台揚科技股份有限公司
MICROELECTRONICS TECHNOLOGY INC.
ISO 9001 Certified

MTI RU00-M06-X RFID PCIe M.2 Module

Command Reference Manual

Version 1.0

Contents

1 Introduction 6
1.1 Terminology 6
2 System Development Overview 7
2.1 MTI RFID PCIe M.2 Module Application Overview 7
2.2 Theory of Operation 7
3 Command Set 9
3.1 Command Set Summary 9
3.2 Packet Format Specification 12
3.2.1 Command Packet Format 12
3.2.2 Response Packet Format 12
3.2.3 Report Packet Format 13
4 Command Introduction 23
4.1 RFID PCIe M.2 Module Configuration 23
4.1.1 RFID PCIe M.2 Module Device ID 23
4.1.2 Configuring the Operation Mode 24
4.1.3 RFID PCIe M.2 Module Link Profiles 25
4.1.4 Accessing Low-Level MAC Registers 26
4.1.5 Accessing Low-Level MAC Banked Registers 27
4.1.6 Retrieving Low-Level MAC Register Information 28
4.1.7 Controlling the Inventory-Response Packet Format 29
4.2 Antenna Port Configuration 31
4.2.1 Antenna-Port State 31
4.2.2 Antenna-Port Configuration 32
4.2.3 Global Antenna-Port Sense Threshold 35
4.3 ISO 18000-6C Tag-Select Operation/Tag-Access Parameters 37
4.3.1 Specifying Tag-Selection Criteria 37
4.3.2 Specifying Post-Singulation Match Criteria 42
4.3.3 Inventory-Round Configuration Parameters 44
4.3.4 Accessing the Tag Access Password 49
4.3.5 Accessing Tag Writing Data Buffer 50
4.4 ISO 18000-6C Tag-Protocol Operation 52
4.4.1 Tag Inventory Operation 52
4.4.2 Tag Read Operation 53
4.4.3 Tag Write Operation 54

4.4.4 Tag Kill Operation 55
4.4.5 Tag Lock Operation 56
4.4.6 Tag Multiple Write Operation 58
4.4.7 Tag Block Write Operation 60
4.4.8 Tag Block Erase Operation 61
4.5 RFID PCIe M.2 Module Control Operation 63
4.5.1 Canceling a Tag-Protocol Operation 63
4.5.2 Pausing a Tag-Protocol Operation 63
4.5.3 Resuming a Tag-Protocol Operation 63
4.5.4 Performing a Software Reset 64
4.5.5 Perform a Firmware Reset to Bootloader 64
4.5.6 RFID PCIe M.2 Module Power Management 64
4.6 RFID PCIe M.2 Module Firmware Access 66
4.6.1 Retrieving the MAC Firmware Version Information 66
4.6.2 Retrieving the MAC Firmware Debug Value 66
4.6.3 Clearing a MAC Firmware Error 66
4.6.4 Retrieving a MAC Firmware Error Code 67
4.6.5 Retrieving the Bootloader Firmware Version Information 68
4.6.6 Accessing MAC-Resident OEM Configuration Data 68
4.6.7 Accessing RFID PCIe M.2 Module Hardware Registers 69
4.6.8 Accessing Regulatory Region of Operation 71
4.6.9 Retrieving the MAC-Resident OEMCfg Version Information 72
4.6.10 Retrieving the MAC-Resident OEMCfg Update Number Information 73
4.7 RFID PCIe M.2 Module Region Test Support 74
4.6.11 Accessing the Test Antenna-Port Configuration 74
4.6.12 Accessing the Test Frequency Configuration 75
4.6.13 Accessing the Pulse Time of Random Data Transmission 76
4.6.14 Accessing the Test Inventory Configuration 77
4.6.15 CW Control with No Data Modulation 78
4.6.16 Injecting a Random Data 79
4.6.17 Transmitting a Random Data with Modulation 79
4.7 RFID PCIe M.2 Module Engineering Test Support 81
4.7.1 Retrieving the Temperature 81
4.7.2 Retrieving the RF Power Level 81
5 Use Cases 82
5.1 Initial Configuration of the RFID PCIe M.2 Module 82
5.2 Inventory for a Single Tag 83
5.3 Read the PC Value from EPC Memory Bank of a Tag 85

5.4 Read the EPC Value from EPC Memory Bank of a Tag 86
5.5 Write the EPC Value to EPC Memory Bank of a Tag 87
5.6 Write Once the Full EPC Value to EPC Memory Bank of a Tag 89
5.7 Kill Single Tag 91
5.8 Inventory with Tag-selection Criteria for a Specific Tag 94
6 APPENDIX A - Singulation Algorithm Introduction 96
6.1 Fixed Q (Generic) Algorithm 96
6.2 Dynamic Q Algorithm 98
6.3 Fixed Q Algorithm with Access 100
7 APPENDIX B - MTI MAC Firmware Error Codes 102
8 APPENDIX C - Calculation of CRC-16 111
8.1 CRC-16 Encoder/Decoder 111
8.2 Example C Code to Generate the CRC-16 Value 112
8.3 Examples for Calculated Result of CRC-16 112
9 APPENDIX D - 6C Tag Memory Map 113
9.1 ISO 18000-6C Tag Memory Map 113
10 APPENDIX E - Frequency Channel Tables 114
10.1 United States/Canada/Mexico Region Frequency Channel Table 114
10.2 Europe Region Frequency Channel Table (ETSI EN 302 208) 114
10.3 China Region Frequency Channel Table 114
10.4 Japan Region Frequency Channel Table 115
10.5 Europe2 Region Frequency Channel Table (ETSI EN 300 220) 115
10.6 Taiwan Region Frequency Channel Table 115
10.7 South Korea Region Frequency Channel Table 115
10.8 Australia/New Zealand Region Frequency Channel Table 115
10.9 Brazil Region Frequency Channel Table 116
10.10 Israel Region Frequency Channel Table 116
10.11 India Region Frequency Channel Table 116

Revision History

Version Number	Description	Revision Date
1.0	(1) First release of document. The revision is used on firmware version 0.9.0 or later.	October 2016

1 Introduction

The *MTI RU00-M06-X RFID Module Command Reference Manual* provides detailed information for configuring, controlling, and accessing the MTI RFID PCIe M.2 module. Each command and its parameters are then described in detail.

The intended audience for this document includes the following:

- RFID middleware software developers who will be creating software for configuring, controlling, and accessing the MTI RFID PCIe M.2 module.
- RFID reader manufacturers who will need to understand how to configure, control, and access the MTI RFID PCIe M.2 module during development and testing phases.

1.1 Terminology

Table 1.1 - Terminology

Term	Description
CRC	Cyclic Redundancy Check
CW	Carrier Wave
EPC	Electronic Product Code
GPIO	General Purpose I/O
PCIe M.2	PCI Express M.2
ISO	International Standards Organization
LBT	Listen Before Talk
MAC	Media Access Control
OEM	Original Equipment Manufacturer
RFID	Radio Frequency Identification
RX	Receiver
TID	Tag Identifier
TX	Transmitter
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

2 System Development Overview

2.1 MTI RFID PCIe M.2 Module Application Overview

Figure 2.1 illustrates a typical MTI RFID PCIe M.2 module application. The MTI RFID PCIe M.2 module accesses (reads, writes, etc.) ISO 18000-6C (EPC Class 1 Generation 2) RFID tags and passes the digital data stream via the USB to a host processor. The host processor connects to and controls the MTI RFID PCIe M.2 module via the USB interface. For the USB interface, after connecting the MTI RFID PCIe M.2 module to the computer, it is automatically installed as a HID (Human Interface Device). The host processor uses the USB VID and PID combination to find the MTI RFID PCIe M.2 module. The USB VID and PID numbers are both as follows,

USB Vendor ID: 0x24E9 / Product ID: 0x5206

Connection to the external RFID network infrastructure is achieved via an application layer. This application layer is depended on the embedded processor. The detailed description of this functionality is outside the scope of this document.

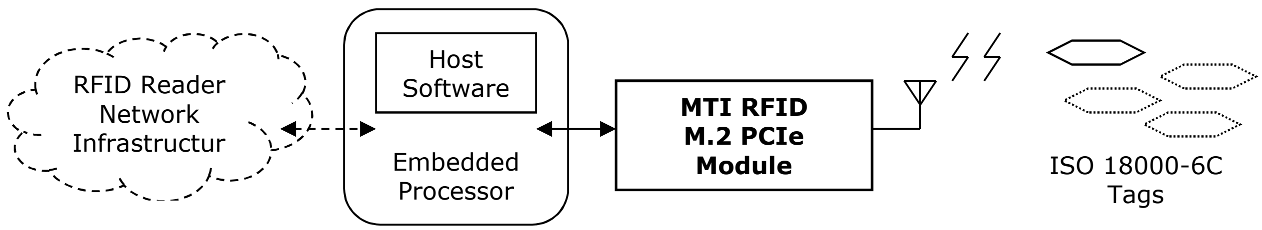

Figure 2.1 - Typical MTI RFID PCIe M.2 Module Application Architecture

The MTI RFID PCIe M.2 module is based on Impinj Indy R2000 chip technology.

2.2 Theory of Operation

The command set provides many programming commands for controlling ISO 18000-6C compatible MTI RFID PCIe M.2 module. The host can configure the MTI RFID PCIe M.2 module for operation, and tag protocol operations can be issued.

The host initiates transactions with ISO 18000-6C tags or tag populations by executing ISO 18000-6C tag-protocol operations. The command set exposes direct access to the following ISO 18000-6C tag-protocol operations:

- Inventory
- Read
- Write
- Kill
- Lock
- Block Write
- Block Erase

When executing tag-protocol operations, the command set provides the host with the ability to configure tag-selection (i.e. ISO 18000-6C Select) criteria and query (i.e. ISO 18000-6C Query) parameters. The command set extends the ISO 18000-6C tag-protocol operations by additionally providing the host the ability to specify a post-singulation

Electronic Product Code (EPC) match mask, as well as the number of tags to which the operation is to be applied.

When a tag access operation is issued to an RFID PCIe M.2 module, it acts as a blocking operation on that RFID PCIe M.2 module. This means any further requests to that RFID PCIe M.2 module is rejected. The host must allow the access operation to complete or explicitly cancel the operation.

If the host receives a non-critical fault notification from the RFID PCIe M.2 module, it needs to implement an appropriate fault handling routine. For example, this may require canceling any blocking tag access operation that is in progress and investigating the details of the fault indication. The exact implementation of the routing is use case dependant.

The MTI MAC firmware provides some report packets for presenting tag-protocol operation response data to the host. Tag-protocol operation results include EPC values returned by the Inventory operation, read data returned by the Read operation, and operation status returned by the Write, Kill, Lock, Block Write, and Block Erase operations.

The command set supports the configuration and control of the individual antenna ports on the RFID PCIe M.2 module. By controlling which antenna ports are enabled/disabled, a host can exercise fine-grained control upon which antenna ports a tag-protocol operation is executed. A host may wish to take advantage of the ability to selectively enable a single antenna for the purpose of controlling the sequencing and initiation of tag-protocol operations. If instead a host wishes to leave sequencing and initiation of a tag-protocol operation on a series of enable antennas to the RFID PCIe M.2 module, the command set also supports automatic execution of a tag-protocol operation sequentially on multiple antennas, which furthermore can be configured to execute once or to execute continuously until explicitly stopped by the host. The host is given fine-grained control to configure, on a per-antenna-port basis:

- A time limit for performing tag-protocol operations (dwell time)
- The number of times a tag-protocol operation is executed (number of inventory rounds)
- RF characteristics (for example, RF power)

The command set provides the host with access to (i.e. read and write) the OEM configuration data area on the RFID PCIe M.2 module. The host can use the OEM configuration data area to store and retrieve the specific hardware configuration and capabilities of the RFID PCIe M.2 module. The host can read an RFID PCIe M.2 module's OEM configuration data area immediately after gaining exclusive control of the RFID PCIe M.2 module and use that data to configure and control low-level parameters of the RFID PCIe M.2 module.

The command set also supports low-level control of the RFID PCIe M.2 module's MAC firmware. First, it provides a software reset operation that causes the RFID PCIe M.2 module's MAC firmware to perform a software reboot, returning the RFID PCIe M.2 module to a default idle state. Second, for low power standby mode, the relative command provides the host a means for entering this power mode to reduce power consumption. Third, for in-the-field upgrade, the relative command provides the host a means for updating the RFID PCIe M.2 module's MAC firmware image.

3 Command Set

3.1 Command Set Summary

Table 3.1 - Command Set Summary

CMD ID	Command Name	Simple Description	Note
RFID PCIe M.2 Module Configuration			
0x00	RFID_RadioSetDeviceID	Setting the Device ID	
0x01	RFID_RadioGetDeviceID	Retrieving the Device ID	
0x02	RFID_RadioSetOperationMode	Setting the Operation Mode	
0x03	RFID_RadioGetOperationMode	Retrieving the Operation Mode	
0x04	RFID_RadioSetCurrentLinkProfile	Setting the Current Link Profile	
0x05	RFID_RadioGetCurrentLinkProfile	Retrieving the Current Link Profile	
0x06	RFID_RadioWriteRegister	Setting Low-Level MAC Registers	
0x07	RFID_RadioReadRegister	Retrieving Low-Level MAC Registers	
0x08	RFID_RadioWriteBankedRegister	Setting Low-Level MAC Banked Registers	
0x09	RFID_RadioReadBankedRegister	Retrieving Low-Level MAC Banked Registers	
0x0A	RFID_RadioReadRegisterInfo	Retrieving Low-Level MAC Register Information	
0x0E	RFID_RadioSetInventoryPacketFormat	Setting the Inventory-Response Packet Format	
0x0F	RFID_RadioGetInventoryPacketFormat	Retrieving the Inventory-Response Packet Format	
Antenna Port Configuration			
0x10	RFID_AntennaPortSetState	Setting Antenna-Port State	
0x11	RFID_AntennaPortGetState	Retrieving Antenna-Port State	
0x12	RFID_AntennaPortSetConfiguration	Setting Antenna-Port Configuration	
0x13	RFID_AntennaPortGetConfiguration	Retrieving Antenna-Port Configuration	
0x14	RFID_AntennaPortSetSenseThreshold	Setting Global Antenna-Port Sense Threshold	
0x15	RFID_AntennaPortGetSenseThreshold	Retrieving Global Antenna-Port Sense Threshold	
ISO 18000-6C Tag-Select Operation			
0x20	RFID_18K6CSetActiveSelectCriteria	Setting Active Tag-Selection Criteria	
0x21	RFID_18K6CGetActiveSelectCriteria	Getting Active Tag-Selection Criteria	
0x22	RFID_18K6CSetSelectCriteria	Setting Tag-Selection Criteria	
0x23	RFID_18K6CGetSelectCriteria	Getting Tag-Selection Criteria	
0x24	RFID_18K6CSetSelectMaskData	Setting Tag-Selection Mask Data	
0x25	RFID_18K6CGetSelectMaskData	Getting Tag-Selection Mask Data	
0x26	RFID_18K6CSetPostMatchCriteria	Setting Post-Singulation Match Criteria	
0x27	RFID_18K6CGetPostMatchCriteria	Getting Post-Singulation Match Criteria	
0x28	RFID_18K6CSetPostMatchMaskData	Setting Post-Singulation Match Mask Data	
0x29	RFID_18K6CGetPostMatchMaskData	Getting Post-Singulation Match Mask Data	
ISO 18000-6C Tag-Access Parameters			
0x30	RFID_18K6CSetQueryTagGroup	Setting the Tags of Interest	
0x31	RFID_18K6CGetQueryTagGroup	Getting the Tags of Interest	
0x32	RFID_18K6CSetCurrentSingulation Algorithm	Setting the Current Singulation Algorithm	
0x33	RFID_18K6CGetCurrentSingulation Algorithm	Getting the Current Singulation Algorithm	

0x34	RFID_18K6CSetSingulationAlgorithm Parameters	Setting Singulation Algorithm Parameters	
0x35	RFID_18K6CGetSingulationAlgorithm Parameters	Getting Singulation Algorithm Parameters	
0x36	RFID_18K6CSetTagAccessPassword	Setting the Tag Access Password	
0x37	RFID_18K6CGetTagAccessPassword	Getting the Tag Access Password	
0x38	RFID_18K6CSetTagWriteDataBuffer	Setting Tag Writing Data Buffer	
0x39	RFID_18K6CGetTagWriteDataBuffer	Getting Tag Writing Data Buffer	
ISO 18000-6C Tag-Protocol Operation			
0x40	RFID_18K6CTagInventory	Tag Inventory Operation	
0x41	RFID_18K6CTagRead	Tag Read Operation	
0x42	RFID_18K6CTagWrite	Tag Write Operation	
0x43	RFID_18K6CTagKill	Tag Kill Operation	
0x44	RFID_18K6CTagLock	Tag Lock Operation	
0x45	RFID_18K6CTagMultipleWrite	Tag Multiple Write Operation	
0x46	RFID_18K6CTagBlockWrite	Tag Block Write Operation	
0x47	RFID_18K6CTagBlockErase	Tag Block Erase Operation	
RFID PCIe M.2 Module Control Operation			
0x50	RFID_ControlCancel	Canceling a Tag-Protocol Operation	
0x52	RFID_ControlPause	Pausing a Tag-Protocol Operation	
0x53	RFID_ControlResume	Resuming a Tag-Protocol Operation	
0x54	RFID_ControlSoftReset	Performing a Software Reset	
0x55	RFID_ControlResetToBootloader	Performing a Firmware Reset to Bootloader	
0x56	RFID_ControlSetPowerState	Setting the Power Management State	
0x57	RFID_ControlGetPowerState	Retrieving the Power Management State	
RFID PCIe M.2 Module Firmware Access			
0x60	RFID_MacGetFirmwareVersion	Retrieving the MAC Firmware Version Information	
0x61	RFID_MacGetDebug	Retrieving the MAC Firmware Debug Value	
0x62	RFID_MacClearError	Clearing a MAC Firmware Error	
0x63	RFID_MacGetError	Retrieving a MAC Firmware Error Code	
0x64	RFID_MacGetBootloaderVersion	Retrieving the Bootloader Firmware Version Information	
0x65	reserved	Reserved for Future Use	
0x66	RFID_MacWriteOemData	Writing MAC-Resident OEM Configuration Data	
0x67	RFID_MacReadOemData	Reading MAC-Resident OEM Configuration Data	
0x68	RFID_MacBypassWriteRegister	Writing to an Hardware Register	
0x69	RFID_MacBypassReadRegister	Reading from an Hardware Register	
0x6A	RFID_MacSetRegion	Setting the Region of Operation	
0x6B	RFID_MacGetRegion	Retrieving the Region of Operation	
0x6C	RFID_MacGetOEMCfgVersion	Retrieving the MAC-Resident OEMCfg Version Information	
0x6D	RFID_MacGetOEMCfgUpdateNumber	Retrieving the MAC-Resident OEMCfg Update Number Information	
RFID PCIe M.2 Module Region Test Support			
0x80	RFID_TestSetAntennaPortConfiguration	Setting the Test Antenna-Port Configuration	
0x81	RFID_TestGetAntennaPortConfiguration	Retrieving the Test Antenna-Port Configuration	
0x82	RFID_TestSetFrequencyConfiguration	Setting the Test Frequency Configuration	
0x83	RFID_TestGetFrequencyConfiguration	Retrieving the Test Frequency Configuration	
0x84	RFID_TestSetRandomDataPulseTime	Setting the Pulse Time of Random Data	

		Transmission	
0x85	RFID_TestGetRandomDataPulseTime	Retrieving the Pulse Time of Random Data Transmission	
0x86	RFID_TestSetInventoryConfiguration	Setting the Test Inventory Configuration	
0x87	RFID_TestGetInventoryConfiguration	Retrieving the Test Inventory Configuration	
0x88	RFID_TestTurnOnCarrierWave	Turning On the Radio CW	
0x89	RFID_TestTurnOffCarrierWave	Turning Off the Radio CW	
0x8A	RFID_TestInjectRandomData	Injecting a Random Data	
0x8B	RFID_TestTransmitRandomData	Transmitting a Random Data with Modulation	
RFID PCIe M.2 Module Engineering Test Support			
0x91	RFID_EngGetTemperature	Retrieving the Temperature	
0x93	RFID_EngGetRFPower	Retrieving the RF Power Level	

3.2 Packet Format Specification

3.2.1 Command Packet Format

The length of the command packet is fixed at 16 bytes.

Table 3.2 - Command Format Fields

Byte Offset	Name	Description
3:0	Header	Specific pilot header information. These hex values are 0x4D, 0x54, 0x49 and 0x43, i.e. ASCII string "MTIC".
4	DeviceID	PCIe M.2 module's device identification number. Default factory setting value is 0x00. General device ID number is 0xFF for broadcasting.
5	CommandID	See Table 3.1 - Command Set Summary.
13:6	Parameters	Effective parameters size of each command is different. The length of this field is fixed 8 bytes. Pad 0x00 is added to end of Parameters filed to force this field to stuff 8 bytes length.
15:14	Checksum	The checksum is CRC-16 calculated over the Header field to the Parameters field. Consult Section 8: Calculation of CRC-16.

3.2.2 Response Packet Format

The length of the response packet is fixed at 16 bytes.

Table 3.3 - Command Response Format Fields

Byte Offset	Name	Description
3:0	Header	Specific pilot header information. These hex values are 0x4D, 0x54, 0x49 and 0x52, i.e. ASCII string "MTIR".
4	DeviceID	PCIe M.2 module's device identification number. Default factory setting value is 0x00.
5	CommandID	See Table 3.1 - Command Set Summary.
13:6	ReturnedData	Effective returned data size of each command response is different. The length of this field is fixed 8 bytes. Pad 0x00 is added to end of ReturnedData filed to force this field to stuff 8 bytes length. Comprise status; for details, see Table 3.4 - Status Message Define.
15:14	Checksum	The checksum is CRC-16 calculated over the Header field to the ReturnedData field. Consult Section 8: Calculation of CRC-16.

Table 3.4 - Status Message Define

Status Code	Name	Description
0x00	RFID_STATUS_OK	Performed result of command is success.
0xF0	RFID_ERROR_INVALID_ PARAMETER	One or more parameter values in the command packet are invalid.
0xFF	RFID_ERROR_MODULE_ FAILURE	The underlying module encountered an error; MAC firmware error code set appropriately. The RFID PCIe M.2 module indicated a failure. A host may wish to query the MAC firmware error code, via RFID_MacGetError, to determine the reason for the PCIe M.2 module failure.

3.2.3 Report Packet Format

When performing ISO 18000-6C tag operations using the MTI RFID PCIe M.2 module the data is presented to the host as a sequence of packets.

At a generic level, a packet sequence consists of:

- A command-begin packet
- Zero or more inventory-response packets
  (with the appropriate tag-access packet)
- A command-end packet

Figure 3.1 presents a hierarchical view of the report packet data.

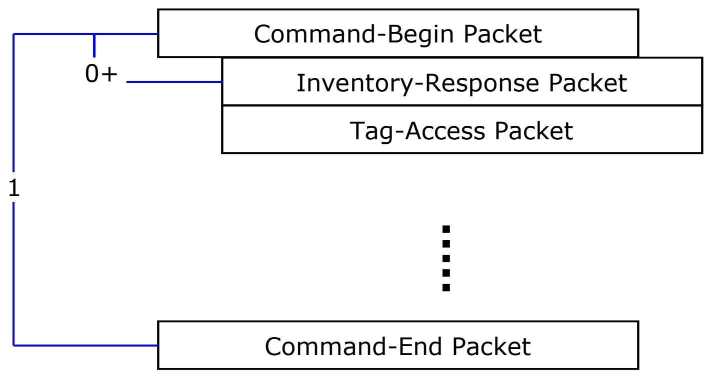

Figure 3.1 - Report Packet Hierarchy

There is a single command-begin/command-end packet pair. The command-begin packet is followed by zero or more inventory-response packets, one for each singulated tag. If the tag-protocol operation is something besides an inventory (i.e. read, write, etc.), then each ISO 18000-6C inventory-response packet is paired with the appropriate ISO 18000-6C tag-access packet.

When perform RFID_18K6CTagInventory command, first get the response packet of command. If the status of response packet is RFID_STATUS_OK, then get report packet that contains command-begin, inventory-response and command-end three packets.
When perform RFID_18K6CTagRead/Write/Kill/Lock/MultipleWrite/BlockWrite/BlockErase seven commands, first get the response packet of each command. If the status of response packet is RFID_STATUS_OK, then get the report packet that contains command-begin, inventory-response, tag-access and command-end four packets.

After get the response packet of command, getting the command-begin packet normally takes a few hundred milliseconds, but, depending upon the state of the RFID PCIe M.2 module, may take several seconds.

3.2.3.1 Report Sequence Behavior on Error Detection

3.2.3.1.1 Critical Error

If a critical (non-recoverable) error occurs during processing, a valid end of packet sequence with an appropriate error indicator still terminates the packet sequence. However, the explicitly-enabled "end" packets are not included in the packet sequence. A host must be prepared to receive a command-end packet at any time.

3.2.3.1.2 Non-Critical Error

A non-critical fault may occur while a tag access operation is executing. A non-critical fault does not cause the tag-protocol operation to abort; instead, the tag-protocol operation continues until one of the following events occurs:

- A non-recoverable error occurs
- The tag-protocol operation completes
- The host cancels the tag-protocol operation

3.2.3.2 Report Packet Format Specification

This section specifies the format of the packets. Note that when a 16- or 32-bit integer appears in a packet, it is represented in the packet format as **little endian**. Therefore, a host programmer needs to be aware that, depending upon the architecture upon which the packet-consumption will run, byte swapping of 16- and 32-bit values may be necessary.

3.2.3.2.1 Byte Ordering

The MTI MAC firmware will use little endian byte ordering in multi-byte words for all MAC firmware generated wire data

- The header, parameters and checksum three fields of command packet
- The header, returned data and checksum three fields of response packet
- All fields of four report packets
- OEM configuration data

Assuming B is the first byte transmitted across the wire; the following table describes Little Endian formatting using byte offsets from B.

Note that this Endianess does NOT apply to:

- Tag memory data. Tag memory data will be returned to the host in the format detailed in the inv_data field of inventory-response packet and acc_data field of tag-access packet for the given tag technology.

Table 3.5 - Little Endian Byte Order mapped to 32 bits

<table>
<tr><th>31 24</th><th>23 16</th><th>15 8</th><th>7 0</th></tr>
<tr><td colspan="4">32 bit word
(B)</td></tr>
<tr><td colspan="2">16 bit word 1
(B + 2)</td><td colspan="2">16 bit word 0
(B)</td></tr>
<tr><td>Byte 3
(B + 3)</td><td>Byte 2
(B + 2)</td><td>Byte 1
(B + 1)</td><td>Byte 0
(B)</td></tr>
</table>

3.2.3.2.2 Common Packet Format

All packets have a set of common fields that appear as the packet preamble except for the pkt_checksum field. These allow the receiver to identify the type of packet quickly and handle it appropriately. In specific packet types, bold text is applied to the value in the Byte Offset column of the common fields to indicate they are part of the common fields.

Table 3.6 - Common Packet Format Fields

<table>
<tr><th>Byte Offset</th><th>Name</th><th>Description</th></tr>
<tr><td>3:0</td><td>pkt_header</td><td>Packet specific pilot header information.
The fixed length, in bytes, of this report packet; it is defined n bytes here.</td></tr>
<tr><td>4</td><td>pkt_relnumber</td><td>Packet specific total relation number.
When the length of information data from byte offset 14 to (n-3) is greater than (n-16) bytes in size, the information data should be split some relational report packets.

1 = Only one report packet.
>1 = Total number of relational report packets, the host must merge them.

For Command-Begin and Command-End both packets, these numbers are fixed values because their lengths of information data are fixed values.</td></tr>
<tr><td>5</td><td>pkt_relseq</td><td>Packet specific relation sequence number.
The initial value is equal to one with the first packet of new one or more relational report packets. Increase the number progressively. If the value of this field is equal to the value of pkt_relnumber field, one or more relational report packets are transmitted completely.</td></tr>
<tr><td>6</td><td>rpt_ver</td><td>Report specific version number.</td></tr>
<tr><td>7</td><td>rpt_flags</td><td>Report flags<table><tr><th>Bit</th><th>Value and Description</th></tr><tr><td>7:0</td><td>Reserved for specific report-type use. Values determined by the specific report type.</td></tr></table></td></tr>
<tr><td>9:8</td><td>rpt_type</td><td>Report type identifier.
0x0001 - 0xFFFF are valid - although not all values have packets assigned to them.</td></tr>
<tr><td>11:10</td><td>rpt_inflen</td><td>The valid length, in 32-bit words, of the information field.</td></tr>
<tr><td>13:12</td><td>rpt_seq</td><td>Report specific sequence number. Increase the number progressively.
The sequence number is the same value with identical information data.</td></tr>
<tr><td>n-3:14</td><td>information</td><td>See Table 3.7 - Command-Begin Packet Fields
Table 3.8 - Command-End Packet Fields
Table 3.9 - ISO 18000-6C Inventory-Response Packet Fields
Table 3.10 - ISO 18000-6C Tag-Access Packet Fields

If the length of actual data in this field of this packet is less than (n-16) bytes in size, pad 0x00 is added to end of this field to force this packet to stuff n bytes length.</td></tr>
<tr><td>n-1:
n-2</td><td>pkt_checksum</td><td>The checksum is CRC-16 calculated over the pkt_header field to the information field.
Consult Section 8: Calculation of CRC-16.</td></tr>
</table>

3.2.3.3 Report Packet Types

The following sections describe the specific packet types that will appear in packet sequences.

3.2.3.3.1 Command-Begin Packet

The command-begin packet indicates the start of a sequence of packets for an ISO 18000-6C tag-protocol operation (i.e. inventory, read, write, etc.). The type of command executed by the RFID PCIe M.2 module determines which data packets appear, and in what order they appear, between the command-begin/end packet pair.

