ISO15693 and ISO14443A/B Multi-Protocol HF Tag Reader RR3036 User's Manual V3.3

Contents

1. Communication Interface Specification

RR3036 series reader communicates with the host (MCU, MPU,Controller) using serial interface RS232 or RS485 and complete corresponding operation according to the host's command. The communication parameter is 19200bps, 1 start bit, 8 data bits, 1 stop bit without parity check bit. In process of the communication, the least significant bit of one byte is transmitted first and the least significant byte of command data sequence is transmitted first.

**The RR3036 series reader in this manual refers to RR3036SR, RR3036USB, RR3036CF, RR3036IP etc.*

2. Protocol Description

A communication procedure is sponsored by the host sending commands and data to the reader and the reader returns the result status and data to host after command execution. The following table shows the process of the host sending command:

HOST	DIRECTION	READER	COMMENT
Command Data Block	→		The interval between two consecutive bytes in the command data block should be less than 15ms. During command data block sending, synchronization will lost if the host receives any data from reader, the host should stop command sending and restart the communication after 15ms.

The command host sent to reader must be in the corresponding model. For example:, the reader must work in 15693 protocol model when a ISO/IEC 15693 command be sent by the host. The command data block the host sending to the reader should conform to the format of the protocol. The block includes reader address、operation command symbol、operation control symbol、command operand and CRC-16 checksum.

The reader completes command execution within 1s (Max., not including host sending data time) except inventory command after receiving host command and returns the results. During the period, it doesn't process any host data. The feedback of command execution results is as follows:

READER	DIRECTION	HOST	COMMENT
Response data block	→		The interval between two consecutive bytes in the response data block should be less than 15ms.

The response data block includes reader address, command execution result status and response data.

After the feedback, a whole communication process finishes.

3. Data Block Format

A. Command Data Block

Len	Com_adr	Cmd	State	Data[]	LSB-CRC16	MSB-CRC16

Len: Command data block length 1 byte (not including itself). The number of Len equals the length of Data[] plus 5.

Com_adr: Reader address, 1 byte. Value range is 0~254. Only will the reader conforming to the address response the command data block.. Value 255 is broadcasting address. All the readers will response to the command data block with a broadcasting address.

Cmd: Operation command symbol, 1 byte.

State: Operation control symbol, 1 byte. Low 4-bits control operation mode (refer to each command description for details); High 4-bits control operation style that value “0” means ISO/IEC 15693 protocol command, value “1” means ISO14443A/B protocol command.

Data[]: Operation command parameters. There is no parameters if the LEN item equals 5.

CRC16: CRC-16 checksum, 2 bytes with least significant byte first.

B. Respone Data Block

Len	Com_adr	Status	Data[]	LSB-CRC16	MSB-CRC16

Len: Response data block, 1 byte. The number of Len equals the length of Data[] plus 4.

Com_adr: Reader address, 1 byte. Value rang is 0~254.

Status: Result status value, 1byte. Refer to following chapters for details.

Data[]: Response data. There is no this item if Len equals 4.

CRC16: CRC-16 checksum, 2 bytes with least significant byte first.

The reader won’t response if any error found in command data block.

The default value of the reader address Com_adr is 0x00. The host may change it by using reader-defined command “Write Com_adr”.

C. Cyclic Redundancy Check

Cyclic Redundancy Check (CRC) computation includes all data from Len. The CRC generation polynomial is the same as that defined in ISO/IEC 15693 protocol but without reversing the computation result. For example, a data block is presented as follows: 0x05, 0xFF, 0x01, 0x00, LSB-CRC, MSB-CRC. CRC checksum is LSB-CRC＝0x5D，MSB-CRC＝0xB2. A reference CRC computation program is presented as follow:

C-Example:

```
#define   POLYNOMIAL 0x8408
#define   PRESET_VALUE 0xffff

int i,j;
unsigned int current_crc_value = PRESET_VALUE;

for(i=0;i<len;i++)  /*len=number of protocol bytes without CRC*/
```

```
{
   current_crc_value=current_crc_value^((unsigned
int)pData[i]);
     for(j=0;j<8;j++)
     {
     if(current_crc_value&0x0001)
     {
       current_crc_value=(current_crc_value>>1)^POLYNOMIAL;
     }
     else
     {
       current_crc_value=(current_crc_value>>1);
     }
     }
   }
// LSB-CRC16
pData[i++]=(unsigned char)(current_crc_value&0x00ff);
// MSB-CRC16
pData[i]=(unsigned char)((current_crc_value>>8)&0x00ff);
```

4. List Of Command Execution Result Status

RESPONSE DATA BLOCK					STATUS	COMMENT
Len	Com_adr	Status	Data[]	CRC-16		
Legnth of Data[] +4	0xXX	0x00	…	LSB+ MSB	Success	Return status 0 to host after command is executed successfully. Data block contains result data.
4	0xXX	0x01	—	LSB+ MSB	Command operand length error	Return status 1 to host when the number of command operands doesn't conform to the command request.
4	0xXX	0x02	—	LSB+ MSB	Command not supported	Return status 2 to host when the reader does not support the command the host sends.
4	0xXX	0x03	—	LSB+ MSB	Operand out of range	Return status 3 to host when one or more operand of command data block sent by host are out of range.
4	0xXX	0x04	—	LSB+ MSB	Operation Not Availible	Return status 4 to host when the requested operation is not available for the reader.

4	0xXX	0x05	—	LSB+ MSB	Inductive field closed	Return status 5 to host when the inductive field is closed and the host sends a ISO15693 ISO14443 protocol command.
4	0xXX	0x06	—	LSB+ MSB	EEPROM operation error	Return status 6 to host when the reader encounters error in EEPROM access.
4	0xXX	0x0A	—	LSB+ MSB	ISO15693 Inventory Operation Error	Return status 0x0A when the reader executing an ISO15693 Inventory command does not get one complete tag's UID before InventoryScanTime overflows.
4	0xXX	0x0B	—	LSB+ MSB	ISO15693 Inventory Operation Error	Return status 0x0B when the reader executing an ISO15693 Inventory command does not get all tags' UIDs before InventoryScanTime overflows.
4	0xXX	0x0C	—	LSB+ MSB	ISO15693 Tag Response Error	Return status 0x0C when the reader finds one or more tag response in a way that is not compatible with ISO15693 protocol definition.
4	0xXX	0x0E	—	LSB+ MSB	ISO15693 Operation No Tag Error	Return ox0E when the reader finds no active tag in the inductive field.
5	0xXX	0x0F	Error_ code	LSB+ MSB	ISO5693 Operation Extension error	Return status 0x0F when an error occurred in ISO15693 command execution and the further information of the error is defined by the Error_code in response data block.
5	0xXX	0x10	Error_ code	LSB+ MSB	ISO14443A Operation error	Return status 0x10 when an error occurred in ISO14443A command execution and the further information of the error is defined by the Error_code in response data block.
5	0xXX	0x1B	Error_ code	LSB+ MSB	ISO14443B Operation error	Return status 0x1B when an error occurred in ISO14443B command execution and the further information of the error is defined by the Error_code in response data block.

