



WasomePLC IDE 使用手册

版本： 1.1.2



版权© 2023 北京万昇智能科技有限公司，保留所有的权利

business@wansheng-intelli.com

未获得本公司书面授权，任何单位及个人不得以任何方式或理由对文档任何内容进行使用、复制、修改、传播与其他产品捆绑使用、销售。

目录

| | | |
|---|----------------------------|----|
| 1 | 前言 | 3 |
| 2 | 安装 WasomePLC IDE 开发环境 | 4 |
| | 2.1 Windows 系统安装 WasomeIDE | 5 |
| | 2.2 Ubuntu 系统安装 WasomeIDE | 7 |
| | 2.3 外部模块导入 | 8 |
| 3 | PLC 目标设备环境准备 | 9 |
| 4 | Wasome Edge 边缘平台管理系统 | 11 |
| | 4.1 系统管理 | 12 |
| | 4.1.1 系统与网络的设置 | 12 |
| | 4.1.2 应用管理 | 13 |
| | 4.1.3 告警与事件管理 | 15 |
| | 4.2 WA-EDGE 工业物联网 | 16 |
| | 4.2.1 插件管理 | 16 |
| | 4.2.2 物联网(IoT)连接设备配置 | 16 |
| | 4.2.3 物联网(IoT)设备连接状态 | 17 |
| | 4.3 WasomePLC 工业自动化 | 18 |
| | 4.3.1 PLC 组网信息 | 18 |
| | 4.3.2 PLC 实例管理 | 19 |
| | 4.3.3 总线配置 - Modbus | 23 |
| | 4.3.4 总线配置 - EtherCAT | 24 |
| 5 | WasomeIDE 项目管理与操作 | 30 |
| | 5.1 项目组成结构 | 30 |
| | 5.2 进入项目管理 | 30 |
| | 5.3 创建项目 | 31 |
| | 5.4 修改当前项目属性 | 32 |
| 6 | 模块操作 | 33 |
| | 6.1 创建内部模块 | 33 |
| | 6.2 导出内部模块 | 34 |
| | 6.3 模块接口设置 | 34 |
| | 6.4 生成模块框架代码 | 36 |
| | 6.5 外部模块引用与解除引用 | 37 |
| 7 | 连接目标 PLC 设备 | 37 |
| | 7.1 连接目标设备 | 37 |
| | 7.2 操作目标 PLC | 38 |

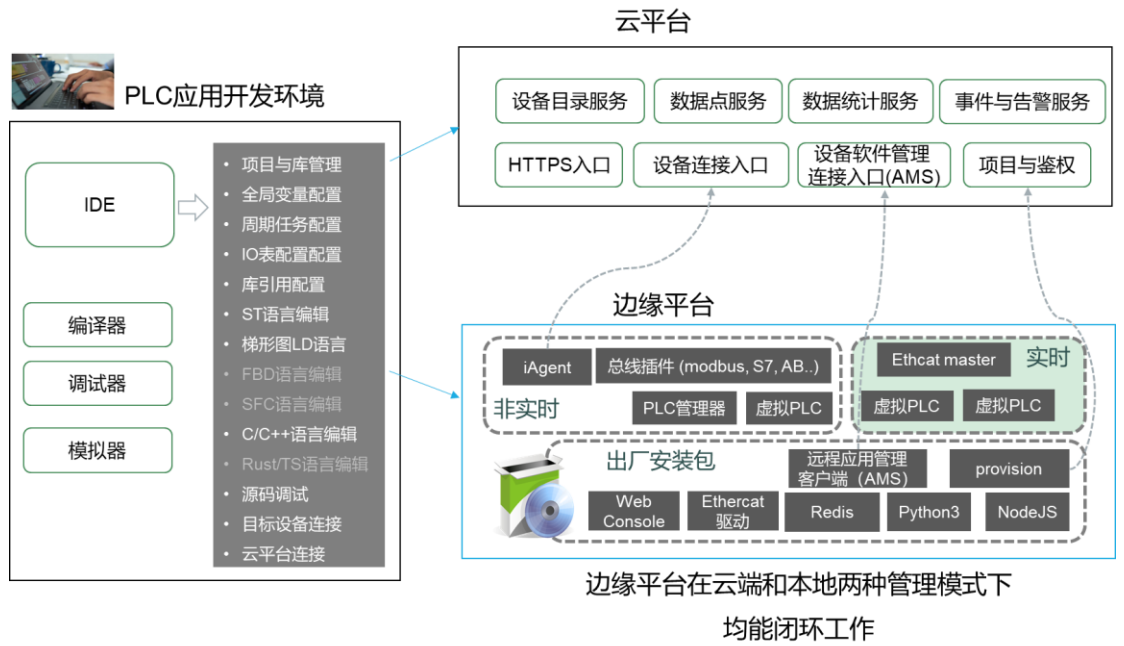
| | | |
|----|---------------------|----|
| 8 | 项目配置..... | 38 |
| | 8.1 全局变量配置..... | 38 |
| | 8.2 IO 组网与通道配置..... | 39 |
| | 8.3 任务配置..... | 45 |
| | 8.4 运动配置..... | 45 |
| 9 | 代码编写..... | 47 |
| | 9.1 梯形图开发..... | 47 |
| | 9.2 ST 代码开发..... | 51 |
| | 9.3 C/C++代码开发..... | 52 |
| 10 | 编译..... | 52 |
| 11 | 调试 PLC 应用..... | 53 |
| | 11.1 启动调试..... | 53 |
| | 11.2 进入调试模式..... | 54 |
| | 11.3 变量调试与强制..... | 55 |
| | 11.4 运行统计..... | 56 |
| 12 | 部署 PLC APP..... | 58 |
| 13 | PLC 云平台..... | 58 |

1 前言

WasomePLC IDE 是一套现代的 PLC 编程开发 IDE，支持 IEC61131-3 编程语言，同时支持 C、C++、Rust*等多种计算机语言混合编程与调试。WasomePLC IDE 基于最新的编程工具 VSCODE 开发，提供功能丰富的编辑、编译、调试、模拟运行等功能。WasomePLC IDE 目前支持在 Ubuntu 系统(18.04 以上版本)上运行。

本文档的目标为帮助使用者了解和掌握 WasomePLC IDE 系统的一般操作使用方法。关于如何开发 PLC 应用请参考《WasomePLC 编程指南》，或者参考 WasomePLC 编程教程开源项目：<https://gitee.com/wasome/tutorials>。

WasomePLC 系统的组成结构如下图所示：



其中 PLC 应用开发环境包含以下组成部分：

编程语言编辑：支持 IEC61131-3 规定的语言如 ST、梯形图等，以及计算机编程语言 C、C++、Rust*等。编辑器提供代码提示、语法高亮、代码块补齐、错误提示等等功能。

编译器 WasomePLC IDE 包含一个功能齐全的编译器与相关工具链，支持 C、C++、Rust、ST 和 LD 等语言的混合编译与链接，对各个编程语言的程序文件能自动构建，输出 PLC 应用程序安装包。

调试器 WasomePLC IDE 自带调试器是基于 IP 网络的调试器，可以同时调试各种不同的语言，既支持在物理机上调试、也支持在开发机上 PLC 模拟器中调试。调试功能支持设置程序断点，单步调试，查看程序的变量以及表达式。

PLC 模拟器 模拟器提供 PLC 运行需要的引擎以及相关 modbus 协议设备模拟，可以在本地调试的程序逻辑

2 安装 WasomePLC IDE 开发环境

WasomePLC IDE 目前支持以下操作系统：

- Ubuntu 18.04 以上版本系统
- Windows10 以上版本系统

2.1 Windows 系统安装 WasomeIDE

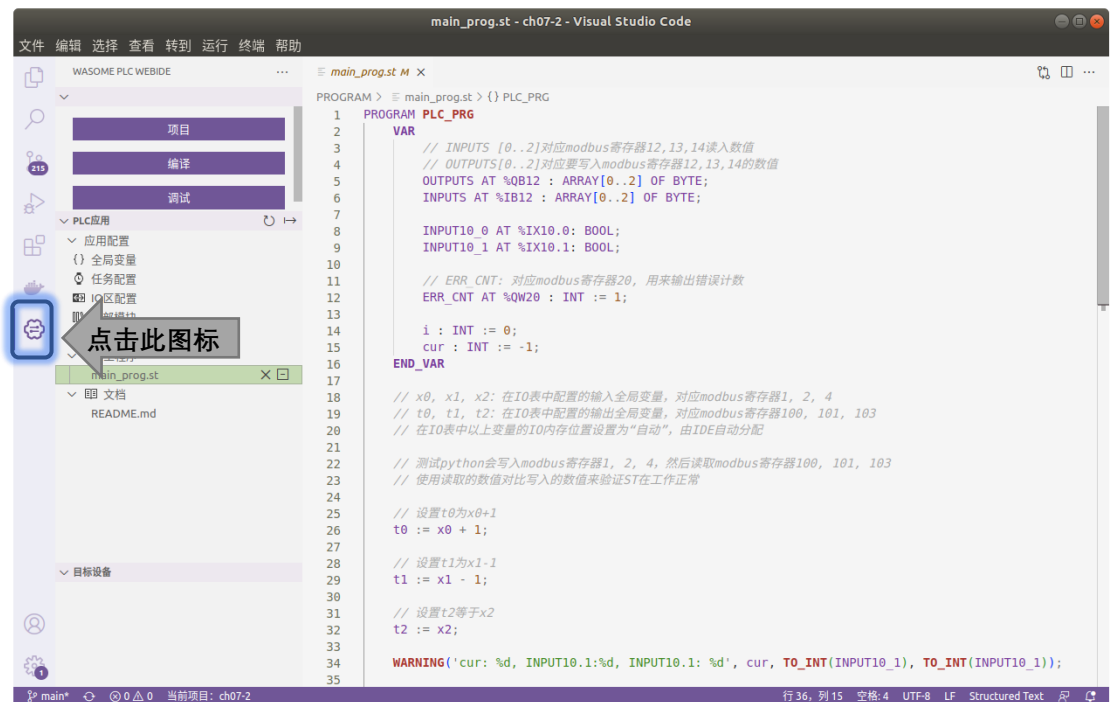
安装 WasomeIDE

执行 WasomeIDE 的安装包文件如 WasomeIDE_Setup_1008.exe, 在安装向导的指引下完成安装。安装完成后, 可以在 Windows 的开始菜单, 看到新添加的“WasomePLC IDE”的程序图标。

在文件浏览器中任意文件夹上, 鼠标右键菜单将会看到菜单项“WasomeIDE”, 点击即可对该文件下的项目打开 IDE 进行开发。



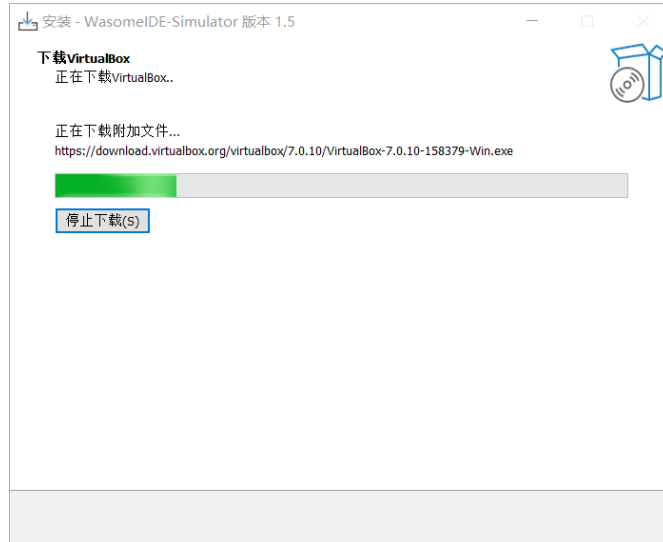
启动后 WasomeIDE 后, 将打开 IDE 窗口如下图, 点击左侧的 IDE 图标, 进入 WasomeIDE 的主视图:



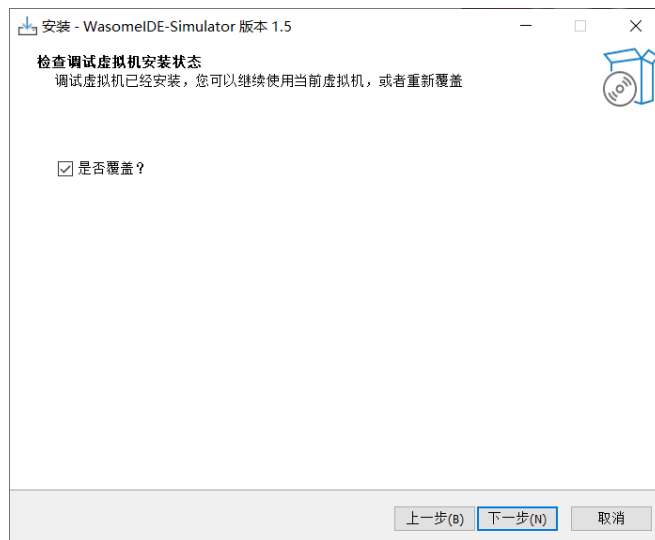
安装 WasomePLC 模拟器

运行安装程序 WasomeSimulator_Setup.exe, 首先会检测开发机是否安装有 VirtualBox 系统, 如果没有将自动下载和安装。

注: 用户也可以从浏览器搜索 VirtualBox 手工下载和安装。



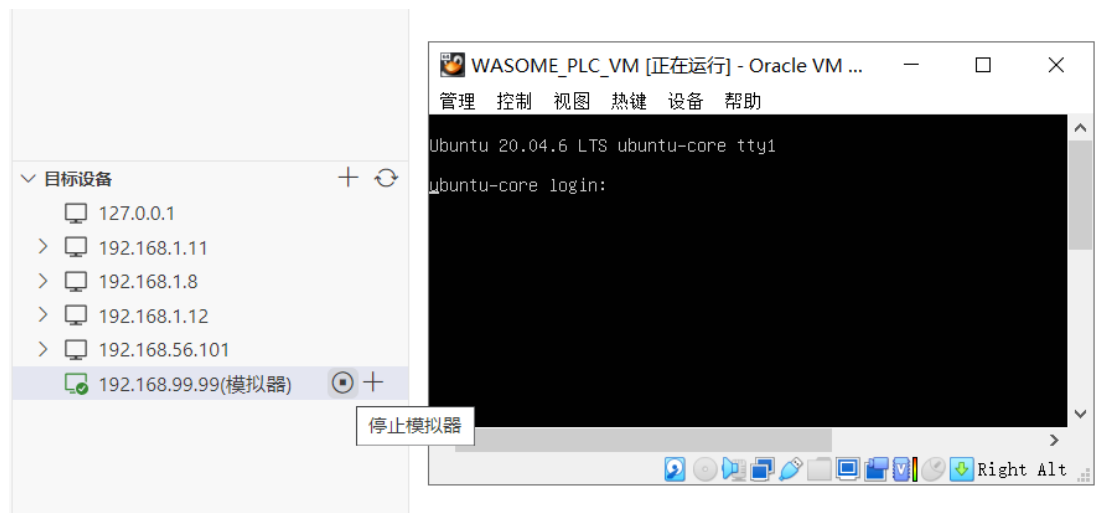
安装 VirtualBox 完毕后, 将检查本机是否已经安装过模拟器, 如果是则提示用户是否覆盖, 建议选址“是”。



安装模拟完成后, 再打开 WasomeIDE, 在目标设备连接视图(如下)中, 将会出现模拟器目标设备, 鼠标选择后, 可以选择按钮“启动模拟器”。



启动模拟器后，将会运行模拟器对应的虚拟机，如下图所示。用户可以使用用户名 softplc，密码 plc@123 登入。同时在 IDE 中模拟器对应的按钮状态展示为“停止模拟器”。



2.2 Ubuntu 系统安装 WasomeIDE

使用 Ubuntu 安装开发环境请注意:

- 强烈建议使用非 root 的普通用户登录安装 WasomePLC IDE!
- 如果当前 docker 是用 snap 安装的话，应该先执行命令 `sudo apt remove docker` 删除 docker 安装，WasomePLC IDE 安装过程会使用 apt 安装合适的 docker 版本。

