

**主题 : AH500 系列 DPID/DPIDE 功能说明 ( 附温控实例 )**

适用机种	AH500 系列
关键词	DPID 功能、DPIDE 功能

## 目录

<b>1. 前言及目的 .....</b>	<b>3</b>
<b>2. 主机DPID指令说明 .....</b>	<b>4</b>
<b>3. 主机DPIDE指令说明 .....</b>	<b>10</b>
<b>4. 主机DPID/DPIDE参数手动调整技巧 .....</b>	<b>20</b>
<b>5. 范例说明 .....</b>	<b>21</b>
5.1 范例一：主机温度DPID控制 ( 主机输出控制小灯箱 ) .....	21
5.2 范例二：主机温度DPIDE控制 ( 主机输出控制小灯箱 ) .....	26
5.3 范例三：主机温度DPID控制 ( 主机输出控制大温箱 ) .....	30
5.4 范例四：主机温度DPIDE控制 ( 主机输出控制大温箱 ) .....	32
5.5 范例五：AH04TC模块温度DPID控制 .....	34

## 1. 前言及目的

前言：

工程实际中，应用最为广泛的调节器控制规律为比例积分微分控制，简称 PID 控制，又称 PID 调节。PID 控制问世至今已有近 60 年的历史了，它以其结构简单、稳定性好、工作可靠、调整方便而成为工业控制主要和可靠的技术工具。

目的：

因使用者第一次对于 PID 控制特性不熟悉，透过本文件能让使用者了解 PID 控制模式原理及如何使用 PID 控制模式。

## 2. 主机 DPID 指令说明

API	指令码			操作数													功能	
0707	D	PID		PID_RUN · SV · PV · PID_MODE · PID_MAN · MOUT_AUTO · CYCLE · K <sub>p</sub> · K <sub>i</sub> · K <sub>D</sub> · PID_DIR · ERR_DBW · MV_MAX · MV_MIN · MOUT · I_MV · MV													PID 运算	

装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
PID_RUN	●	●	●					●	●	●			●				
SV								●	●				●				
PV								●	●				●				
PID_MODE								●	●				●				
PID_MAN	●	●	●					●	●	●			●				
MOUT_AUTO	●	●	●					●	●	●			●				
CYCLE								●	●				●				
K <sub>p</sub>								●	●				●				
K <sub>i</sub>								●	●				●				
K <sub>D</sub>								●	●				●				
PID_DIR	●	●	●					●	●	●			●				
ERR_DBW								●	●				●				
MV_MAX								●	●				●				
MV_MIN								●	●				●				
MOUT								●	●				●				
I_MV								●	●				●				
MV								●	●				●				

脉冲执行型	16 位指令	32 位指令 ( 35 steps )
-	-	AH500

符号 :

DPID	
En	
PID_RUN	MV
SV	
PV	
PID_MOD	
PID_MAN	
MOUT_AL	
CYCLE	
Kp	
Ki	
Kd	
PID_DIR	
ERR_DBW	
MV_MAX	
MV_MIN	
MOUT	
I_MV	

<b>PID_RUN</b>	: 启动 PID 运算	BIT
<b>SV</b>	: 目标值 ( SV )	DWORD
<b>PV</b>	: 现在值 ( PV )	DWORD
<b>PID_MODE</b>	: PID 控制模式	DWORD
<b>PID_MAN</b>	: PID A/M 模式 ( PID_MAN )	BIT
<b>MOUT_AUTO</b>	: MOUT Auto Change 模式	BIT
<b>CYCLE</b>	: 取样时间 ( CYCLE )	DWORD
<b>K<sub>p</sub></b>	: P计算值系数 ( <b>K<sub>p</sub></b> )	DWORD
<b>K<sub>i</sub></b>	: I计算值系数 ( <b>K<sub>i</sub></b> )	DWORD
<b>K<sub>d</sub></b>	: D计算值系数 ( <b>K<sub>d</sub></b> )	DWORD
<b>PID_DIR</b>	: PID 正反向 ( PID_DIR )	BIT
<b>ERR_DBW</b>	: ERR 的不作用范围 ( ERR_DBW )	DWORD
<b>MV_MAX</b>	: MV 上限值 ( MV_MAX )	DWORD
<b>MV_MIN</b>	: MV 下限值 ( MV_MIN )	DWORD
<b>MOUT</b>	: MV 手动值 ( MOUT )	DWORD
<b>I_MV</b>	: 累计积分项的数值 ( I_MV )	DWORD
<b>MV</b>	: MV 输出值 ( MV )	DWORD

