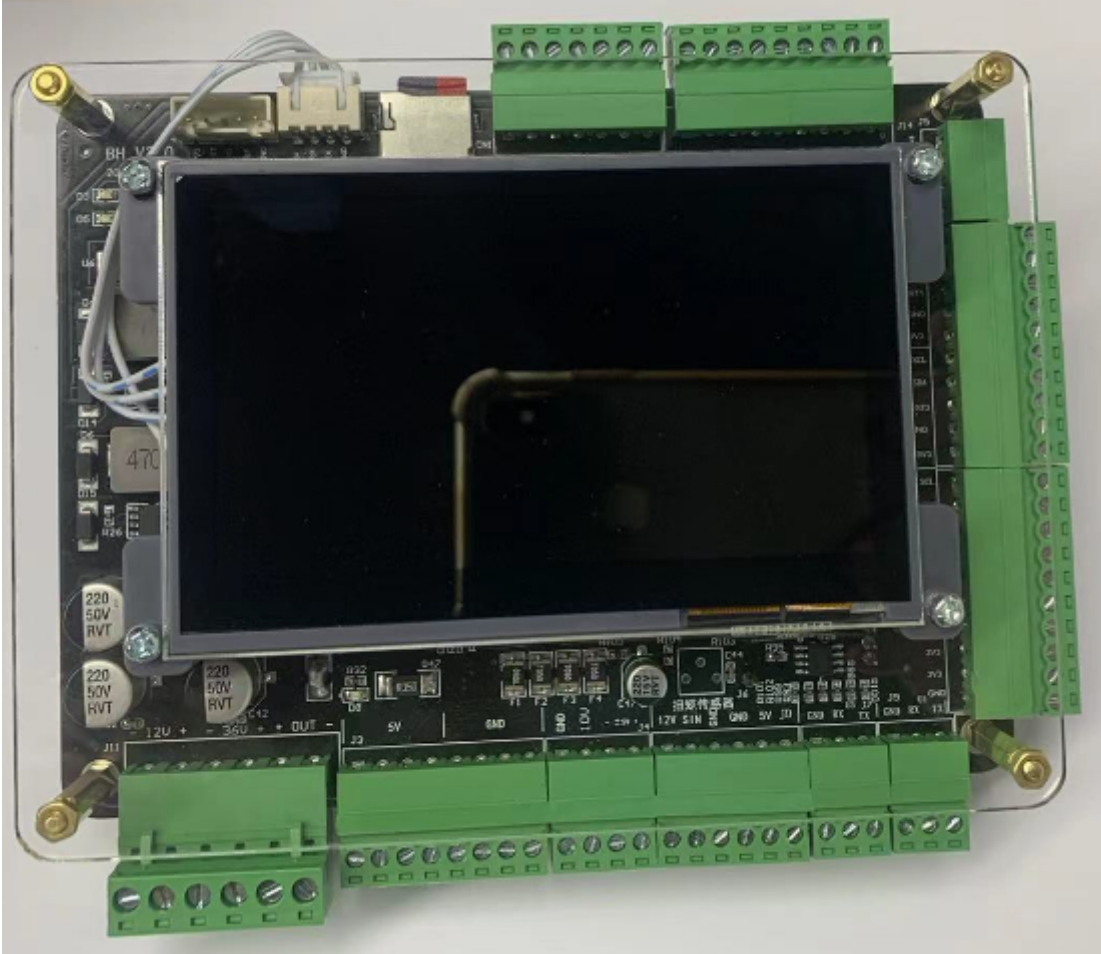


V2.0 使用说明书

时间	内容
2023/03/15	第一版

一、控制器简介

1、外观

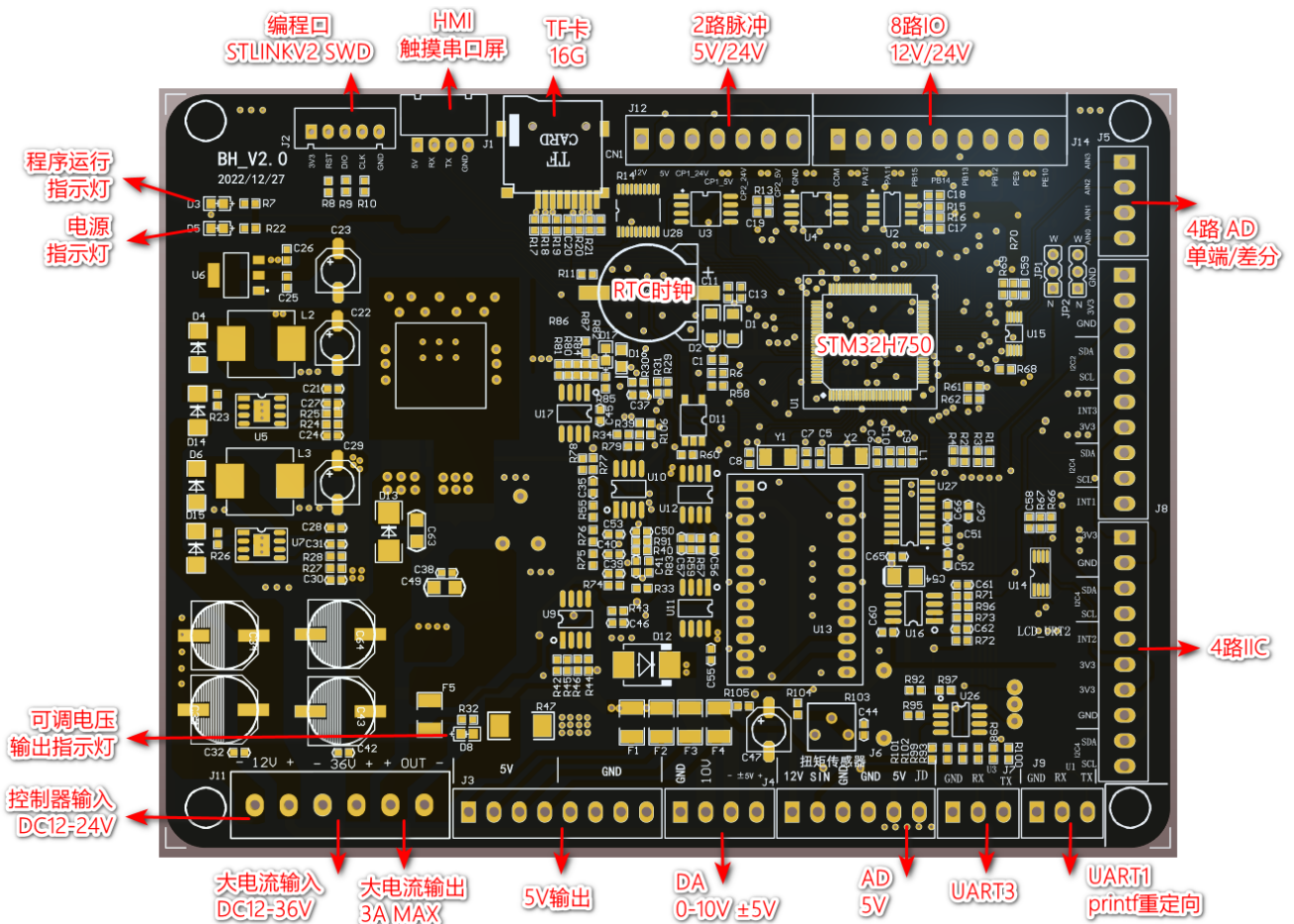


2、功能

- 适用于BH系列磁流变阻尼器的控制器
- 该控制器可实时调整电流输出，控制BH系列磁流变阻尼器的阻尼力大小
- 该控制器附带多种外设口，是一块多功能嵌入式开发板
- 使用STM32H750作为主控芯片，高达480MHz主频，1MB RAM 128KB片内Flash 16/32MB片外FLASH
- 可选配触摸屏，做丰富的显示输出展示

序号	功能名称	描述
----	------	----

序号	功能名称	描述
1	可控大电流输出	1通道, 0-3A, 36V MAX
