Topic: The PID function (Examples of temperature control of an oven)

Applicable model	DVP-EH3 series, DVP-SV2 series, DVP-EH2 series, DVP-SV series, DVP-ES2/EX2 series, DVP-SX2 series, DVP-SX series, DVP-SS2 series, DVP-SE series, DVP-10MC series, DVP-SA2 series, TP04P series, TP70P series
Keyword	PID control

Table of Contents

1. Preface and Purpose

Preface:
Proportional-integral-derivative controller (PID controller) is widely applied in the field of engineering. It has been presented for nearly sixty years. It is a main technical tool used in industrial control systems because its structure is simple, it is stable and reliable, and it can be easily adjusted.

Purpose:
If users use PID control for the first time, they may not be familiar with the characteristics of the PID control. The document helps users understand the principle and the usage of PID control.

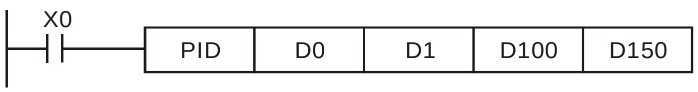
2. PID Control Mode 0~PID Control Mode 5

API	Mnemonic			Operand	Function
88	D	PID		S_1 S_2 S_3 D	PID control

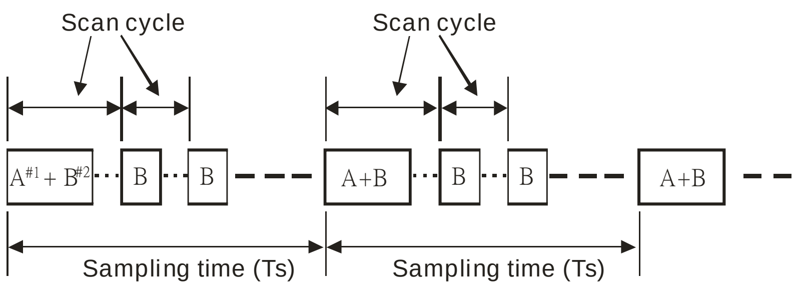
Operands:
S_1: Set value (SV); **S_2**: Present value (PV); **S_3**: Parameter (Twenty consecutive devices will be occupied if the 16-bit instruction is used. Twenty-one devices will be used if the 32-bit instruction is used.); **D**: Output value (MV)

Explanation:
(1) The instruction is used to implement a PID algorithm. After the sampling time set is reached, the PID algorithm will be implemented. PID stands for Proportional, Integral, Derivative. PID control is widely applied to mechanical equipment, pneumatic equipment, and electronic equipment
(2) After all parameters are set, the instruction PID can be executed, and the result will be stored in **D**. **D** has to be an unretentive data register. (If users want to designate a retentive data register, the value in the retentive data register need to be cleared at the beginning of the program created.)

Example:
(1) Before the instruction PID is executed, the parameters of PID need to be set.
(2) If X0 is ON, the instruction will be executed, and the result will be stored in D150. If X0 is OFF, the instruction will not be executed, and the previous data in D150 will remain unchanged.



(3) Timing diagram for the instruction PID (The maximum operation time is 80 us.)



Note: #1→ Time it takes for the algorithm to be calculated during the execution of PID (approximately 72 us)
#2→ Time it takes for PID to be executed without the performance of the algorithm (approximately 8 us)

Additional remarks:

(1) There is no limitation on the number of times of PID can be used. However, the registers designated by $\mathbf{S_3}$~$\mathbf{S_3}$+19 need to be unique.

(2) $\mathbf{S_3}$ will occupy twenty consecutive devices if the 16-bit instruction is used. In the example above, $\mathbf{S_3}$ occupies D100~D119.

(3) Before the 16-bit instruction is executed, users have to use MOV to transmit values to the registers designated by the parameters. If the registers designated by the parameters are retentive registers, the users can transmit the values by executing MOVP once.

(4) $\mathbf{S_3}$ in the 16-bit instruction:

Device No.	Function	Setting range	Explanation
$\mathbf{S_3}$	Sampling time (T_S)	1~2,000 (unit: 10 ms)	A sampling time is the time interval between the performance of a PID algorithm and the updating of an MV. If the T_S set is 0, PID will not be excuted. If the T_S set is less than one program scan, PID will regard $\mathbf{S_3}$ as one program scan, i.e. the minimum T_S set needs to be longer than one program scan.
$\mathbf{S_3}$+1	Propotional gain (K_P)	0~30,000 (%)	It is the magnified proportional value of the error between an SV and a PV.
$\mathbf{S_3}$+2	Integral gain (K_I)	0~30,000 (%)	For control mode 0~control mode 8
	Integral time constant (T_I)	0~30,000 (ms)	For control mode 10
$\mathbf{S_3}$+3	Derivative gain (K_D)	-30,000~30,000 (%)	For control mode 0~control mode 8
	Derivative time constant (T_D)	-30,000~30,000 (ms)	For control mode 10
$\mathbf{S_3}$+4	Control mode	0: Automatic control 1: Forward control (E=SV-PV). 2: Reverse control (E=PV-SV). 3: Parameters are adjusted automatically for temperature control. After the adjustment of the parameters is complete, the value will be changed to K4 automatically, and the approriate parameters K_P, K_I and K_D will be calculated. 4: Temperature control which has been adjusted (not avaliable in the 32-bit instruction) 5: It is an automatic mode. If an MV reaches the upper/lower limit, the accumulation of an integral value will stop. 7: Manual control 1: User set an MV by themselves. The accumulated integral value in PID control increases according to the error. It is suggested that the control mode should be used in a control environment which changes more slowly. 8: Manual control 2: User set an MV by themselves. The accumulated integral value in PID control stops increasing. After the control mode becomes the automatic mode (control mode 5), PID will output an appropriate accumulated integral value according to the last MV.	
$\mathbf{S_3}$+5	Range within which an error (E) is count as 0	0~32,767	An error (E) is the difference between an SV and a PV. After $\mathbf{S_3}$+5 is set to 5, an E will be count as 0 if it is between -5 and 5. If $\mathbf{S_3}$+5 is set to K0, the function will not be enabled.
$\mathbf{S_3}$+6	Maximum output value (MV)	-32,768~32,767	Example: After $\mathbf{S_3}$+6 is set to 1,000, an MV will be 1,000 if it exceeds 1,000. $\mathbf{S_3}$+6 has to be greater than $\mathbf{S_3}$+7, otherwise the maximum output value set and the minimum output value set will be interchanged.

Device No.	Function	Setting range	Explanation
S_3+7	Minimum output value (MV)	-32,768~32,767	Example: After **S_3**+7 is set to -1,000, an MV will be -1,000 if it is less than than -1,000.
S_3+8	Maximum integral value	-32,768~32,767	Example: After **S_3**+8 is set to 1,000, an integral value will be 1,000 if it is greater than 1,000, and the integration will stop. **S_3**+8 has to be greater than or equal to **S_3**+9, otherwise the maximum integral value set and the minimum integral value set will be interchanged.
S_3+9	Minimum integral value	-32,768~32,767	Example: After **S_3**+9 is set as -1,000, an integral value will be -1,000 if it is less than than -1,000, and the integration will stop.
S_3+10, 11	Accumulated integral value	Available 32-bit floating point value range	An accumulated integral value is usually for reference. Users can clear or modify it according to specific needs. (Only 32-bit floating point value can be used.)
S_3+12	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		

(5) If **S_3**+1~3 exceed the maximum/minimum values allowed, the maximum/minimum values will be used.
(6) If the direction set (forward/reverse direction) exceeds the range allowed, it will be set to 0.
(7) PID can be used in an interrupt subroutine, a step, or CJ.
(8) Maximum error of the sampling time (T_S) set = - (1 program scan + 1 ms) ~ + (1 program scan). If the error of the sampling time (T_S) set affects output, please fix a scan cycle, or execute PID instruction in a timer interrupt subroutine.
(9) The PV of PID needs to be stable before a PID algorithm is performed. If users need to take an input value from a special module for the performance of a PID algorithm, they have to note the A/D conversion time of this module.
(10) **S_3** will occupy twenty-one consecutive devices if the 32-bit instruction is used. In the example above, if the 32-bit instruction is used, **S_3** will occupy D100~D120.
(11) Before the 32-bit instruction is executed, users have to use MOV to transmit values to the registers designated by the parameters. If the registers designated by the parameters are retentive registers, the users can transmit the values by executing MOVP once.
(12) **S_3** in the 32-bit instruction:

Device No.	Function	Setup Range	Explanation
S_3	Sampling time (T_S)	1~2,000 (unit: 10 ms)	A sampling time is the time interval between the performance of a PID algorithm and the updating of an MV. If the T_S set is 0, PID will not be excuted. If the T_S set is less than one program scan, PID will regard **S_3** as one program scan, i.e. the minimum T_S set needs to be longer than one program scan.
S_3+1	Propotional gain (K_P)	0~30,000 (%)	It is the magnified proportional value of the error between an SV and a PV.
S_3+2	Integral gain (K_I)	0~30,000 (%)	For control mode 0~control mode 2, and control mode 5
	Integral time constant (T_I)	0~30,000 (ms)	For control mode 10
S_3+3	Derivative gain (K_D)	-30,000~30,000 (%)	For control mode 0~control mode 2, and control mode 5
	Derivative time constant (T_D)	-30,000~30,000 (ms)	For control mode 10

