电气量测控平台显示界面及数据接口说明

一、 目的

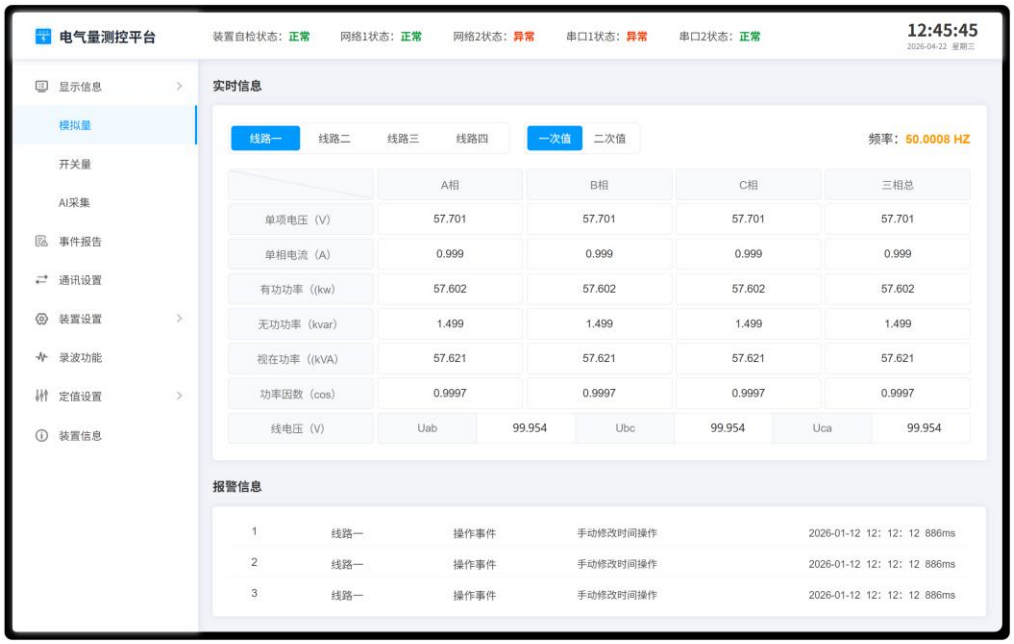
本文件对电气量测控平台人机交互界面显示内容、界面与底层系统之间的数据交互过程与接口做以说明。

二、 界面内容

人机交互界面分为左侧操控菜单和数据显示区域两部分组成。

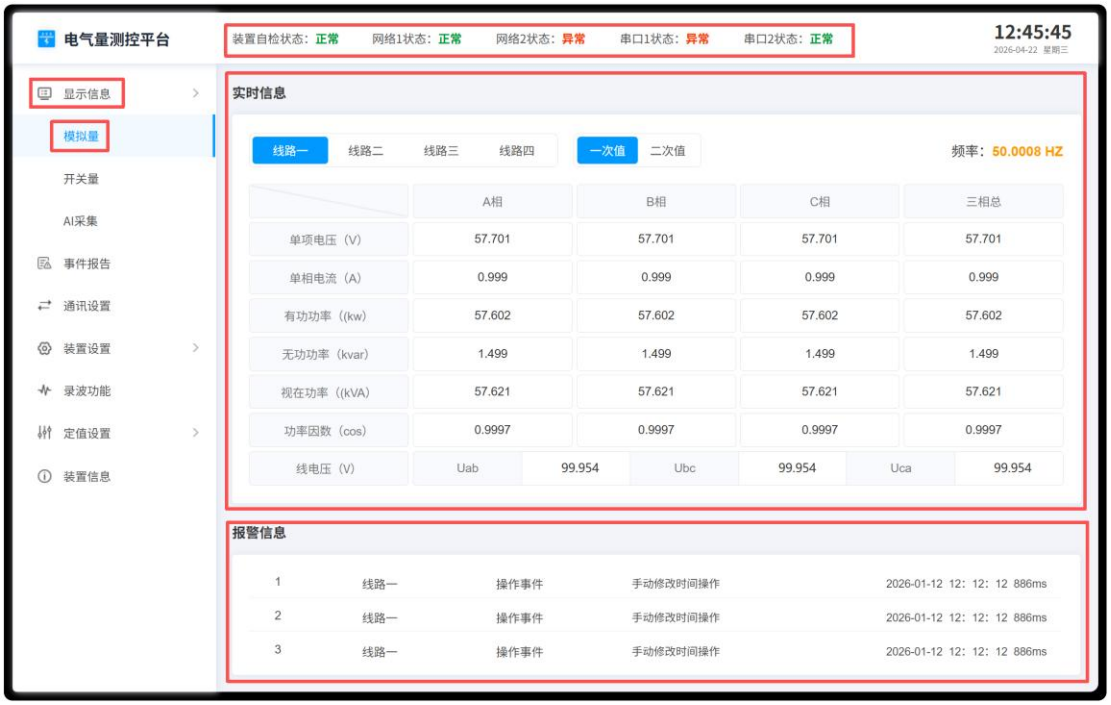
操控菜单主要包含设备运行状态显示、设备参数设置、设备报警设置、设备有关信息显示、设备运行管理设置等选项。

