****

ABxS Command 8BH: Write Family Code All

DESCRIPTION

This command can be used to write the Family Code of all tags in the field. It can also be used to selectively rewrite the family codes of tags with Family Codes already set to a particular value.

The command is formatted as shown below.

	Remarks
Command	Command number in hex preceded by AAH
Current Family Code	Current Tag Family Code
Reserved	00H
Timeout	Timeout value given in 1 ms units (001EH - FFFEH)
New Family Code	The value which will be written to the FamilyCodes of all the addressed tags
Message Terminator	FFFFH

Example

This example will change the Family code depending on the current value of the family code. Two tags are found with Family Code 02H and they are changed to 03H.

Command from hoast		
MSB	**LSB**	**Remarks**
AAH	8BH	Perform Command 8BH
02H	00H	Family ID/ Reserved
07H	D0H	2 second Timeout
00H	03H	new family code
FFH	FFH	Message Terminator

	LSB	**Remarks**
AAH	8BH	Command Echo
02H	00H	Ntags/Status
FFH	FFH	Message Terminator

****

ABxS Command 8CH: Lock Family Code All

DESCRIPTION

This command locks the family codes of tags in the field. Once locked, the Family Codes cannot be changed or unlocked.

DISCUSSION

The command must pass a family code to select the tags whose family code will be locked. The controller will return a response when the timeout period expires. The parameter of the response in the number of tags written.

The command is formatted as shown below.

	Remarks
Command	Command number in hex preceded by AAH
Family Code	Family Code, 00H = all tags
Reserved	00H
Timeout	Timeout value given in 1 ms units (001EH - FFFEH)
Message Terminator	FFFFH

Example

This example will lock the Family Code on all tags with the Family Code of 02H. Two tags are found and locked.

Command from host		
MSB	**LSB**	**Remarks**
AAH	8CH	Perform Command 8CH
02H	00H	Family ID/ Reserved
07H	D0H	2 second Timeout

Response from controller		
MSB	**LSB**	**Remarks**
AAH	8BH	Command Echo
02H	00H	Ntags/Status
FFH	FFH	Message Terminator

Command from host		
MSB	**LSB**	**Remarks**
FFH	FFH	Message Terminator

Response from controller		
MSB	**LSB**	**Remarks**

****

ABxS Command 8DH: Start/Stop Continuous Read All

DESCRIPTION

Starts and stops Continuous Read All mode for multiple tags.

DISCUSSION

The Start/Stop Continuous Read All mode is set by the length byte. To start Continuous Read All mode send the command with valid, non-zero value for the length of the read . Stop the mode by sending the command with a read length of 0. While in this mode, any other command can be issued and it will be handled properly. After processing the new command, the controller will resume the Continuous Read All mode. The command has a parameter, tag repeat count, that can prevent multiple reads of the same tag. A tag is not read a second time until a specified number of tags have been read since it was last read. Allowed values are from 0 to 255 (FFH), where 0 means the tag can be re-read anytime. When Continuous Read All mode is interrupted with other commands, the tag repeat count is saved during execution of the other commands such that all tags will respond to the interrupting command. When the controller resumes continuous reading, the tag repeat count will be restored. The controller will respond with an acknowledge packet followed by data packets for each tag read. CONFIG LED blinks after each packet transmission.

Field	Remarks
Command	Command number in hex preceded by AAH
Family ID	Tag Family ID - 00H = all tags
Reserved	00H
Start Address	Tag address for the start of the read
Read Length	nonzero = start, 0000H = stop

Field	Remarks
Tag Delay	Number of tag that must be read before the same tag will be read again(0-255)
Message Terminator	FFFFH

Example

Reads 4 bytes of data from the tag starting at address 0001H. TheFamily ID byte is set to zero so all tags will be read. The Tag Repeat Count is set to 20 (14H). Three tags respond with read data.

MSB	**LSB**	**Remarks**
AAH	8DH	Perform Command 8D
00H	00H	Tag Family 00/Reserved
00H	01H	Start Address
00H	04H	Read Length
00H	14H	Tag Delay
FFH	FFH	Message Terminator

	LSB	**Remarks**
AAH	8DH	Command Echo
FFH	FFH	Message Terminator

After the controller sends the acknowledgment, it will send the read data from the 3 tags.

Data response from controller

MSB	LSB	Remarks
AAH	8DH	Command Echo/ Tag 1
00H	30H	Data byte 1/Tag 1
00H	31H	Data byte 2/Tag 1
00H	32H	Data byte 3/Tag 1
00H	33H	Data byte 4/Tag 1
FFH	FFH	Terminator/ Tag 1
AAH	8DH	Command Echo/ Tag 2
00H	40H	Data byte 1/Tag 2
00H	41H	Data byte 2/Tag 2

MSB	LSB	Remarks
00H	42H	Data byte 3/Tag 2
00H	43H	Data byte 4/Tag 2
FFH	FFH	Terminator/ Tag 2
AAH	8DH	Command Echo/ Tag 3
00H	34H	Data byte 1/Tag 3
00H	35H	Data byte 2/Tag 3
00H	36H	Data byte 3/Tag 3

****

ABxS Command 8EH: Memory Lock All

DESCRIPTION

This command locks contiguous blocks of read-write memory. Once Bytes are locked, they cannot be written to, nor can they be unlocked.

DISCUSSION

The memory can be locked only in contiguous blocks. The command passes one parameter for the first block and another parameter for the number of blocks to be locked. This is one of only two ABx commands which address the memory of the tag using these parameters. The block structure of the tags is governed by the ISO-15693 specification. ABx command 15H can be used to obtain the memory organization for a particular tag. Attempting to write to locked Bytes will return a Write Security Error.

The command is formatted as shown below.

Field	Remarks
Command	Command number in hex preceded by AAH
Family Code	Tag Family ID - 00H = all tags
Reserved	00H
Timeout	Timeout value given in 1 ms units (001EH - FFFEH)
First Block	First block of memory to lock
Number of blocks	Number of contiguous blocks of memory to lock

Field	Remarks
Message Terminator	FFFFH

Example

This example will lock blocks 0-8 on all tags in range with the Family ID of 02H. Two tags are found and locked.

MSB	**LSB**	**Remarks**
AAH	8EH	Perform Command 8E
02H	00H	Family ID/ Reserved
07H	D0H	2 second timeout
00H	08H	Lock Configuration
FFH	FFH	Message Terminator

	LSB	**Remarks**
AAH	8EH	Command Echo
02H	08H	Ntag/Status
FFH	FFH	Message Terminator

****

ABxS Command 94H: SN Fill

DESCRIPTION

Fills only the RFID tag specified by serial number with a one byte value over multiple contiguous addresses.

DISCUSSION

This command is commonly used to clear an RFID tag's memory. It writes a one byte value repetitively across a specified range of tag addresses. Only the tag with the specified serial number will be affected by this command. The controller will return a response after the successful fill operation or when the timeout expires. The fill function requires a Family ID and a Serial Number, one data value byte, a starting address, and a fill length. It fills the specified tag with the data value byte, starting at the specified start address for the specified number of consecutive bytes. When Fill Length is set to 0, the controller will write fill data from the start address to the end of the tag's memory. The timeout value is given in 1 msec increments and can have a value of 001EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return a syntax error. It returns a response when done or when the timeout expires.

Field	Remarks
Command	Command number in hex preceded by AAH
Family Code	Tag Family ID - 00H = all tags
Reserved	00H
Start Address	The tag address where the fill will start

Field	Remarks
Fill Length	The number of tag addresses to be filled
Timeout	Timeout value given in 1 ms units (001EH - FFFEH)
Tag Serial Number	The 8-byte serial number
Data Value Byte	The byte used to fill
Message Terminator	FFFFH

A response to a successful command will follow this form.

	Remarks
Command	Echo Command number in hex preceded by AAH
Number of Tags filled	0 = tag not found, 1 = tag filled
Command Status	One byte Error status
Message Terminator	FFFFH

Example

Writes 'A' (41H) to a single tag, starting at tag address 0005H for the following next consecutive 40 bytes. The Family Code is set to 00H. A timeout of 2 seconds (07D0H = 2000 x 1 msec increments) is set for the completion of the command.

Command from host		
MSB	**LSB**	**Remarks**
AAH	94H	Perform Command 94
00H	00H	Family ID/ Reserved
00H	05H	Start Address
00H	28H	Fill Length
07H	D0H	Timeout
00H	01H	SN byte 0
00H	ACH	SN byte 1
00H	42H	SN byte 2

Response from controller		
MSB	**LSB**	**Remarks**
AAH	94H	Command Echo
01H	00H	Ntag/Status
FFH	FFH	Message Terminator

Command from host		
MSB	**LSB**	**Remarks**
00H	D0H	SN byte 3
00H	27H SN byte 4	00H
1CH	SN byte 5	00H
65H	SN byte 6	00H
33H	SN byte 7	00H
41H Fill byte	FFH	FFH
Message Terminator		

Response from controller		
MSB	**LSB**	**Remarks**

****

ABxS Command 95H: SN Block Read

DESCRIPTION

Read a block of data from a specified RFID tag.

