TEXAS INSTRUMENTS

Reader Series 4000

S4100 Multi-Function Reader Module RF-MGR-MNMN

ISO 14443 Library Reference Guide

11-01-21-702 October 2003

First Edition - October 2003

This is the first edition of this manual. It describes the **TI Series 4000 Reader.**

It contains a description of the following reader module:

S4100 Multi-Function Reader Module P/N: **RF-MGR-MNMN-N0**

Texas Instruments (TI) reserves the right to make changes to its products or services or to discontinue any product or service at any time without notice. TI provides customer assistance in various technical areas, but does not have full access to data concerning the use and applications of customer's products

Therefore, TI assumes no liability and is not responsible for customer applications or product or software design or performance relating to systems or applications incorporating TI products. In addition, TI assumes no liability and is not responsible for infringement of patents and/or any other intellectual or industrial property rights of third parties, which may result from assistance provided by TI.

TI products are not designed, intended, authorized or warranted to be suitable for life support applications or any other life critical applications which could involve potential risk of death, personal injury or severe property or environmental damage.

Read This First

About This Manual

This reference guide for the Series 4000 Multi-Function (13.56 MHz & 134.2 KHz) Reader is designed for use by TI customers who are engineers experienced with RFID Systems and Radio Frequency Identification Devices (RFID).

Device Name	Boot Loader Firmware Version
RF-MGR-MNMN-N0	1.02

The Regulatory, safety and warranty notices that must be followed are provided in Chapter 2.

Conventions

The following pictograms and designations are used in the operating instructions:

WARNING:

A WARNING IS USED WHERE CARE MUST BE TAKEN, OR A CERTAIN PROCEDURE MUST BE FOLLOWED, IN ORDER TO PREVENT INJURY OR HARM TO YOUR HEALTH.

CAUTION:

This indicates information on conditions, which must be met, or a procedure, which must be followed, which if not needed could cause permanent damage to the system.

Note:

Indicates conditions, which must be met, or procedures which must be followed, to ensure proper functioning.

Information:

Indicates conditions or procedures that should be followed to ensure proper functioning of the system.

If You Need Assistance

Application Centers are located in Europe, North and South America, the Far East and Australia to provide direct engineering support.

For more information, please contact your nearest TIRIS Sales and Application Center. The contact addresses can be found on our home page: http://www.tirfid.com.

Numerical Representations

Unless otherwise noted, numbers are represented as decimal.

Hexadecimal numbers are represented with the suffix $_{16}$, e.g. $A5F1_{16}$

Binary numbers are represented with the suffix $_2$, e.g. 1011_2

Byte representations: the least significant bit (lsb) is bit 0 and the most significant bit (msb) is bit 7.

Document Overview

Chapter 1

ISO 14443-3 Type-A

Topic **Page**

1.1 ISO 14443-3 Type A Overview

The following sections define and detail the Protocol functionality in the 14443 A Module of the MFR Base Application. This information includes 14443 Type-A Protocol Commands and the data/parameters associated with them.

Find Token Request (41_{16})

The host application can send the MFR Base Application a Request Packet to check if a token is present. This packet contains a loop count that sets the maximum number of times the MFR Module Base Application searches for the token. This function allows a great deal of flexibility. It is possible to search for a variety of transponders or a single type of token. If multiple transponders are selected, the function returns the first token that is read. The Application continues to loop as specified unless it detects a token. The Application doesn't return every type of token - just the first one it reads.

After a successful read, the MFR Module Base Application responds with **ERROR_NONE** in the <*Status*> followed by token's RF Technology Type and token data. The RF poll stops once a valid token is found. If a valid token is not found within the number of loops selected, the MFR Module Base Application responds with an **ERROR_TOKEN_NOT_PRESENT** <*Status*> field. The RF Poll stops once the loops are complete.

The ISO14443A token search is handled by the function Find_Token_14443A(). Note that this function is not called when ISO14443A is not in the Priority table unless the request is directed specifically to the ISO14443A library. It is not called if a different token is discovered prior to reaching the ISO14443A format in the priority table.

Information:

- The Find Token for ISO14443A finds one token only. Any collision is reported as an error.
- PICC Activation: The Find Token request is implemented internally as multiple block calls of REQA/WUPA followed by Anti-collision/Select requests for the token found unambiguously (no collisions).
-

Information:

- Protocol Activation: RATS and PPS requests are sent to the token resolved through anti-collision/Select, if the transponders are ISO14443-4 compliant.
- ATQA, RATS and Answer to PPS responses set specific parameters of internal token structures.
- ISO14443 -4 packets (I, R and S blocks) also use the ATS for the particular token at hand to determine whether to use CID and/or NAD.
-

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	01/02	Entity ID
Cmd2	1	41	Find Token
LoopCount	1	00 – FF	Number of attempts to find token (0 ⇒ Loop until next request)

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	ISO14443B Entity ID
Cmd2	1	41	Find Token
Status	1	00 – FF	Standard Error Codes
EntityId	1	03	ISO14443A Library
UID_CL0[1]	6	00 – 0E	Byte 1: Card Identifier
		00	2: Cascade Level 0
		00 – FF	3-6: Unique ID of the token
		— OR —	
UID_CL1[2]	9	00 – 0E	Byte 1: Card Identifier
		01	2: Cascade Level 1
		00 – FF	3-9: Unique ID of the token
		— OR —	
UID_CL2[3]	12	00 – 0E	Byte 1: Card Identifier
		02	2: Cascade Level 2
		00 – FF	3-12: Unique ID of the token
		— OR —	
NoData[4]	0	—	(See <Status> field)

ReplyDat

(1) Contents with valid ATQA response from the token supporting CL0 only.
(2) Contents with valid ATQA response from the token supporting CL1 only.
(3) Contents with valid ATQA response from the token supporting CL2 only.
(4) No Data returned due to condition described in <*Status*> field.

Find Token Example: The request packet specifies 10 loops.
Request Packet: (**01 09 00 03 03 41 0A 43 BC**)

Field	Contents	Summary
SOF	01	Start of Frame
PacketLen	09 00	Packet Length 9 bytes
DeviceID	03	Device is MFR Module
Cmd1	03	ISO14443A Entity ID
Cmd2	41	Find Token Request
LoopCount	0A	10 loops maximum
BCC	43 BC	LRC and ~LRC

Response Packet: (**01 0F 00 03 03 41 00 03 00 34 03 04 09 76 89**)

The response returns information from up to 16 transponders and indicates collisions, if any. For each token found unambiguously (without a collision in its response slot), 5 bytes of data are reported. The first byte is the CID assigned by the PCD (reader) to the token for the session (for example, the duration between transmitter-on through transmitter-off). The next 4 bytes are the PUPI per ISO14443A. The PUPI by its definition is a pseudo random number and MS byte - LS byte distinctions are not relevant as long as the byte sequence order is maintained when communicating with the PCD.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0F 00	Packet Length 15 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443A
Command 2	41	Find Token Request
Status Byte	00	Successful
Entity ID	03	Library Layer – ISO14443A
CID	00	CID per ISO14443A
PUPI	34 03 04 09	PUPI per ISO14443A
BCC	76 89	LRC and ~LRC

Pass-Through Request (45_{16})

The Pass-Though Request Packet provides a way for a Host to have direct communications with an Entity Module library. The Pass-Through Request Packet uses the *<Cmd 1>* field to specify which Entity to direct the data to and then uses the *<Data Layer>* to provide data that the library can parse and use. This function provides a way to isolate and test the library functions without having to make any modifications to the Application Layer.

The ISO14443A library module/entity allows direct access to the RF interface via the pass-through request. When the ISO14443A module receives a pass-through request, the module configures the HF ASIC to the ISO14443A-2 specific RF scheme.

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	Entity ID
Cmd2	1	45	Pass Through
FrameType	1	00-02	Frame types per ISO14443A-3
BitsOrBytes	1(1)	00 - FF	#bits/#bytes to send(1)
Data	0 – n(2)	00 – FF	Data to send via RF channel

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	Entity ID
Cmd2	1	45	Pass Through
Status	1	00 – FF	Standard Error Codes
#bytes	0 - n(2)	00 - FF	Number of bytes received
#Xtrabits	0 – 7	00 - 7F	Number of extra bits received
Data	n / n+1(2)	00 – FF	ISO14443A reply data (if any received)

(1) #bits for FrameType values 0 and 1. #bytes for FrameType = 2.
(2) Length 'n' depends on RF message.

FrameType: Three types of packets are described in ISO14443A specification.

0 for ISO14443A-3 short frame,
1 for ISO14443A-3 bit oriented anti-collision frame and
2 for ISO14443A-3 standard frame.

BitsOrBytes: The interpretation of the contents of this field depends on the packet type. For packet types 1 and 2 it specifies the number of bits to be sent. For packet type 3 it specifies the number of bytes to be sent.

The fields FrameType and BitsOrBytes are mandatory. Payload data bytes: the number of bytes in this field should be sufficient to hold the #bits/#bytes specified in the Payload Length field.

The payload data in the pass-through request is not interpreted and it has meaning only to the token that may receive it. Therefore the process that initiated the request must know the structure and data content of the transponder protocol. Some of these transponder protocol details are specified in ISO14443A layer 3 and layer 4 specification documents. Application and security layer commands may also be sent to the token via the pass-through request as long as the calling process understands and follows the states of the token at hand.

The response data field contains (not including the Response Status Byte):

#bytes field: Indicates how many complete bytes of data were received.
#Xtrabits field: Indicates the number of extra bits received (see example 1 below).

Raw data as received from the token, except for the parity bits which are first verified and then removed from the received data before being stored into the receive buffer.

