RS9113
WiSeConnect™/Connect-io-n™

EVB User Guide

Version 1.10

December 2014

Redpine Signals, Inc.
2107 N. First Street, #680
San Jose, CA 95131.
Tel: (408) 748-3385
Fax: (408) 705-2019
Email: info@redpinesignals.com
Website: www.redpinesignals.com

Redpine Signals, Inc. Proprietary and Confidential.

Table of Contents

1 Introduction 7
2 Hardware Details 8
2.1 Evaluation Board 8
2.1.1 Hardware Description 9
2.1.2 EVB interface pin description 15
2.1.2.1 SPI header pin description 15
2.1.2.2 UART1 header pin description 15
2.1.2.3 GPIO header pin description 16
2.2 Header Pin Orientation 16
2.3 Required Setup 17
2.4 Install virtual COM port drivers 17
3 Bootloading the module 20
3.1 Host Interaction mode 20
3.1.1 Hyper Terminal Configuration for Message Display 20
3.1.2 Auto Baud Rate Detection (ABRD) 20
3.1.3 Startup Operation 20
3.1.4 Messages on EVB power-up 20
3.1.4.1 Loading the Firmware in the Module 21
3.1.4.2 Firmware Upgradation 22
3.2 GPIO Based Bootloader Bypass mode 26
3.2.1 Bypass Mode in UART/USB CDC 27
3.2.1.1 Enable/Disable GPIO based bypass option 27
4 Demo Wi-Fi Applications in UART Mode 30
4.1 Wi-Fi Client Application with Personal Security 30
4.1.1 Demo Application Set-up 30
4.1.2 Wi-Fi Client Configuration 31
4.1.3 Demo Application Test Procedure 32
4.2 Wi-Fi Client Application with Enterprise Security 35
4.2.1 Demo Application Set-up 35
4.2.2 Wi-Fi Client Configuration 35
4.2.3 Demo Application Test Procedure 36
4.2.3.1 EAP-TLS Demo Application 36
4.2.3.2 EAP-TTLS, and PEAP Demo Application 38
4.3 Wi-Fi Direct Client Application 41
4.3.1 Demo Application Setup 41
4.3.2 Wi-Fi Direct Client Configuration 41
4.3.3 Demo Application Test Procedure 45
4.4 Access Point Mode 46
4.4.1 Demo Application Setup 46
4.4.2 Access Point Configuration 46
4.5 Wireless configuration 49
4.5.1 Configuration to join a Specific AP 49
4.5.2 Configuration to create an AP 54
5 Using the module in USB CDC -ACM mode 61
5.1 USB CDC-ACM mode 61
5.1.1 Demo application set up 61

5.1.2 USB CDC-ACM usage 61
6 Evaluation of Bluetooth Classic 63
7 Evaluation of Bluetooth Low Energy 64
8 Evaluation of ZigBee 65
9 Using the module in SPI mode 66
9.1 Sample flow for evaluating SPI mode 67
10 Appendix A: Serial port Host Applications in UART Mode. 70
10.1 Hyper Terminal Configuration 71
10.2 Tera Term Configuration 75
10.3 Serial Port Monitor Configuration 78
11 Appendix B: Radius Server Configuration 81
12 Appendix C: Access Point Configuration 84
12.1 Personal Security AP Settings 84
12.2 Enterprise Security AP Settings 86
13 Appendix D: TCP Command Configuration 87

Table of Figures

Figure 1: RS9113 WiSeConnect/Connect-io-n EVB 8
Figure 2: Header Pin Orientation 17
Figure 3: Selecting USB Serial Port Properties 18
Figure 4: USB Serial Port Properties 19
Figure 5: RS9113 WiSeConnect/Connect-io-n UART Welcome Message 21
Figure 6:RS9113-WiSeConnect Module UART Firmware Loading Message.. 22
Figure 8 :RS9113-WiSeConnect Module UART Firmware Upgrade File Prompt Message 23
Figure 9: RS9113-WiSeConnect Module UART Firmware Upgrade File Selection Message 24
Figure 10: RS9113-WiSeConnect Module UART Firmware Upgrade File transfer Message 25
Figure 11: RS9113-WiSeConnect Module UART Firmware Upgrade Completion Message 26
Figure 15: Enabling the GPIO based bypass mode 28
Figure 16: Disabling the GPIO based bypass mode 29
Figure 17: Demo Application Setup with Personal Security 30
Figure 18: Execution of TCP.exe in Laptop C 34
Figure 19: Demo Application Setup with Enterprise Security 35
Figure 20: Command Flow in Enterprise Security Mode 37
Figure 21: Running of Python Script WiFi_Enterprise_TLS.py 37
Figure 22: Wi-Fi Direct Set-up 41
Figure 23: Android Phone Wi-Fi Direct Settings 41
Figure 24: Command Flow in Wi-Fi Direct Mode 44
Figure 25: Access Point Set-up 46
Figure 26: Module Configured as Access Point 49
Figure 27: Module web page 50
Figure 28: Wireless Configuration page 51
Figure 29: Module Connected to Access Point 52
Figure 30: Wireless Configuration Page 54
Figure 31: Module Configured as Access Point 55
Figure 32: Module web page 56
Figure 33: Wireless Configuration page 57
Figure 34: Module Connected to Access Point 58
Figure 35: Wireless Configuration page 59
Figure 36: USB CDC-AMC set up 61
Figure 37: PC device COM port listening 62
Figure 38: Interface between module and host 66
Figure 43: Flow commands in SPI mode 69
Figure 40: PC Device Manager COM ports listing 70
Figure 41: Hyper Terminal Name field Configuration 71
Figure 42: Hyper Terminal COM port field Configuration 72
Figure 43: Hyper Terminal COM port Properties Configuration 73
Figure 44: Hyper Terminal Properties Selection 73
Figure 45: Hyper Terminal Properties Dialog Box 74
Figure 46: Hyper Terminal ASCII Configuration 75
Figure 47: Tera Term VT 75
Figure 48: Tera Term Serial port Selection 76

Figure 49: Tera Term Serial port Configuration 76
Figure 50: Tera Term Terminal Selection..... 77
Figure 51: Tera Term Terminal Configuration..... 77
Figure 52: Serial Port Monitor 78
Figure 53: Serial Port Monitor Session Type Selection..... 79
Figure 54: Serial Port Monitor Serial Device Selection..... 79
Figure 55: Serial Port Monitor Data Visualizer Configuration 80
Figure 56: Access Point Wireless Settings..... 84
Figure 57: Access Point Settings -Personal Security Mode 85
Figure 58: Access Point Settings - Enterprise Security Mode..... 86

List of Tables

Table 1: UART Host Interface Selection – USB-UART BRIDGE 13
Table 2: UART Host Interface Selection - Header 13
Table 3: USB CDC Host Interface Selection..... 14
Table 4: USB Host Interface Selection..... 14
Table 5: SPI Host Interface Selection..... 14
Table 8: SPI Header Pins..... 15
Table 9: UART Header 15
Table 10: GPIO Header..... 16

1 Introduction

The RS9113 WiSeConnect/Connect-io-n™ is based on Redpine Signals' RS9113 ultra-low-power, single spatial stream, dual-band 802.11n + BT4.0 +Zigbee Convergence SoC. The RS9113 WiSeConnect/Connect-io-n low-cost CMOS integration of a multi-threaded MAC processor, baseband digital signal processing, analog front-end, crystal oscillator, calibration OTP, dual-band RF transceiver, dual-band high-power amplifiers, baluns, diplexers, diversity switch and Quad-SPI Flash thus providing a fully-integrated solution for a range of embedded wireless applications. The M2MCombo™ (RS9113) chips and modules provide WLAN 802.11n, BT4.0 and Zigbee convergence solution for integration into mobile and M2M communication devices. With Redpine's embedded four-threaded processor and on-chip ROM and RAM, these modules enable integration into low-host and zero host load applications. The RS9113 module is mounted on the EVB along with other supporting components including a 40MHz reference oscillator, antenna and other passive components. It can connect to a host processor through UART, SPI or USB.

This document describes usage of the Redpine RS9113 WiSeConnect/Connect-io-n evaluation board. The sequence of commands and set-up requirements are described in relevant sections to quickly evaluate the features and functionality of the EVB. The document should be used in conjunction with the Programming Reference Manual (PRM

Redpine recommends evaluating the EVB in UART mode first before evaluating other modes.

This document covers the below part numbers

1. RS9113-NBZ-S1W-EVB
2. RS9113-NBZ-D1W-EVB
3. RS9113-NBZ-S1C-EVB
4. RS9113-NBZ-D1C-EVB

Note: Wi-Fi Direct™ and Enterprise Security are features specific to WiSeConnect module family and not applicable to Connect-io-n module family. All the grayed out sections, features and commands are currently not supported in firmware.

This EVB user guide applicable for EVB labeled "RS9113 Based Module Reference Design" and "RS9113-B1 Based Module Reference Design Rev 1.0Ver 0.0".

2 Hardware Details

This chapter describes the hardware features of the RS9113 WiSeConnect/Connect-io-n module EVB.

2.1 Evaluation Board

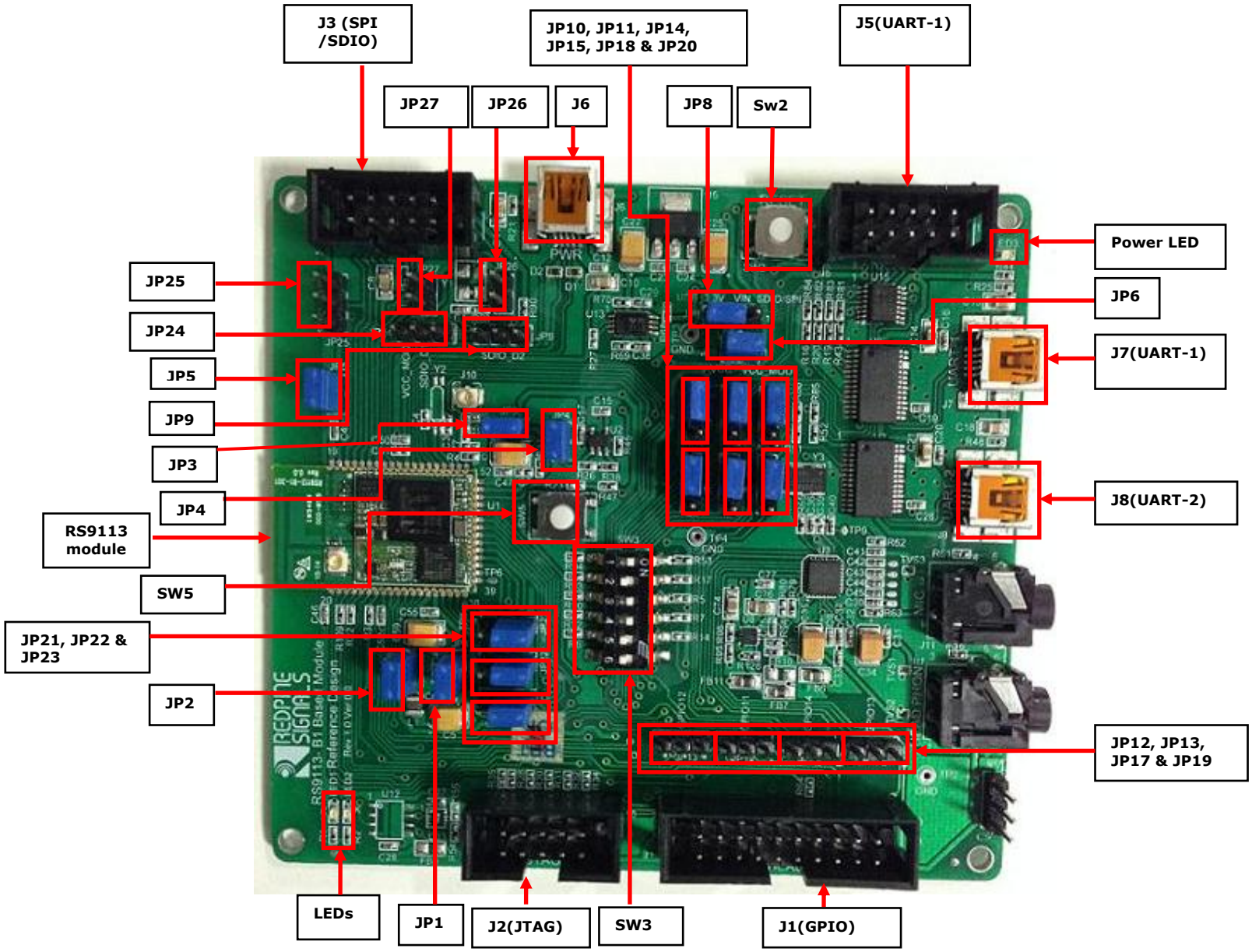

Figure 1: RS9113 WiSeConnect/Connect-io-n EVB

2.1.1 Hardware Description

J1(GPIO):
J1 is a 20-pin GPIO header that provides access to GPIO lines of the module. Refer to RS9113WiSeConnect/Connect-io-n modules datasheet for GPIO pin details.

J2(JTAG):

J2 is a 10-pin header for JTAG programming. This header should not be used.

J3(SPI/SDIO):
J3 is a 10 pin header to provide access to SDIO or SPI interface of the wireless module.

J5(UART-1):
J5 is a 10 pin serial UART (3.3V TTL) connector to provide access to UART-1 interface of the wireless module. The ground pin of this header should be connected to a corresponding ground pin of the MCU.

J6:

J6 is an external power source for the EVB through a mini-USB connector. This interface can also be used as a USB host interface. If the Host MCU is not capable of supplying rated EVB current, then EVB can be powered through J6 connector.

J7(UART-1):
J7 is an USB interface to provide easy access to the UART-1 interface of the wireless module. The connector is interfaced to USB-UART bridge that enable host to perform serial communication over USB interface.

J8(UART-2):
J8 is an USB interface to provide easy access to the UART-2 interface of the wireless module. The connector is interfaced to USB-UART bridge that enable host to capture debug log messages from the module. This provision is currently reserved until supported in the firmware.

Power LED:

Power LED is used to indicate state of power supply to the EVB.

LEDs:

Two LEDs namely LED1 and LED2 are used to indicate Wi-Fi or BT state. These are general purpose LEDs connected to GPIO pins of RS9113 module.

(1) LED1 is connected to GPIO_16 of RS9113 module for Wi-Fi activity.

(2) LED2 is connected to GPIO_19 of RS9113 module for BT activity.

JP1:

Jumper JP1 enables power supply to the RF transceiver. This jumper should always be set.

JP2:

Jumper JP2 facilitates measurement of current consumption of VIN_MOD for the wireless module. This jumper should always be set during normal operations.

JP3:

Jumper JP3 facilitates measurement of current consumption of VBAT for the wireless module. This jumper should always be set during normal operations.

JP5:

This jumper should always be set during operations.

JP6:

Jumper JP6 facilitates measurement of current consumption for the wireless module. This jumper should always be set during normal operations.

JP8:

Jumper JP8 facilitates selection for power supply to the module either through USB or from an external host.

(1) For USB supply, set the jumper to the USB 3.3V position on the EVB by shorting pins 2 and 3 on JP8.

(2) For external MCU supply, set the jumper to the VIN SDIO position on the EVB by shorting pins 1 and 2 on JP8.

JP9:

In UART and USB mode, remove jumper on these pins. In SPI mode, there are two possibilities.

1. If the interrupt on host is configured as active-HIGH, short JP9 pins 1 and 2.
2. If the interrupt on host is configured as active-LOW, short JP9 pins 2 and 3.

JP10:

1. In UART1 mode, set jumper on pins 2 and 3.
2. To use the pin as GPIO_9_Header, set jumper on pins 1 and 2.

JP11:

1. In UART1 mode, set jumper on pins 2 and 3.
2. To access the pin in the GPIO_10_Header, set jumper on pins 1 and 2.

JP12:

1. To use the pin as I2S clock, set jumper on pins 2 and 3.
2. To access the pin in the GPIO_11_Header, set jumper on pins 1 and 2.

JP13:

1. To use the pin as I2S word select, set jumper on pins 2 and 3.
2. To access the pin in the GPIO_12_Header, set jumper on pins 1 and 2.

JP14:

1. In UART1 mode, set jumper on pins 2 and 3.
2. To use the pin as GPIO_17_Header, set jumper on pins 1 and 2.

JP15:

1. In UART1 mode, set jumper on pins 2 and 3.
2. To use the pin as GPIO_18_Header, set jumper on pins 1 and 2.

JP17:

1. To use the pin as I2S data input, set jumper on pins 2 and 3.
2. To access the pin in the GPIO_13 header or to use the pin in UART2 mode set jumper on pins 1 and 2.

JP18:

1. In UART2 mode, set jumper on pins 2 and 3.
2. To use the pin as GPIO_13_Header, set jumper on pins 1 and 2.

JP19:

1. To use the pin as I2S data output, set jumper on pins 2 and 3.
2. To access the pin in the GPIO_14 header or to use the pin in UART-2 mode set jumper on pins 1 and 2.

JP20:

1. In UART2 mode, set jumper on pins 2 and 3.
2. To use the pin as GPIO_14_Header, set jumper on pins 1 and 2.

JP21, JP22 and JP23:

1. In USB mode, set jumper on pins 1 & 2
2. In other mode, set jumper on pins 2 & 3

JP24:

1. In USB mode, for internal clock set jumper on pins 2 & 3
2. In USB mode, for external clock set jumper on pins 1 & 2
3. Remove the jumper in other modes.

JP25:

1. In USB mode for 12MHz clock, set jumper on pins 1 & 2
2. In USB mode for 24MHz clock, set jumper on pins 2 & 3
3. Remove the jumper in other modes.

JP26 and JP27:

1. In USB mode, set jumper on JP26 & JP27
2. In other modes, no set jumper on JP26 & JP27

SW2:

SW2 is a reset push button to provide reset to the RS9113 module. If RESET is always driven by the host, the push button reset is not needed on this pin in which case, ensure that jumper JP4 is not set.

SW5:

SW5 is a general purpose GPIO push button. The functionality for this button is reserved for now until supported in the firmware.

SW3:

This is a DIP type switch used to configure EVB Host interface mode. The following tables described SW3 switch selection logic for configuring EVB in different Host interface modes.

a) UART mode:

In UART mode, the switch can be configured to use the USB-UART bridge or standard 10-pin UART header.

The following table describes, SW3 configuration to select USB-UART bridge. This configuration is useful to connect a PC host to communicate with RS9113 module.

Switch number	Configuration
1	ON
2	OFF
3	ON
4	ON
5	OFF
6	OFF

Table 1: UART Host Interface Selection – USB-UART BRIDGE

The following table describes, SW3 configuration to select 10 pin UART header. This configuration can be used for an MCU host to communicate with RS9113 module.

Switch number	Configuration
1	OFF
2	OFF
3	ON
4	ON
5	OFF
6	OFF

Table 2: UART Host Interface Selection - Header

b) USB CDC mode:

Switch number	Configuration
1	OFF/ON
2	OFF
3	OFF
4	ON
5	OFF
6	OFF

Table 3: USB CDC Host Interface Selection

c) USB mode:

Switch number	Configuration
1	OFF/ON
2	OFF
3	OFF
4	ON
5	ON
6	OFF

Table 4: USB Host Interface Selection

d) SPI mode:

Switch number	Configuration
1	OFF/ON
2	OFF
3	ON
4	OFF
5	OFF
6	OFF

Table 5: SPI Host Interface Selection

2.1.2 EVB interface pin description

The following tables describe interface pin details.

2.1.2.1 SPI header pin description

The following table describes the pins of SPI header. The same header is shared with SDIO interface.

Pin Number	Pin Name	Direction	Description
1	NC	-	This pin must be left open.
2	SPI_CS	Input	SPI slave select from host(active low)
3	GND	-	Ground
4	VDD	-	Supply voltage.
5	SPI_CLK	Input	Serial clock in from the host. The clock can be up to 80 MHz
6	GND	-	Ground
7	SPI_MOSI	Input	SPI data input
8	SPI_MISO	Output	SPI data output
9	SPI_INTR	Output	Active high, level triggered interrupt, used in SPI mode. The interrupt is raised by the EVB to indicate there is data to be read by the Host
10	RESET	Input	Reset to RS9113 module.

Table 6: SPI Header Pins

2.1.2.2 UART1 header pin description

.

Pin number	Pin name	Direction	Description
1	VCC	-	Supply voltage.
2	VCC	-	Supply voltage.
3	VCC	-	Supply voltage.
4	UART-RX1	Input	UART Receive pin of the EVB
5	UART-TX1	Output	UART Transmit pin of the EVB
6	UART-CTS1	Input	Should be connected in case of hardware flow control. If not used, leave it open.
7	UART-RTS1	Output	Should be connected in case of hardware flow control. If not used, leave it open.
8	GND	-	Ground
9	GND	-	Ground
10	GND	-	Ground

Table 7: UART Header

2.1.2.3 GPIO header pin description

This header interface is reserved until supported in the firmware

Pin number	Pin name	Direction	Description
1	GND	-	Ground
2	VCC	-	Supply voltage
3	GPIO_21	Input	Reserved
4	AUX_ADC_IN0	Input	Reserved
5	GPIO_9_Header	Input/Output	Reserved
6	GPIO_10_Header	Input/Output	Reserved
7	GPIO_15	Input/Output	Reserved
8	GPIO_18_Header	Input/Output	Reserved
9	GPIO_17_Header	Input/Output	Reserved
10	GPIO_14_Header	Input/Output	Reserved
11	GPIO_13_Header	Input/Output	Reserved
12	NC	-	This pin must be left open.
13	ULP_GPIO_10	Input/Output	Reserved
14	GPIO_19/LED1/BT_ACTIVE	Input/Output	Reserved
15	GPIO_11_Header	Input/Output	Reserved
16	GPIO_2	Input/Output	Reserved
17	GPIO_12_Header	Input/Output	Reserved
18	NC	-	This pin must be left open.
19	GPIO_8/I2C-SDA	Input/Output	Reserved
20	GPIO_7/I2C-SCL	Input/Output	Reserved

Table 8: GPIO Header

2.2 Header Pin Orientation

The figure below describes the orientation of headers for reference.

SPI/SDIO Header - J3

UART Header - J5

GPIO Header - J1

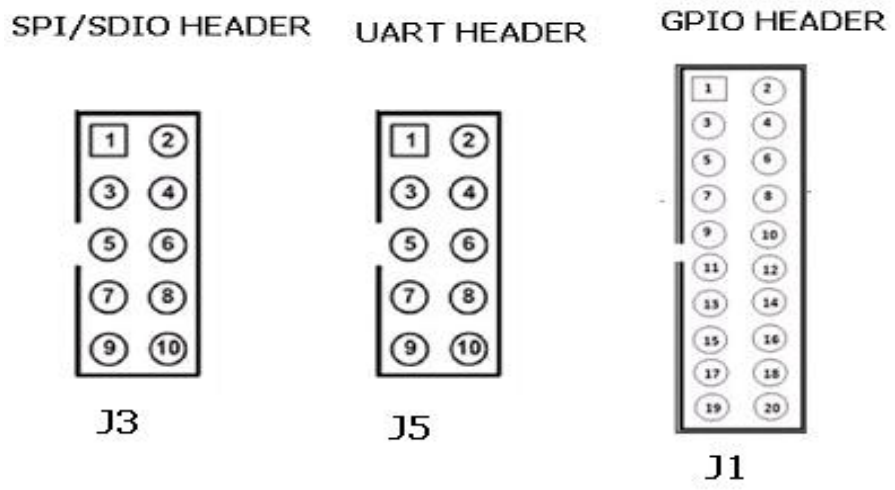

Figure 2: Header Pin Orientation

2.3 Required Setup

The following additional components (EVB not included) are required to evaluate Wi-Fi and BT features of EVB.

