RFID READER

SL144 Reader

Development Handbook

Version 2.0
Aug 2019
StrongLink

CATALOGUE

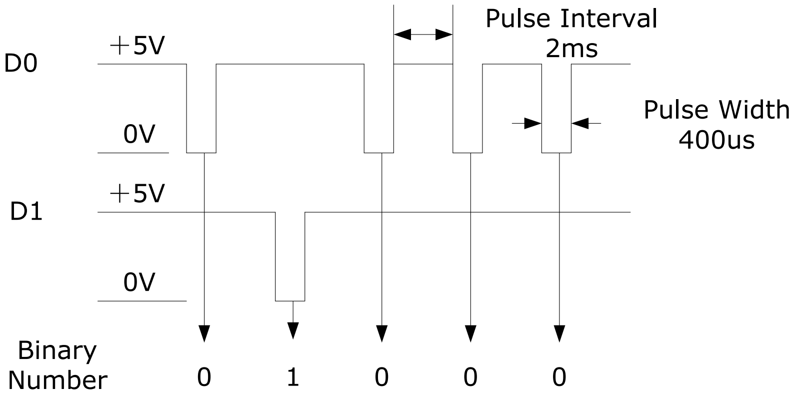
1. Applying development of unilateral communication

1.1 Applying development of unilateral communication

Apply in software development of continuous and trigger working mode, upper computer only need to receive tag ID numbers sent by reader, no need to send reader command.

1.2 Wiegand interface agreement

Wiegand interface transmites unilaterally, can send read card numbers to controller only, but controller can not send signal to reader. Signal transmitted from wiegand interface as follows:



At present, output waveform from Wiegand interface is as follows:

400us in pulse width， 2.0ms in pulse interval

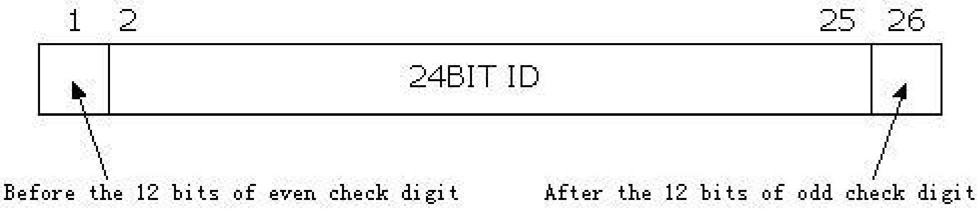
100us in pulse width， 1.6ms in pulse interval

50us in pulse width， 1.0ms in pulse interval

Wiegand interface has 2 types of Wiegand26 and Wiegand34.

1.2.1 Wiegand26 format

Wiegand26 transmits 26 bits data every time, 24 bits of them are valid. We stipulate these 24 bits correspondent for low 24 bits of electronic tag ID, or user defined number. Transmission format as follows:

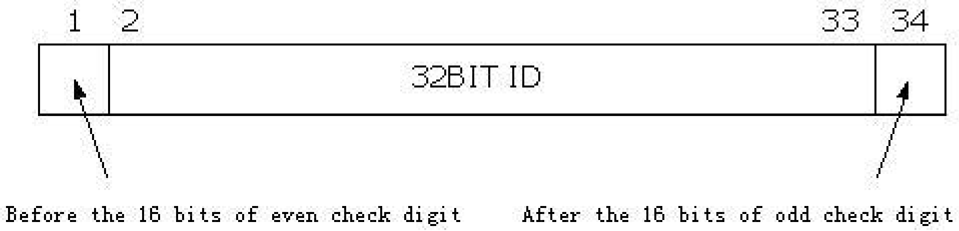

1
2
25
26
24BIT ID
Before the 12 bits of even check digit
After the 12 bits of odd check digit

Even: verified data adds parity bit1 is even number.

Odd: verified data adds parity bit1 is odd number.

1.2.2 Wiegand34 format

Wiegand34 transmits 32 bits of valid data every time. We stipulate these 32 bits correspondent for low 32 bits of electronic tag ID, or user defined number. Transmission format as follows:


1
2
33
34
32BIT ID
Before the 16 bits of even check digit
After the 16 bits of odd check digit

1.3 RS485 interface agreement

When adopt RS485 interface to output data, need to set up communication rate of RS485 interface in advance. Output data format of RS485 is:

BODY				CHECK
STX	DATA		ETX	BCC
02	Antenna serial number(2byte ASCII code)	ID number or User-defined number(N bytes ASCII code)	03	check code

Introduction:

Symbol of data start STX = 02H ， symbol of data end ETX = 03H；

DATA is antenna serial number(2Byte)+tag ID number(NBytes)，length is N+2bytes ASCII code. Expression mode of converting HEX to ASCII:

	Length(byte)	Description
STX	1	Symbol of data startSTX = 02H
Data	N	Antenna serial number(2byte ASCII code) ID number or User-defined number(N bytes ASCII code)
ETX	1	Symbol of data end ETX = 03H
Check	1	Check code

Divides data from high to low, every 4 bits in a team, then put value of 4 binary bits in expression of ASCII code. As value range of 4 binary bits is 0H～FH, converted ASCII code is 30H～39H, 41H～46H. For example: data of 32 serial number is 6A90F103H, it is『36H 41H 39H 30H 46H 31H 30H 33H』after converting to 8 bytes ASCII code. Antenna number 1(ASCII code) is『30H 31H』， antenna number 2(ASCII code) is『30H 32H』.

2. Serial port intercommunication agreement

Two ways for application development:

1) Use control code of serial port communication agreement to operate reader directly.

2) User matched SDK software with reader to call API function to operate reader.

2.1 Summarize

In RFID application system, reader is connected with controller (PC) via RS232 port, and receives commands from controller, then returns the result of command execution. Therefore, we name Data Communication Packet from controller to reader, to be Command Packet, and name that from reader to controller, to be Return Packet.

2.1.1 Command packet format without address

BootCode	Length	Command	Command Param	CheckSum

See chart above , command packet is composed of 5 parts:

	Length(byte)	Description
BootCode	1	Fixed to be 40H
Length	1	Effective length. The length is total bytes of lateral 3 parts
Command	1	Command code
Command Param	uncertain	Command parameter, length is changing with command
CheckSum	1	Checksum, is all bytes from bootcode to command param, discard patch code

2.1.2 Command packet format with address

BootCode	Length	Command	Address	Command Param	CheckSum

See chart above , command packet is composed of 6 parts:

	Length(byte)	Description
BootCode	1	Fixed to be 40H
Length	1	Effective length. The length is total bytes of lateral 4 parts
Command	1	Command code
Address	1	Reader address, 1～254, 0 and 255 is broadcast address
Command Param	uncertain	Command parameter, length is changing with command
CheckSum	1	Checksum, is all bytes from bootcode to command param, discard patch code

2.1.3 Return packet format without address

BootCode	Length	Command	Return Data	CheckSum

See chart above, return packet is composed of 5 parts:

	Length(byte)	Description
BootCode	1	When execute command correctly, bootcode of return packet is F0H. When execute command fail, bootcode of return packet is F4H.
Length	1	Effective length. The length is total bytes of lateral 3 parts.
Command	1	Command code, same as received command code, means that return packet is the reaction to the command.
Return Data	uncertain	Return the result of command execution. Length is changing with command.
CheckSum	1	Checksum, is all bytes from bootcode to ReturnData, discard patch code.

2.1.4 Return packet format with address

BootCode	Length	Command	Address	Return Data	CheckSum

See chart above , return packet is composed of 6 parts:

	Length(byte)	Description
BootCode	1	When execute command correctly, bootcode of return packet is F0H. When execute command fail, bootcode of return packet is F4H.
Length	1	Effective length. The length is total bytes of lateral 4 parts.
Command	1	Command code, same as received command code, means that return packet is the reaction to the command.
Address	1	Reader address, 1～254, 0 and 255 is broadcast address
Return Data	uncertain	Return the result of command execution. Length is changing with command.
CheckSum	1	Checksum, is all bytes from bootcode to ReturnData, discard patch code.

2.1.5 Error code

When execute command fail, bootcode of return packet is F4H, and ReturnData is 1 byte of error code. Common error code is:

00(00H)	Command success or detect correct
01(01H)	Antenna connection fail
02(02H)	Detect no tag
03(03H)	illegal tag
04(04H)	read write power is inadequate
05(05H)	Write protection in this area
06(06H)	Checksum error
07(07H)	Parameter wrong
08(08H)	Nonexistent data area
09(09H)	Wrong password
10(0AH)	kill password can’t be 0
11(0BH)	When reader in Auto mode, the command is illegal.
12(0CH)	Illegal user with unmatched password
13(0dH)	RF interference from external
14(0EH)	Read protection on tag
……	
30(1EH)	Invalid command, such as wrong parameter command
31(1FH)	Unknown command
32(20H)	Other error

2.1.6 For example

For example: set baud rate of reader to be 9600bps, command packet is『40H 03H 01H 04H B8H』

There into :

40H	Boot code
03H	Effective length is 3 bytes
01H	Command code of 『SetBaudRate』

04H	On behalf of 9600bps
B8H	Checksum

Checksum is the patch code of 40H+03H+01H+04H=48H

When execution correct, return packet is: 『F0H 02H 01H 0DH』

When execution fail, return packet is: 『F4H 03H 01H 1FH E9H』

2.1.7 Checksum

Following C language Check Sum calculation program is for reference:

```
unsigned char CheckSum(unsigned char *uBuff, unsigned char uBuffLen)

{

unsigned char i,uSum=0;

for(i=0;i<uBuffLen;i++)

{

uSum = uSum + uBuff[i];

}

uSum = (~uSum) + 1;

return uSum;

}
```

2.2 Control command format of serial

2.2.1 Set BaudRate

Function: to set operation baud rate for RS232 interface.

The original baud rate of RS232 interface is 9600bps, after every new program download for reader. When reader receives the command, it will reset baud rate of serial. No matter power closed or not, the baud rate will keep same to next reset.

Command code: 01H

Command parameter: 1byte BPS, value: 00H~08H， respectively on behalf of:

04H	9600bps
05H	19200bps
06H	38400bps
07H	57600bps
08H	115200bps

Command packet: 『40H 03H 01H BPS CheckSum』

Return data: if command execution correct, return data is null.

『F0H 02H 01H 0DH』

Command format with reader address:

Command code: 01H

Parameter of reader address: address

Command parameter: 1byte BPS, value: 00H~08H, respectively on behalf of:

04H	9600bps

05H	19200bps
06H	38400bps
07H	57600bps
08H	115200bps

Command packet with address: 『40H 04H 01H address BPS CheckSum』

Return data: if command execution correct, return data is null.