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). 5: It is an automatic mode. If an MV reaches the upper/lower limit, the accumulation of an integral value will stop.	
S_3+5, 6	Range within which an error (E) is count as 0	0~2,147,483,647	An error (E) is the difference between an SV and a PV. After (**S_3**+6, **S_3**+5) is set to 5, an E will be count as 0 if it is between -5 and 5. If (**S_3**+6, **S_3**+5) is set to K0, the function will not be enabled.
S_3+7, 8	Maximum output value (MV)	-2,147,483,648~ 2,147,483,647	Example: After (**S_3**+8, **S_3**+7) is set to 1,000, an MV will be 1,000 if it exceeds 1,000. (**S_3**+8, **S_3**+7) has to be greater than (**S_3**+10, **S_3**+9), otherwise the maximum output value set and the minimum output value set will be interchanged.
S_3+9, 10	Minimum output value (MV)	-2,147,483,648~ 2,147,483,647	Example: After (**S_3**+10, **S_3**+9) is set to -1,000, an MV will be -1,000 if it is less than than -1,000.
S_3+11, 12	Maximum integral value	-2,147,483,648~ 2,147,483,647	Example: After (**S_3**+12, **S_3**+11) is set to 1,000, an integral value will be 1,000 if it is greater than 1,000, and the integration will stop. (**S_3**+10, **S_3**+9) has to be greater than or equal to (**S_3**+14, **S_3**+13), otherwise the maximum integral value set and the minimum integral value set will be interchanged.
S_3+13, 14	Minimum integral value	-2,147,483,648~ 2,147,483,647	Example: After (**S_3**+14, **S_3**+13) is set as -1,000, an integral value will be -1,000 if it is less than than -1,000, and the integration will stop.
S_3+15, 16	Accumulated integral value	Available 32-bit floating point value range	An accumulated integral value is usually for reference. Users can clear or modify it according to specific needs. (Only 32-bit floating point value can be used.)
S_3+17,18	Previous PV	-	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		

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

PID algorithm:

(1) **S_3**+4 is K0/K1/K2/K5.

- Automatic control, forward control or reverse control is used. Forward/Reverse control is designated by **S_3**+4. Other settings relevant to a PID algorithm are set by the registers designated by **S_3**~**S_3**+5.

PID algorithm:

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

$PV(t)S$ is the derivative value of $PV(t)$, and $E(t)\frac{1}{S}$ is the integral value of $E(t)$. If $E(t)$ is less than 0 as forward control or reverse control is selected, $E(t)$ will be regarded as "0".

Control mode	PID algorithm
Forward, automatic	$E(t) = SV(t) - PV(t)$
Reverse	$E(t) = PV(t) - SV(t)$

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

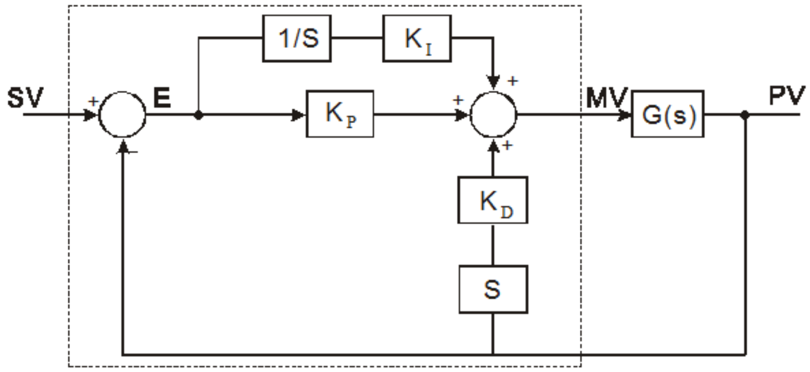


Figure 1 Control diagram for control mode 0/1/2/5 (**S_3**+4 is K0/K1/K2/K5.)

- The algorithm above illustrates that this instruction is different from a general PID instruction in the application of a derivative value. To avoid the fault that a transient derivative value can be too big when a general PID instruction is first executed, this PID instruction monitors the derivative value of a PV. When the variation of a PV is excessive, the instruction will reduce the output of an MV.
- Descriptions of symbols:

MV : Output value

K_P : Proportional gain

$E(t)$: Error

PV : Present value

SV : 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 $E(t)$

(2) **S_3**+4 is K3/K4.
The algorithm for temperature control is shown below.

$$MV = \frac{1}{K_P}\left[E(t) + \frac{1}{K_I}\left(E(t)\frac{1}{S}\right) + K_D * E(t)S\right] \text{ where } \mathrm{E(t) = SV(t) - PV(t)}.$$

- Control diagram: In Figure 2, $1/K_I$ and $1/K_P$ refer to “divided by K_I” and “divided by K_P”. Because this control diagram is for temperature control, users have to use the instruction PID together with the instruction GPWM.

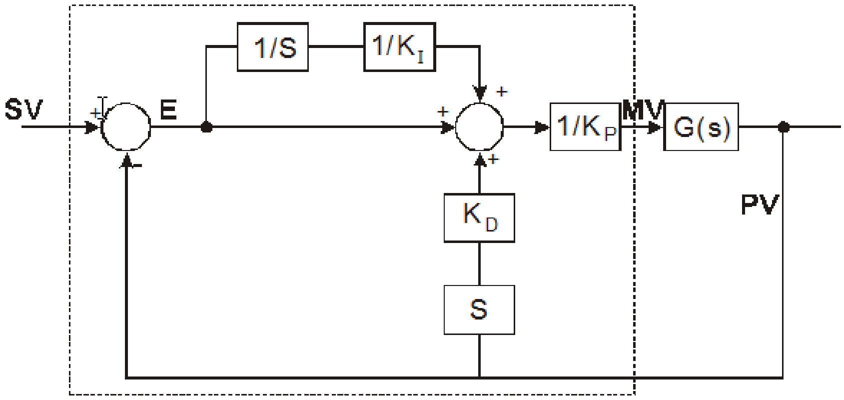


Figure 2 Control diagram for control mode 3/4 (**S_3**+4 is K3/K4.)

- This algorithm is designed for temperature control. Therefore, if the sampling time (T_S) set is four seconds (K400), the output value (MV) range will be K0~K4,000, and the cycle time of the instruction GPWM needs to be four seconds (K4000) as well.
- If users have no idea about parameter adjustment, they can select K3 first. After parameters are adjusted (the value will be change to K4 automatically), users can change the parameters to better ones according to the results of the adjustment.

Suggestions:

(1) There are a lot of environments where PID can be used, and therefore users have to select control functions appropriately. For example, if **S_3**+ 4 is K3, the instruction can not be used in a motor control environment, otherwise improper control may occur.

(2) When users adjust the three main parameters K_P, K_I and K_D (**S_3**+4=K0/K1/K2/K5), they have to adjust the K_P first (according to their experiences), and set the K_I and the K_D to 0. When users can handle the control, they can increase the K_I and the K_D. Please see the example of adjusting PID parameters manually below.

(3) To prevent the parameters which have been adjusted automatically from disappearing after a power cut, it is suggested that users should store the parameters in retentive data registers if **S_3**+4 is K3 or K4. The parameters which have been adjusted automatically are not necessarily suitable for every controlled environment. Therefore, the users can modify the parameters which have been adjusted automatically. However, it is suggested that the users only modify the K_I or the K_D.

(4) The action of PID depends on many parameters. To prevent improper control from occurring, please do not set parameters randomly.

3. Adjusting PID Parameters Manually

Assume that the transfer function of the controlled device G(S) in a control system is the first-order function $G(s) = \frac{b}{s+a}$ (the function for the model of a general motor), the target value (SV) is 1, and the sampling time (TS) is 10 milliseconds. It is suggested that users should follow the steps below.

Step 1: First, set the K_I and the K_D to 0. Next, set the K_P to 5, 10, 20 and 40 successively, and record the target values and the present values. The results are shown in Figure 3.

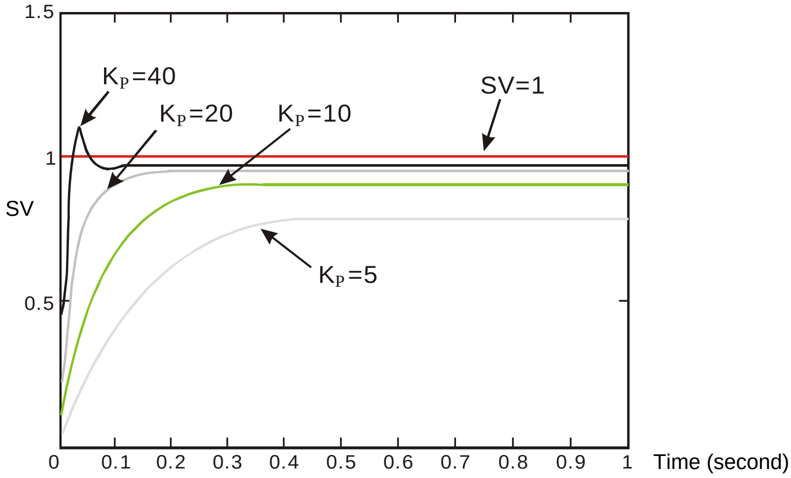


Figure 3

Step 2: When the K_P is 40, there is overreaction. Thus, the K_P is not chosen. When the K_P is 20, the reaction curve of the PV is close to the SV, and there is no overreaction. However, due to the fast start-up, the transient output value (MV) is big. The K_P is not chosen, either. When the K_P is 10, the reaction curve of the PV approaches the SV smoothly. Therefore, the K_P is chosen. When the K_P is 5, the reaction is slow. Thus, the K_P is not chosen.

Step 3: After the K_P is set to 10, increase the K_I. For example, the K_I is set to 1, 2, 4, and 8 successively. The K_I should not be greater than the K_P. Then, increase the K_D. For example, the K_D is set to 0.01, 0.05, 0.1, and 0.2 successively. The K_D should not be greater than ten percent of the K_P. Finally, the relation between the PV and the SV is presented in Figure 4.

