

锁控板通讯协议和规格书

(适用于 8 12 18 24 36 50 位锁的锁控板)

(485 通讯系列)

一：硬件接口：

通讯方式：485 多机通讯，同一 485 总线上可挂 1~99 台设备，

波特率 9600，数据为 8 停止为 1，流控 None

开锁命令为 16 进制的数据格式 hex

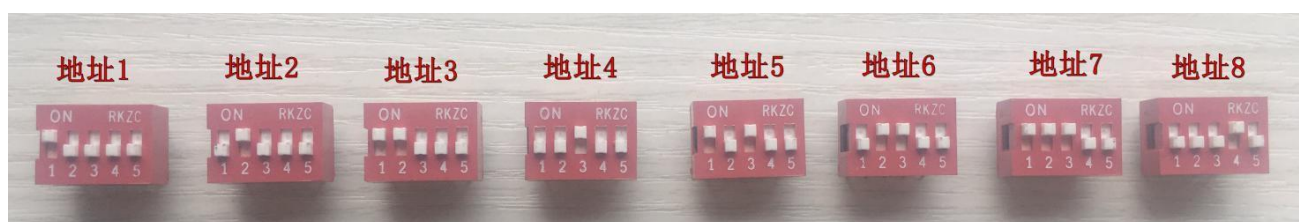
工作电压 12~24 伏 DC

控制电控锁/电磁锁

(2020 年版本)

二 地址开关设定图

最多支持 32 片锁控板串联使用、



三：全开命令：

该命令实现一条命令 打开一个板子上的所有电控锁，

| 命令头 | 板地址 | 锁地址 00 | 开锁功能码 | 校验 BCC/异或 |
|-----|-----|--------|-------|-----------|
| 8A | 01 | 00 | 11 | 9A |

例如上位机发送 8A 01 00 11 9A 地址 1 的锁板接收到此命令后 1 号板子上所有的锁以间隔 300 毫秒的速度 1 到最后一个全开，执行完毕后向下一个锁板就是 02 锁板发送全开命令 8A 02 00 11 99 根据自己的地址接力控制下一片板子的全开，此功能的作用是假设用户级联了 10 片锁控板，用户检测测试的时候只是需要发送控制开始全开的锁板地址发送数据就可以，总线上的锁板会自动执行全开到最后一个，实现的原理是，第一个锁板全完后第一个锁板控制第二个锁板全开，第二个锁板执行完后二控制三……………直到最后一个锁板发出控制下一个锁板的命令时候，并没有实际物理连接的下一个锁板时候结束，每一个锁板执行完全开命令，并发送指令控制下一个锁板以后，自身均待命执行下一条功能命令。

四 开锁命令

| 命令头 | 板地址 | 锁地址 | 开锁功能码 | 校验 BCC/异或 |
|-----|-----|-----|-------|-----------|
| 8A | 01 | 01 | 11 | 9B |

板地址就是，锁控板上的拨码

开关设置的地址。锁地址就是

需要打开锁的编号。

开锁功能码固定为 11

如：上位机发8A 01 01 11 9B（16进制）500毫秒后返回如下数据。

反馈的数据包括板号锁编号以及开锁的当前状态。

五：反馈数据格式如下：

| 命令头 | 板地址 | 锁地址 | 开锁状态 11/00 | 校验 BCC/异或 |
|-----|-----|-----|------------|-----------|
| 8A | 01 | 01 | 00 | 9B |

备注：反馈数据包括了板地址。锁地址，和开锁的状态。开锁的状态又和锁是有关系的。锁分为开门反馈型号和关门反馈型号。如果有开门反馈的锁开门后反馈状态是 11 表示打

开，关门反馈的锁开门反馈 00 表示打开 11 则表示开锁失败。推荐使用开门反馈的锁。例如开门反馈的锁，在开锁后反馈

8A 01 01 00 9B（开锁失败）

8A 01 01 11 9B（开锁成功）

六 读锁状态命令（门开关状态反馈）查询锁状态

（查询单个锁状态）

| 命令头 | 板地址 | 锁地址 | 查询功能码 | 校验 BCC/异或 |
|-----|-----|-----|-------|-----------|
| 80 | 01 | 01 | 33 | B3 |

如：上位机发80 01 01 33 B3（16进制）200秒后锁控板反馈如下数据

| 命令头 | 板地址 | 锁地址 | 反馈的锁状态 | 校验 BCC/异或 |
|-----|-----|-----|--------|-----------|
| 80 | 01 | 01 | 00 | 91 |

在锁状态自己中开门反馈的锁查询状态反馈 11 表示打开，关门反馈的锁查询状态反馈 00

表示打开 11 则表示当前锁状态为关闭。

0X80 0X01 0X01 0X00 0X91 (锁为关)