1. Windows XP/7 PC with USB interfaces
2. Mini USB cable to power RS9113-WiSeConnect/Connect-io-n EVB, the same cable can be used to execute UART commands from PC.
3. HyperTerminal or an equivalent terminal application with Kermit protocol support for module firmware upgrade and serial programming.
4. FT232R virtual com port drivers from FTDI http://www.ftdichip.com/Drivers/VCP.htm.
5. An 802.11a, 11b, 11g, or 11n Access Point with administrative access to change its settings
6. A RADIUS server for operation in enterprise security mode (refer section Wi-Fi Client Application with Enterprise Security for details)
7. A Wi-Fi Direct™ compatible mobile phone or laptop (refer section Wi-Fi Direct Client Application).

2.4 Install virtual COM port drivers

For UART communication, an FT232R USB-UART bridge is used on the EVB. This will enable a PC host to interface with the device using a USB cable.

Before working with the board, configure the EVB for UART mode using SW3 as described in Hardware Description. Then install virtual COM port drivers on the PC host. Detailed installation instructions are available for download from FTDI. The following link directs to the installation guides specific to PC operating systems.

http://www.ftdichip.com/Support/Documents/InstallGuides.htm

After driver installation is done as per the guide, ensure the FT232R chip is configured to match with the default baud settings of the EVB. Baud rate setting and serial port settings can be configure from device manager as shown in the following screen-shots.

Navigate to the Device manager-> USB Serial Port(COMx) -> right-click and select "Properties.".

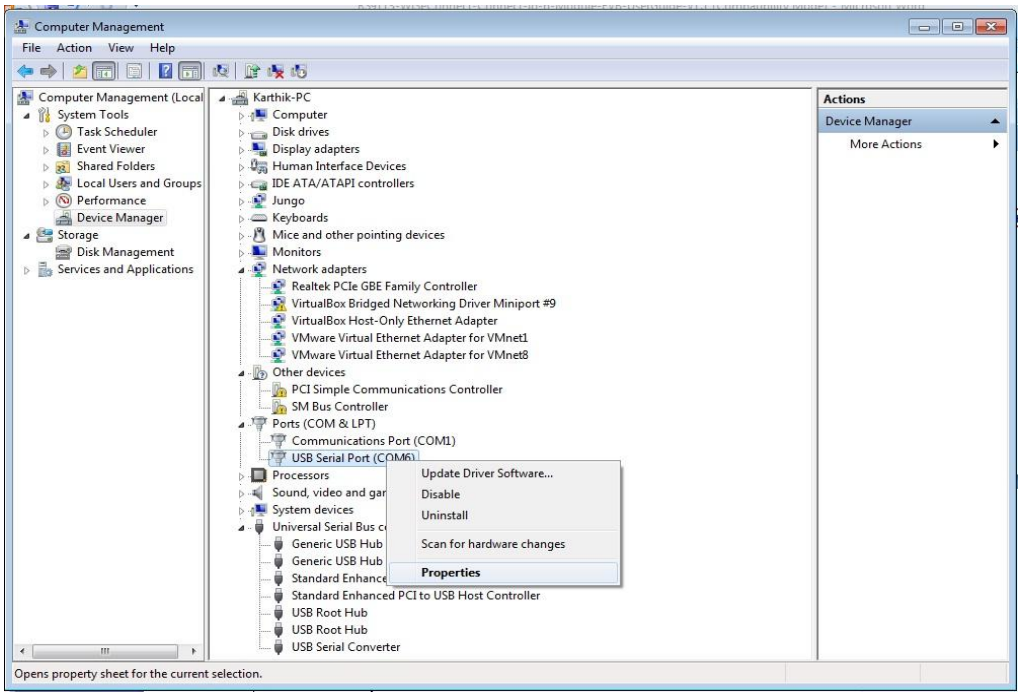

Figure 3: Selecting USB Serial Port Properties

In the "USB Serial Port (COMx)" window, select "Port Settings" tab as shown in the screen-shot below.

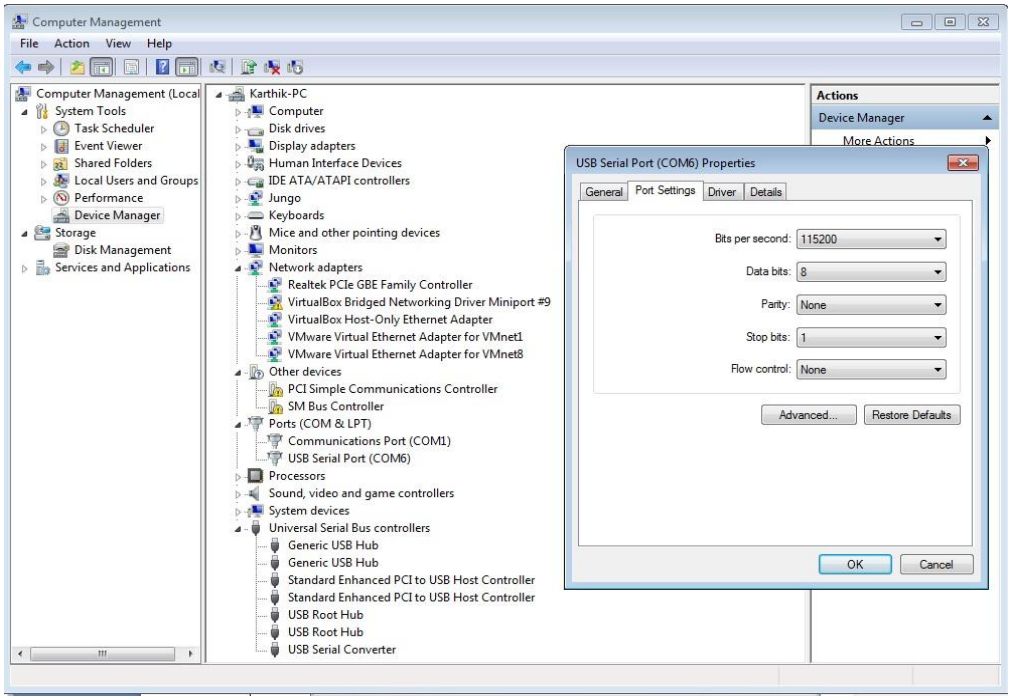

Figure 4: USB Serial Port Properties

Configure the "Port Settings" as mentioned below.

Baud rate: 115200 baud

Data bits: 8 bits

Parity: None

Stop bits: 1

Flow Control: None

3 Bootloading the module

Module supports two Bootloading modes:

- Host interaction (Non-bypass) mode: In this mode host can interact with the bootloader and can give bootup options (commands) to configure different bootup operations. It tells it, as to what operations it has to perform based upon the selections made by the user.
- Bypass mode: In this mode bootloader interactions are completely bypassed and default firmware image is loaded in the module.

3.1 Host Interaction mode

In UART & USB-CDC host interaction mode, bootup options are menu based as explained in below section.

3.1.1 Hyper Terminal Configuration for Message Display

Described in Section:7.1

3.1.2 Auto Baud Rate Detection (ABRD)

Refer to **"ABRD"** section of RS9113 WiSeConnect/Connect-io-n modules Programming Reference Manual(PRM) document for auto baud rate selection.

3.1.3 Startup Operation

On powering up, host is required to carry out ABRD operation and after successful ABRD, module will give boot up options to upgrade or load WLAN/BT firmware or select default image or select bootload bypass mode. Host need to select the appropriate option.

3.1.4 Messages on EVB power-up

To monitor the messages coming from the module HyperTerminal is used as host application. For information on HyperTerminal configuration, refer section **"Hyper Terminal Configuration"** of **RS9113** WiSeConnect/Connect-io-n modules Programming Reference Manual (PRM) document.

After the EVB is powered up, Host and the module exchange data at the baud rate configured in the Host. If the handshake is not completed within 18 seconds then module configures default baud rate to 115200 bps. The welcome message is displayed after completion of handshake. Refer to **"ABRD"** section of RS9113 WiSeConnect/Connect-io-n modules PRM document for auto baud rate selection. Refer figure 5 below for welcome message.

NOTE: EVB firmware version 1.0.10 & above Image-I supports WLAN , BT&ZIGBEE. Image-II is RESERVED for future use.

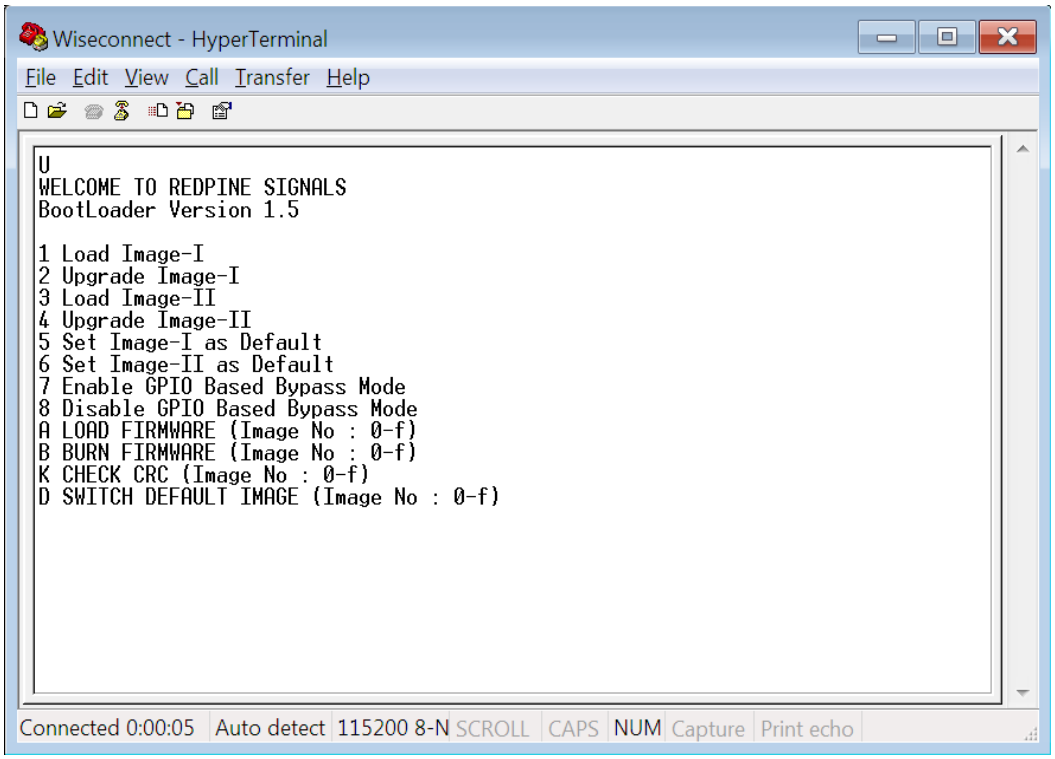

Figure 5: RS9113 WiSeConnect/Connect-io-n UART Welcome Message

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

3.1.4.1 Loading the Firmware in the Module

For loading the firmware from flash of module you have to choose Option 1 i.e. "Load Image–I"

3.1.4.1.1 Load Image - I

1. After welcome message is displayed as shown in figure 5, select option 1 "Load Image–I" for loading WLAN firmware.

   NOTE: If no option is selected after welcome message within 20 seconds then module loads default firmware image.

2. After successfully loading the default firmware, "Loading… Loading Done" message is displayed as in figure 6.

3. After WLAN firmware loading is completed, module is ready to accept commands

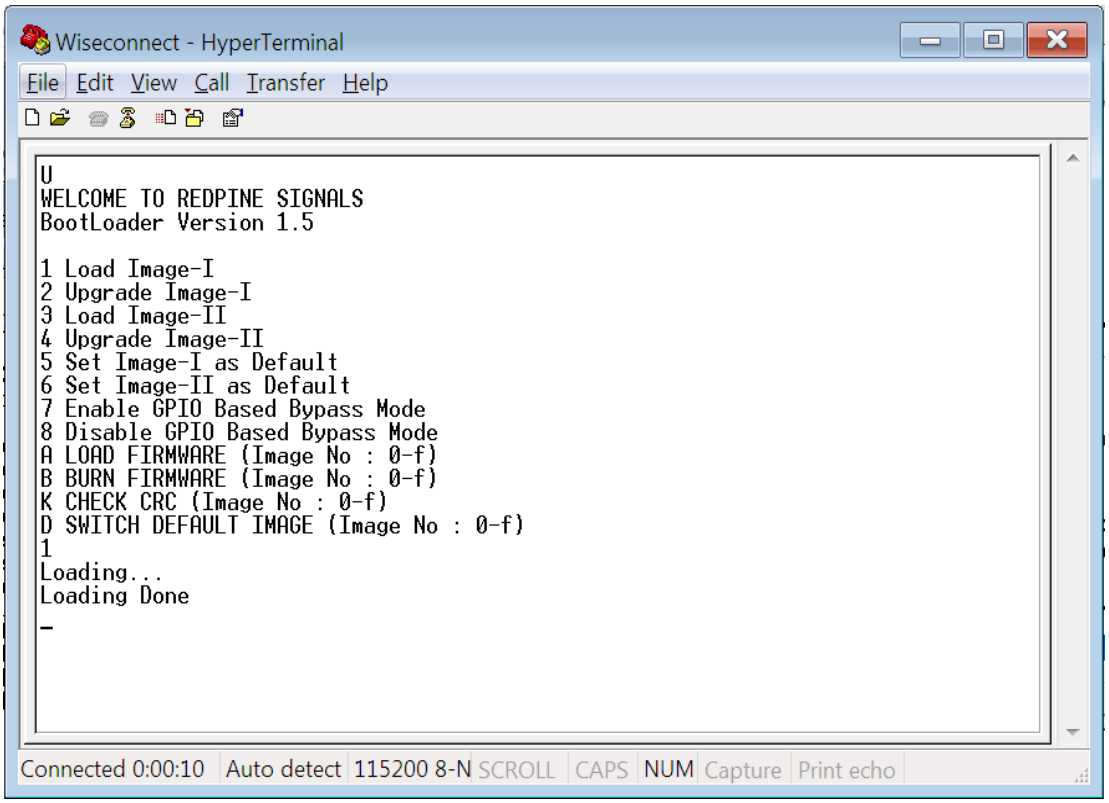

Figure 6:RS9113-WiSeConnect Module UART Firmware Loading Message

3.1.4.2 Firmware Upgradation

On powering up the module, welcome message is displayed as shown in figure 5. For the Firmware upgradation option 2 need to be selected for the WLAN or BT/ZigBee firmware respectively.

3.1.4.2.1 Upgrade WLAN Firmware

1. After welcome message display as shown in figure 5, select option 2 "Upgrade Image- I" for upgrading WLAN firmware.
2. A message "Send RS9113_WC_GENR_x_x_x_wlan.rps" as shown in figure 8 below is displayed.

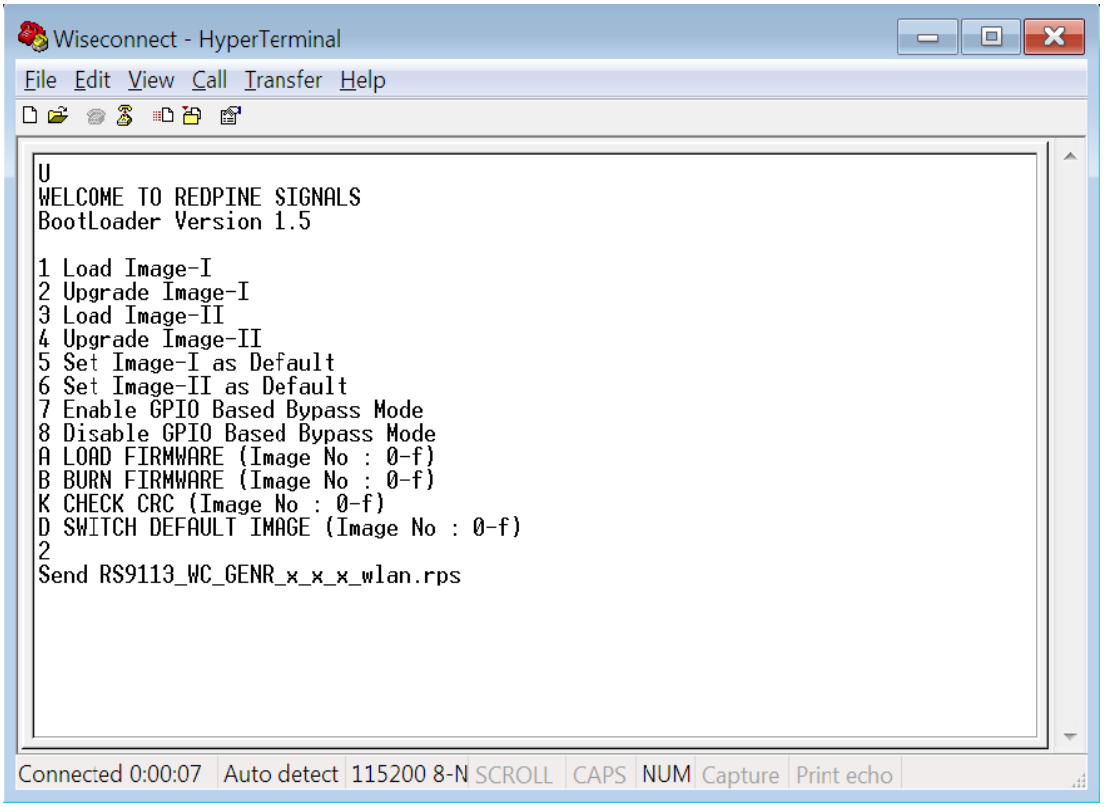

Figure 7 :RS9113-WiSeConnect Module UART Firmware Upgrade File Prompt Message

3. From the "File" menu of HyperTerminal select "send file" option. A dialog box is displayed for file selection. Browse the path where "RS9113_WC_GENR_x_x_x_wlan.rps" is located as shown in the below screen shot and select Protocol type as Kermit. Click "Send" button to load file from the Host into EVB flash memory.

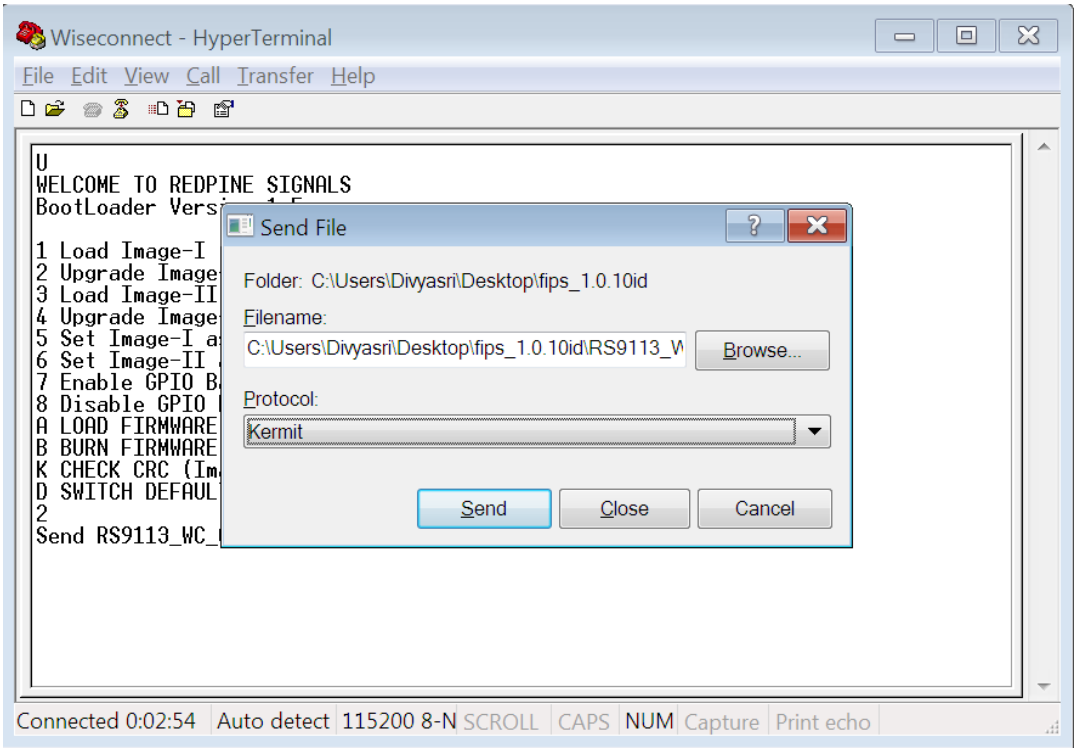

Figure 8: RS9113-WiSeConnect Module UART Firmware Upgrade File Selection Message

4. The dialog box message is displayed while file transfer is in progress as shown in the figure 10 below.

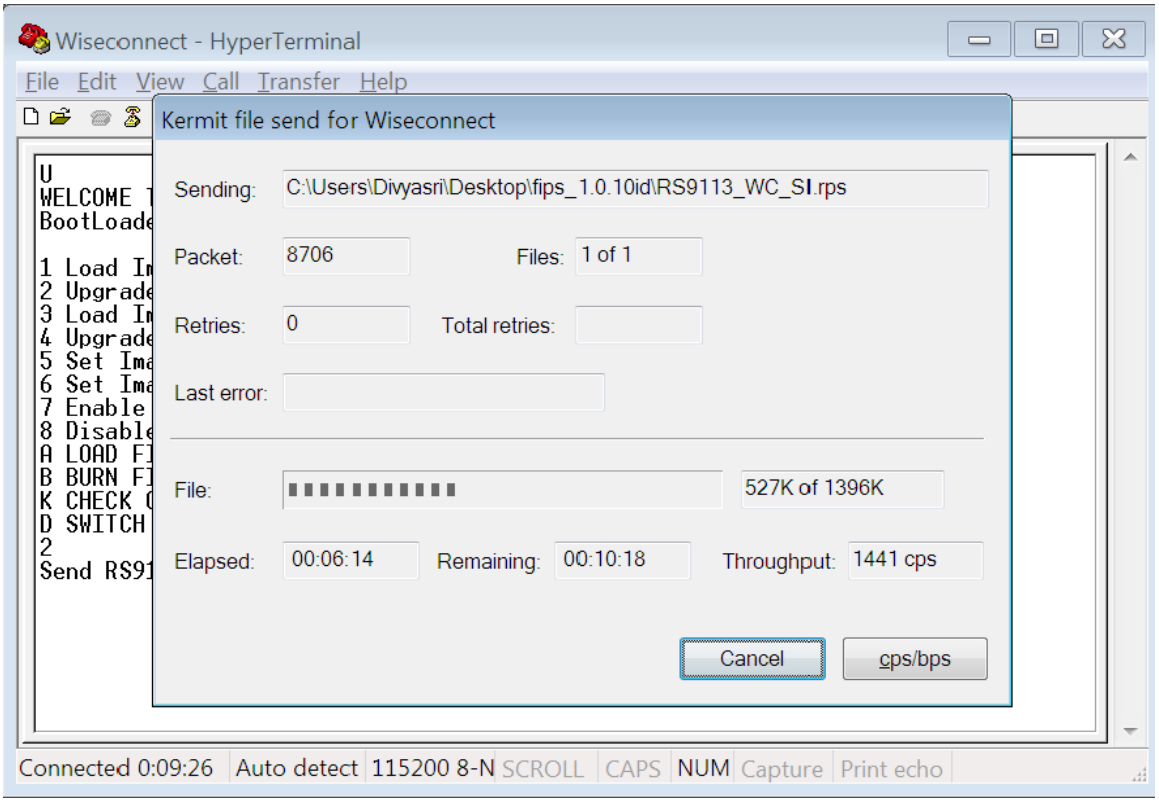

Figure 9: RS9113-WiSeConnect Module UART Firmware Upgrade File transfer Message

5. After successfully completing the file transfer, module sends "Upgradation Successful" message on terminal as shown in the figure 11 below.

