

一体式温控器

TC1-8R-MP、TC1-8T-MP 使用手册

(EIP 协议)

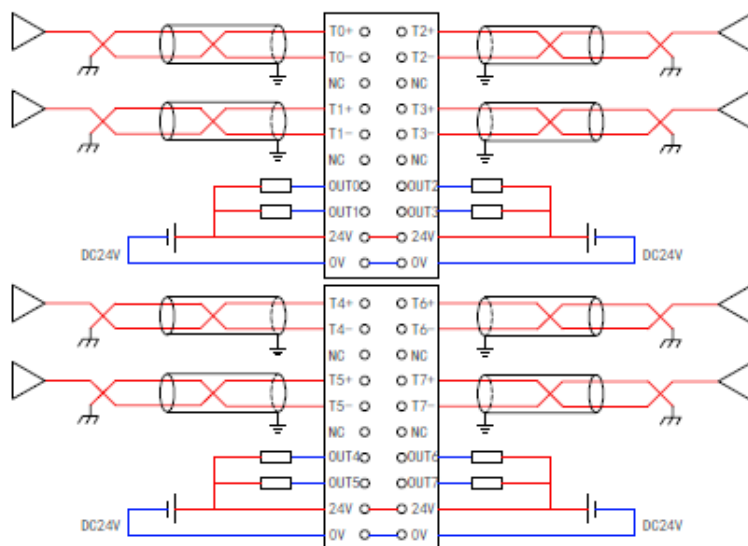
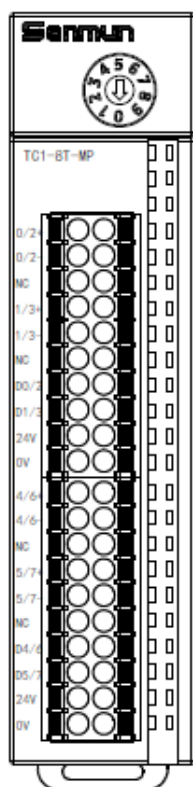
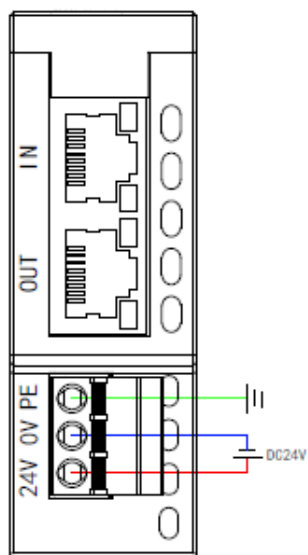
1. 产品规格

TC1-8T-MP、TC1-8R-MP 是一款智能 PID 温控模块。最大支持 8 通道PID加热控制。具有 8 路热电偶（热电阻）温度采集、支持Profinet、Ethernet/IP、CC-Link IE、ModbusTCP 通信，能与市面上常见的PLC，运动控制卡等控制器搭配使用，例如欧姆龙，汇川，倍福等

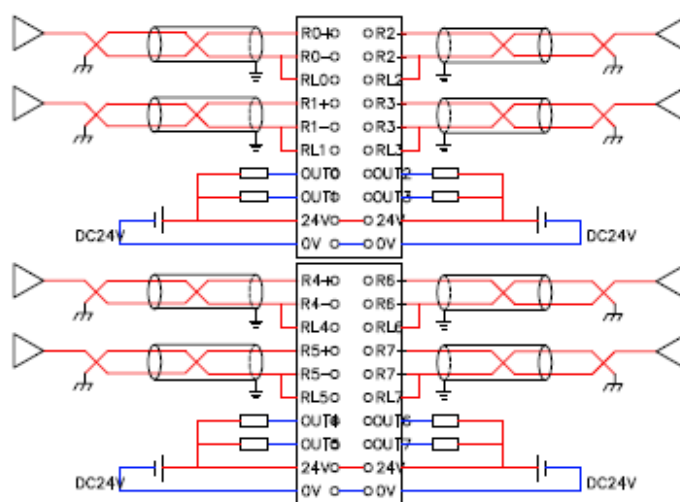
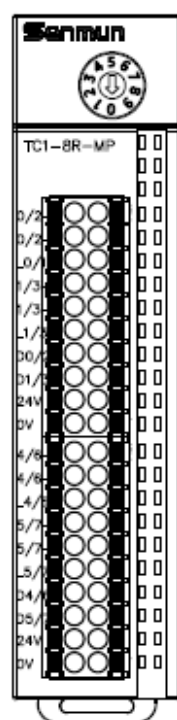
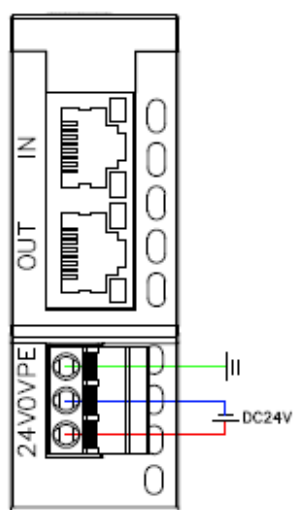


2. 模块接线图

TC1-8T-MP接线图



TC1-8T-MP接线图



3. 配置参数定义

名称	含义	数据类型	
Sensor_Type_CH[n]	传感器类型	枚举	详细见下表1
Kp_CH1--CH[n]	比例	INT	默认0，推荐使用自整定值
Ti_CH1--CH[n]	积分	INT	默认0，推荐使用自整定值
Td_CH1--CH[n]	微分	INT	默认0，推荐使用自整定值
PWM_Period_CH[n]	PWM周期	UINT	必须设置，推荐10
Filter_Algorithm_CH[n]	滤波算法	USINT	默认0，推荐使用默认值
Filter_Size_CH[n]	滤波窗口	UINT	推荐设置值：5
Filter_Amplitude_CH[n]	滤波振幅	UINT	推荐设置值：20
Hold_CH[n]	输出保持	USINT	0：通讯断开后模块通道停止加热 1：通讯断开后模块通道继续加热 (各个通道独立设置)

表1：

1) 热电偶传感器代号，例如0值表示K型热电偶

0: Sensor_Type_K
1: Sensor_Type_J
2: Sensor_Type_T
3: Sensor_Type_E
4: Sensor_Type_N
5: Sensor_Type_S
6: Sensor_Type_R
7: Sensor_Type_B
8: Sensor_Type_C
9: MA

2) 热电阻传感器代号，例如0值表示PT100型热电阻

0: PT100
1: PT200
2: PT500
3: PT1000
4: Ni120
5: 0~2000Ω(0~30000)

4、控制与反馈变量
下行数据

下行数据					
	名称	数据类型	说明		对象类型
通道 开关 命令	Enable_CHx	bit	bit0-bit7	8个通道使能开关	2*UINT
	Enable_PIDRunning_CHx		bit8-bit15	8个通道PID 运行使能开关	
	Enable_AutoMode_CHx		bit0-bit7	8个通道自整定使能开关	
	Enable_SaveParameter		bit8	SavePara： 参数保存， 将 Pwm_T、P_Buf、I_Buf、D_Buf 保存到flash， 上升沿有效	
			bit9-15	保留	
目标 温度	Temp_SV_CH1	int	有符号整数， 1 位小数位， 如写入值1500， 实际设置的是150.0 摄氏度。		8*INT
	Temp_SV_CH2				
	Temp_SV_CH3				
	Temp_SV_CH4				
	Temp_SV_CH5				
	Temp_SV_CH6				
	Temp_SV_CH7				
	Temp_SV_CH8				
补偿 温度	Tcomp_PV_CH1	int	用于人工修正测量温度与实际温度的偏差； 有符号整数， 1 位小数位， 如写入值20， 实际设置的是2.0 摄氏度， 范围：(-50)-(+50)， 大于±50 按±50 处理		8*INT
	Tcomp_PV_CH2				
	Tcomp_PV_CH3				
	Tcomp_PV_CH4				
	Tcomp_PV_CH5				
	Tcomp_PV_CH6				
	Tcomp_PV_CH7				
	Tcomp_PV_CH8				

上行数据

上行数据					
	名称	数据类型	说明		对象类型
当前温度	Current_Temp_CH1	int	当前温度ch0		8*INT
	Current_Temp_CH2		当前温度ch1		
	Current_Temp_CH3		当前温度ch2		
	Current_Temp_CH4		当前温度ch3		
	Current_Temp_CH5		当前温度ch4		
	Current_Temp_CH6		当前温度ch5		
	Current_Temp_CH7		当前温度ch6		
	Current_Temp_CH8		当前温度ch7		
通道当前状态	Status_PIDRunning_CHx	bit	bit0	PID运行中	8*UINT
	Status_AutoMode_Running_CHx		bit1	自整定中	
	Status_AutoMode_Completed_CHx		bit2	自整定完成	
	Status_AutoMode_Error_CHx		bit3	自整定错误	
	Status_SavePara_Complete_CHx		bit4	保持参数成功	
			bit5-bit7	保留	
Kp参数	Kp_CH1	int	比例		8*INT
	Kp_CH2				
	Kp_CH3				
	Kp_CH4				
	Kp_CH5				
	Kp_CH6				
	Kp_CH7				
	Kp_CH8				
Ti参数	Ti_CH1	int	积分		8*INT
	Ti_CH2				
	Ti_CH3				
	Ti_CH4				
	Ti_CH5				
	Ti_CH6				
	Ti_CH7				
	Ti_CH8				
Td 参数	Td_CH1	int	微分		8*INT
	Td_CH2				
	Td_CH3				
	Td_CH4				
	Td_CH5				
	Td_CH6				
	Td_CH7				
	Td_CH8				