DISCUSSION

This command is used to read segments of data from contiguous areas of tag memory. It is capable of handling up to 48 bytes of data transferred to the host with one command if there is no tag Family ID. The timeout value is given in 1 msec increments and can have a value of 001EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return a syntax error. The SN Block Read consists of a serial number, a start address and length, followed by a timeout value. The 8-byte serial number of the target tag is specified. If the read length exceeds the last tag address, the controller will return error message in the status byte. It returns a response when done or when the timeout expires. The data read from the tag is returned in the LSB of the register, and the MSB is always 00H. A special error packet (AAH FFH) is sent if the timeout expires.

Field	Remarks
Command	Command number in hex preceded by AAH
Tag Family	Tag Family ID - 00H = all tags

Field	Remarks
Reserved	00H
Start Address	The tag address where the read will start
Read Length	The numbers of tag addresses to be read
Timeout	Timeout value given in 1 ms units (001EH - FFFEH)
Tag Serial Number	8-byte tag serial number
Message Terminator	FFFFH

Example

Reads 4 bytes of data from the tag specified by serial number starting at address 0001H. A timeout of 2 seconds (07D0H = 2000 x 1 msec increments) is set for the completion of the SN Block Read. The Family ID byte is set to zero.

Command from host		
MSB	**LSB**	**Remarks**
AAH	95H	Perform Command 95
00H	00H	Tag Family 00/ Reserved
00H	01H	Start Address
00H	04H	Read Length
07H	D0H	Timeout
00H	ABH	SN byte 0
00H	02H	SN byte 1
00H	F3H	SN byte 2
00H	55H	SN byte 3
00H	C5H	SN byte 4
00H	2DH	SN byte 5
00H	41H	SN byte 6
00H	A0H	SN byte 7

Response from controller		
MSB	**LSB**	**Remarks**
AAH	95H	Command Echo
00H	30H	Data byte 1
00H	31H	Data byte 2
00H	32H	Data byte 3
00H	33H	Data byte 4
01H	00H	Ntag/Status
FFH	FFH	Message Terminator

Command from host		
MSB	**LSB**	**Remarks**
FFH	FFH	Message Terminator

Response from controller		
MSB	**LSB**	**Remarks**

****

ABxS Command 96 (96H): SN Block Write

DESCRIPTION

Write a block of data to a single RFID tag specified by its serial number.

DISCUSSION

This command is used to write segments of data to contiguous areas of tag memory. It is capable of transferring up to 1 kByte of data transferred from the host to the controller in a single command. The timeout value is given in 1 msec increments and can have a value of 001EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return a syntax error. The SN BlockWrite consists of a Family ID, Serial Number and start address followed by the data stream to be written to the RFID tag. If the write range exceeds the last tag address, the controller will return a syntax error, 21H. It returns a response when done or when the timeout expires. The data to be written to the tag is contained in the LSB of the register, and the MSB is always 00H.

	Remarks
Command	Command number in hex preceded by AAH
Family ID	Tag Family ID - 00H = all tags
Reserved	00H
Start Address	The tag address where the write will start
Number of Bytes	The number of tag addresses to be written to
Timeout	Timeout value given in 1 ms units (001EH - FFFEH)
Tag Serial Number	8-Byte tag serial number
Write Data	The data to be written (1 Byte to 1kByte)
Message Terminator	FFFFH

Example

Writes 4 bytes of data, starting at address 0001H. A timeout of 2 seconds (07D0H = 2000 x 1 msec increments) is set for the completion of the Block Write. The Family ID Code is set to 00H.

Command from host		
MSB	**LSB**	**Remarks**
AAH	96H	Perform Command 96
02H	00H	Tag Family 02/ reserved
00H	01H	Start Address
00H	04H	Write Length
07H	D0H	Timeout
00H	A4H	SN byte 0
00H	6CH	SN byte 1
00H	18H	SN byte 2
00H	92H	SN byte 3
00H	2DH	SN byte 4

Response from controller		
MSB	**LSB**	**Remarks**
AAH	96H	Command Echo
01H	00H	Ntags/Status
FFH	FFH	Message Terminator

Command from host		
MSB	**LSB**	**Remarks**
00H	34H	SN byte 5
00H	DEH	SN byte 6
00H	20H	SN byte 7
00H	40H	Data byte 1
00H	41H	Data byte 2
00H	42H	Data byte 3
00H	43H	Data byte 4
FFH	FFH	Message Terminator

Response from controller		
MSB	**LSB**	**Remarks**

****

ABxS Command 10H: Set Output

DESCRIPTION

Set the states of the output lines and output LEDs "A" through "D."

DISCUSSION

This command will set the state of the digital output lines using a one-Byte parameter, Output Status. The least significant nibble of the Output Status Byte will determine the status of the outputs. The least significant bit of this nibble corresponds to Output A. The most significant bit of this nibble corresponds to Output D. The most significate nibble of the Output Status Byte is reserved and should be set to 0H. An output bit set to one will turn the corresponding LED on and close the electronic switch of the output, allowing current to flow through the output circuit. An output bit reset to zero will turn the corresponding LED off and open the electronic switch in the output

circuit. The following chart shows the hex values for all combinations of Output Status Bytes. To reset all output bits, issue the command with 00H for the Output Status Byte.

	LSB	Remarks	LSB Bit 3 Output D	LSB Bit 2 Output C	LSB Bit 1 Output B	LSB Bit 0 Output A
00H	00H	Reset A, B, C, D	0	0	0	0
00H	01H	Set Output A - Reset B, C, D	0	0	0	1
00H	02H	Set Output B - Reset A, C, D	0	0	1	0
00H	03H	Set Output A, B - Reset C, D	0	0	1	1
00H	04H	Set Output C - Reset A, B, D	0	1	0	0
00H	05H	Set Output A, C - Reset B, D	0	1	0	1
00H	06H	Set Output B, C - Reset A, D	0	1	1	0
00H	07H	Set Output A, B, C - Reset D	0	1	1	1
00H	08H	Set Output D - Reset A, B, C	1	0	0	0
00H	09H	Set Output A, D - Reset B, C	1	0	0	1
00H	0AH	Set Output B, D - Reset A, C	1	0	1	0
00H	0BH	Set Output A, B, D - Reset C	1	0	1	1
00H	0CH	Set Output C, D - Reset A, B	1	1	0	0
00H	0DH	Set Output A, C, D - Reset B	1	1	0	1
00H	0EH	Set Output B, C, D - Reset A	1	1	1	0
00H	0FH	Set Output A, B, C, D	1	1	1	1

Field	Remarks
Command	Command number in hex preceded by AAH
Output	Pattern Hex value for the bit output settings

Field	Remarks
Message Terminator	FFFFH

Example

The following example sets Output B only and resets A, C, and D.

MSB	**LSB**	**Remarks**
AAH	10H	Perform Command 10
00H	02H	Set Output B
FFH	FFH	Message Terminator

	LSB	**Remarks**
AAH	10H	Command echo
FFH	FFH	Message Terminator

****

ABxS Command 11H: Input Status

DESCRIPTION

Returns the status of User Inputs.

DISCUSSION

This command will interrogate the state of the User Input lines and return a one-Byte parameter, Input Status. The least significant nibble of the Input Status Byte is determined by the status of the outputs. The least significant bit of this nibble corresponds to Input A. The most significant bit of this nibble corresponds to Input D. The most significate nibble of the Input Status Byte will be set to 0H. A closed circuit in one of the User Inputs will set the corresponding User Input bit to one, turning the corresponding LED on. An open circuit in one of

the User Inputs will set the corresponding User Input bit to zero, turning off the corresponding LED. The following chart shows all of the valid hex values for the Input Status bytes.

	LSB	Remarks	LSB Bit 3 Input D	LSB Bit 2 Input C	LSB Bit 1 Input B	LSB Bit 0 Input A
00H	00H	Inputs A, B, C, D, OFF	0	0	0	0
00H	01H	Input A, ON - B, C, D, OFF	0	0	0	1
00H	02H	Input B, ON - A, C, D, OFF	0	0	1	0
00H	03H	Input A, B, ON - C, D, OFF	0	0	1	1
00H	04H	Input C, ON - A, B, D, OFF	0	1	0	0
00H	05H	Input A, C, ON - B, D, OFF	0	1	0	1
00H	06H	Input B, C, ON - A, D, OFF	0	1	1	0
00H	07H	Input A, B, C, ON - D, OFF	0	1	1	1
00H	08H	Input D, ON - A, B, C, OFF	1	0	0	0
00H	09H	Input A, D, ON - B, C, OFF	1	0	0	1
00H	0AH	Input B, D, ON - A, C, OFF	1	0	1	0
00H	0BH	Input A, B, D, ON - C, OFF	1	0	1	1
00H	0CH	Input C, D, ON - A, B, OFF	1	1	0	0
00H	0DH	Input A, C, D, ON - B, OFF	1	1	0	1
00H	0EH	Input B, C, D, ON - A, OFF	1	1	1	0
00H	0FH	Input A, B, C, D, ON	1	1	1	1

Field	Remarks
Command	Command number in hex preceded by AAH
Message Terminator	FFFFH

Example

The following example shows only Input B is ON.

