VI-83

RFID Integrated Reader

Development Handbook

- The producer has the right to do revisions for hardware, software and manual of product without statement. And this manual is subject to change without notice.
- Specifications and power of this product is subject to standard in origin. Please being sure that voltage is up to requirement, read and learn about safety precautions, especially in open-air installation of outdoor.

1 Content

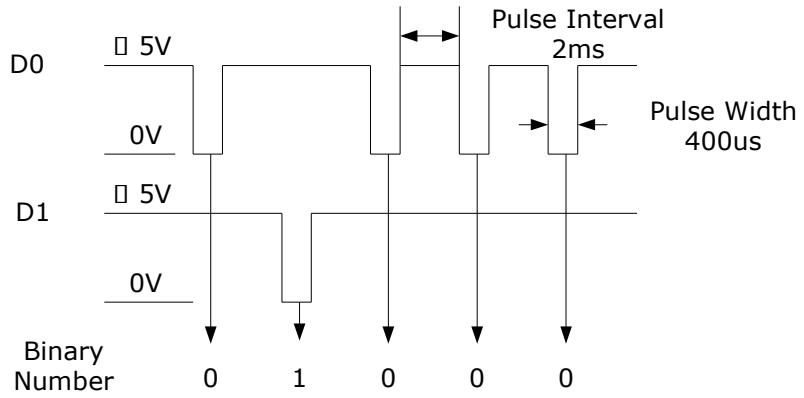
2 Applying development of unilateral communication

2.1 Applying development of unilateral communication

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

2.2 Wiegand interface agreement

Wiegand interface transmits unilaterally, can send read card numbers to controller only, but controller can not send signal to reader. Transmittal signal from Wiegand interface is as follows:



At present, export wave type from Wiegand interface is as follows:

A. A.400us in pulse width, 2.0ms in pulse interval

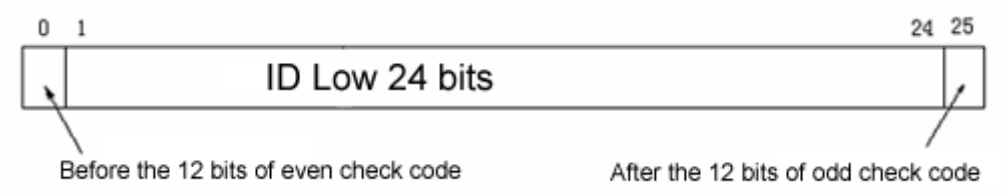
B. 100us in pulse width, 1.6ms in pulse interval

C. 1.0ms 50us in pulse width, 1.0ms in pulse interval

Wiegand interface has wiegand26 and wiegand34 2types.

2.2.1 Wiegand26 format

Wiegand26 transmits 26 digits data every time, 24digits of them are efficient. We stipulates correspondent electronic tag ID for these 24 digits,or user defines numbers. Format of transmission is as follows:

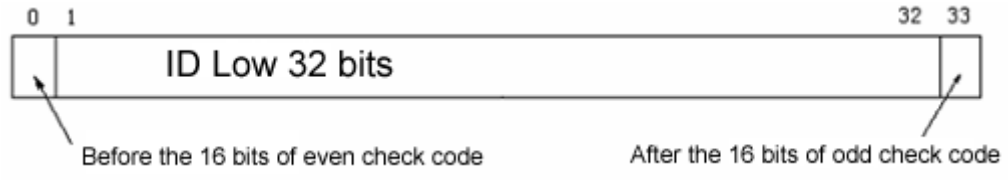


Even: verified data adds parity bit1 is even number.

Odd: Verified data adds parity bit 1 is odd number.

2.3 Wiegand34 format

Wiegand34 transmits 32 digits of efficient data each time,we stipulates electronic tag ID numbers for these 32digits or user self-defines. Transmission format is as follows:



2.4 RS485 interface agreement

When adopts RS485 interface or 232 interface export data, need to set up communication rate of interface RS485 or 232, output data format of interface RS485 or RS232 is:

Introduction:

BODY				CHECK
STX	DATA		ETX	BCC
02	Antenna number(2byte ASCIIcode)	ID number or User-defined Code (8 bytes ASCII)	03	Check Code

A. Symbol of data commence STX=02H, symbol of data ending ETX=03H

B. DATA is antenna serial numbers(2Byte)+tag Id numbers(8Bytes), length is 10bytes ASCII code. Expression mode of converting hexadecimal notation to be ASCII

Divides data from high to low, every 4 digits in a team, then put value of 4 binary digits in expression of ASCII codes. As value range of 4binary digits is 0H-FH, so 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 be ASCII codes. Antenna number 1(ASCII CODE) is (30H 31H), antenna number 2(ASCII code) is (30H 32H).

3 Serial port intercommunication agreement

2 ways for application development:

1. Use control code of intercommunication to operate reader directly.

2. Use matched SDK software with reader, adjust API function to operate reader.

3.1 Summarize

In RFID application system, reader is connected with control (PC) via RS232 port , and receives commands from control, then returns the result that the commands are performed, to control. Therefore, we name Data Communications Packet which sends commands from control to reader, to be Command Packet, and name that which sends results from reader to control, to be Return Packet.

3.1.1 Command packet format without address

BootCode	Length	Command	Command Parameters	CheckSum

Like photo above, command packet composes of 5 parts:

Boot Code: 1byte, fixed to be 40H

Length: including effective length, 1byte. The length is total bytes of lateral 3 parts.

Command: command code, 1byte

Command Parameters: command parameter, length is changing with command.

CheckSum: Checksum, 1byte, is all bytes from bootcode to command param discard patch code.

3.1.2 Command packet format with address

BootCode	Length	Command	Address	Command Param	CheckSum

Like photo above, command packet composes of 5 parts:

BootCode: 1byte, fixed to be 40H

Length: including effective length, 1byte. The length is total bytes of lateral 3 parts.

Command: command code, 1byte

Reader address, 1-254, 0 and 255 is broadcasting address.

Command Param: command parameter, length is changing with command.

Checksum, 1byte, is all bytes from bootcode to command param discard patch code.

3.1.3 Command packet format without

address

BootCode	Length	Command	Return Data	CheckSum

Like photo above, command packet composes of 5 parts:

BootCode: when executes command correctly, return packet is F0H, when command fails to execute, leading code of return packet is F4H, 1byte

Length: including effective length, 1byte. The length is total bytes of lateral 3 parts.

Command: Command code, 1byte, same to received command code, is reaction of return packet.

Return data: returns command and executes result, length is changing with command.

Checksum: 1byte, is all bytes from bootcode to command param discard patch code.

3.1.4 Command packet format with address

BootCode	Length	Command	Address	Return Data	sumcheck

Like photo above, command packet composes of 5 parts:

BootCode: 1byte, fixed to be 40H

Length: including effective length, 1byte. The length is total bytes of lateral 3 parts.

Command: command code, 1byte

Address: Reader address, 1-254, 0 and 255 is broadcasting address.

Command Param: command parameter, length is changing with command.

CheckSum: Checksum, 1byte, is all bytes from bootcode to command param discard patch code.

3.1.5 Error code

When fails to execute command, bootcode of return packet is F4H, and return data is wrong code of 1 byte. Usual error code is:

00(00H)	command success or test correct
01(01H)	connection antenna failure
02(02H)	fail to check tag
03(03H)	illegal tag
04(04H)	read write power is inadequate
05(05H)	write protection in this area
06(06H)	check and error
07 (07H)	parameter mistake
08 (08H)	non-exsite data area
09 (09H)	wrong password
10 (0AH)	killed password can’’t be 0
11(0BH)	when reader is active ,the command is illegal.
12(0CH)	wrong matched password illegal user
13(0dH)	RF disturb from external
14 (0EH)	tag read protection
……	
30(1EH)	Invalid order,say wrong parameter order
31(1FH)	unknown order
32(20H)	other mistakes

3.1.6 For example

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

40H	Lead code
03H	including effective length is 3bytes
01H	set command code of reader baud rate
04H	9600bps represents 9600bps
B8H	checksum

Is 40H+03H+01H+04H=48H's patch code

If executes correctly, return packet is F0H 02H 01H 0DH

If execute wrongly, return packet can be F4H 03H 01H 1FH E9H

3.2 Control command format of serial

3.2.1 SetBaudRate()

Function: set operating baud rate for interface of RS232.

Latest communication rate of interface RAS232 is 9600bps after reader loads new procedure. When reader received the order,it resets baud rate of interface according to command parameter. Whatever power supply of reader is closed or not, the performance rate will keep same to next reset.

Command code: 01H

Command parameter: 1byte BPS, value: 00H-08H, represents:

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

Command packet:【40H 03H 01H BPS Checksum】

Rebound data: If order executes right, return data is empty.

【F0H 02H 01H 0DH】

Command format with reader address

Command code: 01H

Reader address parameter: address

Command parameter: 1byte of BPS, valued: 00H-08H, represents:

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

With address command packet: 【40H 04H 01H address BPS CheckSum】

Return data: If command executes right, return data is empty.

With address command packet: 【F0H 03H 01H address CheckSum】

3.2.2 Get ReaderVersion()

Function: gets version numbers from hardware and software of reader.

Command code: 02H

Command parameter: none

command packet: 【40H 02H 02H BBH】

Return data: If order executed right, then data part in return packet is version

number in 4bytes

Byte0	chief edition of hardware
Byte1	hypo edition of hardware
Byte2	chief edition of software
Byte3	hypo edition of software

For example: If type of reader is 1102, edition numbers of software is V1.5, then return packet is:

F0H 06H 02H 0BH 02H 01H 05H DDH

Format of command with address

Command code: 02H

Parameter of reader address

Command parameter: none

Command packet: 40H 03H 02H address CheckSum

Return data: If order executes right, then data part in return packet is version number in 4bytes

Byte0	chief version of hardware
Byte1	hypo-version of hardware
Byte2	chief version of software
Byte3	hypo version of software

For example: If type of reader is 1102, edition numbers of software is V1.5, then return packet is:

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

3.2.3 SetOutputPower()

Function:set output power coefficient of reader. It takes effect after reader sets new output power, will keep it same till reset, whatever power supply is closed or not.

Command code: 04H

Command parameter: 1 byte of P, expresses power cost, valued 0-160.

command packet: 40H 03H 04H P CheckSum

Return data: If order executes right, return data is empty.

