Jogtek Program introduction

Introduction

The purpose of this document is to help guide the user in setting up and running. It will help to describe some of the features and capabilities of the part and how to control its individual functions as defined by the standards. This document is specific to the multi-protocol reader. Please, see the table below which shows the standards that are supported by each part. The reader can be supported with this document.

	ISO15693	**ISO14443A**	**ISO14443B**
Support	Yes	Yes	Yes

The multi-protocol reader demonstration system supports the **ISO15693, ISO14443A,** and **ISO14443B.** The Command is help aid the user in the development process. The device incorporates and analog front end, protocol handling, framing, error checking and multiple integrated voltage regulators with other features that allow the reader to be customized/configurable for the end application.

Interface:

The demonstration board should be connected to a (USB or RS-232) PC port configured to the following settings:

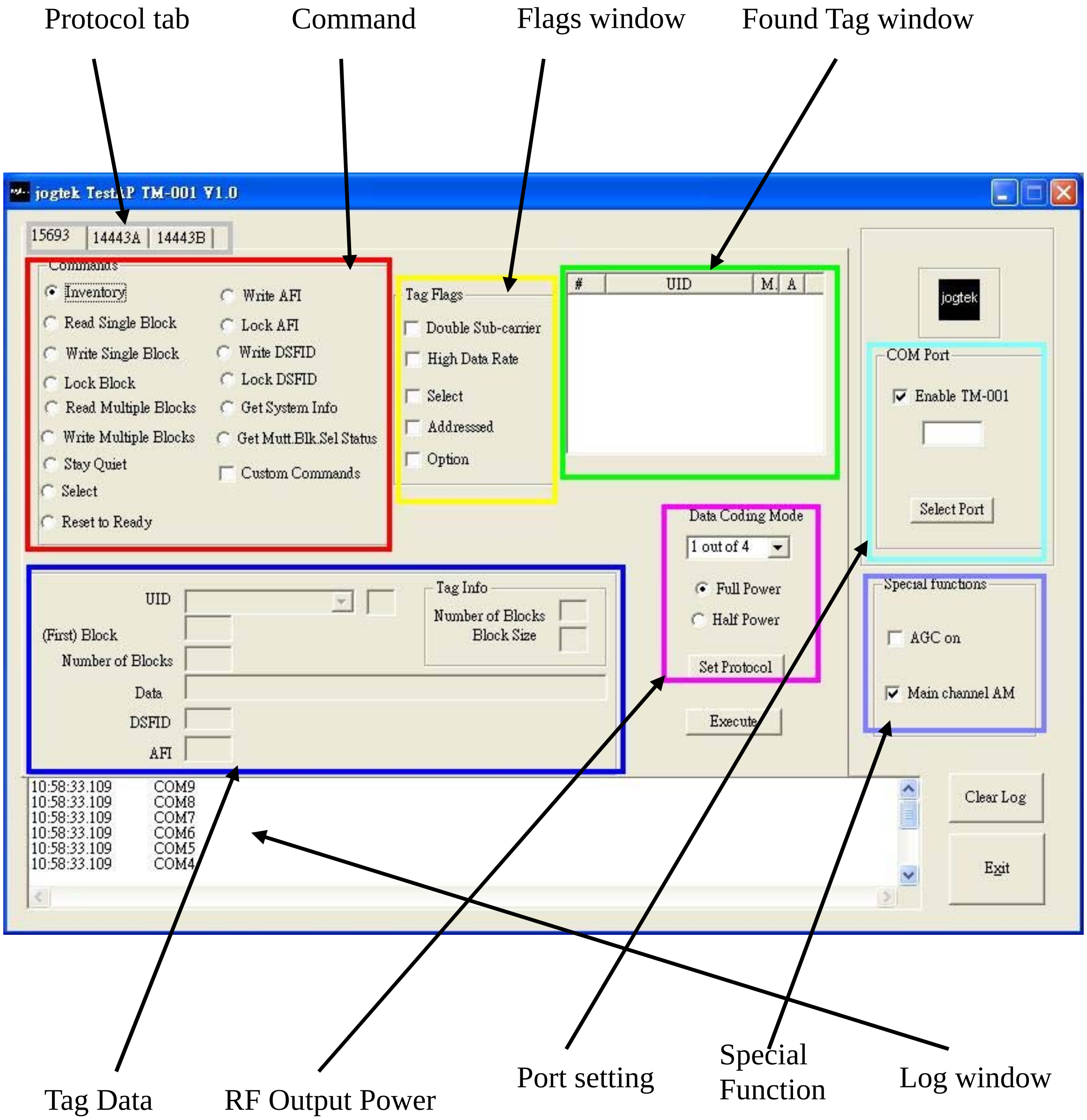
baud rate	**data bits**	**stop bit**	**parity**	**flow control**
115200	**8**	**1**	**no**	**no**

Power Supply*:*

A 5V DC power supply has to be connected to the power input connector.(Note: It is recommended that the user do a power on reset by depressing the mechanical switch situated close to the microcontroller upon power up. All commands in the ISO14443A (layer 3), and ISO15693 are supported. For the ISO14443B standard, only the basic commands are supported. This operation configures the reader to the desired standard.

Graphical User Interface:

ISO15693:



● Inventory

Data Coding Mode: 1 out of 4

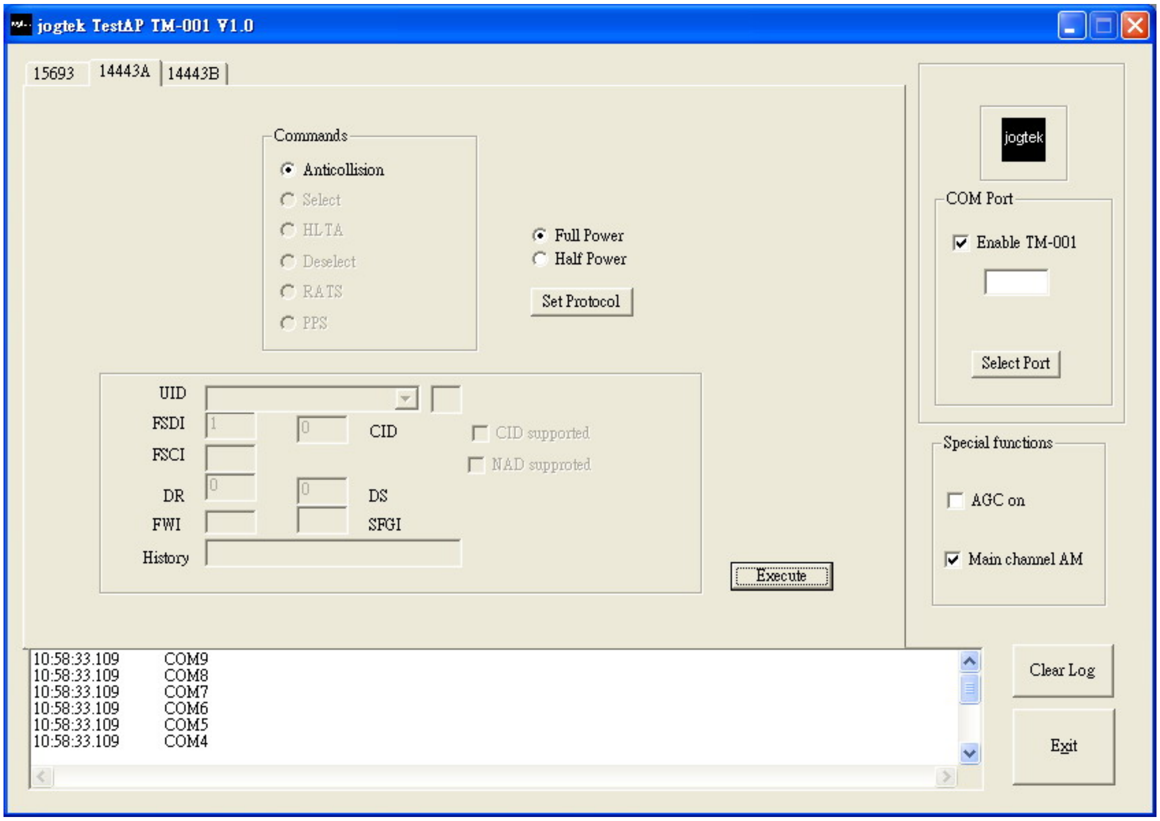
● Full Power

☑ Enable TM-001

☑ Main channel AM

```
10:58:33.109    COM9
10:58:33.109    COM8
10:58:33.109    COM7
10:58:33.109    COM6
10:58:33.109    COM5
10:58:33.109    COM4
```

