# 7 PID Control Function

In this chapter, we mainly introduce the applications of PID instructions for XD, XL series, including: call the instructions, set the parameters, items to notice, sample programs etc.

## 7-1． PID Introduction

PID instruction and auto tune function are added into XD/XL series PLC basic units. Via auto tune method, users can get the best sampling time and PID parameters and improve the control precision.
PID instruction has brought many facilities to the users.
Output can be data form D, HD, and on-off quantity Y, user can choose them freely when programming.
Via auto tune, users can get the best sampling time and PID parameters and improve the control precision.
User can choose positive or negative action via software setting. Positive action is used for heating control; negative action is used for cooling control.
PID control separates the basic units with the expansions, which improves the flexibility of this function.
XD/XL series PLC have two methods for auto tune, step response method and critical oscillation method.
For temperature control object:
Step response method: the PID auto tune will start when current temperature of object controlled is equal to ambient temperature.
Critical oscillation method: the PID auto tune can start at any temperature.

## 7-2． Instruction Form

Brief Introduction of the Instructions
Execute PID control instructions with the data in specified registers.

| PID control [PID] | | | |
|---|---|---|---|
| 16 bits instruction | PID | 32 bits instruction | - |
| Executing condition | Normally ON/normally closed coil trigger | Suitable models | XD/XL |
| Hardware requirement | - | Software requirement | V3.2 |

Operands

| Operands | Function | Type |
|---|---|---|
| S1 | set the address of the target value (SV) | 16bits, BIN |
| S2 | set the address of the tested value (PV) | 16 bits, BIN |
| S3 | set the start address of the control parameters | 16 bits, BIN |
| D | the address of the operation result (MV) or output port | 16 bits, BIN; bit |

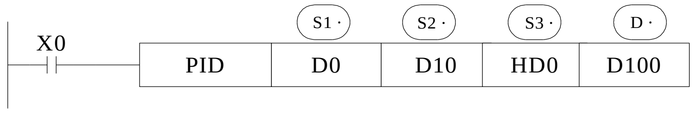
Suitable soft components

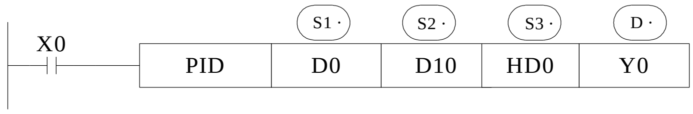
| | Operands | System | | | | | | | | Constant | Module | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | D* | FD | TD* | CD* | DX | DY | DM* | DS* | K/H | ID | QD |
| Word | S1 | ● | ● | | | | | | | ● | | |
| | S2 | ● | ● | | | | | | | | | |
| | S3 | ● | ● | | | | | | | | | |
| | D | ● | ● | | | | | | | | | |

| | Operands | System | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | X | Y | M* | S* | T* | C* | Dn.m |
| Bit | D | | ● | ● | ● | ● | ● | |

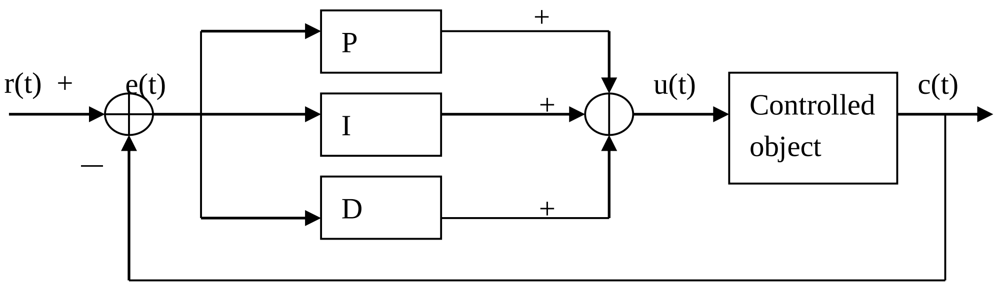
*Note: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.
M includes M, HM, SM; S includes S and HS; T includes T and HT; C includes C and HC.

## Functions and Action




S3~ S3+ 69 will be occupied by this instruction, so please don’t use them as the common data registers.
This instruction executes when each sampling time interval comes.
For the operation result, data registers are used to store PID output values; the output points are used to output the occupy duty ratio in the form of ON/OFF.
PID control rules are shown as below:
P: proportion, I: integral, D: differential




Analog PID control system

$e(t) = r(t) - c(t)$ (1-1)

$u(t) = Kp\,[\,e(t) + 1/Ti \int e(t)dt + TD\, de(t)/dt\,]$ (1-2)

Here, e(t) is offset value, r ( t ) is the setting value, c ( t ) is actual output value and the u(t) is the control value;

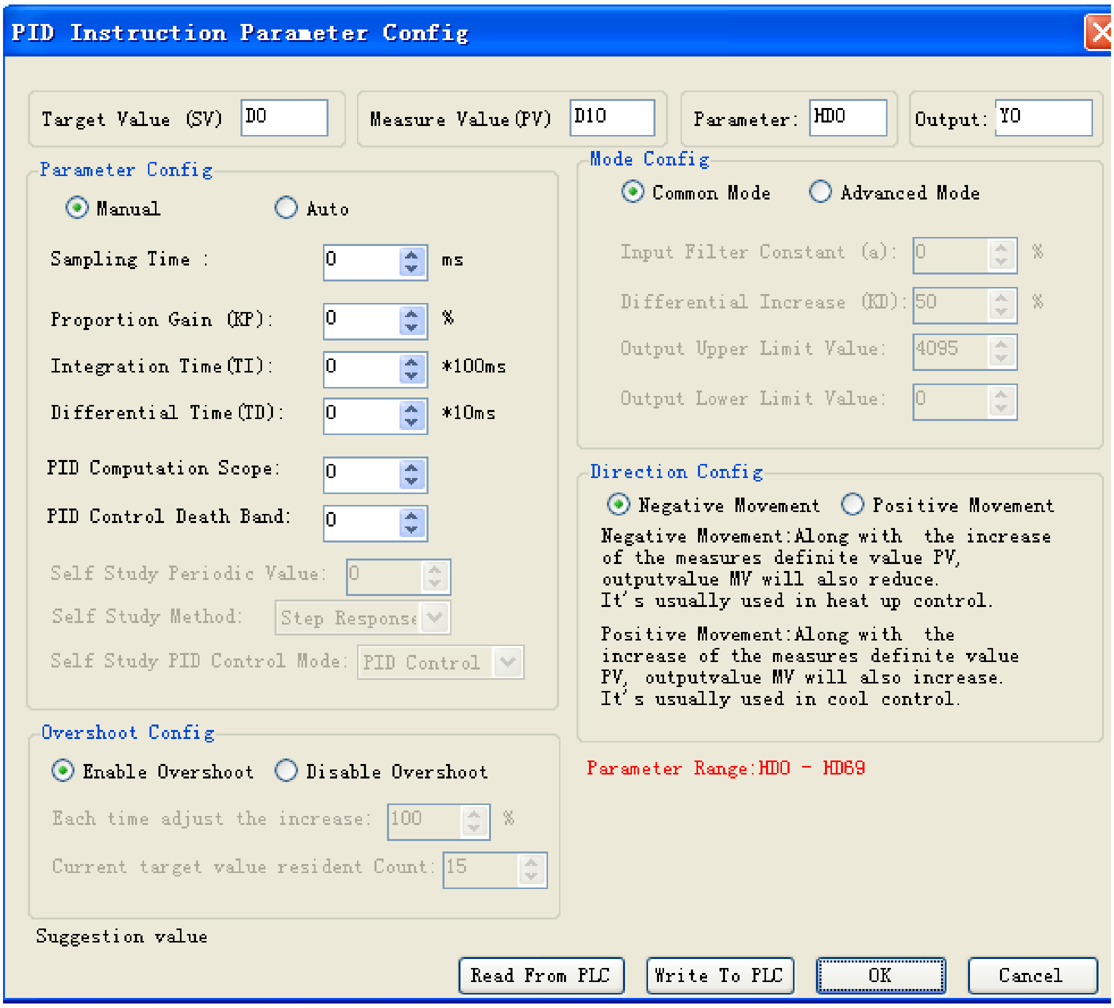
In function (1-2), Kp is the proportion coefficient, Ti is the integration time coefficient, and TD is the differential time coefficient.

