

# LogicLab 树莓派快速上手指南

V1.2 2020/09/07

翌控科技

版本修订历史:

|      |            |   |
|------|------------|---|
| V1.0 | 2020/07/28 | 初始版本  |
| V1.1 | 2020/08/21 | 增加基于树莓派 4B 平台 EtherCAT 配置与操作指南                              |
| V1.2 | 2020/09/07 | 增加网页授权激活描述, 树莓派镜像固定 IP 地址说明 (目前树莓派以太网口默认 IP 为 192.168.5.99) |
|      |            |   |
|      |            |   |
|      |            |   |
|      |            |   |

上海翌控科技版权所有

## 目录

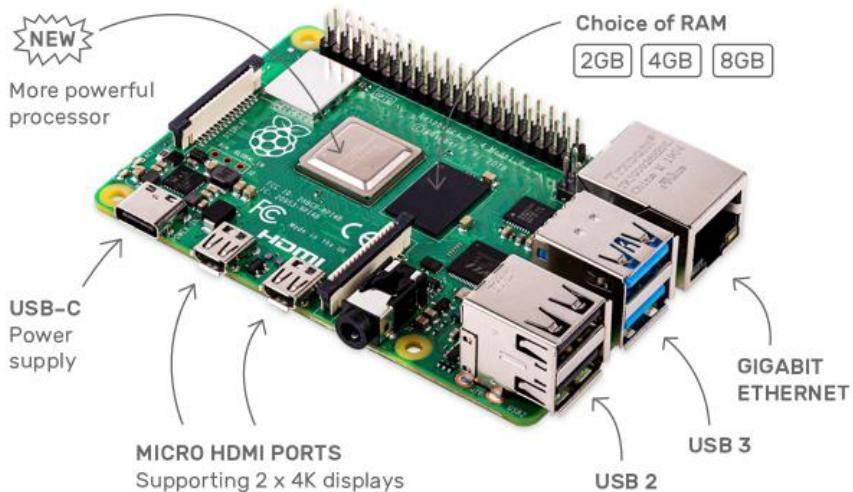
|   |    |
|---|----|
| 1. 初识树莓派.....                               | 6  |
| 2. IEC61131-3 标准软件 LogicLab 解决方案简介.....     | 8  |
| 2. 1. LogicLab 解决方案架构.....                  | 8  |
| 2. 2. LogicLab 编程软件介绍.....                  | 9  |
| 2. 2. 1. IEC61131-3 编程语言.....               | 10 |
| 2. 2. 2. IEC61131-3 功能 (FU) /功能块 (FB) ..... | 12 |
| 2. 2. 3. 基础变量 Variable 类型.....              | 15 |
| 2. 2. 4. 高级类型数组 Array 与结构体 Structure.....   | 16 |
| 2. 2. 5. 扩展数据类型.....                        | 16 |
| 2. 2. 6. 变量地址 Variable Address.....         | 16 |
| 2. 2. 7. 任务 Task 类型.....                    | 17 |
| 2. 2. 8. 工程与库加密.....                        | 18 |
| 2. 2. 9. 在线调试.....                          | 18 |
| 2. 2. 10. LogicLab 支持的启动模式.....             | 23 |
| 2. 2. 11. LogicLab 帮助系统支持.....              | 24 |
| 3. LogicLab 支持的树莓派硬件.....                   | 26 |
| 3. 1. 标准树莓派 3B/3B+/4B.....                  | 27 |
| 3. 2. 欧比特测控 EdgeBox-RPI.....                | 28 |
| 4. 准备工作.....                                | 30 |
| 4. 1. 硬件连接.....                             | 30 |
| 4. 2. 下载翌控科技定制版本的树莓派专用镜像文件.....             | 31 |

|  |    |
|--|----|
| 4. 3. 烧写树莓派专用镜像文件.....                       | 32 |
| 4. 4. 树莓派初次开机与新建工程.....                      | 34 |
| 4. 5. 树莓派硬件授权购买与激活.....                      | 35 |
| 5. 基于树莓派的 LogicLab 入门操作.....                 | 39 |
| 5. 1. 标准树莓派硬件 GPIO 控制.....                   | 39 |
| 5. 2. Modbus TCP.....                        | 41 |
| 5. 3. 1. Modbus TCP 从站.....                  | 41 |
| 5. 3. 2. Modbus TCP 主站.....                  | 44 |
| 5. 3. Modbus RTU.....                        | 49 |
| 5. 3. 1. Modbus RTU 从站.....                  | 49 |
| 5. 3. 2. Modbus RTU 主站.....                  | 50 |
| 5. 4. Node-RED 物联网开发平台.....                  | 52 |
| 5. 4. 1. Node-RED 基础操作入门.....                | 52 |
| 5. 4. 2. LogicLab 与 Node-RED 数据交换.....       | 56 |
| 5. 5. EtherCAT 主站 (仅支持树莓派 4B) .....          | 60 |
| 5. 5. 1. 树莓派 4B WIFI 通讯设置.....               | 60 |
| 5. 5. 2. 树莓派 4B EtherCAT 接口与 PLC 配置文件设置..... | 65 |
| 5. 5. 3. LogicLab 中使用 EtherCAT 主站功能.....     | 66 |
| 6. Q&A.....                                  | 73 |
| 6. 1. LogicLab 编程工具如何下载？如何安装？.....           | 73 |
| 6. 2. 树莓派 PLC 镜像下载地址？.....                   | 74 |
| 6. 3. 如何购买正式版的树莓派 PLC 授权.....                | 74 |

|   |    |
|---|----|
| 6. 4. 树莓派授权激活.....                            | 75 |
| 6. 4. 1. LogicLab 在线激活: .....                 | 76 |
| 6. 4. 2. 网页服务器激活: .....                       | 77 |
| 6. 5. 树莓派 PLC 镜像登录用户名与密码是多少? .....            | 79 |
| 6. 6. 通过 ssh 工具登录树莓派命令行终端.....                | 80 |
| 6. 7. 树莓派设定固定 IP 地址.....                      | 82 |
| 6. 8. 启用树莓派的 WIFI 功能 (树莓派 3B/3B+/4B) .....    | 85 |
| 6. 9. 使用欧比特测控 EdgeBox-RPI 如何开启 4G 功能.....     | 88 |
| 6. 10. 如何自行烧写欧比特测控 EdgeBox-RPI 的树莓派核心板镜像..... | 88 |
| 7. 技术支持与服务.....                               | 90 |

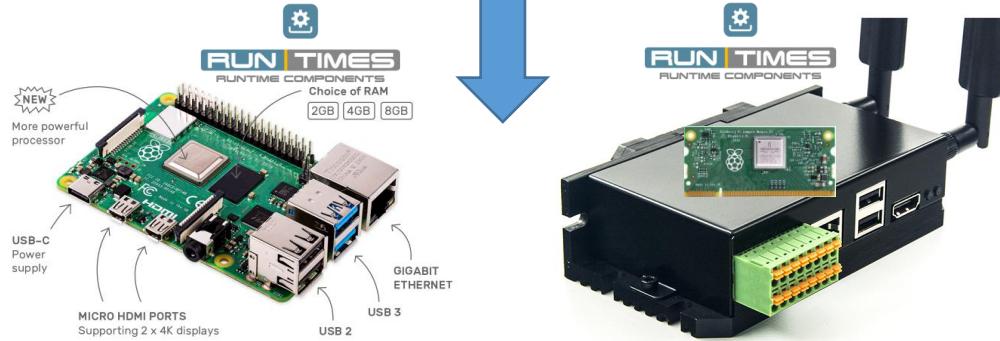
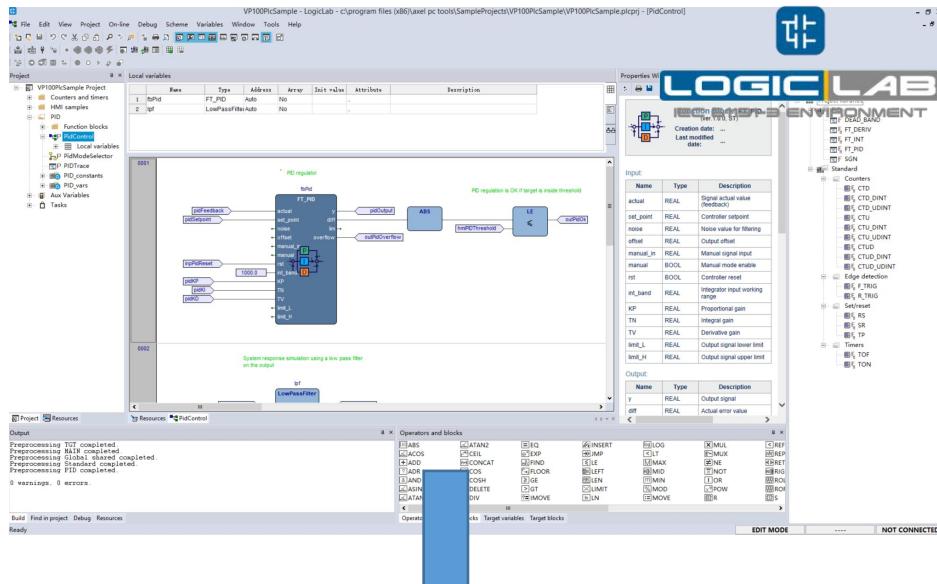
# 1. 初识树莓派

Raspberry Pi(中文名为“树莓派”,简写为 RPi, 只有信用卡大小的微型电脑, 其系统基于 Linux。树莓派基于 ARM 的微型电脑主板, 以 SD/MicroSD 卡为内存硬盘, 卡片主板周围有 USB 接口和一个以太网接口 (A 型没有网口), 可连接键盘、鼠标和网线, 同时拥有视频模拟信号的电视输出接口和 HDMI 高清视频输出接口, 以上部件全部整合在一张仅比信用卡稍大的主板上, 具备所有 PC 的基本功能。



标准的树莓派硬件特别适合 PLC 软件平台的学习与原型研发开发试验, 但是并不适合应用于工业控制领域, 主要是因为标准树莓派硬件应用环境温度不符合工业现场的要求, 并且没有 EMC&RoHS 欧洲标准的认证, 因此选择工业级树莓派核心板 Compute Modules 3+ 实现工业级的控制器设备。

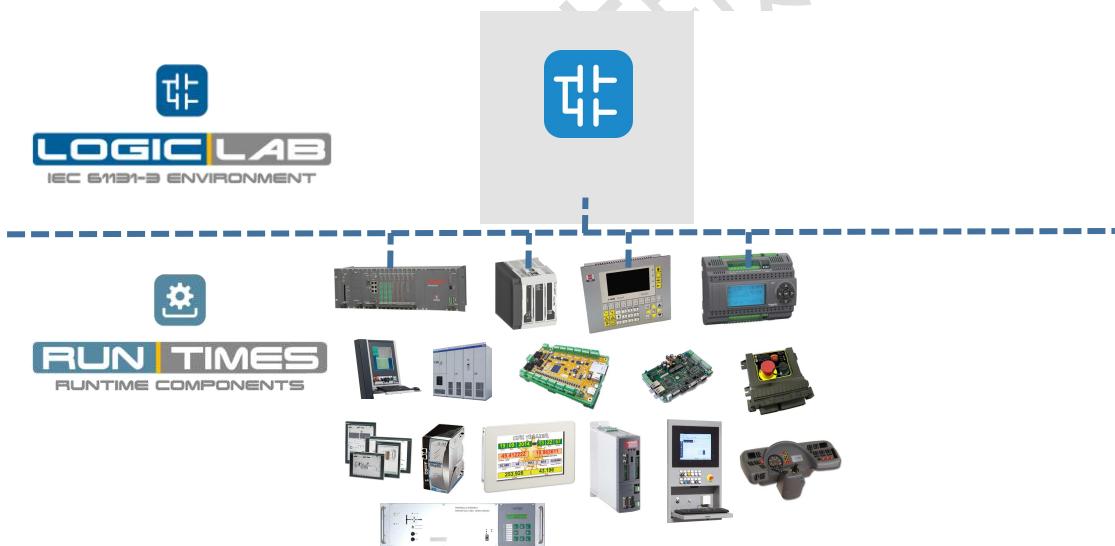
基于标准树莓派硬件或者工业级树莓派核心板 Compute Modules 3+与开放式的 Linux 操作系统，可以在面向工业控制领域应用时增加 SoftPLC 的软件核心，这样可以将标准的树莓派硬件改造成一个开放式的工业 PLC 控制器，例如可以将 LogicLab 运行时核心嵌入到树莓派 Linux 操作系统里，这样就完成一个包含基础功能的工业 PLC 控制器产品：



## 2. IEC61131-3 标准软件 LogicLab 解决方案简介

### 2.1. LogicLab 解决方案架构

LogicLab 解决方案分为运行在 Windows 平台上的编程工具与运行在嵌入式平台上的 Runtime 组件，其中 Runtime 组件可以支持标准 Linux，例如本文介绍的树莓派 Linux 平台：



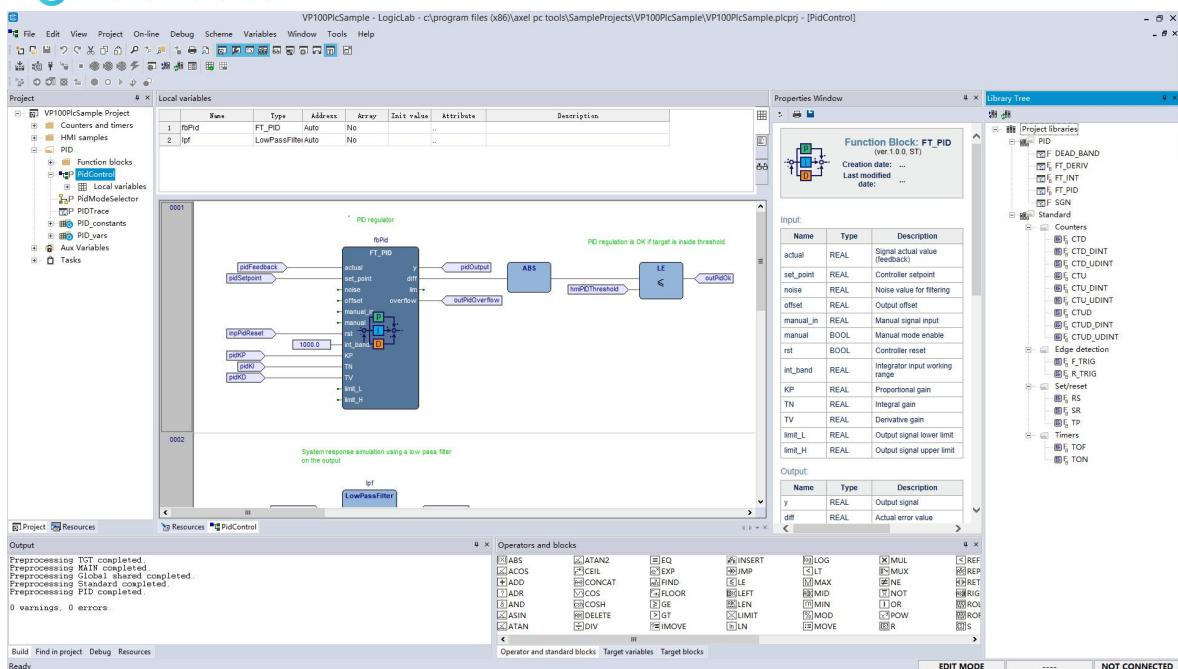
同时 LogicLab Runtime 组件也支持如下处理器与操作系统：

| 目前已经支持的CPU |                        |                    |                   |                     |                 |               |
|------------|------------------------|--------------------|-------------------|---------------------|-----------------|---------------|
| 64 bit     | 32 bit                 |                    |                   | 16 bit              |                 | 8 bit         |
| AMD64      | Intel X86              | Infineon TriCore   | Freescale PowerPC | Freescale DSP56800x | Mitsubishi M16C | Atmel AVR 128 |
|            | ARM (7,9,11)           | Texas TMS320x      | Intel i960        | Infineon C16x       |                 |               |
|            | ARM Cortex (all types) | Freescale ColdFire | Renesas RX core   | Fujitsu 16FX        |                 |               |

| 操作系统  |                    |  |                     |              |                  |            |
|---|--------------------|--|---------------------|--------------|------------------|------------|
|   | 已经支持               |  |                     |              |                  |            |
| Any kind of standard or proprietary OS is supported | (non real time)    | (real time)                            |                     | (embedded)   |                  | Bare metal |
|   | Windows (XP to 10) | IZ RTX32/64 Windows7/10                | VxWorks             | Free RTOS    | OSE              |            |
|   | Linux (from 2.4)   | Windows CE                             | QNX                 | Keil RTX     | Freescale MQX    |            |
|   |                    | Linux R.T. (Xenomai, RTAI, OSADL etc.) | Intime R.T. Windows | Segger EmbOS | Micrium uC/OS-II |            |

## 2. 2. LogicLab 编程软件介绍

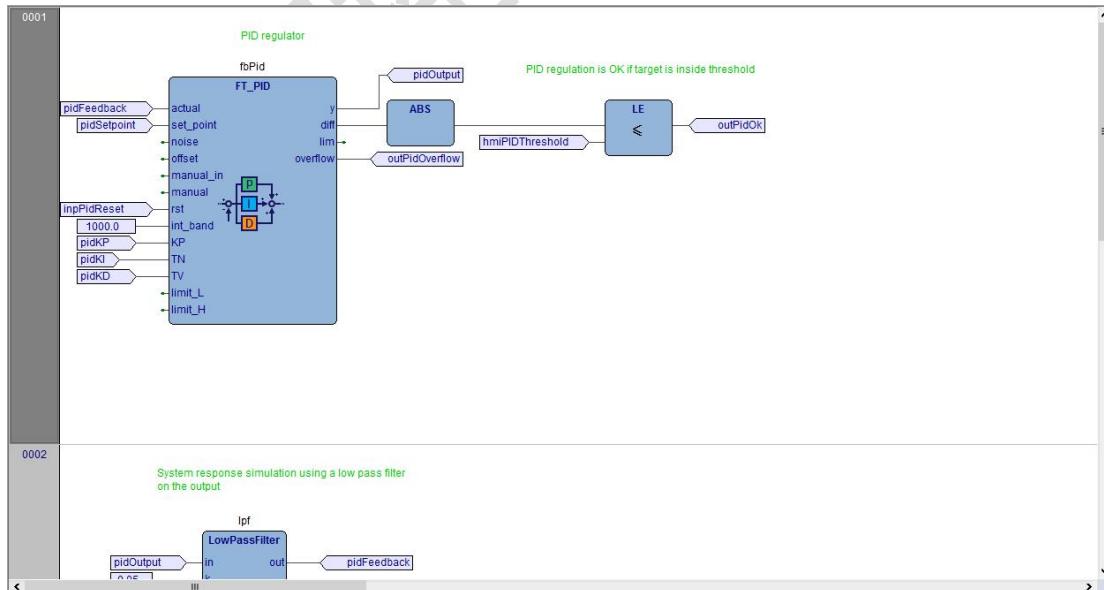
LogicLab 是 IEC61131-3 标准的 SoftPLC 解决方案，支持五种 IEC61131-3 编程语言：功能块图 (FBD) , 梯形图 (LD) , 结构化文本 (ST) , 指令表 (IL) 和顺序流程图 (SFC) 。支持不限数量的多任务架构，任务之间基于优先级进行调度，同时 LogicLab 支持在线断点调试，逻辑分析仪，仿真等等功能，界面如下图所示：



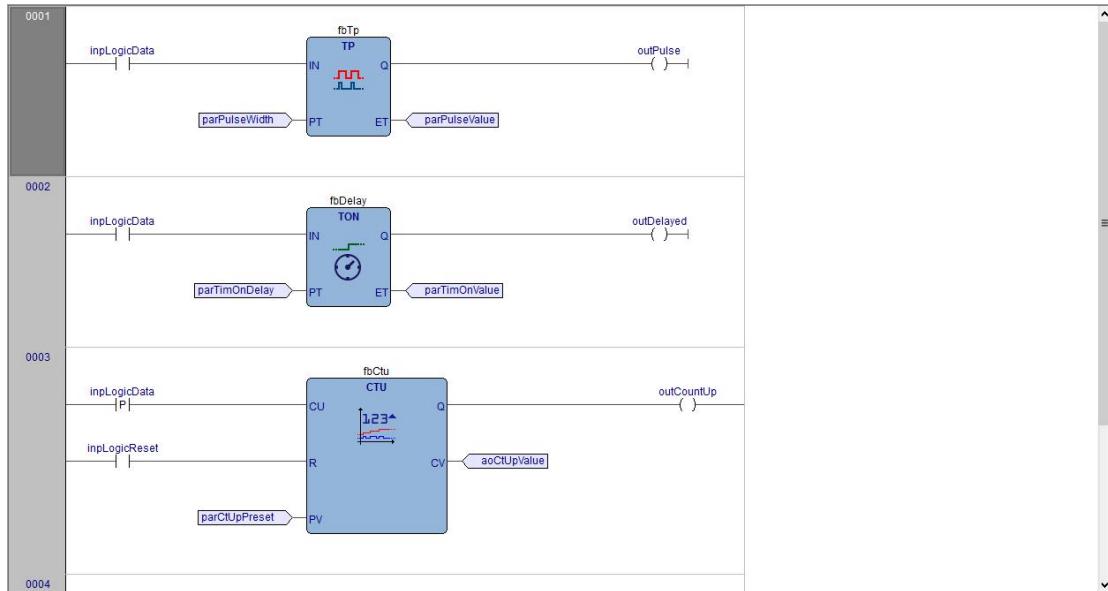
## 2. 2. 1. IEC61131-3 编程语言

LogicLab 支持五种 IEC61131-3 标准中的编程语言，分别如下：

- 功能块图 FBD(Function Block Diagram)



- 梯形图 LD(Ladder)



### - 结构化文本 ST(Structure Text)

```

0001(*
0002version 1.0 2 nov 2007
0003programmer hugo
0004tested by tobias
0005*)
0006DEAD_BAND ist eine lineare Übertragungsfunktion mit Totzone. Die Funktion verschiebt den positiven Teil der Kurve um +L und den negativen Teil der
0007DEAD_BAND = X wenn Abs(X) > L wenn Abs(X) <= L
0008DEAD_BAND = 0 wenn Abs(X) <= L
0009*)
0010
0011IF ABS(x) > l THEN
0012  DEAD_BAND := x - SGN(x) * l;
0013ELSE
0014  DEAD_BAND := 0.0;
0015END_IF;
0016
0017(* revision history
0018  ha 2.11.07 rev 1.0
0019      original version
0020 *)
0021
0022

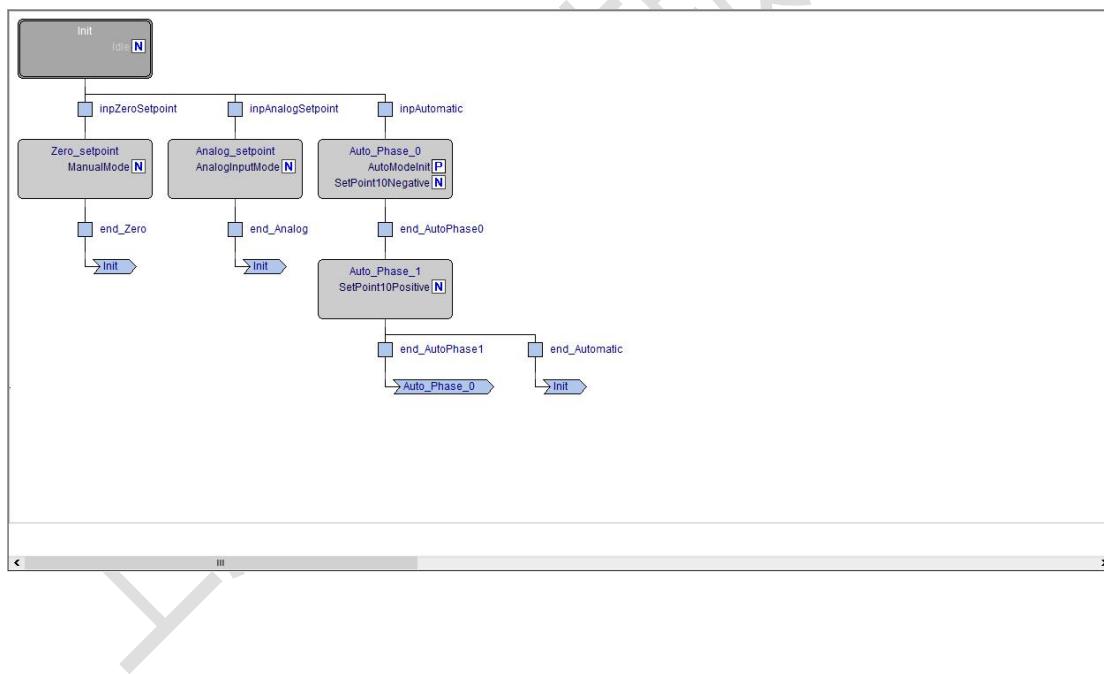
```

### - 指令表 IL(Instruction List)

```

0001      (* Check for reset input *)
0002      LD      ID
0003      JMPCN  10
0004
0005
0006      LD      PV
0007      ST      CV
0008      JMF    11
0009
0010      (* Check for increment enable *)
0011 10:   LD      CD
0012      ANDN   MEMCD
0013      ST      CDEdge
0014
0015
0016      LD      CD
0017      ST      MEMCD
0018
0019      LD      CDEdge
0020      AND(   CV
0021      GT     -32768
0022
0023      JMPCN  11
0024
0025      LD      CV
0026      SUB    1
0027      ST      CV
0028
0029 11:
0030
0031      (* Counter output *)
0032      LD      CV
0033      IIE    0
0034      ST      Q
0035
0036
0037
  
```

- 顺序功能图 SFC(Sequential Function Chart)



## 2. 2. 2. IEC61131-3 功能 (FU) /功能块 (FB)

LogicLab 支持 IEC61131-3 标准中定义的基础功能与功能块，同时可以使用 IEC61131-3 五种编程语言或者是 C/C++（需购买 LogicLab Runtime SDK）封装功能/功能块。

- 基础功能与功能块

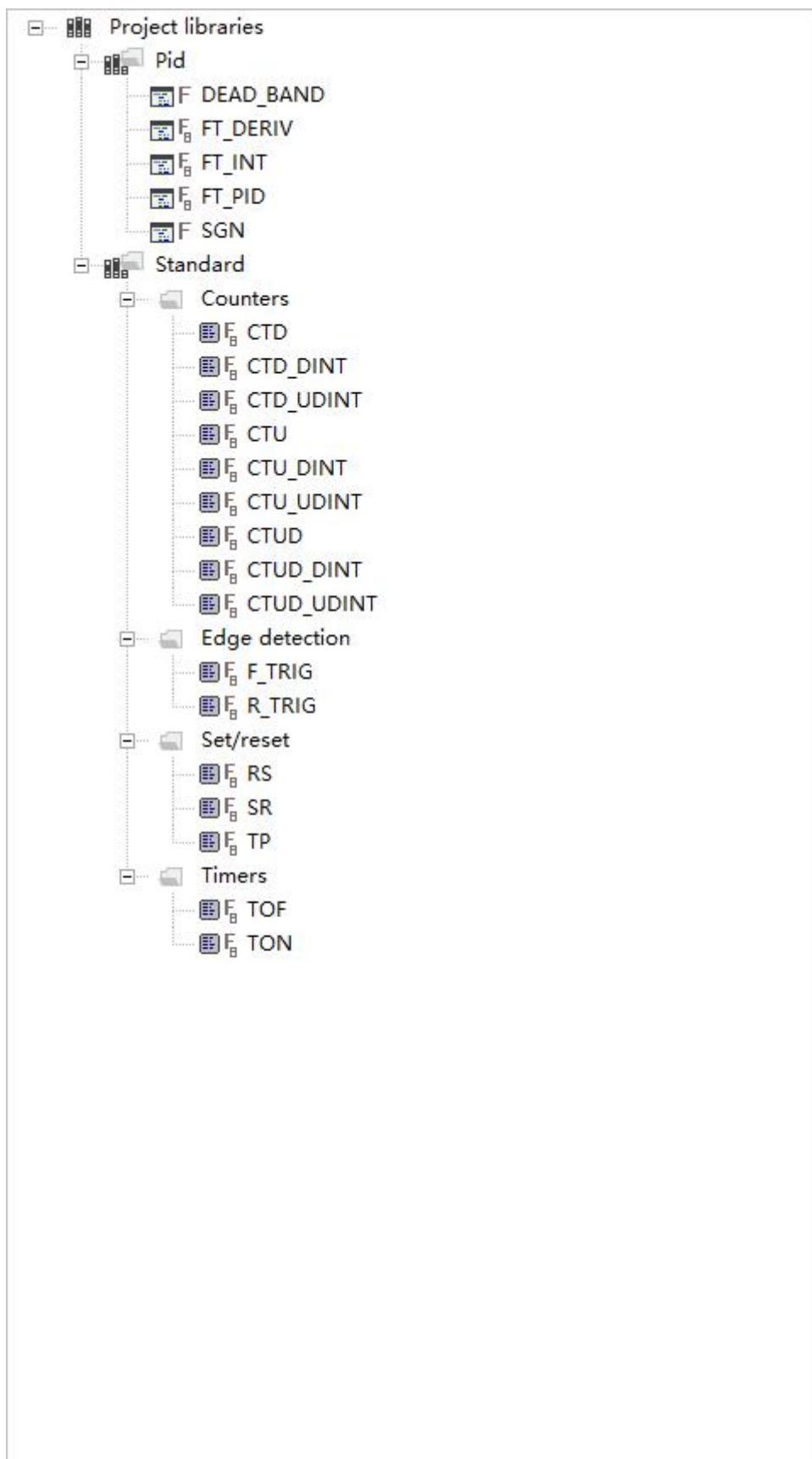
上海市闵行区七莘路 1809 号红点城 8 号楼 6 层 201101 联系方式:13764507241 / 021-64190729