显示区域分为上部状态栏、中间主显示区和下部的报警显示框。中间显示区根据菜单选项动态显示对应的操控内容。



1、设备运行状态监测

主要包含：状态栏、显示信息-（模拟量、开关量、AI 采集）、报警信息。这些信息实时动态更新，对设备的运行状态进行监测与异常报警提示。



a) 状态栏

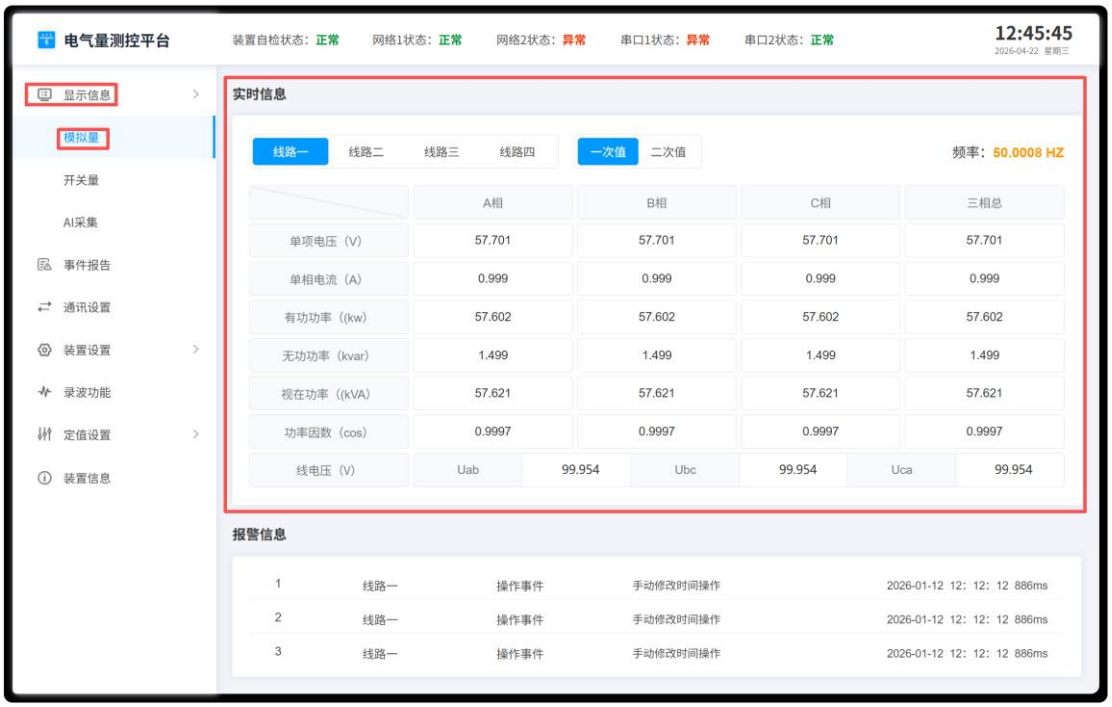
动态监测并实时显示设备的自检结果、各种通讯接口的当前工作状态。

b) 显示信息

i. 显示信息-模拟量

实时显示设备各条线路的运行电气参数，可按线路号选择显示如“线路一“，可切换显示该线路电气量的“一次值”/“二次值”。

数据刷新率不低于 0.5s。



ii. 显示信息-开关量

实时动态显示设备开关量采集通道的 12 路数据状态，以及开出量 12 通道当前状态。点击开出状态图标按钮，可以控制各通道输出节点的“分”/“合”。

。

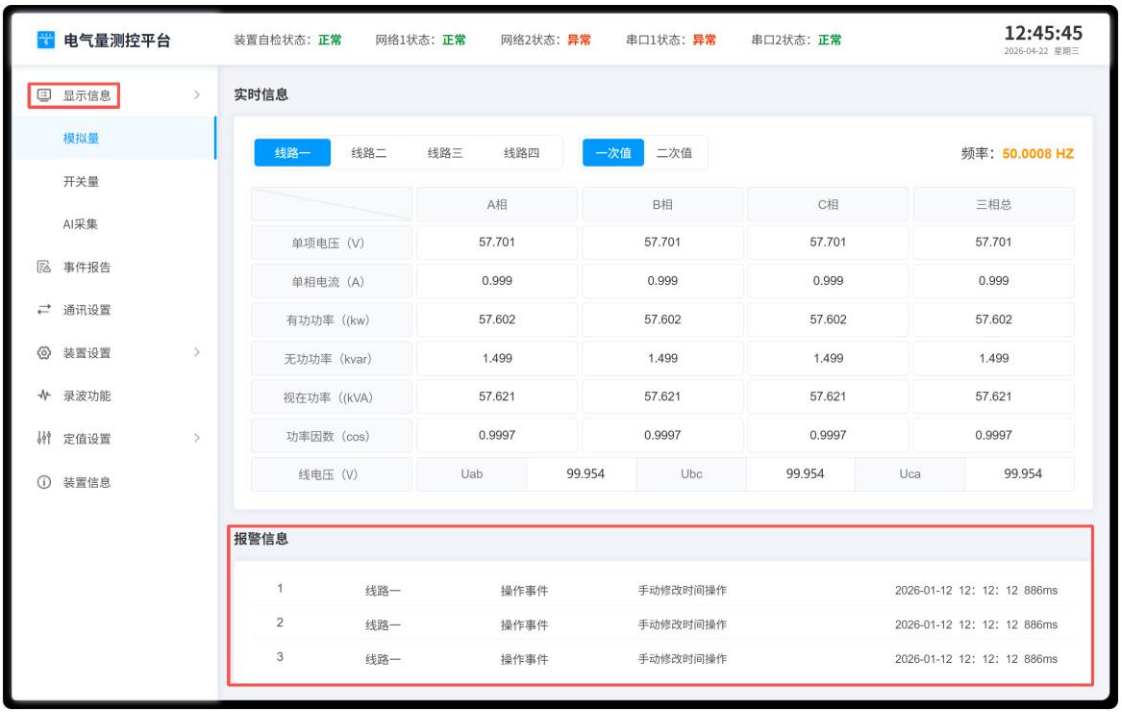
iii. 显示信息-AI 采集

实时动态显示设备 AI 采集通道的 12 路数据状态。本页显示内容由“装置

设置-AI 通道设置”决定，“原始值”为 AI 通道采集的数据，“所属线路”+“类别”为该通道在设置参数中映射的对应数据，“低值”和“高值”对应显示“AI 通道设置“中的“低值”和“高值”。

