

# 720bc

一个朴素的充电器

## 开源许可协议：GPL v3

这个工程由各种开源代码组合而成，请参考各自的文件头。

原理图由控制小板测量所得，存在错误，知识产权由天宝公司所有。

## 接入使用

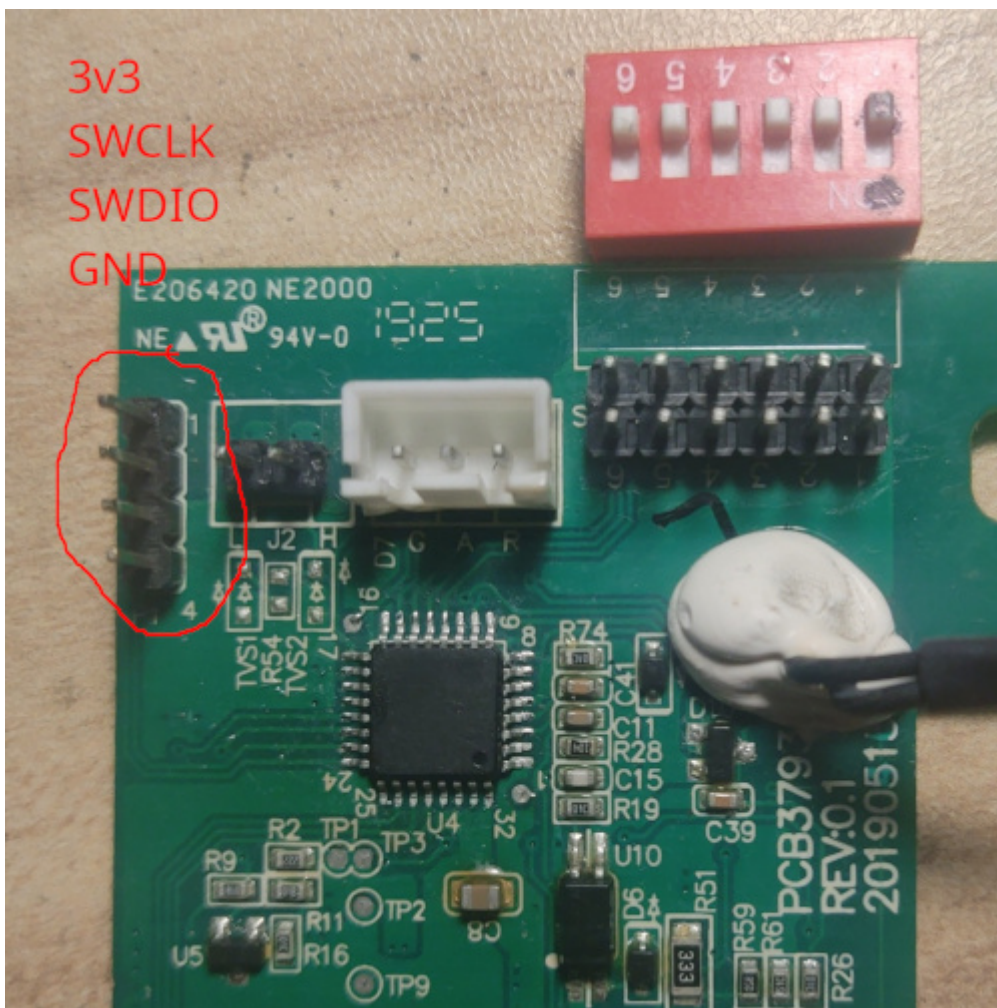
### 刷固件

推荐使用[便宜的stlink](#)



硬件连接下载器和控制小板, 连接4线即可，分别是

下载器	控制小板
3v3	3v3
GND	GND
SWDIO	SWDIO
SWCLK	SWCLK



- 请注意您的st-link上插针定义, 可能与上面图片上的不一致.
- 有群友反馈st-link的SWDIO/SWCLK是反的, 如果上面的连接方式不型用可以尝试交换一下
- 刷机过程中可以不连接充电器到220AC, 由st-link提供3v3供电
- 也可以使用jlink进行下载, 79元有点贵, 操作上也麻烦, 不推荐, 请自行研究.

