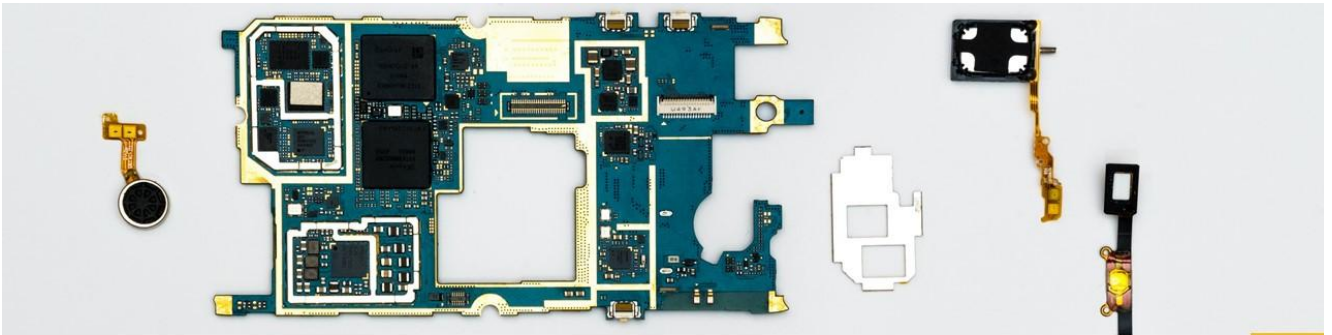


SPC-ESW-015 CONFIGURE TOOL MANUAL

<SPC100 配置工具>

<使用手册 >



SAFETY DESIGNER ENGINEERING TOOL 2023.11

2024 PRAJNASAFE



若彗电子科技

安全解决方案先行者

文档声明

<为求准确，本手册已经过验证和复审。本手册包含的指导和描述对 SPC100 配置工具 v1.2.0 是准确的。但是以后的 SPC100 配置工具及其手册可能变动，恕不另行通知。对直接或间接地由于产品与手册之间的错误、遗漏或差异而引起的损害，若彗电子科技（上海）有限公司不承担任何责任。>