5. 使用案例

汇川H5U使用案例

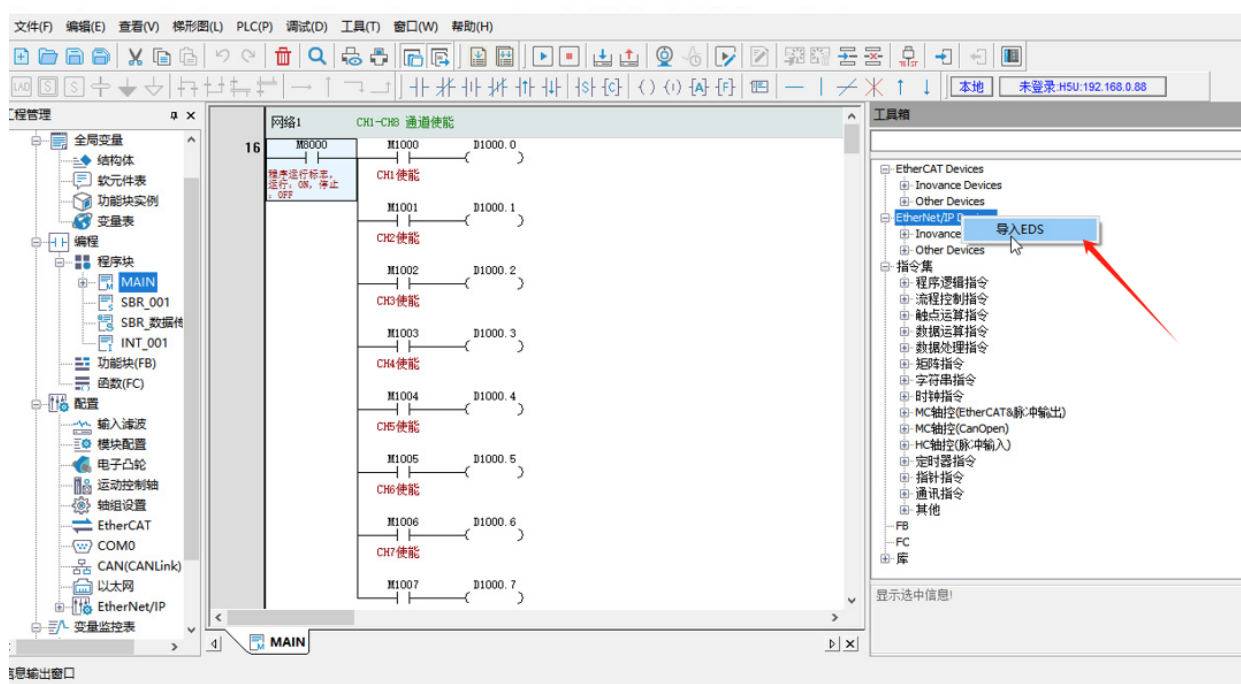
软件：AutoShop

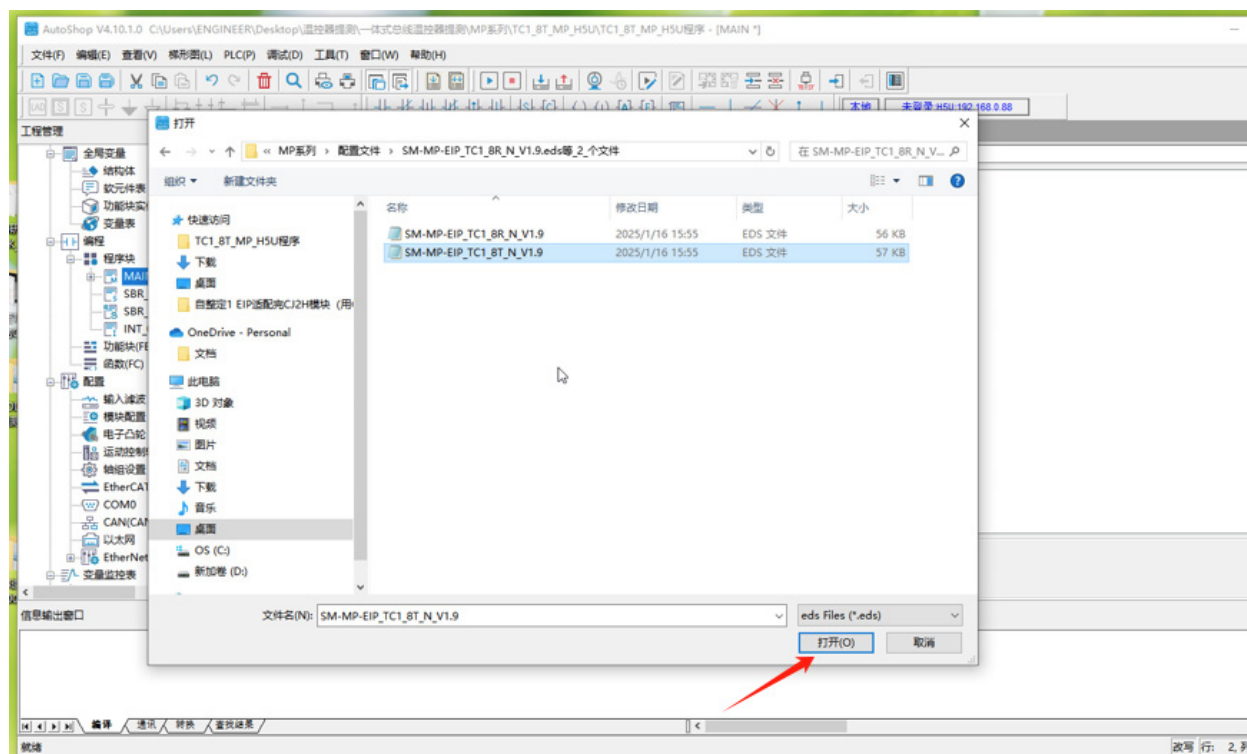
PLC：H5U-A8

模块：TC1-8T-MP

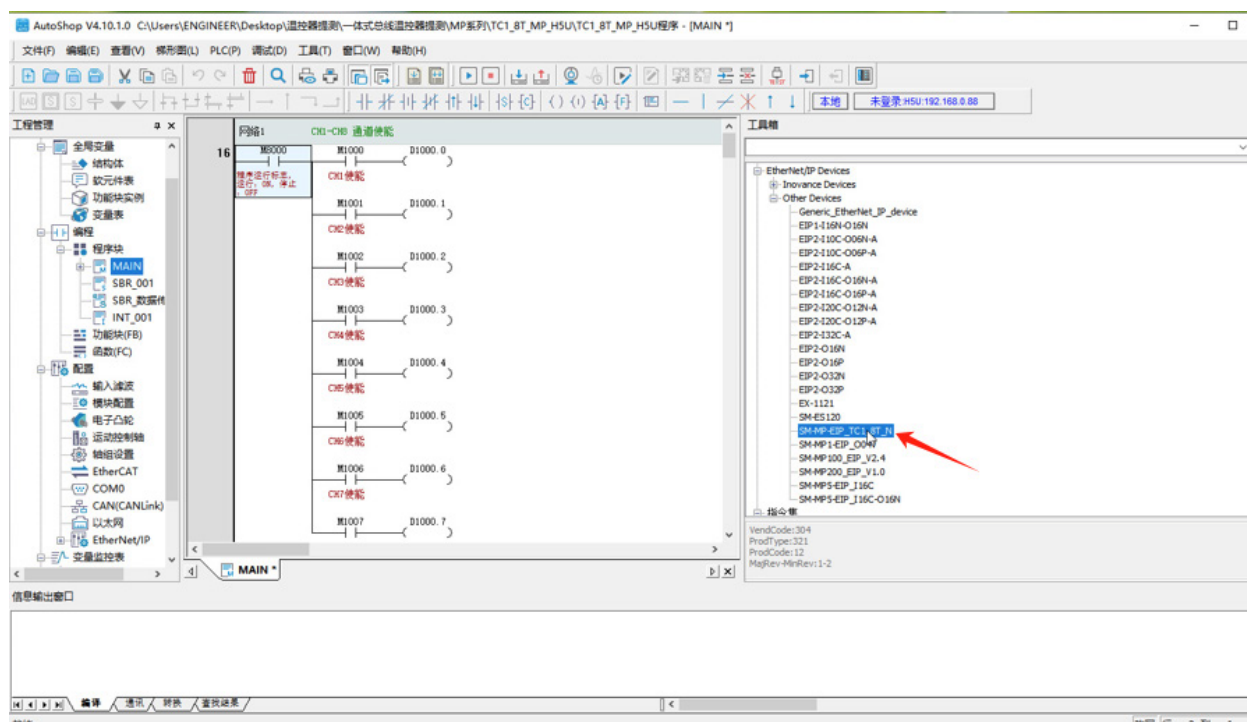
备注：TC1-8R-MP模块也可以参考此案例

1)将TC1-8T-MP模块拨码拨到2位置，连接好PLC和模块后，导入模块的配置文件

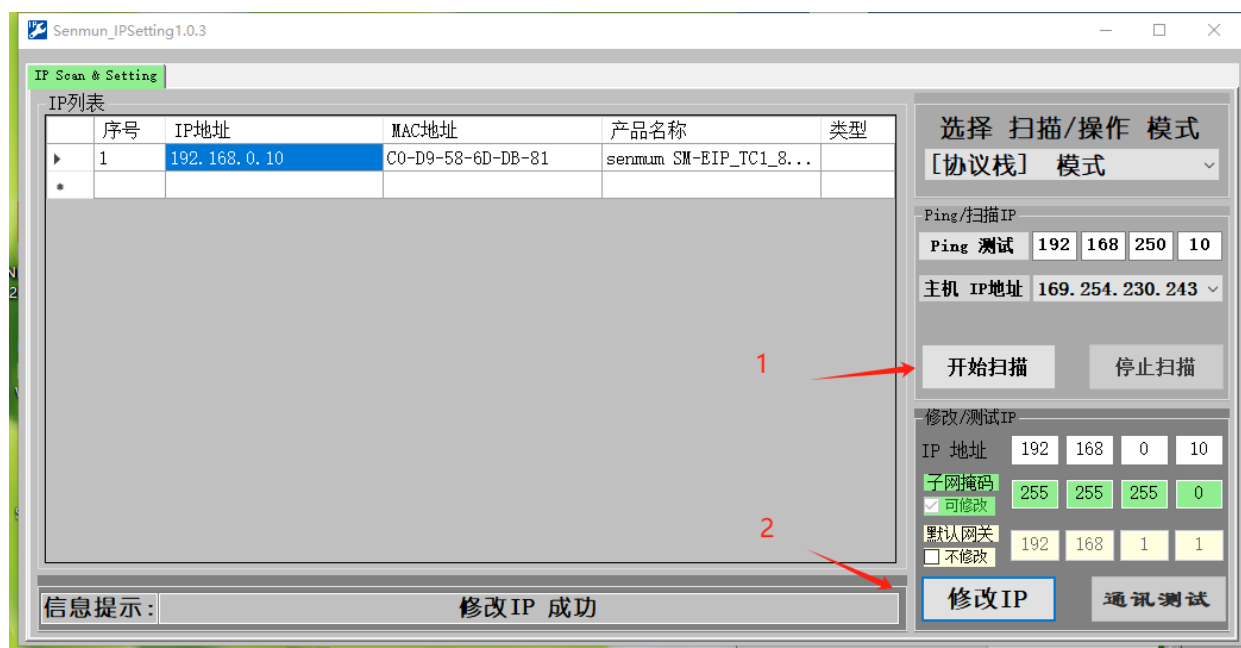




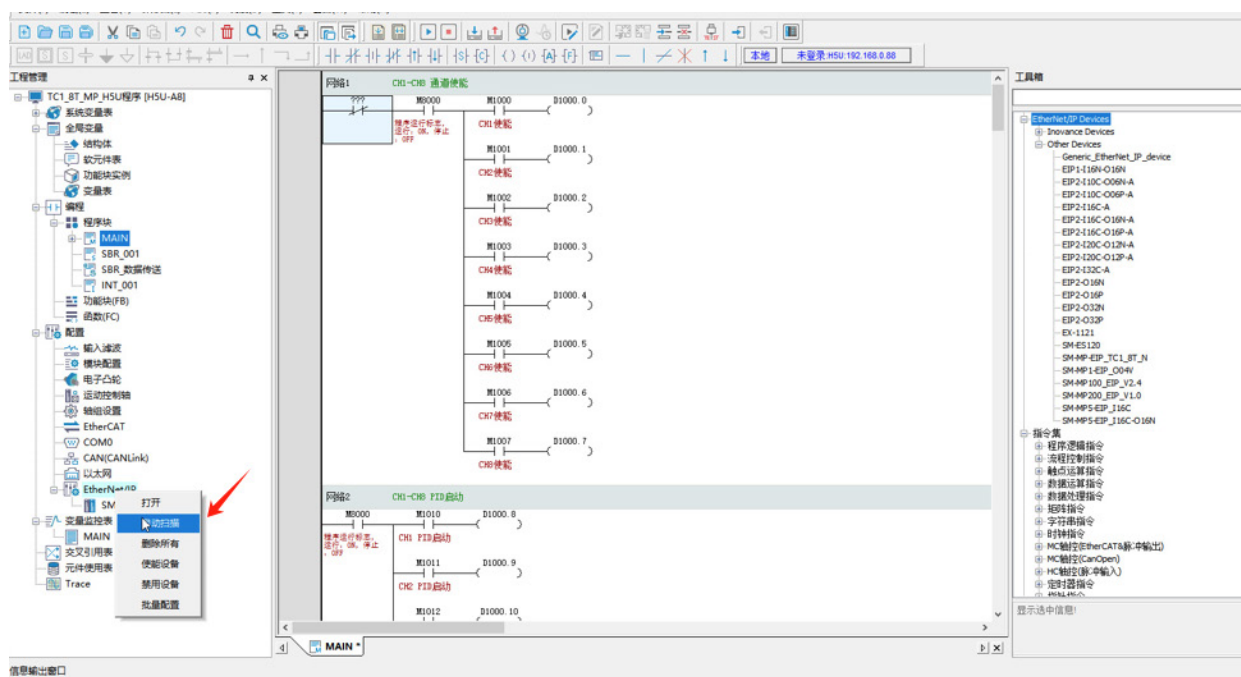
5 使用案例

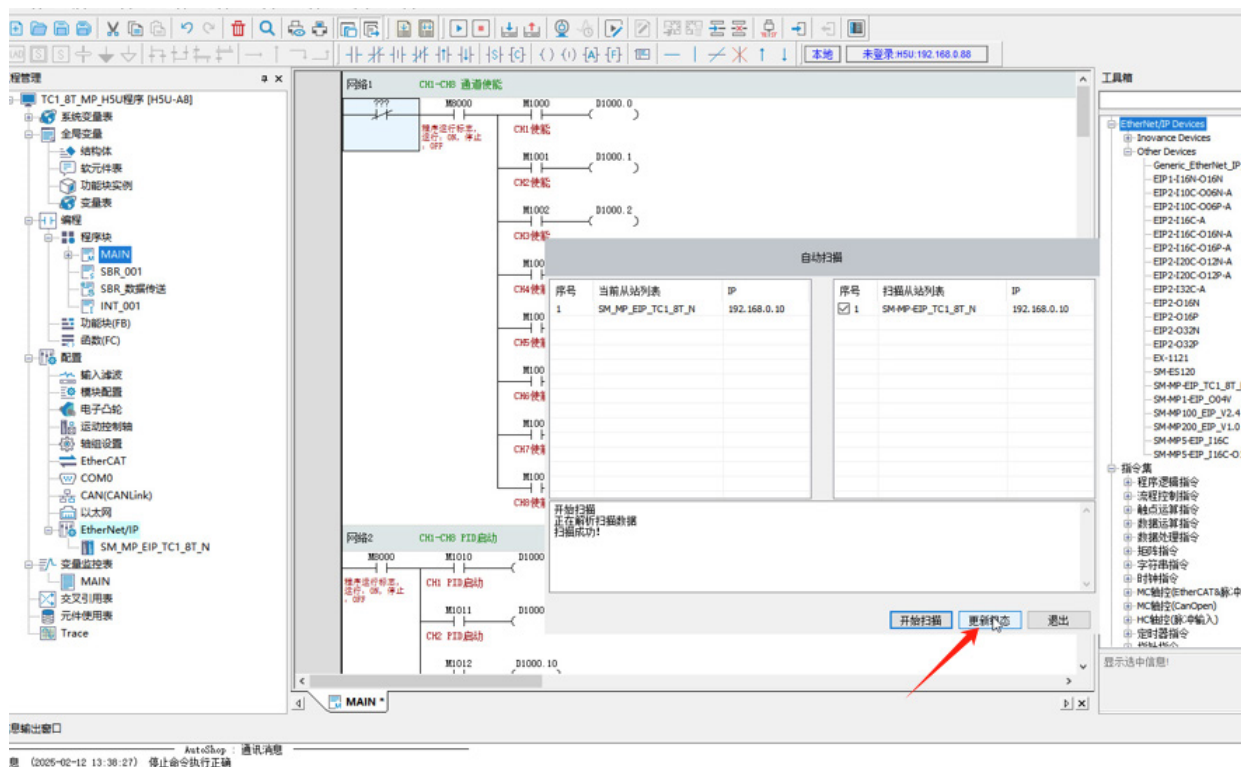
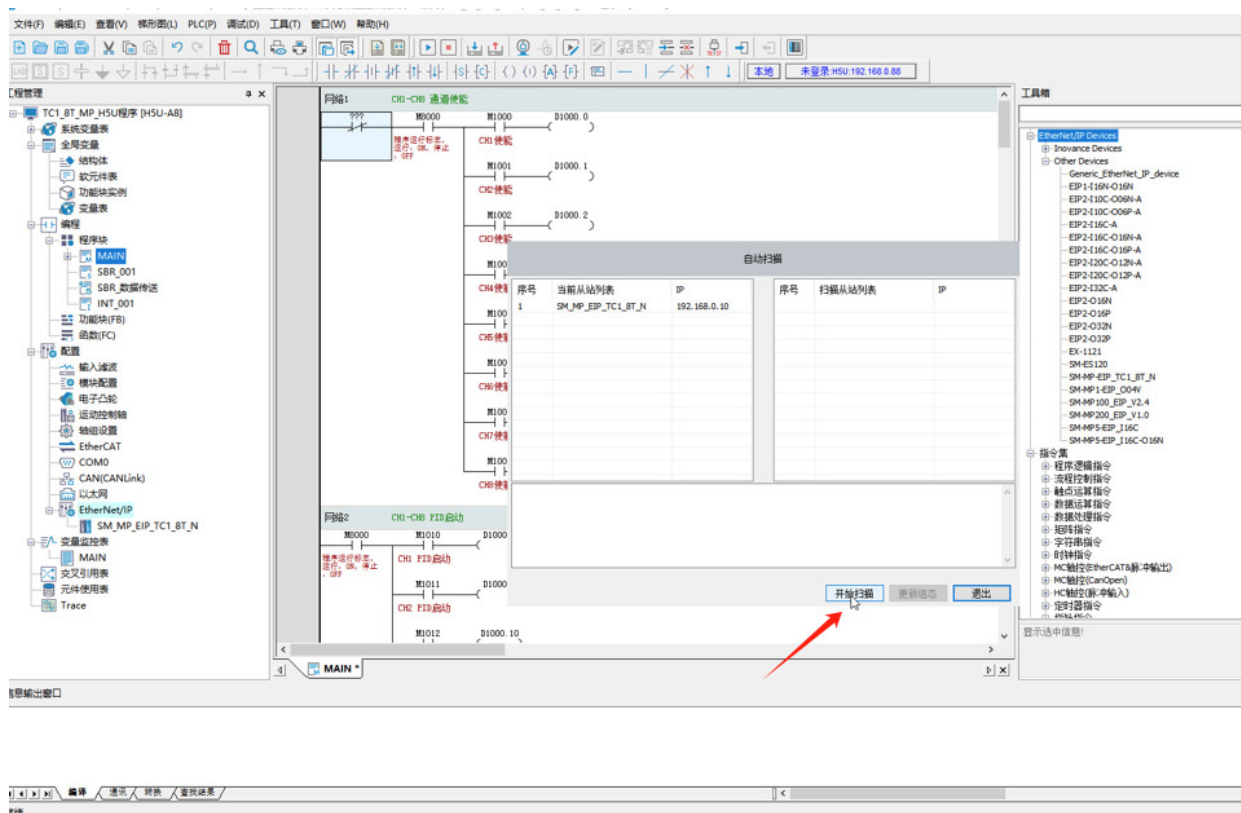


2) 用三铭公司的IP修改软件，修改模块的IP地和PLC的网段地址一致

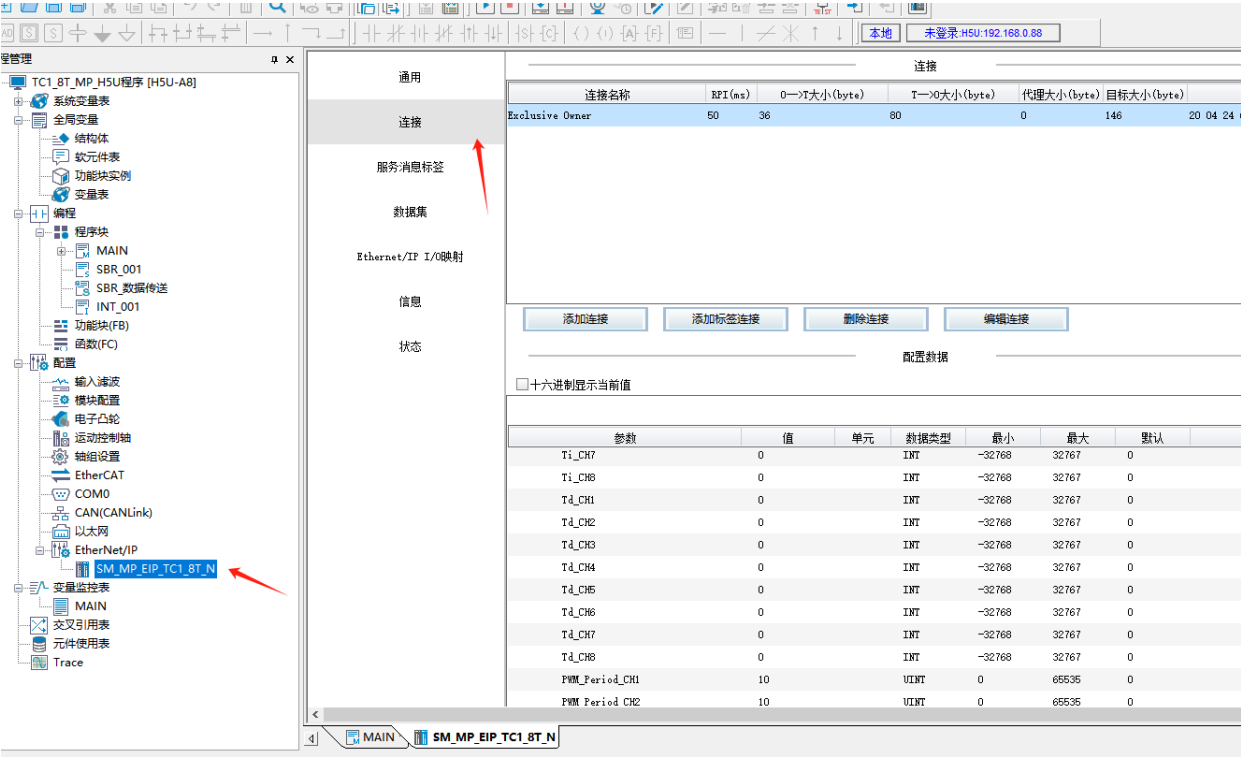


3) 右键项目树如下位置，自动扫描，组态模块





4) 双击扫描得到的模块，点击连接，设置模块的参数

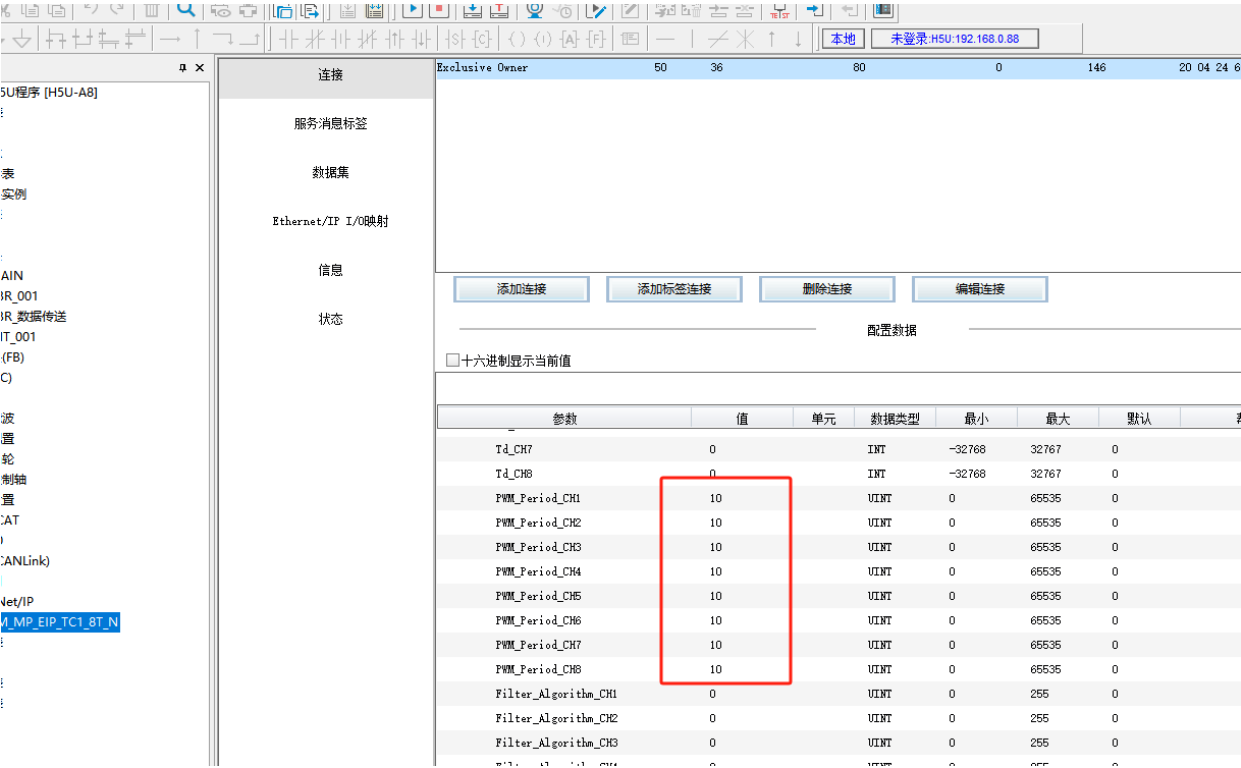


注意：Sensor Type和PWM period是**必须设定的**
推荐：PWM period设置为10，Filter size设置为5，Filter Amplitude设置为20，其他可以不用设置

Kp, TI, Td分别是比例，积分，微分。这三个值需要在模块自整定得到各个通道的PID三个值后手动填入（自整定方法在后面步骤说明）

如下所示

设置如下所示：



连接

服务消息标签

数据集

Ethernet/IP I/O映射

信息

状态

连接名称	RPI (ms)	0→1大小(byte)	1→0大小(byte)	代理大小(byte)	目标大小(byte)	连接
Exclusive Owner	50	36	80	0	146	20 04 24 66

添加连接

添加标签连接

删除连接

编辑连接

配置数据

☐十六进制显示当前值

参数	值	单元	数据类型	最小	最大	默认	帮助
Filter_Algorithm_CH7	0		UINT	0	255	0	
Filter_Algorithm_CH8	0		UINT	0	255	0	
Filter_Size_CH1	5		UINT	0	65535	0	
Filter_Size_CH2	5		UINT	0	65535	0	
Filter_Size_CH3	5		UINT	0	65535	0	
Filter_Size_CH4	5		UINT	0	65535	0	
Filter_Size_CH5	5		UINT	0	65535	0	
Filter_Size_CH6	5		UINT	0	65535	0	
Filter_Size_CH7	5		UINT	0	65535	0	
Filter_Size_CH8	5		UINT	0	65535	0	
Filter_Amplitude_CH1	20		UINT	0	65535	0	
Filter_Amplitude_CH2	20		UINT	0	65535	0	
Filter_Amplitude_CH3	20		UINT	0	65535	0	
Filter_Amplitude_CH4	20		UINT	0	65535	0	

