Software Communications

Protocol

The TM-001 support multi-protocol please see Table.1

ISO 14443A/B				ISO 15693	Tag-it
106kpbs	212kpbs	424kbps	848kbps		

Table: Protocol support

The **Low level 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.

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 read, start address	01 0A 00 03 04 **14 06 01 00** 0000
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-00 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

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:

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	14	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	14	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

Inventory Example:

Request Packet (01 0B 00 03 04 14 01 00 00 00 00)

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

Device ID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	14	Inventory request
IsOneSlot	01	1slot
AfiFlg	00	No AFI byte included
MskLen	00	Mask length
BCC	00 00	LRC and ~LRC

Response Packet :(01 0B 00 03 04 14 01 00 00 00 00)

Field	Contents	Summary
SOF	01	Start of frame
PacketLen	0B 00	Packet length 19 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	14	Inventory request
Status	00	ERROR_NONE
Inv. Resp. Flags	00	No error
DSFID	00	Data storage format ID
UID	E5 B0 81 06 00 00 07 E0	ID of the tag(LSB first)
BCC	00 00	LRC and ~LRC

Read Single Block:

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	01	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

Read Single Block Example:
Request Packet: (010B000304180020050000)

Field	Contents	Summary
SOF	01	Start of frame
PacketLen	0B 00	Packet length 19 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Read single block
IsSelectMsg	00	Not a select message
RequSequrity	20	Return security status
BlkNum	05	Block number 5
UID	B8 9A 92 06 00 00 07 E0	Unique ID of tag
BCC	00 00	LRC and ~LRC

Response Packet: (010B000304180020050000)

Field	Contents	Summary
SOF	01	Start of frame
PacketLen	0F 00	Packet length 15 Bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Read single block
Blk. Security Stat.	00	Block unlocked
Read Data	02 00 05	Data read from block 5
SCC	00 00	LRC and ~LRC

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

Write Single Block Example:

Request Packet: (010F00030418002100000000000000)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	10 F0	Packet length 16 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Write single block
IsSelectMsg	00	Not a select message
RespType	21	Poll for a reply (prog. burst)
BlkNum	00	Block number 5
BlkBytes	00	4 bytes per block
BlkData	00 00 00 00	Data to write to block
BCC	00	A6 LRC and ~LRC

Response Packet: (010F00030418002100000000000000)

Field	**Contents**	**Summary**

SOF	01	Start of frame
PacketLen	0F 00	Packet length 10 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Write single block
Status	00 21	ERROR _NONE
AckResp	00 00 00 00 00	Successful write
BCC	00 00	LRC and ~LRC

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

Lock Block Example:

Request Packet: (010B000304180022220000)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0B 00	Packet length 19 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Lock block
IsSelectMsg	00	Not a select message
RespType	22	Poll for a reply (prog. burst)

BlkNum	11	Lock block number 11
BCC	00 00	LRC and ~LRC

Response Packet: (010B000304180022110000)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0B 00	Packet length 10 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Lock block
Status	00	ERROR _NONE
AckResp	22	Successful lock
BlkNum	11	Lock block number 11
BCC	00 00	LRC and ~LRC

Read Multiple Blocks

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

Read Multiple Blocks Example:

Request Packet: (010C00030418002311220000)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0C 00	Packet length 12 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Read multiple blocks
IsSelectMsg	00	Not a select message
ReqSecurity	23	Return security status
StartBlk	11	Start reading at block 11
NumBlks	22	Read next 22 blocks also
BCC	00 00	LRC and ~LRC

Response Packet: (010C00030418002311220000)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0C 00	Packet length 10 bytes
DeviceID	03	Device

Cmd1	04	ISO15693 entity ID
Cmd2	18	Lock Block
Status	00	ERROR _NONE
RespFlags	23	Successful lock
Block Security Status + Block Read Data	00 44 33 22 11 – Block 0 00 00 00 00 00 – Block 1	*Block Security Status bytes followed by Block Read Data, repeated for each block requested.*
	00 04 03 02 01 – Block 2	
	00 00 00 00 00 – Block 3	
	00 00 00 00 00 – Block 4	
	01 00 00 30 86 – Block 5	*Note: Block 5 is locked in this example.*
	00 00 00 00 00 – Block 6	
	00 00 00 00 00 – Block 7	
BCC	AB 54	LRC and ~LRC