FL6, BUILDING 8, NO1809 QIXIN ROAD, SHANGHAI 201101 +86-021-64190729

第 12 页 共 90 页

| Operators and blocks                       |  |   |   |
|--|--|---|---|
| <input checked="" type="checkbox"/> ABS    | <input checked="" type="checkbox"/> IMOVE  | <input checked="" type="checkbox"/> POW     | <input checked="" type="checkbox"/> TO_BYTE         |
| <input checked="" type="checkbox"/> ACOS   | <input checked="" type="checkbox"/> INSERT | <input checked="" type="checkbox"/> R       | <input checked="" type="checkbox"/> TO_DINT         |
| <input checked="" type="checkbox"/> ADD    | <input checked="" type="checkbox"/> JMP    | <input checked="" type="checkbox"/> REF     | <input checked="" type="checkbox"/> TO_DWORD        |
| <input checked="" type="checkbox"/> ADR    | <input checked="" type="checkbox"/> LE     | <input checked="" type="checkbox"/> REPLACE | <input checked="" type="checkbox"/> TO_INT          |
| <input checked="" type="checkbox"/> AND    | <input checked="" type="checkbox"/> LEFT   | <input checked="" type="checkbox"/> RET     | <input checked="" type="checkbox"/> TO_LINT         |
| <input checked="" type="checkbox"/> ASIN   | <input checked="" type="checkbox"/> LEN    | <input checked="" type="checkbox"/> RIGHT   | <input checked="" type="checkbox"/> TO_LREAL        |
| <input checked="" type="checkbox"/> ATAN   | <input checked="" type="checkbox"/> LIMIT  | <input checked="" type="checkbox"/> ROL     | <input checked="" type="checkbox"/> TO_LWORD        |
| <input checked="" type="checkbox"/> ATAN2  | <input checked="" type="checkbox"/> LN     | <input checked="" type="checkbox"/> ROR     | <input checked="" type="checkbox"/> TO_POINTER      |
| <input checked="" type="checkbox"/> CEIL   | <input checked="" type="checkbox"/> LOG    | <input checked="" type="checkbox"/> S       | <input checked="" type="checkbox"/> TO_REAL         |
| <input checked="" type="checkbox"/> CONCAT | <input checked="" type="checkbox"/> LT     | <input checked="" type="checkbox"/> SEL     | <input checked="" type="checkbox"/> TO_SINT         |
| <input checked="" type="checkbox"/> COS    | <input checked="" type="checkbox"/> MAX    | <input checked="" type="checkbox"/> SHL     | <input checked="" type="checkbox"/> TO_STRING       |
| <input checked="" type="checkbox"/> COSH   | <input checked="" type="checkbox"/> MID    | <input checked="" type="checkbox"/> SHR     | <input checked="" type="checkbox"/> TO_STRINGFORMAT |
| <input checked="" type="checkbox"/> DELETE | <input checked="" type="checkbox"/> MIN    | <input checked="" type="checkbox"/> SIN     | <input checked="" type="checkbox"/> TO_UDINT        |
| <input checked="" type="checkbox"/> DIV    | <input checked="" type="checkbox"/> MOD    | <input checked="" type="checkbox"/> SINH    | <input checked="" type="checkbox"/> TO_UINT         |
| <input checked="" type="checkbox"/> EQ     | <input checked="" type="checkbox"/> MOVE   | <input checked="" type="checkbox"/> SIZEOF  | <input checked="" type="checkbox"/> TO_ULINT        |
| <input checked="" type="checkbox"/> EXP    | <input checked="" type="checkbox"/> MUL    | <input checked="" type="checkbox"/> SQRT    | <input checked="" type="checkbox"/> TO_USINT        |
| <input checked="" type="checkbox"/> FIND   | <input checked="" type="checkbox"/> MUX    | <input checked="" type="checkbox"/> SUB     | <input checked="" type="checkbox"/> TO_WORD         |
| <input checked="" type="checkbox"/> FLOOR  | <input checked="" type="checkbox"/> NE     | <input checked="" type="checkbox"/> TAN     | <input checked="" type="checkbox"/> XOR             |
| <input checked="" type="checkbox"/> GE     | <input checked="" type="checkbox"/> NOT    | <input checked="" type="checkbox"/> TANH    |   |
| <input checked="" type="checkbox"/> GT     | <input checked="" type="checkbox"/> OR     | <input checked="" type="checkbox"/> TO_BOOL |   |

Operator and standard blocks    Target variables

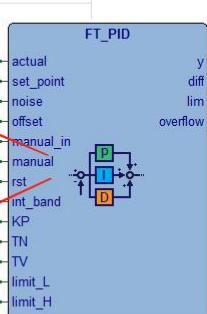


- IEC61131-3 编程语言实现功能/功能块(以 FT\_PID 功能块为例)

Local variables

|    | Class     | Pin | Name      | Type | Array | Init value | Attribute | Description                    |
|----|-----------|-----|-----------|------|-------|------------|-----------|--------------------------------|
| 5  | VAR_INPUT | 0   | actual    | REAL | No    | ..         |           | Signal actual value (feedback) |
| 6  | VAR_INPUT | 1   | set_point | REAL | No    | ..         |           | Controller setpoint            |
| 7  | VAR_INPUT | 2   | noise     | REAL | No    | ..         |           | Noise value for filtering      |
| 8  | VAR_INPUT | 3   | offset    | REAL | No    | ..         |           | Output offset                  |
| 9  | VAR_INPUT | 4   | manual_in | REAL | No    | ..         |           | Manual signal input            |
| 10 | VAR_INPUT | 5   | manual    | BOOL | No    | FALSE      |           | Manual mode enable             |
| 11 | VAR_INPUT | 6   | rst       | BOOL | No    | 100        |           | Controller reset               |
| 12 | VAR_INPUT | 7   | int_band  | REAL | No    | 1.0        |           | Integrator input working range |
| 13 | VAR_INPUT | 8   | KP        | REAL | No    | 1.0        |           | Proportional gain              |
| 14 | VAR_INPUT | 9   | TN        | REAL | No    | 1.0        |           | Integral gain                  |
| 15 | VAR_INPUT | 10  | TV        | REAL | No    | 1.0        |           | Differential gain              |
| 16 | VAR_INPUT | 11  | limit_L   | REAL | No    | -1000      |           | Output signal lower limit      |

接口定义



```

0001 (*
0002 version 1.6 5 jan 2008
0003 programmer hugo
0004 tested by tobias
0005
0006 FT_PID is a pid controller with manual functionality.
0007 The PID controller works according to the formula Y = KP * ( e + 1/TN * INTEG(e) + TV * DERIV(e));
0008 a rst will reset all internal data, while a switch to manual will cause the controller to follow
0009 limit_h and limit_l set the possible output range of Y.
0010 the output flags lim will signal that the output limits are active and overflow will signal that
0011
0012 since rev 1.1 the "trapezregel" is used for more accuracy.
0013 rev 1.2 added selective integratoin which means the integrative component is only active within a
0014 this avoids the integrator to go to limits while an input setpoint change happened and is only c
0015 the int_band is by default 100 which means the int is active all the time and if set to for exa
0016 while the input is between 0.9 and 1.1 of the set_point value.
0017
0018 default values for KP = 1, TN = 1, TV = 1, LIMIT_L = -1000, LIMIT_H = +1000.
0019 *)
0020
0021 IF NOT rst AND NOT manual THEN
0022   (* calculate the process error *)
0023   diff := DEAD_BAND(set_point - actual, noise);
0024   IF ABS(diff) <= int_band AND NOT overflow AND TN > 0.0 THEN
0025     integrator := integrator within int_band and needs to be run */
0026     integ_int := diff, K := 1.0/TN, run := TRUE, rst := FALSE, out_min := int_limit_L, out_max := int_limit_H;
0027   (* check if integrator has reached its limits and set overflow *)
0028   IF integ.out >= int_limit_H OR integ.out <= int_limit_L THEN
  
```

功能块实现

- C/C++编程语言实现功能/功能块

请参阅本文 2.3 章节

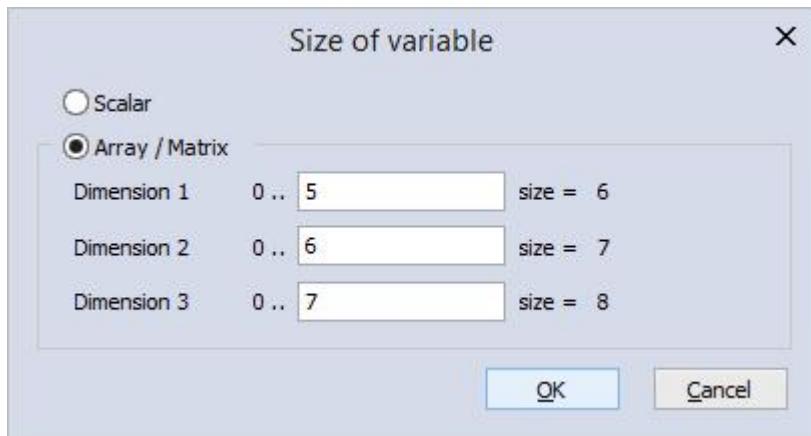
### 2.2.3. 基础变量 Variable 类型

目前 LogicLab 中支持如下基础变量类型：

| 变量类型   | 描述       | 占用存储器空间大小    | 取值范围  |
|--------|----------|--------------|---|
| BOOL   | 逻辑二进制型   | 1Bit         | TRUE或FALSE(1或0)                               |
| BYTE   | 逻辑字节型    | 1Byte        | 16#0 ~ 16#FF(通常用16进制表示, '16#'描述符表示使用16进制代表该值) |
| WORD   | 逻辑字型     | 2Bytes       | 16#0 ~ 16#FFFF                                |
| DWORD  | 逻辑双字型    | 4Bytes       | 16#0 ~ 16#FFFFFF                              |
| SINT   | 8位有符号短整型 | 1Byte        | -128 ~ 127(默认10进制表示整形)                        |
| USINT  | 8位无符号短整型 | 1Byte        | 0 ~ 255                                       |
| INT    | 16位有符号整型 | 2Bytes       | -32768 ~ +32767                               |
| UINT   | 16位无符号整型 | 2Bytes       | 0 ~ +65535                                    |
| DINT   | 32位有符号整型 | 4Bytes       | -2147483648 ~ +2147483647                     |
| UDINT  | 32位无符号整型 | 4Bytes       | 0 ~ +4294967295                               |
| REAL   | 32位浮点型   | 4Bytes       | -3.4E+38 ~ 3.4E+38                            |
| LREAL  | 64位长浮点型  | 8Bytes       | -1.7E-308 ~ 1.7E+308                          |
| STRING | 字符串型     | 4Bytes+字符串长度 | 字符串编码   |

## 2.2.4. 高级类型数组 Array 与结构体 Structure

LogicLab 中支持定义最多三维数组 Array，大小不限（需要硬体存储器空间足够容纳数组以及其他变量的大小）：



LogicLab 中支持定义不限元素数量的结构体 Structure 类型，大小不限（需要硬体存储器空间足够容纳数组以及其他变量的大小）

|   | Name  | Pos. | Type  | Array | Init value |  |
|---|-------|------|-------|-------|------------|--|
| 1 | ELEM1 | 0    | BOOL  | No    |            |  |
| 2 | ELEM2 | 1    | INT   | No    |            |  |
| 3 | ELEM3 | 2    | DINT  | No    |            |  |
| 4 | ELEM4 | 3    | BOOL  | No    |            |  |
| 5 | ELEM5 | 4    | DWORD | No    |            |  |

## 2.2.5. 扩展数据类型

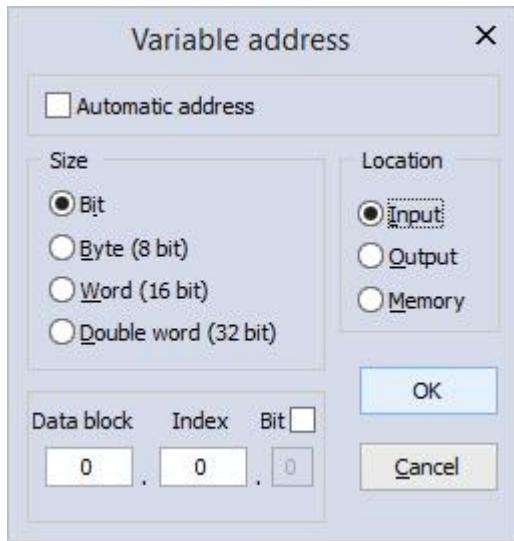
LogicLab 支持扩展的数据类型例如：宏 Macro，枚举 Enumeration，指针 Pointer，类型重定义 TypeDef。

## 2.2.6. 变量地址 Variable Address

LogicLab 支持 IEC61131-3 标准中的地址定义，包括输入 Input、输出 Output 与内部存

储器 Memory, 分别对应地址标识为%I, %Q, %M。在 LogicLab Engineer 中的输入 Input、输出 Output 与内部存储器 Memory 区域对应于 LogicLab Runtime 中一个或者多个 Data Block, 具体实现概念请参阅本文 4.2 章节。

LogicLab 地址定义由对应的窗口来简化用户的使用, 如下图所示:



## 2. 2. 7. 任务 Task 类型

LogicLab 中支持标准 IEC61131-3 中定义的周期类型 (Cyclic) 任务, 事件类型 (Event/Single) 任务, 空闲类型 (Background/Idle) 任务, 并且不限制任务数量。所有任务都会在 LogicLab Runtime 中预先静态定义, LogicLab Engineer 中可以通过设备描述文件来控制具体给用户开放哪些任务, 以及设定其优先级参数, 通过静态定义方式尽可能避免用户对任务误操作引起系统异常, 能够让传统 PLC (不支持多任务的 PLC) 可以尽可能简单切换到 IEC61131-3 的 PLC 系统上来。

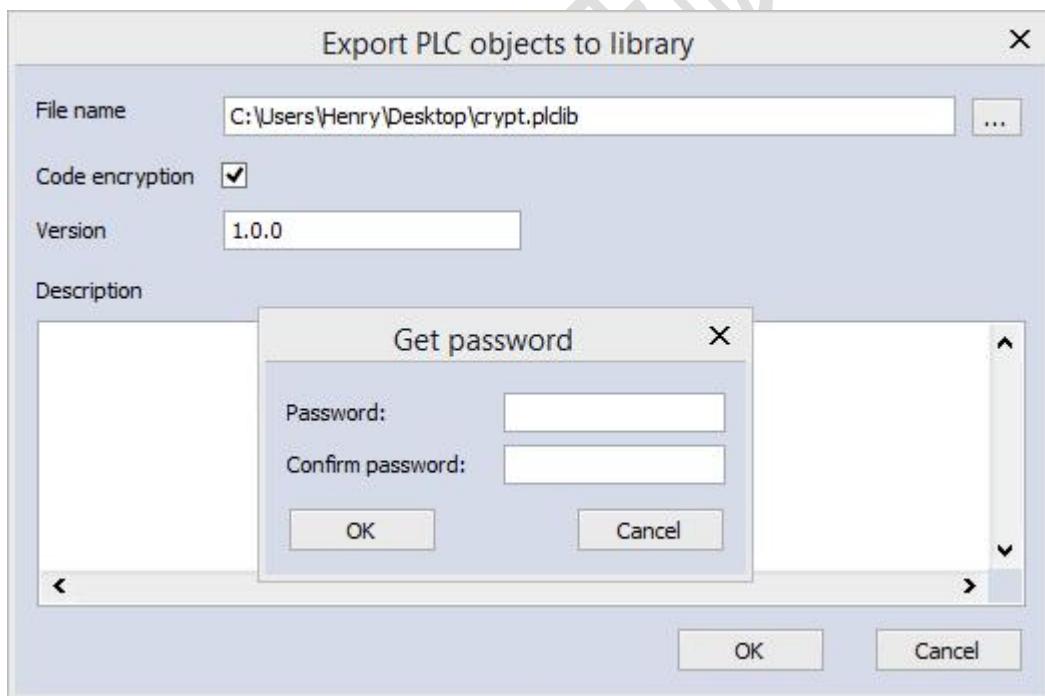
## 2. 2. 8. 工程与库加密

LogicLab Engineer 中的 PLC 工程元素（包括程序，功能，功能块等）可以进行源代码加密功能，尽可能保护应用工程师开发的行业应用算法与源代码资产。

单独加密 POU：



或者导出 POU 加密：



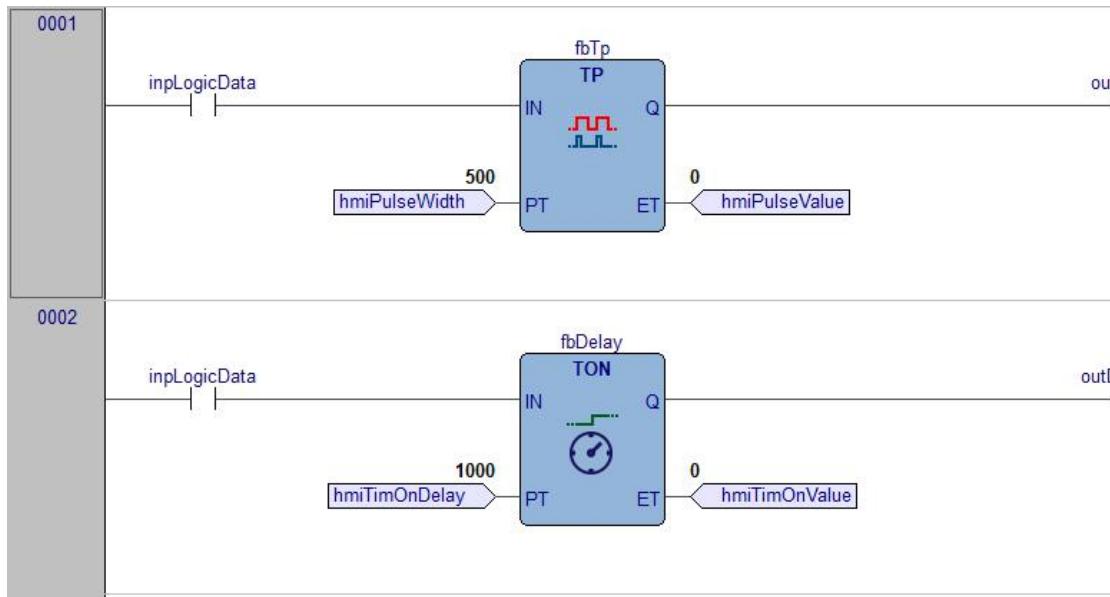
## 2. 2. 9. 在线调试

LogicLab 提供一系列的调试手段方便用户进行工程应用开发，包括：在线数据实时显示、

强制修改改变量值、离散/集中数据监控、断点 BreakPoint 与单步 Step 调试、专业级软件示

波器与 Runtime 状态监控等等功能。

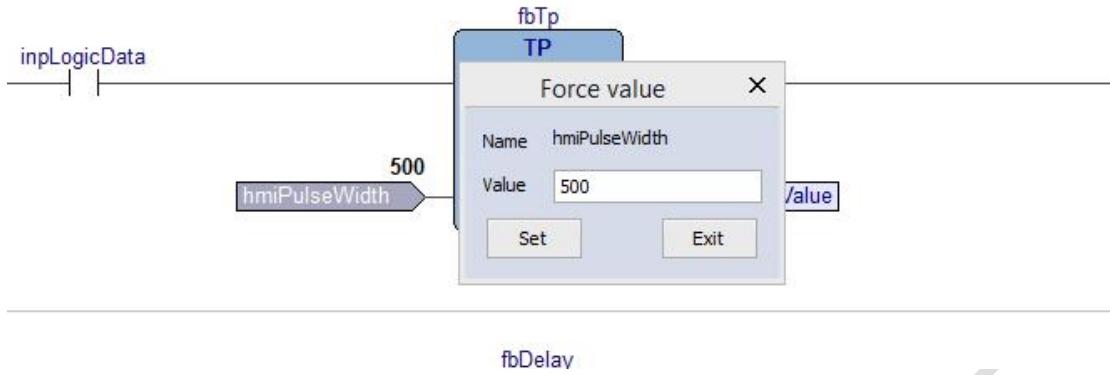
- 在线数据实时显示 (以 LD 与 ST 为例)



```

0001 (* Length of acquisition *)
0002 tracePIDLen 150 := 150;
0003
0004
0005 (* Manage start request *)
0006 IF traceStartAcq FALSE THEN
0007   n 0 := 0;
0008   traceStartAcq FALSE := FALSE;
0009   runAcq FALSE := TRUE;
0010 END_IF;
0011
0012 (* Acquire data *)
0013 IF n 0 < 150 AND runAcq FALSE THEN
0014
0015   tracePIDSetpoint[ n ] 0 := pidSetpoint 0;
0016   tracePIDFeedback[ n ] 0 := pidFeedback 0;
0017   n 0 := n 0 + 1;
0018
0019   traceTrigger FALSE := n 0 = 150;
0020   runAcq FALSE := NOT traceTrigger FALSE;
0021
0022 END_IF;
0023
0024
0025
0026
0027
  
```

- 强制修改改变量值



- 离散/集中数据监控

离散数据监控是指在变量列表中监控已经定义的变量，集中数据监控是指用户添加自己需要监控的变量到 Watch Window 中，在独立的窗口中查看当前数据值。

离散数据（适用于局部变量与全局变量表格）监控如下图所示：

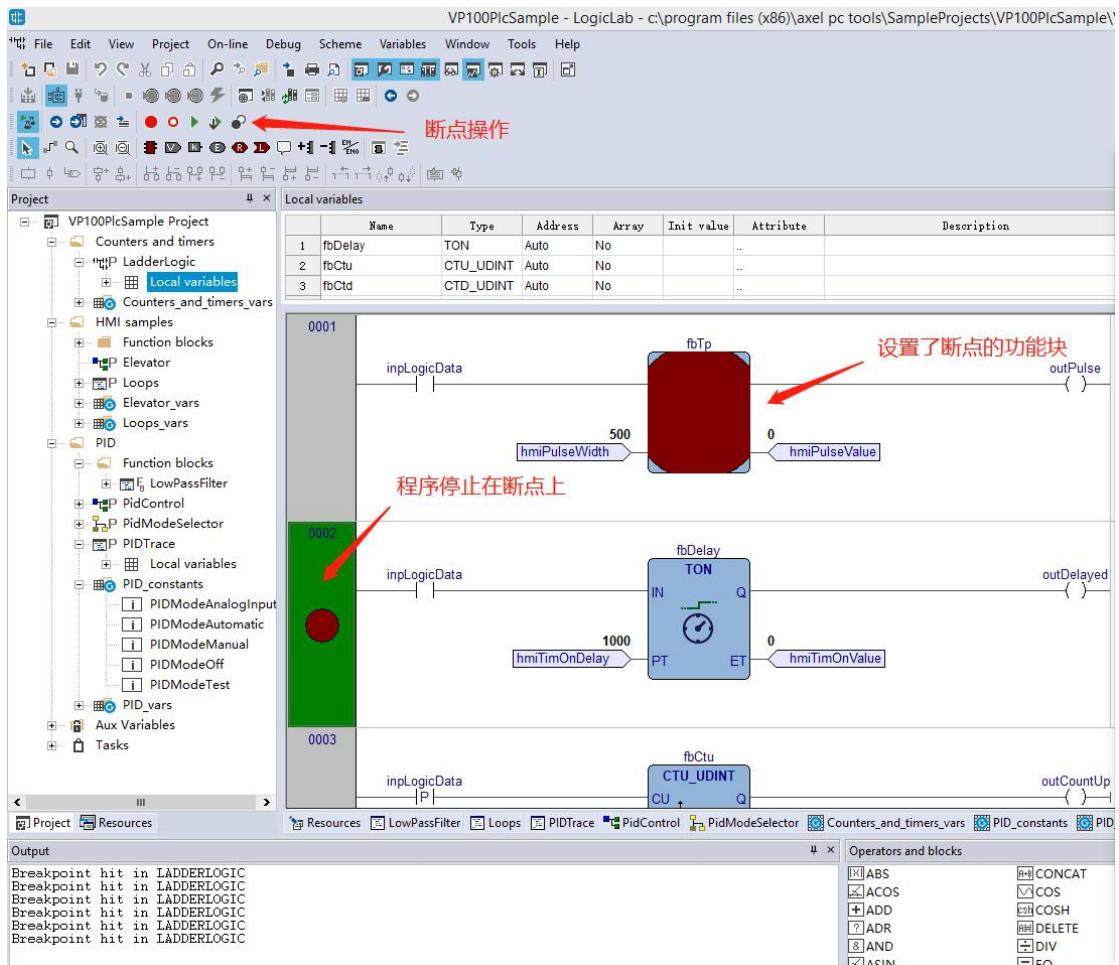


| Local variables |       | 可直接监控到FB内部的数据 |                   |                          | 变量列表直接监控 |  |
|-----------------|-------|---------------|-------------------|--------------------------|----------|--|
| Symbol          | Value | Type          | Location          | Description              |          |  |
| FBDELAY         | -     | TON           | @BACKGROUND:LA... | Timer input source       |          |  |
| IN              | FALSE | BOOL          | @BACKGROUND:LA... | Preset time value (ms)   |          |  |
| PT              | 1000  | UDINT         | @BACKGROUND:LA... |                          |          |  |
| Q               | FALSE | BOOL          | @BACKGROUND:LA... | Timer output             |          |  |
| ET              | 0     | UDINT         | @BACKGROUND:LA... | Timer current value (ms) |          |  |
| FBCTU           | -     | CTU_UDINT     | @BACKGROUND:LA... | Count-up on input ri...  |          |  |
| CU              | FALSE | BOOL          | @BACKGROUND:LA... |                          |          |  |
| R               | FALSE | BOOL          | @BACKGROUND:LA... | Reset counter input      |          |  |
| PV              | 15    | UDINT         | @BACKGROUND:LA... | Preset value             |          |  |

集中数据监控如下图所示的 Watch Windows 显示：

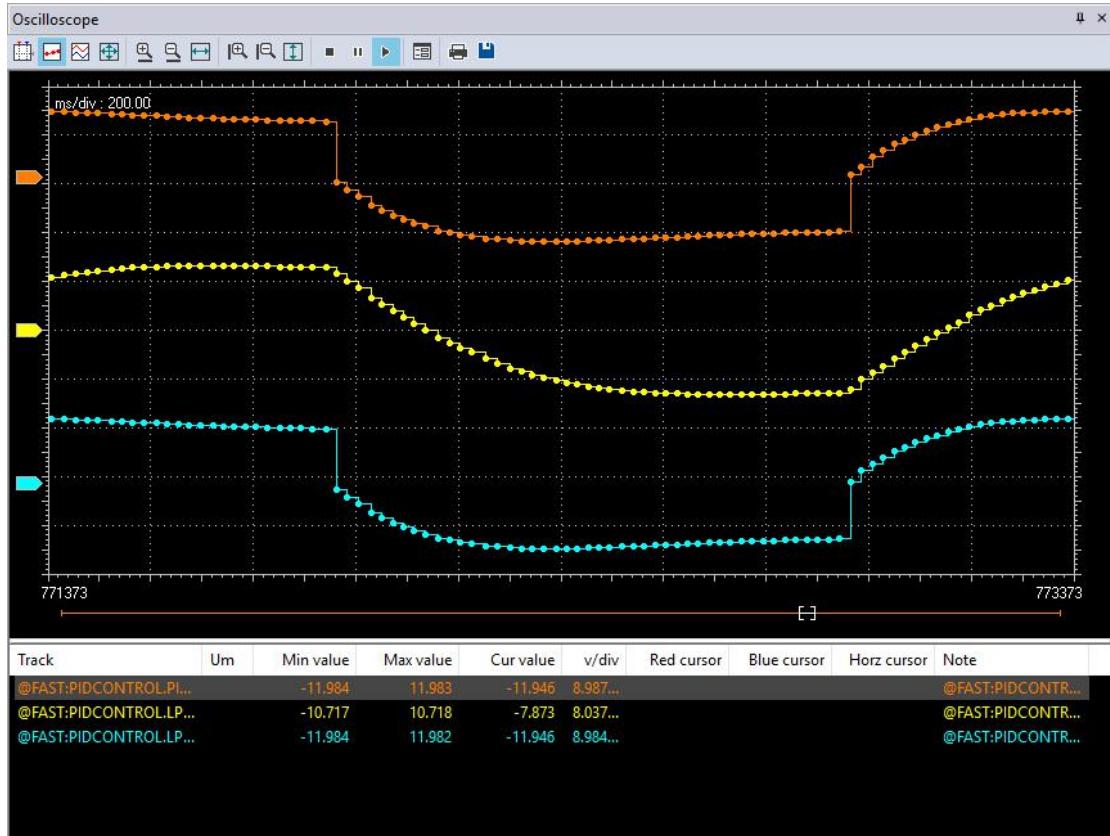
| Watch           |           |        |                  |                                |
|-----------------|-----------|--------|------------------|--------------------------------|
| Symbol          | Value     | Type   | Location         | Description                    |
| — PARCTD...     | 100       | INT    | global           | Counter down preset            |
| — PARPULSEWIDTH | 500       | UDINT  | global           | Pulse width                    |
| — PARTIMONVALUE | 0         | UDINT  | global           | Actual value of the timer      |
| LPF.MBS         | -         | TON    | @FAST:PIDCONTROL |                                |
| IN              | FALSE     | BOOL   | @FAST:PIDCONTROL | Timer input source             |
| PT              | 0         | UDINT  | @FAST:PIDCONTROL | Preset time value (ms)         |
| Q               | FALSE     | BOOL   | @FAST:PIDCONTROL | Timer output                   |
| ET              | 0         | UDINT  | @FAST:PIDCONTROL | Timer current value (ms)       |
| LPF             | -         | LO...  | @FAST:PIDCONTROL |                                |
| IN              | 10.375    | REAL   | @FAST:PIDCONTROL |                                |
| K               | 0.050     | REAL   | @FAST:PIDCONTROL |                                |
| OUT             | 10.618    | REAL   | @FAST:PIDCONTROL |                                |
| FBPID           | -         | FT_PID | @FAST:PIDCONTROL |                                |
| ▼ ACTUAL        | 10.631    | REAL   | @FAST:PIDCONTROL | Signal actual value (feedback) |
| SET_POINT       | 10.000    | REAL   | @FAST:PIDCONTROL | Controller setpoint            |
| NOISE           | 0.000     | REAL   | @FAST:PIDCONTROL | Noise value for filtering      |
| OFFSET          | 0.000     | REAL   | @FAST:PIDCONTROL | Output offset                  |
| MANUAL_IN       | 0.000     | REAL   | @FAST:PIDCONTROL | Manual signal input            |
| MANUAL          | FALSE     | BOOL   | @FAST:PIDCONTROL | Manual mode enable             |
| RST             | FALSE     | BOOL   | @FAST:PIDCONTROL | Controller reset               |
| INT_BAND        | 1000.000  | REAL   | @FAST:PIDCONTROL | Integrator input working range |
| KP              | 0.500     | REAL   | @FAST:PIDCONTROL | Proportional gain              |
| TN              | 0.100     | REAL   | @FAST:PIDCONTROL | Integral gain                  |
| TV              | 0.000     | REAL   | @FAST:PIDCONTROL | Derivative gain                |
| LIMIT_L         | -1000.000 | REAL   | @FAST:PIDCONTROL | Output signal lower limit      |
| LIMIT_H         | 1000.000  | REAL   | @FAST:PIDCONTROL | Output signal upper limit      |
| ▼ Y             | 10.375    | REAL   | @FAST:PIDCONTROL | Output signal                  |
| ▲ DIFF          | -0.631    | REAL   | @FAST:PIDCONTROL | Actual error value             |
| LIM             | FALSE     | BOOL   | @FAST:PIDCONTROL | In-limit output status         |
| OVERFLOW        | FALSE     | BOOL   | @FAST:PIDCONTROL | Integral overflow status       |

- 断点 BreakPoint 与单步 Step 调试

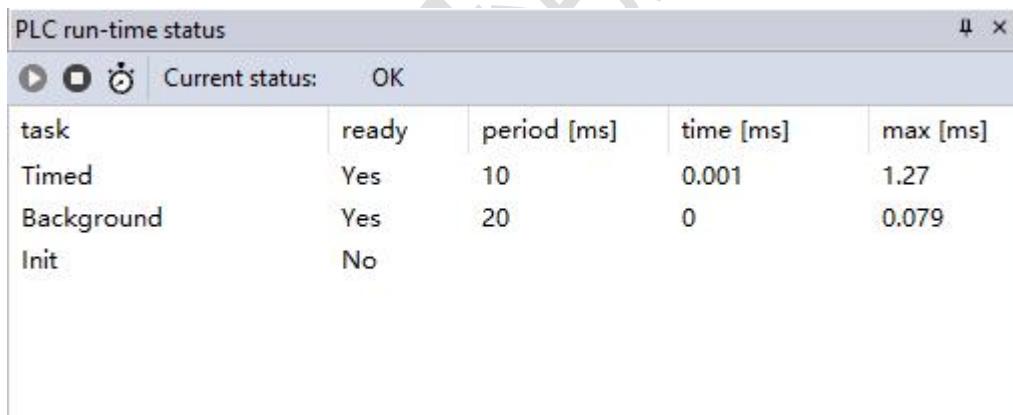


### - 专业级软件示波器 Oscilloscope

应用于伺服驱动器上的专业示波器组件，默认采集 1ms 分辨率的数据，定制版本可达到 100us 采集的频率（主要应用于伺服驱动器 Servo Drive 调试诊断工具中）。



- Runtime 状态监控



| task       | ready | period [ms] | time [ms] | max [ms] |
|------------|-------|-------------|-----------|----------|
| Timed      | Yes   | 10          | 0.001     | 1.27     |
| Background | Yes   | 20          | 0         | 0.079    |
| Init       | No    |             |           |          |

## 2. 2. 10. LogicLab 支持的启动模式

LogicLab 支持的启动模式由冷(Cold)、暖(Warm)与热(Hot)启动，分别含义如下：

- 冷启动(Cold Start)

执行逻辑代码前所有变量值恢复为初始值（如若未设定初始值，则其默认值为 0）。

- 暖启动(Warm Start)

勾选了 RETAIN 属性的变量值保持其上次断电或者程序停止时的值，其他变量值恢复为初始值（如若未设定初始值，则其默认值为 0）。PLC 上电默认执行暖启动，如果暖启动失败则进行冷启动。

- 热启动(Hot Start)

仅仅在调试模式下停止当前程序执行后，才可以执行热启动，所有的变量值维持不变，继续以当前的变量值参与逻辑运算。

## 2. 2. 11. LogicLab 帮助系统支持

- 功能/功能块帮助系统

在 LogicLab 中功能/功能块帮助系统可以直接通过选中功能/功能块后的右键菜单“Object Properties”查看，LogicLab 的 Properties Window 会直接显示该功能/功能块的帮助。PLC 开发工程师也可以将自己的功能/功能块说明通过文本或者 HTML 添加至 Property 中，用户可以直接通过同样的方式查看到由 PLC 厂商开发的功能/功能块说明，如下图所示：

Properties Window

Function Block: CTD (ver.1.0.0, IL)

Creation date: ...  
 Last modified date: ...

