NRC7292 Evaluation Kit User Guide

(Transmit Power Control)

Ultra-low power & Long-range Wi-Fi

Ver 1.0
Aug 21, 2020

NEWRACOM, Inc.

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

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

Contents

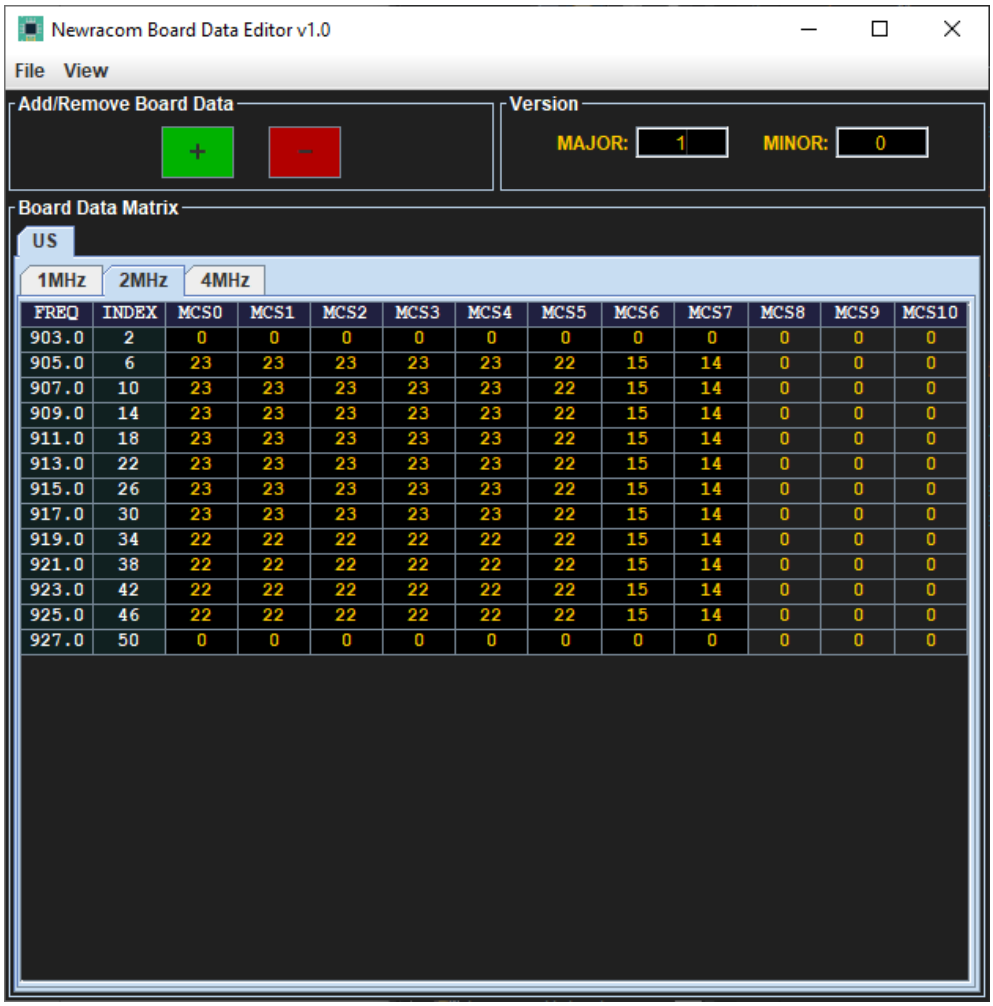
List of Tables

List of Figures

1 Overview

The Transmit Power Control (TPC) mechanism dynamically changes the transmission power level as a function of the country code, operating frequency, bandwidth and MCS. The corresponding mapping information is encoded as a byte array. The mechanism may be useful for controlling the maximum transmission power level to meet the regional regulatory requirements, reducing the power consumption level during operation, and adjusting the communication range between access points and stations.

The board data file which encodes the mapping information can be created, viewed, or edited using the board data editor. See the user guide 'TL-7292-009-Board_Data_Editor.pdf' for more information.