c) 报警信息

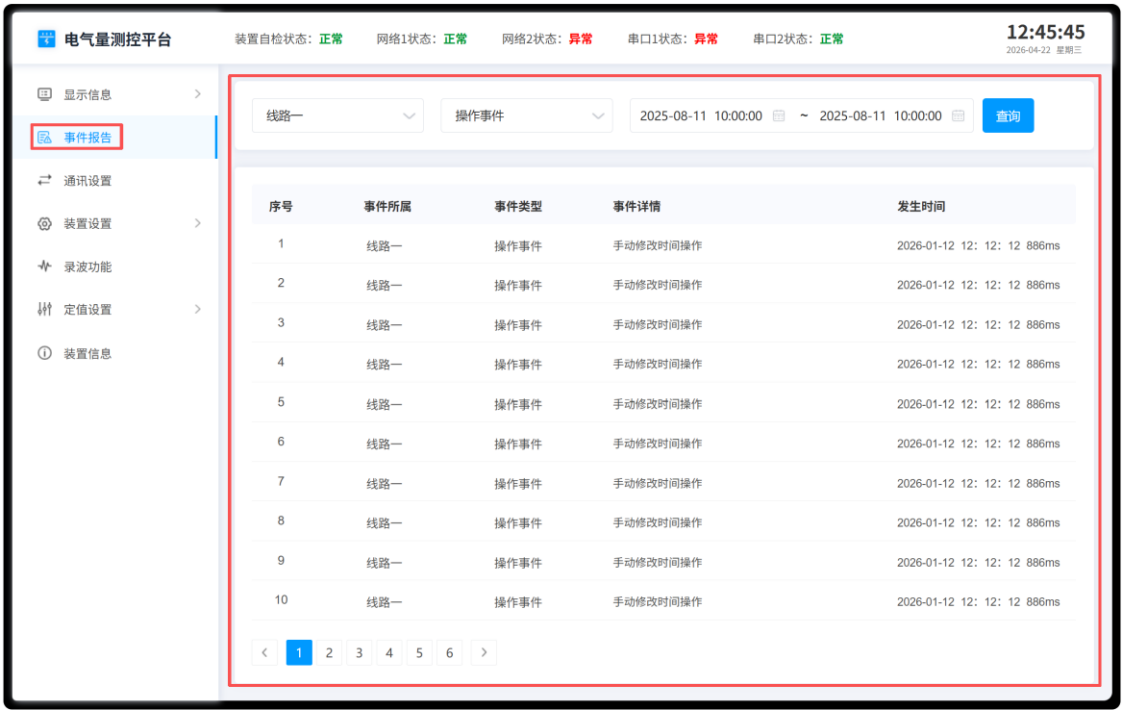
在显示页面的下方实时显示设备最新的报警信息。



2、事件报告

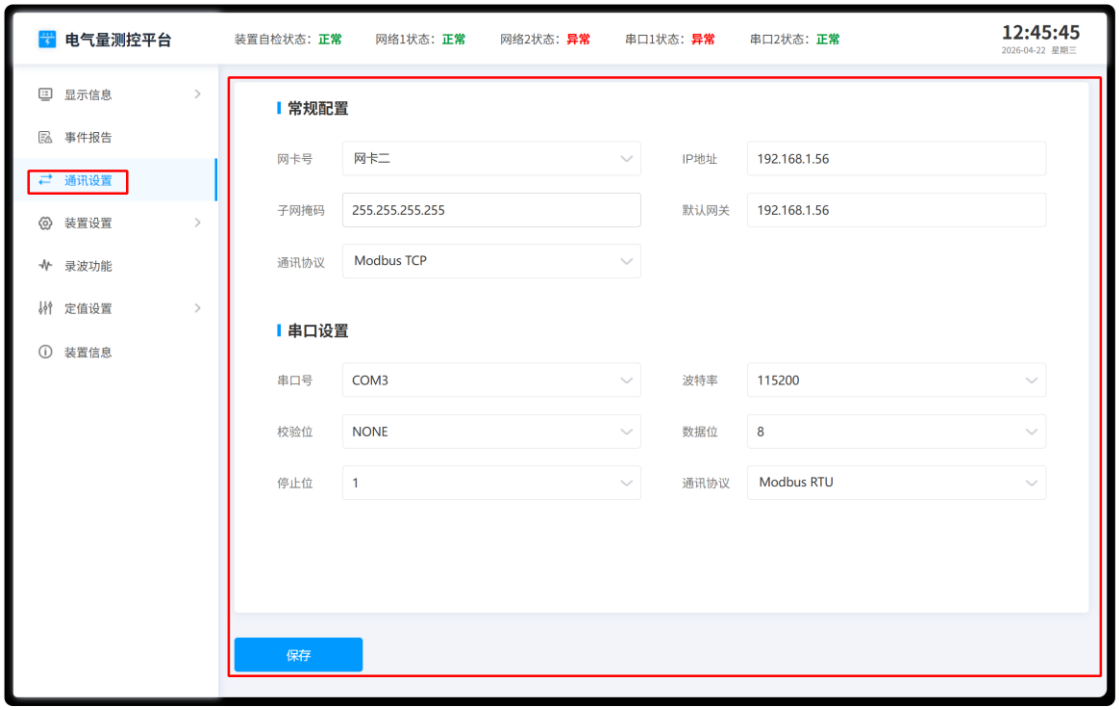
对装置发生过的所有事件记录进行分类查询。可以按照线路名称、时间类型、

时间段等分类检索。



3、 通讯设置

对装置网口、串口进行相关参数设置，选择对应的通讯协议类型。