下载需要使用stm32 编程软件: STM32 ST-LINK Utility 4.2 链接:

[https://pan.baidu.com/s/1\\_IOQvUGugtrA4ngggGN0Uw](https://pan.baidu.com/s/1_IOQvUGugtrA4ngggGN0Uw) 提取码: 83gp

预先编译的固件位于 server/Release/, 可以选择.hex或者.bin文件

其他可能用的上的资料

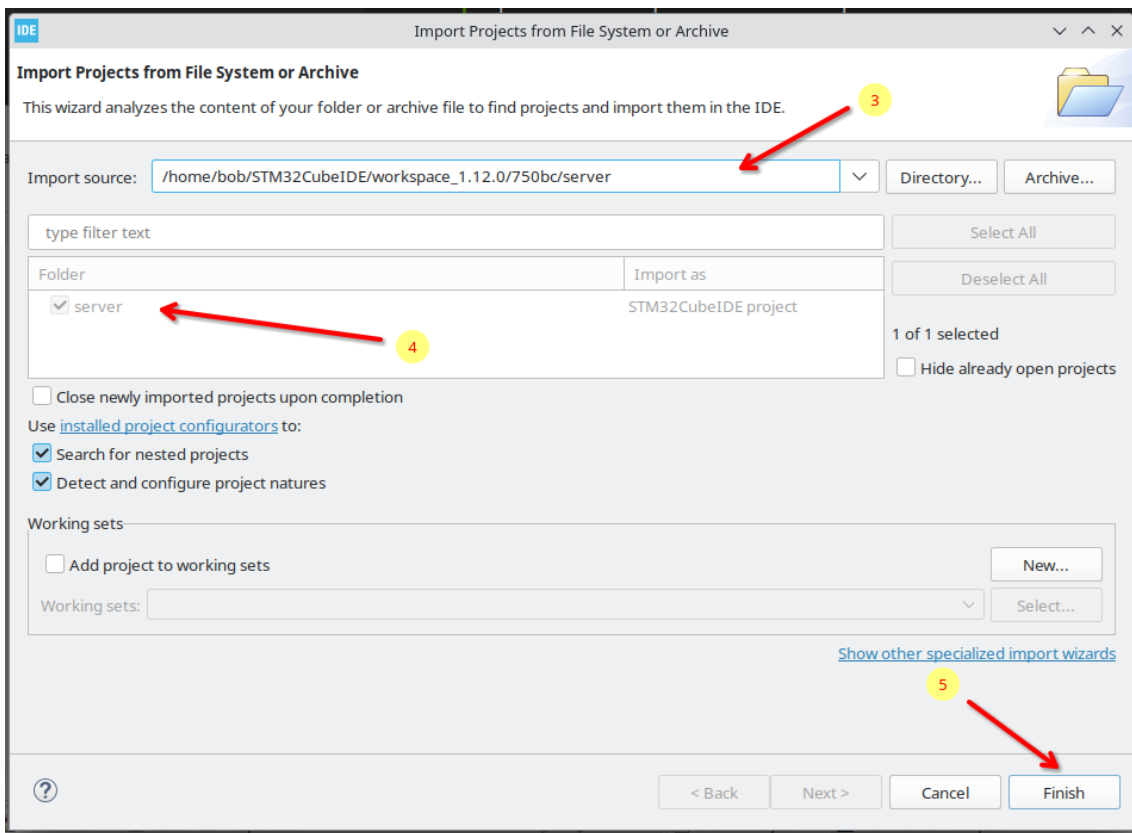
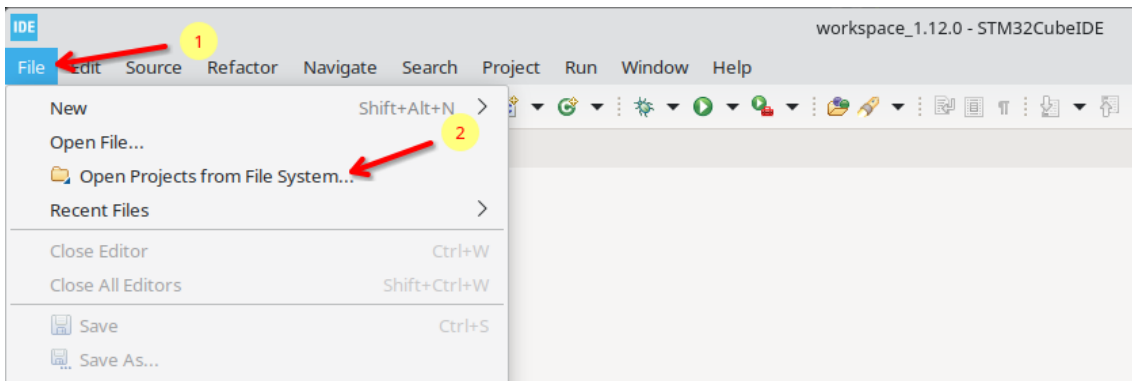
烧录视频教程: [https://v.youku.com/v\\_show/id\\_XNDY3MjkyNjgxNg==.html](https://v.youku.com/v_show/id_XNDY3MjkyNjgxNg==.html) 烧录图片教程:  
[https://blog.csdn.net/qg\\_39592312/article/details/109135564](https://blog.csdn.net/qg_39592312/article/details/109135564)

