API	Mnemonic		Operands	Function	Controllers			
88	D	PID	S1 S2 S3 D	PID control	ES2/EX2	SS2	SA2 SE	SX2

Type / OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	Program Steps
	Bit Devices				Word devices											
S_1													*			PID : 9 steps DPID: 17 steps
S_2													*			
S_3													*			
D													*			

PULSE				16-bit				32-bit			
ES2/EX2	SS2	SA2 SE	SX2	ES2/EX2	SS2	SA2 SE	SX2	ES2/EX2	SS2	SA2 SE	SX2

Operands:

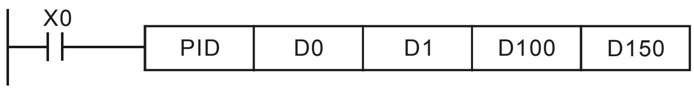
S_1: Set value (SV) **S_2**: Present value (PV) **S_3**: Parameter setting (for 16-bit instruction, uses 20 consecutive devices, for 32-bit instruction, uses 21 consecutive devices) **D**: Output value (MV)

Explanations:

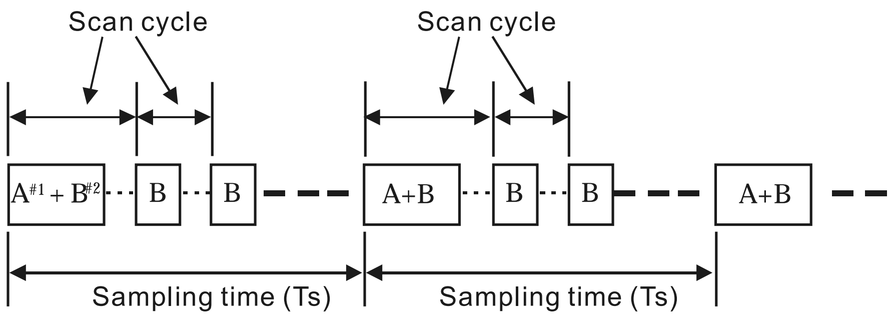
1. This instruction is specifically for PID control. PID operation will be executed only when the sampling time is reached. PID refers to “proportion, integration and derivative”. PID control is widely applied to many mechanical, pneumatic and electronic equipment.
2. After all the parameters are set up, PID instruction can be executed and the results will be stored in **D**. **D** has to be unlatched data register. (If users want to designate a latched data register area, please clear the latched registers to 0 in the beginning of user program.

Program Example:

1. Complete the parameter setting before executing PID instruction.
2. When X0 = ON, the instruction will be executed and the result will be stored in D150. When X0 = OFF, the instruction will not be executed and the previous data in D150 will stay intact.



3. Timing chart of the PID operation (max. operation time is approx. 80us)



Note: #1→ The time for equation calculation during PID operation (approx. 72us)
#2→ The PID operation time without equation calculation (approx. 8us)

Points to note:

1. There is no limitation on the times of using this instruction. However, the register No. designated in **S_3**~ **S_3**+19 cannot be repeated.
2. For 16-bit instruction, **S_3** occupies 20 registers. In the program example above, the area designated in **S_3** is D100 ~ D119.
3. Before the execution of PID instruction, users have to transmit the parameters to the designated register area by MOV instruction. If the designated registers are latched, use MOVP instruction to transmit all parameters only once
4. Settings of **S_3** in the 16-bit instruction:

Device No.	Function	Setup Range	Explanation
S_3:	Sampling time (T_S)	1~2,000 (unit: 10ms)	Time interval between PID calculations and updates of MV. If T_S = 0, PID instruction will not be enabled. If T_S is less than 1 program scan time, PID instruction sets **S_3** as 1 program scan time, i.e. the minimum T_S has to be longer than the program scan time.
S_3+1:	Propotional gain (K_P)	0~30,000(%)	The proportion for magnifying/minifying the error between SV and PV.
S_3+2:	Integral gain (K_I)	0~30,000(%)	The proportion for magnifying/minifying the integral value (The accumulated error). For control mode K0~K8.
	Integral time constant (T_I)	0~30,000 (ms)	For control mode K10
S_3+3:	Derivative gain (K_D)	-30,000~30,000 (%)	The proportion for magnifying/minifying the derivative value (The rate of change of the process error). For control mode K0~K8
	Derivative time constant (T_D)	-30,000~30,000 (ms)	For control mode K10

Device No.	Function	Setup Range	Explanation
S_3+4:	Control mode	0: Automatic control 1: Forward control (E = SV - PV). 2: Reverse control (E = PV - SV). 3: Auto-tuning of parameter exclusively for the temperature control. The device will automatically become K4 when the auto-tuning is completed and K_P, K_I and K_D is set with appropriate value (not avaliable in the 32-bit instruction). 4: Exclusively for the adjusted temperature control (not avaliable in the 32-bit instruction). 5: Automatic mode with MV upper/lower bound control. When MV reaches upper/lower bound, the accumulation of integral value stops. 7: Manual control 1: User set an MV. The accumulated integral value increases according to the error. It is suggested that the control mode should be used in a control environment which change more slowly. DVP-ES2/DVP-EX2/DVP-SS2/DVP-SA2/DVP-SX2 series PLCs whose version is 2.00 (or above), and DVP-SE series PLCs whose version is 1.00 (or above) are supported. 8: Manual control 2: User set an MV. The accumulated integral value will stop increasing. When the control mode becomes the automatic mode (the control mode K5 is used), the instruction PID outputs an appropriate accumulated integral value according to the last MV. DVP-ES2/DVP-EX2/DVP-SS2/DVP-SA2/DVP-SX2 series PLCs whose version is 2.00 (or above), and DVP-SE series PLCs whose version is 1.00 (or above) are supported. 10: T_I / T_D mode: The control changes the integra gain and the differential gain into integral time constant and differential time constant.	
S_3+5:	Tolerable range for error (E)	0~32,767	E = the error between SV and PV. If **S_3** +5 is set as 5, when E is between -5 and 5, E will be 0. When **S_3** +5 = K0, the function will not be enabled.
S_3+6:	Upper bound of output value (MV)	-32,768~32,767	Ex: if **S_3**+6 is set as 1,000, MV will be 1,000 when it exceeds 1,000. **S_3**+6 has to be bigger or equal to **S_3**+7, otherwise the upper bound and lower bound value will switch.
S_3+7:	Lower bound of output value (MV)	-32,768~32,767	Ex: if **S_3**+7 is set as -1,000, MV will be -1,000 when it is smaller than -1,000..
S_3+8:	Upper bound of integral value	-32,768~32,767	Ex: if **S_3**+8 is set as 1,000, the integral value will be 1,000 when it is bigger than 1,000 and the integration will stop. **S_3**+8 has to be bigger or equal **S_3** +9; otherwise the upper bound and lower bound value will switch