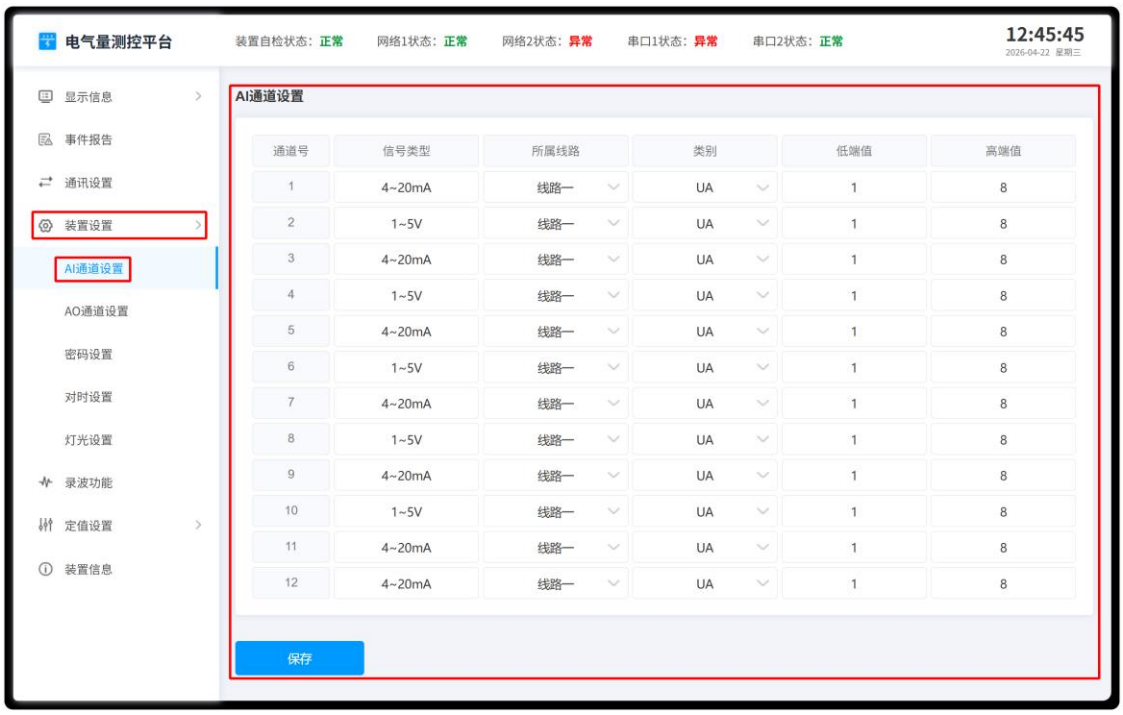


4、 装置设置

对装置的基础运行相关参数进行设置。

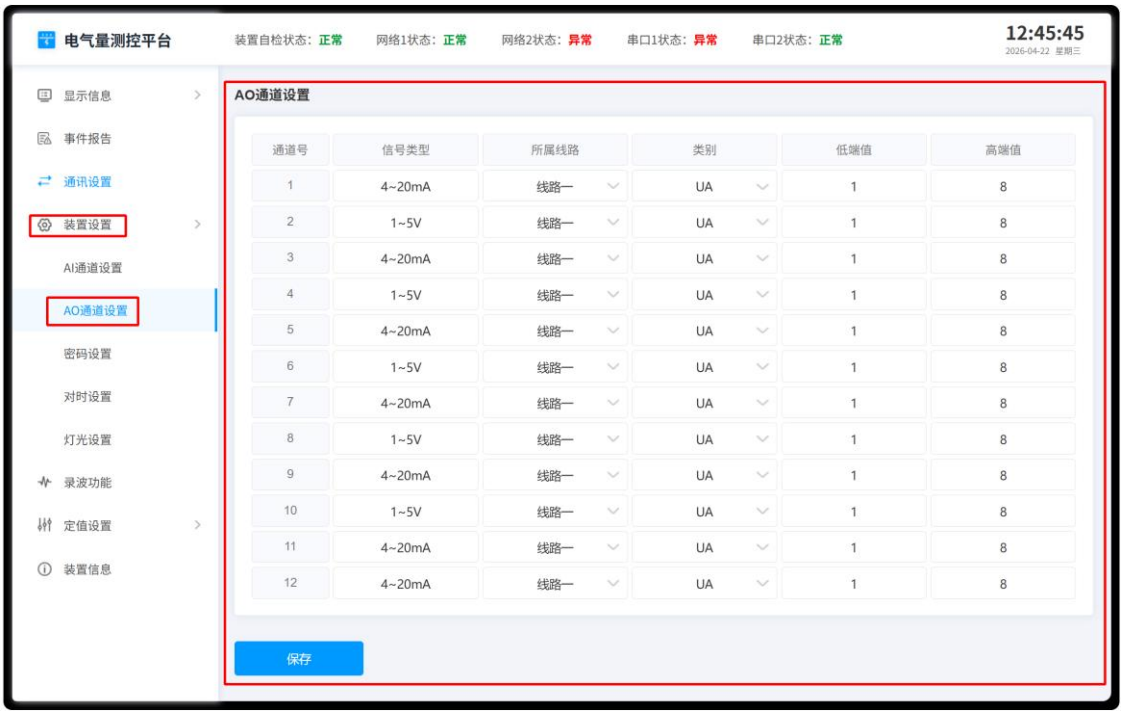
i. 装置设置-AI 通道设置

装置的 AI 采集板具备 12 路模拟量采集功能，可以灵活的选择每路 AI 采集的“信号类型”、本通道数据对应的“所属线路“、“类别”，以及设置本通道数据的“低端值”、“高端值”。