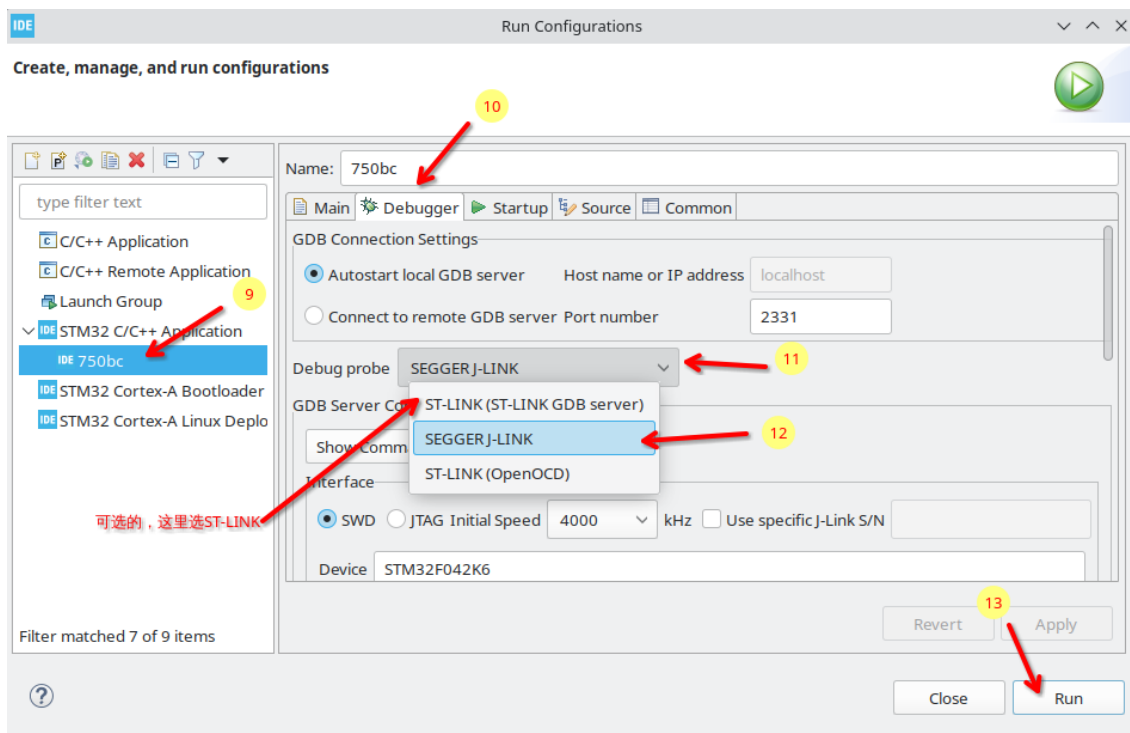
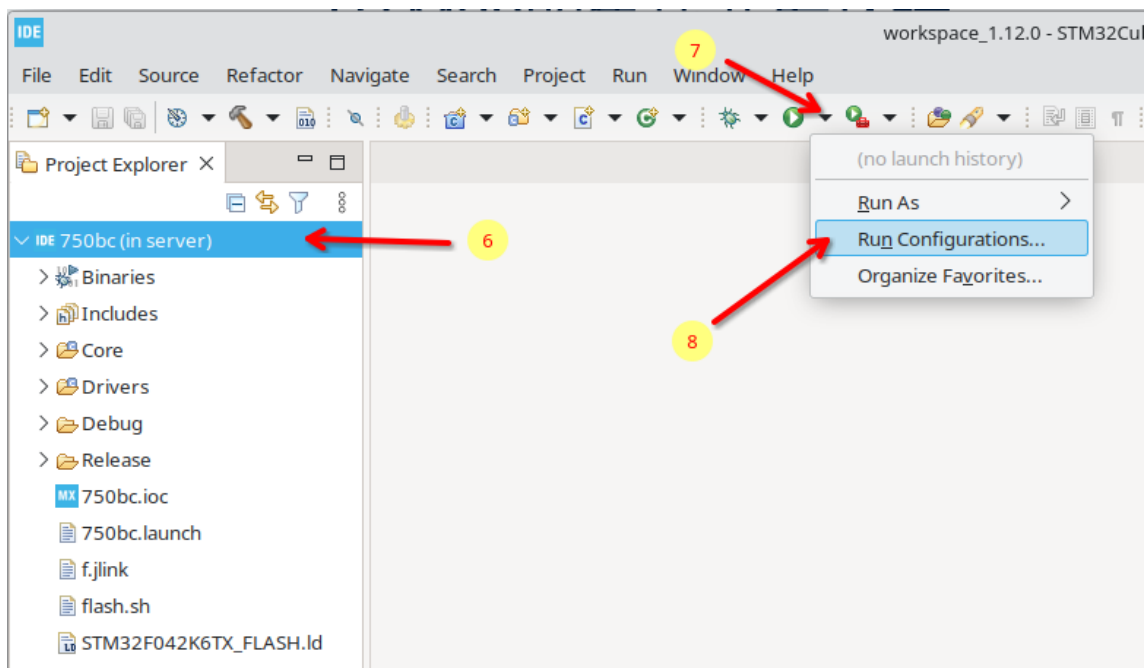
中文用户手册: 链接: [https://pan.baidu.com/s/1hYsy9EBEUzl0g\\_Qdb8ajVQ](https://pan.baidu.com/s/1hYsy9EBEUzl0g_Qdb8ajVQ) 提取码: qq69  
keil5.20下ST-Link V2设置说明-基于编程调试stm32 链接:  
<https://pan.baidu.com/s/1JWjwOZSOs882eLjSnLAEmg> 提取码: i10d

