

北京友我科技有限公司



友我科技
YOWO TECHNOLOGY

射频卡读写模块 YW-203

使用手册

Version 1.1

● 简体中文版

繁體中文版

English Edition

网址: <http://www.youwokeji.com.cn>

EMAIL: coodor@126.com

电话: 010-59395668

手机: 013671114914

联系人: 周先生

目录

1 概述	5
2 YW-203 功能特点	6
3 硬件描述	7
4 数据通信协议	8
5 基本通讯指令表	12
6 命令测试实例	18
7 典型电路	19
8 关于天线	20
9 订购方式	20

版本信息

版本	描述	日期
V1.0	初始版本	2008/05/28

1 概述

§ 1.1 YW200 系列读写模块简介

YW-200 系列射频卡读写模块采用 PHILIPS 公司的非接触技术(13.56M)设计的微型嵌入式非接触式 IC 卡读写模块。 **YW-200** 系列模块内嵌 MF RC400/500/530/531/632 射频基站, 用户不必关心射频基站复杂的控制方法, 只需要简单地通过选定的 UART 或 IIC 接口发送命令就可以对卡片进行所有的操作。**YW-200** 系列模块支持 ISO14443-A Mifare One(S50/S70), UltraLight、MifarePro、ISO14443-4 TypeB、SR176、ISO15693、ICODE SL2 及其兼容卡片。同时针对于 Mifare one 卡片, 模块增加了自动寻卡功能。

该系列模块广泛应用于非接触智能水、电、气三表、交通卡通读写器, 桌面发卡器, 门禁考勤读写卡器, 汽车电子感应锁配套, 办公/商场/洗浴中心储物箱的安全控制, 各种防伪系统及生产过程控制, 数据采集等。用户可以在最短的时间内开发出符合自己需求的稳定的射频卡系统。各种型号支持的卡片请参见 1.2。

§ 1.2 YW200 系列读写模块型号

13.56M 射频卡读写模块 YW-200 系列有以下型号, 请选择合适的射频卡模块。

型号	读卡协议	供电电压	支持接口	支持的射频卡	本页
YW-201	ISO14443-A	+5V	IIC,UART(TTL)	▪ Mifare S50,S70	
YW-202	ISO14443-A ISO14443-B	+5V	IIC,UART(TTL)	▪ Mifare S50,S70 ▪ ISO14443-4 TypeB /A CPU 卡	
YW-203	ISO15693	+5V	IIC,UART(TTL)	▪ ICode、TI-Tag 等标签	√
YW-204	ISO14443-A ISO14443-B ISO15693	+5V	IIC,UART(TTL)	▪ Mifare S50,S70 ▪ ISO14443-4 TypeB /A CPU 卡 ▪ TI-Tag ,ICode2 等标签	

2 YW-203 功能特点

- ☞ 简单的命令集可完成对卡片的全部操作。
- ☞ 支持 ISO15693 协议卡片，包括 ICode、Ti-Tag 等卡片。
- ☞ 通信协议简单可靠，通信协议：
 1. UART: 适用于 PC 机或 8 位 UART 的单片机，波特率 19200Bps。
 2. IIC: 适用于单片机，最大通讯速率 400K。
- ☞ 采用通用的 DIP32 封装，轻松嵌入到各类系统中。
- ☞ +5V 供电。
- ☞ 读卡距离 5~10cm。(与天线和环境有关)
- ☞ 模块内含有电压保护稳定电路，可以应用于电压不太稳定的场合。
- ☞ 极高的性价比。而且提供详细资料让您的开发省时省力省心。

