流量与温度控制回路

被控对象是一个正在加热的液体灌。液体灌的上方有一个温度测量仪表，用于测量罐内温度，侧面有一个调节阀，用于控制输入罐内液体的流速。罐内温度需保持在一定范围内，不能出现大幅度的升降。当温度发生变化后，调整阀门开合程度，使温度保持在一个范围内。罐内液体的高度不考虑。当罐体温度过高或者过低，需要使用动态特性进行报警并显示。

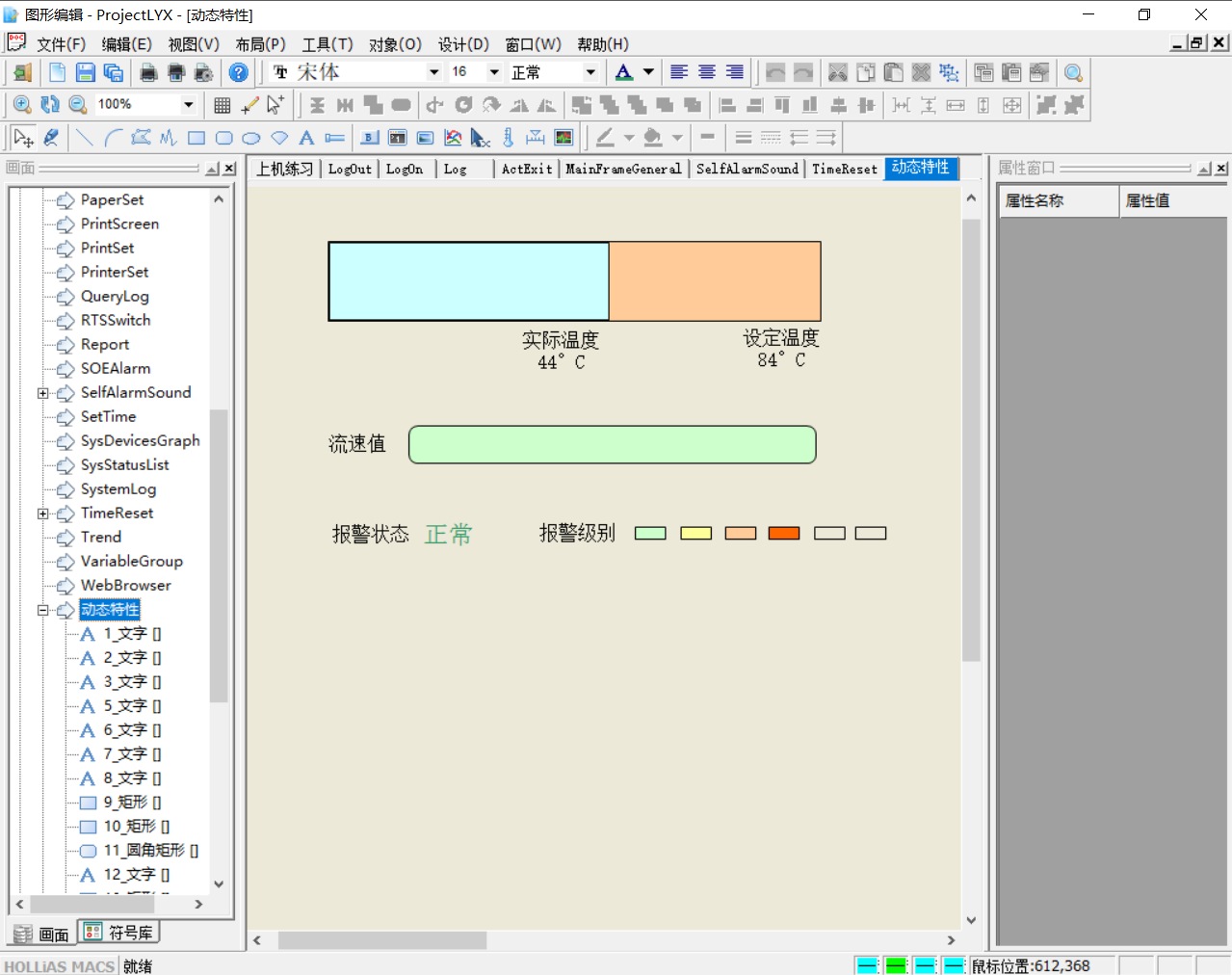
**要求：**

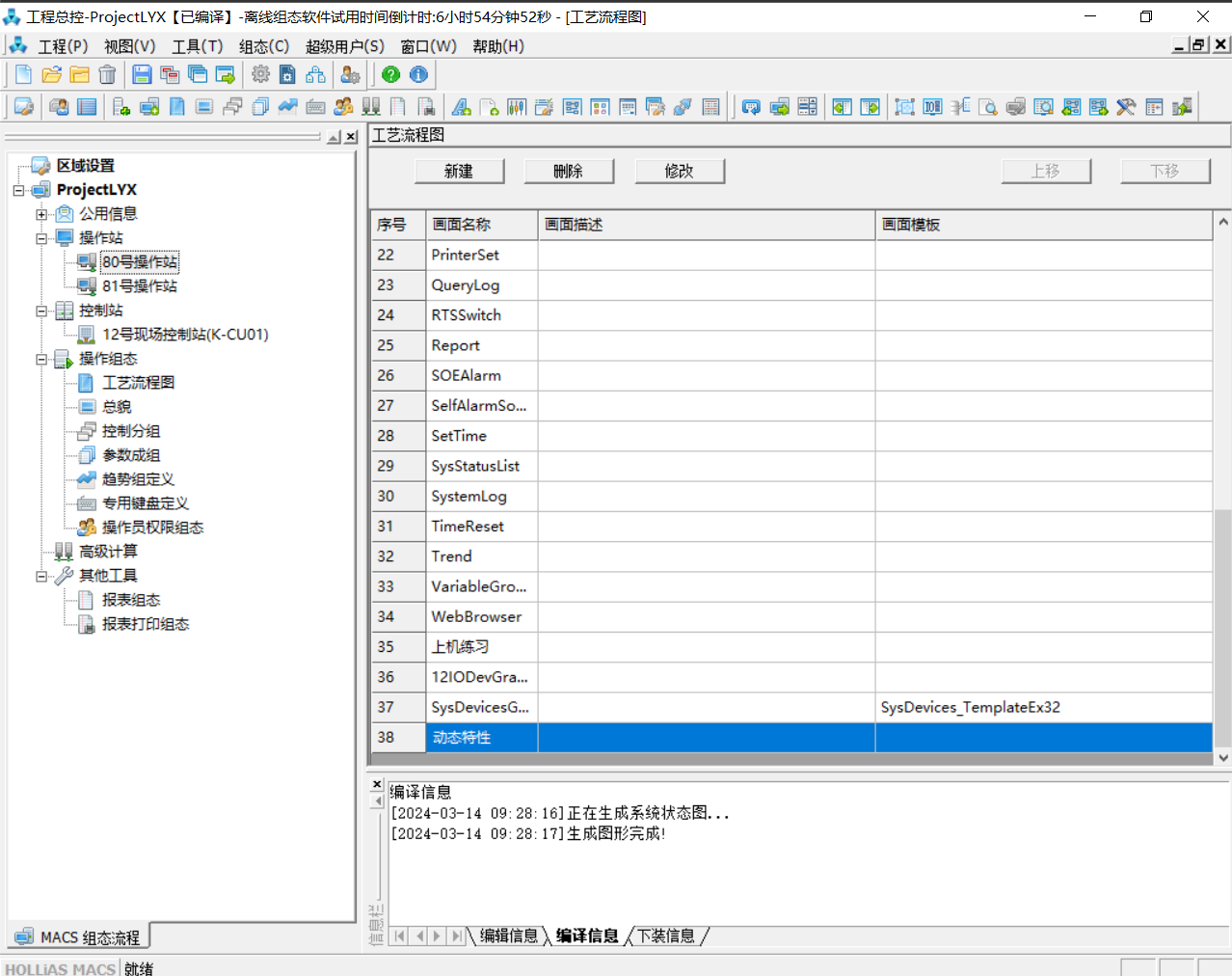
##### 温度与流速的控制逻辑的算法可选择用CFC或者ST语言完成；

1. VAR\_INPUT
2. setpoint : REAL; (\* 温度设定值 \*)
3. process\_value : REAL; (\* 实际罐内温度值 \*)
4. flow\_rate : REAL; (\* 当前液体流速值 \*)
5. END\_VAR
6. VAR\_OUTPUT
7. valve\_command : REAL; (\* 阀门开度指令 \*)
8. alarm\_level : INT; (\* 报警级别，0: 正常，1: 温度过高，2: 温度过低 \*)
9. alarm\_status : BOOL; (\* 报警状态，TRUE: 报警，FALSE: 正常 \*)
10. END\_VAR
11. VAR
12. proportional\_gain : REAL := 10; (\* 比例增益 \*)
13. integral\_time : REAL := 100; (\* 积分时间常数 \*)
14. derivative\_time : REAL := 5; (\* 微分时间常数 \*)
15. error\_integral : REAL := 0; (\* 错误积分 \*)
16. last\_error : REAL := 0; (\* 上一时刻误差 \*)