Table 3.7 - Command-Begin Packet Fields

<table>
<tr><th>Byte Offset</th><th>Name</th><th>Description</th></tr>
<tr><td>3:0</td><td>pkt_header</td><td>These hex values of header information are 0x4D, 0x54, 0x49 and 0x42, i.e. ASCII string “MTIB”.
The fixed length of this report packet is 24 bytes.</td></tr>
<tr><td>4</td><td>pkt_relnumber</td><td>Total relation number = 0x01</td></tr>
<tr><td>5</td><td>pkt_relseq</td><td>Relation sequence number = 0x01</td></tr>
<tr><td>6</td><td>rpt_ver</td><td>Report version number = 0x01</td></tr>
<tr><td>7</td><td>rpt_flags</td><td>Report flags:
<table>
<tr><th>Bit</th><th>Value and Description</th></tr>
<tr><td>0</td><td>Continuous mode flag:
0 = Operation not executed in continuous mode
1 = Operation executed in continuous mode</td></tr>
<tr><td>7:1</td><td>Reserved. Read as zero.</td></tr>
</table></td></tr>
<tr><td>9:8</td><td>rpt_type</td><td>Report type value = 0x0000</td></tr>
<tr><td>11:10</td><td>rpt_inflen</td><td>Information valid length = 0x0002</td></tr>
<tr><td>13:12</td><td>rpt_seq</td><td>Report sequence number = 0x0000</td></tr>
<tr><td>17:14</td><td>command</td><td>The command that initiated the report sequence:
0x0000000F - ISO 18000-6C Inventory
0x00000010 - ISO 18000-6C Read
0x00000011 - ISO 18000-6C Write
0x00000012 - ISO 18000-6C Lock
0x00000013 - ISO 18000-6C Kill
0x0000001E - ISO 18000-6C Block Erase
0x0000001F - ISO 18000-6C Block Write
0x00000022 - Region Test - Transmit Random Data</td></tr>
<tr><td>21:18</td><td>ms_ctr</td><td>MTI MAC firmware millisecond counter when the operation started.</td></tr>
<tr><td>23:22</td><td>pkt_checksum</td><td>The checksum is CRC-16 calculated over the pkt_header field to the ms_ctr field.
Consult Section 8: Calculation of CRC-16.</td></tr>
</table>

3.2.3.3.2 Command-End Packet

The command-end packet indicates the end of a sequence of packets for an ISO 18000-6C tag-protocol operation. A command-end packet is always used to terminate a packet sequence regardless of the fact that a tag-access operation is completed successfully or not.

Table 3.8 - Command-End Packet Fields

Byte Offset	Name	Description
3:0	pkt_header	These hex values of header information are 0x4D, 0x54, 0x49 and 0x45, i.e. ASCII string "MTIE". The fixed length of this report packet is 24 bytes.
4	pkt_relnumber	Total relation number = 0x01
5	pkt_relseq	Relation sequence number = 0x01
6	rpt_ver	Report version number = 0x01
7	rpt_flags	Report flags: **Bit** — **Value and Description** 7:0 — Reserved. Read as zero.
9:8	rpt_type	Report type value = 0x0001
11:10	rpt_inflen	Information valid length = 0x0002
13:12	rpt_seq	Increase the report sequence number progressively.
17:14	ms_ctr	MTI MAC firmware millisecond counter when the operation ended.
21:18	status	The completion status of the operation. Values are: 0x00000000 - Success Consult Section 7: MTI MAC Firmware Error Codes for a list of possible error status codes and descriptions.
23:22	pkt_checksum	The checksum is CRC-16 calculated over the pkt_header field to the status field. Consult Section 8: Calculation of CRC-16.

3.2.3.3.3 ISO 18000-6C Inventory-Response Packet

The ISO 18000-6C inventory-response packet contains the data a tag backscatters during the tag-singulation phase. This packet is generated unconditionally for tag inventories and for ISO 18000-6C tag-access operations (i.e. read, write, etc.). Assuming a valid CRC, the data contains the PC+EPC+CRC16 received during the singulation of a tag.

Table 3.9 - ISO 18000-6C Inventory-Response Packet Fields

<table>
<tr><th>Byte Offset</th><th>Name</th><th>Description</th></tr>
<tr><td>3:0</td><td>pkt_header</td><td>These hex values of header information are 0x4D, 0x54, 0x49 and 0x49, i.e. ASCII string “MTII”.
The fixed length of this report packet is 64 bytes.</td></tr>
<tr><td>4</td><td>pkt_relnumber</td><td>Total relation number = variable</td></tr>
<tr><td>5</td><td>pkt_relseq</td><td>Relation sequence number = variable</td></tr>
<tr><td>6</td><td>rpt_ver</td><td>Report version number = 0x01</td></tr>
<tr><td>7</td><td>rpt_flags</td><td>Report flags:
<table>
<tr><th>Bit</th><th>Value and Description</th></tr>
<tr><td>0</td><td>CRC invalid flag for backscattered tag data:
0 = Valid CRC
1 = Invalid CRC</td></tr>
<tr><td>1</td><td>Transceiver chip:
0 = Indy R1000 chip
1 = Indy R2000 chip</td></tr>
<tr><td>2</td><td>Serialized TID data:
Reserved. Read as zero.</td></tr>
<tr><td>3</td><td>Extra hardware data:
0 = No extra hardware data in the front of inv_data
1 = Extra hardware data included (8 bytes)</td></tr>
<tr><td>5:4</td><td>Reserved. Read as zero.</td></tr>
<tr><td>7:6</td><td>Tag-data padding:
Number of padding bytes added to inv_data force the length of inv_data field to end on the 32-bit boundary.</td></tr>
</table></td></tr>
<tr><td>9:8</td><td>rpt_type</td><td>Report type value = 0x0005</td></tr>
<tr><td>11:10</td><td>rpt_inflen</td><td>Information valid length = variable (greater than or equal to 3)

When pkt_relnumber = 1, the length of this field in words = (hardware data bytes + tag data bytes + tag-data padding bytes) / 4.

The information data consists of hardware data, tag data and tag-data padding three parts. When pkt_relnumber = 1, each length of three parts is as follows:
● The length of hardware data in bytes is 12 from byte offset 14 to 25.
● The length of tag data in bytes is depending on bytes number of tag data.
● The length of tag-data padding in bytes is depending on bytes number of tag-data padding of rpt_flags field.

For other details, see note 1.</td></tr>
<tr><td>13:12</td><td>rpt_seq</td><td>Increase the report sequence number progressively.</td></tr>
<tr><td>17:14</td><td>ms_ctr</td><td>MTI MAC firmware millisecond counter when tag was inventoried.</td></tr>
<tr><td>18</td><td>nb_rssi</td><td>The narrowband receive signal strength indicator (RSSI).
This is the backscattered tag signal. The narrowband RSSI indication is 8-bit value. It is useful for relative signal strength indication. It is important to note that the IF LNA gain in the receive path can vary each time carrier wave is turned on, so the IF LNA gain should be taken into account. Refer to byte offsets 21:20 for a description of the ana_ctrl field, which includes the setting of the IF LNA at the time the RSSI measurement was taken.

Value conversion to dB formula:</td></tr>
</table>

<table>
<tr><td></td><td></td><td>Exponent = bits[7:3], Mantissa = bits[2:0], Mantissa_Size = 3
20 * log10 (2^Exponent * (1 + Mantissa / 2^Mantissa_Size))

Example: Value 0x48
Exponent = 9, Mantissa = 0
20 * log (2^9 * (1 + 0 / 2^3)) = 54.19</td></tr>
<tr><td>19</td><td>wb_rssi</td><td>The wideband receive signal strength indicator (RSSI).
This is the backscattered tag signal. The wide-band RSSI indication is 8-bit value. It is useful for relative signal strength indication. It is important to note that the IF LNA gain in the receive path can vary each time carrier wave is turned on, so the IF LNA gain should be taken into account. Refer to byte offsets 21:20 for a description of the ana_ctrl field, which includes the setting of the IF LNA at the time the RSSI measurement was taken.

Value conversion to dB formula:
Exponent = bits[7:4], Mantissa = bits[3:0], Mantissa_Size = 4
20 * log10 (2^Exponent * (1 + Mantissa / 2^Mantissa_Size))

Example: Value 0x48
Exponent = 4, Mantissa = 8
20 * log (2^4 * (1 + 8 / 2^4)) = 27.60</td></tr>
<tr><td>21:20</td><td>ana_ctrl</td><td>The value of the Indy R1000 or R2000 gain control register at time the RSSI measurement was taken - contains the IF LNA’s gain info for RSSI. See the value of Transceiver chip bit of rpt_flags field for format.
Bits[5:4]: IF LNA gain with Indy R1000 chip (0 = 24dB, 1 = 18dB, 3 = 12dB)
Bits[5:3]: IF LNA gain with Indy R2000 chip (0 = 24dB, 1 = 18dB, 3 = 12dB, 7 = 6dB)
Other bits are reserved for future use.</td></tr>
<tr><td>23:22</td><td>rssi</td><td>The EPC receive signal strength indicator (RSSI). The value is the narrowband RSSI adjusted by the calibration value. The units are tenths of dBm.</td></tr>
<tr><td>25:24</td><td>logic_ant</td><td>The value is the current logical antenna port during the tag-singulation phase.</td></tr>
<tr><td>x:26</td><td>inv_data</td><td>The data that was backscattered by the tag (i.e. PC + EPC + CRC16) during tag singulation. The data is presented in the same format as it is transmitted over the air from the tag to the MTI RFID PCIe M.2 module - i.e. the data has not been changed to match the endianness of the host processor.

These extra hardware data, if available via command, will lead the tag data, as indicated by the Extra hardware data bit of rpt_flags field.
These information of extra hardware data are as follows:
<table>
<tr><th>Byte</th><th>Value and Description</th></tr>
<tr><td>26</td><td>Physical antenna port:
The value is the current physical antenna port during the tag-singulation phase.</td></tr>
<tr><td>27</td><td>Phase:
The phase data bits[6:0] represents two’s complement value from -64 to 63 at the time the EPC is received. The bit 7 is reserved and always 0.

Value conversion to degrees formula:
bits[6:0] / 128 * 360

Value conversion to radian formula:
bits[6:0] / 128 * 2π

Example: Value 0x40 (= -64)
-64 / 128 * 360 = -180 (deg)
-64 / 128 * 2π = -3.142 (rad)

When the value of Transceiver chip bit of rpt_flags field = 0 (Indy R1000 chip), phase is not available and read as zero.</td></tr>
<tr><td>28</td><td>Temperature:
The value is specified in units of degree-C and a two's complement representation.</td></tr>
</table></td></tr>
</table>

		<table><tr><td>29</td><td>Reserved. Read as zero.</td></tr><tr><td>33:30</td><td>Frequency: The value is specified in units of kHz.</td></tr></table>
61:x+1	padding	If the length of actual data in the *information* field of this packet is less than 48 (64-16) bytes in size, pad 0x00 is added to end of *information* field to force this packet to stuff 64 bytes length.
63:62	pkt_checksum	The checksum is CRC-16 calculated over the pkt_header field to the padding field. Consult Section 8: Calculation of CRC-16.

Note:

1. The information data consists of hardware data, tag data and tag-data padding three parts.
   When pkt_relnumber > 1, each length of three parts is as follows:

   - The hardware data should be only appeared in the first packet with pkt_relseq = 1.
     The length of hardware data in bytes is 12 from byte offset 14 to 25.
   - When pkt_relseq = 1, the start of tag data is byte offset 26, and the maximum length in bytes is 36 from byte offset 26 to 61.
     When pkt_relseq > 1, the start of tag data is byte offset 14, and the maximum length in bytes is 48 from byte offset 14 to 61.
   - The tag-data padding is optional field which should be only appeared in the last packet with pkt_relseq = pkt_relnumber.
     The length of tag-data padding in bytes is depending on bytes number of tag-data padding of rpt_flags field.

3.2.3.3.4 ISO 18000-6C Tag-Access Packet

The ISO 18000-6C tag-access packet indicates the result of the tag-access command upon the ISO 18000-6C tag. Valid tag access commands are as follows:

- Read
- Write (Multiple Write)
- Kill
- Lock
- Block Write
- Block Erase

If a tag operation is simply limited to an inventory operation, ISO 18000-6C tag operation packets will not appear in the packet sequence.

Table 3.10 - ISO 18000-6C Tag-Access Packet Fields

<table>
<tr><th>Byte Offset</th><th>Name</th><th>Description</th></tr>
<tr><td>3:0</td><td>pkt_header</td><td>These hex values of header information are 0x4D, 0x54, 0x49 and 0x41, i.e. ASCII string “MTIA”.
The fixed length of this report packet is 64 bytes.</td></tr>
<tr><td>4</td><td>pkt_relnumber</td><td>Total relation number = variable</td></tr>
<tr><td>5</td><td>pkt_relseq</td><td>Relation sequence number = variable</td></tr>
<tr><td>6</td><td>rpt_ver</td><td>Report version number = 0x01</td></tr>
<tr><td>7</td><td>rpt_flags</td><td>Report flags:
<table>
<tr><th>Bit</th><th>Value and Description</th></tr>
<tr><td>0</td><td>Module access error flag:
0 = RFID PCIe M.2 module did not detect an error.
1 = RFID PCIe M.2 module detected an error. See the module_error_code field.</td></tr>
<tr><td>1</td><td>Tag backscatter error flag:
0 = Tag did not backscatter an error.
1 = Tag backscattered an error. See the tag_error_code field.</td></tr>
<tr><td>5:2</td><td>Reserved. Read as zero.</td></tr>
<tr><td>7:6</td><td>Tag-data padding:
Number of padding bytes added to acc_data force the length of acc_data field to end on 32-bit boundary.</td></tr>
</table></td></tr>
<tr><td>9:8</td><td>rpt_type</td><td>Report type value = 0x0006</td></tr>
<tr><td>11:10</td><td>rpt_inflen</td><td>Information valid length = variable (greater than or equal to 3)

When pkt_relnumber = 1, the length of this field in words =
(hardware data bytes + tag data bytes + tag-data padding bytes) / 4.

The information data consists of hardware data, tag data and tag-data padding three parts. When pkt_relnumber = 1, each length of three parts is as follows:
• The length of hardware data in bytes is 12 from byte offset 14 to 25.
• The length of tag data in bytes is depending on bytes number of tag data.
• The length of tag-data padding in bytes is depending on bytes number of tag-data padding of rpt_flags field.

For other details, see note 1 of Section ISO 18000-6C Inventory-Response Packet.</td></tr>
<tr><td>13:12</td><td>rpt_seq</td><td>Increase the report sequence number progressively.</td></tr>
<tr><td>17:14</td><td>ms_ctr</td><td>MTI MAC firmware millisecond counter when tag-access operation occurred.</td></tr>
<tr><td>18</td><td>command</td><td>ISO 18000-6C access command:

0xC2 - Read</td></tr>
</table>

		0xC3 - Write 0xC4 - Kill 0xC5 - Lock 0xC6 - Access 0xC7 - Block Write 0xC8 - Block Erase
19	tag_error_code	If the tag backscattered an error (i.e. the tag backscatter error flag of rpt_flags field is set), this value is the error code that the tag backscattered. Values are: 0x00 - general error (catch-all for errors not covered by codes) 0x03 - specified memory location does not exist or the PC value is not supported by the tag 0x04 - specified memory location is locked and/or permalocked and is not writeable 0x0B - tag has insufficient power to perform the memory write 0x0F - tag does not support error-specific codes
21:20	module_error_code	If the RFID PCIe M.2 module detects an error (i.e. the module access error flag of rpt_flags field is set), and none of the error specific bits are set in the rpt_flags field, this field contains a 16-bit error code. Values are: 0x0000 = no error 0x0001 = handle mismatch 0x0002 = CRC error on tag response 0x0003 = no tag reply 0x0004 = invalid password 0x0005 = zero kill password 0x0006 = tag lost 0x0007 = command format error 0x0008 = read count invalid 0x0009 = out of retries 0xFFFF = operation failed
23:22	write_word_count	The number of individual words successfully written.
25:24	reserved	Reserved. Read as zero.
x:26	acc_data	If there were no errors, and the ISO 18000-6C tag-access operation was a read (i.e. the command field contains 0xC2), this field contains the data that was read from the specified tag memory bank. The data is presented in the same format as it is transmitted over the air from the tag to the RFID PCIe M.2 module. The data should be treated as a sequence of bytes and is presented in the same format as it is transmitted over the air from the tag to the RFID PCIe M.2 module - i.e. the data has not been changed to match the endianness of the host or RFID PCIe M.2 module processor. If the ISO 18000-6C tag-access operation was any other access command, then this field will not be present.
61:x+1	padding	If the length of actual data in the *information* field of this packet is less than 48 (64-16) bytes in size, pad 0x00 is added to end of *information* field to force this packet to stuff 64 bytes length.
63:62	pkt_checksum	The checksum is CRC-16 calculated over the pkt_header field to the padding field. Consult Section 8: Calculation of CRC-16.

4 Command Introduction

4.1 RFID PCIe M.2 Module Configuration

4.1.1 RFID PCIe M.2 Module Device ID

4.1.1.1 Setting the Device ID

Description: Sets the device identification number of the RFID PCIe M.2 module.

Note: The RFID PCIe M.2 module immediately uses the new device identification number for next communication and saves it to the MAC firmware's OEM configuration data area. After the RFID PCIe M.2 module is re-powered on, the host will still use the new device identification number for communication.

Command: **ID: 0x00 / RFID_RadioSetDeviceID**

Parameters:

Byte Offset	Name	Value	Description
0	NewDeviceID	0x00 ~ 0xFE	The device identification number for the RFID PCIe M.2 module. Default factory setting value is 0x00. General device ID number is 0xFF for broadcasting and does not set by host.
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.1.1.2 Retrieving the Device ID

Description: Retrieves the device identification number of the RFID PCIe M.2 module.

Command: **ID: 0x01 / RFID_RadioGetDeviceID**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	CurrentDevice ID	0x00 ~ 0xFE	Receive the device identification number.
7:2	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.1.2 Configuring the Operation Mode

The MTI RFID PCIe M.2 module may operate either in continuous or non-continuous mode. In continuous mode, when a tag-protocol-operation cycle (i.e. one iteration through all enabled antenna ports) has completed, the RFID PCIe M.2 module begins a new tag-protocol-operation cycle with the first enabled antenna port and continues to do so until the operation is explicitly cancelled by the host. In non-continuous mode, only a single tag-protocol-operation cycle is executed upon the RFID PCIe M.2 module.

4.1.2.1 Setting the Operation Mode

Description: Sets the operation mode of the RFID PCIe M.2 module. An RFID PCIe M.2 module's operation mode remains in effect until it is explicitly changed via RFID_RadioSetOperationMode. The operation mode may not be changed while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x02 / RFID_RadioSetOperationMode**

Parameters:

Byte Offset	Name	Value	Description
0	Mode	0, 1	The operation mode for the RFID PCIe M.2 module. 0 = continuous 1 = non-continuous
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.1.2.2 Retrieving the Operation Mode

Description: Retrieves the operation mode for the RFID PCIe M.2 module. The operation mode cannot be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x03 / RFID_RadioGetOperationMode**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	Mode	0, 1	Receive the current operation mode. 0 = continuous 1 = non-continuous
7:2	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.1.3 RFID PCIe M.2 Module Link Profiles

The MTI RFID PCIeM.2 module supports one or more link profiles (see Table 4.1), with one link profile being active at any given time. A link profile is a configuration that specifies operation parameters such as radio modulation type, Tari, tag protocol (i.e. ISO 18000-6C), etc. The exact link profile supported by the RFID PCIe M.2 module is beyond the scope of this document.

4.1.3.1 Setting the Current Link Profile

Description: Allows the host to set the current link profile for the RFID PCIe M.2 module. A link profile will remain in effect until changed by a subsequent call to RFID_RadioSetCurrentLinkProfile. The current link profile cannot be set while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x04 / RFID_RadioSetCurrentLinkProfile**

Parameters:

Byte Offset	Name	Value	Description
0	Profile	0 ~ 3	The link profile to make the current link profile. The MAC firmware uses profile 1 as the default.
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

Table 4.1 - Operation Parameters of Link Profiles

Profile Index	0	1	2	3
Modulation Type	DSB-ASK	PR-ASK	PR-ASK	DSB-ASK
Tari Duration (us)	25	25	25	6.25
Data01 Difference	1	0.5	0.5	0.5
Pulse Width (us)	12.5	12.5	12.5	3.13
R-T Calculation (us)	75	62.5	62.5	15.63
T-R Calculation (us)	200	85.33	71.11	20
Divide Ratio	8	21.33	21.33	8
Data Encoding	FM0	Miller-4	Miller-4	FM0
Pilot Tone	1	1	1	1
Link Frequency (kHz)	40	250	300	400
Data Rate (kbps)	40	62.5	75	400

4.1.3.2 Retrieving the Current Link Profile

Description: Allows the host to retrieve the current link profile for the RFID PCIe M.2 module. The current link profile cannot be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x05 / RFID_RadioGetCurrentLinkProfile**

Parameters:

Byte	Name	Value	Description

Offset			
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	CurrentProfile	0 ~ 3	Receive the current link profile.
7:2	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.1.4 Accessing Low-Level MAC Registers

The MTI RFID PCIe M.2 module allows a host to exercise fine-grained control over the operation of an RFID PCIe M.2 module by exposing the ability to set and retrieve low-level RFID-PCIe M.2-module registers (i.e. MAC virtual registers). The parameters that may be read/written by this mechanism are typically, but not always, set or retrieved once, when a host first obtains exclusive access to a PCIe M.2 module. Examples are wave-shaping tables, frequency-hopping tables and policies etc. The group of low-level parameters exposed is limited to those that do not expose the potential for causing the MTI RFID PCIe M.2 module’s state to become out of synch with the PCIe M.2 module’s state - attempts to configure these parameters via the low-level mechanism are blocked.

Being able to query the MAC’s virtual register that represents the error code for the last operation is useful. For example, when a particular command is called, the host can query the MAC’s error code virtual register to determine the exact cause of the PCIe M.2 module failure.

4.1.4.1 Setting Low-Level MAC Registers

Description: Allows a host to set an RFID PCIe M.2 module’s low-level register (i.e. MAC virtual register). RFID PCIe M.2 module registers may not be set while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x06 / RFID_RadioWriteRegister**

Parameters:

Byte Offset	Name	Value	Description
1:0	Address	0x0000 ~ 0xFFFF	The address of the MAC virtual register to be set.
5:2	Value	0x00000000 ~ 0xFFFFFFFF	The value to which the register is to be set.
7:6	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.1.4.2 Retrieving Low-Level MAC Registers

Description: Allows the host to retrieve the value for an RFID PCIe M.2 module’s low-level register (i.e. MAC virtual register). RFID PCIe M.2 module registers may not be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x07 / RFID_RadioReadRegister**
Parameters:

Byte Offset	Name	Value	Description
1:0	Address	0x0000 ~ 0xFFFF	The address of the MAC virtual register that will be retrieved.
7:2	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
4:1	Value	0x00000000 ~ 0xFFFFFFFF	Receive the value of the RFID PCIe M.2 module register specified.
7:5	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.1.5 Accessing Low-Level MAC Banked Registers

In addition to the individual register access described above, the MTI RFID PCIe M.2 module provides an interface to read/write the RFID PCIe M.2 module banked registers which utilize a selector register to identify which register bank is to be accessed. RFID PCIe M.2 module banked register addresses are unique within the bank but common across the set of valid bank selector values.

4.1.5.1 Setting Low-Level MAC Banked Registers

Description: Allows a host to set an RFID PCIe M.2 module's low-level banked register (i.e. banked MAC virtual register). RFID PCIe M.2 module banked registers may not be set while an RFID PCIe M.2 module is executing a tag-protocol operation. Note that the bank select register is set to the BankSelector value and is not restored.

Command: **ID: 0x08 / RFID_RadioWriteBankedRegister**
Parameters:

Byte Offset	Name	Value	Description
1:0	Address	0x0000 ~ 0xFFFF	The address of the banked MAC virtual register to be set.
3:2	BankSelector	0x0000 ~ 0xFFFF	The bank selector value to write to the corresponding bank select register for the specified address.
7:4	Value	0x00000000 ~ 0xFFFFFFFF	The value to which the banked register is to be set.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.1.5.2 Retrieving Low-Level MAC Banked Registers

Description: Allows the host to retrieve the value for an RFID PCIe M.2 module's low-level banked register (i.e. banked MAC virtual register). RFID PCIe M.2 module banked registers may not be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation. Note that the bank select register is set to the BankSelector value and is not restored.

Command: **ID: 0x09 / RFID_RadioReadBankedRegister**

Parameters:

Byte Offset	Name	Value	Description
1:0	Address	0x0000 ~ 0xFFFF	The address of the banked MAC virtual register that will be retrieved.
3:2	BankSelector	0x0000 ~ 0xFFFF	The bank selector value to write to the corresponding bank select register for the specified address.
7:4	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
4:1	Value	0x00000000 ~ 0xFFFFFFFF	Receive the value of the RFID PCIe M.2 module banked register specified.
7:5	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.1.6 Retrieving Low-Level MAC Register Information

The MTI RFID PCIe M.2 module also provides a command to retrieve information about a particular RFID PCIe M.2 module register. The information available pertains to register access (read only, read write, write only), register type (normal, banked, selector, reserved), and if it is a banked or selector register, with additional details as appropriate.

Description: Allows the host to retrieve the information for an RFID PCIe M.2 module's low-level register (i.e. MAC virtual register). RFID PCIe M.2 module register information may not be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x0A / RFID_RadioReadRegisterInfo**

Parameters:

Byte Offset	Name	Value	Description
1:0	Address	0x0000 ~ 0xFFFF	The address of the MAC virtual register for which information will be retrieved.
7:2	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
1	RegisterType	0 ~ 3	The type of the register requested. 0 = Normal - a standard MAC register

			1 = Banked - a banked MAC register, subordinate to corresponding selector register value 2 = Selector - a selector MAC register for selecting a specific bank of banked MAC registers 3 = Reserved - a reserved MAC register; should not be accessed
2	AccessType	0 ~ 3	The access type of the register requested. 0 = Read/Write access 1 = Write only access 2 = Read only access 3 = Reserved
3	BankSize	0x00 ~ 0xFF	The size of the register bank (i.e. the number of registers per bank selector value) associated with the requested register. This value is only valid for registers whose type is banked or selector.
5:4	SelectorAddress	0x0000 ~ 0xFFFF	The address of the select register for the requested register. This value is only valid for registers whose type is banked.
7:6	CurrentSelector	0x0000 ~ 0xFFFF	The current value of the select register associated with the requested register. This value is only valid for registers whose type is banked or selector.

4.1.7 Controlling the Inventory-Response Packet Format

The MTI RFID PCIe M.2 module supports one or more formats of inventory-response packet to send out specific information.

4.1.7.1 Setting the Inventory-Response Packet Format

Description: Sets the format of inventory-response packet. The packet format cannot be set while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x0E / RFID_RadioSetInventoryPacketFormat**

Parameters:

Byte Offset	Name	Value	Description
0	PacketFormat	0, 1	The supportable state of extra hardware data of inventory-response packet. The length of extra hardware data is fixed 8 bytes. 0 = disabled 1 = enabled
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.1.7.2 Retrieving the Inventory-Response Packet Format

Description: Retrieves the format of inventory-response packet. The packet format cannot be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x0F / RFID_RadioGetInventoryPacketFormat**
Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
1	PacketFormat	0, 1	Receive the supportable state for extra hardware data of inventory-response packet. 0 = disabled 1 = enabled
7:2	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.2 Antenna Port Configuration

The RFID PCIe M.2 module supports active use of one or more logical antenna ports, each mapped to a physical transmit and a physical receive antenna port. These ports are single, combined Tx/Rx antenna. The number of logical and physical antenna ports supported is determined by the MAC firmware that is executing on the RFID PCIe M.2 module. A host may, on a per-logical-antenna-port basis, retrieve status and configure several parameters. These include:

- **Enabled/Disabled Status** - indicates if an antenna port is to be used for tag-protocol operations. If an antenna port is enabled, then it is included in the antenna ports that are used during a tag-protocol-operation cycle - i.e. one iteration through all enabled antenna ports, where one iteration is dependent upon the dwell time, as well as upon the maximum number of inventory cycles on a particular antenna port (see Table 4.2). If an antenna port is disabled, then it is skipped during a tag-protocol operation.
- **Power Level** - the amount of power to be supplied to a transmit antenna port when it is being used for tag-protocol operations.
- **Dwell Time** - this specifies the maximum amount of time that should be spent on an antenna port during a tag-protocol-operation cycle. During a tag-protocol-operation cycle, the amount of time spent on an antenna port may be less than the dwell time - for example, if the maximum number of inventory rounds on that particular antenna port has been performed before the dwell time has transpired, the tag-protocol-operation cycle continues with a new inventory round on the next enabled antenna port.
- **Number of Inventory Cycles** - the maximum number of inventory cycles that are performed on an antenna port during a tag-protocol-operation cycle. An inventory cycle may consist of one or more inventory rounds as defined in *ISO-IEC_CD 18000-6C* (*Information Technology - Radio-frequency identification for item management - Part 6C: Parameters for air interference communications at 860 MHz to 960 MHz*) for a particular inventory-session target (i.e. A or B) and as configured by the singulation algorithm parameters specified. During a tag-protocol-operation cycle, the number of inventory cycles performed on an antenna port may be less than the maximum number of inventory cycles - for example, if the antenna-port dwell time has transpired before the maximum number of inventory cycles has been performed, the current inventory cycle is terminated and the tag-protocol-operation cycle begins a new inventory cycle on the next enabled antenna.
- **Logical-to-Physical Antenna Port Mapping** - indicates which physical transmit and receive antenna ports the logical antenna port represents.

4.2.1 Antenna-Port State

The host may specify which of an RFID PCIe M.2 module's logical antenna ports are to be enabled for tag operations.

Note: When performing any non-inventory ISO 18000-6C tag access operation (i.e. read, write, kill, lock, blockWrite, or blockErase), the RFID PCIe M.2 module will only use the first enabled logical antenna (i.e. the enabled logical antenna with the smallest logical antenna port number) and ignores the antenna dwell time and inventory-cycle count.

4.2.1.1 Setting Antenna-Port State

Description: Allows a host to specify whether or not an RFID PCIe M.2 module's logical antenna port is enabled for subsequent tag operations. The antenna-port state cannot be set while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x10 / RFID_AntennaPortSetState**

Parameters:

Byte Offset	Name	Value	Description
0	AntennaPort	0 ~ 15	The logical antenna port to enable or disable. Logical antenna ports are numbered beginning with zero.
1	State	0, 1	The new state of the logical antenna port. 0 = disabled 1 = enabled

7:2	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.2.1.2 Retrieving Antenna-Port State

Description: Retrieves the state of the requested logical antenna port for a particular RFID PCIe M.2 module. The antenna-port state cannot be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x11 / RFID_AntennaPortGetState**

Parameters:

Byte Offset	Name	Value	Description
0	AntennaPort	0 ~ 15	The logical antenna port for which to retrieve the antenna port state. Logical antenna ports are numbered beginning with zero.
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
1	State	0, 1	Receive the state for the logical antenna port requested. 0 = disabled 1 = enabled
5:2	AntennaSense Value	0x00000000 ~ 0x000FFFFF	The stored value from the last measurement of the antenna-sense resistor for the logical antenna port's physical transmit antenna port. The last measurement taken occurred the last time that the carrier wave was turned on for this antenna port. Note, this means that when retrieving the logical antenna port's state, this does not result in an active measurement of the antenna-sense resistor. This value is specified in ohms.
7:6	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.2.2 Antenna-Port Configuration

4.2.2.1 Setting Antenna-Port Configuration

Description: Allows a host to configure several parameters for a single logical antenna port - for example, dwell time, power level, and number of inventory cycles. Even if the logical antenna port is disabled, a host is allowed to set these configuration parameters. Setting configuration parameters does not cause a logical antenna port to be automatically enabled; the host must still enable the logical antenna port via RFID_AntennaPortSetState. The antenna-port configuration cannot be set while an RFID PCIe M.2 module is executing a tag-protocol operation.

