

标准版 **AXP173【电源管理芯片】最全资料版**

[电源/能源](#) [方案验证板](#) [电源模块](#)

简介：让你的产品获得手机般电源管理，免受电源管理的困扰，所有库函数均注释写法和用法！！！基本榨干所有可用功能！！！

开源协议: GPL 3.0

发布时间： 2022-10-24 10:06:34

1.4k 14 41 52

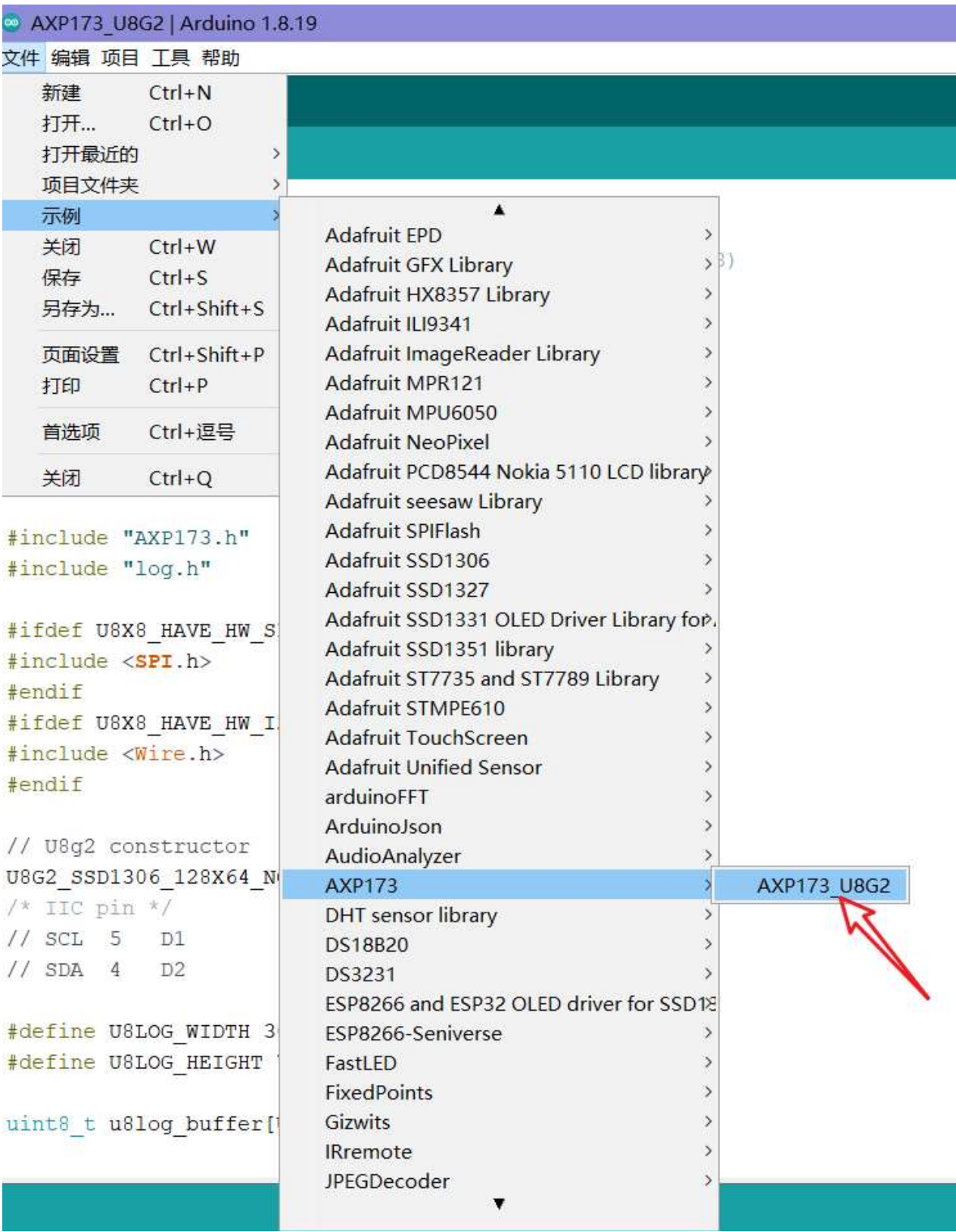
描述

-----10.22更新-----

公告：源码尽量去Github下载，附件里的程序更新时间为10.9日，GitHub可以实时更新源码。不会clone也可以直接下载.zip
代码优化：添加了日志打印库，方便硬件调试

