射频卡读写模块YW-205
使用手册
(version 1.5)

网址：	http://www.youwokeji.com.cn
EMAIL：	Zhou21cn@126.com
电话：	010-59395668
手机：	013671114914
联系人：	周先生

1. YW205

§ 1.1 概述

射频卡读写模块 YW205 非常适合为设备连接的读写器，YW205 使用 PHILIPS 生产的射频基站的非接触 IC 卡读写器，可以读写符合 ISO14443 TypeA 标准的非接触 IC 卡，ISO14443B 标准的非接触 IC 卡，以及符合 ISO15693 协议的智能卡。

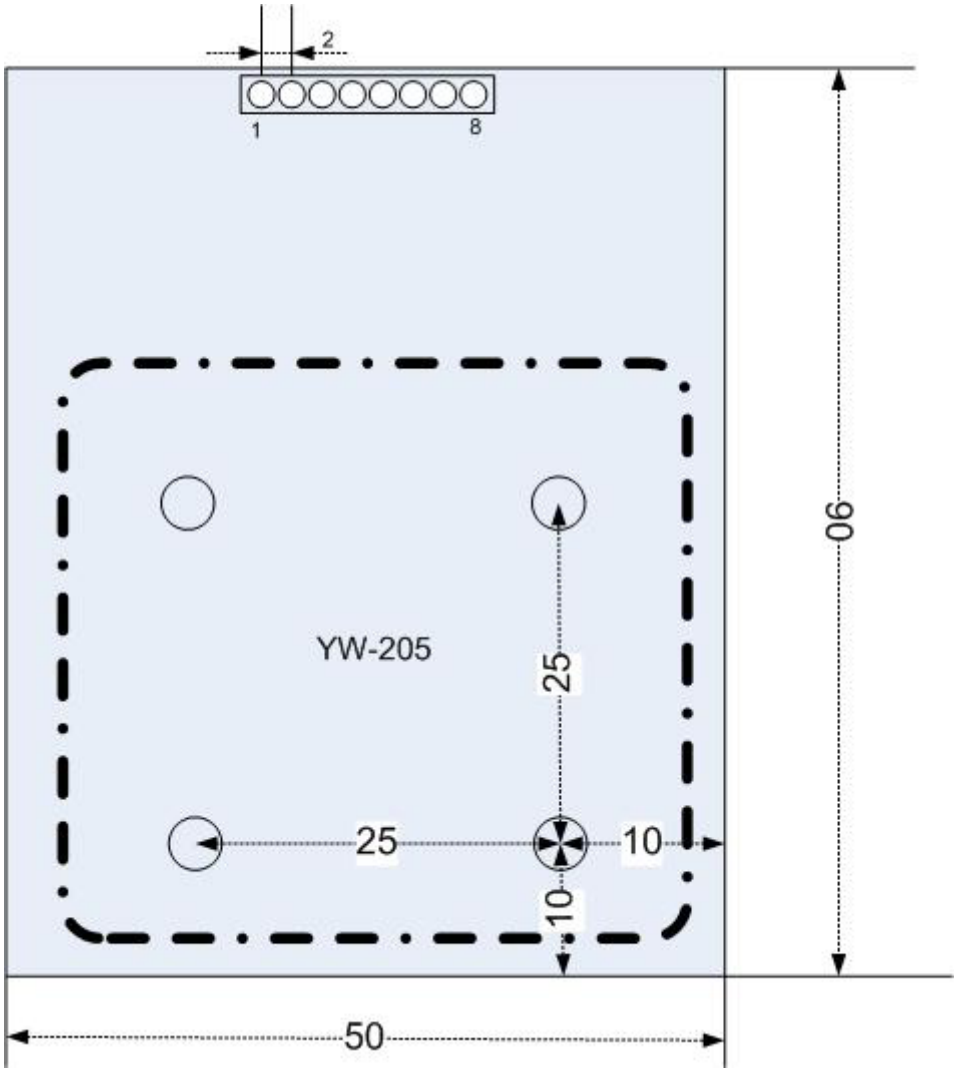
§ 1.2 特点

- 串口波特率：4800 ～ 115200 BPS
- 电源： DC5V
- 最大功耗： 1.5W
- 环境温度： -30℃ ～ +70℃
- 相对湿度： 35% ～ 95%
- 接口： RS232（或 TTL）
- 可扩展内存，功能可订制。