5	0xXX	0x1F	-	LSB+ MSB	Protocol model error	Return status 0x1F when the reader accepts a command not conforming to its current protocol model. For example, the reader accepts a ISO14443A protocol command but its current model is ISO15693.

- Remark：The length (Len) of response data block when a command finished successfully will vary from command to command. Further information could be found in following detailed description of individual command.
- Remark：When the reader return status 0x0F or 0x10 or 0x1B, the Error_code in response data block indicate further explanation to the error occurred in command execution. The definition of Error_code is available in the following chapter.

5. Error_Code Definition

A. ISO15693 protocol operation extension error (when return status=0x0f):

ErrorCode	Description
0x01	Commands not support. For example: invalid command code
0x02	Commands can not be identified. For example: invalid command format
0x03	Operation not supported
0x0f	Unknown error
0x10	Appointed block is not available or don't exist.
0x11	Appointed block has been locked and can't be locked again.
0x12	Appointed block is locked and can't change its content.
0x13	Appointed block does not operate normally.
0x14	Appointed block can't be locked normally.
0xA0~0xDF	User-defined error code.
All others	Reserved

B. ISO/IEC14443A protocol operation error (when return status=0x10) :

ErrorCode	Description
0x10	Halt failed
0x20	No ISO14443A card in the inductive area.
0x21	select failed
0x22	authentication failed
0x23	read failed
0x24	write failed
0x25	e-wallet initialization failed
0x26	read value failed

0x27	decrement/Increment failed
0x28	transfer failed
0x29	write/read E2PROM failes
0x2A	load key failed
0x2B	checkwrite failed
0x2C	data for checkwrite error
0x2D	value operation failed
0x2E	Ultralight card write failed
0x30	Anti-collision failed
0x31	Multiple card entering inductive area forbidden
0x32	Mifare I and Ultralight collision error
0x33	Ultralight card collision failed.

C. ISO/IEC14443B protocol operation error (when return status=0x1B) :

ErrorCode	Description
0x34	No ISO14443B card in the inductive area.
0x35	select failed
0x36	halt failed
0x37	execute transparent command failed
0x38	Anticollision failed

6.Detailed Description Of Operation Command

6.1 Reader-Defined Commands

As to the reader-defined commands, the command symbol byte (cmd) should be "0x00" and the operation control symbol byte (state) decides which reader-defined command will be processed.

6.1.1 Get Reader Information

The host sends this command to get the reader's information including reader's address(Com_adr), firmware version, reader type(_reader_type), supported protocol(_tr_type) and InventoryScanTime value(default value is 0x1e for 3s).

<table>
<tr><th>Len</th><th>Com_adr</th><th>Cmd</th><th>State</th><th>Data[]</th><th colspan="2">CRC-16</th></tr>
<tr><td>0x05</td><td>0xXX</td><td>0x00</td><td>0x00</td><td>—</td><td>LSB</td><td>MSB</td></tr>
<tr><th>Len</th><th>Com_adr</th><th>Status</th><th colspan="2">Data[]</th><th colspan="2">CRC-16</th></tr>
<tr><td>0x0c</td><td>0xXX</td><td>0x00</td><td colspan="2">Version(2bytes),RFU(2bytes)
_reader_type(1byte),_tr_type(2bytes),
_InventoryScanTime(1byte)</td><td>LSB</td><td>MSB</td></tr>
</table>

The reader type is 0x45, stand for RR3036 reader; The supported protocol byte is reserved to 0x00 and 0x0C as follows:

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Function	—	—	—	—	—	—	—	—
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Function	—	—	—	—	ISO/IEC 15693	ISO/IEC 14443	—	—

6.1.2 Close RF

The host sends this command to turn off the RF output of the reader.

Len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC-16**	
0x05	0xXX	0x00	0x01	—	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	—		LSB	MSB

6.1.3 Open RF

The host sends this command to turn on the RF output of the reader and establish the inductive field. The RF is open when the reader is powered on.

Len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC-16**	
0x05	0xXX	0x00	0x02	—	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	—		LSB	MSB

6.1.4 Write Com_adr

The host sends this command to change the address(Com_adr) of the reader. The address data is stored in the reader's inner EEPROM and is nonvolatile after reader powered off. The default value of Com_adr is 0x00. The range of Com_adr is 0x00~0xFE. When the host tries to write 0xFF to Com_adr, the reader will set the value to 0x00 automatically.

Len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC-16**	
0x06	0xXX	0x00	0x03	_Com_adr	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	—		LSB	MSB

6.1.5 Write InventoryScanTime

The host sends this command to change the value of InventoryScanTime of the reader. The value is stored in the reader's inner EEPROM and is nonvolatile after reader powered off.. The default value is 0x1E (corresponding to 30*100ms=3s). The value range is 0x03~0xFF(corresponding to 3*100ms~255*100ms). When the host tries to set value 0x00~0x02 to InventoryScanTime, the reader will set it to 0x03 automatically. In various environments, the actual inventory scan time may be 0~75ms longer than the InventoryScanTime defined.

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x06	0xXX	0x00	0x04	_InventoryScanTime	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	—		LSB	MSB

6.1.6 Change to ISO15693 protocol model

Change reader to ISO15693 model. When powered up, the reader enter this model as default.

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x05	0xXX	0x00	0x06	—	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	—		LSB	MSB

6.1.7 Change to ISO14443A protocol model

Change reader to ISO14443A model. Before executing ISO14443A command, the reader should be changed to ISO14443A model. Otherwise, the reader returns a status value 0x1F (protocol error).

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x05	0xXX	0x00	0x05	—	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	—		LSB	MSB

6.1.8 Change to 14443B procotol model

Change reader to ISO14443B model, before execute ISO14443B command, the reader should be changed to ISO14443B model. Otherwise, the reader returns a status value 0x1F (protocol error).

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x05	0xXX	0x00	0x09	—	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	—		LSB	MSB

6.1.9 Set LED

The host can set LED's action mode such as on/off duration and flash times.

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x08	0xXX	0x00	0x07	On_duration, Off_duration, flash times	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	—		LSB	MSB

6.1.10 Set Beep

The host can set Beep's action mode such as on/off duration and beeping times.

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x08	0xXX	0x00	0x08	On_duration, Off_duration, Beeping_times	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	—		LSB	MSB

6.2 ISO15693 Protocol Command

All ISO15693 tags support 3 operation modes: addressed mode, non-addressed mode and selected mode. These 3 modes correspond to the tag's various states in the field.

ISO15693 protocol commands could be carried out in all the 3 modes or 1 or 2 of them.

In addressed mode, the command should include the tag's UID.

In selected mode, one tag should be set to selected mode first and the command is aiming for this tag.

In non-addressed mode, which is also called broadcasting mode, the command is aiming for all tags in the field.

6.2.1 Inventory

The function of inventory command is to check the existence of any ISO/IEC 15693 tags in inductive field and get the UIDs of the tags.

