GENERAL DESCRIPTION

RHOMBUSMF6201ST is a high performance ISO14443 TYPE A format IC card R/W engine module, which is fit for standard RF Card R/W (can be directly connected to RS232 interface of PC after level is converted), handset, POS toll collector, access control, other toll system, and smart card system.

FEATURES

- Support external balance drive transceiver antenna;
- Maximum effective distance up to 100mm;
- Low power dissipation with single power supply;
- Support TTL level serial port;
- Support r/w operation of Mifare I and its compatible card;
- Very small outline, PIN descriptions compatible with DIP32 socket.

CHARACTERISTICS

Absolute Maximum Ratings

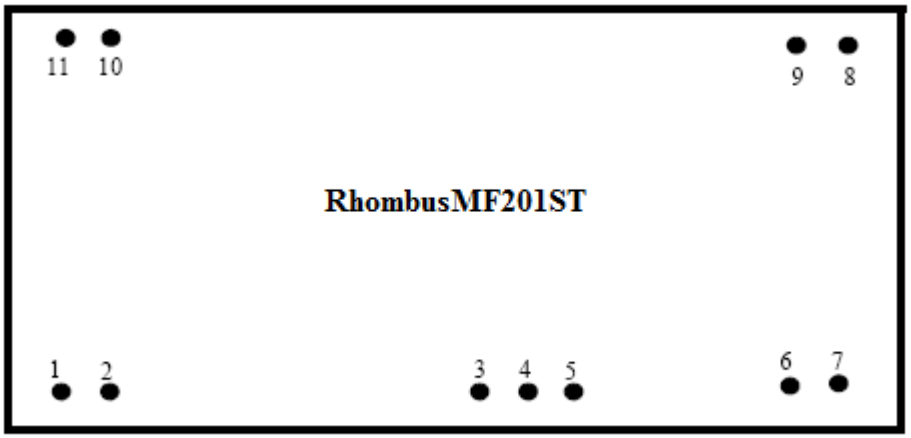
ITEM	SYMBOL	VALUE	UNIT
Power Supply	VCC	6	V
Operating Temp.	T_{OPR}	0~+70	℃
Storage Temp.	T_{STR}	-55~+125	℃

- Electrical and Mechanical Specification
  Under T_A=25℃，VCC= +5V unless specified

ITEM	SYMBOL	MIN	TYP	MAX	UNIT
Power Supply	VCC	4.5	5	5.5	V
Current Supply	I_C（work） I_S（standby）		75 7	120	MA MA
Operation Freq.	F_{REQ}		13.56		MHz
Effective Distance *	DIS	0	70	100	mm

* Effective Distance depends on tags and operating environment.

INTERFACE DESCRIPTION



Bottom View

NUMBER	SYMBOL	DESCRIPTION
1	RXD	Serial data input
2	TXD	Serial data output
3	RST	High effective reset *
4	BEEP	Buzzer drive output (normal interior pull up to high level)
5	CTL	Control bit drive output (normal interior pull up to high level)
6	AGND	Antenna, ground wire
7	RX	Receiving signal input, directly received by antenna loop
8	TX1	Antenna output drive 1
9	TX2	Antenna output drive 2
10	GND	GND
11	VCC	+5V power supply

*RST need not be connected generally, but hung in the air. Pulling up the level can reset the circuit inside the module.

APPLICATION DATA

I Card data

Mifare I and its compatible card follow ISO14443-2&3 TYPE A criterion.

1. Performance Index
   - Accord with ISO14443-2 TYPE A RF interface criterion.
   - Accord with ISO14443-3 TYPE A data frame format criterion
   - Capacity: EEPROM of 8Kbit
   - Divided as 16 sectors, 4 blocks per sector, 16 bytes per block, block being

the access unit.

