



**RN1810 WiFly
Command Reference
User's Guide**

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

**QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
= ISO/TS 16949 =**

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, KeeLoq logo, Klear, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, RightTouch logo, REAL ICE, Ripple Blocker, Serial Quad I/O, SQL, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2016, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-0362-3

Table of Contents

Preface	5
Chapter 1. Overview	
1.1 Introduction	9
1.2 Features	9
1.3 Configuration	10
Chapter 2. Command Reference	
2.1 Introduction	11
2.2 Command Syntax	11
2.3 Command Organization	11
2.4 Set Commands	12
2.5 Get Commands	28
2.6 Action Commands	30
2.7 Show Commands	35
Chapter 3. Advanced Features and Settings	
3.1 Introduction	37
3.2 Soft Access Point (AP) Mode	37
3.3 Sleep and Wake Methods	38
3.4 TX Flush Methods	38
3.5 Scan Output Format	39
Chapter 4. Application Examples	
4.1 Joining Networks and Making Connections	41
4.2 Sending Data Using TCP - Module is a TCP Server	42
4.3 Sending Data Using TCP - Module is a TCP Client	43
4.4 Sending Data Using IPv6 - Module is a TCP Client	43
4.5 Sending Data Using UDP - Module is a UDP Client	44
4.6 Sending Data Using TLS - Module is a TLS Server	44
4.7 Sending Data Using TLS - Module is a TLS Client	45
4.8 FTP Client	46
4.9 Wi-Fi Protected Setup (WPS)	47
4.10 SNTP Client	49
4.11 Configuration Web Server	50
4.12 Using the Web Server to Configure the RN1810	51
4.13 Auto-Connection and Sleep Timers	52
Chapter 5. RN1810 I/O Pins	
5.1 I/O PIN Descriptions	55
5.2 I/O Pin Function Select	58

Appendix A. Command Quick Reference Guide

 A.1 Default Configuration Settings 59

Worldwide Sales and Service63

Preface

NOTICE TO CUSTOMERS

All documentation becomes dated, and this manual is no exception. Microchip tools and documentation are constantly evolving to meet customer needs, so some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our web site (www.microchip.com) to obtain the latest documentation available.

Documents are identified with a “DS” number. This number is located on the bottom of each page, in front of the page number. The numbering convention for the DS number is “DSXXXXXXXXA”, where “XXXXXXXX” is the document number and “A” is the revision level of the document.

For the most up-to-date information on development tools, see the MPLAB® IDE online help. Select the Help menu, and then Topics to open a list of available online help files.

INTRODUCTION

This chapter contains general information that will be useful to know before using the RN1810 module. Items discussed in this chapter include:

- [Document Layout](#)
- [Conventions Used in this Guide](#)
- [Recommended Reading](#)
- [The Microchip Web Site](#)
- [Development Systems Customer Change Notification Service](#)
- [Customer Support](#)
- [Document Revision History](#)

DOCUMENT LAYOUT

This document provides information for configuring the RN1810 module, including a command reference, advanced features, and application examples. The document is organized as follows:

- **Chapter 1. “Overview”** – This chapter introduces the RN1810 module and provides a brief overview of various features.
- **Chapter 2. “Command Reference”** – This chapter provides information on the commands used to configure RN1810 module and gives examples.
- **Chapter 3. “Advanced Features and Settings”** – This chapter describes the WiFly module’s advanced features, including techniques to put the module to sleep and wake it up.
- **Chapter 4. “Application Examples”** – This chapter provides application examples in using the RN1810 module.
- **Chapter 5. “RN1810 I/O Pins”** – This chapter describes the RN1810 I/O pins.

- **Appendix A. “Command Quick Reference Guide”** – This appendix provides a quick reference of all configuration commands.

CONVENTIONS USED IN THIS GUIDE

This manual uses the following documentation conventions:

DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Arial font:		
Italic characters	Referenced books	<i>MPLAB IDE User's Guide</i>
	Emphasized text	...is the <i>only</i> compiler...
Initial caps	A window	the Output window
	A dialog	the Settings dialog
	A menu selection	select Enable Programmer
Quotes	A field name in a window or dialog	"Save project before build"
Underlined, italic text with right angle bracket	A menu path	<u><i>File>Save</i></u>
Bold characters	A dialog button	Click OK
	A tab	Click the Power tab
N'Rnnnn	A number in verilog format, where N is the total number of digits, R is the radix and n is a digit.	4'b0010, 2'hF1
Text in angle brackets < >	A key on the keyboard	Press <Enter>, <F1>
Courier New font:		
Plain Courier New	Sample source code	#define START
	Filenames	autoexec.bat
	File paths	c:\mcc18\h
	Keywords	_asm, _endasm, static
	Command-line options	-Opa+, -Opa-
	Bit values	0, 1
	Constants	0xFF, 'A'
Italic Courier New	A variable argument	<i>file.o</i> , where <i>file</i> can be any valid filename
Square brackets []	Optional arguments	mcc18 [options] <i>file</i> [options]
Curly brackets and pipe character: { }	Choice of mutually exclusive arguments; an OR selection	errorlevel {0 1}
Ellipses...	Replaces repeated text	var_name [, var_name...]
	Represents code supplied by user	void main (void) { ... }

RECOMMENDED READING

This user's guide describes how to configure the RN1810 module. Other useful documents are listed below. The following Microchip document(s) are recommended as supplemental reference resources.

RN1810/RN1810E 2.4 GHz IEEE 802.11b/g/n Wireless Module Data Sheet (DS50002460A)

This data sheet provides the technical specifications for the RN1810/RN1810E modules and is available for download from the Microchip website (www.microchip.com).

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events; and listings of Microchip sales offices, distributors and factory representatives

DEVELOPMENT SYSTEMS CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer The Development Systems product group categories are:

- **Compilers** – The latest information on Microchip C compilers and other language tools
- **Emulators** – The latest information on the Microchip MPLAB[®] REAL ICE[™] in-circuit emulator
- **In-Circuit Debuggers** – The latest information on the Microchip in-circuit debugger, MPLAB ICD 3
- **MPLAB X IDE** – The latest information on Microchip MPLAB X IDE, the Windows[®] Integrated Development Environment for development systems tools
- **Programmers** – The latest information on Microchip programmers including the PICKit[™] 3 development programmer

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at:

<http://www.microchip.com/support>.

DOCUMENT REVISION HISTORY

Revision A (March 2016)

This is the initial release of this document.

Chapter 1. Overview

1.1 INTRODUCTION

The RN1810 radio module is a complete, standalone embedded wireless LAN access device. The device has an on-board TCP/IP stack and applications, and the simplest hardware configuration requires only four pins: Power, TX, RX, and Ground. Once the initial configuration is performed, the device automatically accesses a Wi-Fi® network and sends/receives serial data.

1.2 FEATURES

- Fully qualified 2.4 GHz IEEE 802.11 b/g/n transceiver
- FCC, CE, IC certified, and RoHS compliant
- Ultra-Low-Power:
 - Intelligent, built-in power management with programmable wake-up
 - Accepts 3.3V power supply or 2 to 3V battery when using boost regulators
- Antenna Options:
 - On-board PCB Trace antenna and W.FL connector for external antenna
- Hardware:
 - Optional I/O pins for control and status
 - Real-time clock for wake-up; Auto-sleep and Auto-wake-up modes
- Network Support:
 - Supports Soft Access Point (AP) and Infrastructure networking modes
 - Push button WPS mode for easy network configuration
 - On-board TCP/IP stack
 - Over the air firmware upgrade (TFTP)
 - Secure Wi-Fi authentication via WEP, WPA-PSK (TKIP), and WPA2-PSK (AES)
 - Configuration over UART or wireless interfaces using simple ASCII commands
 - Built-in networking applications: Dynamic Host Configuration Protocol (DHCP) client, Domain Name Service (DNS) client, Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP) ping, FTP client, SNTP client, HTTP, User Datagram Protocol (UDP), and Transmission Control Protocol (TCP)
 - SSL Support (latest TLSv1.2)
 - IPv4 and IPv6 support

1.3 CONFIGURATION

The WiFly module has two primary modes of operation: Data mode and Command mode. In Data mode, the module can accept incoming connections, initiate outgoing connections, and act as a data pump. To configure parameters or view the current configuration, or both, the module must be placed into Command mode (also called as Configuration mode).

1.3.1 Entering and Exiting Command Mode

By default, the module is in Data mode after power-up. Sending the escape sequence of three dollar signs, `$$$`, causes the module to enter Command mode. Refer to [Section 2.6.1 “\\$\\$\\$”](#) for the timing restrictions in entering Command mode. The module replies with `CMD` to indicate it is in Command mode. Once in Command mode, the WiFly device can be configured using simple ASCII commands with each command ending with a carriage return `<cr>`. All valid commands return `AOK` and invalid commands return an `ERR` description. To exit Command mode, send `exit <cr>`. The module responds with `EXIT` to indicate that it has exited Command mode and entered Data mode.

Various parameters can be viewed, such as the SSID, channel, IP address, serial port, and other settings, which can be configured in Command mode. Commands must be sent to the module through the UART. When using the UART interface, the communications settings must match the WiFly module's stored settings. The default settings are 9600 baud, 8 bits, no parity, 1 Stop bit, and Hardware Flow Control disabled. Command mode can be entered locally over the UART interface at any time regardless of an active TCP connection.

Note: Microchip suggests using either the TeraTerm (Windows OS) or minicom (Ubuntu) terminal emulator program.
--

When the WiFly module powers up, it attempts to auto-associate with the access point stored in its configuration settings if the auto-join feature is enabled. The auto-join feature is disabled by default. Enable it using the ASCII command `set wlan join 1`.

Disable the auto-join feature (default behavior) using the `set wlan join 0` command.

Chapter 2. Command Reference

2.1 INTRODUCTION

WiFly modules support a variety of commands for configuration. This section describes these commands and provides examples.

2.2 COMMAND SYNTAX

To issue commands to the module, send a keyword followed by optional parameters. Apply the following syntax rules:

- Commands are case sensitive
- Commands must be less than 120 characters
- Spaces *cannot* be used in parameters – must be replaced by a character (default is \$)
- String text data, such as the SSID, is case sensitive
- Shorthand can be used for the parameters. For example, the following commands are equivalent:

```
- set uart baudrate 115200
- set uart b 115200
- set u b 115200
```

Note: Shorthand *cannot be used* for command keywords. For example, `s uart baudrate 115200` is invalid. There are some commands that *cannot* be abbreviated; these exceptions are noted in the command description.

- Type numbers in decimal or hexadecimal. For example 115200 or 0x7e.

2.3 COMMAND ORGANIZATION

There are four general command categories as shown in [Table 2-1](#).

TABLE 2-1: COMMAND TYPES

Command Type	Section	Description
Set	Section 2.4“Set Commands”	Set commands take effect immediately and are stored to memory when the <code>save</code> command is issued.
Get	Section 2.5“Get Commands”	Get commands retrieve and display the stored information.
Action	Section 2.6“Action Commands”	Action commands perform actions such as scanning, connecting, disconnecting, and so on.
Show	Section 2.7“Show Commands”	Show commands retrieve and display the current state.

Note: Any changes must be saved using the `save` command or the module loads the previous settings upon reboot or power-up.

When the module boots, all configuration data is loaded into RAM variables from FLASH. The Set commands only modify the RAM copy of the system variables. In general, the IP, WLAN, and UART settings require a save and reboot before taking effect. Most of the other commands, such as the COMM settings and timers, take effect immediately allowing the user to change parameters on-the-fly, minimizing power usage, and saving Flash rewrite cycles.

Once the configuration is complete, save the settings to store the configuration data.

2.4 SET COMMANDS

Table 2-2 summarizes the Set command parameters.

TABLE 2-2: SET COMMANDS

Parameter	Description
apmode	Controls the Soft AP parameters.
comm	Sets the communication and data transfer, timers, and matching characters.
dhcp	Sets the DHCP host name.
dns	Sets the DNS host and domain.
ftp	Sets the FTP host address and login information.
ip	Specifies the IP settings.
opt	Sets system options
sys	Sets system settings such as sleep and wake timers.
time	Sets the timer server settings.
uart	Specifies the serial port settings such as baud rate and parity.
wlan	Sets the wireless interface settings, such as SSID, channel, and security options.

2.4.1 set apmode beacon <value>

This command sets the Soft AP beacon interval in milliseconds, where <value> is a decimal number from 0 to 65,436.

Default: 102

Example:

```
set apmode beacon 120 // Beacons are sent every 120 ms
```

2.4.2 set apmode channel <value>

This command sets the Soft AP channel number where <value> is a decimal number from 1 to 11.

Default: 1

Example:

```
set apmode channel 6 // Set channel number to 6
```

2.4.3 set apmode passphrase <string>

This command sets the Soft AP mode passphrase used for WPA2-AES encryption. When set, the module broadcasts a network in Soft AP mode with WPA2-AES encryption enabled. The <string> length must be between 8 and 64 characters.

Default: NULL

Example:

```
set apmode passphrase my_passphrase // Sets the passphrase to
                                     // my_passphrase
```

2.4.4 set apmode ssid <string>

This command sets the Soft AP mode network name (SSID) to be broadcast where <string> is the SSID. The maximum length of the SSID is 32 characters.

Default: NULL

Example:

```
set apmode ssid my_network // Sets the Soft AP network name to
                           // "my_network"
```

2.4.5 set comm \$ <char>

This command sets the character used to enter Command mode to <char>. Use this setting when \$\$\$ (the default string used to enter Command mode) is a possible data string. After saving this setting, upon every subsequent reboot, the module ignores \$\$\$ and looks for <char><char><char> to enter Command mode.

