MPR Series PC Card™
Application Programmer's Interface (API)

Release 2.0

WJ Communications, Inc.
401 River Oaks Pkwy
San Jose, CA 94089
Tel: (800) WJ1-4401
Fax: (408) 577-6621
http://www.wj.com

Change Log

Version	Date	By	Changes
2.0	05/11/05	DJK	- Added Appendix A. Opcodes. - Updated for App 2.0 firmware: o Added Class 1 Progrm ID command o Added GPIO command o Added Class 0 Zuma Program ID command
1.04	04/28/05	DJK	- Fixed order of Class 1 Kill parameters: moved kill password before tag ID - Fixed Class 1 Lock Code (0xA5).
1.03	04/01/05	DJK	- Updated for App1.3 firmware: o Added Class 0+ Programming commands o Added Enter Bootloader command o Added Class 1 Read Single Tag o Added Class 1 Inventory with Anti-collision
1.02	03/08/05	DJK	- Changed title to API
1.01	01/21/05	DJK	- Added MPR7000 power levels - Changed “Class 0+ (Impinj/ZUMA)” to “Zuma”
1.0	12/17/04	DJK	- 1st public release
*	*	*	- Pre-release versions

Table of Contents

Section 1 Introduction

1.1 Scope

The scope of this section is to specify the MPR-series RFID reader's PCMCIA host interface protocol.

1.2 Physical interface

The physical interface is a standard 68-pin PC-Card™ bus. See the document "The PC Card™ Standard", published by the Personal Computer Memory Card International Association (PCMCIA) and the Japan Electronics and Information Technologies Industry Association (JEITA) for further information.

More information about PC Card™ technology and the PCMCIA can be found at their website, http://www.pcmcia.org/.

Section 2 MPR Host-Reader Interface

2.1 UART-over-PC Card™ Bus

The host communicates with the reader over the UART-on-PC Card™ bus. When inserted in the PC Card™ slot, the host recognizes the Reader as a MPR5000, and associates it with an available COM port. The host can then communicate with the reader as if it was attached via a traditional RS232 serial port.

2.1.1 COM Port Settings

For normal command communication with the MPR5000, the standard serial COM port is configured as in Table 1.

Table 1 COM port settings

Baud Rate	Parity	Stop Bits	Handshaking	Data Bits
57600	None	1	None	8

2.2 Messaging Protocol

2.2.1 Protocol Description

- Host – Reader communications follows a Request-Response protocol.
- The Host sends request packets, and the Reader only ever sends Responses.
- The Reader never sends unsolicited traffic.
- Every properly received request is acknowledged with at least one response packet.
- A response may consist of more than one packet.
- Bytes received before a proper SOF (Start of Frame) byte and packets with CRC errors are ignored.
- There is no explicit message termination.
- The Length byte must be used to determine the location of CRC, and therefore, the end of packet.
- Maximum data section length is 64 bytes for host Request Packets.
- Maximum data section length is 256 bytes for reader Response Packets.

2.3 Packet Formats

2.3.1 Request Packet Format

Table 2 Request Packet Format

SOF	Node Address	Length	Command	Data 0	…	Data N	CRC MSB	CRC LSB

Table 3 Request Packet Format Details

Field Name	Size (bytes)	Value	Purpose
SOF	1	0x01	Start of Frame (Packet) delimiter
Node Address	1	0x00	Ignored
Length	1	Packet length excluding SOF	
Command	1	(See command details)	The command to be processed
Data	0-64	(See command details)	Specifies the parameters and data for a command
CRC	2	Bitwise inversion of 16bit CCITT-CRC of packet excluding SOF, MSB first (see Section 2.3.3)	Allows validation of correct reception of the request packet

2.3.2 Response Packet Format

Table 4 Response Packet Format

SOF	Node Address	Length	Status	Data 0	…	Data N	CRC MSB	CRC LSB

Table 5 Response Packet Format Details

Field Name	Size (bytes)	Value	Purpose
SOF	1	0x01	Start of Frame (Packet) delimiter
Node Address	1	0x00	Ignored
Length	1	Packet length excluding SOF	
Status	1	(See Appendix A.)	The status of the last requested command
Data	0-256	(See command details)	The results for the command that was just processed
CRC	2	Bitwise inversion of 16bit CCITT-CRC of packet excluding SOF, MSB first (see Section 2.3.3)	Allows validation of correct reception of the response packet

2.3.3 CRC Calculation

A 16bit CCITT CRC is used for error detection and placed at the end of the frame. The calculation uses all bytes of the frame excluding the leading SOF. The CCITT CRC polynomial is $x^{16} + x^{12} + x^5 + 1$, and the preload value is 0xFFFF. The CRC is appended to the frame after the command data, MSB first. The following code snippet and test vectors can be used as a guide to implement the CRC. The bitwise inversion (CRC XOR 0xFF) of the CRC is included in a transmitted frame. On receipt, the CRC is computed on the bytes between SOF and CRC. For valid frames, this will agree with the transmitted CRC value.

