NRC7292 Evaluation Kit User Guide

(CLI)

Ultra-low power & Long-range Wi-Fi

Ver 1.12
May 30, 2020

NEWRACOM, Inc.

NRC7292 Evaluation Kit User Guide (CLI)
Ultra-low power & Long-range Wi-Fi

Office

Newracom, Inc.
25361 Commercentre Drive, Lake Forest, CA 92630 USA
http://www.newracom.com

Contents

1 Overview 6
1.1 Software structure of CLI application 6
1.2 Build CLI application 7
1.3 Start CLI application 7
2 CLI Commands 9
2.1 show 9
2.2 set 16
2.3 test 19
2.4 gpio 19
2.5 gprf 20
3 Revision History 22

List of Tables

Table 2.1 Category of NRC7292 CLI 9

List of Figures

Figure 1.1 Software structure of CLI application 6
Figure 1.2 Build CLI application 7
Figure 1.3 Execution of CLI application 7
Figure 1.4 “help” CLI command 8
Figure 1.5 Exit of CLI application 8

1 Overview

This document introduces NRC7292 command line interface (CLI) application. The user can utilize the CLI application to check basic information for firmware, monitor channel quality; such as: received signal strength indicator (RSSI), signal to noise ratio (SNR), adjust transmit power, and configure NRC7292 to run in specific operating condition. The source code of this application is offered to user's so users can build an executable file suitable for the host.

1.1 Software structure of CLI application

As shown in Figure 1.1, the CLI application is a user-level application program. The CLI application uses Netlink library to communicate with NRC 11ah driver running on a Linux kernel. The CLI command initiated by the user and goes to the NRC 11ah driver and then to NRC7292 via host serial peripheral interface (HSPI).