本文为若彗电子科技(上海)有限公司财产，包含该公司的商业秘密。
对本文任何未经授权的使用和传播都是严格禁止的。

历史记录

版本号	编写日期	拟稿	描述
V1.0.0	2024-01-13	Scholar Su	初版

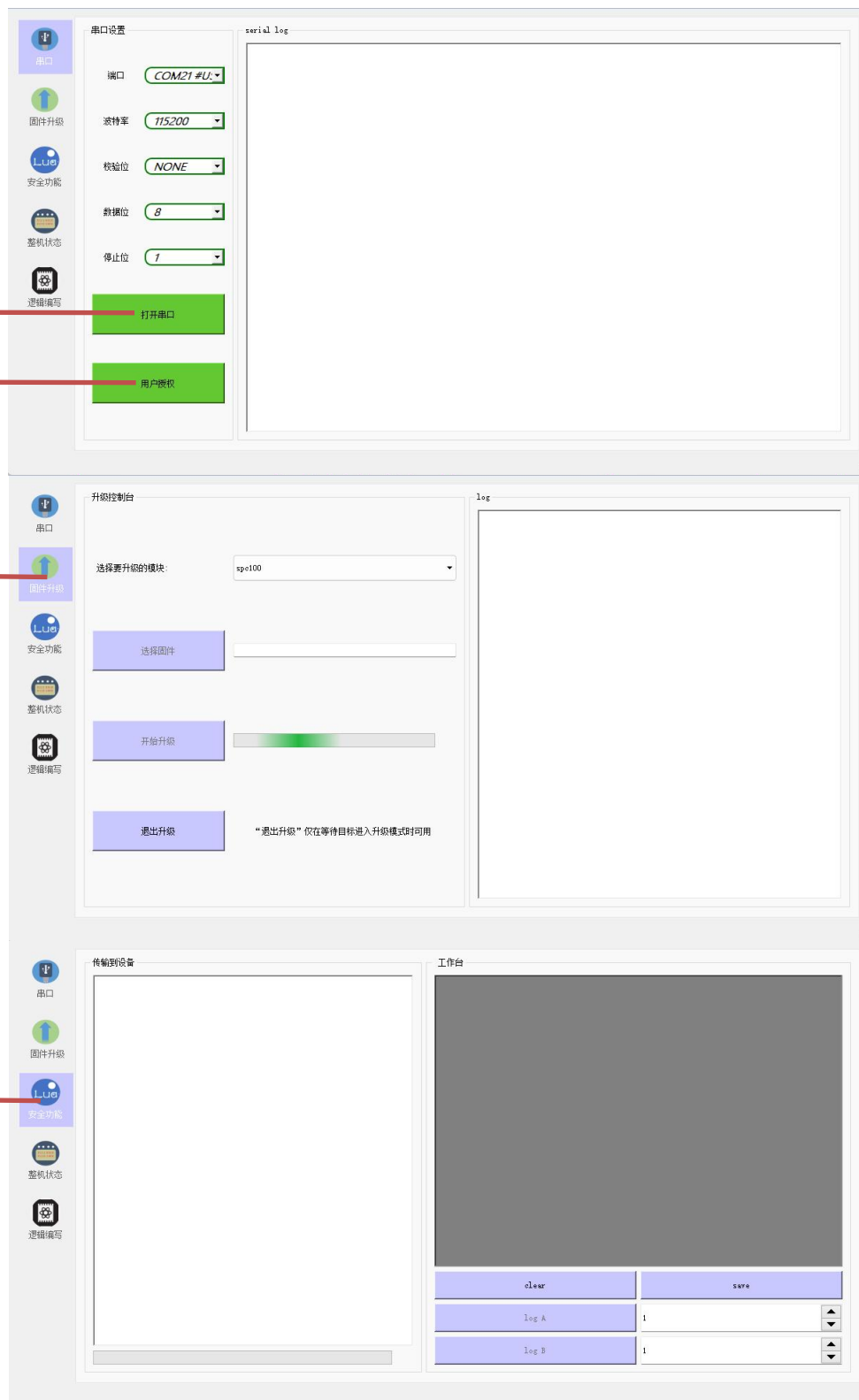
章节指引

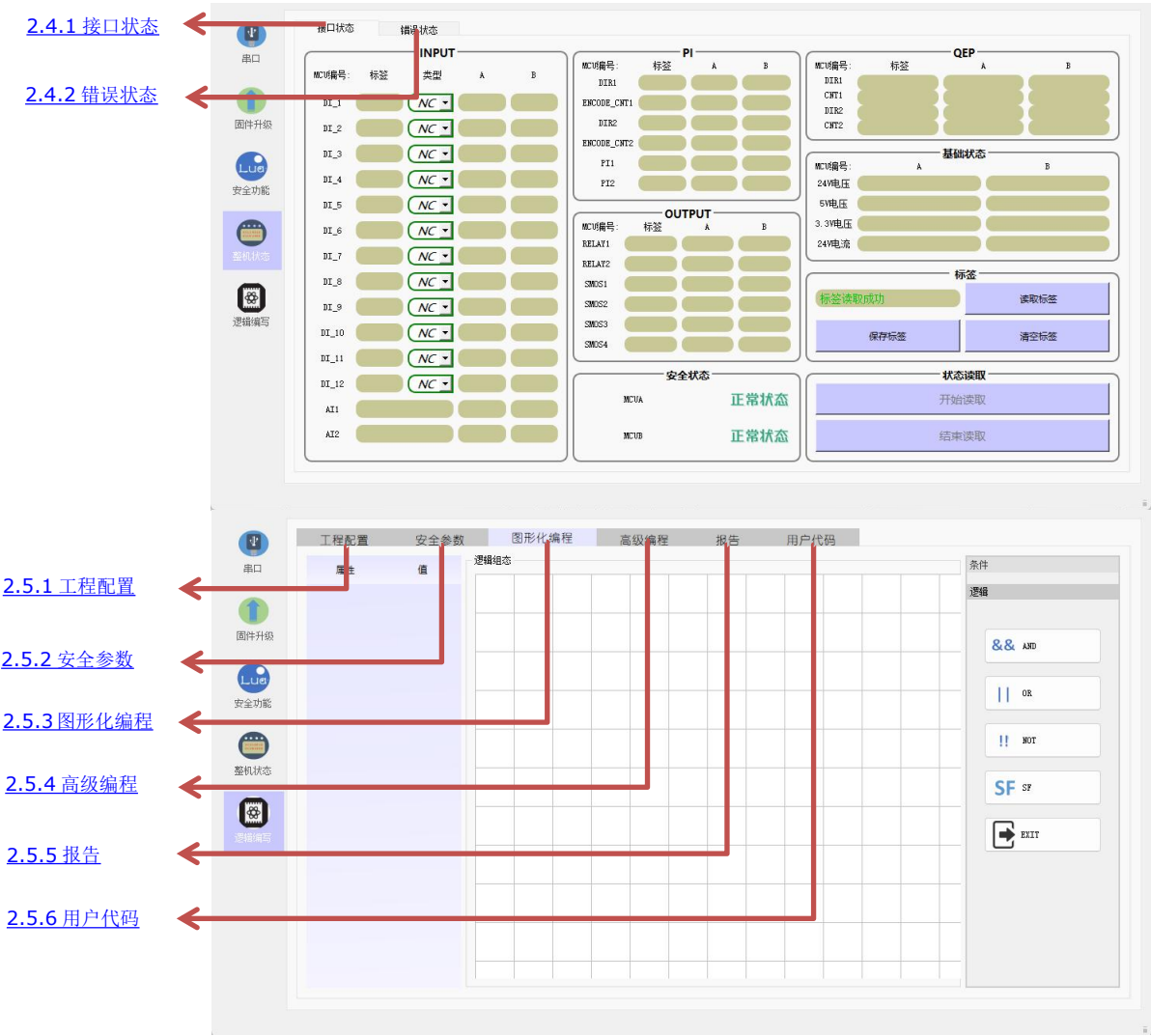
2.1.1 通讯端口

2.1.2 用户授权

2.2 固件升级

2.3 安全功能





1. 工具介绍

1.1 工具简介

用于对本公司生产的 SPC100 型号的安全控制器进行配置,本文档针对 SPC100 配置工具 v1.2.0 版本进行描述,此版本须搭配 SPC100 设备 bootloader 的 v1.6.0 或更高版本和 App 的 v1.7.0 或更高版本进行使用。

1.2 工具功能

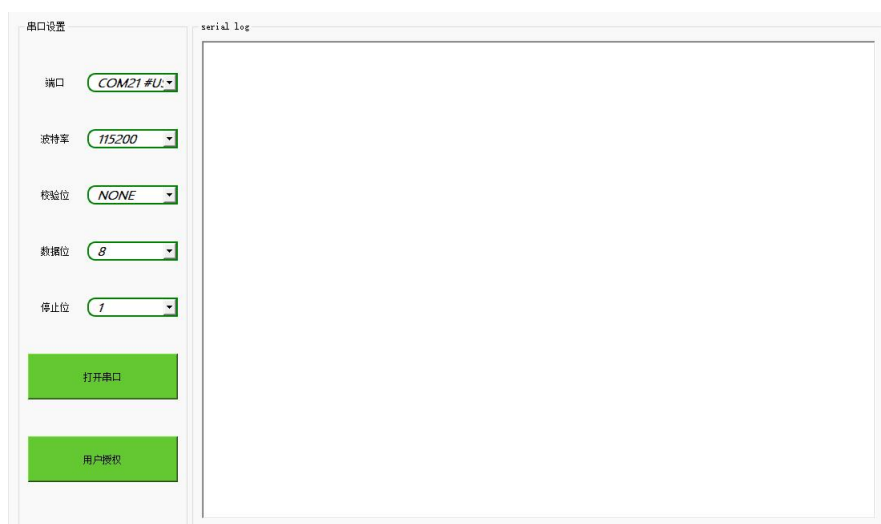
- 配置 SPC100 的安全配置参数
- 修改 SPC100 的设备密码
- 提供 SPC100 专用图形化组态编程
- 读取 SPC100 的接口状态
- 读取 SPC100 的故障状态

- 读取 SPC100 的版本号
- 读取 SPC100 的 Log

2. 工具使用

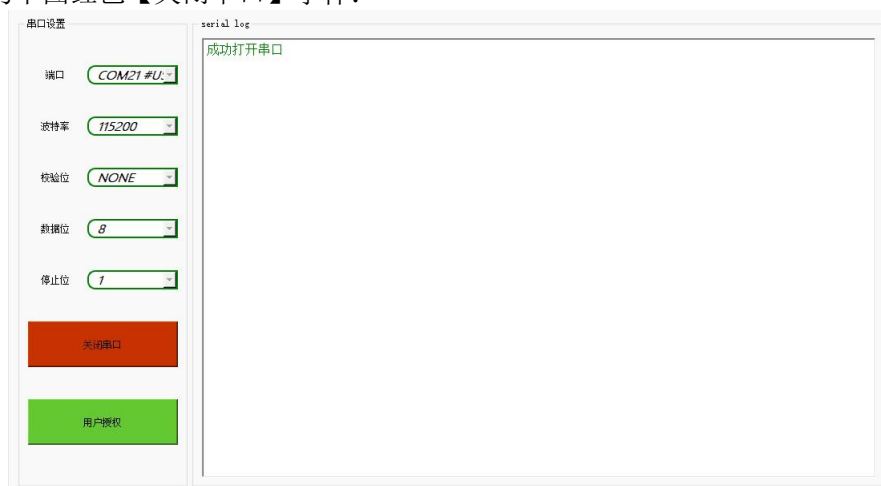
2.1 串口

2.1.1 通讯端口



在设备硬件连接状态良好的情况下，可在【串口设置】框体内的端口下拉框里选择正确的通讯端口，这是获取 SPC100 与配置 SPC100 的基础条件，其余配置保持默认即可。

点击【打开串口】按钮，即可开启端口，如果选择的端口正确且在可用状态，则在开启端口后，【打开串口】按钮则变为下图红色【关闭串口】字样：



2.1.2 用户授权

为了保护用户的设备不会轻易的被他人修改安全方面的配置，每台 SPC100 设备都有一个设备密码，其初始密码为“rhdzkj8888”。

为了防止未经过培训的人员操作配置工具，配置工具有授权密码，当授权密码填写正确时才允许操作配置工具，授权密码为“rhdzkj8888”，且不可更改。

在我们需要使用配置工具对 SPC100 设备进行工程下载、工程读取以及升级固件时，需要用户对工具进行授权。

点击【用户授权】按钮，用户可以在弹出的窗口内填写授权密码和设备密码：



当授权密码正确时，才被允许操作工具内提供的部分功能，如修改设备密码、固件升级等，当配置工具内用户填写的设备密码与 SPC100 设备内的设备密码一致时，设备才会允许工具对设备进行固件升级、工程读写等操作。

SPC100 设备密码的修改也可以在工具内完成，点击菜单栏内的【设备】菜单并选中【修改密码】选项



或点击工具栏内的修改密码工具按钮



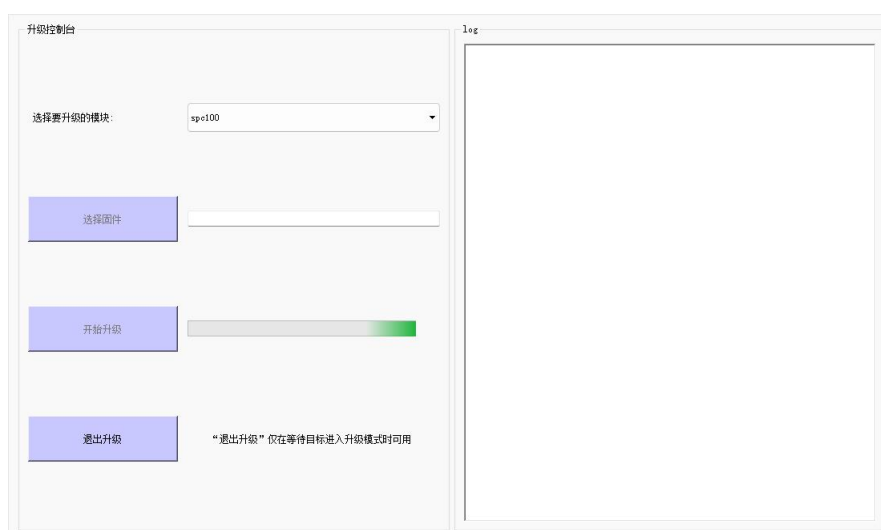
在弹出的修改密码页面可以对设备的密码进行修改：



设备密码修改注意事项：

- 原密码不可与新密码相同
- 密码长度为 1-16 个字符，仅支持大小写字母、数字和部分特殊字符
- 密码修改完成后重启 SPC100 设备才会生效

2.2 固件升级



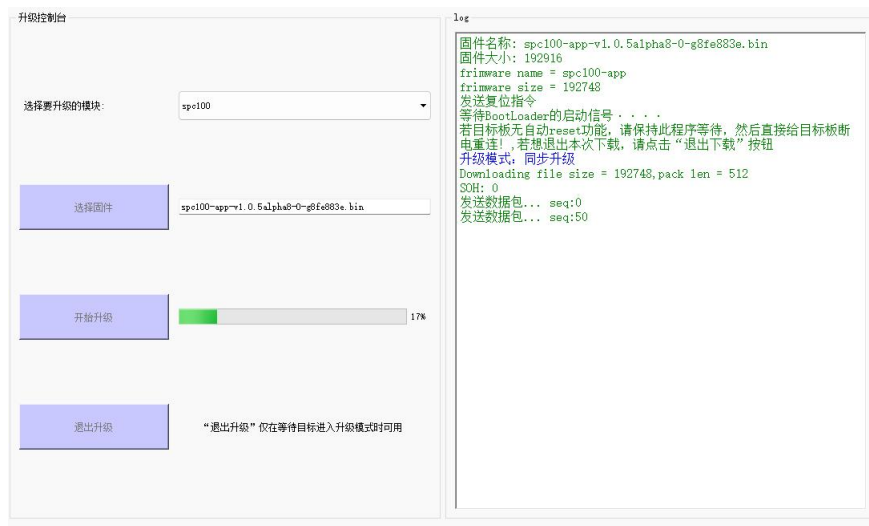
固件升级需要用户授权才可操作，且授权时输入的设备密码需要与 SPC100 设备内的密码相同。