2	输出闭环采样电路	可软件实现恒流 (CC) 控制
3	5V输出	4通道, 500mA
4	UART串口	3通道
5	I2C	3通道
6	AD转换	模式1: 5通道单端 (0-5V) 模式2: 1通道单端 (0-5V), 2通道差分 (-5 - +5V)
7	DA转换	通道1: 0-10V 通道2: -5 - +5V
8	IO输入	8通道
9	脉冲输入	2通道
10	RTC	实时时钟
11	触摸屏	IPS 4.3寸高清串口屏交互 (选配)

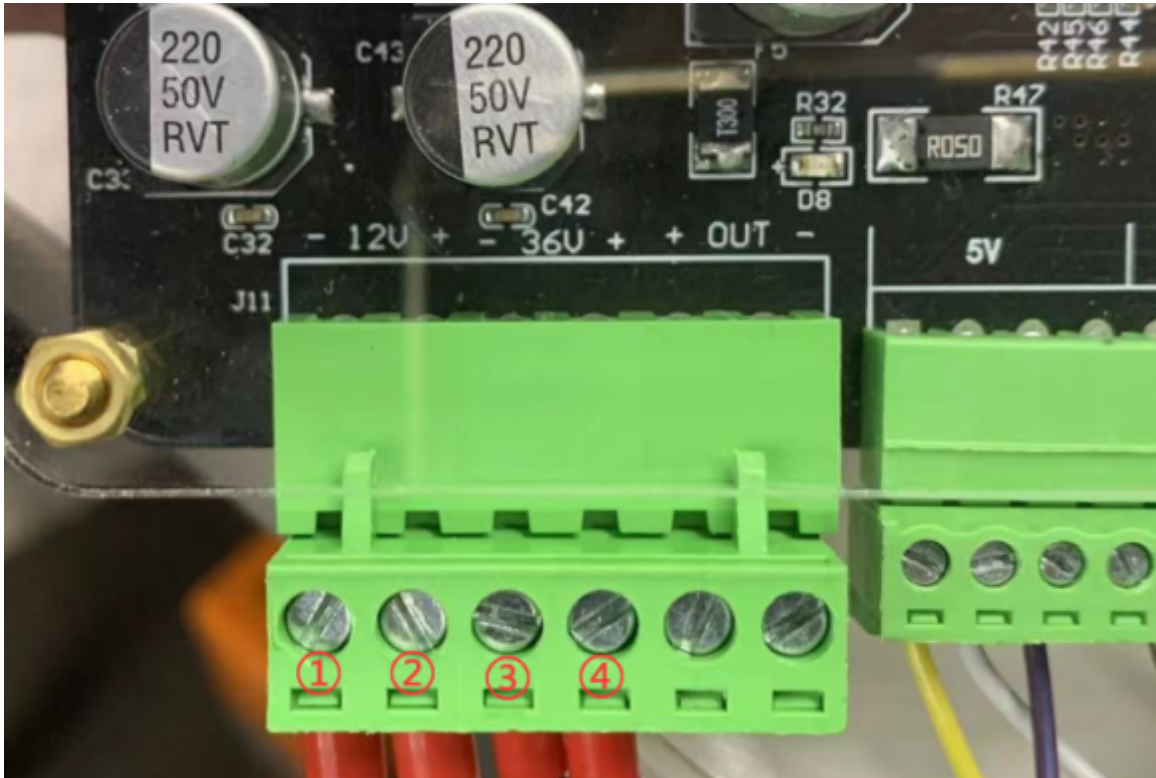


3、清单

名称	型号	数量
----	----	----

名称	型号	数量
控制器	BHV2.0	1
HMI触屏 (选配)	DC80480AM043_1011_0C	1
编程线	STLINKV2 SWD	1
串口线	CH340 TTL转USB	1