Some Test Vectors:

"ABCDEFG" returns 0xB82F

"WJCI RFID" returns 0x9ACF

An array of 256 capital 'N' characters returns 0xE45C

```
/*
This calculation uses a table lookup to generate CCITT CRC values.
The CCITT polynomial is: x^16 + x^12 + x^5 + 1
Forward direction table - i.e. msbit first
*/


static unsigned int crctab[256] = {
0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7, 0x8108, 0x9129,
0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF, 0x1231, 0x0210, 0x3273, 0x2252,
0x52B5, 0x4294, 0x72F7, 0x62D6, 0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C,
0xF3FF, 0xE3DE, 0x2462, 0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485,
0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE, 0xF5CF, 0xC5AC, 0xD58D, 0x3653, 0x2672,
0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695, 0x46B4, 0xB75B, 0xA77A, 0x9719, 0x8738,
0xF7DF, 0xE7FE, 0xD79D, 0xC7BC, 0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861,
0x2802, 0x3823, 0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,
0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12, 0xDBFD, 0xCBDC,
0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A, 0x6CA6, 0x7C87, 0x4CE4, 0x5CC5,
0x2C22, 0x3C03, 0x0C60, 0x1C41, 0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD, 0xAD2A, 0xBD0B,
0x8D68, 0x9D49, 0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13, 0x2E32, 0x1E51, 0x0E70,
0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59, 0x8F78, 0x9188, 0x81A9,
0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F, 0x1080, 0x00A1, 0x30C2, 0x20E3,
0x5004, 0x4025, 0x7046, 0x6067, 0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C,
0xE37F, 0xF35E, 0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,
0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D, 0x34E2, 0x24C3,
0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405, 0xA7DB, 0xB7FA, 0x8799, 0x97B8,
0xE75F, 0xF77E, 0xC71D, 0xD73C, 0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676,
0x4615, 0x5634, 0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB,
0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3, 0xCB7D, 0xDB5C,
0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A, 0x4A75, 0x5A54, 0x6A37, 0x7A16,
0x0AF1, 0x1AD0, 0x2AB3, 0x3A92, 0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B,
0x9DE8, 0x8DC9, 0x7C26, 0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,
0xEF1F, 0xFF3E, 0xCF5D, 0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8, 0x6E17, 0x7E36,
0x4E55, 0x5E74, 0x2E93, 0x3EB2, 0x0ED1, 0x1EF0
};
```

```
unsigned short CalculateBlockCRC16(byte count, byte *pBuf)
{
      unsigned short crc = 0xFFFF;

      while (count--)
            crc = (unsigned short)((crc << 8) ^ crctab[(crc >> 8) ^ *pBuf++]);
      return (unsigned short)(~crc);
}
```

2.4 Command Set

2.4.1 Reader Commands

2.4.1.1 Reader Information Get (01_h)

Reads basic information from the reader.

Returned Information:
Device Model Number
Device Serial Number
Hardware Version Number
Date of Manufacture
Manufacturer Name
Firmware Version Number
Bootloader Version Number

For each entry that is a string, unused bytes are programmed as 0x00.
Firmware and Bootloader Version Numbers are encoded as 2 1-byte decimal numbers. The first byte is the Major release number, second is Minor release number.

Request Packet

Opcode	0x01

Command Data
N/A

Response Packet

Status	0x00	Complete
Device Model Number	8 bytes	String
Device Serial Number	12 bytes	String
Hardware Version Number	8 bytes	String
Manufacture Date	8 bytes	ASCII, Formatted as: yyyymmdd
Manufacturer Name	16 bytes	String
Firmware Version Number	2 bytes	Major.Minor (1 decimal byte each)
Bootloader Version Number	2 bytes	Major.Minor (1 decimal byte each)

2.4.1.2 Enter Bootloader (05_h)

The firmware in the MPR is stored in FLASH memory. Therefore, it can be upgraded in the field by a program running on a host. To perform a download, the host communicates with a special firmware application called the bootloader that resides in separate region of FLASH in the card. This command exits the currently running application and enters this bootloader. The bootloader itself, and the procedure to download new firmware are described in an Application Note, "Upgrading MPR Firmware".

Request Packet

Opcode	0x05

Command Data
N/A

Response Packet

Status	0x00	Complete

2.4.1.3 GPIO (45_h)

The Revision 3 MPR hardware features two GPIO lines that are accessible from the 68-pin PC Card connector when the card is operating in Serial Communications Mode. These GPIO lines can be configured as inputs or outputs. The current value at the pin is returned in either direction as a response to this command.

The Mask parameter specifies which GPIOs to target with the data direction and set value parameters: 0x01 for GPIO 1, 0x02 for GPIO 2.

Request Packet

Opcode	0x45

Command Data

Mask	1 byte	Bit 0: GPIO 1 Bit 1: GPIO 2 1 => use this GPIO with this command; 0 => leave it alone
Data Direction	1 byte	Bit 0: GPIO 1 Bit 1: GPIO 2 1 => GPIO as output; 0 => GPIO as input
Set Value	1 byte	Bit 0: GPIO 1 Bit 1: GPIO 2 1 => GPIO High level (3V); 0 => GPIO Low level (0V)

Response Packet

Status	0x00	Complete
GPIO Read Value	1 byte	Bit 0: GPIO 1 Bit 1: GPIO 2 1 => GPIO Level at pin is HIGH; 0 => GPIO Level at pin is LOW

2.4.2 Class 0 Tag Commands

2.4.2.1 Class 0 Inventory (11_h)

Returns a list of all EPCglobal RFID Class0 tags found in the field of the reader.

Request Packet

Opcode	0x11

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Singulation (Negotiation Page) Field	1 byte	0x00 (ID0) 0x01 (ID1) 0x02 (ID2)
Filter Bit Count	1 byte	0-64 or 0-96
Filter Bits	0-12 bytes	Tag ID filter bits are left justified in bytes, MSByte sent first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.
Note: Filter bits must be left justified within the field. The least significant bits of the LSB may be left unfilled (as 0s).

The response is composed of one or more packets. The non-final packets will have a Status Byte of 0x01 (In Progress) and will contain Tag IDs. The final packet will contain an inventory summary, and a Status Byte of 0x00.

