designed to work with a remote application.

Command Format:

AT Mode:

```
at+rsi_otaf=<ip_version>, <destIPaddr>, <server_port>,<chunk_number>, <rx_timeout> ,<tcp_retry_count>\r\n
```

Command Parameters:
ip_version(1 byte): IP version used, either 4 or 6.

destIPaddr(IPv4 : 4 bytes /IPv6 : 16 bytes):
IP Address of the target server.

server_port(4 bytes):
destination port. Value ranges from 1024 to 49151.

Note:

User can give any Port number from the above range except 30000,as that port number is reserved for specific feature.

chunk_number(2 bytes):
chunk number of the firmware to be upgraded. Initially keep it as 1.

rx_timeout(2 bytes):
To configure timeout.

tcp_retry_count(2 bytes):
To configure tcp retransmissions count.

Response:

After successful up gradation firmware gives a success indication with an asynchronous message as "AT+RSI_FWUPSUCCESS\r\n".
In addition, "reach end of file" message comes at server side.

On firmware upgrade failure, firmware gives error message as
"ERROR<Error_Code><Number of chunk where up gradation stopped>".

User needs to give same command again from the chunk number specified here.

Possible error codes:
0xBB01, 0xBB38

Example:
at+rsi_otaf = 4, 192.168.0.100, 5001, 1, 100, 10\r\n

Response:

AT+RSI_FWUPSUCCESS on success up gradation.

ERROR<Error_Code><number of chunk where up gradation stopped>
ERROR<Error_Code = 0x01 0xBB><number of chunk where upgradation stopped = 0xD2 0x04>

After getting this error give command
at+rsi_otaf = 4,192.168.0.100, 5001, 1234, 100, 10\r\n

Note:

It is recommended to finish pending transactions (like data transfers and disconnect sockets if any) and disable power save before starting firmware upgrade process.

5.86 HTTP OTAF

Description:
This command is used to transmit HTTP OTAF request from the module to a remote HTTP server.
A subsequent HTTP OTAF request can be issued only after receiving the response of the previously issued HTTP OTAF request. Module acts as a HTTP client when this command is issued.

Once this command get issued the Firmware file get download and on module reset firmware get loaded

Command Format:
AT Mode:
at+rsi_httpota=< https_enable >,< http_port >,< User_name >,< Password >,< Host_name >,< Ip_address >,< url >,< Extended_header >\r\n

Command Parameters:
https_enable:

Set BIT(0) to enable HTTPS.
Set BIT(1) to enable NULL delimiter for HTTP buffer instead of comma
Set BIT(2) to use SSL TLS 1.0 version if HTTPS is enabled.
Set BIT(3) to use SSL TLS 1.2 version if HTTPS is enabled.
Set BIT(4) to use SSL TLS 1.1 version if HTTPS is enabled.

Note:
If SSL is enabled by default, it will use SSL TLS 1.0 and TLS 1.2 version. BIT (2) and BIT(3) are valid only when HTTPS is enabled.

Note:
If SSL is enabled module will use same SSL version until module reboots.

Set BIT (5) to enable http_post data feature

Set BIT (6) to enable HTTP version 1.1.

Set BIT (7) to enable user defined http_content type in Flags.

http_port:
HTTP server port number.
If this is not mentioned, default port numbers will be used.

User_name:
User_name for HTTP/HTTPS server authentication. Default user name is 'admin'.

Password:
Password for HTTP/HTTPS server authentication. Default password is 'admin'.

Host_name:
Host name of the HTTP/HTTPS server.

Ip_address:
IPv4/IPv6 address of HTTP/HTTPS server.

url: requested URL.

Extended_header:
The Purpose of this is to append user configurable header fields to the default HTTP/HTTPS header. To write extended header through 'at' command, user must use 'Data stuffing' mentioned separately in this document as well mentioning here also in context of extended header. extended header can have multiple header fields each ended by \r\n(oxd oxa) but here \r\n is our delimiter for whole 'at' command so use data stuffing and replace all \r\n(0xd 0xa) by 0xdb oxdc besides delimiter(\r\n).follow example given below.

default http header contains the following:

Default http Header include:

1.Contentent - Type: It describes about the type of the file which user want to send like html,txt,gif etc**.**

2.Content - Length: It tells about the length of the text which User is using.

The above two mentioned fields are created in the header by the firmware.

Note:

- Maximum supported length for User_name, Password together is 278 bytes.
- Maximum supported length for Buffer is (872-(length of User_name+length of Password)) bytes excluding delimiters.
- If username, password, hostname and extended http headers are not required, user should send empty string separated by delimiter.
- If content of any field contains comma (,) then NULL delimiter should be used.

For example,
https_enable = BIT(0)
http_port = 443
Username : username
Password : password
Hostname: www.google.com
IP = 192.168.40.50
URL=/index.html
Extended HTTP Header = ContentType: **html**ÛÜ

Response:
Module may give http response in multiple chunks for a single HTTP OTAF request.

AT Mode:

Result	Response	Description
Success	AT+RSI_ HTTPOTARSP=< Upgrade Success>\r\n	After HTTP response, for user to understand the upgrade status, below response is added with " Upgrade Success" • AT+RSI_HTTPOTARSP=<More><Status_Code><Data Offset><Data Length><Upgrade Success> • The string AT+RSI_HTTPOTARSP is in uppercase ASCII.
Failure	ERROR<Error_code>\r\n	Failure case with error code, there would be multiple reasons for Upgrade Failure, Refer **Expected Error Codes** Section
	AT+RSI_ HTTPOTARSP=< Upgrade Failed>\r\n	Upgrade Failure response

Response Parameters:
More (2 bytes):
This indicates whether more HTTP data for the HTTP GET request is pending or not.
0–More data is pending. Further interrupts may be raised by the module till all the data is transferred to the Host.
1– End of HTTP data.

Status code (2 bytes):
Provided the HTTP status code as received in the response packet such as 200, 201, 404 etc. A status_code equal to 0 indicates that there was no HTTP header in the received packet, probably a continuation of the frame body received in the previous chunk.

Offset(4 bytes): Always contains '0'.

data_len(4 bytes): data length in current chunk.

Data(Maximum 1400 bytes): Actual http data.

Possible error codes:
Possible error codes for this command are 0x0015,0x0021, 0x0025, 0x002C, 0xFF74, 0xBBF0

Relevance:
This command is relevant when the module is configured in Operating Mode 0,1, 2 and 6.

Example:
AT Mode:
https_enable : 0
http_port : 80
Username : username
Password : password
Host_name : www.google.com
IP Address: 192.168.40.86
URL: /index.html
Extended HTTP Header: ContentType: **html**ÛÜ
the below command is used
at+rsi_httpget=0,80,username,password,hostname,192.168.40.86,/index.html,ContentType: **html**ÛÜ\r\n

5.87 Read MFI Authentication Certificate

Description:
This command is used to read the IAP co-processor Authentication certificate which is used in the authentication process while configuring accessory using MFI WAC server.

Command:
AT Mode:
N/A

Response:
AT Mode:
N/A

Relevance:
This command is relevant in AP and station mode
Possible Error Codes:
Possible Error codes for this command are 0x0021, 0x0015, 0x002C.

Note:

This command is required only if IAP co-processor is mounted on RS9116W with I2C interface. This command is not required if IAP chip is mounted on the host MCU. For more information, contact Silicon Labs.

5.88 Storing Configuration Parameters

In client mode:
The module can connect to a pre-configured access point after it boots up (called auto-join in these sections). This feature facilitates fast connection to a known network.

In Access Point mode:
The module can be configured to come up as an Access Point every time it boots-up (called auto-create in these sections)
The feature is valid in operating modes 0, 2 and 6.

5.88.1 Storing Configuration Parameters in Client mode

5.88.2 Store Configuration in Flash Memory

Description:
This command is used to save the parameters into non-volatile memory which are used either to join to an Access point (auto-join mode) or to create an Access point (auto-create mode) .

Command Format:

AT Mode:
at+rsi_cfgsave\r\n

AT Mode:

OK	Successful execution
ERROR	Failure.

Relevance:

This command is valid when opermode is 0,1,2 or 6.

Possible error codes:
Possible error codes for this command are 0x0021, 0x0025, 0x002C.

5.88.3 Enable auto-join to AP or Auto-create AP

Description:
This command is used to enable or disable the feature of auto-join or auto-create on power up.

Command Format:

AT Mode:
at+rsi_cfgenable=< cfg_enable >\r\n

Command Parameters:

cfg_enable(1 byte):
0 - Disables auto-join or auto-create
1 - Enables auto-join or auto-create

Response:

AT Mode:

OK	For response payload parameters description, refer to the section **Store configuration structure parameters**.
ERROR	Failure.

Relevance:

This command is valid when opermode is 0,1,2 or 6.

Possible error codes:
Possible error codes for this command are 0x0021, 0x0025, 0x002C.

5.88.4 Get Information about Stored Configuration

Description:
This command is used to get the configuration values that have been stored in the module's memory which are used in auto-join or auto-create modes.

Command Format:

AT Mode:
at+rsi_cfgget?

Response:
AT Mode:

OK<response payload>	For response payload parameters description, refer to section **Store configuration structure parameters.**
ERROR	Failure.

Note:
Transparent mode parameters are only valid in UART interface in AT command mode only,

Note:
After firmware upgradation, previous saved configuration may be lost due to checksum fail.

Response Parameters:
For response payload parameters description, refer section Store configuration structure parameters.

Relevance:
This command is valid when opermode is 0,1,2 or 6.

Possible error codes:
Possible error codes for this command are 0x0021, 0x0025, 0x002C.

5.88.5 Store configuration from User

Description:
This command is used to give the configuration values which are supposed to be stored in the module's non-volatile memory and that are used in auto-join or auto-create modes.
This command can be given at any time after opermode is command is given.

Command Format:

AT Mode:

at+rsi_usercfg=<length_of_payload>,<Store configuration parameters >\r\n

Command Parameters:
length_of_payload(): Length in bytes of the Store configuration parameters field.
Store configuration parameters: Store configuration parameter in hex format.

For payload parameters description, refer to the section **Store Configuration structure parameters**.

Response:

AT Mode:

OK	Successful execution
ERROR	Failure.

Relevance:

This command is valid when opermode is 0,1,2 or 6.

Possible Error Codes:
Possible error codes for this 0x003D,0x0021,0x002C,0x0025,0x0015.

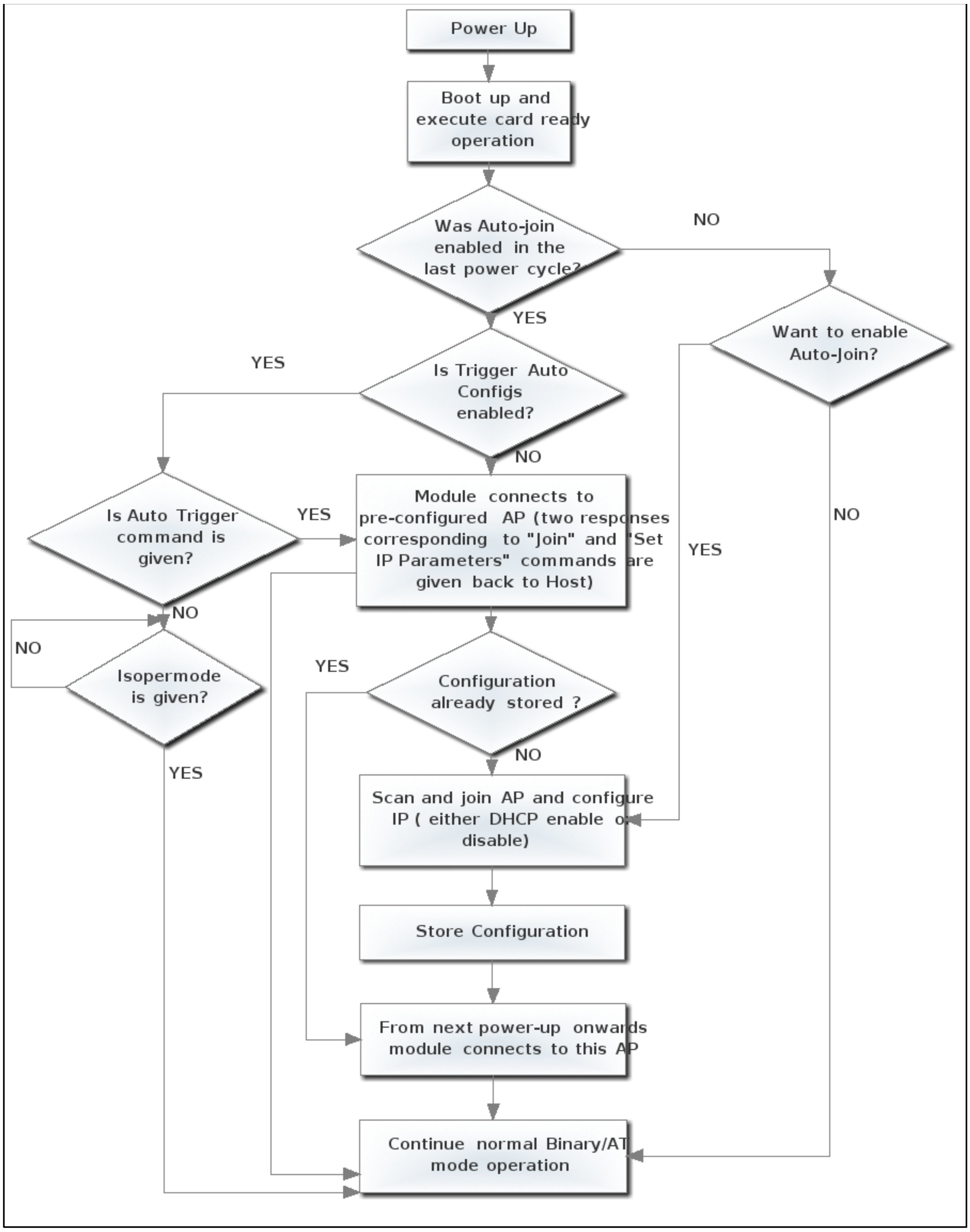

Power Up
Boot up and execute card ready operation
Was Auto-join enabled in the last power cycle?
NO
YES
Want to enable Auto-Join?
YES
Is Trigger Auto Configs enabled?
NO
Is Auto Trigger command is given?
YES
Module connects to pre-configured AP (two responses corresponding to "Join" and "Set IP Parameters" commands are given back to Host)
NO
YES
NO
NO
Isopermode is given?
YES
Configuration already stored ?
NO
YES
Scan and join AP and configure IP (either DHCP enable or disable)
Store Configuration
From next power-up onwards module connects to this AP
Continue normal Binary/AT mode operation

Figure 26: Connecting to Pre-configured AP

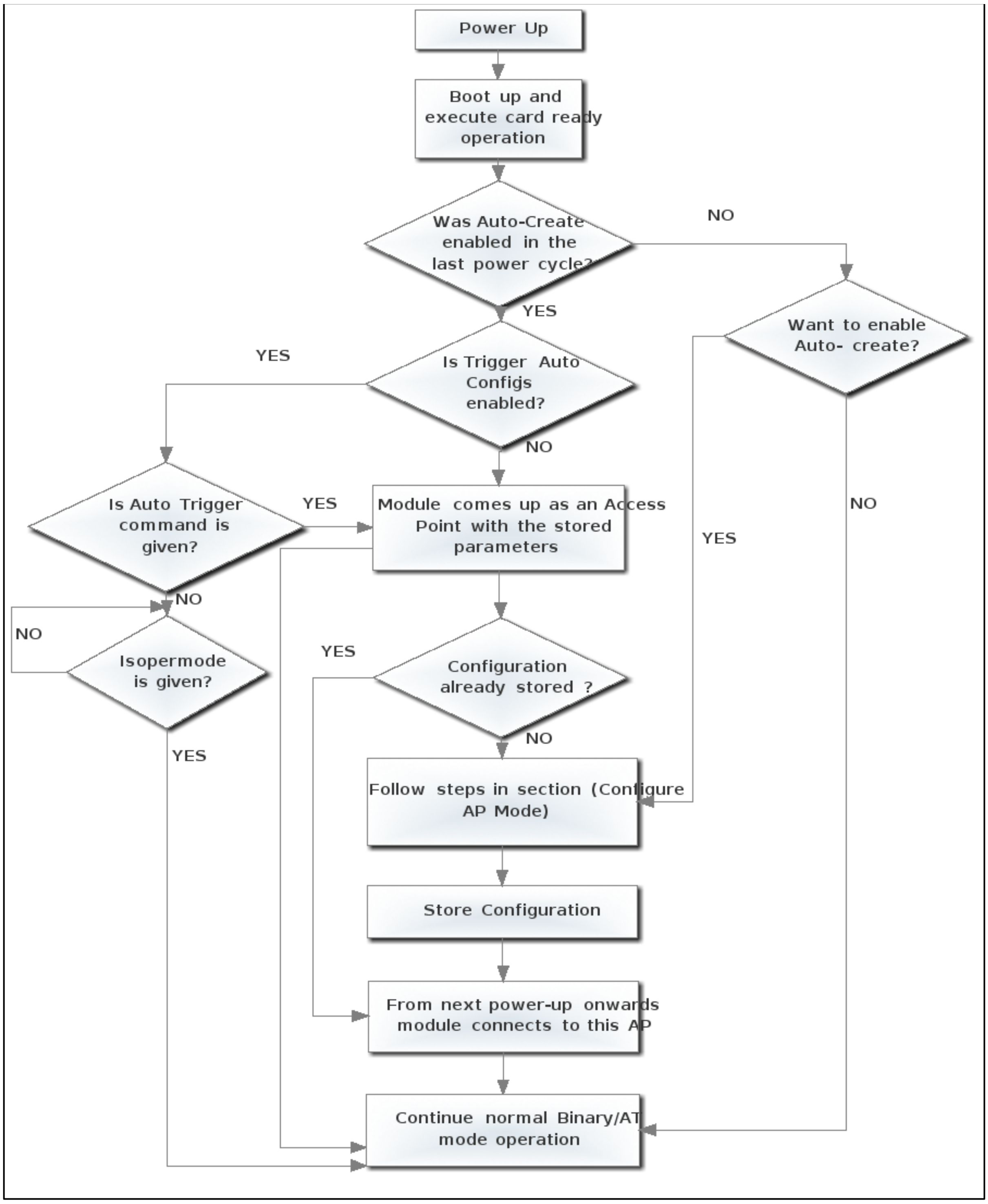

Power Up
Boot up and execute card ready operation
Was Auto-Create enabled in the last power cycle?
NO
YES
Want to enable Auto- create?
Is Trigger Auto Configs enabled?
YES
NO
Is Auto Trigger command is given?
YES
Module comes up as an Access Point with the stored parameters
NO
YES
NO
Isopermode is given?
NO
YES
Configuration already stored ?
YES
NO
Follow steps in section (Configure AP Mode)
Store Configuration
From next power-up onwards module connects to this AP
Continue normal Binary/AT mode operation

Figure 27: Creating Preconfigured AP

5.89 Store Configuration Structure Parameters

The parameters/variables which are used in store configuration are explained in this section. Same structure is used in storing and getting the configuration parameters.

Transparent mode parameters are valid only in AT command mode on UART interface.

cfg_enable (1 byte):
0x00- auto-join or auto-create modes are disabled
0x01- auto-join or auto-create modes are enabled

opermode (4 bytes):

Oper_mode:

Sets the mode of operation. oper_mode contains two parts <wifi_oper_mode, coex_mode>. Lower two bytes represent wifi_oper_mode and higher two bytes represent coex_modes.

oper_mode = ((wifi_oper_mode) | (coex_mode << 16))

Wifi_oper_mode values:
0 - WiFi Client Mode. The module works as a normal client that can connect to an Access Point with different security modes other than enterprise security.
2 – Enterprise Security Client Mode. The module works as a client that can connect to an Access Point with WPA/WPA2-Enterprise security.
6 – Access Point mode**.** In this mode, the module acts as an Access Point, depending on the inputs supplied for the command "Configure AP Mode". In Access Point mode, a maximum of 8 client devices are supported.
8 - PER Mode. This mode is used for calculating packet error rate and mostly used during RF certification tests.

coex_mode bit values: enables respective protocol
BIT 0 : Enable/Disable WLAN mode.
0 – Disable WLAN mode
1 – Enable WLAN mode

BIT 1 : Enable/Disable Zigbee mode (currently not supported)
0 – Disable Zigbee mode
1 – Enable Zigbee mode

BIT 2 : Enable/Disable BT mode.
0 – Disable BT mode
1 – Enable BT mode

BIT 3 : Enable/Disable BTLE mode.
0 – Disable BTLE mode
1 – Enable BTLE mode

Note:
In BTLE mode, need to enable BT mode also.
Following table represents possible coex modes supported:

Table 15 Coex Modes Supported

	Description
0	WLAN only mode
3	WLAN and Zigbee coexistence mode*
5	WLAN and BT coexistence mode
13	WLAN and BTLE coexistence mode

Note:

ZigBee is not supported currently.