MSB	**LSB**	**Remarks**
AAH	11H	Perform Command 11
FFH	FFH	Message Terminator

	LSB	**Remarks**
AAH	11H	Command echo
00H	02H	Input B ON
FFH	FFH	Message Terminator

6.4 Abx Fast Protocol

The ABx Fast protocol differs from the Abx Standard Protocol in that the atomic data element is a Byte instead of a 16-bit word. ABx Fast commands and responses also contain a two-Byte word to indicate the size of the packet being sent. ABx Fast also supports the use of a one-Byte checksum.

Packet Size

The ABx Fast protocol requires that the size of the packet be included following the terminator in every packet. All parameters and data between the packet size Byte and the checksum (if used) or terminator Bytes (if not using the checksum) must be accounted for in the packet size word. The packet size will be the same with, or without, a checksum.

Checksum

The optional checksum must be enabled from the operating mode menu to be available. The checksum is calculated by adding all the Byte values in the packet (less the values in the header, checksum if present, and terminator), discarding Byte overflow and subtracting the Byte sum from FFH. Thus, when the packet length through the checksum are added as Byte values, the sum will be FFH.

Example

The following is a typical command using a checksum.

	Contents
Header	0202H
Command Size	0003H
Command Code	01H
Timeout	07D0H
Checksum	24H
Terminator	03H

The summed values begin with the Command Size and end with the timeout value. That sum, less overflow, is subtracted from FFH for the checksum value.

Thus:

```
00H + 03H + 01H + 07H + D0H = DBH
FFH - DBH = 24H
```

Sum these values to calculate the checksum

ABx Fast Single-tag Command Structure:

Field	Number of Bytes	Content
Header	2	0202H
Command Size	2	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes.
Command	1	Command Code
Byte Addresses	4	The first two Bytes give the start address. The second two Bytes give the length of the address range. Not used on 07H, 08H, 14H, 15H, 16H
Block Addresses	2	The first Byte gives the address of the first block. The second Byte gives the number of blocks to be interrogated. Only used with command 14H.
Timeout	2	0001H to FFFEH milliseconds
Data	varies	command data
Checksum	1	Optional Checksum

Field	Number of Bytes	Content
Terminator	1	03H

ABx Fast Multi-tag Command structure

	Number of Bytes	Content
Header	2	0202H
Command Size	2	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes.
Command	1	Command Code
Family code	1	00H to address all tags in field
Reserved	1	Reserved for future use, set to 00H
Address	4	The first two Bytes give the start address. The second two Bytes give the length of the address range Not used on 87H, 88H, 8EH, 8BH, 8CH
Block Addresses	2	The first Byte gives the address of the first block. The second Byte gives the number of blocks to be interrogated.Only used with command 14H
Timeout	2	0001H to FFFEH milliseconds
Data	varies	command data
Checksum	1	Optional Checksum
Terminator	1	03H

****

ABx Fast Command 04H: Fill Tag

DESCRIPTION

Fill an RFID tag with a one Byte value over multiple contiguous addresses.

DISCUSSION

This command is commonly used to clear an RFID tag's memory. It writes a one-Byte value repetitively across a specified range of tag addresses. The fill function requires one data value Byte, a starting address, and a fill length. It will fill the tag with the data value Byte, starting at the specified start address for the specified number of consecutive Bytes. When Fill Length is set to 0, the controller will write fill data from the start address to the end of the tag's memory. The timeout value is given in 1 msec increments and can have a value of 1EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return a syntax error. When the Fill Length extends beyond the last Byte in the tag, the controller will return error 21H

.

Field	Content
Header	0202H

Field	Content
Command Size	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes. 0008H for this command.
Command	04H
Start Address	2-Byte value for the starting tag address
Number of Bytes	2-Byte value for the length of the fill in number of Bytes
Timeout	2-Byte value for timeout in 1 ms units. (1EH - FFFEH) Bytes
Data value Byte	1 Byte value to be written to all the adddressed Bytes
Checksum	Optional Checksum
Terminator	03H

A response to a successful command will follow this form.

	Content
Header	0202H
Command Size	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes.
Command echo	04H
Checksum	Optional Checksum
Terminator	03H

Example

Writes 'A' (41H) to the tag starting at address 0005H and continuing for the next consecutive 40 Bytes. A timeout of 2 seconds (07D0H =2000 x 1 msec increments) is set for the completion of the configuration.

Command from host	
Field	**Content**
Header	0202H
Command Size	0008H

Response from controller	
Field	**Content**
Header	0202H
Response Size	0001H

Command from host	
Field	**Content**
Command Code	04H
Start address	0005H
Block Size	0028H
Timeout, 2 seconds	07D0H
Data Value Byte	41H
Checksum	AEH
Terminators	03H

Response from controller	
Field	**Content**
Command Echo	04H
Checksum	FAH
Terminators	03H

****

ABx Fast Command 05H: Block Read

DESCRIPTION

Reads a block of data from an RFID tag.

DISCUSSION

This command is used to read segments of data from contiguous areas of tag memory. It is capable of handling up to 1 kByte of data transferred to the host. The timeout value is given in 1 msec increments and can have a value of 1EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return error 21H,

syntax error. The Block Read consists of a start address and length, followed by a timeout value and a message terminator as shown below.

	Content
Header	0202H
Command Size	Packet length in Bytes excluding the header, command Size, checksum and terminator Bytes. 0007H for this command
Command	05H
Start Address	2-Byte value for the starting tag address
Number of Bytes	2-Byte value for the length of the read in number of Bytes
Timeout	2-Byte value for timeout in 1 ms units. (1EH - FFFEH)
Checksum	Optional Checksum
Terminator	03H

Example

Reads 4 Bytes of data from the tag starting at address 0001H. A timeout of 2 seconds (07D0H = 2000 x 1 msec increments) is set for the completion of the Block Read.

Command from host	
Field	**Content**
Header	0202H
Command Size	0007H
Command Code	05H
Start address	0001H
Number of Bytes	0004H

Response from controller	
Field	**Content**
Header	0202H
Response Size	0005H
Command Echo	05H
Data fromaddress 0001H	05H
Data from address 0002H	AAH

Command from host	
Field	**Content**
Timeout,2 seconds	07D0H
Checksum	17H
Terminator	03H

Response from controller	
Field	**Content**
Data fromaddress 0003H	E7H
Data from address 0004H	0AH
Checksum	55H
Terminator	03H

****

ABx Fast Command 6 (06H): Block Write

DESCRIPTION

Write a block of data to an RFID tag.

DISCUSSION

The Block Write command is used to write segments of data to contiguous areas of tag memory. It is capable of addressing 1 kByte of the read/write address space of a tag in a single command. The BlockWrite command requires as parameters the start address, the

number of bytes to be written, and the data to be written to the RFID tag. If the write range exceeds the last tag address, the controller will return an invalid format error message (error code 21H).

	Content
Header	0202H
Packet Size	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes. 0007H plus the number of data Bytes
Command	06H
Start Address	2-Byte value for the starting tag address
Number of Bytes	2-Byte value for the number of Bytes that will be written
Timeout	2-Byte value for timeout in 1 ms units. (1EH - FFFEH)
Data	Data Bytes to be written
Checksum	Optional Checksum
Terminator	03H

Example

Writes 4 Bytes of data to the tag starting at address 0000H. A timeout of 2 seconds (07D0H = 2000 x 1 msec increments) is set for the completion of the BlockWrite.

Command from host	
Field	**Content**
Header	0202H
Command Size	000BH
Command Code	06H

Response from controller	
Field	**Content**
Header	0202H
Response Size	0001H
Command Echo	06H

Command from host	
Field	**Content**
Start address	0000H
Number of Bytes	0004H
Timeout,2 seconds	07D0H
Data to write to address 0000H	52H
Data to write to address 0001H	46H
Data to write to address 0002H	49H
Data to write to address 0003H	44H
Checksum	EEH
Terminators	03H

Response from controller	
Field	**Content**
Checksum	F8H
Terminators	03H

****

ABxFast Command 07H: Read Tag Serial Number

DESCRIPTION

This command retrieves the 8-Byte tag serial number.

DISCUSSION

Each ISO-15693 compliant tag has a unique (over 280 trillion possibilities) serial number. This number can not be changed and is not part of the available read/write address space of a tag. The presence of multiple tags in the field will not cause errors, the controller will simply return the serial number that it reads. This presents a potentially dangerous uncertainty in that there is no general way to predict which tag the controller will read first.

	Content
Header	0202H
Command Size	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes. 0003H for this command
Command	07H
Timeout	2-Byte timeout value in 1 ms increments (1EH - FFFEH)
Checksum	Optional Checksum
Terminator	03H

Example

This example will wait until a tag is in range and then reads the 8-Byte serial number. In this example the serial number is F2720300000104E0.

Command from host	
Field	**Content**
Header	0202H
Command Size	0003H

Response from controller	
Field	**Content**
Header	0202H
Response Size	0009H

Command from host	
Field	**Content**
Command Code	07H
Timeout	07D0H
Checksum	1EH
Terminators	03H

Response from controller	
Field	**Content**
Command Echo	07H
First ID Byte	F2H
Second ID Byte	72H
Third ID Byte	03H
Fourth ID Byte	00H
Fifth ID Byte	00H
Sixth ID Byte	01H
Seventh ID Byte	004H
Eighth ID Byte	E0H
Checksum	A3H
Terminators	03H

****

ABxFast Command 8 (08H): Tag Search

DESCRIPTION

Checks for an RFID tag in the controller field.