Note: Since RFID_AntennaPortSetConfiguration sets all of the configuration parameters that are present in the parameters field; if a host wishes to leave some parameters unchanged, the host should first call RFID_AntennaPortGetConfiguration to retrieve the current settings, update the values in the parameters field that are to be changed, and then call RFID_AntennaPortSetConfiguration.

Command: **ID: 0x12 / RFID_AntennaPortSetConfiguration**

Parameters:

Byte Offset	Name	Value	Description
0	AntennaPort	0 ~ 15	The logical antenna port to configure. Logical antenna ports are numbered, beginning with zero.
2:1	PowerLevel	0 ~ 300	The power level for the logical antenna port's physical transmit antenna. This value is specified in 0.1 (i.e. 1/10th) dBm. **Note:** Not all RFID PCIe M.2 modules support setting an antenna port's power level at 1/10th dBm resolutions. The dBm rounding/truncation policy is left to the RFID PCIe M.2 module and is outside the scope of the MTI RFID PCIe M.2 module.
4:3	DwellTime	0x0000 ~ 0xFFFF	Specifies the maximum amount of time in milliseconds that may be spent on the logical antenna port during a tag-protocol-operation cycle before switching to the next enabled antenna port. A value of zero indicates that there is no maximum dwell time for this antenna port. If this parameter is zero, then NumberInventoryCycles may not be zero. See Table 4.2 for the effect of antenna-port dwell time and number of inventory cycles on the amount of time spent on an antenna port during a single tag-protocol-operation cycle. **Note:** When performing any non-inventory ISO 18000-6C tag-access operation (i.e. read, write, kill, lock, etc.), the RFID PCIe M.2 module ignores the dwell time for the antenna port which is used for the tag-protocol operation.
6:5	NumberInventoryCycles	0x0000 ~ 0xFFFF	Specifies the maximum number of inventory cycles to attempt on the antenna port during a tag-protocol-operation cycle before switching to the next enabled antenna port. An inventory cycle consists of one or more executions of the singulation algorithm for a particular inventory-session target (i.e. A or B). If the singulation algorithm parameters are configured to toggle the inventory-session, executing the singulation algorithm for inventory session A and inventory session B counts as two inventory cycles. A value of zero indicates that there is no maximum number of inventory cycles for this antenna port. If this parameter is zero, then DwellTime may not be zero. See Table 4.2 for the effect of antenna-port dwell time and number of inventory cycles on the amount of time spent on an antenna port during a single tag-protocol-operation cycle. **Note:** When performing any non-inventory ISO 18000-6C tag-access operation (i.e. read, write, kill, lock, etc.), the RFID PCIe M.2 module ignores the number of inventory cycles for the antenna port which is used for the tag-protocol operation.
7	PhysicalPort	0	The physical transmit/receive port that this logical antenna port is mapped to. 0 = ANT1

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.

7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

When performing tag-protocol operations, the dwell time and number of inventory cycles determine how long will be spent on a particular logical antenna port during the tag-protocol-operation cycle. Table 4.2 specifies the effect of setting a logical antenna port's dwell time and number of inventory cycles upon the amount of time spent upon a single antenna port.

Note: Table 4.2 does not apply if a non-inventory tag-protocol operation (i.e. read, write, kill, lock, etc.) is being performed. The RFID PCIe M.2 module executes the non-inventory tag-protocol operation for a single inventory cycle on the first enabled logical antenna (i.e. the enabled logical antenna with the smallest logical antenna port number).

Table 4.2 - Effect of Antenna Dwell Time and Number of Inventory Cycles on Amount of Time Spent on a Single Logical Antenna Port during a Single Tag-Protocol-Operation Cycle

Dwell Time	Number of Inventory Cycles	Result for a Single Tag-protocol-operation Cycle
0	0	Invalid combination. Results in an error.
0	1+	The logical antenna port is used until one of the following happens: ● The specified number of inventory cycles has been performed on the logical antenna port. ● The maximum number of tags has the tag-protocol operation applied to them. ● The tag-protocol operation is explicitly cancelled.
1+ ms	0	The logical antenna port is used until one of the following happens: ● The specified dwell time has expired. ● The maximum number of tags has the tag-protocol operation applied to them. ● The tag-protocol operation is explicitly cancelled.
1+ ms	1+	The logical antenna port is used until one of the following happens: ● The specified dwell time has expired. ● The specified number of inventory cycles has been performed on the logical antenna port. ● The maximum number of tags has the tag-protocol operation applied to them. ● The tag-protocol operation is explicitly cancelled.

4.2.2.2 Retrieving Antenna-Port Configuration

Description: Allows a host to retrieve a single logical antenna port's configuration parameters - for example, dwell time, power level, and number of inventory cycles. Even if the logical antenna port is disabled, a host is allowed to retrieve these configuration parameters. Retrieving configuration parameters does not cause a logical antenna port to be automatically enabled; the host must still enable the logical antenna port via RFID_AntennaPortSetState. The antenna-port configuration cannot be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x13 / RFID_AntennaPortGetConfiguration**

Parameters:

Byte Offset	Name	Value	Description
0	AntennaPort	0 ~ 15	The logical antenna port to configure. Logical antenna ports are numbered beginning with zero.
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
2:1	PowerLevel	0 ~ 300	The power level for the logical antenna port's physical transmit antenna. This value is specified in 0.1 (i.e. 1/10th) dBm. Detail sees description of parameters field of RFID_AntennaPortSetConfiguration.
4:3	DwellTime	0x0000 ~ 0xFFFF	Specifies the maximum amount of time in milliseconds that may be spent on the logical antenna port during a tag-protocol-operation cycle before switching to the next enabled antenna port. Detail sees description of parameters field of RFID_AntennaPortSetConfiguration.
6:5	NumberInventoryCycles	0x0000 ~ 0xFFFF	Specifies the maximum number of inventory cycles to attempt on the antenna port during a tag-protocol-operation cycle before switching to the next enabled antenna port. Detail sees description of parameters field of RFID_AntennaPortSetConfiguration.
7	PhysicalPort	0	The physical transmit/receive port that this logical antenna port is mapped to. 0 = ANT1

4.2.3 Global Antenna-Port Sense Threshold

4.2.3.1 Setting Global Antenna-Port Sense Threshold

Description: The antenna-sense resistance should be considered to be an open circuit (i.e. a disconnected antenna). If it is detected that the antenna-sense resistance is above the threshold, the carrier wave will not be turned on in order to protect the circuit.

Command: **ID: 0x14 / RFID_AntennaPortSetSenseThreshold**

Parameters:

Byte Offset	Name	Value	Description
3:0	AntennaSenseThreshold	0x00000000 ~ 0x000FFFFF	The measured resistance, specified in ohms, above which the antenna-sense resistance should be considered to be an open circuit (i.e. a disconnected antenna). If it is detected that the antenna-sense resistance is above the threshold, the carrier wave will not be turned on in order to protect the circuit. **Note:** This value, while appearing in the per-antenna configuration, is actually a system-wide setting in the current release. Changing it will result in the value being changed for all antennas. To prevent unintentionally changing this value for all antennas, it is best to first retrieve the antenna configuration for the antenna for which configuration will be changed, update the fields that should be changed and then set the configuration.
7:4	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.2.3.2 Retrieving Global Antenna-Port Sense Threshold

Description: Retrieves the threshold of the requested antenna-sense resistance for a particular RFID PCIe M.2 module. If it is detected that the antenna-sense resistance is above the threshold, the carrier wave will not be turned on in order to protect the circuit.

Command: **ID: 0x15 / RFID_AntennaPortGetSenseThreshold**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
4:1	AntennaSense Threshold	0x00000000 ~ 0x000FFFFF	Receive the value of antenna sense threshold. The measured resistance, specified in ohms.
7:5	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3 ISO 18000-6C Tag-Select Operation/Tag-Access Parameters

This section describes the functions through which a host can access ISO 18000-6C tags.

Tag accesses are comprised of three operations:

- The tag population is optionally partitioned to logically group tags of interest, i.e. one or more ISO 18000-6C select commands are performed. The MTI RFID PCIe M.2 module does not provide a host with the ability to directly issue ISO 18000-6C select commands. However, a host may specify tag-selection criteria that are used by the RFID PCIe M.2 module to perform the tag partitioning before tags are singulated.
- The tags of interest are singulated and (optionally) their EPCs are gathered, i.e. the ISO 18000-6C inventory commands (i.e. query, etc.) are performed. A host has control of several configuration parameters that control the operation of the inventory commands. After the MTI RFID PCIe M.2 module has singulated a tag, it can optionally apply an application-supplied post-singulation match mask to all or a portion of the singulated tag's EPC to further filter the singulated tag, f or the situation where a host simply wants to inventory the set (or some subset) of tags in the MTI RFID PCIe M.2 module's range.
- As tags are singulated, the ISO 18000-6C access command is (optionally) applied - this implies that only tags that match the optionally-supplied selection criteria have the access command applied to them. The RFID PCIe M.2 module applies the access command immediately after the tag is singulated (in contrast to gathering an inventory of all tags and then applying the access command to all inventoried tags). The MTI RFID PCIe M.2 module allows the host to explicitly issue the following ISO 18000-6C access commands: read, write, kill, lock, blockWrite, and blockErase. The access commands, which all perform an implicit inventory operation.
The ISO 18000-6C tag-protocol operation is described in the next section for detailed information.

4.3.1 Specifying Tag-Selection Criteria

A host may require that the tag population be logically partitioned into disjoint groups prior to issuing an inventory operation or access command. After the tags are partitioned, the specified operation may then be applied to one of the groups. Six pieces of information are used to partition a tag population into disjoint groups and to control which tags an operation is applied to. These pieces of information are explained in subsequent sections.

4.3.1.1 Setting Active Tag-Selection Criteria

Description: The number of selection criteria in the array indexed from by the CriteriaIndex field. This field must be greater than or equal to zero and less than or equal to seven. These active selection criteria are applied sequentially to the tag population before the tag-protocol operation.
Calling RFID_18K6CSetActiveSelectCriteria with all parameters set to zero results in disabling all selection criteria (i.e. even if the PerformSelect field of tag-protocol operation function is provided to the appropriate RFID_18K6CTag* function, no selections will be issued).

Command: **ID: 0x20 / RFID_18K6CSetActiveSelectCriteria**

Parameters:

Byte Offset	Name	Value	Description
0	CriteriaIndex	0 ~ 7	The selection criteria to enable or disable. All ActiveState fields of the CriteriaIndex from 0 to 7 set to zero results all selection criteria be disabled.
1	ActiveState	0, 1	The new active state of the selection criteria. 0 = disabled 1 = enabled
7:2	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description

0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.1.2 Getting Active Tag-Selection Criteria

Description: Retrieves the status of the indexed selection criteria for a particular RFID PCIe M.2 module.

Command: **ID: 0x21 / RFID_18K6CGetActiveSelectCriteria**

Parameters:

Byte Offset	Name	Value	Description
0	CriteriaIndex	0 ~ 7	The CriteriaIndex for which to retrieve the selection criteria status.
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
1	ActiveState	0, 1	Receive the status for the selection criteria indexed. 0 = disabled 1 = enabled
7:2	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.1.3 Setting Tag-Selection Criteria

Description: Configures the tag-selection criteria for the ISO 18000-6C select command. The supplied tag-selection criteria are used for any tag-protocol operations (i.e. RFID_18K6CTagInventory, etc.) in which the host specifies that an ISO 18000-6C select command should be issued prior to executing the tag-protocol operation (i.e. even if the PerformSelect field of tag-protocol operation function is provided to the appropriate RFID_18K6CTag* function). The tag-selection criteria stay in effect until the next call to RFID_18K6CSetSelectCriteria. Tag-selection criteria may not be changed while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x22 / RFID_18K6CSetSelectCriteria**

Parameters:

Byte Offset	Name	Value	Description
0	CriteriaIndex	0 ~ 7	The criteria index to configure the tag-selection criteria.
1	Bank	0 ~ 3	The memory bank that contains the bits that are compared against the bit pattern specified in the tag mask. For a tag mask, reserved memory (0) is not a valid value. 0 = Reserved memory (reserved for future use) 1 = EPC memory 2 = TID memory 3 = User memory
3:2	Offset	0x0000 ~ 0xFFFF	The offset, in bits, from the start of the memory bank, of the first bit that are matched against the tag mask. If offset falls beyond the end of the memory bank, the tag is considered non-matching.
4	Count	0 ~ 255	The number of bits in the tag mask. A length of zero causes all tags to match. If (Offset+Count) falls beyond the end of the memory bank, the tag is considered non-matching. Valid values are 0 to 255, inclusive.

<table>
<tr><td>5</td><td>Target</td><td>0 ~ 4</td><td>Specifies which flag, selected (i.e. SL) or one of the four inventory flags (i.e. S0, S1, S2, or S3) are modified by the Action field.

0 = S0 inventoried flag
1 = S1 inventoried flag
2 = S2 inventoried flag
3 = S3 inventoried flag
4 = Selected (SL) flag</td></tr>
<tr><td>6</td><td>Action</td><td>0 ~ 7</td><td>Specifies the action that will be applied to the tag populations (i.e. the matching and non-matching tags).
<table>
<tr><th>Action</th><th>Matching</th><th>Non-Matching</th></tr>
<tr><td>0</td><td>assert SL or inventoried -> A</td><td>deassert SL or inventoried -> B</td></tr>
<tr><td>1</td><td>assert SL or inventoried -> A</td><td>do nothing</td></tr>
<tr><td>2</td><td>do nothing</td><td>deassert SL or inventoried -> B</td></tr>
<tr><td>3</td><td>negate SL or (A -> B, B -> A)</td><td>do nothing</td></tr>
<tr><td>4</td><td>deassert SL or inventoried -> B</td><td>assert SL or inventoried -> A</td></tr>
<tr><td>5</td><td>deassert SL or inventoried -> B</td><td>do nothing</td></tr>
<tr><td>6</td><td>do nothing</td><td>assert SL or inventoried -> A</td></tr>
<tr><td>7</td><td>do nothing</td><td>negate SL or (A -> B, B -> A)</td></tr>
</table></td></tr>
<tr><td>7</td><td>Truncation</td><td>0</td><td>Specifies if, during singulation, a tag responds to a subsequent inventory operation with its entire Electronic Product Code (EPC) or only responds with the portion of the EPC that immediately follows the bit pattern (as long as the bit pattern falls within the EPC - if the bit pattern does not fall within the tag’s EPC, the tag ignores the tag partitioning operation). If this parameter is non-zero:
• Bank must be EPC memory.
• Target must be Selected (SL) flag.

This action must correspond to the last tag select operation issued before the inventory operation or access command.

This parameter is not supported in the current MTI MAC firmware release. This field must be set to zero.

0 = disabled
1 = enabled (reserved for future use)</td></tr>
</table>

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.1.4 Getting Tag-Selection Criteria

Description: Retrieves the configured tag-selection criteria for the ISO 18000-6C select command. The returned tag-selection criteria are used for any tag-protocol operations (i.e. RFID_18K6CTagInventory, etc.) in which the host specifies that an ISO 18000-6C select command should be issued prior to executing the tag-protocol operation. Tag-selection criteria may not be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x23 / RFID_18K6CGetSelectCriteria**

Parameters:

Byte Offset	Name	Value	Description
0	CriteriaIndex	0 ~ 7	Receive the configuration for the selection criteria indexed.
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

<table>
<tr><th>Byte Offset</th><th>Name</th><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Status</td><td>0x00/F0/FF</td><td>Performed result of command.</td></tr>
<tr><td>1</td><td>Bank</td><td>0 ~ 3</td><td>The memory bank that contains the bits that are compared against the bit pattern specified in the tag mask.
0 = Reserved memory (reserved for future use)
1 = EPC memory
2 = TID memory
3 = User memory</td></tr>
<tr><td>3:2</td><td>Offset</td><td>0x0000 ~ 0xFFFF</td><td>The offset, in bits, from the start of the memory bank, of the first bit that are matched against the tag mask.</td></tr>
<tr><td>4</td><td>Count</td><td>0 ~ 255</td><td>The number of bits in the tag mask. A length of zero causes all tags to match.</td></tr>
<tr><td>5</td><td>Target</td><td>0 ~ 4</td><td>0 = S0 inventoried flag
1 = S1 inventoried flag
2 = S2 inventoried flag
3 = S3 inventoried flag
4 = Selected (SL) flag</td></tr>
<tr><td>6</td><td>Action</td><td>0 ~ 7</td><td><table>
<tr><th>Action</th><th>Matching</th><th>Non-Matching</th></tr>
<tr><td>0</td><td>assert SL or inventoried -> A</td><td>deassert SL or inventoried -> B</td></tr>
<tr><td>1</td><td>assert SL or inventoried -> A</td><td>do nothing</td></tr>
<tr><td>2</td><td>do nothing</td><td>deassert SL or inventoried -> B</td></tr>
<tr><td>3</td><td>negate SL or (A -> B, B -> A)</td><td>do nothing</td></tr>
<tr><td>4</td><td>deassert SL or inventoried -> B</td><td>assert SL or inventoried -> A</td></tr>
<tr><td>5</td><td>deassert SL or inventoried -> B</td><td>do nothing</td></tr>
<tr><td>6</td><td>do nothing</td><td>assert SL or inventoried -> A</td></tr>
<tr><td>7</td><td>do nothing</td><td>negate SL or (A -> B, B -> A)</td></tr>
</table></td></tr>
<tr><td>7</td><td>Truncation</td><td>0</td><td>Reserved. Read as zero.
0 = disabled
1 = enabled (reserved for future use)</td></tr>
</table>

4.3.1.5 Setting Tag-Selection Mask Data

Description: The tag mask is used to specify a bit pattern that is used to match against one of a tag's memory banks

to determine if it is matching or non-matching. A tag mask is a combination of a memory bank and a sequence of bits that are matched at the specified offset within the chosen memory bank.
The MaskData is from 0 to 255 bits in length. MaskData, which is Count bits long, contains a bit string that a Tag compares against the memory location that begins at Offset and ends Count bits later.

Command: **ID: 0x24 / RFID_18K6CSetSelectMaskData**
Parameters:

Byte Offset	Name	Value	Description
0	CriteriaIndex	0 ~ 7	The criteria index to configure the tag-selection criteria.
1	MaskIndex	0 ~ 7	The mask data index to select the index number of mask data buffer.
2	MaskData0	0x00 ~ 0xFF	A buffer that contains a left-justified bit array that represents that bit pattern to match - i.e. the most significant bit of the bit array appears in the most-significant bit (i.e. bit 7) of the first byte of the buffer (i.e. mask[0]). All bits beyond count are ignored. For example, if the host wished to find tags with the following 12 bits 1000.1100.1101, starting at offset 16 in the EPC memory bank, then the fields would be set as follows: bank = EPC memory offset = 16 count = 12 mask[0] = 0x8C (1000.1100) mask[1] = 0xDX (1101.XXXX), where X is don't care When the MaskIndex is equal to 0, this field saves mask[0] data. When the MaskIndex is equal to 1, this field saves mask[4] data So the content of MaskData0 field is 0x8C.
3	MaskData1	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field saves mask[1] data. For the above example, the content of MaskData1 field is 0xDX.
4	MaskData2	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field saves mask[2] data.
5	MaskData3	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field saves mask[3] data.
7:6	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.1.6 Getting Tag-Selection Mask Data

Description: Receives the contents of mask data buffer.

Command: **ID: 0x25 / RFID_18K6CGetSelectMaskData**
Parameters:

Byte Offset	Name	Value	Description
0	CriteriaIndex	0 ~ 7	Receive the configuration for the selection criteria indexed.
1	MaskIndex	0 ~ 7	Receive the mask data for the mask data buffer indexed.
7:2	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte	Name	Value	Description

Offset			
0	Status	0x00/F0/FF	Performed result of command.
1	MaskData0	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field reads mask[0] data. When the MaskIndex is equal to 1, this field reads mask[4] data.
2	MaskData1	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field reads mask[1] data.
3	MaskData2	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field reads mask[2] data.
4	MaskData3	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field reads mask[3] data.
7:5	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.2 Specifying Post-Singulation Match Criteria

In addition to pre-singulation filtering (i.e. the ISO 18000-6C select command), a host also has the option of filtering tags, after they are singulated, based upon all or part of the tag's Electronic Product Code (EPC). This allows the host to control which tags will have a tag-protocol operation applied to them without having the PCIe M.2 module transmit potentially-sensitive Electronic Product Code data.

4.3.2.1 Setting Post-Singulation Match Criteria

Description: Configures the post-singulation match criteria to be used by the RFID PCIe M.2 module. The supplied post-singulation match criteria will be used for any tag-protocol operations (i.e. RFID_18K6CTagInventory, etc.) in which the host specifies that a post-singulation match must be performed on the tags that are singulated by the tag-protocol operation (i.e. the PerformPostMatch field of tag-protocol operation function is provided to the appropriate RFID_18K6CTag* function). The post-singulation match criteria stay in effect until the next call to RFID_18K6CSetPostMatchCriteria. Post-singulation match criteria may not be changed while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x26 / RFID_18K6CSetPostMatchCriteria**

Parameters:

Byte Offset	Name	Value	Description
0	Match	0, 1	Determines if the associated tag-protocol operation will be applied to tags that match the mask or not. 0 = not match 1 = match
2:1	Offset	0x0000 ~ 0x01FF	The offset in bits, from the start of the Electronic Product Code (EPC), of the first bit that will be matched against the mask. If offset falls beyond the end of EPC, the tag is considered non-matching.
4:3	Count	0 ~ 496	The number of bits in the mask. A length of zero will cause all tags to match. If (Offset+Count) falls beyond the end of the EPC, the tag is considered non-matching. Valid values are 0 to 496, inclusive.
7:5	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.2.2 Getting Post-Singulation Match Criteria

Description: Retrieves the configured post-singulation match criteria to be used by the RFID PCIe M.2 module. The post-singulation match criteria are used for any tag-protocol operations (i.e. RFID_18K6CTagInventory, etc.) in which the host specifies that a post-singulation match should be performed on the tags that are singulated by the tag-protocol operation. Post-singulation match criteria may not be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x27 / RFID_18K6CGetPostMatchCriteria**

Parameters:

Byte Offset	Name	Value	Description
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	Match	0, 1	Determines if the associated tag-protocol operation will be applied to tags that match the mask or not. 0 = not match 1 = match
3:2	Offset	0x0000 ~ 0x01FF	The offset in bits, from the start of the Electronic Product Code (EPC), of the first bit that will be matched against the mask.
5:4	Count	0 ~ 496	The number of bits in the mask. A length of zero will cause all tags to match.
7:6	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.2.3 Setting Post-Singulation Match Mask Data

Description: The post-singulation match mask is used to specify a bit pattern of up to 496 bits that is used to match against the EPC backscattered by a tag during singulation to determine if a tag is matching or non-matching.

Command: **ID: 0x28 / RFID_18K6CSetPostMatchMaskData**

Parameters:

Byte Offset	Name	Value	Description
0	MaskIndex	0 ~ 15	The mask data index to select the index number of mask data buffer.
1	MaskData0	0x00 ~ 0xFF	A buffer that contains a left-justified bit array that represents that bit pattern to match - i.e. the most significant bit of the bit array appears in the most-significant bit (i.e. bit 7) of the first byte of the buffer (i.e. mask[0]). All bits beyond count are ignored. For example, if the application wished to find tags with the following 16 bits 1011.1111.1010.0101, starting at offset 20 in the Electronic Product Code, then the fields would be set as follows: offset = 20 count = 16 mask[0] = 0xBF (1011.1111) mask[1] = 0xA5 (1010.0101) When the MaskIndex is equal to 0, this field saves mask[0] data. When the MaskIndex is equal to 1, this field saves mask[4] data So the content of MaskData0 field is 0xBF.
2	MaskData1	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field saves mask[1] data. For the above example, the content of MaskData1 field is 0xA5.

3	MaskData2	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field saves mask[2] data. **Note:** When the MaskIndex is equal to 15, this field is invalid.
4	MaskData3	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field saves mask[3] data. **Note:** When the MaskIndex is equal to 15, this field is invalid.
7:5	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.2.4 Getting Post-Singulation Match Mask Data

Description: Receives the contents of mask data buffer.

Command: **ID: 0x29 / RFID_18K6CGetPostMatchMaskData**

Parameters:

Byte Offset	Name	Value	Description
1	MaskIndex	0 ~ 15	Receive the mask data for the mask data buffer indexed.
7:2	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
1	MaskData0	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field reads mask[0] data. When the MaskIndex is equal to 1, this field reads mask[4] data.
2	MaskData1	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field reads mask[1] data.
3	MaskData2	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field reads mask[2] data. **Note:** When the MaskIndex is equal to 15, the value of this field is zero.
4	MaskData3	0x00 ~ 0xFF	When the MaskIndex is equal to 0, this field reads mask[3] data. **Note:** When the MaskIndex is equal to 15, the value of this field is zero.
7:5	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.3 Inventory-Round Configuration Parameters

Each of the tag-protocol operations consists of one or more inventory cycles, which may consist of one or more ISO 18000-6C inventory rounds for each enabled logical antenna port. The inventory-round configuration parameters consist of tag-population partitioning parameters (i.e. which tag populations, after being partitioned by one or more selection operations, will participate in the inventory round) and singulation algorithm parameters.

4.3.3.1 Specifying the Tags of Interest

Once the tag population has been partitioned into disjoint groups, a subsequent tag-protocol operation (i.e. an inventory operation or access command) is applied to one of the tag groups.

4.3.3.1.1 Setting the Tags of Interest

Description: Specifies which tag group will have subsequent tag-protocol operations (e.g. inventory, tag read, etc.) applied to it. The tag group may not be changed while a PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x30 / RFID_18K6CSetQueryTagGroup**

Parameters:

Byte Offset	Name	Value	Description
0	Selected	0, 2, 3	Specifies the state of the selected (SL) flag for tags that are to have the operation applied to them. 0 = All selected flag 2 = Deasserted selected flag 3 = Asserted selected flag
1	Session	0 ~ 3	Specifies which inventory session flag (i.e. S0, S1, S2, or S3) is matched against the inventory state specified by target. 0 = S0 session flag 1 = S1 session flag 2 = S2 session flag 3 = S3 session flag
2	Target	0, 1	Specifies the state of the inventory session flag (i.e. A or B), specified by session, for tags that are to have the operation applied to them. 0 = A target 1 = B target
7:3	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.3.1.2 Getting the Tags of Interest

Description: Retrieves the tag group that will have subsequent tag-protocol operations (e.g. inventory, tag read, etc.) applied to it. The tag group may not be retrieved while a PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x31 / RFID_18K6CGetQueryTagGroup**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.

1	Selected	0, 2, 3	Retrieves the state of the selected (SL) flag. 0 = All selected 2 = Deasserted selected 3 = Asserted selected
2	Session	0 ~ 3	Retrieves the inventory session flag (i.e. S0, S1, S2, or S3). 0 = S0 session 1 = S1 session 2 = S2 session 3 = S3 session
3	Target	0, 1	Retrieves the state of the inventory session flag (i.e. A or B). 0 = A target 1 = B target
7:4	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.3.2 Specifying Singulation Algorithm Parameters

Based upon usage scenarios, different singulation algorithms (i.e. Q-adjustment, etc.) may be desired. This section simply documents the mechanisms by which a host can choose and configure singulation algorithms. For the specific information about the singulation algorithms, consult Section 6. The supported singulation algorithms are as follows:

Table 4.3 - Supported Singulation Algorithms

Algorithm	Description
Fixed Q	Fixed Q value algorithm. This is the MAC firmware's singulation algorithm 0. **Note:** When performing non-inventory tag-access operations (i.e. read, write, kill, or lock, etc.), the MAC firmware always uses this singulation algorithm. If a host has not called RFID_18K6CSetSingulationAlgorithmParameters with this singulation algorithm prior to a non-inventory tag-access operation, the MAC firmware uses the most-recently-set fixed-Q singulation algorithm settings (or the power-up defaults if the fixed-Q singulation algorithm has never been configured).
Dynamic Q	Dynamic Q value algorithm. This algorithm uses a Q-modification algorithm that allows the host to control the change of the Q-adjustment-threshold value. This is the MAC's singulation algorithm 1.

4.3.3.2.1 Setting the Current Singulation Algorithm

Description: Allows the host to set the currently-active singulation algorithm (i.e. the one that is used when performing a tag-protocol operation (for example, inventory, tag read, etc.). The currently-active singulation algorithm may not be changed while a PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x32 / RFID_18K6CSetCurrentSingulationAlgorithm**

Parameters:

Byte Offset	Name	Value	Description
0	Algorithm	0, 1	The singulation algorithm that is to be used for subsequent tag-access operations. If this parameter does not represent a valid singulation algorithm, RFID_ERROR_INVALID_PARAMETER is returned. 0 = Fixed Q 1 = Dynamic Q
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.3.2.2 Getting the Current Singulation Algorithm

Description: Allows the host to retrieve the currently-active singulation algorithm. The currently-active singulation algorithm may not be retrieved while a PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x33 / RFID_18K6CGetCurrentSingulationAlgorithm**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	Algorithm	0, 1	Receive the singulation algorithm that is used for tag-access operations. 0 = Fixed Q 1 = Dynamic Q
7:2	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.3.2.3 Setting Singulation Algorithm Parameters

Description: Allows the host to configure the settings for a particular singulation algorithm. A singulation algorithm may not be configured while a PCIe M.2 module is executing a tag-protocol operation.

Note: Configuring a singulation algorithm does not automatically set it as the current singulation algorithm (see RFID_18K6CSetCurrentSingulationAlgorithm).

Command: **ID: 0x34 / RFID_18K6CSetSingulationAlgorithmParameters**

Parameters:

Byte Offset	Name	Value	Description
0	Algorithm	0, 1	The singulation algorithm to be configured. This parameter determines the type of Parameters file. If this parameter does not represent a valid singulation algorithm, RFID_ERROR_INVALID_PARAMETER is returned. 0 = Fixed Q 1 = Dynamic Q
6:1	Parameters	See Table 4.4/4.5	This field contains the singulation algorithm parameters. The type of Parameters file is determined by Algorithm. The length of Parameters field is six bytes for each Algorithm. See the following Tables 4.4 and 4.5 to know these detailed Parameters field descriptions of each singulation algorithm.
7	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters

			to stuff 8 bytes length.

Table 4.4 - Parameters Field Descriptions of Fixed Q (Algorithm = 0)

Byte Offset	Name	Value	Description
1	QValue	0 ~ 15	The Q value to use.
2	RetryCount	0 ~ 255	Specifies the number of times to try another execution of the singulation algorithm for the specified session/target before either toggling the target (if ToggleTarget is non-zero) or terminating the inventory/tag access operation.
3	ToggleTarget	0, 1	A flag that indicates if, after performing the inventory cycle for the specified target (i.e. A or B), the target should be toggled (i.e. A to B or B to A) and another inventory cycle should be run. **Note:** If the target is toggled, RetryCount and RepeatUntilNoTags also apply to the new target. 0 = the target should not be toggled. 1 = the target should be toggled.
4	RepeatUntilNo Tags	0, 1	A flag that indicates whether or not the singulation algorithm should continue performing inventory rounds until no tags are singulated. 0 = a single inventory round should be performed for each execution of the singulation algorithm. 1 = for each execution of the singulation algorithm, inventory rounds should be performed until no tags are singulated.
6:5	RFU	0	This field must be set to zero (reserved for future use).

