票务专用在线式 TD01 设备通讯协议

基础版本设备: TD01

在现有 TR700 考勤版本上改进要求如下:

1、大屏界面布局:



说明:

- 右边方框内为固定内容显示区,显示日期时间和机号
- 左边为六行可自定义内容显示区
- 文字为透明背景色
- 背景图可自己画,但是布局不能改变

2、设备动作结构体 MAC_ACTIVE[256]

● Beep[1]: 蜂鸣器发音 1 字节:

分为高 4 位和低 4 位

高 4 位取值:

0(0000)-不动作,此时低 4 字节内容无效

1(0001)-连续长鸣,此时低 4 字节为发音时长,单位为秒,取值 1-15

2(0010)-连续报警音,此时低 4 字节为发音时长,单位为秒,取值 1-15

3(0011)-短鸣,此时低 4 字节为时长,单位为 100 毫秒,取值 1-15

比如刷卡成功提示,采用短鸣 500ms,则设置为 00110101 (0x35)

比如刷卡失败提示,连续报警 1 秒,则设置为 00100001 (0x21)

- Relay[4]: 四个继电器状态: 0-不动作, 1-动作(动作时间由参数决定)
- Img_id[1]:背景图 ID
- Line1Color[2]: 第1行文字颜色(RGB 16bit 值)
- Line1Text[21]: 第1行20字节+结尾符
- Line2Color[2]: 第 2 行文字颜色(RGB 16bit 值)
- Line2Text[21]: 第 2 行 20 字节+结尾符

- Line3Color[2]: 第 3 行文字颜色(RGB 16bit 值)
- Line3Text[21]: 第 3 行 20 字节+结尾符
- Line4Color[2]: 第 4 行文字颜色(RGB 16bit 值)
- Line4Text[21]: 第 4 行 20 字节+结尾符
- Line5Color[2]: 第 5 行文字颜色(RGB 16bit 值)
- Line5Text[21]: 第 5 行 20 字节+结尾符
- Line6Color[2]: 第 6 行文字颜色(RGB 16bit 值)
- Line6Text[21]: 第 5 行 20 字节+结尾符
- Speak[41]: 语音播报内容(最长 20 个汉字)
- ST[71]: 预留长度

3、心跳包: (包含时钟校对功能)

设备主动上传5字节:

0x31, JH1, JH2, XOR, 0xFF

例如:

应答内容: 35 发给服务的心跳包 应答包: 31002312FF 解释: 31 开头 00 23 机号 12 XOR 校验

无屏幕刷新软件应答 12 字节:

0x31, JH1, JH2, YY, MM, DD, HH, NN, SS, WW, XOR 0xFF

结尾

例如:

应答内容:

FF

机号 35 时间 2015/10/31 09:53:25 星期六

应答包:

3100230F0A1F093519062BFF

解释:

| 01 | | | | | | | 开头 |
|----|----|----|----|----|----|----|----------|
| 00 | 23 | | | | | | 机号 |
| 0F | OA | 1F | 09 | 35 | 19 | 06 | 年月日时分秒星期 |
| 2B | | | | | | | XOR 校验 |
| FF | | | | | | | 结尾 |

有屏幕刷新应答 268 字节:

0x32, JH1, JH2, YY, MM, DD, HH, NN, SS, WW, MAC_ACTIVE[256], XOR, OxFF 说明

0x01: 心跳包命令(固定)

JH1, JH2: 表示设备机号, 2字节数值型

YY, MM, DD, HH, NN, SS: 年月日时分秒, 年份取后两位

WW: 星期(1-7代表周一到周日) MAC_ACTIVE:设备动作指令结构体 XOR: 前面所有字节的异或校验 0xFF: 数据包结束符

例如:

应答内容:

