

SIMATIC IOT2050 使用手册

目录

1.概述.....	5
1.1 SIMTIC IOT2050 产品介绍.....	5
1.2 SIMATIC IOT2050 规格参数.....	7
1.3 SIMATIC IOT2050 硬件接口说明.....	9
1.4 SIMATIC IOT2050 扩展 IO 模块.....	19
1.4.1 IOT2000 输入/输出模块接线方式.....	20
1.4.2 IOT2000 输入模块接线方式（下拉模式/源模式）	22
1.4.3 IOT2000 输入/输出模块规格参数.....	23
1.4.4 IOT2000 输入模块规格参数（下拉模式/源模式）	24
1.4.5 IOT2000 输入/输出模块硬件接口.....	25
1.4.6 IOT2000 输入模块（下拉模式/源模式） 硬件接口.....	26
1.5 IMG 镜像文件说明.....	27
1.6 辅助工具及软件.....	30
1.7 设置 SIMATIC IOT2050 启动项.....	32
2.镜像文件烧写.....	34
2.1 镜像文件下载方法.....	34
2.2 将镜像文件烧写到 SD 卡或 U 盘.....	35

2.3 启动 SIMATIC IOT2050.....	37
2.4 将镜像文件烧写到 EMMC.....	38
2.4.1 通过 SD 卡、U 盘和标准镜像文件进行烧写.....	39
2.4.2 通过 SD 卡和 Flash 烧写专用应用镜像进行烧写.....	42
2.5 Windows 下格式化烧写有镜像文件的 SD 卡/USB.....	44
3.Example Image 设置方法.....	49
3.1 系统登陆.....	49
3.2 参数设置.....	52
3.2.1 OS Settings 设置.....	53
3.2.2 Networking 设置.....	54
3.2.3 Software 设置.....	56
3.2.4 Peripherals 设置.....	57
4.Node-Red 应用.....	59
4.1 启动 Node-Red 应用.....	59
4.2 Node-Red 应用简介.....	60
4.3 Node-Red 三个常用节点.....	63
4.4 dashboad 节点.....	69
4.4.1 button 节点应用.....	71
4.4.2 gauge 节点应用.....	74
4.4.3 chart 节点应用.....	78

4.5 通过 S7 节点读取 PLC 数据.....	84
4.6 通过 Modbus 节点读设备数据.....	89
4.7 制作 UI 界面.....	93
4.8 通过 MQTT 节点上传数据.....	97
5.Edgeconnect Image 设置及应用.....	103
5.1 启动 Edgeconnect 应用.....	104
5.2 系统配置模块.....	106
5.2.1 接口配置.....	106
5.2.2 网络转换.....	109
5.2.3 系统配置.....	111
5.3 数据采集模块.....	112
5.3.1 IOSHIELD 通道.....	113
5.3.2 Modbus_RTU 通道.....	118
5.3.3 Modbus_TCP 通道.....	125
5.3.4 OPCUA 通道.....	130
5.3.5 Siemens S7 通道.....	135
5.3.6 Mitsubishi_MC_QNA3E 通道.....	142
5.3.7 Mitsubishi_MC_A1E 通道.....	148
5.3.8 Omron_FINS_TCP 通道.....	154
5.3.9 AB_Ethernet/IP 通道.....	160
5.4 云端上传模块.....	165

5.4.1 MindSphere 通道.....	166
5.4.2 AliCloud 通道.....	171
5.4.3 OneNET 通道.....	176
5.4.4 OPCUA Server 通道.....	181
5.4.5 General MQTT 通道.....	185
5.4.6 Modbus TCP Server 通道.....	193
5.5 脚本编程（边缘计算）模块.....	197
5.6 网络管理模块.....	205
5.7 证书管理.....	207
5.8 导入/导出.....	208
5.9 SINEMA RC（远程运维/调试）模块.....	209
5.9.1 上传 SINEMA RC 配置文件.....	210
5.9.2 设置 SINEMA RC 设备密码及验证方式.....	211
6. 其他应用方式.....	212
6.1 扩展 4G, 5G 应用.....	213
6.2 扩展 WIFI 应用.....	215
6.3 安装桌面及浏览器.....	216
6.4 安装 MariaDB 数据库.....	217

1. 概述

1.1 SIMATIC IOT2050 产品介绍



SIMATIC IOT2050 物联网智能网关，工业物联网应用解决方案，满足自动化层与 IT 层间日益增加的数据互联和通讯需求。

- 专为工业环境中 7*24 小时使用需求设计，高鲁棒性、高可靠性，通过了 CE, UL 等多项国际测试认证。
- 采用基于 ARM 架构的四核处理器芯片，主频高达 1.1GHZ，算力强大。内部集成 2G 内存和 16GEMMC 存储，可满足大数据分析，处理，边缘计算的性能需求，支持虚拟容器环境。
- 多种现场设备的常用接口，可满足绝大多数 OT 层现场设备通讯需求。
- 集成 Adriuno 和 miniPCIe 扩展接口，可直接采集现场传感器，IO 设备的模拟量，数字量信息，并可扩展 4G,5G,Wifi,Lora 等无线通讯功能。

1.2 SIMATIC IOT2050 规格参数

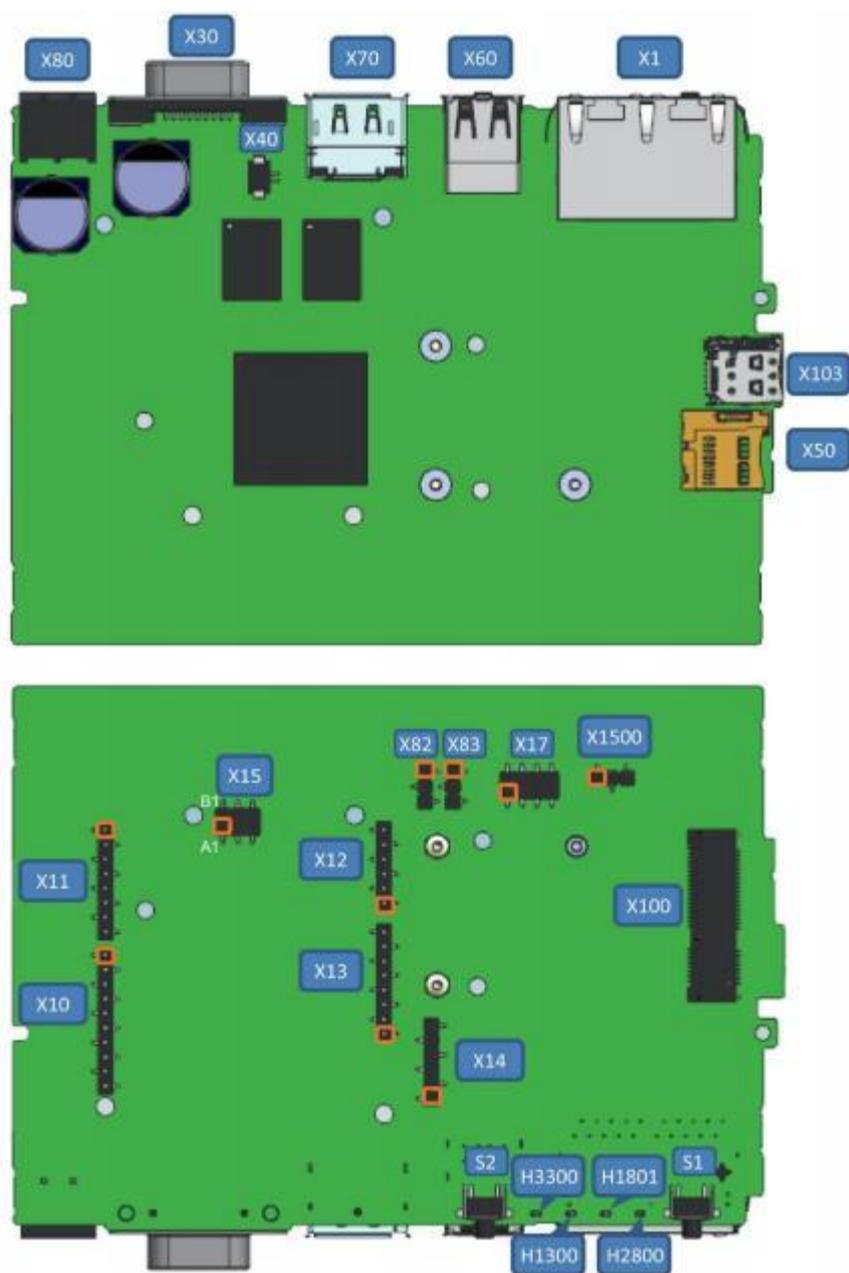
产品名称		IOT2050 Basic	IOT2050 Advance
型号 (订货号)		6ES7647-0BA00-0YA2	6ES7647-0BA00-1YA2
硬件配置	CPU	TI Sitara AM6528 GP, 双核, 1.1GHz	TI Sitara AM6548 HS, 四核, 1.1GHz
	RAM	1 GB DDR4	2 GB DDR4
	BIOS SPI 存储	8M	16M
	EMMC	无	16G
	系统	Debian/Industry OS	Debian/Industry OS
	以太网口	2x 100/1000 以太网 RJ45 接口	2x 100/1000 以太网 RJ45 接口
	USB 口	1x USB 3.0 (蓝) 1x USB 2.0 (黑)	1x USB 3.0 (蓝) 1x USB 2.0 (黑)
	显示接口	1xDP	1xDP
	COM 口	1x RS232/422/485 (可设定)	1x RS232/422/485 (可设定)
	SIM 卡插槽	Nano SIM	Nano SIM
	SD 卡插槽	Micro SD	Micro SD
	IO 扩展板	Arduino Uno R3	Arduino Uno R3
	扩展插槽	1x miniPCIe 全尺寸	1x miniPCIe 全尺寸
	按钮	1x 复位 + 1x 用户定义	1x 复位 + 1x 用户定义
	实时时钟	无	带电池缓冲的实时时钟
技术规范	重量	约 330g (不含支架)	约 340g (不含支架)
	外壳	树脂&金属散热片	
	系统	Debian/Industry OS	
	电源	DC12-24V 非电隔离	
	电流消耗	最大 1.7A	
	功耗	最大 24W	
	噪声传播	<40dB(A), 符合 DIN45635-1 标准	
	防护等级	IP20(符合 IEC 60529 规定)	
	防护类别	防护类别III (符合 IEC61140 规定)	
	污染等级	设备设计为用于污染等级为 2 的环境	
	质量保证	符合 ISO 9001	
认证和证书	CE, ISO9001, UL, FCC 和加拿大, RCM, EAC, IC, WEEE		
电磁兼容性	电源线上的传导干扰抗扰性	± 2 kV, 符合 IEC 61000-4-4 要求; 电快速瞬变脉冲群 ± 1 kV, 符合 IEC 61000-4-5 要求; 非对称浪涌	
	信号线抗扰度	± 1 kV, 符合 IEC 61000-4-4; 电快速瞬变脉冲群; 长度 < 30 m ± 2 kV, 符合 IEC 61000-4-4; 电快速瞬变脉冲群; 长度 > 30 m ± 2 kV, 符合 IEC 61000-4-5; 浪涌冲击; 长度 > 30 m	
	抗静电放电性	± 4 kV 接触放电, 符合 IEC 61000-4-2 ± 8 kV 空气放电, 符合 IEC 61000-4-2	
	抗 RF 干扰性	10 V/m, 80 MHz 到 1 GHz, 80% AM, 符合 IEC 61000-4-3 3 V/m, 1.4 到 6 GHz, 80% AM, 符合 IEC 61000-4-3 10 V, 150 KHz 到 80 MHz, 80% AM, 符合 IEC 61000-4-6	

气候环境条件	环境温度	运行	导轨安装：垂直：0 °C 至 50 °C ； 水平：0 °C 至 40 °C 墙面安装：垂直：0 °C 至 45 °C ； 水平：0 °C 至 40 °C
		存储/运输	-20 °C 至 70 °C
	梯度	运行	最大 10 °C/h
		存储	20 °C/h, 无冷凝
	相对湿度	运行	30°C 时为 5% 至 85%，无冷凝
		存储/运输	25/55 °C 时为 5% 至 95%，无冷凝
	大气压力 海拔高度	运行	1080 至 795 hPa, 相当于海拔高度 -1000 m 至 2000 m
		存储/运输	1080 至 660 hPa, 相当于海拔高度 -1000 m 至 3500 m
机械环境条件	抗振性	运行	振动载荷 1g, 每轴 10 个周期： • 5 至 8.4 Hz, 振幅为 3.5 mm • 8.4 至 200 Hz, 加速度为 9.8 m/s ²
		存储/运输	• 5 至 8.4 Hz : 振幅为 3.5 mm • 8.4 Hz 至 500 Hz : 加速度为 9.8 m/s ²
	抗冲击性	运行	150 m/s ² , 11 ms
		存储/运输	250 m/s ² , 6 ms

1.3 SIMATIC IOT2050 硬件接口说明

主板

下图显示了 SIMATIC IOT2050 的主板。



组件/接口	描述/标记		含义, 注释
H2800	LED	PWR	电源(绿色)
H1801		STAT	操作系统运行中(绿灯闪烁)。
H1300		USER1	用户 LED (绿/红/橙), 可编程
H3300		USER2	用户 LED (绿/红/橙), 可编程
S1	RESET		用于复位 CPU
S2	USER		可通过编程查询状态
X1	以太网 1 和以太网 2		
X30	COM D-Sub9		
X60	USB A 主机		
X70	DisplayPort		
X14	UART 调试		
X40	Battery-Con		
X50	μSD 插槽		
X11、X13	Arduino, 8 针脚 (2x)		
X10	Arduino, 10 针脚		
X12	Arduino, 6 针脚		
X15	ICSP		Arduino 接口的一部分
X17	DediProg 连接器		
X80	电源		
X83	跳线, 3 针脚, SMD		跳线定义 Arduino 扩展板的 IO 电压 : <ul style="list-style-type: none"> • 跳线位于针脚 1-2 上 : 5 V • 跳线位于针脚 2-3 上 : 3.3 V
X1500	跳线, 3 针脚, SMD		跳线定义闪存写保护的启用或禁止 : <ul style="list-style-type: none"> • 跳线位于针脚 1-2 上 : 启用写保护 • 跳线位于针脚 2-3 上 : 禁止写保护 注 : 正常运行期间, 闪存写保护条件应处于写保护启用状态(针脚 1-2)。

外部接口

组件/接口	描述/标记	含义, 注释
X100	MiniPCle 连接器	
X103	Nano SIM 连接器	
X82	VIN 隔离(电源, 24 V DC)	跳线指定 VIN 与 Arduino 扩展板之间的连接 : <ul style="list-style-type: none"> 跳线位于引脚 1-2 上 : VIN 连接到 Arduino 扩展板 跳线位于引脚 2-3 上 : VIN 不连接到 Arduino 扩展板

电源

插头连接器, 2 针

设备上的接口名称 : X80

引脚	分配
1	GND (M)
2	+12 到 24 V DC (L+)

USB

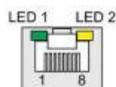
A 型 USB 插槽

设备上的接口名称 : USB1 - X60 - USB2

-引脚	分配
1	+5 VDC, 输出(最大 500 mA)
2	USB-DN
3	USB-DP
4	GND

以太网接口

RJ45 插座



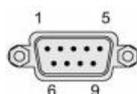
设备上的接口名称：P1 - X1 LAN - P2

引脚	简称	含义
1	BI_DA+	双向数据 A+, 输入/输出
2	BI_DA-	双向数据 A-, 输入/输出
3	BI_DB+	双向数据 B+, 输入/输出
4	BI_DC+	双向数据 C+, 输入/输出
5	BI_DC-	双向数据 C-, 输入/输出
6	BI_DB-	双向数据 B-, 输入/输出
7	BI_DD+	双向数据 D+, 输入/输出
8	BI_DD-	双向数据 D-, 输入/输出

LED

LED	含义	状态	状态含义
LED 1	连接状态	熄灭	无数据传输
		绿灯闪烁	数据传输进行中
LED 2	数据传输速率	熄灭	预留
		绿色	100 Mbps
		黄色	1000 Mbps

串行端口 (RS232/RS485/RS422)



D-sub 插座, 9 针, 带锁紧螺钉

设备上的接口名称: X30 COM

RS 232 分配

引脚	简称	含义
1	DCD	数据载体检测 (I)
2	RxD	接收数据 (I)
3	TxD	发送数据 (O)
4	DTR	数据终端就绪 (O)
5	M	接地
6	DSR	数据准备就绪 (I)
7	RTS	请求发送 (O)
8	CTS	允许发送 (I)
9	RI	振铃 (I)

RS 422 分配

引脚	简称	含义
1	TX-	全双工模式的发送数据 - (O)
2	TX+	全双工模式的发送数据 + (O)
3	RX+	全双工模式的接收数据 + (I)
4	RX-	全双工模式的接收数据 - (I)
5	M	信号接地端
6	nc	
7	nc	
8	nc	
9	nc	

RS 485 分配

引脚	简称	含义
1	数据-	发送/接收数据- (I/O), 用于半双工模式
2	数据+	发送/接收数据+ (I/O), 用于半双工模式
3	nc	
4	nc	
5	M	信号接地端
6	nc	
7	nc	
8	nc	
9	nc	

内部接口

Arduino 扩展板接口

下表根据工作模式列出了 Arduino 扩展板接口的引脚分配。

X10

引脚	操作模式						
	DIGITAL	ANALOG	POWER	SERIAL	SPI	I2C	PWM
1	8						X
2	9						X
3	10				SS		
4	11				MOSI		
5	12				MISO		
6	13				SCK		
7			GND				
8							
9						SDA	
10						SCL	

X11

引脚	操作模式						
	DIGITAL	ANALOG	POWER	SERIAL	SPI	I2C	PWM
1	0			UART Rx			
2	1			UART Tx			
3	2			UART CTS			
4	3			UART RTS			
5	4						X
6	5						X
7	6						X
8	7						X

X12

引脚	操作模式						
	DIGITAL	ANALOG	POWER	SERIAL	SPI	I2C	PWM
1	14	A0					
2	15	A1					
3	16	A2					
4	17	A3					
5	18	A4				SDA	
6	19	A5				SCL	

X13

引脚	操作模式						
	DIGITAL	ANALOG	POWER	SERIAL	SPI	I2C	PWM
1							
2			IOREF				
3			RESET				
4			3.3 V				
5			5 V				
6			GND				
7			GND				
8			VIN				

X15(ICSP)

引脚	操作模式						
	DIGITAL	ANALOG	POWER	SERIAL	SPI	I2C	PWM
A1	12				MISO		
B1			5 V				
A2	13				SCK		
B2	11				MOSI		
A3			RESET				
B3			GND				

UART 调试

X14

Pin	占用
1	GND
2	RTS_N
3	n. c.
4	RxD
5	TxD
6	CTS_N

Mini PCIe接口

X100

引脚	信号名称	引脚	信号名称
51	W_DISABLE2#	52	+3.3 V _{aux}
49	保留	50	GND
47	保留	48	+1.5 V
45	保留	46	LED_WPAN#
43	GND	44	LED_WLAN#
41	+3.3V _{aux}	42	LED_WWAN#
39	+3.3V _{aux}	40	GND
37	GND	38	USB_D+
35	GND	36	USB_D-
33	PETp0	34	GND
31	PETn0	32	SMB_DATA
29	GND	30	SMB_CLK
27	GND	28	+1.5V
25	PERp0	26	GND
23	PERn0	24	+3.3V _{aux}
21	GND	22	PERST#
19	保留 (UIM_C4)	20	W_DISABLE1#
17	保留 (UIM_C8)	18	GND
Nose		Nose	
15	GND	16	UIM_VPP
13	REFCLK+	14	UIM_RESET
11	REFCLK-	12	UIM_CLK
9	GND	10	UIM_DATA
7	CLKREQ#	8	UIM_PWR
5	COEX2	6	1.5 V
3	COEX1	4	GND
1	WAKE#	2	3.3 V _{aux}

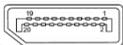
Micro SD 接口

X50

引脚	缩略语	功能
1	Dat2	数据线位 2
2	Dat3	卡检测/数据线位 3
3	CMD	命令线路
4	Vdd	电源电压 2.7–3.6 V
5	Clk	时钟输入
6	GND	信号接地端
7	Dat0	数据线位 0
8	Dat1	数据线位 1

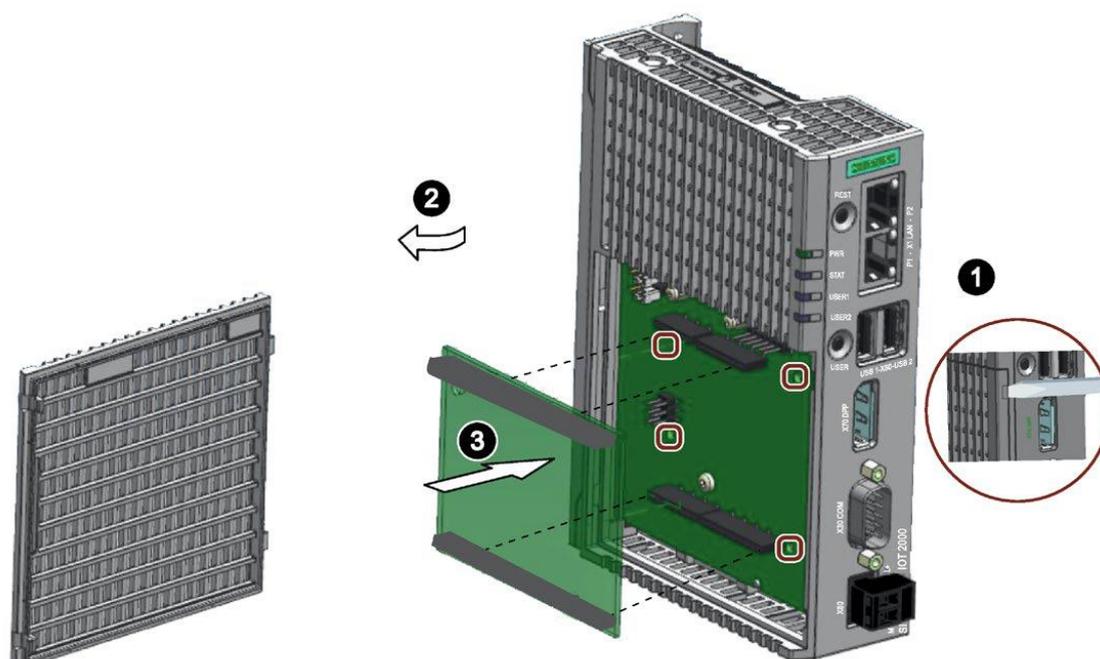
DisplayPort1.1A

X70

DisplayPort 接口 			
引脚编号	缩略语	功能	输入/输出
1	Lane 0_P	通道 0+	输入
2	GND	接地	-
3	Lane 0_N	通道 0-	输入
4	Lane 1_P	通道 1+	输入
5	GND	接地	-
6	Lane 1_N	通道 1-	输入
7	Lane 2_P	通道 2+	输入
8	GND	接地	-
9	Lane 2_N	通道 2-	输入
10	Lane 3_P	通道 3+	输入
11	GND	接地	-
12	Lane 3_N	通道 3-	输入
13	CONFIG1	接地	-
14	CONFIG2	接地	-
15	AUX CH_P	辅助通道+	双向
16	GND	接地	-
17	AUX CH_N	辅助通道-	双向
18	Hot plug	热插拔检测	输出
19	Return	电源返回端接地	-
20	DP_PWR	连接器电源 (3.3 V 500 mA)	电源输出

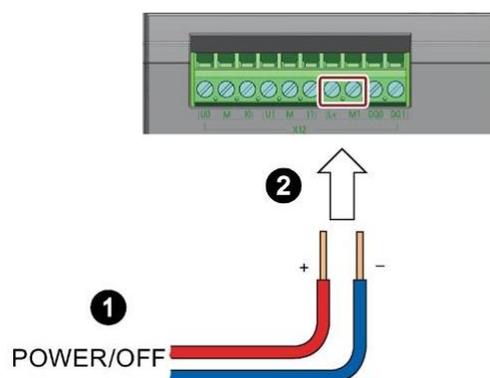
1.4 SIMATIC IOT2050 扩展 IO 模块

SIMATIC IOT2050 上集成了 Arduino Shield 接口，如图所示，可以通过扩展 IO 模块，直接采集设备的数字量和模拟量数据。



扩展 IO 模块可以使用第三方的标准 Arduino 接口模块，也可以使用西门子提供的两种不同规格的扩展 IO 模块。分别为 IOT2000 输入/输出模块和 IOT2000 输入模块

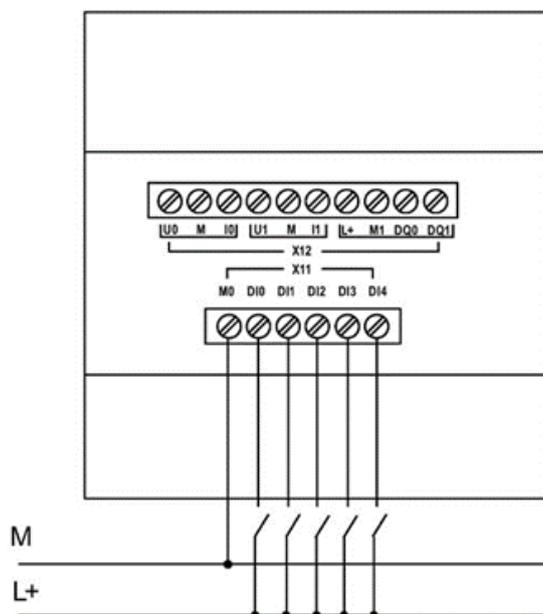
在使用数字输出模块时，需要连接外接电源，电源电缆横截面为 0.75 mm² 至 2.5 mm² 的双芯电缆。连接方式如下图所示。



1.4.1 IOT2000 输入/输出模块接线方式

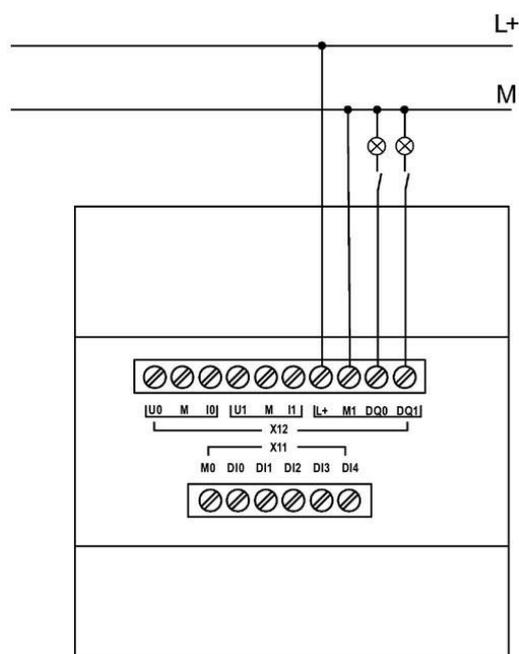
连接数字量输入

下图显示了如何连接 IOT2000 输入/输出模块的数字输入



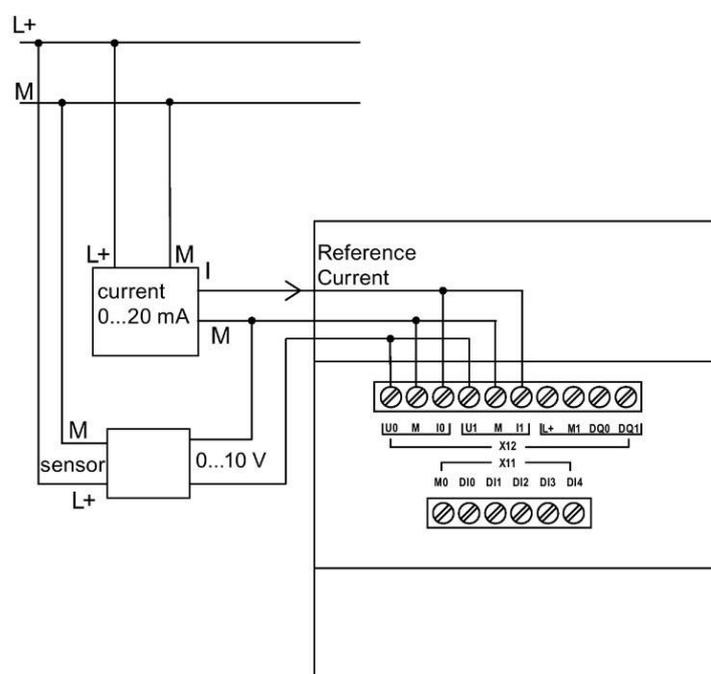
连接数字量输出

下图显示了如何连接 IOT2000 输入/输出模块的数字输出



连接模拟量输入

下图显示了如何连接 IOT2000 输入/输出模块的模拟输入



注意：请务必使用屏蔽电缆连接模拟输入。

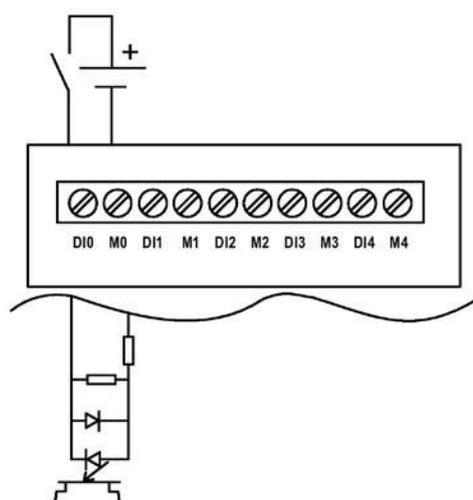
1.4.2 IOT2000 输入模块接线方式（下拉模式/源模式）

注意：操作前请切断 IOT2050 的电源

连接数字量输入

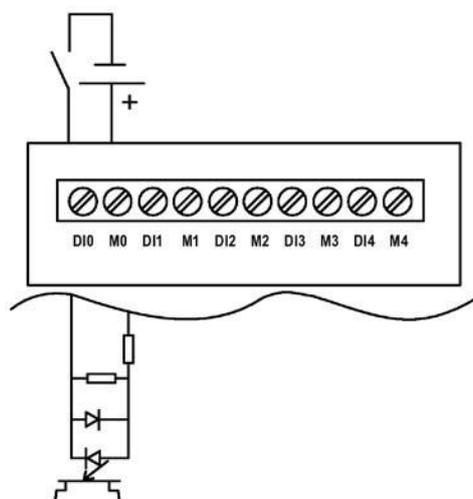
下图显示了如何连接 IOT2000 输入模块的数字输入

24 VDC 下拉模式



下图显示了如何连接 IOT2000 输入模块的数字输入

24 VDC 源模式



1.4.3 IOT2000 输入/输出模块规格参数

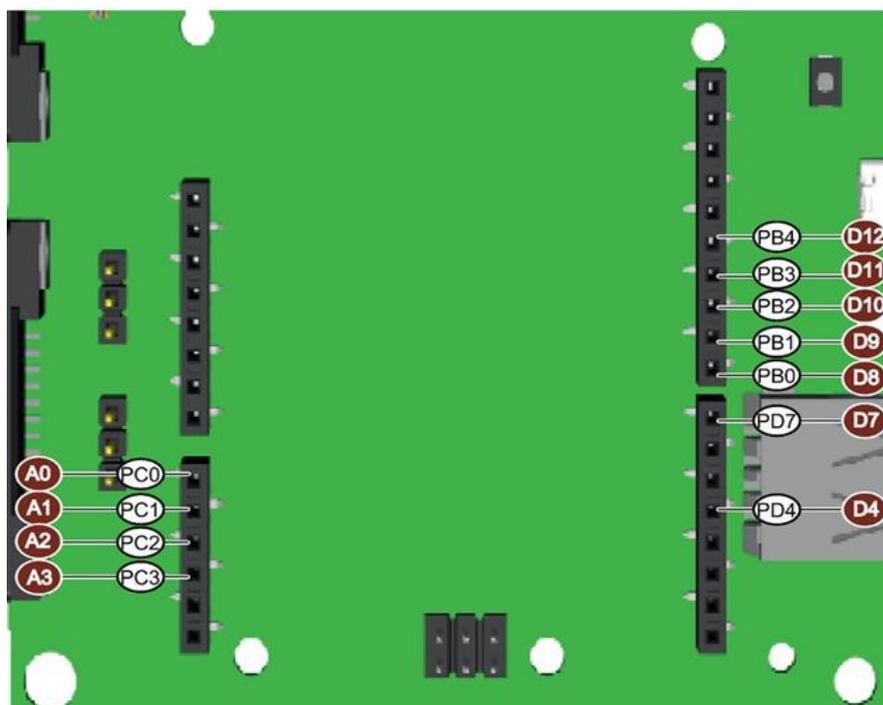
IOT2000 输入/输出模块	
订货号	6ES7647-0KA01-0AA2
数字量输入	
数量 (标准数据)	5
电气隔离	是
数量 (高速输入)	0
标准输入频率	最高 20 Hz
最大连续允许电压	28.8 VDC
Signal 0 输入电压	< 5 VDC
0 to 1 延时	Typ. 1.5 ms
接线长度(非屏蔽)	最长 100 m
模拟量输入	
数量	2 路电流 & 2 路电压
类型	单极
输入范围	0 VDC to 10 VDC (输入阻抗 38 k Ω) 0 mA to 20 mA (输入阻抗 <380 Ω)
分辨率	9 bit
接线长度(屏蔽和twisted)	最长 10 m
电气隔离	无
容差	$\pm 3\%$ 1)
数字量输出	
数量	2
输出类型	晶体管输出, 继电器输出
电气隔离	是
输出电源电压允许范围(L+)	20.4VDC to 28.8 VDC
输出电压	\leq 供电电压
输出电流	最大 0.3 A
短路和过载保护	是
短路电流限制	约 1 A /通道
并联输出电路	不允许
转换速率	
机械	--
电气	10 Hz
电阻负载/灯负载	10 Hz
电感负载	0.5 Hz

1.4.4 IOT2000 输入模块规格参数（下拉模式/源模式）

IOT2000 输入模块（下拉模式/源模式）	
订货号	6ES7647-0KA02-0AA2
数字量输入	
数量	10
电气隔离	Yes
告诉输入模块	0
标准输入频率	Max. 20 Hz
最大连续允许电压	28.8 VDC
输入电压	
信号 0 信号 1	< 5 VDC > 12 VDC
输入电流	
信号 0 信号 1	< 1.0 mA > 2.1 mA
延迟时间	
0 to 1 1 to 0	<ul style="list-style-type: none"> • Typ. 1.5 ms • Typ. 15 ms
线长(非屏蔽)	Max. 100 m

1.4.5 IOT2000 输入/输出模块硬件接口

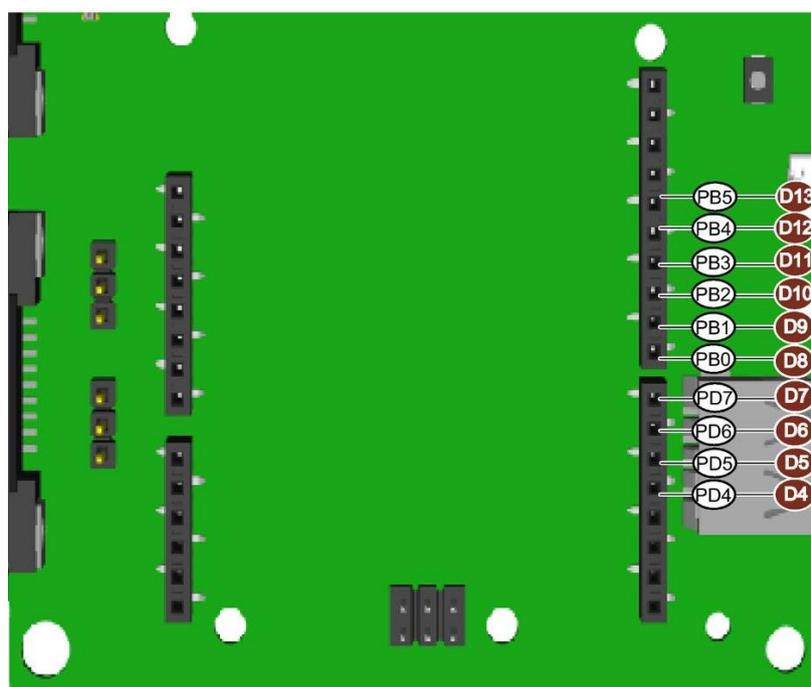
用户可以通过用户应用端口控制 I/O 状态，如下所示。



描述	功能	用户应用端口	内部连接
DI0	数字量输入	D12	PB4
DI1		D11	PB3
DI2		D10	PB2
DI3		D9	PB1
DI4		D4	PD4
DQ0	数字量输出	D8	PB0
DQ1		D7	PD7
U0	模拟量输入	A0	PC0
I0		A1	PC1
U1		A2	PC2
I1		A3	PC3
M	M		
P5V	P5V		
P24V	P24V		

1.4.6 IOT2000 输入模块（下拉模式/源模式）硬件接口

用户可以通过用户应用端口控制 I/O 状态，如下所示。



描述	功能	用户应用端口	内部连接
DI0	数字量输入	D4	PD4
DI1		D5	PD5
DI2		D6	PD6
DI3		D7	PD7
DI4		D8	PB0
DI5		D9	PB1
DI6		D10	PB2
DI7		D11	PB3
DI8		D12	PB4
DI9		D13	PB5
M0, M1, M2, M3, M4, M5, M6, M7, M8, M9	各通道的Mx引脚		

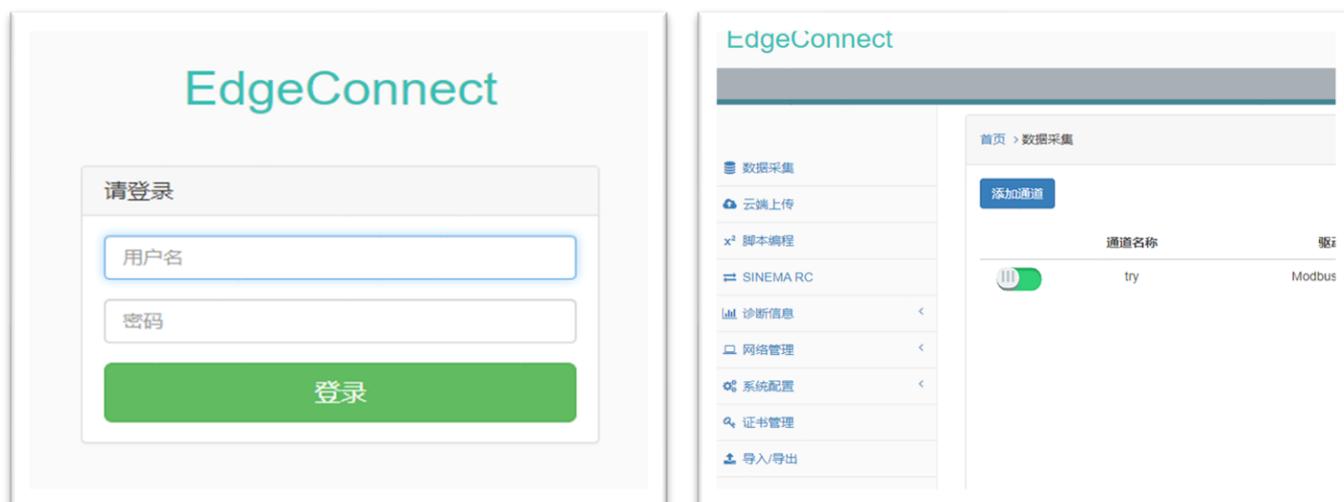
1.5 IMG 镜像文件说明

SIMATIC IOT2050 中运行的系统及应用程序是通过后缀名为的镜像文件烧写到存储设备当中的。用户拿到产品后，可以根据自身需要来选择安装什么系统和应用，需要先将所需的镜像文件烧写到对应的存储设备后（SD 卡，U 盘，EMMC），系统才能启动并运行相关应用。

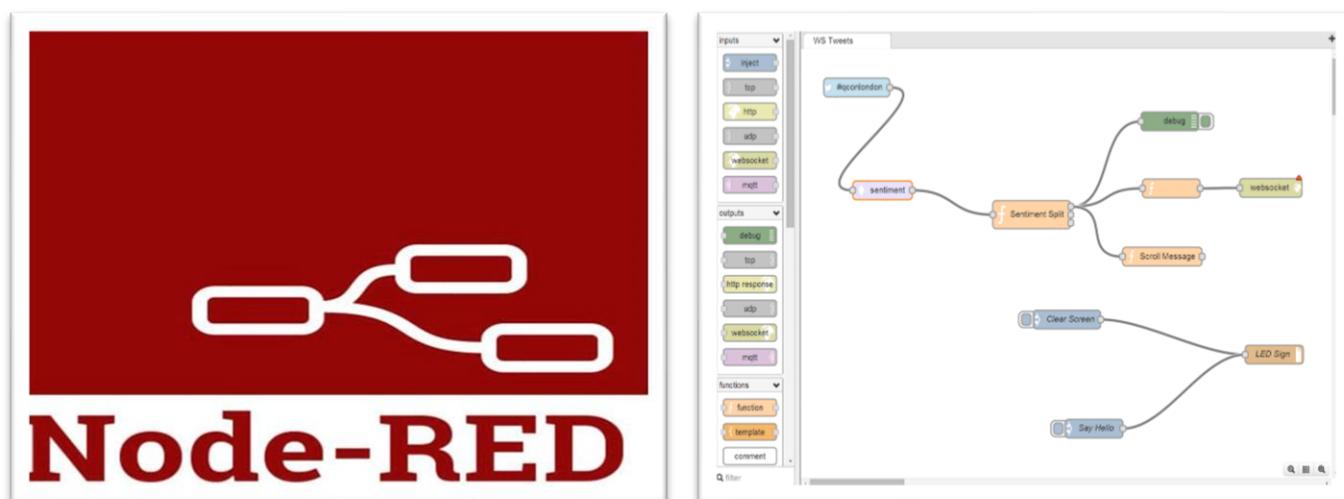
镜像文件是一种后缀名为的压缩文件，可以用来压缩整个软盘或整片光盘的内容。烧写的过程与使用光驱向硬盘中安装打包好的操作系统的过程类似。将烧写好镜像文件的SD卡或优盘插入IOT2050，或者将镜像文件烧写到IOT2050的EMMC后，启动IOT2050就可以进入系统并运行相应的应用程序。

SIMATIC IOT2050 可以提供两种镜像文件，一种是集成了 Debian linux 系统，Edgeconnect应用和Node-red应用的镜像文件，该镜像的Debian linux 系统并没有向用户开放 root 权限，是面向只有应用需求的使用者的。

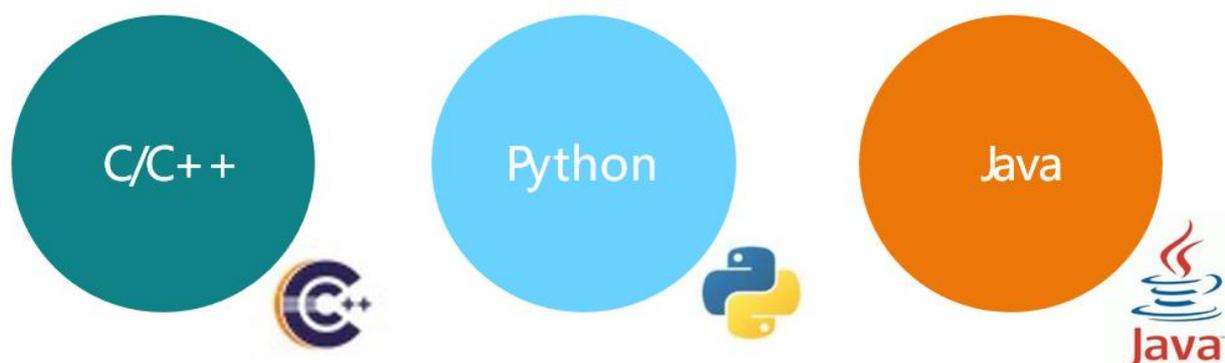
其中 Edgeconnect 应用采用图形用户界面，web 端口访问的形式登陆，内部已经集成了数据采集，云端上传，VPN 连接等多个功能模块，用户登陆后进行简单的参数设置即可轻松实现物联网网关的各种功能。



Node-red 是一个基于浏览器的低代码编程工具，用户可以轻松地使用应用中各种节点，以拖拽的方式将数据流连接在一起，只需进行简单的设置和部署，就可以轻松实现数据采集，上传等工作，并可使用相应节点制作图中所示的 UI 界面。

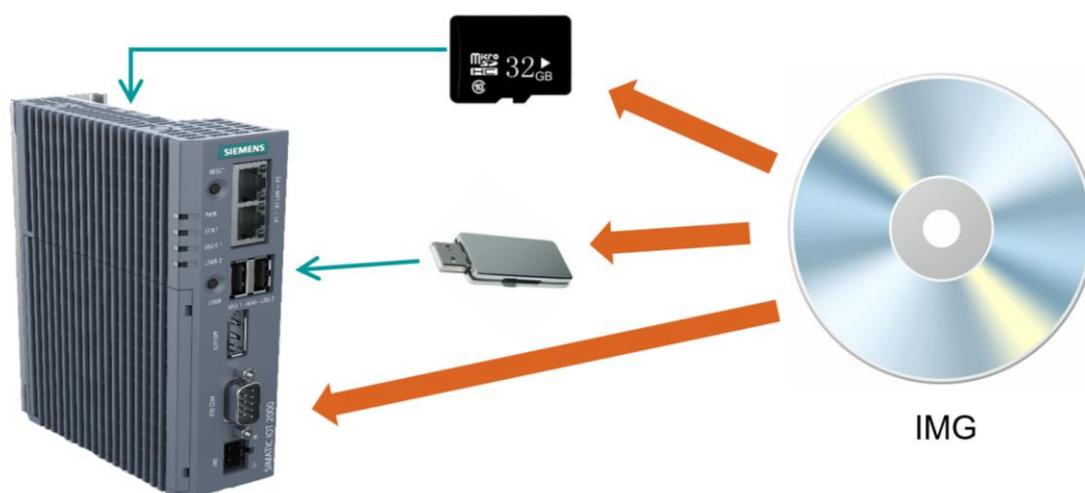


另一种是只集成了 Debian Linux 系统和 Node-red 应用的镜像文件，该镜像中的 Debian linux 系统向用户开放了 root 权限，是面向有开发需求使用者的。用户可以使用 root 管理员的身份登入系统，对系统进行各种操作。可以使用多种高级语言来开发个性化的满足自身要求的各种应用和界面。在 Github 网站和西门子的 SIMATIC IOT2050 论坛上，也提供许多开发工具和 SDK 可以帮助开发者实现快速开发。



用户根据自身的使用需要，选择将其中的一种镜像烧写到对应的存储设备中，启动后就可以在 IOT2050 平台上进行应用或开发工作。另外，伴随着软件与系统的不断升级，IOT 所提供的镜像文件也会不断地进行升级，用户可以获取最新的镜像文件应用于 IOT2050 产品中。

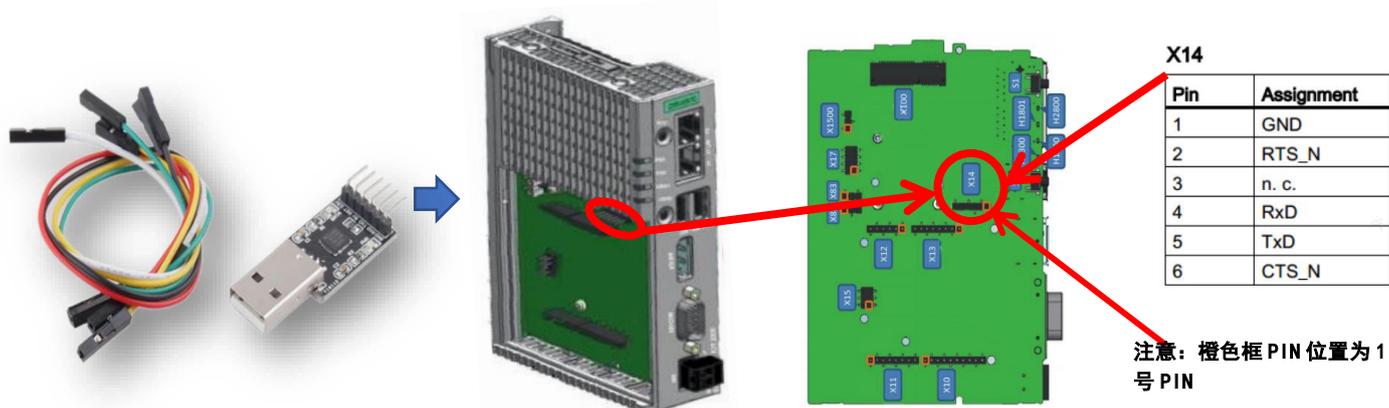
镜像文件可通过访问网址 <https://support.industry.siemens.com/cs/document/109741799> 下载或通过西门子销售人员获取



1.6 辅助工具及软件

辅助工具

TTL 转 USB 转换器：用于通过电脑通过连接 IOT2050 的 X14 UART 调试接口。对 IOT 进行调试，更改 IOT 启动项等相关设置时会使用。



主动式 DP 转接线：用于使用 DP 接口连接 IOT2050 与显示设备。可用于显示 IOT2050 中运行程序的用户界面，进行设置和调试。



辅助软件



Putty 软件：Telnet、SSH、rlogin、纯 TCP 以及串行接口连接软件。使用 Putty 软件可以远程登录 IOT2050，对 IOT2050 进行控制和设置。

软件下载地址：<https://www.putty.be/latest.html/>



Win32DiskImager 软件：Win32 磁盘映像工具，磁盘烧写工具，用于将 img 文件写到 SD 或 U 盘中。使用 Win32DiskImager 软件可以将系统 img 镜像烧写到对应的 SD 卡或 U 盘中，将烧写好对应 img 镜像文件的 SD 卡或 U 盘插入到 IOT2050 中，就可以运行 img 文件中的系统。

软件下载地址：<https://sourceforge.net/projects/win32diskimager/>



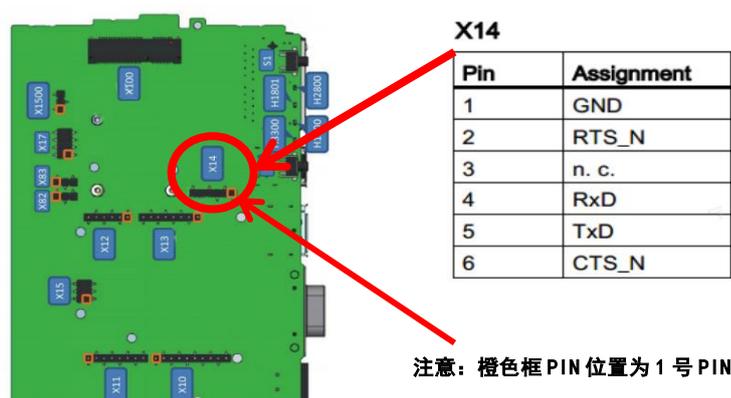
WinSCP 软件：WinSCP 是一个使用 SSH 协议的文件浏览器。主要是用来管理 UNIX(或 Linux/BSD)主机。使用 WinSCP 软件可以用来远程访问 IOT2050，用 FTP、TFTP 等协议来传输文件。

软件下载地址：<https://winscp.net/eng/download.php/>

1.7 设置 SIMATIC IOT2050 启动项

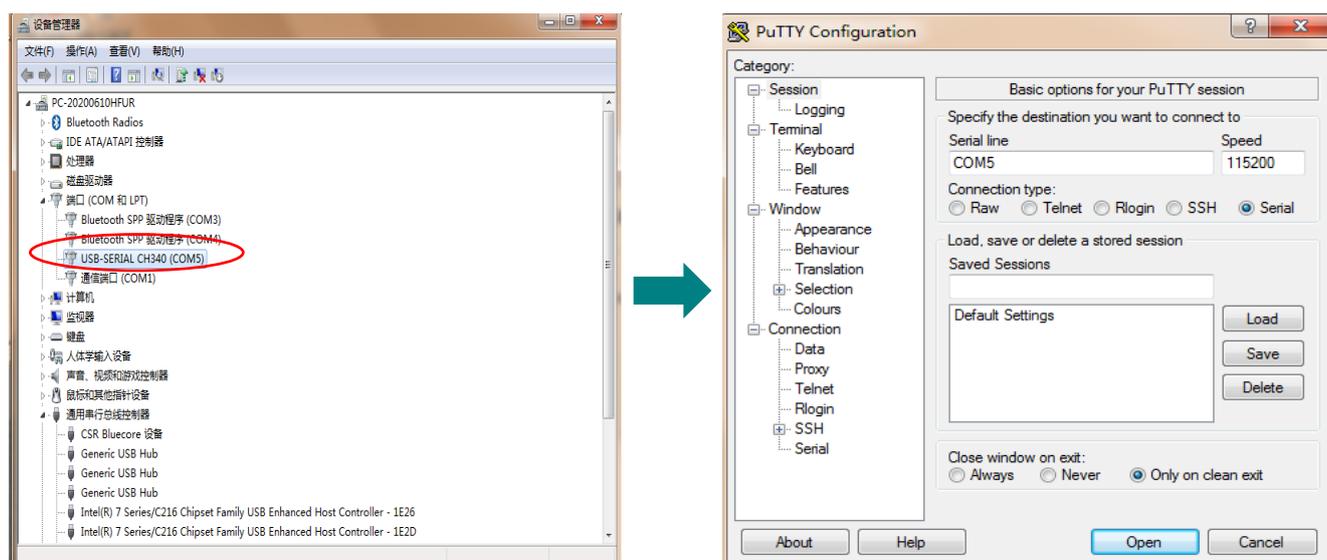
IOT2050 在出厂时，会设置默认的系统启动项，如果不想使用默认的系统启动项，用户可以根据需要更改设备的启动项设置。用户需要使用章节 1.6 中的 TTL 转 USB 转换器和 Putty 软件进行设置。

1) 将 TTL 转 USB 转换器连接至 IOT2050 的 X14 接口以及电脑的 USB 接口。

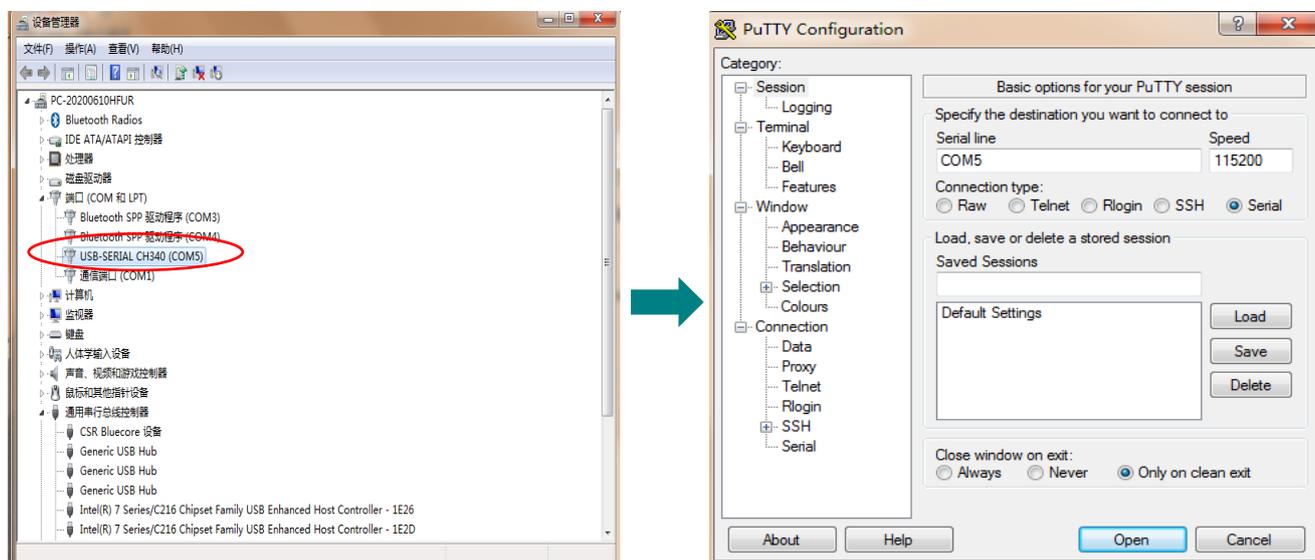


连接 GND, RX, TX 即可，
注意转接线的 TX 连接 X14
的 RX，转接线的 RX 连接
X14 的 TX。

2) 打开 Putty 软件，对连接进行设置：通过电脑的设备管理器中查看端口信息，确认端口号；连接类型为 Serial；填入串口名称，波特率选择 115200。点击打开后即可进入终端界面。



3) 将 IOT2050 电源断开后重新启动，终端界面上会打印启动信息，当显示信息中提示“Hit any key to stop autoboot”时，在两秒内按下 Enter 键打断启动过程，进行启动项设置。



4) 在程序窗口中分别输入下列指令后回车：

```
COM5 - PuTTY
U-Boot 2019.01-V01.00.00.2-0-g8e86139 (Mar 18 2020 - 20:04:56 +0000)

Model: Siemens IOT2050 Advanced Base Board
DRAM: 2 GiB
MMC: sdhci@4f80000: 1, sdhci@04FA0000: 0
Loading Environment from SPI Flash... SF: Detected w25q128 with page size 256 Bytes, erase size 64 KiB, total 16 MiB
OK
In: serial
Out: serial
Err: serial
Net: eth0: pruss0_eth
Hit any key to stop autoboot: 0
=> setenv boot_targets mmc0
=> env save
Saving Environment to SPI Flash... SF: Detected w25q128 with page size 256 Bytes, erase size 64 KiB, total 16 MiB
Erasing SPI flash...Writing to SPI flash...done
Valid environment: 1
OK
=> run distro bootcmd
switch to partitions #0, OK
mmc0 is current device
```

```
setenv boot_targets mmc0/ mmc1/usbx
```

```
env save
```

```
run distro_bootcmd
```

第一行指令中 mmc0 代表 SD 卡，此处也可以用 mmc1 代表 EMMC，usbx 代表 USB 设备，可以根据需要输入选择那个设备作启动项。也可以并行输入多种设备，IOT2050 会按照从左向右的顺序设置启动项的优先级。

2. 镜像文件烧写

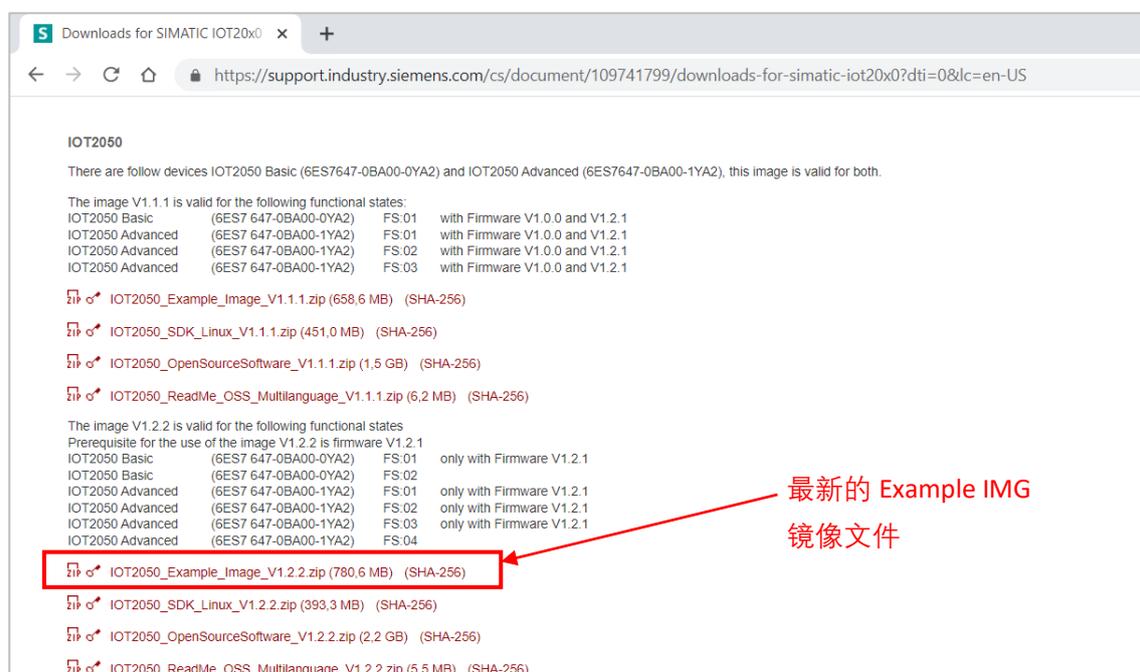
2.1 镜像文件下载方法

在章节【1.5 IMG 镜像文件说明】中提到过，为方便用户使用，西门子可以为用户提供两种镜像文件：一种是面向有开发需求的使用者；只集成了 Debian Linux 系统和 Node-red 应用、开放用户 root 权限的标准镜像文件（Example IMG），另一种是面向有应用需求的使用者；集成了 Debian linux 系统，Edgeconnect 应用和 Node-red 应用、但不开放用户 root 权限的 Edgeconnect 镜像文件。

Example IMG 镜像文件可以通过访问西门子技术支持中心网站：

<https://support.industry.siemens.com/cs/document/109741799> 进行下载。下载

前，需要先进性用户注册和登陆。



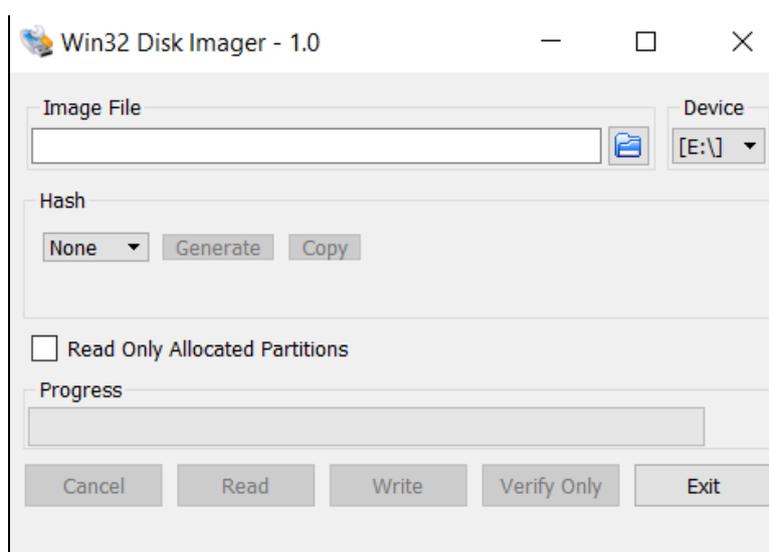
Edgeconnect 镜像文件无法通过访问西门子技术支持中心网站进行下载，需要用户联系对应西门子销售人员进行获取。

2.2 将镜像文件烧写到 SD 卡或 U 盘中

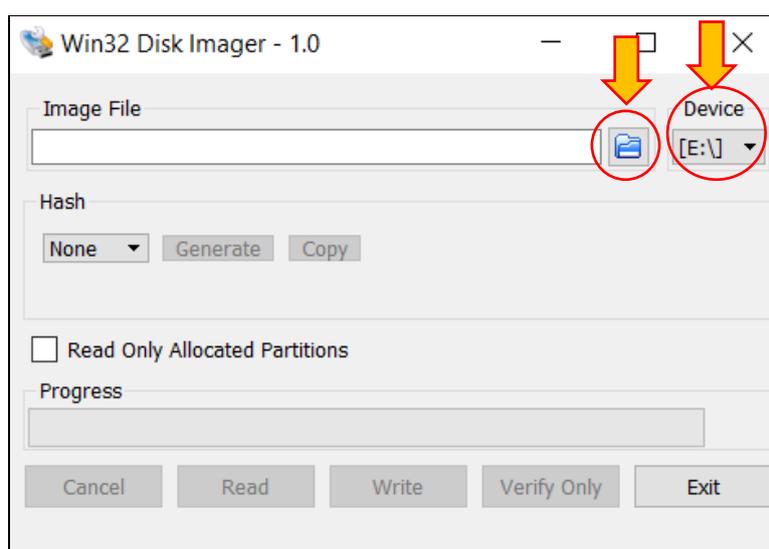
在章节【1.5 辅助工具及软件】中提到过，Win32DiskImager 软件可以将镜像文件烧写到对应的

SD 卡或 U 盘中，具体烧写方法如下：

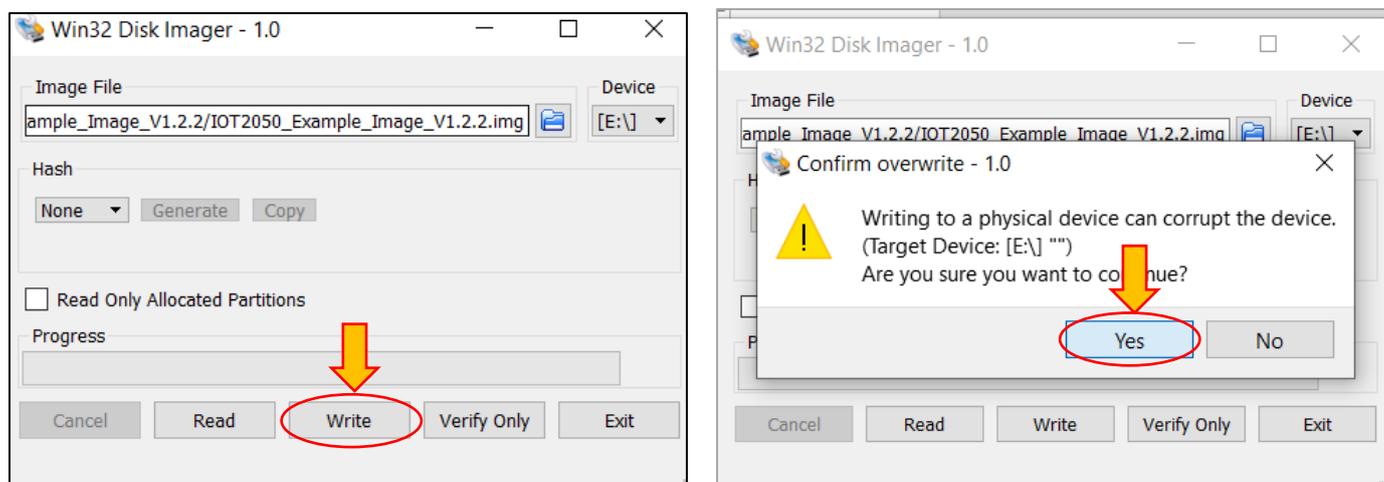
1) 双击 Win32DiskImager 图标运行软件，进入程序。



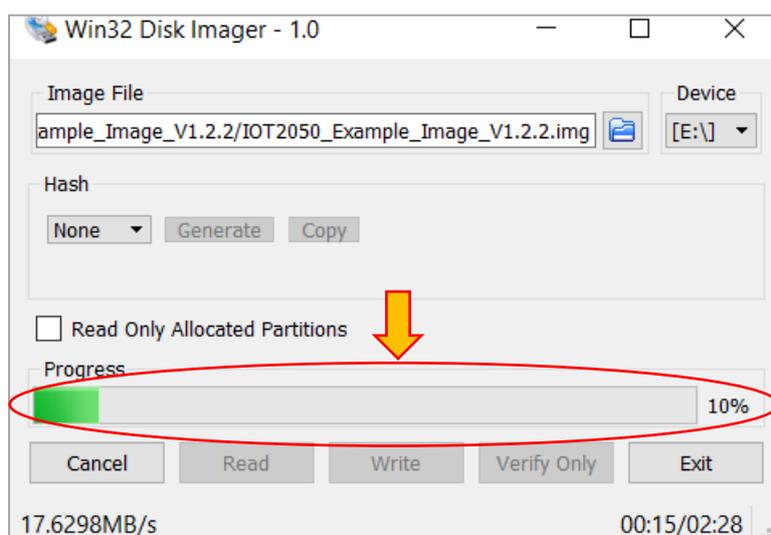
2) 可以通过“Device”选项选择需要烧写的盘符，一般无需更改。点击“Image File”选项右侧的文件夹，可以选择需要烧写的文件路径。



3.选择好需要烧写的镜像后，其他选项保持默认，点击下方的“Write”按钮后，会弹出提示对话框，点击“Yes”按钮后，程序开始烧写镜像文件。



4.烧写过程中，“Progress”下方的进度条中会显示程序烧写的进度。



5.烧写成功后，会弹出烧写完成对话框，点击“OK”按钮，退出程序。

由于镜像中烧入的是 Linux 系统，所以烧写后的 SD 卡将或 U 盘无法被 Windows 系统识别出，Windows 系统可能会要求格式化或无法显示容量大小。拔出 SD 卡或 U 盘，安装到 IOT2050 上，启动 IOT2050，就可以使用其中的应用了。

2.3 启动 SIMATIC IOT2050

将烧写有对应镜像应用程序的 SD 卡或 USB 插入到 IOT2050 后，接通电源（12-24VDC），就可以启动 IOT2050 了。

启动过程中，IOT2050 上的 LED 灯可以显示启动状态。根据应用程序中设置的不同，IOT2050 在启动时 LED 灯显示的状态也会不同。

如果 IOT2050 正常启动，则 PWR LED 灯会一直处于绿色常量状态，STAT LED 指示灯会先处于绿色长亮，而后转为绿色闪烁状态。

如果启动过程中，STAT LED 灯变为其它颜色状态，则表示系统没有启动成功，需要断开电源后重新启动或重新烧写程序后再次启动。

系统启动完成后，STAT LED 指示灯也会一直处于绿色闪烁状态（第一次启动时，完成启动所需时间会比较长）。此时，可以通过远程访问或其他方式进入 IOT2050，进行程序开发、软件设置等操作。

2.4 将镜像文件烧写到 EMMC

SIMATIC IOT2050 Advance 版本自带了 16G 的 EMMC 存储。前面章节中介绍过的应用镜像，可以直接被烧写到 IOT2050 的 EMMC 存储。烧写后，IOT2050 可以直接通过从 EMMC 启动后进入系统，执行相应的应用程序。

用户可以通过多种方法将需要使用的镜像文件烧写到 EMMC 中。

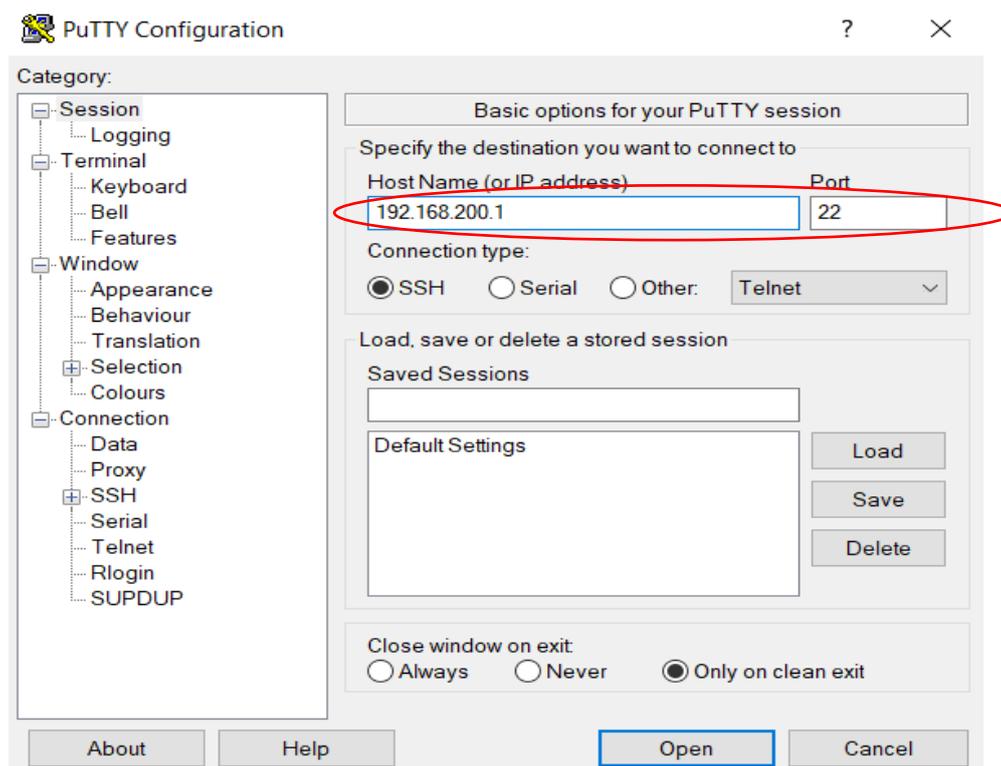
2.4.1 通过 SD 卡、U 盘和标准镜像文件进行烧写

可以使用一张 SD 和一张 U 盘将所需镜像烧写到 EMMC 中：

- 1) 准备好一张烧写有系统镜像（IOT2050_Example_Image_V1.1.1 或 IOT2050_Example_Image_V1.2.2）的 SD 卡和一张存放有需要移植到 EMMC 的应用镜像的 U 盘。
- 2) 按照图中所示将 U 盘插入 IOT2050 的 USB 接口。将烧写有系统镜像的 SD 卡插入 IOT2050 的 SD 卡插槽。使用网线连接电脑的网口与 IOT2050 的 P1 口。



- 3) 启动 IOT2050，在电脑中打开 Putty 软件，对连接进行设置，选择连接类型为 SSH，IP 地址输入 192.168.200.1，端口号使用默认的 22，保存设置后点击 open 进入系统的操作界面。此时系统会要求用户登陆，输入用户名小写的 root，输入密码小写的 root。（第一次登陆时，系统会要求用户更改密码，按照提示更改即可）。



4) 进入到系统后，就可以开始移植操作，依次输入下面的代码

① 输入 `mkdir/mnt/USB` 后回车 (为 USB 创建目录)

② 输入 `mount/dev/sda1/mnt/USB` 后回车(将 USB 挂接到 IOT2050)

③ 输入 `cd/mnt/USB` 后回车 (进入 USB 目录)

④ 输入 `ls` 后回车 (确认拷贝到 USB 中的镜像文件的名称)

⑤ 输入 `dd if=/mnt/USB/镜像文件名称 of=/dev/mmcblk1 bs=100M conv=fsync status=progress` 完成上述操作后，系统开始将 USB 中的镜像文件烧写到 EMMC 中。

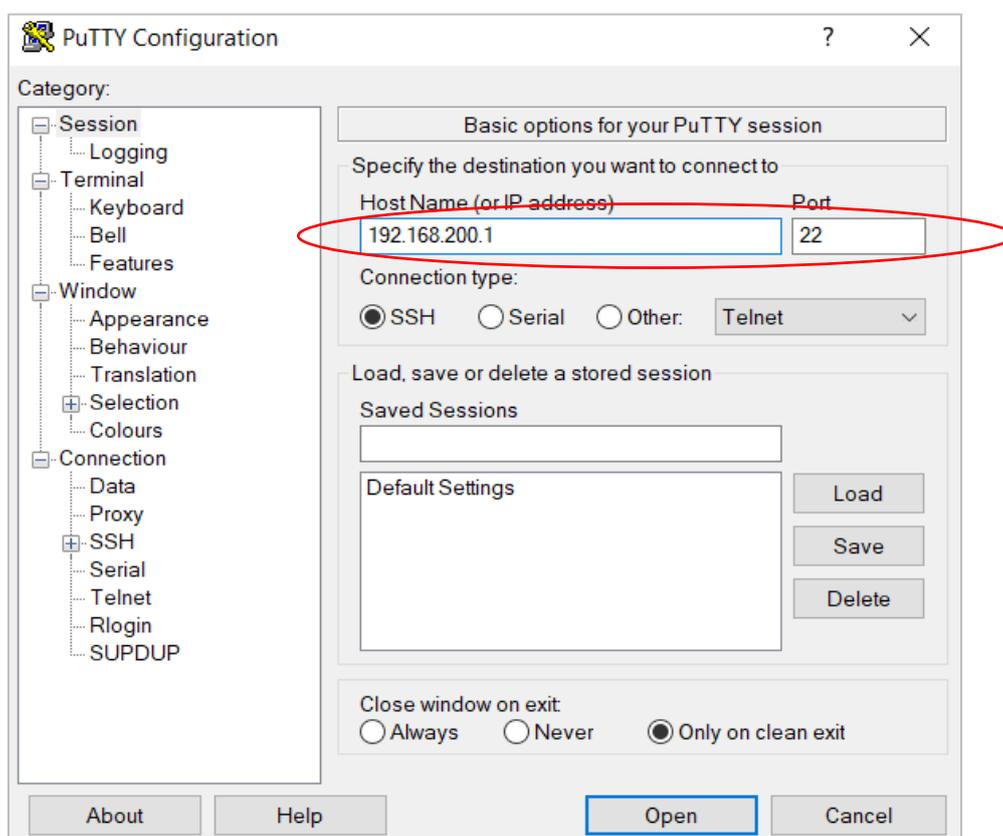
烧写完成，断开 IOT2050 的电源，拔出 SD 卡和 U 盘，重新启动 IOT2050 就会从 EMMC 启动进入系统了。