O映射

添加连接

添加标签连接

删除连接

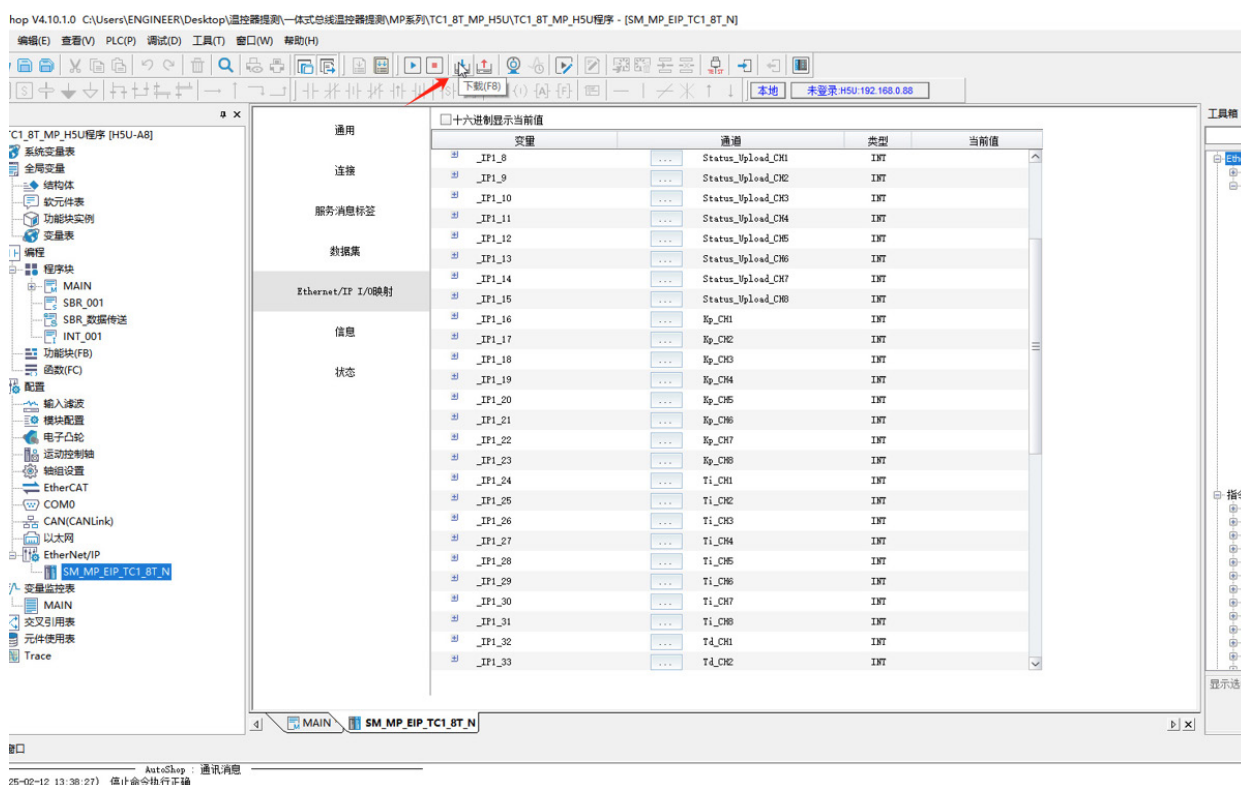
编辑连接

配置数据

☐十六进制显示当前值

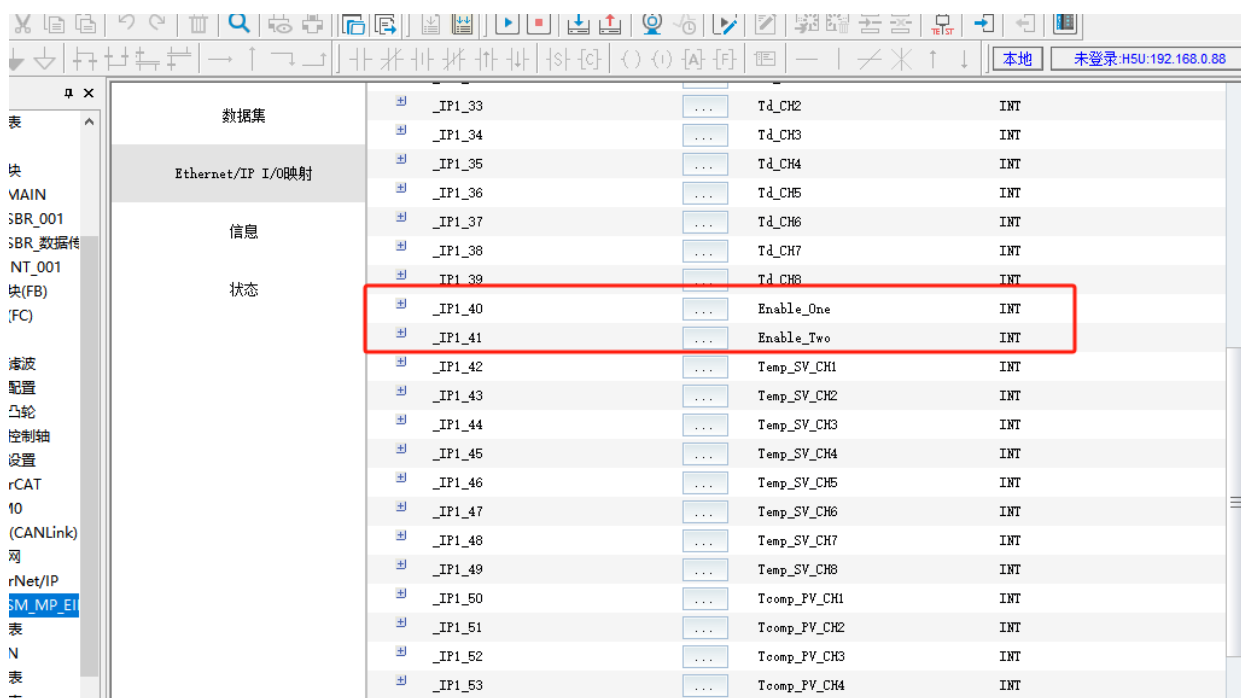
参数	值	单元	数据类型	最小	最大	默认	帮助
Filter_Size_CH5	5		UINT	0	65535	0	
Filter_Size_CH6	5		UINT	0	65535	0	
Filter_Size_CH7	5		UINT	0	65535	0	
Filter_Size_CH8	5		UINT	0	65535	0	
Filter_Amplitude_CH1	20		UINT	0	65535	0	
Filter_Amplitude_CH2	20		UINT	0	65535	0	
Filter_Amplitude_CH3	20		UINT	0	65535	0	
Filter_Amplitude_CH4	20		UINT	0	65535	0	
Filter_Amplitude_CH5	20		UINT	0	65535	0	
Filter_Amplitude_CH6	20		UINT	0	65535	0	
Filter_Amplitude_CH7	20		UINT	0	65535	0	
Filter_Amplitude_CH8	0		UINT	0	65535	0	
Hold_CH1	0		UINT	0	255	0	
Hold_CH2	0		UINT	0	255	0	

5) 下载程序进PLC，切换到监控模式



6) 使能，PID启动，自整定启动三个BOOL参数在变量地址中的排序说明，本案例是_IP1_40，_IP1_41两个字

注：状态反馈里面的地址类似



_IP1_40

CH8 PID 启动	CH7 PID 启动	CH6 PID 启动	CH5 PID 启动	CH4 PID 启动	CH3 PID 启动	CH2 PID 启动	CH1 PID 启动	CH 8 使能	CH 7 使能	CH 6 使能	CH 5 使能	CH 4 使能	CH 3 使能	CH 2 使能	CH 1 使能
------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------	------------	------------	------------	------------	------------	------------	------------

_IP1_40.0

_IP1_41

								CH 8 白整定 启动	CH 7 白整定 启动	CH 6 白整定 启动	CH 5 白整定 启动	CH 4 白整定 启动	CH 3 白整定 启动	CH 2 白整定 启动	CH 1 白整定 启动
--	--	--	--	--	--	--	--	-------------------	-------------------	-------------------	-------------------	-------------------	-------------------	-------------------	-------------------

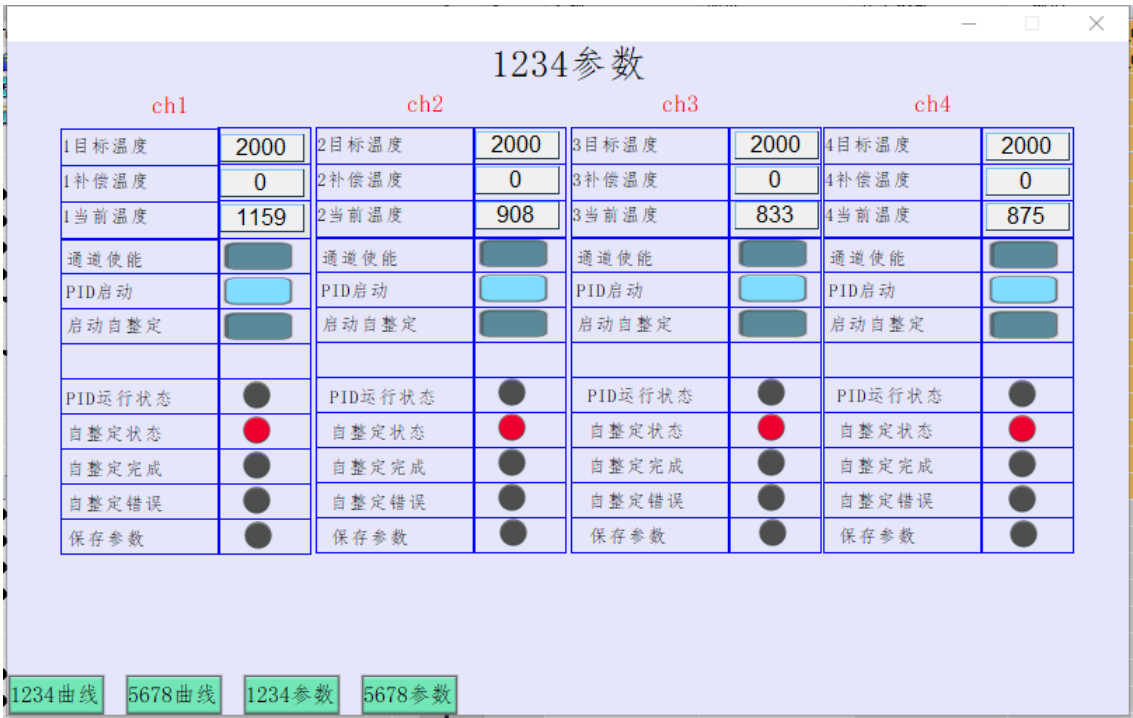
_IP1_41.0

7) 启动自整定获取比例，积分，微分参数值

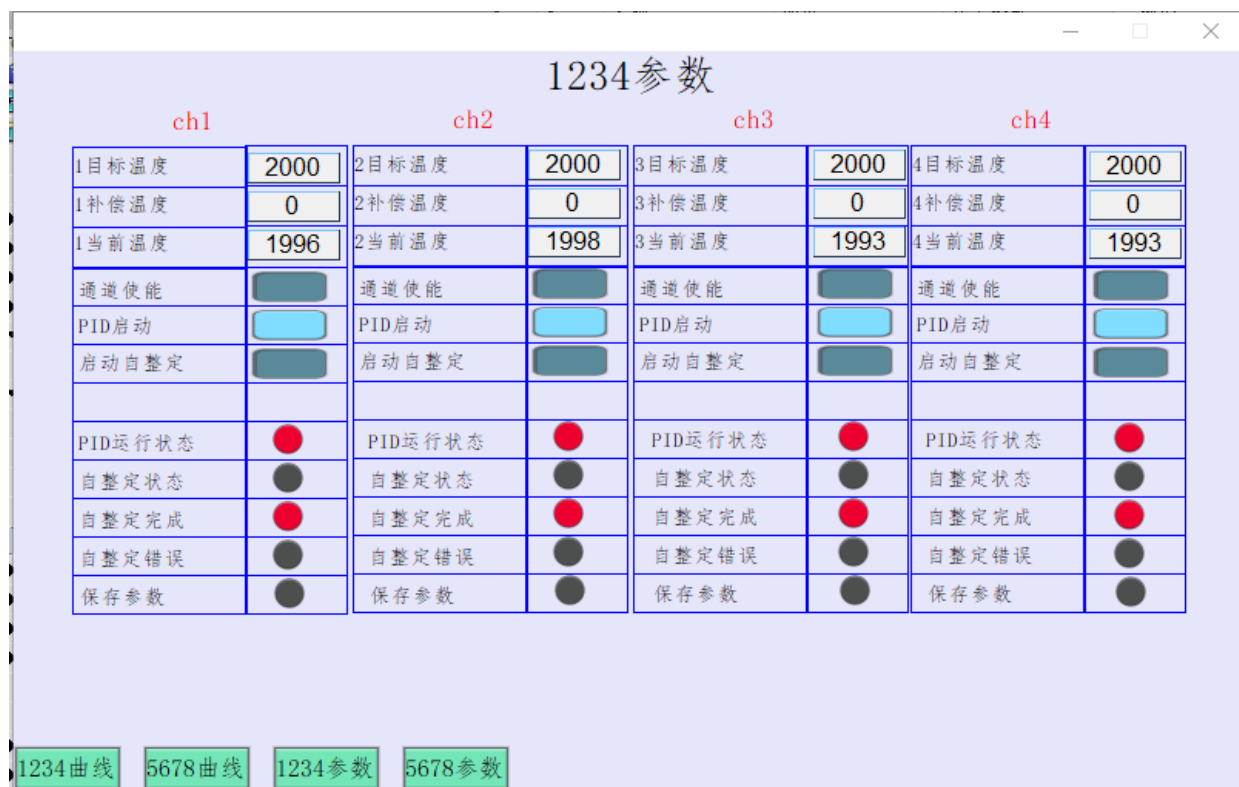
步骤1，设置目标温度后，使能相关通道（BOOL），启动相关通道自整定（BOOL），此时状态反馈的自整定运行中会接通（BOOL），此时外部设备进入加热阶段，再自动运行一段时间后，自整定完成，自整定完成状态会接通（BOOL），完成后在上行数据的PID参数的IW地址会出现得到的结果。完成需要通道的自整定操作后，将得到的PID参数值填入模块组态界面的相关属性值（以便下次启动，直接运行启动控温即可，不再需要经过自整定调节）

后续启动加温，直接启动PID运行（BOOL）即可。

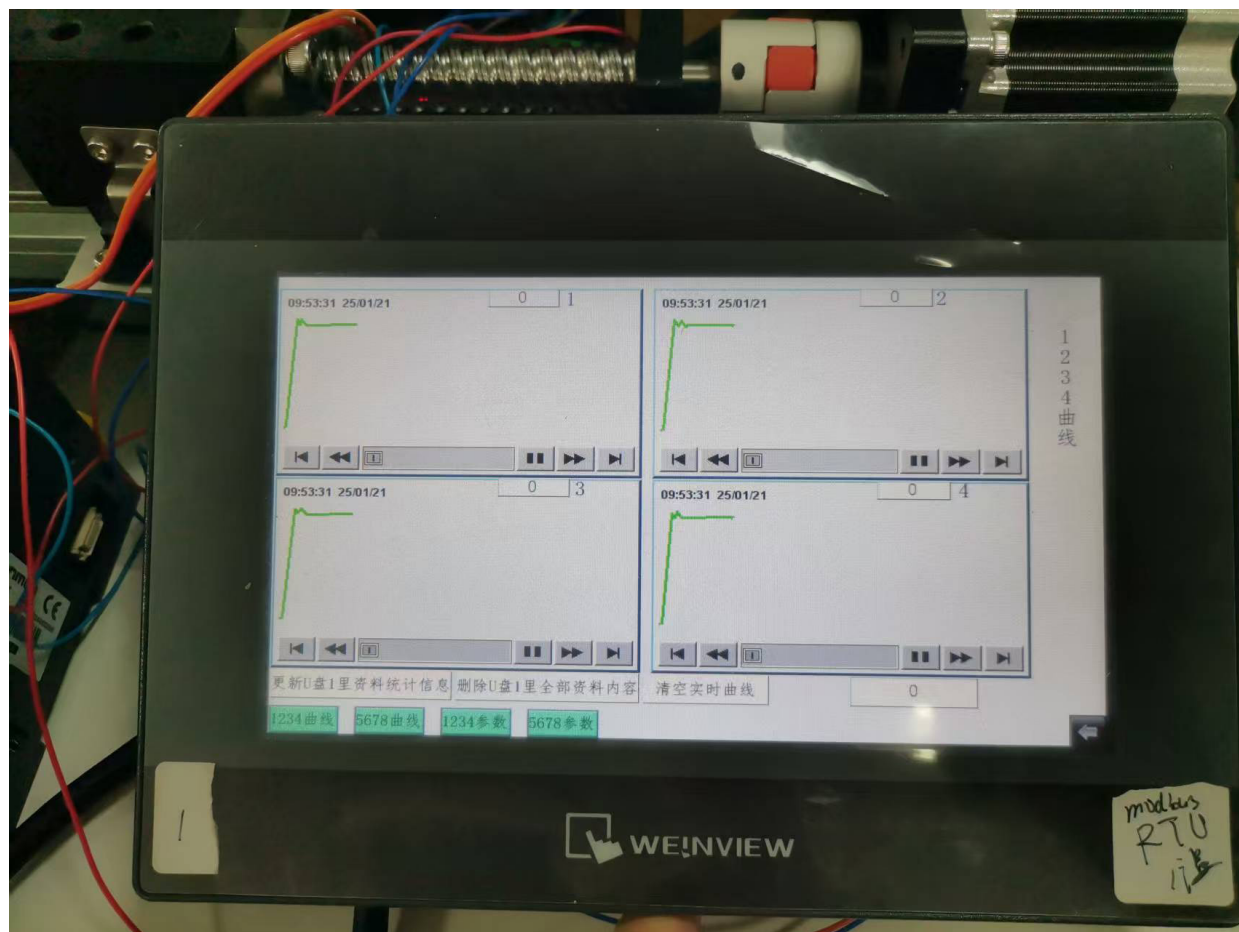
设定目标温度，使能通道，启动自整定后，自整定状态会接通



8)自整定完成后，相关反馈通道接通



9)温度曲线的显示



10) 得到的PID值

通用

连接

消息标签

数据集

et/IP I/O映射

信息

状态

☐ 十六进制显示当前值				
变量	通道	类型	当前值	
☐ _IP1_16	☐ Kp_CM1	INT	181	
☐ _IP1_17	☐ Kp_CM2	INT	94	
☐ _IP1_18	☐ Kp_CM3	INT	218	
☐ _IP1_19	☐ Kp_CM4	INT	162	
☐ _IP1_20	☐ Kp_CM5	INT	177	
☐ _IP1_21	☐ Kp_CM6	INT	127	
☐ _IP1_22	☐ Kp_CM7	INT	224	
☐ _IP1_23	☐ Kp_CM8	INT	195	
☐ _IP1_24	☐ Ti_CM1	INT	29	
☐ _IP1_25	☐ Ti_CM2	INT	22	
☐ _IP1_26	☐ Ti_CM3	INT	28	
☐ _IP1_27	☐ Ti_CM4	INT	29	
☐ _IP1_28	☐ Ti_CM5	INT	30	
☐ _IP1_29	☐ Ti_CM6	INT	23	
☐ _IP1_30	☐ Ti_CM7	INT	34	
☐ _IP1_31	☐ Ti_CM8	INT	36	
☐ _IP1_32	☐ Td_CM1	INT	14	
☐ _IP1_33	☐ Td_CM2	INT	11	
☐ _IP1_34	☐ Td_CM3	INT	14	
☐ _IP1_35	☐ Td_CM4	INT	14	
☐ _IP1_36	☐ Td_CM5	INT	15	
☐ _IP1_37	☐ Td_CM6	INT	11	
☐ _IP1_38	☐ Td_CM7	INT	17	
☐ _IP1_39	☐ Td_CM8	INT	18	
☐ _IP1_40	☐ Enable_One	INT	15	
☐ _IP1_41	☐ Enable_Two	INT	16	

11)填入连接部分的属性值后，在下载程序即可

数据集

Ethernet/IP I/O映射

信息

状态

添加连接

添加标签连接

删除连接

编辑连接

☐十六进制显示当前值

参数	值	单元	数据类型	最小	最大	默认
Sensor_Type_CH8	0		UINT	0	255	0
Kp_CM1	181		INT	-32768	32767	0
Kp_CM2	94		INT	-32768	32767	0
Kp_CM3	218		INT	-32768	32767	0
Kp_CM4	162		INT	-32768	32767	0
Kp_CM5	0		INT	-32768	32767	0
Kp_CM6	0		INT	-32768	32767	0
Kp_CM7	0		INT	-32768	32767	0
Kp_CM8	0		INT	-32768	32767	0
Ti_CM1	29		INT	-32768	32767	0
Ti_CM2	22		INT	-32768	32767	0
Ti_CM3	28		INT	-32768	32767	0
Ti_CM4	29		INT	-32768	32767	0
Ti_CM5	0		INT	-32768	32767	0

