

[Site Explorer](#)  
[Close Site Explorer](#)

高级搜索

热搜： 产品信息 选型 3SB3退市声明 3RW常见问题目录 启停和调速 搜索技巧

您的位置：  
[西门子中国](#)  
[工业支持中心](#)  
S7-200 PID

## S7-200 PID

### 关键字

[数目](#) [反作用](#) [参数作用](#)

- PID指令块：[参数地址](#) [采样时间](#) [块使用](#)
- PID向导：[步骤](#) [参数地址](#) [量程设置](#) [控制面板](#) [没有输出](#) [例程](#)
- PID指令块：[无扰切换](#) [调节步骤](#) [手动至稳定](#) [经验数据](#)
- PID向导：[条件](#) [面板自整定](#) [编程自整定](#) [整定失败](#)

### 简介：

在S7-200中PID功能是通过PID指令功能块实现。通过定时（按照采样时间）执行PID功能块，按照PID运算规律，根据当时的给定、反馈、比例 - 积分 - 微分数据，计算出控制量。本文详细介绍了S7-200的PID类型和各参数作用、通过PID指令功能块和PID向导两种方式实现PID编程，同时给出了PID的调节步骤、手自动无扰切换的实现方式，此外还对通过自整定方式进行PID调节给出了控制面板启动和编程启动两种详细方案。如果想要了解更详细的PID算法，请参考《S7-200系统手册》中PID指令部分的相关内容。



### 目录

#### 1 概览

- 1.1 [PID的实现方式及数目](#)
- 1.2 [PID输入/输出支持的信号类型](#)
- 1.3 [PID实现反作用](#)
- 1.4 [PID各参数作用](#)

#### 2 PID指令块实现PID

- 2.1 [介绍PID回路表](#)
- 2.2 [通过PID指令块实现PID](#)

#### 3 向导实现PID

- 3.1 [PID向导设置步骤](#)
- 3.2 [PID向导生成的组件介绍](#)
- 3.3 [PID设定值与过程变量 量程设置](#)
- 3.4 [PID调节控制面板使用](#)

#### 4 PID调节

- 4.1 [PID调节手/自动无扰切换](#)
- 4.2 [PID调节步骤](#)
- 4.3 [手动调节PID至稳定](#)

#### 5 PID自整定

- 5.1 [PID自整定先决条件](#)
- 5.2 [PID自整定参数介绍](#)
- 5.3 [通过PID控制面板启动自整定](#)
- 5.2 [通过编程启动自整定](#)
- 5.3 [PID自整定失败原因](#)

#### 6 扩展阅读

1 概览

PID是闭环控制系统的比例 - 积分 - 微分控制算法。控制器根据设定值（给定）与被控对象的实际值（反馈）的差值，按照PID算法计算出控制器的输出量，控制执行机构去影响被控对象的变化。  
S7-200PID控制是负反馈闭环控制，能够抑制系统闭环内的各种因素所引起的扰动，使反馈跟随给定变化。

1.1 PID的实现方式及数目

1.S7-200中PID功能实现方式有以下三种：

- PID指令块：**通过一个PID回路表交换数据， 只接受0.0 - 1.0之间的实数（实际上就是百分比）作为反馈、给定与控制输出的有效数值。
- PID向导：**方便地完成输入/输出信号转换/标准化处理。PID指令同时会被自动调用。
- 根据PID算法自己编程**（该文档不涉及）

2.S7-200 CPU最多可以支持**8个PID控制回路（8个PID指令功能块）**，根据PID算法自己编程没有具体数目的限制，但是需要考虑PLC的存储空间以及扫描周期等影响。

常见问题

- 同一个程序里既使用PID指令块又使用向导，PID数目怎样计算？**  
使用PID向导时，对应回路的指令块也会调用，所以PID指令块与向导一共支持8个。
- 指令块与向导使用的PID回路号是否可以重复？**  
不可以重复，使用PID向导时，对应回路的指令块也会调用，所以指令块与向导使用的PID回路号不能重复，否则会产生意想不到的结果。
- S7-200控制变频器，在变频器也有PID控制功能时，应当使用谁的PID功能？**  
可以根据具体情况使用。一般来说，如果需要控制的变量直接与变频器直接有关，比如变频水泵控制水压等，可以优先考虑使用变频器的PID功能。

1.2. PID输入/输出支持的信号类型

|    | 信号类型   |
|----|--|
| 输入 | 电压、电流、热电阻、热电偶或经过变换的实际工程量                       |
| 输出 | 模拟量输出用来控制一些需要模拟量给定的设备，如比例阀、变频器等；               |
|    | 数字量输出实际上是控制输出点的通、断状态按照一定的占空比变化，可以控制固态继电器（加热棒等） |

表1.2.1. PID输入/输出支持的信号类型


 S7-200的PID借助相关库文件，可以实现三步控制 例如：使用混合阀门实现水温的三步控制， 具体参考以下链接：  
<https://support.industry.siemens.com/cs/document/18748232?dti=0&lc=zh-CN>

1.3. PID实现反作用

在有些控制中需要PID反作用调节。例如：在夏天控制空调制冷时，若反馈温度（过程值）低于设定温度，需要关阀，减小输出控制（减少冷水流量等），这就是PID反作用调节（在PID正作用中若过程值小于设定值，则需要增大输出控制）。  
若想实现PID反作用调节，需要把PID回路的增益设为负数。对于增益为0的积分或微分控制来说，如果指定积分时间、微分时间为负值，则是反作用回路。

1.4 PID各参数作用

计算机化的PID控制算法有几个关键的参数 $T_s$ （采样时间）， $K_c$ （Gain，增益）， $T_i$ （积分时间常数）， $T_d$ （微分时间常数）。PID参数的取值，以及它们之间的配合，对PID控制是否稳定具有重要的意义：

- 采样时间**  
计算机必须按照一定的时间间隔对反馈进行采样，才能进行PID控制的计算。采样时间就是对反馈进行采样的间隔。短于采样时间间隔的信号变化是不能测量到的。过短的采样时间没有必要，过长的采样间隔显然不能满足扰动变化比较快、或者速度响应要求高的场合。  
 编程时指定的PID控制器采样时间必须与实际的采样时间一致。S7-200中PID的采样时间精度用**定时中断**（PID向导用SMB34）来保证。
- 增益**（Gain，放大系数，比例常数）  
增益与偏差（给定与反馈的差值）的乘积作为控制器输出中的比例部分。提高响应速度，减少误差，但不能消除稳态误差，当比例作用过大时，系统的稳定性下降。
- 积分时间**（Integral Time）  
偏差值恒定时，积分时间决定了控制器输出的变化速率。  
积分时间的长度相当于在阶跃给定下，增益为“1”的时候，输出的变化量与偏差值相等所需要的时间，也就是输出变化到二倍于初始阶跃偏差的时间。  
如果将积分时间设为最大值，则相当于没有积分作用。
- 微分时间**（Derivative Time）  
偏差值发生改变时，微分作用将增加一个尖峰到输出中，随着时间流逝减小。微分时间越长，输出的变化越大。微分使控制对扰动的敏感度增加，也就是偏差的变化率越大，微分控制作用越强。微分相当于对反馈变化趋势的预测性调整。

- 如果将微分时间设置为0就不起作用，控制器将作为PI调节器工作。
- 比例调节**  
提高响应速度，减少误差，但不能消除稳态误差，当比例作用过大时，系统的稳定性下降。（由小到大单独调节）
  - 积分调节**  
消除稳态误差，使系统的动态响应变慢，积分时间越小，积分作用越大，偏差得到的修正越快，过短的积分时间有可能造成不稳定。（将调好的比例增益调整到50%~80%后，由大到小减小积分时间）
  - 微分调节**  
超前调节，能预测误差变化的趋势，提前抑制误差的控制作用，从而避免了被控量的严重超调。可以改善系统的响应速度和稳定性，对噪声干扰有放大作用，对具有滞后性质的被控对象，应加入微分环节。

2 PID指令块实现PID

2.1 介绍PID回路表/T96

在S7-200中PID功能是通过PID指令功能块实现。通过定时（按照采样时间）执行PID指令块，按照PID运算规律，根据当时的给定、反馈、比例 - 积分 - 微分数据，计算出控制量。

PID指令块通过一个PID回路表交换数据，这个表是在V数据存储区中的开辟，长度为80字节（Micro/WIN4.0之前老版本，未增加PID自整定时间回路表长度为36字节）。只接受**0.0 - 1.0之间的实数**（实际上就是百分比）作为反馈、给定与控制输出的有效数值。

| 偏移量            | 域             | 格式   | 类型    | 描述                     |
|----------------|---------------|------|-------|------------------------|
| 0              | 过程变量（PVn）     | REAL | 输入    | 过程变量，必须在0.0~1.0之间      |
| 4              | 设定值（SPn）      | REAL | 输入    | 设定值，必须在0.0~1.0之间       |
| 8              | 输出（Mn）        | REAL | 输入/输出 | 输出值，必须在0.0~1.0之间       |
| 12             | 增益（Kc）        | REAL | 输入    | 增益是比例常数，可正可负           |
| 16             | 采样时间（Ts）      | REAL | 输入    | 采样时间，单位为秒，必须是正数        |
| 20             | 积分时间或复位（Ti）   | REAL | 输入    | 积分时间或复位，单位是分钟          |
| 24             | 微分时间或速率（Td）   | REAL | 输入    | 微分时间或速率，单位为分钟          |
| 28             | 偏差（MX）        | REAL | 输入/输出 | 积分项前项，必须在0.0~1.0之间     |
| 32             | 过程变量前值（PVn-1） | REAL | 输入/输出 | 包含最后一次执行PID指令时存储的过程变量值 |
| 36~79 保留给自整定变量 |               |      |       |                        |

表2.1.1. PID回路表

2.2 通过PID指令块实现PID/35

⚠ 通过指令块实现PID，需要自己编程实现采样，较复杂，容易出错，也不能用PID控制面板进行调节，不建议使用，如果没有特殊要求，尽量使用PID向导。

由于PID指令块只接受**0.0 - 1.0之间的实数**（实际上就是百分比）作为反馈、给定与控制输出的有效数值。因此，必须把外围实际的物理量与PID功能块需要的（或者输出的）数据之间进行转换。这就是所谓输入/输出的转换与标准化处理。

➡ 《S7-200系统手册》上有详细的介绍。

下面简单做一个采样时间为250ms的PID例程：

第一步，主程序内实现250ms的定时中断，如图2.2.1

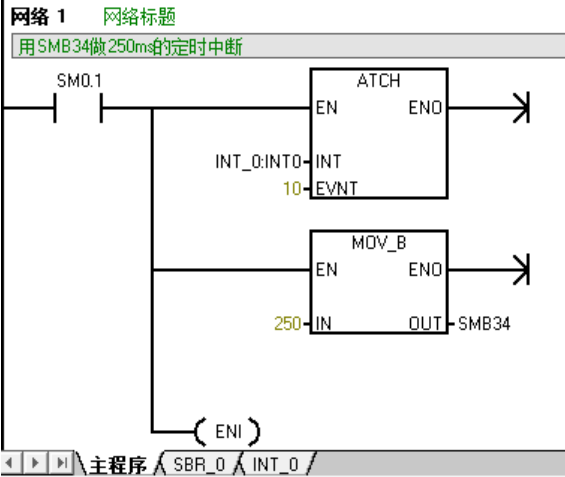


图2.2.1. 定时中断

第二步，在**定时中断**INT\_0内调用PID指令块。

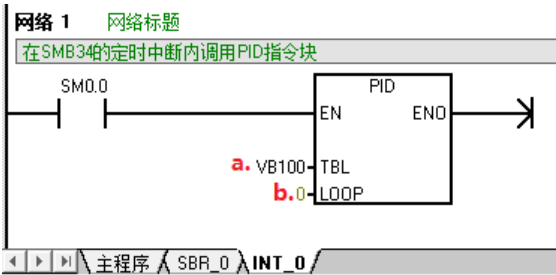


图2.2.2. 调用PID指令块  
图2.2.2中

- a. TBL:控制回路表的起始地址（VB100表示 VB100~VB180地址用于该回路PID，具体地址含义以回路表为准）。
- b. LOOP：PID控制回路号（0~7）。

第三步，根据PID回路表，设置参数，在数据块设置各参数的初始值

过程变量:VD100 0.0  
设定值:VD104 0.0  
输出:VD108 0.0  
增益:VD112 1.0  
采样时间:VD116 0.25  
积分时间:VD120 10.0  
微分时间:VD124 0.0  
PID回路ID:VD136 'PIDA'

//单位 秒  
//单位 分钟  
//单位 分钟  
//PID回路表, 版本A

用户定义1

图2.2.3. 数据块设置参数

第四步，程序下载后根据反馈VD0与给定VD4观察输出VD8（直接通过状态表修改给定反馈即可，真正应用时需要将外围物理量做转换）。

⚡ 模拟量换算可以参考库文件  **量程变换指令库 scale**，使用方法：  
<http://support.automation.siemens.com/CN/view/zh/27043396>