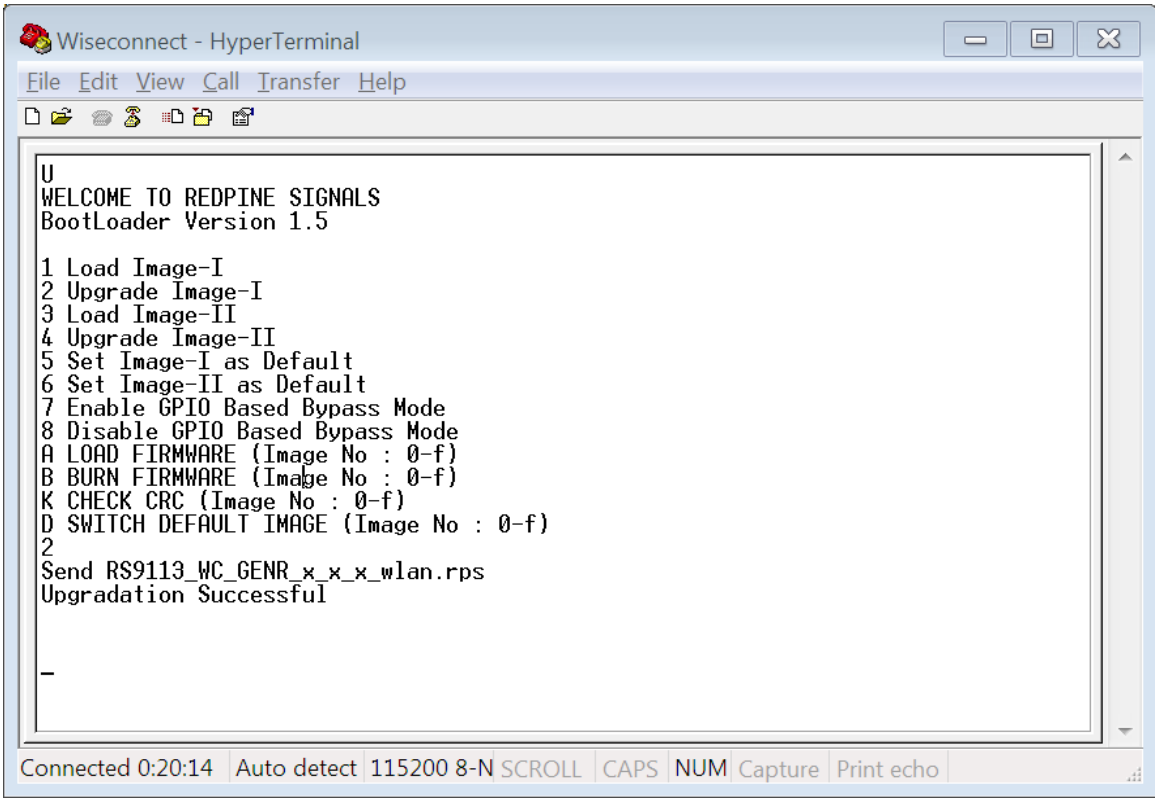

Figure 10: RS9113-WiSeConnect Module UART Firmware Upgrade Completion Message

6. Till this step WLAN firmware was successfully upgraded in the flash memory of the module.
7. Re-boot the module again.
8. Follow the steps mentioned in section Load Image - I to load the firmware from flash, select Option 1 as mentioned in figure 6.
9. The module is ready to accept commands from the Host.

3.2 GPIO Based Bootloader Bypass mode

In GPIO based bootloader bypass mode host interactions with bootloader can be bypassed. There are steps to enable GPIO based Bootloader bypass mode as below:

- Host need to select default image to load in bypass mode.
- Enable Bootloader bypass mode.
- Assert GPIO #15 to Bypass Bootloader on powerup.

NOTE: 1. After selection of an option for example default image selection or Enable Bootload bypass module is required to reboot before issuing next Bootloader option.

2. Out of Box WLAN firmware is selected as default image.

To enable Bootloader Bypass mode host first has to give default image that has to be loaded in bypass mode and then host is expecting to reboot the board again to make the selection of bypass mode (enable).After selecting bypass mode enable reboot the board again. After rebooting the module, it goes to bypass mode and directly loads the default firmware image.

3.2.1 Bypass Mode in UART/USB CDC

Host would select the default firmware image to be loaded when none of the options selected within 20 seconds in host interaction mode or when bootloader bypass mode is enabled.

3.2.1.1 Enable/Disable GPIO based bypass option

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

3.2.1.1.1 Enabling the GPIO based bypass mode

1. If you select option 7, displayed the “Configuration Saved” in figure 15 below.
2. Now GPIO based Bootloader bypass gets enabled.

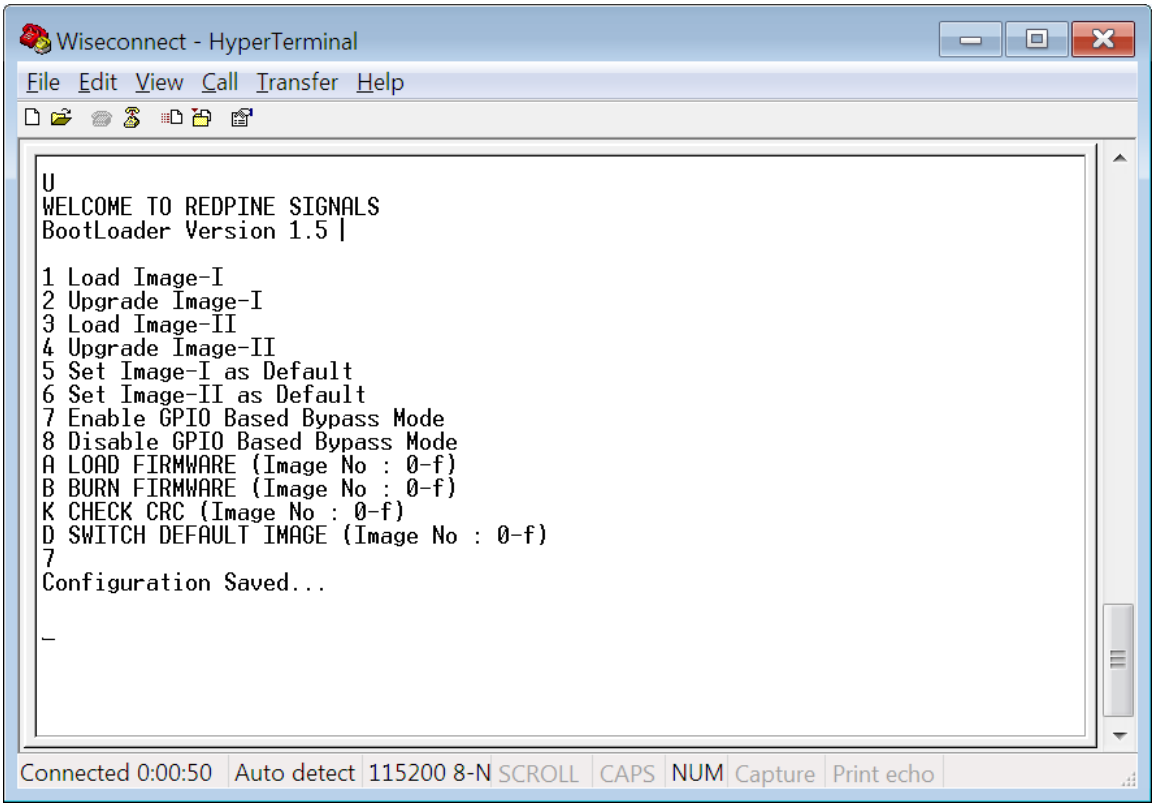

Figure 11: Enabling the GPIO based bypass mode

3. Host is expected to reboot the device after selecting above options
4. Once enabled, from next bootup, Bootloader will latch the value of GPIO #15. If asserted, it will bypass whole bootloading process and load default firmware image selected.

3.2.1.1.2 Disabling the GPIO based bypass mode

1. If you select option 8, displayed the "Configuration Saved" in figure 16 below.
2. Now GPIO based Bootload bypass gets disabled.

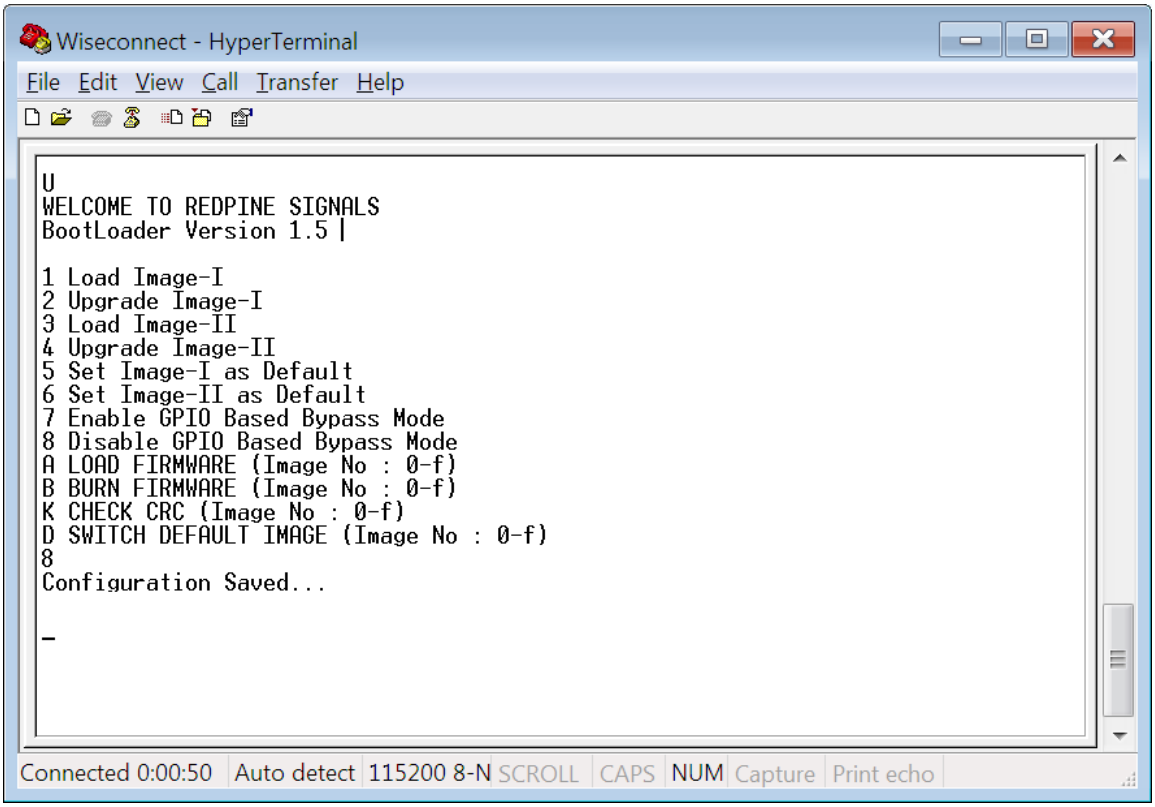

Figure 12: Disabling the GPIO based bypass mode

3. Host is expected to reboot the device after selecting above options

NOTE: GPIO #15 need to deasserted on powerup to move to host interaction mode, To select bootup options like disable Bypass mode or to change default image etc.

4 Demo Wi-Fi Applications in UART Mode

The following sub-sections describe demo Wi-Fi applications in different modes.

4.1 Wi-Fi Client Application with Personal Security

4.1.1 Demo Application Set-up

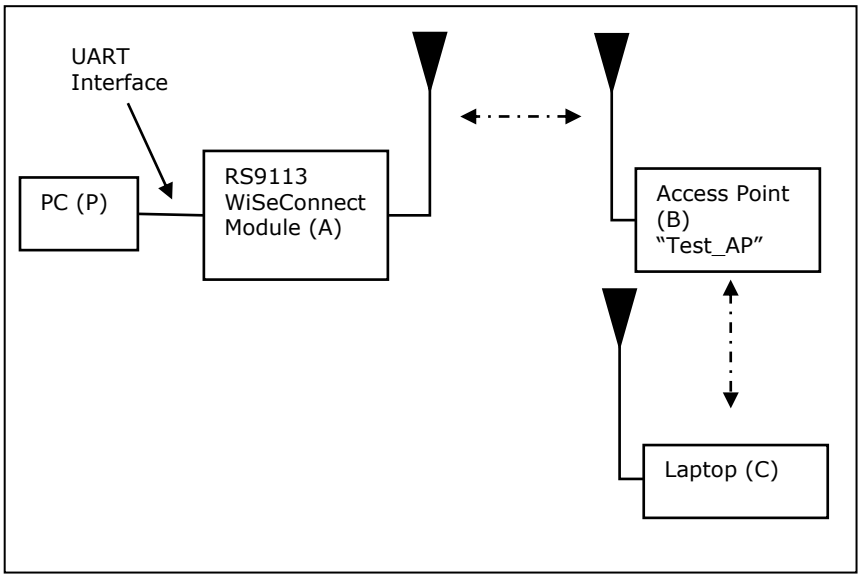

Figure 13: Demo Application Setup with Personal Security

In the set-up as shown in the figure 19, RS9113 WiSeConnect/Connect-io-n acts as Wi-Fi client. It connects to the Access Point configured in WPA2-PSK security mode. The Access point is configured with SSID set as "Test_AP" and IP address set as 192.168.50.1.The SSID and IP address are for illustration only. The user is required to configure the AP based on network domain in which wireless devices operate.

Refer section 11.1 Personal Security AP Settings.

4.1.2 Wi-Fi Client Configuration

1. Configure the EVB in UART mode and connect a USB between UART interface on EVB and the PC host.
2. Open the Serial Port Monitor to observe the actual data exchanged between the module and Host. Open HyperTerminal or Tera Term with the settings described in section Serial port Host Applications in UART Mode.
3. Power up the EVB through USB connector.
4. The terminal application displays the message "WELCOME TO REDPINE SIGNALS" with boot options as described in section 3.1 (Figure 5). Select option 1 to load WLAN firmware. On successful boot-up of the module, "Loading done" message is sent from module to the Host. The following commands can now be issued. Please refer to the RS9113WiSeConnect/Connect-io-n modules PRM document for detailed description of the commands and their responses. A command should not be sent until the response of the previous command is received.

   Note: End of every AT command there should be a carriage return<CR> and a line feed<LF>.

   a. at+rsi_opermode=0,1,4,0

   This command configures the module as Wi-Fi client. The module responds with "OK"

   b. at+rsi_band=0

   This command configures the operating band of Wi-Fi client to 2.4GHz. The module responds with "OK"

   c. at+rsi_init

   This command initializes the Wi-Fi module in the EVB. The module responds with OK<MAC_Address>.

   d. at+rsi_fwversion?

   This command displays current firmware version in use.

   e. at+rsi_scan=0

   This command scans for available Access Points operating in 2.4 GHz band. The module responds with information of the APs scanned.

   f. at+rsi_psk=1,12345678

   This command configures the PSK of the Wi-Fi client with the key entered by user. The User can enter any other key as PSK.

   g. at+rsi_join=Test_AP,0,2

This command connects the Wi-Fi client to the AP with SSID "Test_AP". On successful association, the module responds with OK.

h. at+rsi_ipconf=0,192.168.50.10,255.255.255.0,192.168.50.1

The ip address configured in the above command is for illustration. The user has to configure the ip address as per the network domain the wireless device operates.

This command configures the module's IP to 192.168.50.10 in static mode. Make sure the desired IP is in the same subnet as the Access Point. The module responds to this command by sending the configured IP address to the Host. In the terminal, this response might appear as unreadable characters because of ASCII conversion. The Serial Port Monitor can however be used to see the exact bytes.

To get the IP address allotted by DHCP server in AP, use the following command

at+rsi_ipconf=1, 0, 0, 0

4.1.3 Demo Application Test Procedure

The IP addresses configured in the demo application are meant for illustration only. It is assumed that Wi-Fi client is assigned IP address of 192.168.50.10 and Laptop C connected to AP is assigned IP address of 192.168.50.20.

The application sends and receives TCP packets from client to AP and vice versa.

TCP and UDP applications are provided along with release package for execution on the Laptop. This application is located in the path \RS9113.NBZ.WC.GEN.OSI..x.x.x.x\utils\peer_applications\Windows where RS9113.WSC.GENR.x.x.x.x.x.x is software package directory.

1. Open TCP server socket with the following command on client side.

   at+rsi_ltcp=5001,1,0

   The module responds with

   OK<ip_version ><socket_type> <socket_handle><Lport> <module_ipv4addr>\r\n

   The *socket_handle* parameter will be used in the subsequent sections to send data.

2. Open TCP client socket in the Laptop C by executing the following command.

Make sure that any firewalls, that might prevent opening of sockets, are disabled.

TCP.exe c 2001 192.168.50.10 5001

After executing the command, a command line window is opened with 3 options at laptop C

1 (Send)

2 (Receive)

3 (Exit)

At module side once the socket connection is established below message is displayed

AT+RSI_LTCP_CONNECT=<ip_version><socket descriptor>,<dst_port_no>,<dst_ipv4/ipv6_addr><mss>

Data Receive Application:

a. Type 1 in the command window in Laptop C to send data to the Wi-Fi module.

b. On being prompted "Enter the string to be Transmitted", type any string from the Keyboard ("test" in this example). On pressing "Enter" key on the keyboard, the data is sent from the Laptop C to the module.

c. To read the data from module that was sent by AP, execute the command **AT+RSI_READ** on the client side.

Data Transmit Application:

a. Type 2 in the command window in Laptop C to send data from the Wi-Fi module.

b. Execute the following command on client side

   at+rsi_snd=1,10,0,0,abcdefghij

   The first parameter in the command is the *socket_handle*. It is 1 in this case. It is returned as the response of the command *at+rsi_ltcp* and can be observed in the Serial Port Monitor. Refer to the Programming Reference Manual for more details. The data sent will be displayed in the Laptop C.

Note: Refer section 12 Appendix D: TCP Command Configuration.

The figure shown below displays packets received by the module and packets received from the module.

```
E:\visual studio 2010\Projects\TCP\Release>TCP.exe c 2001 192.168.50.10 5001
Connected To The Server
Enter Option
1- Send, 2- Receive, 3- Exit
1
Enter The String To Be Transmitted:
test
Enter Option
1- Send, 2- Receive, 3- Exit
2
 Waiting For Data To Be Sent From The Module
Server Data: abcdefghij
Complete
Enter Option
1- Send, 2- Receive, 3- Exit
```

Figure 14: Execution of TCP.exe in Laptop C

4.2 Wi-Fi Client Application with Enterprise Security

4.2.1 Demo Application Set-up

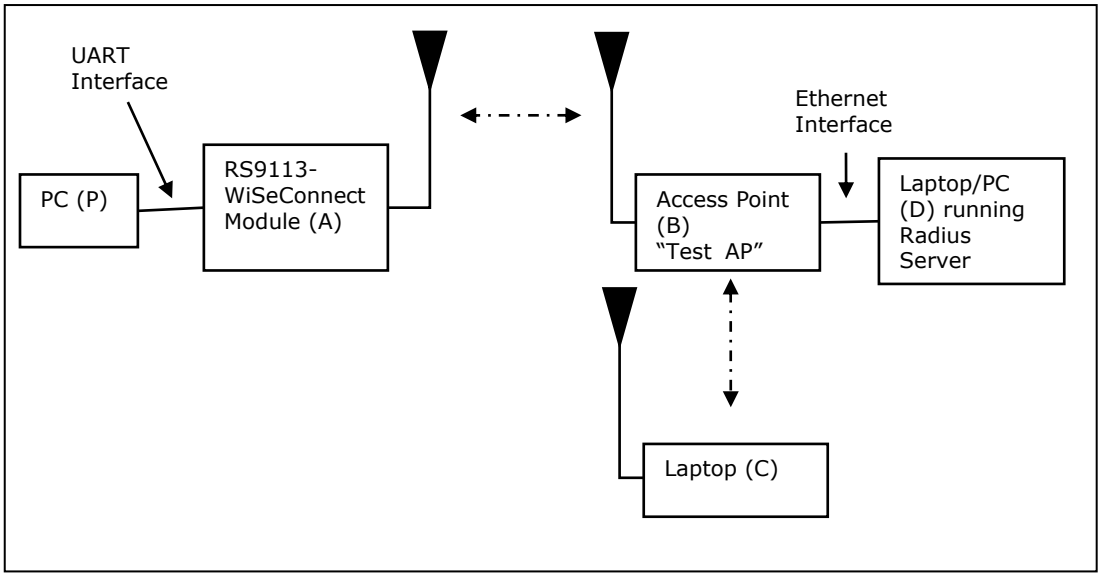

Figure 15: Demo Application Setup with Enterprise Security

In the set-up as shown in the figure 21, RS9113-WiSeConnect acts as Wi-Fi client. It connects to an Enterprise enabled Access Point. RS9113-WiSeConnect supports four Enterprise Security modes:

1. EAP-TLS
2. EAP_TTLS
3. EAP –PEAP
4. EAP- FAST

Refer section 11.2 Enterprise Security AP Settings.

4.2.2 Wi-Fi Client Configuration

1 Configure the EVB in UART mode and connect a USB between the PC and the EVB.

2 Open the Serial Port Monitor to observe the actual data exchanged between the module and Host. Open HyperTerminal or Tera Term with the settings described in section Serial port Host Applications in UART Mode.

3 Power up the EVB through USB connector.

4 The terminal application displays the message "WELCOME TO REDPINE SIGNALS" with boot options as described in section 3.1 (Figure 5). Select option 1 to load WLAN firmware. On successful boot-up of the module, "Loading done" message is sent from module to the Host.

4.2.3 Demo Application Test Procedure

4.2.3.1 EAP-TLS Demo Application

To verify the EAP-TLS mode, a security certificate file should be loaded into the module. A Python based flow is provided to verify this mode as loading of certificate files cannot be done through Hyper-terminal or Tera Term.

1. Enable WPA2-Enterprise in the Access Point settings and start the Radius Server in the Laptop (D) connected to the AP.
2. Install Python on PC(P). The "pyserial" package should be included in the installation to access the serial port.
3. Configure the EVB in UART mode, connect a USB between the PC and the EVB and power up the EVB.
4. Make sure the following things are configured accordingly in the Python script

   \RS9113.NBZ.WC.GEN.OSI.1.0.10k\utils\Python\load_certificate.py

   a. Path of the certificate file [

      \9113_FW\RS9113.NBZ.WC.GEN.OSI.1.0.10k\utils\Radius_Server \raddb\certs\wifiuser.pem

   b. SSID of the Access Point at *rsi_ssid*. It should reflect the SSID of the AP you want to connect to.

   c. Values for parameters *user_identity* and *security_key* should match the values in the file *<radius server path>\raddb\users*

   d. Parameter *oper_mode* should be '2'

   e. COM port name (called *param1* in the script) should match the name of the COM port name in PC.

   f. *rsi_band* should be configured according to the settings of the AP.
5. Open Windows Command Line in the PC (P) and run the script *WiFi_Enterprise_TLS.py*. Power cycle or reset the module before every fresh run of the script.