Table 4.5 - Parameters Field Descriptions of Dynamic Q (Algorithm = 1)

Byte Offset	Name	Value	Description
1	StartQValue	0 ~ 15	The starting Q value to use. StartQValue must be greater than or equal to MinQValue and less than or equal to MaxQValue.
2	MinQValue	0 ~ 15	The minimum Q value to use. MinQValue must be less than or equal to StartQValue and MaxQValue.
3	MaxQValue	0 ~ 15	The maximum Q value to use. MaxQValue must be greater than or equal to StartQValue and MinQValue.
4	RetryCount	0 ~ 255	Specifies the number of times to try another execution of the singulation algorithm for the specified session/target before either toggling the target (if ToggleTarget is non-zero) or terminating the inventory/tag access operation.
5	ToggleTarget	0, 1	A flag that indicates if, after performing the inventory cycle for the specified target (i.e. A or B), the target should be toggled (i.e. A to B or B to A) and another inventory cycle should be run. **Note:** If the target is toggled, RetryCount will also apply to the new target. 0 = the target should not be toggled. 1 = that the target should be toggled.
6	ThresholdMulti plier	0 ~ 255	The multiplier, specified in units of fourths (i.e. 0.25), that will be applied to the Q-adjustment threshold as part of the dynamic-Q algorithm. For example, a value of 7 represents a multiplier of 1.75.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.3.2.4 Getting Singulation Algorithm Parameters

Description: Allows the host to retrieve the settings for a particular singulation algorithm. Singulation-algorithm parameters may not be retrieved while a PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x35 / RFID_18K6CGetSingulationAlgorithmParameters**

Parameters:

Byte Offset	Name	Value	Description
0	Algorithm	0, 1	The singulation algorithm for which parameters are to be retrieved. This parameter determines the type of return data. If this parameter does not represent a valid singulation algorithm, RFID_ERROR_INVALID_PARAMETER is returned. 0 = Fixed Q 1 = Dynamic Q
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
6:1	Parameters	See Table 4.4/4.5	This field contains the singulation algorithm parameters. The type of this file is determined by Algorithm. The length of this field is six bytes for each Algorithm. See Tables 4.4 and 4.5 to know these detailed Parameters field descriptions of each singulation algorithm.
7	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.4 Accessing the Tag Access Password

4.3.4.1 Setting the Tag Access Password

Description: Saves content of tag access password for ISO 18000-6C tag-access operations (i.e. read, write, etc.) used.

Command: **ID: 0x36 / RFID_18K6CSetTagAccessPassword**

Parameters:

Byte Offset	Name	Value	Description
3:0	Access Password	0x00000000 ~ 0xFFFFFFFF	Save the access password for the tags. A value of zero indicates no access password.
7:4	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.4.2 Getting the Tag Access Password

Description: Retrieves access password of the tags for ISO 18000-6C tag-access operations (i.e. read, write, etc.) used.

Command: **ID: 0x37 / RFID_18K6CGetTagAccessPassword**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
4:1	Access Password	0x00000000 ~ 0xFFFFFFFF	Receive the value of tag access password. A value of zero indicates no access password.
7:5	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.5 Accessing Tag Writing Data Buffer

The tag writing data buffer is used to save the contents of tag memory for both tag multiple-write and tag block-write operations.

4.3.5.1 Setting Tag Writing Data Buffer

Description: Saves the contents in the tag writing data buffer.

Command: **ID: 0x38 / RFID_18K6CSetTagWriteDataBuffer**

Parameters:

Byte Offset	Name	Value	Description
0	BufferIndex	0 ~ 255	The buffer index to select the index number of tag writing data buffer. The available range is 0 to 31 for tag multiple-write operation. The available range is 0 to 255 for tag block-write operation.
2:1	BufferData	0x0000 ~ 0xFFFF	The 16-bit value to write to the tag writing data buffer specified by index number.
3	OffsetType	0	This field must be set to zero (reserved for future use).
5:4	DataOffset	0	This field must be set to zero (reserved for future use).
7:6	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.3.5.2 Getting Tag Writing Data Buffer

Description: Receives the contents of tag writing data buffer.

Command: **ID: 0x39 / RFID_18K6CGetTagWriteDataBuffer**

Parameters:

Byte Offset	Name	Value	Description
0	BufferIndex	0 ~ 255	Receive the buffer data for the tag writing data buffer indexed.
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
2:1	BufferData	0x0000 ~ 0xFFFF	Receive a 16-bit value in the tag writing data buffer specified by index number.
4:3	DataOffset	0	Read as zero (reserved for future use).
7:5	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.4 ISO 18000-6C Tag-Protocol Operation

As tags are singulated, the ISO 18000-6C access command is (optionally) applied - this implies that only tags that match the optionally-supplied selection criteria have the access command applied to them. The RFID PCIe M.2 module applies the access command immediately after the tag is singulated (in contrast to gathering an inventory of all tags and then applying the access command to all inventoried tags). The MTI RFID PCIe M.2 module allows the host to explicitly issue the following ISO 18000-6C access commands: read, write, kill, lock, blockWrite, and blockErase. The access commands all perform an implicit inventory operation.

4.4.1 Tag Inventory Operation

An inventory operation allows the host to gather the EPCs for all tags of interest.

Description: Executes a tag inventory for all tags of interest. If the PerformSelect parameter is specified, the tag population is partitioned (i.e. one or more ISO 18000-6C select commands are issued) before the inventory operation. If the PerformPostMatch parameter is specified, a singulated tag's EPC is matched against post-singulation match mask to determine if the tag is to be returned to the host. The function returns the operation-response packets. The function is returned when the inventory operation is completed or has been cancelled. A host may prematurely stop an inventory operation by calling RFID_ControlCancel on another thread. A tag inventory may not be requested while a PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x40 / RFID_18K6CTagInventory**

Parameters:

Byte Offset	Name	Value	Description
0	PerformSelect	0, 1	Indicates that one or more ISO 18000-6C select commands are to be executed prior to the tags being singulated. If RFID_18K6CSetSelectCriteria has not been called previously to configure tag-selection criteria, the RFID PCIe M.2 module will use the default tag-selection criteria. If this PerformSelect is not set, no select commands are executed. 0 = no select commands. 1 = perform select commands.
1	PerformPost Match	0, 1	Indicates that a post-singulation match is to be performed on a singulated tag's EPC to determine if the tag is to be returned to the application. If RFID_18K6CSetPostMatchCriteria has not been called previously to configure post-singulation match criteria, the RFID PCIe M.2 module will use the default post-singulation match criteria. If this PerformPostMatch is not set, then the post-singulation match is not performed, and all singulated tags will be returned to the host. 0 = no post-singulation match command. 1 = perform post-singulation match command.
7:2	Padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

Report Packet: Command-Begin / Inventory-Response / Command-End three packet types.
See Section 3.2.3 to know detailed information about these report packets.

4.4.2 Tag Read Operation

Reading tag data should not be confused with performing an inventory. Whereas an inventory is restricted to returning all of the tag's EPC data, a read operation can be used to read one or more 16-bit words from any of a tag's memory banks. While a read may be used to retrieve a set of tag EPC data, if the EPC is the only desired data, performing an inventory operation is more efficient at the low-level tag access.

Description: Executes a tag read of the specified tag data for all tags of interest. If the PerformSelect parameter is specified, the tag population is partitioned (i.e. one or more ISO 18000-6C select commands are issued) before the tag-read operation. If the PerformPostMatch parameter is specified, a singulated tag's EPC is matched against post-singulation match mask to determine if the tag is to be read from. Reads may only be performed on 16-bit word boundaries and for multiples of 16-bit words. If one or more of the memory words specified by the Offset/Count combination do not exist or are read-locked, the read from the tag fails, and this failure is reported through the operation response packet. The function returns the operation-response packets. The function is returned when the tag-read operation is completed or has been cancelled. A host may prematurely stop a tag-read operation by calling RFID_ControlCancel on another thread. A tag read may not be requested while a PCIe M.2 module is executing a tag-protocol operation.

Note: When performing an ISO 18000-6C tag-read operation, the RFID PCIe M.2 module uses only the first enabled logical antenna (i.e. the enabled logical antenna with the smallest logical antenna port number).

Note: When performing an ISO 18000-6C tag-read operation, the RFID PCIe M.2 module always uses the fixed-Q singulation algorithm with the most-recent settings for that algorithm. If a host has never called RFID_18K6CSetSingulationAlgorithmParameters and specified the fixed-Q singulation algorithm, the RFID PCIe M.2 module uses the power-up defaults for the fixed-Q singulation algorithm.

Command: **ID: 0x41 / RFID_18K6CTagRead**

Parameters:

Byte Offset	Name	Value	Description
0	Bank	0 ~ 3	The memory bank from which to read. Valid values are: 0 = Reserved memory 1 = EPC memory 2 = TID memory 3 = User memory
2:1	Offset	0x0000 ~ 0xFFFF	The offset of the first 16-bit word, where zero is the first 16-bit word in the memory bank, to read from the specified memory bank.
3	Count	1 ~ 253	The number of 16-bit words to be read.
4	RetryCount	0 ~ 7	The number of times to retry this tag-access operation if unsuccessful.
5	PerformSelect	0, 1	Indicates that one or more ISO 18000-6C select commands are to be executed prior to the tags being singulated. If RFID_18K6CSetSelectCriteria has not been called previously to configure tag-selection criteria, the RFID PCIe M.2 module will use the default tag-selection criteria. If this PerformSelect is not set, no select commands are executed. 0 = no select commands. 1 = perform select commands.
6	PerformPost Match	0, 1	Indicates that a post-singulation match is to be performed on a singulated tag's EPC to determine if the tag is to be returned to the application. If RFID_18K6CSetPostMatchCriteria has not been called previously to configure post-singulation match criteria, the RFID PCIe M.2 module will use the default post-singulation match criteria. If this PerformPostMatch is not set, then the post-singulation match is not performed, and all singulated tags will be read from.

			0 = no post-singulation match command. 1 = perform post-singulation match command.
7	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

Report Packet: Command-Begin / Inventory-Response / Tag-Access / Command-End four packet types. See Section 3.2.3 to know detailed information about these report packets.

4.4.3 Tag Write Operation

A tag-write command allows a host to write one 16-bit word to the specified memory bank of the ISO 18000-6C tags of interest.

Description: Executes a tag write of one 16-bit word for all tags of interest. If the PerformSelect parameter is specified, the tag population is partitioned (i.e. one or more ISO 18000-6C select commands are issued) before the tag-write operation. If the PerformPostMatch parameter is specified, a singulated tag's EPC is matched against post-singulation match mask to determine if the tag is to be written to. Writes may only be performed on 16-bit word boundaries. If one memory word specified by the Offset does not exist or is write-locked, the write to the tag fails, and this failure is reported through the operation-response packet. The function returns the operation-response packets. The function is returned when the tag-write operation is completed or has been cancelled. A host may prematurely stop a tag-write operation by calling RFID_ControlCancel on another thread. A tag write may not be requested while a PCIe M.2 module is executing a tag-protocol operation.

Note: When performing an ISO 18000-6C tag-write operation, the RFID PCIe M.2 module only uses the first enabled logical antenna (i.e. the enabled logical antenna with the smallest logical antenna port number).

Note: When performing an ISO 18000-6C tag-write operation, the RFID PCIe M.2 module always uses the fixed-Q singulation algorithm with the most-recent settings for that algorithm. If a host has never called RFID_18K6CSetSingulationAlgorithmParameters and specified the fixed-Q singulation algorithm, the RFID PCIe M.2 module uses the power-up defaults for the fixed-Q singulation algorithm.

Command: **ID: 0x42 / RFID_18K6CTagWrite**

Parameters:

Byte Offset	Name	Value	Description
0	Bank	0 ~ 3	The memory bank in which to write. Valid values are: 0 = Reserved memory 1 = EPC memory 2 = TID memory 3 = User memory
2:1	Offset	0x0000 ~ 0xFFFF	The offset of the first 16-bit word, where zero is the first 16-bit word in the memory bank, to write in the specified memory bank.
4:3	Data	0x0000 ~ 0xFFFF	The 16-bit word value to be written to the tag's specified memory bank.
5	RetryCount	0 ~ 7	The number of times to retry this tag-access operation if unsuccessful.

6	PerformSelect	0, 1	Indicates that one or more ISO 18000-6C select commands are to be executed prior to the tags being singulated. If RFID_18K6CSetSelectCriteria has not been called previously to configure tag-selection criteria, the RFID PCIe M.2 module will use the default tag-selection criteria. If this PerformSelect is not set, no select commands are executed. 0 = no select commands. 1 = perform select commands.
7	PerformPost Match	0, 1	Indicates that a post-singulation match is to be performed on a singulated tag's EPC to determine if the tag is to be returned to the application. If RFID_18K6CSetPostMatchCriteria has not been called previously to configure post-singulation match criteria, the RFID PCIe M.2 module will use the default post-singulation match criteria. If this PerformPostMatch is not set, then the post-singulation match is not performed, and all singulated tags will be written to. 0 = no post-singulation match command. 1 = perform post-singulation match command.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

Report Packet: Command-Begin / Inventory-Response / Tag-Access / Command-End four packet types. See Section 3.2.3 to know detailed information about these report packets.

4.4.4 Tag Kill Operation

A tag-kill command allows a host to kill (i.e. render inoperable) a set of tags of interest.

Description: Executes a tag kill for all tags of interest. If the PerformSelect parameter is specified, the tag population is partitioned (i.e. one or more ISO 18000-6C select commands are issued) before the tag-kill operation. If the PerformPostMatch parameter is specified, a singulated tag's EPC is matched against post-singulation match mask to determine if the tag is to be killed. The function returns the operation-response packets. The function is returned when the tag-kill operation is completed or has been cancelled. A host may prematurely stop a tag-kill operation by calling RFID_ControlCancel on another thread. A tag kill may not be requested while a PCIe M.2 module is executing a tag-protocol operation.

Note: When performing an ISO 18000-6C tag-kill operation, the RFID PCIe M.2 module uses only the first enabled logical antenna (i.e. the enabled logical antenna with the smallest logical antenna port number).

Note: When performing an ISO 18000-6C tag-kill operation, the RFID PCIe M.2 module always uses the fixed-Q singulation algorithm with the most-recent settings for that algorithm. If a host has never called RFID_18K6CSetSingulationAlgorithmParameters and specified the fixed-Q singulation algorithm, the RFID PCIe M.2 module uses the power-up defaults for the fixed-Q singulation algorithm.

Command: **ID: 0x43 / RFID_18K6CTagKill**

Parameters:

Byte Offset	Name	Value	Description
3:0	KillPassword	0x00000000 ~	The kill password for the tags.

		0xFFFFFFFF	
4	RetryCount	0 ~ 7	The number of times to retry this tag-access operation if unsuccessful.
5	PerformSelect	0, 1	Indicates that one or more ISO 18000-6C select commands are to be executed prior to the tags being singulated. If RFID_18K6CSetSelectCriteria has not been called previously to configure tag-selection criteria, the RFID PCIe M.2 module will use the default tag-selection criteria. If this PerformSelect is not set, no select commands are executed. 0 = no select commands. 1 = perform select commands.
6	PerformPost Match	0, 1	Indicates that a post-singulation match is to be performed on a singulated tag's EPC to determine if the tag is to be returned to the application. If RFID_18K6CSetPostMatchCriteria has not been called previously to configure post-singulation match criteria, the RFID PCIe M.2 module will use the default post-singulation match criteria. If this PerformPostMatch is not set, then the post-singulation match is not performed, and all singulated tags will be killed. 0 = no post-singulation match command. 1 = perform post-singulation match command.
7	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

Report Packet: Command-Begin / Inventory-Response / Tag-Access / Command-End four packet types. See Section 3.2.3 to know detailed information about these report packets.

4.4.5 Tag Lock Operation

A tag-permission command (tag lock) allows the host to set the access permissions of a tag. These include the following:

- Set whether or not an access password is required to write to the EPC, TID, or user memory banks.
- Set whether or not the above memory-write permission is permanently set. Once the memory-write permission has been permanently set, attempts to change the permission or turn off the permanent setting fail.
- Set a memory bank to be read-only.
- Set whether or not the individual passwords (i.e. access and kill) may be accessed (i.e. read and written) and, if they are accessible, whether or not an access password is required to read the individual passwords (i.e. access and kill).
- Set whether or not the above password-access permission is permanently set. Once the password-access permission has been permanently set, attempts to change the permission or turn off the permanent setting fail.
- Set the individual passwords to be inaccessible (i.e. unable to be read or written).

For a tag, there are five access permissions that may be set: access permissions for the EPC, TID, and user memory banks and access permissions for the access and kill passwords.

There are several scenarios in which attempting to set a tag's access permissions may fail:

- Attempting to change the access permission for a non-existent memory bank or password.

- Attempting to change an access permission that has been previously set as permanent.
- Attempting to change the permanent status of an access permission that has been previously set as permanent.
- Attempting to lock a password or memory bank that is not lockable.
- Attempting to unlock a password or memory bank that is not unlockable.

Description: Executes a tag lock (setting a tag's access permissions) for all tags of interest. If the PerformSelect parameter is specified, the tag population is partitioned (i.e. one or more ISO 18000-6C select commands are issued) before the tag-lock operation. If the PerformPostMatch parameter is specified, a singulated tag's EPC is matched against post-singulation match mask to determine if the tag is to be locked. The function returns the operation-response packets. The function is returned when the tag-lock operation is completed or has been cancelled. A host can prematurely stop a tag-lock operation by calling RFID_ControlCancel on another thread. A tag lock may not be requested while a PCIe M.2 module is executing a tag-protocol operation.

Note: When performing an ISO 18000-6C tag-lock operation, the RFID PCIe M.2 module uses only the first enabled logical antenna (i.e. the enabled logical antenna with the smallest logical antenna port number).

Note: When performing an ISO 18000-6C tag-lock operation, the RFID PCIe M.2 module always uses the fixed-Q singulation algorithm with the most-recent settings for that algorithm. If a host has never called RFID_18K6CSetSingulationAlgorithmParameters and specified the fixed-Q singulation algorithm, the RFID PCIe M.2 module uses the power-up defaults for the fixed-Q singulation algorithm.

Command: **ID: 0x44 / RFID_18K6CTagLock**

Parameters:

Byte Offset	Name	Value	Description
0	KillPassword Permissions	0 ~ 4	The access permissions for the tag's kill password. 0 = Accessible: The password can be read and written when the tag is in either the open or secured states. 1 = Always Accessible: The password can be read and written when the tag is in either the open or secured states, and this access permission should be set permanently. 2 = Password Accessible: The password can be read or written only when the tag is in the secured state. 3 = Always Not Accessible: The password cannot be read or written, and this access permission should be set permanently. 4 = No Change: The password's access permission should remain unchanged.
1	AccessPasswordPermissions	0 ~ 4	The access permissions for the tag's access password. Five setting values as per the description of KillPasswordPermissions field.
2	EPCMemoryBankPermissions	0 ~ 4	The access permissions for the tag's EPC memory bank. 0 = Writeable: The memory bank is writeable when the tag is in either the open or secured states. 1 = Always Writeable: The memory bank is writeable when the tag is in either the open or secured states, and this access permission should be set permanently. 2 = Password Writeable:

			The memory bank is writeable only when the tag is in the secured state. 3 = Always Not Writeable: The memory bank is not writeable, and this access permission should be set permanently. 4 = No Change: The memory bank's access permission should remain unchanged.
3	TIDMemoryBankPermissions	0 ~ 4	The access permissions for the tag's TID memory bank. Five setting values as per the description of EPCMemoryBankPermissions field.
4	UserMemoryBankPermissions	0 ~ 4	The access permissions for the tag's user memory bank. Five setting values as per the description of EPCMemoryBankPermissions field.
5	RetryCount	0 ~ 7	The number of times to retry this tag-access operation if unsuccessful.
6	PerformSelect	0, 1	Indicates that one or more ISO 18000-6C select commands are to be executed prior to the tags being singulated. If RFID_18K6CSetSelectCriteria has not been called previously to configure tag-selection criteria, the RFID PCIe M.2 module will use the default tag-selection criteria. If this PerformSelect is not set, no select commands are executed. 0 = no select commands. 1 = perform select commands.
7	PerformPost Match	0, 1	Indicates that a post-singulation match is to be performed on a singulated tag's EPC to determine if the tag is to be returned to the application. If RFID_18K6CSetPostMatchCriteria has not been called previously to configure post-singulation match criteria, the RFID PCIe M.2 module will use the default post-singulation match criteria. If this PerformPostMatch is not set, then the post-singulation match is not performed, and all singulated tags will be locked. 0 = no post-singulation match command. 1 = perform post-singulation match command.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

Report Packet: Command-Begin / Inventory-Response / Tag-Access / Command-End four packet types.
See Section 3.2.3 to know detailed information about these report packets.

4.4.6 Tag Multiple Write Operation

A tag-multiple-write command allows a host to write one or more 16-bit words to the specified memory bank of the ISO 18000-6C tags of interest.

Description: Executes a tag multiple write of one or more 16-bit words for all tags of interest. If the PerformSelect parameter is specified, the tag population is partitioned (i.e. one or more ISO 18000-6C select commands are issued) before the tag-multiple-write operation. If the PerformPostMatch parameter is specified, a singulated tag's EPC is matched against post-singulation match mask to determine if the tag is to be written to. Writes may only be performed on 16-bit word boundaries. If one or more memory words specified by the Offset do not exist or are write-locked, the write to the tag fails and this failure is reported through the operation-response packet. The function returns the operation-response packets. The function is returned when the tag-multiple-write operation is completed or has been cancelled. A host may prematurely stop a tag-multiple-write operation by calling RFID_ControlCancel on another thread. A tag multiple write may not be requested while a PCIe M.2 module is executing a tag-protocol operation.

Note: When performing an ISO 18000-6C tag-multiple-write operation, the RFID PCIe M.2 module only uses the first enabled logical antenna (i.e. the enabled logical antenna with the smallest logical antenna port number).

Note: When performing an ISO 18000-6C tag-multiple-write operation, the RFID PCIe M.2 module always uses the fixed-Q singulation algorithm with the most-recent settings for that algorithm. If a host has never called RFID_18K6CSetSingulationAlgorithmParameters and specified the fixed-Q singulation algorithm, the RFID PCIe M.2 module uses the power-up defaults for the fixed-Q singulation algorithm.

Command: **ID: 0x45 / RFID_18K6CTagMultipleWrite**

Parameters:

Byte Offset	Name	Value	Description
0	Bank	0 ~ 3	The memory bank in which to write. Valid values are: 0 = Reserved memory 1 = EPC memory 2 = TID memory 3 = User memory
2:1	Offset	0x0000 ~ 0xFFFF	The offset of the first 16-bit word, where zero is the first 16-bit word in the memory bank, to write in the specified memory bank.
3	DataLength	1 ~ 32	The number of 16-bit words to be written. These contents of 16-bit words are from tag writing data buffer.
4	Reserved	0	This field must be set to zero (reserved for future use).
5	RetryCount	0 ~ 7	The number of times to retry this tag-access operation if unsuccessful.
6	PerformSelect	0, 1	Indicates that one or more ISO 18000-6C select commands are to be executed prior to the tags being singulated. If RFID_18K6CSetSelectCriteria has not been called previously to configure tag-selection criteria, the RFID PCIe M.2 module will use the default tag-selection criteria. If this PerformSelect is not set, no select commands are executed. 0 = no select commands. 1 = perform select commands.
7	PerformPost Match	0, 1	Indicates that a post-singulation match is to be performed on a singulated tag's EPC to determine if the tag is to be returned to the application. If RFID_18K6CSetPostMatchCriteria has not been called previously to configure post-singulation match criteria, the RFID PCIe M.2 module will use the default post-singulation match criteria. If this PerformPostMatch is not set, then the post-singulation match is not performed, and all singulated tags will be written to. 0 = no post-singulation match command. 1 = perform post-singulation match command.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

Report Packet: Command-Begin / Inventory-Response / Tag-Access / Command-End four packet types.

See Section 3.2.3 to know detailed information about these report packets.

4.4.7 Tag Block Write Operation

A tag-blockwrite command allows a host to write multiple consecutive 16-bit words to the specified memory bank of the ISO 18000-6C tags of interest, using a single ISO 18000-6C protocol command. A blockwrite can be performed on a contiguous set of one or more 16-bit words to one of the tag's memory banks.

Description: Executes a tag blockwrite of one or more 16-bit words for all tags of interest. If the PerformSelect parameter is specified, the tag population is partitioned (i.e. one or more ISO 18000-6C select commands are issued) before the tag-blockwrite operation. If the PerformPostMatch parameter is specified, a singulated tag's EPC is matched against the post-singulation match mask to determine if the tag is to be written to. Blockwrites may only be performed on 16-bit word boundaries and for multiples of 16-bit words. If one or more memory words specified by the Offset do not exist or are write-locked, the write to the tag fails and this failure is reported through the operation-response packet. The function returns the operation-response packets. The function is returned when the tag-blockwrite operation is completed or has been cancelled. A host may prematurely stop a tag-blockwrite operation by calling RFID_ControlCancel on another thread. A tag multiple write may not be requested while a PCIe M.2 module is executing a tag-protocol operation.

Note: When performing an ISO 18000-6C tag-blockwrite operation, the RFID PCIe M.2 module only uses the first enabled logical antenna (i.e. the enabled logical antenna with the smallest logical antenna port number).

Note: When performing an ISO 18000-6C tag-blockwrite operation, the RFID PCIe M.2 module always uses the fixed-Q singulation algorithm with the most-recent settings for that algorithm. If a host has never called RFID_18K6CSetSingulationAlgorithmParameters and specified the fixed-Q singulation algorithm, the RFID PCIe M.2 module uses the power-up defaults for the fixed-Q singulation algorithm.

Command: **ID: 0x46 / RFID_18K6CTagBlockWrite**

Parameters:

Byte Offset	Name	Value	Description
0	Bank	0 ~ 3	The memory bank in which to write. Valid values are: 0 = Reserved memory 1 = EPC memory 2 = TID memory 3 = User memory
2:1	Offset	0x0000 ~ 0xFFFF	The offset of the first 16-bit word, where zero is the first 16-bit word in the memory bank, to write in the specified memory bank.
3	DataLength	1 ~ 255	The number of 16-bit words to be written. These contents of 16-bit words are from tag writing data buffer.
4	Reserved	0	This field must be set to zero (reserved for future use).
5	RetryCount	0 ~ 7	The number of times to retry this tag-access operation if unsuccessful.
6	PerformSelect	0, 1	Indicates that one or more ISO 18000-6C select commands are to be executed prior to the tags being singulated. If RFID_18K6CSetSelectCriteria has not been called previously to configure tag-selection criteria, the RFID PCIe M.2 module will use the default tag-selection criteria. If this PerformSelect is not set, no select commands are executed. 0 = no select commands. 1 = perform select commands.
7	PerformPost Match	0, 1	Indicates that a post-singulation match is to be performed on a singulated tag's EPC to determine if the tag is to be returned to the application. If RFID_18K6CSetPostMatchCriteria has not been called

			previously to configure post-singulation match criteria, the RFID PCIe M.2 module will use the default post-singulation match criteria. If this PerformPostMatch is not set, then the post-singulation match is not performed, and all singulated tags will be written to. 0 = no post-singulation match command. 1 = perform post-singulation match command.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

Report Packet: Command-Begin / Inventory-Response / Tag-Access / Command-End four packet types. See Section 3.2.3 to know detailed information about these report packets.

4.4.8 Tag Block Erase Operation

A tag-blockerase command allows a host to erase multiple consecutive 16-bit words to the specified memory bank of the ISO 18000-6C tags of interest, using a single ISO 18000-6C protocol command.

Description: Executes a tag blockerase of one or more 16-bit words for all tags of interest. If the PerformSelect parameter is specified, the tag population is partitioned (i.e. one or more ISO 18000-6C select commands are issued) before the tag-blockerase operation. If the PerformPostMatch parameter is specified, a singulated tag's EPC is matched against post-singulation match mask to determine if the tag is to be erased. Blockerases may only be performed on 16-bit word boundaries and for multiples of 16-bit words. If one or more memory words specified by the Offset do not exist or are write-locked, the erase to the tag fails and this failure is reported through the operation-response packet. The function returns the operation-response packets. The function is returned when the tag-blockerase operation is completed or has been cancelled. A host may prematurely stop a tag-blockerase operation by calling RFID_ControlCancel on another thread. A tag blockerase may not be requested while a PCIe M.2 module is executing a tag-protocol operation.

Note: When performing an ISO 18000-6C tag-blockerase operation, the RFID PCIe M.2 module only uses the first enabled logical antenna (i.e. the enabled logical antenna with the smallest logical antenna port number).

Note: When performing an ISO 18000-6C tag-blockerase operation, the RFID PCIe M.2 module always uses the fixed-Q singulation algorithm with the most-recent settings for that algorithm. If a host has never called RFID_18K6CSetSingulationAlgorithmParameters and specified the fixed-Q singulation algorithm, the RFID PCIe M.2 module uses the power-up defaults for the fixed-Q singulation algorithm.

Command: **ID: 0x47 / RFID_18K6CTagBlockErase**
Parameters:

Byte Offset	Name	Value	Description
0	Bank	0 ~ 3	The memory bank in which to erase. Valid values are: 0 = Reserved memory 1 = EPC memory 2 = TID memory 3 = User memory
2:1	Offset	0x0000 ~ 0xFFFF	The offset of the first 16-bit word, where zero is the first 16-bit word in the memory bank, to write in the specified memory bank.
3	DataLength	1 ~ 255	The number of 16-bit words to be erased. These contents of 16-bit words are from tag writing data buffer.

4	Reserved	0	This field must be set to zero (reserved for future use).
5	RetryCount	0 ~ 7	The number of times to retry this tag-access operation if unsuccessful.
6	PerformSelect	0, 1	Indicates that one or more ISO 18000-6C select commands are to be executed prior to the tags being singulated. If RFID_18K6CSetSelectCriteria has not been called previously to configure tag-selection criteria, the RFID PCIe M.2 module will use the default tag-selection criteria. If this PerformSelect is not set, no select commands are executed. 0 = no select commands. 1 = perform select commands.
7	PerformPost Match	0, 1	Indicates that a post-singulation match is to be performed on a singulated tag's EPC to determine if the tag is to be returned to the application. If RFID_18K6CSetPostMatchCriteria has not been called previously to configure post-singulation match criteria, the RFID PCIe M.2 module will use the default post-singulation match criteria. If this PerformPostMatch is not set, then the post-singulation match is not performed, and all singulated tags will be erased. 0 = no post-singulation match command. 1 = perform post-singulation match command.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

Report Packet: Command-Begin / Inventory-Response / Tag-Access / Command-End four packet types.
See Section 3.2.3 to know detailed information about these report packets.

4.5 RFID PCIe M.2 Module Control Operation

4.5.1 Canceling a Tag-Protocol Operation

If a host wishes to stop the execution of a tag-protocol operation but still wishes to receive the buffered operation-response packets (up to and including the packet that indicates the end of the operation), it can issue a cancel request on another thread. When processing a cancel request, the MAC firmware will send a command-end packet lastly.