WasomePLC IDE 的安装包名称一般为 WebIDE.zip。用户将获取的安装包保存在某个目录，进入该目录后执行以下步骤安装 WasomePLC IDE:

```
unzip WebIDE.zip
cd rel/
./install.sh
```

[注]: WasomePLC IDE 需要使用到 Docker, 所以安装命令 ./install.sh 会自动检测系统是否已经安装 Docker, 如果没有就会静默安装 Docker, 同时将创建 docker 用户组, 将当前用户加入到 docker 组中。

启动 WasomePLC IDE

通过以下命令启动 WasomePLC IDE

```
code [project-path]
```

模拟器安装

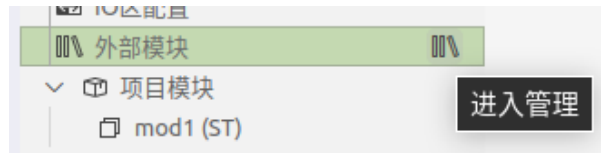
目前不提供 Ubuntu 环境下的模拟器自动安装程序, 用户可以手工下载 VirtualBox 和导入模拟器镜像文件。

用户也可以直接在本机 (Ubuntu) 安装 PLC 设备软件系统, 使用本机作为测试目标环境。

2.3 外部模块导入

WasomePLC IDE 使用模块方式提供库的管理, 全局模块可供任何项目引入使用。WasomePLC IDE 安装包含若干系统模块, 同时支持用户导入更多的外部 PLC 编程模块到系统模块库。

鼠标停留在项目资源区“外部模块”上节点时, 会有提示“进入管理”按钮如下:



点击按钮进入管理即可进入“模块管理”界面, 点击“安装模块”, 选择模块文件例如 module-name.zip 即可导入模块。

模块管理

| | | | | | | | 安装模块 | 添加到当前项目 |
|---|-------|----|--------|--------------------------------|------------------------------|--|------|---------|
| 名称 | 版本 | 驱动 | 类别 | 供应商 | 描述 | | | |
| <input type="checkbox"/> DS402 | 1.0.0 | 是 | source | WaSome Intelligence Technology | Servo driver for DS402 | | | |
| <input type="checkbox"/> plcopen_mc_par | 1.0.0 | 否 | source | WaSome Intelligence Technology | PLCOPEN MOTION CONTROL PART1 | | | |

在“模块管理”界面，点击模块列表中的模块名称，即可进入所导入模块的详细界面，包括开放语言、版本、数据类型列表、POU 列表和依赖的模块等等。

3 PLC 目标设备环境准备

本章介绍如何在一个纯净的 Linux 平台上安装 PLC 目标设备软件组件。用户应该准备好一个 Debian 系统的 Linux 系统，如 Ubuntu18.04 或者 Debian9 以上。

准备软件安装包

出厂安装包(Factory Install Package)

每个安装包基于不同的 CPU 架构和 Linux 的发行版本各有不同的版本。出厂安装包的名称中包含适用的处理器型号、linux 发行版和版本等信息，如“wa-setup-amd64-ubuntu2021-01-24.sh”。目前已经测试验证过如下设备型号：

- INTEL NUC
- INTEL 工业网关（研华科技、深圳得航）
- ARM 工业网关（清捷智能）
- 树莓派

功能软件包

用户还应该获得以下的基础软件安装包：

- Wasome IoT Agent (wa-agent)
- Wasome PLC Framework (wa-plc-framework)

执行出厂安装包

- **运行安装脚本：**如下以 root 权限运行安装程序，

```
sudo ./wa-setup-amd64-ubuntu2021-01-24.sh
```

- **输入板子名称：**在出现下面的提示后，选择合适的主板名称，如回车将使用缺省的配置（generic）

```
Enter the board name (blank for generic):
```

```
-----
```

```
generic
```

- **选择应用工业应用管理系统安装路径：**出现下列提示后，输入工业应用管理系统的安装路径。

Where to install software:([Enter] will install to "/ams"):

- **安装 ethercat 驱动：**出现以下提示后，用户可以选择以下选项：b – 从源码开始包编译驱动（需要有联网），x – 结束安装，否则输入任何其他键将继续（不安装 ethercat 驱动）。

```
start to install eth...
The [ethcat_4.19.66-v7+.tar.gz] not present. Please choose what to go?
>> (b) build from source;
>> (x) Exit;
>> Any other key to continue; >>
```

- **选择系统重启：**在系统安装完成后，出现下列提示后，输入“yes”，重新启动系统。

WA-IoT factory installation is completed successfully!

The system should reboot to take it into effect. Reboot now? yes/[no]:

登录到设备的网页控制台

安装出厂软件包后，从任何一台电脑打开 Web 浏览器，输入目标机的 IP 地址和 3000 端口，将看到 Wasome Edge 管理系统（WebConsole）的首页。登录账户是 admin，缺省密码是 123456。



安装基础软件

进入 WebConsole 的“应用管理”，点击按钮“本地安装”，逐个安装基础软件包。注意每安装一个后，需要刷新网页来确认安装成功，才能开始成功安装下一个。

如下图所示，确认下图中软件 wa-agent 和 wa-plc-framework 已经安装：

| 名字 | 版本 | 软件类别 | 状态 | 安装源 | |
|-------------------------|-------------|-------|---------|-----|----------|
| modbus_device_simulator | 2022-1-5 | 一般软件包 | stopped | 本地 | 启动 停止 删除 |
| plc-app-demo2 | v2020-11-15 | PLC应用 | | 云端 | 启动 停止 删除 |
| wa-agent | 2023-6-7 | 一般软件包 | running | 本地 | 启动 停止 删除 |
| wa-plc-framework | 2023-6-10 | 一般软件包 | running | 本地 | 启动 停止 删除 |

4 Wasome Edge 边缘平台管理系统

现场操作运维人员使用任意电脑浏览器登录目标设备的 Wasome Edge 边缘平台管理系统 (WebConsole)来完成操作与管理。主要的操作内容有三大类：

1. 系统管理

- 系统与网络的设置
- 远程应用配置
- 告警管理
- 事件管理

2. WA-EDGE 工业物联网

- 插件管理
- IoT 设备连接配置
- IoT 设备连接状态

3. WasomePLC 工业自动化

- PLC 组网状态
- PLC 实例管理
- EtherCAT 配置与操作
- Modbus 服务状态

4.1 系统管理

4.1.1 系统与网络的设置

The screenshot displays the 'Wasome Edge 工业边缘平台管理系统' (Wasome Edge Industrial Edge Platform Management System) interface. The top navigation bar includes the logo, system name, '返回首页' (Home), 'WA-EDGE系统管理' (WA-EDGE System Management), and a '帮助' (Help) link.

The main content area is divided into several sections:

- 系统信息 (System Information):** A table showing system details:

| | |
|-------------------------|-------------------------|
| 操作系统: linux | 内核版本: 5.4.0-136-generic |
| CPU架构: x64 | 主机名称: wasomenuc |
| Cgroup v2: 未启用 | |
| PLC根路径: /wa-plc/ | AMS根路径: /ams/ |
| WASOME出厂版本: v2024-01-24 | 授权状态: 未授权 |

 A '获取授权' (Get Authorization) button is present.
- 无线网络 (Wireless Network):** Shows interface 'wlp3s0', connection status 'wifi(连接中)', and SSID 'Xiaomi_wilson'. A '配置' (Configure) button is available.
- 网络配置 (Network Configuration):** A table listing network interfaces:

| 网络接口 | 状态 | mac地址 | IP | 获取地址 |
|---------|-----|-------------------|-------------------|------|
| eno1 | 停止 | 00:1f:c6:9c:5a:07 | | 自动 |
| wlp3s0 | 运行中 | a0:c5:89:14:25:da | 192.168.31.206/24 | 自动 |
| docker0 | | 02:42:55:71:bf:95 | 172.17.0.1/16 | 手动 |
- 串口类型总线 (Serial Port Type Bus):** A table listing serial ports:

| 名称 | 串口 | 波特率 | 校验 | 数据位 | 停止位 | 当前分配 |
|------|------------------|-------|------|-----|-----|--------|
| aaaa | /dev/ttyS7 | 9600 | NONE | 8 | 1 | modbus |
| ddd | /dev/ttyS4 | 19200 | EVEN | 1 | 1 | |
| COM1 | /dev/tty1 (出厂串口) | 9600 | NONE | 8 | 1 | |
- 磁盘空间 (Disk Space):** A table showing disk usage:

| 磁盘 | 总空间 | 已用空间 | 百分比 | 挂载目录 |
|----------------|---------|---------|-----|------|
| udev | 15.60G | 0.00G | | /dev |
| /dev/nvme0n1p1 | 399.22G | 470.00G | | / |

本页面提供目标系统的系统、网络和磁盘的现状查看和配置。如果当前设备尚未授权，可以点击“获取授权”按钮。在下面的对话框中，输入手机号和验证码，获得授权。

The dialog box titled '手机号验证' (Mobile Number Verification) contains the following fields and buttons:

- 手机号码: 请输入手机号 (Mobile Number: Please enter mobile number)
- 验证码: 请输入短信验证码 (Verification Code: Please enter SMS verification code)
- 获取短信验证码 (Get SMS Verification Code) button
- 取消 (Cancel) button
- 确定 (Confirm) button

注：没有授权的设备，目前的 PLC 程序在运行 30 分钟后，会自动重启一次。

操作者还可以使用“系统重启”按钮来完成重启。

系统信息

系统信息中查看目标机的操作系统,内核版本,CPU 架构,主机名等系统信息,

还能查看 WASOME 系统的跟路径以及出场包的版本信息。

无线网络

查看当前 WIFI 网络的连接状态。点击按钮“配置”，可以选择目标 SSID 来连接，或者让主机提供一个热点服务。

网络配置

查看目标机上的网络接口状态，IP 地址。点击“配置”按钮可以配置网口使用 DHCP 或者静态 IP 地址。

修改 IP 地址后，可以点击“重启网络”按钮，然后刷新网页来显示最新状态。

串口类型总线

查看和配置本机的串口总线

磁盘空间

查看当前磁盘的空间使用情况。

4.1.2 应用管理

目标系统可以本地闭环工作，所有的功能不依赖云平台连接。

设备身份认证

如果目标系统需要连接 Wa-Connect 云端平台，首先需要完成身份分配过程。通过下图“设置”按钮输入从云端平台提供的接入密钥，来完成在线身份分配及密钥下发。

设备身份认证

| | |
|-----------------------------|-----------------------------|
| 服务器IP地址: | 端口号: 1804 |
| 设备ID: | 状态: 身份未获取 身份复位 |
| 项目ID: | |
| 身份分配: 设备出厂ID:"777cc3f9fba8" | 密钥名称: 设定 |

工业应用商店

WasomePLC 软件系统支持设备连接云端(或者企业内网)工业应用软件商店，并从服务端配置设备上所安装的软件、版本及其配置数据。

下图中在正确配置服务器的地址和接入密码后，应用商店客户端软件开始连接云端工业应用商店平台，从云端接收远程软件安装和配置管理。

工业应用
商店

| | | |
|----------------------|--------|-----------|
| 服务器地址: | 端口号: | 配置 |
| 客户端状态: 未注册 | ID: | 立即同步 |
| 客户端当前版本: v2023-03-08 | 更新客户端: | 选取文件 安装更新 |
| 云端下载状态: 当前无软件下载 | | |

界面中可查看工业应用商店客户端的当前版本, 以及加载和更新客户端的版本。如果当前正在下载工业应用软件, 可从“云端下载状态”中显示当前下载的软件名称与进展。

软件管理

通过 WebConsole 来管理目标系统上的软件, 是核心的功能。一个软件属于以下类别:

| | |
|--------|--|
| 一般软件包 | 自动开机运行, 可以手动管理启动和停止运行 |
| 容器镜像 | PLC 程序引擎 (Runtime)可以基于 docker 镜像的压缩包来分发和运行 |
| PLC 引擎 | 用于 PLC 程序引擎(Runtime)的分发和运行, 比 docker 更轻量 |
| PLC 应用 | 依赖 runtime_engine 来运行的软件类型, 用于 PLC 的控制应用程序(App)分发与安装 |

网关软件安装列表

点击界面上的“本地安装”, 选择本机上的软件安装包, 一般以 tar.gz 为后缀。该软件包将上载到目标系统并完成安装。刷新网页查看新安装的软件。

如下图所示, 用户可以查看当前系统上安装的所有软件。其中 basic runtime 是我们会常用到的一个 PLC 程序运行引擎, Modbus simulator 经常用于测试。另外, plc-runtime-openvino 是一个 docker 类型的 PLC runtime。

网关软件
安装列表

| 名字 | 版本 | 软件类别 | 状态 | 安装源 | 本地安装 |
|----------------------|-------------|-------|-----|-----|-------------|
| plc-app-demo2 | v2020-11-15 | PLC应用 | | 云端 | 配置 启动 停止 删除 |
| modbus_test_device | v2023-06-25 | 一般软件包 | 运行中 | 本地 | 配置 启动 停止 删除 |
| wa-agent | 2023-6-29 | 一般软件包 | 运行中 | 本地 | 配置 启动 停止 删除 |
| wa-plc-framework | 2023-11-15 | 一般软件包 | 运行中 | 本地 | 配置 启动 停止 删除 |
| plc-runtime-openvino | 2022-10-10 | 容器镜像 | | 本地 | 配置 启动 停止 删除 |
| ch07-2 | 0.0.17-11 | PLC应用 | | 本地 | 配置 启动 停止 删除 |
| basic | 2023-9-18v2 | PLC引擎 | | 本地 | 配置 启动 停止 删除 |

4.1.3 告警与事件管理

告警

告警是高效管理的核心能力，查看当前活动中的告警，参考《WasomePLC 告警手册》进行故障排查和处理。恢复的故障将自动清除活动告警。某些情况下，用户可以选择手动清除告警。

左侧的导航栏一些条目中会显示当前的活动告警数目，如果自上次查看出现的新的告警，该数目用红色显示，否则显示灰色。注意：某些时候活动告警数并没有增加，可能是因为新生的告警已经自动被清除。

| ETHERCAT 3 | | MODBUS (循环) 4 | | PLC设备 1 | | |
|---|--|--|--|--|--|--|
| 告警中 8 已清除 459 全部 467 | | 目标类别: <input type="text" value="请选择"/> | 目标ID: <input type="text" value="请输入"/> | 告警ID: <input type="text" value="请输入"/> | 严重级别: <input type="text" value="请选择"/> | 发生时间: <input type="text" value="请选择"/> |
| ● EtherCAT配置已修改, 需要重启EtherCAT服务生效 (23020) | 告警源: ETHERCAT >> | 发生: 2023-01-10 16:27:56 | | <input type="button" value="手动清除"/> | | |
| 状态: 活动中(持续1小时) | Master config changed. Master:0 | | | | | |
| ● 总线不能连接成功(20000) | 告警源: MODBUS(循环-Bus) (127.0.0.1:1502) >> | 发生: 2023-01-10 11:57:26 | | <input type="button" value="手动清除"/> | | |
| 状态: 活动中(持续5小时) | | | | | | |
| ● 总线连接断开(20010) | 告警源: MODBUS(循环-Bus) (127.0.0.1:1502) >> | 发生: 2023-01-10 11:57:26 | | <input type="button" value="手动清除"/> | | |
| 状态: 活动中(持续5小时) | 代理: default, 最近一次错误: Connection reset by peer(104), 连续错误: 1, 累计错误: 1 | | | | | |
| ● EtherCAT从站离线(23010) | 告警源: ETHERCAT (master:0-slave:20_0) >> | 发生: 2023-01-09 20:47:11 | | <input type="button" value="手动清除"/> | | |
| 状态: 活动中(持续20小时) | | | | | | |