§ 1.2 YW205 相关型号

型号	读卡协议	供电电压	支持接口	支持的射频卡	本页
YW-205-A	ISO14443-A	+5V	UART(232 可选)	▪ Mifare S50,S70	
YW-205-AB	ISO14443-A ISO14443-B	+5V	UART(232 可选)	▪ Mifare S50,S70 ▪ ISO14443-4 TypeB /A CPU 卡	
YW-205-I	ISO15693	+5V	UART(232 可选)	▪ ICode、TI-Tag 等标签	
YW-205-ABI	ISO14443-A ISO14443-B ISO15693	+5V	UART(232 可选)	▪ Mifare S50,S70 ▪ ISO14443-4 TypeB /A CPU 卡 ▪ TI-Tag ,ICode2 等标签	√

§ 1.3 尺寸



单位：mm

§ 1.4 接口

PIN	Signal	线颜色
1	VCC	DC5V
2	GND	地 GND
3	RXD	串口接受
4	TXD	串口发送
5	IO1	扩展未用
6	IO2	扩展未用
7	IO3	扩展未用
8	IO4	扩展未用

2. 数据通讯协议

§ 2.1 Uart 协议

● 电脑传送格式：

HEAD				DATA	CHECK	END
SOH	LEN	ID	COMMAND			
0x02	2Bytes	2Bytes	1Byte	n Btytes	XOR	0x03
起始码	包长度	设备 ID	功能码	资料	校验	结束码

☞帧格式：1 位起始位＋8 位数据位＋1 位停止位(异步半双工)

☞通讯速率：默认19200

☞起始码(SOH)：帧起始标志码(0x02)

☞长度字节(LEN)：从长度字节本身到校验字节，高字节在前

☞设备标志符(ID)：0~65535，高字节在前，0为广播地址

☞功能码(COMMAND)：看通讯指令表

☞资料(DATA)：通讯过程中，命令的内容

☞校验(CHECK)：从长度字节到数据最后一字节异或值(XOR)

☞注意：从长度字节到包尾之间数据，若出现 0x02,0x03,0x10,在之前补 0x10，且长度字节不包括填补值 0x10，计算校验时候插补的 0x10 不用计算。

● Reader 回应格式：

HEAD				DATA		CHECK	END
SOH	LEN	ID	COMMAND	STATUS	DATA		
0x02	2Bytes	2Bytes	1Byte	1Byte	n Btytes	XOR	0x03
起始码	包长度	设备 ID	功能码	状态码	资料	校验	结束码

☞帧格式：1 位起始位＋8 位数据位＋1 位停止位(异步半双工)

☞通讯速率：默认115200

☞起始码(SOH)：帧起始标志码(0x02)

☞长度字节(LEN)：从长度字节本身到校验字节，高字节在前

☞设备标志符(ID)：0~65535，高字节在前

☞功能码(COMMAND)：看通讯指令表

☞状态码(STATUS)：执行结果(见状态列表)

成功：0x00
错误：0x01~0xFF

☞资料(DATA)：通讯过程中，命令的内容

☞校验(CHECK)：从长度字节到数据最后一字节异或值(XOR)

☞注意：从长度字节到包尾之间数据，若出现 0x02,0x03,0x10,在之前补 0x10，且长度字节不包括填补值 0x10，计算校验时候插补的 0x10 不用计算。

3．基本通讯指令表

序号	命令名称	命令方向	长度字	命令字	说明
系统级命令					
	设置读卡器波特率	发送	0x0007	0x01	波特率(1Byte)->0x00->4800bps 0x01->9600bps 0x02->14400bps 0x03->19200bps(开机默认) 0x04->28800bps 0x05->38400bps 0x06->57600bps 0x07->115200bps
		正确返回	0x0007	0x01	0x00
		错误返回	0x0007	0x01	错误信息(1Byte)
	设置设备标识	发送	0x0008	0x02	设备标识(2Bytes)
		正确返回	0x0007	0x02	0x00
		错误返回	0x0007	0x02	错误信息(1Byte)
	查询设备标识	发送	0x0006	0x03	
		正确返回	0x0009	0x03	0x00 设备标识(2Bytes)
		错误返回	0x0007	0x03	错误信息(1Byte)
	查询硬件版本号	发送	0x0006	0x04	
		正确返回	0x0009	0x04	0x00 硬件版本号(2Bytes)
		错误	0x0007	0x04	0xFF