Description: Stops a currently-executing tag-protocol operation (i.e. RFID_18K6CTagInventory, etc.) on the PCIe M.2 module. The MTI RFID PCIe M.2 module delivers the buffered operation-response packets to the host, any operation-response packets that are buffered on the RFID PCIe M.2 module's firmware or were in transit before the firmware had the opportunity to cancel the tag-protocol operation. RFID_ControlCancel does not return until after the cancel request has completed processing (i.e. the corresponding command-end packet has been seen). This normally takes a few hundred milliseconds, but, depending upon the state of the RFID PCIe M.2 module, may take several seconds.

Command: **ID: 0x50 / RFID_ControlCancel**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

No response (packet).

4.5.2 Pausing a Tag-Protocol Operation

Description: Pauses a currently-executing tag-protocol operation (i.e. RFID_18K6CTagInventory, etc.) on the PCIe M.2 module. The MTI RFID PCIe M.2 module delivers the operation-response packet that was in transit to the host and then pause the tag-protocol operation.

Command: **ID: 0x52 / RFID_ControlPause**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

No response (packet).

4.5.3 Resuming a Tag-Protocol Operation

Description: Resumes a paused tag-protocol operation (i.e. RFID_18K6CTagInventory, etc.) on the PCIe M.2 module. The MTI RFID PCIe M.2 module continuously delivers the operation-response packets to the host.

Command: **ID: 0x53 / RFID_ControlResume**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

No response (packet).

4.5.4 Performing a Software Reset

Description: Causes the MAC firmware to perform the specified reset. Any currently-executing operations are aborted, and unconsumed data is discarded. The MAC firmware runs a built-in self-test and reinitializes all board hardware. The RFID PCIe M.2 module is placed in an idle state.

Command: **ID: 0x54 / RFID_ControlSoftReset**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0xFF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.5.5 Perform a Firmware Reset to Bootloader

Description: When the host issues this command, the MAC firmware will immediately reset the MTI RFID PCIe M.2 module, which in turn causes the bootloader to start execution in the same way it would upon a board hardware reset, except that the bootloader will bypass the handoff to the MAC firmware and remain active. All data currently buffered by the MAC firmware is lost when the MTI RFID PCIe M.2 module resets, as with software reset processing.

Command: **ID: 0x55 / RFID_ControlResetToBootloader**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0xFF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.5.6 RFID PCIe M.2 Module Power Management

Description: The MTI RFID PCIe M.2 module enters low power standby mode to reduce power consumption. The following commands should cause the MTI RFID PCIe M.2 module to return ready power mode:

- RFID_RadioSetCurrentLinkProfile
- RFID_18K6CTagInventory
- RFID_18K6CTagRead

- RFID_18K6CTagWrite
- RFID_18K6CTagKill
- RFID_18K6CTagLock
- RFID_18K6CTagMultipleWrite
- RFID_18K6CTagBlockWrite
- RFID_18K6CTagBlockErase
- RFID_MacBypassWriteRegister
- RFID_MacBypassReadRegister
- RFID_TestSetFrequencyConfiguration
- RFID_TestTurnOnCarrierWave
- RFID_TestTurnOffCarrierWave
- RFID_TestInjectRandomData
- RFID_TestTransmitRandomData
- RFID_EngGetTemperature
- RFID_EngGetRFPower

Other commands are still performed and don't change power mode of the MTI RFID PCIe M.2 module.

4.5.6.1 Setting the Power Management State

Command: **ID: 0x56 / RFID_ControlSetPowerState**
Parameters:

Byte Offset	Name	Value	Description
0	PowerState	0, 1	Setting the power state of the RFID PCIe M.2 module. 0 = ready 1 = standby
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.5.6.2 Retrieving the Power Management State

Command: **ID: 0x57 / RFID_ControlGetPowerState**
Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	PowerState	0, 1	Receive the current power state of the RFID PCIe M.2 module. 0 = ready 1 = standby
7:2	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6 RFID PCIe M.2 Module Firmware Access

4.6.1 Retrieving the MAC Firmware Version Information

Description: Retrieves the version number for the RFID PCIe M.2 module's MAC firmware. The RFID PCIe M.2 module's MAC firmware version may not be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x60 / RFID_MacGetFirmwareVersion**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	MajorVersion Number	0 ~ 99	The major version number.
2	MinorVersion Number	0 ~ 99	The minor version number.
3	PatchVersion Number	0 ~ 99	The patch level.
7:4	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.2 Retrieving the MAC Firmware Debug Value

Description: This command will hold fixed debug values for testing purposes only.

Command: **ID: 0x61 / RFID_MacGetDebug**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
4:1	DebugValue	0xA5A5 5A5A	The major version number.
7:5	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.3 Clearing a MAC Firmware Error

Any of the tag-protocol operations (for example, RFID_18K6CTagInventory) may cause the RFID PCIe M.2 module's

MAC firmware to enter an error state. A host can detect when the RFID PCIe M.2 module's MAC firmware has entered an error state by inspecting the status field of the command-end packet (see Section 3.2.3.3.2) for the tag-protocol operation - a non-zero value indicates an error. A host should perform RFID_MacClearError when any error code occurs in the command-end packet.

Note that the MAC's error state will also be automatically cleared on the next access to a MAC register and at the start of the next command from host interface when a failure causes by performing non-tag-protocol operations (e.g. RFID_MacGetDebug, etc.).

Description: Attempts to clear the error state for the RFID PCIe M.2 module's MAC firmware. The MAC's error state may not be cleared while a radio module is executing a tag-protocol operation.

Command: **ID: 0x62 / RFID_MacClearError**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.4 Retrieving a MAC Firmware Error Code

Any of the tag-protocol operations (e.g. RFID_18K6CTagInventory, etc.) may cause the RFID PCIe M.2 module's MAC firmware to enter an error state. A host can detect when the RFID PCIe M.2 module's MAC firmware has entered an error state by inspecting the status field of the command-end packet (see Section 3.2.3.3.2) for the tag-protocol operation - a non-zero value indicates an error - or more directly by calling the RFID_MacGetError. The current error state is returned as well as a cached value of the last error state.

Any of the non-tag-protocol operations (e.g. RFID_MacGetDebug, etc.) may also cause a failure. A host can perform RFID_MacGetError when RFID_ERROR_MODULE_FAILURE occurs in the status field of command-response packet.

Description: Retrieves the error state of the RFID PCIe M.2 module's MAC firmware. The MAC's error state may not be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x63 / RFID_MacGetError**

Parameters:

Byte Offset	Name	Value	Description
0	ErrorType	0, 1	Retrieves the error state of the RFID PCIe M.2 module's MAC firmware. 0 = the current MAC firmware error state. 1 = the last error state reported by the MAC firmware.
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
4:1	ErrorCode	0x00000000	A 32-bit value to indicate the current or last MAC firmware error

		~ 0xFFFFFFFF	code.
7:5	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.5 Retrieving the Bootloader Firmware Version Information

Description: Retrieves the version number for the RFID PCIe M.2 module's bootloader firmware. The RFID PCIe M.2 module's bootloader firmware version may not be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x64 / RFID_MacGetBootloaderVersion**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	MajorVersion Number	0 ~ 99	The major version number.
2	MinorVersion Number	0 ~ 99	The minor version number.
3	PatchVersion Number	0 ~ 99	The patch level.
7:4	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.6 Accessing MAC-Resident OEM Configuration Data

The MAC firmware manages a portion of non-volatile memory that contains OEM configuration data. A host can access (i.e. read and write) this memory. The MAC firmware treats the OEM configuration data as a sequence of 32-bit values, storing them in its processor-native endian format (in this case, little endian). The MAC firmware accepts and returns the OEM configuration data values in the MAC's native format. For multi-byte fields in the OEM configuration area, the host program is required to convert the data to and from the host processor's native format.

When manipulating on the host system, the 32-bit values from the OEM configuration data area, a host should treat them as being in the host processor's native endian format. The host ensures that 32-bit values written to the OEM configuration data area are in the MAC firmware processor's endian format (in this case, little endian) and that when 32-bit values are read from the OEM configuration data area they are in the host processor's endian format.

4.6.6.1 Writing MAC-Resident OEM Configuration Data

Description: Writes one 32-bit value to the MAC firmware's OEM configuration data area. Note that it is the responsibility of the host programmer to ensure that the 32-bit values written to the OEM configuration data area are converted from the host-processor endian format to the MAC firmware-processor endian format before being written. The MAC firmware's OEM configuration data area may not be written while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x66 / RFID_MacWriteOemData**

Parameters:

Byte Offset	Name	Value	Description
1:0	Address	0x0000 ~ 0xFFFF	The 16-bit address into the MAC-resident OEM configuration data area where one 32-bit data word is to be written. The address is a 16-bit address and not a byte address - i.e. address 1 is actually byte 4, address 2 is actually byte 8, etc. An address that is beyond the end of the OEM configuration data area results in an invalid-parameter error.
5:2	Data	0x00000000 ~ 0xFFFFFFFF	The data is written into the MAC-resident OEM configuration data area. The 32-bit values provided must be in the MAC's native format (i.e. little endian).
7:6	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.6.2 Reading MAC-Resident OEM Configuration Data

Description: Reads one 32-bit values from the MAC firmware's OEM configuration data area. Note that the 32-bit values read from the OEM configuration data area are in the MAC firmware-processor endian format, and it is the responsibility of the host to convert to the endian format of the host processor. The MAC firmware's OEM configuration data area may not be read while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x67 / RFID_MacReadOemData**

Parameters:

Byte Offset	Name	Value	Description
1:0	Address	0x0000 ~ 0xFFFF	The 16-bit address into the MAC-resident OEM configuration data area where one 32-bit data word is to be written. The address is a 16-bit address and not a byte address - i.e. address 1 is actually byte 4, address 2 is actually byte 8, etc. An address that is beyond the end of the OEM configuration data area results in an invalid-parameter error.
7:2	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
4:1	Data	0x00000000 ~ 0xFFFFFFFF	Receive the data from MAC-resident OEM configuration data area. The 32-bit values returned are in the MAC's native format (i.e. little endian).
7:5	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.7 Accessing RFID PCIe M.2 Module Hardware Registers

4.6.7.1 Writing to an Hardware Register

The MAC firmware supports a mode, called MAC firmware bypass, which allows a host to write directly to the RFID PCIe M.2 module's hardware registers. Generally, applications do not need to perform any direct accessing of PCIe M.2 module registers. For those applications that require direct access to the underlying RFID PCIe M.2 module's hardware registers, great care must be taken, as inadvertently writing PCIe M.2 module registers may render the RFID PCIe M.2 module inoperable.

Description: Writes directly to an RFID PCIe M.2 module hardware register. The RFID PCIe M.2 module's hardware registers may not be written while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x68 / RFID_MacBypassWriteRegister**

Parameters:

Byte Offset	Name	Value	Description
1:0	Address	0x0000 ~ 0xFFFF	The 16-bit address of the RFID PCIe M.2 module hardware registers to be written. An address that is beyond the end of the RFID PCIe M.2 module registers set results in an invalid-parameter return status.
3:2	Value	0x0000 ~ 0xFFFF	The 16-bit value to write to the RFID PCIe M.2 module hardware register specified by address.
7:4	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.7.2 Reading from an Hardware Register

The MAC firmware supports a mode, called MAC firmware bypass, which allows a host to read directly from the RFID PCIe M.2 module's hardware registers.

Description: Reads directly from an RFID PCIe M.2 module hardware register. The RFID PCIe M.2 module's hardware registers may not be read while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x69 / RFID_MacBypassReadRegister**

Parameters:

Byte Offset	Name	Value	Description
1:0	Address	0x0000 ~ 0xFFFF	The 16-bit address of the RFID PCIe M.2 module hardware registers to be read. An address that is beyond the end of the RFID PCIe M.2 module registers set results in an invalid-parameter return status.
7:2	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
2:1	Value	0x0000 ~ 0xFFFF	Receive a 16-bit value in the RFID PCIe M.2 module hardware register specified by address.
7:3	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.8 Accessing Regulatory Region of Operation

The RFID PCIe M.2 module supports operating in different geographical regions. The parameters that are affected by different geographical region support include, but are not limited to, the number of unique and the frequencies of the channels used, the amount of time spent upon a particular channel, etc.
The MTI RFID PCIe M.2 module exposes two functions that allow a host to set and retrieve the region of operation.

4.6.8.1 Setting the Region of Operation

Description: Configures the RFID PCIe M.2 module's region of operation as specified. The region of operation may not be changed while an RFID PCIe M.2 module is executing a tag-protocol operation.

Note: The RFID PCIe M.2 module saves the new region value of operation to the MAC firmware's OEM configuration data area. The new region of operation will be performed after the RFID PCIe M.2 module is re-powered on.

Command: **ID: 0x6A / RFID_MacSetRegion**
Parameters:

Byte Offset	Name	Value	Description
0	RegionOperation	0 ~ 11	The new region of operation for the RFID PCIe M.2 module. 00 = United States / Canada Mexico (US) / (CA) (MX) 01 = Europe (EU) (ETSI EN 302 208) 02 = Europe 2 (EU2) (ETSI EN 300 220) 03 = Taiwan (TW) 04 = China (CN) 05 = South Korea (KR) 06 = Australia / New Zealand (AU) / (NZ) 07 = Brazil (BR) 08 = Israel (IL) 09 = India (IN) 11 = Japan (JP) 10 = Custom (Reserved for expansion.)
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.8.2 Retrieving the Region of Operation

Description: Retrieves the region of operation for which the RFID PCIe M.2 module is configured and comprises the region supportability. The region of operation may not be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x6B / RFID_MacGetRegion**
Parameters:

Byte	Name	Value	Description

Offset			
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	RegionOperation	0 ~ 11	Receive the current region of operation. 00 = United States / Canada Mexico (US) / (CA) (MX) 01 = Europe (EU) (ETSI EN 302 208) 02 = Europe 2 (EU2) (ETSI EN 300 220) 03 = Taiwan (TW) 04 = China (CN) 05 = South Korea (KR) 06 = Australia / New Zealand (AU) / (NZ) 07 = Brazil (BR) 08 = Israel (IL) 09 = India (IN) 11 = Japan (JP) 10 = Custom (Reserved for expansion.)
5:2	RegionSupport	0x00000000 ~ 0xFFFFFFFF	The region supportability. 0 = the region is not supportable 1 = the region is supportable **Bit** / **Region Name** 0: United States / Canada Mexico (US) / (CA) (MX) 1: Europe (EU) (ETSI EN 302 208) 2: Europe 2 (EU2) (ETSI EN 300 220) 3: Taiwan (TW) 4: China (CN) 5: South Korea (KR) 6: Australia / New Zealand (AU) / (NZ) 7: Brazil (BR) 8: Israel (IL) 9: India (IN) 10: Custom (Reserved for expansion.) 11: Japan (JP)
7:6	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.9 Retrieving the MAC-Resident OEMCfg Version Information

Description: Retrieves the version number for the RFID PCIe M.2 module's MAC-resident OEMCfg. The RFID PCIe M.2 module's MAC-resident OEMCfg version may not be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x6C / RFID_MacGetOEMCfgVersion**

Parameters:

Byte Offset	Name	Value	Description

7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	MajorVersion Number	0x30 ~ 0x39	The major version number. Number type is ASCII code, i.e. '0' ~ '9'.
2	MinorVersion Number1	0x30 ~ 0x39	The tens digit of minor version number. Number type is ASCII code, i.e. '0' ~ '9'.
3	MinorVersion Number2	0x30 ~ 0x39	The units digit of minor version number. Number type is ASCII code, i.e. '0' ~ '9'.
7:4	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.10 Retrieving the MAC-Resident OEMCfg Update Number Information

Description: Retrieves the update number for the RFID PCIe M.2 module's MAC-resident OEMCfg. The RFID PCIe M.2 module's MAC-resident OEMCfg update number may not be retrieved while an RFID PCIe M.2 module is executing a tag-protocol operation.

Command: **ID: 0x6D / RFID_MacGetOEMCfgUpdateNumber**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	Update Number1	0x30 ~ 0x39	The tens digit of update number. Number type is ASCII code, i.e. '0' ~ '9'.
2	Update Number2	0x30 ~ 0x39	The units digit of update number. Number type is ASCII code, i.e. '0' ~ '9'.
7:3	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.7 RFID PCIe M.2 Module Region Test Support

The RFID PCIe M.2 module supports several test modes. Specifically, the host can have direct control over the RFID PCIe M.2 module's transmission of carrier wave signal through some of the following commands.

IMPORTANT: When the host completes the tests, the host must issue RFID_ControlSoftReset or follow the important note of each test command to make the RFID PCIe M.2 module to accord the regulatory operation of region duration executing ISO 18000-6C tag-protocol operations.

4.6.11 Accessing the Test Antenna-Port Configuration

Note: When the host performs "ISO 18000-6C Tag-Protocol Operation commands" for test, the host should perform RFID_AntennaPortSetConfiguration to change the parameters of antennas instead.

4.6.11.1 Setting the Test Antenna-Port Configuration

Description: Configures the PCIe M.2 module's power for the physical antenna port. The antenna-port configuration for testing may not be set while it is executing a tag-protocol operation.

Note: The antenna-port configuration for test purposes only affects relative operations of region test support functions. It is not available for configuration duration executing ISO 18000-6C tag-protocol operations.

Command: **ID: 0x80 / RFID_TestSetAntennaPortConfiguration**

Parameters:

Byte Offset	Name	Value	Description
0	PhysicalPort	0	The physical transmit/receive port. 0 = ANT1
2:1	PowerLevel	0 ~ 300	The power level for the physical antenna port. This value is specified in 0.1 (i.e. 1/10th) dBm. **Note:** Not all RFID PCIe M.2 modules support setting an antenna port's power level at 1/10th dBm resolutions. The dBm rounding/truncation policy is left to the RFID PCIe M.2 module and is outside the scope of the MTI RFID PCIe M.2 module.
7:3	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.11.2 Retrieving the Test Antenna-Port Configuration

Description: Retrieves the PCIe M.2 module's power of physical antenna port. The antenna-port configuration for testing may not be retrieved while it is executing a tag-protocol operation.

Command: **ID: 0x81 / RFID_TestGetAntennaPortConfiguration**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	PhysicalPort	0	The physical transmit/receive port. 0 = ANT1
3:2	PowerLevel	0 ~ 300	The power level for the physical antenna port. This value is specified in 0.1 (i.e. 1/10th) dBm.
7:4	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.12 Accessing the Test Frequency Configuration

4.6.12.1 Setting the Test Frequency Configuration

Description: Configures the PCIe M.2 module's frequency within region of operation for fixed channel operation. When the frequency channel operation is configured to single channel, the RFID reader/module will transmit carrier wave with fixed channel. In this case, even if performing ISO 18000-6C tag-protocol operations, the RFID reader/module will also transmit carrier wave with fixed channel. The frequency configuration for testing may not be set while it is executing a tag-protocol operation.

IMPORTANT: After the host completes the test purpose, the channel flag must be set to region operation to accord the regulatory operation of region duration executing ISO 18000-6C tag-protocol operations.

Command: **ID: 0x82 / RFID_TestSetFrequencyConfiguration**

Parameters:

Byte Offset	Name	Value	Description
0	ChannelFlag	0, 1	The frequency channel operation. 0 = region operation 1 = single channel
4:1	ExactFrequecny	0x00000000 ~ 0xFFFFFFFF	The frequency in kHz.
7:5	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.12.2 Retrieving the Test Frequency Configuration

Description: Retrieves the PCIe M.2 module's frequency within the region of operation for fixed channel operation. The frequency configuration for testing may not be retrieved while it is executing a tag-protocol operation.

Command: **ID: 0x83 / RFID_TestGetFrequencyConfiguration**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	ChannelFlag	0, 1	Receive the frequency channel operation. 0 = region operation 1 = single channel
5:2	ExactFrequecny	0x00000000 ~ 0xFFFFFFFF	Receive the frequency value. The frequency is specified in kHz.
7:6	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.13 Accessing the Pulse Time of Random Data Transmission

4.6.13.1 Setting the Pulse Time of Random Data Transmission

Description: Configures the transmitting random data pulsing on/off time. Timing adjustment may need to be made to compensate for inherent on/off delays. Set to a value of zero to measure the minimum on/off time. The pulse time of random data transmission may not be set while it is executing a tag-protocol operation.

Command: **ID: 0x84 / RFID_TestSetRandomDataPulseTime**

Parameters:

Byte Offset	Name	Value	Description
1:0	OnTime	0x0000 ~ 0xFFFF	The duration of pulsing on time in microseconds. Note the resolution is limited to (32 bits * R-T Calculation / 2) us for a specific link profile.
3:2	OffTime	0x0000 ~ 0xFFFF	The duration of pulsing off time in microseconds.
7:4	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.13.2 Retrieving the Pulse Time of Random Data Transmission

Description: Retrieves the transmitting random data pulsing on/off time. The pulse time of random data transmission may not be retrieved while it is executing a tag-protocol operation.

Command: **ID: 0x85 / RFID_TestGetRandomDataPulseTime**

Parameters:

Byte	Name	Value	Description

Offset			
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
2:1	OnTime	0x0000 ~ 0xFFFF	Receive the duration value of pulsing on time. The on time is specified in microseconds.
4:3	OffTime	0x0000 ~ 0xFFFF	Receive the duration value of pulsing off time. The off time is specified in microseconds.
7:5	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.14 Accessing the Test Inventory Configuration

The test inventory means performing RFID_18K6CTagInventory for test. This command doesn't affect any non-inventory ISO 18000-6C tag access operations (i.e. read, write, etc.).

4.6.14.1 Setting the Test Inventory Configuration

Description: Configures the PCIe M.2 module's transmission on/off operation duration inventory. If the continuous operation for inventory is enabled, the host must stop an inventory by calling RFID_ControlCancel on another thread. The inventory configuration for testing may not be set while it is executing a tag-protocol operation.

IMPORTANT: After the host completed the special test purpose, the continuous operation must be disabled to accord the regulatory operation of region duration executing ISO 18000-6C tag-protocol operations.

Command: **ID: 0x86 / RFID_TestSetInventoryConfiguration**

Parameters:

Byte Offset	Name	Value	Description
0	ContinuousOperation	0, 1	The continuous operation for inventory. If the continuous operation is enabled, the PCIe M.2 module ignores the transmission on/off action and will keep the transmitter turned on for entire duration of inventory. 0 = disabled 1 = enabled
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.14.2 Retrieving the Test Inventory Configuration

Description: Retrieves the PCIe M.2 module's transmission on/off operation duration inventory for operating

RFID_18K6CTagInventory in test mode.. The inventory configuration for testing may not be retrieved while it is executing a tag-protocol operation.

Command: **ID: 0x87 / RFID_TestGetInventoryConfiguration**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
1	ContinuousOperation	0, 1	Receive the continuous operation of inventory. 0 = disabled 1 = enabled
7:2	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.15 CW Control with No Data Modulation

The RFID PCIe M.2 module transmitter and carrier wave output can be controlled by the host. When the RFID_TestTurnOnCarrierWave command is called, continuous CW with no data modulation is broadcast until the PCIe M.2 module is either reset or the CW is turned off.

4.6.15.1 Turning On the Radio CW

Description: Turns on the PCIe M.2 module's carrier wave with no data modulation. The PCIe M.2 module's carrier wave may not be turned on while it is executing a tag-protocol operation.

Command: **ID: 0x88 / RFID_TestTurnOnCarrierWave**

Parameters:

Byte Offset	Name	Value	Description
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.15.2 Turning Off the Radio CW

Description: Turns off the PCIe M.2 module's carrier wave. The PCIe M.2 module's carrier wave may not be turned off while it is executing a tag-protocol operation.

Command: **ID: 0x89 / RFID_TestTurnOffCarrierWave**

Parameters:

Byte	Name	Value	Description

Offset			
7:0	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.16 Injecting a Random Data

The RFID PCIe M.2 module transmitter can be manually enabled to broadcast CW and then inject the random data to generate data modulation duration CW turned on by the host.

Description: Injects the random data duration the PCIe M.2 module's carrier wave is turned on, for the specified bit count. The PCIe M.2 module's random data may not be injected while it is executing a tag-protocol operation.

Command: **ID: 0x8A / RFID_TestInjectRandomData**

Parameters:

Byte Offset	Name	Value	Description
3:0	Count	0x00000000 ~ 0xFFFFFFFF	Number of random data bits. A value of 0 will send the maximum number of bits (4096 * 16).
7:4	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.6.17 Transmitting a Random Data with Modulation

The RFID PCIe M.2 module transmitter can be enabled to broadcast CW with random data modulation by the host. The random data can either generated internally by the PCIe M.2 module or a repeating 511-bit sequence data. When the RFID_TestTransmitRandomData command is called, continuous CW with random data modulation is broadcast until the PCIe M.2 module's CW is cancelled (see Section 4.5.1).

Description: Turns on the PCIe M.2 module's carrier wave with random data modulation, for the specified duration. The PCIe M.2 module's random data may not be transmitted while it is executing a tag-protocol operation.

Command: **ID: 0x8B / RFID_TestTransmitRandomData**

Parameters:

Byte Offset	Name	Value	Description
0	Control	0, 1	The control parameter for the random data transmitted. If the

			control is pulsing random data, the pulse time is depended on setting values of RFID_TestSetRandomDataPulseTime. 0 = continuous 1 = pulsing
4:1	Duration	0x00000000 ~ 0xFFFFFFFF	The duration, in milliseconds, of how long the random CW should be transmitted. A value of 0 will run forever, until the command is cancelled or the radio module is reset.
5	RandomType	0, 1	The random type parameter for the random data transmitted. 0 = the PCIe M.2 module internal random data 1 = fixed 511-bit random pattern (x^9+x^4+1) **Note:** The fixed 511-bit random pattern, a repeating 511-bit sequence of FF83DF1732094ED1E7CD8A91C6D5C4C44021184E5586F4DC8A15 A7EC92DF93533018CA34BFA2C759678FBA0D6DD82D7D540A579 77039D27AEA243385ED9A1DE0, with a continuous repeating 9-bit maximum length sequence with polynomial x^9+x^4+1.
7:6	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
7:1	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

Report Packet: Command-Begin / Command-End two packet types.
See Section 3.2.3 to know detailed information about two report packets.

4.7 RFID PCIe M.2 Module Engineering Test Support

4.7.1 Retrieving the Temperature

Description: Retrieves the temperature of RFID PCIe M.2 module working. It can be used as an estimate of the RFID PCIe M.2 module's case temperature.

Command: **ID: 0x91 / RFID_EngGetTemperature**

Parameters:

Byte Offset	Name	Value	Description
0	Source	0, 1	Indicates a selection sensor. 0 = PA temperature sensor 1 = Ambient temperature sensor
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
2:1	Temperature	0x0000 ~ 0xFFFF	A temperature value of the RFID PCIe M.2 module, in units of degree-C. This is a two's complement representation.
7:3	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

4.7.2 Retrieving the RF Power Level

Description: Retrieves the RF power level of RFID PCIe M.2 module output.

Command: **ID: 0x93 / RFID_EngGetRFPower**

Parameters:

Byte Offset	Name	Value	Description
0	Source	0, 1	Indicates a selection detector. 0 = Forward RF power detector 1 = Reverse RF power detector
7:1	padding	0	Pad 0x00 is added to end of parameters filed to force the parameters to stuff 8 bytes length.

Returned Data:

Byte Offset	Name	Value	Description
0	Status	0x00/F0/FF	Performed result of command.
2:1	PowerLevel	0x0000 ~ 0xFFFF	Represents the RF power level, measured in 0.1 dBm units. This is a two's complement representation.
7:3	padding	0	Pad 0x00 is added to end of returned data filed to force the returned data to stuff 8 bytes length.