0X80 0X01 0X01 0X11 0X80 (锁为开)

七： 读取所有锁状态

在实际使用中用户需要知道每一组柜子所有锁的状态，因此就有了查询所有锁状态，针对一张锁控板上的所有锁状态进行查询。

读取所有锁状态：

| 命令头 | 板地址 | 固定字符 | 功能码 | 校验 BCC/异或 |
|-----|-----|------|-----|-----------|
| 80 | 01 | 00 | 33 | B2 |

例如上位机十六进制发送 80 01 00 33 B2 将读取 1 号板的所有锁

状态反馈的如下反馈的数据：

| 命令头 | 板地址 | 状态 1 | 状态 2 | 状态 3 | 功能码 | 校验 BCC |
|-----|-----|-------|------|------|-----|--------|
| 80 | 01 | 17-24 | 9-16 | 1-8 | 33 | XX |

以 24 路锁控板为例，反馈所有锁状态有三个字节状态，其中

状态 3 为锁 1—8 的锁状态 01 02 04 08 10 20 40 80 八个锁状态

状态 2 为锁 9—16 的锁状态 01 02 04 08 10 20 40 80 八个锁状

状态 1 为锁 17—24 的锁状态 01 02 04 08 10 20 40 80 八个锁状态

例：插上开门反馈的锁如反馈的数据是 80 01 02 01 04 33 B5 则表示锁板 3

号 9 号 18 号锁为打开状态

八 开多个通道锁命令

一条命令选择打开多个锁：

| 命令头 | 板地址 | 状态 1 | 状态 2 | 状态 3 | 校验 |
|-----|-----|-------|--------|---------|----|
| 90 | 01 | 锁 1-8 | 锁 9-16 | 锁 17-24 | XX |

例如上位机发 90 01 02 02 02 93（16 进制）表示开

启 2、10、18 通道锁。执行完该命令后反馈所有锁的状态，

格式参照读所有锁反馈的命令。

八：主动反馈

主动反馈就是，关上门 锁状态发生变化后主动上传到上

位机关门信息，此功能个别产品不带、

| 命令头 | 板地址 | 锁地址 | 锁状态 | 校验 BCC/异或 |
|-----|-----|-----|-------|-----------|
| 81 | 01 | 01 | 11/00 | ** |

例如：1 号板子的 2 号锁使用开门反馈的锁开锁后反馈的

数据是：81 01 02 00 82 其中状态位是 00，因为开门反
馈的锁反馈开关是开锁接通，关锁断开。

当关锁或者锁状态发生变化的时候，主动上传给上位机，
锁状态即表示当时状态，如果开门接通的锁关锁反馈的状
态为 00，如果关门反馈的电控锁关锁 后反馈的状态是 11

九：长通电功能：

该功能适用于控制电磁锁，即持续通电给锁锁产生磁力锁
门，断电磁性消失解锁。该功能个别产品不带。

上位机发数据通电：

| 命令头 | 板地址 | 锁地址 | 功能码固定 | 校验 BCC/异或 |
|-----|-----|-----|-------|-----------|
| 9A | 01 | 01 | 11 | ** |

例如上位机发送：9A 02 01 11 88 执行后是 2 号锁板的 1

号锁长通电。并反馈回传该通道的反馈状态。

上位机发送数据关闭长通电

| 命令头 | 板地址 | 锁地址 | 功能码固定 | 校验 BCC/异或 |
|-----|-----|-----|-------|-----------|
| 9B | 01 | 01 | 11 | ** |

上位机发送 9B 02 01 11 89 锁板执行结果是关闭 2 号锁

板的 1 通电长通电，并反馈上传该通道当前状态。

十：修改开锁通电输出时间：

该功能实现随意更改每一个开锁通道的通电输出时间，时
间范围是 1——9 秒。默认输出时间是 500 毫秒。实现方式
如下：

| 命令头 | 板地址 | 锁地址 | 功能码 11 | 开锁时间 1-9 | 校验 |
|-----|-----|-----|--------|----------|----|
| 8A | 01 | 01 | 11 | 03 | 98 |

该命令实现 1 号锁板 1 号锁通电 3 秒关闭，该功能不影响反馈，反馈回来的数据不变，如无特殊需求不要使用此功能，如果是电控锁可能烧坏电控锁电磁铁。

十一：修改通讯波特率：

| 命令头 | 板地址 | 固定字符 | 功能码 | 校验 BCC/异或 |
|-----|-----|------|-----|-----------|
| 9D | 01 | 00 | 01 | B2 |