常见问题

❓ PID指令块可以在主程序/子程序里调用吗？

可以，但是不推荐，主程序/子程序的循环时间每个周期都可能不同，不能保证精确的采样，建议用定时中断，例如SMB34/SMB35。

❓ SMB34定时最大255ms，如果采样时间是1S怎样实现？

采样时间是1S，要求PID指令块每隔1S调用一次。可以先做一个250ms的定时中断，然后编程累加判断每4次中断执行一次PID指令即可。

❓ PID指令块怎样实现手动调节？

可以简单地使用“调用/不调用”指令的方式控制自动/手动模式。不调用PID指令时，可以手动给输出地址0.0-1.0之间的实数。

❓ PID指令块实现数字量输出？

- 1.通过PWM指令，将PID输出值转换为所需时间基准的整数，送到PWM的Pulse，控制脉宽（该法简单易用，但是要求输出点只能是Q0.0或Q0.1）。
- 2.自己编程实现类似于PWM的输出。（虽然不限制Q点，但编程较复杂，不建议使用，可以直接考虑用PID向导）。

3 向导实现PID

在Micro/WIN中的命令菜单中选择**工具 > 指令向导**，然后在指令向导窗口中选择PID指令：

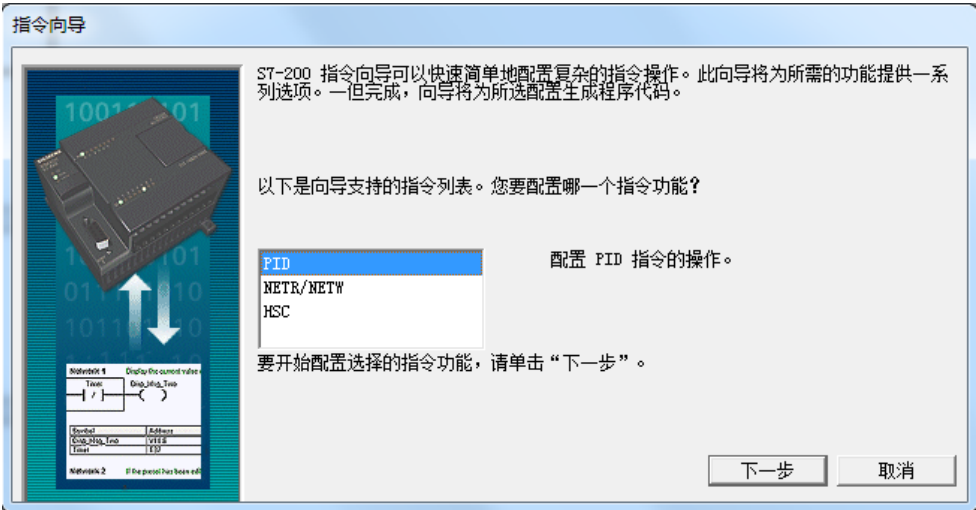


图3.1. 选择PID向导

在使用向导时必须先对项目进行编译，在随后弹出的对话框中选择“是”，确认编译。如果已有的程序中存在错误，或者有没有编  
完的指令，编译不能通过。  
如果你的项目中已经配置了一个PID回路，则向导会指出已经存在的PID回路，并让你选择是配置修改已有的回路，还是配置一个  
新的回路：

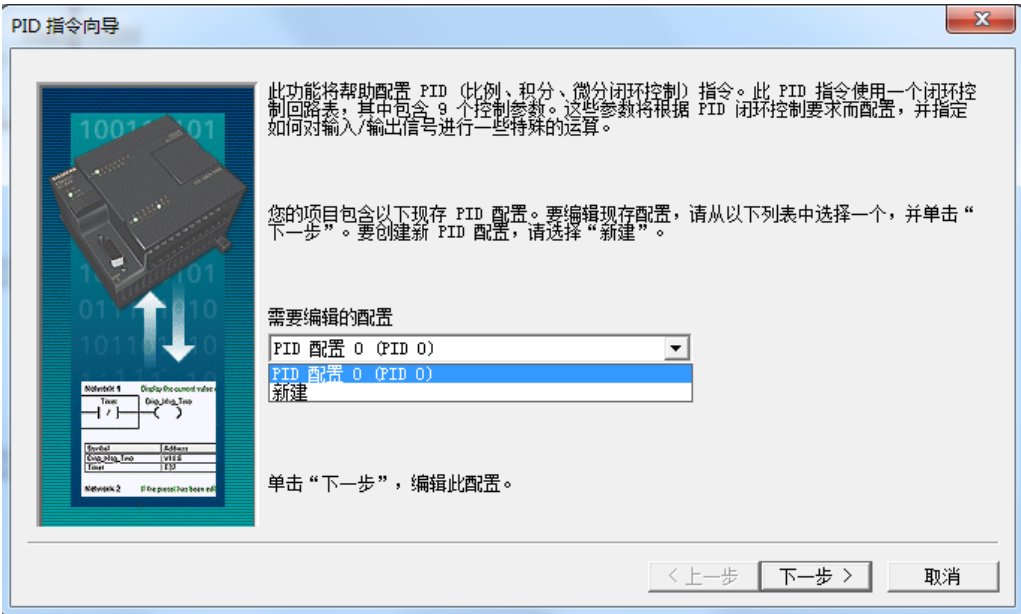


图3.2. 选择需要配置的回路

3.1. PID向导设置步骤

第一步：定义需要配置的PID回路号

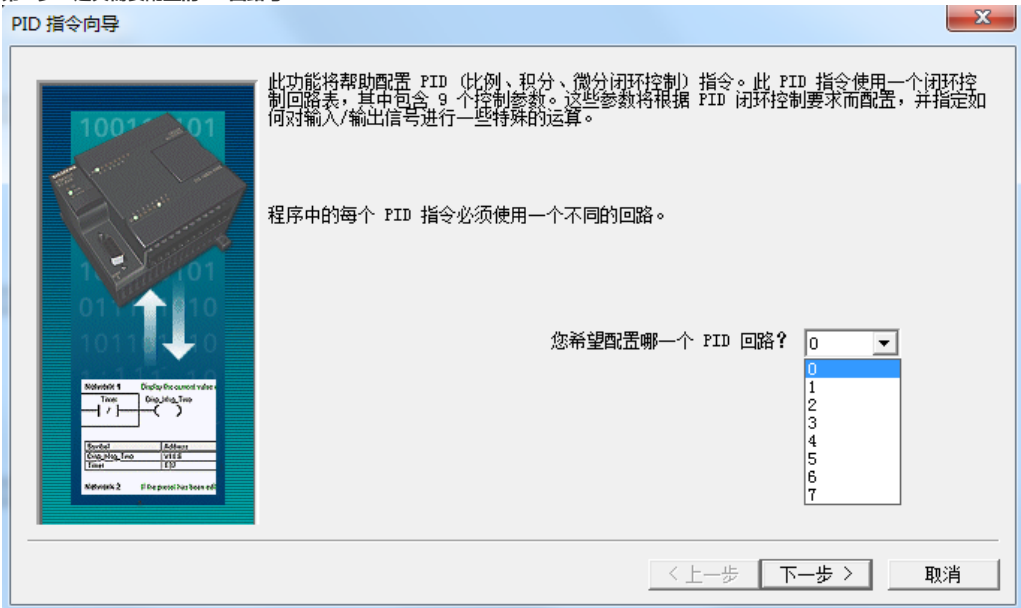


图3.1.1. 选择PID回路号

第二步：设定PID回路参数

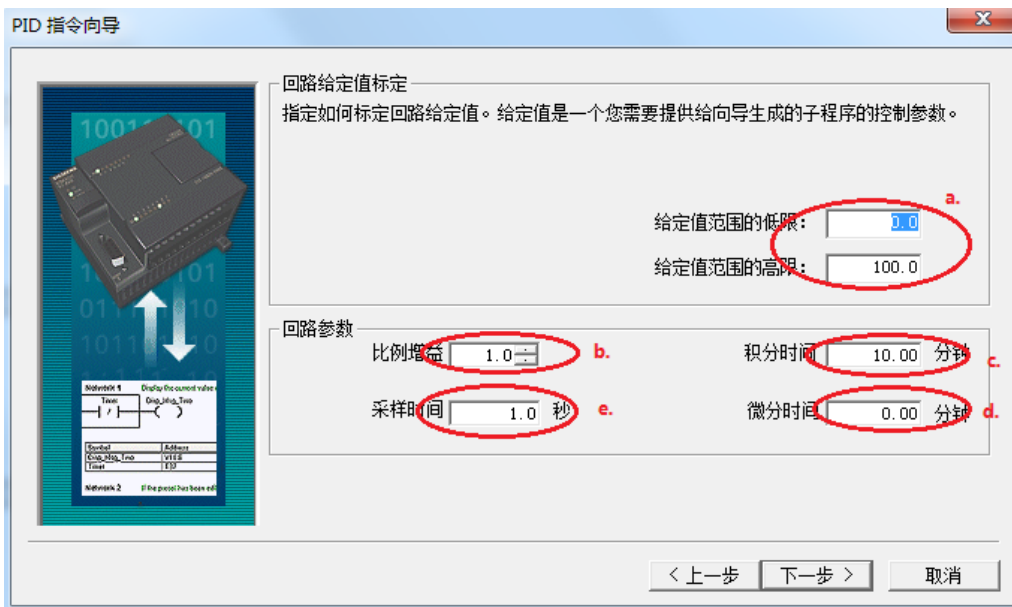


图3.1.2. 设置PID参数

图3.1.2中:

- a. 定义回路设定值 (SP, 即给定) 的范围:

在低限 (Low Range) 和高限 (High Range) 输入域中输入实数, 缺省值为0.0和100.0, 表示给定值的取值范围占过程反馈量程的百分比。💡 这个范围是给定值的取值范围。它也可以用实际的工程单位数值表示。参见:

[PID给定-反馈量程设置](#)

以下定义PID回路参数, 这些参数都应当是实数:

- b. Gain (增益): 即比例常数。
- c. Integral Time (积分时间): 如果不要积分作用, 可以把积分时间设为无穷大: 输入"INF"。
- d. Derivative Time (微分时间): 如果不要微分回路, 可以把微分时间设为0。
- e. Sample Time (采样时间): 是PID控制回路对反馈采样和重新计算输出值的时间间隔。在向导完成后, 若想要修改此数, 则必须返回向导中修改, 不可在程序中或状态表中修改。

⚠ 注意: 关于具体的PID参数值, 每一个项目都不一样, 需要现场调试来定, 没有所谓经验参数。

### 第三步: 设定回路输入输出值

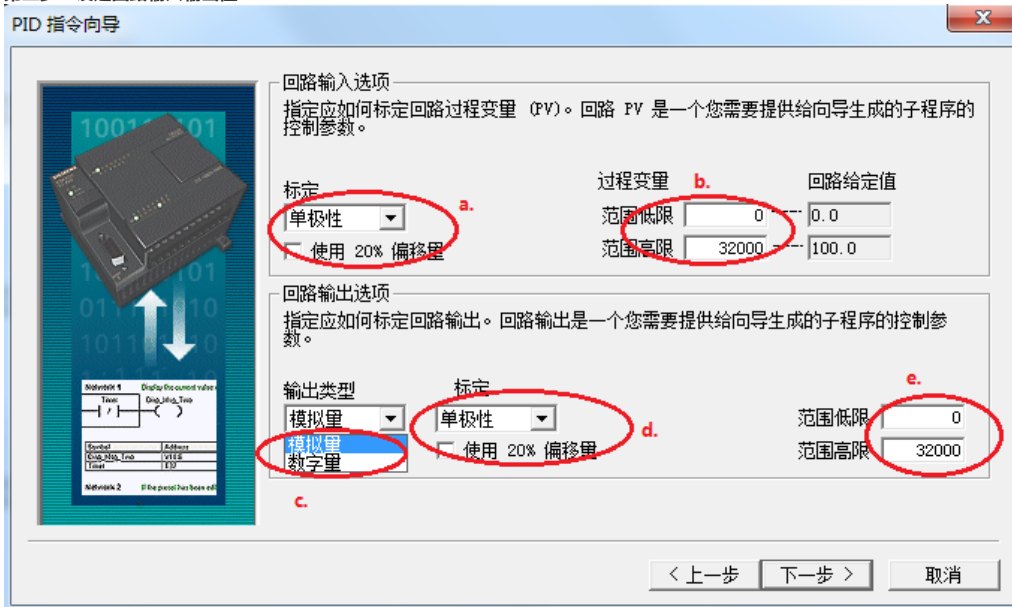


图3.1.3. 设定PID输入输出参数

在图3.1.3中, 首先设定过程变量的范围:

- a. 指定输入类型

- Unipolar: 单极性, 即输入的信号为正, 如0 - 10V或0 - 20mA等
- Bipolar: 双极性, 输入信号在从负到正的范围内变化。如输入信号为±10V、±5V等时选用
- 20% Offset: 使用20%偏移。如果输入为4 - 20mA则选单极性此项, 4mA是0 - 20mA信号的20%, 所以选20% 偏移, 即4mA对应6400, 20mA对应32000

- b. 反馈输入取值范围

- 在a.设置为单极性时, 缺省值为0 - 32000, 对应输入量程范围0 - 10V或0 - 20mA等, 输入信号为正
- 在a.设置为双极性时, 缺省的取值为-32000 - +32000, 对应的输入范围根据量程不同可以是±10V、±5V等
- 在a.使用20%偏移时, 取值范围为6400 - 32000, 不可改变