Down counter

**Input:**

| Name | Type | Description                     |
|------|------|---------------------------------|
| CD   | BOOL | Count-down on input rising edge |
| LD   | BOOL | Preset counter input            |
| PV   | INT  | Preset value                    |

**Output:**

| Name | Type | Description           |
|------|------|-----------------------|
| Q    | BOOL | Count-down output     |
| CV   | INT  | Counter current value |

**Description:**

The CTD block counts down for each FALSE-to-TRUE transition of the input CD and sets the output Q when the accumulated value (CV) reaches the 0 value.



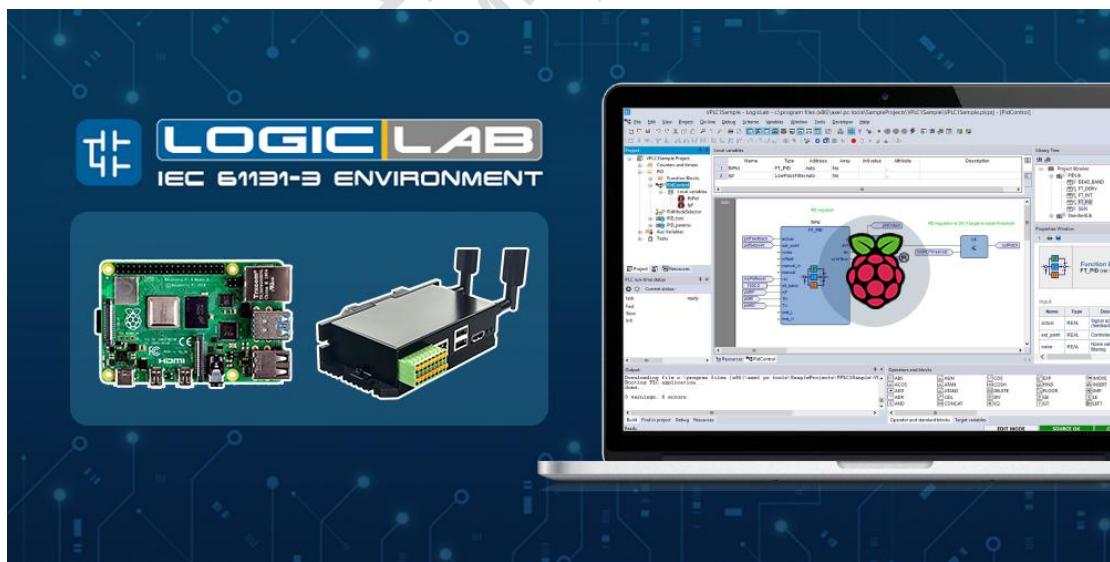
- LogicLab 使用手册帮助

LogicLab 使用手册支持英文与简体中文翻译版本, 简体中文翻译版本由 Nxtrol 持续更新维护。

### 3. LogicLab 支持的树莓派硬件

树莓派开放式社区 ([www.raspberrypi.org](http://www.raspberrypi.org)) 为每一位嵌入式技术爱好者提供了一个非常低成本高性能的开放式平台，可以自由实现自己的创意，在教育和创新领域打造了全新的技术社交，融入创客们的理想和态度，形成了一种非常可贵的技术情怀。官方社区甚至也提供了用于工业级产品设计的 Compute Module 3B+核心板，从而可以将树莓派应用于环境更加复杂的工业控制领域。

翌控科技基于 LogicLab 软件平台提供支持树莓派 3B/3B+/4B 与欧比特测控 EdgeBox-RPI 硬件的 IEC61131-3 PLC 解决方案，为了更好实现 PLC 的实时控制，目前已经在标准系统镜像软件中集成了 Linux Preempt-RT 实时补丁：



并且除了树莓派上传统的 C/C++ 编程、Python 等等编程语言以外，LogicLab 还可以实现如下功能：

- 支持主流 PLC 符合 IEC61131-3 标准五种编程语言 FBD、LD、ST、SFC、IL 进行 PLC 编

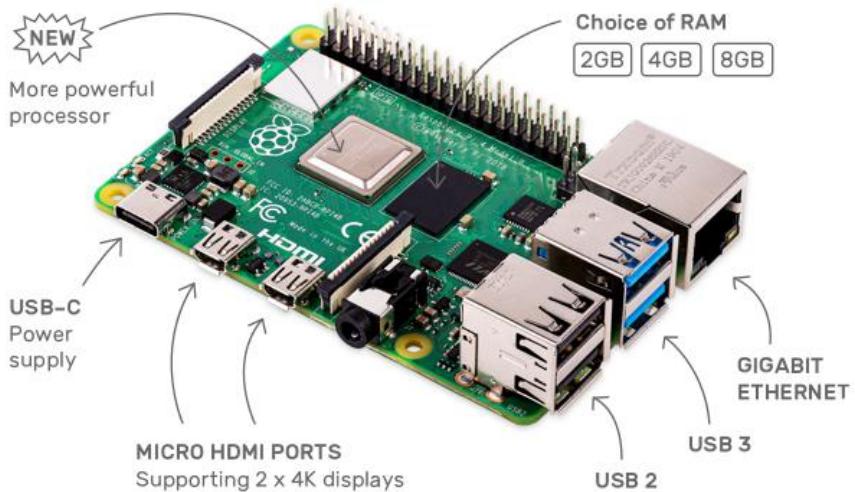
程与调试

- Modbus RTU/TCP 主从站通讯
- CANopen 主站通讯
- NodeRED 物联网开放式平台
- EtherCAT 主站通讯 (\*由于硬件架构限制，EtherCAT 主站功能仅支持树莓派 4B 平台)
- LogicLab PLC 内核中的功能可以通过开放式 SDK 进行二次扩展开发，通过 C 语言实现用户自己的功能、功能块、I/O 驱动、共享内存等等

### 3. 1. 标准树莓派 3B/3B+/4B

Raspberry Pi 极致的开源硬件价格，让全世界无数的技术爱好者有机会在这个平台上面去实现自己的想法和创意，在教育和创新领域打造了全新的技术社交，融入创客们的理想和态度，形成了一种非常可贵的技术情怀。

对于应用于工业现场的 PLC 产品来说，虽然标准的树莓派是无法在工业现场稳定可靠运行，但是作为学习与开发平台来说，已经绰绰有余。同时标准树莓派硬件价格相对较低，非常容易购买，并且外部扩展模块众多，非常适合用于学习、初期研发、搭建原型系统。



### 3. 2. 欧比特测控 EdgeBox-RPI

EdgeBox-RPI 是由欧比特测控公司基于树莓派工业级核心板 Compute Module 3+研发的一款应用于边缘计算控制器，它的设计便于安装和快速部署典型的工业应用，如状态监测，设施管理，数字标志和公用事业的远程控制。具有 4 核心高性能 ARM Cortex-A53 处理器，主频高达 1.2GHZ。其超轻巧紧凑的设计是在空间受限环境中应用的答案，确保了它可以在包括车载应用在内的各种极端环境中可靠运行。



注意：为了确保正确应用翌控科技提供的 IEC61131-3 解决方案，请选用内置 32GB 存储器的 EdgeBox-RPI 硬件。

购买此硬件，请向欧比特测控要求默认出厂烧写好 LogicLab 最新镜像，详情请查看欧比特测控官网与淘宝店：

<https://www.openembed.com/products/51.html>

<https://item.taobao.com/item.htm?spm=a230r.1.14.66.30ac530evWFYSE&id=603991102464&ns=1&abbucket=6#detail>

关于更多使用心得请参考 B 站上的相关视频：

<https://www.bilibili.com/video/BV1DT4y1G73V>

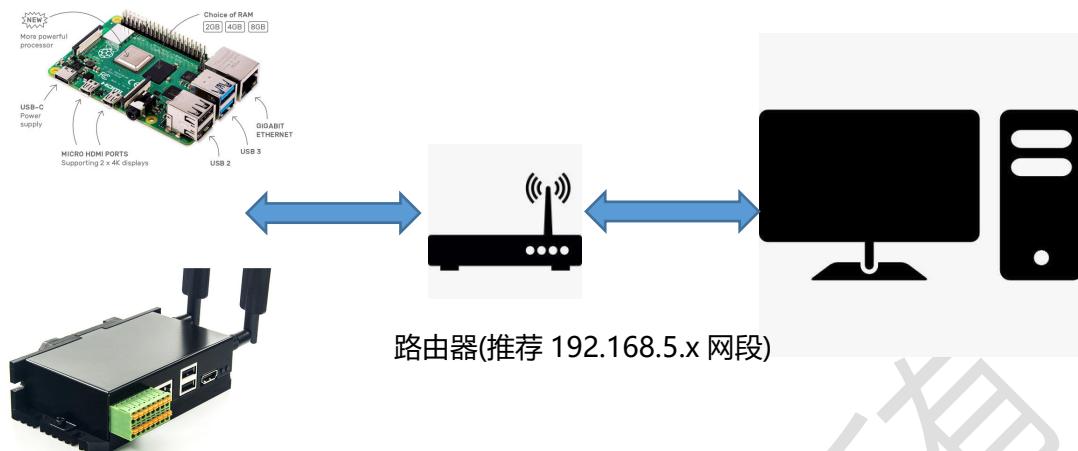
## 4. 准备工作

无论是单独购买标准树莓派硬件，还是 EdgeBox-RPI 工业级产品，都需要使用由翌控科技提供的完整树莓派系统镜像才可以使用符合 IEC61131-3 标准的 LogicLab 软件解决方案，初次安装、调试、配置时需要准备如下硬件资源：

- 安装了 Windows 系统的 PC (推荐安装 Win10 操作系统)
- 路由器 (用于树莓派硬件自动获取 IP 地址或者使用路由器网段静态 IP 进行连接)
- 标准树莓派 3B/3B+/4B 硬件或者欧比特测控的 EdgeBox-RPI 硬件

### 4. 1. 硬件连接

准备好对应的硬件资源，需要将 PC 以及树莓派硬件连接至同一个路由器上，从而保证 PC 与树莓派的网络在同一个网段内，并且路由器会给树莓派自动分配 IP 地址，如下图所示：



## 4. 2. 下载翌控科技定制版本的树莓派专用镜像文件

登录翌控科技官网下载专区: <http://www.nxtrol.com/download.html>, 下载对应树莓派硬件平台的镜像文件（下载前需要您首先注册为会员）, 请下载标准树莓派 3B/3B+/4B 硬件专用镜像或者针对欧比特测控的 EdgeBox-RPI 硬件的专用镜像, 如下图所示:

| 标准树莓派镜像（适用于树莓派3B/3B+/4B）     |          |    |  |
|------------------------------|----------|----|--|
| 标准树莓派镜像.7z.001               | 980.0MB  | 下载 |  |
| 标准树莓派镜像.7z.002               | 980.0MB  | 下载 |  |
| 标准树莓派镜像.7z.003               | 490.37MB | 下载 |  |
| 标准树莓派镜像（适用于欧比特测控EdgeBox-RPI） |          |    |  |
| EdgeBox树莓派镜像.7z.001          | 980.0MB  | 下载 |  |
| EdgeBox树莓派镜像.7z.002          | 980.0MB  | 下载 |  |
| EdgeBox树莓派镜像.7z.003          | 490.37MB | 下载 |  |

注意: 由于镜像文件较大, 因此网站上会拆分为几个压缩包, 请使用 360 压缩工具或者 7zip 工具进行解压缩, 解压完成后会生成一个.img 后缀的镜像文件。

## 4. 3. 烧写树莓派专用镜像文件

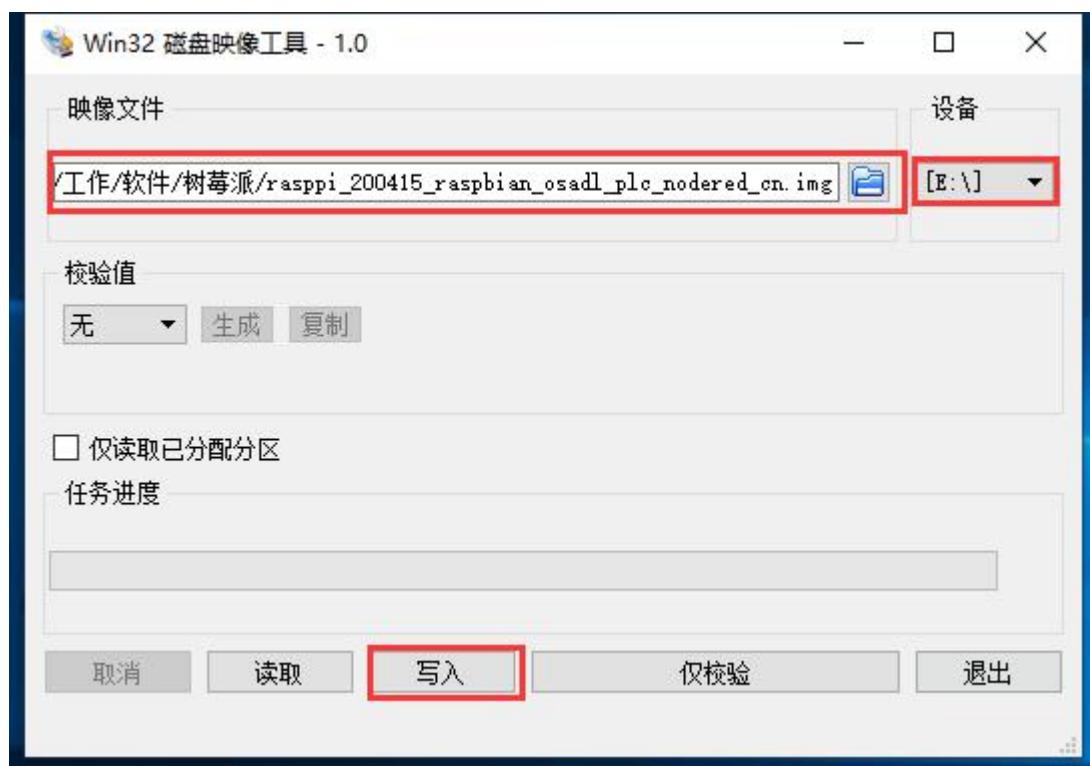
对于标准树莓派硬件采用 TF 卡 (推荐使用 SANDISK 的 16GB TF 存储卡) 方式存储操作系统镜像, 因此需要在 PC 上使用 Win32DiskImager 工具 2.2 章节下载的镜像文件烧写至 TF 卡中。

针对欧比特测控的 EdgeBox-RPI 硬件, 订购该硬件时请向该硬件厂商要求烧写好翌控科技针对 EdgeBox-RPI 硬件的完整镜像, 可以跳过烧写树莓派镜像章节。如果一定需要自行烧写, 请参考 FAQ 章节中内容自行购入相关的烧写设备与自行拆解 EdgeBox-RPI 进行烧写。

本章节烧写过程仅仅针对标准树莓派 3B/3B+/4B 硬件。

步骤 1: 使用 USB 读卡器将 TF 卡插上并插入 PC 上的 USB 口。

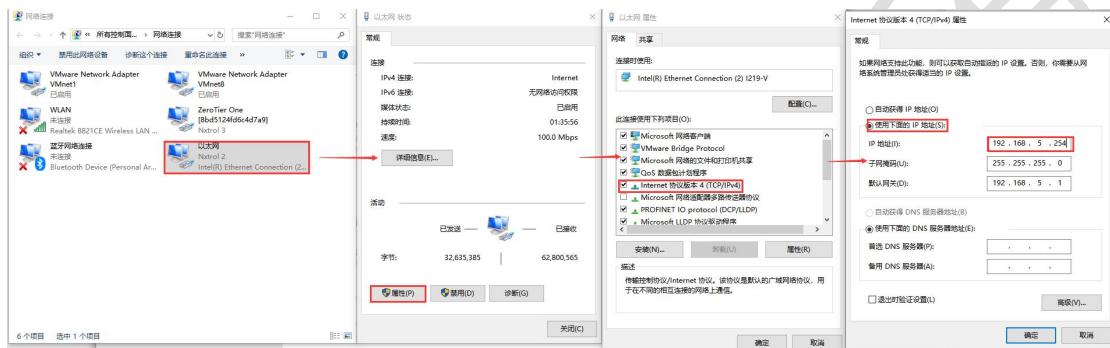
步骤 2: 使用 Win32DiskImager 选择标准树莓派 3B/3B+/4B 系统镜像, 并勾选对应的 USB 读卡器设备, 选择写入



步骤 3: 待写入完成后, 将 PC 上的 USB 读卡器拔出, 同时将 TF 卡插入到树莓派硬件上, 并启动电源。

## 4. 4. 树莓派初次开机与新建工程

将树莓派硬件网口与 PC 网口进行连接，并启动电源，因为树莓派默认 IP 地址 192.168.5.99，需更改 PC 的 IP 地址（或者设定路由器的 IP 地址网段为 192.168.5.X，那么将树莓派以太网口与路由器直连也是可以的），在 192.168.5.X 的网段下，如下图所示：



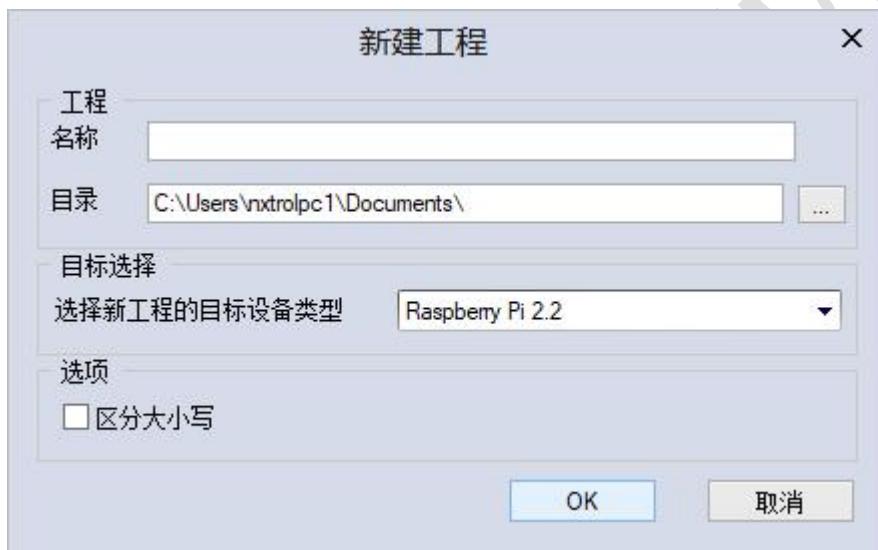
(附：如果需要更改树莓的 IP 地址，请参考<<6.7 树莓派设定固定 IP 地址>>章节内容)

打开 LogicLab 编程软件点击扫描网络：



待树莓派设备正常启动完毕后，LogicLab 返回的扫描结果中即可得知该树莓派硬件设备的 IP 地址与设备名称（如果使用标准树莓派显示为 Raspberry Pi 或者 Raspberry Pi EtherCAT，如实使用欧比特 EdgeBox 显示为 EdgeBox-RPI），并通过新建工程按钮来创建针对该设备的工程：

| 扫描结果         |     |               |      |      |      |      |
|--------------|-----|---------------|------|------|------|------|
| 目标名          | 版本  | IP地址          | 新建工程 | 导入工程 | 应用名称 | 应用版本 |
| Raspberry Pi | 2.2 | 192.168.5.144 |      |      | N/A  | N/A  |



工程创建完毕后，主界面如下图所示，

## 4. 5. 树莓派硬件授权购买与激活

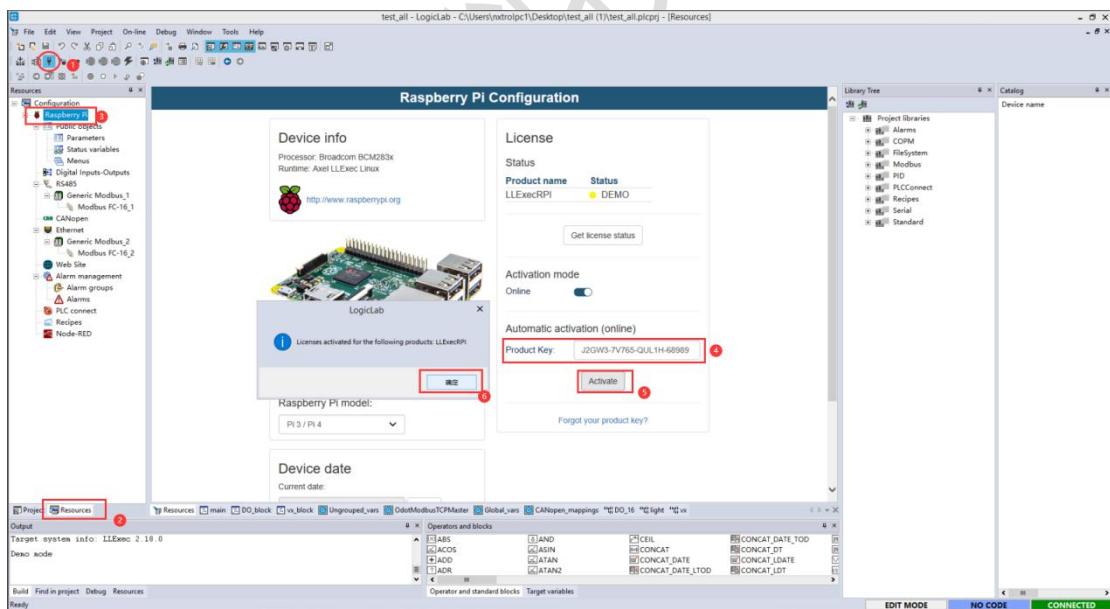
基于树莓派硬件（树莓派 3B/3B+/4B、欧比特 EdgeBox-RPI）的 PLC 核心可以没有任何限制运行 30 分钟，时间到期后，系统会自动停机，此时需要重新启动树莓派硬件。因此如果期望没有任何时间限制运行基于树莓派的 PLC 系统，请访问翌控科技官网在线商店进行购买 ([https://www.nxtrol.com/product\\_rpi.html#\\_pp=123\\_1025](https://www.nxtrol.com/product_rpi.html#_pp=123_1025))：



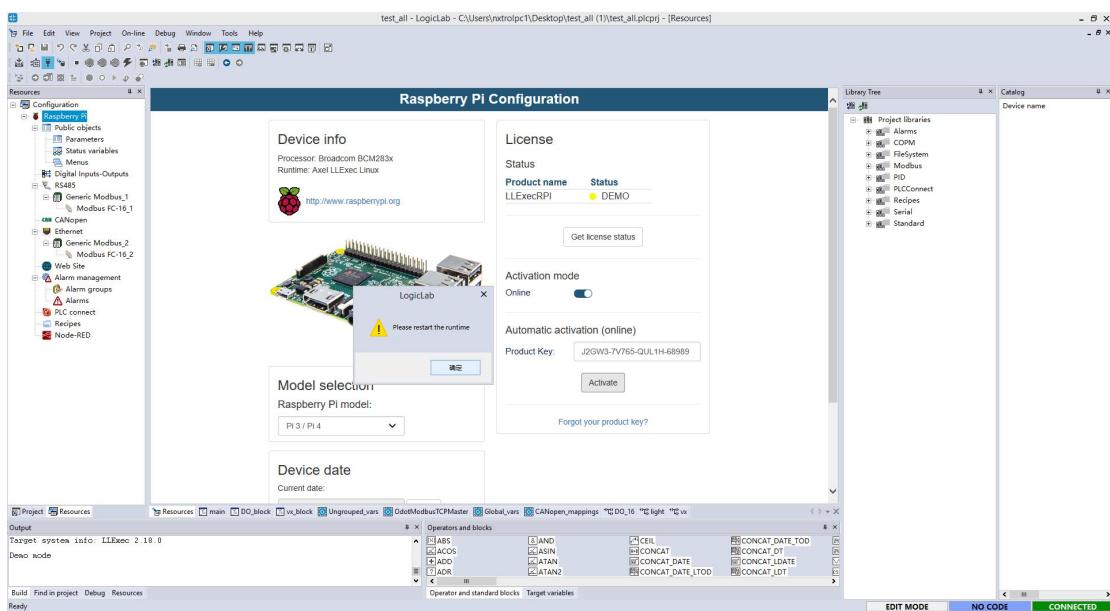
The screenshot shows the LogicLab product page on the Nxtrol Technology website. At the top, there is a navigation bar with links for 首页 (Home), 产品中心 (Product Center), 解决方案 (Solutions), 新闻资讯 (News), 服务支持 (Service Support), 联系我们 (Contact Us), 在线商城 (Online Mall) (highlighted with a red box), 登录 (Login), and 注册 (Register). Below the navigation bar, there is a banner for "LOGIC LAB IEC 61131-3 ENVIRONMENT". A red arrow points from the "在线商城" button on the navigation bar down to the "LogicLab树莓派PLC运行授权" section on the page. This section includes an image of a Raspberry Pi connected to a logic board, a price table, and a "Get license status" button.