F0H 02H 04H 0AH

Command format with reader address

Command code: 04H

Address parameter of reader.

Command parameter: 1 byte P, expresses power cost, values0-160.

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

Return data: If order executes right, return data is empty.

F0H 03H 04H address CheckSum

3.2.4 SetFrequency ()

Function: set output power coefficient of reader. It takes effect after reader sets new output power, will keep it same till reset, whatever power supply is closed or not.

Command code: 05H

Command parameter: 2bytes, byte1 expresses origination frequency fmin, costs1-63, byte2 expresses ending frequency fmax, values1-63. If fmin>fmax, it expresses reader works according to jump frequency, range fmin-fmax. If fmin=fmax, it expresses reader works in mode of fixed frequency, frequency is fmax.

command packet: 40H 04H 05H fmin fmax CheckSum

Return data: If command executes right, return data is empty.

F0H 02H 05H 09H

Command format with reader address

Command code: 05H

Reader address parameter

Command parameter: 2bytes, byte1 expresses origination frequency fmin, costs1-63, byte2 expresses ending frequency fmax, values1-63. If fmin>fmax, it expresses reader works according to jump frequency, range fmin-fmax. If fmin=fmax, it expresses reader works in mode of fixed frequency, frequency is fmax.

command packet: 40H 05H 05H address fmin fmax CheckSum

Return data: If command executed right, return data is empty.

F0H 03H 05H address CheckSum

3.2.5 ReadParam()

Function: operating parameter of command read from reader.

Command code: 06H

Command parameter: none

command packet: 40H 02H 06H B8H

Return data: If succeed, data part of return packet is 32 digits pam for command setting.

F0H 22H 06H PAM CheckSum

Command format with reader address

Command code: 06H

Parameter of reader address

Command parameter: none

command packet: 40H 03H 06H address CheckSum

Return data: If succeed, data part of return packet is 32 digits pam for command setting.

F0H 23H 06H address PAM CheckSum

3.2.6 WriteParam ()

Function: set basic operating parameter of serial port baud rate, launch frequency and output power

Command code: 09H

Command parameter: parameter in 32 bytes.

command packet: 40H 22H 09H PAM CheckSum

Return data: If order executed right, return data is empty. If succeed, data part of return packet is 32 digits pam for command setting.

Command format with reader address F0H 02H 09H 05H

Command code: 09H

Address parameter of reader

Command parameter: parameter in 32 digits

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

Return data: If order executed right, return data is empty. If succeed, data part of return packet is 32 bytes pam for command setting.

F0H 03H 09H address CheckSum

32bytes parameter (1parameter1byte) :

1) communication rate of Serial port,values 00H-08H, represents rate and order default is 07H

2) Launch power, values:30-160

3) Origination of transmiting microwave signal frequency, default is :1-63

4) Antenna options(default1)

1	choose antenna in interface of ANT1
2	choose antenna in interface of ANT2
4	Express antenna in interface ANT3

8	Express antenna in interface ANT4

5) Set operating mode of reader(default 2)

0	timing mode
1	spring mode
2	command mode
3	timing mode2

6) Set reading alternation (default 0): when operation in timing, intermission time of reading.

0	10ms
1	30ms
2	50ms

7) Set output mode of card numbers (default 0): read same card numbers for a long time, check if outputs every time.

0	Direct output: outputs in each time.
1	Standard output: 2 mins in intermission

8) Sets spring mode (default 0): when operation, choose for rising or decline.

0	Low electrical level
1	high electrical level

9) Set card number address of storage in electronic tag (default is 0)

0	numbers of tag
1	user defined card number

10) Set if need to determine validity of card (default 0)

0	no
1	yes

11) Set output interface and format of tag number (default 0)

0	Adopts format of wiegand 26, exports card numbers from wiegand interface.
1	Adopts format of wiegand34, exports card numbers

	from wiegand interface.
2	Exports card numbers from RS485 interface.
3	Export card numbers from RS232 interface.

12) Set most read amount, (default 2).

13) Adjust depth, values 30-160

14) types of tag: default is 01H

01H	ISO18000-6B
02H	EPCC1
04H	ISO18000-6C
08H	ISO18000-6D

03H expresses to read IS018000-6B and EPCC1 tag at the same time analogy.

15) Wiegand interface exports impulse width, default 40.

16) Export impulse alternation of Wiegand, default 200.

17) Set output card numbers at origination, values0-4, default 0

18) Frequency termination of sending microwave signal, values: 1-63 (default 63)

19) ISO18000-6D tag data pages and duration of reading, 4Bits high is data pages, 0 expresses no data page, 1-2 expresses with 1 or 2 page data, low 4bits is duration time: 0-5ms, 1-10ms, 2-20ms, 3-30ms,…, 16-160ms. Default is 14H.

20) Standard output termination; default is 120s, 1-255.

21) function of byte is as follows:

Bit sequence	Bit 7	Bit 6	Bit 5	Bit 4	Bit3	Bit2	Bit1	Bit0
function	-	-	-	-	1:network	1:high	1:collection reader	1:enable

					building 0:no network building	gain 0:low gain	0:non-collection reader	buzzer 0:unable buzzer

22) Address of reader: 0and 155 is broadcasting address, all readers execute when receive orders or export ID actively, 1-254 is reader personal address, reader executes when receives corresponding order to it selves address.

23) 30 reservation

31) Emission model

0	expresses mode of receive and sending
1	expresses sending mode

Definition of sending mode: sending RF instantly.

32) Confection set

0	expresses no confection signal
1	expresses confection signal

3.2.7 SetAntenna()

Function: choose antenna to receive and send signal

Command code: 0AH

Command parameter: 1byte is chosen to be antenna no.

1	no.1 antenna
2	no.2 antenna
4	no.3 antenna
8	no.4 antenna

Command packet: 40H 03H 0AH No CheckSum

Return data: if command executed right, data in return packet is empty

F0H 02H 0AH 04H

Instruction format with reader address

Command code: 0AH

Address parameter of reader

Command parameter: 1byte is chosen to be antenna no.

1	no.1 antenna
2	no.2 antenna
4	no.3 antenna
8	no.4 antenna

Command packet: 40H 04H 0AH address No CheckSum

Return data: if command executed right, return data is empty.

F0H 03H 0AH address CheckSum

3.2.8 Reboot()

Function: reboot reader, same to renew electricity after electricity is off.

Command code: 0EH

Command parameter: none

command packet: 40H 02H 0EH B0H

Return data: if succeed, return packet is empty.

Command format with reader address:F0H 02H 0EH 00H

Command code: 0EH

Address parameter of reader

Command parameter: none

Command packet:40H 03H 0EH address CheckSum

Return data: if succeed, return packet is empty.

F0H 03H 0EH address CheckSum

3.2.9 SetRelay()

Function: set state of reader and relay

Command code: 03H

Command parameter: 1byte

Bit0=1	no1relay close
Bit0=0	No1 relay disconnection
Bit1=1	No2relay close
Bit1=0	No2 relay disconnection

Analogy

command packet:40H 03H 03H K CheckSum

Return data: if succeed, return packet is empty.

F0H 02H 03H 0BH

Address parameter of reader

Command code: 03H

Address parameter of reader

Command parameter: 1byte

Bit0=1	no1relay close
Bit0=0	No1 relay disconnection
Bit1=1	No2relay close
Bit1=0	No2 relay disconnection

Analogy

command packet:□40H 04H 03H address K CheckSum□

Return data: if succeed, return packet is empty.

□F0H 03H 03H address CheckSum□

3.2.10 SetReaderTime()

Function: set reader time

Command code: 11H

Command parameter: 5bytes: year yy/month mm/daydd

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

Return data: if succeed, return packet is empty.

□F0H 02H 11H FDH□

Command format with reader address

11H command code: 11H

Address parameter of reader

Command parameter: none

command packet: 40H 09H 11H address yy mm dd hh ff ss CheckSum

Return data: if succeed, return packet is empty.

F0H 03H 11H address CheckSum

3.2.11 GetReaderTime()

Function: get read time

Command code: 12H

Command parameter: none

Command packet:40H 02H 12H ACH

Return data: If succeed, return data is rear, month, day, hour, min

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

Command format with reader address

Command code: 12H

Address parameter of reader

Command parameter: none

command packet:40H 03H 12H address CheckSum

Return data: If succeed, return data is rear, month, day, hour, min

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

3.2.12 AddLableID()

Function: increase new list based before one for reader

Command code: 13H

Command parameter: 1byte list M (S8), 1byte 1D or EPC length LEN, M pcs ID or EPC data

Command packet:【40H 4+M*LEN 13H M LEN DATA CheckSum】

Return data: If succed, return data is empty

【F0H 02H 13H FBH】

Instruction format with reader address

Command code: 13H

Address parameter of reader

Command parameter: 1byte list M (S8), 1byte 1D or EPC length LEN, M pcs ID or EPC data

Command packet: 【40H 5+M*LEN 13H address M LEN DATA CheckSum】

Return data: If succed, return data is empty

【F0H 03H 13H address CheckSum】

3.2.13 DelLableID()

Function: delete list based on before for reader

Command code: 14H

Command parameter: 1byte list M (S8), 1byte 1D or EPC length LEN, M pcs ID or EPC data

Command packet: 40H 4+M*LEN 14H M LEN DATA CheckSum

Return data: If succeed, return data is empty

F0H 02H 14H FAH

Instruction format with reader address

Command code: 14H

Address parameter of reader

Command parameter: 1byte list M (S8), M=0 expresses to date all list in reader.

Command packet:40H 5+M*LEN 14H address M LEN DATA CheckSum

Return data: If succeed, return data is empty

F0H 03H 14H address CheckSum

3.2.14 GetLableID()

Function: read stored white list in reader

Command code: 15H

Command parameter: 2byte of origination list serial number SADDR, 1byte list number M (≤8)

Command packet:40H 05H 15H SADDR M CheckSum

Return data: if succeed, return packet will be defined as follows:

F0H 4+N*LEN 15H N LEN N*LABLE CheckSum

Note: N(sm) is rebound list, LEN is tag ID and EPN byte length, LABLE is LEN bytes list ID or EPC.