ISO14443A



Commands
- ● Anticollision
- ○ Select
- ○ HLTA
- ○ Deselect
- ○ RATS
- ○ PPS

● Full Power
○ Half Power

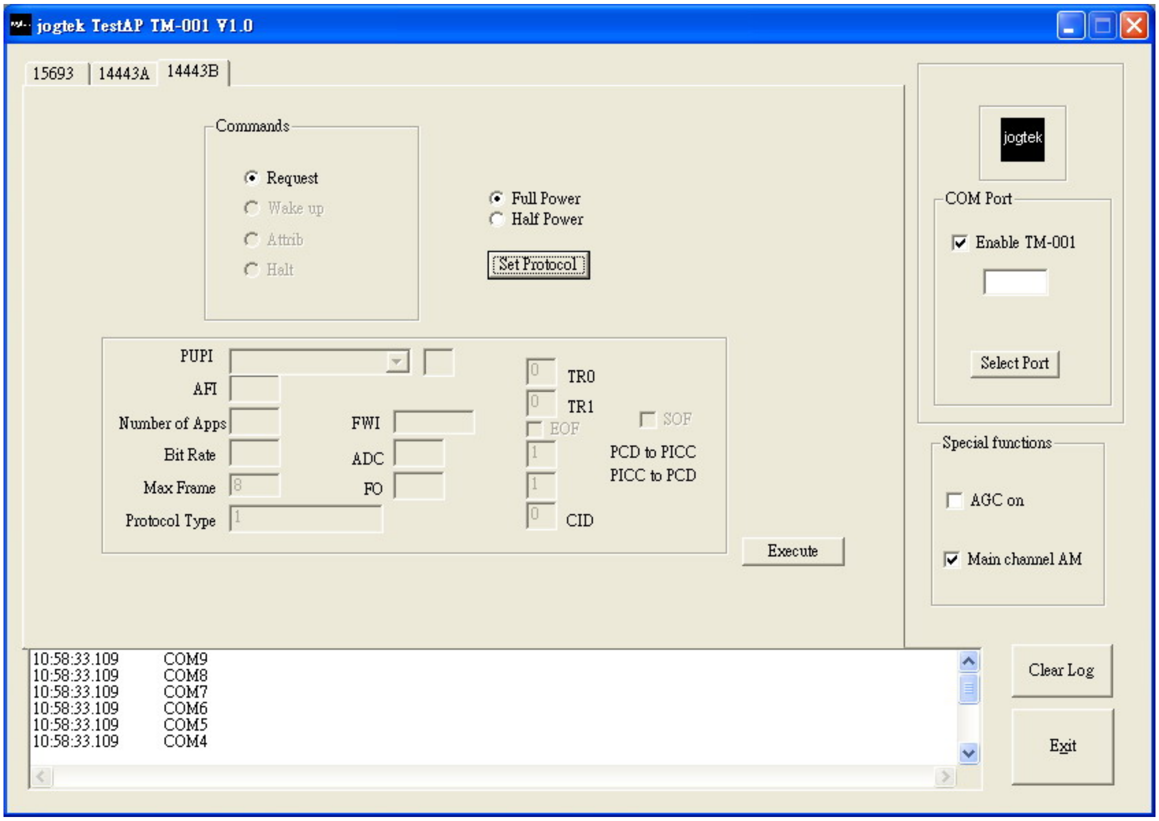
UID
FSDI 1
CID 0
□ CID supported
□ NAD supproted
FSCI
DR 0
DS 0
FWI
SFGI
History

☑ Enable TM-001

Special functions
□ AGC on
☑ Main channel AM

```
10:58:33.109        COM9
10:58:33.109        COM8
10:58:33.109        COM7
10:58:33.109        COM6
10:58:33.109        COM5
10:58:33.109        COM4
```

ISO14443B



Commands
- ● Request
- ○ Wake up
- ○ Attrib
- ○ Halt

● Full Power
○ Half Power

PUPI
AFI
Number of Apps
FWI
Bit Rate
ADC
Max Frame 8
FO
Protocol Type 1
TR0 0
TR1 0
□ EOF
□ SOF
PCD to PICC 1
PICC to PCD 1
CID 0

☑ Enable TM-001

Special functions
□ AGC on
☑ Main channel AM

```
10:58:33.109        COM9
10:58:33.109        COM8
10:58:33.109        COM7
10:58:33.109        COM6
10:58:33.109        COM5
10:58:33.109        COM4
```

Protocol Tabs Window:

The protocol tabs window selects between tag protocols and program functions. Available options are:

(ISO/IEC) 15693 – vicinity cards, which has slightly longer read range
(ISO/IEC) 14443A – proximity cards
(ISO/IEC) 14443B – proximity cards

Flags window:

This window allows the user to set flags for the 15693 and Tag-it protocols. Different flags may be available for different commands – see Appendix A.1. The tag window automatically updates available flags depending on the request chosen.

Tag Data window:

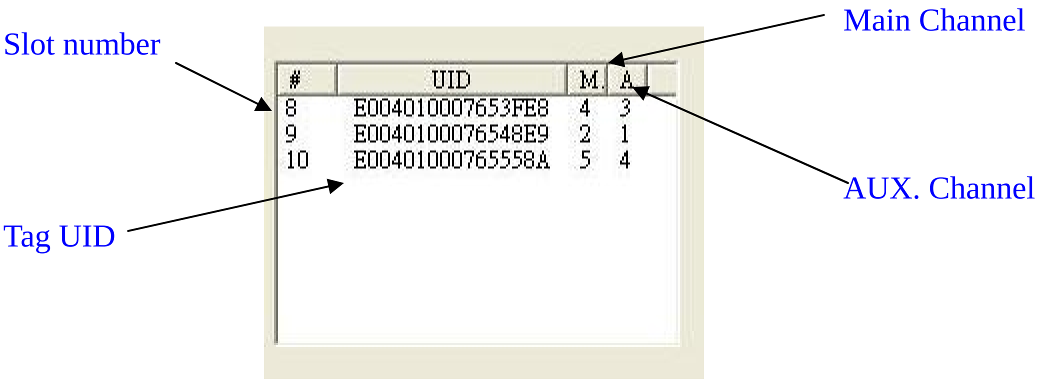
The *Tag Data* window is where the user enters addresses, data, number of bits, and other information required by certain commands. Checking certain flags in the *Flag* window may activate more fields for data entry.

Found Tag window:

The Found Tag field displays the slot number, UID and the Found Tag values of the corresponding tag. If there was a collision and the reader performed a second anticollision procedure, the slot numbers are indicated with an additional character:

A = second procedure
B = third procedure

And so on the main channel, which is AM, is used as the primary one, and PM is the auxiliary channel. The Found Tag maximum value is 7 and minimum value is 0. The corresponding Found Tag values depend on the system design (antenna + reader), and the levels can vary based on the quality of the reception. The specifics of the corresponding input voltage levels to Found Tag levels are defined in the product data sheet.



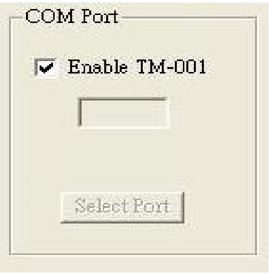
#	UID	M.	A.
8	E004010007653FE8	4	3
9	E0040100076548E9	2	1
10	E00401000765558A	5	4

In the preceding example, one can see that for the tag in slot #10, the Found Tag value is higher for the AM (main) channel, whereas for the tags in slots #8and #9, the Found Tag value is higher for the PM (auxiliary) channel.

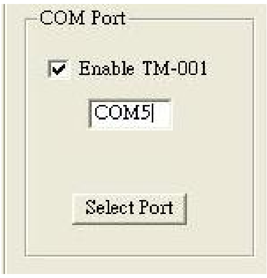
Port setting:

When start the program, the program run search COM port at first. The User will see the message in the Log window. If the message is "can't found the COM port ", you should see the COM port block that is enabled. The user could type COM number in that block and press the "select port" button by oneself. After that, the program can connect to the reader.

Auto gets port **Type by oneself**


COM Port

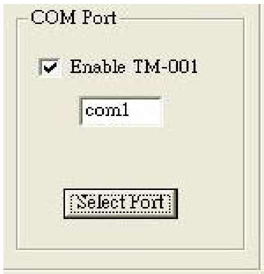
☑ Enable TM-001


COM Port

☑ Enable TM-001

COM5

Or


COM Port

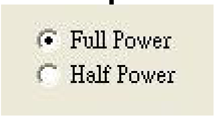
☑ Enable TM-001

com1

RF Output Power:

The RF output power selection enables the user to switch between FULL power (200mW) and half power (100mW), however the antenna matching circuit is tuned to operate with FULL power selection and performance will not be optimal in HALF power selection.

Select power

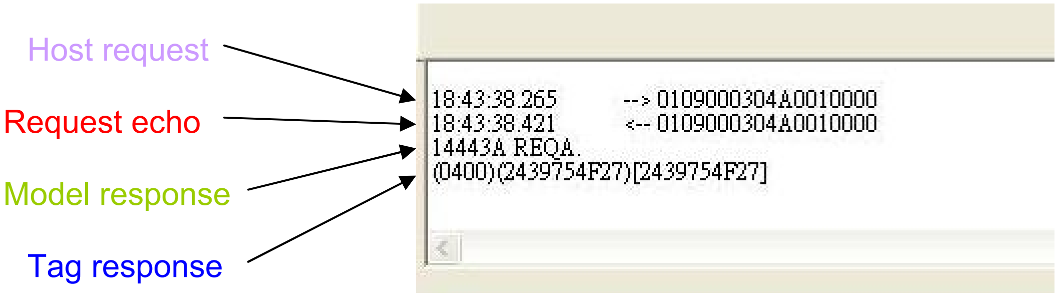

● Full Power

○ Half Power

Log window

In the log window, that shows all communication frames from the host to reader board. (This information is also stored in the "*rfid-reader.log"* file which can be opened by a normal text editor (Notepad).) The model echoes the received request back to the host. This enables the user to control the reader from a normal terminal emulator like a *Hyper Terminal*.