4、供电



控制器需要接入2路DC供电

- ①②脚为控制电路供电，输入DC12-24V
- ③④脚为大电流输出可调电路供电，输入DC12-36V

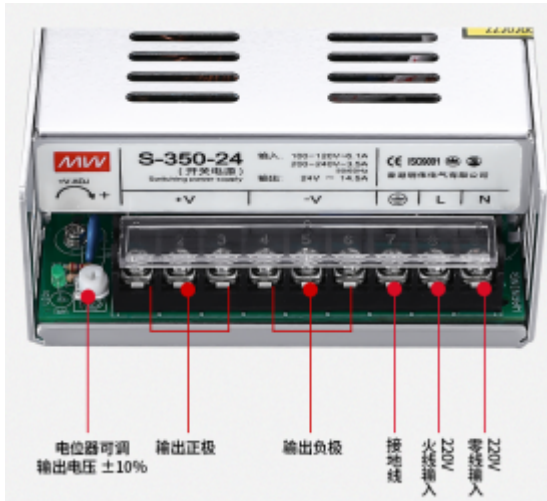
只有当1-4脚正常供电情况下，控制器才能正常工作。

③④脚输入的电压决定了所带负载的最大功率。

如：

- 带1只阻尼器，内阻 5Ω ，电流可调0-3A，负载最大电压 $U=IR=3*5=15V$ ，此时，③④脚要输入如24V，50W才能满足功率要求
- 带2只同型号阻尼器，并联，单只内阻 5Ω ，电流可调0-1.5A（控制器最大输出3A，并联分流），负载最大电压 $U=IR=1.5*5=7.5V$ ，此时，③④脚要输入如12V，25W才能满足功率要求

请根据实际需求,预留一些功率余量，另外购买功率合适的开关电源，供给直流电以满足使用需求。



二、开发环境安装

考资料包中文档【MDK安装及使用.pdf】第1-11页。

所提供软件仅作学习研究，请在下载后24小时内删除，如有需要请购买正版。

1、KeilVision5安装

开发IDE

文件：01 MDK5.25.7z

2、STM32H7安装

文件：02 STM32H7安装包.7z

3、STLINK驱动安装

用于烧录程序

文件：03 ST LINK驱动.zip












4、串口CH340驱动

用于上位机串口通信

文件：04 CH340驱动(USB串口驱动)_XP_WIN7共用.7z

三、功能模块使用

1、目录结构

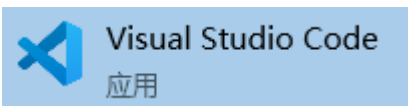
 BSP	2022-12-13 16:26	文件夹	
 CMSIS	2022-12-13 15:05	文件夹	
 CNTLR	2022-12-15 10:25	文件夹	
 HAL	2022-12-08 10:32	文件夹	
 HMI	2022-12-08 10:43	文件夹	
 MDK	2022-12-15 13:46	文件夹	
 STM32H7xx_HAL_Driver	2022-12-08 10:32	文件夹	
 System	2022-12-08 10:32	文件夹	
 User	2022-12-15 10:17	文件夹	
 CMSIS.zip	2020-10-27 0:09	好压 ZIP 压缩文件	287 KB
 keilkill.bat	2017-10-30 22:03	Windows 批处理...	1 KB

- BSP: 所有外设及功能模块的代码
- HAL: FATFS库
- **CNTLR** : 控制器相关功能模块实现
- **MDK**: 工程项目文件，Project.uvprojx
- STM32H7xx_HAL_Driver : STM32提供的库
- User : main.c文件

安装好



双击“MDK\Project.uvprojx”即可打开IDE进行程序编写。由于uv5对编写程序不太友好，建议使用其他三方IDE如

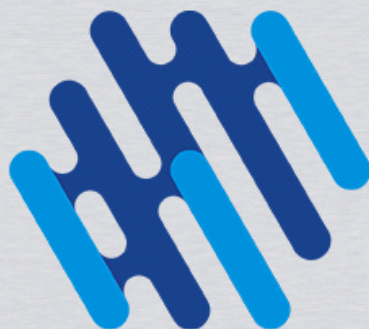


进行代码编写后再在uv5中进行编译下载操作。

2、HMI触屏

控制器可选配一块触摸屏，使用232串口和控制器连接，已编写好相关控制交互例程，并提屏幕源码。

深圳博海新材料技术有限公司



博海磁流变



博海磁流变-综合控制器 V2.0

2023-03-15 15:50:57



大电流输出



AD/DA



I2C传感器



IO/脉冲

2.1 简介



高清IPS

项目	说明
型号	DC80480AM043_1011_0C (RS232/TTL,电容触摸)
核心处理器	400M SOC处理器
操作系统	无操作系统，上电即可运行
协议类型	默认大彩组态指令集，上位机可配置运行MODBUS RTU、XGUS协议
脚本语言	LUA脚本，屏内部可运行用户编写的逻辑、协议和算法功能
尺寸	4.3寸
分辨率	800×480
安装方向	支持0、90、180和270度旋转安装显示
存储空间	128Mbit
字库	内置矢量字体，边缘抗锯齿处理，支持任意大小ASCII、GBK、GB2312、UNICODE（全球语言）字库，也可自定义任意电脑字体显示
图片存储	支持JPEG、PNG压缩，支持任意大小图片存储。
颜色	65K色，16位RGB
电压	4.5-30V

项目	说明
功耗	休眠：0.6W 背光最暗：0.9W 背光最亮：1.2W
通讯方式	RS232/TTL(出厂默认232电平，短接J5为TTL电平)
通讯波特率	1200 ~ 921600bps,典型波特率：115200bps
通讯接插件规格	XH2.54-8P
图片本地下载	SD 卡、串口、U 盘 (需要定制)
固件本地/远程升级	插入SD卡本地升级/支持用户主板远程串口升级屏幕固件
图片远程升级	支持用户远程利用自己主板串口升级屏幕相关图片工程、字库、配置文件等
实时时钟 (RTC)	支持时钟、定时器、倒计时等功能
屏有效显示区 (AA)	长宽 = 96.0mm54.9mm
产品尺寸 (长宽高)	121.774.6mm15.8mm (MAX , 含TP)
配套上位机软件	VisualTFT®
工作温度	-20 ~ +70°C
存储温度	-30 ~ +80°C
AV输入	不支持
音频播放	不支持
音频接插件规格	不支持
视频播放	MP4视频格式，与图片共用存储空间。
WIFI	不支持