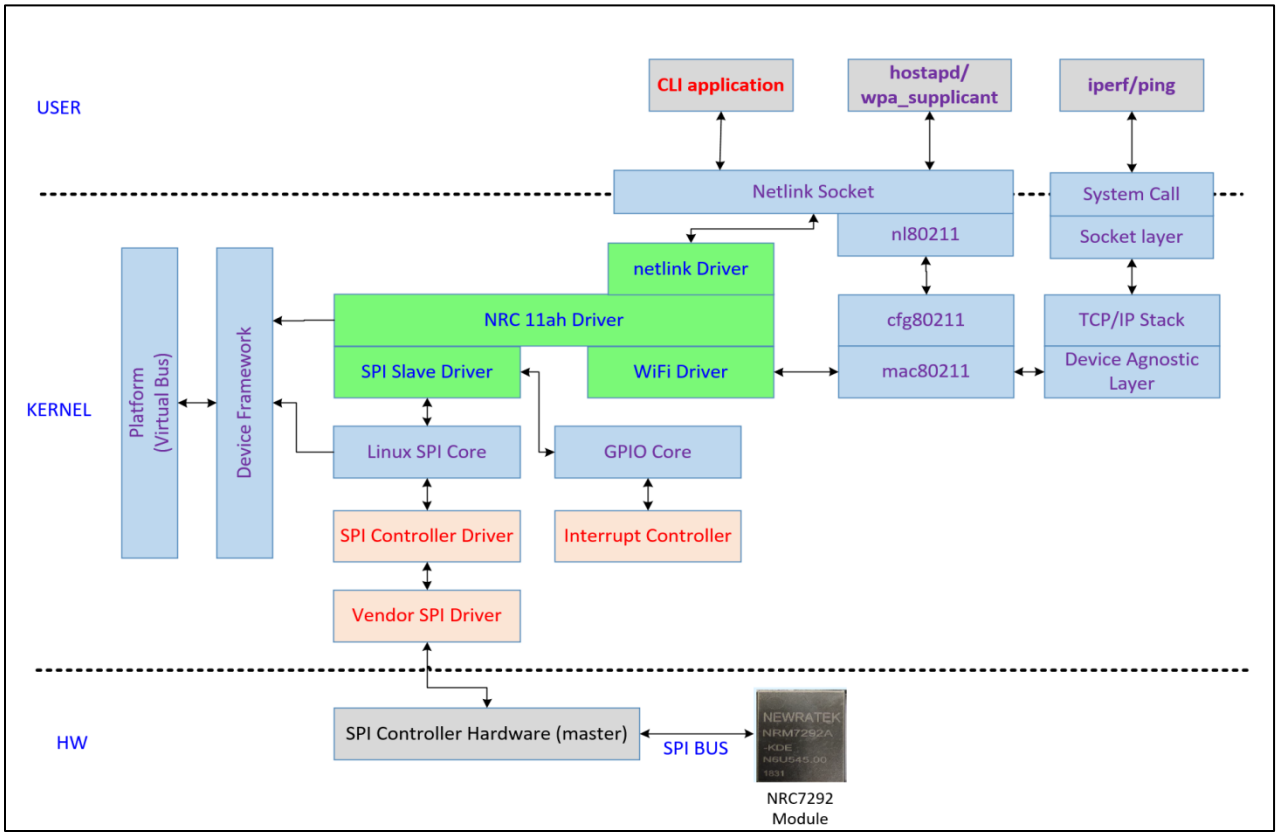

Figure 1.1 Software structure of CLI application

1.2 Build CLI application

A user can build the source code of CLI application with Makefile which is provided along with the source code. Once the user successfully builds it, the user can find "cli_app" executable file in the same directory where the user runs Makefile.

```
pi@raspberrypi:~/cli_app $ make clean; make
rm -f cli_cmd.o cli_netlink.o cli_util.o main.o cli_app
gcc     -c -o cli_cmd.o cli_cmd.c
gcc     -c -o cli_netlink.o cli_netlink.c
gcc     -c -o cli_util.o cli_util.c
gcc     -c -o main.o main.c
gcc cli_cmd.o cli_netlink.o cli_util.o main.o -o cli_app -pthread -lm
```

Figure 1.2 Build CLI application

1.3 Start CLI application

“NRC>” prompt appears if a user executes "cli_app", then the user can enter in a CLI command into the prompt.

```
pi@raspberrypi:~/project/working/NRC_MACSW/host/linux/cli_app $ ./cli_app
=================================================
     Newracom Command Line Application (x.x)
=================================================
NRC>
```

Figure 1.3 Execution of CLI application

"help" is a useful command which displays all the CLI commands with its usages as shown in Figure 1.4.

```
NRC> help
=================================================================================================
 help                                                    :show CLI tree
 read <address> <size in byte>                           :read memory
 exit                                                    :exit program
 show version                                            :show version
 show config [vif_id]                                    :show configuration
 show edca                                               :show EDCA parameters
 show uinfo [vif_id]                                     :show UMAC information
 show ampdu [clear]                                      :show/clear AMPDU count
 show signal [start|stop] [interval] [number]            :show rssi/snr
 show maxagg                                             :show max aggregation
 show duty                                               :show duty cycle
 show autotxgain                                         :show autotxgain
 show cal_use                                            :show cal_use
 show recovery stats                                     :show recovery
 show detection stats                                    :show detection
 show temp                                               :show temp
 show wakeup_pin                                         :show wakeup pin configuration
 show wakeup_source                                      :show wakeup source configuration
 show stats simple_rx                                    :show received packet information
 show mac clear                                          :clear TX/RX Statistics
 show mac tx stats                                       :show TX Statistics
 show mac tx clear                                       :clear TX Statistics
 show mac rx stats                                       :show RX Statistics
 show mac rx clear                                       :clear RX Statistics
 set gi <short|long|auto>                                :set guard interval
 set maxagg <AC(0-3)> <Max(0-13,0:off)> {size:default=0}:set aggregation
 set config <ack[0,1]> <agg[0,1]> <mcs>                  :set ack, aggregation, mcs
 set rc <on|off> [vif_id] [mode]                         :set rate control
 set duty <on|off> {duty window} {tx duration}           :set duty cycle
 set cal_use <on|off>                                    :set cal_use
 set txpwr <value(1~30)>                                 :set txpwrt
 set wakeup_pin {Debounce:on|off} {PIN Number:0~31}      :set wakeup pin for deepsleep
 set wakeup_soruce rtc gpio hspi                         :set wakeup source for deepsleep
 set report {on|off} {time[sec]:default=1}               :lmac Periodic report
 test mcs <mcs value>                                    :set mcs
 test recovery <interval in ms> <count>                  :test recovery
 test assert                                             :test firmware assert
 test ucode                                              :test micro code
 gpio read [pin number]                                  :gpio read
 gpio write [pin number] [0|1]                           :gpio write
 gpio direction [pin index] {[0(input)|1(output)]}       :read/write gpio direction
 gpio pullup [pin index] {[0(off)|1(on)]}                :read/write gpio pullup enable|disable
 gprf read [pin number]                                  :gprf read
 gprf write [pin number] [0|1]                           :gprf write
 gprf direction [pin index] {[0(input)|1(output)]}       :read/write gprf direction
 gprf pullup [pin index] {[0(off)|1(on)]}                :read/write gprf pullup enable|disable
=================================================================================================
OK
```