Default: \$

Example:

```
set comm $ w // Sets Command mode character to 'w'
```

2.4.6 set comm close <string>

This command sets the ASCII string that is sent to the host UART when the TCP port is closed. The <string> is between 1 and 32 characters. If the output requires no string, set <string> to 0.

Default: "CLOS"

Example:

```
set comm close port_closed // Set the string to "port_closed"
set comm close 0           // Do not send any string upon closing a
                           // TCP connection
```

2.4.7 set comm idle <value>

This command closes an idle TCP connection (no data activity) after <value> seconds. A value of 0 means the module is *not* disconnected when the connection goes idle. For the new value to take effect, the `save` and `reboot` commands must be issued after this command.

Default: 0

Example:

```
set comm idle 5 // Close TCP connection when idle for 5 seconds
set comm idle 0 // Do not disconnect when TCP connection idle
```

2.4.8 `set comm match <value> <flag>`

This command sets the match character where *<value>* is a decimal number from 0 to 127 or a hex number from 0x00 to 0x7F. The parameter *<flag>* is either '0' or '1', where '0' excludes the match character in the datastream and '1' includes the match character in the datastream.

Upon receiving the match character in the datastream, WiFly flushes its UART RX queue and sends the data to the Wi-Fi network in an IP packet. The match character itself is optionally sent in the datastream based on the value of the *<flag>*.

Setting the match character to '0' disables the Match method (*cannot* use '0' as a match character).

When the Match method is selected, the Size method is automatically active as a 'fall-back' flush method. The algorithm is:

```
IF match character received AND including match character
    Send bytes in UART RX buffer to Wi-Fi network followed by match character
ELSE IF match character received AND not including match character
    Send bytes in UART RX buffer to Wi-Fi network
ELSE IF number of UART RX bytes equals Size
    Send bytes in UART RX buffer to Wi-Fi network
ENDIF
```

In short, when the Match method is active, the UART RX bytes are transmitted when the match character is received or size bytes are received, whichever occurs first.

If the Match method is active, and the Match value is set to '0' during runtime, then the Match method is disabled and the WiFly module automatically reverts to the Size method. For more information, refer to [Section 3.4 "TX Flush Methods"](#).

Default: 0 (Match method disabled)

Example:

```
set comm match 0x1b 0 // Flush UART RX buffer when a 0x1b is received,
                      // but do not send the match character in the
                      // data stream
set comm match 0x1b 1 // Flush UART RX buffer when a 0x1b is received
                      // and send the match character in the datastream
set comm match 25 0   // Flush UART RX buffer when a 25 is received, but
                      // do not send the match character in the data stream
set comm match 0 0    // Disables the Match method and goes back to the
                      // size flush method
```

2.4.9 `set comm open <string>`

This command sets the ASCII string that is sent to the host UART when a TCP port is opened, where *<string>* is between 1 and 32 characters. If the output requires no string, set *<string>* to 0.

Default: hello

Example:

```
set comm open TCP_OPEN // Send "TCP_OPEN" to host UART upon opening a
                       // TCP connection
set comm open 0         // Do not send any string upon opening a
                       // TCP connection
```

2.4.10 set comm remote <string>

This command sets the ASCII string that is sent to the remote Host when the TCP connection is opened, where <string> is one or more characters up to a maximum of 64. If no string must be sent, use a zero (0) as the <string> parameter.

Default: *HELLO*

Example:

```
set comm remote XYZ // Send "XYZ" to remote Host upon opening a
                    // TCP connection
set comm remote 0   // Do not send any string upon opening a
                    // TCP connection
```

2.4.11 set comm size <value>

This command sets the number of UART RX data bytes at which WiFly flushes its UART RX queue, and sends the data to the Wi-Fi network in an IP packet where <value> is between 1 and 1420 bytes.

If the Match method is disabled (Refer to `set comm match <value> <flag>` command), this method is exclusive – WiFly only flushes the UART RX queue when the specified number of data bytes have been received from the Host. If the Match method is active, this becomes the 'fallback' size where the UART buffer is flushed.

Refer to [Section 3.4“TX Flush Methods”](#) for more information.

Default: 1420

Example:

```
set comm size 1000 // Set the flush size to 1000 bytes
```

2.4.12 set comm timer <value>

This command sets the time period, in milliseconds, when WiFly flushes its UART RX queue and sends the data to the Wi-Fi network in an IP packet. This is the default mode for WiFly. This is an exclusive flush method – WiFly only flushes the UART RX queue at the specified time period. For the new value to take effect, the `save` and `reboot` commands must be issued after this command.

Refer to [Section 3.4“TX Flush Methods”](#) for more information.

Default: 100

Example:

```
set comm timer 150 // Set the flush time period to 150 ms
```

2.4.13 set dhcp hostname <string>

This command sets the host name for the RN1810 module. The host name string can be from 1 to 31 characters.

Default: RN1810_xy (where x and y are the last two bytes of the modules MAC address)

Example:

```
set dhcp hostname My_RN1810
```

2.4.14 `set dhcp lease <start_ip_address> <end_ip_address> <lease_time>`

This command sets the DHCP pool and lease time when the RN1810 is put in Soft AP mode. The parameter `<start_ip_address>` is the starting IP address of the DHCP pool. The parameter `<end_ip_address>` is the ending IP address in the DHCP pool. The parameter `<lease_time>` is the number of seconds a supplied DHCP address can be used before it must be renewed (in seconds).

Note: The `set dhcp lease` command cannot be called until after the `apmode <ssid> <channel>` command is called.

Default: `<start_ip_address>` is 192.168.1.11
`<end_ip_address>` is 192.168.1.20
`<lease_time>` is 86400 seconds

Example:

```
set dhcp lease          // Sets the Soft AP DHCP pool addresses
192.168.1.1 192.168.1.10 from 192.168.1.1 to 192.168.1.10 with a
40000                  lease time of 40000 seconds
```

2.4.15 `set dns address <address>`

This command sets the IP address of the DNS server, where `<address>` is an IP address in the form:

`<value>.<value>.<value>.<value>`

with `<value>` being a number between 0 and 255. This address is automatically set when using DHCP; set the DNS IP address for static IP or automatic IP modes.

Default: 0.0.0.0

Example:

```
set dns address 169.64.1.1 // Set the DNS server address to 169.64.1.1
```

2.4.16 `set dns name <string>`

This command sets the name of the Host for TCP/IP connections to `<string>`, where `<string>` is up to 32 characters.

Default: server1

Example:

```
set dns name mchp1      // Set the DNS host name to mchp1
```

2.4.17 **set ftp addr** <address>

This command sets the FTP servers' IP address, where <address> is an IP address in the form:

- For IPv4: <value>.<value>.<value>.<value> with <value> being a number between 0 and 255.
- For IPv6: xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx (x is a hexadecimal value)

Refer to [Section 4.8“FTP Client”](#) for an FTP example. It applies to all `set ftp ...` commands.

Default: 0.0.0.0

Example:

```
set ftp addr 192.168.1.176 // Set FTP server IP address to 192.168.1.176
```

2.4.18 **set ftp dir** <string>

This command sets the starting directory on the FTP server where <string> is a maximum of 64 characters. To designate the root directory, use the backslash (\) or period (.). Backslashes must be used to specify sub-folders.

Default: .

Example:

```
set ftp dir demo           // Set FTP server starting directory to demo
set ftp dir demo\test      // Set FTP server starting directory to demo\test
set ftp dir .              // Set FTP server starting directory to the
                           // root directory
set ftp dir \              // Set FTP server starting directory to the
                           // root directory
```

2.4.19 **set ftp filename** <string>

This command sets the name of the file that is transferred when the `ftp put` or `ftp get` command is issued. The file name can have a maximum length of 32 characters.

Default: test_file

Example:

```
set ftp filename config_data // Set the file name to config_data
```

2.4.20 **set ftp password** <string>

This command sets the login password for the FTP server, where <string> is up to 16 characters.

Default: Pass123

Example:

```
set ftp password MySecretPassword // Set the FTP server password to
                                   // "MySecretPassword"
```

2.4.21 **set ftp remote** <value>

This command sets the FTP server's port number, where <value> is the port number.

Default: 21

Example:

```
set ftp remote 21 // Set the FTP server port number to 21
```

2.4.22 set ftp timeout <value>

This command sets the FTP connection timeout value, where <value> is the timeout in seconds. The value must be one or more seconds.

Default: 10 seconds

Example:

```
set ftp timeout 20 // Set the FTP connection timeout to 20 seconds
```

2.4.23 set ftp user <string>

This command sets the login name for the FTP server, where <string> is up to 16 characters.

Default: mchp

Example:

```
set ftp user john // Set the FTP user login name to john
```

2.4.24 set ip address <address>

This command sets WiFly's static IP address, where <address> is an IP address in the form:

- For IPv4: <value>.<value>.<value>.<value> with <value> being a number between 0 and 255.
- For IPv6: xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx (x is a hexadecimal value)

Default: 0.0.0.0 (IPv4) 0::0 (IPv6)

Example:

```
set ip address 137.57.1.1 // Set IPv4 address
set ip address 2001:db9::d0ff:474a:3798:2294 // Set IPv6 address
```

2.4.25 set ip dhcp <value>

This command enables/disables DHCP mode, where <value> is a decimal number as shown in [Table 2-3](#). If this parameter is set, the module requests and sets the IP address, gateway, netmask, and DNS server upon association with an access point. Any previously set IP information is overwritten.

TABLE 2-3: DHCP MODES

Mode	Description
0	Turn DHCP off. The module uses its stored static IP address and gateway address.
1	Turn DHCP on. The module attempts to obtain an IP address and gateway from the access point.

Default: 1

Example:

```
set ip dhcp 0 // Disable DHCP client
```

2.4.26 set ip host <address>

This command sets the remote host's IP address, where <address> is an IP address in the form:

- For IPv4: <value>.<value>.<value>.<value> with <value> being a number between 0 and 255.
- For IPv6: xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx (x is a hexadecimal value).

Default: 0.0.0.0 (IPv4) 0::0 (IPv6)

Example:

```
set ip host 137.57.1.1           // Set IPv4 address
set ip host 2001:db9::d0ff:474a:3798:2294 // Set IPv6 address
```

2.4.27 set ip localport <value>

This command sets the local port number, where <value> is a decimal number representing the port.

Default: 2000

Example:

```
set ip localport 1025           // Set local port to 1025
```

2.4.28 set ip netmask <address>

This command sets the network mask, where <address> is an IP address in the form <value>.<value>.<value>.<value> with <value> being a number between 0 and 255. If DHCP is turned on, the netmask is assigned and overwritten when the module associates with the access point.

Default: 255.255.255.0

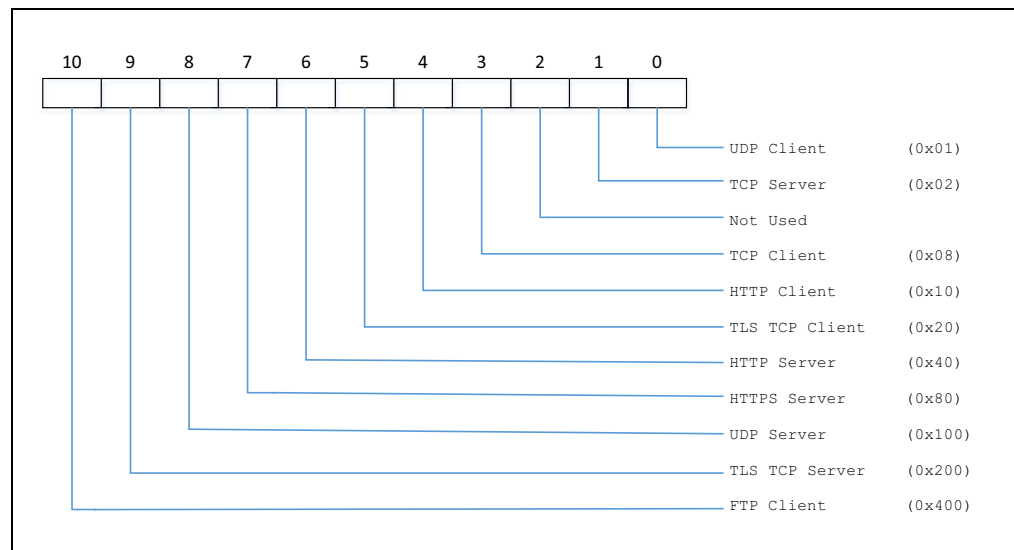
Example:

```
set ip netmask 255.255.0.0      // Sets the netmask to 255.255.0.0
```

2.4.29 set ip protocol <flag>

This command sets the IP protocol, where <flag> is a bit-mapped value described in [Figure 2-1](#). If none of the bits is selected, then no protocol is selected.

FIGURE 2-1: IP BITMAP VALUES



Default: 0x02 (TCP server)

Example:

```
set ip protocol 0x12      // Enable HTTP Client mode and TCP server
set ip protocol 0x20      // Enable TLS TCP Client mode
set ip protocol 0x400     // Enable FTP Client mode
```

2.4.30 set ip remote <value>

This command sets the remote host port number, where <value> is a decimal number representing the port.