The result of the operation:

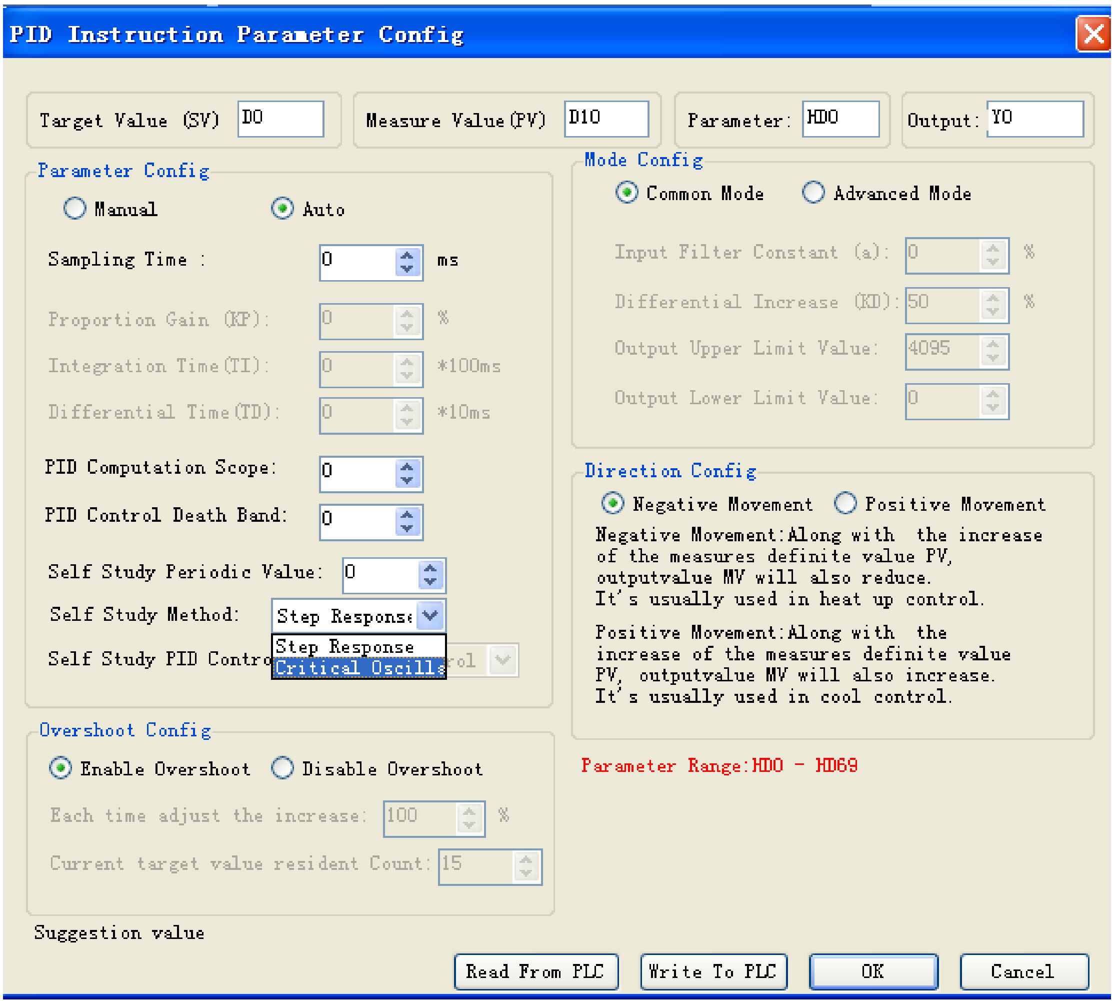
1. Analog output: digital form of MV = u(t), the default range is 0~4095.
2. Digital output: Y = T * [MV / PID output upper limit]. Y is the outputs activate time within the control cycle. T is the control cycle, equals to the sampling time. PID output upper limit default value is 4095.

## 7-3. Parameters setting

Users can call PID in XDP Pro software directly and set the parameters in the window (see graph below), for the details please refer to XDP Pro user manual. Users can also write the parameters into the specified registers by MOV instructions before PID operation.



Auto tune mode:




V3.2 and higher version software can choose auto tune mode: step response or critical oscillation.

