BALLUFF

sensors worldwide

BIS ABx fast protokoll

User's Guide

www.balluff.com

CONTENTS

REFERENCES

CONVENTIONS

This manual uses the following conventions:

“User” or “Operator” refers to anyone using the ABx Fast Protocol software to program an RFID device.

“Device” refers to the Balluff processor units.

“You” refers to the System Administrator or Technical Support person using this manual to install, mount, operate, maintain or troubleshoot an RFID device.

BIS M-41_ , BIS M-62_ and BIS U-62_ processor units are referred to as controllers, or just “the controller ”.

BIS M-410	correspond to the old name	C-0405 unit
BIS M-411	correspond to the old name	C-1007 unit
BIS M-620-068_	correspond to the old name	HF-CNTL-232_ unit
BIS M-620-067_	correspond to the old name	HF-CNTL-485_ unit
BIS M-622-068_	correspond to the old name	HF-CNTL-PBS_ unit
BIS M-623-071_	correspond to the old name	HF-CNTL-DNT_ unit
BIS M-626-069_	correspond to the old name	HF-CNTL-IND_ unit
BIS M-628-075_	correspond to the old name	HF-CNTL-PNT_ unit
BIS U-620-068_	correspond to the old name	UHF-CNTL-232_ unit
BIS U-620-067_	correspond to the old name	UHF-CNTL-485_ unit
BIS U-626-069_	correspond to the old name	UHF-CNTL-IND_ unit
BIS Z-GW-001-DNT	correspond to the old name	GWY-01-DNT-01
BIS Z-GW-001-IND	correspond to the old name	GWY-01-IND-01
BIS Z-GW-001-PBS	correspond to the old name	GWY-01-PBS-01
BIS Z-GW-001-RS232	correspond to the old name	GWY-01-232-01
BIS Z-GW-001-TCP	correspond to the old name	GWY-01-TCP-01
BIS Z-HB-004-IND	correspond to the old name	HUB-04-IND-01
BIS Z-HB-004-TCP	correspond to the old name	HUB-04-TCP-01

REFERENCE DOCUMENTATION

The documentation related to the Gateway, Hub, and the BIS M-41_ , BIS M-62_ and BIS U-62_ processor units management is available on the website:

www.balluff.com

SERVICES AND SUPPORT

Balluff provides several services as well as technical support through its website. Log on to **www.balluff.com** and click on the links indicated for further information including:

- **PRODUCTS**
  Search through the links to arrive at your product page which describes specific Info, Features, Applications, Models, Accessories, and Downloads including:
  - **Dashboard™**: a Windows-based utility program, which allows system testing, monitoring, and configuration using a PC. It provides Serial (RS232 or USB) and Ethernet interface configuration.
  - **C-Macro Builder™**: an easy to use GUI-driven utility for Windows. This software tool allows users with minimal programming experience to “build” their own macro programs (which are stored internally on and executed directly by RFID Processors).

1 ABX FAST COMMAND PROTOCOL

1.1 COMMAND PROTOCOL MATRIX

Balluff RFID products support three basic command protocols: *CBx, ABx Fast* and *ABx Standard*. To determine which command protocol to utilize, please refer to the table below, which lists the different RFID devices and indicates the command protocol supported by each.

Product	CBx	ABx Fast	ABx Standard
BIS M-410-068-001_		X	X
BIS M-411-068-001_		X	X
BIS M-620-068-A01-00_		X	X
BIS U-620-068		X	X
BIS M-622/623/626_	X		
BIS U-626-069_	X		
BIS Z-GW-001_ all models	X		
BIS Z-HB-001_ all models	X		

Table 1-1: Command Protocol Matrix

NOTE

RS485-based Processor units are used in conjunction with Subnet16™ Gateway and Subnet16™ Hub interface modules, which use the CBx Command Protocol.

1.2 ABX FAST COMMAND PROTOCOL OVERVIEW

In order to execute RFID commands properly, the RFID device and host computer must be able to communicate using the same language. The language that is used to communicate is referred to as the *Command Protocol*. **The primary command protocol used by serial connection-based RFID devices is called "A*Bx Fast*"**.

The *ABx Fast Command Protocol* is commonly used by serial connection-based Processor units for point-to-point data transmission.

ABx Fast is based on a single-byte oriented packet structure that permits the rapid execution of RFID commands while requiring the transfer of a minimal number of bytes.

The protocol also supports the inclusion of an optional *Checksum* byte. When increased data integrity is required, the checksum should be utilized. See "Checksum" in par. 1.3.1 for more details on using the checksum parameter.

To issue an RFID command from the host, a packet of data, called the "*Command Packet,*" is sent to the RFID controller. The command packet contains information that instructs the controller to perform a certain task.

The controller automatically parses an incoming data packet, searching for a specific pair of start characters, known as the "*Command Header.*" In ABx Fast, the Command Header characters are *0x02, 0x02*. When a Command Header is recognized, the controller then

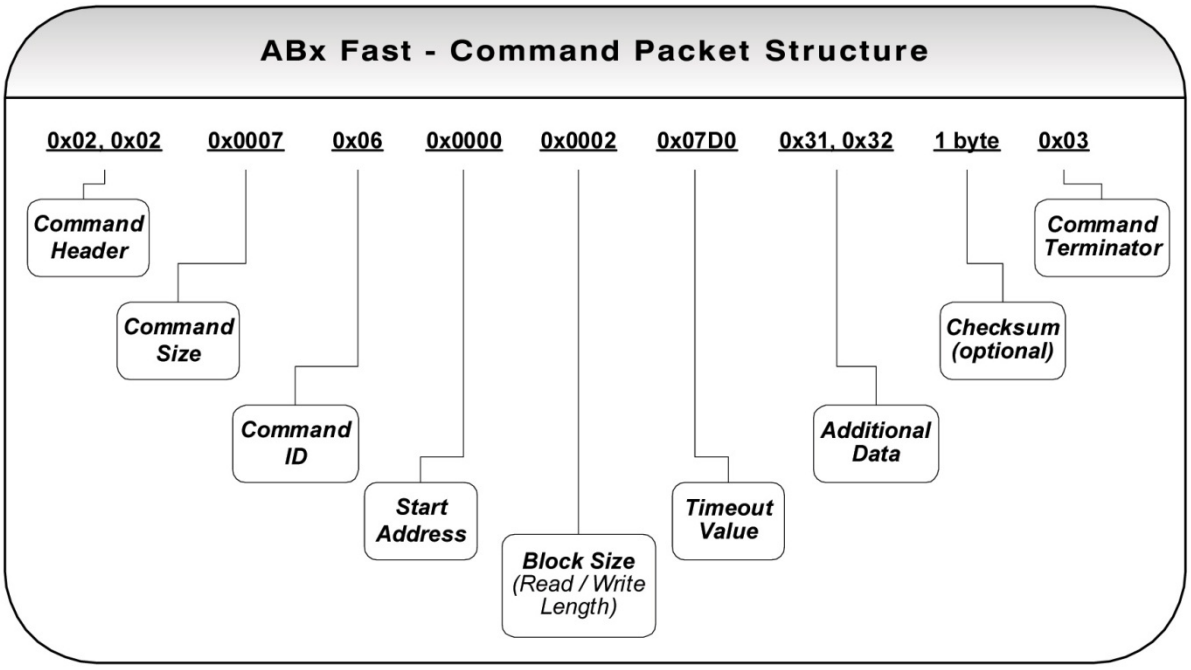
checks for proper formatting and the presence of a single-byte "*Command Terminator*." In ABx Fast, the Command Terminator byte is *0x03*.

Having identified a valid command, the controller will attempt to execute the instructions, after which it will generate a host-bound response message containing *EITHER* the results of the attempted command or an error code if the operation failed.

1.3 ABX FAST COMMAND PROTOCOL PACKET STRUCTURES

1.3.1 ABx Fast Command Packet Structure

The packet structure of all ABx Fast RFID commands contains certain basic elements, including *Command Header*, *Command Size*, *Command ID* and *Command Terminator*. Packet element and parameter availability depends on the command being performed.
Below is the structure of a standard ABx Fast command packet.



Command Packet Element	Content	Size
Command Header: *0x02, 0x02* These are the first two bytes of an ABx Fast command.	0x02, 0x02	2 bytes
Command Size: This 2-byte integer indicates the number of bytes in the command packet (excluding *Header, Command Size, Checksum and Terminator*).	0x0007 + (number of any additional data bytes)	2-byte integer
Command ID: This single-byte value indicates the RFID command to perform	**0x06** (Write Data)	1 byte
Start Address: This 2-byte integer indicates the location of tag memory where a read or write operation will begin	0x0000	2-byte integer
Block Size: This 2-byte integer indicates the number of bytes that are to be read from or written to a tag during the operation	0x0001	2-byte integer

Command Packet Element	Content	Size
Timeout Value: This 2-byte integer represents the length of time allowed for the completion of the command. Measured in one-millisecond increments, the *Timeout Value* can have a value of 0x0001 to 0xFFFE (*1 - 65,534 milliseconds*).	0x07D0 (2 seconds)	2-byte integer
Additional Data: This parameter uses one byte to hold a single character for fill operations and supports the use of multiple bytes when several characters are needed for write commands (when applicable).	0x00	1 or more bytes
Checksum: This optional parameter holds a single-byte checksum (only applicable when using *ABx Fast with Checksum*).	optional	1 byte (when applicable)
Command Terminator: *0x03* The single-byte command packet terminator is always *0x03* for ABx Fast.	0x03	1 byte

Table 1-2: ABx Fast Command Packet Structure

Command Size

The ABx Fast protocol requires that the byte count, known as the *Command Size*, be specified as a two-byte integer within each command packet. To calculate the Command Size, add the total number of bytes within the command packet while excluding the two byte *Command Header*, the two byte *Command Size*, the one byte *Checksum* (if present) and the one byte *Command Terminator* (see example below).

Command Size = number of bytes in these fields

Command Packet Element	# of Bytes	Included in Command Size?
Command Header	2	No
Command Size	2	No
Command ID	1	Yes
Start Address	2	Yes
Read/Block Size	2	Yes
Timeout Value	2	Yes
Additional Data Bytes	1	Yes (if present)
Checksum	1	No
Command Terminator	1	No

Table 1-3: ABx Fast Command Size Parameter

In the above command packet, there are eight bytes of data (located between the *Command Size* parameter and the *Checksum* parameter) that are included in the *Command Size*. Therefore, the *Command Size* for this example is *0x0008*.

Command ID

The one-byte *Command ID* parameter identifies the Hex value of the RFID command to perform (see chapter 2).