机号 35 时间 2015/10/31 09:56:27 星期六

应答包:

| 解释: | | | |
|--|---|---|---------|
| 32 | <mark>开头</mark> | | |
| 00 23 | <mark>机号</mark> | | |
| OF OA 1F O9 35 19 06 | 年月日时分秒星期 | | |
| 00 | 蜂鸣器不发音 | | |
| 01 | 继电器 1 有动作 | | |
| 01 | 继电器 2 有动作 | | |
| 01 | 继电器 3 有动作 | | |
| 01 | 继电器 4 有动作 | | |
| 01 | 背景图 ID(背景图文件 | ‡名字 0,1,2,3) | |
| E007 | | 第1行文字颜色:红色 | |
| 2020202020202020D4C2BFA | 8000000000000000000 | 第1行20字节+结尾符 | (00) |
| E007 | | 第2行文字颜色:红色 | |
| 20202020203036333735323 | 53334310000000000000 | 第2行20字节+结尾符 | (00) |
| E007 | | 第3行文字颜色:红色 | |
| 202020202020BBB6D3ADB9E | 2C1D90000000000000000 | 第3行20字节+结尾符 | (00) |
| E007 | | 第 4 行文字颜色:红色 | |
| 20202020202020C7EBBDF8C | 8EB000000000000000000 | 第 4 行 20 字节+结尾符 | (00) |
| E007 | | 第5行文字颜色:红色 | |
| 2020202020202020202020000 | 000000000000000000000000000000000000000 | 第5行20字节+结尾符 | (00) |
| E007 | | 第6行文字颜色:红色 | |
| 20202020202030393A35363 | A32360000000000000000 | 第6行20字节+结尾符 | (00) |
| BBB6D3ADB9E2C1D90000000 | 000000000000000000000000000000000000000 | 000000000000000000000000000000000000000 | 0000000 |
| 000000000000 语音 | 音播报内容(41字节,最 | 最长 20 个汉字) | |
| 20 | 02020202020202020202020 | 020202020202020202020 | 2020202 |
| | | | |

预留长度(**71** 字节) XOR 校验 结尾

44 FF

4、事件包

设备上传: 249 字节

0x33, JH1, JH2, EID, CID[241], RTYPE[1], ST[1], XOR, 0xFF

EID:事件 ID, 定义如下

0x10=INP1 事件

0x11=INP2 事件

0x12=INP3 事件

0x13=INP4 事件

0x20=M1 或 EM 卡事件

0x21=维庚读头1事件

0x22=维庚读头 2 事件

0x30=身份证

0x31=条码1事件

0x32=条码 2 事件

CID: 卡号,身份证或者条码数据 RTYPE: 设备参数中设置的代码 应答包同心跳包的有屏幕刷新应答

例如:

卡号包内容:

机号 16130 票号 00041FDF1AA04380

卡号包:

040FF

解释:

33开头3F 02机号

31 条码 1 事件

字节+结尾符,其中刷卡十位长度不够前面补零,二维码 0-40 位长度

00 设备标示(例如0可以代表进1代表出,在终端参数设置)

```
    00
    预留

    40
    XOR 校验

    FF
    结尾
```

5、终端参数调整,增加待机6行默认显示内容

参数设置:

下行: 0x39, JH1, JH2, UART_SYS_PARAM [256], CRCL, CRCH

上行:原包返回

参数读取:

下行: 0x3A, JH1, JH2, CRCL, CRCH

上行: 0x3A, JH1, JH2, UART_SYS_PARAM [256], CRCL, CRCH

终端参数结构体 UART_SYS_PARAM {256}

```
typedef struct
    card_format;
                 // 卡号格式, 0=4字节,1=三字节,2=二字节
u8
u8
    card order;
                  // 卡号顺序, 0=正序,1=反序
    rec_code;
                  // 自定义记录代码: 0^{\sim}15
u8
    card_disp_fmt; // 卡号显示格式, 0=十进制,1=十进制维庚,2=十六进制
u8
    wiegand_out_fmt; // 维庚输出格式, 0=关闭, 1=26 位, 2=32 位, 3=34 位
u8
    disp_time; // 刷卡显示时间,1~255
u8
T_LCD_TEXT idle_txt[6]; // 待机显示内容结构体
    logo[21];
                 //开机显示内容
}UART SYS PARAM;
```