💡 此反馈输入也可以是工程单位数值, 参见: [PID给定-反馈量程设置](#) 然后定义输出类型



c. Output Type（输出类型）

可以选择模拟量输出或数字量输出。模拟量输出用来控制一些需要模拟量给定的设备，如比例阀、变频器等；数字量输出实际上是控制输出点的通、断状态按照一定的占空比变化，可以控制固态继电器（加热棒等）

d. 选择模拟量则需设定回路输出变量值的范围，可以选择：

- o Unipolar：单极性输出，可为0 - 10V或0 - 20mA等
- o Bipolar：双极性输出，可为正负10V或正负5V等
- o 20% Offset：如果选中20% 偏移，使输出为4 - 20mA

e. 取值范围：

- o d为Unipolar时，缺省值为 0 到 32000
- o d为Bipolar时，取值-32000到32000
- o d为20% Offset时，取值6400 - 32000，不可改变

如果选择了开关量输出，需要设定此占空比的周期。

第四步：设定回路报警选项

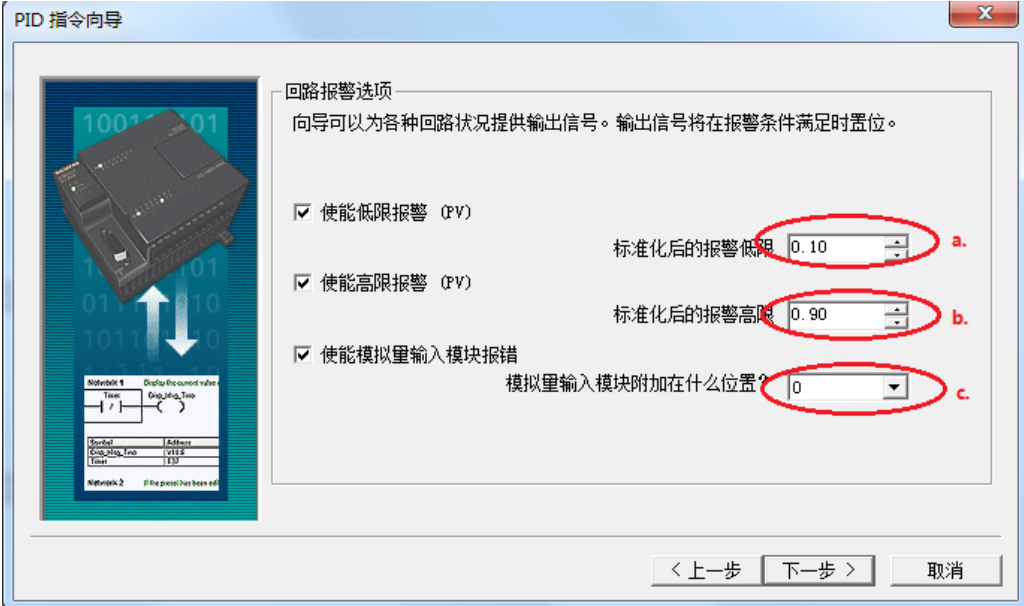


图3.1.4. 设定回路报警限幅值

向导提供了三个输出来反映过程值(PV)的低值报警、高值报警及过程值模拟量模块错误状态。当报警条件满足时，输出置位为1。这些功能在选中了相应的选择框之后起作用。

- a. 使能低值报警并设定过程值(PV)报警的低值，此值为过程值的百分数，缺省值为0.10，即报警的低值为过程值的10%。此值最低可设为0.01，即满量程的1%
- b. 使能高值报警并设定过程值(PV)报警的高值，此值为过程值的百分数，缺省值为0.90，即报警的高值为过程值的90%。此值最高可设为1.00，即满量程的100%
- c. 使能过程值(PV)模拟量模块错误报警并设定模块于CPU连接时所处的模块位置。“0”就是第一个扩展模块的位置

第五步：指定PID运算数据存储区

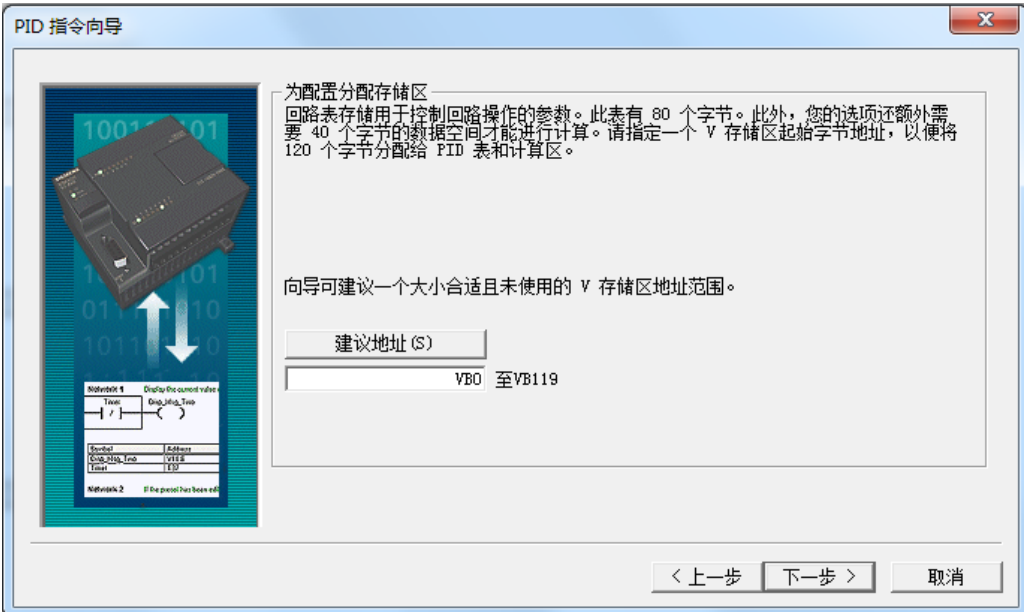



图3.1.5. 分配运算数据存储区

PID指令（功能块）使用了一个120个字节的V区参数来进行控制回路的运算工作；除此之外，PID向导生成的输入/输出量的标准化程序也需要运算数据存储区。需要为它们定义一个起始地址，要保证该地址起始的若干字节在程序的其它地方没有被重复使用。如果点击“建议地址”，则向导将自动为你设定当前程序中没有用过的V区地址。

 自动分配的地址只是在执行PID向导时编译检测到空闲地址。向导将自动为该参数表分配符号名，用户不要再自己为这些参数分配符号名，否则将导致PID控制不执行。

第六步：定义向导所生成的PID初始化子程序和中断程序名及手/自动模式

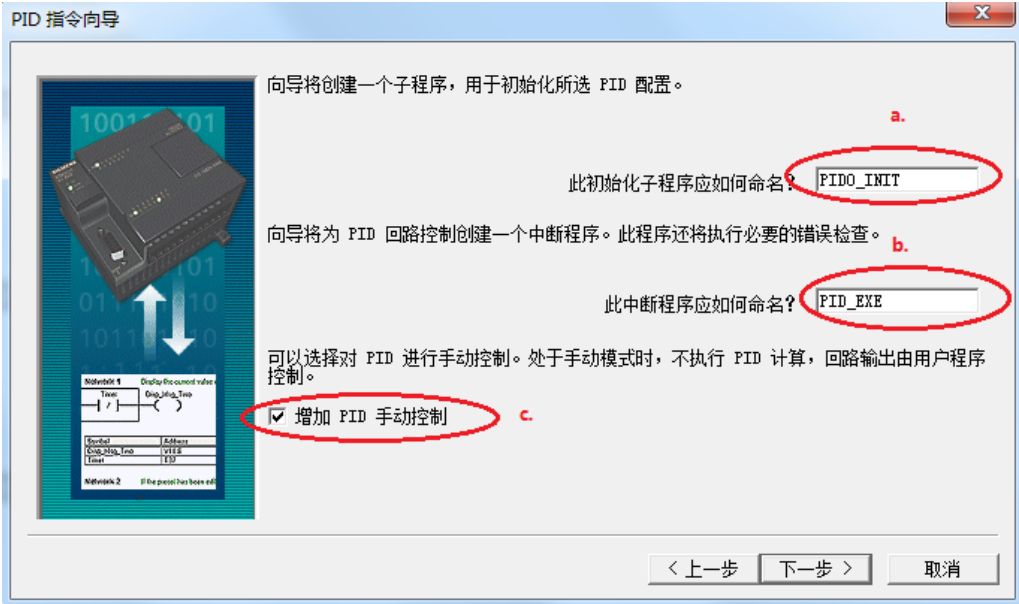


图3.1.6. 指定子程序、中断服务程序名和选择手动控制

向导已经为初始化子程序和中断子程序定义了缺省名，你也可以修改成自己起的名字。

- a. 指定PID初始化子程序的名字。
- b. 指定PID中断子程序的名字

 注意：

- 1. 如果你的项目中已经存在一个PID配置，则中断程序名为只读，不可更改。因为一个项目中所有PID共用一个中断程序，它的名字不会被任何新的PID所更改。
- 2. PID向导中断用的是SMB34定时中断，在用户使用了PID向导后，注意在其它编程时不要再用此中断，也不要向SMB34中写入新的数值，否则PID将停止工作。
- c. 此处可以选择添加PID 手动控制模式。在PID手动控制模式下，回路输出由手动输出设定控制，此时需要写入手动控制输出参数一个0.0 - 1.0的实数，代表输出的0% - 100%而不是直接去改变输出值。

➡ [PID调节手/自动无扰动切换](#)

第七步：生成PID子程序、中断程序及符号表等

一旦点击完成按钮，将在你的项目中生成上述PID子程序、中断程序及符号表等。

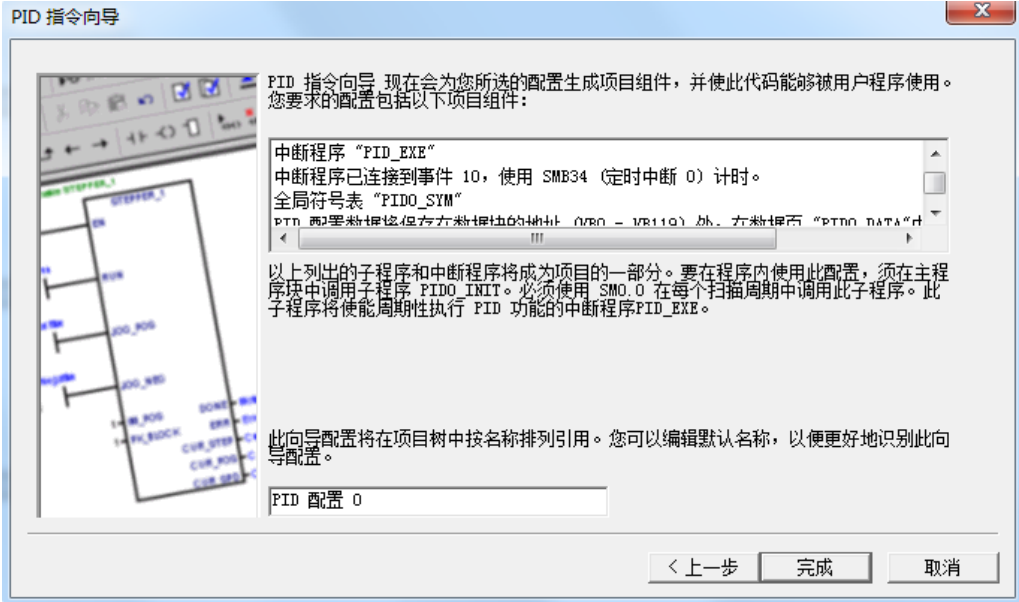


图3.1.7. 生成PID子程序、中断程序和符号表等

第八步：配置完PID向导，需要在程序中调用向导生成的PID子程序（如下图）

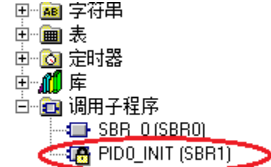


图3.1.8. PID子程序



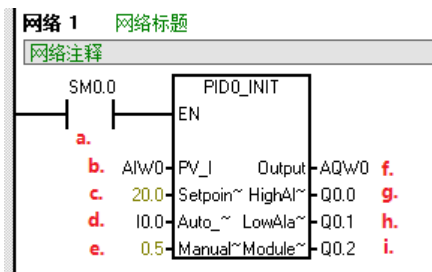


图3.1.9. 调用PID子程序

在用户程序中调用PID子程序时，可在指令树的Program Block（程序块）中用鼠标双击由向导生成的PID子程序，在局部变量表中，可以看到有关形式参数的解释和取值范围。