		返回				
	查询读写器产品序列号	发送	0x0006	0x05		
		正确返回	0x000F	0x05	0x00	产品序列号(8Bytes)
		错误返回	0x0007	0x05	错误信息(1Byte)	
	设置蜂鸣器	发送	0x0009	0x06	响时间(1Byte)+不响时间(1Byte)+循环次数(1Byte)　时间单位 100ms	
		正确返回	0x0007	0x06	0x00	
		错误返回	0x0007	0x06	错误信息(1Byte)	
	设置 LED	发送	0x000A	0x07	LED 序号+亮时间+不亮时间+循环次数 LED 序号： 0x01->黄灯 0x02->绿灯 时间单位 100ms	
		正确返回	0x0007	0x07	0x00	
		错误返回	0x0007	0x07	错误信息(1Byte)	
SAM 卡命令						
	设置 SAM 卡波特率	发送	0x0007	0x10	波特率设置(1Byte) 0x00->9600bps(低速sam卡) 0x01->38400bps(高速sam卡)	
		正确返回	0x0007	0x10	0x00	
		错误返回	0x0007	0x10	错误信息(1Byte)	
	复位 SAM	发送	0x0006	0x11		
		正确返回	0x0007+n	0x11	0x00	复位信息(nBytes)
		错误返回	0x0007	0x11	错误信息(1Byte)	
	向 SAM 发送 COS	发送	0x0006+n	0x12	COS 指令(nBytes)	
		返回	0x0007+n	0x12	0x00	返回信息(nBytes)
ISO14443 卡片级命令						
	设置读写工作模式	发送	0x0007	0x20	工作模式： ‘A’-type A ‘B’-typeB ‘I’ –ISO15693	

		正确返回	0x0007	0x20	0x00	
		错误返回	0x0007	0x20	错误信息(1Byte)	
	设置天线状态	发送	0x07	0x21	天线工作模式： 0x00－>关闭 0x01－>打开	
		正确返回	0x0007	0x21	0x00	
		错误返回	0x0007	0x21	错误信息(1Byte)	
	寻 TypeA 卡片	发送	0x0006	0x25	寻卡模式(1Byte) 寻卡模式： 0x52->寻天线内所有的卡 0x26->寻天线内未休眠的卡	
		正确返回	0x0007	0x25	0x00	卡片类型(2Bytes)＋卡片容量(1Bytes)+卡片序列号(4bytes、7Bytes 或 10bytes)
		错误返回	0x0007	0x25	错误信息(1Byte)	
	使激活的卡进入 Halt 状态	发送	0x0006	0x26		
		正确返回	0x0007	0x26	0x00	
		错误返回	0x0007	0x26	错误信息(1Byte)	
	验证 mifareone 密钥	发送	0x000E	0x28	授权模式(1Byte)+块号(1Byte)+密钥(6Bytes) 授权模式 1Byte: 0x60->A 密钥 0x61->B 密钥	
		正确返回	0x0007	0x28	0x00	
		错误返回	0x0007	0x28	错误信息(1Byte)	
	读取 Mifareone 块数据	发送	0x0007	0x29	块号(1Byte)	
		正确返回	0x0007	0x29	0x00	块数据(16Bytes)
		错误返回	0x0007	0x29	错误信息(1Byte)	
	写入 Mifareone 块数据	发送	0x0017	0x2A	块号(1Byte)+块数据(16Bytes)	
		正确返回	0x0007	0x2A	0x00	
		错误	0x0007	0x2A	错误信息(1Byte)	

