

ZLG500B 读卡模块简介

§1 概述

1、照片



实际尺寸：41.5mm×25.3mm。

2、特征

- 四层电路板设计，双面表贴，EMC 性能优良
- 采用最新 PHILIPS 高集成 IS014443A 读卡芯片—MF RC500
- UART 串行接口，能外接 RS232 或 RS485 芯片
- 自动波特率探测
- 控制线输出口
- 蜂鸣器信号输出口，能用软件控制输出频率及持续时间
- 能读写 RC500 内 EEPROM
- 发光二极管指示模块当前状态
- 可提供 C51 函数库

§2 硬件描述

1、引脚描述

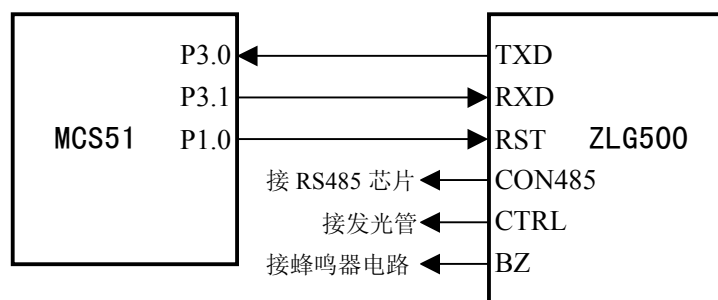
如照片所示，J1 为与天线的接口，J2 为与 MCU 的接口。如下二表所示。

管脚	符号	描述
J1-1	GND	地
J1-2	TX1	天线发送 1
J1-3	GND	地
J1-4	TX2	天线发送 2
J1-5	GND	地
J1-6	RX	天线接收

管脚	符号	类型	描述
J2-1	CTRL	输出	控制线输出，平时为高
J2-2	BZ	输出	蜂鸣器信号输出，平时为高，输出方波或低电平有效
J2-3	CON485	输出	RS485 控制，平时为低，TXD 发送时为高
J2-4	VCC	PWR	电源正端
J2-5	RST	复位	内部 MCU 复位端，高电平有效
J2-6	GND	PWR	电源负端
J2-7	RXD	输入	UART 接收端
J2-8	TXD	输出	UART 发送端

2、串行接口

ZLG500B 模块可方便地与 MCU 进行接口，如下图所示与 MCS51 单片机的典型接口。



也可以与 RS232 或 RS485 芯片直接连接，与外部控制器或 PC 接口。

ZLG500B 的 RST 有两种接法，一是接硬件复位电路，如阻容复位等，这样系统上电后必须要等待 ZLG500B 复位结束；二是接外部 MCU 的一个 I/O 口，由 MCU 控制复位。推荐使用第二种方法，这样在 ZLG500B 出现异常时可由 MCU 控制复位。