Before running the inventory command, users can set InventoryScanTime according to their requirement. The default value of InventoryScanTime is 3s. Its value can be changed by reader-defined command "Write InventoryScanTime". The range of InventoryScanTime is 3*100ms~255*100ms. In most cases, Value 3s is a good choice. User can appropriately set the time to meet their specific requirements. Two points should be taken into consideration. One is the value could be set to a smaller value to speed up the response of inventory command, the other is the value should not be set too small to process all the tags in inductive field efficiently especially when there are a lot of tags at the same time.

Six operation modes are available for Inventory command:

MODE	STATE	DATA[]	COMMENT
Inventory without AFI	0x00	-	All Ready state tags will respond. Only one tag's UID will be returned and that tag will be turned into Quiet state.
Inventory with AFI	0x01	_AFI	All Ready state tags with the appointed AFI will respond. Only one tag's UID will be returned and that tag will be turned into Quiet state.
Inventory-scan without AFI	0x06	—	Renewed Style. The reader will run "Close RF" and "Open RF" automatically before command execution to wake up all the tags in inductive field. So all tags will respond and the reader will return all UIDs that have been decoded before InventoryScanTime overflows. The reader will turn the tags whose UIDs have been decoded into Quiet state.

	0x02	—	Consecutive Style. All Ready state tags will respond. The reader will return all UIDs that have been decoded before InventoryScanTime overflows. The reader will turn the tags whose UIDs have been decoded into Quiet state.
Inventory-scan with AFI	0x07	_AFI	Renewed Style. The reader will run "Close RF" and "Open RF" automatically before command execution to wake up all the tags in inductive field. So all tags with the appointed AFI will respond and the reader will return all UIDs that have been decoded before InventoryScanTime overflows. The reader will turn the tags whose UIDs have been decoded into Quiet state
	0x03	_AFI	Consecutive Style. All Ready state tags with the appointed AFI will respond. The reader will return all UIDs that have been decoded before InventoryScanTime overflows. The reader will turn the tags whose UIDs have been decoded into Quiet state.

6.2.1.1 Inventory without AFI

Len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC-16**	
0x05	0xXX	0x01	0x00	—	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x0d	0xXX	0x00	DSFID,UID		LSB	MSB

- ◆ Remark: The tag whose UID have been decoded by the readers will be turn into Quiet state.
- ◆ Remark: If there is at least one Ready state tag in the field and the reader could not decode any tag's UID before InventoryScanTime overflows(for example, the InventoryScanTime has been set to a too small value), the reader will return the response data block: 0x04,0xXX,0x0A,CRC-16.

6.2.1.2 Inventory with AFI

Len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC-16**	
0x06	0xXX	0x01	0x01	AFI	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x0d	0xXX	0x00	DSFID,UID		LSB	MSB

- ◆ Remark: The tag whose UID have been decoded by the readers will be turn into Quiet state.
- ◆ Remark: If there is at least one Ready state tag in the field and the reader could not decode any tag's UID before InventoryScanTime overflows(for example, the InventoryScanTime has been set to a too small value), the reader will return the response data block: 0x04,0xXX,0x0A,CRC-16.

When the reader runs inventory command in Scan Mode, the reader will try to decode all available tags in field and feedback their UIDs.

6.2.1.3 Inventory-scan without AFI

Len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC-16**	
0x05	0xXX	0x01	0x06(renewed style)	—	LSB	MSB
0x05	0xXX	0x01	0x02(consecutive style)	—	LSB	MSB

Len	Com_adr	Status	Data[]	CRC-16	
0x04+n*9	0xXX	0x00	DSFID,UID-1……DSFID,UID-n	LSB	MSB

- ◆ Remark: n equal to the number of tags being collected, and the maximum value of n is 16.
- ◆ Remark: The length is equal to 4+n*9, n refers to the number of tags being collected;
- ◆ Remark: If there is at least one Ready state tag in the field and the reader could not decode any tag's UID before InventoryScanTime overflows(for example, the InventoryScanTime has been set to a too small value), the reader will return the response data block:

  0x04, 0xXX, 0x0A, CRC-16.
- ◆ Remark: If the InventoryScanTime has been set to a small value and there are many tags in field, the reader may not be able to decode all tags before InventoryScanTime overflows. Under this circumstance, the reader will send back the following response data block:

  0x04+n*9, 0xXX, 0x0B, DSFID,UID-1……DSFID,UID-n, CRC-16

  Users can send a consecutive style inventory-scan command next to let the reader decodes the remaining tags in the field.

6.2.1.4 Inventory-scan with AFI

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x06	0xXX	0x01	0x07(renewed style)	_AFI	LSB	MSB
0x06	0xXX	0x01	0x03(consecutive style)	_AFI	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04+n*9	0xXX	0x00	DSFID,UID-1……DSFID,UID-n		LSB	MSB

- ◆ Remark: n equal to the number of tags being collected, and the maximum value of n is 16.
- ◆ Remark: The length is equal to 4+n*9, n refers to the number of tags being collected;
- ◆ Remark: If there is at least one Ready state tag in the field and the reader could not decode any tag's UID before InventoryScanTime overflows(for example, the InventoryScanTime has been set to a too small value), the reader will return the response data block:

  0x04, 0xXX, 0x0A, CRC-16.
- ◆ Remark: If the InventoryScanTime has been set to a small value and there are many tags in field, the reader may not be able to decode all tags before InventoryScanTime overflows. Under this circumstance, the reader will send back the following response data block:

  0x04+n*9, 0xXX, 0x0B, DSFID,UID-1……DSFID,UID-n, CRC-16

  Users can send a consecutive style inventory-scan command next to let the reader decodes the remaining tags in the field.

6.2.2 Stay Quiet

The host sends this command to turn the corresponding tag into Quiet state. When a tag in Quiet state, it will not reply any inventory command. But it will still reply any command in addressed mode with matching UID.

This command should run in addressed mode.

There are three ways to make the tag leave Quiet state:

- ● Tag is taken out of effective field and reentry in it.
- ● Use Select command to turn the tag into Selected mode.
- ● Use Reset to ready command to turn the tag into Ready state.

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0d	0xXX	0x02	0x00	UID	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	–		LSB	MSB

6.2.3 Read Single Block

The host sends this command to read out the corresponding tag's one block content(4 or 8 bytes) and its security status byte. The blocks number and the size of a block may differ from tag to tag with different manufacturer. Please refer to appendix 1.

This command should run in addressed mode or selected mode.

6.2.3.1 4-Byte Block

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0e	0xXX	0x20	0x00	UID,block_number	LSB	MSB
0x06	0xXX	0x20	0x01	block_number	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x09	0xXX	0x00	block_security_status, block_data(4bytes)		LSB	MSB

6.2.3.2 8-Byte Block

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0e	0xXX	0x20	0x04	UID,block_number	LSB	MSB
0x06	0xXX	0x20	0x05	block_number	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x0d	0xXX	0x00	block_security_status, block_data(8bytes)		LSB	MSB

6.2.4 Write Single Block

The host sends this command to write corresponding tag's one block content(4 or 8 bytes). The blocks number and the size of a block may differ from tag to tag with different manufacturer. Please refer to appendix 1.