指令说明 :

- PID 运算控制的专用指令，于取样时间到达后的该次扫描才执行 PID 运算动作。PID 表示“比例、积分和微分”。PID 控制在机械设备、气动设备和电子设备中具有广泛的应用。
- 参数设定内容如下：

装置编号	数据类型	功能	设定范围	说明
<b>PID_RUN</b>	BOOL	启动 PID 运算	TRUE : PID 指令开始运算。 FALSE : MV 值清除为 0，PID 不运算。	
<b>SV</b>	REAL	SV 值	单精度浮点数 范围	目标值
<b>PV</b>	REAL	PV 值	单精度浮点数 范围	现在值
<b>PID_MODE</b>	DWORD/DINT	PID 控制模式	0 : 自动控制，当 PID_MAN 由 TRUE 转为 FALSE 时，MV 值会由当时输出的 MV 值开始进行自动运算。	

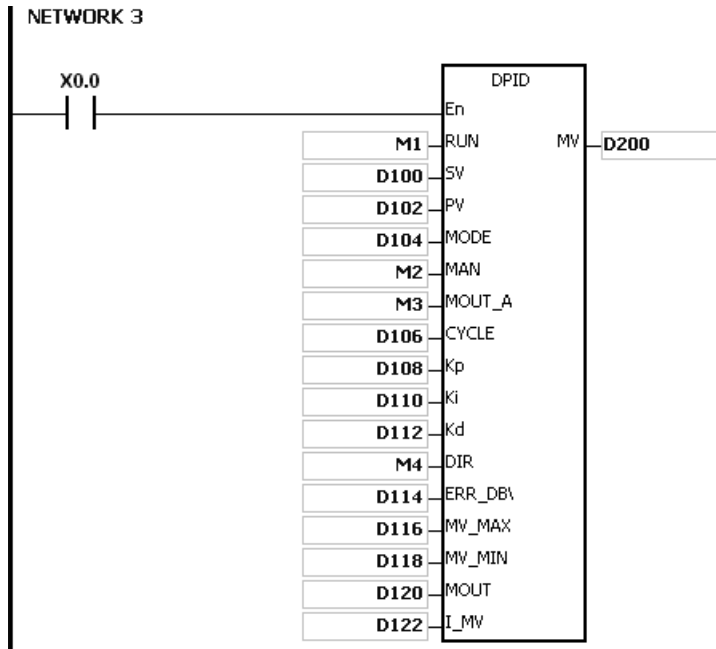
装置编号	数据类型	功能	设定范围	说明
			1 : 自动调整参数功能 · 调整完毕时将自动进入自动控制模式 ( PID_MODE 改为 0 ) · 并且填入最适用的 $K_P$ 、 $K_I$ 及 $K_D$ 等参数。 2 : 与 0 相同 · 但当 PID_MAN 由 TRUE 转为 FALSE 时 · MV 值会由当时内部运算的 MV 值开始进行自动运算。 设定值超出范围 · 将视为 0。	
<b>PID_MAN</b>	BOOL	PID A/M 模式	TRUE : Manual · MV 值会依 MOUT 值输出 · 但仍在 MV_MAX 与 MV_MIN 之间 · 当 PID_MODE 为 1 时此设定无效。 FALSE : Auto · MV 值会依 PID 公式计算后输出 · 输出值在 MV_MAX 与 MV_MIN 之间。	
<b>MOUT_AUTO</b>	BOOL	MOUT Auto change 模式	TRUE : Auto · MOUT 的值会随着 MV 值改变。 FALSE : Normal · MOUT 的值不会随着 MV 值改变。	
<b>Cycle</b>	DWORD/DINT	取样时间 ( $T_S$ )	1~2,000 ( 单位 : 10ms )	每次扫描到本指令时 · 就会以 Cycle 设定的取样时间来计算 PID ( 不会计算扫描时间是否有到达取样时间 ) · 并更新输出值 ( MV ) · $T_S$ 小于 1 则内定为 1 · 大于 2000 则为 2000。 当 PID 指令用于 Interval Interrupt Task 时 · Cycle 时间以 Interval Interrupt Task 的时间为主。
<b><math>K_P</math></b>	REAL	比例增益 ( $K_P$ )	正数单精度浮点数范围	为 SV-PV 间的误差放大比例值 · 如果小于 0 · $K_P$ 将为 0
<b><math>K_I</math></b>	REAL	积分增益 ( $K_I$ )	正数单精度浮点数范围	I 计算值系数 ( $K_I$ ) · 如果小于 0 · $K_I$ 将为 0