The script sends AT commands to the module from the PC and makes the module connect to the Access Point. The sequence of commands executed in the script is shown below.

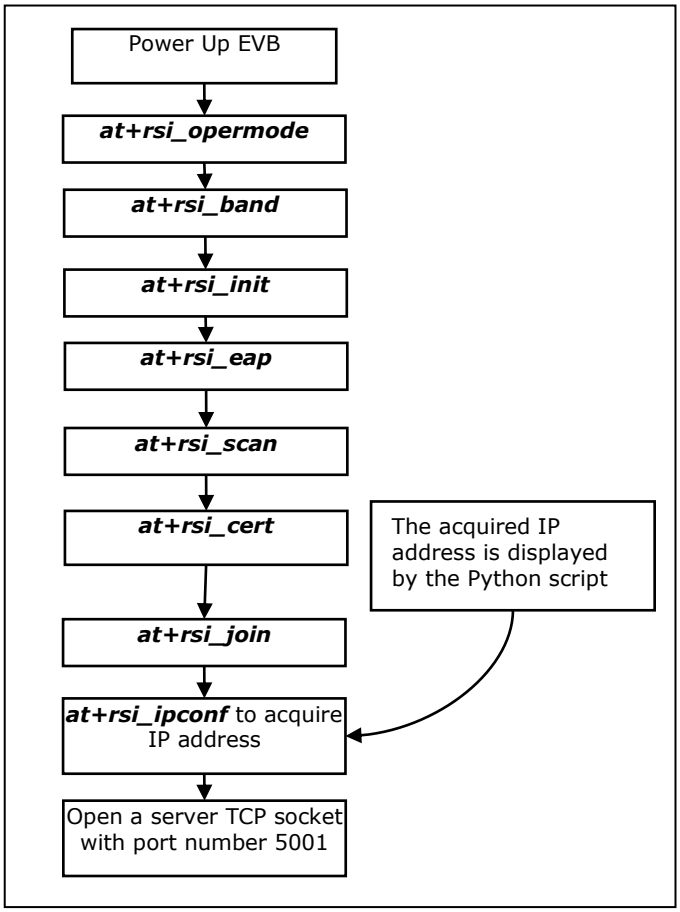

Figure 16: Command Flow in Enterprise Security Mode

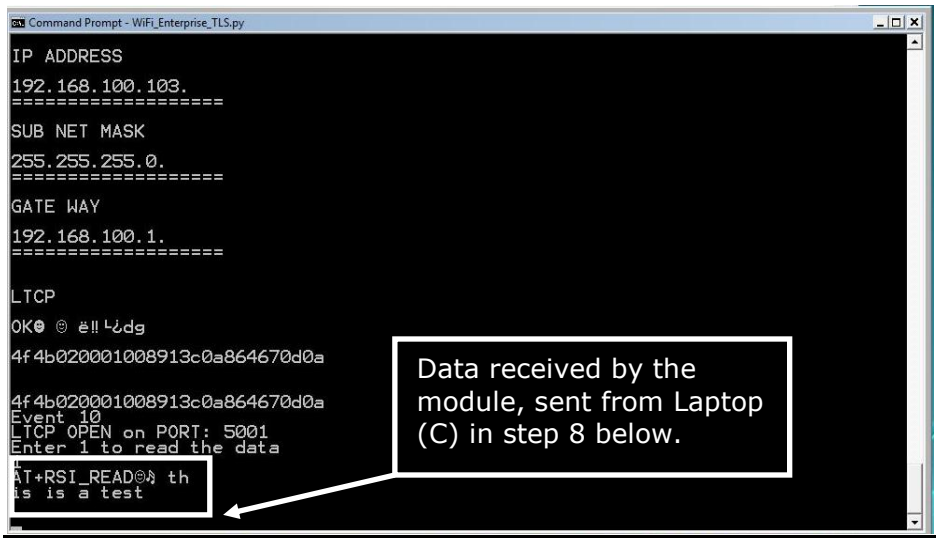

Figure 17: Running of Python Script WiFi_Enterprise_TLS.py

After opening the Server TCP socket, the script prompts to enter '1' to receive data. Enter '1' and proceed to step 6 below.

6. Connect Laptop C, as shown in the figure Demo Application Setup with Enterprise Security . It should have proper security credentials to connect to the AP.
7. A "ping[1]" can be issued from Laptop C to the Wi-Fi module to verify connectivity through the AP.
8. Exchanging data between the module and the Laptop (C)
   a. Open a client TCP socket on the Laptop C by typing the below command in the command line:

      TCP.exe c 2001 <Module's IP address> 5001

      The IP address of the module is displayed in the command window in Laptop (D) where the Python script was invoked. A server socket is already opened in the module by the Python script.

   The command line window in Laptop C will display three options: 1 (Send), 2 (Receive) and 3 (Exit).

   At module side once the socket connection is established below message is displayed

   AT+RSI_LTCP_CONNECT=<ip_version><socket descriptor>,<dst_port_no>,<dst_ipv4/ipv6_addr><mss>

   b. Enter '1' to send data to the Wi-Fi module. On being prompted "Enter the string to be transmitted", type any string from the Keyboard. On pressing "Enter" key on the keyboard, the data is sent from the Laptop(B) to the module and the terminal displays it with the **AT+RSI_READ** message as shown in figure Running of Python Script WiFi_Enterprise_TLS.py.

4.2.3.2 EAP-TTLS, and PEAP Demo Application

These modes can be evaluated using the HyperTerminal or Teraterm. The same set-up as shown in the figure Demo Application Setup with Enterprise Security can be used.

1. Run the radius server in the Laptop (D). Power up the module and issue the following commands.
   a. at+rsi_opermode=2,0,4,0

      This configures the EVB to function in client mode with Enterprise Security. The module responds with "OK".

   b. at+rsi_band=0

[1] The module can respond to a ping request sent from a remote terminal. There is no command to send a ping request from the module. This is true in all the modes- Client, AP and Wi-Fi Direct.

This configures the operating band of the EVB. The module responds with "OK".

c. **at+rsi_init**

This initializes the Wi-Fi module in the EVB. The module responds with OK<MAC_Address>

d. **at+rsi_fwversion?**

Optional command to report the firmware version in use.

e. **at+rsi_eap=TTLS,MSCHAPV2,user1,password1**

This configures the EAP mode of the module. In case of PEAP mode, change "TTLS" to "PEAP". The module responds with "OK".

f. **at+rsi_scan=0**

This makes the module scan for available networks. The module responds with information of the Aps scanned.

g. **at+rsi_join=Test_AP,0,2**

This commands the module to join to the AP "Test_AP". On successful association, the module responds with OK<GO_status>. The *GO_status* parameter can be ignored.

h. **at+rsi_ipconf=0,<desired IP>,<subnet>,<gateway>**

This command configures the module's IP in static mode (not DHCP). Make sure the desired IP is in the same subnet as the Access Point. The module responds to this command by sending the configured IP address to the Host as response to the command. In the terminal, this response might appear as unreadable characters because of ASCII conversion. The Serial Port Monitor can however be used to see the exact bytes. To get the IP is DHCP mode the command is **at+rsi_ipconf=1,0,0,0**

i. **at+rsi_ltcp=5001,1,0**

The module responds with

OK<ip_version ><socket_type> <socket_handle><Lport> <module_ipv4addr>\r\n

The *socket_handle* parameter will be used in the subsequent sections to send data.

2. Connect Laptop C, as shown in the figure Demo Application Setup with Enterprise Security . It should have proper security credentials to connect to the AP

   a. Open a client TCP socket on the Laptop C by typing the below command in the command line:

   TCP.exe c 2001 <Module's IP address> 5001

   The application is found in the path

\RS9113.NBZ.WC.GEN.OSI.1.0.10k\utils\peer_applications\.

The IP address of the module is displayed in the command window in Laptop (D) where the Python script was invoked.

b. The command line window will display three options: 1 (Send), 2 (Receive) and 3 (Exit). Type 1 to send data to the Wi-Fi module. On being prompted "Enter the string to be sent", type any string from the Keyboard. On pressing "Enter" key on the keyboard, the data is sent from the Laptop(B) to the module and the Hyperterminal displays it with the **AT+RSI_READ** message.

c. To send data from the Wi-Fi module, type Option 2 in Laptop C and then type the below command in the module

   at+rsi_snd=<socket_handle>,0,0,0,abcdefgh

   socket_handle is the parameter returned when the TCP socket is opened in the module. Refer to the programming reference manual for more details.

Note: To switch between Enterprise Security and any other security modes (WPA2-PSK, WPA, WEP) Operation mode has to be changed. Enterprise security works in Operation mode 2 and for other security modes the Operation mode is 0. The Change in the Operation mode command can happen just after the Reset (soft reset or hard reset) so Reset is required to toggle between Enterprise Security and other security modes.

4.3 Wi-Fi Direct Client Application

4.3.1 Demo Application Setup

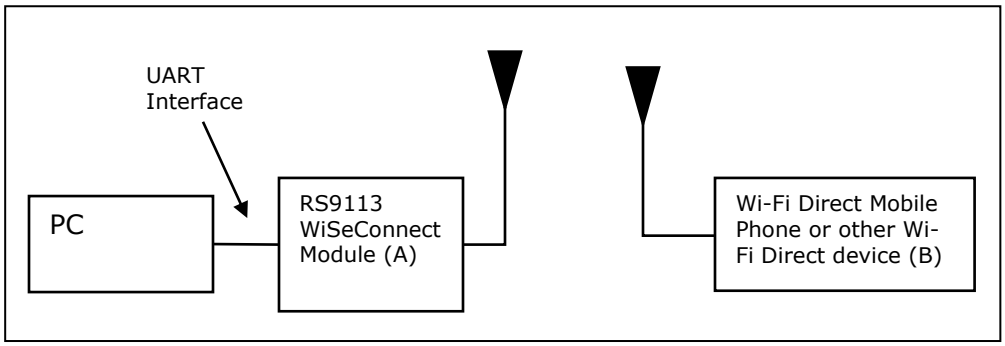

Figure 18: Wi-Fi Direct Set-up

In this set-up, a RS9113 based WiSeConnect EVB is at one end of the set-up and the other end is a Wi-Fi Direct device such as a phone.

4.3.2 Wi-Fi Direct Client Configuration

The following steps should be executed:

1. Enable Wi-Fi Direct Mode in the phone. Below is shown an example with an Android 2.3 phone. The Wi-Fi Direct mode can be configured by going into the phone's Wi-Fi Settings.

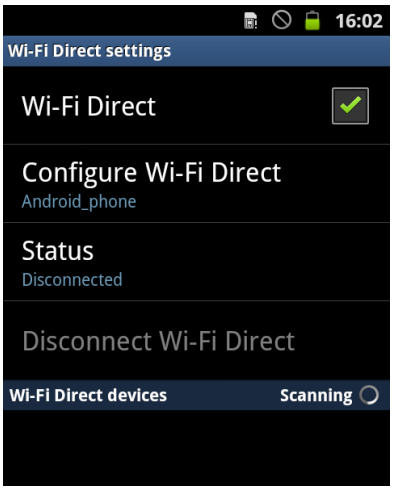

Figure 19: Android Phone Wi-Fi Direct Settings

The "Configure Wi-Fi Direct" button can be clicked to set the device name accordingly (Android_phone in this case).

2. Open the Serial Port Monitor to see the actual bytes exchanged. Open HyperTerminal or Teraterm.
3. Configure the EVB in UART mode; connect a USB and Power on the EVB.
4. The terminal will show the message "WELCOME TO REDPINE SIGNALS" followed by some more boot options. Make any selection as per your choice. Then the module boots up and sends Card Ready on successful completion of boot-up and a string "Loading done" is sent from module to host.
5. Enter the following commands[2]. A command should be entered only after getting the response of the previous command

   a. at+rsi_opermode=1,0,20,0

   This configures the EVB to function in Wi-Fi Direct mode. The module responds with "OK".

   b. at+rsi_band=0

   This configures the operating band of the EVB. The module responds with "OK".

   c. at+rsi_init

   This initializes the Wi-Fi module in the EVB. The module responds with OK<MAC_Address>.

   d. at+rsi_fwversion?

   Optional command to report the firmware version in use.

   e. at+rsi_wfd=0,RED,1,wiseconnect,12345678

   This starts the Wi-Fi Direct functionality in the module.

   The first parameter in this command is called the *Group_Owner_Intent*. It gives the willingness of the module to become a Group Owner. It has been set to the lowest value of 0 in this case as our module currently supports only Wi-Fi Direct client. Refer to the Programming Reference Manual for more details. The module responds with "OK".
6. After issuing the last command, the module starts scanning for Wi-Fi Direct devices, and reports any that are found through the asynchronous message **AT+RSI_WFDDEV**
7. After the module reports Wi-Fi Direct devices, issue the **Join** command to connect.

   at+rsi_join=Android_phone,0,2

   In this example, it is assumed that the device name of the phone is configured as "Android_phone". The phone should also display the device name of the module (RED in this example). Within about 10

[2] Please refer to the Programming Reference Manual for detailed description of all the commands.

secs of issuing the **Join** command in the module, click "Connect" on the phone as well to connect to the module. On successful connection, the module responds with OK and the Phone displays "Connected"

8. If the module has NOT become a GO (Group Owner), issue the below command to get an IP address from the phone. If the module has become a GO, this command need not be issued.

   at+rsi_ipconf=1,0,0,0

   The acquired IP address is returned to the Host and can be observed in Serial Port Monitor.

   Please refer to the description for the command *Join* in the Programming Reference Manual. The parameter *GO_Status* returned with the OK response of the *Join* command is used to determine whether the module became a GO or not. If the module becomes the GO, it will internally assign itself the IP **192.168.100.76** and will act as a DHCP server. The Wi-Fi Direct phone will acquire an IP address from the module automatically (assuming DHCP client is enabled in the phone). Since the *Group_Owner_Intent* in point #5 above has been set to the lowest value, module will become a Wi-Fi Direct client.

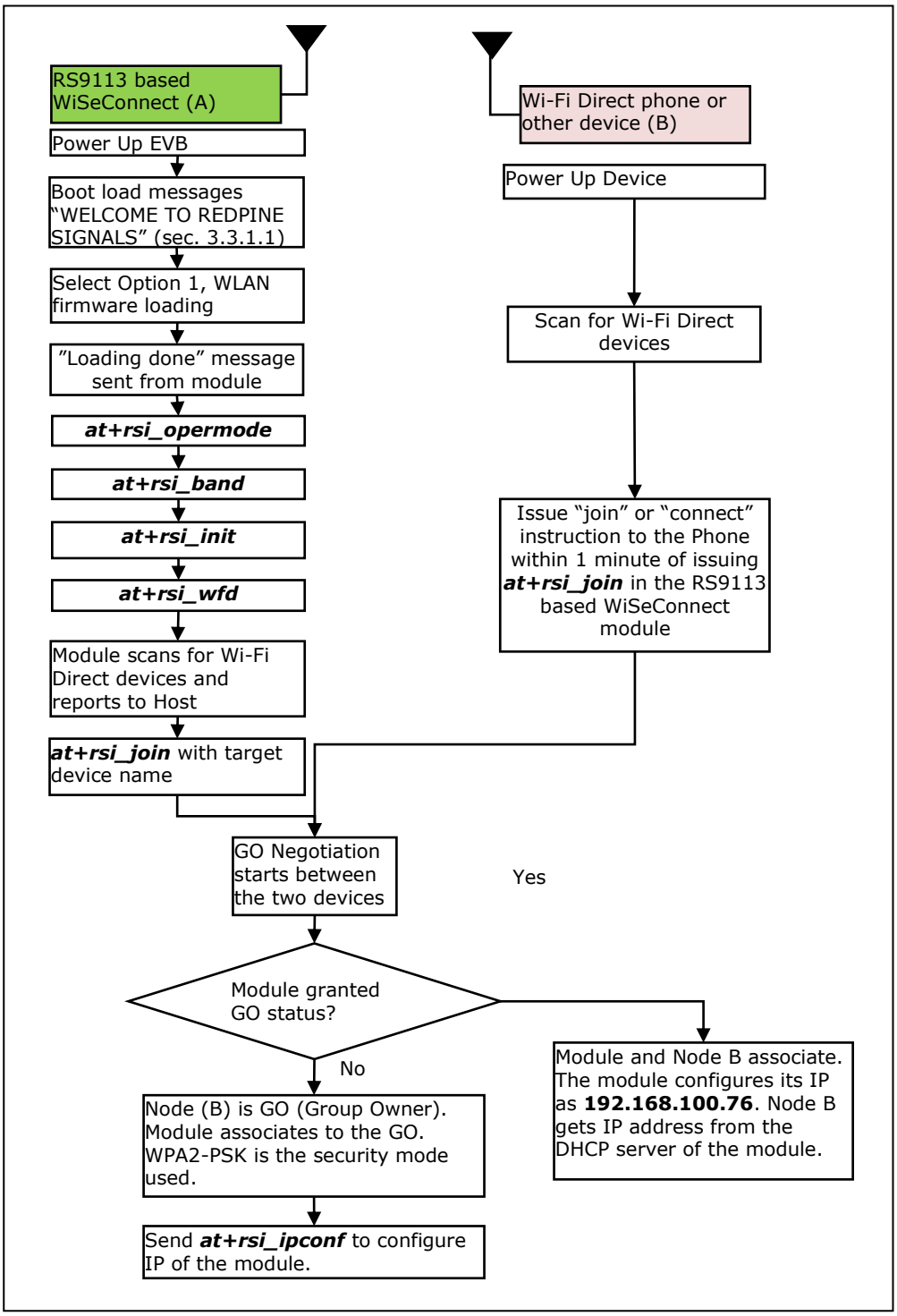

Figure 20: Command Flow in Wi-Fi Direct Mode

9. Exchanging Data with the Module

4.3.3 Demo Application Test Procedure

In the following descriptions it is assumed for illustrative purposes that the module's IP address is **192.168.100.76** and the remote peer (phone) IP address is **192.168.100.77**.

Running Ping Application

A ping application can be run from the Wi-Fi Direct Phone, with the destination address as 192.168.100.76. The module will send the ping response. Ping based applications are freely available for Android phones.

Exchanging data through sockets

For exchanging data between the module and the Wi-Fi Direct Phone, an application may be written by the user at the mobile phone to open sockets and transmit or receive data. At the module side, sockets can be opened by using the below commands:

a. To open a server TCP socket in the module, issue the below command.

   at+rsi_ltcp=5001,1,0

   5001 is the example port number of the socket opened. Once the socket is opened, a client socket should be opened at the remote peer (phone) to connect to this socket and establish a TCP connection. The module responds with

   The module responds with

   OK<ip_version ><socket_type> <socket_handle><Lport> <module_ipv4addr>\r\n

   The *socket_handle* parameter will be used in the subsequent sections to send data.

b. After the TCP connection is established, data can be exchanged between the two nodes. To send data from the module, issue the following command can be typed in Hyperterminal or Teraterm:

   at+rsi_snd=<socket_handle>,8,0,0,abcdefgh where abcdefgh is the data stream to be transmitted to the remote Peer. *Socket_handle* is the parameter returned when the TCP socket is opened in the module. Refer to the programming reference manual for more details.

   If the remote peer sends data to the module, the module would receive it and show the data in the terminal with the message **AT+RSI_READ**<socket handle><size><Source IP><Source port><data stream>

4.4 Access Point Mode

4.4.1 Demo Application Setup

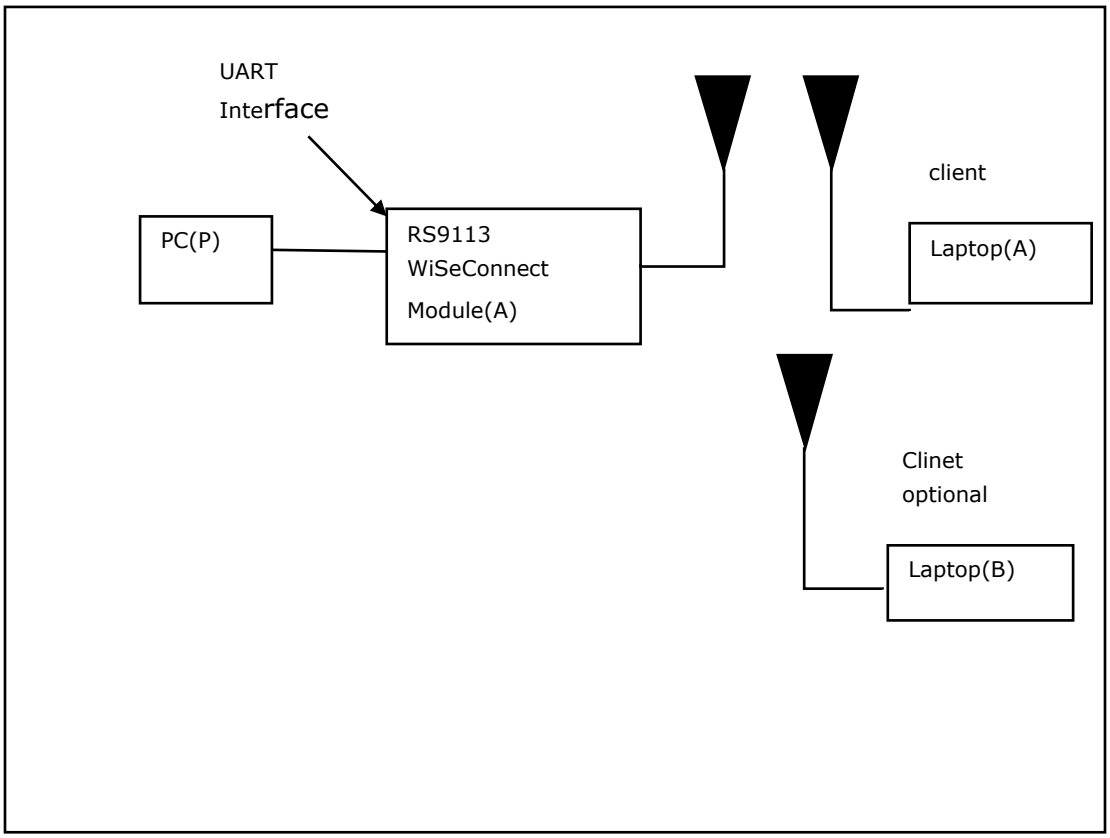

Figure 21: Access Point Set-up

4.4.2 Access Point Configuration

The following steps should be executed:

1. Open the Serial Port Monitor to see the actual bytes exchanged. Open Hyperterminal or Teraterm.
2. Configure the EVB in UART mode, connect a USB and Power on the EVB.
3. The terminal application displays the message "WELCOME TO REDPINE SIGNALS" with boot options as described in section 3.1 (Figure 5). Select option 1 to load WLAN firmware. On successful boot-up of the module, "Loading done" message is sent from module to the Host. The following commands can now be issued. Please refer to the RS9113WiSeConnect/Connect-io-n modules PRM document for detailed description of the commands and their responses.

A command should not be sent until the response of the previous command is received

4. Enter the following commands. A command should be entered only after getting the response of the previous command

a. at+rsi_opermode=6,1,48,0

This configures the EVB to function in AP mode. The module responds with "OK".

b. at+rsi_band=0

This configures the operating band of the EVB. The module responds with "OK".

c. at+rsi_init

This initializes the WiFi module in the EVB. The module responds with OK<mac_address>

d. at+rsi_fwversion?