2.4.2 通过 SD 卡和 Flash 烧写专用应用镜像进行烧写

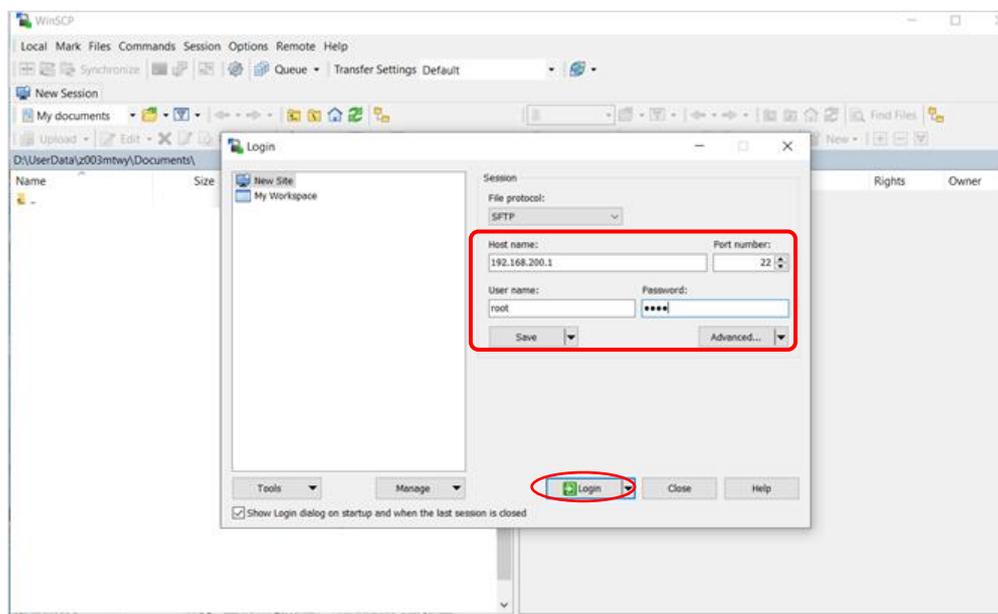
可以使用一张 SD 和专用的 Flash 烧写应用将镜像烧写到 EMMC 中

1) 将 Flash 烧写专用应用镜像烧写到 SD 卡中，将 SD 卡插入 IOT2050。使用网线连接电脑的网口与 IOT2050 的 P1 口。（Flash 烧写专用应用镜像可咨询销售或 IOT2050 产品经理获取。）

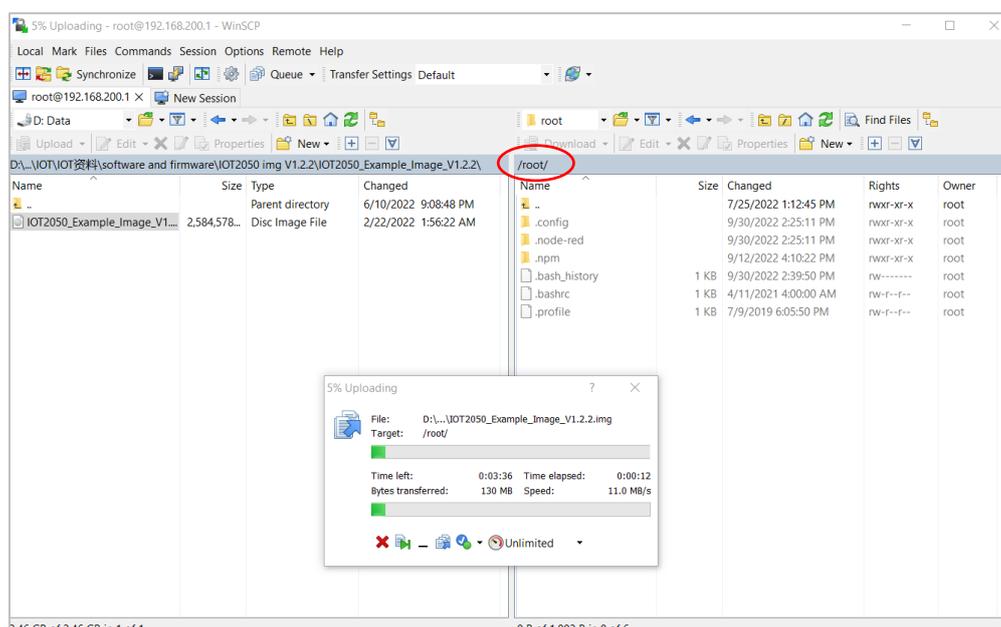
2) 启动 IOT2050，在电脑中打开 Putty 软件，对连接进行设置，选择连接类型为 SSH，IP 地址输入 192.168.200.1，端口号使用默认的 22，保存设置后点击 open 后，进入系统的操作界面。此时系统会要求用户登陆，输入用户名小写的 root，输入密码小写的 root。（第一次登陆时，系统会要求用户更改密码，按照提示更改即可）。



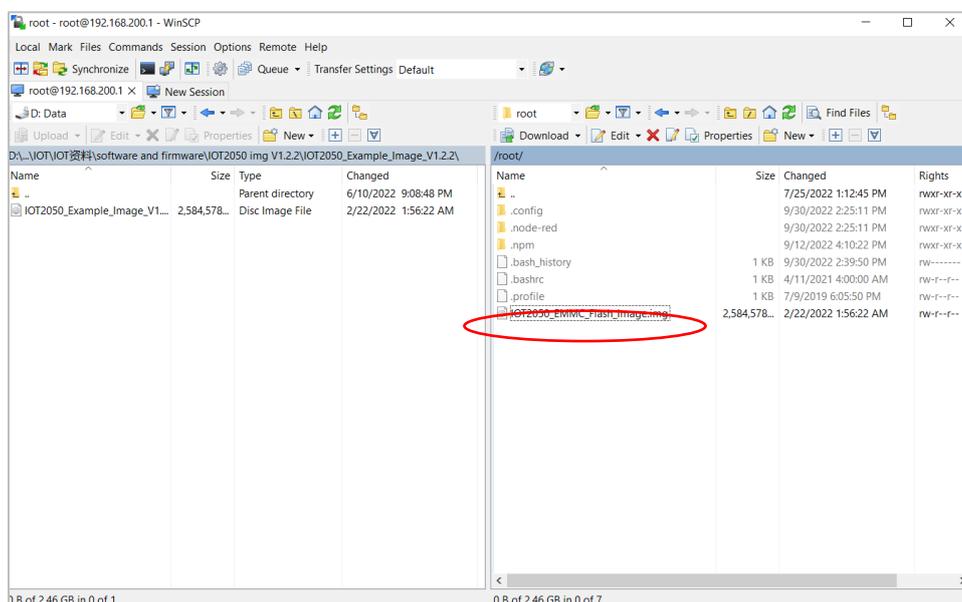
3) 在电脑中打开 WinSCP 软件，在弹出的 Login 页面中输入对应的参数后，点击 Login 按钮进入主页面。（在主页面中，左侧界面内容为所在电脑的对应文件夹中的内容，右侧界面为 IOT2050 的对应文件夹中的内容。）



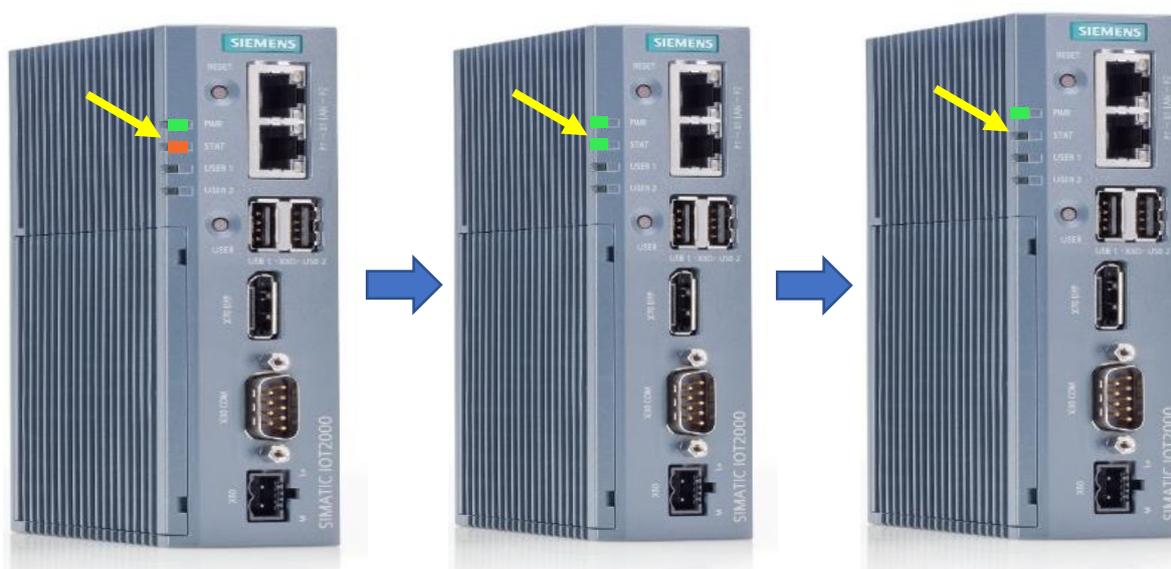
步骤 4：在 WinSCP 的右侧界面中 IOT2050 的文件夹位置定位为 /root 目录下.从左侧的界面中找到需要烧写到 EMMC 中的镜像文件并将其拖拽至右侧的 root 目录下。



步骤 5：将拖拽至 IOT2050 的 root 目录下的镜像文件更名为 IOT2050_EMMC_Flash_Image.img，关闭 WinSCP 软件，切断 IOT2050 电源。



步骤 6：重新上电启动 IOT2050，启动过程中，STAT 灯会切换至黄色并不断闪烁，在 20 秒内按下 User Button 直至 STAT 灯状态切换为绿色闪烁。松开 User Button 按钮，系统会自动进行烧录直至 STAT 灯状态切换为熄灭的状态，程序烧写完成。

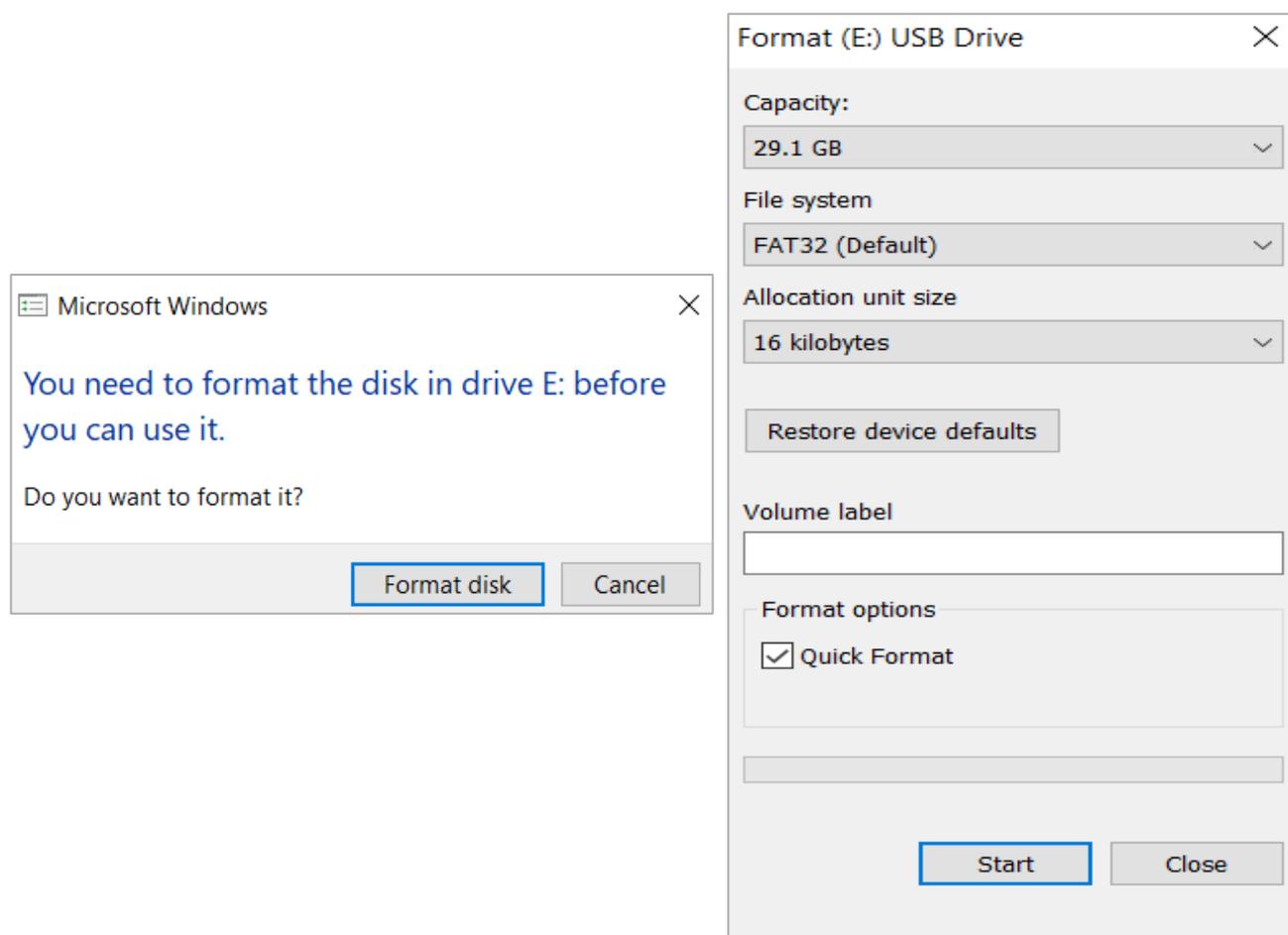


断开 IOT2050 的电源，拔出 SD 卡，重新启动，IOT2050 就会从 EMMC 启动进入系统了。

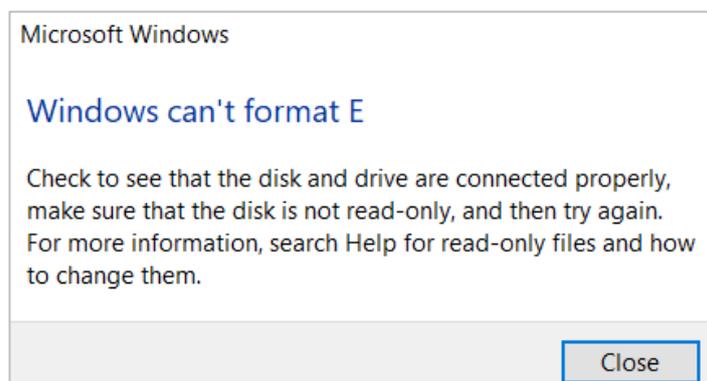
2.5 Windows 下格式化烧写有镜像文件的 SD 卡/U 盘

当需要对已经烧写有镜像文件的 SD 卡或 U 盘重新进行烧写时,由于烧写的到 SD 卡或 USB 里的镜像是基于 Linux 系统的, 写入镜像后的 SD 卡或 U 盘无法再被 Windows 系统正常识别到, 此时需要对 SD 卡或 U 盘重新进行格式化。

有些 windows 系统版本在插入 SD 卡或 U 盘后会直接弹出格式化对话框, 此时直接对 SD 卡或 U 盘进行格式化即可。

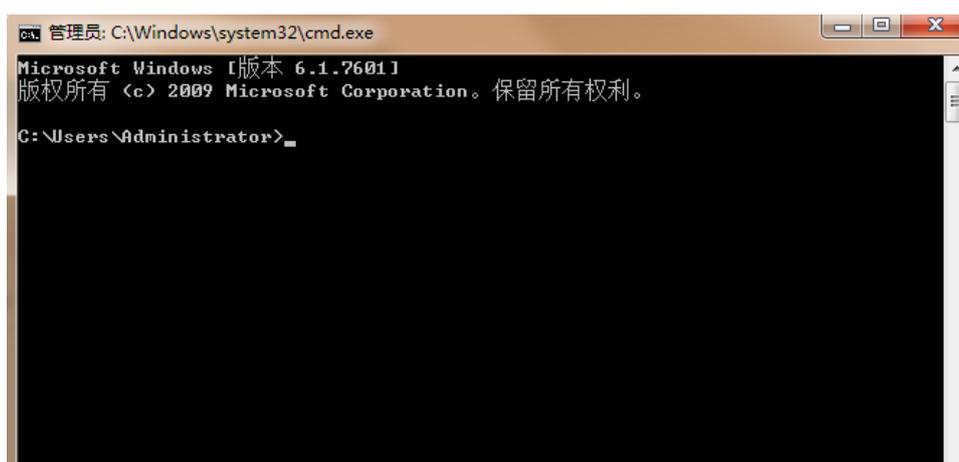


在某些系统中，有可系统可能无法对 SD 卡或 U 盘进行格式化，或在电脑中无法显示出 SD 卡或 U 盘的盘符。

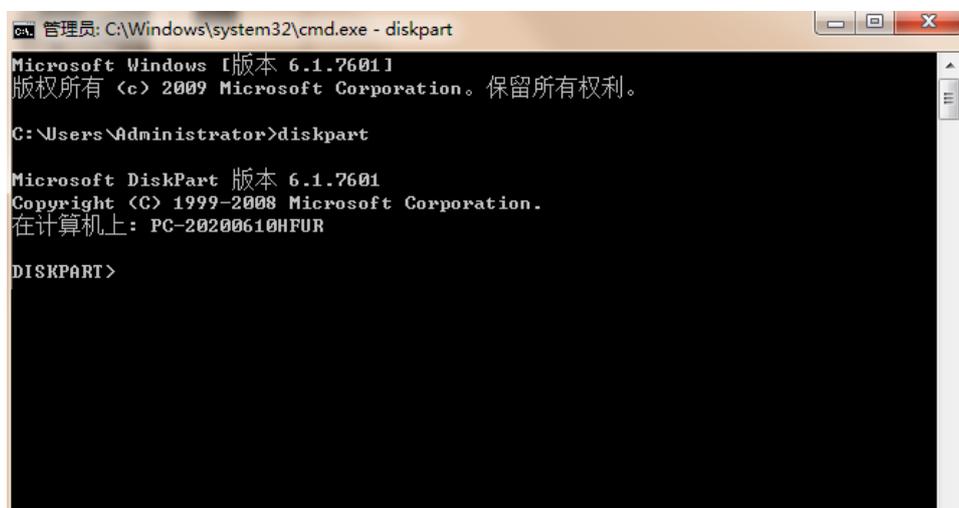


可以使用 Windows 自带的 diskpart 命令行工具来进行格式化：

1) 在搜索栏中输入 cmd 进入命令窗口。



2) 执行 diskpart 命令进行交互环境



3) 执行 list disk 命令查看当前 PC 连接的磁盘：



```
管理员: C:\Windows\system32\cmd.exe - diskpart
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>diskpart

Microsoft DiskPart 版本 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
在计算机上: PC-20200610HFUR

DISKPART> list disk

 磁盘 ###  状态      大小    可用    Dyn  Gpt
-----
 磁盘 0    联机      238 GB  1024 KB
 磁盘 1    联机      15 GB   14 GB

DISKPART>
```

4) 使用 select 命令选中 SD 卡或 U 盘对应的磁盘号（一般为 disk 1），注意不要选择错磁盘，输入 select disk 1 指令选中 disk 1。



```
管理员: C:\Windows\system32\cmd.exe - diskpart
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>diskpart

Microsoft DiskPart 版本 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
在计算机上: PC-20200610HFUR

DISKPART> list disk

 磁盘 ###  状态      大小    可用    Dyn  Gpt
-----
 磁盘 0    联机      238 GB  1024 KB
 磁盘 1    联机      15 GB   14 GB

DISKPART> select disk 1

磁盘 1 现在所选磁盘。

DISKPART>
```

5) 执行 clean 命令清空 U 盘（命令执行完后，系统会弹出磁盘格式化对话框，由于磁盘没有设置主活动分区，所以无法执行格式化，但此时系统中会显示出盘符）

```
管理员: C:\Windows\system32\cmd.exe - diskpart
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>diskpart

Microsoft DiskPart 版本 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
在计算机上: PC-20200610HFUR

DISKPART> list disk

  磁盘 ###  状态      大小    可用    Dyn  Gpt
-----
  磁盘 0    联机      238 GB  1024 KB
  磁盘 1    联机      15 GB   14 GB

DISKPART> select disk 1

磁盘 1 现在是所选磁盘。

DISKPART> clean

DiskPart 成功地清除了磁盘。

DISKPART>
```

6) 执行 `create partition primary` 命令创建一个新的主分区。

```
管理员: C:\Windows\system32\cmd.exe - diskpart

  磁盘 0    联机      238 GB  1024 KB
  磁盘 1    联机      15 GB   14 GB

DISKPART> select disk 1

磁盘 1 现在是所选磁盘。

DISKPART> clean

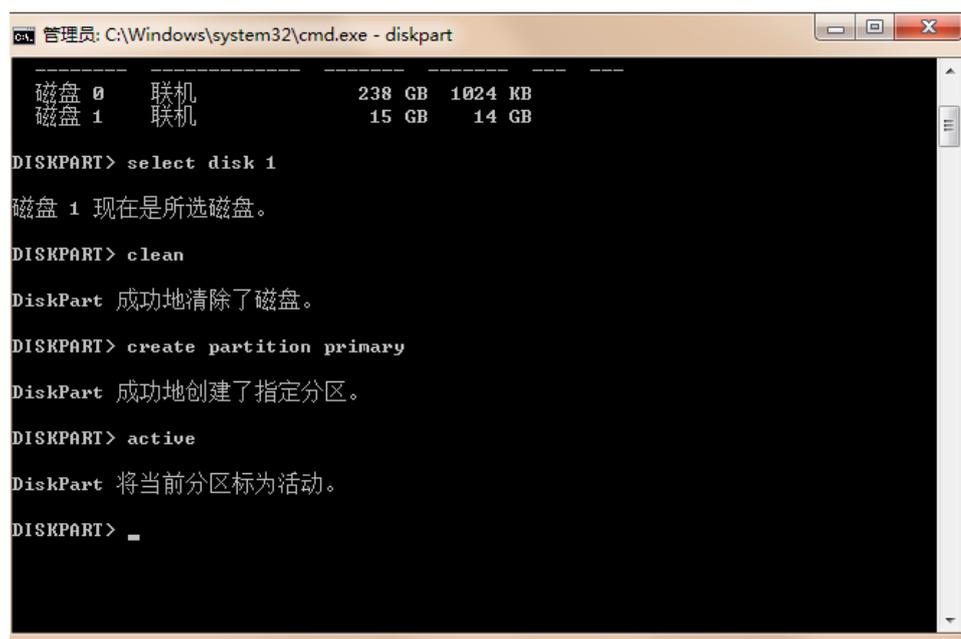
DiskPart 成功地清除了磁盘。

DISKPART> create partition primary

DiskPart 成功地创建了指定分区。

DISKPART> _
```

7) 执行 `active` 命令激活该分区。



```
管理员: C:\Windows\system32\cmd.exe - diskpart
-----
磁盘 0  联机          238 GB  1024 KB
磁盘 1  联机          15 GB   14 GB

DISKPART> select disk 1
磁盘 1 现在是所选磁盘。

DISKPART> clean
DiskPart 成功地清除了磁盘。

DISKPART> create partition primary
DiskPart 成功地创建了指定分区。

DISKPART> active
DiskPart 将当前分区标为活动。

DISKPART> _
```

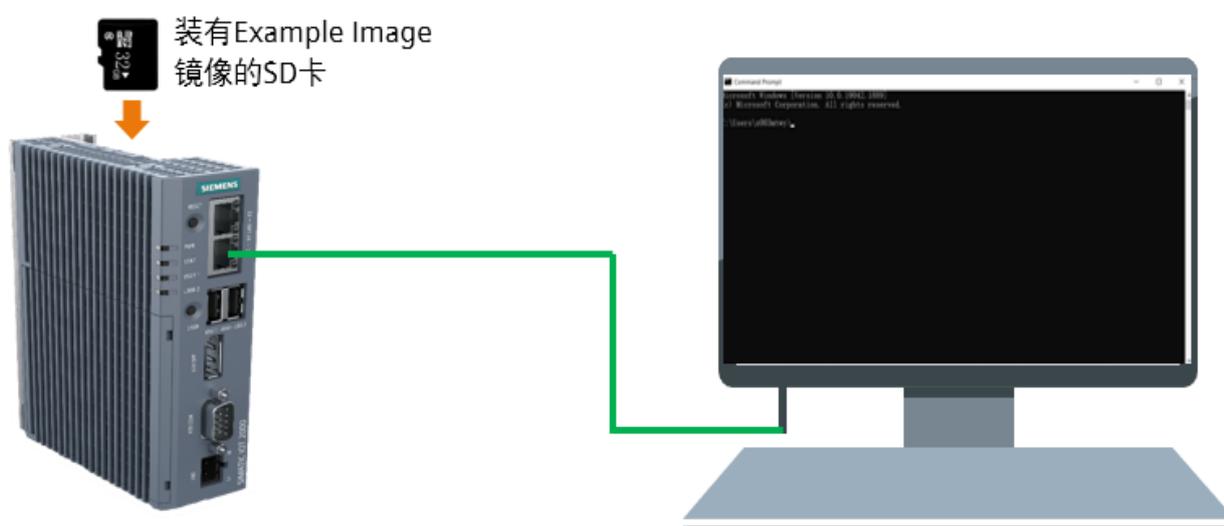
上述操作完成后，就可以通过右键单击磁盘对应盘符，进行格式化操作了。

3.Example Image 设置方法

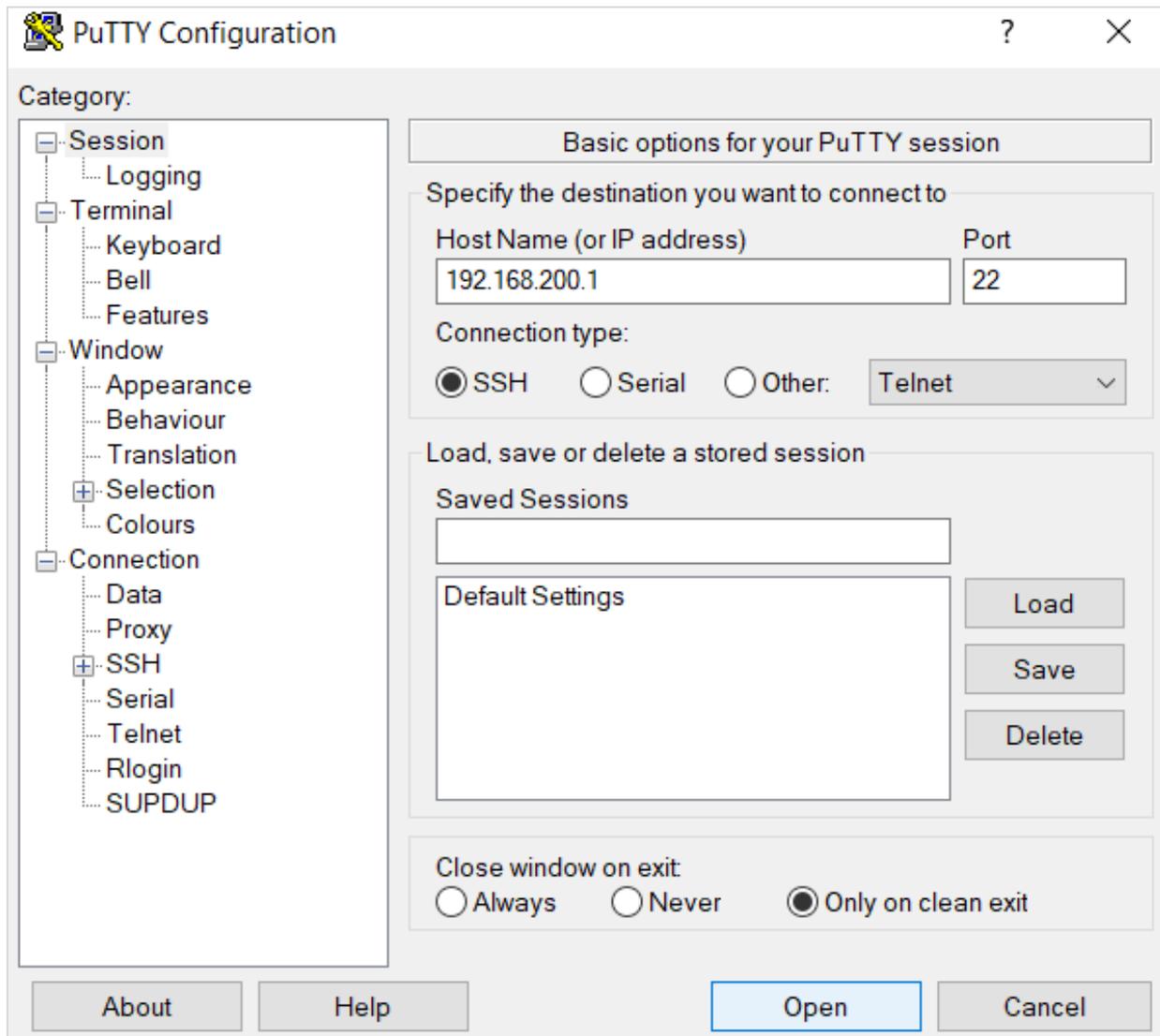
3.1 系统登陆

Example Image 是面向有开发需求的使用者的，集成了 Debian Linux 系统和 Node-red 应用、开放用户 root 权限的镜像文件。用户可以以 root 管理员的身份登陆系统，登陆后，就可以进行系统更新，软件安装，程序开发等各项操作了。

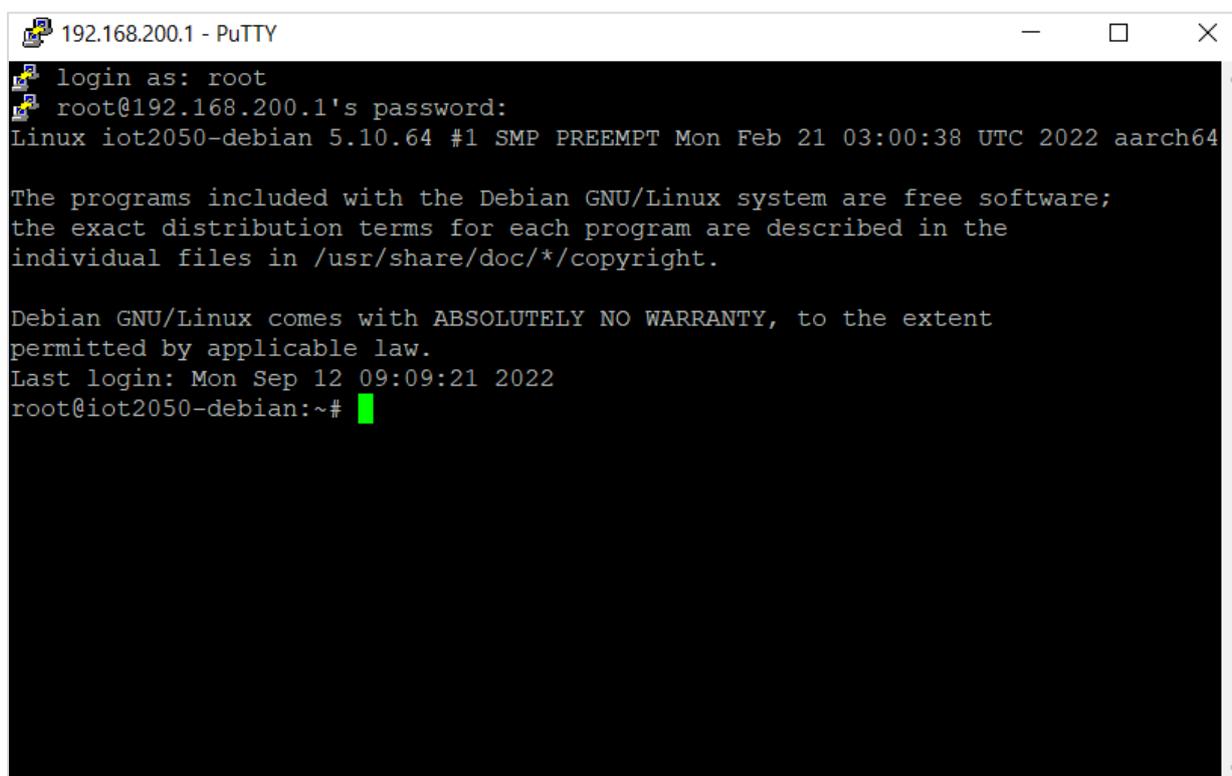
用户可以使用 Putty 软件通过 SSH 连接登陆 IOT2050 系统，电脑与 IOT 的连接如图所示。



Example Image 镜像中默认将 P1 口的 IP 地址设置为了 192.168.200.1，启动 Putty 软件后，可以通过输入 IP 地址建立与 IOT 的 SSH 连接，进入系统登陆页面。



输入用户名“root”和密码“root”后，即可以 root 管理员身份登陆进入系统进行操作。



```
192.168.200.1 - PuTTY
login as: root
root@192.168.200.1's password:
Linux iot2050-debian 5.10.64 #1 SMP PREEMPT Mon Feb 21 03:00:38 UTC 2022 aarch64

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.

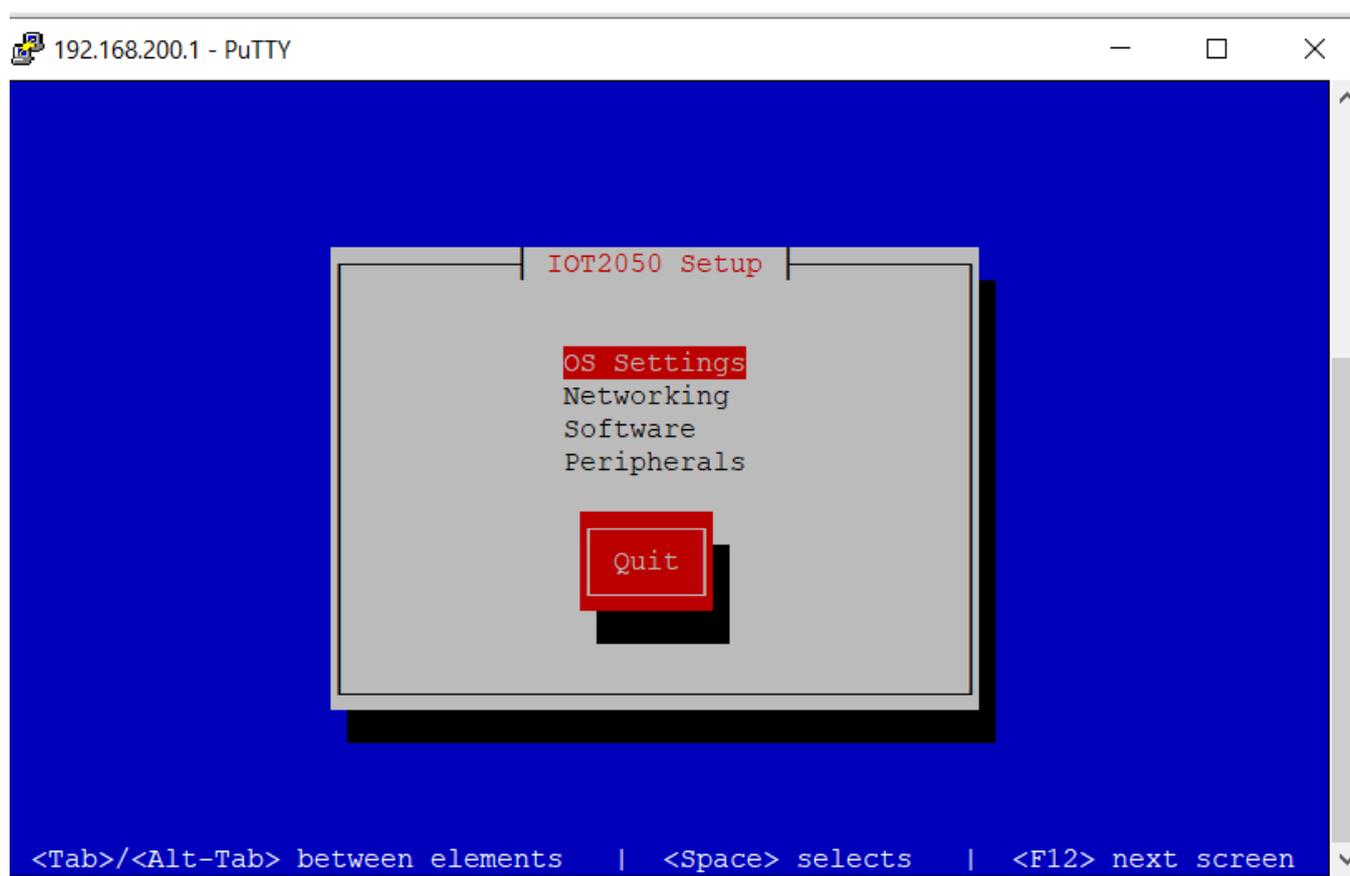
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Sep 12 09:09:21 2022
root@iot2050-debian:~#
```

注意：第一次登陆系统时，系统会要求用户更改密码，此时按照系统提示，先输入原密码，再输入两次新密码后，即可完成新密码的设定。新密码不能与原始密码重复，如果用户想继续使用 root 作为密码，则在更改完密码进入系统后，可以通过输入 passwd 命令将密码更改为初始密码。

进入系统后，用户就可以根据自身需要，进行系统更新，软件安装，程序开发等各项操作了。

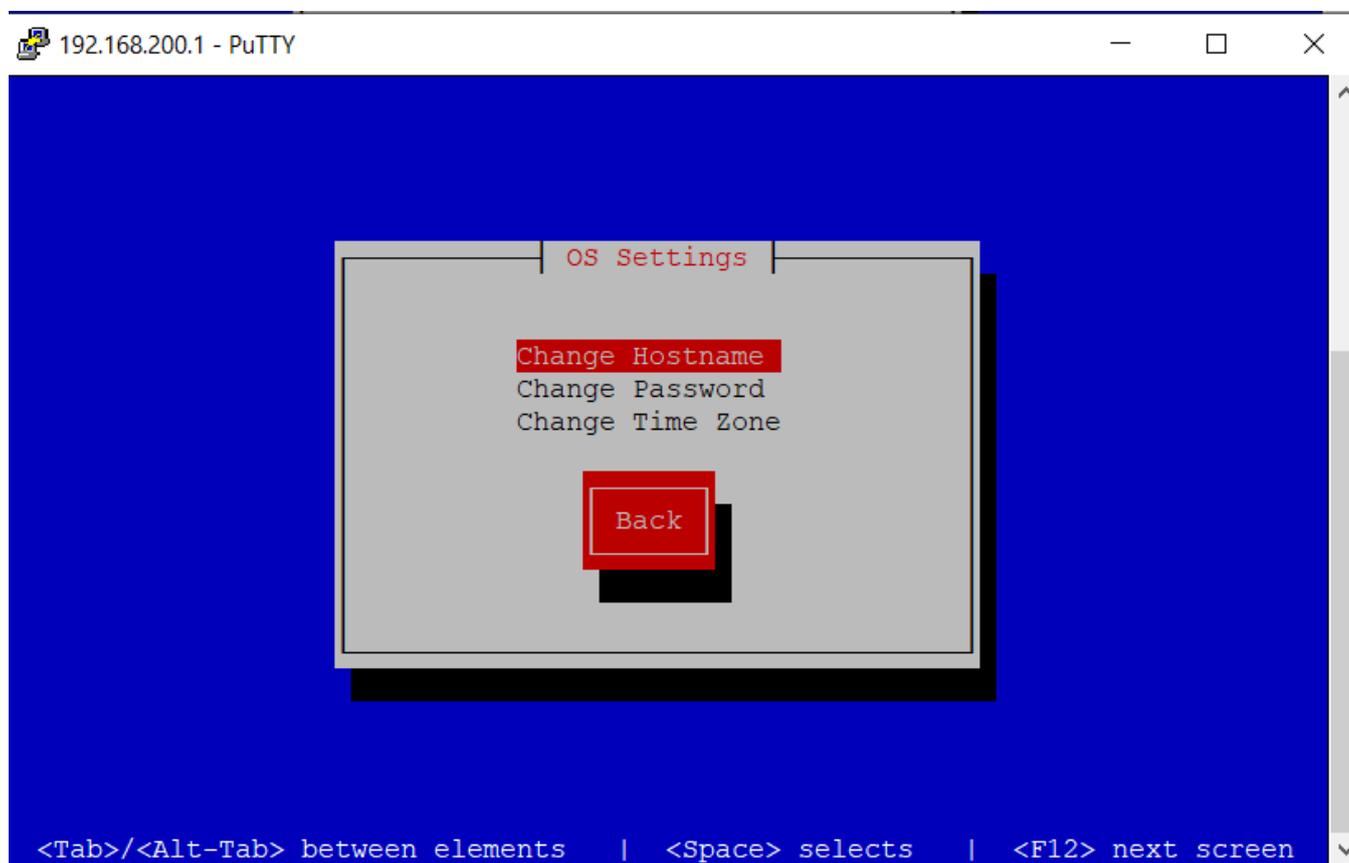
3.2 参数设置

以 root 管理员身份登入 Example image 的系统后，可以使用 `iot2050setup` 命令对 IOT2050 的一些参数进行设置。输入 `iot2050setup` 后回车，可以进入系统的参数设置页面，可以对 OS，网络，软件和硬件接口进行设置。



3.2.1 OS Settings 设置

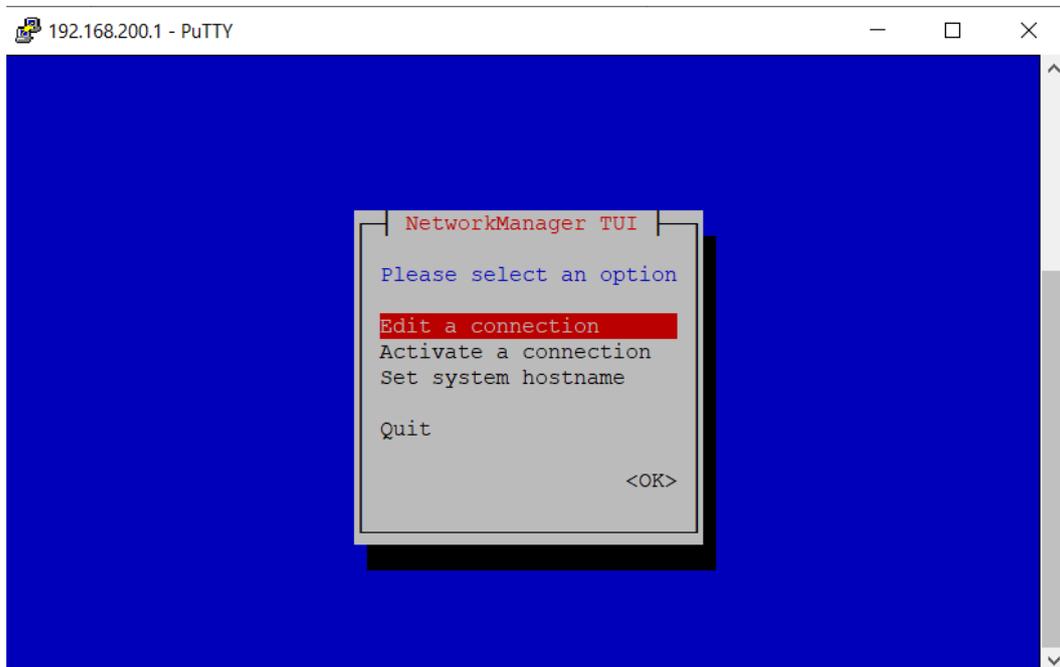
在参数设置页面中使用方向键将光标移动至 OS Setting 项后回车，进入 OS Setting 设置页面。



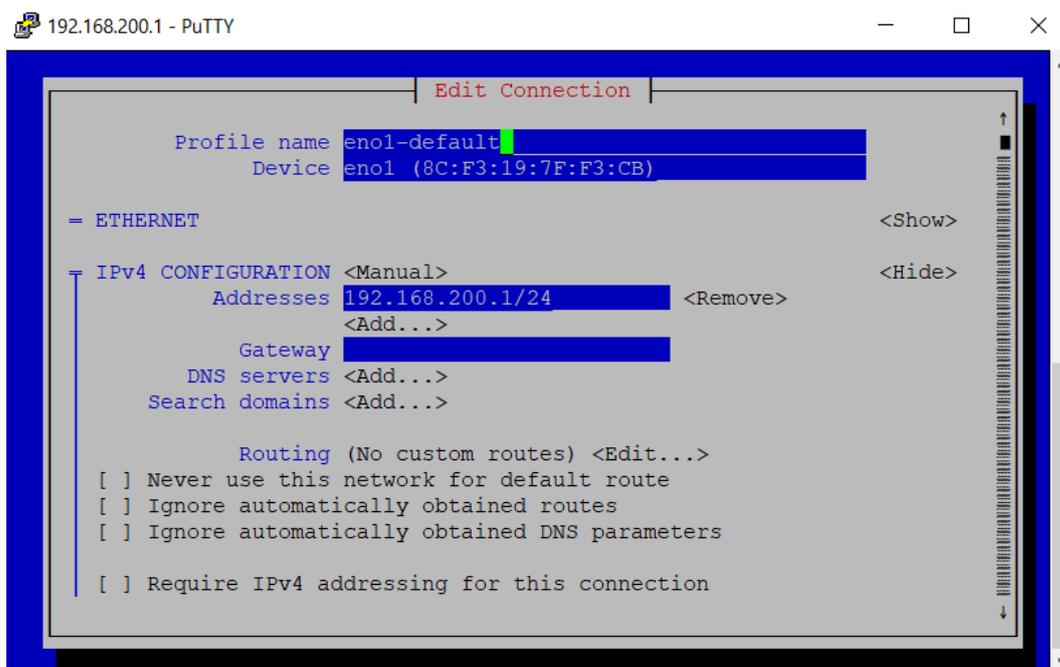
在该页面中，可以设置主机名称，更改密码和设置时钟时区。

3.2.2 Networking 设置

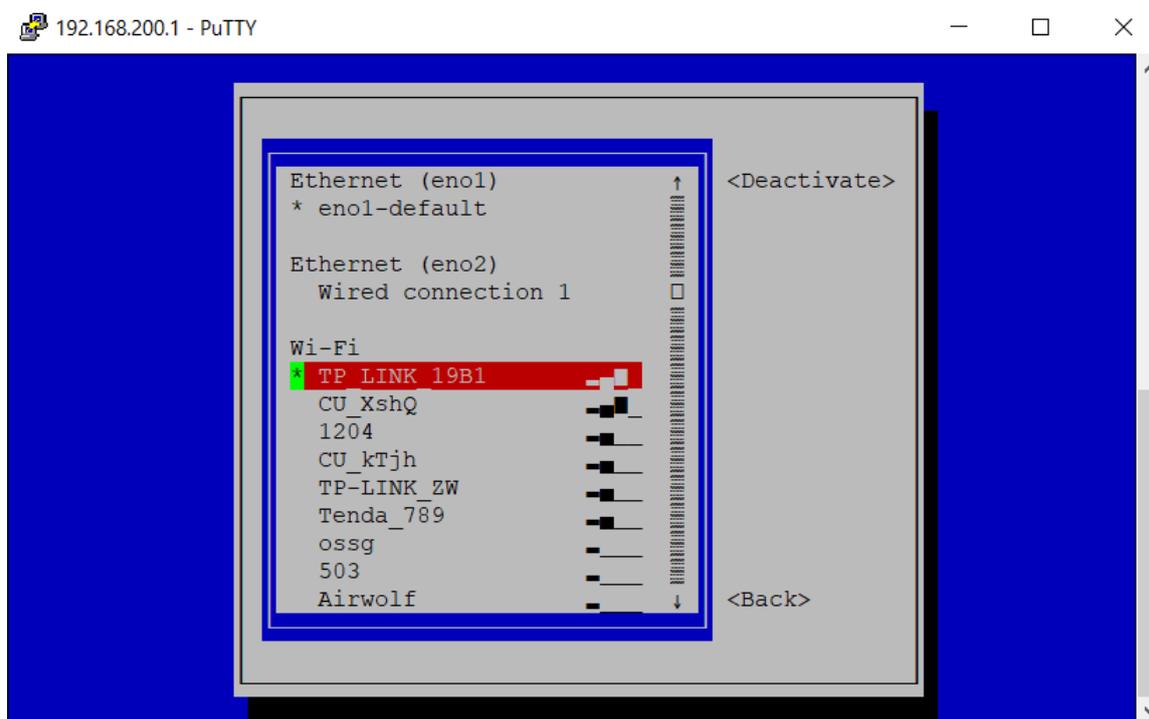
在参数设置页面中使用方向键将光标移动至 Networking 项后回车，进入 Networking 设置页面。



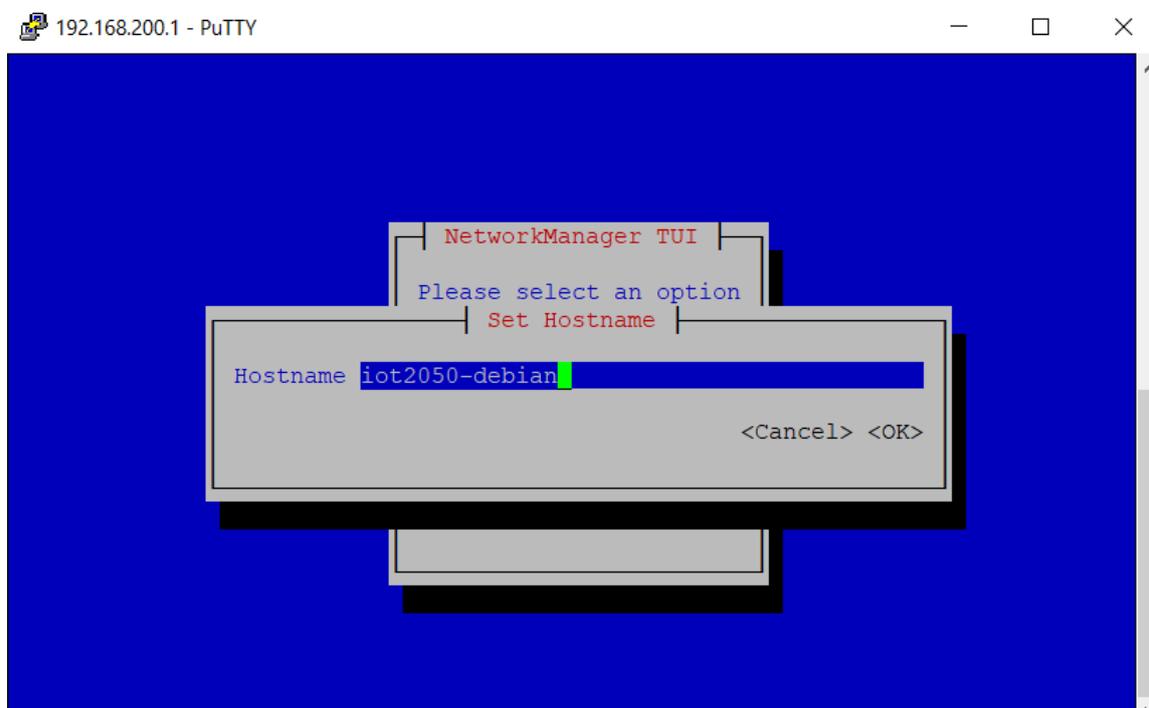
通过 Edit a connection 选项可以设置不同连接的参数，例如 IP 地址，子网掩码等



通过 Active a connection 选项可以查看或激活网络连接。

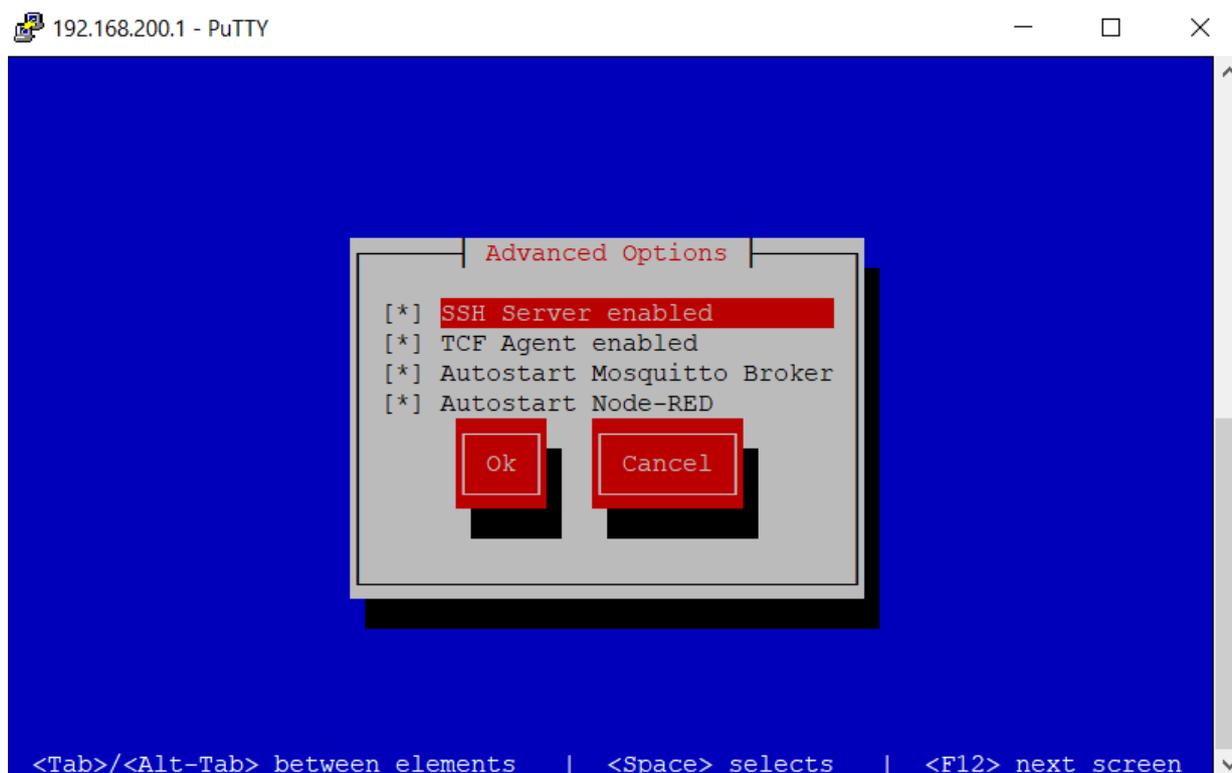


通过 Set system hostname 选项可以设置 IOT2050 的主机名称



3.2.3 Software 设置

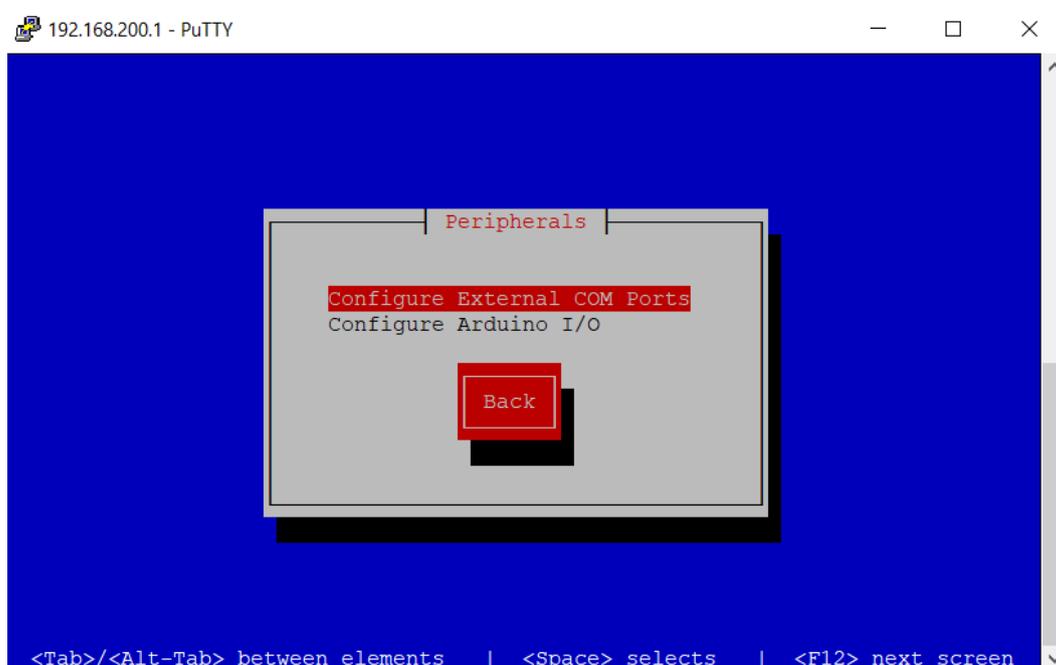
在参数设置页面中使用方向键将光标移动至 Software 项后回车，进入 Software 设置页面。



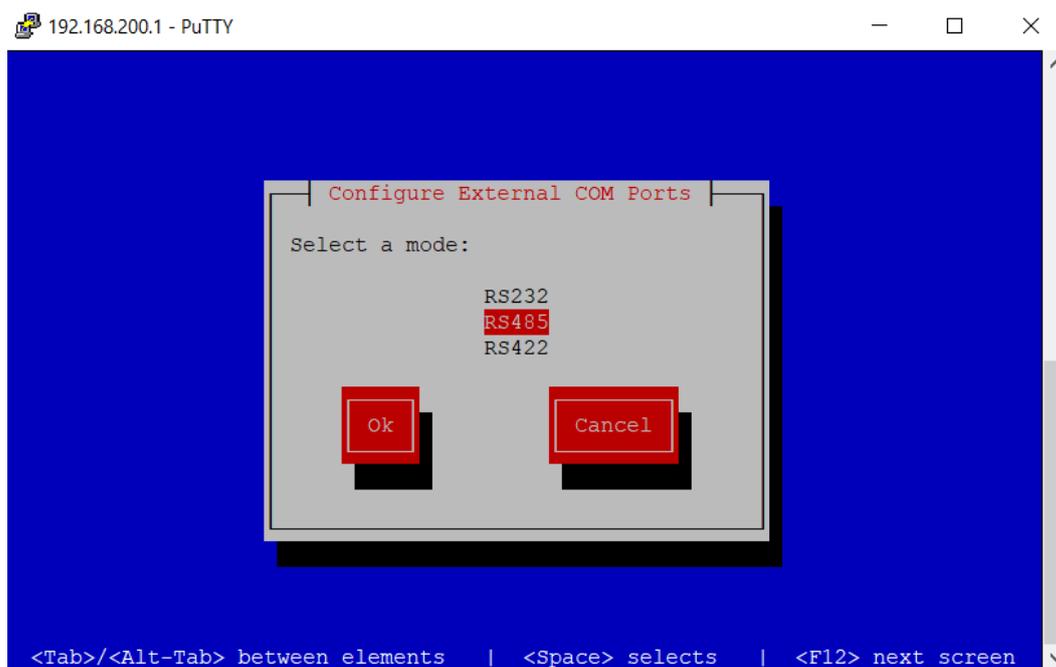
通过 Software 选项可以开启或关闭服务及自启动的应用程序。

3.2.4 Peripherals 设置

在参数设置页面中使用方向键将光标移动至 Peripherals 项后回车，进入 Peripherals 设置页面



通过 Configure External COM Ports 选项可以设置 IOT2050 上 COM 端口的模式。



通过 Configure Arduino I/O 选项可以设置 IOT2050 上 Arduino 扩展接口。

The screenshot shows a PuTTY terminal window titled "192.168.200.1 - PuTTY". The main content is a table with 5 columns: Pin, GPIO_Type, GPIO_Type, GPIO_Type, and Peripheral. The table lists configurations for pins IO3 through IO19. Below the table, there is a red header "Enable GPIO" followed by a list of enabled hardware modules: "Enable I2C on IO18 & IO19", "Enable SPI on IO10-IO13", "Enable UART on IO0-IO3", "Enable PWM on IO4-IO9", and "Enable ADC on IO14-IO19". The terminal also shows a blue bar on the left with "<Tab" and a blue bar on the right with "reen".

IO3	GPIO_Input	GPIO_Input	GPIO_Output	UART_RTS
IO4	GPIO_Input	GPIO_Input	GPIO_Output	PWM_4
IO5	GPIO_Input	GPIO_Input	GPIO_Output	PWM_5
IO6	GPIO_Input	GPIO_Input	GPIO_Output	PWM_6
IO7	GPIO_Input	GPIO_Input	GPIO_Output	PWM_7
IO8	GPIO_Input	GPIO_Input	GPIO_Output	PWM_8
IO9	GPIO_Input	GPIO_Input	GPIO_Output	PWM_9
IO10	GPIO_Input	GPIO_Input	GPIO_Output	SPI_SS
IO11	GPIO_Input	GPIO_Input	GPIO_Output	SPI_MOSI
IO12	GPIO_Input	GPIO_Input	GPIO_Output	SPI_MISO
IO13	GPIO_Input	GPIO_Input	GPIO_Output	SPI_SCK
IO14	GPIO_Input	GPIO_Input	GPIO_Output	ADC_0
IO15	GPIO_Input	GPIO_Input	GPIO_Output	ADC_1
IO16	GPIO_Input	GPIO_Input	GPIO_Output	ADC_2
IO17	GPIO_Input	GPIO_Input	GPIO_Output	ADC_3
IO18	GPIO_Input	GPIO_Input	GPIO_Output	I2C_SDA ADC_4
IO19	GPIO_Input	GPIO_Input	GPIO_Output	I2C_SCL ADC_5

Enable GPIO

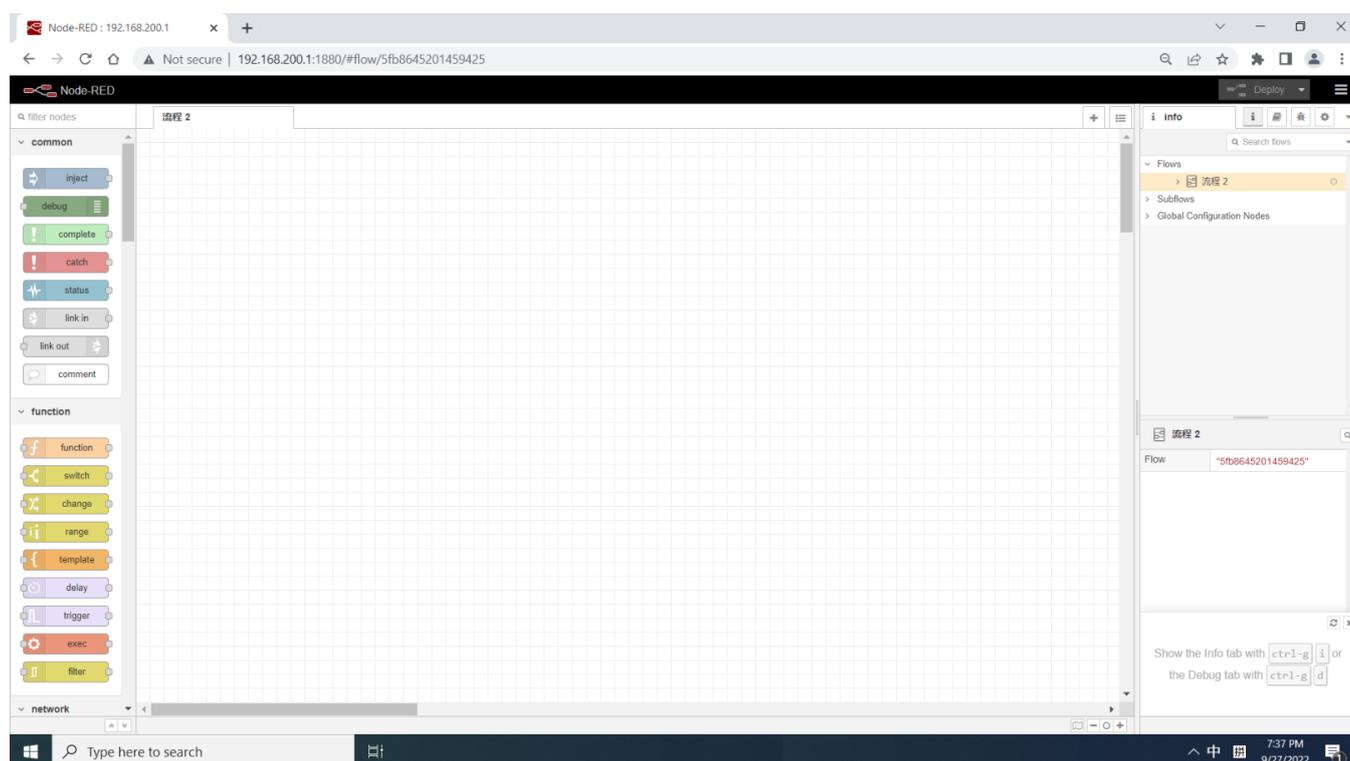
- Enable I2C on IO18 & IO19
- Enable SPI on IO10-IO13
- Enable UART on IO0-IO3
- Enable PWM on IO4-IO9
- Enable ADC on IO14-IO19

4. Node-Red 应用

Node-RED 是 IBM 开发的一款基于数据流(dataflow)的可视化编程工具，可基于浏览器的流程编辑器连接设备、服务器和 API 应用，在工业物联网和控制中已经被广泛应用。

4.1 启动 Node-Red 应用

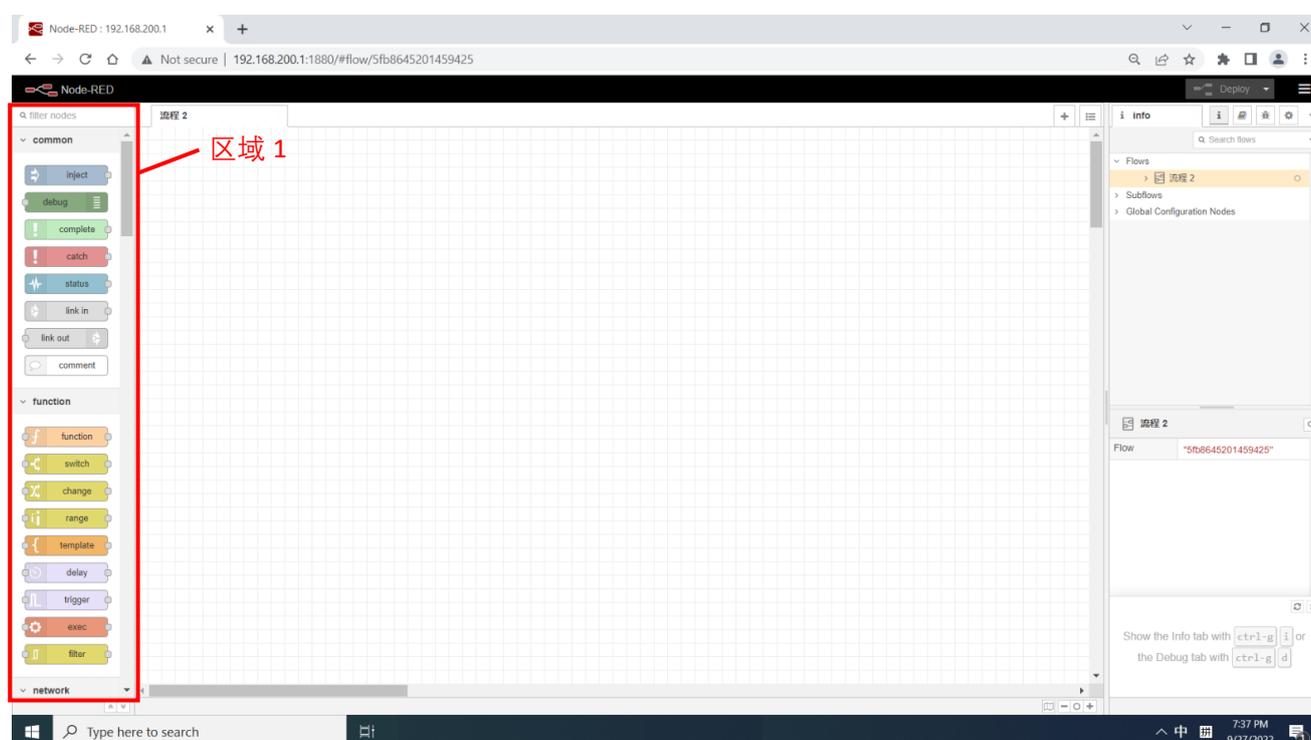
在章节【1.4 IMG 镜像文件说明】中提到过的西门子可提供的两种镜像文件中均带有 Node-RED 应用。连接好设备，IOT2050 正常启动运行后，可以通过访问网址 192.168.200.1:1880 直接访问 Node-RED 应用。



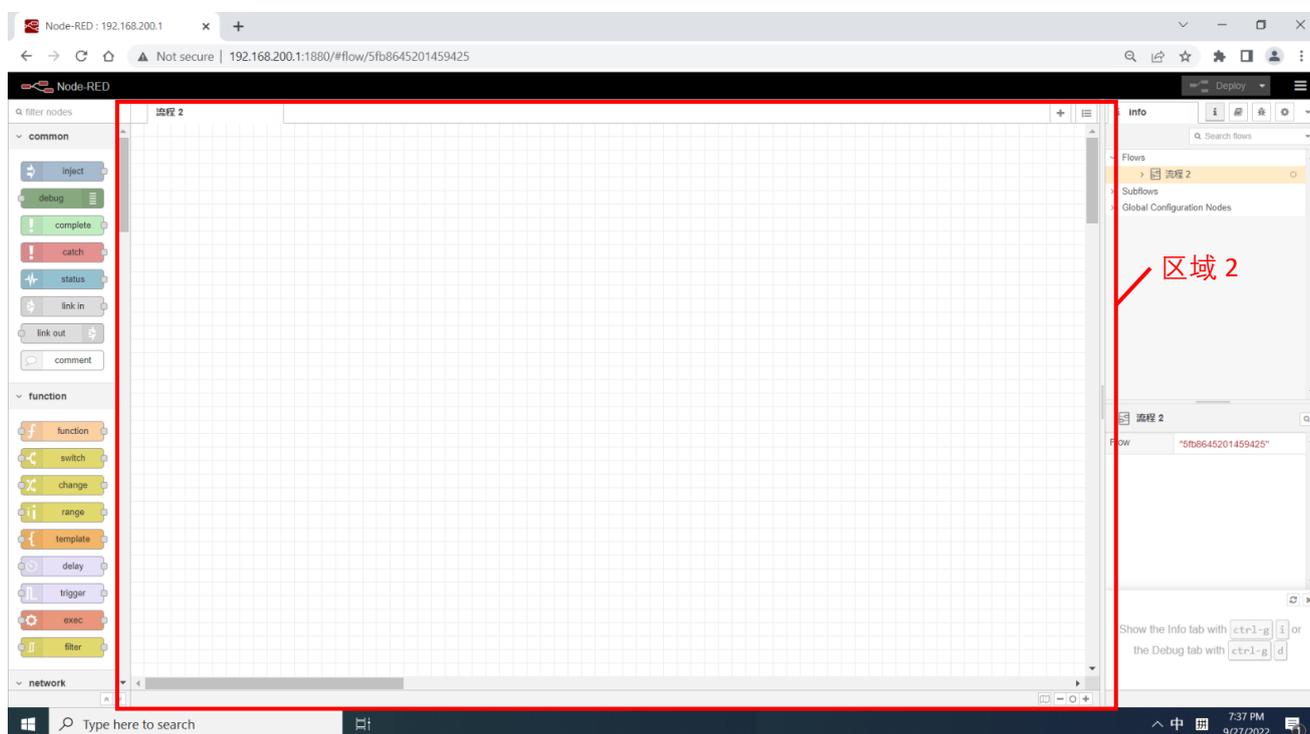
4.2 Node-Red 应用简介

Node-Red 的操作界面包含四个主要区域。区域 1 中包含了多个节点，每个节点可以实现相应的功能，这块区域为节点区。

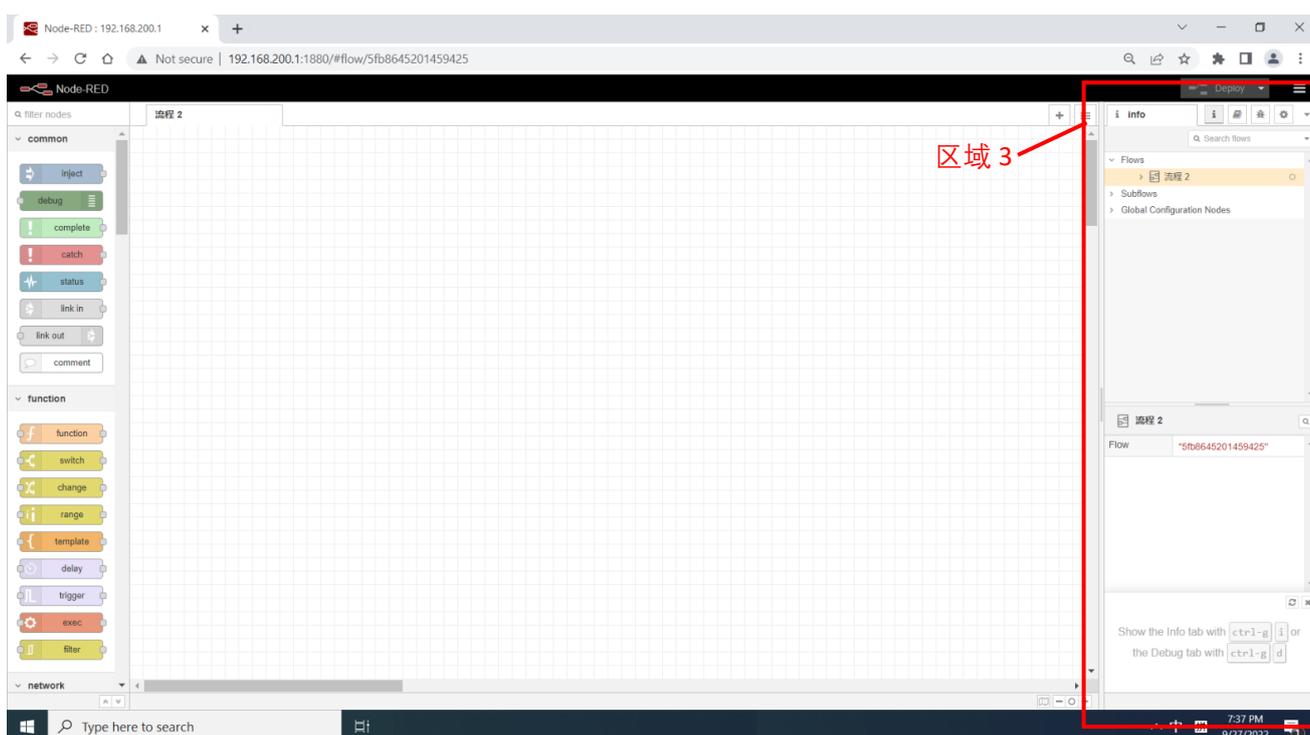
IOT2050 安装的 Node-RED 当中，已经提前为用户安装了开发时常用的不同功能类别的节点，包括通用类节点，功能类节点，协议解析类节点等等。Node-RED 提供了很多开源节点库，用户也可以根据需求自行安装或删除相关的功能节点。