17. END\_VAR
18. (\* PID控制算法 \*)
19. pid\_algorithm :
20. BEGIN
21. error := setpoint - process\_value;
22. error\_integral := error\_integral + error \* DT;
23. derivative := (error - last\_error) / DT;
24. valve\_command := proportional\_gain \* error +
25. integral\_time \* error\_integral +
26. derivative\_time \* derivative;
27. (\* 报警逻辑 \*)
28. IF process\_value > (setpoint + hysteresis) THEN
29. alarm\_level := 1; (\* 温度过高报警 \*)
30. alarm\_status := TRUE;
31. ELSIF process\_value < (setpoint - hysteresis) THEN
32. alarm\_level := 2; (\* 温度过低报警 \*)
33. alarm\_status := TRUE;
34. ELSE
35. alarm\_level := 0;
36. alarm\_status := FALSE;
37. END\_IF;
38. last\_error := error;
39. END pid\_algorithm;
40. (\* 主程序调用PID算法并输出结果 \*)
41. PROGRAM main
42. BEGIN
43. pid\_algorithm();
44. END\_PROGRAM

##### 基于M6平台通用版本实现一个动态特性。动态特性基于文字图元设计；

1. // 动态特性设计
2. INTERFACE
3. TEXT temperatureDisplay;
4. TEXT flowRateDisplay;
5. TEXT alarmLevelDisplay;
6. TEXT alarmStatusDisplay;
7. END\_INTERFACE
8. // 更新显示逻辑
9. ALGORITHM
10. temperatureDisplay := temperature;
11. flowRateDisplay := flowRate;
12. alarmLevelDisplay := alarmLevel;
13. IF alarmStatus THEN
14. alarmStatusDisplay := '报警';
15. ELSE
16. alarmStatusDisplay := '正常';
17. END\_IF;
18. END\_ALGORITHM





##### （3）结合控制逻辑，用设计出的动态特性显示工艺测点值（如流速、温度）、报警级及报警状态。

1. MODULE TemperatureFlowControlLogic
2. // 控制逻辑代码...
3. END\_MODULE
4. MODULE DynamicFeature {
5. ELEMENT TemperatureDisplay {
6. TYPE: TextDisplay;
7. POSITION: (100, 100);
8. VALUE: Temperature;
9. }
11. ELEMENT FlowRateDisplay {
12. TYPE: TextDisplay;
13. POSITION: (100, 200);
14. VALUE: FlowRate;
15. }
17. ELEMENT AlarmDisplay {
18. TYPE: TextDisplay;
19. POSITION: (100, 300);
20. VALUE: Alarm;
21. }
22. }