事件

事件页面展示了系统中非故障的重要过程情况，帮助理解系统的内部运行情况。

| 全部 50 | | 目标类别: <input type="text" value="请选择"/> | 目标ID: <input type="text" value="请输入"/> | 事件ID: <input type="text" value="请输入"/> | 严重级别: <input type="text" value="请选择"/> | 发生时间: <input type="text" value="请选择"/> |
|---|---|--|--|--|--|--|
| ● PLC pending delete(1003) | 事件源: PLC (fff) >> | PLC pending delete | | 发生: 2023-01-10 17:23:54 | | |
| ● EtherCAT process start(4000) | 事件源: ETHERCAT >> | EtherCAT process start(config path: /wa-plc/ethercat/config) | | 发生: 2023-01-10 15:06:31 | | |
| ● EtherCAT process exit(4001) | 事件源: ETHERCAT >> | EtherCAT process exit | | 发生: 2023-01-10 15:06:31 | | |
| ● In service(5000) | 事件源: MODBUS(循环-Broker) (hhh) >> | In service | | 发生: 2023-01-10 15:04:12 | | |
| ● plc-manager process start(1002) | 事件源: IAGENT | plc-manager process start | | 发生: 2023-01-10 15:04:11 | | |
| ● EtherCAT process start(4000) | 事件源: ETHERCAT >> | EtherCAT process start(config path: /wa-plc/ethercat/config) | | 发生: 2023-01-10 15:04:11 | | |

4.2 WA-EDGE 工业物联网

4.2.1 插件管理

运行状态

| | | | |
|---------|----------|-------------------|-------|
| 当前加载插件: | wa-edge | modbus_ocf_bridge | lwm2m |
| | database | modbus_plugin | |

插件管理

选择插件安装包: 本地安装

插件开发: <https://gitee.com/wasome> 插件下载: <https://gitee.com/wasome> 起效插件配置

本地插件搜索: 插件清单编辑器

| | |
|---|--|
| 名称: modbus_ocf_bridge | 版本: 1.0.0 |
| 提供者: Beijing WanSheng Intelligent Corp | 文件名: libmodbus_server.so |
| 说明: wa-edge built-in modbus plugin, supp... | 网址: |
| 插件提供资源: | |
| /modbus | 类别: 标准设备 ● 说明: Bridge modbus devices to OCF standard |
| /mb/bus | 类别: 通用 ● 说明: Query the status of bus used by modbus |

| | |
|--|---|
| 名称: lwm2m | 版本: 1.0.0 |
| 提供者: Beijing WanSheng Intelligent Corp | 文件名: liblwm2m_server.so |
| 说明: lwm2m plugin | 网址: |
| 插件提供资源: | |
| /lwm2m | 类别: 标准设备 ● 说明: bridge the lwm2m devices to OCF standard |

4.2.2 物联网(IoT)连接设备配置

本页面管理目标设备主要的 IoT 外部设备连接配置，一个设备一般需要配置唯一的设备名和连接地址如：IP 或者总线名，以及从站号等。

modbus (OCF)

WA-IOT-EDGE 平台内部支持对市场上常见的 MODBUS 类型设备转换成 Open Connectivity Foundation (OCF)标准组织所定义的 OCF 设备标准。主要用于边缘平台物联网应用场景。

插件设备(OCF 标准)

每一个插件可以代表一种工业协议或者工业设备。使用插件，用户可以配置的该类型设备的虚拟设备。

配置生效

当修改 modbus 的配置后，上面页面中按钮“重启 WA-AGENT 服务”将会变高亮，同时左侧导航栏入口出现“R”标记（如下），提示需要重启 WA-AGENT 服

务来生效最新的配置修改。

总线使用情况与测试从站

下图中“使用总线”显示了当前打开的总线，以及错误情况以及因为过载所抛弃的请求数目。只有当在一个总线上发生了请求，该总线才会被打开。

使用总线

| 名称 | 类型 | 定义 | 连接 | 总错误数 | 过载 |
|-------|-----|-------------------------------|----|------|----|
| bus-1 | tcp | "ip": 127.0.0.1, "port": 5002 | 是 | 0 | 0 |

测试

总线: bus-1 从站: 1 寄存器: 1 功能码: 条数: 编码: 请选择

URL: /mb/bus-1/1/1 发送数据: GET PUT

返回: [200]0

配置总线后，测试连接到该总线上的从站。如上图所示，选择总线并输入从站 ID 和寄存器、功能码等，点击 GET 和 PUT 按钮进行读写测试。PUT 操作时，“发送数据”编辑框中输入要写入 MODBUS 的数值。

第一次操作结果可能显示“Not connected”，可以再测试一次。如果返回显示为 200 表明操作成功，否则错误原因将被显示。

4.2.3 物联网(IoT)设备连接状态

云端物联网平台接入配置

wa-agent 基础软件提供了丰富的物联网功能，支持和云端物联网平台的连接。如果需要和云端平台连接，需要先完成身份分配。

设备接入服务器

| | |
|---------|---------------------|
| IP地址: | |
| 端口: | |
| 设备ID: | lagent_no_ID |
| 连接状态: | connecting |
| 状态开始时间: | [23-01-10 15:04:10] |

边缘平台当前接入设备状态

支持多种设备连接和数据采集，如 Modbus, LWM2M, OCF, WasomePLC, SEMIENS PLC 等。当前连接的设备显示在下面的列表中：

连接设备

显示离线设备

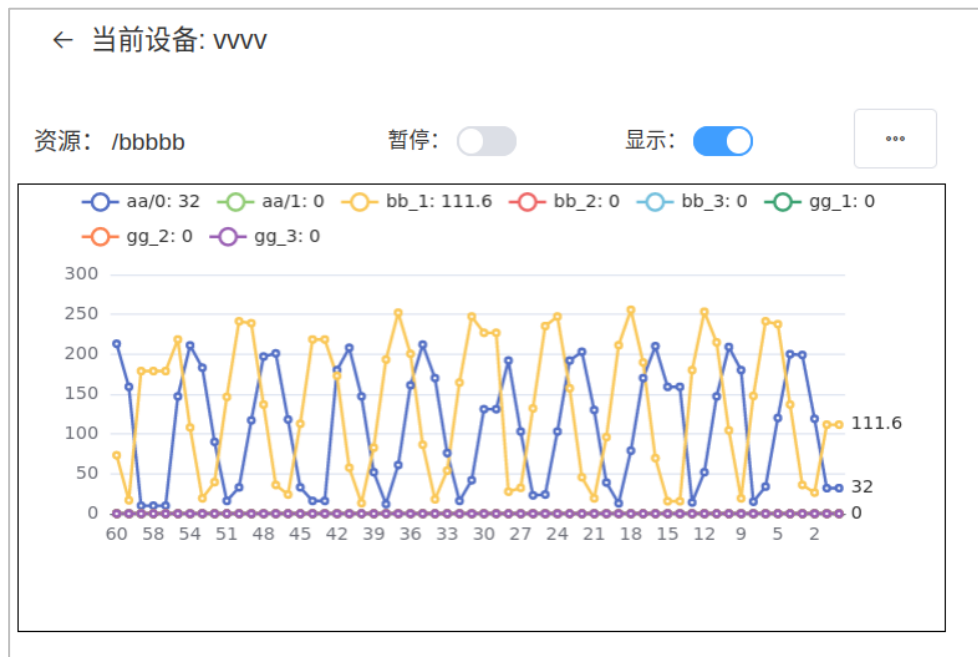
| 名称 | 状态 | 协议 | 进入状态时间 | 设备类型 | 应用配置 | |
|---------------|----|---------|---|--------------|--|----|
| ch702 | on | ep | 2023-04-24 15:16:43 | VPLC | <input type="checkbox"/> 云端 | 进入 |
| abc | on | ep | 2023-04-24 15:16:43 | VPLC | <input type="checkbox"/> 云端 | 进入 |
| lwm2mdaemon-m | on | lwm2m | 2023-04-24 15:16:20 | | <input type="checkbox"/> 云端 | 进入 |
| vvvv | on | modbus | 2023-04-24 15:15:36 | _dt_bus-1_2_ | <input checked="" type="checkbox"/> 本地 | 进入 |
| 资源: /bbbb | | 类型: 自定义 | 采集: <input checked="" type="checkbox"/> | 周期: 1 | | |

如上图所示，每个连接的设备可以在云端配置和本地配置之间进行切换。每个设备可以下拉显示所有包含的资源信息。

如果启用了本地配置，用户可以修改和保存采集配置。

设备数据查看

从每个设备点击“进入”按钮，将离开页面进入当前设备的数据展示与配置界面，如下图所示：



4.3 WasomePLC 工业自动化

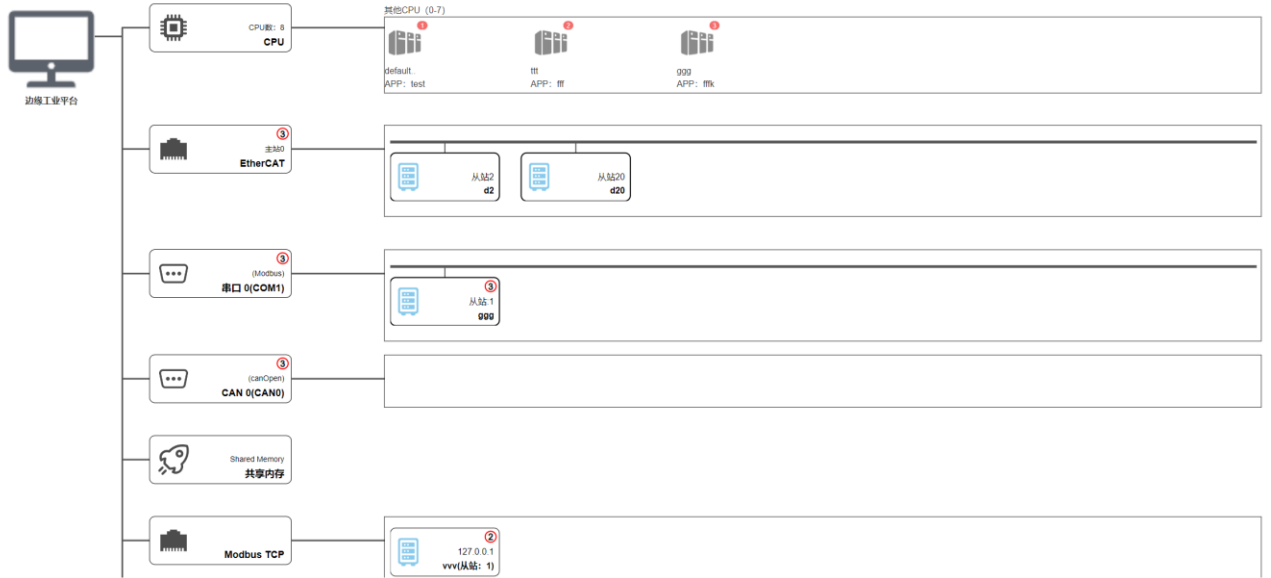
4.3.1 PLC 组网信息

WasomePLC 支持在边缘控制平台创建多个虚拟 PLC，系统为虚拟 PLC 提供如下的支持：

1. CPU 优先级与调度

- 2. 内存资源保证
- 3. 网络接口与外部设备的资源分配
- 4. 容器化与微容器化隔离和调度

在如下系统组网图中，全面展示当前所有的 CPU 资源以及网络接口的分配情况。如果一个网络接口或者连接设备商有一个数字标记，表示该资源已分配给对应的 PLC 实例。



4.3.2 PLC 实例管理

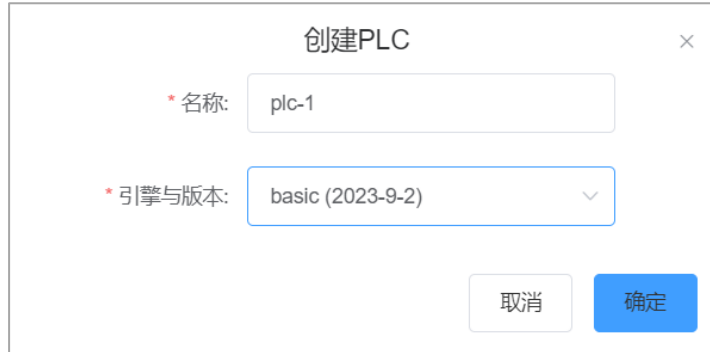
PLC 实例列表

从 PLC 实例管理导航栏，显示目标设备上当前配置的所有 PLC，如下图所示：

| 运行 1 就绪 0 未就绪 2 停止 0 禁用 0 全部 3 | | 标签: 请选择 | 自启动级: 请选择 | 自动升级: 请选择 | 调度策略 | 创建 PLC |
|--------------------------------|---------------------|-------------|-----------|-----------|------|--------|
| remote2 >> | 本地 | 状态: | | | | |
| 引擎: basic (2023-9-16) | 程序: ch07-2 (0.0.17) | 持续: | | | | |
| 设置: 自动启动--开 | 自动升级--关 | 告警: | 2 | | | |
| ggg >> | 本地 未授权 | 状态: Running | | | | |
| 引擎: basic (2023-9-18v2) | 程序: xxx (0.0.2) | 持续: 21分 | | | | |
| 设置: 自动启动--开 | 自动升级--关 | 告警: | 1 | | | |
| plc-1 >> | 本地 | 状态: 未装载 | | | | 装载程序 |
| 引擎: basic (2023-9-18v2) | 程序: () | 持续: | | | | |
| 设置: 自动启动--关 | 自动升级--关 | 告警: | 0 | | | |

创建 PLC

在上面页面中点击按钮“创建 PLC”来创建一个新的 PLC 实例。如下图中，输入唯一的 PLC 名称，然后选择使用的引擎名称和版本，点击确定按钮。



创建 PLC 对话框，包含以下信息：

- 名称: plc-1
- 引擎与版本: basic (2023-9-2)
- 按钮: 取消, 确定

点击确定按钮，可以看到已经创建了新的 PLC，当前的 PLC 处于尚未装载 PLC 程序（APP）的状态。



PLC 实例概览，包含以下信息：

- 名称: plc-1 >>
- 本地: 本地
- 按钮: 装载程序
- 引擎: basic (2023-9-18v2)
- 程序: ()

装载 PLC 程序

WasomePLC 系统支持两种 PLC 程序装载方式：

1. 在开发阶段通过 WasomePLC IDE 装载程序到目标 PLC
2. 在生产环境中，直接通过 WebConsole 来完成装载程序（无需 IDE）

第二种方式在 WebConsole 上选择按钮“装载程序”来完成 PLC APP 的安装：



装载程序对话框，包含以下信息：

- 应用与版本: ch07-2 (0.0.5)
- 按钮: 取消, 确定

选择要装载的 APP 名称与版本后点击确定，进入导入 APP 组网配置界面，系统将检查 APP 所配置的组网资源，是否在目标设备上可用。

PLC组网配置

PLC: `ttt`, 控制程序 (APP): `fffk` 专家设置

APP组网设备: APP项目组网配置: 刷新

| | | |
|---------------------|---|---|
| modbusRTU 从站 | 从站: <code>ggg</code> 总线: COM1, 从站: 1 | 错误: serial bus already assigned to other PLC |
| modbusTCP 从站 | 从站: <code>lll</code> IP: 127.0.0.1: 1502, 从站: 1 | <input checked="" type="checkbox"/> 导入 |
| 扩展槽 | 卡槽名称: slot 3401 插卡: card3401 | 错误: the slot ID already assigned to other PLC |
| | 卡槽名称: slot 3301 插卡: card3301 | 错误: the slot ID already assigned to other PLC |

取消 导入APP配置

PLC 详情

之后自动进入所创建的 PLC 实例的详情页面，首先是该 PLC 的当前运行状态以及操作入口：

状态

| | | |
|---|-------------|---|
| <code>ggg</code> 本地 | 状态: Running | 查看实时变量 |
| 运行引擎: basic (2023-9-18v2) | 持续: 28分 | 统计 |
| 运行程序: xxx (0.0.2-25) | 告警: 1 | 下载运行日志 |
| 状态原因: | 模式: 编译 | 应用活动日 v |