Optional command to report the firmware version in use.

e. at+rsi_ipconf=0,192.168.0.30,255.255.255.0,192.168.0.30

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.

f. at+rsi_apconf=6,redpine,0,0,0,200,3,3

The SSID is configured as "redpine", to operate in channel 6.

g. at+rsi_join=redpine,0,2

This starts the Access Point functionality in the module.

The module is now configured as an Access Point. Its IP address is **192.168.0.31.** A Laptop can now scan for networks and the SSID of the module, "redpine" will be displayed in the Laptop's list of Scanned APs. After the client Laptop (B) connects to the AP, it acquires an IP address. It is assumed for illustrative purposes that the IP of the Laptop is **192.168.0.32**

5. Exchanging data between the client Laptop (B) and the Access Point

a. Open a Listening TCP socket on the module by issuing the command

at+rsi_ltcp=5001,1,0

b. Open a TCP socket on the client Laptop (B) (Windows OS based) by typing the below command on the command line:

TCP.exe c 2001 192.168.0.30 5001 The application is found in the path RS.WSC.x.x.GENR.x.x.x.x.x.x\Resources\Applications\Peer\Windows\. The destination IP (the Access Point in this case) is 192.168.50.1.

At module side once socket connection is established below message is displayed

AT+RSI_LTCP_CONNECT=<ip_version><socket descriptor>,<dst_port_no>,<dst_ipv4/ipv6_addr><mss>

c. The command line window will display three options: 1 (Send), 2 (Receive) and 3 (Exit). Type 1to send data to the Wi-Fi module. On being prompted "Enter the string to be transmitted", type any string from the Keyboard. On pressing "Enter" key on the keyboard, the data is sent from the Laptop(B) to the module and the Hyperterminal displays it with the **AT+RSI_READ** message.

d. To send data from the Wi-Fi module, type Option 2 in Laptop (B) and then type the below command in the module

at+rsi_snd=<socket_handle>,8,0,0,abcdefgh

<socket_handle> is the parameter returned by the module when a socket is opened in the module. Refer to the programming reference manual for more details. A second Laptop I can also be connected to the module‟s AP and data transfers can be done between it and Laptop (B).

4.5 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").

4.5.1 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.

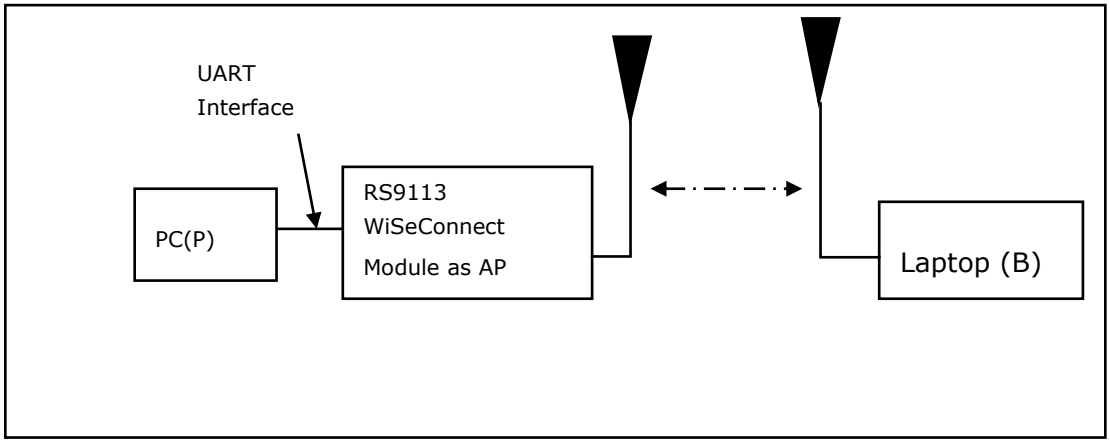

Figure 22: Module Configured as Access Point

1. Connect a PC or Host to the module through the USB and power up the module.
2. Configure the module to become an AP by issuing commands from PC (P) as in section Evaluation of Access Point Mode. The sequence of commands is given below.

   a. at+rsi_opermode=6 ,1,48,0

   b. at+rsi_band=0

   c. at+rsi_init

   d. at+rsi_fwversion?

   e.at+rsi_ipconf=0,192.168.0.30,255.255.255.0,192.168.0.30

   f. at+rsi_apconf=6,redpine,0,0,0,10000,3,3

   g. at+rsi_join=redpine,0,2

The module is now configured as an Access Point. Its IP address is **192.168.0.30.**

3. Connect a Laptop (B) to the created AP. Open the URL **http://<Module's IP address>/Index.htm** in the Laptop. In this case, the URL is http://192.168.0.30/Index.htm Make sure the browser in the laptop does not have any proxies enabled.
4. In the web page that opens, displaying the "Welcome to Redpine Signals" and click on the "GO TO CONFIGURATION PAGE" as shown in below.

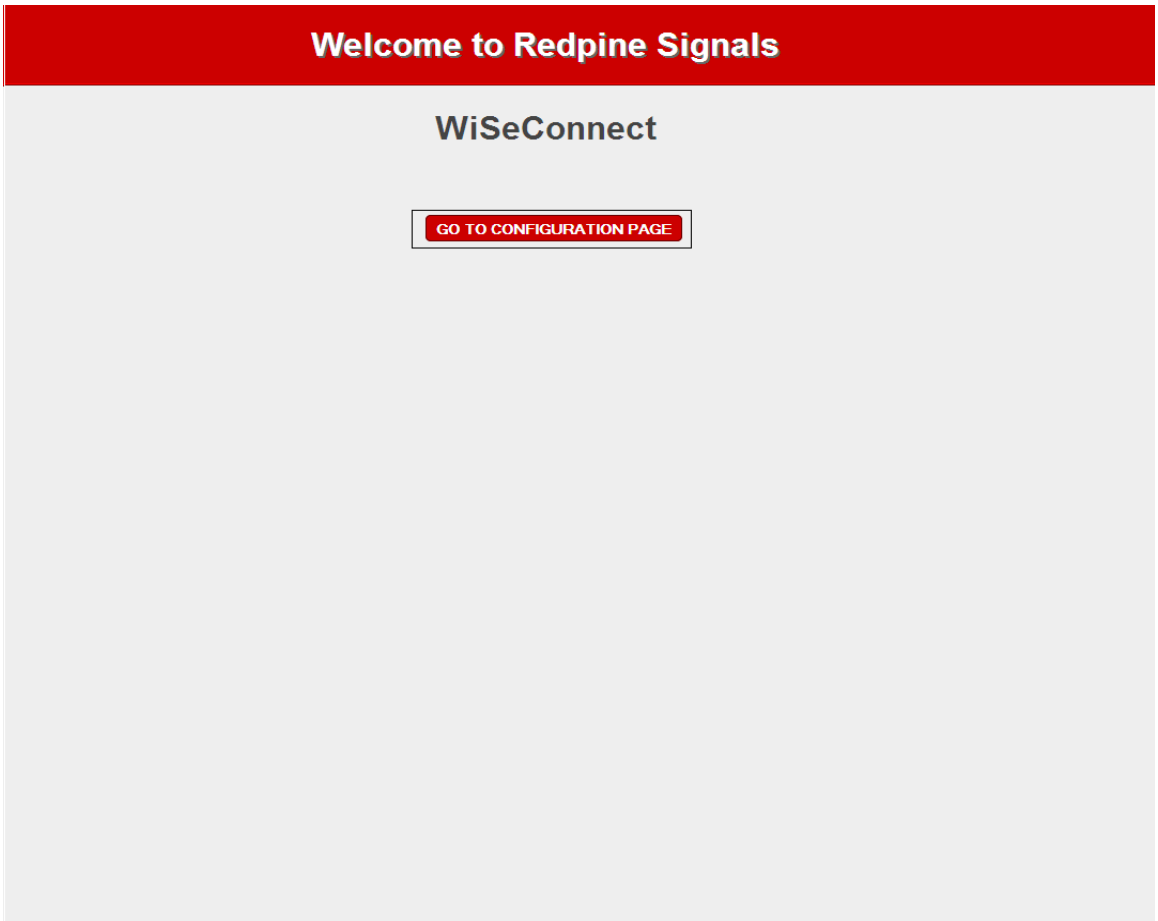

Figure 23: Module web page

5. After that appears the "WiSeConnect Wireless Configuration" page, in the page select

OPERATING MODE: Client

BAND (GHZ): 2.4

SSID: This is the SSID of the AP to which the module should connect after configuration is over.

TX POWER: RF power for Tx. Set to '2'.

CHANNEL: Channel number at which the target AP is present. Set to "11" in this example.

SECURITY ENABLE and PSK: This should match the security mode of the AP to which the module should connect (in this example AP is open mode, so no need to enter the PSK value).

IP VERSION: set IPv4

DHCP: If DHCP is selected, the module will work as a DHCP client, otherwise, an IP should be hard coded in the web page.

For more details, refer to the section on wireless configuration in the PRM.

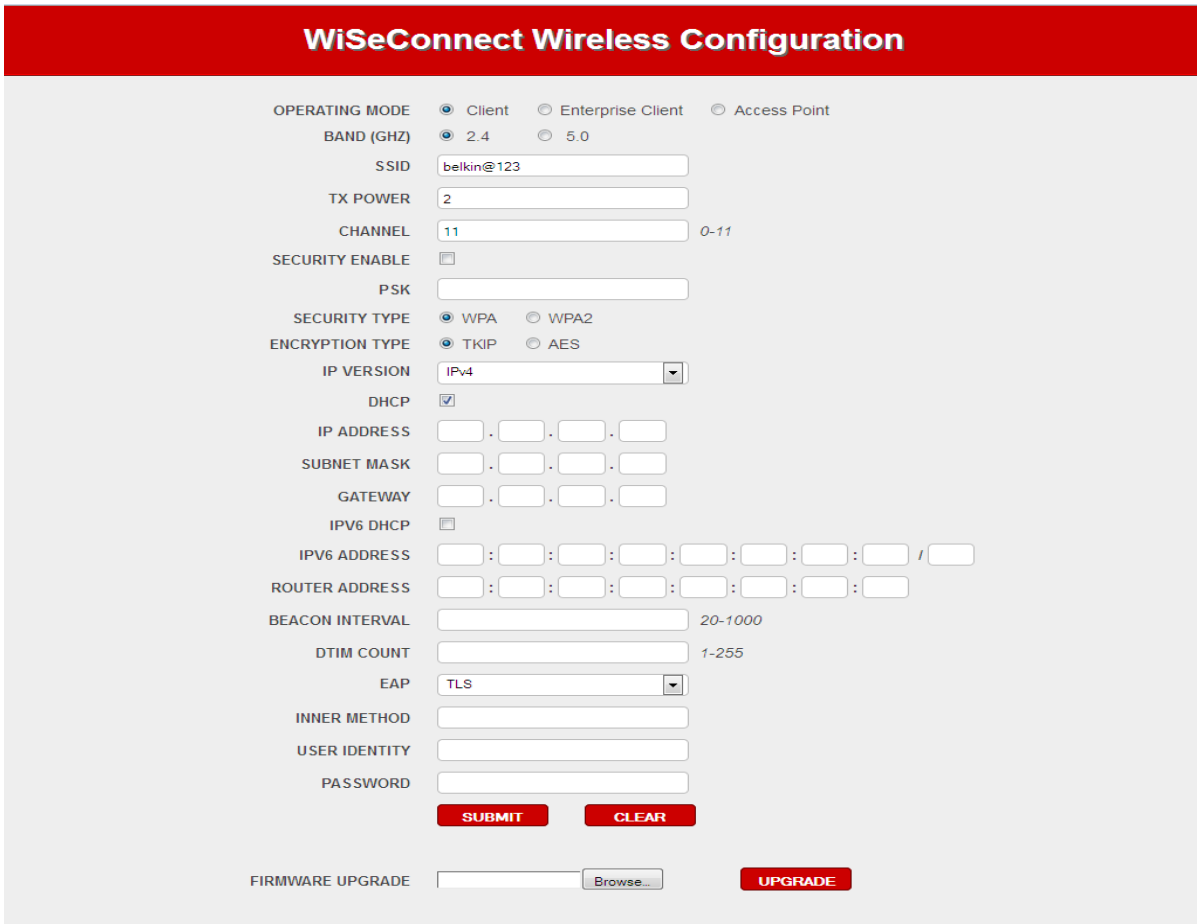

Figure 24: Wireless Configuration page

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 internally given "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 "at+rsi_cfgget". If the auto-connect feature needs to be disabled, issue the command "at+rsi_cfgenable=0". Refer to the PRM for more details on these commands.

Flow 2: In this flow, the module is connected to an AP. A remote device connects to the same AP and configures the module.

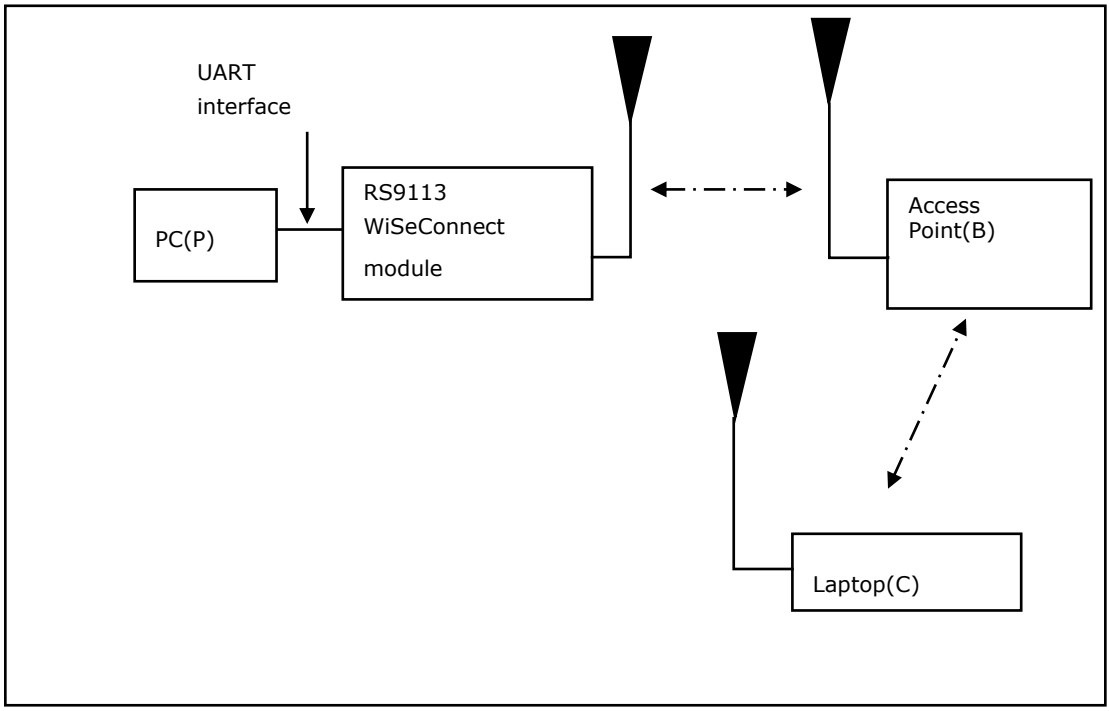

Figure 25: Module Connected to Access Point

1. Connect a PC or Host to the module through the USB and power up the module.
2. Configure the module to become a client and connect to an AP as described in section Evaluation of Client Mode with Personal Security.
3. Connect a Laptop (C) to the same AP. Open the URL **http://<Module's IP address>/Index.htm** in the Laptop. For

example, if the module was configured to have an IP of 192.168.100.20, then the URL is http://192.168.100.20/Index.htm. Make sure the browser in the laptop does not have any proxies enabled.

4. In the web page that opens, displaying the "Welcome to Redpine Signals" and click on the "GO TO CONFIGURATION PAGE" (displaying the web page in figure 23).
5. After that displaying the "WiSeConnect Wireless Configuration" page, in the page select

OPERATING MODE: Client

BAND (GHZ): 2.4

SSID: This is the SSID of the AP to which the module should connect after configuration is over.

TX POWER: RF power for Tx. Set to '2'.

CHANNEL: Channel number at which the target AP is present. Set to "11" in this example.

SECURITY ENABLE and PSK: This should match the security mode of the AP to which the module should connect (in this example AP is open mode, so no need to enter the SECURITY ENABLE and PSK value).

IP VERSION: set IPv4

DHCP: If DHCP is selected, the module will work as a DHCP client, otherwise, an IP should be hard coded in the web page.

For more details, refer to the section on wireless configuration in the PRM.

Figure 26: Wireless Configuration Page

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 internally given "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 "at+rsi_cfgget". If the auto-connect feature needs to be disabled, issue the command "at+rsi_cfgenable=0". Refer to the PRM for more details on these commands.

4.5.2 Configuration 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.

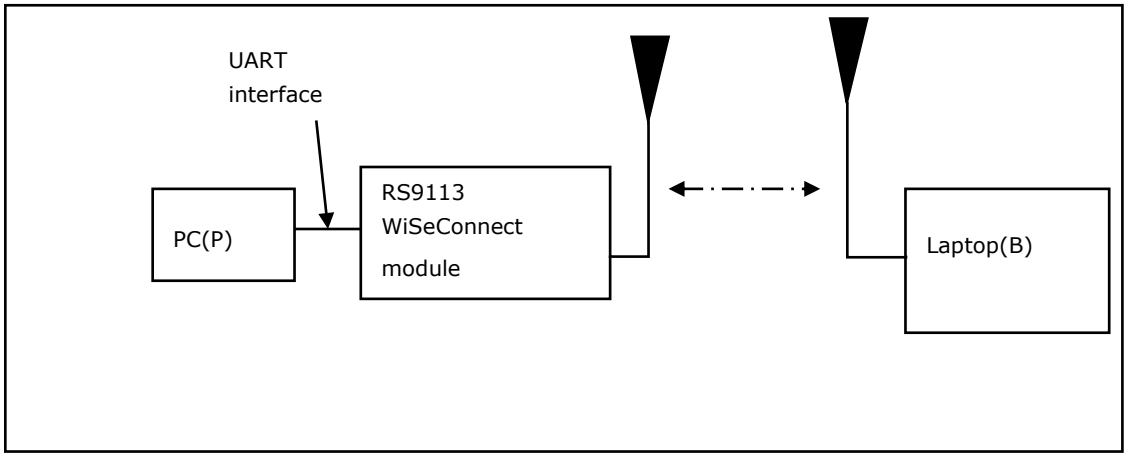

Figure 27: Module Configured as Access Point

1. Connect a PC or Host to the module through the USB and power up the module.
2. Configure the module to become an AP by issuing commands through PC (P).
3. Connect a Laptop (B) to the created AP. Open the URL **http://<Module's IP address>/Index.htm** in the Laptop. For example, if the module was configured to have an IP of 192.168.100.1, then the URL is http://192.168.100.1/Index.htm. Make sure the browser in the laptop does not have any proxies enabled.
4. In the web page that opens, displaying the "Welcome to Redpine Signals" and click on the "GO TO CONFIGURATION PAGE" as shown in below

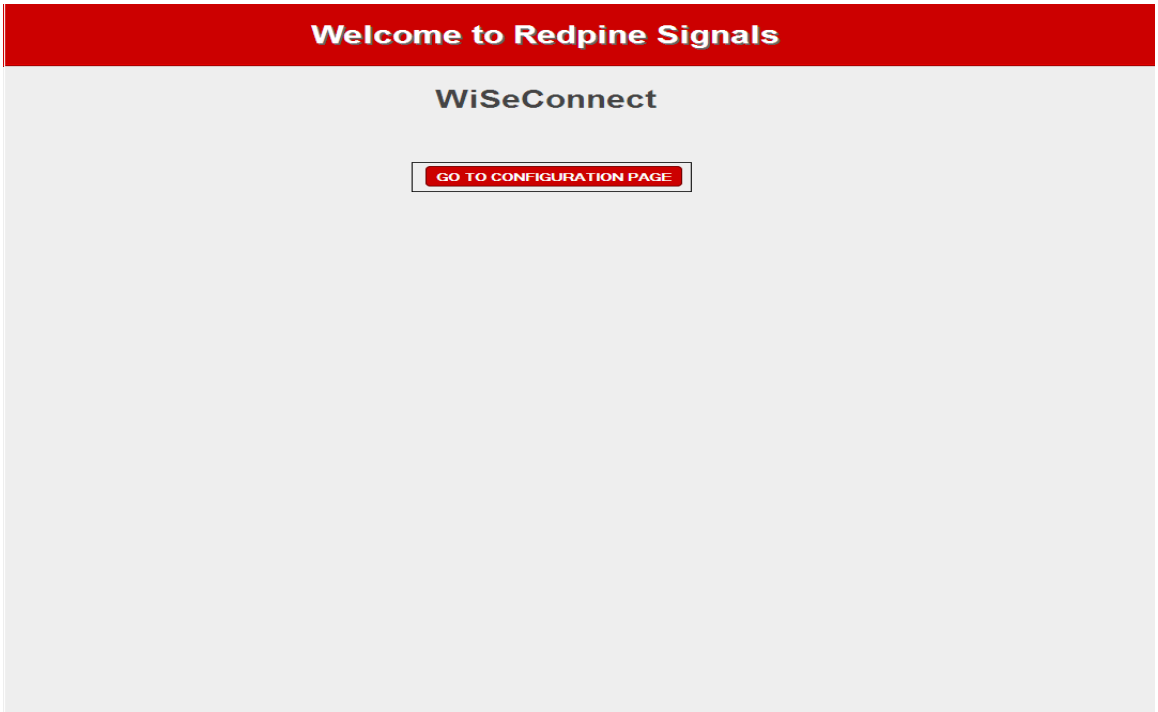

Figure 28: Module web page

5. After that displaying the "WiSeConnect Wireless Configuration" page, in the page select

OPERATING MODE: Access point mode

BAND(GHZ): 2.4

SSID: This is the SSID of the AP which will be created after configuration is over

TX POWER: RF power for Tx. Set this value to "2" .

SECURITY ENABLE, PSK, SECURITY TYPE, ENCRYPTION TYPE: This is to configure the security mode of the AP.

CHANNEL: Channel number at which the target AP is present. Value of "0" is not allowed.

IP ADDRESS, SUBNET MASK, GATEWAY: These parameters set the IP parameters of the AP.

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.