## 7-3-1. Register and their functions

PID control instruction's relative parameters ID, please refer to the below table:

| ID | Function | Description | Memo |
|---|---|---|---|
| S3 | Sampling time | Whatever it is manual or auto mode, all needs to set | 32 bits without sign, Unit ms |
| S3+2 | Mode setting | bit0: 0: negative action;<br>1: positive action<br>bit1～bit6 not usable<br>bit7:<br>0: manual PID;<br>1: auto tune PID<br>bit8: 1: auto tune successful flag<br>bit9～bit10: auto tune method<br>00: step response<br>01: critical oscillation | |

| | | | |
|---|---|---|---|
| | | bit11～bit12：not useful<br>bit13～bit14 auto tune PID mode (valid in critical oscillation mode)<br>00：PID control<br>01：PI control<br>10：P control<br>bit15：<br>0：regular mode；<br>1：advanced mode; | |
| S3+3 | Proportion Gain (Kp) | Range：1～32767[%] | |
| S3+4 | Integration time (TI) | 0～32767[unit: 100ms] | 0 is taken as no integral. |
| S3+5 | Differential time ( TD) | 0～32767[unit: 10ms] | 0 is taken as no differential. |
| S3+6 | PID operation zone | 0～32767 | PID adjustment band width value |
| S3+7 | Control death zone | 0～32767 | PID output value will not change in death zone |
| S3+8 | Sampling temperature filter coefficient | 0～100[%] | Filter the input sampling temperature in advanced mode, 0 is no input filter |
| S3+9 | Differential gain( KD) | 0～100[%] | Only for advanced mode (normal mode default value is 50%), 0 is no differential gain |
| S3+10 | Upper limit value of output | 0～32767 | |
| S3+11 | Lower limit value of output | 0～32767 | |
| S3+12 | Change of Unit Temperature Corresponds to Change of AD Value | full scale AD value *（0.3~1%）<br>default value is 10 | 16-bit no sign, only for step PID |
| S3+13 | PID auto tune overshoot | 0：enable overshoot<br>1： not overshoot（try to reduce the overshoot） | only for step PID |
| S3+14 | Current target value adjusting percentage every time in auto tune end transition stage | Cannot adjust | 16-bit no sign, only for step PID |
| S3+15 | Number of times exceeding the target value in auto tune end transition stage when limiting the overshoot | | only for step PID, default value is 15 |
| S3+16 | PID type and status | Bit0~bit1：<br>00: manual mode<br>01: step mode | Internal use parameters of the system for |

| | | | |
|---|---|---|---|
| | | 10: Critical oscillation mode<br>Bit8：<br>0: manual control status<br>1: auto tune end, enter manual control status | monitoring purposes only |
| S3+17 | PID max output | 0～32767 | Internal use parameters of the system for monitoring purposes only |
| S3+18 | PID min output | 0～32767 | Internal use parameters of the system for monitoring purposes only |
| S3+19 | Last time sampling time | 0~sampling time (unit: ms) | 16-bit no sign, Internal use parameters of the system for monitoring purposes only |
| S3+20 | Actual sampling time space | The value is around the sampling time | 32-bit no sign, Internal use parameters of the system for monitoring purposes only |
| S3+22 | Last time user set target temperature | The value before changing the target temperature | Internal use parameters of the system for monitoring purposes only |
| S3+23 | - | - | Parameter is reserved |

| **The following is the joint address (divided into step setting, critical oscillation setting and manual control)** | | | |
|---|---|---|---|
| **Step part (read only parameters, only for monitoring)** | | | |
| S3+24 | Actual sampling space | 0～4294967296 (unit: ms) | Internal usage parameters of the system |
| S3+26 | Operating segment of auto-tuning PID | 0: Preparation stage<br>1～2: auto tune parameter collection<br>3: calculate PID parameters | Internal usage parameters of the system |
| S3+28 | Duration of auto-tuning PID operating parameters | 0～4294967296 (unit: ms) | Internal usage parameters of the system |
| S3+30 | Real-time accumulation of two inflection points | Clear and recalculate the time when reaching the inflection point<br>0～4294967296 (unit: ms) | Internal usage parameters of the system |

| | | | |
|---|---|---|---|
| S3+32 | Sampling variation of inflection point | Sampling difference between two inflection points<br>-2147483648~2147483647 | Internal usage parameters of the system |
| S3+34 | Sampling interval time of inflection point EK | 0～4294967296 (unit: ms) | Internal usage parameters of the system |
| S3+36 | Time from auto-tuning PID to inflection point | 0～4294967296 (unit: ms) | Internal usage parameters of the system |
| S3+38 | Last sampling temperature | -32767~32767 | Internal usage parameters of the system |
| S3+39 | The time from auto-tuning PID operation to inflection point | -32767~32767 (unit: ms) | Internal usage parameters of the system |
| S3+40 | Starting sampling value of auto-tuning PID operation | -32767~32767 | Internal usage parameters of the system |
| S3+41 | Number of times at inflection point during auto-tuning | 0~65535 | Internal usage parameters of the system |
| S3+42 | Useless time | 0～4294967296 (unit: ms) | Internal usage parameters of the system |
| S3+44 | Stop temperature | Temperature at the end of auto-tuning<br>Range: -32767~32767 | Internal usage parameters of the system |
| **Critical oscillation part (read only parameters, only for monitoring)** | | | |
| S3+24 | PID control mode | 0: PID control<br>1: PI control<br>2: P control | 16-bit no sign, internal usage parameters of the system |
| S3+25 | Current auto-tuning segment | 0: Preparation stage<br>1: start to auto tune<br>2～3: auto-tuning parameter collection<br>4: calculation of PID parameters | 16-bit no sign, internal usage parameters of the system |
| S3+26 | The auto-tuning temperature is located at the number of peaks | 0: first peak<br>1: second peak | 16-bit no sign, internal usage parameters of the system |
| S3+27 | The lowest sampling temperature | -32767～32767 | Internal usage parameters of the system |
| S3+28 | The highest sampling temperature | -32767～32767 | Internal usage parameters of the system |
| S3+30 | sampling time of the lowest sampling temperature | 0～4294967296 (unit: ms) | Internal usage parameters of the system |
| S3+32 | sampling time of the highest sampling temperature | 0～4294967296 (unit: ms) | Internal usage parameters of the system |

| | | | |
|---|---|---|---|
| S3+34 | auto-tuning time cumulative | 0～4294967296 (unit: ms) | Internal usage parameters of the system |
| **Manual control part (read only parameters, only for monitoring)** | | | |
| S3+24 | current target temperature | -32767~32767 | Internal usage parameters of the system |
| S3+25 | Need to update target temperature | 0: no need<br>1: need | 16-bit no sign, internal usage parameters of the system |
| S3+26 | Number of times to reach target temperature | 0~65535 | Internal usage parameters of the system |
| S3+27 | PID upper limit of operational range | -32767~32767 | Internal usage parameters of the system |
| S3+28 | PID lower limit of operational range | -32767~32767 | Internal usage parameters of the system |
| S3+30 | High voltage time when PID uses Y to output | 0～4294967296 (unit: ms) | Internal usage parameters of the system |
| S3+32 | Sampling temperature after last filtering | The filtered temperature acquired in the last sampling time (the input filter constant in the advanced mode needs to be set first) | Floating point, internal usage parameters of the system |
| S3+34 | Last temperature deviation | | Floating point, internal usage parameters of the system |
| S3+36 | Value of last integral term | digital value corresponding to Ui of the last sampling time | Floating point, internal usage parameters of the system |
| S3+38 | Value of last differential term | digital value corresponding to Ud of the last sampling time | Floating point, internal usage parameters of the system |
| S3+40 | Last PID output | | Floating point, internal usage parameters of the system |

Note: When the auto-tuning mode is changed to manual control, the value in the original address of S3+24~S3+40 will be overwritten by the value in manual control mode.