Note:

1. In coexistence mode (3,5,13) module supports only WLAN client mode (Open mode, PSK security).
2. In coexistence mode (0) module supports all WLAN modes and embedded TCP/IP stack.

feature_bit_map: this bitmap used to enable following WLAN features:

feature_bit_map[0]- To enable open mode
0 - Open Mode Disabled
1 - Open Mode enabled (No Security)

feature_bit_map[1]-To enable PSK security
0 - PSK security disabled
1 - PSK security enabled

feature_bit_map[2]-To enable Aggregation
0 - Aggregation disabled
1 - Aggregation enabled

feature_bit_map[3]-To enable LP GPIO hand shake
0 – LP GPIO hand shake disabled
1 – LP GPIO hand shake enabled

feature_bit_map[4]-To enable ULP GPIO hand shake
0 - ULP GPIO hand shake disabled
1 - ULP GPIO hand shake enabled

feature_bit_map[5]-Reserved

feature_bit_map[6]-Reserved

feature_bit_map[7]-To disable WPS support
0 – WPS enable
1 - WPS disable

feature_bit_map[8:31]- Reserved. Should set to be '0'

Note:

feature_bit_map[0], feature_bit_map[1] are valid only in Wi-Fi client mode.

tcp_ip_feature_bit_map: To enable TCP/IP related features.

tcp_ip_feature_bit_map[0]- to enable TCP/IP bypass
0 – TCP/IP bypass mode disabled
1 – TCP/IP bypass mode enabled

tcp_ip_feature_bit_map[1]- to enable http server
0 - HTTP server disabled
1 - HTTP server enabled

tcp_ip_feature_bit_map[2]- to enable DHCPv4 client
0 - DHCPv4 client disabled
1 - DHCPv4 client enabled

tcp_ip_feature_bit_map[3]- to enable DHCPv6 client
0 - DHCPv6 client disabled
1 - DHCPv6 client ewnabled

tcp_ip_feature_bit_map[4]- to enable DHCPv4 server
0 - DHCPv4 server disabled
1 - DHCPv4 server enabled

tcp_ip_feature_bit_map[5]- to enable DHCPv6 server
0 - DHCPv6 server disabled
1 - DHCPv6 server enabled

tcp_ip_feature_bit_map[6]- To enable Dynamic update of web pages (JSON objects)
0 - JSON objects disabled
1 - JSON objects enabled

tcp_ip_feature_bit_map[7]- to enable HTTP client
0 - To disable HTTP client
1 - To enable HTTP client

tcp_ip_feature_bit_map[8]- to enable DNS client
0 - To disable DNS client
1 - To enable DNS client

tcp_ip_feature_bit_map[9]- to enable SNMP agent
0 - To disable SNMP agent
1 - To enable SNMP agent

tcp_ip_feature_bit_map[10]- to enable SSL
0 - To disable SSL
1 - To enable SSL

tcp_ip_feature_bit_map[11]- to enable PING from module(ICMP)
0 - To disable ICMP
1 - To enable ICMP

tcp_ip_feature_bit_map[12]- to enable HTTPS Server
0 - To disable HTTPS Server
1 - To enable HTTPS Server

tcp_ip_feature_bit_map[14]- to send configuration details to host on submitting configurations on wireless configuration page
0 - Do not send configuration details to host
1 - Send configuration details to host

tcp_ip_feature_bit_map[15]- to enable FTP client
0 - To disable FTP client
1 - To enable FTP client

tcp_ip_feature_bit_map[16]- To enable SNTP client
0 - To disable SNTP client
1 - To enable SNTP client

tcp_ip_feature_bit_map[17]- To enable IPv6 mode
0 - To disable IPv6 mode
1 - To enable IPv6 mode
IPv6 will also get enabled if DHCP v6 client/DHCP v6 server is enabled irrespective of tcp_ip_feature_bit_map[17].

tcp_ip_feature_bit_map[19]- To MDNS and DNS-SD
0 - To disable MDNS and DNS-SD
1 - To Enable MDNS and DNS-SD

tcp_ip_feature_bit_map[20]- To enable SMTP client
0 - To disable SMTP client
1 - To Enable SMTP client

tcp_ip_feature_bit_map[21 - 24]- To select no of sockets
tcp_ip_feature_bit_map[25]- To select Single SSL socket
0 – selecting single socket is Disabled

1 – Selecting single socket is enabled

Note:

By default, two SSL sockets are supported.

tcp_ip_feature_bit_map[26]- To allow loading Private & Public certificates
0 – Disable loading private & public certificates

1- Allow loading private & public certificates

tcp_ip_feature_bit_map[27]- To load SSL certificate on to the RAM

tcp_ip_feature_bit_map[28]- To enable TCP-IP data packet Dump on UART2

tcp_ip_feature_bit_map[29]- To enable POP3 client
0 - To disable POP3 client
1 - To Enable POP3 client

tcp_ip_feature_bit_map[13] set to '0'.

tcp_ip_feature_bit_map[30]- To enable OTAF(On The Air Firmware)

upgradation.

tcp_ip_feature_bit_map[31]- This bit is used to enable the tcp_ip extention valid feature bitmap.

1 – To enable Extended tcp_ip feature bitmap

0 – To disable Extended tcp_ip feature bitmap

> **Note**:
>
> SSL (tcp_ip_feature_bit_map[10], tcp_ip_feature_bit_map[12]) is supported only in opermode 0

custom_feature_bit_map: This bitmap used to enable following custom features:

BIT[2]: If this bit is set to '1', the DHCP server behavior, when the module is in AP mode, changes. The DHCP server, when it assigns IP addresses to the client nodes, does not send out a Gateway address, and sends only the assigned IP and Subnet values to the client. It is highly recommended to keep this value at '0' as the changed behavior is required in only very specialized use cases and not in normal AP functionality. The default value of this bit is '0'.

BIT [5]: If this bit is set to '1', Hidden SSID is enabled in case of AP mode. The default value of this bit is '0'.

BIT [6]:To enable/disable DNS server IP address in DHCP offer response in AP mode.

1- In AP mode, DHCP server sends DNS server IP address in DHCP offer

0- Not to include DNS server address in DHCP offer response

BIT[8]: Enable/Disable DFS channel passive scan support
1- Enable
0-Disable

BIT[10]: Used to enable/disable **Asynchronous messages** to host to indicate the module state.
1- Enable asynchronous message to host
0-Disable asynchronous message to host

BIT[11]: To enable/disable packet pending Wake on wireless indication in UART mode

1- Enable packet pending indication

0- Disable packet pending indication

BIT[12]: Used to bypass AP blacklist feature.
1 – Bypass AP black list feature
0 – Enable AP black list feature

BIT[13-16]: Used to set the maximum number of stations or client to support in AP. Possible values are 1 to 16 in AP mode.

> **Note:**
>
> If these bits are not set, default maximum clients supported is set to 4.

BIT[17] : to select between de-authentication or Null data (with power management bit set) based roaming, Depending on selected method station will roam from connected AP to newly selected AP.
0 - To enable de-authentication based roaming
1 - To enable Null data based roaming

BIT[18]: Reserved

BIT[19]: Reserved

BIT[20]: Used to start/stop auto connection process on bootup, until host triggers it using Trigger Auto Configuration command
0 - Enable
1 - Disable

BIT[22]: Used to enable per station power save packet buffer limit. When enabled, only two packets per station will be buffered when station is in power save
1 – Enable
0 – Disable

BIT[23] : To enable/disable HTTP/HTTPs authentication
1 - Enable
0 – Disable

BIT[24]: To enable/disable higher clock frequency in module to improve throughputs
1 - Enable
0 – Disable

BIT[25]: To give HTTP server credentials to host in get configuration command
1 – To include HTTP server credentials in get configuration command response
0 – To exclude HTTP server credentials in get configuration command response

BIT[26]: To accept or reject new connection request when maximum clients are connected in case of LTCP.
1 - Reject
0 – Accept

By default, this bit value is zero.

When BIT[26] is zero: For a LTCP socket when maximum clients are connected if a new connection request is received, then this connection request will not be rejected. Instead module will maintain this connection request in LTCP pending list.

This request will be served when any of the connected client is disconnected.

When BIT[26] is set: For a LTCP socket when maximum clients are connected if a new connection request is received, then this connection request will be rejected immediately. Module will not maintain this connection request in LTCP pending list.

Note:

The below parameters are valid based on the operating mode given

Band (1 byte):
0x00- Module configured to operate in 2.4 GHz
0x01- Module configured to operate in 5 GHz
0x02- Dual band (2.4 Ghz and 5Ghz). Dual band is valid in station mode.

scan_feature_bitmap(1 byte): Scan feature bitmap

BIT[0]: To enable/disable quick scan feature.
1 - To enable quick scan feature.
0 - To disable quick scan feature. BIT[1]-BIT[7]: Reserved.

Join_ssid (32 bytes):

SSID of the AP configured in auto-join or in auto-create mode. If the actual length is not 32 bytes, 0x00 filler bytes are appended to make the length 32 bytes.

uRate (1 byte): Data rate to be configured in the module.

Please refer the **PER Mode** command section, for the data rates supported by the Silicon module

uTXPower (1 byte): Tx power to be configured in the module.

At 2.4GHz
0– Low power (7+/-1) dBm
1– Medium power (10 +/-1)dBm
2– High power (18 +/- 2)dBm

At 5 GHz
0– Low power (5+/-1) dBm
1– Medium power (7 +/-1) dBm
2– High power (12 +/- 2) dBm

join_feature_bitmap(1 byte):

BIT[0]: To enable b/g only mode in station mode, host has to set this bit.
0 – b/g/n mode enabled in station mode
1 – b/g only mode enabled in station mode

BIT[1]: To take listen interval from join command.
0 – Listen interval invalid
1 – Listen interval valid

BIT[2]:To enable/disable quick join feature.
1 - To enable quick join feature.
0 - To disable quick join feature.

BIT[3]-BIT[7]: Reserved.
Reserved_1(1byte):Reserved
Scan_ssid_len(1 byte):Scan ssid length
keys_restore(1byte):Reserved
Csec_mode(1byte): Security mode of access point to connect in auto join mode or security mode of DUT in auto create mode.This variable is used to define the security mode of the Access point to which module is supposed to connect.

BIT[7]:To enable/disable listen interval from power save command

1 - To enable listen interval.

0 - To disable listen interval.

Possible values:
0 – Open mode
1 – WPA security
2 – WPA2 Security

Other values are assumed to be don't care.

psk (64 bytes): Pre shared key of the access point to which module wants to associate in auto-join or auto-create mode. Filler bytes of 0x00 are added to make it 64 bytes if the original PSK is less than 64 bytes.

Scan_ssid (32 bytes): SSID of the AP to be scanned in auto-join. If the actual length is not 32 bytes, 0x00 filler bytes are appended to make the length 32 bytes.

Scan_cnum(1 byte):channel number to be scanned in auto join mode. Refer to the PER Mode command section for the supported channels in 2.4GHz and 5 GHz band

dhcp_enable (1 byte):
0x00- DHCP client is disabled in module (auto-join mode)
0x01- DHCP client is enabled in module (auto-join mode)

ip (4 bytes): Static IP configured in the module in auto-join or auto-create mode. For auto-join mode, this is valid when dhcp_enable is 0.

Sn_mask(4 bytes): Subnet mask, this is valid only if dhcp_enable is 0.

dgw(4 bytes): Default gateway, this is valid only if dhcp_enable is 0.

eapMethod(32 bytes): Should be one of among TLS, TTLS, FAST or PEAP, ASCII character string used to configure the module in Enterprise security mode

innerMethod(32 bytes): Should be fixed to MSCHAPV2, ASCII character string.This parameter is used to configure the module in Enterprise security mode.

user_identity (64 bytes): User ID in enterprise security mode.

Passwd (128 bytes): Password configured for enterprise security. Refer to the parameter *Password* in the command *at+rsi_eap.* Filler bytes of 0x00 are used to make the length 128 bytes, of the original length is less than 128 bytes.

Pmk(32 bytes): PMK key

channel_no(2 bytes): The channel in which the AP would operate. Refer to the PER Mode command section for more details on the channels supported by the module. A value of '0' is not allowed.

Ssid(32 bytes): SSID of the AP to be created

security_type(1 byte): Security type of AP to be configured
0-Open
1-WPA
2-WPA2

encryp_mode(1 byte): Encryption type.
0-Open

1-TKIP
2-CCMP

Psk(64 bytes): PSK of the AP in security mode. If the AP is in Open mode, this parameter can be set to '0'.

beacon_interval(2 bytes) :Beacon interval of the AP in milliseconds. Allowed values are integers from 100 to 1000 which are multiples of 100.

dtim_period(2 bytes): DTIM period to be configured in AP mode

ap_keepalive_type:This is the bitmap to enable AP keep alive functionality and to select the keep alive type.

BIT[0]: To enable/disable keep alive functionality.
1 - To enable keep alive functionality.
0 - To disable keep alive functionality.

BIT[1]: To select AP keep alive method.
1 - To enable null data based keep alive functionality.
0 - To enable based keep alive functionality.

ap_keepalive_period:This is the period after which AP will disconnect the station if there are no wireless exchanges from station to AP. Keep alive period is calculated in terms of 32 multiples of beacon interval (i.e. if there are no wireless transfers from station to AP with in (32*beacon_interval*keep_alive_period) milli seconds time period, station will be disconnected).If null data based method is selected, AP checks the connectivity of station by sending null data packet. If station does not ack the packet, that station will be disconnected.

max_sta_support(2 bytes): Number of clients supported. The maximum value allowed is based on the value given in custom feature select bit map[BIT[13] – BIT[16]].max_sta_support should be less than or equal to the value given in custom feature bitmap given in opermode. For example, if this value is 3, not more than 3 clients can associate to the client.

Note:
AP parameters are valid only if opermode is given as 6
Module_mac(6bytes): Mac address to be set for module.

Note:

Host should send mac address with 00:00:00:00:00:00 values to use module's default mac address.

antenna_select (2 bytes): This variable configures the antenna to be used. RS9116-WiSeConnect provides two options – an inbuilt antenna and a uFL connector for putting in an external antenna.
0– Inbuilt antenna selected
1– UFL connector selected

Reserved_3(2 bytes): reserved

Index(2 bytes): In some APs, there is an option to provide four WEP keys.
0-Key 1 will be used.
1-Key 2 will be used.
2-Key 3 will be used.
3-Key 4 will be used.

Key(32 bytes): Actual keys. There are two modes in which a WEP key can be set in an Access Point- WEP (hex) mode and WEP (ASCII) mode. The module supports WEP (hex) mode only.

dhcpv6_enable(2 bytes):DHCPv6 mode.

prefix_length(2 bytes) : prefix length of ipv6 address.

ip6(16 bytes): IPv6 address of module.

Dgw6(16 bytes): IPv6 address of default router.

tcp_stack_used(1 byte): shows which TCP stack is used. Possible values are:
0x01- ipv4 Stack

0x02- ipv6 Stack
0x03- dual Stack, both IPv4 and IPv6 Stack

bgscan_magic_code(2 bytes): This magic code is used to validate the bgscan parameter present in flash memory.

bgscan_enable(2 bytes): To enable/Disable bgscan
0 – Disable
1 - Enable

bgscan_threshold(2 bytes): This is the threshold in dBm to trigger the bgscan. After bgscan periodicity, if connected AP RSSI falls below this then bgscan will be triggered.

rssi_tolerance_threshold(2 bytes): This is difference of last RSSI of connected AP and current RSSI of connected AP. Here last RSSI means RSSI calculated at last time beacon received and current RSSI is RSSI calculated at current beacon received. If this difference is more than rssi_tolerance_threshold and current RSSI is greater than bgscan_threshold then bgscan will be triggered irrespective of periodicity.

bgscan_periodicity(2 bytes): This is time period in seconds to trigger bgscan if RSSI of connected AP is above (assuming RSSI is positive value) than the given bgscan_threshold.

active_scan_duration(2 bytes): This is active scan duration and it is in ms.

passive_scan_duration(2 bytes): This is passive scan duration in ms.

multi_probe(1 byte): If set to one then module will send two probe request one with specific SSID provided during join command and other with NULL ssid (to scan all the access points).

chan_bitmap_magic_code(2 bytes):This variable is used to validate the given scan channel bitmaps. If magic code is 0x4321,then only the scan channel bitmaps are considered as valid.

scan_chan_bitmap_stored_2_4_GHz(4bytes): channel bitmap for scanning in set of selective channels in 2.4 GHz band

scan_chan_bitmap_stored_5_GHz(4bytes): channel bitmap for scanning in set of selective channels in 5 GHz band

roam_magic_code(2 bytes):This magic code is used to validate the roaming parameters stored in the flash memory.

roam_enable(4 bytes): To Enable/Disable roaming.
0 – Disable
1 - Enable

roam_threshold(4 bytes): If connected AP RSSI falls below this then module will search for new AP from background scanned list.

roam_hysteresis(4 bytes): If module found new AP with same configuration (SSID, Security etc.) and if (connected_AP_RSSI – Selected_AP_RSSI) is greater than roam_hysteresis then it will try to roam to the new selected AP.

rejoin_magic_code(2 bytes): This magic code is used to validate the rejoin parameters stored in the flash memory.

rejoin_max_retry(4 bytes): This is 4 byte unsigned integer. This represents the number of attempts for join before giving up the error.

Note:

If number of rejoin attempts is 0 then module will try infinitely for rejoin.

Rsi_scan_interval(4 bytes):This is 4 byte signed integer. This is time interval in second for the subsequent retry.

Rsi_beacon_missed_count(4 bytes):This is 4 byte signed integer. This is the beacon missed count that module used to declare module connection status. If module found continuous beacon missed is greater than or equal to this value, then it will declare connection as disconnected and will start rejoin process again.

Rsi_first_time_retry_enabled (4 bytes):

This is 4 byte unsigned integer. If this is 1 then module will retry to connect for the first time itself for join. Number of attempt and scan interval may be configured by rejoin_max_attempts and scan_interval respectively.

region_request_from_host(1 byte):

If this variable is 1, region of the module is set either in auto join or auto create mode based on the opermode.

1 – Enable set region in Auto create or Auto join mode
0 – Disable set region in Auto create or Auto join mode

rsi_region_code_from_host(1 byte):

Enable/Disable set region code from user.

If opermode is 0 or 2:

1 - Enable - Use the region information from user command
0 – Disable - Use the region information from beacon (country IE)

If opermode is 6:

0- Disable-Get the region information based on region code from internal memory

region_code(1 byte):

If the region code is given as 0(zero), US domain is considered by default and device is configured according to the US domain regulations.

1-US domain
2-Europe domain
3-Japan Domain

Note:

1. All the magic codes should be 0x4321.Firmware validates the respective detailes if and only if the magic code is matching
2. Below given parameters are transparent mode specific.

Trans_mode_enable(2 bytes): If transparent mode magic word (0x5C5C) is given transparent mode is enabled, after succesfull connection/creation of socket.

packet_len(2 bytes): This is the number of bytes (payload) with which network frames are formed and transmitted (bytes/data received within gap timeout) over TCP/IP network.

Escape_char (1, ASCII): Special character provided which will be detected and its 3-character sequence will have special meaning depending of time of arrival.

Gap_time (2 bytes): Maximum time gap between bytes received from host within which escape characters are not expected/checked.

Frame_time (2 bytes): The timeout period for framing network packet from the bytes received. if this timeout is occurred, bytes available in Rx buffers will be forced to form a network frame and queue it for transmission. Ideally Framing period = 2 * Gap Time.

Escape_time (2 bytes): If escape character sequence is received after framing timeout and within escape time, then module breaks out of transparent mode, making GPIO low. Ideally Escape Time = 2 * Framing period.

Ip_version(2 bytes)*:* Signifies which IP version to be used in transparent mode.
4 – IP version 4
6 - IP version 6

nSocketType (2 bytes): This gives the protocol which is to be used in transparent mode (TCP/UDP/LTCP/LUDP).
0 - TCP
2 – LTCP
4 – LUDP

stLocalPort(2 bytes):station Local port number to be used in transparent mode.

Dst_port(2 bytes): Remote port number, to which module need to communicate with in transparent mode.

Ipv4_address/Ipv6_address(16 bytes)*:* IP(v4/v6) of remote server which is to be communicated with from module(in client mode).

Max_count(2 bytes): maximum no of clients allowed in transparent mode is fixed to 1, so default value should be 1.

TOS(4 bytes): Type of service.

ssl_enabled(1 byte): If ssl socket is to be used corresponding parameters are to be provided here.
0 – To open TCP socket.
1 - To open SSL client socket.
5 - To open SSL socket with TLS 1.0 version.
9 - To open SSL socket with TLS 1.2 version.

Ssl_ciphers(1 byte): If ssl socket is to be used corresponding ciphers used is to be provided here.
BIT(1): 2: TLS_RSA_WITH_AES_256_CBC_SHA256
BIT(2): 4: TLS_RSA_WITH_AES_128_CBC_SHA256
BIT(3): 8: TLS_RSA_WITH_AES_256_CBC_SHA
BIT(4): 16: TLS_RSA_WITH_AES_128_CBC_SHA
BIT(5): 32: TLS_RSA_WITH_AES_128_CCM_8
BIT(6): 64: TLS_RSA_WITH_AES_256_CCM_8

multicast_magic_code(2 bytes):
This magic code is used to validate the multicast parameters stored in the flash memory

multicast_bitmap(2 bytes):
There are two bytes in the command which represent 2 parts. Lower order byte represents the command type (cmd as mentioned below) and higher order byte is the hash value (6 Bits) generated from the desired multicast mac address (48 Bits) using hash function.

multicast_bitmap[0:1]: These 2 bits represents the command type. Possible values are:
0 - RSI_ MULTICAST_MAC_ADD_BIT (To set particular bit in multicast bitmap)
1 - RSI_ MULTICAST_MAC_CLEAR_BIT (To reset particular bit in multicast bitmap)
2 - RSI_ MULTICAST_MAC_CLEAR_ALL (To clear all the bits in multicast bitmap)
3 - RSI_ MULTICAST_MAC_SET_ALL(To set all the bits in multicast bitmap)

multicast_bitmap[2:7]: reserved.

multicast_bitmap[8:13]: 6bit hash value generated from the hash algorithm which corresponds to the multicast mac address is used to set/reset corresponding bit in multicast filter bitmap. This field is valid only if 0 or 1 is selected in command type (multicast_bitmap[0:1]).

multicast_bitmap[14:15]: reserved

powermode_magic_code(2 bytes)*:*
This magic code is used to validate the power mode parameters stored in the flash memory

powermode(1 byte):
powermode variable configures the power save mode of the module.
1–Power save Mode 1
2–Power save Mode 2
3–Power save Mode 3

ulp_mode(1 byte)*:*
0 - Low power mode.
1 - Ultra low power mode with RAM retention.
2 - ULtra low power mode without RAM retention.

Refer section **Powersave operation** for detail description about power save operation.
wmm_ps_magic_code(2 bytes)*:*

This magic code is used to validate the WMM power save parameters stored in the flash memory
wmm_ps_enable(1 byte): To enable or disable WMM
0 - Disable
1- Enable

wmm_ps_type(1 byte):WMM PS type
0 - Tx Based
1- Periodic

wakeup_interval(4 bytes)*:* Wakeup interval in milli seconds.

wmm_ps_uapsd_bitmap(1 byte): Bitmap , 0 to 15 possible values.

wmm_ps_uapsd_bitmap[0]:Access category: voice

wmm_ps_uapsd_bitmap[1]: Access category:video

wmm_ps_uapsd_bitmap[2]: Access category:Back ground

wmm_ps_uapsd_bitmap[3]: Access category:Best effort U-APSD

wmm_ps_uapsd_bitmap[4:7]: All set to '0'. Don't care bits.
Listen_interval(4 byte):
This is valid only if BIT(1) in join_feature_bit_map is set. This value is given in time units (1024 milliseconds). This parameter is used to configure maximum sleep duration in power save.

Listen_interval_dtim(1 byte):

This parameter is valid only if BIT(1) is set in the join_feature_bitmap and valid listen interval is given in join command. If this parameter is set, the module computes the desired sleep duration based on listen interval (from join command) and its wakeup align with Beacon or DTIM Beacon (based on this parameter).

0 - module wakes up before nearest Beacon that does not exceed the specified listen interval time.

1 - module wakes up before nearest DTIM Beacon that does not exceed the specified listen interval time.

private_key_password[82](1 byte): Private Key Password is required for

encrypted private key, format is like "\"12345678\"".

join_bssid: This contains BSSID of selected AP.

Note:

1. All the magic codes should be 0x4321. Firmware validates the respective details if and only if the magic code is matching
2. Below given parameters are transparent mode specific.

