



# **NRC7292 Evaluation Kit**

## **User Guide**

### **(AT Command)**

**Ultra-low power & Long-range Wi-Fi**

**Ver 1.23.4**  
**Jun. 30, 2023**

**NEWRACOM, Inc.**

## **NRC7292 Evaluation Kit User Guide (AT-command) Ultra-low power & Long-range Wi-Fi**

**© 2023 NEWRACOM, Inc.**

All right reserved. No part of this document may be reproduced in any form without written permission from Newracom.

Newracom reserves the right to change in its products or product specification to improve function or design at any time without notice.

### **Office**

Newracom, Inc.

505 Technology Drive, Irvine, CA 92618 USA

<http://www.newracom.com>

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Overview.....</b>                                   | <b>7</b>  |
| <b>2</b> | <b>Basic Setup.....</b>                                | <b>7</b>  |
| 2.1      | Hardware connection.....                               | 7         |
| 2.2      | Downloading the firmware and starting the module ..... | 12        |
| <b>3</b> | <b>AT Command Type .....</b>                           | <b>13</b> |
| <b>4</b> | <b>Return for Commands .....</b>                       | <b>14</b> |
| <b>5</b> | <b>Basic AT Commands .....</b>                         | <b>15</b> |
| 5.1      | AT .....   | 16        |
| 5.2      | ATE .....  | 16        |
| 5.3      | ATZ.....   | 16        |
| 5.4      | AT+VER.....  | 16        |
| 5.5      | AT+UART .....  | 17        |
| 5.6      | AT+GPIOCONF .....                                      | 18        |
| 5.7      | AT+GPIOVAL.....  | 19        |
| 5.8      | AT+ADC .....   | 20        |
| <b>6</b> | <b>Wi-Fi AT Commands .....</b>                         | <b>22</b> |
| 6.1      | AT+WMACADDR.....                                       | 24        |
| 6.2      | AT+WOUNTRY.....  | 24        |
| 6.3      | AT+WTXPOWER .....                                      | 25        |
| 6.4      | AT+WRXSIG .....  | 26        |
| 6.5      | AT+WRATECTRL .....                                     | 27        |
| 6.6      | AT+WMCS .....  | 28        |
| 6.7      | AT+WDUTYCYCLE .....                                    | 29        |
| 6.8      | AT+WCCATHRESHOLD.....                                  | 30        |
| 6.9      | AT+WTXTIME .....                                       | 30        |
| 6.10     | AT+WTSF .....  | 31        |
| 6.11     | AT+WSCAN .....   | 32        |
| 6.12     | AT+WCONN .....   | 35        |
| 6.13     | AT+WDISCONN.....                                       | 37        |
| 6.14     | AT+WSOFTAP .....                                       | 37        |
| 6.15     | AT+WBSSMAXIDLE .....                                   | 39        |
| 6.16     | AT+WSTAINFO.....                                       | 40        |
| 6.17     | AT+WIPADDR .....                                       | 41        |
| 6.18     | AT+WDNS .....  | 42        |
| 6.19     | AT+WDHCP.....  | 43        |
| 6.20     | AT+WDHCPS.....   | 44        |
| 6.21     | AT+WPING.....  | 44        |
| 6.22     | AT+WDEEPSLEEP .....                                    | 45        |

|  |           |
|--|-----------|
| 6.23 AT+WFOTA .....                                | 47        |
| 6.24 AT+WTIMEOUT .....                             | 52        |
| 6.25 +WEVENT .....                                 | 53        |
| <b>7 Socket AT Commands.....</b>                   | <b>55</b> |
| 7.1 AT+SOPEN .....                                 | 56        |
| 7.2 AT+SCLOSE .....                                | 57        |
| 7.3 AT+SLIST .....                                 | 57        |
| 7.4 AT+SSEND.....                                  | 58        |
| 7.5 AT+SRECV .....                                 | 61        |
| 7.6 AT+SRECVMODE.....                              | 62        |
| 7.7 AT+SRECVINFO .....                             | 63        |
| 7.8 AT+SADDRINFO .....                             | 64        |
| 7.9 AT+STCPNODELAY .....                           | 64        |
| 7.10 AT+STIMEOUT .....                             | 66        |
| 7.11 +SEVENT .....                                 | 67        |
| 7.12 +RXD .....                                    | 68        |
| <b>8 Test Application .....</b>                    | <b>70</b> |
| 8.1 Command Line Interface (raspi-atcmd-cli) ..... | 70        |
| 8.1.1 Source files .....                           | 73        |
| 8.1.2 Build.....                                   | 73        |
| 8.1.3 Run .....                                    | 74        |
| 8.1.4 Run with a script.....                       | 78        |
| 8.1.5 lperf .....                                  | 80        |
| 8.2 Remote Server/Client (raspi-atcmd-remote)..... | 90        |
| 8.2.1 Source files .....                           | 90        |
| 8.2.2 Build.....                                   | 90        |
| 8.2.3 Run .....                                    | 91        |
| <b>9 Revision History .....</b>                    | <b>92</b> |

# List of Tables

Table 3.1 AT-command type ..... 13

Table 8.1 raspi-atcmd-cli source files..... 73

Table 8.2 raspi-atcmd-remote source files ..... 90

# List of Figures

|            |  |    |
|------------|--|----|
| Figure 2.1 | NRC7292 evaluation board .....                         | 7  |
| Figure 2.2 | UART connection between EVK and external host.....     | 8  |
| Figure 2.3 | SPI connection between EVK and external host.....      | 9  |
| Figure 8.1 | Pin map of 40-pin header for Raspberry Pi 3 B/B+ ..... | 70 |
| Figure 8.2 | Connection between NRM7292 EVB and Raspberry Pi .....  | 71 |

# 1 Overview

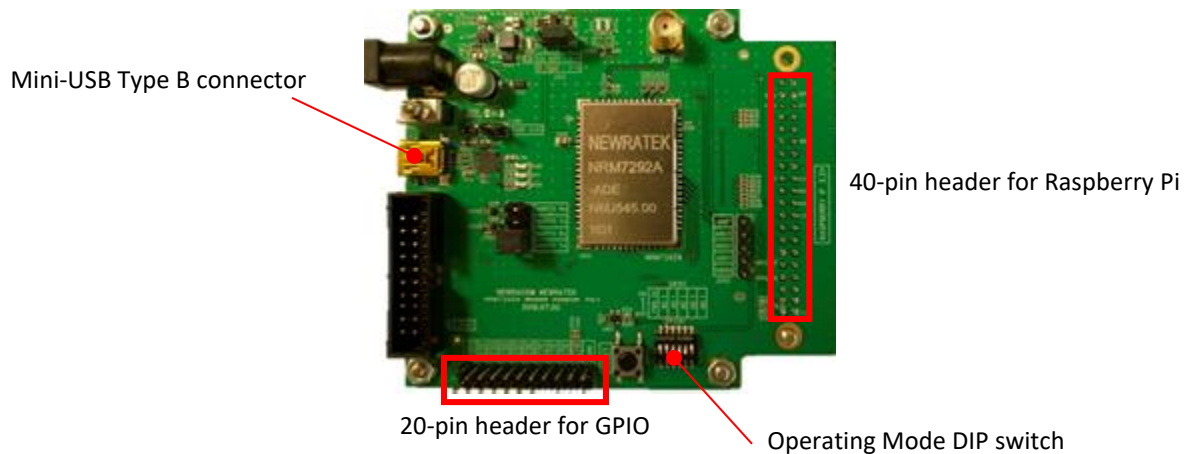
This document introduces the NRC7292 AT-command. The NRC7292 AT-command allows users to apply fine controls over the NRC7292 modules such as: checking the modem status, scanning, connecting to an AP, opening sockets, and exchanging data.

## 2 Basic Setup

The AT-command package with a custom firmware binary to enable AT-command feature is required along with the firmware download tool. Users need to download the firmware binary onto the flash on the NRC7292 module to enable AT-command communication via UART or SPI.

### 2.1 Hardware connection

Figure 2.1 shows an NRC7292 evaluation board (EVB). The AT-command communication is achieved via the UART or SPI interface between an external host and the EVB.

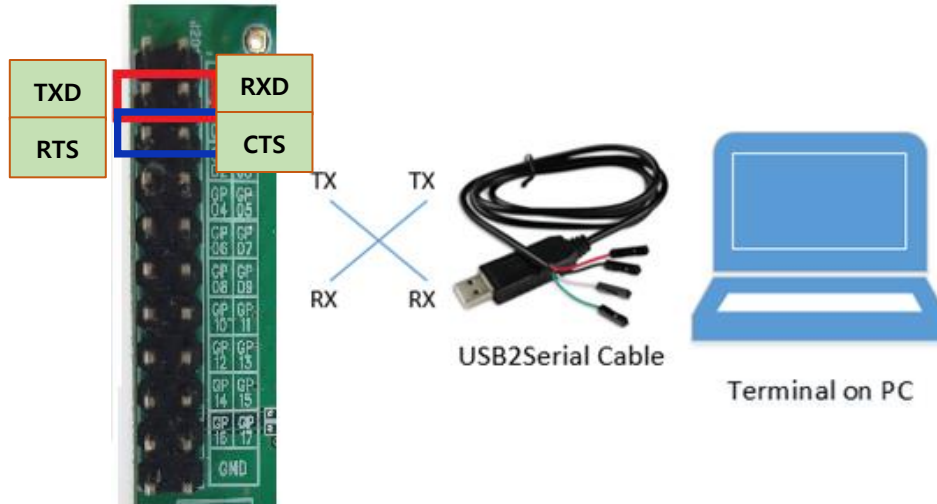


**Figure 2.1 NRC7292 evaluation board**

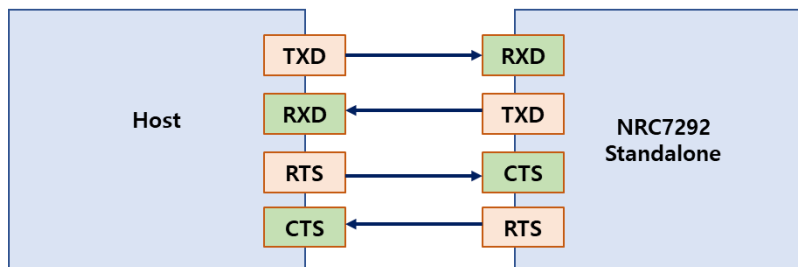
**IMPORTANT:** If the EVB is mounted on a Raspberry Pi host, detach the board from the Raspberry Pi host first before proceeding. The EVB must be used as a standalone for stable AT communication.

## 1) UART

The AT-command uses UART channel 2. The TX and RX of UART channel 2 are placed in a 20-pin header for GPIO.



**Module-to-Host Connection  
(with flow control)**



**Figure 2.2 UART connection between EVK and external host**

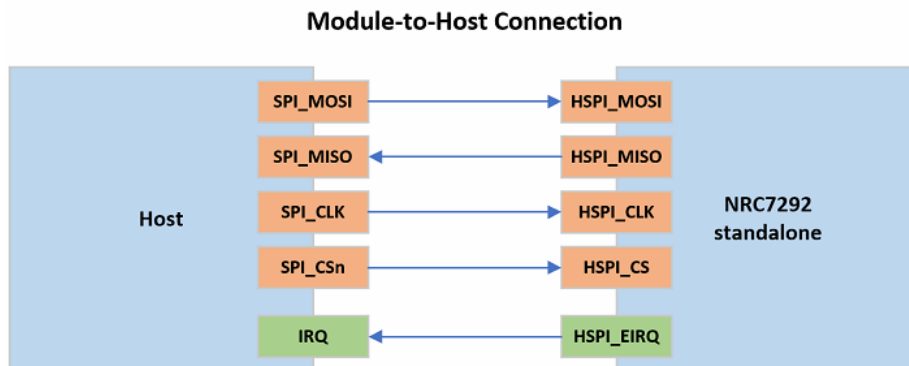
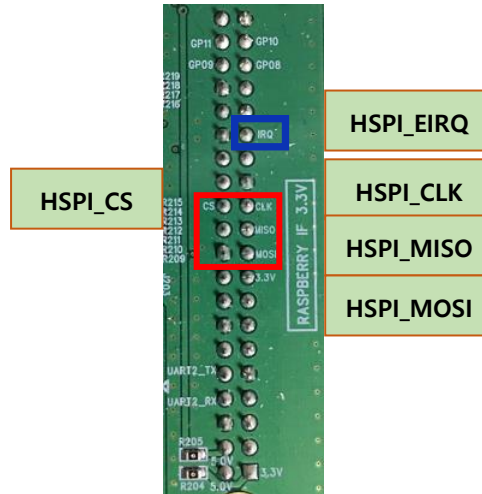
The GP00 and GP01 pins on the 20-pin header correspond to TX and RX of UART channel 2, respectively.

※ (Optional) GP02 and GP03 pins correspond to RTS and CTS of UART channel 2 for HW Flow Control



## 2) HSPI

The NRC7292 has a dedicated SPI slave controller for high speed. The SPI signals are placed in a 40-pin header for Raspberry Pi. The CLI application described in chapter 8 is available to perform AT-command communication via the SPI on Raspberry Pi.



**Figure 2.3** SPI connection between EVK and external host

To perform AT command communication through SPI on Raspberry Pi, spidev of Raspberry Pi must be enabled.

1. Modify /boot/config.txt and enable spi hardware interface configuration.

```
# Uncomment some or all of these to enable the optional hardware interfaces
#dtparam=i2c_arm=on
#dtparam=i2c=on
dtparam=spi=on

# Uncomment this to enable the lirc-rpi module
#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on
enable_uart=1
dtoverlay=pi3-disable-bt
dtoverlay=pi3-disable-wifi
#dtoverlay=pi3-disable-spidev
```

2. After rebooting the Raspberry Pi, spidev0.0 and spidev0.1 could be accessible from the userspace.

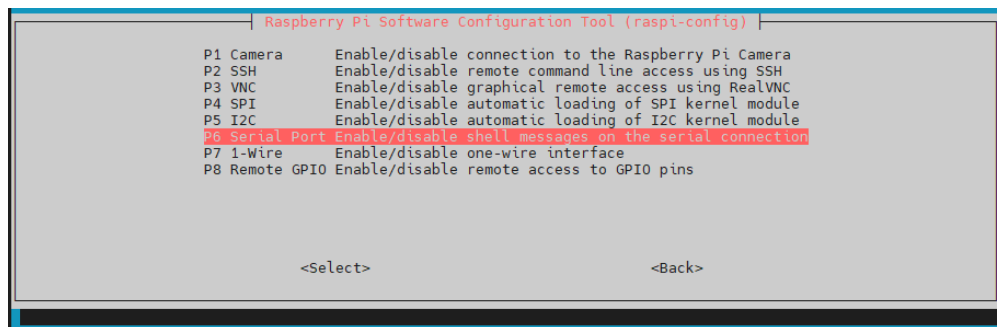
```
pi@raspberrypi:~$ ls /dev
autofs          gpiochip2  loop7      ram0       random     tty11      tty26      tty40      tty55      uhid        vcsa2
block           gpiomem   loop-control ram1       raw         tty12      tty27      tty41      tty56      uinput      vcsa3
btrfs-control   hidraw0    mapper     ram10      rfkill      tty13      tty28      tty42      tty57      urandom     vcsa4
bus             hidraw1    mem        ram11      serial0     tty14      tty29      tty43      tty58      vchiq       vcsa5
cachefiles      hwrng     memory_bandwidth ram12      serial1     tty15      tty3       tty44      tty59      vcio        vcsa6
char            initctl   mmcblk0    ram13      shm         tty16      tty30      tty45      tty6       vc-mem      vcsa7
console         input     mmcblk0p1  ram14      sd          tty17      tty31      tty46      tty60      vcs         vcs8
cpu_dma_latency kmsg      mmcblk0p2  ram15      spidev0.0  tty18      tty32      tty47      tty61      vcs1        vchi
cuse            log        queue     ram2       spidev0.1  tty19      tty33      tty48      tty62      vcs2        watchdog
disk           loop0     net        ram3      stdin       tty2       tty34      tty49      tty63      vcs3        watchdog0
fb0            loop1     network_latency ram4      stdout      tty20      tty35      tty5       tty7       vcs4        zero
fd             loop2     network_throughput ram5      tty         tty21      tty36      tty50      tty8       vcs5
full          loop3     null       ram6      tty         tty22      tty37      tty51      tty9       vcs6
fuse          loop4     ppp        ram7      tty0        tty23      tty38      tty52      ttyAMA0    vcs7
gpiochip0     loop5     ptmx       ram8      tty1        tty24      tty39      tty53      ttyprintk  vcsa
gpiochip1     loop6     pts        ram9      tty10       tty25      tty4       tty54      tty50      vcsa1
```

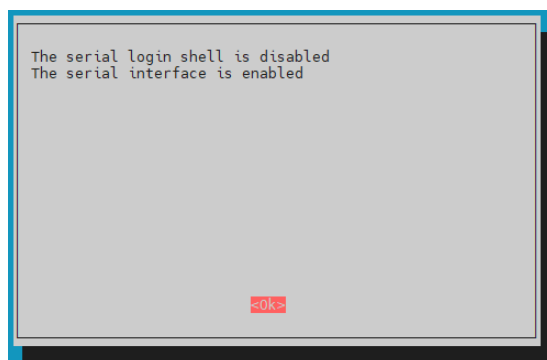
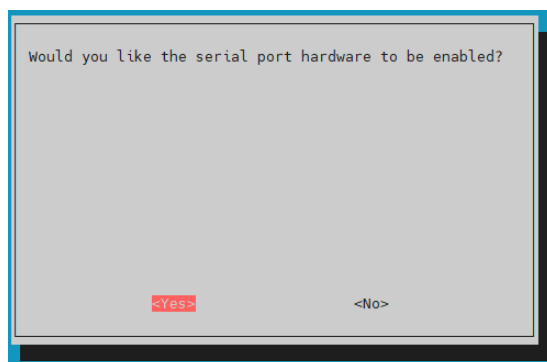
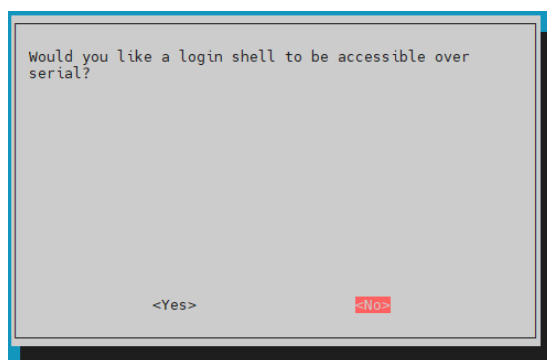
3. Enable SPI

The Raspberry Pi configuration tool can also be run from the command line.