5 Use Cases

5.1 Initial Configuration of the RFID PCIe M.2 Module

[Step 1]

Purpose: Sets the operation mode to continuous mode.
Command: RFID_RadioSetOperationMode (ID: 0x02)
Parameters: Mode (0x00)
Returned Data: RFID_STATUS_OK

[Step 2]

Purpose: Sets the RF power level of the logical antenna port 0 to 30.0dBm.
Command: RFID_AntennaPortSetConfiguration (ID: 0x12)
Parameters: AntennaPort (0x00) / PowerLevel (0x012C) / DwellTime (0x0000) / NumberInventoryCycles (0x2000) / PhysicalPort (0x00)
Returned Data: RFID_STATUS_OK

Table 5.1 - Host Transmits and Module Replies of Use Case 1

Step	Data Flow	Command and Parameters / Response packet / Report packet
1a	Host => Module	43 49 54 4D FF 02 00 00 00 00 00 00 00 00 92 C7
1b	Host <= Module	52 49 54 4D 00 02 00 00 00 00 00 00 00 00 00 17
2a	Host => Module	43 49 54 4D FF 12 00 2C 01 00 00 00 20 00 B7 EB
2b	Host <= Module	52 49 54 4D 00 12 00 00 00 00 00 00 00 00 FE 44

5.2 Inventory for a Single Tag

[Step 1]
Purpose: Sets the operation mode to continuous mode.
Command: RFID_RadioSetOperationMode (ID: 0x02)
Parameters: Mode (0x00)
Returned Data: RFID_STATUS_OK

[Step 2]
Purpose: Sets the RF power level of the logical antenna port 0 to 30.0dBm.
Command: RFID_AntennaPortSetConfiguration (ID: 0x12)
Parameters: AntennaPort (0x00) / PowerLevel (0x012C) / DwellTime (0x0000) / NumberInventoryCycles (0x2000) / PhysicalPort (0x00)
Returned Data: RFID_STATUS_OK

[Step 3]
Purpose: Sets the currently-active singulation algorithm to fixed-Q singulation algorithm.
Command: RFID_18K6CSetCurrentSingulationAlgorithm (ID: 0x32)
Parameters: Algorithm (0x00)
Returned Data: RFID_STATUS_OK

[Step 4]
Purpose: Configures the settings of fixed-Q singulation algorithm.
Command: RFID_18K6CSetSingulationAlgorithmParameters (ID: 0x34)
Parameters: Algorithm (0x00) / QValue (0x03) / RetryCount (0x00) / ToggleTarget (0x01) / RepeatUntilNoTags (0x00) / RFU (0x00)
Returned Data: RFID_STATUS_OK

[Step 5]
Purpose: Executes a tag inventory for a single tag.
Command: RFID_18K6CTagInventory (ID: 0x40)
Parameters: PerformSelect (0x00) / PerformPostMatch (0x00)
Returned Data: RFID_STATUS_OK
Report Packet: Three packet types are as follows,
Command-Begin packet (see step 5c)
Inventory-Response packet (see step 5d/5e/5f/5g)
Command-End packet (see step 5h)

[Step 6]
Purpose: Stops currently-executing RFID_18K6CTagInventory operation.
Command: RFID_ControlCancel (ID: 0x50)
Parameters: padding
Returned Data: RFID_STATUS_OK

Table 5.2 - Host Transmits and Module Replies of Use Case 2

Step	Data Flow	Command and Parameters / Response packet / Report packet
1a	Host => Module	43 49 54 4D FF 02 00 00 00 00 00 00 00 92 C7
1b	Host <= Module	52 49 54 4D 00 02 00 00 00 00 00 00 00 00 17
2a	Host => Module	43 49 54 4D FF 12 00 2C 01 00 00 00 20 00 B7 EB
2b	Host <= Module	52 49 54 4D 00 12 00 00 00 00 00 00 00 FE 44
3a	Host => Module	43 49 54 4D FF 32 00 00 00 00 00 00 00 90 33
3b	Host <= Module	52 49 54 4D 00 32 00 00 00 00 00 00 00 02 E3
4a	Host => Module	43 49 54 4D FF 34 00 03 00 01 00 00 00 00 CB 1B
4b	Host <= Module	52 49 54 4D 00 34 00 00 00 00 00 00 00 8A B9
5a	Host => Module	43 49 54 4D FF 40 00 00 00 00 00 00 00 2C 5E
5b	Host <= Module	52 49 54 4D 00 40 00 00 00 00 00 00 00 BE 8E
5c	Host <= Module	42 49 54 4D 01 01 01 01 00 00 02 00 00 00 0F 00 00 00 35 00 14 00 D7 CE

5d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 45 00 14 00 6B 9D 86 32 DE FE 00 00 30 00 11 11 22 22 33 33 44 44 55 55 66 66 18 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5E A4
5e	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 02 00 D5 01 14 00 6F A6 86 32 F9 FE 00 00 30 00 11 11 22 22 33 33 44 44 55 55 66 66 18 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 DF 6C
6a	Host => Module	43 49 54 4D FF 50 00 00 00 00 00 00 00 00 D2 0D
5f	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 03 00 6D 03 14 00 71 A6 86 32 09 FF 00 00 30 00 11 11 22 22 33 33 44 44 55 55 66 66 18 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 2C B5
5g	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 04 00 F8 04 14 00 70 A9 86 32 FF FE 00 00 30 00 11 11 22 22 33 33 44 44 55 55 66 66 18 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4A C2
5h	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 05 00 F9 04 14 00 00 00 00 00 AD 87

5.3 Read the PC Value from EPC Memory Bank of a Tag

[Step 1]
Purpose: Sets the RF power level of the logical antenna port 0 to 30.0dBm.
Command: RFID_AntennaPortSetConfiguration (ID: 0x12)
Parameters: AntennaPort (0x00) / PowerLevel (0x012C) / DwellTime (0x0000) / NumberInventoryCycles (0x2000) / PhysicalPort (0x00)
Returned Data: RFID_STATUS_OK

[Step 2]
Purpose: Sets the currently-active singulation algorithm to fixed-Q singulation algorithm.
Command: RFID_18K6CSetCurrentSingulationAlgorithm (ID: 0x32)
Parameters: Algorithm (0x00)
Returned Data: RFID_STATUS_OK

[Step 3]
Purpose: Configures the settings of fixed-Q singulation algorithm.
Command: RFID_18K6CSetSingulationAlgorithmParameters (ID: 0x34)
Parameters: Algorithm (0x00) / QValue (0x03) / RetryCount (0x00) / ToggleTarget (0x00) / RepeatUntilNoTags (0x00) / RFU (0x00)
Returned Data: RFID_STATUS_OK

[Step 4]
Purpose: Executes a tag read for a single tag.
Command: RFID_18K6CTagRead (ID: 0x41)
Parameters: Bank (0x01) / Offset (0x0001) / Count (0x01) / RetryCount (0x01) / PerformSelect (0x00) / PerformPostMatch (0x00)
Returned Data: RFID_STATUS_OK
Report Packet: Four packet types are as follows,
Command-Begin packet (see step 4c)
Inventory-Response packet (see step 4d)
Tag-Access packet (see step 4e)
Command-End packet (see step 4f)

Table 5.5 - Host Transmits and Module Replies of Use Case 3

Step	Data Flow	Command and Parameters / Response packet / Report packet
1a	Host => Module	43 49 54 4D FF 12 00 2C 01 00 00 00 20 00 B7 EB
1b	Host <= Module	52 49 54 4D 00 12 00 00 00 00 00 00 00 00 FE 44
2a	Host => Module	43 49 54 4D FF 32 00 00 00 00 00 00 00 00 90 33
2b	Host <= Module	52 49 54 4D 00 32 00 00 00 00 00 00 00 00 02 E3
3a	Host => Module	43 49 54 4D FF 34 00 03 00 00 00 00 00 00 9A B1
3b	Host <= Module	52 49 54 4D 00 34 00 00 00 00 00 00 00 00 8A B9
4a	Host => Module	43 49 54 4D FF 41 01 01 00 01 01 00 00 00 58 96
4b	Host <= Module	52 49 54 4D 00 41 00 00 00 00 00 00 00 00 9D 65
4c	Host <= Module	42 49 54 4D 01 01 01 00 00 00 02 00 00 00 10 00 00 00 AD 29 0A 00 2A FF
4d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 BB 29 0A 00 65 94 86 32 00 00 00 00 30 00 E2 00 34 11 B8 02 01 15 04 34 61 70 98 A4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 8B B1
4e	Host <= Module	41 49 54 4D 01 01 01 80 06 00 04 00 02 00 BF 29 0A 00 C2 00 00 00 00 00 00 00 **30 00** 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 9C 1A
4f	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 03 00 C4 29 0A 00 00 00 00 00 71 FA

5.4 Read the EPC Value from EPC Memory Bank of a Tag

[Step 1]

Purpose: Sets the RF power level of the logical antenna port 0 to 30.0dBm.
Command: RFID_AntennaPortSetConfiguration (ID: 0x12)
Parameters: AntennaPort (0x00) / PowerLevel (0x012C) / DwellTime (0x0000) / NumberInventoryCycles (0x2000) / PhysicalPort (0x00)
Returned Data: RFID_STATUS_OK

[Step 2]

Purpose: Sets the currently-active singulation algorithm to fixed-Q singulation algorithm.
Command: RFID_18K6CSetCurrentSingulationAlgorithm (ID: 0x32)
Parameters: Algorithm (0x00)
Returned Data: RFID_STATUS_OK

[Step 3]

Purpose: Configures the settings of fixed-Q singulation algorithm.
Command: RFID_18K6CSetSingulationAlgorithmParameters (ID: 0x34)
Parameters: Algorithm (0x00) / QValue (0x03) / RetryCount (0x00) / ToggleTarget (0x00) / RepeatUntilNoTags (0x00) / RFU (0x00)
Returned Data: RFID_STATUS_OK

[Step 4]

Purpose: Executes a tag read for a single tag.
Command: RFID_18K6CTagRead (ID: 0x41)
Parameters: Bank (0x01) / Offset (0x0002) / Count (0x06) / RetryCount (0x01) / PerformSelect (0x00) / PerformPostMatch (0x00)
Returned Data: RFID_STATUS_OK
Report Packet: Four packet types are as follows,
Command-Begin packet (see step 4c)
Inventory-Response packet (see step 4d)
Tag-Access packet (see step 4e)
Command-End packet (see step 4f)

Table 5.6 - Host Transmits and Module Replies of Use Case 4

Step	Data Flow	Command and Parameters / Response packet / Report packet
1a	Host => Module	43 49 54 4D FF 12 00 2C 01 00 00 00 20 00 B7 EB
1b	Host <= Module	52 49 54 4D 00 12 00 00 00 00 00 00 00 00 FE 44
2a	Host => Module	43 49 54 4D FF 32 00 00 00 00 00 00 00 00 90 33
2b	Host <= Module	52 49 54 4D 00 32 00 00 00 00 00 00 00 00 02 E3
3a	Host => Module	43 49 54 4D FF 34 00 03 00 00 00 00 00 00 9A B1
3b	Host <= Module	52 49 54 4D 00 34 00 00 00 00 00 00 00 00 8A B9
4a	Host => Module	43 49 54 4D FF 41 01 02 00 06 01 00 00 00 0E 29
4b	Host <= Module	52 49 54 4D 00 41 00 00 00 00 00 00 00 00 9D 65
4c	Host <= Module	42 49 54 4D 01 01 01 00 00 00 02 00 00 00 10 00 00 00 8F 15 0F 00 3D 6A
4d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 A2 15 0F 00 69 96 86 32 00 00 00 00 30 00 E2 00 34 11 B8 02 01 15 04 34 61 70 98 A4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6A F5
4e	Host <= Module	41 49 54 4D 01 01 01 00 06 00 06 00 02 00 A7 15 0F 00 C2 00 00 00 00 00 00 00 **E2 00 34 11 B8 02 01 15 04 34 61 70** 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 29 17
4f	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 03 00 A7 15 0F 00 00 00 00 00 5B F7

5.5 Write the EPC Value to EPC Memory Bank of a Tag

[Step 1]
Purpose: Sets the RF power level of the logical antenna port 0 to 30.0dBm.
Command: RFID_AntennaPortSetConfiguration (ID: 0x12)
Parameters: AntennaPort (0x00) / PowerLevel (0x012C) / DwellTime (0x0000) / NumberInventoryCycles (0x2000) / PhysicalPort (0x00)
Returned Data: RFID_STATUS_OK

[Step 2]
Purpose: Sets the currently-active singulation algorithm to fixed-Q singulation algorithm.
Command: RFID_18K6CSetCurrentSingulationAlgorithm (ID: 0x32)
Parameters: Algorithm (0x00)
Returned Data: RFID_STATUS_OK

[Step 3]
Purpose: Configures the settings of fixed-Q singulation algorithm.
Command: RFID_18K6CSetSingulationAlgorithmParameters (ID: 0x34)
Parameters: Algorithm (0x00) / QValue (0x03) / RetryCount (0x00) / ToggleTarget (0x00) / RepeatUntilNoTags (0x00) / RFU (0x00)
Returned Data: RFID_STATUS_OK

[Step 4]
Purpose: Executes a tag write for a single tag.
Command: RFID_18K6CTagWrite (ID: 0x42)
Parameters: Bank (0x01) / Offset (0x0002) / Data (0xABCD) / RetryCount (0x01) / PerformSelect (0x00) / PerformPostMatch (0x00)
Returned Data: RFID_STATUS_OK
Report Packet: Four packet types are as follows,
Command-Begin packet (see step 4c)
Inventory-Response packet (see step 4d)
Tag-Access packet (see step 4e)
Command-End packet (see step 4f)

If the host doesn't need to check the data of previous write, skip step 5.

[Step 5]
Purpose: Executes a tag read for a single tag in order to check the previous write command.
Command: RFID_18K6CTagRead (ID: 0x41)
Parameters: Bank (0x01) / Offset (0x0002) / Count (0x01) / RetryCount (0x01) / PerformSelect (0x00) / PerformPostMatch (0x00)
Returned Data: RFID_STATUS_OK
Report Packet: Four packet types are as follows,
Command-Begin packet (see step 5c)
Inventory-Response packet (see step 5d)
Tag-Access packet (see step 5e)
Command-End packet (see step 5f)

Table 5.7 - Host Transmits and Module Replies of Use Case 5

Step	Data Flow	Command and Parameters / Response packet / Report packet
1a	Host => Module	43 49 54 4D FF 12 00 2C 01 00 00 00 20 00 B7 EB
1b	Host <= Module	52 49 54 4D 00 12 00 00 00 00 00 00 00 00 FE 44
2a	Host => Module	43 49 54 4D FF 32 00 00 00 00 00 00 00 00 90 33
2b	Host <= Module	52 49 54 4D 00 32 00 00 00 00 00 00 00 00 02 E3
3a	Host => Module	43 49 54 4D FF 34 00 03 00 00 00 00 00 00 9A B1
3b	Host <= Module	52 49 54 4D 00 34 00 00 00 00 00 00 00 00 8A B9
4a	Host => Module	43 49 54 4D FF 42 01 02 00 CD AB 01 00 00 E0 6E
4b	Host <= Module	52 49 54 4D 00 42 00 00 00 00 00 00 00 00 D9 48

4c	Host <= Module	42 49 54 4D 01 01 01 00 00 00 02 00 00 00 11 00 00 00 A1 63 01 00 5D F1
4d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 C3 63 01 00 67 97 86 32 00 00 00 00 30 00 E2 00 34 11 B8 02 01 15 04 34 61 70 98 A4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 DD 5D
4e	Host <= Module	41 49 54 4D 01 01 01 00 06 00 03 00 02 00 CD 63 01 00 C3 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0D 99
4f	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 03 00 D0 63 01 00 00 00 00 00 A8 25
5a	Host => Module	43 49 54 4D FF 41 01 02 00 01 01 00 00 00 DA 4E
5b	Host <= Module	52 49 54 4D 00 41 00 00 00 00 00 00 00 00 9D 65
5c	Host <= Module	42 49 54 4D 01 01 01 00 00 00 02 00 00 00 10 00 00 00 38 24 03 00 39 7D
5d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 46 24 03 00 67 95 86 32 00 00 00 00 30 00 AB CD 34 11 B8 02 01 15 04 34 61 70 BB C6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F2 D9
5e	Host <= Module	41 49 54 4D 01 01 01 80 06 00 04 00 02 00 4A 24 03 00 C2 00 00 00 00 00 00 00 **AB CD** 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 36 E7
5f	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 03 00 50 24 03 00 00 00 00 00 C1 97

5.6 Write Once the Full EPC Value to EPC Memory Bank of a Tag

[Step 1]
Purpose: Sets the RF power level of the logical antenna port 0 to 30.0dBm.
Command: RFID_AntennaPortSetConfiguration (ID: 0x12)
Parameters: AntennaPort (0x00) / PowerLevel (0x012C) / DwellTime (0x0000) / NumberInventoryCycles (0x2000) / PhysicalPort (0x00)
Returned Data: RFID_STATUS_OK

[Step 2]
Purpose: Sets the currently-active singulation algorithm to fixed-Q singulation algorithm.
Command: RFID_18K6CSetCurrentSingulationAlgorithm (ID: 0x32)
Parameters: Algorithm (0x00)
Returned Data: RFID_STATUS_OK

[Step 3]
Purpose: Configures the settings of fixed-Q singulation algorithm.
Command: RFID_18K6CSetSingulationAlgorithmParameters (ID: 0x34)
Parameters: Algorithm (0x00) / QValue (0x03) / RetryCount (0x00) / ToggleTarget (0x00) / RepeatUntilNoTags (0x00) / RFU (0x00)
Returned Data: RFID_STATUS_OK

[Step 4]
Purpose: Sets the tag write data buffer with the full EPC value (six words data).
Command: RFID_18K6CSetTagWriteDataBuffer (ID: 0x38)
Parameters: BufferIndex (0x00 ~ 0x05) / BufferData (0x35E0, 0x0011, 0x2233, 0x4455, 0x6677, 0x8899)
Returned Data: RFID_STATUS_OK

[Step 5]
Purpose: Executes a tag multiple write for a single tag.
Command: RFID_18K6CTagMultipleWrite (ID: 0x45)
Parameters: Bank (0x01) / Offset (0x0002) / DataLength (0x06) / Reserved (0x00) / RetryCount (0x01) / PerformSelect (0x00) / PerformPostMatch (0x00)
Returned Data: RFID_STATUS_OK
Report Packet: Four packet types are as follows,
Command-Begin packet (see step 5c)
Inventory-Response packet (see step 5d)
Tag-Access packet (see step 5e)
Command-End packet (see step 5f)

If the host doesn’t need to check the data of previous write, skip step 6.

[Step 6]
Purpose: Executes a tag read for a single tag in order to check the previous write command.
Command: RFID_18K6CTagRead (ID: 0x41)
Parameters: Bank (0x01) / Offset (0x0002) / Count (0x06) / RetryCount (0x01) / PerformSelect (0x00) / PerformPostMatch (0x00)
Returned Data: RFID_STATUS_OK
Report Packet: Four packet types are as follows,
Command-Begin packet (see step 6c)
Inventory-Response packet (see step 6d)
Tag-Access packet (see step 6e)
Command-End packet (see step 6f)

Table 5.8 - Host Transmits and Module Replies of Use Case 6

Step	Data Flow	Command and Parameters / Response packet / Report packet
1a	Host => Module	43 49 54 4D FF 12 00 2C 01 00 00 00 20 00 B7 EB
1b	Host <= Module	52 49 54 4D 00 12 00 00 00 00 00 00 00 00 FE 44
2a	Host => Module	43 49 54 4D FF 32 00 00 00 00 00 00 00 00 90 33

2b	Host <= Module	52 49 54 4D 00 32 00 00 00 00 00 00 00 02 E3
3a	Host => Module	43 49 54 4D FF 34 00 03 00 00 00 00 00 9A B1
3b	Host <= Module	52 49 54 4D 00 34 00 00 00 00 00 00 00 8A B9
4a	Host => Module	43 49 54 4D FF 38 00 E0 35 00 00 00 00 66 A8
4b	Host <= Module	52 49 54 4D 00 38 00 00 00 00 00 00 00 9A 0C
4c	Host => Module	43 49 54 4D FF 38 01 11 00 00 00 00 00 C1 14
4d	Host <= Module	52 49 54 4D 00 38 00 00 00 00 00 00 00 9A 0C
4e	Host => Module	43 49 54 4D FF 38 02 33 22 00 00 00 00 E9 6C
4f	Host <= Module	52 49 54 4D 00 38 00 00 00 00 00 00 00 9A 0C
4g	Host => Module	43 49 54 4D FF 38 03 55 44 00 00 00 00 FC EB
4h	Host <= Module	52 49 54 4D 00 38 00 00 00 00 00 00 00 9A 0C
4i	Host => Module	43 49 54 4D FF 38 04 77 66 00 00 00 00 B9 9C
4j	Host <= Module	52 49 54 4D 00 38 00 00 00 00 00 00 00 9A 0C
4k	Host => Module	43 49 54 4D FF 38 05 99 88 00 00 00 00 9A FA
4l	Host <= Module	52 49 54 4D 00 38 00 00 00 00 00 00 00 9A 0C
5a	Host => Module	43 49 54 4D FF 45 01 02 00 06 00 01 00 00 65 F4
5b	Host <= Module	52 49 54 4D 00 45 00 00 00 00 00 00 00 72 F9
5c	Host <= Module	42 49 54 4D 01 01 01 00 00 00 02 00 00 00 11 00 00 00 55 E1 01 00 DB F1
5d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 72 E1 01 00 68 98 86 32 00 00 00 00 30 00 AB CD 34 11 B8 02 01 15 04 34 61 70 BB C6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4A F6
5e	Host <= Module	41 49 54 4D 01 01 01 00 06 00 03 00 02 00 B0 E1 01 00 C3 00 00 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 64 EF
5f	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 03 00 B6 E1 01 00 00 00 00 00 C1 E3
6a	Host => Module	43 49 54 4D FF 41 01 02 00 06 01 00 00 00 0E 29
6b	Host <= Module	52 49 54 4D 00 41 00 00 00 00 00 00 00 9D 65
6c	Host <= Module	42 49 54 4D 01 01 01 00 00 00 02 00 00 00 10 00 00 00 E2 91 03 00 9E A8
6d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 F3 91 03 00 67 95 86 32 00 00 00 00 30 00 35 E0 00 11 22 33 44 55 66 77 88 99 09 43 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B9 31
6e	Host <= Module	41 49 54 4D 01 01 01 00 06 00 06 00 02 00 F8 91 03 00 C2 00 00 00 00 00 00 00 **35 E0 00 11 22 33 44 55 66 77 88 99** 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 DB C6
6f	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 03 00 FA 91 03 00 00 00 00 00 87 0B

5.7 Kill Single Tag

[Step 1]
Purpose: Sets the RF power level of the logical antenna port 0 to 30.0dBm.
Command: RFID_AntennaPortSetConfiguration (ID: 0x12)
Parameters: AntennaPort (0x00) / PowerLevel (0x012C) / DwellTime (0x0000) / NumberInventoryCycles (0x2000) / PhysicalPort (0x00)
Returned Data: RFID_STATUS_OK

[Step 2]
Purpose: Sets the currently-active singulation algorithm to fixed-Q singulation algorithm.
Command: RFID_18K6CSetCurrentSingulationAlgorithm (ID: 0x32)
Parameters: Algorithm (0x00)
Returned Data: RFID_STATUS_OK

[Step 3]
Purpose: Configures the settings of fixed-Q singulation algorithm.
Command: RFID_18K6CSetSingulationAlgorithmParameters (ID: 0x34)
Parameters: Algorithm (0x00) / QValue (0x00) / RetryCount (0x00) / ToggleTarget (0x00) / RepeatUntilNoTags (0x01) / RFU (0x00)
Returned Data: RFID_STATUS_OK

[Step 4]
Purpose: Reads the kill password value from the Kill Password location of RESERVED memory bank of the 6C tag.
Command: RFID_18K6CTagRead (ID: 0x41)
Parameters: Bank (0x00) / Offset (0x0000) / Count (0x02) / RetryCount (0x01) / PerformSelect (0x00) / PerformPostMatch (0x00)
Returned Data: RFID_STATUS_OK
Report Packet: Four packet types are as follows,

Command-Begin packet	(see step 4c)
Inventory-Response packet	(see step 4d)
Tag-Access packet	(see step 4e)
Command-End packet	(see step 4f)

A tag whose kill password value is zero will not execute a kill operation; if such a tag receives an RFID_18K6CTagKill, it ignores this command and backscatters an error code.
If the tag's kill password is zero, perform step 5 to change the kill password to nonzero.
If the tag's kill password is nonzero, skip step 5 and step 6 to perform the kill operation directly.

[Step 5]
Purpose: Writes the nonzero value to the Kill Password location of RESERVED memory bank of the 6C tag when the tag's kill password is zero
Command: RFID_18K6CTagWrite (ID: 0x42)
Parameters: Bank (0x00) / Offset (0x0000, 0x0001) / Data (0x1234, 0x5678) / RetryCount (0x01) / PerformSelect (0x00) / PerformPostMatch (0x00)
Returned Data: RFID_STATUS_OK
Report Packet: Four packet types are as follows,

Command-Begin packet	(see step 5c/5i)
Inventory-Response packet	(see step 5d/5j)
Tag-Access packet	(see step 5e/5k)
Command-End packet	(see step 5f/5l)

If the host doesn't need to check the data of previous write, skip step 6.

[Step 6]
Purpose: Reads the kill password value from the Kill Password location of RESERVED memory bank of the 6C tag.
Command: RFID_18K6CTagRead (ID: 0x41)
Parameters: Bank (0x00) / Offset (0x0000) / Count (0x02) / RetryCount (0x01) / PerformSelect (0x00) / PerformPostMatch (0x00)
Returned Data: RFID_STATUS_OK

Report Packet: Four packet types are as follows,
Command-Begin packet (see step 6c)
Inventory-Response packet (see step 6d)
Tag-Access packet (see step 6e)
Command-End packet (see step 6f)

[Step 7]
Purpose: Kills the specific 6C tag with a valid nonzero kill password.
Command: RFID_18K6CTagKill (ID: 0x43)
Parameters: KillPassword (0x12345678) / RetryCount (0x05) / PerformSelect (0x00) / PerformPostMatch (0x00)
Returned Data: RFID_STATUS_OK
Report Packet: Four packet types are as follows,
Command-Begin packet (see step 7c)
Inventory-Response packet (see step 7d)
Tag-Access packet (see step 7e)
Command-End packet (see step 7f)

Table 5.9 - Host Transmits and Module Replies of Use Case 7

Step	Data Flow	Command and Parameters / Response packet
1a	Host => Module	43 49 54 4D FF 12 00 2C 01 00 00 00 20 00 B7 EB
1b	Host <= Module	52 49 54 4D 00 12 00 00 00 00 00 00 00 00 FE 44
2a	Host => Module	43 49 54 4D FF 32 00 00 00 00 00 00 00 00 90 33
2b	Host <= Module	52 49 54 4D 00 32 00 00 00 00 00 00 00 00 02 E3
3a	Host => Module	43 49 54 4D FF 34 00 00 00 00 01 00 00 00 AC 1F
3b	Host <= Module	52 49 54 4D 00 34 00 00 00 00 00 00 00 00 8A B9
4a	Host => Module	43 49 54 4D FF 41 00 00 00 02 01 00 00 00 38 87
4b	Host <= Module	52 49 54 4D 00 41 00 00 00 00 00 00 00 00 9D 65
4c	Host <= Module	42 49 54 4D 01 01 01 00 00 00 02 00 00 00 10 00 00 00 9E DA 09 00 4E A7
4d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 AB DA 09 00 5E 85 86 32 00 00 00 00 30 00 35 E0 00 11 22 33 44 55 66 77 88 99 09 43 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6A C5
4e	Host <= Module	41 49 54 4D 01 01 01 00 06 00 04 00 02 00 AF DA 09 00 C2 00 00 00 00 00 00 00 **00 00 00 00** 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6F 3D
4f	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 03 00 B0 DA 09 00 00 00 00 00 EA 7C
5a	Host => Module	43 49 54 4D FF 42 00 00 00 34 12 01 00 00 5C DC
5b	Host <= Module	52 49 54 4D 00 42 00 00 00 00 00 00 00 00 D9 48
5c	Host <= Module	42 49 54 4D 01 01 01 00 00 00 02 00 00 00 11 00 00 00 3E 32 0A 00 28 56
5d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 4B 32 0A 00 68 9A 86 32 00 00 00 00 30 00 35 E0 00 11 22 33 44 55 66 77 88 99 09 43 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0E 1A
5e	Host <= Module	41 49 54 4D 01 01 01 00 06 00 03 00 02 00 56 32 0A 00 C3 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 35 D2
5f	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 03 00 56 32 0A 00 00 00 00 00 B6 C9
5g	Host => Module	43 49 54 4D FF 42 00 01 00 78 56 01 00 00 13 5A
5h	Host <= Module	52 49 54 4D 00 42 00 00 00 00 00 00 00 00 D9 48
5i	Host <= Module	42 49 54 4D 01 01 01 00 00 00 02 00 00 00 11 00 00 00 3E 32 0A 00 28 56
5j	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 4B 32 0A 00 68 9A 86 32 00 00 00 00 30 00 35 E0 00 11 22 33 44 55 66 77 88 99 09 43 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0E 1A
5k	Host <= Module	41 49 54 4D 01 01 01 00 06 00 03 00 02 00 56 32 0A 00 C3 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 35 D2

5l	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 03 00 56 32 0A 00 00 00 00 00 B6 C9
6a	Host => Module	43 49 54 4D FF 41 00 00 00 02 01 00 00 00 38 87
6b	Host <= Module	52 49 54 4D 00 41 00 00 00 00 00 00 00 00 9D 65
6c	Host <= Module	42 49 54 4D 01 01 01 00 00 00 02 00 00 00 10 00 00 00 B7 7E 0A 00 78 57
6d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 C5 7E 0A 00 58 75 86 32 00 00 00 00 30 00 35 E0 00 11 22 33 44 55 66 77 88 99 09 43 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 52 DA
6e	Host <= Module	41 49 54 4D 01 01 01 00 06 00 04 00 02 00 C9 7E 0A 00 C2 00 00 00 00 00 00 00 **12 34 56 78** 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 D4 7F
6f	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 03 00 CA 7E 0A 00 00 00 00 00 B3 97
7a	Host => Module	43 49 54 4D FF 43 78 56 34 12 05 00 00 00 BD 43
7b	Host <= Module	52 49 54 4D 00 43 00 00 00 00 00 00 00 00 FA A3
7c	Host <= Module	42 49 54 4D 01 01 01 00 00 00 02 00 00 00 13 00 00 00 64 CC 0A 00 4D 3C
7d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 72 CC 0A 00 5A 82 86 32 00 00 00 00 30 00 35 E0 00 11 22 33 44 55 66 77 88 99 09 43 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 DF 46
7e	Host <= Module	41 49 54 4D 01 01 01 00 06 00 03 00 02 00 81 CC 0A 00 C4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 18 51
7f	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 03 00 82 CC 0A 00 00 00 00 00 2E ED

5.8 Inventory with Tag-selection Criteria for a Specific Tag

[Step 1]

Purpose: Sets the operation mode to continuous mode.
Command: RFID_RadioSetOperationMode (ID: 0x02)
Parameters: Mode (0x00)
Returned Data: RFID_STATUS_OK

[Step 2]

Purpose: Sets the RF power level of the logical antenna port 0 to 30.0dBm.
Command: RFID_AntennaPortSetConfiguration (ID: 0x12)
Parameters: AntennaPort (0x00) / PowerLevel (0x012C) / DwellTime (0x0000) / NumberInventoryCycles (0x2000) / PhysicalPort (0x00)
Returned Data: RFID_STATUS_OK

[Step 3]

Purpose: Sets the tags of interest.
Command: RFID_18K6CSetQueryTagGroup (ID: 0x30)
Parameters: Selected (0x00) / Session (0x02) / Target (0x00)
Returned Data: RFID_STATUS_OK

[Step 4]

Purpose: Sets the currently-active singulation algorithm to fixed-Q singulation algorithm.
Command: RFID_18K6CSetCurrentSingulationAlgorithm (ID: 0x32)
Parameters: Algorithm (0x00)
Returned Data: RFID_STATUS_OK

[Step 5]

Purpose: Configures the settings of fixed-Q singulation algorithm.
Command: RFID_18K6CSetSingulationAlgorithmParameters (ID: 0x34)
Parameters: Algorithm (0x00) / QValue (0x03) / RetryCount (0x00) / ToggleTarget (0x00) / RepeatUntilNoTags (0x00) / RFU (0x00)
Returned Data: RFID_STATUS_OK

[Step 6]

Purpose: Enables tag-selection criteria 0.
Command: RFID_18K6CSetActiveSelectCriteria (ID: 0x20)
Parameters: CriteriaIndex (0x00) / ActiveState (0x01)
Returned Data: RFID_STATUS_OK

[Step 7]

Purpose: Configures the settings of tag-selection criteria 0.
Command: RFID_18K6CSetSelectCriteria (ID: 0x22)
Parameters: CriteriaIndex (0x00) / Bank (0x01) / Offset (0x0020) / Count (0x20) / Target (0x02) / Action (0x00) / Truncation (0x00)
Returned Data: RFID_STATUS_OK

[Step 8]

Purpose: Sets mask data of tag-selection criteria 0.
Command: RFID_18K6CSetSelectCriteria (ID: 0x24)
Parameters: CriteriaIndex (0x00) / MaskIndex (0x00) / MaskData0 (0x11) / MaskData1 (0x11) / MaskData2 (0x22) / MaskData3 (0x22)
Returned Data: RFID_STATUS_OK

[Step 9]

Purpose: Executes a tag inventory with tag-selection function for a specific tag whose front two words of EPC value must be 0x1111 2222.
Command: RFID_18K6CTagInventory (ID: 0x40)
Parameters: PerformSelect (0x01) / PerformPostMatch (0x00) / PerformGuardMode (0x00)
Returned Data: RFID_STATUS_OK
Report Packet: Three packet types are as follows,
Command-Begin packet (see step 6c)

Inventory-Response packet (see step 6d/6e/6f/6g)
Command-End packet (see step 6h)

[Step 10]
Purpose: Stops currently-executing RFID_18K6CTagInventory operation.
Command: RFID_ControlCancel (ID: 0x50)
Parameters: padding
Returned Data: RFID_STATUS_OK

Table 5.11 - Host Transmits and Module Replies of Use Case 11

Step	Data Flow	Command and Parameters / Response packet / Report packet
1a	Host => Module	43 49 54 4D FF 02 00 00 00 00 00 00 00 00 92 C7
1b	Host <= Module	52 49 54 4D 00 02 00 00 00 00 00 00 00 00 00 17
2a	Host => Module	43 49 54 4D FF 12 00 2C 01 00 00 00 20 00 B7 EB
2b	Host <= Module	52 49 54 4D 00 12 00 00 00 00 00 00 00 00 FE 44
3a	Host => Module	43 49 54 4D FF 30 00 02 00 00 00 00 00 00 14 95
3b	Host <= Module	52 49 54 4D 00 30 00 00 00 00 00 00 00 00 65 25
4a	Host => Module	43 49 54 4D FF 32 00 00 00 00 00 00 00 00 90 33
4b	Host <= Module	52 49 54 4D 00 32 00 00 00 00 00 00 00 00 02 E3
5a	Host => Module	43 49 54 4D FF 34 00 03 00 **00** 00 00 00 00 9A B1
5b	Host <= Module	52 49 54 4D 00 34 00 00 00 00 00 00 00 00 8A B9
6a	Host => Module	43 49 54 4D FF 20 00 01 00 00 00 00 00 00 68 1E
6b	Host <= Module	52 49 54 4D 00 20 00 00 00 00 00 00 00 00 9B 76
7a	Host => Module	43 49 54 4D FF 22 00 01 20 00 20 02 00 00 29 B4
7b	Host <= Module	52 49 54 4D 00 22 00 00 00 00 00 00 00 00 FC B0
8a	Host => Module	43 49 54 4D FF 24 00 00 **11 11 22 22** 00 00 49 F9
8b	Host <= Module	52 49 54 4D 00 24 00 00 00 00 00 00 00 00 74 EA
9a	Host => Module	43 49 54 4D FF 40 **01** 00 00 00 00 00 00 00 FF 19
9b	Host <= Module	52 49 54 4D 00 40 00 00 00 00 00 00 00 00 BE 8E
9c	Host <= Module	42 49 54 4D 01 01 01 01 00 00 02 00 00 00 0F 00 00 00 35 00 14 00 D7 CE
9d	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 01 00 45 00 14 00 6B 9D 86 32 DE FE 00 00 30 00 **11 11 22 22** 33 33 44 44 55 55 66 66 18 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5E A4
9e	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 02 00 D5 01 14 00 6F A6 86 32 F9 FE 00 00 30 00 11 11 22 22 33 33 44 44 55 55 66 66 18 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 DF 6C
10a	Host => Module	43 49 54 4D FF 50 00 00 00 00 00 00 00 00 D2 0D
9f	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 03 00 6D 03 14 00 71 A6 86 32 09 FF 00 00 30 00 11 11 22 22 33 33 44 44 55 55 66 66 18 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 2C B5
9g	Host <= Module	49 49 54 4D 01 01 01 00 05 00 07 00 04 00 F8 04 14 00 70 A9 86 32 FF FE 00 00 30 00 11 11 22 22 33 33 44 44 55 55 66 66 18 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4A C2
9h	Host <= Module	45 49 54 4D 01 01 01 00 01 00 02 00 05 00 F9 04 14 00 00 00 00 00 AD 87

6 APPENDIX A - Singulation Algorithm Introduction

The MTI RFID PCIe M.2 module currently consists of two main inventory algorithms.