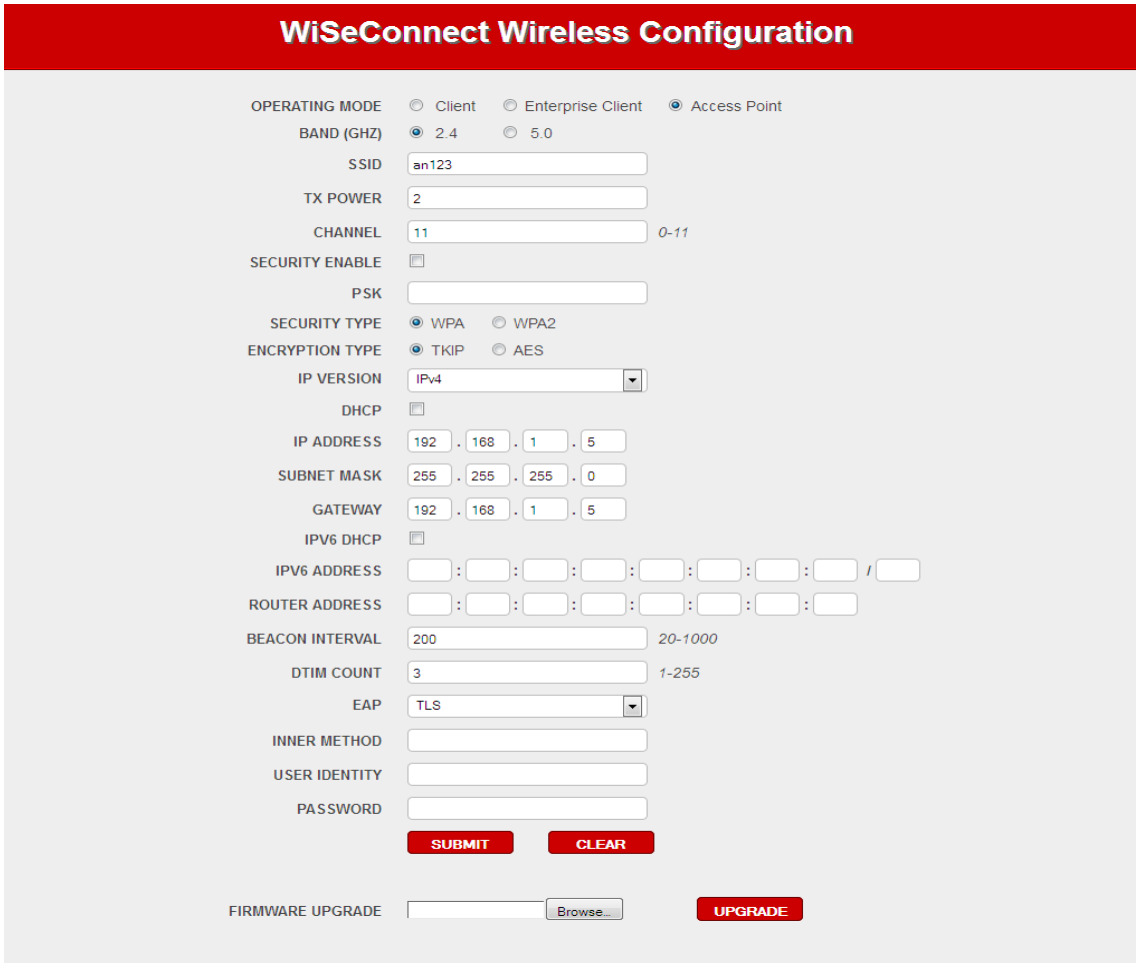

Figure 29: Wireless Configuration page

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 creates and AP with the configured parameters. The module will send out two responses to the Host, the first corresponds to the internally given "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 "at+rsi_cfgget". If the auto-connect feature needs to be disabled, issue the command "at+rsi_cfgenable=0" to the module. Refer to the PRM for more details on these commands.

Flow 2: In this flow, the module is connected to an AP. A remote device connects to the same AP and configures the module.

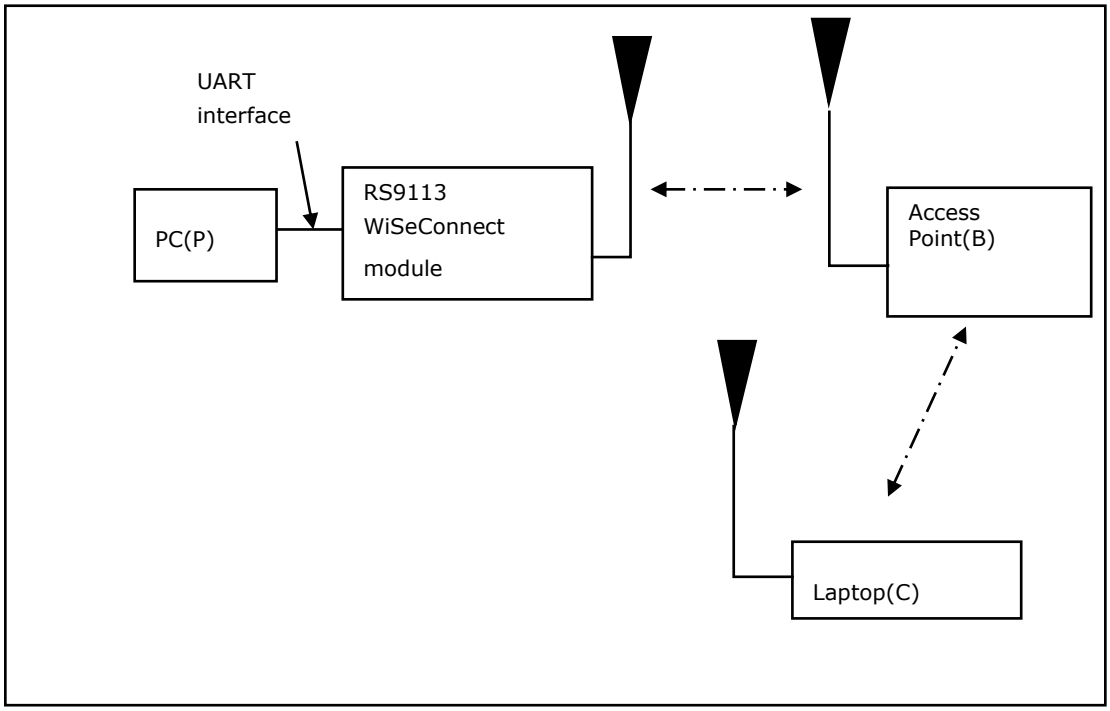

Figure 30: Module Connected to Access Point

1. Connect a PC or Host to the module through the USB and power up the module.
2. Configure the module to become client and connected to an AP by issuing commands through PC (P).
3. Connect a Laptop (c) to the created AP. Open the URL **http://<Module's IP address>/Index.htm** in the Laptop. For example, if the module was configured to have an IP of 192.168.100.1, then the URL is http://192.168.100.1/Index.htm. Make sure the browser in the laptop does not have any proxies enabled.
4. In the web page that opens, displaying the "Welcome to Redpine Signals" and click on the "GO TO CONFIGURATION PAGE" (displaying the web page in figure 28).
5. After that displaying the "WiSeConnect Wireless Configuration" page, in the page select

OPERATING MODE: Access point mode

BAND(GHZ): 2.4

SSID: This is the SSID of the AP which will be created after configuration is over

TX POWER: RF power for Tx. Set this value to "2" .

SECURITY ENABLE, PSK, SECURITY TYPE, ENCRYPTION TYPE: This is to configure the security mode of the AP.

CHANNEL: Channel number at which the target AP is present. Value of "0" is not allowed.

IP ADDRESS, SUBNET MASK, GATEWAY: These parameters set the IP parameters of the AP.

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.

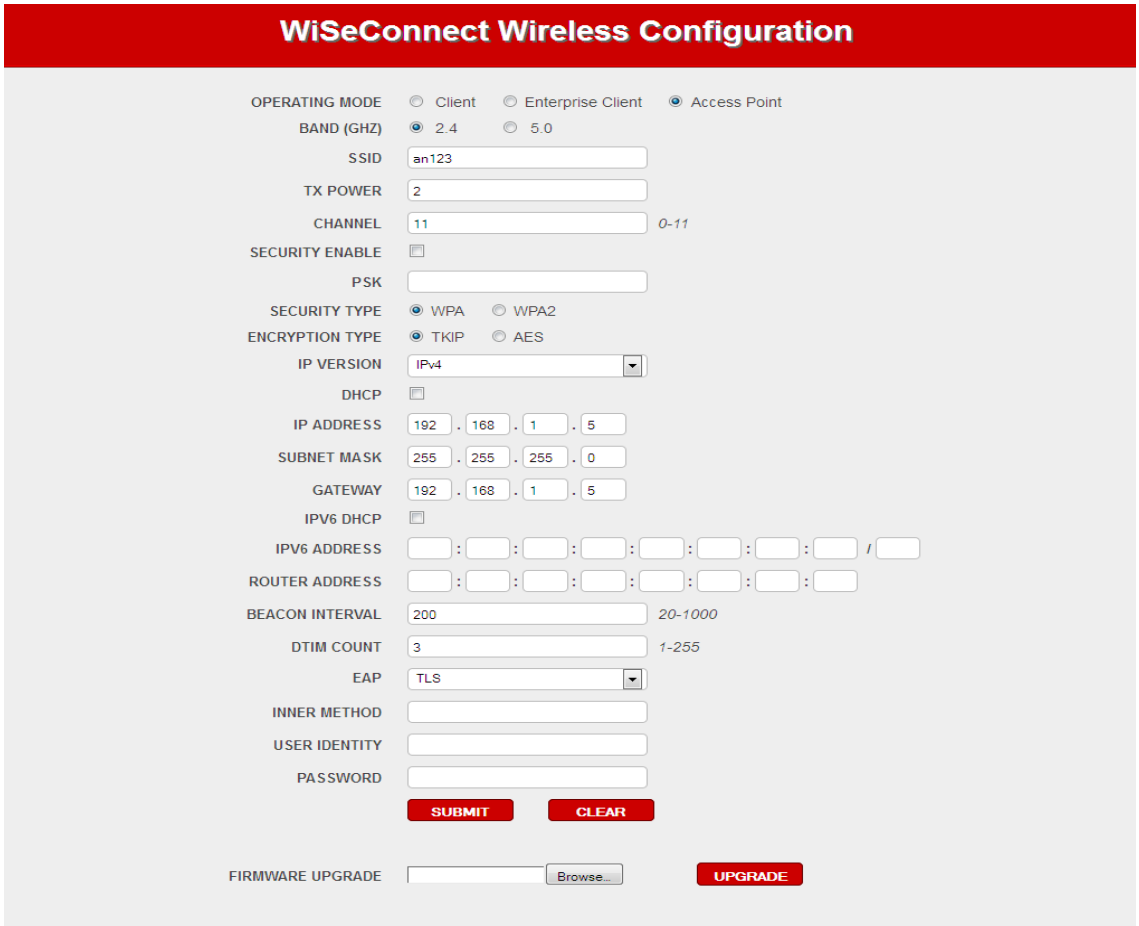

Figure 31: Wireless Configuration page

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 creates and AP with the configured parameters. The module will send out two responses to the Host, the first corresponds to the internally given "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 "at+rsi_cfgget". If the auto-connect feature needs to be disabled, issue the command "at+rsi_cfgenable=0" to the module. Refer to the PRM for more details on these commands.

5 Using the module in USB CDC -ACM mode

This section explains procedure to follow to configure and the send WiFi commands to the module and receive response from the module using USB.

5.1 USB CDC-ACM mode

The USB interface in the mode corresponds to the CDC-ACM class and presents itself as a USB Device to the Host USB. A file is provided with the software package that the user should install in the Host platform, in order to communicate with the module.

5.1.1 Demo application set up

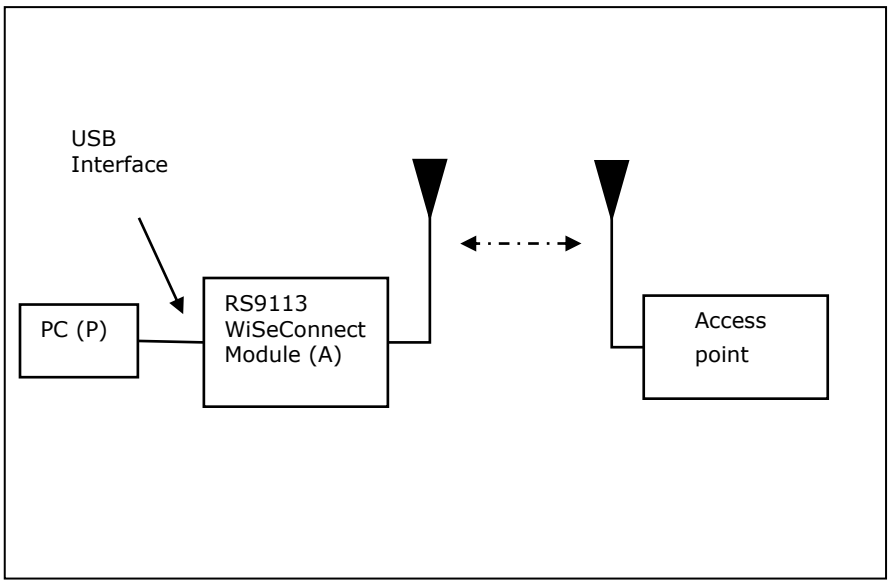

Figure 32: USB CDC-AMC set up

5.1.2 USB CDC-ACM usage

A sample flow is provided below to use the module with a PC"s USB interface.

1. Configure the EVB in USB mode (USB CDC Host Interface Selection) and connect the USB interface of a PC to that of the module. The PC prompts for installing the driver. Install the file from RS9113.WC.GENR.x.x.x\utils\usb_cdc\rsi_usbcdc.inf. The file needs to be installed only once.
2. Power cycle the module. Check the list in "Ports" in the *Device Manager* Settings of the PC. It should show the device as "RSI WSC Virtual Com Port.

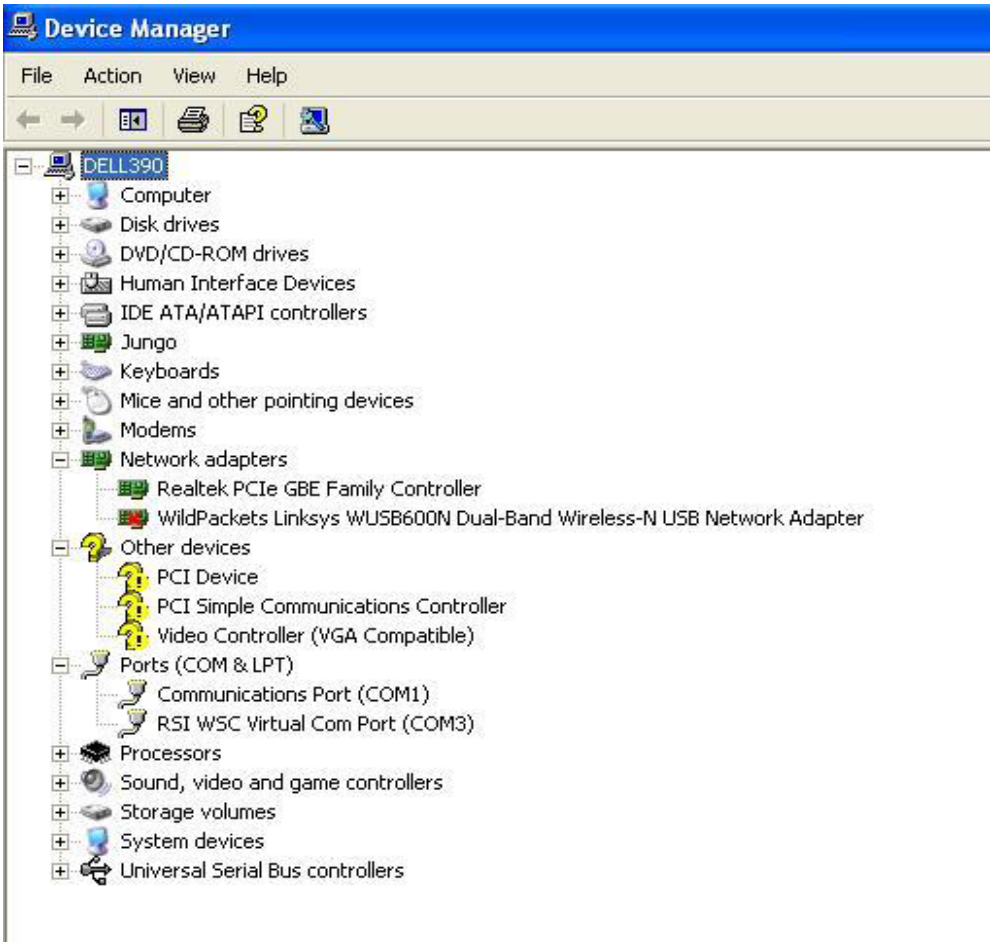

Figure 33: PC device COM port listening

3. Open Hyperterminal and set Flow Control to "None". Baud rate, Data bits, Parity and Stops bits are "Don"t care" fields in USB mode.
4. Enable the GPIO Bypass mode refer the section 3.2.1
5. Now AT commands can be issued to communicate with the module through the virtual Com port. The behavior of the module, commands, command responses, error codes and sequence of commands are exactly same as in the UART mode. The USB interface of the module supports the full speed USB mode (12 Mbps physical data rate).
6. After completion of all commands to disable the GPIO bypass mode refer section3.2.1.1.

6 Evaluation of Bluetooth Classic

For evaluating Bluetooth Classic (BT), please refer document "**RS9113-WiseConnect-BT-Classic-Software-PRM**".

This document describes the commands to operate the RS9113-WiSeConnect Module Family for Bluetooth. OT-Bluetooth stack is used for Host layers and the commands describe various profiles supported by OT-Bluetooth stack. This document should be used by the developer to write software on Host MCU to control and operate the module.

7 Evaluation of Bluetooth Low Energy

For evaluating Bluetooth Low Energy (BLE), please refer document "**RS9113-WiseConnect-BLE-Software-PRM**".

This document describes the commands to operate the RS9113-WiSeConnect Module Family for Bluetooth. OT-Bluetooth stack is used for Host layers and the commands describe various profiles supported by OT-Bluetooth stack. This document should be used by the developer to write software on Host MCU to control and operate the module.

8 Evaluation of ZigBee

For evaluating ZigBee (ZB), please refer document "**RS9113-WiseConnect-ZigBee-Software-PRM**".

This document describes the commands to operate the RS9113-WiSeConnect Module Family for ZigBee. Host layers use RS9113-WiSeConnect module to communicate with other ZigBee devices using various profiles. Various Command requests along with the expected responses from the modules and the parameters in the commands are also described. RS9113-WiSeConnect Module Family can be operated in either End Device or Router or Co-ordinator mode at a time. This document should be referred by the developer to write software on Host to control and operate the module.

9 Using the module in SPI mode

To evaluate the module in SPI mode, the following steps should be followed:

1. SPI interface of the EVB should be interfaced with the Host MCU.

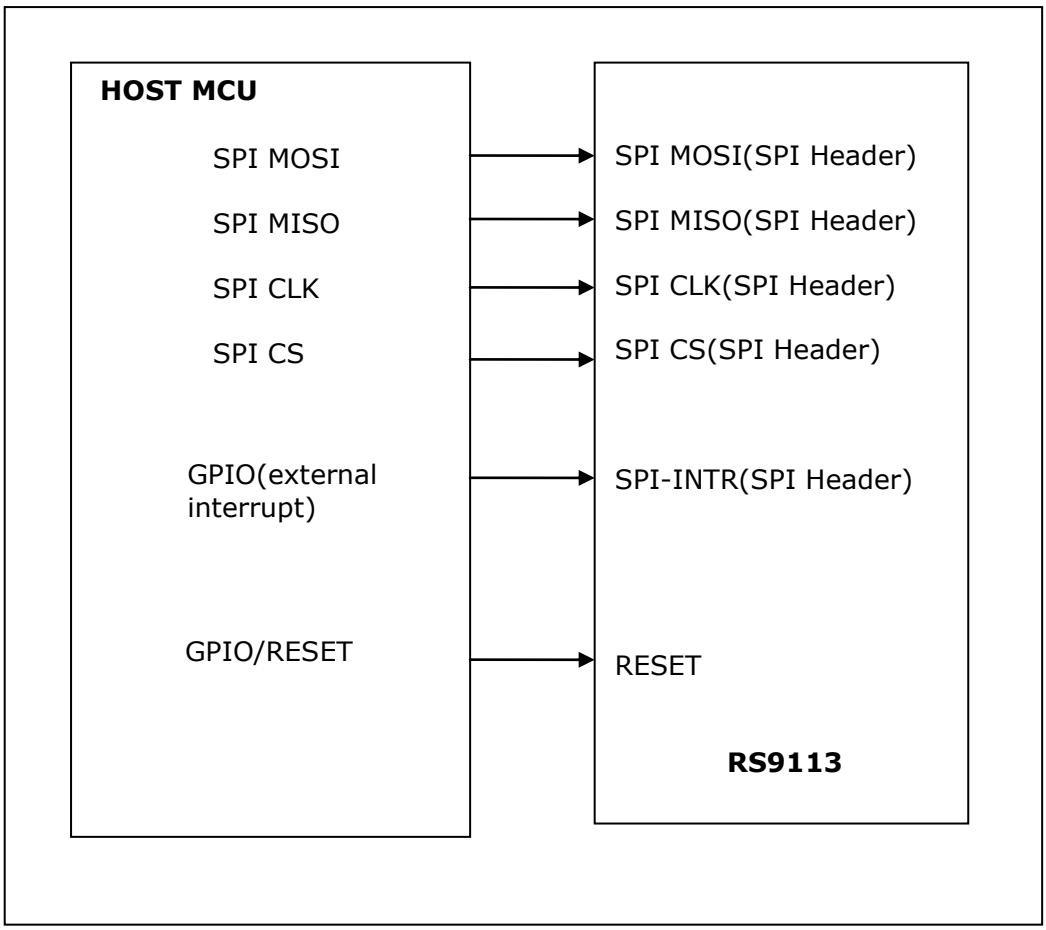

Figure 34: Interface between module and host

Signal Integrity Guidelines for SPI interface: Glitches in the SPI clock may take the SPI interface out of synchronization. The quality and integrity of the clock line should be maintained. The following steps are recommended. This is not an exhaustive list of guidelines and depending on individual cases additional steps may be needed.

1. Avoid using long cables to connect the Host platform with the EVB‟s SPI interface. If a cable is used, minimize its length to as small as possible, preferably to within two inches.
2. Increase the number of ground connections between the Wi-Fi PCB and the MCU PCB.
3. Add a series resistor into the clock line. Choice of value is mentioned in the Module Integration Guide.
4. If the SPI clock line is mapped to a programmable I/O on the MCU, configure that I/O to an output with as high a drive as is available.
5. Ensure that the EVB‟s reset input is mapped to a MCU controllable line, so that the system can recover through a hard reset.

according to table Interface Selection.

3. The source code of a sample driver, API set and application is provided with the Software Package. The developer should port the source code into the target platform. Appropriate HAL changes may be required for porting.
   a. The Driver source code is present in the software package inside RS.WSC.x.x.GENR.x.x.x.x.x.x\host
   b. The porting guidelines to port the driver are present in the document RS WiSeConnect Software-PRM.pdf in the section "**Driver Porting Guide for SPI**".
   c. The section "**RS9113 WiSeConnect SPI mode**" in RS9113-WiSeConnect -Software-PRM should be carefully read to understand all the commands to configure and operate the module.
4. The following settings should be observed to use the SPI interface
   a. SPI CPOL=0
   b. SPI CPHASE=0
   c. Max SPI Clock = 80 Mhz (For detailed timing information on the SPI signals, please refer to the Datasheet).
   d. The Interrupt output from the module is active high, level triggered.

9.1 Sample flow for evaluating SPI mode

The set-ups described in the previous sections describing the UART interface (Client Mode with Personal Security, Client Mode with Enterprise Security, Wi-Fi Direct Mode and Access Point mode can be used to evaluate the module in

SPI mode. A sample flow for evaluating the module in different modes is given below. The Application running in the Host should execute this flow by calling the different APIs provided.