## 7-3-2．Parameters Description

**Movement direction:**

Positive movement: the output value MV will increase with the increasing of the measured value PV, usually used for cooling control.

Negative movement: the output value MV will decrease with the increasing of the measured value PV, usually used for heating control.

**Mode setting**

Common Mode:

Parameters register range: S3～S3+69, and S3～S3+7 need to be set by users;

S3+8～S3+69 are occupied by system, users can't use them.

Advanced Mode

Parameters register range: S3～S3+69, among them S3～S3+7 and S3+8～S3+11 need to be set by users; S3+16～S3+69 are occupied by system, users can't use them.

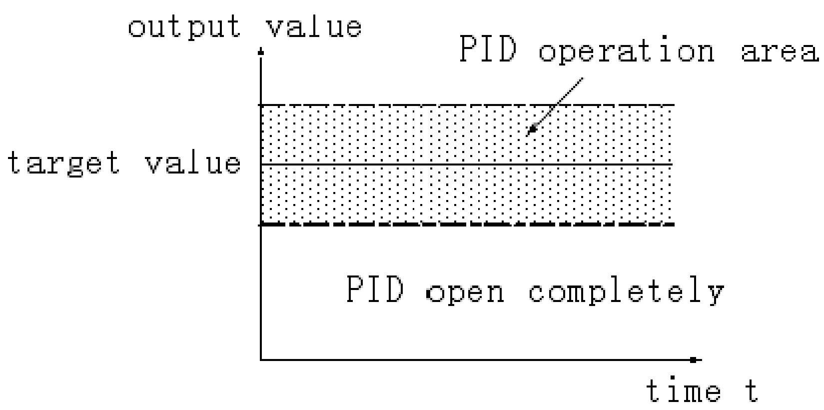
**Sample time[S3]**

The system samples the current values according to some certain interval and compares them with the output value. This time interval is the sample time **T**. There is no requirement for **T** during **DA** output; **T** should be larger than one PLC scan period during port output. **T** value should be chosen among 100~1000 times of PLC scan periods.

**PID Operation Zone[S3+6]**

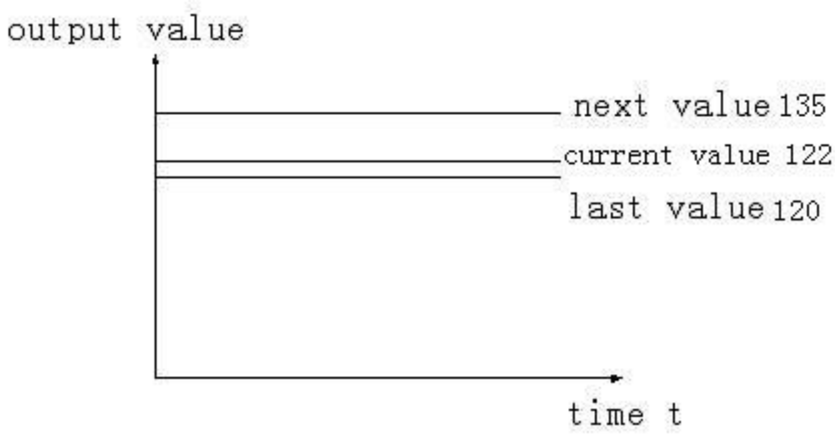
PID control is entirely opened at the beginning and close to the target value with the highest speed (default value is 4095), when it entered into the PID computation range, parameters Kp, TI, TD will be effective.

See graph below:




If the target value is 100, PID operation zone is 10, and then the real PID's operation zone is from 90~110.

**Death Region [S3+7]**
If the measured value changed slightly for a long time, and PID control is still in working mode, then it belongs to meaningless control. Via setting the control death region, we can overcome this situation. See graph below:




Suppose: we see the death region value to be 10. Then in the above graph, the difference is only 2 comparing the current value with the last value. It will not do PID control; the difference is 13 (more than death region 10) comparing the current value with the next value, this difference value is larger than control death region value. it will do the PID control with 135.

## 7-4. Auto Tune Mode

If users do not know how to set the PID parameters, they can choose auto tune mode which can find the best control parameters (sampling time, proportion gain **Kp**, integral time **Ti**, differential time **TD**) automatically.

Auto tune mode is suitable for these controlled objects: temperature, pressure; not suitable for liquid level and flow.

Auto-tuning is the process of extracting PID parameters. Sometimes auto-tuning can not find the best parameters at one time. It needs auto-tuning for many times. It is normal that there is a vibration in the process. After the optimum parameters are found at the end of auto-tuning, please switch to the manual PID mode. If the control object is unstable in the process of manual PID, it can not be controlled at a constant target value, which may be caused by the unsatisfactory adjustment of parameters. It is necessary to re-adjust the parameters of PID to achieve stable control.

For step response method: Users can set the sampling cycle to be 0 at the beginning of the auto tune process then modify the value manually in terms of practical needs after the auto tune process is completed.

For step response method: Before doing auto tune, the system should be under the non-control steady state. Take the temperature for example: the measured temperature should be the same to the environment temperature.

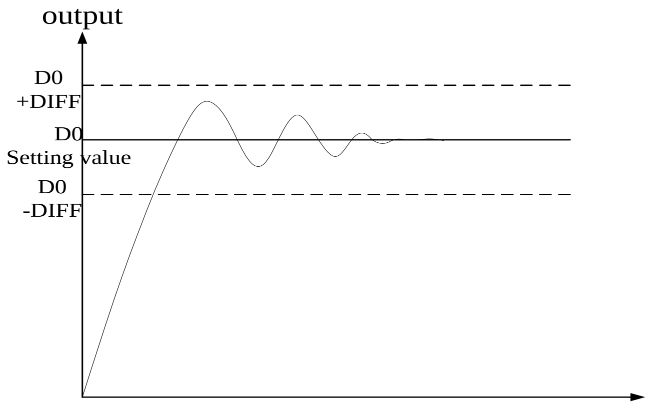
For critical oscillation method: user needs to set the sampling time at the beginning of the auto tune process. For slow response system, 1000ms. For fast response system, 10-100ms.

For critical oscillation method: the system can start the auto tune at any state. For object temperature, the current temperature doesn’t need to be same to ambient temperature.