Device No.	Function	Setup Range	Explanation
S_3+9:	Lower bound of integral value	-32,768~32,767	Ex: if **S_3**+9 is set as -1,000, the integral value will be -1,000 when it is smaller than -1,000 and the integration will stop.
S_3+10, 11:	Accumulated integral value	Available range of 32-bit floating point	The accumulated integral value is usually for reference. Users can clear or modify it (in 32-bit floating point) according to specific needs.
S_3 +12:	The previous PV	-32,768~32,767	The previous PV is usually for reference. Users can clear or modify it according to specific needs.
S_3+13 ~ **S_3**+19	For system use only..		

3

5. For **S_3**+1~3, when parameter setting exceeds its range, the upper / lower bound will be selected as the set value.
6. If the direction setting (Forward / Reverse) exceeds its range, it will be set to 0.
7. PID instruction can be used in interruption subroutines, step ladders and CJ instruction.
8. The maximum error of sampling time T_S = - (1 scan time + 1ms) ~ + (1 scan time). When the error affects the output, please fix the scan time or execute PID instruction in timer interrupt.
9. PV of PID instruction has to be stable before PID operation executes. If users need to take the value input from AIO modules for PID operation, care should be taken on the A/D conversion time of these modules
10. For 32-bit instruction, **S_3** occupies 21 registers. In the program example above, the area designated in **S_3** will be D100 ~ D120. Before the execution of PID instruction, users have to transmit the parameters to the designated register area by MOV instruction. If the designated registers are latched, use MOVP instruction to transmit all parameters only once.
11. Parameter table of 32-bit **S_3**:

Device No.	Function	Set-point range	Explanation
S_3	Sampling time (T_S)	1~2,000 (unit: 10ms)	Time interval between PID calculations and updates of MV. If T_S= 0, PID instruction will not be enabled. If T_S is less than 1 program scan time, PID instruction sets **S_3** as 1 program scan time, i.e. the minimum T_S has to be longer than the program scan time.

Device No.	Function	Set-point range	Explanation
S_3+1	Proportional gain (K_P)	0~30,000 (%)	The proportion for magnifying/minifying the error between SV and PV.
S_3+2	Integration gain (K_I)	0~30,000 (%)	The proportion for magnifying/minifying the integral value (The accumulated error). For control mode K0~K2, K5.
	Integral time constant (T_I)	0~30,000 (ms)	For control mode K10
S_3+3	Derivative gain (K_D)	-30,000~30,000 (%)	The proportion for magnifying/minifying the derivative value (The rate of change of the process error). For control mode K0~K2, K5.
	Derivative time constant (T_D)	-30,000~30,000 (ms)	For control mode K10
S_3+4	Control mode	0: Automatic control 1: Forward control (E = SV - PV). 2: Reverse control (E = PV - SV). 5: Automatic mode with MV upper/lower bound control. When MV reaches upper/lower bound, the accumulation of integral value stops. 10: T_I / T_D mode with MV upper/lower bound control. When MV reaches upper/lower bound, the accumulation of integral value stops.	
S_3+5, 6	Tolerable range for error (E), 32-bit	0~2,147,483,647	E = the error between SV and PV. If **S_3** +5 is set as 5, when E is between -5 and 5, E will be 0. When **S_3** +5 = K0, the function will not be enabled.
S_3+7, 8	Upper bound of output value (MV) , 32-bit	-2,147,483,648~ 2,147,483,647	Ex: if **S_3**+6 is set as 1,000, MV will be 1,000 when it exceeds 1,000. **S_3**+6 has to be bigger or equal to **S_3**+7, otherwise the upper bound and lower bound value will switch
S_3+9, 10	Lower bound of output value (MV) , 32-bit	-2,147,483,648~ 2,147,483,647	Ex: if **S_3**+7 is set as -1,000, MV will be -1,000 when it is smaller than -1,000.
S_3+11, 12	Upper bound of integral value, 32-bit	-2,147,483,648~ 2,147,483,647	Ex: if **S_3**+8 is set as 1,000, the integral value will be 1,000 when it is bigger than 1,000 and the integration will stop. **S_3**+8 has to be bigger or equal **S_3** +9; otherwise the upper bound and lower bound value will switch.
S_3+13, 14	Lower bound of integral value, 32-bit	-2,147,483,648~ 2,147,483,647	Ex: if **S_3**+9 is set as -1,000, the integral value will be -1,000 when it is smaller than -1,000 and the integration will stop.
S_3+15, 16	Accumulated integral value, 32-bit	Available range of 32-bit floating point	The accumulated integral value is usually for reference. Users can clear or modify it (in 32-bit floating point) according to specific needs.

3

Device No.	Function	Set-point range	Explanation
S_3+17, 18	The previous PV, 32-bit	-2,147,483,648~ 2,147,483,647	The previous PV is usually for reference. Users can clear or modify it according to specific needs.
S_3+19, 20	For system use only.		

12. The explanation of 32-bit **S_3** and 16-bit **S_3** are almost the same. The difference is the capacity of **S_3**+5 ~ **S_3**+20.