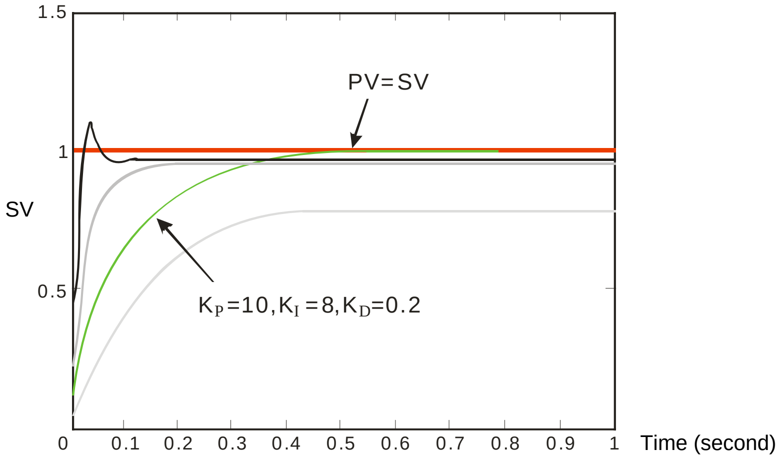


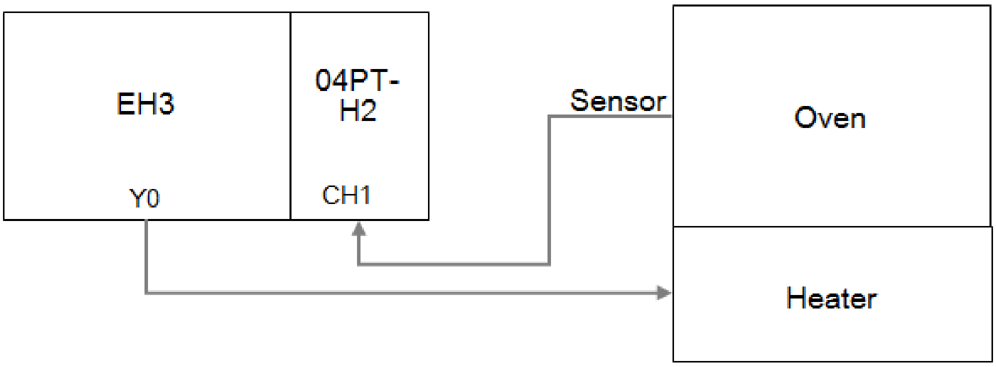
Figure 4

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

4. Examples

4.1 Example 1: Using a PLC to Realize PID Control (Using a PLC to Control an Oven)

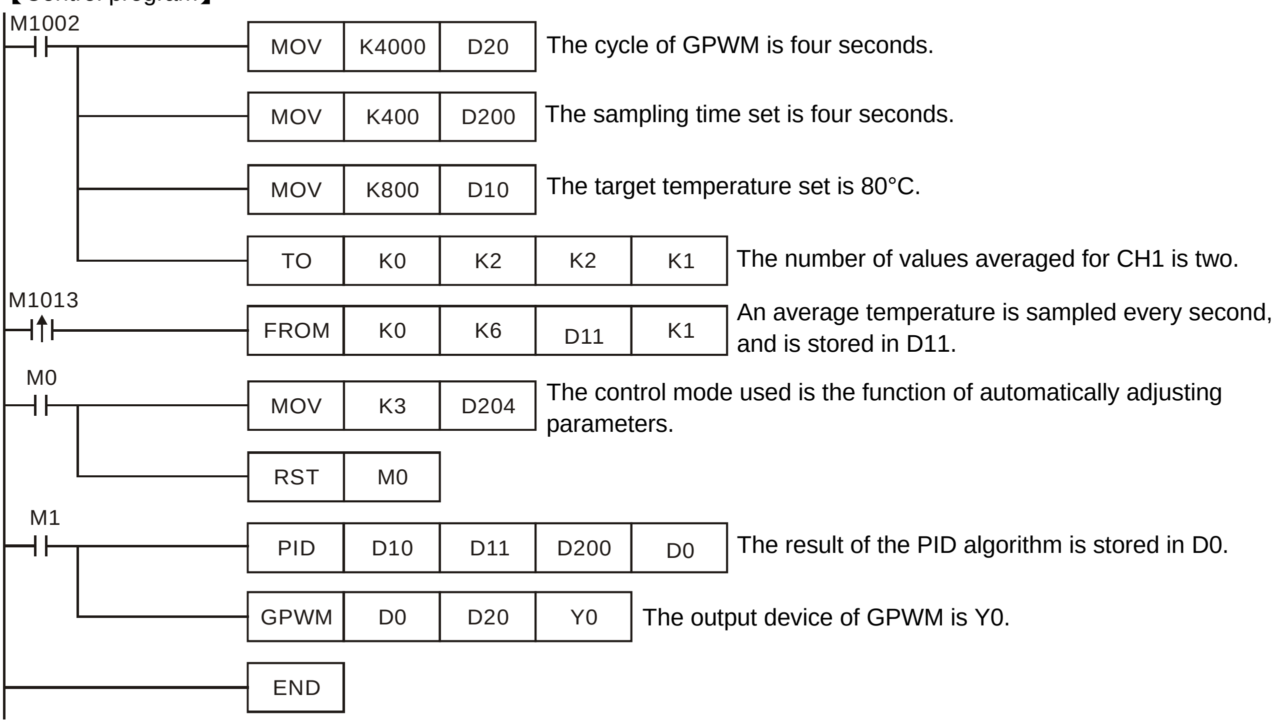
【System structure】



【Control requirement】

The control environment in this example is an oven. DVP32EH00R3 and DVP04PT-H2 are used to control the oven. First, parameters are adjusted automatically for temperature control (the value in D204 is K3). After the adjustment of the parameters is complete, the value in D204 will be changed to K4 automatically, and the parameters (K_P, K_I, and K_D) calculated will be used to realize the PID control of the oven.

【Control program】



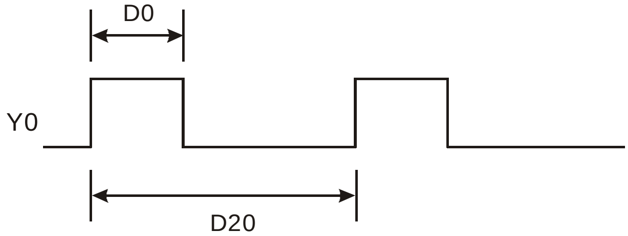
【Device description】

Device	Description
M0	Enabling the control mode
M1	Enabling the PID algorithm
Y0	Output device of GPWM

Device	Description
D0	Result of the PID algorithm
D10	Target temperature
D11	Average temperature
D20	Cycle of GPWM
D200	Sampling time of PID
D204	Control mode

【Program description】

- M0 and M1 are enabled. The temperature measurement module DVP04PT-H2 measures the temperature of the oven, and sends the temperature to the PLC. The PLC adjusts parameters automatically for temperature control (D204=K3), and calculates the best PID temperature control parameters. After the adjustment of the parameters is complete, the value in D204 is automatically changed to K4. The parameters automatically calculates (the K_P in D201, the K_I in D202, and the K_D in D203) are used to realize the PID control of the oven.
- The function of automatically adjusting parameters for temperature control is used for the PID algorithm. The output result of the PID algorithm (in D0) is used as the input of the instruction GPWM. After GPWM is executed, Y0 outputs variable width pulses to control the heater, and the PID control of the oven is realized. (The width of the pulses output by Y0 is determined by D0.)



The experimental result of the initial adjustment of the parameters is shown in Figure 5.

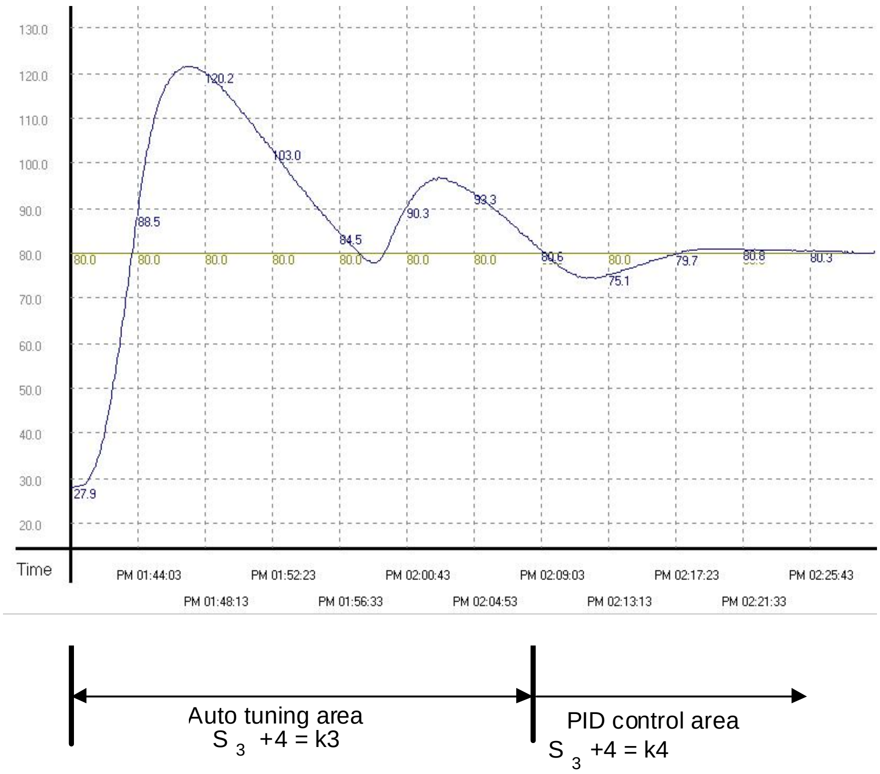


Figure 5

The experimental result of using the parameters after the adjustment for temperature control is shown in Figure 6.

