

帖子标题:【SCL 编程案例】PCS7-CFC 利用 SCL 编程实现跳闸首出

帖子内容:

分享一个水泥余热发电 PCS7 (DCS) 项目, 利用 SCL 编程实现 CFC (Continuous Function Chart, 连续功能图) 自定义跳闸首出 (确定第一个引起汽轮机跳闸的首要条件) 的程序。该功能块有 20 个 DI 输入信号 (跳闸条件)、1 个输出信号 (YTripCH, 跳闸首出 DI 通道号)、1 个复位信号 (Reset, Bool 类型), 可确定 20 个输入信号当中哪个是引起汽轮机跳闸的首要条件。首要条件确定后, 必须通过复位信号且故障信号消失, 才能将输出信号清零。

控制系统选择西门子 S7-417H 冗余 DCS 系统, 通过 PCS7 V7.0 SP2 进行编程, 详细开发过程请见附录 (1. 硬件组态; 2. 网络组态; 3. 定义 DI 点; 4. SCL 编程; 5. 将 SCL 源文件编译为 CFC 可调用的功能块; 6. CFC 编程; 7. 总结)。

附录:

1. 硬件组态

该系统 CPU 选择 417-4H 冗余系统, 主备 CPU 切换时间 100ms 至 300ms 之间, 可实现余热发电系统控制系统无扰切换, 确保控制不中断, 过程状态不丢失。CPU 主站有四个分布式 EM200 子站, 分别是 APL 汽机子站、AQC 子站、SP1&SP2 子站、SP3&SP4&SP5 子站。主站与子站通过 Profibus-DP 通讯协议连接, 由于通讯距离较长, 使用 OLM 模块进行光电转换以实现长距离通讯, 如图 1 所示。

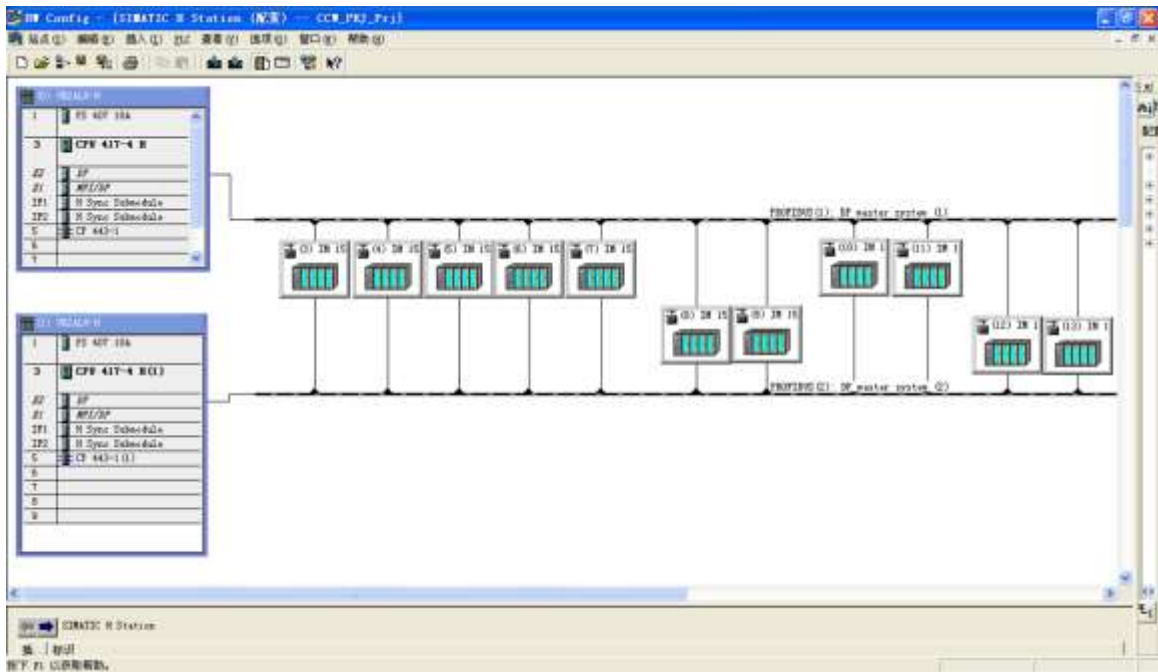


图 1 硬件组态

2. 网络组态

该系统有 1 个 ES（工程师站）、5 个 OS（操作员站）。PCS7 系统可以通过 ES 站将上位机程序便捷的下载到指定的 OS 站，避免了通过 U 盘或设置共享硬盘拷贝程序的繁杂操作，如图 2 所示。