DISCUSSION

This command will activate the controller to "look" for a tag in the RF field. If the controller finds a tag it will return a command echo to the host. If no tag is present it will return an error message. Even though this command is grouped with single-tag commands, the presence of multiple tags within range of the antenna will not cause errors. See Section 6.6 for information on the error messages.

	Content
Header	0202H
Command Size	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes. 0003H for this command
Command	08H
Timeout	2-Byte value for the time in 1 ms units (1EH - FFFEH)
Checksum	Optional Checksum
Terminator	03H

Example

Checks for an RFID tag within range of the antenna. A timeout of 2 seconds (07D0H = 2000 x 1 msec increments) is set for the completion of the Tag Search.

Field	**Content**
Header	0202H
Command Size	0003H
Command Code	08H
Timeout	07D0H
Checksum	1DH
Terminators	03H

	Content
Header	0202H
Response Size	0001H
Command Echo	08H
Checksum	F6H
Terminators	03H
Field	**Content**

****

ABx Fast Command 0DH: Start/Stop Continuous Block Read

DESCRIPTION

Sends block reads continuously to any tag in range of the antenna. When a tag enters the RF field, it is read and the data passed to the host computer. The controller continues to read the tag but will not send the same data to the host until the tag has been outside the RF field for a specified time period. This Delay Between Identical Decodes parameter prevents redundant data transmissions when the controller is in Continuous Block Read mode.

DISCUSSION

The Start/Stop Continuous Block Read command contains three parameters: Start Address, Number of Bytes and Delay Between Duplicate Decodes. The read length parameter switches the mode. Any valid, non-zero length will set the controller into Continuous Block Read mode. A read length value of 00H will stop the Continuous read. The Delay Between Identical Decodes parameter can have a value of 0 to 60 seconds. When the Delay Between Identical Decodes is set to 0, the controller will continuously read and transmit tag data to the host. This can flood the buffers and cause communication errors and data loss. If the controller receives other commands from the host, it will execute them and then resume Continuous Block Read mode. Issuing the command with the Number of Bytes set to 00H will stop the Continuous Block Read.

The command is formatted as follows.

	Content
Header	0202H
Command Size	Command length in Bytes excluding the header, command size, checksum and terminator Bytes.
Command	0DH
Start Address	2-Byte value for the start address in the tag
Number of Bytes	2-Byte value of the number of bytes to be addressed.
Delay Between Duplicate Decodes	Delay value given in 1 second units
Checksum	Optional Checksum
Terminator	03H

This example places the controller in Continuous Block Read mode and reads 8 Bytes of data from the tag starting at address 0001H. A delay between identical reads of 2 seconds (0002H = 2 x 1 second increments) is set.

Field	**Content**
Header	0202H
Command Size	0006H
Command Code	0DH
Start Address	0001H
Number of Bytes	0008H
Delay Between Duplicate Decodes	02H
Checksum	E1H

	Content
Header	0202H
Response Size	0001H
Command Echo	0DH
Checksum	F1H
Terminator	03H

****

ABx Fast Command 14H: Get Block Status

DESCRIPTION

Returns the lock status of the specified blocks of data.

DISCUSSION

This command can be used to determine whether blocks of tag memory are locked; marked "read-only." The number of specified contiguous blocks are addressed from the specified first block. The response from the controller gives the status of each block through a one-Byte value. The value is 00H if the block is unlocked, 01H if locked. The size and organization of the blocks in a particular tag can be found through the use of command 15H, Get Label Information.

	Content
Header	0202H
Command Size	Packet length in Bytes excluding the header, command this command size, checksum and terminator Bytes. 0005H for this command
Command	14H
First Block	One-Byte address of the first block to be Interrogated.
Number of blocks	One-Byte
Timeout	2-Byte value for timeout in 1 ms units. (1EH - FFFEH)
Checksum	Optional Checksum
Terminator	03H

Example

Field	**Content**
Header	0202H
Command Size	0005H
Command Code	14H
First Block	00H
Number of Blocks	04H
Timeout,2 seconds	07D0H
Checksum	0BH
Terminator	03H

	Content
Header	0202H
Response Size	0005H
Command Echo	14H
status of block zero	00H
status of	block one 00H
status of block two	01H
status of block three	00H
Checksum	E5H
Terminator	03H

Interrogates the lock status of the first four blocks of the tag's memory for a timeout period of 2 seconds.

****

ABx Fast Command 15H: Get Label Information

DESCRIPTION:

This command retrieves manufacturer's data and the Family Code from the tag.

	Content
Header	0202H
Command Size	Packet length in Bytes excluding the header, command size, Checksum and terminator Bytes. 0003H for this command
Command	15H
Timeout	2-Byte value for timeout in 1 ms units. (1EH - FFFEH)
Checksum	Optional Checksum
Terminator	03H

Example

This command is issued to retrieve the family code from a tag.

Command from host	
Field	**Content**
Header	0202H
Command Size	0003H
Command Code	15H
Timeout,2 seconds	07D0H
Checksum	10H
Terminator	03H

	Content
Header	0202H
Response Size	0007H
Command Echo	15H
Information Flags Byte	07H
Format Information	00H
Family Code	00H
Block Size	04H
Number of Blocks	1CH
Checksum	BCH
Terminator	03H

****

ABxFast Command 16H: Write Family Code

DESCRIPTION

Change the family code of an RFID tag.

	Content
Header	0202H
Packet Size	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes. Always 0004H for this command.
Command	16H
Timeout	2-Byte value for timeout in 1 ms units. (1EH - FFFEH)
Family Code	The value to which the family code will be changed.
Checksum	Optional Checksum
Terminator	03H

Example

Change the family code of the tag to 00H.

Command from host	
Field	**Content**
Header	0202H
Command Size	0004H
Command Code	16H
Timeout	07D0H
Family Code	00H
Checksum	FCH
Terminator	03H

	Content
Header	0202H
Response Size	0001H
Command Echo	16H
Checksum	E8H
Terminator	03H

****

ABxFast Command 17H: Lock Family Code

DESCRIPTION:

Locks the Family Code Byte to its current value so that it cannot be written. Once locked, the Family Code cannot be unlocked.

Field	**Content**
Header	0202H
Command Size	0003H
Command Code	17H
Timeout, 2 seconds	07D0H
Checksum	0EH
Terminator	03H

	Content
Header	0202H
Response Size	0001H
Command Echo	17H
Checksum	E7H
Terminator	03H

****

ABxF Command 82H: Read Data and SN All

DESCRIPTION

Read a block of data from all RFID tags-in-field or all those with the specified Family Code. Data will be returned with the serial number of the corresponding tag.

DISCUSSION

This command is used to read segments of data from contiguous areas of tag memory. It is capable of handling up to 1KByte of data transferred to the host with one command. The timeout value is given in 1 msec increments and can have a value of 1EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return a syntax error. The response to this command will contain the serial number of the responding tags preceding the data from those tags. The termination packet is transmitted when the timeout expires. Each packet will be sent to the host as soon as it is available. The returned serial numbers can be used to read from or to write to tags-in-field via the SN Block Read or Write commands. The SN Block Read All consists of a header, a family Code, a start address and length, followed by a timeout value and the message terminator, 03H, as shown below. A special termination packet is sent when the timeout expires. If the read range exceeds the last tag address, the controller will return an error message in the status Byte. The command is formatted as follows.

	Content
Header	0202H
Command Size	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes.
Command	82H
Family Code	Tag Family Code - 00H = all tags
Reserved	Ignored Byte, set to 00H
Start Address	2-Byte value for the starting tag address
Number of Bytes	2-Byte value for the number of Bytes to be read
Timeout	2-Byte value for timeout in 1 ms units. (1EH - FFFEH)
Checksum	Optional Checksum
Terminator	03H

Example

Reads 4 Bytes of data from the tag starting at address 0001H. A timeout of 2 seconds (07D0H = 2000 x 1 msec increments) is set for the completion of the SN Block Read All. The Tag Family Byte is set to zero so all tags will be read. Three tags respond with data.