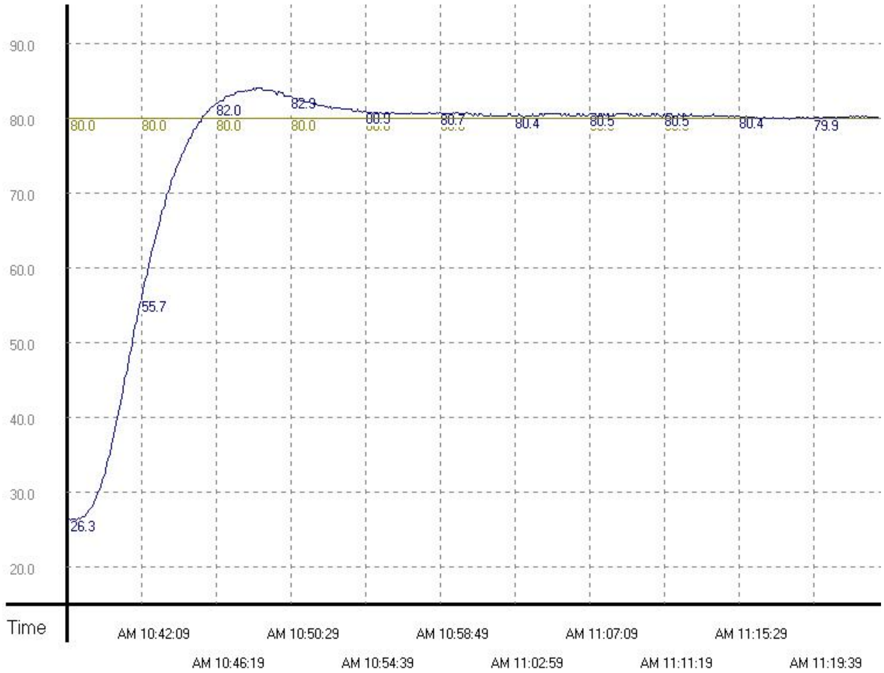


Figure 6

In Figure 6, users can see the temperature control result after the adjustment. It only takes twenty minutes to control the temperature. The result of changing the target temperature from 80°C to 100°C is shown in Figure 7.

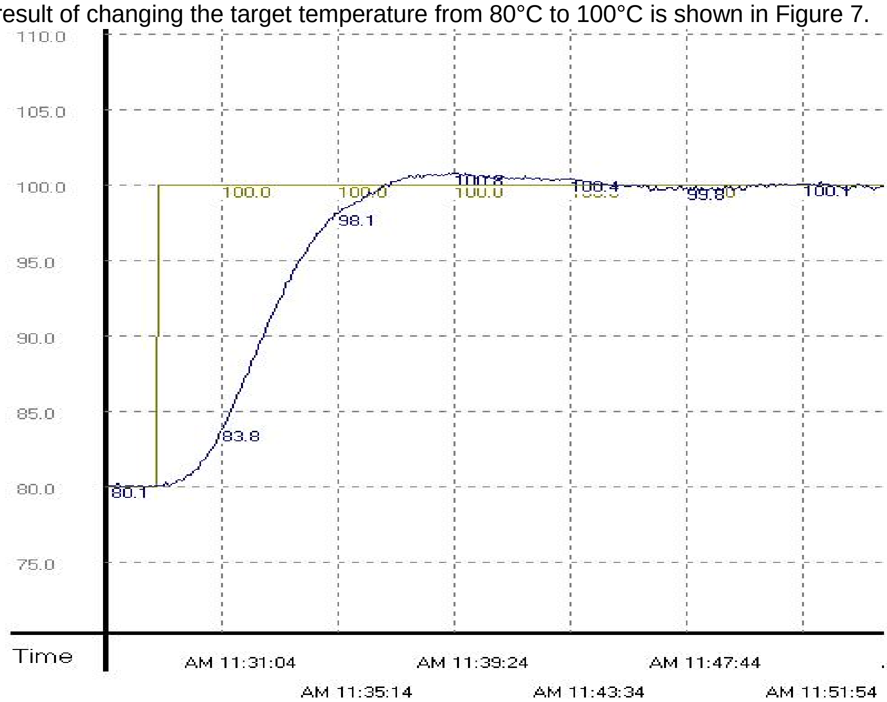
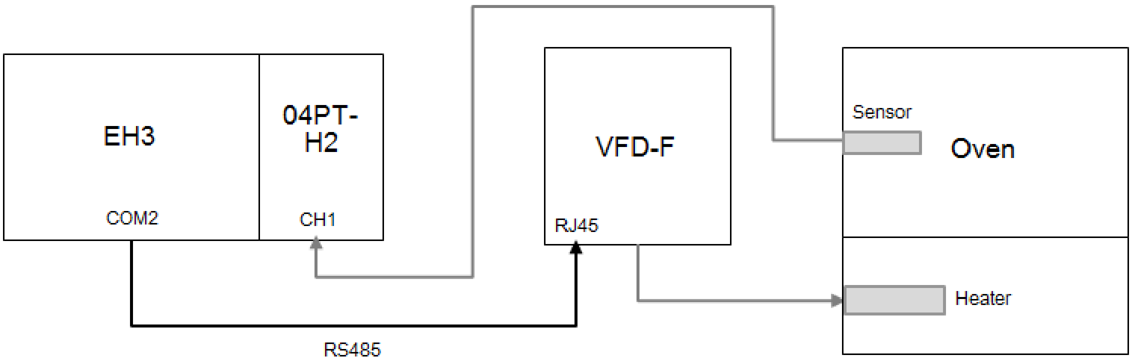


Figure 7

In Figure 7, the users can see that the purpose of controlling the temperature can still be met after the parameters adjusted for 80°C are applied to 100°C, and it does not take much time to control the temperature.