The Command 1 field is set to **02** for the ISO14443A library. The payload data portion of the request packet is passed to the function Pass_Through_14443A(). This allows the host full control over the HF ASIC and what it sends. The data must strictly follow the ISO14443 Type-A layer 3 and 4 protocol.

Information:

- The library only supports a PCD to PICC and PICC to PCD data rate of 106 kbps.
- Interleaving Pass-through requests and non-pass-through requests are not generally recommended. Sending a non-pass-through request may set internal data structures that influence how subsequent non-pass-through requests' parameters are handled.

14443-A Pass-Through REQA Request

Send a valid ISO14443A type Token Pass-through REQA Request Packet to the MFR Module device with the COM port set to 9600 8N1 and validate the data received. Ensure a valid ISO14443A type token is in the field when the request is issued. Any transponders that are in the field should respond with an ATQA after the REQA downlink per ISO14443A spec.

Request Packet: (**01 0B 00 03 02 45 00 07 26 6F 90**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0B 00	Packet Length 11 bytes
Device ID	03	Terminal is MFR Module
Command 1	02	Library Layer – ISO14443A
Command 2	45	Pass-Through Request
Data	00 07 26	REQA request – 1 slot
BCC	6F 90	LRC and ~LRC

Response Packet: (**01 0E 00 03 02 45 00 02 04 04 00 0A 43 BC**)
This response packet indicates that a total of 20 bits were received from the token as indicated by #bytes (2) and #bits (4). Total bits received = #bytes * 8 + #bits.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0E 00	Packet Length 14 bytes
Device ID	03	Terminal is MFR Module
Command 1	02	Library Layer – ISO14443A
Command 2	45	Pass-Through
Status Byte	00	ERROR_NONE
Data	02	# bytes in response
Data	04	# extra bits in response
Data	04 00 0A	Application data
BCC	43 BC	LRC and ~LRC

14443-A Pass-Through WUPA Request

Send a valid 14443A type Token Pass-through WUPA Request Packet to the MFR Module device with the COM port set to 9600 8N1 and validate the data received. Ensure a valid ISO14443A type token is in the field when the request is issued. Any transponders that are in the field should respond with an ATQA after the REQA downlink per ISO 14443A spec.

Request Packet: (**01 0B 00 03 02 45 00 07 52 1B E4**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0B 00	Packet Length 11 bytes
Device ID	03	Terminal is MFR Module
Command 1	02	Library Layer – ISO14443A
Command 2	45	Pass-through request
Data	00 07 52	WUPA request – 1 slot
BCC	1B E4	LRC and ~LRC

Response Packet: (**01 0E 00 03 02 45 00 02 04 04 00 0A 43 BC**)
This response packet indicated that a total of 20 bits were received from the token as indicated by #bytes (2) and #bits (4). Total bits received = #bytes * 8 + #bits.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0E 00	Packet Length 14 bytes
Device ID	03	Terminal is MFR Module
Command 1	02	Library Layer – ISO14443A
Command 2	45	Pass-through request
Status Byte	00	ERROR_NONE
Data	02	# bytes in response
Data	04	# extra bits in response
Data	04 00 0A	Application data
BCC	43 BC	LRC and ~LRC

14443A Pass-Through ANTICOLLISION Request

Send a valid 14443A type Token Pass-through ANTICOLLISION Request Packet to the MFR Module device with the COM port set to 9600 8N1 and validate the data received. Ensure a valid ISO14443A type token is in the field when the request is issued. Send a REQA/WUPA and receive a valid ATQA prior to sending the ANTICOLLISION request.

ANTICOLLISION request/response packet is sent in a bit oriented anti-collision frame format. Once a token responds to this command, the token goes to the ACTIVE state. The Request Packet specifies that 16 bits are to be sent via a bit oriented anti-collision frame. Data contents (**93 20**) $_{16}$ indicate, according to the ISO14443A standard, cascade level 1 and the PCD will send zero bits of UID data. The Response packet shows a UID of value (**61 B0 28 65**) $_{16}$.

Request Packet: (**01 0C 00 03 02 45 01 10 93 20 EB 14**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0C 00	Packet Length 12 bytes
Device ID	03	Terminal is MFR Module
Command 1	02	Library Layer – ISO14443A
Command 2	45	Pass-through request
Data	01	Bit oriented anti-collision frame
Data	10	# bits to send
Data	93 20	Data to send
BCC	EB 14	LRC and ~LRC

Response Packet: (**01 13 00 03 02 45 00 07 05 93 20 61 B0 28 65 9C 21 C6 39**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	13 00	Packet Length 19 bytes
Device ID	03	Terminal is MFR Module
Command 1	02	Library Layer – ISO14443A
Command 2	45	Pass-through request
Status Byte	00	ERROR_NONE
Data	07	# bytes in response
Data	05	# extra bits in response
Data	93 20	SEL and NVB bytes
Data	61 B0 28 65	UID bytes
Data	9C	LRC over UID bytes
BCC	C6 39	LRC and ~LRC

14443A Pass-Through SELECT Request

Send a valid 14443A type Token Pass-through SELECT Request Packet to the MFR Module device with the COM port set to 9600 8N1 and validate the data received. Ensure a valid ISO14443A type token is in the field when the request is issued. Send a REQA/WUPA, receive a valid ATQA, send ANTICOLLISION with 0 UID bits and receive complete UID prior to sending the SELECT request.

SELECT request/response packet is sent in standard frame format. Once a token sends SAK (Select Acknowledge), the token goes to the ISO14443-4 layer. The Request Packet specifies that 9 bytes will be sent via standard frame. Data contents in the Request Packet (**93 70**)$_{16}$ indicate, according to the ISO14443A standard, cascade level 1 and the PCD will send all 40 bits of UID data. Next four bytes (**61 B0 28 65**)$_{16}$ are the UID followed by a 1-byte BCC over the UID. The last two data bytes are CRC over the data bytes per ISO14443A standard frame.

Request Packet: (**01 13 00 03 02 45 02 09 93 70 61 B0 28 65 9C 06 92 2A D5**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	13 00	Packet Length 19 bytes
Device ID	03	Terminal is MFR Module
Command 1	02	Library Layer – ISO14443A
Command 2	45	Pass-through request
Data	02	Standard frame
Data	09	# bytes to send
Data	93 70	Cascade level 1, 40 bits UID
Data	61 B0 28 65	UID of the token at the antenna
Data	9C	LRC over UID data
Data	06 92	CRC_A over token packet
BCC	2A D5	LRC and ~LRC

Response Packet: (**01 0F 00 03 02 45 00 03 05 88 BE 59 26 05 FA**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0F 00	Packet Length 15 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	45	Pass-through request
Status Byte	00	Successful
Data	03	# bytes in response
Data	05	# extra bits in response
Data	88 BE 59	SAK data and CRC_A over SAK
Data	26	Extra bits
BCC	05 FA	LRC and ~LRC

Transmitter On Request (48_{16})

The Transmitter On Request is used to turn the transmitter ON, for a specific entity. The request packet specifies in the <*Cmd1*> field the intended entity. The application layer firmware redirects the request to the intended entity. The entity then sets up appropriate hardware and software configurations needed to implement its specific details and then turns the transmitter on.

The ISO14443A Transmitter on request firstly configures the hardware ports to their proper data directions (input/output), sets up pull-up resistors for input ports and sets up the correct logic levels at the output ports. The transmitter itself is turned on and then the entity communicates with the HF ASIC to set up the ISO14443A protocol related register settings in the HF ASIC.

The request packet has no data bytes. The <*Cmd1*> field indicates the entity intended to turn on the transmitter. The response status byte indicates the success/error status.

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	Entity ID
Cmd2	1	48	Transmitter On

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	ISO14443A Entity ID
Cmd2	1	48	Transmitter On
Status	1	00 – FF	Standard Error Codes

Transmitter ON Example:

Request Packet: (**01 08 00 03 02 48 40 BF**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	48	Transmitter On Request
BCC	40 BF	LRC and ~LRC

Response Packet: (**01 09 00 03 02 48 00 41 BE**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	48	Transmitter On Request
Status Byte	00	Successful
BCC	41 BE	LRC and ~LRC

Transmitter Off Request (49_{16})

Transmitter Off Request is used to turn the transmitter OFF for a specific entity. The request packet specifies in the <*Cmd1*> field the intended entity. The application layer firmware redirects the request to the intended entity. The entity then sets up appropriate hardware and software configurations needed to implement its specific details and then turns the transmitter off.

The ISO14443A entity turns the transmitter off and returns a response. The request packet consists of no data bytes. The <*Cmd1*> field indicates the entity intended to turn off the transmitter. The response status byte indicates the success/error status.

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	Entity ID
Cmd2	1	49	Transmitter Off

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	ISO14443A Entity ID
Cmd2	1	49	Transmitter Off
Status	1	00 – FF	Standard Error Codes

Transmitter OFF Example:

Request Packet: (**01 08 00 03 02 49 41 BE**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	49	Transmitter Off Request
BCC	41 BE	LRC and ~LRC

Response Packet: (**01 09 00 03 02 49 00 40 BF**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	49	Transmitter Off Request
Status Byte	00	Successful
BCC	40 BF	LRC and ~LRC

REQA/WUPA Request ($61/62_{16}$)

The host application can request the ISO14443A Module to send a REQA/WUPA packet over the RF interface according to the ISO14443A-2 standard using this packet. REQA is sent if command 61_{16} is specified and WUPA is sent if command 62_{16} is specified. After a successful REQA/WUPA execution the MFR Module Terminal responds with **ERROR_NONE** in the Response Status byte field and a one-byte CID followed by token's ATQA. The ISO14443A REQA/WUPA request is handled by the function ISO14443A_Handle_REQA_WUPA ().