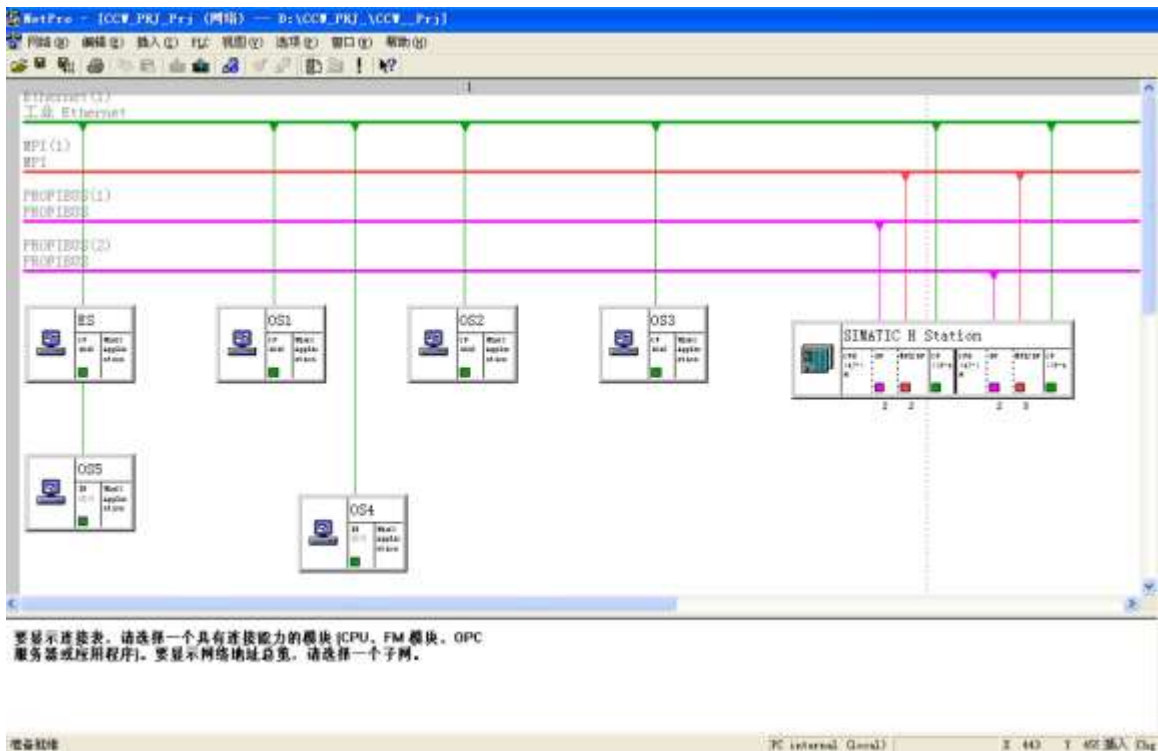


图 2 网络组态

3. 定义 DI 点

利用导入/导出功能，快速将 IO 清单导入到 DB 块中，并设置变量在 PCS 系统是可见的 (S7_m_c:='true' )，如图 3 所示。

Module	Variable	Value	Description
+20.6	Gen_TRIP	BOOL	Generator Trip
+20.6	I10_5	BOOL	DI10 Spare1
+20.6	I10_6	BOOL	DI10 Spare2
+20.6	I10_7	BOOL	DI10 Spare3
+20.6	I2D1180_1	BOOL	Big Front Bearing Vibration Alarm
+20.6	I2D1180_2	BOOL	Big Front Bearing Vibration Stop
+20.6	I2D1181_1	BOOL	Big Rear Bearing Vibration Alarm
+20.6	I2D1181_2	BOOL	Big Rear Bearing Vibration Stop
+20.6	TS1190_1	BOOL	Turbine Overspeed Alarm
+20.6	TS1190_2	BOOL	Turbine Overspeed Stop
+20.6	TP1190_A	BOOL	Low Lubricating Oil Pressure A
+20.6	TP1190_B	BOOL	Low Lubricating Oil Pressure B
+20.6	TP1190_C	BOOL	Low Lubricating Oil Pressure C
+20.6	TP1190_D	BOOL	Low Lubricating Oil Pressure D
+20.6	TP1190_E	BOOL	Low Lubricating Oil Pressure E
+20.6	I2D1180_1	BOOL	Big Front Bearing Vibration Alarm of Generator
+20.6	I2D1180_2	BOOL	Big Front Bearing Vibration Stop of Generator
+20.6	I2D1181_1	BOOL	Big Rear Bearing Vibration Alarm of Generator
+20.6	I2D1181_2	BOOL	Big Rear Bearing Vibration Stop of Generator
+20.6	I21_1	BOOL	DI11 Spare1
+20.6	TP1190_A	BOOL	Low Condenser Vacuum Alarm
+20.6	TP1190_B	BOOL	Low Condenser Vacuum Stop
+20.6	TP1191	BOOL	Low Turbine Main Oil Pump Exit Pressure
+20.6	TP1192	BOOL	Low Security Oil Pressure
+20.6	TA1001	BOOL	Manual Emergency Shutdown Enable
+20.6	SO1_Emergency Shutdown	BOOL	SO1 Emergency Shutdown (From SO1)