4.2 Example 2: Using a PLC to Realize PID Control (A PLC's Control of an Oven by Means of an AC Motor Drive)

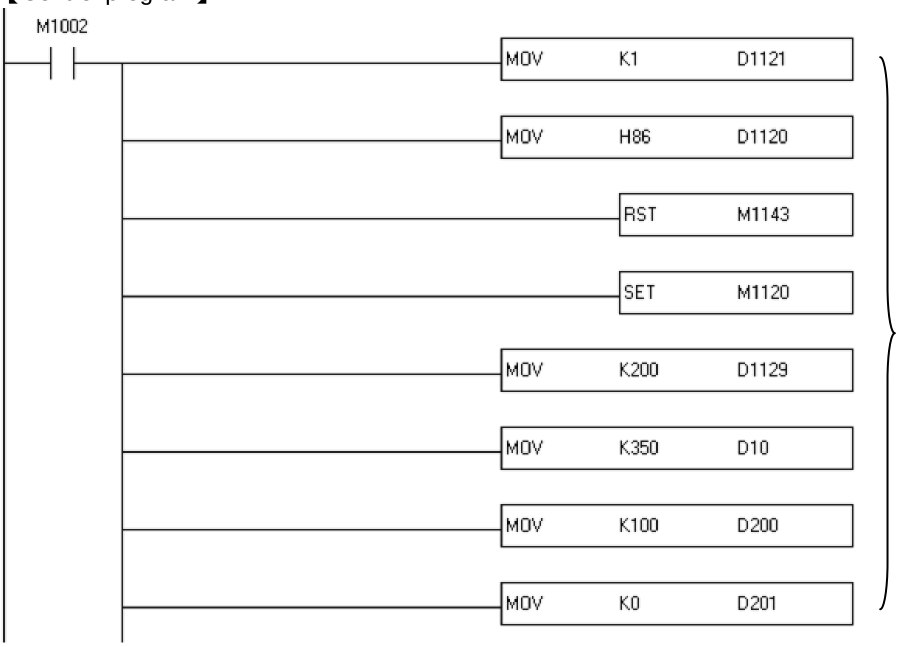
【System structure】



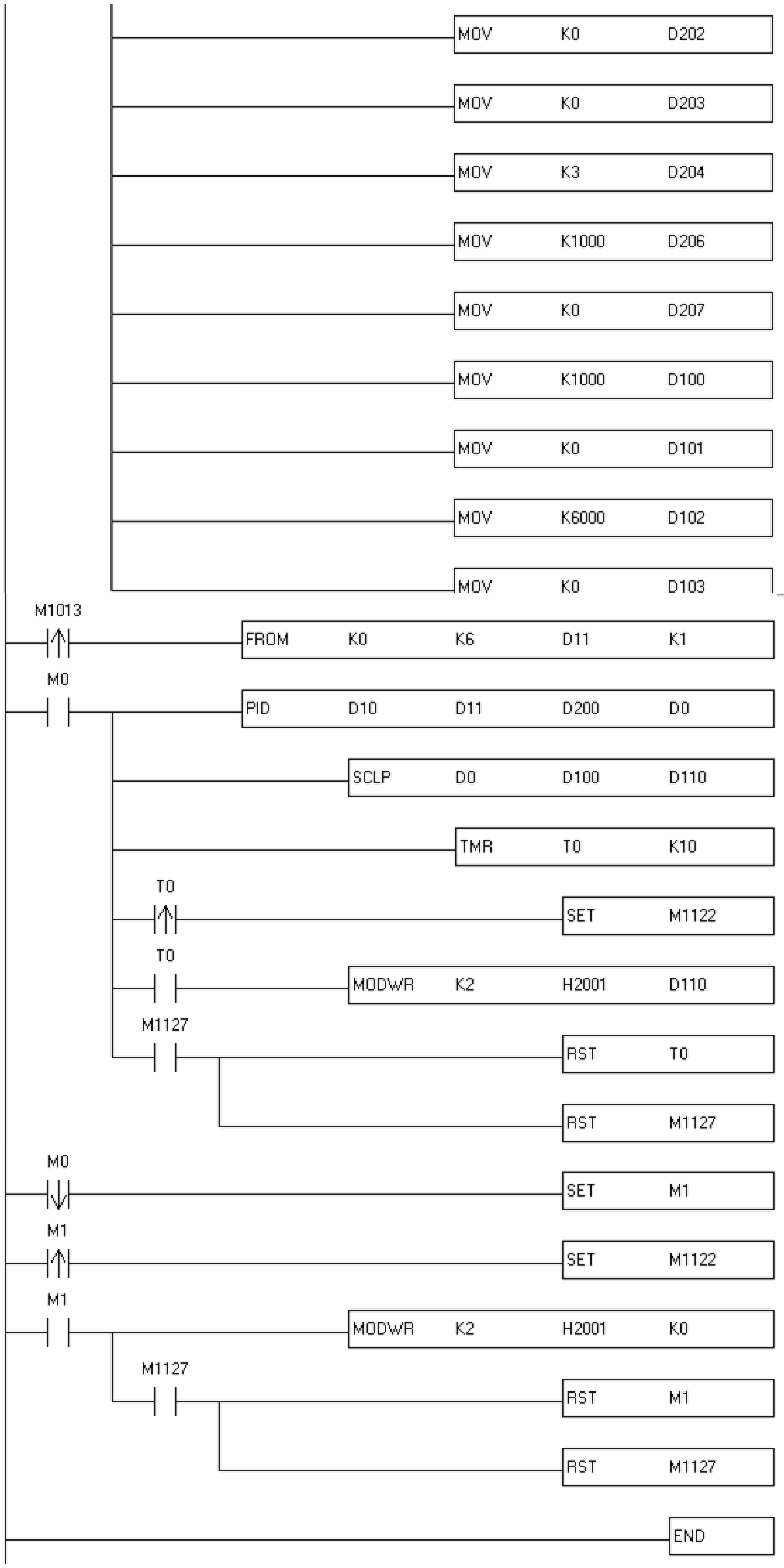
【Control requirement】

The control environment in this example is an oven. DVP32EH00R3 and DVP04PT-H2 control the oven by means of VFD007F23A. First, parameters are adjusted automatically for temperature control (the value in D204 is K3). After the adjustment of the parameters is complete, the value in D204 will be changed to K4 automatically, and the parameters (K_P, K_I, and K_D) calculated will be used to realize the PID control of the oven.

【Control program】



The communication format of COM2, and the parameters of PID are set.



The parameters of PID and SCLP are set.

The temperature of the oven is read.

The output value of the PID algorithm is converted into a scale value by SCLP, and the scale value is written to the AC motor drive by a communication instruction.

M0 is disabled, and the output of the AC motor drive is stopped.

【Device description】

Device	Description
M0	Enabling the automatic adjustment of parameters for temperature control
M1	Setting the frequency of the AC motor drive to 0 Hz
M1013	One second pulse
M1127	The receiving/sending of the data is complete.
D0	Output value of PID (MV)
D10	Target temperature of PID (SV)
D11	Present average degree Celsius for CH1 (PV)
D100~D103	Parameters of SCLP
D110	Destination device of SCLP
D200~D207	Parameters of PID

【Program description】

- The communication parameters P9-00~P9-05 are set. (The communication station address set is 2, and communication format set is "9600 7 E 1 ASCII".) The parameter P1-02 is set to 220. (The voltage of the oven is 220 V AC.) The parameter P2-00 is set to 04. (A frequency command is sent by means of RS-485 communication.) The parameter P2-01 is set to 00. (The operation of the AC motor drive is controlled by keys on the AC motor drive.) The communication station address of the PLC is 1.
- The target temperature set in D10 is 35°C. M0 is manually enabled to execute the instruction PID. The temperature of the oven is read every second by means of the module, and stored in D11. The output value of the PID algorithm is converted into a scale value by SCLP, and the scale value is written to the communication address H2001 in the AC motor drive by the instruction MODWR.
- When the value in D204 is K4, the best PID temperature control parameters (the K_P in D201, the K_I in D202, and the K_D in D203) are automatically calculated. The experimental result is shown in Figure 8.
- Use PID control mode 4 and the PID temperature control parameters (K_P, K_I, and K_D), and enable M0 again to execute the instruction PID. The experimental result is shown in Figure 9.
- Change the target temperature in D10 to 45°C, use PID control mode 4 and the PID temperature control parameters (K_P, K_I, and K_D), and enable M0 again to execute the instruction PID. The experimental result is shown in Figure 10.
- When M0 is disabled, MODWR is used to write 0 to the communication address H2001 in the AC motor drive.

When the target temperature set is 35°C, the experimental result of the initial adjustment of the parameters is like the one shown in Figure 8.

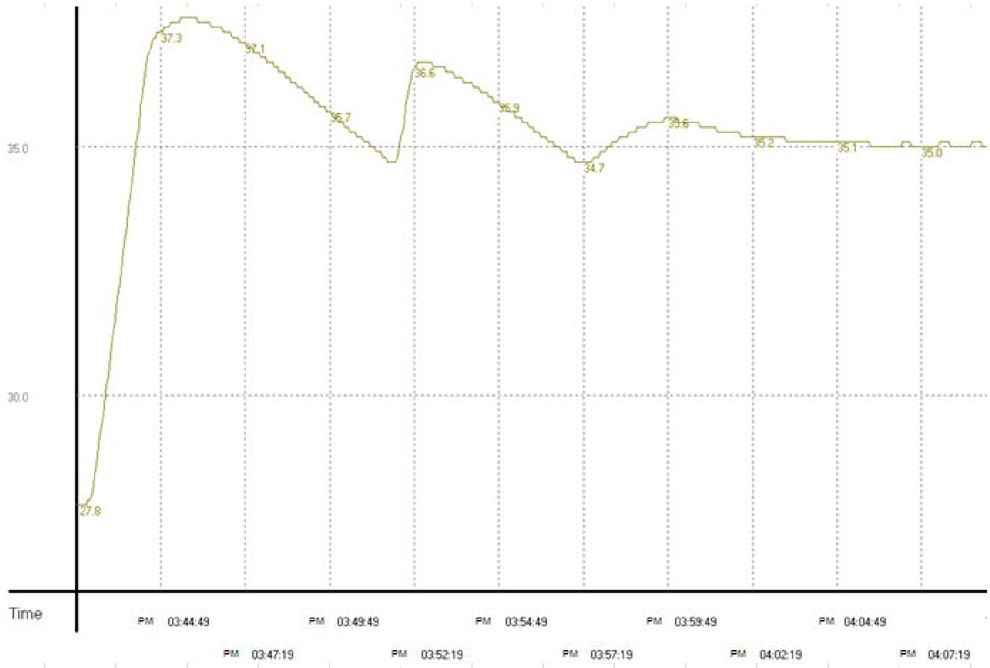


Figure 8

When the target temperature set is 35°C, the experimental result of using the parameters (K_P, K_I, and K_D) after the adjustment for temperature control is like the one shown in Figure 9.

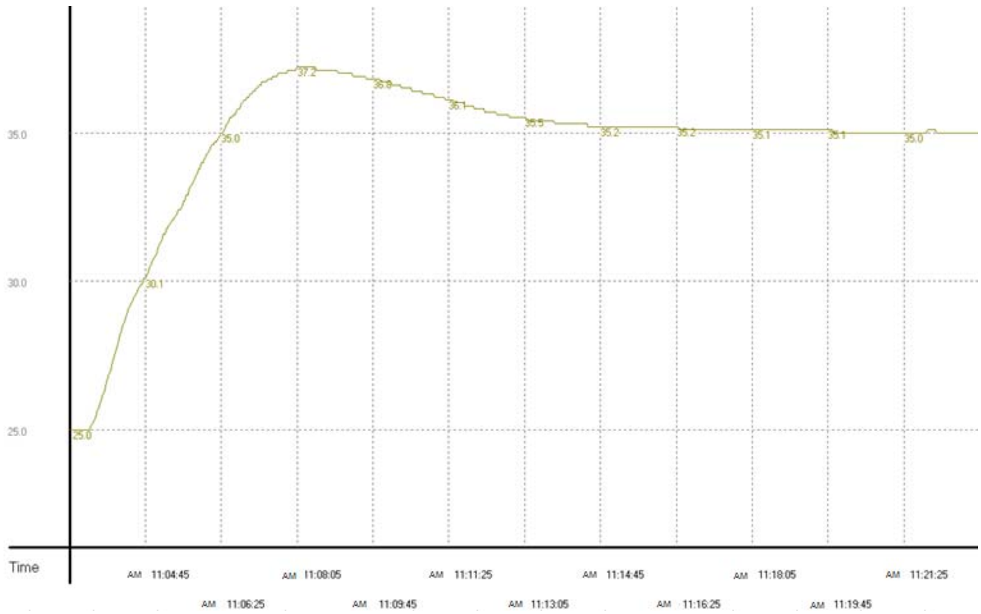


Figure 9

The parameters adjusted for 35°C are applied to 45°C. Please see Figure 10.