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

Write Multiple Blocks Example:

Request Packet: (01 15 00 03 04 69 00 01 02 01 04 12 35 36 38 21 53 63 83 C7 38)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	15 00	Packet length 21 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	69	Lock block
IsSelectMsg	00	Not a select message
RespType	01	Poll for a reply (prog. burst)
StartBlk	02	Start writing at block 2
NumBlks	01	Write next block also
BlkBytes	04	4 bytes per block

BlkData	12 35 36 38 21 53 63 83	Data to write to block 2 Data to write to block 3
BCC	C7 38	LRC and ~LRC

Response Packet: (01 0A 00 03 04 66 00 00 65 9A)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0A 00	Packet length 10 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	69	Write multiple blocks
Status	00	ERROR _NONE
AckResp	00	Successful write
BCC	65 9A	LRC and ~LRC

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

Stay Quiet Example:

Request Packet: (010A0003041800020000)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0A 00	Packet length 16 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Stay quiet
Status	00 02	Unique ID of tag
BCC	00 00	LRC and ~LRC

Response Packet: (**010A0003041800020000**)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0A 00	Packet length 9 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Stay quiet
Status	00	ERROR _NONE
BCC	02 00 00	LRC and ~LRC

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

Select Example:

Request Packet: (010A0003041800250000)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0A 00	Packet length 16 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	ISO15693 entity ID
Select	00 25	Select commands
BCC	00 00	LRC and ~LRC

Response Packet: (010A0003041800250000)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0A 00	Packet length 10 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Select
Status	00	ERROR _NONE
AckResp	25	Successful select
BCC	00 00	LRC and ~LRC

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:

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

Reset to Ready Example:

Request Packet: (010A0003041800260000)

Field	Contents	Summary
SOF	01	Start of frame
PacketLen	09 00	Packet length 16 bytes

DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18 00	Reset to ready
IsSelectMsg	26	Select message
BCC	00 00	LRC and ~LRC

Write AFI

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:

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

Write AFI Example:

Request Packet: (010B000304180027990000)

Field	Contents	Summary
SOF	01	Start of frame
PacketLen	0B 00	Packet length 11 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Write AFI
IsSelectMsg	00	Select message
RespType	27	Polled reply (prog. burst)
AfiVal	99	Application family ID
BCC	00 00	LRC and ~LRC

Response Packet: (010B000304180027990000)

Field	Contents	Summary
SOF	01	Start of frame
PacketLen	0B 00	Packet length 10 bytes

DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Write AFI
Status	00	ERROR _NONE
ErrorResp	27	ISO15693 error flag
	99	Error: Locked address
BCC	00 00	LRC and ~LRC

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

Lock AFI Example:

Request Packet: (010A0003041800280000)

Field	Contents	Summary
SOF	01	Start of frame
PacketLen	0A 00	Packet length 18 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Lock AFI
IsSelectMsg	00	Select message
RespType	28	Polled reply (prog. burst)
BCC	00 00	LRC and ~LRC

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

Write DSFID Example:
Request Packet: (010B000304180029770000)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0B 00	Packet length 11 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Write DSFID
IsSelectMsg	00	Not a select message
RespType	29	Polled reply (prog. burst)
AfiVal	77	Application family ID
BCC	00 00	LRC and ~LRC

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

Lock DSFID Example:

Request Packet: (010A00030418002A0000)

Field	Contents	Summary
SOF	01	Start of frame
PacketLen	0A 00	Packet length 11 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Lock DSFID
IsSelectMsg	00	Select message
RespType	2A	Polled reply (prog. burst)
BCC	00 00	LRC and ~LRC

Response Packet: (010A00030418002A0000)

Field	Contents	Summary
SOF	01	Start of frame
PacketLen	0A 00	Packet length 11 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Lock DSFID
Status	00	ERROR _NONE
ErrorResp	2A	ISO15693 entity flag
BCC	00 00	LRC and ~LRC

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

Get System Info **Example:** (010A00030418002B0000)

Field	**Contents**	**Summary**
SOF	01	Start of frame
PacketLen	0A 00	Packet length 24 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Get system information
Status	00	ERROR _NONE
RespFlags	2B	No errors
InfoFlags	00 00	DSFID, AFI, memory size and IC reg supported

Response Packet Example: (010A00030418002B0000)
Request mode.
[000F74176507000104E077991B0301]

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: (010C00030418002C11220000)

Field	Contents	Summary
SOF	01	Start of frame
PacketLen	0C 00	Packet length 11 bytes
DeviceID	03	Device
Cmd1	04	ISO15693 entity ID
Cmd2	18	Get security status
IsSelectMsg	00 2C	Not a select message
StartBlk	11	Start reading at block 11
NumBlks	22	Read next 22 blocks also
BCC	00 00	LRC and ~LRC

Response Packet: (010C00030418002C11220000)

Request mode.