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	Entity ID
Cmd2	1	61-62	REQA/WUPA

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	ISO14443A Entity ID
Cmd2	1	61-62	REQA/WUPA
Status	1	00 – FF	Standard Error Codes
ATQA(1)	2	00 – FF	Bytes1-2: ATQA from the token

Re

— OR —

NoData[2]	0	—	(See <Status> field)

[1] Contents with valid ATQA response from the token.
[2] No Data returned due to condition described in <*Status*> field.

Request Packet: (**01 08 00 03 02 61 69 96**)

REQA request (61_{16}) Example

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	61	Send REQA
BCC	69 96	LRC and ~LRC

Response Packet: (**01 0B 00 03 02 61 00 04 00 6E 91**)
The response will return the two-byte ATQA (04 00) LS byte first.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0B 00	Packet Length 11 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	61	Response to REQA
Status Byte	00	Successful
ATQA	04 00	ATQA from the token
BCC	6E 91	LRC and ~LRC

WUPA request (62_{16}) Example
Request Packet: (**01 08 00 03 02 62 6A 95**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	62	Send WUPA
BCC	6A 95	LRC and ~LRC

Response Packet: (**01 0B 00 03 02 62 00 04 00 6D 92**)
The response will return the two-byte ATQA (04 00) LS byte first.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0B 00	Packet Length 11 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	62	Response to WUPA
Status Byte	00	Successful
ATQA	04 00	ATQA from the token
BCC	6D 92	LRC and ~LRC

HLTA Request (63_{16})

The host application can request the ISO14443A Module to send a HLTA packet over the RF interface according to the ISO14443A-2 standard using this packet. After a successful HLTA execution the MFR Module Terminal responds with **ERROR_NONE** in the Response Status byte field. According to ISO14443A it is considered an error if the token sends any RF modulation within 1 ms after the HLTA downlink ends. This shows up as a Manchester decode error. However, if there are no transponders to accept the HLTA request, the reader could decode noise and return a Manchester error. The ISO14443A HLTA request is handled by the function ISO14443A_Handle_HLTA ().

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	Entity ID
Cmd2	1	63	HLTA request

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	ISO14443A Entity ID
Cmd2	1	63	HLTA request
Status	1	00 – FF	Standard Error Codes

HLTA request (61_{16}) Example
Request Packet: (**01 08 00 03 02 63 6B 94**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	63	Send HLTA
BCC	6B 94	LRC and ~LRC

Response Packet: (**01 09 00 03 02 63 00 6A 95**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	63	Response to HLTA
Status Byte	00	Successful
BCC	6A 95	LRC and ~LRC

Alternate response Packet: (**01 09 00 03 02 63 57 3D C2**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	63	Response to HLTA
Status Byte	57	ERROR_COLLISION_DETECT
BCC	3D C2	LRC and ~LRC

ANTICOLLISION/SELECT Request (64_{16})

The Host application can request the ISO14443A Module to send either an ANTICOLLISION or a SELECT packet over the RF interface according to the ISO14443A-2 standard using this packet.

The request packet specifies the cascade level of the UID, the number of bits to send to the token(s) using ANTICOLLISION/SELECT frame and the actual data bits/bytes to send. The ANTICOLLISION request is sent over the RF interface in a bit-oriented anti-collision frame according to ISO14443A layer 3 specification. The SELECT request is sent over the RF interface in a "Standard" frame according to ISO14443A-3 specification. The ANTICOLLISION request may specify # bits in the range 0 through 39 i.e. [0, 39]. The SELECT request must always specify 40 bits to send. Even if the #bits specified is less than 40, five bytes of data must follow. A successful REQA/ATQA exchange must be executed before ANTICOLLISION/SELECT can be attempted. The complete UID must be collected from the token before SELECT with 40 bits of UID can be attempted.

After a successful ANTICOLLISION/SELECT execution the MFR Module Terminal responds with **ERROR_NONE** in the Response Status byte field. The data field contains the sent data bits and the data bits of the UID that could be resolved up to any collision or up to the full UID.

Cascade level (CLn) definitions follow:

Cascade level 1: 00_{16}
Cascade level 2: 01_{16}
Cascade level 3: 02_{16}

The ISO14443A ANTICOLLISION/SELECT request is handled by the function ISO14443A_Handle_AC_Select ().

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	Entity ID
Cmd2	1	64	ANTICOLLISION/SELECT
CLn	1	00 - 02	**C**ascade **L**evel **n** (mandatory)
#bits	1	00 - 28	#bits sent by the PCD (mandatory)
DnData	5	00 - FF	Data to send the token (mandatory)

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	ISO14443B Entity ID
Cmd2	1	64	ANTICOLLISION/SELECT
Status	1	00 – FF	Standard Error Codes
ReplyDat Anti-collision[1]	2/7	00 – 0E	Byte 1: #bytes received
		00	2:# extra bits received
		00 – FF	3-7:received data

— OR —

ReplyDat	Select[2]	2/5	00 – 0E	Byte 1: #bytes received
			00	2:# extra bits received
			00 – FF	3:SAK per ISO14443A-3
			00 – FF	4-5:CRC_A over SAK

— OR —

ReplyDat	NoData[3]	0	—	(See <Status> field)

(1) Contents with valid Anti-collision response from the token.
(2) Contents with valid Select response from the token.
(3) No Data returned due to condition described in <*Status*> field.

ANTICOLLISION request (6416) Example

A packet with cascade level 1 and 0 bits of UID specified is shown below. PCD sends SEL and NVB only to token. The example assumes a certain UID as shown in the response packet. This is used again in the next example.

Request Packet: (**01 0F 00 03 02 64 00 00 00 00 00 00 00 6B 94**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0F 00	Packet Length 15 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	64	Send ANTICOLLISION
Cascade level	00	CL1, Cascade level 1
# bits of UID to send	00	PCD sends no UID bits
UID data	00 00 00 00 00	Always 5 bytes
BCC	6B 94	LRC and ~LRC

Response Packet: (**01 10 00 03 02 64 00 93 20 61 B0 28 65 9C C7 38**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	10 00	Packet Length 16 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	64	ANTICOLLISION response
Status Byte	00	Successful
Cascade level	93	Per ISO14443A-3
SEL	20	0 bits sent by PCD
UID at CL1	61 B0 28 65	UID at CL1
BCC over UID	9C	LRC over UID
BCC	C7 38	LRC and ~LRC

SELECT Request (6416) Example, with cascade level 1 and 40 bits of UID specified. PCD sends SEL and NVB only to token. Note that a certain value of UID has been assumed as evident from both examples.
Request Packet: (**01 0F 00 03 02 64 00 28 61 B0 28 65 9C 43 BC**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0F 00	Packet Length 15 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	64	Send SELECT
Cascade level	00	CL1, Cascade level 1
# bits of UID to send	28	PCD sends 40 UID bits
UID data	61 B0 28 65 9C	Always 5 bytes UID
BCC	43 BC	LRC and ~LRC

Response Packet: (**01 0C 00 03 02 64 00 88 BE 59 07 F8**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0C 00	Packet Length 16 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	64	Response to SELECT
Status Byte	00	Successful
SAK	88	Per ISO14443A-3
CRC over SAK	BE 59	CRC over SAK
BCC	07 F8	LRC and ~LRC

RATS (Request Answer To Select) Request (65_{16})

The host application can request the ISO14443A Module to send a RATS packet over the RF interface according to the ISO14443A-2 standard using this packet. Before RATS can be executed the token must have been through PICC activation (REQA/WUPA, ANTICOLLISION and SELECT). RATS is a part of the protocol activation, which is defined in ISO14443A-4 specification. *After a successful RATS execution the MFR Module Terminal responds with ERROR_NONE in the Response Status byte field. The ISO14443A RATS request will be handled by the function ISO14443A_Handle_RATS ().*

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	Entity ID
Cmd2	1	65	RATS

Response Packet:

	Field	Length (Bytes)	Value Range (Hex)	Description
	Cmd1	1	02	ISO14443A Entity ID
	Cmd2	1	65	RATS
	Status	1	00 – FF	Standard Error Codes
Reply	ATS[1]	n[2]	00 – FF	Byte 1: CID assigned to the token
			00 – FF	2-n: ATS data bytes
	— OR —			
	NoData[3]	0	—	(See <Status> field)

(1) Contents with valid ATS response from the token.
(2) Value of n depends on the ATS response from the token.
(3) No Data returned due to condition described in <*Status*> field.

RATS request (65_{16}) Example

Request Packet: (**01 08 00 03 02 65 6D 92**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	65	Send RATS
BCC	6D 92	LRC and ~LRC

Response Packet: (**01 xx xx 03 02 65 00 CID [ATS] LRC1 LRC2**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	xx xx	Packet Length
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	65	Response to RATS
Status Byte	00	Successful
CID	CID	CID assigned to the token
ATS bytes	(ATS bytes)	ATS bytes
BCC	LRC1 LRC2	LRC and ~LRC

PPS (Protocol and Parameter Selection) Request (66_{16})

The Host application can request the ISO14443A Module to send a PPS packet over the RF interface according to the ISO14443A-2 standard using this packet. Before PPS can be executed the token must have been through PICC activation (REQA/WUPA, ANTICOLLISION, SELECT and RATS). PPS is a part of the protocol activation, which is defined in ISO14443A-4 specification.

After a successful PPS execution the MFR Module Terminal responds with ERROR_NONE in the Response Status byte field. The ISO14443A RATS request will be handled by the function ISO14443A_Handle_PPS ().