Start Address

The *Start Address* parameter holds a two-byte integer representing the tag memory address location where a read or write operation is to begin.

Block Size (Read/Write Length)

The two-byte *Block Size* parameter (which is also sometimes called the *Read / Write Length* parameter) indicates the number of bytes that are to be read from or written to the RFID tag.

Timeout Value

Most ABx Fast commands include a two-byte *Timeout Value*, which is used to limit the length of time that the controller will attempt to complete the specified operation.

The *Timeout Value* is measured in 1-millisecond increments and has a maximum supported value of *0xFFFE* or 65,534 milliseconds (which is slightly longer than one minute).
Setting a long *Timeout Value* does not necessarily mean that a command will take any longer to execute. This value only represents the period of time for which the controller will attempt execution of the command.

NOTE

During write commands, the tag must remain within the antenna's RF field until the write operation completes successfully, or until the Timeout Value has expired.
If a write operation is not completed before the tag leaves the controller's RF field, data may be incompletely written.

Checksum

The ABx Fast Command Protocol supports the inclusion of an additional *Checksum* byte that is used to verify the integrity of data being transmitted between host and controller. This option is disabled by default but can be enabled through the *Set Controller Configuration* command (0x36).

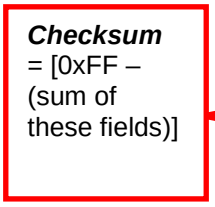
The *Checksum* is calculated by adding together (summing) the byte values in the command packet (less the *Command Header*, *Checksum* and *Command Terminator* parameters), and then subtracting the total byte sum from 0xFF.

Therefore, when the byte values of each parameter (from *Command Size* to *Checksum*) are added together, the byte value sum will equal 0xFF.

Fo sums exceeding 0xFF, only the least significant byte of the sum is used in the calculation.

Checksum Example

The following example depicts *Command 0x05* (*Read Data*) when using a Checksum.



Command Element	Contents	Used in Checksum
Header	0x02, 0x02	n/a
Command Size	0x0007	0x00, 0x07
Command ID	0x05	0x05
Start Address	0x0001	0x00, 0x01
Block Size	0x0004	0x00, 0x04
Timeout Value	0x07D0	0x07, 0xD0
Checksum	**0x17**	n/a
Terminator	0x03	n/a

Table 1-4: ABx Fast - Checksum Example

Add the byte values from the *Command Size, Command ID, Start Address, Block Size* and *Timeout Value* parameters together and subtract from 0xFF. The resulting value will be the Checksum.

[0x07 + 0x05 + 0x01 + 0x04 + 0x07 + 0xD0] = 0xE8

The checksum equation is: [0xFF – 0xE8] = **0x17**

1.3.2 ABx Fast Response Packet Structure

After executing a command, the controller, in most cases, will generate a host-bound response message. The response message will contain EITHER the results of the attempted command or an error code indicating the reason the operation could not be completed successfully. ABx Fast responses contain a *Response Header*, *Response Size*, *Command Echo*, one or more *Response Values* / *Retrieved Data* (when applicable), optional *Checksum* and a *Response Terminator*. Below is the structure of a standard ABx Fast response packet.

Response Packet Element	Content	Size
Response Header: *0x02, 0x02* These are the first two bytes of an ABx Fast response packet.	0x02, 0x02	2 bytes
Response Size: This 2-byte integer indicates the total number of bytes in the response packet (excluding *Response Header, Response Size, Checksum and Terminator*).	0x0001 + (number of retrieved data bytes)	2-byte integer
Command Echo: This single-byte value reiterates the Hex value of the command for which the response packet was generated.	**0x06** (Write Data)	1 byte
Response Values / Rerieved Data: This parameter is used to hold one or more bytes of the data that was requested by the command (when applicable).	Data	1 or more bytes (when applicable)
Checksum: This optional parameter holds a single-byte checksum (only applicable when using *ABx Fast with Checksum*).	optional	1 byte (when applicable)
Response Terminator: *0x03* The single-byte response packet terminator is always *0x03* for ABx Fast.	0x03	1 byte

Table 1-5: ABx Fast Response Packet Structure

1.3.3 ABx Fast Multi-Tag Command Packet Structure

ABx Fast Multi-tag Commands instruct a specified controller to read from or write to several tags at once when multiple tags are simultaneously within RF range. It is also possible to single-out and read from or write to one tag (identified by its unique tag ID number) when multiple tags are present in the RF field simultaneously.

NOTE

Multi-tag commands only support ISO 15693 compliant RFID tags.

Below is the structure of a basic ABx Fast Multi-tag command packet.

Command Packet Element	Content	Size
Command Header: *0x02, 0x02* These are the first two bytes of an ABx Fast command.	0x02, 0x02	2 bytes
Command Size: This 2-byte integer indicates the number of bytes in the command packet (excluding *Header, Command Size, Checksum and Terminator*).	0x0007 + (number of any additional data bytes)	2-byte integer
Command ID: This single-byte value indicates the RFID command to perform	**0x85** (Multi-tag Block Read All)	1 byte
AFI: Single-byte value (0x00-0xFF) specifies a subset of tags. 0x00 = all tags in RF range will respond to the command (*see description in par. below*).	0x00	1 byte
Anti-Collision Mode: Single-byte value allows the user to enable the use of 16 time slots for retrieving data where 0x01 = Multi-Slot, 0x00 = Single-Slot (*see description in par. below*).	0x01	1 byte
Tag Limit: Single-byte value (0x00 - 0x64) for the maximum # of tags expected in RF range, up to 100 (*see description in par. below*).	0x64	1 byte
Start Address: This 2-byte integer indicates the location of tag memory where a read or write operation will begin	0x0000	2-byte integer
Block Size: This 2-byte integer indicates the number of bytes that are to be read from or written to a tag during the operation	0x0001	2-byte integer
Timeout Value: This 2-byte integer represents the length of time allowed for the completion of the command. Measured in one-millisecond increments, the *Timeout Value* can have a value of 0x0001 to 0xFFFE (*1 - 65,534 milliseconds*).	0x07D0 (2 seconds)	2-byte integer
Additional Data: This parameter uses one byte to hold a single character for fill operations and supports the use of multiple bytes when several characters are needed for write commands (when applicable).	0x00	1 or more bytes
Checksum: This optional parameter holds a single-byte checksum (only applicable when using *ABx Fast with Checksum*).	optional	1 byte (when applicable)
Command Terminator: *0x03* The single-byte command packet terminator is always *0x03* for ABx Fast.	0x03	1 byte

Table 1-6: ABx Fast Multi-Tag Command Packet Structure

AFI (Application Family Identifier)

The *AFI* parameter is a one-byte value (*0x00 – 0xFF*) that can be used in multi-tag commands to specify a subset of tags when many are identified simultaneously in RF range. The parameter allows the user to filter tags based on a pre-written value stored at a special location on the tag.

For example, if the AFI value is set to one (*0x01*), only those tags with the pre-written AFI value *0x01* will respond to the given command. When an AFI value of zero (*0x00*) is entered for this parameter, all tag families within RF range will respond to the command.

Anti-Collision Mode

Tag collisions in RFID applications occur when numerous passive RFID tags become simultaneously activated (by the RFID controller) and thus reflect their respective signals back to the reader at the same time, such that the controller cannot differentiate between tags.

Balluff Processor units make use of anti-collision algorithms to enable a single reader/antenna to read more than one tag in the reader's field.
The *Anti-collision Mode* parameter controls the tag-reading algorithm used to achieve the fastest reading speed for the number of tags expected in RF range at any given moment. This parameter helps the reader/antenna avoid data collisions when simultaneously reading multiple tags.

The choices for this parameter are one (0x01) for Multi-Slot and zero (0x00) for Single-Slot.

- *ONE*: Setting this parameter to one (0x01), implements a multi-slot system of 16 time slots. To avoid data collisions when the controller encounters multiple tags simultaneously, data requested from each tag is transferred to the controller only during the time slot that matches a specific pattern in the tag ID number.
- *ZERO*: Setting this parameter to zero (0x00) utilizes a single time slot under which the requested data from all tags is transferred to the controller as soon as it becomes available. This setting can result in faster tag read performance when only a few tags are expected in the RF field.

The Anti-Collision Mode parameter immediately follows the "*AFI*" parameter in the multi-tag command packet string.

Tag Limit

The *Tag Limit* parameter holds a one-byte value that indicates the maximum number of tags expected simultaneously in RF range for the given command operation. This parameter allows users to limit the number of attempted read/write operations the controller will make per execution (users do not have to wait for the Timeout to expire).

The *Tag Limit* value should be set in relation to the maximum number of tags that could possibly be present in the reading field at any one time. Setting a high value increases the number of tags that are expected in the antenna's RF field. Setting a low value can speed up multi-tag operations when only a small number of tags could be present at any given moment.

Setting the proper value is therefore a tradeoff between the number of expected tags in the reading field, and the time required to read/write to them. The permitted values range from one to 100 (0x01 – 0x64).

The *Tag Limit* parameter resides directly after the "*Anti-collision Mode*" parameter in the command string (when applicable).

Timeout Value

Multi-tag commands also contain a two-byte *Timeout Value* parameter that is used to limit the length of time for which the controller will attempt to complete a given operation.
It is important to set a realistic *Timeout Value* that permits enough time for the controller to read/write to all tags specified in the command. Processing multiple-tag operations requires a longer time period than does the execution of single-tag commands.

The value is expressed in one-millisecond increments, with a maximum value of 0xFFFE (65,534 milliseconds) or approximately 60 seconds. It is recommended that users allow at least 100 ms per tag for multi-tag read operations and 150 ms per tag for multi-tag writes.
Using a *Timeout Value* that is too short may cause the controller to inadvertently "time out" before the data has been successfully read from or written to all tags in RF range. For time critical applications, the optimal *Timeout Value* should be obtained through rigorous performance testing.

Tag ID / Serial Number

There are two multi-tag commands that allow the user to retrieve data from or write data to a single tag (specified by its tag ID number) when numerous tags are concurrently present in the RF field (A*Bx Fast Command 0x95 - Multi-Tag Block Read by ID* and *ABx Fast Command 0x96 – Multi-Tag Block Write by ID*). The tag ID number is a unique, read-only, 64-bit (eight-byte) number stored in tag memory. Targeted tags can be recognized through a previously issued *Multi-Tag Get Inventory (0x87)* command.