Fast_psp_enable(1 byte) :
When fast psp is enabled, module will disable power save for monitor interval of time for each data packet received or sent.

Monitor_interval (2 bytes):
This is time in ms to keep module in wakeup state for each Tx or Rx traffic sent or received respectively. Default value for this is 50 ms.

Timeout_value (2 bytes):
timeout value in ms(default 300ms).

timeout_bitmap (4 bytes):

BIT[0]:
sets timeout for association and authentication request.

Request_timeout_magic_word(2 bytes):
Magic word of request time out.

ht_caps_magic_word (2 bytes):
Magic word od HT caps.

dhcp_ap_enable (1 byte):
DHCPv4 mode enable or disable

ap_ip(4 bytes):
Module IP address

ap_sn_mask(4 bytes):
Sub-net mask

ap_dgw(4 bytes):
Default gateway

dhcpv6_ap_enable(2 bytes):
DHCPv6 mode enable or disable

ap_prefix_length(2 bytes):
prefix length of ipv6 address.

ap_ip6(16 bytes):
IPv6 address of module.

ap_dgw6(16 bytes):
IPv6 address of default router.

ext_tcp_ip_feature_bit_map:

BIT[1] - To use DHCP User class Option

1 – To enable DHCP user class option
0 - To disable DHCP user class option

BIT[2] – To bypass the HTTP servers default root configuration page

1 – To enable HTTP server root path bypass option
0 - To disable HTTP server root path bypass option

Note:

Enable BIT[3] in **ext_tcp_ip_feature_bit_map to enable this feature**

BIT[3] – Correcting the ACK sequence number in the TCP packet retransmission path

1 – To enable
0 - To disable

BIT[4] – To enable TCP ACK division factor feature

1 – To enable
0 - To disable

BIT[5] – To enable SSL server certificate validation by host. (By enabling this bit server certificate is sent to host and host validates the server certificate and send the valid response to the module)

1 – To enable
0 - To disable

BIT[6] - Support for SSL 16K record size

1 – To enable

0 - To disable

BIT[7] - To enable DNS_CLIENT_BYPASS by host

1 – To enable

0 - To disable

BIT[8] - To enable TCP window scaling feature. If user wants to use more than 64Kb window size. **tcp_rx_window_size_cap** in socket config command is used to increase the window size.

1 – To enable

0 - To disable

Note:

The above AP IP parameters are valid in concurrent mode.

http_credentials_avail (1 byte): Reserved.
http_username [MAX_HTTP_SERVER_USERNAME](1 byte): HTTP server username.
http_password [MAX_HTTP_SERVER_PASSWORD](1 byte): HTTP server password.

5.90 WLAN Statistics

Note:
This command is supported only in WI-FI alone mode.

Description:
This command is used to query for WLAN statistics of the module. If this command is issued immediately after opermode command, all the stats will be empty. It should be issued after WLAN connection.

Command Format:

at+rsi_get_wlan_stats?\r\n

Command Parameters:

Operatingmode: Module is configured in client mode.

Dtim_period: *DTIM* stands for Delivery traffic indication map or message. The *DTIM interval* (1-255) means the period of time to wake up wireless clients from Sleep Mode.

Ideal Beacon info: The number of beacons without multicast and broadcast indication in TIM field.

Busy Beacon info: The number of beacons with multicast and broadcast indication in TIM field.

Beacon interval: Beacon Broadcast interval is the time lag between each of the beacons sent by your router or access points.

Response

Result	Description
Ok<Wi-fi stats>	Successful execution of command.
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0021, 0xFF82.

Relevance:
This command is relevant when the module is configured in Operating Mode 0, 2.

Example:
AT Mode:

```
at+rsi_wlan_stats?\r\n
```

Response:

```
OK\0x00\0x01\0x0b\0x00\0x0c\0x00\0xc8\0x00
4f 4b 00 01 0b 00 0c 00 c8 00 0d 0a
```

5.91 Set RTC Time

Description:
This command is used to set/initialize the real time clock of the module from the host.

Note:
To enable this feature, host needs to set BIT[28] of custom feature bitmap through opermode command.

Command Format:
at+rsi_host_rtc_time=<second>,<minute>,<hour>,<day>,<month>,<year><weekday>\r\n

Command Parameters:

second: This is current real time clock seconds, which needs to set for module.
minute: This is current real time clock minute, which needs to set for module.
hour: This is current real time clock hour, which needs to set for module.
day: This is current real time clock day, which needs to set for module.
month: This is current real time clock month, which needs to set for module.
year: This is current real time clock year, which needs to set for module.

weekday: This is current real time clock weekday, which needs to set for module.

Note:
hour is 24-hour format only (valid values are 0 to 23)
Valid values for Month are 0 to 11 (January to December)

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0021, 0x0025

Relevance:
This command is relevant in all modes.

Example:

AT Mode:
Below example configure the module rtc time to APRIL 2 10:10:10 2018 6
at+rsi_host_rtc_time=10,10,10,2,3,2018,6\r\n

Response:

OK\r\n
0x4F 0x4B 0x0D 0x0A

5.92 Get RTC Time

Description:
This command is used to get the real time clock of the module from the host.

Command Format:
at+rsi_get_rtc_time?

Response:
AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:

Possible error codes are 0x0021, 0x0025

Relevance:
This command is relevant in all modes.

Example:

AT Mode:
Below example get the configured module rtc time.
at+rsi_get_rtc_time?

Response:

OK< tm_sec>< tm_min>< tm_hour>< tm_mday >< tm_mon >< tm_year ><tm_wday>\r\n

Response parameters:

second: This is current real time clock seconds.
minute: This is current real time clock minute.
hour: This is current real time clock hour.
day: This is current real time clock day.
month: This is current real time clock month.
year: This is current real time clock year.

weekday: This is current real time clock weekday, which needs to be set for module.

5.93 Feature Frame

Description:
This command is used to select internal RF type or External RF type and clock frequency.

Command Format:
at+rsi_feat_frame=<pll_mode>, <rf_type> ,<wireless_mode> ,<enable_ppp>, <afe_type>,<features_enable>\r\n

Command Parameters:

PLL_MODE:
0- PLLMODE0 - Used for generating the clocks for 20Mhz Bandwidth operations.
1- PLLMODE1 - Used for generating the clocks for 40Mhz Bandwidth operations.
2- PLLMODE2-Reserved

RF_TYPE[ONLY FOR 2GHz]:
0- To enable External_RF_8111,
1- To enable Internal_RF_9116,
2- AVIACOM_RF

WIRELESS_MODE:
12- To enable LP chain for PER mode
0- LP chain disable

ENABLE_PPP:
0- Disable_per_packet_TX_programming,
1- Enable_per_packet_TX_programming_mode_1,
2- Enable_per_packet_TX_programming_mode_2

AFE:
0- AFE BYPASS,
1- Internal AFE

FEATURE_ENABLES:

BIT[0] - To enable Preamble duty cycling.
BIT[4] - To enable LP chain for stand-by associate mode.
BIT[5] - To enable hardware beacon drop during power save.

Note:
Remaining bits are not user configurable.

Response:

AT Mode:

Result Code Description
OK Successful execution of the command
ERROR<Error code> Failure

Possible error codes:
Possible error codes are 0x0021, 0xFF74

Relevance:
This command is relevant in all modes.

Example:

AT Mode:
at+rsi_feat_frame=0,0,0,0,1\r\n

Response:

OK\r\n
0x4F 0x4B 0x0D 0x0A

5.94 Get RAM Dump

Description:
This command is used to get ram dump of the module for a given address and offset.

Command Format:
at+rsi_get_ram_dump=<address><length>/r/n

ADDR (4 bytes):

Address in RS9116 module

LENGTH (4 bytes):

Chunk length to read from RS9116 module

Response:

AT Mode:

Result Code Description
OK Successful execution of the command
ERROR<Error code> Failure

Possible error codes:
Possible error codes are 0x0021, 0x003e

Relevance:
This command is relevant in all modes.

Example:

AT Mode:
at+rsi_get_ram_dump=0,4096\r\n

Response:

OK\r\n
0x4F 0x4B 0x0D 0x0A

Note:
This command should be sent only after opermode command.

BROADCAST FILTER

Description:
This command is used to program the ignoring broadcast packet threshold levels when station is in powersave mode and is used to achieve low currents in standby associated mode.

Command Format:
at+rsi_filter_bcast=<beacon_drop_threshold><filter_bcast_in_tim><filter_bcast_tim_till_next_cmd>/r/n

Command Parameters:

BEACON DROP THRESHOLD (2 bytes)
LMAC beacon drop threshold(ms): The amount of time that FW waits to receive full beacon.
Default value is 5000ms.

FILTER BROADCAST IN TIM (1 byte)
If this bit is set, then from the next dtim any broadcast data pending bit in TIM indicated will be ignored
valid values: 0 - 1

Note:
Validity of this bit is dependent on the filter_bcast_tim_till_next_cm.

FILTER BROADCAST TIM TILL NEXT COMMAND (1 byte)

0 -filter_bcast_in_tim is valid till disconnect of the STA
1 - filter_bcast_in_tim is valid till next update by giving the same command

Response:

AT Mode:

Result Code Description
OK Successful execution of the command
ERROR<Error code> Failure

Possible error codes:
Possible error codes are 0x0021,

Relevance:
This command is relevant in all modes.

Example:

AT Mode:
at+rsi_filter_bcast=2,1,1\r\n

Response:
OK\r\n
0x4F 0x4B 0x0D 0x0A

5.95 Configure TX RX Buffer Ratio

Description:
This command is used to configure the tx, rx and global buffer ratio.

Command Format:
at+rsi_buf_alloc = <dynamic_tx_pool><>dynamic_rx_pool<dynamic_global_pool>/r/n

Command Parameters:

DYNAMIC TX POOL
To configure the tx pool ratio.

DYNAMIC RX POOL
To configure the rx pool ratio.

DYNAMIC GLOBAL POOL
To configure the global pool ratio.

Note:
Summation of above three ratios should be max 10 and ratio should be in decimal value.

Response:

AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:
Possible error codes are 0x0021,

Relevance:
This command is relevant in all modes.

Example:

AT Mode:
at+rsi_buff_alloc=1,1,1\r\n

Response:

OK\r\n
0x4F 0x4B 0x0D 0x0A

5.96 Antenna Selection

Description:
This command is used to configure the antenna.

Command Format:
at+rsi_antenna=<AntennaVal><Gain_2G>,<Gain_5G>,<antenna_path>,<antenna_type>\r\n

Command Parameters:

AntennaVal:
0 – RF_OUT_2/Internal Antenna is selected
1 – RF_OUT_1/uFL connector is selected.

For RS9116W single band modules BOO and Q7, 'AntennaVal' should be 0.

For RS9116W dual band modules like CC0, appropriate 'AntennaVal' should be configured based on Wi-Fi band being used. If using 2.4GHz band, 'AntennaVal' should be 0 and if using 5GHz band, 'AntennaVal' should be 1. In the later case, antenna should be mounted on uFL connector.

When using RS9116W modules/EVKs for PER testiing, 'AntennaVal' should be set to 0 in order to use uFL connector.

Note:
rsi_antenna command must be given after rsi_init command.

Note:
Currently Gain_2g, Gain_5g, antenna_path and antenna_type parameters are supported.

Response:

AT Mode:

Result Code	Description
OK	Successful execution of the command
ERROR<Error code>	Failure

Possible error codes:
Possible error codes are 0x0021, 0x002C

Relevance:
This command is relevant in all modes.

Example:

AT Mode:
at+rsi_antenna=1 \r\n

Response:

OK\r\n
0x4F 0x4B 0x0D 0x0A

5.97 MQTT Client

Description:
This section explains different commands to use MQTT client.
This command should be given only after Set IP Parameters command.

Note:
- To Support this feature, enable ext_tcp_ip_feature_bit_map[17] in the opermode command.
- ext_tcp_ip_feature_bit_map gets enabled when BIT(31) is set to '1' in tcp_ip_feature_bit_map.
- After MQTT connection, only 9 sockets can be opened.

Note:
MQTT only supports STA mode in current release.

Init/Create	This command is used to Create MQTT objects. This should be the first command for accessing MQTT. TCP level connection is established in this command. This command has to be issued after "IPCONFIG" command.
Connect	This command is used to Connect to MQTT Server/Broker(mosquitto). MQTT level connection is established in this command. This command has to be issued after "MQTT INIT" command.
Subscribe	This command is used to send MQTT subscribe packet. This command has to be issued after "MQTT CONNECT" command.
Publish	This command is used to send MQTT publish packet. This command has to be issued after "MQTT CONNECT" command.
Unsubscribe	This Command is used to send MQTT unsubscribe packet. This command has to be issued after "MQTT CONNECT" command.
Disconnect	This command is used to Disconnect from the Socket. MQTT and TCP level disconnection is established in this command. This command has to be issued after "MQTT CONNECT" command.
Delete/Destroy	This command is used to delete MQTT client's configuration and TCP level disconnection happens in this command. This command has to be issued after "MQTT INIT" command.

Following table explains list of MQTT commands and their description.

- **Init/Create** should be called as a first command to use MQTT.
- Once create is successful, **connect** should be called to connect to a MQTT Server/Broker.
- After connection is successful, host can issue remaining commands.
- After MQTT operations are done, host has to give **disconnect** command to disconnect from MQTT server/Broker.
- Once disconnect is done, host can again connect to the MQTT server using **connect** command.
- Delete commad is used to delete MQTT client's configuration.

Command Format:
AT Mode:
MQTT client has different command types. Based on the command type next parameters will change.
at+rsi_mqtt=<command_type>,<remaining parameters >\r\n
Following are available command types.

MQTT Command	Command Type	Command Format
Init	**1**	at+rsi_mqtt=1,<ip_version>,<server_ip>,<server_port>,<client_id_length>,<client_id>,<keepalive>,<username_length>,<username>,<password_length>,<password>,<clean_session>,<encrypt>,<client_port>\r\n • <ip_version> ip version used (IPv4-4, IPv6-6) **Note:** IPv6 is not supported. • <server_port> MQTT server port number • <server_ip> MQTT Server/Broker IP address • <client_id_length> length of the client id (Not valid if 0 or more than 60) **Note:** Client id length and length of the client id should be a match. Undefined behavior may be observed if these fields are mismatched. • <client_id> client ID **Note:** ClientID should be unique and should not match with others • <keepalive> keep alive interval (in seconds) **Note:** keepalive interval is not a constant value. It varies. The default keep alive interval is 40 seconds. • <username_length > length of the username • <username> username to be used to authenticate with MQTT broker **Note:** • Maximum supported length for username is 60 bytes. • Username length and length of the Username should be a match. Undefined behavior may be observed if these fields are mismatched. • <password_length > length of the password • <password> username to be used to authenticate with MQTT broker **Note:**

MQTT Command	Command Type	Command Format
		• Maximum supported length for Password is 60 bytes. • Password length and length of the Username should be a match. Undefined behavior may be observed if these fields are mismatched. • <clean_session> clean session(0-1) (Clears historical data if set). Clean session 0 is not supported. It should always be 1. • <encrypt> Type of MQTT socket connection. 0 : Disable SSL, 1 : Enable SSL • <client_port> Client port to be used. If this parameter is not given,1883 is used as client port number
Connect	**2**	at+rsi_mqtt=2,<usr_flag>,<pwd_flag>,<will_flag>,<will_retain>,<will_qos>,<will_topic_length>,<will_topic>,<will_msg_length>,<will_msg>\r\n • <usr_flag> Enable username (0-1) to authenticate with MQTT server. 1- Enable username, 0 - Disable username • <pwd_flag> Enable password (0-1) to authenticate with MQTT server. 1 - Enable password, 0 - Disable password • <will_flag> Reserved • <wIll_retain> Reserved • <will_qos> Reserved • <will_topic_length> Reserved • <will_topic> Reserved • <will_msg_length> Reserved • <will_msg> Reserved
Subscribe	**3**	at+rsi_mqtt=3,<topic_length>,<topic>,<qos>\r\n <topic_length> topic length **Note:** • Maximum supported length for TOPIC is 60 bytes. • TOPIC length and length of the TOPIC should be a match. Undefined behavior may be observed if these fields are mismatched. • <topic> topic to subscribe • <qos> message Qos, can be 0, 1, or 2 . **Note:** QOS2 is not supported
Publish	**4**	at+rsi_mqtt=4,<topic_length>,<topic>,<qos>,<retained>,<dup>,<message_length>,<message>\r\n

MQTT Command	Command Type	Command Format
		• <topic_length> topic length **Note:** • Maximum supported length for TOPIC is 60 bytes. • TOPIC length and length of the TOPIC should be match. Undefined behavior may be observed if these fields are mismatched. • <topic> topic of subscribe message. • <qos> Publish message Qos, can be 0, 1, or 2. **Note:** QOS2 is not supported. • <retained> retained flag, can be 0 or 1. **Note:** If the RETAIN flag is set to 1 in a PUBLISH Packet sent by a Client to a Server, the Server must store the Application Message and its QoS, so that it can be delivered to future subscribers whose subscriptions match its topic name. • <dup> duplicate flag, can be 0 or 1. **Note:** The DUP flag MUST be set to 1 by the Client or Server when it attempts to re-deliver a PUBLISH Packet. The DUP flag MUST be set to 0 for all QoS 0 messages. • <message_len> length of publish message. **Note:** a) The message length is dependent on TOPIC length, total length should not exceed 1460 bytes, if connection is not SSL secured (MQTT Header+Topic + Publish data). b) SSL Maximum supported message_len should not exceed 1370 bytes (MQTT Header+Publish data). • <message> Publish message
Unsubscribe	5	at+rsi_mqtt=5,<topic_length>,<topic>>\r\n • <topic_len> topic length

MQTT Command	Command Type	Command Format
		Note: • Maximum supported length for TOPIC is 60 bytes. • TOPIC length and length of the TOPIC should be match.Behavior may be undefined if these fields are mismatched • <topic> topic of unsubscribe message
Disconnect	**8**	at+rsi_mqtt=8\r\n
Delete/Destroy	**9**	at+rsi_mqtt=9\r\n

Response:
AT Mode:

MQTT Command	Command Response
Create/Init	OK< ip_version =0x04 0x00>< socketType =0x0000 >< socketDescriptor =0x0001>< moduleSocket =0x4d2>< ipv4_addr= 0xC0 0xA8 0x28 0x120x00(12 times)> < mss =0xB4 0x05>< window_size =0x00 0x00 0x01 0x0>\r\n (or)Error Code (failure case).
Connect	OK/Error Code **Value** | **Return Code Response** | **Description** 0 | 0x00 Connection Accepted | Connection accepted 1 | 0x01 Connection Refused, unacceptable protocol version | The Server does not support the level of the MQTT protocol requested by the Client 2 | 0x02 Connection Refused, identifier rejected | The Client identifier is correct UTF-8 but not allowed by the Server 3 | 0x03 Connection Refused, Server unavailable | The Network Connection has been made but the MQTT service is unavailable

MQTT Command	Command Response
	<table><tr><td>4</td><td>0x04 Connection Refused, bad username or password</td><td>The data in the username or password is malformed</td></tr><tr><td>5</td><td>0x05 Connection Refused, not authorized</td><td>The Client is not authorized to connect</td></tr></table>
Subscribe	OK/Error Code
Publish	OK/Error Code
Unsubscribe	OK/Error Code
Destroy/Delete	OK/Error Code ERROR<ERROR-CODE>

Note:

DUT sends received data on published topic to host asynchronously.

Format:

AT+RSI_MQTT_READ_DATA<mqtt_flags><current_chunk_length><topic_length><topic><message>\r\n

Asynchronous Responses:

AT Mode:

Response	Description	Response format	Parameters Description
Remote Terminate	This message is posted to the host when remote terminate /socket closure from peer is received	ERROR<ERROR-CODE>\r\n	
Keep Alive Time out	This message is posted to host when Keepalive(MQTT Ping) response is not received in given time period when keep alive request is sent from module	AT+RSI_MQTT_KA_TIMEOUT\r\n	
Publi sh Message	This message is received when publish message is received from MQTT Broker	AT+RSI_MQTT_READ_DATA<mqtt_flags><current_chunk_length><topic_length><topic><message>\r\n	mqtt_flags : 2 bytes : Has the info regarding MQTT publish message. BIT(0) - Retain flag BIT(1) & BIT(2) - QOS level BIT(3) - DUP flag BIT(4) - More data BIT(5) : BIT(15) - Reserved for future use. This bit is set if more chunks/ Data is coming for same publish. message topic_length: 2 bytes - length of the topic current_chunk_l ength: 2 bytes - message length in current chunk topic: publish topic-

			This field is valid only if topic length is non-zero value message: publish message : This field is valid only if current_chunk_length field is non-zero **Note:** <topic> and <message> are variable fields. These fields are dependent on <topic_length> and <current_chunk _length>. Rest of the parameters are fixed.

Example:

AT Mode:

```
at+rsi_opermode=0,0,2147484676,2147483648,3145728,0,131072,0,0,0
at+rsi_band=0
at+rsi_init
at+rsi_scan=0,ap_ssid_OP_TESTING
at+rsi_psk=1,12345678
at+rsi_join=ap_ssid_OP_TESTING,0,2,2
at+rsi_ipconf=1

//MQTT init and connect
at+rsi_mqtt=1,4,139.196.135.135,1883,38,6789|securemode=3,signmethod=hmacsha1|,200,19,Device1&a1BLKg93
h0r,40,EB9558F74EF496E72C129B5EDF66427683488246,1,0
at+rsi_mqtt=2,1,1,0,0,0,0,0,0
//Subscribe
at+rsi_mqtt=3,7,redpine,1,1
//Publish to cloud
at+rsi_mqtt=4,7,redpine,0,0,0,40,{\"state\":{\"desired\":{\"toggle\":1}}}
 //Unsubscribe
at+rsi_mqtt=5,7,redpine
//Disconnect
at+rsi_mqtt=8
//Destroy
at+rsi_mqtt=9
```

Note:
For SSL connectivity, user needs to configure SSL certificate prior to the above sequence of steps.

Reference Links:

http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html
https://www.oasis-open.org/news/announcements/mqtt-version-3-1-1-becomes-an-oasis-standard

6 WLAN Error Codes

Error Codes

This section lists and describes error codes for different commands.