-----10.9更新-----

- 1.由于该芯片可以定制，输出种类参差不齐，务必购买链接中的通用版本，否则IIC会无法读写！！
2. 使用在自己项目之前请先用我的或者你自己画的PCB学习一下芯片用法与功能！！
3. 附件压缩包解压后的文件夹，直接放入arduino库即可（这是我的，一般都在文档文件夹里C:\Users\69180\Documents\Arduino\libraries），需下载U8G2图形库支持屏幕的显示！！！之后直接可以在arduino示例中找到！！



```
void KeyPressIRQEvent() {
    if (pmu.powerState()) {
        if (pmu.getShortPressIRQState()) { //获取对应位IRQ状态信息 true o
            lprintf(LOG_INFO, "PEK Short Press\n");
            pmu.setShortPressIRQDisabale(); //对应位写1结束中断
            static bool LOCK_STATE = 1; //初始化屏幕状态
            if (LOCK_STATE) { // = 1 睡眠
                // pmu.prepareToSleep();
                u8g2.setPowerSave(1);
                LOCK_STATE = 0;
                lprintf(LOG_INFO, "Screen Lock\n");
            }
            else { // = 0 唤醒
                // pmu.RestoreFromLightSleep();
                u8g2.setPowerSave(0);
                LOCK_STATE = 1;
                lprintf(LOG_INFO, "Screen UnLock\n");
            }
            /* 息屏时设置lightsleep或者deepsleep模式 */
        }
        else if (pmu.getLongPressIRQState()) { //获取对应位IRQ状态信息 true o
            pmu.setOutputEnable(OP_LDO2, true); //LDO2设置为输出
            pmu.setOutputVoltage(OP_LDO2, 3000); //LDO2电压设置为3.000V
            /* Enable and set LDO3 voltage */
            pmu.setOutputEnable(OP_LDO3, true); //LDO3设置为输出
            pmu.setOutputVoltage(OP_LDO3, 3300); //LDO3电压设置为3.300V

            /* Enable and set LDO4 voltage */
            pmu.setOutputEnable(OP_LDO4, true); //LDO4设置为输出
            pmu.setOutputVoltage(OP_LDO4, 3300); //LDO4电压设置为3.300V

            /* Enable and set DCDC1 voltage */
            pmu.setOutputEnable(OP_DCDC1, true); //DCDC1设置为输出
            pmu.setOutputVoltage(OP_DCDC1, 3300); //DCDC1电压设置为3.300V

            /* Enable and set DCDC2 voltage */
            pmu.setOutputEnable(OP_DCDC2, true); //DCDC2设置为输出
            pmu.setOutputVoltage(OP_DCDC2, 2275); //DCDC2电压设置为2.275V

            /* Enable Battery Charging */
            pmu.setChargeEnable(true); //充电功能使能
        }
    }
}
```

5. 代码优化:
- 把枚举体从类中提取出来，免去主函数中需要引用类写法 (例子)
改前: AXP173::OP_LDO2
改后: OP_LDO2
 - 把对象声明在库里完成，不需要在主函数里声明
 - 添加keywords.txt文件

6. 以后代码会更新在GitHub和群里，这里只更新教程和内容!!!

三根数据线&1.6元&超小PCB布局&超低功耗。可以让你的设备获得手机一般的电源管理。

本库寄存器读写方式主要借鉴了m5stack官网的AXP192库。IIC_PORT库来自 Forairaaaaa 大佬。该芯片的通用版本感谢 liuzewen 大佬指路，否则由于该芯片的定制特性导致默认输出不一样IIC无法通讯。

自己注释了所有函数!!! 以及编写了其它自己需要的函数。比如通过识别IRQ中断引脚与中断寄存器设置长按关机与短按息屏等等，还可以自己设置息屏后关闭哪些输出等等。

你可以用它做：蓝牙键盘的充电管理，开发板的充电管理，微型手表的充电管理，小电视的充电管理.....等等等等等等

某宝链接: [AXP173 \(LDO4默认输出版本\)](#)

使用方法：见文中接线图与b站视频演示 (请先参阅程序，接线，pcb以及数据手册，人多了会抽时间写更详细的文档)