Return packet with address:『F0H 03H 01H address CheckSum』

2.2.2 Get Reader Version

Function: to get version number of hardware and software from reader

Command code: 02H

Command parameter: none

Command packet: 『40H 02H 02H BCH』

Return data: if command execution correct, return data is 4 bytes of version number:

Byte0	Major version of hardware
Byte1	Minor version of hardware(hardware version number is to show reader model

	number)
Byte2	Major version of software
Byte3	Minor version of software

For example: if model number of reader is Reader1102, software version number is V1.5, then return packet is:

『F0H 06H 02H 0BH 02H 01H 05H DDH』

Command format with reader address:

Command code: 02H

Parameter of reader address: address

Command parameter: none

Command packet: 『40H 03H 02H address CheckSum』

Return data: if command execution correct, return data is 4 bytes of version number:

Byte0	Major version of hardware
Byte1	Minor version of hardware(hardware version number is to show reader model number)
Byte2	Major version of software
Byte3	Minor version of software

For example: if model number of reader is Reader1102, software version number is V1.5, then return packet is:

『F0H 07H 02H address 0BH 02H 01H 05H CheckSum』

2.2.3 Set Relay

Function: to set relay status for reader

Command code: 03H

Command parameter: 1 byte

Bit0=1	No1 relay on
Bit0=0	No1 relay off
Bit1=1	No2 relay on
Bit1=0	No2 relay off

And so on.

Command packet: 『40H 03H 03H K CheckSum』

Return data: if success, return data is null.

『F0H 02H 03H 0BH』

Command format with reader address:

Command code: 03H

Parameter of reader address: address

Command parameter: 1byte

Bit0=1	No1 relay on
Bit0=0	No1 relay off
Bit1=1	No2 relay on
Bit1=0	No2 relay off

And so on.

Command packet: 『40H 04H 03H address K CheckSum』

Return data: if success, return data is null.

『F0H 03H 03H address CheckSum』

Note: Only one relay is currently supported

2.2.4 Get Relay

Function: to Get relay status for reader

Command code: 0BH

Command packet: 『40H 03H 0BH CheckSum』

Return data: if success, return data is 1 byte

『F0H 02H 0BH K CheckSum』

Bit0=1	No1 relay on
Bit0=0	No1 relay off
Bit1=1	No2 relay on
Bit1=0	No2 relay off

Command format with reader address:

Command code: 0BH

Parameter of reader address: address

Command packet: 『40H 04H 0BH address CheckSum』

Return data: if success, return data is K（1byte）

『F0H　03H　03H　address K CheckSum』

Bit0=1	No1 relay on
Bit0=0	No1 relay off
Bit1=1	No2 relay on
Bit1=0	No2 relay off

2.2.5 Set Output Power

Function: to set the output power coefficient of reader. It takes effect immediately when setting new output power coefficient for reader, and will keep same till reset, no matter power supply is off or not.

Command code:　04H

Command parameter: 1 byte of P expresses power value 0~63.

Command packet:　『40H　03H　04H　P　CheckSum』

Return data: if command execution correct, return data is null.

『F0H　02H　04H　0AH』

Command format with reader address:

Command code:　04H

Parameter of reader address: address

Command parameter: 1 byte of P expresses power value 20~33.

Command packet:　『40H　04H　04H　address　P　CheckSum』

Return data: if command execution correct, return data is null.

『F0H　03H　04H　address　CheckSum』

2.2.6 Set Frequency

Function: to set the frequency channel number for reader to transmit microwave signal. It takes effect immediately when operation frequency is set ready, and will keep same till reset, no matter power supply is off or not.

Command code: 05H

Command parameter: 2 bytes, byte1 expresses start frequency fmin, value 0~59; byte2 expresses end frequency fmax, value 0~59. If fmax>fmin, reader works by FHSS, range is fmin～fmax. If fmax=fmin, reader works in fixed frequency, frequency is fmax. Note: it's linked with countries' frequency.

Command packet: 『40H　04H　05H　fmin　fmax　CheckSum』

Return data: if command execution correct, return data is null.

『F0H　02H　05H　09H』

Command format with reader address:

Command code: 05H

Parameter of reader address:address

Command parameter: 2 bytes, byte1 expresses start frequency fmin, value 0~59; byte2 expresses end frequency fmax, value 0~59. If fmax>fmin, reader works by FHSS, range is fmin～fmax. If fmax=fmin, reader works in fixed frequency, frequency is fmax.

Command packet: 『40H　05H　05H　address　fmin　fmax　CheckSum』

Return data: if command execution correct, return data is null.

『F0H 03H 05H address CheckSum』

Chart. Relation between requency channel and transmitting frequency

Channel number	Frequency inMHz	Channel number	Frequency inMHz	Channel number	Frequency inMHz
0	865.00	20	908.50	40	918.50
1	865.50	21	909.00	41	919.00
2	866.00	22	909.50	42	919.50
3	866.50	23	910.00	43	920.00
4	867.00	24	910.50	44	920.50
5	867.50	25	911.00	45	921.00
6	868.00	26	911.50	46	921.50
7	902.00	27	912.00	47	922.00
8	902.50	28	912.50	48	922.50
9	903.00	29	913.00	49	923.00
10	903.50	30	913.50	50	923.50
11	904.00	31	914.00	51	924.00
12	904.50	32	914.50	52	924.50
13	905.00	33	915.00	53	925.00
14	905.50	34	915.50	54	925.50
15	906.00	35	916.00	55	926.00
16	906.50	36	916.50	56	926.50
17	907.00	37	917.00	57	927.00
18	907.50	38	917.50	58	927.50
19	908.00	39	918.00	59	928.00

2.2.7 Read Param

Function: to read the operation parameter in reader written by last command.

Command code: 06H

Command parameter: none

Command packet: 『40H 02H 06H B8H』

Return data: if command execution correct, return data is 32 bytes PAM from setting command.

『F0H 22H 06H PAM CheckSum』

Command format with reader address:

Command code: 06H

Parameter of reader address: address

Command parameter: none

Command packet: 『40H 03H 06H address CheckSum』

Return data: if command execution correct, return data is 32 bytes PAM from setting command.

『F0H 23H 06H address PAM CheckSum』

2.2.8 Set Param

Function: to set basic parameter of reader, baud rate of serial, transmit power, RF power output and so on.

Command code: 09H

Command parameter: 32 bytes PAM

Command packet: 『40H 22H 09H PAM CheckSum』

Return data: if command execution correct, return data is null. If success, return data of ReadParam is 32 bytes parameters with following sequence.

『F0H 02H 09H 05H』

Command format with reader address:

Command code: 09H

Parameter of reader address: address

Command parameter: 32 bytes PAM

Command packet: 『40H 23H 09H address PAM CheckSum』

Return data: if command execution correct, return data is null. If success, return data of ReadParam is 32 bytes parameters with following sequence.

『F0H 03H 09H address CheckSum』

32 bytes of parameter(1 parameter 1 byte):

Byte number	Definition
1	Baud rate of serial, value: 00H~08H，
2	Transmit power, value: 20～33dbm.
3	Start point of transmitting microwave signal frequency, value(default 7): 0~59
4	End point of transmitting microwave signal frequency, value(default 59): 0~59
5	Null
6	Word mode of reader: 0-Auto， 1-Command
7	RS485 address of reader: 0 and 255 is broadcast address
8	Max tags of once reading
9	Tag type: 01H—ISO18000-6B, 02H—EPCC1, 04H—EPCC1G2
10	Tag reading duration time: RF emission duration time, only valid for EM tag. 0—10ms, 1—20ms, 2—30ms, 3—40ms.
11	Reading times M： reader will execute the command for M times when receiving reading command from host pc.
12	1:enable buzzer 0: disable buzzer
13	Reader IP address1
14	Reader IP address2
15	Reader IP address3
16	Reader IP address4
17	Reader port –high

Byte number	Definition
18	Reader port –low
19	Reader mask1
20	Reader mask2
21	Reader mask3
22	Reader mask4
23	Reader address gateway1
24	Reader address gateway2
25	Reader address gateway3
26	Reader address gateway4
27	Reader MAC1
28	Reader MAC2
29	Reader MAC3
30	Reader MAC4
31	Reader MAC5
32	Reader MAC6

2.2.9 Read Auto Param

Function: to read the auto work parameter in reader written by last command.

Command code: 14H

Command parameter: none

Command packet: 『40H 02H 14H 6DH』

Return data: if success, return data is 32 bytes PAM from setting command.

『F0H 24H 14H PAR Check』

Command format with reader address:

Command code: 14H

Parameter of reader address: address

Command parameter: none

Command packet: 『40H 03H 14H Address Check』

Return data: if success, return data is 32 bytes PAM from setting command.

『F0H 24H 14H Address PAR Check』

2.2.10 Set Auto Param

Function: to set basic parameter for reader in auto mode, baud rate of serial, transmit power, RF power output and so on.

Command code: 13H

Command parameter: 32 bytes PAM

Command packet: 『40H 26H 13H PAM Check』

Return data: if command execution correct, return data is null. If success, return data of Read Auto Param is 32 bytes parameters with following sequence.

『F0H 02H 13H FBH』

Command format with reader address:

Command code: 13H

Parameter of reader address: address

Command parameter: 32 bytes PAM

Command packet: 『40H 27H 13H Address PAM Check』

Return data: if command execution correct, return data is null. If success, return data of ReadParam is 32 bytes parameters with following sequence.

『F0H 03H 13H Address Check』

32bytes parameters (1 parameter 1 byte):

Byte number	Definition
1	Tag reading mode：0-timing，1-trigger
2	Tag storage time：unit：s. default 1. High.
3	Tag storage time：unit：s. default 1. Low.
4	0-10ms, 1-20ms, 2-30ms, 3-50ms, 4-100ms. Default 2. Atumatically read tag once at intervals.
5	Tag storage quantity：default 1. The quantity of read tag ID stored in reader memory. High.
6	Tag storage quantity：default 1. The quantity of read tag ID stored in reader memory. Low.
7	Data output format：0-terse，1-standard，2-XML format. Default0.
8	Output interface: 0-RS232, 1-RS485, 2-RJ45. Default 0. 3- Wiegand26, 4- Wiegand34.
9	Wiegand output pulse width, default40.
10	Wiegand output pulse interval, default200.
11	Set start bit of output tag ID, value 0～4. Default 0. (Wiegand)
12	Set storage address of ID in tag(default0): 0-tag ID number 1-usder defined number(Wiegand)
13	Notify interval：unit：s. Default 1. Automatically notify host pc once at intervals. Default 120s, 1～255.
14	Notify condition：default1. 0-notify now，1-timing，2-add，3-remove，4-change
15	Output port of notify (if no reading for long time, send EPC number with length 0 to output port) 0—no use 1—use time depends on ”notify interval”.
16	Select antenna. 1-ant1,2-ant2,4-ant3,8-ant4
17	Set trigger mode(default0): 0-low level 1-high level
18	Host PC IP address1
19	Host PC IP address2
20	Host PC IP address3
21	Host PC IP address4
22	Host PC port -high
23	Host PC port -low
24	Null
25	Null
26	Null
27	Null
28	Null

Byte number	Definition
29	Null
30	0-no alarm，1-alarm. Detect alarm or not in timing and trigger mode.
31	null
32	Control relay or not in Auto state. 0- no control 1- control

2.2.11 Select Antenna

Function: to select from which antenna to transmit-receive signal.