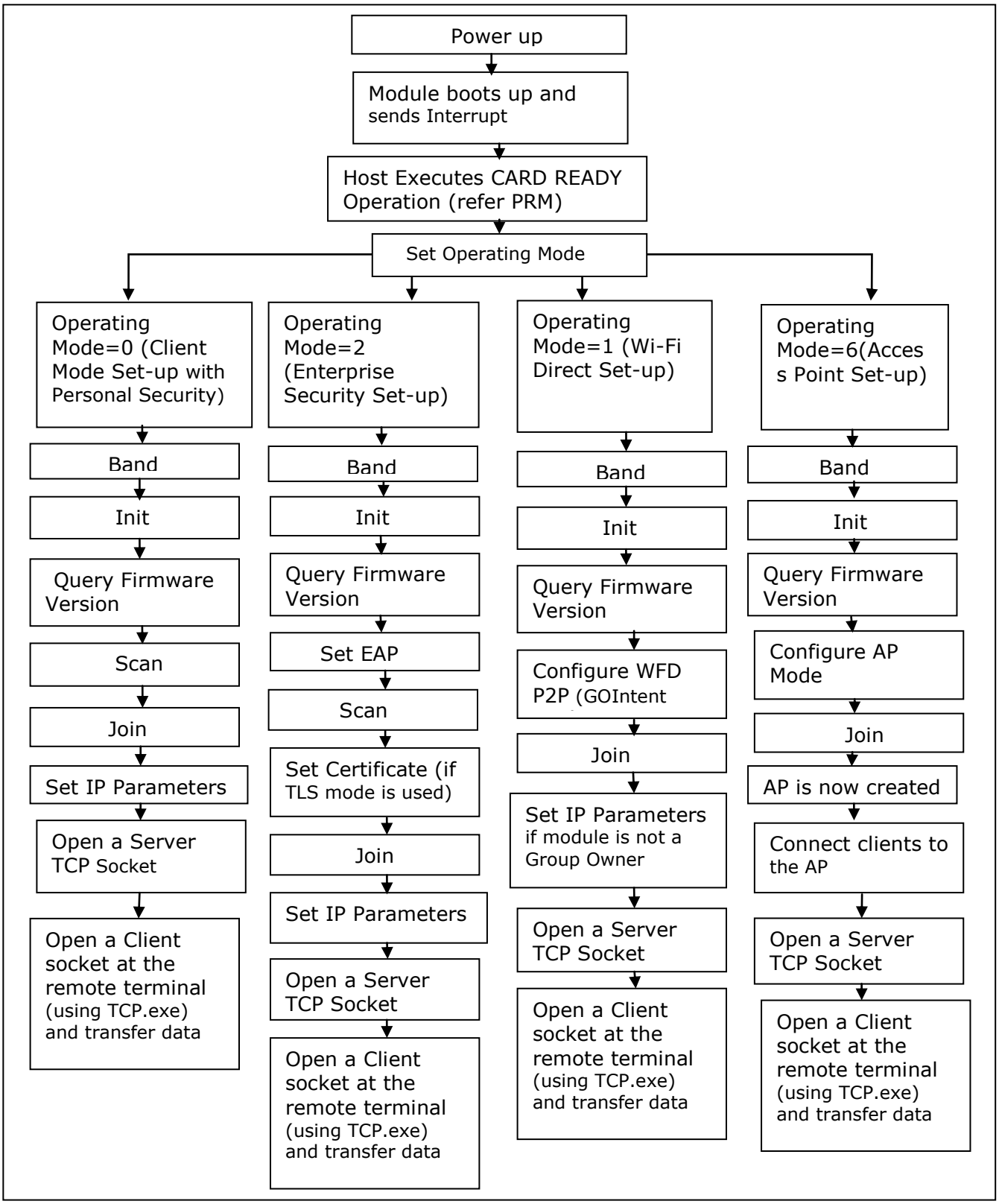

Figure 35: Flow commands in SPI mode

10 Appendix A: Serial port Host Applications in UART Mode

The following sub sections describe the Host applications required and their configuration settings to evaluate the EVB on UART interface.

To find the available COM ports, open the computer's Device Manager settings.

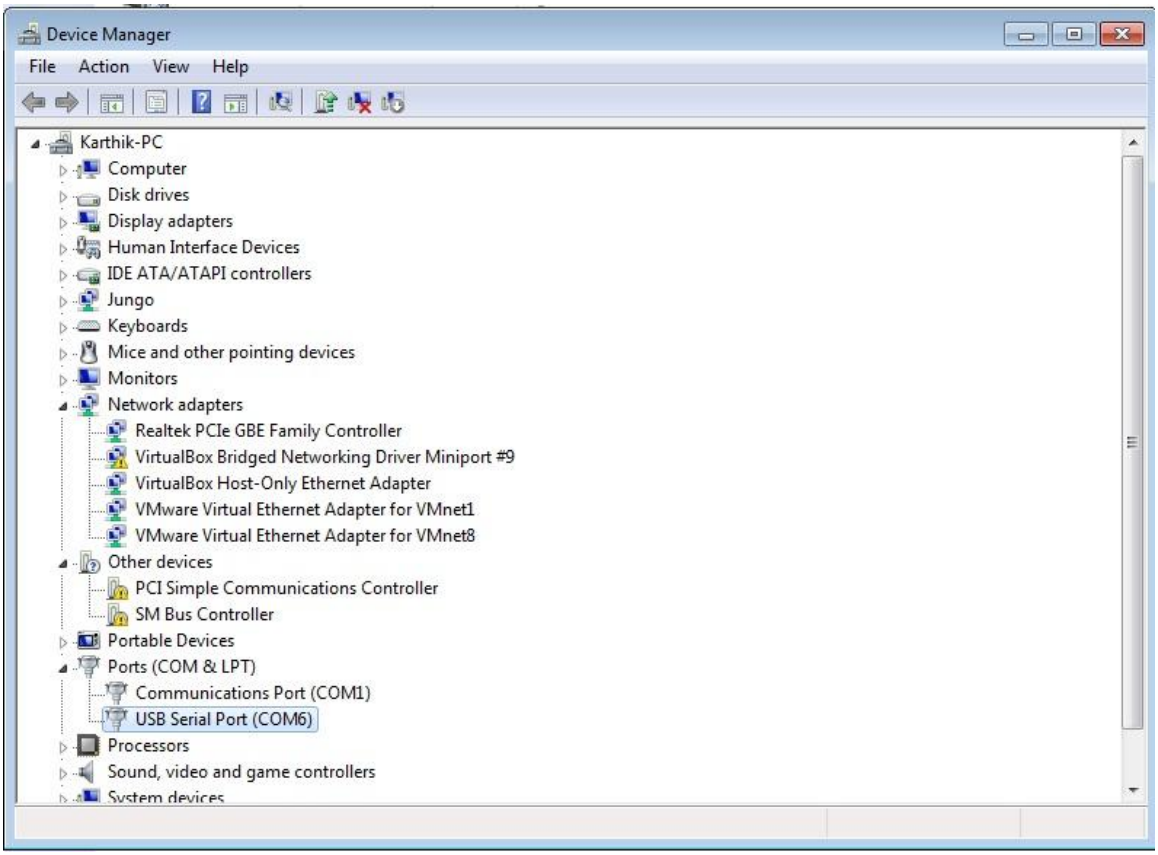

Figure 36: PC Device Manager COM ports listing

In the figure 44 above, listed under "Ports (COM & LPT)", the USB serial port COM6 is free. The COM port numbers will change across PC's. Hence, what is shown in the figures and the sections that follow are for illustration purposes only.

HyperTerminal or Tera Term should be opened and configured accordingly with the COM port identified in the device manager as described above.

10.1 Hyper Terminal Configuration

The steps for configuring HyperTerminal are shown below. The COM port selection in the following description may not be the same as selected in the screen-shot above as the figures are only for illustration purposes only.

- Open HyperTerminal
- Enter any name in the "Name" field then click "OK" button. Here "test" is entered as shown in the figure 33 below.

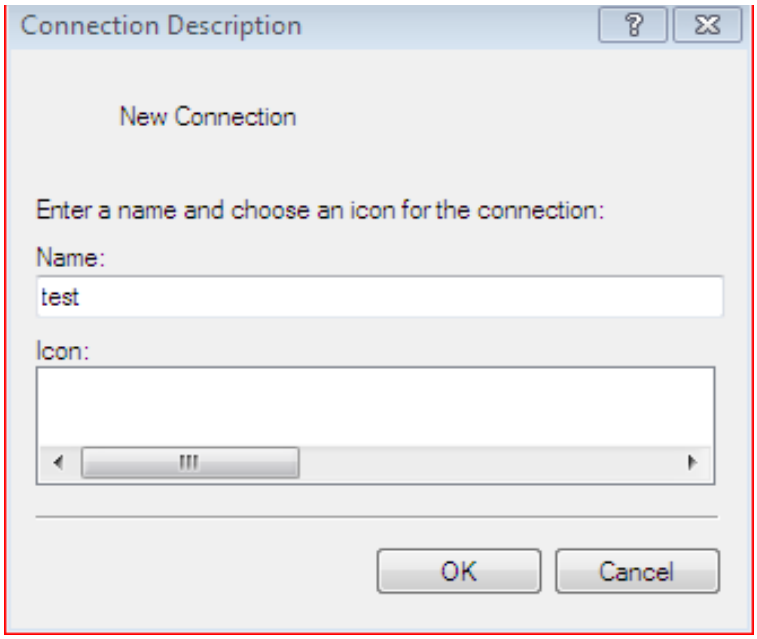

Figure 37: Hyper Terminal Name field Configuration

- In the "Connect using" field select appropriate com port. In the figure 34 below COM1 was selected. Then click "OK" button.

Figure 38: Hyper Terminal COM port field Configuration

In the Port Settings dialog box as shown in figure 35, configure the fields as follows.

- Configure "Bits per second" to 115200.
- Configure "Data bits" to 8.
- Configure "Parity" to None.
- Configure "Stop bits" to 1.
- Configure "Flow control" to None.
- Click "OK" button after entering the data in all the fields.

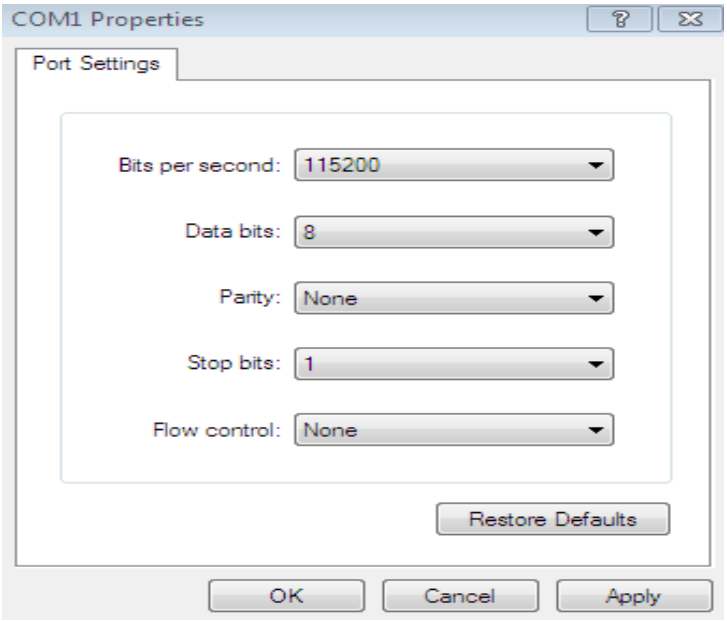

Figure 39: Hyper Terminal COM port Properties Configuration

- From the "File" Menu on Hyper Terminal select "Properties" as shown in figure 36.

Note: After new connection host is excepted to reboot the module.

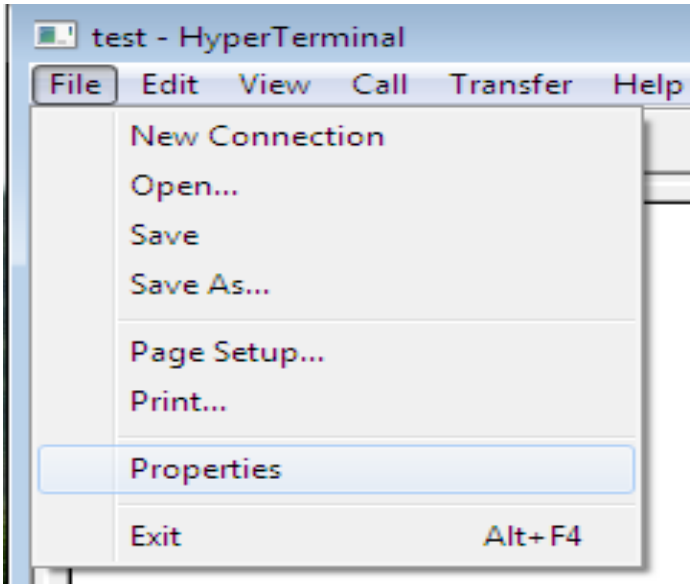

Figure 40: Hyper Terminal Properties Selection

- From the Hyper Terminal Properties dialog box, select "ASCII Setup" as shown in figure 37.

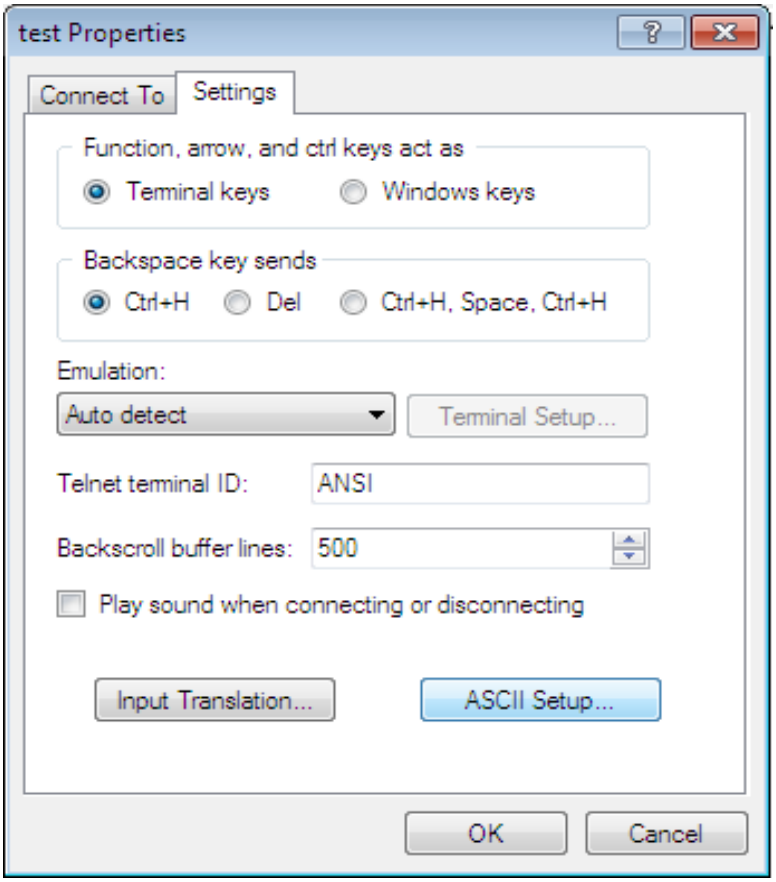

Figure 41: Hyper Terminal Properties Dialog Box

In the ASCII Setup dialog box as shown in figure 38, select the following options.

- Select option "Send line ends with line feeds".
- Select option "Echo typed characters locally".
- Select option "Wrap lines that exceed terminal width".
- Click "OK" button after selecting above options.

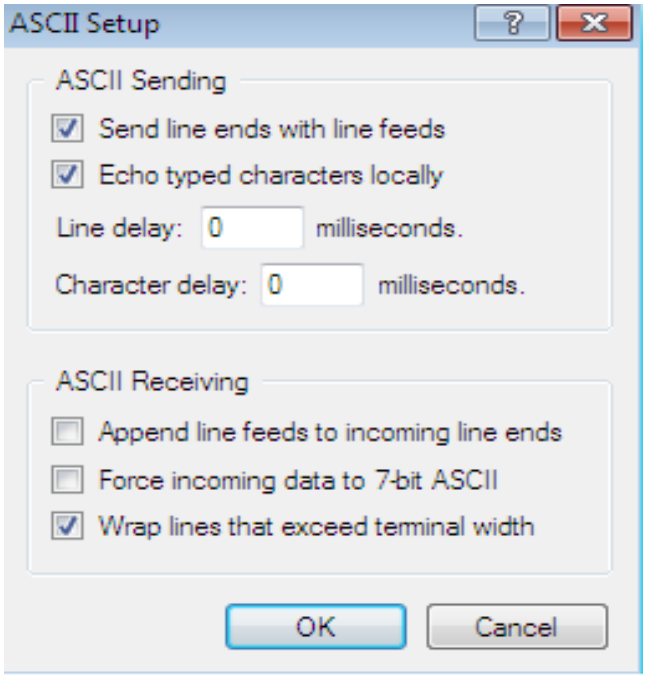

Figure 42: Hyper Terminal ASCII Configuration

10.2 Tera Term Configuration

Teraterm can also be used. The steps for configuring Teraterm are shown below.

- Open Tera Term application

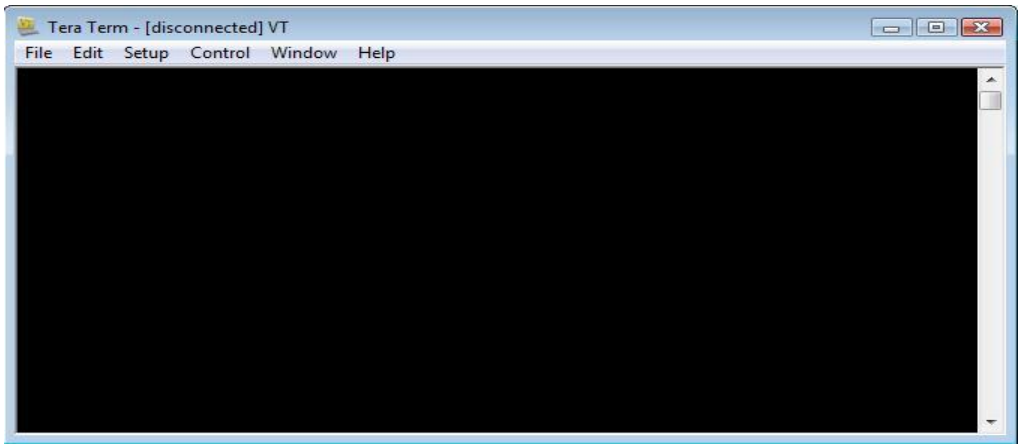

Figure 43: Tera Term VT

- From the "Setup" menu select "Serial port" as shown.

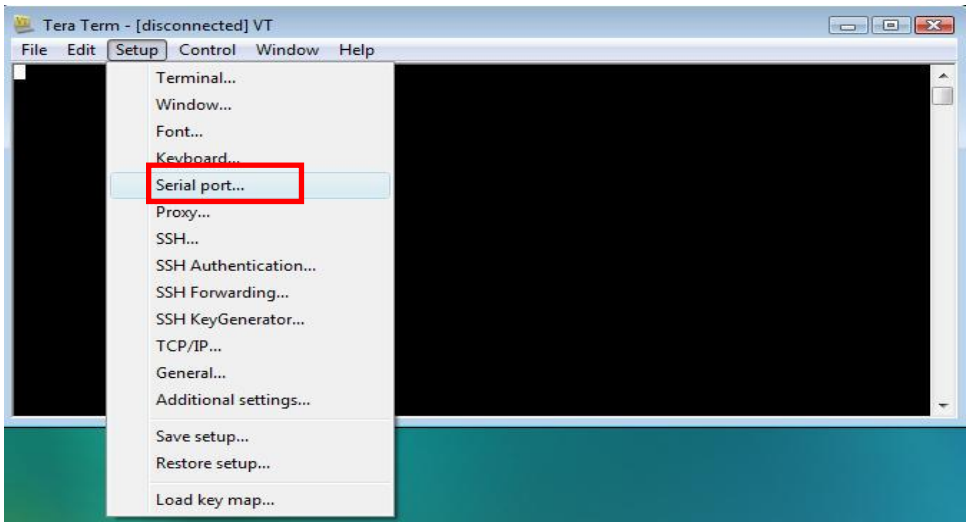

Figure 44: Tera Term Serial port Selection

In the Tera Term Serial port dialog box, configure different fields as shown below.

- Configure "Port" to COM1.
- Configure "Baud rate" to 115200.
- Configure "Data" to 8 bit.
- Configure "Parity" to none.
- Configure "Stop" to 1bit.
- Configure "Flow control" to none.
- Configure "Transmit delay" to 0 msec/char and 0 msec/line.
- Click "OK" button after entering the data in all the fields.

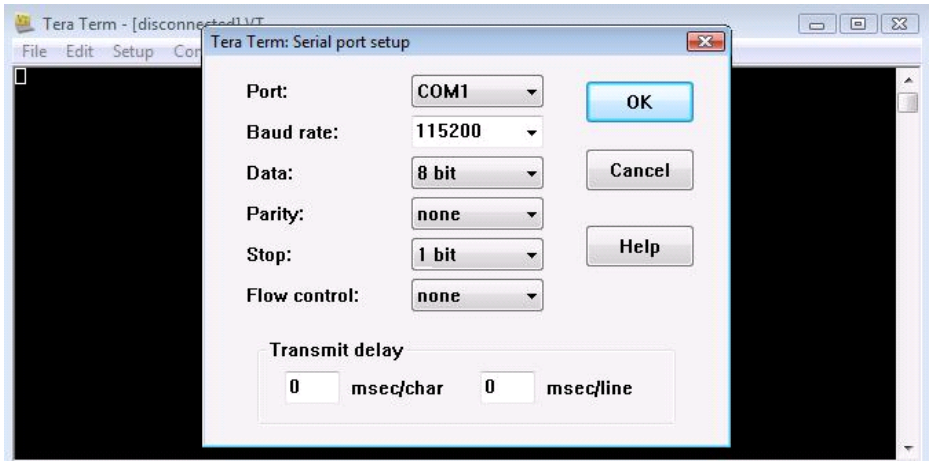

Figure 45: Tera Term Serial port Configuration

- From the "Setup" menu select "Terminal" as shown.

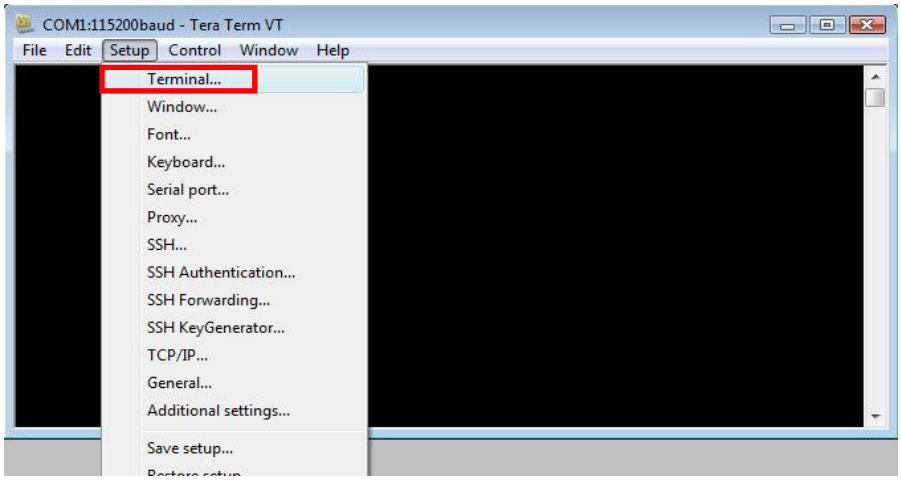

Figure 46: Tera Term Terminal Selection

In the Tera Term Terminal setup dialog box as shown, select the following options.