		返回				
	初始化钱包	发送	0x000B	0x2B	块号(1Byte)＋钱包值(4Bytes)	
		正确返回	0x0007	0x2B	0x00	
		错误返回	0x0007	0x2B	错误信息(1Byte)	
	读取钱包	发送	0x0007	0x2C	块号(1Byte)	
		正确返回	0x000B	0x2C	0x00	钱包值(4Bytes)
		错误返回	0x0007	0x2C	错误信息(1Byte)	
	扣款	发送	0x000B	0x2D	块号(1Byte)+扣款值(4Bytes)	
		正确返回	0x0007	0x2D	0x00	
		错误返回	0x0007	0x2D	错误信息(1Byte)	
	充值	发送	0x000B	0x2E	块号(1Byte)+充值(4Bytes)	
		正确返回	0x0007	0x2E	0x00	
		错误返回	0x0007	0x2E	错误信息(1Byte)	
	Restore	发送	0x0007	0x2F	块号(1Byte)	
		正确返回	0x0007	0x2F	0x00	
		错误返回	0x0007	0x2F	错误信息(1Byte)	
	Transfer	发送	0x0007	0x30	块号(1Byte)	
		正确返回	0x0007	0x30	0x00	
		错误返回	0x0007	0x30	错误信息(1Byte)	
	Ultralight 卡读	发送	0x0007	0x31	块号(1Byte)	
		正确返回	0x0007	0x31	0x00	卡片数据(4Bytes)
		错误返回	0x0007	0x31	错误信息(1Byte)	
	Ultralight 卡写	发送	0x0007	0x32	块号(1Byte)+数据(4Bytes)	
		正确返回	0x0007	0x32	0x00	
		错误返回	0x0007	0x32	错误信息(1Byte)	
	寻感应区内	发送	0x0007	0x33	寻卡模式	

	符合 ISO14443 TYPE_A 标准的 CPU 卡并复位				0x26->未进入休眠的卡 0x52->所有卡	
		正确返回	0x0007＋n	0x33	0x00	复位信息内容(n)
		错误返回	0x0007	0x33	错误信息(1Byte)	
	发送符合 ISO14443 TypeA/TypeB CPU 卡发送 COS 命令	发送	0x0007	0x34	Cos命令	
		正确返回	0x0007＋n	0x34	0x00	返回数据(n)(包含 SW1、SW2)
		错误返回	0x0007	0x34	错误信息(1Byte)	
	寻感应区内 Desfire 卡并复位	发送	0x0007	0x35	寻卡模式 0x26->未进入休眠的卡 0x52->所有卡	
		正确返回	0x0007＋n	0x35	0x00	复位信息内容(n)
		错误返回	0x0007	0x35	错误信息(1Byte)	
	Desfire 发送 COS 命令	发送	0x0007	0x36	Cos命令	
		正确返回	0x0007＋n	0x36	0x00	
		错误返回	0x0007	0x36	错误信息(1Byte)	
	寻感应区内符合 ISO14443 TYPE_B 标准的 CPU 卡并复位	发送	0x0008	0x40	型号(1Byte) 1：WUPB (寻所有卡)；0 寻未休眠的卡	
		正确返回	0x0007＋n	0x40	0x00	
		错误返回	0x0007	0x40	错误信息(1Byte)	
	ISO14443-4 TypeB 卡休眠	发送	0x000A	0x41	PUPI(4 Bytes)	
		正确返回	0x0007	0x41	0x00	
		错误返回	0x0007	0x41	错误信息(1Byte)	
ISO15693 卡片						
28	Inventory 指令	发送	0x0006	0x50		
		正确返回	0x0010	0x50	0x00	DSFID(1Byte)+UID(8Byte)
		错误返回	0x0007	0x50	错误信息(1Byte)	
29	Stay quiet	发送	0x000E	0x51	UID(8Bytes)	