If record non-exist, rebound packet is defined as follows:

F0H 04H 15H 00H 00H F7H

Address parameter of reader

Command code: 15H

Command parameter: 2bytes, original list serial SADDR, 1byte list M (S8)

Command packet:40H 06H 15H address SADDR M CheckSum

Rebound data: If succeed, rebound packet will be defined as follows:

F0H 5+N*LEN 15H address N LEN N*LABLE CheckSum

Note: N(sm) is rebound list, LEN is tag ID and EPN byte length, LABLE is LEN bytes list ID or EPC.

If record non-exist, rebound packet is defined as follows:

F0H 05H 16H address 00H 00H CheckSum

3.2.15 GetRecord ()

Function: read stored tag records in reader.

Command code: 16H

Command parameter: 6bytes, starting time STIMG, 6bytes ending time ETIME, 2bytes origination record serial no SADDR, 1byte register digits M (≤8)

Command packet:□40H 11H 16H STIME ETIME SADDR M CheckSum□

Rebound data: If succeed, rebound packet will be defined as follows:

□F0H 4+N*LEN 16H N LEN N *RECORD CheckSum□

Note: RECORD is 6bytes time+1byte antenna number+Lbytes tagID or EPC.Len is record length=7+L

For example: antenna 3 values 3

If record non-exsited, rebound packet will be defined as follows:

□F0H 04H 16H 00H 00H F6H□

Command format with reader address

Command code: 16H

Address parameter of reader

Command parameter: 6bytes, starting time STIME, 6bytes ending time ETIME, 2bytes starting record serial number SADDR, 1byte record serial M (S8)

Command packet: □40H 12H 16H address STIME ETIME SADDR M CheckSum□

Return dat: if succeed, return packet will be defined as follows:

□F0H 5+N*LEN 16H address N LEN N *RECORD CheckSum□

Note: RECORD is 6bytes time+1byte antenna number+Lbytes tagID or EPC.Len is record length=7+L

For example: antenna 3 values 3

If record non-exsited, rebound packet will be defined as follows:

□F0H 05H 16H address 00H 00H CheckSum□

3.2.16 DelRecord()

Function: delte all stored tag records in reader

Command code: 17H

Command packet:□40H 02H 17H B7H□

Rebound data: if succeed, rebound packet is empty in data

□F0H 02H 17H F7H□

Command format with reader address

Command code: 17H

Address parameter in reader

command packet: □40H 03H 17H address CheckSum□

Rebound data: if succeed, data of rebound is empty

□F0H 03H 17H address CheckSum□

3.2.17 SetReaderNetwork()

Function: set reader address in network

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

Rebound data: if succeed, rebound packet is empty in data part.

F0H 02H 30H DEH

3.2.18 GetReaderNetwork()

Function: get reader network.

Command code: 31H

Command parameter: none

command packet:40H 02H 31H 8DH

Rebound data: if succeed, data part in rebound packet is empty. IP (4Bytes) +PORT (2Bytes) +MASK (4Bytes) +GATEWAY (4Bytes).

F0H 10H 31H IP PORT MASK Gateway CheckSum

3.2.19 SetReaderMAC()

Function: set reader network MAC.

Command code: 32H

Command parameter: 6 bytes parameter MAC

Command packet:□40H 08H 32H MAC CheckSum□

Rebound data: if succeed, data part in rebound packet is empty.

□F0H 02H 32H DBH□

3.2.20 GetReaderMAC()

Function: get reader network MAC

Command code: 33H

Command parameter: none

command packet:□40H 02H 33H 8B□

Rebound data: If succeed, data part in rebound packet is 6 bytes MAC

□F0H 08H 33H MAC CheckSum□

3.3 Command format of Serial read-write tag

3.3.1 Rear-write ISO18000-6B command format

To electronic tag, storage capcity inlay is 2048bits, which divides into 256bytes. There is an address to each byte, correspondent from 0-255.

Thereinto:

- Address0-7eight words (64bits): tag IDnumbers, fixing when products come out, can not be amended.
- Address 8-223user information storage area, can be self distributed.
- Address 224-255wirte protection bytes.

3.3.1.1 ReadLabelID()

Function: list exsited recognized tag ID under range of antenna radiation

Command code: FEH

Command parameter: none

command packet:40H 02H FEH C0H

Rebound data: If succeed, bytes of rebound data part=number of all list tag M (1 byte) + (tag numberof sent out L (<=8)*8(ID) data.

F0H 3+L*8 FEH M L*8 CheckSum

Instruction format with reader address

Command code: FEH

Address parameter in reader

Command parameter: none

command packet:40H 03H FEH address CheckSum

Rebound data: If succeed, bytes of rebound data part=number of all list tag

M (1 byte) + (tag numberof sent out L (<=8)*8(ID) data.

F0H 4+L*8 FEH address M L*8 CheckSum

3.3.1.2 ListIDReport()

Function: get electronic tag ID (already passes rfs_list order) from reader EMS memory.

Command code: FDH

Command parameter: 2bytes, first byte begins from ADDR, second byte is tag number L (<=8)

Command packet:40H 04H FDH ADDR L CheckSum

Rebound data: If succeed, part bytes from rebound data= (tag number*8(ID))

F0H 2+L*8 FDH L*8 CheckSum

Command format with reader address

Command code: FDH

Address parameter in reader

Command parameter: 2bytes, first byte begins from ADDR, second byte is tag number L (<=8)

command packet:40H 05H FDH address ADDR L CheckSum

Rebound data: If succeed,part bytes from rebound data=(tag number*8(ID))

F0H 3+L*8 FDH address L*8 CheckSum

3.3.1.3 ListSelectedID()

Function: list exsited recognized tag ID under range of antenna radiation

Command code: FBH

Command parameter 1: 1 byte is optional tag condition SEL.

00	equal
01	not equal
02	than
03	Less than

Command parameter 2: 1 byte is origination adrress ADDR of tag data, values 0-223

Command parameter 3: 1 byte is data mask; bit of it is corresponding to a comparable word.

0	This byte is not for comparison
1	This byte is for comparison

Command parameter 4: 8 byte is comparable data.

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

Rebound data: if succeed, rebound data part bytes=listed numbered of read tagM (1byte) + (transmitted tag numberL (<=8)*8(ID))

F0H 3+L*8 FBH M L*8 CheckSum

Command format with reader address

Command code: FBH

Address parameter in reader

Command parameter 1:1byte is optional tag condition SEL.

00	equal
01	not equal
02	than
03	Less than

Command parameter 2: 1 byte is tag data origination address ADDR, values 0-223

Command parameter 3: 1 byte is data mask; bit of the byte is corresponding to a comparable byte.

0	This byte is not for comparison
1	This byte is for comparison

Command parameter: 8 byte is comparable data.

command packet: 40H 0EH FBH address SEL ADDR MASK DATA CheckSum

Rebound data: if succeed, rebound data part bytes=listed numbered of read tagM (1byte) + (transmitted tag number L (<=8)*8(ID))

F0H 4+L*8 FBH address M L*8 CheckSum

3.3.1.4 ReadByteBlock()

Function:read data in tag. ISO18000 electronic tag is with memory capacity of 2048bits, 256byte, byte address user can read is 0-223. Length of data module takes byte as unit, stipulates 32 byte can be read each time.

Command code: F6H

Command parameter: 8byte ID, I byte expresses origination address; values 0-223, 1byte expresses module length, values 1-32.

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

Rebound packet: If succeed, rebound data divides nn byte

F0H nn+2 F6H xx …… xx CheckSum

Instruction format with reader address

Command code: F6H

Address parameter in reader

Command parameter: 8byte ID, 1 byte expresses origination address, values 0-223, 1 byte expresses module length, values 1-32.

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

Rebound packet: If succeed, rebound data divides nn byte

F0H nn+3 F6H address xx …… xx CheckSum

3.3.1.5 WriteByteBlock()

Function: write into data to tag, take byte as unit for the data, 4byte at most for once. Byte address user can write is 8-223.

Command code: F5H

Command parameter: 8byte ID, 1byte express origination address, values 8-223,1byte expresses module length, values 1-4. Write data for nn byte.

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

Rebound packet: if succeed, rebound data is empty

F0H 02H F5H 19H

Instruction format with reader address

Command code: F5H

Address parameter in reader

Command parameter: 8 byte ID, 1byte express origination address, values 8-223, 1 byte expresses module length, values 1-4. Write data for nn byte.

Commad packet:□40H 13+nn F5H address id aa nn xx --- xx CheckSum□

Rebound packet: if succeed, rebound data is empty

□F0H 03H F5H address CheckSum□

3.3.1.6 WriteProtect()

Function: set address unit of specified tag to be write protection

Command code: F4H command code

Command parameter: 8byte ID, 1byte address, values 8-223

Command packet: □40H 0BH F4H ID aa CheckSum□

Rebound data: If succeed, leading code of rebound packet is F0H, data part is empty

□F0H 02H F4H 1AH□

Command format with reader address.

Command code: F4H command code

Address parameter of reader

Command parameter: 8byte ID, 1byte address, values 8-223

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

Rebound data: If succeed, leading code of rebound packet is F0H, data part is empty

F0H 03H F4H address CheckSum

3.3.1.7 ReadWriteProtect()

Function: read appointed tag if write protection

Command code: F3H

Command parameter: 8byte ID, 1byte origination address, values0-223

Command packet: 40H 0BH F3H ID aa CheckSum

Rebound data: if succeed, leading code of rebound packet is F0H, 1byte in data part

0	unprotected, F0H 03H F3H 00H 1AH
1	protected, F0H 03H F3H 01H 19H

Instruction format with reader address

Command code: F3H

Address parameter in reader

Command parameter: 8byte ID

1byte origination address, values 0-223

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

Rebound data: if succeed, leading code of rebound packet is F0H, 1byte in data part

0	unprotect, □F0H 04H F3H address 00H CheckSum□
1	protected, □F0H 04H F3H address 01H CheckSum□

3.3.1.8 WriteAByte()

Function: write into data to tag, take byte as unit for the data, 4byte at most for once. Byte address user can write is 8-223.

Command code: F2H

Command parameter: 8 byte ID, 1 byte express origination address, values 8-223, 1 byte expresses module length, values 1-4. Write data for nn byte.

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

Rebound packet: if succeed, rebound data part is empty

□F0H 02H F2H 1CH□

Note: this command adopts to write data to tag by one byte and one byte, slow in rate, only used in the case when tag does not support no.5 writing instruction.

Command format with reader

Command code: F2H

Address parameter in reader

Command parameter: 8 byte in ID, 1 byte expresses origination address, values 8-223, 1 byte expresses, module length (nn), values1-4, written data of nn byte.