Table 16 Error Codes

Error Codes (in hexadecimal format)	Description
0x0002	Scan command issued while the module is already associated with an Access Point
0x0003	No AP found
0x0004	Wrong PSK is issued while the module client tries to join an Access Point with WEP security enabled
0x0005	Invalid band
0x0006	Association not done or in an dissociated state
0x0008	De-authentication received from AP
0x0009	Failed to associate to Access Point during "Join"
0x000A	Invalid channel
0x000E	1. Authentication failure during "Join" 2. Unable to find AP during join which was found during the scan.
0x000F	Missed beacon from AP during the join
0x0013	Non-existent MAC address supplied in "Disassociate" command
0x0014	EAP configuration is not done
0x0015	Memory allocation failed or Store configuration checksum failed
0x0016	Information is wrong or insufficient in Join command
0x0018	Push button command given before the expiry of the previous push button command
0x0019	1. Access Point not found 2. Rejoin failure
0x001A	Frequency not supported
0x001B	Invalid Opermode
0x001C	EAP configuration failed
0x001F	Disconnect in wrong state
0x0020	Unable to join
0x0021	Command given in the incorrect state
0x0023	Unable to form Access Point
0x0024	Wrong Scan input parameters supplied to "Scan" command
0x0025	Command issued during re-join in progress
0x0026	Wrong parameters the command request

Error Codes (in hexadecimal format)	Description
0x0028	PSK length less than 8 bytes or more than 63 bytes
0x0029	Failed to clear or to set the Enterprise Certificate (Set Certificate)
0x002B	Association between nodes failed in WPS Failed due to timeout
0x002C	If a command is issued by the Host when the module is internally executing auto-join or auto-create
0x002D	WEP key is of the wrong length
0x002E	ICMP request timeout error
0x002F	ICMP data size exceeds the maximum limit
0x0030	Send data packet exceeded the limit or length that is mentioned (or) Mqtt publish data and publish data length mismatched (or) MQTT Send data packet exceeded the limit.
0x0031	ARP Cache entry not found
0x0032	UART command timeout happened
0x0033	Fixed data rate is not supported by connecting AP
0x0036	Maximum length exceeded of Username/password/Client_ID/Topic in MQTT.
0x0037	Wrong WPS PIN
0x0038	Wrong WPS PIN length
0x0039	Wrong PMK length
0x003a	SSID not present for PMK generation
0x003b	SSID incorrect for PMK generation(more than 32 bytes)
0x003C	Band not supported
0x003D	User store configuration invalid length
0x003E	Error in length of the command(Exceeds number of characters is mentioned in the PRM)
0x003F	Data packet dropped
0x0040	WEP key not given
0x0041	Wrong PSK length
0x0042	PSK or PMK not given
0x0043	Security mode given in join command is invalid
0x0044	Beacon miscount reaches max beacon miss count (De-authentication due to beacon miss)
0x0045	De-authentication received from the supplicant
0x0046	De-authentication received from AP after channel switching
0x0047	Synchronization missed
0x0048	Authentication timeout occurred
0x0049	Association timeout

Error Codes (in hexadecimal format)	Description
0x004A	BG scan in given channels is not allowed
0x004B	Scanned SSID and SSID given in Join are not matching
0x004C	Given number of clients exceeded max number of stations supported
0x004D	Given HT capabilities are not supported
0x004E	UART Flow control not supported
0x004F	ZB/BT/BLE packet received and protocol is not enabled
0x0050	Parameters error
0x0051	Invalid RF current mode
0x0052	Power save support is not present for a given interface
0x0053	Concurrent AP in connected state
0x0054	Connected AP or Station channel mismatch
0x0055	IAP coprocessor error
0x0056	WPS is not supported in current operating mode
0x0057	Concurrent AP doesn't have same channel as connected station channel
0x0058	PBC session overlap error
0x0059	BT feature bit map invalid
0x005A	4/4 confirmation of 4-way handshake failed
0X005C	Concurrent mode, both AP and Client should UP, to enable configuration
0x005D	Certificate load not allowed in flash
0x005E	Certificate load not allowed in RAM
0x005F	Certificate load failed due to wrong inx
0x0060	AP HT caps not enabled for 40 MHz
0x0061	Address family not supported by protocol.
0x0062	Invalid beacon interval provided.
0x0063	Invalid range of the configuration provided
0x0064	RTS THRESHOLD Config type is invalid.
0x0065	Error with MQTT command
0x0066	listen interval in power save is greater than the join listen interval
0x005B	MAC address does not present in MAC based join
0x00B1	Memory Error: No memory available
0x00B2	Invalid characters in JSON object
0x00B3	Update Commands: No such key found
0x00B4	No such file found: Re-check filename
0x00B5	No corresponding webpage exists with same filename

Error Codes (in hexadecimal format)	Description
0x00B6	Space unavailable for new file
0x00C1	Invalid input data, re-check filename, lengths etc.
0x00C2	Space unavailable for new file
0x00C3	Existing file overwrite: Exceeds size of previous file. Use erase and try again
0x00C4	No such file found. Re-check filename
0x00C5	Memory Error: No memory available
0x00C6	Received more web-page data than the total length initially specified
0x00C7	Error in set region command
0x00C8	Web-page current chunk length is incorrect
0x00CA	Error in AP set region command
0X00CB	Error in AP set region command parameters
0x00CC	Region code not supported
0x00CD	Error in extracting country region from beacon
0x00CE	Module does not have selected region support
0x00D1	SSL Context Create failed
0x00D2	SSL Handshake failed. Socket will be closed
0x00D3	SSL Max sockets reached. Or FTP client is not connected
0x00D4	Cipher set failure
0x00F1	HTTP credentials maximum length exceeded
0x0100	SNMP internal error
0x0104	SNMP invalid IP protocol error
0xBB01	No data received or receive timeout
0xBB0A	Invalid SNTP server address
0xBB0B	SNTP client not started
0xBB10	SNTP server not available, Client will not get any time update service from current server
0xBB15	SNTP server authentication failed
0xBB0E	Internal error
0xBB16	Entry not found for multicast IP address
0xBB17	No more entries found for multicast
0xBB21	IP address error
0xBB22	Socket already bound
0xBB23	Port not available
0xBB27	Socket is not created

Error Codes (in hexadecimal format)	**Description**
0xBB29	ICMP request failed
0xBB33	Maximum listen sockets reached
0xBB34	DHCP duplicate listen
0xBB35	Port Not in close state
0xBB36	Socket is closed or in process of closing
0xBB37	Process in progress
0xBB38	Trying to connect non-existing TCP server socket
0xBB3E	Error in length of the command (Exceeds number of characters is mentioned in the PRM)
0xBB40	Wrong packet info
0xBB41	Invalid Length
0xBB42	Socket is still bound
0xBB45	No free port
0xBB46	Invalid port
0xBB4B	Feature not supported
0xBB50	Socket is not in the connected state. Disconnected from server. In the case of FTP, the user needs to give destroy command after receiving this error
0xBB87	POP3 session creation failed/ POP3 session got terminated
0xBB9C	DHCPv6 Handshake failure
0xBB9D	DHCP invalid IP response
0xBBA0	SMTP Authentication error
0xBBA1	No DNS server was specified, SMTP oversize mail data
0xBBA2	SMTP invalid server reply
0xBBA3	DNS query failed, SMTP internal error
0xBBA4	Bad DNS address, SMTP server error code received
0xBBA5	SMTP invalid parameters
0xBBA6	SMTP packet allocation failed
0xBBA7	SMTP GREET reply failed
0xBBA8	Parameter error, SMTP Hello reply error
0xBBA9	SMTP mail reply error
0xBBAA	SMTP RCPT reply error
0xBBAB	SMTP message reply error
0xBBAC	SMTP data reply error
0xBBAD	SMTP authentication reply error

Error Codes (in hexadecimal format)	Description
0xBBAE	SMTP server error reply
0xBBAF	DNS duplicate entry.
0xBBB1	SMTP oversize server reply
0xBBB2	SMTP client not initialized
0xBBB3	DNS IPv6 not supported
0xBBC5	Invalid mail index for POP3 mail retrieve command
0xBBD2	SSL handshake failed
0xBBD3	FTP client is not connected or disconnected with the FTP server
0xBBD4	FTP client is not disconnected
0xBBD5	FTP file is not opened
0xBBD6	SSL handshake timeout or FTP file is not closed
0xBBD9	Expected [1XX response from FTP server but not received
0xBBDA	Expected [2XX response from FTP server but not received
0xBBDB	Expected [22X response from FTP server but not received
0xBBDC	Expected [23X response from FTP server but not received
0xBBDD	Expected [3XX response from FTP server but not received
0xBBDE	Expected [33X response from FTP server but not received
0xBBE1	HTTP Timeout
0xBBE2	HTTP Failed
0xBBE7	HTTP Timeout for HTTP PUT client
0xBBEB	Authentication Error
0xBBED	Invalid packet length, content length and received data length is mismatching
0xBBEF	The server responds before HTTP client request is complete
0xBBF0	HTTP/HTTPS password is too long
0xBBF1	MQTT ping time out error
0xBBF2	MQTT command sent in incorrect state
0XBBF3	MQTT ACK time out error
0xBBFF	POP3 error for invalid mail index
0XFFFF	Listening TCP socket in the module is not connected to the remote peer or the LTCP the socket is not yet opened in the module
0xFFFE	Sockets not available. The error comes if the Host tries to open more than 10 sockets
0xFFFD	HTTP OTAF invalid packet error

Error Codes (in hexadecimal format)	**Description**
0xFFFC	IP configuration failed
0xFFFB	Cannot create IP in the same interface in concurrent mode
0XFFFA	TCP socket is not connected
0xFFF4	HTTP OTAF incomplete packet
0xFFF8	1. Invalid command (e.g. parameters insufficient or invalid in the command). Invalid operation (e.g. power save command with the same mode given twice, accessing the wrong socket, creating more than allowed sockets)
0xFFF6	MQTT REMOTE TERMINATE ERROR
0xFFF7	Byte stuffing error in AT mode
0xFFF5	Store configuration profile type mismatch or Invalid profile type
0xFFF9	HTTP OTAF no packet error
0xFFC5	Station count exceeded max station supported
0xFFC4	Unable to send TCP data
0xFFBC	Socket buffer too small
0xFFBB	Invalid content in the DNS response to the DNS Resolution query
0xFFBA	DNS Class error in the response to the DNS Resolution query
0xFFB8	DNS count error in the response to the DNS Resolution query
0xFFB7	DNS Return Code error in the response to the DNS Resolution query
0xFFB6	DNS Opcode error in the response to the DNS Resolution query
0xFFB5	DNS ID mismatch between DNS Resolution request and response
0xFFAB	An invalid input to the DNS Resolution query
0xFF42	DNS response was timed out
0xFFA1	ARP request failure
0xFF9D	DHCP lease time expired
0xFF9C	DHCP handshake failure
0xFF88	This error is issued when WebSocket creation failed
0xFF87	This error is issued when the module tried to connect to a non-existent TCP server the socket on the remote side
0xFF86	This error is issued when tried to close the non-existent socket. or invalid socket descriptor
0xFF85	Invalid socket parameters
0xFF82	Feature not supported

Error Codes (in hexadecimal format)	**Description**
0xFF81	Socket already open
0xFF80	Attempt to open more than the maximum allowed number of sockets
0XFF7E	Data length exceeds MSS
0xFF74	Feature not enabled
0xFF73	DHCP server not set in AP mode
0xFF71	Error in AP set region command parameters
0xFF70	SSL not supported
0xFF6F	JSON not supported
0xFF6E	Invalid operating mode
0xFF6D	Invalid socket configuration parameters
0xFF6C	Web socket creation timeout
0xFF6B	Parameter maximum allowed value is exceeded
0xFF6A	Socket read timeout
0xFF69	An invalid command in the sequence
0xFF41	HTTP socket creation failed
0xFF40	TCP socket close command is issued before getting the response of the previous close command
0xFF36	Wait On Host feature not enabled
0xFF35	Store configuration checksum validation failed
0xFF33	TCP keep alive timed out
0xFF2D	TCP ACK failed for TCP SYN-ACK
0xFF2C	Memory limit exceeded in a given operating mode
0xFF2A	Memory limit exceeded in operating mode during auto join/create
0xCC2F	PUF Operation is blocked
0xCC31	PUF Activation code invalid
0xCC32	PUF input parameters invalid
0xCC33	PUF in error state
0XCC34	PUF Operation not allowed
0XCC35	PUF operation Failed
0x5a5a	Auto join or user store configuration going on.
0xFFE1	Improper RSNIE from AP to station

7 Wireless Features and Mechanisms

This document briefs about the following wireless features and mechanisms.

- Connect Application
- Firmware Upgrade Mechanism
- Wake on Wireless

7.1 Connect Application

7.1.1 Description

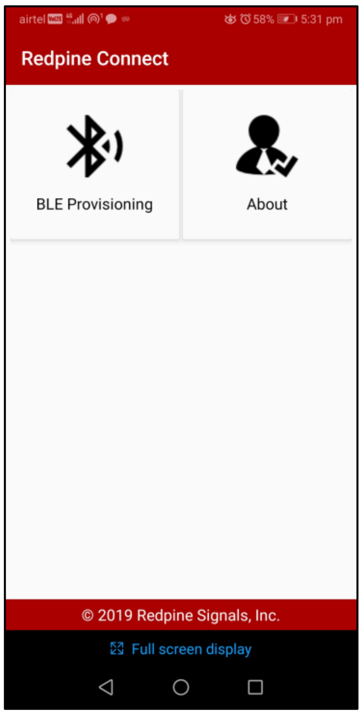
- The BLE Wi-Fi Provisioning example is designed to showcase the provisioning of Wi-Fi parameters (SSID, network key, etc.) over a BLE connection.
- Wi-Fi provisioning over BLE provides various benefits such as eliminating the need to explicitly connect to a shared network.
- Connect Application explains how to establish WLAN Connection using BLE provisioning

In this application,

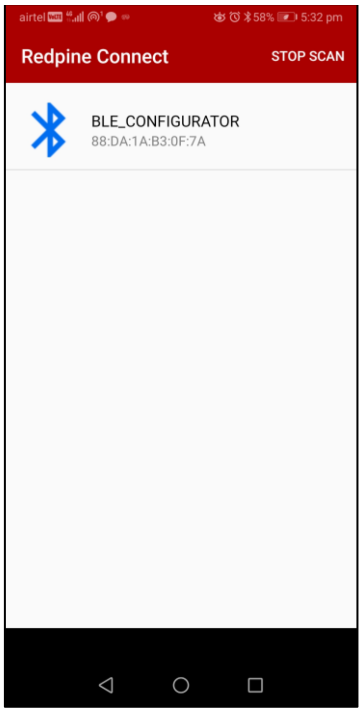
- Silicon Labs Module starts advertising and with BLE Provisioning the Access Point details are fetched
- Silicon Labs device is configured as a Wi-Fi station and connects to an Access Point.

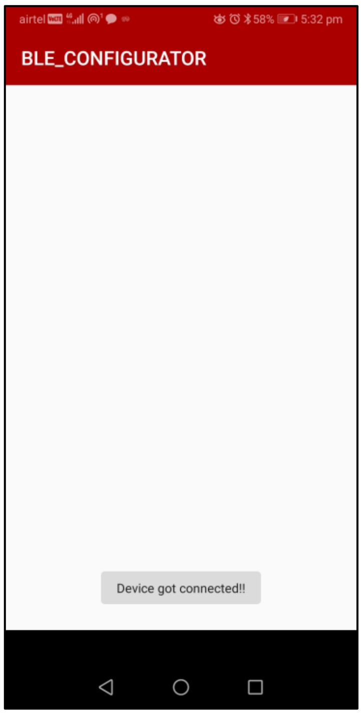
7.1.2 Application Steps

- Configure the Access point with Internet connection in OPEN/WPA-PSK/WPA2-PSK mode to connect the Silicon Labs device in STA mode.
- Connect any serial console for prints.
- Connect app is available in the release 'utils' folder. Currently, this app will be available for Android devices only. path for the application: RS9116.NB0.WC.GENR.OSI.x.x.x/utils
- Launch the Connect App.
- Click on BLE Provisioning.

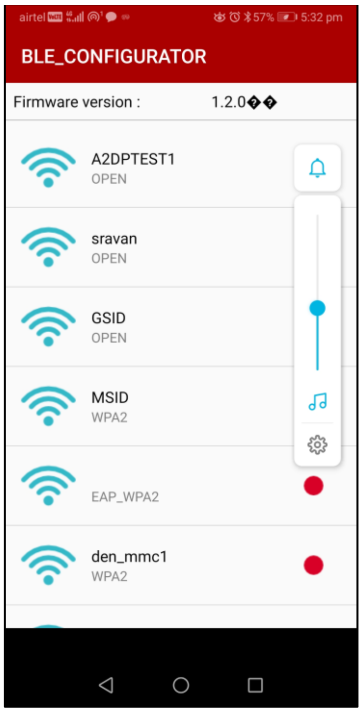


- Click on BLE_CONFIGURATOR, Device gets connected

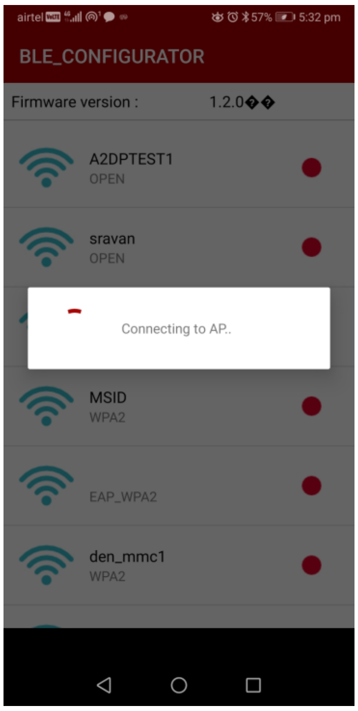


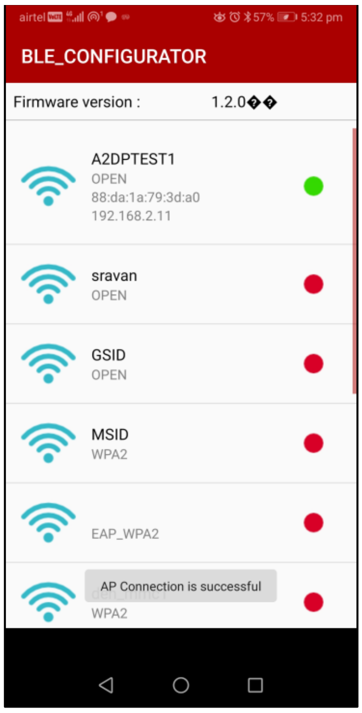


- Once the BLE gets connected, list of available Access Points gets displayed on the screen



- Connect to the Access Point.





BLE_CONFIGURATOR

Firmware version : 1.2.0��

A2DPTEST1
OPEN
88:da:1a:79:3d:a0
192.168.2.11

sravan
OPEN

GSID
OPEN

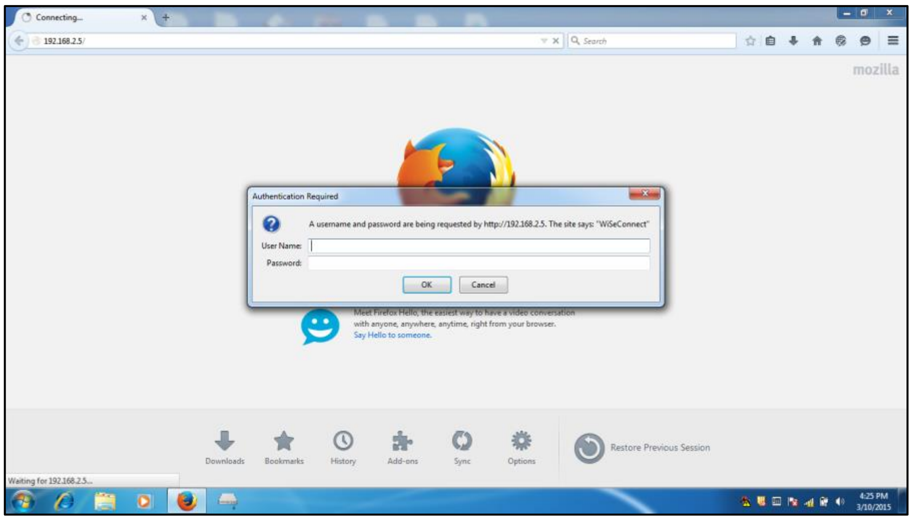
MSID
WPA2

EAP_WPA2

AP Connection is successful
WPA2

7.2 Firmware Upgrade Mechanisms

The firmware of the module can be upgraded wirelessly through web server. To upgrade the firmware wirelessly, user has to open configuration page. In the given example, module is in WLAN client mode. Module and some other host are connected to an AP. Module gets IP 192.168.2.5. When opened the module's webpage on the other host, it asks for the login credentials. To open the modules configuration page, the credentials for Username should be given as "redpine" and password should be given as "admin".



Authentication Required

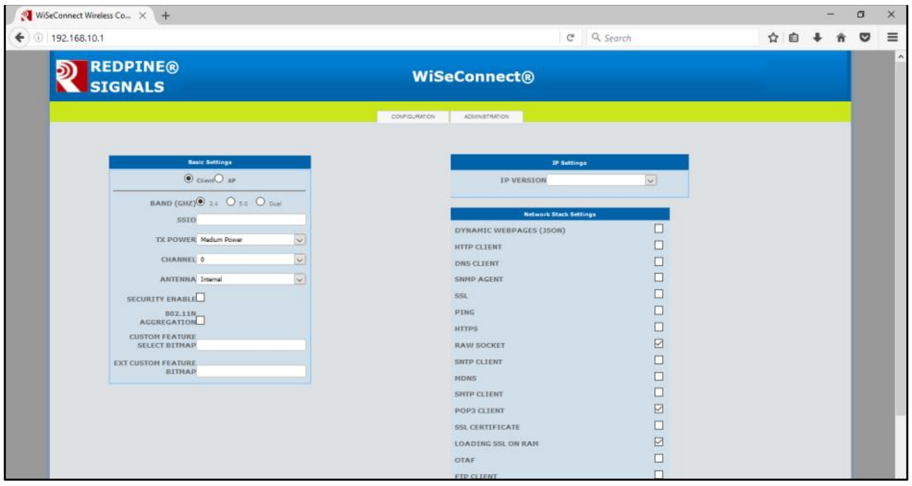
A username and password are being requested by http://192.168.2.5. The site says: "WiSeConnect"

User Name:

Password:

After entering the login credentials, the module's configuration page is opened as shown in the figure below.

Note:
'Authentication Required' pop up window will only appear if BIT[23] in Custom feature bitmap is enabled.