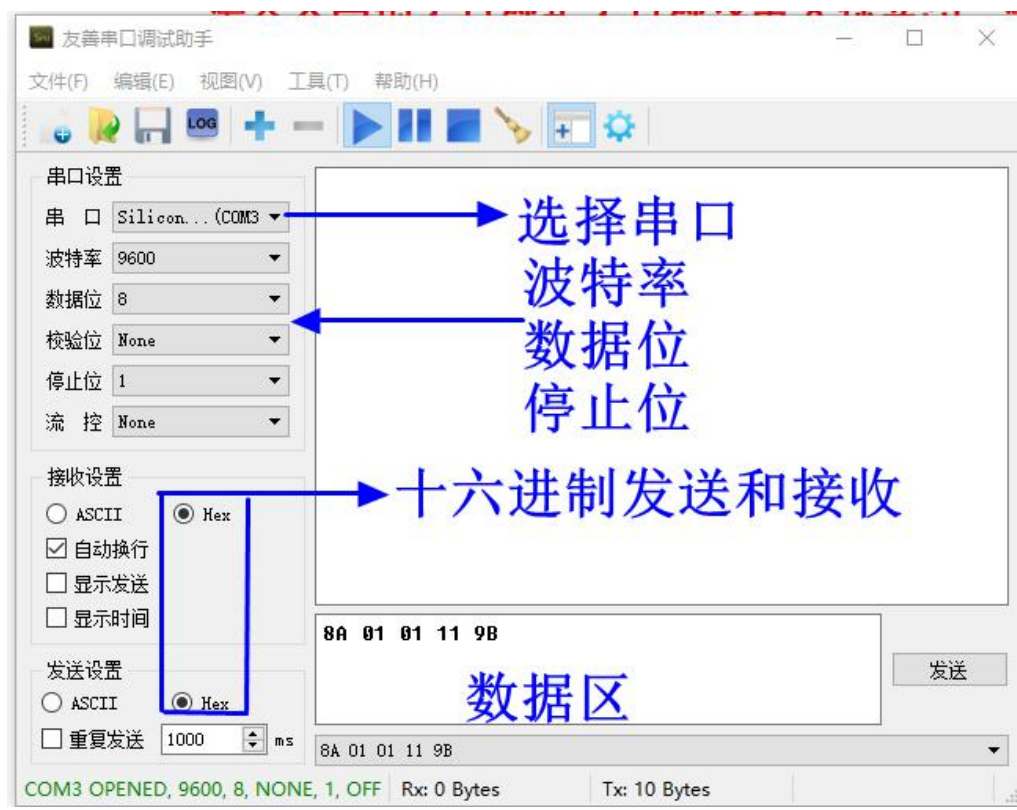
其中：功能码 01 代表 9600 02-19200 03-38400
04-57600 05-115200

十二：关闭主动反馈功能：

| 命令头 | 板地址 | 固定字符 | 功能码 | 校验 BCC/异或 |
|-----|-----|------|-----|-----------|
| 9C | 01 | 00 | 01 | B2 |

功能码：01 打开主动反馈，02 关闭主动反馈，断电保存

十二：串口测试



校验方式：

```
#include <stdio.h>

/*
使用在线工具计算 485 输入的值
*/

unsigned char txbuff[6]={0x90, 0x01, 0x01, 0x11, 0x01, 0x00};

unsigned char CRC_XOR(unsigned char *buff, unsigned char len)
```

```

{
    unsigned char i;
    unsigned char temp;
    for(i=0; i<len; i++)
        temp ^= buff[i];
    return temp;
}

int main () {
    unsigned char i, j;
    for(i=0x00; i<0x19; i++)
    {
        txbuff[2] = i;
        txbuff[5] = CRC_XOR(txbuff, 5);

        for(j=0; j<6; j++ )
        {
            if(txbuff[j]<0x10)
                printf("0%x
", txbuff[j]);

            else
                printf("%x ", txbuff[j]);
        }
        printf("\n");
    }
    return 0;
}

```

////////////////////////////////////

通过协议计算的开锁数据：针对 1 号锁板的数据：

全开：8A 01 00 11 9A

| | | | |
|----------------|--------|----------------|--------|
| 8A 01 01 11 9B | 1 通道 | 8A 01 02 11 98 | 2 通道 |
| 8A 01 03 11 99 | 3 通道 | | |
| 8A 01 04 11 9E | 开通道 4 | 8A 01 05 11 9F | 开通道 5 |
| 8A 01 06 11 9C | 开通道 6 | 8A 01 07 11 9D | 开通道 7 |
| 8A 01 08 11 92 | 开通道 8 | 8A 01 09 11 93 | 开通道 9 |
| 8A 01 0A 11 90 | 开通道 10 | 8A 01 0B 11 91 | 开通道 11 |
| 8A 01 0C 11 96 | 开通道 12 | 8A 01 0D 11 97 | 开通道 13 |
| 8A 01 0E 11 94 | 开通道 14 | 8A 01 0F 11 95 | 开通道 15 |
| 8A 01 10 11 8A | 开通道 16 | 8A 01 11 11 8B | 开通道 17 |
| 8A 01 12 11 88 | 开通道 18 | 8A 01 13 11 89 | 开通道 19 |

