

DS899 内嵌 IC 卡刷卡器电子柜锁 ZCHW485 通讯协议

第一节 DS899 内嵌 IC 卡刷卡器电子柜锁的简介

DS899 内嵌 IC 卡刷卡器电子柜锁是一把 IC 卡内嵌刷卡控制的电子锁,它由 IC 卡内嵌刷卡控制器、电磁阀、机械锁等组成的一体锁。



图 1 DS899 电子锁

特性:

额定电压: 12VDC

电流: 刷卡器电流小于 100mA, 开锁瞬间小于 600mA, 开锁持续电流小于 200mA

防护等级: IP54

IC 卡工作频率: 13.56MHZ

IC 卡协议标准: ISO 14443A

读卡时间: <0.3 秒

开户数: 100 张卡

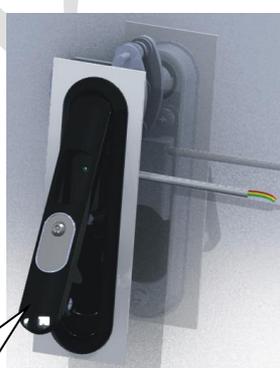
功能:

内嵌 IC 卡刷卡器, 自带控制电路, 刷卡独立开锁;

电子锁可通过 RS485 和控制器或电脑通讯, 可被远程监测和控制;

锁状态检测和被读取;

DS899 内嵌 IC 卡刷卡器电子柜锁开锁过程图:



绿灯闪按此
处弹起手柄

内嵌 IC 卡刷卡
器

图2 DS899 开锁过程图

DS899 内嵌 IC 卡刷卡器电子柜锁使用说明

IC 卡电子柜锁动作指示：红灯、绿灯及蜂鸣器。

- a、平时指示灯不亮和蜂鸣器不响；
- b、刷卡时蜂鸣器嘟一声短声，说明刷卡器动作，通过刷卡器自身对 IC 卡进行鉴权，如成功刷卡器嘟一声长声，绿灯亮，并闪烁，说明刷卡成功，可以开启柜锁，远程开锁时通过 RS485 通讯接口发回“刷卡成功信息”；如嘟嘟二声短声，说明卡不满足条件不能开锁；如嘟一声短声，说明锁内无卡。
- c、按动手柄下方的圆圈位置，手柄自动弹起，（如果不按，绿灯一直闪烁等待开启，直到控制线断电。）旋转手柄，带动钢栓，连杆等锁闭机构运动完成开启动作；柜锁锁把打开时，不管正常还是非正常打开红灯长亮；
- d、锁把关闭时，红灯闪烁 5 秒后关灯；