The reader attempts to read tags until a fixed, maximum number (7) of tag errors occur. There are two types of errors: Under-run errors occur when not enough bits replied from the tag. CRC errors occur when a CRC calculation on the received bits does not match the CRC reported by the tag. When the maximum number of errors occurs, the reader stops attempting reads and completes the command. Therefore, the total number of errors (CRC plus under-run) indicated in the Complete Response packet, should always equal the fixed maximum number of errors (7).

Non-final Response Packet

Status	0x01 (In Progress)
Number of Tag IDs in Packet	1 byte
Tag ID	8 or 12 bytes
…	…
Tag ID	8 or 12 bytes

Tag IDs are returned MSByte first. This first byte of the tag's ID is the EPC tag header. It must be used to determine the tag length. If the two highest order bits are 0b00, the tag ID is an EPC-96, which has 12 ID bytes. All other values indicate an EPC-64 tag with 8 ID bytes.

Tag Type	Tag ID Length (bytes)	Header 2 MS bits
EPC-64	8	01, 11, 10
EPC-96	12	00

The final response packet contains an inventory summary.

Final Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

Example

Using Antenna B, an RF Power of 0xC0, and singulating with ID1, read all tags that match 38 (0x26) Filter bits having a value of 0xC80507A000.

Entire Request Packet:

SOF	Node	Len	Command	Ant	RF Power	Sing Field	Filter Bits	Filter	CRC
01	00	0E	11	01	C0	01	26	C8 05 07 A0 00	26 6B

Response for two matching EPC tags:

Response Packet (first packet)

SOF	Node	Len	Status	Num Tag IDs	EPC Tag ID	EPC Tag ID	CRC
01	00	16	01	02	C8 05 07 A0 00 81 09 30	C8 05 07 A0 00 81 09 2E	39 0B

Response Packet (final packet)

SOF	Node	Len	Status	Total Tags Reported	Under-run errors	CRC errors	CRC
01	00	0B	00	00 02	00 0D	00 00	37 F0

2.4.2.2 Class 0 Kill Tag (12_h)

Attempts to kill one Class0 Tag. Once killed, the tag will no longer respond to commands.

Request Packet

Opcode	0x12

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Singulation (Negotiation Page) Field	1 byte	0x00 (ID0) 0x01 (ID1) 0x02 (ID2)
Kill Passcode	3 bytes	As required to kill the tag
Tag ID bits	8 or 12 bytes	Tag ID bits, MSbit first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response Packet

Status	0x00	Complete
Count	0x00	Tag not found
	0x01	Tag found and processed

2.4.2.3 Class 0+ Erase Page Command ($1B_h$)

Class 0+ tag memory is organized into pages. The Class 0 memory space is described in more detail in the Application Note "RFID Tag Memories". In current tags, only ID0, ID2 and ID3 can be erased. It is impossible to determine if a tag has been erased since the last time it was programmed, so it is highly recommended to erase tags before programming them unless erased tags are maintained as a secure item.

Erasing a page does not clear the data bits, thus an erased tag may read back the same data that was previously stored in the memory. To secure against reading old bits from an erased tag, it might be useful to erase the memory, re-program it with all 0s, and then erase it again. After erasure memory contents may appear random. Erased tags still participate in singulations, so if they are in the field of a reader performing an inventory, the inventory performance could be negatively impacted. *Care must always be taken with erased tags.*

This command singulates all Class 0 tags in the MPR field that match the filter bits and erases the requested page. Page erase time is varies with page length. ID0 is significantly shorter than ID2 and ID3 and will take less time to erase.

Request Packet

Opcode	0x1B

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Singulation (Negotiation Page) Field	1 byte	0x00 (ID0) 0x01 (ID1) 0x02 (ID2)
Page	1 byte	0x00 Erase Page ID0 0x02 Erase Page ID2 0x03 Erase Page ID3
Filter Bit Count	1 byte	Number of filter bits with which to match tag ID (0-64 or 0-96)
Filter Bits	0-12 bytes	Tag ID filter bits are left justified in bytes, MSByte is sent first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response Packet

Status	0x00	Complete
Count	0x00	Tag not found
	0x01	Tag found and processed

2.4.2.4 Class 0 Read Page Command ($1C_h$)

Class 0 tag memory is organized into pages. Class 0 tags currently in production have two of these pages available to be read. These two pages are Page ID2 and ID3. The Class 0 memory space is described in more detail in the Application Note "RFID Tag Memories". This command singulates all Class 0 tags in the MPR field that match the filter bits and reads a number of bits from the requested page.

Request Packet

Opcode	0x1C

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Singulation (Negotiation Page) Field	1 byte	0x00 (ID0) 0x01 (ID1) 0x02 (ID2)
Page	1 byte	0x02 (Page ID2) 0x03 (Page ID3)
Read Bit Count	1 byte	0-120 The number of bits to read from the requested row
Filter Bit Count	1 byte	Number of filter bits with which to match tag ID (0-64 or 0-96)
Filter Bits	0-12 bytes	Tag ID filter bits are left justified in bytes, MSByte is sent first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

The response is composed of one or more packets. The non-final packets will have a Status Byte of 0x01 (In Progress) and will contain a tag ID and data for each singulated Tag. The final packet will contain an inventory summary and a Status Byte of 0x00 (Complete).

Non-final Response Packet

Status	0x01 (In Progress)
Number of Tag IDs in Packet	1 byte
Tag ID	8 or 12 bytes
Tag Data	N bytes
…	…
Tag ID	8 or 12 bytes
Tag Data	N bytes

The data is returned in the minimal number of bytes required to hold the requested number of bits. For example, 45 bits will be returned in 6 bytes.