Command from host	
Field	**Content**
Header	0202H
Command Size	0009H
Command Echo	82H
Family Code	03H
Reserved	00H
Start address	0000H
Block Size	0004H
Timeout,2 seconds	07D0H
Checksum	96H
Terminators <ETX>	03H

Response from controller	
Field	**Content**
Header/tag 1	0202H
Response Size	000DH
Command Code	82H
SN Byte 1/tag 1	A6H
SN Byte 2/ tag 1	72H
SN Byte 3/tag 1	03H
SN Byte 4/tag 1	00H
SN Byte 5/tag 1	00H
SN Byte 6/tag 1	01H
SN Byte 7/tag 1	04H
SN Byte 8/tag 1	E0H
Data Byte 1/ tag 1	52H
Data Byte 2/tag 1	46H
Data Byte 3/tag 1	49H
Data Byte 4/tag 1	44H
Checksum	4BH
Terminator	03H
Header/tag 2	0202H
Response Size	000DH
Command Code	82H
SN Byte 1/tag 2	F1H
SN Byte 2/ tag 2	72H
SN Byte 3/tag 2	03H
SN Byte 4/tag 2	00H
SN Byte 5/tag 2	00H
SN Byte 6/tag 2	01H

Command from host	
Field	**Content**
Header	0202H

Response from controller	
Field	**Content**
Header/tag 1	0202H
SN Byte 7/tag 2	04H
SN Byte 8/tag 2	E0H
Data Byte 1/ tag 2	52H
Data Byte 2/tag 2	46H
Data Byte 3/tag 2	49H
Data Byte 4/tag 2	44H
Checksum	00H
Terminator	03H
Header/tag 3	0202H
Response Size	000BH
Command Echo	82H
SN Byte 1/tag 3	71H
SN Byte 2/ tag 3	72H
SN Byte 3/tag 3	03H
SN Byte 4/tag 3	00H
SN Byte 5/tag 3	00H
SN Byte 6/tag 3	01H
SN Byte 7/tag 3	04H
SN Byte 8/tag 3	E0H
Data Byte 1/ tag 3	53H
Data Byte 2/tag 3	48H
Data Byte 3/tag 3	49H
Data Byte 4/tag 3	54H
Checksum	6FH
Terminator	03H
Header	0202H
Response Size	0003H
Command Echo	FFH
Number of tags	03H

Command from host	
Field	**Content**
Header	0202H

Response from controller	
Field	**Content**
Header/tag 1	0202H
Status Byte	08H
Checksum	F2H
Terminators <ETX>	03H

****

ABxF Command 83H: Start/Stop Continuous SN Read All

DESCRIPTION

Start and Stops continuous read all mode for multiple tags. It reads the Serial Number and tag data. While in this mode, any other command can be issued and it will be handled properly. After processing the interrupting command, the controller will resume the continuous read. Continuous SN Read All is started or stopped through a unique parameter, Start Continuous Read. This parameter is set to 01H to initiate the Continuous Read, and set to zero to stop it. The command uses the parameter Tag Repeat Count to limit multiple reads of the same tag. A tag is not read a second time until this specified number of tags have been read since it was last read. Allowed values are from 0 to 255 (FFH), where 0 means the tag can be reread anytime. When Start/Stop Continuous SN Read All is interrupted with another command, the repeat count is saved during execution of the other commands, allowing any tag to be addressed by the interrupting command. Upon completion of the interrupting command, the value of the Tag Repeat Count is restored. The Reader/Writer will respond after receipt of the command with an acknowledge packet. This will be followed by data packets for each tag read

	Content
Header	0202H
Command Size	Command length in Bytes excluding the header, command size, checksum and terminator Bytes.
Command	83H
Family Code	Tag Family Code - 00H = all tags
Reserved	Ignored Byte, set to 00H
Start Address	Two-Byte address for the first Byte to be read.
Number of Bytes	Number of bytes to be read from the tag.
Tag Repeat Count	Number of tags to be read before data will bereturned twice from any individual tag
Start Continuous	01H to start Continuous Read
Read	00H to stop
Checksum	Optional Checksum
Message Terminator	03H

Example

Starts continuous read of three Bytes starting at address two, Repeat count of four, with the family code set to 00H, so that all tags in the field will respond.

Field	**Content**
Header	0202H
Command Size	0008H
Command Code	83H
Family Code	00H
Reserved	00H
Start Address	0002H
Number of Bytes	0003H
Repeat Count	04H
Start	01H
Checksum	6AH
Terminator	03H

	Content
Header	0202H
Response Size	0001H
Command Echo	83H
Checksum	7BH
Terminator	03H

After the controller sends the acknowledgement, it will send the data read from the tags in the field

Response from controller

	Content
Header	0202H
Response Size	000BH
Command Echo/Tag 1	83H
Data from address 0002H	AAH
Data from address 0003H	21H
Data from address 0004H	44H
Checksum	62H
Terminator	03H

	0202H
Response Size	000BH
Command Echo/Tag 2	83H
Data from address 0002H	AAH
Data from address 0003H	21H
Data from address 0004H	44H
Checksum	62H
Terminator	03H

Header	**0202H**
Response Size	000BH
Command Echo/Tag 3	83H
Data fromaddress 0002H	AAH
Data from address 0003H	21H
Data from address 0004H	44H
Checksum	62H
Terminator	03H

****

ABxF Command 84H: Fill All

DESCRIPTION

Fill all RFID tags-in-field or all tags in the same family, with a one-Byte value over multiple contiguous addresses.

DISCUSSION

This command is commonly used to clear an RFID tag's memory. It writes a one Byte value repetitively across a specified range of tag addresses. All tags present in the antenna field with the specified Family Code will be affected by this command. The Fill All function requires one Data Value Byte, a Start Address, and a Number of Bytes to fill. It will fill the tag with the data value Byte, starting at the specified start address for the specified number of consecutive Bytes. The Fill Length must be set to a non-zero value. The timeout value is given in 1 msec increments and can have a value of 1EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return a syntax error.

	Content
Header	0202H
Packet Size	Packet length in Bytes excluding the header, packet size, checksum and terminator Bytes. 0008H for this command
Command	84H
Family Code	Tag Family Code - 00H = all tags
Reserved	Ignored, set to 00H
Start Address	2-Byte value for the starting tag address
Fill Length	2-Byte value for the length of the fill in number of Bytes (1-48)
Timeout	2-Byte value for timeout in 1 ms units. (1EH - FFFEH)
Data value Byte	1 Byte of fill
Checksum	Optional Checksum
Terminator	03H

Example

Writes 41H -- the ASCII character "A" -- to all tags with Family Code 03H, starting at address 0005H for the following next consecutive 40 Bytes. A timeout of 2 seconds (07D0H = 2000 x 1 msec increments) is set for the completion of the configuration. Four tags are found and filled successfully.

Field	**Content**
Header	0202H
Command Size	000AH
Command Code	84H
Family Code	03H
Reserved	00H
Start address	0005H
Number of Bytes	0028H
Timeout,2 seconds	07D0H
Data Value Byte	41H
Checksum	27H
Terminator	03H

	Content
Header	0202H
Response Size	0003H
Command Echo	84H
Numbers of tags	04H
Status Byte	08H
Checksum	6CH
Terminator	03H

****

ABxF Command 85H: Block Read All

DESCRIPTION

Read a block of data from all RFID tags-in-field or all those with the specified Family Code.

DISCUSSION

This command is used to read segments of data from contiguous areas of tag memory. The timeout value is given in 1 msec increments and can have a value of 1EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return a syntax error. The Block Read All consists of a Family Code, the reserved byte, a start address and length, followed by a timeout value and a message terminator as shown below. If the read length exceeds the last tag address, the controller will return an syntax error message (error code 21H). A special termination packet is sent when the timeout expires. A response is returned after successful completion of the operation or when the timeout expires.

	Content
Header	0202H
Command Size	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes.
Command	85H
Family Code	Tag Family Code - 00H = all tags
Reserved	Ignored, set to 00H
Start Address	2-Byte value for the starting tag address
Number of Bytes	2-Byte value for the number of bytes to be read
Timeout	2-Byte value for timeout in 1 ms units. (1EH - FFFEH)
Checksum	Optional Checksum
Terminator	03H

Example

Reads 4 Bytes of data from tags with Family ID AAH, starting at address 0001H. A timeout of 2 seconds (07D0H = 2000 x 1 msec increments) is set for the completion of the Block Read All. The Family Code is set to zero so all tags will be read.

Field	**Content**
Header	0202H
Command Size	0009H
Command Code	85H
Family Code	00H
Reserved	00H
Start address	0001H
Length of read	0002H
Timeout, 2 seconds	07D0H
Checksum	EBH
Terminator	03H

	Content
Header	0202H
Response Size	0003H
Command Echo/Tag 1	85H
Data from address 0001H	05H
Data from address 0002H	AAH
Checksum	C8H
Terminator	03H
Header	0202H
Response Size	0003H
Command Echo/Tag 2	85H
Data from address 0001H	05H
Data from address 0002H	AAH
Checksum	C8H
Terminator	03H
Header	0202H
Response Size	0003H
Command Status	FFH
Number of tags	02H
Status Byte	08H
Checksum	F3H
Terminator	03H

****

ABxF Command 86H: Block Write All

DESCRIPTION

Write a block of data to an RFID tag.

DISCUSSION

The Block Write All command is used to write segments of data to contiguous areas of tag memory. The timeout value is given in 1 msec increments and can have a value of 1EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return a syntax error. The Block Write All consists of a Family ID, the Reserved Byte, a Start address, and the Number of Bytes followed by the Bytes to be written to the RFID tag. If the write range exceeds the last tag address, the controller will return a syntax error message (error code 21H). The controller will return a response when the timeout expires.