6.1 Fixed Q (Generic) Algorithm

Features:

- Fixed Q value.
- Optionally executes rounds until no tags are read.
- Optionally retries a round “n” times.
- Optionally flips A/B flag at end of round

This algorithm executes all inventory rounds with a single Q value. In this algorithm an inventory cycle consists of one or more rounds, each of which attempts to read every slot. The number of slots to search is given by 2^Q.

For example, a Q of 7 will cause the algorithm to search 128 slots on each round.

Caution: If the time it takes to execute the round is greater than the frequency hop time (and the session is 0) or antenna dwell time, the round will never complete.

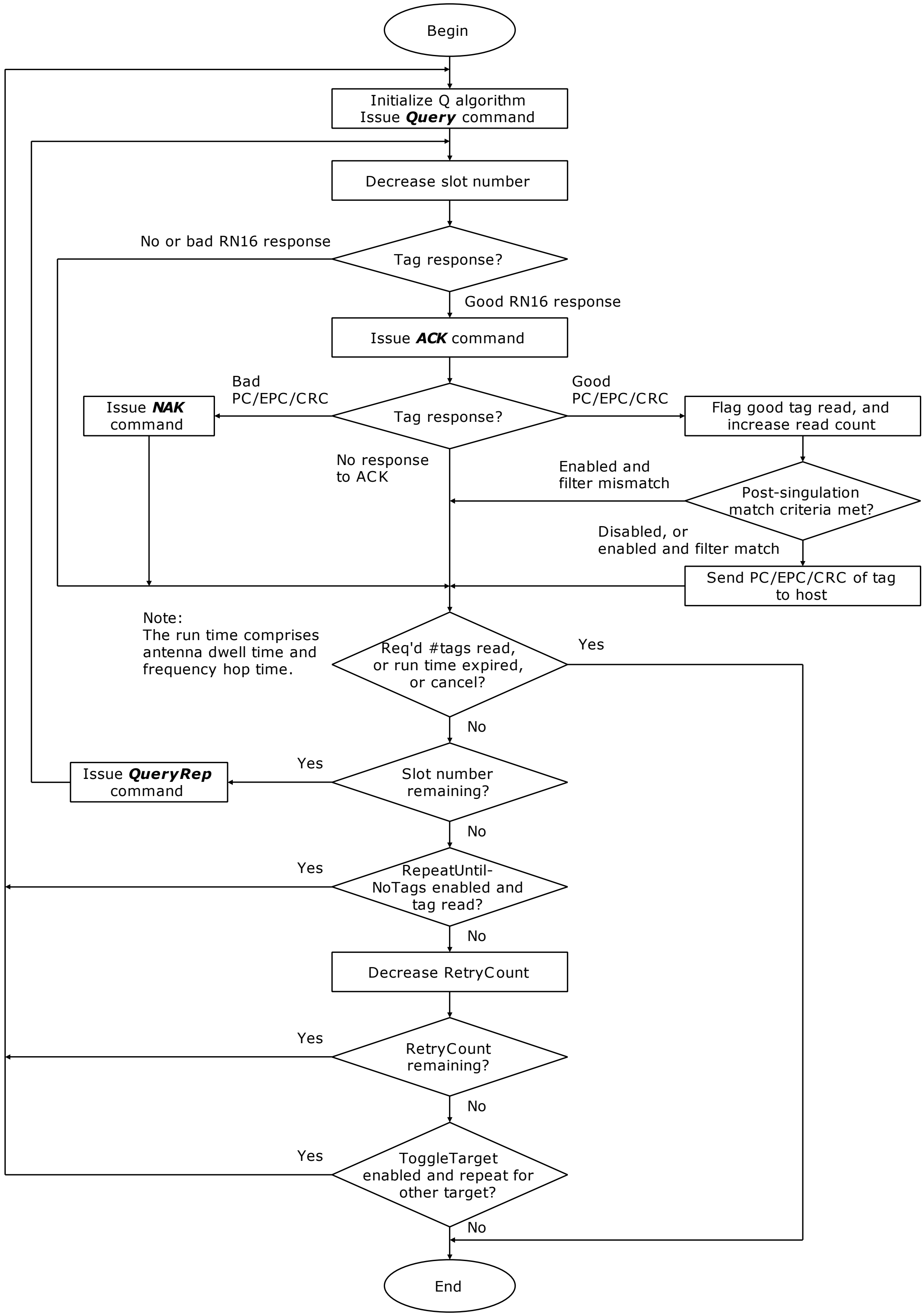

Figure 6.1 - Fixed Q (Generic) Algorithm

6.2 Dynamic Q Algorithm

Features:

- A Q adjustment algorithm.
- Uses Qstart, Qmax, and Qmin parameters to control the range of Q.
- Uses single threshold multiplier to control Q adjustment.
- Uses QueryAdjust command to modify Q value.

The value of Q is adjusted based on the continuous evaluation of the relative frequency of RN16 timeouts vs. EPC timeouts.

An inventory cycle consists of a single round initiated by a Query command.
Following the Query command, up to ((2^Q)-1) QueryRep commands are issued.

If in the course of operation the number RN16 timeouts exceeds the adjusted number of EPC timeouts by a calculated threshold, the value of Q is decremented (presumed empty slots outnumber presumed collisions). If the adjusted number of EPC timeouts exceeds the number of RN16 timeouts by a calculated threshold, the value of Q is incremented (presumed collisions outnumber presumed empty slots). While the relative number of RN16 time outs vs. the adjusted number of EPC time outs falls within the threshold, Q is unchanged.

When the value of Q changes, or if all slots under the current Q value have been inventoried, the slot counters of the participating tag population is refreshed using a QueryAdjust command.

The calculated threshold equals the current value of Q times a multiplier (set by default to 1).

Q remains unchanged while well matched to the population and changes quickly when not well matched. An inventory cycle is terminated when all slots have been checked with Q = Qmin.

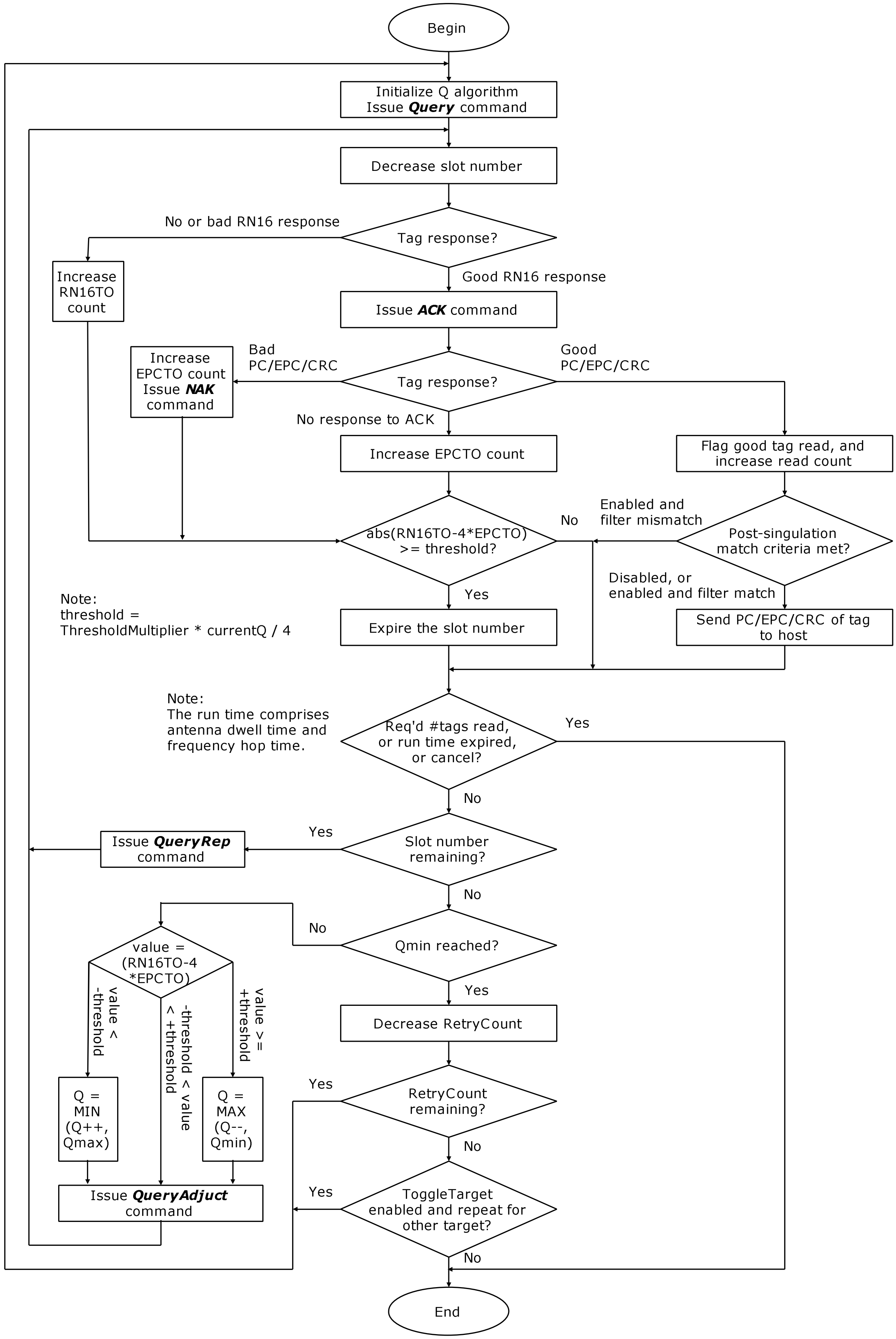

Figure 6.2 - Dynamic Q Algorithm

6.3 Fixed Q Algorithm with Access

When performing access, only the use of a fixed Q algorithm is supported. The configuration parameters and features are those of section 6.1.

Note also that the following inventory a tag may have not complete access because of regulatory channel dwell time requirements.

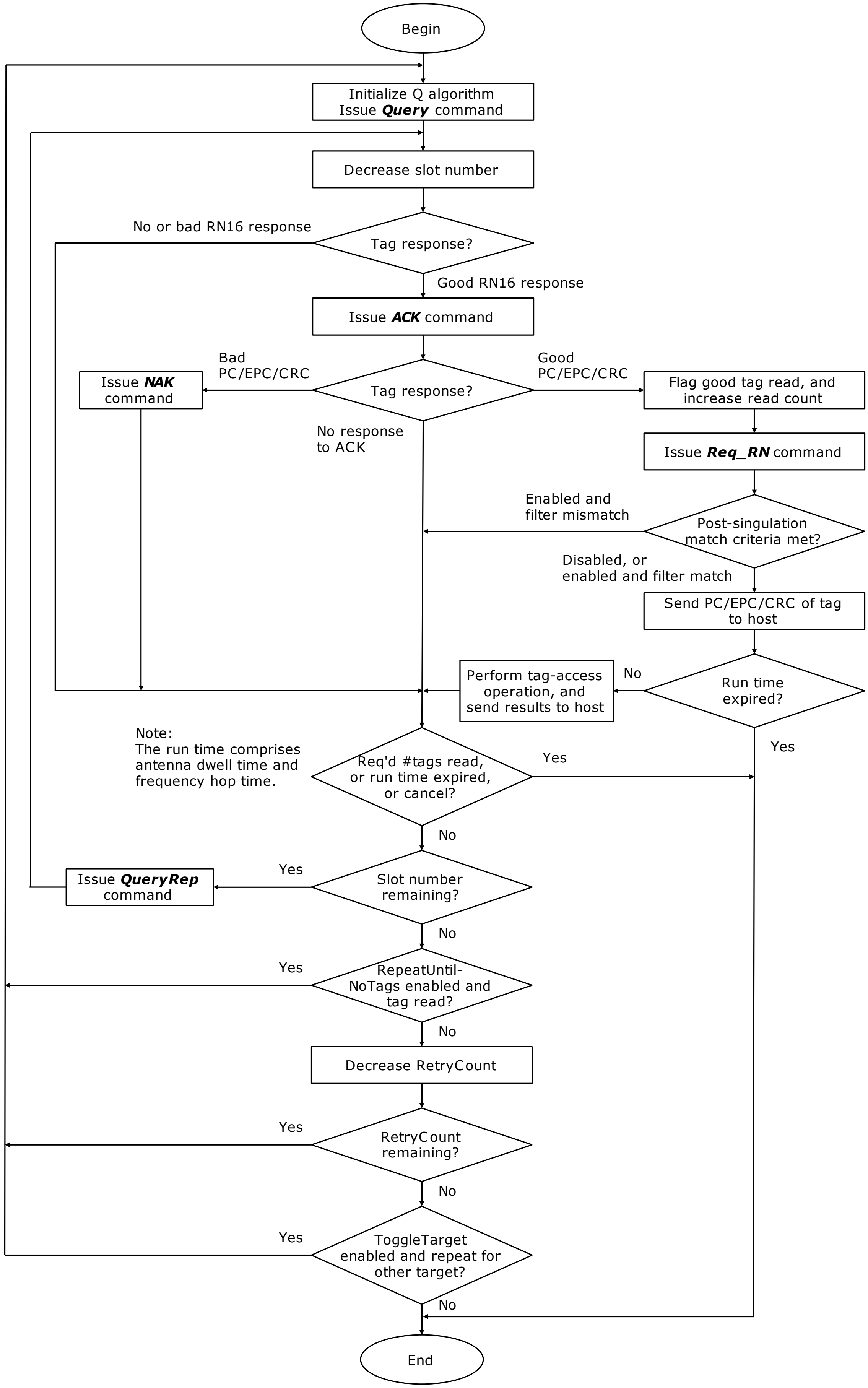

Figure 6.3 - Fixed Q Algorithm with Access

7 APPENDIX B - MTI MAC Firmware Error Codes

This section provides a description of the MTI MAC firmware error codes.

Error codes are set by the MTI MAC firmware for a variety of reasons and can be read by host via 2 methods:

- The RFID_MacGetError command
- The status field of the Command-End packet

Error codes are 32 bit values that are generally grouped together based on the subsystem that generates the error code.

Table 7.1 - Error Code Ranges/Module Table

Error Code Number Range	Subsystem Name
0x0000	Command successful with no errors.
0x0001 - 0x0100	Core State Machine
0x0101 - 0x0200	Host Interface Module
0x0201 - 0x0300	RFID Protocol Modules
0x0301 - 0x0400	RFID Transceiver Control Module
0x0401 - 0x0500	GPIO, MCU support modules, OEM Config. Module
0x0501 - 0x0600	RESERVED
0x0601 - 0x0700	MTI RFID PCIe M.2 Module low level interface module
0x0701 - 0x0800	BIST Module (Build In Self Test)

Table 7.2 - Error Code Details

Code	Sub-System	Description
Core State Machine		
0x0000	MACERR_SUCCESS	Command successful with no errors.
0x0001	CSM_ERR_UNKNOWNCMD	This error is set when an invalid command has been issued to the MAC firmware. The MAC firmware performs basic bounds checking on command values.
0x0002	CSM_ERR_PREEXECPROC	An error occurred during pre-command execution processing. This may happen if the MAC firmware is unable to transmit a Command-Begin packet to the host.
0x0003	CSM_ERR_POSTEXECPROC	An error occurred during post-command execution processing. This may happen if the MAC firmware is unable to flush the host TX buffers after the main processing of a given command is complete.
0x0004	CSM_ERR_BADENGTESTSUBCMD	This is set when an unsupported ENGTEST sub-command has been indicated via the HST_ENGTST_ARG0 register, bits 7:0. FYI - BUG - currently only set if particular engineering test sub-commands have not been compiled into the MAC firmware image. Eventually this will be reported for all invalid sub-command values in HST_ENGTEST_ARG0.
0x0005	CSM_ERR_MBPRDADDR	Set if an invalid / unsupported UHF RFID transceiver register is detected in the HST_MBP_ADDR after an MBPRDREG command is issued to the MAC firmware.
0x0006	CSM_ERR_MBPWRADDR	Set if an invalid / unsupported UHF RFID transceiver register is detected in the HST_MBP_ADDR after an MBPWRREG command is issued to the MAC firmware.
0x0007	CSM_ERR_SUBSYSINIT_CPU	Set if the CPU module fails to initialize on MAC firmware boot.
0x0008	CSM_ERR_SUBSYSINIT_DBG	Set if the Debug module fails to initialize on MAC firmware boot.

0x0009	CSM_ERR_SUBSYSINIT_CSM	Set if the Core State Machine fails to initialize on MAC firmware boot.
0x000A	CSM_ERR_SUBSYSINIT_OEMCFG	Set if the OEM configuration module fails to initialize on MAC firmware boot.
0x000B	CSM_ERR_SUBSYSINIT_HOSTIF	Set of the HOST interface module fails to initialize on MAC firmware boot.
0x000C	CSM_ERR_SUBSYSINIT_TILIF	Set if the UHF RFID transceiver low level interface module fails to initialize on MAC firmware boot.
0x000D	CSM_ERR_SUBSYSINIT_BIST	Set if the BIST module fails to initialize on MAC firmware boot.
0x000F	CSM_ERR_SUBSYSINIT_GPIO	Set if the GPIO module fails to initialize on MAC firmware boot.
0x0010	CSM_ERR_SUBSYSINIT_RFTC	Set of the RF Transceiver Control module fails to initialize on MAC firmware boot.
0x0011	CSM_ERR_SUBSYSINIT_PROT	Set if the RFID Protocol module(s) fail to initialize on MAC firmware boot.
0x0012	CSM_ERR_PROTSCHED_UNKST	Set if the RFID protocol scheduler module detects an unknown state - likely indicates firmware corruption or runtime SRAM corruption by errant code.
0x0013	CSM_ERR_PROTSCHED_AMBANT	Set if the Antenna configuration dwell time and inventory round count are both zero - which is illegal and ambiguous.
0x0014	CSM_ERR_PROTSCHED_NODESC	Set if the protocol scheduler detects that no logical antennas have been enabled using the HST_ANT_DESC_CFG register bank.
0x0015	CSM_ERR_PROTSCHED_PORTDEF	Set when a bogus physical antenna port definition value is used - this likely means that the TX and RX port values are not the same - which is required for MTI RFID Development Platform.
0x0016	CSM_ERR_PROTSCHED_NOFRQCH	Set by the protocol scheduler when no frequency channels have been enabled.
0x0017	CSM_ERR_PROTSCHED_BADREGION	Set by the protocol scheduler when a bogus regulatory region has been detected in HST_REGULATORY_REGION.
0x0018	CSM_ERR_PROTSCHED_BADFTIME	Set by the protocol schedulers FCC state machine when a bogus FCC frequency hop value has been written to HST_PROTSCH_FTIME, Bank 0 - only 100, 200, 400 milliseconds are valid values.
0x0019	CSM_ERR_PROTSCHED_FTUNETO	Not currently set by firmware.
0x001A	CSM_ERR_SUBSYSINIT_OEMHWOPTS	Set if the OEM hardware-option configuration module fails to initialize on MAC firmware boot.
0x001B	CSM_ERR_SUBSYSINIT_NVMEMUPD	Set if the firmware failed to initialize the NV Memory Update module at boot time.
0x001C	CSM_ERR_BAD_RESET_KEY	Set if the firmware CPU module's reset device logic is called with a bogus key. This will generally only happen if the system has experienced a crash and this logic is being called through an invalid call chain - likely due to some sort of corruption.
0x001D	CSM_ERR_DEV_RESET_FAILED	Set if the device reset logic fails to actually reset the device - likely due to a MCU related hardware failure or system corruption.
0x001E	CSM_ERR_NVMEMUPD_ABORT_MACERRNO	Set *prior* to entering non-volatile memory update mode if the current global MAC firmware error status is indicating an error. The MAC will not enter non-volatile memory update mode if there is currently an error. The host should use the CLRERR command to clear any errors; if this doesn't work, the device may need to be manually updated using the recovery method indicated in the MAC firmware datasheet.
0x001F	CSM_ERR_NVMEMUPD_INT_MEMBNDS	Set if an internal memory bounds check fails while in non-volatile memory update mode. If these errors occurred the MAC firmware tries very hard not to update non-volatile memory with bogus data. This error occurs likely due to a system corruption.
0x0020	CSM_ERR_NVMEMUPD_ENTRYKEY	Set if the non-volatile memory mode entry logic detects an invalid key. This would occur if the calling logic erroneously called the non-volatile memory logic due to system corruption / firmware error.
0x0021	CSM_ERR_NVMEMUPD_NVFLUSH	Set if, during non-volatile memory update mode, the firmware fails to write flash at the lowest level. This is likely due to flash lock bits being set (i.e. via tools like SAM-BA) or a system corruption.
0x0022	CSM_ERR_NVMEMUPD_WRVERFAIL	Set if write verification logic fails after writing data at the lowest

		level to flash. This may indicate problems with the MCU device flash hardware. This can occur if the MCU device flash has been updated too many times.
0x0023	CSM_ERR_INVAL_START_CHAN	Set by the protocol scheduler if the HST_RFTC_FRQCH_CMDSTART register has been set to an invalid channel.
0x0024	CSM_ERR_PROTSCHED_UNK_ALGO	Set by the protocol scheduler if an invalid protocol algorithm has been selected via the HST_INV_CFG register.
0x0025	CSM_ERR_INVAL_PWRMODE	Set by the core state machine if an invalid power management mode has been specified in the HST_PWRMGMT register.
0x0026	CSM_ERR_PWRMODE_CORRUPT	This is set if a system corruption has occurred and the logic is unable to determine the desired power management mode.
0x0027	CSM_ERR_NVMEMUPD_TXFAIL	Set if the non-volatile memory mode logic fails to transmit a packet to the host during non-volatile memory update.
0x0028	CSM_ERR_NVMEMUPD_UPD_BOUNDS	Set during non-volatile memory update if the range indicated for updates falls outside the valid non-volatile memory ranges available on the device.
0x0029	CSM_ERR_NVMEMUPD_UNKNOWN	An unknown error has occurred during non-volatile memory updates - likely a system corruption.
0x002A	CSM_ERR_NVMEMUPD_RXTO	Set during non-volatile memory mode if the firmware does not receive a packet from the host within 60 seconds. This may occur if the host has crashed or the physical interface has been removed or corrupted.
0x002B	CSM_ERR_GPIO_NOTAVAIL	This error code is generated when the host / user attempts to use a GPIO pin that has previously been configured as unavailable in the OEM configuration area entry GPIO_AVAIL.
0x002C	CSM_ERR_ANT_NOTAVAIL	This error code is generated when the host / user attempts to use an antenna pin that has previously been configured as unavailable in the OEM configuration area entry ANT_AVAIL.
0x002D	CSM_ERR_CMDNOTAVAILABLE	Set by the command processor when a command is invoked from the host, which has been defined, but is not available in the MAC firmware codebase. This situation can occur if, for instance, a command is disabled by means of a compile-time switch.
0x002E	CSM_ERR_NOCORDICDEF	Set by the protocol scheduler when no CORDIC values are found in the OEM configuration area. CORDIC values are part of the LBT configuration. See the OEM configuration section of the firmware datasheet for more details on these settings. Cordic configuration values are only required when LBT is enabled.
0x002F	CSM_ERR_SUBSYSINIT_DEBUG	Set if the firmware failed to initialize the Debug subsystem at boot time.
0x0030	CSM_ERR_SUBSYSINIT_TRACE	Set if the firmware failed to initialize the Trace subsystem at boot time.
0x0031	CSM_ERR_BUILD_TARGET_DEVICE_MISMATCH	Set if the firmware failed the Target Build and Physical Device Check at boot time.
0x0032	CSM_ERR_DIAGNOSTICS	Set if the firmware failed to properly set MAC Error diagnostic codes. Actual MAC Error may not correctly be reflected by the MAC Error register.
0x0033	CSM_ERR_SUBSYSINIT_HOSTIFREGS_INIT	Set if the MAC register default value initialization module fails to initialize on MAC firmware boot.
0x0034	CSM_ERR_SUBSYSINIT_HANDSHAKE	Set if the firmware failed to initialize the Handshake interface subsystem at boot time.
0x0035	CSM_ERR_NVMEMUPD_INVALID_MODE	Set if the HST_NV_UPDATE_CONTROL MAC register had an invalid update_mode set.
0x0036	CSM_ERR_INVALID_CMD_WHILE_IN_CRIT_ERROR	Set if a Gen2 command is attempted following a critical error during system initialization. Typically caused by a failed OEM read attempt and can usually be resolved by formatting OEM.
0x0037	CSM_ERR_CRITICAL_ERROR_UNKNOWN	Set if an unknown critical error is detected at the end of system initialization. Typically caused by a failed OEM read attempt and can usually be resolved by formatting OEM.
Host Interface Module		
0x0101	RESERVED	RESERVED
0x0102	HOSTIF_ERR_USBDESC	Set by the USB interface module when an unsupported descriptor TYPE has been requested by the host (i.e. not a device, string, configuration descriptor type. This may be due to compatibility problems with the USB host.