Command code: 0AH

Command parameter: 1 byte of selected antenna No.

1	Select No.1 antenna
2	Select No.2 antenna
4	Select No.3 antenna
8	Select No.4 antenna

Command packet: 〖40H   03H   0AH   No.   CheckSum〗

Return data: if command execution correct, return data is null.

〖F0H   02H   0AH   04H〗

Command format with reader address:

Command code: 0AH

Parameter of reader address: address

Command parameter: 1 byte of selected antenna No

1	Select No.1 antenna
2	Select No.2 antenna
4	Select No.3 antenna
8	Select No.4 antenna

Command packet: 『40H 04H 0AH address No. CheckSum』

Return data: if command execution correct, return data is null.

『F0H 03H 0AH address CheckSum』

2.2.12 Restore factory settings

Function: The parameters of the Reader device back to factory

Command code: 0EH

Command parameter: none

Command packet: 『40H 02H 0DH Check』

Return data: if success, return data is null.

『F0H 02H 0DH Check』

2.2.13 Reboot

Function: reboot reader, which equivalent to power on again after power off.

Command code: 0EH

Command parameter: none

Command packet: 『40H 02H 0EH B0H』

Return data: if success, return data is null.

『F0H 02H 0EH 00H』

Command format with reader address:

Command code: 0EH

Parameter of reader address: address

Command parameter: none

Command packet: 〖40H 03H 0EH address CheckSum〗

Return data: if success, return data is null.

〖F0H 03H 0EH address CheckSum〗

2.2.14 Start/ stop automode

Function: Set the operation mode of the Reader.

Command code: 0FH

Command parameter: 1 byte, automode is the working mode of Reader.

Command packet: 〖40H 03H 0FH Automode Check〗

Automode is 0, mean stop automode

Automode is 1, mean start automode

Return data: if success, return data is null.

〖F0H 02H 0FH 00H〗

2.2.15 Clear Memory

Function: Remove the cache of tag data in Reader.

Command code: 10H

Command parameter: Null

Command packet: 『40H 02H 10H AEH』

Return data: if success, return data is null.

『F0H 02H 10H FEH』

2.2.16 Set Reader Time

Function: to set time for reader.

Command code: 11H

Command parameter: 6bytes:yy/mm/dd/hour/minute/second

Command packet: 『40H 08H 11H yy mm dd hh ff ss CheckSum』

Return data: if success, return data is null.

『F0H 02H 11H FDH』

Command format with reader address:

Command code: 11H

Parameter of reader address:address

Command parameter: none

Command packet: 『40H 09H 11H address yy mm dd hh ff ss CheckSum』

Return data: if success, return data is null.

『F0H 03H 11H address CheckSum』

2.2.17 Get Reader Time

Function: to get time from reader.

Command code: 12H

Command parameter: none

Command packet: 『40H 02H 12H ACH』

Return data: if success, return data is yy/mm/dd/hour/minute/second.

『F0H 08H 12H yy mm dd hh ff ss CheckSum』

Command format with reader address:

Command code: 12H

Parameter of reader address: address

Command parameter: none

Command packet: 『40H 03H 12H address CheckSum』

Return data: if success, return data is yy/mm/dd/hour/minute/second.

『F0H 09H 12H address yy mm dd hh ff ss CheckSum』

2.2.18 Set Report Filter

Function: read tag chosen from setted filter can be put into output list.

Command code: 15H

Command parameter1: 2 bytes of mask address ADDR, high digit before, low digit after. That’s bit address.

Command parameter2: 2 bytes of mask length LEN, high digit before, low digit after. That’s bit length.

Command parameter3: M bytes of data, high digit before, low digit after. If

LEN/8 is integer, then M=LEN/8；if LEN/8 is not integer, then M=LEN/8＋1, lastbyte data in high-order-position, low-order filling zero.

Command packet: 『40H 4+M 15H ADDR LEN M CheckSum』

ReturnData: if success, return data will be null.

『F0H 02H 15H F9H』

Note:

1. When LEN＝0, filter unavailable, without command parameter3.

Command packet:『40H 04H 15H 00 A7H』

2. Filter object:

ISO18000-6B	64 bits ID
EPCC1	EPC No.
ISO18000-6C	EPC No.
EM Tag	64 bits ID

2.2.19 Get Report Filter

Function: to get parameter of filter.

Command code: 16H

Command parameter: No

Command packet:〖40H 02H 16H A8H〗

ReturnData: if success, return data will be (2 bytes of mask ADDR＋2 bytes of mask LEN＋M bytes of data). If LEN/8 is integer, then M=LEN/8；if LEN/8 is not integer, then M=LEN/8＋1, last byte data in high-order-position, low-order filling zero.

〖F0H 4+M 16H ADDR LEN M CheckSum〗

2.2.20 Set Reader Network Address

Function: to set network address for reader.

Command code: 30H

Command parameter: 14bytes parameter.

(IP(4Bytes)+PORT(2Bytes)+MASK(4Bytes)+Gateway(4Bytes)).

Command packet: 〖40H 10H 30H IP PORT MASK Gateway CheckSum〗

Return data: if success, return data is null.

〖F0H 02H 30H DEH〗

2.2.21 Get Reader Network Address

Function: to get network address from reader.

Command code: 31H

Command parameter: none

Command packet: 〖40H 02H 31H 8DH〗

Return data: if success, return data is (IP(4Bytes)+PORT(2Bytes)+MASK(4Bytes)+Gateway (4Bytes)).

『F0H 10H 31H IP PORT MASK Gateway CheckSum』

2.2.22 SetReader MAC

Function: to set network MAC for reader.

Command code: 32H

Command parameter: 6bytes parameter MAC

Command packet: 『40H 08H 32H MAC CheckSum』

Return data: if success, return data is null.

『F0H 02H 32H DBH』

2.2.23 Get Reader MAC

Function: to get network MAC from reader.

Command code: 33H

Command parameter: none

Command packet: 『40H 02H 33H 8B』

Return data: if success, return data is 6bytes MAC.

『F0H 08H 33H MAC CheckSum』

2.2.24 Report Now

Function: to send all reserved tag information to PC when reader receive the command.

Command code: 54H

Command parameter: no

Command packet: 『40H 02H 54H 6AH』

ReturnData: if success, return data will be null.

『F0H 02H 54H BAH』

Following is some data packets. Each data packet with 1 tag information and each tag information is fixed as 120 bytes. Format of Data packet in appendix A.

2.2.25 Get Tag Info.

Function: reader will send the Tag information reader memory stored to PC immedicately after receiving this command.

Command code: 57H

Command parameter1:2 bytes of initial record address ADDR,higher bit first,lower bit last.

Command parameter2；1byte of record size LEN(<=5)

Command packet: 『40H 05H 57H ADDR LEN CheckSum』

ReturnData: if success, return packet definition as following:

『F0H 4+N*(17+L) 57H N N *RECORD CheckSum』

N(<=5) is the record number reader return. Record formate is HEX: 6bytes ofstart time＋6bytes of end time＋2bytes of reading times＋1byte of antenna No.＋1byte of tag type＋1byte of EPC Length＋L byte of EPC. Records range fromsmall to large according to EPC value.

If record parameter appointed is not exist, return packet definition as

following:

『F0H 04H 57H 00H 00H B5H』

2.2.26 GetReaderID

Function: to get reader ID No.

Command code: 8CH

Command parameter: Null

Command packet: 『40H 02H 8CH CheckSum』

ReturnData: if success, return data will be 6 bytes of ID.

『F0H 0AH 8CH ID CheckSum』

2.3 Command format of serial read-write tag

2.3.1 Read-write ISO18000-6B command format

For electronic tag, internal storage capacity is 2048bits, which is devided into 256bytes. An address is for each byte, corresponding to 0～255.

Thereinto:

- Address 0～7, 8 words(64bits): tag ID number. Fixed before product leaves factory, can't be modified.

- Address 8～223: user information storage area, can be distributed at discretion.

- Address 224～255: write protection byte.

2.3.1.1 List Tag ID

Function: to list readable tag ID in antenna radiation field.

Command code: FEH

Command parameter: none

Command packet: 『40H 02H FEH C0H』

Return data: if success, byte count of return data = number of all listed tags M(1byte)+ M*data of (tag length sent out L(<=8，1byte)+ 8(ID)).

『F0H 3+M*（L+8）FEH M M*（L+8）CheckSum』

Command format with reader address:

Command code: FEH

Parameter of reader address:address

Command parameter: none

Command packet: 『40H 03H FEH address CheckSum』

Return data: if success, byte count of return data = number of all listed tags M(1byte)+M* data of (tag Length sent out L(<=8，1byte)*8(ID)).

『F0H 4+M*（L+8） FEH address M M*(L+8) CheckSum』

2.3.1.2 Get ID List

Function: to get listed tag ID by rfs_ListID command from reader memory.

Command code:FDH

Command parameter: 2bytes, the first byte is start number ADDR, the second byte is tag count M (<=8,1byte).

Command packet: 『40H 04H FDH ADDR M CheckSum』

Return data: if success, byte count of return data=M+M*data of (tag length L*8(ID)).

『F0H 2+M*（L+8） FDH M M*(L+8) CheckSum』

Command format with reader address:

Command code: FDH

Parameter of reader address: address

Command parameter: 2bytes, the first byte is start number ADDR, the second byte is tag count M(<=8).

Command packet: 『40H 05H FDH address ADDR M CheckSum』

Return data: if success, byte count of return data=M+M*data of (tag length L*8(ID)).

『F0H 3+M*(L+8) FDH address M M*(L+8) CheckSum』

2.3.1.3 List Selected ID

Function: to list readable tag ID in antenna radiation field based on following parameter condition.

Command code: FBH

Command parameter1: 1byte is the condition of selected tag.

00	Equal to
01	Unequal to
02	Greater than
03	Less than

Command paramter2: 1byte is start address of tag data ADDR, value 0～223.

Command parameter3: 1byte is data MASK. Each bit in the byte is corresponding to a comparative byte.

0	The byte not for comparison
1	The byte for comparison

Command parameter4: 8bytes is data for comparison.

Command packet: 『40H 0DH FBH SEL ADDR MASK DATA CheckSum』

Return data: if success, byte count of return data= number of all listed tags M(1byte)+ M *data of (tag length L (1byte)*8(ID)).

『F0H 3+M*(L+8) FBH M M*（L+8）CheckSum』

Command format with reader address:

Command code:FBH

Parameter of reader address:address

Command parameter1: 1byte is the condition of selected tag.