DS899 内嵌 IC 卡刷卡器电子柜锁安装方式，开孔尺寸



图3 安装示意图

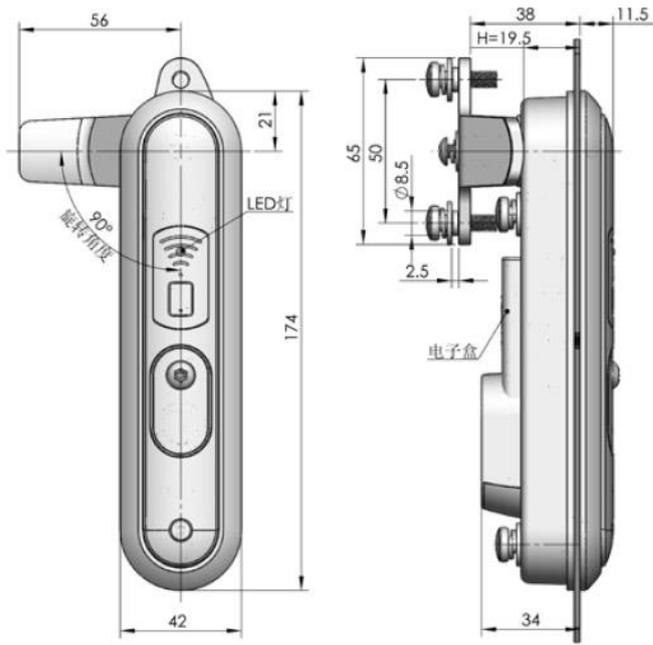
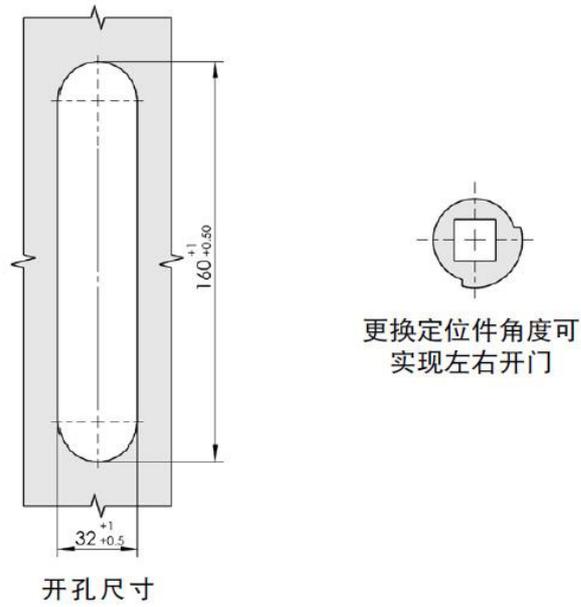


图 4 DS899 机构尺寸



开孔尺寸

图 5 开孔尺寸

第二节 485 通讯规则

1、 数据格式:

1 个起始位, 8 个数据位, 1 个停止位, 无校验;

2、 波特率:

9600

4、 帧格式:

帧头	目的地址	源地址	目的进程号	源进程号	净荷长度	报文消息		CRC	帧尾
1 byte	2 byte	Signal 2 byte	消息参数	2 byte	2 byte				

5、 帧格式说明:

- ● 帧头/帧尾
帧头为 0x7E;
帧尾为 0x7E 0x7E;
0x7D 作为转义符; 帧数据中的 0x7E, 使用 0x7D 0x5E 两个字符来替代; 帧数据中的 0x7D, 使用 0x7D 0x5D 两个字符来替代;
- ● 目的地址/源地址
数据的来源和目的地址, 0x0 ~ 0xFE 是设备地址, 0xFF 为广播地址
- ● 目的进程号、源进程号位置保留, 数据暂不使用
- ● 净荷长度为报文消息的长度
- ● CRC 为净荷校验值
- ● 报文消息 (data)

Signal:

取值:

3-指示灯闪烁

4-指示灯停止闪烁

无消息参数

6、 CRC 说明:

CRC 校验内容: CRC 校验是对范围为目的地址到报文消息的全部数据的校验, 其字节数为净荷长度+6。详见第五节。

第三节 上位机发送命令(数字为 16 进制数)

1) 鉴权成功命令---开锁(远程开锁命令)

Signal 取值: 0005---鉴权成功 无消息参数 净荷长度: 2

[7E0101000000020005C1777E7E](#)

返回: Signal 取值: 0005---鉴权成功

消息参数: 01—成功;

2) 鉴权失败—闭锁(远程闭锁命令)

Signal 取值: 0006---鉴权失败 无消息参数 净荷长度: 2

7E0101000000020006F1147E7E

返回: Signal 取值: 0006---鉴权失败

消息参数: 01—成功;

3) 查询锁状态\刷卡卡号及其他信息等事件命令

Signal 取值: 0016---查询锁事件 无消息参数 净荷长度: 2

7E0101000000020016E3257E7E

返回: Signal 取值: 0016---查询锁事件

锁状态	RFID 卡号	保留	保留	保留	保留	锁事件	卡有效
1 byte	4 byte	3 byte	1 byte				

消息参数:

锁状态:

01—锁柄开启; 00---锁柄关闭

无卡时, RFID 卡号为: 00000000 授权还是未授权都传输卡号

锁事件:

- 1) “正常开门” ----- A5
- 2) “非正常开门” ----- A9
- 3) “正常开门后关门” ----- A6
- 4) “非正常开门后关门” ----- A7