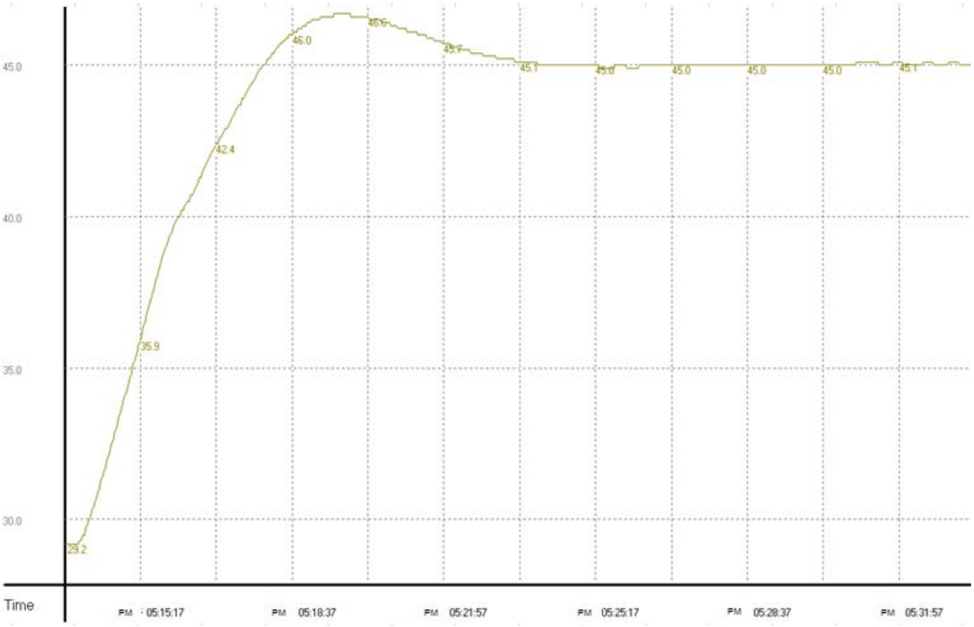
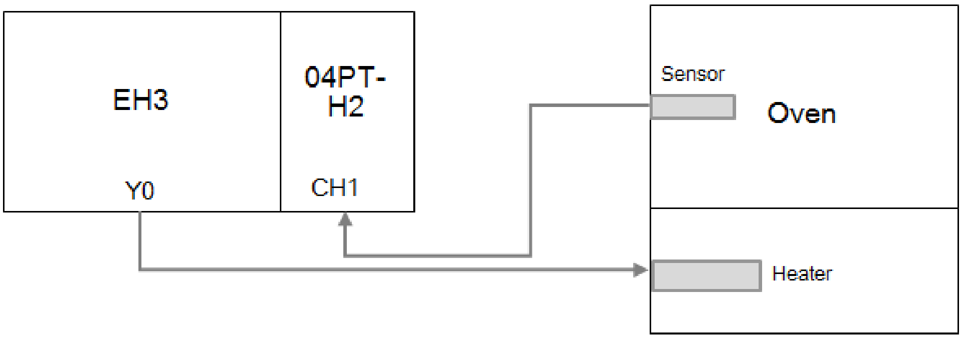


Figure 10

4.3 Example 3: Using DVP04PT-H2 to Realize PID Control

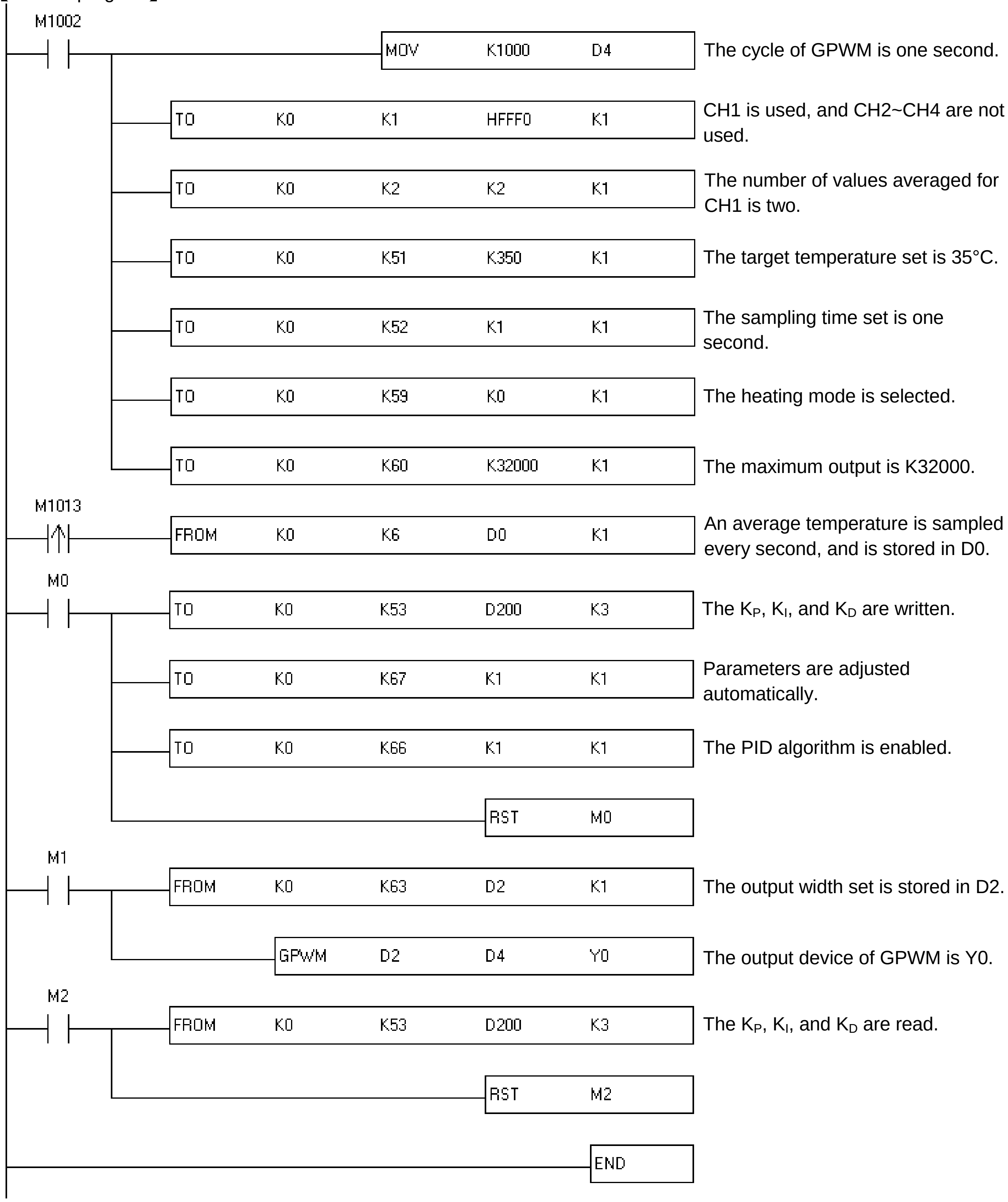
【System structure】



【Control requirement】

The control environment in this example is an oven. DVP32EH00R3 and DVP04PT-H2 are used to control the oven. First, parameters are adjusted automatically for temperature control. After the adjustment of the parameters is complete, the parameters (K_P, K_I, and K_D) calculated will be used to realize the PID control of the oven.

【Control program】



【Device description】

Device	Description
M0	Writing K_P, K_I, and K_D, and enabling the control mode
M1	Enabling the pulse width modulation

Device	Description
M2	Reading K_P, K_I, and K_D
Y0	Variable width pulse output
D0	Average degree Celsius for CH1
D2	Output width
D4	Cycle of GPWM
D200	K_P
D201	K_I
D202	K_D

【Program description】

- 0 is written to D5~D7 in the PLC. M0 and M1 are enabled. The module is used to automatically adjust parameters for temperature control. The temperature of the oven (in D0) is read by means of the module every second. The PLC uses the instruction GPWM to make Y0 enable the heating of the heater.
- When the temperature remains 35°C, the best PID temperature control parameters (the K_P in D200, the K_I in D201, and the K_D in D202) are automatically calculated. The experimental result is shown in Figure 11.
- After M2 is enabled, the PID temperature control parameters (K_P, K_I, and K_D) in the module will be read. Change the value in CR#67 in the module to K0.
- Use the PID temperature control parameters (K_P, K_I, and K_D) calculated, and enable M0 and M1 again. The experimental result is shown in Figure 12.
- Change the target temperature in CR#51 in the module to 45°C, use the PID temperature control parameters (K_P, K_I, and K_D) calculated, and enable M0 and M1 again. The experimental result is shown in Figure 13.

Descriptions of PID control registers in DVP04PT-H2:

CR#				Retentive		Register	Description
CH1	CH2	CH3	CH4				
#51	#71	#91	#111	O	R/W	Temperature setting value	Default value: K0
#52	#72	#92	#112	O	R/W	Sampling time	Range: K1~K30, Unit: Second Default value: K2
#53	#73	#93	#113	O	R/W	K_P	Default value: K121
#54	#74	#94	#114	O	R/W	K_I	Integral constant Default value: K2,098
#55	#75	#95	#115	O	R/W	K_D	Derivative constant Default value: K-29
#56	#76	#96	#116	O	R/W	Maximum integral value	Range: K-32,760~K32,760 Default value: K0
#57	#77	#97	#117	O	R/W	Minimum integral value	Range: K-32,760~K32,760 Default value: K0
#58	#78	#98	#118	X	R	Integral value	Present accumulated offset Default value: K0
#59	#79	#99	#119	O	R/W	Heating/Cooling	0: Heater 1: Cooler Default value: K0
#60	#80	#100	#120	O	R/W	Maximum output	Range: K-32,760~K32,760 Default value: K4,000
#61	#81	#101	#121	O	R/W	Minimum output	Range: K-32,760~K32,760 Default value: K0
#62	#82	#102	#122	X	R	Output percentage	Range: K0~K1,000 Unit: 0.1% Default value: K0

CR#				Retentive		Register	Description
CH1	**CH2**	**CH3**	**CH4**				
#63	#83	#103	#123	X	R	Output width	Control output width Unit: Millisecond Default value: K0
#64	#84	#104	#124	X	R	Output cycle	Control output cycle Unit: Millisecond Default value: K0
#65	#85	#105	#125	X	R	Output	Default value: K0
#66	#86	#106	#126	X	R/W	Enabling/Stopping a PID algorithm	0: Stopping a PID algorithm 1: Enabling a PID algorithm Default value: K0
#67	#87	#107	#127	X	R/W	Adjusting parameters automatically	0: No action 1: Adjusting parameters automatically Default value: K0 。

The experimental result of the initial adjustment of the parameters is shown in Figure 11.

