Application Note

The Diagnostics Channel
Protocol
Model p400

Revision: **1.07**

Microhard Systems Inc.
#17, 2135 - 32nd Ave. N.E.
Calgary, Alberta T2E 6Z3
Phone: (403) 248-0028
Fax: (403) 248-2762
www.microhardcorp.com

Warranty

Microhard Systems Inc. warrants that each product will be free of defects in material and workmanship for a period of one (1) year for its products. The warranty commences on the date the product is shipped by Microhard Systems Inc. Microhard Systems Inc.'s sole liability and responsibility under this warranty is to repair or replace any product which is returned to it by the Buyer and which Microhard Systems Inc. determines does not conform to the warranty. Product returned to Microhard Systems Inc. for warranty service will be shipped to Microhard Systems Inc. at Buyer's expense and will be returned to Buyer at Microhard Systems Inc.'s expense. In no event shall Microhard Systems Inc. be responsible under this warranty for any defect which is caused by negligence, misuse or mistreatment of a product or for any unit which has been altered or modified in any way. The warranty of replacement shall terminate with the warranty of the product.

Warranty Disclaims

Microhard Systems Inc. makes no warranties of any nature of kind, expressed or implied, with respect to the hardware, software, and/or products and hereby disclaims any and all such warranties, including but not limited to warranty of non-infringement, implied warranties of merchantability for a particular purpose, any interruption or loss of the hardware, software, and/or product, any delay in providing the hardware, software, and/or product or correcting any defect in the hardware, software, and/or product, or any other warranty. The Purchaser represents and warrants that Microhard Systems Inc. has not made any such warranties to the Purchaser or its agents MICROHARD SYSTEMS INC. EXPRESS WARRANTY TO BUYER CONSTITUTES MICROHARD SYSTEMS INC. SOLE LIABILITY AND THE BUYER'S SOLE REMEDIES. EXCEPT AS THUS PROVIDED, MICROHARD SYSTEMS INC. DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PROMISE.

MICROHARD SYSTEMS INC. PRODUCTS ARE NOT DESIGNED OR INTENDED TO BE USED IN ANY LIFE SUPPORT RELATED DEVICE OR SYSTEM RELATED FUNCTIONS NOR AS PART OF ANY OTHER CRITICAL SYSTEM AND ARE GRANTED NO FUNCTIONAL WARRANTY.

Indemnification

The Purchaser shall indemnify Microhard Systems Inc. and its respective directors, officers, employees, successors and assigns including any subsidiaries, related corporations, or affiliates, shall be released and discharged from any and all manner of action, causes of action, liability, losses, damages, suits, dues, sums of money, expenses (including legal fees), general damages, special damages, including without limitation, claims for personal injuries, death or property damage related to the products sold hereunder, costs and demands of every and any kind and nature whatsoever at law.

IN NO EVENT WILL MICROHARD SYSTEMS INC. BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL, BUSINESS INTERRUPTION, CATASTROPHIC, PUNITIVE OR OTHER DAMAGES WHICH MAY BE CLAIMED TO ARISE IN CONNECTION WITH THE HARDWARE, REGARDLESS OF THE LEGAL THEORY BEHIND SUCH CLAIMS, WHETHER IN TORT, CONTRACT OR UNDER ANY APPLICABLE STATUTORY OR REGULATORY LAWS, RULES, REGULATIONS, EXECUTIVE OR ADMINISTRATIVE ORDERS OR DECLARATIONS OR OTHERWISE, EVEN IF MICROHARD SYSTEMS INC. HAS BEEN ADVISED OR OTHERWISE HAS KNOWLEDGE OF THE POSSIBILITY OF SUCH DAMAGES AND TAKES NO ACTION TO PREVENT OR MINIMIZE SUCH DAMAGES. IN THE EVENT THAT REGARDLESS OF THE WARRANTY DISCLAIMERS AND HOLD HARMLESS PROVISIONS INCLUDED ABOVE MICROHARD SYSTEMS INC. IS SOMEHOW HELD LIABLE OR RESPONSIBLE FOR ANY DAMAGE OR INJURY, MICROHARD SYSTEMS INC.'S LIABILITY FOR ANYDAMAGES SHALL NOT EXCEED THE PROFIT REALIZED BY MICROHARD SYSTEMS INC. ON THE SALE OR PROVISION OF THE HARDWARE TO THE CUSTOMER.

Proprietary Rights

The Buyer hereby acknowledges that Microhard Systems Inc. has a proprietary interest and intellectual property rights in the Hardware, Software and/or Products. The Purchaser shall not (i) remove any copyright, trade secret, trademark or other evidence of Microhard Systems Inc.'s ownership or proprietary interest or confidentiality other proprietary notices contained on, or in, the Hardware, Software or Products, (ii) reproduce or modify any Hardware, Software or Products or make any copies thereof, (iii) reverse assemble, reverse engineer or decompile any Software or copy thereof in whole or in part, (iv) sell, transfer or otherwise make available to others the Hardware, Software, or Products or documentation thereof or any copy thereof, except in accordance with this Agreement.

1 Overview

This application note describes how to use the Diagnostics channel for on-line diagnostics and set up. The Diagnostics protocol can be used for p400 modems.

2 Diagnostics Protocol and Structure of Diagnostics Commands.

Diagnostic data on COM2 is exchanged in packets at 115.2 kbps rate and data format 8N1.

If user's password is not blank, in order to use on-line Diagnostics, user must be logged in using either the AT-command menu or special Diagnostics function.

Packets originated by the user are requests; packets originated by the modem are responses.

There must be a time gap of at least 20 ms between the requests. The packet content is protected by 16 bit CRC.

Requests can be unicast and broadcast. Address 0 is reserved for local access. Responses always have the real address of the modem, including responses to requests with the local address.

There is a 16 bit magic number which is user specific and returned unchanged with the response. It can be used by the user as a sequence number.

There are three types of requests:
-parameter reads,
-parameter writes and
-special functions.

Each type has its unique command IDs followed by optional parameters.

Every diagnostics request is acknowledged by either sending data back or by other, function specific forms of acknowledgement. A bit in control byte allows to suppress the acknowledgement. This is useful when a broadcast command is sent to all modems and no ACK is needed.

Read and write commands have a bit in the control byte to report an error in request processing.

IMPORTANT: There is little error checking on the value of parameters that the user can write to the diagnostics. This is a low level interface to the modem. All range checking and validation of parameters must be done by the user before sending data to the modem.

All diagnostics packets have the same structure (Figure 1).