	Content
Header	0202H
Command Size	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes. 0009H plus the number of data Bytes
Command	86H
Family ID	Tag Family ID - 00H = all tags
Reserved	Ignored, set to 00H
Start Address	2-Byte value for the starting tag address
Number of Bytes	2-Byte value for the number of bytes to be writen
Timeout	2-Byte value for timeout in 1 ms units. (1EH - FFFEH)
Data	Bytes of data to be written.
Checksum	Optional Checksum
Terminator	03H

Example

Writes 4 Bytes of data to the tag starting at address 0000H. A timeout of 2 seconds (07D0H = 2000 x 1 msec increments) is set for the completion of the Block Write All. Family ID is set to 00H so all tags-in-field will be written to. Five tags are written to.

Field	**Content**
Header	0202H
Command Size	000DH
Command Code	86H
Family ID	00H
Reserved	00H
Start address	0000H
Number of Bytes	0004H
Timeout,2 seconds	07D0H
Data to write to address 0000H	52H
Data to write toaddress 0001H	46H
Data to write toaddress 0002H	49H
Data to write toaddress 0003H	44H
Checksum	68H
Terminator	03H

	Content
Header	0202H
Response Size	0003H
Command Echo	86H
Number of Tags	05H
Status Byte	08H
Checksum	69H
Terminator	03H

****

ABxF Command 87H: Read Tag SN All

DESCRIPTION

This command retrieves the 8-Byte tag serial number from all tags-in-field or those with the specified Family ID.

DISCUSSION

Each ISO-15693 compliant tag has an unique (over 280 trillion possibilities) serial number. This number can not be changed and is not part of the available read/write address space. A special termination packet is sent when the timeout expires. A response is returned when the timeout expires.

	Content
Header	0202H
Command Size	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes. 0005H for this command
Command	87H
Family ID	Tag Family ID - 00H = all tags
Reserved	Ignored, set to 00H
Timeout 2-Byte	timeout value in 1 ms increments (1EH - FFFEH)
Checksum	Optional Checksum
Terminator	03H

Example

This example will read the 8-Byte serial number from all tags permitted by the Family ID and Reserved . In this example, one tag responds and the serial number is 1E6E3DC200000000H

Field	**Content**
Header	0202H
Command Size	0005H
Command Code	87H
Family ID	00H
Reserved	01H
Timeout	07D0H
Checksum	9BH
Terminator	03H

	Content
Header	0202H
Response Size	0009H
Command Echo	87H
SN Byte 1	F2H
SN Byte 2	75H
SN Byte 3	03H
SN Byte 4	00H
SN Byte 5	00H
SN Byte 6	00H
SN Byte 7	04H
SN Byte 8	E0H
Checksum	E4H
Terminator	03H
Header	0202H
Response Size	0003H
Command Status	FFH
Number of Tags	01H
Status Byte	08H
Checksum	F4H
Terminator	03H

****

ABxF Command 88H: Tag Search All

DESCRIPTION

Searches for tags within range of the antenna.

DISCUSSION

This command will activate the controller to "look" for tags within range of the antenna. As soon as the controller finds a tag it will return a command echo to the host. The timeout value is given in 1 msec increments and can have a value of 1EH to FFFEH (65,534 ms). If no tag is present it will return an error message. See Section 6.6 for information on the error messages.

	Content
Header	0202H
Command Size	Packet length in Bytes excluding the header, command size, checksum and terminator Bytes. 0005H for this command
Command	88H
Family ID	Tag Family ID - 00H = all tags
Reserved	Ignored, set to 00H
Timeout	2-Byte value for the time in 1 ms units (1EH - FFFEH)
Checksum	Optional Checksum
Terminator	<ETX>

Example

Checks for the presence of an RFID tag within range of the antenna. Atimeout of 2 seconds (07D0H = 2000 x 1 msec increments) is set for thecompletion of the Tag Search All. A tag is found.

Field	**Content**
Header	0202H
Command Size	0005H
Command Code	88H
Family ID	00H
Reserved	00H
Timeout	07D0H
Checksum	99H
Terminator	03H

	Content
Header	0202H
Response Size	0003H
Command Echo	88H
Number of Tags	01H
Status Byte	00H
Checksum	73H
Terminators	03H

****

ABxF Command 8DH: Stop/Start Continuous Read All

DESCRIPTION

Starts and stops Continuous Read mode for multiple tags.

DISCUSSION

The Start/Stop Continuous Read All mode is initiated by issuing the command with the Number of Bytes set to any valid nonzero value. Stop the mode by sending the command with a read length of 0. While in thismode, any other command can be issued and it will be handled properly. After processing the new command, the controller will resume the Continuous Read All mode. The command has a parameter, Tag Repeat Count, that can prevent multiple reads of the same tag. A tag is not read a second time until a specified number of tags have been read since it was last read. Allowed values are from 0 up to 255 (FFH), where 0 means the tag can be re-read anytime. When Continuous Read All mode is interrupted with other commands, the tag delay count is stopped during execution of the other commands and then resumed. The controller will respond with an acknowledge packet followed by data packets for each tag read. CONFIG LED blinks after each packet transmission.

	Content
Header	0202H
Command Size	Command length in Bytes excluding the header command size, checksum and terminator Bytes.
Command	8DH
Family ID	Tag Family ID - 00H = all tags
Reserved	Ignored, set to 00H
Start Address	Tag address for the start of the read
Number of Bytes	0 = stop. Any valid nonzero value indicates the number of Bytes to be read
Tag Delay	Number of tags that must be read before the same tag will be read again (0-225)
Checksum	Optional Checksum
Message Terminator	03H

Example

Reads 4 Bytes of data from the tag starting at address 0001H. The Family ID Byte is set to zero so all tags will be read. The Tag Delay is set to 20 (14H). Two tags respond with read data.

Field	**Content**
Header	0202H
Command Size	0008H
Command Code	8DH
Family ID	00H
Reserved	00H
Start Address	0001H
Read Length	0004H
Tag Delay	14H
Checksum	50H
Terminator	03H

	Content
Heade	0202H
Response Size	0001H
Command Echo	8DH
Checksum	71H
Terminator	03H

After the controller sends the acknowledgment, it will send the read data from the 2 tags.

Field	**Content**
Header	0202H
Response Size	0005H
Command Echo/ Tag 1	8DH
Data from address 0001H	05H
Data from address 0002H	AAH
Data from address 0003H	21H
Data from address 0004H	44H

Field	Content
Checksum	59H
Terminator	03H
Header	0202H
Response Size	0003H
Command Echo/ Tag 2	85H
Data from	address 0001H
05H	Data from
address 0002H	AAH
Data from	address 0003H
21H	Data from
address 0004H	44H
Checksum	59H
Terminator	03H

****

ABxF Command 8EH: Memory Lock All

DESCRIPTION

This command locks contiguous blocks of read-write memory. Once Bytes are locked, they cannot be written to, nor can they be unlocked.

DISCUSSION

The memory can be locked only in contiguous blocks. The command passes one parameter for the first block and another parameter for the number of blocks to be locked. This is one of only two ABx commands which address the memory of the tag using these parameters. The block structure of the tags is governed by the ISO-15693 specification. ABx command 15H can be used to obtain the memory organization for a particular tag. Attempting to write to locked Bytes will return a Write Security Error.

The command is formatted as shown below.

	Content
Header	0202H
Command Size	Command length in Bytes excluding the header,command size, checksum and terminator Bytes. 07H for this command.
Command	8EH
Family Code	Tag Family ID - 00H = all tags
Reserved	Ignored, set to 00H
Timeout	Timeout value given in 1 ms units
Starting Block	First block of memory to be locked
Number of Blocks	The number of blocks whose memory will be locked
Checksum	Optional Checksum
Message Terminator	03H

Example

This example will lock blocks 0-3 on all tags-in-field with the FamilyCode of 02H. Two tags are found and locked.

Field	**Content**
Header	0202H
Command Size	0007H
Command Code	8EH
Family ID	02H
Reserved	00H
Timeout	07D0H
First block	00H
Number of blocks	03H
Checksum	8EH
Terminator	03H

	Content
Header	0202H
Response Size	0003H
Command Echo	8EH
Number of Tags	02H
Status Byte	08H
Checksum	64H
Terminator	03H

****

ABxF Command 8BH: Write Family Code All

DESCRIPTION

This command can be used to write the Family Code of all tags in the field. It can also be used to selectively rewrite the family codes of tags with Family Codes already set to a particular value.

The command is formatted as shown below.

	Content
Header	0202H
Command Size	Command length in Bytes excluding the header, command size, checksum and terminator Bytes. 06H for this command.
Command	8BH
Current Family Code	Tag Family ID - 00H = all tags used to select which tags are addressed
Reserved	Ignored, set to 00H
Timeout	Timeout value given in 1 ms units
New Family Code	The value which will be written to the Family Codes of all the addressed tags
Checksum	Optional Checksum
Message Terminator	03H

Example

This example will change the Family code depending on the current value of the family code. Two tags are found with Family Code 02H and they are changed to 03H.

Field	**Content**
Header	0202H
Command Size	0006H
Command Code	8BH
Family ID	02H
Reserved	00H
Timeout	07D0H
New Family Code	03H
Checksum	92H
Terminator	03H

	Content
Header	0202H
Command Size	0006H
Command Code	8BH
Family ID	02H
Reserved	00H
Timeout	07D0H
New Family Code	03H
Checksum	92H
Terminator	03H

****

ABxF Command 8CH: Lock Family Code All

DESCRIPTION

This command locks the family codes of tags in the field. Once locked, the Family Codes cannot be changed or unlocked.