**Two different methods and PID control diagram:**

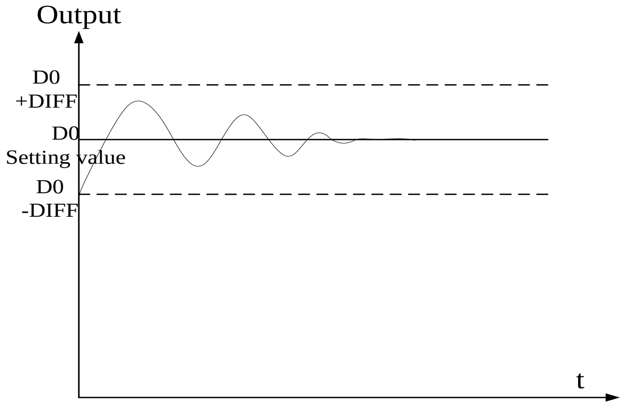
(1) Step response method

Make sure current temperature is equal to ambient temperature




(2) Critical oscillation method

The auto tune start temperature can be any value.




To enter the auto tune mode, please set bit7 of **(S3+ 2)** to be 1 and turn on PID working condition. If bit8 of **(S3+ 2)** turn to 1, it means the auto tune is successful.

**PID auto tune period value [S3+12]**

Set this value in S3+12 during auto tune. This value decides the auto tune performance, in a general way, set this value to be AD result corresponding to one standard tested unit. The default value is 10. The suggested setting range: fall-scale AD result×0.3~1%.

User doesn't need to change this value. However, if the system is interfered greatly by outside, this value should be increased modestly to avoid wrong judgment of positive and negative movement. If this value is too large, the PID control period (sampling time) got from the auto tune process will be too long. As the result do not set this value too large.

※1：If users have no experience, please use the default value 10, set PID sampling time (control period) to be 0msthen start the auto tune.

**PID auto tune overshooting permission setting [S3+13]**

If set 0, overshooting is permitted, and the system can study the optimal PID parameters all the time. But in auto tune process, detected value may be lower or higher than the target value, safety factor should be considered here.

If set 1, overshooting is not permitted. For these objectives which have strict safety demand such as pressure vessel. Set **[S3+13]** to be 1 to prevent from tested value over the target value seriously.

In the process, if **[S3+2]** bit8 changes from 0 to 1, it means the auto tune is successful and the optimal parameters are got; if **[S3+2]** bit8 keeps 0, when **[S3+2]** bit7 changes from 1 to 0, it means auto tune is finished, but the parameters are not the best and they need to be modified by hand.

**Every adjustment percent of current target value in auto tune end transition stage [S3+14]**

This parameter is effective only when **[S3+13]** is 1.

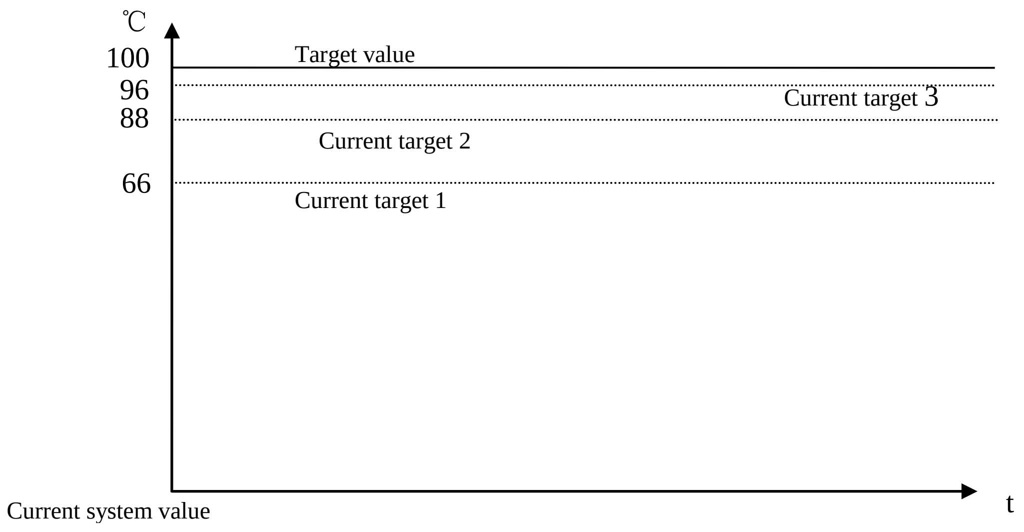
If doing PID control after auto tune, small range of overshooting may be occurred. It is better to decrease this parameter to control the overshooting. But response delay may occur if this value is too small. The defaulted value is 100% which means the parameter is not effective. The recommended range is 50~80%.

**Cutline Explanation:**

Current target value adjustment percent is 2/3（**S3 + 14** = 67%), the original temperature of the system is 0 ℃, target temperature is 100 ℃, and the current target temperature adjustment situation is shown as below:

Next current target value = current target value + (final target value – current target value) × 2/3;

So the changing sequence of current target is 66 ℃, 88 ℃, 96 ℃, 98 ℃, 99 ℃, 100 ℃.




**Over target value times in auto-tuning end transition stage when limiting the overshoot [S3+15]**

This parameter is valid only when [S3+13] is 1;

If entering into PID control directly after auto tune, small range of overshoot may occur. It is good to prevent the overshoot if increasing this parameter properly. But it will cause response lag if this value is too large. The default value is 15 times. The recommended range is from 5 to 20.

## 7-5. Advanced Mode

Users can set some parameters in advanced mode in order to get better PID control effect. Enter into the advanced mode, please set **[S3+2]** bit 15 to be 1, or set it in the XDP Pro software.

Input Filter constant [S3+8]

It will smooth the sampling value. The default value is 0%, which means no filter.

Differential Gain[S3+9]

The low pass filtering process will relax the sharp change of the output value. The default value is 50%; the relaxing effect will be more obviously if increasing this value. Users do not need to change it.

Upper-limit and lower-limit value [S3+10], [S3+11]

Users can choose the analog output range via setting this value.