Figure 1.4 "help" CLI command

To stop the CLI application, a user can use "exit" command.

```
NRC> exit
=================================================
      Exit Newracom Command Line Application
=================================================
pi@raspberrypi:~/cli_app $
```

Figure 1.5 Exit of CLI application

2 CLI Commands

Four categories of CLI commands is defined in Table 2.1. The user can use the phy, set, and test CLI commands only for test purposes.

Table 2.2 Category of NRC7292 CLI

Category	Description
show	display statistics, status, signal, etc.
set	set MAC-layer parameters
test	set test parameters
read	read memory
gpio	write/read gpio, set gpio dirction & pullup
gprf	write/read gpio-rf, set gpio-rf dirction & pullup

2.1 show

2.1.1 show version

Display Firmware version, gerrit/master number and board revision

Parameters

N/A

2.1.2 show config [vif_id]

Display device configurations including device mode, MAC address, frequency, bandwidth, etc.

Parameters

vif_id : interface ID (default 0, vif_id can be 0 or 1 when the concurrent mode is enabled)

2.1.3 show edca

Display EDCA parameters per access category (AC)

Parameters

N/A

2.1.4 show uinfo <vif_id>

Display 11ah capability information

Parameters

vif_id : interface ID (default 0, vif_id can be 0 or 1 when the concurrent mode is enabled)

2.1.5 show ampdu [clear]

Display statistics for aggregated MPDU (AMPDU)

Parameters

clear : clear all statistics

2.1.6 show signal [start|stop] [interval] [number]

Display channel quality information (RSSI and SNR)

Parameters

start : start periodic display with interval (Ex. show signal start 1: display RSSI & SNR every 1 second)
stop : stop displaying
interval : period in second unit (default 1 second)
number : number of samples to display

Returns

```
NRC> show signal
MAC addr: 00:10:40:39:78:62     rssi    :9              snr     :29
MAC addr: 02:00:eb:1a:ce:6b     rssi    :11             snr     :26
OK
```

Total : total number of samples displayed
average : average value of RSSI and SNR
std_dev : standard deviation

```
NRC> show signal start
OK
NRC> 0 [ 00:10:40:39:78:62] rssi:9      snr:29
1 [ 02:00:eb:1a:ce:6b] rssi:11  snr:26

show signal stop
OK
NRC> ----------------------
[Total]   : 1
[RSSI]
 average  : 9.000
 std_dev  : 0.000
[SNR]
 average  : 29.000
 std_dev  : 0.000
----------------------
----------------------
[Total]   : 1
[RSSI]
 average  : 11.000
 std_dev  : 0.000
[SNR]
 average  : 26.000
 std_dev  : 0.000
----------------------
```

2.1.7 show maxagg

Display aggregation status per AC

Parameters

N/A

2.1.8 show stats simple_rx

Display received packet information

Parameters

N/A

Returns

RSSI : received signal strength indication

CS_Cnt : number of carrier sense counted

PSDU_Succ : number of PSDU count successfully received (SIG CRC OK)

MPDU_Rcv : number of MPDU count received

MPUD_Succ : number of MPDU count successfully received (FCS OK)

SNR : signal to noise ratio

※ The PSDU_Succ counts NDP packets. However, the MPDU_Rcv does not count NDP packet. In addition, The PSDU_Suss regards A-MPDU packet as 1 packet.