1.3.4 ABx Fast Multi-Tag Response Packet Structures

When executing multi-tag commands designed to retrieve information from several tags at once (for example *ABx Fast Command 0x82: Multi-Tag Read ID and Data All)*, the RFID controller will generate separate host-bound response packets for each tag that has been read. Below is the structure of a basic ABx Fast multi-tag response packet generated by the controller.

ABx Fast Multi-tag Response Packet Structure (One Packet for Each Tag Read)

Response Packet Element	Content	Size
Response Header: *0x02, 0x02* These are the first two bytes of an ABx Fast response packet.	0x02, 0x02	2 bytes
Response Size: This 2-byte integer indicates the total number of bytes in the response packet (excluding *Response Header, Response Size, Checksum and Terminator*).	0x0009 + (number of retrieved data bytes)	2-byte integer
Command Echo: This single-byte value reiterates the Hex value of the command for which the response packet was generated.	**0x82** (*Multi-Tag Read ID and Data All*)	1 byte
Tag ID: 8 bytes (when applicable).	8-byte Tag ID	8 bytes
Rerieved Data	N bytes	N bytes
Checksum: This optional parameter holds a single-byte checksum (only applicable when using *ABx Fast with Checksum*).	optional	1 byte (when applicable)
Response Terminator: *0x03* The single-byte response packet terminator is always *0x03* for ABx Fast.	0x03	1 byte

Table 1-7: ABx Fast Multi-Tag Response Packet Structure

ABx Fast Multi-Tag Response Final Termination Packet Structure

After the RFID controller has issued response packets for each tag identified and/or read, a final termination packet is generated. Below is the structure of a standard ABx Fast multi-tag response final termination packet generated by the controller.

Response Packet Element	Content	Size
Response Header: *0x02, 0x02* These are the first two bytes of an ABx Fast response packet.	0x02, 0x02	2 bytes
Response Size: This 2-byte integer indicates the total number of bytes in the response packet (excluding *Response Header, Response Size, Checksum and Terminator*).	0x0003	2-byte integer
Final Termination Packet Identifier: 0xFF indicates that this packet is the final termination packet.	**0xFF**	1 byte
Number of Tags Read/Written: single-byte value that indicates the number of tags read from or written to during the operation.	N tags	1 byte
Status: (0x00 = operation completed successfully, 0x07 = Read Tag ID failed / Tag Not Found).	0x00	1 byte
Checksum: This optional parameter holds a single-byte checksum (only applicable when using *ABx Fast with Checksum*).	optional	1 byte (when applicable)
Response Terminator: *0x03* The single-byte response packet terminator is always *0x03* for ABx Fast.	0x03	1 byte

Table 1-8: ABx Fast Multi-Tag Response Final Termination Packet Structure

1.4 ERROR MESSAGES

ABx Fast Error Response Packet Structure

ABx Fast error responses contain a two-byte Header, a two-byte Response Size parameter followed by a single-byte Error Flag (0xFF) and a single-byte Error Code parameter, which identifies the error that occurred.

Error Response Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0002
Error Flag:	**0xFF**
Error Code:	Single-byte Error Code
Checksum: This optional parameter holds a single-byte checksum (only applicable when using *ABx Fast with Checksum*).	optional
Response Terminator:	0x03

Table 1-9: ABx Fast Error Response Packet Structure

Error Response Example

Below is an example of a ABx Fast Error Response for **Error Code 0x07** (Tag Not Found).

Error Response Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0002
Error Flag:	**0xFF**
Error Code:	**0x07**
Checksum: This optional parameter holds a single-byte checksum (only applicable when using *ABx Fast with Checksum*).	optional
Response Terminator:	0x03

See the following Error Code Table for details.

1.4.1 Error Codes

Error Code	Error	Description
0x04	FILL TAG FAILED	Fill Operation Failed
0x05	READ DATA FAILED	Read Data Command Failed
0x06	WRITE DATA FAILED	Write Data Command Failed
0x07	READ TAG ID / TAG SEARCH FAILED	Read Tag ID / Tag Search Command Failed (Tag Not Found)
0x21	INVALID SYNTAX	Command Contained a Syntax Error
0x23	INVALID TAG TYPE /RF COMMAND	Invalid or Unsupported Tag Type or RF Command
0x27	LOCK TAG BLOCK FAILED	Lock Tag Block Operation Failed
0x30	INTERNAL CONTROLLER ERROR	Generic Internal Controller Error, Buffer Overflow
0x31	INVALID CONTROLLER TYPE	Invalid Controller Type (when Setting Configuration)
0x32	INVALID PROGRAMMING ADDRESS	Invalid Tag Programming Address Specified in the Command
0x33	CRC ERROR	Cyclic Redundancy Check Error
0x34	INVALID SOFTWARE VERSION	Invalid Software Version
0x35	INVALID RESET	Invalid Hardware Reset
0x36	SET CONFIGURATION ERROR	Controller Configuration not Written
0x37	GET CONFIGURATION ERROR	Controller Configuration not Read

2 ABX FAST RFID COMMANDS

2.1 ABX FAST RFID COMMAND TABLE

Command ID	Command Name	Description
RFID Tag Commands		
0x04	Fill Tag	Writes a specified data byte value to all defined tag addresses
0x05	Read Data	Reads a specified length of data from a contiguous (sequential) area of tag memory
0x06	Write Data	Writes a specified number of bytes to a contiguous area of tag memory
0x07	Read Tag ID	Reads a tag's unique tag ID number
0x08	Tag Search	Instructs the controller to search for a tag in its RF field
0x0D	Start Continuous Read	Instructs the controller to start or stop Continuous Read mode.
0x0E	Read Tag ID and Data	Reads a tag's ID and the requested number of bytes from tag memory
0x0F	Start Continuous Read Tag ID and Data	Places the controller into (or out of) Continuous Read mode and (when evoked) will retrieve a tag's ID.
0x27	Lock Memory Block	Write protects a block of tag memory
0xC2	Read EPC Code	Reads the 12-byte EPC memory area of an EPC Class 1 Gen 2 tag
0xC3	Write EPC Code	Writes the 12-byte EPC memory area of an EPC Class 1 Gen 2 tag
Controller Commands		
0x35	Reset Controller	Resets power to the controller
0x36	Set Controller Configuration	Used to set (configure or modify) the controller's configuration parameters and settings
0x37	Get Controller Configuration	Retrieves the controller's configuration settings
0x38	Get Controller Info	Retrieves hardware, firmware and serial number information from the controller
0x51	Set Controller Time	Used to set the time for the controller
0x72	Execute Macro	Instructs the controller to execute one of its eight possible macros
Additional UHF-Series Controller Commands		
0xC0	Set UHF Controller Configuration	Used to set (configure or modify) additional UHF controller configuration parameters
0xC1	Get Controller Configuration	Retrieves the UHF controller's additional configuration parameter settings

Command ID	Command Name	Description
I/O Commands		
0x10	Set/Clear Digital Outputs	Used to set and/or clear any or all of the RFID Controller digital outputs
0x11	Get Digital Inputs	Retrieves the status of the RFID Controller digital input
0x12	Get Digital Outputs	Retrieve the status of the RFID Controller digital outputs
Multi-Tag RFID Commands		
0x82	Multi-Tag Read ID and Data All	Retrieves the tag ID number and a contiguous segment of data from all RFID tags in range
0x85	Multi-Tag Block Read All	Retrieves a contiguous segment of data from all RFID tags in range
0x86	Multi-Tag Block Write All	Writes a contiguous segment of data to all RFID tags in range
0x87	Multi-Tag Get Inventory	Retrieves the tag ID number from all RFID tags found in range
0x88	Multi-Tag Search All	Checks for the presence of RFID tags in RF range and returns only the number of tags found
0x95	Multi-Tag Block Read by ID	Reads a contiguous segment of data from a specific RFID tag identified by its tag ID
0x96	Multi-Tag Block Write by ID	Writes a contiguous segment of data to a specific RFID tag identified by its tag ID
0xC4	Multi-Tag Read EPC Code	Reads the 12-byte EPC memory area of all EPC Class 1 Gen 2 tags found in range

Table 2-1: ABx Fast RFID Command Table

ABX FAST COMMAND 0X04: FILL TAG

Command 0x04 - Description

The *Fill Tag Command* instructs the specified RFID controller to fill multiple contiguous addresses of an RFID tag with a single byte value. This command is commonly used to clear sequential segments of tag memory by writing the one-byte value repeatedly across a specified range of tag addresses.
This command requires one *Data Byte Value*, a *Start Address* and a *Fill Length*. It will then proceed to fill the tag with the *Data Byte Value*, for the specified *Fill Length* (number of consecutive bytes), beginning at the *Start Address.*
When the *Start Address* is set to zero (*0x0000*), the fill will begin at the first available byte of tag memory. When the *Fill Length* is set to zero (*0x0000*), the controller will write fill data from the *Start Address* to the end of the tag's memory. If the *Fill Length* value extends beyond the last byte in the tag, the controller will return an error.

NOTE

The "Fill Length" in this command represents the number of bytes to fill on the tag, not the length of the 'Data Byte Value' provided in the command, which is always one byte.

Command 0x04 - ABx Fast Example

This example instructs the controller to fill an entire tag with the ASCII character 'A' (*Data Byte Value 0x41*) starting at the beginning of the tag (*Start Address 0x0000*). A *Timeout Value* of 2 seconds (*0x07D0 = 2000 x one-millisecond increments*) is set for the completion of the command.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0008
Command ID:	**0x04**
Start Address: two-byte integer	0x0000
Fill Length: two-byte integer	0x0000
Timeout Value: two-byte integer in ms	0x07D0
Data Byte Value: one-byte	0x41
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0x04**
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X05: READ DATA

Command 0x05 - Description

The *Read Data Command* instructs the controller to retrieve a specific number of bytes of data from a contiguous (*sequential*) area of an RFID tag's memory.
When the *Start Address* is set to zero (*0x0000*), the controller will start reading at the beginning (or first accessible byte) of the tag. The minimum *Block Size* is one byte, the maximum *Block Size* for read operations is 1024 bytes or the entire read/write address space of the tag (whichever is less). If the *Block Size* exceeds the last available tag address, the controller will return an error code.