装置编号	数据类型	功能		设定范围	说明
<b>K<sub>D</sub></b>	REAL	微分增益 ( K <sub>D</sub> )		正数单精度浮点数范围	D计算值系数 ( K <sub>D</sub> ) · 如果小于 0 · K <sub>D</sub> 将为 0
<b>PID_DIR</b>	BOOL	PID 正反向		TRUE : 反向动作 ( E=SV-PV ) FALSE : 正向动作 ( E=PV-SV )	
<b>ERR_DBW</b>	REAL	偏差量 ( E ) 不作用范围		单精度浮点数范围	偏差量 ( E ) 等于 SV-PV 的误差值 · 当设定值为 0 即表示不启动此功能 · 例 : 设定值为 5 或 -5 · 则 E 在 -5~5 的区间偏差量将为 0
<b>MV_MAX</b>	REAL	输出值 ( MV ) 饱和上限		单精度浮点数范围	例 : 设定 1,000 · 则输出值 ( MV ) 大于 1,000 时将以 1,000 输出 · 需大于 S3+7 · 否则上限值与下限值将互换
<b>MV_MIN</b>	REAL	输出值 ( MV ) 饱和下限		单精度浮点数范围	例 : 设定 -1,000 · 则输出值 ( MV ) 小于 -1,000 时将以 -1,000 输出
<b>MOUT</b>	REAL	MV 手动值		配合 PID_MAN 模式使用 · 使用者直接设定 MV 输出值 ·	
<b>I_MV</b> ( 连续占用 6 个 DWord 装置 )	REAL	I_MV	暂存累积的积分值	单精度浮点数范围	为累积的积分值 · 通常只供参考用 · 但是使用者还是可以依需求清除或修改 · 不过须以单精度浮点数修改之 · 当 MV 超出 MV_MAX 或 MV_MIN 时 · I_MV 值不会再改变
		I_MV+1	暂存前次 PV 值	为前次测定值 · 通常只供参考用 · 但是使用者还是可以依需求修改 ·	
		I_MV+2 ~ I_MV+5	系统用参数 · 使用者请勿使用 ·		
<b>MV</b>	REAL	MV 输出值		MV 值会介于 MV_MAX 与 MV_MIN 之间 ·	

**程序范例 :**

1. 执行 PID 指令前先将参数设定完成 ·

2. X0.0=ON 的时候指令被执行，M1=ON 的时候 PID 指令才开始进行运算，M1=OFF 时 MV 值为 0，MV 值存于 D200 中。X0.0 变成 OFF 时，指令不被执行，之前的数据没有变化。



#### 补充说明：

1. 指令无使用次数之限制，但是 I\_MV~I\_MV<sub>+5</sub> 所指定的寄存器号码不可重复。
2. I\_MV 占用 12 个寄存器，于上述程序例当中 I\_MV 指定 PID 指令的参数设定区域为 D122~D133。
3. PID 指令只能使用在 Cyclic Task 与 Interval Interrupt Task 当中，当指令用于 Interval Interrupt Task 时，Cycle 时间以 Interval Interrupt Task 的时间为主。
4. PID 被启动后，每次只要被扫描到都会以 Cycle 所设定的时间来做 PID 运算并直接输出 MV 值，并不会自动计算扫描时间是否有到达取样时间才输出，但如果本指令使用在时间中断程序内，取样时间将与时间中断程序的间隔时间相同，PID 指令会以时间中断程序的中断间隔时间来做 PID 的运算。
5. PID 的测定值 ( PV ) 在 PID 执行运算动作前必须是一个稳定值。如果要抓取特殊模块的输入值作 PID 运算时，请注意这些模块的 A/D 转换时间。

#### PID 计算公式：

1. 当 PID\_MODE 控制模式选择为 0、2，自动控制模式。
2. 当 PID\_MODE 控制模式选择为 1，自动调整模式，当自动调整完成，PID\_MODE 会变成 0 自动控制模式。
  - a) PID 的运算分成正向动作，逆向动作。而正逆向动作由 DIR 来指定。