- A group of independent codes and accessing control per sector
- Exclusive 32-bit serial number per card
- Possess anti-collision mechanism, support multi-card operation
- No power supply, self-antenna, there are encrypted control logic and communication logic circuit inside.
- Date could be saved for 10 years, could be rewritten 100,000 times, could be read permanently
- Operating freq.: 13.56MHz
- Communication speed: 106kbps

2. Memory Structure

1) Mifare I card is divided to 16 sectors, every sector consists of 4 blocks (block 0,block 1,block 2,block3).The 64 blocks of the 16 sectors is numbered as 0~63 according to the absolute address. The memory Structure is as follows:

Sector 0	Block 0		Data Block 0
	Block 1		Data Block 1
	Block 2		Data Block 2
	Block 3	Code A access control code B	Data Block 3
Sector 1	Block 0		Data Block 4
	Block 1		Data Block 5
	Block 2		Data Block 6
	Block 3	Code A access control code B	Data Block 7
. . .	. . .	. . .	. . .
Sector 15	Block 0		Data Block 60
	Block 1		Data Block 61
	Block 2		Data Block 62
	Block 3	Code A access control code B	Data Block 63

ii) Sector 0 block 0(namely absolute address block 0) is for placing the factory code, which is fixed and can not be changed.

iii) Block 0, block 1, block 2 of every sector are the data block, for storing data. Data block could be used in two ways: first, it could be saved as common data, can be read and written; second, it could be used as data value block, for operation such as initialization value, increment, decrement and read.

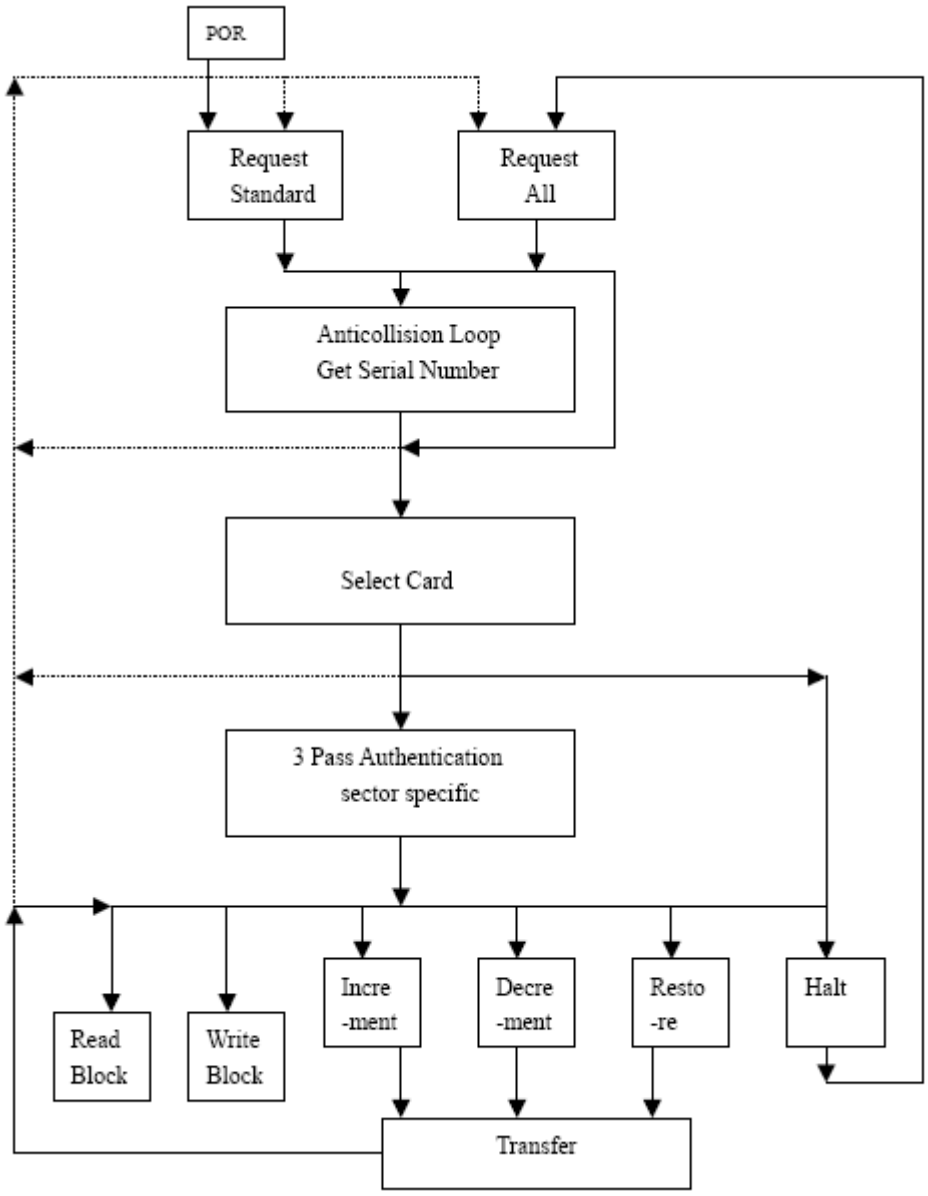
iv) Block 3 in every sector is the control block; it consists of code A,

