JMY501G User's Manual

(Revision 3.42)

Jinmuyu Electronics Co. LTD

2011/6/27

Please read this manual carefully before using. If any problem, please mail to: Jinmuyu@vip.sina.com

Contents

1 Product introduction 3
2 Characteristics 3
3 Physical parameter and pin outs 4
3.1 Photo 4
3.2 Dimension 4
3.3 Pin configurations and pin outs 5
3.4 Antennas 5
3.5 Connection schematics 6
3.6 JMY500 testing board 6
3.7 Model rule 7
3.7.1 Model format 7
3.7.2 Card operating type 7
4 Communication protocols 8
4.1 Overview 8
4.2 UART protocol 8
4.2.1 Parameters 8
4.2.2 Data send format 8
4.2.3 Data return format 8
4.3 IIC protocol 9
4.3.1 Module IIC address and multi device communications 9
4.3.2 IIC device operation 9
4.3.2.1 Clock and data transaction 9
4.3.2.2 Start condition 9
4.3.2.3 Stop condition 9
4.3.2.4 Acknowledge (ACK) 10
4.3.2.5 Bus state 10
4.3.2.6 Device addressing 10
4.3.2.7 Write operation 10
4.3.2.8 Read operation 11
4.3.3 Data transaction 11
4.3.4 Data send format 11
4.3.5 Data return format 11
4.3.6 Description of IIC command transaction 12
5 Description of commands 13
5.1 List of commands 13
5.2 Explanation of commands 14
5.2.1 Read product information 14
5.2.2 Module working mode set 14
5.2.3 Set module idle 15
5.2.4 EEPROM read 15
5.2.5 EEPROM write 16
5.2.6 Set UART communication baud rate 16

5.2.7 Set IIC communication address 16
5.2.8 Set multi-card operation 17
5.2.9 Set ISO15693 automatic detecting card AFI and AFI enable 17
5.2.10 Set automatic detecting card interval time 18
5.2.11 ISO15693 inventory 18
5.2.12 ISO15693 stay quiet 19
5.2.13 ISO15693 get system information 19
5.2.14 ISO15693 reset to ready 19
5.2.15 ISO15693 read blocks 20
5.2.16 ISO15693 write blocks 20
5.2.17 ISO15693 block lock 20
5.2.18 ISO15693 AFI write 21
5.2.19 ISO15693 AFI lock 21
5.2.20 ISO15693 DSFID write 21
5.2.21 ISO15693 DSFID lock 22
5.2.22 ISO15693 get blocks security 22
5.3 About automatic detecting card 22
5.4 Example of commands 23
5.4.1 About UART communication protocol 23
5.4.2 UART commands sample 23
5.4.3 IIC commands sample 23
5.5 Interface program source code 23

1 Product introduction

JMY501G is RFID read/write module with an UART and IIC serial port. JMY501G has various functions and supports multi ISO/IEC standard of contactless card. The RF protocol is complex. The designer combined some frequent used command of RF card and then user could operate the cards with full function by sending simple command to the module.

The impedance between RF module and antenna was tuned by impedance analyzer. And then the module has excellent performance and stability.

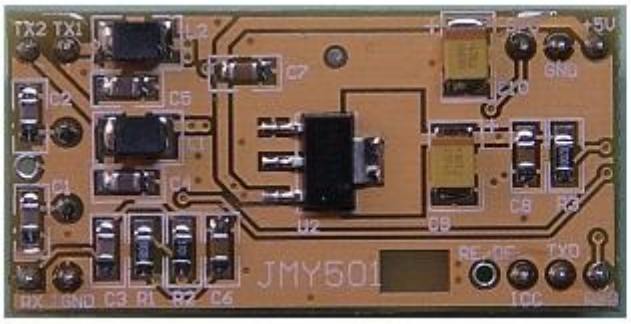
The module and antenna is split design. 4 wires are linked the antenna and module. Normally the wire should be less than 200mm, or it will affect the module's performance and stability.

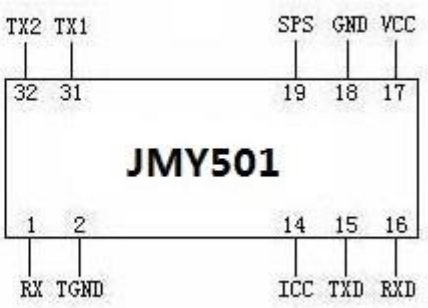
2 Characteristics

- PCD model: NXP SL RC400
- Working frequency: 13.56MHz
- Supported standard: ISO15693
- Card supported: TI TagIt, NXP I.Code SLI, ST LRI and ISO15693 tags
- Anti collision ability: Full function anti collision; be able to process multi-cards; be able to set operate single card only
- Auto detecting card: Supported, default ON
- EEPROM: 512 Bytes
- Power supply: DC 5V (±0.5V)
- Interface: IIC & UART (selected by SPS pin, recommend to use IIC)
- Communication rate: IIC: 400Kbps
  UART: 19.2Kbps/115.2Kbps
- Max. command length: 254 Bytes
- Interface level: 3.3V (TTL level; 5V tolerance)
- Power consumption: 70mA
- Operating distance: 100mm (depending on card and antenna design)
- Dimension: 21mm * 42mm
- Package: DIP32
- Weight: About 15g
- ISP: Supported
- Operating temperature: -25 to +85℃
- Storage temperature: -40 to +125℃
- RoHS: Compliant