If the appointed block has been locked, operation will fail and the reader will return the error_code.

There are two types of write operation feedback style corresponding to the Option_flag bit in ISO15693 defined write command's flag byte. When tag supports write operation with Option_flag=1, it is a type A tag. When tag supports write operation with Option_flag=0, it is a type B tag. Please refer to appendix 1.

This command should run in addressed mode or selected mode.

6.2.4.1 4-Byte Block

Tags of type A:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x12	0xXX	0x21	0x00	UID,block_number,data(4bytes)	LSB	MSB
0x0a	0xXX	0x21	0x01	block_number,data(4bytes)	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	–		LSB	MSB

Tags of type B:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x12	0xXX	0x21	0x08	UID,block_number,data(4bytes)	LSB	MSB
0x0a	0xXX	0x21	0x09	block_number,data(4bytes)	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	-		LSB	MSB

6.2.4.2 8-Byte Block

Tags of type A:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x16	0xXX	0x21	0x04	UID, block_number, data(8bytes)	LSB	MSB
0x0e	0xXX	0x21	0x05	block_number, data(8bytes)	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	–		LSB	MSB

Tags of type B:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x16	0xXX	0x21	0x0C	UID, block_number, data(8bytes)	LSB	MSB
0x0e	0xXX	0x21	0x0D	block_number, data(8bytes)	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	–		LSB	MSB

6.2.5 Lock Block

The host sends this command to let the reader lock the appointed block. When a block has been locked, its content should not be able to change any more.

Lock block is a write-alike command and also has two styles according the tag's A or B type. Please refer to 6.2.4 and appendix 1.

This command should run in addressed mode or selected mode.

Tags of type A:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0e	0xXX	0x22	0x00	UID,block_number	LSB	MSB
0x06	0xXX	0x22	0x01	block_number	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	–		LSB	MSB

Tags of type B:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0e	0xXX	0x22	0x08	UID, block_number	LSB	MSB
0x06	0xXX	0x22	0x09	block_number	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	–		LSB	MSB

6.2.6 Read Multiple Block

The host sends this command to let the reader read out the corresponding tag's several blocks' content (4 or 8 bytes per block) and their security status bytes. The blocks number and the size of a block may differ from tag to tag with different manufacturer. Please refer to appendix 1.

The reader can read out as many as 28 blocks one time when block size is 4 bytes and as many as 15 blocks when block size is 8 bytes.

This command should run in addressed mode or selected mode.

6.2.6.1 4-Byte Block

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0f	0xXX	0x23	0x00	_UID, _first block number, _number of blocks	LSB	MSB
0x07	0xXX	0x23	0x01	_first block number, _number of blocks	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0xXX	0xXX	0x00	block_security_status, block_data(4bytes) (Repeated times equals _number of blocks)		LSB	MSB

6.2.6.2 8-Byte Block

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0f	0xXX	0x23	0x04	_UID, _first block number, _number of blocks	LSB	MSB
0x07	0xXX	0x23	0x05	_first block number, _number of blocks	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0xXX	0xXX	0x00	block_security_status, block_data(8bytes) (Repeated times equals _number of blocks)		LSB	MSB

6.2.7 Select

The host sends this command to set the appointed tag into Selected state. All selected mode commands are aiming to the tag in Selected state. There is only one tag in Selected state at a time in one field. When turning a new tag into Selected state, the former tag in Selected state will turn into Ready state. User can also use this command to turn a tag from Quiet state into Selected state. This command can only run in Addressed mode and the command data block must contain UID of the appointed tag.

This command should run in addressed mode.

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0d	0xXX	0x25	0x00	UID	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	–		LSB	MSB

6.2.8 Reset to Ready

The host sends this command to turn the appointed tag from Quiet state into Ready state.

This command should run in addressed mode or non-address mode.

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0d	0xXX	0x26	0x00	UID	LSB	MSB
0x05	0xXX	0x26	0x01	–	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	–		LSB	MSB

This command with state=0x01 will turn all the tags in field from Quiet state into Ready state.

6.2.9 Write AFI

The host sends this command to let the reader write a new AFI value to the appointed tag.

MSB of AFI	LSB of AFI	SIGNIFICATION
0	0	Entire types and subtypes
X	0	Entire subtypes of X type
X	Y	Y subtype of X type
0	Y	Entire Y subtypes
1	0,Y	Transportation
2	0,Y	Finance
3	0,Y	Identity authentication
4	0,Y	Communication
5	0,Y	Medicine
6	0,Y	Multimedia
7	0,Y	Gambling
8	0,Y	Data storage
9	0,Y	Item management
A	0,Y	Express package
B	0,Y	Post office
C	0,Y	Airmail package
D	0,Y	
E	0,Y	
F	0,Y	

Both “X” and “Y” represents 1～F

Write AFI is a write-alike command and also has two styles according the tag's A or B type. Please refer to 6.2.4 and appendix 1.

This command should run in addressed mode or selected mode.

Tags of type A:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0e	0xXX	0x27	0x00	UID,AFI	LSB	MSB
0x06	0xXX	0x27	0x01	AFI	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	–		LSB	MSB

Tags of type B:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0e	0xXX	0x27	0x08	UID, AFI	LSB	MSB
0x06	0xXX	0x27	0x09	AFI	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	-		LSB	MSB

6.2.10 Lock AFI

The host sends this command to lock the tag's AFI value. When a tag's AFI has been locked, its value could not be able to change any more. Lock AFI is a write-alike command and also has two styles according the tag's A or B type. Please refer to 6.2.4 and appendix 1.

This command should run in addressed mode or selected mode.

Tags of type A:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0d	0xXX	0x28	0x00	UID	LSB	MSB
0x05	0xXX	0x28	0x01	-	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	-		LSB	MSB

Tags of type B:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0d	0xXX	0x28	0x08	UID	LSB	MSB
0x05	0xXX	0x28	0x09	-	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	-		LSB	MSB

6.2.11 Write DSFID

The host sends this command to write a new DSFID value to the appointed tag.

Write DSFID is a write-alike command and also has two styles according the tag's A or B type. Please refer to 6.2.4 and appendix 1.

This command should run in addressed mode or selected mode.

Tags of type A:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0e	0xXX	0x29	0x00	UID,DSFID	LSB	MSB
0x06	0xXX	0x29	0x01	DSFID	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	-		LSB	MSB

Tags of type B:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0e	0xXX	0x29	0x08	UID,DSFID	LSB	MSB
0x06	0xXX	0x29	0x09	DSFID	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC-16**	
0x04	0xXX	0x00	-		LSB	MSB

6.2.12 Lock DSFID

The host sends this command to lock the tag's DSFID value. When a tag's DSFID has been locked, its value should not be able to change any more.

Lock DSFID is a write-alike command and also has two styles according the tag's A or B type. Please refer to 6.2.4 and appendix 1.

This command should run in addressed mode or selected mode.