图 3 定义 DI 变量

4. SCL 编程

通过 PCS7 SCL 编辑器根据应用场景及业务逻辑进行编程，如图 4 所示。

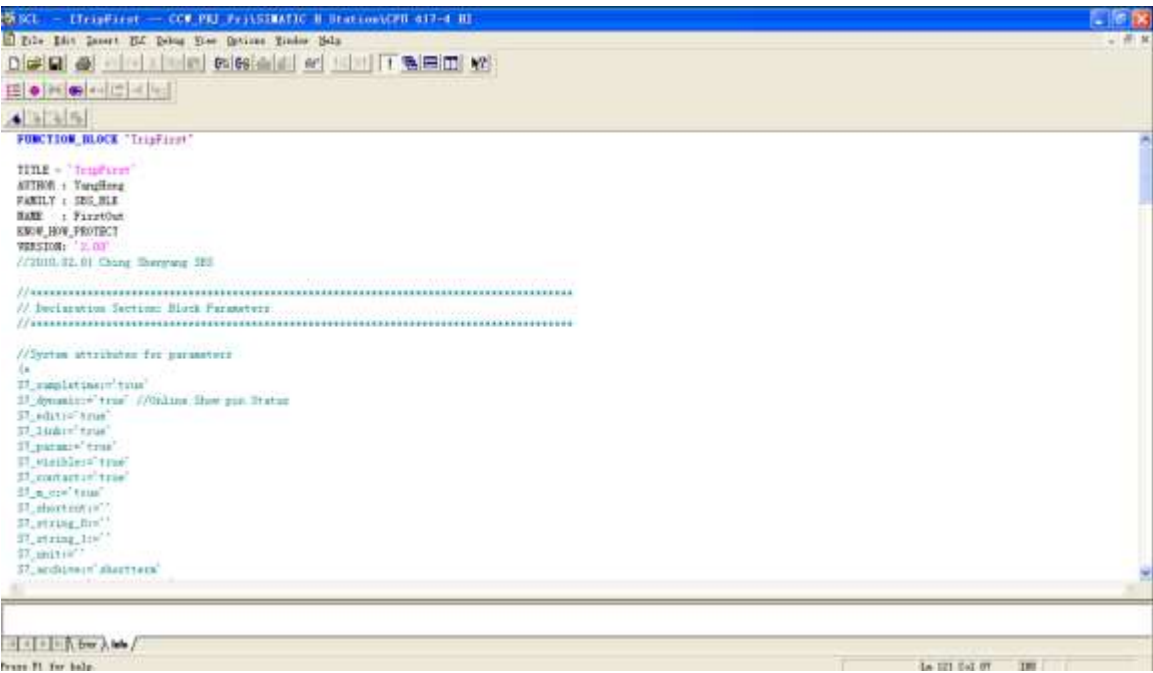


图 4 SCL 编程

主要源程序如下：

```
FUNCTION_BLOCK "TripFirst"

TITLE = 'TripFirst'

AUTHOR : YangHong

FAMILY : SBS_BLK

NAME   : FirstOut

KNOW_HOW_PROTECT

VERSION: '2.00'

//2010.02.01 Ching Shenyang

//*****

// Declaration Section: Block Parameters

//*****
```

```
//System attributes for parameters
```

```
(*
```

```
S7_sampletime:='true'
```

```
S7_dynamic:='true' //Online Show pin Status
```

```
S7_edit:='true'
```

```
S7_link:='true'
```

```
S7_param:='true'
```

```
S7_visible:='true'
```

```
S7_contact:='true'
```

```
S7_m_c:='true'
```

```
S7_shortcut:=''
```

```
S7_string_0:=''
```

```
S7_string_1:=''
```

```
S7_unit:=''
```

```
S7_archive:='shortterm'
```

```
S7_server:='alarm_archiv'
```

```
S7_a_type:='alarm_8p'
```

```
*)
```

```
VAR_INPUT
```

```
IN1 {S7_dynamic:='true';S7_m_c:='true'}:BOOL:=FALSE;
```

```
IN2 {S7_dynamic:='true';S7_m_c:='true'}:BOOL:=FALSE;
```

```
IN3 {S7_dynamic:='true';S7_m_c:='true'}:BOOL:=FALSE;
```

```
IN4 {S7_dynamic:='true';S7_m_c:='true'}:BOOL:=FALSE;
```

```
IN5 {S7_dynamic:='true';S7_m_c:='true'}:BOOL:=FALSE;
```

```
IN6 {S7_dynamic:='true';S7_m_c:='true'}:BOOL:=FALSE;
```

```
IN7{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN8{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN9{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN10{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN11{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN12{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN13{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN14{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN15{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN16{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN17{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN18{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN19{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
IN20{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
```

```
Reset:BOOL:=FALSE; //Reset Sign:Pulse
```

```
END_VAR
```

```
VAR_OUTPUT
```

```
OUT1{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
OUT2{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
OUT3{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
OUT4{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
OUT5{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
OUT6{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
OUT7{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
OUT8{S7_dynamic:= ' true' ;S7_m_c:= ' true' }:BOOL:=FALSE;
```

```

OUT9{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;
OUT10{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;
OUT11{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;
OUT12{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;
OUT13{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;
OUT14{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;
OUT15{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;
OUT16{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;
OUT17{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;
OUT18{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;
OUT19{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;
OUT20{S7_dynamic:=' true' ;S7_m_c:=' true'}:BOOL:=FALSE;