There are two side messages in the log window. First side, the received data (tag response, register content) is always in brackets to distinguish it from the host to reader data exchange. The other side, the log window is used to display all the messages and data sent commands from host to reader board. The log file is typically generated in the same directory as the executable for the GUI.



```
18:43:38.265        --> 0109000304A0010000
18:43:38.421        <-- 0109000304A0010000
14443A REQA.
(0400)(2439754F27)[2439754F27]
```

Other functions:

Other functions on the main Test AP control panel are:

Set protocol, which configures the program for the selected protocol once the protocol tab, has been selected.

Execute button, which runs the selected protocol

Data coding mode, which is used in conjunction with the 15693 protocol

Host to Reader Protocol

The communication is organized into frames from host to reader. Each frame is consisted of 6 fields:

SOF(0X01)	Number of bytes	0x00	0x0304	command+ parameters	EOF(0x0000)

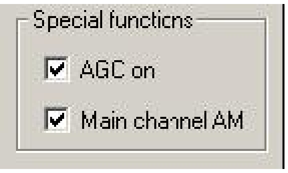
The communication starts with SOF (0x01). The second byte defines the number of bytes in the frame including SOF. The third byte should be kept at 0x00, fourth byte at 0x03 and the fifth byte at 0x04. The sixth byte is the command code, which is followed by parameters or data. The communication ends with 2 bytes of 0x00.

Meaning	**command**	**parameters**	**Example**
Write single register	0x10	Address, data, address, data….	01 0A 00 03 04 **10 15 67** 0000
Write continuous	0x11	Address, data, data…	01 0C 00 03 04 **11 13 67 46 A4** 0000
Read single register	0x12	Address, data, data…	01 0B 00 03 04 13 05 03 0000
Read continuous	0x13	NR. Of bytes to	01 0A 00 03 04

		read, start address	**14 06 01 00** 0000
Meaning	**command**	**parameters**	**Example**
Inventory (ISO 15693)	0x14	FIFO data	01 0B 00 03 04 **14 06 01 00** 0000
Direct command	0x15	Direct command code	01 09 00 03 04 **15 0F** 0000
Write raw	0x16	Data or commands	01 10 00 03 04 **16 91 3D 00 40 AA BB CC DD** 0000
Request command ISO15693, Tag-it, 14443B Halt	0X18	Flags, Command code, data… (as specified in ISO and Tag-it)	01 0B 00 03 04 **18 06 20 01** 0000
SID poll(Tag-it)	0x34	Flags, command code, mask (as specified in Tag-it)	01 0B 00 03 04 **34 00 50 00** 0000
Direct mode	0x0F	/	01 08 00 03 04 **0F** 0000
AGC selection	0xF0	0x00-AGC enable 0xFF-AGC disable	01 09 00 03 04 **F0 FF** 0000
AM/PM input selection	0XF1	0X00-FM input 0XFF-AM input	01 09 00 03 04 **F1 00** 0000
TM-001 enable/disable	0x03	0x00-reader enable 0XFF-reader disable	01 09 00 **03 FF** 0000
REQB(14443B)	0xB0	/	01 08 00 03 04 **B0** 0000
REQA(14443A)	0XA0	/	01 08 03 04 **A0** 0000
Select (14443A)	0XA2	CID	01 0D 00 03 04 **A2 11 22 33 44 44** 0000

Special Functions

These functions allow the user to enable the Automatic Gain Control (AGC) feature and also allows for the switching from the AM to PM input.



Special functions

☑ AGC on

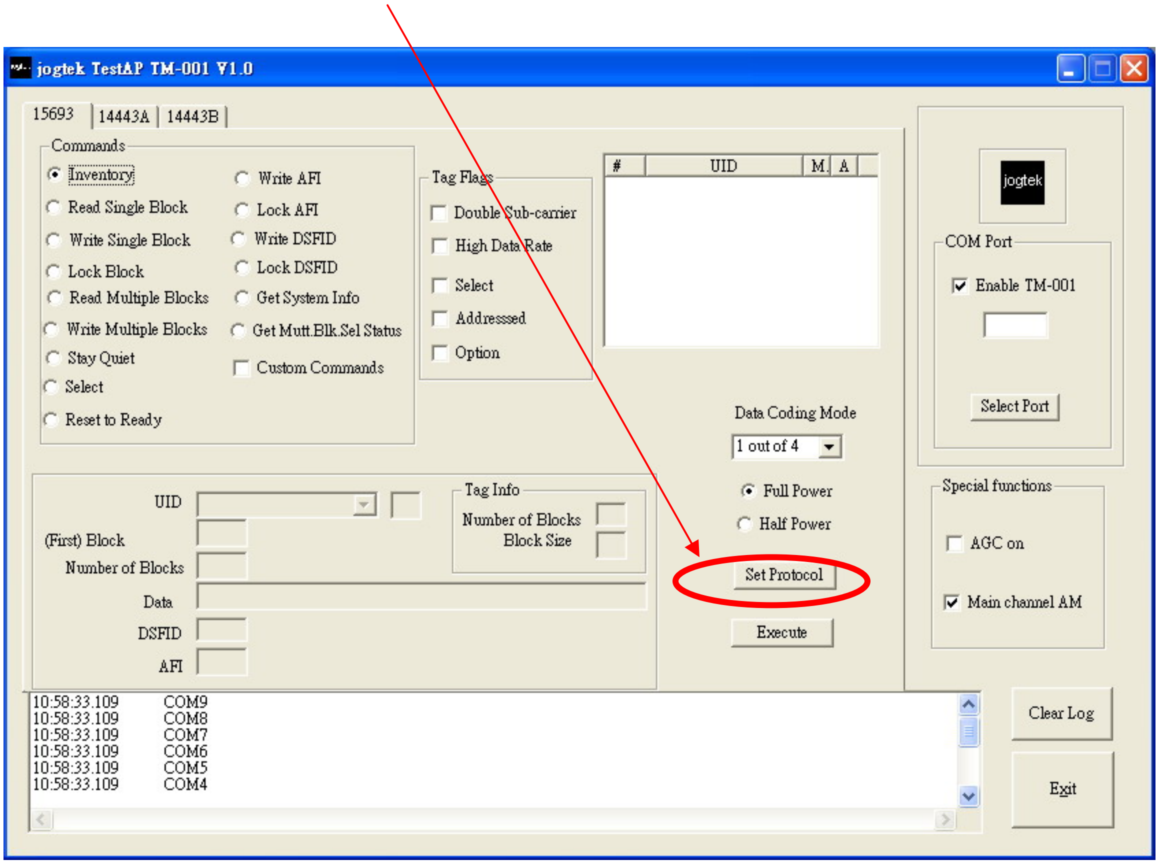
☑ Main channel AM

The AGC is turned off after the power on reset (POR) and can be enabled when desired (Especially in noisy environments). By default the input channel is AM and can be switched to PM if the RSSI value for the PM channel is higher than the AM.

Set Protocol:

Selecting a protocol with a protocol tab does not automatically set the program to that protocol. The user must manually click on the *Set Protocol* button:

“Set Protocol “Button



● Inventory

Data Coding Mode: 1 out of 4

● Full Power

☑ Enable TM-001

☑ Main channel AM

```
10:58:33.109        COM9
10:58:33.109        COM8
10:58:33.109        COM7
10:58:33.109        COM6
10:58:33.109        COM5
10:58:33.109        COM4
```

When the *Set Protocol* button is pressed, the software sets the parameters for the corresponding standard. These settings are also available through the register address space in the model. The following registers are currently implemented in the RFID ASIC: *Modulator and CL_SYS control (09), RX special setting (0A), and Regulator control (0B)*. In the demonstration board, this registers are set through an SPI interface implemented in the RFID ASIC.
The following table shows the settings for the different standards.