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

Rebound packet: if succeed, rebound data is empty

F0H 03H F2H address CheckSum

3.3.2 Read write EPC C1G2command format

Memory bank of IS08000-6Ctag divides into 4 areas.

A. EPC area: area of storing EPC code, stores 96bits EPC code at most, can read and write.

B. TID area: keep ID number set by tag manufacturer, 32 and 64Bits two type ID for now.

C. User area: this area is different for various manufacturer. G2 tag from Impinj company has no user area, company of Philips has 96 Bits, can write and read

D. Password area: has 32bits visit password and 32Bits kill password, can read and write

3.3.2.1 EPC1G2_ListTagID()

Function: Identify tag ID under radiation range of antenna, according to mask code condition.

Command code: EEH

Command parameter 1: 1 byte men, choose for data area.

0	password area
1	EPC code
2	IDnumber of TID tag
3	User

Command parameter 2: 2 byte, introduces origination of mask code.

Command parameter 3: 1 byte, introduces mask code length

Command parameter 4: m byte, mask□If LEN□8=0, then m=LEN/8□If LEN□8≠0, then m=⌊LEN/8⌋+1□

Command parameter 4: m byte, mask

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

Rebound data: If succeed, byte of rebound data part = list number of read tagM (1byte) + (transmitted tag numberL (<=8)*8(ID))*L (EPC digits+EPC))

Note: LEN=0expresses all tag ID can be identified under range of antenna radiation

EPC digit: 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

Address parameter in reader

Command parameter 1: 1 byte mem, choose for data area

0	password area
1	EPCcode
2	ID number in TID tag
3	User

Command parameter 2: 2 byte, introduces origination address of mask

Command parameter 3: 1 byte, introduces mask length LEN.

Command parameter 4: m byte, 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□

Rebound data: If succeed, byte of rebound data part = list number of read tagM (1byte) + (transmitted tag numberL (<=8)*8(ID))*L (EPC digits+EPC))

Note: LEN=0expresses that all identified ID under the range of antenna radiation

EPC digits: 00H-0Word, 01H-1Word, 02H-2Word, ……, FFH-256Word

□F0H 4+L*N EEH address M L*N CheckSum□

3.3.2.2 EPC1G2_GetIDList ()

Function: electronic tag ID got from reader EMS memory

Command code: EDH

Command parameter: 2byte. First byte is serial number of commence, second byte is tag number m (<=8)

Command packet: 40H 04H EDH no m CheckSum□ command packet

Rebound data: If succeed, digits of rebound data = (1byte tag number M*L Bytes (EPC digit+EPC)

□F0H 2+L*8 EDH L*M CheckSum□

Command format with reader address.

Command code: EDH

Address parameter with address

Command parameter: 2 byte. First byte is serial number of commence, second byte is tag number m (<=8)

Command packet:□40H 05H EDH address no m CheckSum□

Rebound data: If succeed, digit of rebound data part = (1byte tag numberM*L Bytes (EPC digit+EPC)

□F0H 3+L*8 EDH address L*M CheckSum□

3.3.2.3 EPC1G2_ReadWordBlock()

Function: read data from designate area of tag, data block is 16 bits in length, unit in word.

Command code: ECH

Command parameter 1: 1 byte EPC digits L, introduces word number for EPC

Command parameter 2: L*2byte EPC number, introduces to read which tag data

Command parameter 3:1 byte mem, choose data area

0	Password area
1	EPC number
2	IDnumber in TID tag
3	User

Command parameter 4: 1 byte origination address (Unite: word)

Command parameter 5: 1 byte data length len

Command parameter 6: 4 byte AcessPassword, password

command packet:40H 10+L*2 ECH L EPC mem addr len AccessPassword CheckSumcommand packet

Rebound data: If succeed, rebound data part is len*2byte data

F0H len*2+2 ECH xx …… xx CheckSum

Note: Access passworsjust works when password area is in password lock.

Command format with reader address

Command code: ECH

Address parameter in reader

Command parameter 1: 1 byte EP digit L, introduces word number of EPC

Command parameter2: L*2byte EPC number, introduces to read which tag data.

Command parameter 3: 1 byte mem, choose for data area

0	password area
1	EPCnumber
2	ID number in TID tag
3	User

Command parameter 4: 1 byte origination address (unit: word)

Command parameter 5: 1 byte data length (unit: word)

Command parameter 6: 4 byte AcessPassword, password

command packet:40H 11+L*2 ECH address L EPC mem addr len

AccessPassword CheckSum

Rebound data: if succeed, rebound data parties len*2 byte dataxx.

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

3.3.2.4 EPC1G2_WriteWordBlock()

Function: write data into tag; write data length is unit by word.

Command code: EBH

Command parameter 1: 1 byte EPC digits L, introduces word number for EPC code.

Command parameter 2: L*2 byte EPC number

Command parameter 3: 1 byte mem, choose data area MemBank

0	password area
1	No use
2	IDnumber in TID tag
3	User

Command parameter4:1byte origination address (unit word)

Command parameter 5: 1 byte data legth len

Command parameter 6: Len*2 byte data

Command parameter 7: 4 byte accesspassword, password

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

Rebound data: if succeed, rebound data is empty

F0H 02H EBH 23H

Note: Accesspassword is only efficient when data area is locked.when unlocked, it writes none password, when data is forever locked, password is useless.

Instruction format with reader address

Command code: EBH

Address parameter in reader

Command parameter 1: 1 byte EPC digits L, introduces word numbers of EPC code.

Command parameter 2: L*2byte EPC number, introduces to write data for which tag.

Command parameter 3: one byte mem, choose data area MemBank

0	Password area
1	EPC code
2	ID number in TID tag
3	User

Command parameter 4: 1 byte origination address.

Command parameter 5: 1 byte data length len(Unite:word)

Command parameter 6: len*2byte written data

Command parameter 7: 4 byte Access password, password

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

Rebound packet：If succeed, rebound data part is empty.

：F0H 03H EBH address CheckSum：

3.3.2.5 EPC1G2_SetLock()

Function: set write protection in designated area of tag

Command code: EAH

Command parameter 1: 1 byte of EPC digits L, introduces word numbers of EPC code

Command parameter2: L*2byte EPC code, introduces to set read and write protection for which tag.

Command parameter: 1 byte mem, choose protection area MemBank

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

Command parameter 4: 1 byte controlled word lock

0	Can write
1	Can write permanently
2	Write with code
3	Can not write permanently
4	can read write
5	Can read write permanently
6	Read write with code
7	Can not read write permanently

Password：0-3 is only for EPC, TID and User 3data areas, 4-7 is only for kill password and access password.

Command parameter 5: 4 byte access password, password.

command packet: 40H 9+L*2 EAH L EPC mem Lock AccessPassword CheckSum

Rebound data: If succeed, leading code in rebound packet is F0H, data part is empty.

F0H 02H EAH 24H

Instruction format with reader address

Command code: EAH

Address parameter in reader

Command parameter 1: 1 byte EPC digitsL, introduces word numbers of EPC code

Command parameter 2: L*2byte EPC code, introduces to set read-write protection for which tag

Command parameter 3: 1 byte mem, choose protection data area MemBank

0	Kill Password
1	Access Password
2	EPC code
3	IDnumber in TID tag
4	User

Command parameter 4: 1 byte control word lock

0	Can write
1	Can write permanently
2	Write with code
3	Can not write permanently
4	Can read write
5	Can read write permanently
6	Read write with code
7	Can not read write permanently

0-3is only for EPC, TID and User 3data areas, 4-7 is only for kill password and access password.

Command parameter 5: 4 byte AccessPassword

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

Rebound data: if succeed, leading code of rebound packet is F0H, data part is empty

F0H 03H EAH address CheckSum

3.3.2.6 EPC1G2_WriteEPC ()

Function: write EPC data into tag EPC unit, written data length is in unit of word

Command code: E7H

Command parameter 1: 1 byte EPCdigitsL, introduces word number of EPC code

Command parameter 2: L*2byte EPC code

Command parameter 3: 4 byte accesspassword

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

Rebound packet: if succeed, rebound data part is empty

F0H 02H E7H 27H

Note: Access password works only when data area is locked by password. When data not locked, can write none password, if data is forever locked, password is useless.

Command format with reader address

Command code: E7H

Address parameter in reader

Command parameter 1: 1 byte EPC digits L, introduces word number of EPC code.

Command parameter 2: L*2 byte EPC code,

Command parameter 3: 4 byte AccessPassword

command packet: 40H 8+L*2 E7H address L EPC AccessPassword CheckSum

Rebound packet: if succeed, rebound data part is empty

F0H 03H E7H address CheckSum

3.3.2.7 EPC1G2_changeeas ()

Function: replace for Eas of tag, works for UCODE EPC G2 tag of Philips only.

Command code: E5H

Command parameter 1: 1 byte EPC digits L, introduces word numbers of EPC code

Command parameter 2: L*2byte EPC number, introduces to replace for which tag.

Command parameter 3: state EASstate 1digit, 0-no alarm, 1-alarm

Command parameter 4: Access password 4 byte

command packet: 40H 8+L*2 E5H L EPC EASstate AccessPassword CheckSum

Rebound packet: if succeed, rebound data part is empty

F0H 02H E5H 29H

Command format with reader address

Command code: E5H command code

Address parameter in reader

Command parameter 1: 1 byte EPC digits L, introduces word number of EPC code.

Command parameter 2: L*2byte EPC code, introduces to replace for which tag.

Command parameter 3: state EASstate 1byte 0-no alarm, 1-alarm

Command parameter 4: AccessPassword 4 byte

Command packet: 40H 9+L*2 E5H address L EPC EASstate AccessPassword CheckSum

Rebound packet: If succees, rebound data is empty.

F0H 03H E5H address CheckSum

3.3.2.8 EPC1G2_EasAlarm()

Function: EAS replaced tag responds alarm, works for UCODE EPC G2 tag of Philips only.

Command code: E4H

None command parameter

Command packet:40H 02 E4H DAH

Rebound packet: If detected tag alarms, rebound data is empty.

F0H 02H E4H 2AH

If tag is not alarm, tag does not work for the instruction, rebound packet is:

F4H 03H E4H 02H 2AH

Data part is wrong information, did not detect tag

Instruction format with reader address

Command code: E4H

Address parameter in reader

None command parameter

Command packet: 40H 03H E4H address CheckSum

Rebound packet: If detected tag alarm, rebound data part is empty

F0H 03H E4H address CheckSum

If tag is not alarm, tag does not response for instruction, rebound packet is:

F4H 04H E4H address 02H CheckSum, Data part is wrong information, fail to detect tag.