YAlarm:BOOL:=FALSE;  //First:Alarm

YTripCH:INT:=0;      //First:Channel Number

YStep:INT:=0;

END_VAR

LABEL

    YAlarmFirst;//Jump Lable

END_LABEL

VAR_TEMP

//Here Define Temp Variable

END_VAR

BEGIN

    IF Reset = 0 THEN

```

```

IF YAlarm = 0 THEN

    YStep:=1;

    IF IN1 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=1;OUT1:=TRUE;END_IF;
    IF IN2 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=2;OUT2:=TRUE;END_IF;
    IF IN3 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=3;OUT3:=TRUE;END_IF;
    IF IN4 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=4;OUT4:=TRUE;END_IF;
    IF IN5 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=5;OUT5:=TRUE;END_IF;
    IF IN6 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=6;OUT6:=TRUE;END_IF;
    IF IN7 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=7;OUT7:=TRUE;END_IF;
    IF IN8 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=8;OUT8:=TRUE;END_IF;
    IF IN9 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=9;OUT9:=TRUE;END_IF;
    IF IN10 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=10;OUT10:=TRUE;END_IF;
    IF IN11 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=11;OUT11:=TRUE;END_IF;
    IF IN12 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=12;OUT12:=TRUE;END_IF;
    IF IN13 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=13;OUT13:=TRUE;END_IF;
    IF IN14 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=14;OUT14:=TRUE;END_IF;
    IF IN15 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=15;OUT15:=TRUE;END_IF;
    IF IN16 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=16;OUT16:=TRUE;END_IF;
    IF IN17 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=17;OUT17:=TRUE;END_IF;
    IF IN18 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=18;OUT18:=TRUE;END_IF;
    IF IN19 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=19;OUT19:=TRUE;END_IF;
    IF IN20 = 1 AND YAlarm=FALSe THEN  YAlarm:=TRUE;YTripCH:=20;OUT20:=TRUE;END_IF;

ELSIF YAlarm = 1 THEN

    YStep:=2;

END_IF;

ELSIF Reset = 1 THEN

    IF YAlarm = 0 THEN

```

```

YTripCH:=0;

YStep:=3;

ELSIF YAlarm = 1 THEN

    YStep:=4;

    CASE YTripCH OF

        1:IF IN1 = 1 THEN ;ELSE YAlarm :=FALSE;OUT1:=FALSE;GOTO
                YAlarmFirst;END_IF;

        2:IF IN2 = 1 THEN ;ELSE YAlarm :=FALSE;OUT2:=FALSE;GOTO
                YAlarmFirst;END_IF;

        3:IF IN3 = 1 THEN ;ELSE YAlarm :=FALSE;OUT3:=FALSE;GOTO
                YAlarmFirst;END_IF;

        4:IF IN4 = 1 THEN ;ELSE YAlarm :=FALSE;OUT4:=FALSE;GOTO
                YAlarmFirst;END_IF;

        5:IF IN5 = 1 THEN ;ELSE YAlarm :=FALSE;OUT5:=FALSE;GOTO
                YAlarmFirst;END_IF;

        6:IF IN6 = 1 THEN ;ELSE YAlarm :=FALSE;OUT6:=FALSE;GOTO
                YAlarmFirst;END_IF;

        7:IF IN7 = 1 THEN ;ELSE YAlarm :=FALSE;OUT7:=FALSE;GOTO
                YAlarmFirst;END_IF;

        8:IF IN8 = 1 THEN ;ELSE YAlarm :=FALSE;OUT8:=FALSE;GOTO
                YAlarmFirst;END_IF;

        9:IF IN9 = 1 THEN ;ELSE YAlarm :=FALSE;OUT9:=FALSE;GOTO
                YAlarmFirst;END_IF;

        10:IF IN10 = 1 THEN ;ELSE YAlarm :=FALSE;OUT10:=FALSE;GOTO
                YAlarmFirst;END_IF;

        11:IF IN11 = 1 THEN ;ELSE YAlarm :=FALSE;OUT11:=FALSE;GOTO
                YAlarmFirst;END_IF;