其它：文档后续会持续完善，最近比较忙。不懂得可以先来QQ群问：735791683

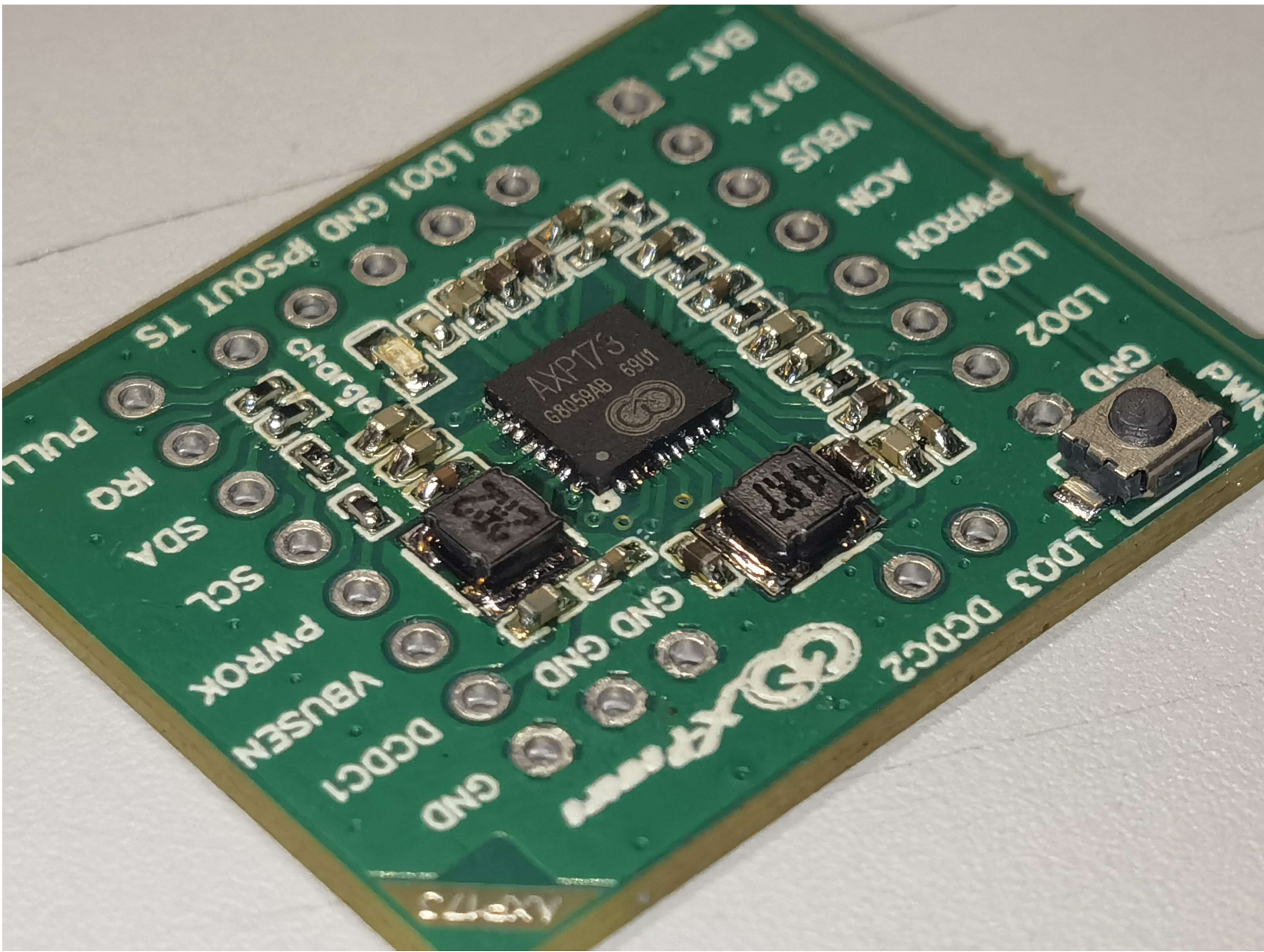
b站视频链接👁️: **【[自制开源]AXP173 PMIC电源管理芯片最全资料-哔哩哔哩】** <https://b23.tv/QMnHXd9>

