OMT-680B User's Manual

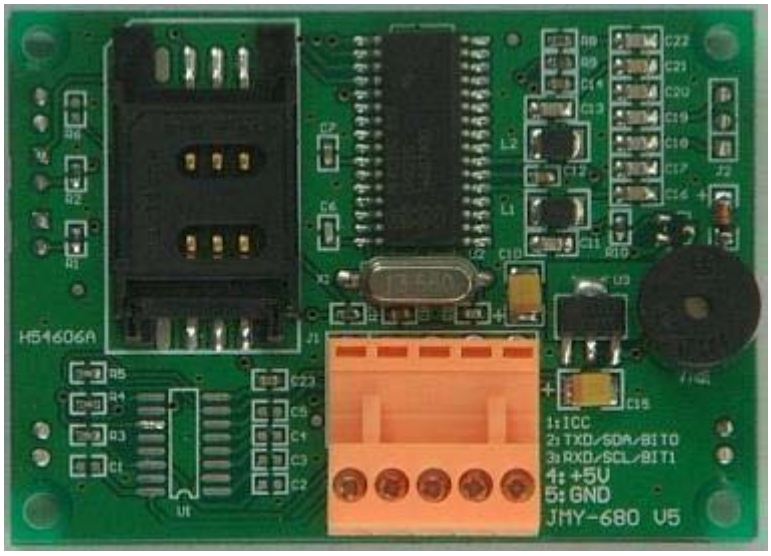
(Revision 2.10)

1 Features

- Supported card types: Mifare 1K/4K; UltraLight; ISO14443-4 TYPE A&B; ISO7816
- Auto detecting card: Supported, default OFF
- EEPROM: 512 bytes
- Power Supply: DC 4.5V to 5.5V
- Interface level: 3.3V (TTL level, 5V tolerance or RS232 level)
- Max. Power Consumption: 150mA
- Operate Distance: up to 70mm (depending on tag)
- Dimension: 50 * 70 (mm)
- Weight: About 100g
- ISP: Supported
- Operating temperature: -25 to +85 ℃
- Storage temperature: -40 to +125 ℃
- Wire connections: wire fix by screwdriver

2 Dimensions and Pins

2.1 Photo



2.2 Pin Configurations and Pinouts

Pin	Function	Type	Description
1	ICC	Output	Card in/out indicate 0: card in; 1: card out
2	TXD/SDA	Input/Output	UARTTXD/IIC SDA
3	RXD/SCL	Input	UART RXD/IIC SCL
4	VCC	Power	VCC
5	GND	Power	GND

2.3 Models

- OMT-680BI IIC interface
- OMT-680BST UART interface, TTL level
- OMT-680BSR RS232, (UART interface, RS232 level)

3 Protocols

3.1 UART protocol

3.1.1 Parameters

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

Baud rate: 19200 bps
Data: 8 bits
Stop: 1 bit
Parity: None
Flow control: None

3.1.2 Package format

Length	Command	Data	Checksum

Length: 1 byte, number of bytes from Command length to the last byte of Data.
Command: 1 byte, command.
Data: data, length depending on the command type, it may be empty.
Checksum: Exclusive OR (XOR) result from Command length byte to the last byte of data.

3.1.3 Data return format of UART

Success:

Length	Command	Data	Checksum

Failure:

Length	~ Command	Checksum

Remark: ~ Command is invert command, if the command is 0x12, ~ Command is 0xED

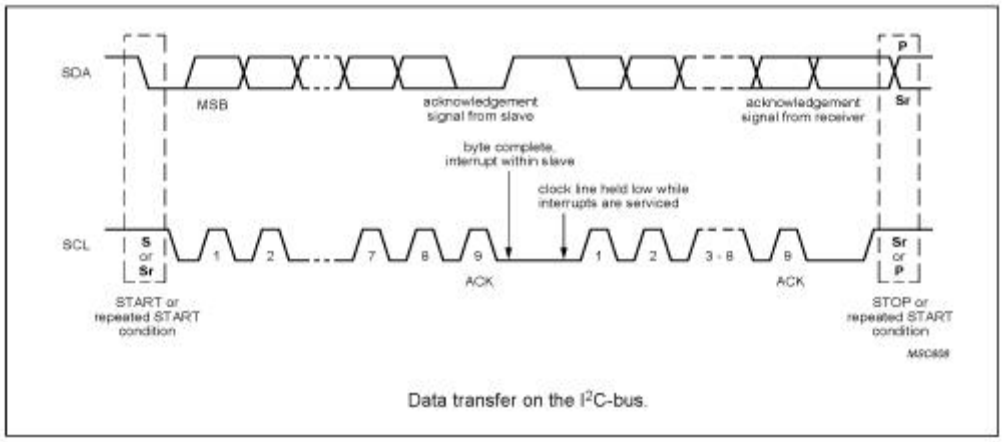
3.2 IIC protocol

3.2.1 IIC address of module is 0xA0

3.2.2 IIC Device Operation

3.2.2.1 Clock and data transactions

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.