3 硬件描述

§ 3.1 管脚配置

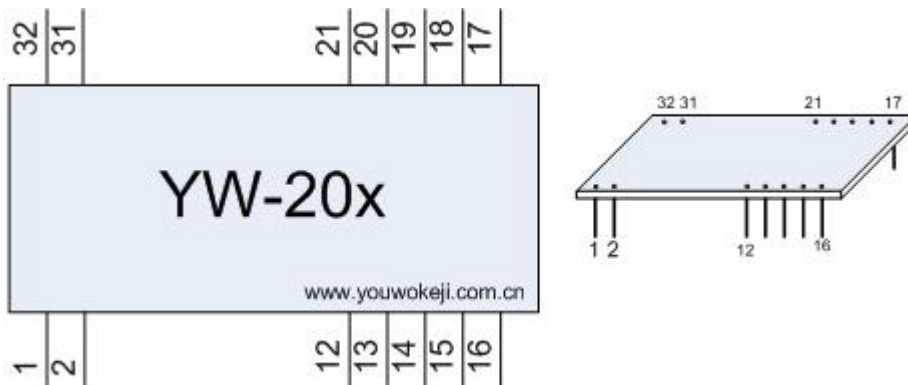


图 1 YW-203 引脚图

管脚定义如【表格 1】所示：

表格一：

管脚	符号	描述
1	RX	天线接收脚
2	TGND	天线地脚
12	NC	未用
13	NC	未用
14	NC	未用
15	TXD/SDA	UART 发送脚 或 IIC SDA 脚
16	RXD/SCL	UART 接收脚 或 IIC SCL 脚
17	VCC	+5V 电源
18	GND	电源地
19	PORTSEL	通讯接口选择(0: IIC, 1: UART)
20	NC	未用
21	NC	未用
31	TX2	天线输出
32	TX1	天线输出

注：该封装尺寸遵循 DIP32 封装大小

§ 3.2 封装尺寸

YW-203 射频卡读写模块采用 DIP32 封装，使用简单可靠。尺寸如下图所示：

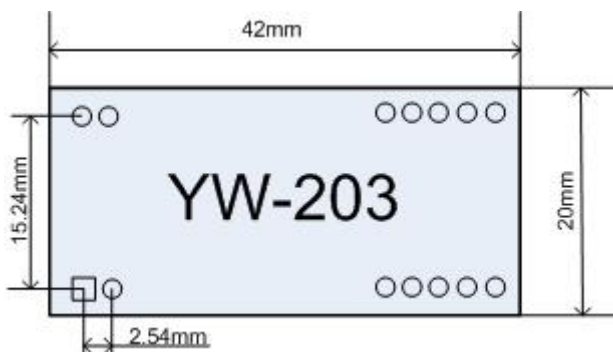


图 2 YW-203 封装图

4 数据通信协议

§ 4.1 Uart 协议

● 控制器传送格式:

SOH	LEN	COMMAND	DATA	CHECK	END
0x02	1 Byte	1 Byte	n Bytes	1 Byte	0x03
起始码	包长度	功能码	资料	XOR 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)

☞ 包长度 (LEN): 从长度字节本身到校验字节, 高字节在前

☞ 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。

● Module 应答格式：

SOH	LEN	COMMAND	DATA		CHECK	END
			STATUS	DATA		
0x02	1 Byte	1 Byte	1Byte	n Bytes	1 Byte	0x03
起始码	包长度	功能码	状态码	资料	XOR 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)

☞ 包长度(LEN)：从长度字节本身到校验字节，高字节在前

☞ 功能码(COMMAND)：看通讯指令表

☞ 功能码(COMMAND)：看通讯指令表

☞ 状态码(STATUS)：执行结果

 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。

发送实例：

要发送 命令号：0x10，数据内容：0x00 的数据包。

起始码：0x02。

包长度：0x04 (=长度 1 字节+命令 1 字节+内容 1 字节+校验 1 字节)

功能码：0x10，0x10 (实际命令号 0x10，约定在 0x02，0x03，0x10 之前加 0x10)

资料： 0x00

校验字节：0x14 (=0x04^0x10^0x00)

结束码：0x03

所以实际发送的数据包: 0x02,0x04,0x10,0x10,0x00,0x14,0x03

应答实例:

接收到以下数据包

02 08 10 10 00 4D 56 A2 57 F6 03,

对其解析如下:

第一步: 去掉所有额外增加的 0x10, 得到 02 08 10 00 4D 56 A2 57 F6 03

第二步: 校验 $08 \oplus 10 \oplus 00 \oplus 4D \oplus 56 \oplus A2 \oplus 57 = F6$, 校验正确。

第三步: 分解

起始码: 0x02

包长度: 0x08

功能码: 0x10

状态码: 0x00, 表示成功

资料: 4D 56 A2 57

校验字节: F6

结束码: 0x03

§ 4.2 IIC 协议

● 控制器传送格式:

ADDRESS	LEN	COMMAND	DATA	CHECK
写: 0xA0	1 Byte	1 Byte	n Bytes	1Byte
读: 0xA1				
模块地址 (W/R)	包长度	功能码	资料	校验字节

☞ 模块地址 (W/R):

模块地址为: 0xA0, 写bit0 为0, 则写指令为: $0xA0 + 0x00 = 0xA0$

模块地址为: 0xA0, 读bit0 为1, 则读指令为: $0xA0 + 0x01 = 0xA1$

☞ 包长度: 从长度字节本身到校验字节

☞ 功能码: 看通讯命令表

☞ 资料: 通讯过程中, 命令的内容

☞ 校验字节: 从长度字节到数据最后一字节异或值

● Module 应答格式:

LEN	COMMAND	DATA		CHECK
		STATUS	DATA	

1 Byte	1 Byte	1Byte	n Bytes	1Byte
包长度	功能码	状态码	资料	校验字节

☞ 包长度：从长度字节本身到校验字节

☞ 功能码：看通讯命令表

☞ 状态码 (STATUS)：执行结果

成功：0x00

错误：0x01~0xFF

☞ 资料：通讯过程中，命令的内容

☞ 校验字节：从长度字节到数据最后一字节异或值

§ 4.3 通讯端口切换

YW-203 读卡模块同时支持 UART 和 IIC 通讯接口。切换方式：

PORTSEL = 1 (High)，模块使用 UART 通讯 (波特率 19200)。

PORTSEL = 0 (Low)，模块使用 IIC 通讯 (最大通讯速率 400K)。

注意：切换要保持一定时间，建议 50ms。

5 基本通讯指令表

序号	功能码名称	方向	包长度 LEN	功能码 COMMAND	资料(资料注释) DATA	
模块级指令						
1	模块设置	发送	0x04	0x01	模块设置(1Byte) 0: 关闭天线 1: 打开天线	
		正确返回	0x04	0x01	0x00	
		错误返回	0x04	0x01	错误代码(1Byte)	
2	节电模式设置	发送	0x03	0x02		
		正确返回	0x04	0x02	0x00	
		错误返回	0x04	0x02	错误代码(1Byte)	
3	读 EEPROM	发送	0x06	0x03	地址(2B)+长度(1Bytes) 注: 1. 地址: 高地址在前 2. 长度: 最大为 16Bytes	
		正确返回	n+4	0x03	0x00	数据(nBytes)
		错误返回	0x04	0x03	错误代码(1Byte)	
4	写 EEPROM	发送	n+5	0x04	地址(2Bytes)+Data(nBytes) 注: 1. 地址: 高地址在前 2. Data: 最大为 16Bytes	
		正确返回	0x04	0x04	0x00	
		错误返回	0x04	0x04	错误代码(1Byte)	
5	设置读写工作模式	发送	0x04	0x05	工作模式(1Byte) 注: 'A'-type A(0x41) 'B'-typeB(0x42) '1' -ISO15693(0x31)(YW-203 只支持该模式)	