Default: 0

Example:

```
set ip remote 1025        // Sets the remote IP host port to 1025
```

2.4.31 set ip version <value>

This command sets the version of IP used:

0: IPv4

1: IPv6

Default: 0 (IPv4)

Example:

```
set ip version 1          // Set IP to IPv6
```

2.4.32 set opt replace <value>

This command sets the replacement character for space characters in the SSID or passphrase where <value> is the replacement character.

Default: \$

Example:

```
set opt replace *         // Set space replacement character to '*'
```

2.4.33 set sys auto <value>

This command sets the HTTP client auto-connect timer, where <value> is the number of seconds. For the new value to take effect, the `save` and `reboot` commands must be issued after this command.

Note: The token “auto” <i>cannot</i> be abbreviated.
--

Default: 0

Example:

```
set sys auto 10           // Set auto-connect timer to 10 seconds
```

2.4.34 set sys autoconn <value>

This command causes the RN1810 to connect to a Host periodically, where <value> controls how often to connect to the remote Host as shown in [Table 2-4](#). This command only pertains to a TCP client. For the new value to take effect, the `save` and `reboot` commands must be issued after this command.

TABLE 2-4: AUTO-CONNECT TIMER SETTING

Value	Description
0	Disable the auto-connect timer.
1	Connect to remote host immediately upon power-up or awaking from Sleep mode.
2:254	Connect to a remote host every <value> seconds.
255	Connect to a remote host immediately upon power-up or when awaking from Sleep mode and go back to sleep immediately upon closing the TCP connection. Note: Be aware that when a client goes to sleep immediately after losing the connection, the transaction has a higher chance of <i>not</i> succeeding. In which case, the server thinks that the client is still connected. If using an RN1810 server, which only supports a single connection, a client reconnecting is unable to do so until the server times out and closes the previous connection.

Note: The token “autoconn” *cannot* be abbreviated.

Default: 0

Example:

```
set sys autoconn 1           // TCP client will connect immediately upon
                             // Reset or awaking
```

2.4.35 set sys autosleep <value>

This command sets the UDP auto-sleep timer where <value> is a multiplier used in conjunction with `set comm timer <value>`. For example, the following two commands creates a UDP sleep timer of 400 ms:

```
set comm time 100
set sys autosleep 4
```

The resulting timer is $100 * 4$, or 400 ms.

For the new value to take effect, the `save` and `reboot` commands must be issued after this command.

Note: The token “autosleep” *cannot* be abbreviated.

Default: 0

Example:

```
set sys autosleep 4
```

2.4.36 `set sys sleep <value>`

This command specifies the duration in seconds the RN1810 waits after a TCP connection is closed before it goes to sleep. For the new value to take effect, the `save` and `reboot` commands must be issued after this command.

Default: 0 (disabled)

Example:

```
set sys sleep 10                // Module goes to sleep 10 seconds after
                                // TCP connection closes
```

2.4.37 `set sys wake <value>`

This command specifies the duration the RN1810 must stay asleep, in seconds, before waking up. For the new value to take effect, the `save` and `reboot` commands must be issued after this command.

Default: 5 seconds

Example:

```
set sys wake 10                 // The module wakes from Sleep mode
                                // after 10 seconds
```

2.4.38 `set time address <string>`

This command sets the name of the NTP server that must be queried, where *<string>* is the domain name of the NTP server. The server name must be 68 or less characters. Do *not* use an IP address. Refer to [Section 4.10 "SNTP Client"](#).

Default: pool.npt.org

Example:

```
set time address time.nist.gov // Set NTP server as time.nist.gov
```

2.4.39 `set time enable <value>`

This command configures when the SNTP client fetches the time from the NTP server, where *<value>* is:

- 0 - Disable the SNTP client
- 1 - Perform a single fetch from the SNTP server
- 2:N - Fetch time from the SNTP server every N minutes

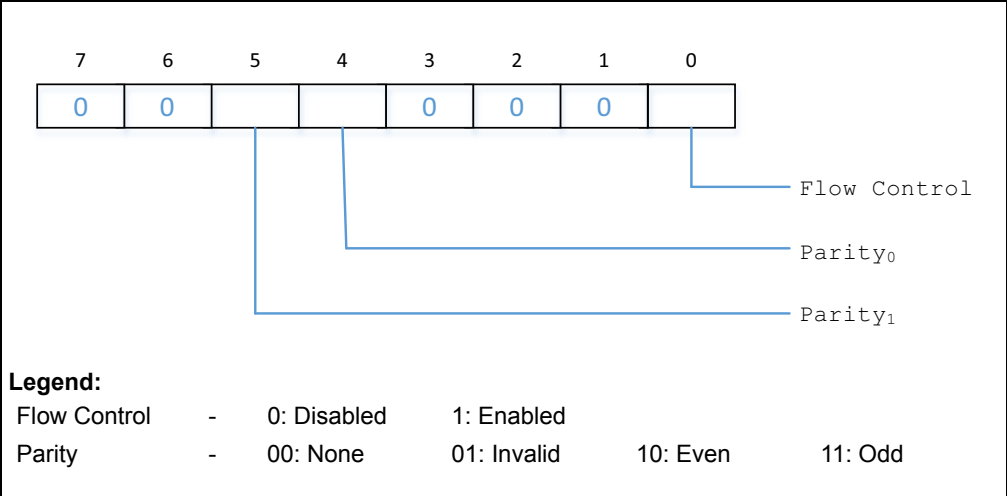
The enable value is also checked at boot time and, if greater than 0, it fetches the time from the NTP server as soon as associated to the network and assigned an IP address (static or dynamic). Note that WiFly must be on a network that can access a NTP server. Refer to [Section 4.10 "SNTP Client"](#).

Default: 0

Example:

```
set time enable 120             // SNTP client will fetch time from server
                                // every 120 minutes
```


FIGURE 2-2: FLOW CONTROL MODE AND PARITY BITMAP



When flow control is enabled:

UART0_RTS

- This pin is an RN1810 output pin (input to the host MCU).
- When the RN1810 UART0_RTS pin is low, the Host can send bytes to the RN1810.
- When the RN1810 UART0_RTS pin is high, it is unable to accept more characters from the Host and the Host must hold off transmitting.

UART0_CTS

- This pin is an RN1810 input pin (output from the host MCU)
- When the Host sets the UART0_CTS pin high, the RN1810 stops sending characters to the host MCU (Refer to [Note 2](#)).
- When the Host sets the UART0_CTS pin low, the RN1810 can send characters to the host MCU.

Note 1: Flow Control must be enabled if using 115200 baud.

2: When the Host sets UART0_CTS high the RN1810 may send up to ten more characters before the flow control takes effect and the RN1810 stops sending characters.

Default: Flow Control disabled, Parity is None

Example:

```
set uart flow 0x01 // Enable flow control, no parity bits
set uart flow 0x21 // Enable flow control, even parity
set uart flow 0x31 // Enable flow control, odd parity
set uart flow 0x30 // Disable flow control, odd parity
```

2.4.43 set uart instant <value>

This command immediately changes the baud rate, where <value> can be: 2400, 4800, 9600, 19200, 38400, 57600 or 115200. This command is useful when testing baud rate settings or when switching the baud rate on-the-fly while connected over TCP via Telnet. Using this command does *not* affect configuration. The module returns the AOK response, and then the module exits Command mode.

If used in Local mode, the baud rate changes and the module sends AOK using the new baud rate. If the Host immediately switches to the new baud rate, the Host may see the AOK string at the new baud rate. Depending on the baud rate, it takes at least ten times the bit rate for the module to issue the first character.

Default: Not applicable

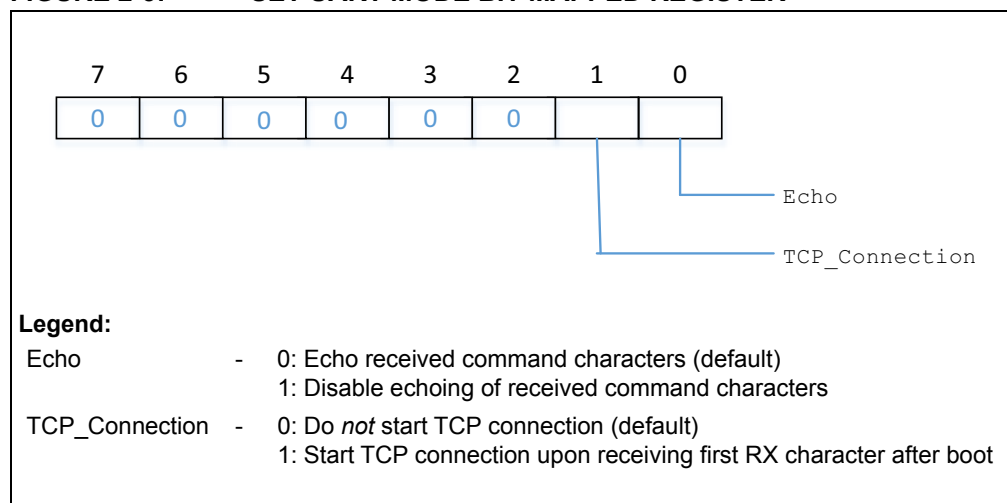
Example:

```
set uart instant 19200           // Set UART baud rate to 19200
```

2.4.44 set uart mode <value>

This command configures the WiFly UART special modes, where <value> is a bit-mapped number as shown in Figure 2-3.

FIGURE 2-3: SET UART MODE BIT-MAPPED REGISTER



Default: 0x0

Example:

```
set uart mode 0x02           // Start TCP connection upon receiving first  
                             // RX character after reboot
```

2.4.45 set uart raw <value>

This command sets a raw UART value, where <value> is a decimal number representing the baud rate. Use this command to set non-standard baud rates. The lowest possible baud rate is 2400.

Using non-standard raw baud rates with hardware flow control can be more useful at high speeds as the microcontroller interfaced to the module may be able to better match the UART speed and get better results.

Default: Not applicable

Example:

```
set uart raw 7200            // Set UART baud rate to 7200
```

2.4.46 set wlan auth <value>

This command sets the authentication mode, where <value> is shown in [Table 2-5](#). The firmware supports the following security modes:

- WEP-64 and WEP-128 (only Open-Key mode, *not* Shared-Key mode)
- WPA2-PSK (only AES)
- WPA-PSK (only TKIP)
- WPA-PSK mixed mode (some access points, *not* all are supported).

TABLE 2-5: AUTHENTICATION VALUES

Value	Authentication Mode
0	Open (default)
1	WEP-128 Open Key
2	WPA-PSK TKIP
3	WPA/WPA2-PSK Mixed
4	WPA2-PSK AES
8	WEP-64 Open Key

Default: 0 (Open)

Example:

```
set wlan auth 4           // Set security to WPA2-PSK authentication
```

2.4.47 set wlan hide <value>

This command configures how the passphrase is displayed by WiFly in response to the `get wlan` and `get everything` commands, where <value> is:

0: display passphrase

1: do *not* display passphrase, show "*" instead

Default: 0

Example:

```
set wlan hide 1           // WiFly displays '*'s instead of the actual passphrase
```

2.4.48 set wlan join <value>

This command sets the policy for automatically associating with network access points, where <value> is one of the options shown in [Table 2-6](#). The module uses this policy on power-up, including waking up from the sleep timer.

TABLE 2-6: NETWORK ASSOCIATION POLICY

Value	Policy
0	Manual. Do <i>not</i> try to automatically associate with a network.
1	Try to associate with the access point that matches the stored SSID and pass key. The RN1810 scans all valid channels when searching for the access point.
7	Create a Soft AP network using the stored SSID, IP address, netmask, channel, and so on.

Default: 0

Example:

```
set wlan join 7           // Create Soft AP network
```

2.4.49 set wlan key <value>

This command sets the WEP-64 or WEP-128 key, where <value> is the hexadecimal representation of the key. A WEP-64 key must be 10-hex digits. A WEP-128 key must be 26-hex digits.

Default: 0

Example:

```
set wlan key aabbccddeeff00112233445566 // Set WEP-64 key
set wlan key aabbccddeeff00112233445566 // Set WEP-128 key
```

2.4.50 set wlan mask <mask>

This command sets the WLAN channel mask used for scanning channels with an auto-join policy of 1 (Refer to [Table 2-6](#)), where <mask> is a hex number (bit 0 = channel 1). Reducing the number of channels scanned for association increases battery life.

Note: The token “mask” *cannot* be abbreviated.

Default: All channels are scanned

Example:

```
set wlan mask 0x0421 // Scan channels 1, 6, 11
```

2.4.51 set wlan mode_phy <value>

This command sets the Wireless Physical mode, where <value> is:

- 2: 11B mode
- 3: 11G mode
- 4: 11N mode

Note: The token “mode_phy” *cannot* be abbreviated.

Default: 0 (Mixed mode). *Cannot* be set by this command, but is set after a factory RESET.

Example:

```
set wlan mode_phy 3 // Set physical mode to 802.11G
```

2.4.52 set wlan number <value>

This command sets the WEP index to be used, where <value> is 1 through 4.

Default: 1

Example:

```
set wlan number 2 // Set WEP index to 2
```

2.4.53 `set wlan phrase <string>`

This command sets the passphrase for WPA and WPA2 security modes, where *<string>* is 1 to 64 characters (64 bytes). The passphrase is alphanumeric, and is used with the SSID to generate a unique 32-byte Pre-Shared Key (PSK), which is then hashed into a 256-bit number. When you change either the SSID or the passphrase, the module recalculates and stores the PSK.

If you enter exactly 64 characters, the module assumes that the passphrase is an ASCII hex representation of the 32-byte PSK, and the value is simply stored.