The final response packet contains an inventory summary.

Final Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

2.4.2.5 Class 0+ Write Page Command ($1D_h$)

The Matrics (Symbol) Class 0+ tag standard, an extension to EPC Global Class 0, has a re-writeable tag memory organized into pages. Class 0+ tags currently in production have three of these writable pages. These three pages are Page ID0, ID2 and ID3. The Class 0 memory space is described in more detail in the Application Note "RFID Tag Memories". Writing to these tags is described in the Application Note "Programming Tag Memory with the MPR". This command singulates one Class 0 tag in the MPR's field that matches the filter bits and writes the data bits, starting at the start bit, to the requested page.

Request Packet

Opcode	0x1D

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Singulation (Negotiation Page) Field	1 byte	0x00 (ID0) 0x01 (ID1) 0x02 (ID2)
Page	1 byte	0x00 (Page ID0) 0x02 (Page ID2) 0x03 (Page ID3)
Start Bit	1 byte	The bit of the page from which to start writing
Data Bit Count	1 byte	0-120, The number of bits to read from the requested row
Data Bits	0-15 bytes	The data to be written to the tag
Filter Bit Count	1 byte	Number of filter bits with which to match tag ID (0-64 or 0-96)
Filter Bits	0-12 bytes	Tag ID filter bits are left justified in bytes, MSByte is sent first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response Packet

Status	0x00 (Complete)
# Tags Written	1, or 0 if no tag singulated

2.4.2.6 Class 0 Global Read Page Command ($1E_h$)

Class 0 tag memory is organized into pages. Class 0 tags currently in production have two of these pages available to be read. These two pages are Page ID2 and ID3. The Class 0 memory space is described in more detail in the Application Note "RFID Tag Memories". This command reads a number of bits from the requested page. It does NOT singulate a tag before reading. It is similar to the Class 1 Verify command in that the raw bits decoded from the air protocol are reported. No CRC or other validity checks are made on this data.

Request Packet

Opcode	0x1E

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Page	1 byte	0x02 (Page ID2) 0x03 (Page ID3)
Read Bit Count	1 byte	0-120 The number of bits to read from the requested row

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

The response is composed of one or more packets. The non-final packets will have a Status Byte of 0x01 (In Progress) and will contain a tag ID and data for each singulated Tag. The final packet will contain an inventory summary and a Status Byte of 0x00 (Complete).

The non-final response packets will all have their Tag ID field filled with a dummy 96-bit tag of all 0's. Only one Tag ID + Data pair will be returned per command.

Non-final Response Packet

Status	0x01 (In Progress)
Number of Tag IDs in Packet	1 byte
Tag ID	8 or 12 bytes
Tag Data	N bytes

The data is returned in the minimal number of bytes required to hold the requested number of bits. For example, 45 bits will be returned in 6 bytes (N = 6 in this case).

The final response packet contains an inventory summary.

Final Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

2.4.2.7 Class 0+ Program EPC Command ($1F_h$)

The Matrics (Symbol) Class 0+ tag standard, an extension to EPC Global Class 0, stores the Tag ID in memory page ID2. This command first singulates a tag using ID0, then performs all the steps necessary to program a supplied EPC, and its computed CRC, into memory page ID2 of the singulated tag. The command operates only on the first tag singulated. The tag MUST be erased using the Class 0+ Erase Page command before programming. Since this command is used to program EPCs, Data Bit Count must be 64 or 96, and Data Bits must have the appropriate length. No check is made to ensure that the EPC header supplied (the first byte of the EPC) is valid for the requested EPC length. The EPC is verified by reading back ID2 after programming.

Request Packet

Opcode	0x1F

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Options	1 byte	Reserved for future features. Leave as 0x00.
Data Bit Count	1 byte	64 or 96, the length of the EPC to be written
Data Bits	0-15 bytes	The EPC to be written to the tag

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

2.4.2.8 Zuma Commands (18_h)

The Impinj Field Rewritable Class 0 (Zuma) RFID Tag capabilities are accessed via a single command (0x18). Specific Zuma operations are accessed by subcommands.

2.4.2.8.1 Zuma Write Row Subcommand (00_h)

Writes a row (3 bytes) of data to a specified row address of a Zuma Tag. If a filter is supplied, tags are first singulated then processed. If no filter is supplied, the global mode is used for processing.

Request Packet

Opcode	0x18

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Singulation (Negotiation Page) Field	1 byte	0x00 (ID0) 0x01 (ID1) 0x02 (ID2)
Subcommand	1 byte	0x00
Row Number	1 byte	0-15
Row Data	3 bytes	18 bits are right justified in bytes, MSByte is sent first Valid range: 0x000000 – 0x3FFFFF
Filter Bit Count	1 byte	Number of filter bits with which to match tag ID (0-64 or 0-96)
Filter Bits	0-12 bytes	Tag ID filter bits are left justified in bytes, MSByte is sent first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.
Note: If writing to row 0 or at the completion of writing the EPC ID and CRC, the tag MUST be power cycled.

Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

2.4.2.8.2 Zuma Read Row Subcommand (01_h)

Reads a row of data from a Zuma tag. This subcommand only operates on singulated tags.

Request Packet

Opcode	0x18

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Singulation (Negotiation Page) Field	1 byte	0x00 (ID0) 0x01 (ID1) 0x02 (ID2)
Subcommand	1 byte	0x01
Row Number	1 byte	0-15
Filter Bit Count	1 byte	0-64 or 0-96
Filter Bits	0-12 bytes	Tag ID filter bits are left justified in bytes, MSByte is sent first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

The response is composed of one or more packets. The non-final packets will have a Status Byte of 0x01 (In Progress) and will contain a tag ID and a row of data for each singulated Tag. The final packet will contain an inventory summary and a Status Byte of 0x00 (Complete).