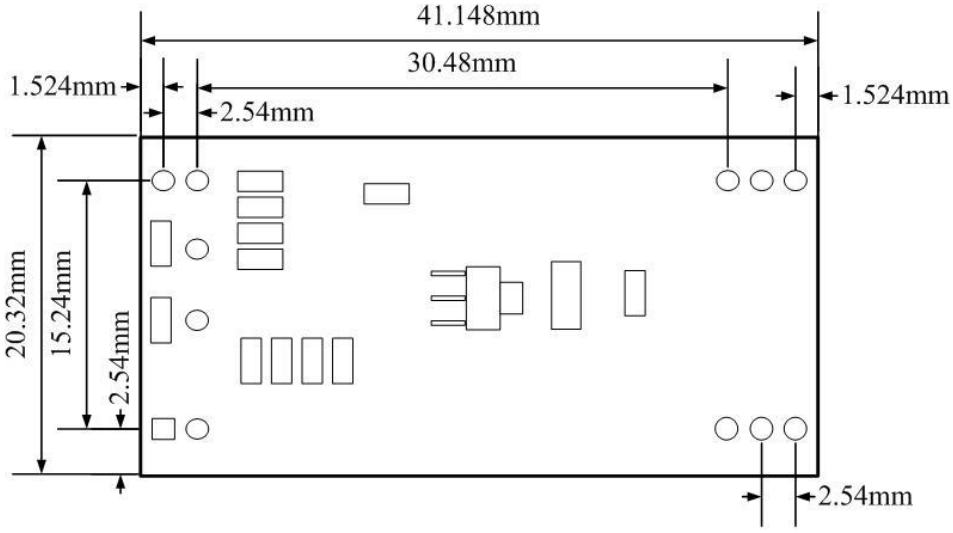
3 Physical parameter and pin outs

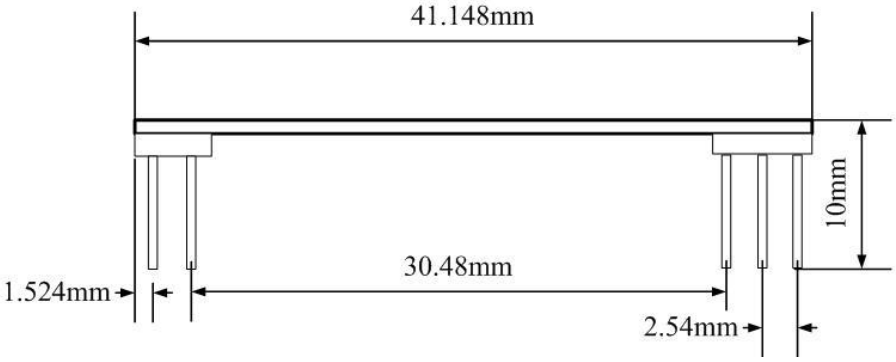
3.1 Photo





3.2 Dimension





3.3 Pin configurations and pin outs

PIN	Function	Type	Description
1	RX	RF Analog	Antenna receive
2	TGND	RF Analog	Antenna GND
13	RE	Output	RE/DE 485 directional control output
14	ICC	Output	Card in/out indication 0: card in; 1: card out
15	TXD/SDA	Input/output	UART TXD/IIC SDA
16	RXD/SCL	Input	UART RXD/IIC SCL
17	VCC	Power	VCC
18	GND	Power	GND
19	SPS	Input	Serial port selector 0: IIC 1: UART
31	TX1	RF Analog	Antenna output 1
32	TX2	RF Analog	Antenna output 2