0x0103	HOSTIF_ERR_USBDESCIDX	Set by the USB interface module when an unsupported device descriptor index has been requested by the Host.
0x0104	HOSTIF_ERR_USBTXEP0	Set by the USB interface module when it is unable to transmit the response to a request on USB endpoint 0 (aka control endpoint). This may be due to compatibility or synchronization problems with the USB host.
0x0105	RESERVED	RESERVED
0x0106	HOSTIF_ERR_USBRXBUFFSZ	Set by the USB interface module when higher level firmware requests an unsupported buffer length. This may be due to a firmware build error or corrupted firmware in flash.
0x0107	HOSTIF_ERR_RXUNKNOWN	This is set by the Host interface module when the underlying physical interface module returns an unknown error code on receive from the host. This may be due to a firmware build issue, corrupted firmware image or corrupted SRAM due to errant MAC firmware code.
0x0108	HOSTIF_ERR_TXUNKNOWN	This is set by the Host interface module when the underlying physical interface module returns an unknown error code on transmit to the Host. This may be due to a firmware build issue, corrupted firmware image or corrupted SRAM due to errant code.
0x0109	HOSTIF_ERR_BADIFSTATE	This is set when the Host interface code detects that its internal state machine out of sync. This could be due to a corrupted firmware image or corrupted SRAM due to errant MAC firmware code.
0x010A	RESERVED	RESERVED
0x010B	HOSTIF_ERR_REGADDR	Set by the host interface module when an invalid MAC firmware register read or write is attempted (either by the host or internally by the MAC firmware).
0x010C	RESERVED	RESERVED
0x010D	HOSTIF_ERR_USBDESCINIT	This is set by the host interface module during initialization if it is unable to retrieve USB string descriptors from non-volatile memory (i.e. flash) OEM configuration area. This may be due to a corrupt or unformatted OEM configuration area. It may also be due to a firmware build issue if the OEM configuration definition is out of sync with the MAC firmware code.
0x010E	HOSTIF_ERR_SELECTORBNDS	This is set when the host attempts to *write* a value to a selector type register that is out of range for that selector.
0x010F	RESERVED	RESERVED.
0x0110	HOSTIF_ERR_PKTALIGN	Not currently set by MAC firmware.
0x0111	HOSTIF_ERR_BADRAWMODE	Set by the low level host interface logic if an upper level requests an unsupported raw mode. This may occur if the system is corrupted.
0x0112	HOSTIF_ERR_UNKLNKSTATE	Set by the low level host interface logic if a system corrupt occurs and the link manager cannot determine the current link state.
0x0113	HOSTIF_ERR_UNKUSBSETUP	Set by the low level host interface logic if an unknown / unsupported control command is received from the host. This may occur if the host logic and the MAC firmware logic are out of sync, in terms of the lowest level host interface (UART, USB).
0x0114	HOSTIF_ERR_UARTRXBUFFSZ	This is set if the upper layer host logic attempts to receive data and the lower layer cannot support the buffer size requested. This will happen if the system is corrupted.
0x0115	HOSTIF_ERR_RAWMODECTL	Set by the low level host interface logic if a control command is received from the host while in raw mode - which is not allowed. This would happen if the host caused the MAC firmware to enter non-volatile memory update mode, which uses the raw mode, and then the host proceeded to issue control commands.
0x0116	HOSTIF_ERR_UNKHOSTIF	Set by the host interface module at boot time if the OEM configuration area is specifying an unsupported host interface.
0x0117	HOSTIF_ERR_UNKREGSTD	Set by the host interface module at boot time if the OEM configuration area is specifying an unsupported regulatory standard.
0x0118	HOSTIF_ERR_DEBUGID	Set by host interface module if Debug Id is invalid.
0x0119	HOSTIF_ERR_DEBUGOVERFLOW	Set by host interface module if Debug Buffer overflows.
0x011A	HOSTIF_ERR_REGREADONLY	Set by the host interface module when a Read-Only MAC firmware register write is attempted by the host.
0x011B	HOSTIF_ERR_REGWRITEONLY	Set by the host interface module when a Write-Only MAC

		firmware register read is attempted by host.
0x011C	HOSTIF_ERR_BADREGIONINITVALUES	Set by the host interface module if the default region dependent parameters are invalid.
0x011D	HOSTIF_ERR_INVALIDENGTESTARG	Set by an ENGTEST sub-command with an invalid argument.
0x011E	HOSTIF_ERR_INVALIDSETFREQARG	Set by Set Frequency command with an invalid argument. When this error is set, the result registers will be set to 0xFFFFFFFF.
0x011F	HOSTIF_ERR_INVALID_RSSI_FILTERING	Set when an invalid Inventory RSSI Filtering configuration has been configured.
0x0120	HOSTIF_ERR_INVALID_TAGACC_CNT	Set when an invalid HST_TAGACC_CNT value is specified.
0x0121	HOSTIF_ERR_INVALID_BW_MODE	Set when an invalid BlockWrite mode is specified in HST_IMPINJ_EXTENSIONS.
0x0122	HOSTIF_ERR_OEM_MAC_REG_INIT_CTRL_ERROR	Set when an invalid MAC Register Initialization pair (Control/Data) is found during the MAC Register initialization.
0x0123	HOSTIF_ERR_OEM_MAC_REG_INIT_WRITE_ERROR	Set when an invalid MAC Register Initialization write occurs found during the MAC Register initialization.
RFID Protocol Modules		
0x0200	PROTOCOL_ERR_TRUNCATION_UNSUPPORTED	Set by protocol if truncation is set in the Select configuration register, since truncation is unsupported.
RF Transceiver Control Module		
0x0300	RFTC_ERR_BADFRQCHAN	This is set during the PLL lock logic when a bounds check fails while checking the frequency channel configuration registers.
0x0301	RFTC_ERR_BADHOPMODE	This is set if an unsupported frequency hopping mode is detected - during the PLL lock logic.
0x0302	RFTC_ERR_PLLFAILEDTOLOCK	This is set if the PLL fails to lock.
0x0303	RFTC_ERR_XCVRADC_TIMEDOUT	This is set when the RFTC module's AUX ADC function times out waiting for an ADC conversion.
0x0304	RFTC_ERR_FILTTUNE_TIMEOUT	This is set when the RFTC module times out waiting for UHF RFID transceiver to indicate RX or TX filter tuning is complete.
0x0305	RFTC_ERR_AMBIENTTEMPTOOHOT	This is set when the RFTC module detects that the ambient temperature sensor indicates too hot.
0x0306	RFTC_ERR_XCVRTEMPTOOHOT	This is set when the RFTC module detects that the transceiver temperature sensor indicates too hot.
0x0307	RFTC_ERR_PATEMPTOOHOT	This is set when the RFTC module detects that the PA temperature sensor indicates too hot.
0x0308	RFTC_ERR_PADELTATEMPTOOBIG	This is set when the RFTC module detects that the delta between the PA temperature and the ambient temperature is too great.
0x0309	RFTC_ERR_REVPWRLEVTOOHIGH	This is set when the reverse power level is too high as measured by the configured reverse power level threshold in the register set.
0x030A	RFTC_ERR_BADIFLNAGAIN	This is set when an incorrect current gain setting is passed into the IFLNA gain adjustment logic. May indicate corrupted code.
0x030B	RFTC_ERR_TXRF_BIT_FAILED	Returned by RFTC code when errors occur in transmitting a bit over the RF interface.
0x030C	RFTC_ERR_TXRF_BYTE_FAILED	Returned by RFTC code when errors occur in transmitting a buffer of bytes over the RF interface.
0x030D	RFTC_ERR_TXRF_EOT_FAILED	Returned by RFTC code when errors occur in transmitting an "end of transfer" command over the RF interface.
0x030E	RFTC_ERR_TXRF_PREAM_FAILED	Returned by RFTC code when errors occur in transmitting a "preamble" command over the RF interface.
0x030F	RFTC_ERR_TXRF_FSYNC_FAILED	Returned by RFTC code when errors occur in transmitting a "frame-sync" command over the RF interface.
0x0310	RFTC_ERR_RXRF_ISR_TIMEOUT	Indicates that the RF transceiver failed to set expected ISR bits in a timely fashion. Indicates a failure in either the RFTC state machine logic or in the RF transceiver state machine logic.
0x0311	RFTC_ERR_INVALIDLINKPARMS	This is set when invalid link parameters are detected when the filter tuning logic is run.
0x0312	RFTC_ERR_RXRF_INTERPKTTIMEOUT	This indicates a failure in either the RFTC state machine logic or in the RF transceiver state machine logic. This error can only occur if the RF transceiver starts filling its RX FIFO with received data, but fails to return the requested number of bits in a timely fashion.
0x0313	RFTC_ERR_NO_LINKPROFHDR	Not currently in use. May occur in the future when switching between link profiles if some of the required information is not

		properly coded in the MAC firmware.
0x0314	RFTC_ERR_PROFILE_INVALID	This error occurs if the RF transceiver is being loaded with an invalid profile.
0x0315	RFTC_ERR_DBMVALOUTOFRANGE	Internal error. The error is the direct result of the MAC firmware having to do a "dBm to linear" conversion on a dBm measurement that is outside the range of -99dBm through +45dBm. It the unlikely event that this error is encountered, it is probably the result of a faulty RF Peak Detector, a bug in the code that computes the dBm value from the RF Peak Detector ADC reading, or a faulty external PA circuit.
0x0316	RFTC_ERR_FWDPWRLEVTOOHIGH	If, during RF power-ramping, it is determined that the RF power at the antenna port has momentarily exceeded 35dBm, or has exceeded 33dBm steady-state, this error will be thrown. Encountering this error is often the result attempting to transmit on an open antenna port or in other cases an incorrect calibration of the gross gains. Make sure an antenna is connected on the physical port in use or see MAC firmware command 0x1B for more information on how to calibrate the system.
0x0317	RFTC_ERR_NO_GROSSPWRENTRY	Internal error that may occur if memory is corrupted.
0x0318	RFTC_ERR_TARGETPWRTOOHIGH	Indicates that the target power (in MAC firmware Virtual Register 0x706) is higher than the maximum allowed output power, which is +33dBm.
0x0319	RESERVED	RESERVED.
0x031A	RFTC_ERR_ANTENNADISCONNECTED	Indicates that the measured value of the antenna-sense resistor (reported in the MAC firmware Virtual Register 0x703) exceeds the threshold specified (specified in the MAC firmware Virtual register 0xB12). To determine which antenna was disconnected, the list of enabled antennas will need to be scanned for the one exceeding the threshold (this is done by iterating through all valid selectors in register 0x701 and examining the MAC_ANT_DESC_STAT register at address 0x703.
0x031B	RFTC_ERR_UNREC_HWOPTFORMAT	Indicates that the OEMCFG's HW_OPTIONS_FORMAT value is not recognized by the RFTC subsystem.
0x031C	RFTC_ERR_HWOPT_BADFWDPWROPT	Indicates that the forward power detection option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem.
0x031D	RFTC_ERR_HWOPT_BADREVPWROPT	Indicates that the reverse power detection option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem.
0x031E	RFTC_ERR_HWOPT_BADDRMFILTOPT	Indicates that the DRM Filter option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem.
0x031F	RFTC_ERR_HWOPT_BADAMBTEMPOPT	Indicates that ambient temperature sensor option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem.
0x0320	RFTC_ERR_HWOPT_BADPATEMPOPT	Indicates that PA temperature sensor option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem.
0x0321	RFTC_ERR_HWOPT_BADXCVRTEMPOPT	Indicates that transceiver temperature sensor option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem.
0x0322	RFTC_ERR_HWOPT_BADANTSENSOPT	Indicates that antenna-sense resistor sensor option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem.
0x0323	RFTC_ERR_BADIFLNAAGCRANGE	The range specified for the IF LNA AGC gain limits is bad. Either the "min" is higher than the "max", or the min or max setting is incorrect.
0x0324	RFTC_ERR_LPROFBADSELECTOR	When invoking the CMD_LPROF_RDXCVRREG or CMD_LPROF_WRXCVRREG commands, one of the arguments is the selector of a valid link profile. New link profile selectors cannot be created through these commands, so if a selector outside this range is passed, the RFTC_ERR_LPROFBADSELECTOR error will be generated.
0x0325	RFTC_ERR_BADXCVRADDR	One of the arguments to the CMD_LPROF_RDXCVRREG or CMD_LPROF_WRXCVRREG commands is the RF transceiver register address to configure. If the address passed is not a valid transceiver address, this error will be thrown. This error is also generated if an invalid transceiver address is detected in an OEM custom profile.

0x0326	RFTC_ERR_XCVRADDRNOTINLIST	Not all valid transceiver addresses may be configured through the link profiles. The excluded addresses include those registers which are read-only (refer to the transceiver register map) and the indirect address for the R2T command register: 0x0105.
0x0327	RFTC_ERR_BAD_RFLNA_GAIN_REQ	Set by the RFTC module if an unsupported RFLNA gain level is requested.
0x0328	RFTC_ERR_BAD_IFLNA_GAIN_REQ	Set by the RFTC module if an unsupported IFLNA gain level is requested.
0x0329	RFTC_ERR_BAD_AGCMIX_GAIN_REQ	Set by the RFTC module if an unsupported AGC/MIXER gain level is requested.
0x032A	RFTC_ERR_HWOPT_BADFWDPWRCOMPOPT	Set by the RFTC module if an unsupported compensation option is detected at OEMCFG address 0xA1.
0x032B	RFTC_ERR_INVALID_PLL_DIVIDER_VALUE	This error is generated if the PLL Divider Value is zero.
0x032C	RFTC_ERR_SJC_EXTERNALLOTOOLOW	This error is generated if the external LO signal level is below the threshold specified in register HST_RFTC_SJC_EXTERNALLOTHRSH.
0x032D	RFTC_ERR_SJC_EXTERNALLONOTSELECTED	This error is generated if SJC is enabled, and the LO source is not external.
0x032E	RFTC_ERR_BADLOSOURCE	This error is generated if the LO source is incorrectly defined in the OEM Config registers.
0x032F	RFTC_ERR_GENERALRANDOMDATA	This error is generated if there is a general error in the Random Data Transmit function.
0x0330	RFTC_ERR_XVCR_HEALTH_CHECK_FAIL	This error is generated if there is transceiver health check failure and the handler is set to enable Mac Error. See OEM Config XCVR_HEALTH_CHECK_CFG.
0x0331	RFTC_ERR_INVALID_OEM_PROFILE_HEADER	This error is generated if the OEM custom profile header is invalid.
0x0332	RFTC_ERR_AUTO_READ_RX_FIFO	This error is generated if an error during the Auto Read of the Rx FIFO Read is detected.
0x0333	RFTC_ERR_DC_OFFSET_CALIBRATION	This error is general error generated if an error occurs during the DC Offset Calibration.
0x0334	RFTC_ERR_LBT_RSSI_CALIBRATION	This error is general error generated if an error occurs during the LBT RSSI Calibration. If noise floor versus calibration value do not have a significant difference this error will occur. User should check the injected reference signal for level and frequency.
0x0335	RFTC_ERR_PA_BIAS_CAL_CONFIG	This error is related to a PA Bias Calibration Configuration error.
0x0336	RFTC_ERR_FWDPWRLEVERROR	This error is generated when the requested forward power level is not achieved during power ramp. See HST_ANT_DESC_RFPOWER for the power level requested, MAC_RFTC_PAPWRLEV for the power level achieved, and HST_RFTC_FWDPWRTHRSH for the error threshold.
0x0337	RFTC_ERR_HWOPT_BADPABIASDACCTL	Indicates that PA Bias DAC Control option found in OEMCFG's HW_OPTIONS2 field is not recognized by the RFTC subsystem.
0x0338	RFTC_ERR_PA_BIAS_CAL_MEASUREMENT	This error is related to a PA Bias Calibration measurement variation error.
0x0339	RFTC_ERR_PA_BIAS_CAL_NOT_FOUND	This error is related to a PA Bias Calibration when the target current is not found.
0x033A	RFTC_ERR_GROSSGAIN_CONFIG_INVALID	This error is generated when the Gross Gain Config Value in the OEM is invalid. Min index must be less than Max, and Max must be less than the absolute max of 32.
0x033B	RFTC_ERR_SJC_NOT_AVAILABLE_R500	This error is generated if SJC is enabled with an R500 device.
GPIO, MCU IO, NV Memory, OEM Configuration		
0x0400	IO_PERIPHERAL_PROG_ERR	This is set by the CPU module when programing IO wrong. This is likely due to errant MAC firmware code.
0x0401	IO_INVAL_RDMASK	This is set by the CPU support module when an attempt is made to read IO lines not configured for input. This may be due to internal firmware errors or the host having incorrectly configured the MTI RFID Development Platform GPIO lines.
0x0402	IO_INVAL_WRMASK	This is set by the CPU support module when an attempt is made to write IO lines not configured for output. This may be due to internal firmware errors or the host having incorrectly configured the MTI RFID Development Platform GPIO lines.
0x0403	IO_INVAL_PTR_RAM	This is set by the CPU module when a bounds check fails when

		accessing non-volatile memory - the caller has passed an incorrect RAM address. This is likely due to errant MAC firmware code.
0x0404	IO_INVAL_PTR_NV	This is set by the CPU module when a bounds check fails when attempting to read or write to non-volatile memory. This is likely due to errant MAC firmware code.
0x0405	IO_INVAL_PTR_NV_ALIGN	This is set by the CPU module when a bounds check fails when attempting to read or write to non-volatile memory. This is likely due to errant MAC firmware code.
0x0406	IO_NV_LOCK_ERR	This is set by the CPU module while attempting to write to non-volatile memory (i.e. flash). This is a flash lock error and may be due to corrupted image or misconfigured firmware or hardware problems. If this error is detected by the host, it may which to attempt to read the devices OEM configuration area and save it on the host in order to preserve device specific settings.
0x0407	IO_NV_PROG_ERR	This is set by the CPU module while attempting to write to non-volatile memory (i.e. flash). This is a low-level flash write error and may be due to a misconfigured firmware image, timing problems stemming from board hardware failures, or because the flash has exceeded is limitations for writes. If this error is detected by the host, it may which to attempt to read the devices OEM configuration area and save it on the host in order to preserve device specific settings.
0x0408	IO_OEMCFG_ADDR_BOUNDS	This is set by the OEM Configuration module when an OEM configuration Address bounds check fails when accessing the OEM configuration space. This may be due to errant MAC firmware code or errant Host code.
0x0409	IO_OEMCFG_NV_BOUNDS	This is set by the OEM Configuration module when a non-volatile memory bounds check fails when accessing the OEM configuration space. This may be due to errant MAC firmware code or errant Host code.
0x040A	IO_OEMCFG_FMT_KEY	This is set by the OEM Configuration module's format facility used as the code calling it fails to pass in the correct "format key" argument. This is a failsafe to prevent errant code from inadvertently reformatting flash - due to an invalid branch instruction, etc. This will occur when errant code jumps to the format facility incorrectly.
0x040B	IO_OEMCFG_FLUSH	This is set by the OEM Configuration module when it fails to flush in memory buffers to non-volatile memory. This may be due to a misconfigured firmware image, timing problems stemming from board hardware failures, or because the flash has exceeded its limitations for writes. If this error is detected by the host, it may switch to attempt to read the device’s OEM configuration area and save it on the host in order to preserve device specific settings.
0x040C	IO_OEMCFG_FORMAT	This is set by the OEM Configuration module when it fails to detect the correct low level file system headers for the OEM configuration area. This means that the OEM configuration area has not been formatted - due to a misconfigured board or that the OEM Configuration area has become corrupt and should not be trusted without attempting recovery or reconfiguration.
0x040D	IO_INVAL_IORSVD	This is set by the CPU module when an attempt is made to configure reserved IO pins. This is likely due to a misconfigured firmware build or errant MAC firmware code.
0x040E	IO_OEMCFG_STRING_TYPE	This is set by the OEM Configuration module when an invalid string type is selected.
0x040F	IO_OEMCFG_STRING_LENGTH	This is set by the OEM Configuration module when an invalid string length is entered.
0x0410	IO_OEMCFG_STRING_CHARACTER	This is set by the OEM Configuration module when an invalid character is entered.
0x0411	IO_OEMCFG_STRING_CURRENT_INVALID	This is set by the OEM Configuration module when a string read cannot be read correctly since the current string has an invalid header.
0x0412	IO_OEMCFG_FORMAT_KEY_INVALID	This is set by the OEM Configuration module when the generated key does not match the check key when attempting to format the OEM Configuration space.
0x0413	IO_OEMCFG_FORMAT_CONFIGURATION	This is set by the OEM Configuration module when an invalid

	_INVALID	format configuration is specified.
0x0414	IO_INVAL_NV_SECTOR	This is set by the CPU module while attempting to lock or unlock a flash sector and the specified sector is invalid.
Low Level MTI RFID PCIe M.2 Module Interface		
0x0601	TILDENIF_ERR_ADDRMISMAT	This is set by the UHF RFID transceiver interface module when ar UHF RFID transceiver register read, when configured for Serial pc mode, returns the incorrect register address in the serial respons frame. This could be due to board or UHF RFID transceiver hardwa problems or errant MAC firmware code.
0x0602	TILDENIF_ERR_RDFAILSAFE	This is set by the UHF RFID transceiver interface module when failsafe logic is activated due to no response from the UHF RFID transceiver. This happens on UHF RFID transceiver register reads. This could be due to board or UHF RFID transceiver hardware problems.
0x0603	TILDENIF_ERR_INVALPWRST	Set by the low level interface logic if, during power management, an invalid power state is requested. This will likely only occur if the system is corrupt.
0x0604	TILDENIF_ERR_INVALID_SETTING_R500	Set by the low level interface logic if, during a write, an invalid setting is selected.
Built-In Self Test		
0x0701	BIST_ERR_RF_IO_REG_CHK	This error code is set during firmware boot when the Built-In Self Test code is executed. This error indicates that certain register power up defaults on UHF RFID transceiver were not detected - possibly indicating a hardware problem.
0x0702	BIST_ERR_RF_REG_BITS	This error code is set during firmware boot when the Built In Self Test code is executed. This error indicates that a walking 1's or walking 0's bus test failed - possibly indicating a hardware problem.

8 APPENDIX C - Calculation of CRC-16

8.1 CRC-16 Encoder/Decoder

An exemplary schematic diagram for a CRC-16 encoder/decoder is shown in Figure 8.1, using the polynomial and preset defined in Table 8.1.

To encode a CRC-16, first preload the entire CRC register (i.e. C[15:0]) with 0xFFFF, then clock the data bits to be encoded into the input labeled DATA, MSB first. After clocking in all the data bits, C[15:0] holds the ones-complement of the CRC-16 value. Finally, the CRC-16 value should be inverted, and attach the inverted CRC-16 to the end of the packet.

To decode a CRC-16, first preload the entire CRC register (C[15:0]) with 0xFFFF, then clock the received data and CRC-16 {data, CRC-16} bits into the input labeled DATA, MSB first. The CRC-16 check passes if C[15:0] = 0x1D0F.

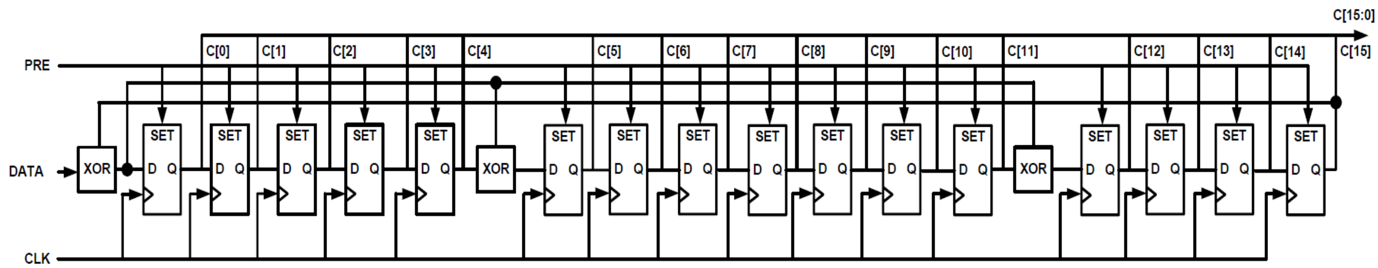

Figure 8.1 - CRC-16 Circuit

Table 8.1 - CRC-16 Precursor

CRC Type	Length	Polynomial	Preset	Residue
ISO/IEC 13239	16 bits	$X^{16} + X^{12} + X^5 + 1$	0xFFFF	0x1D0F

8.2 Example C Code to Generate the CRC-16 Value

```
/* CRC-16 */

#define POLY 0x1021

unsigned short crc16(unsigned char *buf, unsigned
short bit_length)
{
  unsigned short shift, data, val;
  int i;

  shift = 0xFFFF;

  for(i = 0; i < bit_length; i++)
  {
    if((i % 8) == 0)
      data = (*buf++) << 8;

    val = shift ^ data;
    shift = shift << 1;
    data = data << 1;

    if(val & 0x8000)
      shift = shift ^ POLY;
  }

  return shift;
}
```

```
void main(void)
{
  unsigned char packet[16];
  unsigned short crc, verification;

  /* Invert the resulting CRC value. */
  crc = ~crc16(packet, 14*8);

  packet[14] = crc >> 8;
  packet[15] = crc & 0xFF;

  verification = crc16(packet, 16*8);
  if(verification == 0x1D0F)
    printf("The CRC-16 checksum is correct.");
  else
    printf("The CRC-16 checksum is invalid.");
}
```

8.3 Examples for Calculated Result of CRC-16

[Example 1]

0xC1AA55	→ Calculate CRC-16 and invert	= **0xDA41**
0xC1AA55 + **0xDA41**	→ Calculate CRC-16 for checking	= 0x1D0F

[Example 2]

0x3000555555555555555555555555	→ Calculate CRC-16 and invert	= **0xBCAD**
0x3000555555555555555555555555 + **0xBCAD**	→ Calculate CRC-16 for checking	= 0x1D0F

[Example 3]

0x3000AAAAAAAAAAAAAAAAAAAAAAAA	→ Calculate CRC-16 and invert	= **0x7F8C**
0x3000AAAAAAAAAAAAAAAAAAAAAAAA + **0x7F8C**	→ Calculate CRC-16 for checking	= 0x1D0F

[Example 4]

0x3000A02A051012A000832A011102	→ Calculate CRC-16 and invert	= **0x33AF**
0x3000A02A051012A000832A011102 + **0x33AF**	→ Calculate CRC-16 for checking	= 0x1D0F

9 APPENDIX D - 6C Tag Memory Map

9.1 ISO 18000-6C Tag Memory Map

Tag memory shall be logically separated into four distinct banks, each of which may comprise zero or more memory words. A logical memory map of the 6C tag is shown in Figure 9.1.

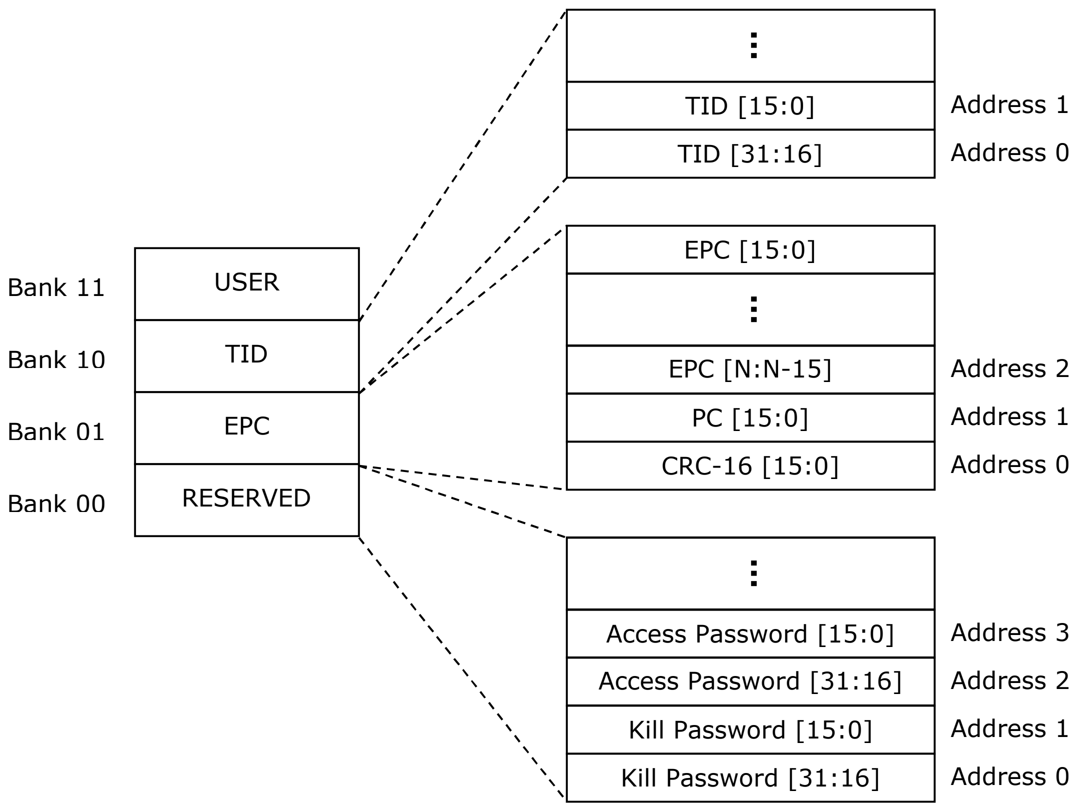

Figure 9.1 - Logical Memory Map of 6C Tag

10 APPENDIX E - Frequency Channel Tables

10.1 United States/Canada/Mexico Region Frequency Channel Table

The frequency range of those regions, which are United States ,Canada and Mexico regions, is from 902 to 928 MHz. A table of all 50 channels is shown in Table 10.1.

Table 10.1 - Frequency Channel Table of US/CA/MX Band

Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	902.75	2	903.25	3	903.75	4	904.25	5	904.75
6	905.25	7	905.75	8	906.25	9	906.75	10	907.25
11	907.75	12	908.25	13	908.75	14	909.25	15	909.75
16	910.25	17	910.75	18	911.25	19	911.75	20	912.25
21	912.75	22	913.25	23	913.75	24	914.25	25	914.75
26	915.25	27	915.75	28	916.25	29	916.75	30	917.25
31	917.75	32	918.25	33	918.75	34	919.25	35	919.75
36	920.25	37	920.75	38	921.25	39	921.75	40	922.25
41	922.75	42	923.25	43	923.75	44	924.25	45	924.75
46	925.25	47	925.75	48	926.25	49	926.75	50	927.25

10.2 Europe Region Frequency Channel Table (ETSI EN 302 208)

The frequency range of Europe region is from 865.6 to 867.6 MHz. A table of all 4 channels is shown in Table 10.2.

Table 10.2 - Frequency Channel Table of EU Band

Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	865.7	2	866.3	3	866.9	4	867.5

10.3 China Region Frequency Channel Table

The frequency range of China region is from 920.5 to 924.5 MHz. A table of all 16 channels is shown in Table 10.3.

Table 10.3 - Frequency Channel Table of CN Band

Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	920.625	2	920.875	3	921.125	4	921.375	5	921.625
6	921.875	7	922.125	8	922.375	9	922.625	10	922.875
11	923.125	12	923.375	13	923.625	14	923.875	15	924.125
16	924.375								

10.4 Japan Region Frequency Channel Table

The frequency range of Japan region is from 916.7 to 920.9 MHz. A table of all 4 channels is shown in Table 10.4.

Table 10.4 - Frequency Channel Table of JP Band

Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	916.8	2	918	3	919.2	4	920.4

10.5 Europe2 Region Frequency Channel Table (ETSI EN 300 220)

The frequency of Europe2 region is only 869.85 MHz. A table of 1 channel is shown in Table 10.5.

Table 10.5 - Frequency Channel Table of EU2 Band

Channel	Frequency (MHz)
1	869.85

10.6 Taiwan Region Frequency Channel Table

The frequency range of Taiwan region is from 922 to 928 MHz. A table of all 12 channels is shown in Table 10.6.

Table 10.6 - Frequency Channel Table of TW Band

Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	922.25	2	922.75	3	923.25	4	923.75	5	924.25
6	924.75	7	925.25	8	925.75	9	926.25	10	926.75
11	927.25	12	927.75						

10.7 South Korea Region Frequency Channel Table

The frequency range of South Korea is from 917 to 920.8 MHz. A table of all 6 channels is shown in Table 10.7.

Table 10.7 - Frequency Channel Table of KR Band

Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	917.3	2	917.9	3	918.5	4	919.1	5	919.7
6	920.3								

10.8 Australia/New Zealand Region Frequency Channel Table

The frequency range of both Australia and New Zealand regions is from 920 to 926 MHz. A table of all 7 channels is shown in Table 10.8.

Table 10.8 - Frequency Channel Table of AU/NZ Band

Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	922.25	2	922.75	3	923.25	4	923.75	5	924.25
6	924.75	7	925.25						

10.9 Brazil Region Frequency Channel Table

The frequency range of Brazil region is from 902 to 907.5 MHz and from 915 to 928 MHz. A table of all 35 channels is shown in Table 10.9.

Table 10.9 - Frequency Channel Table of BR Band

Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	902.75	2	903.25	3	903.75	4	904.25	5	904.75
6	905.25	7	905.75	8	906.25	9	906.75	10	907.25
11	915.25	12	915.75	13	916.25	14	916.75	15	917.25
16	917.75	17	918.25	18	918.75	19	919.25	20	919.75
21	920.25	22	920.75	23	921.25	24	921.75	25	922.25
26	922.75	27	923.25	28	923.75	29	924.25	30	924.75
31	925.25	32	925.75	33	926.25	34	926.75	35	927.25

10.10 Israel Region Frequency Channel Table

The frequency range of Israel region is from 915 to 917 MHz. A table of all 2 channels is shown in Table 10.10.

Table 10.10 - Frequency Channel Table of IL Band

Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	915.75	2	916.25

10.11 India Region Frequency Channel Table

The frequency range of India region is from 865 to 867 MHz. A table of all 2 channels is shown in Table 10.11.

Table 10.11 - Frequency Channel Table of IN Band

Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	865.7	2	866.3