access control, and code B. The structure is as follows:

A0 A1 A2 A3 A4 A5	FF 07 80 69	B0 B1 B2 B3 B4 B5
Code A (6-byte)	access control(4-byte)	code B(6-byte)

v) The code and access control in every sector are independent. Sectors could set its respective code and access control.

vi) The card typical operating flow chart is as follows:



II. **Serial port protocol**

1. Serial port Specification

Data communication could be realized through serial port between RHOMBUSMF6201ST and PC (SCM, MPU, and Controller). Corresponding operation is executed according to the command and requirements of PC. There is 1

start bit, 8 data bits, 1stop bit in the data frame of serial port, but there is no parity bit. Baud rate: 9600. In serial communication, the least significant byte is transferred first, and the least significant bit of each byte is transferred first.

2. Control Character

Four kinds of control characters are defined to describe the start, the end, responsion, and no-responsion of the serial communication.

Definition	Symbol	Value
Start Symbol	STX	0x02
End Symbol	ETX	0x03
Responsion	ACK	0x06
No Responsion	NAK	0x15

3. Protocol Description

The communication process: First, PC send command and data to RHOMBUSMF6201ST R/W engine, then RHOMBUSMF6201ST send the execution result state of the command and data back to PC. The communication two parties need a handshake course to start a communication, and then data are transferred.

The sending process of the PC command is as follows:

PC	Data Transfer Direction	RHOMBUSMF6201ST	Specification
STX	→		If PC does not receive ACK or NAK in 20ms, STX could be re-sent. After PC receives
	←	ACK	

command Data Block + ETX	→		ACK, command data block starts to be sent in 45ms. Every two adjacent byte should be sent less than 15ms one after another. In the process of sending command data block, if PC receives any message that RHOMBUSMF6201ST send, it all indicates lost synchronization of the synchronous communication between PC and RHOMBUSMF6201ST. Then PC could stop sending command, waiting 15ms to re-start handshake process.

First, after PC send STX and wait for ACK that is sent back by RHOMBUSMF6201ST to start up transferring data block, if ACK or NAK is not received within 20ms, PC could choose to re-send STX or stop communication, and carry out disposal according to communication failure. After PC receives the ACK response of RHOMBUSMF6201ST, it will send command data block including operating command, operating data to RHOMBUSMF6201ST.lastly, PC will send ETX character to end the transmission, and wait for the sent- back command execution result.

RHOMBUSMF6201ST will finish command execution and send back the result in 300ms after it receives the command of PC. The following form is the returning process of the command execution result.

RHOMBUSMF6201ST	Data Transfer Direction	PC	SPECIFICATION
STX	→		RHOMBUSMF6201ST will end transmission if it does not receive ACK in 45ms. RHOMBUSMF6201ST starts to respond transmission of data block in 45ms after it receives ACK.
	←	ACK	
Response Data Block + ETX	→		

After RHOMBUSMF6201ST sends STX and waits for ACK that is sent back by PC to start up transferring data block, if ACK is not received within 45ms, RHOMBUSMF6201ST could stop this communication. After RHOMBUSMF6201ST

receives the ACK response of PC, it will send response data block including command execution state, execution result data to PC. Lastly, RHOMBUSMF6201ST will send ETX character to end the transmission.