FREQ	INDEX	MCS0	MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9	MCS10
903.0	2	0	0	0	0	0	0	0	0	0	0	0
905.0	6	23	23	23	23	23	22	15	14	0	0	0
907.0	10	23	23	23	23	23	22	15	14	0	0	0
909.0	14	23	23	23	23	23	22	15	14	0	0	0
911.0	18	23	23	23	23	23	22	15	14	0	0	0
913.0	22	23	23	23	23	23	22	15	14	0	0	0
915.0	26	23	23	23	23	23	22	15	14	0	0	0
917.0	30	23	23	23	23	23	22	15	14	0	0	0
919.0	34	22	22	22	22	22	22	15	14	0	0	0
921.0	38	22	22	22	22	22	22	15	14	0	0	0
923.0	42	22	22	22	22	22	22	15	14	0	0	0
925.0	46	22	22	22	22	22	22	15	14	0	0	0
927.0	50	0	0	0	0	0	0	0	0	0	0	0

Figure 1.1 Newracom Board Data Editor

2 Board Data Binary Structure

2.1 Overview

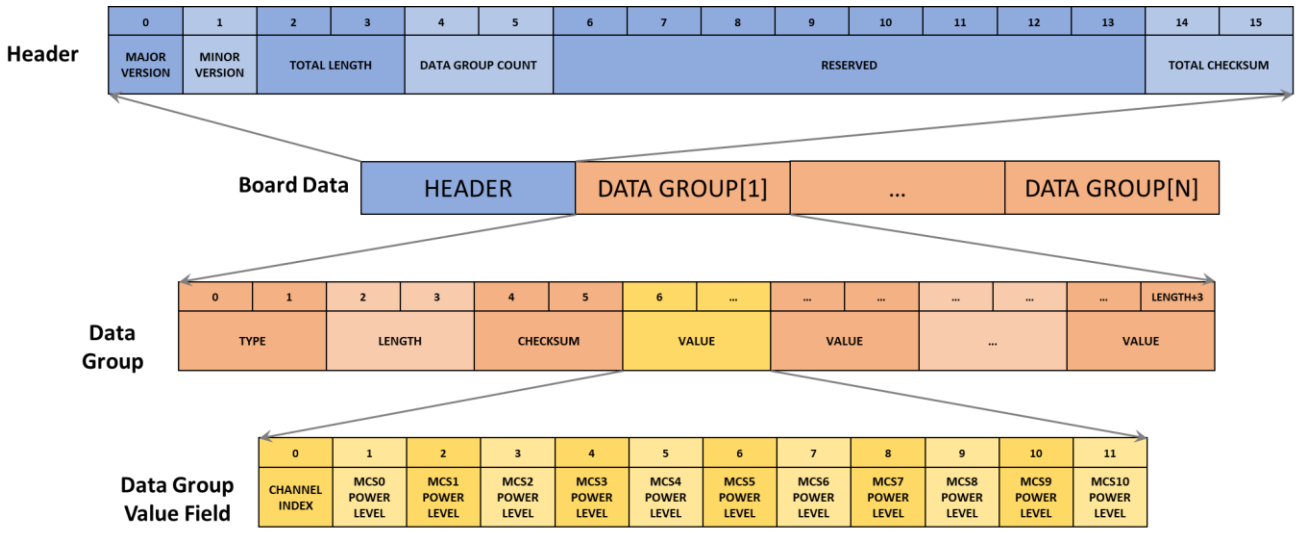


Figure 2.1 Board Data Format

The board data is a byte array consisting of a header segment followed by one or more data group segments, where each data group corresponds to the mapping information for a single country code.

2.2 Header Segment

The header segment consists of 16 bytes:

Table 2.1 Header segment

Byte Offset	Item	Description
0	Major Version	Board data version. (Major)
1	Minor Version	Board data version. (Minor)
2	Total Length	The sum of the byte lengths of all data groups. In other words, it is the byte length of the board data file minus the header size.
3		
4	Data Group Count	The number of data groups.
5		
6	Reserved	Reserved bytes. Must be filled with 0s.
7		
8		
9		
10		
11		
12		
13		
14	Total Checksum	The 2-byte checksum over all data groups. The checksum computation method is not available to the user.
15		

- The total length, data group count and checksum bytes use the little-endian byte order.

2.3 Data Group Segments

Each data group segment is given in a TLV (Type-Length-Value) format.