00	Equal to
01	Unequal to
02	Greater than
03	Less than

Command paramter2: 1byte is start address of tag data ADDR, value 0～223.

Command parameter3: 1byte is data MASK. Each bit in the byte is corresponding to a comparative byte.

0	The byte not for comparison
1	The byte for comparison

Command parameter4: 8bytes is data for comparison.

Command packet: 『40H 0EH FBH address SEL ADDR MASK DATA CheckSum』

Return data: if success, byte count of return data= number of all listed tags M(1byte)+ M *data of (tag length L (1byte)*8(ID)).

『F0H 4+M*(L+8) FBH address M M*(L+8) CheckSum』

2.3.1.4 Read Byte Block

Function: to read a block of data starting from appointed address of appointed tag. Memory capacity of ISO18000-6B tag is 2048bits(256bytes). Readable byte address for user is 0～223. Length of data block takes byte as unit. Stipulates 32bytes can be read at most each time.

Command code: F6H

Command parameter: 8bytes ID(subject to the sequence of tag ID number); 1byte start address(aa), value 0~223; 1byte length of block(nn), value 1~32.

Command packet: 『40H 0CH F6H id aa nn CheckSum』

Return packet: if success, return data is nn bytes of data.

『F0H nn+2 F6H xx …… xx CheckSum』

Command format with reader address:

Command code: F6H

Parameter of reader address:address

Command parameter: 8 bytes ID(subject to the sequence of tag ID number); 1byte start address(aa), value 0~223; 1byte length of block(nn), value 1~32.

Command packet: 『40H 0DH F6H address id aa nn CheckSum』

Return packet: if success, return data is nn bytes of data.

『F0H nn+3 F6H address xx …… xx CheckSum』

2.3.1.5 Write Byte Block

Function: to write data into the appointed address unit of appointed tag. Data length to write takes byte as unit, and 4bytes can be written in at most once. Writable byte address for user is 8～223.

Command code: F5H

Command parameter: 8 bytes ID(subject to the sequence of tag ID number); 1byte start address(aa), value 8~223; 1byte length of block(nn), value 1~4; nn bytes written data.

Command packet: 『40H 12+nn F5H id aa nn xx --- xx CheckSum』

Return packet: if success, return data is null.

『F0H 02H F5H 19H』

Command format with reader address:

Command code: F5H

Parameter of reader address: address

Command parameter: 8 bytes ID(subject to the sequence of tag ID number); 1byte start address(aa), value 8~223; 1byte length of block(nn), value 1~4; nn bytes written data.

Command packet: 『40H 13+nn F5H address id aa nn xx --- xx CheckSum』

Return packet: if success, return data is null.

『F0H 03H F5H address CheckSum』

2.3.1.6 Set Write Protect

Function: to set write protection for the appointed addess unit of appointed tag.

Command code:F4H

Command parameter: 8 bytes ID(subject to the sequence of tag ID number); 1byte start address(aa), value 8~223.

Command packet: 『40H 0BH F4H ID aa CheckSum』

Return data: if success, boot code of return packet is F0H, data part is null.

『F0H 02H F4H 1AH』

Command format with reader address:

Command code: F4H

Parameter of reader address: address

Command parameter: 8 bytes ID(subject to the sequence of tag ID number); 1byte start address(aa), value 8~223.

Command packet: 『40H 0CH F4H address ID aa CheckSum』

Return data: if success, boot code of return packet is F0H, data part is null.

『F0H 03H F4H address CheckSum』

2.3.1.7 Read Write Protect

Function: to read the appointed address unit of appointed tag if with write protect or not.

Command code: F3H

Command parameter: 8 bytes ID(subject to the sequence of tag ID number); 1byte start address(aa), value 0~223.

Command packet: 『40H 0BH F3H ID aa CheckSum』

Return data: if success, boot code of return packet is F0H, data part is 1 byte.

0	Without protection, 『F0H 03H F3H 00H 1AH』
1	With protection, 『F0H 03H F3H 01H 19H』

Command format with reader address:

Command code:F3H

Parameter of reader address:address

Command parameter: 8 bytes ID(subject to the sequence of tag ID number) ; 1byte start address(aa), value 0~223.

Command packet: 『40H 0CH F3H address ID aa CheckSum』

Return data: if success, boot code of return packet is F0H, data part is 1 byte.

0	Without protection, 『F0H 04H F3H address 00H CheckSum』
1	With protection, 『F0H 04H F3H address 01H CheckSum』

2.3.1.8 Slow Write A Byte

Function: to write data into the appointed address unit of appointed tag. Length of written data takes byte as unit, and 4bytes can be written in at most once. Writable byte address for user is 8～223.

Command code: F2H

Command parameter: 8 bytes ID(subject to the sequence of tag ID number); 1byte start address(aa), value 8~223; 1byte length of block(nn), value 1~4; nn bytes written data.

Command packet: 『40H 12+nn F2H id aa nn xx --- xx CheckSum』

Return packet: if success, return data is null.

『F0H 02H F2H 1CH』

Note: the command writes data into tag byte by byte, slow in speed. Only used for tags not support previous write command.

Command format with reader address:

Command code: F2H

Parameter with reader address: address

Command parameter: 8 bytes ID(subject to the sequence of tag ID number); 1byte start address(aa), value 8~223; 1byte length of block(nn), value 1~4; nn bytes written data.

Command packet: 『40H 13+nn F2H address id aa nn xx --- xx CheckSum』

Return data: if success, return data is null.

『F0H 03H F2H address CheckSum』

2.3.2 Read write ISO18000-6C command format

Storage unit of ISO18000-6C is devided into 4 memory banks:

A. EPC memory bank: to store EPC number. It can store 96bits EPC number at most presently. Readable and writable.

B. TID memory bank: to store ID number set by tag manufacturer. There are 32bits and 64bits ID presently. Readable but non-writable.

C. User memory bank: different from various manufacturer.

D. Password memory bank: with 32Bits access password and 32Bits kill password. Readable and writable.

2.3.2.1 List Tag ID

Function: to identify the readable tag ID in antenna radiation field based on MASK condtion.

Command code: EEH

Command parameter1: 1byte mem, select memory bank.

0	Password
1	EPC
2	TID
3	User

Command parameter2: 2bytes, start address of MASK (unit: bit).

Command parameter3: 1byte, the length of MASK (unit: bit).

Command parameter4: m bytes, Mask. If LEN％8=0, then m=LEN/8. If LEN％8≠0, then m=⌊LEN/8⌋+1.

Command packet: 『40H m+6 EEH mem addr LEN Mask CheckSum』

Return data: if success, byte count of return data= number of all listed tags M(1byte)+ data of (tag number sent out L(<=8)*L(digit count of EPC+EPC)).

Note: LEN=0, to identify all readable tag ID in antenna radiation field.

Digit count of EPC:00H-0Word，01H-1Word，02H-2Word，……，FFH-256Word

『F0H 3+L*N EEH M L*N CheckSum』

Command format with reader address:

Command code: EEH

Parameter of reader address: address

Command parameter1: 1byte mem, select memory bank.

0	Password

1	EPC
2	TID
3	User

Command parameter2: 2bytes addr, start address of MASK (unit: bit).

Command parameter3: 1byte LEN, the length of MASK (unit: bit).

Command parameter4: m bytes, Mask. If LEN％8=0, then m=LEN/8. If LEN％8≠0, then m=⌊LEN/8⌋+1.

Command packet: 『40H m+7 EEH address mem addr LEN Mask CheckSum』

Return data: if success, byte count of return data= number of all listed tags M(1byte)+ data of (tag number sent out L(<=8)*L(digit count of EPC+EPC)).

Note: LEN=0, to identify all readable tag ID in antenna radiation field.

Digit count of EPC:00H-0Word，01H-1Word，02H-2Word，……，FFH-256Word

『F0H 4+L*N EEH address M L*N CheckSum』

2.3.2.2 Get ID List

Function: to get listed tag ID by rfs_ListID command from reader memory.

Command code:EDH

Command parameter: 2bytes, the first byte is start number NO, the second byte is tag count L(<=8).

Command packet: 『40H 04H EDH no m CheckSum』

Return data: if success, byte count of return data=data of (1byte tag count M*L Bytes(digit count of EPC+EPC)).

『F0H 2+L*8 EDH L*M CheckSum』

Command format with reader address:

Command code:EDH

Parameter of reader address:address

Command parameter:2bytes, the first byte is start number NO, the second byte is tag count L(<=8).

Command packet: 『40H 05H EDH address no m CheckSum』

Return data: if success, byte count of return data=data of (1byte tag count M*L Bytes(digit count of EPC+EPC)).

『F0H 3+L*8 EDH address L*M CheckSum』

2.3.2.3 Read Word Block

Function: to read a block of data starting from appointed address of appointed tag. Length of data block takes word(16bits) as unit.

Command code: ECH

Command parameter1: 1byte L, digit count of EPC, to show the word count of EPC number.

Command parameter2: L*2bytes EPC number, to show which tag data to read.

Command parameter3: 1byte mem, select memory bank.

0	Password
1	EPC
2	TID
3	User

Command parameter4: 1byte addr, start address(unit:Word).

Command parameter5: 1byte len, length of data(unit:Word).

Command parameter6: 4bytes, AccessPassword.

Command packet: 『40H 10+L*2 ECH L EPC mem addr len AccessPassword CheckSum』

Return packet: if success, return data is len*2bytes data xx.

『F0H len*2+2 ECH xx …… xx CheckSum』

Note: AccessPassword is workable when password memory bank is locked.

Command format with reader address:

Command code: ECH

Parameter of reader address:address

Command parameter1: 1byte L, digit count of EPC, to show the word count of EPC number.

Command parameter2: L*2bytes EPC number, to show which tag data to read.

Command parameter3: 1byte mem, select memory bank.

0	Password
1	EPC
2	TID
3	User

Command parameter4: 1byte addr, start address(unit:Word).

Command parameter5: 1byte len, length of data(unit:Word).

Command parameter6: 4bytes, AccessPassword.

Command packet: 『40H 11+L*2 ECH address L EPC mem addr len AccessPassword CheckSum』

Return packet: if success, return data is len*2bytes data xx.

『F0H len*2+3 ECH address xx …… xx CheckSum』

2.3.2.4 Write Word Block

Function: write data into the appointed address unit in appointed memory bank of appointed tag. Length of written data takes word as unit.

Command code: EBH

Command parameter1: 1byte L, digit count of EPC, to show the word count of EPC number.

Command parameter2: L*2bytes EPC number, to show which tag data to write.

Command parameter3: 1byte mem, select memory bank.

0	Password
1	EPC
2	TID
3	User

Command parameter4: 1byte addr, start address(unit:Word).

Command parameter5: 1byte len, length of data(unit:Word).

Command parameter6: len*2bytes written data.

Command parameter7: 4bytes, AccessPassword.