1. PID 的表达式：

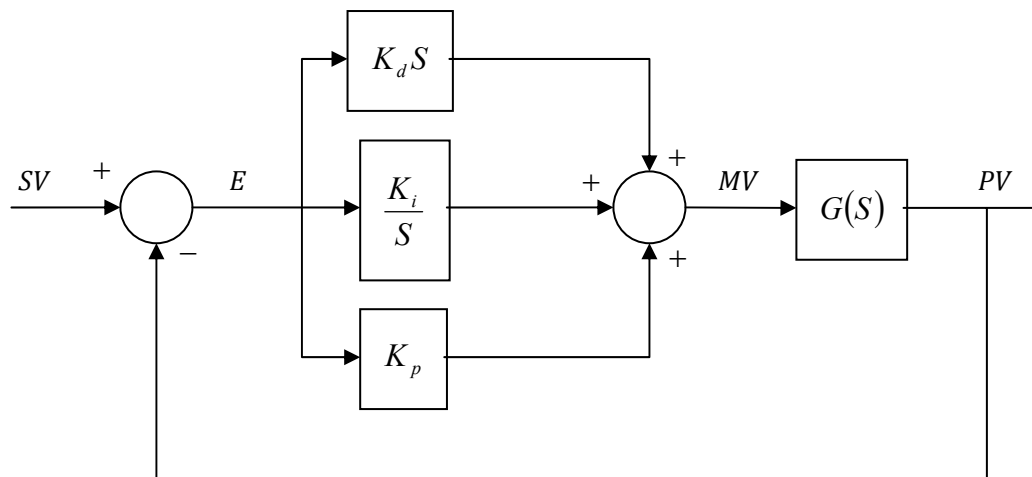
$$2. MV = K_P E(t) + K_I \int_0^t E(t) dt + K_D * \frac{dE(t)}{dt}$$

3. 其中  $E(t)$  表示  $E(t)$  的微分值，以及  $E(t) \frac{1}{S}$  表示  $E(t)$  的积分值。



动作方向	PID 演算方式
正向动作、自动	$E(t) = PV(t) - SV(t)$
逆向动作	$E(t) = SV(t) - PV(t)$

● 控制方块图：



● 符号说明：

$MV$  : 输出值

$E(t)$  : 偏差量。正向动作  $E(t) = PV - SV$ ，反向动作  $E(t) = SV - PV$

$K_p$  : 比例增益

$PV$  : 测定值

$SV$  : 目标值

$K_D$  : 微分增益

$K_I$  : 积分增益

注意事项和建议：

1. 由于可使用 PID 指令的控制环境很多，因此请适当的选取控制功能，例如：当选择自动调整参数 (PID\_MODE=1) 功能时，就请勿使用于马达控制环境中，以免造成控制不当的现象发生。
2. 使用者于调整  $K_p$ 、 $K_i$  及  $K_D$  三个主要参数时 (PID\_MODE=0、2)，请先调整  $K_p$  值 (依经验值设定)，而  $K_i$  及  $K_D$  值先设定为 0，等到调整到大致上可控制时，再依序调整  $K_i$  值 (由小到大) 以及  $K_D$  值 (由小到大)，调整范例如范例四所示。其中  $K_p$  值为 1 则表示 100%，即对偏差值的增益为 1，小于 100% 将对偏差值衰减，大于 100% 将对偏差值放大。
3. 当使用者选用 (PID\_MODE=1) 时，建议请使用在停电保持区之 D 缓存器来储存参数，以免自动调整过的参数因停电后而消失。经过自动调整过的参数，并不能保证一定适用于每个控制的环境，因此使用者当然可自行修改调整过的参数，不过建议最好只修改  $K_i$  或  $K_D$  数值就好。
4. 本指令动作须配合许多参数值控制，因此请勿随意设定参数值，以免造成无法控制之现象。

### 3. 主机 DPIDE 指令说明

API	指令码			操作数								功能				
0708	D	PIDE		PID_RUN · SV · PV · PID_MODE · PID_MAN · MOUT_AUTO · CYCLE · Kc_Kp · Ti_Ki · Td_Kd · Tf · PID_EQ · PID_DE · PID_DIR · ERR_DBW · MV_MAX · MV_MIN · MOUT · BIAS · I_MV · MV								PID 运算				