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	02	Entity ID
Cmd2	1	66	PPSS
CID	1	00 – 0E	CID of the token

Response Packet:

	Field	Length (Bytes)	Value Range (Hex)	Description
	Cmd1	1	02	ISO14443A Entity ID
	Cmd2	1	66	PPSS
	Status	1	00 – FF	Standard Error Codes
Reply	PPSS[1]	4	00 – FF	Byte 1: CID assigned to the token
			00 – FF	2: PPSS byte
			00 – FF	3-4: CRC_A over PPSS byte
			— OR —	
	NoData[2]	0	—	(See <Status> field)

(1) Contents with valid PPSS response from the token.
(2) No Data returned due to condition described in <*Status*> field.

PPS request (66_{16}) Example

Request Packet: (**01 09 00 03 02 66 01 6E 91**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	66	Send PPS
CID	01	CID of the token
BCC	6E 91	LRC and ~LRC

Response Packet: (**01 0D 00 03 02 66 00 CID PPSS CRC_A1 CRC_A2 LRC1 LRC2**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0D 00	Packet Length
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	66	Response to PPS
Status Byte	00	Successful
CID	CID	CID assigned to the token
PPSS	(PPSS byte)	PPSS byte
CRC_A over PPSS	CRC_A1, CRC_A2	CRC_A over PPSS
BCC	LRC1 LRC2	LRC and ~LRC

Chapter 2

ISO 14443-3 Type-B

Topic **Page**

2.1 ISO 14443-3 Type B Overview

The following sections define and detail the Protocol functionality in the 14443 B Module of the MFR Base Application. This information includes 14443 B Protocol Commands and the data/parameters associated with them.

Find Token Request (41_{16})

The Host application can send the MFR Base Application a Request Packet to check if a token is present. This packet contains a loop count that sets the maximum number of times the MFR Module Base Application will search for the token. This function allows a great deal of flexibility. It is possible to search for a variety of transponders or a single type of token. If multiple transponders are selected, the function returns the first token that it reads. The Application continues to loop as specified unless it detects a token. The Application doesn't return every type of token - just the first one it reads.

After a successful read, the MFR Module Base Application responds with **ERROR_NONE** in the <*Status*> followed by token's RF Technology Type and token data. The RF poll stops once a valid token is found. If a valid token is not found within the number of loops selected, the MFR Module Base Application responds with an **ERROR_TOKEN_NOT_PRESENT** <*Status*> field. The RF Poll stops once the loops are complete.

The ISO14443B token search is handled by the function Find_Token_14443B(). Note that this function is not called when ISO14443B is not in the Priority table unless the request is directed specifically to the ISO14443B library. It is not called if a different token is discovered prior to reaching the ISO14443B format in the priority table.

Information:

- The Find Token request is implemented internally as multiple block calls of REQB/WUPB followed by ATTRIB requests for each token found unambiguously (no collisions).
- ATQB and Answer to ATTRIB responses set specific parameters of internal token structures and key off of these structures for the REQB, WUPB, HLTB and ATTRIB request packet parameters.
- ISO14443B – 4 layer packets (I, R and S blocks) also use the ATQB for the particular token at hand to determine whether to use CID and/or NAD.
- The ISO14443B Library Entity limits the number of ISO14443B transponders reported to 16 and concatenates their responses; if more transponders are detected, the entity responds with ERROR_COLLISION_DETECT.

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	01 / 03	Entity ID
Cmd2	1	41	Find Token
LoopCount	1	00 – FF	Number of attempts to find token (00 ⇒ Loop until next request)

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	01 / 03	Entity ID
Cmd2	1	41	Find Token
Status	1	00 – FF	Standard Error Codes
EntityId	1	03	ISO14443B Library
ToknDat 1	0 / 10	00 – 0E	Byte 1: Card IDentifier
		00 – FF	2-5: Pseudo Unique PICC Identifier
≡			
ToknDat n[(1)]	0 / 10	00 – FF	Byte 1: Inventory Response Flags
		00 – FF	2-5: Pseudo Unique PICC Identifier

(1) $n \leq 16$

Find Token Example:
The request packet specifies 10 loops.

Request Packet: (**01 09 00 03 03 41 0A 43 BC**)

Field	Contents	Summary
SOF	01	Start of Frame
PacketLen	09 00	Packet Length 9 bytes
DeviceID	03	Device is MFR Module
Cmd1	03	ISO14443B Entity ID
Cmd2	41	Find Token Request
LoopCount	0A	10 loops maximum
BCC	43 BC	LRC and ~LRC

Response Packet: (**01 0F 00 03 03 41 00 03 00 34 03 04 09 76 89**)

The response returns information for up to 16 transponders and indicates collisions, if any. For each token found unambiguously (without a collision in its response slot), 5 bytes of data are reported. The first byte is the CID assigned by the PCD (reader) to the token for the session (for example, the duration between

transmitter-on through transmitter-off). The next 4 bytes are the PUPI per ISO14443B. The PUPI by its definition is a pseudo random number and MS byte - LS byte distinctions are not relevant as long as the byte sequence order is maintained when communicating with the PCD.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0F 00	Packet Length 15 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	41	Find Token Request
Status Byte	00	Successful
Entity ID	03	Library Layer – ISO14443B
CID	00	CID per ISO14443B
PUPI	34 03 04 09	PUPI per ISO14443B
BCC	76 89	LRC and ~LRC

Pass-Through Request (45_{16})

The Pass-Though Request Packet provide a way for a host to have direct communications to an Entity Module library. The Pass-Through Request packet specifies the Entity to direct the data to and then provides data that the library can send directly to its token. This function provides a way to access the low level RF protocol directly, bypassing any abstraction provided by the library and Application Layer.

The ISO14443B library module/entity allows direct access to the RF interface via the pass-through request. When the ISO14443B module receives a pass-through request, the module configures the HF ASIC to the ISO14443B-2 specific RF scheme. The ISO14443B library supports three levels of pass-through functionality. The first data byte in the data field of the request packet to the MFR reader, <*PT Option*>, specifies the pass-through support level and is mandatory.

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	Entity ID
Cmd2	1	45	Pass Through
PT Option	1	00-02	Pass-through option
Data	0 – n(1)	00 – FF	Data to send via RF channel

(1) Length ‘n’ depends on RF message

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	Entity ID
Cmd2	1	45	Pass Through
Status	1	00 – FF	Standard Error Codes
Data	0 – n(1)	00 – FF	ISO14443B reply data (if any received)

(1) Length 'n' depends on RF message

Information:

The library only supports PCD to PICC data rate of 106 kbps. The host must make sure that the data rate specified in the ATTRIB pass-through command is set to 106 kbps for the PCD to PICC transmissions.

Pass-through level 0 (PT0)

If the value of the <*PT Option*> is 0, PT0 is assumed. In this mode all the data bytes that follow the pass-through level indicator byte are sent to the token via ISO14443B-2 type RF modulation scheme. The data in the pass-through request (except for the <*PT Option*> byte) is not interpreted and it has meaning only to the token that may receive it. Therefore the process that initiated the request must know the structure and data content of the transponder protocol. Some of these transponder protocol details are specified in ISO14443B-3 and ISO14443-4 specification documents. Application and security layer commands may also be sent to the token via the pass-through request as long as the calling process understands and follows the states of the token at hand.

PT0 always uses default values for Frame Wait Index and **F**rame **W**ait **T**ime Extension. Due to this, if at the ISO14443-4 or at the application layer, a different FWT is negotiated using S (WTXM) packets via PT0, the reader will not be able to adjust to it. For such cases, use PT1 or PT2 as a workaround.

This allows the host full control over the HF ASIC and what it sends. The data must strictly follow the ISO14443B-3 and ISO14443-4 protocol.

Pass-Through level 0 REQB example:

Send a valid 14443-B level 0 Pass-through REQB Request Packet to the MFR Module device with the COM port set to 9600 8N1 and validate the data received. Ensure a valid ISO14443B type token is in the field when the request is issued. This example specifies that the MFR reader should not attempt to handle S (WTXM) packets automatically and that the REQB should have 1 slot only. It means that there is no need to send any slot markers after the REQB is sent. Any transponders that are in the field should respond with an ATQB after the REQB downlink per ISO14443B specification.

Request Packet: (**01 0E 00 03 03 45 00 05 00 00 71 FF C1 3E**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0E 00	Packet Length 14 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	45	Pass-through Request
PT Option	00	PT0
Data	05 00 00 71 FF	REQB request – 1 slot
BCC	C1 3E	LRC and ~LRC

Response Packet: (**01 17 00 03 03 45 00 50 34 03 04 09 63 22 33 44 00 00 02 FA 07 F0 0F**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	17 00	Packet Length 23 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	45	Pass-through response
Status Byte	00	Successful
Data	50	ATQB start byte
Data	34 03 04 09	PUPI
Data	63 22 33 44	Application data
Data	00 00 02	Protocol info
Data	FA 07	CRC for ATQB
BCC	F0 0F	LRC and ~LRC

Pass-Through level 0 WUPB example:

Send a valid 14443-B level 0 Pass-through WUPB Request Packet to the MFR device with the COM port set to 9600 8N1 and validate the data received. Ensure a valid ISO14443B type token is in the field when the request is issued. This example specifies 1 slot only. It means that there is no need to send any slot markers after the WUPB is sent. Any transponders that are in the field should respond with an ATQB after the WUPB downlink per ISO14443B spec.