2.1.9 show mac tx stats

Display MAC-layer TX statistics

Parameters

N/A

Returns

```
NRC> show mac tx stats
-------------------------------------------------------------------------
 MAC TX Statistics (OK count:81, RTX count:11, last MCS:8)
-------------------------------------------------------------------------
- AC[BK]         : OK(          0/          0)  RTX(          0/          0)
- AC[BE]         : OK(         12/       1134)  RTX(         11/       1111)
- AC[VI]         : OK(          0/          0)  RTX(          0/          0)
- AC[VO]         : OK(          0/          0)  RTX(          0/          0)
- AC[BC]         : OK(         69/       2353)  RTX(          0/          0)
- AC[GP]         : OK(          0/          0)  RTX(          0/          0)
-------------------------------------------------------------------------
- TYPE[MGMT]     : OK(          0/          0)  RTX(          0/          0)
- TYPE[CTRL]     : OK(          0/          0)  RTX(          0/          0)
- TYPE[DATA]     : OK(         12/       1134)  RTX(         11/       1111)
- TYPE[BEAC]     : OK(         69/       2353)  RTX(          0/          0)
-------------------------------------------------------------------------
- MCS[ 0]        : OK(          0/          0)  RTX(          0/          0)
- MCS[ 1]        : OK(          0/          0)  RTX(          0/          0)
- MCS[ 2]        : OK(          0/          0)  RTX(          0/          0)
- MCS[ 3]        : OK(          0/          0)  RTX(          0/          0)
- MCS[ 4]        : OK(          0/          0)  RTX(          0/          0)
- MCS[ 5]        : OK(          0/          0)  RTX(          0/          0)
- MCS[ 6]        : OK(          0/          0)  RTX(          5/        505)
- MCS[ 7]        : OK(         10/       1010)  RTX(          5/        505)
- MCS[10]        : OK(         71/       2477)  RTX(          1/        101)
-------------------------------------------------------------------------
OK
```

OK (number of packets successfully transmitted / total aggregated bytes successfully transmitted)

RTX (number of packets retransmitted / total aggregated bytes retransmitted)

※ AC[GP] : GP stands for general purpose. This is used to send a frame which is the highest priority. This is a vendor-specific function.

2.1.10 show mac rx stats

Display MAC-layer RX statistics

Parameters

N/A

Returns

```
NRC> show mac rx stats
-------------------------------------------------------------------------
 MAC RX Statistics (OK count:159, NOK count:1, last MCS:8)
-------------------------------------------------------------------------
- AC[BK]         : OK(          0/          0) NOK(          0/          0)
- AC[BE]         : OK(         12/       1134) NOK(          1/        101)
- AC[VI]         : OK(          0/          0) NOK(          0/          0)
- AC[VO]         : OK(          0/          0) NOK(          0/          0)
- AC[BC]         : OK(        147/       4232) NOK(          0/          0)
- AC[GP]         : OK(          0/          0) NOK(          0/          0)
-----------------  -------------------------  -------------------------
- TYPE[MGMT]     : OK(          0/          0) NOK(          0/          0)
- TYPE[CTRL]     : OK(          0/          0) NOK(          0/          0)
- TYPE[DATA]     : OK(         12/       1134) NOK(          1/        101)
- TYPE[BEAC]     : OK(        147/       4232) NOK(          0/          0)
-----------------  -------------------------  -------------------------
- MCS[ 0]        : OK(          0/          0) NOK(          1/        101)
- MCS[ 1]        : OK(          0/          0) NOK(          0/          0)
- MCS[ 2]        : OK(          0/          0) NOK(          0/          0)
- MCS[ 3]        : OK(          0/          0) NOK(          0/          0)
- MCS[ 4]        : OK(          0/          0) NOK(          0/          0)
- MCS[ 5]        : OK(          3/        304) NOK(          0/          0)
- MCS[ 6]        : OK(          6/        606) NOK(          0/          0)
- MCS[ 7]        : OK(          1/        101) NOK(          0/          0)
- MCS[10]        : OK(        148/       4356) NOK(          0/          0)
-------------------------------------------------------------------------
OK
```

OK (number of packets successfully received / total aggregated bytes successfully received)

NOK (number of packets received but discarded / total aggregated bytes discarded)

※ NOK : Even though the MPDU is successfully received, it can be discarded due to duplication or address mismatch.