```

```

12:IF IN12 = 1 THEN ;ELSE YAlarm :=FALSE;OUT12:=FALSE;GOTO
        YAlarmFirst;END_IF;
13:IF IN13 = 1 THEN ;ELSE YAlarm :=FALSE;OUT13:=FALSE;GOTO
        YAlarmFirst;END_IF;
14:IF IN14 = 1 THEN ;ELSE YAlarm :=FALSE;OUT14:=FALSE;GOTO
        YAlarmFirst;END_IF;
15:IF IN15 = 1 THEN ;ELSE YAlarm :=FALSE;OUT15:=FALSE;GOTO
        YAlarmFirst;END_IF;
16:IF IN16 = 1 THEN ;ELSE YAlarm :=FALSE;OUT16:=FALSE;GOTO
        YAlarmFirst;END_IF;
17:IF IN17 = 1 THEN ;ELSE YAlarm :=FALSE;OUT17:=FALSE;GOTO
        YAlarmFirst;END_IF;
18:IF IN18 = 1 THEN ;ELSE YAlarm :=FALSE;OUT18:=FALSE;GOTO
        YAlarmFirst;END_IF;
19:IF IN19 = 1 THEN ;ELSE YAlarm :=FALSE;OUT19:=FALSE;GOTO
        YAlarmFirst;END_IF;
20:IF IN20 = 1 THEN ;ELSE YAlarm :=FALSE;OUT20:=FALSE;GOTO
        YAlarmFirst;END_IF;

ELSE

        YAlarm:=FALSE;

        YStep:=5;

END_CASE;

ELSE//

        YStep:=6;;

END_IF;

ELSE

YStep:=7;

```

```

;
END_IF;
RETURN;

YAlarmFirst:
YStep:=8;

IF IN1 = 1 AND Reset = 1 THEN  YAlarm:=TRUE;YTripCH:=1;OUT1:=TRUE;RETURN;END_IF;
IF IN2 = 1 AND Reset = 1 THEN  YAlarm:=TRUE;YTripCH:=2;OUT2:=TRUE;RETURN;END_IF;
IF IN3 = 1 AND Reset = 1 THEN  YAlarm:=TRUE;YTripCH:=3;OUT3:=TRUE;RETURN;END_IF;
IF IN4 = 1 AND Reset = 1 THEN  YAlarm:=TRUE;YTripCH:=4;OUT4:=TRUE;RETURN;END_IF;
IF IN5 = 1 AND Reset = 1 THEN  YAlarm:=TRUE;YTripCH:=5;OUT5:=TRUE;RETURN;END_IF;
IF IN6 = 1 AND Reset = 1 THEN  YAlarm:=TRUE;YTripCH:=6;OUT6:=TRUE;RETURN;END_IF;
IF IN7 = 1 AND Reset = 1 THEN  YAlarm:=TRUE;YTripCH:=7;OUT7:=TRUE;RETURN;END_IF;
IF IN8 = 1 AND Reset = 1 THEN  YAlarm:=TRUE;YTripCH:=8;OUT8:=TRUE;RETURN;END_IF;
IF IN9 = 1 AND Reset = 1 THEN  YAlarm:=TRUE;YTripCH:=9;OUT9:=TRUE;RETURN;END_IF;

IF IN10 = 1 AND Reset = 1  THEN
YAlarm:=TRUE;YTripCH:=10;OUT10:=TRUE;RETURN;END_IF;

IF IN11 = 1 AND Reset = 1  THEN
YAlarm:=TRUE;YTripCH:=11;OUT11:=TRUE;RETURN;END_IF;

IF IN12 = 1 AND Reset = 1 THEN
YAlarm:=TRUE;YTripCH:=12;OUT12:=TRUE;RETURN;END_IF;

IF IN13 = 1 AND Reset = 1 THEN
YAlarm:=TRUE;YTripCH:=13;OUT13:=TRUE;RETURN;END_IF;

IF IN14 = 1 AND Reset = 1 THEN
YAlarm:=TRUE;YTripCH:=14;OUT14:=TRUE;RETURN;END_IF;

IF IN15 = 1 AND Reset = 1 THEN
YAlarm:=TRUE;YTripCH:=15;OUT15:=TRUE;RETURN;END_IF;

IF IN16 = 1 AND Reset = 1 THEN