- 必须用SM0.0来使能 PIDx\_INIT 子程序，SM0.0 后不能串联任何其他条件，而且也不能有越过它的跳转；如果在子程序中调用 PIDx\_INIT 子程序，则调用它的子程序也必须仅使用 SM0.0 调用，以保证它的正常运行
- 此处输入过程值（反馈）的模拟量输入地址
- 此处输入设定值变量地址（VDxx），或者直接输入设定值常数，根据向导中的设定0.0 - 100.0，此处应输入一个0.0 - 100.0的实数，例：若输入20，即为过程值的20%，假设过程值AIW0是量程为0 - 200度的温度值，则此处的设定值20代表40度（即200度的20%）；如果在向导中设定给定范围为0.0 - 200.0，则此处的20相当于20度
- 此处用I0.0控制PID的手/自动方式，当I0.0为1时，为自动，经过PID运算从AQW0输出；当I0.0为0时，PID将停止计算，AQW0输出为ManualOutput（VD4）中的设定值，此时不要另外编程或直接给AQW0赋值。若在向导中没有选择PID手动功能，则此项不会出现
- 定义PID手动状态下的输出，从AQW0输出一个满值范围内对应此值的输出量。此处可输入手动设定值的变量地址（VDxx），或直接输入实数。数值范围为0.0-1.0之间的一个实数，代表输出范围的百分比。例：如输入0.5，则设定为输出的50%。若在向导中没有选择PID手动功能，则此项不会出现
- 此处键入控制量的输出地址
- 当高报警条件满足时，相应的输出置位为1，若在向导中没有使能高报警功能，则此项将不会出现
- 当低报警条件满足时，相应的输出置位为1，若在向导中没有使能低报警功能，则此项将不会出现
- 当模块出错时，相应的输出置位为1，若在向导中没有使能模块错误报警功能，则此项将不会出现

💡 调用PID子程序时，不用考虑中断程序。子程序会自动初始化相关的定时中断处理事项，然后中断程序会自动执行。

### 第九步：实际运行并调试PID参数

没有一个PID项目的参数不需要修改而能直接运行，因此需要在实际运行时调试PID参数。具体调节过程可以参考 [PID调节](#)

为了更好地理解 PID向导的编程，可参考下面的例程。

通过上述向导步骤实现PID，为求程序简单可读，未设置回路报警选项，仅简单常用配置，具体参见例程，关注程序注释以及符号表内容可帮助更快理解程序。



例程： [S7-200 PID](#)

⚠ 注意：此指令程序的作者和拥有者对于该软件的功能性和兼容性不负任何责任。使用该软件的风险完全由用户自行承担。由于它是免费的，所以不提供任何担保，错误纠正和热线支持，用户不必为此联系西门子技术支持与服务部门

### 3.2. PID向导生成的组件介绍

PID向导完成后，自动生成的除PID子程序（上一节已经介绍）以外还有数据块、符号表以及中断程序等组件。

#### 1.数据块

```
// 下列内容由 S7-200 的 PID 指令向导生成。
// PID 0 的参数表。
//
VD0      0.0      //过程变量
VD4      0.0      //回路给定值
VD8      0.0      //回路输出计算值
VD12     1.0      //回路增益
VD16     1.0      //采样时间
VD20     10.0     //积分时间
VD24     0.0      //微分时间
VD28     0.0      //积分项前值
VD32     0.0      //上次运算时存储的过程变量前值。
VB36     'PIDA'   //扩展回路表标志
VB40     16#00    //算法控制字节
VB41     16#00    //算法状态字节
VB42     16#00    //算法结果字节
VB43     16#03    //算法配置字节
VD44     0.08     //从'高级'按钮或默认设置的偏差值
VD48     0.02     //从'高级'按钮或默认设置的滞后死区值
VD52     0.1      //从'高级'按钮或默认设置的起始输出步长值
VD56     7200.0   //从'高级'按钮或默认设置的看门狗超时值
VD60     0.0      //由自动调节算法决定的增益值
VD64     0.0      //由自动调节算法决定的积分时间值
VD68     0.0      //由自动调节算法决定的微分时间值
VD72     0.0      //选择自动计算选项时由算法计算的偏差值
VD76     0.0      //选择自动计算选项时由算法计算的滞后死区值
VD112    0.9      //报警高限
VD116    0.1      //报警低限
```

图3.2.1. PID向导生成数据块

完成PID Wizard配置后，会为每个PID回路生成一个数据块PIDx\_DATA(x=0-7)。图中可以看出数据块的内容实际就是PID回路表以及报警的高/低限值。实际也说明PID向导是基于PID指令块编程经过调整后呈现给用户。

#### 2.符号表

|    | 符号              | 地址    | 注释                     |
|----|-----------------|-------|------------------------|
| 1  | PID0_Low_Alarm  | VD116 | 报警低限                   |
| 2  | PID0_High_Alarm | VD112 | 报警高限                   |
| 3  | PID0_Mode       | V82.0 |                        |
| 4  | PID0_WS         | V882  |                        |
| 5  | PID0_D_Counter  | VW80  |                        |
| 6  | PID0_D_Time     | VD24  | 微分时间                   |
| 7  | PID0_I_Time     | VD20  | 积分时间                   |
| 8  | PID0_SampleTime | VD16  | 采样时间 (要修改请重新运行 PID 向导) |
| 9  | PID0_Gain       | VD12  | 回路增益                   |
| 10 | PID0_Output     | VD8   | 标准化的回路输出计算值            |
| 11 | PID0_SP         | VD4   | 标准化的过程给定值              |
| 12 | PID0_PV         | VD0   | 标准化的过程变量               |
| 13 | PID0_Table      | V80   | PID 0 的回路表起始地址         |

用户定义1 / POU 符号 / PID0\_SYM /

图3.2.2. PID向导生成符号表

完成PID Wizard配置后，会为每个PID回路生成一个数据块PIDx\_SYM(x=0-7)。图中可以看出符号表内容也是回路表相关参数。

3.中断程序 PID向导生成的中断程序PID\_EXE自动加密，相关功能已经附在向导生成的子程序内，这里不涉及。

查看Data Block（数据块），以及Symbol Table（符号表）相应的PID符号标签的内容，可以找到包括PID核心指令所用的控制回路表，包括比例系数、积分时间等等。将此表的地址复制到Status Chart（状态表）中，可以在监控模式下在线修改PID参数，而不必停机再次做配置。

参数调试合适后，用户可以在数据块中写入，也可以再做一次向导，或者编程向相应的数据区传送参数。

常见问题

做完PID向导后，如何知道向导中设定值，过程值及PID等参数所用的地址？

- 做完PID向导后可在Symbol Table（符号表）中，查看PID向导所生成的符号表(上例中为PID0\_SYM)，可看到各参数所用的详细地址，及数值范围。
- 在Data Block（数据块）中，查看PID指令回路表的相关参数。

怎样在上位机或触摸屏上修改PID参数？

图3.2.1/3.2.2中均能查到对应参数地址，在上位机获触摸屏中修改即可，注意数据类型。

数据块内设定值与过程变量以及输出的范围是多少？

数据块内的变量范围完全与回路表内变量范围一致，即都是0.0-1.0之间的浮点数，如果向导里这些变量是模拟量，或者实际工程量，向导会自动换算成0.0-1.0的标准化数值。通过这些变量也可以确认向导内设置的设定值与过程变量量程是否正确。

3.3. PID设定值与过程变量 量程设置

下图是PID向导生成的子程序：

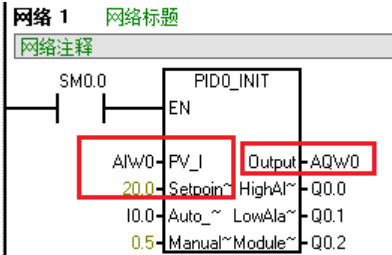


图3.3.1. 调用PID子程序

各参数在3.1章节中已经介绍，这里对以下几个参数着重解释：

- PV\_I：过程反馈参数值的入口
- Setpoint\_R：给定参数值的入口
- Output：PID调节器的输出值

在这里，给定、反馈的入口参数不是PID指令功能块所需要的0.0 - 1.0之间的实数，而可以是实际的反馈地址，或是其他变量。例如，PV\_I可以是模拟量输入地址AIW0，也可以是存储器地址VW100等；Setpoint则往往来自V变量存储区，这样可以从人机操作界面（HMI）设备输入给定值。

注意：

对于PID控制系统来说，必须保证给定与过程反馈的一致性：

- 给定与反馈（过程变量）的物理意义一致  
这取决于被控制的对象，如果是压力，则给定也必须对应于压力值；如果是温度，则给定也必须对应于温度。
- 给定与反馈（过程变量）的数值范围对应  
如果给定直接是摄氏温度值，则反馈必须是对应的摄氏温度值；如果反馈直接使用模拟量输入的对应数值，则给定也必须向反馈的数值范围换算。如果给定与反馈的换算有特定的比例关系也可以。如给定也可以表示为以反馈的数值范围的百分比数值。

给定与反馈的数值具体是什么数值，其取值范围究竟如何，完全取决于我们在使用“PID向导”编程时指定的给定与反馈的数值范围。其中，反馈量的数值范围不能随便自己定义，而要取决于具体应用的模拟量输入模块。

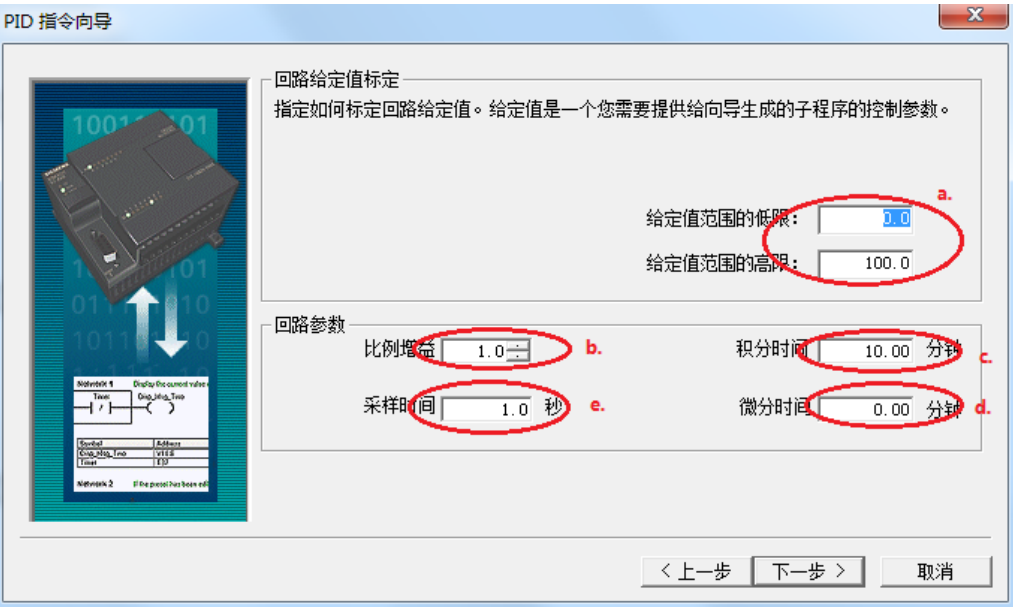


图3.3.2. 在图中a.处设置给定范围

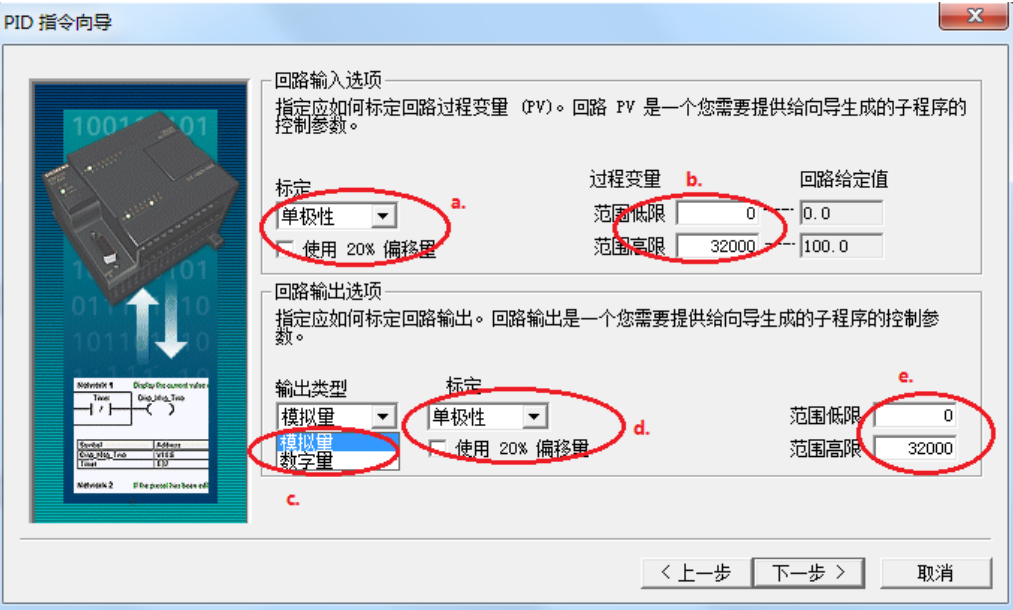


图3.3.3. 在图中b.处设置反馈范围

实例

1. 假定一个PID控制系统的控制对象是压力，反馈元件的测量范围为0 - 16MPa。反馈器件的信号经过变换，以0 - 20mA（或4 - 20mA）电流信号的形式输入到EM231模拟量输入模块中。据此，我们可以按下表设置给定、反馈（过程变量）的范围。

|    | 反馈（单极性） |                | 给定                    |          |
|----|---------|----------------|-----------------------|----------|
|    | 实际物理量   | 模拟量输入数值        | 百分比形式（占0 - 16MPa的百分比） | 物理工程单位形式 |
| 高限 | 16 MPa  | 32000          | 100.0                 | n×16.0   |
| 低限 | 0 MPa   | 0（0 - 20mA）    | 0.0                   | 0.0      |
|    |         | 6400（4 - 20mA） |                       |          |

