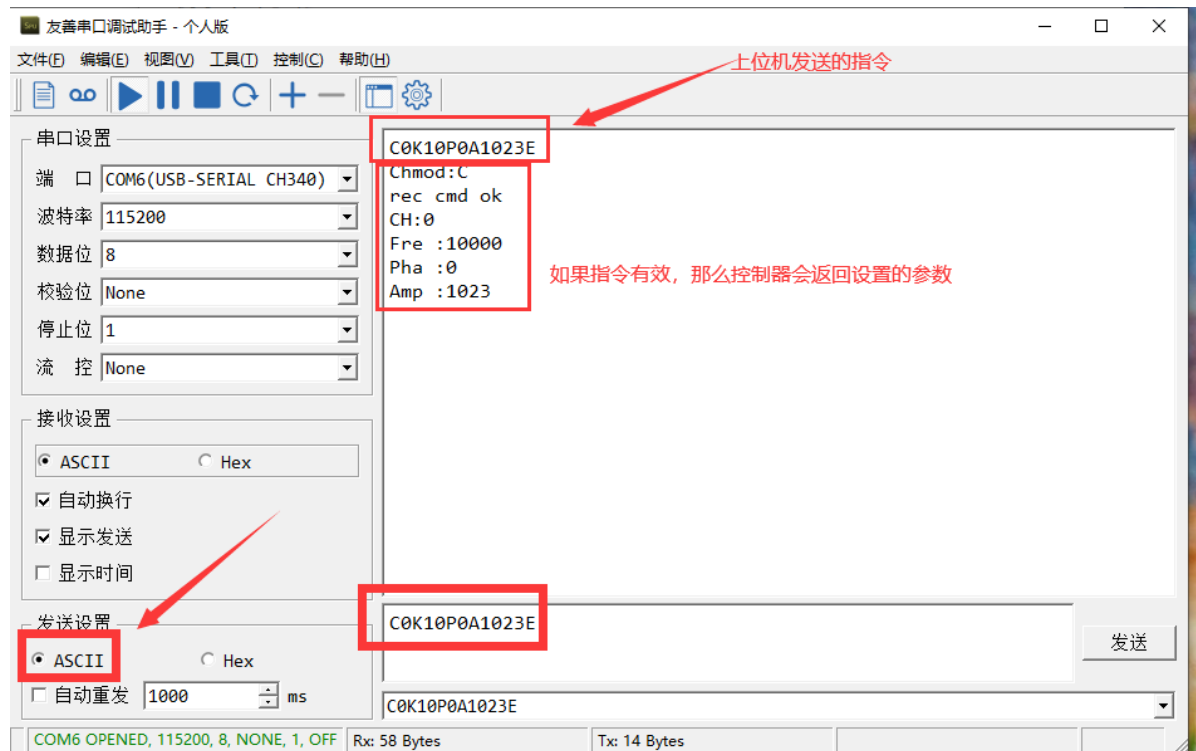


指令说明

程序烧录到板子上后，不管使用串口助手发送指令或者用其他板子发送指令给控制器，都要以ASCII码的形式发送，而不是发送16进制数据。

比如：设置 通道0 输出频率10Khz 相位偏置为0 以最大幅度输出

则指令应该为：C0K10P0A1023E



指令实际上由4位控制字符和一位结束字符组成

第一位控制字符C 表示这一条指令要设置的输出通道

C后面的数字表示输出通道编号 取值为0, 1, 2, 3

第二位控制字符K 表示这一条指令要设置的输出频率 第二位控制字符除了K以外，还可以是M和H

M表示设置的频率单位为Mhz

K表示设置的频率单位为Khz

H表示Hz

后面的数字表示输出频率 可以是小数

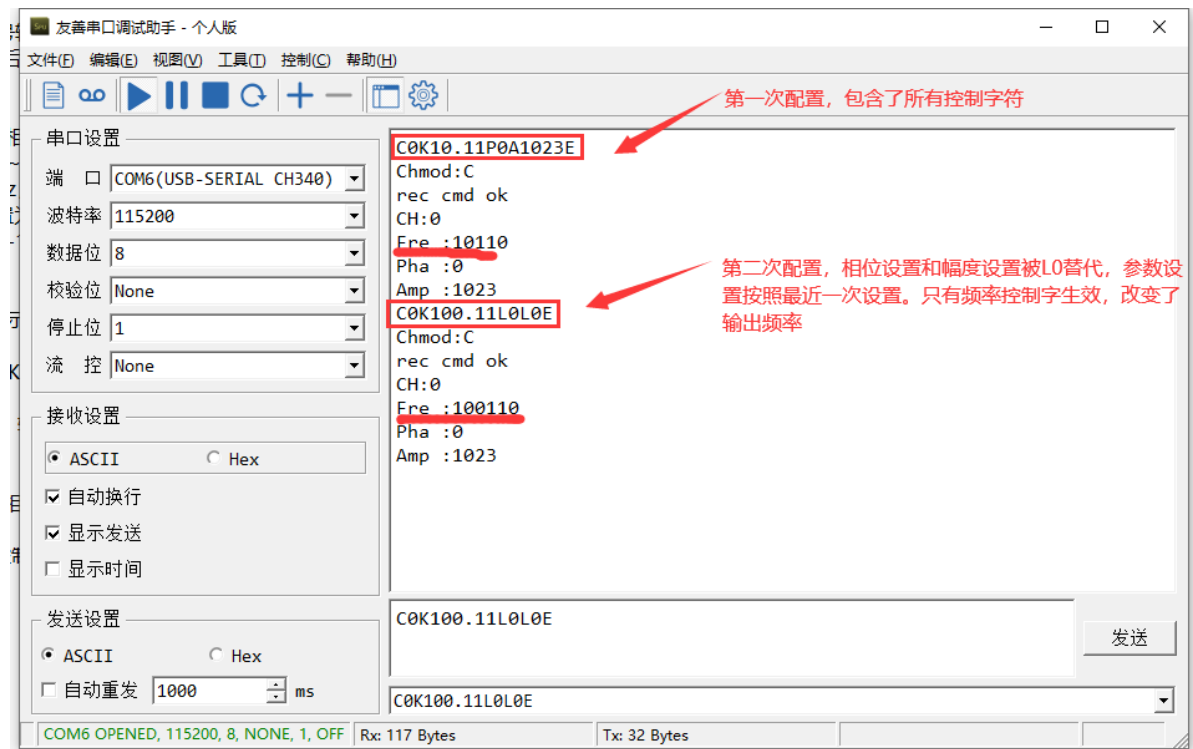
第三位控制字符P 表示这一条指令要设置的输出相位差 一般为0 参数范围为：0-16383 必须是整数 对应0-360°

第四位控制字符A 表示这一条指令要设置的输出幅度，实际没法设置具体的幅度大小，这是一个相对幅度，数据范围为0-1023 必须是整数

设置为1023，则表示以最大输出幅度输出

使用说明

实际使用时，一般情况下，我们只需要配置一次相位差，幅度，后续只需要修改频率即可
因此，在发送指令时，第一次的指令需要包含所有的控制字符，后续如果只需要修改频率就可以使用 L0 代替相位差和幅度关键字，表示使用最近的相位差和幅度配置。



如果不需要控制器返回参数确认信息，依照下图将指定代码注释掉即可

```
Parameter[CH][0] = (uint32_t)Fre;
}
if (Fremod == 'L') {
    Parameter[CH][0] = Parameter[CH][0];
}
if (Phamod == 'L') {
    Parameter[CH][1] = Parameter[CH][1];
} else {
    Parameter[CH][1] = (uint32_t)Pha;
}
if (Ampmod == 'L') {
    Parameter[CH][2] = Parameter[CH][2];
} else {
    Parameter[CH][2] = (uint32_t)Amp;
}
Channel_ParameterSet(CH, Parameter[CH][0], Parameter[CH][1], Parameter[CH][2]);
EEPROM_WriteCH(CH);

//如果不需要回显确认 可以把下面的输出语句注释掉
Serial.print("CH:");
Serial.println(CH);
Serial.print("Fre :");
Serial.println(Parameter[CH][0]);
Serial.print("Pha :");
Serial.println(Parameter[CH][1]);
Serial.print("Amp :");
Serial.println(Parameter[CH][2]);
}
while (Serial.read() >= 0) {} //清空串口缓存
}
}
}
```

保存完成

注意，相邻两次指令发送间隔至少大于200ms，否则会出现设置失败的情况