	指令	正确返回	0x0007	0x51	0x00
		错误返回	0x0007	0x51	错误信息(1Byte)
30	Select 指令	发送	0x000E	0x52	UID(8Bytes)
		正确返回	0x0007	0x52	0x00
		错误返回	0x0007	0x52	错误信息(1Byte)
31	Reset To Ready 指令	发送	0x000F	0x53	Mode(1Byte)+UID(8Bytes) 注： 1. Mode 操作模式 Bit0：Select_Flag 0－该操作对给定 UID 卡操作，那么 BIT1= 1，后面 8 个 UID 是实际要操作卡片的 UID 1－该操作对选定的卡操作，那么 BIT1 = 0，后面 8 个 UID 可以为任意值，在执行该指令前，必须执行 Select 指令。 Bit1：Address_flag 0－非寻址请求，后面所带 UID 无效 1－寻址请求，后面所带 UID 为当前操作卡片 UID 2. 对于选定卡操作，那么后面 UID 可以为任意值，对于非选定卡操作，则后面 UID 为实际操作卡 UID
		正确返回	0x0007	0x53	0x00
		错误返回	0x0007	0x53	错误信息(1Byte)
32	Read Block 指令	发送	0x0011	0x54	Mode(1Byte)+UID(8Bytes)+起始块号(1Byte)+块数(1Byte) 注： 1. Mode 及 UID 设置见上面说明
		正确返回	0x0007 +n	0x54	0x00 | BlockData(nBytes)
		错误返回	0x0007	0x54	错误信息(1Byte)
33	Write Block 指令	发送	0x0014	0x55	Mode(1Byte)+UID(8Bytes)+块号(1Byte)+块数据(4Byte) 注： 1. Mode 及 UID 设置见上面说明
		正确返回	0x0007	0x55	0x00

		错误返回	0x0007	0x55	错误信息(1Byte)
34	Lock Block 指令	发送	0x0010	0x56	Mode(1Byte)+UID(8Bytes)+块号(1Byte) 注： 1．Mode 及 UID 设置见上面说明
		正确返回	0x0007	0x56	0x00
		错误返回	0x0007	0x56	错误信息(1Byte)
35	Write AFI 指令	发送	0x0010	0x57	Mode(1Byte)+UID(8Bytes)+AFI(1Byte) 注： 1．Mode 及 UID 设置见上面说明
		正确返回	0x0007	0x57	0x00
		错误返回	0x0007	0x57	错误信息(1Byte)
36	Lock AFI 指令	发送	0x000F	0x58	Mode(1Byte)+UID(8Bytes) 注： 1．Mode 及 UID 设置见上面说明
		正确返回	0x0007	0x58	0x00
		错误返回	0x0007	0x58	错误信息(1Byte)
37	Write DSFID 指令	发送	0x0010	0x59	Mode(1Byte)+UID(8Bytes) +DSFID(1Byte) 注： 1．Mode 及 UID 设置见上面说明
		正确返回	0x0007	0x59	0x00
		错误返回	0x0007	0x59	错误信息(1Byte)
38	Lock DSFID 指令	发送	0x000F	0x5A	Mode(1Byte)+UID(8Bytes) 注： 1．Mode 及 UID 设置见上面说明
		正确返回	0x0007	0x5A	0x00
		错误返回	0x0007	0x5A	错误信息(1Byte)
39	Get System Information 指令	发送	0x000F	0x5B	Mode(1Byte)+UID(8Bytes) 注： 1．Mode 及 UID 设置见上面说明
		正确返回	0x0007 +n	0x5B	0x00 | SystemInfo(nBytes)
		错误返回	0x0007	0x5B	错误信息(1Byte)

40	Get Multiple Block Security status 指令	发送	0x0011	0x5C	Mode(1Byte)+UID(8Bytes)+起始块号(1Byte)+块数(1Byte) 注： 1．Mode 及 UID 设置见上面说明	
		正确返回	0x0007+n	0x5C	0x00	Security status(nBytes)
		错误返回	0x0007	0x5C	错误信息(1Byte)	

5．通讯例子(红色部分不能算入校验，斜体加粗是应答数据)

1、reader Baudrate

Send: 02 00 07 00 00 01 07 01 03

Ack: 02 00 07 00 01 01 ***00*** 07 03

2、设定设备标识

Send: 02 00 08 00 00 10 02 00 04 0E 03

Ack : 02 00 07 00 04 10 02 ***00*** 01 03

3、查询设备标识符

Send: 02 00 06 00 00 10 03 05 03

Ack : 02 00 09 00 04 10 03 ***00 00 04*** 0A 03

4、查询硬件版本号

Send: 02 00 06 00 00 04 10 02 03

Ack : 02 00 09 00 04 04 ***00 01 00*** 08 03

5、查询产品序列号

Send: 02 00 06 00 00 05 10 03 03

Ack : 02 00 0F 00 04 05 ***00 59 57 2D 36 30 31 52 41*** 09 03

6、设置蜂鸣器

Send: 02 00 09 00 00 06 05 05 04 0B 03

Ack : 02 00 07 00 04 06 ***00*** 05 03

7、设置读卡类型

Send：02 00 07 00 00 20 41 66 03（’A’）

ACK： 02 00 07 00 01 20 ***00*** 26 03

8、Turn on antenna(打开天线)