Table 2.2 Data group segment

<table>
<tr><th>Byte Offset</th><th>Item</th><th>Description</th></tr>
<tr><td>0</td><td rowspan="2">Type</td><td rowspan="2">The country code corresponding to the data group segment:
(US:1,JP:2,KR:3,TW:4,EU:5,CN:6)</td></tr>
<tr><td>1</td></tr>
<tr><td>2</td><td rowspan="2">Total Length</td><td rowspan="2">The sum of the byte lengths of the checksum and all value fields:
(US:542,JP:134,KR:206[Default] or 98[KR_MIC],TW:266,EU:86,CN:362)</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td><td rowspan="2">Checksum</td><td rowspan="2">The 2-byte checksum over all value fields. The checksum computation method is not available to the user.</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td><td rowspan="4">Value Field 1</td><td rowspan="13">Value fields are sequenced in ascending order of bandwidth and channel index. Each value field consists of 12 bytes.
<table>
<tr><th>Byte Offset</th><th>Item</th><th>Comments</th></tr>
<tr><td>0</td><td>Channel Index</td><td>S1G channel index</td></tr>
<tr><td>1</td><td>Power Level (MCS0)</td><td rowspan="12">All power levels are in dBm. The valid power ranges from 1 to 30, and 0 is used to mark unsupported settings. MCS10 for bandwidths 2/4MHz and MCS 8/9 for all bandwidths are not supported so their values are always set to 0.</td></tr>
<tr><td>2</td><td>Power Level (MCS1)</td></tr>
<tr><td>3</td><td>Power Level (MCS2)</td></tr>
<tr><td>4</td><td>Power Level (MCS3)</td></tr>
<tr><td>5</td><td>Power Level (MCS4)</td></tr>
<tr><td>6</td><td>Power Level (MCS5)</td></tr>
<tr><td>7</td><td>Power Level (MCS6)</td></tr>
<tr><td>8</td><td>Power Level (MCS7)</td></tr>
<tr><td>9</td><td>Power Level (MCS8)</td></tr>
<tr><td>10</td><td>Power Level (MCS9)</td></tr>
<tr><td>11</td><td>Power Level (MCS10)</td></tr>
</table></td></tr>
<tr><td>7</td></tr>
<tr><td>…</td></tr>
<tr><td>18</td></tr>
<tr><td>19</td><td rowspan="4">Value Field 2</td></tr>
<tr><td>20</td></tr>
<tr><td>…</td></tr>
<tr><td>31</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>12(K-1) + 1</td><td rowspan="4">Value Field K</td></tr>
<tr><td>12(K-1) + 2</td></tr>
<tr><td>…</td></tr>
<tr><td>12K + 6</td></tr>
</table>

- The type, total length and checksum bytes use the little-endian byte order.

3 Board Data Example

Table 3.1 Board Data Example (562 bytes) – Byte Table

Offset	Values															
0x0000	01	00	22	02	01	00	00	00	00	00	00	00	00	00	2A	FC
0x0010	01	00	1E	02	05	FD	01	00	00	00	00	00	00	00	00	00
0x0020	00	00	03	17	17	17	17	17	17	12	0E	00	00	17	05	17
0x0030	17	17	17	17	17	12	0E	00	00	17	07	17	17	17	17	17
0x0040	17	12	0E	00	00	17	09	17	17	17	17	17	17	12	0E	00
0x0050	00	17	0B	17	17	17	17	17	17	12	0E	00	00	17	0D	17
0x0060	17	17	17	17	17	12	0E	00	00	17	0F	17	17	17	17	17
0x0070	17	12	0E	00	00	17	11	17	17	17	17	17	17	12	0E	00
0x0080	00	17	13	17	17	17	17	17	17	12	0E	00	00	17	15	17
0x0090	17	17	17	17	17	12	0E	00	00	17	17	17	17	17	17	17
0x00A0	17	12	0E	00	00	17	19	17	17	17	17	17	17	12	0E	00
0x00B0	00	17	1B	17	17	17	17	17	17	12	0E	00	00	17	1D	16
0x00C0	16	16	16	16	16	12	0E	00	00	16	1F	16	16	16	16	16
0x00D0	16	12	0E	00	00	16	21	16	16	16	16	16	16	12	0E	00
0x00E0	00	16	23	16	16	16	16	16	16	12	0E	00	00	16	25	16
0x00F0	16	16	16	16	16	12	0E	00	00	16	27	16	16	16	16	16
0x0100	16	12	0E	00	00	16	29	16	16	16	16	16	16	12	0E	00
0x0110	00	16	2B	16	16	16	16	16	16	12	0E	00	00	16	2D	16
0x0120	16	16	16	16	16	12	0E	00	00	16	2F	16	16	16	16	16
0x0130	16	12	0E	00	00	16	31	16	16	16	16	16	16	12	0E	00
0x0140	00	16	33	00	00	00	00	00	00	00	00	00	00	00	02	00
0x0150	00	00	00	00	00	00	00	00	00	00	06	17	17	17	17	17
0x0160	16	0F	0E	00	00	00	0A	17	17	17	17	17	16	0F	0E	00
0x0170	00	00	0E	17	17	17	17	17	16	0F	0E	00	00	00	12	17
0x0180	17	17	17	17	16	0F	0E	00	00	00	16	17	17	17	17	17
0x0190	16	0F	0E	00	00	00	1A	17	17	17	17	17	16	0F	0E	00
0x01A0	00	00	1E	17	17	17	17	17	16	0F	0E	00	00	00	22	16
0x01B0	16	16	16	16	16	0F	0E	00	00	00	26	16	16	16	16	16
0x01C0	16	0F	0E	00	00	00	2A	16	16	16	16	16	16	0F	0E	00
0x01D0	00	00	2E	16	16	16	16	16	16	0F	0E	00	00	00	32	00
0x01E0	00	00	00	00	00	00	00	00	00	00	08	00	00	00	00	00
0x01F0	00	00	00	00	00	00	10	15	15	15	15	15	15	0F	0E	00
0x0200	00	00	18	15	15	15	15	15	15	0F	0E	00	00	00	20	15
0x0210	15	15	15	15	15	0F	0E	00	00	00	28	15	15	15	15	15
0x0220	15	0F	0E	00	00	00	30	00	00	00	00	00	00	00	00	00
0x0230	00	00														