点击【选择固件】按钮，在弹出的文件系统中选择本公司给出的 spc100 设备固件升级文件，升级文件一般格式为 spc100-app-v1.0.6-0-gb1a4802.bin 或更新版本

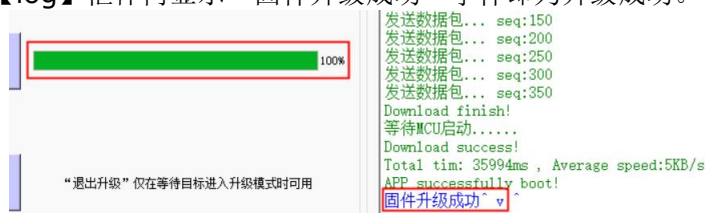
点击【开始升级】按钮，即可开始升级 SPC100 设备，但需要注意的是，SPC100 设备不允许版本降级行为，如当前 SPC100 设备版本为 v1.6.0，则无法升级为 v1.6.0 及更低版本，否则工具会报错：

A-Error: 当前版本较高，无需更新!
B-Error: 当前版本较高，无需更新!

正常升级流程如下：



当进度条达到 100%，【log】框体内显示“固件升级成功”字样即为升级成功。



SPC100 内有版本回滚功能，在升级失败时重启 SPC100 设备后会自动回滚到升级前的版本。

在通讯硬件链路存在问题的情况下，点击【开始升级】按钮后工具不会收到设备的回复，此时工具会等待设备连接：

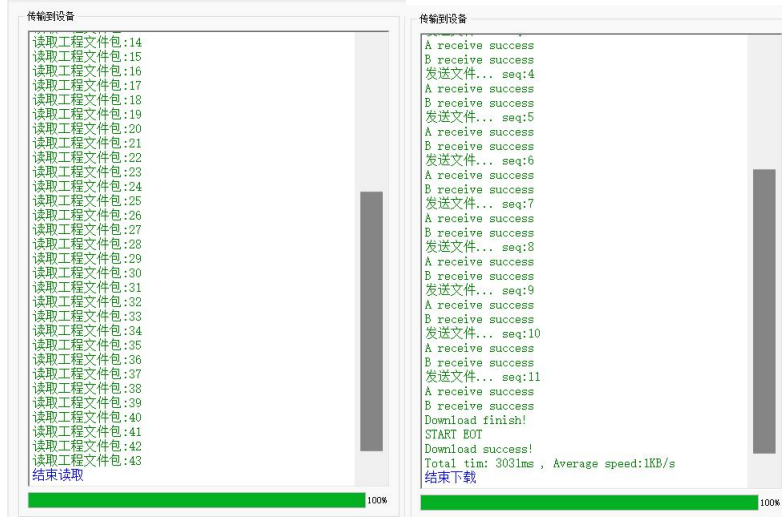
```
固件名称: spc100-app-v1.0.5alpha8-0-g8fe883e.bin
固件大小: 192916
firmware name = spc100-app
firmware size = 192748
发送复位指令
等待BootLoader的启动信号...
若目标板无自动reset功能, 请保持此程序等待, 然后直接给目标板断电重连!, 若想退出本次下载, 请点击“退出下载”按钮
```

此时请检查设备连接是否良好，在设备连接正常后重启设备或点击【退出升级】按钮退出本次升级再重新点击【开始升级】按钮即可。

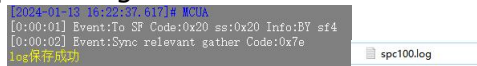
2.3 安全功能



【传输到设备】框体内会显示从设备读取工程和下载工程到设备的信息和进度：



【工作台】框体显示 SPC100 设备的启动信息、Log 等，点击【clear】按钮可以清除显示内容，点击【save】按钮可以保存显示内容到本地的 spc100.log 文件内，文件与工具本身同路径：



点击【log A】按钮可以读取 MCUA 的 log，在输入框内可以输入要读取 log 的大小，单位为 Kbyte。



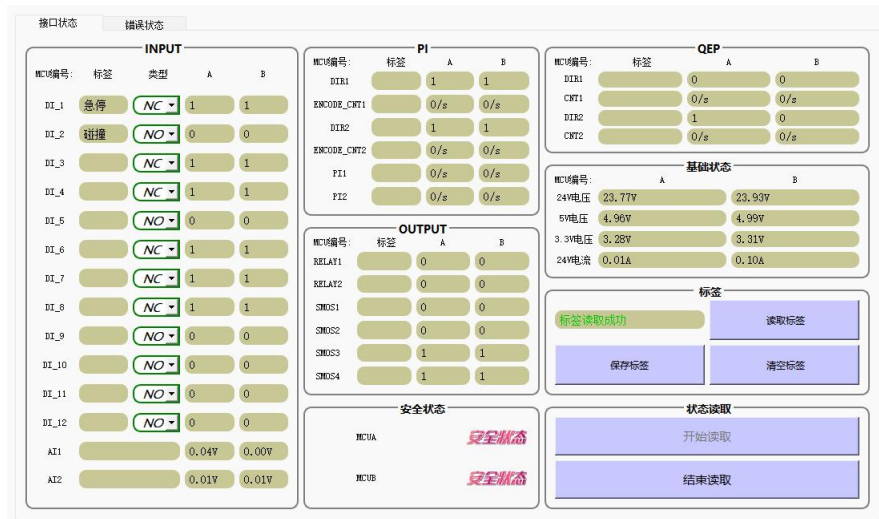
点击【log B】按钮可以读取 MCUB 的 log，在输入框内可以输入要读取 log 的大小，单位为 Kbyte。



2.4 整机状态

整机状态分为【接口状态】和【错误状态】两种

2.4.1 接口状态



The interface shows the '接口状态' (Interface Status) tab. It contains several sections: INPUT, PI, QEP, OUTPUT, 基础状态 (Basic Status), 安全状态 (Safety Status), and 状态读取 (Status Reading). The INPUT section lists DI1-DI12 with labels, types (NC/NO), and status (A/B). The PI section lists EMCOB2_CMT1, EMCOB2_CMT2, FI1, and FI2. The QEP section lists DI1, CMT1, DI2, and CMT2. The OUTPUT section lists RELAT1, RELAT2, SMO1, SMO2, SMO3, and SMO4. The 基础状态 section shows 24V, 5V, 3.3V, and 24V current/voltage levels. The 安全状态 section shows MCUA and MCUB safety status. The 状态读取 section has buttons for '开始读取' (Start Reading) and '结束读取' (End Reading).

在【接口状态】界面内，点击【开始读取】按钮，工具开始读取 SPC100 的各个接口状态，包括 AI、DI 等输入状态和 MOS、Relay 等输出状态以及电源、安全状态等等。

【标签】一列用户可以编辑，为输入资源编写标签防止输入混淆，而【类型】是 DI 资源独有的设置，可以设置 DI 默认是 NC/NO，当 DI 实际状态与默认状态不一致时，数据栏背景颜色会变为红色以提醒用户。

【标签】可以保存在本地，用户编辑完标签后点击【保存标签】，工具会在本地生成一个 spc100.label 文件并将标签存储在其中，在工具启动时会自动读取标签文件。

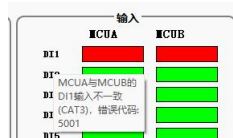
【清空标签】会清空当前显示的标签内容，但不会影响已经存储的标签文件。

2.4.2 错误状态



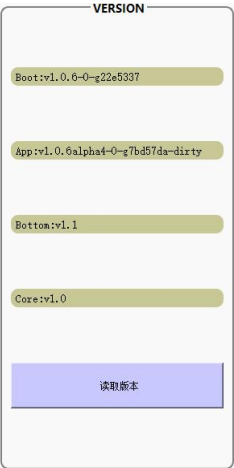
The interface shows the '错误状态' (Error Status) tab. It contains several sections: 电源 (Power), 输入 (Input), 输出 (Output), 芯片 (Chip), and 工程 (Project). The 电源 section shows 24V, 5V, 3.3V, and 24V current/voltage levels. The 输入 section shows DI1-DI12, AI1, AI2, QEP1, QEP2, FI_QEP1, FI_QEP2, FI_QEP_CN, FI1, and FI2. The 输出 section shows MOS1, MOS2, MOS3, MOS4, RELAT1, and RELAT2. The 芯片 section shows MCUA and MCUB status for various components like From, Flash, Register, Clock, Counter, 堆栈溢出 (Stack Overflow), Ram, 同步断联 (Synchronous Disconnection), and CAN断联 (CAN Disconnection). The 工程 section shows software version, project version, safety parameters, and project validity.