Basic Settings

[✓ Client] AP

BAND (GHZ) [✓ 2.4] 5.0 Dual

SSID	
TX POWER	Medium Power
CHANNEL	0
ANTENNA	Internal
SECURITY ENABLE	☐
802.11N AGGREGATION	☐
CUSTOM FEATURE SELECT BITMAP	
EXT CUSTOM FEATURE BITMAP	

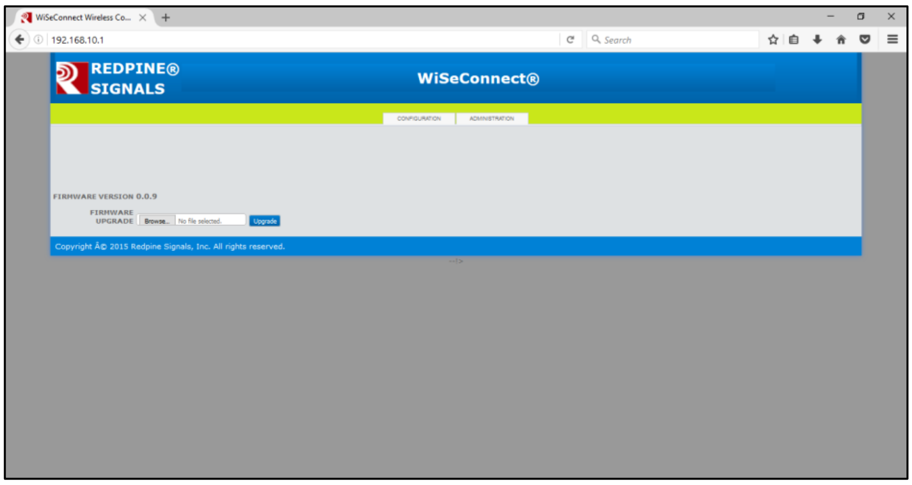
IP Settings

IP VERSION

Network Stack Settings

DYNAMIC WEBPAGES (JSON)	☐
HTTP CLIENT	☐
DNS CLIENT	☐
SNMP AGENT	☐
SSL	☐
PING	☐
HTTPS	☐
RAW SOCKET	✓
SNTP CLIENT	☐
MDNS	☐
SMTP CLIENT	☐
POP3 CLIENT	✓
SSL CERTIFICATE	☐
LOADING SSL ON RAM	✓
OTAF	☐
FTP CLIENT	☐

Click on ADMINISTRATION button to go to the wireless firmware up-gradation page.

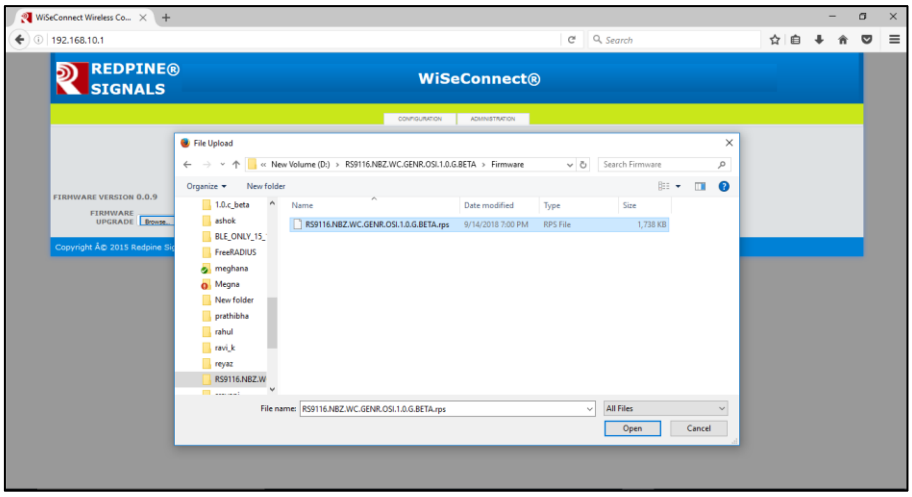


FIRMWARE VERSION 0.0.9

FIRMWARE UPGRADE No file selected.

Copyright Â© 2015 Redpine Signals, Inc. All rights reserved.

Browse for the rps file(RS9116.WC.GEN.OSI.x_x_x.rps) on the host to upgrade the module firmware and click on UPGRADE, as shown in the figure below.



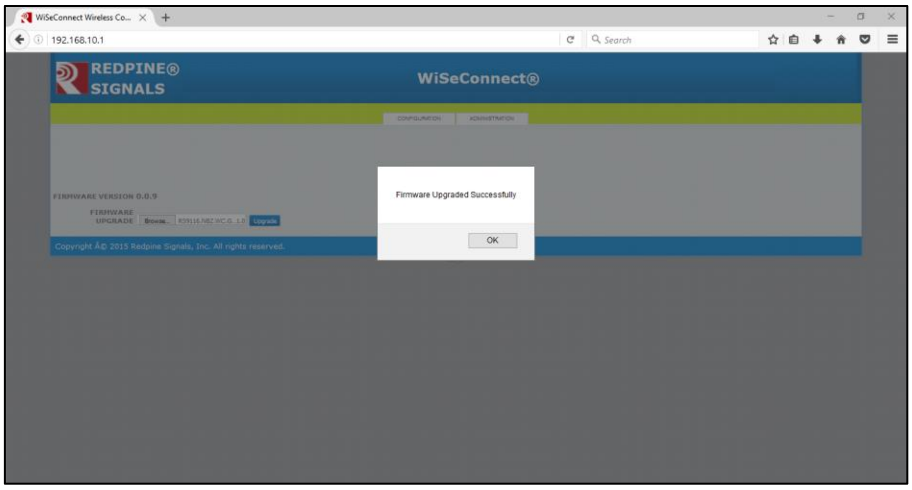
« New Volume (D:) › RS9116.NBZ.WC.GENR.OSI.1.0.G.BETA › Firmware

Name	Date modified	Type	Size
RS9116.NBZ.WC.GENR.OSI.1.0.G.BETA.rps	9/14/2018 7:00 PM	RPS File	1,738 KB

File name: RS9116.NBZ.WC.GENR.OSI.1.0.G.BETA.rps All Files

Once the remote peer has pressed upgrade button on the webpage, if module is connected through UART or USB-CDC interface then host will get asynchronous notification as "AT+RSI_FWUPREQ". So, host has to issue AT+RSI_FWUPOK. For SPI and USB interfaces, an asynchronous message with response id 0x59 is sent to host and then host has to respond with request id 0x59 (through API). If host fails to reply within the specified timeout (~20 seconds), request will expire, and upgradation process terminates.

After the firmware upgraded is successful, a popup appears on remote peer screen to intimate the process complete. Similarly, asynchronous success message (for UART/USB-CDC "AT+RSI_FWUPSUCCESS" and for SPI/USB response id 0x5A) will be forward to host connected with the module.



Firmware Upgraded Successfully

Note:

1. Wireless firmware upgradation is supported only for the latest versions of Firefox and Google chrome.
2. When user clicks on upgrade button, module starts erasing flash for storing image. This may take few seconds and upgradation starts automatically.
3. After wireless firmware upgrade, after reboot user needs to wait for few minutes (~ 1.5 minutes) for the bootloader to copy upgraded image into actual flash location.

7.3 Wake on Wireless

RS9116 Module can send packets to host in wake on wireless mode in any of the two modes listed below.

7.3.1 Active High Interrupt Mode

If BIT(16) of config_feature_bit_map is enabled in opermode command, then active high interrupt mode is enabled.

When the RS9116 Module wants to send packets to host, it asserts WAKEUP_FROM_DEV pin.

The following are the sequence of steps in UART mode:

1. RS9116 Module asserts WAKEUP_FROM_DEV pin when data is pending from module and polls for ack (HOST_WAKEUP_INDICATION pin to be high) before starting the transfer.
2. After recognizing WAKEUP_FROM_DEV pin asserted, host should ack the request from module by asserting HOST_WAKEUP_INDICATION pin and poll WAKEUP_FROM_DEV pin for module confirmation (WAKEUP_FROM_DEV is low).
3. After recognizing ack (HOST_WAKEUP_INDICATION pin high) from host, module de-asserts WAKEUP_FROM_DEV pin and start transfer.
4. Once WAKEUP_FROM_DEV is de-asserted, host should de-assert HOST_WAKEUP_INDICATION pin and receive the packet from module.

The flow chart below shows how the data is received from RS9116 Module in active high interrupt mode :

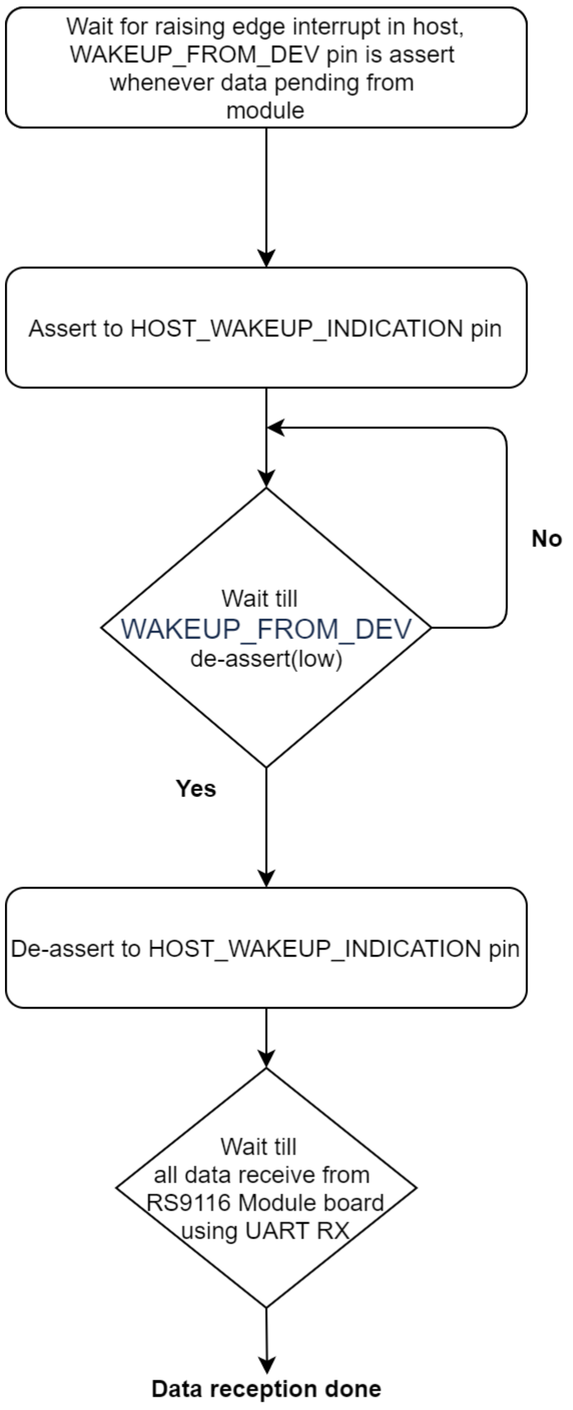

Wait for raising edge interrupt in host, WAKEUP_FROM_DEV pin is assert whenever data pending from module
Assert to HOST_WAKEUP_INDICATION pin
Wait till WAKEUP_FROM_DEV de-assert(low)
No
Yes
De-assert to HOST_WAKEUP_INDICATION pin
Wait till all data receive from RS9116 Module board using UART RX
Data reception done

Figure 28: Active High Interrupt Mode

7.3.2 Active Low Interrupt Mode

If BIT(16) of config_feature_bit_map is not enabled in opermode command, then active low interrupt mode is enabled.

When the RS9116 Module wants to send packets to host, it de-asserts WAKEUP_FROM_DEV pin.

The following are the sequence of steps in UART mode:

1. RS9116 Module de-asserts WAKEUP_FROM_DEV pin when data is pending from module and polls for ack (HOST_WAKEUP_INDICATION pin to be high) before starting the transfer.
2. After recognizing WAKEUP_FROM_DEV pin de-asserted, host should ack the request from module by asserting HOST_WAKEUP_INDICATION pin and poll WAKEUP_FROM_DEV pin for module confirmation (WAKEUP_FROM_DEV pin is high).

3. After recognizing ack (HOST_WAKEUP_INDICATION pin high) from host, module asserts WAKEUP_FROM_DEV pin and start transfer.
4. Once WAKEUP_FROM_DEV is asserted, host should de-assert HOST_WAKEUP_INDICATION pin and receive the packet from module.

The flow chart below shows how the data is received from RS9116 Module in active low interrupt mode:

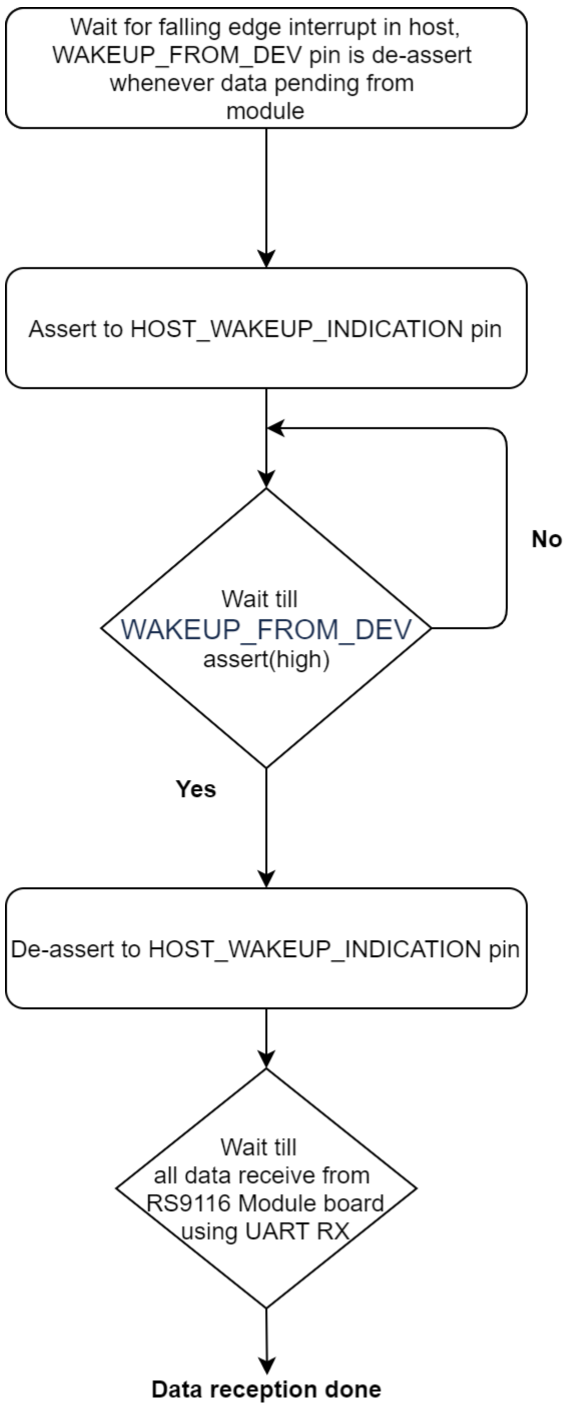

Wait for falling edge interrupt in host, WAKEUP_FROM_DEV pin is de-assert whenever data pending from module
Assert to HOST_WAKEUP_INDICATION pin
Wait till WAKEUP_FROM_DEV assert(high)
No
Yes
De-assert to HOST_WAKEUP_INDICATION pin
Wait till all data receive from RS9116 Module board using UART RX
Data reception done

Figure 29: Active Low Interrupt Mode

Note:

Since UART is asynchronous interface, Polling mechanism is suggested.

Polling for HOST_WAKEUP_INDICATION is valid only if BIT(11) in custom feature bit map of opermode command is enabled in UART mode.

If BIT(11) of custom_feature_bit_map is not enabled in opermode command, the WOW feature in UART follows the below steps like other interfaces.

For other host interfaces like SDIO /SPI /USB /USB-CDC:

When RS9116W wants to send packets to host,

1. RS9116W drives WAKEUP_FROM_DEV pin low ('0').
2. Host should wake up and take the packet from RS9116W.
3. Once the packet is received by host, RS9116W drives WAKEUP_FROM_DEV pin high ('1').

Note:

BIT(11) of custom_feature_bit_map is not valid in SPI/SDIO/USB/USB-CDC interfaces.

8 WLAN AT Command Changes/Enhancements

S.No	Configuration/Parameter	Existing Configuration	New/Modified Configuration	Comments
-	-	-	-	-

9 Revision History

Revision Number	Version Number	Date	Changes
1	1.0		Advance version
2	1.2		A note to specify the usage of channel bit map. Added procedure to switch from binary to AT and vice-versa.
3	1.3		Added Broadcast API in the chapter WLAN Commands
4	1.4		Added the broadcast API, added the tcp_rx_window_size_cap parameter in the socket_config command, added the note for the TCP/IP stack and added the antenna command in the chapter WLAN Commands.
5	1.5		1. Added Error code in Error codes section 2. Added HTTP PUT Response structure 3. Added WLAN keep alive configuration support in request_timeout command 4. Added UART debug prints selection and de selection option in exteded custom feature bitmap in opermode command 5. 0 – Selecting single socket is enabled changed as 1 -Selecting single socket is enabled 6. Modified the UART debug prints information
6	1.6		1. Added Setregion command and DFS channels in wireless Configuration 2. Changed the PER Continuous flow and Added the preamble duty cycling bit in the feature frame command 3. Added the TCP ACK division factor changes in ext_tcp_ip_feature_bitmap and socket_config command 4. Added Feature frame command request and response type in Binary Command Mode 5. Added changes for "at+rsi_host_rtc_time" command in WLAN Commands 6. Corrected join security possible values information in WLAN Commands 7. Added ext tcp ip feature bit map in WLAN Commands and certificate valid command in Binary Command Mode 8. Added Bit map in ext_tcp_ip_feature_bit_map[7] for DNS_CLIENT_BYPASS
7	1.7		1. Added information about RAW socket feature 2. Added rsi_config command and possible error codes in the PRM 3. Added note section in band command related to wifi derect and 11j mode 4. Modified rsi_wireless_antenna api. 5. Added few error codes 6. Added comment in ext_custom_feature_bit_map bit 6, 40mhz not supported in 11j AP 7. Edit note section in at+rsi_snmp_get_rsp, to explain counter64 "data reverse send case" 8. Added gain table user configurable command information 9. Added a Note in opermode for custom feature bit map 10. Added precondition for Antenna selection. 11. Safe upgrade in progress is mentioned for IMAGE_STORED_IN_DUMP in Bootloader for similarity with UART messages

Revision Number	Version Number	Date	Changes
			12. Added description of BIT(22) in extended custom feature bitmap as crystal clock selection
8	1.8		Added rsl_get_ram_dump command
9	1.9		1. Added config feature bitmap in opermode 2. Updated soft reset command description for UART/USB-CDC 3. Added no.of sockets supported in MQTT connection
10	1.10		1. Added ext_tcp_ip_feature_bit_map[16] description in ext_tcp_ip_feature_bit_map. 2. Added BIT(1) SCAN RESULTS TO HOST description in scan_feature_bitmap in scan command
11	1.11		1. Added MQTT AT commands 2. Added Error type for MQTT Timeout 3. Added next sequence of commands for MQTT user commands 4. Added Response for MQTT_INIT command and a note for possible opermode for MQTT client usage 5. Removed note which says Antenna selection command is not supported
12	1.12		1.Modified the parameters of "at+rsi_setregion", command by adding the third parameter-"module type". 2.Removed the commands, which appeared twice in " Response ID's for Rx operation table”, in binary command mode document. 3.Removed the twice appeared error codes, in "WLAN error codes" table. 4.Modified the note which says that UART flow control feature is required only for 926100 baudrate.
13	1.13		Added a Note to run the CW mode
14	1.14		1. Added UART CTS/RTS pin set selection from host in UART flow control command 2. Updated XTAL clk and Power save GPIOs info according to latest data sheet 3. Removed module_type from setregion command 4. Rephrased description about AP blacklisting feature
15	1.15		1. Added support for additional memory for SSL connections by configuring BIT(30) in extended TCP/IP feature bitmap. 2. Updated error codes
16	1.16		Updated revision
17	1.17		1. Updated MQTT command parameters and MQTT error codes.
18	1.18		Updated MQTT Document 1.Removed Timeout command. 2.Added Prerequisites for every command. 3.Added MAX length in publish command and few more changes related to MQTT.
19	1.9	Apr 2020	Updated HTTP response with additional field for status code
20	2.0	Sep 2020	1. Added the description of BIT(7) for http_client commands, "that bit is used to enable user defined http_content_type". 2. Updated MQTT commands and error codes.

Revision Number	Version Number	Date	Changes
			3. Added the weekday parameter and its description in Set and Get RTC time command. 4. Added NOTE for the configuration of listen_interval in association request related to join command. 5. Added NOTE in 'Open Socket' section, regarding max number of LTCP sockets supported with MQTT connection. 6. Added a note specifying tcp_ip_feature_bit_map[8] has to enabled in oper_mode in "Associate to an Access Point (with WPA2-PSK security) as a client" example and modified the oper_mode accordingly. 7. Modified the description of ht_caps_bit_map[1], "it is used for Channel_Width_Support". Also modified ht_caps_bit_map[8]. 8. Merged 'Architecture Overview' and 'Wi-Fi Software Programming' sections. 9. Corrected maximum length of SSID to 32 bytes from 34 bytes. 10. Added description for 'ext_tcp_ip_feature_bit_map[31]'. 11. Removed 'Related Resources' section. 12. Moved 'SPI Interface', 'UART Interface', 'USB Interface' and 'SDIO Interface' sections to 'Host Interfaces'. 13. Moved 'Command Mode Selection' section to 'Bootloader' section. 14. Removed 'PUF Commands'. 15. Renamed 'Using Different Wi-Fi Operation' section to 'Wi-Fi Operation Modes' and moved to Appendix A. 16. Moved 'Wireless Configuration' to Appendix B. 17. Removed 'Wireless Firmware Upgrade', 'Wake on Wireless' sections and 'Power save Modes description from 'Power Mode' section. 18. Renamed 'Appendix A: Sample flow of commands for Wi-Fi over UART' to 'Appendix C: Sample AT command sequences'. 19. Renamed document name from 'Embedded WLAN Software Programming Reference Manual (PRM)' to 'RS9116W Wi-Fi AT Command Programming Reference Manual'. 20. Added the description of WLAN Statistics command. 21. Added a NOTE specifying that the User can configure listen_interval dynamically in Power Mode command. 22. Added a NOTE point in Set Certificate with Indices saying that the user has to set BIT(31) in tcp_ip_feature_bit_map & BIT(29) in ext_tcp_ip_feature_bit_map to open 3 SSL Client sockets. 23. Provided some information about the support of 3 SSL Client certificates loaded into FLASH in 'Set Certificate with Indices' command. 24. Added the new bit configurations supported in config_feature_bitmap of oper_mode. 25. Added the description of Bit 5 in MODE argument of 'rsi_ipconf' command. 26. Added description for the bit BIT(5) in 'FLAGS' parameter in all the HTTP_CLIENT related commands. 27. Added a NOTE about the port 30000, in Open Socket command 28. Updated Channel Bitmap for 5 GHz 29. Removed all Binary Commands

Revision Number	Version Number	Date	Changes
			30. Modified the Username to "redpine" in 'Http Server Credentials from Host' command. 31. Modified open 3 SSL socket information in WLAN PRM 32. Added SNMP GET response in WLAN PRM 33. Renamed the username 'ap_ssid' to 'redpine' in http section. 34. Added note - 'maximum data that can be sent over TCP-SSL/LTCP-SSL is 1370 Bytes only'. 35. Removed SDIO, USB, SPI from Host Interfaces section. 36. Added 'Wireless Features and Mechanisms' section with sub sections 'Connect Application', 'Firmware Upgrade Mechanisms', 'Power Save Modes' and 'Wake on Wireless'. 37. Added 'Changes/Enhancements in WLAN AT Commands, Configurations and Mechanisms' section. 38. Added a note in Opermode section in WLAN AT Commands section. 39. Added/modified description for 'WLAN Statistics' command in 'WLAN Commands' section. 40. Added length for each AT CMD (parameters and responses.) 41. Added bitmap for TLSv1.0 and TLSv1.2 in config_feature_bitmap. 42. Added note for Wi-Fi stats command (only support Wi-Fi client mode). 43. Added supported Curve IDs for Ciphers in section ‘Open Socket’. 44. Added HTTP OTAF AT command. 45. Changed HTTP OTAF response format.