4. The format of data block

A. command data block

SeqNo	*Cmd*	*Len*	*Data[]*	*BCC*

SeqNo：Data package of 1 byte exchange serial number, range: 0～255. After one correct data exchange, when PC sends the next command, add 1 to the serial number. RHOMBUSMF6201ST will return the most newly-received data package number in the response data block. The PC program can utilize this number to strengthen the integrality of the communication process, and can also neglect the disposal to this number to simplify the communication process.

Cmd: Operating command notation of 1 byte, there are 25 commands defined in all.

Len: the length of command operating data of 1 byte, range:0~22, 0 denotes non operating data.

Data[]: command operating data of Len bytes, if Len=0,it indicates there is no data.

BCC: Check sum of 1 byte is the XOR byte by byte of all data in the data block (except BCC).

BCC=SeqNo ⊕ *Cmd* ⊕ Len ⊕ Data[]

B. Response data block

SeqNo	*Status*	*Len*	*Data[] BCC*

SeqNo： Most recent-received data package exchange serial number of 1 byte.

Status： Command execution result state value of 1 byte, o denotes successful command execution. Other values all denote error code.

Len： The length of the response data of 1 byte.range:0~16. If error occurs in the command execution, the value is 0.

Data[]：Response data of Len bytes. If Len=o, it means no data.

BCC：Check sum of 1 byte is the XOR byte by byte of all data in the data block(except BCC).

BCC=SeqNo ⊕ *Cmd* ⊕ Len ⊕ Data[]

5. The collection of operating command

COMMAND		PARAMETER		SPECIFICATION
NAME	VALUE	SEND	RECEIVE	

Request	0x41	_Mode	_Tag Type	Send enquiry command to check if there is card in effective area.
Anticoll	0x42	_Reserved	_Snr	Start anti-collision check and return serial number of a card
Anticoll2	0x71	_Encoll ，Reserved	_Snr	Permit or forbid the entry of multi-card, return a card number
Select	0x43	_Snr	_Size	Select card, return the card capacity
Authentication	0x44	_Mode，Secnr	--	Start authentication operation
Authentication2	0x72	_Mode ，_Secnr ，_Keynr	--	Select key section authentication
AuthKey	0x73	_Mode ，_Secnr，_Key,	--	Direct code authentication
Halt	0x45	--	--	Set the card as hanging mode
Read	0x46	_Adr	_Data	Read 16 bytes data in the corresponding block in the card
Write	0x47	_Adr，_Data	--	Write 16 bytes data in the corresponding block in the card
Increment	0x48	_Adr，_Value	--	Increase data value in the accessing block, save the data value in the register inside the card
Decrement	0x49	_Adr，_Value	--	Decrease data value in the accessing block, save the data value in the register inside the card
Restore	0x4A	_Adr	--	Recover data value in the accessing block to the register inside the card
Transfer	0x4B	_Adr	--	Transmit data value in the register inside the card to the accessing block
Value	0x70	_Mode，_Adr，_Value _Trans，_Adr	--	Include increment, decrement, recovery and transmission
LoadKey	0x4C	_Mode，_Secnr，_nkey	--	Change the key in the EEPROM key section
Reset	0x4E	_Msec	--	Turn off the antenna according to the specified millisecond to reset the card
Set_Control_Bit	0x50	--	--	Set the control output PIN of RHOMBUSMF6201ST as high level
Clr_Control_Bit	0x51	--	--	Set the control output PIN of RHOMBUSMF6201ST as high level
Config	0x52	--	--	Reset and config RHOMBUSMF6201ST
Check_Write	0x53	_Snr，_Authmode，	--	Compare the transmitted data with the data which is written last time

		_Adr _Data		
Buzzer	0x60	_Frequence， _Opentm， _Closetm， _Repcnt	--	Control buzzer drive output according to the specified freq., time interval and repeated times
Close	0x3F	--	--	Set RHOMBUSMF6201ST as standby state
Read_E2	0x61	_Adr， _Length	Data	Read the content of EEPROM in RHOMBUSMF6201ST
Write_E2	0x62	_Adr， _Length， _Data	--	Write the content of EEPROM in RHOMBUSMF6201ST