Request Packet: (**01 0E 00 03 03 45 00 05 00 08 39 73 0D F2**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0E 00	Packet Length 14 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	45	Pass-through Request
PT Option	00	PT0
Data	05 00 08 39 73	WUPB request – 1 slot
BCC	0D F2	LRC and ~LRC

Response Packet: (**01 17 00 03 03 45 00 50 34 03 04 09 63 22 33 44 00 00 02 FA 07 F0 0F**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	17 00	Packet Length 23 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	45	Pass-through Response
Status Byte	00	Successful
Data	50	ATQB start byte
Data	34 03 04 09	PUPI
Data	63 22 33 44	Application data
Data	00 00 02	Protocol info
Data	FA 07	CRC for ATQB
BCC	F0 0F	LRC and ~LRC

Pass-Through level 0 Slot Marker example:

Slot Markers can be sent only if the REQB/WUPB request that precedes the slot markers, specify more than 1 slot per ISO14443B. Slot Markers can range from 1 through 15.

Send a valid WUPB/REQB command specifying 4 slots to the MFR Module device with the COM port set to 9600 8N1 and validate the data received. Ensure a valid ISO14443B type token is in the field when the request is issued. The duration after the WUPB/REQB request itself is the first slot. Three slot markers may be sent next with the appropriate slot numbers. Now send valid 14443-B level 0 Pass-through Slot Marker Request Packets for slot numbers 1,2 and 3.

Request Packet: (**01 0C 00 03 03 45 00 15 54 B7 BE 41**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0C 00	Packet Length 12 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	45	Pass-through Request
PT Option	00	PT0
Data	15 54 B7	Slot Marker #1
BCC	BE 41	LRC and ~LRC

Response Packet: (**01 17 00 03 03 45 00 50 34 03 04 09 63 22 33 44 00 00 02 FA 07 F0 0F**)
This is the case where the token responds with ATQB.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	17 00	Packet Length 23 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	45	Pass-through Response
Status Byte	00	Successful
Data	50	ATQB start byte
Data	34 03 04 09	PUPI
Data	63 22 33 44	Application data
Data	00 00 02	Protocol info
Data	FA 07	CRC for ATQB
BCC	F0 0F	LRC and ~LRC

Pass-Through level 0 ATTRIB example:

Send a valid Find 14443-B type Token Pass-through ATTRIB Request Packet to the MFR Module device with the COM port set to 9600 8N1 and validate the data received. Ensure a valid ISO14443B type token is in the field when the request is issued. Send a REQB/WUPB and receive a valid ATQB prior to sending the ATTRIB request. ATTRIB request is always addressed to a specific token by its PUPI. Response to the ATTRIB request is an Answer to ATTRIB. Once a token sends an Answer to ATTRIB, the token goes to the ACTIVE state and only responds to the layer 4 packets. The Request Packet assumes PUPI value of $(\mathbf{34\ 03\ 04\ 09})_{16}$.

Information:

The library only supports a PCD to PICC data rate of 106 kbps. The Host must ensure that the data rate specified in the ATTRIB pass-through command is set to 106 kbps for the PCD to PICC transmissions.

Request Packet: (**01 14 00 03 03 45 00 1D 34 03 04 09 00 08 00 00 93 97 7B 84**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	14 00	Packet Length 20 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	45	Pass-through Request
PT Option	00	PT0
Data	1D 34 03 04 09 00 08 00 00 93 97	ATTRIB request
BCC	7B 84	LRC and ~LRC

Response Packet: (**01 0C 00 03 03 45 00 00 78 F0 C0 3F**)
This is the case where the token acknowledges the ATTRIB.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0C 00	Packet Length 12 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	45	Pass-through Response
Status Byte	00	Successful
Data	00	Answer to ATTRIB
Data	78 F0	CRC of the Answer to ATTRIB
BCC	C0 3F	LRC and ~LRC

Pass-Through level 0 HLTB example:

Send a valid Find 14443 B type Token Pass-through HLTB Request Packet to the MFR device with the COM port set to 9600 8N1 and validate the data received. Ensure a valid ISO14443B type token is in the field when the request is issued. Send a REQB/WUPB and receive a valid ATQB prior to sending the HLTB request. A HLTB request is always addressed to a specific token by its PUPI. Response to the HLTB request is an Answer to HLTB and the Answer to HLTB will acknowledge that the token was halted. Any halted token can only be woken up by a WUPB request as long as the carrier is present.

Request Packet: (**01 10 00 03 03 45 00 50 34 03 04 09 CE 91 61 9E**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	10 00	Packet Length 16 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	45	Pass-through Request
S(WTXM) handling	00	MFR does not handle S (WTXM) automatically.
Data	50 34 03 04 09 CE 91	HLTB request
BCC	61 9E	LRC and ~LRC

Response Packet: (**01 0C 00 03 03 45 00 00 78 F0 C0 3F**)
This is the case where the token acknowledges the HLTB.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0C 00	Packet Length 12 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	45	Pass-through Response
Status Byte	00	ERROR_NONE
Data	00	HLTB acknowledgement
Data	78 F0	CRC for ATQB
BCC	C0 3F	LRC and ~LRC

Response Packet: (**01 09 00 03 03 45 45 08 F7**)
This response packet is the case when no token responds in this slot.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	45	Pass-through Request
Status Byte	45	ERROR_BPSK_TR0_TIME OUT
BCC	08 F7	LRC and ~LRC

Pass-through level 1 (PT1)

If the value of the *<PT Option>* is 1, PT1 is assumed. In this mode the MFR reader looks into the token's response and if it detects an S (WTXM) packet, it will send an S (WTXM) packet back to the token. When the token finally returns with an I-block or R-block response, the response is returned to the calling entity. If there is no response or if any error occurs, an error code is returned as appropriate. It is our recommendation that PT1 be used for ISO14443-4 layer or application layer packets only. ISO14443-3 layer packets should be used with PT0 packets if needed via pass-through.

S (WTXM) packet is an S-block according to the ISO14443-4 document. The token sends an S (WTXM) packet in response to an I-block packet if the token needs more time than is allowed by the Wait Time Index. The reader has to acknowledge with an S (WTXM) packet downlink so that the token can proceed to perform the task requested.

Pass-Through level 1 - TI Apollo GetUID request (45_{16}) example:

Ensure that a valid TI Apollo token is present in the antenna field. Send a valid WUPB request (either using pass-through packet or REQB command) and make sure that a valid ATQB is received. Next execute a valid ATTRIB request and Answer to ATTRIB request/response sequence. Now the token is operating at ISO14443B layer 4.

Now send a valid pass-through level 1 – TI Apollo GetUID request and validate the response.

Request Packet: (**01 12 00 03 03 45 01 05 01 B0 30 00 00 09 4F F7 62 9D**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	12 00	Packet Length 18 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	45	Send Pass-through
Pass-through level	01	PT1 - MFR handles S(WTXM)
I-block prologue	05 01	PCB, CID
APDU	B0 30 00 00 09	TI Apollo GetUID
I-block epilogue	4F F7	CRC_B over I-block
BCC	62 9D	LRC and ~LRC

Response Packet: (**01 16 00 03 03 45 00 05 01 07 08 07 06 05 04 03 02 01 6E AA 9D 62**)
The response shown below returns the UID response APDU embedded in an I-block.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	16 00	Packet Length 22 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	45	Response to Pass-through
Status Byte	00	Successful
I-block prologue	05 01	PCB, CID
APDU	07	TI Apollo Life Cycle
APDU	08 07 06 05 04 03 02 01	TI Apollo UID
I-block epilogue	6E AA	CRC_B over I-block
BCC	9D 62	LRC and ~LRC

Pass-through level 2 (PT2)

If the value of the <*PT Option*> is 2, PT2 is assumed. In this mode the MFR reader manages the ISO14443-4 (transport layer packet types I and S blocks). R-blocks are not handled in this option. The data bytes that follow the <*PT Option*> byte are CID, NAD, Chaining option and an APDU packet. PT2 option always sends an I-block to the token. It is our recommendation that PT1 be used for application layer packets only. Lower layer packets may be sent using either PT0 or PT1 as permitted under those options (as described in pervious sections for PT0 and PT1 modes).

CID, NAD and Chaining option bytes are mandatory. CID specifies the Card IDentifier per ISO14443-4 specification. NAD specifies **N**ode **AD**dress per ISO14443-4 specification. Chaining option byte specifies whether the I-block packet initiated should indicate the chaining option to the token. Note that only an I-block can indicate the Chaining option. Since R-blocks are not handled in this option, chaining option byte shall be set to 00_{16}.

The APDU packet is a command that the Security/Application layer of the token may understand. This is the part of the packet that is not verified by the MFR reader. An ISO14443-4 I-block is constructed using the CID and NAD information. The APDU bytes are placed into the INF field of the ISO14443-4 I-block.

Pass-Through level 2 - TI Apollo GetUID request (45_{16}) example:

Ensure that a valid TI Apollo token is present in the antenna field. Send a valid WUPB request (either using pass-through packet or REQB command) and make sure that a valid ATQB is received. Next execute a valid ATTRIB request and Answer to ATTRIB request/response sequence. Now the token is operating at ISO14443B layer 4.

Now send a valid pass-through level 2 – TI Apollo GetUID request and validate the response.

Request Packet: (**01 12 00 03 03 45 01 05 01 B0 30 00 00 09 4F F7 62 9D**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	12 00	Packet Length 18 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	45	Send Pass-through
Handle S(WTXM)	01	MFR handles S(WTXM)
I-block prologue	05 01	PCB, CID
APDU	B0 30 00 00 09	TI Apollo GetUID
I-block epilogue	4F F7	CRC_B over I-block
BCC	62 9D	LRC and ~LRC

Response Packet: (**01 16 00 03 03 45 00 05 01 07 08 07 06 05 04 03 02 01 6E AA 9D 62**)
The response shown below returns the UID response APDU embedded in an I-block.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	16 00	Packet Length 22 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	45	Response to Pass-through
Status Byte	00	ERROR_NONE
I-block prologue	05 01	PCB, CID
APDU	07	TI Apollo Life Cycle
APDU	08 07 06 05 04 03 02 01	TI Apollo UID
I-block epilogue	6E AA	CRC_B over I-block
BCC	9D 62	LRC and ~LRC

Transmitter On Request (48_{16})

Transmitter On Request is used to turn the transmitter ON, for a specific entity. The request packet specifies in the <*Cmd1*> field the intended entity. The application layer firmware redirects the request to the intended entity. The entity then sets up the appropriate hardware and software configurations needed to implement its specific details and then turns the transmitter on.