2.1.11 show mac tx clear

Clear MAC-layer TX statistics

Parameters

N/A

2.1.12 show mac rx clear

Clear MAC-layer RX statistics

Parameters

N/A

2.1.13 show mac clear

Clear MAC-layer TX and RX statistics

Parameters

N/A

2.1.14 show duty

Show status of duty cycle function

If it is on, it shows duty window, tx duration, and remain tx duration time in usec

Parameters

N/A

Returns

Duty cycle : off

or

Duty cycle : on

Duty window : 60000000

Tx duration : 5000000

Remain tx duration : 5000000

2.1.15 show autotxgain

Show status of autotxgain function

If it is on, it shows Tx power index for each MCS

Parameters

N/A

Auto txgain : off

or

Auto txgain : on

Tx power index for MCS 0 : 22

Tx power index for MCS 1 : 21

Tx power index for MCS 2 : 20

Tx power index for MCS 3 : 19

Tx power index for MCS 4 : 19

Tx power index for MCS 5 : 18

Tx power index for MCS 6 : 17

Tx power index for MCS 7 : 16

Tx power index for MCS 10 : 23

2.1.16 show cal_use

Display RF calibration usage status

Parameters

N/A

2.1.17 show recovery stats

The count statistics of recovery function entered

Parameters

N/A

2.1.18 show detection stats

The count statistics of detection function entered, which are tx triggered

Parameters

N/A

2.1.19 show temp

The temperature of temperature sensor. If temperature sensor is not existed, it display 0.

Parameters

N/A

2.1.20 show wakeup_pin

Get configuration of wakeup gpio pin from deep sleep mode

Parameters

N/A

2.1.21 show wakeup_source

Get configuration of wakeup source from deep sleep mode

Parameters

N/A

2.2 set

2.2.1 set rc <on|off> [vif_id]

rate control on / off

Parameters

on : enable rate control (automatic selection of MCS based on the link condition) (default)

off : disable rate control (user can select MCS manually by using “test mcs” command)

2.2.2 set gi <short|long|auto>

Set Guard Interval

Parameters

short : Short Guard Interval

long : Long Guard Interval (default)

auto : Change the guard interval from long to short automatically in MCS7

2.2.3 set maxagg <ac> <max num> {size(optional, 0: default)}

Set maximum number of MPDU of AC in AMDPU

Parameters

ac : access category

max num : number of MPDU in AMPDU (1~13, 0: off)

size : aggregation threshold size (default : 0)

2.2.4 set config <ack> <agg> <mcs>

Set MAC-layer configuration

Parameters

ack : ack mode (0: no ack, 1: ndp ack, 2: normal ack) (default: ndp ack)

agg : aggregation (AMPDU) (0: off, 1:on)

mcs: 0~7, 10 (10 can be used only in 1MHz bandwidth)

2.2.5 set duty <on|off> {duty window in usec} {tx duration in usec}

Set duty cycle

Parameters

on|off : enable|disable duty cycle function (default; off)

duty window : specify duty cycle window in usec. (default: 60 sec)

tx duration : specify allowed tx duration within duty cycle window in usec. (default: 5 sec)

(ex) duty window 10000000 (10sec) and tx duration 1000000 (1sec) will be set to access the channel 1 second during every 10 seconds.

2.2.6 set cal_use <on|off>

Set RF calibration usage on|off, if avaliable RF calibration is existed.

Parameters

on|off : enable|disable RF calibration usage

2.2.7 set txpwr <value>

Set tx power

Parameters

value : 1~30

2.2.8 set wakeup_pin <debounce> <pin index>

Configure a wakeup-gpio-pin when system state is uCode or deepsleep

Parameters

debounce : on|off

pin index : 0~31

2.2.9 set wakeup_source <wakeup_sources>

Configure wakeup sources when system state is deepsleep.