在【接口状态】页面点击【开始读取】按钮后，工具也会同步读取错误状态并显示，当 SPC100 正常时对应的界面指示灯为绿色；有错误时，错误指示灯变为红色。



This detail view shows the error status for DI1. It indicates that DI1 is in an error state (red). The error code is 5003, and the error description is 'DI D11输入不一致' (DI D11 input inconsistent).

如上图，鼠标悬停在 DI1 上时会弹出错误释义及错误代码，用户也可读取 log 来查看错误记录。



点击读取版本，可以读取 SPC100 设备内当前的软件版本。

2.5 逻辑编写

由于工具刚启动时无工程，所以逻辑编写界面在初始时是不可用的，当新建工程、导入工程或从设备成功读取工程后才会变为可编辑状态。

2.5.1 工程配置

2.5.1.1 工程设置



用户在此界面可以编辑项目相关信息，在项目第一次保存之后，项目名称就不可再被编辑。
项目保存路径在用户导入本地工程和保存工程到本地后会自动显示工程路径，无需用户编辑

【启用高级编程】是为有编程基础的客户提供比较复杂的线程编写功能，可供用户编写自己的复杂逻辑到线程，使能【启用高级编程】后，会新增【高级编程】界面在【图形化编程】界面之后。

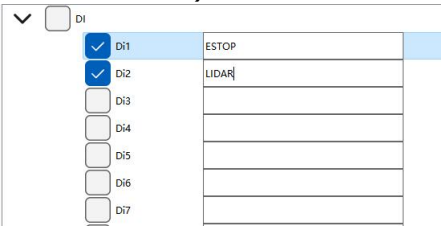


有关【高级编程】的功能介绍请查看 [2.5.4 高级编程](#) 章节

2.5.1.2 设备输入资源配置



设备输入配置用于用户配置要使用的 SPC100 设备上的输入资源，以 DI 资源为例，选中 DI1 与 DI2 作为将要使用的输入资源，请分别命名为 ESTOP 和 LIDAR(别名编辑框最多可输入 25 字符，并仅支持输入大小写字母、数字及_，并且第一个字符必须为大小写字母)



对于 SPC100 设备，其 DI9 和 DI10 分别是 AI1 和 AI2 的复用，而 DI11 和 DI12 是 PI1 和 PI2 的复用，他们之间存在互斥关系。

如当 PI1 和 PI2 被复用为 QEP 时，则 PI1、PI2、DI11、DI12 则变为灰色不可选中状态，具体复用设置及互斥关系请查看 [2.5.2 安全参数](#)

2.5.1.3 设备输出资源配置



设备输出配置内可以配置 SPC100 的输出模式，用户最多可设置 10 种输出模式，在 SPC100 设备上含有 Realy 以及 mos 共 6 个输出，在此处可分别对这 6 个输出做模式配置，点击鼠标右键调出右键菜单可选择【新增】和【删除】来新增模式或删除某个模式



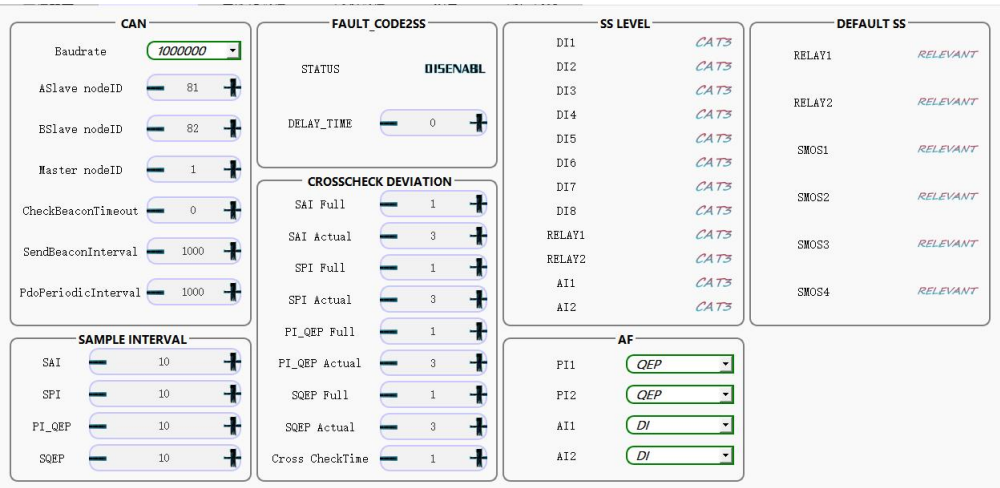
点击【新增】，会在当前页面新增一条输出模式，他们的编码(code)会自动生成并且唯一

可以在下拉框内选取输出资源与输出模式相关(relevant)和不相关(not_relevant)，与输出模式相关的资源在输出模式被调用时会将相关资源的输出断开，如果不相关将不会有任何操作，输出模式会在【图形化编程】内配置给某个安全功能，当安全功能被触发时，会调用其配置内的输出模式。

设备输出配置内会存在一个默认的输出模式，此模式不可被删除，当系统内的一些非用户设置的系统级的安全功能触发时(如自检失败等)会调用此模式，此模式编码固定为 0xff,此设置在【安全参数】界面也有设置项，属于 SPC100 设备安全参数的一部分。

在【设备输出配置】界面修改默认输出模式，【安全参数】界面的设置项也会同步改动，具体参数请查看 [2.5.2 安全参数](#)

2.5.2 安全参数



鼠标悬停在参数上可以查看参数的阈值范围和单位

【CAN】参数栏内可以设置与 CANopen 相关的一些参数：

Baudrate: CAN 波特率

ASlave nodeID: MCUA 节点 ID

BSlave nodeID: MCUB 节点 ID

Master nodeID: 主机节点 ID

CheckBeaconTimeout: 主机心跳检查时间，用户周期检测主机心跳判断主机是否掉线，为 0 时不检查主机心跳

SendBeaconInterval: SPC100 设备节点发送心跳的事件间隔

PdoPeriodicInterval: CANopen 的 PDO 数据的定时发送时间间隔，为 0 时为事件发送

【SAMPLE INTERVAL】参数栏内可以设置输入资源的采样间隔

【FAULT_CODE2SS】参数栏内可以设置是否启用硬件错误进入安全状态并设置延时时间，如果使能，则当硬件发生故障时(如 CAT3 架构下 DI1-a 与 DI-b 不一致)延迟 DELAY_TIME 后进入安全状态。

【CROSSCHECK DEVIATION】参数栏内可以设置 MCUA 与 MCUB 之间允许的输入差异大小。

FULL 为满量程百分比，Actual 为当前实时数据的百分比。

Cross CheckTime: MCU 之间的互检间隔，单位为(25ms)。

【SS LEVEL】参数栏内可以设置输入的架构等级(CAT2/CAT3)

【AF】参数栏内可以设置 PI 和 AI 的复用

当 PI 复用为 PI 时，PI_QEP1、PI_QEP2、DI11、DI12 将不允许使用

当 PI 复用为 QEP 时，PI1、PI2、DI11、DI12 将不允许使用

当 PI 复用为 DI 时，PI_QEP1、PI_QEP2、PI1、PI2 将不允许使用

当 AI1 复用为 AI 时，DI9 将不允许使用

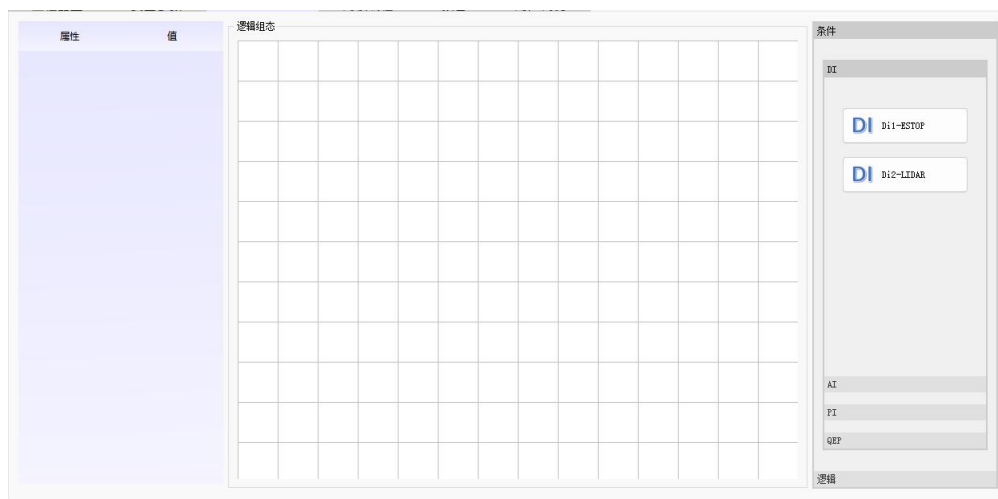
当 AI1 复用为 DI 时，AI1 将不允许使用

当 AI2 复用为 AI 时，DI10 将不允许使用

当 AI2 复用为 DI 时，AI2 将不允许使用

【DEFAULT SS】为 默认输出模式，并可设置输出资源与输出模式是否相关，当系统默认(非用户设置)的安全状态触发时会调用此模式，此输出模式 code 为 0xff，在此处修改参数会在【设备输出配置】界面同步修改。

