# PIDLIB

## PIDAT（FB）

PID : Parameter Self-Tuning Function Block

|  |  |  |
| --- | --- | --- |
| Name | PIDAT（FB）PID Parameter Self-Tuning Function Block | |
|  | | ST |
|  | | PIDAT(  RUN:= ,  ManualCtrl:= ,  AutoTune:= ,  SV:=,  PV:=,  CycleTime:= , Proportional:= , IntegrationTime:= , DerivativeTime:= , RangLow:= ,  RangUp:= ,  MVLow:=,  MVUp:= ,  ManualMV:= , Direction:= ,  Options:= ,  ATDone=> ,  ATBusy=> ,  Error=> ,  ErrorID=> ,  MV=> ); |

1. Variable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input/Output | Parameters | name | Data type | Description |
| Input | RUN | Execute FB | BOOL | TRUE executes the command, FALSE stops it. |
| Input | ManualCtrl | Manual control | BOOL | When RUN is TRUE, this parameter enables manual control if set to TRUE and disables it if set to FALSE. |
| Input | AutoTune | Self-Tuning | BOOL | When RUN is TRUE and ManualCtrl is FALSE, this parameter enables PID parameter self-tuning if set to TRUE and disables it if set to FALSE. |
| Input | SV | Setpoint | REAL | Target value, range: RangLow to RangUp. |
| Input | PV | Current Value | REAL | Current value, range: RangLow to RangUp. |
| Input | CycleTime | PID Processing Cycle | LTIME | PID Processing Cycle |
| Input | RangLow | Lower Limit of Controlled Object | REAL | Lower limit of target and current values |
| Input | RangUp | Upper Limit of Controlled Object | REAL | Upper limit of target and current values |
| Input | MVLow | Lower Limit of Manipulated Variable | REAL | Lower limit of manipulated variable |
| Input | MVUp | Upper Limit of Manipulated Variable | REAL | Upper limit of manipulated variable |
| Input | ManualMV | Manual MV Value | REAL | Manipulated variable during manual control |
| Input | Direction | Direction | BOOL | TRUE: Reverse (E=PV-SV) |
| Input | Options |  | Options PIDAT | FALSE: Forward (E=SV-PV) |
| Output | ATDone | Self-Tuning Complete | BOOL | ATHystrs: Error value during parameter self-tuning. Smaller values yield higher accuracy. |
| Output | ATBusy | Self-Tuning in Progress | BOOL | Parameter self-tuning completed |
| Output | Error | Error | BOOL | Parameter self-tuning in progress |
| Output | ErrorlD | Error ID | WORD | Command execution error |
| Output | MV | Manipulated Variable | REAL | Error code when command execution fails |
| Input/Output | Proportional | Proportional Coefficient | REAL | PID calculation result. |
| Input/Output | IntegrationTime | Integral Time | LTIME | Proportional coefficient |
| Input/Output | DerivativeTime | Derivative Time | LTIME | Integral time |

**要点说明**

⚫ 一般情况下，PIDAT(FB)与PWM\_S(FB)相互配合使用，配合使用时，一般地，会将PIDAT功能块的引脚【CycleTime】要小于PWM\_S引脚的【CycleTime】且成倍数关系，这样Output占空比的都是PID新更新的，不至于重复使用上次的MV导致控制效果差；

⚫ 初次整定时，功能块PID 为0，可将PID赋值上初值，若开启自整定功能，该PID参数无效，功能块会自己调节，当调整完成后会算出正确的PID参数覆盖上去，否则使用该PID参数；

⚫ 更改目标温度时无需重新RUN；

⚫ 一般的，会将比例、积分、微分时间设置为掉电保持区，以便后续使用；

⚫ PID 参数。具体操作方法会在下文举例中作详细介绍。

Key Points

⚫ Typically, PIDAT(FB) and PWM\_S(FB) are used in conjunction. When combined, the [CycleTime] pin of the PIDAT function block should be set to a value smaller than the [CycleTime] pin of PWM\_S and be a multiple thereof. This ensures the output duty cycle reflects the newly updated PID value, preventing reuse of the previous MV that could degrade control performance.

⚫ During initial tuning, the PID block value is set to 0. You may assign an initial value to PID. If auto-tuning is enabled, this PID parameter becomes invalid as the block self-adjusts. Upon completion, it calculates correct PID parameters to overwrite the initial value; otherwise, the assigned PID parameter remains in effect.

⚫ Changing the target temperature does not require restarting the RUN cycle.

⚫ Typically, the proportional, integral, and derivative time constants are set as power-off retention parameters for subsequent use.

⚫ PID parameters. Detailed operational methods will be illustrated in the examples below.

## Example

### Hardware：

(1) Omron thermostat transmits temperature values via RS-485.

(2) Solid-state relay controls heater on/off.

(3) Type K thermocouple.

(4) Heating platform.