选择按钮“查看实时变量”进入本 PLC 运行状态变量查看页面，点击“统计”进入运行统计数据页面，或下载本 PLC 引擎和应用各自运行日志。

操作

启动
停止
重启
关闭

在版本区域中查看当前配置的引擎(runtime)版本，和应用程序版本。可以选择一个新的版本号然后“升级引擎”或者“升级应用”：

版本设置

| | | | | | | |
|--------|-------|-------|------------|---------|------------|-------------------|
| 引擎命名: | basic | 配置版本: | 2023-9-... | 升级目标版本: | 2023-9-18v | 升级引擎 |
| PLC程序: | xxx | 配置版本: | 0.0.2-26 | 升级目标版本: | 0.0.2 | 升级程序 |

重启生效版本配置

改变正在运行状态 PLC 的引擎版本号，需要重启 PLC 后才能生效。系统会发出告警提示需要重启来生效新引擎版本。

配置当前 PLC 实例的管理参数如下图：

管理设置

| | | | |
|----------|--|---------|---------------------------------|
| 自动启动: | <input type="text" value="是"/> | 禁用: | <input type="text" value="否"/> |
| 自动升级: | <input type="text" value="是"/> | 锁定云端配置: | <input type="text" value="否"/> |
| 运行模式: | <input type="text" value="请选择"/> | 启用资源控制: | <input type="text" value="是"/> |
| 最少内存(M): | <input type="text" value="15"/> (程序推荐配置: 引擎=15M) | 绑定CPU核: | <input type="text" value="自动"/> |

配置项说明:

| | |
|-----------------|---|
| 自动启动: | 系统开机自动启动 PLC 并加载运行 APP，否则 PLC 启动后会待在 RunReady 状态，直到用户点击“启动”才会进入运行状态 |
| 禁用: | 设置后开机后不会启动 PLC 进程 |
| 自动升级 | 当配置新版本后，该设置启动自动升级到新的应用版本。升级中间会有短暂服务停止(小于 1 秒) |
| 运行模式 | PLC 应用解释方式运行，或者编译到机器指令运行 |
| 锁定云端配置 | 锁定后不再接收云端的任何配置 |
| 启动资源控制 | 保证本 PLC 运行所需的最少内存，即便在其他进程消耗过多的系统内存情况下也不受影响 |
| 最少内存 | 在“启动资源控制”情况下，配置当前 PLC 需要保证的最少可用内存 |
| 绑定 CPU 核 | 配置当前 PLC 所绑定的 CPU 核，用于提供高度实时的控制能力 |

在 IO 网络配置中可查看当前 PLC APP 所用到的 IO 设备，如果某些 IO 资源在当前设备上没有配置，将提示如下图：

IO网络配置

当前PLC应用程序配置需要以下资源:

| | |
|--|--|
| modbus总线 (服务) : 总线: bus-1 从站: 1 | 配置状态 <input checked="" type="checkbox"/> |
| EtherCAT: 主站: 0 从站: 0 数据对象(索引/子索引): 5888/0 | 配置状态 <input type="checkbox"/> 未配置 (前往创建>>) |

查看当前 PLC 的最近告警和事件，还可以删除本 PLC。

告警

| | | |
|---------------------|--------------------|--------|
| 2023-01-17 22:20:06 | 告警: PLC由于IO故障暂停运行 | 持续2天 |
| 2023-01-19 23:28:22 | 告警: 需要重启PLC来生效当前配置 | 持续20小时 |
| 2023-01-20 17:17:34 | 告警: 输入持续故障 | 持续2小时 |

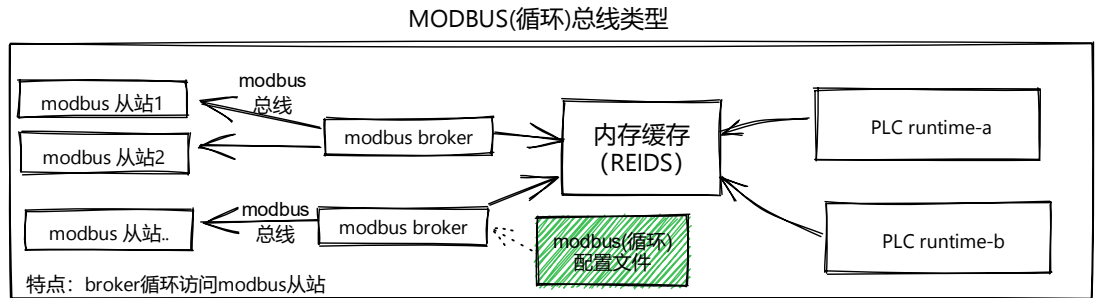
事件

| | | |
|---------------------|---|-----------|
| 2023-01-19 21:19:27 | 事件: Execute local operation command: start | 发生: 22小时前 |
| 2023-01-19 07:38:42 | 事件: Execute Reboot operation command | 发生: 1天前 |
| 2023-01-19 07:38:42 | 事件: Execute local operation command: reboot | 发生: 1天前 |
| 2023-01-19 07:33:26 | 事件: Execute local operation command: start | 发生: 1天前 |
| 2023-01-18 22:14:50 | 事件: Execute local operation command: start | 发生: 1天前 |

4.3.3 总线配置 - Modbus

Modbus 设备服务

modbus 服务(有时也称 modbus broker)使用周期循环方式从 modbus 从站寄存器读取数据然后写入 IO 内存,同时从 IO 内存读取数据将其写入对应的 modbus 从站。



在一台机器上可以运行多个 modbus 服务,并具有灵活的配置能力。主要的配置模式有:

- **公有 Modbus 服务:** 每个主机包含一个公用的 modbus 服务,可以服务于多个 PLC 的 modbus 访问需求。
- **PLC 专有 Modbus 服务:** 每一个 PLC 使用自己私有的 Modbus 服务
- **5G 远端 Modbus 服务:** 当 PLC 运行于 5G 边缘工业云平台,使用运行于设备现场的远端 Modbus 服务

公用 Modbus 服务与 PLC 专用服务

本地 modbus 服务 (服务本机 PLC)

主机使用场景: 现场边缘云平台 5G 云 PLC 平台

本机ID: 1693826259172

| modbus 服务: 公共服务 | 服务状态: ● 运行中 (pid: 2508) | 加载配置: 最新 | 重启 |
|---------------------------|-------------------------|----------|----|
| 虚拟PLC: test-20221202-09>> | | 使用从站数: 1 | |
| 虚拟PLC: n2>> | | 使用从站数: 1 | |
| 虚拟PLC: ddd>> | | 使用从站数: 1 | |
| 虚拟PLC: kkk>> | | 使用从站数: 1 | |

PLC 专用服务

| modbus 服务: test-20221202-09(PLC)>> | 服务状态: ● 运行中 (pid: 2413) | 使用从站数: 1 | 重启 |
|------------------------------------|-------------------------|----------|----|
| | | | |

远端 MODBUS 服务

在某些环境中,需要设置远程 MODBUS 服务。例如在 5G 边缘云(MEC)上部署的 PLC,需要通过运行在远端现场的 MODBUS 服务来读写 MODBUS 从站。如

下图所示



用户创建了远程服务后，需要导出配置，然后在远端设备上将配置导入，远端设备将运行服务于云端的 PLC。



配置生效

当修改 modbus (PLC) 的相关配置后，上面页面中按钮“重启 MODBUS 服务”将会变高亮,同时左侧导航栏入口出现“R”标记(如下),提示需要重启 MODBUS 服务来生效最新的配置修改。

4.3.4 总线配置- EtherCAT

主站管理

在确认出厂包安装过程中安装了 EtherCAT 驱动后，可以开始创建第一个 EtherCAT 总站。点击下图中的“创建总线”按钮，创建总线 0。创建的总线状态显示如下：



主站配置如下：

主站配置

| | | | |
|---------------|-------------------------------------|----------------------|--|
| 网卡MAC: | 00:1f:c6:9c:5a:07 | 网卡类型: | generic |
| 扫描周期 (微秒): | 1000 | | |
| 主站状态更新 (周期数): | 10 | 从站状态更新 (周期数): | 10 |
| 启用: | <input checked="" type="checkbox"/> | | 确认修改 |
| 配置从站个数: 4 | | 配置从站 | 注意: 重启EtherCAT服务会导致瞬间中断服务 重启EtherCAT服务 |

从系统与网络页面查看连接了 EtherCAT 网络的网口机及其 MAC, 在本页中选择对应的 MAC。在本页以及从站的配置页面做了任何修改后, 页面将会高亮按钮“重启 EtherCAT 服务”, 用户需要重启服务来生效配置。

总站上的运行统计显示如下:

运行统计

| | |
|----------------------|-------------------|
| Tx frames: | 14039067 |
| Tx bytes: | 1989802724 |
| Rx bytes: | 1989802574 |
| Tx errors: | 0 |
| Tx frame rate [1/s]: | 1000 999 1000 |
| Tx rate [KByte/s]: | 140.3 140.1 140.2 |
| Rx frame rate [1/s]: | 1000 999 1000 |
| Rx rate [KByte/s]: | 140.3 140.1 140.2 |

ESI 文件管理

一般 EtherCAT 从站设备商会提供一份设备的 ESI 描述文件, 包含了设备的所有相关信息。使用 ESI 帮助快捷和充分使用配置和使用 EtherCAT 从站设备。

选择按钮“ESI 文件管理”, 进入 ESI 管理页面。选择按钮“导入 ESI 文件”, 然后选择对应的 ESI XML 格式文件, 导入后, 将显示设备厂商, 分类, 以及分类下的所有设备类型。

← 返回上一级 [导入 ESI 文件](#)

| 名称 | ID | 种类 | Physics | 操作 |
|---|-----------------------------------|---------------|---------|---------------------------------------|
| <input type="checkbox"/> Beijing iTegva Technology Co., Ltd. | 厂商ID: #x00000A09 | | | |
| <input type="checkbox"/> EtherCAT Remote IO Module (E7x) (Et... | | E7x Series | | |
| <input type="checkbox"/> Inovance(汇川技术) | 厂商ID: #x00100000 | | | |
| <input type="checkbox"/> Servo Drives(伺服驱动器) | | InoServo | | |
| IS620N_ECAT_v2.6.8 | 产品ID: #x000C0108 产品版本: #x00010001 | IS620N | YY | 查看 删除 |
| <input type="checkbox"/> VMORE | 厂商ID: #x942 | | | |
| <input type="checkbox"/> Leadshine Technology Co.,Ltd.(深圳市雷... | 厂商ID: 0x00004321 | | | |
| <input type="checkbox"/> Stepper Drives(步进驱动器) | | Stepper Drive | | |

从站列表视图

在主站页面中, 可以看到当前主站下配置的从站数量。点击按钮“配置从站”,

进入从站列表视图，如下图所示。

每个从站包含如下信息：

- 设备名称
- 状态：如果为空，表示设备离线状态。只有进入 OP 状态，设备才能正常工作
- ID：厂商 ID, 设备 ID
- 别名：当前从站所设置的地址号
- 告警：设备的当前告警数量

| 从站管理 [所在总站:0] 运行 2 未就绪 2 故障 0 禁用 0 全部 | | | | 标签: 请选择 | 排序: 请选择 | ESI文件管理 | 手动添加设备 | 扫描设备 |
|---|---|-------|------------|---------|------------|-------------------------------------|--------|--------|
|  | IS620N_ECAT_v2.6.8 >> | | | | | | | 状态: |
| 厂商ID: | 0x00100000 | 产品ID: | 0x000C0108 | 产品版本: | 0x00010001 | esi配置: IS620N-Ecat_v2.6.8.xml | | |
| 位置: | 0 | 别名: | 19 | | | | | 告警: 1 |
|  | E7.820.004 32-ch Dlg.Output(Access_Bit) >> | | | | | | | 状态: |
| 厂商ID: | 0x00000A09 | 产品ID: | 0x00000400 | 产品版本: | 0x00000064 | esi配置: E7x Series EtherCAT_V1.5.xml | | |
| 位置: | 0 | 别名: | 18 | | | | | 告警: 1 |
|  | IS620N_ECAT_v2.6.2 >> | | | | | | | 状态: OP |
| 厂商ID: | 0x00100000 | 产品ID: | 0x000C0108 | 产品版本: | 0x00010001 | esi配置: IS620N-Ecat_v2.6.8.xml | | |
| 位置: | 0 | 别名: | 20 | | | | | 告警: 0 |
|  | E7.820.001 16-ch Dlg.In/16-ch Relay Out(Access_Byte) >> | | | | | | | 状态: OP |
| 厂商ID: | 0x00000A09 | 产品ID: | 0x00000101 | 产品版本: | 0x00000064 | esi配置: E7x Series EtherCAT_V1.5.xml | | |
| 位置: | 0 | 别名: | 16 | | | | | 告警: 0 |

添加 EtherCAT 从站

从 ESI 离线添加从站

点击按钮“手动添加设备”，弹出对话框如下。输入设备的别名和位置，选择设别类型，然后确定添加。

可选的设备类型来自于所导入的 ESI 文件。

添加设备
×

别名: - 1 +

位置: - 0 +

设备类型: DM3E-556(COE) v

前往管理

取消
确定

从在线设备添加

用户需要完成从站设备的物理连线，加电后主站将会扫描到从站。如下图，如果一个新的设备在线，但是没有被配置，将显示“新设备”的标签。如果从站的地址为 0，先设置从站的别名地址。

可以选择“从设备添加”，系统将读取设备上的 ESI 信息，来初始化设备同步管理器 (SM) 和周期数据对象 (PDO) 的配置。注意这种模式下，系统不能获得设备所有支持的字典对象列表，用户可能需要参考设备手册来进行更多的配置。

另外一种添加方法是“从 ESI 添加”，当该设备已经有对应的 ESI 被导入后出现此选项，一般建议使用此方式。

E7.820.001 16-ch Dig.In/16-ch Relay Out(Access_Byte)

新设备
从设备添加

厂商ID: 0x00000A09 产品ID: 0x00000101 产品版本: 0x00000064 esi匹配: E7x Series EtherCAT_V1.5.xml [从esi添加](#)

位置: 0 别名: 16 [↗](#)

配置从站

EtherCAT 从站列表中进入从站的配置界面如下：

信息

IS62N_ECACAT_v2.6.2

状态:

厂商ID: 1048576 产品ID: 786696 产品版本: 65537

位置: 0 别名: 20
告警: 0

根据从站的手册配置时钟：

时钟

同步方式: SYNC0

SYNC0同步周期(纳秒): 1000000 SYNC1同步周期(纳秒): 0

SYNC0时间偏移(纳秒): 500000 SYNC1时间偏移(纳秒): 0

确认修改

配置同步管理器的 PDO 分配:

PDO分配

| SM | 大小 | 类型 | PDO个数 |
|----|-----|------|-------|
| 0 | 128 | 邮箱输出 | 0 |
| 1 | 128 | 邮箱输入 | 0 |
| 2 | 0 | 输出 | 1 |
| 3 | 0 | 输入 | 1 |

| PDO分配 (SM3) | 编号 |
|-------------------------------------|------------------------|
| <input checked="" type="checkbox"/> | 6656 (0x1a00) |
| <input type="checkbox"/> | 6913 (0x1b01) [排除6656] |
| <input type="checkbox"/> | 6914 (0x1b02) [排除6656] |
| <input type="checkbox"/> | 6915 (0x1b03) [排除6656] |

确认修改

PDO 列表:

PDO List

| 索引 | 名称 | 标志 | 类型/数据类型 | SM | 操作 |
|-------------------|---------------------------|----|----------------|----|---------|
| 5632 (0x1600) | 1st receive PDO Mapping | | RxPDO(输出) | 2 | + 删除 刷新 |
| 24640 (0x6040) :0 | Controlword | | UINT(bits: 16) | | 删除 刷新 |
| 24698 (0x607a) :0 | Target position | | DINT(bits: 32) | | 删除 刷新 |
| 24760 (0x60b8) :0 | Touch probe function | | UINT(bits: 16) | | 删除 刷新 |
| 5889 (0x1701) | 258th receive PDO Mapping | F | RxPDO(输出) | | + 删除 刷新 |
| 5890 (0x1702) | 259th receive PDO Mapping | F | RxPDO(输出) | | + 删除 刷新 |
| 5891 (0x1703) | 260th receive PDO Mapping | F | RxPDO(输出) | | + 删除 刷新 |
| 5892 (0x1704) | 261th receive PDO Mapping | F | RxPDO(输出) | | + 删除 刷新 |
| 5893 (0x1705) | 262th receive PDO Mapping | F | RxPDO(输出) | | + 删除 刷新 |