Byte 1 Size (Excluding this byte)	Bytes 2-7 MAC address (high byte first)	Bytes 8, 9 Magic number	Byte 10 Control byte	Byte 11 and following Payload	…	Byte Last-1 CRC(H)	Byte Last CRC(L)

a) Diagnostics request

Byte 1 Size (Excluding this byte)	Bytes 2-7 MAC address (high byte first)	Bytes 8, 9 Magic number	Byte 10 Status byte	Byte 11 and following Payload	…	Byte Last-1 CRC(H)	Byte Last CRC(L)

b) Diagnostics response

Figure 1. Diagnostics request and Diagnostics response.

For example to read COM1 baud rate locally the following packet must be sent to COM2

Request:

1	**2**	**3**	**4**	**5**	**6**
0x0b	0x00 0x00 0x00 0x00 0x00 0x00	0x12 0x34	0x02	0x14 0x02	0xeb 0x18

Response:

1	**2**	**3**	**4**	**5**	6
0x0C	0x00 0x00 0x00 0x00 0x00 0x02	0x12 0x34	0x00	0x14 0x02 0x01	0xF5 0x03

Request

1) Size
It is the size of the diagnostics packet EXCLUDING the size field itself and two bytes of CRC.
2) Six bytes of MAC address: local access MAC address 0x000000000000.
3) User defined magic number: 0x12, 0x34.
4) Control byte: 0x02;
5) Two bytes of payload:
0x14 - command ID to read parameters and
0x02 - parameter ID for COM1 baud rate.

6) Two bytes of CRC: 0xeb18 - CRC is not included in the size

A) The minimum size value is 10 bytes: (6 bytes for address, 2 for magic number, one for control and one for command ID). The max size is 254 bytes.
B) MAC address: Contains 6 bytes of MAC address of the destination modem for this packet. The modem p400 uses a 2-byte unit address (S105), the rest bytes must be zeros.
Local address is all zeroes. The MAC address is sent high byte first.

C) Control byte:
Bit 0 - reserved, should be left cleared.
Bit 1 - ACK needed. When set, the modem that gets the packet will acknowledge it. ACK bit works for broadcasts too and should normally be cleared.
Bit 2 - reserved, must be cleared
Bit 3 - if set in response - the modem detected an error in the request
Bit 4 - if set in response - access was denied. Login via login function 0xc (or via AT command) is needed.
Bits 5 - 7 are reserved and MUST be cleared.

D) Payload:
The first byte of payload is command ID. Currently there are six command IDs:

Model p900	
Command ID	Command description
0x14	Read multiple parameters
0x15	Reserved
0x18	Write multiple parameters
0x19	Reserved
0x1C	Special functions
0x20	Login functions

E) CRC: CRC-16 ANSI, polynomial x16+x15+x2+1
Initialized to 0xffff. Includes the size byte, address, control byte and payload.
transmitted high byte first. See software example in the Appendix B.
CRC check can be disabled by parameter with ID=17 (register S163). If CRC check is disabled, two dummy bytes must be sent.

Response:
1) Size
It is the size of the diagnostics packet EXCLUDING the size field itself and two bytes of CRC.
2) Six bytes of MAC address: local access MAC address 0x000000000002.
3) User defined magic number: 0x12, 0x34.
4) Status byte: 0x00;

Status byte	Description
0x00	Success
0x01	Error
0x02	Access denied
0x03-0x09	Reserved
0x0A	Wrong format

5) Three bytes of payload:
0x14 - command ID to read parameters,
0x02 - parameter ID for COM1 baud rate,
0x01 – value of parameter: 115.2 kbps.
6) Two bytes of CRC: 0x F503 - CRC is not included in the size

3 Diagnostics Commands.

3.1. Read Parameters.

Command ID 0x14 - read parameters with either fixed size or size that can be computed (like zero-terminated strings).

This command can read multiple parameters in one request. It consists of a list of parameter ID's.

Each parameter is of a specific size. The user application must know the size of each parameter in order to parse the response.

Some parameters are of variable size, such as strings. Normally these parameters should be read in separate requests.

The modem will return the following information in the payload part: parameter ID, value, parameter ID, value... , etc.

The modem will skip invalid parameters and set an error bit in the control byte.

Some of parameters are read/written from/to a proxy structure. The proxy contains a shadow copy of parameters but is not used by the modem during operation. Writes to this proxy will not affect the modem until saved in the NVRAM and the modem restarted. To save these parameters in NVRAM, a write command must be issued. Modem responds on this command and starts 5 seconds delay. This command will interrupt data communication and reset the modem.

Write to some parameters immediately updates their working values but can't be saved. To permanently change them they must be written to the proxy structure and saved.

Read Only parameters can't be changed.

Some parameters are not available to the user via AT command interface and used only for development and advanced troubleshooting.

Multi-byte data is represented in little endian format - lower byte first.

3.2. Write Parameters.

Command ID 0x18 - write parameters.

The structure of the write command is similar to that of a read response - a parameter ID is followed by its value. The acknowledgement contains an error status bit in the control byte. The modem stops parsing on the first unknown parameter ID or wrong size of the parameter.

A returned code will be returned for this parameter Id.

If a parameter exists but the value is out of range, parsing will continue, but a returned code will be returned for this parameter(s) in the response.

The Returned Codes are:
0x00 - all is ok
0x01 - unknown parameter ID
0x02 - parameter is out of range (when checking is done on the parameter value)
0x03 - wrong size of parameter

For example, to write a new value of power to the proxy structure of all modems the following command is sent out:

1	**2**	**3**	**4**	**5**	**6**
0x0c	0xff 0xff 0xff 0xff 0xff 0xff	0x12 0x34	0x02	0x18 0x03 0x1a	0x1b 0xf4,

where
1) 0x0c is the size
2) 0xffffff is the broadcast address
3) 0x1234 is the user magic number
4) 0x02 - request with ACK
5) Payload:
0x18 - command ID - write
0x03 - parameter ID for power
0x1a - a new value of 26 dBm
6) 0x1bf4 - CRC

A modem will respond with an ack:

<u>control byte:</u>

00 - bit 1 set to zero means it was a response. Bit 3 cleared means there were not errors.

<u>Payload:</u>

nothing to report, as there were no errors.

3.3. Special Functions.

Command ID 0x1C – Special functions.

The general structure of the function request is the same as for parameter write except that only one function can be executed in one request and each function has its own set of parameters and response.

Functions are different from the simple read/write in that they can execute delayed actions like flash upgrade, reset or parameter save.

Function arguments are passed in the payload and an ACK of some form will be generated in response.