Tags of type A:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0d	0xXX	0x2A	0x00	UID	LSB	MSB
0x05	0xXX	0x2A	0x01	-	LSB	MSB

Len	Com_adr	Status	Data[]	CRC-16	
0x04	0xXX	0x00	–	LSB	MSB

Tags of type B:

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0d	0xXX	0x2A	0x08	UID	LSB	MSB
0x05	0xXX	0x2A	0x09	-	LSB	MSB

Len	Com_adr	Status	Data[]	CRC-16	
0x04	0xXX	0x00	–	LSB	MSB

6.2.13 Get System Information

The host sends this command to get the detailed information of the appointed tag's. The information include 1 byte information_flag, 8 byte UID, 1 byte DSFID, 1 byte AFI, 2 bytes Memorey_size and 1 byte IC_reference. Each bit's meaning of the information_flag is as follows:

BIT	NAME	STATUS	COMMENT
Bit1	DSFID	0	DSFID nonexistence or nonsupport
		1	DSFID existence or support
Bit2	AFI	0	AFI nonexistence or nonsupport
		1	AFI existence or support
Bit3	Memory_size	0	Memory_size nonexistence or nonsupport
		1	Memory_size existence or support
Bit4	IC_Reference	0	IC_Reference nonexistence or nonsupport
		1	IC_Reference existence or support
Bit5	RFU	-	Set to 0
Bit6	RFU	-	Set to 0
Bit7	RFU	-	Set to 0
Bit8	RFU	-	Set to 0

This command should run in addressed mode or selected mode.

Len	Com_adr	Cmd	State	Data[]	CRC-16	
0x0d	0xXX	0x2B	0x00	UID	LSB	MSB
0x05	0xXX	0x2B	0x01	—	LSB	MSB

Len	Com_adr	Status	Data[]	CRC-16	
0x12	0xXX	0x00	Information_flag,UID,DSFID,AFI, Memory_size(2bytes),IC_reference	LSB	MSB

Remark: The above chart only describe the situation when “Information flag”=0x0f. If Information_flag is other value, the length and contents of the response data block will alter according to Information_flag's definition.

6.3 ISO14443A Protocol Command

6.3.1 Request

Send request command to detect if any ISO14443A card exists in the field.

Len	Com_adr	Cmd	State	Data[]	CRC—16	
0x06	0xXX	0x41	0x10	Mode	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x06	0xXX	0x00	tagType[] (2Byte)		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Mode=0,request all cards in area except halted status;

Mode=1, request all cards in area;

tagType[]: type of the card with Least Significant Byte first.

6.3.2 Anticoll

Process ISO 14443A anti-collision operation after request command, return one tag's UID.

Len	Com_adr	Cmd	State	Data[]	CRC—16	
0x06	0Xxx	0x42	0x10	Reserved（0x00）	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x08	0Xxx	0x00	Data[] (4Byte)		LSB	MSB
0x05	0Xxx	0x10	Error code		LSB	MSB

Reserved = 0x00;

Data[]: Card's UID number, least significant byte first.

6.3.3 Anticoll2

Process ISO 14443A anti-collision operation after request command

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x07	0xXX	0x71	0x10	Encoll,Reserved （0x00）(2Bytes)	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x08	0xXX	0x00	Data[] (4Bytes)		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Encoll = 1: enable multiple tags in the inductive area.

Encoll = 0:disable multiple tags in the inductive area. If more than one tag in the area, the reader will return a error code;

Reserved = 0x00;

Data[]: Card's UID with Least Significant Byte first.

6.3.4 ULAnticoll

Process UltraLight anti-collision operation after request command

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x06	0xXX	0x7A	0x10	Reserved	LSB	MSB
len	Com_adr	Status	Data[]		CRC—16	
0x0B	0xXX	0x00	SN[] (7Bytes)		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Reserved = 0x00;

SN[]: 7 bytes tag's serial number with Least Significant Byte first

6.3.5 Select

Select one card with its UID and return its capacity

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x09	0xXX	0x43	0x10	UID[] (4Bytes)	LSB	MSB
len	Com_adr	Status	Data[]		CRC—16	
0x05	0xXX	0x00	Size (1Byte)		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

UID[]:Card's UID number with Least Significant Byte first

Size: tag's capacity.

6.3.6 Anthentication

Authenticate with KEY in the reader's EEPROM. The card should be authenticated before read, write and other data sector operation.

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x07	0xXX	0x44	0x10	Mode,Addr	LSB	MSB
len	Com_adr	Status	Data[]		CRC—16	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Mode =0, authenticate with KEY A

Mode =1, authenticate with KEY B

Addr: The number of the sector (0~15) to be authenticated with the KEY from the corresponding area in reader's EEPROM

6.3.7 Anthentication2

Cross-authenticate with KEY in reader's EEPROM. The card should be authenticated before read, write and other data sector operation

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x08	0xXX	0x72	0x10	Mode, SecNr, KeyNr	LSB	MSB
len	Com_adr	Status	Data[]		CRC—16	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Mode =0, authenticate with KEY A;

Mode =1, authenticate with KEY B;

SecNr: The number of the sector to be authenticated;
KeyNr: The number of the sector in EEPROM storing the KEY.

6.3.8 AuthKey

Directly authenticate with dictated KEY. The card should be authenticated before read, write and other data sector operation

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x0D	0xXX	0x73	0x10	Mode,Addr,Key[]	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Mode =0, authenticate with KEY A
Mode =1, authenticate with KEY B
Addr: The number of the sector to be authenticated;
Key[]; 6 bytes KEY with Least Significant Byte first

6.3.9 Halt

Set the active card in halt status.

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x05	0xXX	0x45	0x10	—	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

6.3.10 Read

Read out 1 block data from the card.

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x06	0xXX	0x46	0x10	Addr (1Byte)	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x14	0xXX	0x00	Data[] (16Byte)		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Addr: Absolute block address;
Data[]: 16 bytes block data with Least Significant Byte first

6.3.11 Write

Write 1 block data in the card

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x16	0xXX	0x47	0x10	Addr,Data[]	LSB	MSB

len	Com_adr	Status	Data[]	CRC—16	
0x04	0xXX	0x00	—	LSB	MSB
0x05	0xXX	0x10	Error code	LSB	MSB

Addr: Absolute block address;

Data[]: The 16 bytes data to be written with Least Significant Byte first

6.3.12 ULWrite

Write 4 bytes data in the UltraLight card

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x0A	0xXX	0x7B	0x10	Addr,Data[] (5Bytes)	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00			LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Addr: Absolute block address;

Data[]: The 4 bytes data to be written with Least Significant Byte first

6.3.13 InitValue

Initialize the block to value-block with designated value.

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x0A	0xXX	0x78	0x10	Addr,value[]	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Addr: block address to initialize;

value : 4 bytes initializing value with Least Significant Byte first

6.3.14 ReadValue

Read out the content of value-block.

Len	Com_adr	Cmd	State	Data[]	CRC—16	
0x06	0xXX	0x79	0x10	Addr (1Byte)	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x08	0xXX	0x00	Data[] (4Bytes)		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Addr: block address to read,the block must a value-block.

Data[]: 4 bytes value data with Least Significant Byte first.