Command packet: 『40H 10+L*2+len*2 EBH L EPC mem addr len data AccessPassword CheckSum』

Return packet: if success, return data is null.

『F0H 02H EBH 23H』

Note: AccessPassword is workable only when memory bank is locked. When unlocked, writable without password. When permanently locked, password is useless.

Command format with reader address:

Command code: EBH

Parameter of reader address:address

Command parameter1: 1byte L, digit count of EPC, to show the word count of EPC number.

Command parameter2: L*2bytes EPC number, to show which tag data to write.

Command parameter3: 1byte mem, select memory bank.

0	Password
1	EPC
2	TID
3	User

Command parameter4: 1byte addr, start address(unit:Word).

Command parameter5: 1byte len, length of data(unit:Word).

Command parameter6: len*2bytes written data.

Command parameter7: 4bytes, AccessPassword.

Command packet: 『40H 11+L*2+len*2 EBH address L EPC mem addr len data AccessPassword CheckSum』

Return packet: if success, return data is null.

『F0H 03H EBH address CheckSum』

2.3.2.5 Kill Tag

Function: Permanentkill tag

Command code: E8H

Command parameter1: 1byte L, digit count of EPC, to show the word count of EPC number.

Command parameter2: L*2bytes EPC number, to show which tag to set read-write protect.

Command parameter3: kill password , 4bytes mem

Command packet: 『40H L*2+7 E8H L EPC KillPassword Check』

Command packet with reader address: 『 40H L*2+8 E8H Address L EPC KillPassword Check』

Head	Len	Cmd	Address	Data	Check
40H	L*2+7/L*2+8	E8H		L EPC KillPassword	1Byte

Specific command package parameters definition to see the table below:

L	1byte，digit count of EPC (Unit:Word)
EPC	L*2 bytes EPC
KillPassword	4bytes

ReturnData: if success, return data will be null.

『F0H 02H E8H Check』

2.3.2.6 Set Lock

Function: to set write protect for the appointed memory bank of appointed tag.

Command code:EAH

Command parameter1: 1byte L, digit count of EPC, to show the word count of EPC number.

Command parameter2: L*2bytes EPC number, to show which tag to set read-write protect.

Command parameter3: 1byte mem, select memory bank.

0	Kill Password
1	Access Password
2	EPC number
3	ID number in TID
4	User

Command parameter4: 1byte Lock, control word.

0	Writable
1	Writable permanently
2	Writable in secured state
3	Never writable
4	Readable and writable
5	Readable and writable permanently
6	Readable and writable in secured state
7	Never readable and writable

0～3 only apply to EPC, TID and User memory bank; 4～7 only apply to Kill Password and Access Password.

Command parameter5: 4bytes AccessPassword.

Command packet: 『40H 9+L*2 EAH L EPC mem Lock AccessPassword

CheckSum』

Return data: if success, boot code of return packet is F0H, return data is null.

『F0H　02H　EAH　24H』

Command format with reader address:

Command code:EAH

Parameter of reader address:address

Command parameter1: 1byte L, digit count of EPC, to show the word count of EPC number.

Command parameter2: L*2bytes EPC number, to show which tag to set read-write protect.

Command parameter3: 1byte mem, select memory bank.

0	Kill Password
1	Access Password
2	EPC number
3	ID number in TID
4	User

Command parameter4: 1byte Lock, control word.

0	Writable
1	Writable permanently
2	Writable in secured state
3	Never writable
4	Readable and writable
5	Readable and writable permanently
6	Readable and writable in secured state
7	Never readable and writable

0～3 only apply to EPC, TID and User memory bank; 4～7 only apply to Kill Password and

Access Password.

Command parameter5: 4bytes AccessPassword.

Command packet: 『40H 10+L*2 EAH address L EPC mem Lock AccessPassword CheckSum』

Return data: if success, boot code of return packet is F0H, return data is null.

『F0H 03H EAH address CheckSum』

2.3.2.7 Write EPC

Function: to write EPC data into tag's EPC bank. Word as unit of written data length.

Command code: E7H

Command parameter 1: 1 byte EPC digit L, indicate number of Word of EPC No.;

Command parameter 2: L*2 bytes EPC number;

Command parameter 3: 4 bytes AccessPassword;

Command packet: 『40H 7+L*2 E7H L EPC AccessPassword CheckSum』

ReturnData: if success, return data will be null.

『F0H 02H E7H 27H』

Note: AccessPassword is available for MemBank in password locked status only. If MemBank unlocked, write without password; if MemBank locked forever, password is useless.

2.4 Collection of operation command

2.4.1 ISO18000-6C command

Serial Number	Command	Function
1	EEH	To identify readable tag ID in antenna radiation field based on mask condition
2	EDH	To get listed tag ID by rfs_ListTagID command from reader memory
3	ECH	To read a block of data starting from the appointed address in appointed memory bank of appointed tag
4	EBH	To write data into the appointed address unit in appointed memory bank of appointed tag
5	EAH	To set write protect for appointed memory bank of appointed tag
6	E7H	To write EPC number into EPC memory bank of tag
7		
8		
9		
10		

2.4.2 ISO18000-6B command

Serial number	Command	Function
1	FEH	To list readable tag ID in antenna radiation field
2	FDH	To get listed tag ID by rfs_ListTagID command from reader memory
3	FBH	To list readable tag ID in antenna radiation field based on following parameter condition
4	F6H	To read a block of data starting from the appointed address of appointed tag
5	F5H	To write data into the appointed address unit of appointed tag
6	F4H	To set write protect for the appointed address unit of appointed tag

Serial number	Command	Function
7	F3H	To read if the appointed address unit of appointed tag is set write protect
8	F2H	To write data into the appointed address unit of appointed tag

2.4.3 Other command

Serial number	Command	Function
1	01H	To set operation baud rate for RS232 interface
2	02H	To get hardware and software version number of reader
3	03H	To set relay state for reader
4	04H	To set transmitting power coefficient for reader
5	05H	To set frequency channel number of transmitting microwave signal for reader
6	06H	To read operation parameter written by last command from reader
7	09H	To set basic operation parameter: baud rate, transmitting frequency, RF power output and so on
8	0AH	To select which antenna to transmit and receive signal
9	0EH	To reboot reader
10	10H	Delete all tag records stored in reader
11	11H	To set time for reader
12	12H	To get time from reader
13	13H	To set the auto work parameter in reader
14	14H	To read the auto work parameter in reader
15	15H	To set parameter of filter.
16	16H	To get parameter of filter.
17	30H	To set network addess for reader
18	31H	To get network address in reader
19	32H	To set network MAC for reader
20	33H	To get network MAC in reader
21	54H	To send all reserved tag information to PC
22	57H	Get Tag records stored in reader

2.5 Electronic tag storage area and notes

（1） Memory unit of ISO18000-6C tag is devided into 4 areas:

◆ EPC memory bank: to store EPC number. It can store 96bits EPC number at most presently. Readable and writable.

◆ TID memory bank: to store ID number set by tag manufacturer. There are 32bits and 64bits ID presently. Readable but non-writable.

◆ User memory bank: different from various manufacturer. G2 tag from Impinj has no user memory bank, while NXP with 96bits. Readable and writable.

◆ Password memory bank: with 32Bits access password and 32Bits kill password. Readable and writable.

ISO18000-6C tag can set different protection mode for different memory bank, 4 kinds of protection modes for each memory bank:

◆ EPC, TID and User memory bank of ISO18000-6C tag

EPC, TID and User memory bank of ISO18000-6C tag with write protect function but without read protect:

Writable from any state—can be written without access password, and can be set to writable from secured state or permanently writable or never writable later;

Permanently Writable—can be written without access password, but can't be set to writable from secured state or never writable later;

Writable from secured state—can be written with access password only, and can be set to never writable or writable from any state or permanently writable later;

Never Writable—can't be written even if with access password.

◆ Password memory bank of ISO18000-6C tag

Password memory bank of ISO18000-6C tag can be set read protect and write protect. The read and write protect state won’t influence the usage of password.

Readable and Writable from any state—can be read and written without access password, and can be set to readable and writable from secured state or permanently readable and writable or never readable and writable later;

Permanently Readable and Writable—can be read and written without access password, but can’t be set to readable and writable from secured state or never readable and writable later;

Readable and Writable from secured state—can read and modify password with access password only, and can be set to permanently readable and writable or readable and writable from any state or never readable and writable later;

Never Readable and Writable—can’t be read or write even if with access password, can’t read or modify password forever.

Note: to set read and write protect for tag, must know access password of tag in advance.

（2） Memory unit of ISO-18000-6B is devided into 2 memory banks. Internal storage capacity is 2048bits, which is devided into 256bytes. An address is for each byte, corresponding to 0～255.

Thereinto:

➢ Address 0～7, 8 words(64bits): tag ID number. Fixed before product leaves factory, can’t be modified.

➢ Address 8～223: user information storage area, can be distributed at discretion. Can be rewrited, and can be locked. Once locked, can’t be rewrited again, and can’t be unlocked.

➢ Address 224～255: write protection byte.

3. SDK software development

3.1 SDK compose

SDK is provided in the package of SL144, mainly including:

A. Reader1400dll.dll file —— dynamic link library

B. Reader1400.lib file —— static link library

C. Reader1400API.h file —— statement file of API function

D. SDK catalogue —— including example program for learning API function application

3.2 Design introduction

3.2.1 Basic constant and figure

3.2.1.1 Constant definition

Description	Introduction
#define ID_MAX_SIZE_64BIT 8	//tag ID number is 64bit
#define ID_MAX_SIZE_96BIT 13	//tag ID number is 128bit
#define MAX_LABELS 100	// no more than 100 tags for once read and write operation

3.2.1.2 API function return code

#define _OK	0x00	// operation success
//error message for communication		
#define _init_rs232_err	0x81	//communication interface initialization fail
#define _no_scanner	0x82	//can not find reader
#define _comm_error	0x83	//communication data error
#define _baudrate_error	0x84	//set baud rate error

// error message returned from reader		
#define _no_antenna	0x01	//antenna connection fail
#define _no_label	0x02	//detect no tag
#define _invalid_label	0x03	//illegal tag
#define _less_power	0x04	//read-write power not enough
#define _write_prot_error	0x05	//the memory bank in write protection
#define _check_sum_error	0x06	//checksum error
#define _parameter_error	0x07	//parameter error
#define _memory_error	0x08	//the memory bank nonexistence
#define _password_error	0x09	//password error
#define _killpassword_error	0x0a	//kill password of G2 tag all zero
#define _nonlicet_command	0x0b	//illegal command
#define _nonlicet_user	0x0c	//illegal user with unmatched password
#define _invalid_command	0x1e	//invalid command, such as command with wrong parameter
#define _other_error	0x1f	//unknown command
//function input error		
#define _no_cardID_input	0x20	//other error

3.2.1.3 Data type definition

```
typedef USHORT apiReturn;      // type of function return value
```

it will return a value of apiReturn type after all API functions executed. It can judge if the function execution is successful. If fail, what’s the failure reason and etc.