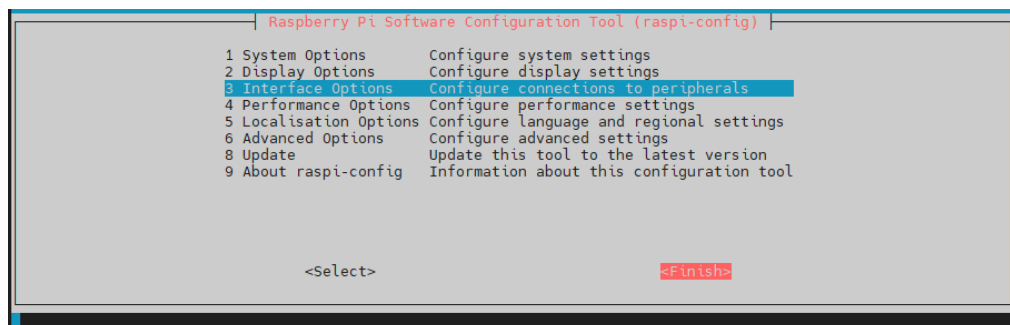
```
# sudo raspi-config
```

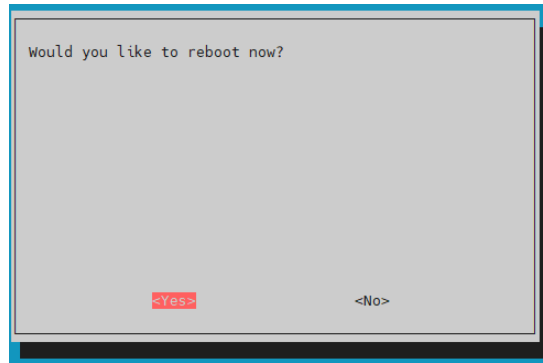
- Enable Serial Port





- Fininsh and reboot





## 2.2 Downloading the firmware and starting the module

Refer to the user guide **UG-7292-004-Standalone SDK.pdf** for instructions on how to download the firmware binary. (3.2 Download the unified binary)

### 3 AT Command Type

There are four types of AT-commands: HELP, GET, SET and RUN.

| Type       | Format                      | Description  |
|------------|-----------------------------|--|
| HELP       | AT+<CMD>=?                  | List the input argument format and description.          |
| SET or RUN | AT+<CMD>                    | Run with no argument.                                    |
|            | OR<br>AT+<CMD>=<X1,X2,...>  | OR<br>Set or run with the given arguments.               |
| GET        | AT+<CMD>?                   | Query the current values with no argument.               |
|            | OR<br>AT+<CMD>?=<X1,X2,...> | OR<br>Query the current values with the given arguments. |

Table 3.1 AT-command type

- String input parameter values must be enclosed between double quotation marks (“”).
- Parameters enclosed between a pair of square brackets ‘[]’ indicate optional parameters.
- Optional parameters may be nested.
- All AT commands must be in upper-case letters and terminated by CR-LF.
- Default optional values in the parameter descriptions are indicated by the asterisk ‘\*’ characters.

## 4 Return for Commands

| Return Message    | Description  |
|-------------------|--|
| OK                | The operation for command completes successfully.                        |
| ERROR             | The command is not supported.  |
| +<CMD>:1<br>ERROR | The parameter for command is not valid.                                  |
| +<CMD>:2<br>ERROR | The previous operation for command is in progress.                       |
| +<CMD>:3<br>ERROR | The operation for command failed with some error.                        |
| +<CMD>:4<br>ERROR | The operation for command is still in progress after the specified time. |

## 5 Basic AT Commands

| Commands    | Description   |
|-------------|---|
| AT          | Check the AT serial interface status.                       |
| ATE         | Enable or disable echo.                                     |
| ATZ         | Reset the hardware and restart the firmware.                |
| AT+VER      | Fetch the AT firmware version and software package version. |
| AT+UART     | Configure the serial UART parameters.                       |
| AT+GPIOCONF | Configure the GPIO pin mode, direction and pull-up option.  |
| AT+GPIOVAL  | Read or write the output GPIO pin level.                    |
| AT+ADC      | Fetch the ADC value at the selected ADC channel index.      |

## 5.1AT

|                    |                                       |
|--------------------|---------------------------------------|
| <b>Command</b>     | AT                                    |
| <b>Response</b>    | OK                                    |
| <b>Description</b> | Check the AT serial interface status. |
| <b>Example</b>     | AT<br>OK                              |

## 5.2ATE

|                    |  |
|--------------------|--|
| <b>Command</b>     | ATE0 or ATE1   |
| <b>Response</b>    | OK   |
| <b>Description</b> | Enable (ATE1) or disable (ATE0) echo. (default: disable)<br><br>NOTE:<br>Echo should typically be enabled for manual communication via a terminal. |
| <b>Example</b>     | ATE1<br>OK<br><br>ATE0<br>OK   |

## 5.3ATZ

|                    |  |
|--------------------|--|
| <b>Command</b>     | ATZ  |
| <b>Response</b>    |  |
| <b>Description</b> | Reset the hardware and restart the firmware. |
| <b>Example</b>     | ATZ  |

## 5.4AT+VER

|                 |   |
|-----------------|---|
| <b>Command</b>  | <u>GET</u><br>AT+VER?                               |
| <b>Response</b> | <u>GET</u><br>+VER: <sdk_version>,<command_version> |



|                    |  |
|--------------------|--|
|                    | OK   |
| <b>Description</b> | Fetch the version information of current firmware. |
| <b>Example</b>     | AT+VER?<br>+VER:"1.4.0","1.23.5"<br>OK             |

## 5.5 AT+UART

|                    |  |
|--------------------|--|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+UART=<baud_rate>,<HFC><br><u><b>GET</b></u><br>AT+UART?  |
| <b>Response</b>    | <u><b>SET</b></u><br>OK<br><u><b>GET</b></u><br>+UART:<baud_rate>,<data_bits>,<stop_bits>,<parity>,<HFC><br>OK   |
| <b>Parameters</b>  | <p><b>&lt;baud rate&gt;</b><br/>9600, 19200, 38400, 57600, 115200*,<br/>230400, 460800, 500000, 576000, 921600, 1000000,<br/>1152000, 1500000, 2000000</p> <p><b>&lt;data bits&gt;</b><br/>Always 8 (8-bit)*</p> <p><b>&lt;stop bits&gt;</b><br/>Always 1 (1-bit)*</p> <p><b>&lt;parity&gt;</b><br/>Always 0 (None)*</p> <p><b>&lt;HFC&gt;</b><br/>0 : RTS/CTS disabled*<br/>1 : RTS/CTS enabled</p> |
| <b>Description</b> | Configure the baud rate and HFC for the UART.  |

|                |  |
|----------------|--|
| <b>Example</b> | AT+UART=115200,1<br>OK<br><br>AT+UART?<br>+UART:115200,8,1,0,1<br>OK |
|----------------|--|

## 5.6 AT+GPIOCONF

|                    |  |
|--------------------|--|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+GPIOCONF=<number>,<direction>,<pull-up><br><br><u><b>GET</b></u><br>AT+GPIOCONF?<br>AT+GPIOCONF?=<number>  |
| <b>Response</b>    | <u><b>SET</b></u><br>OK<br><br><u><b>GET</b></u><br>+GPIOCONF=<number>,<direction>,<pull-up><br>:<br>OK  |
| <b>Parameters</b>  | <b>&lt;number&gt;</b><br>GPIO pin number. (8, 9, 10, 11, 12, 13, 14, 15, 16, 17)<br><br><b>&lt;direction&gt;</b><br>0 : input<br>1 : output<br><br><b>&lt;pull-up&gt; (input pin only)</b><br>0 : pull-down<br>1 : pull-up |
| <b>Description</b> | Configure the GPIO pin direction and pull-up option.   |
| <b>Example</b>     | AT+GPIOCONF?<br>+GPIOCONF:8,1,0<br>+GPIOCONF:9,1,0<br>+GPIOCONF:10,1,0<br>+GPIOCONF:11,1,0   |

|  |   |
|--|---|
|  | +GPIOCONF:12,1,0<br>+GPIOCONF:13,1,0<br>+GPIOCONF:14,1,0<br>+GPIOCONF:15,1,0<br>+GPIOCONF:16,1,0<br>+GPIOCONF:17,1,0<br>OK<br><br>AT+GPIOCONF=10,0,1<br>OK<br><br>AT+GPIOCONF?=10<br>+GPIOCONF:10,0,1<br>OK |
|--|---|

## 5.7 AT+GPIOVAL

|                    |   |
|--------------------|---|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+GPIOVAL=<number>,<level><br><u><b>GET</b></u><br>AT+GPIOVAL?<br>AT+GPIOVAL?=<number>                        |
| <b>Response</b>    | <u><b>SET</b></u><br>OK<br><u><b>GET</b></u><br>+GPIOVAL:<number>,<level><br>OK   |
| <b>Parameters</b>  | <b>&lt;number&gt;</b><br>GPIO pin number. (8, 9, 10, 11, 12, 13, 14, 15, 16, 17)<br><br><b>&lt;level&gt;</b><br>0 : low<br>1 : high |
| <b>Description</b> | Read or write the output GPIO pin level.  |
| <b>Example</b>     | AT+GPIOVAL?<br>+GPIOVAL:8,1   |

|  |  |
|--|--|
|  | <div>+GPIOVAL:9,1<br/>+GPIOVAL:10,1<br/>+GPIOVAL:11,1<br/>+GPIOVAL:12,1<br/>+GPIOVAL:13,1<br/>+GPIOVAL:14,1<br/>+GPIOVAL:15,1<br/>+GPIOVAL:16,1<br/>+GPIOVAL:17,1<br/>OK<br/><br/>AT+GPIOVAL=9,0<br/>OK<br/><br/>AT+GPIOVAL?=9<br/>+GPIOVAL:9,0<br/>OK</div> |
|--|--|

## 5.8AT+ADC

|             |  |
|-------------|--|
| Command     | <b><u>GET</u></b><br>AT+ADC?=<channel>                                   |
| Response    | <b><u>GET</u></b><br>+ADC:<channel>,<value><br>OK                        |
| Parameters  | <b>&lt;channel&gt;</b><br>1, 2, 3<br><br><b>&lt;value&gt;</b><br>0 ~ 511 |
| Description | Fetch the ADC value at the selected ADC channel.                         |
| Example     | AT+ADC?<br>+ADC:1,396<br>+ADC:2,384<br>+ADC:3,400<br>OK                  |

|  |                               |
|--|-------------------------------|
|  | AT+ADC?=3<br>+ADC:3,408<br>OK |
|--|-------------------------------|

## 6 Wi-Fi AT Commands

| Commands         | Description   |
|------------------|---|
| AT+WMACADDR      | Read the MAC address.   |
| AT+WOUNTRY       | Configure the Wi-Fi country code                                |
| AT+WTXPOWER      | Configure the transmission power level.                         |
| AT+WRXSIG        | Fetch or monitor the RSSI (dBm) and SNR (dB) values.            |
| AT+WRATECTRL     | Toggle the MCS rate control option.                             |
| AT+WMCS          | Set the MCS index applied when the rate control is disabled.    |
| AT+WDUTYCYCLE    | Configure duty cycle operation.                                 |
| AT+WCCATHRESHOLD | Set CCA threshold.  |
| AT+WTXTIME       | Set carrier sense time and pause time.                          |
| AT+WTSF          | Read the elapsed TSF timer duration.                            |
| AT+WSCAN         | Perform Wi-Fi scanning.   |
| AT+WCONN         | Connect to a new AP.  |
| AT+WDISCONN      | Disconnect from the AP or abort an on-going connection process. |
| AT+WSOFTAP       | Run as the AP mode.   |
| AT+WBSSMAXIDLE   | Configure the BSS Max idle service for SoftAP.                  |
| AT+WSTAINFO      | Get information of associated STAs on AP mode.                  |
| AT+WIPADDR       | Configure the IPv4 address.                                     |
| AT+WDNS          | Configure the IP address for the DNS server.                    |
| AT+WDHCP         | Request dynamic IP allocation from the DHCP server.             |
| AT+WDHCPS        | Run the DHCP sever in SoftAP mode.                              |
| AT+WPING         | Send ICMP ECHO_REQUEST to network hosts with IPv4 address.      |
| AT+WDEEPSLEEP    | Configure deep-sleep mode to save power.                        |
| AT+WFOTA         | Enable or disable Firmware Over-the-Air (FOTA).                 |
| AT+WTIMEOUT      | Configure the response timeout for the specified command.       |

|         |   |
|---------|---|
| +WEVENT | Asynchronously raised Wi-Fi event logs. |
|---------|---|

## 6.1 AT+WMACADDR

|                    |   |
|--------------------|---|
| <b>Command</b>     | <u>GET</u><br>AT+WMACADDR?  |
| <b>Response</b>    | <u>GET</u><br>+WMACADDR:"<MAC address>"<br>OK   |
| <b>Parameters</b>  | <b>&lt;MAC address&gt;</b><br>The MAC address 'HH:HH:HH:HH:HH:HH' where H is a hexadecimal character. |
| <b>Description</b> | Read the MAC address.   |
| <b>Example</b>     | AT+ WMACADDR?<br>+WMACADDR:"2F:33:4F:65:11:20"<br>OK  |

## 6.2 AT+WCCOUNTRY

|                   |  |
|-------------------|--|
| <b>Command</b>    | <u>SET</u><br>AT+WCCOUNTRY="<country code>"<br><u>GET</u><br>AT+WCCOUNTRY?   |
| <b>Response</b>   | <u>SET</u><br>OK<br><u>GET</u><br>+WCCOUNTRY="<country code>"<br>OK  |
| <b>Parameters</b> | <b>&lt;country code&gt;</b> <ul style="list-style-type: none"> <li>- AU : Australia</li> <li>- CN : China</li> <li>- EU : Europe</li> <li>- JP : Japan</li> <li>- NZ : New Zealand</li> <li>- TW : Taiwan</li> <li>- US : United States</li> <li>- K0 : Korea USN (2M BW support)</li> <li>- K1 : Korea USN</li> <li>- K2 : Korea MIC</li> </ul> |



|                    |  |
|--------------------|--|
| <b>Description</b> | Configure the Wi-Fi country code.<br><br>NOTE:<br>The country code may need to be set after booting. |
| <b>Example</b>     | AT+ WOUNTRY ="US"<br>OK<br><br>AT+WOUNTRY?<br>+WOUNTRY:"US"<br>OK                                    |

### 6.3 AT+WTXPOWER

|                    |  |
|--------------------|--|
| <b>Command</b>     | <u>SET</u><br>AT+WTXPOWER=<tx_power><br><u>GET</u><br>AT+WTXPOWER?   |
| <b>Response</b>    | <u>SET</u><br>OK<br><u>GET</u><br>+WTXPOWER:<tx_power>   |
| <b>Parameters</b>  | <b>&lt;tx power&gt;</b><br>Transmission Power Level (unit : dBm) (1 ~ 30)  |
| <b>Description</b> | Configure the transmission power level.<br>Set to 0 to use AUTO mode, not FIXED mode.<br>AUTO mode sets TX power automatically according to MCS.<br>Default is AUTO mode.<br><br>NOTE:<br>Depending on the country and channel frequency, the maximum allowed TX power may be limited to less than 30 dBm. |
| <b>Example</b>     | AT+WTXPOWER?<br>+WTXPOWER:30<br>OK<br><br>AT+WTXPOWER=17   |

|  |   |
|--|---|
|  | OK<br><br>AT+WTXPOWER?<br>+WTXPOWER:17<br>OK<br><br>AT+WTXPOWER=0<br>OK<br><br>AT+WTXPOWER?<br>+WTXPOWER:30<br>OK |
|--|---|

## 6.4 AT+WRXSIG

|                    |   |
|--------------------|---|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+WRXSIG =<time><br><u><b>GET</b></u><br>AT+WRXSIG?   |
| <b>Response</b>    | <u><b>SET</b></u><br>+WRXSIG:<RSSI>,<SNR><br>...<br>+WRXSIG:<RSSI>,<SNR><br>OK<br><u><b>GET</b></u><br>+WRXSIG:<RSSI>,<SNR><br>OK |
| <b>Parameters</b>  | <time><br>Monitoring duration in seconds.   |
| <b>Description</b> | Fetch or monitor the RSSI (dBm) and SNR (dB) values.  |
| <b>Example</b>     | AT+WRXSIG?<br>+WRXSIG:-68,31<br>OK<br><br>AT+WRXSIG=10<br>+WRXSIG:-68,31  |

|  |  |
|--|--|
|  | +WRXSIG:-68,30<br>+WRXSIG:-68,32<br>+WRXSIG:-68,32<br>+WRXSIG:-68,32<br>+WRXSIG:-68,32<br>+WRXSIG:-68,30<br>+WRXSIG:-68,31<br>+WRXSIG:-68,32<br>+WRXSIG:-68,32<br>OK |
|--|--|

## 6.5 AT+WRATECTRL

|                    |  |
|--------------------|--|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+WRATECTRL=<mode><br><u><b>GET</b></u><br>AT+WRATECTRL?                             |
| <b>Response</b>    | <u><b>SET</b></u><br>OK<br><u><b>GET</b></u><br>+WRATECTRL=<mode><br>OK                                    |
| <b>Parameters</b>  | <mode><br>0 : disable<br>1 : enable*   |
| <b>Description</b> | Toggle the MCS rate control option.  |
| <b>Example</b>     | AT+WRATECTRL?<br>+WRATECTRL:1<br>OK<br><br>AT+WRATECTRL=0<br>OK<br><br>AT+WRATECTRL?<br>+WRATECTRL:0<br>OK |

## 6.6 AT+WMCS

|                    |   |
|--------------------|---|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+WMCS=<index><br><u><b>GET</b></u><br>AT+WMCS?   |
| <b>Response</b>    | <u><b>SET</b></u><br>OK<br><u><b>GET</b></u><br>+WMCS=<index><br>OK   |
| <b>Parameters</b>  | <index><br>Modulation Coding Scheme index (0, 1, 2, 3, 4, 5, 6, 7 and 10)   |
| <b>Description</b> | Set the MCS index applied when the rate control is disabled.  |
| <b>Example</b>     | AT+WMCS?<br>ERROR<br><br>AT+WMCS=1<br>ERROR<br><br>AT+WRATECTRL?<br>+WRATECTRL:1<br>OK<br><br>AT+WRATECTRL=0<br>OK<br><br>AT+WRATECTRL?<br>+WRATECTRL:0<br>OK<br><br>AT+WMCS?<br>+WMCS:7<br>OK<br><br>AT+WMCS=1 |

|  |          |
|--|----------|
|  | OK       |
|  | AT+WMCS? |
|  | +WMCS:1  |
|  | OK       |

## 6.7 AT+WDUTYCYCLE

|                    |   |
|--------------------|---|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+WDUTYCYCLE=<window>[,<duration>[,<margin>]]<br><u><b>GET</b></u><br>AT+WDUTYCYCLE?  |
| <b>Response</b>    | <u><b>SET</b></u><br>OK<br><u><b>GET</b></u><br>+WDUTYCYCLE=<window>,<duration>,<margin><br>OK  |
| <b>Parameters</b>  | <p><b>&lt;window&gt;</b><br/>Duty cycle window in microseconds</p> <p><b>&lt;duration&gt;</b><br/>TX duration in microseconds allowed within duty cycle window</p> <p><b>&lt;margin&gt;</b><br/>Duty margin in microseconds</p> |
| <b>Description</b> | Configure duty cycle operation.   |
| <b>Example</b>     | AT+WDUTYCYCLE?<br>+WDUTYCYCLE:0,0,0<br>OK<br><br>AT+WDUTYCYCLE=1000000,100000<br><br>AT+WDUTYCYCLE?<br>+WDUTYCYCLE:1000000,100000,0<br>OK   |

|  |  |
|--|--|
|  | AT+WDUTYCYCLE=0<br>OK<br><br>AT+WDUTYCYCLE?<br>+WDUTYCYCLE:0,0,0<br>OK |
|--|--|