### Usage example

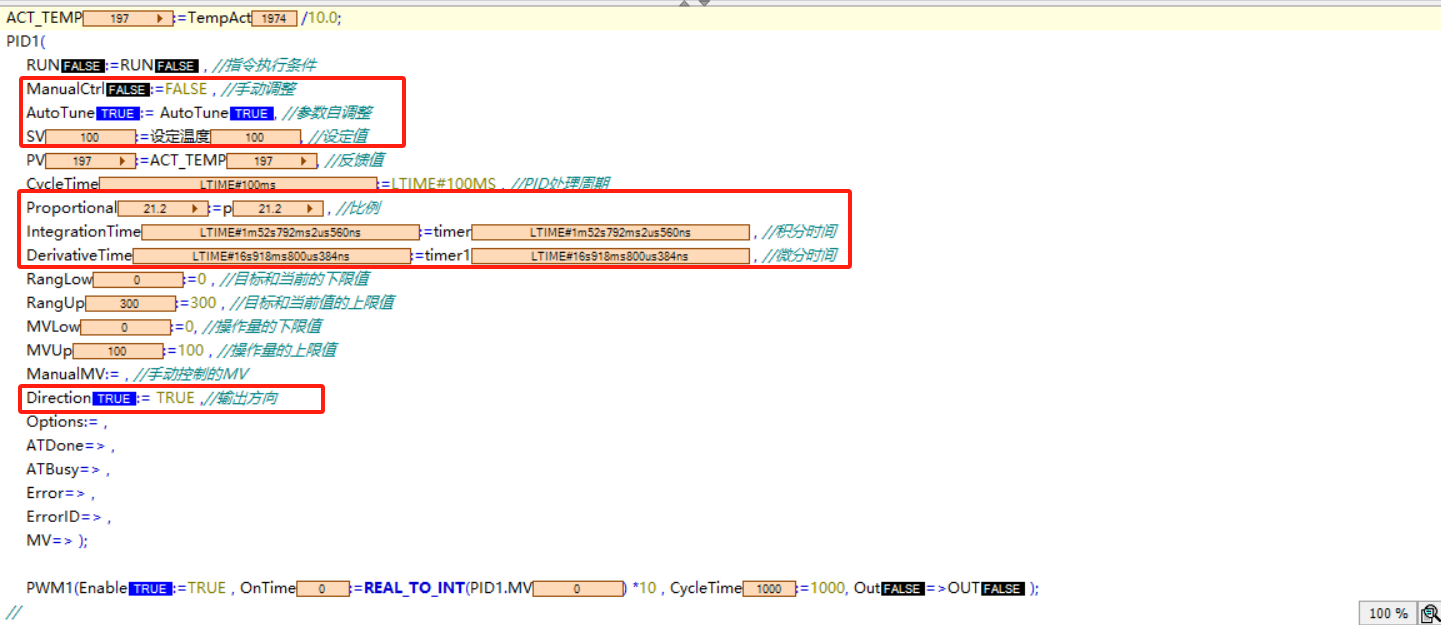
【1】This example utilizes the temperature from a Type K thermocouple, read via the 485T communication of an Omron temperature controller, as the input (PV) to the function block's pin. This serves as the controlled variable. The function block's output pin (MV) is then used as the OnTime for PWM\_S. The modulated pulse width controls the switching of the solid-state relay to regulate heating.

【2】PWM\_S's CycleTime is 1000 ms units. PWM\_S's OnTime is set to PIDAT's MV multiplied by 10, where PIDAT's MV ranges from 0 to 100. This effectively adjusts the ON time as a percentage of 1 second.

【3】The Output represents one output terminal of the controller unit.



【4】As shown in the function block diagram below, enable auto-tuning, set the target temperature to 100, and assign arbitrary PID values (PID parameters cannot be zero, otherwise an error will occur). In auto-tuning mode, this set of PID parameters is not used. Select positive direction for the output and negative feedback for the feedback loop.