[0001000000000000000000000]

Command Sample for get UID: (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	ISO 15693 Inventory request.

	(No get Tags)	[,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

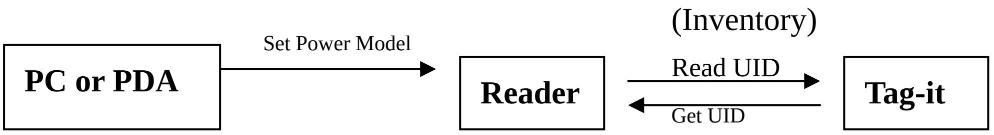
Initialize (Power Model)	input→	**010C00030410002101000000**
	Respond	com (port number)
	Respond	**010C00030410002101000000**
	Respond	Register write request
	input →	**0109000304F0000000**
	Respond	**0109000304F0000000**
	input →	**0109000304F1FF0000**
	Respond	**0109000304F1FF0000**
Read UID	input →	**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) UID is **E00401000752EEC5**	ISO 15693 Inventory request. [,40] [,40] [,40] [,40] [,40] [C5EE5207000104E0,63] [,40] [,40] [,40] [,40]

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

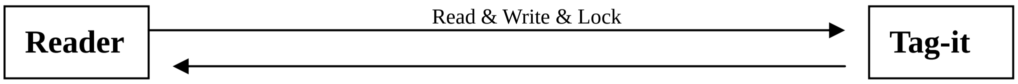
For Tag-it Commands flow

Fellow those two steps, the Reader could to control and manage the Tag-it. If you use the command to lock the block, the block will change to be only read.

Step1:



Step2: After get UID, the reader can communicate to Tag



First command (make sure the com port is open)

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

Before communicate the Tag-it, first step is to read the Tag UID.