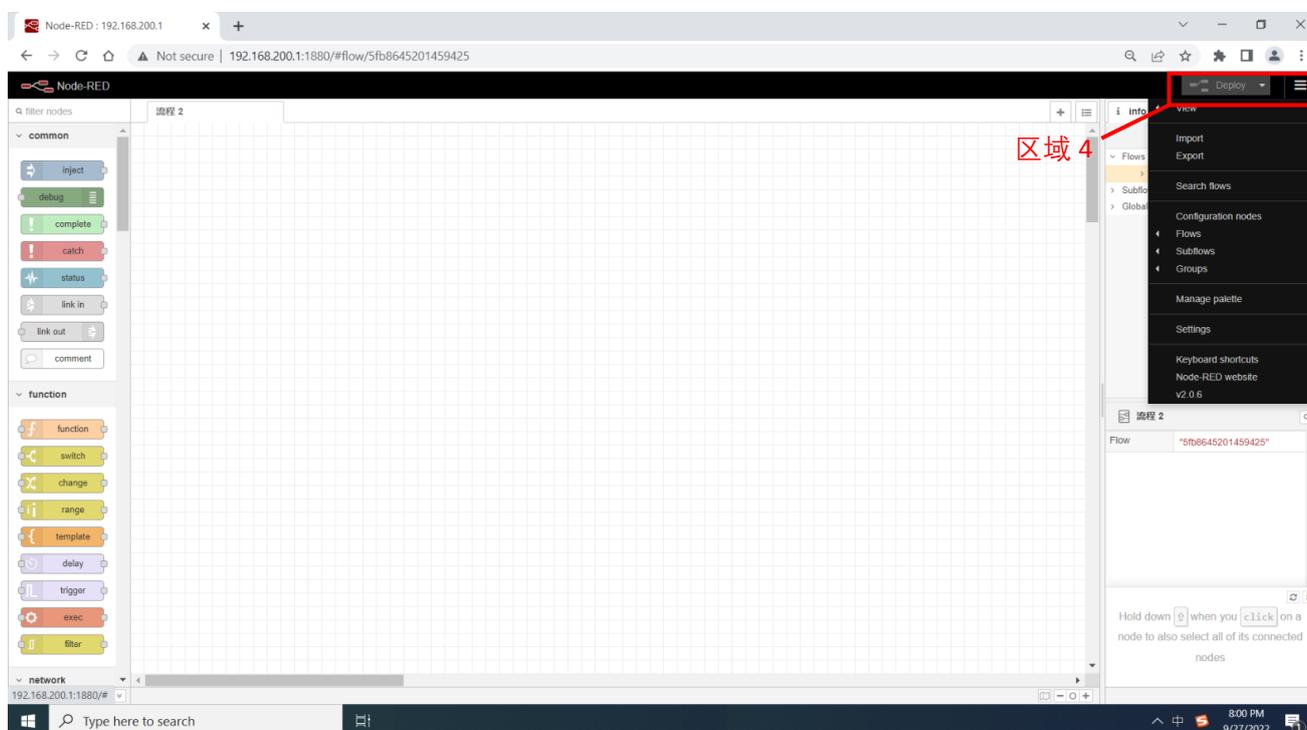
区域 2 为工作区，用户可以将节点区的节点拖拽到工作区，在工作区中对节点进行设置和编辑，就可以实现想要的功能和效果。



区域 3 为信息显示区域，在该区域中，用户可以查看节点，调试，帮助及 UI 设置的相关信息,并进行相关操作。



区域 4 为 Node-RED 的工具栏，通过部署按钮用户可以将编辑好的最新的程序部署到 IOT2050 中并开始执行。单机右侧的工具栏图标，会显示出下拉的功能列表，点击后可进行相关的设置和操作。



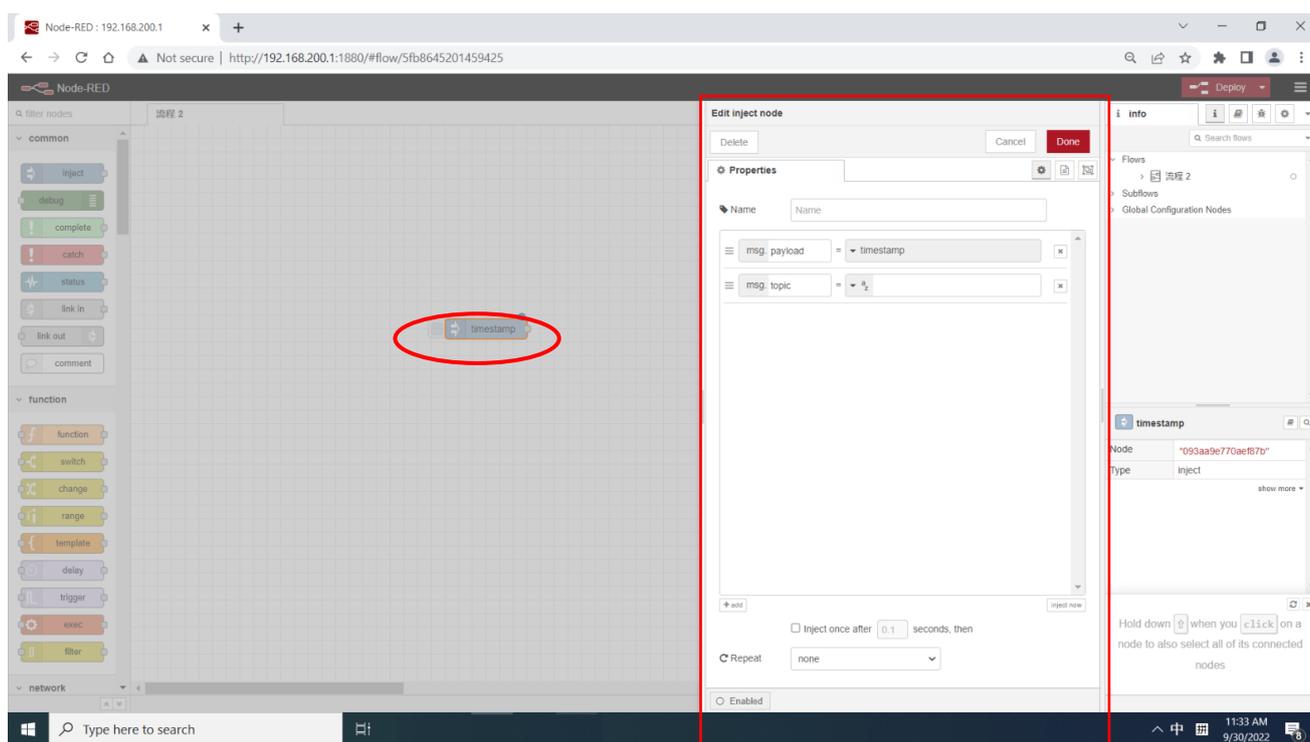
4.3 Node-Red 三个常用节点

在使用 Node-red 进行应用开发时，不同的应用会使用到各种不同功能的节点。其中有三个非常重要的节点，几乎在开发任何程序时，都会使用到，这三个节点分别是：inject 节点，debug 节点和 function 节点。

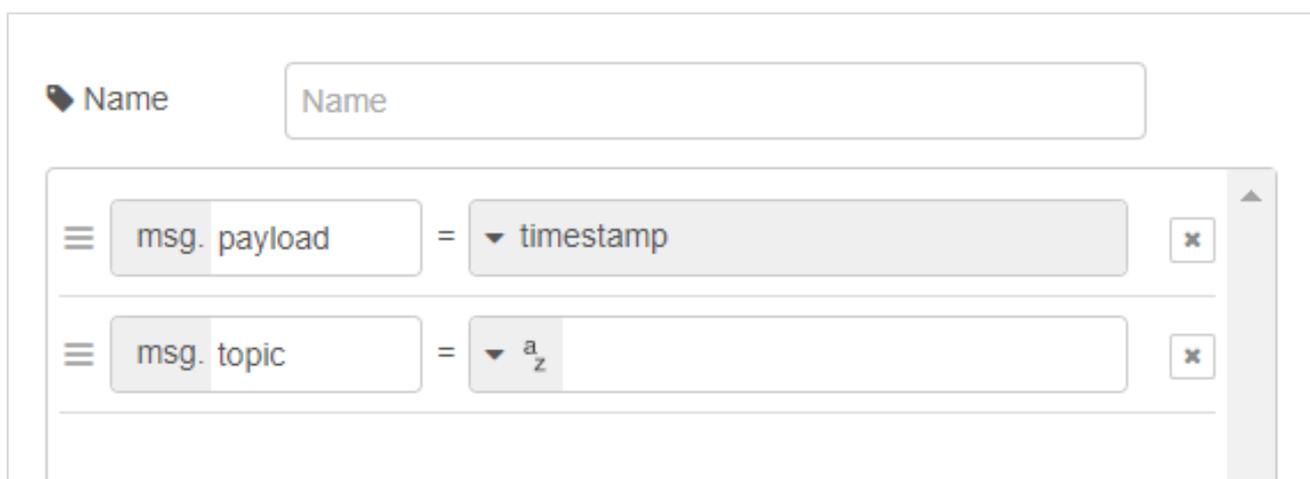
inject（时间戳）节点：

通过 inject 节点，可以手动或定期地将消息注入到消息流中。

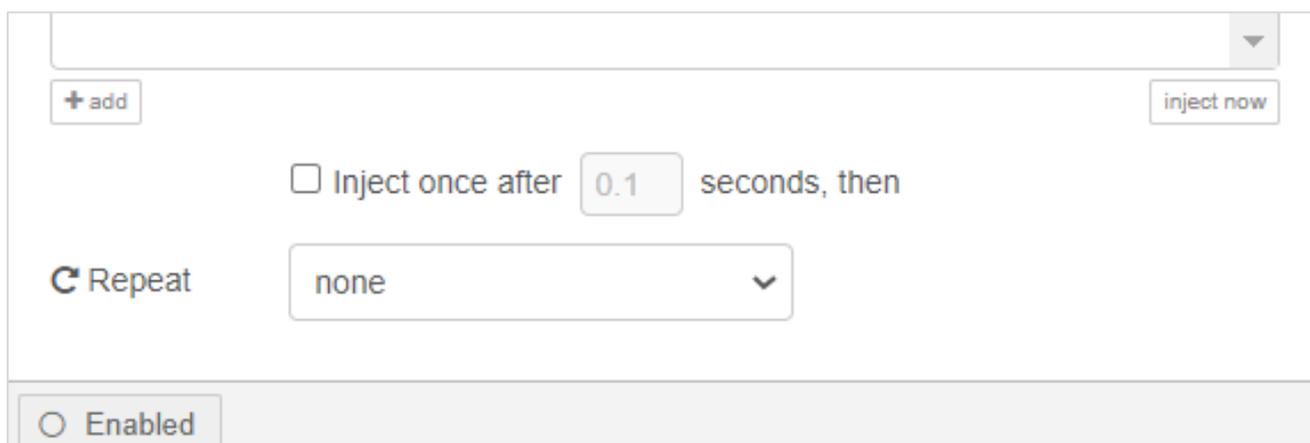
拖动 inject 节点到工作区域中，双击 inject 节点，就会弹出节点的设置界面。



在设置界面中，可以设置节点名称，消息流的属性（默认值为 payload 和 topic），属性所对应的数据类型和数值。



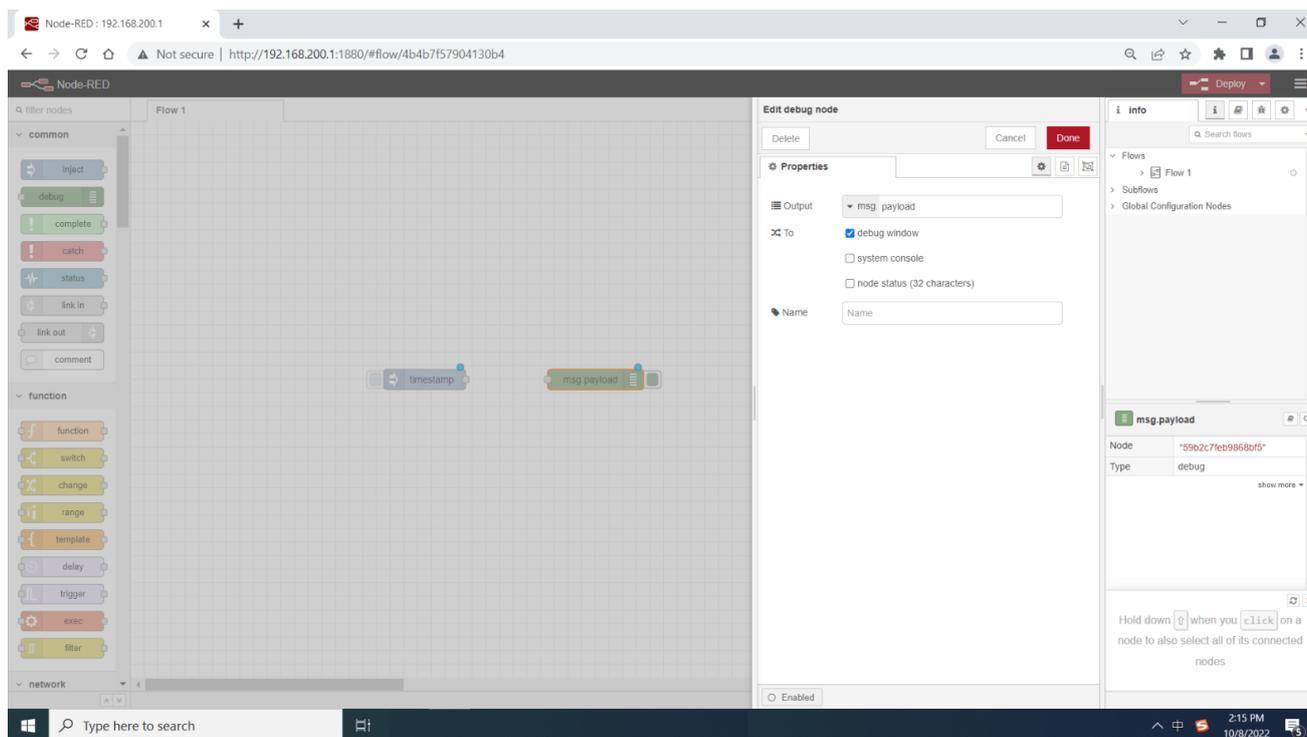
也可以设置节点是否有效，数据流是否重复发送及发送周期等参数。



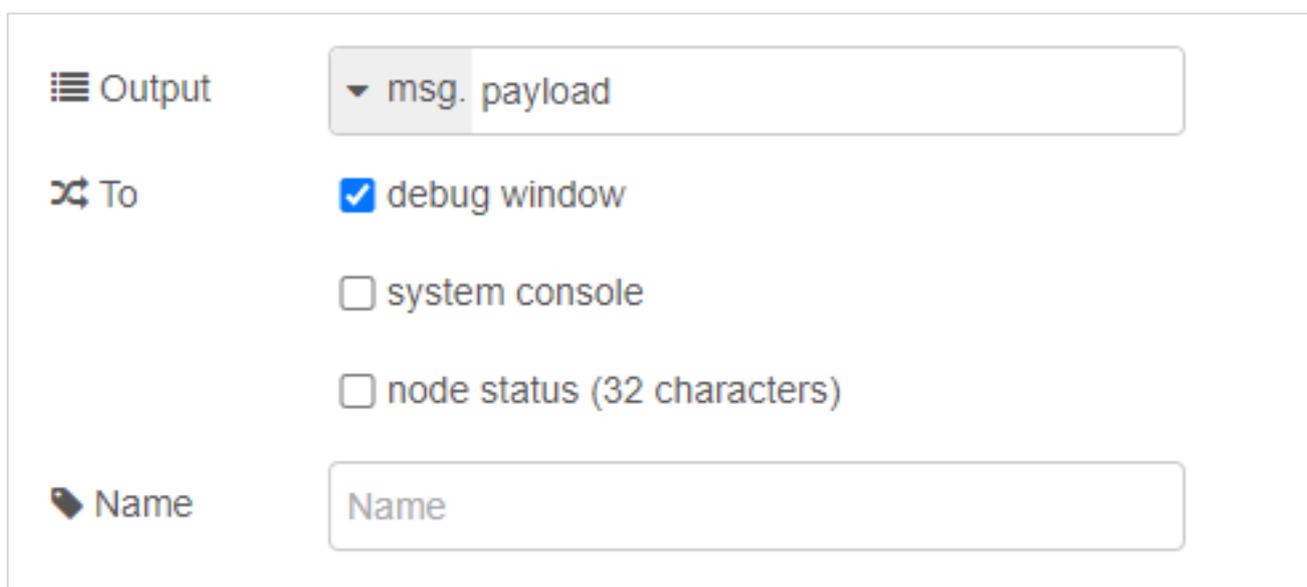
debug（调试）节点： 

通过 Debug 节点，可以在调试窗口中打印出消息流的信息，利用该节点可以十分方便的进行程序调试。

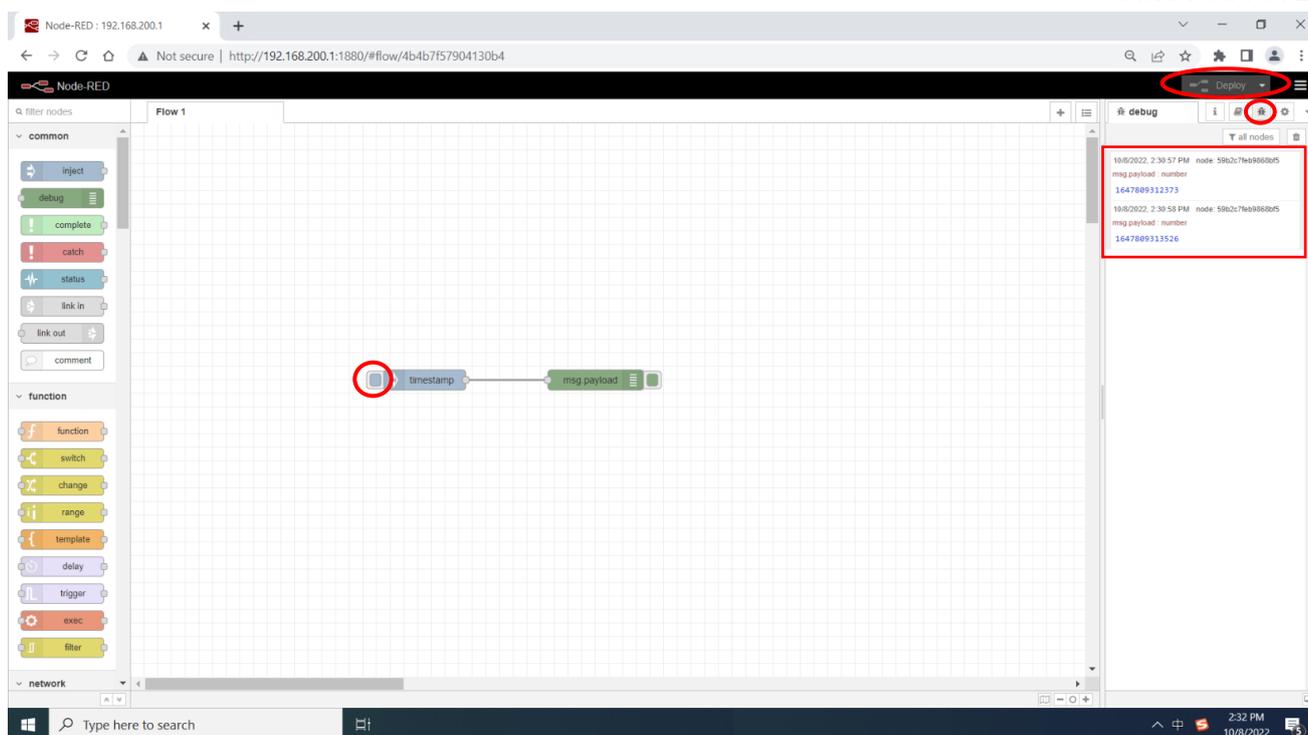
拖动 Debug 节点到工作区域中，双击 Debug 节点，就会弹出节点的设置界面。



在设置界面中，可以设置需要打印的消息流的属性（默认值为 payload），打印信息在哪个窗口显示以及设置节点的名称。



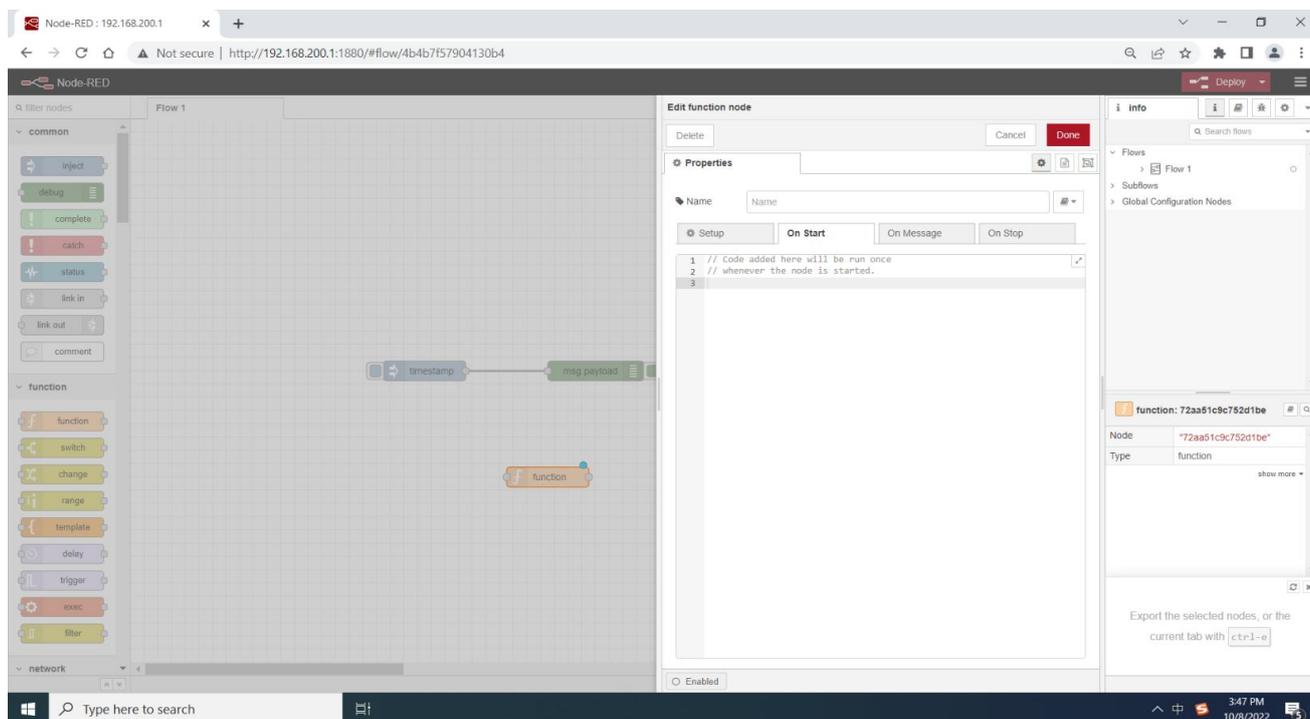
连接 inject 和 debug 节点，就可以完成一个最简单的闭合消息流，点击右上角的部署按钮完成部署。再次点击时间戳节点左侧的触发按钮，在信息显示窗口中点击调试信息按钮，就可以在调试窗口中看到打印出的消息流的信息。



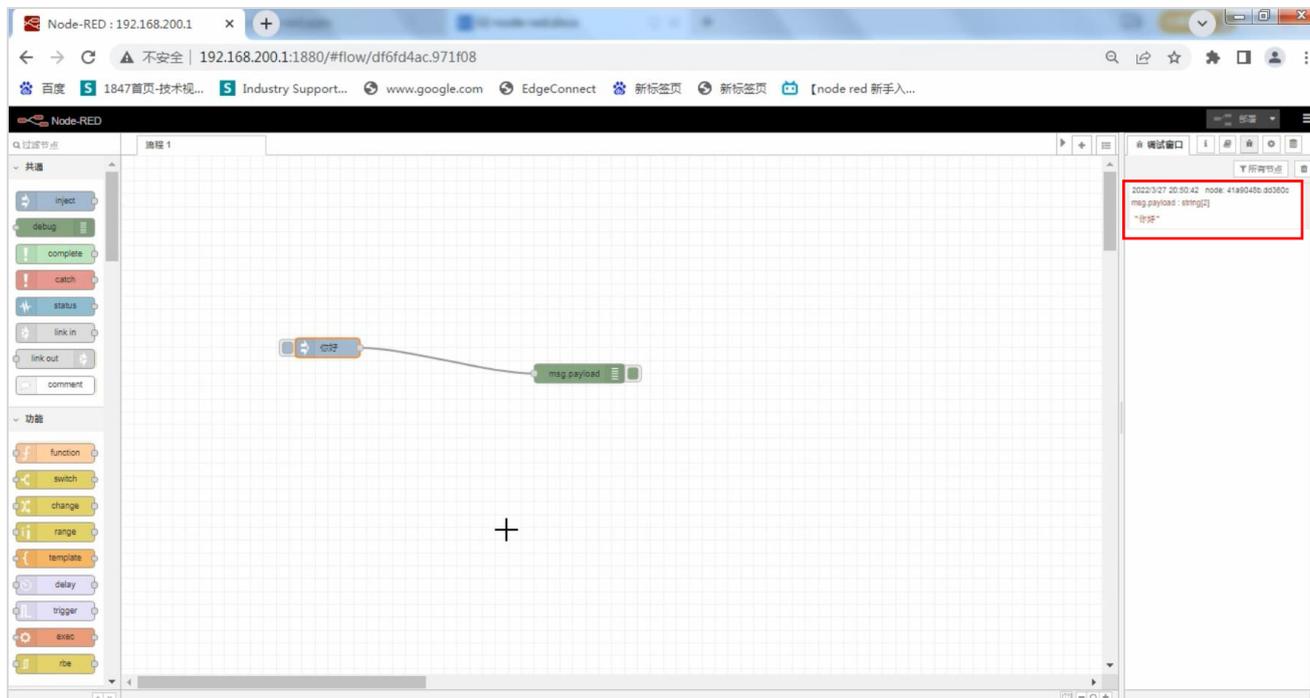
function (功能) 节点： 

通过 function 节点，用户可以在节点中编辑 JavaScript 代码，可以对流入节点的消息流执行各种运算，最后可以将运算后的结果返回给消息，以继续消息流。

拖动 function 节点到工作区域中，双击 function 节点，就会弹出节点的设置界面。在设置界面中，用户可以设置节点的名称，**编辑**节点在不同的条件下的 JavaScript 程序代码。



双击 inject 节点，将消息流的内容选择为文字列，输入“你好”后点击完成后部署并触发 inject 节点，可以看到输出值为“你好”。

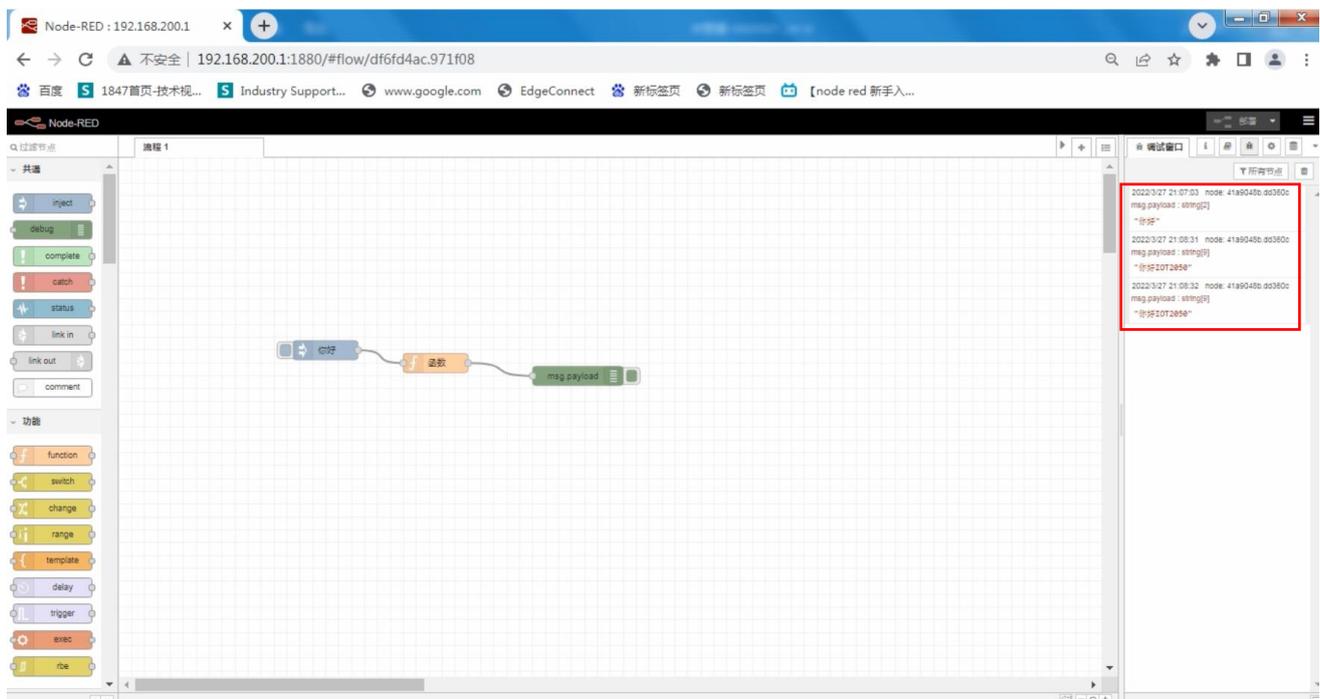


依次连接 inject，function 和 debug 节点，在 function 节点中输入如下所示 JavaScript 代码，点击右上角的部署按钮完成部署。

```
Setup      On Start      On Message      On Stop

1  var str=msg.payload+"IOT2050";
2  msg.payload=str ;
3  return msg;
```

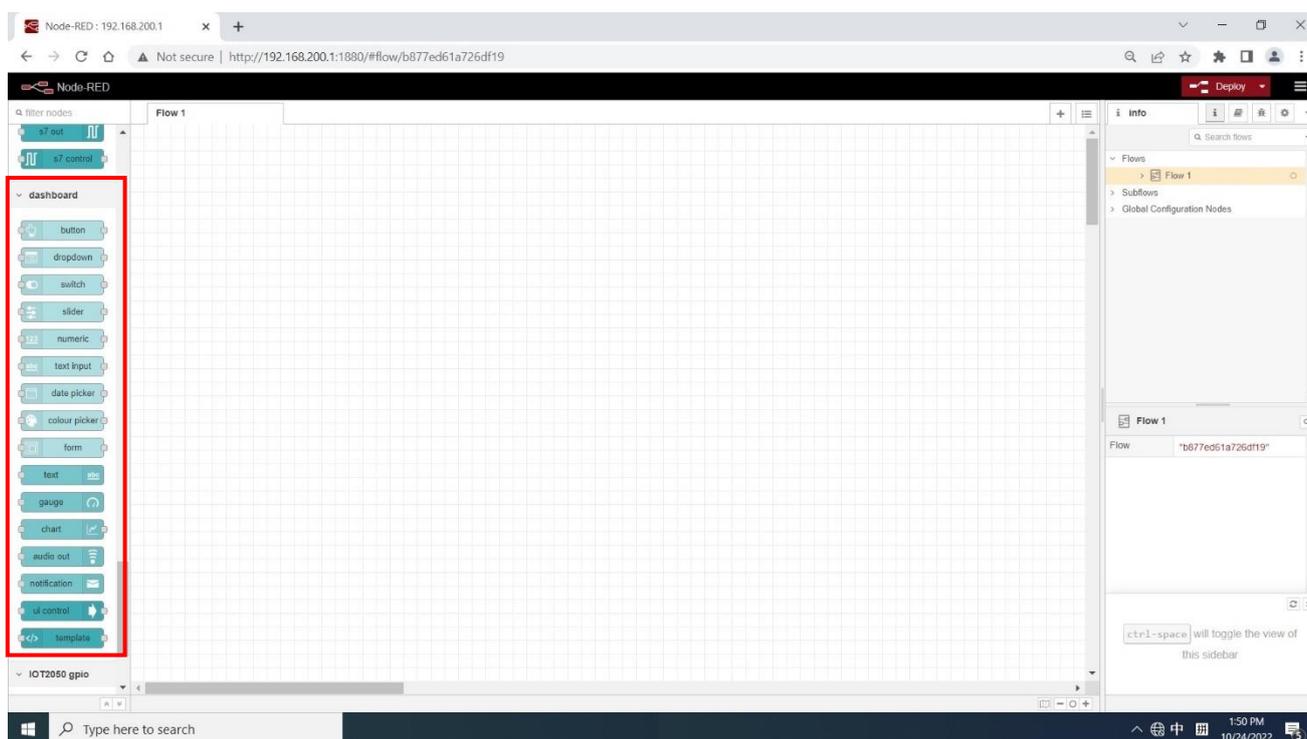
再次点击时间戳节点左侧的触发按钮，可以在调试窗口中看到流经function 节点后输出的消息信息“你好 IOT2050”。



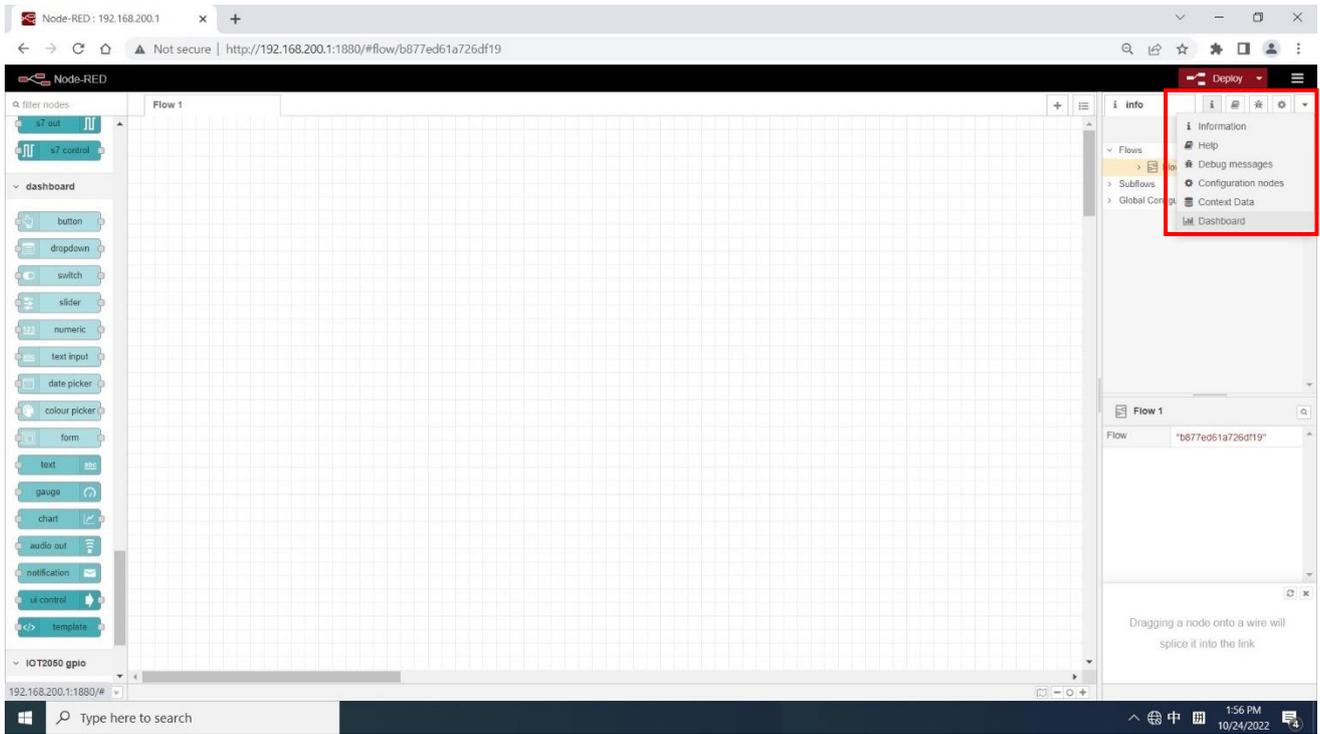
4.4 dashboard 节点

Node-red 应用中除了带有对数据进行分析，运算和处理的功能节点外，还带有能够制作 UI 界面的节点组，其中最常用的就是 dashboard 节点组。SIMATIC IOT2050 镜像中所带的 Node-red 应用中，已经提前安装好了 dashboard 节点组，用户可直接拖拽出相应功能的节点制作满足自身要求的应用界面。

向下拉动节点区域的滚动条至 dashboard 节点区域。再该区域内，可以看到很多我们在 UI 页面中经常使用会使用到的节点，例如图表，仪表盘，选项框等节点。

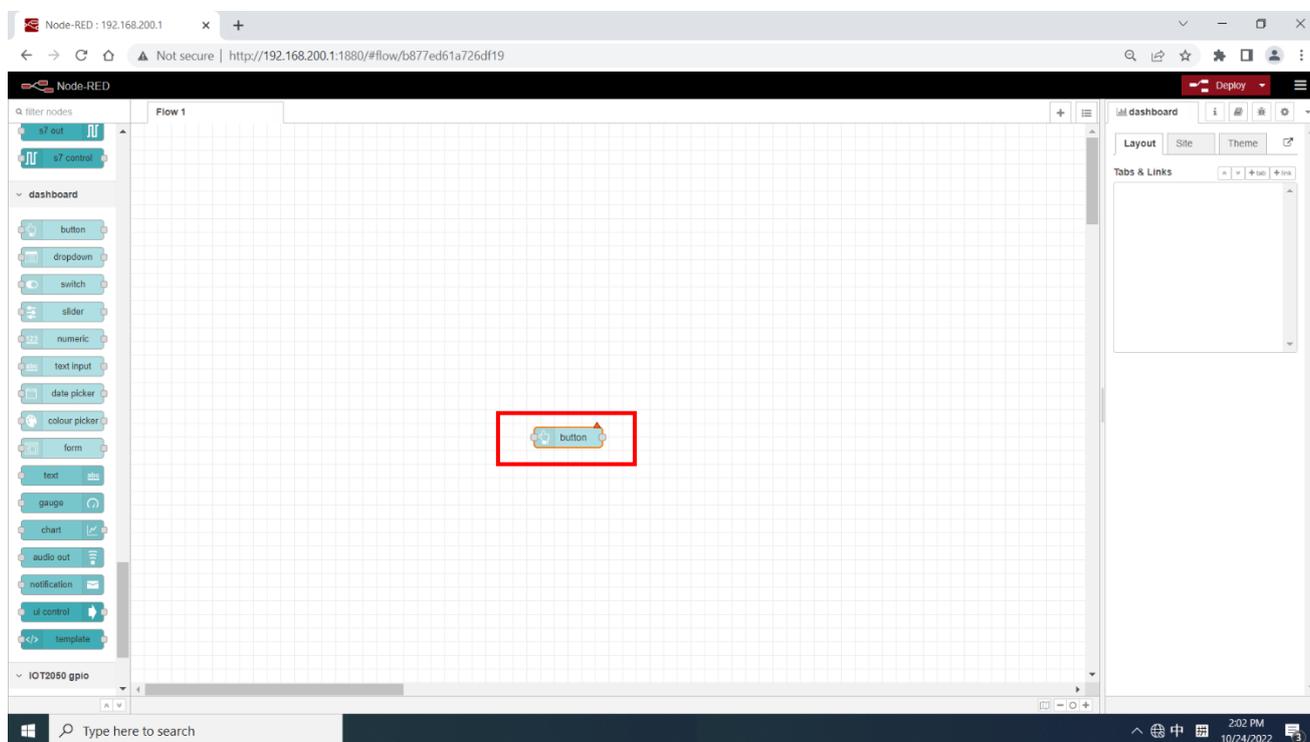


在最右侧的信息显示区，可以选择显示 dashboard 的内容，在这里可以设置 dashboard 中的节点布局，主体，色彩方案等内容。



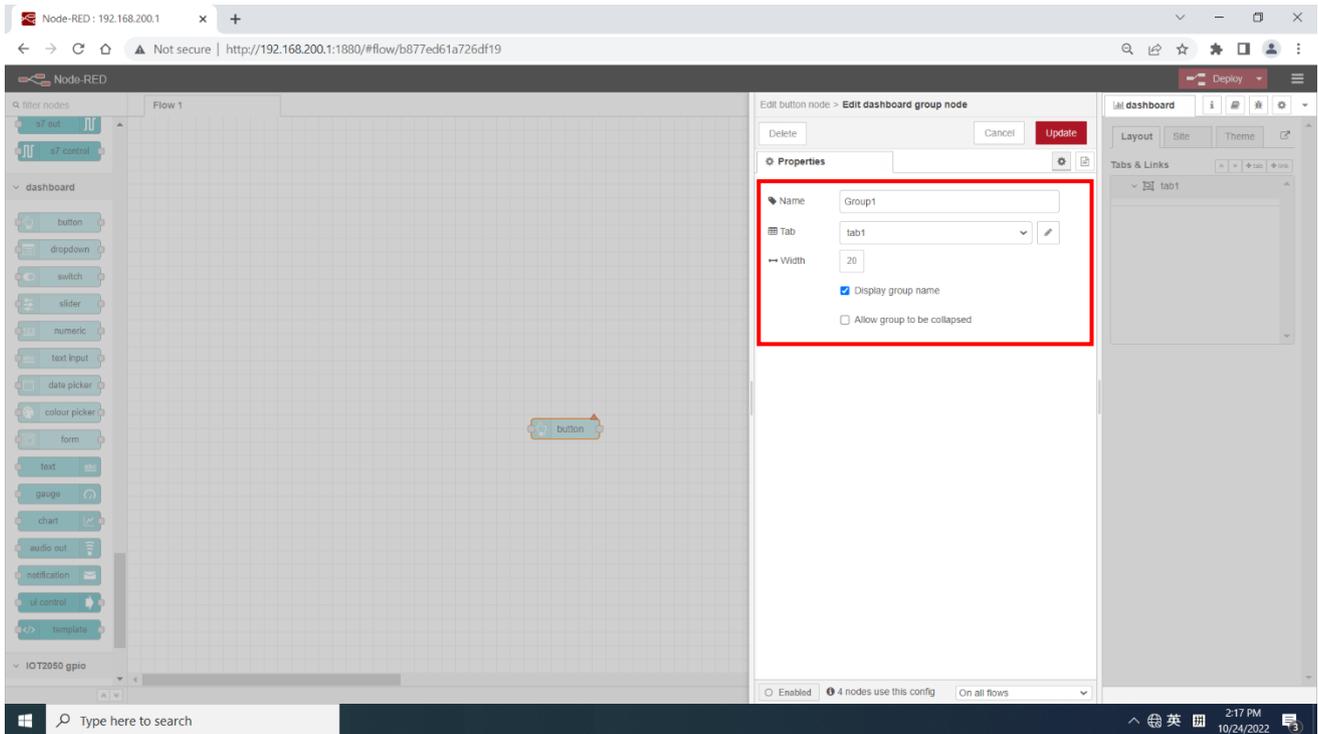
4.4.1 button 节点应用

拖拽一个 button 节点到工作区中，此时该节点上会出现一个红色的三角标志，该标志表明节点缺少相关的参数，需要进行设置。

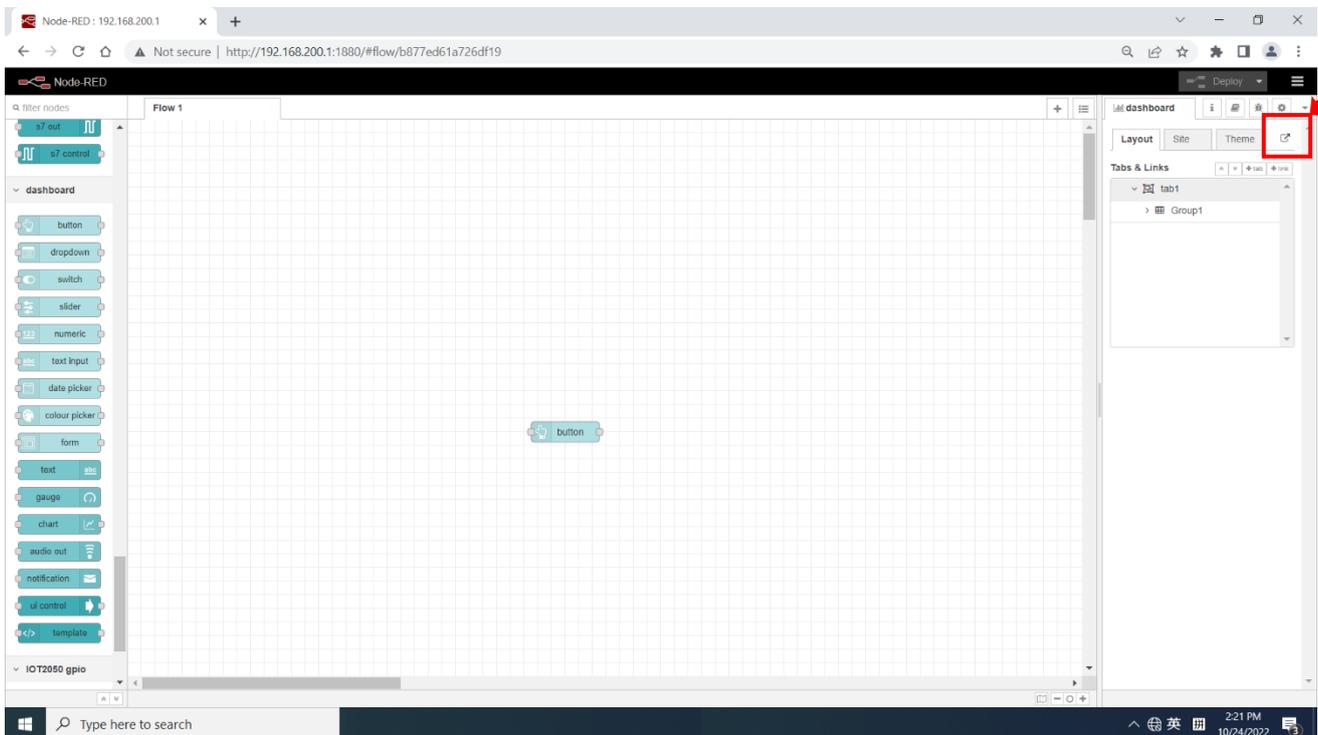


双击 button 节点，在弹出的属性窗口中可以根据自身需要更改 button 节点的对应属性。所有节点都需要被分配在相应的 tab 页面和 group 分组当中的，也可以根据节点的应用分类，将节点设置到不同的 group 分组当中。

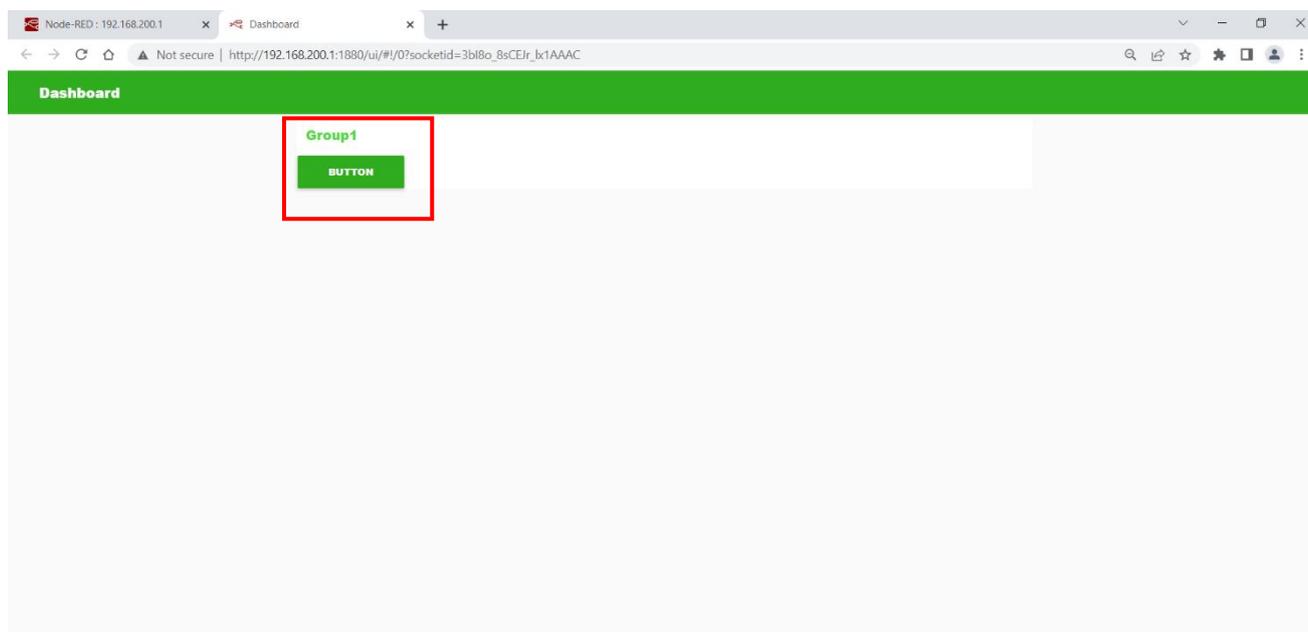
双击 button 节点，在弹出的属性窗口中可以根据自身需要更改 button 节点的对应属性。在 group 下拉列表框中选择“添加新的 ui_group 节点”，点击右侧的按钮，进入 group 属性页面，在这里可以更改 group 的名称为 group1。在 tab 列表框中选择添加新的 ui_tab 节点，点击右侧按钮进入 tab 属性页，name 栏中输入 tab1。



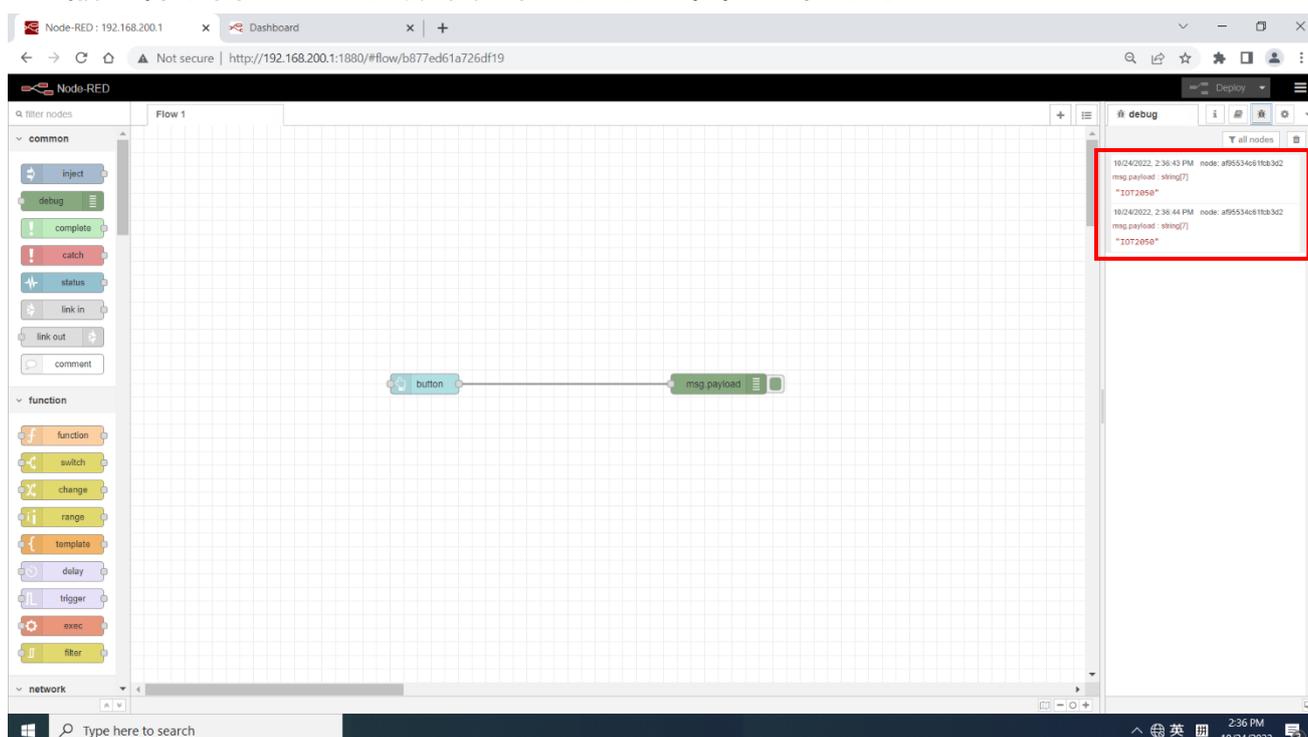
依次点击右上角的红色添加按钮和完成按钮，完成按钮的部署。再次点击右上角 dashboard 的 UI 页面链接图标



此时就可以在 UI 页面中看到我们部署在 tab1 窗口下，group1 分组中的 button 节点。

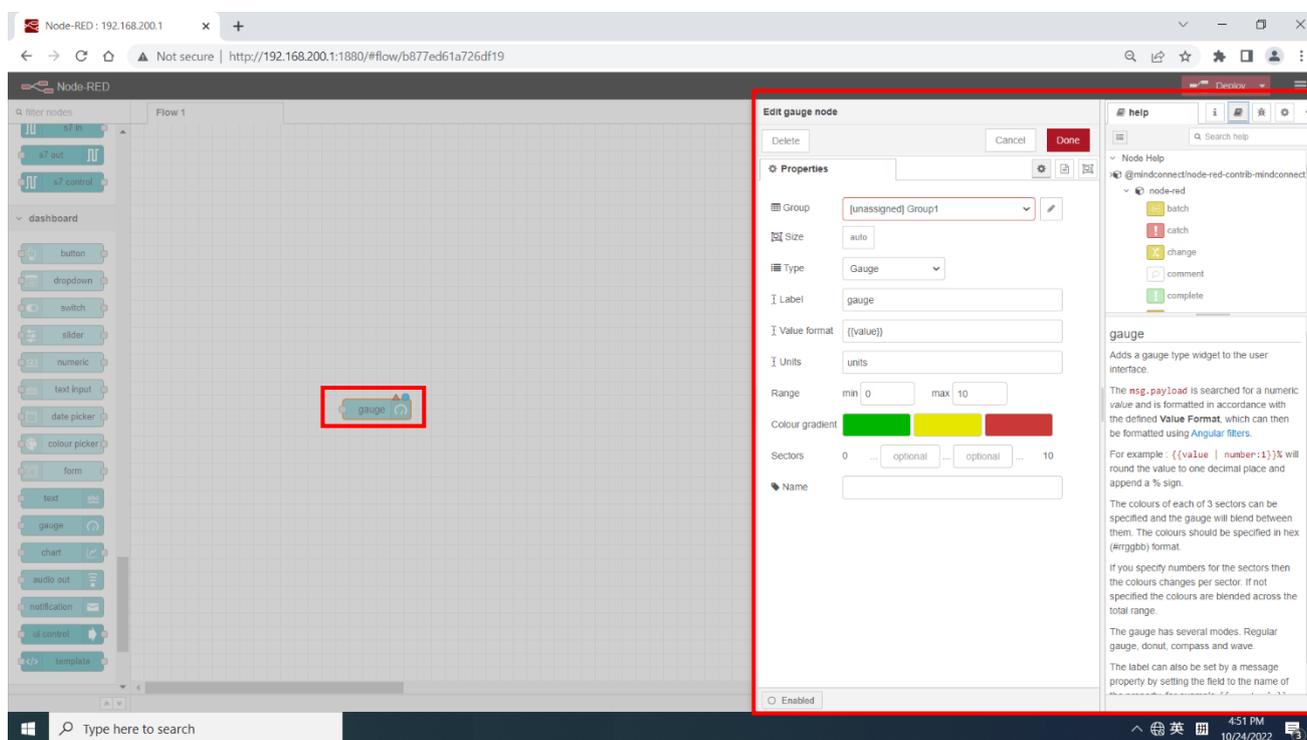


返回应用页面，将 debug 节点拖拽到工作区域中，连接 button 节点和 debug 节点，双击 button 节点，在 payload 下拉列表框中选择输出类型为文字列，内容输入“IOT2050”，点击完成后部署应用。切换至 UI 页面点击按钮后，返回调试窗口，此时可以看到设置的按钮输出内容，已经被打印到调试窗口中了。



4.4.2 gauge 节点应用

拖拽一个 gauge 节点到工作区域，双击 gauge 节点，可以在弹出的属性窗口中设置 gauge 节点的参数。从右侧的帮助栏中也可以查看 gauge 节点对属性相关说明。



可在属性设置窗口对节点进行设置：

Size 选项：可以设置 gauge 的大小，gauge 的最大宽度由所设置的 group 的宽度所决定。

type 选项：可以选择不同 gauge 的样式类型，用户可以根据需要选择使用合适的 gauge 样式。

Label 选项：设定 gauge 节点的标签，更改后，UI 页面中节点的标签会被更改为设置的内容。在没有设置 name 选项的情况下，工作区域中节点标签的名称也会被更改。

Value format : gauge 节点数据区域所显示的数值, 默认情况下显示 gauge 节点接收到的数值, 用户可以将其删除或更改为其他内容, 使用时该选项一般会保持默认。

Range, color gradient 和 Sectors 选项 : 只有在节点的样式类型被选择为 Gauge 和 Dount 时才会出现。三个选项配合使用可以在节点接收到不同范围的数值时显示不同的色彩效果。

Range 选项 : 设置 gauge 节点中所能显示数值的最大最小值范围, 当此处被设置时, Sectors 选项中的起始和终止数值也会发生相应变化。

sectors 选项 : 将 gauge 节点的显示数值进行分段, 最多可设置为 3 个不同的数值区域。

color gradient 选项 : 为不同的数值区域设置不同的显示颜色。

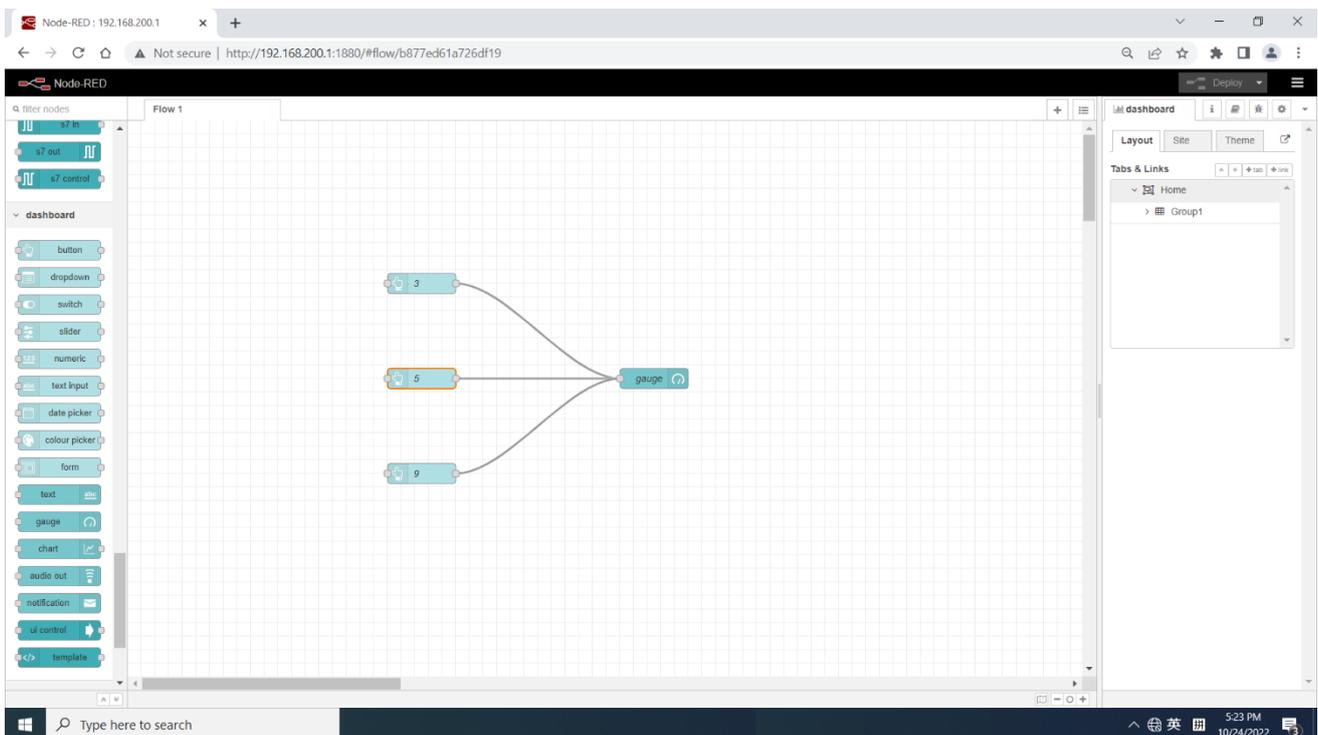
Name 选项 : 设置 gauge 节点在工作区域中所显示的名称, 可以看到, 当更改 name 选项的设置后, 工作区域中 gauge 节点标签的名称也会随之变换。

最后, 还可以通过属性页中最下面的按钮来选择节点是否有效, 如果选择无效, 则整个节点的部分显示为虚线, UI 界面中也不会再出现设置为无效的 gauge 节点的内容

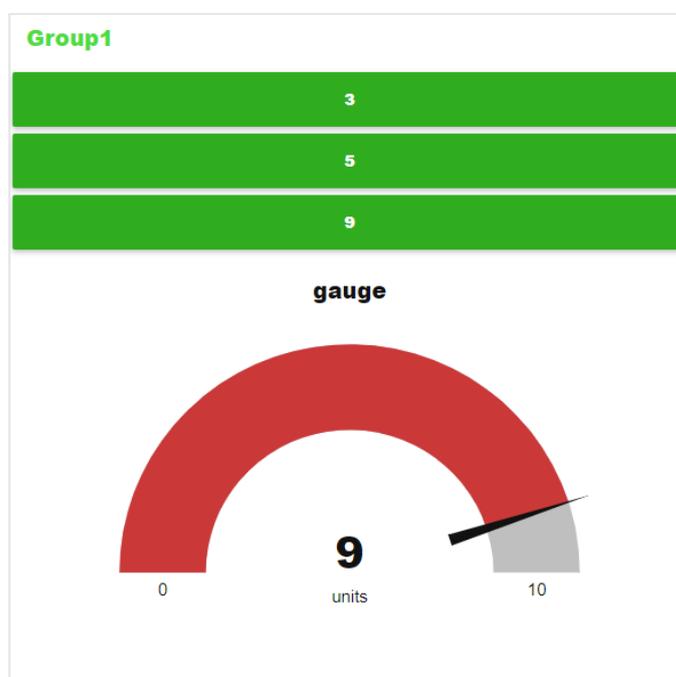
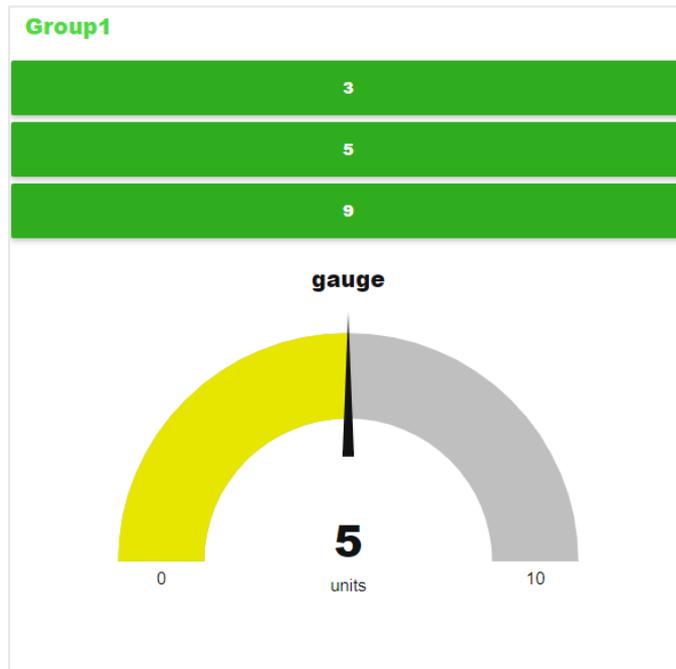
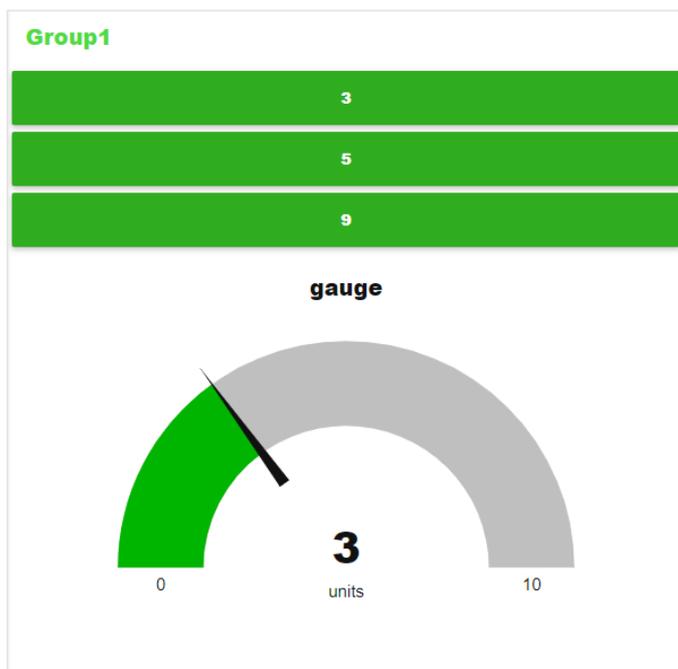
以下图中所示为例对各项参数进行设置后设置。



分别将 3 个按钮和一个 gauge 节点拖动至工作区，双击第一个按钮，设置分组，将 label 选项和 payload 选项都设置为数值 3，使用同样的方法，将另外两个按钮的 label 和 payload 选项分别设置成 5 和 9。将三个按钮节点分别与 gauge 节点进行连接后，点击部署按钮部署设置。



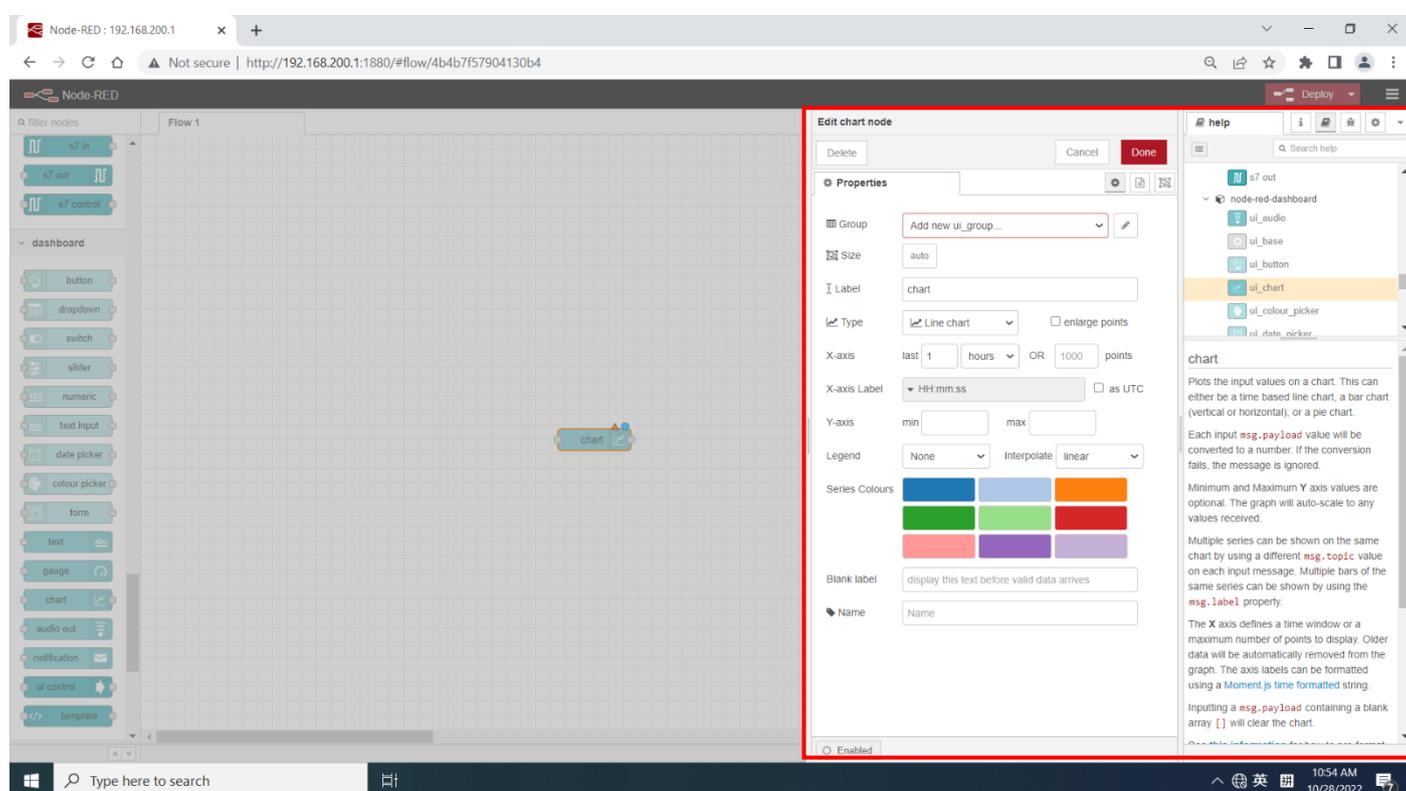
部署成功，进入 UI 页面后，按下发出相应数值的按钮后，gauge 就会显示对应的数值，gauge 中指针区域的颜色也会显示为该数值所在区域对应的颜色



4.4.3 chart 节点用

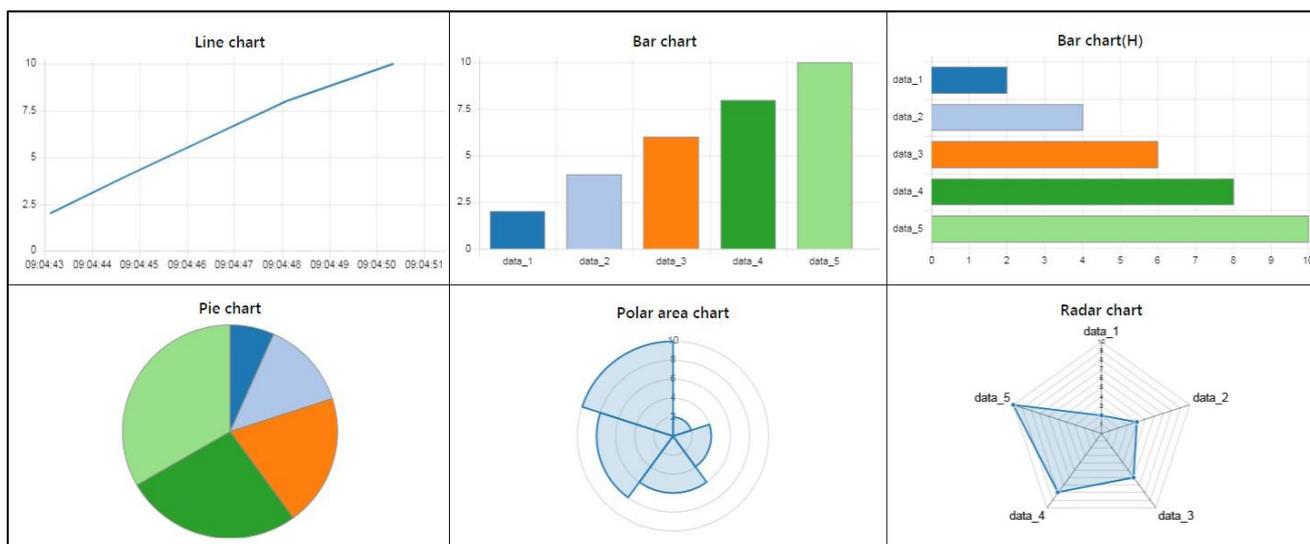
在一般的 UI 页面中，为了更好的展示效果和um提高可读性，页面中通常会带有线性图，柱状图和饼图等等。这些表形效果，都可以通过 chart 节点来实现。

拖拽一个图表控件到工作区域，双击图表控件，可以在对应栏目中查看图表控件的属性，说明以及帮助文档等等

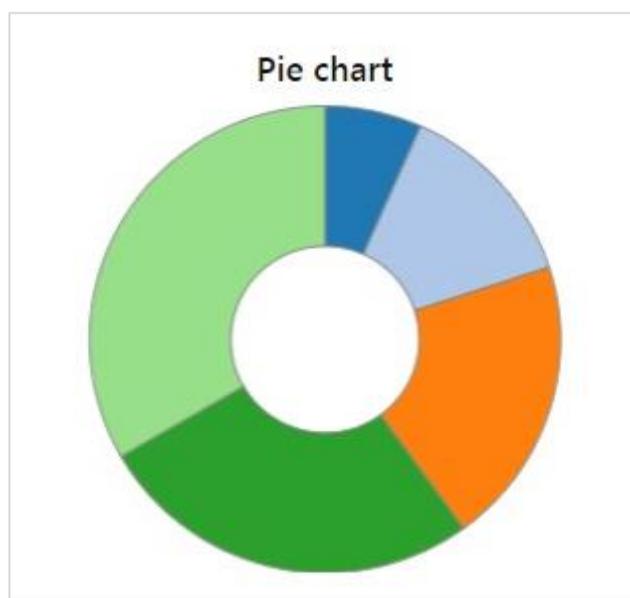


在属性窗口中，可以对图表的分组，大小，类别，颜色等属性进行设置，一些标准属性的设置，可以参考上一章节，一下只对图表独有的一些属性选项进行相关说明：

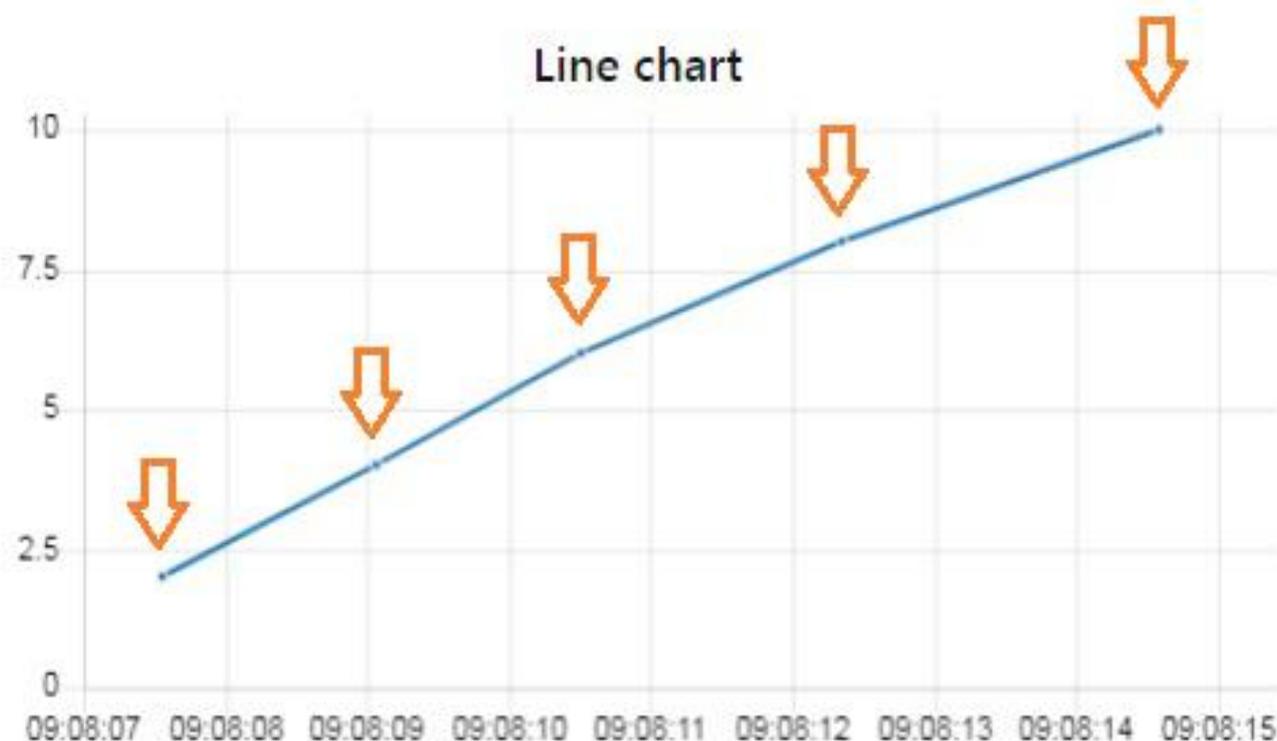
Type 选项：可以选择不同的图表类型，各种不同的选项对应的图表样式如下图所示。