		正确返回	0x04	0x05	0x00
		错误返回	0x04	0x05	错误代码(1Byte)
ISO 15693 协议卡片 (TI-Tag)					
21	Inventory 指令	发送	0x03	0x50	
		正确返回	0x04	0x50	0x00 DSFID(1Byte)+UID(8Byte)
		错误返回	0x04	0x50	错误信息(1Byte)
22	Stay quiet 指令	发送	0x0B	0x51	UID(8Bytes)
		正确返回	0x04	0x51	0x00
		错误返回	0x04	0x51	错误信息(1Byte)
22	Select 指令	发送	0x0B	0x52	UID(8Bytes)
		正确返回	0x04	0x52	0x00
		错误返回	0x04	0x52	错误信息(1Byte)
23	Reset To Ready 指令	发送	0x0C	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 Bit2: 0 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	0x04	0x53	0x00
		错误返回	0x04	0x53	错误信息(1Byte)
24	Read Block	发送	0x0E	0x54	Mode(1Byte)+UID(8Bytes)+ 起始块号

	指令				(1Byte)+块数(1Byte) 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 Bit2: 0
		正确返回	0x04+n	0x54	0x00 BlockData(nBytes)
		错误返回	0x04	0x54	错误信息(1Byte)
25	Write Block 指令	发送	0x11	0x55	Mode(1Byte)+UID(8Bytes)+块号(1Byte)+块数据(4Byte) 注: 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 Bit2: 1
		正确返回	0x04	0x55	0x00
		错误返回	0x04	0x55	错误信息(1Byte)
26	Lock Block 指令	发送	0x0D	0x56	Mode(1Byte)+UID(8Bytes)+块号(1Byte) 注:

Mode 操作模式

					<p>Bit0: Select_Flag</p> <p>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</p> <p>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。</p> <p>Bit1: Address_flag</p> <p>0—非寻址请求, 后面所带 UID 无效</p> <p>1—寻址请求, 后面所带 UID 为当前操作卡片 UID</p> <p>Bit2: 1</p>
		正确返回	0x04	0x56	0x00
		错误返回	0x04	0x56	错误信息(1Byte)
27	Write AFI 指令	发送	0x0D	0x57	<p>Mode(1Byte)+UID(8Bytes)+AFI(1Byte)</p> <p>注:</p> <p>Mode 操作模式</p> <p>Bit0: Select_Flag</p> <p>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</p> <p>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。</p> <p>Bit1: Address_flag</p> <p>0—非寻址请求, 后面所带 UID 无效</p> <p>1—寻址请求, 后面所带 UID 为当前操作卡片 UID</p> <p>Bit2: 1</p>
		正确返回	0x04	0x57	0x00
		错误返回	0x04	0x57	错误信息(1Byte)
28	Lock AFI 指令	发送	0x0C	0x58	<p>Mode(1Byte)+UID(8Bytes)</p> <p>注:</p> <p>Mode 操作模式</p> <p>Bit0: Select_Flag</p>

0—该操作对给定 UID 卡操作, 那么 BIT1=

					<p>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。</p> <p>Bit1: Address_flag</p> <p>0—非寻址请求, 后面所带 UID 无效</p> <p>1—寻址请求, 后面所带 UID 为当前操作卡片 UID</p> <p>Bit2: 1</p>
		正确返回	0x04	0x58	0x00
		错误返回	0x04	0x58	错误信息(1Byte)
29	Write DSFID 指令	发送	0x0D	0x59	<p>Mode(1Byte)+UID(8Bytes) + DSFID(1Byte)</p> <p>注:</p> <p>Mode 操作模式</p> <p>Bit0: Select_Flag</p> <p>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</p> <p>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。</p> <p>Bit1: Address_flag</p> <p>0—非寻址请求, 后面所带 UID 无效</p> <p>1—寻址请求, 后面所带 UID 为当前操作卡片 UID</p> <p>Bit2: 1</p>
		正确返回	0x04	0x59	0x00
		错误返回	0x04	0x59	错误信息(1Byte)
30	Lock DSFID 指令	发送	0x0C	0x5A	<p>Mode(1Byte)+UID(8Bytes)</p> <p>注:</p> <p>Mode 操作模式</p> <p>Bit0: Select_Flag</p> <p>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</p>