基恩士KV7500使用案例

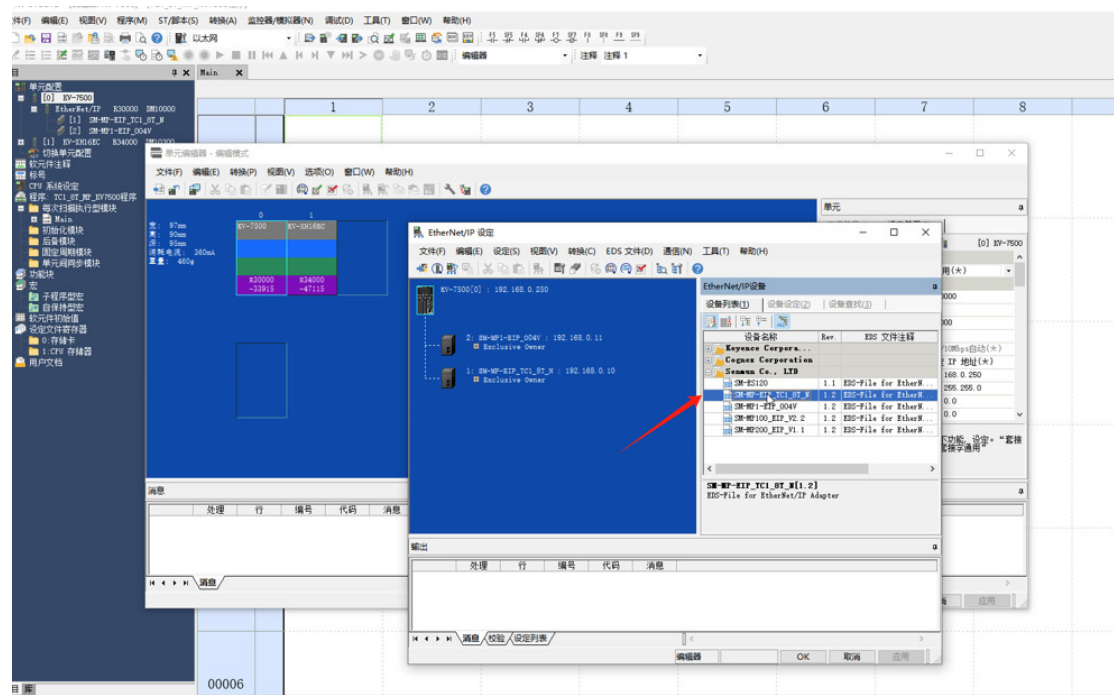
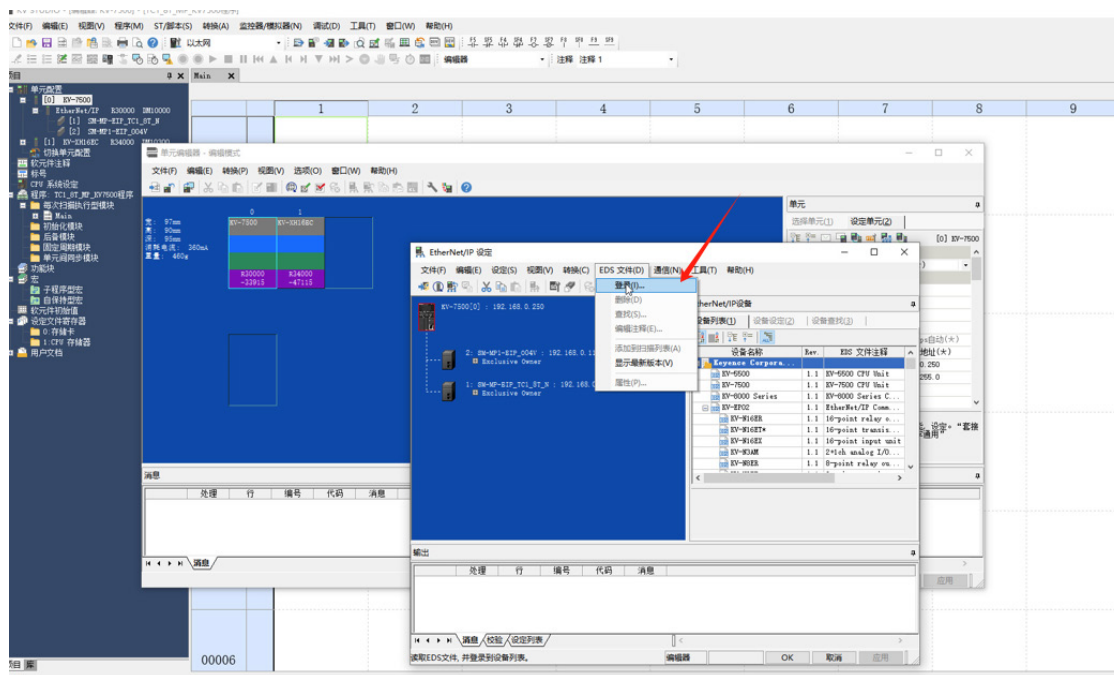
软件: KV STUDIO

PLC: KV7500

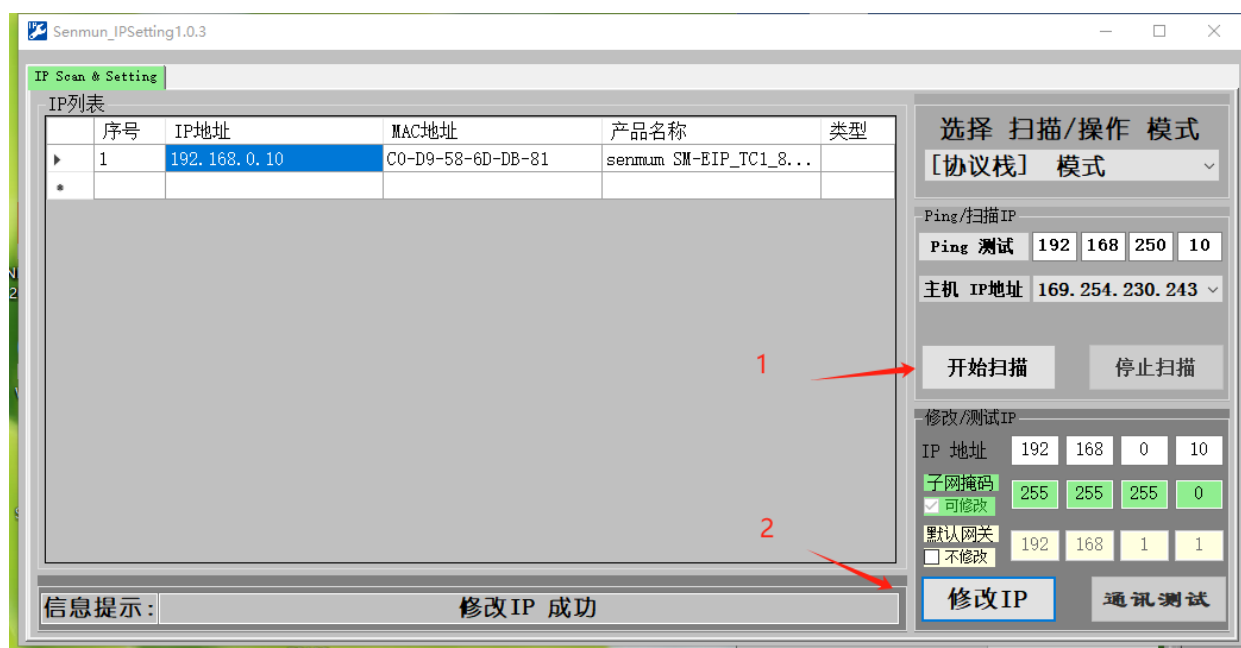
模块: TC1-8T-MP

备注: TC1-8R-MP模块也可以参考此案例

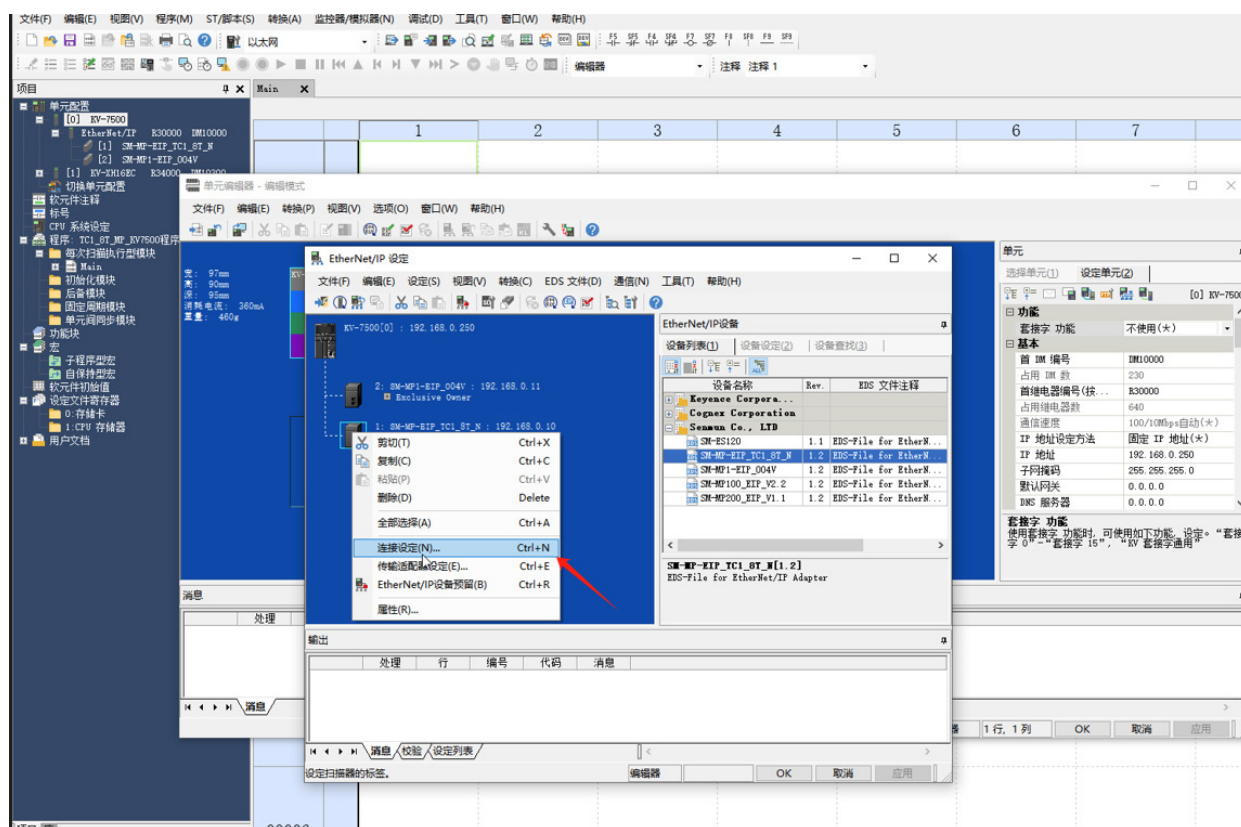
1) 将TC1-8T-MP模块拨码拨到2位置, 连接好PLC和模块后, 打开单元编辑器, 导入模块的配置文件