Send：02 00 07 00 00 21 01 27 03 (Turn on)

ACK： 02 00 07 00 01 21 ***00*** 27 03

Send：02 00 07 00 00 21 00 26 03 (Turn off)

ACK： 02 00 07 00 01 21 ***00*** 27 03

9、Request(寻卡，返回卡类型)

Send：02 00 07 00 00 25 52 70 03

ACK：02 00 11 00 01 25 ***00 44 00 00 04 32 A3 E1 BF 10 02 80*** 38 03(Ultralight)

ACK：02 00 0E 00 01 25 ***00 04 00 08 F7 AE A1 67*** B9 03(Mifare)

Card type(卡类型):

0x44、0x00 = ultra_light

0x04、0x00 = Mifare_One(S50)

0x02、0x00 = Mifare_One(S70

0x44、0x03 = Mifare_DESFire

0x08、0x00 = Mifare_Pro

0x04、0x03 = Mifare_ProX

10、Halt Card

Send：02 00 06 00 00 26 20 03

ACK : 02 00 07 00 01 26 ***00*** 20 03

11、Authentication Card (KEY:A， Block : 1， KEY: FF FF FF FF FF FF)

Send：02 00 0E 00 00 28 60 01 FF FF FF FF FF FF 47 03

ACK : 02 00 07 00 01 28 ***00*** 2E 03

12、Read a Block (Block: 1)

Send：02 00 07 00 00 29 01 2F 03

ACK : 02 00 17 00 01 29 00 ***00 00 00 79 FF FF FF 86 00 00 00 79 01 FE 01 FE*** 46 03

13、Write a block (Block: 1, DATA: 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11)

Send：02 00 17 00 00 2A 01 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 3C 03

ACK : 02 00 07 00 01 2A ***00*** 2C 03

14、Initial purse(Block: 1 money:1(00 00 00 01))

Send：02 00 0B 00 00 2B 01 00 00 00 01 20 03

ACK : 02 00 07 00 01 2B ***00*** 2D 03

15、Read purse(Block：1)

Send：02 00 07 00 00 2C 01 2A 03

ACK : 02 00 0B 00 01 2C ***00 00 00 00 01*** 27 03

16、Decrement(Block：1， money：1(00 00 00 01))

Send：02 00 0B 00 00 2D 01 00 00 00 01 26 03

ACK : 02 00 07 00 01 2D ***00*** 2B 03

17、Increment(Block：1， money：0x02(00 00 00 02))

Send：02 00 0B 00 00 2E 01 00 00 00 10 02 26 03

ACK : 02 00 07 00 01 2E ***00*** 28 03

18、Restore(Block：1)

Send：02 00 07 00 00 2F 01 29 03

ACK ：02 00 07 00 01 2F *00* 29 03

19、Transfer(Block：1)