2.2 参考资料

帮助文档：

<http://doc.gz-dc.com/>

开发IDE Visual TFT：

<https://www.gz-dc.com/category/typeid/8#mainTop>

串口协议：

<https://www.gz-dc.com/category/typeid/410>

3、功能模块

3.1 调试信息

UART1重定向了printf函数，可以直接输出到上位机。



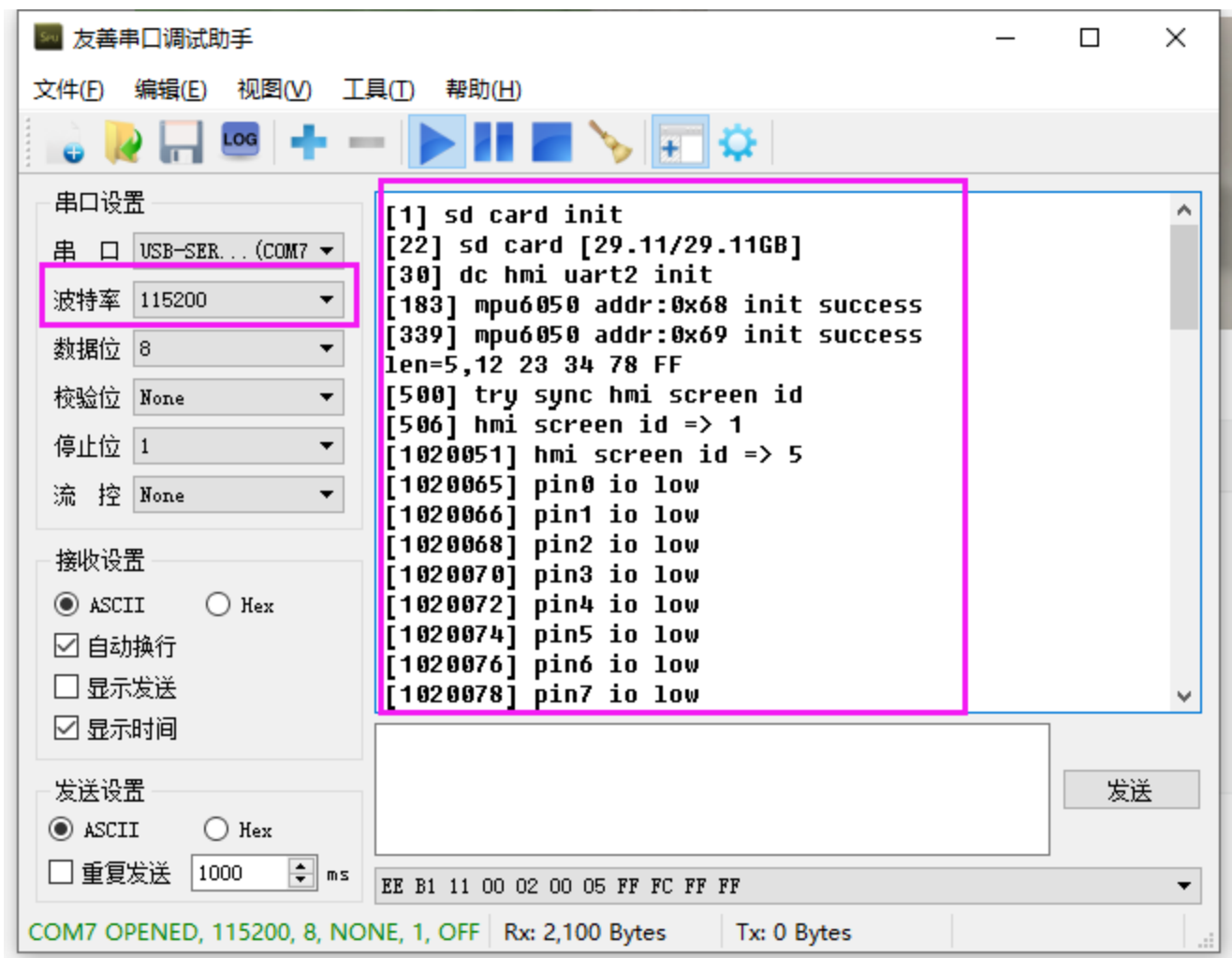
推荐使用该函数打印：

```
//CNTLR\Utils\bh_debug.h

/// @brief 调试输出日志（printf），注意只定义了256字节缓冲区，防止溢出
/// @param format
/// @param
void bh_debug_log(const char *format, ...);

//使用
bh_debug_log("set da pn5v = %.3f", vo1PN5v);
```

上位机串口工具查看：



3.2 0-3A大电流输出

请确保按照要求给控制器供电。

此功能由DAC输出一个模拟电压控制电源输出电路输出相应的大功率电压，使用此功能需要初始化DAC外设：

```
// BSP\DAC\dac.h
// DAC初始化，用于大电流输出控制
Stm32DAC_Init();
```

根据负载电阻，控制电流，计算DAC值，并使能输出电压：

```
// CNTLR\output_mgr.c
/// @brief 大电流输出使能
/// @param current 电流
/// @param resistance 电阻
void output_mgr_enable(float current, float resistance)
{
    float dacValue = current * resistance * 100;
    DAC_SetVoltage((UINT)dacValue);
}
```