6.3.15 Increment

Read out the content of a value-block, check its structure and add it with the value assigned, then store the result in card's internal register. The value-block should have a specific format that is defined in Mifare series cards' datasheet

len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC—16**	
0x0A	0xXX	0x48	0x10	Addr,Data[]	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Addr: Absolute block address;

Data[]:4 bytes value to be added with Least Significant Byte first

6.3.16 Decrement

Read out the content of a value-block, check its structure and subtract it with the value assigned, then store the result in card's internal register. The value-block should have a specific format that is defined in Mifare series cards' datasheet.

len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC—16**	
0x0A	0xXX	0x49	0x10	Addr,Data[]	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Addr: Absolute block address;

Data[]:4 bytes value to be subtracted with Least Significant Byte first

6.3.17 Restore

Read out the content of a value-block, check its structure and store it in card's internal register. The value-block should have a specific format which is defined in Mifare series cards' datasheet.

len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC—16**	
0x06	0xXX	0x4A	0x10	Addr	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Addr: Absolute block address.

6.3.18 Transfer

Transfer the content of card's internal register to a valid value-block. The value-block should have a specific format which is defined in Mifare series cards' datasheet. This operation could only be used after increment, decrement or restore process

Len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC—16**	
0x06	0xXX	0x4B	0x10	Addr	LSB	MSB

Len	Com_adr	Status	Data[]	CRC—16	
0x04	0xXX	0x00	—	LSB	MSB
0x05	0xXX	0x10	Error code	LSB	MSB

Addr: Absolute block address.

6.3.19 LoadKey

Save a KEY in the reader's EEPROM.

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x0D	0xXX	0x4C	0x10	Mode, Secnr,Key[]	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Mode =0, KEY A;

Mode =1, KEY B;

Addr: Sector number in EEPROM and should be less than 16;

Key[];6 bytes KEY with least significant byte first

6.3.20 CheckWrite

Compare the data written in the card with known data. In this operation, RR3036 use the KEY in its EEPROM with the same sector number of the card data block to be checked.

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x1B	0xXX	0x53	0x10	SN[],Authmode,Addr,Data[]	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

SN[]:4 bytes Card's UID number, least significant byte first;

Authmode: Authentication mode;

Addr: Absolute block address;

Data[]:16 data bytes for checking with Least Significant Byte first

6.3.21 ReadE2

Read out the content of the reader's EEPROM. The address should be less than 0x80. The content of the EEPROM with the address over 0x80 is used for KEY storage and could not be read out.

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x07	0xXX	0x61	0x10	Addr,Length	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04+n	0xXX	0x00	Data[]		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Addr: Begin address of the EEPROM to be read out (less than 0x80).
Length: Byte length of the data to be read out (less than 20 bytes).
Data[]:Data being read out.

6.3.22 WriteE2

Write data in the reader's EEPROM. Data in address 0x00~0x0f are read-only product information. Data in address 0x10~0x2f are initialization data and should not be altered. Address area 0x80~0x1ff are used for KEY storage and could be written with LoadKey command

len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC—16**	
0x05+n	0xXX	0x62	0x10	Addr,Length,Data[]	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Addr: Begin address of the EEPROM to be written (0x30-0x7e);
Length: Byte length of the data to be written (less than 20 bytes);
Data[]:Data to be written.

6.3.23 Value

Perform value-block related operation between one value block and one transfer block. The two block should both have value-block format and in the same sector. The value-block related operation includes increment, decrement and restore. Auto transfer is supported.

len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC—16**	
0x0C	0xXX	0x70	0x10	Mode,Addr,value[],Trans_Adr	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00	—		LSB	MSB
0x05	0xXX	0x10	Error code		LSB	MSB

Mode=0xc0, decrement ;
Mode=0xc1, increment ;
Mode=0xc2, restore ;
Addr : Absolute block address;
value[]:Value when in decrement or increment. There is not this item in restore operation;
Trans_Adr: Absolute block address of the transfer block

6.4 ISO14443B Protocol Command

6.4.1 Request

6.4.1.1 Request without AFI

Detect that if any ISO/IEC 14443B card is in the inductive area and return the card's PUPI, application data and protocol information if a card responds to the request.

len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC—16**	
0x06	0xXX	0x81	0x1B	Mode（1byte）	LSB	MSB

len	Com_adr	Status	Data[]	CRC—16	
0x0f	0xXX	0x00	PUPI[](4Byte)、 Application Data[](4bytes)、 Protocol Information[](3bytes)	LSB	MSB
0x05	0xXX	0x1B	Error code	LSB	MSB

Mode=0,request all cards in area except halted status;

Mode=1, request all cards in area;

PUPI[]:4 bytes card's Pseudo Unique PICC Identifier with Least Significant Byte first;

Application Data[]:4 bytes card's application data with Least Significant Byte first;

Protocol Information[]:3 bytes card's protocol information with Least Significant Byte first.

6.4.1.2 Request with AFI

All tags with the appointed AFI will respond. the tag's PUPI will be returned.

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x07	0xXX	0x82	0x1B	Mode(1byte),AFI(1byte)	LSB	MSB
len	Com_adr	Status	Data[]		CRC—16	
0x0f	0xXX	0x00	PUPI(4Byte)、 Application Data(4bytes)、 Protocol Information (3bytes)		LSB	MSB
0x05	0xXX	0x1B	Error code		LSB	MSB

Mode=0,request all cards in area except halted status;

Mode=1, request all cards in area;

AFI: Application Family Identifier; if AFI = 0, all tags will respond.

PUPI[]:4 bytes card's Pseudo Unique PICC Identifier with Least Significant Byte first;

Application Data[]:4 bytes card's application data with Least Significant Byte first;

Protocol Information[]:3 bytes card's protocol information with Least Significant Byte first.

6.4.2 Anticoll

6.4.2.1 Anticoll without AFI

All tags will respond and one tag's PUPI will be returned.

len	Com_adr	Cmd	State	Data[]	CRC—16	
0x06	0xXX	0x83	0x1B	Mode（1byte）	LSB	MSB
len	Com_adr	Status	Data[]		CRC—16	
0xXX	0xXX	0x00	PUPI(4Byte)、 ApplicationData-1(4bytes)、 ProtocolInfo-1(3bytes)		LSB	MSB
0x05	0xXX	0x1B	Error code		LSB	MSB

Mode=0,request all cards in area except halted status;

Mode=1, request all cards in area;

PUPI[]:4 bytes card's Pseudo Unique PICC Identifier with Least Significant Byte first;

Application Data[]:4 bytes card's application data with Least Significant Byte first;

Protocol Information[]:3 bytes card's protocol information with Least Significant Byte first.