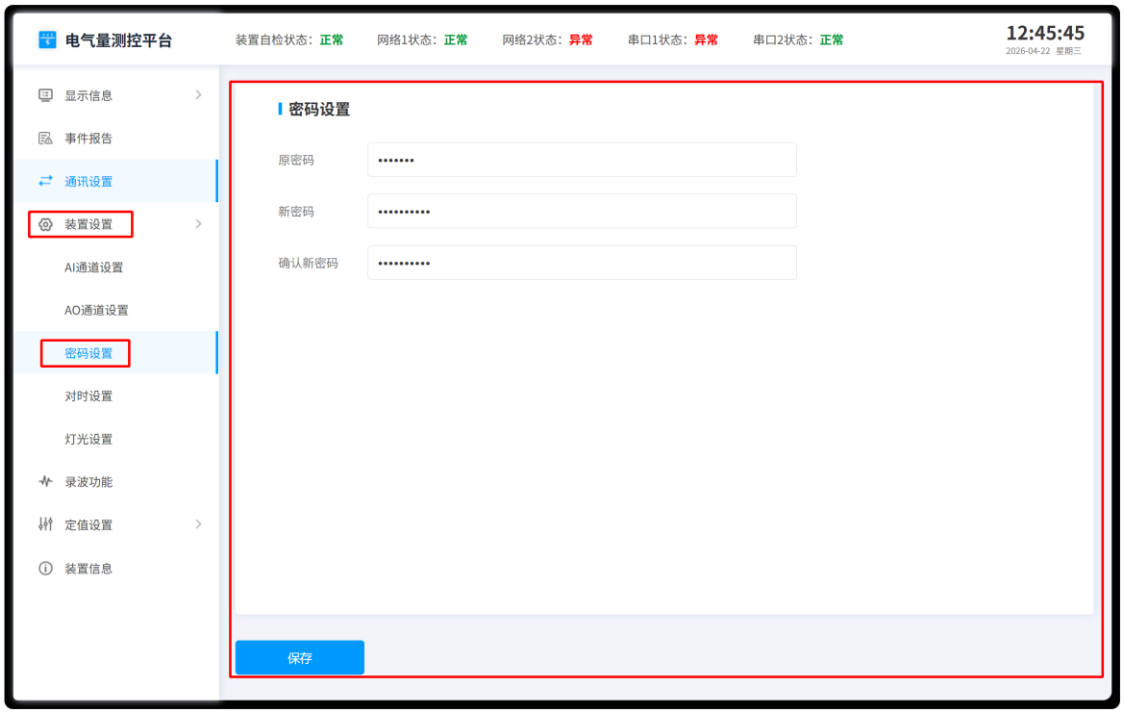


ii. 装置设置-AO 通道设置

装置的 AO 输出板具备 12 路模拟量输出功能，可以灵活的选择每路 AO 输出的“信号类型”、本通道数据对应的“所属线路“、“类别”，以及设置本通道数据的“低端值”、“高端值”。