| | | | |
|-------------------------|--------|----------------|--------|
| 8A 01 14 11 8E | 开通道 20 | 8A 01 15 11 8F | 开通道 21 |
| 8A 01 16 11 8C | 开通道 22 | 8A 01 17 11 8D | 开通道 23 |
| 8A 01 18 11 82 | 开通道 24 | | |
| 80 01 00 33 B2 | 读所有通道 | | |
| 反馈 80 01 08 00 00 33 BA | | | |

其中 FE 对应 8-1 即 0000 1000b, 表示锁 4 打开

| | | | |
|-------------------|--------|----------------|--------|
| 80 01 01 33 B3 | 读通道 1 | | |
| 反馈 80 01 01 11 91 | | 80 01 02 33 B0 | 读通道 2 |
| 80 01 03 33 B1 | 读通道 3 | 80 01 04 33 B6 | 读通道 4 |
| 80 01 05 33 B7 | 读通道 5 | 80 01 06 33 B4 | 读通道 6 |
| 80 01 07 33 B5 | 读通道 7 | 80 01 08 33 BA | 读通道 8 |
| 80 01 09 33 BB | 读通道 9 | 80 01 0A 33 B8 | 读通道 10 |
| 80 01 0B 33 B9 | 读通道 11 | 80 01 0C 33 BE | 读通道 12 |
| 80 01 0D 33 BF | 读通道 13 | 80 01 0E 33 BC | 读通道 14 |
| 80 01 0F 33 BD | 读通道 15 | 80 01 10 33 A2 | 读通道 16 |
| 80 01 11 33 A3 | 读通道 17 | 80 01 12 33 A0 | 读通道 18 |
| 80 01 13 33 A1 | 读通道 19 | 80 01 14 33 A6 | 读通道 20 |
| 80 01 15 33 A7 | 读通道 21 | 80 01 16 33 A4 | 读通道 22 |
| 80 01 17 33 A5 | 读通道 23 | 80 01 18 33 AA | 读通道 24 |