Default value: lower-limit output =0

Upper-limit =4095

## 7-6. Application outlines

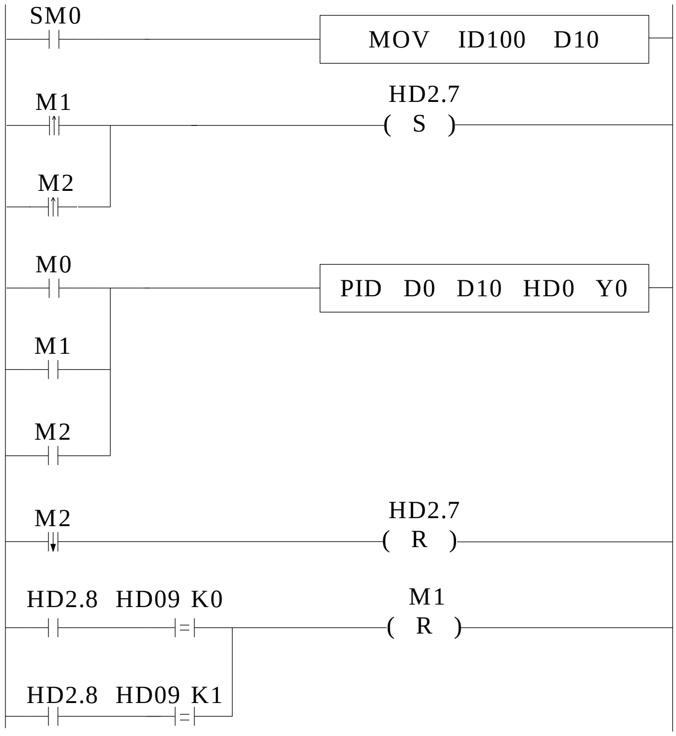
Under the circumstances of continuous output, the system whose effect ability will die down with the change of the feedback value can do auto tune, such as temperature or pressure. It is not suitable for flux or liquid level.
Under the condition of overshooting permission, the system will get the optimal PID parameters from auto tuning.
Under the condition that overshoot not allowed, the PID parameters got from auto tune is up to the target value, it means that different target value will produce different PID parameters which are not the optimal parameters of the system and for reference only.
If the auto tune is not available, users can set the PID parameters according to practical experience. Users need to modify the parameters when debugging. Below are some experience values of the control system for your reference:

- Temperature system: P (%) 2000 ~ 6000, I (minutes) 3 ~ 10, D (minutes) 0.5 ~ 3
- Flux system: P (%) 4000 ~ 10000, I (minutes) 0.1 ~ 1
- Pressure system: P (%) 3000 ~ 7000, I (minutes) 0.4 ~ 3
- Liquid level system: P (%) 2000 ~ 8000, I (minute) 1 ~ 5

## 7-7. Application

Example 1:

PID control program is shown below:




// Move ID100 content into D10

// auto tune mode, or set to autotune mode after auto tune end

// start PID, D0 is target value, D10 is the measured value, from HD0 is PID parameters area; output PID result byY0

// PID control finish, close auto tune PID mode

// if auto tune is successful, and overshoot is permitted, close auto tune control bit, auto tune will finish;

If auto tune turns to be manual mode, and overshoot is not permitted, close auto tune control bit.

**Soft element function comments:**

HD2.7: Auto tune bit

HD2.8: Successful flag of auto tune

M0: Normal PID control

M1: Auto tune control

M2: Enter PID control after auto tune

**Operation steps:**

1. Send the actual temperature to PID collection register
2. Set probably value for P, I, D, sampling period
3. Set ON auto tune control bit M1 to startup PID auto tune
4. M1 will be reset after the auto tune is finished
5. Set ON M0, use the PID parameters getting from auto tune
6. If the PID effect is not good by using the auto tune PID parameters, user can adjust the PID parameters to get good effect.

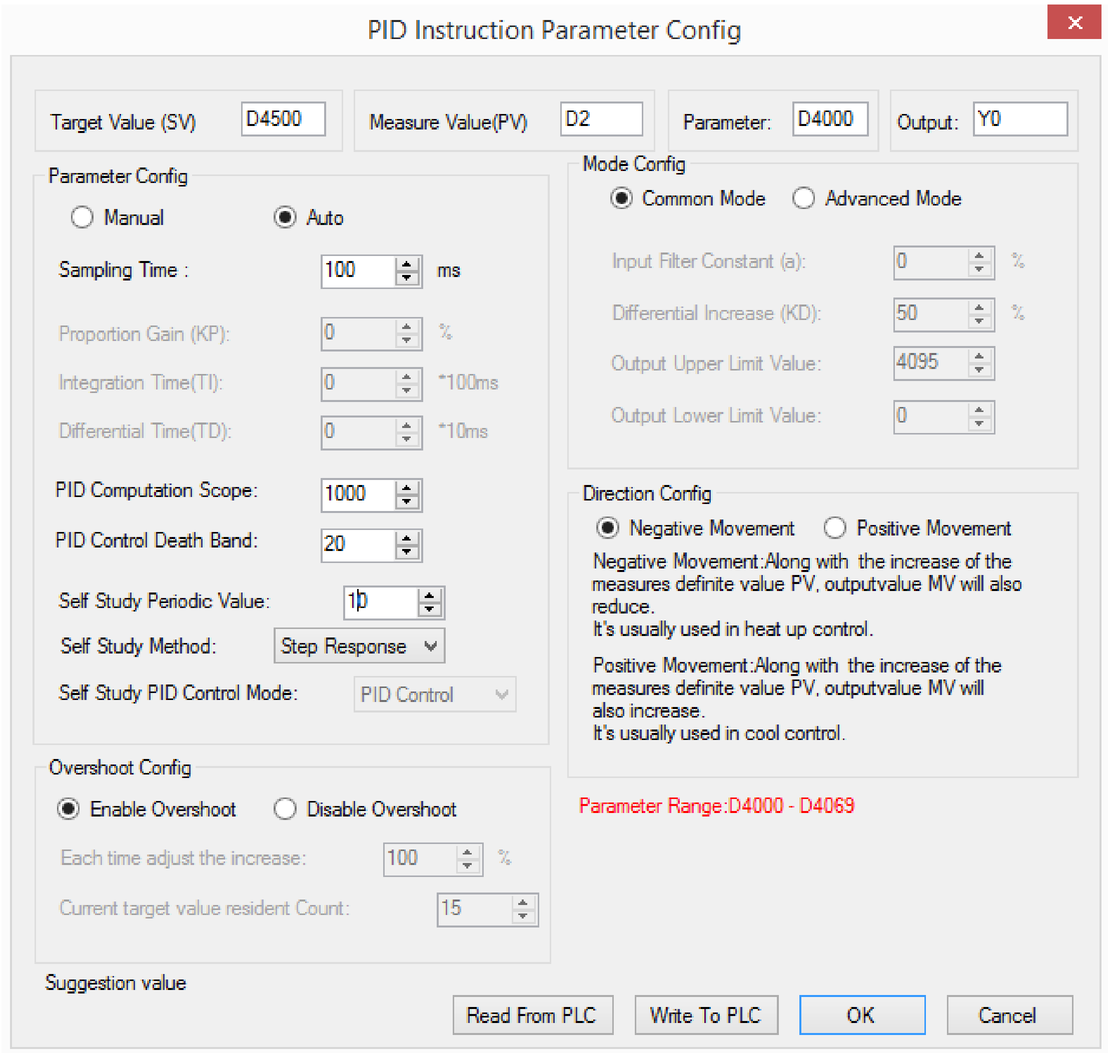
Note: This PLC temperature PID control program is applicable to almost all temperature control projects.