The following functions are supported:

Function ID	Description
0x01	Flash Upgrade
0x02	Statistics Clear
0x03	Saving Proxy parameters in NVRAM
0x04	Modem Reset
0x05	RSSI report
0x06	Current hop pattern information
0x07	Spectrum analysis
0x08	Channel Table Editor
0x09	Frequency Hopping Table Editor

3.3.1. Flash Upgrade Function.

Flash upgrade function requires command ID and optional parameters.

There are five commands:

Command ID	Description
0x00	Memory preload
0x01	Read status byte
0x02	One 128 byte image block upload
0x03	Start of FLASH CRC checking
0x04	Arming for Flash upgrade
0x05	Start flash upgrade

Unknown command IDs will return code 10.
The format is: 0x1C 0x01 <Command ID>
The format of response is: 0x1C 0x06 <Returned Code>….

1) **Command ID 0** - Memory preload

This command sets the flash image in RAM into a known blank state. In this case the user only needs to upload blocks of image that are not blank.

The payload format is like this:
0x1C 0x01 0x00
where 0x1C - function request
0x01 - function ID - flash upgrade,
0x00 - parameter - memory preload

The response contains an acknowledgement with a return code.
0x1C 0x01 0x00 0x00 where

0x1C - function request
0x01 - function ID - flash upgrade,
0x00 - SUCCESS code.
0x00 - parameter - memory preload

This request can return code 2 (wrong command size or code 3 (busy)
Wrong format means too much data in the payload.
Busy means any of the following:
- previous preload request is still being processed
- CRC checking of the flash image is in process
- flash upgrade is in process
- reset is scheduled

2) **Command ID 1** - Read status byte
Payload: 0x1C 0x01 0x01
Returns 0x1C 0x01 0x04 0x01 in case of bad command format, where 04 is the returned code
or
0x1C 0x01 0x00 0x01 0x08 where
Modem returns:
00 is SUCCESS returned code and 08 is the value of the status byte. The same status byte can be read by the normal parameter read

3) **Command ID 2** - uploading a 128 byte block of image
The format is as following:
0x1C 0x01 0x02 followed by 2 bytes of the block number and 128 bytes of data
The block number is in little endian format.

The request returns
-SUCCESS or
-returned code - busy, as above, or
-0x07 if the block number exceeds the limit of 2560, or
-0x05 if the size of the command was wrong.

4) **Command ID 3** - start of flash CRC checking
The format:
0x1C 0x01 0x03
Returns an ACK with the returned code 8 if the format was wrong, 9 in case the modem was busy and 0 if success.

5) **Command ID 4** - arming the flash upgrade. The arming timeout is 20 seconds. The flash upgrade must be started within this time.
The format:
0x1C 0x01 0x04 0x17 0x71
Returns:
- 0x0a if the size was wrong,
- 0x0b if the arming value parameter (0x7117) was wrong,
- 0x0c if the modem was busy and
- 0x00 if success

6) **Command ID 5** - start flash upgrade.
The format
0x1C 0x01 0x05
Returns

- 0x0d if the size was wrong,
- 0x0e if the arming timeout has expired,
- 0x0f if the modem was busy or
- 0x00 if success.

3.3.2. Statistics clear.

Statistics Clear.
This command clears all statistics. See Note 3.
The format is 0x1C 0x02
The format of response is: 0x1C 0x02 <Returned Code>….
Returned code:
- 0x00 if success,
- 0x01 if wrong size.

3.3.3. Saving proxy parameters in NVRAM.

Saving Proxy parameters in NVRAM.
This command starts the process of saving the proxy structure into NVRAM.
The format is 0x1C 0x03
The format of response is: 0x1C 0x03 <Returned Code>….

This is a delayed function - if started successfully it will wait for: 100ms in case of local command or 5 seconds in case of remote command, before executing and resetting the modem.
It returns:
- 0x00 is success or
- 0x01 if the size was wrong or the modem is already in the process of saving and resetting.

Example for the saving proxy parameters in NVRAM.

1	**2**	**3**	**4**	**5**	**6**
0x0B	0x00 0x00 0x00 0x00 0x00 0x00	0x12 0x34	0x02	0x1C 0x03	0xEB 0xDE
Response:					
1	**2**	**3**	**4**	**5**	**6**
0x0C	0x00 0x00 0x00 0x00 0x00 0x02	0x12 0x34	0x00	0x1C 0x03 0x00	0x67 0x42

3.3.4. Modem Reset.

Modem reset
This command starts the process of resetting the modem.
The format is 0x1C 0x04
The format of response is: 0x1C 0x04 <Returned Code>….

This is a delayed function - if started successfully it will wait for: 100ms in case of local command or 5 seconds in case of remote command, before resetting the modem.
It returns:
- 0x00 is success or
- 0x01 if the size was wrong or the modem is already in the process of resetting.

Example for the Reset function.

1	**2**	**3**	**4**	**5**	**6**
0x0B	0x00 0x00 0x00 0x00 0x00 0x00	0x12 0x34	0x02	0x1C 0x04	0x29 0x9F

Response:

1	**2**	**3**	**4**	**5**	**6**
0x0C	0x00 0x00 0x00 0x00 0x00 0x02	0x12 0x34	0x00	0x1C 04 0x00	0x57 0x40

3.3.5. RSSI Report.

This command requires two parameters - an action and a selection
The format is 0x1C 0x05
The format of response: 0x1C 0x05 <Returned Code> <Action><Selection>

An action can be:

Action	Selection
0-Get Noise	0- an array of average noise values at each hop index for Master
	1- an array of average noise values at each hop index for Slave
	2- an array of min noise values at each hop index for Master
	3- an array of min noise values at each hop index for Slave
	4- an array of max noise values at each hop index for Master
	5- an array of max noise values at each hop index for Slave
1-Clear Noise	0- clear the array of average noise values at each hop index for Master
	1- clear the array of average noise values at each hop index for Slave
	2- clear the array of min noise values at each hop index for Master
	3- clear the array of min noise values at each hop index for Slave
	4- clear the array of max noise values at each hop index for Master
	5- clear the array of max noise values at each hop index for Slave
2-Get RSSI	0- an array of average RSSI at each hop index for Master
	1- an array of average RSSI at each hop index for Slave
3-Clear RSSI	0 clear the array of average RSSI at each hop index for Master
	1 clear the array of average RSSI at each hop index for Master

The RSSI report returns code:
0x00 if success,
0x01 - if the size was wrong,
0x11 to 0x14 if a selection was wrong for the action chosen,
0x15 if the action was invalid.