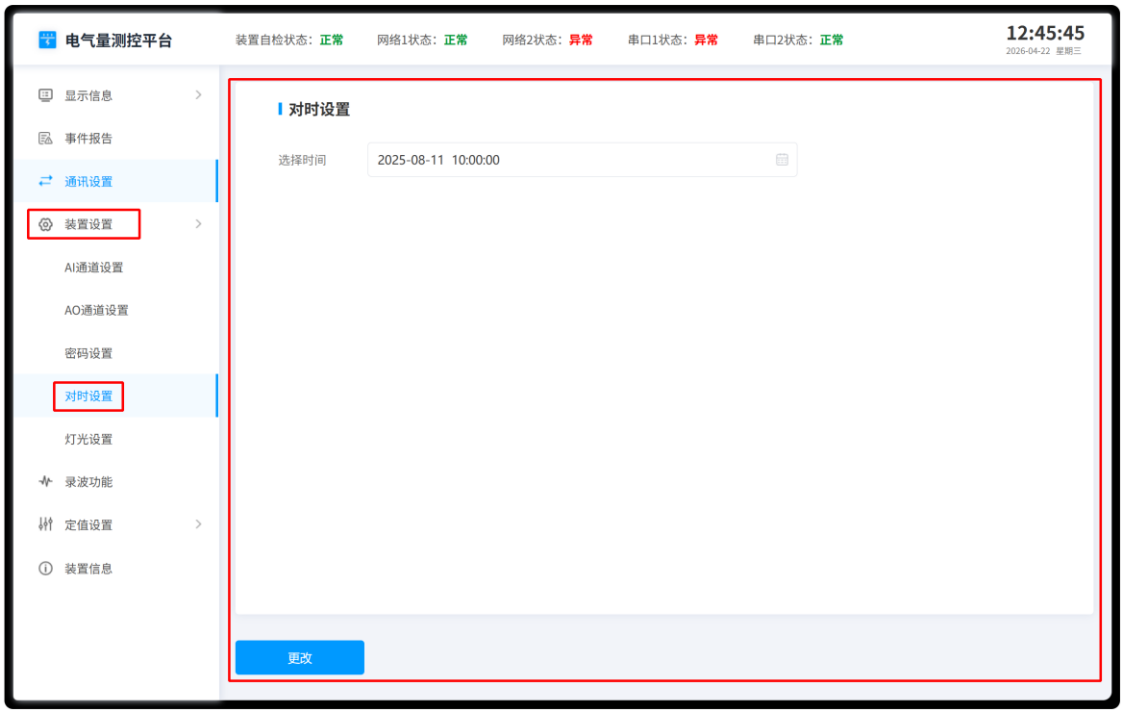


iii. 装置设置-密码设置



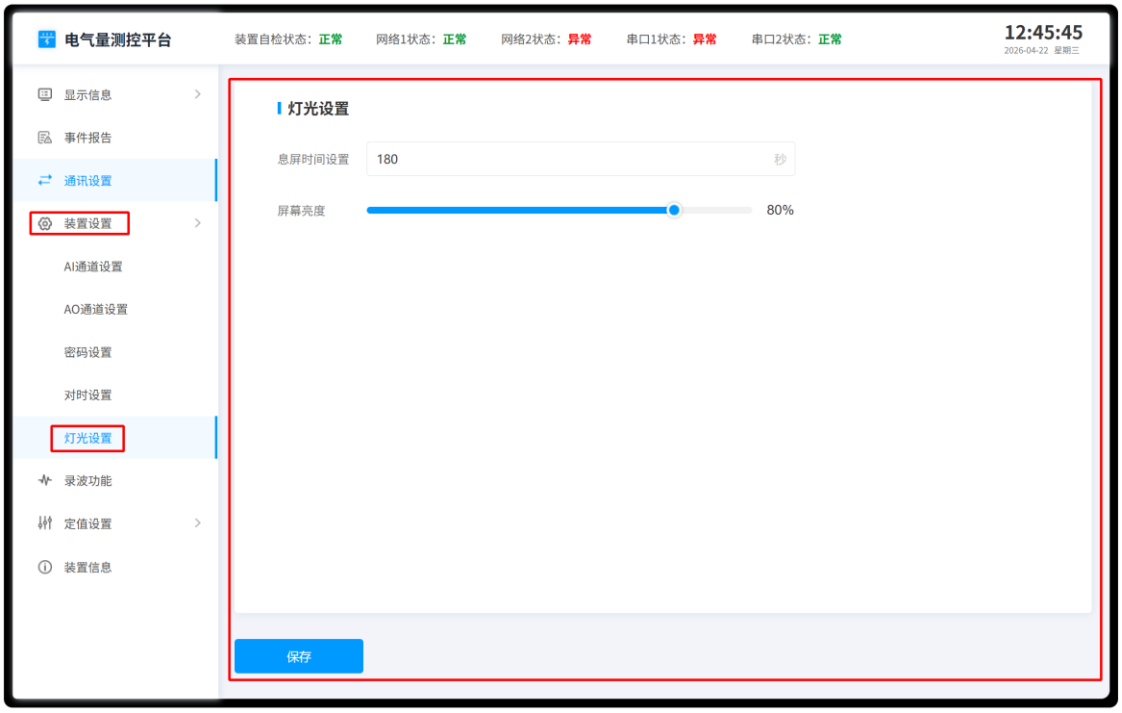
iv. 装置设置-对时设置

选择装置的对时方式，分为手动对时，B 码对时，主站网络对时等多种方式。



v. 装置设置-灯光设置

设置装置液晶屏幕进入屏保的时间，以及屏幕亮度。



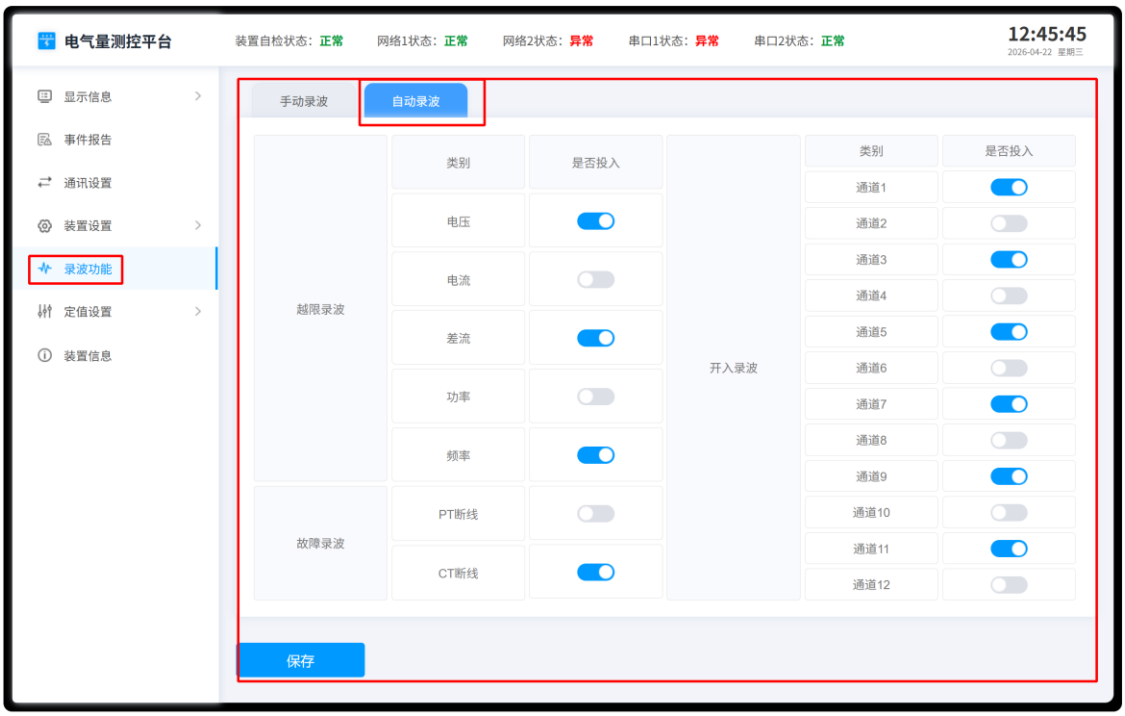
5、 录波功能

i. 录波功能-手动录波

点击“启动录波”按钮，可以生成当前状态下的录波数据，用于检验录波功能是否正常，以及查看录波数据具体包含信息。

ii. 录波功能-自动录波

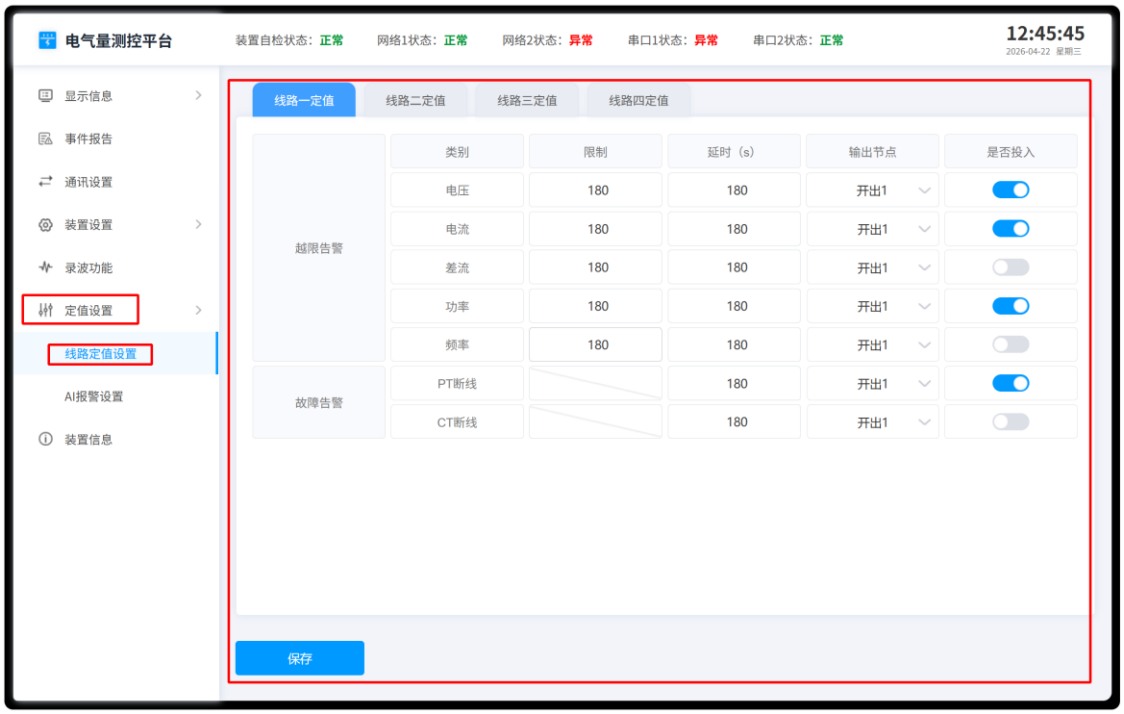
自动录波页面用于设置各类电气量超限录波、开关量通道状态变化录波、以及故障录波的投退设置。



6、 定值设置

i. 定值设置-线路定值设置

设置装置各条线路的超限报警“限值”、越限检测时间“延时”、选择报警动



作“输出节点”、以及“是否投入”报警。设置各条线路的“PT 变比”、“CT 变比”以及故障告警检测“延时”、告警动作“输出节点”、报警“是否投入”。

ii. 定值设置-AI 报警设置

设置装置 AI 采集通道的报警“下限值”和“上限值”、越限检测时间“延时”、选择报警动作“输出节点”、以及“是否投入”报警。

7、装置信息

三、 数据类型