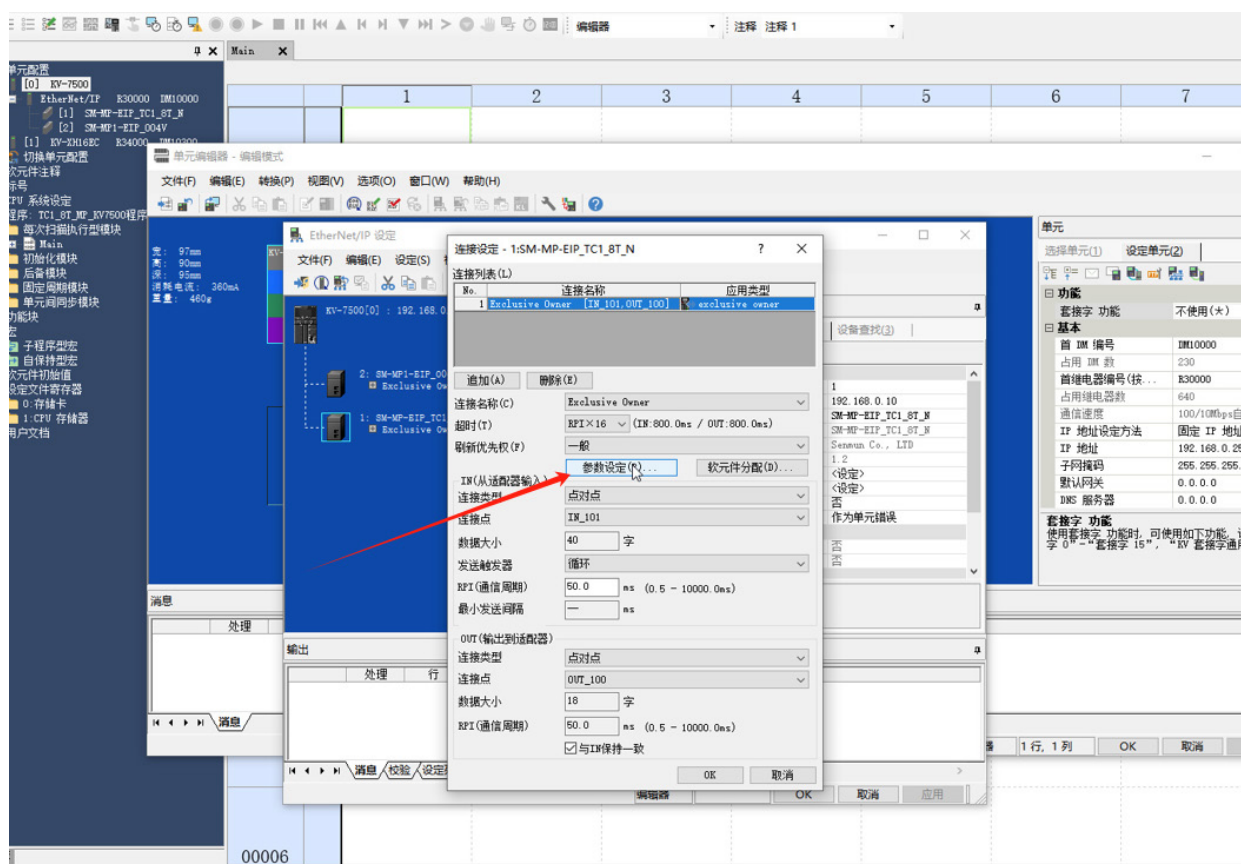
2) 用三铭公司的IP修改软件, 修改模块的IP地和PLC的网段地址一致



3) 组态模块, 将模块拖到左侧设备数下, 右键模块, 点击连接设定



4) 点击参数设定



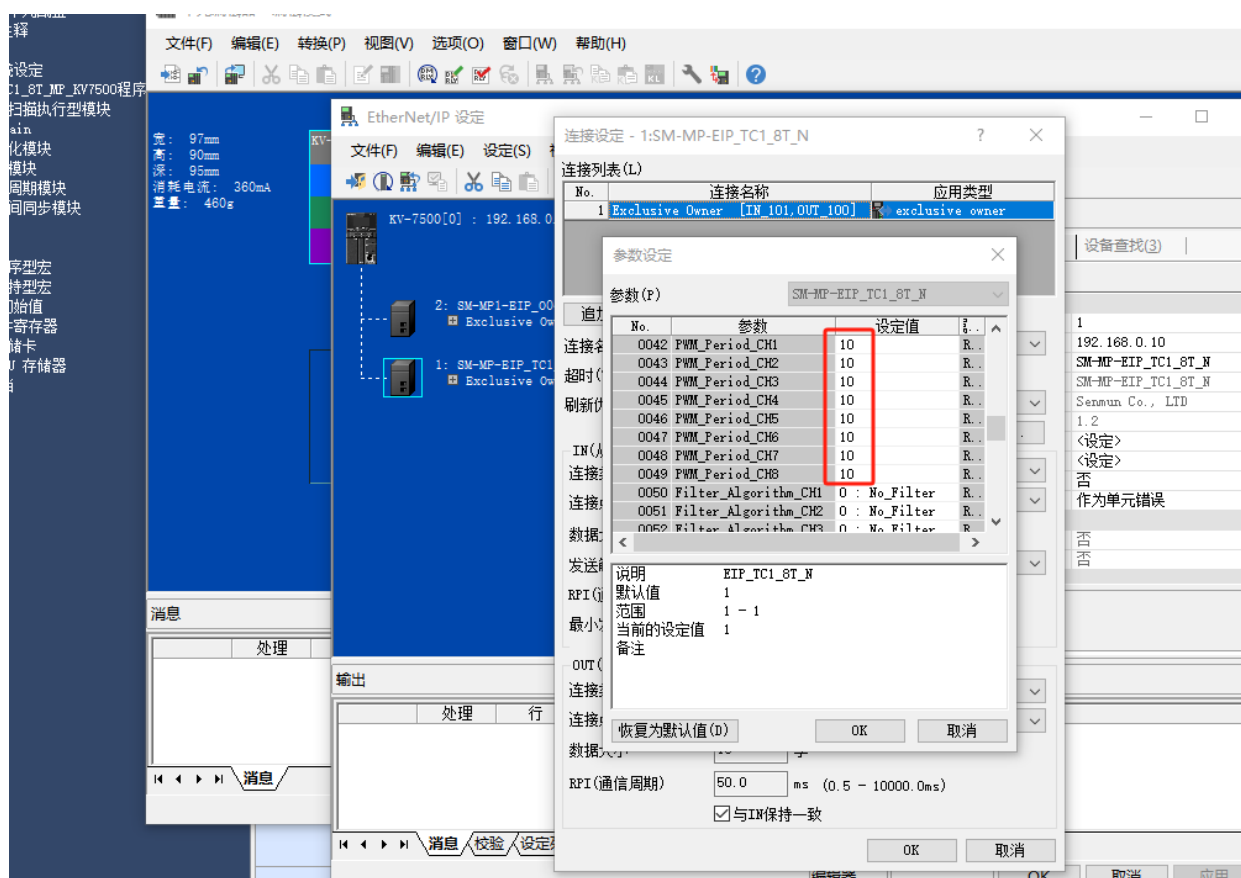
5) 注意：Sensor Type和PWM period是**必须设定的**

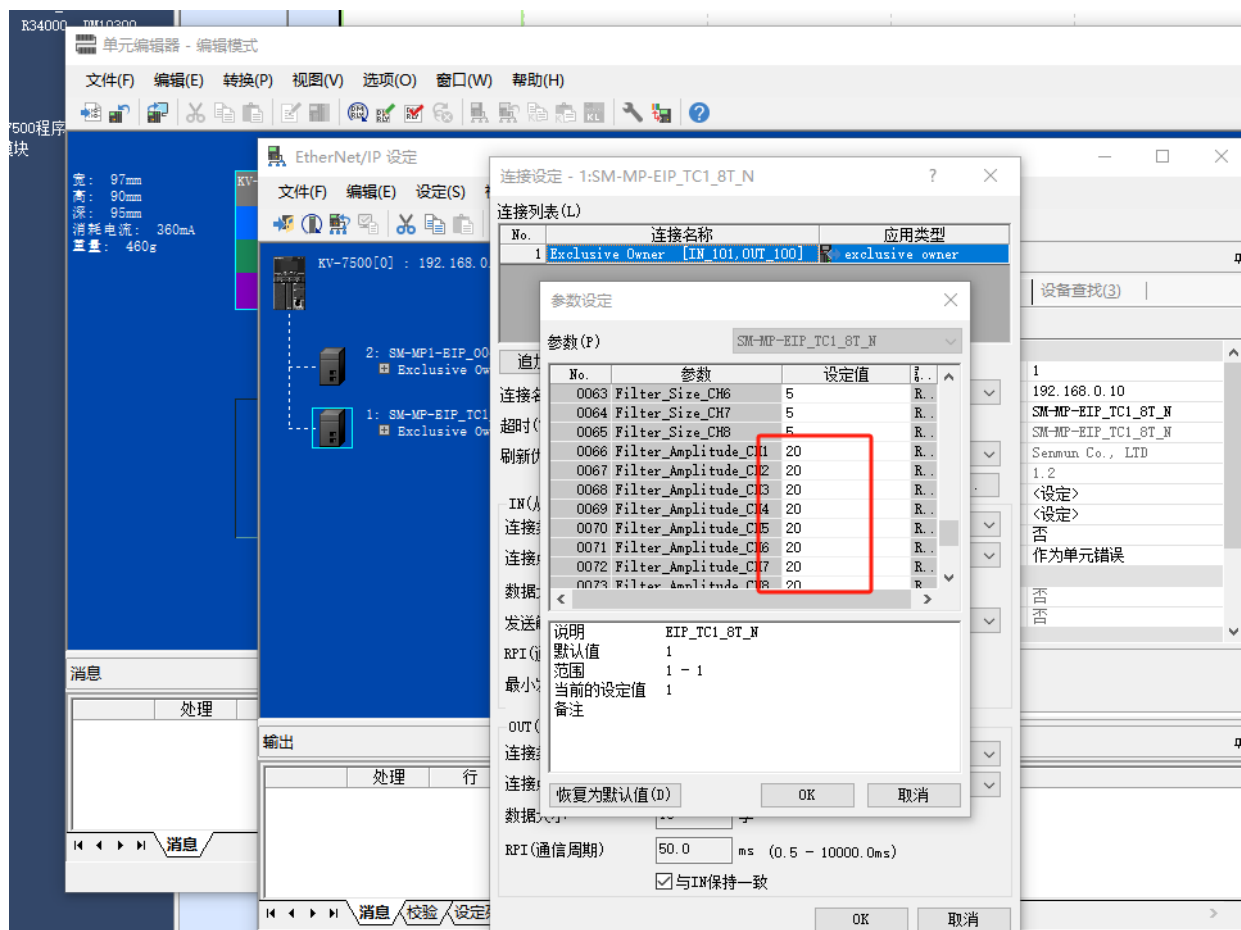
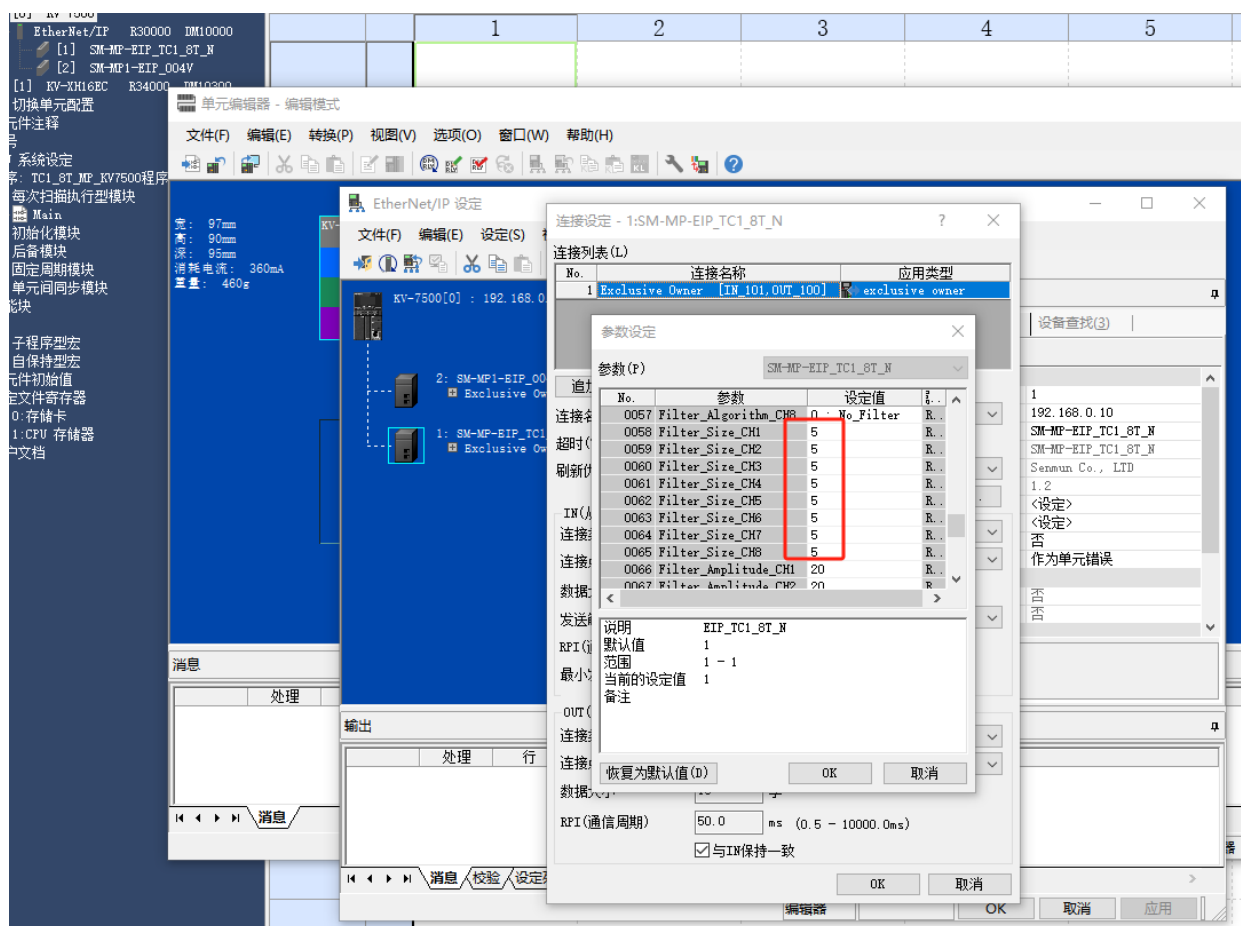
推荐：PWM period设置为10，Filter size设置为5，Filter Amplitude设置为20，其他可以不用设置

Kp, Ti, Td分别是比例，积分，微分。这三个值需要在模块自整定得到各个通道的PID三个值后手动填入（自整定方法在后面步骤说明）

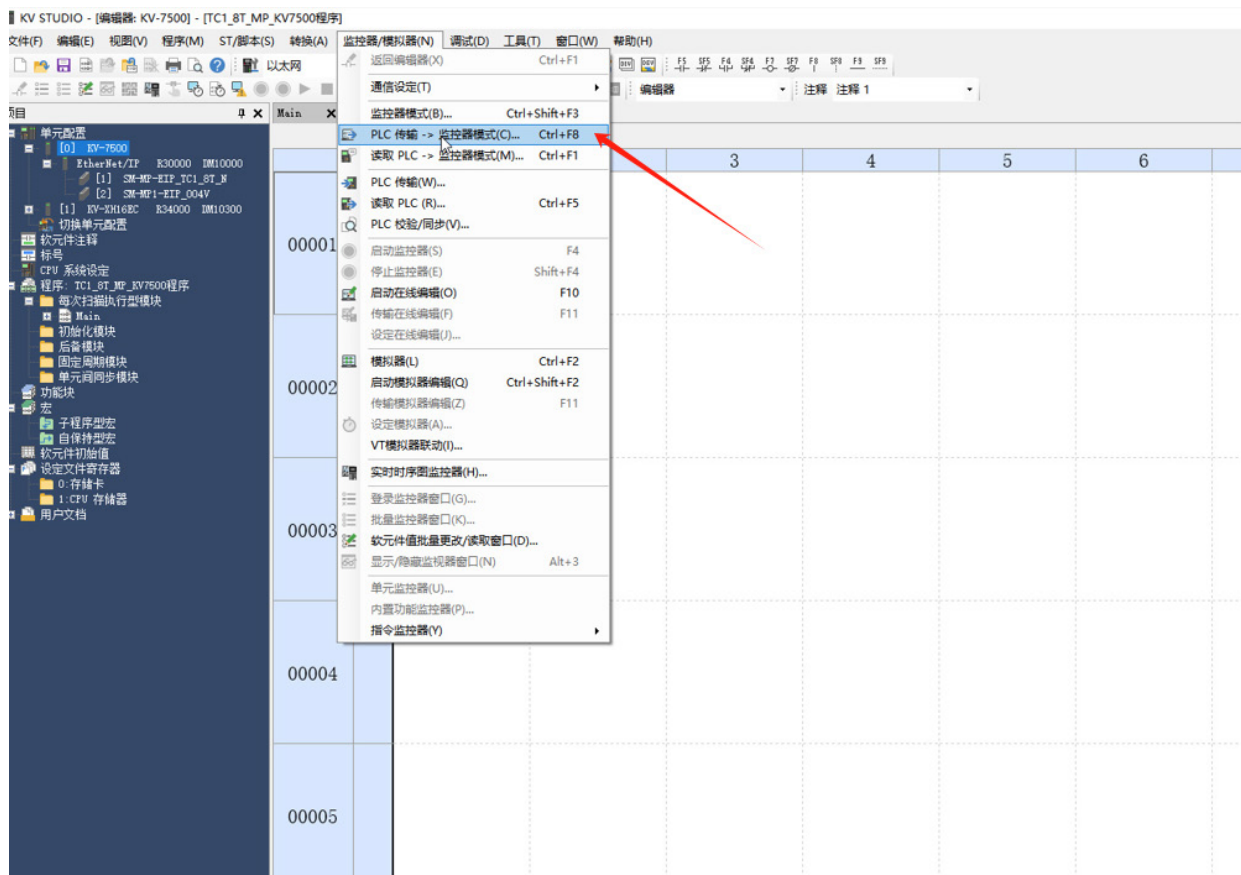
如下所示

设置如下所示：





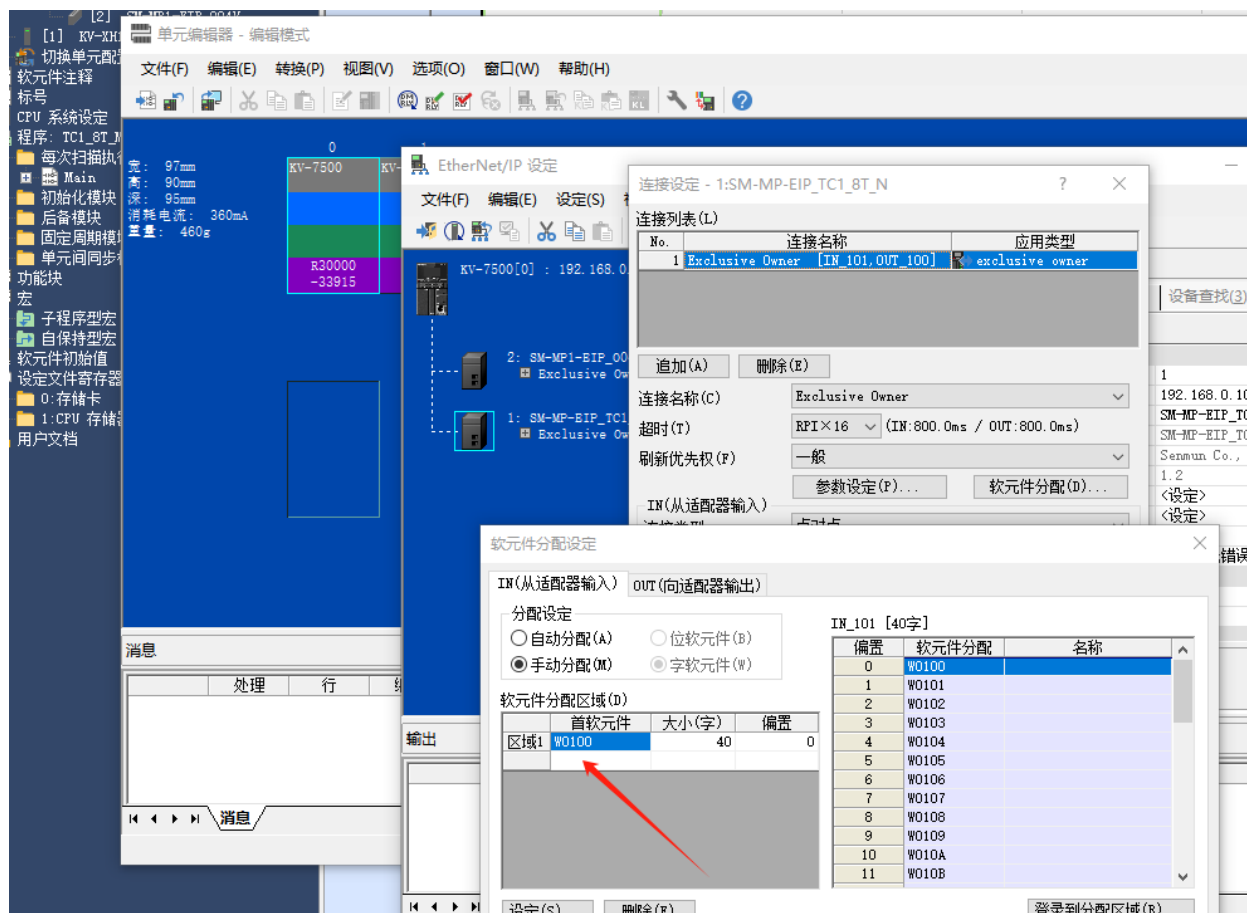
6) 下载程序进PLC，切换到监控模式



7) 使能, PID启动, 自整定启动三个BOOL参数在变量地址中的排序说明, 本案例是的起始为W100和输出地址为W1000

使能占用的2个字为W1000,W1001

注: 状态反馈里面的地址类似



W1000

CH8_PID 启动	CH7_PID 启动	CH6_PID 启动	CH5_PID 启动	CH4_PID 启动	CH3_PID 启动	CH2_PID 启动	CH1_PID 启动	CH_8 使能	CH_7 使能	CH_6 使能	CH_5 使能	CH_4 使能	CH_3 使能	CH_2 使能	CH_1 使能
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	------------	------------	------------	------------	------------	------------	------------	------------

W1000.0

W1001

								CH_8 白整定启动	CH_7 白整定启动	CH_6 白整定启动	CH_5 白整定启动	CH_4 白整定启动	CH_3 白整定启动	CH_2 白整定启动	CH_1 白整定启动
--	--	--	--	--	--	--	--	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

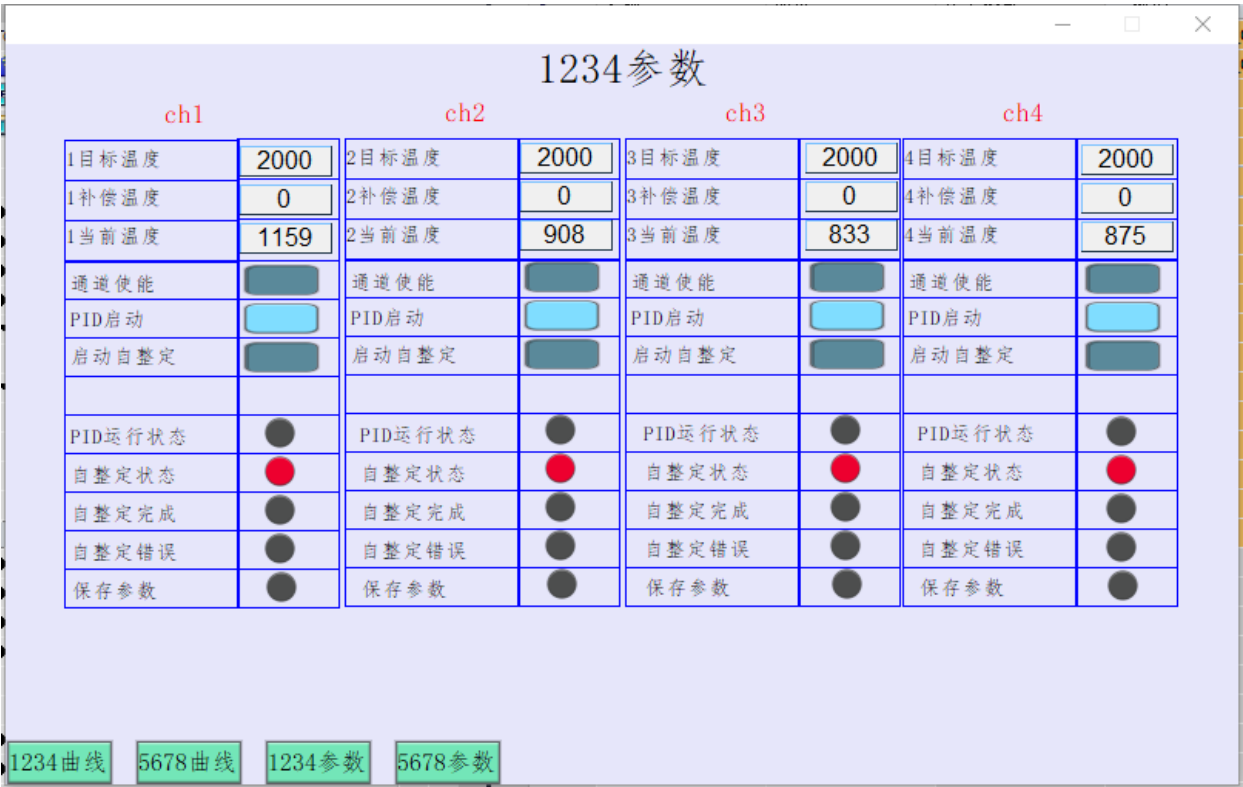
W1001.0

8) 启动自整定获取比例，积分，微分参数值

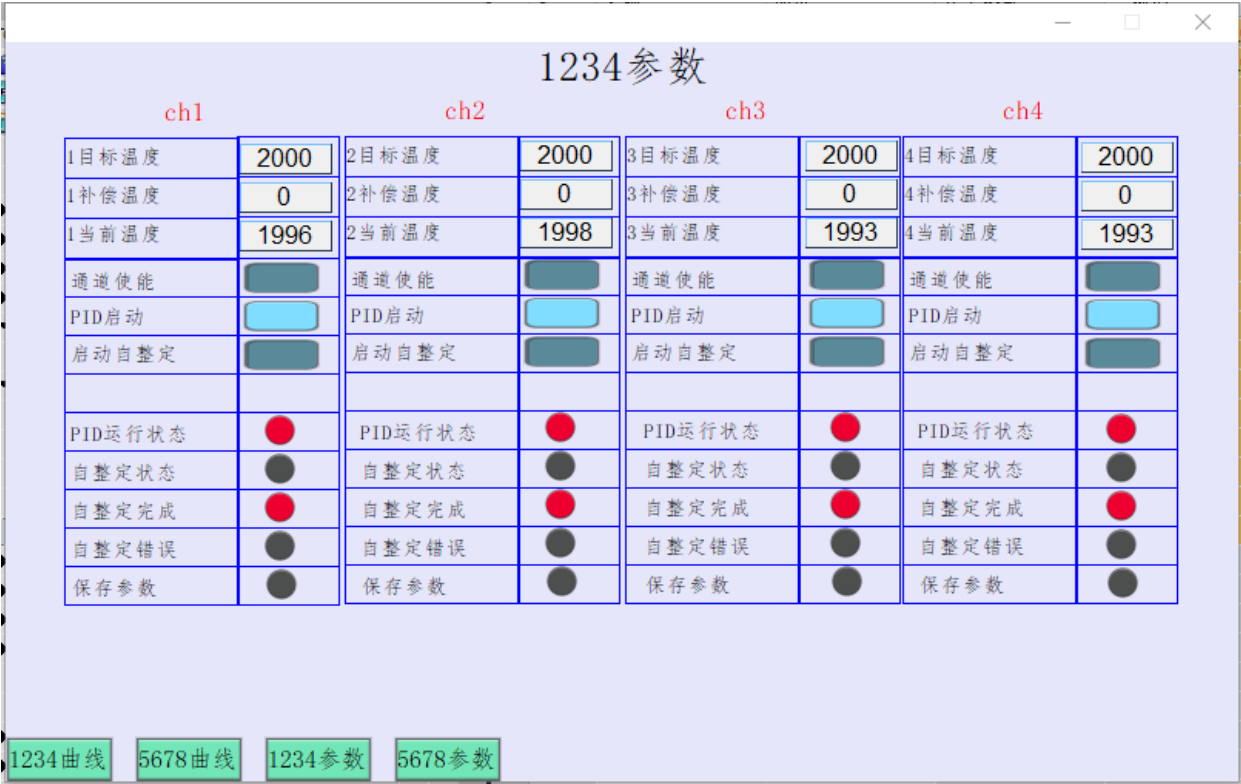
步骤1，设置目标温度后，使能相关通道（BOOL），启动相关通道自整定（BOOL），此时状态反馈的自整定运行中会接通（BOOL），此时外部设备进入加热阶段，再自动运行一段时间后，自整定完成，自整定完成状态会接通（BOOL），完成后在上行数据的PID参数的IW地址会出现得到的结果。完成需要通道的自整定操作后，将得到的PID参数值填入模块组态界面的相关属性值（以便下次启动，直接运行启动控温即可，不再需要经过自整定调节）