Note: The *<string>* *cannot* contain spaces. If the SSID contains spaces, use a replacement character. Refer to `set opt replace <value>` command.

Default: !microchip

Example:

```
set wlan phrase my_password // Set passphrase to my_password
```

2.4.54 `set wlan ssid <string>`

This command sets the SSID with which the module associates, where *<string>* is 1 to 32 characters (32 bytes).

Note: The *<string>* *cannot* contain spaces. If the SSID contains spaces, use a replacement character. Refer to `set opt replace <value>` command.

Default: microchip1

Example:

```
set wlan ssid my_network // Set SSID to my_network
```

2.4.55 `set wlan tx <value>`

This command sets a fixed transmit power level for the RN1810 module, where *<value>* is a value between 1 and 16 dBm.

Default: 16

Example:

```
set wlan tx 8 // Set transmit power to 8 dBm
```

2.5 GET COMMANDS

These commands begin with the keyword `get` and displays the current values of the module. Except where noted, the Get commands do *not* have any parameters.

2.5.1 `get console`

This command displays the console settings.

Example: `get console`

2.5.2 `get dns`

This command displays the DNS settings.

Example: `get dns`

2.5.3 **get everything**

This command displays many configuration settings.

Example: `get everything`

2.5.4 **get ftp**

This command displays the FTP settings.

Example: `get ftp`

2.5.5 **get ip**

This command displays the IP address and port number settings.

Example: `get ip`

2.5.6 **get mac**

This command displays the device's MAC address.

Example: `get mac`

2.5.7 **get softap**

This command displays the Soft AP settings.

Example: `get softap`

2.5.8 **get system**

This command displays the system settings.

Example: `get system`

2.5.9 **get time**

This command displays the current SNTP client configuration information.

Example: `get time`
`ENA=0 // Value set in set time enable <value>`
`SRV= pool.ntp.org // Value set in set time address <string>`
`ZONE= UTC-07:00,E // Value set in set time zone <value>`

2.5.10 **get uart**

This command displays the UART settings.

Example: `get uart`

2.5.11 **get version**

This command displays the firmware version number.

Example: `get version`

2.5.12 **get wlan**

This command displays the WLAN settings.

Example: `get wlan`

2.6 ACTION COMMANDS

Action commands are used to enter/exit Command mode, to perform factory Reset and run tests and applications, among others. Except where noted, the Action commands do *not* have any parameters.

2.6.1 \$\$\$

Use this command to enter Command mode. Type \$\$\$ sequentially with no additional characters before or after each \$ character. Do *not* type a carriage return (<cr>) after the \$\$\$ to enter Command mode. There must be a 250 ms guard-band period before and after typing the three \$'s. If the preceding rules are *not* followed, the \$ characters are treated as data and the RN1810 remains in Data mode. In summary, to enter Command mode perform the following steps:

250 ms of silence (no characters sent)

Send three \$ character (must be less than 250 ms between each \$)

250 ms of silence (no characters sent)

The module replies with CMD and displays the prompt <WIFLY> to indicate it is in Command mode.

To use a different character to enter Command mode (*not* \$\$\$), use the `set comm $ <char>` command.

Example:

```
$$$ // Enter Command mode
```

2.6.2 apmode <ssid> <channel>

This command initiates Soft AP mode. The following are the valid inputs:

```
apmode
```

```
apmode ssid
```

```
apmode ssid channel
```

Note: When Soft AP mode is invoked, the IP address of the module is 192.168.1.1. The default DHCP server pool is 192.168.1.11 through 192.168.1.20. The IP address of the module can be changed via the `set ip address <address>` command. The DHCP server pool and lease time can be changed via the `set dhcp lease` command. However, the gateway address is unchangeable and remain as 192.168.1.10 which must *not* cause any issue in network packet distribution.

Example:

```
apmode // Start Soft AP mode using previously set SSID and channel
apmode my_app // Start Soft AP mode with SSID set to my_app and
// previously set channel
apmode my_ssid 6 // Start Soft AP mode with SSID set to my_ssid and using
// channel 6
```

2.6.3 close

This command disconnects a TCP client connection. When the module is configured as a TCP/TLS Server, this command closes the connection with the connected client - the server continues to run. When the module is configured as a TCP/TLS Client, this command closes the clients connection with the server.

Example:

```
close                                // Close TCP connection
```

2.6.4 exit

This command exits Command mode and enters Data mode. After leaving Command mode, the module responds with EXIT.

Example:

```
exit                                // Exit Command mode
```

2.6.5 factory RESET

This command reboots the RN1810 settings to their factory default state. All previous settings are lost.

Example:

```
factory RESET                        // Reset configuration to factory defaults
```

2.6.6 ftp get

This command initiates a file read from the FTP server using the settings in the previous `set ftp ...` commands. For more information on how to use this command, refer to [Section 4.8“FTP Client”](#).

Example:

```
ftp get
```

2.6.7 ftp put

This command initiates a file write to the FTP server using the settings in the previous `set ftp ...` commands. For more information on how to use this command, refer to [Section 4.8“FTP Client”](#).

Example:

```
ftp put
```

2.6.8 join <string>

This command instructs the WiFly module to join the network indicated by <string>. If the network has security enabled, set the passphrase with the `set wlan pass` command prior to issuing the `join` command.

Note: The <string> must <i>not</i> contain spaces.

Example:

```
join                                // Join previously saved SSID
                                   // Refer to set wlan ssid <string>

join mchp                           // Join open network mchp
set wlan pass password              // Set the password to password
join mchp1                           // Join network mchp1
```

2.6.9 leave

This command instructs the WiFly module to leave the Wi-Fi network it is currently associated with.

Example:

```
leave // Leave current network
```

2.6.10 lookup <string>

This command causes the module to perform a DNS query, where <string> is the host name to search.

Example: lookup mchp1 // Search for the remote host mchp1

2.6.11 open <host> <port_number>

This command opens a TCP client, HTTP client or a TLS TCP client connection to the stored remote host IP address and the remote port number. The type of connection to open depend on the previous set ip protocol command. If <host> is a host name, the DNS client resolves it to an IP address. If <host> is an IP address, it directly used. A TLS TCP client must only use the open command with no parameters.

Upon successfully opening the connection, the RN1810 transitions from Command mode to Data mode.

Example:

```
open // Open TCP connection using previously set
// IP address and port number

open 192.168.1.102 2000 // Open TCP connection to 192.168.1.102,
// port 2000

open my_remote_host 2000 // Resolve my_remote_host to an IP address,
// open connection to port 2000
```

2.6.12 ota upgrade <file_name> <server_addr>

This command upgrades the module firmware using a Wi-Fi connection to a TFTP server. Before invoking this command, ensure that the following are set:

1. WiFly is connected to a wireless network
2. PC running the TFTP server is connected to the same wireless network
3. Upgrade file is at a location where the TFTP server can find it
4. TFTP server is running

Example:

```
ota upgrade wifly.bin 192.168.1.176 // Upgrade WiFly firmware
```

2.6.13 ping <address>

This command sends a ping, where <address> is the IPv4 address of the ping target. If the ping is successful, WiFly can output a response (for example, PING reply from 192.168.1.106). A single packet is sent.

Example:

```
ping 192.168.1.106 // Ping IP address 192.168.1.106
```

2.6.14 ping6 <address>

This command sends a ping, where <address> is the IPv6 address of the ping target. If the ping is successful, WiFly can output a response (for example, PING reply from 2001:db9::d0ff:474a:3798:2294). A single packet is sent.

Example:

```
ping 2001:db9::d0ff:474a:3798:2294    // Ping IP address
                                      // 2001:db9::d0ff:474a:3798:2294
```

2.6.15 reboot

This command forces the module to reboot (similar to a power cycle).

Example:

```
reboot                                // Force the module to reboot
```

2.6.16 release

This command forces the WiFly DHCP server to clear its DHCP client pool. This command is only applicable if WiFly is in Soft AP mode.

Example:

```
release                              // Clear DHCP client pool
```

2.6.17 rfctest <rate> <num_tries> <num_bytes> <channel> <header_type> [addr1] [addr2] [addr3] [addr4]

This command causes WiFly to transmit Wi-Fi packets. This is useful for regulatory testing, or verifying RF functionality. The command format is:

```
rfctest rate num_tries num_bytes channel header_type [addr1]
[addr2 [addr3] [addr4]
```

where

rate is Transmission rate; 0 through 19 as follows:

0: 1 mbps	10: 48 mbps
1: 2 mbps	11: 54 mbps
2: 5.5 mbps	12: 6.5 mbps
3: 11 mbps	13: 13.0 mbps
4: 6 mbps	14: 19.5 mbps
5: 9 mbps	15: 26 mbps
6: 12 mbps	16: 39 mbps
7: 18 mbps	17: 52 mbps
8: 24 mbps	18: 58.5 mbps
9: 36 mbps	19: 19.5 mbps

num_tries is Number of packets to transmit (1 - 14)

num_bytes is Number of data bytes in each packet (0 - 1400)

channel is Wi-Fi channel to use (1 - 11), or 0 (use current channel)

header_type is 0 - beacon frame
1 - QOS data frame
2 - data frame with four addresses

[addr1] is MAC address 1, format is xx:xx:xx:xx:xx:xx [Receiver]
Default: FF:FF:FF:FF:FF:FF

[addr2]	is	MAC address 2, format is xx:xx:xx:xx:xx:xx [Transmitter] Default: MAC address of the module
[addr3]	is	MAC address 3, format is xx:xx:xx:xx:xx:xx [Destination] Default: 00:1E:C0:DD:DD:D
[addr4]	is	MAC address 4, format is xx:xx:xx:xx:xx:xx [Source] Default: 00:1E:C0:EE:EE:EE

Note: If [addr2] is *not* specified, the default MAC address is the MAC address of the module.

Example:

```
rftest 2 10 100 1 0 // Rate = 2 mbps, num_tries = 10, num_bytes = 100,
// channel = 1, header_type = beacon frame, addr1,
// addr2, addr3, addr4 are using defaults
rftest 17 5 1000 6 2 // Rate = 52 mbps, num_tries = 5, num_bytes = 1000,
// channel = 6, header_type = data frame, addr1,
// addr2, addr3, addr4 are using defaults
rftest 5 10 300 11 2 00:1e:c0:bb:bb:bb
// Rate = 9 mbps, num_tries = 10, num_bytes = 300,
// channel = 11, header_type = data frame,
// addr1 = 00:1e:c0:bb:bb:bb, addr2, addr3, addr4 are
// using defaults
```

2.6.18 run <string>

This command runs applications, where

<string>	is	wps	Run WPS application in Infrastructure mode
		web_app	Run web application in Soft AP mode

Example:

```
run web_app // Run web server in Soft AP mode
```

2.6.19 save

This command saves the current configuration settings to module FLASH.

Example:

```
save // Save configuration settings
```

2.6.20 scan

This command scans for existing Wi-Fi networks. Refer to [Section 3.5“Scan Output Format”](#)) for the output format of the `scan` command.

Example:

```
scan // Active scan on all 13 channels with default dwell time
```

2.6.21 show <value>

This command shows various states. Refer to [Section 2.7“Show Commands”](#) for a detailed explanation of this command.

2.6.22 sleep

This command puts the module to sleep. Wake the module by using a Reset, using the wake timer, or the WAKEUP I/O line.

Example:

```
sleep                                // Put the module to sleep
```

2.6.23 time

This command sets the real-time clock by running an SNTP client to retrieve the time from an SNTP server. Before invoking this command the SNTP client must be configured via the 'set time ...' commands in 2.4.38 through 2.4.40.

Example:

```
time                                // Start SNTP client
```

2.7 SHOW COMMANDS

These commands begin with the keyword `show` and then displays the module's current states.

2.7.1 show ap

This command shows the devices connected to WiFly when it is in Soft AP mode.

Example: `show ap`

2.7.2 show io

This command displays the state of the RN1810 pins.

Example: `show io`

2.7.3 show ip

This command shows the IP state of the WiFly.

Example: `show ip`

2.7.4 show rssi

This command shows the current RSSI value for the module (in dB).

Example: `show rssi`

2.7.5 show net

This command shows the module's network state.

Example: `show net`

2.7.6 show time

This command displays the current time fetched from the NTP server. Once the time is successfully fetched, WiFly continues to update the time based on its internal clock.

Example:

```
<WIFLY> show time
Time=14:18:58           // Current time of day (24 hour clock)
Date=08/19/2015         // Current date
UTC=1439993938          // Unix epoch time; number of seconds since
                        // Jan 1, 1970
Uptime=217 sec          Time since last reboot (independent of NTP server)
```

Note 1: If the result of this command is what is shown below, then either the SNTP client has *not* yet run, or, the SNTP client was unable to reach the NTP server:

```
<WIFLY> show time
Time=18:00:00
Date=02/01/2000
UTC=949428000
Uptime=217 sec          // Always valid
```

2: The result below occurs if the SNTP client is disabled:

```
<WIFLY> show time
Time NOT SET
Uptime=3 sec           // Always valid
```

Chapter 3. Advanced Features and Settings

3.1 INTRODUCTION

This section describes the WiFly module's advanced features, including techniques for waking up the module and methods to open a TCP connection when the module is awake. It also describes the UART flow control, and the real-time clock.