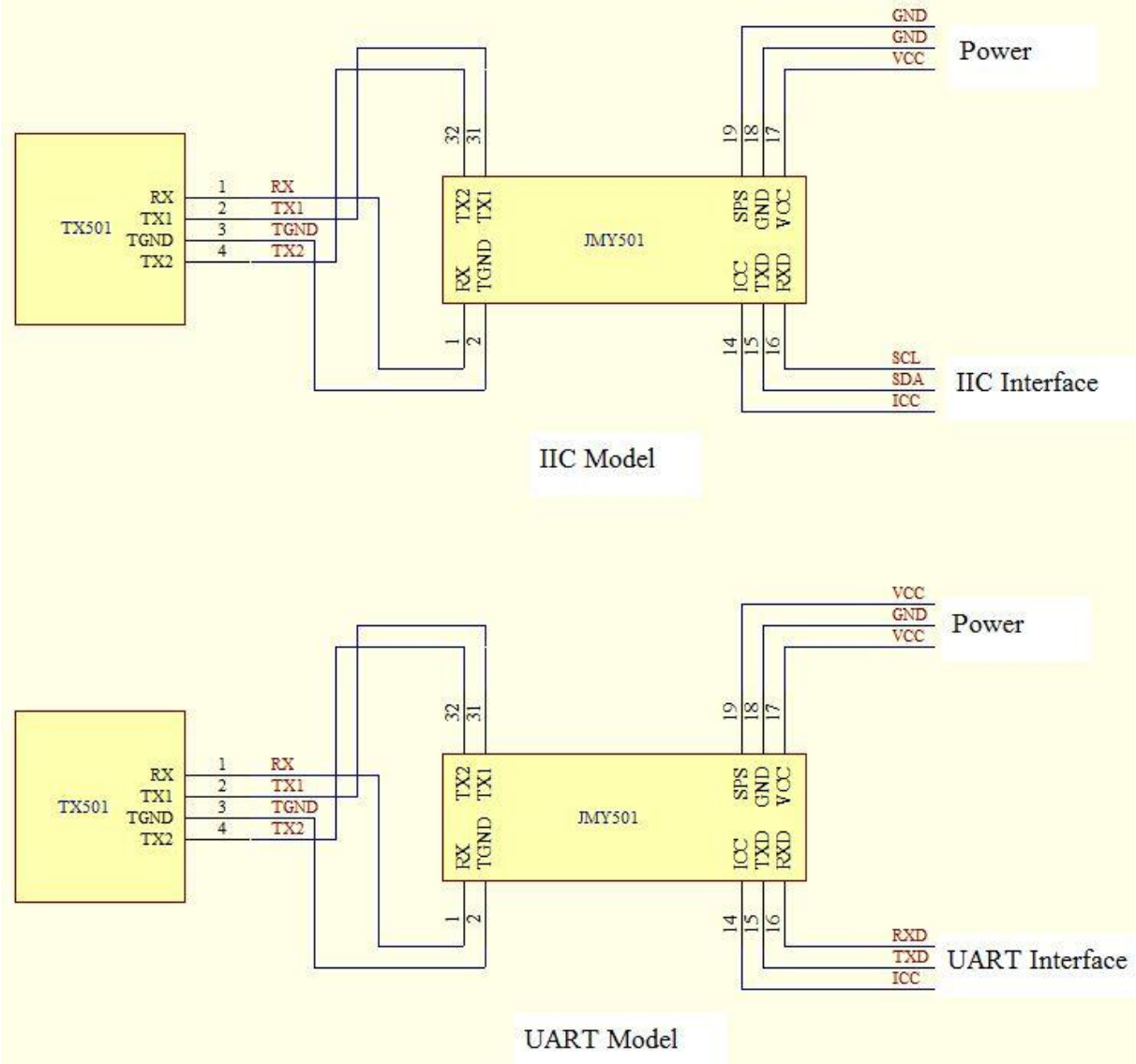
3.4 Antennas

Normally, as the size of TX50x may not meet the actual demands, the antenna needs to be customized, especially in some compact systems. The following information for customization is needed: 1. Dimension of the antenna PCB; 2. the position and direction of the antenna outlet and the connector; 3. the description of the antenna periphery. Jinmuyu will design the most proper antenna according to the user's exact requirements.

We provide many models of antenna. Please visit our website to get more information. There are some recommended models in the table:

Antenna model	Size of antenna	Card operating distance
TX500-2	70mm * 70mm	90mm
TX501-2	50mm * 50mm	70mm
TX502-2	30mm * 30mm	60mm

3.5 Connection schematics



3.6 JMY500 testing board

JMY500 testing board is a tool designed for testing of JMY50x series module, it could test the module completely with several steps. JMY500 operate the module via MCS51 MCU and it could to change the communication port (IIC or UART) of module. According to our source program (include IIC and UART), user is able to finish the program of application system.

JMY500 is also communicating with PC through RS232 port. Then user programs the testing software and completes the test to the module.