由于磁流变阻尼器负载是绕组线圈，负载电阻具有温漂特性，因此，在复杂的使用工况下，建议使用软件恒流算法控制，以保证磁流变阻尼器恒定电流供应。

控制原理：读取电流、电压采样电路的AD值，实时计算负载电阻并目标值对比，超过设定的调整阈值，反馈修改输出电压，参考代码如下：

```
// CNTLR\mrf_cntlr_mgr.c
/// @brief 输出电流实时采样
void output_realtime_sample()
{
    // 采样
    static OUTPUT_SAMPLE_DATA sampleData = {0};
    output_sample(&sampleData);

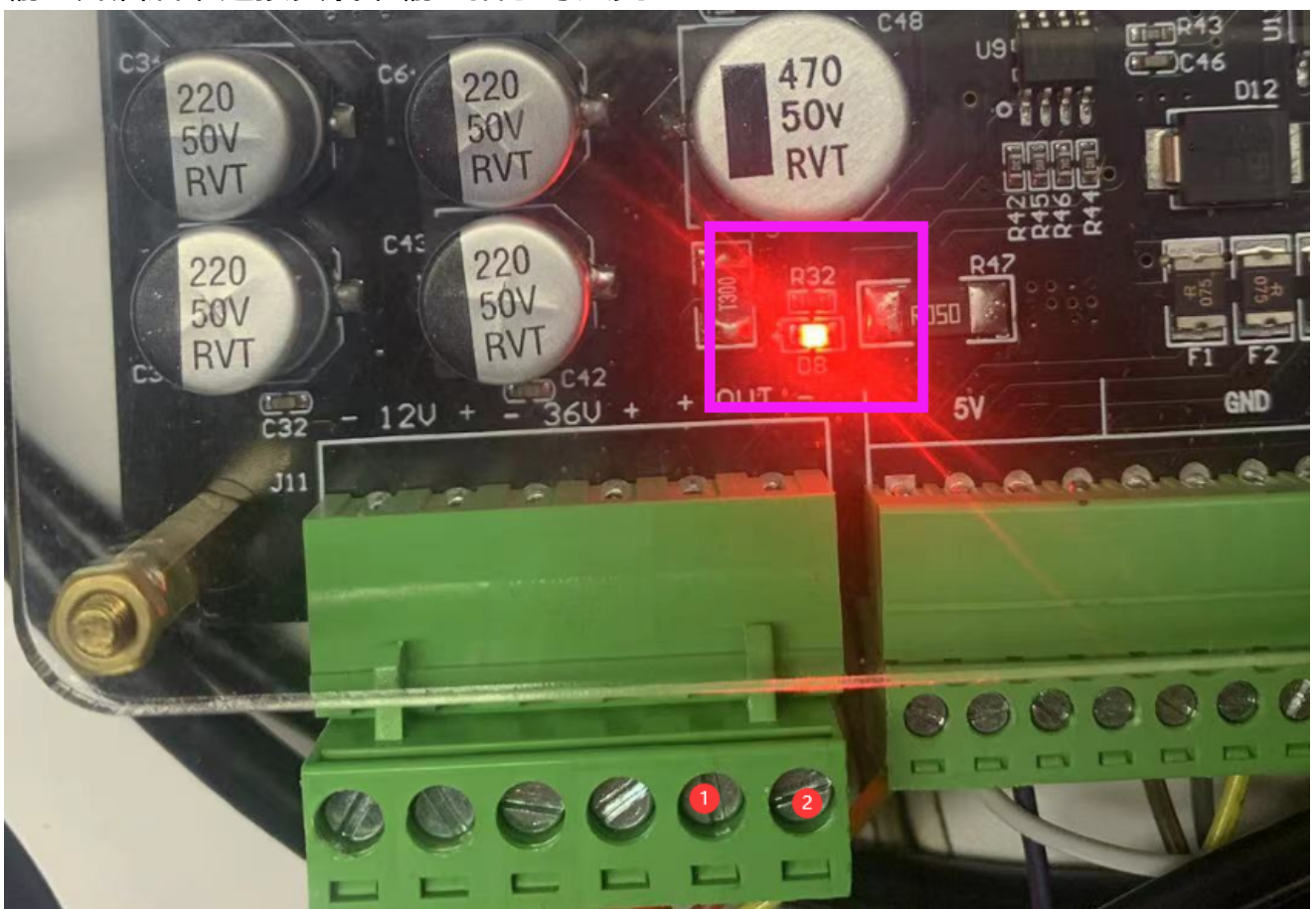
    g_hmi_data.output_data.real_current = sampleData.current;
    g_hmi_data.output_data.real_voltage = sampleData.voltage;
    if (sampleData.current > 0)
    {
        //实时负载值
        g_hmi_data.output_data.real_resistance = sampleData.voltage /
sampleData.current;
    }
}

//输出
output_mgr_enable(current, resistance);
```

若选配屏幕，界面如下：



输出开启后，**连接负载**，输出指示灯点亮：



3.3 AD输入/DA输出

3.3.1 DA输出

DA初始化：

```
// BSP\PWM\pwm.h
// PWM初始化 控制DA模拟量输出 10KHz
PWM_Init(10000, 1);
```

输出调用：

```
// BSP\POWER\power.h
/// @brief 输出模拟量电压值
/// @param chType
/// @param voltage
void da_voltage_output(DA_CHANNEL_TYPE chType, float voltage);

//正负5V输出
da_voltage_output(DA_CHANNEL_PN5V, -2.5f);
//0-10V输出
da_voltage_output(DA_CHANNEL_10V, 8.3);
```

3.3.2 AD输入

4通道AD初始化：

```
// BSP\ADS1115\iic_ads1115.h
IIC_ADS_Init();
```

输入调用：

```
// BSP\ADS1115\iic_ads1115.h
/// @brief 读取4通道模拟量
/// @param pData
/// @param chType 读取类型 AD_NORMAL AD_DIFF
void adc1115_read(PADS115_DATA pData, ADS1115_CH_TYPE chType);

//示例
ADS115_DATA data = {0};
adc1115_read(&data, AD_NORMAL);
float ch0 = data.ch0; //通道0
float ch1 = data.ch1;
//差分模式时，通道2/3无效
float ch2 = data.ch2;
float ch3 = data.ch3;
```