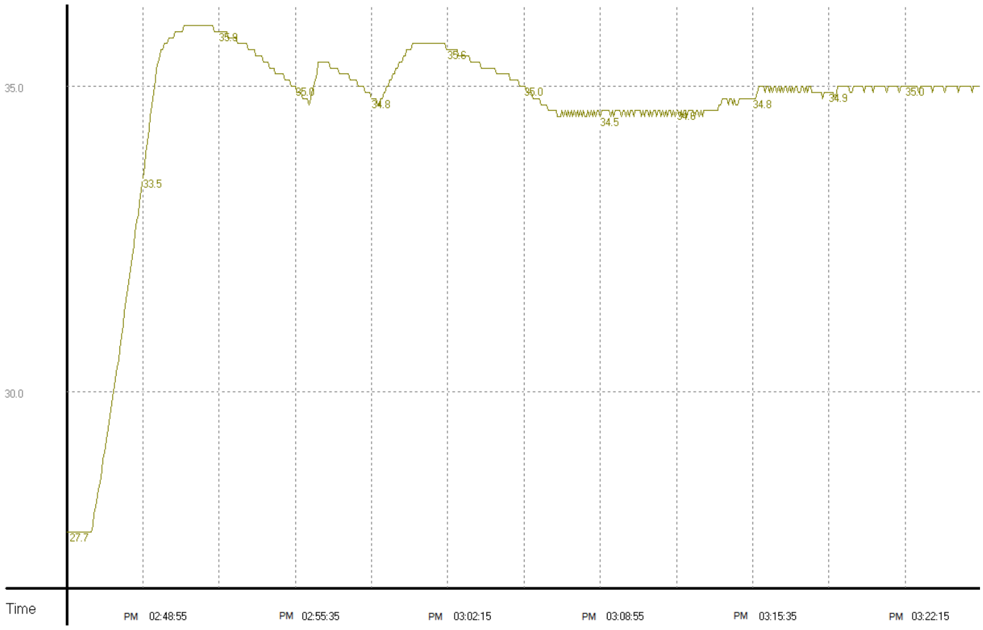


Figure 11

The experimental result of using the parameters (K_P, K_I, and K_D) after the adjustment for temperature control is shown in Figure 12.

Figure 12

The parameters adjusted for 35°C are applied to 45°C. Please see Figure 13.

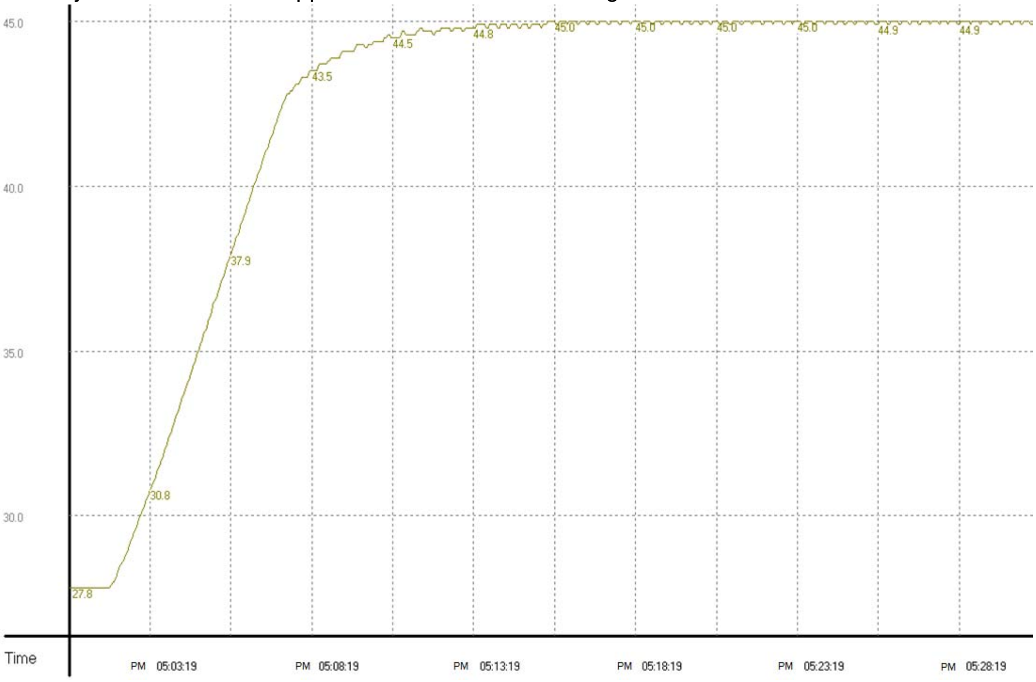
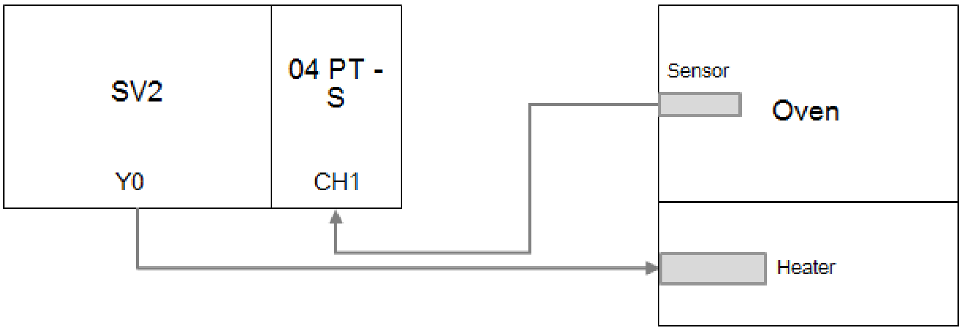


Figure 13

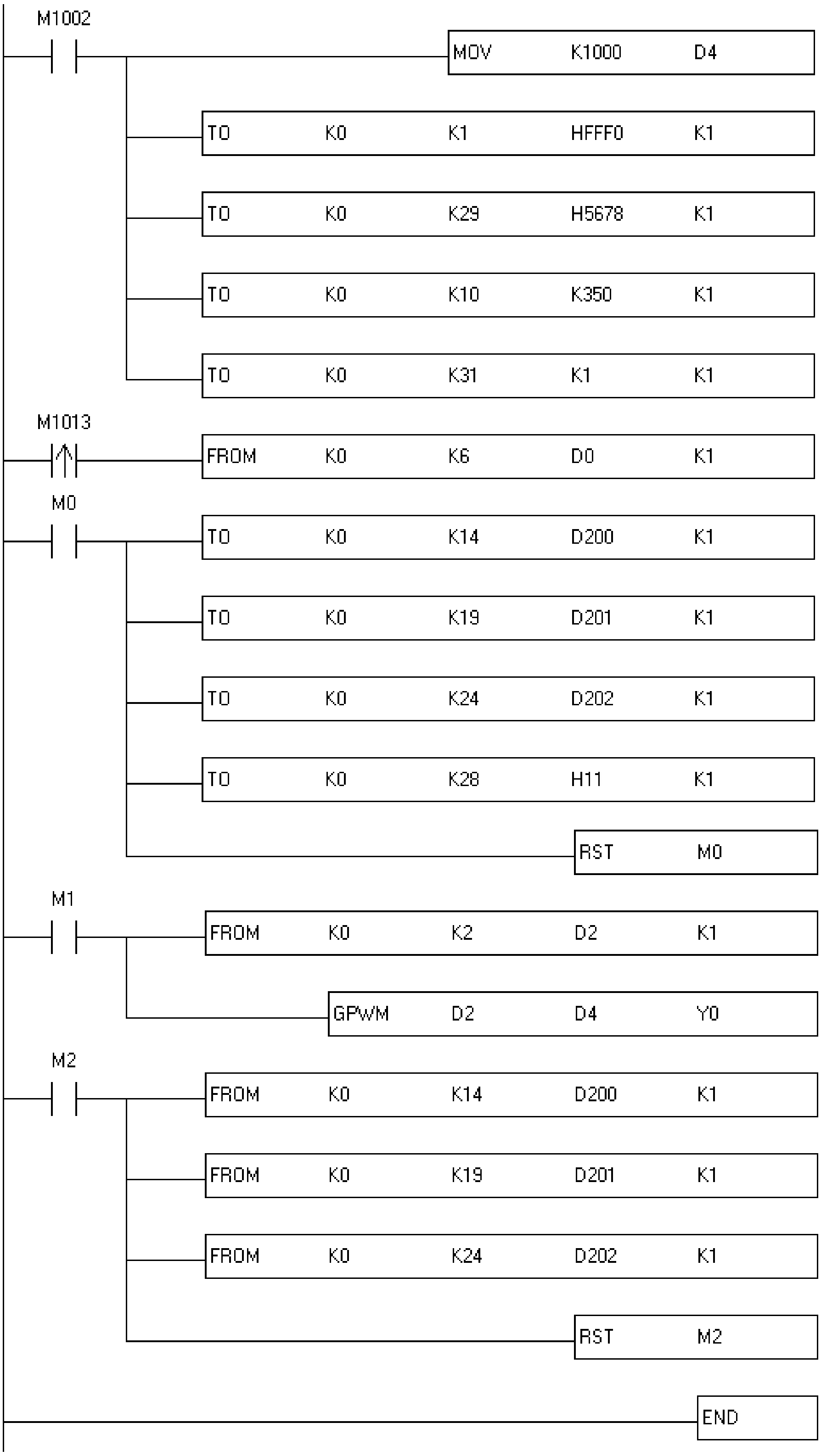
4.4 Example 4: Using DVP04PT-S to Realize PID Control

【System structure】



【Control requirement】
The control environment in this example is an oven. DVP28SV11R2 and DVP04PT-S are used to control the oven. First, parameters are adjusted automatically for temperature control. After the adjustment of the parameters is complete, the parameters (K_P, K_I, and K_D) calculated will be used to realize the PID control of the oven.

【Control program】



The cycle of GPWM is one second.

CH1 is used, and CH2~CH4 are not used.

H5678 is written to CR#29 in the module.

The target temperature set is 35°C.

The sampling time set is one second.

An average temperature is sampled every second, and is stored in D0.

The K_P, K_I, and K_D are written.

The PID algorithm is enabled, and parameters are adjusted automatically.

The output width set is stored in D2.

The output device of GPWM is Y0.