DISCUSSION

The command must pass a family code to select the tags whose family code will be locked. The controller will return a response when the timeout period expires. The parameter of the response in the number of tags written.

The command is formatted as shown below.

	Content
Header	0202H
Command Size	Command length in Bytes excluding the header, command size, checksum and terminator Bytes. 07H for this command.
Command	8CH
Family Code	Tag Family ID - 00H = all tags
Reserved	Ignored, set to 00H
Timeout	Timeout value given in 1 ms units
Checksum	Optional Checksum
Message Terminator	03H

Example

This example will lock the Family Code on all tags with the Family Code of 02H. Two tags are found and locked.

Field	**Content**
Header	0202H
Command Size	0005H
Command Code	8CH
Family ID	02H
Reserved	00H
Timeout	07D0H
Checksum	95H
Terminator	03H

	Content
Header	0202H
Response Size	0003H
Command Echo	8CH
Number of Tags	02H
Status Byte	08H
Checksum	67H
Terminator	03H

****

ABxF Command 94H: SN Fill

DESCRIPTION

Fills only the RFID tag specified by serial number with a one byte value over multiple contiguous addresses.

DISCUSSION

This command is commonly used to clear an RFID tag's memory. It writes a one byte value repetitively across a specified range of tag addresses. Only the tag with the specified serial number will be affected by this command. The controller will return a response after the successful fill operation or when the timeout expires. The fill function requires a Serial Number, one data value byte, a starting address, and a fill length. It fills the specified tag with the data value byte, starting at the specified start address for the specified number of consecutive bytes. When Fill Length is set to 0, the controller will write fill data from the start address to the end of the tag's memory. The timeout value is given in 1 msec increments and can have a value of 1EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return a syntax error. It returns a response when done or when the timeout expires.

	Content
Header	0202H
Command Size	Command length in Bytes excluding the header, command size, checksum and terminator Bytes.
Command	94H
Family Code	Ignored, set to 00H
Reserved	Ignored, set to 00H
Start Address	2-Byte value for the starting tag address
Number of Bytes	2-Byte value for the length of the fill in number of Bytes
Timeout	Timeout value given in 1 ms units
Serial Number	Serial Number of the tag to be addressed
Data Value	Value to be written to all addressed bytes
Checksum	Optional Checksum
Message Terminator	03H

A response to a successful command will follow this form.

	Content
Header	0202H
Response size	01H
Command Echo	94H
Checksum	Optional Checksum
Message Terminator	FFFFH

****

ABxF Command 95H: SN Block Read

DESCRIPTION

Read a block of data from a specified RFID tag.

DISCUSSION

This command is used to read segments of data from contiguous areas of tag memory. It is capable of handling up to 1 kByte of data transferred to the host with one command. The timeout value is given in 1 msec increments and can have a value of 1EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return a syntax error. The SN Block Read consists of Family ID and Serial Number a start address and length, followed by a timeout value. The 8-byte serial number of the target tag is specified. If the read length exceeds the last tag address, the controller will return an error message in the status byte. It returns a response when done or when the timeout expires.

	Content
Header	0202H
Command Size	Command length in Bytes excluding the header, command size, checksum and terminator Bytes.
Command	95H
Family Code	Ignored, set to 00H
Reserved	Ignored, set to 00H
Start Address	2-Byte value for the starting tag address
Number of Bytes	2-Byte value for the length of the fill in number of Bytes
Timeout	Timeout value given in 1 ms units
Serial Number	Serial Number of the tag to be addressed
Checksum	Optional Checksum
Message Terminator	03H

A response to a successful command will follow this form.

	Content
Header	0202H
Response Size	0001H
Command Echo	95H
Checksum	Optional Checksum
Message Terminator	FFFFH

****

ABxF Command 96H: SN Block Write

DESCRIPTION

Write data to a single RFID tag specified by its serial number.

DISCUSSION

This command is used to write data to contiguous areas of tag memory. It is capable of transferring up to 1 kByte of data from the host to the controller in a single command. The timeout value is given in 1 msec increments and can have a value of 1EH to FFFEH (65,534 ms). When the timeout is set to 0, the controller will return a syntax error. The SN BlockWrite consists of a Family ID, Serial Number and start address followed by the data stream to be written to the RFID tag. If the write range exceeds the last tag address, the controller will return a syntax error, 21H. It returns a response when done or when the timeout expires.

	Content
Header	0202H
Command Size	Command length in Bytes excluding the header, command size, checksum and terminator Bytes.
Command	96H
Family Code	Ignored, set to 00H
Reserved	Ignored, set to 00H
Start Address	2-Byte value for the starting tag address
Number of Bytes	2-Byte value for the length of the fill in number of Bytes
Timeout	Timeout value given in 1 ms units
Serial Number	Serial Number of the tag to be addressed
Data	Data to be written to the tag
Checksum	Optional Checksum
Message Terminator	03H

A response to a successful command will follow this form.

	Content
Header	0202H
Response size	01H
Command Echo	96H
Checksum	Optional Checksum
Message Terminator	FFFFH

****

ABxF Command 10H: Set Output

DESCRIPTION

Set the states of the output lines and output LEDs "A" through "D."

DISCUSSION

This command will set the state of the digital output lines using a one-Byte parameter, Output Status. The least significant nibble of the Output Status Byte will determine the status of the outputs. The least significant bit of this nibble corresponds to Output A. The most significant bit of this nibble corresponds to Output D. The most significate nibble of the Output Status Byte is reserved and should beset to 0H. An output bit set to one(1) will turn the corresponding LED on and close the electronic switch of the output, allowing current to flow through the output circuit. An output bit reset to zero will turn the corresponding LED off and open the electronic switch in the output circuit. The following chart shows the hex values for all combinations of Output Status Bytes. To reset all output bits, issue the command with 00H for the Output Status Byte.

Output Status Byte	Remarks	LSB Bit 3 Output D	LSB Bit 2 Output C	LSB Bit 1 Output B	LSB Bit 0 Output A
00H	Reset A, B, C, D	0	0	0	0
01H	Set Output A - Reset B, C, D	0	0	0	1
02H	Set Output B - Reset A, C, D	0	0	1	0
03H	Set Output A, B - Reset C, D	0	0	1	1
04H	Set Output C - Reset A, B, D	0	1	0	0
05H	Set Output A, C - Reset B, D	0	1	0	1
06H	Set Output B, C - Reset A, D	0	1	1	0
07H	Set Output A, B, C - Reset D	0	1	1	1
08H	Set Output D - Reset A, B, C	1	0	0	0

Output Status Byte	Remarks	LSB Bit 3 Output D	LSB Bit 2 Output C	LSB Bit 1 Output B	LSB Bit 0 Output A
09H	Set Output A, D - Reset B, C	1	0	0	1
0AH	Set Output B, D - Reset A, C	1	0	1	0
0BH	Set Output A, B, D - Reset C	1	0	1	1
0CH	Set Output C, D - Reset A, B	1	1	0	0
0DH	Set Output A, C, D - Reset B	1	1	0	1
0EH	Set Output B, C, D - Reset A	1	1	1	0
0FH	Set Output A, B, C, D	1	1	1	1

Field	Content
Header	0202H
Command Size	Packet length in bytes excluding the header, command size, checksum and terminator bytes.
Command	10H
Output Status Byte	1 byte representing the desired output settings in bits 0-3
Checksum	Optional Checksum
Terminator	03H

Example

The following example sets Output B only and resets A, C, and D

.

Field	**Content**
Header	0202H
Command Size	0002H
Command Code	10H
Output Value Byte	02H
Checksum	EBH
Terminator	03H

	Content
Header	0202H
Response Size	0001H
Command Echo	10H
Checksum	EEH
Terminator	03H

****

ABxF Command 11H: Input Status

DESCRIPTION

Returns the status of User Inputs.

DISCUSSION

This command will interrogate the state of the User Input lines and return a one-Byte parameter, Input Status. The least significant nibble of the Input Status Byte is determined by the status of the outputs. The least significant bit of this nibble corresponds to Input A. The most significant bit of this nibble corresponds to Input D. The most significate nibble of the Input Status Byte will be set to 0H. A closed circuit in one(1) of the User Inputs will set the corresponding User Input bit to one, turning the corresponding LED on. An open circuit in one of the User Inputs will set the corresponding User Input bit to zero, turning off the corresponding LED. The following chart shows all of the valid hex values for the Input Status bytes.