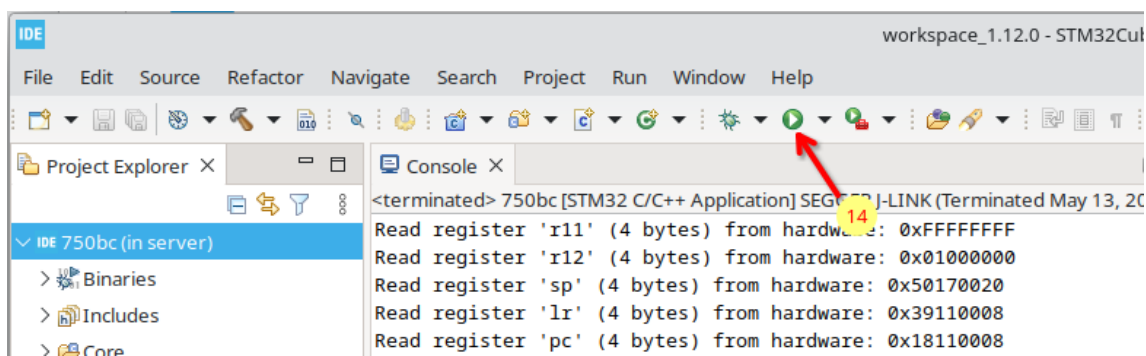
## 可选的, 进行二次开发

MCU对应的源码位于server文件夹, 可以修改和二次开发.