- Configure "Terminal size" to 80x24.
- Select option "Echo typed characters locally".
- Select option "Wrap lines that exceed terminal width".

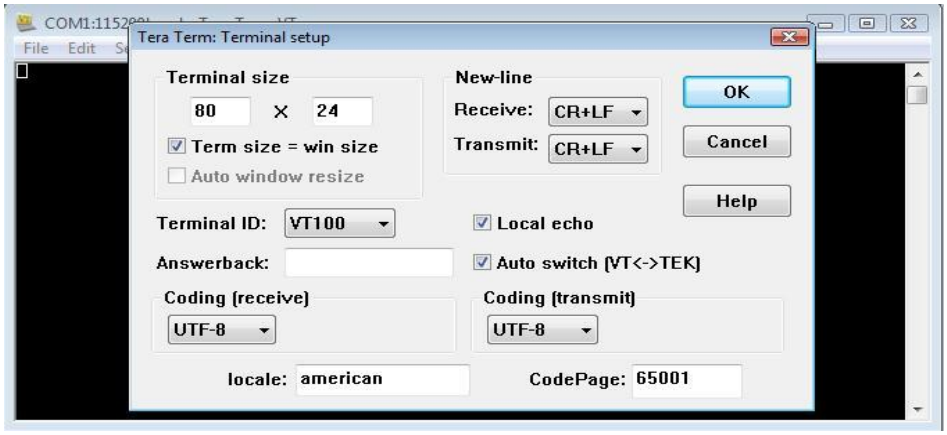

Figure 47: Tera Term Terminal Configuration

10.3 Serial Port Monitor Configuration

The actual bytes exchanged between the module and the Host PC may not all be readable on HyperTerminal or Tera Term because of ASCII conversion. A serial port monitor[3] can be used to see the actual bytes. A free package is available at http://www.serial-port-monitor.com/ , while other similar software such as Hterm, Docklight etc. also provides good interfaces to simultaneously view ASCII and actual bytes exchanged in the UART interface. The following sections assume that HHD Free Serial Port Monitor has been installed from the above link.

Open the "HHD Free Serial Port Monitor" before opening the HyperTerminal or Tera Term.

- From the "File" Menu on Serial Port Monitor select "New Session" as shown.

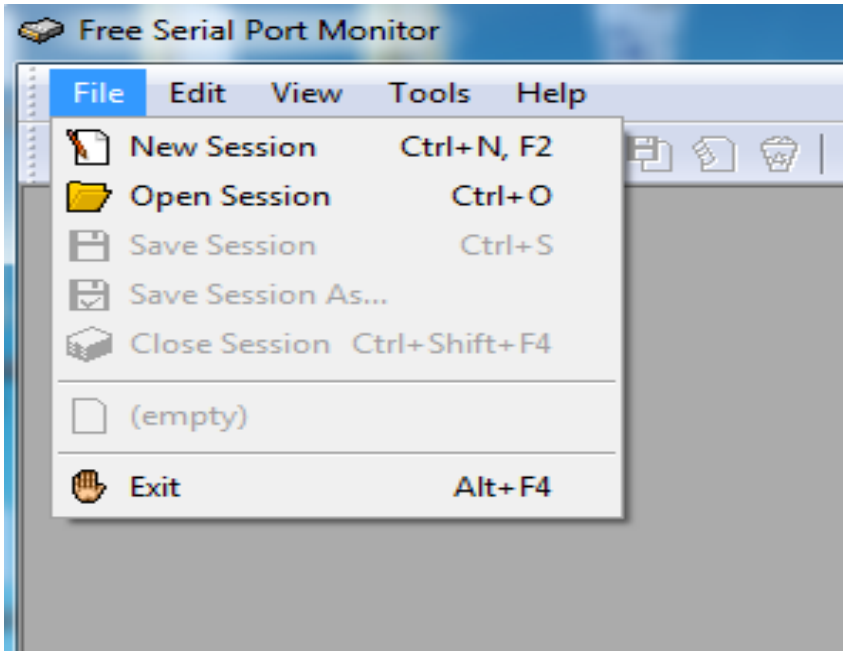

Figure 48: Serial Port Monitor

- Select "Serial Port Monitor" as Session Type as shown.

[3] Redpine Signals Inc. assumes no liability for damages of any kind resulting from use of third party software.

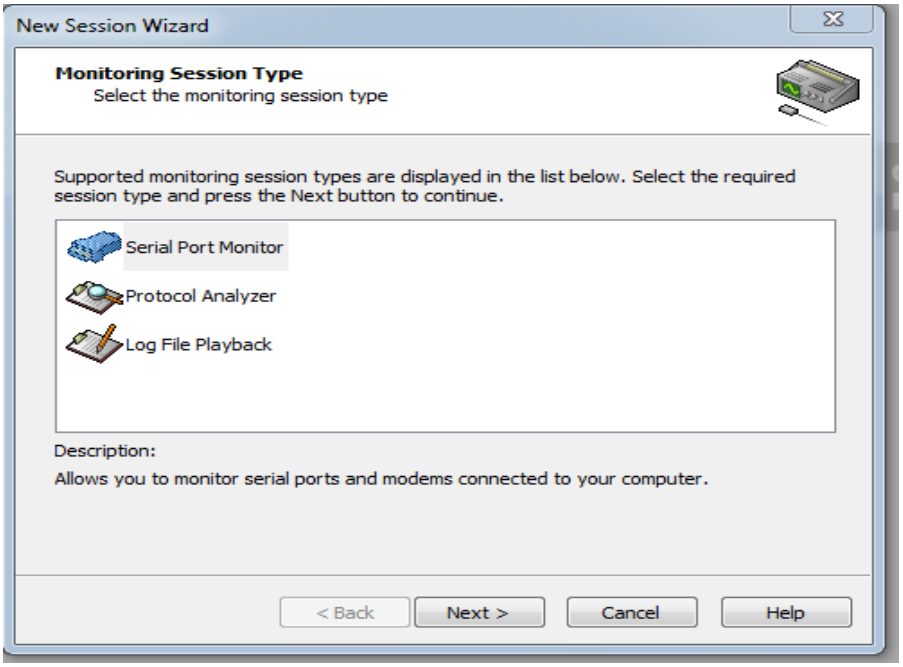

Figure 49: Serial Port Monitor Session Type Selection

- Select available serial port for Serial device name. In the figure below "COM6" is selected as Serial device name.

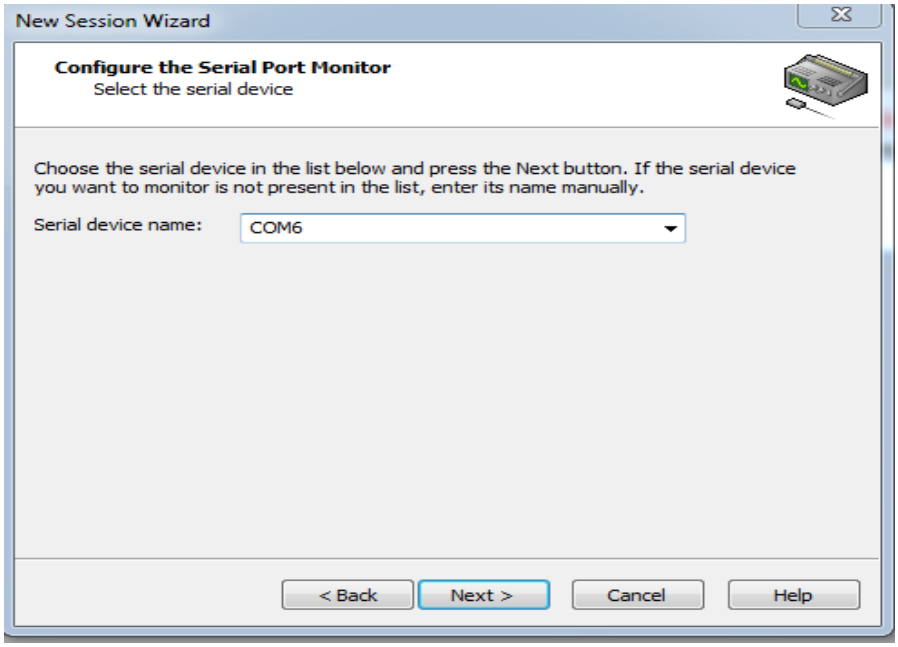

Figure 50: Serial Port Monitor Serial Device Selection

- Select option "Request View" as data visualizer as shown.

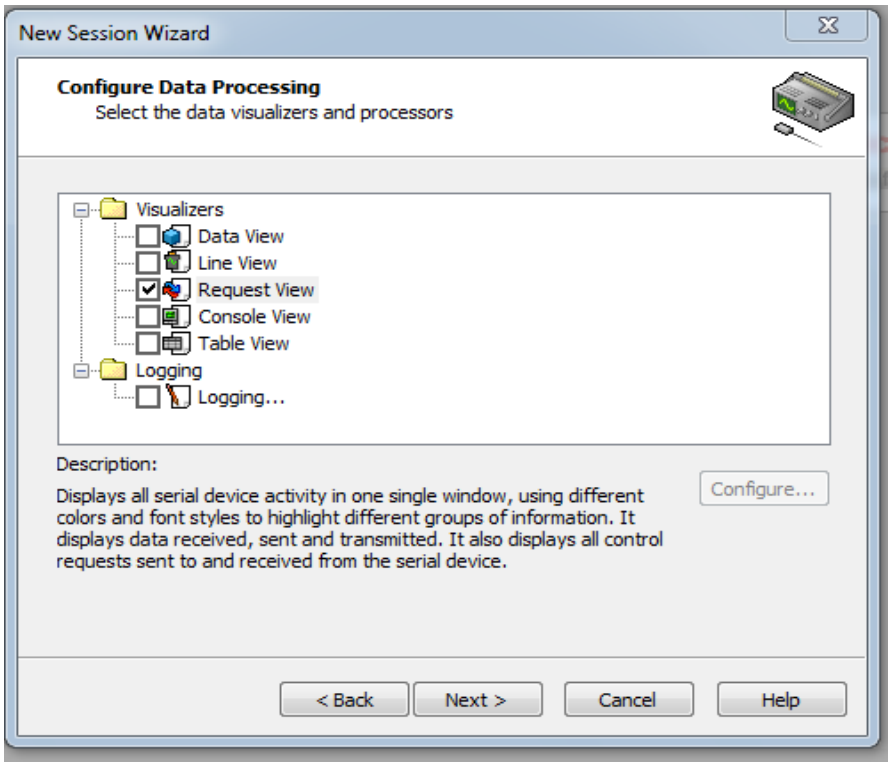

Figure 51: Serial Port Monitor Data Visualizer Configuration

All bytes exchanged through the UART interface will now be visible on the monitor.

Note: The Serial Port Monitor mentioned in the document works reliably in Windows XP machines

11 Appendix B: Radius Server Configuration

To evaluate any of the Enterprise Security modes, a Radius server should be running on a Laptop/PC (node D in Section 4.2.1) connected to the Access Point. A Linux compatible radius server package is downloadable from http://freeradius.org. Installation instructions can be found in the documentation included in the package. Additionally, the install process is described below.

Installation of the Radius Server on a Linux PC

If the user wants to use the sample radius server for evaluation, he can install it on a Linux PC (recommended Fedora 2.6.30).

1. Uncompress the package freeradius-server-2.1.12.tar in the desired location in the Linux PC. After uncompressing is done, change directory to the *freeradius-server-2.1.12* folder
2. Issue the following commands:

   ./configure

   make

   make install
3. The radius server is now installed in the PC (D). A folder "raddb" is created inside /usr/local/etc. Replace this folder with the raddb folder provided inside Redpine's software package RS9113.WSC.GENR.x.x.x.x.x.x\ Resources\Applications\Radius_server\
4. To start the Radius Server, issue the below command on the Linux terminal

   radiusd –X

NOTE: The user can use other Radius Server software also for evaluation. This radius server is provided for reference.

Important files for the Radius Server:

1. *RS9113.WSC.GENR.x.x.x.x.x.x\Resources\Applications\Radius_server \raddb\certs\ wifiuser.pem*

   This is the default certificate file provided with the software package. The file used for the parameter < *certificate*> in the command **at+rsi_cert**, when EAP-TLS mode is used in the module. This certificate file should be present in PC (P).

NOTE: To generate a new certificate, the below process may be used in the Linux PC where *freeradius-server* was installed.

Create Normal Certificate

```
mkdir new_certs
cd new_certs/
mkdir sslcert
chmod 0700 sslcert
cd sslcert
mkdir certs private
echo '100001' >serial
touch certindex.txt
vi openssl.cnf

/* CA root */
openssl req –new –x509 –extensions v3_ca –keyout
private/cakey.pem –out cacert.pem –days 365 –config ./openssl.cnf

/* Certificates Request */
openssl req –new –nodes –out redpine-req.pem –keyout
private/redpine-key.pem –days 365 –config ./openssl.cnf


<openssl.cnf is the configuration file used to generate the certificate.
A sample file is provided at
RS9113.WSC.GENR.x.x.x.x.x.x\Resources\Applications\Radius_server
\openssl.cnf>


/* Signing the certificates with ca root certificate generated in section
CA root */
openssl ca –out redpine-cert.pem –days 365 –config ./openssl.cnf –
infiles redpine-req.pem

Finally concatenate the redpine-key.pem, redpine-cert.pem
cacert.pem
cat redpine-key.pem >> redpine-cert.pem  >> cacert.pem.

File redpine-key.pem is the new certificate.


Create Encrypted Certificate

mkdir new_certs

cd new_certs/

mkdir sslcert

chmod 0700 sslcert

cd sslcert

mkdir certs private

echo '100001' >serial

touch certindex.txt
```

```
vi openssl.cnf

/* CA root */

openssl req -new -x509 -extensions v3_ca -keyout private/cakey.pem
-out cacert.pem -days 365

-config ./openssl.cnf

/*Encrypt Private Key - use this encrypted key (private/cakey1.pem )
in openssl.cnf*/

openssl pkcs8 -in private/cakey.pem -topk8 -out private/cakey1.pem

/* Certificates Request */

openssl req -new -nodes -out redpine-req.pem -keyout
private/redpine-key.pem -days 365 -config

./openssl.cnf

/* Signing the certificates with ca root certificate generated in section
CA root */

openssl ca -out redpine-cert.pem -days 365 -config ./openssl.cnf -
infiles redpine-req.pem /*Encrypt Key*/

openssl pkcs8 -in private/redpine-key.pem -topk8 -out
private/redpine-key1.pem

/*Finally concatenate the redpine-key1.pem, redpine-cert.pem
cacert.pem in the order */

cat redpine-key1.pem redpine-cert.pem cacert.pem > cert.pem
```

2. *RS9113.WSC.GENR.x.x.x.x.x.x\Resources\Applications\Radius_server \raddb\users*

   This file contains the user id and password, to be used in <*user_identity*> and <*password*> fields of the command **at+rsi_eap**

3. *RS9113.WSC.GENR.x.x.x.x.x.x\Resources\Applications\Radius_server \raddb\eap.conf*

   This file is used to select the EAP methods. The password "wifi" (<private_key_password = wifi> inside the file) should be used for the field <*key_password*> in the command **at+rsi_cert**.

12 Appendix C: Access Point Configuration

12.1 Personal Security AP Settings

A snapshot is shown for AP Settings to work in Personal Security mode. A Linksys AP (model no. WRT400N) is used for illustration.

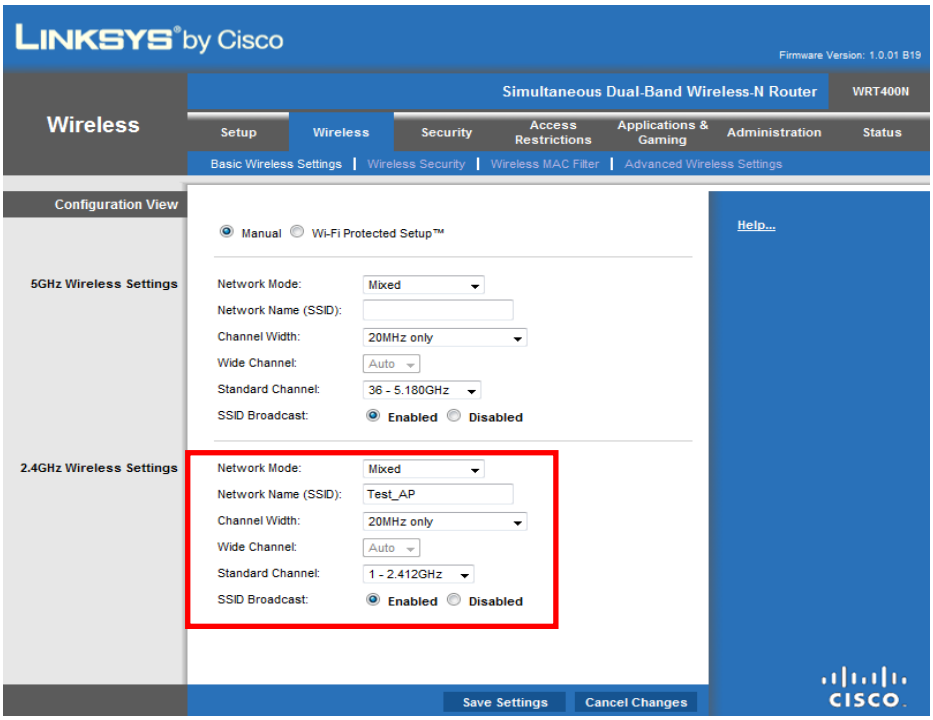

Figure 52: Access Point Wireless Settings

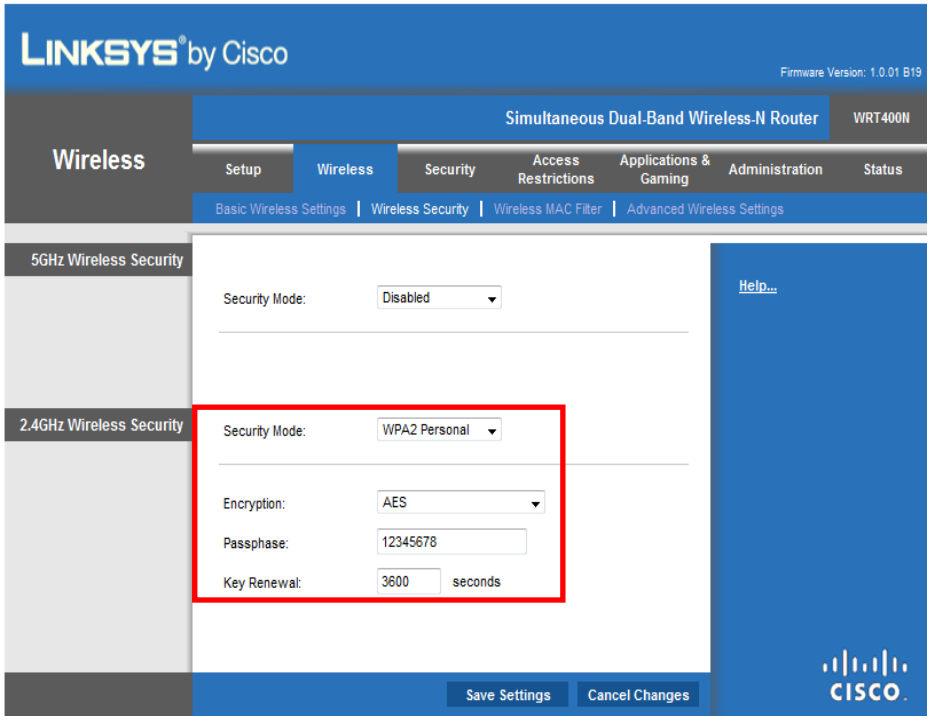

Figure 53: Access Point Settings -Personal Security Mode

12.2 Enterprise Security AP Settings

A snapshot is shown for AP Settings to work in Enterprise Security mode. A Linksys AP (model no. WRT400N) is used for illustration.

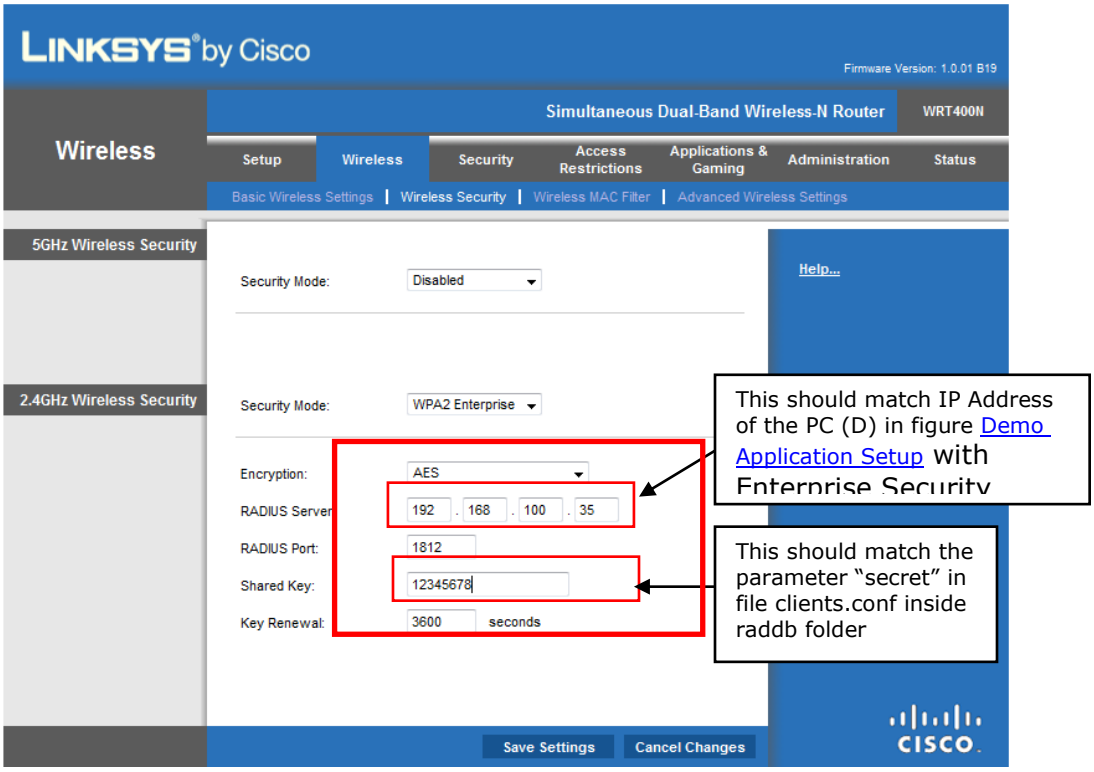

Figure 54: Access Point Settings - Enterprise Security Mode

13 Appendix D: TCP Command Configuration

Usage of PC applications:

TCP.exe <s for server> <lport> <dipaddr> <dport>

TCP.exe <c for client> <lport> <dipaddr> <dport>

s – to open a server TCP socket

c – to open a client TCP socket

lport – Local port number

dipaddr – IP address of the destination

dport – Port number of the destination

UDP.exe is also used in the same way.

Revision History:

Rev.	Ver. No.	Date	Changes
1.	1.7	June	Updated the sections 3.4,5.6 and section7
2.	1.8	August	1.Removed BT section from the EVB user Guide 2.Updated the example sections 3.Removed the SDIO information
3.	1.9	December	1.Section 3:Bootloading the module: updated with relevant changes to support EVB firmware versions1.0.10 and above.
4.	1.10	December	1. Removed information related toImage22. Added Reference documents about BT, BTLE and Zigbee.