2.5.3 图形化编程



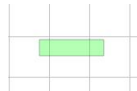
界面右侧的窗体为我们图形化编程的模块，分为【条件块】和【逻辑块】。

中间【逻辑组态】窗体为用户进行图形化编程的界面，用户可以拖动右侧的模块到中间窗体内进行生成对应的模块。

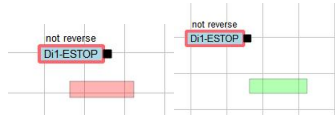
左侧为属性窗体，点击【逻辑组态】窗体我们已经生成的块，可以显示块的状态和属性。

2.5.3.1 块的生成

用户可以点击并拖动右侧窗体中的模块到【逻辑组态】窗体内,当拖动到【逻辑组态窗体】内时会跟随鼠标出现一个虚化的框体



绿色代表此处可放置并生成块, 由于块与块之间不可重叠且块块之间需要保持一定的距离, 所以如果【逻辑组态】内已经有了一个用户原来生成的块, 如果拖动一个新的模到这个已有的块附近时虚化框体会变为红色, 代表此处不可被放置, 而在块更远的位置放置则会被允许放置。



2.5.3.2 块的状态

鼠标点击【逻辑组态】内我们已经生成的块上, 在左侧属性栏中可以看到块的属性及状态

属性	值
ID	1
Name	Di1-ESTOP
Source	Di1
Reverse	Not
Error	Output1 not connect

当一个块处于错误状态时, 其外框体会变为红色, 在属性框体的【Error】行可以看到错误内容



例如上图, 错误状态为当前块的输出点未连接其他块, 值得注意的是, 我们拖动到【逻辑组态】内的块都需要是有意义的。

当一个块处于正常状态时, 其外框体会变为黑色

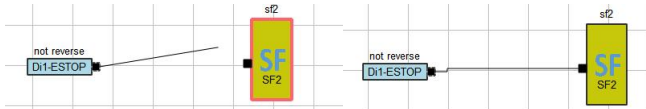


2.5.3.3 块的连接

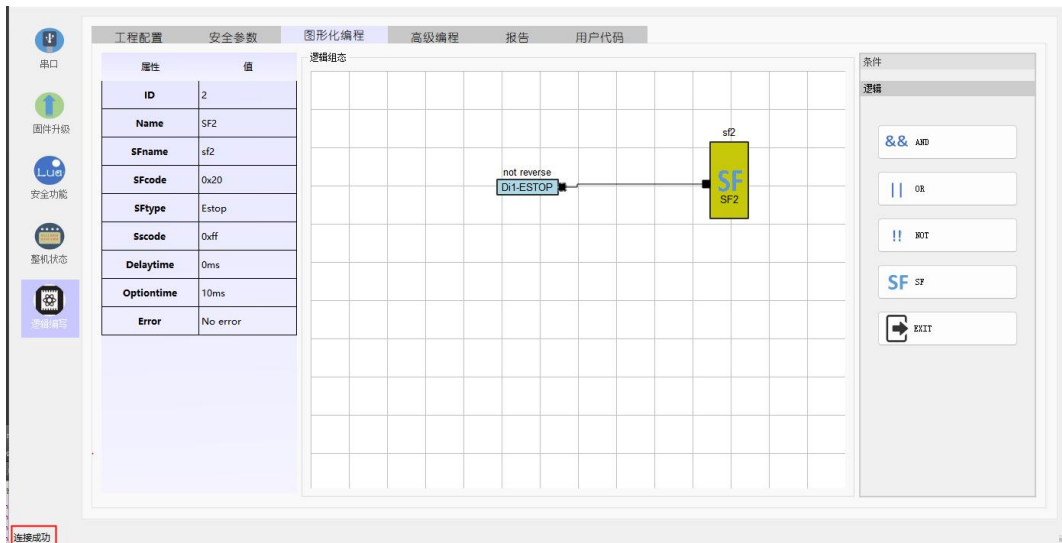
在每一个块的左侧或右侧会有黑色方块连接点, 默认为左侧为输入连接点, 右侧为输出连接点, 仅支持输入连接点和输出连接点的连接, 输出与输出或输入与输入之间不可连接。

值得注意的是块本身的输出连接点不可连接自身以及其父辈的输入以防止生成死循环逻辑。

鼠标点击块上的连接点并拖动鼠标到另一个连接点上并松开鼠标, 就可以完成块与块之间的连接



在界面的左下角可以看到我们连接的结果



2.5.3.4 块的删除

对于【逻辑组态】窗体内用户已经生成的块，鼠标悬停在块上并点击鼠标右键可以调出块的右键菜单，点击菜单内的删除即可删除已有的块



也可按键盘上的 **delete** 键来快速删除选中的块

当块被删除时，连接在块上的连接线也会被自动删除

2.5.3.5 条件块

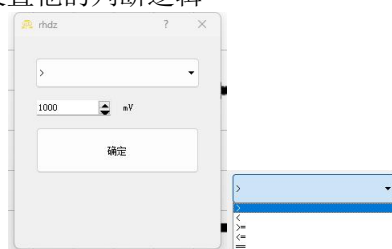
条件块分别有 DI/AI/PI/QEP，对应【工程配置】界面【设备输入配置】中我们对资源的选取和配置。

对于已经生成的块，我们可以鼠标悬停在块上又按下鼠标右键来选中右键菜单中的设置按钮(或双击块)来设置块，但每种条件块的设置是有区别的

对于 DI 条件块的设置本版本仅有输出反向的设置，DI 条件块默认输入高电平有效(输入高电平条件成立)，如果想更改为输入低电平有效则可以在设置界面选中输出反向，DI 条件块当前的设置简报会显示在块的上方，其他块同理



对于 AI/PI/QEP 条件块，我们可以设置他的判断逻辑



AI/PI/QEP 之间的区别在于阈值的单位和大小范围

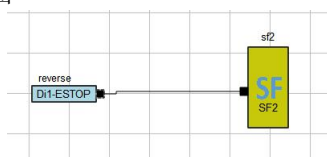
所有条件块都只有输出点，没有输入点。

2.5.3.6 逻辑块

逻辑块当前版本共有五种，其中三种为常用的与或非，主要介绍另外两种，分别为 SF 和 EXIT

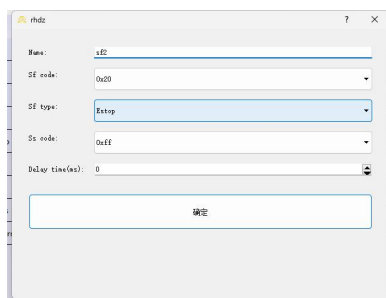


SF 为安全功能逻辑块，所有逻辑的终点都会指向 SF，所以 SF 逻辑块仅有输入点，没有输出点，SF 的输入为触发这个安全功能所需要的条件，如图



当 DI1 输入为低电平时触发安全功能 sf2

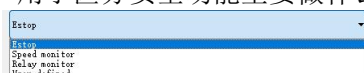
右键设置菜单和双击 SF 块可以对 SF 进行设置



【Name】输入框允许用户对安全功能进行命名，此名称会显示在块的上方。

【Sf code】可以选择安全功能的编码，每个安全功能的编码都是唯一的，目前仅支持最多 19 个用户定义的安全功能。

【Sf type】可以选择安全功能的类型，用于区分安全功能主要做什么功能，目前提供四种



【Ss code】为输出模式，输出模式在【工程配置】界面的【设备输出配置】内进行设置，设置方式见 3.5.1 章节，当安全功能触发时会根据配置的输出模式进行输出控制。

【Delay time】当安全功能类型不为 Relay monitor 时，安全功能的触发条件成立后延迟多久进入安全状态并执行输出模式

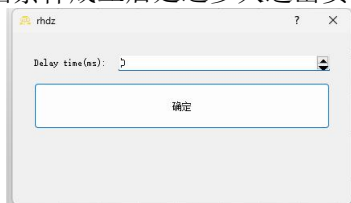
【Option time】当安全功能类型为 Relay monitor 时，延迟时间变为此项设置，并且最小值为 10ms



EXIT 为退出安全功能块，当安全功能的条件成立时 SPC100 会进入安全状态，但当条件不成立的时候，会退出安全状态回到正常状态，但如何退出安全状态则在 EXIT 内设置。