PID Equations:

1. When control mode (**S_3**+4) is selected as K0, K1, K2 and K5:
   - In this control mode, PID operation can be selected as Automatic, Forward, Reverse and Automatic with MV upper/lower bound control modes. Forward / Reverse direction is designated in **S_3**+4. Other relevant settings of PID operation are set by the registers designated in **S_3** ~ **S_3**+5.
   - PID equation for control mode k0~k2:

     $$MV = K_P * E(t) + K_I * E(t)\frac{1}{S} + K_D * PV(t)S$$

     where

     MV: Output value

     K_P: Proprotional gain

     $E(t)$: Error value

     $PV(t)$: Present measured value

     $SV(t)$: Target value

     K_D: Derivative gain

     $PV(t)S$: Derivative value of PV(t)

     K_I: Integral gain

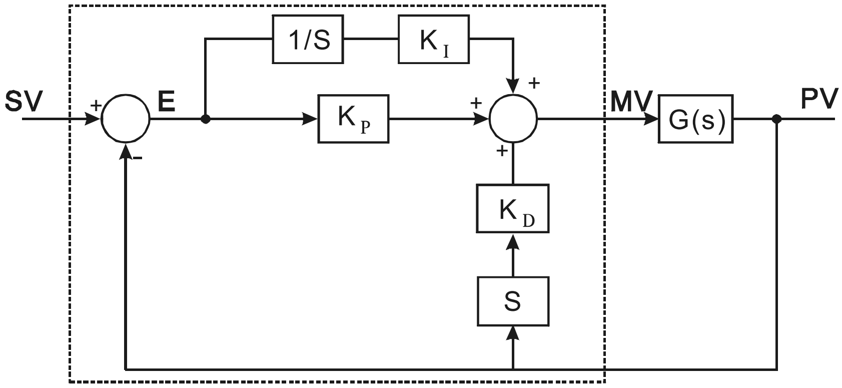
     $E(t)\frac{1}{S}$: Integral value of E(t)

   - When E(t) is smaller than 0 as the control mode is selected as forward or inverse, E(t) will be regarded as "0"

Control mode	PID equation
Forward, automatic	E(t) = SV – PV
Inverse	E(t) = PV – SV

   - Control diagram:
     In diagram below, S is derivative operation, referring to "(PV− previous PV) ÷ sampling time". 1 / S is integral operation, referring to "previous integral value + (error value × sampling time)". G(S) refers to the device being controlled.

PID operation is within dotted area



- The equation above illustrates that this operation is different from a general PID operation on the application of the derivative value. To avoid the fault that the transient derivative value could be too big when a general PID instruction is first executed, our PID instruction monitors the derivative value of the PV. When the variation of PV is excessive, the instruction will reduce the output of MV/.

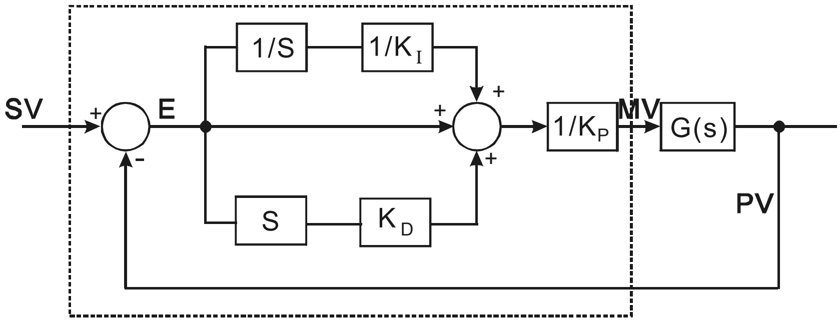
2. When control mode (**S_3**+4) is selected as K3 and K4:
   - The equation is exclusively for temperature control will be modified as:

$$MV = \frac{1}{K_P}\left[E(t) + \frac{1}{K_I}\left(E(t)\frac{1}{S}\right) + K_D * E(t)S\right],$$

   where $\mathrm{E(t) = SV(t) - PV(t)}$

   - Control diagram:
     In diagram below, $1/K_I$ and $1/K_P$ refer to "divided by K_I" and "divided by K_P". Because this mode is exclusively for temperature control, users have to use PID instruction together with GPWM instruction. See **Application 3** for more details

PID operation is within dotted area



   - This equation is exclusively designed for temperature control. Therefore, when the sampling time (T_S) is set as 4 seconds (K400), the range of output value (MV) will be K0 ~ K4,000 and the cycle time of GPWM instruction used together has to be set as 4 seconds (K4000) as well.

- If users have no idea on parameter adjustment, select K3 (auto-tuning). After all the parameters are adjusted (the control direction will be automatically set as K4), users can modify the parameters to better ones according to the adjusted results.

3. When control mode (**S_3**+4) is selected as K10:
   - **S_3**+2 (K_I) and **S_3**+3 (K_D) in this mode will be switched to parameter settings of Integral time constant (T_I) and Derivative time constant (T_D).
   - When output value (MV) reaches the upper bound, the accumulated integral value will not increase. Also, when MV reaches the lower bound, the accumulated integral value will not decrease.
   - The equation for this mode will be modified as:

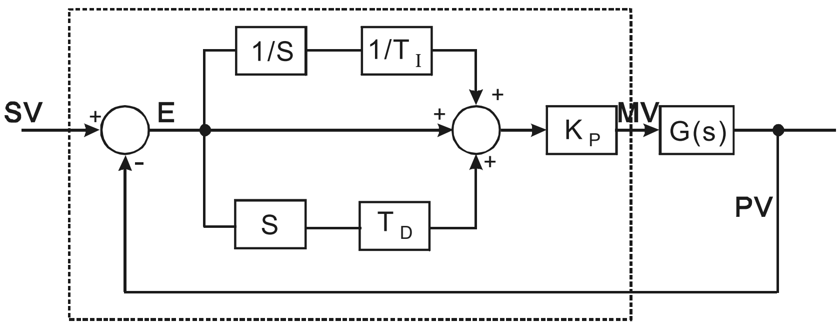
$$MV = K_P \times \left[E(t) + \frac{1}{T_I}\int E(t)dt + T_D \frac{d}{dt}E(t) \right]$$

Where

$$E(t) = SV(t) - PV(t)$$

Control diagram:

PID operation is within dotted area



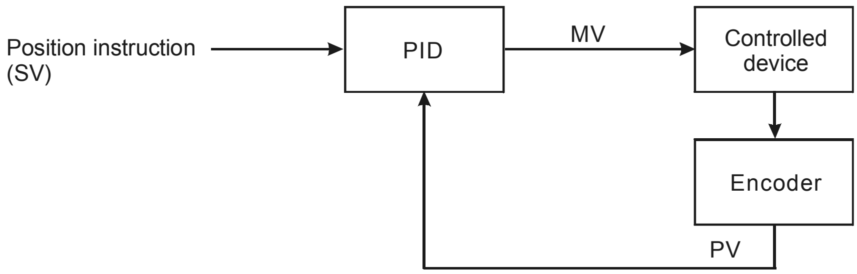
Notes and suggestion:

1. **S_3** + 3 can only be the value within 0 ~ 30,000.
2. There are a lot of circumstances where PID instruction can be applied; therefore, please choose the control functions appropriately. For example, when users select parameter auto-tuning for the temperature (**S_3** + 4 = K3), the instruction can not be used in a motor control environment otherwise improper control may occur.
3. When you adjust the three main parameters, K_P, K_I and K_D (**S_3** + 4 = K0 ~ K2), please adjust K_P first (according to your experiences) and set K_I and K_D as 0. When the output can roughly be controlled, proceed to increase K_I and K_D (see example 4 below for adjustment methods). K_P = 100 refers to 100%, i.e. the proportional gain to the error is 1. K_P < 100% will decrease the error and K_P > 100% will increase the error
4. When temperature auto-tuning function is selected(**S_3** + 4 = K3, K4), it is suggested that store the parameters in D register in latched area in case the adjusted parameters will disappear after

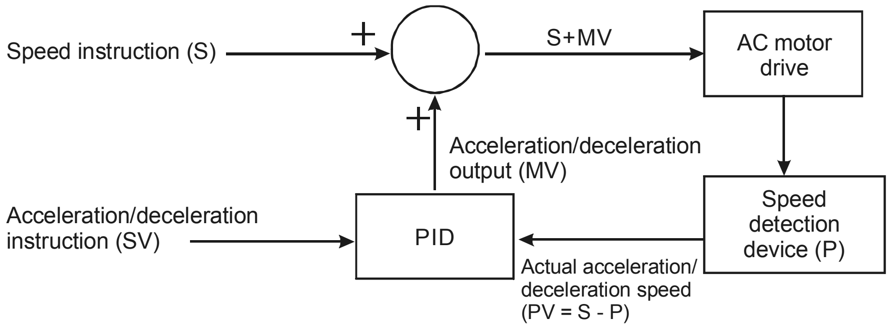
the power is cut off. There is no guarantee that the adjusted parameters are suitable for every control requirement. Therefore, users can modify the adjusted parameters according to specific needs, but it is suggested to modify only K_I or K_D.

5. PID instruction has to be controlled with many parameters; therefore care should be taken when setting each parameter in case the PID operation is out of control.

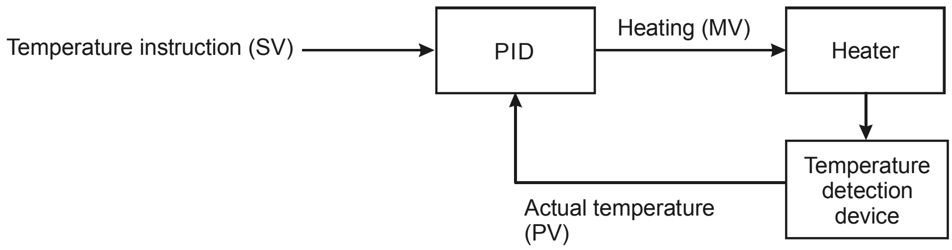
Example 1: Block diagram of application on positioning (**S_3**+4 = 0)



Example 2: Block diagram of application on AC motor drive (**S_3**+4 = 0)



Example 3: Block diagram of application on temperature control (**S_3**+4 = 1)

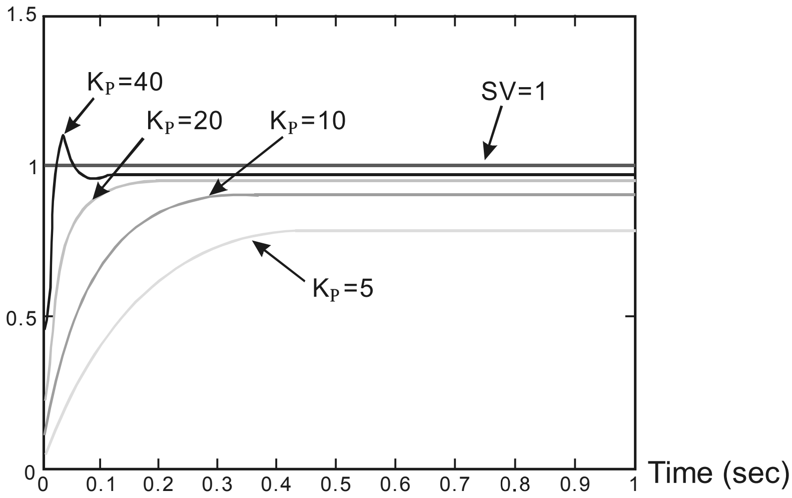


Example 4: PID parameters adjustment

Assume that the transfer function of the controlled device G(S) in a control system is a first-order function $G(s)=\frac{b}{s+a}$ (model of general motors), SV = 1, and sampling time (T_S) = 10ms. Suggested steps for adjusting the parameters are as follows:

Step1:

Set K_I and K_D as 0, and K_P as 5, 10, 20, 40. Record the SV and PV respectively and the results are as the figure below.



Step 2:

When K_P is 40, response overshoot occurs, so we will not select it.

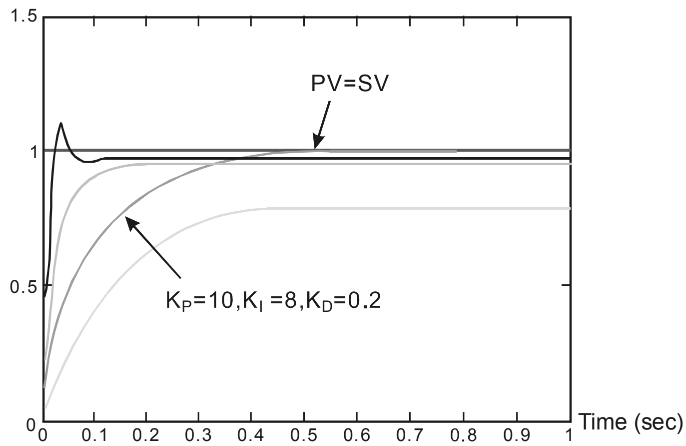
When K_P is 20, PV response is close to SV and won't overshoot, but transient MV will be to large due to a fast start-up. We can put it aside and observe if there are better curves.

When K_P is 10, PV response is close to SV and is smooth. We can consider using it.

When K_P is 5, the response is too slow. So we won't use it.