- 我使用的是[STM32的集成开发环境 STM32CubeIDE](#), 按照下面的步骤操作应该就可以刷程序了







- 刷机过程中会有窗口一闪而过，马上就提示成功了。

## 连接控制

这个控制小板的程序会使用UART向外通信, 可以使用 蓝牙模块 [doBT-M02模块 券后2.8元](#) 进行透传

度云物联
doiot

双模蓝牙模块 SPP 2.1+BLE 5.1  
doBT-M02

自研芯片、支持定制

AT透传固件

13x16mm, PCB天线

IoT、蓝牙从机

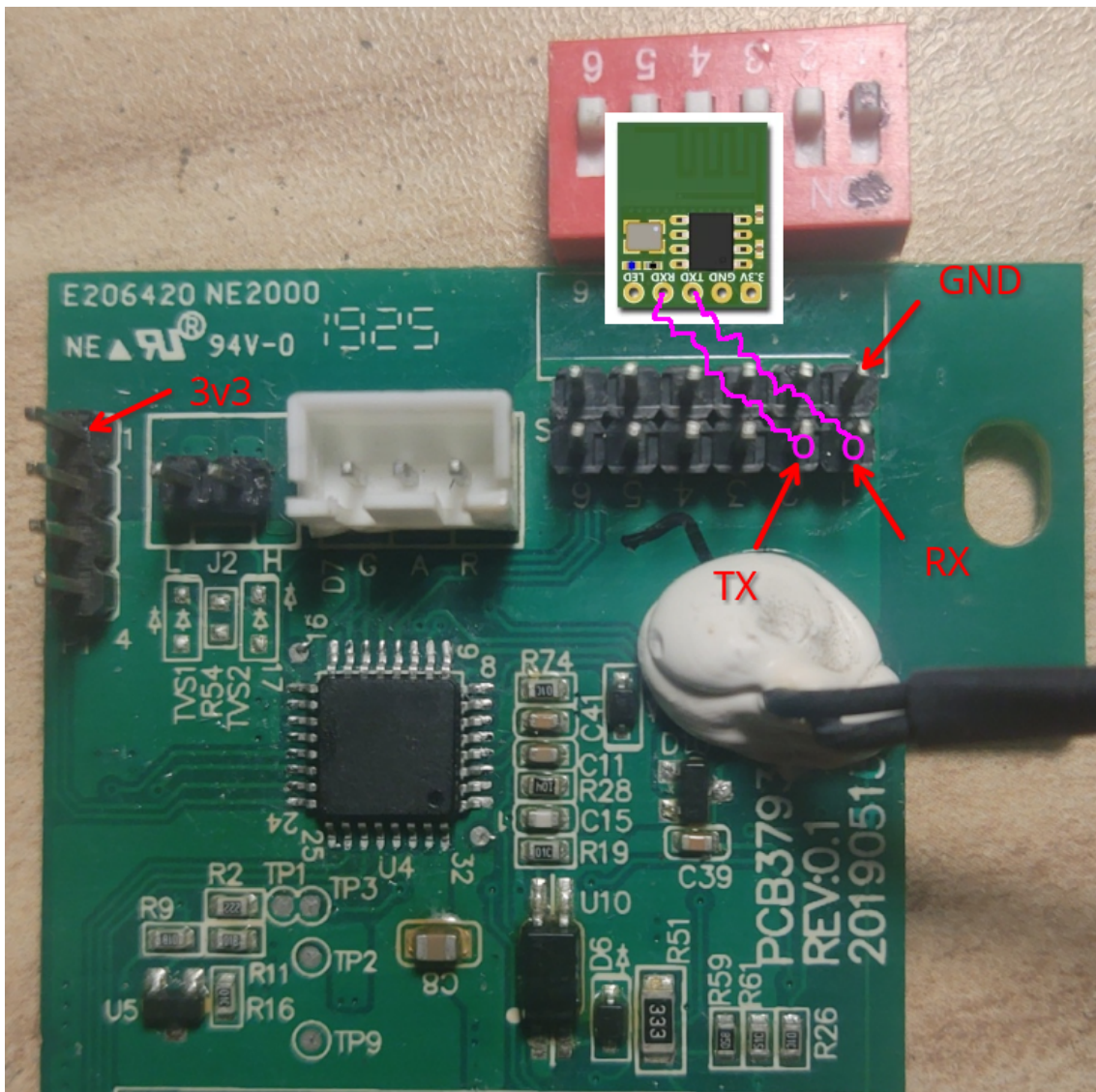
原理图 · PCB · 程序 · 视频教程  
开源资料、免费下载

旗舰店专享价:  
3.8<sup>¥</sup>

需要连接4线，分别是

蓝牙	控制小板
3.3v	3v3
GND	GND
TXD	RX
RXD	TX





- 引出UART接口，可以选择拆除拨码开关，或者直接焊线。我选择拆除拨码开关后装上排针。
- UART接口可以使用USB-TTL模块（如CH340，PL2303）在电脑上查看通信报文，协议见数据包格式章节。