Command 0x05 - ABx Fast Example

This example instructs the controller to read four bytes of data from a tag starting at address 0x0001. A *Timeout Value* of 2 seconds (*0x07D0 = 2000 x one-millisecond increments*) is set for the completion of the command.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0007
Command ID:	**0x05**
Start Address: two-byte integer	0x0001
Block Size: two-byte integer	0x0004
Timeout Value: two-byte integer in ms	0x07D0
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0005
Command Echo:	**0x05**
Data from Address 0x0001:	0xAA
Data from Address 0x0002:	0xE7
Data from Address 0x0003:	0x0A
Data from Address 0x0004:	0xAA
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X06: WRITE DATA

Command 0x06 - Description

The *Write Data Command* instructs the controller to write specified information to an RFID tag. This command is used to store segments of data in contiguous tag memory locations. It is capable of transferring up to 100 bytes of data from the host to the tag with one command. When the *Start Address* is set to zero (*0x0000*), the controller will begin writing to the first available byte of tag memory. The shortest possible *Block Size* is one byte, the maximum *Block Size* for write operations is 1024 bytes or the entire read/write address space of the tag (whichever is less). If the *Block Size* exceeds the last available tag address, the controller will return an error code.

Command 0x06 - ABx Fast Example

This example instructs the controller to write the five ASCII characters H, E, L, L, O (*Data Byte Values: 0x48, 0x45, 0x4C, 0x4C and 0x4F*) to a tag starting at address 0x000. A *Timeout Value* of 2 seconds (*0x07D0 = 2000 x one-millisecond increments*) is set for the completion of this command.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x000C
Command ID:	**0x06**
Start Address: two-byte integer	0x0000
Block Size: two-byte integer	0x0005
Timeout Value: two-byte integer in ms	0x07D0
Data Byte Value 1: one-byte	0x48
Data Byte Value 2: one-byte	0x45
Data Byte Value 3: one-byte	0x4C
Data Byte Value 4: one-byte	0x4C
Data Byte Value 5: one-byte	0x4F
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0x06**
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X07: READ TAG ID

Command 0x07 - Description

The *Read Tag ID Command* instructs the RFID controller to locate a tag in range and retrieve its unique tag identification number. If a tag is not located before the Timeout Value expires, an error will be returned.
RFID tags are assigned a unique tag ID number or Serial Number during the manufacturing process. After a tag ID number has been assigned to a tag, the value cannot be altered and is not considered part of the available read/write memory space of the tag.

- ISO 14443 compliant tags receive a 4-byte tag ID number. By using just four bytes, tag manufacturers can generate over 4.2 billion possible ISO 14443 compliant tag ID numbers.
- ISO 15693 compliant tags are given an 8-byte tag ID number. When using eight bytes, manufacturers can generate over 280 trillion possible tag ID numbers.

Command 0x07 - ABx Fast Example

This example instructs the controller to retrieve a tag's ID, which, in this example, is the eight-byte value *E0040100002E16AD*. A *Timeout Value* of 2 seconds (*0x07D0 = 2000 x one-millisecond increments*) is set for the completion of the command.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0003
Command ID:	**0x07**
Timeout Value: two-byte integer in ms	0x07D0
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller (Tag Found)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0009
Command Echo:	**0x05**
Tag ID byte 1	0xE0
Tag ID byte 2	0x04
Tag ID byte 3	0x01
Tag ID byte 4	0x00
Tag ID byte 5	0x00
Tag ID byte 6	0x2E
Tag ID byte 7	0x16
Tag ID byte 8	0xAD
Checksum:	optional
Command Terminator:	0x03

Response from Controller (Tag Not Found)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0002
Error Flag:	**0xFF**
Error Code: Tag Not Found	0x07
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X08: TAG SEARCH

Command 0x08 - Description

The *Tag Search Command* instructs the controller to search for the presence of a tag within RF range of the antenna. If the controller finds a tag it will return a *Command Response* to the host. If a tag is not located before the Timeout Value expires, an error will be returned.

Command 0x08 - ABx Fast Example

This example instructs the controller to search for the presence of a tag within RF range of the antenna. A *Timeout Value* of 2 seconds (*0x07D0 = 2000 x one-millisecond increments*) is set for the completion of the command.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0003
Command ID:	**0x08**
Timeout Value: two-byte integer in ms	0x07D0
Checksum: one-byte	1D
Command Terminator:	0x03

Response from Controller (Tag Found)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0x08**
Checksum:	0xF6
Command Terminator:	0x03

Response from Controller (Tag Not Found)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0002
Error Flag:	**0xFF**
Error Code: Tag Not Found	0x07
Checksum:	0xF7
Command Terminator:	0x03

ABX FAST COMMAND 0X0D: START CONTINUOUS READ

Command 0x0D - Description

The *Start Continuous Read Command* instructs the controller to begin (or stop, when evoked) the continual reading of any tag that enters RF range.

When the controller is in *Continuous Read* mode, it will constantly emit RF energy in an attempt to read any tag that comes into range of the antenna. As a tag enters the antenna field, it is immediately read and the data is passed to the host. The controller will continue to read the tag but will not re-send the same data to the host until the tag has moved outside the RF field for a specified time period. This parameter is known as the *Duplicate Read Delay*, which prevents redundant data transmissions when the controller is in Continuous Read mode.

If another RFID command is executed while the controller is in *Continuous Read* mode, the Controller will temporarily stop continuous reading to execute the command, after which the controller will return to *Continuous Read* mode.

The *Continuous Read* command contains three primary components: a *Start Address*, a *Block Size* and a *Duplicate Read Delay* value.

Start Address: The *Start Address* is a 2-byte integer indicating the tag address location where the read will begin.

Block Size: The *Block Size* is a 2-byte integer that represents the number of tag data bytes to retrieve. By setting this parameter to one (*0x0001*) or higher, *Continuous Read* mode will be switched ON at the completion of the command. Setting the *Block Size* to zero (*0x0000*) will disable or turn *Continuous Read* mode off.

Duplicate Read Delay: During *Continuous Read* mode, any tag that comes within range of the antenna will be constantly read and the requested data from the tag will be passed to the host. This single-byte delay parameter indicates the number of seconds that a tag must remain out of RF range before it can be re-read and have its data sent to the host for a second time. It is implemented to enable the operator to limit the volume of information sent by the controller. The *Duplicate Read Delay* parameter can have a value of 0 to 60 seconds. When the *Duplicate Read Delay* value is set to zero, the controller will continuously read AND transmit duplicate tag data to the host.

Continuous Read at Power-up

By default, *Continuous Read* mode is not restarted if the controller is reset. However, through the use of the *Balluff Dashboard™ Configuration Tool ,* the controller can be configured to enter *Continuous Read* mode automatically after a reset or power-up. For more information regarding the Balluff Dashboard™ Configuration Tool , visit the website at *www.balluff.com*.

Command 0x0D - ABx Fast Example

This example places the controller in *Continuous Read* mode and retrieves 4 bytes of data from the tag starting at address 0x0001. The *Duplicate Read Delay* is set to 2 seconds (*0x02 = 2 x 1 second increments*).

Command from Host (Starting Continuous Read)

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0006
Command ID:	**0x0D**
Start Address: two-byte integer	0x0001
Block Size: two-byte integer	0x0004
Duplicate Read Delay: two-byte integer in seconds	0x02
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller (Continuous Read Evoked)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0005
Command Echo:	**0x0D**
Data from Address 0x0001:	0xAA
Data from Address 0x0002:	0xE7
Data from Address 0x0003:	0x0A
Data from Address 0x0004:	0xAA
Checksum:	optional
Command Terminator:	0x03

To exit out of Continuous Read mode, re-issue the command with zero (0x0000) for the Block Size.

Command from Host (Stopping Continuous Read)

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0006
Command ID:	**0x0D**
Start Address: two-byte integer	0x0001
Block Size: two-byte integer	0x0000
Duplicate Read Delay: two-byte integer in seconds	0x02
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller (Continuous Read Stopped)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0x0D**
Checksum:	optional
Command Terminator:	0x03

Continuous Read Mode Controller LED Behavior

LED	Behavior	Description
READY	ON	Controller is powered and functioning
COM	ON	Duplicate Read Delay ≥ 1 and a tag has entered the RF field. COM LED will remain ON while a tag is in the RF field. After the tag has exited the RF field the COM light will remain ON for the duration of the Duplicate Read Delay before turning OFF.
COM	BLINKING	Duplicate Read Delay = 0 and a tag is in the RF field
RF	ON	Continuous Read mode is enabled

Table 2-2: Continuous Read Mode LED Behavior

ABX FAST COMMAND 0X0E: READ TAG ID AND DATA

Command 0x0E - Description

The *Read Tag ID and Data Command* instructs the RFID controller to retrieve a tag's unique identification (*Tag ID*) number followed by the requested data. The minimum *Block Size* for read operations is one byte, the maximum *Block Size* for read operations is 1024 bytes or the entire read/write address space of the tag (minus the number of tag ID bytes), whichever is less.

Command 0x0E - ABx Fast Example

This example instructs the controller to retrieve the tag ID and two bytes beginning at address 0x0001 from a tag within range of the controller. A *Timeout Value* of 2 seconds (*0x07D0 = 2000 x one-millisecond increments*) is set for the completion of the command.
In this example the tag ID number retrieved is *E0040100002E16AD*.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0007
Command ID:	**0x0E**
Start Address: two-byte integer	0x0001
Block Size: two-byte integer	0x0002
Timeout Value: two-byte integer in ms	0x07D0
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x000B
Command Echo:	**0x0E**
Tag ID byte 1	0xE0
Tag ID byte 2	0x04
Tag ID byte 3	0x01
Tag ID byte 4	0x00
Tag ID byte 5	0x00
Tag ID byte 6	0x2E
Tag ID byte 7	0x16
Tag ID byte 8	0xAD
Data from Address 0x0001:	0xAA
Data from Address 0x0002:	0xE7
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X0F: START CONTINUOUS READ TAG ID AND DATA

Command 0x0F - Description

The *Start Continuous Read Tag ID and Data Command* instructs the controller to repeatedly attempt to retrieve the tag ID and a specified number of tag data bytes from any tag that enters RF range. This command is similar to *Command 0x0D*; however, *Command 0x0F* additionally retrieves the tag's ID number.
This command contains four primary parameters: *Start Address*, *Block Size, Start/Stop Flag,* and *Duplicate Read Delay*.

Start Address: The *Start Address* is a 2-byte integer indicating the tag address location where the read will begin.

Block Size: The *Block Size* is a 2-byte integer that represents the number of tag data bytes to retrieve, beginning at the specified *Start Address* location.

Duplicate Read Delay: The *Duplicate Read Delay* is a single-byte value representing the number of seconds that a tag must remain OUT of RF range before it can be re-read and have its data sent to the host for a second time. It is used to limit or prevent the volume of redundant information that is transmitted while the controller is performing continuous reads. The parameter can have a value of 0 to 60 seconds (*0x00 – 0x3C*).
For example, when the *Duplicate Read Delay* is set to two (0x02), a tag that has already been read must exit the antenna's RF field for at least two seconds before the controller will recognize it again and re-send its data to the host. When the value is set to zero, the controller will continuously read AND transmit duplicate tag data to the host.