The actions 0 and 2 return an array of RSSI. The size of the array is the size of the hop pattern. The hop pattern size for FCC compliant country codes is 50 and the modem returns 50 values starting from the hop index 0.

3.3.6. Current Hop Pattern Information.

The format is 0x1C 0x06
Return code 0 if success, 1 if wrong size.
The format of response: 0x1C 0x06 <Returned Code> <Action><Selection>
The following data is returned in little endian format

Action	Selection
0-Narrow Band 400MHz	0- primary channels
	1- secondary channels
1-Frequency hopping 900MHz	0- primary hopping
	1- secondary hopping
2-Frequency hopping 400MHz	0- primary hopping
	1- secondary hopping

Response for Narrow Band 400MHz modem

1 byte	Returned code;
1 byte	Action;
1 byte	Selection;
1 byte Hopping mode. 0 - on channel table, 3 - on frequency, 4- on test table;	
1 byte	Primary channel Tx (register S131)
1 byte	Primary channel Rx (register S132)
1 byte	Secondary channel Tx (register S191)
1 byte	Secondary channel Rx (register S192)
4 bytes	Primary Tx Frequency in Hz
4 bytes	Primary Rx Frequency in Hz
4 bytes	Secondary Tx Frequency in Hz
4 bytes	Secondary Rx Frequency in Hz
4 bytes	Start Frequency in Hz
4 bytes	Stop Frequency in Hz
40 x 5 bytes	First 40 entries of Channel table (total size: 64) 5 bytes each (four bytes channel frequency and one byte channel information: max allowed bandwidth and direction).

Response for Frequency hopping 900MHz modem

1 byte	Returned code;
1 byte	Action;
1 byte	Selection;
1 byte	Hopping mode. 0 - on hop pattern, 1 - on frequency table, 3 - on frequency, 4- on test table;
2 bytes	Test Channel;
1 byte	Hop Pattern size;
2 bytes	Min Channel;
2 bytes	Max Channel;
2 bytes	Channel Space in Hz;
4 bytes	Start Frequency in Hz;
4 bytes	Stop Frequency in Hz;
pattern size x 1 byte	Hopping pattern that has size dependable on selected country (for North America 50) entries one byte each.

Response for Frequency hopping 400MHz modem

1 byte	Returned code;
1 byte	Action;
1 byte	Selection;
1 byte	Hopping mode. 0 - on channel table, 3 - on frequency, 4- on test table;
1 byte	Hop Pattern size
4 bytes	Start Frequency in Hz

4 bytes	Stop Frequency in Hz
50 x 4 bytes	Frequency table that has size 50 entries (four bytes each).

3.3.7. Spectrum Analysis.

This Diagnostics command is supported only in Narrow band modes (410-480 MHz). Modem doesn’t receive and doesn’t transmit data during execution of this command. The next Spectrum Analysis diagnostics command can be sent only after receiving response on previous one.

The format of Spectrum Analysis command call is:

0x1C 0x07 <Start Freq> <Stop Freq> <Step> <Dwell>, where

<Start Freq> - start frequency of spectrum analysis in Hz;
<Stop Freq> - stop frequency of spectrum analysis in Hz;
<Step> - step in Hz, allowed range 1 - 1,000,000 Hz;
<Dwell> - dwell time in ms, allowed range 1-1000 ms;

The maximum number of frequency points is 70. It is calculated according to the formula:

$$N = \frac{\text{Stop Freq} - \text{Start Freq}}{\text{Step}}$$

If Spectrum Analysis command’s parameters are wrong, modem returns error response right away:
0x1C 0x07 <Returned Code> <CRC>, where Returned Code is defined from the table

Returned Code	Description
0x00	Success code, being sent after completion of Spectrum Analysis
0x33	Modem type is not Narrow Band 400 MHz
0x34	Start Frequency is out of the range 410 - 480 MHz
0x35	Stop Frequency is out of the range 410 - 480 MHz
0x36	Start Frequency is greater than Stop Frequency
0x37	Step is not in the range 1 - 1,000,000 Hz
0x38	Number of frequency points is greater than 70
0x39	Dwell time is out of the range 1-1000 ms

If Spectrum Analysis command’s parameters are properly selected, modem starts spectrum analysis and returns the result table when it is done. The response time not shorter than: N x Dwell Time. Modem returns the following message:

0x1C 0x07 <Returned Code> <Number of Freq Points> <RSSI table>, where

<Returned Code > - returned code, code of SUCCESS is 0;
<Number of Freq Points> - number of reported frequency points;
<Rssi table> - Table of measured Rssi values, every record has three bytes: mean RSSI, minimum SSSI, maximum RSSI measured during dwell time.

Example for the Spectrum Analyzer function.

Request:

1	**2**	**3**	**4**	**5**	**6**
0x1b	0x00 0x00 0x00 0x00 0x00 0x00	0x12 0x34	0x02	0x1C 0x07	0x80 0x74 0xD2 0x1A

7	**8**	**9**	**10**
0x00 0x38 0x9C 0x1C	0x20 0xA1 0x07 0x00	0x0A 0x00 0x00 0x00	0x66 0xC4

where

field 6: Start Frequency (450,000,000 Hz)

field 7: Stop Frequency (480,000,000 Hz)
field 8: Step (500,000 Hz)
field 9: Dwell time (10 ms)
field 10:CRC

Response:

1	**2**	**3**	**4**	**5**	**6**	**7**
0xC1	0x00 0x00 0x00 0x00 0x00 0x02	0x12 0x34	0x00	0x1C 0x07	0x00	0x3C

8_1	**8_i**	**8_60**	**9**
0x7B 0x7B 0x7B	0x7E 0x7E 0x7E	0x7F 0x7F 0x7F	0xCC 0x29

where
field 6: Returned Code of operation (SUCCESS)
field 7: Number of frequency points (60 points)
field 8: Mean Rssi, Min Rssi, Max Rssi in dBm
field 9: CRC

3.3.8. Channel Table Editor.

This Diagnostics command is supported only in Narrow band modes (410-480 MHz).

Modem uses it’s current Channel Table while proxy Channel Table is being edited. All changes will be taken in affect only after saving parameters and modem’s reset. The saving proxy channel table is possible only if entered password matches with saved password.

The format of Channel Table Editor is:

0x1C 0x08 <Action> <Optional Parameters>, where
<Action> - action defined by the following table;
<Selection> - selection defined by the following table;
<Optional Parameters> - parameters that are optional that are based on combination of Action and Selection fields.