3.3.2.9 EPC1G2_ReadProtect()

Function:operates read protection to specified tag, tag can not read EPC content after success. Only efficient to tag of Philips and UCODE G2XM

Command code: E3H

Command parameter 1: 4 byte AccessPassword

Command parameter 2: 1 byte EPCdigits L, L=1—6, introduces word number of EPC code

Command parameter 3: L*2byte EPC code, introduces to write data for which tag

Command packet:□40H 7+L*2 E3H AccessPassword L EPC CheckSum□

Rebound packet: If succeed, rebound data part is empty

□F0H 02H E3H 2BH□

Instruction format with reader address

Command code: E3H

Address parameter in reader

Command parameter 1: 4 byte

Command parameter 2: 1 byte EPC digitL, L=1—6, introduces word number of EPC code

Command parameter 3: L*2byte EPC code, introduces to write data for which tag

command packet:□40H 8+L*2 E3H address AccessPassword L EPC CheckSum□

Rebound packet: If succeed, rebound data part is empty

□F0H 03H E3H address CheckSum□

3.3.2.10 EPC1G2_RSTReadProtect()

Function: release from read protection to designated tag, tag can read EPC content

after success. Note:there is only one tag in field, works for UCODE G2XM tag of Philips

Command code: E2H command code

Command parameter 1: 4 byte AccessPassword

command packet:□40H 06 E2H AccessPassword CheckSum□

Rebound packet: If succeed, rebound data part is empty

□F0H 02H E2H 2CH□

Command format with reader address

Command code: E2H

Address parameter in reader

Command parameter 1: 4 byte AccessPassword

Command packet:□40H 07 E2H address AccessPassword CheckSum□

Rebound packet: If succeed, rebound data part is empty

□F0H 03H E2H address CheckSum□

3.3.3 Read write ISO18000-6D command format

ISO18000-6Dhas no data area, TK901has 8 pages of data area, 8 byte in each page, every page can be write protection individually. It can write once some page is written protection.

3.3.3.1 ISO18000-6D_ListID ()

Function: list all discernible tag under the range of tag radiation.

Command code: CEH

Command code: none command parameter

command packet:□40H 02H CEH F0H□

Rebound data: If succeed, rebound data part byte=list tag numbers of this readingM (1byte) + (sending tag number for this time L (<=8)*8(ID)

□F0H 3+L*8 CEH M L*8 CheckSum□

Command format with reader address

Command code: CEH

Address parameter in reader

None command parameter

Command packet: □40H 03H CEH address CheckSum□

Rebound data: If succeed, byte in rebound data part = list tag numbers of this readingM (1byte)+(sending tag number for this time L(<=8)*8(ID)

□F0H 4+L*8 CEH address M L*8 CheckSum□

3.3.3.2 ISO18000-6D_ListIDReport()

Function: get electronic tag ID (passed rfs-listID command) from reader EMS

memory

Command code: CDH

Command parameter 1: 1 byte is origination serial number ADDR

Command parameter 1: 1 byte is tag numberM (<=8)

40H 04H CDH ADDR M CheckSum

Rebound data: If succeed.byte in rebound data= (tag number*8(ID)

F0H 2+M*8 CDH M*8 CheckSum

Command format with reader address

Command code: CDH

Address parameter in reader

Command parameter 1: 1 byte is origination serial number

Command parameter: 1 byte is tag numberM (<=8)

Command packet: 40H 05H CDH address ADDR M CheckSum

Rebound data: If succeed, byte in rebound data= (tag number*8(ID)

F0H 3+M*8 CDH address M*8 CheckSum

3.3.3.3 ISO18000-6D_ReadPageBlock()

Function: to read one piece of data in designated tag. Length of data area is in unit of page (8bytes). TK901has 15 pages data area, the final page is data of write protection. All pages can read, the final page can not read, can read one pag each

time.

Command code: CCH command code

Command parameter 1: 8 byte ID

Command parameter 1: 1 byte expresses origination address AA, values0-7

Command parameter 1: 1 byte expresses data length nn (unit page), values 1-8.

Command packet:40H 0CH CCH ID aa nn CheckSum

Rebound packet: if succeed, rebound data part is data in nn byte.

F0H nn+2 CCH xx …… xx CheckSum

Command format in reader address

Command code: CCH

Address parameter in reader

Command parameter 1: 8 byte ID

Command parameter 1: 1 byte expresses origination address aa (unit: page), values 0-7

Command parameter 1: 1 byte expresses data length nn(unit:page), values 1-8.

Command packet:40H 0DH CCH address ID aa nn CheckSum

Rebound packet: If success, rebound data part is data of nn byte

F0H nn+3 CCH address xx …… xx CheckSum

3.3.3.4 ISO18000-6D_WritePageBlock()

Function: write data to specified area in designated tag, written data length is in unit if page, 1 page at most for once.

Command code: CBH

Command parameter: 8 byte ID, 1byte expresses origination address aa (unit: page), values 1-7, written data of 8 byte

Command packet:□40H 13H CBH id aa xx --- xx CheckSum□

Rebound packet: If succeed, rebound data part is empty

□F0H 02H CBH 43H□

Command format with reader address.

Command code: CBH command code

Address parameter in address

Command parameter: 8 byte ID (ID compositor in tag is standard). 1byte expresses origination aa (unit: page), values1-7,8byte of written data.

Command packet:□40H 14H CBH address id aa xx --- xx CheckSum□

Rebound packet: If succeed, rebound data part is empty.

□F0H 03H CBH address CheckSum□

3.3.3.5 ISO18000-6D_SetProtect()

Function: set appointed area in specified tag to be written protection

Command code: CAH

Command parameter 1: 8 byet ID

Command parameter1:1byte expresses origination address aa (unit: page), values1-7

Command parameter 1: 1 byte expresses data length nn (unit: page), values1-7

Command packet:40H 0CH CAH ID aa nn CheckSum

Rebound data: If succeed, leading code of rebound packet is F0H, data part is empty

F0H 02H CAH 44H

Command format with reader address

Command code: CAH

Address parameter in reader

Command parameter 1: 8 byte ID

Command parameter 1: 1 byte expresses origination address aa (unit: oage), values 1-7

Command parameter 1: 1 byte expresses data length nn (unit: page), values1-7

Command packet: 40H 0DH CAH address ID aa nn CheckSum

Rebound data: If success, leading code in rebound packet is F0H, data part is empty

F0H 03H CAH address CheckSum

3.3.3.6 ISO18000-6D_GetProtect()

Function: Read specified address unit in designated tag if written protection

Command code: C9H

Command parameter 1: 8 byte ID

Command parameter 1: 1 byte expresses origination address aa =0FH

Command parameter 1: 1 byre expresses data length nn=1

Command packet: 40H 0CH C9H ID 0FH 01H CheckSum

Rebound data: If succeed, leading code in rebound packet is F0H, data part divides 2 byte a and b.

F0H 04H C9H a b CheckSum

Byte b from LSB to MSB expresses 0-7 data page, byte a from LSB to MSB-1 expresses 8-14 data page, each Bit=0 expresses this page is unlocked, Bit=1 expresses this page is locked.

Command format with reader address

Command code: C9H

Address parameter in reader

Command parameter 1: 8 byte ID

Command parameter 1: 1 byte expresses origination address aa =0FH

Command parameter 1: 1 byte expresses data length nn=1.

Command packet:40H 0DH C9H address ID 0FH 01H CheckSum

Command packet

Rebound data: If succeed, leading code in rebound packet is F0H, data part divides 2 byte a and b.

F0H 05H C9H address a b CheckSum

Byte b from LSB to MSB expresses 0-7 data page, MSB of byte a expresses 15th systematic page, each bit=0 expresses this page is unlocked, bit-=1 expresses this page is locked.

3.4 Collection of operation command

3.4.1 EPC Class1Gen2command

Serial number	command	function
1	EEH	To identify tag ID under radiation range of antenna by mask conditions
2	EDH	to get listed electronic tag ID from reader memory
3	ECH	To read block data in appointed data area of appointed tag
4	EBH	to write data in appointed data area of tag
5	EAH	Set appointed data area to be write protection in appointed tag
6	E7H	write EPC data into EPC unit of tag
7	E5H	reboot EAS state of tag
8	E4H	EAS SET tag response for alarm
9	E3H	to do read protection for appointed tag
10	E2H	To do release read protection for appointed tag

3.4.2 ISO18000-6Bcommand

Serial number	command	function
1	FEH	list readable tag ID under the range of antenna radiation
2	FDH	to read electronic tag ID from memory of reader
3	FBH	list readable tag ID in the range of antenna radiation according to parameters as follows
4	F6H	To read block data of appointed tag
5	F5H	Write data in address of appointed tag
6	F4H	Set appointed address unit in tag to be write protection
7	F3H	To read if appointed address of tag write protection
8	F2H	Write data to appointed address unit of tag

3.4.3 ISO18000-6Dcommand

Serial number	command	function
1	CEH	list readable tag ID under the range of antenna radiation
2	CDH	to get listed electronic tag ID from reader memory
3	CCH	To read block data in appointed address of tag
4	CBH	write data to appointed address unit of tag
5	CAH	set address unit in appointed tag to be write protection
6	C9H	To read if appointed address of tag write protection

3.4.4 Other command

Serial number	command	function
1	01H	set operation board rate for RS232 interface
2	02H	get version numbers of hardware and software in reader
3	03H	set relay state in reader
4	04H	set launch power coefficient in reader
5	05H	set frequency tunnel number of microwave signal lunched by reader.
6	06H	To read operation parameter from last command
7	09H	set basic operation parameter of board rate,launch frequency,output power in reader
8	0AH	choose to receive and send signal from which antenna
9	0EH	reader reboot
11	11H	set reader time
12	12H	to get reader time
13	13H	increase new list based on original one in reader
14	14H	delete new list based on original one in reader

15	15H	to read stored white list in reader
16	16H	to read stored tag record in reader
17	17H	delete all tag record stored in reader
18	30H	set reader network address
19	31H	to get network address in reader
20	32H	set reader network MAC
21	33H	get reader network MAC

3.5 Electronic tag storage area and notes

1. Memory Bank of EPC Class1 Gen2 tag divides to be 4 areas.

   - EPC area (EPC): Storage area of EPC code leaves 96 Bits EPC code at most for now, can read and write.