## 6.8 AT+WCCATHRESHOLD

|                    |  |
|--------------------|--|
| <b>Command</b>     | <u>SET</u><br>AT+WCCATHRESHOLD=<threshold><br><u>GET</u><br>AT+WCCATHRESHOLD?  |
| <b>Response</b>    | <u>SET</u><br>OK<br><u>GET</u><br>+WCCATHRESHOLD=<threshold><br>OK   |
| <b>Parameters</b>  | <threshold><br>CCA threshold.(unit: dBm) (-100 ~ -35)  |
| <b>Description</b> | Set CCA threshold.   |
| <b>Example</b>     | AT+WCCATHRESHOLD?<br>+WCCATHRESHOLD:-75<br>OK<br><br>AT+WCCATHRESHOLD=-80<br>OK<br><br>AT+WCCATHRESHOLD?<br>+WCCATHRESHOLD:-80<br>OK |

## 6.9 AT+WTXTIME

|                |   |
|----------------|---|
| <b>Command</b> | <u>SET</u><br>AT+WTXTIME=<cs_time>,<pause_time> |
|----------------|---|

|                    |  |
|--------------------|--|
|                    | <b><u>GET</u></b><br>AT+WTXTIME?   |
| <b>Response</b>    | <b><u>SET</u></b><br>OK<br><b><u>GET</u></b><br>+WTXTIME:<cs_time>,<pause_time><br>OK  |
| <b>Parameters</b>  | <b>&lt;cs_time&gt;</b><br>Carrier sensing time in microseconds (0~13260), Listen before talk<br><br><b>&lt;pause_time&gt;</b><br>Tx pause time in microseconds |
| <b>Description</b> | Set carrier sense time and pause time.   |
| <b>Example</b>     | AT+WTXTIME?<br>+WTXTIME:0,0<br>OK<br><br>AT+WTXTIME=128,2000<br>OK<br><br>AT+WTXTIME?<br>+WTXTIME:128,2000<br>OK   |

## 6.10 AT+WTSF

|                    |  |
|--------------------|--|
| <b>Command</b>     | <b><u>GET</u></b><br>AT+WTSF?                                      |
| <b>Response</b>    | <b><u>GET</u></b><br>+WTSF:<time><br>OK                            |
| <b>Parameters</b>  | <b>&lt;time&gt;</b><br>Elapsed TSF timer duration in microseconds. |
| <b>Description</b> | Read the elapsed TSF timer duration.                               |
| <b>Example</b>     | AT+WTSF?   |

```
+WTSF:44142384
OK
```

## 6.11 AT+WSCAN

|             |  |
|-------------|--|
| Command     | <p><b><u>RUN</u></b><br/>AT+WSCAN</p> <p><b><u>SET</u></b><br/>AT+WSCAN=[{+ -}]&lt;freq&gt;[,&lt;freq&gt; ...]</p> <p><b><u>GET</u></b><br/>AT+WSCAN?</p>  |
| Response    | <p><b><u>RUN</u></b><br/>+WSCAN:&lt;bssid&gt;,&lt;freq&gt;,&lt;sig_level&gt;,&lt;flags&gt;,&lt;ssid&gt;<br/>:<br/>OK</p> <p><b><u>SET</u></b><br/>OK</p> <p><b><u>GET</u></b><br/>+WSCAN:&lt;freq&gt;[,&lt;freq&gt; ...]<br/>OK</p>  |
| Parameters  | <p><b>&lt;bssid&gt;</b><br/>The BSSID of the AP.</p> <p><b>&lt;freq&gt;</b><br/>The center frequency of the channel. (MHz)</p> <p><b>&lt;sig_level&gt;</b><br/>The RSSI (Received Signal Strength Indicator) in dBm.</p> <p><b>&lt;flags&gt;</b><br/>Service set flags.</p> <p><b>&lt;ssid&gt;</b><br/>The SSID of the AP.</p> |
| Description | <p><b><u>RUN</u></b><br/>Perform Wi-Fi scanning.</p> <p><b><u>SET/GET</u></b></p>  |



|                       |  |
|-----------------------|--|
|                       | <p>Set the frequencies of the channel to scan or get a list of them.</p> <p>In the SET command, if the first frequency value has a '+' or '-' prefix, a new frequency is added or a specific frequency is excluded.</p> <p>"AT+WSCAN=0" command resets the scan frequency list to scan all supported channels.</p> <p>NOTE:</p> <p>The SET command cannot be used while connected to the AP and responds with ERROR.</p> <p>After "AT+WCOUNTRY" and "AT+WDISCONN" commands, the scan frequency list is reset to scan all supported channels.</p>   |
| <p><b>Example</b></p> | <pre> AT+WCOUNTRY="US" OK  AT+WSCAN? +WSCAN:902.5,903.0,903.5,904.5,905.0,905.5,906.0,906.5,907.0,907.5 +WSCAN:908.5,909.5,910.5,911.5,912.5,913.5,914.5,915.5,916.5,917.5 +WSCAN:918.5,919.5,920.5,921.5,922.5,923.5,924.5,909.0,911.0,913.0 +WSCAN:915.0,917.0,919.0,921.0,923.0,925.0,910.0,914.0,918.0,922.0 +WSCAN:925.5,926.5,927.5,927.0,926.0 OK AT+WSCAN +WSCAN:"02:00:eb:13:d3:4a",922.5,-39,"[ESS]","halow_open" +WSCAN:"68:27:eb:0e:07:27",922.5,-30,"[WPA2-PSK-CCMP][ESS]","halow_wpa2" +WSCAN:"8c:0f:fa:00:28:1f",906.0,-54,"[WPA3-SAE-CCMP][ESS]","halow_sae" +WSCAN:"8c:0f:fa:00:29:46",921.0,-75,"[WPA3-SAE-CCMP][ESS]","halow_sae2" OK  AT+WSCAN=922.5 OK AT+WSCAN? +WSCAN:922.5 OK AT+WSCAN +WSCAN:"02:00:eb:13:d3:4a",922.5,-39,"[ESS]","halow_open" +WSCAN:"68:27:eb:0e:07:27",922.5,-30,"[WPA2-PSK-CCMP][ESS]","halow_wpa2" </pre> |

```
OK

AT+WSCAN=+906,921
OK
AT+WSCAN?
+WSCAN:906.0,921.0,922.5
OK
AT+WSCAN
+WSCAN:"02:00:eb:13:d3:4a",922.5,-39,"[ESS]","halow_open"
+WSCAN:"68:27:eb:0e:07:27",922.5,-30,"[WPA2-PSK-CCMP][ESS]","halow_wpa2"
+WSCAN:"8c:0f:fa:00:28:1f",906.0,-54,"[WPA3-SAE-CCMP][ESS]","halow_sae"
+WSCAN:"8c:0f:fa:00:29:46",921.0,-75,"[WPA3-SAE-CCMP][ESS]","halow_sae2"
OK

AT+WSCAN=-921,922.5
OK
AT+WSCAN?
+WSCAN:906.0
OK
AT+WSCAN
+WSCAN:"8c:0f:fa:00:28:1f",906.0,-54,"[WPA3-SAE-CCMP][ESS]","halow_sae"
OK

AT+WSCAN=0
OK
AT+WSCAN?
+WSCAN:902.5,903.0,903.5,904.5,905.0,905.5,906.0,906.5,907.0,907.5
+WSCAN:908.5,909.5,910.5,911.5,912.5,913.5,914.5,915.5,916.5,917.5
+WSCAN:918.5,919.5,920.5,921.5,922.5,923.5,924.5,909.0,911.0,913.0
+WSCAN:915.0,917.0,919.0,921.0,923.0,925.0,910.0,914.0,918.0,922.0
+WSCAN:925.5,926.5,927.5,927.0,926.0
OK

AT+WSCAN=922.5
OK
AT+WSCAN
+WSCAN:"02:00:eb:13:d3:4a",922.5,-39,"[ESS]","halow_open"
```

|  |  |
|--|--|
|  | +WSCAN:"68:27:eb:0e:07:27",922.5,-30,"[WPA2-PSK-CCMP][ESS]","halow_wpa2"<br>OK<br>AT+WCONN="halow_open"<br>OK<br>AT+WSCAN?<br>+WSCAN=922.5<br>OK<br>AT+WSCAN=+906,921<br>ERROR<br><br>AT+WDISCONN<br>OK<br>AT+WSCAN?<br>+WSCAN:902.5,903.0,903.5,904.5,905.0,905.5,906.0,906.5,907.0,907.5<br>+WSCAN:908.5,909.5,910.5,911.5,912.5,913.5,914.5,915.5,916.5,917.5<br>+WSCAN:918.5,919.5,920.5,921.5,922.5,923.5,924.5,909.0,911.0,913.0<br>+WSCAN:915.0,917.0,919.0,921.0,923.0,925.0,910.0,914.0,918.0,922.0<br>+WSCAN:925.5,926.5,927.5,927.0,926.0<br>OK<br><br>AT+WCCOUNTRY="EU"<br>OK<br>AT+WSCAN?<br>+WSCAN:863.5,864.5,865.5,866.5,867.5,864.0,866.0<br>OK |
|--|--|

## 6.12 AT+WCONN

|                 |   |
|-----------------|---|
| <b>Command</b>  | <u><b>SET</b></u><br>AT+WCONN="<ssid bssid>","<security>","<password>"]<br><u><b>GET</b></u><br>AT+WCONN?           |
| <b>Response</b> | <u><b>SET</b></u><br>OK<br><u><b>GET</b></u><br>+WCONN="<ssid>","<bssid>","<security>","<password>","<state>"<br>OK |

|             |  |
|-------------|--|
| Parameters  | <p><b>&lt;ssid&gt;</b><br/>The SSID of the AP.</p> <p><b>&lt;bssid&gt;</b><br/>The BSSID of the AP.</p> <p><b>&lt;security&gt;</b><br/>open*, wpa2-psk (or psk), wpa3-owe (or owe), wpa3-sae (or sae)</p> <p><b>&lt;password&gt; (wpa2/wpa3-sae security option only)</b><br/>The password when wpa2/wpa3-sae security option is used. (length : 8 ~ 63)</p> <p><b>&lt;state&gt;</b><br/>State indicator: "connecting", "connected", "disconnecting" or "disconnected"</p> |
| Description | <p>Connect to a new AP or retrieves information about the current AP.</p> <p>NOTE:<br/>If an "ERROR" is returned with the error number INPROGRESS(2) or TIMEOUT(4), the AT-STA needs to be disconnected from the AP with the "AT+WDISCONN" command before a connection is attempted again with "AT+WCONN".</p>   |
| Example     | <p><b>OPEN :</b><br/>AT+WCONN="halow_ap"<br/>OK<br/>AT+WCONN?<br/>+WCONN:"halow_ap","8C:0F:FA:00:2B:A1","open","","connected"<br/>OK</p> <p><b>WPA2-PSK :</b><br/>AT+WCONN="halow_ap","wpa2-psk","12345678"<br/>OK<br/>AT+WCONN?<br/>+WCONN:"halow_ap","8C:0F:FA:00:2B:A1","wpa2-psk","12345678","connected"<br/>OK</p> <p><b>WPA3-OWE :</b><br/>AT+WCONN="halow_ap","wpa3-owe"<br/>OK<br/>AT+WCONN?</p>   |

|  |   |
|--|---|
|  | +WCONN:"halow_ap","8C:0F:FA:00:2B:A1","wpa3-owe","","connected"<br>OK<br><br><b>WPA3-SAE :</b><br>AT+WCONN="halow_ap","wpa3-sae","12345678"<br>OK<br>AT+WCONN?<br>+WCONN:"halow_ap","8C:0F:FA:00:2B:A1","wpa3-sae","12345678","connected"<br>OK |
|--|---|

### 6.13 AT+WDISCONN

|             |   |
|-------------|---|
| Command     | <u><b>RUN</b></u><br>AT+WDISCONN                                |
| Response    | <u><b>RUN</b></u><br>OK   |
| Description | Disconnect from the AP or abort an on-going connection process. |
| Example     | AT+WDISCONN<br>OK   |

### 6.14 AT+WSOFTAP

|            |   |
|------------|---|
| Command    | <u><b>SET</b></u><br>AT+WSOFTAP=<frequency>[@<bandwidth>], "<ssid>" [, "<security>" [, "<password>"]]<br><u><b>GET</b></u><br>AT+WSOFTAP? |
| Response   | <u><b>SET</b></u><br>OK<br><u><b>GET</b></u><br>+WSOFTAP=<frequency>,"<ssid>","<security>","<password>" [, "dhcp"]<br>OK                  |
| Parameters | <b>&lt;frequency&gt;</b><br>S1G channel frequency (MHz)<br><br><b>&lt;bandwidth&gt;</b><br>S1G channel bandwidth (1/2/4 MHz)              |

|                    |  |
|--------------------|--|
|                    | <p><b>&lt;ssid&gt;</b><br/>The SSID of the AP.</p> <p><b>&lt;security&gt;</b><br/>open*, wpa2-psk (or psk)</p> <p><b>&lt;password&gt; (wpa2 security option only)</b><br/>The password when wpa2 security option is used. (length : 8 ~ 63)</p> <p><b>&lt;dhcp&gt;</b><br/>Only included when the DHCP server is running.</p>  |
| <b>Description</b> | <p>Run as the AP mode or retrieves information about the current settings.</p> <p>NOTE:<br/>The system should be reset to exit the AP mode.<br/>Software Reset is possible with the ATZ command.</p>   |
| <b>Example</b>     | <pre>AT+WCCOUNTRY="JP" OK  AT+WSCAN? +WSCAN:923.5,924.5,925.5,926.5,921.0,923.0,924.0,925.0,926.0,927.0 +WSCAN:924.5,925.5 OK  AT+WSOFTAP=925.5@4,"halow_softap_psk","psk","12345678" OK  AT+WSOFTAP? +WSOFTAP:4,925.5,"halow_softap_psk","wpa2-psk","12345678" OK  AT+WDHCPS +WDHCPS:192.168.200.27,255.255.255.0,192.168.200.1 OK  AT+WSOFTAP? +WSOFTAP:4,925.5,"halow_softap_psk","wpa2-psk","12345678","dhcp" OK</pre> |

## 6.15 AT+WBSSMAXIDLE

|                    |  |
|--------------------|--|
| <b>Command</b>     | <u>SET</u><br>AT+WBSSMAXIDLE=<period>[,<retry>]<br><u>GET</u><br>AT+WBSSMAXIDLE?   |
| <b>Response</b>    | <u>SET</u><br>OK<br><u>GET</u><br>+WBSSMAXIDLE:<period>,<retry><br>OK  |
| <b>Parameters</b>  | <b>&lt;period&gt;</b><br>BSS MAX IDLE period in seconds (default: 0)<br><br><b>&lt;retry&gt;</b><br>retry count for receiving keep alive packet from STA (3 ~ 100, default: 3)   |
| <b>Description</b> | <p>Configure the BSS MAX IDLE service for SoftAP.</p> <p>SoftAP disconnects STA that is inactive for BSS MAX IDLE time. If the AP does not receive a keep alive packet from the STA for BSS MAX IDLE time, it is determined that the STA is in an inactive state.</p> <p>Example:</p> <ul style="list-style-type: none"> <li>- BSS max idle period = 60 secs</li> <li>- retry count = 5</li> <li>- BSS max idle time = 60 x 5 = 300 secs</li> </ul> <p>If the period is set 0, the service is stopped.</p> |
| <b>Example</b>     | AT+WBSSMAXIDLE?<br>+WBSSMAXIDLE:0,3<br>OK<br>AT+WBSSMAXIDLE=60,60<br>OK<br>AT+WBSSMAXIDLE?<br>+WBSSMAXIDLE:60,60<br>OK   |

|  |  |
|--|--|
|  | AT+WSOFTAP=918.5,"halow_softap_wpa2","wpa2-psk","12345678"<br>OK<br>AT+WDHCPS<br>+WDHCPS:"192.168.50.1","255.255.255.0","192.168.50.1"<br>OK<br><br>AT+WBSSMAXIDLE=60,5<br>OK<br>AT+WBSSMAXIDLE?<br>+WBSSMAXIDLE:60,5<br>OK<br><br>AT+WBSSMAXIDLE=0<br>OK<br>AT+WBSSMAXIDLE?<br>+WBSSMAXIDLE:0,3<br>OK |
|--|--|

## 6.16 AT+WSTAINFO

|                   |   |
|-------------------|---|
| <b>Command</b>    | <u>SET</u><br>AT+WSTAINFO=<aid><br><u>GET</u><br>AT+WSTAINFO?   |
| <b>Response</b>   | +WSTAINFO=<aid>,"<mac_address>",<rssi>,<snr>,<mcs_index><br>OK  |
| <b>Parameters</b> | <aid><br>Association ID<br><br><mac_address><br>Hardware address of associated station<br><br><rssi><br>Received Signal Strength indication<br><br><snr><br>Signal to Noise Ratio |



|                    |  |
|--------------------|--|
|                    | <b>&lt;mcs_index&gt;</b><br>Modulation Coding Scheme index   |
| <b>Description</b> | Get information of associated STAs <u>when the device is in AP mode.</u>   |
| <b>Example</b>     | <pre> AT+WSOFTAP=918.5,"halow_softap","wpa2-psk","12345678" OK AT+WIPADDR="192.168.1.1","255.255.255.0","192.168.1.1" OK AT+WDHCPS +WDHCPS:"192.168.1.1","255.255.255.0","192.168.1.1" OK  Wait for one or more stations to be associated ...  AT+WSTAINFO? +WSTAINFO:1,"8c:0f:fa:00:2b:a1",-34,31,7 +WSTAINFO:2,"8c:0f:fa:00:2b:a2",-45,34,7 +WSTAINFO:3,"8c:0f:fa:00:2b:a3",-16,21,7 OK AT+WSTAINFO=1 +WSTAINFO:1,"8c:0f:fa:00:2b:a1",-33,34,7 OK </pre> |