The ISO14443B Transmitter On request firstly configures the hardware ports to their proper data directions (input/output), sets up pull-up resistors for input ports and sets up the correct logic levels at the output ports. The transmitter itself is turned on and then the entity communicates with the HF ASIC to set up the ISO14443B protocol related register settings in the HF ASIC.

The request packet has no data bytes. The <*Cmd1*> field indicates the entity intended to turn on the transmitter. The response status byte indicates the success/error status.

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	Entity ID
Cmd2	1	48	Transmitter On

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	ISO14443B Entity ID
Cmd2	1	48	Transmitter On
Status	1	00 – FF	Standard Error Codes

Request Packet: (**01 08 00 03 03 48 41 BE**)

Transmitter ON Example:

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	48	Transmitter On Request
BCC	41 BE	LRC and ~LRC

Response Packet: (**01 09 00 03 03 48 00 40 BF**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	48	Transmitter On Request
Status Byte	00	Successful
BCC	40 BF	LRC and ~LRC

Transmitter Off Request (49_{16})

The transmitter Off Request is used to turn the transmitter OFF for a specific entity. The request packet specifies in the <*Cmd1*> field the intended entity. The application layer firmware redirects the request to the intended entity. The entity then sets up the appropriate hardware and software configurations needed to implement its specific details and then turns the transmitter off.

The ISO14443B entity turns the transmitter off and returns a response. The request packet consists of no data bytes. The <*Cmd1*> field indicates the entity intended to turn off the transmitter. The response status byte indicates the success/error status.

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	Entity ID
Cmd2	1	49	Transmitter Off

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	ISO14443B Entity ID
Cmd2	1	49	Transmitter Off
Status	1	00 – FF	Standard Error Codes

Request Packet: (**01 08 00 03 03 49 40 BF**)

Transmitter OFF Example:

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	49	Transmitter Off Request
BCC	40 BF	LRC and ~LRC

Response Packet: (**01 09 00 03 03 49 00 41 BE**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR Module
Command 1	03	Library Layer – ISO14443B
Command 2	49	Transmitter Off Request
Status Byte	00	Successful
BCC	41 BE	LRC and ~LRC

REQB/WUPB Request ($61/62_{16}$)

The host application can request the ISO14443B Module to send a REQB/WUPB packet over the RF interface according to the ISO14443B-2 standard using this packet. REQB is sent if command 61_{16} is specified and WUPB is sent if command 62_{16} is specified. The data field contains one byte of “slot index” information according to the ISO14443B-3 standard. The Slot index (SI) is converted into #slot markers to be sent after the REQB/WUPB as follows:

#slot markers to be sent = ($2^{SI} - 1$)

Here, SI is in the range [0, 4]. Therefore #slot markers is in the range [0, 15]. The slot markers have to be sent separately by issuing the slot marker requests. After a successful REQB/WUPB execution the MFR Module Terminal responds with **ERROR_NONE** in the Response Status byte field and a one-byte CID followed by token's ATQB.

The ISO14443B REQB/WUPB request is handled by the function ISO14443B_Handle_REQB_WUPB ().

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	Entity ID
Cmd2	1	61-62	REQB/WUPB
SI	1	00-04	# Slots to follow per formula above.

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	ISO14443B Entity ID
Cmd2	1	61-62	REQB/WUPB
Status	1	00 – FF	Standard Error Codes
ReplyDat ATQB[1]	14	00 – FF	Byte 1: Constant (always 50_{16})
		00 – FF	2-5: PUPI
		00 – FF	6-9: Application data
		00 – FF	10-12: Protocol information
		00 – FF	13-14: CRC_B over ATQB

— OR —

NoData[2]	0	—	(See <Status> field)

(1) Contents with valid ATQB response from the token.
(2) No Data returned due to condition described in <*Status*> field.

Find 14443 B Token REQB Example:

Send a valid Find 14443 B type Token Pass-through REQB Request Packet to the MFR Module device with the COM port set to 9600 8N1 and validate the data received. Ensure a valid ISO14443B type token is in the field when the request is issued. This example specifies 1 slot only. It means that there is no need to send any slot markers after the REQB is sent. Any transponders that are in the field should respond with an ATQB after the REQB downlink per ISO14443B-3 specification.

Request Packet: (**01 09 00 03 03 61 00 69 96**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	61	Send REQB
Number of slot	00	Slot index N is 0 => #slots =1
BCC	69 96	LRC and ~LRC

Response Packet: (**01 18 00 03 03 61 00 05 50 FF 00 00 80 00 EC 92 00 00 21 45 21 8D E4 1B**)

The response packet assumes that the token at the antenna was assigned CID of 0x05 and the PUPI is $(FF\ 00\ 00\ 80)_{16}$.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	18 00	Packet Length 24 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	61	Response to REQB
Status Byte	00	ERROR_NONE
CID assigned	05	CID assigned to the token
Constant	50	ATQB always returns this if successful
PUPI	FF 00 00 80	Pseudo-Unique PICC identifier
Application Data	00 EC 92 00	AFI = 00, CRC_B(AID) = EC 92, # applications = 00
Protocol Information	00 21 45	BitRate = 00(106 kb/s), MaxFrame = 2 (32 bytes) , Protocol type = 1(PICC compliant with 14443-4) , FWI = 4, ADC = 01(application is coded as described in clause 7.9.3), FO = 01(CID supported by token)
CRC	21 8D	CRC_B over ATQB
BCC	E4 1B	LRC and ~LRC

Find 14443 B Token WUPB Example

Send a valid Find 14443 B type Token Pass-through WUPB Request Packet to the MFR Module device with the COM port set to 9600 8N1 and validate the data received. Ensure a valid ISO14443B type token is in the field when the request is issued. This example specifies 1 slot only. It means that there is no need to send any slot markers after the WUPB is sent. Any transponders that are in the field should respond with an ATQB after the WUPB downlink per ISO14443B-3 specification.

Request Packet: (**01 09 00 03 03 62 00 6A 95**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	62	Send WUPB
Number of slot	00	Slot index N is 0 => #slots =1
BCC	6A 95	LRC and ~LRC

Response Packet: (**01 18 00 03 03 62 00 01 50 07 90 F9 FC 00 EC 92 00 00 21 45 CB 26 4F B0**)

The response packet assumes that the token at the antenna was assigned a CID of 0x01 and the PUPI is $(07\ 90\ F9\ FC)_{16}$.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	18 00	Packet Length 24 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	62	Response to WUPB
Status Byte	00	ERROR_NONE
CID	01	CID of the token
Constant	50	ATQB always returns this if successful
PUPI	07 90 F9 FC	Pseudo-Unique PICC identifier
Application Data	00 EC 92 00	AFI = 00, CRC_B(AID) = EC 92, Number of applications = 00
Protocol Information	00 21 45	BitRate = 00(106 kb/s), MaxFrame = 2 (32 bytes) , Protocol type = 1(PICC compliant with 14443-4) , FWI = 4, ADC = 01(application is coded as described in clause 7.9.3), FO = 01(CID supported by token)
CRC	CB 26	CRC_B over ATQB
BCC	4F B0	LRC and ~LRC

Slot Marker Request (63_{16})

The host application can request the ISO14443B Module to send a Slot Marker packet over the RF interface according to the ISO14443B-2 standard using this packet. REQB/WUPB specifying maximum # slot markers would have to be sent prior to sending slot marker packets. The data field of REQB/WUPB packet contains one byte of "slot index" information according to the ISO14443B-3 standard. Slot index (SI) is converted into #slot markers to be sent after the REQB/WUPB as follows:

#slot markers to be sent = $(2^{SI} - 1)$

Here, SI is in the range [0, 4]. Therefore #slot markers is in the range [0, 15]. The data field of the Slot Marker packet contains one byte of "Slot Number" in the range [0, 15]. After a successful Slot Marker execution the MFR Module Terminal responds with **ERROR_NONE** in the Response Status byte field and a one-byte CID followed by token's ATQB. The ISO14443B Slot Marker request will be handled by the function ISO14443B_Handle_Slot_Marker ().

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	Entity ID
Cmd2	1	63	Slot marker
SI	1	01 - 0F	Slot number (range depends on SI)

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	ISO14443B Entity ID
Cmd2	1	63	Slot marker
Status	1	00 – FF	Standard Error Codes
ReplyDat ATQB[1]	14	00 – FF	Byte 1: Constant (always 50_{16})
		00 – FF	2-5: PUPI
		00 – FF	6-9: Application data
		00 – FF	10-12: Protocol information
		00 – FF	13-14: CRC_B over ATQB

— OR —

NoData[2]	0	—	(See <Status> field)

(1) Contents with valid ATQB response from the token.
(2) No Data returned due to condition described in <*Status*> field.