////////////////////////////////////
硬件接口针脚定义



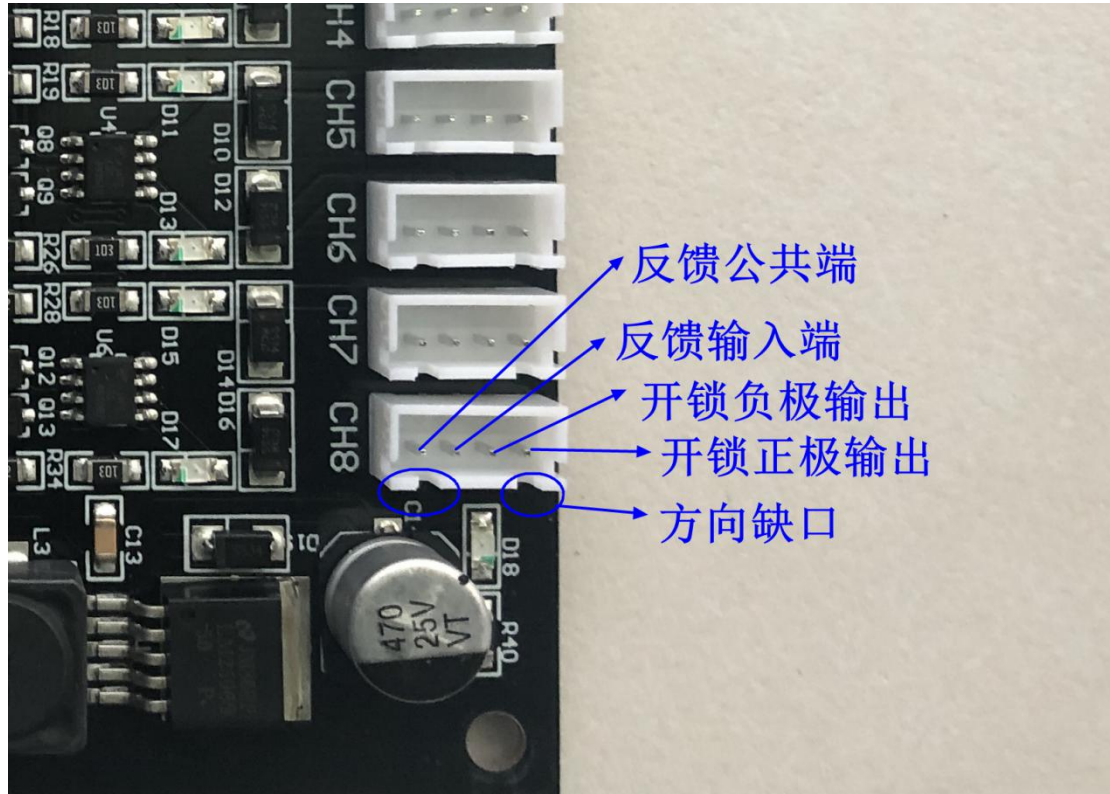
这个是锁控板的通讯和电源输入端子，按照图片中的端子圆圈标记点，分别是

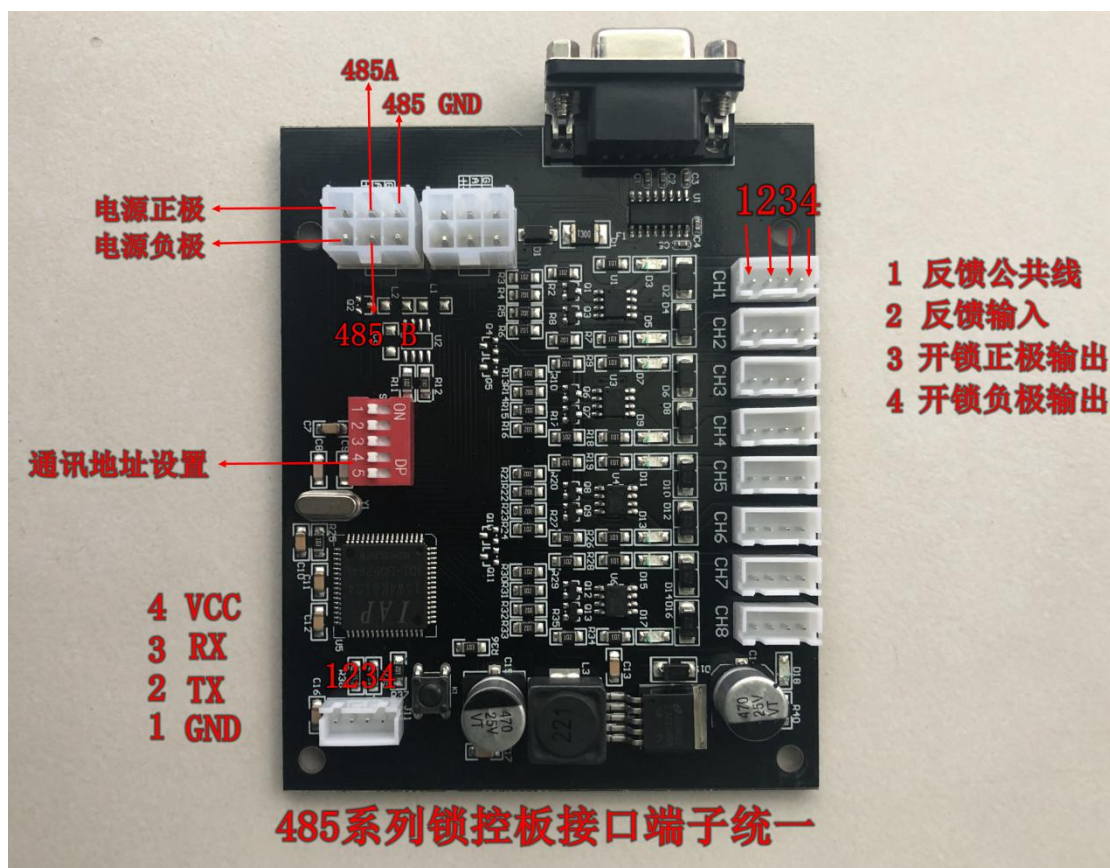
| | |
|---|------------|
| 1 | 电源 12——24V |
| 2 | 电源负极 |
| 3 | 485 通讯 A |