Action	Optional Parameters	Response Format
0- Read current channel table	<Start Entry0-63> <Number Entries1-40>	<Returned code> <Action> <Frequency,Hz> <Channel Information>
1- Read proxy channel table	<Start Entry0-63> <Number Entries1-40>	<Returned code> <Action> <Frequency,Hz> <Channel Information>
2- Clear proxy channel table	<Index0-63> <Number Entries1-40>	<Returned code> <Action>
3- Set proxy channel table	<Start Entry0-63> <Number Entries1-40> <Frequency,Hz> <Channel Information> …………………………………………… <Frequency,Hz> <Channel Information>	<Returned code> <Action>
4- Save proxy channel table and reset	<Dealer’s PASSWORD, 8 Bytes>	<Returned code> <Action>

Channel table entry has the following format:
<Frequency,Hz> - frequency from the range 410000000 and 480000000 Hz;
<Channel Information> - channel information:

D7	D6	D5	D4	D3	D2	D1	D0

Reserved	0 – maximum BW 6.25kHz	0 – RX direction
	1 – maximum BW 12.5kHz	1 – TX direction
	2 – maximum BW 25kHz	2 – Both Rx and Tx direction

Modem returns the following codes during Channel table editing via Diagnostics.

Returned Code	Description
0x00	Success code
0x01	Wrong total size
0x3A	Not supported action
0x3B	Wrong size, some parameters are missed
0x3C	Start entry is out of range
0x3D	Number of entries is out of range
0x3E	Channel frequency or channel information are out of range
0x3F	Dealer’s password is wrong

Example of channel table reading function of current channel table:

Request:

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0x0E	0x00 0x00 0x00 0x00 0x00 0x00	0x12 0x34	0x02	0x1C	0x08	0x00	0x00

9	**10**
0x05	0xee 0x29

Response:

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0x26	0x00 0x00 0x00 0x00 0x00 0x02	0x12 0x34	0x00	0x1C	08	00	00

9	**10**	**11**
0x81 0x74 0xD2 0x1A	0xA0	0x1F 0x85
0x82 0x74 0xD2 0x1A	0xA0	
0x83 0x74 0xD2 0x1A	0xA0	
0x84 0x74 0xD2 0x1A	0xA0	
0x00 0x00 0x00 0x00	0x00	

where

field 7: Returned Code of operation (SUCCESS)
field 8: Action
field 9: Frequency (450000001/450000002/450000003/450000004/blank)
field 10 Channel information (maximum BW 25kHz, Both Rx and Tx direction/…./blank)
field 11: CRC

Example of channel table reading function of Proxy channel table:

Request:

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0x0E	0x00 0x00 0x00 0x00 0x00 0x00	0x12 0x34	0x02	0x1C	0x08	0x01	0x00

9	**10**
0x05	0x2e 0x78

Response:

1	2	3	4	5	6	7	8
0x26	0x00 0x00 0x00 0x00 0x00 0x02	0x12 0x34	0x00	0x1C	08	00	01

9	10	11
0x80 0x74 0xD2 0x1A	0xA0	0x7B 0x63
0x81 0x74 0xD2 0x1A	0x00	
0x82 0x74 0xD2 0x1A	0x50	
0x83 0x74 0xD2 0x1A	0x90	
0x00 0x00 0x00 0x00	0x00	

where
field 7: Returned Code of operation (SUCCESS)
field 8: Action
field 9: Frequency (450000000/450000001/450000002/45000000/blank)
field 10 Channel information (maximum BW 25kHz, Both Rx and Tx direction/ BW 6.25kHz, Rx/ BW 12.5kHz, Tx/ BW 12.5kHz, Both Rx and Tx/ blank)
field 11: CRC

Request with wrong parameters:

1	2	3	4	5	6	7	8
0x0E	0x00 0x00 0x00 0x00 0x00 0x00	0x12 0x34	0x02	0x1C	0x08	0x0F	0x00

9	10
0x05	0xED 0x19

Response:

1	2	3	4	5	6	7	8
0x0D	0x00 0x00 0x00 0x00 0x00 0x02	0x12 0x34	0x08	0x1C	0x08	3A	0F

9
0x57 0x25

Set two entries of channel table starting from 0:

1	2	3	4	5	6	7	8
0x18	0x00 0x00 0x00 0x00 0x00 0x00	0x12 0x34	0x02	0x1C	0x08	0x03	0x00

9	10	11
0x02	0x80 0x74 0xD2 0x1A 0xA0	0x55 0x2f
	0x80 0x74 0xD2 0x1A 0xA0	

Response:

1	2	3	4	5	6	7	8
0x0D	0x00 0x00 0x00 0x00 0x00 0x02	0x12 0x34	0x00	0x1C	0x08	0x00	0x03

9
0x33 0xD6

Saving channel table:

1	2	3	4	5	6	7	8
0x14	0x00 0x00 0x00 0x00 0x00 0x00	0x12 0x34	0x02	0x1C	0x08	0x04	0x41 0x42 0x43

	9
0x44 0x45 0x46 0x47 0x48	0x0e 0xae

Response:

1	2	3	4	5	6	7	8
0x0b	0x00 0x00 0x00 0x00 0x00 0x02	0x12 0x34	0x00	0x1C	0x08	0x00	0x04

9

0xcc 0xbe

Updated channel table will be saved with some delay: 100ms in case of local command or 5 seconds in case of remote command, modem will be reset and started up with new channel table.

3.3.9. Frequency Hopping Table Editor.

This Diagnostics command is supported only in 400 MHz frequency hopping modes (410-480 MHz).

Modem uses it's current frequency hopping table while proxy frequency hopping tables are being edited. All changes will be taken in affect only after saving parameters and modem's reset.

The format of Channel Table Editor is:

0x1C 0x09 <Action> <Optional Parameters>, where

<Action> - action defined by the following table;

<Selection> - selection defined by the following table;

<Optional Parameters> - parameters that are optional that are based on combination of Action and Selection fields.

Action	Selection	Optional parameters	Response Format
0- Read selected frequency table	0- Primary Frequency table 1- Secondary Frequency table		<Returned code> <Action> <Selection> <Frequency0-49, Hz>
1- Read selected proxy frequency table	0- Primary Frequency table 1- Secondary Frequency table		<Returned code> <Action> <Selection> <Frequency0-49, Hz>
2- Set selected proxy frequency table	0- Primary Frequency table 1- Secondary Frequency table	<Frequency0-49, Hz>	<Returned code> <Action> <Selection>
3- Save both proxy frequency tables and reset	2- Both Frequency tables		<Returned code> <Action> <Selection>

Frequency table entry has the following format:

<Frequency,Hz> - frequency from the range 410000000 and 480000000 Hz;

Modem returns the following codes during Frequency tables editing via Diagnostics.