Request Packet: (**01 09 00 03 03 63 02 69 96**)

Slot Marker Example

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	63	Send Slot marker
Slot number	02	Slot number
BCC	69 96	LRC and ~LRC

Response Packet: (**01 18 00 03 03 63 00 01 50 07 90 F9 FC 00 EC 92 00 00 21 45 CB 26 4E B1**)

The response packet assumes that the token at the antenna was assigned a CID of 0x01 and the PUPI is $(07\ 90\ F9\ FC)_{16}$.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	18 00	Packet Length 24 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	63	Response to Slot Marker
Status Byte	00	ERROR_NONE
CID	01	CID of the token
Constant	50	ATQB always returns this if successful
PUPI	07 90 F9 FC	Pseudo-Unique PICC identifier
Application Data	00 EC 92 00	AFI = 00, CRC_B(AID) = EC 92, Number of applications = 00
Protocol Information	00 21 45	BitRate = 00(106 kb/s), MaxFrame = 2 (32 bytes) , Protocol type = 1(PICC compliant with 14443-4) , FWI = 4, ADC = 01(application is coded as described in clause 7.9.3), FO = 01(CID supported by token)
CRC	CB 26	CRC_B over ATQB
BCC	4E B1	LRC and ~LRC

ATTRIB Request (64_{16})

The host application can request the ISO14443B Module to send an ATTRIB packet over the RF interface according to the ISO14443B-2 standard using this packet. A successful ATTRIB packet will take the token from layer 3 into layer 4. The data field of ATTRIB packet contains one byte of "CID" information according to the ISO14443B-3 standard.

After a successful Slot Marker execution the MFR Module Terminal responds with **ERROR_NONE** in the Response Status byte field and a one-byte CID followed by a 3-byte token's Answer to ATTRIB. Data field of the response packet contains the CID sent by the request packet as well as that indicated by the token. This is necessary since the token may not accept the CID assigned by the PCD. An example would be when the PCD assigns a non-zero CID but the token can only handle a zero value CID. ATTRIB response is the method of updating the CID assigned to the token at the PCD if a zero value CID is allowed by the PCD at the time. If the PCD has already assigned a zero value CID to another token, the PCD may report an error (ERROR_MANY_CID_NO_SUPRT_TRANSPONDERS) to indicate that there is more than one token that cannot support a non-zero CID. ATTRIB request sets up default data rates for both uplink and downlink, max frame size of 256 bytes, protocol type and the assigned CID in the downlink packet. The ISO14443B ATTRIB request will be handled by the function ISO14443B_Handle_ATTRIB ().

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	Entity ID
Cmd2	1	64	ATTRIB
TokenCID	1	00-0E	CID of the token to address

Response Packet:

	Field	Length (Bytes)	Value Range (Hex)	Description
Rply	Cmd1	1	03	ISO14443B Entity ID
	Cmd2	1	64	ATTRIB
	Status	1	00 – FF	Standard Error Codes
	MBLI[1]	3	00 – FF	Byte 1: MBLI + CID
			00 – FF	2-3: CRC_B over MBLI
	— OR —			
	NoData[2]	0	—	(See <Status> field)

(1) Contents with valid Answer to ATTRIB response from the token.
(2) No Data returned due to condition described in <*Status*> field.

Request Packet: (**01 09 00 03 03 64 02 6E 91**)

ATTRIB Request Example:

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	64	Send ATTRIB
CID	02	CID of the token
BCC	6E 91	LRC and ~LRC

Response Packet: (**01 0D 00 03 03 64 00 02 02 6A D3 D1 2E**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0D 00	Packet Length 13 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	64	Response to ATTRIB
Status byte	00	ERROR_NONE
CID	02	CID specified by the request packet
MBLI + CID	02	MBLI =0 and CID = 2 from the token
CRC_B	6A D3	CRC_B over MBLI byte
BCC	D1 2E	LRC and ~LRC

HLTB Request (65_{16})

The host application can request the ISO14443B Module to send a HLTB packet over the RF interface according to the ISO14443B-2 standard using this packet. The data field of REQB/WUPB packet contains one byte of “CID” information according to the ISO14443B-3 standard. After a successful Slot Marker execution the MFR Module responds with **ERROR_NONE** in the Response Status byte field and a 3-byte

answer to HLTB. The ISO14443B HLTB request will be handled by the function ISO14443B_Handle_Slot_Marker ().

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	03	Entity ID
Cmd2	1	65	HLTB
CID	1	00-0E	CID of the token to be halted

Response Packet:

	Field	Length (Bytes)	Value Range (Hex)	Description
	Cmd1	1	03	ISO14443B Entity ID
	Cmd2	1	65	HLTB
	Status	1	00 – FF	Standard Error Codes
Reply	ATQB[1]	3	00 – FF	Byte 1: Answer to HLTB
			00 – FF	2-3: CRC_B over Answer to HLTB
	— OR —			
	NoData[2]	0	—	(See <Status> field)

(1) Contents with valid Answer to HLTB response from the token.
(2) No Data returned due to condition described in <*Status*> field.

Request Packet: (**01 09 00 03 03 65 02 6F 90**)

HLTB Request Example:

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	65	Send HLTB
CID	02	CID of the token
BCC	6F 90	LRC and ~LRC

Response Packet: (**01 0C 00 03 03 65 00 00 78 F0 E0 1F**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0C 00	Packet Length 12 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	65	Response to HLTB
Status byte	00	ERROR_NONE
Answer to HLTB	00 78 F0	Answer to HLTB
BCC	E0 1F	LRC and ~LRC

Chapter 3

ISO 14443-4 Type-B

Topic **Page**

3.1 ISO 14443-4 Overview

The purpose of this chapter is to define those functions that allow the MFR Module to support ISO14443-4 Type-B level transponders. More specifically, the chapter focuses on the protocol of the Application Layer with reference to the functionality of the ISO14443-4 Type-B library.

I-Block Request (61_{16})

The host application can request the ISO14443 Layer 4 Module to send an I-block packet over the RF interface according to the ISO14443-2 standard using this packet. The I-block is an ISO14443-4 layer command used in the ACTIVE state. This is used to pass Security and Application layer commands between PCD and PICC. If the embedded APDU packet generates an S (WTXM) response from the PICC (token), the S (WTXM) downlink from the PCD to the PICC is NOT handled automatically.

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	07	ISO14443-4 Entity ID
Cmd2	1	61	I-block
CID	1	00 – 0E	CID of the token
NAD	1	00 - FF	NAD of the token
Chaining	1	00 - 01	Chaining required (01_{16}) or not (00_{16})
APDU	N(1)	00 - FF	APDU request (Application layer)

Response Packet:

	Field	Length (Bytes)	Value Range (Hex)	Description
	Cmd1	1	07	ISO14443-4 Entity ID
	Cmd2	1	61	I-block
	Status	1	00 – FF	Standard Error Codes
Reply	RespData(2)	N(2)	00 – 0E	Byte 1: CID of the token
			00 – FF	2: NAD of the token
			00 – FF	3-N: I-block data
			— OR —	
	NoData(3)	0	—	(See <Status> field)

(1) Contents and #bytes N depend on the request APDU.
(2) Contents and #bytes N depend on the response APDU.
(3) No Data returned due to condition described in <*Status*> field.

Notes:

- Chaining option is relevant only when I-blocks are sent.
- NAD usage rule "c" in ISO14443-4 section 7.1.3.3 states that during chaining the NAD shall be sent only in the first block of the chain. The calling function is responsible for making sure that this rule is adhered to.
- R-block responses are generated when chaining is used.
- While chaining I-blocks are supposed to ensure that the accumulated length is never greater than MBL. Does this mean that the PCD can send a bunch of I-blocks without waiting for the R-blocks? The protocol rules do not describe such an arrangement nor do the examples in the standard. So the PCD requires that the I-Block INF field's maximum length should be less than (Max_Frame_Size - ISO14443_4_OVHD) where ISO14443_4_OVHD (05_{16}) is the Frame overhead for any ISO14443 layer 4 packet. The token specifies the Max_Frame_Size value in the ATQB.

TI Apollo GetUID request (61_{16}) Example

Request Packet: (**01 10 00 03 07 61 01 00 00 B0 30 00 00 09 F5 0A**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	10 00	Packet Length 16 bytes
Device ID	03	Terminal is MFR
Command 1	07	Library ISO14443 Layer 4
Command 2	61	Send I-Block
Data	01 00 00	CID, NAD, Chaining
Data - APDU	B0 30 00 00 09	TI Apollo GetUID
BCC	F5 0A	LRC and ~LRC

Response Packet: (**01 1A 00 03 07 61 00 01 00 0A 81 03 07 06 05 04 03 02 01 00 11 BB 5D A2**)
The response will return the UID response APDU embedded in an I-block.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	1A 00	Packet Length 26 bytes
Device ID	03	Terminal is MFR
Command 1	07	Library ISO14443 Layer 4
Command 2	61	Response to I-Block
Status Byte	00	Successful
Data	01 00	CID, NAD
Data - APDU	0A 81	I-block: PCB, CID
Data - APDU	03	TI Apollo Life Cycle
Data - APDU	07 06 05 04 03 02 01 00	TI Apollo UID
Data - APDU	90 00	SW1, SW2
Data - APDU	11 BB	CRC1, CRC2
BCC	5D A2	LRC and ~LRC

R-Block Request (62_{16})

The host application can request the ISO14443 Layer 4 Module to send an R-block packet over the RF interface according to the ISO14443-2 standard using this packet. R-block is an ISO14443-4 layer command used in the ACTIVE state. This is used to **ACK**nowledge/**N**ot **A**c**K**nowledge the packets sent previously. This is used during large data transfers between the PCD and the PICC where multiple packets are required. ACK/NAK using R-blocks are used to keep track of the packets sent and confirm delivery of packets. R-blocks have no data in the INF field.