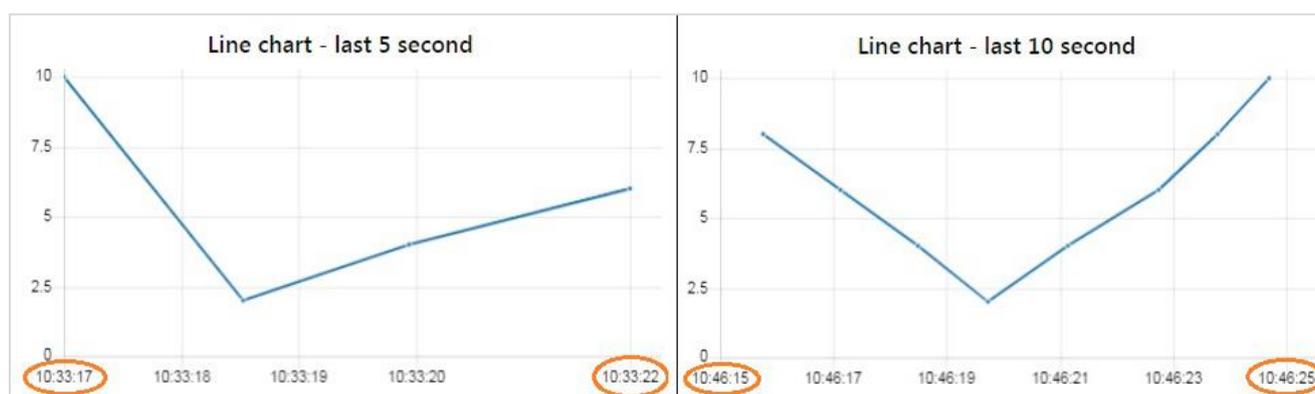
当选择不同图表类型时，后面的可更改的属性选项也会发生相应变化。例如通过设置 Pie chart 的 Cutout 属性值，可以将饼图改变为相应的环形图



当选择 Line chart 图表类型时，可通过勾选 enlarge points 选项来突出显示曲线中各节点的效果。

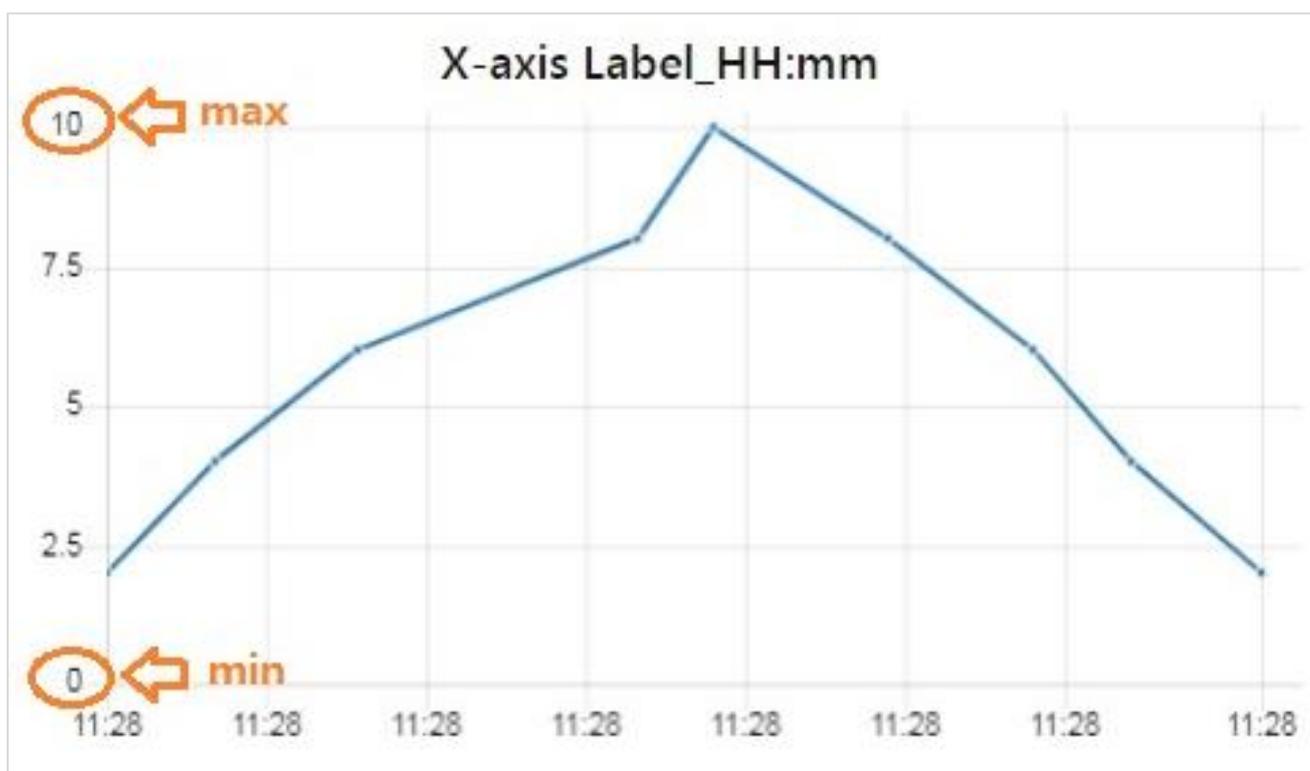


X-axis 选项：可以设置数据点的保留时长或保留个数，在图例中，当分别将数值设置为 5 秒和 10 秒时，对应的图表就会显示最后 5 秒和 10 秒时间的曲线。



X-axis Label 选项：可以设置 X 轴标签的格式，用户可通过该选项控制 X 轴标签，如图例所示，当选择以分钟为标签时，即使在一分钟内有多数据点，数据点对应的 X 轴标签也不会发生变化。

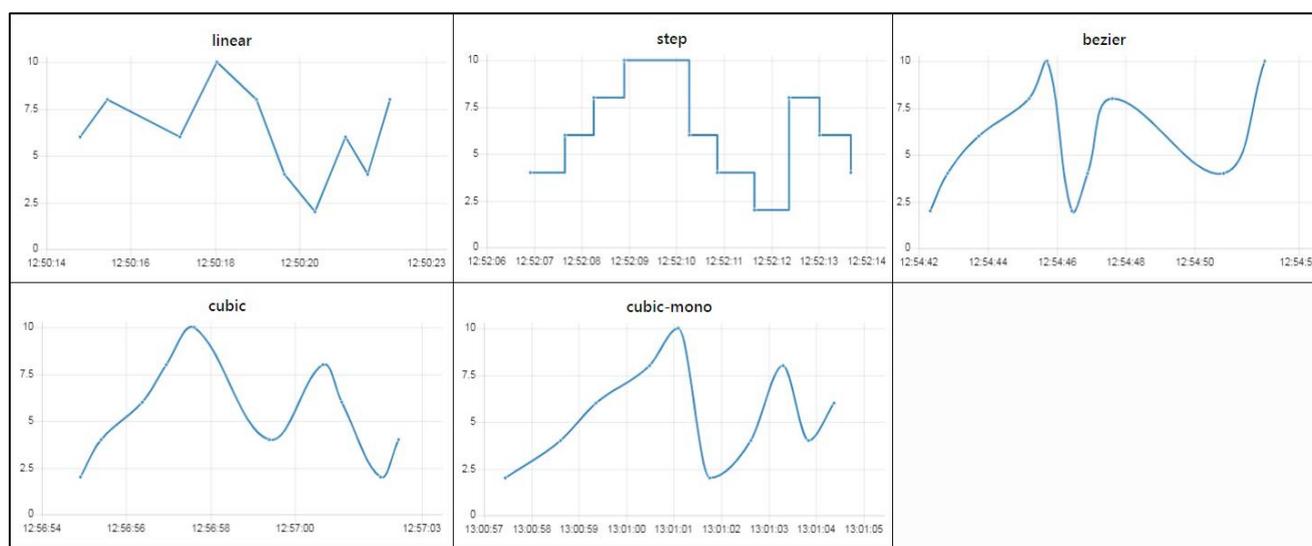
Y-axis 选项：通过 min 和 max 选项可以设置图表 Y 轴的最大和最小值。



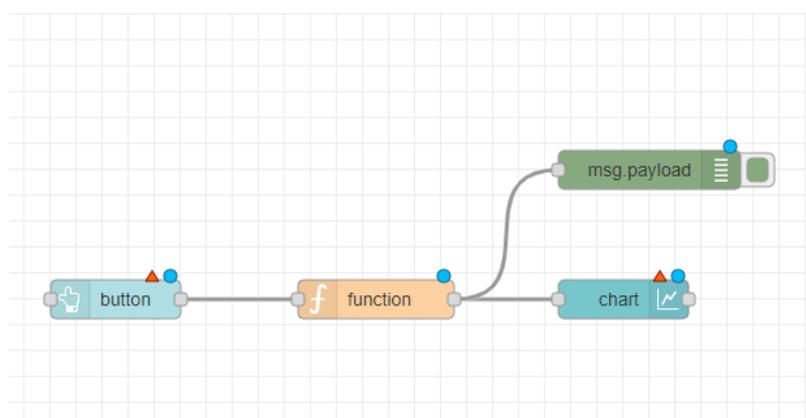
Legend 选项：可以设置是否显示数据标签，如图例中所示，当 Legend 选项设置为 show 时，图表中会显示数据标签



Interpolate 选项：可以设置图表中数据点连接时的过度的方式，选择不同选项时图表的效果如图例中所显示。



分别将一个按钮控件，函数控件，图表控件和 debug 控件拖动至工作区并如图中所示依次连接。



设置按钮属性中的 Group 选项和 payload 选项，将 payload 选项设置为布尔值 true。

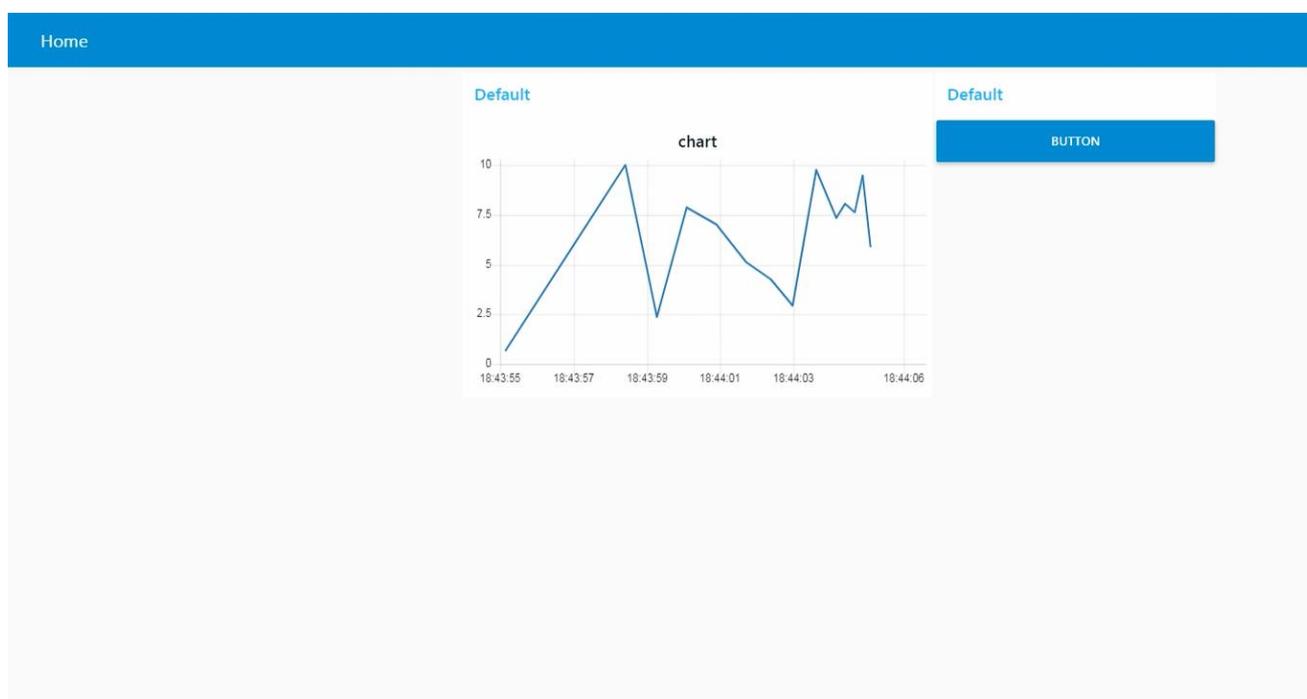
双击函数控件，在弹出的属性框中选中函数库，按照图例所示输入对应函数内容。

```
Setup On Start On Message On Stop
1 var value1 ;
2 var topic1 ;
3
4 value1=Math.random()*10 ;
5 topic1="data" ;
6
7 msg.payload=value1 ;
8 msg.topic = topic1 ;
9
10 return msg;
```

通过该函数使每次按钮被按下后，都会向图表发送一个 0 至 10 的随机数来模拟从设备采集的数据。

双击图表函数，设定 Group 选项为与按钮相同的 Group，设定 Y-axis 属性，min 值为 0，max 值为 10 后点击完成并部署。

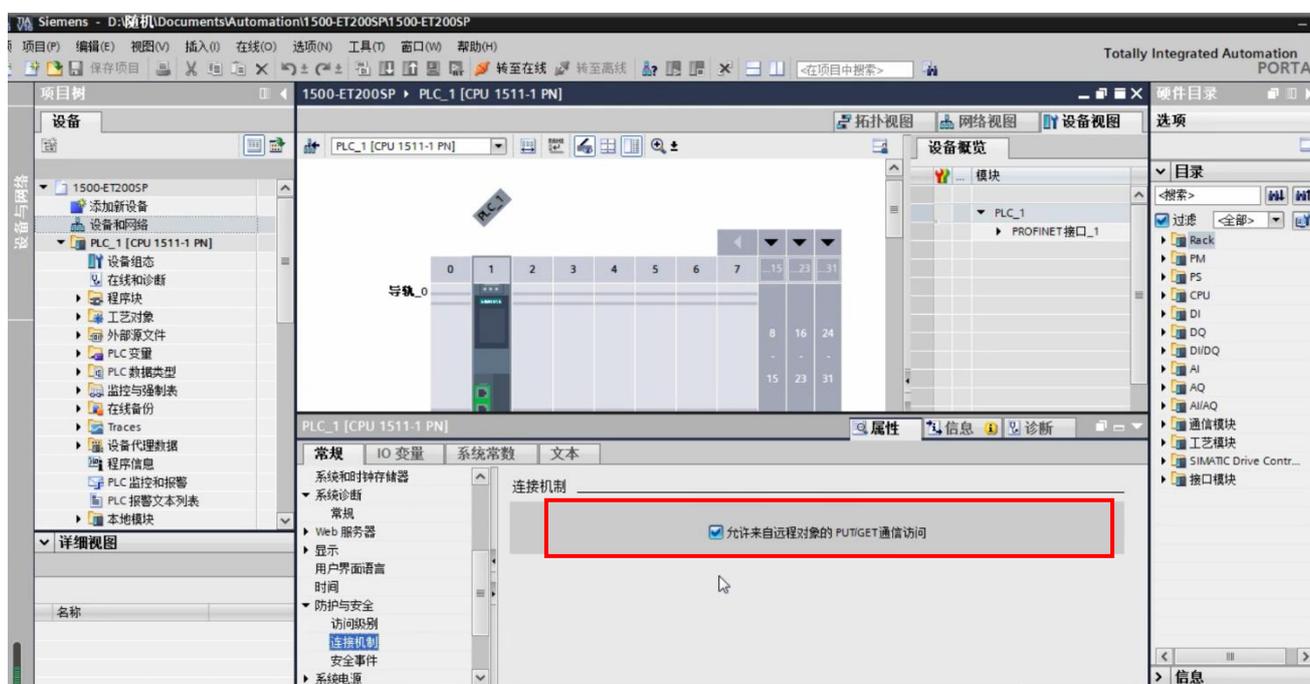
跳转至 UI 页面后，每次按下按钮后，图表中就会根据接收到的数据持续的绘制曲线。



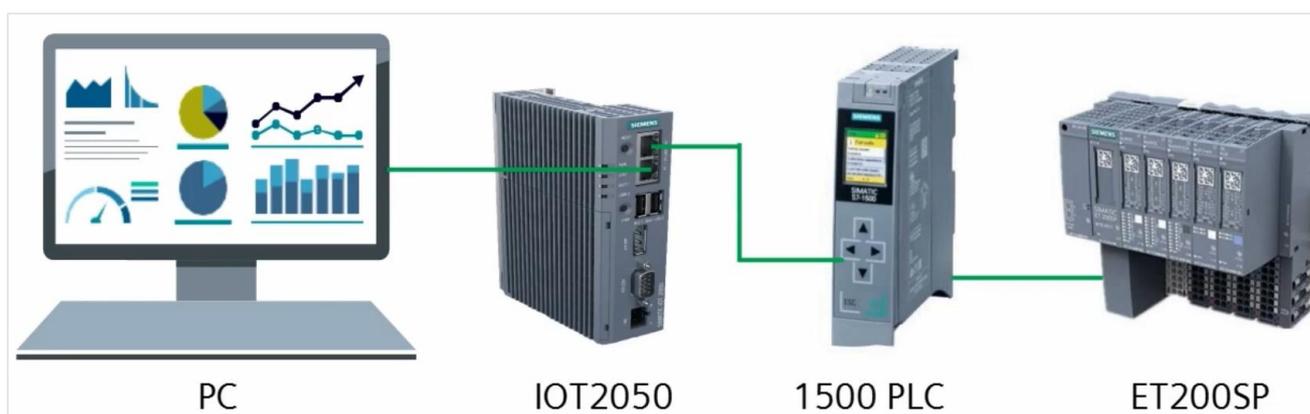
4.5 通过 S7 节点读取 PLC 数据

SIMATIC IOT2050 应用镜像中集成的 Node-red 应用，已经预安装了 S7 节点，经过简单的参数设置，可以通过 S7 节点读取 PLC 中的数据。

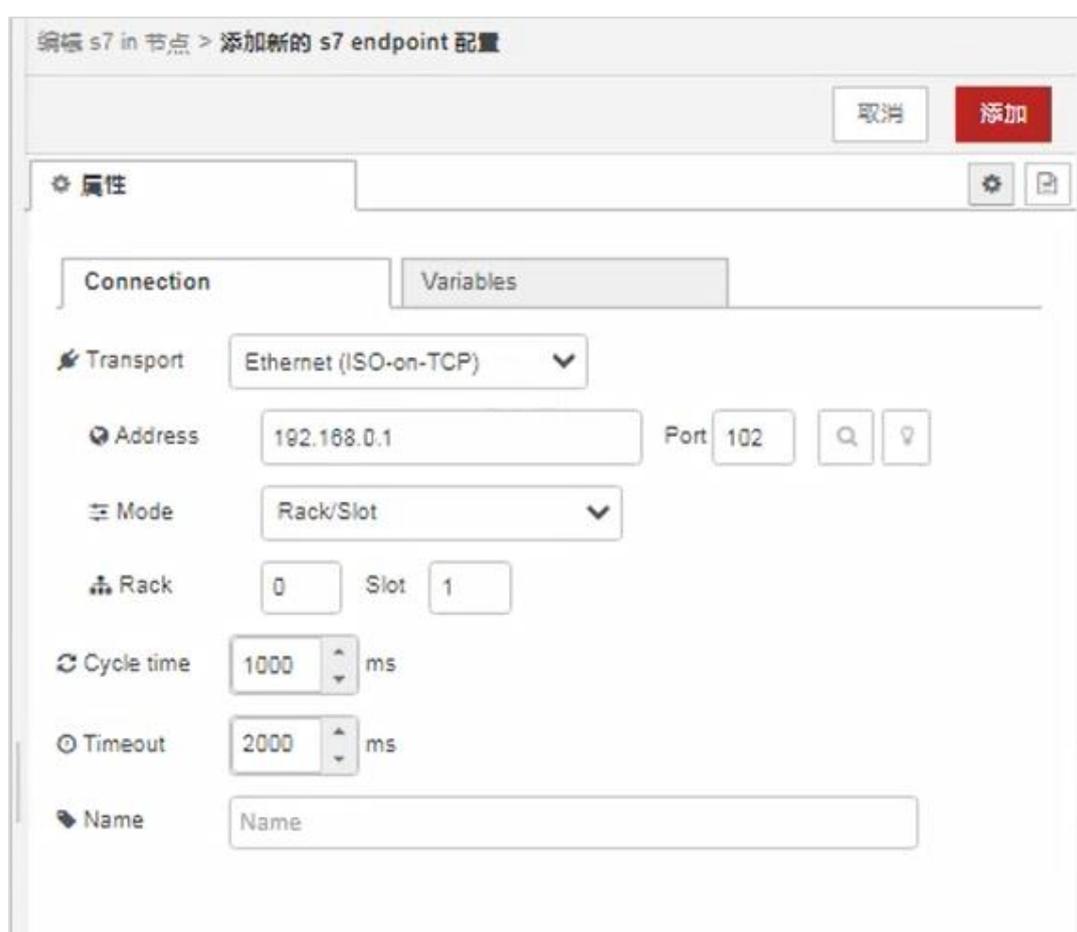
在通过 IOT2050 对 PLC 的数据进行采集前，需要先使用博图软件对 PLC 和 ET200SP 进行正确的组态，保证设备正常运行。为了能让 IOT2050 访问并读取到 PLC 的数据，组态时需要开启视频中所示 PLC 的“允许来自远程对象的 put/get 通信访问”功能。



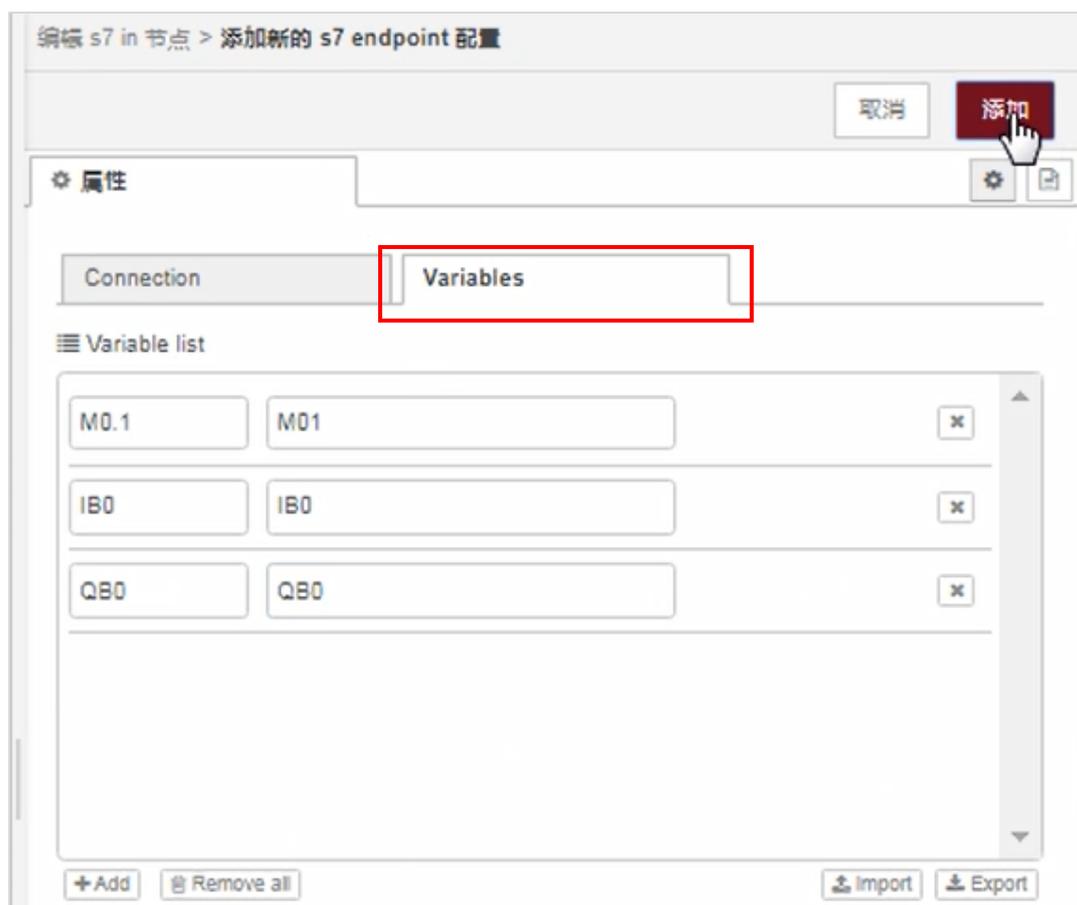
实际设备连接如下图所示：



设备连接和组态完成后，启动 IOT2050 进入 node-red 应用界面。在节点区域中找到 plc 节点组，将 S7 in 节点拖拽至工作区域中，双击节点进行参数设置。点击 plc 选项后端的铅笔符号设置连接属性。在 Address 处输入 PLC 地址，可在 Cycle time 处设置发送周期。此处的参数，应该和组态时的设置保持一致。通常情况下，只需要设置 PLC 的 IP 地址和更改 PLC 的槽号，其他设置保持默认即可。



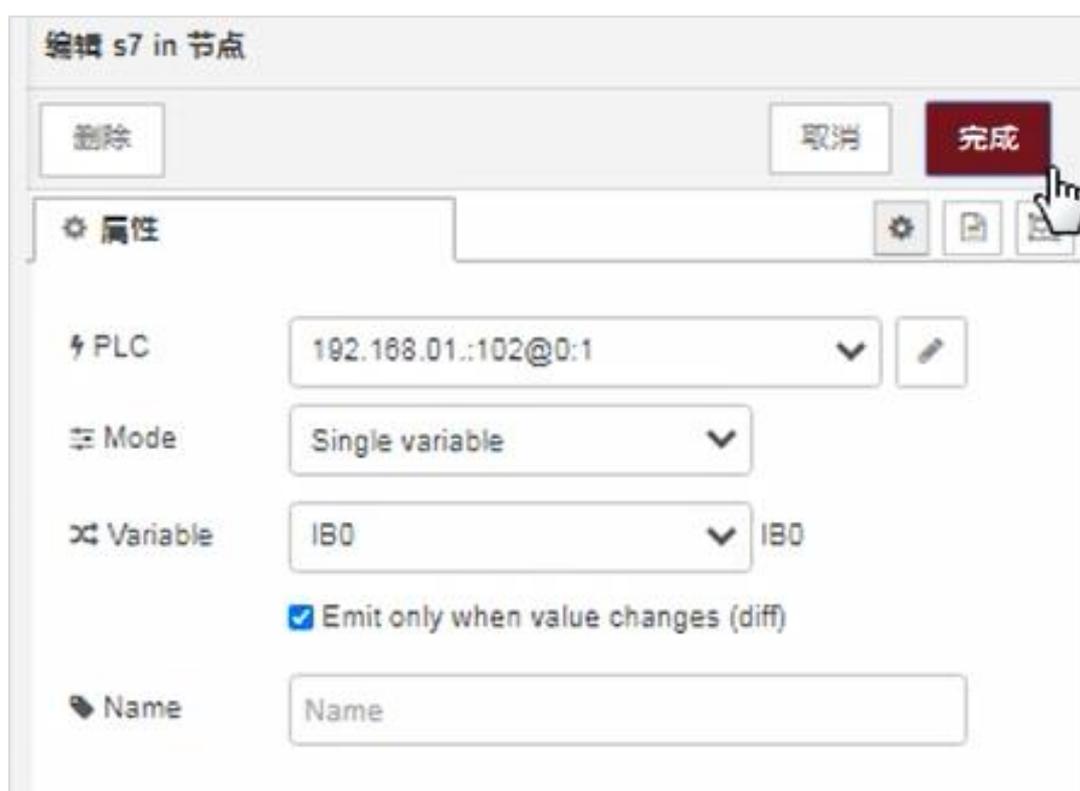
点击变量按钮将设置界面切换至变量列表窗口，在这里可以添加我们想要采集的变量。可以是输入，输出，寄存器或 DB 块等数据。



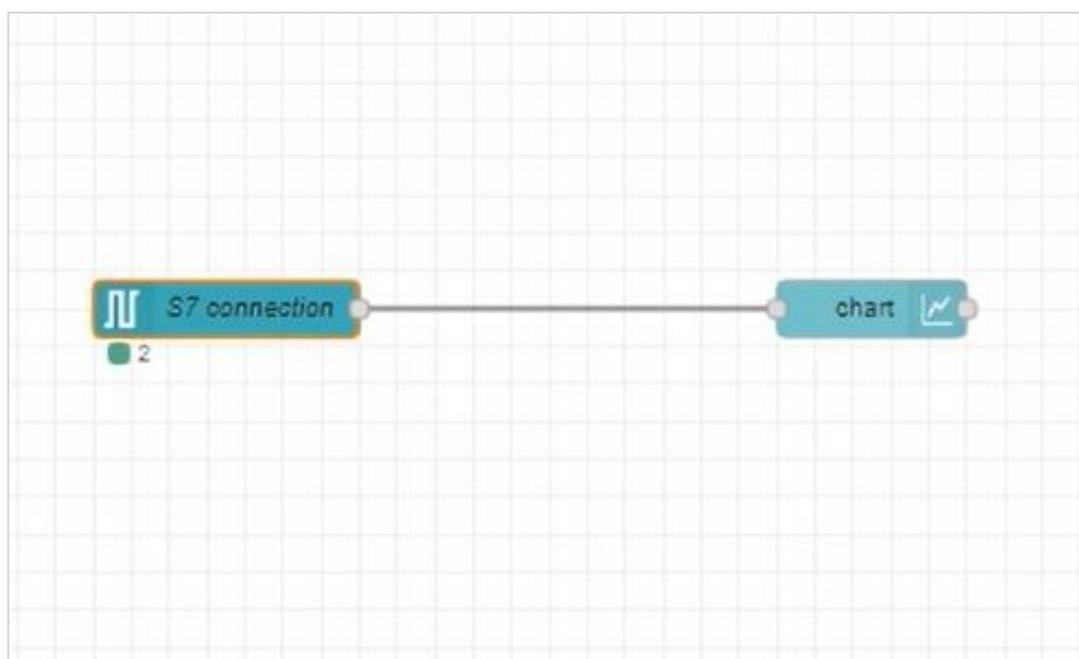
不同变量在 node-red 中的寻址方式，可以下面表格。

	数据区域	数据类型	PLC 寻址方式	Node-red 寻址方式
PLC 变量	I	BOOL	Ix.y	Ix.y
		BYTE,CHAR	IBx	IBx
		WORD,INT	IWx	IWx
		DWORD,DINT, REAL	IDx	IDx
	Q	BOOL	Qx.y	Qx.y
		BYTE,CHAR	QBx	QBx
		WORD,INT	QWx	QWx
		DWORD,DINT, REAL	QDx	QDx
	M	BOOL	Mx.y	Mx.y
		BYTE,CHAR	MBx	MBx
		WORD,INT	MWx	MWx
		DWORD,DINT, ,REAL	MDx	MDx
T	TIMER (only support S7-300/400)	Tn		
C	COUNTER (only support S7-300/400)	Cn		
用户数据块	DB	REAL		DBn,REALx
		DWORD		DBn,DWORDx
		DINT		DBn,DINTx
		INT		DBn,INTx
		WORD		DBn,WORDx
		BYTE or B		DBn,Bx
		STRING or S		DBn,S (x - 1).length
		CHAR or C		DBn,Cx

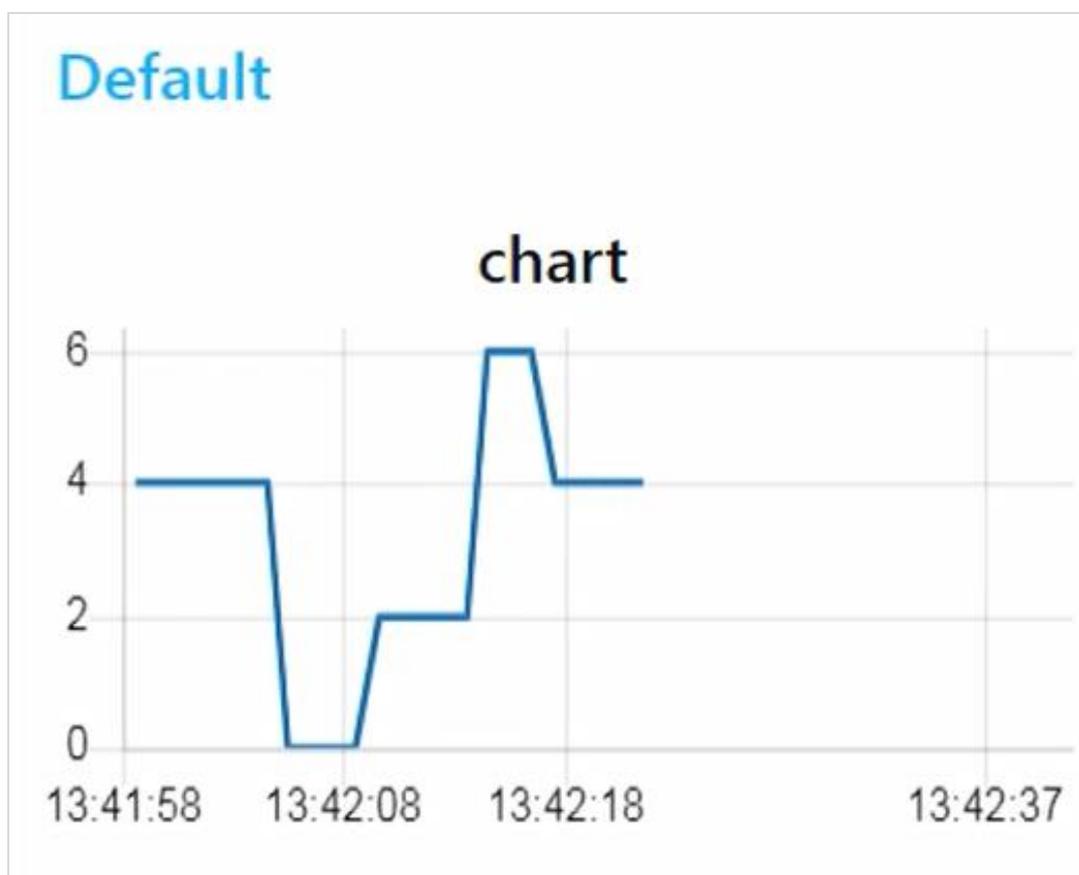
连接和变量设置好后，点击右上角的添加按钮返回属性界面。在 mode 选项框中可以设置读取单个还是全部变量，本示例中我们选择单个变量。在变量选项框中，可以选择采集哪个一变量。Name 选项框可以设置节点在页面中显示的名称。设置好后点击完成退出设置界面。



从节点区域中拖动一个图表节点到工作区，双击节点设置分组及其他属性，设置后点击完成。连接 S7-in 节点和图表节点，点击部署将设置好的程序下载到 IOT2050 中。可以看到 S7-in 节点的左下角会显示连接的状态，连接完成后，可以显示当前采集回来的变量的数值。



跳转至 UI 页面，可以看到从 PLC 采集到的数据已经自动被绘制为图表，当 PLC 中的变量发生变化时，图表上对应的数值也会发生相应的变化。



通过浏览器进入 node-red 应用界面，在节点区域中找到 Modbus 节点，将 Modbus Read 节点拖拽至工作区域中。双击节点进入设置界面，名称项和主题项用户可根据需要自行定义。Unit-ID 为传感器的 Modbus 站号，FC 为功能码，有 4 种功能码可供选择，此处选择 FC3 读取保持寄存，Address 项为数据读取寄存器的起始地址，Quantity 为数据读取的寄存器数量，这两项需要根据传感器的实际属性进行设置。Poll Rate 为轮询周期，用户可根据实际需要选择单位设置大小。

点击下方 Server 输入框右侧的铅笔按钮，进入属性设置页面。

编辑 Modbus-Read 节点

删除 取消 完成

属性

Settings Optionals

名称 Name

主题 Topic

Unit-Id

FC FC 3: Read Holding Register

Address 32

Quantity 16

Poll Rate 1 second(s)

Delay on start

Server modbustest

名称项用户可以自行定义。Type 选项可以设置传输的类型，由于本次我们要使用 ModbusRTU 通讯，所以选择 Serial 选项。点击 Serial port 右侧的放大镜图标，可以查看到 IOT2050 的所有接口，如果使用的是 IOT2050 Basic 设备，此处选择 /dev/ttyS2 选项。如果使用的是 IOT2050 Advance 设备，则此处选择 /dev/ttyUSB0 选项。在 Serial type 选项框中，如果使用的是 IOT2050 Basic 设备，则此处选择 RTU 选项。如果使用的是 IOT2050 Advance 设备，则此处选择 RTU-BUFFERD 选项。Baud rate 选项框需要根据传感器的实际参数进行设置，此处我们设置为 9600. 剩余选项保持默认即可。点击“更新”按钮返回设置页，点击“完成”按钮完成设置。

编辑 Modbus-Read 节点 > 编辑 modbus-client 节点

删除 取消 更新

属性

名称 modbustest

Type Serial

Serial port /dev/ttyUSB0

Serial type RTU-BUFFERD

Baud rate 9600

Unit-Id 1

Timeout (ms) 1000

Reconnect on timeout

Reconnect timeout (ms) 2000

UnitId's in parallel

Log states changes

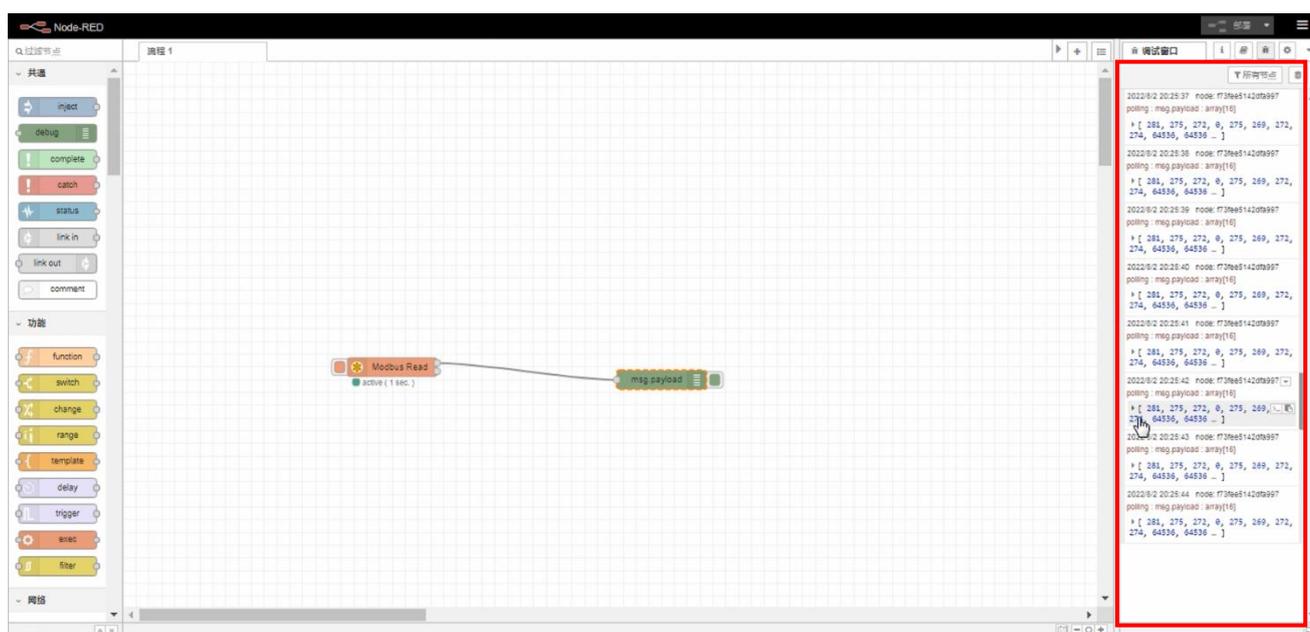
Queue Logging

Queue commands

Queue delay (ms) 1

有效 0 个节点使用此配置 所有流程

从节点列表中拖动一个 debug 节点到工作区域中，连接 Modbus read 节点右上方的输出端和 Debug 节点的输入端，点击“部署”按钮将程序部署到 IOT2050 中。在调试窗口中就可以看到 IOT2050 从传感器测采集到的数据了。接下来就可以对采集上来的数据进行处理或上传到上层系统中了。

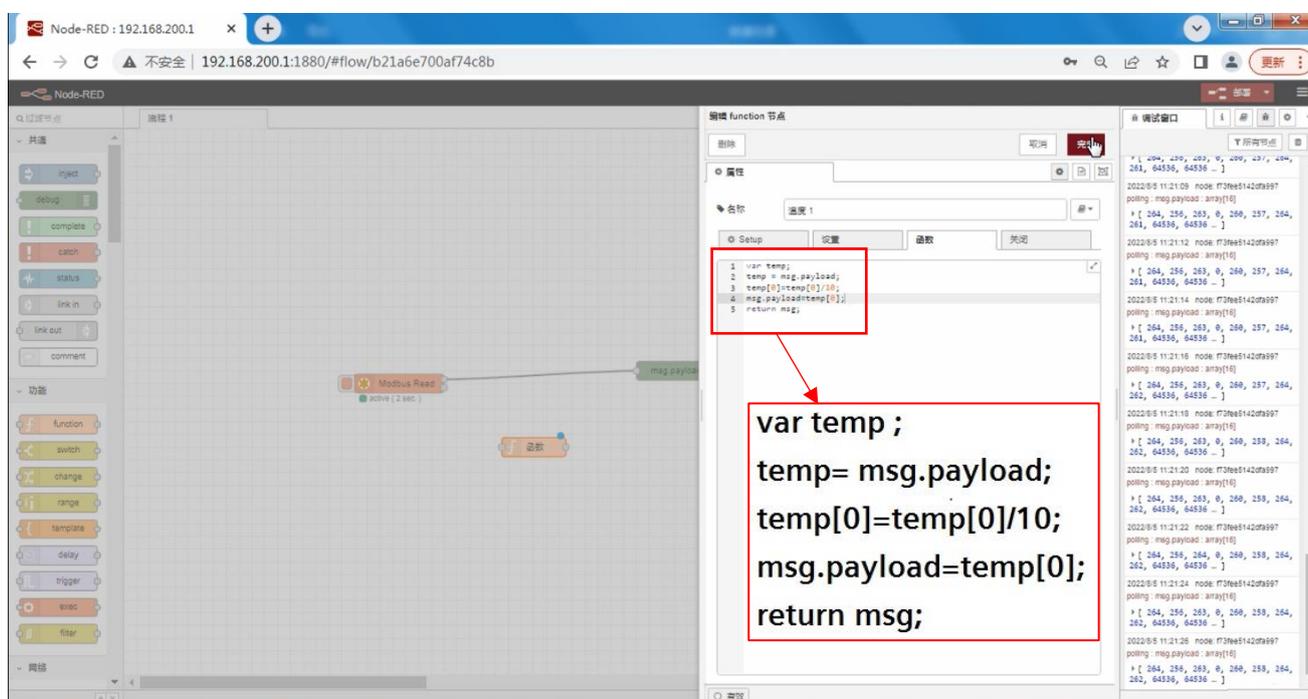


4.7 制作 UI 界面

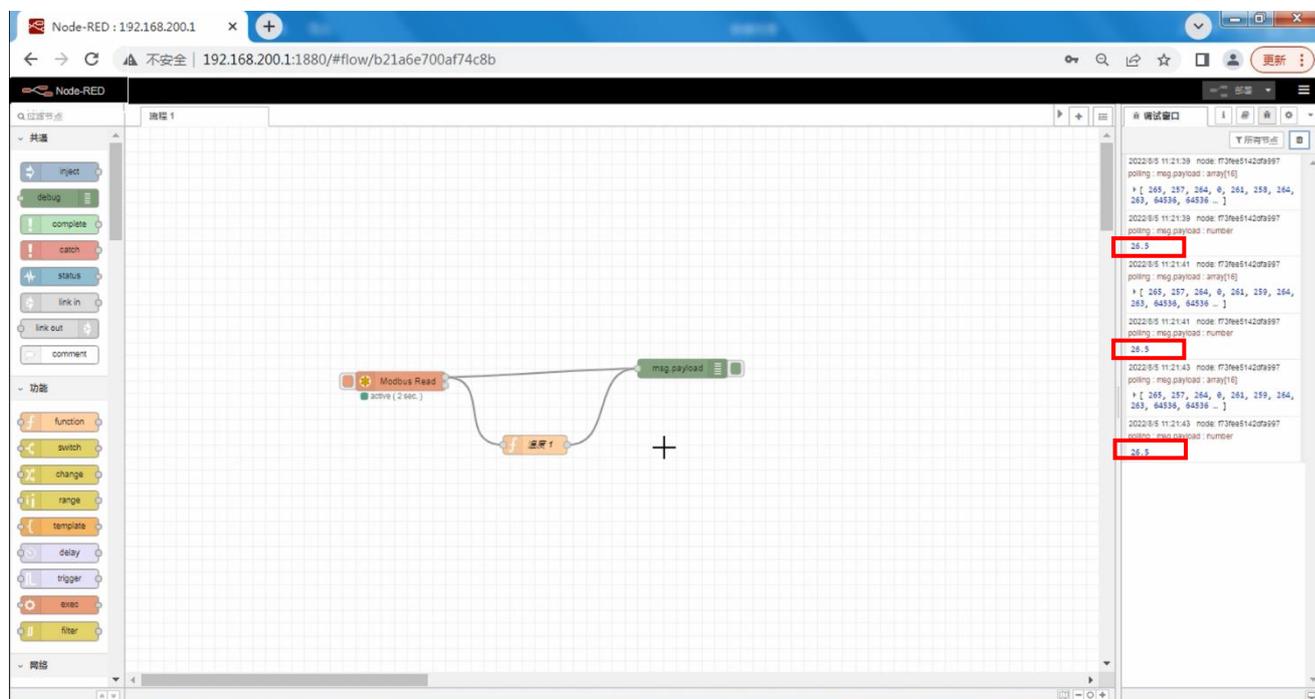
通过不同的功能节点，可以对 Modbus 节点采集上的传感器数据进行处理，并制作成 UI 界面。

在上一章节中，从传感器采集上来的数据，是被以数组形式保存的，可以利用 function 节点，通过简单编程，新建变量并将数组中的数值分配给对应的变量。

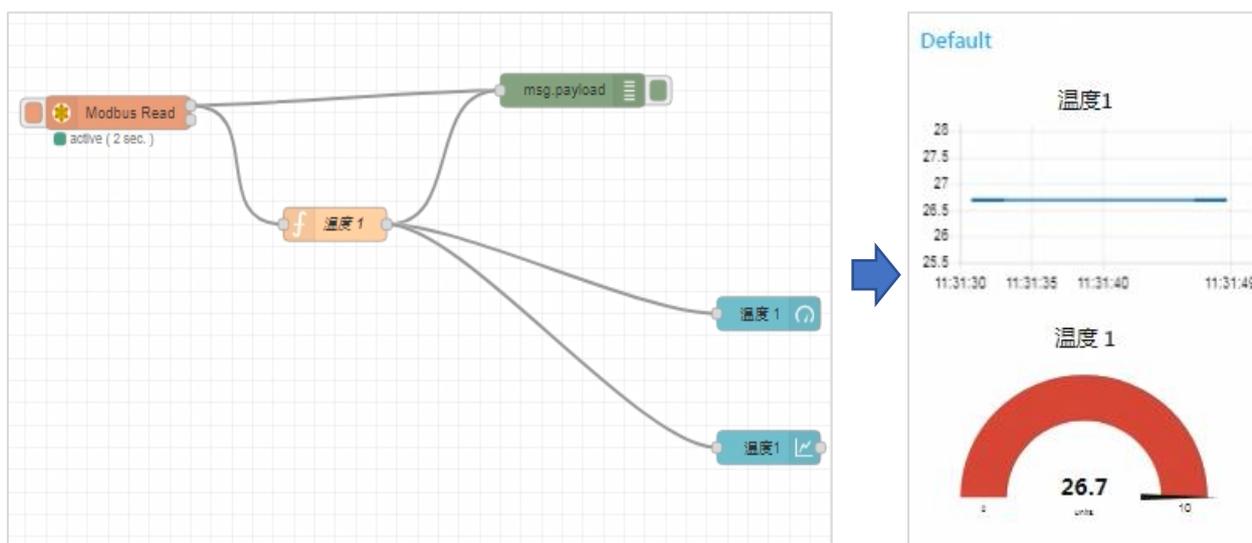
拖动一个 function 节点到工作区域中，双击节点进入编辑页面，为了方便识别，将函数名称命名为“温度 1”，输入图中所示代码，将数组中的第一组数据提取并换算为正常的温度值后重新赋值给 payload 参数。点击完成保存设置。



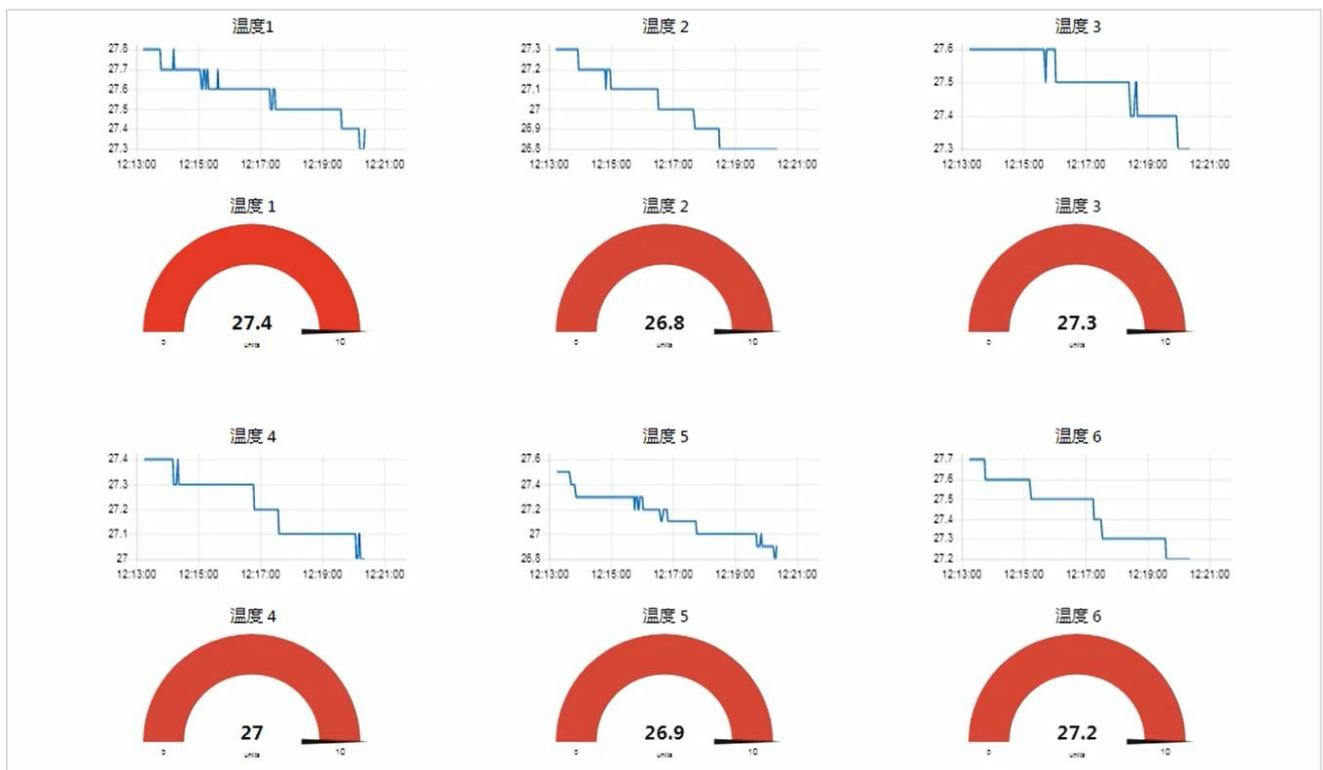
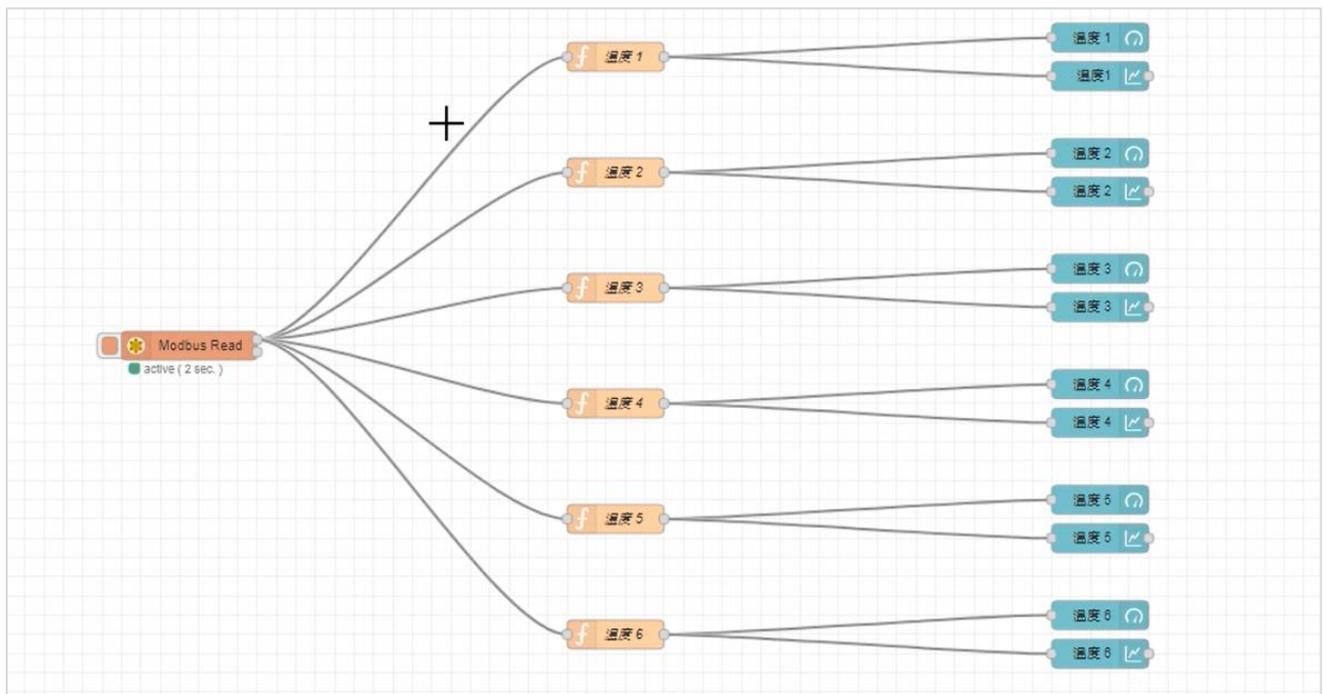
连接好对应节点的输入输出点，点击部署将程序部署到 IOT2050 中。部署后，数组中的第一组数据会被转换为有效数值并显示到调试窗口中。



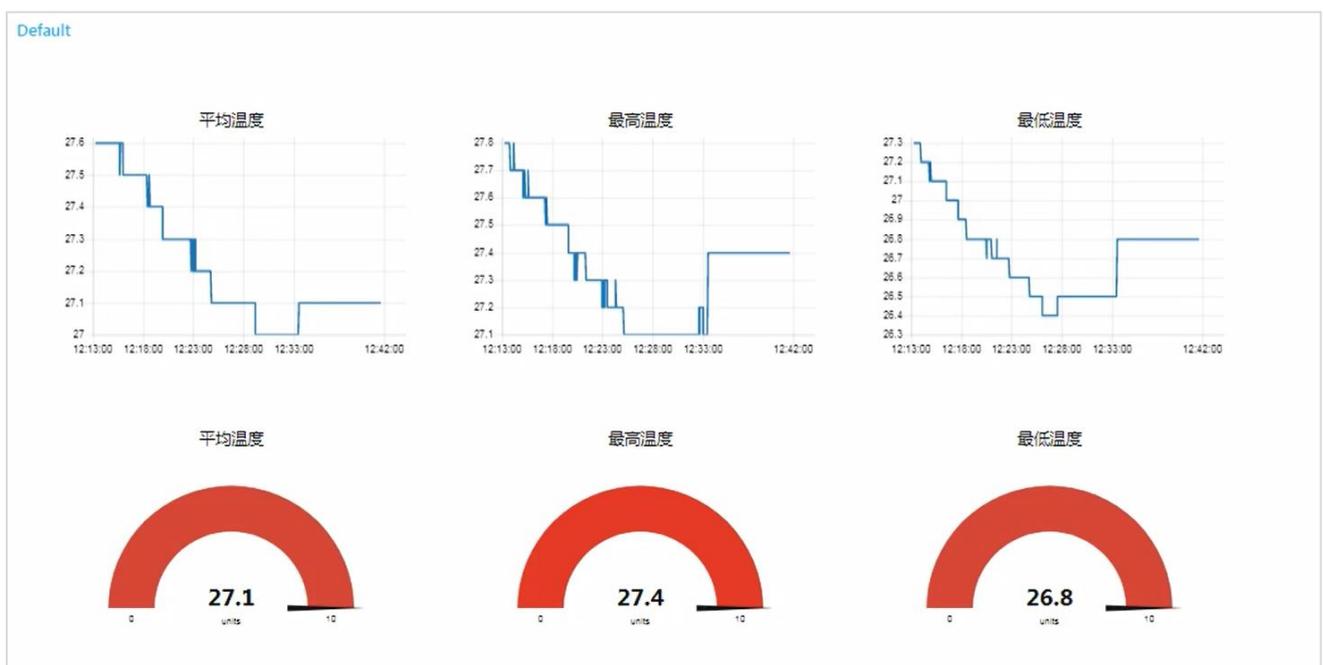
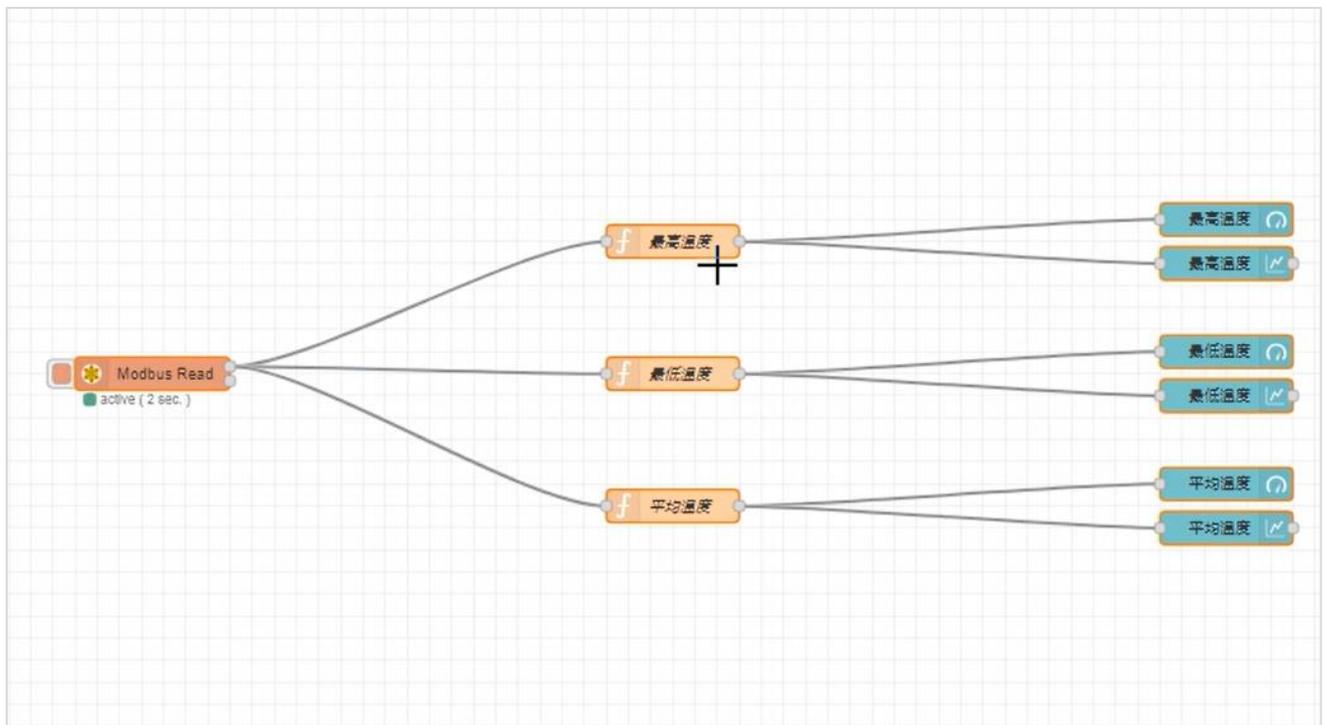
拖动一个仪表盘节点和图表节点到工作区域中，按照前面介绍过的内容对仪表盘节点和图表节点进行设置，设置好后，连接节点并部署。进入 UI 页面，就可以见到做出的画面效果。当对应传感器的数值发生变化时，仪表盘和图表中的内容也会随之变化。



按照同样的方法，我们就可以将从传感器采集到的数组中的几组有效数据全部提取出来，制作成我们想要的画面效果。



除此之外，我们还可以利用 function 节点对数据进行进一步处理，例如计算所有数据的最大、最小和平均值等等，使数据更加简单，直观和有效。

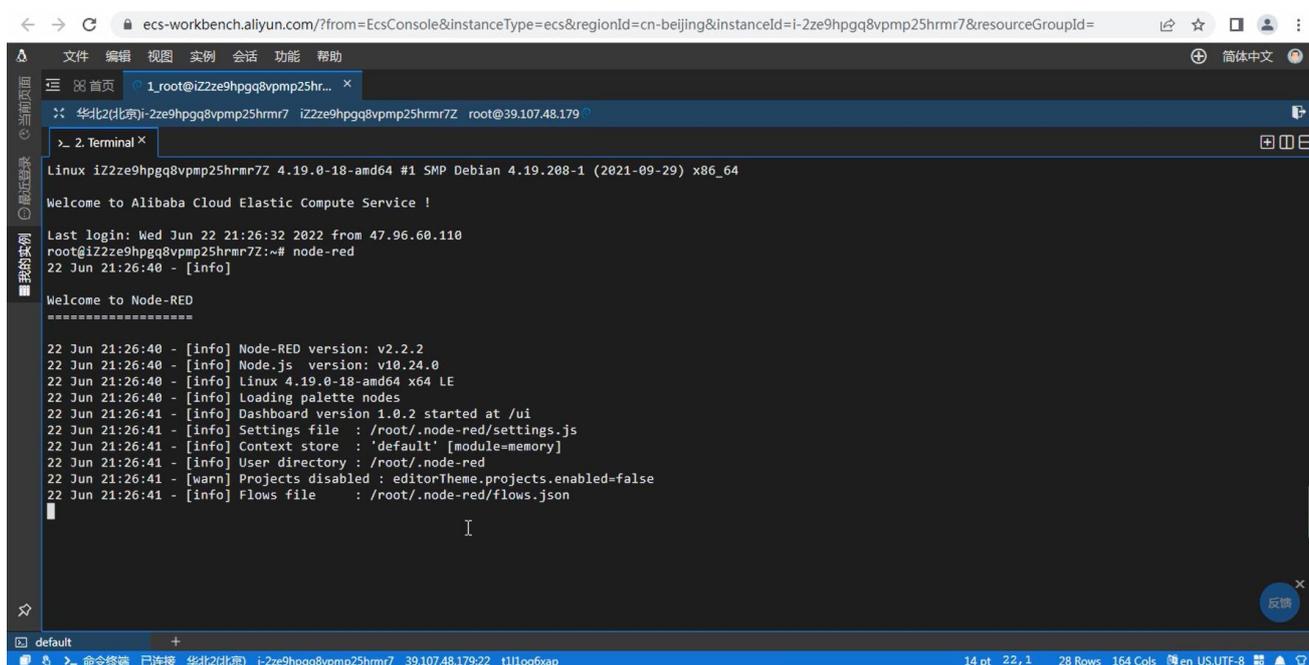


4.8 通过 MQTT 节点上传数据

通过 SIMATIC IOT2050 集成 Node-red 应用的 mqtt out 节点，将数据传送至上层系统或云平台上。本章节以阿里云为例，介绍 IOT2050 如何通过 mqtt out 节点数据上传到阿里云上。

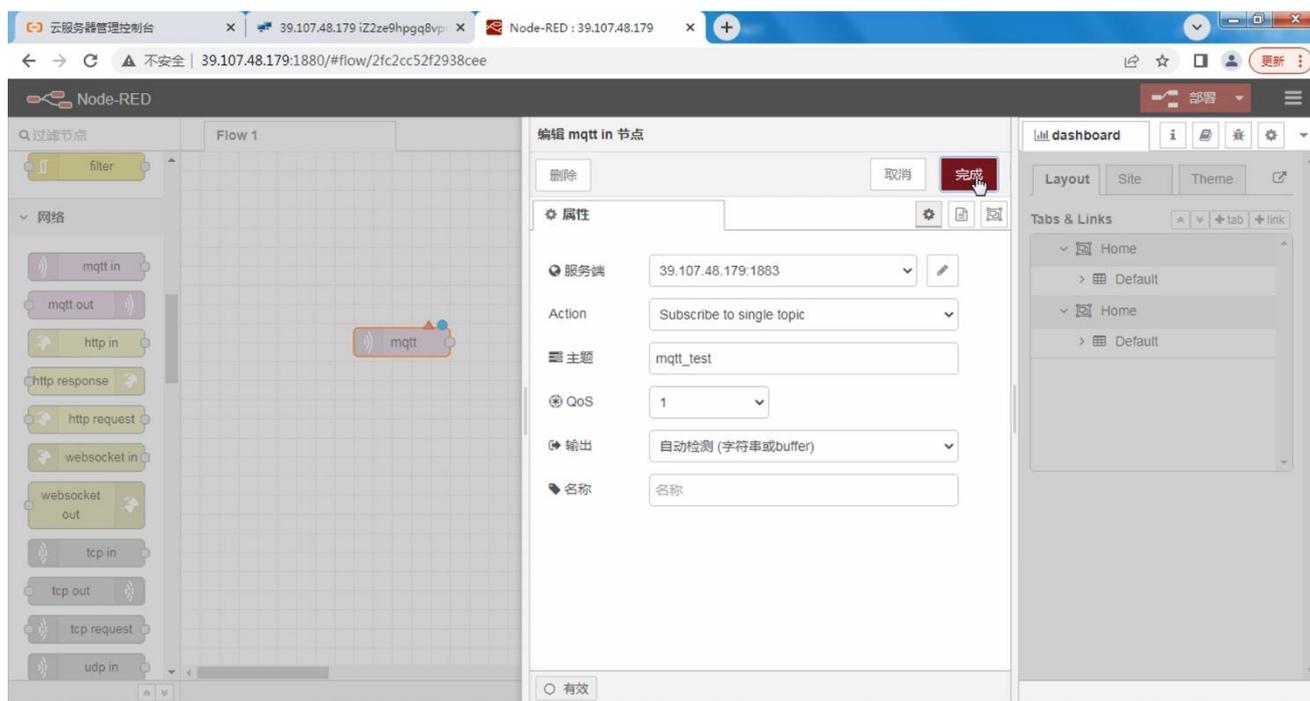
为了能让数据在云平台上展示，首先需要用户申请一台自己的云服务器，并在服务其中安装好支持 mqtt 服务的相关程序。在使用 IOT2050 进行 mqtt 节点的设置前，需要先在云服务器上把 mqtt 的数据接收端设置好。

启动运行云服务器，运行服务其中安装好的 node-red 应用。



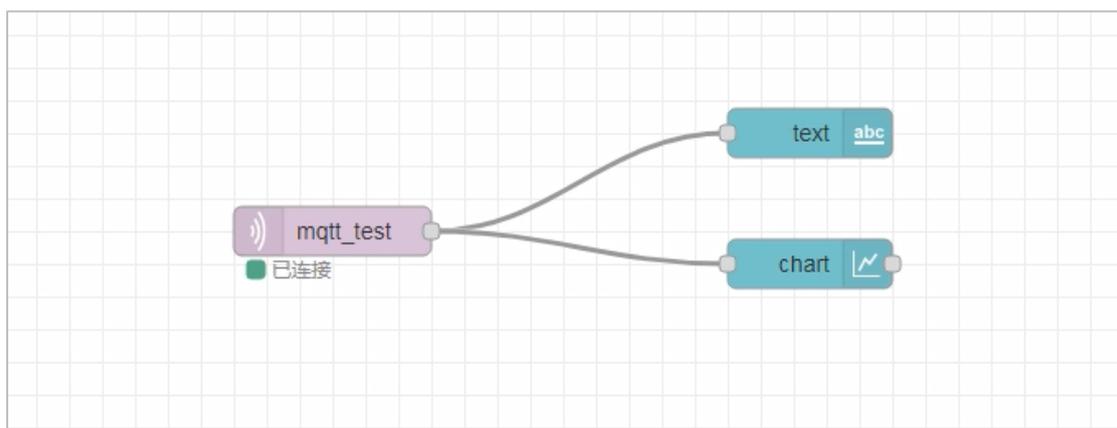
```
ecs-workbench.aliyun.com/?from=EcsConsole&instanceType=ecs&regionId=cn-beijing&instanceId=i-2ze9hpgq8vpmp25hrmr7&resourceGroupId=
文件 编辑 视图 实例 会话 功能 帮助
1_root@i22ze9hpgq8vpmp25hr...
华北2(北京)i-2ze9hpgq8vpmp25hrmr7 i22ze9hpgq8vpmp25hrmr7Z root@39.107.48.179
Terminal
Linux i22ze9hpgq8vpmp25hrmr7Z 4.19.0-18-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) x86_64
Welcome to Alibaba Cloud Elastic Compute Service !
Last login: Wed Jun 22 21:26:32 2022 from 47.96.60.110
root@i22ze9hpgq8vpmp25hrmr7Z:~# node-red
22 Jun 21:26:40 - [info]
Welcome to Node-RED
=====
22 Jun 21:26:40 - [info] Node-RED version: v2.2.2
22 Jun 21:26:40 - [info] Node.js version: v10.24.0
22 Jun 21:26:40 - [info] Linux 4.19.0-18-amd64 x64 LE
22 Jun 21:26:40 - [info] Loading palette nodes
22 Jun 21:26:41 - [info] Dashboard version 1.0.2 started at /ui
22 Jun 21:26:41 - [info] Settings file : /root/.node-red/settings.js
22 Jun 21:26:41 - [info] Context store : 'default' [module=memory]
22 Jun 21:26:41 - [info] User directory : /root/.node-red
22 Jun 21:26:41 - [warn] Projects disabled : editorTheme.projects.enabled=false
22 Jun 21:26:41 - [info] Flows file : /root/.node-red/flows.json
I
default
命令终端 已连接 华北2(北京) i-2ze9hpgq8vpmp25hrmr7 39.107.48.179:22 t11log6xap 14 pt 22,1 28 Rows 164 Cols en_US.UTF-8
```

在浏览器中输入云服务器的地址和端口号进入 node-red 应用界面。找到 mqtt in 节点并将其拖拽至工作区域中，双击节点进入设置页面，点击服务端选项框右端的铅笔按钮，进入连接设置页。在服务端选项框中输入云服务器的地址，端口号保持 1883 不变，其余各选项保持默认不变，点击更新按钮。在主题选项框中填写想要设置的名称，注意此处的主题名称要与后面在 IOT2050 上设置 mqtt 节点参数时的主题名称保持一致，此处填写为 mqtt_test。Qos 处为 mqtt 数据的服务质量，此处选择为 1，其他内容保持默认不变。点击完成保存设置。

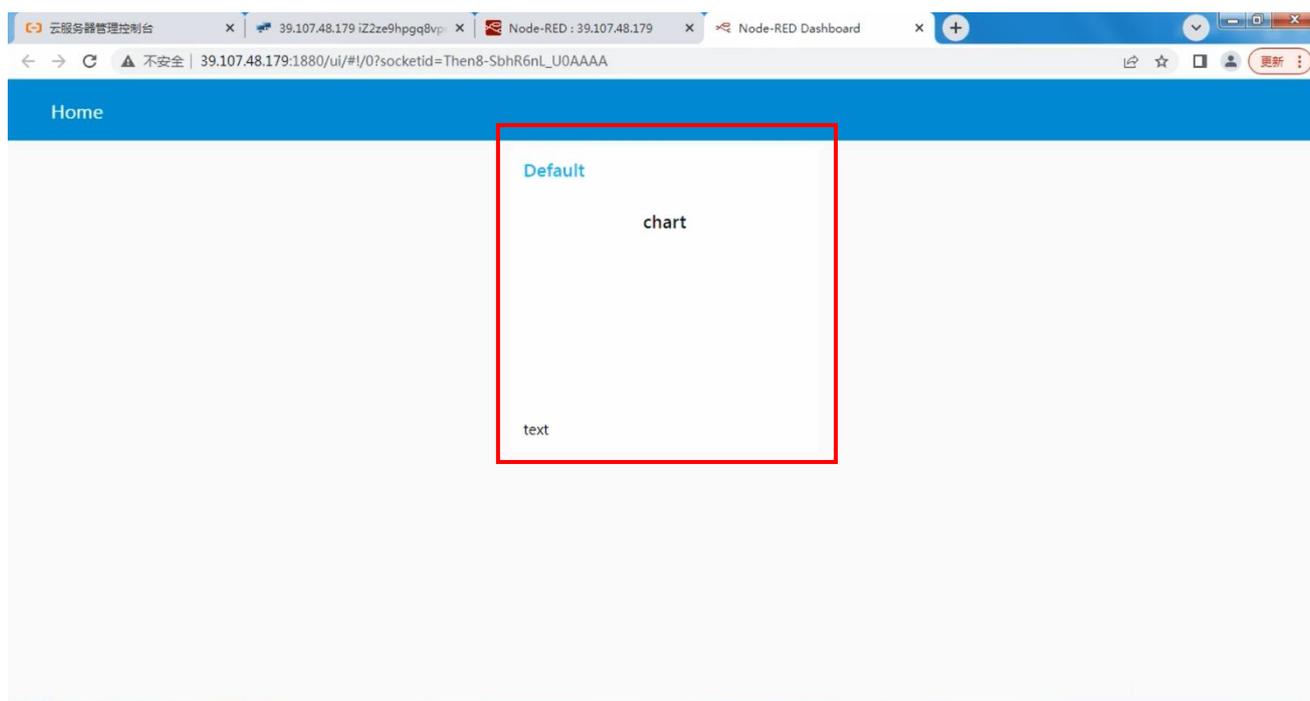


分别将一个 text 节点和 chart 节点拖入工作区中，进行相关设置后，将 mqtt in 节点的输出端与两节点的输入端进行连接，点击部署将程序保存到云服务器中。