Non-final Response Packet

Status	0x01 (In Progress)
Number of Tag IDs in Packet	1 byte
Tag ID	8 or 12 bytes
Tag Data	3 bytes
…	…
Tag ID	8 or 12 bytes
Tag Data	3 bytes

The final response packet contains an inventory summary.

Final Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

Example

Using Antenna B, an RF Power of 0xC0, and singulating with ID1, read all tags that match 38 (0x26) Filter bits having a value of 0xC80507A000 and return data from row 13 (0x0d).

Entire Request Packet:

SOF	Node	Len	Cmd	Ant	Power	Sing	Subcmd	Row #	# Filt Bits	Filter	CRC
01	00	10	18	01	C0	01	01	0D	26	C8 05 07 A0 00	C9 93

Response for two matching EPC tags:

Response Packet (first packet)

SOF	Node	Len	Stat	#Tags	EPC Tag ID	Tag data
01	00	1C	01	02	C8 05 07 A0 00 81 09 30	00 45 67

EPC Tag ID	Tag data	CRC
C8 05 07 A0 00 81 09 2E	00 12 34	D5 B5

Response Packet (final packet)

SOF	Node	Len	Status	Total Tags Reported	Under-run errors	CRC errors	CRC
01	00	0B	00	00 02	00 0D	00 00	A1 FE

2.4.2.8.3 Zuma INIT Subcommand (02_h)

Performs a Zuma INIT command. If a filter is supplied, tags are first singulated then processed. If no filter is supplied, the global mode is used for processing.

Zuma Initialization performs the following tasks:

1. All rows of the Zuma memory are cleared to "0", including the Control Word, all Lock bits and the Kill bit.
2. The Control Word is set to "0b 02000h," indicating EPC Length = 2.
3. The Golden Word ("0b 997Ah") is loaded into the Fab Protect row.
4. If the Lock Flag parameter is set, the Lock bit of the Fab Protect row is set, locking the Fab Protect Row, and disabling all future un- and re-initializations of the tag.

Important: After INIT the tag must complete a power cycle (power off, then power on) so that the new Control Word and Fab Protect values will be properly loaded into the tag's cache.

Request Packet

Opcode	0x18

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Singulation (Negotiation Page) Field	1 byte	0x00 (ID0) 0x01 (ID1) 0x02 (ID2)
Subcommand	1 byte	0x02
Lock Flag	1 byte	0x00 (Do Not Lock) 0x01 (Lock)
Filter Bit Count	1 byte	0-64 or 0-96
Filter Bits	0-12 bytes	Tag ID filter bits are left justified in bytes, MSByte is sent first

Note: Any tag that processes this command MUST then complete a power cycle.
Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

2.4.2.8.4 Zuma Write ACK Subcommand (03_h)

Performs a Zuma Write ACK command. This subcommand only operates on singulated tags.

Request Packet

Opcode	0x18

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Singulation (Negotiation Page) Field	1 byte	0x00 (ID0) 0x01 (ID1) 0x02 (ID2)
Subcommand	1 byte	0x03
Row Number	1 byte	0-15
Row Data	3 bytes	18 bits are right justified in bytes, MSByte is sent first
Filter Bit Count	1 byte	0-64 or 0-96
Filter Bits	0-12 bytes	Tag ID filter bits are left justified in bytes, MSByte is sent first

Note: If writing to row 0 or at the completion of writing the EPC ID and CRC, the tag MUST then complete a power cycle.
Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

2.4.2.8.5 Zuma Program ID Subcommand (05_h)

Programs a Class 0 Zuma tag with a 64- or 96-bit Tag ID and 24-bit Kill Passcode. This command erases the tag, computes the Tag ID CRC, and then programs the CRC, Tag ID, and Kill Passcode. No explicit verify is performed to determine success, but the Zuma air-interface protocol provides explicit confirmations during the programming procedure. The newly programmed tag should respond to a class 0 inventory command upon completion of the Program subcommand.

Request Packet

Opcode	0x18

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Programming Options	1 byte	Reserved for future features. Leave as 0x00
Subcommand	1 byte	0x05
ID Bit Count	1 byte	64 or 96
ID Bytes	8 or 12 bytes	The Tag ID to write to the Zuma tag
Kill Passcode	3 bytes	24-bit Class 0 Kill Passcode

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

2.4.3 Class 1 Tag Commands

2.4.3.1 Class 1 Inventory (21_h)

Returns a list of all Class1 tags found in the field of the reader.

Request Packet

Opcode	0x21

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Filter Bit Count	1 byte	0-64 or 0-96
Filter Bits	0-12 bytes	Tag ID filter bits are left justified in bytes, MSByte sent first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response:

The response is composed of one or more packets. The non-final packets will have a Status Byte of 0x01 (In Progress) and will contain Tag IDs. The final packet will contain an inventory summary, and a Status Byte of 0x00.

The reader attempts to read tags until a fixed, maximum number (7) of tag errors occur. There are two types of errors: Under-run errors occur when not enough bits replied from the tag. CRC errors occur when a CRC calculation on the received bits does not match the CRC reported by the tag. When the maximum number of errors occurs, the reader stops attempting reads and completes the command. Therefore, the total number of errors (CRC plus under-run) indicated in the Complete Response packet, should always equal the fixed maximum number of errors (7).