表3.3.1.反馈/给定设置  
n 为比例系数，为了精度高些可以设置n=10等等

2. 一个温度控制的PID系统，温度值直接由热电偶测量，输入到EM231 TC（热电偶）模块转换为温度值。热电偶为J型，其测量范围为 -150.0℃ - 1200.0℃。则可按下设置给定的范围。

|    | 反馈（双极性） |         | 给定                          |          |
|----|---------|---------|-----------------------------|----------|
|    | 实际物理量   | 模拟量输入数值 | 百分比形式（占-150℃ - 1200.0℃的百分比） | 物理工程单位形式 |
| 高限 | 1200.0℃ | 12000   | 100.0                       | 1200     |
| 低限 | -150.0℃ | -1500   | 0.0                         | -150     |

表3.3.2.反馈/给定设置

在表3.3.2的例子中，给定和反馈可以按照如下方法设置

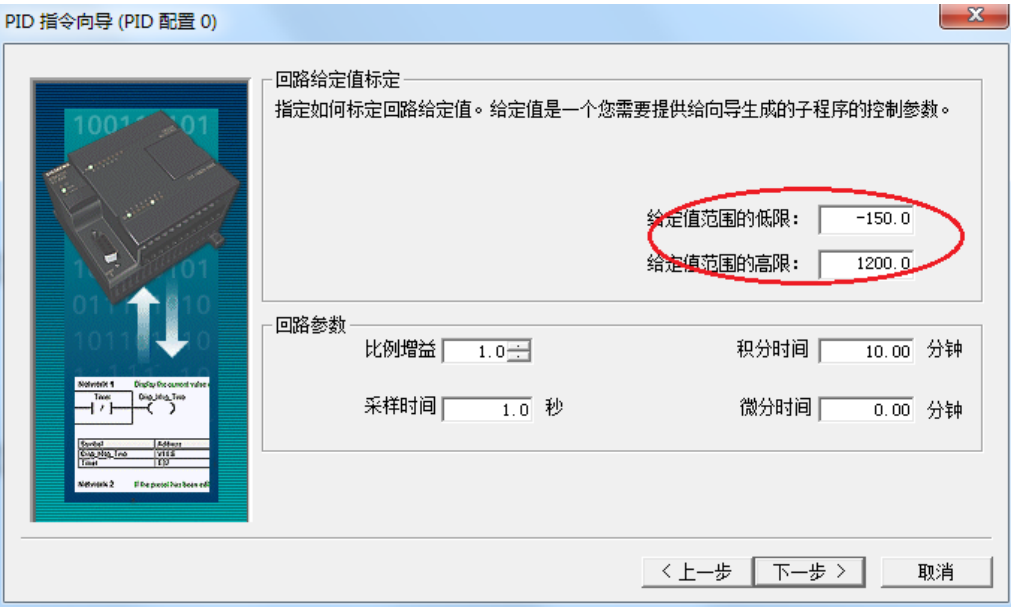


图3.3.4. 给定范围设置

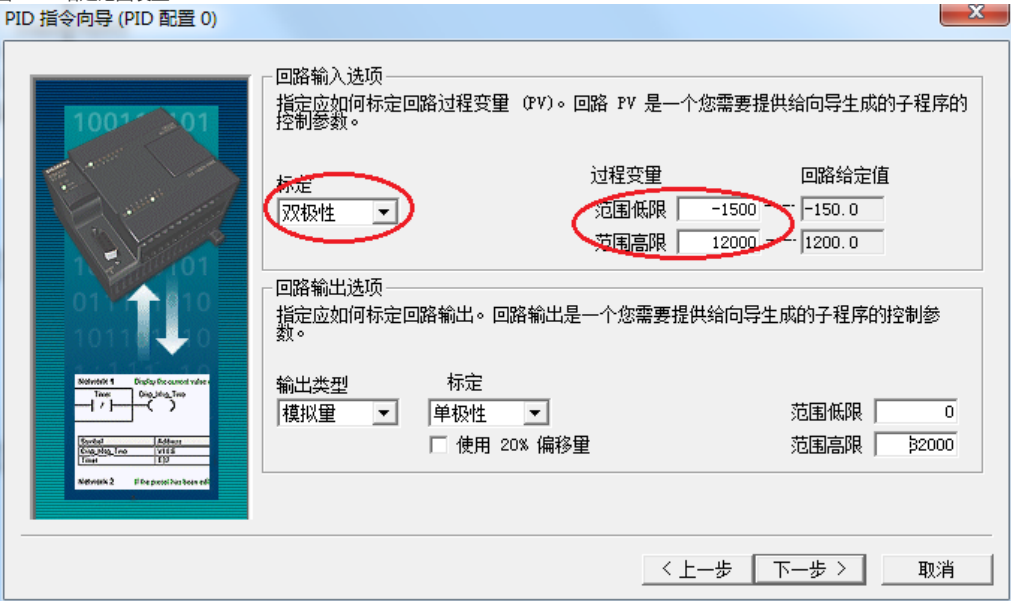


图3.3.5. 反馈范围设置

3.4. PID调节控制面板使用

STEP 7-Micro/WIN V4.0中提供了一个PID调节控制面板，具有图形化的给定、反馈、调节器输出波形显示，可以用于手动调试PID参数。另外从面板中还可以启动、停止自整定功能，对于没有“自整定PID”功能的老版CPU，也能实现PID手动调节。

⚠ 要使用PID调节控制面板，PID编程必须使用PID向导完成。

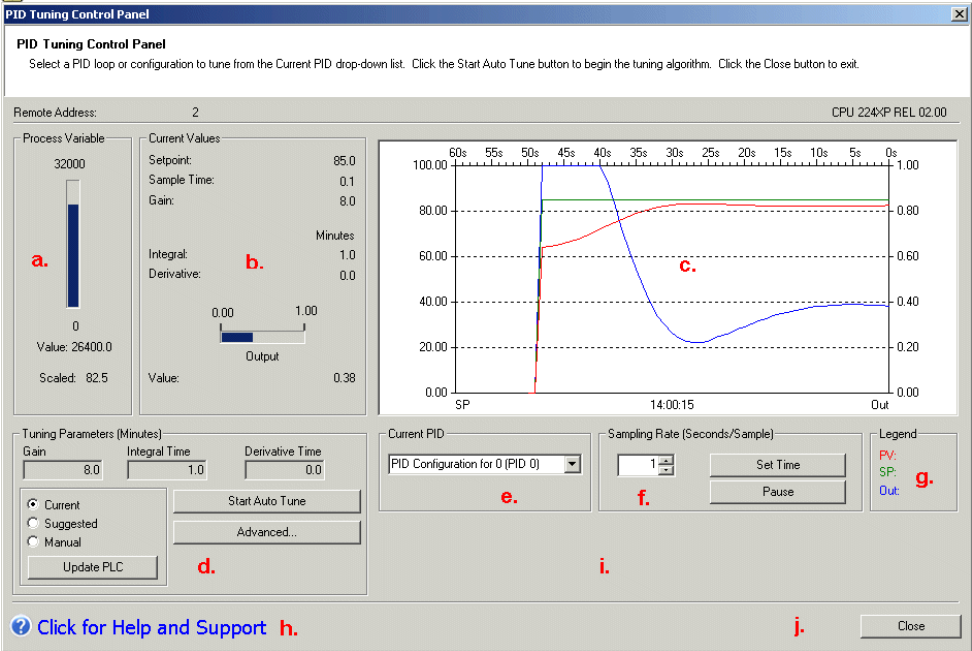


图3.4.1. PID调节控制面板

- 在图3.4.1中：
- a. 过程值指示  
显示过程变量的值及其棒图
  - b. 当前的输出值指示  
显示当前使用的设定值、采样时间、PID 参数值及显示当前的输出值和棒图
  - c. 可显示过程值、设定值及输出值的PID趋势图

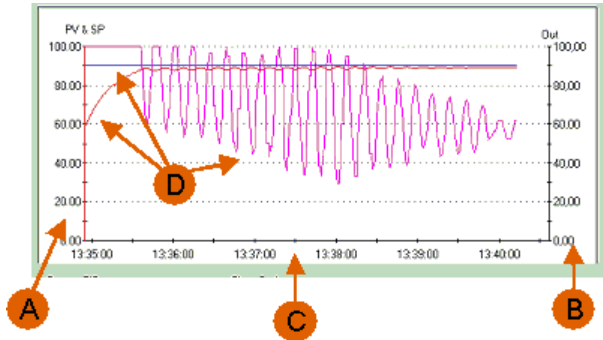


图3.4.2. 图形显示区

图中：

- A. 过程变量和设定值的取值范围及刻度
  - B. PID输出的取值范围及刻度
  - C. 实际PC时间
  - D. 以不同颜色表示的设定值、过程变量及输出的趋势图
- d. 调节参数
- 这里你可以：
- 选择PID参数的显示：当前参数（Current）、推荐参数（Suggested）、手动输入（Manual）
  - 在Manual模式下，可改变PID参数，并按Update PLC按钮来更新PLC中的参数
  - 启动PID自整定功能
  - 选择 **Advanced (高级) 按钮** 进入高级参数设定
- e. 当前的PID回路号
- 这里你可以选择需要监视或自整定的PID回路
- f. 时间选项设定
- 这里你可以设定趋势图的时基，时基以分为单位
- g. 图例颜色
- 这里你可以看到趋势图中不同的颜色代表不同的值的趋势
- h. 帮助按钮
- i. PID信息显示窗口
- j. 关闭PID调节面板

常见问题

**使用PID调节面板手动调整增益、积分时间、微分时间参数，修改的数值能否进入到PLC？**

可以,但是需要在参数设置完以后手动点击调节面板的Update PLC（更新PLC）按钮，来更新PLC中的参数。

**PID已经调整合适，如何正式确定参数？**

可以在Data Block（数据块）中直接写入参数。

**做完PID向导后，能否查看PID生成的子程序，中断程序？**

PID向导生成的子程序，中断程序用户是无法看到的，也不能对其进行修改。没有密码能够打开这些子程序，一般的应用也没有必要打开查看。

**PID向导生成的程序为何不执行或没有输出？**

- 必须保证用SM0.0无条件调用PID0\_INIT程序
- 在程序的其它部分不要再使用SMB34定时中断，也不要对SMB34赋值
- 确认当前工作状态：手动还是自动

**如何根据工艺要求有选择地投入PID功能？**

可使用"手动/自动"切换的功能。PID向导生成的PID功能块只能使用SM0.0的条件调用。

4 PID调节

PID控制的效果就是看反馈（也就是控制对象）是否跟随设定值（给定），是否响应快速、稳定，是否能够抑制闭环中的各种扰动而回复稳定。要衡量PID参数是否合适，必须能够连续观察反馈对于给定变化的响应曲线；而实际上PID的参数也是通过观察反馈波形而调试的。因此，没有能够观察反馈的连续变化波形曲线的有效手段，就谈不上调试PID参数。观察反馈量的连续波形，可以使用带慢扫描记忆功能的示波器（如数字示波器），波形记录仪，或者在PC机上做的趋势曲线监控画面等。

4.1. PID调节手/自动无扰动切换



有些工程项目中可能需要根据工艺要求在不同的时刻投入、或者退出 PID 自动控制；退出 PID 自动控制时，控制器的输出部分可以由操作人员直接手动控制。这就是所谓的 PID 手动/自动切换。

- PID 控制处于**自动方式**时，PID 控制器（S7-200 中的 PID 调节功能）会按照 PID 算法，自动通过输出的作用使过程反馈值跟随给定值变化，并保持稳定。这是一个自动的闭环控制系统。操作人员可以根据现场工艺的要求，改变给定（即设定值）的值。
- PID 控制处于**手动方式**时，PID 控制器不再起自动计算的作用。这时，控制回路的输出是由操作人员手动控制、调整，由操作人员观察现场的控制效果，从而构成人工闭环控制。

有些控制系统的执行机构不能承受较大的冲击，这就要求在进行 PID 自动/手动切换时，保持控制**输出**的稳定。这就是要求**无扰动切换**。

为了达到 PID 自动/手动控制的无扰动切换，需要在编程时注意一些相关事项。下面分别就直接使用 PID 指令编程，和使用 PID 向导编程两种情况作一介绍。

1.直接使用 PID 指令编程时的 PID 自动/手动无扰切换

直接使用 PID 指令块编写 PID 控制程序时，可以简单地使用“调用/不调用”指令的方式控制自动/手动模式。因为 PID 指令本身已经具有实现无扰动切换的能力，此时在 PID 指令控制环节之外编程没有多大必要。

PID 指令的 EN 输入端使能（为“1”）时，我们认为自动控制模式；EN 输入端未使能（为“0”）时，我们认为手动控制模式。

PID 指令本身有一个“能流历史状态位”，以记录指令的状态切换。在 EN 端从“0”变为“1”时，PID 指令认为这是从“手动”模式向“自动”模式切换。PID 指令此时会自动执行一系列动作，以配合无扰动切换：