3.2 SOFT ACCESS POINT (AP) MODE

WiFly modules support two methods for accessing Wi-Fi networks. In addition to Infrastructure mode the firmware also supports Access Point (AP) mode. AP mode provides several advantages. In AP mode:

- The module creates a Soft AP network to which all devices (smartphones and tablets) can join.
- The module runs a DHCP server and issues IP addresses to the clients.
- The WiFly module supports security
- The module supports routing between clients

The following sections describe how to use AP mode with WiFly products, including configuring the module to act as an AP, enabling AP mode in hardware and software, and sending data to the module from a remote Host.

3.2.1 Enabling Soft AP Mode

Enable Soft AP mode in the software by using the `set wlan join 7` command. The network settings such as the SSID and the channel in the software can be customized. For example, the following set of commands create a Soft AP network:

```
set wlan join 7           // Enable Soft AP mode
set apmode channel <value> // Specify the channel to create network
set apmode ssid <string>  // Setup network Broadcast SSID
                           // (BSSID)
set apmode passphrase <string> // Set passphrase
save                      // Store settings
reboot                    // Reboot the module in Soft AP mode
```

After rebooting, the module is in Soft AP mode with the above settings applied.

- If *no* channel is specified, the module starts the network on channel 1.
- If *no* SSID is specified, the module starts the network using an SSID of "WiFly-RN1810-xy", where xy is the last byte of the module's MAC address.
- If passphrase is specified, then Soft AP network uses WPA2-PSK (AES) security. If passphrase is *not* specified, then Open-Key security mode is used.
- Default IP address is 192.168.1.1
- Default mask is 255.255.255.0

3.3 SLEEP AND WAKE METHODS

Table 3-1 describes the methods for putting the module to sleep.

TABLE 3-1: METHODS FOR PUTTING THE MODULE TO SLEEP

Method	Interface	Description
Sleep Command	UART	Enter the Command mode using \$\$\$ and issue the <code>sleep</code> command.
SLEEP I/O	I/O Pin	Rising edge on SLEEP I/O pin puts the module in Sleep mode.

Table 3-2 describes methods for waking the module.

TABLE 3-2: METHODS FOR WAKING THE MODULE

Method	Interface	Description
Wake Timer	Internal RTC	The wake timer wakes the module based on the <code>set sys wake <value></code> command setting.
WAKEUP I/O	I/O Pin	Set WAKEUP I/O low to wake-up RN1810 from Sleep state.

When the module wakes up from sleep, it takes time (in milliseconds) to initialize the internal hardware. During this time, any data sent to the WiFly module over the UART is *not* processed. The STATUS_RDY I/O line can be monitored to determine when the module is ready for operations as described in [Section 5.2 “I/O Pin Function Select”](#). Alternately, the Host software can wait for the “*READY*” string to be received from WiFly.

3.4 TX FLUSH METHODS

When WiFly is in Data mode the Host sends TX data bytes via the UART. As WiFly receives this data, it is stored in its UART buffer. At some point, the data in the UART buffer is 'flushed', (encapsulated into an IP packet) and sent out to the wireless network. There are three available methods to decide when WiFly performs this 'flush' action as shown in [Table 3-3](#).

TABLE 3-3: WIFLY FLUSH METHODS

Flush Method	Description
Time	Every N ms WiFly encapsulates all UART RX bytes into an IP packet and sends the packet. This is the default method, with the default time at 100 ms. Corresponding Command: <code>set comm timer <value></code>
Size	When N bytes have accumulated in the UART RX buffer, WiFly encapsulates those bytes into an IP packet and sends the packet. Corresponding Command: <code>set comm size <value></code>
Match	When a specified byte is received, WiFly encapsulates all UART RX bytes into an IP packet and sends the packet. Corresponding Command: <code>set comm match <value> <flag></code>

3.4.1 UART Receiver and RTS/CTS Hardware Flow Control

At lower baud rates (less than 115000), the system can send data over TCP/IP without flow control.

Depending on the frequency and the quantity of the data being sent, the `comm` parameters optimize Wi-Fi performance by specifying when the system sends IP packets. To minimize latency and TCP/IP overhead, use the flush size or match character to send data in a single IP packet. In most cases, set the flush timer to a large number to avoid fragmentation. For high throughput, increase the UART baud rate, set the flush size to 1,460, and set the flush timer to a large value so that full IP packets are sent.

Refer to [Section 3.4 “TX Flush Methods”](#) for a description of how to control packet forwarding.

If the module is sending more than a few hundred thousand bytes in a single transaction, Hardware Flow Control must be enabled. The hardware must actively monitor the UART0_CTS pin. Flow control is *not* enabled by default as it is set using the `set uart flow 1` command.

3.5 SCAN OUTPUT FORMAT

The firmware supports a comma-delimited scan output format; for example:

```
02,01,-26,0000,0421,24:de:c6:4f:51:01,guest
```

The fields separated by comas are:

Index	Channel	RSSI	Security Mode	Capabilities	MAC Address	SSID
-------	---------	------	---------------	--------------	-------------	------

[Table 3-4](#) describes the scan fields.

TABLE 3-4: SCAN FIELDS

Field	Format	Description
Index	2-digit decimal number	Scan result index, starting at 01.
Channel	2-digit decimal number	AP Channel number
RSSI	Negative 2-digit decimal number	RSSI value
Security mode	2-byte (4-nibble) hex value	Bit map of Security modes. If <i>no</i> bit is set then AP has open security. Refer to Table 3-5 .
Capabilities	2-byte (4-nibble) hex value	Bit map of capabilities value. Refer to Table 3-6 .
MAC address	String	MAC address of AP (format is xx:xx:xx:xx:xx:xx)
SSID	String	SSID name of AP

TABLE 3-5: SECURITY BIT MASK

15	14	13	12	11:7	6	5	4:3	2	1	0
8021X WPA2	WPA2_P SK	N/A	WPA PSK	N/A	WPA2 CCMP	WPA2 TKIP	N/A	WPA TKIP	N/A	WEP (64 or 128)

TABLE 3-6: CAPABILITIES INFO

15:14	13	12:11	10	9:8	7	6	5	4	3	2	1	0
Reserved	DSSS- OFDM	Reserved	Short slot time	Reserved	Channel Agility (802.11b)	PBCC (802.11b)	Short Preamble (802.11b)	Privacy	CF-Poll Request	CF- Pollable	IBSS	ESS

An example output from the `scan` command is shown below:

```
<WIFLY> scan
SCAN:Found 3
01,01,-26,0000,0421,24:de:c6:4f:51:01,guest
02,01,-24,8040,0431,24:de:c6:4f:51:02,mchp-secure
03,06,-26,0000,1421,28:cf:da:b9:f6:2d,wpd_airport_A1408
END:
```

Note: The string `END:` signifies the end of scan data.

Chapter 4. Application Examples

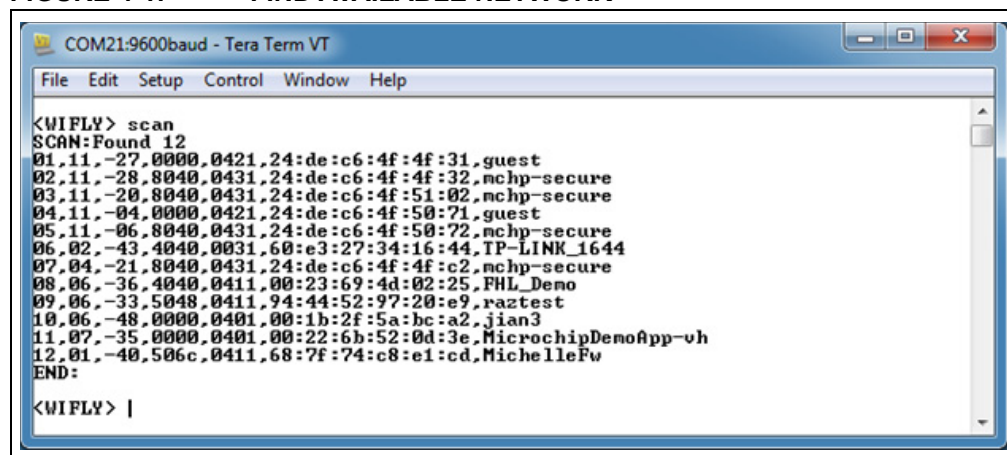
4.1 JOINING NETWORKS AND MAKING CONNECTIONS

Configuring the module to make connections involves associating with an access point and opening a connection. This chapter describes how to configure the module via its USB UART connector. Open a corresponding terminal emulator on the COM port associated with the module. The default baud rate is 9600, 8 bits, and *no* parity. Flow control is disabled by default.

4.1.1 Associate with an Access Point

From within the terminal window, put the module into Command mode by typing `$$$`. The module responds with `CMD`, indicating that it is in Command mode. Use the `scan` command to find the available networks as shown in [Figure 4-1](#).

FIGURE 4-1: FIND AVAILABLE NETWORK



```

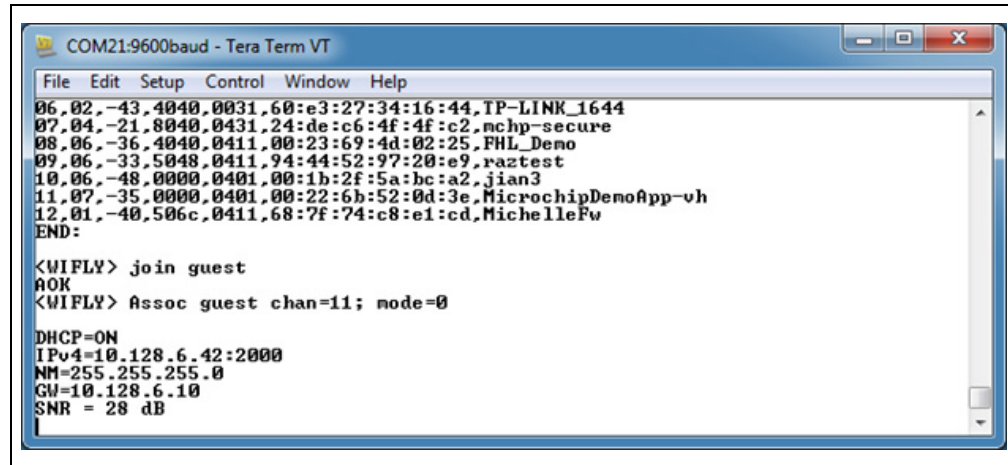
COM21:9600baud - Tera Term VT
File Edit Setup Control Window Help

<WIFLY> scan
SCAN:Found 12
01,11,-27,0000,0421,24:de:c6:4f:4f:31,guest
02,11,-28,8040,0431,24:de:c6:4f:4f:32,mchp-secure
03,11,-20,8040,0431,24:de:c6:4f:51:02,mchp-secure
04,11,-04,0000,0421,24:de:c6:4f:50:71,guest
05,11,-06,8040,0431,24:de:c6:4f:50:72,mchp-secure
06,02,-43,4040,0031,60:e3:27:34:16:44,IP-LINK_1644
07,04,-21,8040,0431,24:de:c6:4f:4f:c2,mchp-secure
08,06,-36,4040,0411,00:23:69:4d:02:25,FHL_Demo
09,06,-33,5048,0411,94:44:52:97:20:e9,raztest
10,06,-48,0000,0401,00:1b:2f:5a:bc:a2,jian3
11,07,-35,0000,0401,00:22:6b:52:0d:3e,MicrochipDemoApp-vh
12,01,-40,506c,0411,68:7f:74:c8:e1:cd,MichelleFw
END:

<WIFLY> |
  
```

If you are connecting to an open network, ensure that the DHCP client is enabled (for example `set ip dhcp 1`) and then use the `join` command to associate with the access point. The scan list in [Figure 4-1](#) shows that the `guest` is an open access point. Type `join guest` to associate with the network as shown in [Figure 4-2](#).

FIGURE 4-2: JOIN THE NETWORK



If the access point is secure, set the passphrase prior to issuing the `join` command. To set the WPA passphrase, use the `set wlan phrase <string>` command.

4.2 SENDING DATA USING TCP - MODULE IS A TCP SERVER

Setting up a TCP Server is very simple. Setup the SSID and passphrase, turn on auto-join and then save and reboot. When the module reboots, it connects to the SSID, gets an IP address, and starts the TCP server on port 2000. Open a telnet session from any machine connected to the same SSID, and enter “open <ipaddr> 2000”, where <ipaddr> is the assigned IP address of the module (Refer to command `get ip`), and then enter the characters from the window. Pressing <Enter> sends the data to the module. The module can also send data to the Client when the user enters data in the console window opened via the serial port of the module. This is a simple chat demo with the module.

```
set ip dhcp 1           // Enable DHCP client
set wlan ssid <string>  // Set the network name
If WPA2-PSK is supported:
set wlan phrase <string> // Set the passphrase for WPA and WPA2 modes
set wlan auth 4         // Set the authentication type to WPA and
                        // WPA2 modes
set wlan join 1         // Auto join
save                    // Save the configuration to the Flash
reboot                  // Reboot the module
```

To force the client to disconnect, enter Command mode and issue the `close` command. This forces the connected client to close its connection to the server where the server continues to run, and can only stop via a Reset.