可以看到 mqtt in 节点的状态变为已连接，该节点接收到数据后，会将数据传送给 test 和 chart 节点，在 UI 页面中显示出来

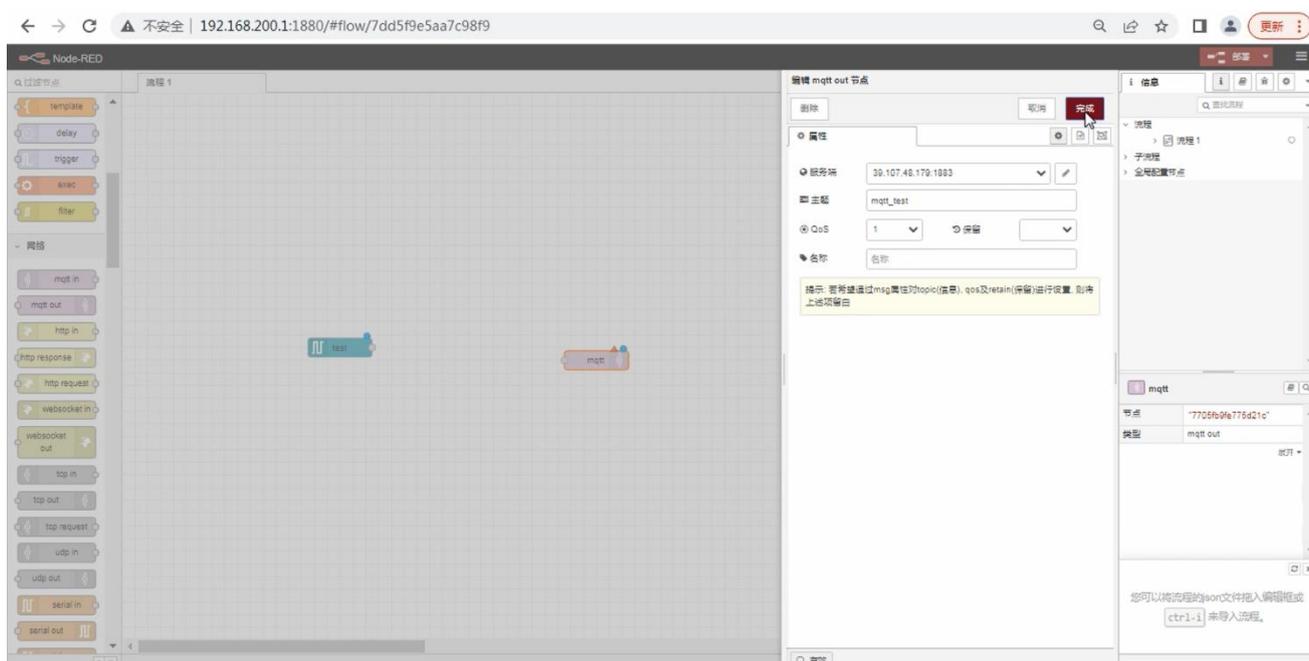


进入 UI 页面，此时可以看到，由于并没有数值传送到 mqtt in 节点，所以此时 UI 页面的图表和文本框中并没有内容。

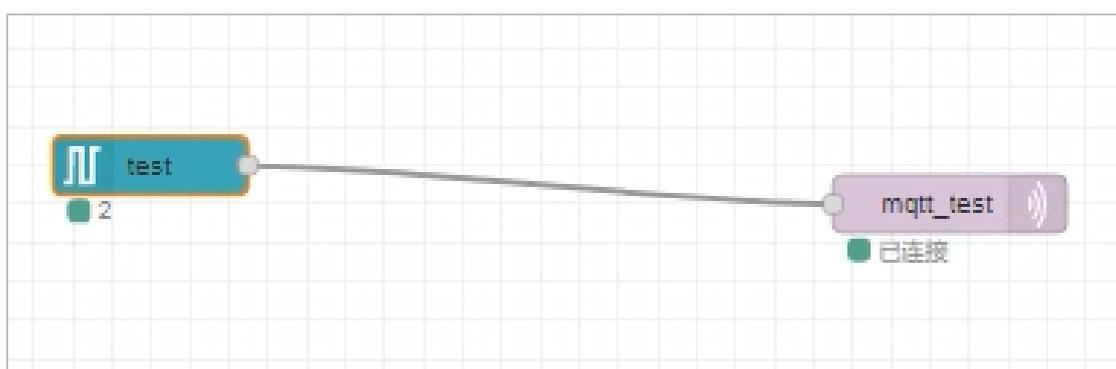


启动 IOT2050，进入 node-red 操作界面。拖动 S7 in 节点节点到工作区并进行相应设置，S7-in 节点的设置方法参考章节[“通过 S7 节点读取 PLC 数据”](#)

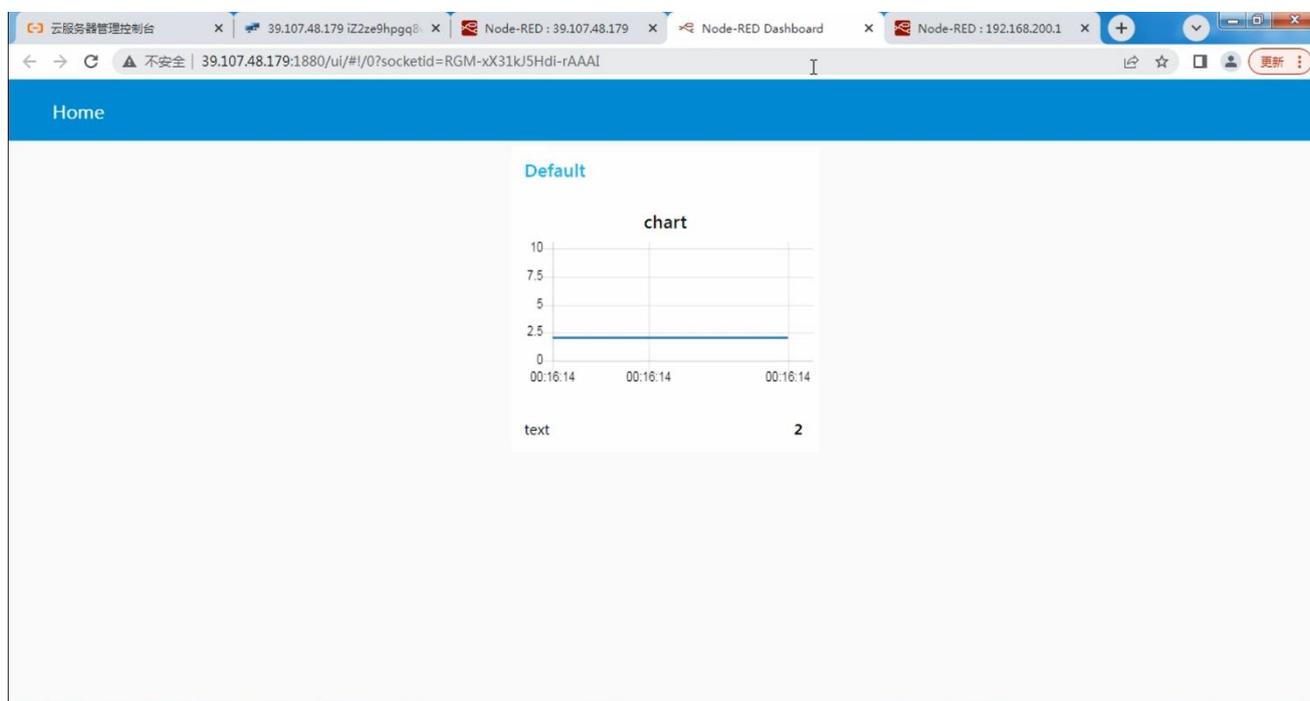
拖动 mqtt out 节点到工作区，双击节点进入设置页面，点击服务端选项框右端的铅笔按钮，进入连接设置页面，在服务端选项框中输入云服务器的地址，端口号保持 1883 不变，其余各选项保持默认内容不变，点击更新按钮。在主题选项框中填入与云端 mqtt in 节点相同的主题名称 mqtt_test，QoS 选项也填写与云端一致的内容，选择 1。点击完成保存设置。



将 S7 in 节点的输出端与 mqtt out 节点的输入端进行连接，点击部署按钮将程序保存到 IOT2050 中。可以看到 mqtt out 节点的状态也变成了已连接。



此时通过 S7 in 节点采集到的 PLC 的数据将会通过 mqtt out 节点发送到云端。再次打开云服务器的 UI 页面，可以看到 UI 页面中的文本框和图表中已经有数值被传送上来



5.Edgeconnect Image 设置及应用

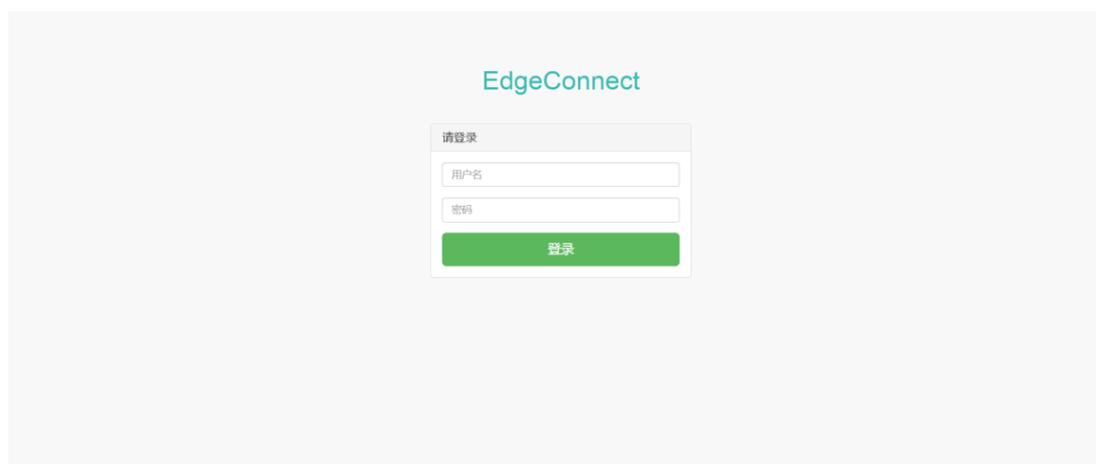
EdgeConnect 是针对西门子 SIMATIC IOT2000 系列工业智能网关的边缘解决方案。该解决方案提供 WEB 管理界面，用户可通过 WEB 界面进行数据采集、云端上传、脚本编程、远程维护、网络管理、系统设置、证书管理及导入导出等功能的操作。

EdgeConnect 以.img 系统镜像的形式进行分发。用户拿到的系统镜像文件为压缩文件，可使用 7-Zip 等工具对压缩文件进行解压，解压之后的文件为.img 系统镜像。

用户可以使用 Win32 Disk Imager 等软件将.img 系统镜像烧录至 SD 卡，将 SD 卡放入 SIMATIC IOT2050 中，上电启动，启动约需 1 分钟左右。第一次上电启动时间较长，请耐心等待直至 STAT 指示灯不停闪烁为止。

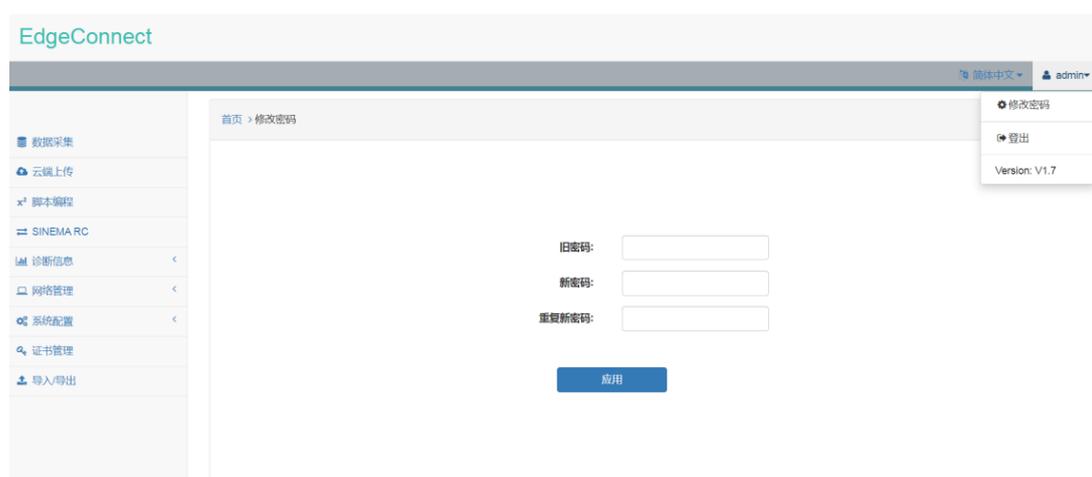
5.1 启动 Edgeconnect 应用

将 PC 通过网线连接至 SIMATIC IOT2050 的 LAN P1 网口，设置 PC 的 IP 为 192.168.200.X 网段，在 PC 的浏览器中输入 <http://192.168.200.1:8000>，即可进入 EdgeConnect WEB 管理界面。



WEB 管理界面登录

首次登陆时需要使用默认用户名及密码，默认用户名为 admin，默认密码为 123。用户登录之后请立即修改默认密码，防止非法用户登录。



修改默认密码

用户可根据使用习惯切换语言为简体中文或英文。



切换用户
界面语言

注：由于
设备性能
及协议相
关限制，

EdgeConnect 配置过程中请遵循以下原则：

- 1) 通道数量 ≤ 32 (含数据采集通道、脚本编程通道及云端上传通道)
- 2) MindSphere 通道数量 ≤ 1
- 3) 每秒钟采集点数 ≤ 200 (取决于云端平台限制)
- 4) 通道及变量名称命名不可重复

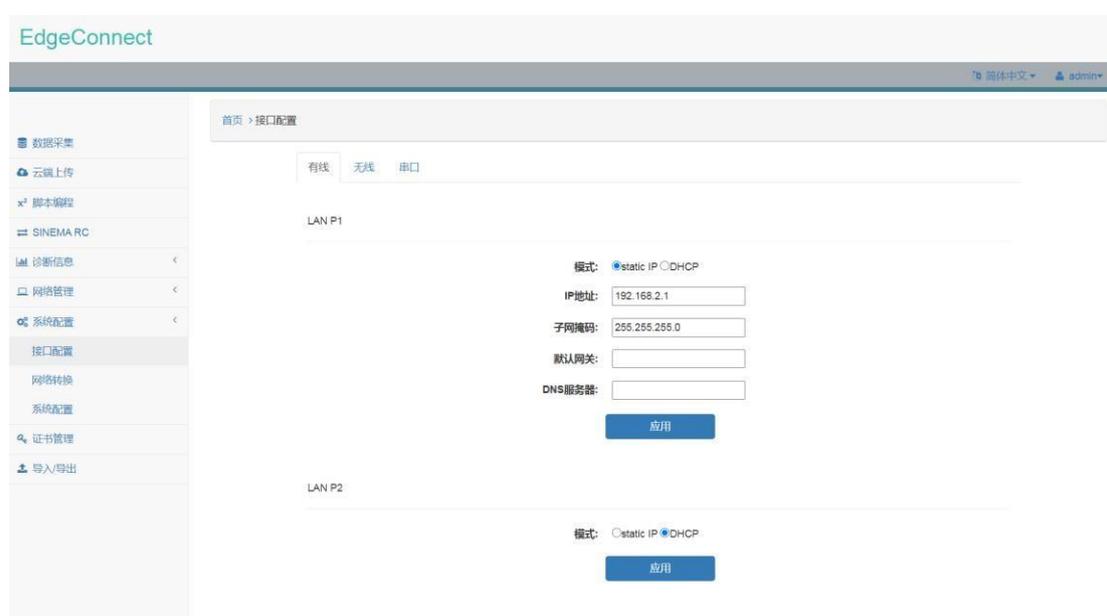
5.2 系统配置模块

5.2.1 接口配置

用户可以对有线、无线及串口等接口进行配置。

有线接口配置

有线接口配置支持对 LAN P1 及 LAN P2 接口进行配置。接口可配置成静态 IP 或者 DHCP 动态获取两种模式。



有线接口配置界面

需要注意的是：

- 1) 设置 LAN P1 或 LAN P2 接口为 DHCP 动态获取模式时，请将该接口连接至支持 DHCP 服务的路由设备。
- 2) 在使用 SINEMA RC 对现场设备进行远程维护时，需要将现场设备连接的以太网接口 IP 设置在现场设备的同一个网段。

无线接口配置

无线接口配置支持对 WIFI 及 4G/5G 接口进行配置。

1) WIFI

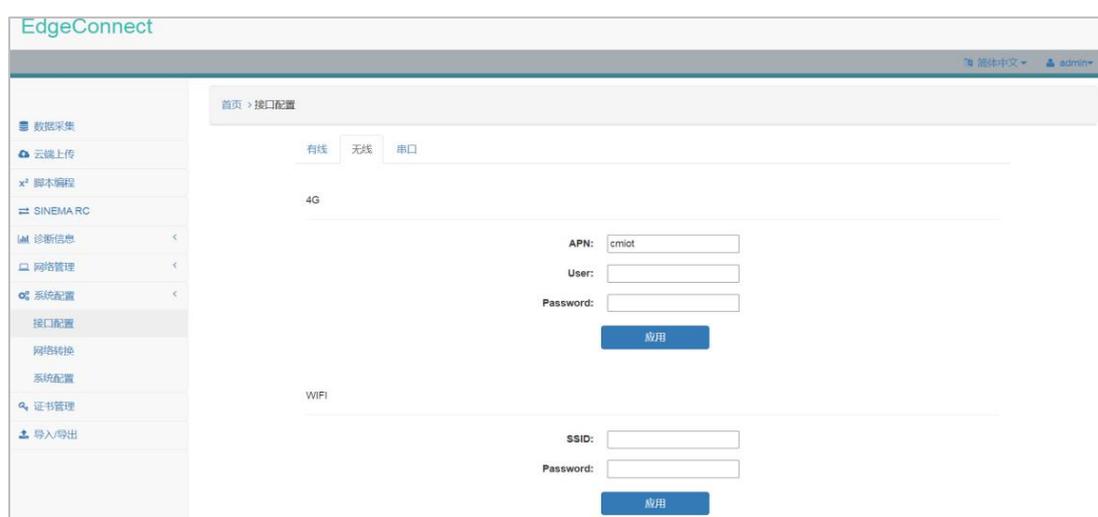
使用 WIFI 之前，需要插入 USB 免驱无线网卡，可以通过系统状态信息页面查看 WIFI 网卡是否安装。

WIFI 接口可配置 SSID 以及密码等信息。

2) 4G/5G

使用 4G/5G 之前，需要安装 miniPCIe 接口的 4G/5G 模组，如 5G 模组为 M.2 接口，可通过 M.2 转 miniPCIe 转接板安装。用户可以通过系统状态信息页面查看 4G/5G 模组是否正确安装并识别。

4G/5G 接口可以设置 APN、用户名及密码等信息，其中，APN、用户名及密码等信息可以从 SIM 卡提供商处获取。建议使用运营商提供的物联网卡。如 SIM 卡设置了 PIN，请取消该设置。



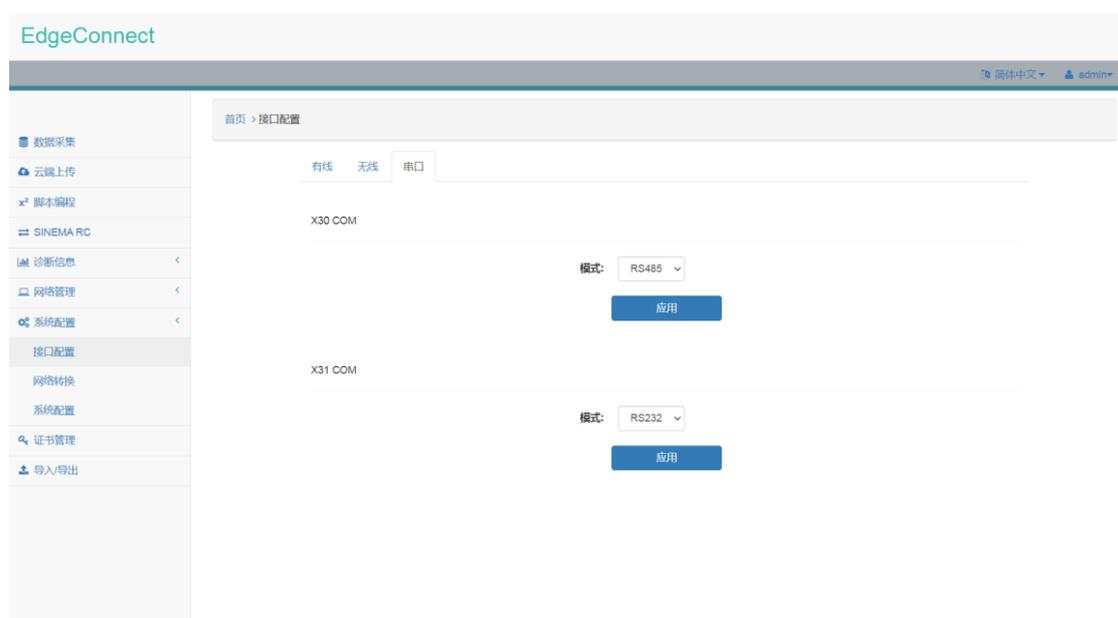
The screenshot displays the EdgeConnect web interface for configuring wireless connections. The page title is 'EdgeConnect' and the user is logged in as 'admin'. The main navigation menu on the left includes options like '数据采集', '云端上传', '脚本编程', 'SINEMA RC', '诊断信息', '网络管理', '系统配置', '接口配置', '网络转换', '证书管理', and '导入/导出'. The '接口配置' (Interface Configuration) section is active, showing tabs for '有线' (Wired), '无线' (Wireless), and '串口' (Serial). Under the '无线' tab, there are two sections: '4G' and 'WIFI'. The '4G' section has input fields for 'APN' (containing 'cmiot'), 'User', and 'Password', with an '应用' (Apply) button below. The 'WIFI' section has input fields for 'SSID' and 'Password', also with an '应用' (Apply) button below.

无线接口配置界面

串口接口配置

串口支持对 X30 COM 以及 X31 COM 进行配置，可配置三种模式：RS485、RS232 以及RS422。

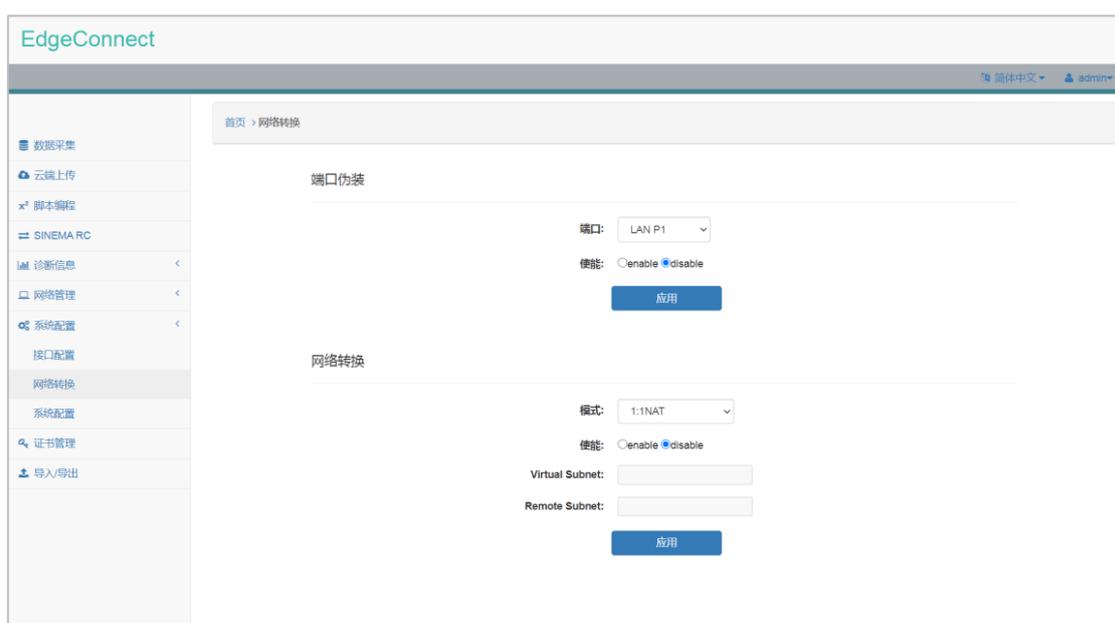
IOT2050 仅支持X30 COM，如对 X31 COM 进行配置，则会返回“Device is not available”提示信息。



串口接口配置界面

5.2.2 网络转换

在 SINEMA RC 模块成功建立连接后，需要进行相应的网络转换设置才可以实现远程维护。网络转换设置包括端口伪装以及 NAT 转换。



网络转换配置界面

端口伪装

端口伪装可以对 LAN P1、LAN P2、SINEMA RC 及 Cellular 4G 接口进行设置。

在实际使用中，SIMATIC IOT2000 的 LAN P1 或者 LAN P2 端口连接到现场设备的以太网端口。通过 SINEMA RC 连接转发过来的数据包在现场设备未设置默认网关的情况下无法从现场设备返回 SIMATIC IOT2000。通过对 LAN P1 或 LAN P2 的端口伪装，可以解决现场设备未设置默认网关或无法更新默认网关的问题。

在多个 EdgeConnect 设备接入 SINEMA RC 服务器的情况下，如一个 EdgeConnect 设备希望访问其他 EdgeConnect 设备，则需对 SINEMA RC 接口进行端口伪装设置。具体设置请参考文档“基于 SIMATIC IOT2000 及 SINEMA RC 的工业远程维护方案”。

在 EdgeConnect 设备通过 4G 联网的情况下，如现场 PC 希望借助 EdgeConnect 设备 4G 网络连接至 Internet，则需对 Cellular 4G 接口进行端口伪装设置。在实际使用中，现场 PC 通过网线连接至 SIMATIC IOT2000 的 LAN P1 或 LAN P2 接口，将现场 PC 的 IP 地址设置为 LAN P1 或 LAN P2 同一网段地址，默认网关设置为 LAN P1 或 LAN P2 的 IP 地址，DNS 服务器设置为公用的 DNS 服务器。通过以上设置，现场 PC 即可借助 EdgeConnect 设备 4G 网络连接至 Internet。

网络转换

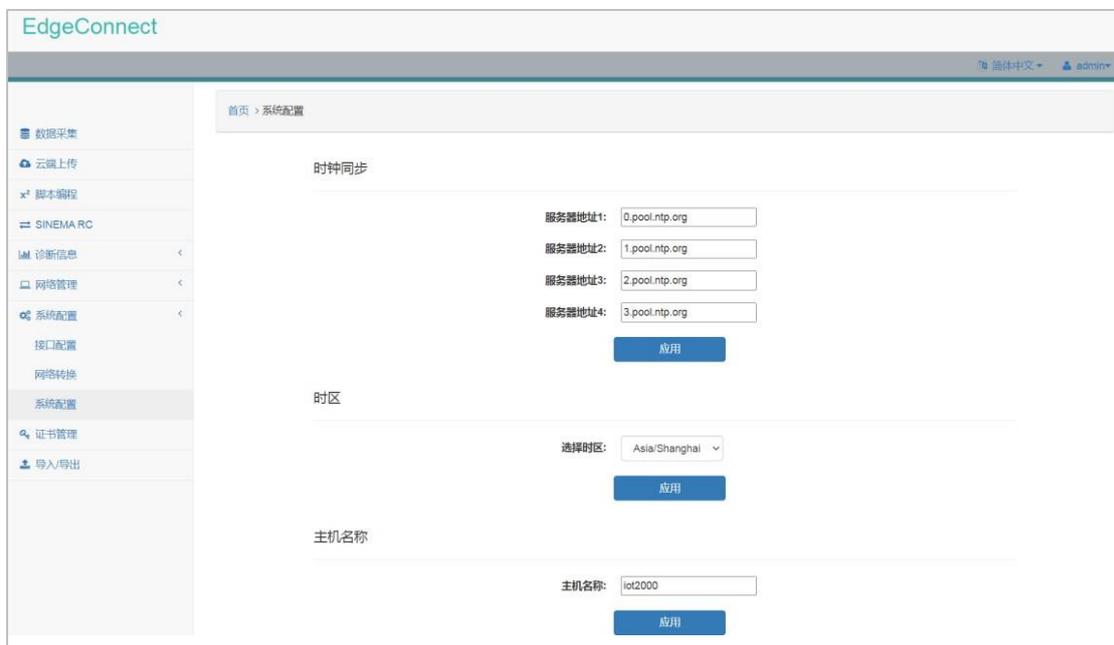
在多个 EdgeConnect 设备接入 SINEMA RC 服务器的情况下，可能会存在现场设备拥有相同 IP 地址，即工程软件无法区分不同现场设备的问题。

SINEMA RC 通过将现场设备 IP 映射到虚拟 IP 的方式来解决该问题，虚拟 IP 映射支持 1:1 NAT 或 1 to 1 NAT。

- 1) 1:1 NAT 是将整个网段进行映射，如 172.10.2.0/24 映射到 192.168.2.0/24
- 2) 1 to 1 NAT 是将单个 IP 进行映射，如 172.10.2.1 映射到 192.168.2.1

5.2.3 系统配置

用户可设置时钟同步、时区及主机名称。时钟同步需用户手动进行同步。时区可选择 UTC 时间或者北京时间。

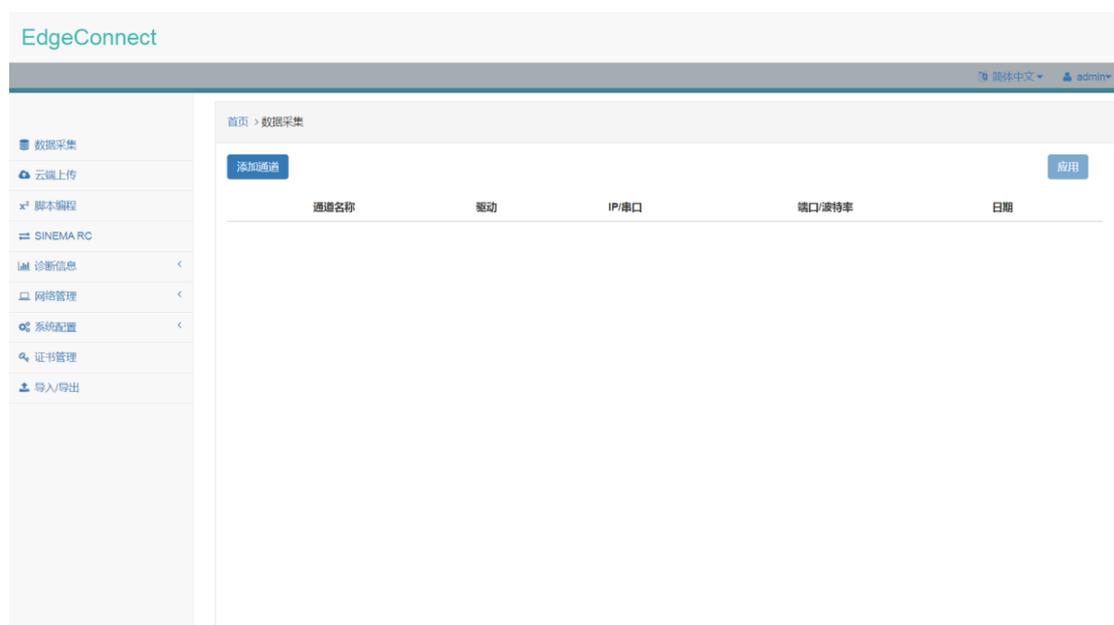


The screenshot displays the 'EdgeConnect' system configuration interface. The left sidebar contains a navigation menu with options: 数据采集, 云端上传, 脚本编程, SINEMA RC, 诊断信息, 网络管理, 系统配置 (highlighted), 接口配置, 网络转换, 证书管理, and 导入/导出. The main content area is titled '系统配置' and includes three sections: '时钟同步' with four NTP server address input fields (all set to '0.pool.ntp.org') and an '应用' button; '时区' with a dropdown menu set to 'Asia/Shanghai' and an '应用' button; and '主机名称' with an input field set to 'iot2000' and an '应用' button. The top right of the interface shows '简体中文' and 'admin'.

系统配置界面

5.3 数据采集模块

数据采集模块目前支持多种现场设备通信协议，如IOSHIELD、Modbus_RTU / TCP、OPCUA及主流 PLC 协议等。本章节将介绍如何使用数据采集模块。



数据采集模块界面

用户通过点击“添加通道”按钮，可以选择不同的设备通信协议，并进行相应的配置。

所有通道配置结束之后，将配置完成的通道状态由停止状态切换至运行状态（红色->绿色），并点击“应用”按钮即可下发数据采集模块配置，并启动数据采集程序。

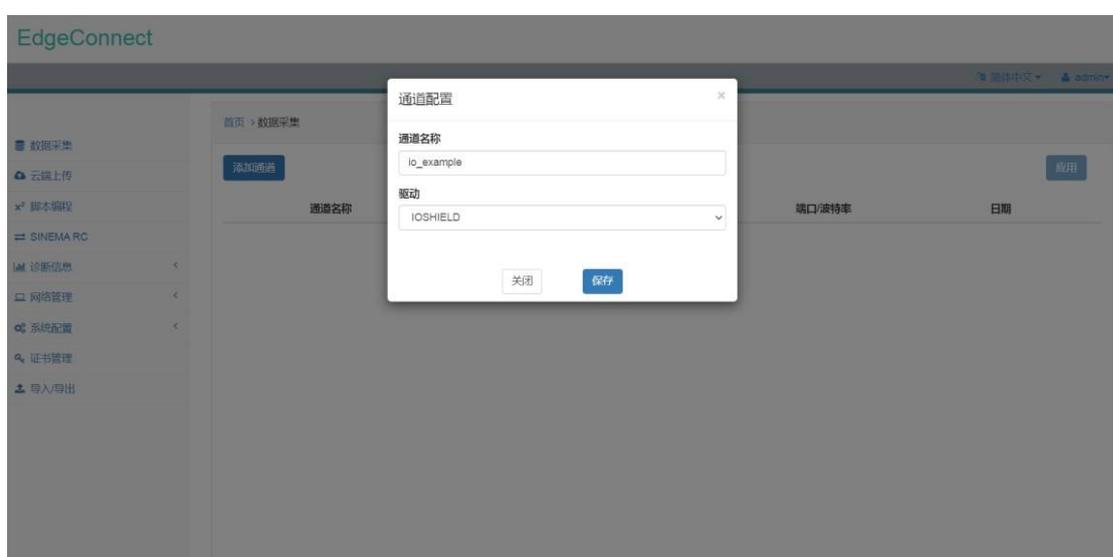
5.3.1 IOSHIELD通道

通道配置

在“驱动”下拉菜单中选中 IOSHIELD。

使用该通道之前，需安装好扩展 IO 模块，并根据产品使用手册完成相应接线。

IOSHIELD 通道仅支持数据读取功能。



IOSHIELD 通道配置界面

配置好 IOSHIELD 通道之后，IOSHIELD 通道信息会以列表形式显示在页面中。

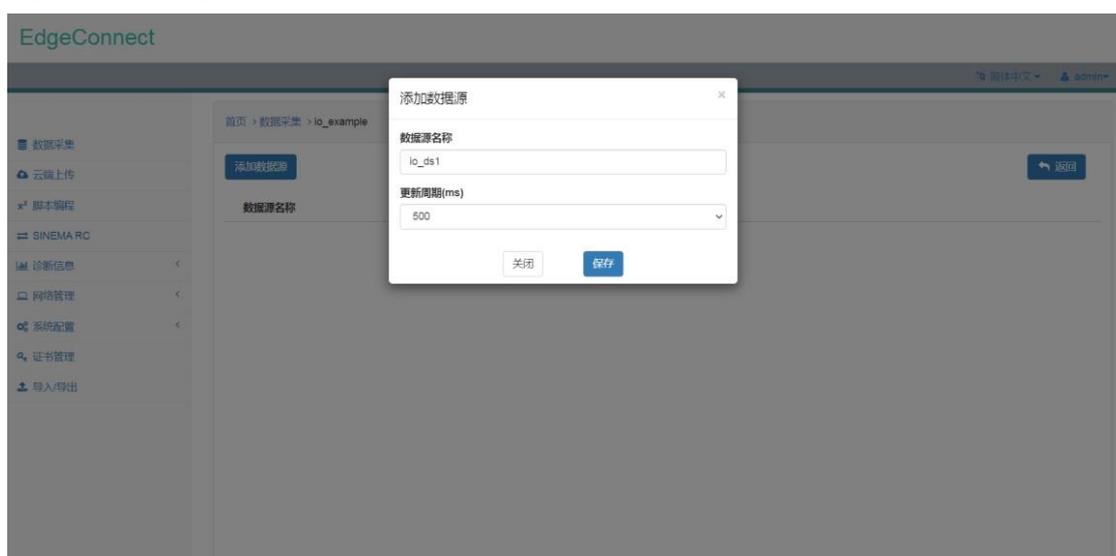


IOSHIELD 通道配置信息

通道详情配置

点击 IOSHIELD 通道列表项右侧 ，进入通道详情配置页面。

在通道详情配置页面中，用户可以点击“添加数据源”添加数据源（数据点分组），在数据源配置界面中设置数据源名称以及更新周期，即可完成数据源配置。



IOSHIELD 数据源配置界面

配置完数据源之后，数据源配置信息会以列表形式显示在页面中。

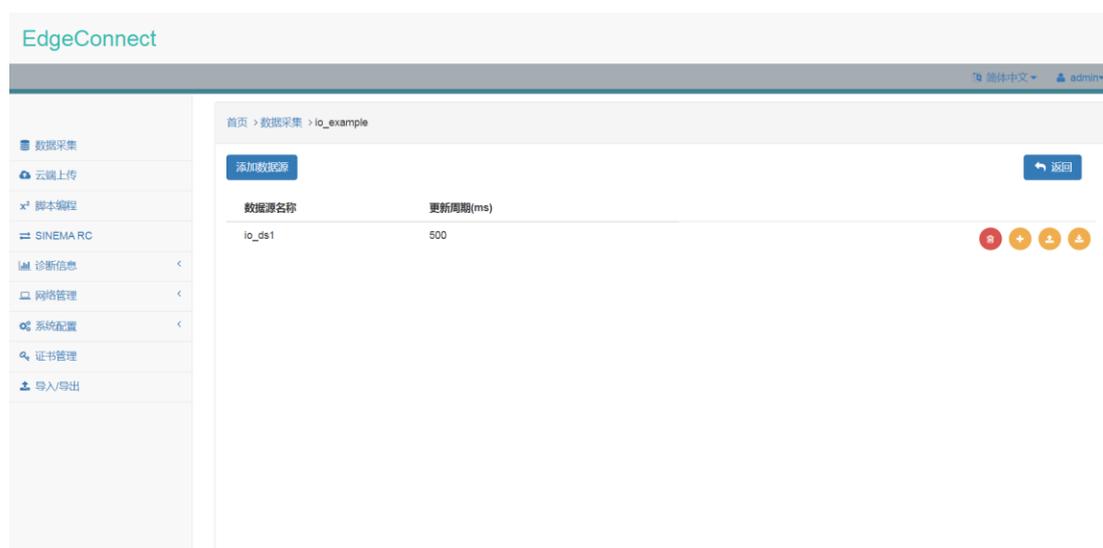
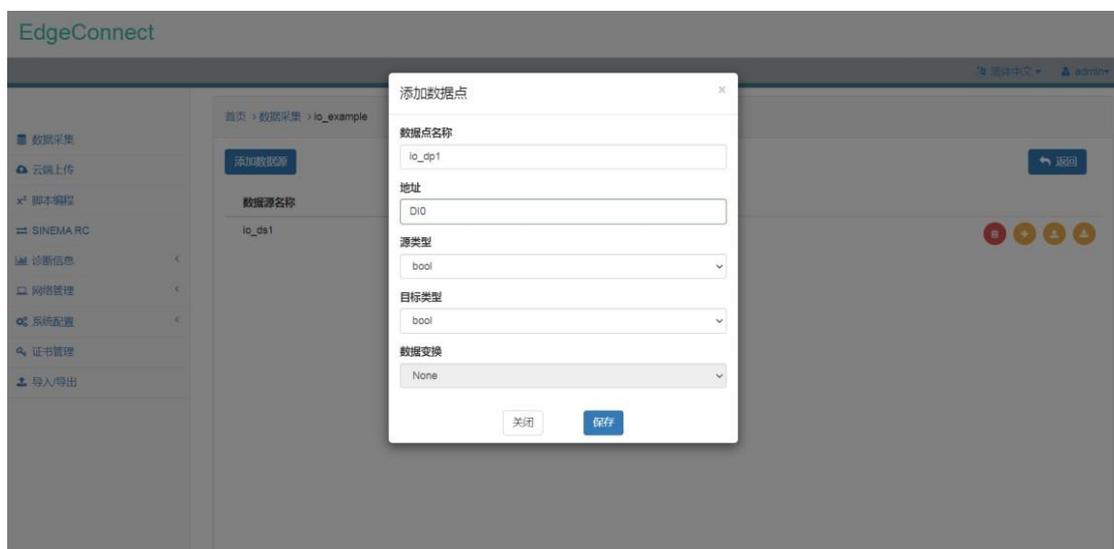


图4.5 IOSHIELD 数据源配置信息

点击数据源列表项右侧 \oplus ，即可打开数据点配置界面。

在数据点配置界面中，可以配置数据点名称、地址、源类型、目标类型、以及数据变换等信息。



IOSHIELD 数据点配置界面

其中，对于数据点地址需要按如下规则填写：

区域	地址格式
U	Ux, 其中 x 为 0 或 1
I	Ix, 其中 x 为 0 或 1
DI	DIx, 其中 x 为 0-4

IOSHIELD 数据点地址定义规则

IOSHIELD 支持的数据点源类型及对应的目标类型如下表所示：

源类型	源类型说明	目标类型
bool	1 bit	bool, int
float	4 bytes	double

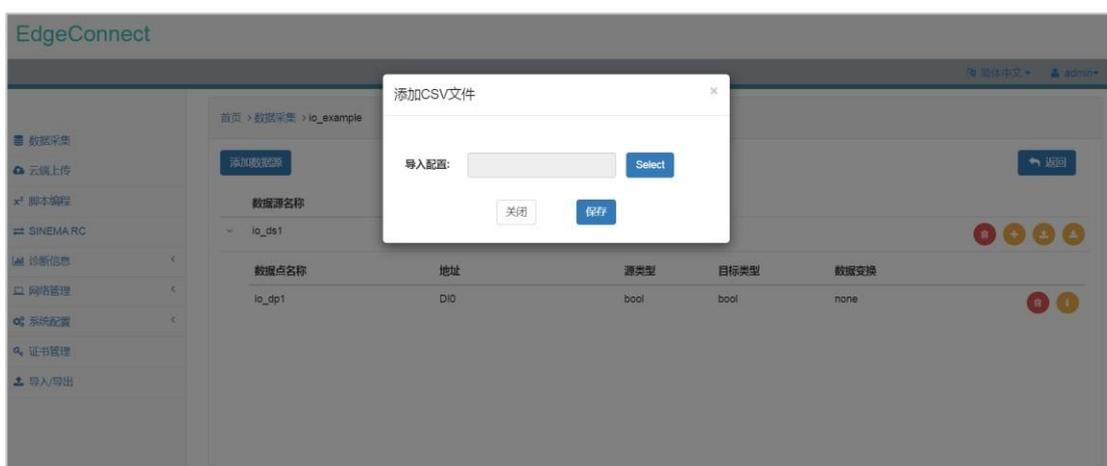
IOSHIELD 支持源类型及对应目标类型

对于数据变换，目前支持对数据点进行线性处理。在数据点目标类型为 int、double、long 时方可选择对数据点进行线性处理。线性处理的转换公式为 $V = xp + y$ 其中，x 为 multiplier，y 为 offset。配置完数据点信息之后，数据点信息会以列表形式显示在数据源中。通道配置应用下发后，点击数据点列表项右侧 ，可查看数据点当前最新值。



IOSHIELD 数据点配置信息

点击数据源列表项右侧 ，即可打开数据点批量导入界面，用户可以通过 CSV 文件批量添加数据点配置信息。



IOSHIELD 数据点批量添加界面

点击数据源列表项右侧 ，即可以 CSV 文件形式导出数据点配置信息

5.3.2 Modbus_RTU 通道

通道配置

在“驱动”下拉菜单中选中 Modbus_RTU，并选择相应的串口设备、波特率、数据位、停止位、校验位、内存布局、轮询延时及响应超时即可完成相应配置。

Modbus_RTU 通道仅支持数据读取功能。

1) 串口设备

- X30 COM, 默认支持 RS485 通信, 支持 SIMATIC IOT2050 Basic
- X30 COM (SIMATIC IOT2050 Advanced), 默认支持 RS485 通信, 支持 SIMATIC IOT2050 Advanced
- X31 COM, 默认支持 RS232 通信, 适用于 SIMATIC IOT2040

2) 波特率

波特率支持以下选项：

- 300
- 600
- 1200
- 2400
- 4800
- 9600
- 19200
- 38400
- 57600
- 76800
- 115200

3) 数据位

数据位支持以下选项：

- 5
- 6
- 7
- 8

4) 停止位

停止位支持以下选项：

- 1
- 2

5) 校验位

校验位支持以下选项：

- Odd
- None
- Even

6) 内存布局

读取浮点数或长整型需要选择对应的内存布局，内存布局支持以下选项：

- ABCD
- BADC
- CDAB
- DCBA

7) 轮询延时

轮询延时时间，默认为 20ms。

8) 响应超时

响应超时时间，默认为 1000ms。



Modbus_RTU 通道配置界面

配置好 Modbus_RTU 通道之后，Modbus_RTU 通道信息会以列表形式显示在页面中。

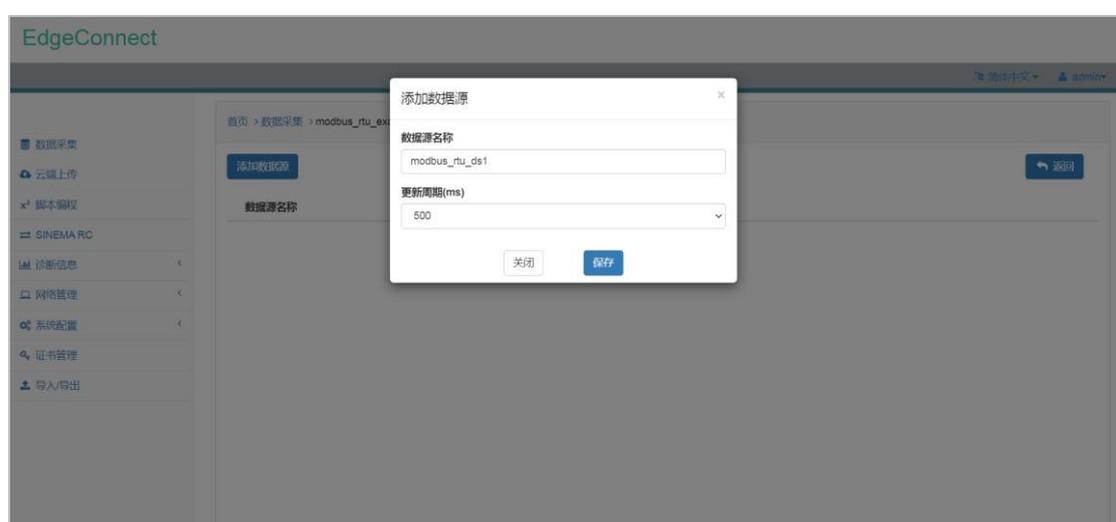


Modbus_RTU 通道配置信息

通道详情配置

点击 Modbus_RTU 通道列表项右侧 ，即可进入通道详情配置页面。

在通道详情配置页面，用户可以点击“添加数据源”添加数据源（数据点分组），在数据源配置界面中设置数据源名称以及更新周期，即可完成数据源配置。



Modbus_RTU 数据源配置界面

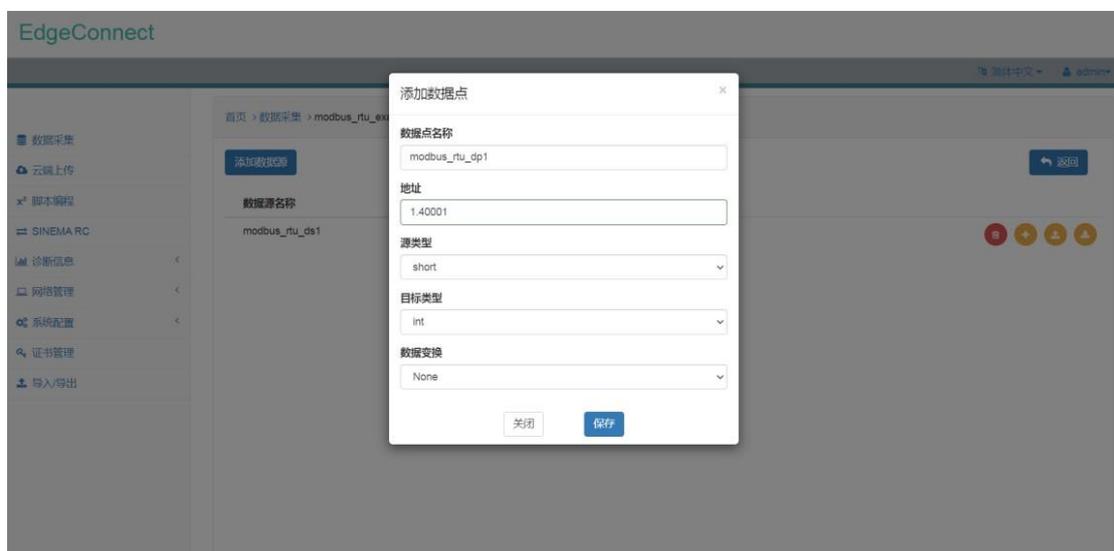
配置完数据源之后，数据源配置信息会以列表形式显示在页面中。



Modbus_RTU 数据源配置信息

点击数据源列表项右侧 \oplus ，即可打开数据点配置界面。

在数据点配置界面中，可以配置数据点名称、地址、源类型、目标类型、以及数据变换等信息。



Modbus_RTU 数据点配置界面

其中，对于数据点地址需要按如下规则填写：

X.YZ.F (如1.40001 或1.30002.10)

类型	功能	范围
X	Modbus 站号	1 - 247
Y	地址区域	线圈寄存器为 0 离散输入寄存器为 1 输入寄存器为 3 保持寄存器为 4
Z	字/位偏移地址	长度为 4 位整数，若不足 4 位，高位补 0 注意事项： 线圈寄存器及离散输入寄存器为位偏移地址输入寄存器及保持寄存器为字偏移地址
F	位偏移地址	0 - 15 注意事项： 仅输入寄存器或保持寄存器时有效

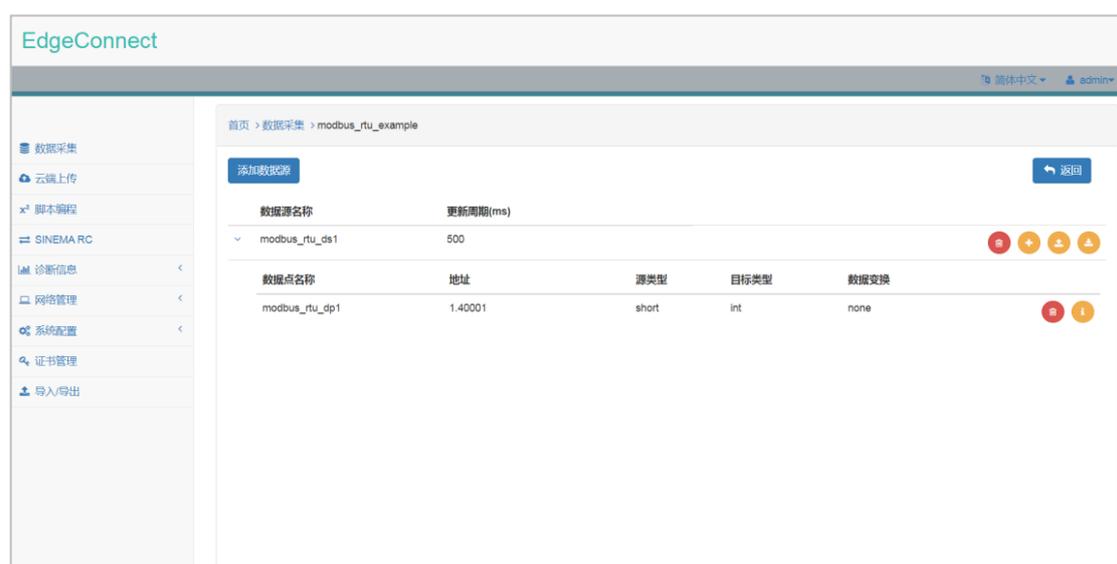
Modbus_RTU 数据点地址定义规则

Modbus_RTU 支持的数据点源类型及对应的目标类型如下表所示：

源类型	源类型说明	目标类型
bool	1 bit	bool, int
short	2 bytes, 有符号	int, double
int	4 bytes, 有符号	long, double
word	2 bytes, 无符号	int, double
dword	4 bytes, 无符号	long, double
float	4 bytes	double

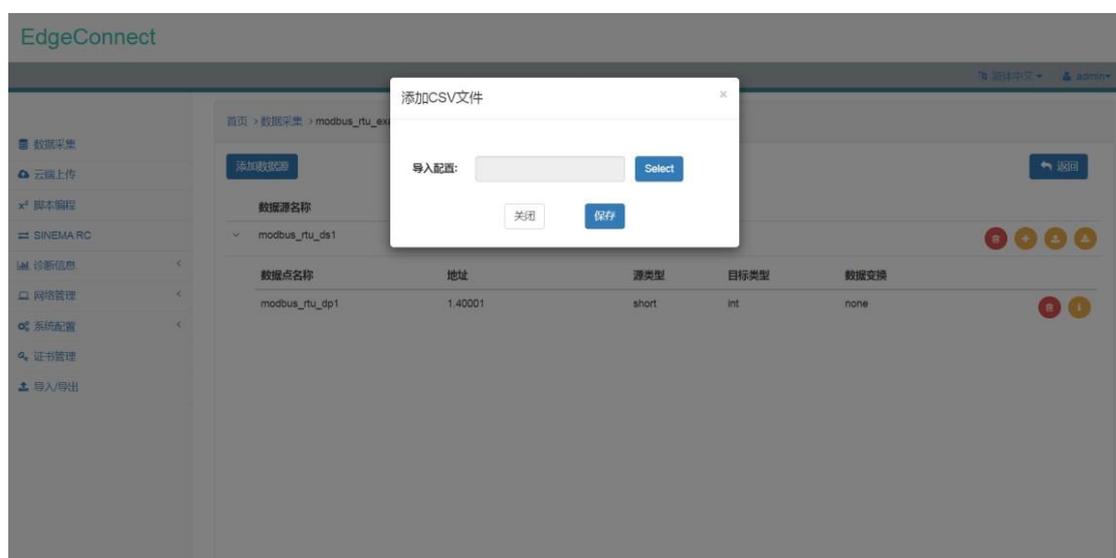
Modbus_RTU 支持源类型及对应目标类型

对于数据变换，目前支持对数据点进行线性处理。在数据点目标类型为 int、double、long 时方可选择对数据点进行线性处理。线性处理的转换公式为 $V = xp + y$ ，其中 x 为 multiplier， y 为 offset。配置完数据点信息之后，数据点信息会以列表形式显示在对应数据源中。通道配置应用下发后，点击数据点列表项右侧 ，可查看数据点当前最新值。



Modbus_RTU 数据点配置信息

点击数据源列表项右侧 ，即可打开数据点批量导入界面，用户可通过 CSV 文件批量添加数据点配置信息。



Modbus_RTU 数据点批量添加界面

点击数据源列表项右侧 ，即可以 CSV 文件形式导出数据点配置信息。

5.3.3 Modbus_TCP 通道

通道配置

在“驱动”下拉菜单中选中 Modbus_TCP，并设置相应的 IP 地址、端口号、内存布局及响应超时即可完成配置。

Modbus_TCP 通道仅支持数据读取功能。

端口号

Modbus_TCP 的端口号默认为 502。

内存布局

读取浮点数需要选择对应的内存布局，内存布局支持以下选项：

ABCD

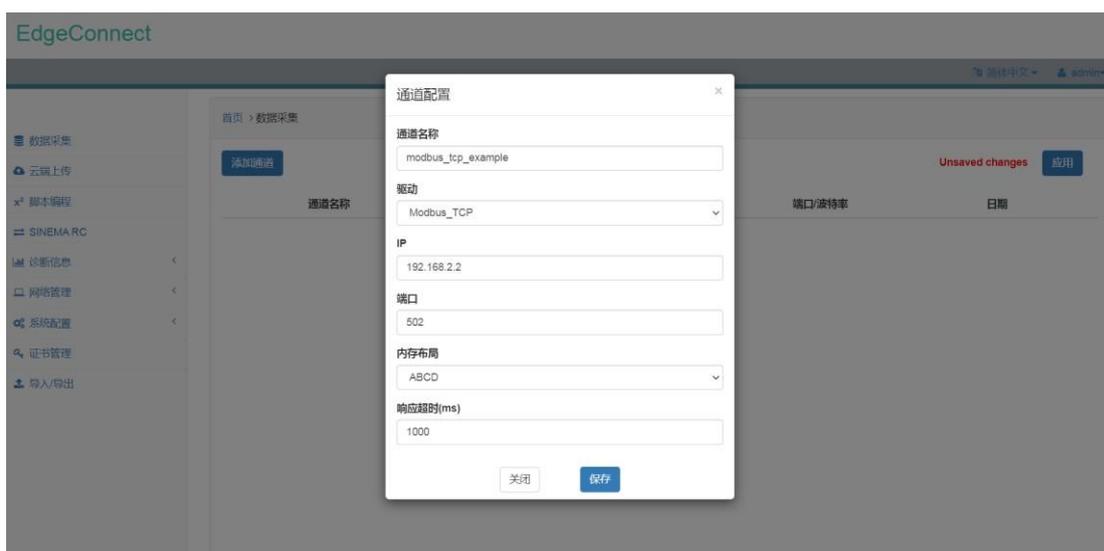
BADC

CDAB

DCBA

响应超时

响应超时时间，默认为 1000ms。



Modbus_TCP 通道配置界面

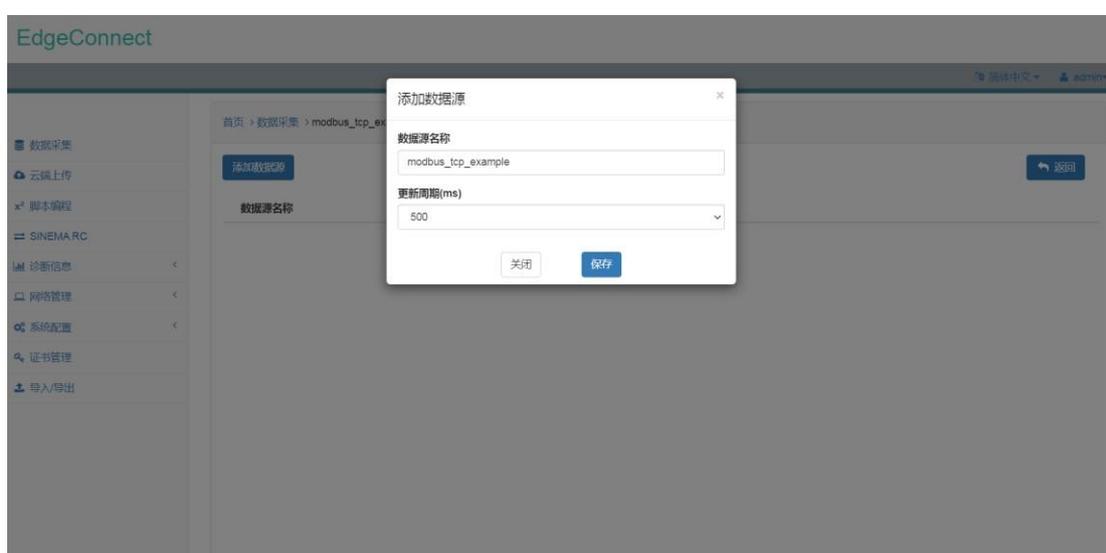
配置好 Modbus_TCP 通道之后，Modbus_TCP 通道信息会以列表形式显示在页面中。



Modbus_TCP 通道配置信息

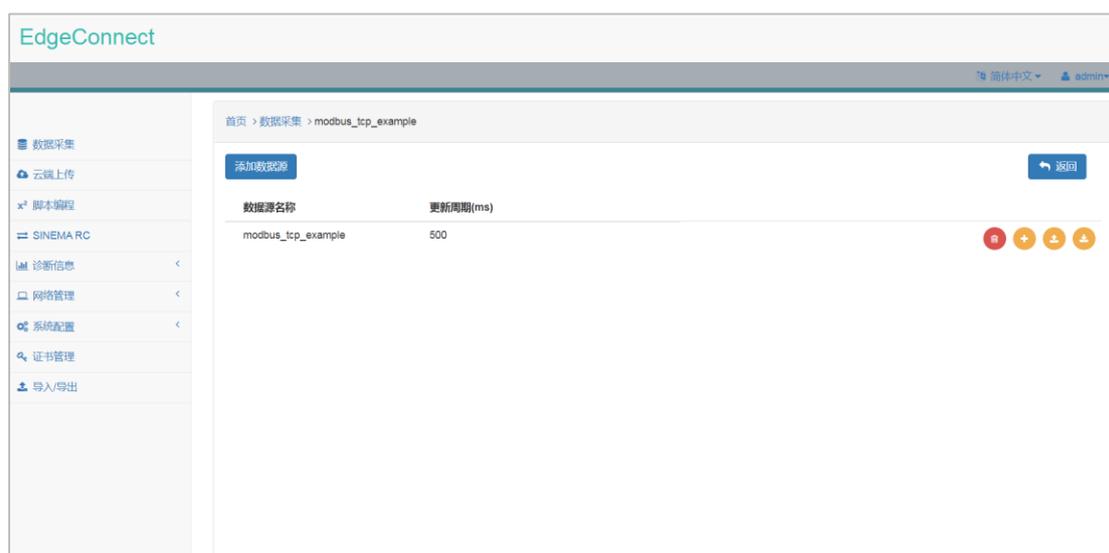
通道详情配置

点击 Modbus_TCP 通道列表项右侧 ，进入通道详情配置页面。在通道详情配置页面中，用户可以点击“添加数据源”添加数据源（数据点分组），在数据源配置界面中设置数据源名称以及更新周期，即可完成数据源配置。



Modbus_TCP 数据源配置界面

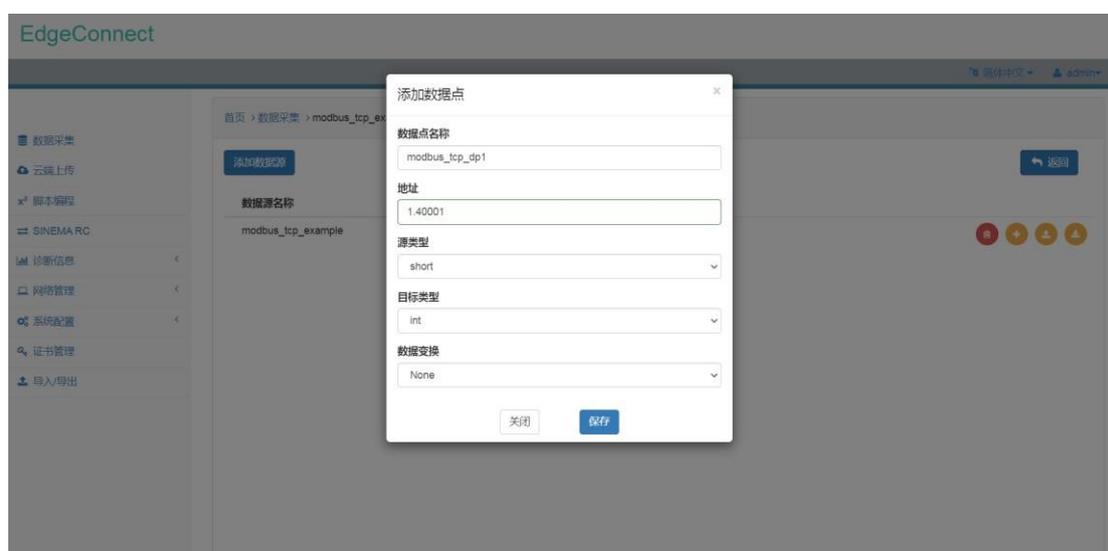
配置完数据源之后，数据源配置信息会以列表形式显示在页面中。



Modbus_TCP 数据源配置信息点击数据源

列表项右侧 \oplus ，即可打开数据点配置界面。

在数据点配置界面中，可以配置数据点名称、地址、源类型、目标类型、以及数据变换等信息。



Modbus_TCP 数据点配置界面其中，对于数据点地址需要按如下规则填写：

X.YZ.F (如1.40001 或1.30002.10)

类型	功能	范围
X	Modbus 站号	1 - 247
Y	地址区域	线圈寄存器为 0 离散输入寄存器为 1 输入寄存器为 3 保持寄存器为 4
Z	字/位偏移地址	长度为 4 位整数，若不足 4 位，高位补 0
		注意事项： 线圈寄存器及离散输入寄存器为位偏移地址 输入寄存器及保持寄存器为字偏移地址
F	位偏移地址	0 - 15 注意事项： 仅输入寄存器或保持寄存器时有效

Modbus_TCP 数据点地址定义规则

Modbus_TCP 支持的数据点源类型及对应的目标类型如下表所示：

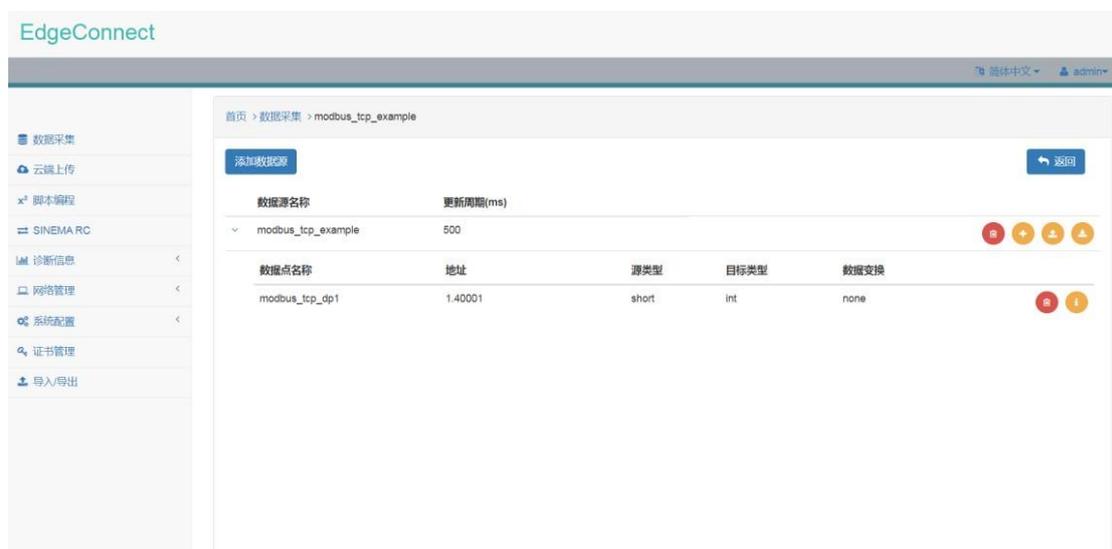
源类型	源类型说明	目标类型
bool	1 bit	bool, int
short	2 bytes, 有符号	int, double
int	4 bytes, 有符号	long, double
word	2 bytes, 无符号	int, double
dword	4 bytes, 无符号	long, double
float	4 bytes	double

Modbus_TCP 支持源类型及对应目标类型

对于数据变换，目前支持对数据点进行线性处理。在数据点目标类型为 int、double、long 时方可选择对数据点进行线性处理。线性处理的转换公式为 $V = xp + y$

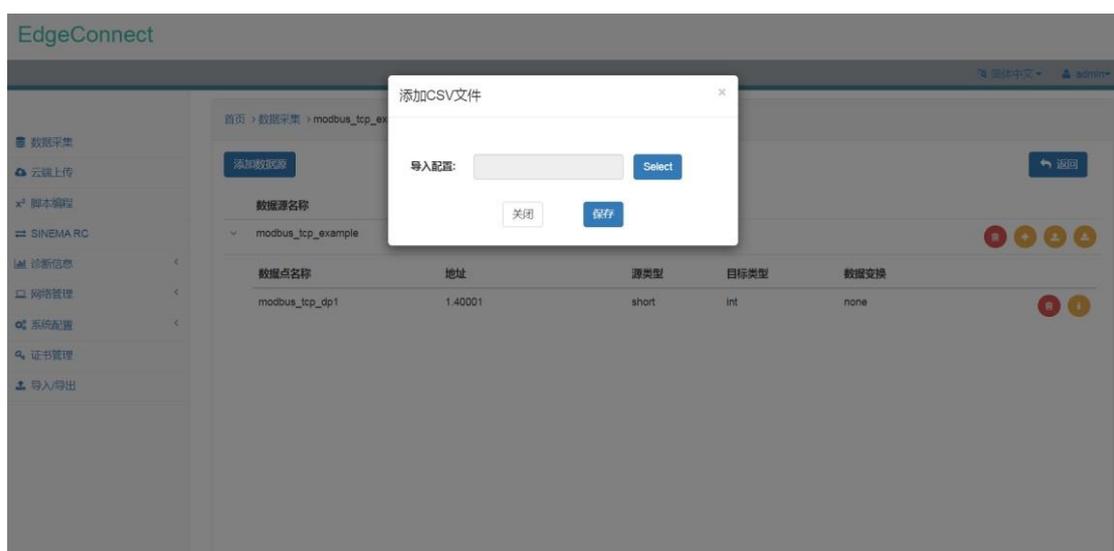
其中，x 为 multiplier，y 为 offset。

配置完数据点信息后，数据点信息会以列表形式显示在数据源中。通道配置应用下发后，点击数据点列表项右侧，可查看数据点当前最新值。



Modbus_TCP 数据点配置信息

点击数据源列表项右侧, 即可打开数据点批量导入界面，用户可通过 CSV 文件批量添加数据点配置信息。



Modbus_TCP 数据点批量添加界面

点击数据源列表项右侧, 即可以 CSV 文件形式导出数据点配置信息。

5.3.4 OPCUA 通道

通道配置

在“驱动”下拉菜单中选中 OPCUA，并设置相应的 IP 地址、端口号、验证信息（无验证或用户名/密码验证）及响应超时即可完成相应配置。OPCUA 通道仅支持数据读取功能。

1) 端口号

端口号由 OPCUA Server 确定。

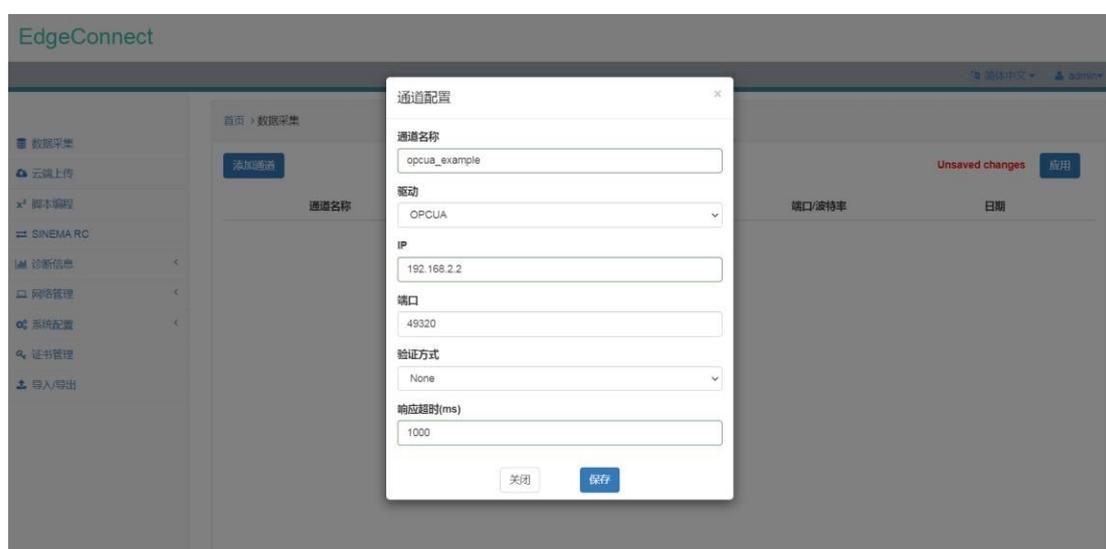
2) 验证方式

支持None（无验证）、Basic（用户名/密码验证）等两种方式。

如选择 None，无需其他操作。如选择 Basic，用户需输入用户名及密码信息。

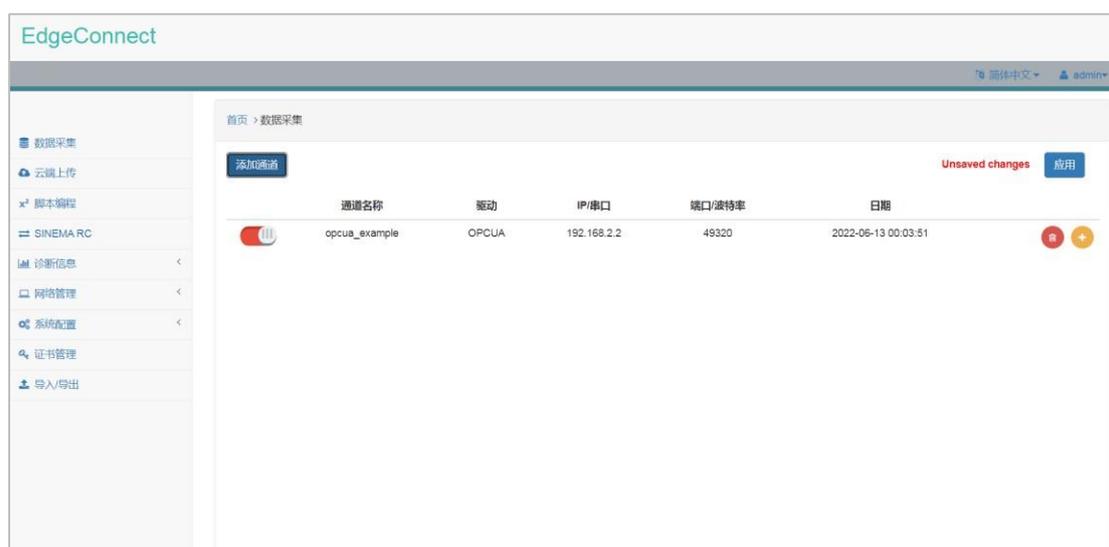
3) 响应超时

响应超时时间，默认为 1000ms。



OPCUA 通道配置界面

配置好 OPCUA 通道之后，OPCUA 通道信息会以列表形式显示在页面中。

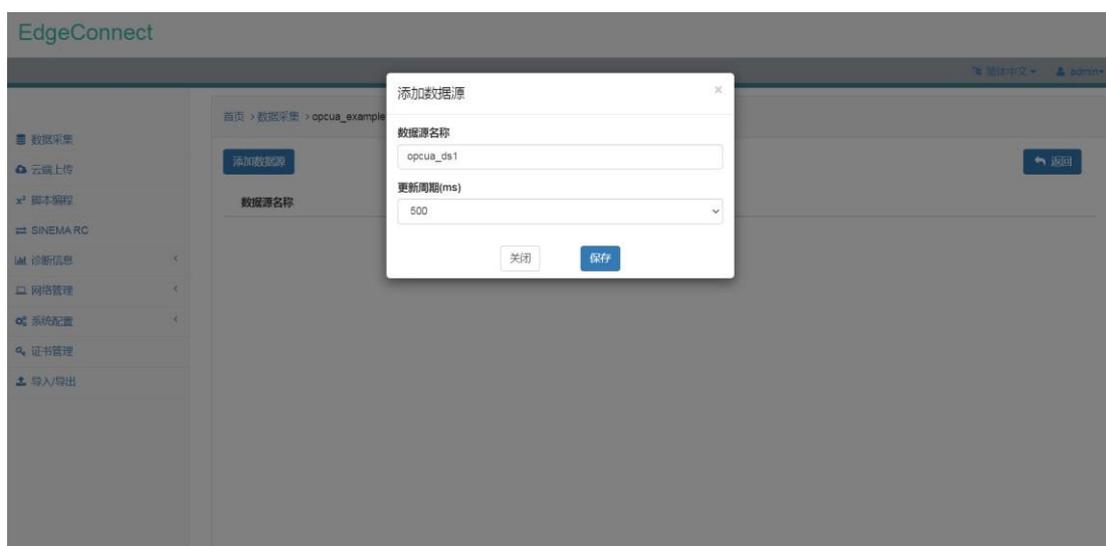


OPCUA 通道配置信息

通道详情配置

点击 OPCUA 通道列表项右侧 , 进入通道详情配置页面。

在通道详情配置页面中，用户可以点击“添加数据源”添加数据源（数据点分组），在数据源配置界面中设置数据源名称以及更新周期，即可完成数据源配置。



OPCUA 数据源配置界面

配置完数据源之后，数据源配置信息会以列表形式显示在页面中。

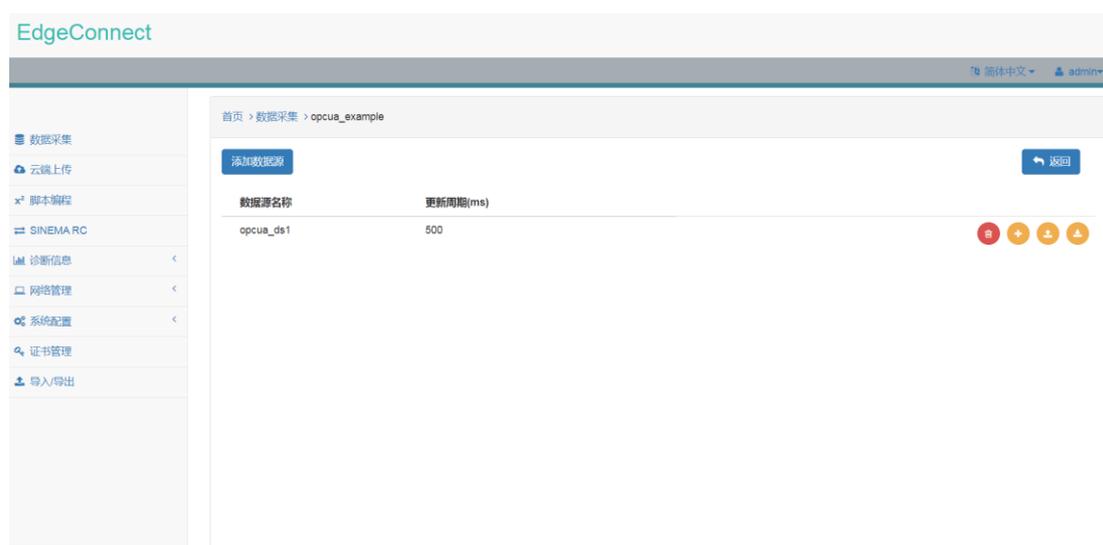


图4.26 OPCUA 数据源配置信息

点击数据源列表项右侧 \oplus ，即可打开数据点配置界面。

在数据点配置界面中，可以配置数据点名称、地址、源类型、目标类型、以及数据变换等信息。

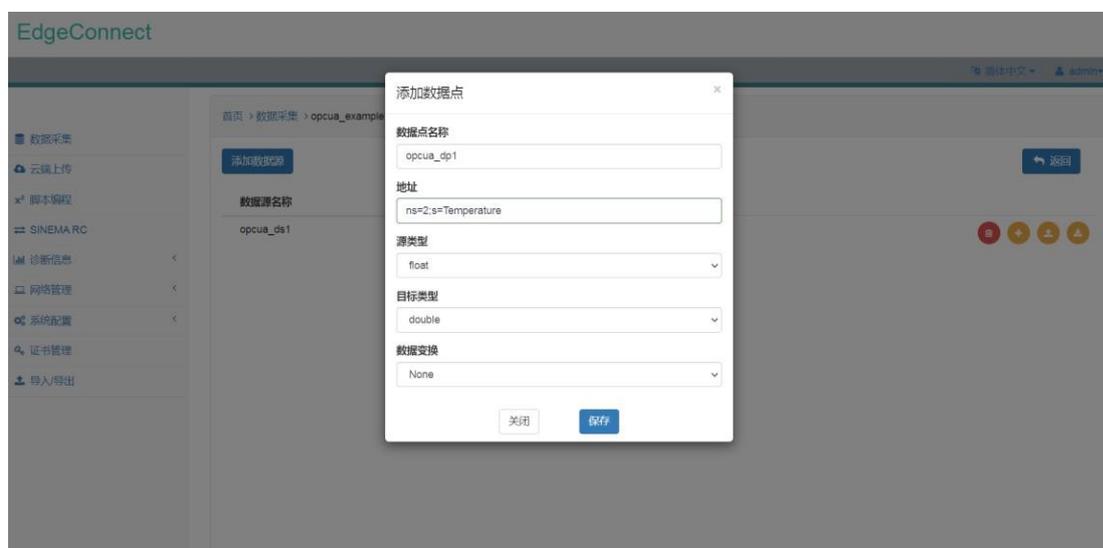


图 4.27 OPCUA 数据点配置界面

其中，数据点的地址支持如下两种格式：

地址格式	说明
ns=x;s=<string>	x 为 NamespaceIndex, <string>为字符串类型变量地址
ns=x;i=y	x 为 NamespaceIndex, y 为整型变量地址