Standard	0x09	0x0A	0x0B
ISO15693	0x29	0x40	0x07
Tag-it	0x29	0x40	0x07

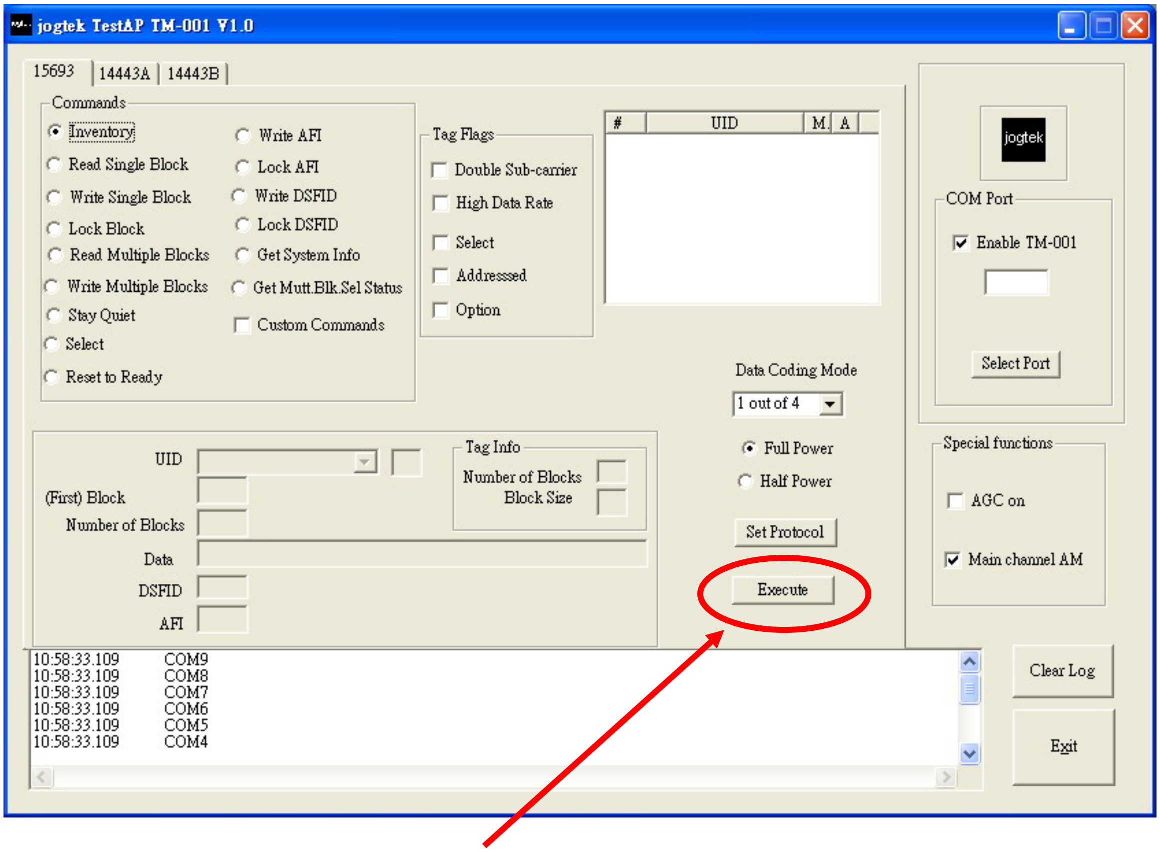
ISO 14443B	0x2D	0x00	0x07

These settings can be written only by pressing the *Set Protocol* button. Once the protocol has been set, the user can select one of the commands in the *Command* window. These commands are mutually exclusive – only one command can be executed at a time.

ISO/IEO 15693 Protocol:

This section describes commands for the 15693 protocol. Once a command has been selected by highlighting the command in the *Commands* window, the user should set any flag that is needed, and if appropriate, enter data in the *Tag Data* window.

After the parameters are set up, the user should click on the *Execute* button:



```
10:58:33.109     COM9
10:58:33.109     COM8
10:58:33.109     COM7
10:58:33.109     COM6
10:58:33.109     COM5
10:58:33.109     COM4
```

"Execute" Button

The preceding screen shows the results of an *Inventory Command.* When a command is executed, the program first runs an anticollision sequence to insure that only one tag is read at a time (if multiple tags are in range). It then sends the formatted command, framed in a Start of Frame (SOF) and an End of Frame (EOF).

Tag States:

A tag can be in only one of the four following states at any given time:
Power off – the tag cannot respond to the reader.

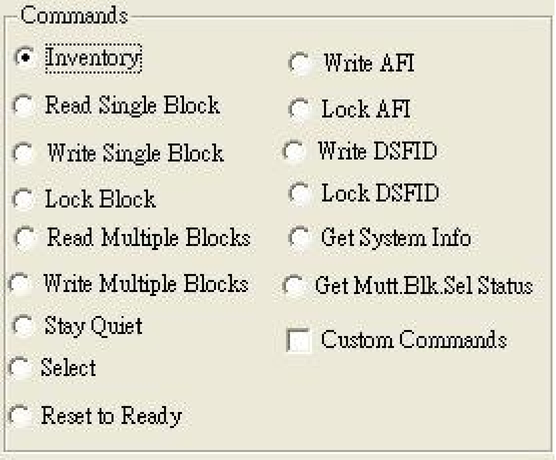
Ready – the tag is in the Ready state when it is activated by the reader. It processes any command where the select flag is not set.

Quiet – When in the quiet state, the tag processes any command where the Inventory flag is not set and where the Address flag is set.

Selected – Only a tag in the selected state processes commands having the Select flag set. The intention of the select flag is that only one tag should be in the *selected* state at a time. The selected state is an optional tag feature.

The transition between the different states is defined by commands below.

Inventory:



The Inventory command is used to acquire the unique IDs (UID) of ISO15693 tags in the read zone. Two inventory methods supported are slotted and non-slotted. A non-slotted request allows all transponders in the read zone to reply to a single command. In cases where more than one tag is present, such a request would cause a data collision and yield no discernable response. A slotted inventory sequence decreases the likelihood of a data collision by forcing compliant transponders to respond in 1 of 16 slots based on a portion of their UIDs. To perform a slotted sequence, the *Slot Marker/End-of-Frame* request is used in conjunction with this command. Any collision that does occur in a slotted sequence can be further arbitrated by using the anticollision mask in an algorithm similar to that outlined in the ISO15693 standard.

To inventory a tag, the user should follow those:

1. Click the button for *Inventory* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flags* window.
3. *Execute* the command.

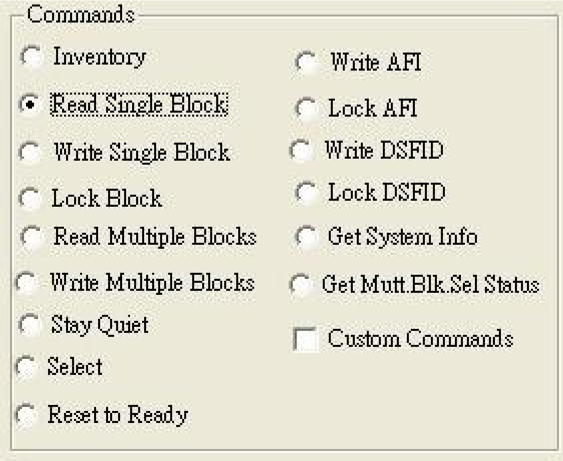
Request Packet:

Field	Length	Value Range	Description
Cmd1	1	04	Entity ID
Cmd2	1	62	Inventory
IsOneSlot	1	00-FF	0:initiate 16-slot request
AfiFlg	1	00-FF	(other):transmit AfiVal in response
AfiVal	0/1	00-FF	Application family ID, sent in if previous field is nonzero
MskLen	1	00-08	Length of next field
MskVal	0-8	00-FF	Anticollison mask per ISO15693-3

Response packet:

Field	Length	Value Range	Description
Cmd1	1	04	ISO15693 Entity ID
Cmd2	1	62	Inventory
Status	1	00-FF	Standard Error Code
InvReply	0/10	00-FF	Byte 1:Inventory Response Flags
		00-FF	Byte2:DSFID
		00-FF	00-FF Byte3-10:Unique ID

Read Single Block:



Commands

○ Inventory
● Read Single Block
○ Write Single Block
○ Lock Block
○ Read Multiple Blocks
○ Write Multiple Blocks
○ Stay Quiet
○ Select
○ Reset to Ready
○ Write AFI
○ Lock AFI
○ Write DSFID
○ Lock DSFID
○ Get System Info
○ Get Mutt.Blk.Sel Status
□ Custom Commands

The Read Single Block command gets the data from one memory block of the responding tag. In addition to this data, a Block Security Status byte can be requested. This byte shows the write-protection of the block specified [e.g., unlocked, (user/factory) locked, etc.].

To read a single block, the user should follow those:

1. Click the button for *Read Single Block* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flag*s window.
3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. Enter two digits corresponding to the block number in the *(First) Block Number* field in the *Tag Data* window.
5. *Execute* the command.

When the Read Single Block command is executed, the program reads the requested block from the tag and sends back its value.

Request packet:

Field	Length	Value Range	Description
Cmd1	1	04	Entity ID
Cmd2	1	65	Read single block
IsSelect Msg	1	00-FF	0:Do not set Select flag
			(other):Request block security status
ReqSecurity	1	00-FF	0:No block security status
			(other):Request block security status
BikNum	1	00-FF	Specifies block that is to be read
UID	0/8	00-FF	Unique ID of tag

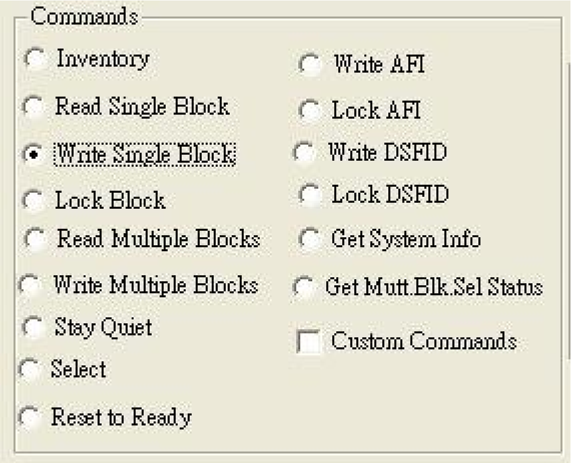
(1) If UID field is not present, all tags in the read zone are addressed.