Returned Code	Description
0x00	Success code
0x01	Wrong total size
0x40	Action is not supported
0x41	Wrong selection
0x42	Wrong size of command, some parameters are missed
0x43	Frequency is out of allowed range
0x44	Wrong saving parameter

Examples:

Reading Primary Frequency Table:

1	2	3	4	5	6	7	8
0x0D	0x00 0x00 0x00 0x00 0x00 0x00	0x12 0x34	0x02	0x1C	0x09	0x00	0x00

9
0xEB 0x3F

Response:

1	2	3	4	5	6	7	8	9
0xD6	0x00 0x00 0x00 0x00 0x00 0x01	0x12 0x34	0x00	0x1C	0x09	0x00	0x00	0x00

10	11
0xC0 0x92 0xED 0x19 (entry 0)	0x74 0xC9
0xC0 0x92 0xED 0x19 (entry 1)	
…………………………………	
0xC0 0x92 0xED 0x19 (entry 49)	

where
field 7: Returned Code of operation (SUCCESS)
field 8: Action
field 9: Selection
field 10: Frequency (435000000…)
field 11: CRC

4. Login Functions.

Command ID 0x20 - special functions.

The structure of this function request is the same as for general functions. The difference is that special functions only work via local connection (address 000000000000) and they are processed even if the user is not logged in.

The following functions are supported:

Function ID	Description
0x20	Login

4.1. Login Function.

Login description:

As with login via AT command interface, the user has 5 attempts to enter login information after which the modem will lock the user out for 10 minutes.

After 10 minutes of complete inactivity on the diagnostics port the attempt count will be restored. If the power is cycled, even after 10 minutes, on power up the user
will still have to wait for 10 minutes before logging in.

The payload format is like this:
0x20 0x01 0x00 <login password> 0x00,
where
0x20 - special function request
0x01 - function ID (login function)

0x00 - function parameter (sending login information)
<login password> - the string set by at login command in AT interface - 1 to 32 chars.
0x00 - string termination

The function returns codes:
- 0x01 if size was wrong,
- 0x02 if the login was incorrect but the user still has more attempts allowed to try, or
- 0x10 if unknown command,
- 0x16 if the user is locked out for 10 minutes, or
0x00 if success.

5. Warnings For Users of Diagnostics Protocol

Using diagnostics channel customer must follow special rules:

1) Diagnostics channel has a 20ms character time-out timer. If you have inter-command delay shorter than 20 ms, commands could be overlapped.
2) If modem returns a response on user's diagnostics command, user must wait for this response before sending next diagnostics command.
3) Some diagnostics parameter are being taken in effect immediately, other are written in a special proxy structure that will be saved in NVRAM using special command. Saved parameters will be used only after modem's reset.

Appendix A

Parameters IDs for p400 modem Table 1

Parameter ID	Parameter	Size, Bytes	S-Register	Comments
1	Serial Channel Mode	1	S142	Proxy
2	Serial Baud Rate	1	S102	Proxy
3	Output Power, dBm (effective after reset)	1	S108	Proxy
4	Hop Time	1	S109	Proxy
5	String Version	var	ATI3	Read only
6	Packet Size Minimum	2	S111	Proxy
7	Status 1	1	System	**Note 1**
8	Packet Size Maximum	2	S112	Proxy
9	Packet Retransmissions	1	S113	Proxy
10	Repeaters in System	1	S141	Proxy
11	Protocol Type	1	S217	Proxy
12	Handshaking	1	&K	Proxy
13	Operating Mode	1	S101	Proxy
14	Wireless Link Rate	1	S103	Proxy, **Note 2**
15	Escape Character	1	S2	Proxy
16	Destination Address	2	S140	Proxy
17	Diagnostics CRC Control	1	S163	Proxy
18	Power Up Mode	1	S0	Proxy
19	Serial Data Format	1	S110	Proxy
20	DCD Control	1	&C	Proxy
21	Master Bandwidth, %	1	S250	Proxy
22	reserved			
23	DCD Pulse Period	1	S183	Proxy
24	Master system time	4	System	
25	Network ID	4	S104	Proxy
26	Repeat Interval	1	S115	Proxy
27	Character Time-Out	1	S116	Proxy
28	Roaming	2	S118	Proxy
29	Output Power, dBm	1	S108	Immediate
30	Channel Request Mode	1	S244	Proxy
31	reserved			
32	reserved			
33	Diagnostics Retransmissions	1	S214	Proxy
34	Reserved			
35	Reserved			
36	Diagnostics Retransmissions		S214	Immediate
37	Max Buffers IN Storage	2	S232	Proxy
38	Sync Time-Out	2	S248	Proxy
39	Packets Per Hop Tx Limit	1	S249	Proxy
40	Slave Channel Allocation Limit	1	S252	Proxy
41	Forward Error Correction	1	S158	Proxy, **Note2**
42	Network type	1	S133	Proxy
43	Routing Time to Live	1	S235	Proxy

Parameter ID	Parameter	Size, Bytes	S-Register	Comments
44	Master Channel Request Time-Out	1	S234	Proxy
45	Max Buffers OUT Storage	2	S236	Proxy
46	DSR Control	1	&S	Proxy
47	DTR Control	1	&D	Proxy
48	Transmit Done Time-Out	2	S146	Proxy
49	Receive Done Time-Out	2	S147	Proxy
50	Adapt. Change Time-Out	2	S151	Proxy
51	Reserved			
52	Address Tag	1	S153	Proxy
53	User String	var	ATI0	Proxy
54	Country Code	1	S247	Read only
55	Temperature	1	System	**Note 3**
56	Synchronized level	2	System	
57	Power voltage sensor	2	System	
58	Data Time to Live	2	S184	Proxy
59	Encryption Mode	1	S159	Proxy
60	RSSI Averaged Master	1	System	S124
61	RSSI Averaged Slave	1	System	S123
62	Aggregate noise Master	1	System	
63	Aggregate noise Slave	1	System	
64 - 79	Reserved			
80	Unit Address	2	S105	Proxy
81	Command Echo	1	ATE	Proxy
82	Quiet Mode	1	ATQ	Proxy
83	Verbose Mode	1	ATV	Proxy
84	Word Result Mode	1	ATW	Proxy
85	Set factory defaults	1	AT&Fi	**Note 4**
86	Reserved			
87	Message to user (up-to 128 characters)	var	System	Read only
88	No sync data intake	1	S130	Proxy
121	Multi-Master Mode	1	S154	Proxy
122	Secondary Network ID	4	S241	Proxy
123	Slave’s Tuning Time	2	S230	Proxy
124	Master Hop Allocation Time Out	1	S251	Proxy
125	Restrictions Enable	1	S148	Proxy
126	Sync Mode	1	S150	Proxy
127	Fast Sync Hold on ACK	1	S152	Proxy
128	Current Saving Mode	1	S239	Proxy
129	Hopping Pattern Size	1	S165	Proxy
130	RF Emission Control	1	S167	Proxy
131	State Filter Address	2	System	
132	Filtering Mode	1	S168	Proxy
133	Rate Change TMO	1	S162	Proxy
134	Restriction Zone	1	S178	Proxy