手机android端控制程序已经预编译，位于750bc-controller/build/outputs/apk/debug/750bc-controller-debug.apk，其源码位于750bc-controller目录，可以使用[Android Studio](#)进行开发修改。

2:28



## Bluetooth terminal

not connected



Enter command here

SEND

RELAY ☐

FAN ☐

ON ☐

电压调节



电流调节



0.000V

0.000A

0.000V





2:28

# Bluetooth terminal

not connected



Enter command here

SEND

RELAY

FAN

ON

电压调节

电压调节

## Bluetooth terminal

No paired devices

PAIR NEW DEVICE...

0.

0.000A

0.000V

2:28



已连接的设备



与新设备配对

之前连接的设备



查看全部

连接偏好设置

蓝牙



向其他设备显示



2:28



## 与新设备配对



设备名称

可用的设备



BLE\_S750BC7201004



SPP\_S750BC7201005



手机的蓝牙地址：



2:29



## 与新设备配对



设备名称

可用的设备



BLE\_S750BC7201004

要与SPP\_S750BC7201005配对吗？

☐ 允许访问您的通讯录和通话记录

取消

配对

2:29



## 已连接的设备



其他设备



SPP\_S750BC7201005



与新设备配对

之前连接的设备



查看全部

连接偏好设置

蓝牙



向其他设备显示为



2:29

## Bluetooth terminal

not connected



Enter command here

SEND

RELAY ☐

FAN ☐

ON ☐

# 电压调节

### Bluetooth terminal

SPP\_S750BC7201005  
5A:74:8E:FD:C5:F5

PAIR NEW DEVICE...

电

0.