Response Packet:

Field	Length	Value Range	Description
Cmd1	1	04	ISO15693 entity ID
Cmd2	1	65	Read single block
Status	1	00-FF	Standard error codes
StdResp	n	00-FF	Byte 1-n:ISO15693 response flags
			Byte 2-n: Read data
-OR-			
SecureResp	n	00-FF	Byte1:ISO15693 response flags
			Byte2:Block security status
			Byte3-n:Read data
-OR-			
ErrorResp	2	00-FF	Byte1:ISO15693 response flags
			Byte2:Error code
-OR-			
NoData	0	-	Byte1:ISO15693 response flags

(1) Response when tag responds with data read from its memory
(2) The value of *n* varies with the block size for the specific tag read
(3) Response when tag responds with block security status and memory data
(4) Response when error flag is set in tag reply
(5) No data returned due to condition described in <*Status*> field

Write Single Block:



The Write Single Block request writes data to one memory block of the addressed tag(s). In order to successfully write data, the host must know the size of the memory block of the tag. This information is available through the *Get System Information* request, if supported by the tag. A corrupted response or lack of response does not necessarily indicate a failure to perform the write operation. Additionally, multiple transponders may process a nonaddressed request.

To write a single block, the user should:
1. Click the button for *Write Single Block* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flags* window.
3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. Enter two digits corresponding to the block number in the *(First) Block Number* field in the *Tag Data* window.
5. Enter 8 hexadecimal digits corresponding to the data to be written in the *Data* field in the *Tag Data* window.
6. *Execute* the command.

Request Packet:

Field	Length	Value Range	Description
Cmd1	1	04	Entity ID
Cmd2	1	66	Write single block
IsSelect Msg	1	00-FF	0:n Do not set select flag
			(other):set select flag in response
Resp Type	1	00-FF	0:Asynchronous reply
			(other): Polled reply(prog. burst)
BikNum	1	00-FF	Specifies block that is to be written
BikBytes	1	00-FF	Length of next field
BikData	0-32	00-FF	Data to be written to specified block
UID	0-8	00-FF	Unique ID of tag

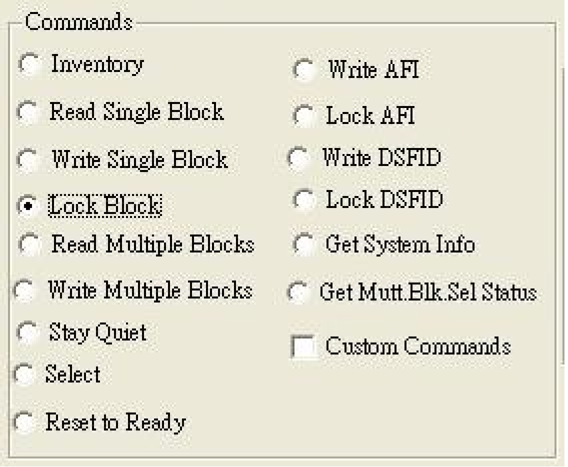
(2) Length specified by previous field.
(3) If UID field is not present, all tags in the read zone are addressed.

Response Packet:

Field	Length	Value Range	Description
Cmd1	1	04	ISO15693entity ID
Cmd2	1	66	write single block
status	1	00-FF	Standard Error codes
AckResp	1	00-FF	Byte1: ISO15693 response flag

-OR-			
ErrorResp	2	00-FF	Byte1:ISO15693 response flags Byte2:Error Code
-OR-			
NoData	0	-	Byte1:ISO15693 response flags

Look Block:



The Lock Block command write-protects one memory block of the addressed tag(s). A corrupted response or lack of response does not necessarily indicate a failure to perform the lock operation. Additionally, multiple transponders may process a nonaddressed request.

To lock a block, the user should:
1.Click the button for *Lock Block* in the *Command* window.
2.Click on any flags that must be set in the *Tag Flags* window.
3.Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. Enter two digits corresponding to the block number in the *(First) Block Number* field in the *Tag Data* window.
5. *Execute* the command.

Request Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	Entity ID

Cmd2	1	67	Lock block
IsSelect Msg	1	00–FF	0: Do not set Select flag
			(Other): Set Select flag in response
RespType(1)	1	00–FF	0: Asynchronous reply
			(Other): Polled reply (prog. burst)
BlkNum	1	00–FF	Specifies block that is to be written
UID(2)	0–8	00–FF	Unique ID of tag

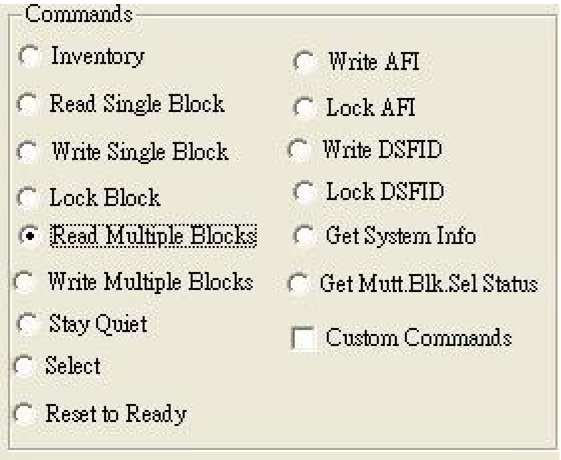
(1) A nonzero value is required for Tag-it HF-I tags
(2) If UID field is not present, all tags in the read zone are addressed.

Response Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	ISO15693 entity ID
Cmd2	1	67	Lock block
Status	1	00–FF	Standard error codes
AckResp(1)	1	00–FF	Byte 1: ISO15693 response flags
— OR —			
ErrorResp(2)	2	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Eror Code
— OR —			
NoData(3)	0	—	Byte 1: ISO15693 response flags

(1) Contents with tag reply *Success*
(2) Contents with tag reply *Error*
(3) No data returned due to condition described in <*Status*> field

Read Multiple Blocks



Commands

○ Inventory
○ Read Single Block
○ Write Single Block
○ Lock Block
● Read Multiple Blocks
○ Write Multiple Blocks
○ Stay Quiet
○ Select
○ Reset to Ready
○ Write AFI
○ Lock AFI
○ Write DSFID
○ Lock DSFID
○ Get System Info
○ Get Mutt.Blk.Sel Status
□ Custom Commands

The Read Multiple Blocks command gets the data from multiple memory blocks of the responding tag. In addition to this data, a Block Security Status byte can be requested for each block. This byte shows the write-protection of the block specified [e.g., unlocked, (user/factory) locked, etc.].

To read multiple a blocks, the user should:

1. Click the button for *Read Multiple Blocks* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flags* window.
3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. Enter two digits corresponding to the starting block number in the *(First) Block Number* field in the *Tag Data* window. The blocks are numbered from *00* to *FF* (0 to 255).
5. Enter two digits corresponding to the number of blocks to be written in the *Number of Blocks* field in the *Tag Data* window. The number of blocks in the request is one less than the number of blocks that the tag returns in its response. E.g., a value of *06* in the *Number of Blocks* field requests to read 7 blocks. A value of *00* requests to read a single block.
6. *Execute* the command.

Request Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	Entity ID
Cmd2	1	68	Read multiple blocks
IsSelect Msg	1	00–FF	0: Do not set Select flag
			(Other): Set Select flag in response
ReqSecurity	1	00–FF	0: No block security status
			(Other): Request block

			security status
StartBlk	1	00–FF	Specifies first block that is to be read
NumBlks	1	00–FF	Number of blocks to be read after 1st block
UID	0-8	00-FF	Unique ID of tag

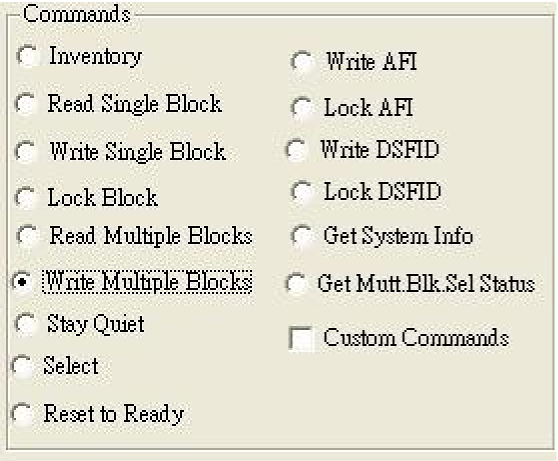
(1) If UID field is not present, all tags in the read zone are addressed.

Response Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	ISO15693 Entity ID
Cmd2	1	68	Read multiple blocks
Status	1	00–FF	Standard error codes
StdResp(1)	n(2)	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2 – n(2): Concatenated block read data
— OR —			
SecureResp(3)	n(4)	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Block sescurity status
		00–FF	Byte 3 – s(5): Read data
		Previous two fields repeated <NumBlks> times	
— OR —			
ErrorResp(6)	2	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Error code
— OR —			
NoData(7)	0	—	Byte 1: ISO15693 response flags

(1) Response when tag responds with data read from its memory
(2) n = [(<*NumBlks*> + 1) × (Tag Block Size)] +1
(3) Response when tag responds with Block Security Status and memory data
(4) n = [(<*NumBlks*> + 1) × (Tag Block Size + 1)] + 1
(5) (<*Block Read Data*> length = Tag Block Size); therefore, s = Tag Block Size + 2
(6) Response when error flag is set in tag reply
(7) No data returned due to condition described in <*Status*> field

Write Multiple Blocks:



The *Write Multiple Blocks* command writes data to multiple memory blocks of the addressed tags. In order to successfully write data, the host must know the size of the memory block of the tag. *Write Multiple Blocks* is an optional command, and may not be supported by the tag (see the following screen capture).