如果购买了 LogicLab 树莓派正式运行授权后，您会通过邮件接收到一串序列码，并通过下面步骤可以进行在线激活：

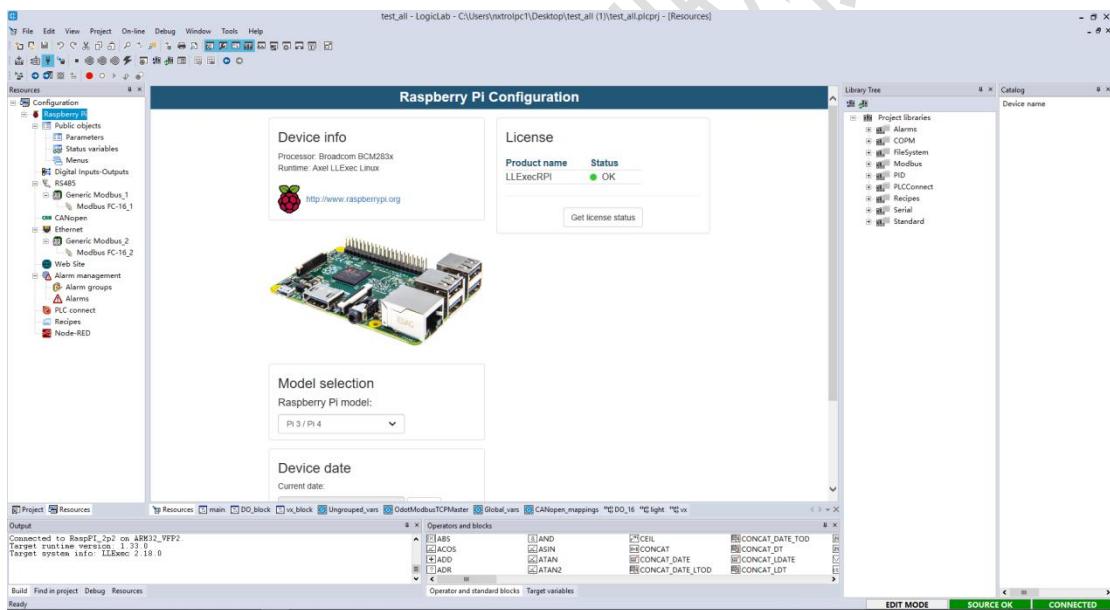
步骤 1：启动 LogicLab 编程工具，成功连接树莓派在设备信息主页面显示当前 License 状态，并可以选择在线(Online)激活



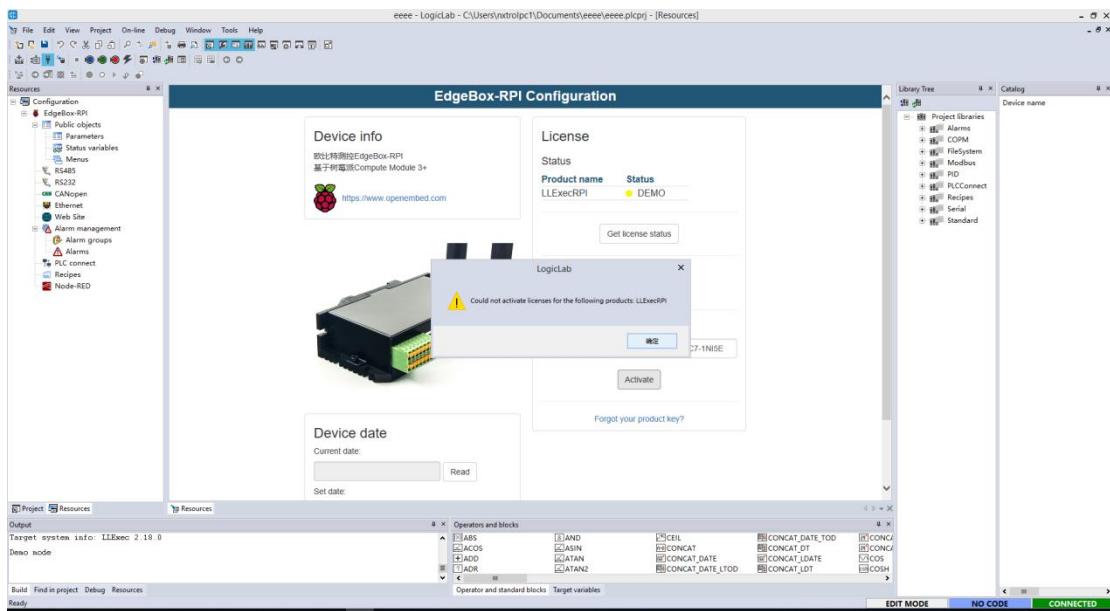
步骤 2：等待激活提示为“Please restart the runtime”后，点击确定并手动重启树莓派。



步骤 3：重启树莓派后，连接到在线状态，状态如下图所示，代表激活成功。



注意：在网络状况不佳的情况下，点击 activate 后，可能会提示 “could not activate licenses for the following products”，这时只需重复点击 Activate 即可。



如反复上述操作依然无法激活授权，请参考<<6.4 树莓派授权激活>>章节内容。

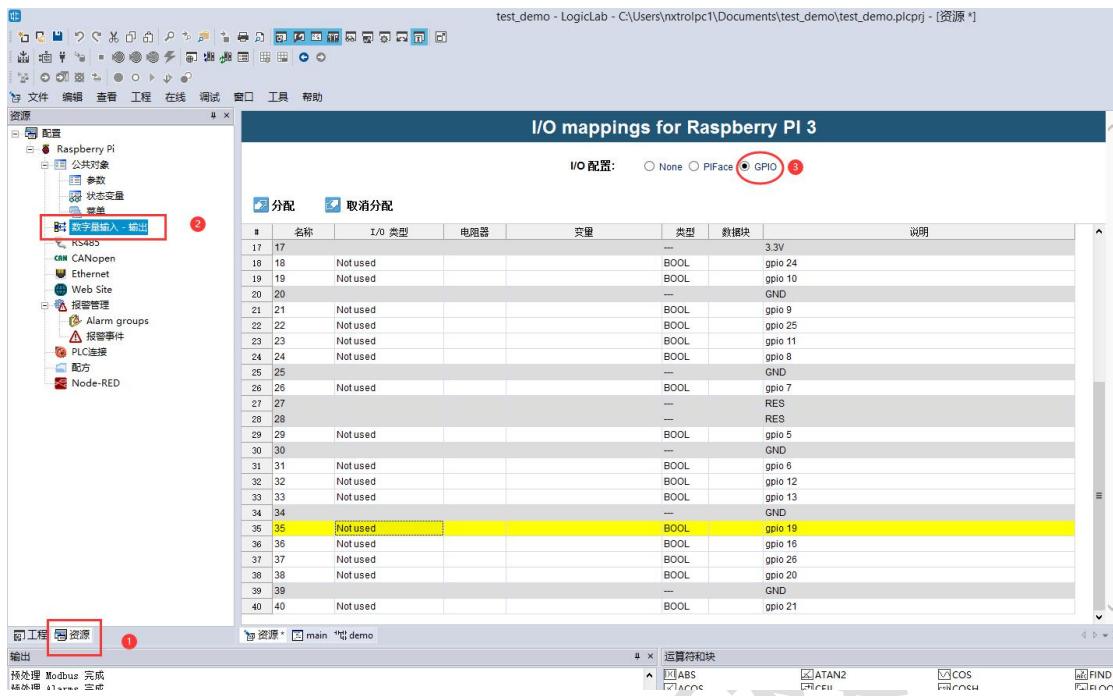
## 5. 基于树莓派的 LogicLab 入门操作

LogicLab 通用使用方法请参考<<LogicLab 编程指南.pdf>>, 在本章节内容中, 主要涉及树莓派相关功能在 LogicLab 上的操作:

- 标准树莓派硬件 GPIO
- Modbus TCP
- Modbus RTU
- EtherCAT 主站 (仅支持树莓派 4B)
- NodeRED 物联网开发平台

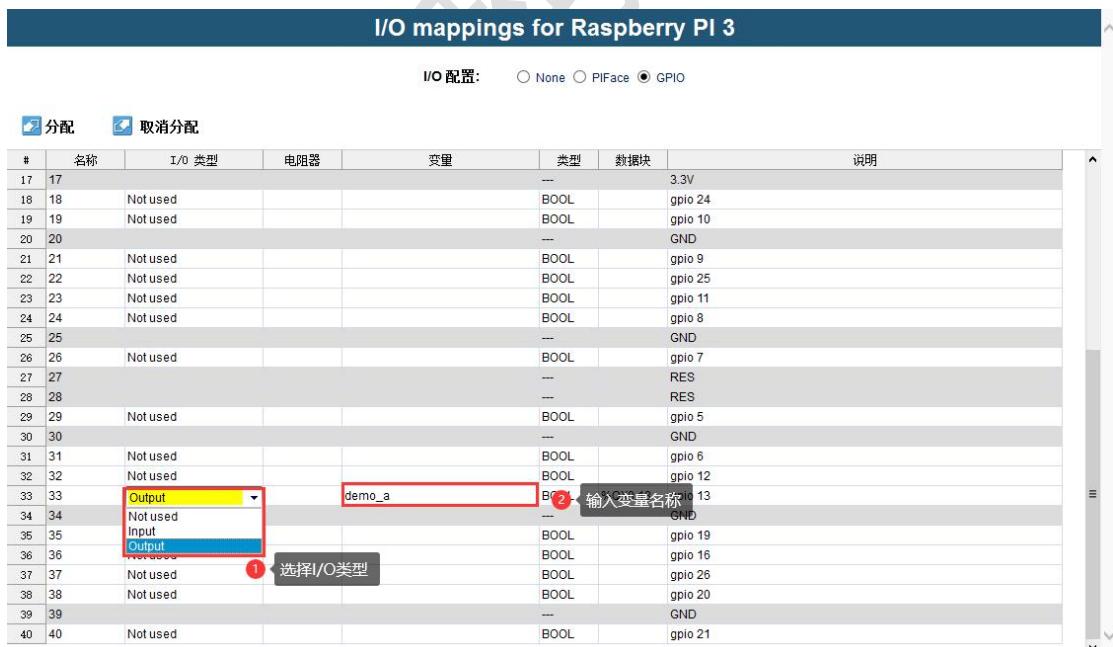
### 5. 1. 标准树莓派硬件 GPIO 控制

按照如下图方式, 进入标准树莓派 I/O 配置页面 (注意: 如果使用的是欧比特 EdgeBox-RPI 硬件, 此功能则会被禁用) :



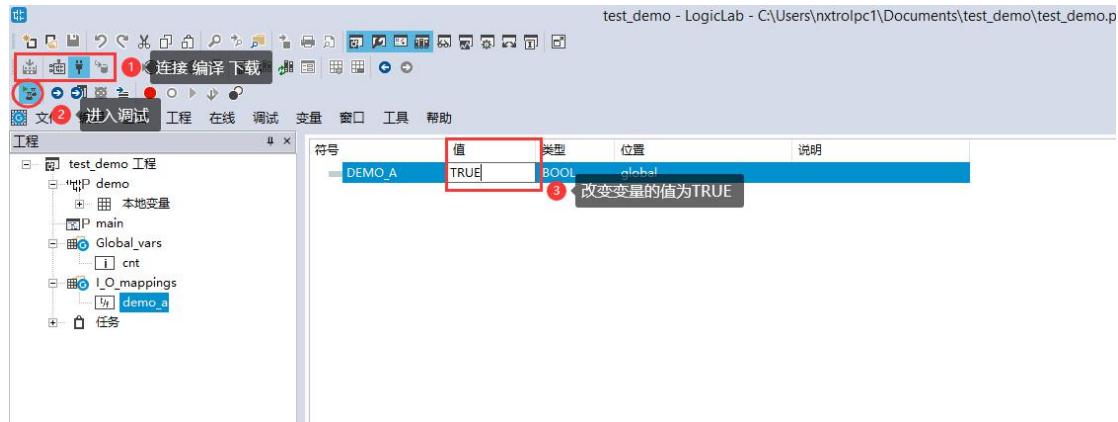
选择需要配置的 I/O 点 点击下拉按钮选择 I/O 类型(我们设定的 I/O 口 33 作为输出类型)。

， 并给 I/O 点添加 PLC 变量名称，用于通过此 PLC 变量访问该 I/O 信息：



此时切换回 LogicLab 工程界面，可以在 I\_O\_mappings 中找到该 I/O 对应的变量信息，

编译下载当前 PLC 程序后，可以切换调试状态，并且在线查看到该 I/O 信息，如果该 I/O 配置为输出类型，那么可以直接在线修改对应的变量值，从而来控制该 I/O 信号。

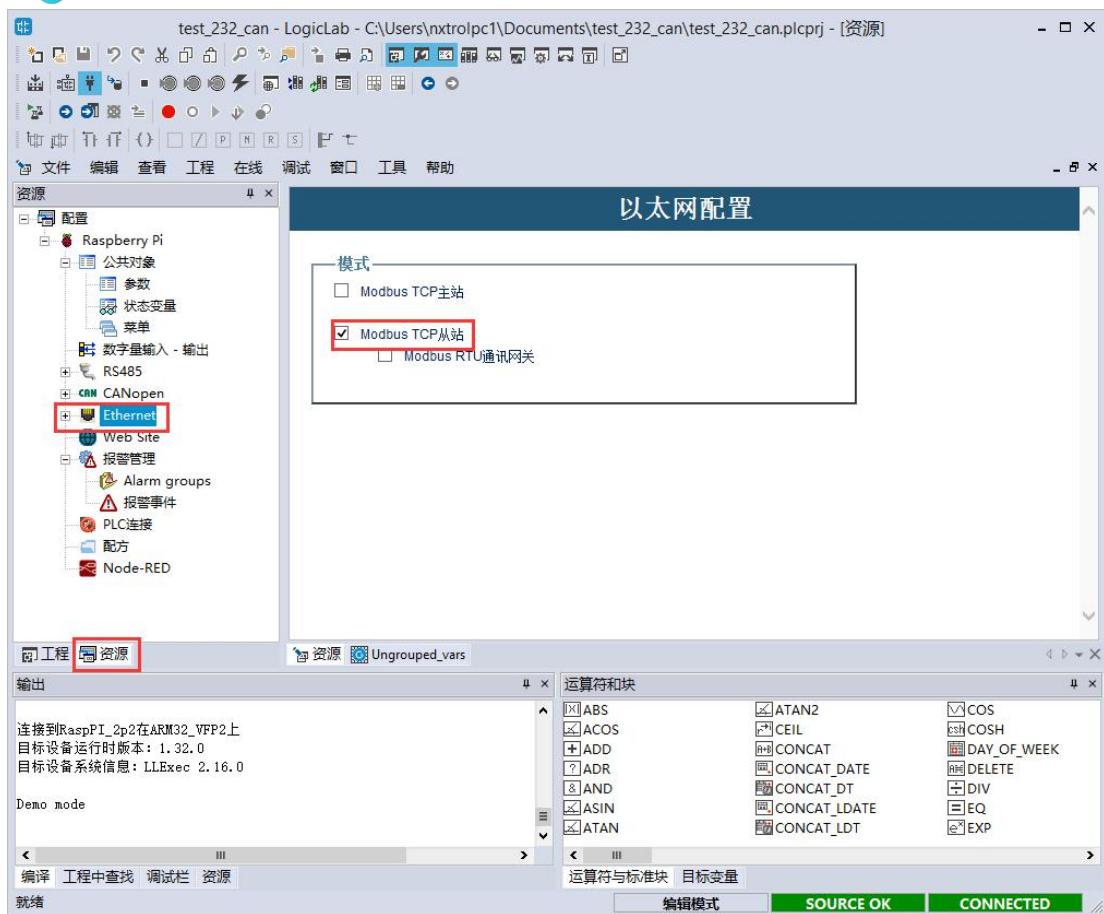


## 5. 2. Modbus TCP

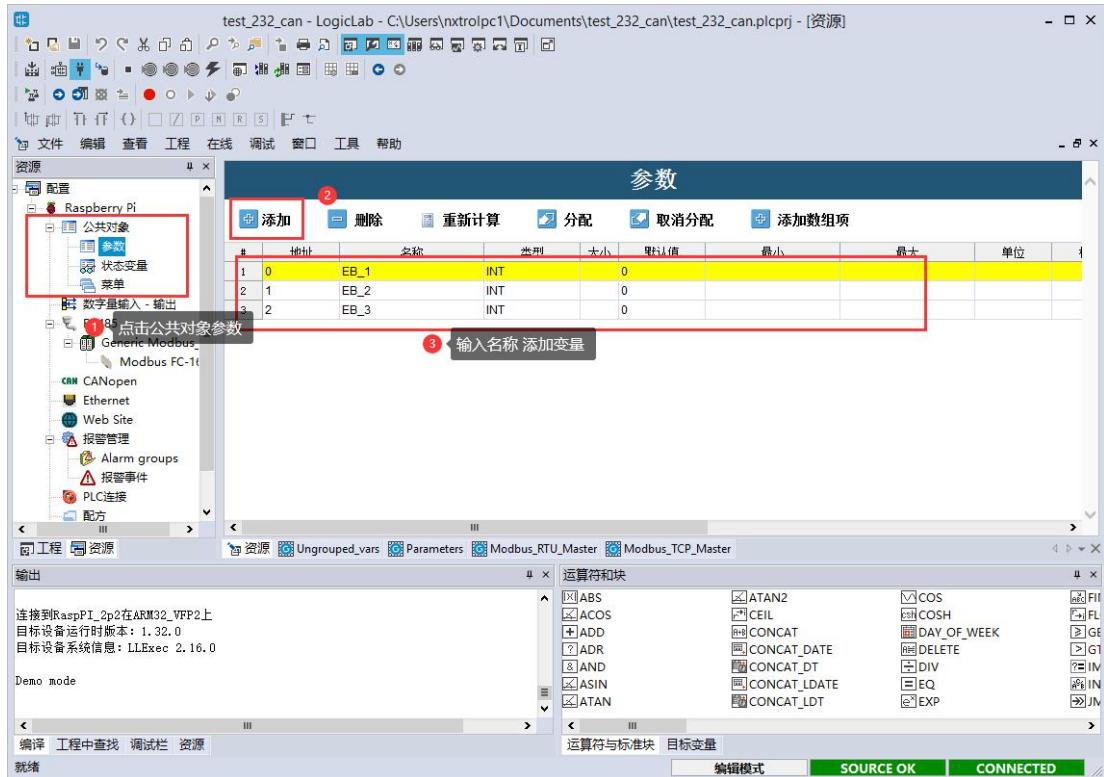
树莓派标准硬件与欧比特测控的 EdgeBox 都可以支持基于以太网的 Modbus TCP 通讯协议，包括 Modbus TCP 主站与从站功能。

### 5. 3. 1. Modbus TCP 从站

Modbus TCP 从站功能需要在资源页面的“Ethernet”节点下进行配置，配置为 Modbus TCP 从站选项如下图所示：



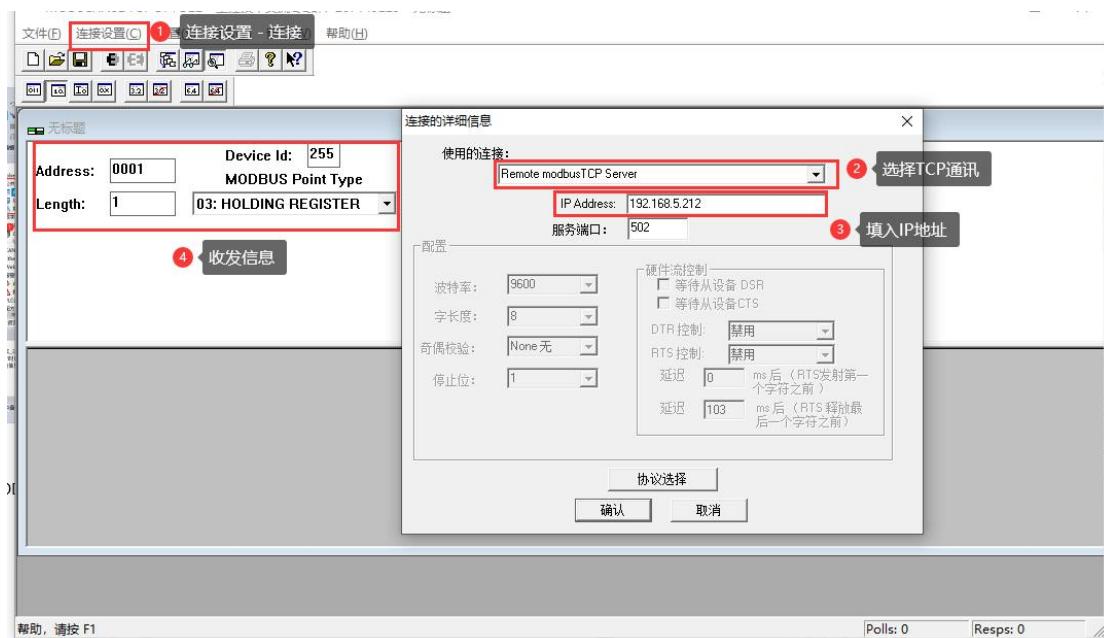
同时在公共对象中添加对应的数据源，默认地址从 0 开始，并关联 PLC 变量：



公共对象地址为全局唯一，因此同一个公共对象地址可以映射为多个 Modbus 地址，如需要同时支持多种不同类型的 Modbus 地址，请合理分配好公共对象的地址空间，其对应关系如下：

| 公共对象地址 | Modbus 地址               |
|--------|-------------------------|
| 0      | 00001/30001/40001/10001 |
| 1      | 00002/30002/40002/10002 |
| 2      | 00003/30003/40003/10003 |
| .....  | .....                   |

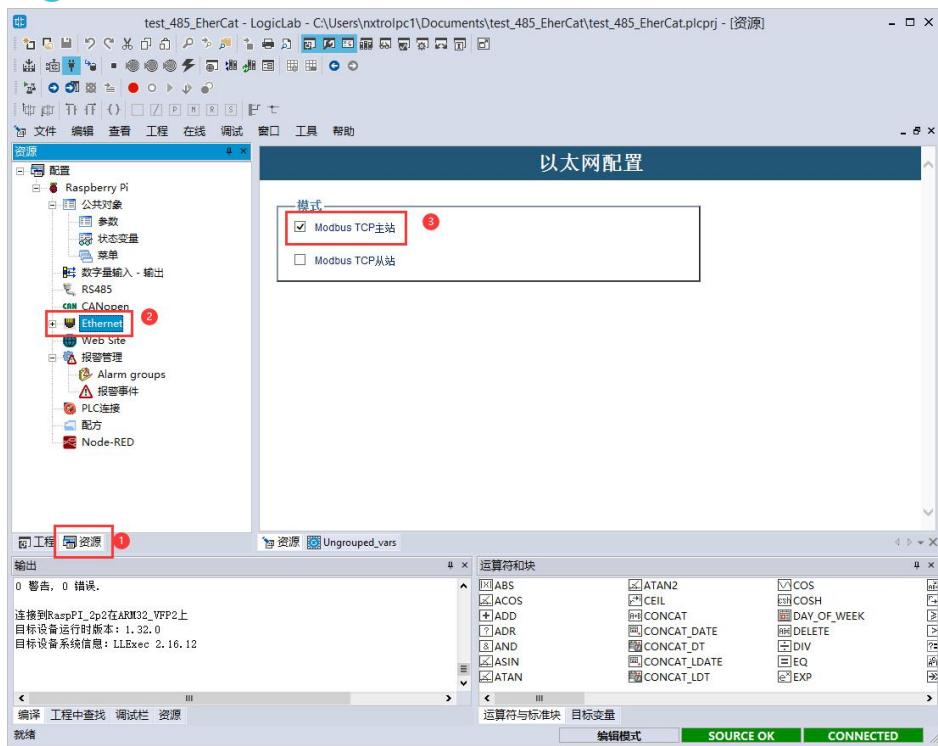
通过 MODSCAN 模拟主站可以验证树莓派 Modbus TCP 功能，配置如下树莓派以太网 IP 地址，并设定访问的 Device ID 为：255，寄存器类型等等信息：



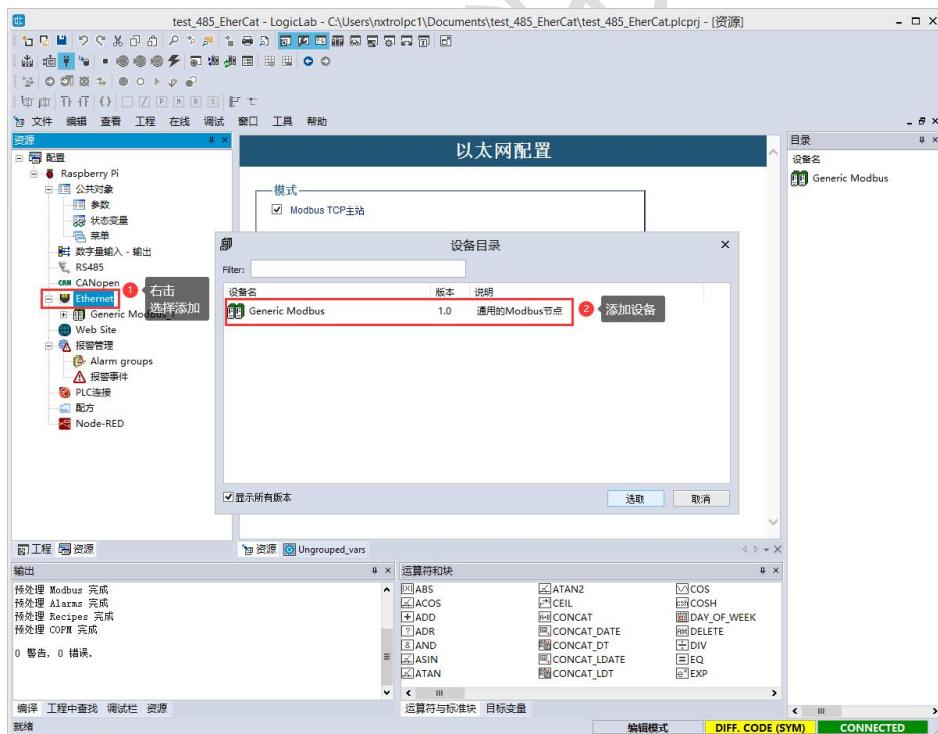
注意 MODSCAN 访问树莓派 Modbus 寄存器地址范围必须与 LogicLab 中定义的一致，否则树莓派可能会返回异常码。LogicLab 中参数配置完成后需要确保该工程进行编译下载后，才能启动已经配置的 Modbus TCP 从站功能。

### 5. 3. 2. Modbus TCP 主站

Modbus TCP 主站功能需要在资源页面的“Ethernet”节点下进行配置，配置为 Modbus TCP 主站选项如下图所示：

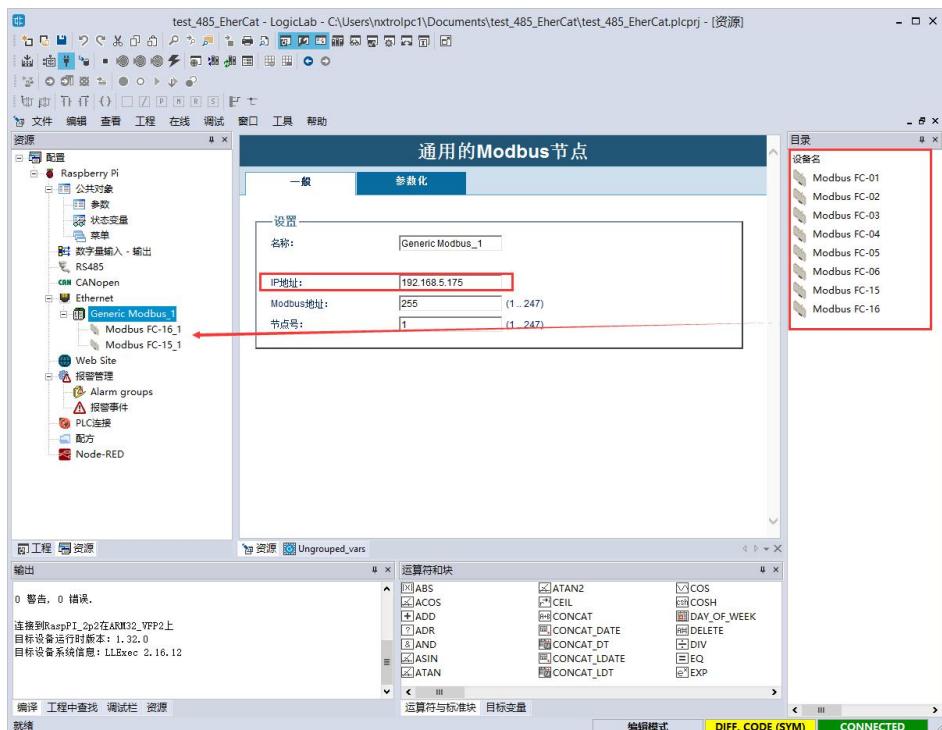


通过在“Ethernet”节点上的右键菜单添加设备，选择“Generic Modbus”设备：

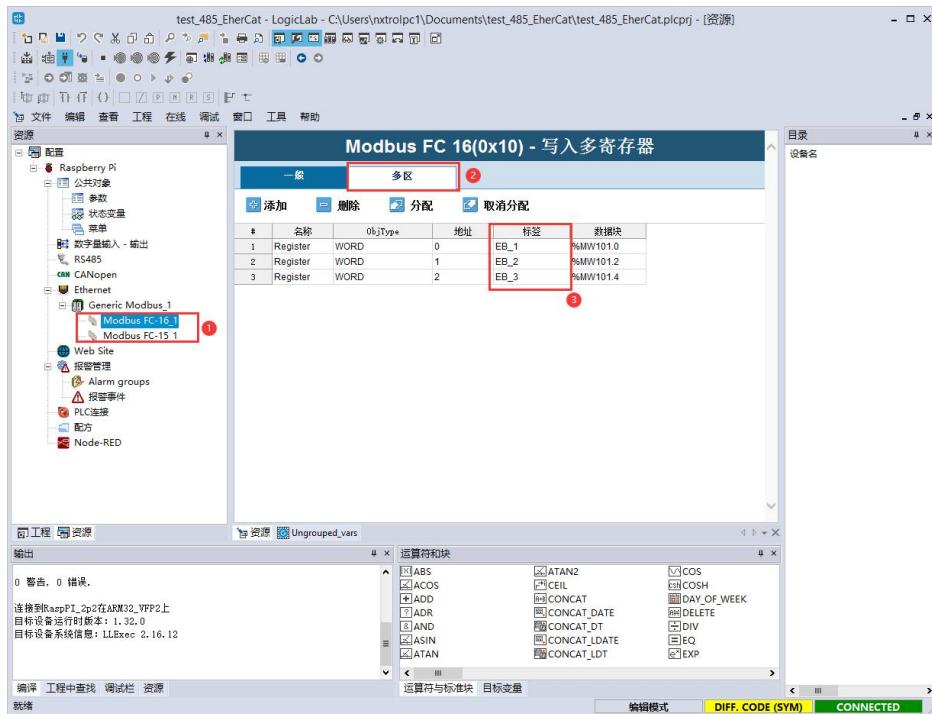


并在“Generic Modbus”节点中配置好需要访问的目标 Modbus TCP 从站的 IP 地址 (本

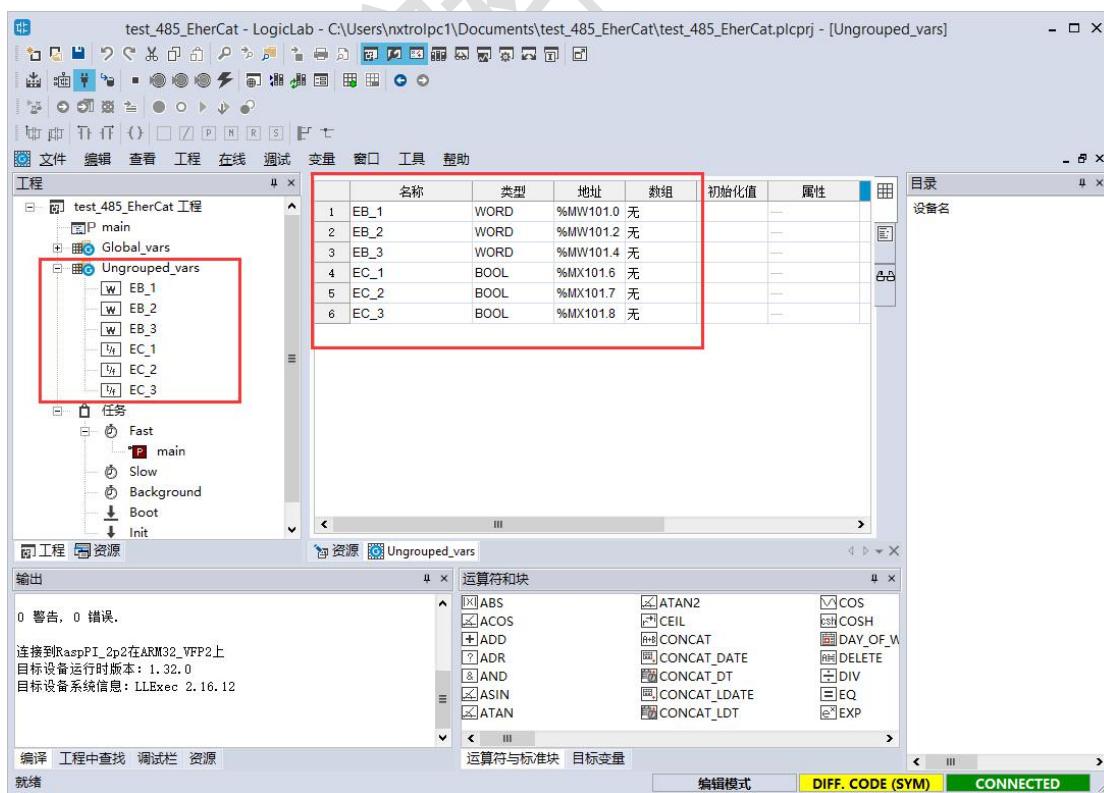
文使用 PC 仿真从站模拟，实际从站可以为 I/O 模块，传感器，或者其他 Modbus TCP 设备等），并可以在右侧设备目录找到目前支持的 Modbus 功能码，可以将期望访问的功能码鼠标左键拖入“Generic Modbus”节点下，如下图所示：