Example 2:

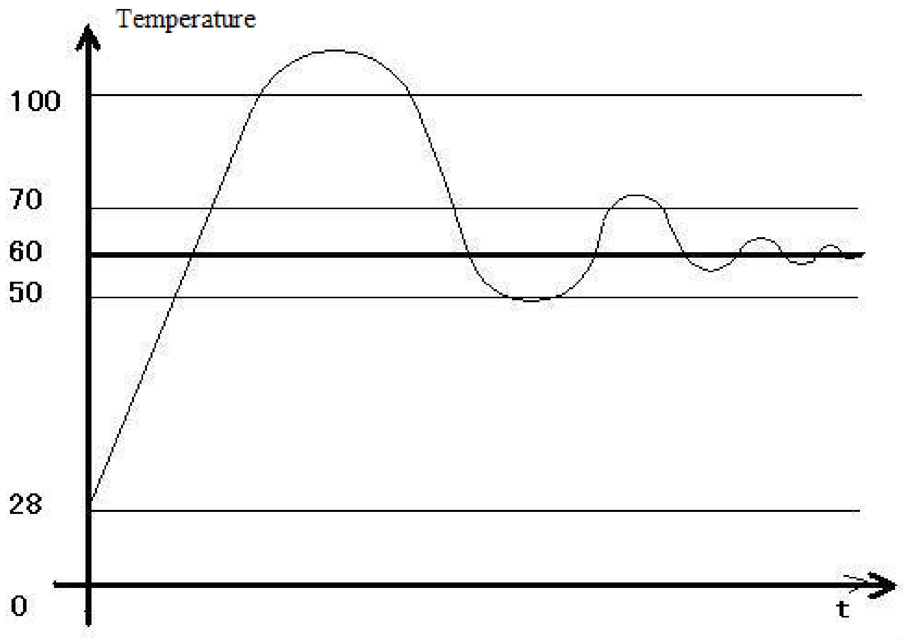
To control the target temperature 60℃ in step response mode.

**Overshoot is permitted:**

1. The target temperature 60℃ (600)
2. Parameters setting




3. The result curve




Explanation:

The target temperature is 60 degree, PID calculation range is 10 degree, PID control dead area is 0.2 degree, auto tune period changing value is 10. When the PID control works in normal atmospheric temperature, the PID output terminal will heat the temperature from 28 to 100 degree, then the output stops, the temperature keeps increasing to 110 degree (max temperature) as the remaining warmth. Then the temperature keeps decreasing to 60 degree, the output starts to heat again to 70 degree and stops. The temperature increases a little then decreases again. This process will repeat. Finally, the temperature will fluctuate close the target temperature.

Note:

1. When the temperature reaches 100 degree and stops heating, the PID start bit D4002.7 will not reset at once, it has delay before reset.
2. When the temperature reaches 100 degree and stops heating, the PID auto tune success bit D4002.8 will be ON at once.
3. When it starts PID calculation, the PLC will auto set a sampling time (about 2500). This parameter will be replaced by the PID best sampling time after stoping heating at 100 degree.
4. When it starts PID calculation, the PLC will auto set the PID parameters (P=4454, I=926, D=2317). These parameters will be replaced by the best PID value after stoping heating at 100 degree.
5. When the temperature reaches 100 degree and stops heating, the PID start bit D4002.7 will not reset at once, it has delay before reset. At this time, the sampling temperature is higher than target temperature. If user sets ON the PID auto tune again, PLC will get all the PID parameters as 0. Please set ON the PID after the temperature decreases under the normal atmospheric temperature.
6. If PID auto tune start bit and auto tune success bit are power-off retentive, please set or reset them propably to avoid calculation error when starting the PLC next time.

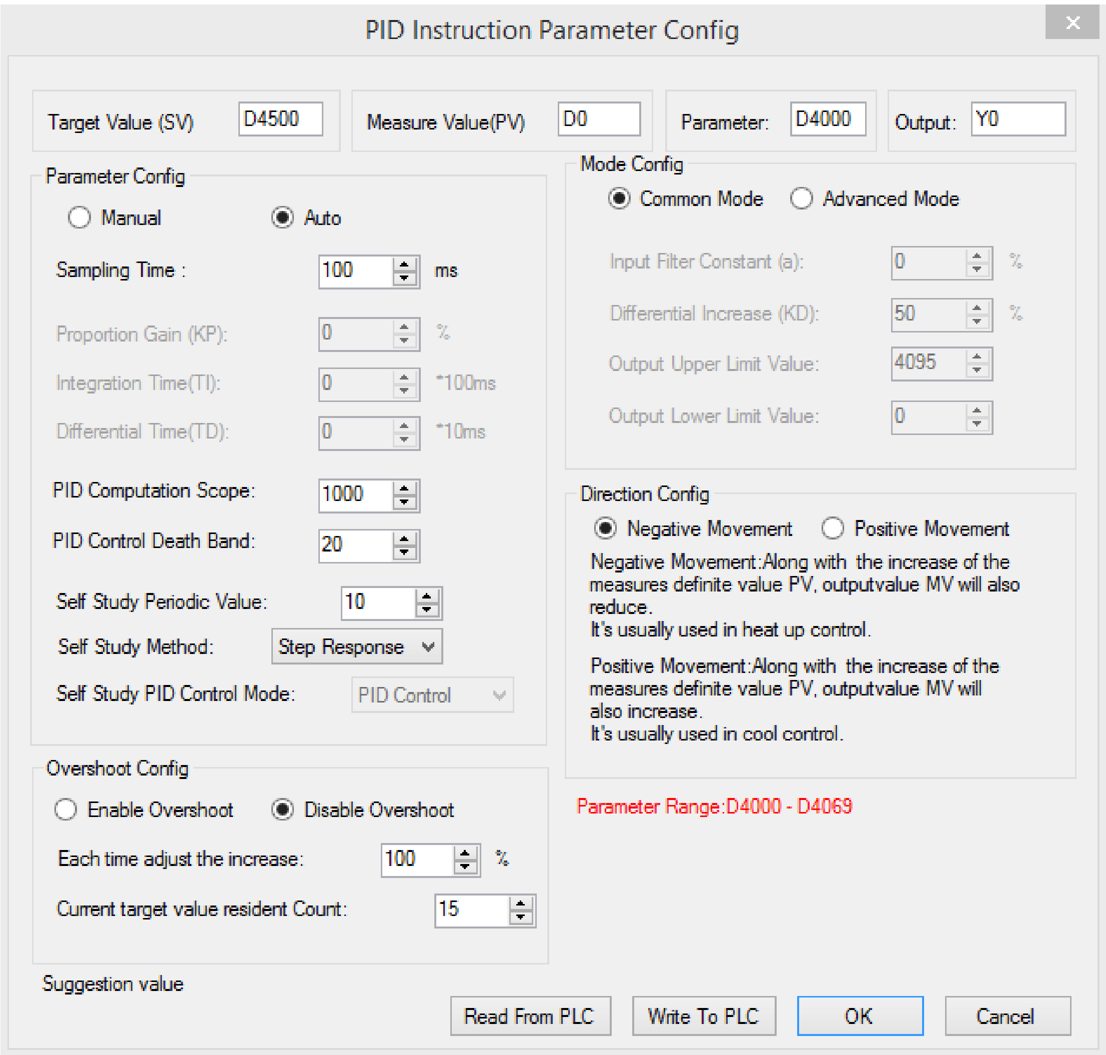
7. The final heating temperature will up to 110 degree when the overshoot is permitted. It is over the target temperature by 50 degree, the overshoot amount is too large.