To write multiple blocks, the user should:

1. Click the button for *Write Multiple Blocks* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flags* window.
3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. Enter two digits corresponding to the starting block number in the *(First) Block Number* field in the *Tag Data* window. The blocks are numbered from *00* to *FF* (0 to 255).
5. Enter two digits corresponding to the number of blocks to be written in the *Number of Blocks* field in the *Tag Data* window. The number of blocks in the request is one less than the number of blocks that the tag returns in its response. E.g., a value of *06* in the *Number of Blocks* field requests to read 7 blocks. A value of *00* requests a read of a single block.
6. *Execute* the command.

Request Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	Entity ID
Cmd2	1	69	Write multiple blocks
IsSelect Msg	1	00–FF	0: Do not set Select flag
			(Other): Set Select flag in response
RespType	1	00–FF	0: Asynchronous reply

			(Other): Polled reply (prog. burst)
StartBLk		00–FF	Specifies block that is to be written
NumBlks		00–FF	No. of blocks to write after 1st block
BlkBytes	1	00–1F	Length of next field
BlkData	n(1)	00–FF	Data to be written to specified block
UID(2)	0–8	00–FF	Unique ID of tag

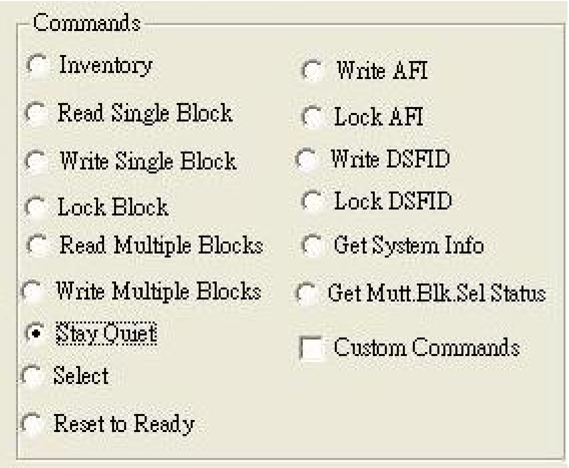
(1) n = (<NumBlks>) x <BlkBytes>
(2) If UID field is not present, all tags in the read zone are addressed.

Response Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	ISO15693 entity ID
Cmd2	1	69	Write multiple blocks
Status	1	00–FF	Standard error codes
AckResp(1)	1	00–FF	Byte 1: ISO15693 response flags
— OR —			
ErrorResp(2)	2	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Error Code
— OR —			
NoData(3)	0	—	Byte 1: ISO15693 response flags

(1) Contents with tag reply *Success*
(2) Contents with tag reply *Error*
(3) No data returned due to condition described in <*Status*> field

Stay Quiet:



The *Stay Quiet* command is used to silence a tag, preventing it from responding to any nonaddressed or inventory related commands. The tag does, however, respond to requests with matching UID. As there is no response to this request from the receiving tag, only request status and errors are reported.

To command a tag to stay quiet, the user should:
1. Click the button for *Stay Quiet* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flags* window.
3.Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. *Execute* the command.

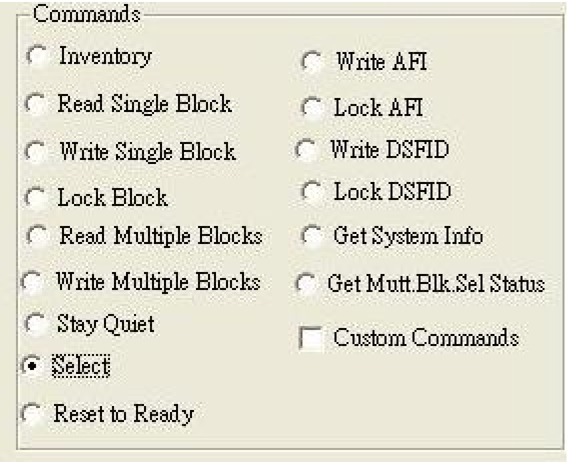
Request Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	Entity ID
Cmd2	1	64	Stay Quiet
UID	8	00–FF (ea)	Unique ID of tag to silence

Response Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	ISO15693 entity ID
Cmd2	1	69	Stay quiet
Status	1	00–FF	Standard error codes

Select



The *Select* command places the addressed tag in the *Select* state. In this state, it responds to requests with the ISO15693 *Select Flag* set. This flag is directly controlled by the <*IsSelectMsg*> field present in many ISO15693 library request messages. Any receiving tag currently in the *Select* state with UID not matching the value sent in the request command exits that state and enters the *Ready* state but does not send a reply.

To select a tag, the user should:

1. Click the button for *Select* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flags* window.
3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. *Execute* the command.

Request Packet:

Field	Length	Value Range	Description
Cmd1	1	04	Entity ID
Cmd2	1	6A	Select
UID	8	00–FF	Unique ID of tag

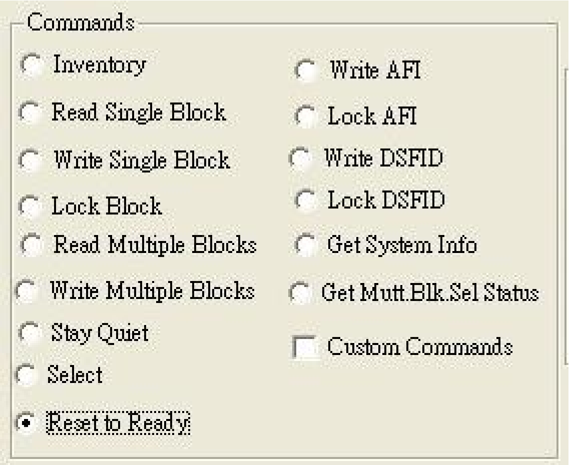
Response Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	ISO15693 entity ID
Cmd2	1	6A	Select
Status	1	00–FF	Standard error codes
AckResp(1)	1	00–FF	Byte 1: ISO15693 response Tag Flagss
— OR —			

ErrorResp(2)	2	00–FF	Byte 1: ISO15693 response Tag Flagss
		00–FF	Byte 2: Error Code
— OR —			
NoData(3)	0	—	Byte 1: ISO15693 response Tag Flagss

(1) Contents with tag reply *Success*
(2) Contents with tag reply *Error*
(3) No Data returned due to condition described in <*Status*> field

Reset To Ready:



The *Reset To Ready* command places the addressed tag in the *Ready* state. In this state, it does not respond to requests with the ISO15693 *Select Tag Flags* set, but to any nonaddressed request or request matching its UID. This command is, in effect, the complement of the *Select* command, and undoes it.

To reset a tag, the user should:

1. Click the button for *Reset to Ready* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flags* window.
3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. *Execute* the command.

Request Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	Entity ID
Cmd2	1	6B	Reset to ready
IsSelectMsg	1	00–FF	0: Do not set Select flag (Other): Set Select flag in

			response
UID(1)	8	00–FF	Unique ID of tag

(1) If UID field is not present, all tags in the read zone are addressed.

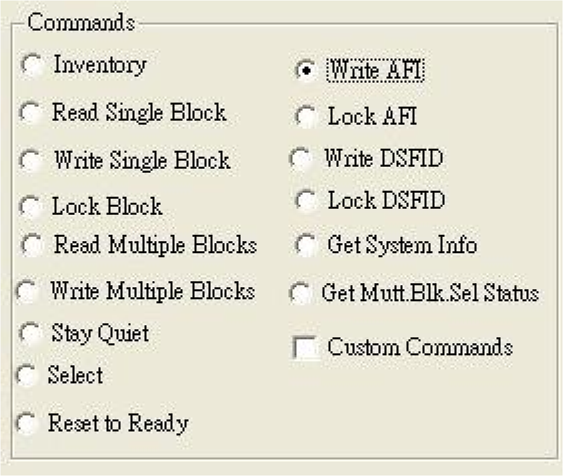
Response Packet:

Response Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	ISO15693 entity ID
Cmd2	1	6B	Reset to ready
Status	1	00–FF	Standard error codes
AckResp(1)	1	00–FF	Byte 1: ISO15693 response flags
— OR —			
ErrorResp(2)	2	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Error Code
— OR —			
NoData(3)	0	—	Byte 1: ISO15693 response flags

(1) Contents with tag reply *Success*
(2) Contents with tag reply *Error*
(3) No Data returned due to condition described in <*Status*> field

Write AFI



Commands

○ Inventory
○ Read Single Block
○ Write Single Block
○ Lock Block
○ Read Multiple Blocks
○ Write Multiple Blocks
○ Stay Quiet
○ Select
○ Reset to Ready
● Write AFI
○ Lock AFI
○ Write DSFID
○ Lock DSFID
○ Get System Info
○ Get Mutt.Blk.Sel Status
□ Custom Commands

The *Write AFI* command records a new value to the AFI register (see Section A.5 for AFI codes) of the addressed tag(s). A corrupted response or lack of response does not necessarily indicate a failure to perform the write operation. Additionally, multiple transponders may process a nonaddressed request.

To write a tag’s AFI, the user should:

1. Click the button for *Write AFI* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flags* window.
3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. Enter the desired AFI code in the *AFI* field in the *Tag Data* window.
5. *Execute* the command.