Revision Number	Version Number	Date	Changes
21	2.1	Feb 2021	1. Modified the Power save mode 9 figure Master WLAN Commands and Master Power Save modes. 2. Removed redundant information on "Power save modes" under section Wireless Features and Mechanisms. 3. Corrected Section 5.97 by removing Band command description as it is already described in section 5.2. 4. Added a column specifying the bit_info in join_feature_bitmap. 5. Removed "Generate MFI Authentication Signature" command as that command is not validated with the current release. 6. Specified that at+rsi_gain_table region based user gain values in 2g must be doubled before loading. 7. Added config_feature_bitmap[25:24] to configure 40MHz XTAL Good time in µs. Refer WLAN Commands section 5.1, Set Operating Mode. 8. Removed SPI and USB host interaction and Bypass Mode in SPI / USB details from section 2 Bootloader. 9. Updated config_feature_bitmap[25:24], refer to WLAN Commands in Section 5.1. 10. Removed draw.io Watermark from the Power Save Mode figures under Section 5.16. 11. Removed information about power save modes, under Wireless Features and Mechanisms in Section 7. 12. Corrected the word RS9916W to RS9116W under Host Interfaces, Section 3. 13. Corrected the phrase Silicon's labs to Silicon Labs in a Note point at config_feature_bitmap under Section 5.1. 14. Mentioned the default port number as 80 for all the http commands, under Sections 5.45, 5.46, 5.47, and 5.48. 15. Corrected the phrase HTTP GET to HTTP POST at Response Parameters of HTTP Post command under Section 5.46. 16. Mentioned the Keep alive time-period at MQTT Init command under Section 5.97. 17. Mentioned to set bit[31] in tcp_ip_feature_bitmap in order to enable ext_tcp_ip_feature_bit_map[17] in the opermode command at MQTT client, under Section 5.97. **Note**: This document should be used with WiSeConnect version 2.3.0.

10 Appendix A: Wi-Fi Operation Modes

The module can be configured in the following modes:

- Access Point Mode
- Client Mode to connect to an AP in open mode or with Personal Security
- Client mode to connect to an AP with Enterprise Security
- PER mode

Access Point Mode

The following sequence of commands should be used to create an Access Point in the module. The module can support eight external clients when it is configured in Access Point mode. By default the module can act as a DHCP server.

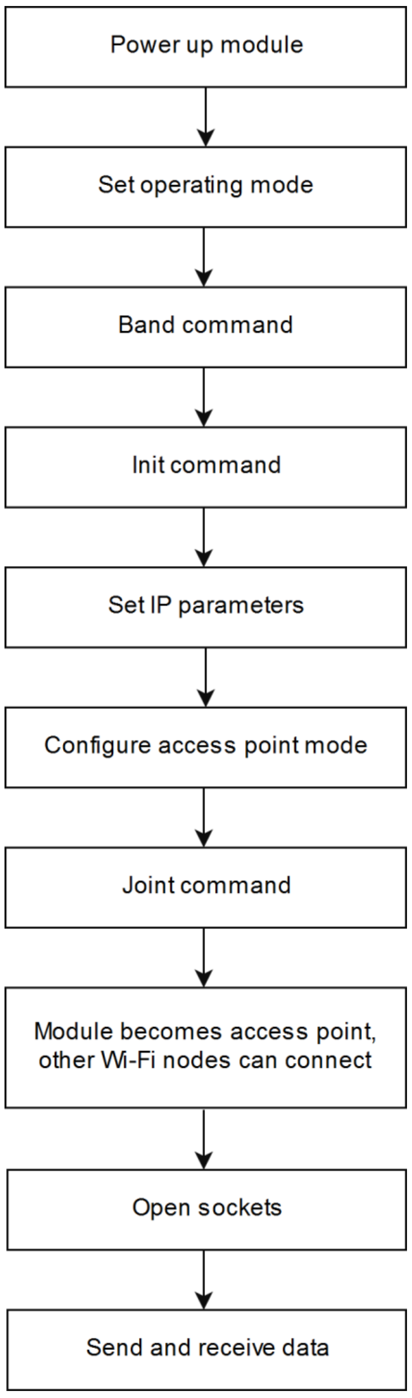


Figure 30: Access Point Mode

Client Mode with Personal Security

In this mode, the module works as a Wi-Fi client. It can connect to an Access Point with open mode or Personal Security.

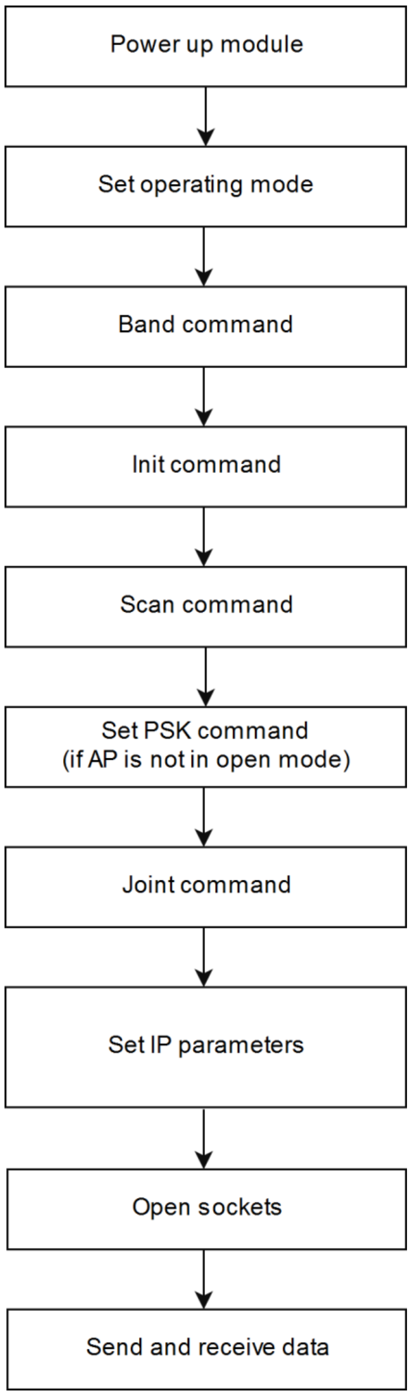


Figure 31: Client Mode with Personal Security

Client Mode with Enterprise Security

In this mode, the module works as a client to connect to an Enterprise security enabled network that Hosts a Radius Server.

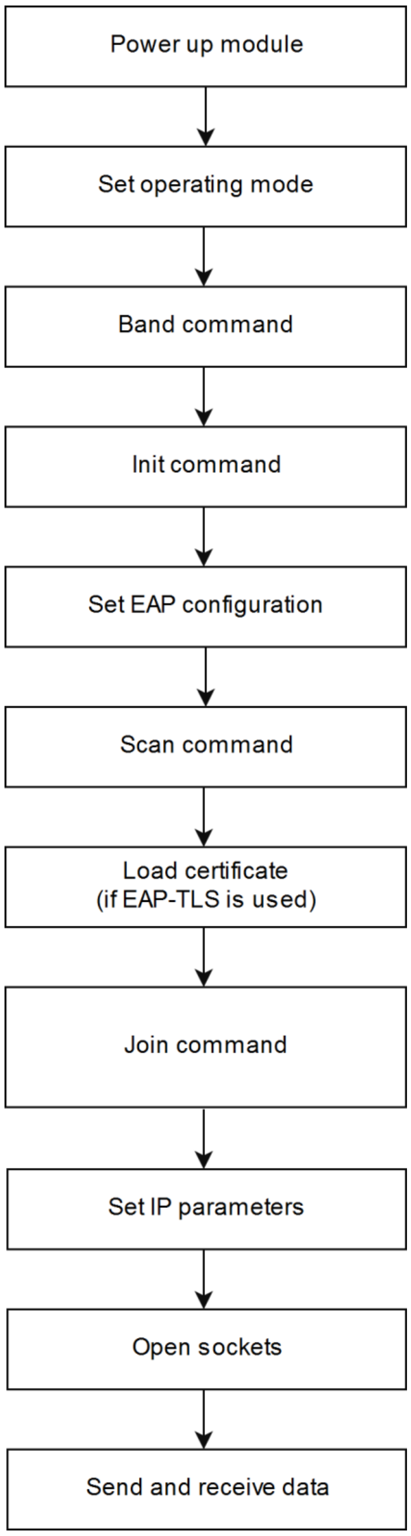


Figure 32: Client Mode with Enterprise Security

PER Mode

This mode is used for Packet Error Rate analysis and regulatory and compliance certification e.g. FCC/IC/CE/TELEC

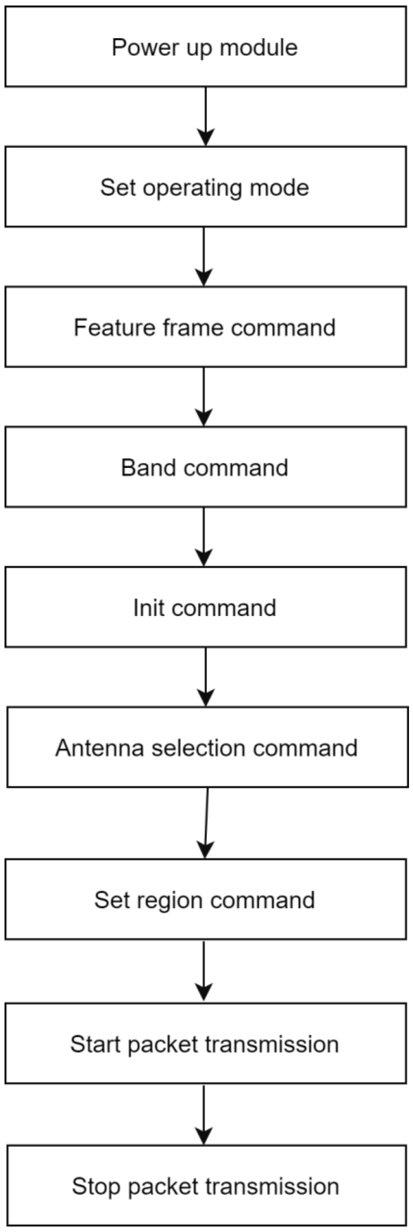


Figure 33: PER Mode

11 Appendix B: Wireless Configuration

The module can be configured wirelessly to join a specific AP (referred to as "auto-connect") or create an Access Point (referred to as "auto-create").

Configuration to Join a Specific AP

Flow 1: In this flow, an AP is first created in the module, to which a remote device connects and configures the module.

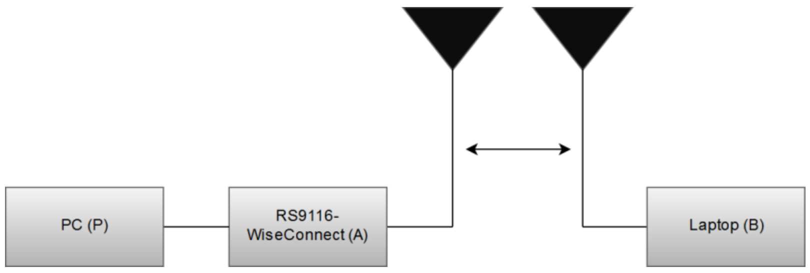


Figure 34: Setup for Configuration to Join a Specific AP – Flow 1

1. Connect a PC or Host to the module through any interface and power up the module.
2. Configure the module in order to act as an AP by issuing commands from PC (P) (refer 'APPENDIX C: Sample AT commands Sequences').
3. Connect a Laptop (B) to the created AP. Open the URL **http://<Module's IP address>** in the Laptop. For example, if the module is configured to have an IP of 192.168.2.5, then the URL will be http://192.168.2.5. Make sure the browser in the laptop does not have any proxies enabled. This will open the following index page:
4. Enter the login credentials username as "redpine" and password as "admin" and click on **OK** button to make changes in configurations.
5. In the opened web page, select **"Client mode"** and enter desired values.
    a. SSID: This is the SSID of the AP to which the module should connect after configuration is over.
    b. Band : Single band (2.4 GHz or 5.0 GHz) or dual band (2.4Ghz and 5GHz).
    c. Tx Power: RF power for Tx (refer to the TxPower parameter in command Join). Allowed values are 0, 1 and 2.
    d. Set_region : This is used to configure the device to operate according to the regulations of its operating country.
    e. Channel: Channel number at which the target AP is present. Refer to the **PER Mode** command section for more details.
    f. Security Enable: This should match the security mode of the AP to which the module should connect.
    g. IPversion: select IPv4 / IPv6 / Dual stack mode.
    h. DHCP: If DHCP is selected, the module will work as a DHCPv4 client, otherwise, an IPv4 address should be hard coded in the web page.
    i. IPv6 DHCP: If IPv6 DHCP is selected, the module will work as a DHCPv6 client, and otherwise, an IPv6 address should be hard coded in the web page.
    j. Enable optional features like Dynamic webpages, HTTP client, SNMP client, DNS Client, PING, SSL Feature select bit maps based on features used. Refer **Set Operating Mode command** for further details.
6. Click on **"Submit"** button. The information is sent to the module and stored in its internal flash.

The module should now be power cycled or hard reset. It boots up and then automatically scans channels for the target AP and connects to it and gets an IP address. The module will send out two responses to the host, the first corresponds to the "Join" command and the second to the "Set IP Parameters" command. Note that once the module is restarted, no commands need to be given. The module automatically scans and joins the target AP, after which the stored configuration parameters can be retrieved using the command Get Information about Stored Configuration. If the auto-connect feature needs to be disabled, issue the command Enable auto-join to AP or Auto-create AP to the module.

Note:
If the information sent is wrong than module will throw an error while scanning or joining. In that case user need to disable the auto configuration. User can issue the command at+rsi_configsave=0\r\n and power cycle the module.

Flow 2: In this flow, the module is connected to an AP. A remote device connects to the same AP and configures the module.

1. Connect a PC or Host to the module through any interface and power up the module.
2. Configure the module to become a client and connect to an AP, by issuing commands from PC (P) (refer 'APPENDIX C: Sample AT commands Sequences')**.**

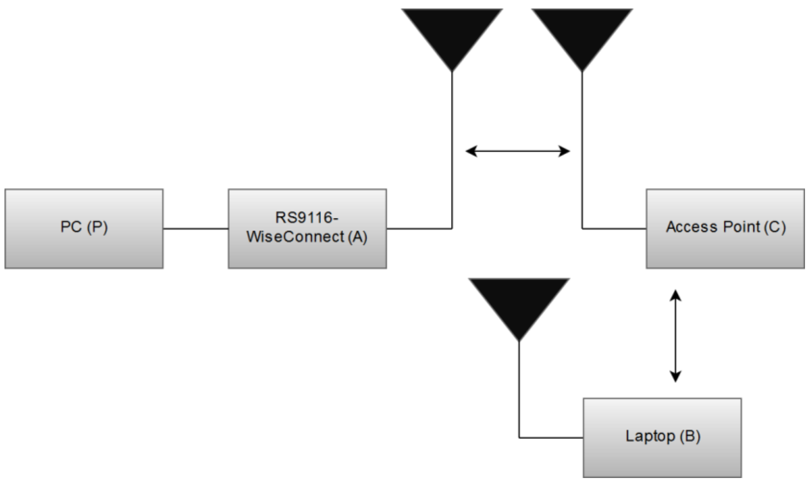
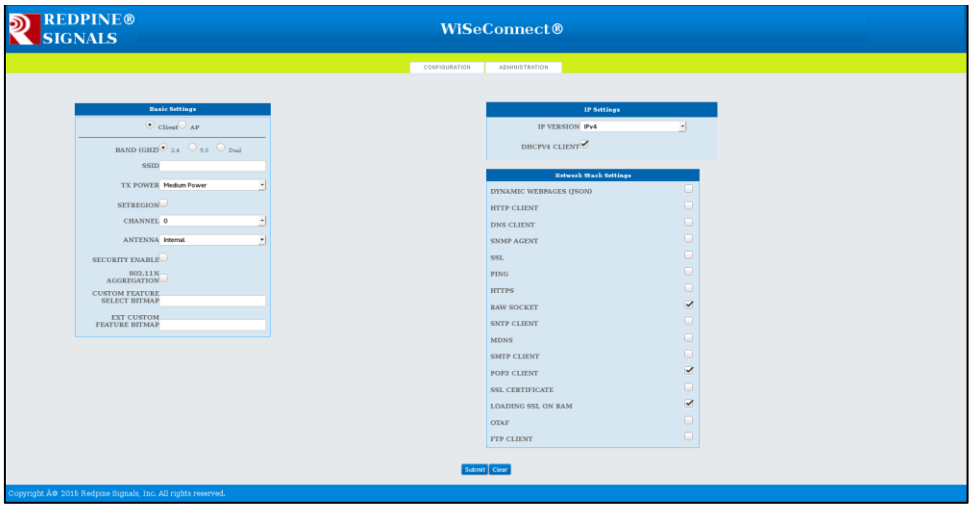


Figure 35: Setup for Configuration to Join a Specific AP – Flow 2

Connect a Laptop (B) to the same AP. Open the URL **http://<Module's IP address>** in the Laptop. For example, if the module is configured to have an IP of 192.168.2.5, then the URL will be http://192.168.2.5. Make sure the browser in the laptop does not have any proxies enabled. This will open the index page as shown in Flow 1 above. Enter the credentials such as username- **"redpine**" and password -**"admin"**.

3. In the opened web page , select **"Client mode"** and enter desired values.
   a. SSID: This is the SSID of the AP to which the module should connect after the configuration is over.
   b. Band: Single Band (2.4GHz or 5GHz) or Dual Band (2.4 GHz and 5.0GHz).
   c. Tx Power: RF power for Tx (refer to the TxPower parameter in command Join) . Allowed values are 0, 1 and 2.
   d. Set-region : This is used to configure the device to operate according to the regulations of its operating country.
   e. Security Enable: This should match the security mode of the AP to which the module should connect.
   f. DHCP: If DHCP is selected, the module will work as a DHCP client, otherwise, an IP should be hard coded in the web page.
   g. Channel: Channel number at which the target AP is present. Refer to the **PER Mode** command section for more details.



Basic Settings

(● Client) ( ) AP

BAND (GHZ)	(● 2.4) ( ) 5.0 ( ) Dual
SSID	
TX POWER	Medium Power
SETREGION	☐
CHANNEL	0
ANTENNA	Internal
SECURITY ENABLE	☐
802.11N AGGREGATION	☐
CUSTOM FEATURE SELECT BITMAP	
EXT CUSTOM FEATURE BITMAP	

IP Settings

IP VERSION	IPv4
DHCPV4 CLIENT	✓

Network Stack Settings

DYNAMIC WEBPAGES (JSON)	☐
HTTP CLIENT	☐
DNS CLIENT	☐
SNMP AGENT	☐
SSL	☐
PING	☐
HTTPS	☐
RAW SOCKET	✓
SNTP CLIENT	☐
MDNS	☐
SMTP CLIENT	☐
POP3 CLIENT	✓
SSL CERTIFICATE	☐
LOADING SSL ON RAM	✓
OTAF	☐
FTP CLIENT	☐

Submit
Clear
Copyright Â© 2016 Redpine Signals, Inc. All rights reserved.

Figure 36: Webpage Screenshot

h. Security Enable: Select if security mode is enable and fill the PSK, Security Type, encryption Type fields accordingly.

i. IPconfiguration: Select the IP version (IPv4/IPv6/Both) and provide static IP details or enable DHCP.

4. Enable optional features like Dynamic WebPages, HTTP client, SNMP client, DNS Client, Feature select bit maps based on features used. Refer **Operation Mode command** for further details.
5. Click on "Submit" button. The information is sent to the module and stored in its internal flash.
6. The module should now be power cycled or hard reset. It boots up and then automatically scans channels for the target AP and connects to it and gets an IP address. The module will send out two responses to the Host, the first corresponds to the "Join" command and the second to the "Set IP Parameters" command. Note that once the module is restarted, no commands need to be given. The module automatically scans and joins the target AP, after which the stored configuration parameters can be retrieved using the command Get Information about Stored Configuration. If the auto-connect feature needs to be disabled, issue the command Enable auto-join to AP or Auto-create AP to the module.

Configuring to Create an AP

Flow 1:
In this flow, an AP is first created in the module, to which a remote device connects and configures the module.