## 6.17 AT+WIPADDR

|                    |  |
|--------------------|--|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+WIPADDR=" <b>&lt;address&gt;</b> "," <b>&lt;netmask&gt;</b> "," <b>&lt;gateway&gt;</b> "<br><u><b>GET</b></u><br>AT+WIPADDR? |
| <b>Response</b>    | <u><b>SET</b></u><br>OK<br><u><b>GET</b></u><br>+WIPADDR=" <b>&lt;address&gt;</b> "," <b>&lt;netmask&gt;</b> "," <b>&lt;gateway&gt;</b> "<br>OK      |
| <b>Parameters</b>  | <b>&lt;address&gt;</b> , <b>&lt;netmask&gt;</b> , <b>&lt;gateway&gt;</b><br>IPv4 address   |
| <b>Description</b> | Configure the IPv4 address.  |
| <b>Example</b>     | AT+WIPADDR="192.168.200.20","255.255.255.0","192.168.200.1"  |

|  |  |
|--|--|
|  | OK<br>AT+WIPADDR?<br>+WIPADDR="192.168.200.20","255.255.255.0","192.168.200.1"<br>OK |
|--|--|

## 6.18 AT+WDNS

|                    |   |
|--------------------|---|
| <b>Command</b>     | <u>SET</u><br>AT+WDNS="<DNS1>"[, "<DNS2>"]<br><u>GET</u><br>AT+WDNS?  |
| <b>Response</b>    | <u>SET</u><br>OK<br><u>GET</u><br>+WDNS="<DNS1>","<DNS2>"<br>OK   |
| <b>Parameters</b>  | <DNS1>,<DNS2><br>IPv4 address   |
| <b>Description</b> | Configure the IP address of the DNS server.   |
| <b>Example</b>     | AT+WDNS?<br>+WDNS="192.168.200.1","0.0.0.0"<br>OK<br><br>AT+WDNS="8.8.8.8"<br>OK<br>AT+WDNS?<br>+WDNS="8.8.8.8","0.0.0.0"<br>OK<br><br>AT+WDNS="8.8.8.8","8.8.4.4"<br>OK<br>AT+WDNS?<br>+WDNS="8.8.8.8","8.8.4.4"<br>OK |

## 6.19 AT+WDHCP

|                    |   |
|--------------------|---|
| <b>Command</b>     | <b><u>RUN</u></b><br>AT+WDHCP<br><b><u>SET</u></b><br>AT+WDHCP=<mode><br><b><u>GET</u></b><br>AT+WDHCP?   |
| <b>Response</b>    | <b><u>RUN</u></b><br>+WDHCP:"<address>","<netmask>","<gateway>"<br>OK<br><b><u>SET</u></b><br>OK<br><b><u>GET</u></b><br>+WDHCP:{0 1}<br>OK   |
| <b>Parameters</b>  | <b>&lt;address&gt;, &lt;netmask&gt; and &lt;gateway&gt;</b><br>IPv4 Address<br><br><b>&lt;mode&gt;</b><br>0 : run manually after connection<br>1 : run automatically connection or reconnection |
| <b>Description</b> | Request dynamic IP allocation from the DHCP server.<br><br>NOTE:<br>Wi-Fi connection must be established before using this command.   |
| <b>Example</b>     | AT+WCONN="halow_ap","wpa3-sae","12345678"<br>OK<br>AT+WDHCP<br>+WDHCP:"192.168.200.20","255.255.255.0","192.168.200.1"<br>OK<br>AT+WDISCONN<br>OK<br>AT+WDHCP?<br>+WDHCP:0<br>OK                |

|  |  |
|--|--|
|  | AT+WDHCP=1<br>OK<br>AT+WCONN="halow_ap","wpa3-sae","12345678"<br>OK<br>+WEVENT:"DHCP_RUN"<br>+WEVENT:"DHCP_SUCCESS","192.168.200.18","255.255.255.0","192.168.200.1"<br>+WEVENT:"DISCONNECT","", "halow_ap","wpa3-sae"<br>+WEVENT:"CONNECT_SUCCESS","", "halow_ap","wpa3-sae"<br>+WEVENT:"DHCP_RUN"<br>+WEVENT:"DHCP_SUCCESS","192.168.200.18","255.255.255.0","192.168.200.1" |
|--|--|

## 6.20 AT+WDHCPS

|                    |   |
|--------------------|---|
| <b>Command</b>     | <u><b>RUN</b></u><br>AT+WDHCPS  |
| <b>Response</b>    | <u><b>RUN</b></u><br>+WDHCPS:"<IP>","netmask>","<gateway>"<br>OK  |
| <b>Parameters</b>  | <IP>, <netmask> and <gateway><br>'A.B.C.D' where A, B, C and D are between 0 and 255, inclusive.  |
| <b>Description</b> | Run the DHCP sever in SoftAP mode.<br><br>NOTE:<br>SoftAP must be established before using this command.<br>Refer to chapter 6.15. (AT+WSOFTAP) |
| <b>Example</b>     | AT+WDHCPS<br>+WDHCPS:"192.168.50.1","255.255.255.0","192.168.50.1"<br>OK  |

## 6.21 AT+WPING

|                 |   |
|-----------------|---|
| <b>Command</b>  | <u><b>SET</b></u><br>AT+WPING="<remote address>"[,<time>]<br><u><b>GET</b></u><br>AT+WPING? |
| <b>Response</b> | <u><b>SET</b></u>   |

|             |  |
|-------------|--|
|             | +WPING:<size>,"<remote address>",<sequence number>,<TTL>,<elapsed time><br>:<br>+WPING:<size>,"<remote address>",<sequence number>,<TTL>,<elapsed time><br>OK<br><b>GET</b><br>+WPING:"<remote address>",<time>  |
| Parameters  | <b>&lt;remote address&gt;</b><br>The remote IPv4 address of the recipient.<br><br><b>&lt;time&gt;</b><br>Monitoring duration in seconds. (Default: 5)<br><br><b>&lt;sequence number&gt;</b><br>ICMP sequence number.<br><br><b>&lt;TTL&gt;</b><br>Time to leave (TTL).<br><br><b>&lt;elapsed time&gt;</b><br>Time since the start of the session in seconds. |
| Description | Send ICMP ECHO_REQUEST to network hosts with IPv4 address. <ul style="list-style-type: none"> <li>- Interval Time : 1 sec</li> <li>- Packet Size : 64-bytes</li> </ul>   |
| Example     | AT+WPING ="192.168.200.1",10<br>+WPING:64,"192.168.200.1",1,64,4<br>+WPING:64,"192.168.200.1",2,64,4<br>:<br>+WPING:64,"192.168.200.1",9,64,4<br>+WPING:64,"192.168.200.1",10,64,4<br>OK   |

## 6.22 AT+WDEEPSLEEP

|          |  |
|----------|--|
| Command  | <b>SET</b><br>AT+WSLEEP=<timeout>[,<gpio>] |
| Response | <b>SET</b><br>OK                           |

|             |  |
|-------------|--|
| Parameters  | <p><b>&lt;timeout&gt;</b><br/>Time in milliseconds.<br/>0 for TIM mode.</p> <p><b>&lt;gpio&gt;</b><br/>GPIO number to use as external signal input.<br/>Available GPIO numbers are between 8 and 17.</p>   |
| Description | <p>Configure deep-sleep mode to save power.</p> <p>Deep sleep mode powers off most peripherals to use minimal power. The RTC and retention RAM are always powered. The CPU is powered only in TIM mode to run the uCode stored in the retention RAM. And the GPIO may be powered for external signal input.</p> <p>In TIM mode, the NRC7292 wakes up when there are frames to receive. However, in Non-TIM mode, it cannot be woken up until a timeout.</p> <p>If there are frames to send, the NRC7292 can only be woken up via the GPIO input. The GPIO input level should be low in active mode. If it is high in deep sleep mode, the NRC7292 wakes up.</p> <p>After waking up, the CPU resets and the firmware reboots. When the firmware reboot is finished, the host application or terminal program will receive a "DEEPSLEEP_WAKEUP" event message.</p> |
| Example     | <p><b>&lt; Deep Sleep, TIM mode &gt;</b><br/> AT+WCONN="halow_ap","wpa2-psk","12345678"<br/> OK<br/> AT+WDHCP<br/> +WDHCP:"192.168.200.18","255.255.255.0","192.168.200.1"<br/> OK<br/> AT+WDEEPSLEEP=0,11<br/> OK<br/> <br/> +WEVENT:"DEEPSLEEP_WAKEUP"<br/> <br/> AT+WCONN="halow_ap","wpa2-psk","12345678"<br/> OK</p>  |

|  |  |
|--|--|
|  | <p>AT+WDHCP<br/> +WDHCP:"192.168.200.18","255.255.255.0","192.168.200.1"<br/> OK<br/> AT+WPING="192.168.200.1",2<br/> +WEVENT:"PING",64,"192.168.200.1",1,64,5<br/> +WEVENT:"PING",64,"192.168.200.1",2,64,4<br/> OK</p> <p><b>&lt; Deep Sleep, Non-TIM mode &gt;</b><br/> AT+WCONN="halow_ap","wpa2-psk","12345678"<br/> OK<br/> AT+WDHCP<br/> +WDHCP:"192.168.200.18","255.255.255.0","192.168.200.1"<br/> OK<br/> AT+WDEEPSLEEP=5000,11<br/> OK</p> <p>+WEVENT:"DEEPSLEEP_WAKEUP"</p> <p>AT+WCONN="halow_ap","wpa2-psk","12345678"<br/> OK<br/> AT+WDHCP<br/> +WDHCP:"192.168.200.18","255.255.255.0","192.168.200.1"<br/> OK<br/> AT+WPING="192.168.200.1",2<br/> +WEVENT:"PING",64,"192.168.200.1",1,64,6<br/> +WEVENT:"PING",64,"192.168.200.1",2,64,4<br/> OK</p> |
|--|--|

## 6.23 AT+WFOTA

|                |  |
|----------------|--|
| <b>Command</b> | <p><b><u>SET</u></b><br/> AT+WFOTA=&lt;check_time&gt;[,\"&lt;server_url&gt;\"[,\"&lt;bin_name&gt;\",&lt;bin_crc32&gt;]]</p> <p><b><u>GET</u></b><br/> AT+WFOTA?</p> <p><b><u>RUN</u></b><br/> AT+WFOTA</p> |
|----------------|--|

|             |   |
|-------------|---|
| Response    | <p><b>SET</b><br/>OK</p> <p><b>GET</b><br/>+WFOTA:&lt;check_time&gt;,"&lt;server_url&gt;","&lt;bin_name&gt;",&lt;bin_crc32&gt;<br/>OK</p> <p><b>RUN</b><br/>OK</p>  |
| Parameters  | <p><b>&lt;check_time&gt;</b><br/>Interval time in seconds to get new firmware information from the server.<br/>Set to 0 to stop the getting or get manually.<br/>Set to -1 to disable FOTA operation.</p> <p><b>&lt;server_url&gt;</b><br/>HTTP or HTTPS Server URL</p> <p><b>&lt;bin_name&gt;</b><br/>Firmware binary name with extension .bin.</p> <p><b>&lt;bin_crc32&gt;</b><br/>32bit CRC value to detect data corruption of downloaded firmware.<br/>A hexadecimal number prefixed with 0x.</p>   |
| Description | <p>FOTA(Firmware Over-the-Air) is enabled with the SET command and disabled by AT+WFOTA=-1 command.</p> <p>When FOTA is enabled, the current firmware starts checking for new firmware on the server. The server check interval can be controlled through the &lt;check_time&gt; parameter.</p> <p>To check for new firmware, the current firmware downloads the fota.json file from the server. The server should have a fota.json file as well as firmware binary. The contents of the fota.json file are as follows.</p> <pre> 1 { 2   "AT_SDK_VER" : "10.10.10", 3   "AT_CMD_VER" : "10.10.10", 4 5   "AT_HSPI_BIN" : "nrc7292_standalone_xip_ATCMD_HSPI.bin", 6   "AT_HSPI_CRC" : "750243d8", 7 8   "AT_UART_BIN" : "nrc7292_standalone_xip_ATCMD_UART.bin", 9   "AT_UART_CRC" : "793066ec", 10 11  "AT_UART_HFC_BIN" : "nrc7292_standalone_xip_ATCMD_UART_HFC.bin", 12  "AT_UART_HFC_CRC" : "8f564369" 13 }</pre> <p>If new firmware has a higher version, the current firmware sends a FOTA_VERSION event to the terminal or host.</p> <p>+WEVENT:"FOTA_VERSION","&lt;sdk_version&gt;","&lt;atcmd_version&gt;"</p> |



After receiving the FOTA\_VERSION event, the terminal or host can use the RUN command to download new firmware from the server.

If there is no fota.json file on the server, the firmware information to be downloaded can be set with the bin\_name and bin\_crc32 parameters. And the terminal or host can use the RUN command without receiving the FOTA\_VERSION event.

The terminal or host can check the download process through FOTA\_BINARY and FOTA\_DOWNLOAD events from the current firmware.

+WEVENT: "FOTA\_BINARY", "<binary\_name>"

+WEVENT: "FOTA\_DOWNLOAD", "<total\_size>,<download\_size>"

When the download is complete and ready to update, the terminal or host will receive a FOTA\_UPDATE event from the current firmware.

+WEVENT: "FOTA\_UPDATE"

If an error occurs during the above process, the terminal or host will receive a FOTA\_FAIL event from the current firmware.

+WEVENT: "FOTA\_FAIL"

And FOTA will be automatically disabled.

If there are no errors, the current firmware will be replaced with the new firmware after a software reset. A software reset is possible with the ATZ command.

Firmware replacement will take about 10 seconds or more.

If an error occurs while accessing the flash memory for firmware replacement, the current firmware cannot be restored. If the error still occurs after a hardware reset, the firmware can only be restored through the download tool.

#### EVENT:

| Name          | Description  |
|---------------|--|
| FOTA_VERSION  | The version of new firmware on the server.<br>- User SDK version<br>- AT Command Set version           |
| FOTA_BINARY   | The binary name of new firmware to download from the server.   |
| FOTA_DOWNLOAD | The binary size of new firmware being downloaded from the server.<br>- Total size<br>- Downloaded size |
| FOTA_UPDATE   | The current firmware is ready to be replaced with the new firmware.                                    |
| FOTA_FAIL     | An error occurred during the FOTA process.   |

#### TEST:

The AT+WFOTA command can be tested using the python-http-server package in the SDK.

Path : atcmd/host/python-http-server

This package has the shell and python scripts to run HTTP/HTTPS server.

```
python-http-server/
├── fota.json
├── nrc7292_standalone_xip_ATCMD_HSPI.bin
├── nrc7292_standalone_xip_ATCMD_UART.bin
├── nrc7292_standalone_xip_ATCMD_UART_HFC.bin
├── python
│   ├── crc.py
│   └── https-server.py
├── Run-server.sh
├── ssl-cert
│   ├── server.crt
│   ├── server.csr
│   ├── server.key
│   └── server.key.origin
└── Update-fota-info.sh
```

| Shell Script        | Description   |
|---------------------|---|
| Run-sever.sh        | Run HTTP or HTTPS server.<br>Usage:<br>\$ ./Run-server.sh http<br>\$ ./Run-server.sh https  |
| Update-fota-info.sh | Calculate the CRC value of firmware binaries and update the fota.json file.<br>Usage:<br>\$ ./Update-fota-info.sh [options]<br><br>Firmware version and binary name can be set by editing this file.<br><pre>6 SDK_VER="10.10.10" 7 CMD_VER="10.10.10" 8 9 HSPI_BIN="nrc7292_standalone_xip_ATCMD_HSPI.bin" 10 UART_BIN="nrc7292_standalone_xip_ATCMD_UART.bin" 11 UART_HFC_BIN="nrc7292_standalone_xip_ATCMD_UART_HFC.bin"</pre> <p>Alternatively, it can be set as options when executing the script. Available options can be checked with the -h or --help option. Values set as options overwrite values set in the file.</p> <p>If a binary is replaced with a new one, the fota.json should be updated by Update-fota-info.sh.</p> |

#### Example

AT+WFOTA?  
+WFOTA:0,"","",0x0

OK

**< Get new firmware information from fota.json file >**

AT+WFOTA=10,"https://192.168.200.1:4443"

AT+WFOTA=10,"https://192.168.200.1:4443"

OK

AT+WFOTA?

+WFOTA:10,"https://192.168.200.1:4443","",0x0

OK

+WEVENT:"FOTA\_VERSION","10.10.10","10.10.10"

+WEVENT:"FOTA\_VERSION","10.10.10","10.10.10"

+WEVENT:"FOTA\_VERSION","10.10.10","10.10.10"

\*Stop the getting to switch manually.

AT+WFOTA=0

OK

AT+WFOTA=0

OK

+WEVENT:"FOTA\_VERSION","10.10.10","10.10.10"

**< Set new firmware information without fota.json file >**

AT+WFOTA=0,"https://192.168.200.1:4443","nrc7292\_atcmd\_hspi.bin",0x3e47cf92

OK

AT+WFOTA?