6.4.2.2 Anticoll with AFI

All tags with the appointed AFI will respond and one tag's PUPI will be returned.

len	Com_adr	Cmd	State	Data[]		CRC—16	
0x07	0xXX	0x84	0x1B	Mode(1byte）	AFI(1byte)	LSB	MSB
len	**Com_adr**	**Status**	**Data[]**			**CRC—16**	
0xXX	0xXX	0x00	PUPI(4Byte)、 ApplicationData(4bytes)、 ProtocolInfo(3bytes)			LSB	MSB
0x05	0xXX	0x1B	Error code			LSB	MSB

Mode=0,request all cards in area except halted status;

Mode=1, request all cards in area;

PUPI[]:4 bytes card's Pseudo Unique PICC Identifier with Least Significant Byte first;

Application Data[]:4 bytes card's application data with Least Significant Byte first;

Protocol Information[]:3 bytes card's protocol information with Least Significant Byte first.

6.4.3 SelectAndSet

Assign a channel number (CID) for a specified card in the inductive area by its PUPI. The card will be in active state if the operation succeeds.

<table>
<tr><th>len</th><th>Com_adr</th><th>Cmd</th><th>State</th><th colspan="4">Data[]</th></tr>
<tr><td>0x0E</td><td>0xXX</td><td>0x85</td><td>0x1B</td><td>PUPI(4Bytes)</td><td>CID(1Byte)</td><td colspan="2">Param1（1Byte）</td></tr>
<tr><th colspan="6">Data[]</th><th colspan="2">CRC—16</th></tr>
<tr><td colspan="2">Param2（1Byte）</td><td colspan="2">Param3（1Byte）</td><td colspan="2">Param4（1Byte）</td><td>LSB</td><td>MSB</td></tr>
<tr><th>len</th><th>Com_adr</th><th>Status</th><th colspan="3">Data[]</th><th colspan="2">CRC—16</th></tr>
<tr><td>0x05</td><td>0xXX</td><td>0x00</td><td colspan="3">Content(1Byte)</td><td>LSB</td><td>MSB</td></tr>
<tr><td>0x05</td><td>0xXX</td><td>0x1B</td><td colspan="3">Error code</td><td>LSB</td><td>MSB</td></tr>
</table>

PUPI: 4 bytes card's Pseudo Unique PICC Identifier with Least Significant Byte first.

CID: channel number assigned for card by reader(0~14);

param1: whether break EOF or SOF.

param1 = 0: don't break EOF and SOF.

param1 = 1: don't break EOF, break SOF

param1 = 2: don't break SOF, break EOF

param1 = 3: break EOF and SOF.

param2: low 4bits indicate minimum delay before cards responding after the end of a command sent by reader(TR0); high 4bits indicate the minimum delay between subcarrier modulation start and beginning of data transmission(TR1).

low 4bits: 0000. TR0 = 64/fs;
0001. TR0 = 48/fs;
0010. TR0 = 16/fs.

high 4bits: 0000. TR1 = 80/fs;
0001. TR1 = 64/fs;
0010. TR1 = 16/fs.

param3: define maximum frame size that can be received by reader.

0: RFU;	1:RFU;	2:32bytes;	3:40bytes;
4:48bytes;	5:64bytes;	6:96bytes;	7:128bytes;
8:256bytes;	9~255:RFU		

param4: Bit rate selection

low 4bits: PCD to PICC:

0000:PCD to PICC，letu=128/fc，Bit rate is 106kbit/s；
0001:PCD to PICC，letu=64/fc， Bit rate is 212kbit/s；
0010:PCD to PICC，letu=32/fc，Bit rate is 424kbit/s；
0011:PCD to PICC，letu=16/fc，Bit rate is 847kbit/s；RFU for other。

high 4bits: PICC to PCD:

0000:PICC to PCD，letu=128/fc，Bit rate is 106kbit/s；
0001:PICC to PCD，letu=64/fc，Bit rate is 212kbit/s；
0010:PICC to PCD，letu=32/fc，Bit rate is 424kbit/s；
0011:PICC to PCD，letu=16/fc，Bit rate is 847kbit/s；RFU for other.

Content: low 4bits of Content echoes back the CID value assigned by this command and the upper 4bits of the Content echoes back the Maximum Buffer Length Index (MBLI), which communicates to the PCD how many bytes the PICC is capable of receiving as a chained frame. If the PICC does not support chained frames, then this parameter is 0

6.4.4 Halt

Set the current active ISO/IEC 14443B card into halt status

Len	**Com_adr**	**Cmd**	**State**	**Data[]**	**CRC—16**	
0x09	0xXX	0x86	0x1B	PUPI（4bytes）	LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**		**CRC—16**	
0x04	0xXX	0x00	——		LSB	MSB
0x05	0xXX	0x1B	Error code		LSB	MSB

PUPI: 4 bytes card's Pseudo Unique PICC Identifier with Least Significant Bytes first.

6.4.5 TransparentCmd

Turn the reader into a RF front-end engine to access tags. The reader encapsulates the host's data in ISO14443B protocol format and sends them to tags and decodes the feedback data from tag.

Len	**Com_adr**	**Cmd**	**State**	**Data[]**	
0xXX	0xXX	0x89	0x1B	Time_M(1byte)	Time_N(1byte)
Data[]				**CRC—16**	
CMD_Length (1byte)	RSP_Length (1byte)	CMD_Data (1byte~63bytes)		LSB	MSB
Len	**Com_adr**	**Status**	**Data[]**	**CRC—16**	
0x05	0xXX	0x00	RSP_Data(0byte~100bytes)	LSB	MSB
0x05	0xXX	0x1B	Error code	LSB	MSB

Time_M, Time_N: Set the time interval before receiving tag respone data after command transmission. The time interval can be calculated by:

$$Time = Time\_M * \frac{2^{Time\_N}}{13.56MHz} us$$

Time_M: 0~255;

Time_N: 0~21;

CMD_Length: The byte length of the data the reader should send to the tag ranging from 1 to 63.

RSP_Length: The response data byte length expected from the tag which ranges from 0 to 100.

CMD_Data: the data should be sent to the tag by reader

RSP_Data: the data responded from the tag.

7. List of reader-defined command

Command			**Description**
cmd name	**Cmd value**	**State**	
Get Reader Information	0x00	0x00	get the reader's information including reader's address, firmware version, reader type, supported protocol and InventoryScanTime value.
Close RF	0x00	0x01	turn off the RF output of the reader and close the inductive field.
Open RF	0x00	0x02	turn on the RF output of the reader and open the inductive field.
Write Com_adr	0x00	0x03	change reader's address.
Write InventoryScanTime	0x00	0x04	change reader's InventoryScanTime
ChangToISO14443A	0x00	0x05	Change reader's model to iso14443A model.
ChangToISO15693	0x00	0x06	Change reader's model to iso15693 model.
SetLED	0x00	0x07	control of the output of LED.
SetBeep	0x00	0x08	control of the output of Beep.
ChangToISO14443B	0x00	0x09	Change reader's model to iso14443B model.