同样的 EXIT 也仅有输入没有输出，输入为退出安全状态的条件，当所有安全功能都没有触发且退出条件成立后 SPC100 设备会回到正常状态。

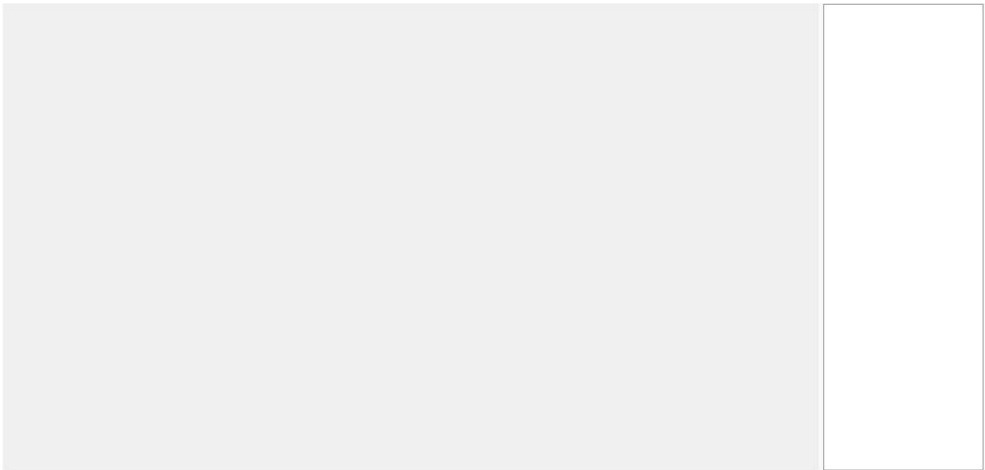
EXIT 同样拥有设置界面，可以设置退出条件成立后延迟多久退出安全状态回到正常状态



EXIT 功能块仅允许存在一个实体，当【逻辑组态】内已经有 EXIT 块，则右侧框体内的逻辑块内的 EXIT 逻辑将变为灰色不可使用状态。

当 EXIT 未使用时，生成代码是会默认 EXIT 条件恒成立且延迟时间为 0ms。

2.5.4 高级编程



高级编程提供给有编程基础的用户自定义线程的功能，用户可以编辑最多五个线程，每个线程最对可输入1024 个字符，右边框体内支持右键菜单，可以新增和删除线程，左侧框体内可以编辑线程内容



点击【新增】，会在当前列表内增加一个线程，双击列表可以编辑线程名称



线程有重名检测，不允许拥有相同名称的线程
线程编辑方式请参考文档可参考《SPC100 API 使用说明》和《SPC100 安全功能编写指导书》。

2.5.5 报告

【项目信息】

包含工程配置界面的工程设置信息

项目信息	
项目名称:	待设置
作者:	待设置
公司:	待设置
项目版本标识:	待设置
项目保存路径:	项目未保存
高级编程:	已启用

【输入资源】

对应【工程配置】界面【设备输入资源】的配置以及【图形化编程】的使用情况
如果在【工程配置】界面选中了资源，则会显示已配置和其别名，如果在【图形化编程】的【逻辑组态】中使用了资源，则显示已使用。

输入资源分配			
DI1	已配置	已使用	ESTOP
DI2	已配置	未使用	LIDAR
DI3	未配置	未使用	

【输出资源】

输出资源分配(1/10)							
序号	Code	Relay1	Relay2	MOS1	MOS2	MOS3	MOS4
1/10	0x20	not relevant	not relevant	not relevant	not relevant	not relevant	not relevant

对应【工程配置】界面【设备输出配置】

【线程使用】

线程使用(1/5)	
test	线程资源已使用: 0.00%

显示已有线程的字符使用量

【安全功能】

安全功能(1/19)						
SF2	sf2	0x20	Relay_monitor	0xff	0ms	10ms

统计【逻辑组态】内用户设置的安全功能及安全功能的配置情况

2.5.6 用户代码

生成

```
---
Estop = 0
Speed_monitor = 1
Relay_monitor = 2
User_defined = 3
not_relevant = 0
relevant = 1

set_lus_version("")

function SF2_func() return ( tdi() == false ) end
function lus_delay_ms(delay_time)
  local start_time = sys_tick()
  while true do
    if ((start_time + delay_time) < sys_tick()) or (get_module_state() == 3) then
      break
    end
    coroutine.yield()
  end
end

test = coroutine.create(function()
  while true do
    coroutine.yield()
  end
end)

function main()
  set_sys(0x20, not_relevant, not_relevant, not_relevant, not_relevant, not_relevant, not_relevant)
  while true do
    if('sf2', 0x20, SF2_func(), Relay_monitor, 0xff, 0, 10)
      exit_sys(true, 0)
      local success, errorMessage = coroutine.resume(test)
      if not success then
        sf('test_error', 0xff, true, Estop, 0xff, 0, nil)
      end
    end
  end
end
```

当【逻辑组态】内用户设置的所有块都处于正常状态时(无逻辑错误)，点击【生成】按钮即可生成用户代码，此界面的开放是为了方便用户调试。

3. 搭建工程示例

本章节将以示例的方式，讲解如何快速使用配置工具对 SPC100 设备进行配置。

❖ 打开通讯端口

将 SPC100 上电，并使用数据线正确连接到电脑端，然后选择正确的端口并点击【打开串口】按钮打开串口

❖ 授权

点击【用户授权】按钮，输入授权密码和设备密码(初始密码均为 rhdzkj8888)

❖ 修改设备密码

对设备没有修改密码需求的用户可忽略此项，出于安全考虑建议用户对新设备修改密码为用户自定义密码，修改密码选项按钮如下：



❖ 新建工程

点击【新建工程】选项，【新建工程】按钮如下：



当新工程构建完成后会跳转到【逻辑编写】的【工程配置】页面。

❖ 配置工程

依次填写完成【工程设置】页面的工程信息：

工程设置

项目名称:

test

作者:

scholar su

公司:

若慧

版本标识:

v1.1.0

项目保存路径:

☒

启用高级编程(自定义线程)

这里为了举例勾选了【启用高级编程】，对于大多数用户而言基本用不到高级编程

❖ 设备输入配置

在 DI 资源中选用了 DI1、DI3、DI4、DI5，分别命名为 ESTOP、RESET、stop1、stop2

DI

☒

DI1

ESTOP

☐

DI2

☒

DI3

RESET

☒

DI4

stop1

☒

DI5

stop2

在 AI 资源中选用 AI1，并命名为 Current,但需要注意的是，对于新工程而言，安全参数也是默认参数，AI1 和 AI2 是默认复用为 DI 的，所以要使用 AI1，需要在【安全参数】页面将 AI1 复用为 AI 使用：

工程配置 安全参数 图形化编程 高级编程 报告 用户代码

CAN

Baudrate 1000000

ASlave nodeID 81

ESlave nodeID 82

Master nodeID 1

CheckBeaconTimeout 0

SendBeaconInterval 1000

PdoPeriodicInterval 1000

SAMPLE INTERVAL

SAI 10

SPI 10

PI_QEP 10

SQEP 10

FAULT_CODE2SS

STATUS DI5ENABL

DELAY_TIME 0

CROSSCHECK DEVIATION

SAI Full 1

SAI Actual 3

SPI Full 1

SPI Actual 3

PI_QEP Full 1

PI_QEP Actual 3

SQEP Full 1

SQEP Actual 3

Cross CheckTime 1

SS LEVEL

DI1 CAT3

DI2 CAT3

DI3 CAT3

DI4 CAT3

DI5 CAT3

DI6 CAT3

DI7 CAT3

DI8 CAT3

RELAY1 CAT3

RELAY2 CAT3

AI1 CAT3

AI2 CAT3

DEFAULT SS

RELAY1 RELEVANT

RELAY2 RELEVANT

SMOS1 RELEVANT

SMOS2 RELEVANT

SMOS3 RELEVANT

SMOS4 RELEVANT

AF

PI1 QEP

PI2 QEP

AI1 AI

AI2 DI

AI

☒

AI1

Current

☐

AI2

❖ 设备输出配置

系统默认的输出模式为 0xff，默认都是 relevant

Code	Relay1	Relay2	MOS1	MOS2	MOS3	MOS4
0xff	relevant	relevant	relevant	relevant	relevant	relevant

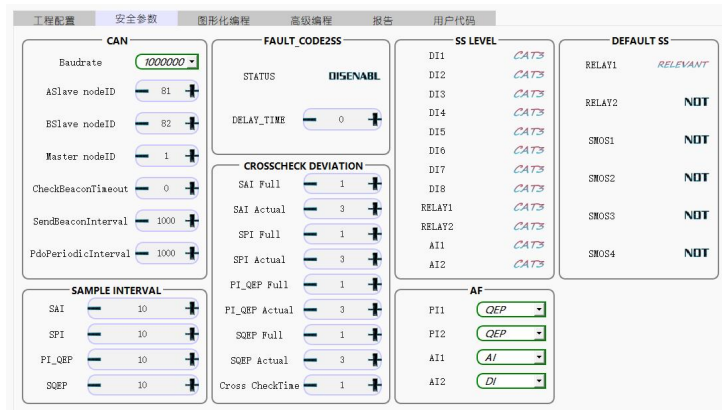
修改为仅 relay1 相关

Code	Relay1	Relay2	MOS1	MOS2	MOS3	MOS4
0xff	relevant	not_relevant	not_relevant	not_relevant	not_relevant	not_relevant