+WEVENT:0,"https://192.168.200.1:4443","nrc7292\_atcmd\_hspi.bin",0x3E47CF92

OK

**< Download the firmware binary >**

AT+WFOTA

OK

+WEVENT:"FOTA\_BINARY","nrc7292\_atcmd\_hspi.bin"

+WEVENT:"FOTA\_DOWNLOAD",897632,90112

+WEVENT:"FOTA\_DOWNLOAD",897632,180224

+WEVENT:"FOTA\_DOWNLOAD",897632,270336

:

+WEVENT:"FOTA\_DOWNLOAD",897632,720896

+WEVENT:"FOTA\_DOWNLOAD",897632,811008

|  |   |
|--|---|
|  | +WEVENT:"FOTA_DOWNLOAD",897632,897632<br>+WEVENT:"FOTA_UPDATE"<br><br>< Reset and update ><br>ATZ |
|--|---|

## 6.24 AT+WTIMEOUT

|                    |   |   |
|--------------------|---|---|
| <b>Command</b>     | <u>SET</u><br>AT+WTIMEOUT="<command>",<timeout><br><u>GET</u><br>AT+WTIMEOUT?   |   |
| <b>Response</b>    | <u>SET</u><br>OK<br><u>GET</u><br>+WTIMEOUT:"<command>",<timeout><br>...<br>OK  |   |
| <b>Parameters</b>  | <command><br>"WCONN", "WDISCONN", "WDHCP"<br><br><timeout><br>Timeout in seconds. (0: no timeout)   |   |
| <b>Description</b> | Configure the response timeout for the specified command.<br><br>Default timeout : <ul style="list-style-type: none"> <li>- WCONN : 60 secs</li> <li>- WDISCONN : 60 secs</li> <li>- WDHCP : 60 secs</li> </ul> |   |
| <b>Example</b>     | AT+WTIMEOUT?<br>+WTIMEOUT:"WCONN",60<br>+WTIMEOUT:"WDISCONN",60<br>+WTIMEOUT:"WDHCP",60<br>OK   | AT+WTIMEOUT="WCONN",120<br>OK<br>AT+WTIMEOUT?<br>+WTIMEOUT:"WCONN",120<br>+WTIMEOUT:"WDISCONN",60<br>+WTIMEOUT:"WDHCP",60<br>OK |

## 6.25 +WEVENT

| Response    | +WEVENT:<event>   |
|-------------|---|
| Parameters  | <p><b>&lt;event&gt;</b></p> <p>"CONNECT_SUCCESS", "&lt;bssid&gt;", "&lt;ssid&gt;", "&lt;security&gt;"</p> <p>"DISCONNECT", "&lt;bssid&gt;", "&lt;ssid&gt;", "&lt;security&gt;"</p> <p>"DHCP_START"</p> <p>"DHCP_STOP"</p> <p>"DHCP_BUSY"</p> <p>"DHCP_FAIL"</p> <p>"DHCP_SUCCESS", "&lt;address&gt;", "&lt;netmask&gt;", "&lt;gateway&gt;"</p> <p>"DHCP_TIMEOUT", &lt;time&gt;</p> <p>"STA_CONNECT", "&lt;mac_addr&gt;"</p> <p>"STA_DISCONNECT", "&lt;mac_addr&gt;"</p> <p>"FOTA_VERSION", "&lt;sdk_version&gt;", "&lt;atcmd_version&gt;"</p> <p>"FOTA_BINARY", "&lt;binary_name&gt;"</p> <p>"FOTA_DOWNLOAD", "total_size", "download_size"</p> <p>"FOTA_UPDATE"</p> <p>"FOTA_FAIL"</p> <p>"DEEPSLEEP_WAKEUP"</p> |
| Description | Asynchronously raised Wi-Fi event logs.   |
| Example     | <pre>+WEVENT:"CONNECT_SUCCESS","8c:0f:fa:00:2b:a1","halow_sae","wpa3-sae" +WEVENT:"DISCONNECT","8c:0f:fa:00:2b:a1","halow_sae","wpa3-sae"  +WEVENT:"DHCP_START" +WEVENT:"DHCP_STOP" +WEVENT:"DHCP_BUSY" +WEVENT:"DHCP_FAIL" +WEVENT:"DHCP_SUCCESS","192.168.200.18","255.255.255.0","192.168.200.1" +WEVENT:"DHCP_TIMEOUT",60</pre>   |

|  |   |
|--|---|
|  | <pre>+WEVENT:"STA_CONNECT","8C:0F:FA:00:39:0D"<br/>+WEVENT:"STA_DISCONNECT","8C:0F:FA:00:39:0D"<br/><br/>+WEVENT:"FOTA_VERSION","10.10.10","10.10.10"<br/>+WEVENT:"FOTA_BINARY","nrc7292_atcmd_hspi.bin"<br/>+WEVENT:"FOTA_DOWNLOAD",897632,90112<br/>+WEVENT:"FOTA_UPDATE"<br/>+WEVENT:"FOTA_FAIL"<br/><br/>+WEVENT:"DEEPSLEEP_WAKEUP"</pre> |
|--|---|

## 7 Socket AT Commands

| Commands       | Description  |
|----------------|--|
| AT+SOPEN       | Create a TCP/UDP socket for IPv4 domain.                         |
| AT+SCLOSE      | Close an existing socket.  |
| AT+SLIST       | List all currently open sockets.                                 |
| AT+SSEND       | Send data through a socket.                                      |
| AT+SRECV       | Read buffered data from the network stack (lwip).                |
| AT+SRECVMODE   | Configures how data is read from the network stack (lwip).       |
| AT+SRECVINFO   | Configure the information level of “+RXD” message.               |
| AT+SADDRINFO   | Check the IP address from the domain name.                       |
| AT+STCPNODELAY | Enable or disable TCP Nagle’s algorithm.                         |
| AT+STIMEOUT    | Configure the response timeout for the specified socket command. |
| +SEVENT        | Asynchronously raised socket event logs.                         |
| +RXD           | An event log for a received packet with payload.                 |

## 7.1 AT+SOPEN

|                    |   |
|--------------------|---|
| <b>Command</b>     | <b><u>SET</u></b><br>AT+SOPEN="udp",<local_port>[,<reuse_addr>]<br>AT+SOPEN="tcp",<local_port>[,<reuse_addr>]<br>AT+SOPEN="tcp",<server address>,<server port>[,<reuse_addr>]   |
| <b>Response</b>    | <b><u>SET</u></b><br>+SOPEN=<socket ID><br>OK   |
| <b>Parameters</b>  | <b>&lt;local_port&gt; (UDP)</b><br>The outgoing local port.<br><br><b>&lt;local_port&gt; (TCP Server)</b><br>Local port to listen on.<br><br><b>&lt;server address&gt;,&lt;server port&gt; (TCP Client)</b><br>The IPv4 address and port number of the TCP server.<br><br><b>&lt;reuse_addr&gt;</b><br>SO_REUSEADDR option (0:disable, 1:enable)<br><br><b>&lt;socket ID&gt;</b><br>The ID allocated to the socket. |
| <b>Description</b> | Create a TCP/UDP socket for IPv4 domain.<br><br>A socket for TCP server will listen on the given port in the background and asynchronously raise the event CONNECT to notify incoming connections.  |
| <b>Example</b>     | AT+SOPEN="UDP",60000<br>+SOPEN=0<br>OK<br><br>AT+SOPEN="TCP",50000<br>+SOPEN=1<br>OK<br>+SEVENT: "CONNECT",2<br><br>AT+SOPEN="TCP","192.168.200.100",5001<br>+SOPEN=3   |



|  |    |
|--|----|
|  | OK |
|--|----|

## 7.2 AT+SCLOSE

|                    |   |
|--------------------|---|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+SCLOSE=<socket ID><br><u><b>RUN</b></u><br>AT+SCLOSE  |
| <b>Response</b>    | <u><b>SET</b></u><br>+SCLOSE:<socket ID><br>OK<br><u><b>RUN</b></u><br>+SCLOSE:<socket ID><br>:<br>+SCLOSE:<socket ID><br>OK  |
| <b>Parameters</b>  | <b>&lt;socket ID&gt;</b><br>The ID allocated to the socket.   |
| <b>Description</b> | Close an existing socket. To close all existing sockets, run a command without the parameter <socket ID>. If a server socket is closed, all client sockets connected to the server socket will close automatically. |
| <b>Example</b>     | AT+SCLOSE=1<br>+SCLOSE:1<br>OK<br><br>AT+SCLOSE<br>+SCLOSE:0<br>+SCLOSE:2<br>+SCLOSE:3<br>OK  |

## 7.3 AT+SLIST

|                 |   |
|-----------------|---|
| <b>Command</b>  | <u><b>GET</b></u><br>AT+SLIST?  |
| <b>Response</b> | <u><b>GET</b></u><br>+SLIST:<socket ID>,"<protocol>","<remote address>",<remote port>,<local port><br>: |

|                    |  |
|--------------------|--|
|                    | +SLIST:<socket ID>,"<protocol>","<remote address>",<remote port>,<local port><br>OK  |
| <b>Parameters</b>  | <p><b>&lt;socket ID&gt;</b><br/>The ID allocated to the socket.</p> <p><b>&lt;protocol&gt;</b><br/>TCP or UDP</p> <p><b>&lt;remote address&gt;,&lt;remote port&gt;,&lt;local port&gt;</b><br/>The remote address, remote port and local port associated with the socket.</p> |
| <b>Description</b> | List all currently open sockets.   |
| <b>Example</b>     | AT+SLIST?<br>+SLIST:0,"UDP","0.0.0.0",0,60000<br>+SLIST:1,"TCP","0.0.0.0",0,50000<br>+SLIST:2,"TCP","192.168.200.100",55354,0<br>+SLIST:3,"TCP","192.168.200.100",5001,52433<br>OK   |

## 7.4 AT+SSEND

|                   |  |
|-------------------|--|
| <b>Command</b>    | <p><b>SET</b></p> AT+SSEND =<ID>[,<length>[,<done_event>]]<br>AT+SSEND =<ID>,"<remote host>",<remote port>[,<length>[,<done_event>]]   |
| <b>Response</b>   | <p><b>SET</b></p> OK   |
| <b>Parameters</b> | <p><b>&lt;ID&gt;</b><br/>The ID allocated to the socket.</p> <p><b>&lt;remote host&gt; (UDP only)</b><br/>IPv4 address or domain name of the UDP server/client.</p> <p><b>&lt;remote port&gt; (UDP only)</b><br/>Port number of the UDP server/client.</p> <p><b>&lt;length&gt;</b><br/>The (signed) number of raw bytes to send. (See the description)</p> <p><b>&lt;done_event&gt;</b></p> |

|                    |   |
|--------------------|---|
|                    | SEND_DONE event. (0:disable, 1:enable)  |
| <b>Description</b> | <p>Send data through a socket.</p> <p>In synchronous mode, the value of the &lt;length&gt; parameter must be positive, and its maximum value is 2048. The payload byte sequence of &lt;length&gt; bytes must be directly followed by "AT+SSEND=&lt;ID&gt;,&lt;length&gt;\r\n". The payload byte sequence does not have to be followed by "\r" or "\n" and the next payload byte sequence can be sent again after receiving the "OK\r\n" response code from the firmware.</p> <p>In passthrough mode, the value of the &lt;length&gt; parameter must be 0, so that the command takes the form "AT+SSEND=&lt;ID&gt;,0\r\n". As soon as the firmware receives the command, the firmware enters the active passthrough state; all bytes fed into the AT stream is redirected to the associated socket stream. To exit the passthrough state, no byte should be fed into the AT stream for the duration of SSEND timeout duration in seconds (default: 1 second) to transition the active passthrough state to the idle passthrough state. The transition is notified by the +SEVENT:"SEND_IDLE" event. Upon receiving the idle event notification, the four magic bytes "AT\r\n" should be fed into the AT stream to exit the passthrough state. The magic bytes themselves will not be regarded as part of the payload as long as they are fed into the AT stream following the idle event notification, but if the characters following the idle event notification are different from the magic bytes, the fed bytes will indeed be regarded as part of the payload. The +SEVENT:"SEND_EXIT" event is raised upon exiting the passthrough mode.</p> <p>In buffered passthrough mode, the value of the &lt;length&gt; parameter must be positive, and its maximum value is 2048. The command takes the form "AT+SSEND=&lt;ID&gt;,-&lt;length&gt;\r\n", with the "-" sign preceding the &lt;length&gt; parameter. The buffered passthrough mode operates similarly to the passthrough mode. However, unlike the passthrough mode, the firmware maintains an internal byte buffer of size &lt;length&gt; and transfers the buffered byte onto the send queue only when the byte buffer is full. However, using this mode still does not guarantee that the receiver will always receive the payload in &lt;length&gt; bytes without fragmentation, as other factors such as the MTU size limit and other implementation-dependent features may affect the payload transfer process differently.</p> <p><b>NOTE:</b></p> <p>UART without HFC supports only synchronous mode.</p> |
| <b>Example</b>     | <p><b>[ Synchronous Mode ]</b></p> <p>AT+SSEND=0,6</p>  |

OK

Hello!

AT+SSEND=0,6,1

OK

Hello!

+SEVENT:"SEND\_DONE",6

### **[ Passthrough Mode ]**

AT+SSEND=0

Hello, World!

Goodbye, World!

[Wait for SSEND timeout duration to change the internal state to receive magic bytes and exit the continuous transmission state]

+SEVENT:"SEND\_IDLE",0,28,0,0

AT

OK

+SEVENT:"SEND\_EXIT",0,28,0

### **[Buffered Passthrough Mode]**

AT+SSEND=0,-8,1

TEST0001

+SEVENT:"SEND\_DONE",8

TEST0002

+SEVENT:"SEND\_DONE",8

+SEVENT:"SEND\_IDLE",0,16,0,0

TEST0003

+SEVENT:"SEND\_DONE",8

+SEVENT:"SEND\_IDLE",0,24,0,0

AT

OK

+SEVENT:"SEND\_EXIT",0,24,0

## 7.5 AT+SRECV

|             |  |
|-------------|--|
| Command     | <u><b>SET</b></u><br>AT+SRECV=<socket ID>[,<length>]<br><u><b>GET</b></u><br>AT+SRECV?   |
| Response    | <u><b>SET</b></u><br>OK<br><u><b>GET</b></u><br>+SRECV:<socket_ID>,<bufferd_length><br>...<br>OK   |
| Parameters  | <p><b>&lt;socket ID&gt;</b><br/>The ID allocated to the socket.</p> <p><b>&lt;length&gt;</b><br/>The maximum number of raw bytes to read</p> <p><b>&lt;bufferd_length&gt;</b><br/>The number of raw bytes currently buffered<br/>If omitted or set to 0, it is set to the maximum value supported by the firmware.</p> |
| Description | Read buffered data from the network stack (lwip).<br><br><b>NOTE:</b> <ol style="list-style-type: none"> <li>1) AT+SRECV command can be used only when passive mode is set with AT+SRECVMODE command.</li> <li>2) If it is UDP data, it will be lost when the buffer is full.</li> </ol>                               |
| Example     | AT+SLIST?<br>+SLIST:0,"TCP","192.168.200.1",50000,0<br>+SLIST:1,"UDP","0.0.0.0",0,60001<br>OK<br><br>+SEVENT:"RECV_READY",0,1024<br>+SEVENT:"RECV_READY",1,1024<br><br>AT+SRECV?<br>+SRECV:0,7168<br>+SRECV:1,7168   |

|  |  |
|--|--|
|  | OK<br>AT+SRECV=0<br>+RXD:0,4096,"192.168.200.1",50000<br>OK<br>AT+SRECV=1<br>+RXD:1,1024,"192.168.200.1",60000<br>OK<br><br>+SEVENT:"RCV_READY",0,3072<br>+SEVENT:"RCV_READY",1,6144 |
|--|--|

## 7.6 AT+SRECVMODE

|                    |  |
|--------------------|--|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+SRECVMODE=<mode>[,<event>]<br><br><u><b>GET</b></u><br>AT+SRECVMODE?   |
| <b>Response</b>    | <u><b>SET</b></u><br>OK<br><br><u><b>GET</b></u><br>+SRECVMODE:<mode>,<event><br>OK  |
| <b>Parameters</b>  | <b>&lt;mode&gt;</b><br>0 : active*<br>1 : passive<br><br><b>&lt;event&gt;</b><br>0 : ready event disable<br>1 : ready event enable*  |
| <b>Description</b> | Configures how data is read from the network stack (lwip).<br>If the event parameter is set to 1 in passive mode, a RCV_READY event occurs when there is buffered data.<br>The event does not occur again until the buffered data is read with the AT+SRECV command. |
| <b>Example</b>     | AT+SRECVMODE=1   |

|  |   |
|--|---|
|  | OK<br>AT+SRECVMODE?<br>+SRECVMODE:1,0<br>OK<br>AT+SRECVMODE=1,1<br>OK<br>AT+SRECVMODE?<br>+SRECVMODE:1,1<br>OK<br><br>AT+SRECVMODE=0<br>OK<br>AT+SRECVMODE?<br>+SRECVMODE:0,0<br>OK |
|--|---|

## 7.7 AT+SRECVINFO

|                    |   |
|--------------------|---|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+SRECVINFO=<mode><br><u><b>GET</b></u><br>AT+SRECVINFO?  |
| <b>Response</b>    | <u><b>SET</b></u><br>OK<br><u><b>GET</b></u><br>+SRECVINFO:<mode><br>OK   |
| <b>Parameters</b>  | <mode><br>0 : terse*<br>1 : verbose   |
| <b>Description</b> | Configure the information level of “+RXD” message.<br><br>NOTE:<br>The AT+SRECVINFO command is the same as the previous AT+SRXLOGLEVEL command. Only the command name is different. |
| <b>Example</b>     | AT+SRECVINFO =1<br>OK   |

|  |                                      |
|--|--------------------------------------|
|  | AT+SRECVINFO?<br>+ SRECVINFO:1<br>OK |
|--|--------------------------------------|

## 7.8 AT+SADDRINFO

|                    |  |
|--------------------|--|
| <b>Command</b>     | <u>SET</u><br>AT+SADDRINFO="<domain_name>"   |
| <b>Response</b>    | <u>SET</u><br>+SADDRINFO:"<address>"<br>OK   |
| <b>Parameters</b>  | <domain_name><br>Domain name<br><address><br>IPv4 address  |
| <b>Description</b> | Check the IP address from the domain name.   |
| <b>Example</b>     | AT+SADDRINFO =" <a href="http://www.google.com">www.google.com</a> "<br>+SADDRINFO:"142.250.199.100"<br>OK |

## 7.9 AT+STCPNODELAY

|                   |   |
|-------------------|---|
| <b>Command</b>    | <u>SET</u><br>AT+STCPNODELAY=<socket ID>,{0 1}<br><u>GET</u><br>AT+STCPNODELAY? |
| <b>Response</b>   | <u>SET</u><br>OK<br><u>GET</u><br>+STCPNODELAY:<socket_ID>,<status><br>OK       |
| <b>Parameters</b> | <socket ID><br>The ID allocated to the socket.<br><status><br>0 : disable       |



|                    |   |
|--------------------|---|
|                    | 1 : enable  |
| <b>Description</b> | Enable or disable TCP Nagle's algorithm.  |
| <b>Example</b>     | <p><b>&lt; TCP Server &gt;</b></p> <pre> AT+SOPEN="TCP",50000 +SOPEN=0 OK +SEVENT:"CONNECT",1 AT+SLIST? +SLIST:0,"TCP","0.0.0.0",0,50000 +SLIST:1,"TCP","192.168.200.2",52432,0 OK  AT+STCPNODELAY? +STCPNODELAY:1,0 OK AT+STCPNODELAY=1,1 OK AT+STCPNODELAY? +STCPNODELAY:1,1 OK  <b>&lt; TCP Client &gt;</b> AT+SOPEN="TCP","192.168.200.1",50000 +SOPEN:0 OK AT+SLIST? +SLIST:0,"TCP","192.168.200.1",50000,0 OK  AT+STCPNODELAY? +STCPNODELAY:0,0 OK AT+STCPNODELAY=0,1 OK AT+STCPNODELAY? +STCPNODELAY:0,1 OK </pre> |

## 7.10 AT+STIMEOUT

|                    |   |
|--------------------|---|
| <b>Command</b>     | <u><b>SET</b></u><br>AT+STIMEOUT="<command>",<timeout><br><u><b>GET</b></u><br>AT+STIMEOUT?   |
| <b>Response</b>    | <u><b>SET</b></u><br>OK<br><u><b>GET</b></u><br>+STIMEOUT:"<command>",<timeout><br>...<br>OK  |
| <b>Parameters</b>  | <command><br>"SOPEN", "SSEND"<br><timeout><br>Timeout in seconds. (0 : no timeout)  |
| <b>Description</b> | Configure the response timeout for the specified socket command.<br><br>Default timeout : <ul style="list-style-type: none"> <li>- SOPEN : 30 secs</li> <li>- SSEND : 1 sec</li> </ul>                    |
| <b>Example</b>     | AT+STIMEOUT?<br>+STIMEOUT:"SOPEN",30<br>+STIMEOUT:"SSEND",1<br>OK<br><br>AT+STIMEOUT="SOPEN",60<br>OK<br>AT+STIMEOUT="SSEND",3<br>OK<br>AT+STIMEOUT?<br>+STIMEOUT:"SOPEN",60<br>+STIMEOUT:"SSEND",3<br>OK |

