

高压模拟器控制协议

版本	作者	完成日期	备注
v1.1	lbs	2018-08-13	适用于新版高压模拟器 v1.1.0

基本信息

STM32F407 芯片+ UCOS-II 组成系统构平台

通信方式

采用 UDP 网络通信，出厂默认 IP 192.168.1.100 端口号 10000，可通过 KL_iMaC（科梁接口系统配置软件进行查看、更改）

控制包协议

be be be be be be be be cmd len (内容) crc ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed
总长度 = 8 + 1 + 1 + len + 1 + 8 + 8 = 27+len
len = 内容的字节数
crc = 内容的和校验
ff 保留字

继电器 RELAY 控制

对 Hvs 的继电器 RELAY 控制分为两个步骤

1、对想要控制的继电器进行配置,采用 11 个字节对所有继电器进行控制，每个 BIT 位代表一个 RELAY 继电器，低位优先，十六进制表示

be be be be be be be be 01 0b byte0 byte1 byte2 byte3 byte4 byte5 byte6 byte7 byte8 byte9
byte10 crc ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed
crc：1 字节，是 byte0 ~ byte10 的累加和校验，
byte0：对应 RELAY1~8，即 bit0 = RELAY1 ~ bit7 = RELAY8
.
.
.
Byte11：对应 RELAY81~86，即 bit0 = RELAY1 ~ bit5 = RELAY86

Hvs-v1.1.0 版本，用户可控 RELAY 列表如下：

2	3	5	8
11	16		
17	18	19	20 21 22 23 24
25	26	27	28 29 30 31 32
33	34	35	36 37
78	79	80	
81	82	83	84 86

2、激活配置

发送如下固定包

be be be be be be be be 02 01 01 01 ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

主正、主负电阻设置

对 Hvs 的电阻控制同样分为配置、激活两个步骤，

1、电阻配置

用户可设主正电阻值范围 150~50428850 Ω

RELAY38：主正电阻总开关，

RELAY39~57 决定主正电阻值，

假如需要电阻值为 resZ ，

则换算成配置字节算法为 $res_set = (resZ - 150) / 100$

将 res_set 赋值到对应字节进行配置

用户可设主负电阻值范围 150~50428850 Ω

RELAY58：主负电阻总开关，

RELAY59~77 决定主负电阻值，

假如需要电阻值为 resF ，

则换算成配置字节算法为 $res_set = (resF - 150) / 100$

将 res_set 赋值到对应字节位进行配置

2、电阻激活

发送如下固定包

be be be be be be be be 02 01 01 01 ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

示例程序

继电器 RELAY 控制配置算法示例

```
QByteArray relay_data;
relay_data.resize(38);
for (int i = 0; i < 8; i++)
{
    relay_data[i] = 0xbe;
}

relay_data[8] = 0x01;
relay_data[9] = 0xb; //len

int sCnt = 0;
for (int j=0; j<11; j++)
{
    relay_data[j+10] = 0;

    for (int k=0; k<8; k++)
    {
        sCnt++;
        switch (sCnt)
        {
            case 2:case 3:case 5:case 8:
            case 11:case 16:
            case 17:case 18:case 19:case 20:case 21:case 22:case 23:case 24:
            case 25:case 26:case 27:case 28:case 29:case 30:case 31:case 32:
            case 33:case 34:case 35:case 36:case 37:
            case 78:case 79:case 80:
            case 81:case 82:case 83:case 84:case 86:
            {
                if (relaytable->item(j, k)->checkState() == Qt::Checked)
                {
                    relay_data[j + 10] = (relay_data[j + 10] | (1 << k));
                }
            }
            break;

            default:
                break;
        }
    }
}
```

```

        crc += relay_data[j+10];
    }

    relay_data[21] = crc;

    for (int i = 22; i < 30; i++)
    {
        relay_data[i] = 0xff;
    }
    for (int i = 30; i < 38; i++)
    {
        relay_data[i] = 0xed;
    }

```

主正电阻配置算法示例

```

QByteArray relay_data;
relay_data.resize(38);
for (int i = 0; i < 8; i++)
{
    relay_data[i] = 0xbe;
}
for (int i = 8; i < 38; i++)
{
    relay_data[i] = 0;
}

relay_data[8] = 0x01;
relay_data[9] = 0xb;//len

if (resZ == 0)
{
    relay_data[14] = relay_data[14] & ~(1 << 5); //relay38 主正总开关
}
else
{
    relay_data[14] = relay_data[14] | (1 << 5);
    resZ = resZ - 150;
    resZ = resZ / 100;
}

if (resZ & 0x1)

```

```

{
    relay_data[14] = relay_data[14] | (1 << 6);
}
if ((resZ >> 1) & 0x1)
{
    relay_data[14] = relay_data[14] | (1 << 7);
}

for (int j = 0; j < 8; j++)
{
    if ((resZ >> (j+2)) & 0x1)
    {
        relay_data[15] = relay_data[15] | (1 << j);
    }
}
for (int j = 0; j < 8; j++)
{
    if ((resZ >> (j + 10)) & 0x1)
    {
        relay_data[16] = relay_data[16] | (1 << j);
    }
}
if ((resZ >> 18) & 0x1)
{
    relay_data[17] = relay_data[17] | 1;
}

for (int j = 0; j<11; j++)
{
    crc += relay_data[j + 10];
}

relay_data[21] = crc;

for (int i = 22; i < 30; i++)
{
    relay_data[i] = 0xff;
}
for (int i = 30; i < 38; i++)
{
    relay_data[i] = 0xed;
}

```

主负电阻配置算法示例

```

QByteArray relay_data;
relay_data.resize(38);
for (int i = 0; i < 8; i++)
{
    relay_data[i] = 0xbe;
}
for (int i = 8; i < 38; i++)
{
    relay_data[i] = 0;
}

relay_data[8] = 0x01;
relay_data[9] = 0xb; //len

if (resF == 0)
{
    relay_data[17] = relay_data[17] & ~(1 << 1); //relay58 主负总开关
}
else
{
    relay_data[17] = relay_data[17] | (1 << 1); //relay58 主负总开关
    resF = resF - 150;
    resF = resF / 100;
}

for (int j = 0; j < 6; j++)
{
    if ((resF >> j) & 0x1)
    {
        relay_data[17] = relay_data[17] | (1 << (j+2));
    }
}

for (int j = 6; j < 14; j++)
{
    if ((resF >> j) & 0x1)
    {
        relay_data[18] = relay_data[18] | (1 << (j - 6));
    }
}

for (int j = 14; j < 19; j++)
{
    if ((resF >> j) & 0x1)
    {
        relay_data[19] = relay_data[19] | (1 << (j - 14));
    }
}

```

```
    }  
}  
  
for (int j = 0; j<11; j++)  
{  
    crc += relay_data[j + 10];  
}  
  
relay_data[21] = crc;  
  
for (int i = 22; i < 30; i++)  
{  
    relay_data[i] = 0xff;  
}  
for (int i = 30; i < 38; i++)  
{  
    relay_data[i] = 0xed;  
}
```