新增一条输出模式 0x20，设置为仅 relay2 相关

Code	Relay1	Relay2	MOS1	MOS2	MOS3	MOS4
0xff	relevant ▼	not_relevant ▼	not_relevant ▼	not_relevant ▼	not_relevant ▼	not_relevant ▼
0x20	not_relevant ▼	relevant ▼	not_relevant ▼	not_relevant ▼	not_relevant ▼	not_relevant ▼

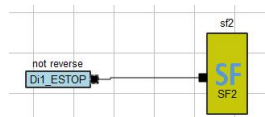
❖ 设置安全参数



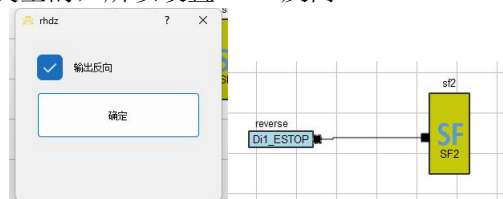
安全参数可根据项目需求自行设置

❖ 图形化编程

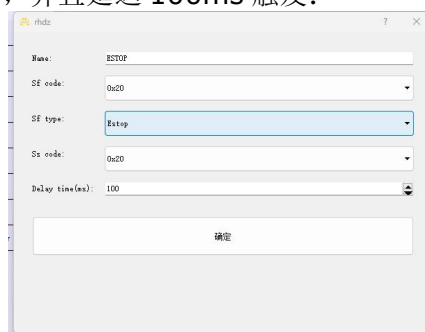
先来设置第一个安全功能，第一个安全功能设为急停(DI1)触发，先拖出一个 DI1 模块，再拖出一个 sf 模块，然后连接两个模块：



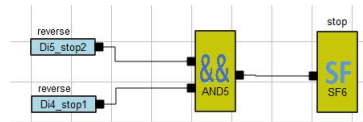
由于急停开关选用的是 NC 类型的，所以设置 DI1 反向

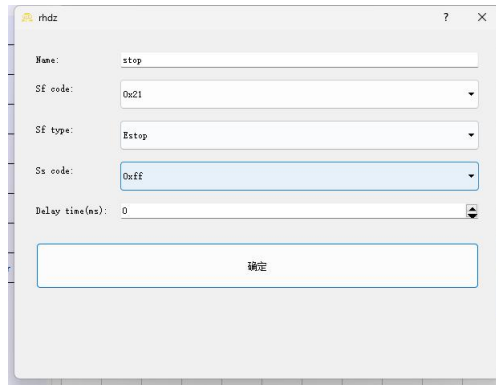


设置安全功能的输出模式为 0x20，并且延迟 100ms 触发：

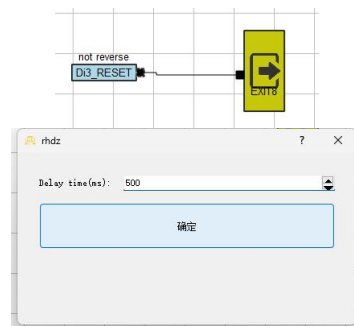


然后按照这种方式设置第二个安全功能，输出模式为 0xff:





最后设置 **EXIT**，退出安全状态的功能块，当 SPC100 设备触发了任一安全功能后进入安全状态如按下急停，当安全功能不再处于触发状态如松开急停，那么当 **EXIT** 的条件成立后才能退出安全状态回到正常状态：



如上图设置，当 DI3 输入高电平后延迟 500ms 解除安全状态。

❖ 高级编程

高级编程主要面对有特殊需求的有编程基础的客户，但在本示例中会写一个简单程序。

```
if get_module_state() == 3 then
    smos3(1)
    smos4(1)
else
    smos3(0)
    smos4(0)
end
```

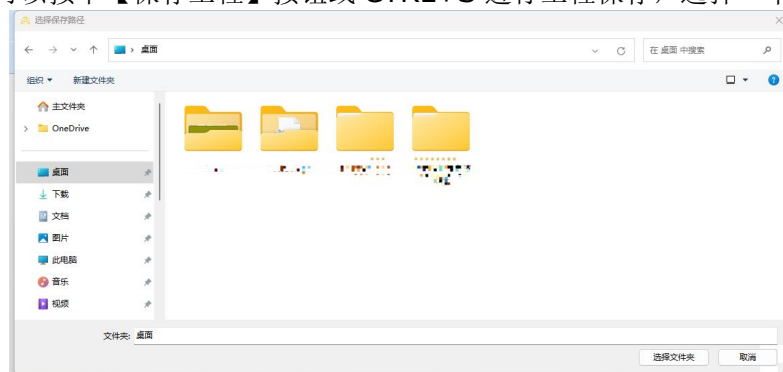
代码内容为 SPC100 设备处于安全状态时，就控制 MOS3 和 MOS4 闭合，否则就断开

❖ 报告和代码

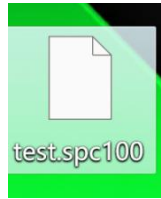
可以在【报告】界面查看整个项目的报告，在【用户代码】界面生成代码，此代码仅用于和图形化编程进行对照参考，具体使用查看 [2.5.5 报告](#)和 [2.5.6 用户代码](#)

❖ 工程保存

在工程配置完成后可以按下【保存工程】按钮或 CTRL+S 进行工程保存，选择一个路径进行存储：



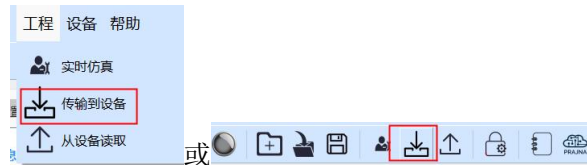
文件名称与项目名称同名，文件后缀为 spc100:



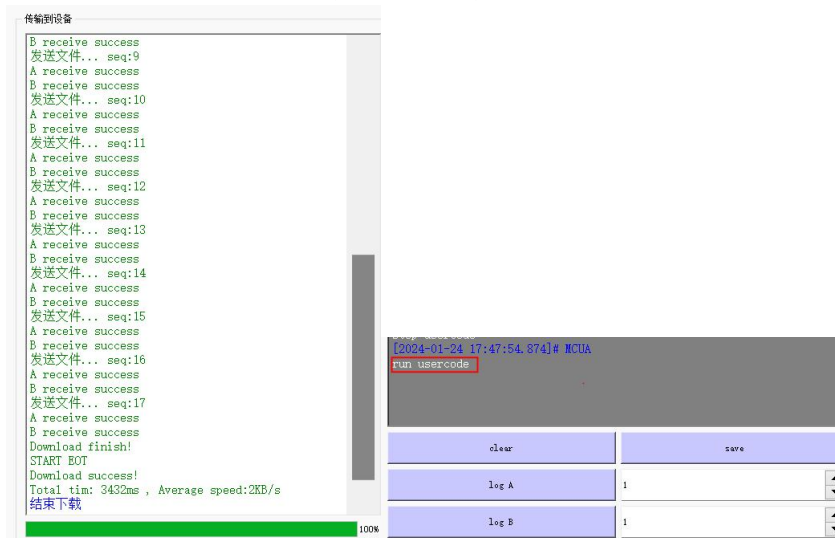
也可以选择不保存到本地，工程传输到设备后也会在设备内进行存储，也可以从设备内读取工程。