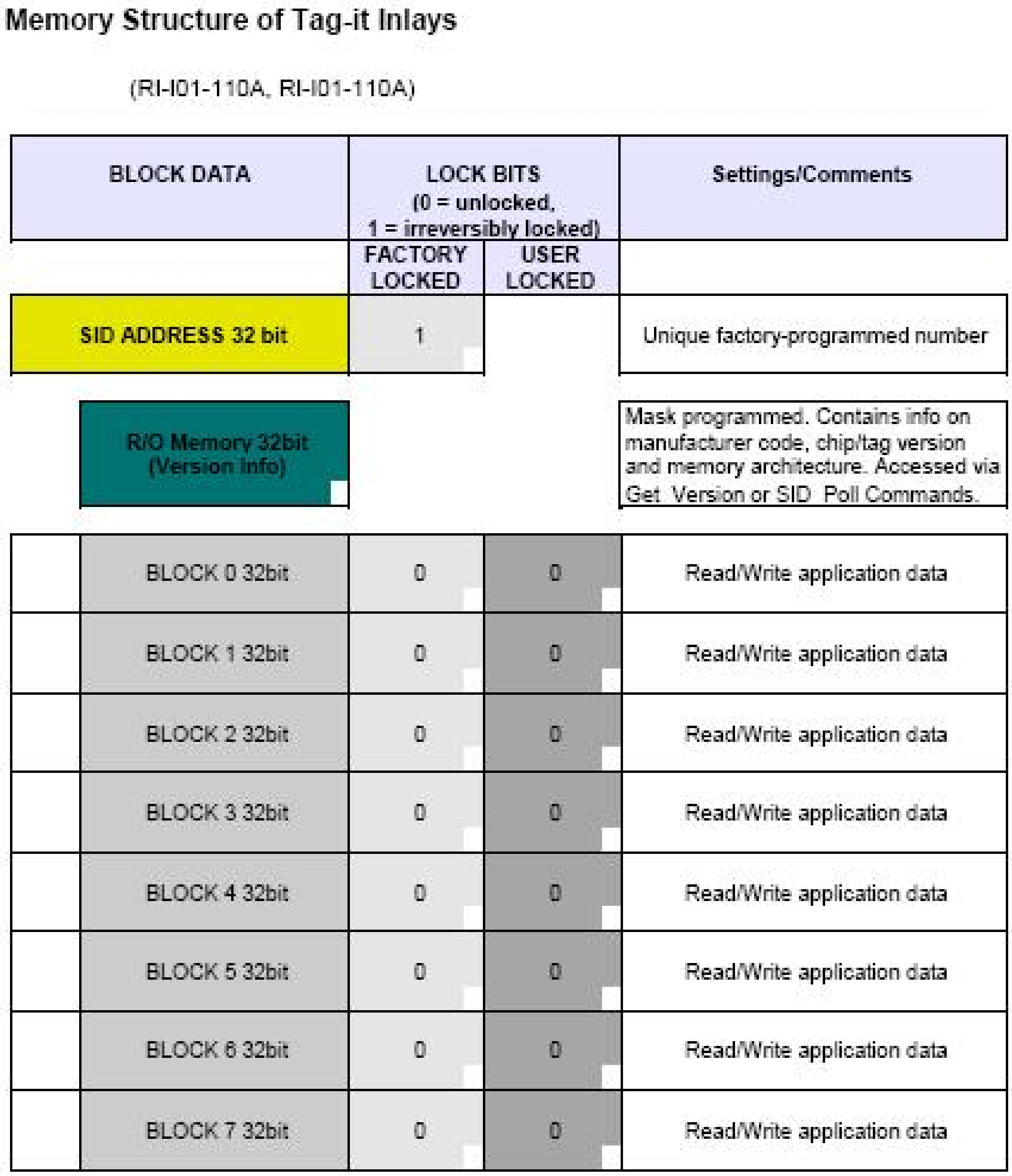
Read UID

<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) UID is **E00401000752EEC5**	ISO 15693 Inventory request. [,40] [,40] [,40] [,40] [,40] [C5EE5207000104E0,63] [,40] [,40] [,40] [,40]