确认修改

添加, 编辑 PDO

编辑PDO ×

* 名称

* 索引

标志 强制 固定内容

* 方向 TxPDO(输入) RxPDO(输出)

排除

添加 PDO 的子项:

添加子项 ×

Index

Subindex

名称

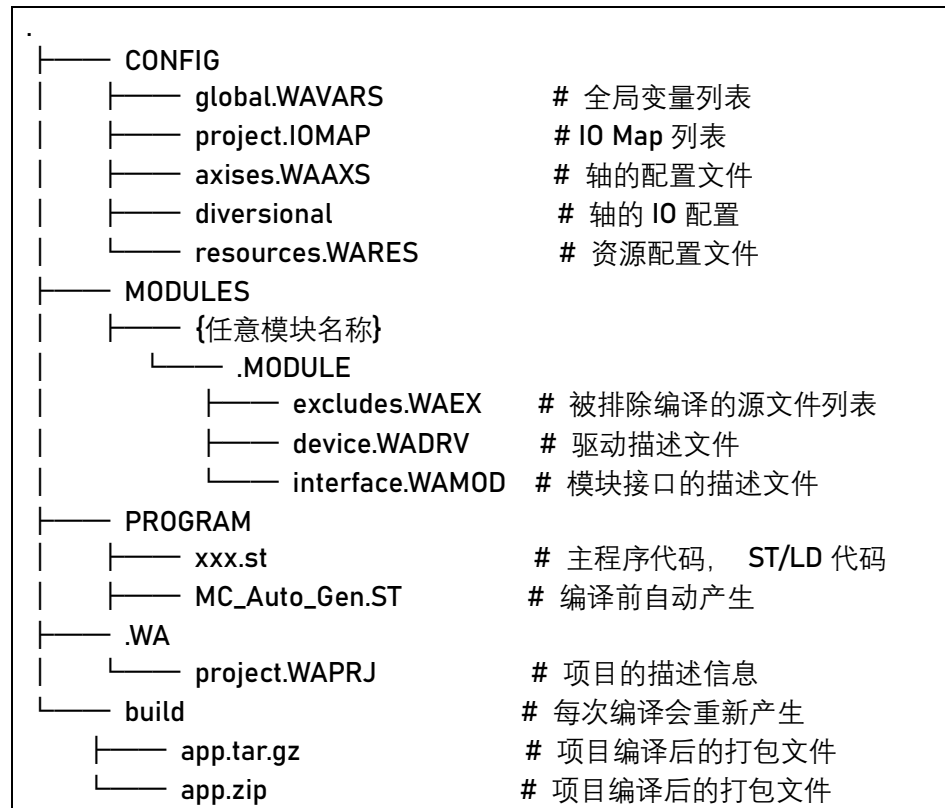
* 数据类型:

提醒: 所有的配置修改后, 都需要重新启动 EtherCAT 服务来生效配置。系统将提供告警与提示, 操作者选择合适时间来生效配置。

5 WasomeIDE 项目管理与操作

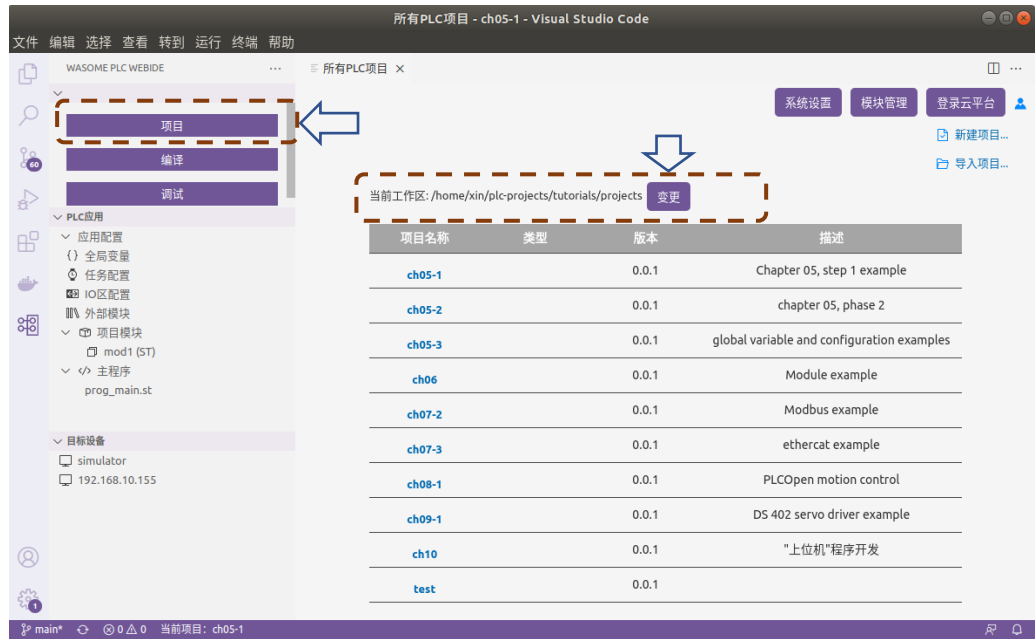
5.1 项目组成结构

WasomePLC IDE 创建的一个典型的 PLC APP 项目，包含如下文件。不同项目中，用户可以创建任意数量的模块和源程序。



5.2 进入项目管理

首次启动后，需要设置工作区目录。如下图点击“项目”，如果当前没有打开有效的 PLC 项目，将进入“所有项目”页，点击“变更”来设置项目工作区路径。变更后，将显示当前工作区内的所有项目列表。



5.3 创建项目

进入“所有 PLC 项目”界面，然后点击右上角的“新建项目..”。



在如下弹窗的对话框中输入项目名称与说明，然后点击按钮“创建”。

新建项目 ×

项目名称:

项目说明:

主程序类型: ▼

PLC型号: ▼

启用版本控制功能:

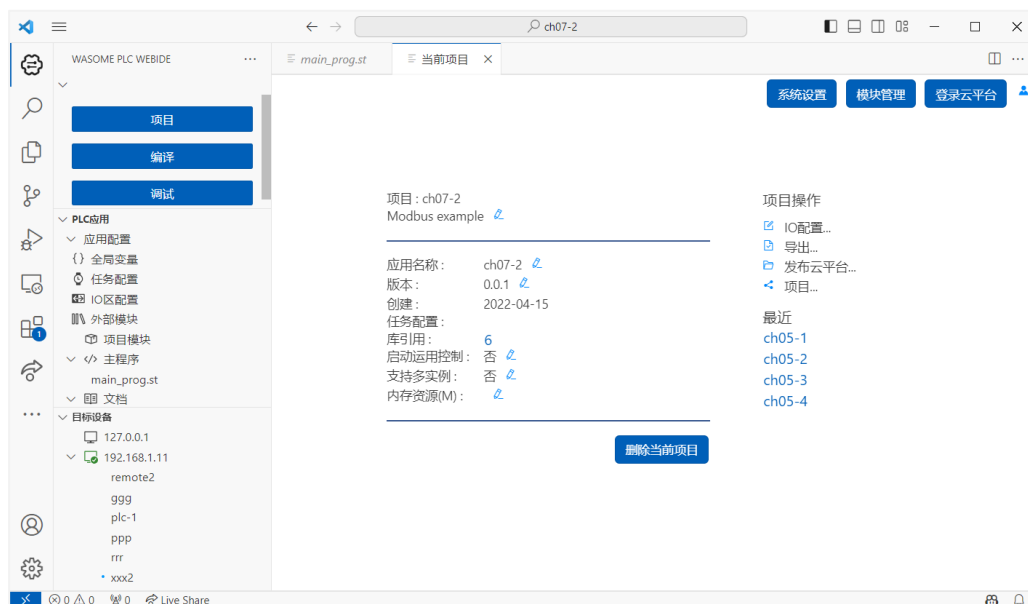
导入项目

一个项目可以导出为 zip 包文件。得到项目包文件后，也可以将其导入到 WasomePLC IDE，来创建一个新项目。

在“所有 PLC 项目”界面，点击右上角按钮“导入项目”，弹出“导入项目”对话框，选中需要导入文件路径名。

5.4 修改当前项目属性

如果已经打开了一个 PLC 项目，点击“项目”按钮将进入“当前项目”界面，点击对应项的修改按钮即可修改项目。如下图所示：



属性配置

包含以下项目属性设置：

- **应用名称，版本**
- **启动运动控制**
- **支持多实例**：是否可以使用本程序在一台机器上创建多个 PLC 实例
- **内存资源**：定义本 PLC 程序运行时刻所需要的最少内存资源数量，在多任务运行环境(如实时 Linux)中，持续运行中所需的内存资源将被得到保障。

删除项目

删除项目要谨慎。在“当前项目”界面，点击“删除当前项目”，确认即可删除当前项目。

导出项目

在“当前项目”界面，点击“导出”修改，进入“项目导出”界面，选择项目导出包的目标路径，即可生成文件如 project-name.zip

进入“全部项目”

从右部点击按钮“项目..”会进入“全部项目”界面。

6 模块操作

WasomePLC 支持多语言的混合编程，每一个模块使用一种编程语言（LD, ST, C, C++）。每个模块都可以单独编译，并导出为模块安装包，然后在本机或者其他主机的 WasomePLC IDE 环境中作为外部模块导入。

6.1 创建内部模块

在项目左边栏，鼠标停留在“项目模块”上时，会有提示菜单如下：



点击“新建”按钮新建，即可进入“新建模块”界面

The 'New Module' form contains the following fields:

- 模块名称 (Module Name): mod1
- 供应商 (Supplier): WanSr
- 版本 (Version): 0.1
- 开发语言 (Development Language): ST
- 调试信息 (Debug Information): 是
- 发行类别 (Release Category): 源码
- 伺服驱动 (Servo Drive): 否
- 描述信息 (Description): 请输入

界面中各个字段信息如下：

| 名称 | 意义 | 可选值 | 必填项 |
|------|----------------------------|------------------|-----|
| 模块名称 | 作为 ID 使用 | 英文和数字组成的字符串，不要重名 | 是 |
| 供应商 | 供应商名称 | 任意字符串 | 否 |
| 版本 | 模块的版本号 | x.x.x 格式 | 是 |
| 开发语言 | 模块中的编程语言，选择：LD, ST, C, C++ | 选择选项 | 是 |

| | | | |
|------|----------------|-------|---|
| 调试信息 | 是否支持调试 | 选择选项 | 是 |
| 发行类别 | 用于在模块导出时是否包含源码 | 选择选项 | 是 |
| 描述信息 | 模块的说明 | 任意字符串 | 否 |

创建后左边的项目模块下显示刚创建的模块如下：



6.2 导出内部模块

模块可以独立编译、打包，便于分发共享。在项目左边栏，鼠标停留在想要导出的模块上时，会有提示菜单，点击“导出”按钮模块导出，进入“模块导出”界面，选择导出文件名如 module-name.zip 的存放路径。



6.3 模块接口设置

鼠标停留在要设置的模块上时，会有提示菜单，点击菜单“模块设置”按钮模块设置，进入“模块设置”界面。

| | | | | |
|-------|----------|-------|-----|----|
| 模块名称： | mod1 | | | 🗑️ |
| 供应商： | WanSheng | 版本： | 0.1 | |
| 开发语言： | ST | 调试信息： | 是 | |
| 发行类别： | 源码 | 伺服驱动： | 否 | |
| 描述信息： | | | | |

新建数据类型定义



新建函数定义

创建函数的过程如下：

1. 点击“POU”按钮开始进行接口信息编辑。
2. 点击“新增 POU”按钮；
3. 在“POU”右侧的编辑框中填写函数名，并将“POU 分类”选择为“函数”；
4. 选择函数的返回值类型；
5. 若函数需要有参数，点击“新增变量”按钮并填写变量的名称、分类、数据类型，直到所有的参数编辑完成；
6. 点击编辑框右上角的保存按钮；



新建功能块定义

点击“新增 POU”按钮出现一个新的 POU 编辑条目如下：




设置功能块的过程如下：

1. 在“POU”右侧的编辑框中填写功能块名称,并将“POU 分类”选择为“功能块”或者“函数”;
2. 如果是功能块且当前模块是基于 C/C++ 语言,还需要选择“Class”,该选项说明实现功能块的实现方法:
 - “C”:表示采用 C 语言的方式编写功能块逻辑,
 - “C++”:表示采用 C++ class 的方式编写功能块,
 - “C++ asset”:和“C++”类似,其生成的框架代码用于集成存量的 C++ 代码
3. “APP 不可见”:如果为是,表示此功能块或函数为内部使用,不能被用户程序所调用
4. 若功能块需要有成员变量,点击“新增成员变量”按钮并填写变量的名称、分类、数据类型,直到所有成员变量编辑完成;
5. 点击编辑框右上角的保存按钮;

删除内部模块

进入“模块设置”界面,点击右上角删除按钮模块删除,确认后删除。

6.4 生成模块框架代码

完成上述编辑后,将鼠标移动到左侧边栏“项目模块”栏并展开,找到刚才创建的模块,点击其右侧的 (生成接口代码)按钮,这一步将为当前模块生成框架代码:

```
.MODULE/cpp/wa_interface.h  
implements/cpp/counter.cpp  
implements/cpp/wa_func.cpp
```

生成的代码主要包含三类:

- 头文件:即上述“wa_interface.h”,包含类型定义和函数原型声明。用户不应该修改该文件,且已设置为只读。
- 函数实现文件:即上述“wa_func.cpp”,所有函数类型的 POU 的实现都在这一个文件中。
- 功能块实现文件:即上述“counter.cpp”,和函数不同,每个功能块将有自己独立的文件。

6.5 外部模块引用与解除引用

模块引用

一个项目可以引用外部模块（或全局模块）。鼠标停留在“外部模块”上时，会有提示菜单，点击“进入管理”按钮进入管理，即可进入“模块管理”界面。

在“模块管理”界面，选中需要引用的模块，点击“添加到当前项目”即可引用选中的模块。

解除引用

鼠标停留在需要解除的外部模块上时，会有提示菜单，点击“解除引用”按钮解除引用，即可解除引用该模块。

7 连接目标 PLC 设备

7.1 连接目标设备

在 WasomePLC IDE 中，可以连接多个目标设备，进行开发与调试。

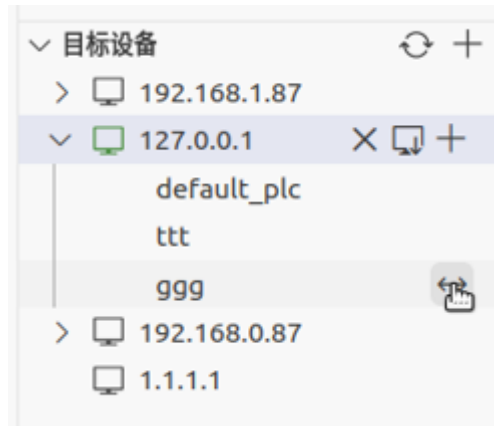
添加目标设备

鼠标停留在“目标设备”上时，点击加号图标按钮，会有提示“手动添加目标设备”进入管理，即可进入正上方的“新建目标调试设备”界面，填上目标设备的 IP 地址即可，WasomePLC IDE 会自动更新目标设备下的 PLC 列表。