8. When the PID starts to work, the output will heat the object from 28 degree to 60 degree, then the output is forced to stop heating to avoid overshoot, but this will interrupt the PID auto tune process.

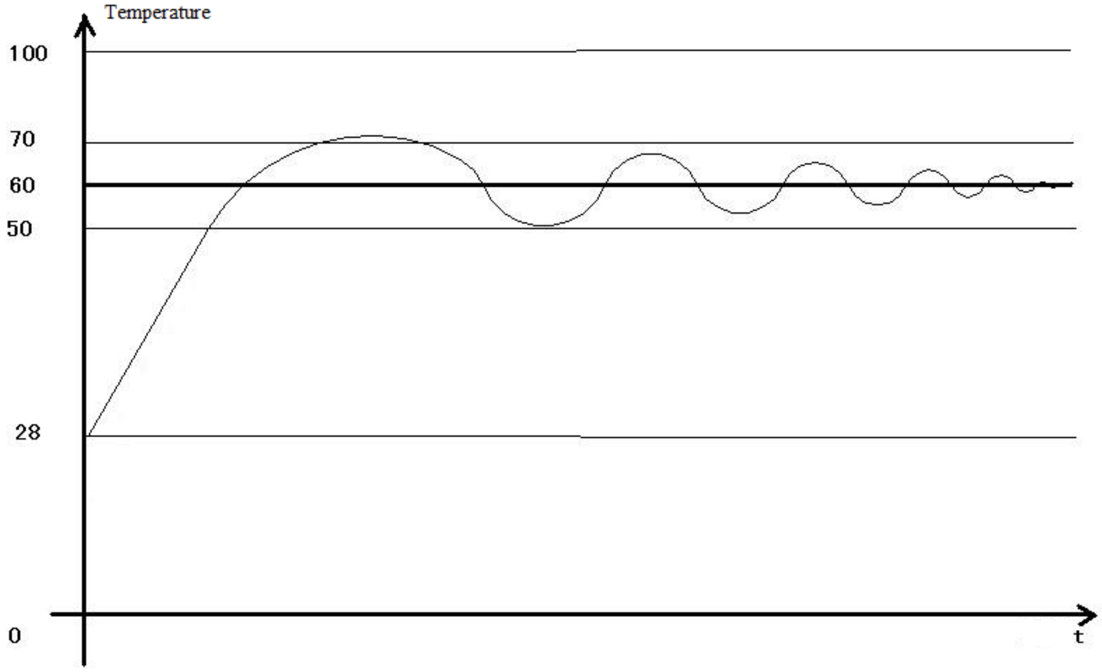
9. To enlarge the PID calculation range can suppress the heating overshoot.

**Overshoot is not permitted:**

1. The target temperature is 60 degree (600)
2. The related parameter settings:




3. The result curve




Explanation:

The target temperature is 60 degree, PID calculation range is 10 degree, PID control dead area is 0.2 degree, auto tune period changing value is 10. When the PID control works in normal atmospheric temperature, the PID output terminal will heat the temperature from 28 to 48 degree, then the output stops, the temperature keeps increasing to 70 degree (max temperature) as the remaining warmth. Then the temperature keeps decreasing to 60 degree, the output starts to heat again to 62 degree and stops. The temperature increases a little (about 64 degree) then decreases again. This process will repeat. Finally, the temperature will fluctuate close the target temperature. The precision is $\pm$ 0.25 degree.

Note:

1. When the temperature reaches 48 degree and stops heating, the PID start bit D4002.7 will not reset at once, it has delay before reset.
2. When the temperature reaches 48 degree and stops heating, the PID auto tune success bit D4002.8 will not be ON at once. It hasn't set ON even when the auto tune succeeded.
3. When it starts PID calculation, the PLC will auto set a sampling time (about 2500). This parameter will be replaced by the PID best sampling time after stoping heating at 48 degree.
4. When it starts PID calculation, the PLC will auto set the PID parameters (P=4454, I=926, D=2317). These parameters will be replaced by the best PID value after stoping heating at 48 degree.
5. When the temperature reaches 48 degree and stops heating, the PID start bit D4002.7 will not reset at once, it has delay before reset. At this time, the sampling temperature is higher than target temperature. If user sets ON the PID auto tune again, PLC will get all the PID parameters as 0. Please set ON the PID after the temperature decreases under the normal atmospheric temperature.
6. If PID auto tune start bit and auto tune success bit are power-off retentive, please set or reset them propably to avoid calculation error when starting the PLC next time.

7. The final heating temperature will up to 70 degree when the overshoot is permitted. It is over the target temperature by 10 degree, the overshoot amount is small.
8. To enlarge the PID calculation range can suppress the heating overshoot.

# 8 C Language Function Block

In this chapter, we focus on C language function block's specifications, edition, instruction calling, application points etc. We also attach the common function list.

## 8-1. Summary

XD, XL supports almost all C language function in XDPPro software (also supports global variable). Users can call the function at many places and call different functions, which greatly increase program security and programmer's efficiency.

## 8-2. Instruction Format

1. Instruction Summary

Call the C language Function Block at the specified place.

| Call the C language function block [NAME_C] | | | |
|---|---|---|---|
| 16 bits instruction | NAME_C | 32 bits Instruction | - |
| Execution condition | Normally ON/OFF, Rising/Falling Edge activation | Suitable Models | XD, XL |
| Hardware | | Software | |

2. Operands

| Operands | Function | Type |
|---|---|---|
| S1 | Name of C Function Block, defined by the user | String |
| S2 | Corresponding start ID of word W in C language function | 16 bits, BIN |
| S3 | Corresponding start ID of word B in C language function | bit, BIN |