3.7 Model rule

3.7.1 Model format

1	2	3
JMY	501	X

1: company code; 2: product series code; 3: card operating type

3.7.2 Card operating type

M: PCD is RC500, support Mifare Class
A: PCD is RC500, support ISO14443A and Mifare Class
C: PCD is RC531, support ISO14443A, ISO14443B and Mifare Class
G: PCD is RC400, support ISO15693
H: PCD is RC632, support ISO15693, ISO14443A, ISO14443B and Mifare Class
D: PCD is RC500, support ISO14443A and Mifare Class with 511 bytes communication buffer
E: PCD is RC531, support ISO14443AB and Mifare Class with 511 bytes communication buffer

4 Communication protocols

4.1 Overview

The module has IIC and UART interfaces. We recommend using IIC interface whose communication rate is up to 400Kbps (normal rate is 100Kbps). But the baud rate of UART is 19.2Kbps and 115.2Kbps. We supply sample source code in C and ASM of MCS51 of the interface program both in IIC and UART. IIC mode is very convenient, user no need to modify the sample code except pin definition in actual using.

Whatever what type of interface user chooses. Please read this chapter before programming and refer to the sample program. There are detailed comments in the sample source code.

4.2 UART protocol

4.2.1 Parameters

The communication protocol is byte oriented. Both sending and receiving bytes are in hexadecimal format. The communication parameters are as follows:

- Baud rate: 19200bps(default), 115200bps
- Data bits: 8 bits
- Stop bits: 1 bit
- Parity check: None
- Flow control: None

4.2.2 Data send format

Header	Length	Command	Data	Checksum

- Header: 2 bytes, they are 0xAA 0xBB
- Length: 1 byte, number of bytes from Command length byte to the last byte of Data
- Command: 1 byte, the command of this instruction
- Data: length depends on the command type, length from 0 to 251 bytes
- Checksum: 1 byte, Exclusive OR (XOR) results from length byte to the last byte of data
- If there is 0xAA in data package, and then MUST insert 0x00 follow to distinguish with header. But Length byte in the package does NOT increase

4.2.3 Data return format

- Success:

Header	Length	Command	Data	Checksum

- Failure:

Header	Length	Invert Command	Checksum

4.3 IIC protocol

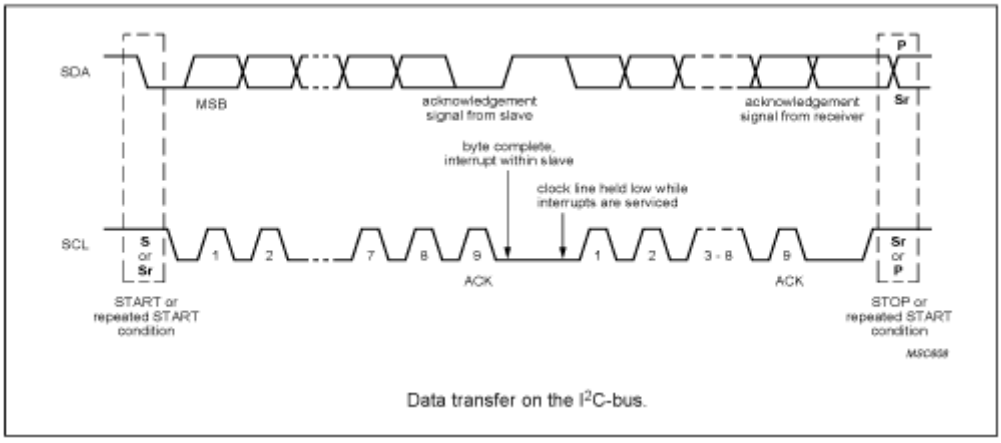
4.3.1 Module IIC address and multi device communications

IIC bus is able to connect with 128 devices. The IIC address of module is default 0xA0. Users change the address setting via sending the command (0x19), so that user could connect multi module on the same IIC bus.

4.3.2 IIC device operation

4.3.2.1 Clock and data transaction

The SDA pin is normally pulled high with an external device. Data on the SDA pin may change only during SCL low time periods. Data changes during SCL high periods will indicate a start or stop condition as defined below.



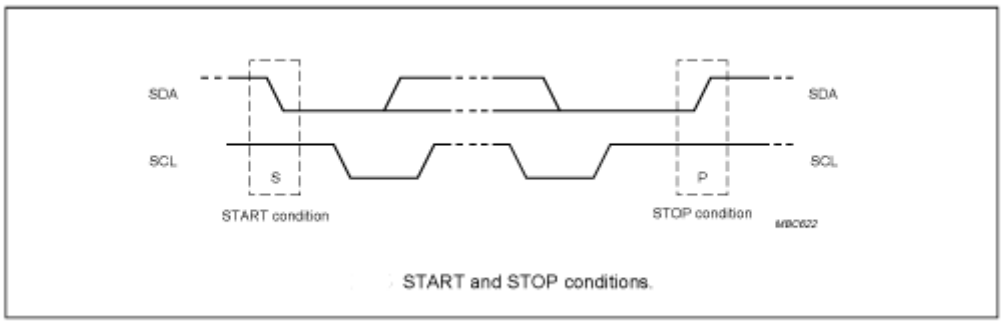
Data transfer on the I²C-bus.