6. Status Value List

NAME	VALUE	DESCRIPTION
MI_OK，COMM_OK	0	Successful function transfer
MI_NOTAGERR	1	There is no card in the effective area
MI_CRCERR	2	CRC error
MI_EMPTY	3	Value overflow
MI_AUTHERR	4	Cannot be authenticated
MI_PARITYERR	5	Parity check error
MI_CODEERR	6	BCC error
MI_SENDERR	8	Card serial number error
MI_KEYERR	9	Error Code
MI_NOTAUTHERR	10	Card is not authenticated
MI_BITCOUNTERR	11	Inaccurate quantity bit received from the card
MI_BYTECOUNTERR	12	bytes of inaccurate quantity received from the card
MI_TRANSERR	14	TANSNFER error
MI_WRITEERR	15	WRITE error
MI_INCRERR	16	INCREMENT error
MI_DECRERR	17	DECREMENT error
MI_READERR	18	READ error
MI_COLLERR	24	Collision error
MI_ACCESSTIMEOUT	27	Overtime accessing
COMM_ERR	255	Serial communication error

7. Detailed description of operating command

7.1 **Request**

Send Request command, and check if there are cards in the effective area. This command must be executed before selecting a new card.

PC→RHOMBUSMF6201ST		
Command notation	0x41	
Length	1	
Data[0]	Mode	Mode=0，request cards in IDLE state in the area of antenna(except cards in HALT state) Mode=1,request all the cards in the antenna area

RHOMBUSMF6201ST→PC		
Status value	MI__OK， MI_QUIT， MI_NOTAGERR， MI_BITCOUNTERR，COMM_ERR	
Length	2	
Data[0]	_Tagtype（ low byte）	If error occurs, no content will be returned.
Data[1]	_Tagtype（ high byte）	If error occurs, no content will be returned.

7.2 **Anticoll**

After Request command, anti-collision should be invoked immediately. If serial number of the selected card is known, there is no need to invoke, only to use Select after Request command.

PC→RHOMBUSMF6201ST		
Command notation	0x42	
Length	1	
Data[0]	Reserved	Byte reserved, set as 0

RHOMBUSMF6201ST→PC		
Status value	MI_OK，MI_QUIT，MI_NOTAGERR，MI_BITCOUNTERR，COMM_ERR	
Length	4	
Data[0]	snr(LL)	Returned 4-bit serial number of the card
Data[1]	snr(LH)	

Data[2]	snr(HL)	
Data[3]	snr(HH)	

7.3 **Select**

Select card of specified number, return card capacity symbol to PC.

PC→RHOMBUSMF6201ST		
Command notation	0x43	
Length	4	
Data[0]	snr(LL)	4-bit serial number of card
Data[1]	snr(LH)	
Data[2]	snr(HL)	
Data[3]	snr(HH)	

RHOMBUSMF6201ST→PC		
Status Value	MI_OK,MI_QUIT，MI_NOTAGERR，MI_CRCERR， MI_PARITYERR MI_BYTECOUNTERR，COMM_ERR	
Length	1	
Data[0]	_size	The card capacity symbol

7.4 **Authentication**

Before carrying out read, writing, increment and decrement operation to the card, authentication must be executed. If the key of one sector in the card could be matched with the code of the corresponding key memory section in RHOMBUSMF6201ST, authentication succeeds.

PC→RHOMBUSMF6201ST		
Command notation	0x44	
Length	2	
Data[0]	_Mode	_Mode=0,use key A to carry out authentication _Mode=1,use key B to carry out authentication
Data[1]	_SecNr	Sector number of the accessing card must be less than 16, key section number is the same with the sector number of the card

RHOMBUSMF6201ST→PC		
Status value	MI_OK，MI_QUIT，MI_NOTAGERR，AUTHERR， MI_BITCOUNTERR，MI_PARITYERR, COMM_ERR	
Length	0	

7.5 **Halt**

Set the selected card as hanging state. If re-selection is needed, then apply ALL mode to invoke Request, or reset the card(If the card is out of the antenna operation area, place the card in again),or execute Reset.

PC→RHOMBUSMF6201ST		
Command notation	0x45	
Length	0	

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT，COMM_ERR	
Length	0	

7.6 **Read**

Read 16 bytes of the specified block after the selected card is authenticated.