如果目标设备上尚无 PLC，可以点击加号按钮，创建一个 PLC 实例。

连接 PLC

如下图，选择连接某个 PLC 图标，当前 IDE 将连接该 PLC，作为调试、部署、装载等操作的目标 PLC。



删除目标设备

鼠标停留在要删除的目标设备上时，会有提示菜单，点击“删除目标设备”按钮删除，即可删除目标设备。

7.2 操作目标 PLC

在连接上目标 PLC 后，将显示如下视图。用户可进行装载和卸载 APP，查看变量，关闭和重启等操作。



8 项目配置

8.1 全局变量配置

全局变量可以被各个模块和主程序访问使用。点击左侧边栏中的“全局变量”，可以进入“PLC 全局变量”界面。

| 变量名 | 种类 | 类型 | 初始值 | 特性 | 注释 | 操作 |
|---------|------------|--------|---------|----|----|-------|
| g_iVar1 | VAR_GLOBAL | INT | 1 | | | — ↑ ↓ |
| g_iVar2 | VAR_GLOBAL | INT | | | | — ↑ ↓ |
| g_bVar | VAR_GLOBAL | BOOL | FALSE | | | — ↑ ↓ |
| g_fVar1 | VAR_GLOBAL | REAL | 0.1 | | | — ↑ ↓ |
| g_sVar | VAR_GLOBAL | STRING | 'hello' | | | — ↑ ↓ |

添加变量

各个字段意义如下：

名称 意义 可选值 必填项

变量名 作为 ID 使用 英文和数字组成的字符串，全局唯一 是

种类 变量类型 选项值 是

类型 数据类型 选项值 是

初始值 初始化使用 需要与数据类型匹配 否

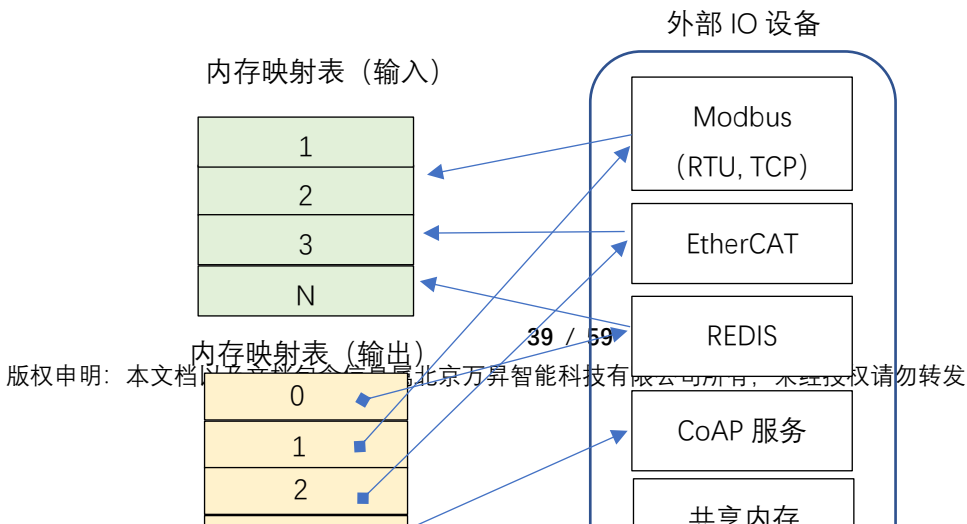
特性 变量的特性描述，如：RETAIN 或 CONSTANT 选项值 否

说明：本界面只编辑非 IO 类型的全局变量。关于 IO 类型全局变量定义请参考章节 8.2。

8.2 IO 组网与通道配置

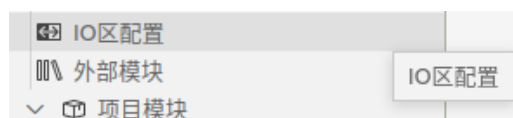
外部 IO 设备的寄存器可以映射到 PLC 的内存输入区（又称 I 区）、输出区（又称 Q 区）。WasomePLC 提供机制让每一个内存位置通过配置映射到不同类型的外部数据（如下图），目前支持的外部数据类型包含：

- Modbus
- Ethercat、CANOPEN、Ethernet/IP
- Profinet, Profibus
- 控制器扩展槽插卡
- REDIS
- CoAP 服务
- 共享内存

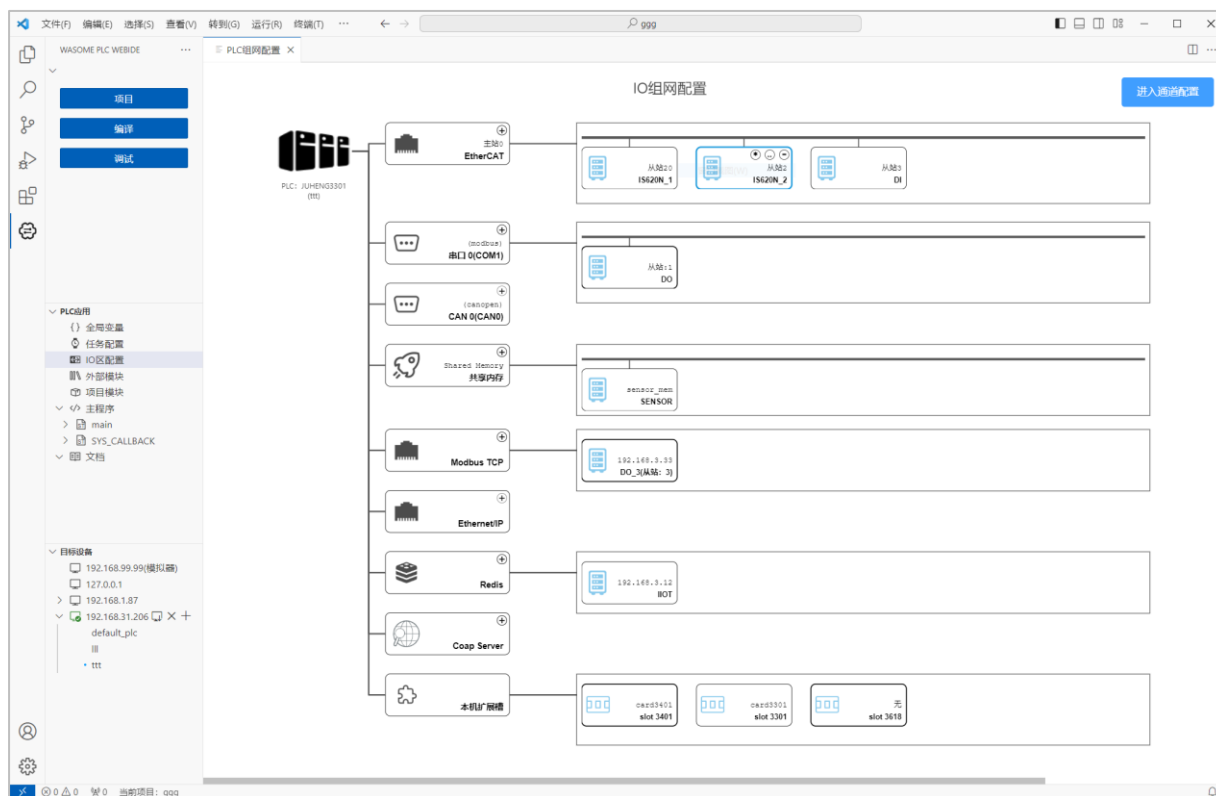


组网编辑

点击左侧项目面板中的“IO 区配置”（如右图所示），进入“PLC IO 组网配置”窗口。



组网配置界面如下：



组网图将根据当前项目所选择的 PLC 型号，显示组网图。用户可在每一种接口上，点击“加号”图标，添加所连接的设备或者从站。

在本机扩展槽中，点击添加所实际配置的物理插卡。

进入通道配置：点击右上角的按钮“进入通道配置”，或者选择从站边框内最左

边的图标，将直接定位至本设备的动态配置。

IO 通道配置



PLC 连接

如果需要联机调试，需要事先在上图右下角的“目标设备”视图中点击目标主机一个下属的 PLC 连接。

或者在上图“PLC 连接”区域中点击“配置”按钮，弹出以下对话框，选择目标主机上要连接的某个 PLC。



设置 IO 刷新周期

点击该区域的按钮“修改”，可以进入以下界面配置 IO 的刷新周期：

IO周期与故障处理编辑

设置IO故障缺省周期 : 0 秒

0 微秒

设置IO故障触发条件

读失败连续次数 : 100

写失败连续次数 : 10

设置IO故障缺省处理 : 停机

取消 确定

配置 IO 故障处理策略

进入“IO 故障处理”小节的编辑模式，设置触发 IO 故障处理的条件，可以是任一个 IO 的连续读失败次数，或者为连续写失败次数。

当 IO 故障被触发，用户配置缺省的处理策略：

- **停机**：控制 APP 程序将进入停止状态。即便 IO 故障恢复，仍然需要用户操作“启动”来重新运行 APP。注意：在 IO 故障中导致停机时，用户无法成功操作“启动”。
- **忽略**：在 IO 故障条件发生后，程序继续运行
- **APP 处理**：在 IO 故障条件发生后，系统将调用 APP 提供的处理函数。APP 通过函数返回值来决定：1) 忽略当前 IO 本次故障，2) 永久忽略当前 IO 的故障，3) 或者进入故障停机。

设置IO故障缺省处理 : 停机

停机

忽略

APP处理

特定 IO 通道上的故障策略

在缺省 IO 故障处理策略的基础上，可为每一个 IO 设置特别处理策略。假设缺省为停机，设置某个 IO 为忽略，则该 IO 发生故障将被忽略。

如下图所示，在从站进入编辑模式后，点击任意 IO 项的菜单，从菜单项“IO 故障处理”中选择本 IO 的处理策略。



外部 IO 配置状态

在外部 IO 配置状态章节中，可以查看当前各种 IO 类型配置的统计数量。

编辑 MODBUS 从站

| MODBUS从站配置 | | | | | | | 设置MODBUS服务 | + | |
|------------|-----|-------|------|------|---------------------------------|------|------------|----|---|
| 从站编号: 0 | | 设备地址: | | 从站号: | | 从站名: | | 🗑️ | 🔗 |
| 编号 | 寄存器 | 地址 | 双字编码 | 连续条数 | 变量名 | | | | |
| 输入区 | | | | | | | | | |
| 0 | 3 | 100 | | 1 | s1_fc3_100: INT | | | | |
| 1 | 3 | 101 | | 3 | s1_fc3_101: ARRAY[0..2] OF INT | | | | |
| 2 | 3 | 104 | abcd | 2 | s1_fc3_104: ARRAY[0..1] OF REAL | | | | |
| 3 | 1 | 80 | | 2 | s1_fc1_80: ARRAY[0..1] OF SINT | | | | |
| 从站编号: 1 | | | | | | | | | |
| 设备地址: | | 从站号: | | 从站名: | | 🗑️ | | | 🔗 |
| 编号 | 寄存器 | 地址 | 双字编码 | 连续条数 | 变量名 | | | | |
| 输出区 | | | | | | | | | |
| 0 | 6 | 100 | | 1 | s2_fc6_100: INT | | | | |
| 1 | 6 | 101 | | 3 | s2_fc6_101: ARRAY[0..2] OF INT | | | | |
| 2 | 6 | 104 | abcd | 2 | s2_fc6_104: ARRAY[0..1] OF REAL | | | | |
| 3 | 5 | 80 | | 2 | s2_fc5_80: ARRAY[0..1] OF SINT | | | | |

首先应该点击“设置 MODBUS 服务”，选择服务模式。一般建议使用“私有”模式，让系统为本 PLC 配置一个专用的 MODBUS 服务。

选择modbus服务模式 :

modbus Redis :

说明: modbus服务为远端5G时必须填modbus Redis

然后点击“加号”按钮添加 MODBUS 从站，或者编辑在该从站下所有需要访问的寄存器。从站的总线可以选择要绑定的目标设备上总线，或者为一个 MODBUS-TCP 设备，或者待部署时再确定。然后输入从站号。

Modbus从站

从站 0 所在总线: 从站ID: 说明: + ✎ 🗑

| 寄存器 | 地址 | 双字编码 | 连续条数 | 变量名 |
|-----|----|------|------|-------------|
| 输入区 | 3 | 1 | 1 | sensor1:INT |

从站 1 所在总线: 请选择 从站ID: 说明: 请输入 ✎ 🗑 ✕

| 寄存器 | 地址 | 双字编码 | 连续条数 | 变量名 | + 输入 | + 输出 |
|-----|---|--|--|--|---|---|
| 输入区 | 线圈-1 | 1 | 请选择 | 8 | dig_inpt | SINT |
| 输出区 | 线圈-5 | 1 | 请选择 | 8 | dig_outt | SINT |

寄存器： modbus 设备访问的操作码

地址： 寄存器地址

编码： MODBUS 设备使用双寄存器表示浮点或者整数时所采用的编码

条目： 表示从连续的寄存器一次读取或写入数据个数。

变量名： 每一个 modbus 寄存器均需映射到一个全局变量，该变量可以从 ST 或者 LD 程序中访问。如果条目数大于 1，则对应到一个数组类型的全局变量。界面上用户需要选择变量对应的数据类型。

编辑 ETHERCAT 从站

EtherCAT从站配置 刷新配置

从站编号: **1000** 主站: 0 从站: 20 从站名: IS620N_1 🗑 ✎

| 通道 | 对象索引(10进制) | 子索引 | 变量名 | 输入 | 输出 | |
|------|------------|---------------|-----|--|--|---|
| 输入通道 | 0 | 24641(0x6041) | 0 | d1000_Statusword | UINT | ⋮ |
| | 1 | 24676(0x6064) | 0 | d1000_Position_actual_value | DINT | ⋮ |
| | 2 | 24761(0x60b9) | 0 | d1000_Touch_probe_status | UINT | ⋮ |
| | 3 | 24762(0x60ba) | 0 | d1000_Touch_probe_pos1_pos_ | DINT | ⋮ |
| | 4 | 24764(0x60bc) | 0 | d1000_Touch_probe_pos2_pos_ | DINT | ⋮ |
| | 5 | 24639(0x603f) | 0 | d1000_Error_code | UINT | ⋮ |
| | 6 | 24829(0x60fd) | 0 | d1000_Digital_inputs | UDINT | ⋮ |
| | 7 | 24672(0x6060) | 0 | d1000_fff | USINT | ⋮ |
| 输出通道 | 0 | 24640(0x6040) | 0 | d1000_Controlword | UINT | ⋮ |
| | 1 | 24698(0x607a) | 0 | d1000_Target_position | DINT | ⋮ |
| | 2 | 24760(0x60b8) | 0 | d1000_Touch_probe_function | UINT | ⋮ |

添加 EtherCAT 从站后，该从站所配置 PDO 的数据对象将自动同步，用户可编辑对应的程序变量名。

Redis 与 CoAP server 配置

IDE 支持使用 Redis 和 CoAP server 来配置 IO 通道。这些通道的数据将自动与目标 server 上的数据进行同步。

Redis配置 +

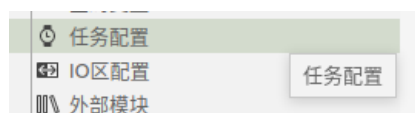
| Server编号: 100 | | IP/PORT: 127.0.0.1:1111 | 非订阅模式 | 说明: | 🗑️ ✎ |
|---------------|------|-------------------------|-------|-----------|------|
| 编号 | 键名 | 请求 | 字段名 | 变量名 | |
| 输入区 | 0 ee | GET | | eff: UINT | |

CoAP Server配置 +

| Server编号: 200 | | IP: 127.0.0.1:1234 | 说明: | 🗑️ ✎ |
|---------------|---------------|--------------------|-------|----------|
| 编号 | URL | 连续条数 | 字段名 | 变量名 |
| 输入区 | 0 fdfdasfasdf | 1 | ffffa | f: SINT |
| 输出区 | 0 aaaaa | 1 | ffff | f1: SINT |

8.3 任务配置

PLC 程序是可以配置成为不同周期来循环执行的，一个周期配置的一个程序称为任务，可以同时配置多个任务。点击左侧边栏中的“任务配置”，可以进入“PLC 资源任务”界面。



在“PLC 资源任务”界面，先“添加任务”，可以同时添加多个任务，然后再“添加实例”，添加实例时需要选择“执行入口”，执行入口是 PLC 程序中的“PROGRAM”的名称。

| 任务列表 | | | | 添加任务 |
|-------|------|------|-----|------|
| 任务名称 | 触发方式 | 周期 | 优先级 | 操作 |
| task0 | 循环 | 1000 | 1 | 删除 |

| 程序 | | | | 添加实例 |
|---------|----|-------|----|------|
| 执行入口 | 次序 | 所属任务 | 操作 | |
| PLC_PRG | | task0 | 删除 | |