表 4.7 OPCUA 数据点地址定义规则

OPCUA 支持的数据点源类型及对应的目标类型如下表所示：

源类型	源类型说明	OPCUA 内置类型	目标类型
bool	1 byte	Boolean	bool, int
byte	1 byte, 无符号	Byte	int, double
word	2 bytes, 无符号	UInt16	int, double
dword	4 bytes, 无符号	UInt32	int, double
char	1 byte, 有符号	SByte	int, double
short	2 bytes, 有符号	Int16	int, double
Int	4 bytes, 有符号	Int32	int, double
long	8 bytes, 有符号	Int64	long
float	4 bytes, 有符号	float	double
string		String	string

表 4.8 OPCUA 支持源类型及对应目标类型

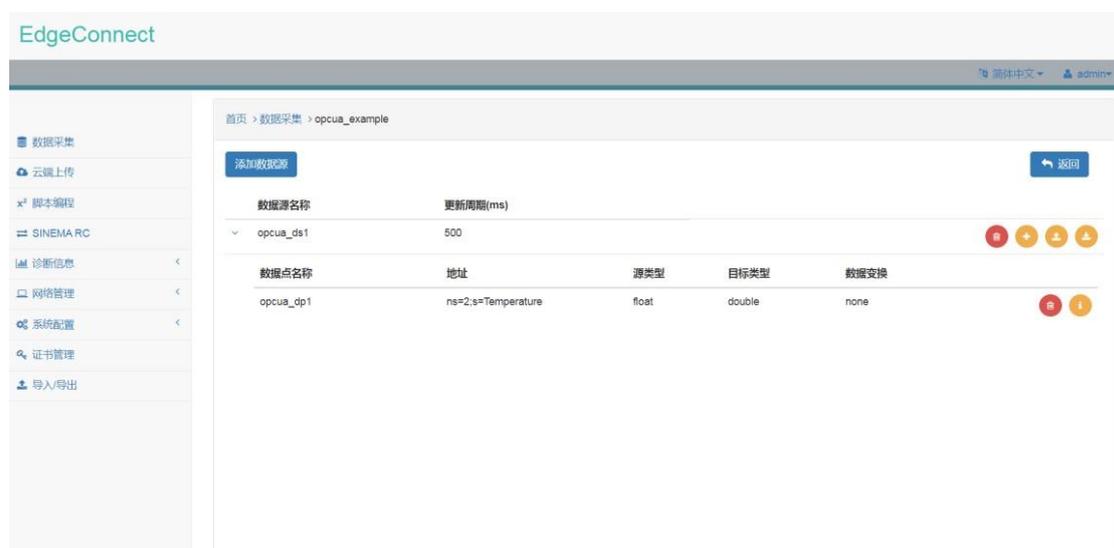
对于数据预处理，目前支持对数据点进行线性处理。在数据点目标类型为 int、double、long 时方可选择对数据点进行线性处理。

线性处理的转换公式为：

$$V = xp + y$$

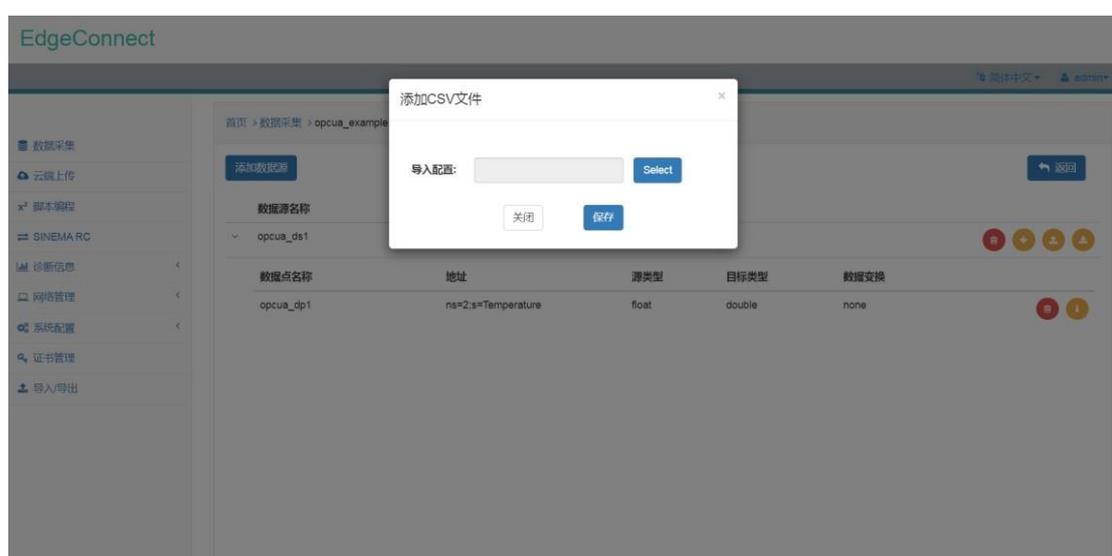
其中，x 为 multiplier，y 为 offset。

配置完数据点信息之后，数据点信息会以列表形式显示在数据源中。通道配置应用下发后，点击数据点列表项右侧，可查看数据点当前最新值。



OPCUA 数据点配置信息

点击数据源列表项右侧, 即可打开数据点批量导入界面，用户可以通过 CSV 文件批量添加数据点配置信息。



OPCUA 数据点批量添加界面

点击数据源列表项右侧, 即可以 CSV 文件形式导出数据点配置信息。

5.3.5 Siemens_S7 通道

通道配置

在“驱动”下拉菜单中选中 Siemens_S7，并设置 PLC 类型、IP 地址、端口号、机架号、槽号及响应超时即可完成相应配置。

Siemens_S7 通道同时支持数据读取及写入功能。PLC 类型

- S7-200
- S7-200Smart
- S7-300
- S7-400
- S7-1200
- S7-1500

1) 端口号

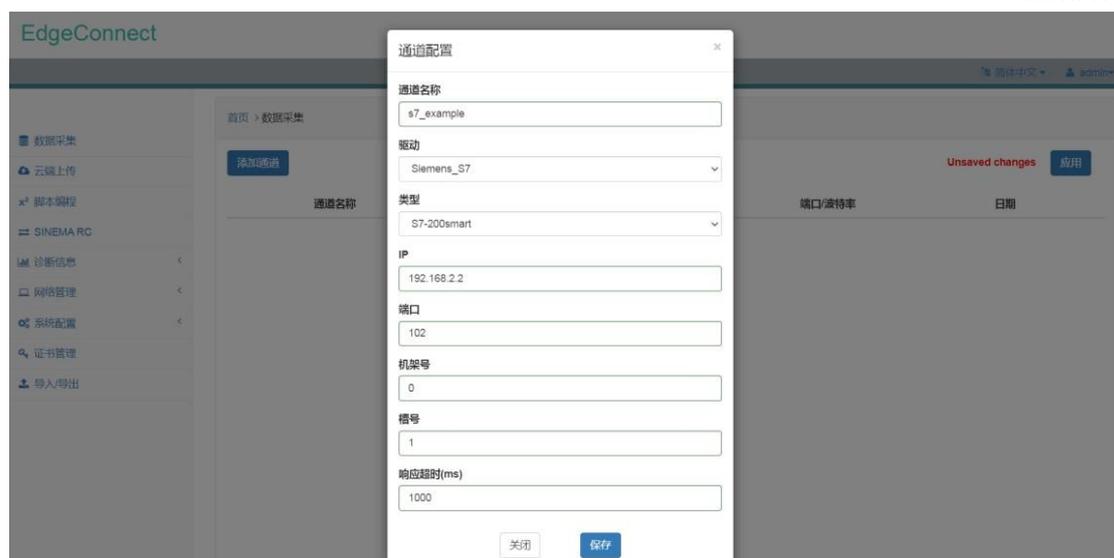
Siemens_S7 的端口号默认为 102。

2) 机架号及槽号

根据 S7 PLC 组态填写机架号及槽号，机架号默认为 0，槽号默认为 1。

3) 响应超时

响应超时时间，默认为 1000ms



Siemens_S7 通道配置界面

配置好 Siemens_S7 通道之后，Siemens_S7 通道信息会以列表形式显示在页面中。



Siemens_S7 通道配置信息

支持的 S7 设备类型

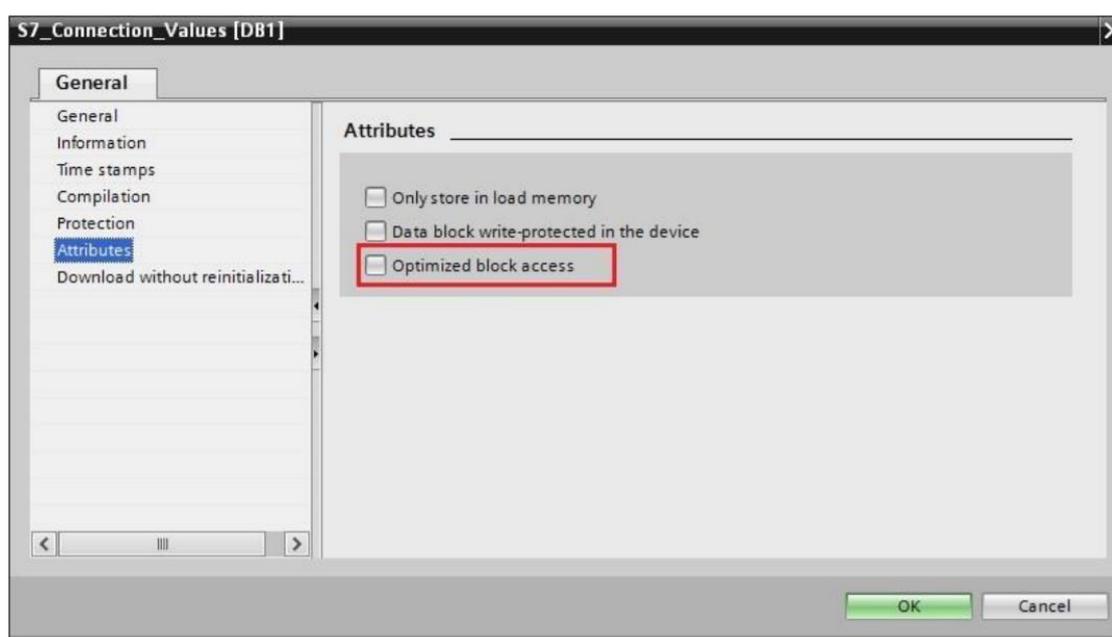
- S7-200
- S7-200Smart
- S7-300
- S7-400
- S7-1200
- S7-1500

PLC 通信配置

对于 S7-1200 / S7-1500 系列 PLC，为确保 S7 正常通信，需要在 TIA Portal 进行如下配置。

1) 取消数据块优化

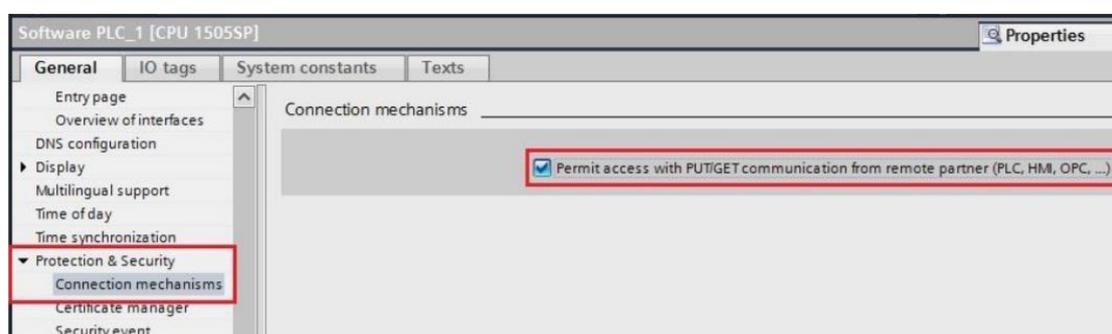
在数据块的属性中进行设置，请确保**没有**勾选该选项。



取消数据块优化

2) 启用PUT/GET 通信

必须在 CPU 配置中启用PUT/GET 通信，请确保**勾选**该选项。

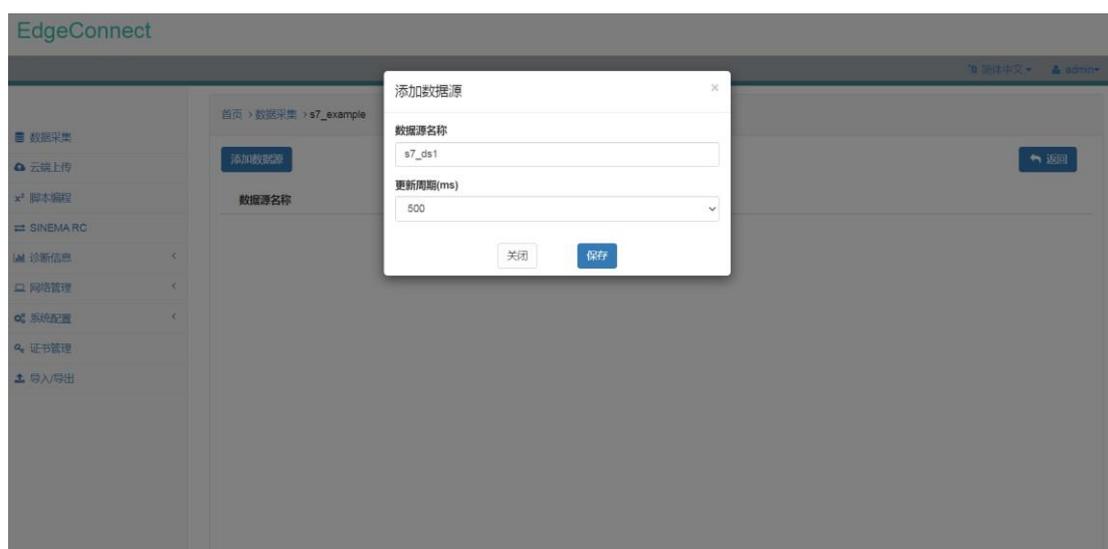


启用 PUT/GET 通信

通道详情配置

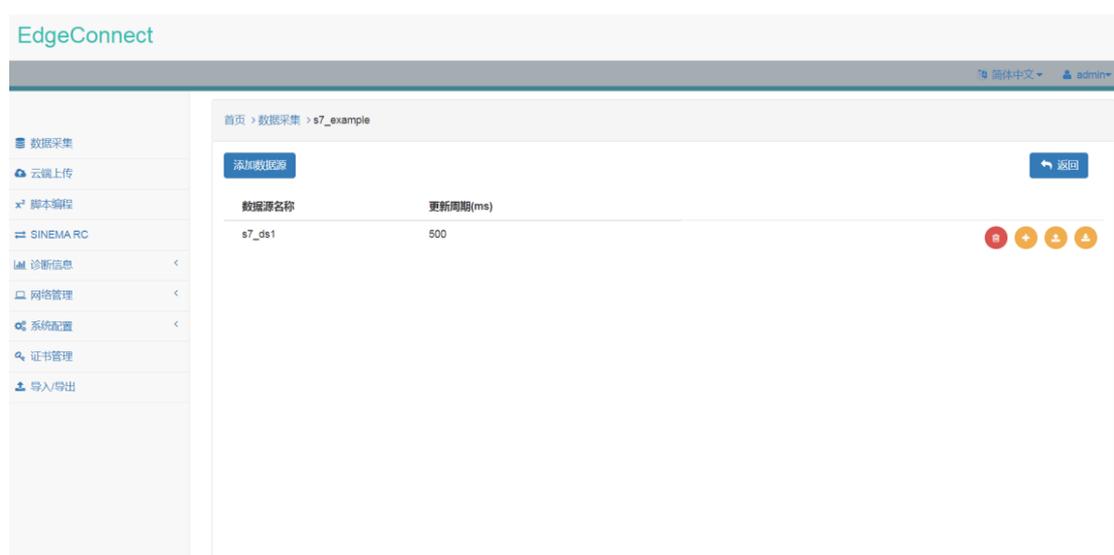
点击 Siemens_S7 通道列表项右侧 , 进入通道详情配置页面。

在通道详情配置页面中, 用户可以点击“添加数据源”添加数据源 (数据点分组), 在数据源配置界面中设置数据源名称以及更新周期, 即可完成数据源配置。



Siemens_S7 数据源配置界面

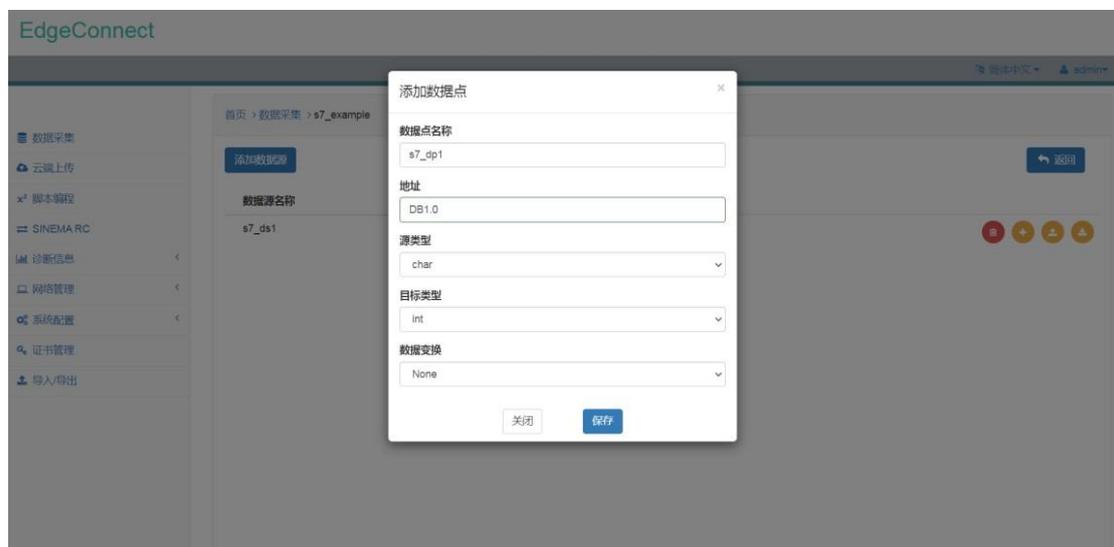
配置完数据源之后, 数据源配置信息会以列表形式显示在页面中。



Siemens_S7 数据源配置信息

点击数据源列表项右侧 \oplus ，即可打开数据点配置界面。

在数据点配置界面中，可以配置数据点名称、地址、源类型、目标类型、以及数据变换等信息。



Siemens_S7 数据点配置界面

其中，数据点地址定义如下表所示：

区域	地址格式
I	Ix, 其中 x 为字节偏移量, 如 I0
	Ix.y, 其中 x 为字节偏移量, y 为位偏移量, y 取值范围为 0 ~ 7, 如 I0.0
Q	Qx, 其中 x 为字节偏移量, 如 Q0
	Qx.y, 其中 x 为字节偏移量, y 为位偏移量, y 取值范围为 0 ~ 7, 如 Q0.0
M	Mx, 其中 x 为字节偏移量, 如 M0
	Mx.y, 其中 x 为字节偏移量, y 为位偏移量, y 取值范围为 0 ~ 7, 如 M0.0
DB 对于 S7-200smart, V 区即为 DB1	DBx.y, 其中 x 为 DB 偏移量, y 为字节偏移量, 如 DB1.0
	DBx.y.z, 其中 x 为 DB 偏移量, y 为字节偏移量, 对于 string 源类型变量, z 为字符串的长度; 对于 bool 源类型变量, z 为位偏移量, 取值范围为 0 ~ 7, 如 DB1.0.1

Siemens_S7 数据点地址定义规则

Siemens_S7通道支持的数据点源类型及对应的目标类型

如下表所示：

源类型	源类型说明	S7 内置类型	目标类型
bool	1 bit	Bool	bool, int
byte	1 byte, 无符号	Byte / USint	int, double
word	2 bytes, 无符号	Word / UInt	int, double
dword	4 bytes, 无符号	DWord / UInt	int, double
char	1 byte, 有符号	Char	int, double
int	2 bytes, 有符号	Int	int, double
dint	4 bytes, 有符号	Dint	int, double
float	4 bytes	Real	double
string	1- 254 bytes		string

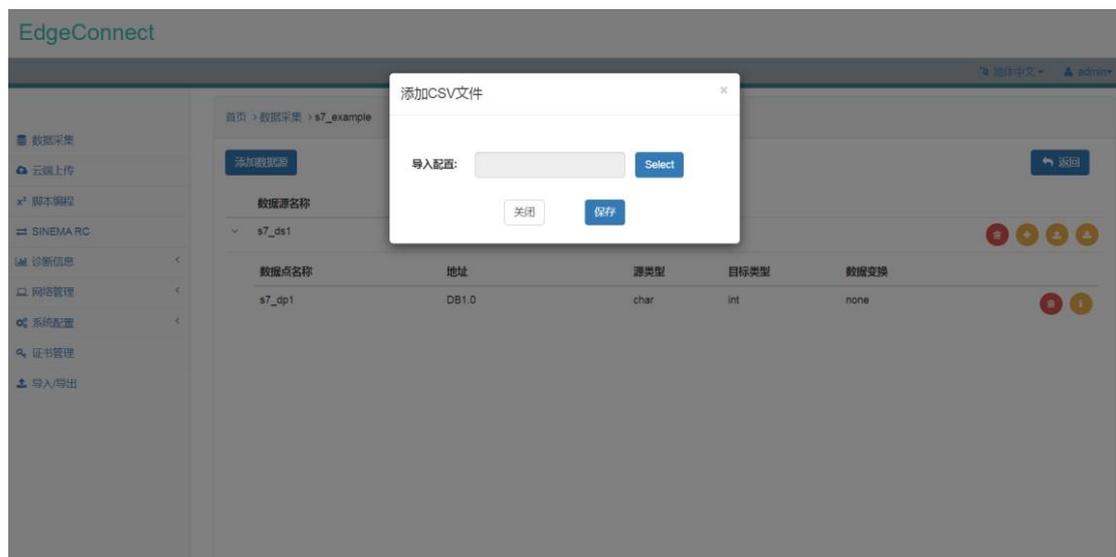
Siemens_S7 支持源类型及对应目标类型

对于数据变换，目前支持对数据点进行线性处理。在数据点目标类型为 int、double、long 时方可选择对据点进行线性处理。线性处理的转换公式为 $V = xp + y$ 其中，x 为 multiplier，y 为 offset。配置完数据点信息之后，数据点信息会以列表形式显示在数据源中。通道配置应用下发后，点击数据点列表项右侧 ，可查看数据点当前最新值。



图 4.37 Siemens_S7 数据点配置信息

点击数据源列表项右侧, 即可打开数据点批量导入界面, 用户可以通过 CSV 文件批量添加数据点配置信息。



Siemens_S7 数据点批量添加界面

点击数据源列表项右侧, 即可以 CSV 文件形式导出数据点配置信息。

5.3.6 Mitsubishi_MC_QNA3E 通道

通道配置

在“驱动”下拉菜单中选中 Mitsubishi_MC_QNA3E，并填入 IP 地址、端口号及响应超时即可完成相应配置。

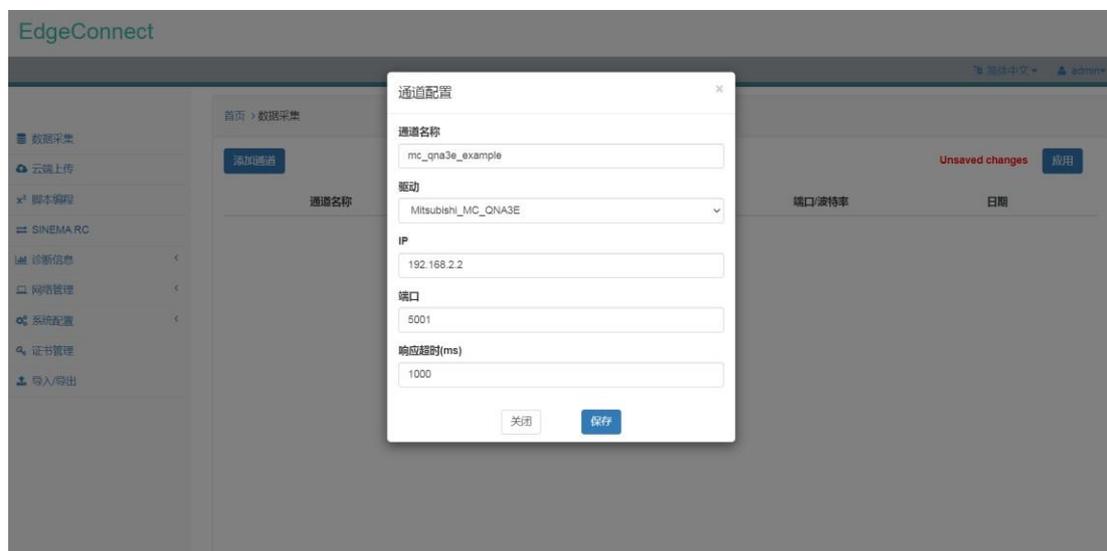
Mitsubishi_MC_QNA3E 通道仅支持数据读取功能。

1) 端口号

端口号由 PLC 通信配置确定。

2) 响应超时

响应超时时间，默认为 1000ms。



Mitsubishi_MC_QNA3E 通道配置界面

配置好 Mitsubishi_MC_QNA3E 通道之后，Mitsubishi_MC_QNA3E 通道信息会以列表形式显示在页面中。



Mitsubishi_MC_QNA3E 通道配置信息

支持的 MC 设备类型

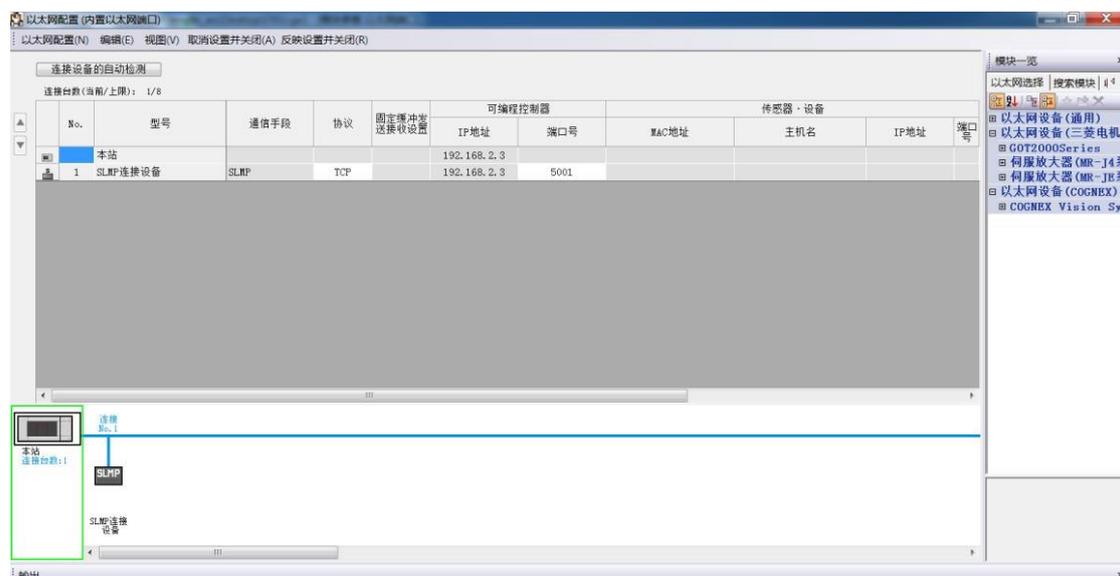
- FX5U
- Q Series
- L Series

PLC 通信配置

为确保 MC_QNA3E 正常通信，MELSOFT GX Works3 需要在参数->CPU->模块参数->以太网端口->对象设备连接配置设置中进行如下设置：

添加 SLMP 连接

从右侧以太网设备（通用）中拖动 SLMP 连接设备至配置区域，协议设置为 TCP，端口号可自定义设置。

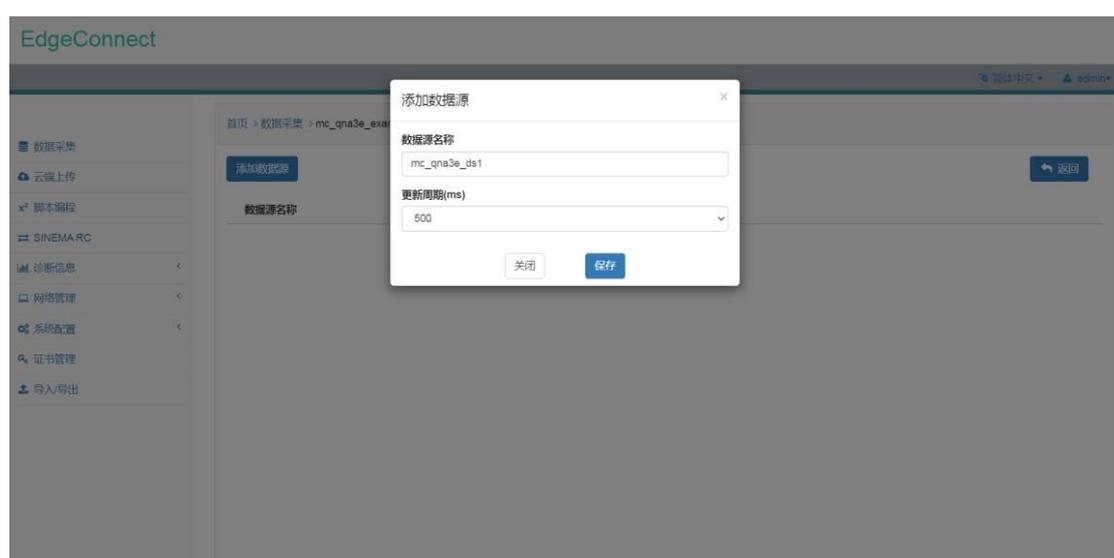


添加 SLMP 连接配置

通道详情配置

点击 Mitsubishi_MC_QNA3E 通道列表项右侧 , 进入通道详情配置页面。

在通道详情配置页面中, 用户可以点击“添加数据源”添加数据源 (数据点分组), 在数据源配置界面中设置数据源名称以及更新周期, 即可完成数据源配置。



Mitsubishi_MC_QNA3E 数据源配置界面

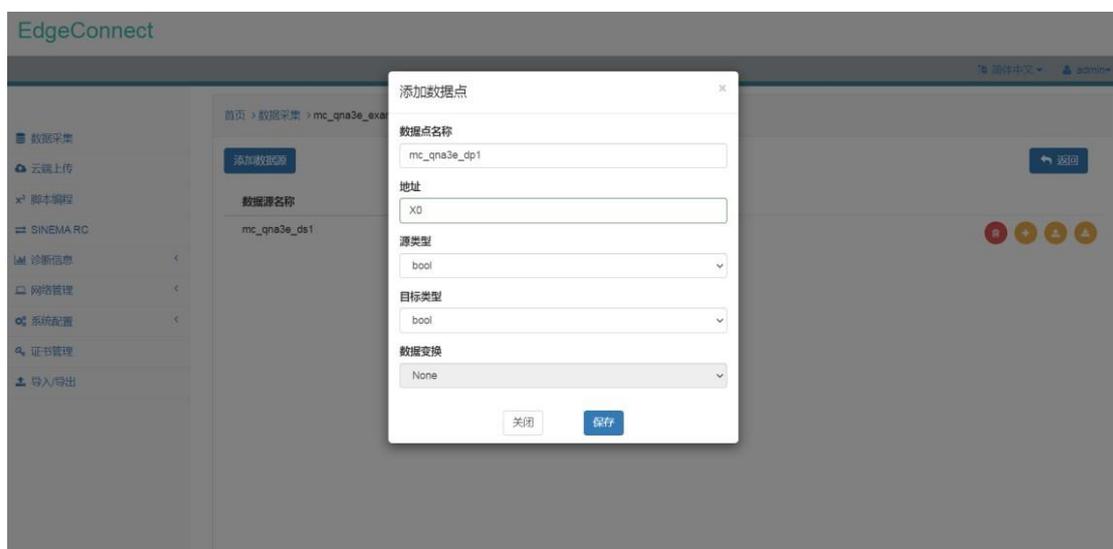
配置完数据源之后，数据源配置信息会以列表形式显示在页面中。



Mitsubishi_MC_QNA3E 数据源配置信息

点击数据源列表项右侧 \oplus ，即可打开数据点配置界面。

在数据点配置界面中，可以配置数据点名称、地址、源类型、目标类型、以及数据变换等信息。



Mitsubishi_MC_QNA3E 数据点配置界面

其中，数据点地址定义如下表所示：

区域	地址格式
X	Xx, 其中x为位偏移量, 如X0
Y	Yx, 其中x为位偏移量, 如Y0
M	Mx, 其中x为位偏移量, 如M0
D	Dx, 其中x为字节偏移量, 如D0
	Dx.y, 其中x为字节偏移量, 对于string源类型变量, y为字符串的长度; 对于bool类型变量, y为位偏移量, 取值范围为0~15, 如D1.1

Mitsubishi_MC_QNA3E 数据点地址定义规则

Mitsubishi_MC_QNA3E 通道支持的数据点源类型及对应的目标类型如下表所示：

源类型	源类型说明	目标类型
bool	1 bit	bool, int
short	2 bytes, 有符号	int, double
word	2 bytes, 无符号	int, double
float	4 bytes	double
string		string

表 4.12 Mitsubishi_MC_QNA3E 支持源类型及对应目标类型

对于数据变换，目前支持对数据点进行线性处理。在数据点目标类型为 int、double、long 时方可选择对数据点进行线性处理。

线性处理的转换公式为 $V = xp + y$

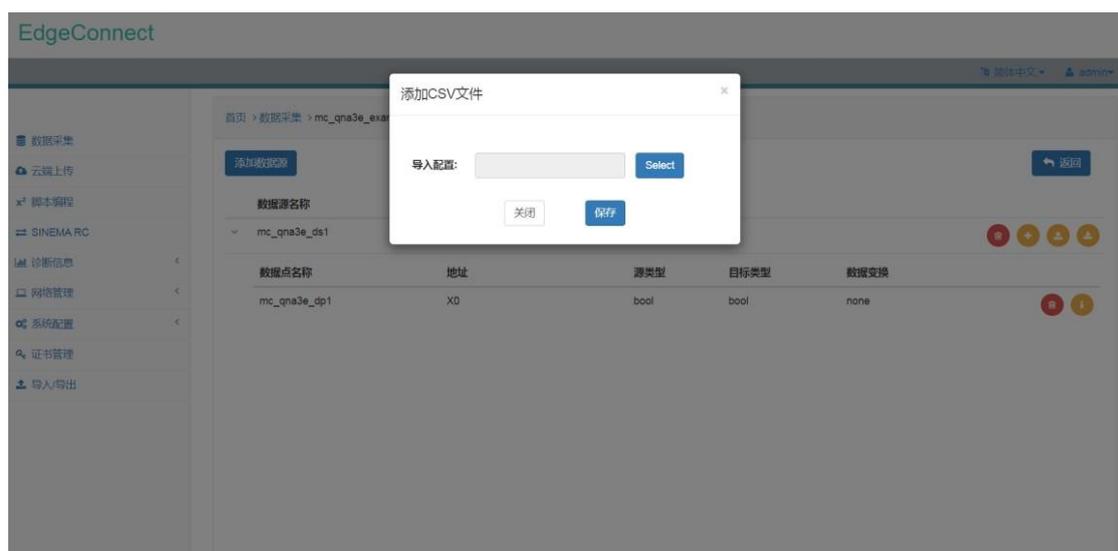
其中，x 为 multiplier，y 为 offset。

配置完数据点信息之后，数据点信息会以列表形式显示在数据源中。通道配置应用下发后，点击数据点列表项右侧，可查看数据点当前最新值。



Mitsubishi_MC_QNA3E 数据点配置信息

点击数据源列表项右侧 , 即可打开数据点批量导入界面, 用户可通过 CSV 文件批量添加数据点配置信息。



Mitsubishi_MC_QNA3E 数据点批量添加界面

点击数据源列表项右侧 , 即可以 CSV 文件形式导出数据点配置信息。

5.3.7 Mitsubishi_MC_A1E 通道

通道配置

在“驱动”下拉菜单中选中 Mitsubishi_MC_A1E，并设置 IP 地址、端口号及响应超时即可完成相应配置。

Mitsubishi_MC_A1E 通道仅支持数据读取功能。

端口号:端口号由 PLC 通信配置确定。

响应超时:响应超时时间，默认为 1000ms。

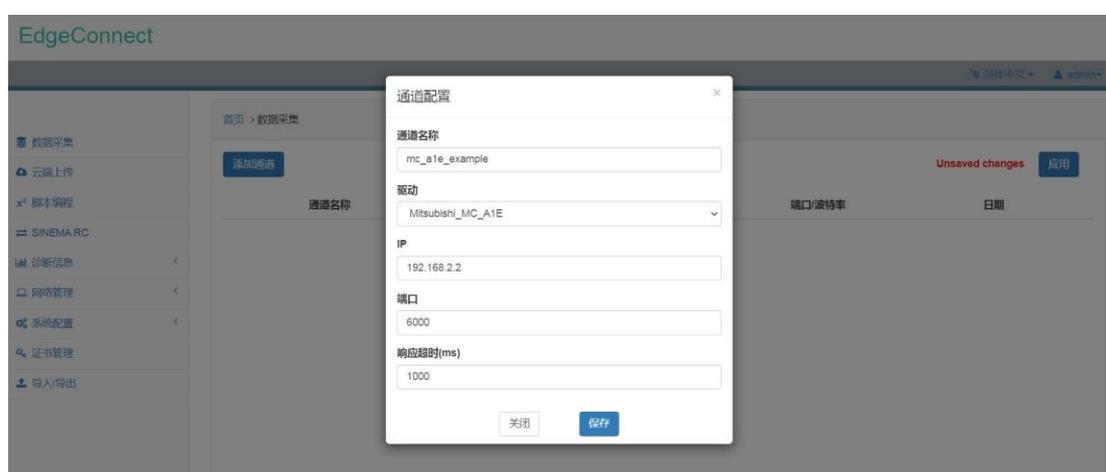


图 4.47 Mitsubishi_MC_A1E 通道配置界面

配置好 Mitsubishi_MC_A1E 通道之后，Mitsubishi_MC_A1E 通道信息会以列表形式显示在页面中。



Mitsubishi_MC_A1E 通道配置信息

支持的MC 设备类型

- FX3U + FX3U_ENET_ADAP 模块

PLC 通信配置

为确保 MC_QNA1E 正常通信，MELSOFT GX Works2 需要在参数->PLC 参数->以太网端口设置中进行如下设置：

1) 选择二进制码通信

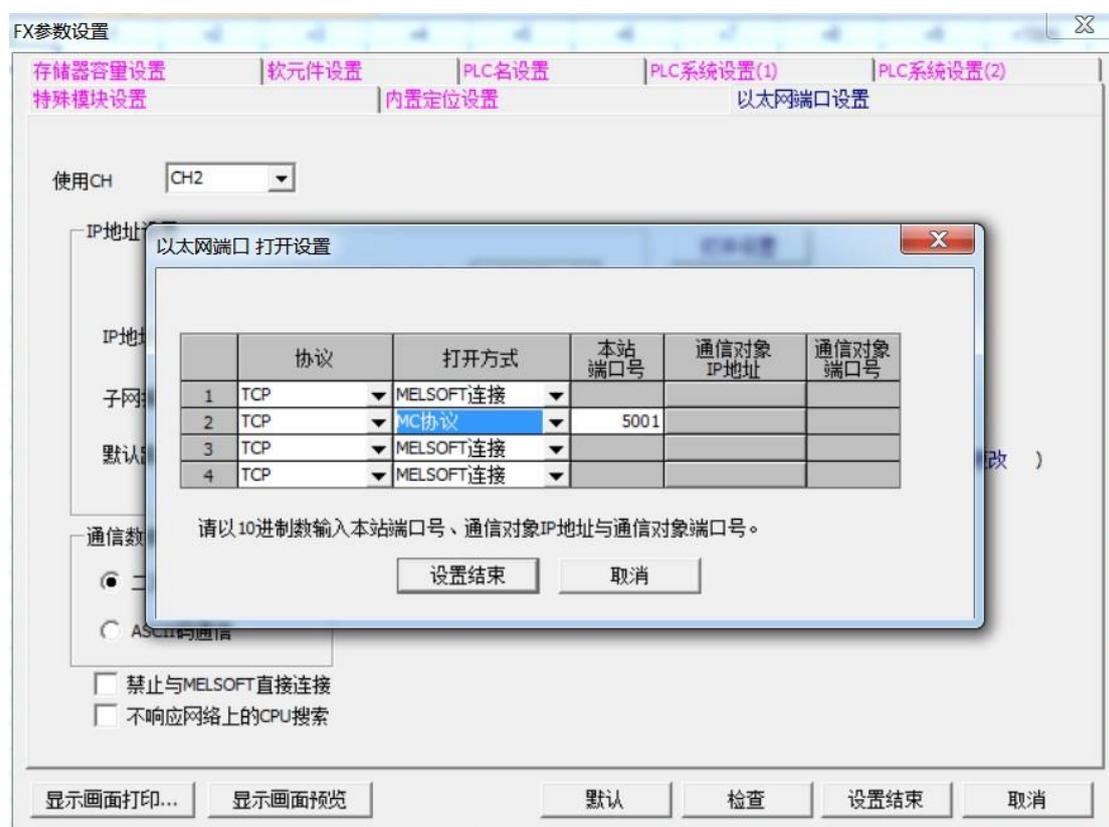
通信数据代码设置中选择二进制码通信。



选择二进制码通信

2) 添加 MC 协议连接

协议选择 TCP，打开方式选择 MC 协议，端口号用户可自定义。



添加 MC 协议连接

通道详情配置

点击 Mitsubishi_MC_A1E 通道列表项右侧 , 进入通道详情配置页面。在通道详情配置页面中，用户可以点击“添加数据源”添加数据源（数据点分组），在数据源配置界面中设置数据源名称以及更新周期，即可完成数据源配置。

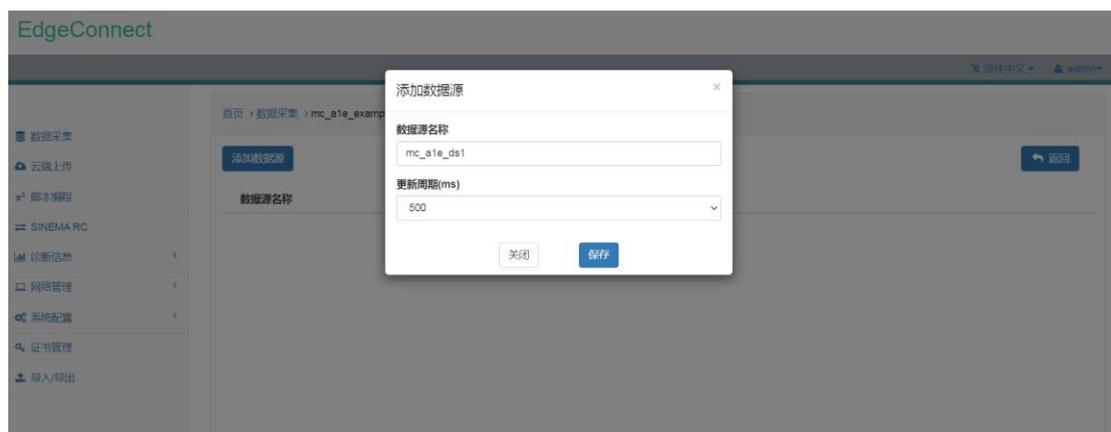


图 4.51 Mitsubishi_MC_A1E 数据源配置界面

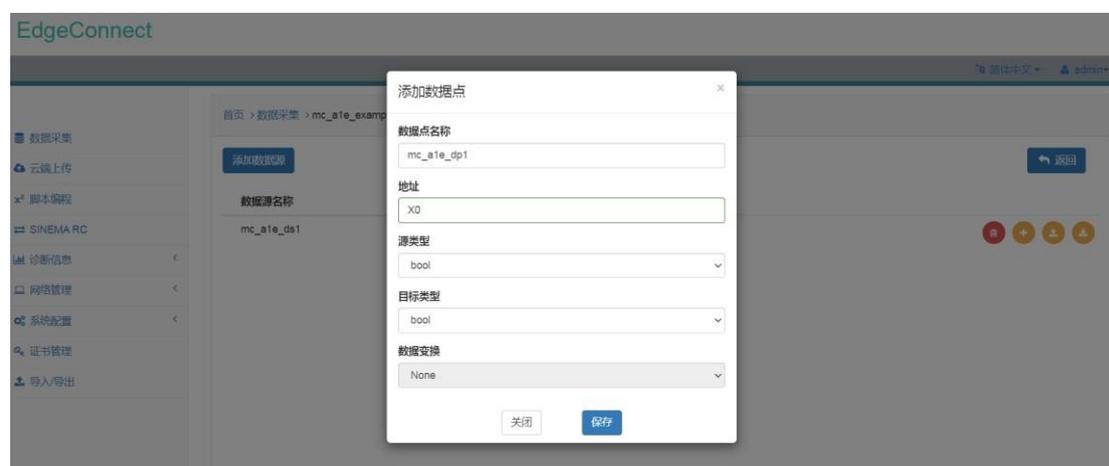
配置完数据源之后，数据源配置信息会以列表形式显示在页面中。



图4.52 Mitsubishi_MC_A1E 数据源配置信息

点击数据源列表项右侧 \oplus ，即可打开数据点配置界面。

在数据点配置界面中，可以配置数据点名称、地址、源类型、目标类型、以及数据变换等信息。



Mitsubishi_MC_A1E 数据点配置界面

其中，数据点地址定义如下表所示：

区域	地址格式
X	Xx, 其中x为位偏移量, 如X0
Y	Yx, 其中x为位偏移量, 如Y0
M	Mx, 其中x为位偏移量, 如M0
D	Dx, 其中x为字节偏移量, 如D0
	Dx.y, 其中x为字节偏移量, 对于string源类型变量, y为字符串的长度; 对于bool类型变量, y为位偏移量, 取值范围为0~15, 如D1.1

表 4.13 Mitsubishi_MC_A1E 数据点地址定义规则

Mitsubishi_MC_A1E 通道支持的数据点源类型及对应的目标类型如下表所示：

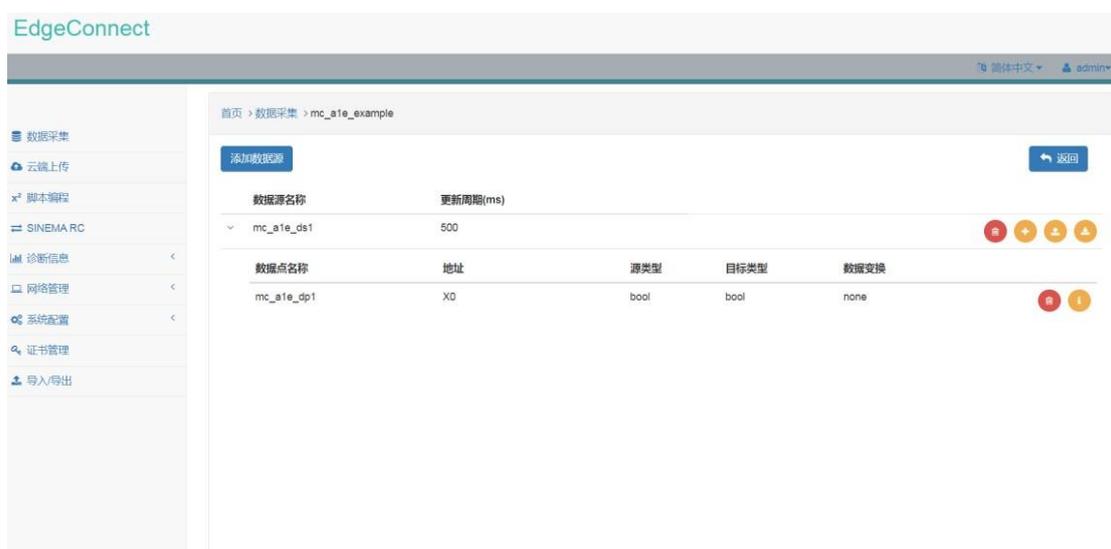
源类型	源类型说明	目标类型
bool	1 bit	bool, int
short	2 bytes, 有符号	int, double
word	2 bytes, 无符号	int, double
float	4 bytes	double
string		string

表 4.14 Mitsubishi_MC_A1E 支持源类型及对应目标类型

对于数据变换，目前支持对数据点进行线性处理。在数据点目标类型为 int、double、long 时方可选择对数据点进行线性处理。线性处理的转换公式为 $V = xp + y$

其中，x 为 multiplier，y 为 offset。

配置完数据点信息之后，数据点信息会以列表形式显示在数据源中。通道配置应用下发后，点击数据点列表项右侧，可查看数据点当前最新值。



Mitsubishi_MC_A1E 数据点配置信息

点击数据源列表项右侧 , 即可打开数据点批量导入界面, 用户可以通过 CSV 文件批量添加数据点配置信息。

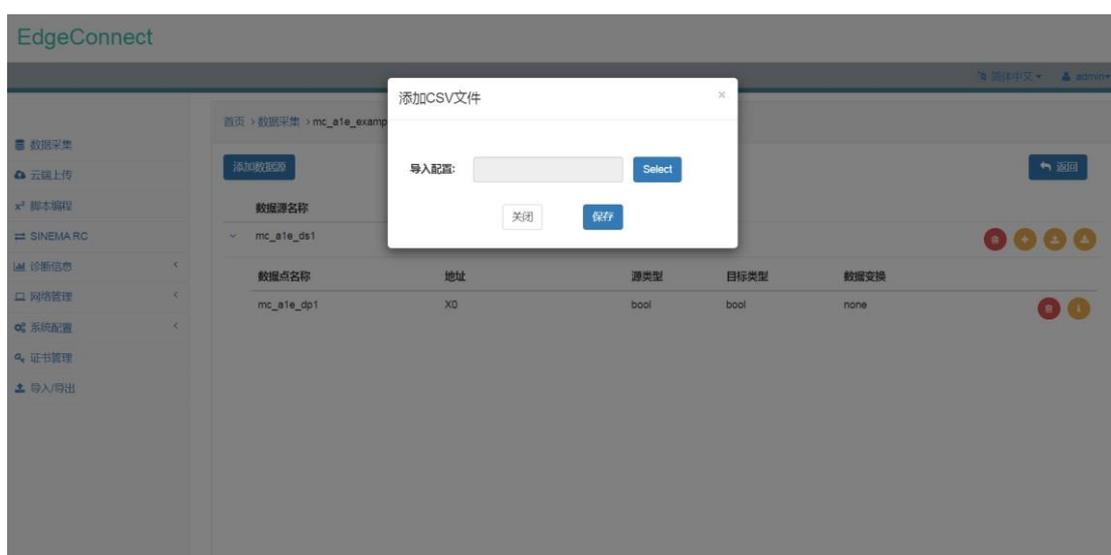


图 4.55 Mitsubishi_MC_A1E 数据点批量添加界面

点击数据源列表项右侧 , 即可以 CSV 文件形式导出数据点配置信息。

5.3.8 Omron_FINS_TCP 通道

通道配置

在“驱动”下拉菜单中选中 Omron_FINS_TCP，并设置 IP 地址、端口号、SA1、DA1 及响应超时即可完成相应配置。

Omron_FINS_TCP 通道仅支持数据读取功能。

1) 端口号：

Omron_FINS_TCP 的端口号默认为 9600。

2) SA1

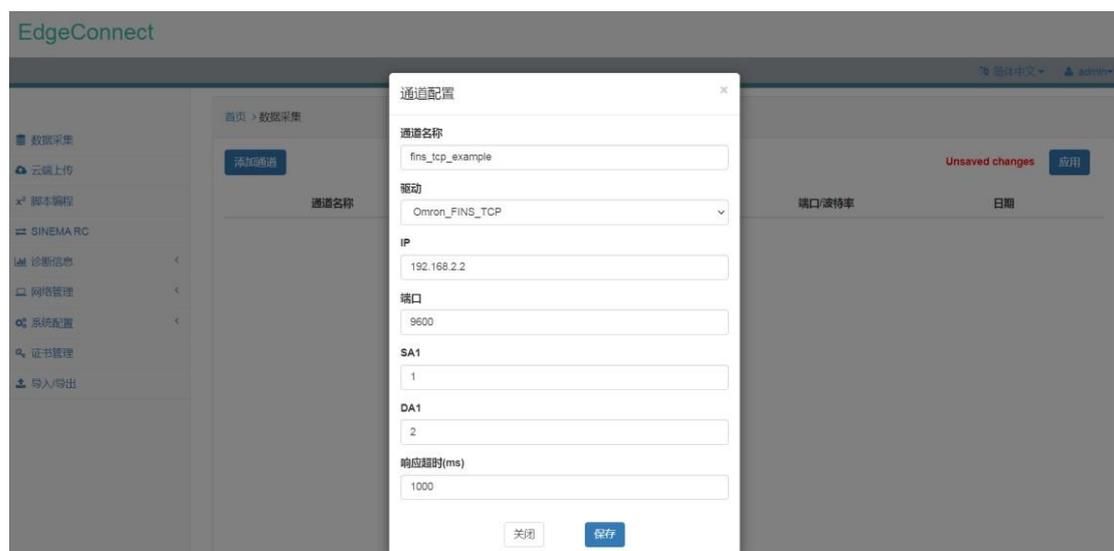
SA1 为设备本机 IP 地址的最后一位，如 IP 地址为 192.168.2.1，则 SA1 为 1。

3) DA1

DA1 为目标设备 IP 地址的最后一位，如 IP 地址为 192.168.2.2，则 DA1 为 2。

4) 响应超时

响应超时时间，默认为 1000ms。



Omron_FINS_TCP 通道配置界面

配置好 Omron_FINS_TCP 通道之后，Omron_FINS_TCP 通道信息会以列表形式显示在页面中。



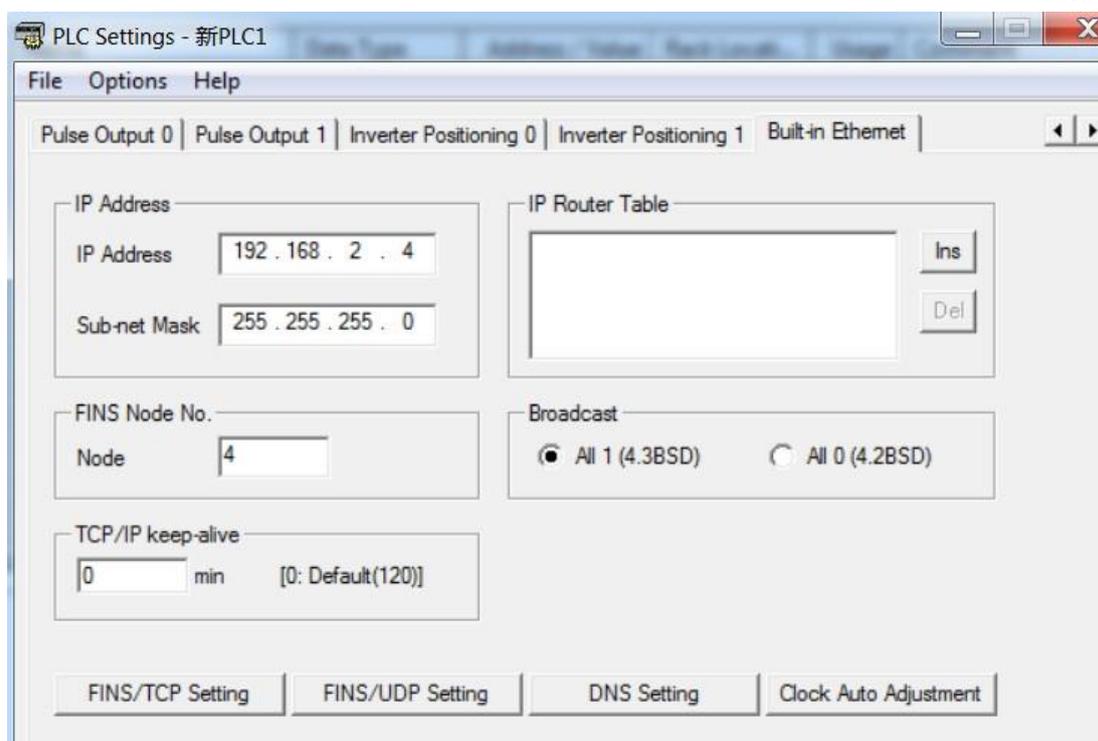
图 4.57 Omron_FINS_TCP 通道配置信息

支持的 Omron 设备类型

- CP Series
- CJ Series
- CS Series
- C200

PLC 通信配置

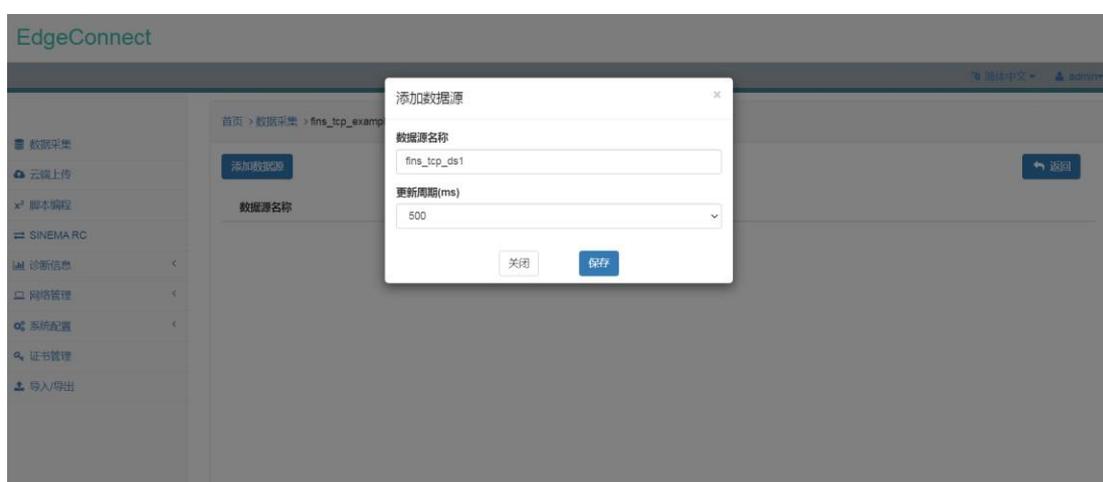
为确保 FINS_TCP 正常通信，CX-Programmer 需要在 CPU->Setting->Built-in Ethernet 进行如下设置：设置 FINS Node No.将 FINS Node No.配置中 Node 的值设置为 PLC IP 地址的最后一位。



设置 FINS Node No.

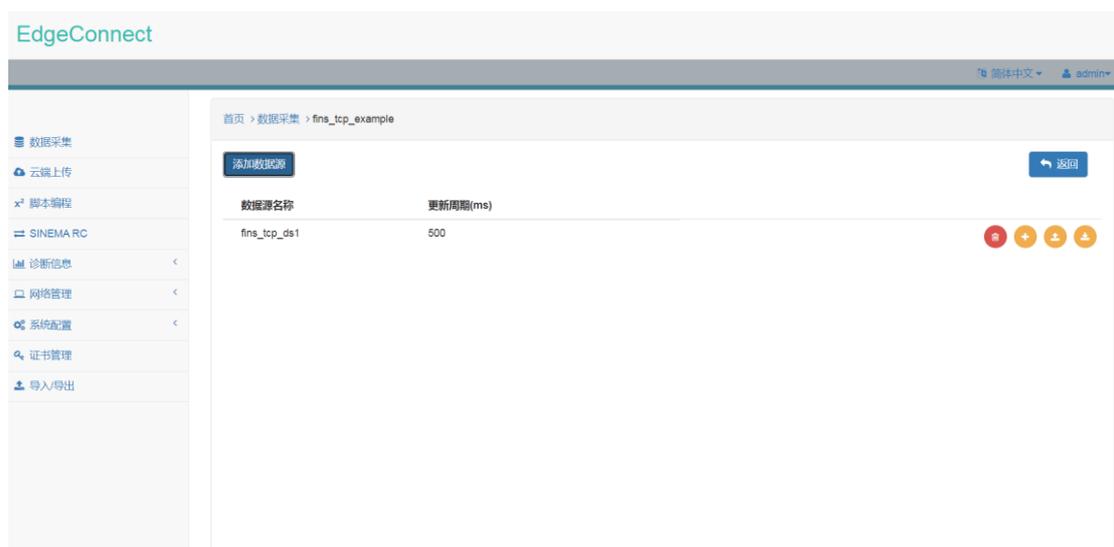
通道详情配置

点击 Omron_FINS_TCP 通道列表项右侧 , 进入通道详情配置页面。在通道详情配置页面中, 用户可以点击“添加数据源”添加数据源 (数据点分组), 在数据源配置界面中设置数据源名称以及更新周期, 即可完成数据源配置。



Omron_FINS_TCP 数据源配置界面

配置完数据源之后，数据源配置信息会以列表形式显示在页面中。



Omron_FINS_TCP 数据源配置信息

点击数据源列表项右侧 \oplus ，即可打开数据点配置界面。

在数据点配置界面中，可以配置数据点名称、地址、源类型、目标类型、以及数据变换等信息。

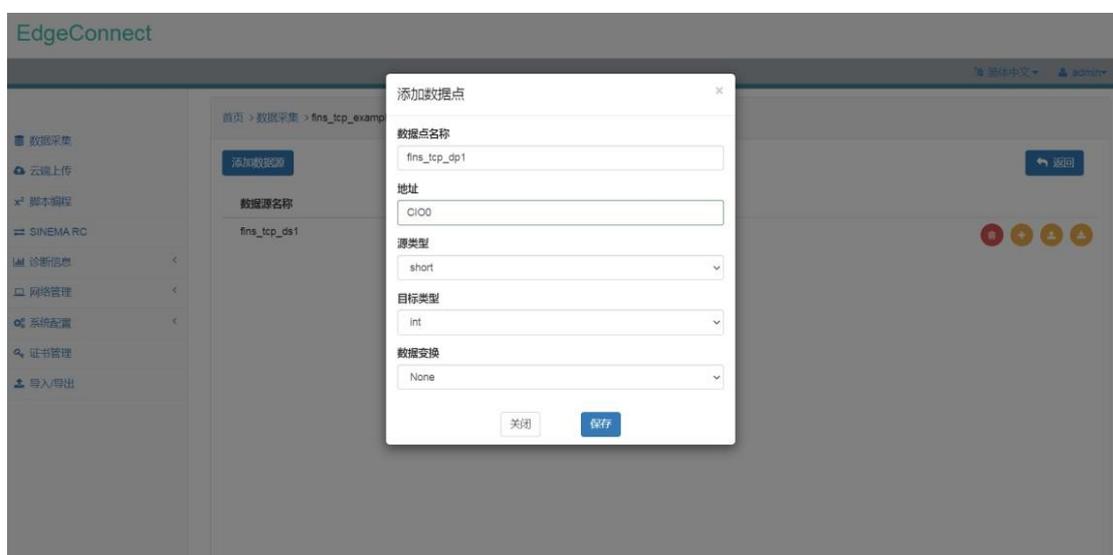


图 4.61 Omron_FINS_TCP 数据点配置界面

其中，数据点地址定义如下表所示：

区域	地址格式
CIO	CIOx, 其中 x 为字节偏移量, 如 CIO0
	CIOx.y, 其中 x 为字节偏移量, y 为位偏移量, y 取值范围为 0 ~ 15, 如 CIO0.0
D	Dx, 其中 x 为字节偏移量, 如 D0
	Dx.y, 其中 x 为字节偏移量, 对于 string 源类型变量, y 为字符串的长度; 对于 bool 源类型变量, y 为位偏移量, 取值范围为 0 ~ 15, 如 D1.1
W	Wx, 其中 x 为字节偏移量, 如 W0
	Wx.y, 其中 x 为字节偏移量, y 为位偏移量, y 取值范围为 0 ~ 15, 如 W0.0

Omron_FINS_TCP 数据点地址定义规则

Omron_FINS_TCP 通道支持的数据点源类型及对应的目标类型如下表所示：

源类型	源类型说明	目标类型
bool	1 bit	bool, int
short	2 bytes, 有符号	int, double
word	2 bytes, 无符号	int, double
float	4 bytes	double
string		string

表 4.16 Omron_FINS_TCP 支持源类型及对应目标类型

对于数据变换，目前支持对数据点进行线性处理。在数据点目标类型为 int、double、long 时方可选择对数据点进行线性处理。线性处理的转换公式为 $V = xp + y$

其中，x 为 multiplier，y 为 offset。

配置完数据点信息之后，数据点信息会以列表形式显示在数据源中。通道配置应用下发后，点击数据点列表项右侧，可查看数据点当前最新值。



Omron_FINS_TCP 数据点配置信息

点击数据源列表项右侧，即可打开数据点批量导入界面，用户可以通过 CSV 文件批量添加数据点配置信息。

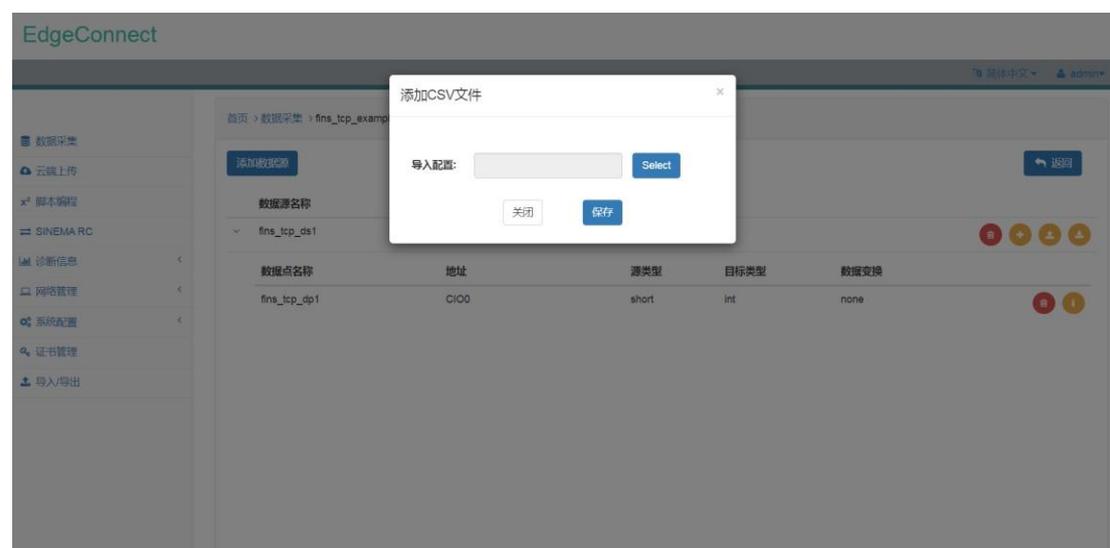


图 4.63 Omron_FINS_TCP 数据点批量添加界面

点击数据源列表项右侧，即可以 CSV 文件形式导出数据点配置信息。

5.3.9AB_Ethernet/IP 通道

通道配置

在“驱动”下拉菜单中选中 AB_Ethernet/IP，并设置 PLC 类型、IP 地址、端口号、槽号及响应超时即可完成相应配置。

AB_Ethernet/IP 通道仅支持数据读取功能。

PLC 类型

- Micro800
- CompactLogix
- ControlLogix

1) 端口号

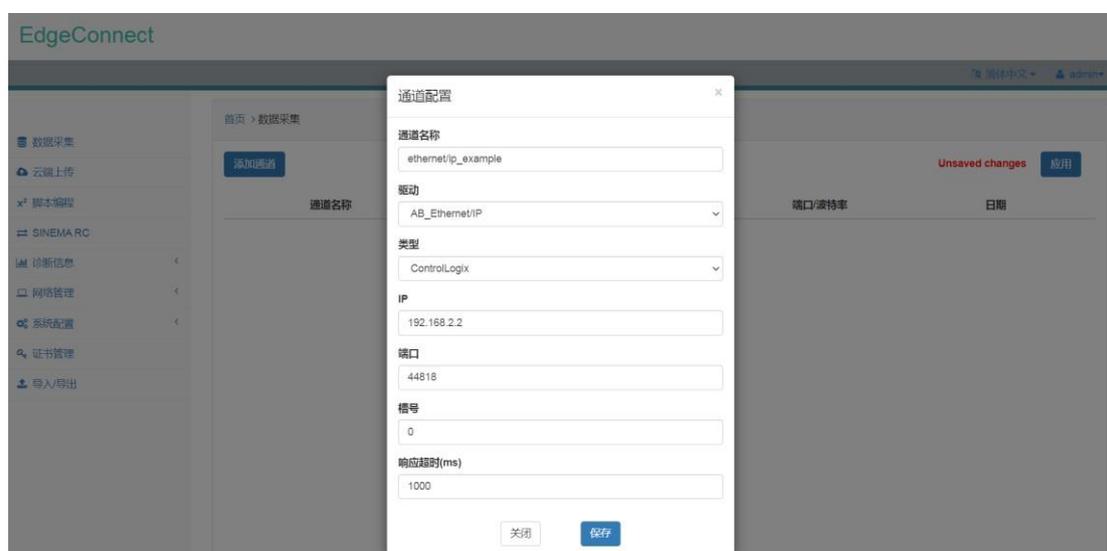
AB_Ethernet/IP 的端口号默认为 44818。

2) 槽号

槽号一般为 0。

3) 响应超时

响应超时时间，默认为 1000ms。



AB_Ethernet/IP 通道配置界面

配置好AB_Ethernet/IP 通道之后，AB_Ethernet/IP 通道信息会以列表形式显示在页面中。



图 4.65 AB_Ethernet/IP 通道配置信息

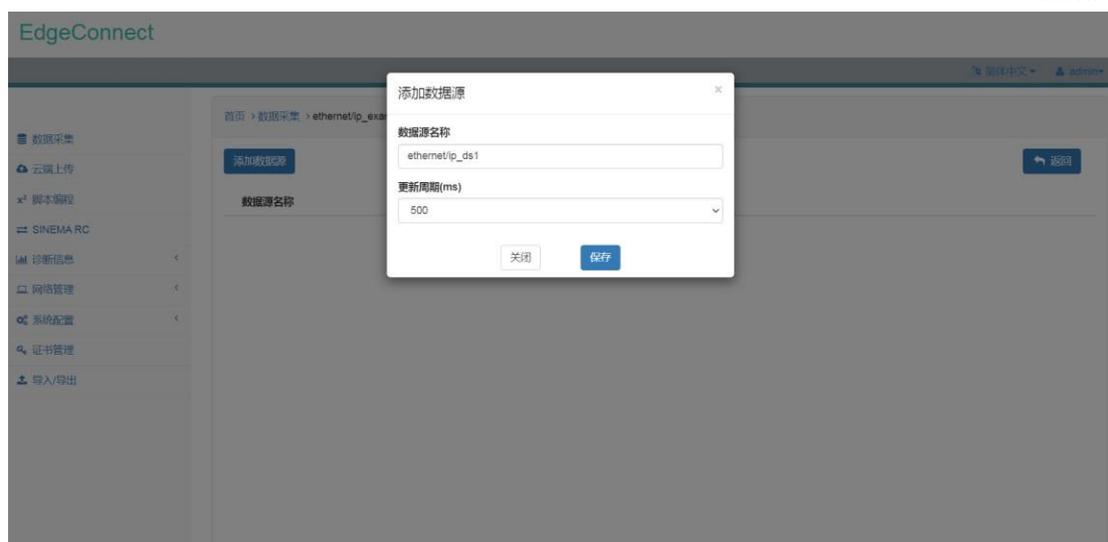
支持的 AB 设备类型

- Micro 820 / 850
- CompactLogix
- ControlLogix

通道详情配置

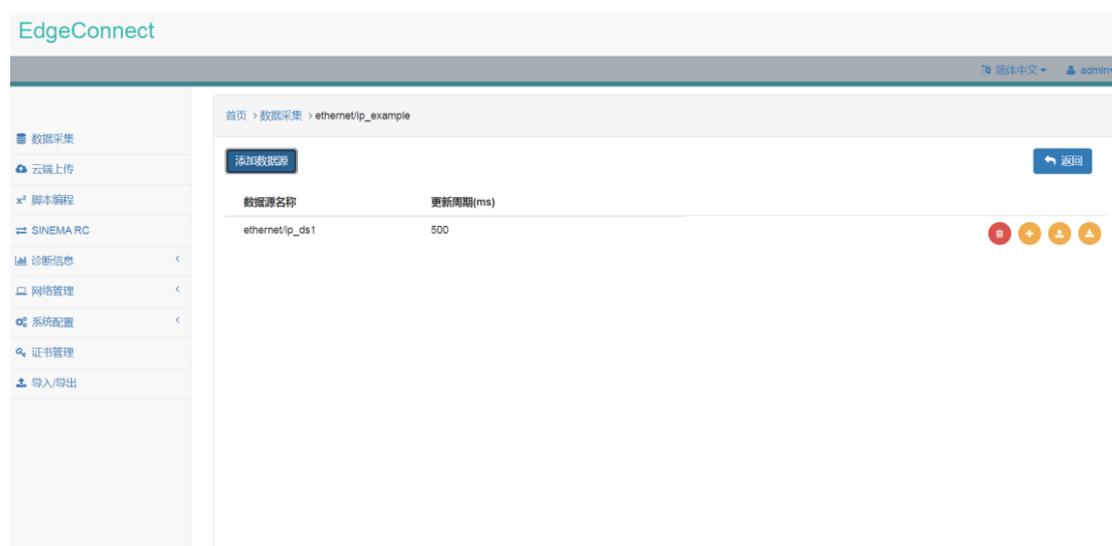
点击AB_Ethernet/IP 通道列表项右侧 \oplus ，进入通道详情配置页面。