3.2.1.4 Parameter figure in reader

```
typedef struct tagReaderBasicParam

{

    BYTE    BaudRate;                //(1)baud rate of serial, value：00H~08H.

    BYTE    Power;                   //(2)RF power output, value：20~30dBm.(0-63)

    BYTE    Min_Frequence;        //(3)start point of transmitting microwave signal frequency,
```

value： 1~63.

BYTE Max_Frequence; //(4)end point of transmitting microwave signal frequency, value： 1~63.

BYTE Reserve5; //(5)reserve, changed into modulation depth later.

BYTE WorkMode; //(6)work mode of reader： 0-Auto, 1-Command

BYTE ReaderAddress; //(7)RS485 address:0--255

BYTE NumofCard; //(8)max tags of once reading.

BYTE TagType; //(9)type of tag：01H—ISO18000-6B，02H—EPCC1，04H—EPCC1G2， 08H—EM4442。

BYTE ReadDuration; //(10)tag reading duration time：RF emission duration time is only effective for EM tag; 0—10ms， 1—20ms， 2—30ms， 3—40ms.

BYTE ReadTimes; //(11)read times M： reader will execute the command for M times when receiving reading command from host computer.

BYTE EnableBuzzer; //(12)1:enable buzzer 0:disable buzzer

BYTE IP1; //(13)IP address of reader

BYTE IP2; //(14)

BYTE IP3; //(15)

BYTE IP4; //(16)

BYTE Port1; //(17)high-order of reader port

BYTE Port2; //(18)

```
        BYTE    Mask1;                  //(19)reader mask1

        BYTE    Mask2;                  //(20)reader mask2

        BYTE    Mask3;                  //(21)reader mask3

        BYTE    Mask4;                  //(22)reader mask4

        BYTE    Gateway1;               //(23)reader address gateway

        BYTE    Gateway2;               //(24)

        BYTE    Gateway3;               //(25)

        BYTE    Gateway4;               //(26)

        BYTE    MAC1;                   //(27)MAC address of reader

        BYTE    MAC2;                   //(28)

        BYTE    MAC3;                   //(29)

        BYTE    MAC4;                   //(30)

        BYTE    MAC5;                   //(31)

        BYTE    MAC6;                   //(32)

    } ReaderBasicParam;

                                        //Auto parameter of reader

    typedef struct tagReaderAutoParam

    {
```

```
BYTE    AutoMode;           //(1)tag reading mode：0-timing, 1-trigger.
BYTE    TimeH;              //(2)tag storage time: unit: second. Default 1.
BYTE    TimeL;              //(3)
BYTE    Interval;       //(4)0-10ms，1-20ms，2-30ms，3-50ms，4-100ms. Default 2. Auto reading tag once at intervals.
BYTE    NumH;               //(5)tag storage quantity：default 1. The quantity of read tag ID stored in reader memory.
BYTE    NumL;               //(6)
BYTE    OutputManner;       //(7)data output format：0-terse，1-standard，2-XML. Default 0.
BYTE    OutInterface;   //(8)output interface：0—RS232，1—RS485，2—RJ45. Default 0.   3- Wiegand26    4- Wiegand34
BYTE    WiegandWidth;        //(9)value of Weigand pulse width.
BYTE    WiegandInterval;  //(10)value of Weigand pulse interval.
BYTE    ID_Start;         //(11)start address of output ID, value 0～4.
BYTE      IDPosition;     //(12)storage address for tag ID in tag.
BYTE    Report_Interval;  //(13) report interval：unit is second. Default 1. Automatically notify host pc once at intervals.
BYTE    Report_Condition; //(14)condition of report：default 1. 0-notify now，1-timing，2-add，3-remove，4-change
```

```
    BYTE    Report_Output;              //(15)report output port

    BYTE    Antenna;                //(16)select antenna.1-ant1,2-ant2,4-ant4,8-ant8

    BYTE    TriggerMode;            //(17)trigger mode(default0): 0-low level 1-high level

    BYTE    HostIP1;                //(18)notified IP address

    BYTE    HostIP2;                //(19)

    BYTE    HostIP3;                //(20)

    BYTE    HostIP4;                //(21)

    BYTE    Port1;                      //(22)notified port

    BYTE    Port2;                      //(23)

    BYTE    Reserve24;                  //(24)notified MAC,mofi by mqs 20121207 reserve

    BYTE    ArgentinaSim;           //(25)//emulation mode(argentina),0—non-emulation，
1--emulation

    BYTE    CardTime1;                  //(26)//reading time-out 1

    BYTE    CardTime2;                  //(27)//reading time-out 2

    BYTE    ArgentinaMode;              //(28)//5 modes for argentina, 0---Only ATA ; 1---Only
EPC; 2---Only EPC & TID; 3---ATA + EPC; 4---ATA + EPC & TID.

    BYTE    Reserve29;                  //(29)

    BYTE    Alarm;                      //(30)0-no alarm， 1-alarm. To detect if alarm in timing
and trigger modes.
```

```
    BYTE    Reserve31;                  //(31)time interval for standard output，default value is
120s，1～255.

    BYTE    EnableRelay;            //(32)to control relay or not in Auto mode 1:control    0: no
control

} ReaderAutoParam;

                                    //frequency for various countries

static const tagReaderFreq stuctFreqCountry[]=

{

    {"00---FCC(American)", 63, 400, 902600},
    //(0),{"00---FCC(American)", 50, 500, 902750},

    {"01---ETSI EN 300-220(Europe300-220)", 11, 200, 865500},          //(1),{"01---ETSI
EN 300-220(Europe300-220)", -1, -1, -1},

    {"02---ETSI EN 302-208(Europe302-208)", 4, 600, 865700},           //(2)

    {"03---HK920-925(Hong Kong)", 10, 500, 920250},                    //(3)

    {"04---TaiWan 922-928(Taiwan)", 12, 500, 922250},                  //(4)

    {"05---Japan 952-954(Japan)", 0, 0, 0},                            //(5)

    {"06---Japan 952-955(Japan)", 14,200, 952200},                     //(6)

    {"07---ETSI EN 302-208(Europe)", 4, 600, 865700},                  //(7)

    {"08---Korea 917-921(Korea)", 6, 600, 917300},                     //(8)

    {"09---Malaysia 919-923(Malaysia)", 8, 500, 919250},               //(9)
```

```
        {"10--China 920-925(China)", 16, 250, 920625},                      //(10)

        {"11--Japan 952-956(Japan)", 4, 1200, 952400},                      //(11)

        {"12--South Africa 915-919(Poncho)", 17, 200, 915600},              //(12)

        {"13--Brazil 902-907/915-928(Brazil)", 35, 500, 902750},            //(13)

        {"14--Thailand 920-925(Thailand)", -1, -1, -1},                     //(14)

        {"15--Singapore 920-925(Singapore)", 10, 500, 920250},              //(15)

        {"16--Australia 920-926(Australia)", 12, 500, 920250},              //(16)

        {"17--India 865-867(India)", 4, 600, 865100},                       //(17)

        {"18--Uruguay 916-928(Uruguay)", 23, 500, 916250},                  //(18)

        {"19--Vietnam 920-925(Vietnam)", 10, 500, 920250},                  //(19)

        {"20--Israel 915-917", 1, 0, 916250},                               //(20)

        {"21--Philippines 918-920(Philippines)", 4, 500, 918250},           //(21)

        {"22--Canada 902-928(Canada)", 42, 500, 902750},                    //(22)

        {"23--Indonesia 923-925(Indonesia)", 4, 500, 923250},               //(23)

        {"24--New Zealand 921.5-928(New Zealand)", 11, 500, 922250},        //(24)

    };
```

3.2.1.5 Function return code

When command execution fail, function returns error code. Common error code:

Command	Function
00(00H)	Command success or detection correct
01(01H)	Antenna connection fail
02(02H)	Detect no tag
03(03H)	Illegal tag
04(04H)	Read-write power not enough
05(05H)	Read and write protection in the memory bank
06(06H)	Checksum error
07(07H)	Parameter error
08(08H)	Memory bank nonexistence
09(09H)	Password error
10(0AH)	Kill password is all zero
11(0BH)	When reader in auto mode, only receive AutoMode and Reboot commands, other command is illegal
12(0CH)	Illegal user with unmatched password
13(0DH)	External RF interference
14 (0EH)	Tag with read protection
……	……
30(1EH)	Invalid command, such as wrong command
31(1FH)	Unknown command
32(20H)	Other error

3.2.2 Control command function

3.2.2.1 Connect reader

Using serial port Connection:

apiReturn ConnectScanner(HANDLE *hScanner, char *szPort, int nBaudRate);

Function: to establish communication connection with reader, and set baud rate for reader.

Input parameter:

szPort : character pointer directing at communication port, eg. 『COM1』、『COM2』……

nBaudRate: baud rate of serial, effective value is: 9600，19200，38400，57600，115200.

Output parameter: to judge if connection success or failure reason according to apiReturn value returned by function.

hScanner : reader handle

command with reader adress:

apiReturn _stdcall ConnectScanner485 (HANDLE *hScanner, char *szPort, int nBaudRate,int Address);

Function: to establish communication connection with reader, and set baud rate for reader.

Input parameter:

szPort : character pointer directing at communication port, eg.『COM1』、『COM2』……

nBaudRate: baud rate of serial, effective value is: 9600，19200，38400，57600，115200.

Address: reader address.

Output parameter: to judge if connection success or failure reason according to apiReturn value returned by function.

hScanner : reader handle

connection via Ethernet port

apiReturn _stdcall Net_ConnectScanner(SOCKET *hSocket,char *nTargetAddress,UINT nTargetPort,char *nHostAddress,UINT nHostPort);

Function: to establish communication connection with reader, and set baud rate for reader.

Input parameter:

nTargetAddress: target addess, such as『192.168.0.1』……

nTargetPort: target communication port, such as 『1969』

nHostAddress: host address, such as 『192.168.0.2』……

nHostPort: host communication port, such as 『5000』

output parameter:

hSocket : reader communication handle

return: if return value of function is OK, it means connection success, otherwise, connection fail.

Note: each reader shall execute the command to get corresponding reader communication handle hSocket.

3.2.2.2 Disconnection

RS232,RS485 disconnection with reader

apiReturn   DisconnectScanner(HANDLE hScanner)；

apiReturn Net_DisconnectScanner();

Function: to stop the connection with reader to release serial resource.

Input parameter:

hSacnner: reader communication handle

3.2.2.3 Set baud rate

apiReturn _stdcall SetBaudRate(HANDLE hScanner, int nBaudRate,int Address)

apiReturn Net_SetBaudRate(SOCKET hSocket, int nBaudRate);

Function: to set operation baud rate of RS232 port.