Non-final Response Packet

Status	0x01 (In Progress)
Number of Tag IDs in Packet	1 byte
Tag ID	8 or 12 bytes
…	…
Tag ID	8 or 12 bytes

Tag IDs are returned MSByte first. This first byte of the tag's ID is the EPC tag header. It must be used to determine the tag length. If the two highest order bits are 0b00, the tag ID is an EPC-96, which has 12 ID bytes. All other values indicate an EPC-64 tag with 8 ID bytes.

Tag Type	Tag ID Length (bytes)	Header 2 MS bits
EPC-64	8	01, 11, 10
EPC-96	12	00

The final response packet contains an inventory summary.

Final Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

Example

Using Antenna A, and RF Power 0xB0, read all Class 1 tags that match 38 (0x26) Filter bits having a value of 0xC80507A000.

Entire Request Packet:

SOF	Node	Len	Command	Ant	Power	Filt Bits	Filter	CRC
01	00	0D	11	00	B0	26	C8 05 07 A0 00	44 B5

Response for two matching EPC tags:

Response Packet (first packet)

SOF	Node	Len	Status	Num Tag IDs	EPC Tag ID	EPC Tag ID	CRC
01	00	16	01	02	C8 05 07 A0 00 81 09 30	C8 05 07 A0 00 81 09 2E	39 0B

Response Packet (final packet)

SOF	Node	Len	Status	Total Tags Reported	Under-run errors	CRC errors	CRC
01	00	0B	00	00 02	00 0D	00 00	37 F0

2.4.3.2 Class 1 Kill Tag (22_h)

This command renders the Class 1 tag specified by the supplied tag ID non-functional. The tag will no longer respond to commands of any type.

In some Class 1 tag implementations, the kill operation clears the tag memory, including the CRC, so it will not be read during an inventory command. If the tag has not been previously locked, however, programming commands (verify, erase, write) can still be used, so a killed tag can be re-programmed.

Request Packet

Opcode	0x22

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Kill Password	1 bytes	As required to kill the tag, must match passcode programmed into tag with tag ID.
Tag ID bits	8 or 12 bytes	Tag ID bits, MSB first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

The response to this command does not contain any command related data. The status in the response should be used for protocol verification.

Response Packet

Status	0x00	Complete

2.4.3.3 Class 1 Program ID (28_h)

Programs a Class 1 tag with a 64- or 96-bit Tag ID and 8-bit Kill Passcode. This command erases the tag, computes the EPC Class 1 CRC, and then programs the CRC, Tag ID, and Kill Passcode. After writing an explicit verify is performed to determine success writing to the tag memory.

Request Packet

Opcode	0x28

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Programming Options	1 byte	Reserved for future features. Leave as 0x00
ID Bytes	8 or 12 bytes	The Tag ID to write to the Class 1 tag
Kill Passcode	1 byte	8-bit Class 1 Kill Passcode

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

2.4.3.4 Class 1 Tag Write (23_h)

Write 16 bits to a Class 1 Tag starting at the Pointer bit. The pointer must point to a 16-bit boundary. Thus, it must be one of: 0, 16, 32, 48, 64, 80, 96, 112, 128.

The Lock ID function is accomplished using this command and writing the lock code **(0xA5)** to the last byte of the memory.

Request Packet

Opcode	0x23

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Pointer	1 byte	The bit address at which to start writing. Must be on a 16-bit boundary: 0, 16, 32, 48, 64, 80, 96, 112 or 128.
Data	2 bytes	16 bits, sent MSByte first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

The response to this command does not contain any command related data. The status in the response acknowledges proper receipt and processing of the request. No verification is performed to determine if the bits were properly written to the tag.

Response Packet

Status	0x00	Complete

2.4.3.5 Class 1 Verify ID (24_h)

Read all bits directly from tag memory using the Class 1 VERIFY air interface command. This command returns whatever is read from the air interface. A Tag is NOT singulated before reading. The raw bits decoded from the air protocol are reported. No CRC or other validity checks are made on this data, so the data may appear corrupted if a good air link is not established between a single tag and the MPR.

Request Packet

Opcode	0x24

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response:

The response is composed of one or more packets. The non-final packets will have a Status Byte of 0x01 (In Progress) and will contain Tag CRCs, Ids and Passwords. The final packet will contain a summary, and a Status Byte of 0x00 (Complete).

Non-final Response Packet

Status	0x01 (In Progress)	In Progress
Number of Tag IDs in Packet	1 byte	
Tag CRC (MSByte first)	1 byte	MSByte first
Tag ID	8 or 12 bytes	
Tag Kill Passcode	1 byte	
…	…	
Tag CRC (MSByte first)	1 byte	MSByte first
Tag ID	8 or 12 bytes	
Tag Kill Passcode	1 byte	

The final response packet contains an inventory summary.

Final Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

2.4.3.6 Class 1 Erase ID (25_h)

Erase the ID that was previously programmed into a Class 1 tag.

Request Packet

Opcode	0x25

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

The response to this command does not contain any command related data. The status in the response should be used for protocol verification.

Response Packet

Status	0x00	Complete

2.4.3.7 Class 1 Single Tag Read (26_h)

Read a single EPC tag in the field of the reader that matches the given filter.

Request Packet

Opcode	0x26

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Retries	1 byte	Number of times to try to read a tag before failing.
Filter Bit Count	1 byte	0-64 or 0-96
Filter Bits	0-12 bytes	Tag ID filter bits are left justified in bytes, MSByte sent first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response:

The response is composed of one or 2 packets. If a tag was found, the first packet will have a Status Byte of 0x01 (In Progress) and will contain the number of tags in the packet (always 1) and the Tag Id. The final packet will contain a summary, and a Status Byte of 0x00 (Complete).