1通道AD (角度传感器) 初始

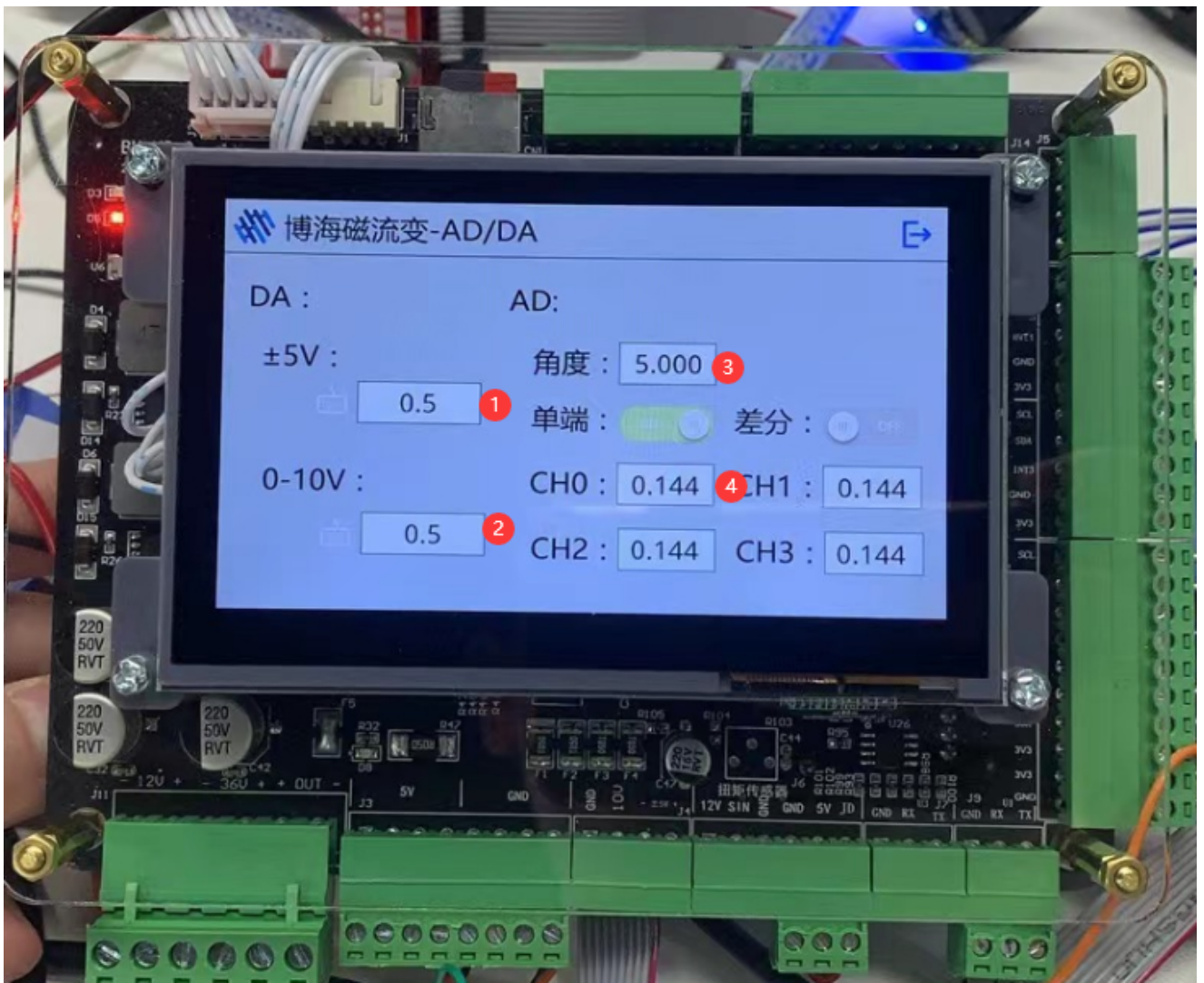
```
// BSP\ADC\adc.h
// ADC初始化, 角度传感器和输出电流电压采样
STM32_ADC_Init(ADC1, NULL);
```

调用：

```
// BSP\ADC\adc.h
/// @brief adc采样
/// @param pData
void adc_sample(PADC_SAMPLE_DATA pData);

ADC_SAMPLE_DATA sampleData;
adc_sample(&sampleData);
//角度传感器值
float angle = sampleData.adc_angle;
```