```

```

YAlarm:=TRUE;YTripCH:=16;OUT16:=TRUE;RETURN;END_IF;

    IF IN17 = 1 AND Reset = 1 THEN

YAlarm:=TRUE;YTripCH:=17;OUT17:=TRUE;RETURN;END_IF;

    IF IN18 = 1 AND Reset = 1 THEN

YAlarm:=TRUE;YTripCH:=18;OUT18:=TRUE;RETURN;END_IF;

    IF IN19 = 1 AND Reset = 1 THEN

YAlarm:=TRUE;YTripCH:=19;OUT19:=TRUE;RETURN;END_IF;

    IF IN20 = 1 AND Reset = 1 THEN

YAlarm:=TRUE;YTripCH:=20;OUT20:=TRUE;RETURN;END_IF;

    IF YAlarm = 0 AND Reset = 1 THEN

        YAlarm :=FALSE;YTripCH :=0;YStep:=0;

OUT1:=FALSE;OUT2:=FALSE;OUT3:=FALSE;OUT4:=FALSE;OUT5:=FALSE;OUT6:=FALSE;OUT7:=FALSE;OU
T8:=FALSE;OUT9:=FALSE;OUT10:=FALSE;

OUT11:=FALSE;OUT12:=FALSE;OUT13:=FALSE;OUT14:=FALSE;OUT15:=FALSE;OUT16:=FALSE;OUT17:=F
ALSE;OUT18:=FALSE;OUT19:=FALSE;OUT20:=FALSE;

        END_IF;

    RETURN;

END_FUNCTION_BLOCK