Send：02 00 07 00 00 30 01 36 03

ACK ：02 00 07 00 01 30 *00* 36 03

6．通讯例子代码

```
/*****************************************************************************
*Function：UART Receive (PC receive)
*****************************************************************************/
unsigned char xdata g_cReceiveBuffer[240] ;              //Receive buffer
unsigned int g_iReceiveCounter;                          //Length of receiving data
unsigned char g_bReceived0x10Flag;                       //Is receiving 0x10?
unsigned char g_bPackageStarted;                         //Package start
unsigned char g_bReceiveOK;                              //Package has been received
void seri_int (void) interrupt 4 using 1
{
static unsigned char data i;
unsigned int j;
unsigned char cCheckSum;

if(RI)
{
      i   = SBUF;
      RI  = 0;

      if(!g_bReceiveOK)        //Receive datas
      {
            if((0x02 == i) && (0 == g_bReceived0x10Flag))
            {
                  g_bPackageStarted = 1;
                  g_iReceiveCounter = 0;
                  cCheckSum = 0;
            }
            else if((0x03 == i) && (0 == g_bReceived0x10Flag) && (g_bPackageStarted))          //package end
            {
                  g_bPackageStarted = 0;
```

```
                if      (((g_cReceiveBuffer[2]      ==      0)      &&      (g_cReceiveBuffer[3]      ==      0))      ||
((g_cReceiveBuffer[2]==g_cICDEVH)&&(g_cReceiveBuffer[3]==g_cICDEVL)))
                {
                      j  = (unsigned int)(g_cReceiveBuffer[0] <<8) + g_cReceiveBuffer[1];
                      if ((g_iReceiveCounter > 5)&&(j == g_iReceiveCounter))
                      {
                            cCheckSum = 0;
                            for(j=0; j<g_iReceiveCounter; j++)
                            {
                                  cCheckSum ^= g_cReceiveBuffer[j];
                            }
                            if(cCheckSum == 0)
                            {
                                  g_cCommand    = g_cReceiveBuffer[4];
                                  g_bReceiveOK = 1;
                                  ES                  = 0;
                            }
                            else
                            {
                                  g_bPackageStarted = 0;
                                  g_iReceiveCounter = 0;
                            }
                      }
                }
          }
          else if((0x10 == i) && (0 == g_bReceived0x10Flag))
          {
                g_bReceived0x10Flag = 1;
          }
          else if(g_bPackageStarted)
          {
                g_bReceived0x10Flag = 0;

                if(g_iReceiveCounter < sizeof(g_cReceiveBuffer))
                {
                      g_cReceiveBuffer[g_iReceiveCounter++] = i;
                }
                else
                {
                      g_bPackageStarted = 0;
                      g_iReceiveCounter = 0;
                }
          }
          else
```

```
            {
                  g_bReceived0x10Flag = 0;
            }
      }
}
if (TI)
{
      TI = 0;
}
}

/*****************************************************************************
*Function：Send Datas(PC-→Reader)
*****************************************************************************/
void SendDataToPC(unsigned char *cpOutBuffer, unsigned int iLen, unsigned char cCommand)
{
      unsigned int i;
      unsigned int iSendLen;
      unsigned char cHeader[5];
      unsigned char cCheckSum = 0;

      iSendLen = iAnswerNum;

      cHeader[0] = (unsigned char)((iSendLen+6)>>8);
      cHeader[1] = (unsigned char)(iSendLen+6);
      cHeader[2] = g_cICDEVH;                         //Device ID MSB
      cHeader[3] = g_cICDEVL;                         //Device ID LSB  (0X00, 0X00 BroadCast ID)
      cHeader[4] = cCommand                           //Command

      //Caculate checksum
      for(i=0; i<5; i++)
      {
            cCheckSum ^= cHeader[i];
      }

      for(i=0; i<iSendLen; i++)
      {
            cCheckSum ^= cpOutBuffer[i];
      }

      //send header
      SendByte(0x02);

      //send data
```

```
	for(i=0; i<5; i++)
	{
		if((0x02 == cHeader[i]) || (0x03 == cHeader[i]) || (0x10 == cHeader[i]))
		{
			SendByte(0x10);
		}
		SendByte(cHeader[i]);
	}

	for(i=0; i<iSendLen; i++)
	{
		if((0x02 == cpOutBuffer[i]) || (0x03 == cpOutBuffer[i]) || (0x10 == cpOutBuffer[i]))
		{
			SendByte(0x10);
		}
		SendByte(cpOutBuffer[i]);
	}

	//send checksum
	if((0x02 == cCheckSum) || (0x03 == cCheckSum) || (0x10 == cCheckSum))
	{
		SendByte(0x10);
	}
	SendByte(cCheckSum);

	//send stx
	SendByte(0x03);
}
```

订购方式

可以通过我们的网站或电话订购。或者联系当地的经销商。

北京友我科技有限公司

网站：http://www.youwokeji.com.cn

电话：010-59395668

24小时手机：13671114914

Email：Zhou21cn@126.com