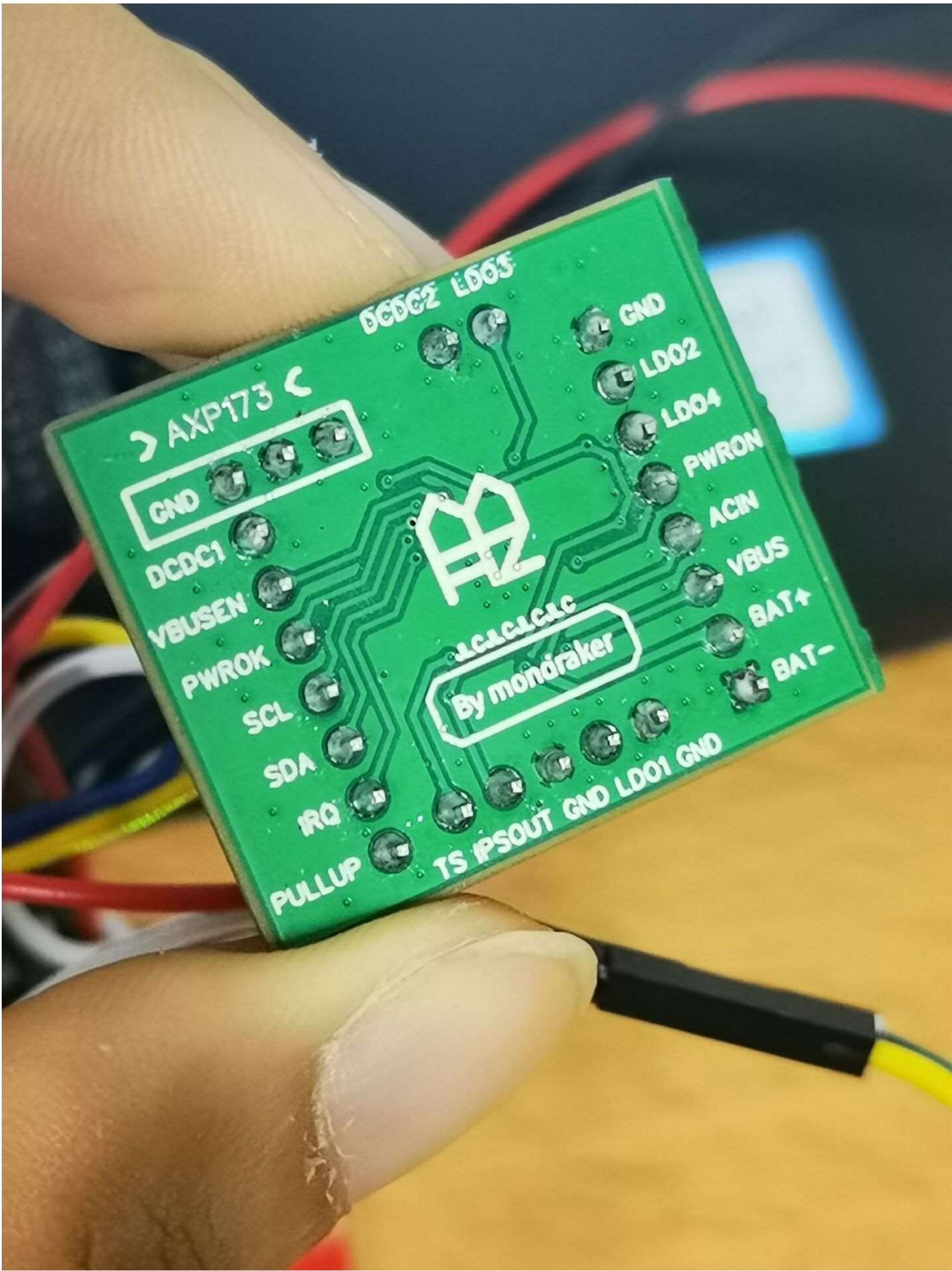
arduino, esp8266, esp32等等直接无需移植, 直接拿来即用

芯片基本功能（基本榨干常用功能）



先看板子





再看接线


```
void setPmuPower() { //电源通道电压输出设置，交换位置可以设置上电时序，中间加delay
/* Enable and set LD02 voltage */
pmu.setOutputEnable(AXP173::OP_LD02, true); //LD02 设置为输出
pmu.setOutputVoltage(AXP173::OP_LD02, 3000); //LD02 电压设置为3.000V

/* Enable and set LD03 voltage */
pmu.setOutputEnable(AXP173::OP_LD03, true); //LD03 设置为输出
pmu.setOutputVoltage(AXP173::OP_LD03, 3300); //LD03 电压设置为3.300V

/* Enable and set LD04 voltage */
pmu.setOutputEnable(AXP173::OP_LD04, true); //LD04 设置为输出
pmu.setOutputVoltage(AXP173::OP_LD04, 3300); //LD04 电压设置为3.300V

/* Enable and set DCDC1 voltage */
pmu.setOutputEnable(AXP173::OP_DCDC1, true); //DCDC1 设置为输出
pmu.setOutputVoltage(AXP173::OP_DCDC1, 3300); //DCDC1 电压设置为3.300V

/* Enable and set DCDC2 voltage */
pmu.setOutputEnable(AXP173::OP_DCDC2, true); //DCDC2 设置为输出
pmu.setOutputVoltage(AXP173::OP_DCDC2, 2275); //DCDC2 电压设置为2.275V
}
```

```
void setPmuConfig() { //电源芯片ADC，库仑计等功能设置
/* Set off time */
pmu.setPowerOffTime(AXP173::POWEROFF_4S); //设置关机后所有电源通道关闭时长为4S

/* Set on time */
pmu.setPowerOnTime(AXP173::POWERON_128mS); //设置开机后电源通道启动输出时长为128mS

/* Enable VBUS ADC */
pmu.setADCEnable(AXP173::ADC_VBUS_V, true); //VBUS ADC 电压使能
pmu.setADCEnable(AXP173::ADC_VBUS_C, true); //VBUS ADC 电流使能

/* Enable Battery ADC */
pmu.setADCEnable(AXP173::ADC_BAT_V, true); //Battery ADC 电压使能
pmu.setADCEnable(AXP173::ADC_BAT_C, true); //Battery ADC 电流使能

/* Enable Battery Charging */
pmu.setChargeEnable(true); //充电功能使能