```

5. 将 SCL 源文件编译为 CFC 可调用的功能块

SCL 源文件编写完成后，将源文件编译，如图 5 所示。

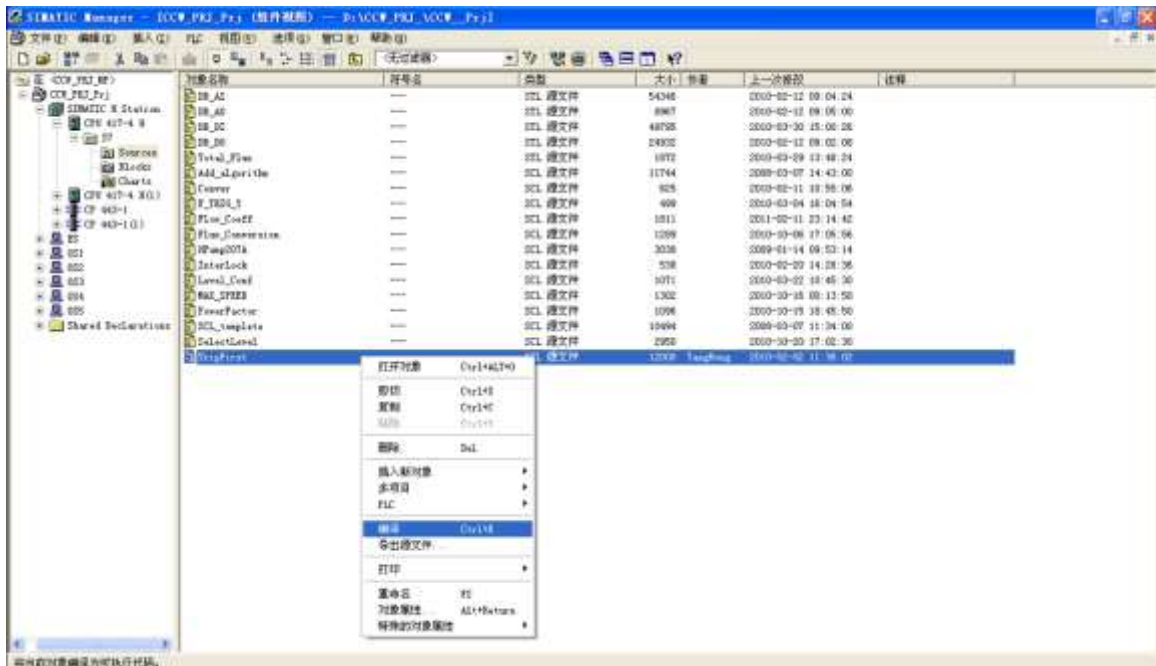


图 5 编译 SCL 源文件

编译后可生成 CFC 跳闸首出功能块，可右键【属性】查看管脚的说明，如图 6 所示。

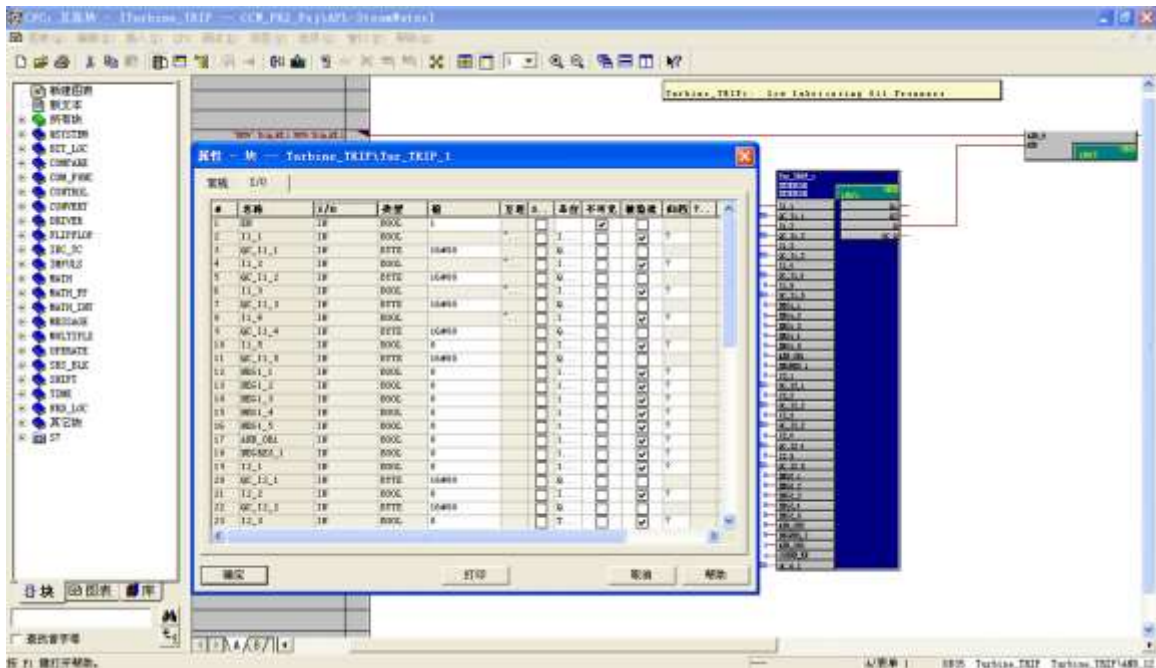


图 6 生成跳闸首出功能块及管脚说明

6. CFC 编程

在 CFC 编辑环境中，建立跳闸首出模块 I1_1 输入与 DI.TPS1300_A 点位的连接；I1_2 输入与 DI.TPS1300_B 点位的连接；I1_3 输入与 DI.TPS1300_C 点位的连接；I1_4 输入与