PC→RHOMBUSMF6201ST		
Command Notation	0x46	
Length	1	
Data[0]	_Adr	block Address of the data that is read

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT ， MI_NOTAGERR，AUTHERR ，MI_CRCERR ，MI_NOTAUTHERR ，MI_BITCOUNTERR，MI_PARITYERR，COMM_ERR	
Length	16	
Data[0]	Data（0）	data of 16 bytes that is Read
Data[1]	Data（1）	
. . .	. . .	
Data[15]	Data（15）	

7.7 **Write**

The 16 bytes written in the specified block after the selected card is authenticated.

PC→RHOMBUSMF6201ST		
Command notation	0x47	
Length	17	
Data[0]	_Address	The block address of data that is read
Data[1]		The first byte to be written in block
…		…
Data[16]		The last byte to be written in block

RHOMBUSMF6201ST→PC

Status value	MI_OK，MI_QUIT，MI_NOTAGERR，AUTHERR，MI_BITCOUNTERR，MI_WRITEERR，COMM_ERR	
Length	0	

7.8 **Increment**

Read the accessed value block ; check the structure of the data. Transferred value plus the value of the value block, and save the result in the register inside the card. Value block has standard format..

PC→RHOMBUSMF6201ST		
Command notation	0x48	
Length	5	
Data[0]	_Address	The address of data block to be operated
Data[1]	_value(ll)	Value to be increased
Data[2]	_value(lh)	
Data[3]	_value(hl)	
Data[4]	_value(hh)	

RHOMBUSMF6201ST→PC		
Status value	MI_OK ，MI_QUIT，MI_NOTAGERR， MI_NOTAUTHERR，MI_BITCOUNTERR ，MI_PARITYERR，COMM_ERR	
Length	0	

7.9 **Decrement**

Read the accessed value block. check the structure of the data. Transferred value minus the value of the value block, and save the result in the register inside the card. Value block has standard format.

PC→RHOMBUSMF6201ST		
Command notation	0x49	
Length	5	
Data[0]	_Address	The address of data block to be operated
Data[1]	_value(ll)	Value to be decreased
Data[2]	_value(lh)	
Data[3]	_value(hl)	
Data[4]	_value(hh)	

RHOMBUSMF6201ST→PC		
Status value	MI_OK ，MI_QUIT，MI_NOTAGERR， MI_NOTAUTHERR，MI_BITCOUNTERR ，MI_PARITYERR，COMM_ERR	
Length	0	

7.10 **Restore**

Read the accessed value block ; check the structure of the data, and save the result in the register inside the card. Value block has standard format.

PC→RHOMBUSMF6201ST		
Command notation	0x4a	
Length	1	
Data[0]	_Address	Address of data block to be restored

RHOMBUSMF6201ST→PC		
Status value	MI_OK，MI_QUIT，MI_NOTAGERR，MI_NOTAUTHERR，MI_BITCOUNTERR，MI_PARITYERR，COMM_ERR	
Length	0	

7.11 **Transfer**

Transfer the content of register inside the card to the selected block. Authentication must be passed before this operation. This function can only be operated after the operation of Increment, Decrement or Restore.

PC→RHOMBUSMF6201ST		
Command notation	0x4b	
Length	1	
Data[0]	_Adr	The address to be transferred inside he card

RHOMBUSMF6201ST→PC		
Status value	MI_OK, MI_QUIT, MI_NOTAGERR, MI_NOTAUTHERR, MI_BITCOUNTERR, MI_PARITYERR，COMM_ERR	
Length	0	

7.12 **Load Key**

Write a new key to the key section of EEPROM memorizer. After this command, Code can be truly written in only after resetting RHOMBUSMF6201ST(invoke Config command or Close+config command)

PC→RHOMBUSMF6201ST		
Command notation	0x4c	
Length	8	
Data[0]	_Mode	_Mode=0: carry out authentication using code A

		_Mode=1: carry out authentication using code B
Data[1]	Secnr	The sector number where the key is must be less than 16.
Data[2]	_nkey[0]	The key of 6 bytes: the lowest byte is sent first. The lowest bit of the lowest byte is sent first.
…	…	
Data[7]	_nkey[5]	

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT， MI_PARITYERR，COMM_ERR	
Length	0	

7.13 **Reset**

RF circuit is closed in a given time , if Msec=0, part of RF circuit remain in closure status until the next Request command comes. Closing RF could reset all the cards in the area of antenna.