Input parameter:

hSacnner/hSocket: reader communication handle

nBaudRate: vaule:9600，19200，38400，57600，115200

Address: RS485 networking address of reader，Address =0 no networking.

return: if function return value is OK, setting success, otherwise failure reason.

3.2.2.4 Read version

apiReturn _stdcall GetReaderVersion(HANDLE hScanner, WORD *wHardVer, WORD *wSoftVer,int Address)

apiReturn Net_GetReaderVersion(SOCKET hSocket, WORD *wHardVer, WORD *wSoftVer,BYTE * IPaddress);

Function: to read hardware and software version number of reader.

Input parameter:

hSacnner/hSocket: reader communication handle

Address: RS485 networking address of reader，Address =0 no networking.

Output parameter:

wHardVer: hardware version number of reader.

WSoftVer: software version number of reader.

return: if function return value is OK, read success, otherwise fail.

3.2.2.5 Set output power

apiReturn _stdcall SetOutputPower(HANDLE hScanner, int nPower1,int Address)

apiReturn Net_SetOutputPower(SOCKET hSocket, int nPower,BYTE * IPaddress);

Function: to set RF output power of reader.

Input parameter:

hSacnner/hSocket: reader communication handle

nPower1/nPower: output power value

Address: RS485 networking address of reader， Address =0 no networking.

return: if function return value is OK， it means setting success, otherwise fail.

3.2.2.6 Set operation frequency

apiReturn _stdcall SetFrequency(HANDLE hScanner, int Min_Frequency, int Max_Frequency,int Address)

apiReturn _stdcall Net_SetFrequency(SOCKET hSocket, int Min_Frequency, int Max_Frequency)

Function: to set operation frequency of reader.

Input parameter:

hSacnner/hSocket: reader communication handle

Min_Frequency: start frequency of reader， vaule is 0-59.

Max_Frequency: end frequency of reader， value is 0-59.

When Min_Frequency = Max_Frequency，reader works in fixed frequency.

Address: RS485 networking address of reader，Address =0 no networking.

function: if function return value is OK，it means setting success, otherwise fail.

3.2.2.7 Read reader basic operation parameter

apiReturn _stdcall ReadBasicParam(HANDLE hScanner, ReaderBasicParam * pParam,int Address);

apiReturn Net_ ReadBasicParam (SOCKET hSocket, ReaderBasicParam * pParam);

Function: to read operation parameter written by last command in reader.

Input parameter:

hSacnner/hSocket: reader communication handle

Address: RS485 networking address of reader，Address =0 no networking.

Output parameter:

pParam: return operation parameter of reader, 32 bytes

return: if function return value is OK, it means read success, otherwise failure reason.

3.2.2.8 Set reader basic operation parameter

apiReturn _stdcall WriteBasicParam(HANDLE hScanner, ReaderBasicParam * pParam,int Address);

apiReturn Net_ WriteBasicParam (SOCKET hSocket, ReaderBasicParam * pParam);

Function: to reader operation parameter for reader.

Input parameter:

hSacnner/hSocket: reader communication handle

pParam: operation parameter of reader, 32 bytes

Address: RS485 networking address of reader， Address =0 no networking.

return: if function return value is OK， it means setting success, otherwise failure reason.

3.2.2.9 Read reader auto parameter

apiReturn _stdcall ReadAutoParam (HANDLE hScanner, ReaderAutoParam* pParam,int Address);

apiReturn Net_ ReadAutoParam (SOCKET hSocket, ReaderAutoParam* pParam);

Function: to read operation parameter written by last command in reader.

Input parameter:

hSacnner/hSocket: reader communication handle

Address: RS485 networking address of reader， Address =0 no networking.

Output parameter:

pParam: return operation parameter of reader, 32 bytes

return: if function return value is OK， it mean read success, otherwise failure reason.

3.2.2.10 Set reader auto parameter

apiReturn _stdcall WriteAutoParam (HANDLE hScanner, ReaderAutoParam* pParam,int Address);

apiReturn Net_ WriteAutoParam (SOCKET hSocket, ReaderAutoParam* pParam);

Function: to set auto operation parameter in reader.

Input parameter:

hSacnner/hSocket: reader communication handle

pParam: operation parameter of reader, 32 bytes

Address: RS485 networking address of reader， Address =0 no networking.

return: if function return value is OK， it means set success, otherwise failure reason.

3.2.2.11 Select antenna

apiReturn _stdcall SetAntenna(HANDLE hScanner, int Antenna,int Address) ;

apiReturn Net_SetAntenna(SOCKET hSocket, int Antenna);

Function: to select which antenna to receive and transmit signal.

Input parameter:

hSacnner/hSocket: reader handle

Antenna: antenna number， 1-No1 antenna， 2-No2 antenna， 4-No3 antenna， 8-No4 antenna

Address: RS485 networking address of reader， RS485Address=0 no networking.

Output parameter:

return: if function return value is OK, it means set success, otherwise failure reason.

3.2.2.12 Set relay state in reader

apiReturn _stdcall SetRelay(HANDLE hScanner, int Relay,int Address) ;

apiReturn Net_SetRelay(SOCKET hSocket, int Relay);

Function: to set relay state for reader.

Input parameter:

hSacnner/hSocket: reader handle

Relay: 1byte. Bit0=1, No1 relay on; Bit0=0, No1 relay off. Bit1=1, No2 relay on; Bit1=0, No2 relay off. And so on.

Address: RS485 networking address of reader，Address =0 no networking.

return: if function return value is OK, it means set success, otherwise failure reason.

3.2.2.13 Reboot reader

apiReturn Reboot(HANDLE hScanner,int Address);

apiReturn Net_Reboot(SOCKET hSocket);

Function: to reboot reader, power on again.

Input parameter:

hSacnner/hSocket: reader communication handle

Address: RS485 networking address of reader， Address =0 no networking.

return: if function return value is OK, it means set success, otherwise failure reason.

3.2.2.14 Set time

apiReturn _stdcall SetReaderTime(HANDLE hScanner, ReaderDate time ,int Address) ;

apiReturn Net_SetReaderTime(SOCKET hSocket, ReaderDate time)

Function: to set time for reader based on Host pc time.

Input parameter:

hSacnner/hSocket: reader communication port handle

time: host pc time， 6bytes

Address: RS485 networking address of reader, Address =0 no networking.

return: if function return value is OK, it means set success, otherwise failure reason.

3.2.2.15 Get time

apiReturn _stdcall GetReaderTime(HANDLE hScanner, ReaderDate *time ,int Address) ;

apiReturn GetReaderTime(SOCKET hSocket, ReaderDate *time)

Function: to read time of reader.

Input parameter:

hSacnner/hSocket: reader communication handle

Address: RS485 networking address of reader， Address =0 no networking.

Output parameter:

time: return time of reader, 6bytes

return: if function return value is OK，it means read success, otherwise failure reason.

3.2.2.16 Get record

apiReturn _stdcall GetRecord(HANDLE hScanner, ReaderDate *stime, ReaderDate *etime, int startaddr, int listlen, int *relistlen, int *taglen, BYTE * data) ;

apiReturn _stdcall Net_GetRecord(SOCKET hSocket, ReaderDate *stime, ReaderDate *etime, int startaddr, int listlen, int *relistlen, int *taglen, BYTE * data);

Function: to read record of identified tags in reader.

Input parameter:

hSacnner/hSocket: reader communication handle

stime: start time

etime: end time

startaddr: start record

listlen: records count to read

output parameter:

relistlen: records count by actual reading

taglen: length of each record by actual reading

data: records by reading

return: if function return value is OK， it means read success, otherwise failure reason.

3.2.2.17 Delete all records

apiReturn _stdcall DeleteAllRecord(HANDLE hScanner) ;

apiReturn _stdcall Net_DeleteAllRecord(SOCKET hSocket);

Function: to delete all records in reader.

Input parameter:

hSacnner/hSocket: reader communication port handle

return: if function return value is OK, it means set success, otherwise failure reason.

3.2.3 Network command

3.2.3.1 Set IP address for reader

apiReturn _stdcall SetReaderNetwork(HANDLE hScanner, BYTE IP_Address[4], int Port, BYTE Mask[4], BYTE Gateway[4],int Address) ;

apiReturn _stdcall Net_SetReaderNetwork(SOCKET hSocket, BYTE IP_Address[4], int Port, BYTE Mask[4], BYTE Gateway[4]);

Function: to set network IP address for reader.

Input parameter:

hSacnner/hSocket: reader communication port handle

IP_Address[4]: IP address of reader

Port: network port number of reader

Mask[4]: network IP address mask of reader

Gateway[4]: gateway of reader

Address: RS485 networking address of reader，Address =0 no networking.

Return: if function return value is OK, it means set success, otherwise failure reason.

3.2.3.2 Get IP address in reader

apiReturn _stdcall GetReaderNetwork(HANDLE hScanner, BYTE *IP_Address, int *Port, BYTE *Mask, BYTE *Gateway,int Address) ;

apiReturn _stdcall Net_GetReaderNetwork(SOCKET hSocket, BYTE *IP_Address, int *Port, BYTE *Mask, BYTE *Gateway);

Funtion: to get network IP addess of reader.

Input parameter:

hSacnner/hSocket: reader communication port handle

output parameter:

IP_Address[4]: IP address of reader

Port: network port number of reader

Mask[4]: network IP address mask of reader

Gateway[4]: gateway of reader

Address: RS485 networking address of reader，Address =0 no networking.

return: if function return value is OK, it means set success, otherwise failure reason.

3.2.3.3 Set reader MAC address

apiReturn _stdcall SetReaderMAC(HANDLE hScanner, BYTE MAC[6],int Address) ;

apiReturn _stdcall Net_SetReaderMAC(SOCKET hSocket, BYTE MAC[6]);

Function: to set network MAC address for reader.

Input parameter:

hSacnner/hSocket: reader communication port handle

MAC[6]: network MAC address of reader

Address: RS485 networking address of reader， Address =0 no networking.

return: if function return value is OK, it means set success, otherwise failure reason.

3.2.3.4 Get reader MAC address

apiReturn _stdcall GetReaderMAC(HANDLE hScanner, BYTE *MAC,int Address) ;

apiReturn _stdcall Net_GetReaderMAC(SOCKET hSocket, BYTE *MAC);

Function: to get network MAC address of reader.

Input parameter:

hSacnner/hSocket: reader communication port handle

output parameter:

MAC: network MAC address of reader

Address: RS485 networking address， Address =0 no networking.

return: if function return value is OK, it means set success, otherwise failure reason.

3.2.4 Read write ISO18000-6B function

3.2.4.1 Identify tag ID number

apiReturn _stdcall ISO6B_ReadLabelID(HANDLE hScanner, BYTE *IDBuffer, int *nCounter,int Address) ;

apiReturn _stdcall Net_ISO6B_ReadLabelID(SOCKET hSocket, BYTE *IDBuffer, int *nCounter);

Function: to read all ID number of all readable tags in antenna radiation field.