Request Packet:

Field	Length	Value Range	Description
Cmd1	1	04	Entity ID
Cmd2	1	6C	Reset to ready
IsSelectMsg	1	00–FF	0: Do not set Select flag (Other): Set Select’ flag in response
RespType(1)	1	00–FF	0: Asynchronous reply (Other): Polled reply (prog. burst)
AfiVal	1	00–FF	Application family ID
UID(2)	8	00–FF	Unique ID of tag

(1) A nonzero value is required for Tag-it tags
(2) If UID field is not present, all tags in the read zone are addressed.

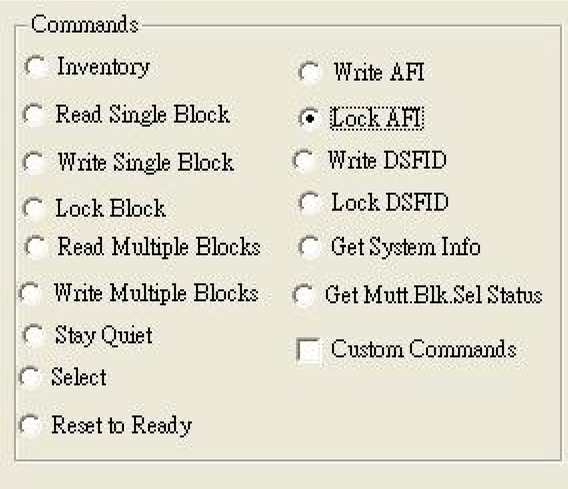
Response Packet:

Response Packet:

Field	Length	Value Range	Description
Cmd1	1	04	ISO15693 entity ID
Cmd2	1	6C	Write AFI
Status	1	00–FF	Standard error codes
AckResp(1)	1	00–FF	Byte 1: ISO15693 response flags
— OR —			
ErrorResp(2)	2	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Error Code
— OR —			
NoData(3)	0	—	Byte 1: ISO15693 response flags

(1) Contents with tag reply *Success*
(2) Contents with tag reply *Error*
(3) No data returned due to condition described in <*Status*> field

Lock AF



The *Lock AFI* command write-protects the AFI register of the addressed tag(s). A corrupted response or lack of response does not necessarily indicate a failure to perform the lock operation. Additionally, multiple transponders may process a nonaddressed request.

To a lock tag's AFI, the user should:
1. Click the button for *Lock AFI* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flags* window.
3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. Enter the desired AFI code in the *AFI* field in the *Tag Data* window.
5. *Execute* the command.

Request Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	Entity ID
Cmd2	1	6D	Lock AFI
IsSelectMsg	1	00–FF	0: Do not set Select flag (Other): Set Select flag in response
RespType(1)	1	00–FF	0: Asynchronous reply (Other): Polled reply (prog. burst)
UID(2)	0–8	00–FF	Unique ID of tag

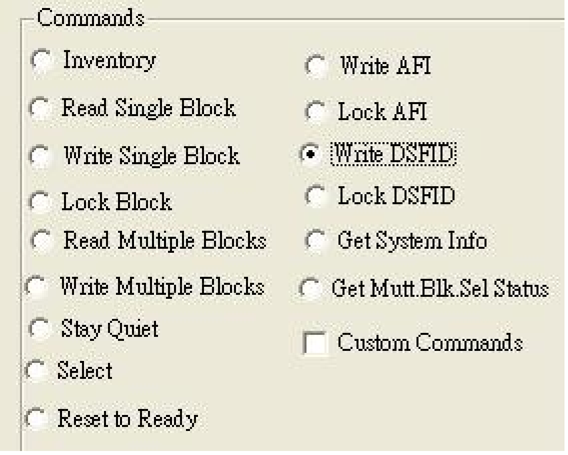
(1) A nonzero value is required for Tag-it tags.
(2) If UID field is not present, all tags in the read zone are addressed.

Response Packet:

Field	Length	Value Range	Description
Cmd1	1	04	ISO15693 entity ID
Cmd2	1	6D	Lock AFI
Status	1	00–FF	Standard error codes
AckResp(1)	1	00–FF	Byte 1: ISO15693 response flags
— OR —			
ErrorResp(2)	2	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Error Code
— OR —			
NoData(3)	0	—	Byte 1: ISO15693 response flags

(1) Contents with tag reply *Success*
(2) Contents with tag reply *Error*
(3) No data returned due to condition described in <*Status*> field

Write DSFID:



The *Write DSFID* (data storage format ID) command writes a new value in the DSFID register of the addressed tag(s). A corrupted response or lack of response does not necessarily indicate a failure to perform the write operation. Additionally, multiple transponders may process a nonaddressed request.

To write a tag’s DSFID, the user should:

1. Click the button for *Write DSFID* in the *Commands* Window.
2. Click on any flags that must be set in the *Tag Flags* window.

3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. Enter the desired DSFID code in the *DSFID* field in the *Tag Data* window.
5. *Execute* the command.

Request Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	Entity ID
Cmd2	1	6E	Write DSFID
IsSelectMsg	1	00–FF	0: Do not set Select flag (Other): Set Select flag in response
RespType(1)	1	00–FF	0: Asynchronous reply (Other): Polled reply (prog. burst)
AfiVal	1	00–FF	Application family ID
UID(2)	0–8	00–FF	Unique ID of tag

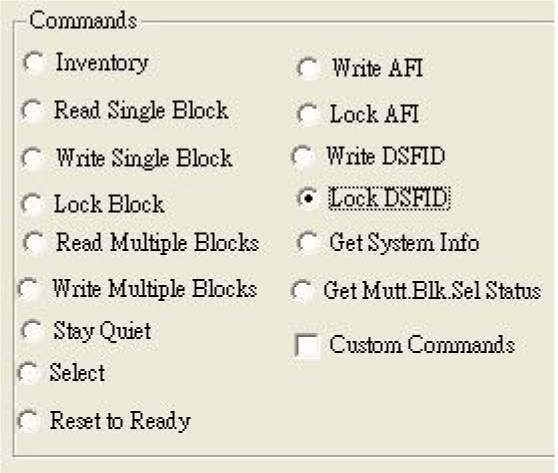
(1) A nonzero value is required for Tag-it tags.
(2) If UID field is not present, all tags in the read zone are addressed.

Response Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	ISO15693 Entity ID
Cmd2	1	6E	Write DSFID
Status	1	00–FF	Standard error codes
AckResp(1)	1	00–FF	Byte 1: ISO15693 response flags
— OR —			
ErrorResp(2)	2	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Error code
— OR —			
NoData(3)	0	—	Byte 1: ISO15693 response flags

(1) Contents with tag reply *Success*
(2) Contents with tag reply *Error*
(3) No data returned due to condition described in <*Status*> field

Lock DSFID



ISO/IEC 15693 Protocol
The *Lock DSFID* command write-protects the DSFID register of the addressed tag(s). A corrupted response or lack of response does not necessarily indicate a failure to perform the lock operation.

Additionally, multiple transponders may process a nonaddressed request.
To a lock tag's DSFID, the user should:
1. Click the button for *Lock DSFID* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flags* window.
3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. *Execute* the command.

Request Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	Entity ID
Cmd2	1	6F	Lock DSFID
IsSelectMsg	1	00–FF	0: Do not set Select flag (Other): Set Select flag in response
RespType(1)	1	00–FF	0: Asynchronous reply (Other): Polled reply (prog. burst)
UID(2)	0–8	00–FF	Unique ID of tag

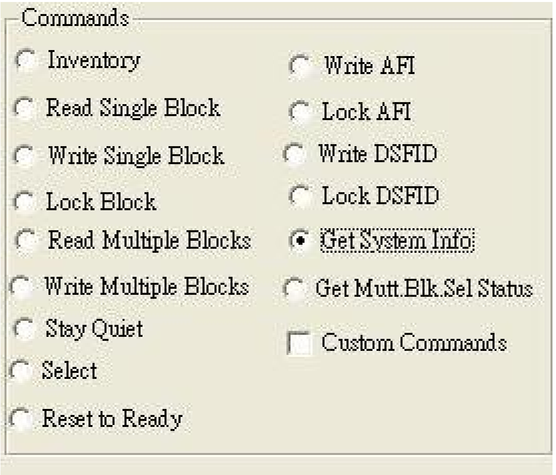
(1) A nonzero value is required for Tag-it tags.
(2) If UID field is not present, all tags in the read zone are addressed.

Response Packet:

Field	Length	Value Range	Description
Cmd1	1	04	ISO15693 entity ID
Cmd2	1	6F	Lock DSFID
Status	1	00–FF	Standard error codes
AckResp(1)	1	00–FF	Byte 1: ISO15693 response flags
— OR —			
ErrorResp(2)	2	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Error Code
— OR —			
NoData(3)	0	—	Byte 1: ISO15693 response flags

(1) Contents with tag reply *Success*
(2) Contents with tag reply *Error*
(3) No data returned due to condition described in <*Status*> field

Get System Info



The *Get System Info* command retrieves identification, application family, and data formatting and sizes as specified in the ISO15693 standard.

To get system information, the user should:
1. Click the button for *Get System Info* in the *Commands* window.
2.Click on any flags that must be set in the *Tag Flags* window.
3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).

4. *Execute* the command.

Request Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	Entity ID
Cmd2	1	70	Get system information
IsSelectMsg	1	00–FF	0: Do not set Select flag (Other): Set Select flag in response
UID(1)	0–8	00–FF	Unique ID of tag

(1) If UID field is not present, all tags in the read zone are addressed.