   - TID area: Keep sett ID numbers by tag manufacturer, there are 32 and 64Bites two kinds for now, can read, can not write.

   - User: different area for different manufacturer,G2 tag of Impinj company has no user. NXP company has 96 Bits, can read, can write.

   - Password: has 32Bits access password, and 32 bits kill password. Can read, can write, can make different protection for these two areas.

EPC Class1 Gen2tag can set different protection mode for different storage area, protection mode of each storage area is of 4 types:

- EPC, TID and user of G2 tag.

Read in EPC, TID and user area of G2tag is not protected, write protection function

Writeable from any state---can write none accesspassword,can set password lock or permanent write or permanent lock.

Permanently writable---can wite none accesspassword, and can not be passwordlocked or permanent locked.

Writable from secured state---can write in the case of know accesspassword

Never writable---can not write even know password.

Read and write in password area of G2 tag can be protected, read-write protection state in password area does not affect usage of password, and can put protection function to these 2 areas.

Readable and Writable from any state---can read and write none accesspassword, can be password secured or permanent read write or permanent secured.

Can read and write none accesspassword, and can not be password secured later.

In the case of knowing password, can read and revise password, can set to be permanent secured or permanent read-write later

Can not read or revise password, even know it.

Note: set tag read and write protection, must know tag accesspassword.

2. Memory bank in the tag of ISO-18000-6B divides to two area, storage capacity inside is 2048bits, and divides to be 256 byte. There is one address for each byte, 0-255 in correspondence.

   - Address 0-7 eight byte (64bits): is tag ID numbers, solidify before products come out, can not be revised.

   - Address 8-233 user information can be left in user area, can self-distributed according to details, can be revised and locked, but can not revise once locked and unlocked.

- Address 224-255 writes protection byte.

4 SDK software development

4.1 SDK compose

Package in VI-85 module provides SDK; it mainly composes with files as follows:

A. Reader1000DLL.dll file --- dynamic connection
B. Reader1000DLL.lib file --- state connection
C. Reader1000API.h file ---State file in API function
D. Reader1000SDK catalogue---Including example procedure of learning API function

4.2 Design introduction

4.2.1 Basic constant and figure

4.2.1.1 Constant definition

description		introduction
#define ID_MAX_SIZE_64BIT	8	ID numbers in electronictag is 64bit
#define ID_MAX_SIZE_96BIT	13	ID numbers in electronictag is 128bit
#define MAX_LABELS	100	Can not be more than 100 tags for one time of read write operation

4.2.1.2 APIfunction rebound code

#define_OK	0x00	// operation success
wrong information in communictation		
#define _init_rs232_err	0x81	//initialization failed in

		communication interface
#define _no_scanner	0x82	//can not find reader
#define _comm_error	0x83	//wrong in send and receive communication data
#define _baudrate_error	0x84	// mistake in set baud rate
// rebound mistake information from reader		
#define _no_antenna	0x01	//fail to connect antenna
#define _no_label	0x02	//fail to detect tag
#define _invalid_label	0x03	//llegal tag
#define _less_power	0x04	//read write power is not enough
#define _write_prot_error	0x05	//area write protection
#define _check_sum_error	0x06	//adjustment and mistake
#define _parameter_error	0x07	//parameter wrong
#define _memory_error	0x08	//data area not exsit
#define _password_error	0x09	//password not correct
#define _killpassword_error	0x0a	//G2tag kills password to be 0
#define _nonlicet_command	0x0b	//illegal command
#define _nonlicet_user	0x0c	//illegal user with unmatched password
#define _invalid_command	0x1e	// Inefficient command,say,command of wrong parameter
#define _other_error	0x1f	//unknown command
//function input wrong		
#define _no_cardID_input	0x20	//other mistake

4.2.1.3 Data type definition

Typedef USHORT apiReturn; // rebound value type of function

One type of apiReturn value will be rebound after all API function executed, can judge if function executed successfully from the value, if failure, what is failure reason, and so on…

4.2.1.4 Parameter figure in reader

Typedef struct tagReader1000Param

{

BYTE BaudRate; //Communiction rate of Serial port,values 00H-08H,represents rate and command.

BYTE Power; //Launch power, values 30-160.

BYTE Min_Frequence; //

Origination of sending microwave signal frequency,values1-63

BYTE Antenna; //1-ant1,2-ant2,4-ant3,8-ant4

Antenna options.

BYTE WorkMode; //

Work mode of reader: 1. Timing mode, 2. Spring mode, 3. Command mode, 4. Timing mode, 5. Spring mode

BYTE ReadInterval; // 0-10ms, 1-30ms, 2-50ms

Interval of read

BYTE OutMode; //

Output mode of card number.

BYTE TriggerMode; //Spring mode

BYTE IDPosition; //

Deposit address of card number in electronic tag.

BYTE IfTestValidity; //If tag is legal

BYTE OutInterface; //(Wiegan26, Wiegan34, RS485, RS232,

RJ45)output interface and format of card number

BYTE NumofCard; //Most amount in card reading

BYTE Power2; //

Launch power coefficient2, values: 30-160

BYTE TagType; //01H-ISO18000-6B, 02H-EPCC1, 04H-ISO18000-6C, 08H-ISO18000-6D

Tag types

BYTE WiegandWidth; //Weigand Value of Impulse width

BYTE WiegandInterval; //Weigand Value of impulse interval

BYTE ID_Start; //

Origination of output card number, values 0-4

BYTE Max_Frequence; //

Ending origination in transmitting microwave signal frequency, values1-63

BYTE ReadDuration; // RF emission duration, just directed at tag ISO18000-6D 10ms, 1-20ms, 2-30ms, 3-40ms

RF emission duration, just directed at tag ISO18000-6D

BYTE StandardTime; //Standard output interval, default is 120s,1-255

BYTE EnableBuzzer; //Enable buzzer0: unable buzzer

```
    BYTE    ReaderAddress;      //
```

Reader address0-255,0and 255 is address of broadcast

```
    BYTE    HostIP1;            //
```

Epigyny machine IP address

```
    BYTE    HostIP2;            // Epigyny machine IP address
    BYTE    HostIP3;            // Epigyny machine IP address
    BYTE    HostIP4;            // Epigyny machine IP address
BYTE    HostPort1;          // Epigyny machine IP address
```

Epigyny machine interface

```
    BYTE    HostPort2;          // Epigyny machine IP address
    BYTE    Reserve29;          // Reservation
    BYTE    Reserve30;          // Reservation
    BYTE    TX_Mode;            //
```

Emission mode: 0expresses receive and emission mode, 1expresses emission mode

```
    BYTE    Modulation;         //Confection set: 0expresses no confection
```

signal, 1 expresses with confection signal

```
} Reader1000Param;
```

4.2.1.5 Function rebound code

When commands fail to be executed, function rebound with wrong code.

Usual wrong code:

command	function
00(00H)	command succeed or test correctly
01(01H)	Antenna fail to connect
02(02H)	fail to test tag
03(03H)	illegal tag
04(04H)	read power is not enough
05(05H)	this area read write protection
06(06H)	adjustment and mistakes
07 (07H)	parameter mistake
08 (08H)	data area non-exist
09 (09H)	wrong password
10 (0AH)	destroyed password is 0
11(0BH)	when reader is positive in work,it receives AutoMode and Reboot command,other command is illegal command
12(0CH)	illegal user ,unmatched with password
13(0DH)	expresses RF interference from outside
14 (0EH)	expresses tag read protection
……	……
30(1EH)	expresses invalid command,wrong command of parameter
31(1FH)	unknown command
32(20H)	other errors

4.2.2 Control command function

4.2.2.1 Connect reader

Connection by serial port

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

Function: establish communication connection with reader, and set communication rate

Input parameter

SzPort: Character pointers directed at communication port, eg. COM1, COM2……

nBaudRate: Baurd rate of Serial port communication, efficient communication rates: 9600, 19200, 38400, 57600, 115200

Output parameter

Judge if connect successfully according to rebound apiReturn value in function, or failure reason.

Hscanner: reader handle

Command with reader address

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

Function: establish communication connection with reader and set communication rate.

Input parameter

szPort : word pointer, directed at communication port, say: serial port COM1COM2……

nBaudRate: Baurd rate in communication of serial port, efficient communication rate has: 9600, 19200, 38400, 57600, 115200

Address: Reader address

Input parameter: judge if connection succeed according to apiReturn value

of rebound function, or failure reason

hScanner: reader handle

Connection by network:

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

Function: establish communication connection with reader and set communication rate.

Input parameter

nTargetAddress:192.168.0.1.....

nTargetPort:1969

nHostAddress:192.168.0.2.....

nHostPort:eg.5000

Output parameter

hSocket: reader communication handle

Return: If rebound value fo function is OK, then expresses connection is successful or connection failure.

Note: execute this command to direct at each reader, to get correspondent reader hSocket

4.2.2.2 Disconnection

apiReturn DisconnectScanner(HANDLE hScanner)

apiReturn Net_DisconnectScanner();

Function: close connection with reader, release serial port resources

Input parameter

hSacnner: reader communication handle

4.2.2.3 Set baud rate

apiReturn _stdcall SetBaudRate(HANDLE hScanner, int nBaudRate,int RS485Address)

apiReturn Net_SetBaudRate(SOCKET hSocket, int nBaudRate);

Function: set operation baurd rate of RS232port

Input parameter

hSacnner/hSocket: reader communication handle

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

RS485Address:

Reader RS485 net address, RS485 Address=0 expresses no net.

Rebound: If rebound value of function is OK, expresses set successfully or it is failure reason.

4.2.2.4 Read version

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

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

Function: read version numbers of reader hardware and software

Input parameter

hSacnner/hSocket:reader communication handle

RS485Address: Reader RS485 net address, RS485 Address=0expresses no net

Output parameter

wHardVer: Hardware verion number in reader

WSoftVer: Software version number in reader

Rebound: if rebound value of function is OK, expresses read successfully or read failure

4.2.2.5 Set output power

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

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

Function: set reader output power

Input parameter

hSacnner/hSocket: reader communication handle

nPower: output power

RS485Address: Reader RS485 net address, RS485 Address=0expresses none team net

Rebound: if rebound value of function is OK, expresses read successfully,or set in failure.