- 使设定值（SPn）= 当前过程反馈变量（PVn）
- 设置上次采样过程变量（PVn-1）= 当前过程反馈变量（PVn）
- 设置积分偏差和（或所谓积分前项）（Mx）= 当前输出值（Mn）

使设定值等于当前反馈值可以避免出现偏差，使之不存在调整的要求；当然如果有工艺要求，也可以后续调整设定值。其他的动作都是为了使 PID 在后续的操作中不改变输出的值。

在编程时要注意：

- 从自动模式向手动模式切换时，PID 指令的 EN 端不再有能流，计算停止，输出值 Mn 不再变化。此时如果需要操作人员人工观察控制的结果，手动控制输出量，就可以通过用户程序直接改变回路表中的输出值存储单元内容（见数据块或系统手册的相关部分内容）。如果有必要，操作人员的操作可能要进行一些标准化换算。
- 为保证从手动模式向自动模式的切换无扰动，需要在手动控制时，或在切换过程中，禁止对 PID 回路表中设定值的更新，以便切换时 PID 指令用当前过程反馈值替代设定值。切换完成后，操作人员可以调整设定值。

2.使用 PID 向导编程时的 PID 自动/手动无扰切换

使用 PID 指令向导编程时，指令向导会自动调用 PID 指令，并且编写外围的控制变量标准化换算、定时采样等功能。用户在使用 PID 指令向导时，需要在用户程序中用 SM0.0 调用指令向导生成的子程序（如 PIDx\_INIT 子程序）。PID 向导可以生成带自动/手动切换功能的子程序，这个子程序使用一个数字量点为“1”、“0”的状态来控制是否投入 PID 自动控制。

⚠ 到目前为止（STEP 7-Micro/WIN V4.0 SP5），使用 PID 向导生成的子程序时，由于用户程序不能直接使用 PID 指令，它的无扰切换能力因为隔了外壳子程序，所以受到了局限。如果对无扰切换要求比较严格，需要另外编一些程序加以处理。

考察如下 PID 控制子程序。

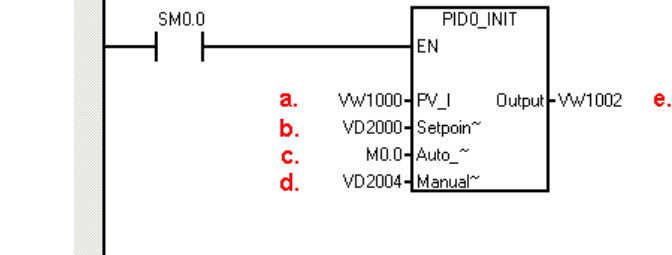


图 4.1.1. PID 向导生成的指令图中：

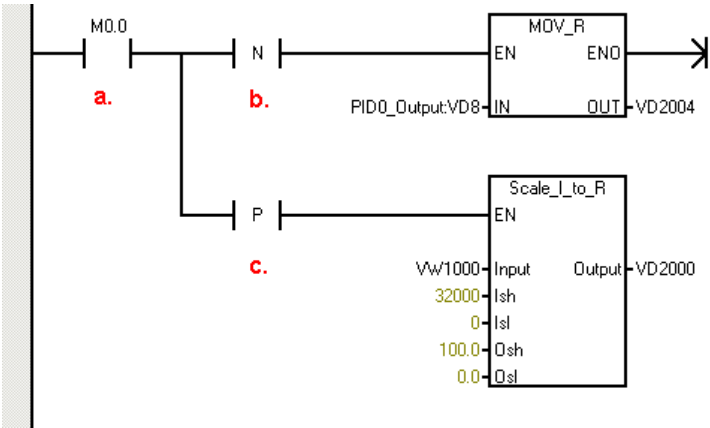
- a. 过程反馈量
- b. 设定值，实数
- c. 自动/手动控制，“1”=自动，“0”=手动
- d. 手动控制输出值，0.0 - 1.0 之间的一个实数
- e. PID 控制输出值

要实现无扰动切换，必须：

- 在从自动向手动切换时，使手动输出值（VD2004）等于当前的实际控制输出值；
- 在从手动向自动切换使，使设定值相当于当前的过程反馈值。

为此，可编写类似下图所示的程序，放在 PID 控制子程序之前：





4.1.2. 无扰切换处理程序  
图中：

- a. 自动/手动切换控制点
- b. 从自动向手动切换时，使手动输出值等于实际当前值
- c. 从手动向自动切换时，把当前反馈量换算为相应的给定值

💡 上述程序中的 Scale\_I\_to\_R 就是 [量程变换指令库](#) 中的子程序。这是为了解决过程反馈与设定值之间的换算问题。用户也可以自己编程换算，或者根据反馈与给定的取值范围决定是否需要进行换算。

⚠️ 此段程序适用于 STEP 7-Micro/WIN V4.0 SP5 及以前版本，仅供参考，如果在实际项目中使用，上述程序未必一定适用。用户需要根据实际工艺决定自己的编程思路。

4.2. PID调节步骤简介

建议PID参数调节步骤：

- (1) 前提条件：反馈信号是否稳定，执行机构是否正常以及控制器的正反作用。（确保PID在自动模式下）
- (2) 积分时间设置为无穷大INF（或9999.9），此时积分作用近似为0；将微分时间设置为0.0，此时微分作用为0。然后开始调节比例作用，逐步增大比例增益
- (3) 当过程变量达到给定值且在给定值上下波动，将调好的比例系数调整到50%~80%后，由大到小减小积分时间，直到过程值与设定值相等或无限接近

PID调节有很多种方法，以上仅是建议步骤，也并未考虑微分作用，客户依据实际情况灵活调节，同时可以参考 [反馈与给定的曲线图](#)

用户经常会遇到这样的问题：尝试了很多组PID参数，都无法满足控制器的要求，此时需要考虑PID的**采样时间**是否适合当前系统。采样时间就是对反馈进行采样的间隔。短于采样时间间隔的信号变化是不能测量到的。采样时间过短，两次实测值的变化量太小，也不合适，而且增加PLC的运算负担；采样间隔过长，将会引起有用信号的丢失，使系统品质变差，不能满足扰动变化比较快、或者速度响应要求高的场合。除此以外，也有可能是系统自身的问题，无法调节到稳定，例如，不规律的干扰，或者反馈信号不稳定。

4.3. 手动调节PID至稳定

问题与解决方法：

- 1. PID输出总是输出很大的值，并在这一区间内调节变化

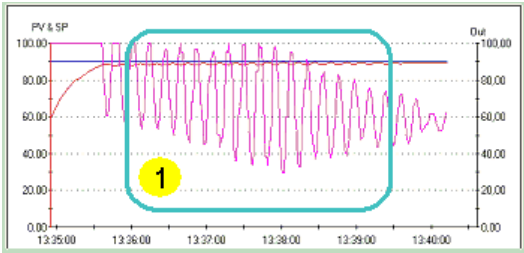


图 4.3.1

产生原因：

- 增益（Gain）值太高
- PID扫描时间（sample time）太长（对于快速响应PID的回路）

解决方法：降低增益（Gain）值并且/或选择短一些的扫描时间

- 2. 过程变量超过设定值很多（超调很大）



图 4.3.2

产生原因：积分时间 (Integral time)可能太高  
解决方法：降低积分时间 3. 得到一个非常不稳定的PID

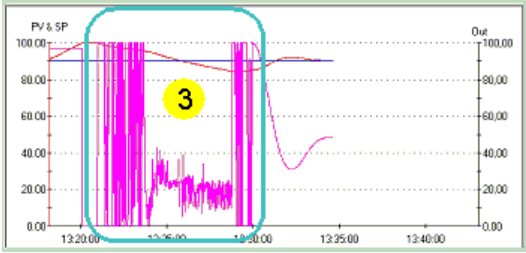


图 4.3.3

产生原因：

- 如果用了微分，可能是微分参数有问题
- 没有微分，可能是增益 (Gain) 值太高

解决方法：

- 调整微分参数到0 - 1的范围内
- 根据回路调节特性将增益值降低，最低可从0.x 开始逐渐增大往上调，直到获得稳定的PID。

❓ 如何获取一组合适的参数，实现快速并稳定的PID控制？

PID调节过程中，用户通常需要做多次的参数调节才能获得最优的控制效果。从下面反馈（过程变量）与给定之间的曲线图中，可以看到黄色曲线较理想。用户可以将调节的PID反馈与给定曲线与下图中对比，并修改相关参数（但是因为现场情况不一样，用户还需具体问题具体对待，下图中的建议仅供参考）：

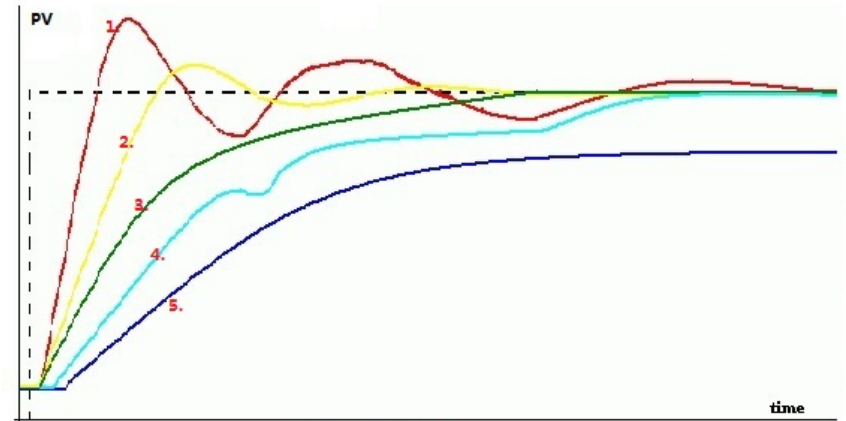


图 4.3.4 反馈与给定曲线

- 1.超调过大，减小比例，增大积分时间
- 2.迅速变化，存在小超调
- 3.实际值缓慢接近设定值，并且无超调的达到设定值
- 4.增益系数太小和/或微分时间太长
- 5.益系数太小和/或积分时间太长

常见问题

❓ 没有采用积分控制时，为何反馈达不到给定？

这是必然的。因为积分控制的作用在于消除纯比例调节系统固有的“静差”。没有积分控制的比例控制系统中，没有偏差就没有输出量，没有输出就不能维持反馈值与给定值相等。所以永远不能做到没有偏差。

❓ 对于某个具体的PID控制项目，是否可能事先得知比较合适的参数？有没有相关的经验数据？

虽然有理论上计算PID参数的方法，但由于闭环调节的影响因素很多而不能全部在数学上精确地描述，计算出的数值往往没有什么实际意义。因此，除了实际调试获得参数外，没有什么可用的经验参数值存在。甚至对于两套看似一样的系统，都可能通过实际调试得到完全不同的参数值。

❓ PID控制不稳定怎么办？如何调试PID？

闭环系统的调试，首先应当做开环测试。所谓开环，就是在PID调节器不投入工作的时候，观察：

- 反馈通道的信号是否稳定
- 输出通道是否动作正常

可以试着给出一些比较保守的PID参数，比如放大倍数（增益）不要太大，可以小于1，积分时间不要太短，以免引起振荡。在这个基础上，可以直接投入运行观察反馈的波形变化。给出一个阶跃给定，观察系统的响应是最好的方法。

如果反馈达到给定值之后，历经多次振荡才能稳定或者根本不稳定，应该考虑是否增益过大、积分时间过短；如果反馈迟迟不能跟随给定，上升速度很慢，应该考虑是否增益过小、积分时间过长……

总之，PID参数的调试是一个综合的、互相影响的过程，实际调试过程中的多次尝试是非常重要的步骤，也是必须的。

5 PID自整定

S7-200 中使用的自整定算法是基于 K.J. Åström 和 T. Hägglund 在 1984 年提出的延时反馈算法。经过这二十年，继电反馈算法已被应用于工业控制的各个领域。可以使用操作员面板中的用户程序或者 PID 整定控制面板来启动自整定功能。在同一时间，不仅仅只有一个 PID 回路可以进行自整定，如果需要的话，所有 8 个 PID 回路可以同时进行自整定。PID 自整定算法向您推荐增益值、积分时间值和微分时间值。您也可以为您的调节回路选择快速响应、中速响应、慢速响应或者极慢速响应等调节类型。

5.1. PID自整定先决条件

启动自整定先决条件：

- 要进行自整定的回路必须处于自动模式

在开始PID自整定调整前，整个PID控制回路必须工作在相对稳定的状态（稳定的PID是指过程变量接近设定值，输出不会不规则的变化，且回路的输出值在控制范围中心附近变化。）

理想状态下，自整定启动时，回路输出值应该在控制范围中心附近。自整定过程在回路的输出中加入一些小的阶跃变化，使得控制过程产生振荡。如果回路输出接近其控制范围的任一限值，自整定过程引入的阶跃变化可能导致输出值超出最小或最大范围限值。如果发生这种情况，可能会生成自整定错误条件，当然也会使推荐值并非最优。

为什么启动自整定之前，需要PID控制回路工作在相对稳定状态？

启动自整定后，回路计算自滞后序列时，不能执行正常的 PID 计算，此时回路输出时一个定值，不会根据偏差变化。因此，在启动自整定序列之前，控制过程应处于稳定状态。这样可以得到更好的滞后值结果，同时也可以保证自滞后序列期间控制过程不会失控。