3.1 Example Board Data - Header Segment

The header segment corresponds to the region marked in blue in the byte table.

Table 3.2 Board Data Example - Header Segment

Byte Offset	Item	Value (HEX)	Value (DEC)
0	Major Version	0x01	1
1	Minor Version	0x00	0
2	Total Length	0x0222	546 (562 [Board Data Size]– 16 [Header Size])
3			
4	Data Group Count	0x0001	1
5			
6	Reserved	0x0	0
7			
8			
9			
10			
11			
12			
13			
14	Total Checksum	0xFC2A	64554
15			

3.2 Board Data Example – Data Group Segment

As indicated by the header segment, the board data example contains a single data group segment.

3.2.1 Data group header

The data group header segment corresponds to the region marked in blue in the byte table.

Table 3.3 Board Data Example - Header Segment

Byte Offset	Item	Value (HEX)	Value (DEC)
0	Type	0x0001	1 [Country Code: US]
1			
2	Length	0x021E	542 (2 [Checksum Size] + 12 x 45 [Value Field Count])
3			
4	Checksum	0xFD05	64773
5			

3.2.2 Data group values fields

The body segment containing the value fields following the data group header segment can be divided into three parts: 1MHz, 2MHz and 4MHz, each corresponding to the region marked in green, yellow and red, respectively.

Table 3.4 Board Data Example – 1MHz Data

Frequency [MHz]	Index	Power levels [dBm] for each MCS										
		M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
902.5	**1**	0	0	0	0	0	0	0	0	0	0	0
903.5	**3**	23	23	23	23	23	23	18	14	0	0	23
904.5	**5**	23	23	23	23	23	23	18	14	0	0	23
905.5	**7**	23	23	23	23	23	23	18	14	0	0	23
906.5	**9**	23	23	23	23	23	23	18	14	0	0	23
907.5	**11**	23	23	23	23	23	23	18	14	0	0	23
908.5	**13**	23	23	23	23	23	23	18	14	0	0	23
909.5	**15**	23	23	23	23	23	23	18	14	0	0	23
910.5	**17**	23	23	23	23	23	23	18	14	0	0	23
911.5	**19**	23	23	23	23	23	23	18	14	0	0	23
912.5	**21**	23	23	23	23	23	23	18	14	0	0	23
913.5	**23**	23	23	23	23	23	23	18	14	0	0	23
914.5	**25**	23	23	23	23	23	23	18	14	0	0	23
915.5	**27**	23	23	23	23	23	23	18	14	0	0	23
916.5	**29**	22	22	22	22	22	22	18	14	0	0	22
917.5	**31**	22	22	22	22	22	22	18	14	0	0	22
918.5	**33**	22	22	22	22	22	22	18	14	0	0	22
919.5	**35**	22	22	22	22	22	22	18	14	0	0	22
920.5	**37**	22	22	22	22	22	22	18	14	0	0	22
921.5	**39**	22	22	22	22	22	22	18	14	0	0	22
922.5	**41**	22	22	22	22	22	22	18	14	0	0	22
923.5	**43**	22	22	22	22	22	22	18	14	0	0	22
924.5	**45**	22	22	22	22	22	22	18	14	0	0	22
925.5	**47**	22	22	22	22	22	22	18	14	0	0	22
926.5	**49**	22	22	22	22	22	22	18	14	0	0	22
927.5	**51**	0	0	0	0	0	0	0	0	0	0	0