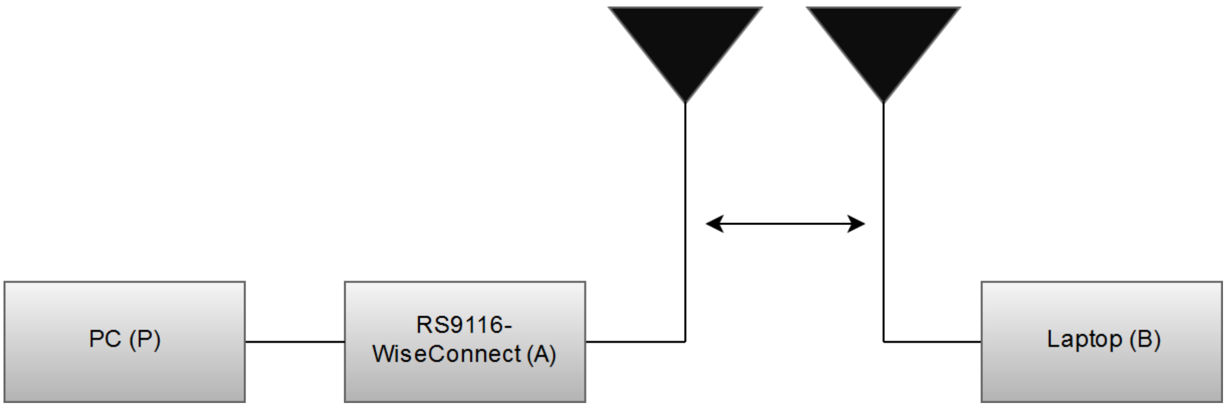
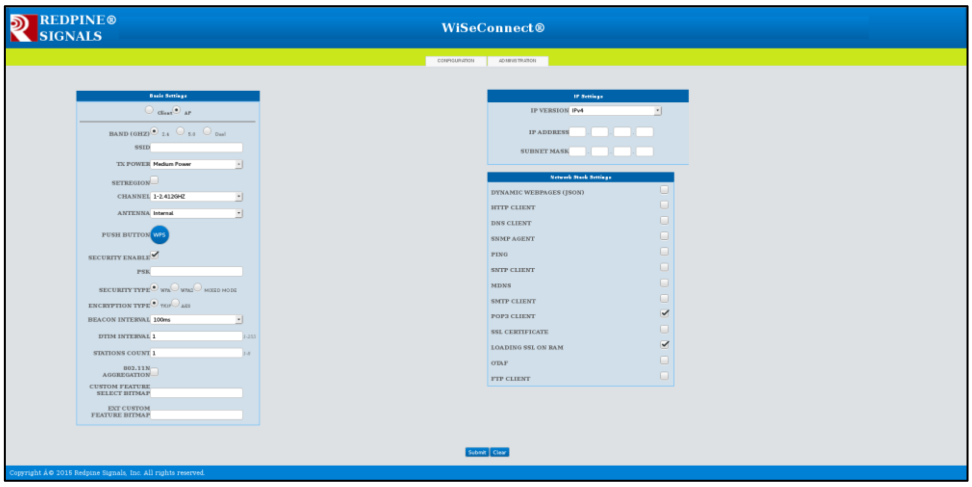


Figure 37: Setup for Configuration to Create an AP – Flow 1

1. Connect a PC or Host to the module through any interface and power up the module.
2. Configure the module to become an AP by issuing commands through PC (P). (refer APPENDIX A: Sample flow of commands for Wi-Fi over UART).
3. Connect a Laptop (B) to the created AP. Open the URL **http://<Module's IP address>** in the Laptop. For example, if the module is configured to have an IP of 192.168.2.5, then the URL will be http://192.168.2.5. Make

sure the browser in the laptop does not have any proxies enabled. This will open the index page as shown in section 6.1 Flow 1 give the credentials username as "redpine" and password as "admin".

4. In the web page that opens, select "Access Point" mode and enter desired values.
   a. SSID: This is the SSID of the AP which will be created after configuration is over.
   b. Band: Single Band (2.4GHz or 5.0GHz).
   c. Tx Power: RF power for Tx (refer to the TxPower parameter in command Join) . Allowed values are 0, 1 and 2.
   d. Setregion : This is used to configure the device to operate according to the regulations of its operating country.
   e. Security Enable: PSK, security type, encryption type. This is to configure the security mode of the AP.
   f. Channel: Channel number at which the target AP is present. Refer to the **PER Mode** command section for more details. Value of '0' is not allowed.
   g. Security Type:WPA/WPA2
   h. Encryption type: Type of encryption(TKIP/AES).
   i. IP, Mask, and Gateway: These parameters set the IP parameters of the AP.
   j. Beacon Interval and DTIM interval: This to set the beacon parameters of the AP. For example, if beacon interval is 200 (msecs) and DTIM count is 3, the DTIM interval would be 2x300=600 msecs.
5. Enable optional features like aggregation Dynamic Webpages, HTTP client, SNMP client, DNS Client, Feature select bit maps based on features used. Refer **Operation Mode command** for further details.



Basic Settings

( ) Client [✓ AP]

BAND (GHZ)	[✓ 2.4] ( ) 5.0 ( ) Dual
SSID	
TX POWER	Medium Power
SETREGION	☐
CHANNEL	1-2.412GHZ
ANTENNA	Internal
PUSH BUTTON	WPS
SECURITY ENABLE	✓
PSK	
SECURITY TYPE	[✓ WPA] ( ) WPA2 ( ) MIXED MODE
ENCRYPTION TYPE	[✓ TKIP] ( ) AES
BEACON INTERVAL	100ms
DTIM INTERVAL	1 (1-255)
STATIONS COUNT	1 (1-8)
802.11N AGGREGATION	☐
CUSTOM FEATURE SELECT BITMAP	
EXT CUSTOM FEATURE BITMAP	

IP Settings

IP VERSION	IPv4
IP ADDRESS	. . .
SUBNET MASK	. . .

Network Stack Settings

DYNAMIC WEBPAGES (JSON)	☐
HTTP CLIENT	☐
DNS CLIENT	☐
SNMP AGENT	☐
PING	☐
SNTP CLIENT	☐
MDNS	☐
SMTP CLIENT	☐
POP3 CLIENT	✓
SSL CERTIFICATE	☐
LOADING SSL ON RAM	✓
OTAF	☐
FTP CLIENT	☐

Copyright Â© 2015 Redpine Signals, Inc. All rights reserved.

Figure 38: Webpage Screenshot

6. Click on "Submit" button. The information is sent to the module and stored in its internal flash.
7. The module should now be power cycled or hard reset. It boots up and then automatically creates an AP with the configured parameters. The module will send out two responses to the Host, the first corresponds to the "Set IP Parameters" command and the second to the "Join" command. Note that once the module is restarted, no commands need to be given. The module automatically and internally executes the commands to create an AP. The stored configuration parameters can be retrieved using the command Get Information about Stored Configuration. If the auto-connect feature needs to be disabled, issue the command Enable auto-join to AP or Auto-create AP to the module.

Flow 2:
In this flow, the module is connected to an AP. A remote device connects to the same AP and configures the module.

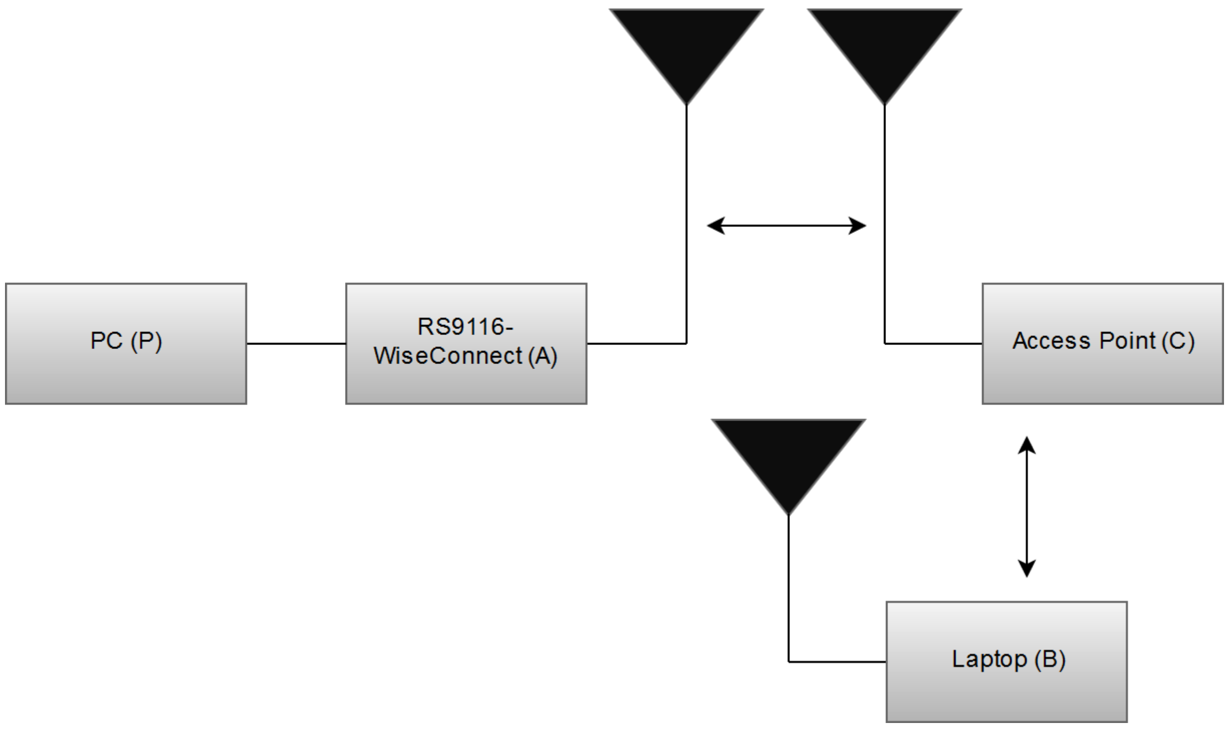
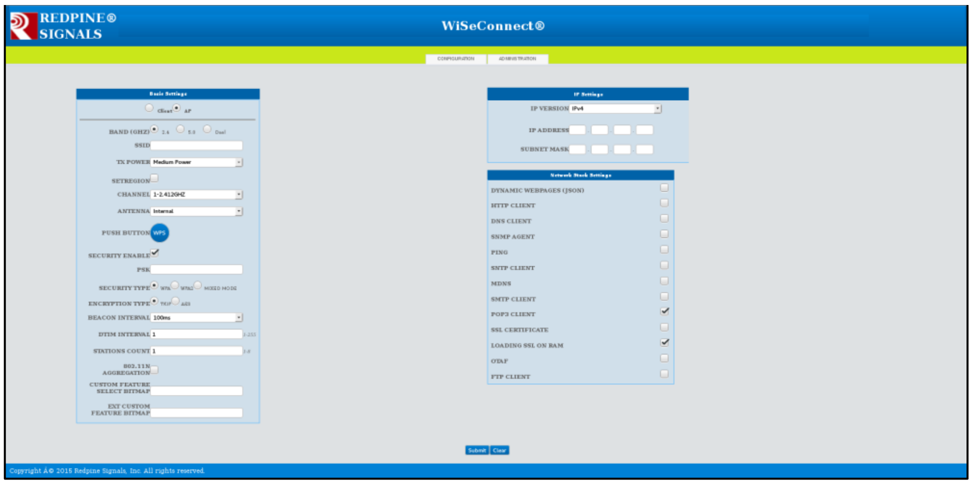


Figure 39: Setup for Configuration to Create an AP – Flow 2

1. Connect a PC or Host to the module through any interface and power up the module.
2. Configure the module to become a client and connect to an AP by issuing commands from the PC (P) (refer APPENDIX A: Sample flow of commands for Wi-Fi over UART).
3. Connect a Laptop (B) to the created AP. Open the URL **http://<Module's IP address>** in the Laptop. For example, if the module is configured to have an IP of 192.168.2.5, then the URL will be http://192.168.2.5. Make sure the browser in the laptop does not have any proxies enabled. Enter the credentials such as username - **"redpine"** and password**- "admin"**.
4. In the web page that opens, select **"Access Point"** mode and enter desired values.
   a. SSID: This is the SSID of the AP which will be created after configuration is over.
   b. Band: Single band (2.4GHz) or Dual Band (5GHz).
   c. Tx Power: RF power for Tx (refer to the TxPower parameter in command Join) . The allowed values are 0, 1 and 2.
   d. Setregion : This is used to configure the device to operate according to the regulations of its operating country.
   e. Security Enable: PSK, security type, encryption type: This is to configure the security mode of the AP.
   f. Channel: Channel number at which the target AP is present. Refer to the **PER Mode** command section for more details. Value of '0' is not allowed.
   g. IP, Mask, and Gateway: These parameters set the IP parameters of the AP.
   h. Beacon Interval and DTIM count: This to set the beacon parameters of the AP. For example, if beacon interval is 200 (msecs) and DTIM count is 3, the DTIM interval would be 2x300=600 msecs.
5. Enable optional features like Aggregation, Dynamic WebPages, HTTP client, SNMP client, DNS Client, Feature select bit maps based on features used. Refer **Operation Mode command** for further details.



REDPINE® SIGNALS

WiSeConnect®

CONFIGURATION | ADMINISTRATION

Basic Settings

Client [✓ AP]

BAND (GHZ) [✓ 2.4] 5.0 Dual

SSID

TX POWER Medium Power

SETREGION ☐

CHANNEL 1-2.412GHZ

ANTENNA Internal

PUSH BUTTON WPS

SECURITY ENABLE ✓

PSK

SECURITY TYPE [✓ WPA] WPA2 MIXED MODE

ENCRYPTION TYPE [✓ TKIP] AES

BEACON INTERVAL 100ms

DTIM INTERVAL 1 (1-255)

STATIONS COUNT 1 (1-8)

802.11N AGGREGATION ☐

CUSTOM FEATURE SELECT BITMAP

EXT CUSTOM FEATURE BITMAP

IP Settings

IP VERSION IPv4

IP ADDRESS

SUBNET MASK

Network Stack Settings

DYNAMIC WEBPAGES (JSON)	☐
HTTP CLIENT	☐
DNS CLIENT	☐
SNMP AGENT	☐
PING	☐
SNTP CLIENT	☐
MDNS	☐
SMTP CLIENT	☐
POP3 CLIENT	✓
SSL CERTIFICATE	☐
LOADING SSL ON RAM	✓
OTAF	☐
FTP CLIENT	☐

Submit | Clear

Copyright Â© 2015 Redpine Signals, Inc. All rights reserved.

Figure 40: Webpage Screenshot

6. Click on **"Submit"** button. The information is sent to the module and stored in its internal flash.

The module should now be power cycled or hard reset. It boots up and then automatically creates an AP with the configured parameters. The module will send out two responses to the Host, the first corresponds to the "Set IP Parameters" command and the second to the "Join" command. Note that once the module is restarted, no commands need to be given. The module automatically and internally executes the commands to create an AP. The stored configuration parameters can be retrieved using the command Get Information about Stored Configuration. If the auto-connect feature needs to be disabled, issue the command Enable auto-join to AP or Auto-create AP to the module.

Configuration to Join an AP with Enterprise Security

Flow 1: In this flow, an AP is first created in the module, to which a remote device connects and configures the module.

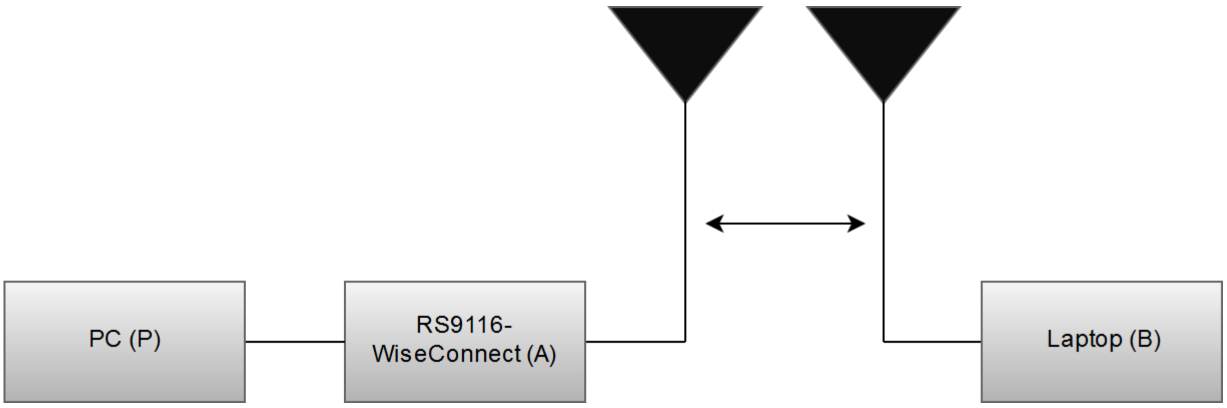


Figure 41: Setup for Configuration to Join an AP with Enterprise Security – Flow 1

1. Connect a PC or Host to the module through any interface and power up the module.
2. Configure the module to become an AP by issuing commands from PC (P) (refer APPENDIX A: Sample flow of commands for Wi-Fi over UART).
3. Connect a Laptop (B) to the created AP. Open the URL **http://<Module's IP address>** in the Laptop. For example, if the module was configured to have an IP of 192.168.2.5, then the URL will be http://192.168.2.5. Make sure the browser in the laptop does not have any proxies enabled. Enter the credentials such as username - **"redpine"** and password -"admin".
4. In the web page that opens, select "Enterprise" in security mode and enter desired values.
    a. SSID: This is the SSID of the AP to which the module should connect after configuration is over.
    b. Band: Single Band(2.4GHz or 5GHz) or Dual Band(5GHz and 2.4GHz).

c. Tx Power: RF power for Tx (refer to the TxPower parameter in command Join) . Allowed values are 0, 1 and 2.

d. Setregion : This is used to configure the device to operate according to the regulations of its operating country.

e. Security type : This should match the security mode of the AP to which the module should connect.

f. DHCP: If DHCP is selected, the module will work as a DHCP client, otherwise, an IP should be hard coded in the web page.

g. Channel: Channel number at which the target AP is present. Refer to the **PER Mode** command section for more details.

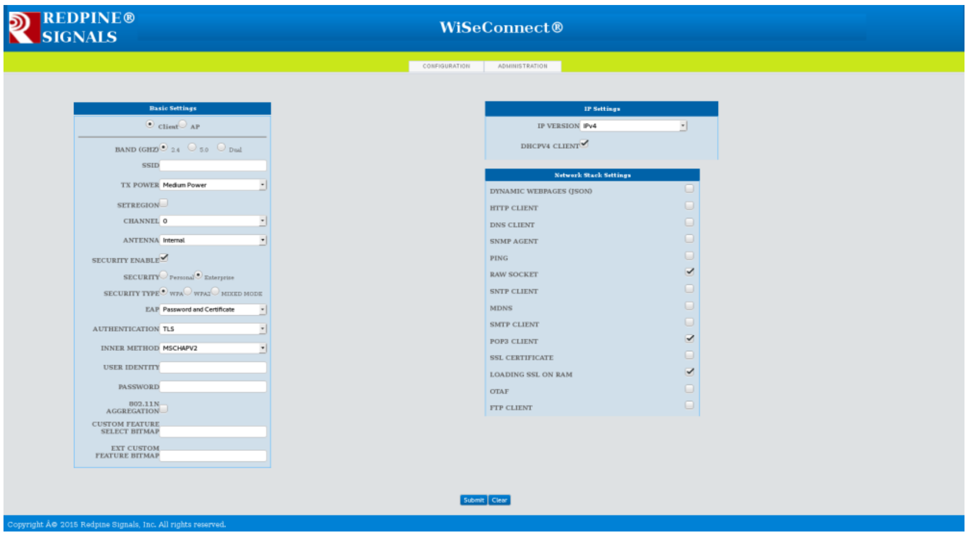
h. EAP: EAP method of the target AP.

i. Inner method: Inner method of the target AP.

j. User identity: User ID. This is present in the user configuration file in the radius sever.

k. Password: This should be same as the password in the user configuration file in the Radius Server for that User Identity.

5. Enable optional features like Aggregation, Dynamic WebPages, HTTP client, SNMP client, DNS Client, Feature select bit maps based on features used. Refer **Operation Mode command** for further details.



Basic Settings

[✓ Client] AP

BAND (GHZ)	[✓ 2.4] 5.0 Dual
SSID	
TX POWER	Medium Power
SETREGION	☐
CHANNEL	0
ANTENNA	Internal
SECURITY ENABLE	✓
SECURITY	Personal [✓ Enterprise]
SECURITY TYPE	[✓ WPA] WPA2 MIXED MODE
EAP	Password and Certificate
AUTHENTICATION	TLS
INNER METHOD	MSCHAPV2
USER IDENTITY	
PASSWORD	
802.11N AGGREGATION	☐
CUSTOM FEATURE SELECT BITMAP	
EXT CUSTOM FEATURE BITMAP	

IP Settings

IP VERSION	IPv4
DHCPV4 CLIENT	✓

Network Stack Settings

DYNAMIC WEBPAGES (JSON)	☐
HTTP CLIENT	☐
DNS CLIENT	☐
SNMP AGENT	☐
PING	☐
RAW SOCKET	✓
SNTP CLIENT	☐
MDNS	☐
SMTP CLIENT	☐
POP3 CLIENT	✓
SSL CERTIFICATE	☐
LOADING SSL ON RAM	✓
OTAF	☐
FTP CLIENT	☐

Copyright Â© 2015 Redpine Signals, Inc. All rights reserved.

Figure 42: Webpage Screenshot

6. Click on **"Submit"** button. The information is sent to the module and stored in its internal flash.

7. The module should now be power cycled or hard reset. It boots up and then automatically scans channels for the target AP and connects to it and gets an IP address. The module will send out two responses to the Host, the first corresponds to the "Join" command and the second to the "Set IP Parameters" command. Note that once the module is restarted, no commands need to be given. The module automatically scans and joins the target AP after which the stored configuration parameters can be retrieved using the command Get Information about Stored Configuration. If the auto-connect feature needs to be disabled, issue the command Enable auto-join to AP or Auto-create AP to the module.

Flow 2:
In this flow, the module is connected to an AP. A remote device connects to the same AP and configures the module.