后续启动加温，直接启动PID运行（BOOL）即可。

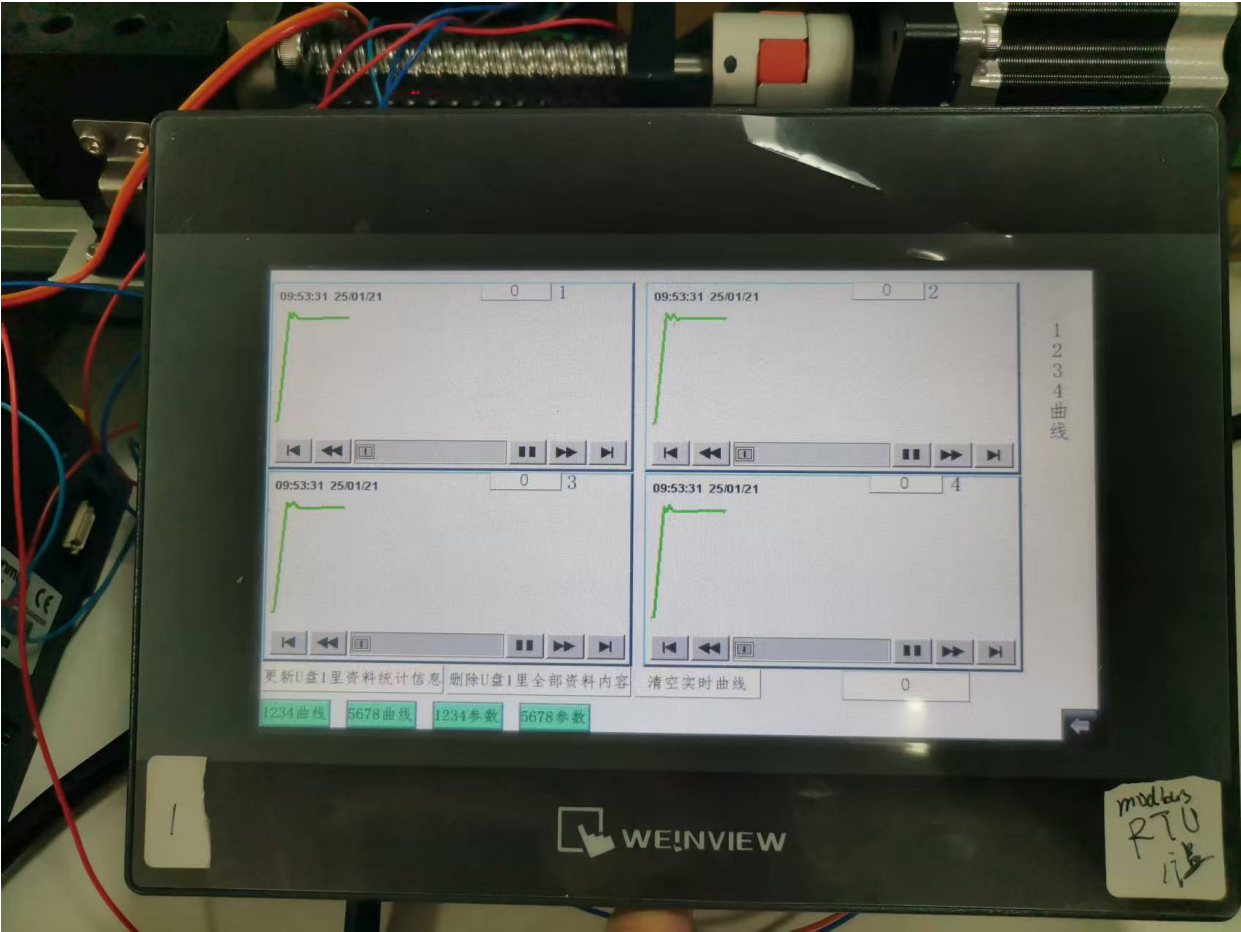
设定目标温度，使能通道，启动自整定后，自整定状态会接通



9)自整定完成后，相关反馈通道接通



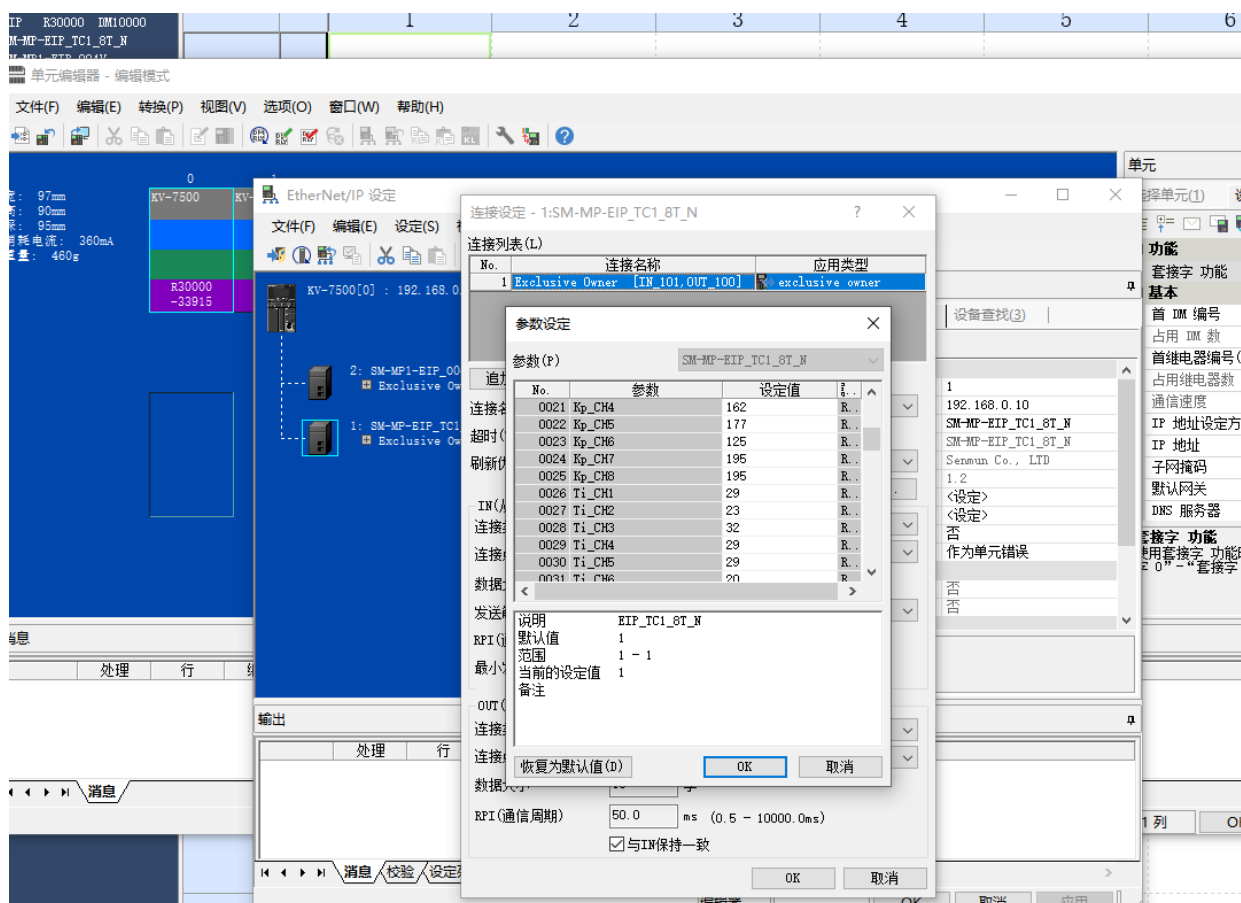
10)温度曲线的显示



11)得到的PID值

● 软元件(D) ○ 扩展单元缓冲存储器(G) ○ 扩展单元的内部软元件(V)						
程序	软元件	当前值	显示格式	设定值	接点	注释
全局	W0107	55537	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0108	5	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0109	5	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W010A	5	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W010B	5	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W010C	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W010D	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W010E	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W010F	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0110	181	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0111	134	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0112	195	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0113	162	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0114	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0115	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0116	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0117	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0118	29	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0119	23	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W011A	32	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W011B	29	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W011C	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W011D	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W011E	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W011F	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0120	14	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0121	11	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0122	16	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0123	14	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0124	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N
全局	W0125	0	16 位十进制数			KV-7500[0]. SM-MP-EIP_TC1_8T_N

12)填入连接部分的属性值后，在下载程序即可



欧姆龙NX1P使用案例

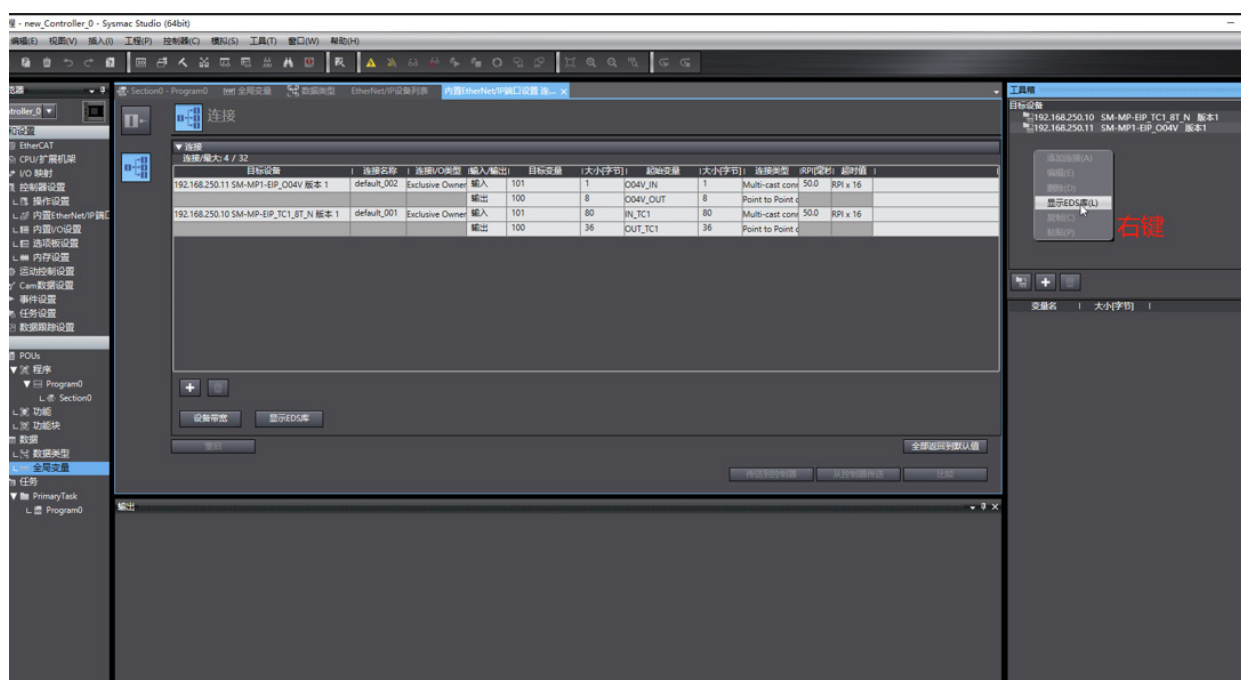
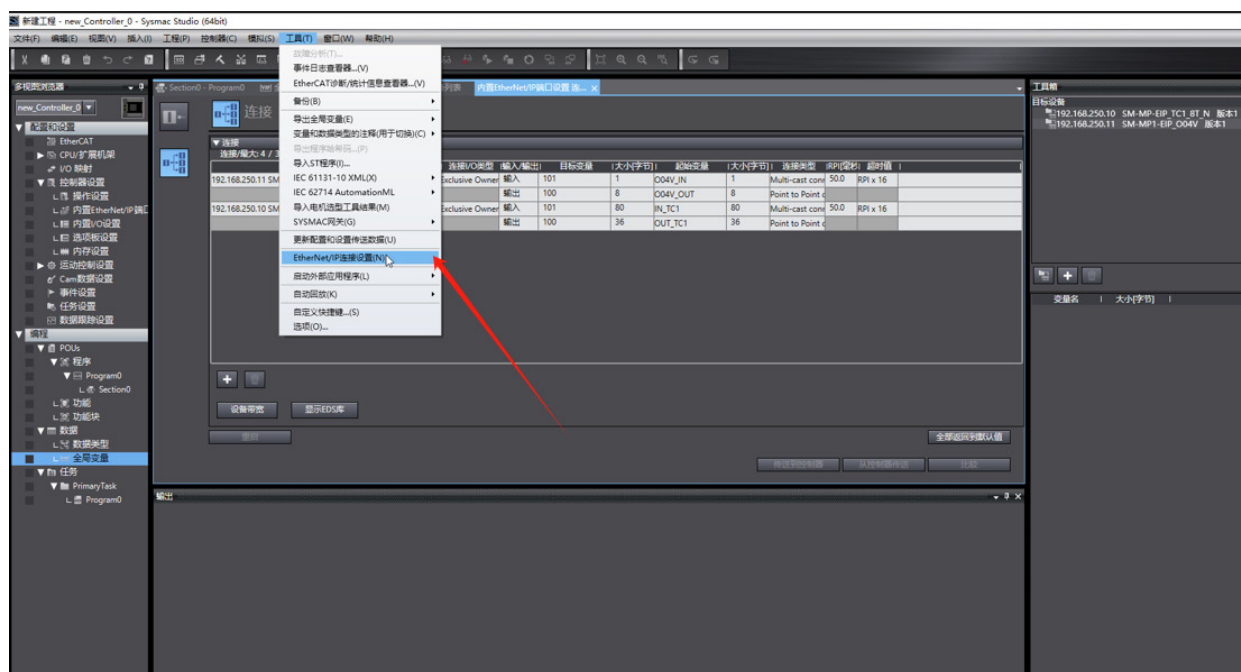
软件: Sysmac Studio