For example:

Msec=0, RF circuit has been closed all the time.

_Msec=1, RF circuit has been closed for 1ms.

Msec=0xff: RF circuit has been closed for 255ms.

PC→RHOMBUSMF6201ST		
Command Notation	0x4e	
Length	1	
Data[0]	_Msec	The period when RF circuit is closed(the unit is ms)

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT， COMM_ERR	
Length	0	

7.15 **Set_Control_Bit**

Set the control bit of RHOMBUSMF6201ST as high level. (interior pulling up to high level)

PC→RHOMBUSMF6201ST		
Command notation	0x50	
Length	0	

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT， COMM_ERR	

Length	0	

7.16 **Clr_Control_Bit**
Set the control bit of RHOMBUSMF6201ST as low level.

PC→RHOMBUSMF6201ST		
Command Notation	0x51	
Length	0	

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT， COMM_ERR	
Length	0	

7.17 **Config**

Every time after module is electrified and reset, further operation could only be carried out after this command is invoked to initialize the module.

PC→RHOMBUSMF6201ST		
Command Notation	0x52	
Length	0	

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT， COMM_ERR	
Length	0	

7.18 **Check_Write**

Check the data which is written into the card. Re-start Request/Select/Authenticated operation ,and compare the data in the corresponding address and the feedback data. If correct, return MIS_CHECK_OK. If the two data does not agree with each other, return MIS_CHK_ COMPERR. If other error occurs, return MIS_CHK_FAILED.

In this process, the number of the key block where the authenticated key is should be the same with the number of the sector where the block adr is.

PC→RHOMBUSMF6201ST		
Command Notation	0x53	
Length	22	

Data[0]	_Snr(ll)	Serial number of the card to be checked
Data[1]	_Snr(lh)	
Data[2]	_Snr(hl)	
Data[3]	_Snr(hh)	
Data[4]	_Authode	The authentication mode when command is written last time
Data[5]	_Adr	The address of the data block to be checked
Data[6]	Data(0)	Data of 16 bytes to be checked
. .	. .	
Data[20]	Data(21)	

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT， COMM_ERR	
Length	0	

7.19 **Buzzer**

Send a square wave or low-level drive external buzzer .The Frequency, the duration and the time interval and the repeated times could be controlled.

PC→RHOMBUSMF6201ST		
Command Notation	0x60	
Length	4	
Data[0]	_Frequency	Send the frequency of the square wave, range（0-255）; the corresponding frequency（0.73-4k）,0 as direct current，198 corresponds to 2k
Data[1]	_Opentm	The duration in which square wave is sent, range（0-255），resolution of 15ms
Data[2]	_Closetm	The time interval, range（0-255），resolution of 15ms
Data[3]	_Repcnt	The repeated times

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT， COMM_ERR	
Length	0	

7.20 **Read EEPROM**

Read the data of EEPROM in RHOMBUSMF6201ST.The data address must be less than 0x80.Keys are kept in the addresses starting from 0x80,and cannot be read.

PC→RHOMBUSMF6201ST		
Command Notation	0x61	
Length	2	
Data[0]	_Adr	The first address of EEPROM which is read must be less than 0x80
Data[1]	_Length	The length of data which is read（The data read for one time should be less than 20 bytes）

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT， COMM_ERR	
Length	_Length	
Data[0]	Byte（0）	The data which is read
. .	. .	
Data[_Length-1]	Byte（_Length-1）	

7.21 **Write EEPROM**

Write data in the EEPROM of RHOMBUSMF6201ST. 0x00-0x0F of EEPROM is read-only product information sections. 0x10-0x2f is register initialization documents starting-up sections, where data can not be rewritten. 0x80-0x1ff is read-only key sections, and keys can be written in with Load key command.