## 7.11 +SEVENT

| Response   | +SEVENT:<event>,<socket ID>[,<parameter 1>,...,<parameter N>]  |
|------------|--|
| Parameters | <p><b>&lt;event&gt;</b></p> <p>“CONNECT”,&lt;socket ID&gt;</p> <p>“CLOSE”,&lt;socket ID&gt;</p><br><p>“SEND_IDLE”,&lt;socket ID&gt;,&lt;done&gt;,&lt;drop&gt;,&lt;wait&gt;</p> <p>“SEND_DONE”,&lt;socket ID&gt;,&lt;done&gt;</p> <p>“SEND_DROP”,&lt;socket ID&gt;,&lt;drop&gt;</p> <p>“SEND_EXIT”,&lt;socket ID&gt;,&lt;done&gt;,&lt;drop&gt;</p> <p>“SEND_ERROR”,&lt;socket ID&gt;,&lt;error&gt;,”&lt;description&gt;”</p><br><p>“RECV_READY”,&lt;socket ID&gt;,&lt;length&gt;</p> <p>“RECV_ERROR”,&lt;socket ID&gt;,&lt;error&gt;,”&lt;description&gt;”</p><br><p><b>&lt;socket ID&gt;</b></p> <p>Socket ID</p><br><p><b>&lt;done&gt;</b></p> <p>The length of the sent payload.</p><br><p><b>&lt;drop&gt;</b></p> <p>The length of the dropped payload.</p><br><p><b>&lt;wait&gt;</b></p> <p>The length of the buffered payload.</p><br><p><b>&lt;length&gt;</b></p> <p>The length of the receivable payload.</p><br><p><b>&lt;error&gt;</b></p> <p>error code</p><br><p><b>&lt;description&gt;</b></p> <p>string describing the error code</p> |

|                    |   |
|--------------------|---|
|                    | <p>NOTE:</p> <p>The error code may not match the POSIX error code.</p> <p>The error code defined in the errno.h file included in the ARM Toolchain is different from the POSIX error code.</p>  |
| <b>Description</b> | Asynchronously raised socket event logs.  |
| <b>Example</b>     | <pre>+SEVENT:"CONNECT",1 +SEVENT:"CLOSE",1  +SEVENT:"SEND_IDLE",1,1500,152,200 +SEVENT:"SEND_DROP",1,152 +SEVENT:"SEND_EXIT",1,1700,152 +SEVENT:"SEND_ERROR",1,-104,"Connection reset by peer"  +SEVENT:"RECV_READY",1,1488 +SEVENT:"RECV_ERROR",1,-128,"Socket is not connected"</pre> |

## 7.12 +RXD

|                   |  |
|-------------------|--|
| <b>Response</b>   | <p><b><u>RX Log Level (Terse)</u></b></p> <p>+RXD:&lt;socket ID&gt;,&lt;actual read length&gt;<br/>&lt;raw bytes&gt;</p> <p><b><u>RX Log Level (Verbose)</u></b></p> <p>+RXD:&lt;socket ID&gt;,&lt;actual read length&gt;,"&lt;remote IP&gt;",&lt;remote port&gt;<br/>&lt;raw bytes&gt;</p>  |
| <b>Parameters</b> | <p><b>&lt;socket ID&gt;</b><br/>The ID allocated to the socket.</p> <p><b>&lt;max read length&gt;</b><br/>The maximum number of bytes to read. (Max: 2048)</p> <p><b>&lt;actual read length&gt;</b><br/>Actual number of bytes read.</p> <p><b>&lt;remote IP&gt;,&lt;remote port&gt;</b><br/>The remote IP and port.</p> <p><b>&lt;raw bytes&gt;</b><br/>The received raw bytes (0x00~0xFF) payload.</p> |

|                    |  |
|--------------------|--|
| <b>Description</b> | <p>An event log for a received packet with payload.</p> <p>Upon receiving packets, +RXD event logs will automatically appear on the terminal output.</p> <p>Note that there will be no 'OK' message following the event log.</p> |
| <b>Example</b>     | <p><b><u>RX Log Level (Terse)</u></b></p> <p>+RXD=0,15<br/>ABCDE12345,.?="+</p> <p><b><u>RX Log Level (Verbose)</u></b></p> <p>+RXD=0,12,"192.168.200.1",5025<br/>HELLO,WORLD!</p>   |

## 8 Test Application

### 8.1 Command Line Interface (raspi-atcmd-cli)

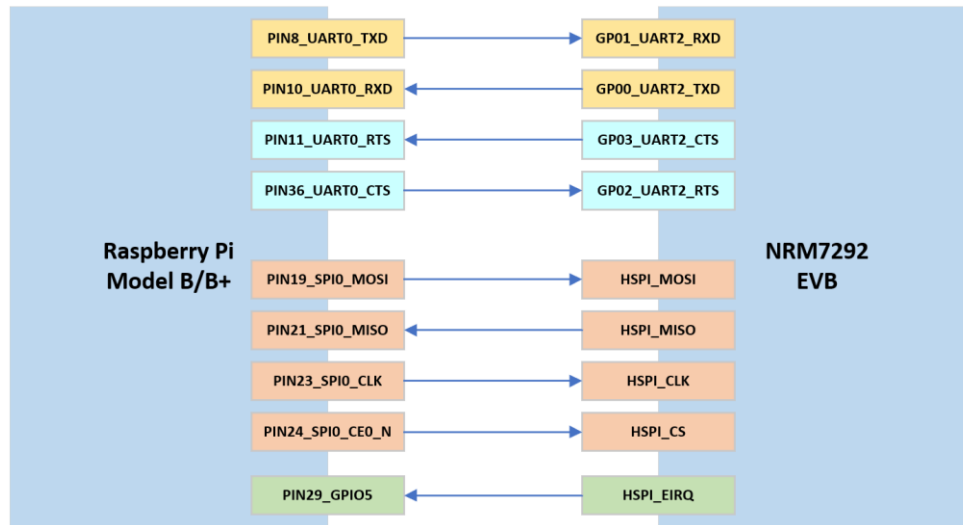
CLI application is a Linux program running on Raspberry Pi for AT-command communication via UART or SPI. In the CLI application, as in terminal program via UART, the user can enter the AT command and check the response to the command.

The NRM7292 EVB is using the Raspberry Pi 3 B/B+ as a host. The Raspberry Pi board is connected to the NRM7292 EVB through a 40-pin header. The 40-pin header has signals for UART and SPI.



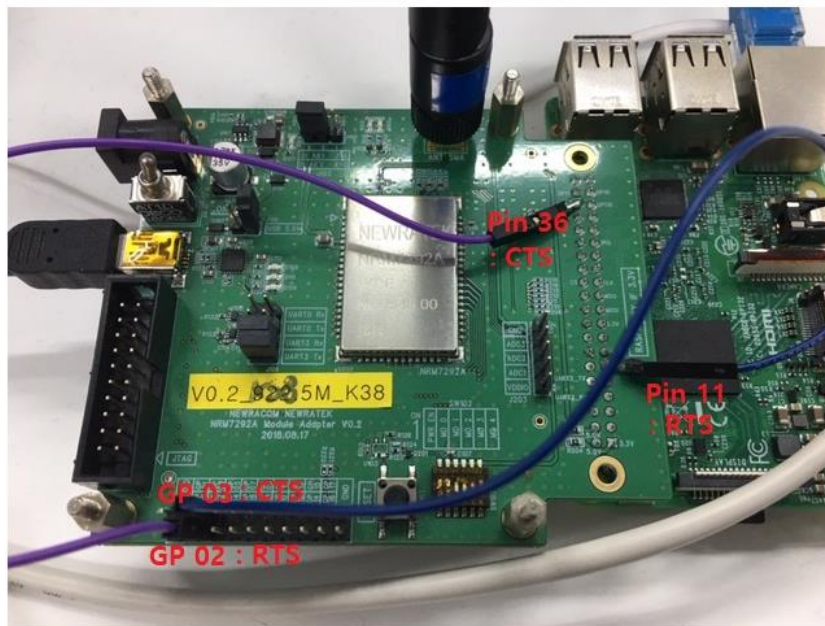
Figure 8.1 Pin map of 40-pin header for Raspberry Pi 3 B/B+

The NRM7292 EVB and Raspberry Pi board is connected as shown in the Figure 8.2.

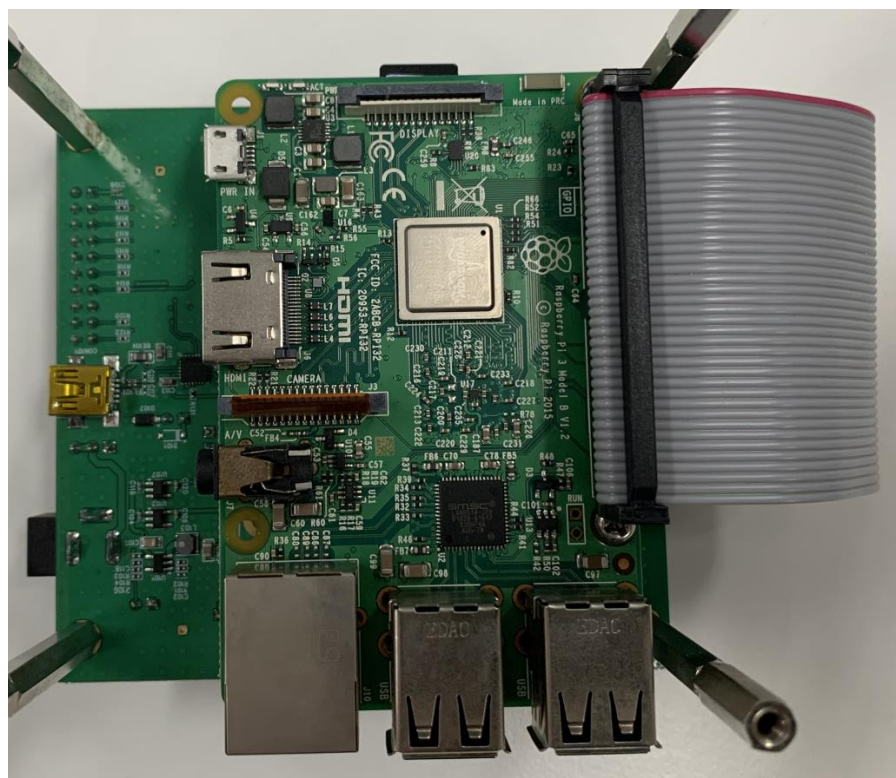
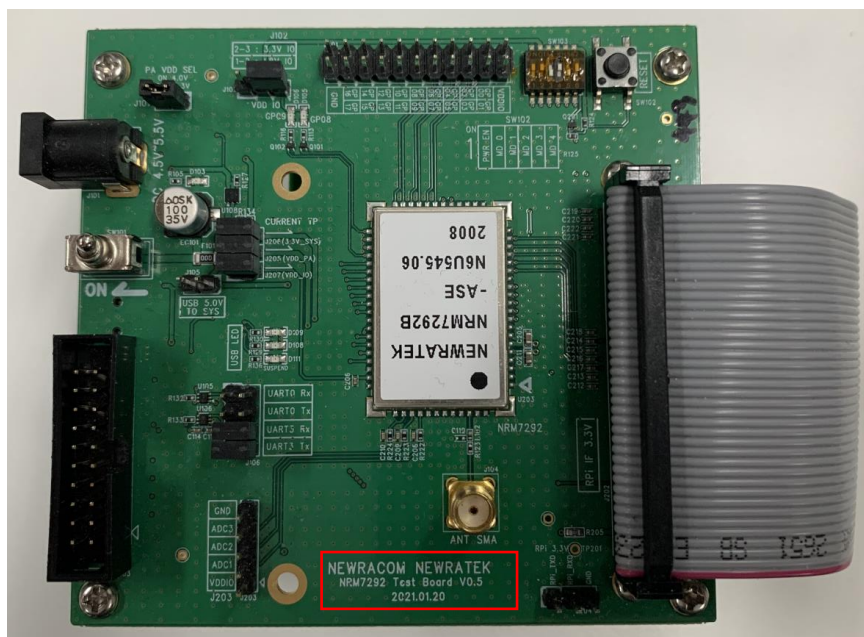


**Figure 8.2 Connection between NRM7292 EVB and Raspberry Pi**

Both PIN11\_UART0\_RTS and PIN36\_UART0\_CTS used for hardware flow control on the UART needs to be directly connected to a 20-pin header in the NRM7292 EVB v0.2 by a jumper wire.



The NRM7292 EVB v0.5 can use the hardware flow control on the UART without a jumper wire.





### 8.1.1 Source files

| File                       | Description  |
|----------------------------|--|
| common.h                   | Common header file   |
| main.c                     | CLI related functions.   |
| Makefile                   | Make file for building.  |
| nrc-atcmd.c<br>nrc-atcmd.h | AT command handler   |
| nrc-hspi.c<br>nrc-hspi.h   | Protocol driver for HSPI.<br><b>*Refer to this file to communicate with the ATCMD firmware via HSPI.</b> |
| nrc-iperf.c<br>nrc-iperf.h | Iperf server/client  |
| raspi-hif.c<br>raspi-hif.h | Wrapper for user mode driver.  |
| raspi-eirq.c               | User mode driver for GPIO EIRQ.  |
| raspi-spi.c                | User mode driver for SPI.  |
| raspi-uart.c               | User mode driver for UART.   |
| scripts/                   | Script files   |

**Table 8.1 raspi-atcmd-cli source files**

### 8.1.2 Build

Copy the “atcmd/host/raspi-atcmd-cli” directory to the Raspberry Pi’s home directory. And build the CLI application with the make command.

```
$ cd $HOME
```

```
$ cd raspi-atcmd-cli
```

```
$ make clean
```

```
removed 'raspi-atcmd-cli'
```

```
$ make
```

```
cc -g -o raspi-atcmd-cli raspi-spi.c raspi-uart.c raspi-eirq.c raspi-hif.c nrc-hspi.c nrc-atcmd.c nrc-iperf.c main.c  
-pthread -Wall -lpthread
```

### 8.1.3 Run

- **Help**

\$ ./raspi-atcmd-cli [-h|--help]

```
raspi-atcmd-cli version 1.3.3
Copyright (c) 2019-2023 <NEWRACOM LTD>

Usage:
$ ./raspi-atcmd-cli -S [-D <device>] [-E <trigger>] [-c <clock>] [-s <script> [-n]]
$ ./raspi-atcmd-cli -U [-D <device>] [-b <baudrate>] [-s <script> [-n]]
$ ./raspi-atcmd-cli -U -f [-D <device>] [-b <baudrate>] [-s <script> [-n]]

UART/SPI:
-D, --device #       Specify the device. (default: /dev/spidev0.0, /dev/ttyAMA0)
-s, --script #       Specify the script file.
-n, --noexit #       Do not exit the script when the AT command responds with an error.

SPI:
-S --spi             Use the SPI to communicate with the target.
-E, --eirq #         Use EIRQ mode for the SPI. (0:low, 1:high, 2:falling, 3:rising)
-c, --clock #        Specify the clock frequency for the SPI. (default: 20000000 Hz)

UART:
-U --uart            Use the UART to communicate with the target.
-f --flowctrl        Enable RTS/CTS signals for the hardware flow control on the UART. (default: off)
-b, --baudrate #     Specify the baudrate for the UART. (default: 115200 bps)

Miscellaneous:
-v, --version        Print version information and quit.
-h, --help           Print this message and quit.
```

- **SPI**

The maximum clock frequency is 20MHz.

\$ sudo ./raspi-atcmd-cli -S [-D <device>] [-E <trigger>] [-c <clock>] [-s <script> [-n]]

```
$ sudo ./raspi-atcmd-cli -S -c 20000000 -E 2

[ SPI ]
- device: /dev/spidev0.0
- clock: 20000000 Hz
- eirq: falling

#
```

- **UART**

The maximum baud rate is 115,200bps without the hardware flow control.

\$ sudo ./raspi-atcmd-cli -U [-D <device>] [-b <baudrate>] [-s <script> [-n]]

```
$ sudo ./raspi-atcmd-cli -U -b 115200
```

```
[ UART ]
```

```
- device: /dev/ttyAMA0
```

```
- baudrate : 115200
```

```
#
```

- **UART\_HFC**

If the baud rate setting is more than 115,200bps, the hardware flow control needs to be enabled with -f option on the UART.

\$ sudo ./raspi-atcmd-cli -U -f [-D <device>] [-b <baudrate>] [-s <script> [-n]]

```
$ sudo ./raspi-atcmd-cli -U -f -b 2000000
```

```
[ UART_HFC ]
```

```
- device: /dev/ttyAMA0
```

```
- baudrate : 2000000
```

```
#
```

- **Examples**

Getting the informations.

```
# AT
```

```
SEND: AT
```

```
RECV: OK
```

```
# AT+VER?
```

```
SEND: AT+VER?
```

```
RECV: +VER:"1.3.4","1.23.5"
```

```
RECV: OK
```

```
# AT+WMACADDR?
```

```
SEND: AT+WMACADDR?
```

```
RECV: +WMACADDR:"8c:0f:fa:00:29:43"
```

RCV: OK

# AT+WCCOUNTRY?

SEND: AT+WCCOUNTRY?

RCV: +WCCOUNTRY:"US"

RCV: OK

# AT+WTPXPOWER?

SEND: AT+WTPXPOWER?

RCV: +WTPXPOWER:17

RCV: OK

# AT+WRRATECTRL?

SEND: AT+WRRATECTRL?

RCV: +WRRATECTRL:1

RCV: OK

# AT+WIPADDR?

SEND: AT+WIPADDR?

RCV: +WIPADDR:"0.0.0.0","0.0.0.0","0.0.0.0"

RCV: OK

### Connecting to an AP.

# AT+WCONN?

SEND: AT+WCONN?

RCV: +WCONN:"halow","00:00:00:00:00:00","open","","disconnected"

RCV: OK

# AT+WSCAN

SEND: AT+WSCAN

RCV: +WSCAN:"8c:0f:fa:00:28:1f",906.0,-39,"[WPA3-SAE-CCMP][ESS]","halow\_atcmd\_sae"

RCV: +WSCAN:"8c:0f:fa:00:28:11",925.0,-68,"[WPA3-OWE-CCMP][ESS]","halow\_fota"

RCV: +WSCAN:"8c:0f:fa:00:28:1e",903.5,-93,"[ESS]","halow\_s1g\_demo\_open"

RCV: OK

# AT+WCONN="halow\_atcmd\_sae","sae","12345678"

SEND: AT+WCONN="halow\_atcmd\_sae","sae","12345678"

RCV: OK

# AT+WCONN?

SEND: AT+WCONN?