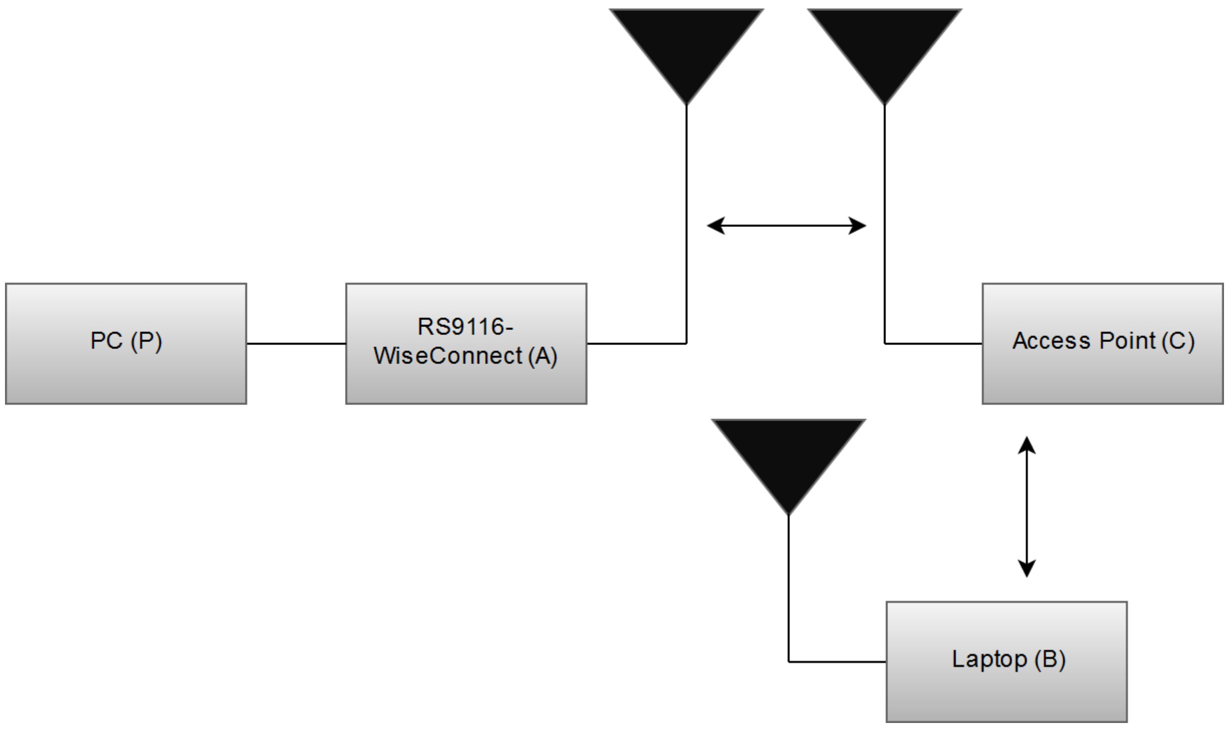
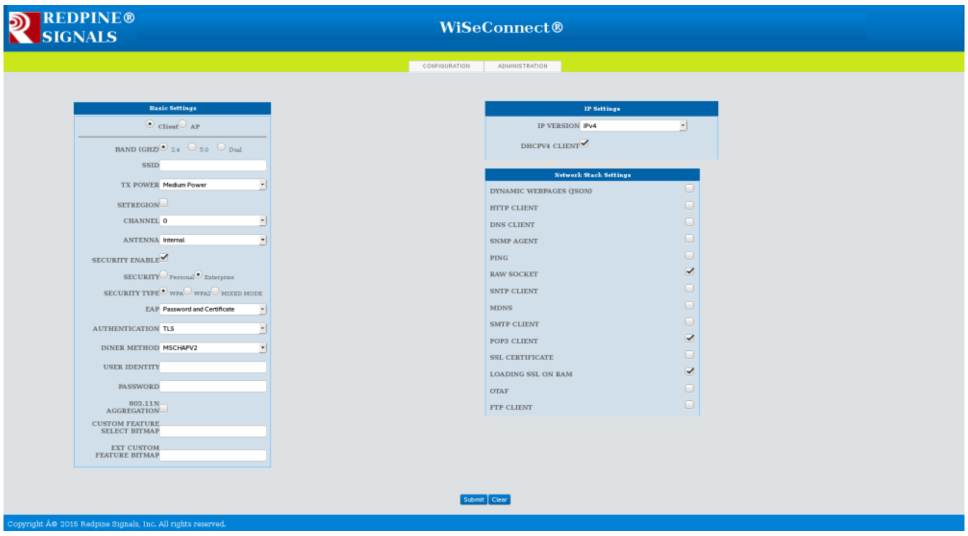


Figure 43: Setup for Configuration to Join an AP with Enterprise Security – Flow 2

1. Connect a PC or Host to the module through any interface and power up the module.
2. Configure the module to become a client and connect to an AP by issuing commands from PC (P) (refer APPENDIX A: Sample flow of commands for Wi-Fi over UART).
3. Connect a Laptop (B) to the same AP. Open the URL **http://<Module's IP address>** in the Laptop. For example, if the module is configured to have an IP of 192.168.2.5, then the URL will be http://192.168.2.5. Make sure the browser in the laptop does not have any proxies enabled. Enter the credentials such as username- **"redpine"** and password- **"admin"**.
4. In the web page that opens, select "Client mode" and enter desired values.
   a. SSID: This is the SSID of the AP to which the module should connect after configuration is over.
   b. Data rate: Physical data rate (refer to the **PER Mode** command section for more details).
   c. Tx Power: RF power for Tx (refer to the TxPower parameter in command Join) . Allowed values are 0, 1 and 2.
   d. Security mode and PSK: This should match the security mode of the AP to which the module should connect.
   e. DHCP: If DHCP is selected, the module will work as a DHCP client, otherwise, an IP should be hard coded in the web page.
   f. Channel: Channel number at which the target AP is present. Refer to the **PER Mode** command section for more details.
   g. EAP: EAP method of the target AP.
   h. Inner method: Inner method of the target AP.
   i. User identity: User ID. This is present in the user configuration file in the radius sever.
   j. Password: This should be same as the password in the user configuration file in the Radius Server for that User Identity.
5. Enable optional features like Aggregation, Dynamic WebPages, HTTP client, SNMP client, DNS Client and Feature select bit maps based on features used. Refer **Operation Mode command** for further details.



Basic Settings

[✓ Client] AP

BAND (GHZ)	[✓ 2.4] 5.0 Dual
SSID	
TX POWER	Medium Power
SETREGION	☐
CHANNEL	0
ANTENNA	Internal
SECURITY ENABLE	✓
SECURITY	Personal [✓ Enterprise]
SECURITY TYPE	[✓ WPA] WPA2 MIXED MODE
EAP	Password and Certificate
AUTHENTICATION	TLS
INNER METHOD	MSCHAPV2
USER IDENTITY	
PASSWORD	
802.11N AGGREGATION	☐
CUSTOM FEATURE SELECT BITMAP	
EXT CUSTOM FEATURE BITMAP	

IP Settings

IP VERSION	IPv4
DHCPV4 CLIENT	✓

Network Stack Settings

DYNAMIC WEBPAGES (JSON)	☐
HTTP CLIENT	☐
DNS CLIENT	☐
SNMP AGENT	☐
PING	☐
RAW SOCKET	✓
SNTP CLIENT	☐
MDNS	☐
SMTP CLIENT	☐
POP3 CLIENT	✓
SSL CERTIFICATE	☐
LOADING SSL ON RAM	✓
OTAF	☐
FTP CLIENT	☐

Copyright Â© 2015 Redpine Signals, Inc. All rights reserved.

Figure 44: Webpage Screenshot

6. Click on **"Submit"** button. The information is sent to the module and stored in its internal flash.
7. The module should now be power cycled or hard reset. It boots up and then automatically scans channels for the target AP and connects to it and gets an IP address. The module will send out two responses to the Host, the first corresponds to the "Join" command and the second to the "Set IP Parameters" command. Note that once the module is restarted, no commands need to be given. The module automatically scans and joins the target AP, after which the stored configuration parameters can be retrieved using the command Get Information about Stored Configuration. If the auto-connect feature needs to be disabled, issue the command Enable auto-join to AP or Auto-create AP to the module.

12 Appendix C: Sample AT Command Sequences

Sample command sequences are shown below for operating the module in different modes.

Create an Access Point

at+rsi_opermode=6,1,48,0\r\n
This command sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_init\r\n
This command initializes the module.

at+rsi_ipconf=0,192.168.50.1,255.255.255.0,192.168.50.1\r\n
This command can be used optionally in this flow to configure the IP (192.168.50.1 in this example) of the AP.
If this command is not issued, a default IP of 192.168.100.76 will be used.

at+rsi_apconf=1,AP_SSID,2,2,12345678,300,2,4\r\n
This command will configure the SSID of the AP to “AP_SSID” and password will be set to “12345678”.

at+rsi_join=AP_SSID,0,2 \r\n
This command will create the Access Point with SSID 'AP_SSID' where xy is a pair of alphanumeric character.
A client device (Named “Device A” in this example) can now associate to the AP, open sockets and transfer data.
For example,

at+rsi_ltcp=5001,5,0\r\n
Opens a server TCP socket inside the module with port number 5001 with tos (type ofservice) type 0 and max clients support count as 5.
A client socket at the remote node (Device A) can connect to the server socket.
To send a test string “This is a test” from the module to the remote node, issue the below command

at+rsi_snd=1,14,0,0,This is a test\r\n
If the remote node (Device A) sends data, the module receives the data and transfers to the Host with a AT+RSI_READ message
Another client (Named “Device B” in this example) can also connect to the Access Point in the module and data transfer can be executed between Device A and Device B through the AP. A maximum of 4 clients are supported.

Associate to an Access Point (With WPA2-PSK Security) as Client

at+rsi_opermode=0,1,256,0\r\n
This command sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_init\r\n
This command initializes the module.

at+rsi_scan=0\r\n
This command scans for Aps and reports the Aps found.

Note:
After a scan, if the user want to join an AP as enterprise client then user need to issue a soft reset first and then follow the flow of commands as in “Associate to an Enterprise Security enabled Access Point as a client”

at+rsi_psk=1,12345678\r\n
This command configures the PSK to be used to associate to the Access Point.

at+rsi_join=Test_AP,0,2,2\r\n
This command associates the module to the AP.
It is assumed that the SSID of the AP is Test_AP with WPA2-PSK security key of 12345678.

at+rsi_ipconf=1,0,0,0\r\n
This configures the IP address of the module in DHCP mode.

at+rsi_dnsserver=1,0,0\r\n
Optional command to provide the IP address of a DNS server.

Note:
If the user wants to use the DNS command to get DNS server IP then the user has to enable tcp_ip_feature_bit_map[8] in the oper_mode command.

at+rsi_dnsget=<domain_name>,1\r\n
Optional command to resolve IP of a given domain name.

at+rsi_ltcp=5001,5,0\r\n
Opens a server TCP socket inside the module with port number 5001 with tos(type of service) type 0 and max clients support count as 5.
Now connect another client (called “Device A” in this example) to the same Access Point and open a client socket to bind to the module’s socket.
To send a test string “This is a test” from the module to the remote node, issue the below command

at+rsi_snd=1,14,0,0,This is a test\r\n
If the remote node (Device A) sends data, the module receives the data and transfers to the Host with a AT+RSI_READ message

Associate to an Access Point (With WEP Security) as Client

at+rsi_opermode=0,2,4,0\r\n
This command sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_init\r\n
This command initializes the module.

at+rsi_scan=0\r\n
This command scans for APs and reports the APs found.

at+rsi_wepkey=0,ABCDE12345,0,0,0\r\n
This command configures the PSK to be used to associate to an Access Point.

at+rsi_join=Test_AP,0,2,3\r\n
This command associates the module to the AP.

at+rsi_ipconf=1,0,0,0\r\n

This configures the IP address of the module in DHCP mode.

at+rsi_ltcp=5001,5,0\r\n
opens a server TCP socket inside the module with port number 5001 with tos(type of service) type 0 and max clients support count as 5.
Now connect another client (called “Device A” in this example) to the same Access Point and open a client socket to bind to the module’s socket.
To send a test string “This is a test” from the module to the remote node, issue the below command

at+rsi_snd=1,14,0,0,This is a test\r\n
If the remote node (Device A) sends data, the module receives the data and transfers to the Host with a AT+RSI_READ message

Associate to a WPS Enabled Access Point

at+rsi_opermode=0,2,4,0\r\n
This command sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_init\r\n
This command initializes the module.

at+rsi_scan=0\r\n
This command scans for available Aps and reports the Aps found.

at+rsi_join=,0,2\r\n
This command associates the module to the AP using WPS push button method. Note that, we have to send null in place of ssid .

at+rsi_ipconf=1,0,0,0\r\n

This command configures the IP address of the module in DHCP mode.

at+rsi_ltcp=5001,5,0\r\n

opens a server TCP socket inside the module with port number 5001 with tos (type of service) type 0 and max clients support count as 5.
Now connect another client (called "Device A" in this example) to the same Access Point and open a client socket to bind to the module's socket.
To send a test string "This is a test" from the module to the remote node, issue the below command

at+rsi_snd=1,14,0,0,This is a test\r\n

If the remote node (Device A) sends data, the module receives the data and transfers to the Host with a AT+RSI_READ message.

Associate to an Enterprise Security Enabled Access Point as Client

The example demonstrates the flow for EAP-TLS mode.

at+rsi_opermode=2,0,4,0\r\n
This sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_init\r\n
This command initializes the module.

at+rsi_eap=TLS,MSCHAPV2,user1,password1\r\n
This command sets the Enterprise mode.

at+rsi_scan=0\r\n
This command scans for Aps and reports the Aps found.

> **Note**
>
> After a scan, if the user want to join an AP with WPA2-AES PSK then user need to issue a soft reset first and then follow the flow of commands as in "Associate to an Access Point (with WPA2-PSK security) as a client"

at+rsi_cert=1,cert_len,key_password,<TLS certificate>\r\n
This command provides the TLS certificate to the module

at+rsi_join=Test_AP,0,2,4\r\n
This command associates the module to the AP. It is assumed that the SSID of the AP is Test_AP.

at+rsi_ipconf=1,0,0,0\r\n
This command configures the IP address of the module in DHCP mode.

at+rsi_ltcp=5001,5,0\r\n
opens a server TCP socket inside the module with port number 5001 with tos(type of service) type 0 and max clients support count as 5.
Now connect another client (called "Device A" in this example) to the same Access Point and open a client socket to bind to the module's socket.
To send a test string "This is a test" from the module to the remote node (Device A), issue the below command

at+rsi_snd=1,14,0,0,This is a test\r\n
If the remote node (Device A) sends data, the module sends the received data with a AT+RSI_READ message to the Host.

Per Mode Command Flow in Burst Mode

at+rsi_opermode=8\r\n
This command sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_feat_frame=0,1,0,0,1,0\r\n
This command sets the correct RF type and path

at+rsi_init\r\n

This command initializes the module.

at+rsi_antenna=0\r\n

This is the command for antenna .

at+rsi_setregion = 1,4\r\n

This command configures the region to worldwide

at+rsi_per=1,127,139,1000,0,6,0,0,0,0\r\n

This command sends the packets in burst mode in channel number 6, with packet length 1000, with max power, with data rate 6Mbps.

at+rsi_per=0\r\n

To stop transmission

Per Mode Command Flow in Continuous Mode

at+rsi_opermode=8\r\n
This command sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_feat_frame=0,1,0,0,1,0\r\n
This command sets the correct RF type and path

at+rsi_init\r\n
This command initializes the module.

at+rsi_antenna=0\r\n

This is the command for antenna .

at+rsi_setregion = 1,4\r\n

This command configures the region to worldwide

at+rsi_per=1,127,139,260,1,6,0,0,0,0\r\n

Before starting Continuous mode, it is required to start Burst mode with power and channel values which is intended to be used in Continuous mode.

at+rsi_per=0\r\n

To stop transmission

at+rsi_per=1,127,0,260,2,6,0,0,0,0\r\n

This command sends the packets in continuous Wave mode in channel number 6, with packet length 260, with max power, with data rate 1Mbps.

Per Stats

at+rsi_per_stats=0,1\r\n

To enable the PER stats.

at+rsi_per_stats=1\r\n

To disable the PER stats.

Per Receive

at+rsi_opermode=8\r\n

This command sets the operating mode of the module.

at+rsi_band=0\r\n

This command sets the operating band of the module.

at+rsi_feat_frame=0,1,0,0,1,0\r\n

This command sets the correct RF type and path

at+rsi_init\r\n

This command initializes the module.

at+rsi_setregion = 1,4\r\n

This command configures the region to worldwide

at+rsi_per_stats=0,1\r\n\

This command here is used to see the WLAN Receive stats.

at+rsi_per_stats=1\r\n\

This command is used to stop the WLAN PER RECEIVE STATS

Enabling BG Scan in Open Mode as Client

at+rsi_opermode=0,1,4,0\r\n
This command sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_init\r\n
This command initializes the module.

at+rsi_scan=0\r\n
This command scans for available APs in the channels mentioned in channel bitmap and reports the APs found.

at+rsi_join=test,0,2,0\r\n
This command associates the module to the AP in open mode.

at+rsi_bgscan=1,1,10,4,10,15,20,1\r\n
This command enables the back ground scan functionality in module.
Here instant bgscan is enabled hence module will send probe requests in the air on to the channels mentioned in the channel bitmap and results will go to host.

at+rsi_ipconf=1,0,0,0\r\n
This command configures the IP address of the module in DHCP mode.

Enabling Roaming in Open Mode as Client

at+rsi_opermode=0,1,4,0\r\n
This command sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_init\r\n
This command initializes the module.

at+rsi_scan=0\r\n
This command scans for available APs in the channels mentioned in channel bitmap and reports the APs found.

at+rsi_join=test,0,2,0\r\n
This command associates the module to the AP in open mode.

at+rsi_bgscan=1,1,10,4,10,15,20,1\r\n
This command enables the back ground scan functionality in module.
Here instant bgscan is enabled hence module will send probe requests in the air and results will go to host and in background scan will be continued based upon the parameters given in the command.

at+rsi_roam_params= 1,5,2\r\n
This command enables roaming in module.

at+rsi_ipconf=1,0,0,0\r\n
This command configures the IP address of the module in DHCP mode.

Enabling WMM PS in Open Mode as Client

at+rsi_opermode=0,1,4,0\r\n
This command sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_init\r\n
This command initializes the module.

at+rsi_scan=6,test\r\n
This command scans for available AP in the given channels with the given SSID and reports.
if the is AP found.

at+rsi_wmm_config=1,0,0,1\r\n
This command enables the WMM feature in the module.

at+rsi_join=test,0,2,0\r\n
This command associates the module to the AP in open mode.

at+rsi_pwmode=1\r\n
This command configures the power save mode 1 in the module.

at+rsi_ipconf=1,0,0,0\r\n
This command configures the IP address of the module in DHCP mode.

Open a Multicast IPv4 Socket in Client Mode

at+rsi_opermode=0,1,4,0\r\n
This command sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_init\r\n
This command initializes the module.

at+rsi_scan=0\r\n
This command scans for APs and reports the APs found.

at+rsi_join=Test_AP,0,2,0\r\n
This command associates the module to the AP.

at+rsi_ipconf=1,0,0,0\r\n
This configures the IP address of the module in DHCP mode.

at+rsi_multicast=1,239.0.0.0\r\n
To join IPv4 multicast group with address 239.0.0.0.

at+rsi_ludp=5001\r\n
To open ludp socket with port number 5001.

at+rsi_multicast=0,239.0.0.0\r\n
To leave multicast group with address 239.0.0.0.

Open a Multicast IPv6 Socket in Client Mode

at+rsi_opermode=0,1,8,0\r\n
This command sets the operating mode of the module.

at+rsi_band=0\r\n
This command sets the operating band of the module.

at+rsi_init\r\n
This command initializes the module.

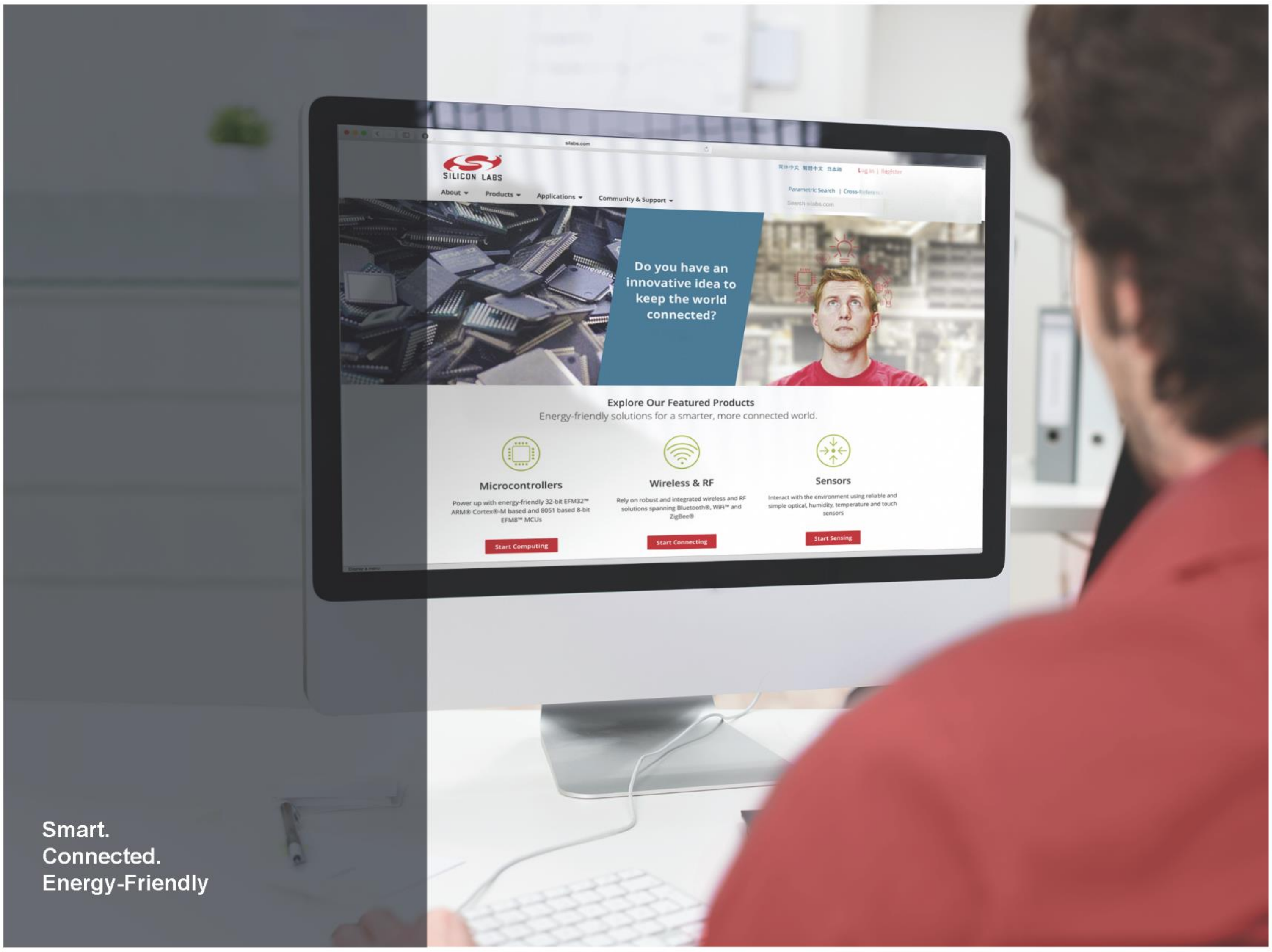
at+rsi_scan=0\r\n
This command scans for APs and reports the APs found.

at+rsi_join=Test_AP,0,2,0\r\n
This command associates the module to the AP.

at+rsi_ipconf=1,0,0,0\r\n
This command configures the IP address of the module in DHCP mode.

at+rsi_ltcp=5001,2,0,1,0\r\n
Opens a SSL server socket inside the module with port number 5001 with tos (type of service) type 0, with maximum 2 SSL clients support and with all SSL ciphers support.
To send a test string “This is a test” from the module to the remote node (Device A), issue the below command

at+rsi_snd=1,14,0,0,This is a test\r\n
If the remote node (Device A) sends data, the module sends the received data with an AT+RSI_READ message to the Host.


Smart.
Connected.
Energy-Friendly
Do you have an innovative idea to keep the world connected?
Explore Our Featured Products
Energy-friendly solutions for a smarter, more connected world.
Microcontrollers
Wireless & RF
Sensors
Start Computing
Start Connecting
Start Sensing

Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701

http://www.silabs.com