0.000A

0.000V







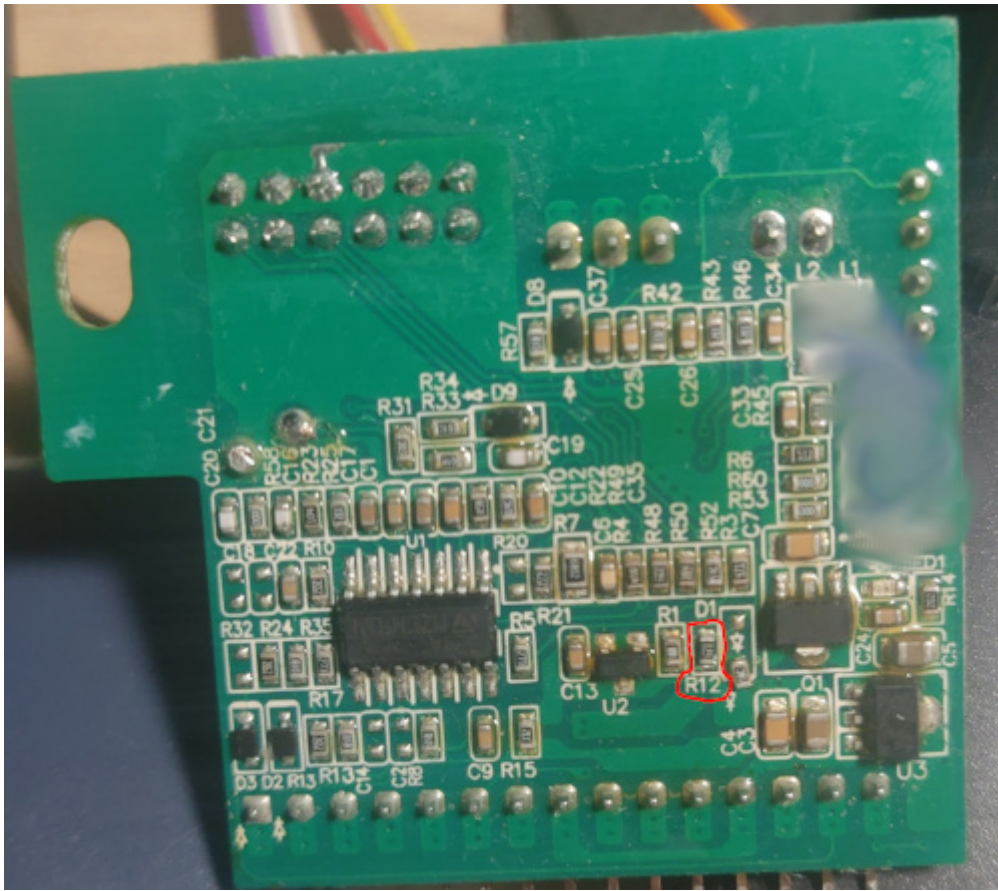
至此已经可以完成控制内容了

最高显示78v左右,



默认最低是30v上下, 如果还想再往下调节, 可以拆除R12, 或者将R12换成330K电阻, 最低可以到2.xV。

- 注意, 替换R12之后当电压占空比为100%时, 控制端显示为78v左右, 但实际电压为84v左右. 这是由于的采样电路的保护(D8)将单片机引脚限制在3.3v左右, 无法再高. 可以通过硬改降低输出电压采样的下臂(R52)的阻值来提升采样量程, 请不要忘记修改控制端的设置界面, 否则会显示偏小.





- 注意，电压低时，不太稳定. 有负载后，电压会稳定一点.

## 控制协议

### Server端

MCU为Server，其以UART（或其他）形式向外发送MCU状态数据。

client端

MCU接受client端（比如Android）发过来的控制数据。

数据包格式

数据包采用文本形式传递，每一行为一个数据包，对应格式为 CMD\_LIST\_LINE

```
CMD_LIST_LINE: ('>' | '<') CMD* ('\r' | '\n');
```

为了避免与现有的蓝牙模块的控制协议冲突, 特别在前面添加了 > 和 < 符号, 分别表示 单片机 向手机发送数据, 和手机向单片机发送数据。

CMD\_LIST\_LINE 中可以有多条命令，对应格式 CMD ， 每个命令包含 KEY 和可选的 VALUE 。

```
CMD: KEY VALUE?;
```

其中 KEY 的取值如下

```
KEY: 'R' | 'F' | 'O' | 'P' | 'T' | 'I' | 'B' | 'C' | 'V' | 'A' | 'c' | 'v' | 'M' | 'S' | 'D' | 'f';
```

VALUE 包含多个数字字符

```
VALUE: ('0' ... '9')+;
```

特别约定：

- 当 VALUE 不存在时， VALUE 为默认 0
- VALUE 数据值是小的在前面(LE)，比如串 "001" 代表的是十进制 100 CMD\_LIST\_LINE 需要按照先后顺序解析，可以包含重复的 KEY
- 当 VALUE 值为0时，可以编码为 v (推荐),或者 v0 ，或者 v000
- 串口中敲回车实际上是在数据后面添加了 \r\n 两个字符，在解析时会出现两个 CMD\_LIST\_LINE ，只是后面那个没有 CMD

举例：如果要发送命令，那么对应的命令行为

命令	编码（省略换行）
v=100	v001
v=100, c=30	v001c03
f=0, c=0, R=0	fcR

Server向Client发送数据

ID	功能	KEY	说明	默认值
RELAY	继电器	R	0关闭, 1开启	
FAN	散热器	F	0关闭, 1开启	

ON	初级	O	0关闭, 1开启	
PROTECTION	反接	P	0正常, 1反接	
ADC_TEMP	NTC采样电压	T	单位mV	
ADC_VREF	芯片内基准电压	I	单位mV	
ADC_VBATTERY	电池采样电压	B	单位mV	
ADC_CURRENT	电流采样电压	C	单位mV	
ADC_VOUT	主输出采样电压	V	单位mV	
ADC_AC_UV	??采样电压	A	单位mV	
PWM_CURRENT	电流占空比	c	% <sub>000</sub> (万分之)	
PWM_VOLTAGE	电压占空比	v	% <sub>000</sub> (万分之)	
MODE	工作模式	M	见工作模式	0
未实现 STAGE	充电阶段	S		
未实现 DONE	充电完成	D	0未完成, 1完成	

工作模式

值	功能	说明
0	供电模式	由用户自行控制电流,电压占空比
1	未实现 48v	自动进行 48v 电池充电
2	未实现 60v	自动进行 60v 电池充电
3	未实现 72v	自动进行 72v 电池充电

TODO, 电池类型不同, 可能要不同的充电方式: 铅酸 \ 锂电池 三元? 铁锂? TODO, 电池容量不同, 可能要不同的充电方式: 12AH \ 20AH \ 30AH TODO, 限制充电时间, 结束条件