5.2. PID自整定参数介绍

S7-200 中的 PID 自整定参数是回路表40-80字节，见下图：

| 36 | PID回路表ID           | ASCII码 | 常数        | "PIDA" (PID回路表, 版本A): ASCII常数                                   |
|----|--------------------|--------|-----------|---|
| 40 | a. AT控制(ACNTL)     | BYTE   | 输入        |   |
| 41 | b. AT状态(ASTAT)     | BYTE   | 输出        |   |
| 42 | c. AT结果(ARES)      | BYTE   | 输入/<br>输出 |   |
| 43 | d. AT配置(ACNFG)     | BYTE   | 输入        |   |
| 44 | e. 偏移(DEV)         | REAL   | 输入        | 最大过程变量振幅的标准化值(范围: 0.025-0.25)。                                  |
| 48 | f. 滞后(HYS)         | REAL   | 输入        | 过程变量滞后的标准化值，用于确定零相交(范围: 0.005到0.1)。当DEV与HYS的比率小于4时，自整定过程中会发出警告。 |
| 52 | g. 初始输出阶跃幅度(STEP)  | REAL   | 输入        | 输出阶跃幅度变化的标准化大小，用于减小过程变量的振动(范围: 0.05到0.4)                        |
| 56 | h. 看门狗时间(WDOG)     | REAL   | 输入        | 两次零相交之间允许的最大时间间隔，单位是秒(范围: 60到7200)                              |
| 60 | i. 推荐增益(AT_KC)     | REAL   | 输出        | 自整定过程推荐的增益值   |
| 64 | j. 推荐积分时间(AT_TI)   | REAL   | 输出        | 自整定过程推荐的积分时间值   |
| 68 | k. 推荐微分时间(AT_TD)   | REAL   | 输出        | 自整定过程推荐的微分时间值   |
| 72 | l. 实际输出阶跃幅度(ASTEP) | REAL   | 输出        | 自整定过程确定的归一化以后的输出阶跃幅度。   |
| 76 | m. 实际滞后(AHYS)      | REAL   | 输出        | 自整定过程确定的归一化以后的过程变量滞后值。  |

图 5.2.1. PID扩展回路表

a. AT控制 (ACNTL)：启动/中止自整定

|                            |  |
|----------------------------|--|
| AT控制(ACNTL)<br>输入一字节       | <div><div>MSB</div><div>7</div><div>0</div><div>0</div><div>0</div><div>0</div><div>0</div><div>0</div><div>0</div><div>EN</div><div>LSB</div><div>0</div></div> |
| EN - 设为1用于启动自整定；设为0用于中止自整定 |  |

图 5.2.2. PID自整定AT控制字节

b. AT状态 (ASTAT)：自整定的输出状态字节，PID控制面板自整定时的相关状态也是根据由该字节判断。

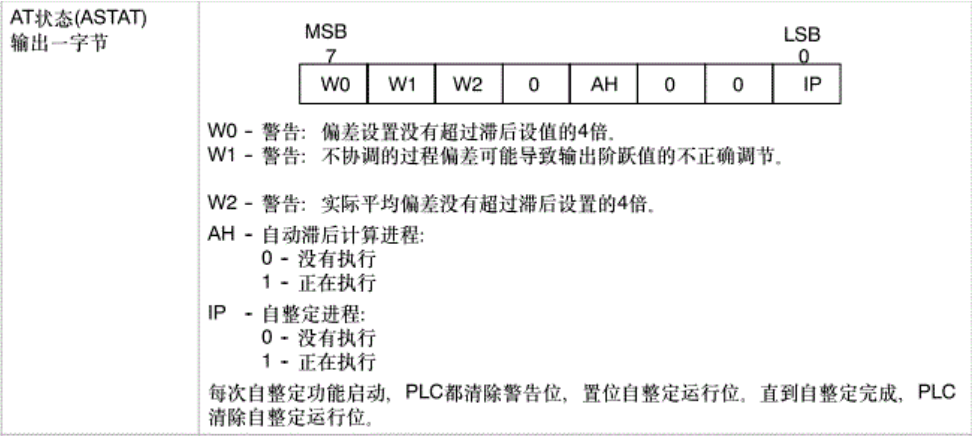


图 5.2.3. PID自整定AT状态字节

c.AT结果 (ARES)：PID自整定结果，需要注意，启动PID自整定之前需要确保该字节0位为0，尤其是自己编程启动自整定，可能需要手动设置为0。

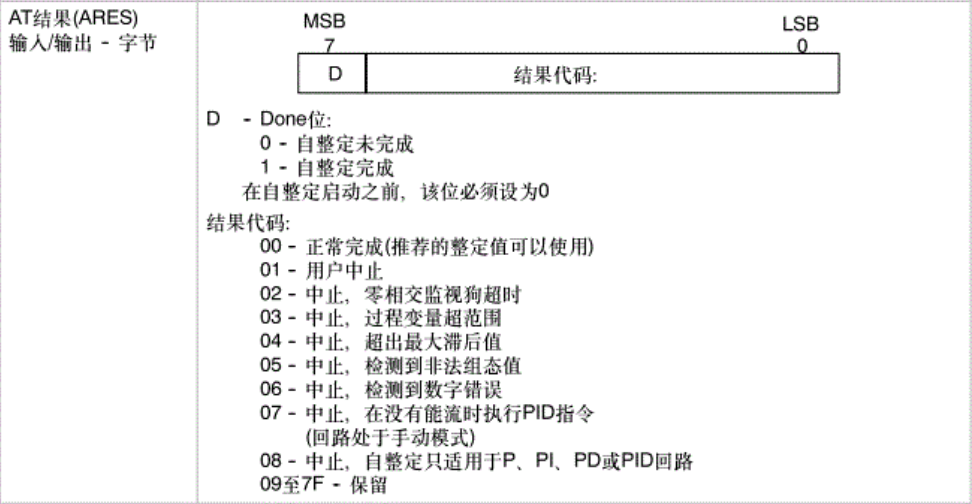


图 5.2.4. PID自整定AT结果字节

d.AT配置 (ACNFG)：自整定之前先对响应模式、偏差、滞后等做相关设置。




图 5.2.5. PID自整定AT配置字节

- e.偏移 (DEV) ~ h.看门狗时间 (WDOG)：参考[PID自整定高级参数设置](#)。
- i.推荐增益 (AT\_Kc) ~ k.推荐微分时间 (AT\_Td：PID自整定完成后，整定所得[推荐参数](#)放置在该地址。
- l.实际输出阶跃幅度 (ASTEP)：调节开始后，PID计算出的新的输出阶跃值。
- m.实际滞后 (AHYS)：重新计算得到的实际滞后值。

5.3. 通过PID控制面板启动自整定

S7-200的PID自整定实现方式有两种，一种是通过控制面板，另一种是自己编程。

-  需要注意：

• 通过控制面板实现PID自整定，PID必须是向导生成

• 自己编程实现PID自整定，向导或者PID指令块实现PID均可，具体步骤可以查看：[如何编程启动自整定](#)

[下面介绍通过PID控制面板实现自整定：](#)

**第一步：**在PID Wizard (向导) 中完成PID功能配置，正确调用PID子程序。

**第二步：**打开PID调节控制面板，设置PID回路调节参数。

在Micro/WIN V4.0在线的情况下，从主菜单Tools（工具）> PID Tune Control Panel（PID调节控制面板）进入PID调节控制面板，如果面板没有被激活（所有地方都是灰色），可点击Configure（配置）按钮运行CPU。

在PID调节面板图3.4.1的e.区选择要调节的PID回路号，在d.区选择Manual（手动），调节PID参数并点击Update（更新），使新参数值起作用，监视其趋势图，根据调节状况改变PID参数直至调节稳定。

⚠ 为了使PID自整定顺利进行，应当做到：

- 使PID调节器基本稳定，输出、反馈变化平缓，并且使反馈比较接近给定
- 设置合适的给定值，使PID调节器的输出远离趋势图的上、下坐标轴，以免PID自整定开始后输出值的变化范围受限制

⇒ 参见：[PID调节](#)

**第三步：**在图3.4.1的d.区点击Advanced（高级）按钮，设定PID自整定选项。如果不是很特殊的系统，也可以不加理会。

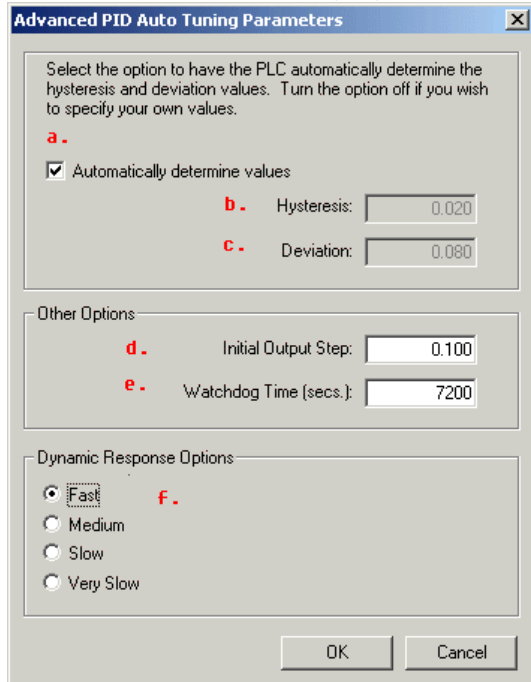


图5.3.1. 设置PID自整定高级选项

在此允许你设定下列参数：

- 你可以选中复选框，让自整定来自动计算死区值和偏移值。⚠ 对于一般的PID系统，建议使用自动选择。
- Hysteresis（滞回死区）：死区值规定了允许过程值偏离设定值的最大（正负）范围，过程反馈在这个范围内的变化不会引起PID自整定调节器改变输出，或者使PID自整定调节器“认为”这个范围内的变化是由于自己改变输出进行自整定调节而引起的。PID自整定开始后，只有过程反馈值超出了该区域，PID自整定调节器才会认为它对输出的改变发生了效果。这个值用来减少过程变量的噪声对自整定的干扰，从而更精确地计算出过程系统的自然振动频率。如果选用自动计算，则缺省值为2%。  
如果过程变量反馈干扰信号较强（噪声大）自然变化范围就大，可能需要人为设置一个较大的值。但这个值的改变要与下面的偏差值保持1:4的关系。
- Deviation（偏差）：偏差值决定了允许过程变量偏离设定值的峰峰值。如果选择自动计算该值，它将是死区的4倍，即8%。有些非常敏感的系统不允许过程量偏离给定值很多，也可以人工设置为比较小的值，但是要和上述“死区”设置保持比例关系。这就是说，一个精度要求高的系统，其反馈信号必须足够稳定。
- Initial Output Step（初始步长值）：PID调节的初始输出值。PID自整定开始后，PID自整定调节器将主动改变PID的输出值，以观察整个系统的反应。初始步长值就是输出的变动第一步变化值，以占实际输出量程的百分比表示。
- Watchdog Time（看门狗时间）：过程变量必须在此时间（时基为秒）内达到或穿越给定值，否则会产生看门狗超时错误。PID自整定调节器在改变输出后，如果超过此时间还未观察到过程反馈（从下至上或从上至下）穿越给定曲线，则超时。如果能够事先确定实际系统响应应非常慢，可以加长这个时间。
- 动态响应选项：根据回路过程（工艺）的要求可选择不同的响应类型：快速、中速、慢速、极慢速。
  - 快速：可能产生超调，属于欠阻尼响应
  - 中速：在产生超调的边缘，属于临界阻尼响应
  - 慢速：不会产生任何超调，属于过阻尼响应
  - 极慢速：不会产生任何超调，属于严重过阻尼响应
 ⚠ 用户在这里指定需要达到的系统控制效果，而不是对系统本身响应快慢的判断。
- 设定完参数点击OK键回到PID调节控制面板的主画面。

**第四步：**在手动将PID调节到稳定状态后，即过程值与设定值接近，且输出没有不规则的变化，并最好处于控制范围中心附近。此时可点击图3.4.1的d.区内的Start Auto Tune按钮启动PID自整定功能，这时按钮变为Stop Auto Tune。这时只需耐心等待，系统完成自整定后会自动将计算出的PID参数显示在图3.4.1的d.区。当按钮再次变为Start Auto Tune时，表示系统已经完成了PID自整定。