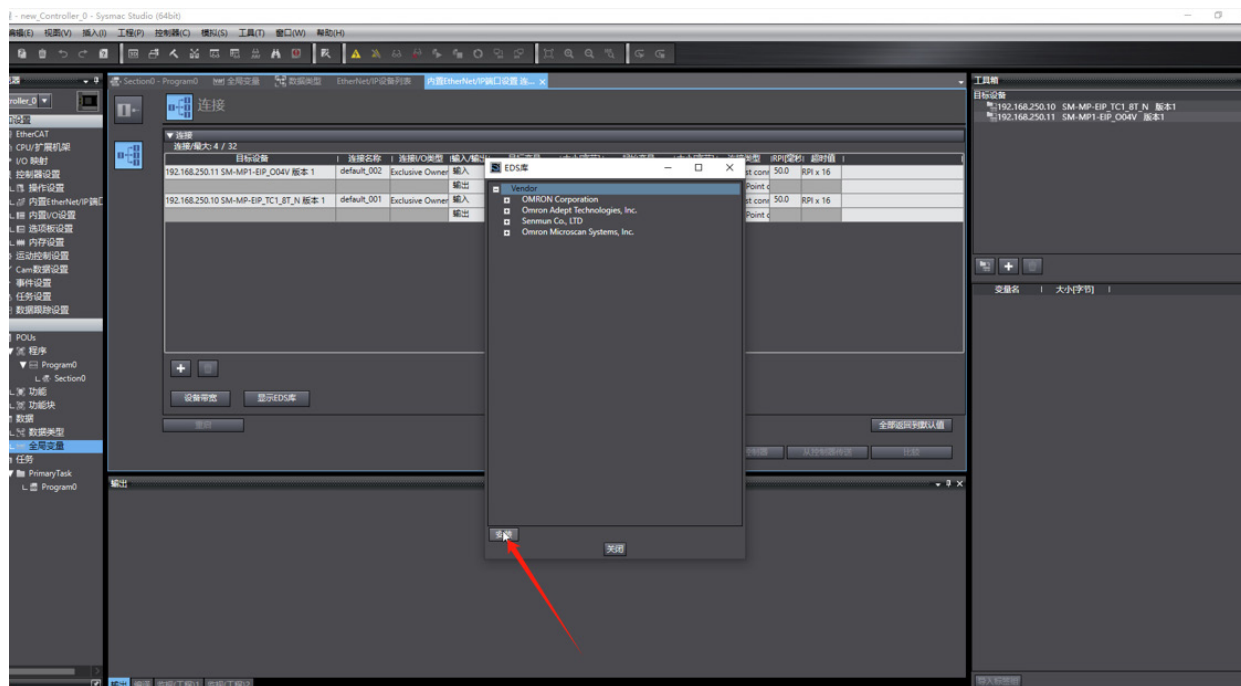
PLC: NX1P2

模块: TC1-8T-MP

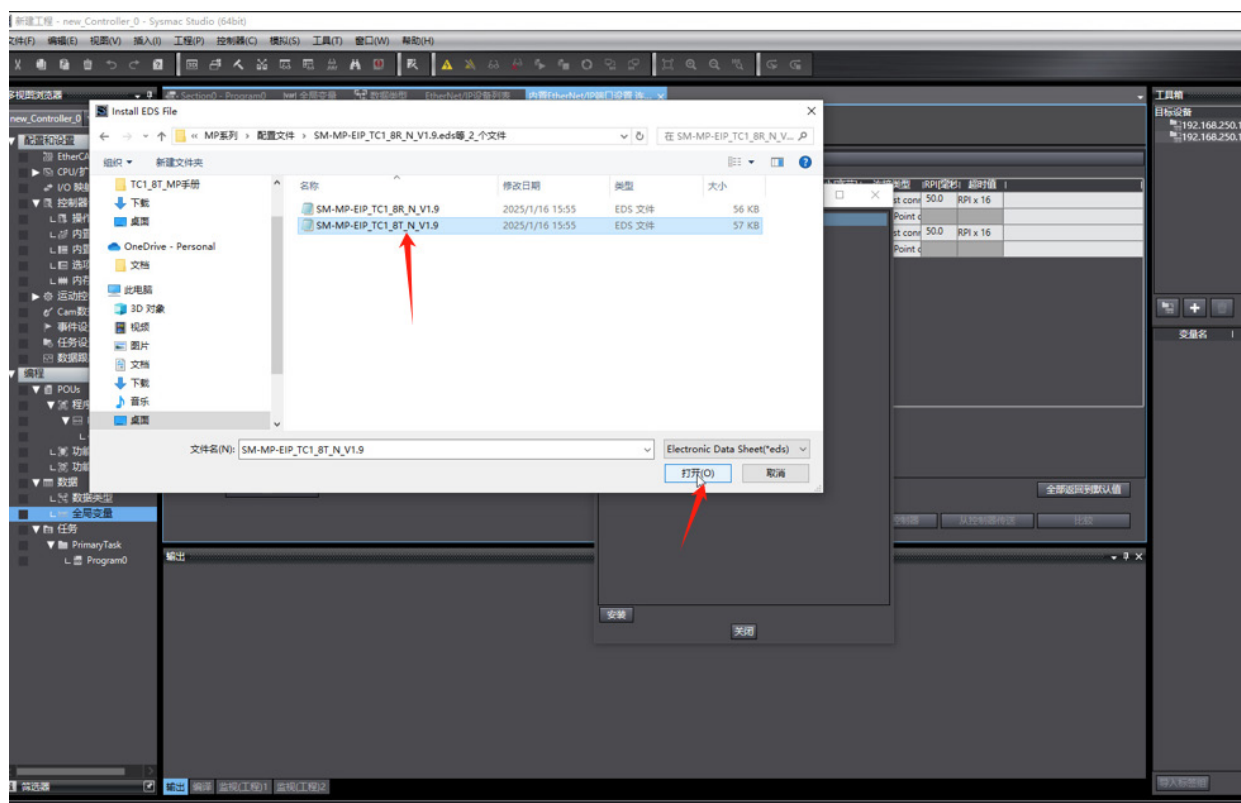
备注: TC1-8R-MP模块也可以参考此案例

1) 将TC1-8T-MP模块拨码拨到**2位置**, 连接好PLC和模块后, 打开EIP连接设置, 导入模块的配置文件, 如下操作

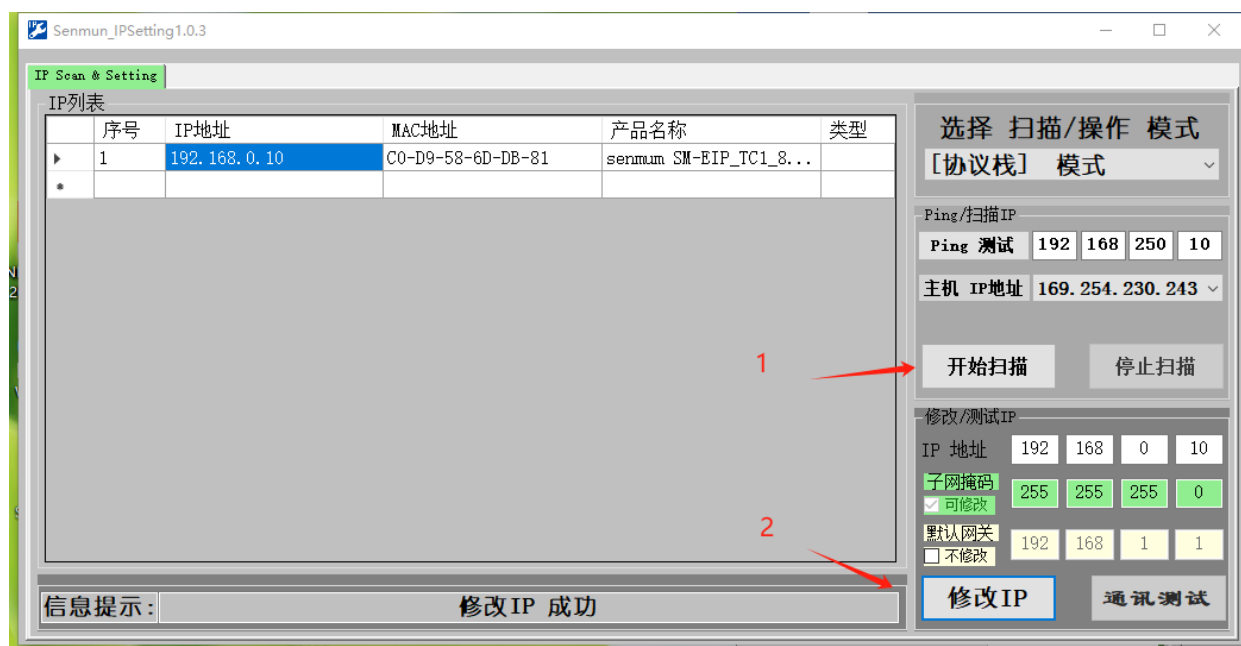




5 使用案例

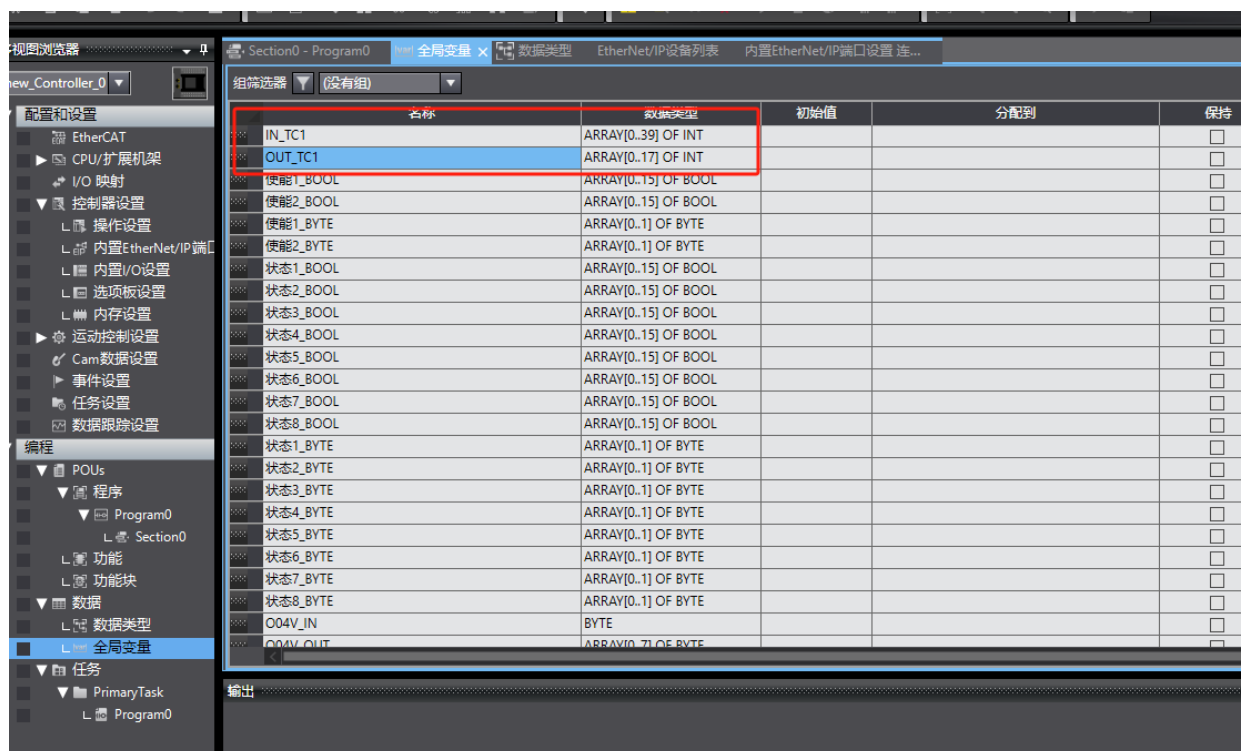


2) 用三铭公司的IP修改软件，修改模块的IP地和PLC的网段地址一致

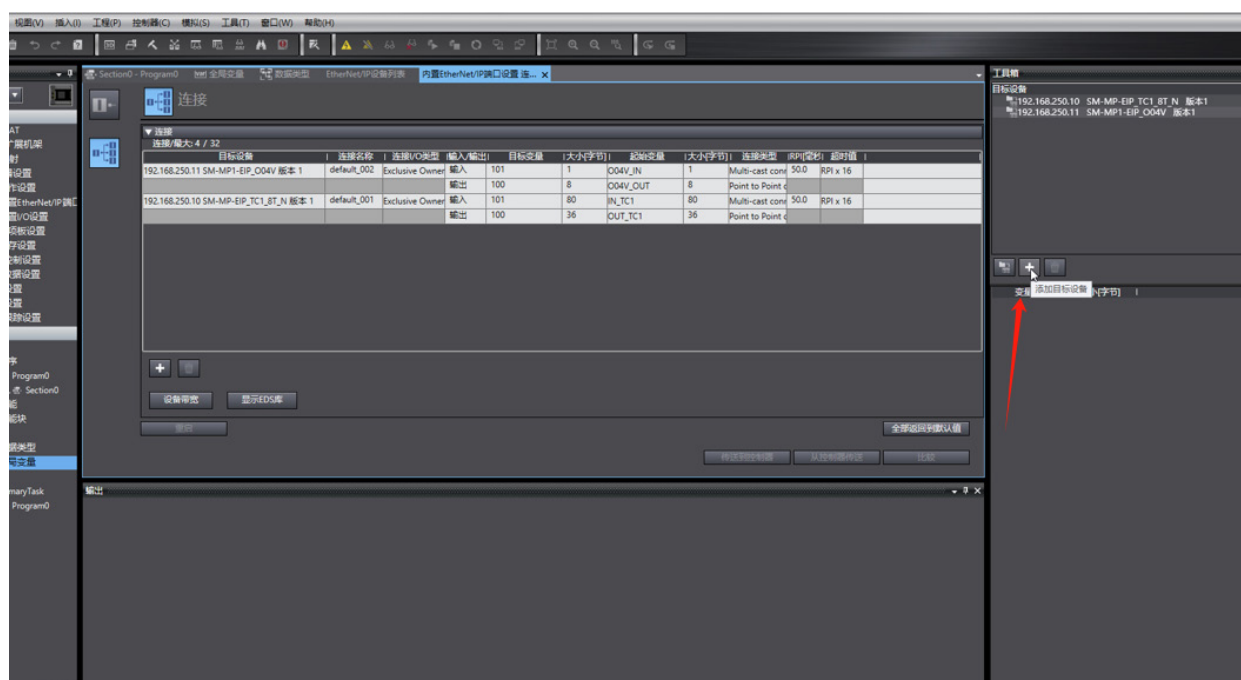


3) 组态模块

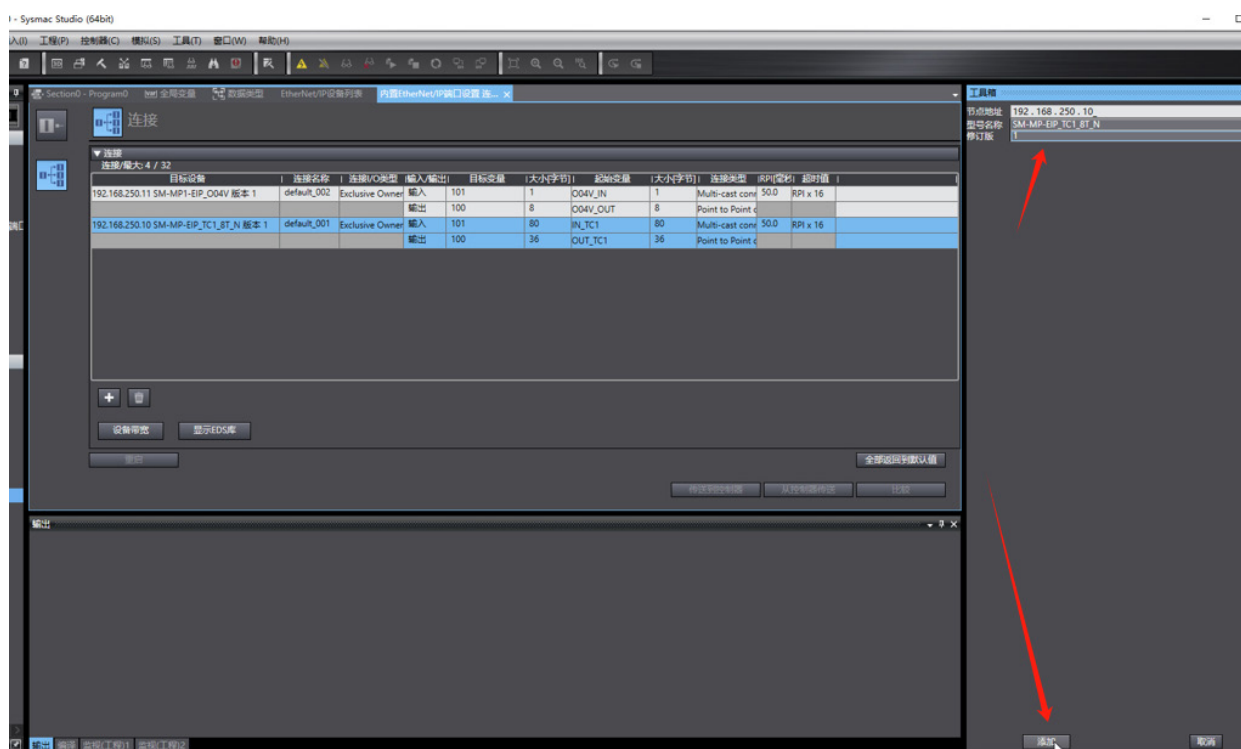
建立模块需要的全局变量



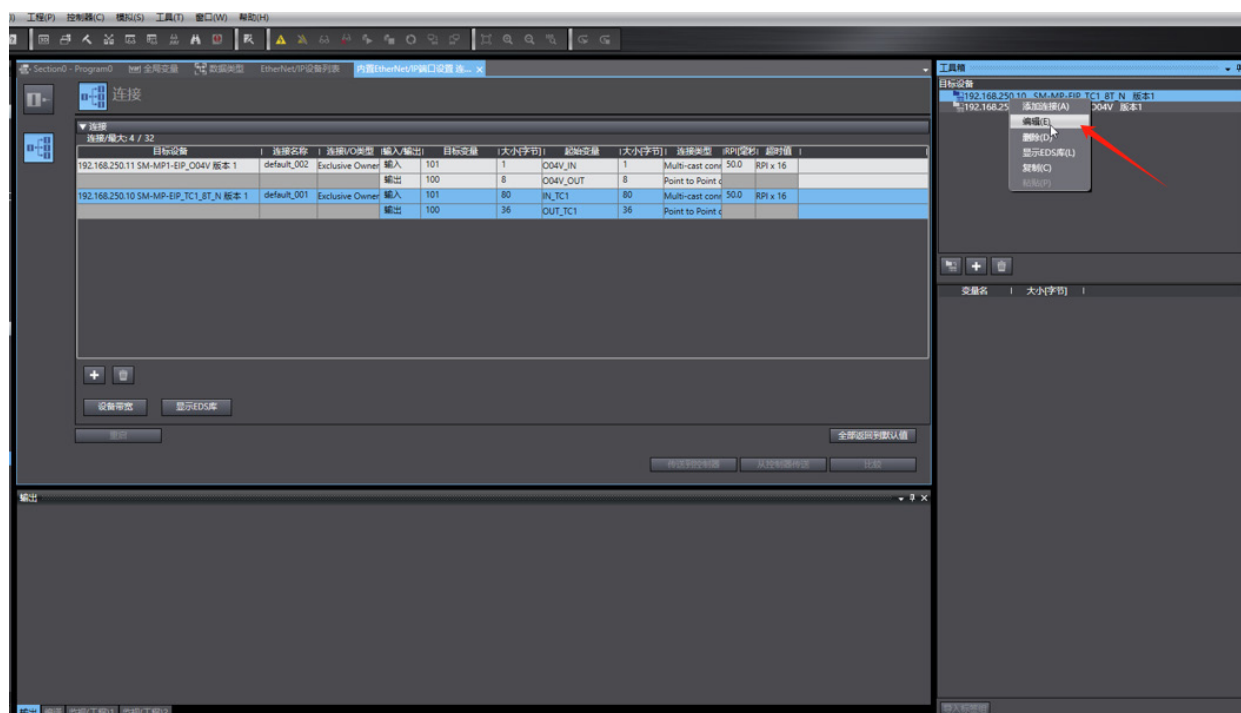
4) 点击+号，增加目标模块



5) 输入模块IP地址后，点击添加，如下



6)右键添加后的模块，点击编辑



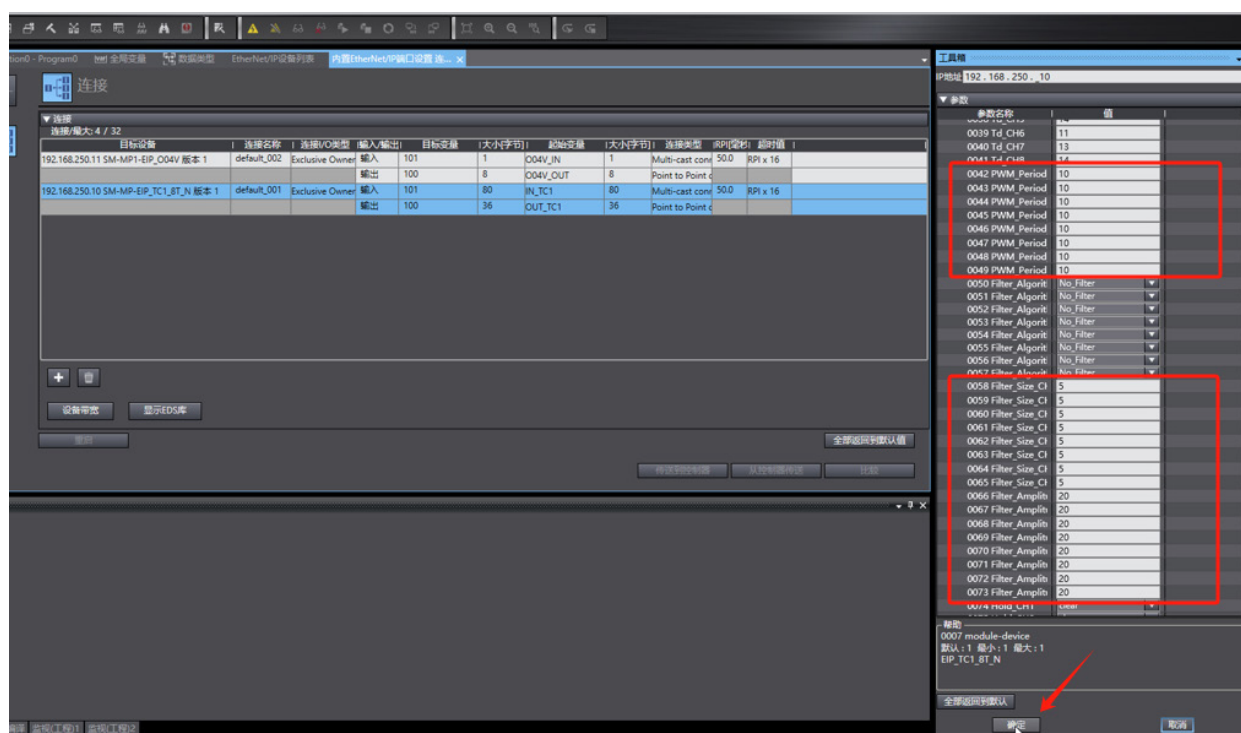
7)注意: Sensor Type和PWM period是**必须设定的**

推荐: PWM period设置为10, Filter size设置为5, Filter Amplitude设置为20, 其他可以不用设置

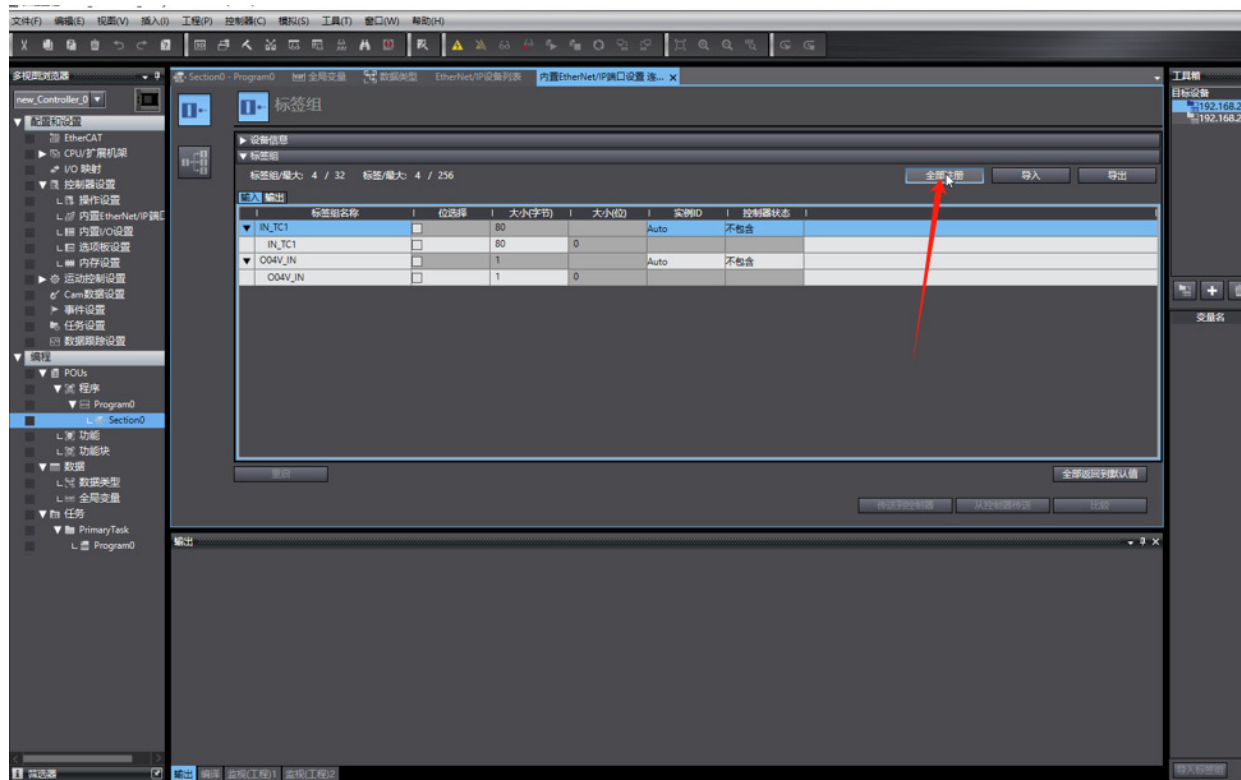
Kp, TI, Td分别是比例, 积分, 微分。这三个值需要在模块自整定得到各个通道的PID三个值后手动填入(自整定方法在后面步骤说明)