4.3 SENDING DATA USING TCP - MODULE IS A TCP CLIENT

The module is configured to be a TCP client using the following command sequence. Start a Remote TCP Server on a Linux/Windows machine (for example, “ncat -l -k 50008”), and the module connects to this TCP server. Once connected, data is sent from the console window.

```
set ip dhcp 1           // Enable DHCP client
set wlan ssid <string> // Set the network name

If WPA2-PSK is supported:

set wlan phrase <string> // Set the passphrase for WPA and WPA2 modes
set wlan auth 4          // Set the authentication type to WPA and
                        // WPA2 modes

set wlan join 1          // Auto join
set ip protocol 8        // TCP Client
set ip host <ip_addr>    // IP address of the remote TCP server
set ip remote 50008      // Remote TCP server port number
save                     // Save the configuration to the Flash
reboot                  // Reboot the module
$$$                     // Enter Command mode
open                    // Open a connection to the TCP Server
                        // (transition to Data mode)

...                      // Send data
$$$                     // Enter Command mode
close                   // Close client connection to server
```

4.4 SENDING DATA USING IPV6 - MODULE IS A TCP CLIENT

The module is configured to be a TLS client using the following command sequence. Start a Remote SSL/TLS Server on a Linux/Windows machine (for example, “ncat -l -k --ssl --ssl-key foo.key --ssl-cert foo.pem 1443”), and the module connects to this TCP Server. Once connected, data is sent from the console window.

```
set ip dhcp 1           // Enable DHCP client
set wlan ssid <string> // Set the network name—must be an IPv6 AP

If WPA2-PSK is supported:

set wlan phrase <string> // Set the passphrase for WPA and WPA2 modes
set wlan auth 4          // Set the authentication type to WPA and
                        // WPA2 modes

set wlan join 1          // Auto join
set ip version 1         // IP version is IPv6
set ip protocol 8        // TCP Client
set ip host <ip_addr>    // IPv6 IP address of the remote TCP Server
set ip remote 50008      // Remote TCP Server port number
save                     // Save the configuration to Flash
reboot                  // Reboot the module
$$$                     // Enter Command mode
open                    // Open a connection to the TCP Server
                        // (transition to Data mode)
```

Note: To send/receive data using IPv6 for any of the TCP/UDP/TLS Server/Client, add only the `set ip version 1` command (Set IP version to IPv6), and let the rest of the commands untouched. This enables IPv6 based send/receive.

4.5 SENDING DATA USING UDP - MODULE IS A UDP CLIENT

UDP is a connectionless protocol where there is *no* initial handshaking between the hosts to setup the UDP connection, and the receiver does *not* send an acknowledgment when it receives UDP packets. Therefore, UDP is an unreliable protocol because there is *no* guarantee that the data is correctly delivered. However, because it is connectionless, UDP is suited for applications that *cannot* tolerate too much latency, but can tolerate some errors in the data, for example, video transmission.

To use UDP with the module, enable the UDP protocol using the `set ip proto 1` command. The remote host's IP address and the local and remote port numbers to use for UDP communications must also be specified. [Example 4-1](#) and [Example 4-2](#) show the commands used to enable UDP data transfer:

EXAMPLE 4-1: ASSOCIATE WITH A NETWORK

```
set wlan ssid <string> // Set the network name
If WPA2-PSK is supported:
set wlan phrase <string> // Set the passphrase for WPA and WPA2 modes
set wlan auth 4 // Set the authentication type to WPA and
// WPA2 modes
```

EXAMPLE 4-2: SETUP PROTOCOL AND PORT NUMBER

```
set ip proto 1 // Enable UDP as the protocol
set ip host <address> // Set the remote host's IP address
set ip remote <value> // Set remote port on the Host that listens
set ip local <value> // Set port number on the module that listens
save // Save settings in the configuration file
reboot // Reboot the module
```

As UDP is a connectionless protocol, data begins flowing as soon as the module is rebooted. Unlike TCP, it is *not* required to send an open command to establish the connection. The module acts like a data pipe where the UART data is sent over the Wi-Fi link via the UDP protocol (in this case) and the data coming from the Wi-Fi link (via UDP protocol in this case) is sent to the UART.

4.6 SENDING DATA USING TLS - MODULE IS A TLS SERVER

Setting up a TLS Server is very simple. Setup the SSID and passphrase, turn on auto-join, and then save and reboot. When the module reboots, it connects to the SSID, gets an IP address and starts the TLS Server on port 2000. Open an SSL session from any machine (for example, on a Linux machine - "ncat --ssl <ipaddr_server> 2000"), and enter characters from the window. Pressing <Enter> send the data to the module. The module can also send data to the Client when the user enters data in the console window opened via the serial port of the module. This is a simple chat demo with the module.

```

set ip dhcp 1           // Enable DHCP client
set wlan ssid <string>  // Set the network name
If WPA2-PSK is supported:
set wlan phrase <string> // Set the passphrase for WPA and WPA2 modes
set wlan auth 4         // Set the authentication type to WPA and
                        // WPA2 modes

set wlan join 1         // Auto join
set ip protocol 0x200   // Setup the protocol as TLS Server
save                   // Save the configuration to Flash
reboot                // Reboot the module

```

To force the Client to disconnect, enter Command mode and issue the `close` command. This forces the connected Client to close its connection to the server where the server can only stop via a Reset.

Note: When the module is a TLS server and the `sleep` command is entered, the module forcefully shutdown the TLS connection, disconnect the client, and go to sleep. When the module wakes up, it comes up as a TLS Server with a new connection, and the client must open a new socket connection with the server and perform the TLS handshake in order to exchange data.

4.7 SENDING DATA USING TLS - MODULE IS A TLS CLIENT

The module is configured to be a TLS client using the following command sequence. Start a Remote TLS Server on a Linux/Windows machine (for example, “`ncat --ssl --ssl-key foo.key --ssl-cert foo.pem -l -k 50008`”), and the module connects to this TLS Server. Once connected, data is sent from the console window.

```

set ip dhcp 1           // Enable DHCP client
set wlan ssid <string>  // Set the network name
If WPA2-PSK is supported:
set wlan phrase <string> // Set the passphrase for WPA and WPA2 modes
set wlan auth 4         // Set the authentication type to WPA and
                        // WPA2 modes

set wlan join 1         // Auto join
set ip protocol 0x20    // TLS Client
set ip host <ip_addr>   // IP address of the remote TLS Server
set ip remote 50008     // Remote TLS Server port number
save                   // Save the configuration to Flash
reboot                // Reboot the module
$$$                  // Enter Command mode
open                  // Open a connection to the TLS Server
                    // (transition to Data mode)

```

To close the connection to the server, enter Command mode followed by the `close` command.

Note: When the module is a TLS client and the `sleep` command is entered, the module forcefully shutdown the TLS connection, disconnect from the server, and go to sleep. When the module wakes up, it comes up as a TLS client without any connection, and the client must open a new socket connection with the server and perform the TLS handshake in order to exchange data.

4.8 FTP CLIENT

WiFly supports FTP client functionality allowing the Host to transfer files to and from an FTP server. This is controlled through WiFly commands and pins for handshaking. WiFly only supports FTP passive mode.

4.8.1 Reading a File from FTP Server

This section gives the typical command and pin handshaking sequence to setup a transfer of file from the FTP Server to the Host. The command that initiates the FTP client file read is `ftp get`. To read a file from the FTP server use the `TCP_STATUS` pin.

TABLE 4-1: HOST SEQUENCE TO READ FILE FROM FTP SERVER

Host	WiFly
Host has previously connected to an AP and has received an IP address. WiFly is in Command mode.	—
Sets FTP client protocol (<code>set ip protocol 0x400</code>) Setup FTP client transfer using <code>set ftp ...</code> commands. For example: <pre>set ftp address 192.168.1.176 set ftp remote 21 set ftp user john set ftp password my_password set ftp dir set ftp file test_file.txt set ftp timeout 20</pre>	Stores the FTP settings
Sends <code>ftp get</code> command.	Receives the <code>ftp get</code> command. Sets <code>TCP_STATUS</code> low.
Host waits for <code>TCP_STATUS</code> to go high.	WiFly establishes a connection with the FTP server. When ready for file transfer, sets <code>TCP_STATUS</code> high, signaling to the Host that the file data is going to be transferred to Host. WiFly automatically transitions from Command mode to Data mode.
Detects <code>TCP_STATUS</code> going high. This signals to the Host that: 1) WiFly has transitioned to Data mode and 2) All subsequent RX bytes are file data bytes.	WiFly reads file from FTP server and sends data to Host.
Host receives file data bytes	When last file data byte transferred, WiFly sets <code>TCP_STATUS</code> low, signaling that all file data bytes have been transferred to the Host.
Detects <code>TCP_STATUS</code> going low, signaling that the file transfer is complete.	—

4.8.2 Writing a File to FTP Server

This section gives the typical command and pin handshaking sequence to setup a transfer of file from the Host to an FTP server. The command that initiates the FTP client file write is `ftp put`. To write a file to the FTP server, use both the `TCP_CTRL` and `TCP_STATUS` pins.

TABLE 4-2: HOST SEQUENCE TO WRITE FILE TO FTP SERVER

Host	WiFly
Host has previously connected to an AP and has received an IP address. WiFly is in Command mode.	—
Set FTP client protocol (<code>set ip protocol 0x400</code>) Setup FTP client transfer using the <code>set ftp ...</code> commands. For example: <code>set ftp address 192.168.1.176</code> <code>set ftp remote 21</code> <code>set ftp user john</code> <code>set ftp password my_password</code> <code>set ftp dir</code> <code>set ftp file test_file.txt</code> <code>set ftp timeout 20</code>	Stores the FTP settings
a) Send <code>ftp_put</code> command. b) Set <code>TCP_CTRL</code> low to enable WiFly to detect if status is going high later in the sequence. c) Wait for <code>TCP_STATUS</code> to go high.	Receives <code>ftp_put</code> . Establishes a connection with the FTP server. When ready for file data transfer, sets <code>TCP_STATUS</code> high, signaling to the Host that it can now transfer file data. WiFly automatically transitions from Command mode to Data mode.
Detects <code>TCP_STATUS</code> going high. Send file data to WiFly.	WiFly receives file data and sends to FTP server.
After all file data have been transferred, sets <code>TCP_CTRL</code> high, signaling that WiFly file data transfer is complete and the WiFly can close the FTP connection.	Detects <code>TCP_CTRL</code> going high. Closes connection to FTP server. Sets <code>TCP_STATUS</code> low.
Waits for, and detects <code>TCP_STATUS</code> going low. FTP transaction is terminated and the next command can be sent.	—
Sets <code>TCP_CTRL</code> low (allows for future connections)	—

4.9 WI-FI PROTECTED SETUP (WPS)

Wi-Fi Protected Setup (WPS) created by the Wi-Fi Alliance is a standard for easy and secure establishment of a wireless home network.

The goal of the WPS protocol is to simplify the process of configuring security on wireless networks. The protocol is meant to allow home users with little knowledge of wireless security and being intimidated by the available security options to configure Wi-Fi Protected Access which is supported by most Wi-Fi certified devices that are available for purchase today.

The most common mode of WPS is the Push Button Configuration (PBC) mode in which the user simply pushes a button on both the access point and the wireless client (for example, the WiFly module). Refer to [Figure 4-3](#).

FIGURE 4-3: PUSH BUTTON WPS



4.9.1 Launching a WPS Application

Use the `run wps` command in the console. Use an I/O pin to launch WPS. Refer to [Section 5.2 "I/O Pin Function Select"](#).

When the WPS application launches, it negotiates the SSID and passphrase with the AP and reboots the module to associate with the WPS-enabled access point.

By default, during the WPS process, the module prints messages on the UART as it scans channels, detects access points, and tries to complete WPS. Refer to [Figure 4-4](#).

FIGURE 4-4: WPS PROCESS

```
COM25:115200baud - Tera Term VT
File Edit Setup Control Window Help
CMD
<WIFLY> set i d 1Erri:sndfrm 4492bc
AOK
<WIFLY> run wpsWPS connecting ...
<WIFLY>
NO Tx aggr.ni_flags: 0x0
Connecting AP ...:
Erri:sndfrm 4492fc
Assoc PHL_Deno chan=1; mode=0 RSSI 0
_dhc_setip Got IP address c0a80176
DHCP=ON
IPv4=192.168.1.118:2000
NM=255.255.255.0
GW=192.168.1.1
SOCK_OPEN : index 0 handle 4469332
Socket created
SOCK_BIND: index 0 handle 4469332 res 0
Socket Bind
```

4.10 SNTP CLIENT

Example 4-3 shows the sequence of commands entered to acquire the time from an SNTP server, showing both the commands and responses.

EXAMPLE 4-3: ACQUIRE TIME FROM AN SNTP SERVER

Command	Response	Description
\$\$\$	CMD <WIFLY>	Enter Command mode
set wlan ssid MyAp	AOK <WIFLY>	Set SSID
set wlan auth 0	AOK <WIFLY>	Set open security
set wlan join 1	AOK <WIFLY>	Set auto-join
set ip version 0	AOK <WIFLY>	Set IPv4
set ip dhcp 1	AOK <WIFLY>	Enable DHCP client
save	Verify config data: succeeded <WIFLY>	Save above settings
reboot	*Reboot* Version: 0_7_8 r631 for RN1810 Build: 3.3.5.115 Mac Addr STA=00:1e:c0:0c:eb:9e Mac Addr SoftAP=00:1e:c0:0c:eb:9e *READY* Assoc MyAp chan=1; mode=0 RSSI 0 DHCP=ON IPv4=192.168.1.100:2000 NM=255.255.255.0 GW=192.168.1.1	Reboot and connect
\$\$\$	CMD <WIFLY>	Enter Command mode
set time address time.nist.gov	AOK <WIFLY>	Set address of NTP time server
set time zone UTC-07:00,E	AOK <WIFLY>	Set time zone and offset
set time enable 1	AOK <WIFLY>	Enable SNTP client to fetch
get time	ENA=1 SRV=time.nist.gov ZONE=UTC-07:00,E AOK <WIFLY>	Verify SNTP client settings (optional)
[...wait a few seconds for SNTP client to perform the fetch]		
show time	Time=14:17:07 Date=12/01/2015 UTC=1448979427 sec Uptime=9 sec AOK <WIFLY>	Get the time fetched by the SNTP client

4.11 CONFIGURATION WEB SERVER

This section describes how to configure the module using its built-in web server.