⚠ 要使用自整定功能，必须保证PID回路处于自动模式。开始自整定后，给定值不能再改变。



**第五步：**如果用户想将PID自整定的参数应用到当前PLC中，则只需点击Update PLC，将整定后推荐参数更新到PID的增益、积分时间、微分时间内。

💡 完成PID调整后，最好下载一次整个项目（包括数据块），使新参数保存到CPU的EEPROM中。

❓ **控制面板上会显示当前自整定的状态吗？**

可以，在控制面板右下方空白处会显示当前的自整定所处阶段，见下图：

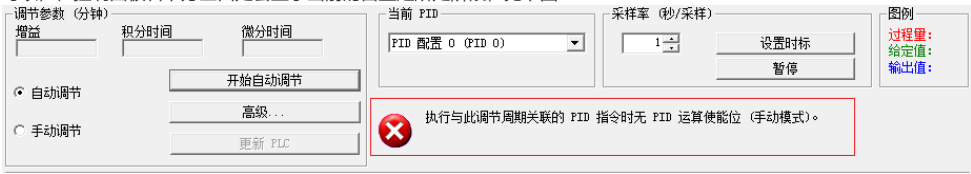


图5.3.2. PID自整定状态  
图中错误是因为在手动状态下启动自整定。

❓ **PID自整定具体是怎样实现的？**

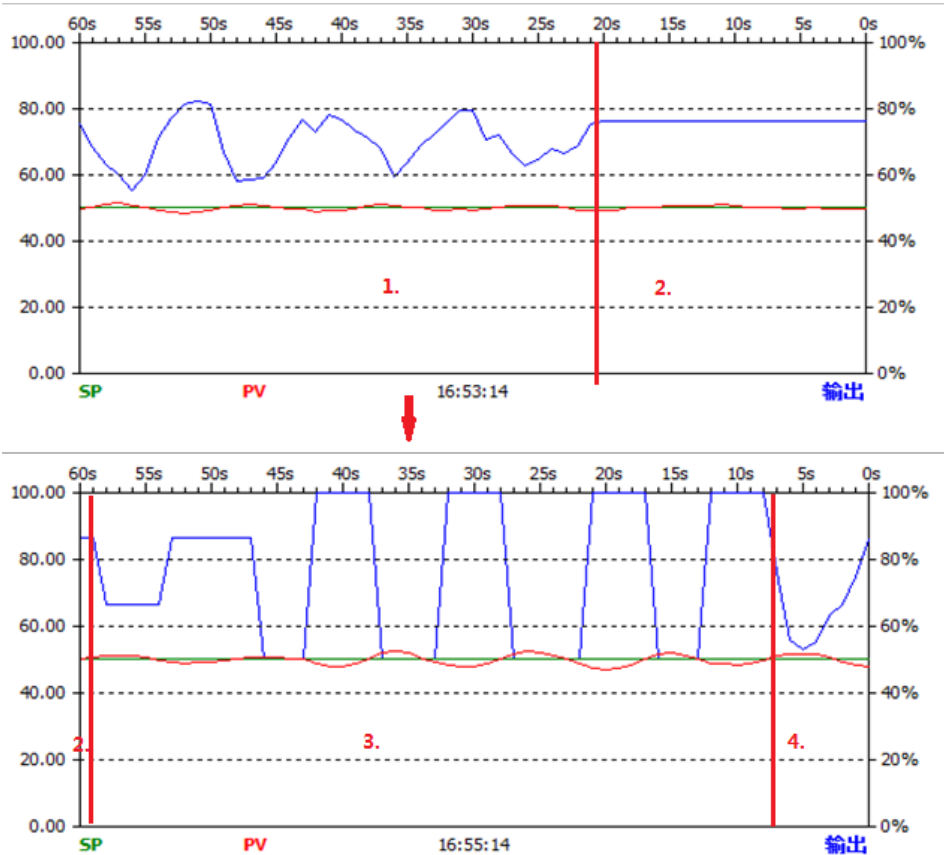


图5.3.3. PID自整定

1. 回路满足条件，启动PID自整定。
2. 计算自滞后和自偏移：如果选择自动确定滞后值/偏移值，PID 自整定器将进入滞后确定序列。该序列包含一段时间内的过程变量采样值（为了得到具有统计意义的采样数据，至少要有100个采样值。如果回路的采样时间为200毫秒，则采集100个样本需要20秒。采样时间较长的回路需要更多时间。即使回路的采样时间小于200毫秒，从而采样100次的时间不需要20秒，滞后确定序列仍然需要至少20秒的采样时间。计算自滞后序列时，不能执行正常的PID计算，输出保持上一周期PID计算值。），然后根据采样结果计算出标准偏移。滞后参数指定了相对于设定值的偏移（正或负），PV（过程变量）在此偏移范围内时，不会导致控制器改变输出值。偏移用于减小PV信号中噪声的影响，从而更精确地计算出过程的固有振动频率。
3. 自整定序列：在得到滞后值和偏移值之后开始执行自整定序列（输出值的上述变化会导致过程变量值产生相应的变化。当输出的变化使PV远离设定值以至于超出滞后区范围时，自整定器就会检测到过零事件。每次发生过零事件时，自整定器将反方向改变输出。整定器会继续对PV进行采样，并等待下一个过零事件。要完成序列，整定器总共需要12次过零事件。过程变量的PV振动幅度和频率代表着控制过程增益和自然频率。），根据自整定过程期间采集到的过程的频率和增益的相关信息，能够计算出最终增益和频率值。通过这些值可以计算出增益（回路增益）、复位（积分时间）和速率（微分时间）的建议值。
4. 自整定序列完成后，回路输出会恢复到初始值。下一次执行回路时，将执行正常的PID计算。

想要了解PID自整定的详细过程，请查看[S7-200系统手册](#)第15章 PID自整定和PID控制面板。

**5.4. 通过编程启动自整定**

**第一步：**在数据块设置自整定相关参数：偏差、滞后、初始输出阶跃响应、看门狗时间、动态响应。

```
AT配置:VB143 16#03 // 快速响应，不使用回路表中的偏差值、滞后值
初始输出阶跃幅度:VD152 0.1
看门狗时间:VD156 7200.0
```

图5.4.1. PID自整定参数设置

**第二步：**确认PID自整定的2个先决条件是否满足：PID指令块正常调用，同时被控系统处于相对稳定状态。



第三步：确认自整定AT结果字节0位为0，设置AT控制字节为1，(M3.1)启动自整定。

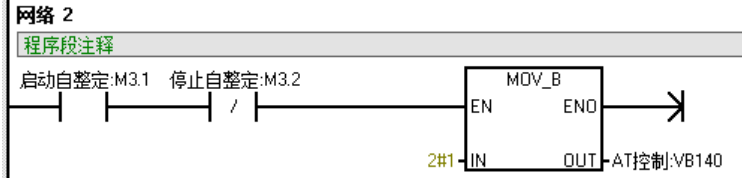


图5.4.2. 启动自整定

第四步：整定过程中如果想要暂停自整定，可以设AT控制为0，并手动(M3.2) 将AT结果清零，复位启动按钮。

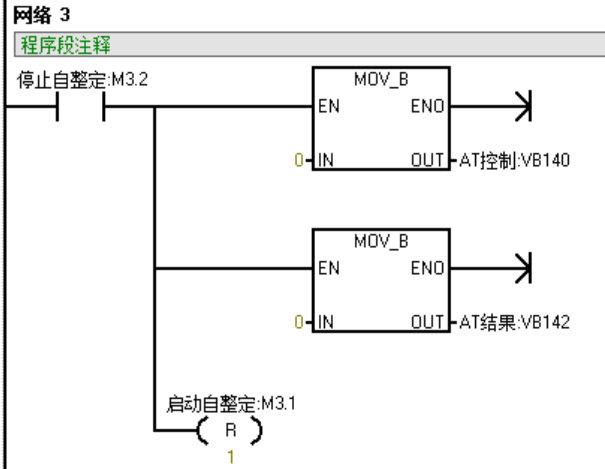


图5.4.3. 停止自整定

第四步：自整定成功 (VB142=128) 后，选择(M3.3)是否将整定后的推荐参数送至PID参数并复位相关自整定参数。

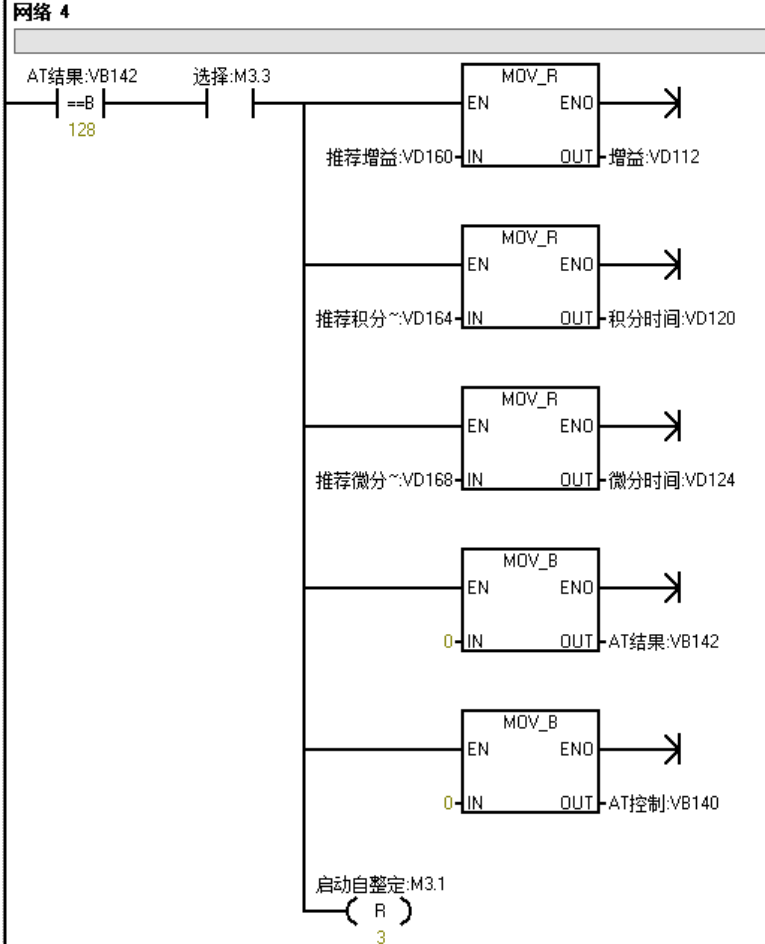


图5.4.4. 整定后参数设置

以上步骤仅供参考，需要根据实际情况，做不同的变换。

5.5. PID自整定失败的原因

1. PID输出在最大值与最小值之间振荡（曲线接触到坐标轴）

2.1 介绍PID回路表

2.2 通过PID指令块实现PID

3 向导实现PID

3.1 PID向导设置步骤

3.2 PID向导生成的组件介绍

3.3 PID设定值与过程变量程设置

3.4 PID调节控制面板使用

4 PID调节

4.1 PID调节手/自动无扰动换

4.2 PID调节步骤

4.3 手动调节PID至稳定

5 PID自整定

5.1 PID自整定先决条件

5.2 PID自整定参数介绍

5.3 通过PID控制面板启动自整定

5.4 通过编程启动自整定

5.5 PID自整定失败原因

附录

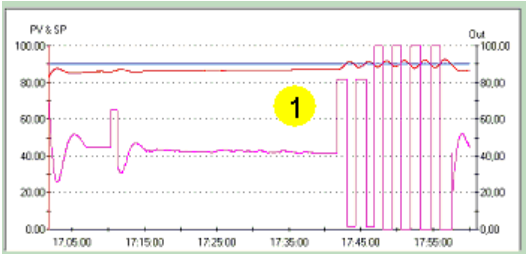


图5.5.1. 整定趋势图

解决方法：降低PID初始输出步长值（initial output step）

2. 经过一段时间后，PID自整定面板显示如下信息：“The Auto Tune algorithm was aborted due to a zero-crossing watchdog timeout.” 即自整定计算因为等待反馈穿越给定值的看门狗超时而失败。

解决方法：确定在启动PID自整定前，过程变量和输出值已经稳定。并检查Watchdog Time的值，将其适当增大。

➡ 对于其它错误，可参考下表或者[S7-200系统手册](#)。

| 结果代码（在AT结果 ARES中查看）      | 情况  |
|--------------------------|---|
| 01 由用户中止                 | EN位在调谐进行中清除   |
| 02 由于过零看门狗超时而中止          | 半循环持续时间超过过零看门狗间隔  |
| 03 由于进程超出范围而中止           | PV超出范围：<br>-在自动滞后序列过程中<br>-在第四次过零前两次<br>-在第四次过零后  |
| 04 由于滞后值超出最大值而中止         | 用户指定的滞后值或自动确定的滞后值 > 最大值   |
| 05 由于非法配置值而中止            | 下列范围检查错误：<br>-初始回路输出值 < 0.0 或 > 1.0<br>-用户指定的偏差值=滞后值或 > 最大值<br>-初始输出步长=0.0 或 > 最大偏差<br>-过零看门狗间隔时间 < 最小值<br>-回路表中的抽样时间值为负数 |
| 06 由于数字错误而中止             | 遇到非法的浮点数或除数为零   |
| 07 执行PID指令时无使能位（手动模式）    | 当自动调谐在进行中或被请求时PID指令以无使能位的方式执行   |
| 08 自动调谐只可用于P、PI、PD或PID等回 | 回路类型不是P、PI、PD或PID   |

表5.5.1. PID自整定错误代码

6 扩展阅读

技术论坛

西门子S7-200系列PLC的PID功能块的应用经验  
如何进行PID参数整定？

视频中心

S7-200CN: PID向导编程2-跟我学  
S7-200CN: PID向导编程2-跟我学  
S7-200CN: PID向导编程3-跟我学

找答案

s7 200 PID 死区设置问题  
PID自整定功能  
西门子S7-200中PID，向导和自己做PID各有哪些优缺点，一般选择哪种方式？  
200plc中pid和中断做流量累积如何同时使用  
S7-200的PID调节。当过程量与设定值不一样时，Pid却没有输出，这时怎么回事啊？