Step 3:

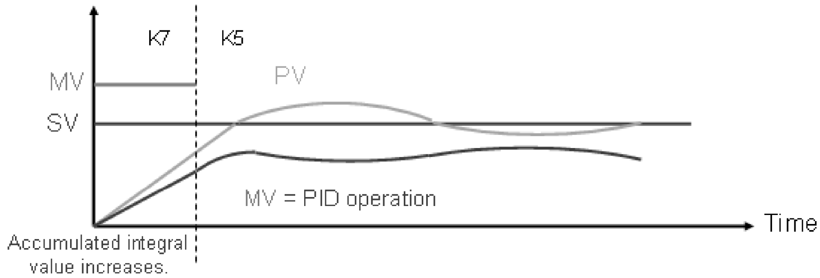
Select K_P = 10 and increase K_I gradually, e.g. 1, 2, 4, 8. K_I should not be bigger than K_P. Then, increase K_D as well, e.g. 0.01, 0.05, 0.1, 0.2. K_D should not exceed 10% of K_P. Finally we obtain the figure of PV and SV below.



Note: The example is only for reference. Users have to adjust parameters according to the condition of the actual control system.

Example 5: Transition between the manual mode (K7) and the automatic mode (K5)

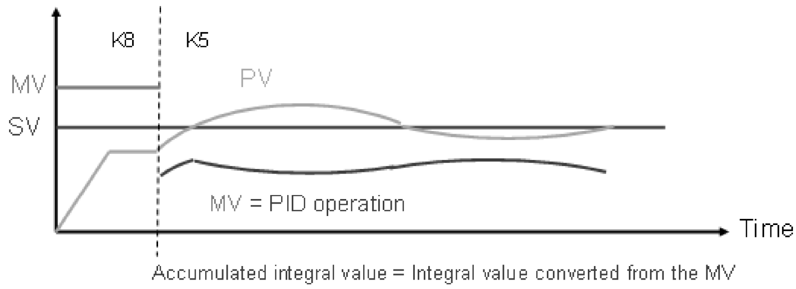
If the setting of the PID parameters is complete, and the control mode is the manual mode (K7), the control curve will be as shown below.



If the control mode becomes the automatic mode (K5), the output value MV changes from the output value set by users to the output value of the PID operation.

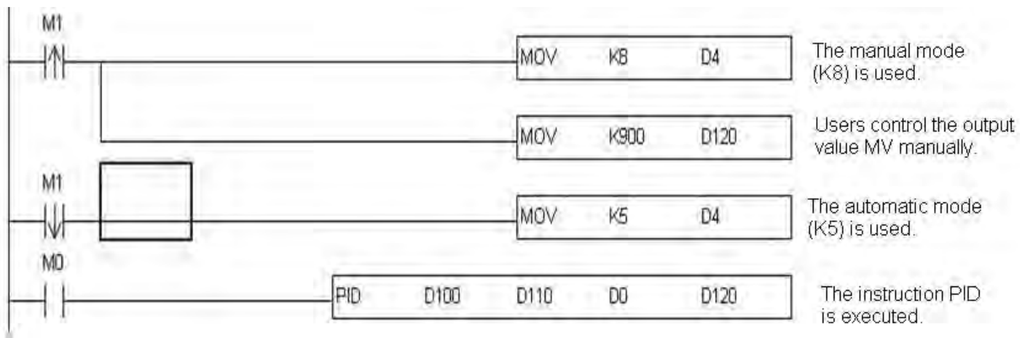
Example 6: Transition between the manual mode (K8) and the automatic mode (K5)

If the setting of the PID parameters is complete, and the control mode is the manual mode (K8), the control curve will be as shown below.



If the control mode becomes the automatic mode (K5), the accumulated integral value will be the integral value converted from the last MV, and the accumulated integral value will be converted into the output value of the PID operation.

The program for example 5 and program 6 are shown below. In the figure below, ,M0 is a flag for enabling the instruction PID. When M1 is On, the manual mode is used. When M1 is Off, the automatic mode is used.



Application 1:

PID instruction in pressure control system. (Use block diagram of example 1)

Control purpose:

Enabling the control system to reach the target pressure.

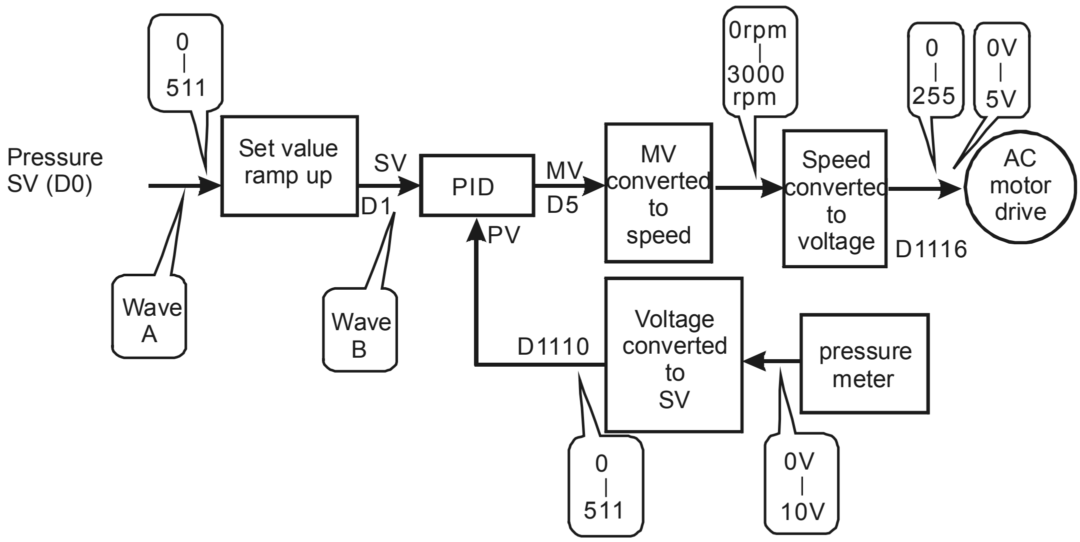
Control properties:

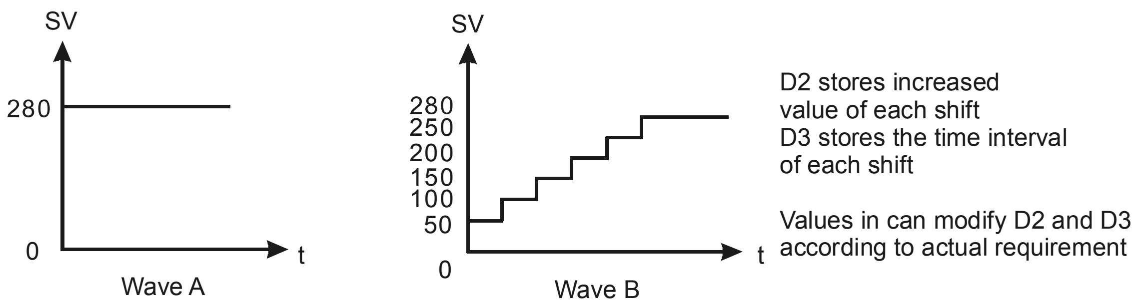
The system requires a gradual control. Therefore, the system will be overloaded or out of control if the process progresses too fast.

Suggested solution:

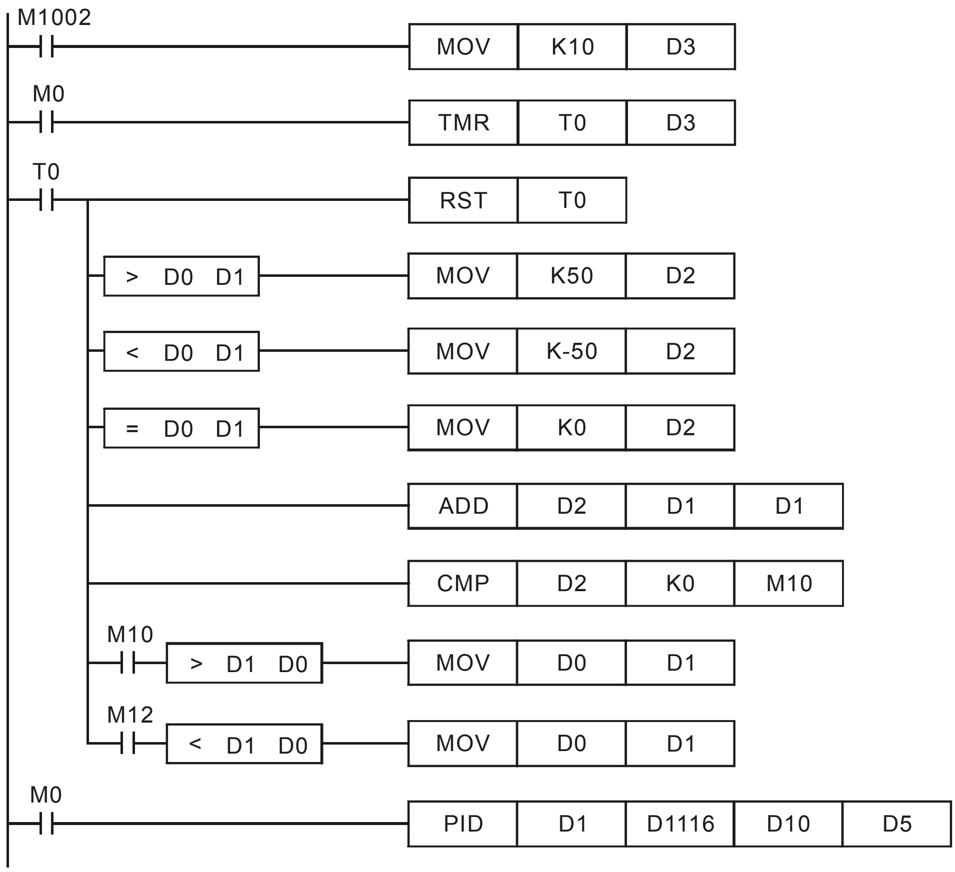
Solution 1: Longer sampling time

Solution 2: Using delay instruction. See the figure below





Example program of SV ramp up function:



Application 2:

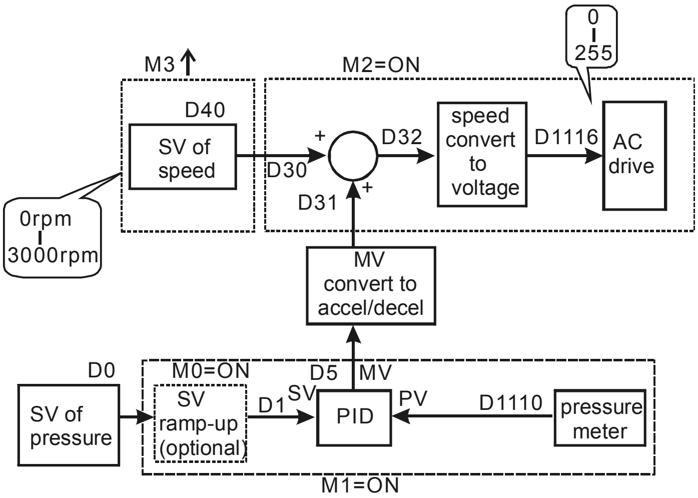
Speed control system and pressure control system work individually (use diagram of Example 2)

Control purpose:

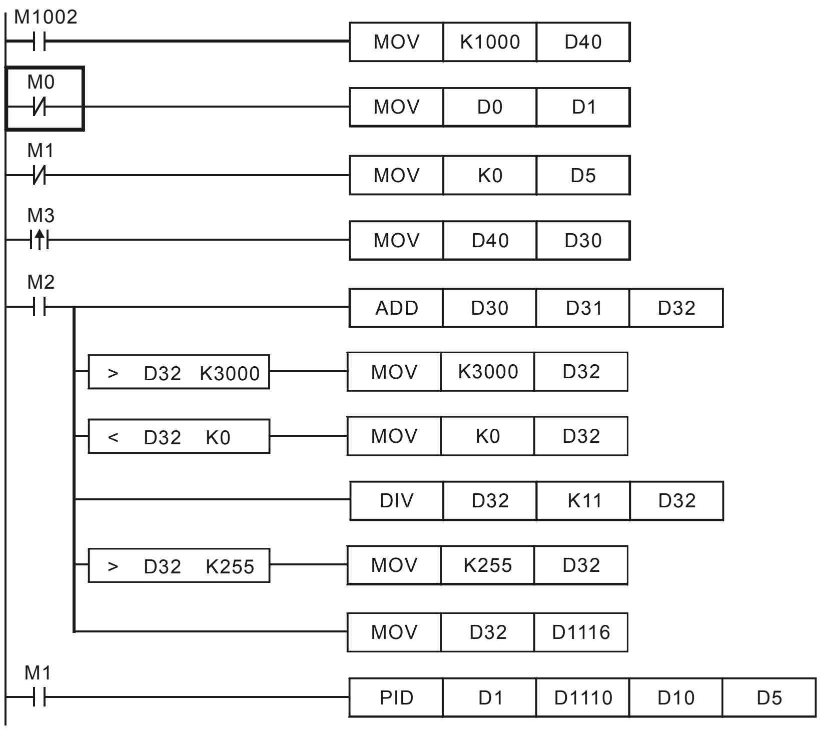
After the speed control operates in open loop for a period of time, adding pressure control system (PID instruction) to perform a close loop control.