pmu.setChargeCurrent(AXP173::CHG_360mA); //设置充电电流为450mA

/* Enable Coulometer and set COULOMETER_ENABLE*/
pmu.setCoulometer(AXP173::COULOMETER_ENABLE, true); //库仑计使能

u8g2log.printf("AXP173 Set OK\n");
}
```

获取信息


```
void printPmuInfo() { //需要打印在屏幕上的芯片信息

    /* Get PMU temp info */
    u8g2log.printf("CoreTemp :%.2f 'C\n", pmu.getAXP173Temp()); //芯片温度

    /* Get VBUS info */
    // u8g2log.printf("VBUS_voltage :%.2f V\n", pmu.getVBUSVoltage()); //VBUS输入电压
    // u8g2log.printf("VBUS_current :%.2f mA\n", pmu.getVBUSCurrent()); //VBUS输入电流

    if(pmu.isBatExist()){
        //u8g2log.printf("Battery :Battery Exist\n"); //电池接入状态
        /* Get Battery info */
        pmu.isCharging() ? u8g2log.printf("Charging :Is Charging s\n") : u8g2log.printf("Charging :NO or END Charging\n");//充电状态
        u8g2log.printf("Bat_voltage :%.2f V\n", pmu.getBatVoltage()); //电池电压
        u8g2log.printf("Bat_Current :%.2f mA\n", pmu.getBatCurrent()); //电池电流 正为充电，负为放电

        //u8g2log.printf("Bat_Level :%.2f %%\n", pmu.getBatLevel()); //电池电量百分比显示
        //u8g2log.printf("Bat_BatPower :%.2f W", pmu.getBatPower()); //电池瞬时功率

        u8g2log.printf("GetBatCoulombInput :%.2f C\n", pmu.GetBatCoulombInput()); //Get Coulomb charge Data
        u8g2log.printf("GetBatCoulombOutput :%.2f C\n", pmu.GetBatCoulombOutput()); //Get Coulomb Discharge Data
        u8g2log.printf("CoulometerData :%.2f C\n", pmu.getCoulometerData()); //get coulomb val affter calculation
    }
    else{
        u8g2log.printf("Battery :NO Battery\n"); //没电池就输出这
    }
}
```