Parameter ID	Parameter	Size, Bytes	S-Register	Comments
135	Header Type	1	S176	Proxy
136	Master Channel Allocation Limit	1	S253	Proxy
137	Data CRC Control	1	S221	Proxy
138	Fast TDMA Max. Packet Size	2	S212	Proxy
139	Packet Retry Limit for Repeaters	1	S213	Proxy
140	Reset Sync	1	System	
141	DCD On	1	S120	Proxy
142	DCD Off	1	S121	Proxy
143	Routing Request Time to Live	2	S219	Proxy
144	Modem Type	1	S128	Proxy
145	NB Bandwidth	1	S125	Proxy
146	Modulation Scheme	1	S127	Proxy
147	RSSI Threshold for CSMA	1	S51	Proxy
148	Primary Channel Tx	1	S131	Proxy
149	Primary Channel Rx	1	S132	Proxy
150	Secondary Channel Tx	1	S191	Proxy
151	Secondary Channel Rx	1	S192	Proxy
152	Data Buffering Mode	1	S231	Proxy
153	Time Slot Size for CSMA	1	S137	Proxy
154	Tx Attack Delay	2	S185	Proxy
155	NB Protocol Selection	1	S186	Proxy
156	TrimTalk Discard Duplicated Data	1	S187	Proxy
157	TrimTalk Strip off Markers	1	S188	Proxy
158	TrimTalk Enable Uplink	1	S189	Proxy
159	PCC Ignore Received Unit Address	1	S190	Proxy
160	After Tx Delay	2	S138	Proxy
161	Satel Full CRC Use	1	S129	Proxy
162	Reserved			
163	Reserved			
164	Tx / Rx priority	1	S136	Proxy
165	Compatibility with MHX921 at 345kbps	1	S139	Proxy
166	String Serial Number	Variable	ATI7	Read only
167	Statistics	Variable		Read only, **Note 5**
168	Memory usage	Variable		Read only, **Note 6**
169	Compatibility Type	1	S226	Proxy
170	Trimtalk Compatibility	1	S227	Proxy
171	Call Sign ID	16	S228	Proxy
172	Call Sign Interval	1	S233	Proxy

Notes.

Note 1

Par ID 7 - status1 represents the following structure that is mostly used in flash upgrade.

D0- CRCcheckDone;

D1- CRCcheckInProgress;
D2- CRCcheckResult;
D3- flashPreloadDone;
D4- flashPreloadInProgress;
D5- upgradeScheduled;
D6- paramSaveScheduled;
D7- resetScheduled;

Note 2

These parameters' IDs, as S-registers S103 and S158 are shared by FH and NB modems. Valid ranges for settings are based on currently used modem type (register S128).

Note 3

Temperature is offset by 55. To obtain actual temperature subtract 55 from the reading. For example reading 95 indicates the actual temperature of 95-55 = +40C. Reading 40 means 40-55 = -15C

Note 4

The following parameters IDs are valid

Parameter ID	Description
1	FH Master Fast PMP
2	FH Slave Fast PMP
3	FH Repeater Fast PMP
4	FH Master Slow PMP
5	FH Slave Slow PMP
6	FH Master Fast PP
7	FH Slave Fast PP
8	FH Master Slow PP
9	FH Slave Slow PP
10	FH Master Fast PMP no Time ACK
11	FH Master Fast P2P no Time ACK
12	FH Master Fast PP no Time ACK
15	FH Master WL
16	FH Slave WL
18	FH Master Fast TDMA
19	FH Slave Fast TDMA
51	NB Transparent Protocol
52	NB Transparent Protocol w Rep.
53	NB Pacific Crest Protocol, needs different UAs
54	NB TrimTalk Protocol no Rep.
55	NB TrimTalk Protocol Rep.1
56	NB TrimTalk Protocol Rep.2
57	NB TrimTalk Protocol Base w Rep.
58	NB 3AS Protocol, SRate=9.6kbps, BW=12.5kHz
59	NB 3AS Protocol, SRate=19.2kbps, BW=25kHz

Parameter ID	Description
60	NB Satel 3AS Protocol, BitRate=9.6kbps,BW=12.5kHz,FEC Off,4FSK,Type 1
61	NB Satel 3AS Protocol, BitRate=19.2kbps,BW=25kHz,FEC Off,4FSK,Type 1
62	NB Pac.Crest Trans.Protocol, Bit Rate=4.8kbps,BW=12.5kHz,FEC On,2FSK
63	NB Trimtalk Protocol, Bit Rate=4.8kbps, BW=12.5kHz, 2FSK
64	NB Pac.Crest 4FSK Protocol,BitRate=9.6kbps,BW=12.5kHz,FEC On,4FSK
65	NB Pac.Crest 4FSK Protocol,BitRate=19.2kbps,BW=25kHz,FEC On,4FSK"
66	NB Pac.Crest FST Protocol,BitRate=9.6kbps,BW=12.5kHz,FEC On,4FSK
67	NB Pac.Crest FST Protocol,BitRate=19.2kbps,BW=25kHz,FEC On,4FSK
68	NB Pac.Crest FST Protocol,BitRate=9.6kbps,BW=12.5kHz,FEC On,4FSK,Type2
69	NB Pac.Crest FST Protocol,BitRate=19.2kbps,BW=25kHz,FEC On,4FSK,Type2
100	Reset Hopping Modes

Note 5

Statistics represents the following structure:

- bytesRx	-4 bytes;
- kBytesRx	-4 bytes;
- bytesTx	-4 bytes;
- kBytesTx	-4 bytes;
- errCorrect	-4 bytes;
- syncLost	-4 bytes;
- packetsTx	-4 bytes;
- packetsRx	-4 bytes;
- errCRC	-4 bytes;
- pktError	-4 bytes;
- pktErrorType	-4 bytes;
- syncCount	-4 bytes;

The number and size of these parameters can be changed between firmware versions. Newer version may have extra members added to the end of the structures of those parameters.