ClientServer向发送数据

ID	功能	KEY	说明	默认值
RELAY	控制继电器	R	0断, 1合	0
ON	控制初级	O	0关闭, 1开启	0
fan	控制散热器	f	见散热器模式	2
PWM_CURRENT	控制电流占空比	c	% <sub>000</sub> (万分之)	0
PWM_VOLTAGE	控制电压占空比	v	% <sub>000</sub> (万分之)	0
WRITE	保存配置	W	1保存	



- 注意, 保存配置后, 下次上电将会使用之前保存的参数. 如果不做检测, 直接供电, 有可能导致电压/电流不匹配, 烧毁充电器或者受电设备, 甚至造成火灾.
- 注意, 当断开继电器时, 会同步断开初级
- 注意, 请避免在大电流情况下断开继电器, 这很容易烧毁继电器.

散热器模式

值	功能	说明
0	强制关闭	
1	强制开启	
2	自动	温度高于45°C开启风扇, 低于35°C关闭风扇
3	模式3	在自动模式的基础上, 当电流大于1.1A时开启风扇, 低于0.9A时关闭风扇

保护

高温保护

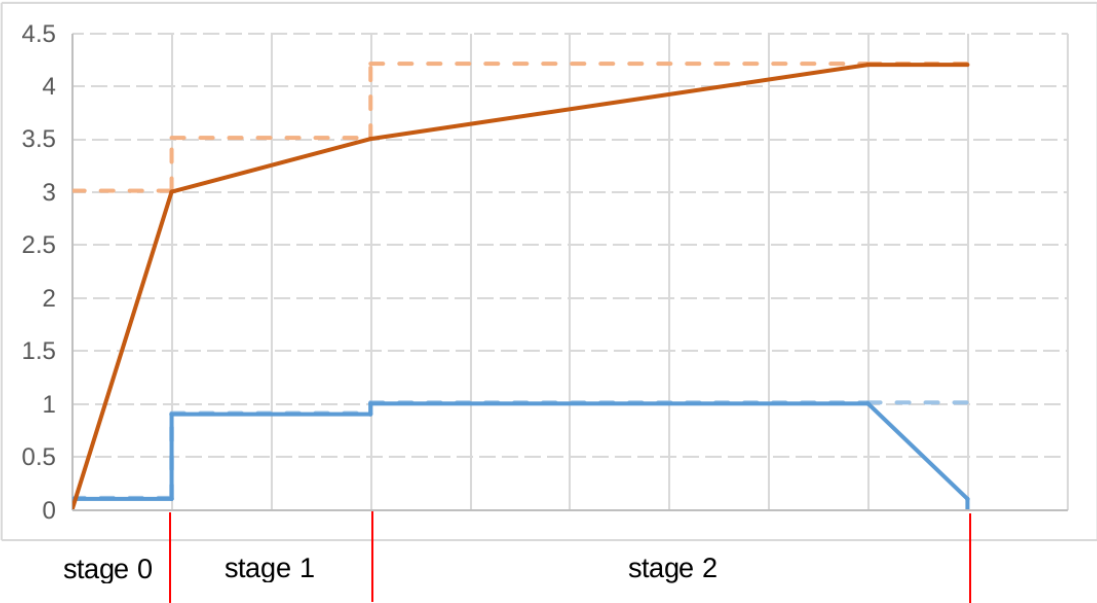
温度高于70°C关闭输出

反接保护

开启继电器时, 如果检测到反接则拒绝开启继电器; 运行过程中如果触发反接, 将会关闭继电器, 断开输出。

未实现, 以下为设计 充电模式

参考TP4056



以电池电压为判断条件, 将充电过程分为几个阶段

阶段		范围	限制电压	限制电流
stage 0	CC	0.0-3.0v	3.0v	0.1a
stage 1	CC	3.0-3.5v	3.5v	0.9a
stage 2	CC & CV	3.5-4.2v	4.2v	1.0a

每个阶段满足连续5s,电压达到限制值的95%进入下一个阶段。最后一个阶段还需满足电流 < 0.1a

那么应该发送命令 MODE=1, END=100, STAGE=3, LIMIT=3000,100,3500,900,4200,1000

对应串口命令行 M1E001S2L0003,001,0053,009,0024,0001