WiFly modules can operate in one of two modes:

- **Infrastructure mode** - the module can join a network created by an access point (AP)
- **Soft AP mode** - The module behaves as an AP with limited functionality

A key challenge when using any embedded device in Infrastructure mode is to provision it to associate with an AP. This process requires storing the AP's settings, such as the SSID and passphrase, in the embedded device.

Embedded Wi-Fi modules can be configured or provisioned to join an infrastructure network in several ways:

- Sending ASCII commands to the module over a UART
- Using Wi-Fi Protected Setup (WPS)
- Sending commands to the module remotely using a web interface

4.11.1 Using the Configuration Web Server

Configuring the embedded WiFly module to associate with an AP in Infrastructure mode involves the following process:

1. Invoke the module's configuration web server.
2. Connect your client device (PC, smartphone, tablet, and so on) to the module's Soft AP network.
3. Access the module's configuration web page from your client device's web browser.
4. Save the settings (SSID and Security mode) in your web browser and exit.

4.11.1.1 INVOKE THE CONFIGURATION WEB SERVER

There are two methods for running the RN1810 web server application the web server:

- In software: via the `run web_app` command
- In hardware: via the FUNC_CONFIG I/O pin (Refer to I/O Pin Function Select in [Section 5.2 "I/O Pin Function Select"](#))

When you run the configuration web server, it creates a Soft AP network with the settings shown in [Table 4-3](#).

TABLE 4-3: SOFT AP NETWORK SETTINGS

Setting	Soft AP Mode Default
SSID	RN1810_XX where XX is the last byte of the module's MAC address
Channel	1
DHCP Server	Enabled
IP Address	192.168.1.10
Netmask	255.255.255.0

4.12 USING THE WEB SERVER TO CONFIGURE THE RN1810

This section describes how to use the web server to configure the WiFly module with the AP's SSID and Security mode. The example uses the Internet Explorer web browser running on a Windows 7 machine; however, the same concepts apply to any device with a Wi-Fi interface (for example, iPhone®, Android™ smartphones, tablets or PCs,) running a web browser (for example, Chrome™, Firefox® or Safari®).

To configure the module using a web browser, perform the following steps:

1. Associate the PC to the module's AP network. Launch your web browser.
2. Type <http://192.168.1.10> to go to the home page of the web server running on the module. The page has two tabs displayed by default:
 - **Network Configuration** - Used to set the AP's SSID and Security mode.
 - **Information** - Displays information about the WiFly module.

Figure 4-5 and Figure 4-6 show the screen shots displaying the two web pages.

FIGURE 4-5: NETWORK CONFIGURATION PAGE

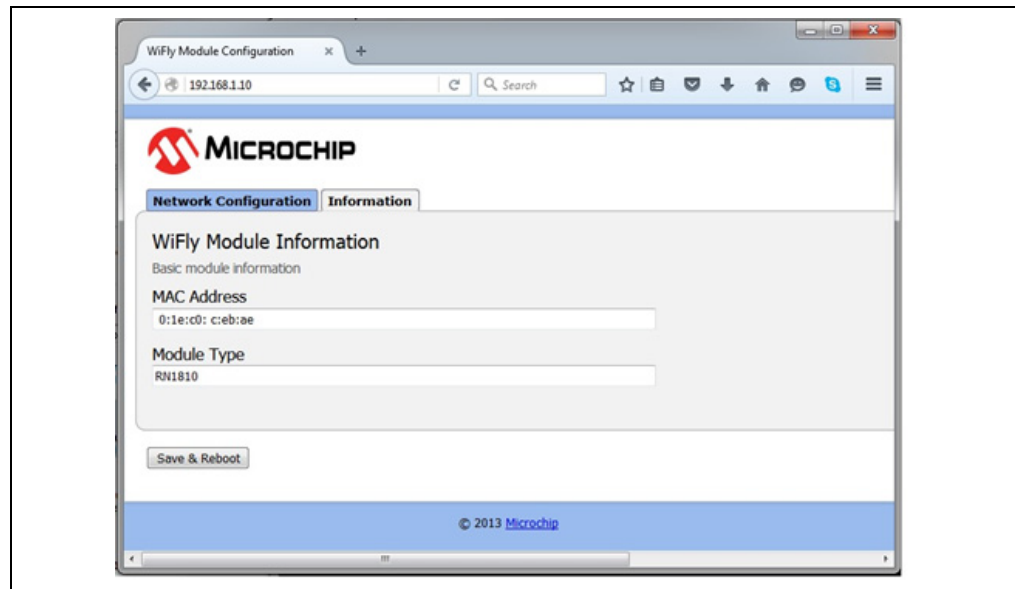
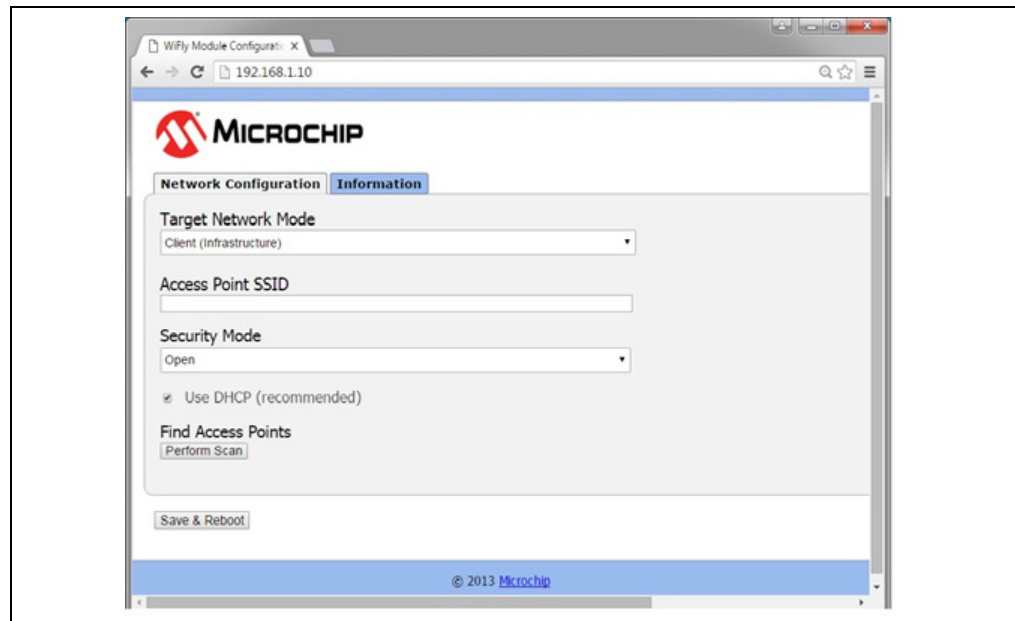


FIGURE 4-6: INFORMATION PAGE



4.13 AUTO-CONNECTION AND SLEEP TIMERS

The RN1810 can be configured to periodically sleep and wake-up. Going to sleep and waking up as well as be triggered by connections and sending data. There are five timers that are used for various operations:

TABLE 4-4: TIMERS FOR VARIOUS OPERATIONS

Timer	Description
Sleep Timer	Determines in seconds how long the module must sleep. This is a 32-bit value, corresponding to a maximum value of 1.19 hours. The sleep timer is set with the <code>set sys sleep <value></code> command.
Wake Timer	Determines in seconds how long the module must remain in Sleep mode before waking up. This is a 22-bit number corresponding to a maximum value of 1,165 hours. The wake timer is set with the <code>set sys wake <value></code> command.
Auto-connect Timer (TCP)	Determines in seconds how long the module must wait after reboot before opening a TCP connection. The Auto-connect timer is set with the <code>set sys autoconn <value></code> command.
Auto-connect Timer (HTTP)	Determines in seconds how often an HTTP client must open a connection to a HTTP server. The timer is set with the <code>set sys auto <value></code> command.
Idle Timer	Determines in seconds how long it takes to close a TCP connection that is idle. The Idle timer is set with the <code>set comm idle <value></code> command.

4.13.1 Auto-Connect and Periodic Sleep-Wake

The following command sequence shows how to configure the RN1810 with several timer options.

```
set ip remote_port 2000 // Setup the remote machine's IP port
set sys autoconn 1      // Connect immediately upon waking up
set com idle 5          // Disconnect after 5 seconds of no data activity
set sys sleep 2         // Sleep 2 seconds after connection is closed
set sys wake 60         // after 1 minute of sleep
```

4.13.2 HTTP Client Connect Periodically to Web Server

The following command sequence shows how to configure the RN1810 to connect to a web server every 30 seconds and send a string upon opening the connection.

```
set comm remote GET$/ob.php?obvar=WEATHER // String to send
set sys auto 30                          // Auto-connect every
                                           // 30 seconds
```

4.13.3 UDP Auto-Sleep

The RN1810 is capable of automatically going into Sleep mode for a designated period of time after sending a UDP packet. The sleep time is configured via two commands:

```
set sys autosleep <value> and  
set comm timer <value>.
```

The time interval is a product of these two values where the comm timer value is in milliseconds and the auto-sleep value is a multiplier. Examples are shown in [Table 4-5](#).

TABLE 4-5: AUTO-SLEEP PERIODS

set sys autosleep <value>	set comm timer <value>	Sleep Period
4	10 ms	40 ms (4 * 10)
2	20 ms	40 ms (2 * 20)
10	10 ms	100 ms (10 * 10)

[Example 4-4](#) shows a command sequence to auto-sleep 120 ms after sending a UDP packet.

EXAMPLE 4-4: SETTING 120 MS AUTO-SLEEP AFTER SENDING A UDP PACKET

```
set ip 0x01 // UDP protocol  
set ip remote 2000 // Port 2000 on remote Host  
set ip host 192.168.1.176 // IP address of remote Host  
set sys autosleep 4 // Set multiplier to 4  
set comm timer 30 // 4 * 30 = 120 ms sleep period  
save // Save settings  
reboot // Reboot module so settings take effect  
join // Join Wi-Fi network (presumes SSID,  
// security, etc. previously saved)  
... wait for connection  
exit // Exit to Data mode  
... send UDP packet // After packet sent and 120 ms elapsed,  
// module goes to sleep
```

NOTES:

Chapter 5. RN1810 I/O Pins

5.1 I/O PIN DESCRIPTIONS

Table 5-1 and Table 5-2 describe the RN1810 I/O pins. With a few exceptions described in the tables, most I/O pins are optional and their functionality can be replaced by a command or status text sent by WiFly.

TABLE 5-1: RN1810 OUTPUT PINS

RN1810 Output Pin	Description										
CMD_STATUS Pin 5	WiFly outputs different patterns to indicate various states. Typically this pin is tied to an LED. <table> <tr> <th>Pattern</th><th>Description</th></tr> <tr> <td>Low</td><td>In WPS mode</td></tr> <tr> <td>One toggle per second</td><td>In Data mode and: 1. In Infrastructure mode, or 2. In Web App mode with <i>no</i> associated client </td></tr> <tr> <td>Four toggles per second</td><td> 1. In Command mode, or 2. In AP mode </td></tr> <tr> <td>High</td><td>In Web App mode with associated client (and possibly a socket open by client)</td></tr> </table>	Pattern	Description	Low	In WPS mode	One toggle per second	In Data mode and: 1. In Infrastructure mode, or 2. In Web App mode with <i>no</i> associated client	Four toggles per second	1. In Command mode, or 2. In AP mode	High	In Web App mode with associated client (and possibly a socket open by client)
Pattern	Description										
Low	In WPS mode										
One toggle per second	In Data mode and: 1. In Infrastructure mode, or 2. In Web App mode with <i>no</i> associated client										
Four toggles per second	1. In Command mode, or 2. In AP mode										
High	In Web App mode with associated client (and possibly a socket open by client)										
IP_STATUS Pin 8	WiFly outputs different patterns to indicate various states. <table> <tr> <th>Pattern</th><th>Description</th></tr> <tr> <td>Low</td><td><i>Not</i> connected or associated</td></tr> <tr> <td>High</td><td>Connected to an available AP and/or IP address</td></tr> <tr> <td>Four toggles per second</td><td>In Web App mode with client associated (and possibly a socket open by client)</td></tr> </table>	Pattern	Description	Low	<i>Not</i> connected or associated	High	Connected to an available AP and/or IP address	Four toggles per second	In Web App mode with client associated (and possibly a socket open by client)		
Pattern	Description										
Low	<i>Not</i> connected or associated										
High	Connected to an available AP and/or IP address										
Four toggles per second	In Web App mode with client associated (and possibly a socket open by client)										

TABLE 5-1: RN1810 OUTPUT PINS (CONTINUED)