4.3.2.2 Start condition

A high-to-low transition of SDA with SCL high is a start condition, which must precede any other command.

4.3.2.3 Stop condition

A low-to-high transition of SDA with SCL high is a stop condition.



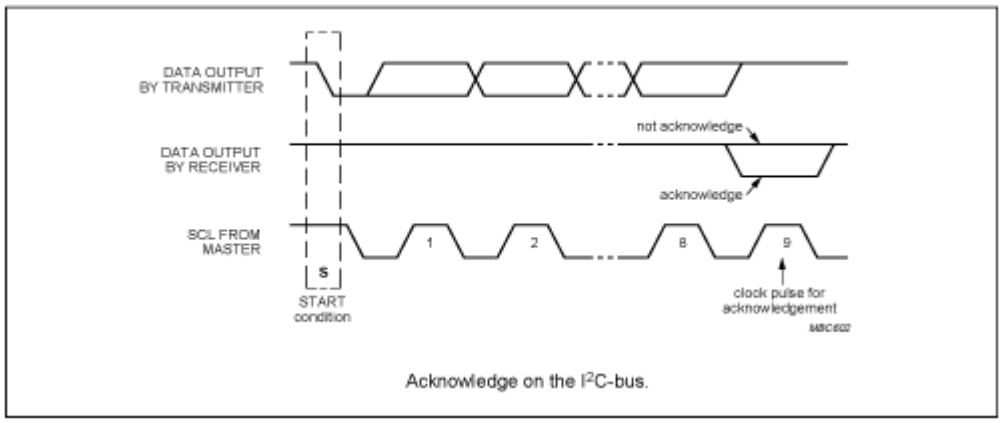
START and STOP conditions.

4.3.2.4 Acknowledge (ACK)

All addresses and data words are serially transmitted to and from the module in 8-bit words. The module sends a zero to acknowledge that it is not busy and has received each word. This happens during the ninth clock cycle.

4.3.2.5 Bus state

When the module has received command, and then doesn't acknowledge IIC bus until ends with the card communication.



Acknowledge on the I²C-bus.

4.3.2.6 Device addressing

The module requires a 7-bit device address following a start condition to enable the chip for a read or write operation.

The device address word consists of 7 addressing bits and 1 operation select bit.

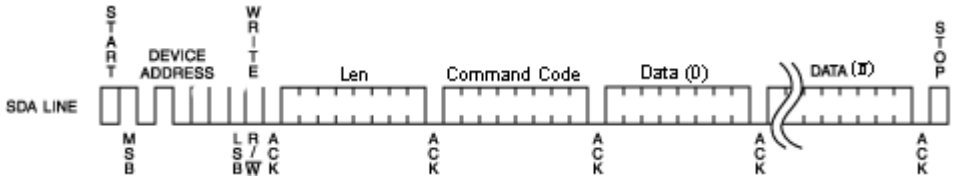
The first 7 bits of the module address are 1010000 (0xA0 in hex)

The eighth bit of the device address is the read/write operation select bit. A read operation is initiated if this bit is high and a write operation is initiated if this bit is low.

The first byte after the START procedure.

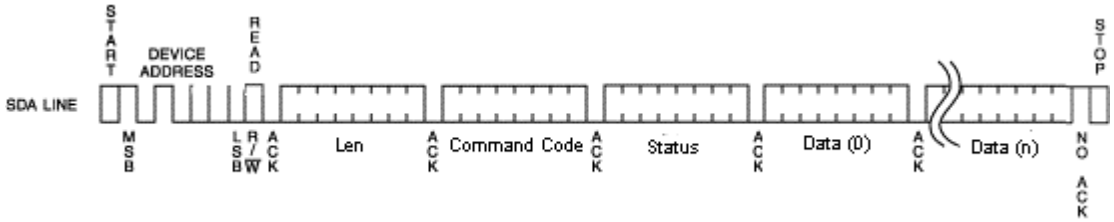
4.3.2.7 Write operation

The host device sends a command to module via write operation.



4.3.2.8 Read operation

The host device gets result via read operation.



4.3.3 Data transaction

The module is a slave device of the IIC bus, then the host need to write the command package to module. The module will execute the command. Then the host needs to poll the status of the module while it is working by sending out the command of "read" continuously. If the module answered to a read operation, then the last command execution were finished. At this time the host could read the result and/or data from the module. The read and write operation see chapter 4.3.2.7 and 4.3.2.8.