上行数据：4 条线路电气量测量数据（UA、UB、UC、IA、IB、IC、P、Q、S、F、Cosφ、UAB、UBC、UCA）、12 通道 AI 采集数据、12 通道 AO 输出数据、12 路开关量采集数据、12 路开出控制数据、一周波（20ms）的原始采样数据（UA、UB、UC、IA、IB、IC、开关量）、b 码对时数据、事件记录（报警）数据;

下行数据：通过界面下发的各类参数数据，包括通讯类的网口和串口设置参数、装置类的 AI/AO 通道设置参数、装置管理密码/对时方式/灯光等设置参数。

需实时传输、保持动态刷新的数据：所有上行数据;

不定时传输的数据：根据数据发生修改、变化时传输，所有下行数据。

本地生成数据：各种界面操作记录，如开出控制操作、手动对时、参数修改记录、运行状态改变及自检产生的结果等，按照事件记录格式生成报警信息。

四、 数据的使用

1、 **上行数据：**通过表格、状态栏、文本窗、曲线波形方式，实时更新数据，直观展示设备当前运行状态。

列表显示：4 条线路电气量测量数据（UA、UB、UC、IA、IB、IC、P、Q、S、F、Cosφ、UAB、UBC、UCA）、12 通道 AI 采集数据、12 通道 AO 输出数据、12 路开关量采集数据、12 路开出控制数据;

状态栏：装置自检状态、网络与串口通讯状态、当前时间动态显示;状态改变提醒及事件记录生成与保存，如网络通讯故障、自检错误等

事件报告生成。

文本窗：实时报警信息显示、保存（生成 SOE 文件），历史事件报告可分类检索显示。

曲线波形：汇总每周波（20ms）的原始采样数据（UA、UB、UC、IA、IB、IC、开关量），以曲线方式绘制波形图，直观展示数据动态变化情况。按照相关标准生成录波文件并保存。