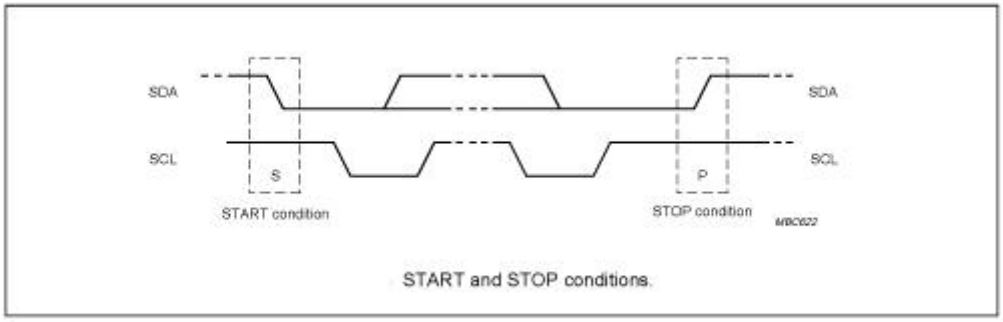
Data transfer on the I²C-bus.

3.2.2.2 Start condition

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

3.2.2.3 Stop condition

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



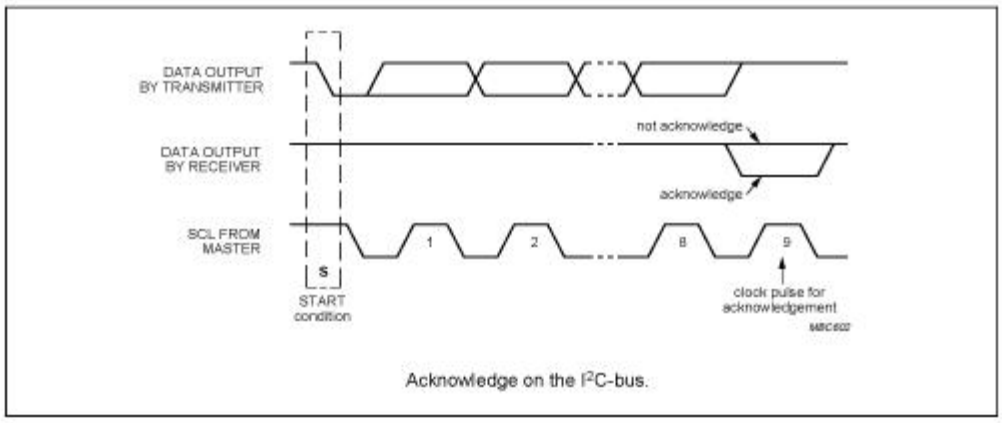
START and STOP conditions.

3.2.2.4 Acknowledge

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.

3.2.2.5 Bus state

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



Acknowledge on the I²C-bus.

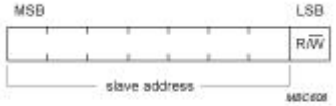
3.2.2.6 Device addressing

The modules require an 8-bit device address word 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)

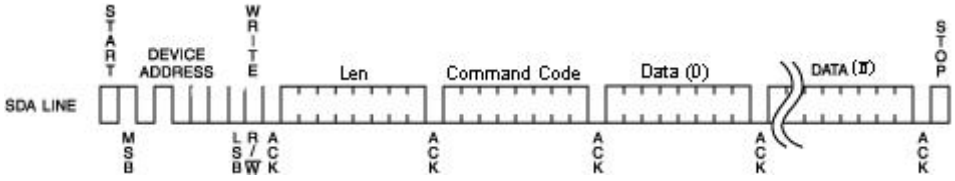
The eighth bit of the device address is the read/write operation select bit. A read operation is initiated if this bit is high and a write operation is initiated if this bit is low.



The first byte after the START procedure.