After a successful I-block execution the MFR Module Terminal responds with **ERROR_NONE** in the Response Status byte field, CID followed by the token's I-block/R-block response. R-block responses are generated when chaining is used.

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	07	ISO14443-4 Entity ID
Cmd2	1	62	R-block
CID	1	00 – 0E	CID of the token
ACK_NAK	1	00 / 80	ACK (0x00) or NAK (0x80)

Response Packet:

	Field	Length (Bytes)	Value Range (Hex)	Description
	Cmd1	1	07	ISO14443-4 Entity ID
	Cmd2	1	62	R-block
	Status	1	00 – FF	Standard Error Codes
Reply	ReplyBlock[1]	N[1]	00 – 0E	Byte 1: CID of the token
			00 – FF	2-N: I/R block data
	— OR —			
	NoData[2]	0	—	(See <Status> field)

[1] Contents and #bytes N depend on the request APDU.
[2] No Data returned due to condition described in <*Status*> field.

Notes:

- The ISO14443-4 library takes care of block numbering.
- Response to an R-block can be either an I-block or another R-block.

R-block ACK request (62_{16}) Example
Request Packet: (**01 0A 00 03 07 62 01 00 6C 93**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0A 00	Packet Length 10 bytes
Device ID	03	Terminal is MFR
Command 1	07	Library ISO14443 Layer 4
Command 2	62	Send R-Block
Data	01 00	CID, ACK
BCC	6C 93	LRC and ~LRC

S-Block Request WTXM (63_{16})

The host application can request the ISO14443 Layer 4 Module to send an S-block **W**ait **T**ime e**X**tension **M**ultiplier (WTXM) packet over the RF interface according to the ISO14443-2 standard using this packet. S-block is an ISO14443-4 layer command used in the ACTIVE state.

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	07	ISO14443-4 Entity ID
Cmd2	1	63	S-block WTXM
CID	1	00 – 0E	CID of the token
WTXM/DES	1	00	WTXM (00)
WTXM	1	01 - 59	WTXM value per ISO14443-4

Response Packet:

	Field	Length (Bytes)	Value Range (Hex)	Description
	Cmd1	1	07	ISO14443-4 Entity ID
	Cmd2	1	63	S-block WTXM
	Status	1	00 – FF	Standard Error Codes
Reply	RespData[1]	N[1]	00 – 0E	Byte 1: CID of the token
			00 – FF	2-N: I-block data
	— OR —			
	NoData[2]	0	—	(See <Status> field)

[1] Contents and #bytes N depend on the response APDU.
[2] No Data returned due to condition described in <*Status*> field.

Notes:

- The WTXM value sent to the token shall be identical to that sent by the token.
- S (WTXM) packets are always executed in pairs – one initiated by the token followed by an acknowledgement of the request by the reader.
- Block numbering is taken care of by the ISO14443-4 library.
- Response to an S (WTXM) downlink has to be an I-block uplink.

S-block WTXM downlink packet (63 hex) Example

ISO14443 token may request a time extension beyond the time provided by the **F**rame **W**ait **I**nteger (FWI) value, if the response time for the requested action is longer than that allowed by the FWI value.

As an example, let's consider an I – Block request for TI Apollo GetUID sent to a TI Apollo token using a pass-through level 0 packet. The token sends back a S(WTXM) packet requesting a WTX value of 30 hex as shown below:

Request: (**01 12 00 03 03 45 00 0A 01 B0 30 00 00 09 BB EE 81 7E**)
Response: (**01 0E 00 03 03 45 00 FA 81 B0 1D 19 85 7A**)

The S-block is: **FA 81 B0 1D 19**
Where the least significant 6 bits of the third byte (B0 hex) are the WTX multiplier. The value of the WTX multiplier turns out to be 30 hex in this case.

Now let's acknowledge the token's request with another S-block using the S-block request WTXM packet. The token computes and responds with an I-block containing the TI Apollo GetUID response APDU.

Request Packet: (**01 0B 00 03 07 63 01 01 30 5D A2**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0B 00	Packet Length 11 bytes
Device ID	03	Terminal is MFR
Command 1	07	Library ISO14443 Layer 4
Command 2	63	Send S-Block
Data	01 01 30	CID, WTXM, WTXM Value
BCC	5D A2	LRC and ~LRC

The response packet depends on the previous I-block request. The TI Apollo token's GetUID() response should be as below:

Response Packet: (**01 19 00 03 07 63 00 01 0A 81 01 50 84 64 E4 C0 00 01 07 90 00 67 E9 78 87**)

The response will return the UID response APDU embedded in an I-block.

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	19 00	Packet Length 25 bytes
Device ID	03	Terminal is MFR
Command 1	07	Library ISO14443 Layer 4
Command 2	63	Response to S-Block
Status Byte	00	Successful
Data	01	CID
Data - APDU	0A 81	I-block: PCB, CID
Data - APDU	01	TI Apollo Life Cycle
Data - APDU	50 84 64 E4 C0 00 00 01 07	TI Apollo UID
Data - APDU	90 00	SW1, SW2
Data - APDU	67 E9	CRC1, CRC2
BCC	78 87	LRC and ~LRC

S-Block Request DESELECT (64_{16})

The host application can request the ISO14443 Layer 4 Module to send an S-block DESELECT packet over the RF interface according to the ISO14443-2 standard using this packet. S-block is an ISO14443-4 layer command used in the ACTIVE state.

Request Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	07	ISO14443-4 Entity ID
Cmd2	1	64	S-block DESELECT
CID	1	00 – 0E	CID of the token
WTXM/DES	1	01 - FF	DESELECT (non-zero value)
WTXM	1	00 - FF	Dummy data (mandatory)

Response Packet:

Field	Length (Bytes)	Value Range (Hex)	Description
Cmd1	1	07	ISO14443-4 Entity ID
Cmd2	1	64	S-block DESELECT
Status	1	00 – FF	Standard Error Codes
RespData[1]	N[1]	00 – 0E	Byte 1: CID of the token
		00 – FF	2-N: I-block data
		— OR —	
NoData[3]	0	—	(See <Status> field)

Reply

[1] Contents and #bytes N depend on the response APDU.
[2] No Data returned due to condition described in <*Status*> field.

Notes:

- S (WTXM) packets are always executed in pairs.
- Block numbering is taken care of by the ISO14443-4 library.
- Response to an S (DESELECT) downlink has to be an S (DESELECT) uplink.
- Block handling rules for ISO14443-4 indicate that if the PCD does not receive any response to the S(DESELECT) request, the reader can continue as though the token has been deselected or try to deselect the token again.

S-block DESELECT request (63 hex) Example

Assuming CID = 01, the S-block request packet to DESELECT a token operating at ISO14443-4 layer is shown below:

Request Packet: (**01 0B 00 03 07 64 01 00 00 6B 94**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0B 00	Packet Length 11 bytes
Device ID	03	Terminal is MFR
Command 1	07	Library ISO14443 Layer 4
Command 2	64	Send S-Block DESELECT
Data	01 00 00	CID, DESELECT, WTXM Value
BCC	6B 94	LRC and ~LRC

The token may or may not respond with a response packet. The reader is allowed to either retry deselect packet or assume that the token has been deselected and continue on. A response packet when the token acknowledges the deselect packet is shown below:

Response Packet: (**01 0E 00 03 07 64 00 01 CA 81 1C AD 94 6B**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0E 00	Packet Length 14 bytes
Device ID	03	Terminal is MFR
Command 1	07	Library ISO14443 Layer 4
Command 2	64	Response to S-Block DESELECT
Status Byte	00	Successful
Data	01	CID
Data - APDU	CA 81	I-block: PCB, CID
Data - APDU	1C AD	CRC1, CRC2
BCC	94 6B	LRC and ~LRC

Chapter 4

Regulatory and Warranty Notices

Topic **Page**

4.1 FCC Conformity

The Series 4000 Multi-Function Reader is an intentional radiator. The transmitter portion operates at 13.56 MHz and is subject to FCC Part 15, Subpart C, “Intentional Radiator,” paragraph 15.225 (13.553-13.567MHz). Radiated emissions from the device are subject to the limits in Section 15.209 of the Rules outside of the 13.56 +/- 0.007 MHz band.

Note:

Any device or system incorporating the Series 4000 reader, in full or in part, needs to obtain FCC certification as part of the system within which this reader unit resides. A system containing this product may be operated only under an experimental license or final approval issued by the relevant approval authority. Before any such device or system can be marketed, an equipment authorization must be obtained form the relevant approval authority.

4.2 ETSI Conformity

Any device or system incorporating the Series 4000 reader, in full or in part, may need to comply with European Standard EN300330. It is the responsibility of each system integrator to have their complete system tested and to obtain approvals as required from the local authorities before operating or selling this system.

4.3 CE Conformity

Any device or system incorporating the Series 4000 reader, in full or in part, may need to have a CE Declaration of Conformity stating that it meets European EMC directive 99/5/EC. This must be issued by the system integrator or user of such a system prior to marketing or operating it in the European community.

4.4 Warranty and Liability

The "General Conditions of Sale and Delivery" of Texas Instruments Incorporated or a TI subsidiary apply. Warranty and liability claims for defect products, injuries to persons and property damages are void if they are the result of one or more of the following causes:

- Improper use of the reader module.
- Unauthorized assembly, operation and maintenance of the reader module.
- Operation of the reader modules with defective and/or non-functioning safety and protective equipment.
- Failure to observe the instructions during transport, storage, assembly, operation, maintenance and setting up of the reader modules.
- Unauthorized changes to the reader modules.
- Insufficient monitoring of the reader modules' operation or environmental conditions.
- Improperly conducted repairs.
- Catastrophes caused by foreign bodies and acts of God.