按键事件（这里由于是测试代码，我直接用if实现，大型代码请自行使用线程中断）

```
void KeyPressIRQEvent(){
    if (pmu.powerState()) {
        if (pmu.getShortPressIRQState()) { //获取对应位IRQ状态信息 true or false
            u8g2log.printf("PEK Short Press\n");
            pmu.setShortPressIRQDisabale(); //对应位写1结束中断
            static bool LOCK_STATE = ~ pmu.getShortPressIRQState(); //初始化屏幕状态
            if (LOCK_STATE) { // = 1 睡眠
                // pmu.prepareToSleep();
                u8g2.setPowerSave(1);
                LOCK_STATE = 0;
                u8g2log.printf("Screen Lock\n");
            }
            else { // = 0 唤醒
                // pmu.RestoreFromLightSleep();
                u8g2.setPowerSave(0);
                LOCK_STATE = 1;
                u8g2log.printf("Screen UnLock\n");
            }
            /* 息屏时设置lightsleep或者deepsleep模式 */
        }
        else if (pmu.getLongPressIRQState()) { //获取对应位IRQ状态信息 true or false
            // u8g2log.printf("PEK Long Press\n");
            pmu.setLongPressIRQDisabale(); //对应位写1结束中断
            // u8g2log.printf("Shut Down\n");
            pmu.powerOFF(); //长按 2 秒关机
            /* 关机设置deepsleep模式待机或者直接关机 */
        }
    }
    else {
        // u8g2log.printf("AXP173 ERROR!\n");
        delay(1000);
    }
}
```