装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
PID_RUN	●	●	●					●	●	●			●				
SV								●	●				●				●
PV								●	●				●				●
PID_MODE								●	●				●				
PID_MAN	●	●	●					●	●	●			●				
MOUT_AUTO	●	●	●					●	●	●			●				
CYCLE								●	●				●				
Kc_Kp								●	●				●				
Ti_Ki								●	●				●				
Td_Kd								●	●				●				
Tf								●	●				●				
PID_EQ	●	●	●					●	●	●			●				
PID_DE	●	●	●					●	●	●			●				
PID_DIR	●	●	●					●	●	●			●				
ERR_DBW								●	●				●				●
MV_MAX								●	●				●				●
MV_MIN								●	●				●				●
MOUT								●	●				●				
BIAS								●	●				●				●
I_MV								●	●				●				
MV								●	●				●				

脉冲执行型	16 位指令	32 位指令 ( 43 steps )
-	-	AH500

符号 :

DPIDE	
En	
PID_RUN	MV
SV	
PV	
PID_MODE	
PID_MAN	
MOUT_AUTO	
CYCLE	
Kc_Kp	
Ti_Ki	
Td_Kd	
Tf	
PID_EQ	
PID_DE	
PID_DIR	
ERR_DBW	
MV_MAX	
MV_MIN	
MOUT	
BIAS	
I_MV	

<b>PID_RUN</b>	: 启动 PID 运算	BIT
<b>SV</b>	: 目标值	DWORD
<b>PV</b>	: 现在值	DWORD
<b>PID_MODE</b>	: PID 控制模式	DWORD
<b>PID_MAN</b>	: PID Auto/Manual 模式	BIT
<b>MOUT_AUTO</b>	: 手动值 ( MOUT ) 自动更新模式	BIT
<b>CYCLE</b>	: 取样时间 ( ms )	DWORD
<b>Kc_Kp</b>	: 比例项系数	DWORD
<b>Ti_Ki</b>	: 积分项系数 ( sec 或 1/sec )	DWORD
<b>Td_Kd</b>	: 微分项系数 ( sec )	DWORD
<b>Tf</b>	: 微分作用时间常数 ( sec )	DWORD
<b>PID_EQ</b>	: PID 计算式选择	BIT
<b>PID_DE</b>	: PID 微分项误差计算选择	BIT
<b>PID_DIR</b>	: PID 正反向	BIT
<b>ERR_DBW</b>	: ERR 的不作用范围	DWORD
<b>MV_MAX</b>	: MV 输出上限值	DWORD
<b>MV_MIN</b>	: MV 输出下限值	DWORD
<b>MOUT</b>	: MV 手动值	DWORD
<b>BIAS</b>	: 前馈控制输出值	DWORD
<b>I_MV</b>	: 积分项累计值	DWORD
<b>MV</b>	: 输出值	DWORD

指令说明 :

- PID 进阶运算控制的专用指令，当指令被主机执行时才会进行 PID 运算与处理。PID 表示“比例、积分和微分”。PID 控制在机械设备、气动设备和电子设备中具有广泛的应用。
- 参数设定内容如下：

装置编号	数据类型	功能	设定范围	说明
<b>PID_RUN</b>	BOOL	启动 PID 运算	TRUE : PID 指令开始运算。 FALSE : MV 值清除为 0，PID 不运算。	
<b>SV</b>	REAL	SV 值	单精度浮点数范围	目标值

装置编号	数据类型	功能	设定范围	说明
<b>PV</b>	REAL	PV 值	单精度浮点数范围	现在值
<b>PID_MODE</b>	DWORD/DINT	PID 控制模式	0 : 自动控制 · 当 PID_MAN 由 TRUE 转为 FALSE 时 · MV 值会由当时输出的 MV 值开始进行自动运算。 1 : 自动调整参数功能 · 调整完毕时将自动进入自动控制模式 ( PID_MODE 改为 0 ) · 并且填入最适用的 Kc_Kp、Ti_Ki、Td_Kd 及 Tf 等参数。	
<b>PID_MAN</b>	BOOL	PID A/M 模式	TRUE : Manual · MV 值会依 MOUT 值输出 · 但仍在 MV_MAX 与 MV_MIN 之间 · 当 PID_MODE 为 1 时此设定无效。 FALSE : Auto · MV 值会依 PID 公式计算后输出 · 输出值在 MV_MAX 与 MV_MIN 之间。	
<b>MOUT_AUTO</b>	BOOL	MOUT 自动更新模式	TRUE : Auto · MOUT 的值会随着 MV 值改变。 FALSE : Normal · MOUT 的值不会随着 MV 值改变。	
<b>CYCLE</b>	DWORD/DINT	取样时间 ( $T_s$ )	1~40,000 ( 单位: ms )	每次扫描到本指令时 · 就会以 CYCLE 设定的取样时间来计算 PID ( 主机并不会依照此数值来自动判断时间并自动执行 ) · 并更新输出值 ( MV ) 。 CYCLE 小于 1 则内定为 1 · 大于 40,000 时则为 40,000 。当 PID 指令用于时间中断时 · 主机会自动以时间中断的中断时间来计算 PID · 此时 CYCLE 的设定无效。