Start/Stop Flag: By setting the single-byte *Start/Stop Flag* parameter to one (*0x01*), continuous read mode will be switched ON upon execution of the command. Setting the parameter value back to zero (*0x00*) and re-issuing the command will turn continuous read mode OFF.

Command 0x0F - ABx Fast Example

This example places the controller into *Continuous Read Tag ID and Data* mode, retrieves the tag ID number and reads two bytes of data starting at address 0x0001 from any tag within range. The *Duplicate Read Delay* is set for 2 seconds (*0x02 = 2 x 1 second increments*).

Command from Host (Starting Continuous Read)

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0007
Command ID:	**0x0F**
Start Address: two-byte integer	0x0001
Block Size: two-byte integer	0x0002
Duplicate Read Delay: two-byte integer in seconds	0x02
Start/Stop Flag: 0x01 = Start, 0x00 = Stop	0x01
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller (Continuous Read Evoked)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x000B
Command Echo:	**0x0F**
Tag ID byte 1	0xE0
Tag ID byte 2	0x04
Tag ID byte 3	0x01
Tag ID byte 4	0x00
Tag ID byte 5	0x00
Tag ID byte 6	0x2E
Tag ID byte 7	0x16
Tag ID byte 8	0xAD
Data from Address 0x0001:	0xAA
Data from Address 0x0002:	0xE7
Checksum:	optional
Command Terminator:	0x03

To exit out of Continuous Read Tag ID and Data mode, re-issue the command with zero (0x00) in the *Start/Stop Flag* field.

Command from Host (Stopping Continuous Read)

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0007
Command ID:	**0x0F**
Start Address: two-byte integer	0x0001
Block Size: two-byte integer	0x0002
Duplicate Read Delay: two-byte integer in seconds	0x02
Start/Stop Flag: 0x01 = Start, 0x00 = Stop	0x00
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller (Continuous Read Stopped)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0x0F**
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X10: SET/CLEAR DIGITAL OUTPUTS

Command 0x10 - Description

The *Set/Clear Digital Outputs Command* is used to set or clear any or all of the device's digital outputs.

The 8-bit *Digital Output Mask* is used to select which of the Digital Outputs you wish to affect (i.e., which Digital Outputs will be set or cleared).

The 8-bit *Data State* is used to indicate whether the affected Digital Outputs will either be set (1 = SET) or cleared (0 = CLEAR).

Currently HF and UHF I/O model processor units are equipped with two Digital Outputs.
Only the lowest two bits are used (one bit for each Digital Output, where the lowest bit = Output 1, the next highest bit = Output 2).

Command 0x10 – ABx Fast Example

This example sets outputs 1 and 2.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0003
Command ID:	**0x10**
Output Mask:	0x03
Data State:	0x03
Checksum:	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0x10**
Checksum:	optional
Command Terminator:	0x03

Additional Output Mask / Data State Examples:

Example 2:
Clear both outputs 1 and 2

LSB HEX VALUE		BINARY VALUE		RESULT
		always zero	Outputs **x x 2 1**	
Output Mask	0x03	0 0 0 0	0 0 1 1	Affect both Digital Outputs
Data State	0x00	0 0 0 0	0 0 0 0	Clear both Outputs

Example 3:
Set output 2 and clear output 1

LSB HEX VALUE		BINARY VALUE		RESULT
		always zero	Outputs **x x 2 1**	
Output Mask	0x03	0 0 0 0	0 0 1 1	Affect both outputs
Data State	0x02	0 0 0 0	0 0 1 0	Set output 2 and clears output 1

Example 4:
Set output 1: output 2 is not affected
If a bit in the Data State is set, yet the corresponding bit in the Output Mask is not set, it will be ignored.

LSB HEX VALUE		BINARY VALUE		RESULT
		always zero	Outputs **x x 2 1**	
Output Mask	0x01	0 0 0 0	0 0 0 1	Affect only output 1
Data State	0x0F	0 0 0 0	1 1 1 1	Set output 1, other bits are ignored

ABX FAST COMMAND 0X11: GET DIGITAL INPUT

Command 0x11 - Description

The *Get Digital Inputs Command* is used to retrieve the 8-bit *Digital Input State* value, which identifies the status of the device's digital input. Currently HF and UHF I/O model processor units are equipped with one Digital Input. See the *Digital Input State Table* for definitions.

Command 0x11 – ABx Fast Example

This example retrieves the status of the Controller Input.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0001
Command ID:	**0x11**
Checksum:	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0002
Command Echo:	**0x11**
Digital Input State:	**0x00**
Checksum:	optional
Command Terminator:	0x03

Digital Input State Table

LSB	BINARY VALUE		RESULT
	always zero	Input **x x x 1**	
0x01	0 0 0 0	0 0 0 1	Digital Input 1 = Set
0x00	0 0 0 0	0 0 0 0	Digital Input 1 = Clear

Table 2-3: Digital Input State

ABX FAST COMMAND 0X12: GET DIGITAL OUTPUTS

Command 0x12 - Description

The *Get Digital Outputs Command* is used to retrieve the 8-bit *Digital Output State* value, which identifies the status of the device's digital outputs. Currently HF and UHF I/O model processor units are equipped with two Digital Outputs. See the *Digital Output State Table* for definitions.

Command 0x12 – ABx Fast Example

This example retrieves the status of the two Controller Outputs.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0001
Command ID:	**0x12**
Checksum:	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0002
Command Echo:	**0x12**
Digital Output State:	**0x02**
Checksum:	optional
Command Terminator:	0x03

Digital Output State Table

LSB	BINARY VALUE		RESULT
	always zero	Output **x x 2 1**	
0x03	0 0 0 0	0 0 1 1	Both Digital Outputs 1 and 2 = Set
0x02	0 0 0 0	0 0 1 0	Digital Output 1 = Clear, Output 2 = Set
0x01	0 0 0 0	0 0 0 1	Digital Output 1 = Set, Output 2 = Clear
0x00	0 0 0 0	0 0 0 0	Both Digital Outputs 1 and 2 = Clear

Table 2-4: Digital Output State

ABX FAST COMMAND 0X27: LOCK MEMORY BLOCK

Command 0x27 - Description

The *Lock Memory Block Command* allows the user to write protect or lock a block of tag memory to prevent data from being overwritten.
The *Starting Block* parameter specifies the first block of tag memory addresses to be locked.
The *Number of Blocks* parameter specifies the number of blocks to lock, (*1 ~ N*).
Depending on the architecture of the tag used, a "block" can be either 4-bytes or 8-bytes. Users must know the memory architecture and block size of their tag before using this command.
This command only supports ISO 15693 compliant RFID tags.

NOTE

Extreme caution should be taken when using this command. Once a block of tag memory is locked, it cannot be unlocked and all data written to the block is permanent.

Command 0x27 – ABx Fast Example

This example instructs the controller to lock two blocks of tag memory, beginning at block address 0x00 (the first available block). A *Timeout Value* of two seconds (*0x07D0 = 2000 x one-millisecond increments*) is set for the completion of this command.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0005
Command ID:	**0x27**
Starting Block: one-byte	0x00
Number of Blocks: one-byte	0x02
Timeout Value: two-byte integer in ms	0x07D0
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0x27**
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X35: RESET CONTROLLER

Command 0x35 - Description

The *Reset Controller Command* is used to reset power to the specified controller without clearing any stored configuration information. However, the controller's configuration can be reset to default values when *Command 0x35* is issued AND a Configuration Tag is placed in the antenna's RF field prior to execution.

Command 0x35 - ABx Fast Example

This example resets power to the controller.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0001
Command ID:	**0x35**
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0x35**
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X36: SET CONTROLLER CONFIGURATION

Command 0x36 - Description

The *Set Controller Configuration Command* is used to modify and save changes to the controller's configuration, which will be stored in the unit's flash memory.

NOTE

It is recommended that the user first run the Command 0x37: "Get Controller Configuration" and make note of the current controller configuration values prior to executing this command.

Command 0x36 - ABx Fast Example

This example sets the following configuration parameters for the controller.

Command from Host

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0014
Command Echo:	**0x36**
Continuous Read at Power-up - Start Address: (When performing a Continuous Read at power-up or reset, this 2-byte value indicates the starting tag address for the read (0x0000 = begin reading at first available byte).	0x0000
Continuous Read at Power-up - Block Size: (When performing a Continuous Read at power-up or reset, this 2-byte value indicates the number of continuous bytes to be read. If this value is zero, the controller will not enter Continuous Read mode at power-up or reset. Default is zero 0x0000).	0x0000
Continuous Read at Power-up - Duplicate Read Delay: (This single-byte value indicates the number of seconds that a tag must remain out of the RF field before it will be read and have its data sent to the host again. Only applicable when Block Size ≥ 1 thereby enabling Continuous Read at power-up).	0x00
Node ID: For -485 models only, this single-byte value indicates the controller's Node ID number (between 01–16), for all other processor units, the default = one (0x01)	0x01
Reserved	0x00
ABx Protocol: Single-byte value indicating the ABx protocol in use: 0x00 = ABx Fast without checksum* 0x01 = ABx Fast with checksum	0x00
Tag Type: Single-byte value indicates tag IC in use: 0x01 = BIS M-1__-03_ (ISO 15693, I-CODE SLI) 0x03 = BIS M-1__-07_ (ISO 15693) 0x05 = BIS M-1__-10_ (Mifare) 0x06 = BIS M-1__-02_ (ISO 15693) This parameter is ignored by BIS U-62_ processor unit.	0x01
Reserved	0x00
Communication Mode: (See *Communications Mode Table* below)	0x44