4.3.4 Data send format

Length	Command	Data	Checksum

- Length: 1 byte, number of bytes from length to the last byte of Data
- Command: 1 byte, the command of this instruction
- Data: Data length depending on the command type, length from 0 to 251 bytes
- Checksum: 1 byte, Exclusive OR (XOR) results from length byte to the last byte of data

4.3.5 Data return format

- Success:

Length	Command	Data	Checksum

- Failure:

Length	Invert Command	Checksum

4.3.6 Description of IIC command transaction

E.g.: to read the block 1 of Mifare card, the steps:
Send command: 0A210001FFFFFFFFFFFF2A
There are steps here:

A. Write command to module
   1. Start condition
   2. Send control byte, it is 0xA0, the meaning is: address 0xA0 + write control 0x00
   3. Send module command: 0x0A210001FFFFFFFFFFFF
   4. Send command checksum: 0x2A
   5. Stop condition

B. Send IIC read command. If module no ACK, then the module is working. Repeat this step.
   1. Start condition
   2. Send control byte 0xA1, it is IIC slave address 0xA0 + read control 0x01
   3. If module is no ACK, go to step B. if yes, go to step C

C. Get the data bytes from module
   1. Get the first byte and send ACK, if the data is 0x12, the meaning is there are 18 bytes useful bytes in this package.
   2. Get the else 17 bytes data and send ACK after every byte
   3. Get the checksum and send NACK
   4. Stop condition

D. Verify the checksum. if ok then the communication is ok

E. Verify the received data from second byte; this byte is the status of the command just executed. If equal to the command (0x21) then the command execute successful. Then the 16 bytes data started from third byte are correct.

5 Description of commands

5.1 List of commands

Command code	Command function
0x10	Read product information
0x11	Module working mode set
0x12	Sets module idle
0x15	EEPROM read
0x16	EEPROM write
0x17	Set UART communication baud rate
0x19	Set IIC address
0x1A	Set multi-card operation
0x1B	Set ISO15693 automatic detecting card AFI and AFI enable
0x1C	Set automatic detecting card interval time
0x5C	ISO15693 inventory
0x5D	ISO15693 stay quiet
0x5E	ISO15693 get system information
0x5F	ISO15693 reset to ready
0x54	ISO15693 read blocks
0x55	ISO15693 write blocks
0x56	ISO15693 block lock
0x57	ISO15693 AFI write
0x58	ISO15693 AFI lock
0x59	ISO15693 DSFID write
0x5A	ISO15693 DSFID lock
0x5B	ISO15693 get blocks security

5.2 Explanation of commands

5.2.1 Read product information

Function: read the product information of CURRENT PRODUCT, includes product name, firmware version, firmware date and configuration information.

Host sends:

0x02	0x10	Checksum

Module returns success:

0x1D	0x10	Information	Checksum

Information: 27 bytes, 8 bytes product name, 4 bytes firmware version, 8 bytes firmware date, 1 byte UART baud rate code, 1byte RFU, 1 byte IIC address, 1 byte multi-card operation enable state, 1 byte ISO15693 automatic detecting card's AFI, 1 byte ISO15693 automatic detecting card's AFI enable state, 1 byte automatic detecting card interval (multiple of 10ms).

Module returns failure:

0x02	0xEF	Checksum

5.2.2 Module working mode set

Function: set the antenna RF output ON/OFF; set the automatic detecting card ON/OFF. Antenna RF output is default ON, and automatic detecting card is OFF. The module will NOT SAVE the setting, and all settings will LOSE on next power up. The multi-card operation will be prohibited while users turn ON the automatic detecting card. If there is more than one card in the RF electric field then the operation will fail.

Host sends:

0x03	0x11	Mode	Checksum

Mode: 1 byte

Antenna status: BIT0 = 0: OFF; BIT0 = 1: ON

Auto request: BIT1 = 0: OFF; BIT1 = 1: ON

Module returns success:

0x02	0x11	Checksum

Module returns failure:

0x02	0xEE	Checksum

5.2.3 Set module idle

Function: set the module idle. In idle mode, the module of RF output turn to OFF, PCD power down, and CPU in idle mode, so the power consumption reduces to about 100uA. Sending the next command to module will wake up the module, and then the RF output ON and automatic detecting card restore default settings. The module will enter into idle mode after the answer procedure is finished. In IIC mode, host need to read the answer and then the module will goes into idle mode.

Host sends:

0x03	0x12	Random data	Checksum

Random data: 1 byte random data, for example: 0x55

Module returns success:

0x02	0x12	Checksum

Module returns failure:

0x02	0xED	Checksum

5.2.4 EEPROM read

Function: read data in EEPROM of the module.