- | | |
|---|------------|
| 4 | 485 通讯 B |
| 5 | 485 通讯 GND |

相同的两个端子的功能是一样

电控锁端的针脚定义：



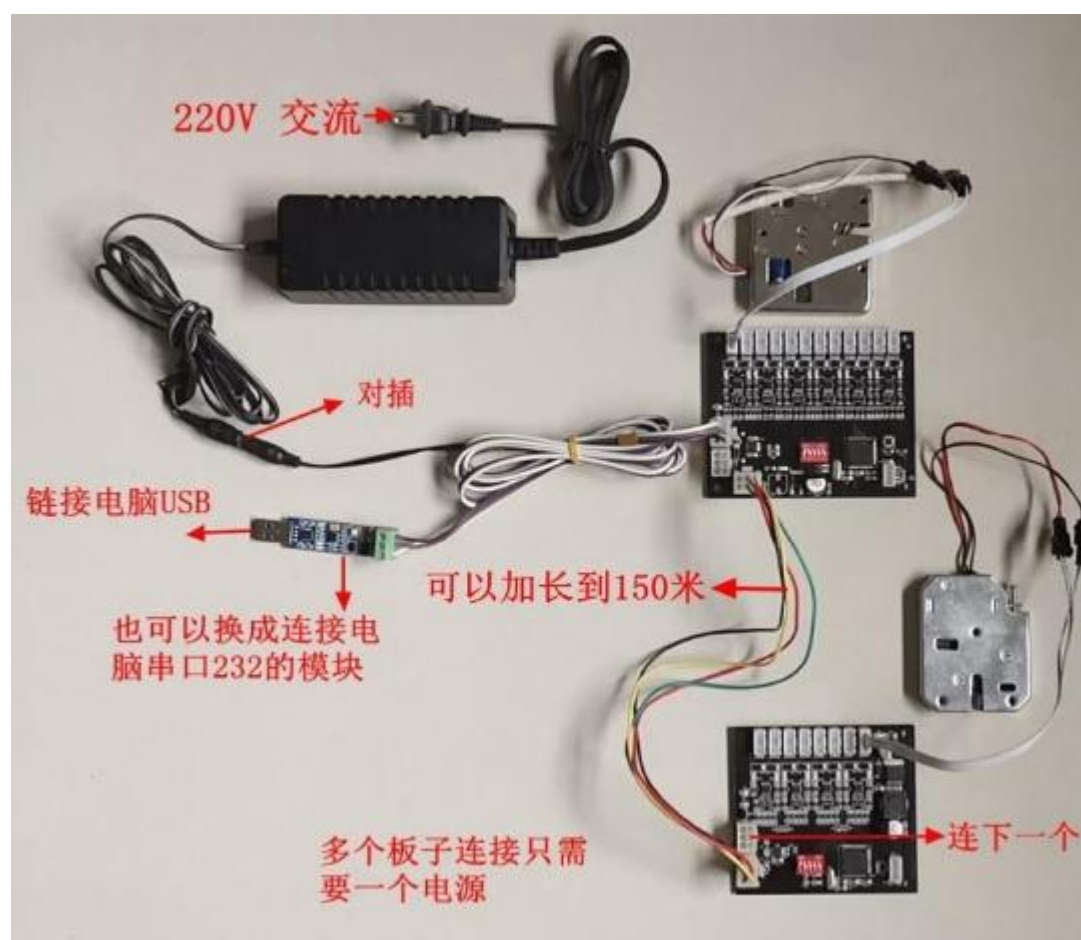
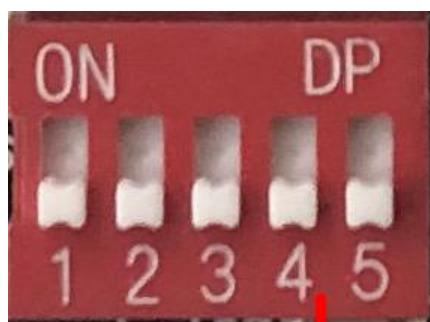


通讯地址设置图

| 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|----|
| 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 2 |
| 1 | 1 | 0 | 0 | 0 | 3 |
| 0 | 0 | 1 | 0 | 0 | 4 |
| 1 | 0 | 1 | 0 | 0 | 5 |
| 0 | 1 | 1 | 0 | 0 | 6 |
| 1 | 1 | 1 | 0 | 0 | 7 |
| 0 | 0 | 0 | 1 | 0 | 8 |
| 1 | 0 | 0 | 1 | 0 | 9 |
| 0 | 1 | 0 | 1 | 0 | 10 |
| 1 | 1 | 0 | 1 | 0 | 11 |
| 0 | 0 | 1 | 1 | 0 | 12 |
| 1 | 0 | 1 | 1 | 0 | 13 |
| 0 | 1 | 1 | 1 | 0 | 14 |
| 1 | 1 | 1 | 1 | 0 | 15 |
| 0 | 0 | 0 | 0 | 1 | 16 |
| 1 | 0 | 0 | 0 | 1 | 17 |
| 0 | 1 | 0 | 0 | 1 | 18 |
| 1 | 1 | 0 | 0 | 1 | 19 |
| 0 | 0 | 1 | 0 | 1 | 20 |
| 1 | 0 | 1 | 0 | 1 | 21 |