若选配屏幕，界面如下：

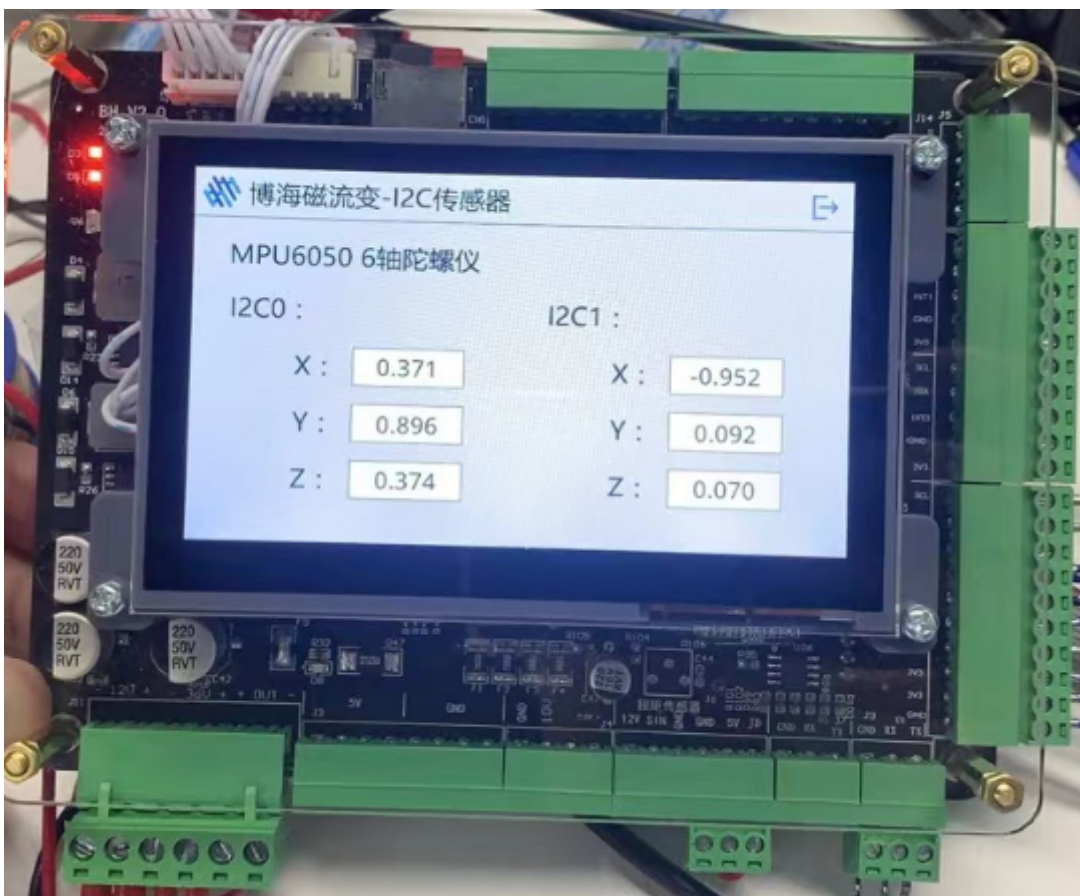


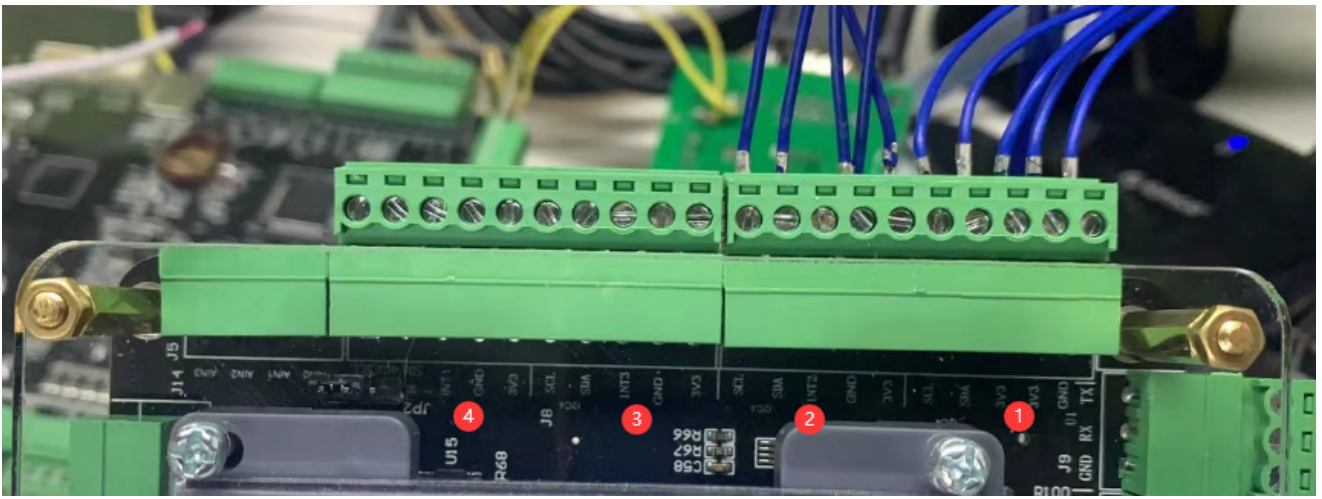
位置展示：


```
void mpu6050_read_accelerometer(uint8_t addr, MPU6050_DATA pData);

//示例
void mrf_cntlr_read_mpu6050(BYTE addr, PGYROSCOPE_ACC_DATA pData)
{
    static MPU6050_DATA data = {0};
    mpu6050_read_accelerometer(addr, &data);
    pData->x = mpu6050_to_g(data.x);
    pData->y = mpu6050_to_g(data.y);
    pData->z = mpu6050_to_g(data.z);
}
```

若选配屏幕，界面如下：





3.5 IO/脉冲输入

IO输入支持12V/24V

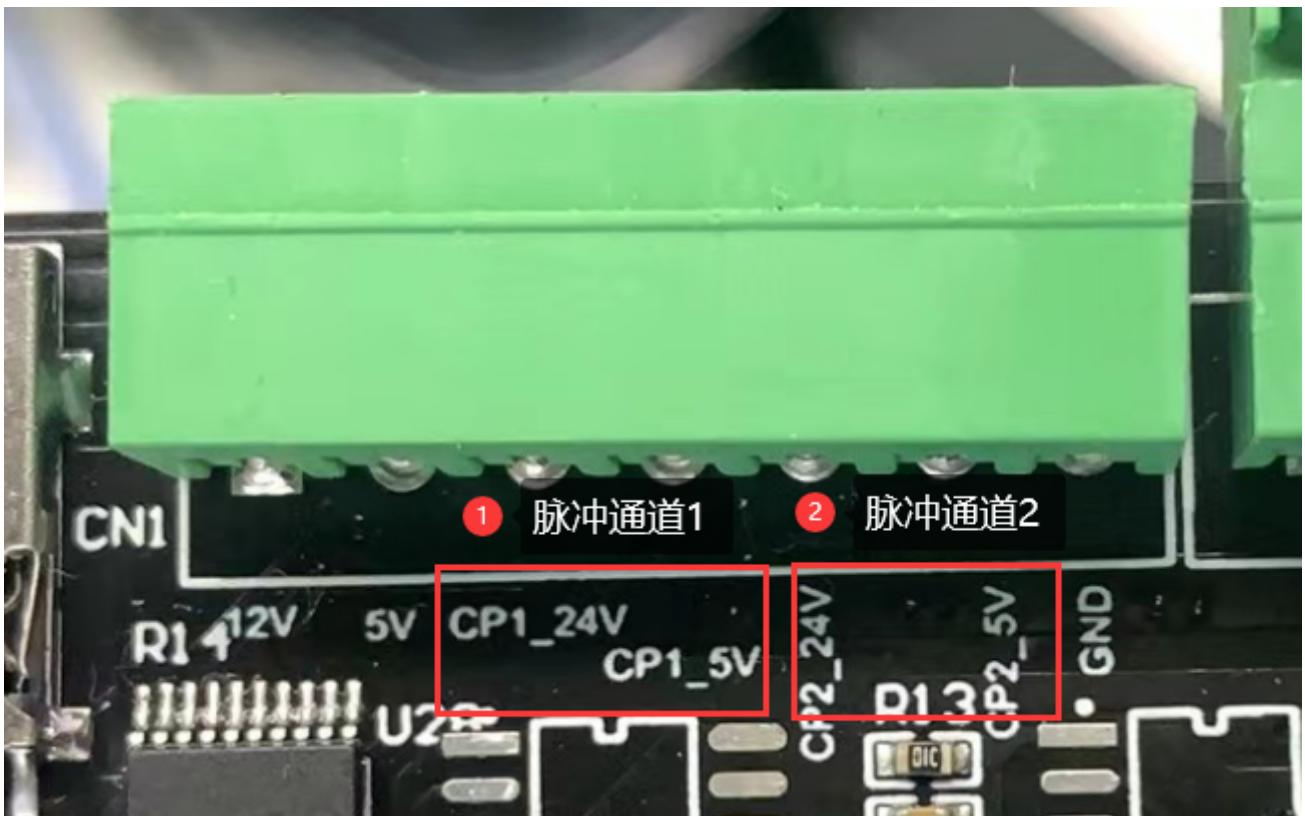
初始化IO

```
// BSP\IO_Input\IO_Input.h  
IO_Input_Init();
```