2、 **下行数据：**通过各种参数设置页面，完成数据修改后更新参数保存文件，并将更新数据下发给底层应用。

3、 **本地生成数据：**对 12 路开出量的分/合操作控制、以及各种参数的修改生成报警信息并保存，通过“报警信息“、“事件报告”窗口进行显示。

五、 数据接口

上下行数据通过 3568 系统内部核间通讯完成传输。具体数据定义如下：

```
#ifndef __GUI_H
#define __GUI_H

#include "time.h"

#define LINE_TOTAL_NUM 4 //线路数量
#define ANALOG_CH_NUM   12 //模拟量通道数

//实时量
typedef struct
{
    //三相电压
    float Ua;   //A 相电压
    float Ub;   //B 相电压
    float Uc;   //C 相电压
```

```
    //三相电流
    float Ia;   //A 相电流
    float Ib;   //B 相电流
    float Ic;   //C 相电流

    //有功功率
    float Pa;
    float Pb;
    float Pc;
    float Pt; //总有功功率

    //无功功率
    float Qa;
    float Qb;
    float Qc;
    float Qt; //总无功功率

    //视在功率
    float Sa;
    float Sb;
    float Sc;
    float St; //总视在功率

    //功率因数
    float PFa;
    float PFb;
    float PFc;
    float PFt; //总功率因数

    //线电压
    float Uab;
    float Ubc;
    float Uca;

    //频率
    float frq;

}measure_unit_t;

//B 码对时，时间数据
typedef union
{
    uint8_t b_code[6];
```

```
    struct
    {
        uint8_t ascii_s_unit     : 4;
        uint8_t ascii_s_ten      : 4;
        uint8_t ascii_min_unit   : 4;
        uint8_t ascii_min_ten    : 4;
        uint8_t ascii_hour_unit : 4;
        uint8_t ascii_hour_ten   : 4;
        uint8_t ascii_m_day_unit: 4;
        uint8_t ascii_m_day_ten : 4;
        uint8_t ascii_month_unit: 4; //4bit
        uint8_t ascii_month_ten : 4; //4bit
        uint8_t ascii_year_unit : 4; //4bit
        uint8_t ascii_year_ten   : 4; //4bit
    }time;
}b_code_t;

//开入、开出
typedef struct
{
    uint16_t in;     //每一位对应一个通道状态 ：  1 = 合，  0 = 分
    uint16_t out;    //每一位对应一个通道状态 ：  1 = 合，  0 = 分
}d_io_t;

//模拟量:AI 、AO
typedef struct
{
    float in[ANALOG_CH_NUM];     //模拟量输入
    float out[ANALOG_CH_NUM];    //模拟量输出
}a_io_t;

//事件
typedef struct
{
    struct tm t; //事件发生时间
    uint32_t ms; //事件发生的毫秒时刻
    uint8_t evt; //类型：0 = 无事件，  1 = 事件发生
}event_t;

//波形数据,
typedef struct
{    uint32_t stamp; //时标
    int16_t ch[8];   //4 条线路波形数据时：8 个通道：ua、ub、uc, ia、ib、ic、频率
系数、开入量
```

```
                    //AI 采集通道波形数据时 ： 8 个 AI 通道对应 D 的各个输入点
}ch_data_t;

//波形数据
typedef struct
{
    uint8_t    line_index; //线路号 ：  0 ~ 3 = 4 路电压电流采样数据，4 = AI 通道采集数据
    ch_data_t dot[64];    //64 个点组成一个周波
}waveform_data_t;


/**上行数据***************小核数据 -> 大核*******************************/
//小核数据 -> 大核
typedef struct
{
    measure_unit_t measure_unit[LINE_TOTAL_NUM];//总共 4 条线路，按序排列

    a_io_t d_io;      //模拟量输入、输出

    d_io_t a_io;      //数字量，开入、开出

    b_code_t b_code; //B 码对时

    event_t event[2];    //事件标志:event[0] = pt 断线，  event[1] = CT 断线

    waveform_data_t waveform; //实时一周波数据

}gui_in_t;

/****下行数据***********大核数据 -> 小核**********************************/
//界面任一参数修改，即将整个参数表更新下发一次

//串口参数设置
typedef struct
{
    uint32_t baud_tate; //波特率 ：  9600 115200（默认）
    uint8_t  parity;    //校验位 ：  0 = 无校验，  1 = 奇校验，  2 = 偶校验
    uint8_t  data;      //数据位 ：  0 = 8 位
    uint8_t  stop;      //停止位 ：  0 = 1 位，  1 = 2 位
    uint8_t  protocol;  //通讯协议 ： 0 = 无协议，1 = modbus
}com_param_t;

//AI、AO 模拟通道参数值
```

```
typedef struct
{
    uint8_t sig_type;//信号类型 ： 0 = 4~20ma 信号， 1 = 1~5V 信号
    uint8_t line_sub;//所属线路 ： 1 = 线路 1， 2 = 线路 2， 3 =线路 3， 4 = 线路 4， 0 = 装置
    uint8_t att;      //类别      ： 0 = UA, 1 = UB, 2 = UC, 3 = UAB, 4 = UBC, 5 = UCA, 6 = P, 7 = Q, 8 = F
    uint8_t u_limit; //上限值   :
    uint8_t d_limit; //下限值   :
}analog_ch_param_t;

//线路定值设置
typedef struct
{
    // uint8_t index;       //线路号:   1 = 线路 1, 2 = 线路 2， 3 = 线路 3, 4 = 线路 4
    uint8_t limit;   //限值:
    uint8_t delay;   //延时:          [1 ~ 10 秒]
    uint8_t out_num; //输出节点:     [1 = 开出 1、 2 = 开出 2、...]
    uint8_t select;  //投退:         [0 = 不投， 1 = 投入]
}line_param_t;

typedef struct
{
    line_param_t type[7]; //7 个类别：电压，电流，差流，功率，频率，PT 断线，CT 断线
}line_type_t;

//AI 报警设置
typedef struct
{
    uint8_t u_limit; //上限值   :
    uint8_t d_limit; //下限值   :
    uint8_t delay;   //延时:          [1 ~ 10 秒]
    uint8_t out_num; //输出节点:     [1 = 开出 1、 2 = 开出 2、...]
    uint8_t select;  //投退:         [0 = 不投， 1 = 投入]
}ai_alarm_param_t;

//下行数据：大核数据 -> 小核
typedef struct
{
    struct tm t; //时间设置
    d_io_t a_io; //数字量，开入、开出
    com_param_t com_param[4];              //串口参数设置
```

```
    line_type_t line_param[4];              //线路定值设置:  4 路定值一起下发
    analog_ch_param_t a_in_param[12];    //12 路模拟输入通道设置
    analog_ch_param_t a_out_param[12];  //12 路模拟输出通道设置
    ai_alarm_param_t ai_alarm_param[12];//12 路模拟输入  报警参数设置

}gui_out_t;

#endif
```

六、 补充说明

无