RN1810 Output Pin	Description										
MISC_STATUS Pin 26	WiFly outputs different pulse patterns to indicate various connection states. Typically this pin would be tied to an LED. <table> <tr> <th>Pattern</th><th>Description</th></tr> <tr> <td>Low</td><td> 1. In Infrastructure mode and associated with AP, or 2. In AP mode and client associated, or 3. In Web App mode and client connected (and possibly a socket open by client) </td></tr> <tr> <td>One toggle per second</td><td>In Web App mode and <i>no</i> client connected</td></tr> <tr> <td>Two toggles per second</td><td> 1. In Infrastructure mode and <i>not</i> associated with AP, or 2. In AP mode and <i>no</i> client connected </td></tr> <tr> <td>Four toggles per second</td><td>In WPS mode and <i>not</i> connected</td></tr> </table>	Pattern	Description	Low	1. In Infrastructure mode and associated with AP, or 2. In AP mode and client associated, or 3. In Web App mode and client connected (and possibly a socket open by client)	One toggle per second	In Web App mode and <i>no</i> client connected	Two toggles per second	1. In Infrastructure mode and <i>not</i> associated with AP, or 2. In AP mode and <i>no</i> client connected	Four toggles per second	In WPS mode and <i>not</i> connected
Pattern	Description										
Low	1. In Infrastructure mode and associated with AP, or 2. In AP mode and client associated, or 3. In Web App mode and client connected (and possibly a socket open by client)										
One toggle per second	In Web App mode and <i>no</i> client connected										
Two toggles per second	1. In Infrastructure mode and <i>not</i> associated with AP, or 2. In AP mode and <i>no</i> client connected										
Four toggles per second	In WPS mode and <i>not</i> connected										
STATUS_RDY Pin 6	Set by WiFly during Reset or after <code>reboot</code> command. 0: WiFly <i>not</i> ready for operations 1: WiFly ready for operations										
TCP_STATUS Pin 12	This WiFly output can be used in two contexts: 1. Set by WiFly to signal TCP connection status; it can be checked after the Host sends an <code>open <host> <port_number></code> command (or uses the TCP_CTRL pin) to initiate a TCP connection. 0: TCP connection <i>not</i> yet complete 1: TCP connection complete 2. This pin is <i>required</i> for FTP transfers. Refer to Section 4.8“FTP Client”.										
UART0_RTS Pin 10	<i>Required</i> if using UART handshake. Refer to <code>set uart flow <value></code> command.										
UART0_TX Pin 7	WiFly UART TX pin. <i>Required.</i>										

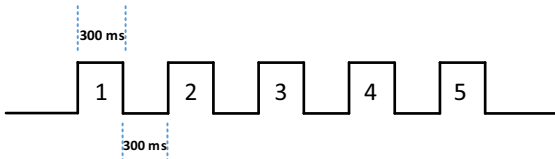
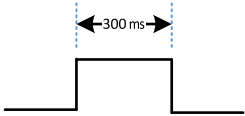
TABLE 5-2: RN1810 INPUT PINS

RN1810 Input Pin	Description
CMD_CTRL Pin 20	Rising edge forces RN1810 into Command mode (identical to \$\$\$ command). Falling edge forces RN1810 into Data mode (identical to exit command).
FUNC_CONFIG Pin 14	Selects different RN1810 modes. Refer to Section 5.2 “I/O Pin Function Select” .
TCP_CTRL Pin 22	This input can be used in two contexts: - Set by Host to command WiFly to open or close TCP connection. 0: Close TCP connection (identical to <code>close</code> command) 1: Open TCP connection (identical to <code>open</code> command) - This pin is <i>required</i> for FTP transfers. Refer to Section 4.8“FTP Client”.
RESET Pin 19	Set by Host to force WiFly module reset (identical to the <code>reboot</code> command). To force a WiFly reset (presuming CHP_PWD_L is normally high): - Set RESET low - Delay at least 650 ns - Set RESET high Note 1: This pin must be configured as open drain or the RN1810 <i>cannot</i> drive this pin low when it goes into Sleep state, and the Sleep state do <i>not</i> work correctly. 2: If the module is in Sleep mode it must be first taken out of Sleep mode via the WAKEUP pin. Then this RESET pin can be toggled.
SLEEP Pin 13	Rising edge puts module into Sleep state for number of seconds specified in the last <code>set sys wake <value></code> command. Identical to the <code>sleep</code> command.
UART0_CTS Pin 9	WiFly UART CTS pin (connected to Host UART RTS pin)
UART0_RX/ MODE0 Pin 21	WiFly UART RX pin. <i>Required.</i>
WAKEUP Pin 25	Takes WiFly out of Sleep state. The Host must default this pin high, and set it to 0 to take the RN1810 out of the Sleep state. The WAKEUP pin can be shorted to the UART0_RX pin only if the application wakes up a sleeping RN1810 when it receives a character.

5.2 I/O PIN FUNCTION SELECT

The FUNC_CONFIG pin, either stand-alone or in conjunction with the RESET pin, can be used for selecting RN1810 modes of operation as shown in [Table 5-3](#).

TABLE 5-3: I/O FUNCTION SELECT

Function	I/O Sequence
Enter Soft AP mode	This I/O sequence forces the RN1810 into Soft AP mode: 1. Set FUNC_CONFIG high 2. Set RESET low 3. Set RESET high 4. RN1810 enters Soft AP mode. Wait for one second or for the RN1810 to output: <code>softap ssid: WiFly-Rn1810-xy</code> [where xy is the last byte of the module's MAC address] 5. Set FUNC_CONFIG low
Perform Factory RESET	This I/O sequence performs the same function as issuing the <code>factory RESET</code> command. 1. RN1810 must be in Soft AP mode for a minimum of 100 ms 2. Set FUNC_CONFIG low to high five times with 300 ms between transitions as shown below:  3. RN1810 performs factory Reset 4. Send <code>reboot</code> command
Launch Web Application	This I/O sequence launches the web application. 1. RN1810 must be in Client mode 2. Set FUNC_CONFIG high 3. Delay 5 seconds 4. Web Application starts 5. Wait for RN1810 output: <code>HTTP SERVER start successful</code> 6. Set FUNC_CONFIG low
Connect to AP with WPS	This I/O sequence allows the RN1810 to connect via WPS. 1. Configure the AP for WPS 2. Set FUNC_CONFIG low to high once with 300ms between transitions as shown below:  3. RN1810 enters WPS mode 4. Press WPS button on AP 5. RN1810 connects to AP

Appendix A. Command Quick Reference Guide

A.1 DEFAULT CONFIGURATION SETTINGS

This section describes the default configuration settings and how to restore them. [Table A-1](#) summarizes all the commands and the default values are described in [Chapter 2. "Command Reference"](#).

TABLE A-1: COMMAND QUICK REFERENCE GUIDE

Command	Default	Description
Set Commands		
set apmode		
set apmode beacon <value>	102	Sets the SOFT AP network beacon interval in milliseconds.
set apmode channel <value>	1	Sets the Soft AP channel number.
set apmode passphrase <string>	Null	Sets the Soft AP passphrase.
set apmode ssid <string>	Null	Sets the Soft AP SSID.
set comm		
set comm \$ <char>	\$	Sets the character used to enter Command mode.
set comm close <string>	*CLOS*	Sets the string sent to host UART when TCP connection closed.
set comm idle <value>	0	Sets the timeout (in seconds) to close a TCP connection when idle.
set comm match <value> <flag>	0	Sets the match character used to flush TX data to Wi-Fi network.
set comm open <string>	*OPEN*	Sets the string sent to host UART when TCP connection opened.
set comm remote <string>	*HELLO*	Sets the string sent to remote host when TCP connection opened.
set comm size <value>	1420	Sets the number of RX bytes used to flush TX data to Wi-Fi network.
set comm timer <value>	100	Sets the number of ms used to flush TX data to Wi-Fi network.
set dhcp		
set dhcp hostname <string>	RN1810_xy	Sets the host name for the RN1810 module.
set dhcp lease <start_ip_address> <end_ip_address> <lease_time>	<start_ip_address> 192.168.1.11 <end_ip_address> 192.168.1.20 <lease time> 86400 seconds	sets the DHCP pool and lease time when the RN1810 is put in Soft AP mode.

TABLE A-1: COMMAND QUICK REFERENCE GUIDE (CONTINUED)

Command	Default	Description
set dns		
set dns address <address>	0.0.0.0	Sets the IP address of DNS server.
set dns name <string>	server1	Sets the name of the host for TCP/IP connections.
set ftp		
set ftp addr <address>	0.0.0.0	Sets the FTP server IP address.
set ftp dir <string>	.	Sets the starting directory on FTP server.
set ftp filename <string>	test_file	Sets the file name to access on FTP server.
set ftp password <string>	Pass123	Sets the password for FTP server.
set ftp remote <value>	21	Sets the port number for FTP server.
set ftp timeout <value>	10	Sets the FTP server connection timeout (in seconds).
set ftp user <string>	mchp	Sets the user name for FTP server.
set ip		
set ip address <address>	0.0.0.0 (IPv4) :: (IPv6)	Sets the WiFly static IP address.
set ip dhcp <value>	1	Sets the backup remote host IP address.
set ip host <address>	0.0.0.0 (IPv4) :: (IPv6)	Sets the remote host IP address.
set ip localport <value>	2000	Sets the local port number.
set ip netmask <address>	255.255.255.0	Sets the network mask.
set ip protocol <flag>	0x02	Sets IP protocol.
set ip remote <value>	0	Sets the remote host port number.
set ip version <value>	0	Sets the IP version.
set opt		
set opt replace <value>	\$	Sets replacement character for space characters in the SSID or passphrase.
set sys		
set sys auto <value>	0	Sets HTTP client auto-connect timer in seconds.
set sys autoconn <value>	0	Sets TCP client periodic connection timer.
set sys autosleep <value>	0	Sets UDP auto-sleep timer multiplier.
set sys sleep <value>	0	Sets duration, in seconds, module is awake before sleeping.
set sys wake <value>	0	Sets duration, in seconds, module is asleep before awaking.
set time		
set time address <string>	pool.ntp.org	Sets the string name of SNTP server.
set time enable <value>	0	Sets period, in seconds, of fetches from SNTP server.
set time zone <value>	UTC-07:00,E	Sets the time zone adjustment of time fetched from STNP server.
set uart		
set uart baud <value>	9600	Sets the UART baud rate.
set uart flow <value>	0x00	Sets the UART flow control.
set uart instant <value>	N/A	Sets the UART instant baud rate.
set uart mode <value>	0	Sets UART mode.
set uart raw <value>	N/A	Sets custom UART baud rate.

TABLE A-1: COMMAND QUICK REFERENCE GUIDE (CONTINUED)

Command	Default	Description
set wlan		
set wlan auth <value>	0	Sets Wi-Fi authentication mode.
set wlan hide <value>	0	Displays or masks passphrase.
set wlan join <value>	0	Sets Wi-Fi association policy.
set wlan key <value>	0	Sets the WEP key.
set wlan mask <mask>	All channels	Sets scan channel mask.
set wlan mode_phy <value>	0	Sets the Wireless Physical mode.
set wlan number <value>	1	Sets the WEP key index.
set wlan phrase <string>	!microchip	Sets WPA/WPA2 passphrase.
set wlan ssid <string>	microchip1	Sets Wi-Fi network SSID.
set wlan tx <value>	16	Sets Wi-Fi fixed transmit power level.
Get Commands		
get console	—	Output sconsole settings.
get dns	—	Outputs DNS settings.
get everything	—	Output sa variety of settings.
get ftp	—	Output sFTP settings.
get ip	—	Outputs IP settings.
get mac	—	Outputs the MAC address.
get softap	—	Outputs Soft AP settings.
get system	—	Outputs system settings.
get time	—	Outputs SNTP client settings.
get uart	—	Outputs UART settings.
get version	—	Outputs the firmware version.
get wlan	—	Outputs Wi-Fi settings.
Action Commands		
\$\$\$	—	Enters Command mode.
apmode <ssid> <channel>	—	Initiates Soft AP mode.
close	—	Closes TCP connection.
exit	—	Exits Command mode (to Data mode).
factory RESET	—	Restores all default configurations (must reboot).
ftp get	—	Reads file from FTP server.
ftp put	—	Writes file to FTP server.
join <string>	—	Joins an Wi-Fi network.
leave	—	Leaves Wi-Fi network.
lookup <string>	—	Performs DSN query.
open <host> <port_number>	—	Opens a TCP client, HTTP client, or TLS TCP client connection.
ota upgrade <file_name> <server_addr>	—	Performs and over-the-air firmware upgrade.
ping <address>	—	Performs IPv4 ping.
ping6 <address>	—	Performs IPv6 ping.
reboot	—	Reboots WiFly module.
release	—	Clears DHCP server entries (only in Soft AP mode).

TABLE A-1: COMMAND QUICK REFERENCE GUIDE (CONTINUED)

Command	Default	Description
<code>rfctest <rate> <num_tries> <num_bytes> <channel> <header type> [addr1] [addr2] [addr3] [addr4]</code>	—	Raw mode transmission of Wi-Fi packets.
<code>run <string></code>	—	Runs the specified application.
<code>save</code>	—	Saves current configurations to module FLASH.
<code>scan</code>	—	Scans for existing Wi-Fi networks.
<code>show <value></code>	—	Refer to show commands.
<code>sleep</code>	—	Puts the WiFly module in Sleep mode.
<code>time</code>	—	Sets real-time clock by running SNTP client.
Show Commands		
<code>show ap</code>	—	Outputs devices connected to Soft AP network.
<code>show io</code>	—	Outputs RN1810 pin states.
<code>show ip</code>	—	Outputs WiFly IP state.
<code>show rssi</code>	—	Outputs current RSSI value.
<code>show net</code>	—	Outputs module's network state.
<code>show time</code>	—	Outputs time fetched from NTP server.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://www.microchip.com/support>
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820