					卡片 UID	
					Bit2: 1	
		正确返回	0x04	0x5A	0x00	
		错误返回	0x04	0x5A	错误信息(1Byte)	
31	Get System Information 指令	发送	0x0C	0x5B	Mode(1Byte)+UID(8Bytes) 注: 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 Bit2: 0	
		正确返回	0x04+n	0x5B	0x00	SystemInfo(nBytes)
		错误返回	0x04	0x5B	错误信息(1Byte)	
32	Get Multiple Block Security status 指令	发送	0x0E	0x5C	Mode(1Byte)+UID(8Bytes)+ 起始块号(1Byte)+块数(1Byte) 注: 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 Bit2: 0	
		正确返回	0x04+n	0x5C	0x00	Security status(nBytes)

6.7 Select 卡

Send: 02 0B 52 **CE BE 7F 30 00 01 04 E0** 83 03

Receive: 02 04 52 **00** 56 03

6.8 Read block

Send: 02 0E 54 00 04 FD 46 25 00 00 07 E0 00 01 26 03

Receive: 02 07 54 00 00 00 00 53 03

6.9 WriteBlock

Send: 02 11 55 06 04 FD 46 25 00 00 07 E0 00 00 00 00 00 3F 03

Receive: 02 04 55 00 51 03

7 典型电路

采用 YW-203 模块构建射频卡系统， 无须额外的元器件， 即可实现功能起强大， 性能稳定的射频卡系统。 【图 3】采用 YW-203 射频卡读写模块,配合 YW-300 天线模块即可实现性价比高， 性能稳定的射频卡系统。

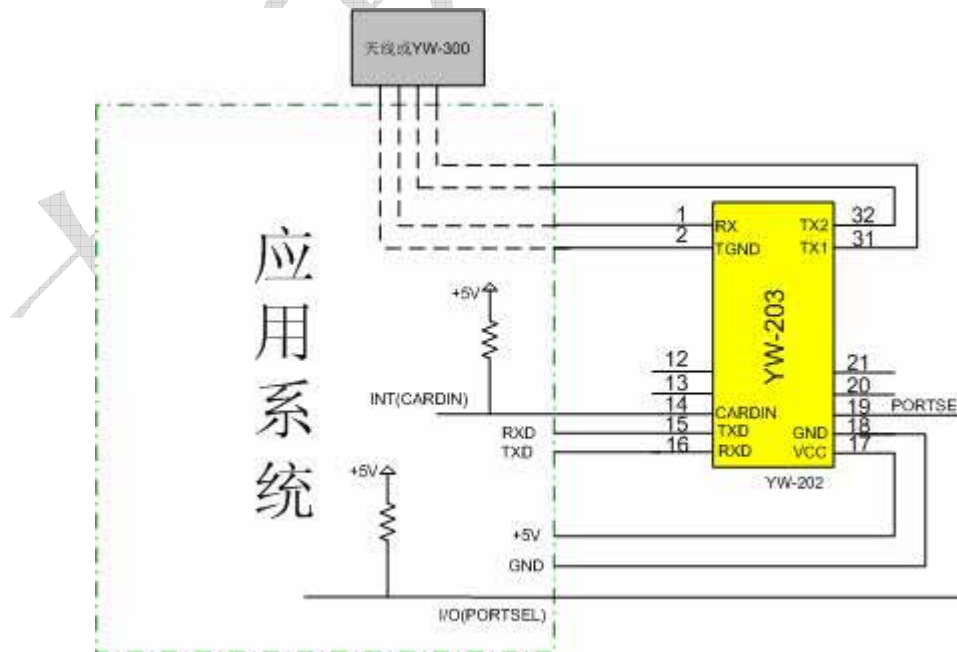


图 3 YW203 应用电路

8 关于天线

YW-203 射频卡读写模块可以选配 YW-300 天线模块，也可以单独制作天线。为了获得稳定的性能，建议采用 YW-300 天线模块。关于 YW-300 天线模块请登录网站 <http://www.youwokeji.com.cn> 查看。

9 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: coodor@126.com