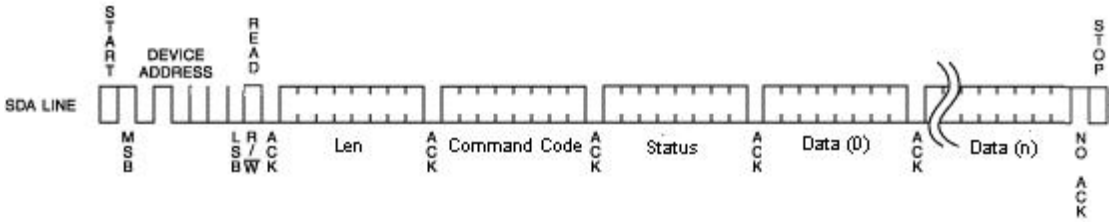
3.2.2.7 Write Operations

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



3.2.2.8 Read Operations

The host device using read operation to gets the result.



3.2.3 Data transaction

The module is a slave device of the IIC bus, then the host need to write the command package to module. After writing, host need to poll the status of the module while it is working. If the module answered to a read operation, then the last command execution were finished, then the host could read the result and/or data from the module. The read and write operation see chapter 3.2.2.7 and 3.2.2.8.

3.2.4 Data format of package

Length	Command	Data	Checksum

Length: 1 byte, number of bytes from length to the last byte of Data.
Command: 1byte, command.
Data: data, length depending on the command type, maybe empty.
Checksum: 1byte, exclusive OR (XOR) result from Length byte to the last byte of data.

3.2.5 Data return format of IIC protocol

Success:

Length	Command	Data	Checksum

Failure:

Length	~ Command	Checksum

4 Description of commands

4.1 Overview of commands

Command	Description
0x11	Module working mode set
0x13	LED set
0x14	Buzzer set
0x20	Request cards
0x21	Data block read
0x29	Sector (4 blocks) Read
0x22	Data block Write
0x23	Initializing a purse block
0x24	Purse read
0x25	Purse increments
0x26	Purse decrements
0x27	Purse copy
0x28	Mifare card halt
0x2D	Download card key to module
0x15	EEPROM read
0x16	EEPROM write
0x30	ISO14443-4 TYPE A card reset
0x31	Send APDU to ISO14443-4 card
0x41	UltraLight card read
0x42	UltraLight card write
0x70	Set card operate model
0x60	ISO14443-4 TYPE B card reset
0x62	ISO14443-4 TYPE B card halt
0x51	Reset SAM
0x53	Send APDU to SAM

4.2 List of commands

4.2.1 Module working mode set

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 return success:

0x02	0x11	Checksum

Module return failure:

0x02	0xEE	Checksum

Remark: the module will NOT SAVE the settings. All settings will be lost on next power up.

4.2.2 LED set

Host sends:

0x03	0x13	State	Checksum

State: 1 byte, 0: OFF; 1: ON

Module return success:

0x02	0x13	Checksum

Module return failure:

0x02	0xEC	Checksum

4.2.3 Buzzer set

Host sends:

0x03	0x14	Time	Checksum

Time: 1 byte, unit is 10mS. If Time is 0x0A then the beep time is 100mS

Module return success:

0x02	0x14	Checksum

Module return failure:

0x02	0xEC	Checksum

4.2.4 Request cards

Host sends:

0x03	0x20	Mode	Checksum

Mode: 1 byte, 0: WUPA (request all); all other values: REQA (Request not halted only)

Module return success:

-	0x20	Data	Checksum

Data: 4; 7 or 10 bytes card serial number + 2 bytes ATQA + 1 byte SAK

Module return failure:

0x02	0xDF	Checksum

4.2.5 Data block read

Host sends:

0x0A	0x21	KeyID	Block	Key	Checksum

Key ID: 1 byte, Key identification

BIT0=0: Key A; BIT0=1: Key B;

BIT1=0: using the key follow; BIT1=1: using the downloaded by command 0x2D

(IMPORTANT: please read Chapter 4.3 about Key identification)

Block: 1 byte, Block number to read, 0 to 0x3F for S50; 0 to 0xFF for S70

Key: 6 bytes, the key of the card

Module return success:

0x12	0x21	Data	Checksum

Data: 16 bytes card data

Module return failure:

0x02	0xDE	Checksum

4.2.6 Sector (4 blocks) Read

Host sends:

0x0A 0x29	KeyID	Sector	Key	Checksum

Key ID: 1 byte, Key identification

Sector: 1 byte, Sector number to read, 0 to 0x0F for S50; 0 to 0x3F for S70

Key: 6 bytes, the key of the card

Module return success:

0x42	0x29	Data	Checksum

Data: 64 bytes card data

Module return failure:

0x02	0xD6	Checksum

Remark: for S50 and sector number less than 32 of S70, this command is called read sector, it will read the sector trailer. For sector 32 to 39 of S70, this command is called "read 4 blocks". Because the sectors are include 16 blocks, then module will read 4 blocks. If you need to read the 16 blocks in these sectors, you need do this command 4 times to fill the require. The "Sector" in package is: read start block number shift right 2 bits.