DI. TPS1300_D 点位的连接；如图 7 所示。

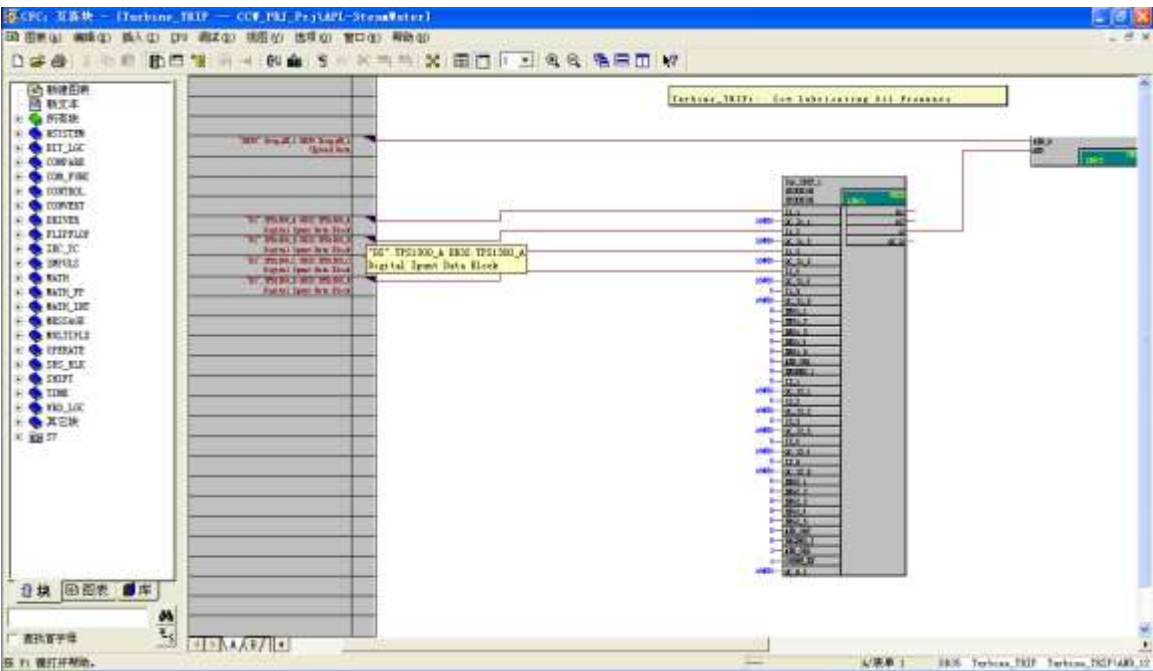


图 7 CFC 编程

7. 总结

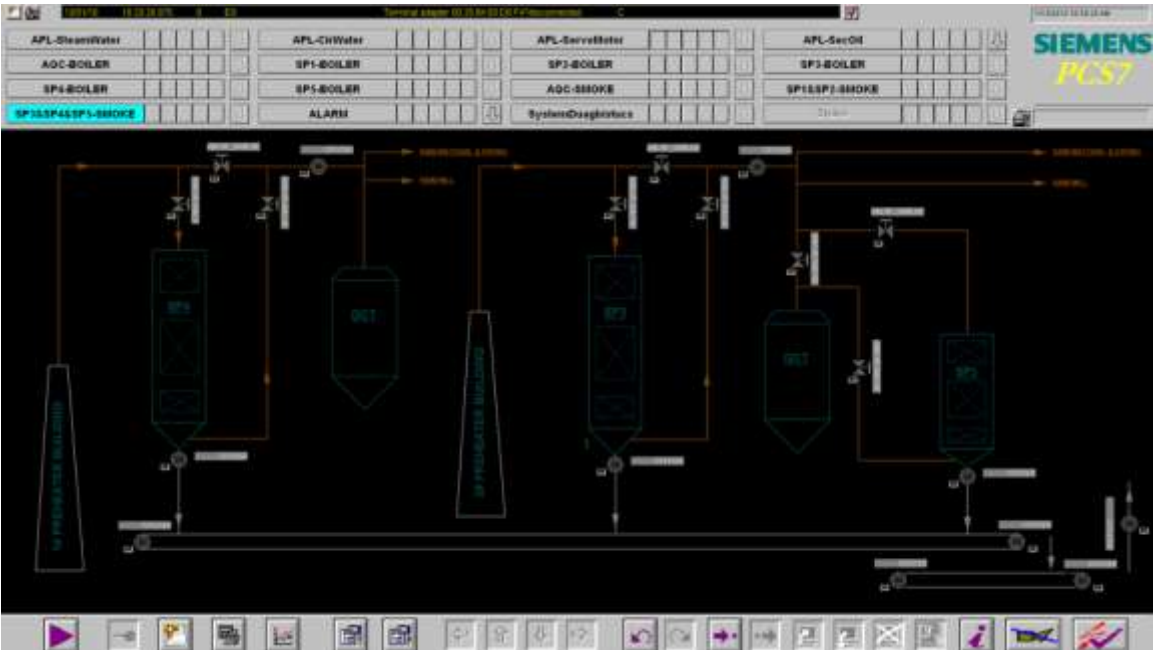


图 8 水泥余热发电工艺流程

基于 SCL 编程构建的核心控制系统自 2010 年 8 月正式投运至今，已连续安全稳定运行近 15 年。该系统依托 SCL 语言严谨的结构化特性、强数据类型校验及模块化设计优势，在

长期高负荷运行中始终保持零重大事故记录，充分验证了 SCL 在工业关键控制场景中的超高可靠性与鲁棒性。其代码执行的确定性、异常处理机制的完备性，以及冗余架构下的无缝故障切换能力，为装置的长周期安全运行提供了坚实的技术保障，是工业自动化领域经得起时间考验的编程典范。