PC→RHOMBUSMF6201ST		
Command Notation	0x62	
Length	_Length +2	
Data[0]	_Adr	The fist address of EEPROM where data is written in，range（0x30-0x7e）
Data[1]	_Length	The length of the data which is written in（the data which is written in for one time must be less than 20 bytes.
Data[2]	Data（0）	The data to be written in
Data[length+1]	Data(Length)	

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT， COMM_ERR	
Length	0	

7.22 **Value Operation**

Carry out increment, decrement or data backup of some block in the card. The block should be value block format, and support automatic transfer. The address of this block and the address of the transfer block should be in the same sector.

PC→RHOMBUSMF6201ST		
Command notation	0x70	
Length	7	
Data[0]	_Mode	_Mode=0xc0：decrement _Mode=0xc1: increment _Mode=0xc2： restore
Data[1]	_Adr	The block address inside the card. Carry out value operation to this block, range: 0-63.
Data[2]	_value(ll)	When increment or decrement operation is carried out，this data is addend or subtrahend；when restoring operation is carried out, this value is null value.
Data[3]	_value(lh)	
Data[4]	_value(hl)	
Data[5]	_value(hh)	
Data[6]	_Trans_Adr	Transfer block address, range: 0-63

RHOMBUSMF6201ST→PC		
Status Value	MI_OK ， MI_QUIT ， COMM_ERR ， MI_NOTAERR ， MI_BITCOUNTERR，MI_TRANSERR	
Length	0	

7.23 **Close RHOMBUSMF6201ST**

Set RHOMBUSMF6201ST as standby state. Set operating current as smallest. If re-start, invoke Config command.

PC→RHOMBUSMF6201ST		
Command Notation	0x3f	
Length	0	

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT， COMM_ERR	
Length	0	

7.24 **Anticoll2**

Start anti-collision operation, which should be invoked immediately after Request command. If the serial number of the card is already known, anti-collision operation

is not needed , Select could be invoked directly.

PC→**RHOMBUSMF6201ST**		
Command Notation	0x71	
Length	2	
Data[0]	_Encoll	If it is 1，multi cards could enter; If it is 0，multi cards are not allowed to enter;if multi cards enter, return error cue.
Data[1]	Reserve	Reserve=0

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT，COMM_ERR，MI_NOTAGERR，MI_COLLERR，MI_BITCOUNTERR	
Length	4	
Data[0]	Snr(LL)	
Data[1]	Snr(LH)	
Data[2]	Snr(HL)	
Data[3]	Snr(HH)	

7.25 **Authentication2**

Before carrying out read, write, increment and decrement operation to the card, authentication must be executed. If the key of the key section accords with the key of the sector, authentication succeeds.

PC→RHOMBUSMF6201ST		
Command Notation	0x72	
Length	3	
Data[0]	_Mode	_Mode=0,use key A to carry out authentication _Mode=0,use key B to carry out authentication
Data[1]	_SecNr	Sector number of the accessed card should be less than 16,
Data[2]	_KeyNr	Key section number to be authenticated should be less than 16

RHOMBUSMF6201ST→PC		
Status Value	MI_OK ， MI_QUIT ， MI_NOTAGERR ， MI_PAROTERR ， MI_PAROTUERR，COMM_ERR	
Length	0	

7.26 **AuthKey**

Before carrying out read, write, increment and decrement operation to the card, authentication must be executed. If the transferred code accords with the key of the sector, authentication succeeds.

PC→RHOMBUSMF6201ST		
Command Notation	0x73	
Length	8	
Data[0]	_Mode	_Mode=0,use key A to carry out authentication _Mode=0,use key B to carry out authentication
Data[1]	_SecNr	Sector number of the accessed card should be less than 16
Data[2] . , Data[7]	_key(0) . . _key(5)	Code to be authenticated

RHOMBUSMF6201ST→PC		
Status Value	MI_OK，MI_QUIT，MI_NOTAGERR，MI_PAROTERR， MI_PAROTUERR，COMM_ERR	
Length	0	

Note: Rhombus' products must work with linear regulated power supply, and other kinds of power supply are prohibited.