4.2.7 Data block Write

Host sends:

0x1A	0x22	Key ID	Block	Key	Data	Checksum

Key ID: 1 byte, Key identification

Block: 1 byte, Block number to write, 0 to 0x3F for S50; 0 to 0xFF for S70

Key: 6 bytes, the key of the card

Data: 16 bytes data to write

Module return success:

0x02	0x22	Checksum

Module return failure:

0x02	0xDD	Checksum

4.2.8 Initializing a purse block

Host sends:

0x0E	0x23	Key ID	Block	Key	Value	Checksum

Key ID: 1 byte, Key identification

Block: 1 byte, Block number to initialize, 0 to 0x3F for S50; 0 to 0xFF for S70

Key: 6 bytes, the key of the card

Value: 4 bytes, initialized value, LSB first

Module return success:

0x02	0x23	Checksum

Module return failure:

0x02	0xDC	Checksum

4.2.9 Purse read

Host sends:

0x0A	0x24	KeyID	Block	Key	Checksum

Key ID: 1 byte, Key identification

Block: 1 byte, block number of the value to read, 0 to 0x3F for S50; 0 to 0xFF for S70

Key: 6 bytes, the key of the card

Module return success:

0x06	0x24	Data	Checksum

Data: 4 bytes value data, LSB first

Module return failure:

0x02	0xDB	Checksum

Remark: module will read the data in block and check if it is a purse format. If yes, return 4 bytes value data, if no, return failure.

4.2.10 Purse increments

Host sends:

0x0E	0x25	KeyID	Block	Key	Value	Checksum

Key ID: 1 byte, Key identification

Block: 1 byte, block number to initialize, 0 to 0x3F for S50; 0 to 0xFF for S70

Key: 6 bytes, the key of the card

Value: 4 bytes, increment value, LSB first

Module return success:

0x02	0x25	Checksum

Module return failure:

0x02	0xDA	Checksum

4.2.11 Purse decrements

Host sends:

0x0E	0x26	KeyID	Block	Key	Value	Checksum

Key ID: 1 byte, Key identification

Block: 1 byte, Block number to initialize, 0 to 0x3F for S50; 0 to 0xFF for S70

Key: 6 bytes, the key of the card

Value: 4 bytes, increment value, LSB first

Module return success:

0x02	0x26	Checksum

Module return failure:

0x02	0xD9	Checksum

4.2.12 Purse copy

Host sends:

0x0E	0x27	KeyID	Source	Target	Key	Checksum

Key ID: 1 byte, Key identification

Source: 1 byte, block number to copy, 0 to 0x3F for S50; 0 to 0xFF for S70

Target: 1 byte, copy the purse to this block (source and target need in 1 sector)

Key: 6 bytes, the key of the card

Module return success:

0x02	0x27	Checksum

Module return failure:

0x02	0xD8	Checksum

4.2.13 Mifare card halt

Host sends:

0x02	0x28	Checksum

Module return success:

0x02	0x28	Checksum

Module return failure:

0x02	0xD7	Checksum

4.2.14 Download card key to module

Host sends:

0x09	0x2D	KeyIndex	Key	Checksum

Key Index: 1 byte, Key Index of store in the module

Key: 6 bytes, the key of the card to store in module

Module return success:

0x02	0x2D	Checksum

Module return failure:

0x02	0xD2	Checksum

4.2.15 EEPROM read

Host sends:

0x05	0x15	Address	Bytes	Checksum

Address: 2 bytes, read start address

Bytes: 1 byte, number of bytes to read, max. 16 bytes

Module return success:

-	0x15	Data	Checks

Data: data read

Module return failure:

0x02	0xEA	Checksum

4.2.16 EEPROM write

Host sends:

-	0x16	Address	Bytes	Data	Checksum

Address: 2 bytes, read start address

Bytes: 1 byte, number of bytes to read, max. 16 bytes

Data: "Bytes" data to write

Module return success:

0x02	0x16	Data	Checksum

Module return failure:

0x02	0xE9	Checksum

4.2.17 ISO14443-4 card reset

Host sends:

0x02	0x30	Checksum

Module return success:

-	0x30	Info	Checksum

Info: card reset information, length determined by card

Module return failure:

0x02	0xCF	Checksum

Remark: length is "-" mean depending on the actual card response

Remark: this command must be used after detect card, see chapter 4.2.3

4.2.18 Send APDU to card

Host sends:

-	0x31	APDU	Checksum

APDU: APDU to send

Module return success:

-	0x31	Response	Checksum

Response: card response

Module return failure:

0x02	0xCE	Checksum

4.2.19 UltraLight card read

Host sends:

0x05	0x41	Block	Checksum

Block: 1 byte, read start block number

Module return success:

0x12	0x41	Data	Checksum

Data: 16 bytes card data of 4 blocks

Module return failure:

0x02	0xBE	Checksum

4.2.20 UltraLight card write

Host sends:

0x05	0x42	Block	Data	Checksum

Block: 1 byte, write block number

Data: 4 bytes data to write

Module return success:

0x12	0x42	Checksum

Module return failure:

0x02	0xBD	Checksum

4.2.21 Set card operate model

Host sends:

0x03	0x70	Model	Checksum

Model: 1 byte, 0: ISO14443 type A; 1: ISO14443 type B

Module return success:

0x12	0x70	Checksum

Module return failure:

0x02	0x8F	Checksum

4.2.22 ISO14443-4 TYPE B card reset

Host sends:

0x03	0x60	Model	Checksum

Model: 1 byte, 0: WUPB (request all); all other values: ATQB (request not halted card only)

Module return success:

-	0x60	Info	Checksum

Info: 12 bytes, card reset information

Module return failure:

0x02	0x9F	Checksum

4.2.23 ISO14443-4 TYPE B card halt

Host sends:

0x03	0x62	PUPI	Checksum

PUPI: 4 bytes, PUPI of the card to halt

Module return success:

0x02	0x62	Checksum

Module return failure:

0x02	0x9D	Checksum

4.2.24 Reset SAM

Host sends:

0x02	0x51	Checksum

Module success return:

-	0x51	Info	Checksum

Info: card reset information, length determined by card

Module failure return:

0x02	0xAE	Checksum

4.2.25 Send APDU to SAM

Host sends:

-	0x53	APDU	Checksum

APDU: APDU to send

Module success return:

-	0x53	Response	Checksum

Response: SAM response

Module failure return:

0x02	0xAC	Checksum

4.3 About KEY Identification

There is a byte of KEY identification in some commands. This byte will identify the way to get the card key.

Key Identification							
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0							

BIT0 0: KEY A; authenticate Key A of the card.
1: KEY B; authenticate Key B of the card.

BIT1 0: Using the following Key in command.
1: Using the downloaded Key by command 0x2D.

BIT6:BIT5:BIT4:BIT3:BIT2: Index of the Key already downloaded (0 to 0x1F).

If BIT1 is 0, then these 5 bits (BIT6 to BIT2) are unused. If BIT1 is 1, then use the already downloaded key. Users need to download key(s) by using command 0x2D first; and then the 6 bytes key in the command are left unused.

4.4 Example of commands

4.4.1 How to use

For example:

Read block 1: 0A210001AABBCCDDEEFF2A

0A: package length; from 0A to FF are total 0x0A bytes

21: instruction of read

00: Authenticate KEY A, using the key in package. The key is "AABBCCDDEEFF"

01: block number to read

AABBCCDDEEFF: key of the sector of the card

2A: 0A ^ 21 ^ 00 ^ 01 ^ AA ^ BB ^ CC ^ DD ^ EE ^ FF = 2A, in sample program, the function will calculate it, see chapter 4.5

4.4.2 UART commands

Read block 1 0A210001FFFFFFFFFFFF2A

Read block 255 (S70) 0A2100FFFFFFFFFFFFFFFD4

Write block 1 1A220001FFFFFFFFFFFF1234567890ABCDEF1234567890ABCDEF39

Request card (WUPA) 03200023

Halt card 021210

4.4.3 IIC commands

Read block 1 0A210001FFFFFFFFFFFF2A

Read block 255 (S70) 0A2100FFFFFFFFFFFFFFFD4

Write block 1 1A220001FFFFFFFFFFFF1234567890ABCDEF1234567890ABCDEF39

Request card (WUPA) 03200023

Halt card 021210