The K_P, K_I, and K_D are read.

【Device description】

Device	Description
M0	Writing K_P, K_I, and K_D, and enabling the control mode
M1	Enabling the pulse width modulation
M2	Reading K_P, K_I, and K_D
Y0	Variable width pulse output
D0	Average degree Celsius for CH1
D2	Output width
D4	Cycle of GPWM
D200	K_P
D201	K_I
D202	K_D

【Program description】

- 0 is written to D5~D7 in the PLC. M0 and M1 are enabled. The module is used to automatically adjust parameters for temperature control. The temperature of the oven (in D0) is read by means of the module every second. The PLC uses the instruction GPWM to make Y0 enable the heating of the heater.
- When the temperature remains 35°C, the best PID temperature control parameters (K_P, K_I, and K_D) are automatically calculated. The experimental result is shown in Figure 14.
- After M2 is enabled, the PID temperature control parameters (the K_P in D200, the K_I in D201, and the K_D in D202) in the module will be read. Change the value in CR#28 in the module to H1.
- Use the PID temperature control parameters (K_P, K_I, and K_D) calculated, and enable M0 and M1 again. The experimental result is shown in Figure 15.
- Change the target temperature in CR#10 in the module to 45°C, use the PID temperature control parameters (K_P, K_I, and K_D) calculated, and enable M0 and M1 again. The experimental result is shown in Figure 16.

Descriptions of PID control registers in DVP04PT-S: If the value in CR#29 in DVP04PT-S version 3.08 is H5678, CR#0~CR#34 can be used for PID control.

PID mode			
CR#0	Model name	CR#22	K_I set for CH4
CR#2	PID output percentage for CH1	CR#24	K_D set for CH1
CR#3	PID output percentage for CH2	CR#25	K_D set for CH2
CR#4	PID output percentage for CH3	CR#26	K_D set for CH3
CR#5	PID output percentage for CH4	CR#27	K_D set for CH4
CR#2~CR#5: Range: 0~1000 (Unit: 0.1%)		CR#28	Enabling/Stopping a PID algorithm and adjusting parameters automatically
CR#6	Average degree Celsius for CH1		Bit 0: Enabling/Stopping a PID algorithm (CH1)
CR#7	Average degree Celsius for CH2		Bit 1: Enabling/Stopping a PID algorithm (CH2)
CR#8	Average degree Celsius for CH3		Bit 2: Enabling/Stopping a PID algorithm (CH3)
CR#9	Average degree Celsius for CH4		Bit 3: Enabling/Stopping a PID algorithm (CH4)
CR#6~CR#9: Unit: 0.1°C			0: Stopping a PID algorithm 1: Enabling a PID algorithm
CR#10	Temperature set for CH1		Bit 4: Adjusting parameters automatically by means of CH1
CR#11	Temperature set for CH2		Bit 5: Adjusting parameters automatically by means of CH2
CR#12	Temperature set for CH3		Bit 6: Adjusting parameters automatically by means of CH3
CR#13	Temperature set for CH4		Bit 7: Adjusting parameters automatically by means of CH4

PID mode			
CR#10~CR#13: PID target value (SV)		CR#29	If bit 4/bit 5/bit6/bit 7 is 1, parameters will be adjusted automatically. After the adjustment of the parameters is complete, 1 will be changed to 0 automatically.
CR#14	K_P set for CH1		Entering a PID mode (H5678) K0: Exiting a PID mode
CR#15	K_P set for CH2	CR#30	Error code
CR#16	K_P set for CH3	CR#31	Sampling time (CH1)
CR#17	K_P set for CH4	CR#32	Sampling time (CH2)
CR#19	K_I set for CH1	CR#33	Sampling time (CH3)
CR#20	K_I set for CH2	CR#34	Sampling time (CH4)
CR#21	K_I set for CH3	CR#31~CR#34: Range: 1~30 (Unit: 1 second)	
Note: Users have to write H5678 to CR#29 before they set other control registers.			

The experimental result of the initial adjustment of the parameters is shown in Figure 14.

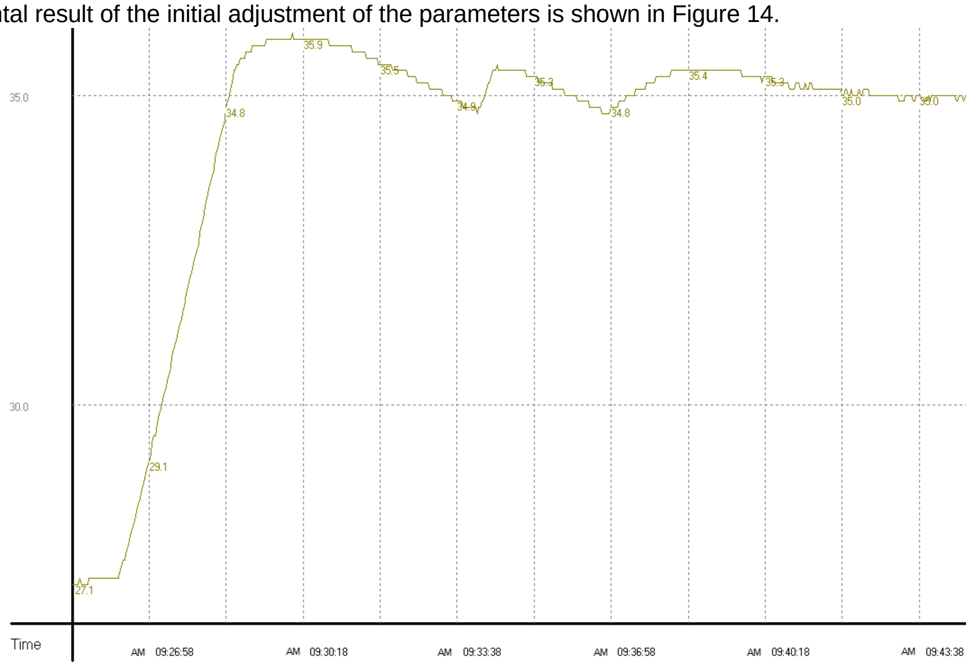


Figure 14

The experimental result of using the parameters (K_P, K_I, and K_D) after the adjustment for temperature control is shown in Figure 15.

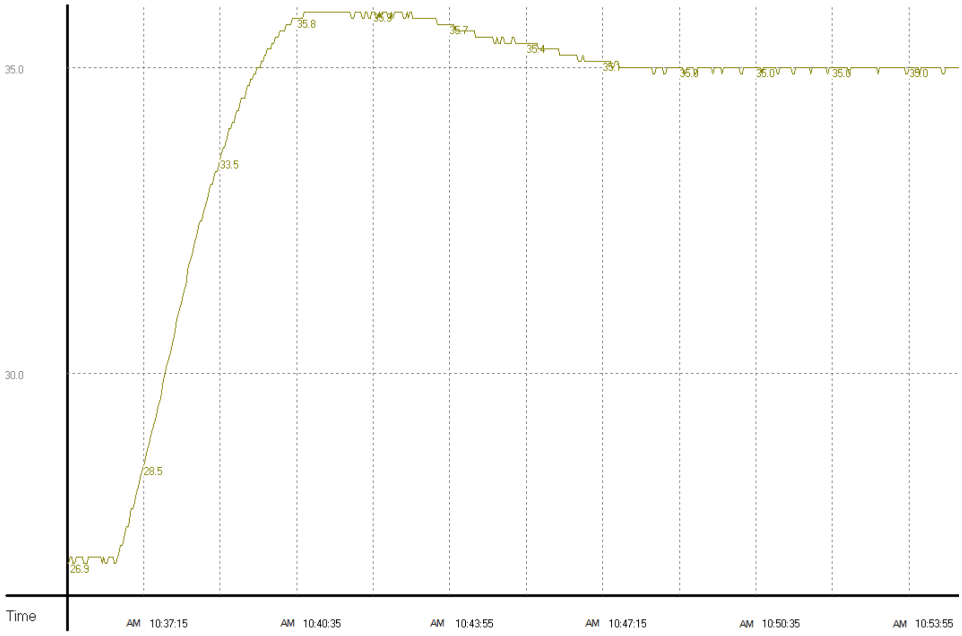


Figure 15

The parameters adjusted for 35°C are applied to 45°C. Please see Figure 16.

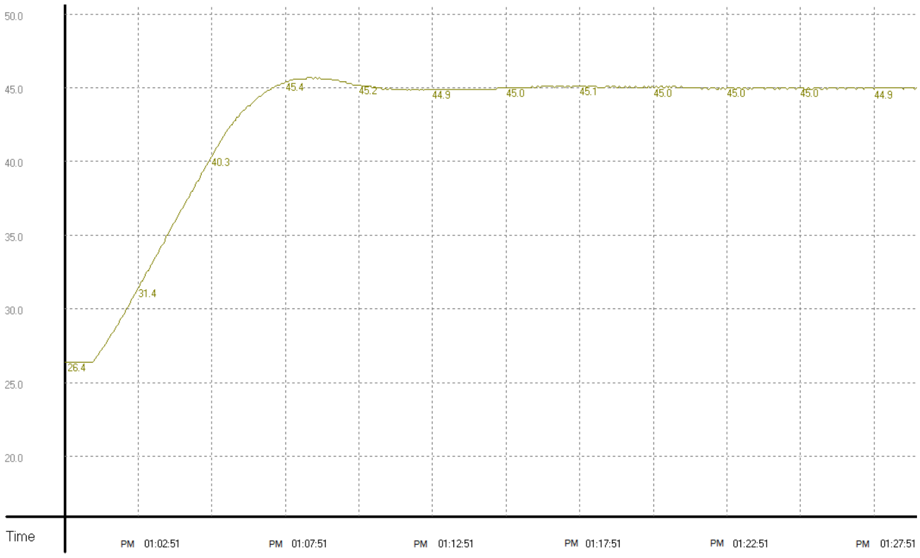


Figure 16