Response Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	ISO15693 entity ID
Cmd2	1	6F	Get system information
Status	1	00–FF	Standard error codes
InfoResp(1)	10–15	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Info flags(2)
		00–FF	Bytes 3-10: Unique ID of tag
		00–FF	Remaining bytes: Information fields(2)
— OR —			
ErrorResp(3)	2	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Error code
— OR —			
NoData(4)	0	—	Byte 1: ISO15693 response flags

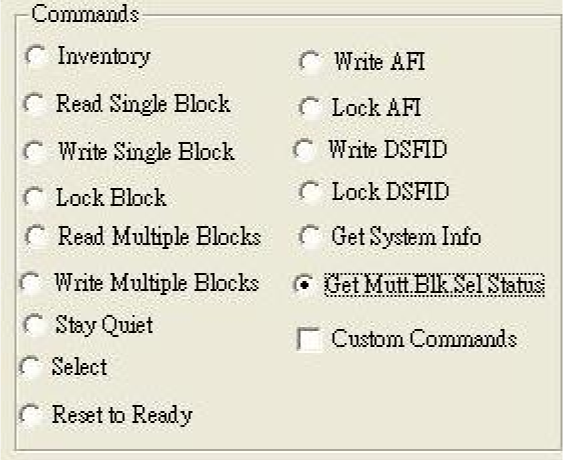
(1) Contents with tag reply *Success*
(2) See ISO15693 Standard for details on this field.
(3) Contents with tag reply *Error*
(4) No data returned due to condition described in <*Status*> field

Response Packet: (01 18 00 03 04 70 00 00 0F E5 B0 81 06 00 00 07 E0 AC AF 3F 03 88 E3 1C)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	18 00	Packet length 24 bytes

DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	70	Get system information
Status	00	ERROR _NONE
RespFlags	00	No errors
InfoFlags	0F	DSFID, AFI, memory size and IC reg supported
UID	E5 B0 81 06 00 00 07 E0	Unique ID of tag
DSFID	AC	Data storage format ID
AFI	AF	Application family ID
VICCMemorySize	3F 03	64 blocks at 4 bytes each
IcReference	88	Manufacturer information
BCC	E3 1C	LRC and ~LRC

Get Multiple-Block Security Status (Get Mult_Blk Sel Status)



ISO/IEC 15693 Protocol

The *Get Multiple-Block Security Status (Get Mutt. Blk. Sel Status)* command gets a block security status byte for each block requested. This byte encodes the write protection of the block specified (e.g., unlocked, (user/factory) locked, etc.).

To get multiple block security status, the user should:

1. Click the button for *Get Mutt.Blk.Sel Status* in the *Commands* window.
2. Click on any flags that must be set in the *Tag Flags* window.
3. Select a tag from the *UID* pull down list in the *Tag Data* window (if only one tag is present, only one choice is available).
4. Enter two digits corresponding to the starting block number in the *(First) Block Number* field in the *Tag Data* window. The blocks are numbered from *00* to *FF* (0 to 255).
5. Enter two digits corresponding to the number of blocks to be written in the *Number of Blocks* field in the *Tag Data* window. The number of blocks in the request is one less than the number of blocks that the tag returns in its response. E.g., a value of *06* in the *Number of Blocks* field requests to read 7 blocks. A value of *00* requests to read a single block.

6. *Execute* the command.

Request Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	Entity ID
Cmd2	1	71	Get Multiple-block security status
IsSelectMsg	1	00–FF	0: Do not set Select flag (Other): Set Select flag in response
StartBlk	1	00–FF	Specifies the first block that is to be read
Numblks	1	00–FF	Number of blocks to read after 1st block
UID(1)	0–8	00–FF	Unique ID of tag

(1) If UID field is not present, all tags in the read zone are addressed.

Response Packet:

Field	**Length**	**Value Range**	**Description**
Cmd1	1	04	ISO15693 entity ID
Cmd2	1	6F	Get system information
Status	1	00–FF	Standard error codes
InfoResp(1)	n(2)	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2 – n(2): Block security status
		— OR —	
ErrorResp(3)	2	00–FF	Byte 1: ISO15693 response flags
		00–FF	Byte 2: Error code
		— OR —	
NoData(4)	0	—	Byte 1: ISO15693 response flags

(1) Contents with tag reply *Success*
(2) n = <NumBlks> + 1
(3) Contents with tag reply *Error*
(4) No data returned due to condition described in <*Status*> field

Get Multiple Block Security Status Examples:
Request Packet: (01 0B 00 03 04 71 00 00 3F 43 BC)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0B 00	Packet length 11 bytes

DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	71	Get security status
IsSelectMsg	00	Not a select message
StartBlk	00	Start reading at block 0
NumBlks	3F	Read next 63 blocks also
BCC	43 BC	LRC and ~LRC

Response Packet: (01 4A 00 03 04 71 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3C C3)

Field	**Contents**	**Sumary**
SOF	01	Start of frame
PacketLen	4A 00	Packet length 74 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	71	Get security status
Status	00	ERROR _NONE
RespFlags	00	No errors
InfoFlags	0F	DSFID, AFI, memory size and IC reg supported
BlockSecurityStat	00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Concatenated block security status bytes for 64 blocks (block 2 is locked, all others are unlocked)
BCC	3C C3	LRC and ~LRC

Command Sample: (all use ASCII)

First command (make sure the com port is open)

Input -------------------------→"0108000304FF0000"
Respond ←---------------------"0108000304FF0000"

< The same respond is meaning that the port is opening successful.>

ISO 15693
<Using in multi- Reader>

	Action Step	Command
Read UID	input→	01090003040CFF0000
	input→	010C00030410002101020000
	Respond	com (port number)
	Respond	010C00030410002101020000
	Respond	Register write request
	input →	010B000304140601000000
	Respond	010B000304140601000000
	Respond (No get Tags)	ISO 15693 Inventory request. [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40]
	Respond (get Tags) UID is **E00401000752EEC5**	ISO 15693 Inventory request. [,40] [,40] [,40] [,40] [,40] [C5EE5207000104E0,63] [,40]

		[,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40]

Read UID
<Using in single- Reader>

	Action Step	Command
Initialize (Power Model)	input→	**010C00030410002101000000**
	Respond	com (port number)
	Respond	**010C00030410002101000000**
	Respond	Register write request
	input →	**0109000304F0000000**
	Respond	**0109000304F0000000**
	input →	**0109000304F1FF0000**
	Respond	**0109000304F1FF0000**
Read UID	input →	**010B000304140401000000**
	Respond	**010B000304140401000000**
	Respond (No get Tags)	ISO 15693 Inventory request. [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40]
	Respond (get Tags)	ISO 15693 Inventory request. [,40] [,40] [,40] [,40]

	UID is **E00401000752EEC5**	[,40] [C5EE5207000104E0,63] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40] [,40]

ISO 14443A
Read UID
<Using in multi- Reader>

	Action Step	Command
Read UID	input→	01090003040AFF0000
	input→	010C00030410002101090000
	Respond	com(port number)
	Respond	010C00030410002101090000
	Respond	Register write request.
	input→	0109000304A0010000
	Respond	0109000304A0010000
	Respond	14443A REQA.
	(No get Tags)	()
	(Get Tags)	(0400)(14443A UID)[14443A UID]

Read UID
<Using in single- Reader>

	Action Step	Command
Initialize (Power Model)	input→	010C00030410002101090000
	Respond	com(port number)
	Respond	010C00030410002101090000
	Respond	Register write request.
	input→	0109000304F0000000
	Respond	0109000304F0000000
Read UID	input →	0109000304A0010000
	Respond	0109000304A0010000
	Respond (No get Tags)	()
	(Get Tags)	(0400)(14443A UID)[14443A UID]

Power Model:

Full Power:

	Action Step	Command
Set Full Power	input→	010C00030410002101090000
	Respond	com(port number)
	Respond	010C00030410002101090000
	Respond	Register write request.
	input→	0109000304F0000000
	Respond	0109000304F0000000

Half Power

	Action Step	Command
Set Half Power	input→	010C00030410003101090000
	Respond	com (port number)
	Respond	010C00030410003101090000
	Respond	Register write request.
	input→	0109000304F0000000
	Respond	0109000304F0000000
	input→	0109000304F1FF0000
	Respond	0109000304F1FF0000

ISO 14443B
Read UID
<Using in multi- Reader>

	Action Step	Command
Read UID	input→	01090003040BFF0000
	input→	010C000304100021010C0000
	Respond	com(port number)
	Respond	010C000304100021010C0000
	Respond	Register write request.
	input→	0109000304B0040000
	Respond	0109000304B0040000
	Respond	14443B REQB.
	(No get Tags)	[]
	(Get Tags)	(0400)(14443B UID)[14443B UID]

Read UID

<Using in single- Reader>

	Action Step	Command
Initialize (Power Model)	input→	010C000304100021010C0000
	Respond	com(port number)
	Respond	10C000304100021010C0000
	Respond	Register write request.
	input →	0109000304F0000000
	Respond	0109000304F0000000
	input →	0109000304F1FF0000
	Respond	0109000304F1FF0000
Read UID	input →	0109000304F1FF0000
	Respond	0109000304F1FF0000
	Respond (No get Tags)	14443B REQB. [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] []