卡有效: 1---授权卡 2---非授权卡

4) 清空卡

控制器发送清空卡指令, 本指令将清除 DSS899 里所有卡。

Signal 取值: 00e2—清空卡

无消息参数

净荷长度: 2

返回: Signal 取值: 00e2—清空卡

消息参数: 01—成功; 00—失败;

5) 加卡 (最多发卡 100 张卡)

控制器发送加卡指令, 本指令将向 DS899 里增加卡。

Signal 取值: 00e3—加卡

消息参数: 4 字节卡号

净荷长度: 6

返回: Signal 取值: 00e3—加卡
消息参数: 01—成功; 11—满; 00—失败; 14--已有

6) 删除卡

控制器发送删除卡指令, 本指令将在 DS899 里删除本卡。

Signal 取值: 00eb—删除卡

消息参数: 4 字节卡号

净荷长度: 6

返回: Signal 取值: 00eb—删除卡

消息参数: 01—成功; 00—无此卡; 04--删除有误

7) 机号设定命令 (只允许连接一台内嵌刷卡器)

Signal 取值: 00d3---设定机器号

消息参数: 机器号—1 字节

7Efff0101000300D3012e3a7E7E

7E01010000000300D30267797E7E

返回: Signal 取值: 00d3---设定机器号

消息参数: 01—成功; 00—失败

8) 读取机号命令 (只允许连接一台内嵌刷卡器)

Signal 取值: 00d4—读取机器号

7E01010000000200D41A2B7E7E

返回: Signal 取值: 00d4—读取机器号

消息参数: 机器号—1 字节

9) 初始化机器命令

Signal 取值: 00d5—初始化机器

7E01010000000200D50A0A7E7E

返回: Signal 取值: 00d5—初始化机器

消息参数: 01—成功; 00—失败

10) 参数设定命令

Signal 取值: 00e5---设定参数

消息参数:

区号—1 字节 保留—1 字节 延时—1 字节

返回: Signal 取值: 00e5---设定参数

消息参数: 机号—1 字节 保留—1 字节 保留—1 字节 延时—1 字节

11) 读取参数

Signal 取值: 00e6—读取参数

7E01010000000200e61A2B7E7E

返回: Signal 取值: 00e6—读取参数

消息参数: 机号—1 字节 保留—1 字节 保留—1 字节 延时—1 字节

第四节 接线方法:

DS899 内嵌 IC 卡电子柜锁的引线定义:

红 (12V)、黑 (GND)、黄 (485-)、白 (485+) 4 根线;

第五节 CRC 函数:

```
//  
//用软件计算 CRC 函数  
//  
void CalculateCRCbySoft(unsigned char *pucData,unsigned char wLength,unsigned char  
*pOutData)  
{  
    unsigned char ucTemp;  
    unsigned int wValue;  
    unsigned int crc_tbl[16]={0x0000,0x1021,0x2042,0x3063,0x4084,0x50a5,0x60c6,0x70e7,  
    0x8108,0x9129,0xa14a,0xb16b,0xc18c,0xd1ad,0xe1ce,0xf1ef}; //四位余式表  
  
    wValue=0;  
  
    //本字节的 CRC 余式等于上一字节的 CRC 余式的低 12 位左移 4 位后,  
    //再加上上一字节 CRC 余式右移 4 位 (也既取高 4 位) 和本字节之和后所求得的 CRC 码  
  
    while(wLength--!=0)  
    {  
        //根据四位 CRC 余式表, 先计算高四位 CRC 余式  
  
        ucTemp=((wValue>>8))>>4;  
        wValue<<=4;  
        wValue^=crc_tbl[ucTemp^((*pucData)>>4)];  
        //再计算低四位余式  
        ucTemp=((wValue>>8))>>4;  
        wValue<<=4;  
        wValue^=crc_tbl[ucTemp^((*pucData)&0x0f)];  
        pucData++;  
    }  
    pOutData[0]=wValue;  
    pOutData[1]=(wValue>>8);  
}
```