§ 3 ZLG500B 读卡模块 MCU 命令 C51 函数

命令		参数		补充
名称	数值	发送	接收	
Request	0x41	<i>_Mode</i>	<i>_TagType</i>	发出询问命令，检查在有效范围内是否有卡存在
Anticoll	0x42	<i>_Bcnt</i>	<i>_SNR</i>	开始防冲突操作，返回卡的序号
Anticoll2	0x71	<i>_Encoll, _Bcnt</i>	<i>_SNR</i>	可禁止或允许许多张卡进入
Select	0x43	<i>_SNR</i>	<i>_Size</i>	选择卡，返回卡的存贮容量
Authentication	0x44	<i>_Mode, _SecNr</i>	--	开始验证操作
Authentication2	0x72	<i>_Mode, _SecNr, _KeyNr</i>	--	可选择密匙区验证
Halt	0x45	--	--	将卡置于挂起模式
Read	0x46	<i>_Adr</i>	<i>_Data</i>	从卡中相应地址中读出一个 16 字节的块
Write	0x47	<i>_Adr, _Data</i>	--	向卡中相应地址写入一 16 字节的数据块
Increment	0x48	<i>_Adr, _Value</i>	--	增加访问单元块的字节数，并将结果保存在卡的内部寄存器
Decrement	0x49	<i>_Adr, _Value</i>	--	减少访问单元块的字节数，并将结果保存在卡的内部寄存器
Resore	0x4A	<i>_Adr</i>	--	将所访问单元块的字节数保存在卡的内部寄存器中
Transfer	0x4B	<i>_Adr</i>	--	将卡内部寄存器的内容转输到访问快的字节数
Value	0x70	<i>_Mode, _Adr, _Value, _Trans_Adr</i>	--	包含加、减、恢复函数，并带自动传送
LoadKey	0x4C	<i>_Mode, _SecNr, _Nkey</i>	--	改变存贮在 EEPROM 中的密钥
Reset	0x4E	<i>_Msec</i>	--	关闭天线输出数 ms，使卡复位
Get Info	0x4F	--	<i>_Info</i>	读取固件信息 RC500 序列号
Set Control Bit	0x50	--	--	将控制位置为高电平
Clr Control Bit	0x51	--	--	将控制位置为低电平
Config	0x52	--	--	复位且配置 RC500
Close	0x3F	--	--	关闭 RC500
Check Write	0x53	<i>_SNR, _Authmode, _Adr, _Data</i>	--	将所传送的数据和上一次所写的数据进行比较
Buzzer	0x60	<i>_Freguence, _Opentm, _Closetm, _Repcnt</i>	--	输出驱动蜂鸣器信号，能控制动作时间、间隙时间和重复次数
Read E2	0x61	<i>_Adr, _Length</i>	<i>_Data</i>	读 RC500 内 EEPROM 的内容
Write E2	0x62	<i>_Adr, _Length, _Data</i>	--	写数据到 RC500 内 EEPROM

3.8 应用程序举例

例子：读出RC500和MIFARE卡的序列号，然后选择一个卡，将该卡的20块初始化成值块，且备份到21块中，最后使该卡进入HALT状态，且驱动2KHz蜂鸣器响200毫秒。

将” ZLG500B.c ”、及主程序文件” main.c ” 放于同一项目中，然后在主程序中输入以下代码。

```
#define __SRC
#include "main.h"
#undef __SRC

#include "zlg500B.h"

sbit zlg500B_RST=P1^4;
uchar baud_num;
uchar card_snr[4];
uchar code Nkey_a[6] = {0xA0, 0xA1, 0xA2, 0xA3, 0xA4, 0xA5};
uchar code Nkey_b[6] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};

main()
{
    uchar idata tt[2];
    uchar idata size;
    uchar idata bankdata[16];
    ulong idata value=1;
    uchar i,j;

    zlg500_RST=1;           //使ZLG500复位
    for(i=255;i>0;i--)
        for(j=255;j>0;j--);
    zlg500_RST=0;
    for(i=255;i>0;i--)
        for(j=255;j>0;j--);
    baud_num=3;           //波特率为57600
    serial_init();
    EA=1;
    i=mifs_config();      //ZLG500配置
    i=mifs_get_info(bankdata); //读信息
    memcpy(RC500_snr,&bankdata[8],4); //存RC500序列号
    mifs_load_key(KEYA,5,Nkey_b); //装载密钥
    while(1)
    {
        while(mifs_request(IDLE,tt)!=0); //请求
        if(mifs_anticoll(0,card_snr)!=0) continue; //防碰撞
        if(mifs_select(card_snr,&size)!=0) continue; //选择
        if(mifs_authentication(KEYA,5)!=0) continue; //证实
        bankdata[0]=0x10;
        bankdata[4]=~0x10;
    }
}
```

```
bankdata[8]=0x10;
for(i=1;i<4;i++)
{
    bankdata[i]=0x00;
    bankdata[4+i]=0xff;
    bankdata[8+i]=0x00;
}
bankdata[12]=0x14;
bankdata[13]=~0x14;
bankdata[14]=0x14;
bankdata[15]=~0x14;
if(mifs_write(20,bankdata)!=0) continue; //写一个值块
if(mifs_check_write(card_snr,KEYA,20,bankdata)!=0) continue; //检查写
if(mifs_restore(20)!=0) continue; //恢复20块的数据
if(mifs_transfer(21)!=0) continue; //传送到21块
if(mifs_read(21,bankdata)!=0) continue; //读出
mifs_halt(); //使卡进入HALT状态
mifs_buzzer(198,20,0,1); //蜂鸣器口输出2KHz方波，持续200毫秒
}
}
```