如下所示

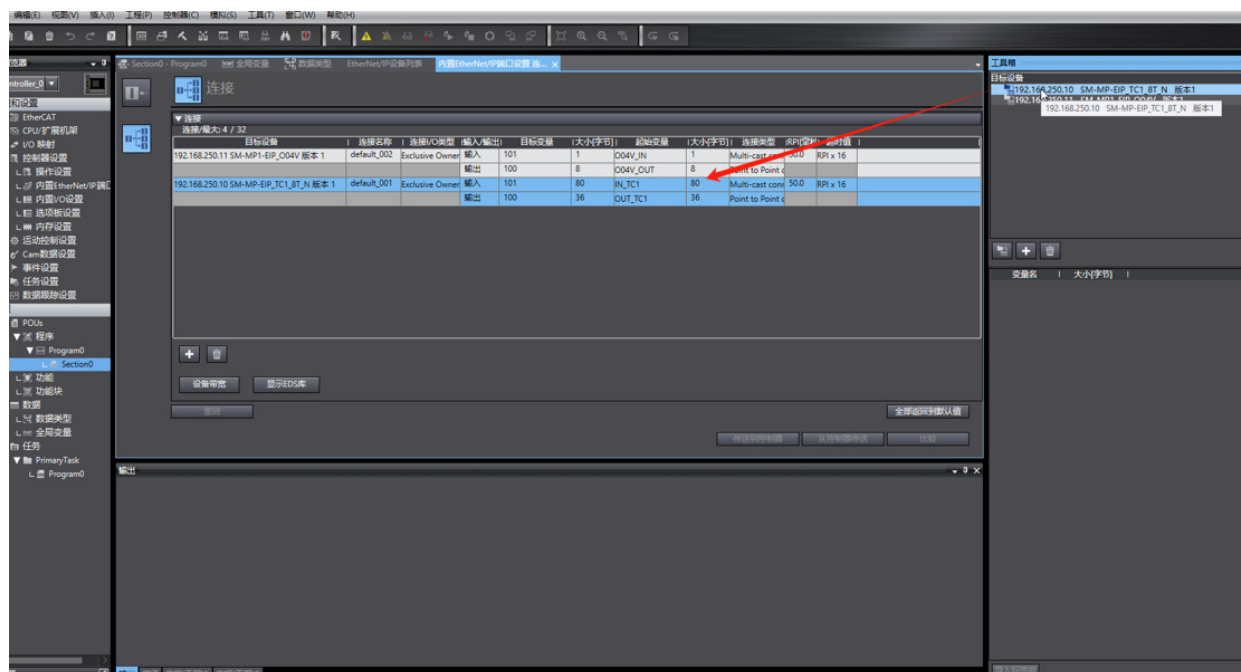
设置如下所示:



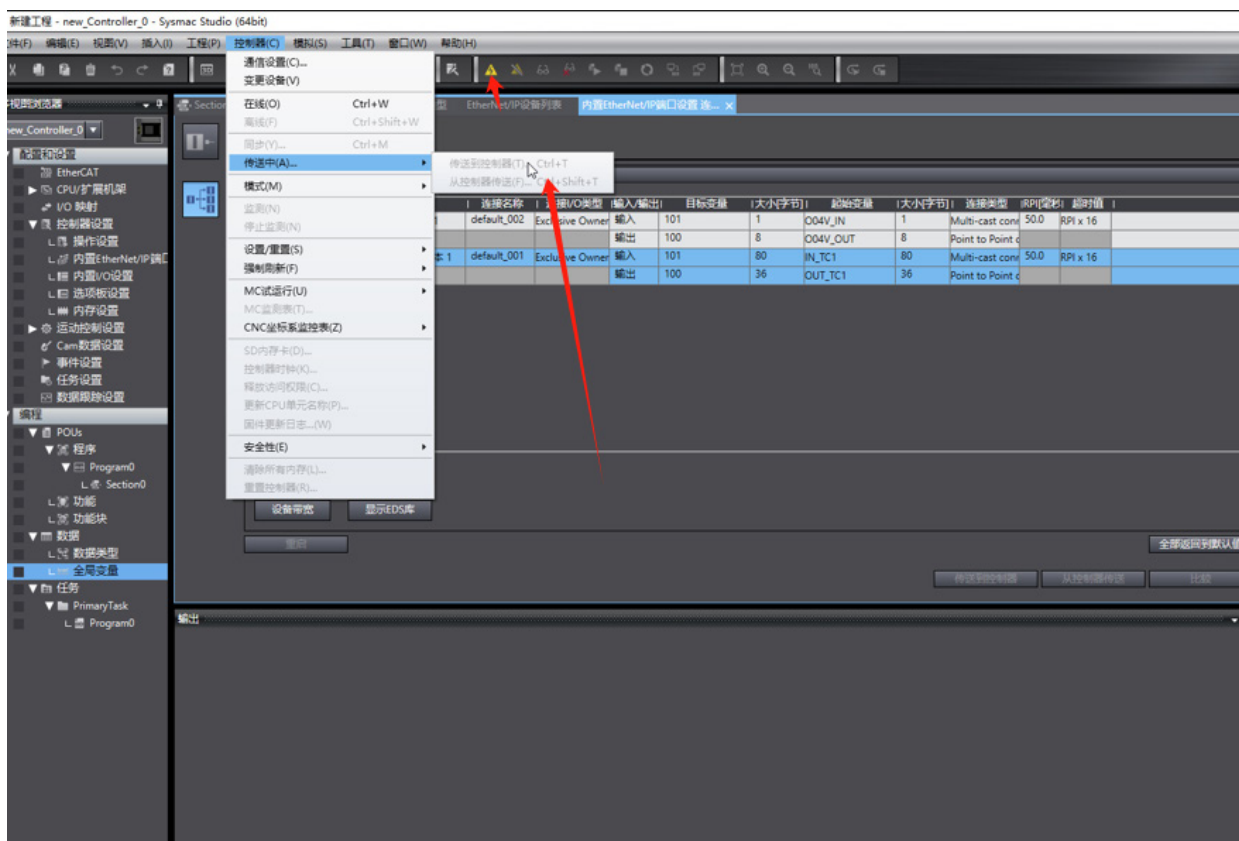
1) 点击注册变量，将建立的变量和模块关联起来



2) 将模块拖入界面中，填入如下信息即可



3) 连接到PLC后，下载程序进PLC



4) 使能，PID启动，自整定启动三个BOOL参数在变量地址中的排序说明，本案例是的输入起始变量为IN_TC1[0]和输出起始变量为OUT_TC1[0]

使能占用的2个字为OUT_TC1[0], OUT_TC1[1]

注：状态反馈里面的地址类似

_OUT_TC1[0]

CH8_PID启动	CH7_PID启动	CH6_PID启动	CH5_PID启动	CH4_PID启动	CH3_PID启动	CH2_PID启动	CH1_PID启动	CH 8 使能	CH 7 使能	CH 6 使能	CH 5 使能	CH 4 使能	CH 3 使能	CH 2 使能	CH 1 使能
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	---------	---------	---------	---------	---------	---------	---------	---------

_OUT_TC1[0].0

_OUT_TC1[1]

								CH 8 自整定启动	CH 7 自整定启动	CH 6 自整定启动	CH 5 自整定启动	CH 4 自整定启动	CH 3 自整定启动	CH 2 自整定启动	CH 1 自整定启动
--	--	--	--	--	--	--	--	------------	------------	------------	------------	------------	------------	------------	------------

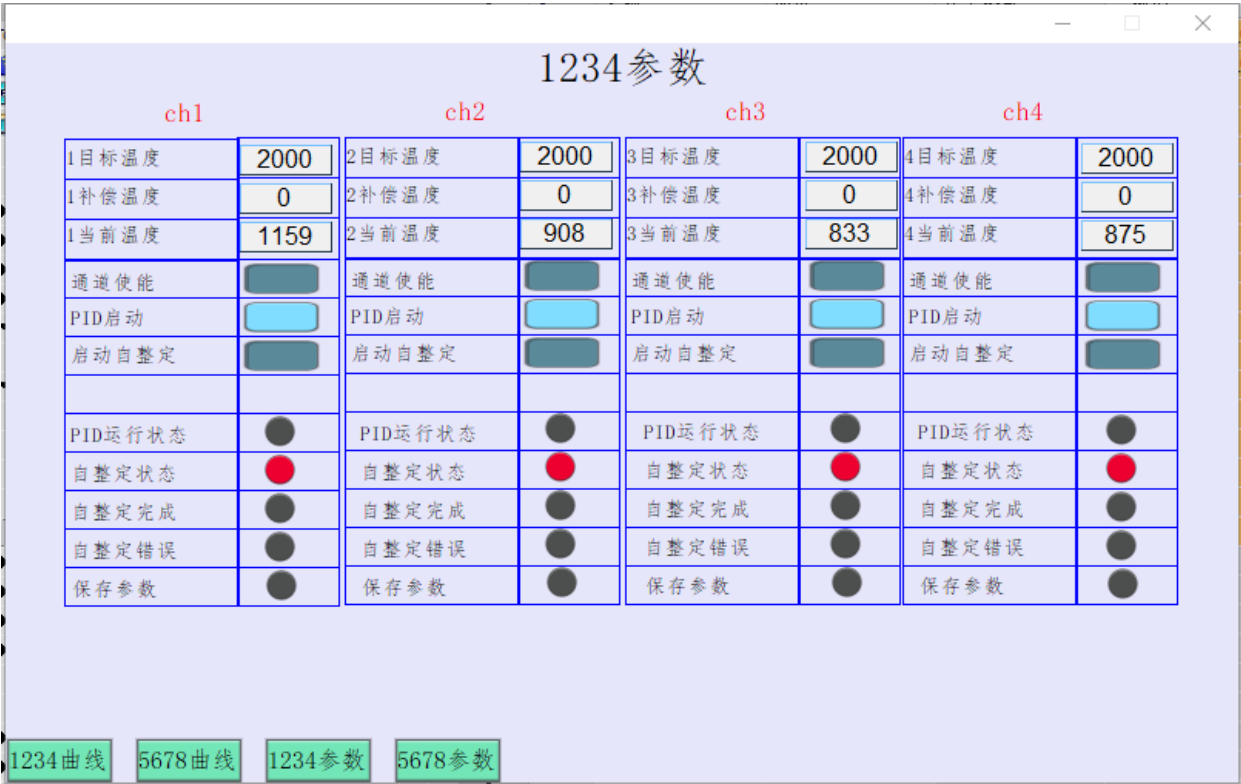
OUT_TC1[1].0

5) 启动自整定获取比例，积分，微分参数值

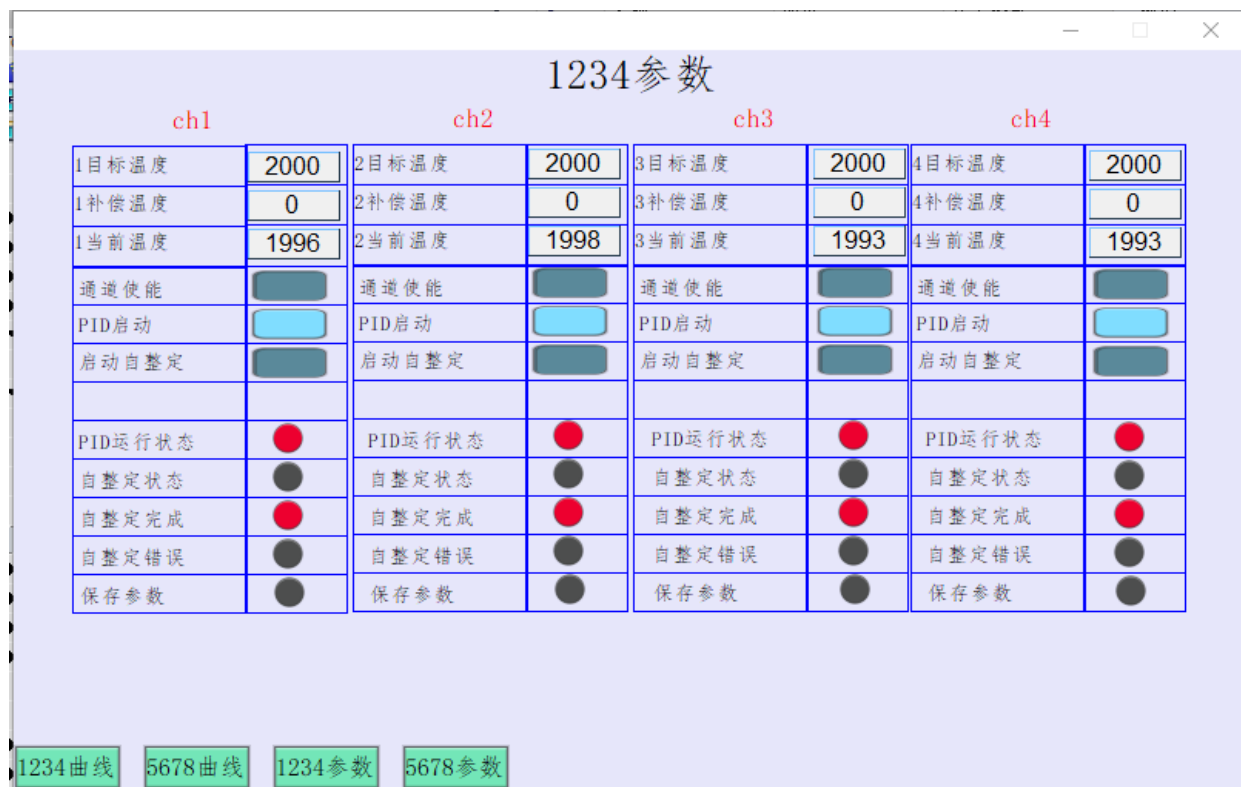
步骤1，设置目标温度后，使能相关通道（BOOL），启动相关通道自整定（BOOL），此时状态反馈的自整定运行中会接通（BOOL），此时外部设备进入加热阶段，再自动运行一段时间后，自整定完成，自整定完成状态会接通（BOOL），完成后在上行数据的PID参数的IW地址会出现得到的结果。完成需要通道的自整定操作后，将得到的PID参数值填入模块组态界面的相关属性值（以便下次启动，直接运行启动控温即可，不再需要经过自整定调节）

后续启动加温，直接启动PID运行（BOOL）即可。

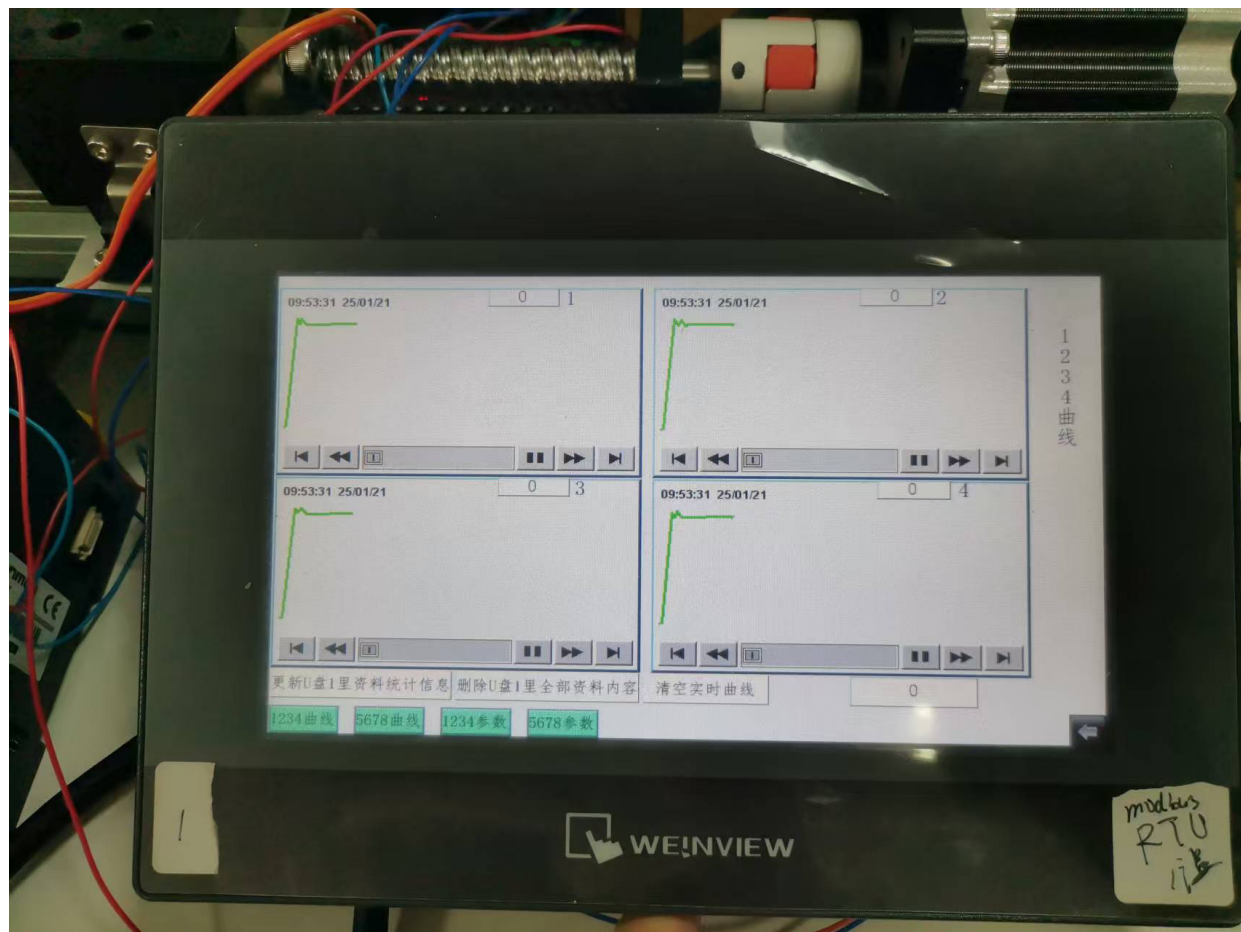
设定目标温度，使能通道，启动自整定后，自整定状态会接通



6) 自整定完成后, 相关反馈通道接通



7) 温度曲线的显示



8)得到的PID值

IN_TC1	ARRAY[0..39] OF INT		
OUT_TC1	ARRAY[0..17] OF INT		
使能1_BOOL	ARRAY[0..15] OF BOOL		
使能2_BOOL	ARRAY[0..15] OF BOOL		
使能1_BYTE	ARRAY[0..1] OF BYTE		
使能2_BYTE	ARRAY[0..1] OF BYTE		

监视(工程)2				
设备名称	名称	在线值	修改	
new_Controller_0	IN_TC1[16]	146		
new_Controller_0	IN_TC1[17]	117		
new_Controller_0	IN_TC1[18]	195		
new_Controller_0	IN_TC1[19]	146		
new_Controller_0	IN_TC1[20]	0		
new_Controller_0	IN_TC1[21]	0		
new_Controller_0	IN_TC1[22]	0		
new_Controller_0	IN_TC1[23]	0		
new_Controller_0	IN_TC1[24]	26		
new_Controller_0	IN_TC1[25]	18		
new_Controller_0	IN_TC1[26]	28		
new_Controller_0	IN_TC1[27]	25		
new_Controller_0	IN_TC1[28]	0		
new_Controller_0	IN_TC1[29]	0		
new_Controller_0	IN_TC1[30]	0		
new_Controller_0	IN_TC1[31]	0		
new_Controller_0	IN_TC1[32]	13		
new_Controller_0	IN_TC1[33]	9		
new_Controller_0	IN_TC1[34]	14		
new_Controller_0	IN_TC1[35]	12		
new_Controller_0	IN_TC1[36]	0		
new_Controller_0	IN_TC1[37]	0		
new_Controller_0	IN_TC1[38]	0		
new_Controller_0	IN_TC1[39]	0		
new_Controller_0	输入名称...			

9) 右键组态界面处模块，编辑，填入得到的PID值后，在下载程序即可