在通道详情配置页面中，用户可以点击“添加数据源”添加数据源（数据点分组），在数据源配置界面中设置数据源名称以及更新周期，即可完成数据源配置。



AB_Ethernet/IP 数据源配置界面

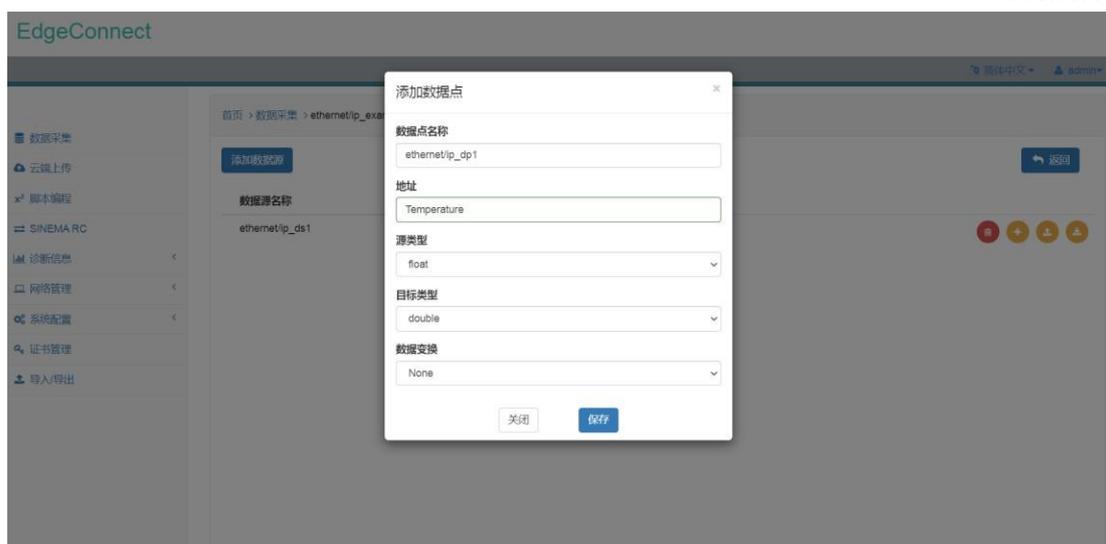
配置完数据源之后，数据源配置信息会以列表形式显示在页面中。



AB_Ethernet/IP 数据源配置信息

点击数据源列表项右侧 \oplus ，即可打开数据点配置界面。

在数据点配置界面中，可以配置数据点名称、地址、源类型、目标类型、以及数据变换等信息。



AB_Ethernet/IP 数据点配置界面

其中，数据点地址定义如下表所示：

地址格式	说明
TagName	
TagName[1]	数组成员变量
TagName.Thing	UDT 成员变量
TagName[4].Thing[2].Length	数组+UDT 成员变量

AB_Ethernet/IP 数据点地址定义规则

源类型	源类型说明	AB 内置类型	目标类型
bool	1 byte	BOOL	bool, int
dword	4 bytes, 无符号		int, double
char	1 byte, 有符号	SINT	int, double
Int	2 bytes, 有符号	INT	int, double
dint	4 bytes, 有符号	DINT	int, double
long	8 bytes, 有符号	LINT	long
float	4 bytes	REAL	double
string		STRING	string

AB_Ethernet/IP 支持源类型及对应目标类型

对于数据变换，目前支持对数据点进行线性处理。在数据点目标类型为 int、double、long 时方可选择对数据点进行线性处理。

线性处理的转换公式为 $V = xp + y$

其中，x 为 multiplier，y 为 offset。

配置完数据点信息后，数据点信息会以列表形式显示在数据源中。

通道配置应用下发后，点击数据点列表项右侧 ，可查看数据点当前最新值。



AB_Ethernet/IP 数据点配置信息

点击数据源列表项右侧 ，即可打开数据点批量导入界面，用户可通过 CSV 文件批量添加数据点配置信息。

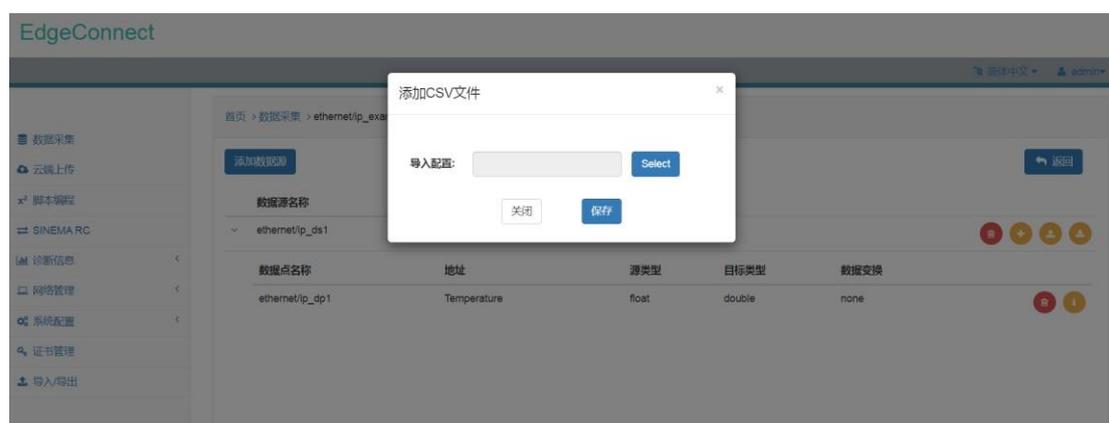
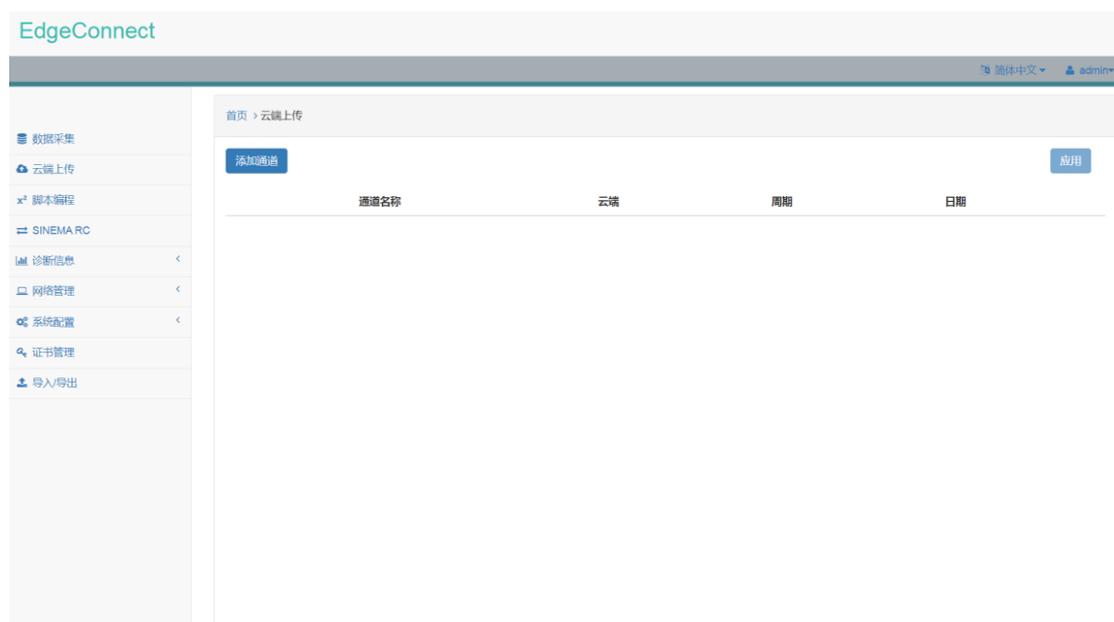


图 4.70 AB_Ethernet/IP 数据点批量添加界面

点击数据源列表项右侧 ，即可以 CSV 文件形式导出数据点配置信息。

5.4 云端上传模块

目前云端上传模块支持 MindSphere、AliCloud、OneNET、General_MQTT、OPCUA_Server及 Modbus_TCP_Server 等云端协议。本章节将介绍如何使用云端上传模块。



云端上传模块界面

用户通过点击“添加通道”按钮，可以选择不同的云端协议，并进行相应的配置。

所有通道配置结束之后，将配置完成的通道状态由停止状态切换至运行状态（红色->绿色），并点击“应用”按钮即可下发云端上传模块配置，并启动云端上传程序。

5.4.1 MindSphere 通道

通道配置

在“云端”下拉菜单中选中 MindSphere，并设置上传周期、配置 ID 及上线密钥即可完成配置。

在 MindSphere 平台创建 Asset 时，请务必选择 **MindConnectLib** 类型。

上传周期

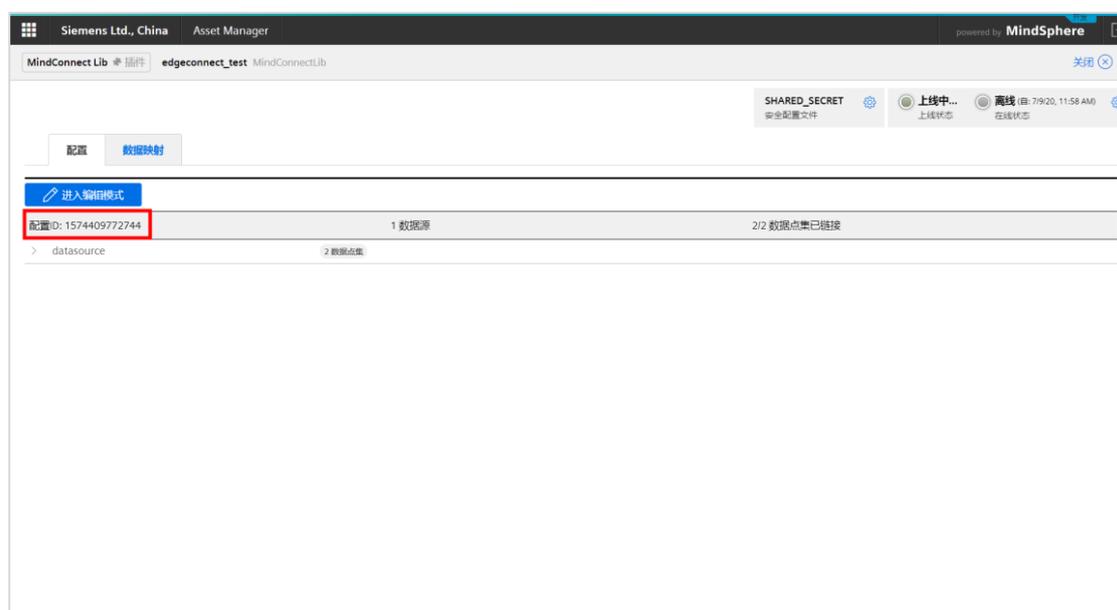
上传周期支持以下选择：

- 1s
- 2s
- 5s
- 10s
- 20s
- 30s
- 1m
- 5m
- 10m
- 20m
- 30m
- 1h

建议设置上传周期 $\geq 5s$ ，以减少 CPU 压力。

1) 配置 ID

需要在 MindSphere 平台配置生成。



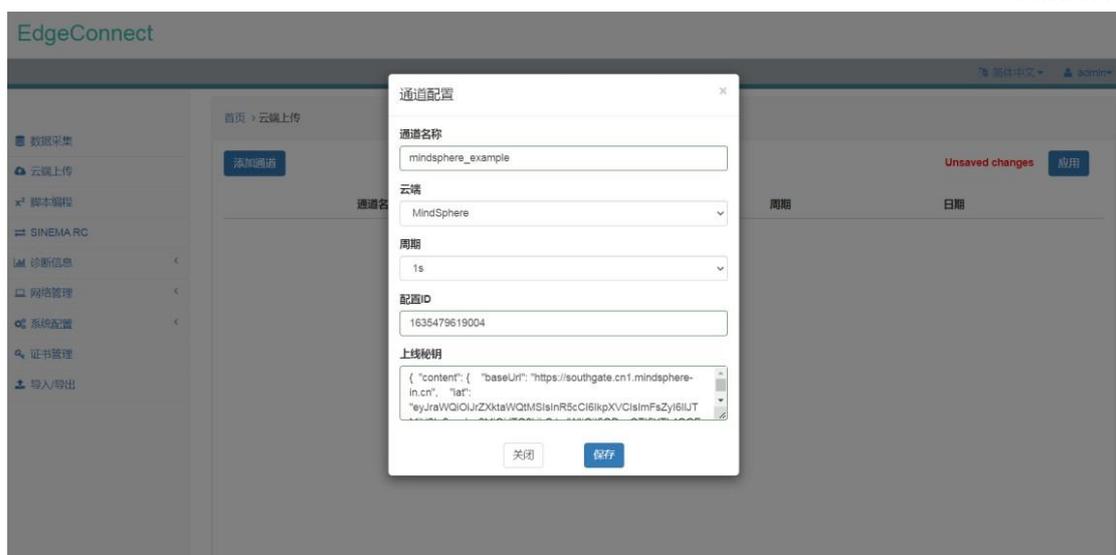
MindSphere 平台配置 ID

2) 上线密钥

需要在 MindSphere 平台配置生成，安全配置文件请务必选择 **SHARED_SECRET**。

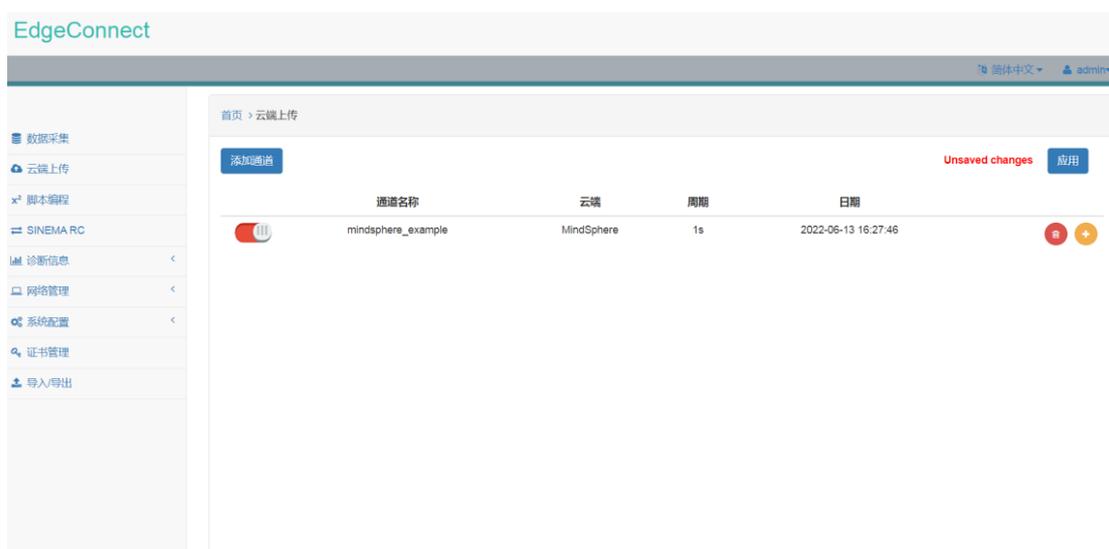


MindSphere 平台上线密钥



MindSphere 通道配置界面

配置好 MindSphere 通道之后，MindSphere 通道信息会以列表形式显示在页面中。

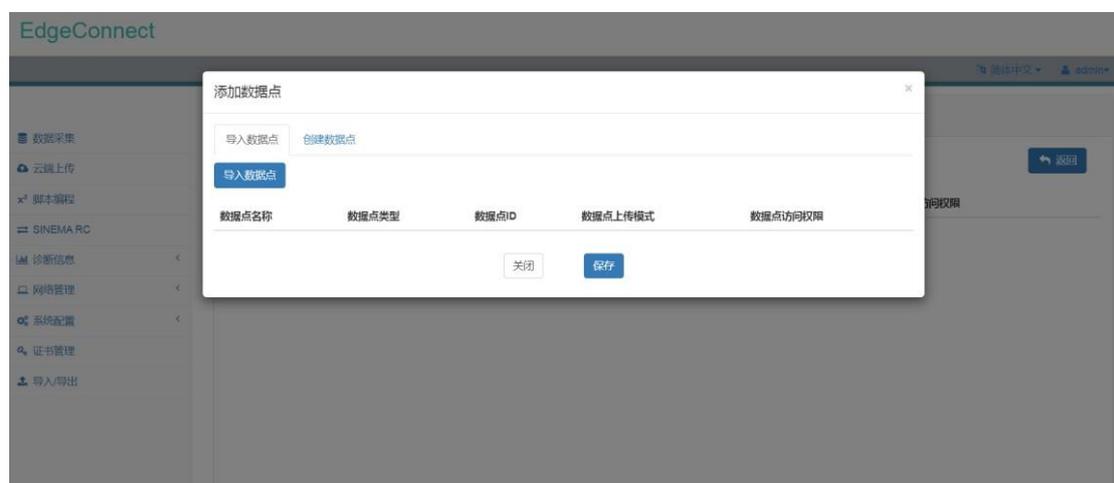


MindSphere 通道配置信息

通道详情配置

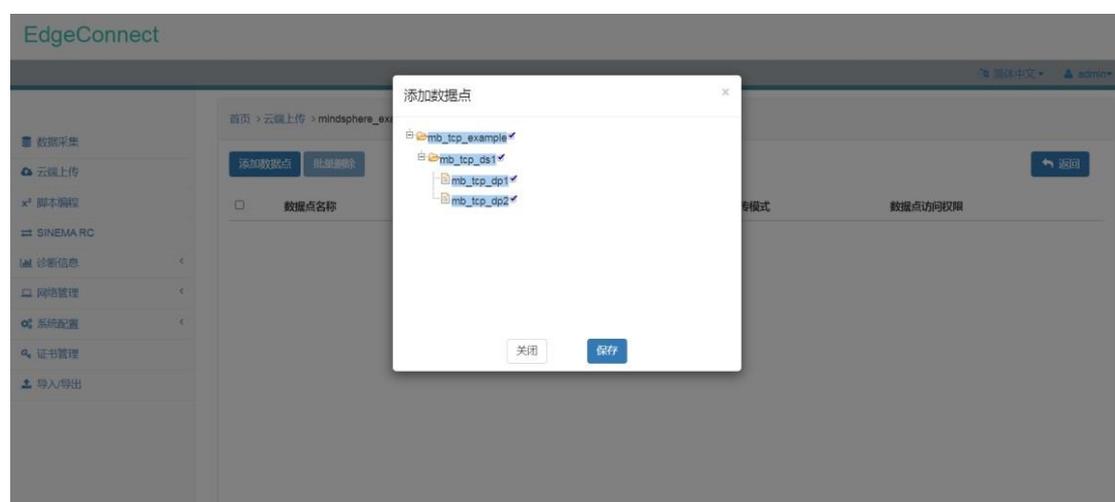
点击 MindSphere 通道列表项右侧 ，即可进入通道详情配置页面。

在通道详情配置页面，用户可以点击“添加数据点”打开数据点配置界面。



MindSphere 数据点配置界面

在数据点配置界面中，点击“导入数据点”，会出现所有数据采集通道、脚本编程通道及对应通道内数据点的信息。用户可以选择需要上传的数据点。



MindSphere 选择数据点界面

选中数据点并保存，会返回至数据点配置界面，并加载已选中数据点的信息。用户需要填入在 MindSphere 云平台上配置好的数据点 ID 信息，并选择数据点上传模式及读写权限。

数据点上传模式支持两种：

1) Continuous：

连续上传模式，采集到的数据点连续不断上传。

2) Onchange：

逢变上传模式，采集到的数据点变化时上传。

读写权限仅支持只读权限：

Read：

只读权限，仅支持读取采集到的数据点，不允许将数据点的值写入数据采集通道。

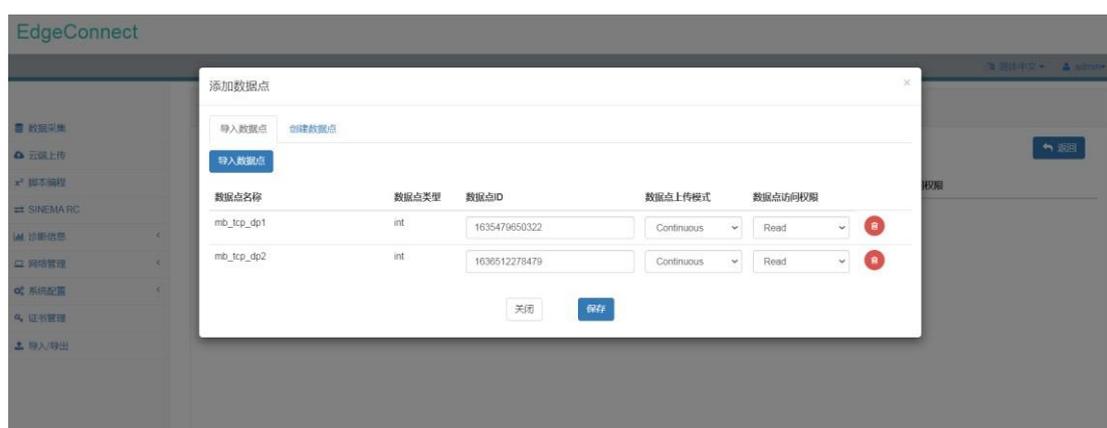


图 5.8 MindSphere 数据点配置界面

配置完数据点信息之后，数据点信息会以列表形式显示在页面上。

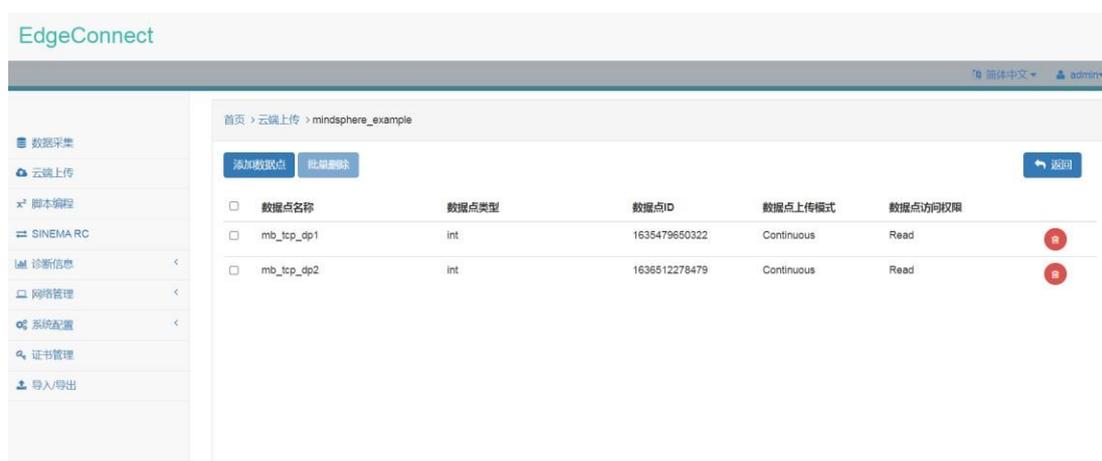


图 5.9 MindSphere 数据点配置信息

5.4.2 AliCloud 通道

通道配置

在“云端”下拉菜单中选中 AliCloud，并设置上传周期、实例类型（公共实例或企业实例）、主机名（企业实例需设置）、产品 KEY、设备名称、设备密钥及 QOS 级别即完成配置。

1) 上传周期

上传周期支持以下选择：

- 1s
- 2s
- 5s
- 10s
- 20s
- 30s
- 1m
- 5m
- 10m
- 20m
- 30m
- 1h

AliCloud 上传时会将所有数据点 ID 对应的数据点上传，数据点的内容为最近一次采集的数据点内容。

2) 实例类型

实例类型支持以下选择：

Public Instance，公共实例

Enterprise Instance，企业实例

企业实例用户需设置主机名，主机名可在企业实例->开发配置->公网终端节点（Endpoint）->MQTT 设备接入查看。

3) 产品KEY

需要在阿里云 IOT 平台配置生成。

4) 设备名称

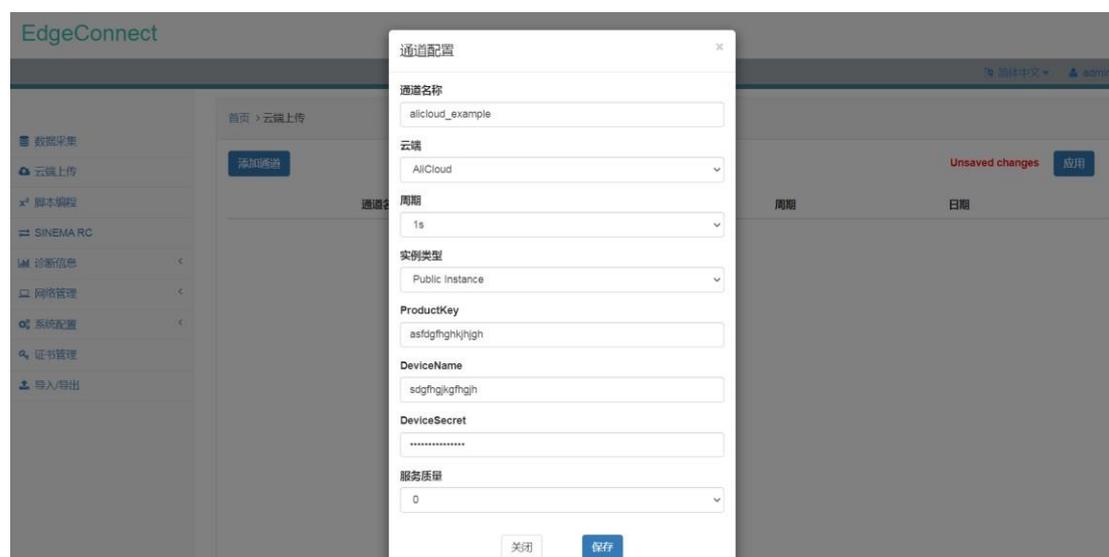
需要在阿里云 IOT 平台配置生成。

5) 设备密钥

需要在阿里云 IOT 平台配置生成。

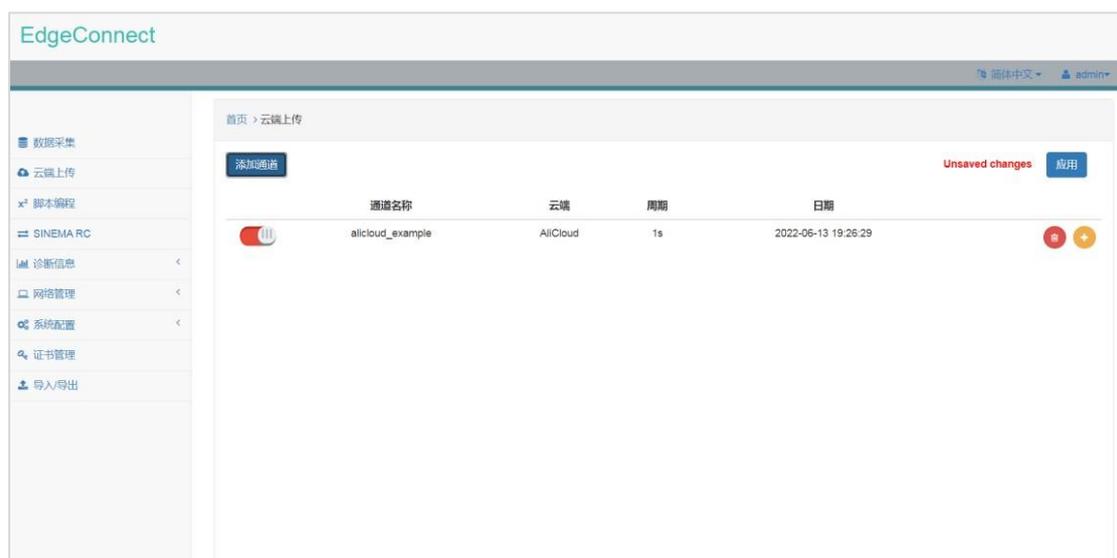
6) QOS

支持 QOS 0 或 QOS 1。推荐选择 QOS1。



AliCloud 通道配置界面

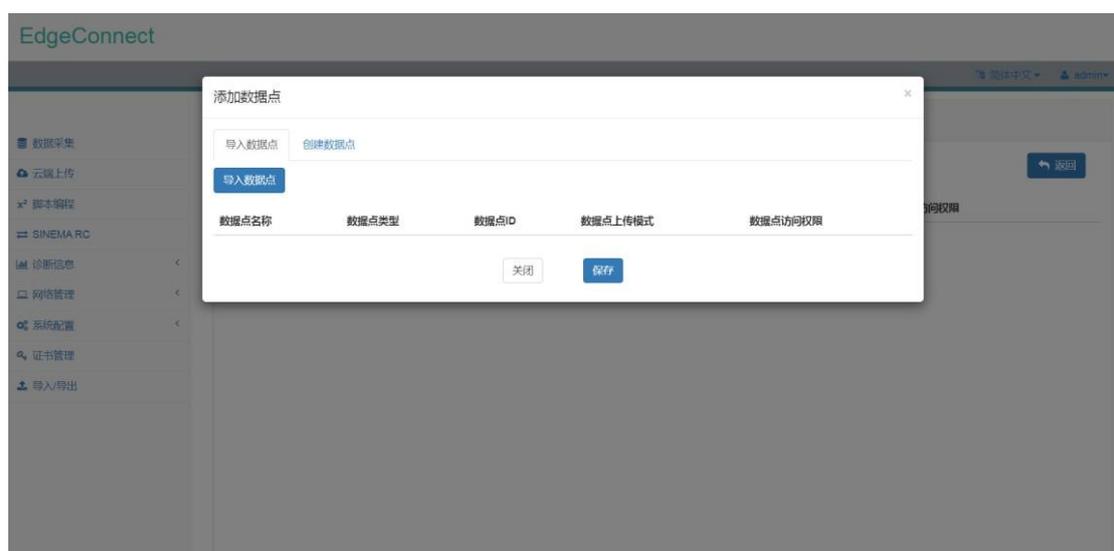
配置好 AliCloud 通道之后，AliCloud 通道信息会以列表形式显示在页面中。



AliCloud 通道配置信息

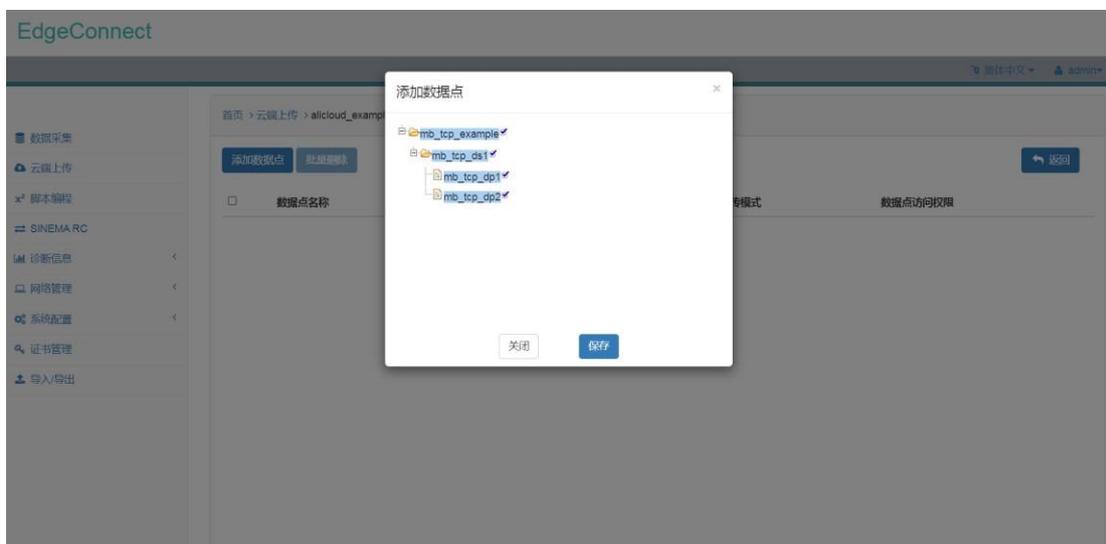
通道详情配置

点击 AliCloud 通道列表项右侧 , 即可进入通道详情配置页面。在通道详情配置页面，用户可以点击“添加数据点”打开数据点配置界面。



AliCloud 数据点配置界面

在数据点配置界面中，点击“导入数据点”，会出现所有数据采集通道、脚本编程通道及对应通道内数据点的信息。用户可以选择需要上传的数据点。



AliCloud 选择数据点界面

选中数据点并保存，会返回至数据点配置界面，并加载已选中数据点的信息。用户需要填入在阿里云 IOT 平台上配置好的功能名称，并选择数据点上传模式及读写权限。

数据点上传模式支持两种：

1) Continuous

连续上传模式，采集到的数据点连续不断上传。

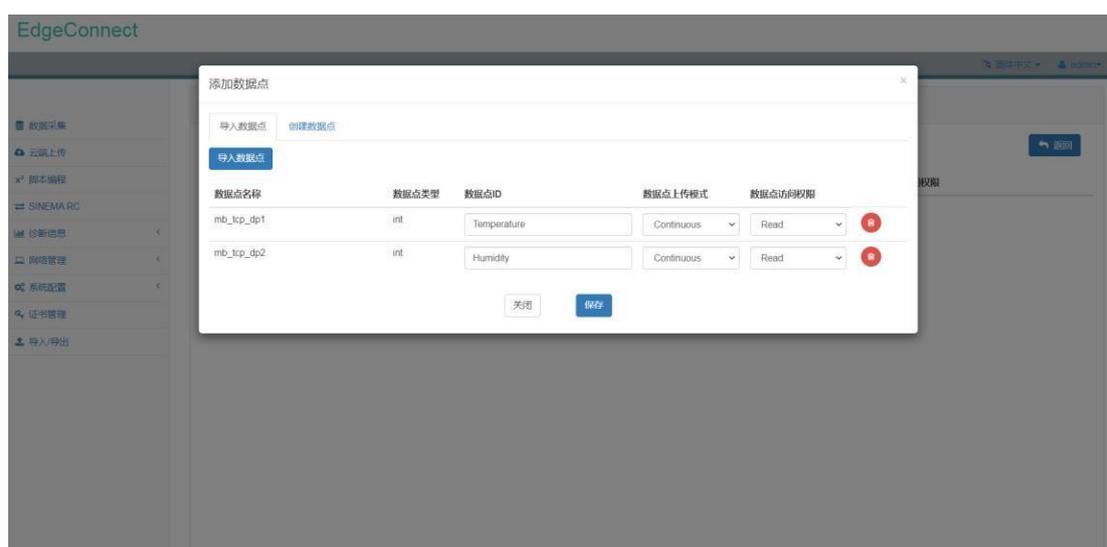
2) Onchange

逢变上传模式，采集到的数据点变化时上传。

读写权限仅支持只读权限：

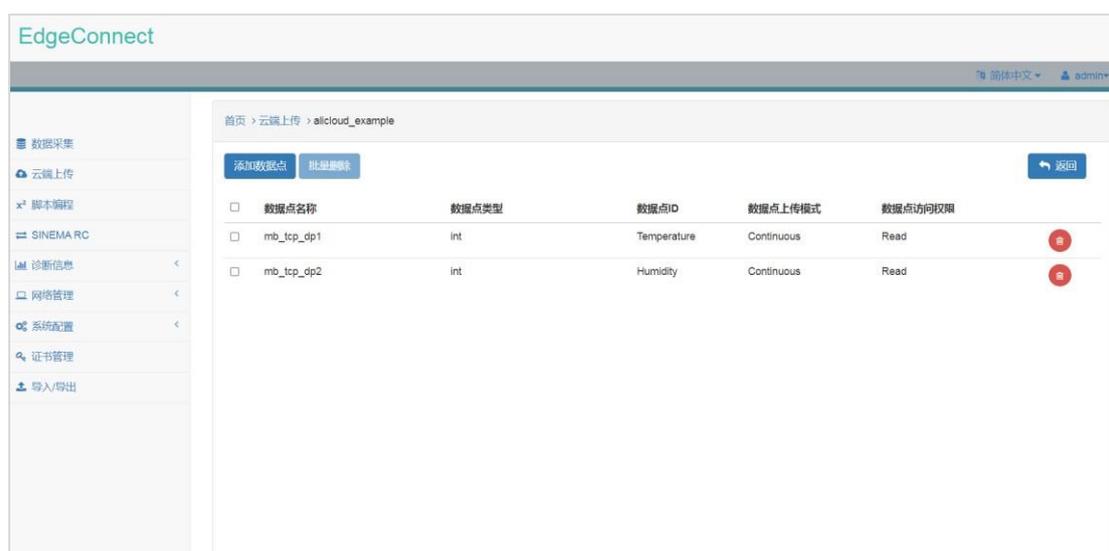
1) Read

只读权限，仅支持读取采集到的数据点，不允许将数据点的值写入数据采集通道。



AliCloud 数据点配置界面

配置完数据点信息之后，数据点信息会以列表形式显示在页面上。



AliCloud 数据点配置信息

5.4.3 OneNET 通道

通道配置

在“云端”下拉菜单中选中 OneNET，并设置上传周期、产品 ID、设备名称、设备 KEY 及 QOS 级别即完成配置。

1) 上传周期

上传周期支持以下选择：

- 1s
- 2s
- 5s
- 10s
- 20s
- 30s
- 1m
- 5m
- 10m
- 20m
- 30m
- 1h

OneNET 上传时会将所有数据点 ID 对应的数据点上传，数据点的内容为最近一次采集的数据点内容。

2) 产品 ID

需要在 OneNET 平台 MQTT 物联网套件配置生成。

3) 设备名称

需要在 OneNET 平台 MQTT 物联网套件配置生成。

4) 设备KEY

需要在 OneNET 平台 MQTT 物联网套件配置生成。

5) QOS

支持 QOS 0 或 QOS 1。推荐选择 QOS1。



OneNET 通道配置界面

配置好 OneNET 通道之后，OneNET 通道信息会以列表形式显示在页面中。

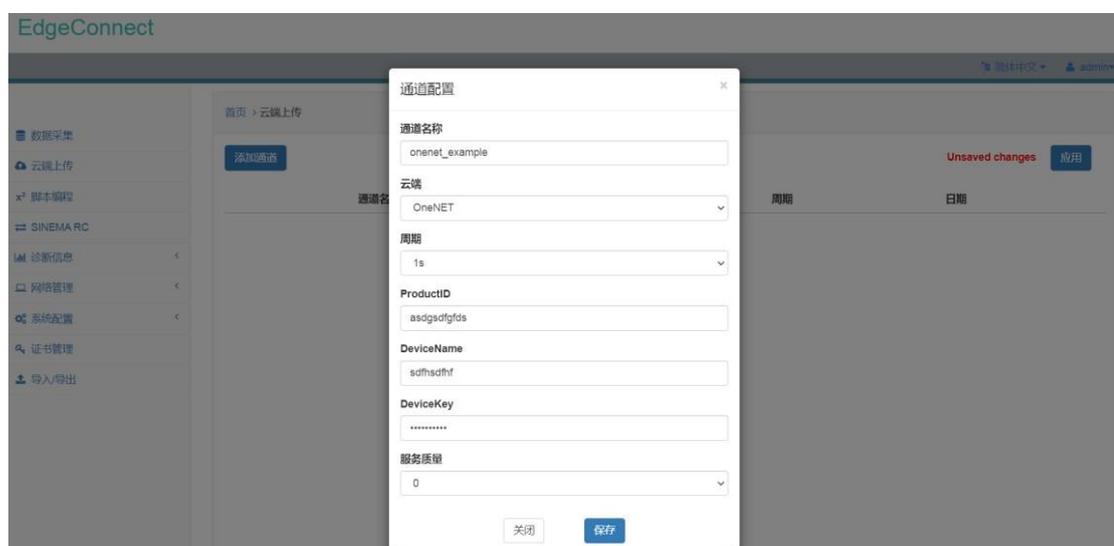
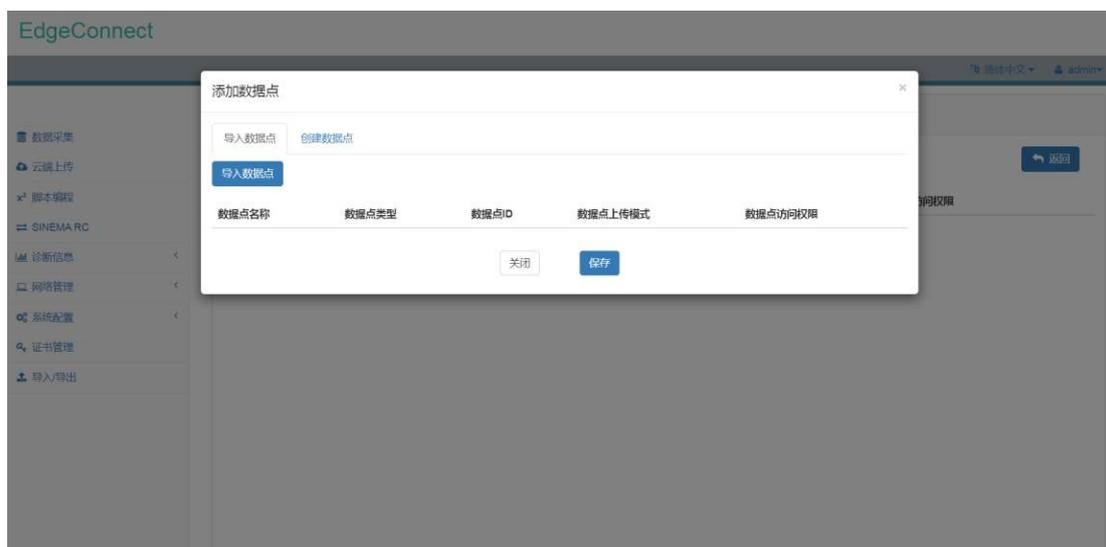


图 5.17 OneNET 通道配置信息

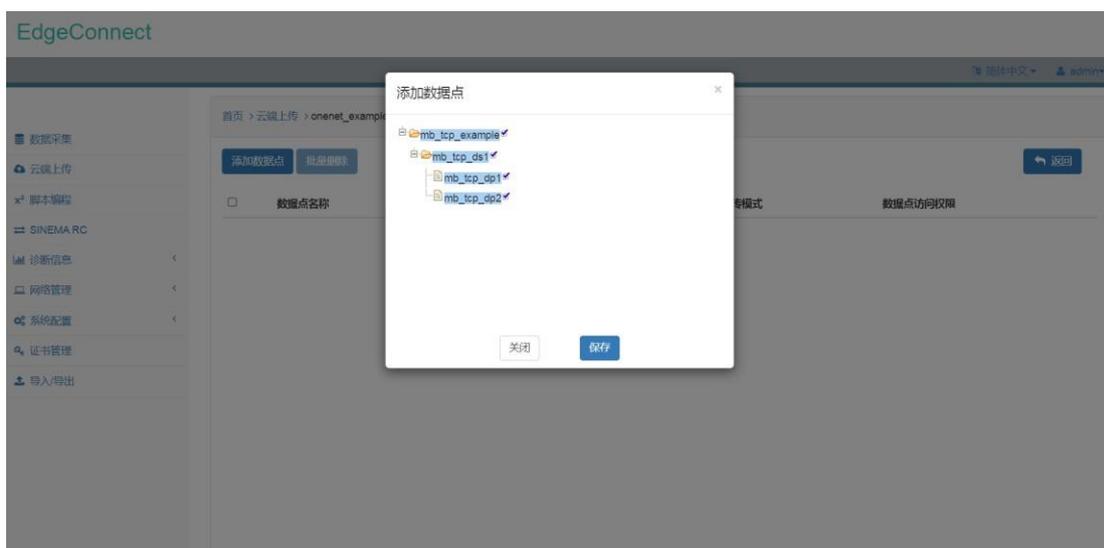
通道详情配置

点击OneNET 通道列表项右侧 \oplus ，即可进入通道详情配置页面。在通道详情配置页面，用户可以点击“添加数据点”打开数据点配置界面。



OneNET 数据点配置界面

在数据点配置界面中，点击“导入数据点”，会出现所有数据采集通道、脚本编程通道及对应通道内数据点的信息。用户可以选择需要上传的数据点。



OneNET 选择数据点界面

选中数据点并保存，会返回至数据点配置界面，并加载已选中数据点的信息。用户需要填入在 OneNET 平台 MQTT 物联网套件配置好的数据流名称，并选择数据点上传模式及读写权限。

数据点上传模式支持两种：

1) Continuous

连续上传模式，采集到的数据点连续不断上传。

2) Onchange

逢变上传模式，采集到的数据点变化时上传。

读写权限支持两种：

1) Read

只读权限，仅支持读取采集到的数据点，不允许将数据点的值写入数据采集通道。

2) Read & Write

读写权限，同时支持读取采集到的数据点及将数据点的值写入数据采集通道。

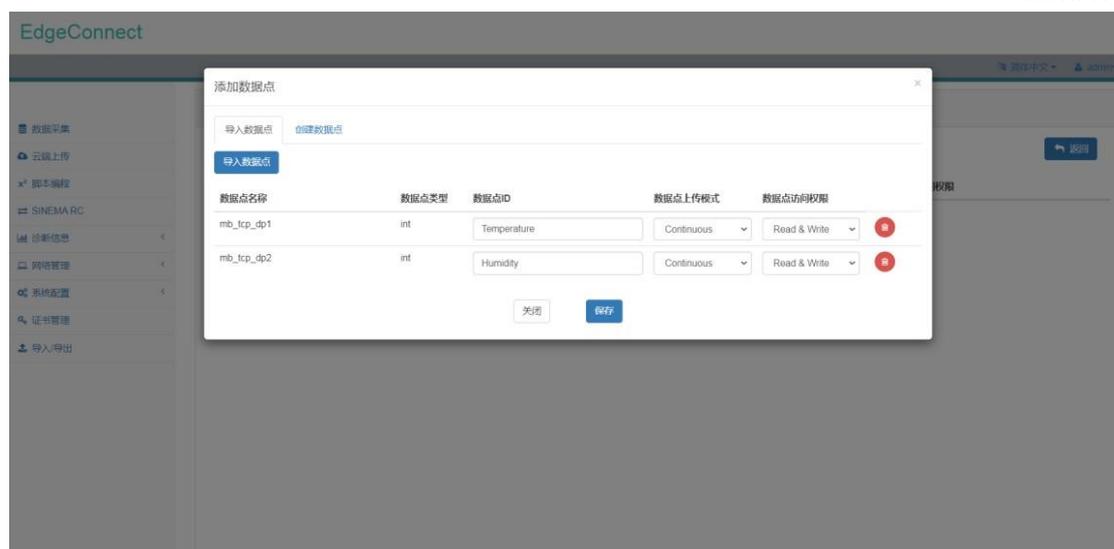


图 5.20 OneNET 数据点配置界面

配置完数据点信息之后，数据点信息会以列表形式显示在页面上。

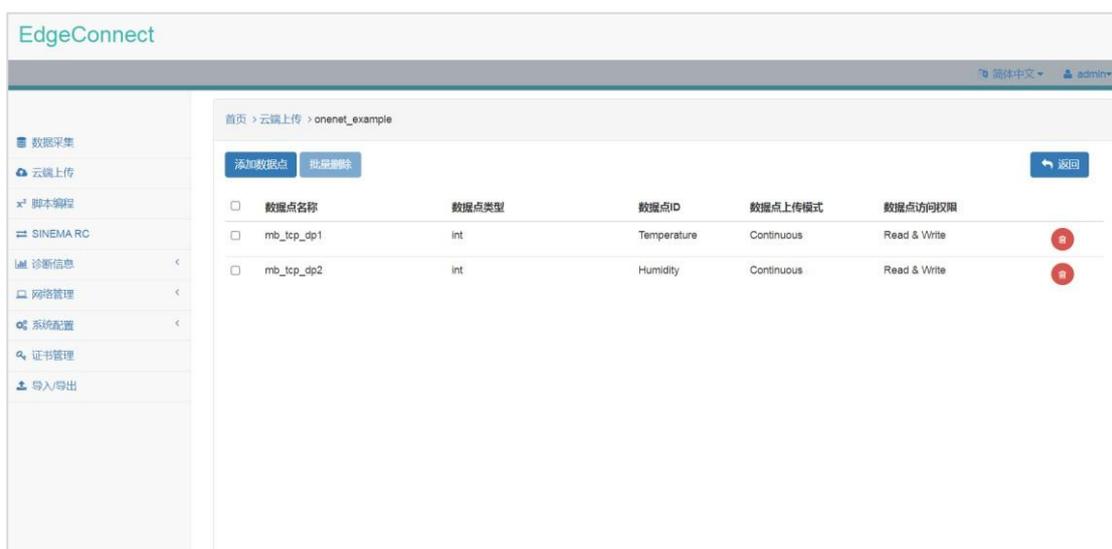
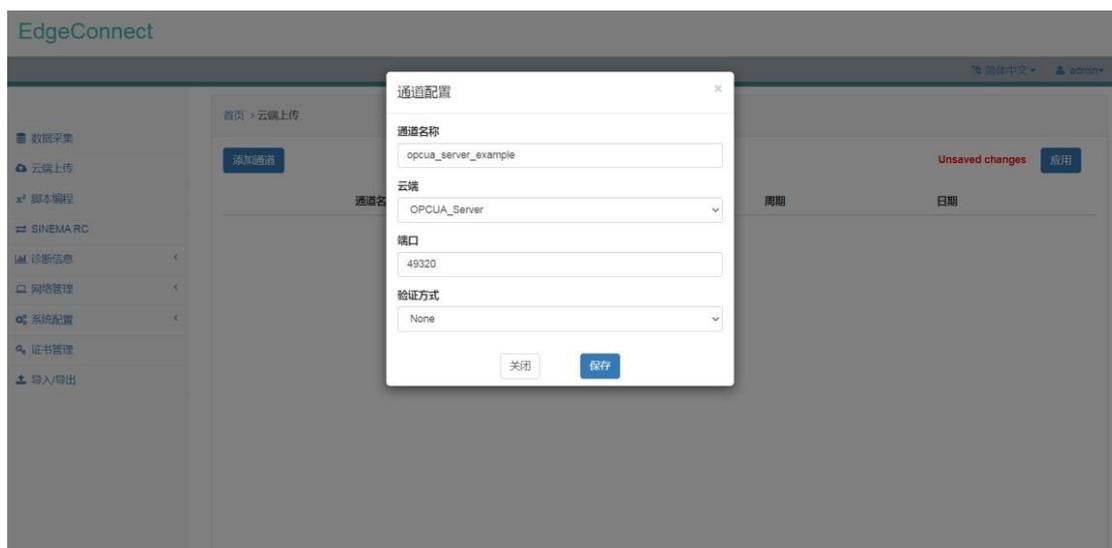


图 5.21 OneNET 数据点配置信息

5.4.4 OPCUA_Server 通道

通道配置

在“云端”下拉菜单中选中 OPCUA_Server，并设置端口号、验证信息（无验证或用户名/密码验证）即完成配置。



OPCUA_Server 通道配置界面

配置好 OPCUA_Server 通道之后，OPCUA_Server 通道信息会以列表形式显示在页面中。

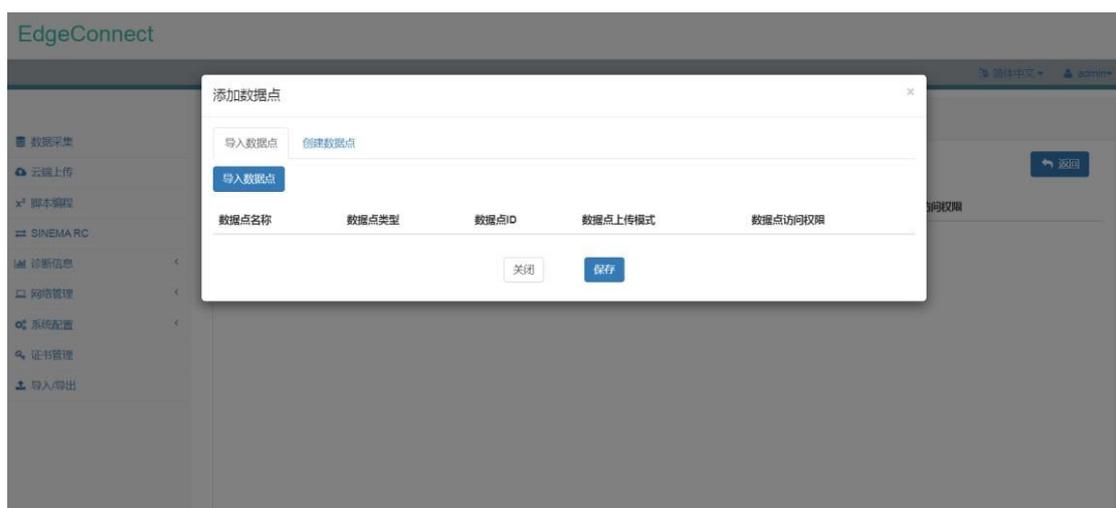


OPCUA_Server 通道配置信息

通道详情配置

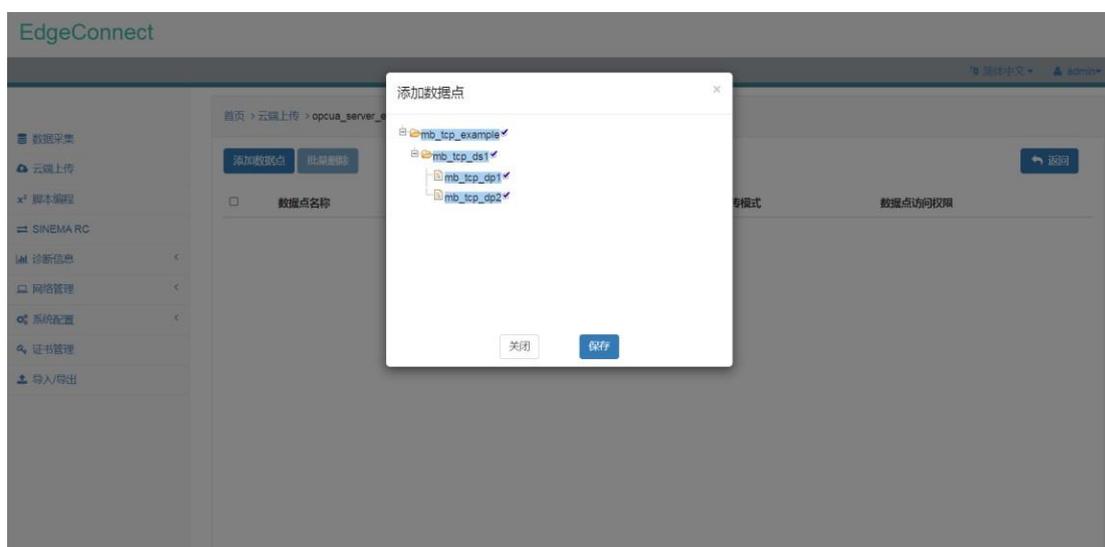
点击OPCUA_Server 通道列表项右侧 \oplus ，即可进入通道详情配置页面。

在通道详情配置页面，用户可以点击“添加数据点”打开数据点配置界面。



OPCUA_Server 数据点配置界面

在数据点配置界面中，点击“导入数据点”，会出现所有数据采集通道、脚本编程通道及对应通道内数据点的信息。用户可以选择需要上传的数据点。



OPCUA_Server 选择数据点界面

选中数据点并保存，会返回至数据点配置界面，并加载已选中数据点的信息。用户需要填入 OPCUA_Server 地址空间中对应的变量名称并选择数据点上传模式及读写权限。

数据点上传模式支持两种：

1) Continuous

连续上传模式，采集到的数据点连续不断上传。

2) Onchange

逢变上传模式，采集到的数据点变化时上传。

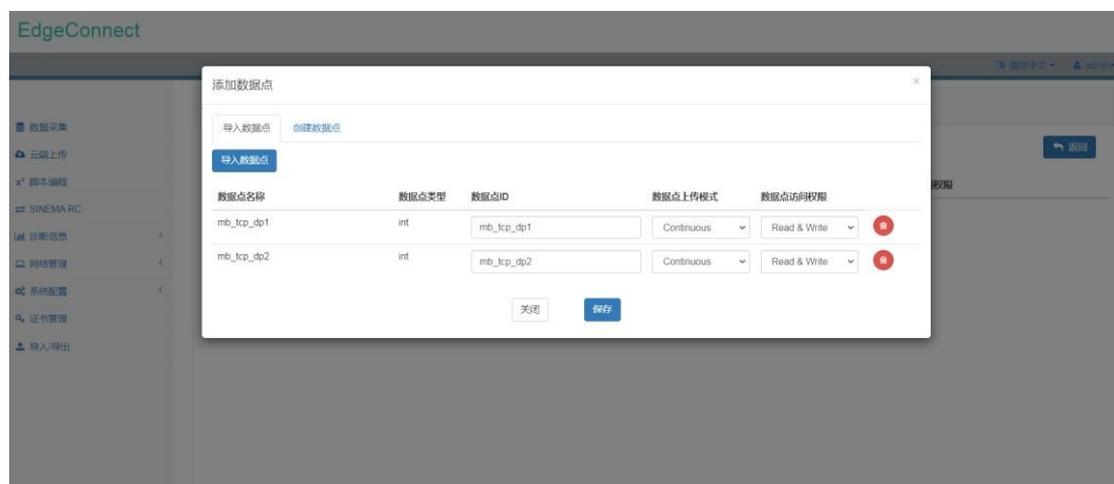
读写权限支持两种：

1) Read

只读权限，仅支持读取采集到的数据点，不允许将数据点的值写入数据采集通道。

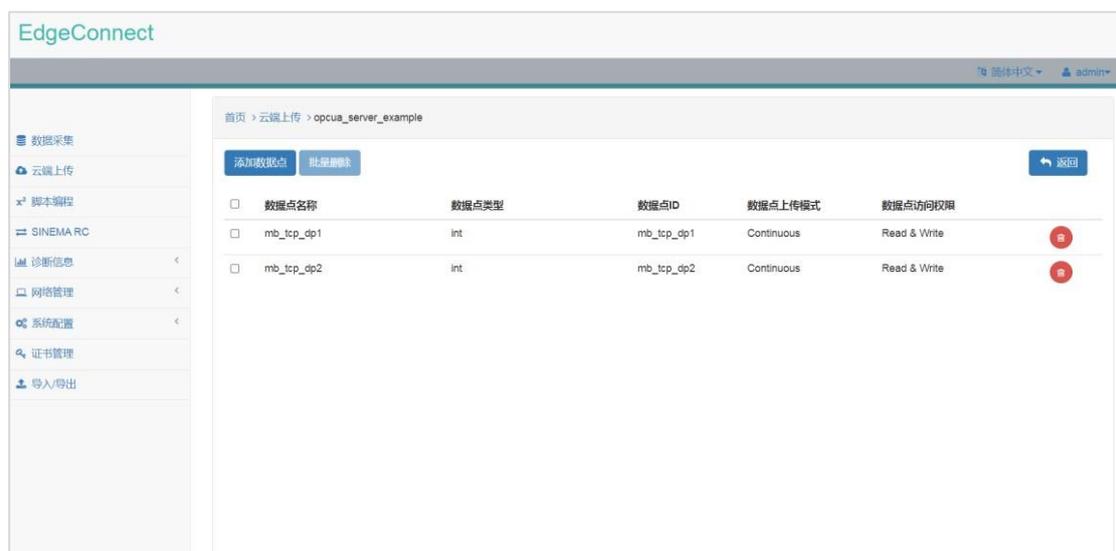
2) Read & Write

读写权限，同时支持读取采集到的数据点及将数据点的值写入数据采集通道。



OPCUA_Server 数据点配置界面

配置完数据点信息之后，数据点信息会以列表形式显示在页面上。



OPCUA_Server 数据点配置信息

访问 OPCUA_Server

1) Endpoint

EdgeConnect OPCUA_Server 的 EndPoint 为：

opc.tcp//<IP>:<Port>, 其中<IP>为设备 IP 地址, <Port>为设置的端口号。

2) Namespace

Namespace 为 ns1。

5.4.5 General_MQTT 通道

通道配置

在“云端”下拉菜单中选中 General_MQTT，并设置上传周期、主机名、端口、ClientID、用户名、密码、QOS 级别、安全机制、离线缓存、发布主题、订阅主题及数据格式即完成配置。

数据格式 Format 2 及 Format 3 支持上报 GPS 定位信息，如选择上述两种数据格式，需安装支持 GPS 功能的 4G 模组，并连接好 GPS 天线。

1) 上传周期

上传周期支持以下选择：

- 1s
- 2s
- 5s
- 10s
- 20s
- 30s
- 1m
- 5m
- 10m
- 20m
- 30m
- 1h

General_MQTT 上传时会将所有数据点 ID 对应的数据点上传，数据点的内容为最近一次采集的数据点内容。

2) 主机名

MQTT Broker 的域名或者 IP 地址。

3) 端口号

MQTT Broker 的端口号。

4) ClientID

MQTT 客户端 ID，该 ID 需要唯一，可留空。如留空，会自动生成 ClientID。

5) 用户名

MQTT 客户端用户名，可留空。

6) 密码

MQTT 客户端密码，可留空。

7) 服务质量

支持 QOS 0、QOS 1 及 QOS2。

8) 安全机制

支持None（无验证）、Server Authentication（单向验证）或 Mutual Authentication（双向验证）等三种方式。

如选择 None，无需其他操作；

如选择 Server Authentication，需在证书管理模块上传 CA 证书，并选择该证书；

如选择 Mutual Authentication，需在证书管理模块上传 CA 证书、客户端证书、客户端密钥，并选择对应证书。

9) 离线缓存

支持 Disable 及 Enable。

10) 发布主题

MQTT 消息的发布主题，该主题用于向 MQTT Broker 发布数据。

11) 订阅主题

MQTT 消息的订阅主题，该主题用于从 MQTT Broker 订阅数据。

订阅消息格式为：

```
{  
    "temperature": 10.1,  
    "Humidity": 20.2  
}
```

12) 数据格式

支持 Format 0、Format 1、Format 2、Format 3 等四种数据格式。其中，Format 2 及 Format 3 支持 GPS 定位信息上报。

a) Format 0 格式为：

```
{  
    "timestamp": 1592799105,  
    "temperature": 10.1,  
    "Humidity": 20.2  
}
```

B) Format 1 格式为：

```
{  
    "temperature": {  
        "value": 11,  
        "timestamp": 1592799105233  
    },  
    "Humidity": {  
        "value": 11,  
        "timestamp": 1592799105233  
    }  
}
```

C) Format 2 格式为：

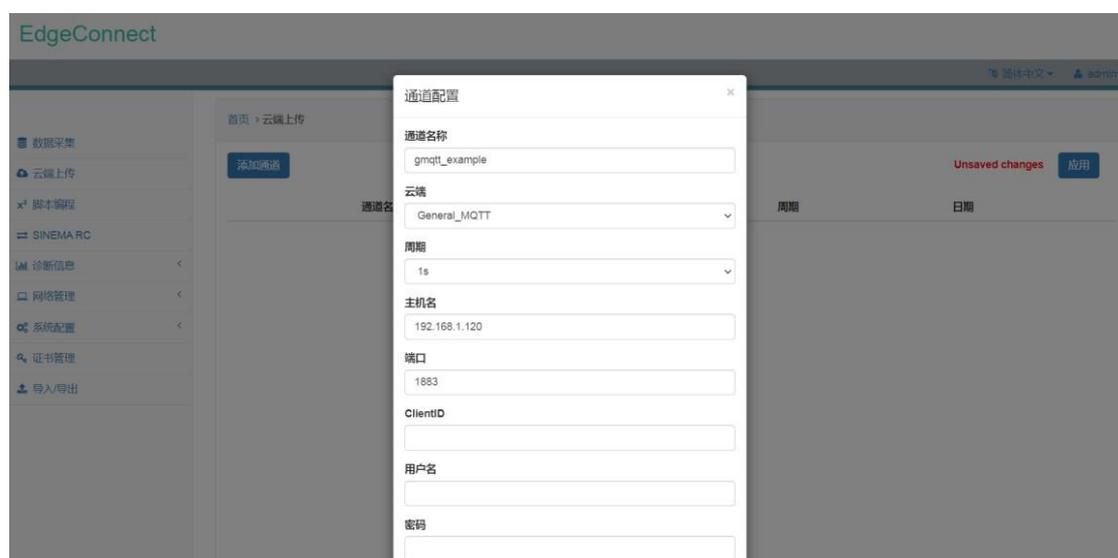
```
{  
    "timestamp": 1592799105,  
    "location": {  
        "latitude": 50.23345,  
        "longtitude": 130.23432  
    },  
    "temperature": 10.1,  
    "Humidity": 20.2  
}
```

D) Format 3 格式为：

```

{
  "location": {
    "latitude": 50.23345,
    "longtitude": 130.23432
  },
  "temperature": {
    "value": 11,
    "timestamp": 1592799105233
  },
  "Humidity": {
    "value": 11,
    "timestamp": 1592799105233
  }
}

```



General_MQTT 通道配置界面

配置好General_MQTT 通道之后，General_MQTT 通道信息会以列表形式显示在页面中。

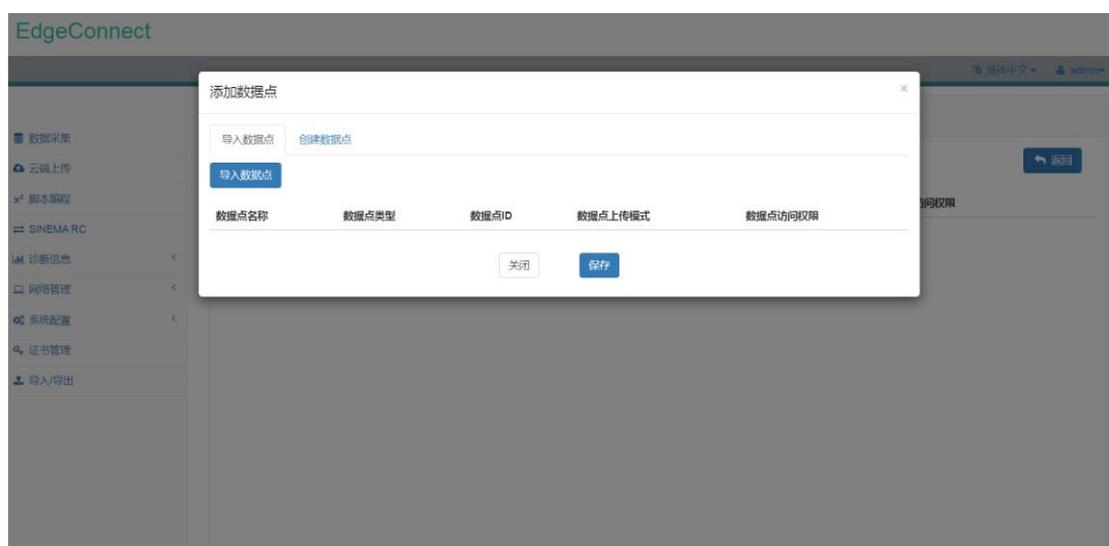


General_MQTT 通道配置信息

通道详情配置

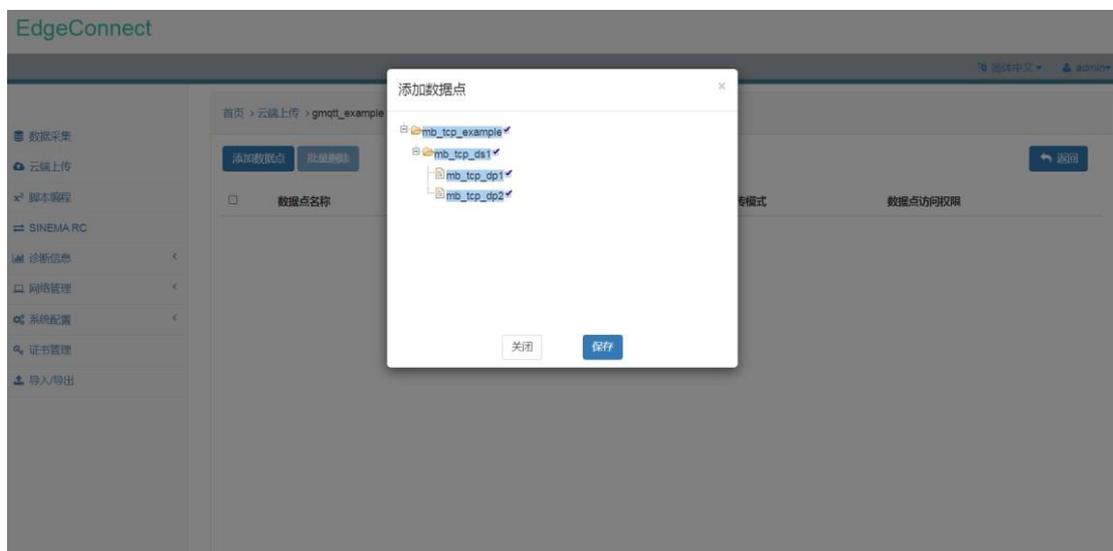
点击 General_MQTT 通道列表项右侧 , 即可进入通道详情配置页面。

在通道详情配置页面，用户可以点击“添加数据点”打开数据点配置界面。



General_MQTT 数据点配置界面

在数据点配置界面中，点击“导入数据点”，会出现所有数据采集通道、脚本编程通道及对应通道内数据点的信息。用户可以选择需要上传的数据点。



General_MQTT 选择数据点界面

选中数据点并保存，会返回至数据点配置界面，并加载已选中数据点的信息。用户需要填入对应的数据点 ID，并选择数据点上传模式及读写权限。

数据点上传模式支持两种：

1) Continuous

连续上传模式，采集到的数据点连续不断上传。

2) Onchange

逢变上传模式，采集到的数据点变化时上传。

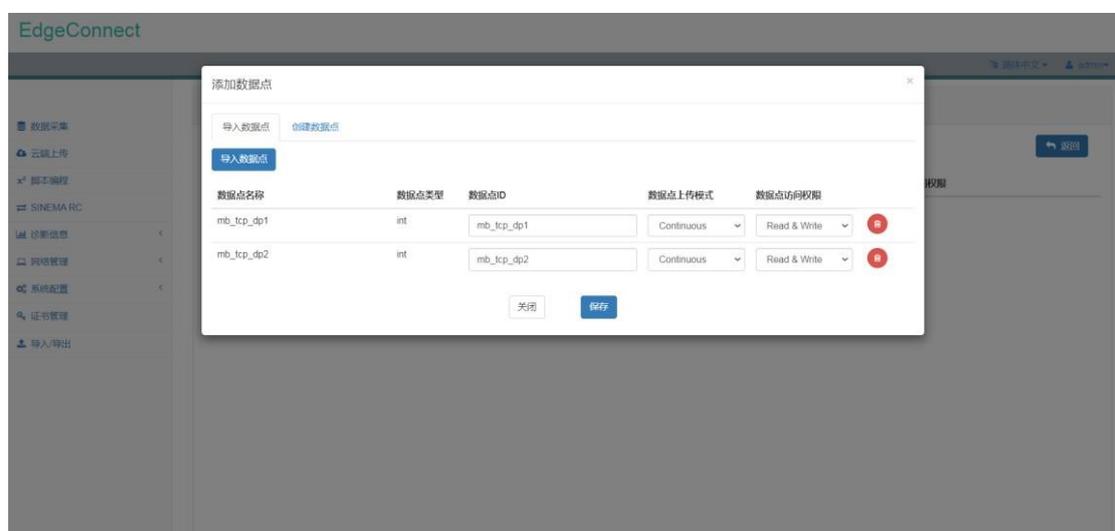
读写权限支持两种：

1) Read

只读权限，仅支持读取采集到的数据点，不允许将数据点的值写入数据采集通道。

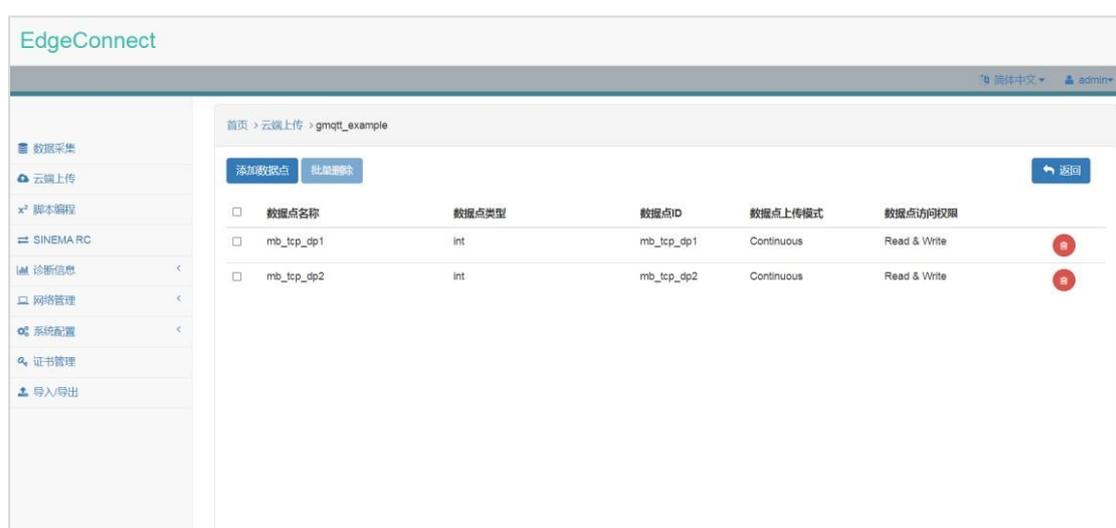
2) Read & Write

读写权限，同时支持读取采集到的数据点及将数据点的值写入数据采集通道。



General_MQTT 数据点配置界面

配置完数据点信息之后，数据点信息会以列表形式显示在页面上。



General_MQTT 数据点配置信息

5.4.6 Modbus_TCP_Server 通道

通道配置

在“云端”下拉菜单中选中 Modbus_TCP_Server，并设置端口号、站号、内存布局即完成配置。

Modbus_TCP_Server 通道仅支持保持寄存器数据区域（03、06、10 功能码）。

1) 端口号

Modbus_TCP_Server 的端口号默认为 502。

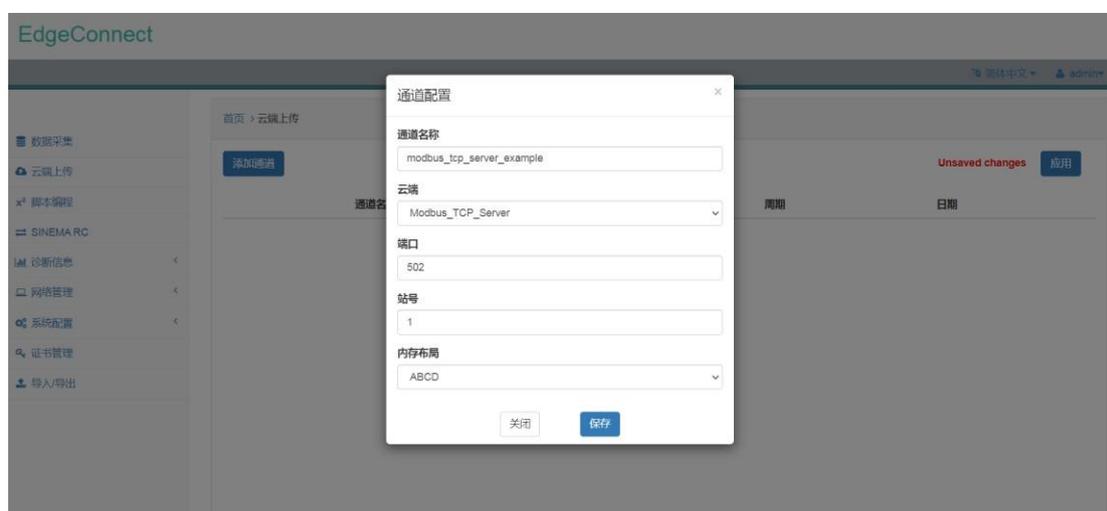
2) 站号

站号可自定义，范围为 1 - 247。

3) 内存布局

存储浮点数需要选择对应的内存布局，内存布局支持以下选项：

- ABCD
- BADC
- CDAB
- DCBA



Modbus_TCP_Server 通道配置界面

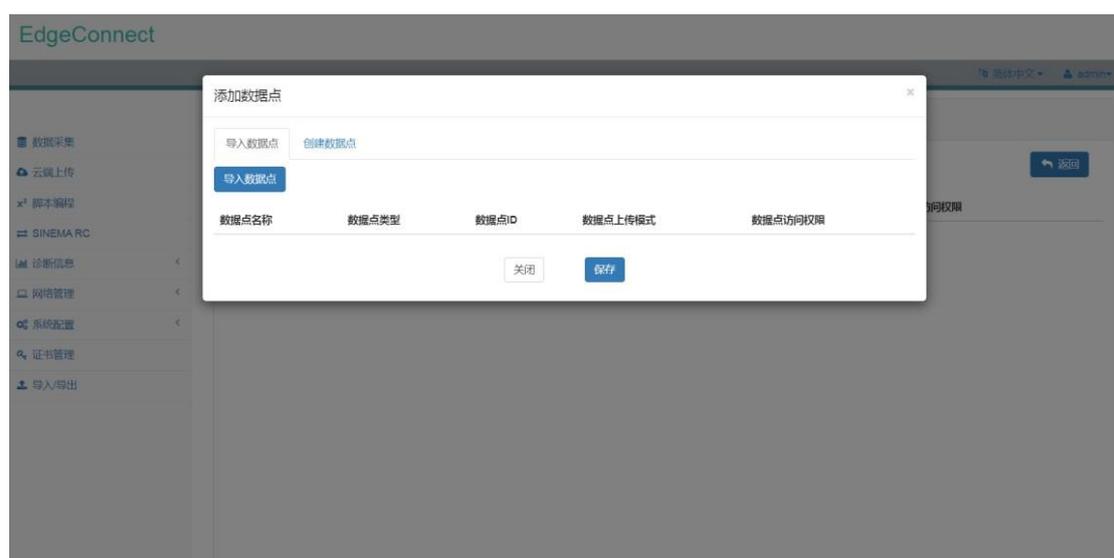
配置好 Modbus_TCP_Server 通道之后，Modbus_TCP_Server 通道信息会以列表形式显示在页面中。