<table>
<tr><th>Response Packet Element</th><th>Content</th></tr>
<tr><td>Options Byte 1
<table><tr><th>BIT</th><th>Description</th></tr><tr><td>0</td><td>Tag IDs – LSB First</td></tr><tr><td>1</td><td>Reserved</td></tr><tr><td>2</td><td>Reserved</td></tr><tr><td>3</td><td>Reserved</td></tr><tr><td>4</td><td>Reserved</td></tr><tr><td>5</td><td>Tag IDs in Continuous Read</td></tr><tr><td>6</td><td>Tag Presence ON</td></tr><tr><td>7</td><td>Toggle RF</td></tr></table></td><td>0x00</td></tr>
<tr><td>Reserved</td><td>0x00</td></tr>
<tr><td>Options Byte 2
<table><tr><th>BIT</th><th>Description</th></tr><tr><td>0</td><td>4-Byte Tag IDs (Mifare)</td></tr><tr><td>1</td><td>Legacy Error Codes</td></tr><tr><td>2</td><td>Reserved</td></tr><tr><td>3</td><td>Reserved</td></tr><tr><td>4</td><td>Reserved</td></tr><tr><td>5</td><td>Reserved</td></tr><tr><td>6</td><td>Reserved</td></tr><tr><td>7</td><td>Reserved</td></tr></table></td><td>0x00</td></tr>
<tr><td>Controller Type: one byte value
Single-byte value indicates controller type:
0x02 = BIS M-62_
0x03 = BIS M-410_
0x04 = BIS M-411_
0x0B = BSI U-62_</td><td>0x02</td></tr>
<tr><td>Software Major Release Digit (first digit in the software version)
Example software version: 2.0.A.0
Major Release Digit in this example = 2</td><td>0x02</td></tr>
<tr><td>Software Minor Release Digit (second digit in the software version)
Example software version: 2.0.A.0
Minor Release Digit in this example = 0</td><td>0x00</td></tr>
<tr><td>Software Correction Release Digit (first digit in the software version)
Example software version: 2.0.A.0
Correction Release Digit in this example = A</td><td>0x41</td></tr>
<tr><td>Software Point Release Digit (first digit in the software version)
Example software version: 2.0.A.0
Point Release Digit in this example = 0</td><td>0x00</td></tr>
<tr><td>Checksum:</td><td>optional</td></tr>
<tr><td>Command Terminator:</td><td>0x03</td></tr>
</table>

Communications Mode Table

BITS	7 6	5 4	3	2 1 0
0 1	Interface: 00 = RS422 01 = RS232 / USB 10 = RS485 11 = Others	Reserved	XON/XOFF: 0 = disabled 1 = enabled	Baud Rate: 000 = 9600 001 = 19200 010 = 38400 011 = 57600 100 = 115200

Table 2-5: Communications Mode Table

NOTE

In the above example, the Communications Mode value of 0x44 = 01000100 in binary, meaning that the controller is an RS232 or USB device that is communicating at 115200 baud.

Response from Controller

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0001
Command ID:	**0x36**
Checksum: one-byte	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X37: GET CONTROLLER CONFIGURATION

Command 0x37 - Description

The *Get Controller Configuration Command* is used to retrieve the configuration settings stored in the controller's flash memory.

Command 0x37 - ABx Fast Example

This example queries a BIS M-410-068-001-09-S72 controller and uploads to the host, the stored configuration parameters and settings from the controller's flash memory.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0002
Command ID:	**0x37**
Controller Type: one byte value Single-byte value indicates controller type: 0x02 = BIS M-62_ 0x03 = BIS M-410_ 0x04 = BIS M-411_ 0x0B = BSI U-62_	0x03
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0014
Command Echo:	**0x37**
Continuous Read at Power-up - Start Address: (When performing a Continuous Read at power-up or reset, this 2-byte value indicates the starting tag address for the read (0x0000 = begin reading at first available byte).	0x0000
Continuous Read at Power-up - Block Size: (When performing a Continuous Read at power-up or reset, this 2-byte value indicates the number of continuous bytes to be read. If this value is zero, the controller will not enter Continuous Read mode at power-up or reset. Default is zero 0x0000).	0x0000
Continuous Read at Power-up - Duplicate Read Delay: (This single-byte value indicates the number of seconds that a tag must remain out of the RF field before it will be read and have its data sent to the host again. Only applicable when Block Size ≥ 1 thereby enabling Continuous Read at power-up).	0x00
Node ID: For -485 models only, this single-byte value indicates the controller's Node ID number (between 01–16), for all other processor units, the default = one (0x01)	0x01
Reserved	0x00
ABx Protocol: Single-byte value indicating the ABx protocol in use: 0x00 = ABx Fast without checksum* 0x01 = ABx Fast with checksum	0x00

<table>
<tr><th>Response Packet Element</th><th>Content</th></tr>
<tr><td>Tag Type: Single-byte value indicates tag IC in use:
0x01 = BIS M-1__-03_ (ISO 15693, I-CODE SLI)
0x03 = BIS M-1__-07_ (ISO 15693)
0x05 = BIS M-1__-10_ (Mifare)
0x06 = BIS M-1__-02_ (ISO 15693)
This parameter is ignored by BIS U-62_ processor unit.</td><td>0x01</td></tr>
<tr><td>Reserved</td><td>0x00</td></tr>
<tr><td>Communication Mode: (See Communications Mode Table below)</td><td>0x44</td></tr>
<tr><td>Options Byte 1
<table><tr><th>BIT</th><th>Description</th></tr><tr><td>0</td><td>Tag IDs – LSB First</td></tr><tr><td>1</td><td>Reserved</td></tr><tr><td>2</td><td>Reserved</td></tr><tr><td>3</td><td>Reserved</td></tr><tr><td>4</td><td>Reserved</td></tr><tr><td>5</td><td>Tag IDs in Continuous Read</td></tr><tr><td>6</td><td>Tag Presence ON</td></tr><tr><td>7</td><td>Toggle RF</td></tr></table></td><td>0x00</td></tr>
<tr><td>Reserved</td><td>0x00</td></tr>
<tr><td>Options Byte 2
<table><tr><th>BIT</th><th>Description</th></tr><tr><td>0</td><td>4-Byte Tag IDs (Mifare)</td></tr><tr><td>1</td><td>Legacy Error Codes</td></tr><tr><td>2</td><td>Reserved</td></tr><tr><td>3</td><td>Reserved</td></tr><tr><td>4</td><td>Reserved</td></tr><tr><td>5</td><td>Reserved</td></tr><tr><td>6</td><td>Reserved</td></tr><tr><td>7</td><td>Reserved</td></tr></table></td><td>0x00</td></tr>
<tr><td>Controller Type: one byte value
Single-byte value indicates controller type:
0x02 = BIS M-62_
0x03 = BIS M-410_
0x04 = BIS M-411_
0x0B = BSI U-62_</td><td>0x03</td></tr>
<tr><td>Software Major Release Digit (first digit in the software version)
Example software version: 2.0.A.0
Major Release Digit in this example = 2</td><td>0x02</td></tr>
<tr><td>Software Minor Release Digit (second digit in the software version)
Example software version: 2.0.A.0
Minor Release Digit in this example = 0</td><td>0x00</td></tr>
<tr><td>Software Correction Release Digit (first digit in the software version)
Example software version: 2.0.A.0
Correction Release Digit in this example = A</td><td>0x41</td></tr>
<tr><td>Software Point Release Digit (first digit in the software version)
Example software version: 2.0.A.0
Point Release Digit in this example = 0</td><td>0x00</td></tr>
<tr><td>Checksum:</td><td>optional</td></tr>
<tr><td>Command Terminator:</td><td>0x03</td></tr>
</table>

Communications Mode Table

BITS	7 6	5 4	3	2 1 0
0 1	Interface: 00 = RS422 01 = RS232 / USB 10 = RS485 11 = Others	Reserved	XON/XOFF: 0 = disabled 1 = enabled	Baud Rate: 000 = 9600 001 = 19200 010 = 38400 011 = 57600 100 = 115200

Table 2-6: Communications Mode Table

NOTE

In the above example, the Communications Mode value of 0x44 = 01000100 in binary, meaning that the controller is an RS232 or USB device that is communicating at 115200 baud.

ABX FAST COMMAND 0X38: GET CONTROLLER INFO

Command 0x38 - Description

The *Get Controller Info Command* is used to retrieve hardware version, serial number and installed firmware identification information from the specified controller.

Command 0x38 - ABx Fast Example

This example retrieves information from the controller.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0001
Command ID:	**0x38**
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x001B
Command Echo:	**0x38**
Controller Type: one byte value Single-byte value indicates controller type: 0x02 = BIS M-62_ 0x03 = BIS M-410_ 0x04 = BIS M-411_ 0x0B = BSI U-62_	0x02
Software Major Release Digit (first digit in the software version) Example software version: *2.0.A.0* Major Release Digit in this example = 2	0x02
Software Minor Release Digit (second digit in the software version) Example software version: *2.0.A.0* Minor Release Digit in this example = 0	0x00
Software Correction Release Digit (first digit in the software version) Example software version: *2.0.A.0* Correction Release Digit in this example = A	0x41
Software Point Release Digit (first digit in the software version) Example software version: *2.0.A.0* Point Release Digit in this example = 0	0x00
Reserved	0x01
Software CRC: two-byte value	0x0000
Loader Software Version: two-byte value	0x0000
Processor ID: five-byte processor ID number	0x30FFFF0F04
Processor RFU: three-byte processor RFU value	0x000000
Processor Serial Number: four-byte value	0xF143A63B

Response Packet Element	Content
Processor Internal Information: two-byte value	0xA568
Processor RsMaxP: one-byte value	0x5E
Processor Information CRC: one-byte value for the RC632 Information CRC	0xBD
Checksum:	optional
Command Terminator:	0x03

NOTE

Controller information retrieved may be different from the values listed above.

ABX FAST COMMAND 0X51: SET CONTROLLER TIME

Command 0x51 - Description

The *Set Controller Time Command* is used to set the controller's internal clock and calendar. Date and Time are specified in the following format:

- Year (2-byte Integer)
- Month (byte), Day of Month (byte)
- Hour (byte), Minute (byte)
- Seconds (byte)

Command 0x51 - ABx Fast Example

This example sets the date and time on the controller to *March 19, 2007 - 9:30 a.m*.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0008
Command ID:	**0x51**
Year: two-byte integer (*2007*)	0x07D7
Month, Day: two bytes (*March, 19th*)	0x03, 0x13
Hours, Minutes: two bytes (*9:30 AM*)	0x09, 0x1E
Seconds: one byte	0x00
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0x51**
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X72: EXECUTE MACRO

Command 0x72 - Description

The *Execute Macro Command* is used to perform one of the controller's eight possible macros.

To design your own RFID command macros, use the C-Macro™ Builder software tool.

The value 0x00 in the *Macro Number* means stop all macro execution.

Command 0x72 - ABx Fast Example

This example instructs the controller to execute *Macro #1*.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0002
Command ID:	**0x72**
Macro Number: (0x01 to 0x08, 0x00 = stop all macros)	0x01
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0x72**
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X82: MULTI-TAG READ ID AND DATA ALL

Command 0x82 - Description

The *Multi-Tag Read ID and Data All Command* is used to retrieve the tag ID and a contiguous segment of data from all RFID tags in range (or those with the specified *AFI*).