8. List of ISO15693 command

Command		Description
cmd name	cmd value	
Inventory	0x01	Check the existence of ISO 15693 compatible tags.
Stay Quiet	0x02	Turn the corresponding tag into Quiet state
Read Single Block	0x20	Read out the corresponding tag's one block content(4 or 8 bytes) and its security status byte
Write Single Block	0x21	Write one block content (4 or 8 bytes) to corresponding tag
Lock Block	0x22	Lock the appointed block
Read Multiple Block	0x23	Read out the corresponding tag's several blocks' content (4 or 8 bytes per block) and their security status bytes
Select	0x25	Turn the appointed tag into Selected state
Reset to Ready	0x26	Turn the appointed tag from Quiet state into Ready state
Write AFI	0x27	Write a new AFI value to the appointed tag
Lock AFI	0x28	Lock the tag's AFI value
Write DSFID	0x29	Write a new DSFID value to the appointed tag
Lock DSFID	0x2A	Lock the tag's DSFID value.
Get System Information	0x2B	Get the detailed information of the appointed tag

9. List of ISO14443A command

Command		Description
cmd name	**cmd value**	
Request	0x41	Check if any ISO14443A card exist in inductive area.
Anticoll	0x42	Process ISO14443A anti-collision operation after request command, return one tag's UID.
Anticoll2	0x71	Get one tag's UID with Enabling/Disabling multiple tags in the inductive area .
ULAnticoll	0x7A	Process Ultralight tag anti-collision and return one tag's UID.
Select	0x43	Select a card with known UID and return its capacity
Authentication	0x44	Authenticate with KEY in the reader's EEPROM
Authentication2	0x72	Cross-authenticate with KEY in reader's EEPROM
AuthKey	0x73	Directly authenticate with dictated KEY.
Halt	0x45	Set the active card in halt status
Read	0x46	Read out 1 block data (16 bytes) from the card.
Write	0x47	Write 1 block data (16 bytes) in the card.
ULWrite	0x7B	Write 4 bytes data in the UltraLight card
Initvalue	0x78	Initialize a block to value-block with designated value
Readvalue	0x79	Read out the content of value-block
Increment	0x48	Read out the content of a value-block, add it with the value assigned and then store the result in card's internal register.
Decrement	0x49	Read out the content of a value-block, subtract it with the value assigned and then store the result in card's internal register
Restore	0x4A	Read out the content of a value-block, check its structure and store it in card's internal register
Transfer	0x4B	Transfer the content of card's internal register to a valid value-block.
LoadKey	0x4C	Save a KEY in the reader's EEPROM.
CheckWrite	0x53	Compare the data written in the card with known data.
ReadE2	0x61	Read out the content of the reader's EEPROM
WriteE2	0x62	Write data in the reader's EEPROM.
Value	0x70	Perform value-block related operation between one value block and one transfer block

ISO14443 typeA state diagram：

REQA：REQA Command；

WUP：WAKE-UP Command；

AC：ANTICOLLISION Command with matched UID；

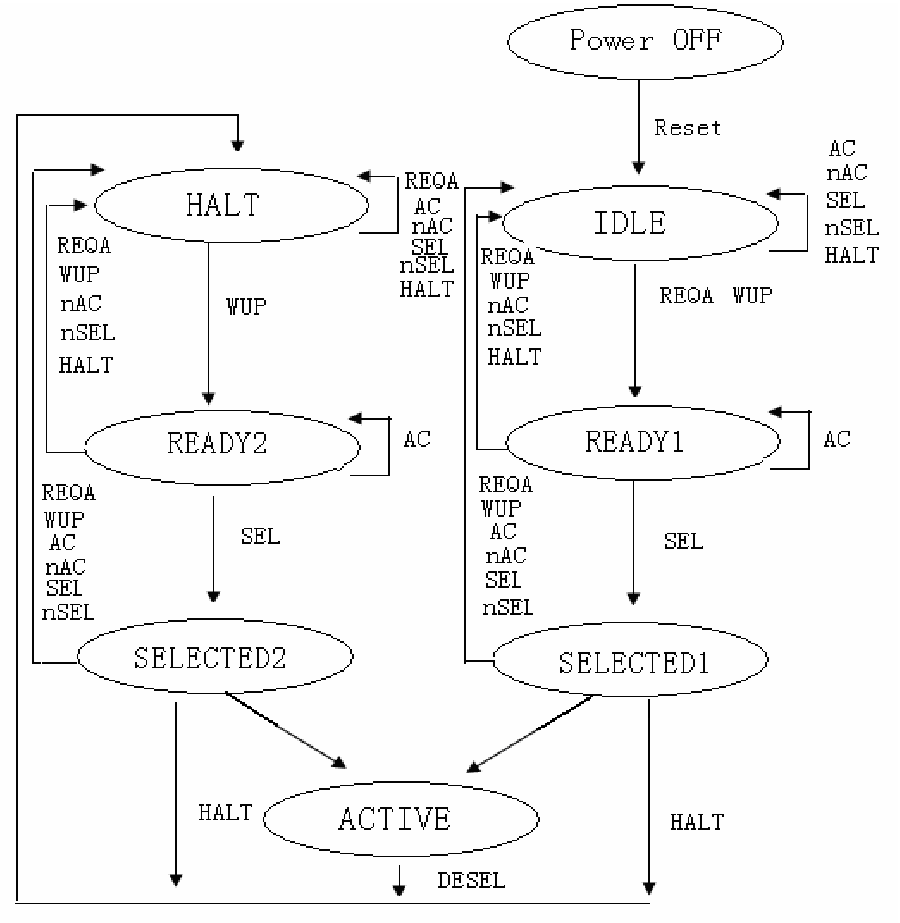
nAC：ANTICOLLISION Command with not-matched UID;

SEL：SELECT Command with matched UID；

nSEL：SELECT Command with not-matched UID

HALT ：HALT Command；

DESEL ：DESELECT Command defined in ISO/IEC14443-4。



10. List of ISO14443B command

Command		Description
cmd name	cmd value	
request without AFI	0x81	Get a tag's PUPI, application data, protocol information
request witht AFI	0x82	Get a tag's information with the appointed AFI
anticoll without AFI	0x83	Process anti-collision and get one tag's PUPI.
anticoll with AFI	0x84	Process anti-collision and get one tag's information with the appointed AFI.
selectandset	0x85	Select a tag and assign a CID for it.
halt	0x86	Set the current active card into halt status
transparentcmd	0x89	Send user-defined command to the tag by using the reader as a RF front-end engine

Appendix 1

Mainstream ISO15693 HF Tag

MANUFACTURER	Manu. CODE	BLOCK INFORMATION		TYPE	
		BLOCK NUMBER	BLOCK SIZE	TYPE A	TYPE B
Infineon (ISO Address mode)	0x05	256 (user available range:0~249)	4 bytes		√
		64 (user available range:0~57)	4 bytes		√
STMicroelectronics (LRI512)	0x02	16 (user available range:0~15)	4 bytes		√
Fujitsu (MB89R116)	0x08	256 (user available range:0~249)	8 bytes	√	√
Philips (I-Code SLI)	0x04	32 (user available range:0~27)	4 bytes		√
Texas Instruments (Tag-it HF-I)	0x07	64 (user available range:0～63)	4 bytes	√	
EM Microelectronics	0x16	50 (user available range:13～48)	8 bytes	√	
		14 (user available range:3～11)	4 bytes	√	

- ◆ As to tags not included in the appendix, please refer to their datasheets.