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

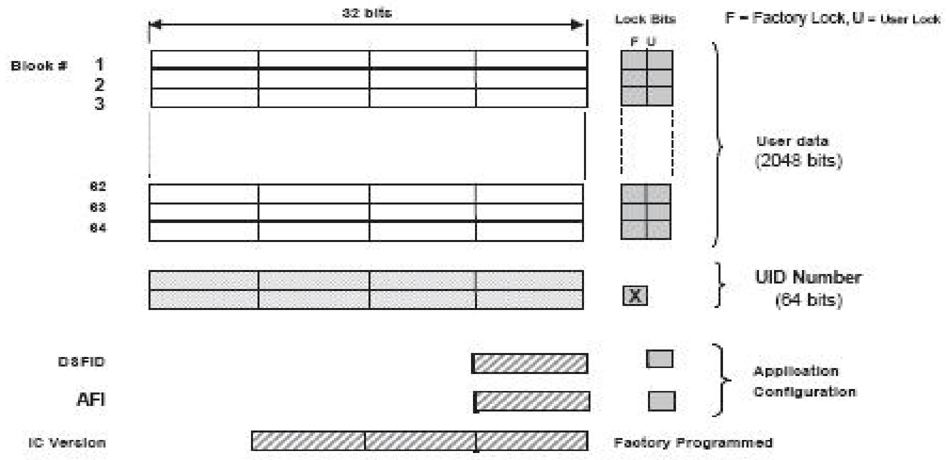


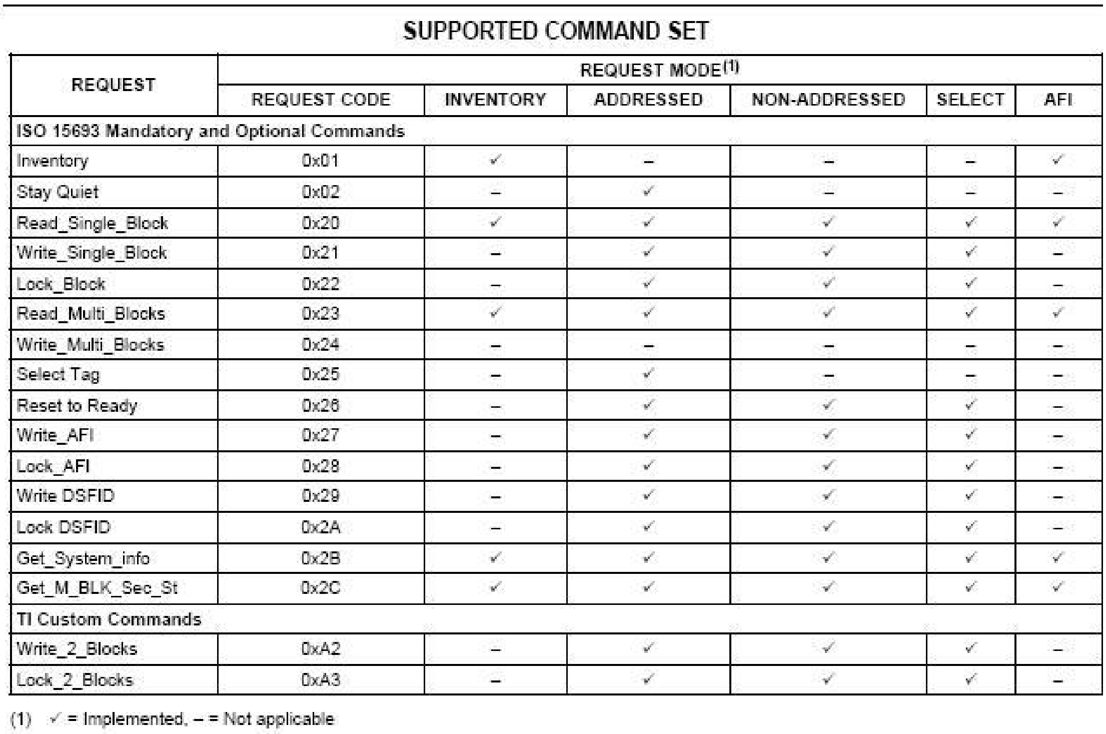
Memory Structure of Tag-it Inlays

(RI-I01-110A, RI-I01-110A)

BLOCK DATA	LOCK BITS (0 = unlocked, 1 = irreversibly locked)		Settings/Comments
	FACTORY LOCKED	USER LOCKED	
SID ADDRESS 32 bit	1		Unique factory-programmed number
R/O Memory 32bit (Version Info)			Mask programmed. Contains info on manufacturer code, chip/tag version and memory architecture. Accessed via Get_Version or SID_Poll Commands.
BLOCK 0 32bit	0	0	Read/Write application data
BLOCK 1 32bit	0	0	Read/Write application data
BLOCK 2 32bit	0	0	Read/Write application data
BLOCK 3 32bit	0	0	Read/Write application data
BLOCK 4 32bit	0	0	Read/Write application data
BLOCK 5 32bit	0	0	Read/Write application data
BLOCK 6 32bit	0	0	Read/Write application data
BLOCK 7 32bit	0	0	Read/Write application data

MEMORY ORGANIZATION





SUPPORTED COMMAND SET

REQUEST	REQUEST MODE(1)					
	REQUEST CODE	INVENTORY	ADDRESSED	NON-ADDRESSED	SELECT	AFI
ISO 15693 Mandatory and Optional Commands						
Inventory	0x01	✓	–	–	–	✓
Stay Quiet	0x02	–	✓	–	–	–
Read_Single_Block	0x20	✓	✓	✓	✓	✓
Write_Single_Block	0x21	–	✓	✓	✓	–
Lock_Block	0x22	–	✓	✓	✓	–
Read_Multi_Blocks	0x23	✓	✓	✓	✓	✓
Write_Multi_Blocks	0x24	–	–	–	–	–
Select Tag	0x25	–	✓	–	–	–
Reset to Ready	0x26	–	✓	✓	✓	–
Write_AFI	0x27	–	✓	✓	✓	–
Lock_AFI	0x28	–	✓	✓	✓	–
Write DSFID	0x29	–	✓	✓	✓	–
Lock DSFID	0x2A	–	✓	✓	✓	–
Get_System_info	0x2B	✓	✓	✓	✓	✓
Get_M_BLK_Sec_St	0x2C	✓	✓	✓	✓	✓
TI Custom Commands						
Write_2_Blocks	0xA2	–	✓	✓	✓	–
Lock_2_Blocks	0xA3	–	✓	✓	✓	–