Table 3.5 Board Data Example – 2MHz Data

Frequency [MHz]	Index	Power levels [dBm] by MCS										
		M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
903.0	**2**	0	0	0	0	0	0	0	0	0	0	0
905.0	**6**	23	23	23	23	23	22	15	14	0	0	0
907.0	**10**	23	23	23	23	23	22	15	14	0	0	0
909.0	**14**	23	23	23	23	23	22	15	14	0	0	0
911.0	**18**	23	23	23	23	23	22	15	14	0	0	0
913.0	**22**	23	23	23	23	23	22	15	14	0	0	0
915.0	**26**	23	23	23	23	23	22	15	14	0	0	0
917.0	**30**	23	23	23	23	23	22	15	14	0	0	0
919.0	**34**	22	22	22	22	22	22	15	14	0	0	0
921.0	**38**	22	22	22	22	22	22	15	14	0	0	0
923.0	**42**	22	22	22	22	22	22	15	14	0	0	0
925.0	**46**	22	22	22	22	22	22	15	14	0	0	0
927.0	**50**	0	0	0	0	0	0	0	0	0	0	0

Table 3.6 Board Data Example – 4MHz Data

Frequency [MHz]	Index	Power levels [dBm] by MCS										
		M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
906.0	**8**	0	0	0	0	0	0	0	0	0	0	0
910.0	**16**	21	21	21	21	21	21	15	14	0	0	0
914.0	**24**	21	21	21	21	21	21	15	14	0	0	0
918.0	**32**	21	21	21	21	21	21	15	14	0	0	0
922.0	**40**	21	21	21	21	21	21	15	14	0	0	0
926.0	**48**	0	0	0	0	0	0	0	0	0	0	0

4 SDK Package TPC Usage Configuration for Host Mode

The body segment containing the value fields following the data group header segment can be divided into three parts

For host mode operation, the TPC parameters ('bd_download' and 'bd_name') in the 'start.py' script file must be modified to enable or disable TPC usage. Setting the value of the parameter 'bd_download' to 1 will enable the TPC usage and setting it to 0 will disable the TPC usage. The value of the parameter 'bd_name' must be set to the name of the board data file. The specified board data file must be present in the directory: 'nrc_pkg/sw/firmware'. The board data file will be copied to the directory 'lib/firmware' upon executing the script. After executing the script, the user can check whether or not the TPC usage is enabled by typing './cli_app show autotxgain' inside the same directory.

```
################################################################################
# Default Configuration (you can change value you want here)
model        = 7292      # 7292 or 7192
hif_speed    = 16000000  # HSPI Clock
gain_type    = 'phy'     # 'phy' or 'nrf(legacy)'
txpwr_val    = 17        # TX Power
maxagg_num   = 8         # 0(AMPDU off) or  >2(AMPDU on)
cqm_off      = 0         # 0(CQM on) or 1(CQM off)
fw_download  = 1         # 0(FW Download off) or 1(FW Download on)
fw_name      = 'uni_s1g.bin'
bd_download  = 0         # 0(Board Data Download off) or 1(Board Data Download on)
bd_name      = 'nrc7292_bd.dat'
guard_int    = 'long'    # 'long (LGI) or 'short' (SGI)
supplicant_debug = 0     # WPA Supplicant debug option : 0(off) or 1(on)
hostapd_debug    = 0     # Hostapd debug option    : 0(off) or 1(on)
max_cpuclock     = 1     # RPi Max CPU Clock : 0(off) or 1(on)
relay_type       = 0     # 0 (wlan0: STA, wlan1: AP) 1 (wlan0: AP, wlan1: STA)
power_save       = 0     # power save : 0(off) or 1(on)
################################################################################
```

Figure 4.1 TPC parameters in 'start.py' (host mode)

5 Revision History

Revision No	Date	Comments
Ver 1.0	08/21/2020	First version