Input parameter:

hSacnner/hSocket: reader communication port handle

output parameter:

nCounter: return tag count by actual reading ID number

IDBuffer: store ID number of tag by reading in cache

Address: RS485 networking address of reader， Address =0 no networking.

return: if function return value is OK, it means identify success, otherwise failure reason.

3.2.4.2 Identify selected tag ID number

apiReturn _stdcall ISO6B_ListSelectedID(HANDLE hScanner, int Cmd, int ptr, BYTE Mask, BYTE *Data, BYTE *IDBuffer, int *nCounter,int Address) ;

apiReturn _stdcall Net_ISO6B_ListSelectedID(SOCKET hSocket, int Cmd, int ptr, BYTE Mask, BYTE *Data, BYTE *IDBuffer, int *nCounter);

Function: to read ID number of selected tags in antenna radiation field.

Input parameter:

hSacnner/hSocket: reader communication port handle

Cmd: condition of selected tag

00	Equal to
01	Unequal to
02	Greater than
03	Less than

ptr: start address of tag data, value 0～223

Mask: data mask； each bit of the byte is corresponding to a comparative bit. 0 means the byte not for comparison； 1 means the byte for comparison.

Data: data to compare

Address: RS485 networking address of reader， Address =0 no networking.

Output parameter:

nCounter: return tag count by actual reading ID number

IDBuffer: store ID number of tag by reading in cache

return: if function return value is OK, it means identify success, otherwise failure reason.

3.2.4.3 Read data block

apiReturn ISO6B_ReadByteBlock(HANDLE hScanner, BYTE *IDBuffer, BYTE ptr, BYTE len,BYTE *Data,int Address)

apiReturn _stdcall Net_ISO6B_ReadByteBlock(SOCKET hSocket, BYTE *IDBuffer, BYTE ptr, BYTE len,BYTE *Data);

Function: to read data in a section of continous memory of tag.

Input parameter:

hSacnner/hSocket: reader communication port handle

IDBuffer: ID number of tag to read

ptr: start addess of tag memory to read(0～223 Byte)

len: length of data block, that’s how many bytes for once reading(Byte)

Address: RS485 networking address of reader， Address =0 no networking.

Output parameter:

Data: return data by reading

Note: nLen shall be ≤ 32.　(nAddress+nLen) ≤223。

Return: if function return value is OK, it means read success, otherwise failure reason.

3.2.4.4 Write data block

apiReturn _stdcall ISO6B_WriteByteBlock(HANDLE hScanner, BYTE *IDBuffer, BYTE ptr, BYTE len, BYTE *Data,int Address) ;

apiReturn _stdcall Net_ISO6B_WriteByteBlock(SOCKET hSocket, BYTE *IDBuffer, BYTE ptr, BYTE len, BYTE *Data);

Function: to write data into appointed address unit of tag

Input parameter:

hSacnner/hSocket: reader communication port handle

IDBuffer: ID number of tag to write

ptr: start address of tag memory to write(8～223)

len: length of data block, that how many words for once wirting(4Bytes/word)

Data: data to write

Note: ptr shall be integral multiple of 4. (nAddress+nLen) ≤223.

Address: RS485 networking address of reader， Address =0 no networking.

return: if function return value is OK， it means write success, otherwise failure reason.

3.2.4.5 Slow write data block

apiReturn _stdcall ISO6B_WriteAByte(HANDLE hScanner, BYTE *IDBuffer, BYTE ptr, BYTE len, BYTE *Data,int Address);

apiReturn _stdcall Net_ISO6B_WriteAByte(SOCKET hSocket, BYTE *IDBuffer, BYTE ptr, BYTE len, BYTE *Data);

Function: to write data into appointed address unit of tag byte by byte.

Input parameter:

hSacnner/hSocket: reader communication port handle

IDBuffer: ID number of tag to write

ptr: start address of tag memory to write(8～223)

len: length of data block， that's how many words to write once(4Bytes/word)

Data: data to write

Note: (nAddress+nLen) ≤223。

Address: RS485 networking address of reader， Address =0 no networking.

return: if function return value is OK， it means write success, otherwise failure reason.

Note: the command writes data into tag byte by byte, slow in speed. Only used for tags not support previous write command.

3.2.4.6 Set write protection

apiReturn _stdcall ISO6B_WriteProtect(HANDLE hScanner, BYTE *IDBuffer, BYTE ptr,int Address);

apiReturn _stdcall Net_ISO6B_WriteProtect(SOCKET hSocket, BYTE *IDBuffer, BYTE ptr);

Function: to set write protect for appointed address unit of appointed tag.

Input parameter:

hSacnner/hSocket: reader communication port handle

IDBuffer: ID number of tag to write

ptr: memory address of tag to set write protect(8～223)

Address: RS485 networking address of reader， Address =0 no networking.

return: if function return value is OK, it means set success, otherwise failure reason.

3.2.4.7 Read write protection

apiReturn _stdcall ISO6B_ReadWriteProtect(HANDLE hScanner, BYTE *IDBuffer, BYTE ptr, BYTE *Protected,int Address);

apiReturn _stdcall Net_ISO6B_ReadWriteProtect(SOCKET hSocket, BYTE *IDBuffer, BYTE ptr, BYTE *Protected);

Function: to read if the appointed address unit of appointed tag is set write protect.

Input parameter:

hSacnner/hSocket: reader communication port handle

IDBuffer: ID number of tag to write

ptr: memory address of tag to read write protection state (0～223)

Address: RS485 networking address of reader， Address =0 no networking.

Output parameter:

Protected: protection state， 0- no protect， 1-protected

return: if function return value is OK, it means read success, otherwise failure reason.

3.2.5 Read write ISO18000-6C function

3.2.5.1 Identify EPC number of ISO18000-6C tag

apiReturn _stdcall EPC1G2_ReadLabelID(HANDLE hScanner, BYTE mem, int ptr, BYTE len, BYTE *mask, BYTE *IDBuffer, int *nCounter,int Address);

apiReturn _stdcall Net_EPC1G2_ReadLabelID(SOCKET hSocket, BYTE mem, int ptr, BYTE len, BYTE *mask, BYTE *IDBuffer, int *nCounter);

Function: to read EPC number of all eligible readable tag in antenna radiation field.

Input parameter:

hSacnner/hSocket: reader communication port handle

mem: select memory bank；

0	Passwrod
1	EPC
2	TID
3	User

ptr: start address of mask(unit:Bit)

len: length of mask(unit:Bit)

mask: mask(unit:Byte)，if len/8 is integer，length of mask is len/8；if len/8 is not integer,length of mask is len/8+1. Last byte data of mask in high-order position, zero fill in low-order position.

Address: RS485 networking address of reader，Address =0 no networking.

Output parameter:

IDBuffer: EPC number of tag by read

NCounter: tag count by read

return: if function return value is OK， it means identify success, otherwise failure reason.

Note: LEN=0, to identify all readable tag ID in antenna radiation field.

3.2.5.2 Read a block data

apiReturn _stdcall EPC1G2_ReadWordBlock(HANDLE hScanner, BYTE EPC_WORD, BYTE *IDBuffer, BYTE mem, BYTE ptr, BYTE len, BYTE *Data, BYTE *AccessPassword,int Address);

apiReturn _stdcall Net_EPC1G2_ReadWordBlock(SOCKET hSocket, BYTE EPC_WORD, BYTE *IDBuffer, BYTE mem, BYTE ptr, BYTE len, BYTE *Data, BYTE *AccessPassword);

Function: to read data in a section of continous address of tag.

Input parameter:

hSacnner/hSocket: reader communication port handle

EPC_WORD: length of EPC, L(unit:Word)； such as length of 96BitsEPC L=6(Words)；

IDBuffer: EPC number of selected tag

mem: select memory bank； 0-Password， 1-EPC， 2-TID， 3-User.

ptr: start address to read(unit:WORD)

len: length to read(unit:WORD)

AccessPassword: 4bytes AccessPassword

Address: RS485 networking address of reader， Address =0 no networking.

Output parameter:

Data: data to read

return: if function return value is OK， it means read success, otherwise failure reason.

Note: AccessPassword only workable for password memory bank in password lock state.

3.2.5.3 Write a block data

apiReturn _stdcall EPC1G2_WriteWordBlock(HANDLE hScanner, BYTE EPC_WORD, BYTE *IDBuffer, BYTE mem, BYTE ptr, BYTE len, BYTE *Data, BYTE *AccessPassword,int Address);

apiReturn _stdcall Net_EPC1G2_WriteWordBlock(SOCKET hSocket, BYTE EPC_WORD, BYTE *IDBuffer, BYTE mem, BYTE ptr, BYTE len, BYTE *Data, BYTE *AccessPassword);

Function: to write data into appointed address unit of tag.

Input parameter:

hSacnner/hSocket: reader communication port handle

EPC_WORD: length of EPC, L(unit:Word)； such as length of 96BitsEPC L=6(Words)；

IDBuffer: EPC number of selected tag

mem: select memory bank

0	Password
1	EPC
2	TID
3	User

ptr: start address to write(unit:WORD)

len: length to write(unit:WORD)

Data: data to write

AccessPassword: 4bytes AccessPassword

Address: RS485 networking address of reader， Address =0 no networking.

return: if function return value is OK， it means write success, otherwise failure reason.

Note: AccessPassword only workable for memory bank in password lock state. If memory bank unlocked, can be written without password. If memory bank locked permanently, password useless.

3.2.5.4 Set read write protection state

apiReturn _stdcall EPC1G2_SetLock(HANDLE hScanner, BYTE EPC_WORD, BYTE *IDBuffer, BYTE mem, BYTE Lock, BYTE *AccessPassword,int Address);

apiReturn _stdcall Net_EPC1G2_SetLock(SOCKET hSocket, BYTE EPC_WORD, BYTE *IDBuffer, BYTE mem, BYTE Lock, BYTE *AccessPassword);

Function: to set write protect for appointed memory bank of appointed tag.

Input parameter:

hSacnner/hSocket: reader communication port handle

EPC_WORD: length of EPC, L(unit:Word)； such as length of 96BitsEPC L=6(Words)；

IDBuffer: EPC number of selected tag

mem: select memory bank

0	Kill Password
1	Access Password
2	EPC number
3	TID ID number
4	User

Lock: control word.

0	Writable
1	Permanently writable
2	Writable from secured state
3	Never writable
4	Readable and writable
5	Permanently readable and writable
6	Readable and writable from secured state
7	Never readable and writable

Note:0～3 only apply to EPC、TID and User memory bank；4～7 only apply to Kill Password and Access Password.

AccessPassword: 4bytes AccessPassword

Address: RS485 networking address of reader， Address =0 no networking.

return: if function return value is OK， it means set success, otherwise failure reason.