| | | | | | |
|---|---|---|---|---|----|
| 0 | 1 | 1 | 0 | 1 | 22 |
| 1 | 1 | 1 | 0 | 1 | 23 |
| 0 | 0 | 0 | 1 | 1 | 24 |
| 1 | 0 | 0 | 1 | 1 | 25 |
| 0 | 1 | 0 | 1 | 1 | 26 |
| 1 | 1 | 0 | 1 | 1 | 27 |
| 0 | 0 | 1 | 1 | 1 | 28 |
| 1 | 0 | 1 | 1 | 1 | 29 |
| 0 | 1 | 1 | 1 | 1 | 30 |
| 1 | 1 | 1 | 1 | 1 | 31 |

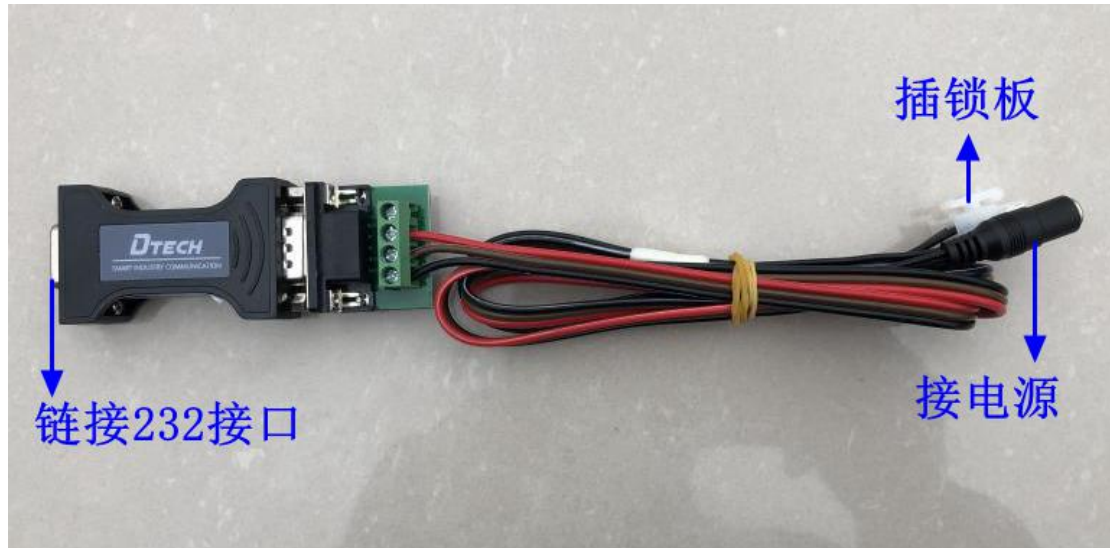
其中 1 2 3 4 5 是锁板拨码开关的 12345 拨到 ON DIP 一边是 1 拨到 12345 这边