Example:

Request

1	**2**	**3**	**4**	**5**	**6**	**7**
0x0b	0x00 0x00 0x00 0x00 0x00 0x00	0x12 0x34	0x02	0x14	0xA7	0x90 0xD8

Response:

1	**2**	**3**	**4**	**5**	**6**
0x3B	0x00 0x00 0x00 0x00 0x00 0x01	0x12 0x34	0x00	0x14	0xA7

7		**8**
0x0A 0x00 0x00 0x00	bytesRx	0x93 0x09
0x00 0x00 0x00 0x00	kbytesRx	
0x00 0x00 0x00 0x00	bytesTx	
0x00 0x00 0x00 0x00	kbytesTx	
0x00 0x00 0x00 0x00	errCorrect	
0x00 0x00 0x00 0x00	syncLost	
0x00 0x00 0x00 0x00	packetsTx	
0x0A 0x00 0x00 0x00	packetsRx	

0x00 0x00 0x00 0x00	errCRC
0x00 0x00 0x00 0x00	pktError
0x00 0x00 0x00 0x00	pktErrorType
0x0A 0x00 0x00 0x00	syncCount

Note 6

Memory usage represents the sizes of all buffer queues in the following order:

Q_FREE	ID=0	1 byte;
Q_UP	ID=1	1 byte;
Q_DOWN	ID=2	1 byte;
Q_USER	ID=3	1 byte;
Q_DOWN_TX	ID=5	1 byte;
Q_USER_IN	ID=7	1 byte;

The number and size of these parameters can be changed between firmware versions. Newer version may have extra members added to the end of the structures of those parameters.

Appendix B

Function for calculation CRC16 in language C.

```
static u16          crc_value__;

const unsigned short crc_16table[] = {
        0x0000, 0xc0c1, 0xc181, 0x0140, 0xc301, 0x03c0, 0x0280, 0xc241,
        0xc601, 0x06c0, 0x0780, 0xc741, 0x0500, 0xc5c1, 0xc481, 0x0440,
        0xcc01, 0x0cc0, 0x0d80, 0xcd41, 0x0f00, 0xcfc1, 0xce81, 0x0e40,
        0x0a00, 0xcac1, 0xcb81, 0x0b40, 0xc901, 0x09c0, 0x0880, 0xc841,
        0xd801, 0x18c0, 0x1980, 0xd941, 0x1b00, 0xdbc1, 0xda81, 0x1a40,
        0x1e00, 0xdec1, 0xdf81, 0x1f40, 0xdd01, 0x1dc0, 0x1c80, 0xdc41,
        0x1400, 0xd4c1, 0xd581, 0x1540, 0xd701, 0x17c0, 0x1680, 0xd641,
        0xd201, 0x12c0, 0x1380, 0xd341, 0x1100, 0xd1c1, 0xd081, 0x1040,
        0xf001, 0x30c0, 0x3180, 0xf141, 0x3300, 0xf3c1, 0xf281, 0x3240,
        0x3600, 0xf6c1, 0xf781, 0x3740, 0xf501, 0x35c0, 0x3480, 0xf441,
        0x3c00, 0xfcc1, 0xfd81, 0x3d40, 0xff01, 0x3fc0, 0x3e80, 0xfe41,
        0xfa01, 0x3ac0, 0x3b80, 0xfb41, 0x3900, 0xf9c1, 0xf881, 0x3840,
        0x2800, 0xe8c1, 0xe981, 0x2940, 0xeb01, 0x2bc0, 0x2a80, 0xea41,
        0xee01, 0x2ec0, 0x2f80, 0xef41, 0x2d00, 0xedc1, 0xec81, 0x2c40,
        0xe401, 0x24c0, 0x2580, 0xe541, 0x2700, 0xe7c1, 0xe681, 0x2640,
        0x2200, 0xe2c1, 0xe381, 0x2340, 0xe101, 0x21c0, 0x2080, 0xe041,
        0xa001, 0x60c0, 0x6180, 0xa141, 0x6300, 0xa3c1, 0xa281, 0x6240,
        0x6600, 0xa6c1, 0xa781, 0x6740, 0xa501, 0x65c0, 0x6480, 0xa441,
        0x6c00, 0xacc1, 0xad81, 0x6d40, 0xaf01, 0x6fc0, 0x6e80, 0xae41,
        0xaa01, 0x6ac0, 0x6b80, 0xab41, 0x6900, 0xa9c1, 0xa881, 0x6840,
        0x7800, 0xb8c1, 0xb981, 0x7940, 0xbb01, 0x7bc0, 0x7a80, 0xba41,
        0xbe01, 0x7ec0, 0x7f80, 0xbf41, 0x7d00, 0xbdc1, 0xbc81, 0x7c40,
        0xb401, 0x74c0, 0x7580, 0xb541, 0x7700, 0xb7c1, 0xb681, 0x7640,
        0x7200, 0xb2c1, 0xb381, 0x7340, 0xb101, 0x71c0, 0x7080, 0xb041,
        0x5000, 0x90c1, 0x9181, 0x5140, 0x9301, 0x53c0, 0x5280, 0x9241,
        0x9601, 0x56c0, 0x5780, 0x9741, 0x5500, 0x95c1, 0x9481, 0x5440,
        0x9c01, 0x5cc0, 0x5d80, 0x9d41, 0x5f00, 0x9fc1, 0x9e81, 0x5e40,
        0x5a00, 0x9ac1, 0x9b81, 0x5b40, 0x9901, 0x59c0, 0x5880, 0x9841,
        0x8801, 0x48c0, 0x4980, 0x8941, 0x4b00, 0x8bc1, 0x8a81, 0x4a40,
        0x4e00, 0x8ec1, 0x8f81, 0x4f40, 0x8d01, 0x4dc0, 0x4c80, 0x8c41,
        0x4400, 0x84c1, 0x8581, 0x4540, 0x8701, 0x47c0, 0x4680, 0x8641,
        0x8201, 0x42c0, 0x4380, 0x8341, 0x4100, 0x81c1, 0x8081, 0x4040};

// Function: crc16Add(u8 data)
//
// Description: Adds CRC for parameter data.
//
void crc16Add(u8 data)
{
        u8 x;

        x = (u8)(crc_value__ ^ (u16)data);
        crc_value__ = (u16)(crc_value__ >> 8);
        crc_value__ ^= crc_16table[x];
}
```