The controller will generate separate responses containing the tag ID and the requested data for each responding tag. Each response packet will be sent to the host as soon as it is available. The returned tag IDs can be used to read from/write to a single tag (via the *Multi-Tag Block Read/Write by ID* commands) when multiple tags are identified simultaneously.

A final termination packet is sent when the *Timeout Value* expires.

When the *Start Address* is set to zero (*0x0000*), the controller will start reading at the beginning (or first accessible byte) of the tag.

If the *Block Size* exceeds the last tag address, the controller will return a Command Syntax error message (*error code 0x21*).

Command 0x82 - ABx Fast Example

This example instructs the controller to read the tag ID and 16 bytes of data from each tag in range starting at address 0x0001. A *Timeout Value* of 3 seconds (*0x0BB8 = 3000 x 1msec increments*) is set for the completion of the command. The *AFI* byte is set to zero so that all tags in range will respond.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x000A
Command ID:	**0x82**
AFI: one-byte	0x00
Anti-collision Mode: (0x00 = single slot, 0x01 = 16-Slot)	0x01
Tag Limit: maximum # of tags expected in RF range up to 100	0x64
Start Address: two-byte integer	0x0001
Block Size: two-byte integer	0x0010
Timeout Value: two-byte integer in ms	0x0BB8
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller (for Each Tag Found)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0019
Command Echo:	**0x82**
Tag ID (Serial Number) byte 1	<SN1>
Tag ID (Serial Number) byte 2	<SN2>
Tag ID (Serial Number) byte 3	<SN3>
Tag ID (Serial Number) byte 4	<SN4>
Tag ID (Serial Number) byte 5	<SN5>
Tag ID (Serial Number) byte 6	<SN6>
Tag ID (Serial Number) byte 7	<SN7>
Tag ID (Serial Number) byte 8	<SN8>
Read Data Byte 1:	<D1>
Read Data Byte 2:	<D2>
. . .	
Read Data Byte 16:	<D16>
Checksum:	optional
Command Terminator:	0x03

Final Response Packet

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0003
Final Termination Packet ID:	**0xFF**
Number of Tags Read	<N tags>
Status: 0x00 = OK, any other value = error	0x00
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X85: MULTI-TAG BLOCK READ ALL

Command 0x85 - Description

The *Multi-Tag Block Read All Command* is used to retrieve a contiguous segment of data from all RFID tags within RF range (or those with the specified *AFI*).

The controller will generate separate responses containing the requested data for each responding tag. Each response packet will be sent to the host as soon as it is available.

A final termination packet is sent when the *Timeout Value* expires.

When the *Start Address* is set to zero (*0x0000*), the controller will start reading at the beginning (or first accessible byte) of the tag.

If the *Block Size* exceeds the last tag address, the controller will return a Command Syntax error message (*error code 0x21*).

Command 0x85 - ABx Fast Example

This example instructs the controller to read 16 bytes of data from each tag in range starting at address 0x0001. A *Timeout Value* of 3 seconds (*0x0BB8 = 3000 x 1msec increments*) is set for the completion of the command. The *AFI* byte is set to zero so all tags in range will respond.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x000A
Command ID:	**0x85**
AFI: one-byte	0x00
Anti-collision Mode: (0x00 = single slot, 0x01 = 16-Slot)	0x01
Tag Limit: maximum # of tags expected in RF range up to 100	0x64
Start Address: two-byte integer	0x0001
Block Size: two-byte integer	0x0010
Timeout Value: two-byte integer in ms	0x0BB8
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller (for Each Tag Found)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0011
Command Echo:	**0x85**
Read Data Byte 1:	<D1>
Read Data Byte 2:	<D2>
. . .	
Read Data Byte 16:	<D16>
Checksum:	optional
Command Terminator:	0x03

Final Response Packet

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0003
Final Termination Packet ID:	**0xFF**
Number of Tags Read	<N tags>
Status: 0x00 = OK, any other value = error	0x00
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X86: MULTI-TAG BLOCK WRITE ALL

Command 0x86 - Description

The *Multi-Tag Block Write All Command* is used to write a contiguous segment of data to all RFID tags in range (or those with the specified *AFI*).

When the *Start Address* is set to zero (*0x0000*), the controller will start writing at the beginning (or first accessible byte) of the tag.

If the *Block Size* exceeds the last tag address, the controller will return a Command Syntax error message (*error code 0x21*).

Command 0x86 - ABx Fast Example

This example instructs the controller to write 16 bytes of data to each RFID tag in range starting at address 0x0001. A *Timeout Value* of 3 seconds (*0x0BB8 = 3000 x 1msec increments*) is set for the completion of the command. The *AFI* byte is set to zero so that data will be written to all tags.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x001A
Command ID:	**0x86**
AFI: one-byte	0x00
Anti-collision Mode: (0x00 = single slot, 0x01 = 16-Slot)	0x01
Tag Limit: maximum # of tags expected in RF range up to 100	0x64
Start Address: two-byte integer	0x0001
Block Size: two-byte integer	0x0010
Timeout Value: two-byte integer in ms	0x0BB8
Write Data Byte 1:	<D1>
Write Data Byte 2:	<D2>
. . .	
Write Data Byte 16:	<D16>
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0003
Final Termination Packet ID:	**0xFF**
Number of Tags Written	<N tags>
Status: 0x00 = OK, any other value = error	0x00
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X87: MULTI-TAG GET INVENTORY

Command 0x87 - Description

The *Multi-Tag Get Inventory Command* is used to retrieve the tag ID from all RFID tags in range (or those with the specified *AFI*). Each tag has a unique Tag ID or Serial Number. This number cannot be changed and is not considered part of the tag's available memory.

The controller will generate separate responses containing the tag ID for each responding tag. Each response packet will be sent to the host as soon as it is available. The returned tag IDs can be used to read from/write to a single tag (via the *Multi-Tag Block Read/Write by ID* commands) when multiple tags are identified simultaneously.

A final termination packet is sent when the *Timeout Value* expires.

Command 0x87 - ABx Fast Example

This example instructs the controller to read the tag ID from each tag in range. A *Timeout Value* of 3 seconds (*0x0BB8 = 3000 x 1msec increments*) is set for the completion of the command. The *AFI* byte is set to zero so all tags in range will respond.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0006
Command ID:	**0x87**
AFI: one-byte	0x00
Anti-collision Mode: (0x00 = single slot, 0x01 = 16-Slot)	0x01
Tag Limit: maximum # of tags expected in RF range up to 100	0x64
Timeout Value: two-byte integer in ms	0x0BB8
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller (for Each Tag Found)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0009
Command Echo:	**0x87**
Tag ID (Serial Number) byte 1	<SN1>
Tag ID (Serial Number) byte 2	<SN2>
Tag ID (Serial Number) byte 3	<SN3>
Tag ID (Serial Number) byte 4	<SN4>
Tag ID (Serial Number) byte 5	<SN5>
Tag ID (Serial Number) byte 6	<SN6>
Tag ID (Serial Number) byte 7	<SN7>
Tag ID (Serial Number) byte 8	<SN8>
Checksum:	optional
Command Terminator:	0x03

Final Response Packet

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0003
Final Termination Packet ID:	**0xFF**
Number of Tags Read	<N tags>
Status: 0x00 = OK, any other value = error	0x00
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X88: MULTI-TAG SEARCH ALL

Command 0x88 - Description

The *Multi-Tag Search All Command* is used to check for the presence of RFID tags in the antenna field.

As soon as the controller finds a tag, it will return a response to the host. If no tags are present, and the *Timeout Value* has expired, the controller will return an error.

Command 0x88 - ABx Fast Example

This example instructs the controller to search for the presence of any RFID tag in range. A *Timeout Value* of 3 seconds (*0x0BB8 = 3000 x 1msec increments*) is set for the completion of the command. The *AFI* byte is set to zero so all tags in range will respond.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0006
Command ID:	**0x88**
AFI: one-byte	0x00
Anti-collision Mode: (0x00 = single slot, 0x01 = 16-Slot)	0x01
Tag Limit: maximum # of tags expected in RF range up to 100	0x64
Timeout Value: two-byte integer in ms	0x0BB8
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0003
Final Termination Packet ID:	**0xFF**
Number of Tags Read	<N tags>
Status: 0x00 = OK, any other value = error	0x00
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X95: MULTI-TAG BLOCK READ BY ID

Command 0x95 - Description

The *Multi-Tag Block Read by ID Command* is used to read a contiguous segment of data from a specific RFID tag identified by its 8-byte tag ID.

An error packet is sent if the specific tag is not present and the *Timeout Value* expires.

When the *Start Address* is set to zero (*0x0000*), the controller will start reading at the beginning (or first accessible byte) of the tag.

If the *Block Size* exceeds the last tag address, the controller will return a Command Syntax error message (*error code 0x21*).

Command 0x95 - ABx Fast Example

This example instructs the controller to read 16 bytes of data from the tag containing the specified tag ID, beginning at tag address 0x0001. A *Timeout Value* of 3 seconds (*0x0BB8 = 3000 x 1msec increments*) is set for the completion of the command.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size: (always 0x11)	0x0011
Command ID:	**0x95**
AFI: one-byte	0x00
Anti-collision Mode: (0x00 = single slot, 0x01 = 16-Slot)	0x01
Start Address: two-byte integer	0x0001
Block Size: two-byte integer	0x0010
Timeout Value: two-byte integer in ms	0x0BB8
Tag ID (Serial Number) byte 1	<SN1>
Tag ID (Serial Number) byte 2	<SN2>
Tag ID (Serial Number) byte 3	<SN3>
Tag ID (Serial Number) byte 4	<SN4>
Tag ID (Serial Number) byte 5	<SN5>
Tag ID (Serial Number) byte 6	<SN6>
Tag ID (Serial Number) byte 7	<SN7>
Tag ID (Serial Number) byte 8	<SN8>
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0011
Command Echo:	**0x95**
Read Data Byte 1:	<D1>
Read Data Byte 2:	<D2>
. . .	
Read Data Byte 16:	<D16>
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0X96: MULTI-TAG BLOCK WRITE BY ID

Command 0x96 - Description

The *Multi-Tag Block Write by ID Command* is used to write a contiguous segment of data to a specific RFID tag identified by its 8-byte tag ID.

An error packet is sent if the specific tag is not present and the *Timeout Value* expires.

When the *Start Address* is set to zero (*0x0000*), the controller will start writing at the beginning (or first accessible byte) of the tag.

If the *Block Size* exceeds the last tag address, the controller will return a Command Syntax error message (*error code 0x21*).