RCV: +WCONN:"halow\_atcmd\_sae","8c:0f:fa:00:28:1f","wpa3-sae","12345678","connected"

RCV: OK

# AT+WDHCP

SEND: AT+WDHCP

RCV: +WDHCP:"192.168.200.18","255.255.255.0","192.168.200.1"

RCV: OK

```
# AT+WIPADDR?
SEND: AT+WIPADDR?
RECV: +WIPADDR:"192.168.200.18","255.255.255.0","192.168.200.1"
RECV: OK

# AT+WPING="192.168.200.1"
SEND: AT+WPING="192.168.200.1"
RECV: +WPING:64,"192.168.200.1",1,64,5
RECV: +WPING:64,"192.168.200.1",2,64,5
RECV: +WPING:64,"192.168.200.1",3,64,149
RECV: +WPING:64,"192.168.200.1",4,64,4
RECV: +WPING:64,"192.168.200.1",5,64,5
RECV: OK
```

### Sending and receiving the data with a socket for TCP client.

```
# AT+SOPEN="TCP","192.168.200.1",50000
SEND: AT+SOPEN="TCP","192.168.200.1",50000
RECV: +SOPEN:0
RECV: OK

# AT+SLIST?
SEND: AT+SLIST?
RECV: +SLIST:0,"TCP","192.168.200.1",50000,52432
RECV: OK

# AT+SSEND=0,10
SEND: AT+SSEND=0,10
RECV: OK

# ABCDEFGHIJKLMNOPQRSTUVWXYZ
SEND: DATA 10

#   RECV: +RXD:0,10

# AT+SSEND=0
SEND: AT+SSEND=0
RECV: OK

# DAJFKDAJFKDAJFDKAJFAKFJDK
SEND: DATA 25

#   RECV: +RXD:0,25
RECV: +SEVENT:"SEND_IDLE",0,25,0,0

# DKAJFKDAJFEKJAFKDJFADKJFAKDJFAKEJFKADJFAKEJFKAJDFKDJAFDKJFADK
SEND: DATA 61
```

```
# RECV: +RXD:0,61
RECV: +SEVENT:"SEND_IDLE",0,86,0,0

# AT
SEND: AT
RECV: OK

# RECV: +SEVENT:"SEND_EXIT",0,86,0
```

Closing all sockets.

```
# AT+SLIST?
SEND: AT+SLIST?
RECV: +SLIST:0,"TCP","192.168.200.1",50000,52432
RECV: OK

# AT+SCLOSE
SEND: AT+SCLOSE
RECV: +SCLOSE:0
RECV: OK

# EXIT
```

### 8.1.4 Run with a script

CLI application provides the option to run the script file. (-s/--script)

|                |  |
|----------------|--|
| UART/SPI:      |  |
| -s, --script # | Specify the script file.   |
| -n, --noexit # | Do not exit the script when the AT command responds with an error. |

The script file can be created using the AT command and script command.

| Command             | Description  | Example                                  |
|---------------------|--|--|
| CALL <script_file>  | Read and run the specified script file.  | CALL wifi_connect<br>CALL wifi/connect   |
| LOOP <line> <count> | Repeat next lines.<br><line>: number of lines to repeat<br><count>: number of repetitions. | LOOP 2 5<br>AT+SSEND=0,1024<br>DATA 1024 |
| DATA <length>       | Send payload with random value.  | DATA 1024                                |

|                    |  |                                       |
|--------------------|--|---------------------------------------|
| WAIT <time>{s m u} | Wait for the specified time.<br>s: sec<br>m: msec<br>u: usec | WAIT 1s<br>WAIT 1000m<br>WAIT 100u    |
| ECHO "<message>"   | Print a message.   | ECHO "AT Command"                     |
| TIME               | Print current time.  | TIME                                  |
| HOLD               | Pause until there is keyboard input.                         | ECHO "Run an AP in open mode"<br>HOLD |
| EXIT               | Exit script.   | EXIT                                  |

Users can refer to the script files under the "raspi-atcmd-cli/scripts" directory.

```

raspi-atcmd-cli/scripts/
├── socket-tcp-client-send
├── socket-tcp-client-send-passthrough
├── socket-tcp-client-send-passthrough-buffered
├── socket-tcp-server
├── socket-tcp-server-send
├── socket-tcp-server-send-passthrough
├── socket-tcp-server-send-passthrough-buffered
├── socket-udp-client-send
├── socket-udp-client-send-passthrough
├── socket-udp-client-send-passthrough-buffered
├── socket-udp-server
├── socket-udp-server-send
├── socket-udp-server-send-passthrough
├── socket-udp-server-send-passthrough-buffered
├── softap-tcp-client-send-normal
├── softap-tcp-client-send-passthrough
├── softap-tcp-server
├── softap-udp-client-send-normal
├── softap-udp-client-send-passthrough
├── softap-udp-server
├── sta-tcp-client-send-normal
└── sta-tcp-client-send-passthrough

```

- |— sta-tcp-server
- |— sta-udp-client-send-normal
- |— sta-udp-client-send-passthrough
- |— sta-udp-server
- |— wifi-connect-open-dhcp-auto-kr-mic
- |— wifi-connect-open-dhcp-auto-us
- |— wifi-connect-open-dhcp-kr-mic
- |— wifi-connect-open-dhcp-kr-usn
- |— wifi-connect-open-dhcp-us
- |— wifi-connect-wpa2-psk-dhcp-auto-kr-mic
- |— wifi-connect-wpa2-psk-dhcp-auto-us
- |— wifi-connect-wpa2-psk-dhcp-kr-mic
- |— wifi-connect-wpa2-psk-dhcp-us
- |— wifi-connect-wpa3-owe-dhcp-auto-kr-mic
- |— wifi-connect-wpa3-owe-dhcp-auto-us
- |— wifi-connect-wpa3-owe-dhcp-kr-mic
- |— wifi-connect-wpa3-owe-dhcp-us
- |— wifi-connect-wpa3-sae-dhcp-auto-kr-mic
- |— wifi-connect-wpa3-sae-dhcp-auto-us
- |— wifi-connect-wpa3-sae-dhcp-kr-mic
- |— wifi-connect-wpa3-sae-dhcp-us
- |— wifi-softap-open-dhcps-kr-mic
- |— wifi-softap-open-dhcps-kr-usn
- |— wifi-softap-open-dhcps-us
- |— wifi-softap-wpa2-psk-dhcps-kr-mic
- |— wifi-softap-wpa2-psk-dhcps-us

### 8.1.5 Iperf

The CLI application supports the iperf2 command used for network performance measurement. However, the available options are limited as shown below.

# iperf {-h|--help}

Usage: iperf {-s}|{-c <host>} [options]

Client/Server:

-i, --interval #            seconds between periodic bandwidth reports (default: 1 sec)



|                     |  |
|---------------------|--|
| -p, --port #        | server port to listen on/connect to (default: 5001)                        |
| -u, --udp           | use UDP rather than TCP  |
| Server specific:    |  |
| -s, --server        | run in server mode   |
| Client specific:    |  |
| -c, --client <host> | run in client mode, connecting to <host>                                   |
| -t, --time #        | time in seconds to transmit for (default: 10 sec)                          |
| -P, --passthrough   | transmit in passthrough mode   |
| -N, --negative      | use negative length for buffered passthrough mode (always negative in UDP) |
| -D, --done_vent     | enable SEND_DONE event   |
| Miscellaneous:      |  |
| -h, --help          | print this message and quit  |

The iperf command can be run after completing the Wi-Fi connection and IP setup.

Wi-Fi connection and IP setup can be done in one of two ways:

- Enter AT command in the CLI application.

```
# AT+WSCAN
SEND: AT+WSCAN
RECV: +WSCAN:"8c:0f:fa:00:28:1f",914.0,-38,"[WPA3-SAE-CCMP][ESS]","halow_atcmd_sae"
RECV: OK

# AT+WCONN="halow_atcmd_sae","sae","12345678"
SEND: AT+WCONN="halow_atcmd_sae","sae","12345678"
RECV: OK

# AT+WDHCP
SEND: AT+WDHCP
RECV: +WDHCP:"192.168.200.18","255.255.255.0","192.168.200.1"
RECV: OK
```

- Specify a script file containing AT command with the -s option when running the CLI application.

```
$ sudo ./raspi-atcmd-cli -S -s scripts/example/wifi-connect-wpa3-sae-dhcp
```

```
CALL: scripts/examples/wifi-connect-wpa3-sae-dhcp

SEND: AT
RECV: OK
SEND: AT+WDISCONN
RECV: OK
```

```

ECHO: Run an AP in WPA3-SAE.
ECHO:  - SSID : halow_atcmd_sae
ECHO:  - Password : 12345678
ECHO:  - IP : 192.168.200.1
ECHO:  - DHCP Server
HOLD: Press ENTER to continue.

SEND: AT+WSCAN
RECV: +WSCAN:"8c:0f:fa:00:28:1f",906.0,-39,"[WPA3-SAE-CCMP][ESS]","halow_atcmd_sae"
RECV: OK
SEND: AT+WDISCONN
RECV: OK
SEND: AT+WCONN="halow_atcmd_sae","wpa3-sae","12345678"
RECV: OK
SEND: AT+WCONN?
RECV: +WCONN:"halow_atcmd_sae","8c:0f:fa:00:28:1f","wpa3-sae","12345678","connected"
RECV: OK
SEND: AT+WDHCP
RECV: +WDHCP:"192.168.200.18","255.255.255.0","192.168.200.1"
RECV: OK

DONE: scripts/examples/wifi-connect-wpa3-sae-dhcp

```

## ● Iperf TCP Client

### Normal Send Mode

```

# iperf -c 192.168.200.1

[ IPERF OPTION ]
- role: client
- protocol: tcp
- server_port: 5001
- server_ip: 192.168.200.1
- send_time: 10
- send_passthrough: off
- report_interval: 1

[ IPERF TCP Client ]
Sending 1470 byte datagram ...

```

| Interval      | Transfer      | Bandwidth |
|---------------|---------------|-----------|
| 0.0 ~ 1.0 sec | 159.35 KBytes | 1.30 Mb/s |
| 1.0 ~ 2.0 sec | 166.52 KBytes | 1.36 Mb/s |
| 2.0 ~ 3.0 sec | 167.96 KBytes | 1.37 Mb/s |
| 3.0 ~ 4.0 sec | 153.60 KBytes | 1.26 Mb/s |

```

4.0 ~ 5.0 sec  167.96 KBytes  1.37 Mb/s/sec
5.0 ~ 6.0 sec  165.09 KBytes  1.35 Mb/s/sec
6.0 ~ 7.0 sec  166.52 KBytes  1.36 Mb/s/sec
7.0 ~ 8.0 sec  169.39 KBytes  1.38 Mb/s/sec
8.0 ~ 9.0 sec  169.39 KBytes  1.38 Mb/s/sec
9.0 ~ 10.0 sec 163.65 KBytes  1.37 Mb/s/sec
0.0 ~ 10.0 sec  1.61 MBytes  1.35 Mb/s/sec
Sent 1149 datagrams
Done

```

### Passthrough Send Mode

```

# iperf -c 192.168.200.1 -P

[ IPERF OPTION ]
- role: client
- protocol: tcp
- server_port: 5001
- server_ip: 192.168.200.1
- send_time: 10
- send_passthrough: on
- report_interval: 1

[ IPERF TCP Client ]
Sending 1470 byte datagram ...
Interval      Transfer      Bandwidth
0.0 ~ 1.0 sec  341.66 KBytes  2.78 Mb/s/sec
1.0 ~ 2.0 sec  337.35 KBytes  2.76 Mb/s/sec
2.0 ~ 3.0 sec  340.22 KBytes  2.77 Mb/s/sec
3.0 ~ 4.0 sec  341.66 KBytes  2.79 Mb/s/sec
4.0 ~ 5.0 sec  344.53 KBytes  2.80 Mb/s/sec
5.0 ~ 6.0 sec  334.48 KBytes  2.74 Mb/s/sec
6.0 ~ 7.0 sec  343.10 KBytes  2.79 Mb/s/sec
7.0 ~ 8.0 sec  335.92 KBytes  2.74 Mb/s/sec
8.0 ~ 9.0 sec  341.66 KBytes  2.80 Mb/s/sec
9.0 ~ 10.0 sec 312.95 KBytes  2.67 Mb/s/sec
0.0 ~ 10.0 sec  3.29 MBytes  2.76 Mb/s/sec
Sent 2350 datagrams
Done

```

### Buffered Passthrough Send Mode

```

# iperf -c 192.168.200.1 -P -N

```

## [ IPERF OPTION ]

```
- role: client
- protocol: tcp
- server_port: 5001
- server_ip: 192.168.200.1
- send_time: 10
- send_passthrough: on (-)
- report_interval: 1
```

## [ IPERF TCP Client ]

Sending 1470 byte datagram ...

| Interval       | Transfer      | Bandwidth |
|----------------|---------------|-----------|
| 0.0 ~ 1.0 sec  | 302.90 KBytes | 2.48 Mb/s |
| 1.0 ~ 2.0 sec  | 297.16 KBytes | 2.42 Mb/s |
| 2.0 ~ 3.0 sec  | 298.59 KBytes | 2.43 Mb/s |
| 3.0 ~ 4.0 sec  | 295.72 KBytes | 2.42 Mb/s |
| 4.0 ~ 5.0 sec  | 297.16 KBytes | 2.43 Mb/s |
| 5.0 ~ 6.0 sec  | 298.59 KBytes | 2.45 Mb/s |
| 6.0 ~ 7.0 sec  | 295.72 KBytes | 2.40 Mb/s |
| 7.0 ~ 8.0 sec  | 298.59 KBytes | 2.43 Mb/s |
| 8.0 ~ 9.0 sec  | 294.29 KBytes | 2.41 Mb/s |
| 9.0 ~ 10.0 sec | 279.93 KBytes | 2.37 Mb/s |
| 0.0 ~ 10.0 sec | 2.89 MBytes   | 2.42 Mb/s |

Sent 2061 datagrams

Done

## Remote Iperf TCP Server

```
pi@raspberrypi:~ $ iperf -s -i 1
```

```
-----
Server listening on TCP port 5001
```

```
TCP window size: 85.3 KByte (default)
-----
```

```
[ 4] local 192.168.200.1 port 5001 connected with 192.168.200.18 port 52437
```

| [ ID] | Interval     | Transfer    | Bandwidth |
|-------|--------------|-------------|-----------|
| [ 4]  | 0.0- 1.0 sec | 159 KBytes  | 1.31 Mb/s |
| [ 4]  | 1.0- 2.0 sec | 167 KBytes  | 1.36 Mb/s |
| [ 4]  | 2.0- 3.0 sec | 168 KBytes  | 1.37 Mb/s |
| [ 4]  | 3.0- 4.0 sec | 154 KBytes  | 1.26 Mb/s |
| [ 4]  | 4.0- 5.0 sec | 168 KBytes  | 1.38 Mb/s |
| [ 4]  | 5.0- 6.0 sec | 165 KBytes  | 1.35 Mb/s |
| [ 4]  | 6.0- 7.0 sec | 166 KBytes  | 1.36 Mb/s |
| [ 4]  | 7.0- 8.0 sec | 168 KBytes  | 1.38 Mb/s |
| [ 4]  | 8.0- 9.0 sec | 168 KBytes  | 1.37 Mb/s |
| [ 4]  | 9.0-10.0 sec | 166 KBytes  | 1.36 Mb/s |
| [ 4]  | 0.0-10.0 sec | 1.61 MBytes | 1.35 Mb/s |

```
[ 5] local 192.168.200.1 port 5001 connected with 192.168.200.18 port 52438
```

```
[ 5] 0.0- 1.0 sec 326 KBytes 2.67 Mb/s
```

```

[ 5] 1.0- 2.0 sec  334 KBytes  2.73 Mbits/sec
[ 5] 2.0- 3.0 sec  340 KBytes  2.78 Mbits/sec
[ 5] 3.0- 4.0 sec  339 KBytes  2.77 Mbits/sec
[ 5] 4.0- 5.0 sec  342 KBytes  2.81 Mbits/sec
[ 5] 5.0- 6.0 sec  333 KBytes  2.73 Mbits/sec
[ 5] 6.0- 7.0 sec  342 KBytes  2.80 Mbits/sec
[ 5] 7.0- 8.0 sec  334 KBytes  2.74 Mbits/sec
[ 5] 8.0- 9.0 sec  342 KBytes  2.80 Mbits/sec
[ 5] 9.0-10.0 sec  325 KBytes  2.66 Mbits/sec
[ 5] 10.0-11.0 sec  17.0 KBytes  139 Kbits/sec
[ 5] 11.0-12.0 sec  0.00 Bytes  0.00 bits/sec
[ 5] 0.0-12.0 sec  3.29 MBytes  2.30 Mbits/sec
[ 4] local 192.168.200.1 port 5001 connected with 192.168.200.18 port 52439
[ 4] 0.0- 1.0 sec  288 KBytes  2.36 Mbits/sec
[ 4] 1.0- 2.0 sec  296 KBytes  2.42 Mbits/sec
[ 4] 2.0- 3.0 sec  294 KBytes  2.41 Mbits/sec
[ 4] 3.0- 4.0 sec  296 KBytes  2.43 Mbits/sec
[ 4] 4.0- 5.0 sec  294 KBytes  2.41 Mbits/sec
[ 4] 5.0- 6.0 sec  299 KBytes  2.45 Mbits/sec
[ 4] 6.0- 7.0 sec  294 KBytes  2.41 Mbits/sec
[ 4] 7.0- 8.0 sec  298 KBytes  2.44 Mbits/sec
[ 4] 8.0- 9.0 sec  295 KBytes  2.41 Mbits/sec
[ 4] 9.0-10.0 sec  288 KBytes  2.36 Mbits/sec
[ 4] 10.0-11.0 sec  16.1 KBytes  132 Kbits/sec
[ 4] 11.0-12.0 sec  0.00 Bytes  0.00 bits/sec
[ 4] 0.0-12.0 sec  2.89 MBytes  2.02 Mbits/sec

```

**NOTE:**

When sending data in passthrough mode with the -P option, the socket can only be closed after receiving the SEND\_IDLE event. It takes more than 1 second after sending the last data. So, the remote iperf tcp server stops after 1 second.