Modbus_TCP_Server 通道配置信息

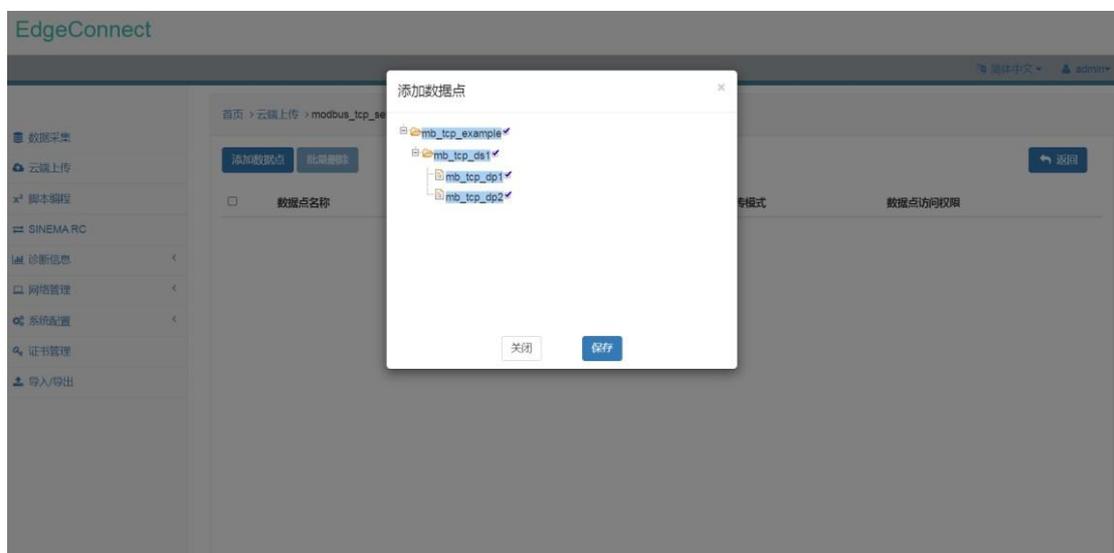
通道详情配置

点击 Modbus_TCP_Server 通道列表项右侧 , 即可进入通道详情配置页面。在通道详情配置页面，用户可以点击“添加数据点”打开数据点配置界面。



Modbus_TCP_Server 数据点配置界面

在数据点配置界面中，点击“导入数据点”，会出现所有数据采集通道、脚本编程通道及对应通道内数据点的信息。用户可以选择需要上传的数据点。



Modbus_TCP_Server 选择数据点界面

选中数据点并保存，会返回至数据点配置界面，并加载已选中数据点的信息。用户需要填入 Modbus_TCP_Server 对应的寄存器地址并选择数据点上传模式及读写权限。

数据点上传模式支持两种：

1) Continuous

连续上传模式，采集到的数据点连续不断上传。

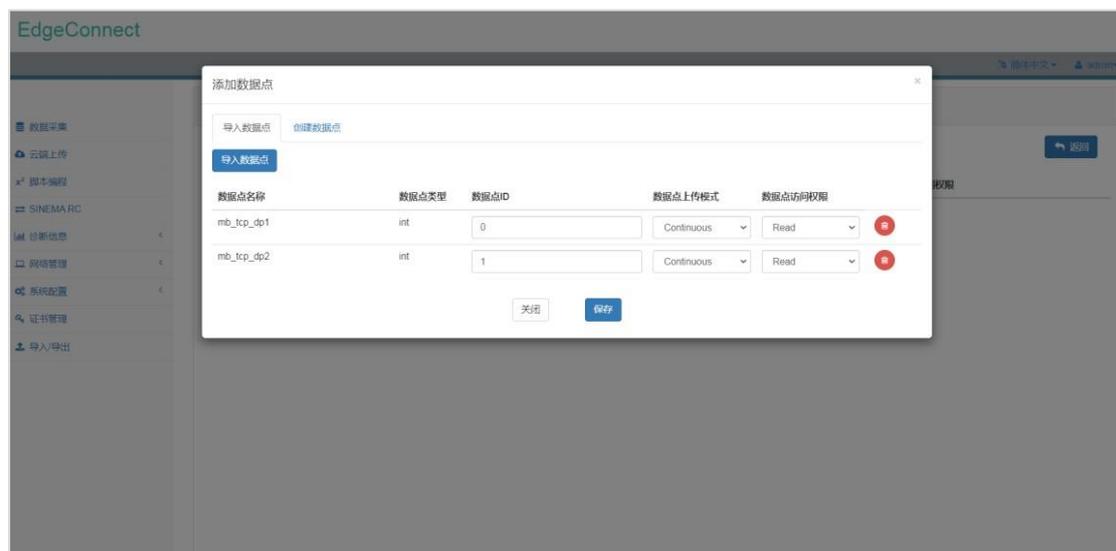
2) Onchange

逢变上传模式，采集到的数据点变化时上传。

读写权限仅支持只读权限：

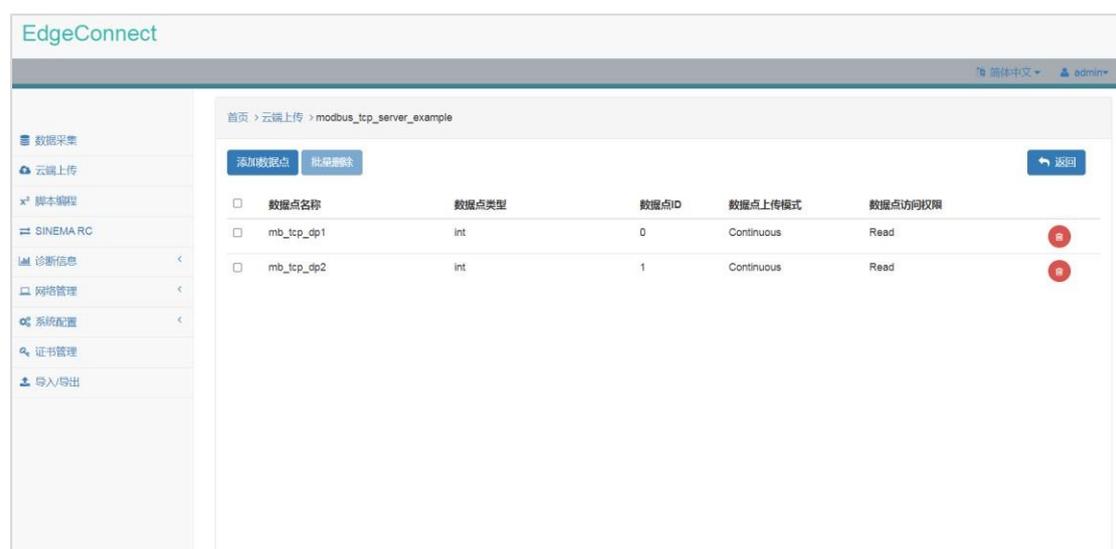
1) Read

只读权限，仅支持读取采集到的数据点，不允许将数据点的值写入数据采集通道。



Modbus_TCP_Server 数据点配置界面

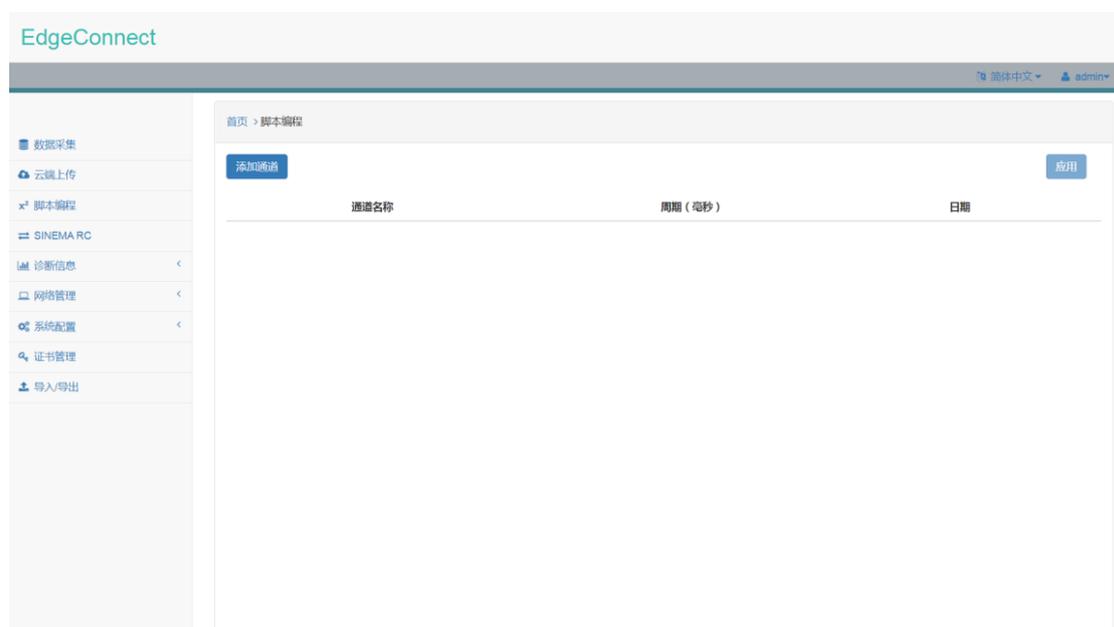
配置完数据点信息之后，数据点信息会以列表形式显示在页面上。



Modbus_TCP_Server 数据点配置信息

5.5 脚本编程（边缘计算）模块

目前脚本编程模块仅支持 Python 脚本编程。本章节将介绍如何使用脚本编程模块。



脚本编程模块界面

用户通过点击“添加通道”按钮，可以添加 Python 脚本通道，并进行相应的配置。

所有通道配置结束之后，将配置完成的通道状态由停止状态切换至运行状态（红色->绿色），并点击“应用”按钮即可下发脚本编程模块配置，并启动脚本程序。

Python 脚本通道

通道配置

设置通道名称、重复执行、延迟及周期即可完成相应配置。

1) 重复执行

重复执行支持以下选择：

- None
- Interval

其中None 为不重复执行，Interval 为重复执行。

2) 延迟（毫秒）

延迟支持以下选择：

- 100
- 200
- 500
- 1000
- 2000
- 5000
- 10000

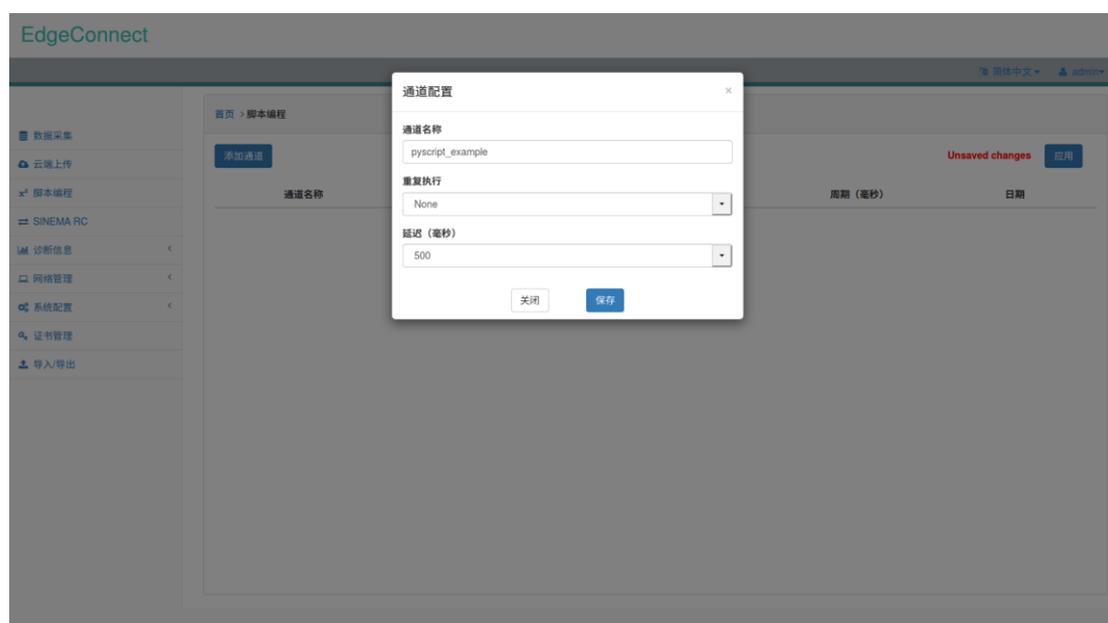
仅当重复执行为 None 时有效。

3) 周期（毫秒）

周期支持以下选择：

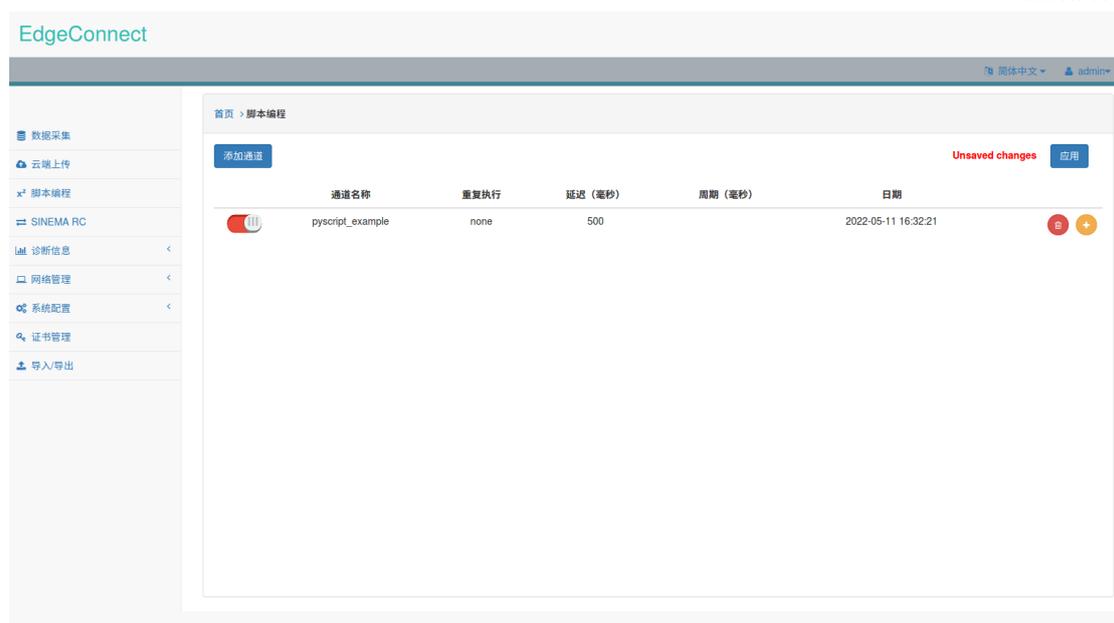
- 100
- 200
- 500
- 1000
- 2000
- 5000
- 10000

仅当重复执行为 Interval 时有效。



Python 脚本通道配置界面

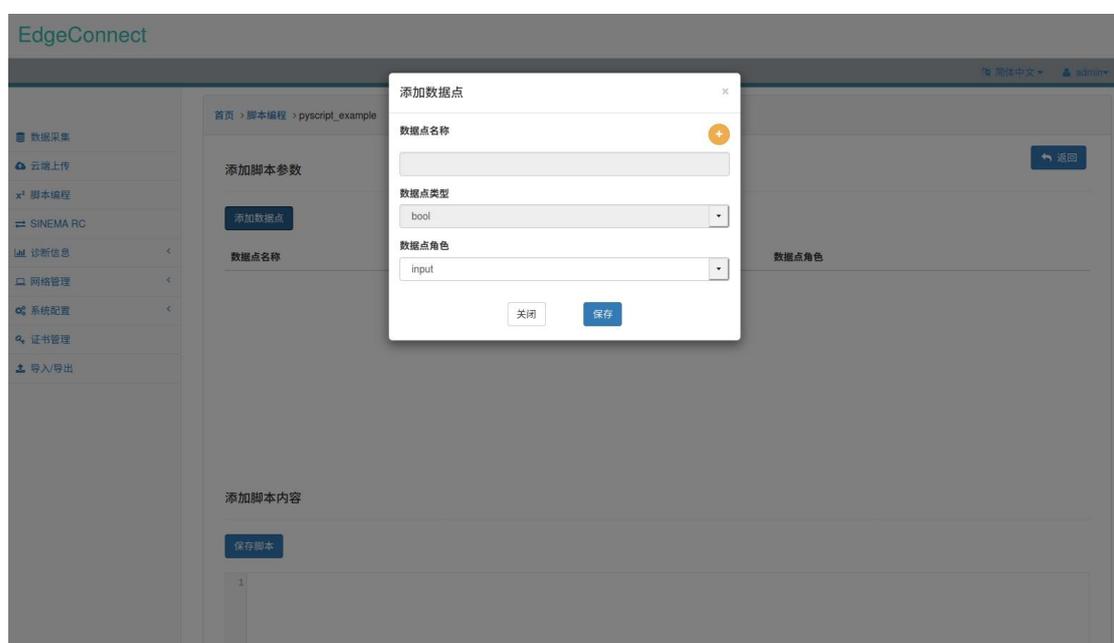
配置好 Python 脚本通道之后，Python 脚本通道信息会以列表形式显示在页面中。



Python 脚本通道配置信息

通道详情配置

点击 Python 脚本通道列表项右侧 ，即可进入通道详情配置页面。在通道详情配置页面，用户可以点击“添加数据点”打开脚本参数配置界面。

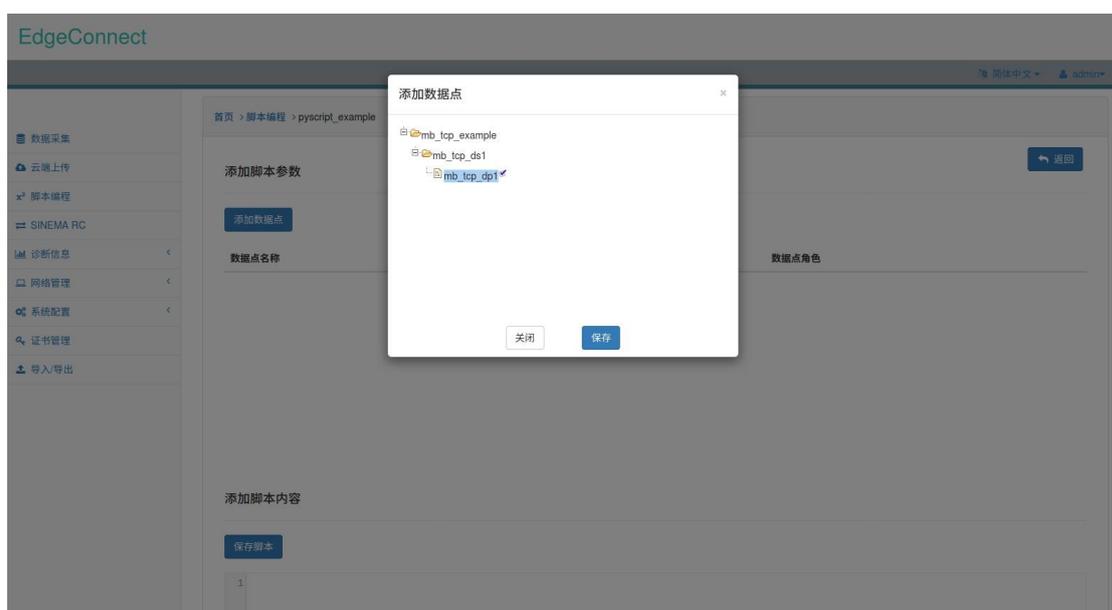


Python 脚本参数配置界面

通过切换数据点角色为 input 或 output，可以配置 Python 脚本的输入输出参数。

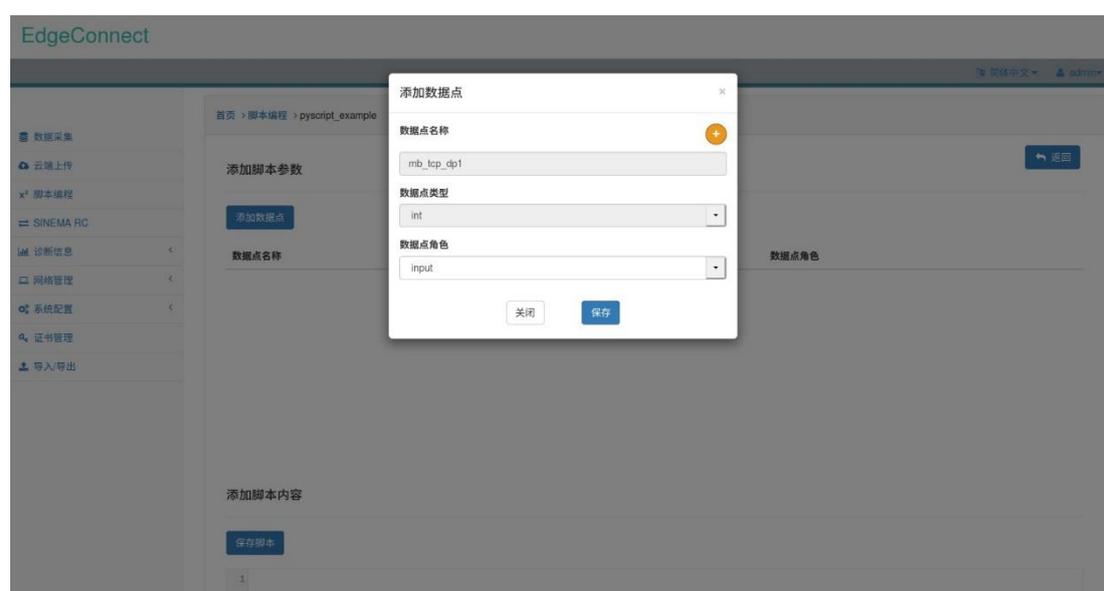
1) 配置脚本输入参数

切换数据点角色为 input。在脚本参数配置界面中，点击右上角 ，会出现所有采集通道及采集通道数据点的信息。用户可以选择需要处理的采集通道数据点。



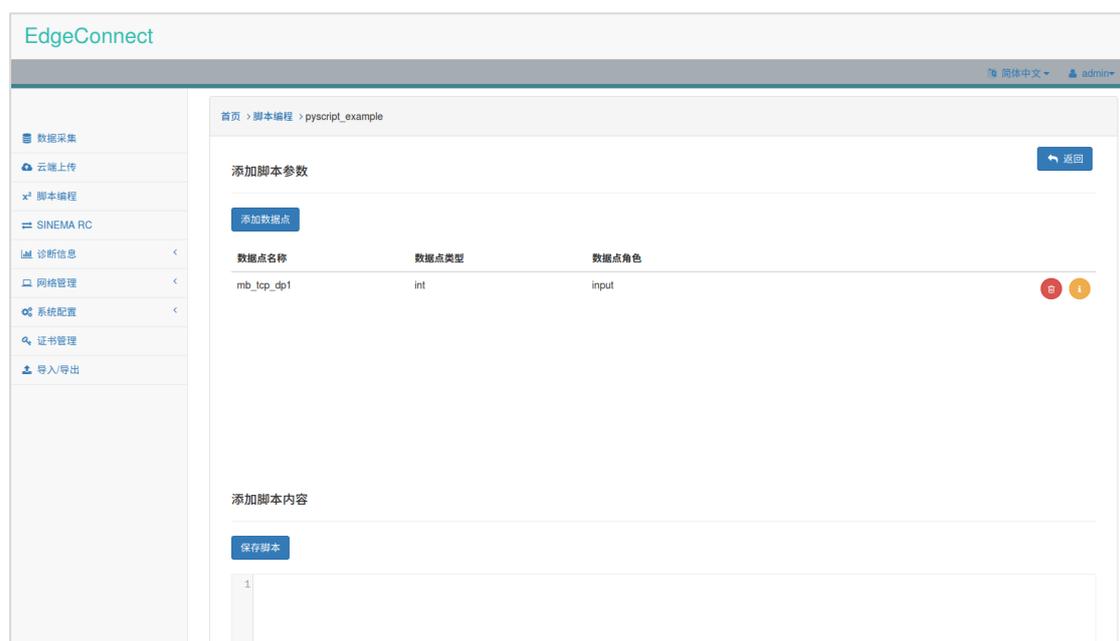
Python 脚本选择采集通道数据点界面

选中采集通道数据点并保存，会返回至脚本参数配置界面，并加载已选中采集通道数据点的信息。



Python 脚本输入参数配置界面

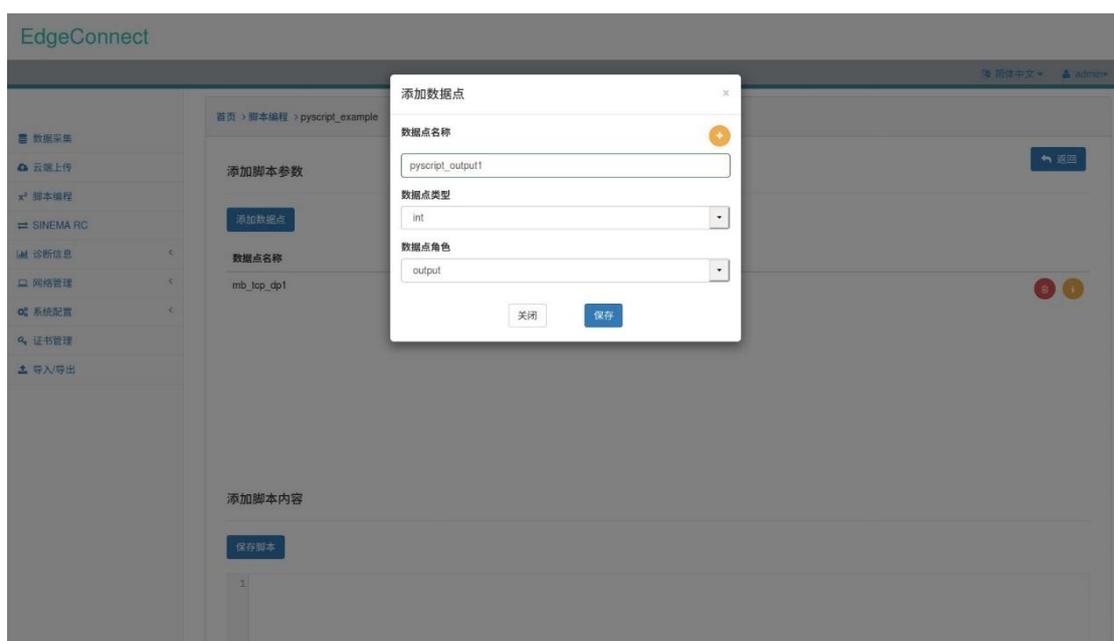
配置完输入参数信息之后，输入参数信息会以列表形式显示在页面上。



Python 脚本输入参数配置信息

2) 配置脚本输出参数

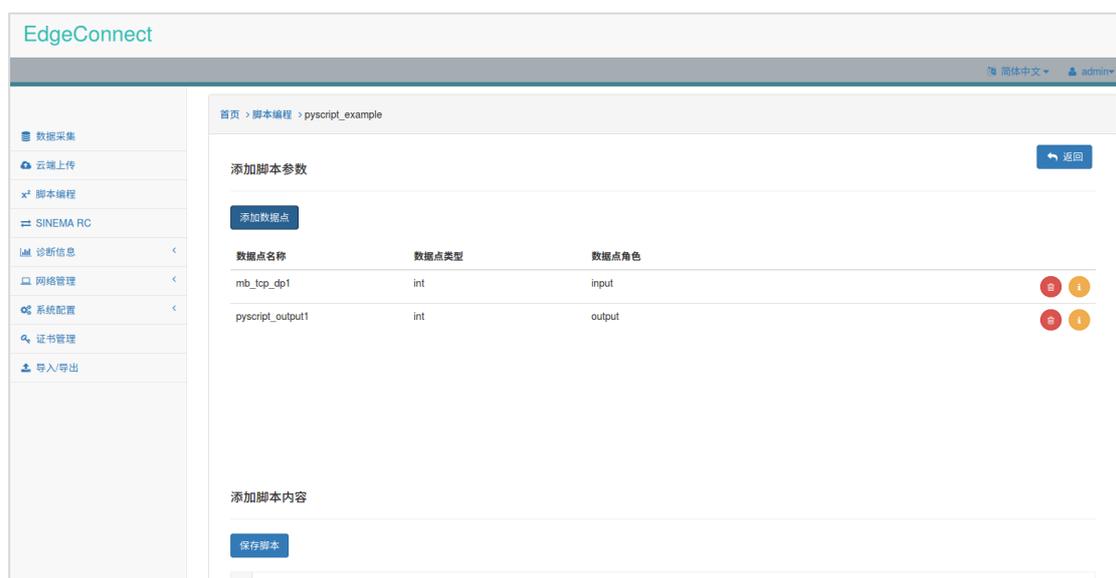
切换数据点角色为 output。在脚本参数配置界面中，填入数据点名称，并选择相应的类型。



Python 脚本输出参数配置界面

配置完输出参数信息之后，输出参数信息会以列表形式显示在页面上。

通道配置应用下发后，点击数据点列表项右侧，可查看数据点当前最新值。



Python 脚本输出参数配置信息

Python 脚本的输入输出参数配置好之后，在添加脚本内容区域编辑框内可输入自定义脚本内容。

开发自定义脚本需遵循以下规则：

a) 遵循 Python3 语法

b) 脚本可导入 Python3 标准库或 edgeconnect 模块，其中 edgeconnect 模块提供如下函数：

- `getVal("dpname")`，其中 `dpname` 为角色为 `input` 的数据点名称
- `setVal("dpname", dpval)`，其中 `dpname` 为角色为 `output` 的数据点名称，`dpval` 为待设置的值
- `debug("msg")`，其中 `msg` 为需要打印的 `debug` 信息

c) 脚本必须定义main()函数， main()函数为脚本执行的默认入口。

脚本内容编辑好之后， 点击“保存脚本”即可保存脚本内容。脚本参考示例如下所示：

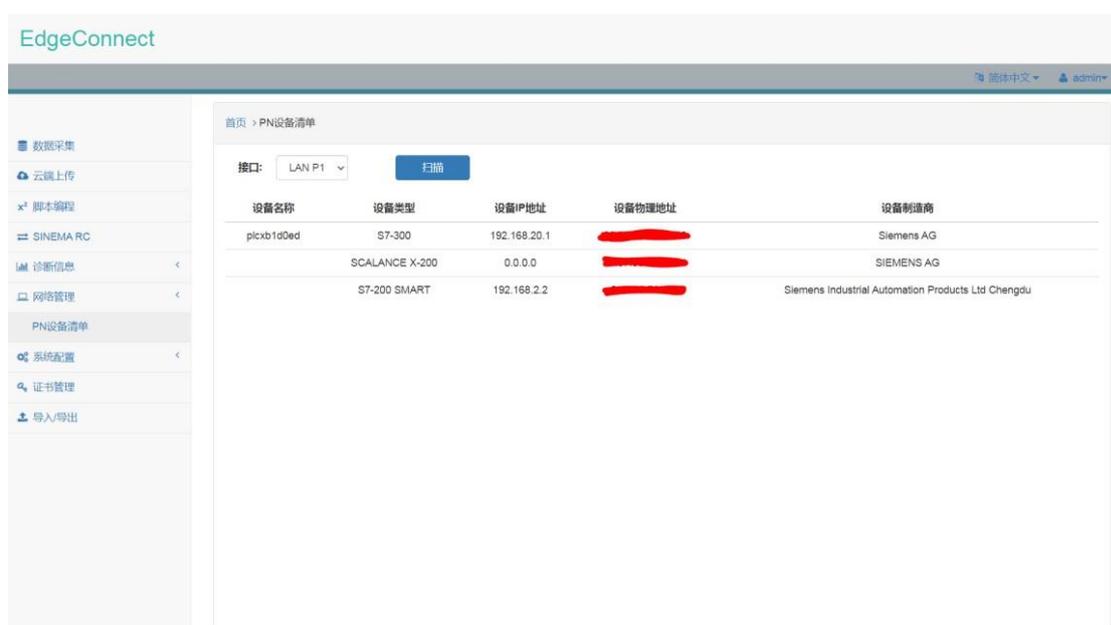
```
添加脚本内容
保存脚本
1 from edgeconnect import getVal, setVal, debug
2
3 def main():
4     dp1 = getVal("mb_tcp_dp1")
5
6     # sum example
7     dp2 = dp1 + 100
8     setVal("pyscript_output1", dp2)
9
10    # print debug information
11    msg = "pyscript_output1 is " + str(dp2)
12    debug(msg)
```

Python 脚本参考示例

5.6 网络管理模块

PN 设备清单

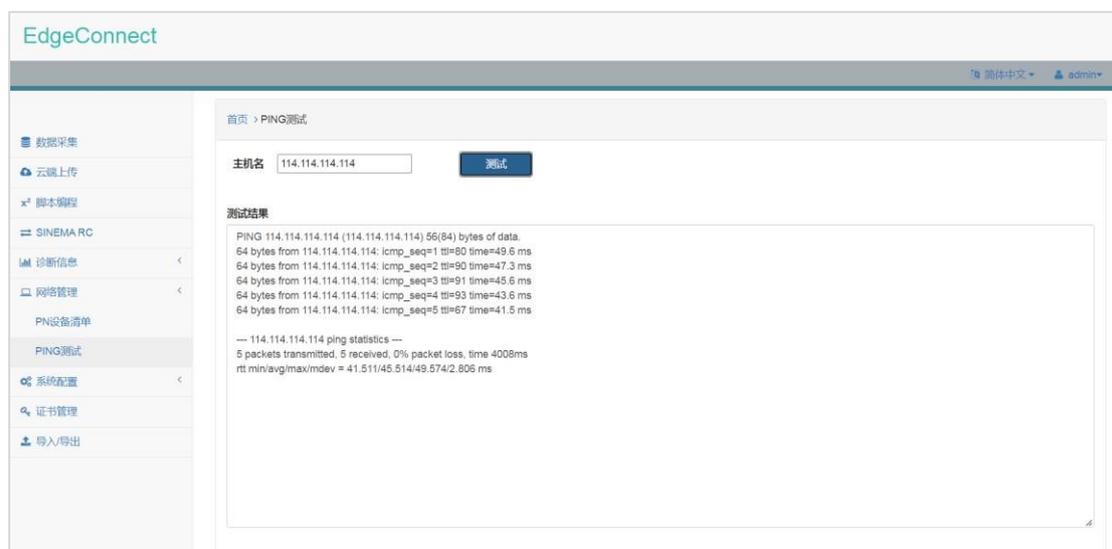
用户可以对指定端口所在的局域网进行扫描，并显示该局域网中的所有 PROFINET 设备信息，包括设备名称、设备类型、设备 IP 地址、设备 MAC 地址、设备制造商等信息。



PN 设备清单界面

Ping 测试

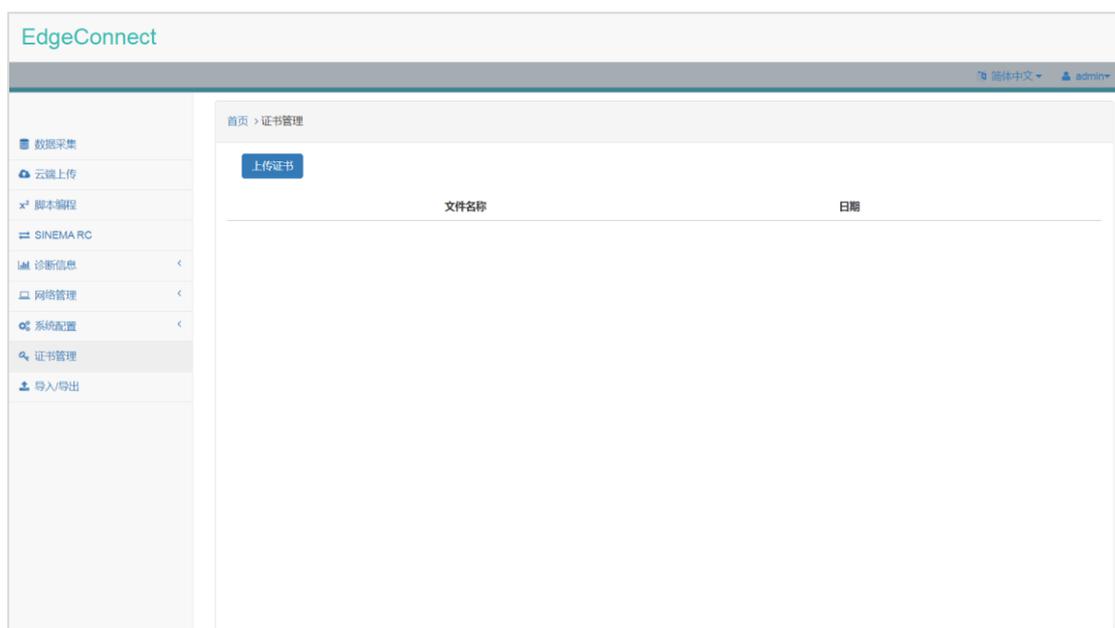
用户可以对指定 IP 或域名进行 ping 测试，并显示测试结果。



Ping 测试界面

5.7 证书管理

证书管理模块支持导入、导出及删除证书。导入证书后，可以在其他模块中选择该证书。证书后缀为.crt、.pem、.key、.der 及.cer。



证书管理界面

5.8 导入/导出

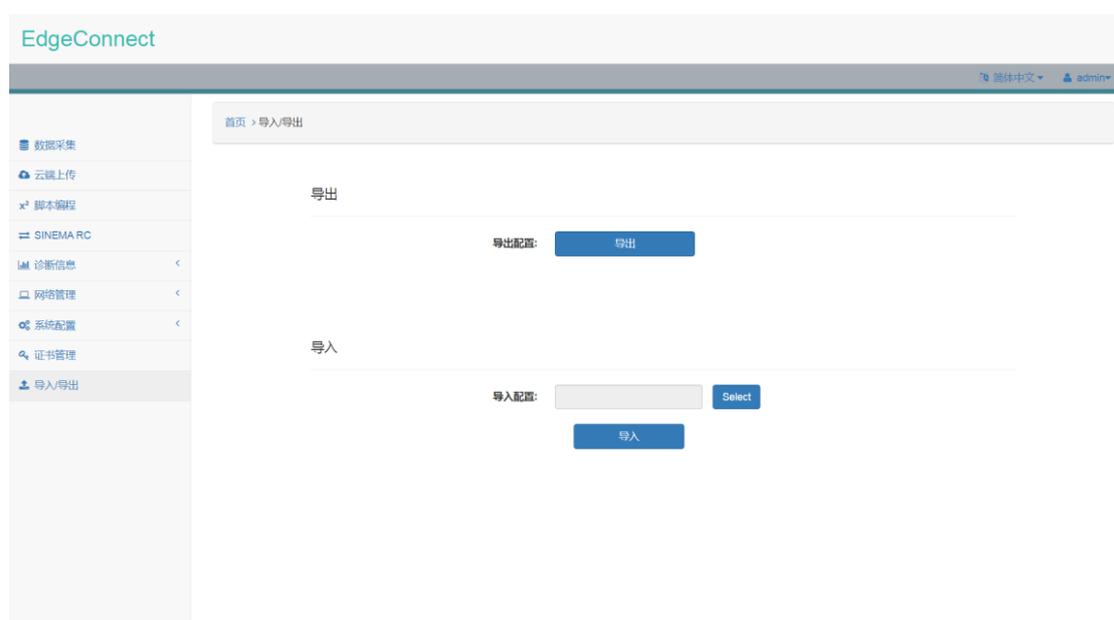
导入/导出模块支持将数据采集模块、云端上传模块以及脚本编程模块的配置信息导入或导出。

1) 导出

点击“导出”按钮，将以文件形式导出数据采集模块、云端上传模块以及脚本编程模块的配置信息，文件名为 edgeconnect_configuration.bak。

2) 导入

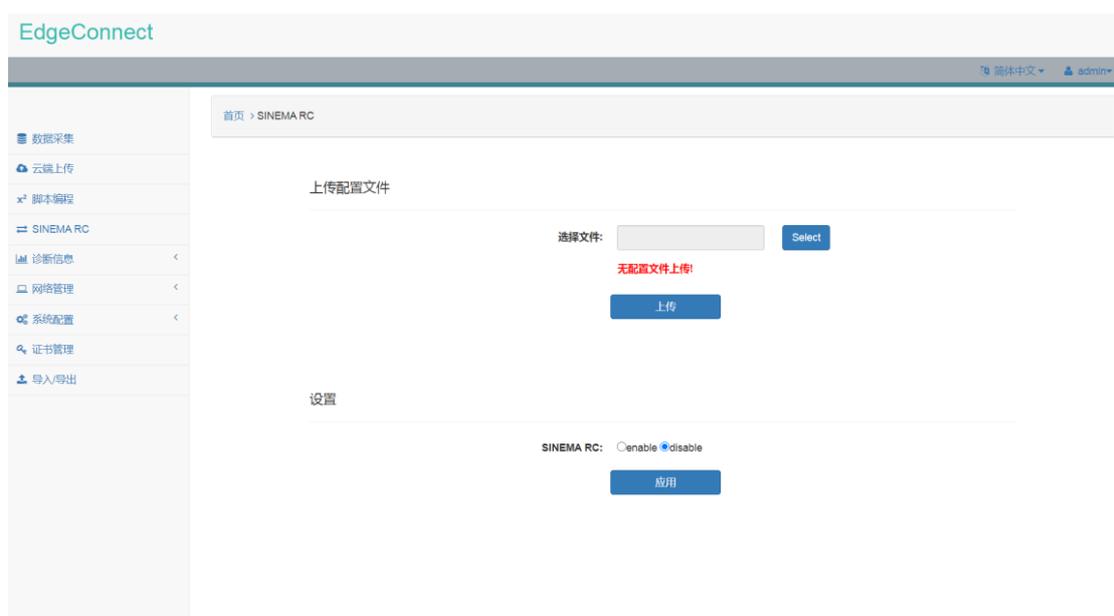
点击“Select”按钮，选择已导出的配置文件，并点击“导入”按钮，执行导入。



导入/导出界面

5.9 SINEMA RC 模块

SINEMA RC 模块支持西门子 SINEMA RC 接入功能，可以接入到 SINEMA RC 服务器，实现对现场设备的远程上传、下载及在线诊断等功能。



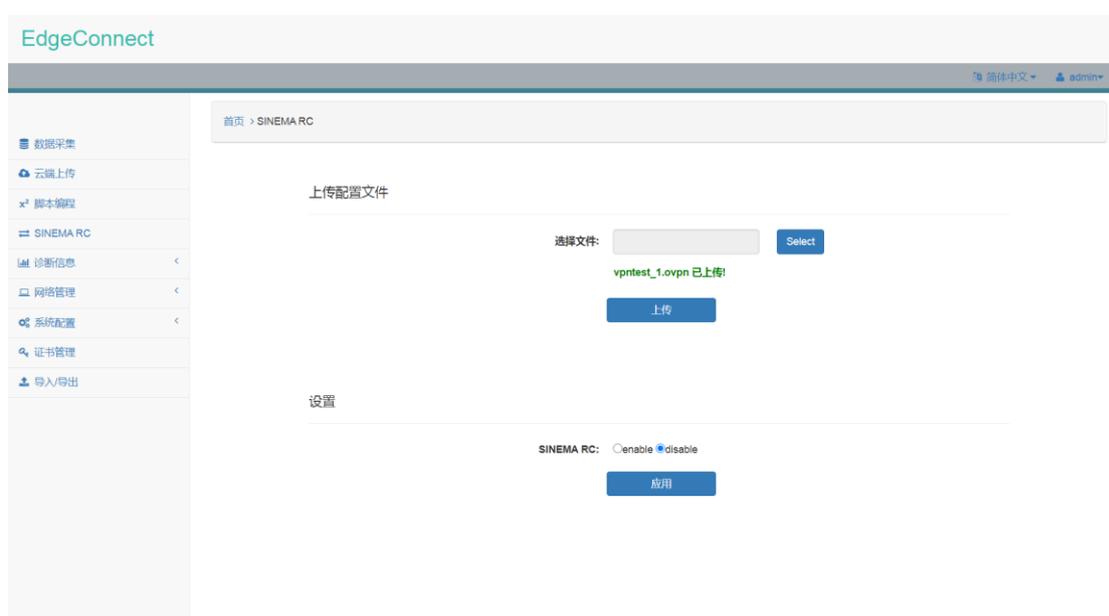
SINEMA RC 模块配置界面

通过本模块与 SINEMA RC 服务器建立连接后，还需要进行路由配置及网络转换配置，关于这一部分的配置，请参考章节 5.9.1 及 5.9.2。

5.9.1 上传 SINEMA RC 设备配置文件

关于 SINEMA RC 服务器的配置请参考相关文档。在 SINEMA RC 服务器上配置好 SINEMA RC 设备信息之后，可以导出.ovpn 的设备配置文件。

点击“Select”弹出文件选择界面，选择.ovpn 设备配置文件，点击“上传”进行文件上传。配置文件上传成功之后会提示上传结果。



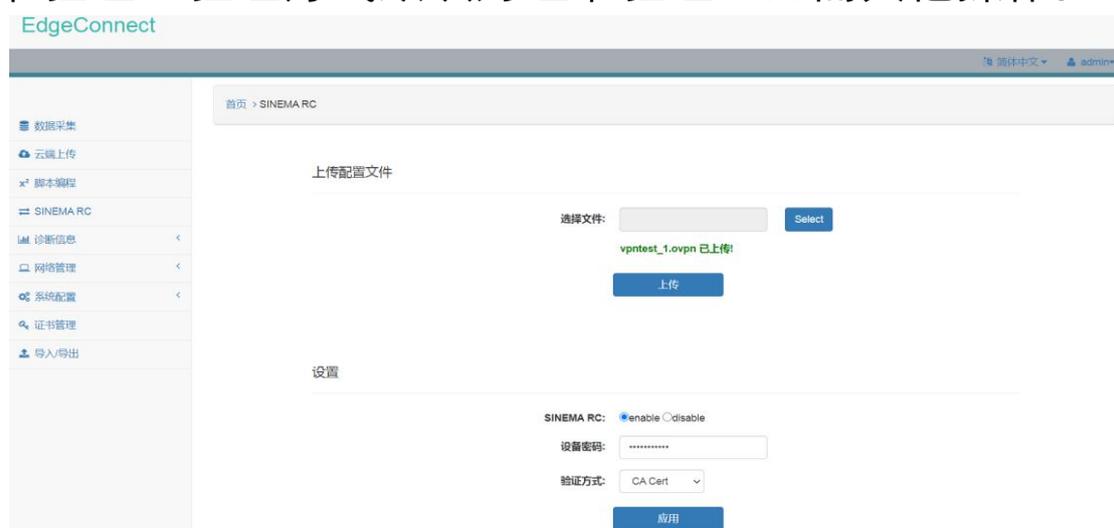
上传 SINEMA RC 设备配置文件

5.9.2 设置 SINEMA RC 设备密码及验证方式

将 SINEMA RC 选项设置为 enable 状态，并在“设备密码”输入框中输入在 SINEMA RC 管理平台上设置的设备密码。

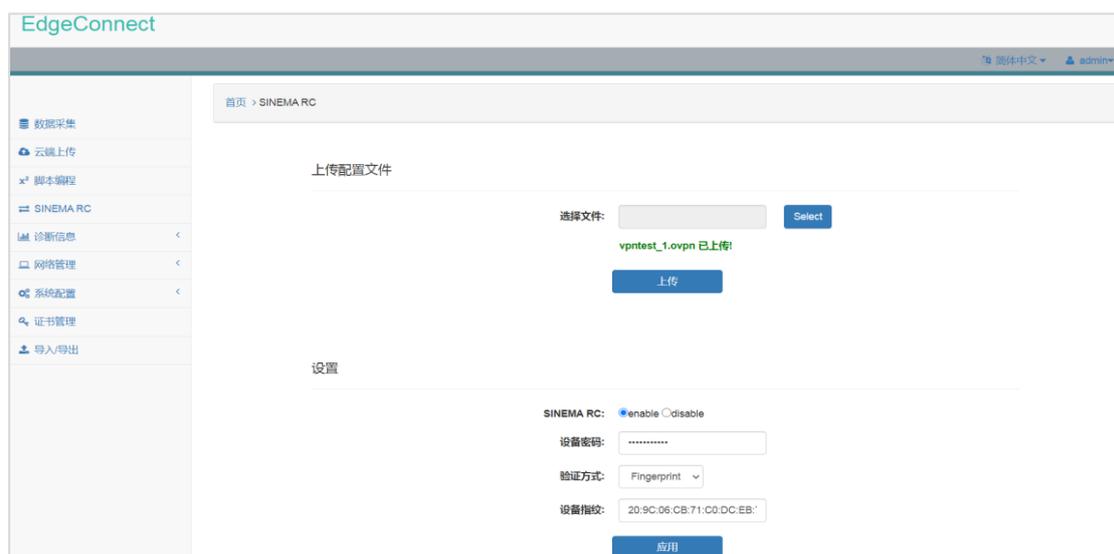
验证方式支持两种：证书验证或指纹验证。使用指纹验证在证书过期的情况下仍然能够连接至 SINEMA RC VPN 服务器。

1) 证书验证，验证方式默认为证书验证，无需其他操作。



设置 SINEMA RC 设备密码及证书验证

2) 指纹验证指纹验证需要输入 SINEMA RC 服务器 VPN Server 的指纹信息，指纹信息需为 SHA256 格式。

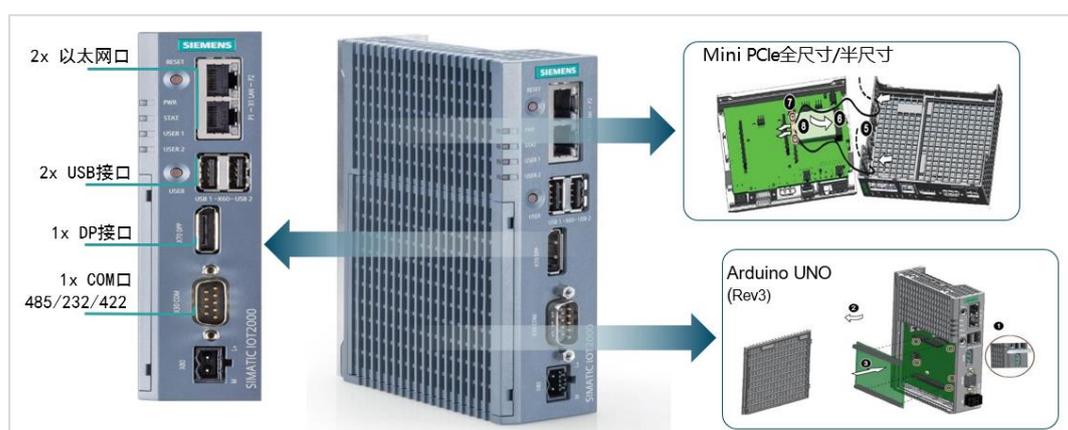


设置 SINEMA RC 设备密码及指纹验证

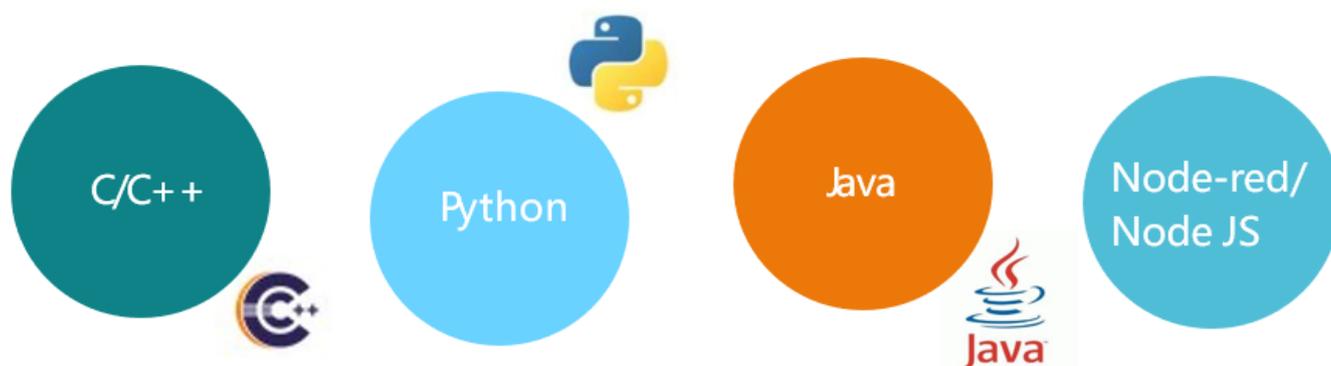
配置完成 SINEMA RC 设备密码及验证信息后，点击“应用”即可启动 SINEMA RC 接入服务

6.其他应用方式

SIMATIC IOT2050 具有极高的开放性和可扩展性，SIMATIC IOT2050 带有多种硬件接口，包括：以太网口，USB 口，DP 图形显示接口和串口以及 Arduinio Shield I 接口和 MINIPCIE 接口。通过这些接口和扩展接口，用户可以扩展多种应用，包括扩展 Arduinio Shield IO 扩展板，4G，5G，Wifi，Lora 等无线应用。



SIMATIC IOT2050 的 Example IMG 镜像向用户开放了 root 权限，用户可以自行安装和开发各种软件应用。



6.1 扩展 4G, 5G 应用

通过 SIMATIC IOT2050 的 MINIPCIE 接口用户可以通过通过这些接口来扩展 4G 无线应用。

在连接 4G 无线网络前，需要提前准备好扩展 4G 网络连接所需的硬件，包括：

- 1) 支持 miniPCIE 接口的 4G 模组，
- 2) 天线
- 3) 连接模组与天线的馈线
- 4) 4G 物联网 sim 卡

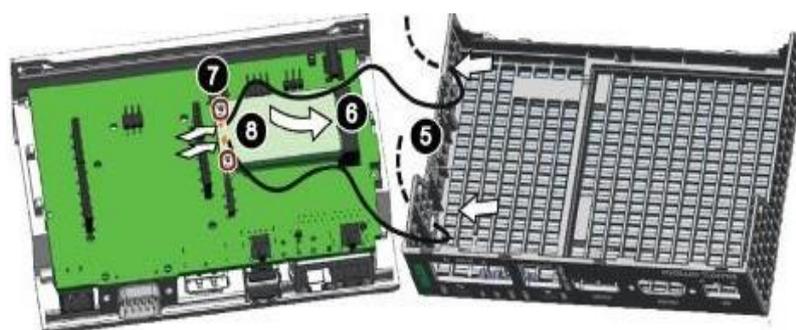


相关扩展硬件的推荐型号可以参考下图。

品牌	类型	型号	照片
Quectel (移远)	4G模组	EC20CEFRSG-MINIPCIE	
	天线	YE0009AA	
	馈线	YM0003AA	

IOT2050 中预装了列表中 4G 模组的驱动，安装后可以自动识别。其他品牌和型号的 4G 模组 IOT2050 产品也可以支持，但有可能需要用户自行为模组安装驱动程序。

取下 IOT2050 的端盖后，可以通过电路板上的 miniPICE 安装 4G 模组，端盖侧面留有天线预留孔，可用于安装天线及馈线。



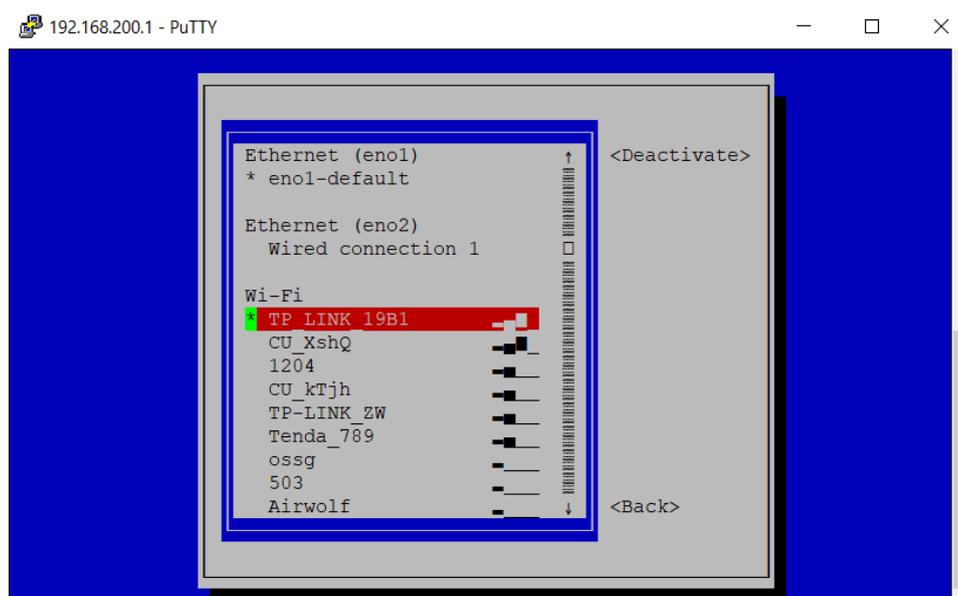
安装好相关的硬件，在卡槽中插入 SIM 卡，启动 IOT2050 后就可以连接 4G 网络了。

6.2 扩展 WIFI 应用

SIMATIC IOT 2050 带有 2 个 USB 接口，可以扩展支持 USB 的 Wifi 模块，SIMATIC IOT2050 所带的 Example img 系统开放 root 权限，用户可以为 Wifi 模块安装驱动程序。Example img 所带系统中也集成了部分 Wifi 芯片的驱动程序，用户连接 Wifi 模块，进行设置后即可连接使用。



安装完成，启动 IOT2050 后，可以通过参考章节“3.2.2 Networking 设置”中的内容，选择对应的 Wifi 信号进行连接。



如果使用的是 Edgeconnect 镜像，请参考章节“[5.2.1 接口配置](#)”。

6.3 安装桌面及浏览器

西门子提供的系统镜像中的 Linux 系统，不带有桌面和浏览器，用户智能通过其他 PC 桌面的浏览器访问来展示系统镜像中应用的 UI 用户界面。

由于 SIMATIC IOT2050 的系统镜像可以向用户开放 root 权限，用户可以为系统安装桌面及浏览器，从而直接通过 IOT2050 的 DP 图形显示接口连接显示器，展示 IOT2050 中运行应用的界面效果，实现更多应用场景。

为 SIMATIC IOT2050 安装桌面及浏览器的方法如下：

1) 准备环境

系统版本：Example Image V1.1.1

硬件：IOT2050 Advanced / Basic，主动式 DP 转 HDMI 连接线，HDMI 接口显示器，有线键盘及鼠标

电脑：安装好 putty 等工具

2) 硬件连接

1) 使用 DP 转 HDMI 线连接 IOT2050 及显示器

2) 连接鼠标及键盘至 IOT2050

3) 将 IOT2050 连接至 Internet，便于安装软件

3) 安装桌面及 chromium 浏览器

a) 更新源

```
sed -i 's#http://deb.debian.org#https://mirrors.aliyun.com#g'  
/etc/apt/sources.list.d/bootstrap.list  
apt update
```

b) 安装 lxde

```
apt install lxde  
apt install lxde.desktop  
apt install xfce4-power-manager
```

c) 安装 chromium

```
apt install chromium
```

d) 添加新用户

```
groupadd iot2050  
useradd -m iot2050 -g iot2050 -s /bin/bash -d /home/iot2050  
passwd iot2050
```

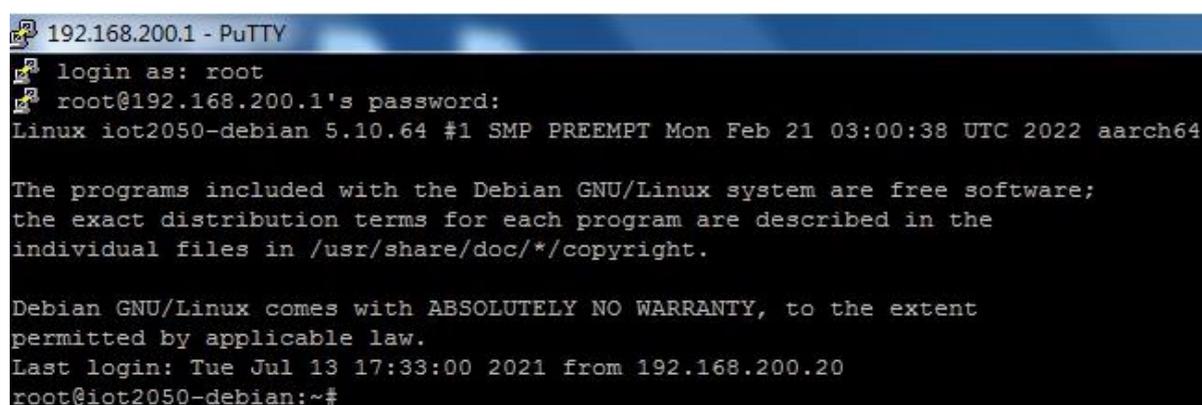
连续输入两次密码并确定。

完成以上步骤后重启设备后，就可以通过前面设置的用户名和密码登陆桌面，运行桌面上的浏览器了。

6.4 安装 MariaDB 数据库

用户在使用 IOT2050 的过程中，如果需要使用数据库对采集的数据进行处理，可在 IOT2050 的 linux 系统中安装数据库程序，本章节以 MariaDB 的数据库安装为例，来介绍数据库的安装方法。

1. 启动 IOT2050，登陆 root 进入 linux 系统，将 IOT2050 与网络进行链接（参考章节 3.2.2）。



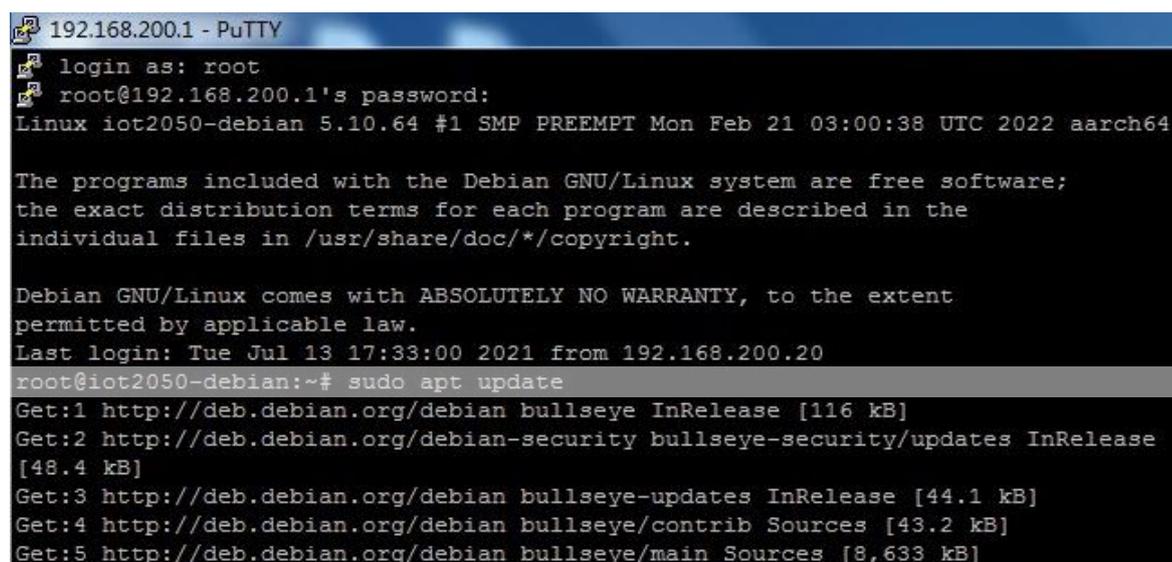
```
192.168.200.1 - PuTTY
login as: root
root@192.168.200.1's password:
Linux iot2050-debian 5.10.64 #1 SMP PREEMPT Mon Feb 21 03:00:38 UTC 2022 aarch64

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: Tue Jul 13 17:33:00 2021 from 192.168.200.20
root@iot2050-debian:~#
```

2. 输入下面代码，更新包索引：

```
sudo apt update
```



```
192.168.200.1 - PuTTY
login as: root
root@192.168.200.1's password:
Linux iot2050-debian 5.10.64 #1 SMP PREEMPT Mon Feb 21 03:00:38 UTC 2022 aarch64

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: Tue Jul 13 17:33:00 2021 from 192.168.200.20
root@iot2050-debian:~# sudo apt update
Get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:2 http://deb.debian.org/debian-security bullseye-security/updates InRelease
[48.4 kB]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 http://deb.debian.org/debian bullseye/contrib Sources [43.2 kB]
Get:5 http://deb.debian.org/debian bullseye/main Sources [8,633 kB]
```

注：如果网络不稳定，会导致升级不完整，无法顺利完成后续安装操作。

3. 通过运行以下命令安装 MariaDB 服务器和客户端的软件包：

```
sudo apt install mariadb-server
```

```
192.168.200.1 - PuTTY
128 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@iot2050-debian:~# sudo apt install mariadb-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  galera-4 libaiol libconfig-inifiles-perl libdbi-perl libmariadb3 libpopt0
  lsof mariadb-client-10.5 mariadb-client-core-10.5 mariadb-common
  mariadb-server-10.5 mariadb-server-core-10.5 mysql-common rsync socat
Suggested packages:
  libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl mailx
  mariadb-test netcat-openbsd
Recommended packages:
  libdbd-mariadb-perl | libdbd-mysql-perl libterm-readkey-perl
  libhtml-template-perl
The following NEW packages will be installed:
  galera-4 libaiol libconfig-inifiles-perl libdbi-perl libmariadb3 libpopt0
  lsof mariadb-client-10.5 mariadb-client-core-10.5 mariadb-common
  mariadb-server mariadb-server-10.5 mariadb-server-core-10.5 mysql-common
  rsync socat
0 upgraded, 16 newly installed, 0 to remove and 128 not upgraded.
Need to get 15.5 MB of archives.
After this operation, 152 MB of additional disk space will be used.
```

(安装过程中，系统会显示程序占用空间，确认是否继续安装)

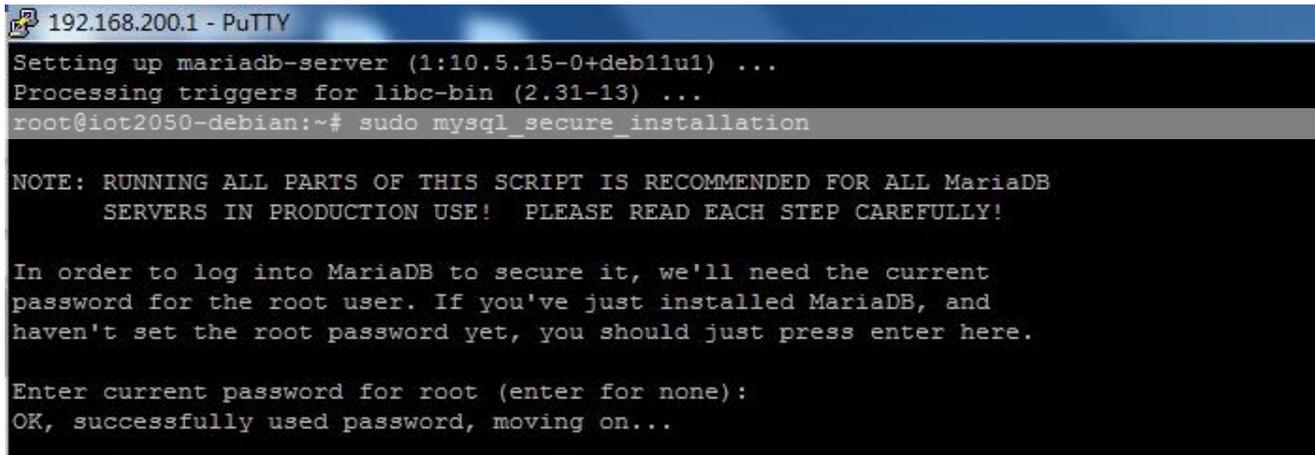
Do you want to continue? [Y/n] Y

```
192.168.200.1 - PuTTY
0 upgraded, 16 newly installed, 0 to remove and 128 not upgraded.
Need to get 15.5 MB of archives.
After this operation, 152 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bullseye/main arm64 mysql-common all 5.8+1.0.
7 [7,464 B]
Get:2 http://deb.debian.org/debian bullseye/main arm64 mariadb-common all 1:10.5
.15-0+deb11u1 [36.7 kB]
Get:3 http://deb.debian.org/debian bullseye/main arm64 galera-4 arm64 26.4.11-0+
deb11u1 [692 kB]
Get:4 http://deb.debian.org/debian bullseye/main arm64 libdbi-perl arm64 1.643-3
+b1 [774 kB]
Get:5 http://deb.debian.org/debian bullseye/main arm64 lsof arm64 4.93.2+dfsg-1.
1 [313 kB]
Get:6 http://deb.debian.org/debian bullseye/main arm64 libconfig-inifiles-perl a
ll 3.000003-1 [52.1 kB]
Get:7 http://deb.debian.org/debian bullseye/main arm64 libmariadb3 arm64 1:10.5.
15-0+deb11u1 [166 kB]
Get:8 http://deb.debian.org/debian bullseye/main arm64 mariadb-client-core-10.5
arm64 1:10.5.15-0+deb11u1 [764 kB]
Get:9 http://deb.debian.org/debian bullseye/main arm64 mariadb-client-10.5 arm64
1:10.5.15-0+deb11u1 [1,492 kB]
Get:10 http://deb.debian.org/debian bullseye/main arm64 libaiol arm64 0.3.112-9
[12.3 kB]
```

4. 通过运行以下命令安装安全脚本，设置 root 用户密码：

```
sudo mysql_secure_installation
```

(此时系统会要求输入 root 密码，初始时密码为空，直接按 enter 键即可)



```
192.168.200.1 - PuTTY
Setting up mariadb-server (1:10.5.15-0+deb11u1) ...
Processing triggers for libc-bin (2.31-13) ...
root@iot2050-debian:~# sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...
```

接下来系统会出现一系列选择设置，操作如下：

```
Switch to unix_socket authentication [Y/n] n
```

```
...
```

```
Change the root password? [Y/n] Y
```

```
New password:
```

```
Re-enter new password:
```

```
...
```

```
Remove anonymous users? [Y/n] Y
```

```
...
```

```
Disallow root login remotely? [Y/n] Y
```

```
...
```

```
Remove test database and access to it? [Y/n] Y
```

```
...
```

```
Reload privilege tables now? [Y/n] Y
```

```
...
```

```
Thanks for using MariaDB!
```

```
192.168.200.1 - PuTTY
Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

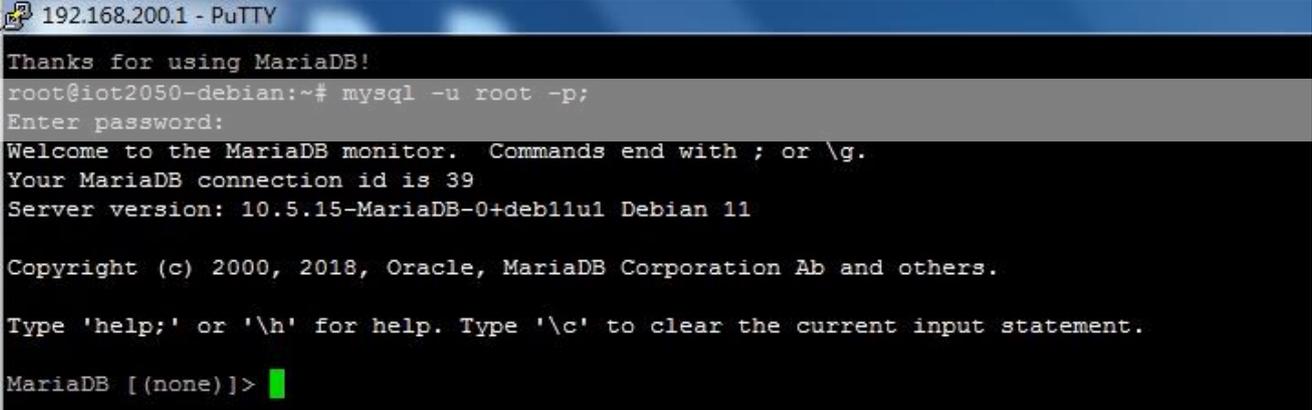
Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

5. 输入下面命令并输入前面设置好的密码后，就可以以 root 用户身份登陆和使用数据库了：

```
mysql -u root -p;
```



```
192.168.200.1 - PuTTY
Thanks for using MariaDB!
root@iot2050-debian:~# mysql -u root -p;
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.5.15-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```