Command 0x96 - ABx Fast Example

This example instructs the controller to write 16 bytes of data to the tag containing the specified tag ID, beginning at tag address 0x0001. A *Timeout Value* of 3 seconds (*0x0BB8 = 3000 x 1msec increments*) is set for the completion of the command.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size: (Block Size + 0x11)	0x0021
Command ID:	**0x96**
AFI: one-byte	0x00
Anti-collision Mode: (0x00 = single slot, 0x01 = 16-Slot)	0x01
Start Address: two-byte integer	0x0001
Block Size: two-byte integer	0x0010
Timeout Value: two-byte integer in ms	0x0BB8
Tag ID (Serial Number) byte 1	<SN1>
Tag ID (Serial Number) byte 2	<SN2>
Tag ID (Serial Number) byte 3	<SN3>
Tag ID (Serial Number) byte 4	<SN4>
Tag ID (Serial Number) byte 5	<SN5>
Tag ID (Serial Number) byte 6	<SN6>
Tag ID (Serial Number) byte 7	<SN7>
Tag ID (Serial Number) byte 8	<SN8>
Write Data Byte 1:	<D1>
Write Data Byte 2:	<D2>
. . .	
Write Data Byte 16:	<D16>
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0x96**
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0XC0: SET UHF CONTROLLER CONFIGURATION

Command 0xC0 - Description

The *Set UHF Controller Configuration Command* is used to modify and save changes to the UHF-Series controller's configuration, which will be stored in the unit's flash memory.

NOTE

It is recommended that the user first run the Command 0xC1: "Get UHF Controller Configuration" and make note of the current controller configuration values prior to executing this command.

Command 0xC0 - ABx Fast Example

This example permits the user to modify or write the indicated configuration parameters to the flash memory of the UHF controller. The total number of bytes available for this purpose is **nine**.

Command from Host

<table>
<tr><th>Response Packet Element</th><th>Content</th></tr>
<tr><td>Command Header:</td><td>0x02, 0x02</td></tr>
<tr><td>Command Size:</td><td>0x000A</td></tr>
<tr><td>Command ID:</td><td>0xC0</td></tr>
<tr><td>UHF Configuration Byte 1 & 2: two-byte integer
These two bytes represent the Reader Output Power (value from 0 to 500 mW).</td><td>0x01F4</td></tr>
<tr><td>UHF Configuration Byte 3
UHF Configuration Byte 4</td><td><Reserved>*</td></tr>
<tr><td>UHF Configuration Byte 5
UHF Configuration Byte 6</td><td><Reserved>*</td></tr>
<tr><td>UHF Configuration Byte 7</td><td><Reserved>*</td></tr>
<tr><td>UHF Configuration Byte 8
Byte 8 is partially reserved according to the table below. This byte permits the user to select the specific UHF Channel through which commands are transmitted. The user can write one of the following values: 0, 3, 6, or 9 in bits 4 to 7.
<table>
<tr><th>BIT</th><th>Description</th></tr>
<tr><td>0</td><td>Reserved*</td></tr>
<tr><td>1</td><td>Reserved*</td></tr>
<tr><td>2</td><td>Reserved*</td></tr>
<tr><td>3</td><td>Reserved*</td></tr>
<tr><td>4</td><td><Channel ID value></td></tr>
<tr><td>5</td><td><Channel ID value></td></tr>
<tr><td>6</td><td><Channel ID value></td></tr>
<tr><td>7</td><td><Channel ID value></td></tr>
</table>
This parameter is ignored by BIS U-62_-xxx-111_ US models.</td><td><Partially Reserved></td></tr>
</table>

<table>
<tr><th>Response Packet Element</th><th>Content</th></tr>
<tr><td>UHF Configuration Byte 9
Byte 9 is partially reserved according to the table below. This byte permits the user to enable/disable the Choose Nearest One property.
If the Choose Nearest One property is disabled (bit = 0), an error response is generated whenever a single-tag read/write command is executed in a multi-tag environment.
If the Choose Nearest One property is enabled (bit = 1), the read/write command is executed on the tag with the strongest signal.
<table><tr><th>BIT</th><th>Description</th></tr><tr><td>0</td><td>Reserved*</td></tr><tr><td>1</td><td><Choose Nearest One></td></tr><tr><td>2</td><td>Reserved*</td></tr><tr><td>3</td><td>Reserved*</td></tr><tr><td>4</td><td>Reserved*</td></tr><tr><td>5</td><td>Reserved*</td></tr><tr><td>6</td><td>Reserved*</td></tr><tr><td>7</td><td>Reserved*</td></tr></table></td><td><Partially Reserved></td></tr>
<tr><td>Checksum:</td><td>optional</td></tr>
<tr><td>Command Terminator:</td><td>0x03</td></tr>
</table>

NOTE

** Leave the default value retrieved through the 0xC1 "Get UHF Controller Configuration" command.*

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0xC0**
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0XC1: GET UHF CONTROLLER CONFIGURATION

Command 0xC1 - Description

The *Get UHF Controller Configuration Command* is used to retrieve the configuration settings stored in the UHF-Series controller's flash memory. These are the same values that are set with the *Set UHF Controller Configuration Command*.

Command 0xC1 - ABx Fast Example

This example retrieves the stored configuration settings from the flash memory of the UHF controller.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0001
Command ID:	**0xC1**
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x000B
Command Echo:	**0xC1**
UHF Configuration Bytes 1 & 2: two-byte integer These two bytes represent the **Reader Output Power** (value from 0 to 500 mW).	0x01F4
UHF Configuration Byte 3 **UHF Configuration Byte 4**	
UHF Configuration Byte 5 **UHF Configuration Byte 6**	
UHF Configuration Byte 7 **UHF Configuration Byte 8**	
UHF Configuration Byte 9	
Checksum:	optional
Command Terminator:	0x03

ABX FAST COMMAND 0XC2: READ EPC CODE

Command 0xC2 - Description

The *Read EPC Code Command* instructs the controller to retrieve the EPC memory area of a single UHF Class 1 Gen 2 RFID tag memory.

Command 0xC2 - ABx Fast Example

This example instructs the controller to read the EPC memory from a tag. A *Timeout Value* of 2 seconds (*0x07D0 = 2000 x one-millisecond increments*) is set for the completion of the command.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0003
Command ID:	**0xC2**
Timeout Value: two-byte integer in ms	0x07D0
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller (Tag Found)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x000D
Command Echo:	**0xC2**
Read Data Byte 1:	<D1>
Read Data Byte 2:	<D2>
. . .	
Read Data Byte 12:	<D12>
Checksum:	optional
Command Terminator:	0x03

Response from Controller (Tag Not Found)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0002
Error Flag:	**0xFF**
Error Code	0x07
Checksum:	optional
Command Terminator:	0x03

See par. EPC Class 1 Gen 2 Tag Memory Structure in Appendix for details.

ABX FAST COMMAND 0XC3: WRITE EPC CODE

Command 0xC3 - Description

The *Write EPC Code Command* instructs the controller to write the EPC memory area of a single UHF Class 1 Gen 2 RFID tag memory.

Command 0xC3 - ABx Fast Example

This example instructs the controller to write the specified bytes in the EPC memory of a tag. A *Timeout Value* of 2 seconds (*0x07D0 = 2000 x one-millisecond increments*) is set for the completion of this command.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x000F
Command ID:	**0xC3**
Write Data Byte 1:	<D1>
Write Data Byte 2:	<D2>
. . .	
Write Data Byte 12:	<D12>
Timeout Value: two-byte integer in ms	0x07D0
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0001
Command Echo:	**0xC3**
Checksum:	optional
Command Terminator:	0x03

See par. EPC Class 1 Gen 2 Tag Memory Structure in Appendix for details.

ABX FAST COMMAND 0XC4: MULTI-TAG READ EPC CODE

Command 0xC4 - Description

The *Multi-Tag Read EPC Code Command* is used to retrieve the EPC data from all tags within RF range. A final termination packet is sent when the *Timeout Value* expires.

Command 0xC4 - ABx Fast Example

This example instructs the controller to read the EPC data from each tag in range. A *Timeout Value* of 3 seconds (*0x0BB8 = 3000 x 1msec increments*) is set for the completion of the command.

Command from Host

Command Packet Element	Content
Command Header:	0x02, 0x02
Command Size:	0x0003
Command ID:	**0xC4**
Timeout Value: two-byte integer in ms	0x0BB8
Checksum: one-byte	optional
Command Terminator:	0x03

Response from Controller (for Each Tag Found)

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x000D
Command Echo:	**0xC4**
Read Data Byte 1:	<D1>
Read Data Byte 2:	<D2>
. . .	
Read Data Byte 12:	<D12>
Checksum:	optional
Command Terminator:	0x03

Final Response Packet

Response Packet Element	Content
Response Header:	0x02, 0x02
Response Size:	0x0003
Final Termination Packet ID:	**0xFF**
Number of Tags Read	<N tags>
Status: 0x00 = OK, any other value = error	0x00
Checksum:	optional
Command Terminator:	0x03

See par. EPC Class 1 Gen 2 Tag Memory Structure in Appendix for details.

A REFERENCES

EPC CLASS 1 GEN 2 TAG MEMORY STRUCTURE

The memory in Balluff's EPC Class 1 Gen 2 tag *UHF-G2-525xx* is organized in **three** areas:

Name	Description	Size
EPC	EPC memory according to the EPCglobal Standard	96-bit (12 bytes)
TID	Read Only Unique identifier	64-bits (8 bytes)
USER	User memory	512-bit (64 bytes)

EPC

EPC is a numbering scheme that allows assignment of a unique identifier to any physical object. It can be regarded as the next generation of Universal Product Code (UPC), which is used on most products today.

EPC enables the means to assign a unique identifier to each item, thus allowing every item to be uniquely identified.

To have more details on the structure of the EPC memory area please consult:

- *EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 Mhz – 960 Mhz, Version 1.1.0 (December 17, 2005)*

In our UHF-G2-525xx tag this memory area is pre-programmed with the TID unique identifier and padded with zeroes. The user can change that but it's important to note that only tags with **different** EPC codes will be discriminated in a multi-tag reading environment.

TID

This is a read-only area that holds a unique tag identifier number. This area can be accessed using the common ABx Fast Read ID commands.

USER

This is the normal data area that can be accessed using the common ABx Fast Read and Write commands.

NOTE

The fastest access memory is the EPC area. For applications where speed is important the use of this memory is recommended.

www.balluff.com

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Germany
Phone +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de
www.balluff.com

Nr. 898325 E · Edition 1308; Subject to modification.