Host sends:

0x05	0x15	Address	Bytes	Checksum

Address: 2 bytes, read start address, address from 0x0000 to 0x01FF, MSB first

Bytes: 1 byte, number of bytes to read, max. 64 bytes

Module returns success:

-	0x15	Data	Checksum

Remark: the byte length is "-", means the byte length depends on the card feedback information. (The same to below)

Data: data read

Module returns failure:

0x02	0xEA	Checksum

5.2.5 EEPROM write

Function: write data into EEPROM of the module

Host sends:

-	0x16	Address	Bytes	Data	Checksum

Address: 2 bytes, read start address, address from 0x0000 to 0x01FF, MSB first

Bytes: 1 byte, number of bytes to read, max. 64 bytes

Data: “Bytes” data to write

Module returns success:

0x02	0x16	Checksum

Module returns failure:

0x02	0xE9	Checksum

5.2.6 Set UART communication baud rate

Function: set UART communication baud rate of the module. After module receive the command, it will first save the new setting, and then send the execute result according to the host. At last it will validate the new setting. UART communication baud rate is default 19200bps. Settings will SAVE in the module; it will not be lost after power OFF.

Host sends:

0x03	0x17	Baud rate	Checksum

Baud rate: 1 byte, baud rate code; 0: 19200bps; 1: 115200bps; other values: RFU

Module returns success:

0x02	0x17	Checksum

Module returns failure:

0x02	0xE8	Checksum

5.2.7 Set IIC communication address

Function: set IIC communication address of the module. After module receive the command, it will first save the new address, and then send the executed result to the host. At last it will validate the new settings. The IIC address of the module is 1 byte HEX data. Lsb is 0; the address of module must be the even number, and the invalid address will NOT be

accepted. Settings will save in the module, and it will be not lost after power OFF.

Host sends:

0x03	0x19	Address	Checksum

Address: 1 byte, Lsb is 0; address must be the even number

Module returns success:

0x02	0x19	Checksum

Module returns failure:

0x02	0xE6	Checksum

5.2.8 Set multi-card operation

Function: set multi-card operation. If users need select on card from multi-card, then need to use the multi-card operation. If users set the automatic detecting card, the multi-card operation will be prohibited. If there is more than one card in the RF effective field then the operation will fail. Settings will save in the module; it will be not lost after power OFF. Multi-card operation default enables.

Host sends:

0x03	0x1A	Multi-card enable	Checksum

Multi-card enable: 1 byte, 0: disable multi-card; 1: enable multi-card; other values: RFU

Module returns success:

0x02	0x1A	Checksum

Module returns failure:

0x02	0xE5	Checksum

5.2.9 Set ISO15693 automatic detecting card AFI and AFI enable

Function: set AFI and AFI enables of automatic detecting card in ISO15693 mode. If users set AFI and AFI enables, then automatic detecting card only detects the AFI of the card equal to the set AFI. Settings will save in the module; it will be not lost after power OFF. AFI is default 0, AFI function is disable.

Host sends:

0x04	0x1B	AFI	AFI enable	Checksum

AFI: 1 byte, AFI, data values: 0~0xFF

AFI enable: 1 byte, 0: unable; 1: enable; other data value: RFU

Module returns success:

0x02	0x1B	Checksum

Module returns failure:

0x02	0xE4	Checksum

5.2.10 Set automatic detecting card interval time

Function: set interval time between two automatic detecting card

Host sends:

0x03	0x1C	Interval Time	Checksum

Interval Time: 1 byte, 0x00 to 0xFF, unit is 10mS, 0x01 means 10mS.

Module returns success:

0x02	0x1C	Checksum

Module returns failure:

0x02	0xE3	Checksum

5.2.11 ISO15693 inventory

Function: Find a card in RF effective field. If success, set the tag as CURRENT TAG

Host sends:

0x03	0x5C	AFI	Checksum

AFI: 1byte AFI, inventory card equal to AFI only

If not use AFI, then host sends:

0x02	0x5C	Checksum

Module returns success:

0x0B	0x5C	DSFID	UID	Checksum

DSFID: 1 byte, DSFID of CURRENT TAG

UID: 8 bytes, UID of CURRENT TAG

Module returns failure:

0x02	0xA3	Checksum

5.2.12 ISO15693 stay quiet

Function: set the CURRENT TAG stay quiet

Host sends:

0x02	0x5D	Checksum

Module returns success:

0x02	0x5D	Checksum

Module returns failure:

0x02	0xA2	Checksum

5.2.13 ISO15693 get system information

Function: get the system information of CURRENT TAG

Host sends:

0x02	0x5E	Checksum

Module returns success:

-	0x5E	Data	Checksum

Data: tag information, the length is a variable, depends on the manufacturer of the tag

Module returns failure:

0x02	0xA1	Checksum

5.2.14 ISO15693 reset to ready

Function: set a stay quiet TAG reset to ready

Host sends:

0x0A	0x5F	Data	Checksum

Data: 8 bytes, UID of the tag to reset to ready

Module returns success:

0x02	0x5F	Checksum

Module returns failure:

0x02	0xA0	Checksum

5.2.15 ISO15693 read blocks

Function: read data blocks of CURRENT TAG

Host sends:

0x04	0x54	Start	Blocks	Checksum

Start: 1 byte, read start block

Blocks: 1 byte, number of blocks to read, max. 62 blocks in one command

Module returns success:

-	0x54	Data	Checksum

Data: Blocks * 4 bytes,

Module returns failure:

0x02	0xAB	Checksum

5.2.16 ISO15693 write blocks

Function: write data blocks of CURRENT TAG

Host sends:

-	0x55	Start	Blocks	Data	Checksum

Start: 1 byte, write start block

Blocks: 1 byte, number of blocks needs to write, max. 62 blocks

Data: Blocks * 4 bytes, data to write to tag

Module returns success:

0x02	0x55	Checksum

Module returns failure:

0x02	0xAA	Checksum

5.2.17 ISO15693 block lock

Function: lock a block of CURRENT TAG

Host sends:

0x03	0x56	Block	Checksum

Block: 1 byte, block number to lock

Module returns success:

0x02	0x56	Checksum

Module returns failure:

0x02	0xA9	Checksum

5.2.18 ISO15693 AFI write

Function: write AFI to CURRENT TAG

Host sends:

0x03	0x57	AFI	Checksum

AFI: 1 byte, AFI value to write to tag

Module returns success:

0x02	0x57	Checksum

Module returns failure:

0x02	0xA8	Checksum

5.2.19 ISO15693 AFI lock

Function: lock AFI of CURRENT TAG

Host sends:

0x02	0x58	Checksum

Module returns success:

0x02	0x58	Checksum

Module returns failure:

0x02	0xA7	Checksum

5.2.20 ISO15693 DSFID write

Function: write DSFID of CURRENT TAG

Host sends:

0x03	0x59	DSFID	Checksum

DSFID: 1 byte, DSFID value to write to tag

Module returns success:

0x02	0x59	Checksum

Module returns failure:

0x02	0xA6	Checksum

5.2.21 ISO15693 DSFID lock

Function: lock DSFID of CURRENT TAG

Host sends:

0x02	0x5A	Checksum

Module returns success:

0x02	0x5A	Checksum

Module returns failure:

0x02	0xA5	Checksum

5.2.22 ISO15693 get blocks security

Function: get blocks security of CURRENT TAG

Host sends:

0x04	0x5B	Start	Blocks	Checksum

Start: 1 byte, start block

Blocks: 1 byte, number of blocks

Module returns success:

-	0x5B	Data	Checksum

Data: bytes equal to the sent blocks in the command, the locked info of data block

Module returns failure:

0x02	0xA4	Checksum

5.3 About automatic detecting card

The automatic detecting card function supports ISO15693. It is default ON. User could set the automatic detecting card on by send command 0x11. This setting will be lost on next power up.

Automatic detecting card supports full function of ISO15693.

Automatic detecting card supports only one card operation. If there is more than one card in the RF effective field then the operation may fail. Then the multi-card operation will automatically turn OFF while the automatic detecting card function turned ON.

5.4 Example of commands

5.4.1 About UART communication protocol

For example:

Write block, start block 0x08, write 2 blocks: AABB 0C55080211223344AA00BBCCDD17

AABB: Header of UART protocol, IIC protocol no this part

0C: package length; from 0C to DD are total 0x0C bytes, the 00 in red is a protocol byte, see chapter 4.2.2

55: instruction of ISO15693 writes block(s)

08: write start block

02: blocks to write

11223344AABBCCDD: data to write

00: protocol byte, used to distinguish header, see chapter 4.2.2

17: 0C ^ 55 ^ 08 ^ 02 ^ 11 ^ 22 ^ 33 ^ 44 ^ AA ^ BB ^ CC ^ DD = 17, in sample program, the function will calculate it, see chapter 4.5

5.4.2 UART commands sample

ISO15693 inventory	AABB035C005F
ISO15693 read blocks	AABB0454000858
ISO15693 write blocks	AABB0C55080211223344AA00BBCCDD17
ISO15693 get system information	AABB025E5C

5.4.3 IIC commands sample

ISO15693 inventory	035C005F
ISO15693 read blocks	0454000858
ISO15693 write blocks	0C55080211223344AABBCCDD17
ISO15693 get system information	025E5C

5.5 Interface program source code

We have interface program source code to help users. They are KELL project in C51 or ASM51 format. Please mail to jinmuyu@vip.sina.com to obtain the program.