在对应的功能码节点下按照自己的需求填入 PLC 变量名称：

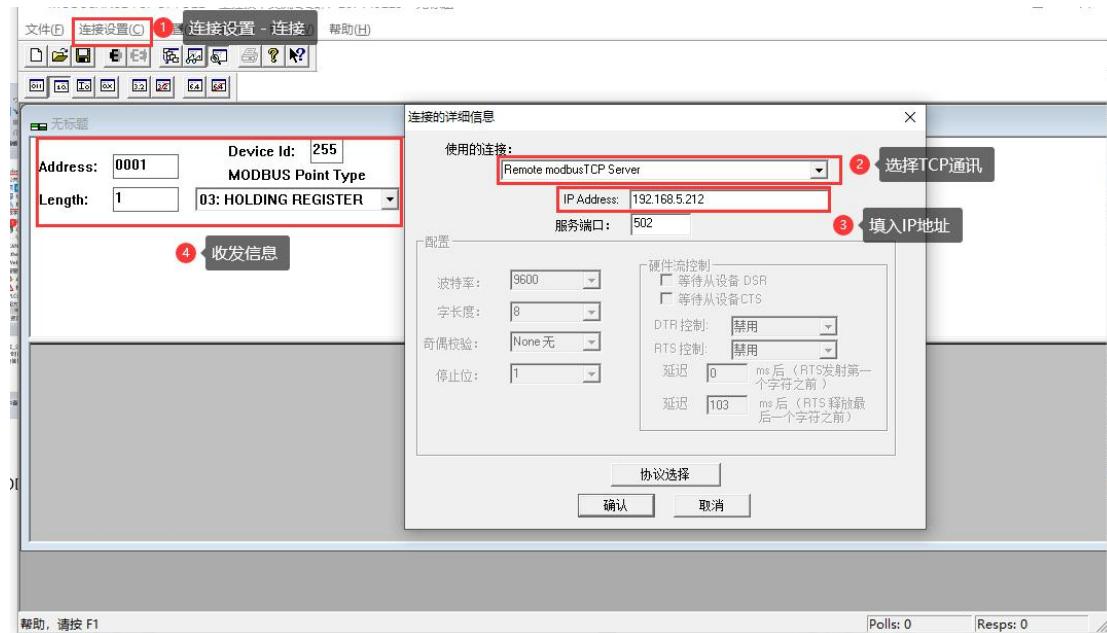


LogicLab 会自动将该功能码对应的 Modbus 远程变量与 PLC 变量进行关联，用户实际最终通过该变量访问远程 Modbus TCP 从站的对应寄存器数据，如下图所示：

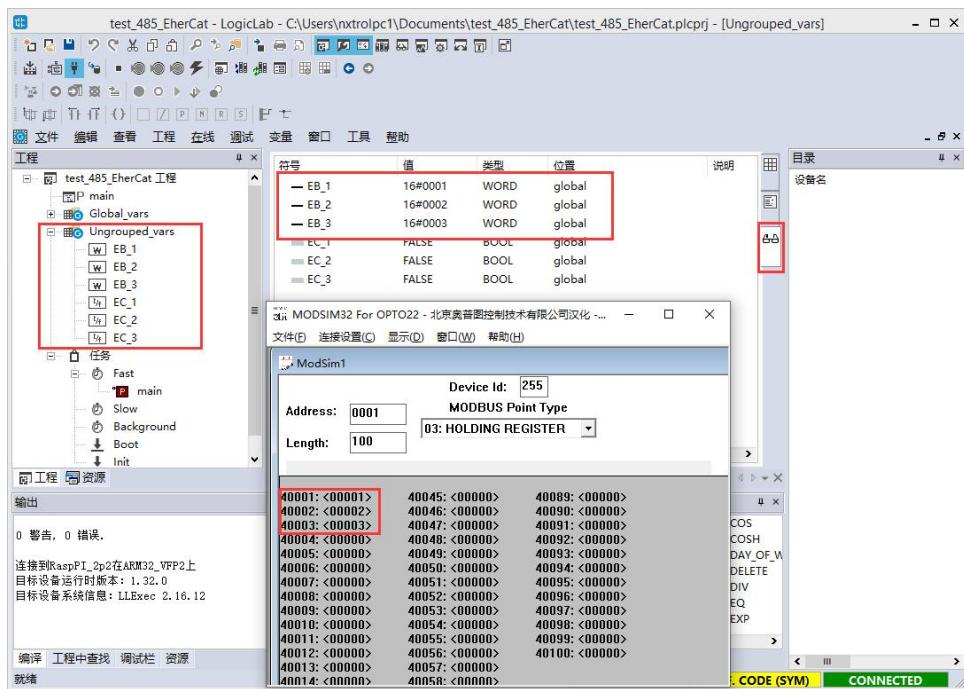


通过 MODSIM 软件可以在 PC 上仿真 Modbus TCP 从站，从而由树莓派设备中的 LogicLab

Modbus TCP 主站进行访问，MODSIM 软件配置流程如下：



在 LogicLab 中强制改变主站变量的值，在对应的 MODSIM 模拟从站中对应寄存器的值也会随之改变：



## 5.3. Modbus RTU

Modbus RTU 使用方式与 Modbus TCP 一致，在参数配置界面上会有稍稍差别，本章节仅介绍 Modbus RTU 的主站与从站的参数配置界面。

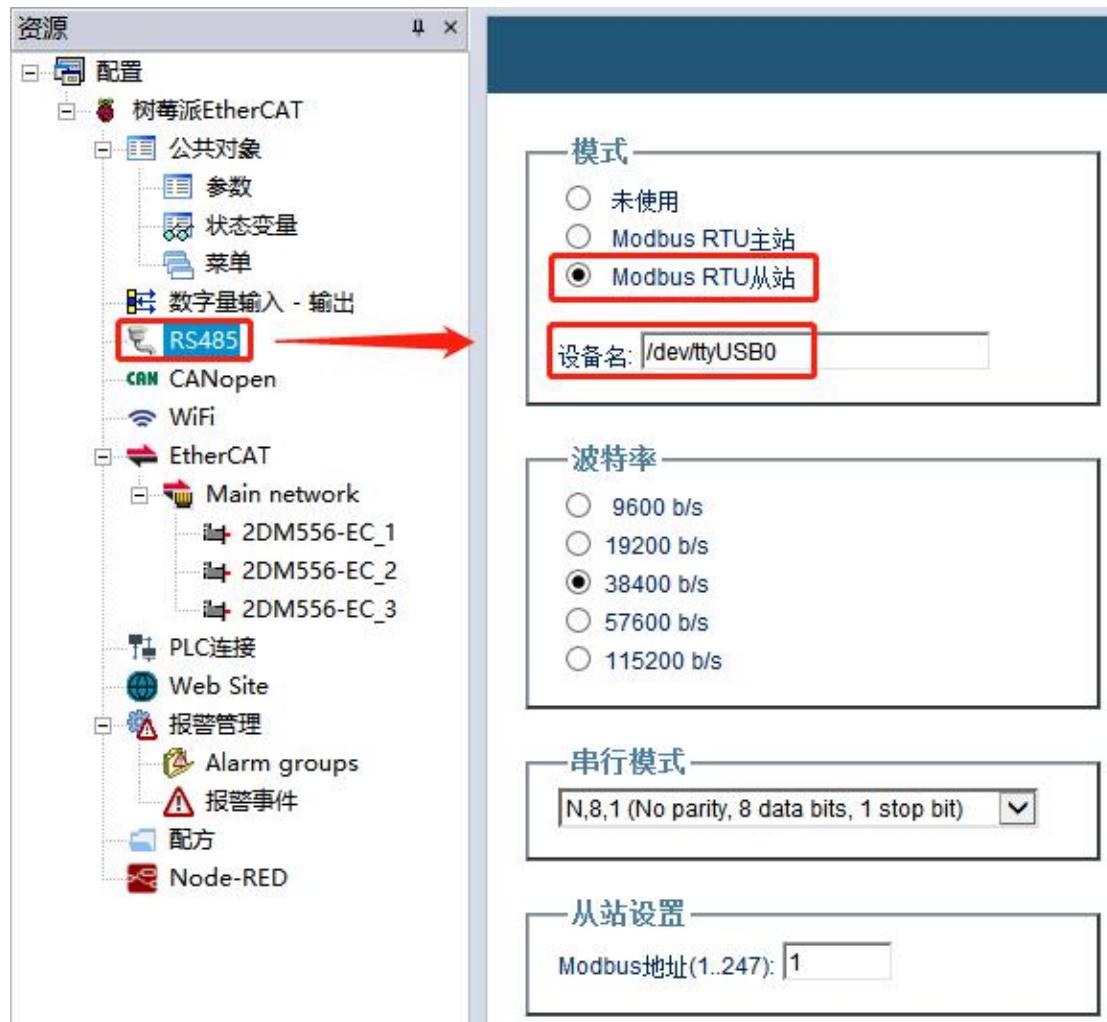
### 5.3.1. Modbus RTU 从站

Modbus RTU 可以基于 RS232 或者 RS485 接口实现，目前在标准树莓派硬件上通过外接的 USB 转 RS232/RS485 设备或者独立的扩展板（推荐微雪 RS485 扩展板）实现支持 RS232/RS485 接口。欧比特测控 EdgeBox-RPI 设备可选配隔离的 RS232&RS485 接口板本。

Modbus RTU 从站配置如下，标准树莓派硬件需要指定对应的设备名，该设备名为 RS232/RS485 在树莓派 Linux 系统下的设备驱动名称，默认为：/dev/ttyUSB0，可以根据实际选用的 RS232/RS485 设备来确定该设备名。对于欧比特测控 EdgeBox-RPI 设备，

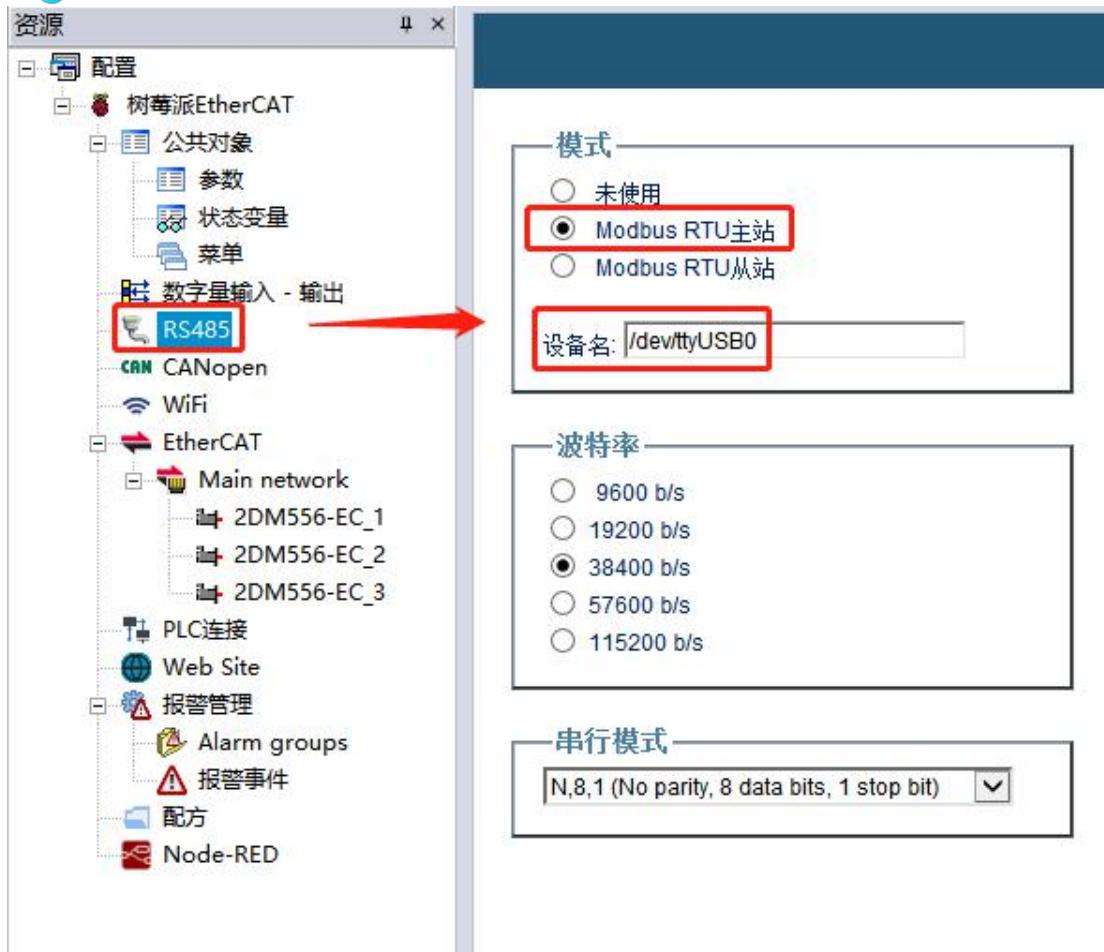
RS232 与 RS485, LogicLab 已经默认配置好对应的设备名，用户无需配置。

波特率、串行模式与 Modbus 地址设定需要与当前网络使用的参数保持一致。



### 5.3.2. Modbus RTU 主站

Modbus RTU 主站参数配置与从站基本一致，如下图所示：



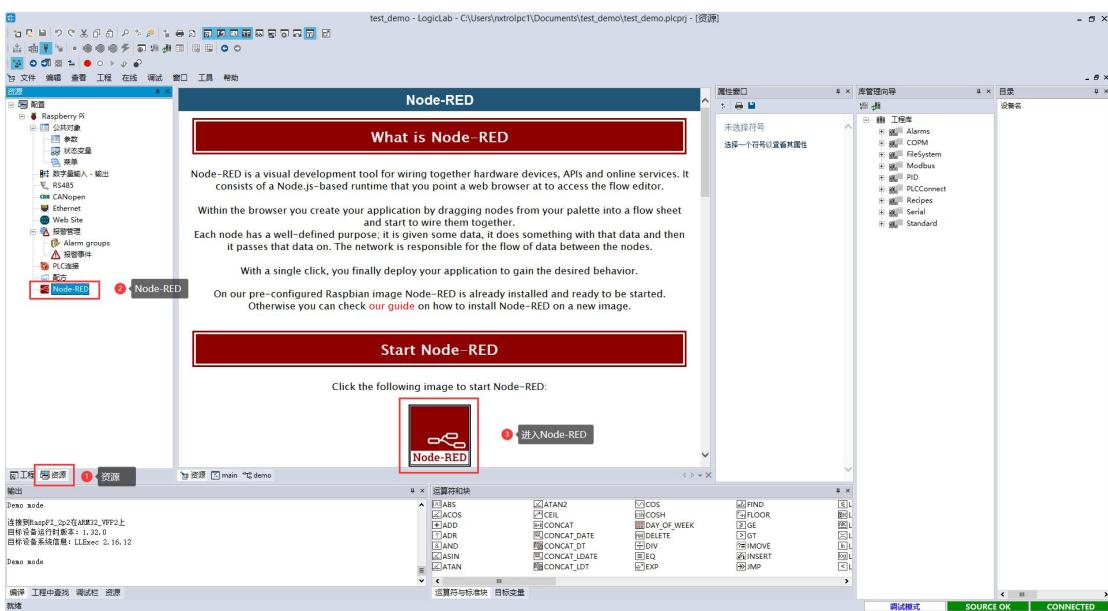
## 5. 4. Node-RED 物联网开发平台

Node-RED 是构建物联网应用程序的一个强大工具，其重点是简化代码块的"连接"以执行任务。它使用可视化编程方法，允许开发人员将预定义的代码块（也叫做"节点"）连接起来执行任务。连接的节点，通常是输入节点、处理节点和输出节点的组合，当它们连接在一起时，构成一个"流"。最初是 IBM 在 2013 年末开发的一个开源项目，以满足他们快速连接硬件和设备到 Web 服务和其他软件的需求--作为物联网的一种粘合剂--它很快发展成为一种通用的物联网编程工具。重要的是，Node-RED 已经迅速开发出一个重要的、不断增长的用户基础和一个活跃的开发人员社区。

LogicLab 作为符合 IEC61131-3 标准的 PLC 平台，编写 PLC 逻辑控制对象主要是工业自动化现场层的信号，也即是 OT(Operational Technology)，而 Node-RED 优势在于面向 IT(Information Technology)的数据处理与通讯，实现开放式 IT&OT 融合的边缘计算控制器，LogicLab 与 Node-RED 将是最佳搭档。

### 5. 4. 1. Node-RED 基础操作入门

LogicLab 编程工具正常与树莓派 4B 连接情况下，可以直接通过 LogicLab 编程工具中的 Node-RED 选项自动打开浏览器进入 Node-RED 界面，如下图所示：

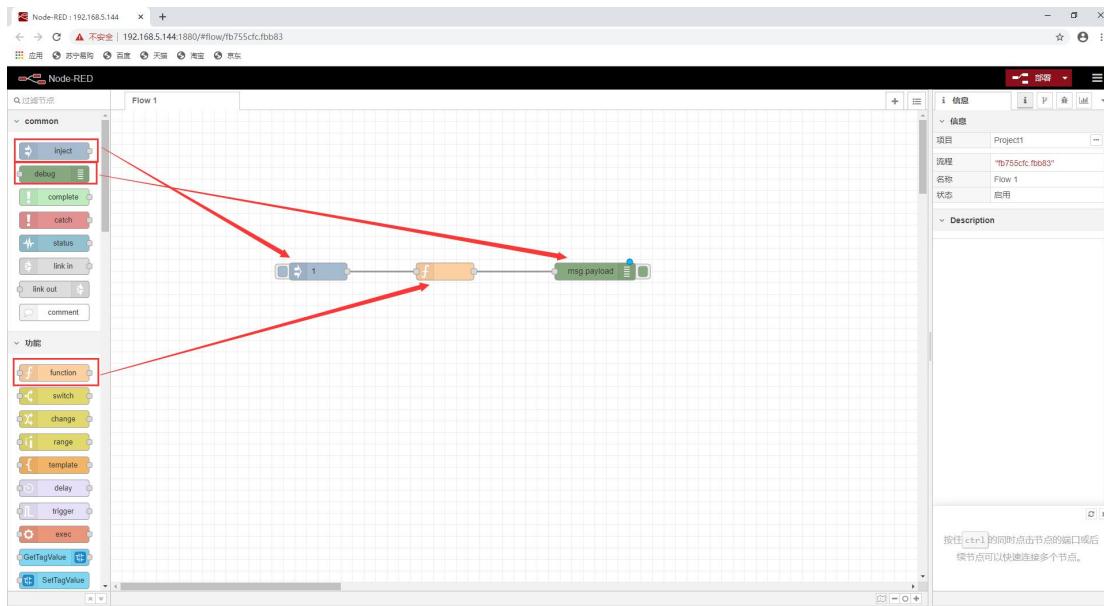


登录 Node-RED 系统需要用户名与密码，并且不同用户具有不同的操作权限，下面为

Node-RED 默认用户名密码信息：

- 管理员用户名:admin, 密码:nxtrol
- 访客用户名:guest, 密码:guest

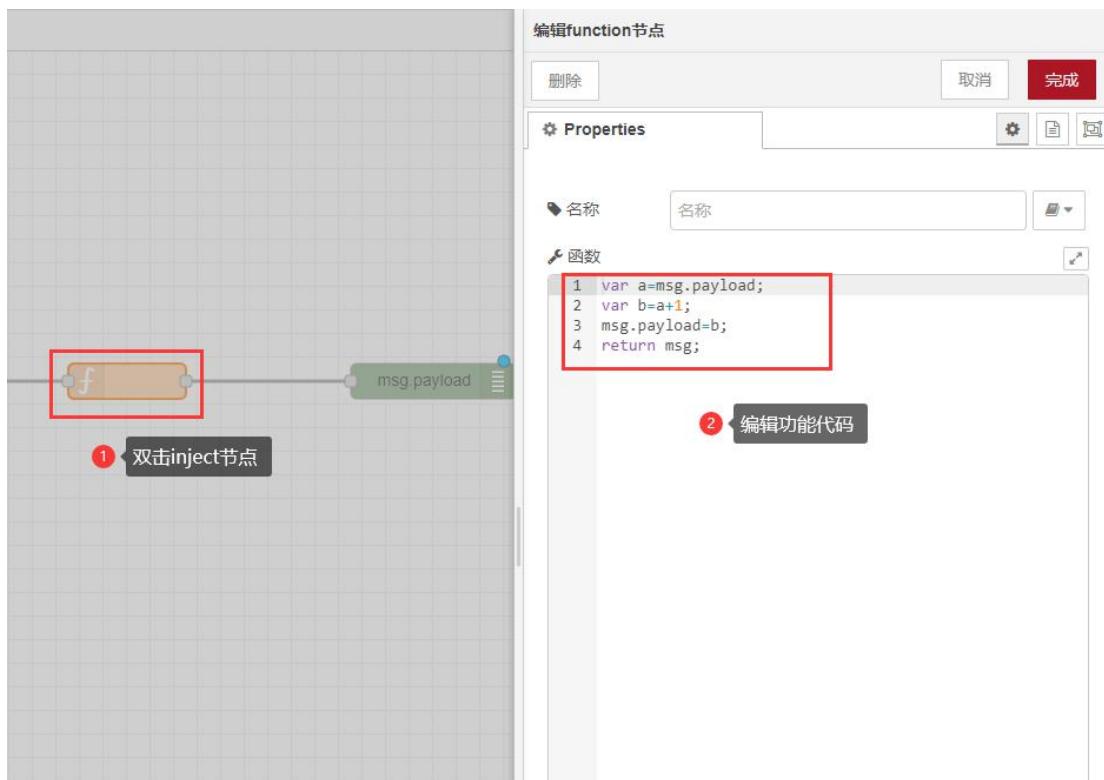
登录至 Node-RED 主界面以后，可以通过拖拽左侧的节点至流程编辑区，并且将节点之间进行连接，从而完成节点间的数据流程。下图中拖拽了 inject(注入), function(功能)与 debug(调试)这三个节点到流程编辑区，并进行连接：



双击 inject 节点，可以选择内容类型为数字，输入值为 1，重复类型可以根据自己的需要进行配置，如下图所示：



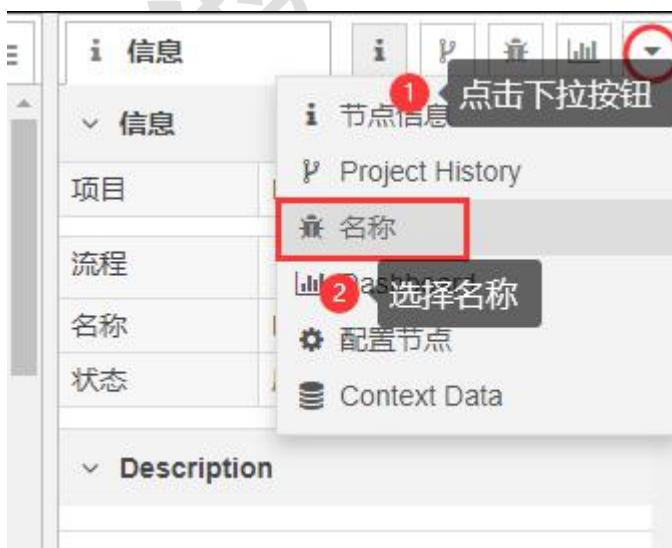
双击 function 节点，可以通过自行编辑 javascript 代码对输入的数据进行处理，结果输出给下一个节点。在下图示例代码中，该 function 第 1 行的 msg.payload 是由前个节点 inject 传入的值，而第 3 行的 msg.payload 的值将会传给后个节点 debug：



点击右上角的部署按钮，完成当前 Node-RED 流程部署，如下图所示：

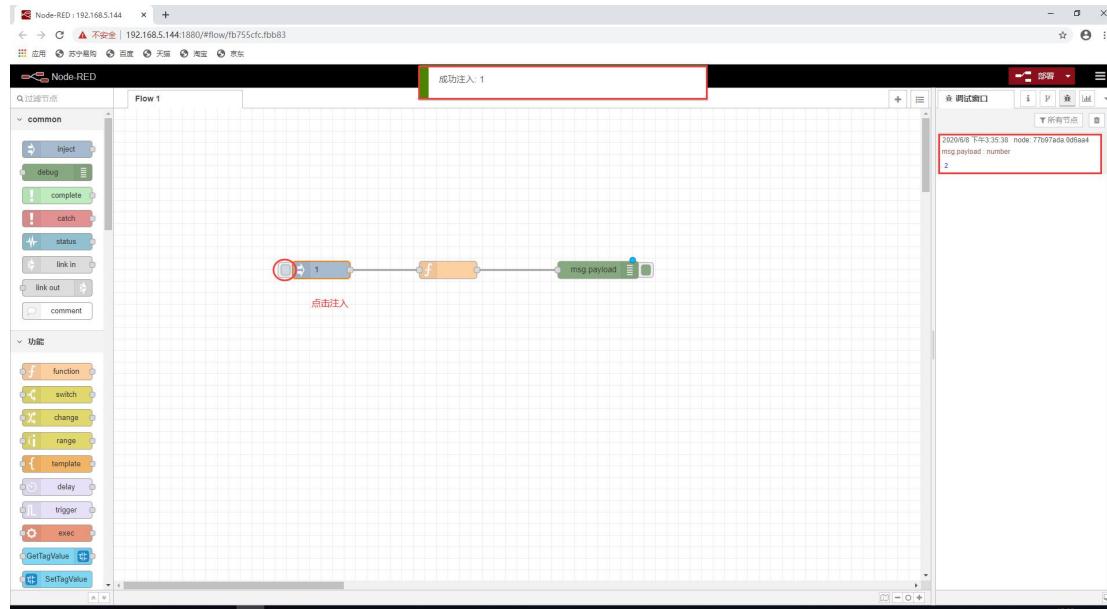


部署完成后，在右侧窗口中打开调试信息窗口，如下图所示：



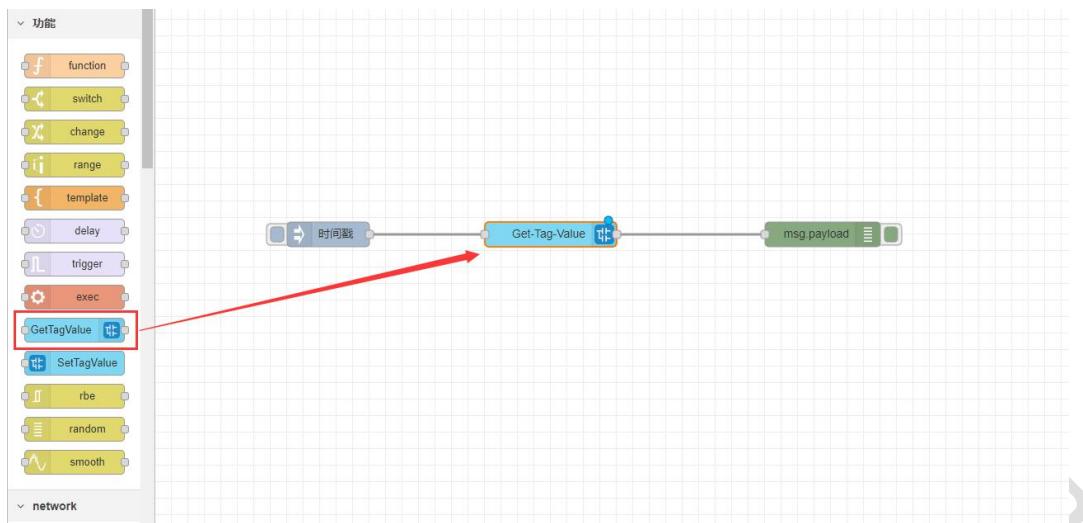
点击 inject 左边的注入按钮 上方显示成功注入 右边调试窗口显示结果，证明当前

Node-RED 流程已经执行，如下图所示：



## 5. 4. 2. LogicLab 与 Node-RED 数据交换

在 Node-RED 的指令中，默认已经包含读写 LogicLab PLC 变量的功能指令：GetTagValue 与 SetTagValue，用户可以直接将该指令拖拽至流程编辑区域，并进行数据连接，如下图所示：



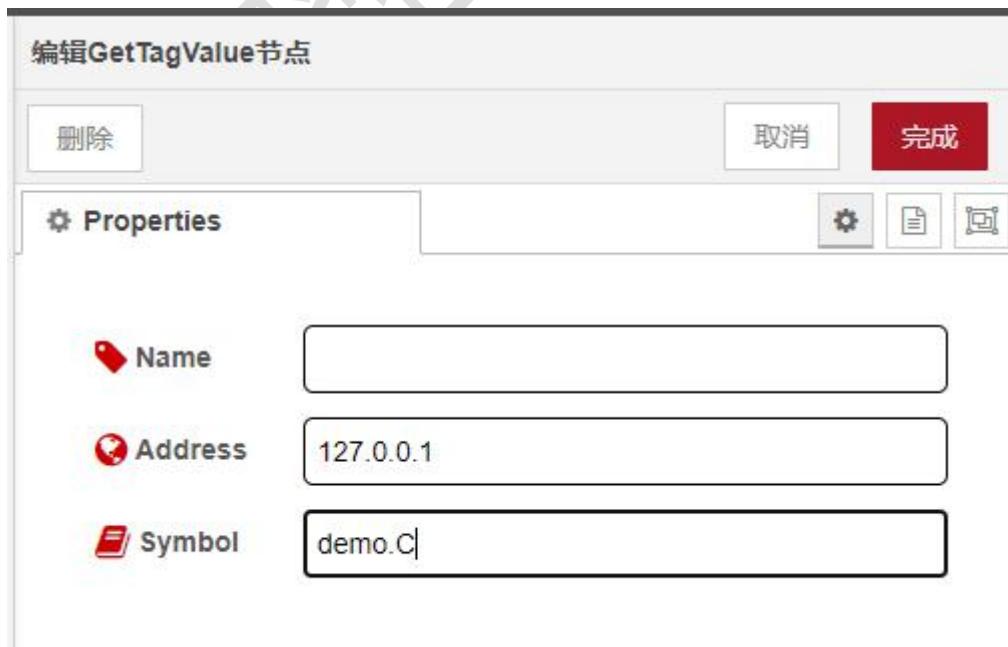
通过双击 Get-Tag-Value 节点，可以打开参数配置界面，保持 Address 为本地 IP 地址 127.0.0.1，Symbol 名称为 LogicLab 中定义的变量名，其中需要注意该名称规则如下：