8.4 运动配置

如果当前项目的设置中，启用了“启动运动控制”，则可以进行运动控制配置，在左侧面板“应用配置”下会出现“运动轴”的菜单。

运动轴创建

在左边项目面板，鼠标停留在“运动轴”上时，会有提示菜单，点击“新建”按钮新建，即可进入“创建运动轴”界面



在如下见面中输入“轴名称”、“驱动类型”后创建运动轴，左侧项目面板中出现新创建的轴“Axis2”。



鼠标点击项目面板“运动轴”下任意轴对象，进入该轴设置界面。点击修改按钮显示如下。设置驱动运动轴的伺服器通讯“协议”、“主站”和“从站”等信息，与轴的运动控制等相关参数。



运动轴删除

运动轴的删除要谨慎操作。鼠标停留在要删除的“运动轴”上时，会有提示菜单，点击“删除”按钮删除，即可删除运动轴。

9 代码编写

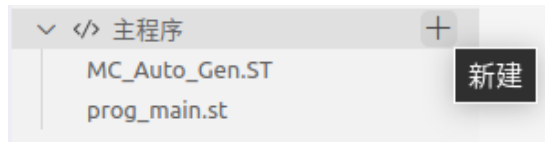
9.1 梯形图开发

WasomePLC IDE 提供具有现代和简洁的特色的梯形图编辑器，具备易用和实用的特点。主要如下：

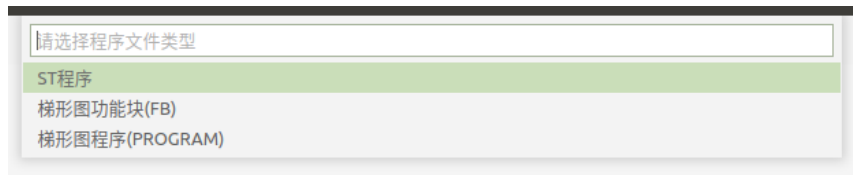
- 1) **所见即所得**：所有的操作入口都在所选择的对象上完成。
- 2) **支持平板电脑上的操作习惯**：无菜单，无鼠标右键菜单，无拖拽操作。
- 3) **右侧停靠的变量面板**：既为 LD 编辑操作释放更多的空间，也无需滚动就展示更多的变量。使用标签方式显示变量属性，简洁美观。
- 4) **简化的功能块工具箱操作**：先插入空功能块，然后从选择功能块/函数类别完成实例化。

创建梯形图文件

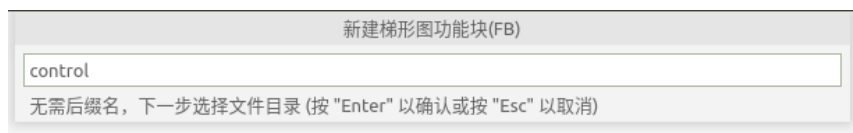
在左面项目面板的主程序上，点击“新建”按钮如下：



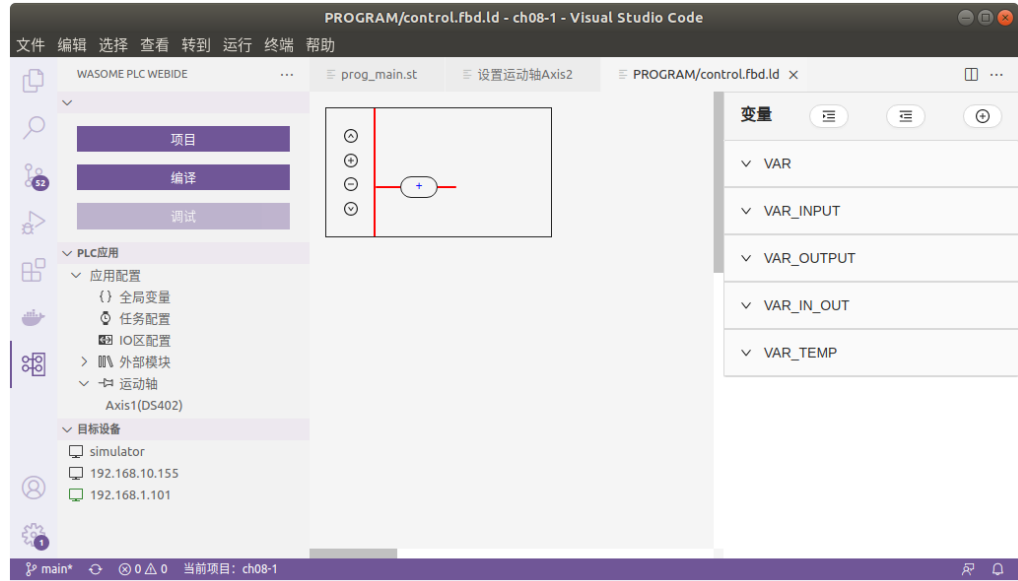
选择梯形图功能块，或者梯形图程序如下：



输入梯形图文件名如下：



IDE 然后创建一个新的梯形图文件，包含一个空的网格如下：



网格操作

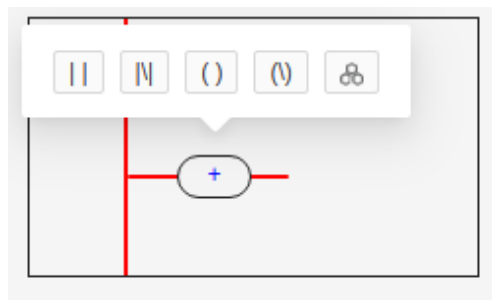
鼠标移动进入网格左侧的头部区域，将自动显示四个按钮，从上之下分别提供以下功能：

- 向上移动本网格
- 创建一个新的网格
- 删除本网格
- 向下移动本网格

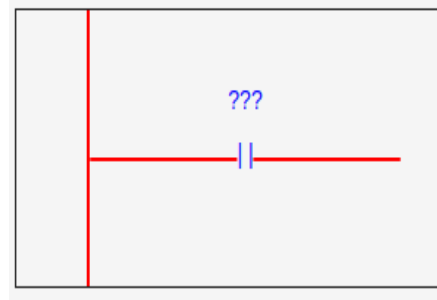
逻辑编辑

插入节点

鼠标停留在在能流线的“加号”按钮上，将弹出一个节点类型图标面板如下左图，点击触点类型图标，将插入一个触点如下右图：



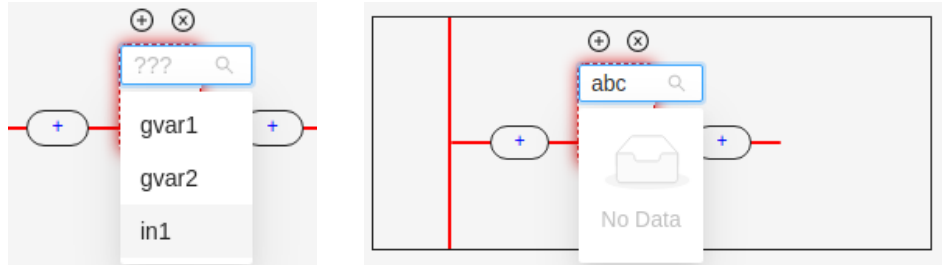
左图：插入节点前



右图：插入节点后

定义变量

点击触点节点上面的“???”，可以从下拉菜单选择已经定义的变量。可以输入新的变量名，如下图中输入新的变量名“abc”后回车：



弹出对话框如下，选择变量类型和数据类型，也可以输入初始值。然后点击确认。

变量操作

* 变量:

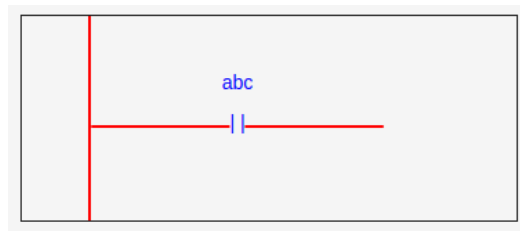
* 变量类型:

* 数据类型:

初始值:

注释:

确认后该触点节点对应到新建的变量“abc”如下图：

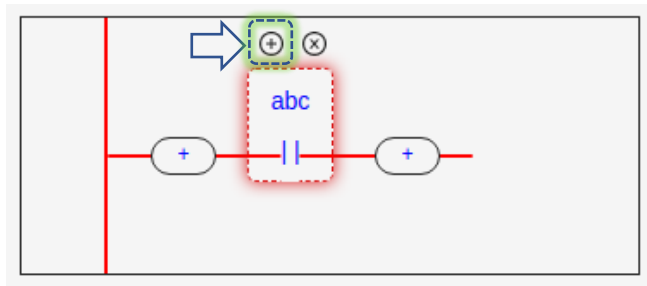


在编辑器右侧的变量面板中，出现新建的变量“abc”。同时可以看到之前定义的全局变量，其中包含地址符如“%IX1”的标签表示该全局变量对应一个 IO 内存，如下图所示：

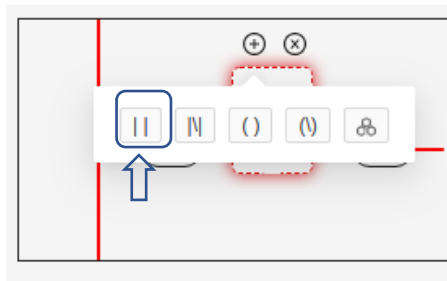


插入并行分支

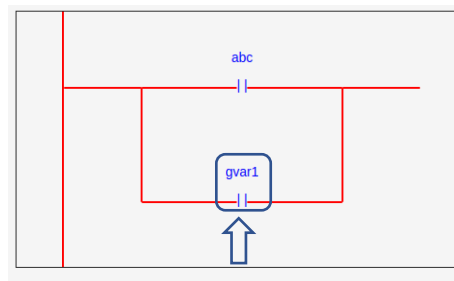
选中一个对象，对象左右连线上将显示加号按钮，用来添加前续和后续节点。上方出现两个图标，分别代表添加并行节点和删除本节点。



鼠标移动到上方的加号图标，显示可添加的节点类型图标面板如下左图，然后从弹出面板中选择触点图标，添加并行的触点后如下右图：



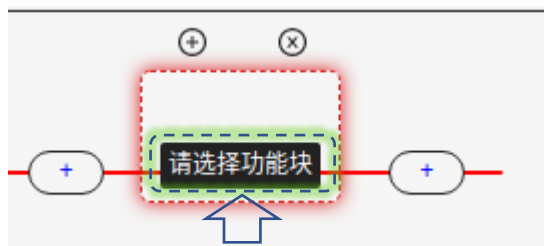
左图:插入并行分支前



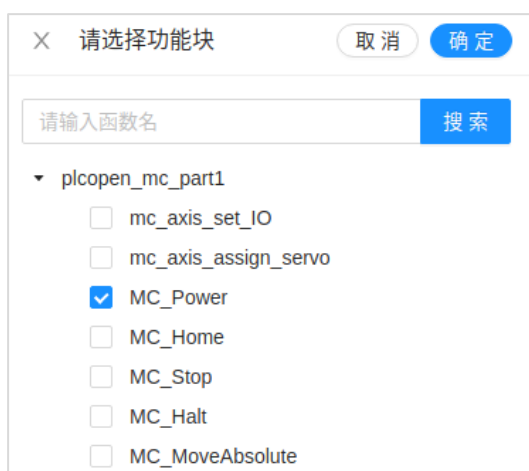
右图: 插入并行分支后

功能块/函数：

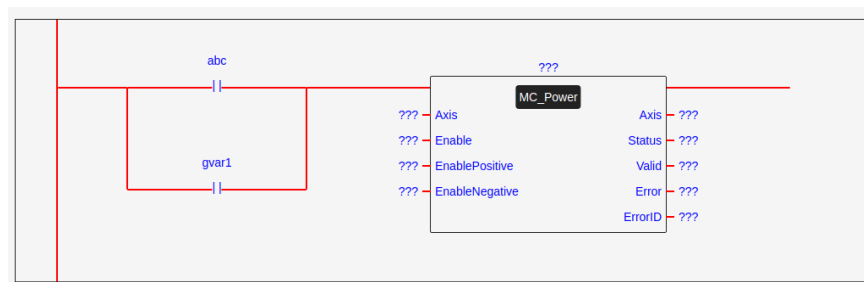
插入节点时选择图标  插入一个空的功能块如下：



上图中点击“请选择功能块”来进一步确定具体的功能块或者函数类型，右侧弹出如下面板：



在上图中选择功能块类型 MC_Power，该空功能块完成实例化变成如下：



9.2 ST 代码开发

在创建文件时，选择 ST，然后输入 ST 文件名，将新建一个 ST 文件。用户可以从编辑区编码 ST 源程序。

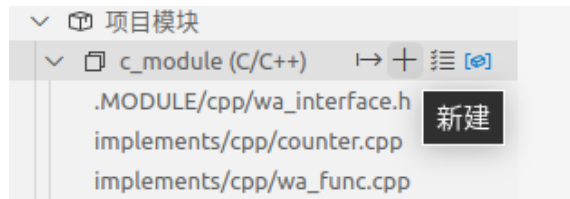
注：IDE 不提供单独的 ST 程序变量的图形编辑操作，用户应需要使用 IEC61131-3 中 ST 的标准语法来定义变量，具体编程参考《WasomePLC 编程指南》

9.3 C/C++代码开发

用户要使用 C/C++开发 PLC 应用，需要先创建一个内部模块，并设定模块的编程语言为 C/C++。

然后在模块内创建任意数量的 C/C++源文件，并通过模块的接口文件实现 PLC 函数或者功能块，提供给主程序(ST 或梯形图)来调用。

如下图，鼠标在项目面板中选择所创建的内部模块，点击“新建”（显示加号图标）按钮来创建一个新的 C/C++源程序。



10 编译

编译与输出

PLC 程序编码完成后，点击左侧面板中的“编译”按钮编译整个项目。WasomePLC IDE 会开启一个下方的终端输出区，显示编译的过程与结果（如下图所示）。



如果没有错误，编译的结果会出现在项目的 build 目录中，包含如下输出文

件与子目录:

| | |
|--------------------|---------------------|
| . | |
| ├── build | |
| │ ├── app.tar.gz | #用于本地安装的 PLC 应用包 |
| │ ├── app.zip | #用于从云上安装部署的 PLC 应用包 |
| │ ├── debug | # 调试版本的编译输出文件 |
| │ └── release | # 发布版本的编译输出文件 |

编译错误定位

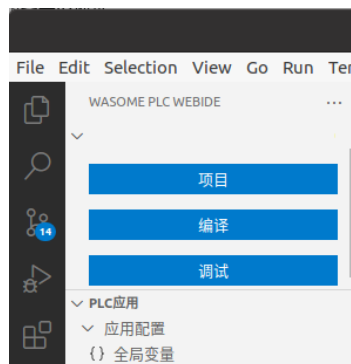
编译过程中，如果程序中有语法或者句法错误，编译器输出的错误信息将被显示在终端窗口中（如下图）。将鼠标移动至每个具体的错误信息上，点击鼠标的同时按下 ctrl 键，将自动定位到程序编辑器中对应的位置。

```
docker run -it -u $(id -u):$(id -g) -w /home/demo/tutorials/projects/ch07-2 -v /home/demo/tutorials/projects/ch07-2:/home/demo/tutorials/projects/ch07-2 -v /home/xin@xin-ubuntu1804:/home/demo/tutorials/projects/ch07-2 -v /home/demo/tutorials/projects/ch07-2:/home/demo/tutorials/projects/ch07-2 -v /home/xin/.iec_compiler/lib:/home/.local/iec_compiler/lib --rm iecc-tools --mode=prj -L /home/.local/iec_compiler/lib --pack /home/demo/tutorials/projects/ch07-2 -o /home/demo/tutorials/projects/ch07-2/build/app
[Err:E2056] missing ';' at 'END_VAR' @ /home/demo/tutorials/projects/ch07-2/PROGRAM/main_prog.st:16:5
16 |           END_VAR
    |           ^
[ProjectBuilder] Failed during release build
```