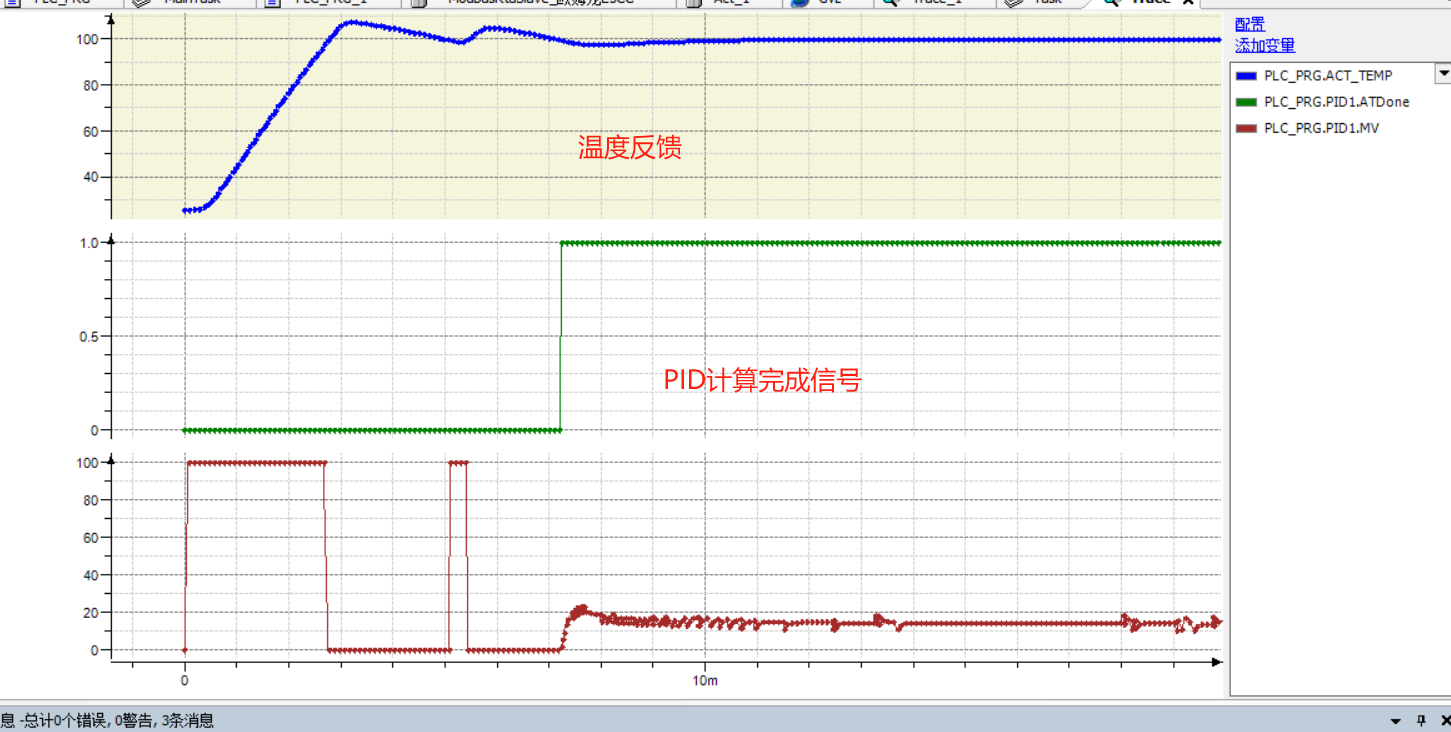


【5】For example, set the PID processing time to 100ms and the MV range to 0-100.



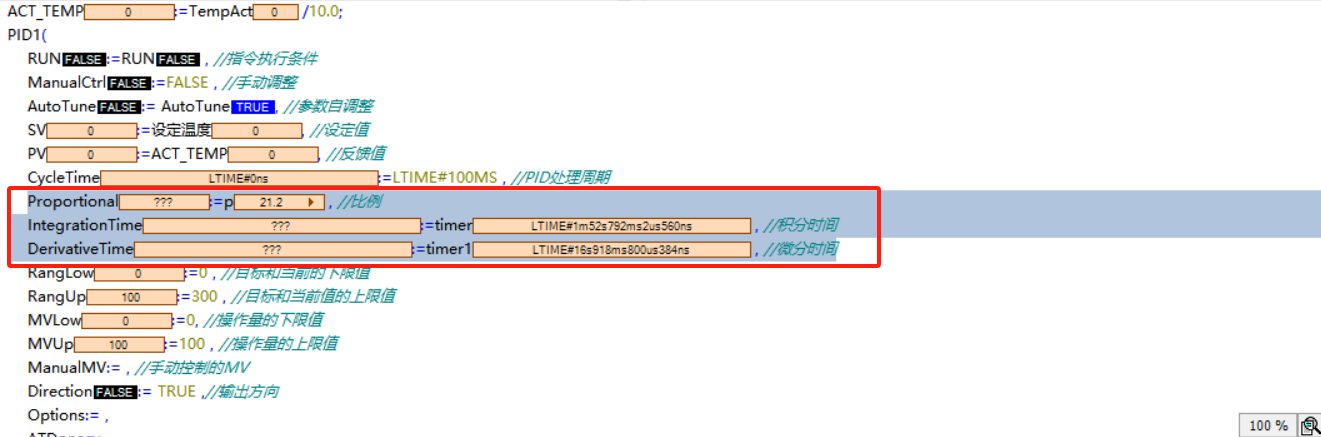
【6】Execute RUN to start the function block. Output ATBUSY indicates auto-tuning PID execution has begun.

【7】When ATDone is set, it signifies calculation completion, and new PID parameters are generated. The diagram below shows the PID self-tuning process. When the ATDone signal is set, it indicates calculation completion, and the results will be written to the proportional coefficient, derivative time, and integral time pins.



Temperature Feedback

PID calculation completion signal



【8】To reuse this parameter set without auto-tuning next time, reset AutoTune and run directly to use these PID parameters.