4.2.2.6 Set operation frequency

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

apiReturn Net_SetFrequency(SOCKET hSocket, int Min_Frequency, int Max_Frequency, BYTE * IPaddress);

Function: set operation frequency for present reader.

Input parameter

hSacnner/hSocket: reader coummunication ahndle

Min_Frequency:

Reader origination frequency, values1-63

Max_Frequency: Reader ending frequency, values 1-63.

When Min_Frequency = Max_Frequency, reader works frequently.

RS485Address: Reader RS485 net address, RS485 Address=0expresses none team net

Rebound: if rebound value of function is OK,expresses read successfully or set in failure.

4.2.2.7 Set reader operation parameter

apiReturn ReadParam(HANDLE hScanner, Reader2200Param * pParam,int Address)

apiReturn Net_ReadParam(SOCKET hSocket, Reader2200Param * pParam);

Function: from reader to read operation parameter from last command.

Input parameter

hSacnner/hSocket: reader communication handle

RS485Address: Reader RS485 net address, RS485 Address=0expresses none team net

Output parameter

pParam: Rebound operation parameter in reader, 32byte.

Rebound: if rebound value of function is OK, expresses read successfully or set in failure.

4.2.2.8 Set reader operation parameter

apiReturn WriteParam(HANDLE hScanner, Reader2200Param * pParam, int Address)

apiReturn Net_WriteParam(SOCKET hSocket, Reader2200Param * pParam);

Function: set reader operation parameter

Input parameter

hSacnner/hSocket: reader communication handle

pParam: operation parameter in reader, 32byte

RS485Address:

Reader RS485 net address, RS485 Address=0expresses none team net

Rebound: if rebound value of function is OK, expresses read successfully or set in failure.

4.2.2.9 Antenna option

apiReturn SetAntenna(SOCKET hSocket, int Antenna,BYTE * IPaddress);

apiReturn Net_SetAntenna(SOCKET hSocket, int Antenna);

Function: choose to receive and send signal by which antenna.

Input parameter

hSacnner/hSocket: reader handle

Antenna:

Antenna numbers, 1-1antenna, 2-2antenna, 4-3antenna, 8-4antenna

RS485Address: Reader RS485 net address, RS485 Address=0expresses none team net

Output parameter

IPaddress: Rebound IP address in reader

Rebound: if rebound value of function is OK, expresses read successfully or set in failure.

4.2.2.10 Set relay state in reader

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

apiReturn Net_SetRelay(SOCKET hSocket, int Relay);

Function: set relay state of reader

Input parameter

hSacnner/hSocket: reader handle

Relay: Relay: 1byte. Bit0=1 is for close of relay number1, Bit1=0 is disconnection of number 1 relay, Bit1=1 is for close of number 2 relay, Bit1=0 is for disconnection of relay number 2 analogy.

RS485Address: Reader RS485 team net address, RS485Address=0 expresses no team net

Rebound: if rebound value of function is ok, expresses read successfully or set in failure.

4.2.2.11 Reboot reader

apiReturn Reboot(HANDLE hScanner, int Address);

apiReturn Net_Reboot(SOCKET hSocket);

Function: replace reader, same to electrify for reader.

Input parameter

hSacnner/hSocket: reader communication handle

S485Address: Reader RS485 team net address, RS485Address=0 expresses no team net

Rebound: if rebound value of function is OK, expresses read successfully or set in failure.

4.2.2.12 time set

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

apiReturn Net_SetReaderTime(SOCKET hSocket, ReaderDate time)

Function: in time of epigyny, set reader time

Input parameter

hSacnner/hSocket: reader communication handle

Time: time of epigyny, 6 byte

RS485Address: Reader RS485 team net address, RS485Address=0 expresses no team net

Rebound: if rebound value of function is OK, expresses read successfully or

set in failure.

4.2.2.13 gain time

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

apiReturn GetReaderTime(SOCKET hSocket, ReaderDate *time)

Function: read time of reader.

Input parameter

hSacnner/hSocket: reader communication handle

RS485Address: Reader RS485 team net address, RS485Address=0 expresses no team net

Output parameter

Time: time to rebound reader, 6byte

Rebound: if rebound value of function is OK, expresses read successfully or set in failure.

4.2.2.14 append list

apiReturn AddLableID(HANDLE hScanner, int listlen, int datalen, BYTE * data)

apiReturn AddLableID(SOCKET hSocket, int listlen, int datalen, BYTE * data)

Function: to append white list in reader

Input parameter

hSacnner/hSocket: reader communication handle

Listlen: additive list number

Datalen: length of each list

Data: Append list

Rebound: if rebound value of function is ok,expresses read successfully or set in failure.

4.2.2.15 delete list

apiReturn DelLableID(HANDLE hScanner, int listlen, int datalen, BYTE * data)

apiReturn DelLableID(SOCKET hSocket, int listlen, int datalen, BYTE * data)

Function: delete appointed list in reader

Input parameter

hSacnner/hSocket: reader communication handle

Listlen: list

Datalen: length of each list

Data: Appointed list to delete

Rebound: if rebound value of function is ok, expresses read successfully or

set in failure.

4.2.2.16 Get list

apiReturn GetLableID(HANDLE hScanner, int startaddr, int listlen, int *relistlen, int *taglen, BYTE * data)

apiReturn GetLableID(SOCKET hSocket, int startaddr, int listlen, int *relistlen, int *taglen, BYTE * data)

Function: to read list in reader

Input parameter

hSacnner/hSocket: reader coummunication handle

Startaddr: origination list

Listlen: list numbers to read

Output parameter

Relistlen: actual read list

Taglen: length of each list actual read

Data: Read list

Rebound: if rebound value of function is OK, expresses read successfully or set in failure.

4.2.2.17 Gain record

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

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

Function: read identified tag record in reader

Input parameter

hSacnner/hSocket: reader communication handle

stime: starting time

etime: ending time

startaddr: start record

listlen: records to read

Output parameter

relistlen: record numbers in actual read

taglen: length of each record in actual read

data: read cord

Rebound: if rebound value of function is OK, expresses read successfully or set in failure.

4.2.2.18 Delete all records

apiReturn DeleteAllRecord(HANDLE hScanner)

apiReturn DeleteAllRecord(SOCKET hSocket)

Function: delete all recordsin reader

Input parameter

hSacnner/hSocket: reader communication handle

Rebound: if rebound value of function is OK, expresses read successfully or set in failure.

4.2.3 Network command

4.2.3.1 Set IP address for reader

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

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

Function: set network IP address in reader

Input parameter

hSacnner/hSocket: handle in reader communication port

IP_Address[4]: reader IP address

Port: interface number of network in reader

Mask[4]:

Network IP address mask in reader.

Gateway[4]: Gateway in reader

Rebound: if rebound value of function is OK, expresses read successfully or set in failure.

4.2.3.2 Gain IP address in reader

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

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

Function: to get reader network IP address.

Input parameter

hSacnner/hSocket: handle in communication port of reader

Output parameter

IP_Address[4]: reader IP address

Port: network port number in reader

Mask[4]: mask of network IP address in reader

Gateway[4]: Gateway in reader

Rebound: if rebound value of function is OK, expresses read successfully or set in failure.

4.2.3.3 Set reader MAC address

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

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

Function: set network MAC address in reader

Input parameter

hSacnner/hSocket: handle in communication port of reader

MAC[6]: network MAC address in reader

Rebound: if rebound value of function is OK,expresses read successfully or set in failure.

4.2.3.4 Get reader MAC address

apiReturn GetReaderMAC(HANDLE hScanner, BYTE *MAC);

apiReturn Net_GetReaderMAC(SOCKET hSocket, BYTE *MAC);

Function: to get network MAC address in reader

Input parameter

hSacnner/hSocket: handle in communication port of reader

Output parameter

MAC: Network MAC address in reader

Rebound: If rebound value of function is OK, it expresses that set

successfully or it is failure reason.

4.2.4 Read write ISO18000-6B function

4.2.4.1 Identify tag IDnumber

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

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

Function: to read all electronic ID numbers under the range of antenna radiation.

Input parameter

hSacnner/hSocket: handle in communication port of reader

Output parameter

nCounter: return tag numbers that ID numbers are read

IDBuffer: Storage of read tag ID numbers

RS485Address: Reader RS485 team net address, RS485Address=0 expresses no team net

Rebound: If rebound value of function is OK, it expresses that set successfully, or it is failure reason.

4.2.4.2 Identify selected tag ID number

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

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

Function: to identify ID numbers of optioned electronic tag, under the range of antenna radiation.

Input parameter

hSacnner/hSocket: handle in communication port of reader

Cmd: Conditions of optioned tag

00	equal
01	unequal
02	Than
03	Less than

Ptr: origination address of tag data, values range0-223.

Mask: Data mask, each bit in the byte is correspondent to a comparable byte.0 expresses this byte is not for comparison, 1expresses that the byte is for comparison

Data: comparision data

RS485Address: Reader RS485 team net address, RS485Address=0 expresses no team net

Output parameter

nCounter: return tag numbers of ID read

IDBuffer: Read ID number of tag deposit

Rebound: If rebound value of function is OK,it expresses that set successfully or it is failure reason.

4.2.4.3 Read data block

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

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

Function: read a section data of EMSmeomory on electronic tag

Input parameter

hSacnner/hSocket: handle in communication port of reader

IDBuffer: desired to read ID numbers on tag.

ptr: read origination address of tag memory(0223 Byte)

Read start address in MES memory bank of tag

len: data block length, how many byte read in once

RS485Address: Reader RS485 team net address, RS485Address=0 expresses no team net

Output parameter rebound read data

Note: nLen must be ≤ 32.(nAddress+nLen) ≤223▫

Rebound: If rebound value of function is OK, it expresses that set successfully or it is failure reason.

4.2.4.4 write data block

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

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

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

Input parameter

hSacnner/hSocket: handle in communication port of reader

IDBuffer: to wrote tag ID numbers

ptr: write start address of EMS memory in tag.

Len: data block length,how many bytes to wirte once.

Data: to write data

Note:ptr should be integral times of 4 (nAddress+nLen) ≤223▫

RS485Address: Reader RS485 team net address, RS485Address=0 expresses no team net

Rebound: If rebound value of function is OK, it expresses that set successfully or it is failure reason.