(1) ✓ = Implemented, – = Not applicable

Each Command for Tag-it

Commands	**Commands Code(ASCII)**
Inventory	010B000304140401000000
Read Single Block	010B000304180020[][]0000
Write Single Block	010F0003041840212[][]{_ _ _ _ _ _ _ _ }0000
Lock Block	010B000304184022[][]0000
Read Multi-Blocks	010C00030418402300010000
Write Multi-Blocks	Non-Support
Select Tag	0112000304182025<Tag UID >
Reset to Ready	010A0003041800260000

Write AFI	010B000304184027<AFI>0000
Lock AFI	010A0003041840280000
Write DSFID	010B000304184029< DSFID >0000
Lock DSFID	010A00030418402A0000
Get System info	010A00030418402B0000
Get M-BLK-Sec-St	010C00030418402C[][][][]0000
TI Custom commands	
Write-2-Blocks	01140003041810A20707<Data>0000
Lock-2-Block	010C0003041810A307070000

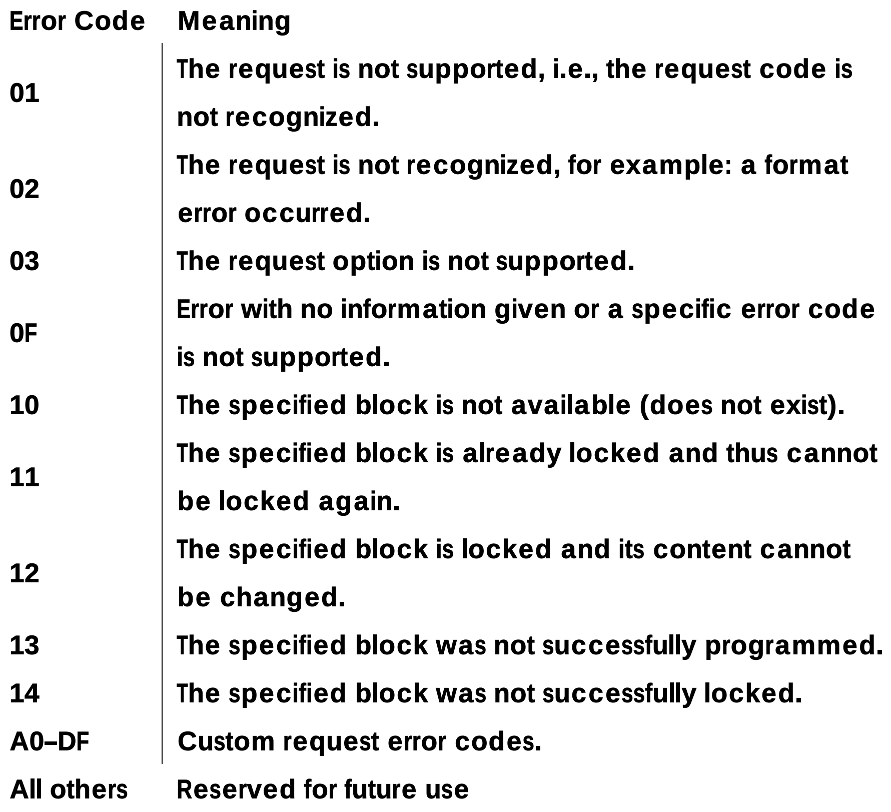
*[][]-Block number(00~3F)   ex. 01

*{_ _ _ _ _ _ _ _ }-write data (0~F)   ex. 06321F8F

*<Tag UID > - ex. B0E82124000007E00000 -> UID: E00700002421E8B0

Error codes -----

When you get the response that is an error codes, you can search the table to know the meaning.



Error Code	Meaning
01	The request is not supported, i.e., the request code is not recognized.
02	The request is not recognized, for example: a format error occurred.
03	The request option is not supported.
0F	Error with no information given or a specific error code is not supported.
10	The specified block is not available (does not exist).
11	The specified block is already locked and thus cannot be locked again.
12	The specified block is locked and its content cannot be changed.
13	The specified block was not successfully programmed.
14	The specified block was not successfully locked.
A0–DF	Custom request error codes.
All others	Reserved for future use