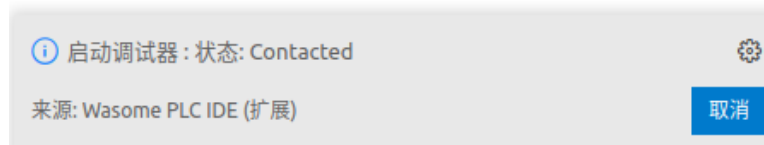
11 调试 PLC 应用

11.1 启动调试

编译成功后，左侧边栏中的“调试”按钮才会高亮。在调试前，请确认已经选择一个在线（图标为绿色）的目标设备为默认调试设备。点击“调试”按钮后：



然后会在 WasomePLC IDE 的右下角，有进入调试的过程及状态，如下



正常情况下，几秒钟后会进入调试状态。如果有错误发生，右下角会提供具体的错误原因。

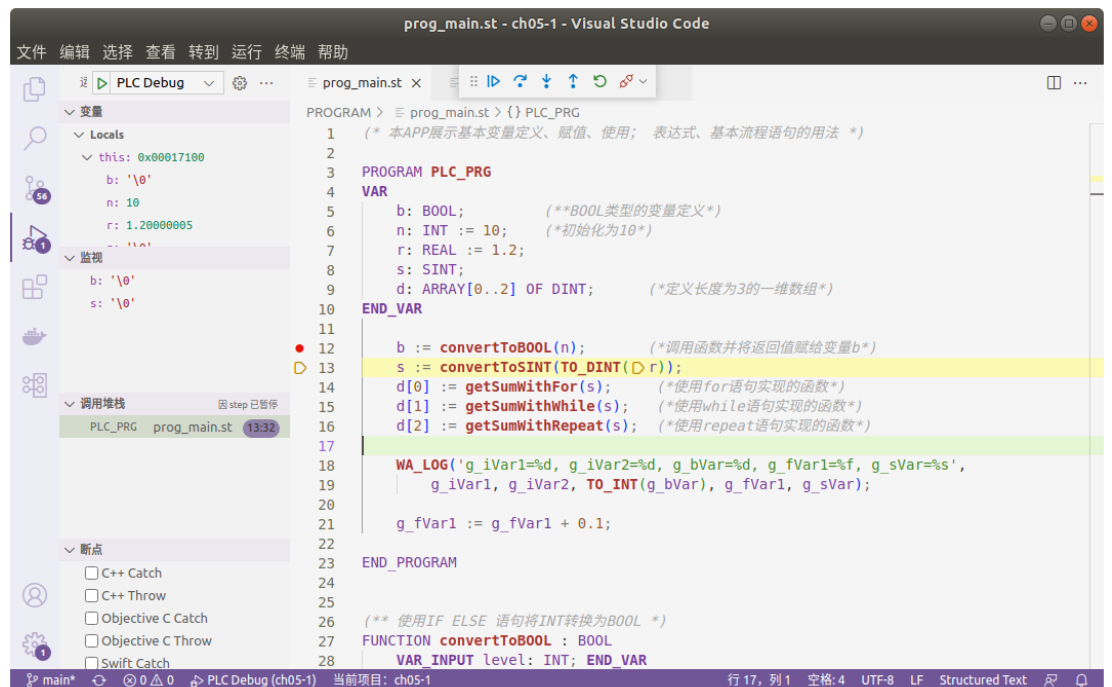
进入调试状态前，可以随时点击右下角的“取消”按钮，取消本次调试。

11.2 进入调试模式

进入调试状态后，WasomePLC IDE 会进入调试界面，在编辑区的正上方会出现调试工具条。



在调试界面，可以打开要调试的文件，设置断点，由于 PLC APP 是周期循环执行，设置断点后程序一般会停在断点处，在左侧的调试窗口，可以观看局部变量、全局变量、调用栈等等。

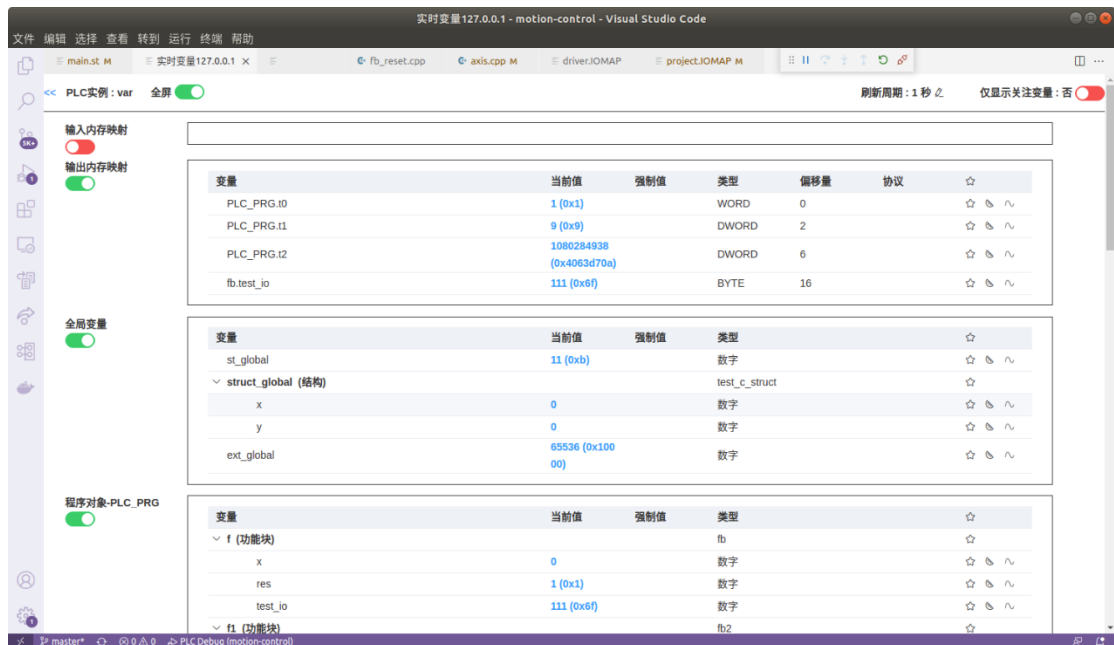


11.3 变量调试与强制

在调试过程中，可以查看变量的实时数值以及设置强制值。当一个目标设备从 IDE 中进入调试状态后，在如下图目标设备条目中会出现中间图标。点击图标打开 APP 的运行变量窗口。



运行变量窗口显示如下：



生产环境中调试变量

在生产环境中通过查看实时变量能帮助快速定位一些运行中碰到的问题，如查看输入内存变量，确定外部输入如某开关的断通状态。或查看输出内存变量，定位某外部输出是否符合 PLC 程序执行预期。

从 WebConsole 的“PLC 设备”列表中进入目标 PLC 的详情页面，从状态区中点击按钮“实时变量”打开运行变量界面。

设置强制值

在上节运行变量窗口中选择一个变量设置强制值，在以下弹出窗口中输入强制值：



设置成功后，可以看到强制值已经生效：

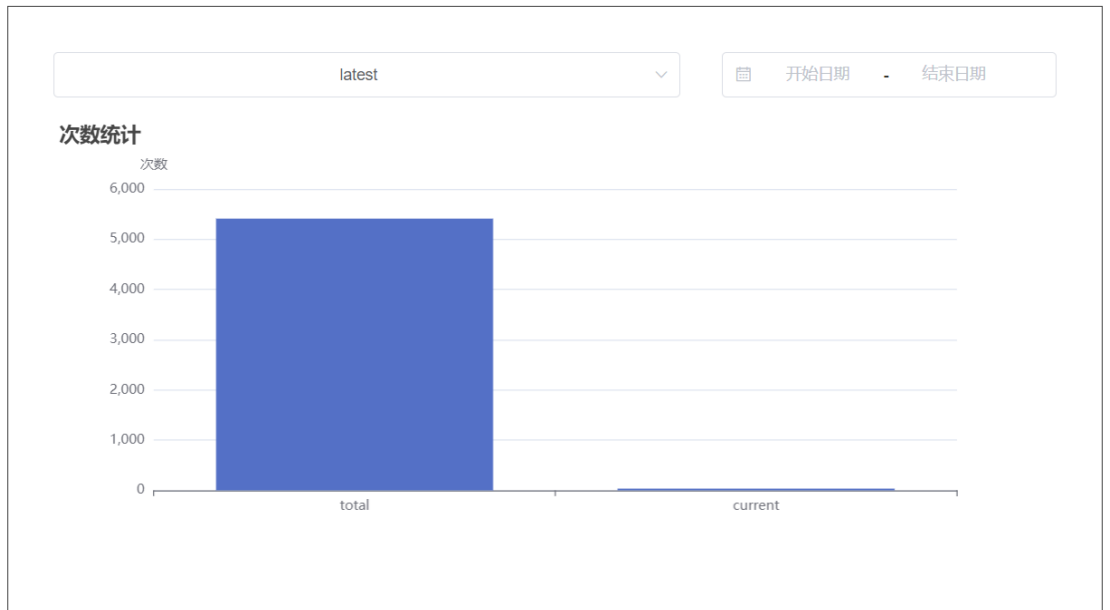
| 变量 | 当前值 | 强制值 | 类型 | |
|----------------------|-----------------|-----|---------------|--------|
| st_global | 11 (0xb) | | 数字 | ☆ 🗑️ ~ |
| ▼ struct_global (结构) | | | test_c_struct | ☆ |
| x | 100 (0x64) | 100 | 数字 | ☆ 🗑️ ~ |
| y | 0 | | 数字 | ☆ 🗑️ ~ |
| ext_global | 65536 (0x10000) | | 数字 | ☆ 🗑️ ~ |

11.4 运行统计

从 WebConsole 中进入一个 PLC 的详情页面，点击按钮“统计”查看本 PLC 的运行统计数据。

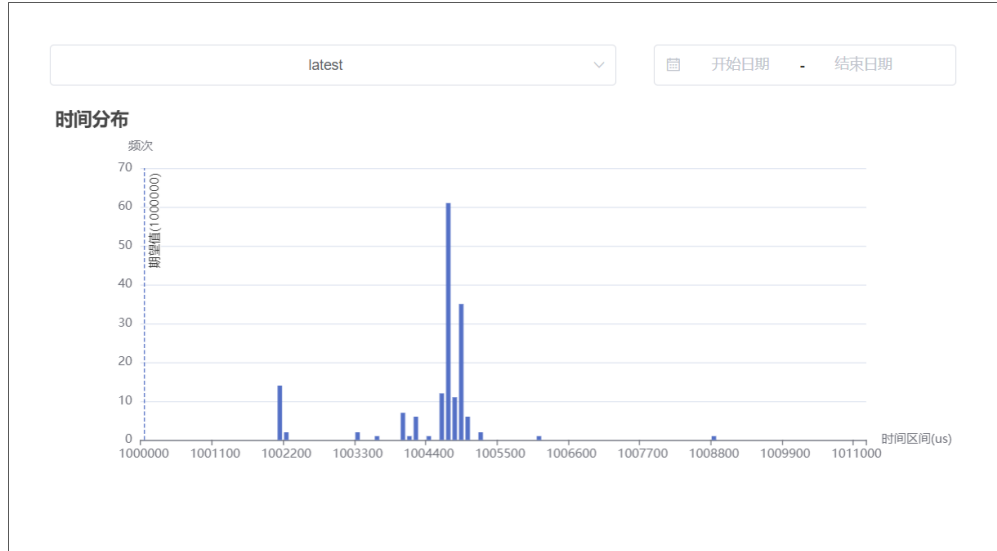
周期执行次数

显示每个任务（Task）自开机总执行次数，以及本次统计周期的执行次数



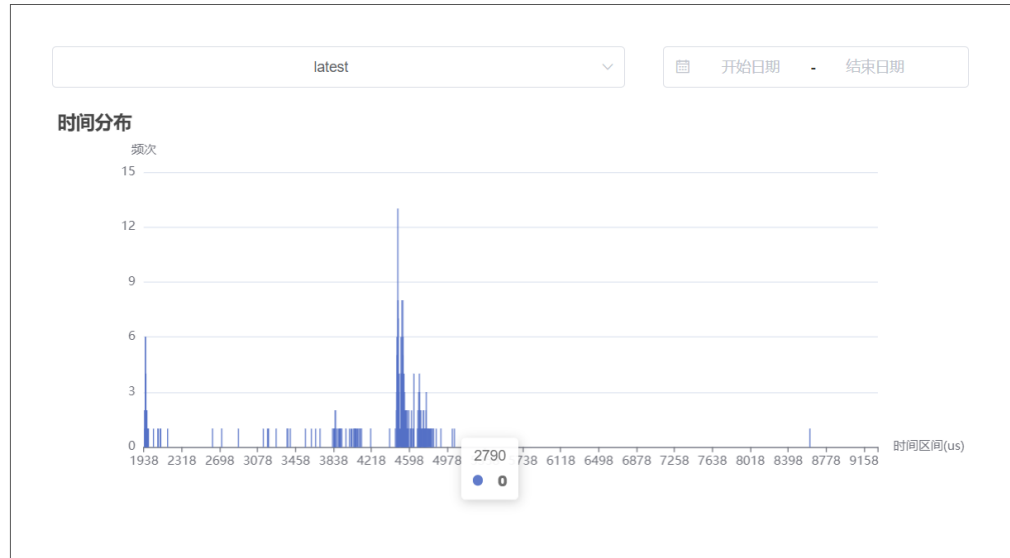
运行周期

显示任务运行周期的时间长度（两次任务开始之间的时间间隔）统计，横轴是时长区间，纵轴是每个区间内总发生次数。



执行时间

显示每次任务运行时间长度（从开始执行任务到完成用时）的统计。横轴是时长区间，纵轴是每个区间内总发生次数。



其他运行统计还包含 IO 操作时长等。

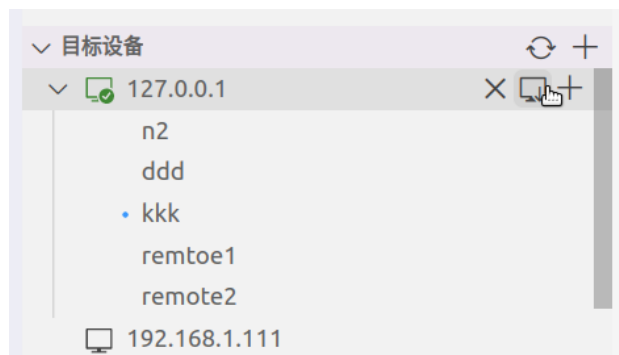
12 部署 PLC APP

在 WebConsole 的章节中，介绍了通过“创建 PLC”来生成 PLC 实例后，然后使用本机上已经部署的 PLC APP 安装包，来装载到目标的 PLC 实例。

在 IDE 开发环境中提供了更加快捷的部署方式，

部署 PLC APP

如下图，在一个目标设备上点击“部署 APP”图标按钮，将当前编译好的 PLC APP 包安装到目标设备的软件库中。以便在生产环境中在 WebConsole 上装载到 PLC 上。



装载 APP

在当前连接的 PLC 上，选择“装载 APP”图标，将把当前开发完成的 APP 装载到目标 PLC 中。

13 PLC 云平台

说明： WasomePLC 无需 PLC 云平台可独立闭环工作。用户如无此使用场景可以略过本章节的内容。

配置云平台

如果要从 WasomePLC IDE 中直接部署 PLC APP 到云端，然后从云端下载到软 PLC 上，则需要设置云端服务器的地址。

进入“当前项目”或“所有 PLC 项目”的界面， 点击按钮“系统设置”， 进入如下设置界面：



配置工业应用商店“服务器 IP 地址”和“登录用户”信息， 然后返回“当前项目”或“所有 PLC 项目”的界面， 点击“登陆云平台”按钮， 登陆成功后即可进入下一步发布 PLC APP。