4.2.4.5 Slow write data block

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

apiReturn Net_ISO6B_WriteAByteBlock(SOCKET hSocket, BYTE *IDBuffer, BYTE ptr, BYTE len, BYTE *Data);

Function: write data to appointed address in tag by byte and byte.

Input parameter

hSacnner/hSocket: handle in communication port of reader

IDBuffer: to write tag ID numbers

ptr: write start address of tag EMS memory(8-223)

len: data length, write how many bytes in once.

Data: desired to wirte data

note: (nAddress+nLen) ≤223□

RS485Address: Reader RS485 team net address, RS485Address=0 expresses no team net

Rebound: If rebound value of function is OK, it expresses that set successfully or it is failure reason.

Note: The command adopts to write data into tag by byte and byte,

slow in rate. Just use when tag does not support previous write instruction.

4.2.4.6 WriteProtect

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

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

Function: set write protection of appointed address unit in appointed tag

Input parameter

hSacnner/hSocket: handle in communication port of reader

IDBuffer: to write tag ID numbers

Ptr: to place write protection EMS memory address of tag

RS485Address: Reader RS485 team net address, RS485Address=0 expresses no team net

Rebound: If rebound value of function is OK, it expresses that set successfully or it is failure reason.

4.2.4.7 ReadWriteProtect

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

apiReturn Net_ISO6B_ReadWriteProtect(SOCKET hSocket, BYTE *IDBuffer,

BYTE ptr, BYTE *Protected);

Function: read appointed address unit of appointed tag if write protection

Input parameter

hSacnner/hSocket: handle in communication port of reader

IDBuffer: to write tag ID numbers to read protected EMS memory address in tag (0-223)

RS485Address:

Reader RS485 team net address, RS485Address=0 expresses no team net

Output parameter

Protected: Protective dtate, 0-no protection, 1-protected

Rebound: If rebound value of function is OK, it expresses that set successfully or it is failure reason.

4.2.5 Read write EPC C1G2 function

4.2.5.1 Identify EPCnumbers of EPC1G2 tag

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

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

Function: read all EPC numbers in correspondence with identified electronic tag under the range of antenna radiation.

Input parameter

hSacnner/hSocket: handle in communication port of reader

mem: choose data area

0	password area
1	EPC number
2	IDnumber in TID tag
3	User

ptr: start address of mask(Unit:Bit)

len: length of mask(Unit:Bit)

Mask, if len/8 is integer, then length of mask is len/8, if len/8 is not integer, mask length is len/8+1, final byte data in mask should be in high, low is 0

RS485Address:

Reader RS485 team net address, RS485Address=0 expresses no team net.

Output parameter

IDBuffer: Read EPC code in tag

NCounter: Read numbers of tag

Rebound: If rebound value of function is OK, it expresses that set successfully or it is failure reason.

Note: LEN=0expresses all recognizable ID of tag under the range of antenna radiation.

4.2.5.2 Read a block data

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

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

Function: read data in EMS memory of electronic tag

Input parameter

hSacnner/hSocket: handle in communication port of reader

EPC_WORD: EPClength L (Unit: word); say 96Bits EPC lengthL=6(words)

IDBuffer: selected tag EPC code

mem: Choose data area: 0-password area, 1-EPC code, 2-TID tag ID number, 3-user

ptr: read start address(Unit :word)

len: length of read(Unit :word)

AccessPassword: 4 byte AccessPassword

RS485Address:

Reader RS485 team net address, RS485Address=0 expresses no team net.

Output parameter

Data: read data

Rebound:If rebound value of function is OK, it expresses that set successfully or it is failure reason.

Note: AccessPasswors works for password only when it is password locked

4.2.5.3 Write a block data

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

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

Function: write data to appointed address unit in tag

Input parameter

hSacnner/hSocket: handle in communication port of reader

EPC_WORD: EPC lengthL(UNIT:word), say 96Bits EPC lengthL=6(Words)

IDBuffer: EPC code of optioned tag

mem: choose for data area

0	password area
2	IDnumbers in TID tag
3	User

Ptr: to write start address(unite:WORD)

len: to write length(unite:WORD)

Data: to write data

AccessPassword: 4 byte AccessPassword

RS485Address:

Reader RS485 team net address, RS485Address=0 expresses no team net.

Rebound: If rebound value of function is OK, it expresses that set successfully or it is failure reason.

Note: AccessPassword works only when data area is in password locked. Writes none password when data is unlocked. Password is useless when data is permanently locked.

4.2.5.4 Set readwrite protection state

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

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

Function: set appointed data area in tag to be write protection.

Input parameter

hSacnner/hSocket: handle in communication port of reader

EPC_WORD: EPC□□L(□□:Word)□□96BitsEPC□□L=6(Words)□

EPC lengthL(Unit:Word), say 96Bits EPC length L=6(Words)

IDBuffer: selected EPC number of tag

mem: choose for data area

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

Lock: Control word lock

0	can write
1	can write permanently
2	Write with password
3	can not write permanently
4	can read write
5	can read write permanently
6	read write with password
7	can not read write permanently

Note: 0-3 is only for PEC,TID and user area, 4-7 is only for kill password and access password.

AccessPassword: 4 byte AccessPassword

RS485Add Reader RS485 network building address, RS485Address=0 expresses no network building .

Rebound: If rebound value of function is OK, it expresses that set successfully or it is failure reason.

4.2.5.5 write EPC number

apiReturn EPC1G2_WriteEPC(HANDLE hScanner, BYTE len, BYTE *Data, BYTE *AccessPassword, int Address)

apiReturn Net_EPC1G2_WriteEPC(SOCKET hSocket,BYTE len, BYTE *Data, BYTE *AccessPassword);

Function: to write EPC data into tag EPC unite

Input parameter

hSacnner/hSocket: handle in communication port of reader

len: EPClength L(Unite:word), say 96Bits EPC length L=6(Words)

Data: to write EPC data

AccessPassword: 4 byte AccessPassword

RS485Address:

RS485Add Reader RS485 network building address,RS485Address=0 expresses no network building .

Rebound: If rebound value of function is OK, it expresses that set successfully or it is failure reason.

Note: AccessPassword works only when data area is in password locked. Writes none password when data is unlocked. Password is useless when data is permanently locked.

4.2.5.6 EASstate operation command

apiReturn EPC1G2_ChangeEas(HANDLE hScanner, BYTE EPC_WORD, BYTE *IDBuffer, BYTE State, BYTE *AccessPassword, int Address)

apiReturn Net_EPC1G2_ChangeEas(SOCKET hSocket, BYTE EPC_WORD, BYTE *IDBuffer, BYTE State, BYTE *AccessPassword);

Function: to replace for Easy state of tag, is for Philips and UCODE EPC G2 tag only

Input parameter

hSacnner/hSocket: handle in communication port of reader

EPC_WORD: EPC length L(Unit:word), say 96Bits EPC length L=6(Words)

IDBuffer: optioned EPC code of tag

State: Alert state

0	Not alert
1	alert

AccessPassword: 4 byte AccessPassword, whatever accesspassword of tag is 0 or not, have to fill

RS485Address:

RS485Add Reader RS485 network building address, RS485Address=0 expresses no network building .

Rebound: If rebound value of function is OK, it expresses that set successfully or it is failure reason.

4.2.5.7 EAS alert command

apiReturn EPC1G2_EasAlarm(HANDLE hScanner, int Address)

apiReturn Net_EPC1G2_EasAlarm(SOCKET hSocket);

Function: tag of EAS replace responses this alarm test instruction. Works for UCODE EPC G2 tag of Philips.

Input parameter

hSacnner/hSocket: handle in communication port of reader

RS485Address:

RS485Add Reader RS485 network building address, RS485Address=0 expresses no network building .

Return: if rebound value of function is OK, then tag alerts or no alert.

4.2.5.8 Set read protection (EPC1G2_ReadProtect)

apiReturn _stdcall EPC1G2_ReadProtect(HANDLE hScanner,BYTE *AccessPassword, BYTE EPC_WORD, BYTE *IDBuffer, int Address)

apiReturn Net_EPC1G2_ReadProtect(SOCKET hSocket,BYTE *AccessPassword, BYTE EPC_WORD, BYTE *IDBuffer)

Function: to do read protection for appointed tag, tag can not read actual EPC content after success. Works for UCODE G2XM tag of Philips

Input parameter

hSacnner/hSocket: handle in communication port of reader

AccessPassword: tag accesspassword

EPC_WORD: introduces word numbers of EPC code

IDBuffer:EPC, introduces to write data for which tag

RS485Address:

RS485Add Reader RS485 network building address, RS485Address=0 expresses no network building.

Rebound: If rebound value of function is OK, it expresses that set successfully or it is failure reason.

4.2.5.9 release read protection (EPC1G2_RSTReadProtect)

apiReturn EPC1G2_RStreadProtect(HANDLE hScanner, BYTE *AccessPassword,int Address)

apiReturn Net_EPC1G2_RStreadProtect(SOCKET hSocket, BYTE *AccessPassword)

Function: to release read protection for appointed tag, tag can read EPC content after success. Note: only one tag in the filed.and only works for UCODE G2XM tag of Philips.

Input parameter

hSacnner/hSocket: handle in communication port of reader

AccessPassword: tag access password

RS485Address:

RS485Add Reader RS485 network building address, RS485Address=0 expresses no network building.

Return:If rebound value of function is OK, it means release successfully or it is failure reason.

4.2.6 Read write ISO18000-6Dfunction

4.2.6.1 Identify tag ID number

apiReturn ISO18000-6D_ReadLabelID(HANDLE hScanner, BYTE *IDBuffer, int *nCounter, int Address)

apiReturn Net_EM4442_ReadLabelID(SOCKET hSocket, BYTE *IDBuffer, int *nCounter);

Function: identify all ID numbers of electronic tag under the range of antenna radiation

Input parameter

hSacnner/hSocket: handle in communication port of reader

nMax: Reserve the parameter

RS485Address:

RS485Add Reader RS485 network building address, RS485Address=0 expresses no network building .

Output parameter

nCounterreturn to ID number of tag that read in actual

IDBuffer: to reserve ID numbers of read tag

Return: If rebound value of function is OK, it means identified successfully, return ID number of tag or it is failure reason.

ASKA
ul. Wędkarska 2A/B1, 04-869 Warszawa
tel. 22 4985908/9, fax 22 6177020
e-mail: ask@aska.com.pl www.kodykreskowe.com