- LogicLab 全局变量

直接使用变量名作为 Symbol 的值

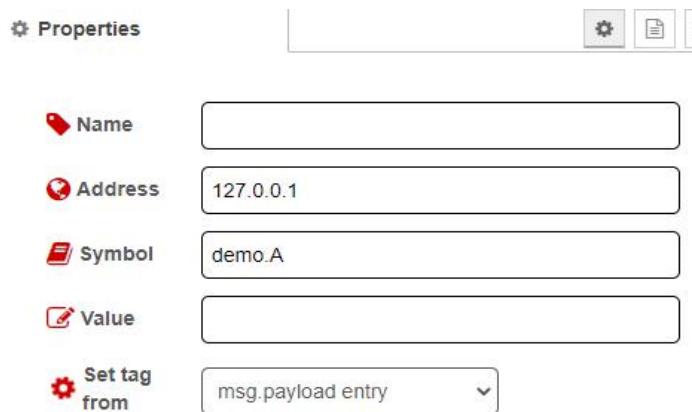
- LogicLab 局部变量

使用 程序名.局部变量名 作为 Symbol 的值

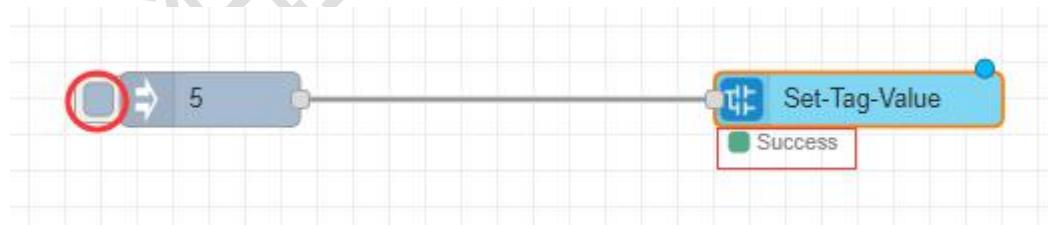


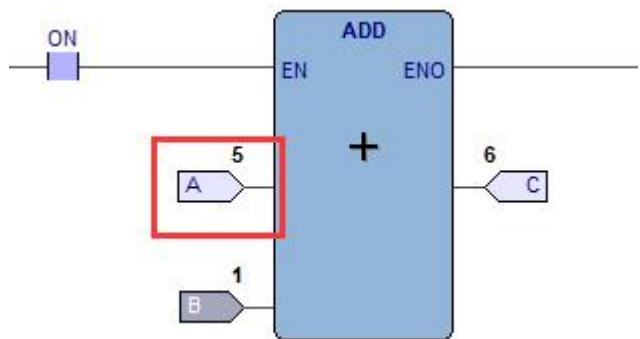
点击 inject 按钮左方图标，通过注入一次操作，执行一次后续流程。Get-Tag-Value 指令将会从 LogicLab 内核中直接获取到指定的 PLC 变量数据值，并且将其传递到 Debug 节点进行输出。

SetTagValue 使用方法类似，Address 与 Symbol 配置规则与 GetTagValue 一样，SetTagValue 的数据来源可以为下图中配置的 Value 值，也可以来自于节点的输入 payload 或者 property。



使用注入可以进行简单的测试，将常数直接写入对应的 LogicLab PLC 变量中，如下图所示：





上海翌控科技版权所有

## 5. 5. EtherCAT 主站 (仅支持树莓派 4B)

基于树莓派 4B 可以实现 EtherCAT 主站功能，但由于树莓派 Linux 系统实时性影响，EtherCAT 主站功能仅供评估与学习使用，请勿在商业产品中进行使用。因为标准树莓派 4B 硬件只包含有一个以太网口，并且树莓派 PLC 镜像默认已经使用该太网接口用于 TCP/IP 通讯，因此需要按照下面步骤首先设定树莓派 4B 的 WIFI 用于连接到无线路由器，用于 TCP/IP 通讯，然后基于 LogicLab 使用 EtherCAT 主站功能。

### 5. 5. 1. 树莓派 4B WIFI 通讯设置

树莓派 PLC 镜像烧写至树莓派 4B 使用的 TF 卡中，通过网线将树莓派 4B 的以太网接口与无线路由器的网口进行连接，同时安装了 LogicLab Automation Suite 的 PC 也通过网线 /WIFI 连接至该路由器上，确保 PC 与树莓派 4B 处于同一个局域网内，连接树莓派 4B 的电源，等待约 60 秒，树莓派 4B 内置的 PLC 系统将启动完毕。

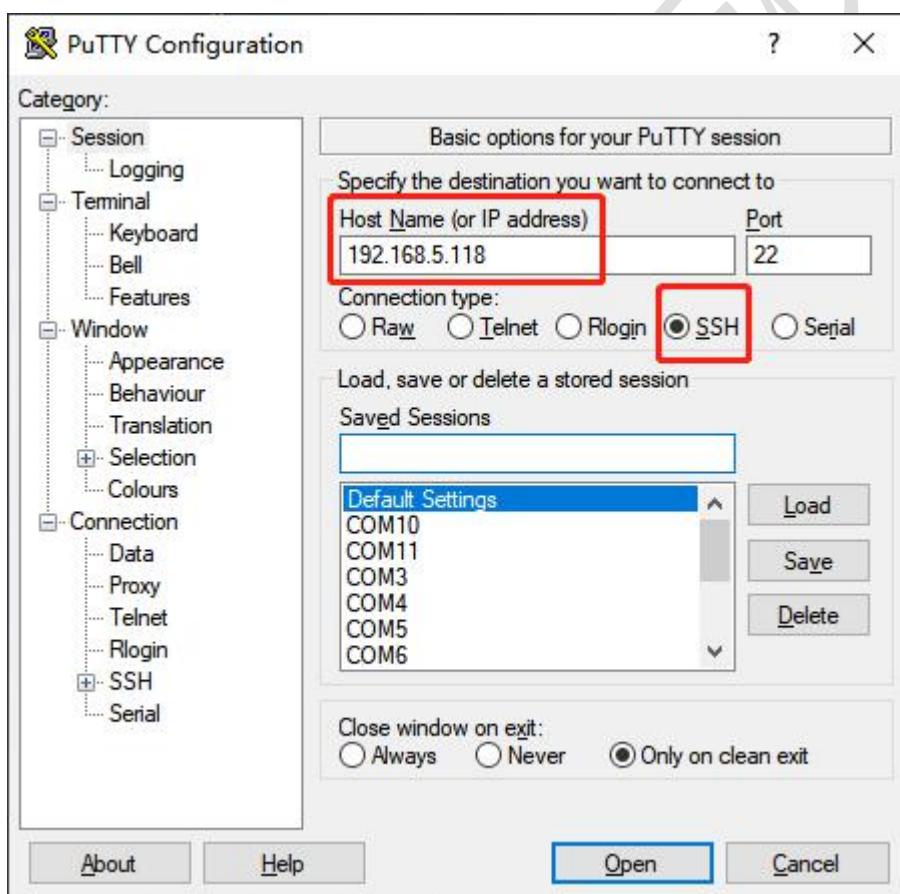
打开 LogicLab 编程软件，在主界面上点击“扫描网络”，如果通讯一切正常，那么在扫描结果下会出现当前网络中所有树莓派设备的信息，包括其 IP 地址，因此在任何时候如果忘记树莓派硬件 IP 地址时，都可以使用此方法来进行确认。

# LOGIC|LAB

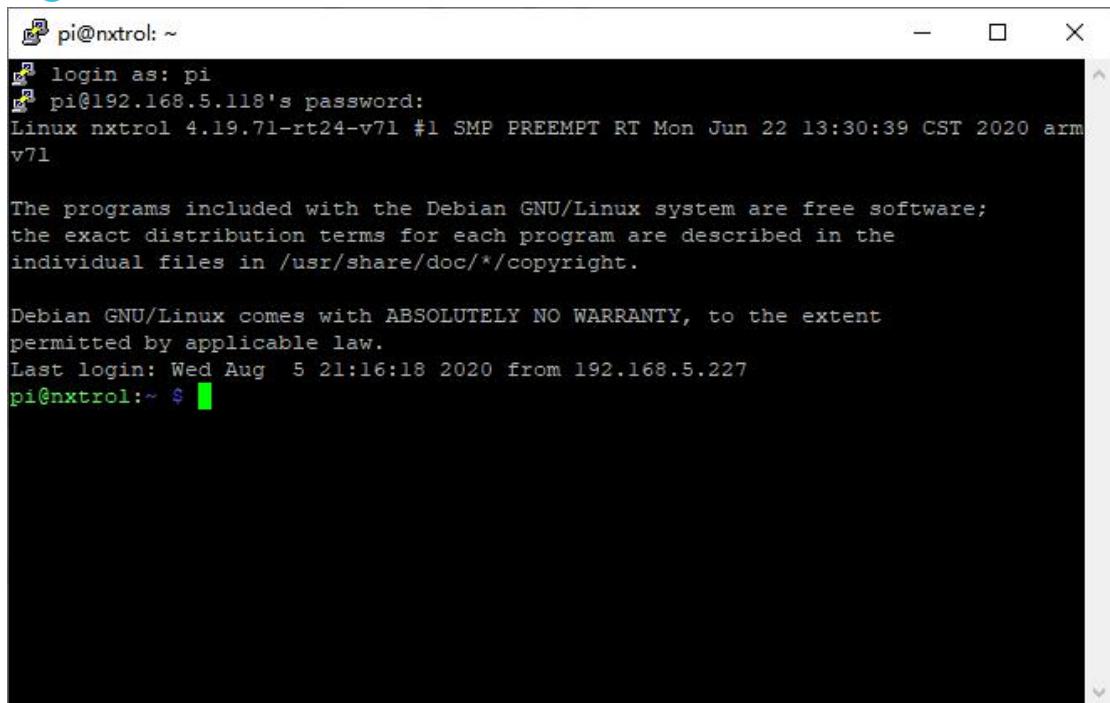


| 扫描结果 |     |               |      |      |      |      |
|------|-----|---------------|------|------|------|------|
| 设备名  | 版本  | IP地址          | 新建工程 | 导入工程 | 应用名称 | 应用版本 |
| 树莓派  | 2.2 | 192.168.5.118 |      |      |      | N/A  |

获得树莓派 4B 的 IP 地址后，可以通过 SSH 串口终端（推荐使用 Putty 软件）远程连接至树莓派 4B，如下图所示：



输入用户名:pi，密码:nxtrol 后即可进入树莓派 4B 的 Linux 系统终端。

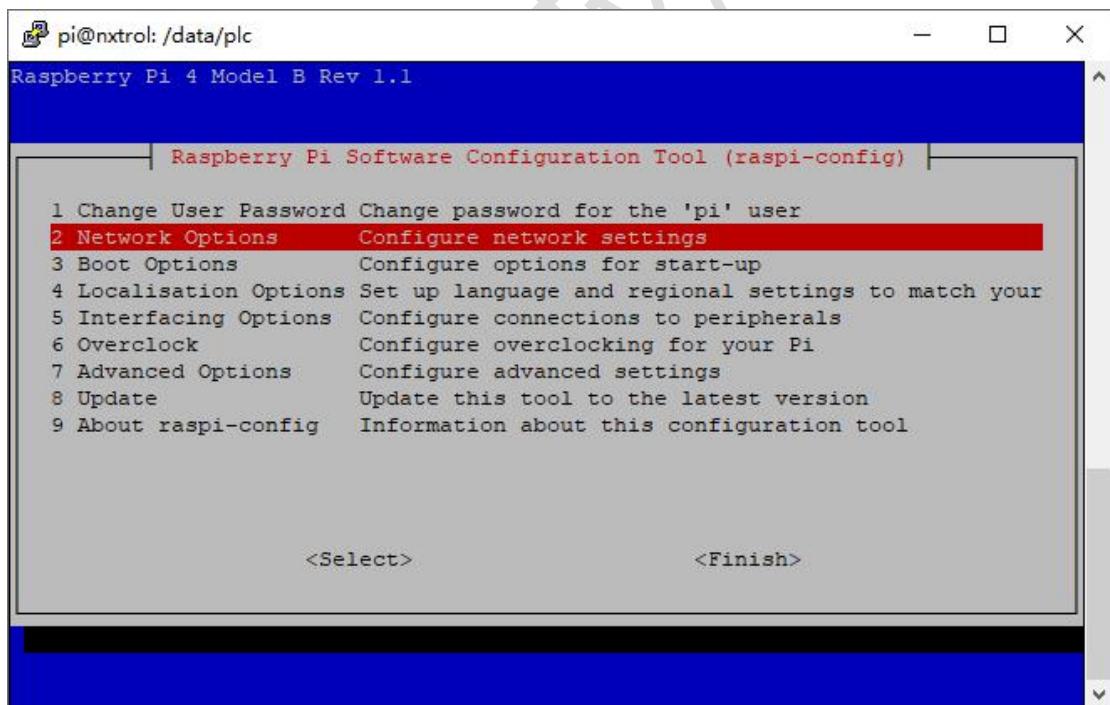


```
pi@nxtrol: ~
pi login as: pi
pi@192.168.5.118's password:
Linux nxtrol 4.19.71-rt24-v7l #1 SMP PREEMPT RT Mon Jun 22 13:30:39 CST 2020 arm
v7l

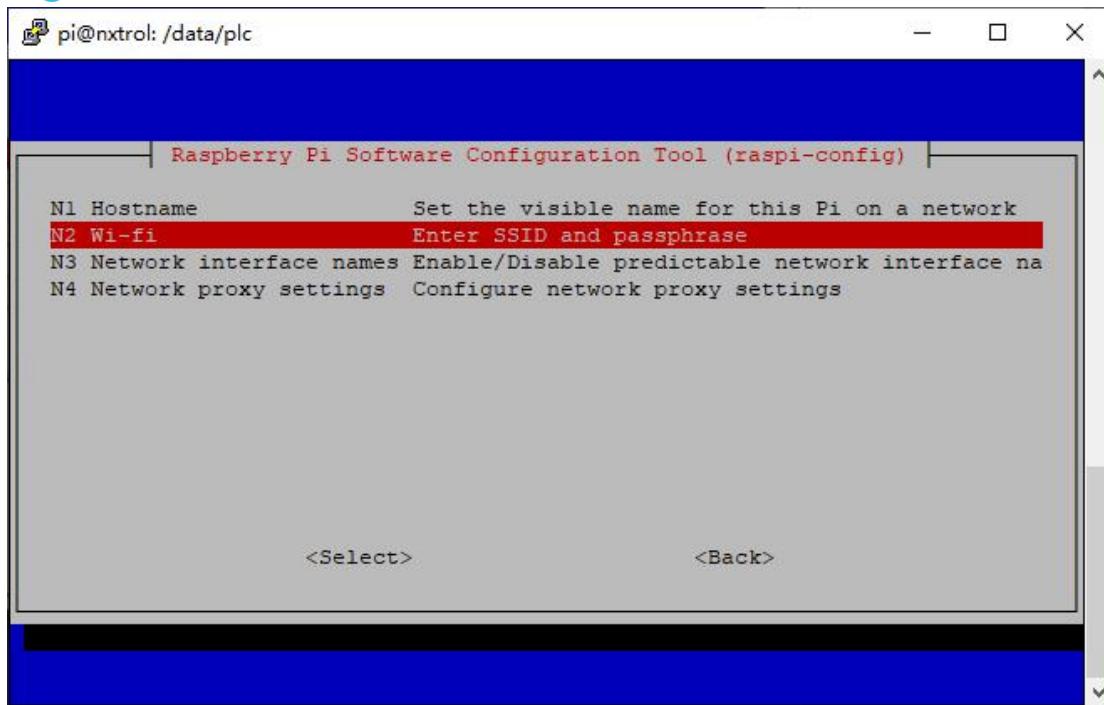
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Aug  5 21:16:18 2020 from 192.168.5.227
pi@nxtrol:~ $
```

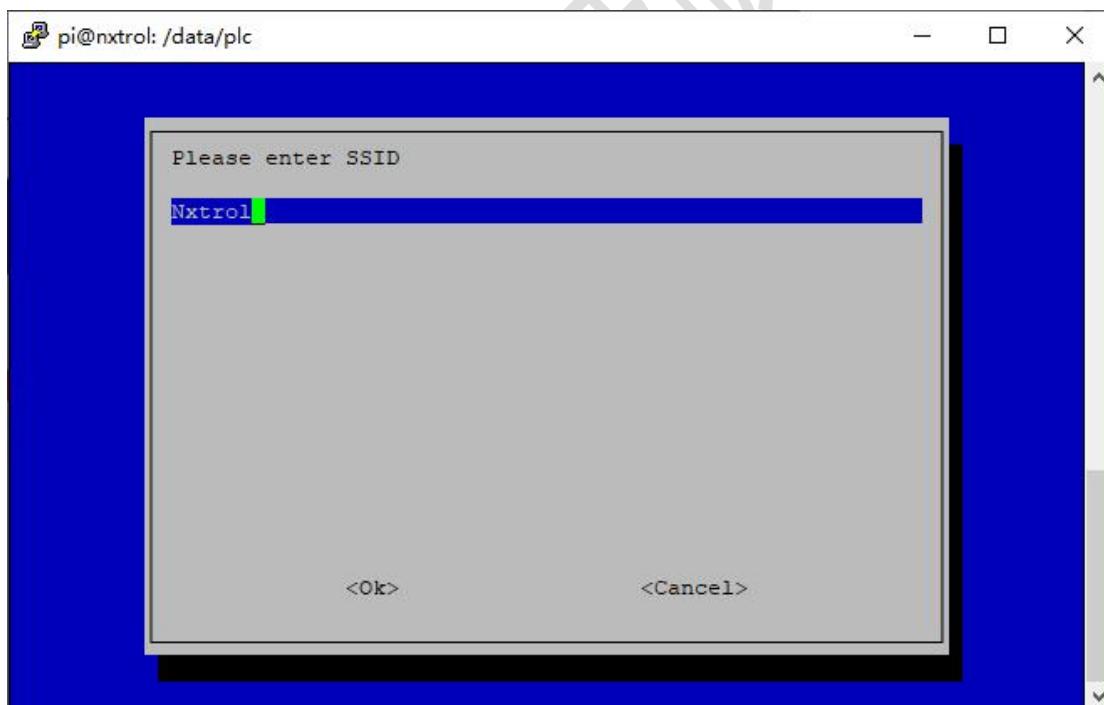
通过输入下面命令开启树莓派设置界面: sudo raspi-config, 如下图所示:



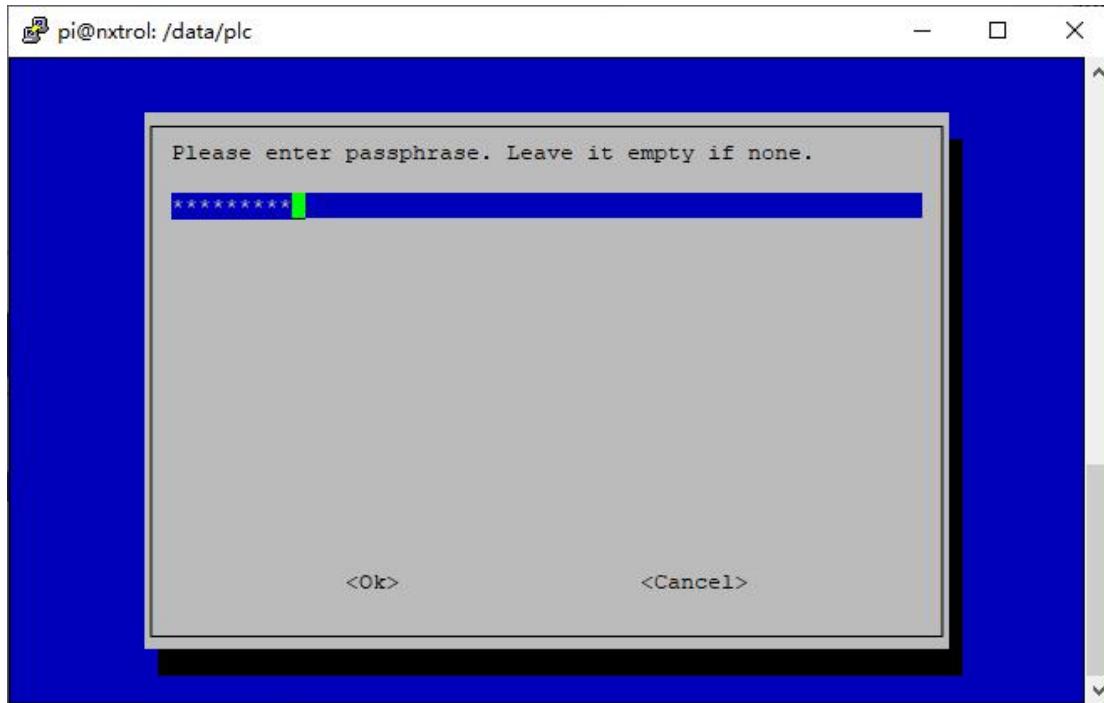
选择 Network Options 后, 按回车键可以进入网络设置界面, 如下图所示:



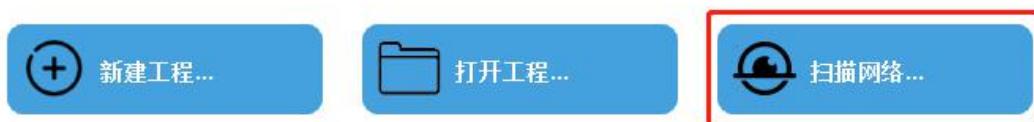
选择 Wi-fi 选项，回车键进入 WIFI 用户名与密码设置，如下图所示：



输入无线路由器的 WIFI 名称 SSID 以及在后续界面中设定 WIFI 密码(请按照当前使用的无线路由器实际值来进行设定)，如下图所示：



设定 WIFI 信息完成后，键盘选择 OK，并在配置主界面中选择 Finish 完成 WIFI 设置，此时建议断开树莓派 4B 硬件与无线路由器的网线后，重新上电，并等待约 60s 后，再次使用 LogicLab 扫描网络，此时如一切正常的话，LogicLab 可以再次扫描到树莓派 4B 的设备，此时可以确定树莓派 4B 的 WIFI 工作正常。



| 扫描结果 |     |               |   |   |   |      |     |
|------|-----|---------------|---|---|---|------|-----|
| 设备名  | 版本  | IP地址          | 新建工程  | 导入工程  | 应用名称  | 应用版本 |     |
| 树莓派  | 2.2 | 192.168.5.118 |  |  |  | N/A  | N/A |

如发生扫描不到树莓派设备的情况，请参考如下步骤进行排查：

1. 树莓派 4B 的 WIFI 连接到的无线路由器名称 SSID 与密码是否正确。

2. 树莓派 4B 是否与运行 LogicLab Automation Suite 的 PC 同时连接至同一个无线路由器。

3. 插上树莓派 4B 的网线至无线路由器上，重复以上步骤。

### 5.5.2. 树莓派 4B EtherCAT 接口与 PLC 配置文件设置

注意此步骤会修改树莓派 4B 的以太网接口信息，必须在 5.4.1 章节步骤成功完成后才可以开始。

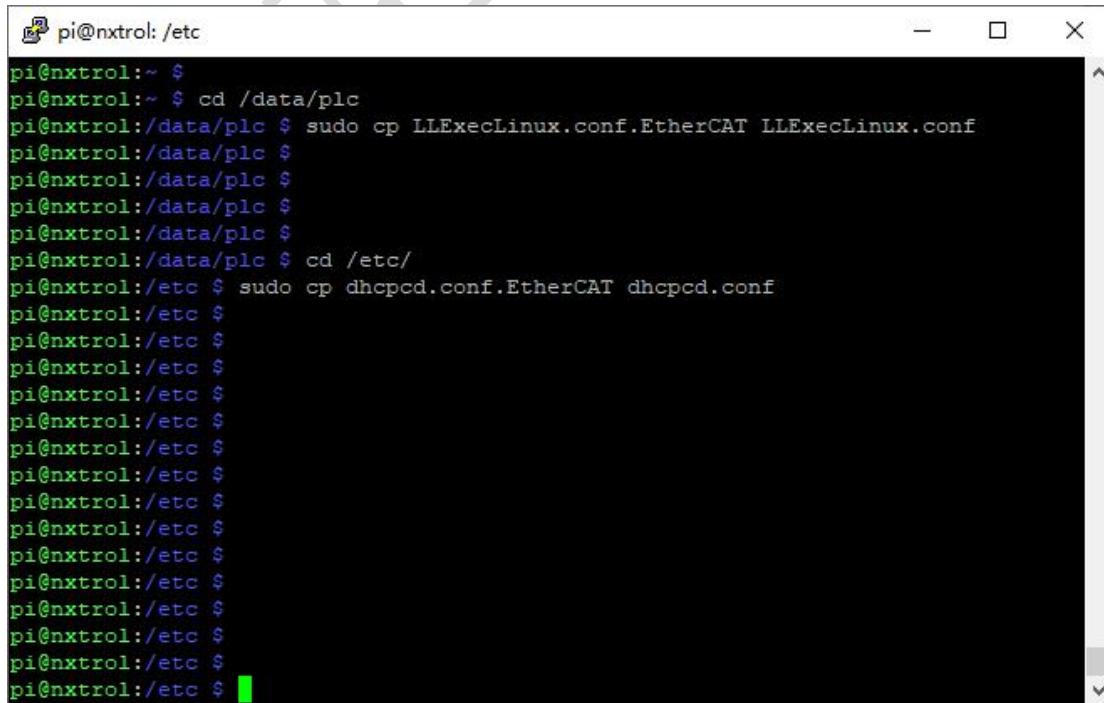
使用 SSH 登录树莓派 4B 的 Linux 系统终端，进行如下配置文件拷贝操作：

```
cd /data/plc
```

```
sudo cp LLExecLinux.conf.EtherCAT LLExecLinux.conf
```

```
cd /etc
```

```
sudo cp dhcpcd.conf.EtherCAT dhcpcd.conf
```



```
pi@nxtrol:~ $ cd /data/plc
pi@nxtrol:/data/plc $ sudo cp LLExecLinux.conf.EtherCAT LLExecLinux.conf
pi@nxtrol:/data/plc $ 
pi@nxtrol:/data/plc $ 
pi@nxtrol:/data/plc $ 
pi@nxtrol:/data/plc $ cd /etc/
pi@nxtrol:/etc $ sudo cp dhcpcd.conf.EtherCAT dhcpcd.conf
pi@nxtrol:/etc $ 
```

注意使用 sudo 命令进行管理员权限操作，系统会在第一次输入带有 sudo 命令时提示输入密码，密码为:nxtrol

完成上述命令后，EtherCAT 网络接口与 PLC 配置文件修改完成（该命令只需执行一次，如重新烧写树莓派 4B 的 TF 卡镜像后，需要重新 5.4.1 与 5.4.2 章节的操作），重新上电树莓派 4B 硬件，等待约 60 秒后，可以使用 LogicLab 软件执行网络扫描功能，扫描结果如下图所示：



此时树莓派 4B 的设备名称改变为“树莓派 EtherCAT”，并且可以通过新建工程按钮直接新建适用于该树莓派 4B 的 PLC 工程（通过在线扫描方式新建工程，工程通讯参数已经自动设定，无需手动设定通讯参数）。

### 5.5.3. LogicLab 中使用 EtherCAT 主站功能

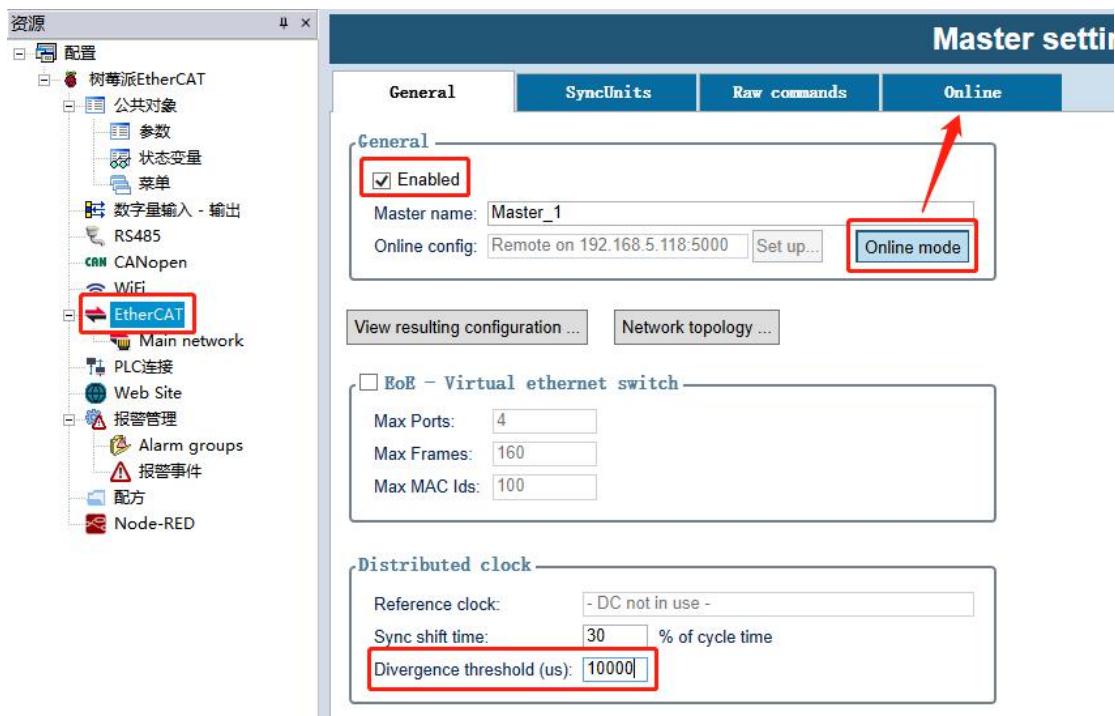
通过正确新建树莓派 4B 工程后，首先需要在如下位置导入需要使用到的 EtherCAT 从站设备 ESI 文件：



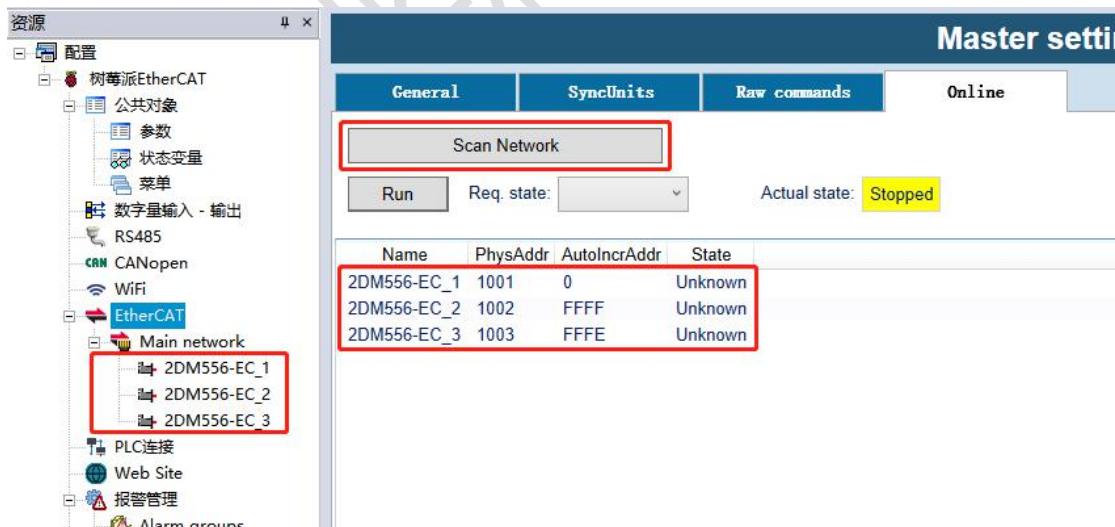
导入 ESI 文件完成后，鼠标左键选择 EtherCAT 节点下的“Main network”后，右侧的设备目录中将会显示新导入的 ESI 设备与 LogicLab 中内置的 EtherCAT 设备名称。