Control properties:

Since the speed and pressure control systems are not interrelated, we have to structure an open loop for speed control first following by a close loop pressure control. If users afraid that the pressure control system changes excessively, consider adding the SC ramp-up function illustrated in **Application 1** into this control. See the control diagram below.



Part of the example program:



Application 3:

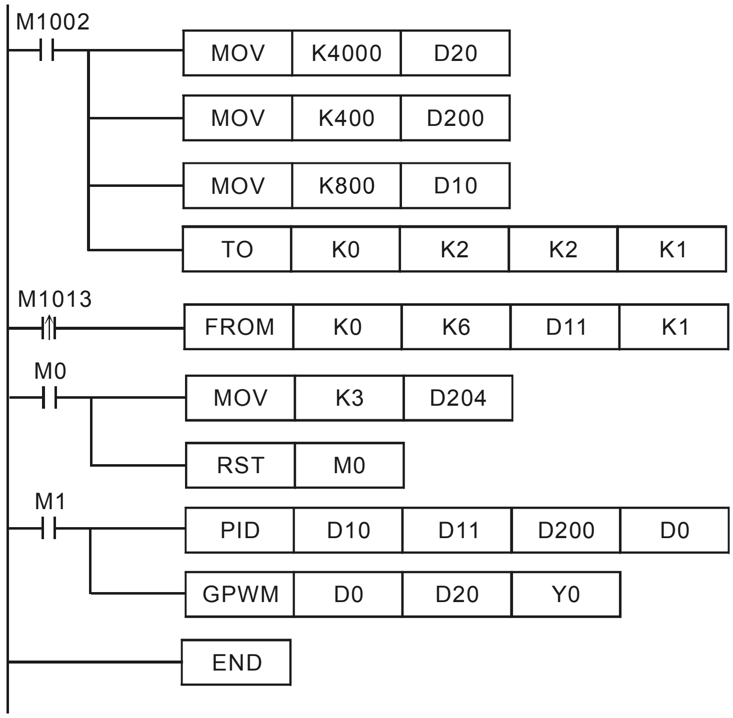
Using auto-tuning for temperature control

<u>Control purpose:</u>

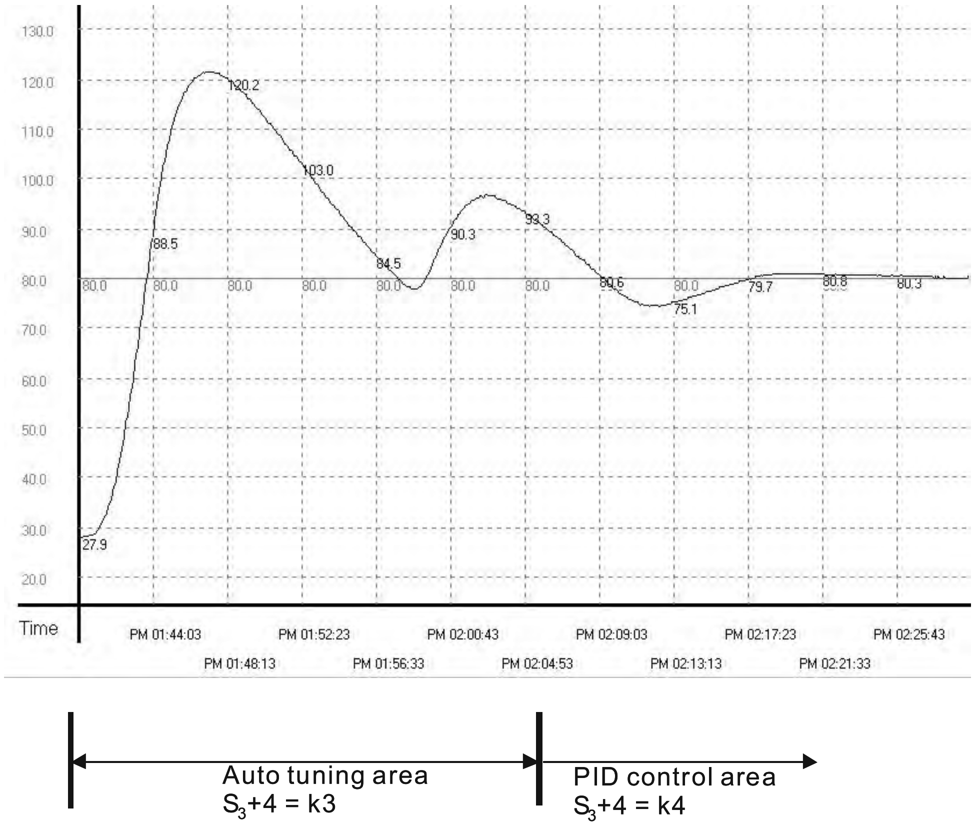
Calculating optimal parameter of PID instruction for temperature control