读取IO状态：

```
///  
/// @brief 读取IO状态，存入全局变量g_io_data  
/// @param  
void IO_Input_Read(void);
```

脉冲可同时使用2路,每一路都支持5V/24V输入



初始化脉冲

```
/// @brief 5v 24v 2通道脉冲初始化
void pulse_init()
{
    Tim5_CapInit(FREQ_COUNTER, pulse_isr_ch1); // PA0
    Tim1_CapInit(FREQ_COUNTER, pulse_isr_ch2); // PA1
}
```

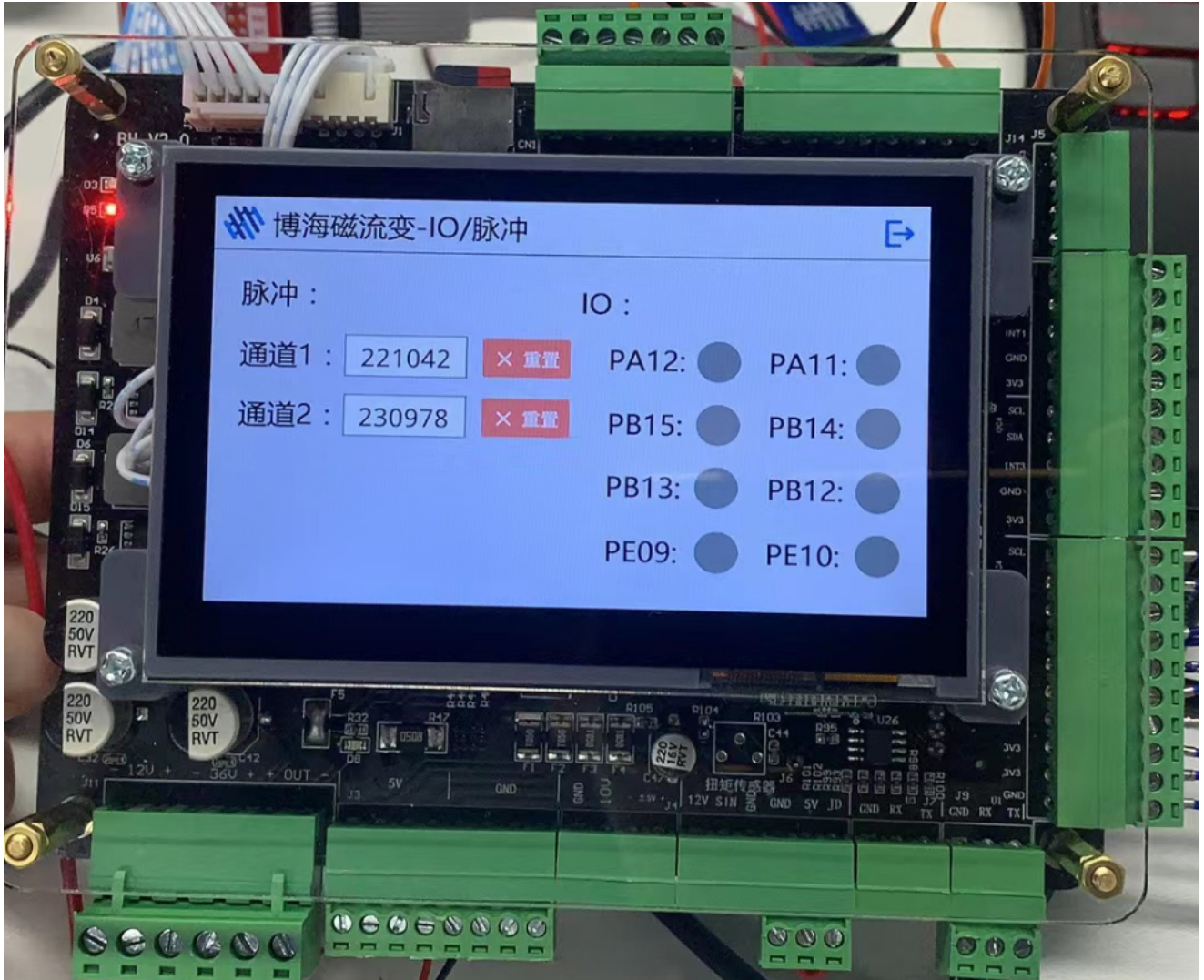
脉冲计数：

```
/// @brief 脉冲通道1中断回调
/// @param
void pulse_isr_ch1(void)
{
    if (TimCapHandle.Channel == HAL_TIM_ACTIVE_CHANNEL_1)
    {
        g_hmi_data.pulse.ch1_count++;
    }
}

/// @brief 脉冲通道2中断回调
/// @param
void pulse_isr_ch2(void)
{
```

```
if (TimCapHandle_TIM1.Channel == HAL_TIM_ACTIVE_CHANNEL_4)
{
    g_hmi_data.pulse.ch2_count++;
}
}
```

若选配屏幕，界面如下：



3.6 SD卡读写

初始化SD卡：

```
// BSP\LOG\log.h
sd_card_init();
```

写日志，默认存储到SD卡 /Logs/下：

```
/// @brief 写日志到文件
/// @param fmt
```



```
/// @param  
void cntlr_log_wirte(const char *fmt, ...);
```

3.7 EEPROM读写

初始化：

```
// BSP\24CXX\24cxx.h  
EEPROM_Init();
```

读写：

```
// BSP\24CXX\24cxx.h  
uint32_t EEPROM_Read(uint8_t *pBuffer, uint16_t ReadAddr, uint16_t  
NumToRead);  
uint32_t EEPROM_Write(uint8_t *pBuffer, uint16_t WriteAddr, uint16_t  
NumToWrite);
```

3.8 RTC

RTC操作

```
// RTC初始化  
RTC_Init(NULL);  
//设置时间  
void RTC_SetDateTime(_Calendar_obj datetime);  
//读取时间  
_Calendar_obj RTC_GetDateTime(void);
```