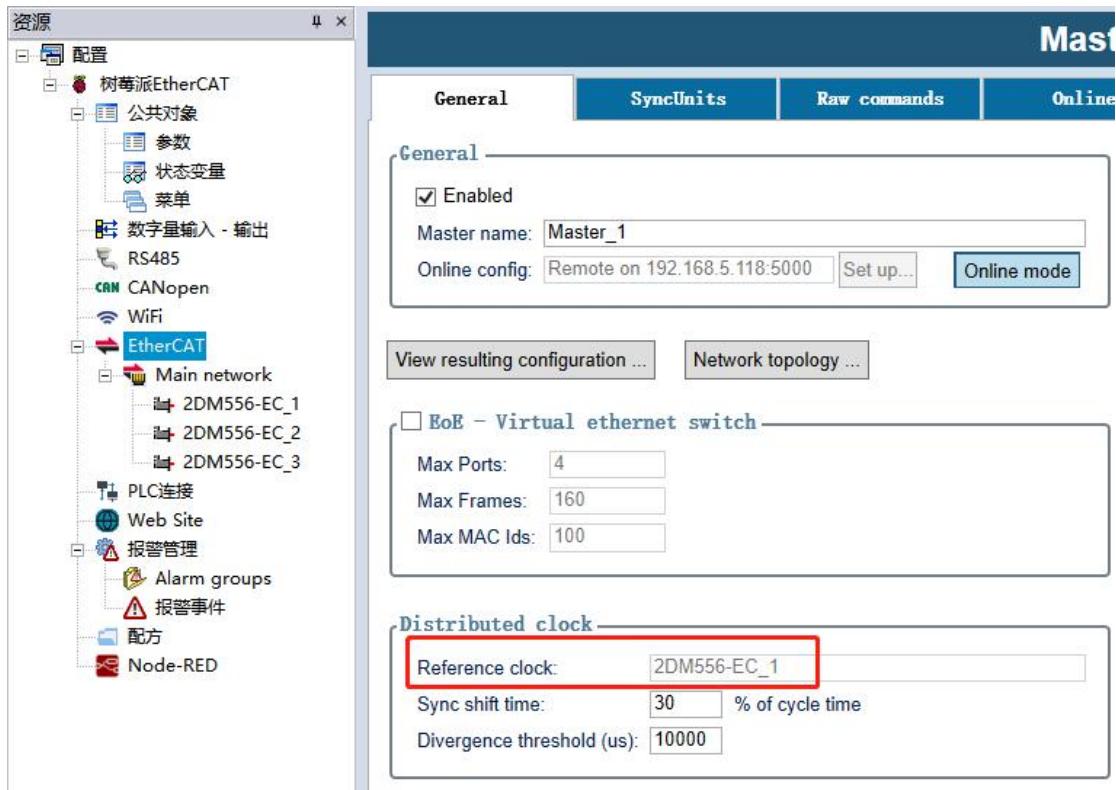
鼠标左键选择“EtherCAT”节点，通过勾选“Enabled”激活树莓派 4B 的 EtherCAT 主站功能，并设定“Divergence threshold”值为 10000us（由于树莓派 4B 的实时 Linux 系统使用 PREEMPT-RT 补丁，最差 Linux 系统抖动约为 100us-200us 左右，因此该参数默认值 250us 并不能很好支持开启树莓派 DC 分布式时钟功能，如使用 Xenomai Linux 实时扩展则可以保持默认值 250us）。选择“Online mode”后，可以切换至 EtherCAT Online 模式：



进入在线模式后，点击“Scan Network”扫描树莓派 4B 以太网口连接的所有 EtherCAT 设备（注意所有已连接的设备必须预先导入其 ESI 配置文件，否则可能无法扫描到对应的设备）。

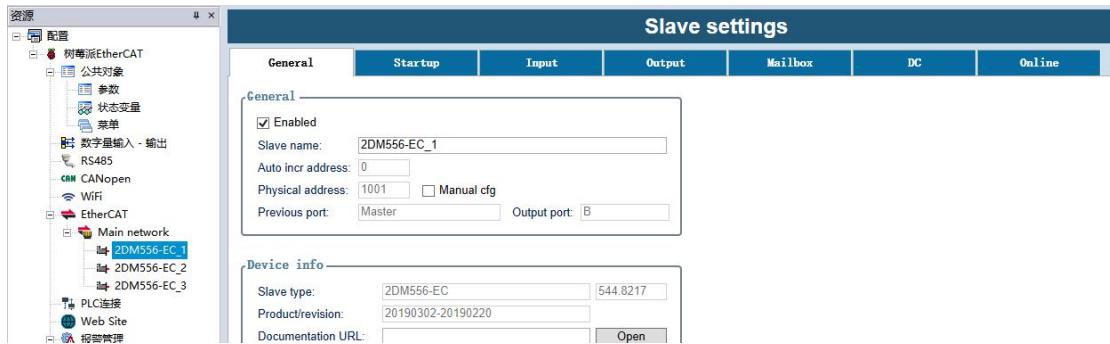


如果已经扫描到设备支持 DC 分布式时钟，那么 DC 默认参考时钟为第一个支持 DC 分布式时钟的 EtherCAT 从站，LogicLab 会自动进行设置：

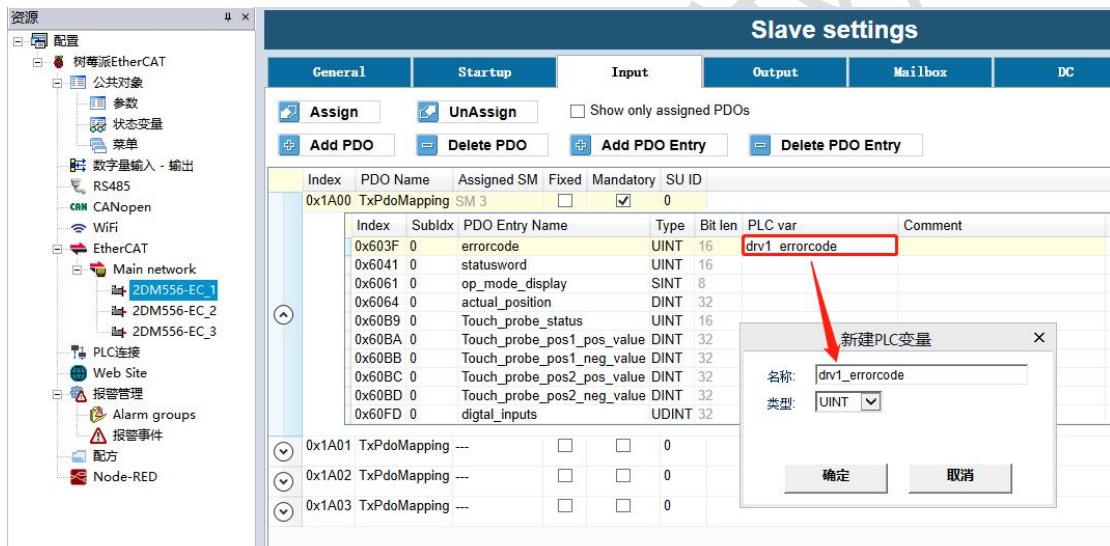


选择对应的 EtherCAT 从站设备，可以依次切换至对应的选项卡进行信息查看或者参数设定：

1. General - 基础信息查看
2. Startup - 初始化命令设置
3. Input - 输入变量映射
4. Output - 输出变量映射
5. Mailbox - 邮箱参数设置
6. DC - 分布式时钟设置
7. Online - 在线对象字典诊断



EtherCAT 输入/输出与 PLC 变量关联非常简单与方便，可以在 Input 以及 Output 中的“PLC var”选项中直接输入期望的 PLC 变量名，按回车键即可将 EtherCAT PDO 与 PLC 变量关联，如下图所示：

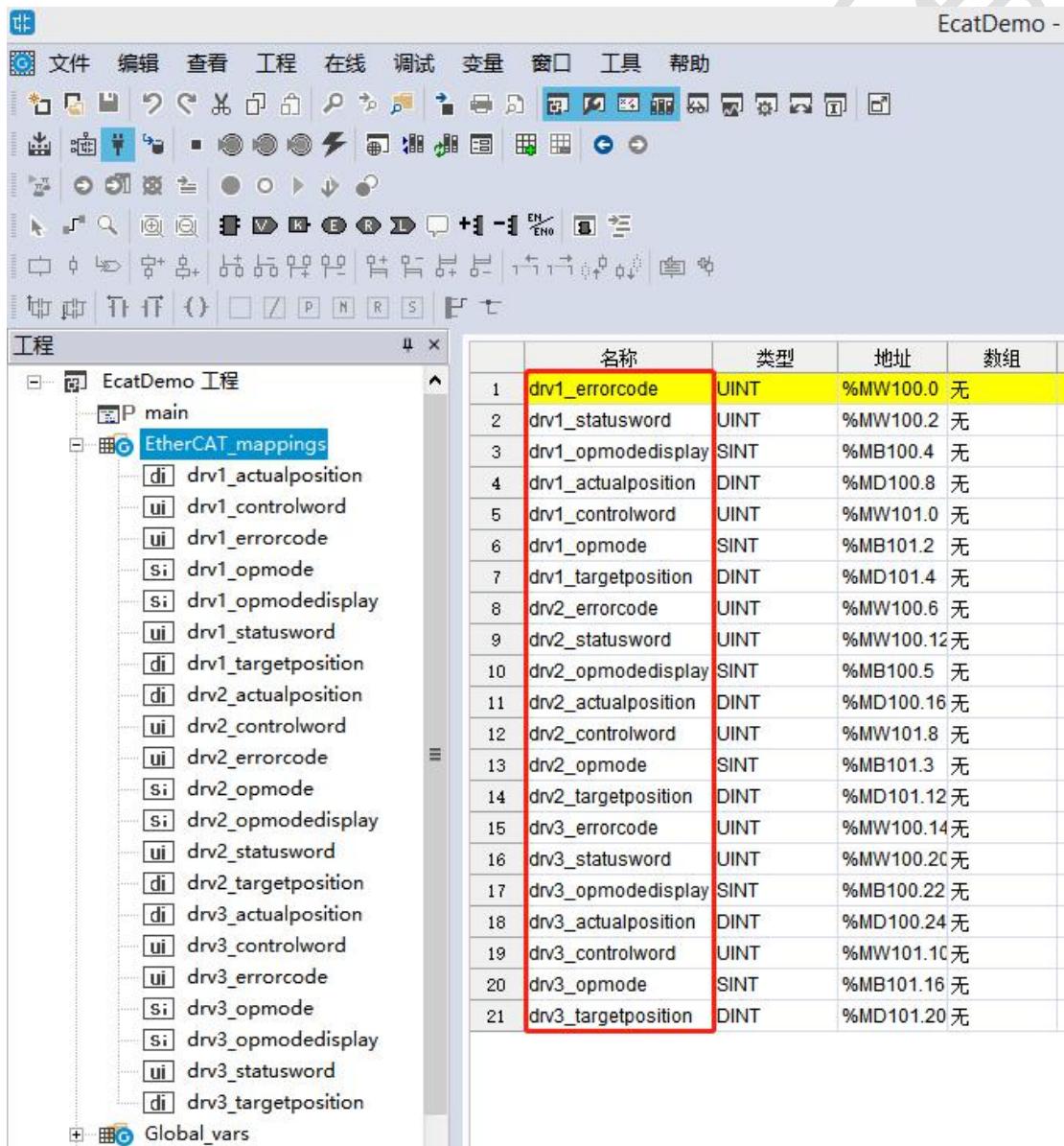


将所有需要的 EtherCAT PDO 对象新建对应的 PLC 变量与其关联后，在全局变量表格中可以找到对应的变量定义，后续操作 EtherCAT 从站设备可通过对这些 PLC 变量进行编程。

编译下载 PLC 工程后，系统可能提示如下信息，直接选择“是”即可，等待 PLC 自动重新启动以及加载新设定的配置文件：



并且可以通过切换到变量列表在线监控模式直接进行监控，如下图所示：



The screenshot shows the EcatDemo software interface. The title bar says "EcatDemo -". The menu bar includes: 文件 (File), 编辑 (Edit), 查看 (View), 工程 (Project), 在线 (Online), 调试 (Debug), 变量 (Variables), 窗口 (Windows), 工具 (Tools), 帮助 (Help). The toolbar below has various icons for file operations, search, and monitoring.

The left sidebar shows the project structure under "工程":

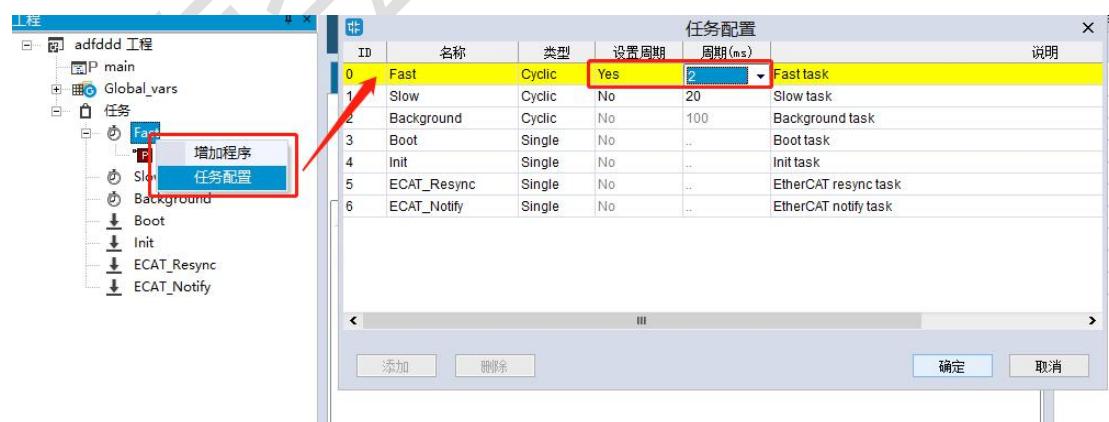
- EcatDemo 工程
  - main
  - EtherCAT\_mappings
    - drv1\_actualposition
    - drv1\_controlword
    - drv1\_errorcode
    - drv1\_opmode
    - drv1\_opmodedisplay
    - drv1\_statusword
    - drv1\_targetposition
    - drv2\_actualposition
    - drv2\_controlword
    - drv2\_errorcode
    - drv2\_opmode
    - drv2\_opmodedisplay
    - drv2\_statusword
    - drv2\_targetposition
    - drv3\_actualposition
    - drv3\_controlword
    - drv3\_errorcode
    - drv3\_opmode
    - drv3\_opmodedisplay
    - drv3\_statusword
    - drv3\_targetposition
  - Global\_vars

The main area displays a table of variables:

|    | 名称                  | 类型   | 地址        | 数组 |
|----|---------------------|------|-----------|----|
| 1  | drv1_errorcode      | UINT | %MW100.0  | 无  |
| 2  | drv1_statusword     | UINT | %MW100.2  | 无  |
| 3  | drv1_opmodedisplay  | SINT | %MB100.4  | 无  |
| 4  | drv1_actualposition | DINT | %MD100.8  | 无  |
| 5  | drv1_controlword    | UINT | %MW101.0  | 无  |
| 6  | drv1_opmode         | SINT | %MB101.2  | 无  |
| 7  | drv1_targetposition | DINT | %MD101.4  | 无  |
| 8  | drv2_errorcode      | UINT | %MW100.6  | 无  |
| 9  | drv2_statusword     | UINT | %MW100.12 | 无  |
| 10 | drv2_opmodedisplay  | SINT | %MB100.5  | 无  |
| 11 | drv2_actualposition | DINT | %MD100.16 | 无  |
| 12 | drv2_controlword    | UINT | %MW101.8  | 无  |
| 13 | drv2_opmode         | SINT | %MB101.3  | 无  |
| 14 | drv2_targetposition | DINT | %MD101.12 | 无  |
| 15 | drv3_errorcode      | UINT | %MW100.14 | 无  |
| 16 | drv3_statusword     | UINT | %MW100.20 | 无  |
| 17 | drv3_opmodedisplay  | SINT | %MB100.22 | 无  |
| 18 | drv3_actualposition | DINT | %MD100.24 | 无  |
| 19 | drv3_controlword    | UINT | %MW101.10 | 无  |
| 20 | drv3_opmode         | SINT | %MB101.16 | 无  |
| 21 | drv3_targetposition | DINT | %MD101.20 | 无  |

| 符号                   | 值     | 类型   | 位置     |  |
|----------------------|-------|------|--------|--|
| —DRV1_CONTROLWORD    | 15    | UINT | global |  |
| —DRV1_OPMODE         | 8     | SINT | global |  |
| —DRV1_TARGETPOSITION | 0     | DINT | global |  |
| —DRV1_ERRORCODE      | 20480 | UINT | global |  |
| —DRV1_STATUSWORD     | 5687  | UINT | global |  |
| —DRV1_OPMODEDISPLAY  | 8     | SINT | global |  |
| —DRV1_ACTUALPOSITION | 0     | DINT | global |  |
| —DRV2_CONTROLWORD    | 15    | UINT | global |  |
| —DRV2_OPMODE         | 8     | SINT | global |  |
| —DRV2_TARGETPOSITION | 0     | DINT | global |  |
| —DRV2_ERRORCODE      | 20480 | UINT | global |  |
| —DRV2_STATUSWORD     | 5687  | UINT | global |  |
| —DRV2_OPMODEDISPLAY  | 8     | SINT | global |  |
| —DRV2_ACTUALPOSITION | 0     | DINT | global |  |
| —DRV3_CONTROLWORD    | 15    | UINT | global |  |
| —DRV3_OPMODE         | 8     | SINT | global |  |
| —DRV3_TARGETPOSITION | 0     | DINT | global |  |
| —DRV3_ERRORCODE      | 20480 | UINT | global |  |
| —DRV3_STATUSWORD     | 5687  | UINT | global |  |
| —DRV3_OPMODEDISPLAY  | 8     | SINT | global |  |
| —DRV3_ACTUALPOSITION | 0     | DINT | global |  |

EtherCAT 任务周期与 Fast 任务一致，如需要改变 EtherCAT 任务周期可以直接修改 Fast 任务的周期，重新编译下载工程：



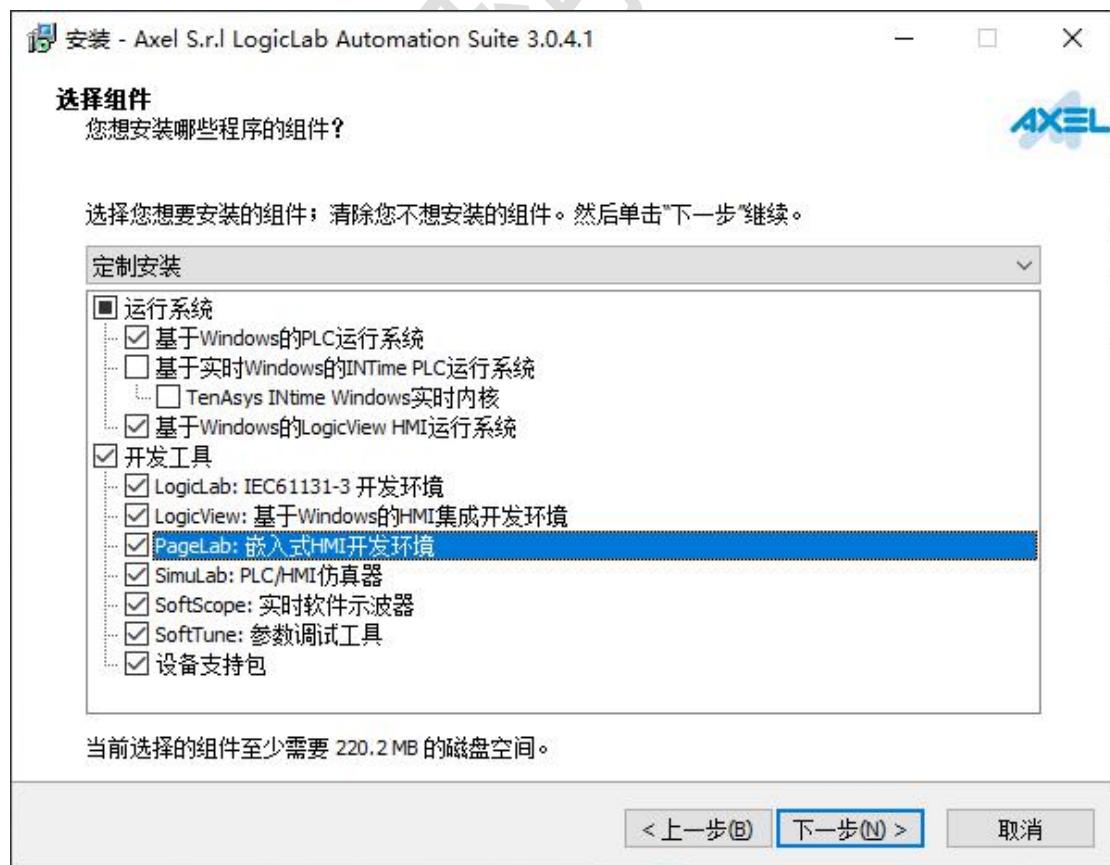
## 6. Q&A

### 6. 1. LogicLab 编程工具如何下载？如何安装？

登录 <https://www.nxtrol.com/download.html> 下载 LogicLab Automation Suite 最新版本（实际 LogicLab 安装文件会持续更新，发布版本可能会有不一样）：



基于树莓派 PLC 目前支持的功能，在安装 LogicLabAutomationSuite 过程中选择如下组件：



根据提示一步一步完成安装。

## 6. 2. 树莓派 PLC 镜像下载地址?

登录 <https://www.nxtrol.com/download.html> 根据使用的树莓派硬件型号进行下载（实际镜像文件会持续更新，发布日期可能会有不一样）：

| 标准树莓派镜像（适用于树莓派3B/3B+/4B）       |          |    |  |
|--------------------------------|----------|----|--|
| 标准树莓派PLC镜像20200731.7z.001      | 1000.0MB | 下载 |  |
| 标准树莓派PLC镜像20200731.7z.002      | 1000.0MB | 下载 |  |
| 标准树莓派PLC镜像20200731.7z.003      | 443.77MB | 下载 |  |
| 标准树莓派镜像（适用于欧比特测控EdgeBox-RPI）   |          |    |  |
| EdgeBox树莓派PLC镜像20200731.7z.001 | 1000.0MB | 下载 |  |
| EdgeBox树莓派PLC镜像20200731.7z.002 | 1000.0MB | 下载 |  |
| EdgeBox树莓派PLC镜像20200731.7z.003 | 442.42MB | 下载 |  |

下载完毕后使用 7ZIP 或者 360 压缩工具进行解压缩获得最终镜像。镜像文件中包含了存放有校验码的文件“Hash 校验码.txt”，可以通过网站下方的辅助工具包中的 HASH 工具对解压缩出来的 img 镜像文件进行校验，如果获得的校验码与“Hash 校验码.txt”中的值不一样，则该镜像解压缩过程中可能存在数据不一致情况，此时需要重新解压缩或者下载上述镜像文件重新解压缩。

## 6. 3. 如何购买正式版的树莓派 PLC 授权

试用版本的树莓派 PLC 内核会在运行 1 个小时左右停机一次，需要重启系统后才可以继续使用，试用版本功能上没有任何限制。购买树莓派 PLC 内核授权可以没有任何时间限制运行 PLC 内核，在线购买链接地址如下：

[http://www.nxtrol.com/product\\_rpi.html](http://www.nxtrol.com/product_rpi.html)

进入购买页面后，点击“立即购买”，并填写好联系人手机（用于我们与您取得联系）以及邮箱地址（用于我们发送授权码到您邮箱），由于授权码可以与任何树莓派硬件完全绑定，因此不支持退货，因此购买前建议在试用版中详细进行评估。目前在线支付方式仅仅支持微信支付：



提交表单

注意事项  
该产品仅为基于树莓派硬件的PLC内核软件授权码，并不包含树莓派硬件本身，用户需要根据自己需求额外购买树莓派3B/3B+/4B/欧比特测控EdgeBox-RPI

联系手机  
 \*

邮箱地址  
 \*

验证码  
 8k5f [换一张](#) \*

支付金额： ￥ 350.00 [在线支付](#)

**提交**

## 6. 4. 树莓派授权激活

激活需要在翌控科技官网购买的授权码，如果未拥有授权码，请前往翌控科技官方网站购买。  
一个授权码只和一个树莓派绑定，可以重复激活，但不能激活多台设备。

### 6. 4. 1. LogicLab 在线激活:

打开 logiclab, 连接控制器至在线状态, 在资源窗口双击树莓派打开树莓派配置界面, 输入

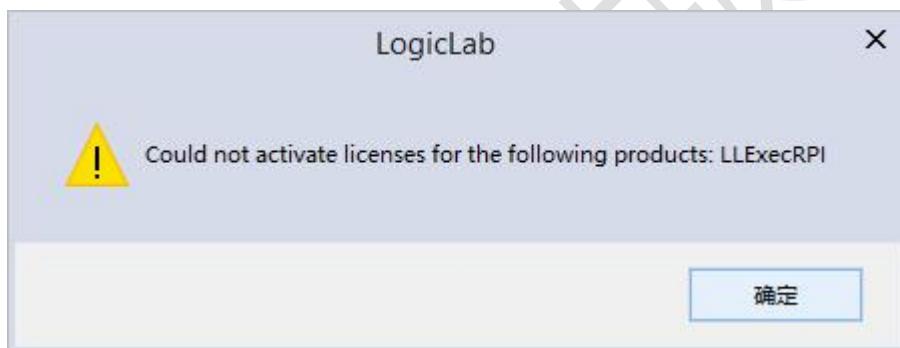
授权码输入栏输入授权码后, 点击激活。如下图所示



重启树莓派设备, 刷新状态, 激活成功, 如下图所示:

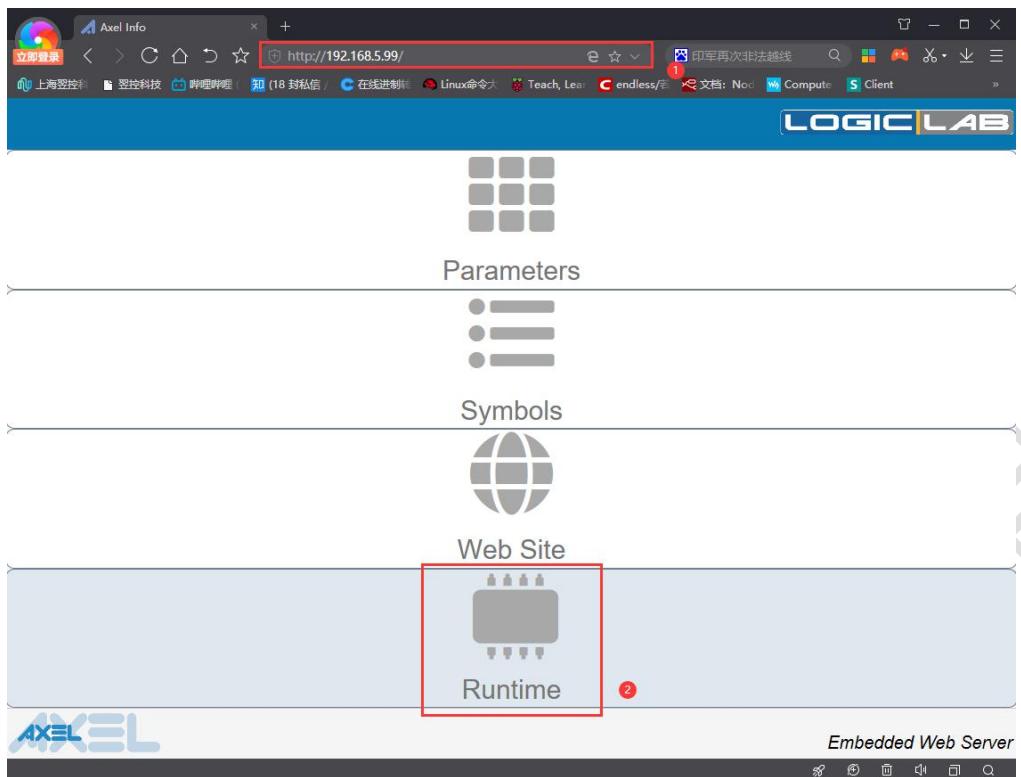


网络状态不佳，可能导致激活失败，出现如下图所示提示，可尝试进行多次重复激活，如仍未成功，可采用网页服务器的方式进行激活。

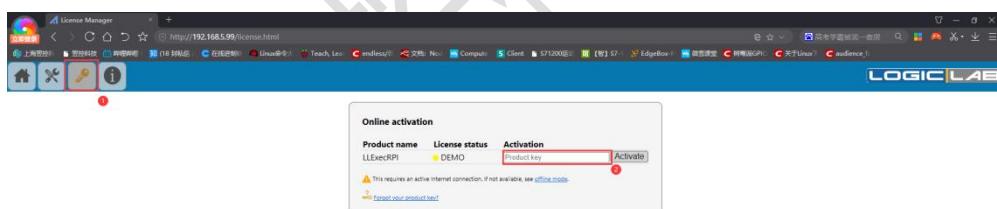


## 6. 4. 2. 网页服务器激活：

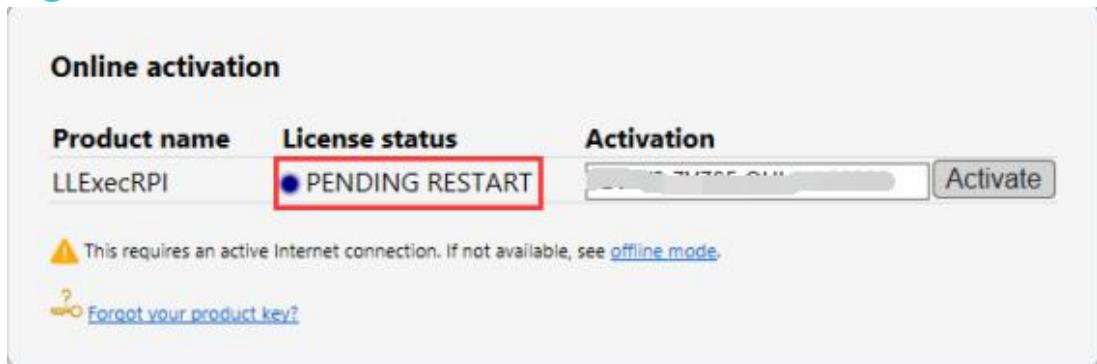
树莓派可上外网的状态下，在网页网址栏输入设备的 IP 地址，打开设备的 web server 界面。如下图所示：