<u>Control properties:</u>

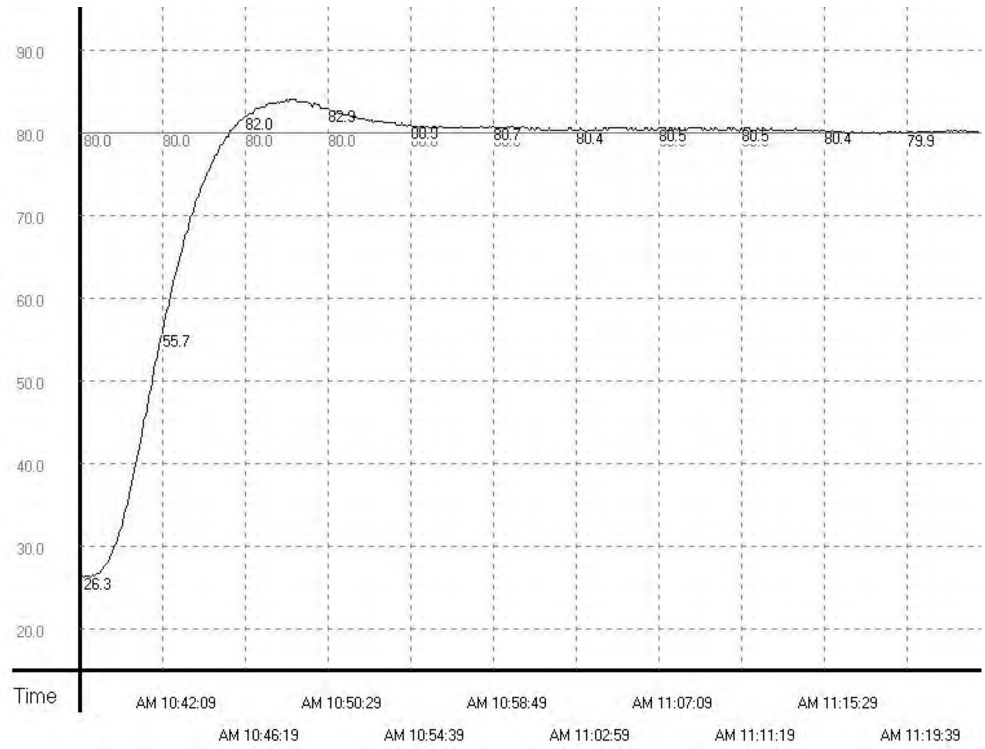
Users may not be familiar with a new temperature environment. In this case, selecting auto-tuning (**S₃**+4 = K3) for an initial adjustment is suggested. After initial tuning is completed, the instruction will auto modify control mode to the mode exclusively for adjusted temperature (**S₃**+4 = K4). In this example, the control environment is a heating oven. See the example program below.



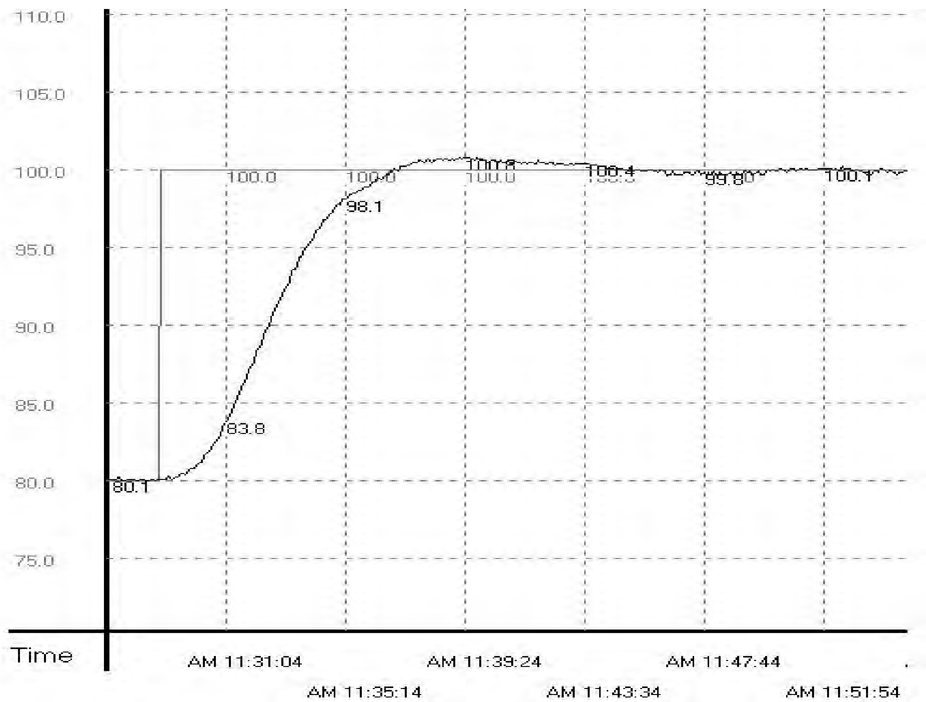
Results of initial auto-tuning



Results of using adjusted parameters generated by initial auto-tuning function.



From the figure above, we can see that the temperature control after auto-tuning is working fine and it spent only approximately 20 minutes for the control. Next, we modify the target temperature from 80°C to 100°C and obtain the result below.



From the result above, we can see that when the parameter is 100°C, temperature control works fine and costs only 20 minutes same as that in 80°C.

API	Mnemonic	Operands	Function	Controllers			
				ES2/EX2	SS2	SA2 SE	SX2
89	PLS	(S)	Rising-edge output				

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	PLS: 3 steps
S		*	*													

PULSE				16-bit				32-bit			
ES2/EX2	SS2	SA2 SE	SX2	ES2/EX2	SS2	SA2 SE	SX2	ES2/EX2	SS2	SA2 SE	SX2

Operands:

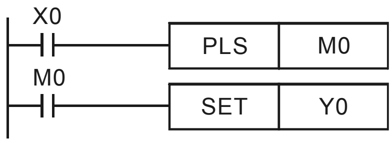
S: Rising pulse output device

Explanations:

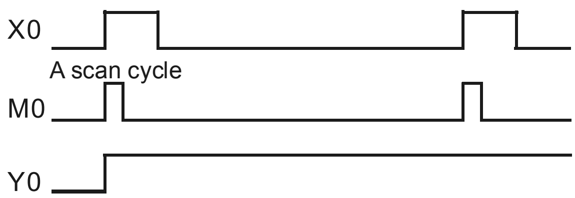
When X0 goes from OFF to ON (Rising-edge trigger), PLS instruction executes and **S** generates a cycle pulse for one operation cycle.

Program Example:

Ladder Diagram:



Timing Diagram:



Instruction Code:

		Operation:
LD	X0	; Load NO contact of X0
PLS	**M0**	; M0 rising-edge output
LD	M0	; Load NO contact of M0
SET	Y0	; Y0 latched (ON)