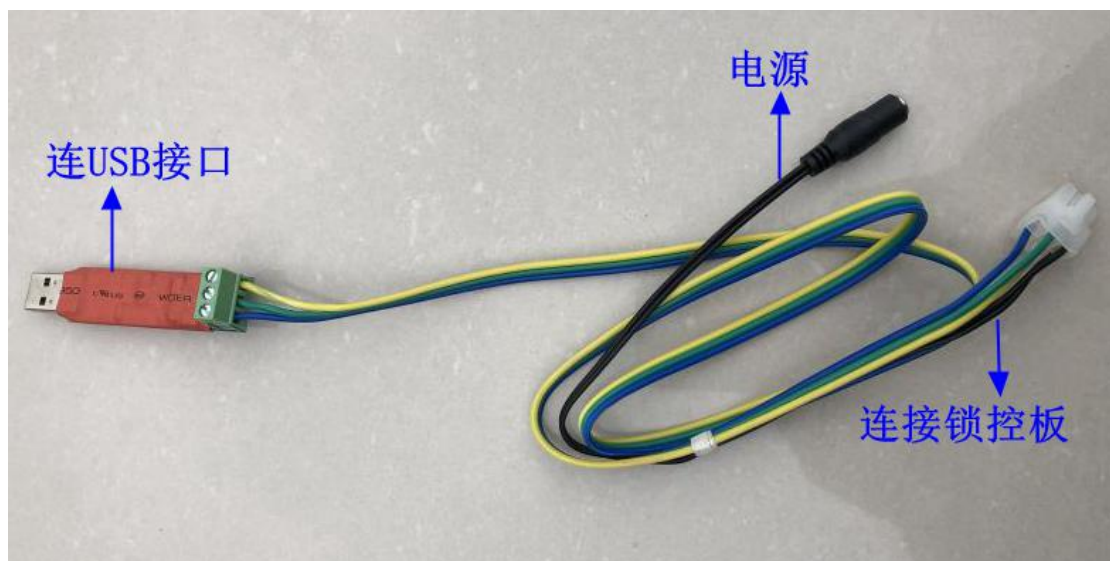
关于锁控板和上位机连接

锁控板和上位机连接可以通过上位机的 232 接口连 USB 接口连，485 接口连接和 TTL 接口连接，锁板出厂均不含线材，需要用户自己购买连线材。

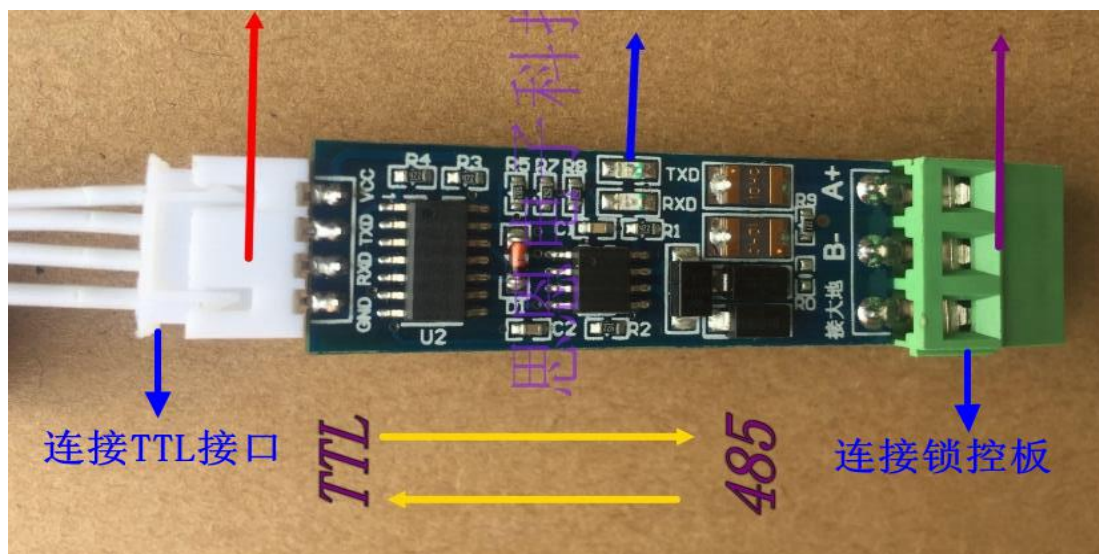
使用主机的 232 接口连锁控板输入第一个锁控板使用以下连接设备



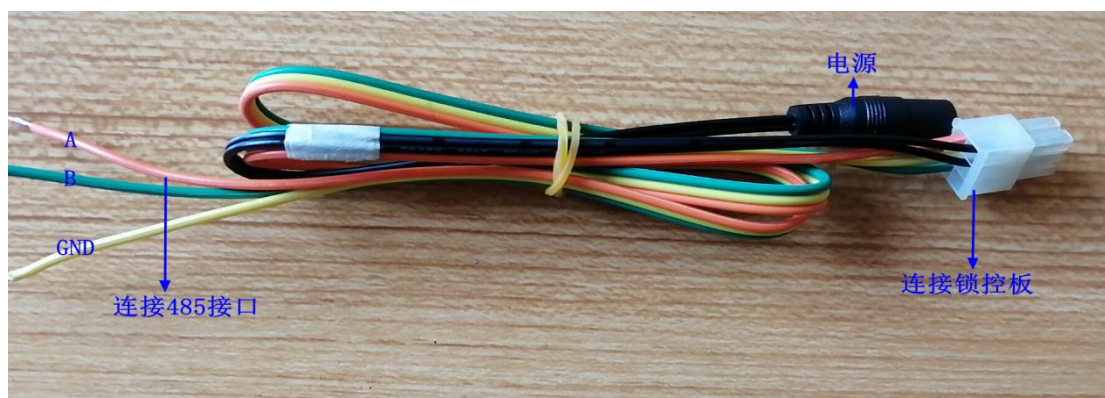
使用主机 USB 接口连接第一个锁控板使用以下设备



使用主机 TTL 接口连接第一个锁控板使用以下设备



使用主机 485 接口链接锁控板 使用以下设备



如果需要将两个锁控板级联需要如下线束



锁控板与电控锁之间的连接