Input Status Byte	Remarks	LSB Bit 3 Input D	LSB Bit 2 Input C	LSB Bit 1 Input B	LSB Bit 0 Input A
00H	Inputs A, B, C, D, open	0	0	0	0
01H	Input A, closed - B, C, D, open	0	0	0	1
02H	Input B, closed - A, C, D, open	0	0	1	0
03H	Input A, B, closed - C, D, open	0	0	1	1
04H	Input C, closed - A, B, D, open	0	1	0	0
05H	Input A, C, closed - B, D, open	0	1	0	1
06H	Input B, C, closed - A, D, open	0	1	1	0
07H	Input A, B, C, closed - D, open	0	1	1	1
08H	Input D, closed - A, B, C, open	1	0	0	0

Input Status Byte	Remarks	LSB Bit 3 Input D	LSB Bit 2 Input C	LSB Bit 1 Input B	LSB Bit 0 Input A
09H	Input A, D, closed - B, C, open	1	0	0	1
0AH	Input B, D, closed - A, C, open	1	0	1	0
0BH	Input A, B, D, closed - C, open	1	0	1	1
0CH	Input C, D, closed - A, B, open	1	1	0	0
0DH	Input A, C, D, closed - B, open	1	1	0	1
0EH	Input B, C, D, closed - A, open	1	1	1	0
0FH	Input A, B, C, D, closed	1	1	1	1

Field	Content
Header	0202H
Command Size	Packet length in bytes excluding the header,command size, checksum and terminator bytes.
Command	11H
Checksum	Optional Checksum
Terminator	03H

Example

The following example shows only Input B is ON and A, C, and D are OFF.

	Content
Header	0202H
Command Size	0001H
Command Code	11H
Checksum	EDH
Terminator	03H

	Content
	0202H
Response Size	0002H
Command Echo	11H
Input Status Byte	02H
Checksum	EAH
Terminator	03H

6.5 ABx ASCII Protocol

The ABx ASCII Protocol is based on the ABx Fast protocol. It uses the same headers and terminator (already ASCII characters) and converts the hex value of command and data bytes to printable ASCII (2 digit Hexadecimal notation). In another words, the hex values given in an ABx Fast command are transmitted as separate ASCII characters. Since it is an ASCII protocol, the Xon/Xoff handshake can be used.

Command Packet Structure

The command protocol is based on the following minimal packet structure. The data field and the checksum may not be present depending on the command type and your checksum setting.

6.6 ABx ASCII Protocol Command Structure

	Number of ASCII Characters	Content
Header	2	<STX><STX> (02H, 02H)
Command Size	4	Packet length in bytes excluding the header, Command size, checksum and terminator bytes.
Command	2	Command Code
(Data)	variable	command data/parameters
Checksum	2	Optional Checksum
Terminator	1	<ETX>(03H)

Following a successful operation, the controller will respond with the following. The data field and the checksum may not be present depending on the command and your checksum setting. If a checksum is enabled in the Configuration Menu, then it is always present for every command.

6.7 ABx ASCII Protocol Response Structure

	Number of ASCII Characters	Content
Header	2	<STX><STX> (02H, 02H)
response Size	4	Packet length in bytes excluding the header, Command size, checksum and terminator bytes.
Command	2	Command Echo
(Data)	variable	response data
Checksum	2	Optional Checksum
Terminator	1	<ETX> (03H)

If the controller encounters a fault it will respond with the following:

6.8 ABx ASCII Protocol Response Structure

Field	Number of ASCII Characters	Content
Header	2	<STX><STX> (02H, 02H)
response Size	4	Packet length in bytes excluding the header, Command size, checksum and terminator bytes.
Error Flag	2	FFH
Error Code	2	Hex error code, see Table 11 for details
Checksum	2	Optional checksum
Terminator	1	<ETX>(03H)

Most RF operations will also require additional parameters and data that will be included in the command stream between the command code or echo and the terminator. The Header and Terminator are always STX and ETX respectively. Any other field value is in ASCII hex notation. Allowed values: '0'-'9', 'A'-'F'. Example: the value ABH (decimal 171) in ASCII protocol is transmitted as a 2-character string «AB», i.e. the 2 bytes: 41H 42H (ASCII values for 'A' and 'B'). The hex

value of the hex digits given in ASCII are: '0'-'9' = 30H - 39H, 'A'-'F' = 41H-46H. The sequence for each command is given with the response format in the preceding section. Referring to the ABx Fast command you can structure the ABx ASCII commands by using ASCII values for each digit of the hex values, excluding the header and terminator that are already ASCII characters.

Command/Response Size

The ABx ASCII requires the length of the packet be included in the command. All parameters and data between the Command Size and the terminator or checksum byte must be accounted for in the packet size word. This includes all command codes and parameters such as field definition for Block Read/Writes. The packet size remains the same with, or without the checksum.

Checksum

The optional checksum must be enabled from the operating mode menu to be available. The checksum is calculated by adding all the byte values (not the ASCII translation values) in the packet (less the values in the header, checksum if present, and terminator), discarding byte overflow and subtracting the byte sum from FFH.

Example ASCII Command

Fill Tag

This command fills the specified number of cells from the specified start address with the specified value. Block size = 0 means filling to the end of the memory. The command will take the same form as the ABx Fast command.

Field	Content
Header	<STX><STX>
Command Size	Packet length in bytes excluding the header, command size, checksum and terminator bytes. Given as four ASCII character value. 0008H for this command
Command	<30H><34H> (04)
Start Address	4 ASCII character value for the starting tag address
Fill Length	4 ASCII character value for the length of the fill in number of bytes
Timeout	4 ASCII character value for timeout in 1 ms units. (1EH - FFFEH)

Field	Content
Data value byte	2 ASCII character value for 1 byte of fill
Checksum	2 ASCII character value for Optional Checksum
Terminator	<ETX>

The ASCII character string for a fill of 32 bytes, from address 0 with 55H value, timeout 5 sec., follows on the next page.

Field	**ASCII Hex Value**	**ASCII String**
Header	02H	STX
<STX><STX>	02H	STX
Command Size	30H	0
	30H	0
	30H	0
	38H	8
Command	30H	0
	34H	4
Start Address	30H	0
	30H	0
	30H	0
	30H	0
Fill Length	30H	0
	30H	0
	32H	2
	80H	0
Timeout Value	31H	1
	33H	3
	38H	8
	38H	8
Data Byte Value	35H	5
	35H	5
Checksum	31H	1
	33H	3
Terminators <ETX>	03H	ETX

	ASCII Hex Value	**ASCII String**
Header	02H	STX
<STX><STX>	02H	STX
Response Size	30H	0
	30H	0
	31H	1
	30H	0
Command Echo	30H	0
	34H	4
Checksum	44H	D
	41H	A
Terminators <ETX>	03H	ETX

6.9 ABx Error Codes

Multi-tag Error Codes

The LRP2000 will return an error if it encounters a fault during operation. Table 8 lists the possible error codes in Hexadecimal format.

	Description
04H	Fill Operation has failed
05H	Block Read has failed
06H	Block Write has failed
07H	Block Write security error
08H	Search Tag Operation failed
21H	Input Command does not match pre-defined format (syntax error)

Additionally there are internal DSP errors, F1H through F5H, for use by Escort Memory Systems technical support.

ABx Standard

ABxS error codes are returned in the LSB of the second word passed to the host. The format of the error response is shown below.

MSB	LSB	Remarks
AAH	FFH	Command Error
00H	XXH	Error Code
FFH	FFH	Message Terminator

Example

A BlockWrite fail error message would appear as:

```
AAFF 0006 FFFFH.
```

ABx Fast

The format of the error response is shown in the table below.

	Bytes Contents
Header	02H
<STX><STX>	02H
Response Size	00H 02H
Error Flag	FFH
Error Code	XXH
Checksum	XXH
Terminators <ETX>	03H

Example

A BlockWrite fail error message would appear as:

```
0202 0002 FF06 F803H.
```

ABx ASCII

The format of the error response is shown below.

Field	Number of ASCII Characters	Contents
Header		
<STX><STX>	2	<STX> 02H
<STX> 02H		<STX> 02H
Response Size	4	Packet length in bytes excluding the header, response size,checksum and terminator bytes.
Error Flag	2	FFH
Error Code	2	XXH - see Table 7 for details
Checksum	2	XXH - optional checksum
Terminators <ETX>	1	<ETX> 03H

A BlockWrite fail error message would appear as an ASCII character string:

<STX><STX>0002FF06F8<ETX>.

In hexadecimal the commands appear as:

02H 02H 30H 30H 30H 32H 46H 46H 30H 36H 46H 38H 03H

Multi-tag Command Error Code

When the multi-tag commands encounter a fault condition they indicate the error in a STATUS byte returned in the response. If any of the flag bits of the Status byte are set, then an error has occurred during command execution. The format of the response is in all other ways, the same as a successful response.

The STATUS byte is defined as follows:

	Reserved
6	R/W Error
5	Reserved
4	Internal Error
3	Timeout
2	Verify Error
1	Write Security Error
0	Reserved

Some of the error conditions are the same as found in the single-tag commands, some are new and relate only to the anticollision. If any of the flag bits of the returned Status are set, then that condition occurred during the command execution. Multiple conditions can occur in the same command.

Antenna Failure	There is an error at the antenna
R/W error	Error during the tag memory access
Collision	Collisions detected: more than one tag in the field answered to the LRP2000 at the same time, meaning a higher

Antenna Failure	**There is an error at the antenna**
Anticollision	This bit is reserved
Internal Error	Internal error in low-level firmware (contact Escort Memory Systems technical assistance)
Timeout	Timeout expired
Verify Error	Set when re-read verification fails
Syntax Errors	Syntax errors (error code 21H) will be returned in the same format as described for the single-tag commands.