打开授权码激活页面，在激活输入栏输入授权码，点击激活。如下图所示：



系统提示激活成功，需重启设备，如下图所示：



重启树莓派设备，刷新网页，激活成功，如下图所示：



## 6. 5. 树莓派 PLC 镜像登录用户名与密码是多少？

翌控科技树莓派 PLC 镜像文件可以通过 SSH 工具进行登录(推荐使用 PUTTY)，其中 Linux 系统的默认用户名与密码如下：

- 默认用户名:pi 密码:nxtrol
- 管理员用户名:root 密码:nxtrol

Node-RED 开发环境默认用户名与密码如下：

- 管理员用户名:admin 密码:nxtrol
- 访客用户名:guest 密码:guest

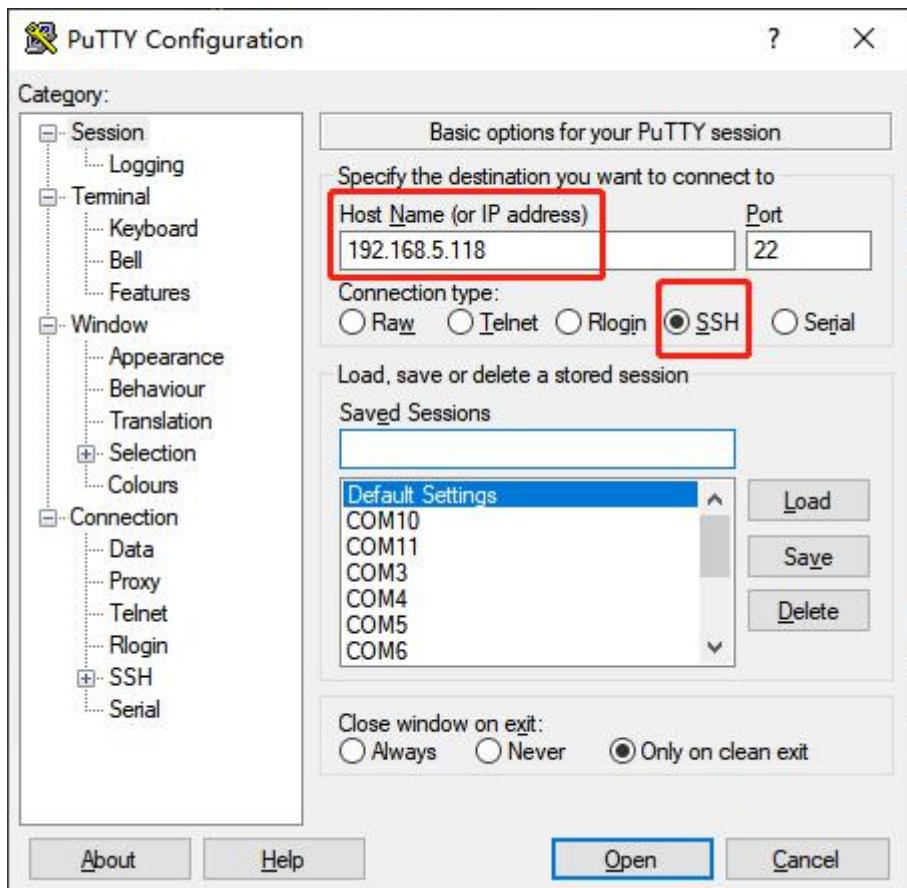
## 6. 6. 通过 ssh 工具登录树莓派命令行终端

树莓派硬件的以太网接口与无线路由器的网口进行连接，同时安装了 LogicLab Automation Suite 的 PC 也通过网线/WIFI 连接至该路由器上，确保 PC 与树莓派硬件处于同一个局域网内，连接树莓派硬件的电源，等待约 60 秒，树莓派硬件内置的 PLC 系统将启动完毕。

打开 LogicLab 编程软件，在主界面上点击“扫描网络”，如果通讯一切正常，那么在扫描结果下会出现当前网络中所有树莓派设备的信息，包括其 IP 地址，因此在任何时候如果忘记树莓派硬件 IP 地址时，都可以使用此方法来进行确认。



获得树莓派硬件的 IP 地址后，可以通过 SSH 串口终端（推荐使用 Putty 软件）远程连接至树莓派，如下图所示：



输入用户名:pi, 密码:nxtrol 后即可进入树莓派硬件的 Linux 系统终端。

```
pi@nxtrol: ~
pi login as: pi
pi@nxtrol's password:
Linux nxtrol 4.19.71-rt24-v7l #1 SMP PREEMPT RT Mon Jun 22 13:30:39 CST 2020 arm
v7l

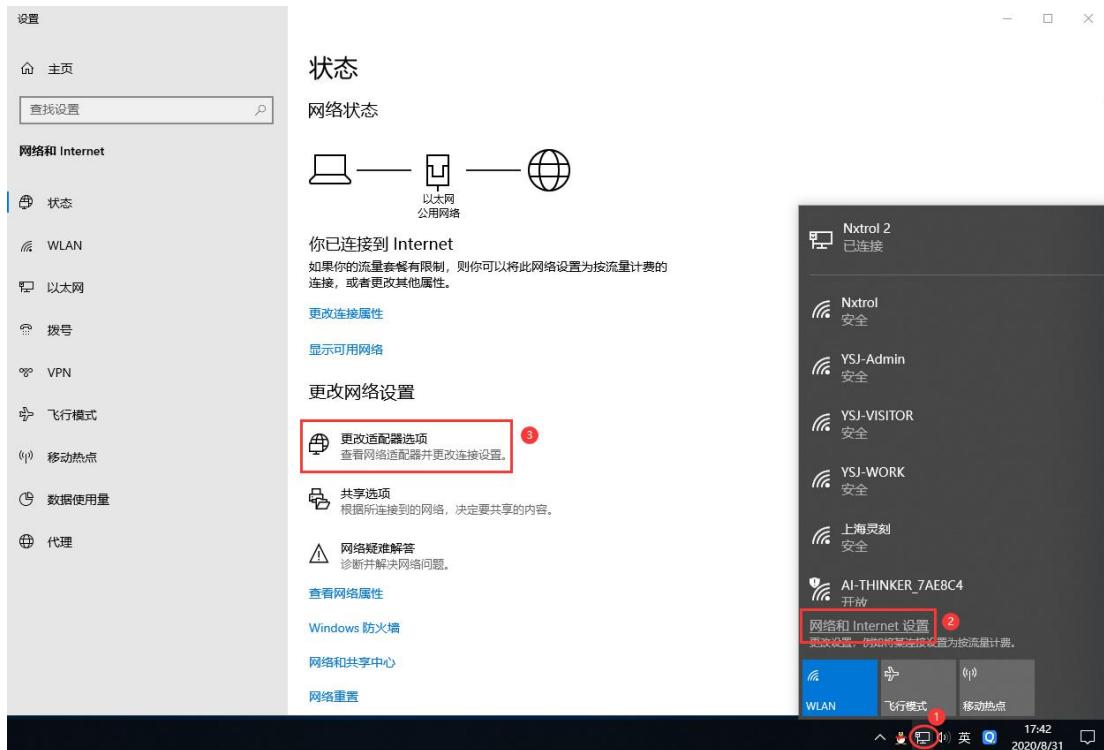
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Aug  5 21:16:18 2020 from 192.168.5.227
pi@nxtrol:~ $
```

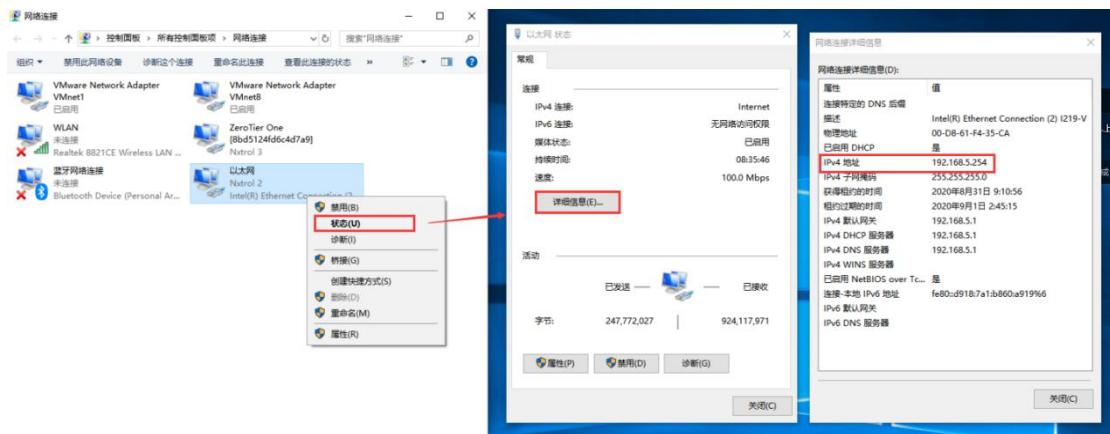
## 6. 7. 树莓派设定固定 IP 地址

树莓派的 IP 地址与 PC 需在同一网段，首先查看 PC 的 IP 地址；

打开网络连接控制面板，如下图所示：



查看 PC 端 IPV4 的地址，如下图所示：



使用 Putty 软件通过 ssh 登录树莓派 Linux 终端,(树莓派初始默认 IP 地址 192.168.5.99)。

输入 sudo nano /etc/dhcpcd.conf 命令开启命令行文本编辑器，在文件结尾添加如下配置：

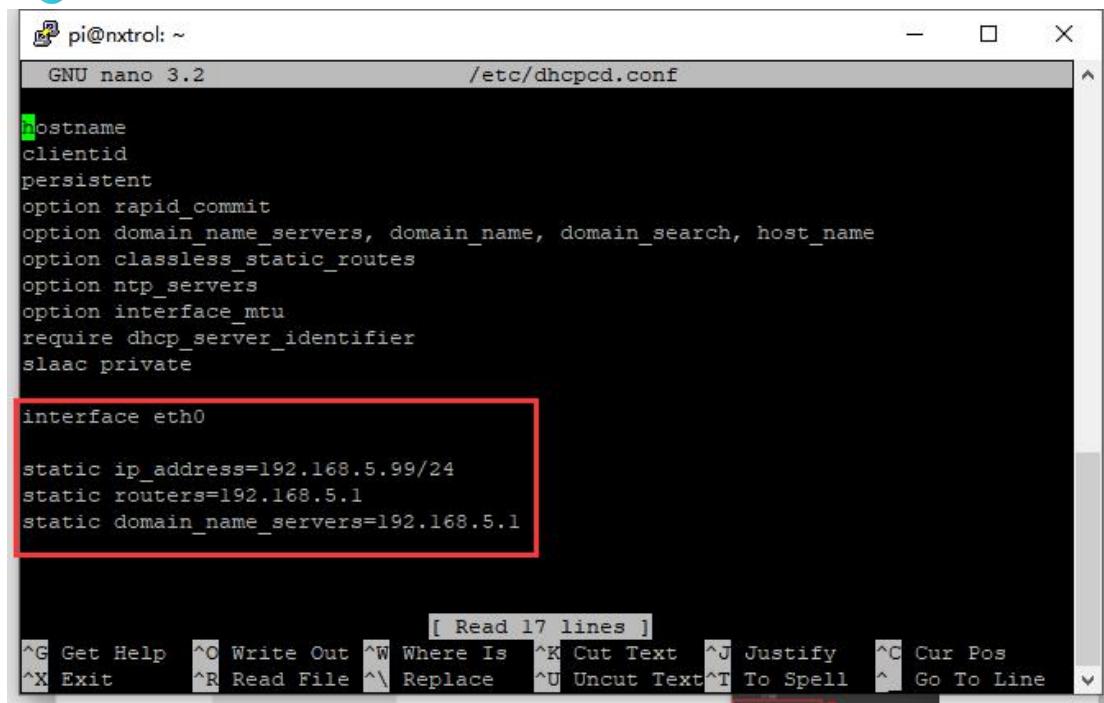
interface eth0 //此处若需要设定 wifi 的固定 IP, 那么 interface 名称为 wlan0

static ip\_address=你的内网 ip 地址/24

static routers=内网网关 ip 地址

static domain\_name\_servers=内网路由器的 IP 地址

如下图所示：



```
pi@nxtrol: ~
GNU nano 3.2                               /etc/dhcpd.conf

hostname
clientid
persistent
option rapid_commit
option domain_name_servers, domain_name, domain_search, host_name
option classless_static_routes
option ntp_servers
option interface_mtu
require dhcp_server_identifier
slaac private

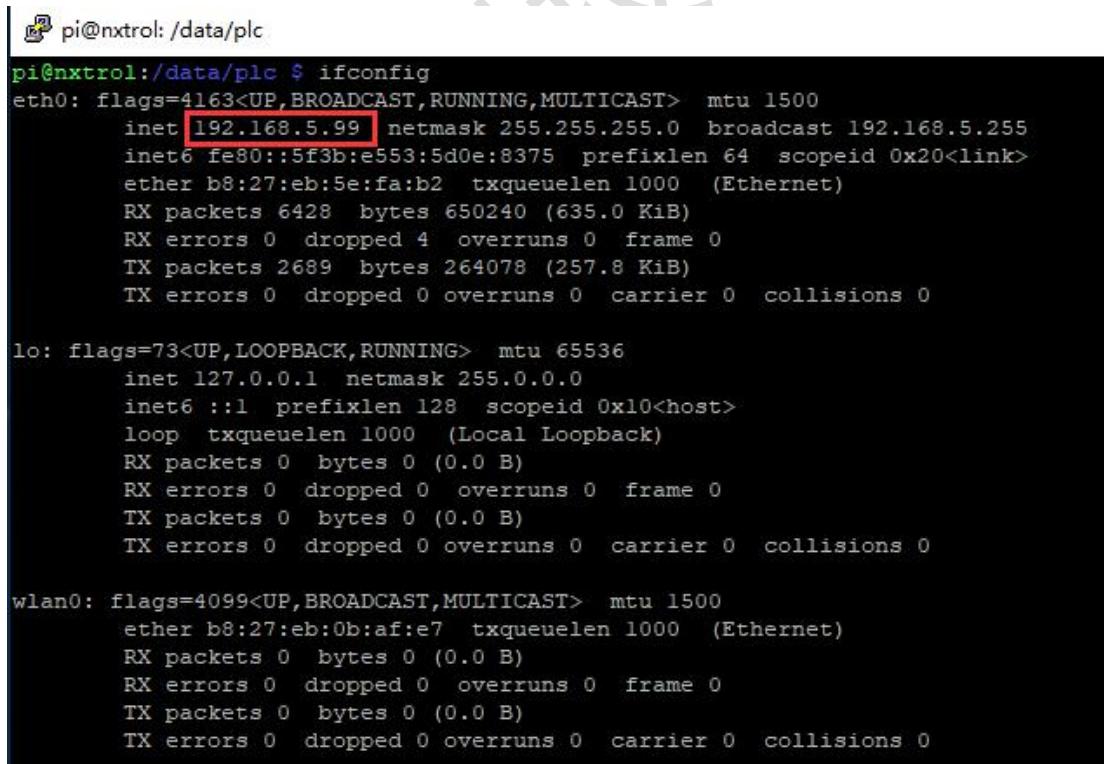
interface eth0

static ip_address=192.168.5.99/24
static routers=192.168.5.1
static domain_name_servers=192.168.5.1

[ Read 17 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^ Go To Line
```

ctrl+o 保存编辑 ctrl+x 退出编辑 重启树莓派

重新登入树莓派终端输入 ifconfig 命令可以查看树莓派 IP 地址，如下图所示：



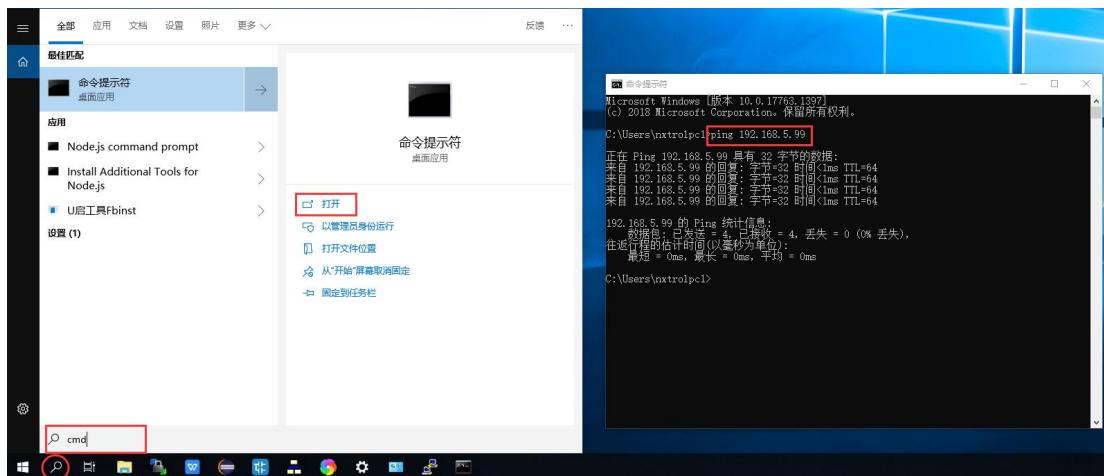
```
pi@nxtrol:/data/plc
pi@nxtrol:/data/plc $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.5.99  netmask 255.255.255.0  broadcast 192.168.5.255
                ether b8:27:eb:5f:3b:e5  txqueuelen 1000  (Ethernet)
                  RX packets 6428  bytes 650240 (635.0 KiB)
                  RX errors 0  dropped 4  overruns 0  frame 0
                  TX packets 2689  bytes 264078 (257.8 KiB)
                  TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
                ether ::1  txqueuelen 1000  (Local Loopback)
                  RX packets 0  bytes 0 (0.0 B)
                  RX errors 0  dropped 0  overruns 0  frame 0
                  TX packets 0  bytes 0 (0.0 B)
                  TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:0b:af:e7  txqueuelen 1000  (Ethernet)
                  RX packets 0  bytes 0 (0.0 B)
                  RX errors 0  dropped 0  overruns 0  frame 0
                  TX packets 0  bytes 0 (0.0 B)
                  TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

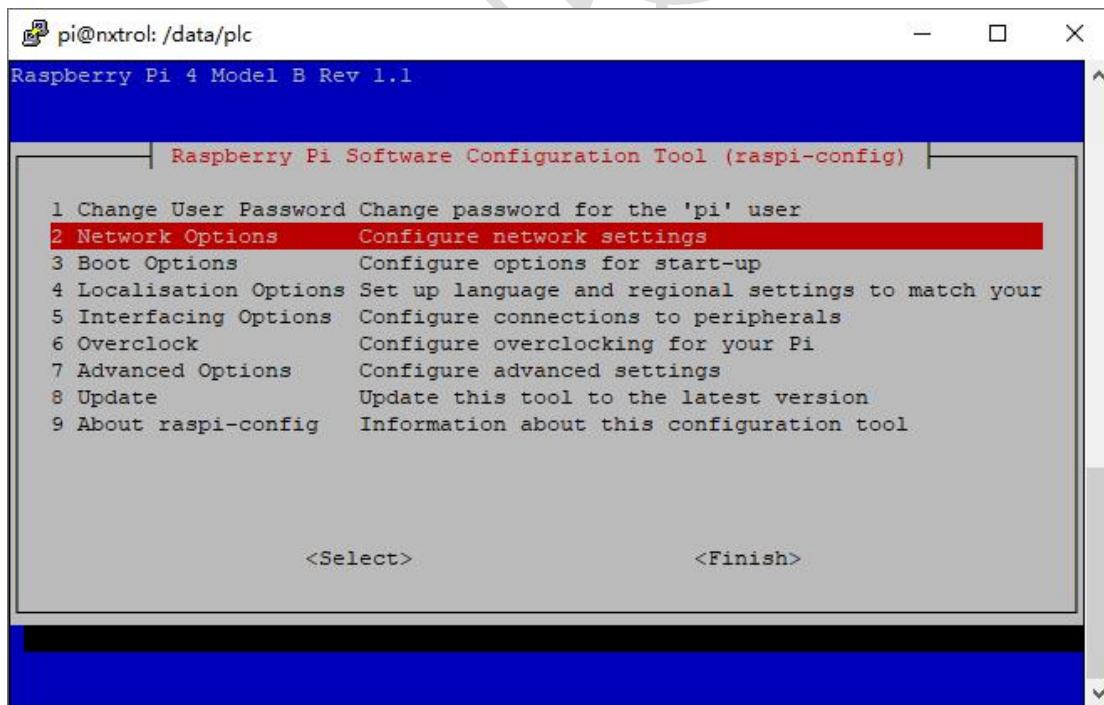
验证树莓派 IP 地址是否设置成功并与电脑是否连接

打开 PC 端命令终端命令终端，输入 ping+IP 地址指令,如下图所示：

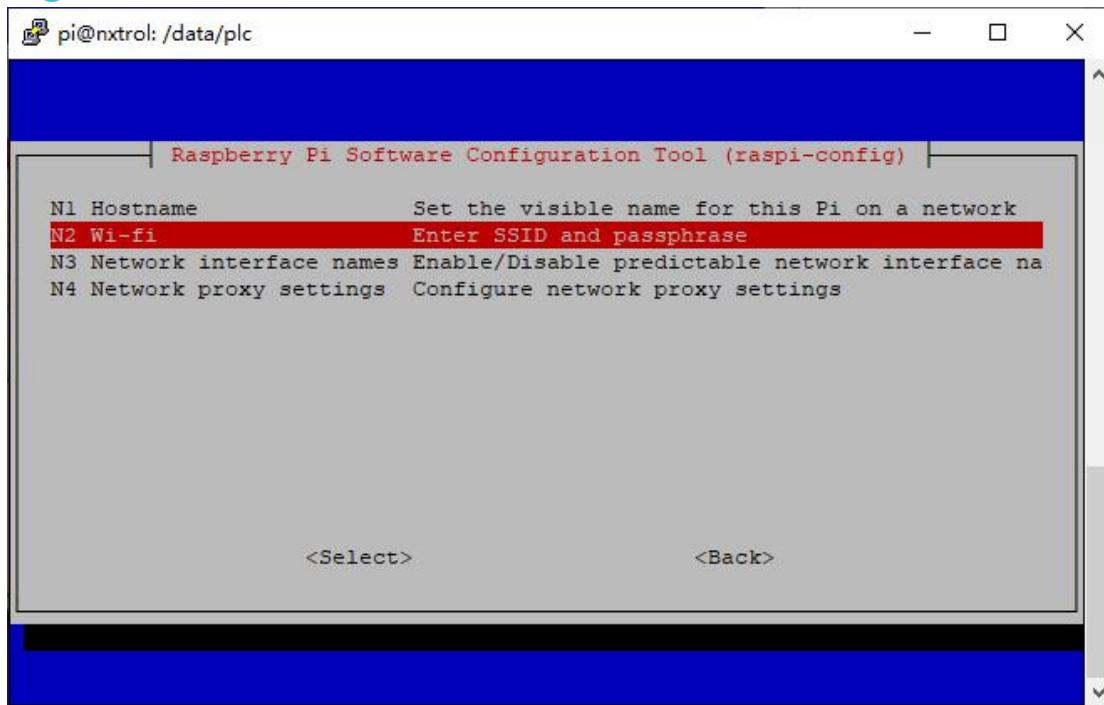


## 6. 8. 启用树莓派的 WIFI 功能 (树莓派 3B/3B+/4B)

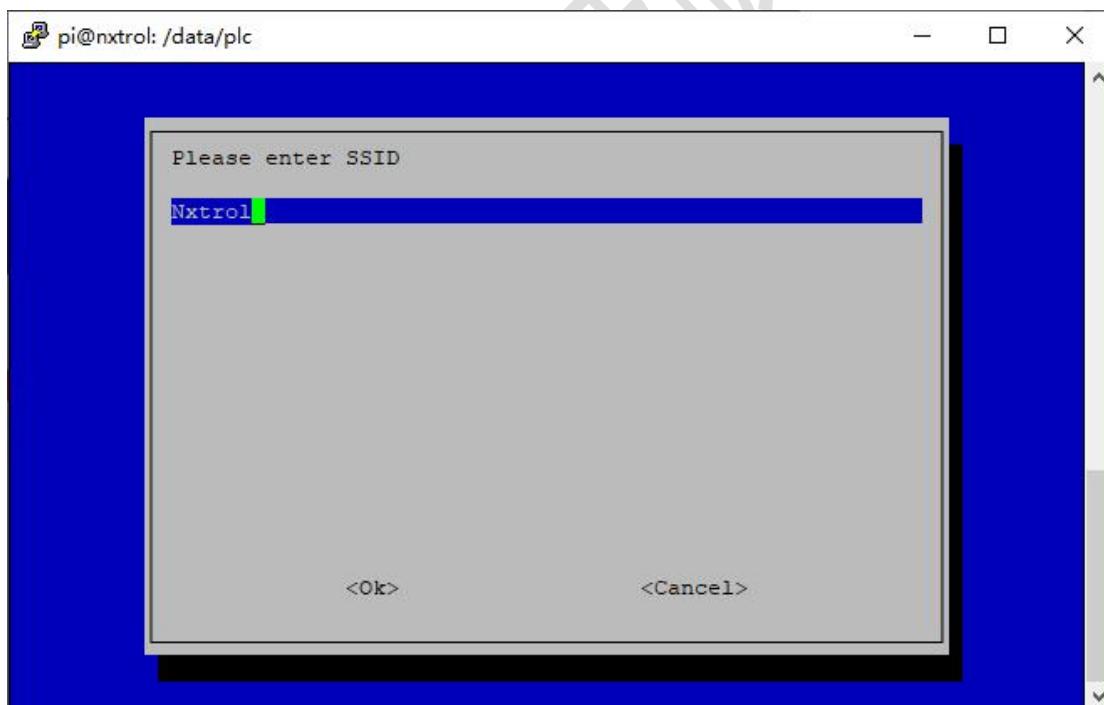
使用 Putty 软件通过 ssh 登录树莓派 Linux 终端，输入下面命令开启树莓派设置界面: sudo raspi-config，如下图所示：



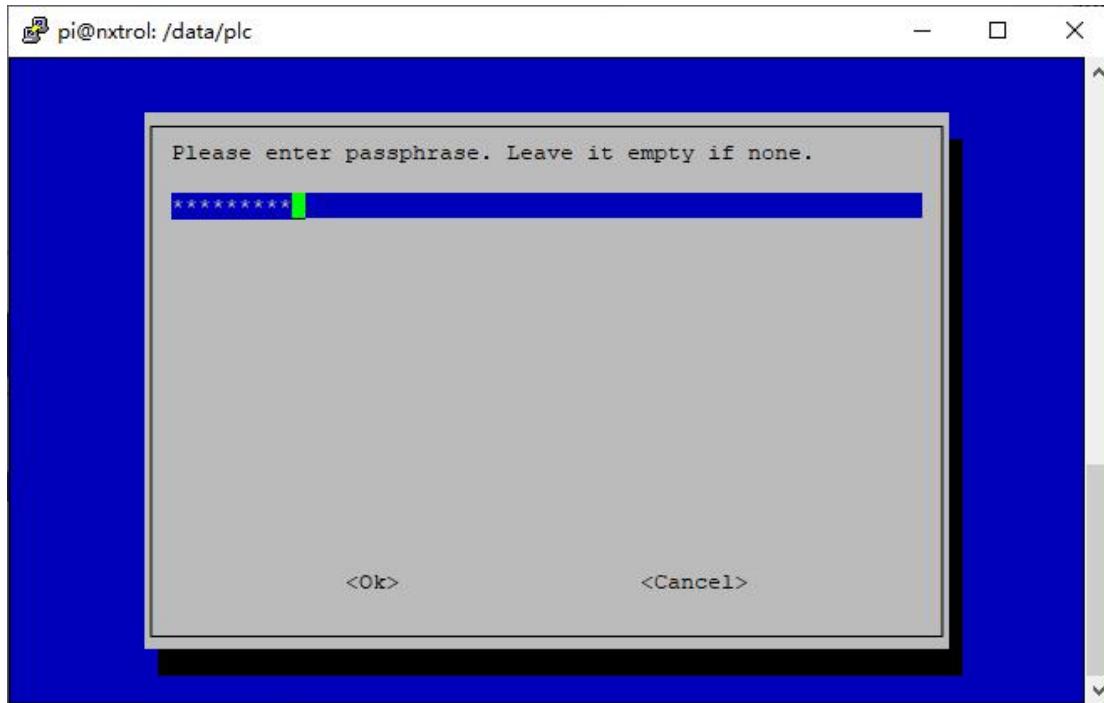
选择 Network Options 后，按回车键可以进入网络设置界面，如下图所示：



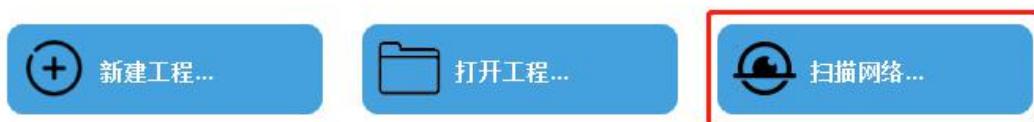
选择 Wi-fi 选项，回车键进入 WIFI 用户名与密码设置，如下图所示：



输入无线路由器的 WIFI 名称 SSID 以及在后续界面中设定 WIFI 密码(请按照当前使用的无线路由器实际值来进行设定)，如下图所示：



设定 WIFI 信息完成后，键盘选择 OK，并在配置主界面中选择 Finish 完成 WIFI 设置，此时建议断开树莓派硬件与无线路由器的网线后，重新上电，并等待约 60s 后，再次使用 LogicLab 扫描网络，此时如一切正常的话，LogicLab 可以再次扫描到树莓派的设备，此时可以确定树莓派的 WIFI 工作正常。



| 扫描结果 |     |               |   |   |   |      |
|------|-----|---------------|---|---|---|------|
| 设备名  | 版本  | IP地址          | 新建工程  | 导入工程  | 应用名称  | 应用版本 |
| 树莓派  | 2.2 | 192.168.5.118 |  |  |  | N/A  |

如发生扫描不到树莓派设备的情况，请参考如下步骤进行排查：

1. 树莓派 4B 的 WIFI 连接到的无线路由器名称 SSID 与密码是否正确。

2. 树莓派 4B 是否与运行 LogicLab Automation Suite 的 PC 同时连接至同一个无线路由器。
3. 插上树莓派的网线至无线路由器上，重复以上步骤。

## 6. 9. 使用欧比特测控 EdgeBox-RPI 如何开启 4G 功能

购买了欧比特测控 EdgeBox-RPI 后，如果选配了 4G 模组，可以通过下面串口终端命令行开启或者关闭 4G 功能，开启 4G 功能之前确保已经插入 SIM 卡（推荐使用电信网络）：

- 开启 4G 功能

```
sudo systemctl enable 4g.service
```

- 关闭 4G 功能

```
sudo systemctl disable 4g.service
```

## 6. 10. 如何自行烧写欧比特测控 EdgeBox-RPI 的树莓派核心板镜像

购买欧比特测控的 EdgeBox-RPI 设备，可以向厂家要求烧写好最新版本的翌控科技 LogicLab 树莓派镜像文件，一般情况下无需自主烧写。如果需要自主烧写镜像，需要购买微雪电子的树莓派 Compute Module 底板，拆开 EdgeBox-RPI 硬件取出核心板，插入在微雪电子的树莓派 Compute Module 底板上，并通过下面教程进行烧写（注意烧写的镜像必须在翌控科技官网上下载，下载地址参考 6.1 章节）：

[http://www.waveshare.net/wiki/Compute\\_Module\\_IO\\_Board\\_Plus](http://www.waveshare.net/wiki/Compute_Module_IO_Board_Plus)



上海翌控科技有限公司  
Shanghai Nxtrol Technology Co., Ltd.

视频烧写教程请参考如下链接：

<https://www.bilibili.com/video/BV1Lp4y1973E>

上海翌控科技版权所有

## 7. 技术支持与服务

如果您在使用树莓派镜像中有任何问题或者建议都可以通过下面地址提交给我们或者邮件

联系：

技术问题提交地址: <https://www.nxtrol.com/service.html>

技术支持邮箱:support@nxtrol.com

上海翌控科技版权所有