打印电池相关信息

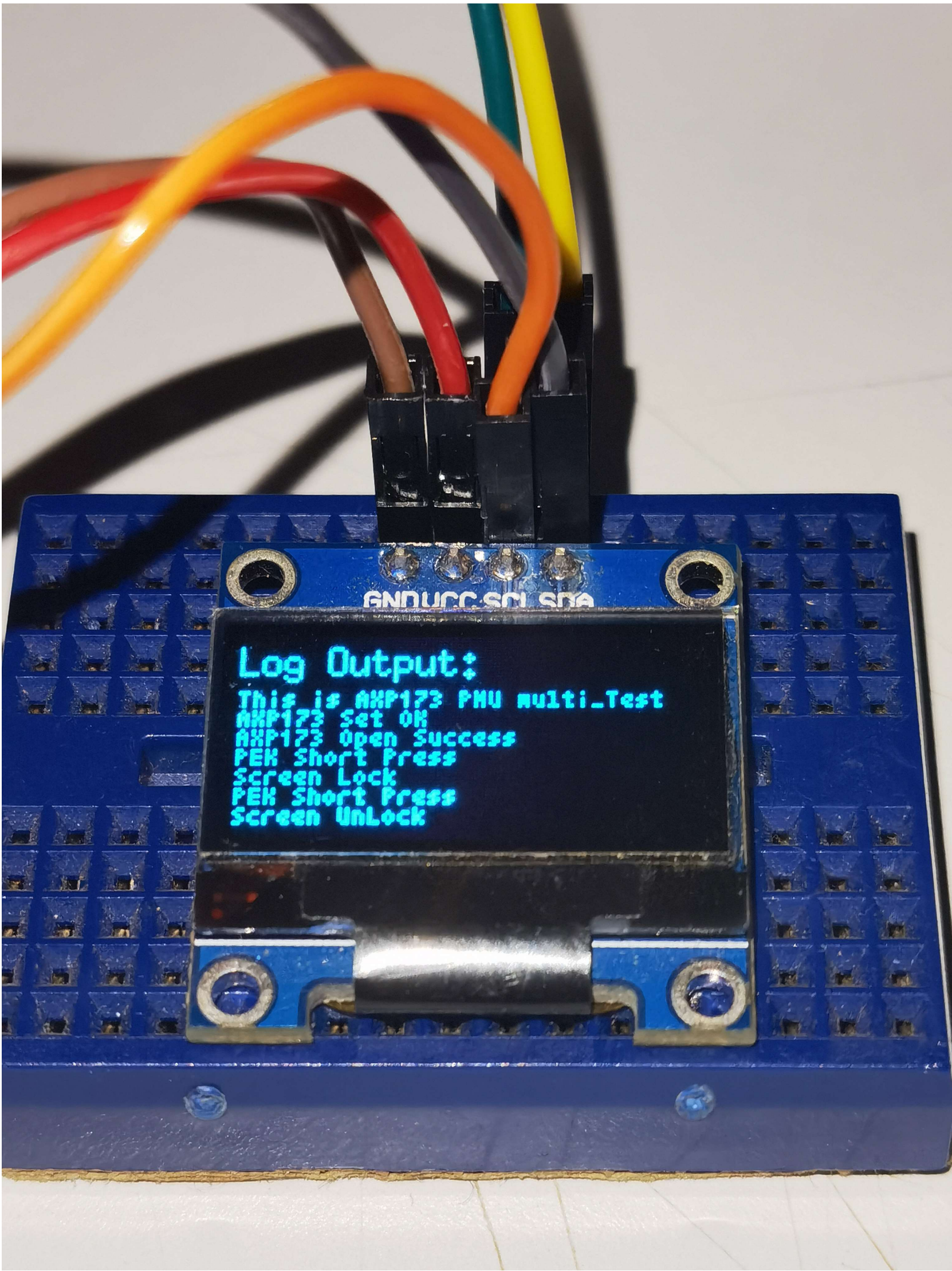
充电中



未充电

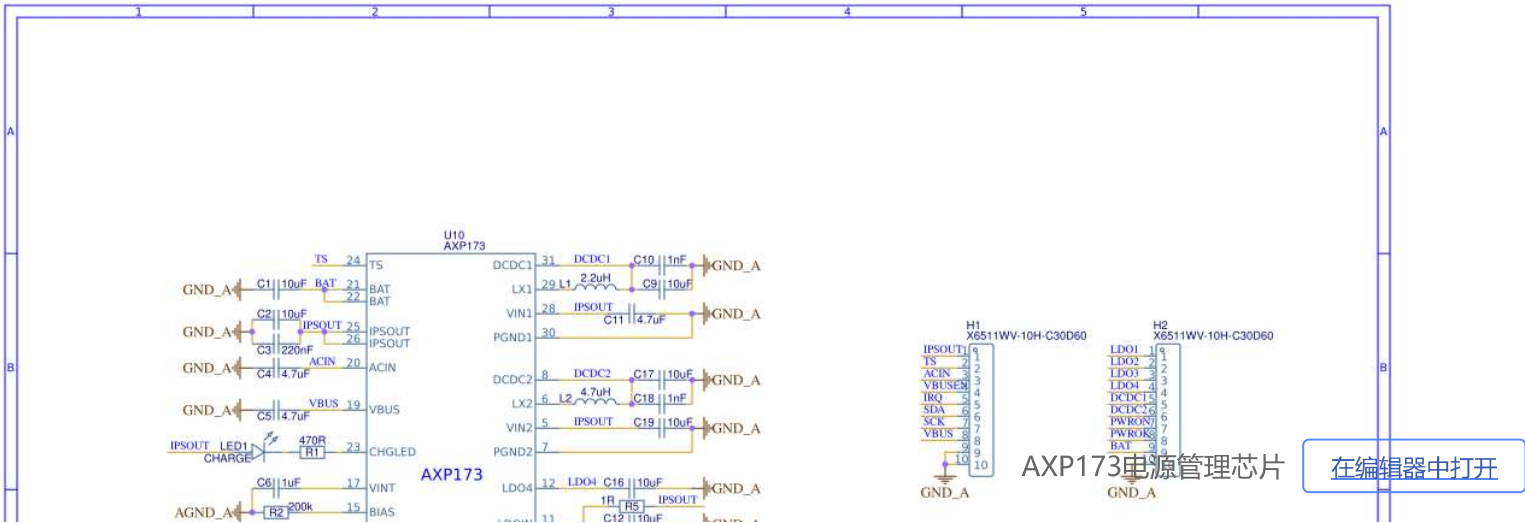


短按键息屏+长按键关机功能（需要设置IRQ中断引脚）视频演示比较清楚

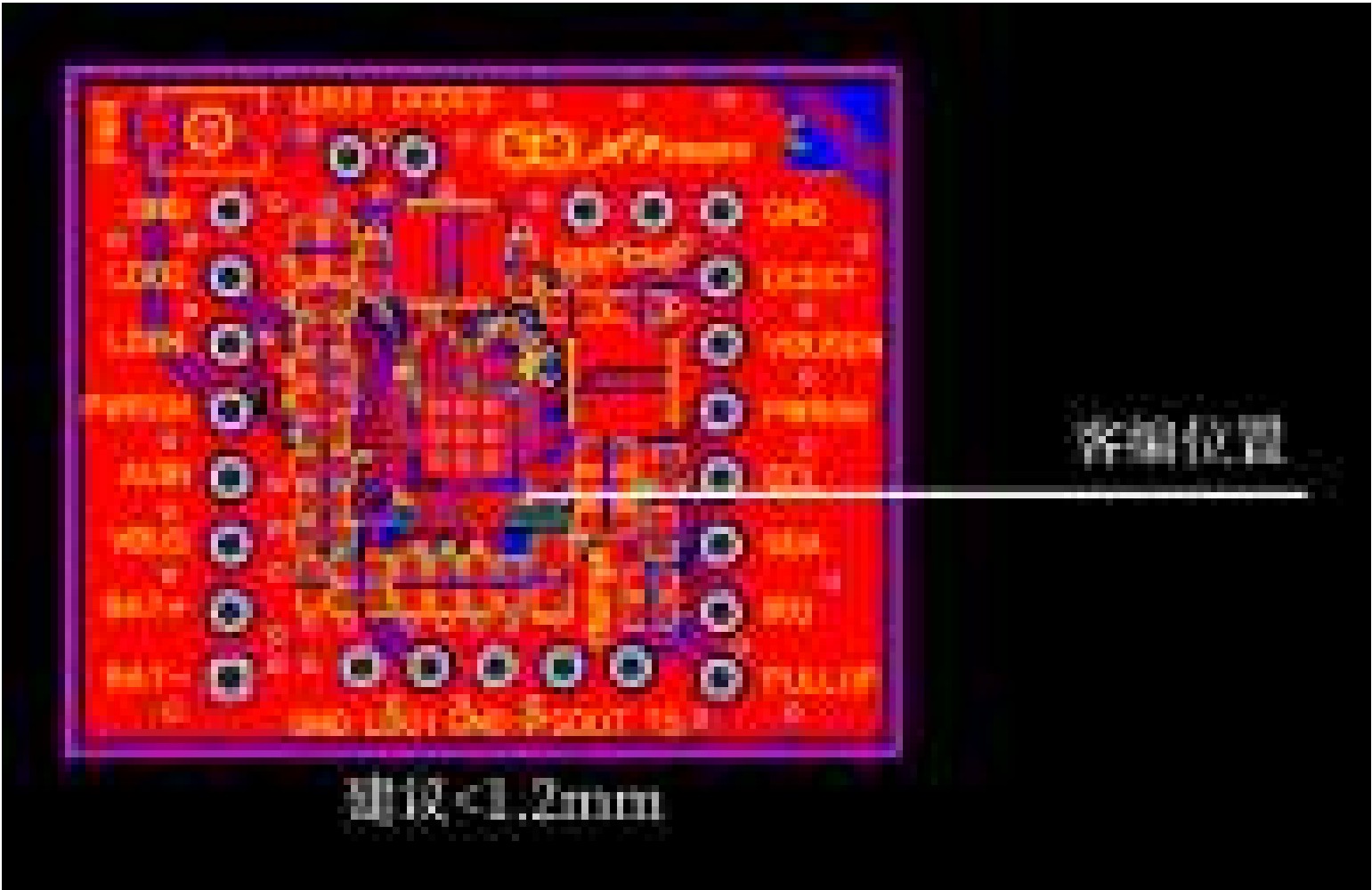


设计图

原理图



PCB



PCB AXP173

BOM

[BOM下载](#) 在立创商城下单

ID	Name	Designator	Footprint	Quantity	BOM_Supplier Part	BOM_Supplier	BOM_Material
1	10uF	C1,C2,C9,C12,C16,C17,C19	C0402	7	C315248	LCSC	CL05A10K
2	220nF	C3,C7	C0402	2	C76593	LCSC	C1005X5I
3	4.7uF	C4,C5,C11,C14,C15	C0402	5	C76995	LCSC	GRM155F
4	1uF	C6,C8,C13	C0402	3	C83063	LCSC	0402X10S
5	1nF	C10,C18	C0402	2	C14442	LCSC	CL05B10K

[展开](#)

工程附件

序号	文件名称	下载次数
1	AXP173.rar	32