Non-final Response Packet

Status	0x01 (In Progress)	In Progress
Number of Tag IDs in Packet	1 byte	
Tag ID	8 or 12 bytes	

The final response packet contains an inventory summary.

Final Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

2.4.3.8 Class 1 Inventory with Tree-Walk (27_h)

Returns a list of all Class1 tags found in the field of the reader. This command is nearly identical to "Class 1 Inventory", except it uses a different algorithm. This algorithm may take slightly longer to perform, but should be significantly more accurate when reading larger populations of tags.

Request Packet

Opcode	0x27

Command Data

Antenna	1 byte	0x00: Ant A 0x01: Ant B
RF Power Level	1 byte	MinPower to MaxPower, in increments of 1 dBm MPR5000, MPR6000: 15-27 **MPR7000: 18-30**
Filter Bit Count	1 byte	0-64 or 0-96
Filter Bits	0-12 bytes	Tag ID filter bits are left justified in bytes, MSByte sent first

Note: If an RF Power Level outside the valid range is requested, an Unsupported RF Power Level Error (0xF3) is returned.

Response:

The response is composed of one or more packets. The non-final packets will have a Status Byte of 0x01 (In Progress) and will contain Tag IDs. The final packet will contain an inventory summary, and a Status Byte of 0x00.

The reader attempts to read tags until a fixed, maximum number (7) of tag errors occur. There are two types of errors: Under-run errors occur when not enough bits replied from the tag. CRC errors occur when a CRC calculation on the received bits does not match the CRC reported by the tag. When the maximum number of errors occurs, the reader stops attempting reads and completes the command. Therefore, the total number of errors (CRC plus under-run) indicated in the Complete Response packet, should always equal the fixed maximum number of errors (7).

Non-final Response Packet

Status	0x01 (In Progress)
Number of Tag IDs in Packet	1 byte
Tag ID	8 or 12 bytes
…	…
Tag ID	8 or 12 bytes

Tag IDs are returned MSByte first. This first byte of the tag's ID is the EPC tag header. It must be used to determine the tag length. If the two highest order bits are 0b00, the tag ID is an EPC-96, which has 12 ID bytes. All other values indicate an EPC-64 tag with 8 ID bytes.

Tag Type	Tag ID Length (bytes)	Header 2 MS bits
EPC-64	8	01, 11, 10
EPC-96	12	00

The final response packet contains an inventory summary.

Final Response Packet

Status	0x00 (Complete)
Total Tags reported	2 bytes
Under-run error count	2 bytes
Tag CRC error count	2 bytes

Example

Using Antenna A, and RF Power 0xB0, read all Class 1 tags that match 38 (0x26) Filter bits having a value of 0xC80507A000.

Entire Request Packet:

SOF	Node	Len	Command	Ant	Power	Filt Bits	Filter	CRC
01	00	0D	11	00	B0	26	C8 05 07 A0 00	44 B5

Response for two matching EPC tags:

Response Packet (first packet)

SOF	Node	Len	Status	Num Tag IDs	EPC Tag ID	EPC Tag ID	CRC
01	00	16	01	02	C8 05 07 A0 00 81 09 30	C8 05 07 A0 00 81 09 2E	39 0B

Response Packet (final packet)

SOF	Node	Len	Status	Total Tags Reported	Under-run errors	CRC errors	CRC
01	00	0B	00	00 02	00 0D	00 00	37 F0

Section 3 Host Side Drivers

3.1 Standard Serial over 16-bit PC Card™ (PCMCIA) Interface

The MPR appears as a standard COM port to the host. Commands are sent and received via standard operating system serial interface APIs that can be accessed by all common programming languages.

The MPR, when used in a Windows® 95/98/XP operating system will use the standard Microsoft serial drivers. An information file (Microsoft® Windows® .inf file: *WJCIRFID.inf*) is provided to associate the MPR5000 with the Windows® 95/98/XP built-in serial drivers. During the installation process, the user will be prompted to supply this file.

During the installation of the MPR Series PC Card™ on Windows® XP platforms, the user will be warned that the drivers are not digitally signed.

Warning
Microsoft strongly recommends you only use device drivers with the **Designed for Microsoft® Windows® XP** logo. Installing device drivers that have not been digitally signed by Microsoft may disable the system, allow viruses onto your computer, or otherwise impair the correct operation of your computer either immediately or in the future.

This is not a problem. The MPR will work properly without the digital signature. This hardware uses *Microsoft's own drivers*! There is no additional risk of disabling the system, allowing viruses on your computer or to otherwise impair the correct operation of your computer.

3.2 Special Functions

The vast majority of signaling with the MPR is done using the host protocol commands described in previous sections of this document. This signaling is done per the serial port settings described in Table 1. One special function that the host may perform is to hard reset the MPR. This function is initiated by first changing the COM port settings to the values specified in Table 6, by sending a sequence of bytes listed in Section 3.2.1.

Table 6 Special Function COM port settings

Baud Rate	Parity	Stop Bits	Handshaking	Data Bits
57600	None	1.5 (See Note)	None	5

Note: Some serial port APIs do not correctly interpret the 1.5 stop bit setting (LabView's VISA is one such API). If these functions don't work, try using 2 stop bits instead of 1.5.

Always return the serial settings to those specified in* Table 1 *(2.1.1) for normal operation of the reader.

3.2.1 Hard Reset

This function performs a hard reset of the MPR5000. This could be used to cause the MPR to recover from a catastrophic fault, or to return it to its initial powered-on state.