❖ 传输到设备

当我们上述功能都配置完成后将工程传输到设备内去运行：



传输完成后出现“run usercode”字样就代表着用户代码已经在运行中了



❖ 仿真

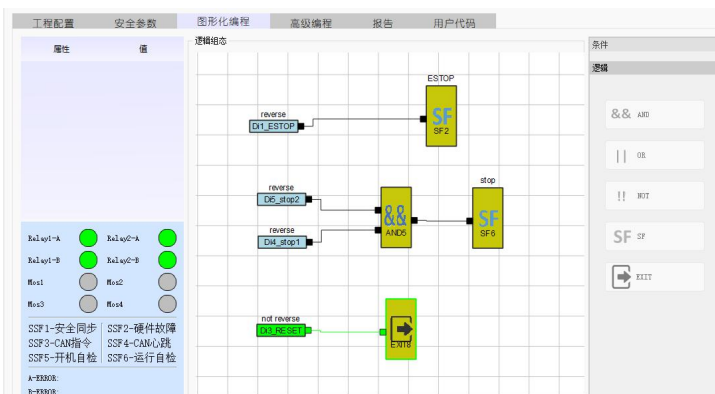
当我们的工程传输到设备后，可以通过仿真功能进行调试我们的安全功能



当仿真开始后，仿真按钮变为蓝色，代表在仿真中，此时部分按钮变为不可用状态，且图形化编程不再可编辑：

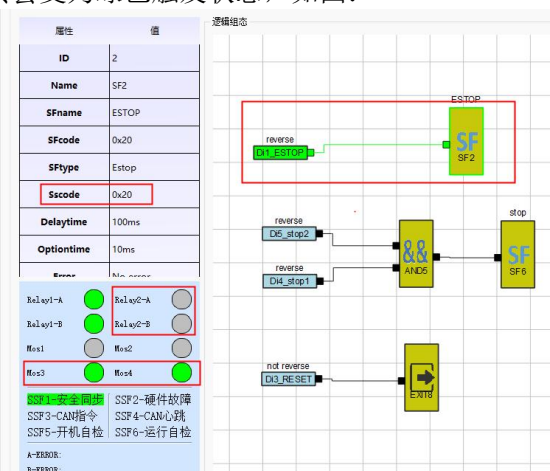


仿真界面：



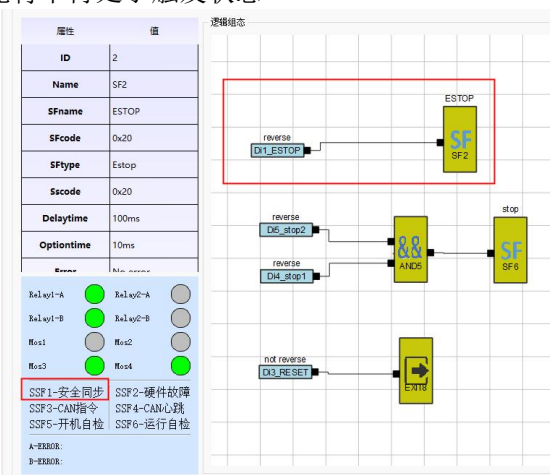
左下角为 RELAY 和 MOS 的输出状态、MCUA/MCUB 的错误状态(错误码)、系统内的系统安全功能(非用户设置)的触发状态。

当一个模块的条件成立时，其会变为绿色触发状态，如图：



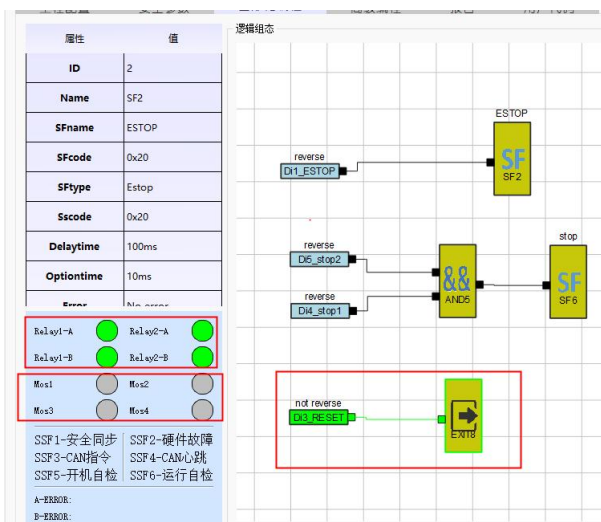
当我们急(DI1)按下后，ESTOP 的安全功能就被触发了，其输出模式为 0x20(只有 relay2 相关)，所以可以在左下角看到 REALY2-A 和 REALY2-B 输出断开，且高级编程内设定的安全状态下 MOS3 和 MOS4 闭合，所以可以在左下角看到 MOS3 和 MOS4 输出闭合。

当我们松开急停后，安全功能将不再处于触发状态

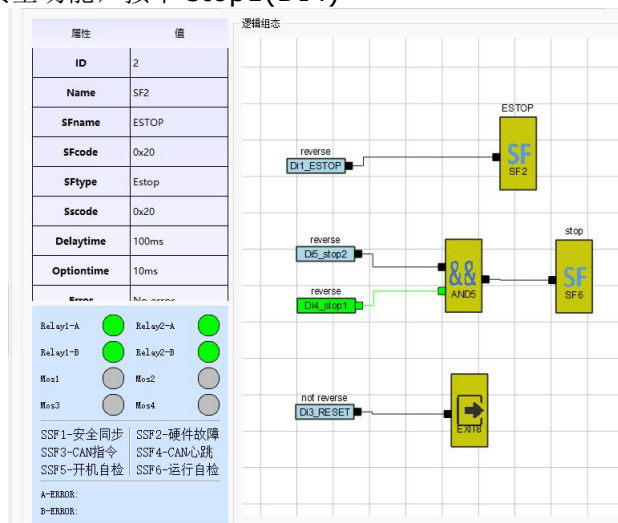


但此时还处于安全状态，因为 EXIT 的条件不成立

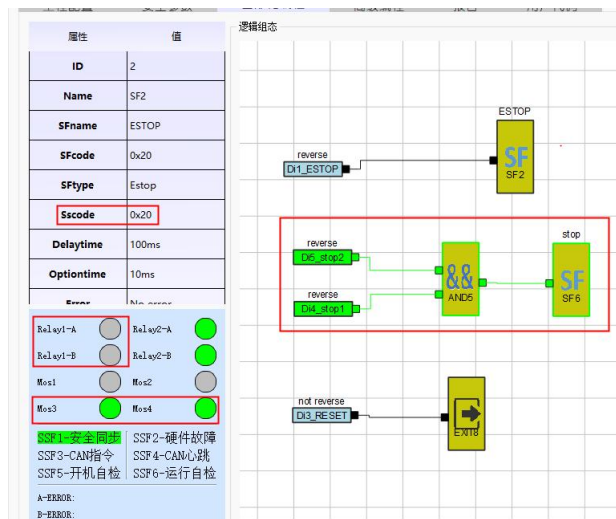
按下 RESET 按钮(DI3),退出条件成立后，延迟 100ms 退出安全状态，relay 和 mos 的输出回到触发前的状态：



那么再测试一下 stop 安全功能，按下 stop1(DI4)

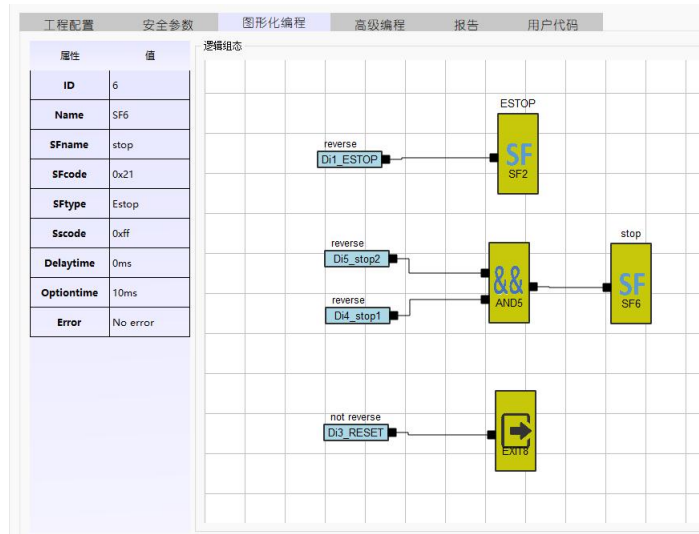


可以看到由于需要 stop1 按下并且 stop2 也按下条件才会成立，而只按下 stop1 是不会触发安全功能的，那么再按下 stop2(DI5)

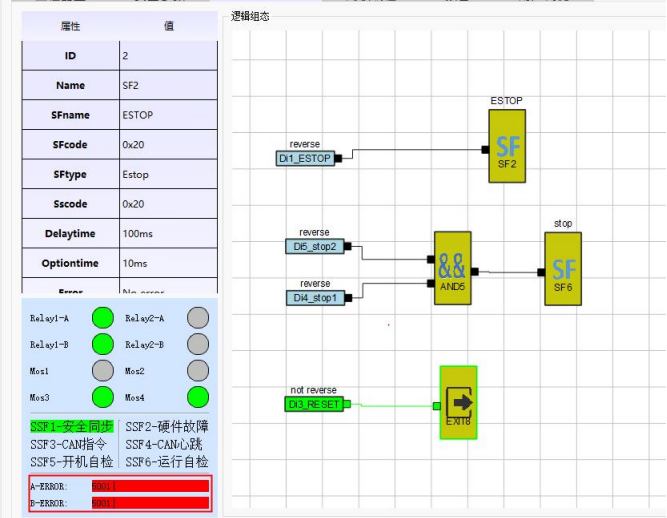


可以看到条件成立，安全功能被触发，因为 stop 安全功能配置的输出模式为 0xff(只有 realy1 相关)，所以 realy1 被断开，MOS3 和 MOS4 如高级编程内所设置被闭合。

再次点击仿真按钮将会退出仿真，此时【逻辑组态】界面恢复为可编辑状态，且左下角仿真弹窗消失。



值得注意的是，SPC100 设备上有两个 MCU 分别进行逻辑决策，所以仿真实际上是对其中一个 MCU 的仿真，如上述所述仿真，如果连接的是 MCUA。此时若 MCUB 的 DI1 触发而 MCUA 的 DI1 未触发，仿真页面内 DI1_ESTOP 的块是不会变为绿色的，但在仿真页面的左下角 ERROR 处会上报 DI1 不一致的错误码。



由于两个 MCU 之间的安全功能是会同步的，所以两个 MCU 依然会进入相同的安全状态