## ● Iperf UDP Client

### Normal Send Mode

```

# iperf -c 192.168.200.1 -u

[ IPERF OPTION ]
- role: client
- protocol: udp
- server_port: 5001
- server_ip: 192.168.200.1
- send_time: 10
- send_passthrough: off

```

```
- datagram_size: 1470
- report_interval: 1
```

[ IPERF UDP Client ]

Sending 1470 byte datagrams ...

| Interval       | Transfer      | Bandwidth |
|----------------|---------------|-----------|
| 0.0 ~ 1.0 sec  | 221.07 KBytes | 1.81 Mb/s |
| 1.0 ~ 2.0 sec  | 222.51 KBytes | 1.81 Mb/s |
| 2.0 ~ 3.0 sec  | 223.95 KBytes | 1.82 Mb/s |
| 3.0 ~ 4.0 sec  | 221.07 KBytes | 1.80 Mb/s |
| 4.0 ~ 5.0 sec  | 221.07 KBytes | 1.80 Mb/s |
| 5.0 ~ 6.0 sec  | 222.51 KBytes | 1.81 Mb/s |
| 6.0 ~ 7.0 sec  | 225.38 KBytes | 1.84 Mb/s |
| 7.0 ~ 8.0 sec  | 223.95 KBytes | 1.82 Mb/s |
| 8.0 ~ 9.0 sec  | 223.95 KBytes | 1.83 Mb/s |
| 9.0 ~ 10.0 sec | 215.33 KBytes | 1.82 Mb/s |
| 0.0 ~ 10.0 sec | 2.17 MBytes   | 1.82 Mb/s |

Sent 1547 datagrams

Done

### (Buffered) Passthrough Send Mode

```
# iperf -c 192.168.200.1 -u -P
```

[ IPERF OPTION ]

```
- role: client
- protocol: udp
- server_port: 5001
- server_ip: 192.168.200.1
- send_time: 10
- send_passthrough: on (-)
- datagram_size: 1470
- report_interval: 1
```

[ IPERF UDP Client ]

Sending 1470 byte datagrams ...

| Interval       | Transfer      | Bandwidth |
|----------------|---------------|-----------|
| 0.0 ~ 1.0 sec  | 538.33 KBytes | 4.40 Mb/s |
| 1.0 ~ 2.0 sec  | 525.41 KBytes | 4.28 Mb/s |
| 2.0 ~ 3.0 sec  | 522.54 KBytes | 4.28 Mb/s |
| 3.0 ~ 4.0 sec  | 522.54 KBytes | 4.28 Mb/s |
| 4.0 ~ 5.0 sec  | 528.28 KBytes | 4.30 Mb/s |
| 5.0 ~ 6.0 sec  | 523.97 KBytes | 4.28 Mb/s |
| 6.0 ~ 7.0 sec  | 522.54 KBytes | 4.28 Mb/s |
| 7.0 ~ 8.0 sec  | 522.54 KBytes | 4.27 Mb/s |
| 8.0 ~ 9.0 sec  | 522.54 KBytes | 4.28 Mb/s |
| 9.0 ~ 10.0 sec | 513.93 KBytes | 4.29 Mb/s |

```

0.0 ~ 10.0 sec   5.12 MBytes   4.29 Mbits/sec
Sent 3652 datagrams
Done

```

### Remote Iperf UDP Server

```

pi@raspberrypi:~ $ iperf -s -u -i 1
-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 160 KByte (default)
-----
[ 3] local 192.168.200.1 port 5001 connected with 192.168.200.18 port 50000
[ ID] Interval      Transfer    Bandwidth    Jitter    Lost/Total Datagrams
[ 3] 0.0- 1.0 sec    224 KBytes  1.83 Mbits/sec  0.771 ms    0/ 156 (0%)
[ 3] 1.0- 2.0 sec    221 KBytes  1.81 Mbits/sec  0.749 ms    0/ 154 (0%)
[ 3] 2.0- 3.0 sec    224 KBytes  1.83 Mbits/sec  0.606 ms    0/ 156 (0%)
[ 3] 3.0- 4.0 sec    220 KBytes  1.80 Mbits/sec  0.857 ms    0/ 153 (0%)
[ 3] 4.0- 5.0 sec    220 KBytes  1.80 Mbits/sec  1.427 ms    0/ 153 (0%)
[ 3] 5.0- 6.0 sec    221 KBytes  1.81 Mbits/sec  0.804 ms    0/ 154 (0%)
[ 3] 6.0- 7.0 sec    225 KBytes  1.85 Mbits/sec  0.601 ms    0/ 157 (0%)
[ 3] 7.0- 8.0 sec    223 KBytes  1.82 Mbits/sec  0.821 ms    0/ 155 (0%)
[ 3] 8.0- 9.0 sec    224 KBytes  1.83 Mbits/sec  0.843 ms    0/ 156 (0%)
[ 3] 0.0-10.0 sec    2.17 MBytes  1.82 Mbits/sec  0.699 ms    0/ 1547 (0%)
[ 4] local 192.168.200.1 port 5001 connected with 192.168.200.18 port 50000
[ 4] 0.0- 1.0 sec    524 KBytes  4.29 Mbits/sec  1.878 ms    1/ 366 (0.27%)
[ 4] 1.0- 2.0 sec    521 KBytes  4.27 Mbits/sec  1.967 ms    0/ 363 (0%)
[ 4] 2.0- 3.0 sec    523 KBytes  4.28 Mbits/sec  2.186 ms    0/ 364 (0%)
[ 4] 3.0- 4.0 sec    523 KBytes  4.28 Mbits/sec  2.015 ms    0/ 364 (0%)
[ 4] 4.0- 5.0 sec    525 KBytes  4.30 Mbits/sec  1.932 ms    0/ 366 (0%)
[ 4] 5.0- 6.0 sec    523 KBytes  4.28 Mbits/sec  1.815 ms    0/ 364 (0%)
[ 4] 6.0- 7.0 sec    521 KBytes  4.27 Mbits/sec  1.874 ms    0/ 363 (0%)
[ 4] 7.0- 8.0 sec    523 KBytes  4.28 Mbits/sec  1.835 ms    0/ 364 (0%)
[ 4] 8.0- 9.0 sec    523 KBytes  4.28 Mbits/sec  1.737 ms    0/ 364 (0%)
[ 4] 9.0-10.0 sec    523 KBytes  4.28 Mbits/sec  2.109 ms    0/ 364 (0%)
[ 4] 0.0-10.0 sec    5.12 MBytes  4.28 Mbits/sec  1.798 ms    1/ 3652 (0.027%)

```

### ● Iperf TCP Server

```

# iperf -s

[ IPERF OPTION ]
- role: server
- protocol: tcp
-- ipv4: on

```

```
-- ipv6: on
- server_port: 5001
- report_interval: 1
```

[ IPERF TCP Server ]

Connected with client: 192.168.200.1 port 55360

| Interval       | Transfer      | Bandwidth |
|----------------|---------------|-----------|
| 0.0 ~ 1.0 sec  | 316.30 KBytes | 2.59 Mb/s |
| 1.0 ~ 2.0 sec  | 315.34 KBytes | 2.58 Mb/s |
| 2.0 ~ 3.0 sec  | 317.23 KBytes | 2.60 Mb/s |
| 3.0 ~ 4.0 sec  | 313.44 KBytes | 2.56 Mb/s |
| 4.0 ~ 5.0 sec  | 312.51 KBytes | 2.56 Mb/s |
| 5.0 ~ 6.0 sec  | 314.16 KBytes | 2.57 Mb/s |
| 6.0 ~ 7.0 sec  | 317.44 KBytes | 2.60 Mb/s |
| 7.0 ~ 8.0 sec  | 314.41 KBytes | 2.57 Mb/s |
| 8.0 ~ 9.0 sec  | 314.16 KBytes | 2.57 Mb/s |
| 9.0 ~ 10.2 sec | 365.05 KBytes | 2.59 Mb/s |
| 0.0 ~ 10.2 sec | 3.13 MBytes   | 2.58 Mb/s |

Done

Press ENTER to continue or type any key to stop : q

### Remote Iperf TCP Client

```
pi@raspberrypi:~ $ iperf -c 192.168.200.18 -i 1
```

```
-----
Client connecting to 192.168.200.18, TCP port 5001
TCP window size: 43.8 KByte (default)
-----
```

```
[ 3] local 192.168.200.1 port 55360 connected with 192.168.200.18 port 5001
```

| [ ID] | Interval     | Transfer    | Bandwidth |
|-------|--------------|-------------|-----------|
| [ 3]  | 0.0- 1.0 sec | 384 KBytes  | 3.15 Mb/s |
| [ 3]  | 1.0- 2.0 sec | 384 KBytes  | 3.15 Mb/s |
| [ 3]  | 2.0- 3.0 sec | 256 KBytes  | 2.10 Mb/s |
| [ 3]  | 3.0- 4.0 sec | 384 KBytes  | 3.15 Mb/s |
| [ 3]  | 4.0- 5.0 sec | 256 KBytes  | 2.10 Mb/s |
| [ 3]  | 5.0- 6.0 sec | 256 KBytes  | 2.10 Mb/s |
| [ 3]  | 6.0- 7.0 sec | 384 KBytes  | 3.15 Mb/s |
| [ 3]  | 7.0- 8.0 sec | 256 KBytes  | 2.10 Mb/s |
| [ 3]  | 8.0- 9.0 sec | 384 KBytes  | 3.15 Mb/s |
| [ 3]  | 9.0-10.0 sec | 256 KBytes  | 2.10 Mb/s |
| [ 3]  | 0.0-10.1 sec | 3.12 MBytes | 2.60 Mb/s |



## ● Iperf UDP Server

```
# iperf -s -u

[ IPERF OPTION ]
- role: server
- protocol: udp
-- ipv4: on
-- ipv6: on
- server_port: 5001
- report_interval: 1

[ IPERF UDP Server ]
Connected with client: 192.168.200.1 port 45616
```

| Interval       | Transfer      | Bandwidth | Jitter   | Lost/Total Datagrams |
|----------------|---------------|-----------|----------|----------------------|
| 0.0 ~ 1.0 sec  | 327.30 KBytes | 2.68 Mb/s | 0.903 ms | 0/ 228 (0%)          |
| 1.0 ~ 2.0 sec  | 328.74 KBytes | 2.69 Mb/s | 0.884 ms | 0/ 229 (0%)          |
| 2.0 ~ 3.0 sec  | 328.74 KBytes | 2.69 Mb/s | 0.984 ms | 0/ 229 (0%)          |
| 3.0 ~ 4.0 sec  | 330.18 KBytes | 2.70 Mb/s | 0.887 ms | 0/ 230 (0%)          |
| 4.0 ~ 5.0 sec  | 328.74 KBytes | 2.69 Mb/s | 1.086 ms | 0/ 229 (0%)          |
| 5.0 ~ 6.0 sec  | 330.18 KBytes | 2.70 Mb/s | 0.701 ms | 0/ 230 (0%)          |
| 6.0 ~ 7.0 sec  | 330.18 KBytes | 2.70 Mb/s | 0.913 ms | 0/ 230 (0%)          |
| 7.0 ~ 8.0 sec  | 328.74 KBytes | 2.69 Mb/s | 1.167 ms | 0/ 229 (0%)          |
| 8.0 ~ 9.0 sec  | 328.74 KBytes | 2.69 Mb/s | 0.890 ms | 0/ 229 (0%)          |
| 9.0 ~ 10.0 sec | 335.92 KBytes | 2.73 Mb/s | 0.883 ms | 0/ 234 (0%)          |
| 0.0 ~ 10.0 sec | 3.22 MBytes   | 2.70 Mb/s | 0.883 ms | 0/ 2297 (0%)         |

```
Done: 2297/2297

Press ENTER to continue or type any key to stop : q
```

## Remote Iperf UDP Client

```
pi@raspberrypi:~$ iperf -c 192.168.200.18 -u -b 2.7M -i 1
-----
Client connecting to 192.168.200.18, UDP port 5001
Sending 1470 byte datagrams, IPG target: 4355.56 us (kalman adjust)
UDP buffer size: 160 KByte (default)
-----
[ 3] local 192.168.200.1 port 45616 connected with 192.168.200.18 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec    332 KBytes   2.72 Mb/s
[ 3] 1.0- 2.0 sec    330 KBytes   2.70 Mb/s
[ 3] 2.0- 3.0 sec    329 KBytes   2.69 Mb/s
[ 3] 3.0- 4.0 sec    330 KBytes   2.70 Mb/s
[ 3] 4.0- 5.0 sec    329 KBytes   2.69 Mb/s
[ 3] 5.0- 6.0 sec    330 KBytes   2.70 Mb/s
[ 3] 6.0- 7.0 sec    330 KBytes   2.70 Mb/s
```

```

[ 3] 7.0- 8.0 sec  329 KBytes  2.69 Mbits/sec
[ 3] 8.0- 9.0 sec  330 KBytes  2.70 Mbits/sec
[ 3] 9.0-10.0 sec  329 KBytes  2.69 Mbits/sec
[ 3] 0.0-10.0 sec  3.22 MBytes  2.70 Mbits/sec
[ 3] Sent 2297 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec  3.22 MBytes  2.70 Mbits/sec  0.882 ms  0/ 2297 (0%)

```

## 8.2 Remote Server/Client (raspi-atcmd-remote)

A remote server/client application run one server or client. This application is a Linux application and can be executed on Raspberry Pi.

### 8.2.1 Source files

| File     | Description                             |
|----------|---|
| main.c   | UDP/TCP server/client related functions |
| Makefile | Make file for building                  |

**Table 8.2 raspi-atcmd-remote source files**

### 8.2.2 Build

Copy the “atcmd/host/raspi-atcmd-remote” directory to the Raspberry Pi's home directory. And build the remote application with the make command.

```
$ cd $HOME
```

```
$ cd raspi-atcmd-remote
```

```
$ make clean
```

```
removed 'raspi-atcmd-remote'
```

```
$ make
```

```
cc -g -o raspi-atcmd-remote main.c -Wall -Wno-unused-function -DCONFIG_VERBOSE
```

### 8.2.3 Run

\$ ./raspi-atcmd-remote [-h|--help]

```
raspi-atcmd-remote version 1.2.0
Copyright (c) 2019-2023  <NEWRACOM LTD>

Usage:
$ ./raspi-atcmd-remote -s [-p <listen_port>] [-u] [-e]
$ ./raspi-atcmd-remote -c <server_ip> [-p <server_port>] [-u] [-e]

Options:
-s, --server          run in server mode
-c, --client #        run in client mode
-p, --port #          set server port to listen on or connect to (default: 50000)
-u, --udp             use UDP
-e, --echo            enable echo for received packets (default: off)
-v, --version         print version information and quit
-h, --help           print this message and quit
```

#### Examples:

| Mode   | Protocol | Command   |
|--------|----------|---|
| Server | TCP      | \$ ./raspi-atcmd-remote -s -p 50000 [-e]                  |
|        | UDP      | \$ ./raspi-atcmd-remote -s -u -p 60000 [-e]               |
| Client | TCP      | \$ ./raspi-atcmd-remote -c 192.168.200.1 -p 50000 [-e]    |
|        | UDP      | \$ ./raspi-atcmd-remote -c 192.168.200.1 -u -p 60000 [-e] |

## 9 Revision History

| Revision No | Date       | Comments   |
|-------------|------------|--|
| 1.0         | 03/28/2019 | Initial version for customer release created   |
| 1.1         | 07/02/2019 | Sample Applications updated  |
| 1.2         | 08/01/2019 | HW Flow Control added  |
| 1.3         | 09/17/2019 | Additional AT-commands added   |
| 1.4         | 11/18/2019 | Download binary update & remove description wpa security   |
| 1.5         | 02/14/2020 | Improved command descriptions  |
| 1.6         | 03/25/2020 | SPI connection and CLI application added   |
| 1.7         | 03/31/2020 | AT+STXMODE, AT+SRXMODE, AT+SRXAVAIL and AT+SRECV commands removed  |
| 1.8         | 04/07/2020 | Socket related events removed and added<br>CLI application updated   |
| 1.9         | 05/15/2020 | Ping size parameter removed<br>Test Application added  |
| 1.10        | 05/22/2020 | AT+WDHCPS, AT+WSOFTAP commands added   |
| 1.11        | 06/03/2020 | AT+SLEEP command added   |
| 1.12        | 07/15/2020 | "Chapter 2.2 Building the firmware" added  |
| 1.13        | 08/04/2020 | UART default baudrate changed (38400 -> 115200)<br>"4) Run with script file" in chapter 8.1 added  |
| 1.14        | 08/13/2020 | BSSID in AT+WCONN command added  |
| 1.15        | 08/24/2020 | AT+WROAM command added<br>ROAMING event added  |
| 1.16        | 09/02/2020 | AT+WFOTA command added<br>FOTA event added   |
| 1.17        | 10/08/2020 | In raspi-atcmd-cli application, lperf command supported  |
| 1.18        | 11/24/2020 | FOTA updated <ul style="list-style-type: none"> <li>- New events added</li> <li>- Get-bin-crc.sh removed</li> <li>- Update-fota-info.sh added</li> </ul> |
| 1.19        | 06/15/2021 | AT+WSTAINFO command added  |
| 1.20        | 06/25/2021 | WPA3-OWE/SAE security added  |
| 1.21        | 07/12/2021 | AT+WMCS command removed  |
| 1.21.1      | 07/29/2021 | Some examples fixed  |
| 1.22.0      | 10/21/2021 | AT+SLEEP command removed<br>AT+WSLEEP command added<br>DEEPSLEEP_WAKEUP event added  |
| 1.22.1      | 11/12/2021 | Country code added (AU, NZ)  |
| 1.22.2      | 12/16/2021 | AT+WFOTA command updated <ul style="list-style-type: none"> <li>- fota.json file in JSON format that describes new firmware</li> </ul>                   |

|        |            |   |
|--------|------------|---|
|        |            | <ul style="list-style-type: none"> <li>- bin_name and bin_crc32 parameters to set new firmware</li> <li>- description and example</li> </ul>  |
| 1.22.3 | 02/03/2022 | Added setting form enable serial port<br>Change event name from TCP CONNECT to CONNECT  |
| 1.22.4 | 02/25/2022 | SCAN_DONE event removed<br>ROAMING event removed<br>CONNECT_SUCCESS event changed<br>CONNECT_FAIL event changed<br>DISCONNECT event changed   |
| 1.22.5 | 03/08/2022 | SEND_IDLE event changed<br>SEND_DROP event changed<br>SEND_EXIT event changed<br>SEND_ERROR event changed   |
| 1.22.6 | 03/16/2022 | AT+WMCS command added<br>AT+WTIME command added<br>AT+WDUTYCYCLE command added<br>AT+WCCATHRESHOLD command added<br>AT+WBSSMAXIDLE command added  |
| 1.22.7 | 03/28/2022 | AT+WSCAN SET/GET command added  |
| 1.22.8 | 04/22/2022 | AT+WCONN command updated<br>AT+WTIMEOUT command updated<br>AT+SSEND command updated   |
| 1.22.9 | 08/08/2022 | AT+WDNS command added<br>AT+SOPEN command updated<br>AT+SSEND command updated<br>AT+SRECV command added<br>AT+SRECVMODE command added<br>AT+SRXLOGLEVEL command removed<br>AT+SRECVINFO command added<br>AT+SADDRINFO command added<br>RECV_READY event added |
| 1.23.0 | 01/13/2023 | AT+WDHCP SET/GET command added<br>DHCP related events added<br>Country code "K1" and "K2" added<br>Test Application updated   |
| 1.23.1 | 03/10/2023 | AT+STCPNODELAY command added  |
| 1.23.2 | 03/31/2023 | AT+WROAM command removed<br>CONNECT_FAIL event removed<br>STA_CONNECT/STA_DISCONNECT events added   |
| 1.23.3 | 05/04/2023 | AT+WCCATHRESHOLD command updated <ul style="list-style-type: none"> <li>- CCA threshold range changed</li> </ul>  |
| 1.23.4 | 06/30/2023 | Country code "K0" added   |

|  |  |   |
|--|--|---|
|  |  | AT+WTXPOWER command updated<br>AT+WSOFTAP command updated<br>AT+WSLEEP command removed<br>AT+WDEEPSLEEP command added<br>AT+SOPEN command updated<br>AT+SSEND command updated<br>SEND_ERROR/RCV_ERROR events updated<br>SEND_DONE event added |
|--|--|---|