Parameters

wakeup_sources : rtc gpio hspi

※ It is possible to assign multiple sources (ex) set wakeup_source rtc gpio

2.3 test

2.3.1 test mcs <value>

Set MCS

Parameters

value : 0~7, 10 (10 can be used only in 1MHz bandwidth)

2.3.2 test recovery <interval> <count>

runs the recovery function [count] times every [interval in ms]

Parameters

interval : period of recovery function entered (it should be over than 1000. unit is ms)

count : total number

2.3.3 test assert

runs the assert function

Parameters

N/A

2.4 gpio

The GPIO CLI commands listed below can be used to configure GPIO options to read from and write to GPIO pins. Note that GPIO command usage is valid for non-dedicated GPIO pins only. Refer to the NRC7292 Technical Reference Manual for the list of dedicated GPIO pins.

2.4.1 gpio direction <pin index> {direction}

Get or set the direction (0: input / 1: output) of a GPIO pin.

Parameters

pin index: GPIO pin index

(For set command only) direction: 0: input / 1: output

2.4.2 gpio pullup <pin index> {pull-up option}

Get or set the pull-up option (0: disable / 1: enable) for a GPIO pin.

Parameters

pin index: GPIO pin index

(For set command only) pull-up option: 0: disable / 1: enable

2.4.3 gpio read <pin index>

Read the value (0: low / 1: high) from a GPIO pin.

Parameters

pin index: GPIO pin index

2.4.4 gpio write <pin index> <value>

Write a value (0: low / 1: high) to a GPIO pin.

Parameters

pin index: GPIO pin index

value: 0: low / 1: high

2.5 gprf

The GPIO-RF CLI commands listed below can be used to configure GPIO-RF options to read from and write to GPIO-RF pins. Note that GPIO-RF command usage is valid for non-dedicated GPIO-RF pins only. Refer to the NRC7292 Technical Reference Manual for the list of dedicated GPIO-RF pins.

2.5.1 gprf direction <pin index> {direction}

Get or set the direction (0: input / 1: output) of a GPIO-RF pin.

Parameters

pin index: GPIO-RF pin index

(For set command only) direction: 0: input / 1: output

2.5.2 gprf pullup <pin index> {pull-up option}

Get or set the pull-up option (0: disable / 1: enable) for a GPIO-RF pin.

Parameters

pin index: GPIO-RF pin index

(For set command only) pull-up option: 0: disable / 1: enable

2.5.3 gprf read <pin index>

Read the value (0: low / 1: high) from a GPIO-RF pin.

Parameters

pin index: GPIO-RF pin index

2.5.4 gprf write <pin index> <value>

Write a value (0: low / 1: high) to a GPIO-RF pin.

Parameters

pin index: GPIO-RF pin index

value: 0: low / 1: high

3 Revision History

Revision No	Date	Comments
Ver 1.0	02/26/2019	Initial version for customer release created
Ver 1.1	03/25/2019	Add CLI commands (2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.1.6, 0, 0, 0, 0, 2.1.12, 2.1.13,2.2.1, 2.2.2) Removed CLI commands (“show rx <start | stop> <duration>”, “phy txpower <value>”)
Ver 1.2	10/03/2019	CLI commands description updated
Ver 1.3	11/01/2019	Add set duty, show duty, set autotxgain, show autotxgain commands
Ver 1.4	11/18/2019	Add response example about show siganl start/stop
Ver 1.5	12/05/2019	Add response example about set cal_use & show cal_use
Ver 1.6	12/13/2019	Add ‘auto’ option for “set gi <short|long|auto>” command
Ver 1.7	12/19/2019	Add “test recovery <interval> <count>”, “show recovery stats”, “show detection stats” command Add parameter for “set maxagg {ac} {maxagg} {size(optional, 0: default)}” and “set txpwr [val]”
Ver 1.8	12/27/2019	Add ‘test assert’ option for recovery testing
Ver 1.9	01/17/2020	Add gpio, gprf commands
Ver 1.10	04/13/2020	Remove a ‘set autotxgain’ command
Ver 1.11	04/22/2020	Add ‘show temp’
Ver 1.12	05/30/2020	Add ‘set/show wakeup_pin’, ‘set/show wakeup_source’