Byte Sequence: `0x10, 0x00`

Appendix A. Opcodes

Table 7 Command Opcodes

Opcode	Command
0x01	Get Reader Info
0x05	Enter Bootloader
0x11	Class 0 Inventory
0x12	Class 0 Kill Tag
0x18	Class 0 Zuma Commands (see subcommand list)
0x1B	Class 0+ Erase Page
0x1C	Class 0+ Read Page
0x1D	Class 0+ Write Page
0x1E	Class 0+ Global Read Page
0x1F	Class 0+ Program ID
0x21	Class 1 Inventory
0x22	Class 1 Kill Tag
0x23	Class 1 Write
0x24	Class 1 Verify
0x25	Class 1 Erase
0x26	Class 1 Single Tag Read
0x27	Class 1 Inventory with Tree-Walk
0x28	Class 1 Program ID
0x45	GPIO

Table 8 Zuma Subcommand Opcodes

Code	Meaning
0x00	Write Row
0x01	Read Row
0x02	INIT Tag Memory
0x03	Write Row with Acknowledge
0x05	Program ID

Appendix B. Status Codes

Table 9 Response Status Codes

Code	Meaning
0xFF	Error
0x00	Complete
0x01	In Progress

Appendix C. Error Codes

Table 10 Error Codes

Code	Meaning	Description
0xDA	Ping Mask Overflow	Internal Firmware Error while Performing Class 1 Inventory
0xDB	Ping Mask Negative Shift	Internal Firmware Error while Performing Class 1 Inventory
0xDC	Ping Job Overflow	Internal Firmware Error while Performing Class 1 Inventory
0xDD	Current Out Of Range	Supply Current Measured is out of the specified range.
0xDE	Voltage Out Of Range	Supply Voltage Measured is out of the specified range.
0xDF	App Header Access Error	Existence of a valid Application Header could not be confirmed.
0xE0	Error Erasing Page	An error occurred while Erasing a Class 0+ memory page.
0xE1	Error Programming Page	An error occurred while Programming a Class 0+ memory page.
0xE2	Error Programming TIB	An error occurred while Setting the Class 0+ Traversal Inhibit Bit (TIB).
0xE3	Error Locking Page	An error occurred while Locking a Class 0+ memory page.
0xE4	Error Page Locked	Operation could not be completed because the requested memory page is already locked.
0xE5	Verify Failed	Verification of a programmed EPC failed.
0xE6	Error Reading Page	An error occurred while Reading a Class 0+ memory page.
0xE7	No Tag Singulated Error	No tag was found while trying to perform a singulated command.
0xF0	Invalid command parameter(s)	A parameter is out of range for the issued command
0xF1	Insufficient data	Based on the command issued and the message length, there is not enough data in the message to support the command
0xF2	Command not supported	The command is not supported or is unable to be processed by the Reader at this time
0xF3	Unsupported RF Power Level Error	A command has been issued with an unsupported power level
0xF4	PLL Lock Fail	On commanding the transmit frequency, the PLL which generates the RF up conversion failed to converge on the required frequency
0xF5	Antenna Fault (not present or shorted)	The antenna was either not present or shorted
0xF6	Subcommand not supported / Unknown subcommand	The subcommand issued is not one of the subcommands that can be processed by the reader
0xF7	Invalid subcommand parameter(s)	A subcommand parameter is out of range for the issued subcommand
0xF8	Invalid NVS table	The Non-Volatile Storage table CRC is invalid
0xFE	Invalid Command	The command entered is not supported in the current mode, e.g., maintenance mode commands are only valid while in maintenance mode.
0xFF	Undefined Error	

Note: Error Codes are present in the first data byte of a Response message when the Response Status Code = 0xFF (Error).

Section 4 LEGAL NOTICES

Please note that there are very specific requirements governing the terms and usage of the MPR Series RFID Cards.

4.1 Regulatory Compliance

The MPR5000 and MPR6000 products received license grants as Part 15 Spread spectrum Transmitters on 12 December, 2004.

4.1.1 FCC Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

-- Reorient or relocate the receiving antenna.
-- Increase the separation between the equipment and receiver.
-- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
-- Consult the dealer or an experienced radio/TV technician for help.

NOTE: Changes or modifications not expressly approved by WJ Communications could void the user's authority to operate the equipment described in this manual.

The MPR6000 has been approved for use only with approved external antennas described in this manual; use of any other antenna may void the user's authority to operate the equipment.

4.1.2 RF Radiation Exposure Statement

These devices comply with FCC radiation exposure limits set forth for an uncontrolled environment, and users must follow specific operating instructions for satisfying RF exposure compliance.

To comply with RF radiation exposure requirements in FCC's Rules, the MPR6000 product must be installed so there is a separation distance of at least 23 cm (9 in) between all persons and the antenna. These devices may not be co-located with any other transmitter or transmitter antenna.

Model MPR5000 has been SAR – evaluated and is authorized for use in laptop and notebook computers.

4.2 Generic Legal Notices

This document is copyrighted© property of WJ Communications Inc. (WJ) and is provided for reference only. They are not to be reproduced, distributed, copied, licensed, resold or transferred in any manner, in whole or in part, without prior written approval from WJ. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THIS INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. THE ENTIRE RISK ARISING OUT OF USE OF THIS INFORMATION REMAINS WITH THE READER/USER.

To the maximum extent permitted by applicable law, in no event shall WJ be liable for incidental, consequential, direct, punitive, special or any other damages whatsoever (including, without limitation, damages for loss of business profit, business interruption, loss of business information, or any other pecuniary loss) arising out of the review, use of, misuse of, or inability to use this information. The maximum amount of WJ's liability and your exclusive remedy under these terms shall not exceed $5.

Please note that WJ's MPR Series RFID Cards and related RFID products are not authorized for use as critical components in medical and life support applications where product performance could affect or create a situation of personal injury or death.