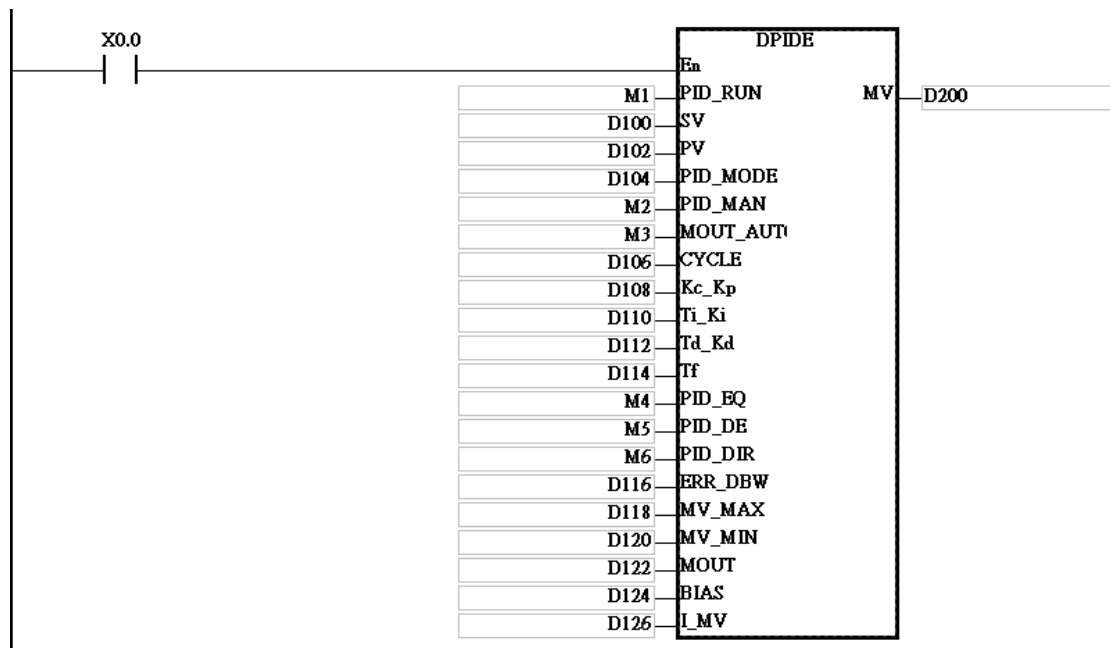
装置编号	数据类型	功能	设定范围	说明
<b>Kc_Kp</b>	REAL	比例项系数 ( Kc or Kp · 依 PID_EQ 参数决定使用何种系数 )	正数单精度浮点数范围	P 计算值系数 · 如果小于 0 则 Kc_Kp 将为 0 · 在 Independent 下若 Kc_Kp 等于 0 则表示不使用 P 控制 ·
<b>Ti_Ki</b>	REAL	积分项系数 ( Ti or Ki · 依 PID_EQ 参数决定使用何种系数 )	正数单精度浮点数范围 ( 单位 : Ti = sec ; Ki = 1/sec )	I 计算值系数 · 如果小于 0 则 Ti_Ki 将为 0 · 当 Ti_Ki 等于 0 时则表示不使用 I 控制 ·
<b>Td_Kd</b>	REAL	微分项系数 ( Td or Kd · 依PID_EQ参数决定使用何种系数 )	正数单精度浮点数范围 ( 单位 : sec )	D 计算值系数 · 如果小于 0 则 Td_Kd 将为 0 · 当 Td_Kd 等于 0 时则表示不使用 D 控制 ·
<b>Tf</b>	REAL	微分作用时间常数 ( Tf )	正数单精度浮点数范围 ( 单位 : sec )	微分作用时间常数 · 如果小于 0 则 Tf 将为 0 · 当 Tf 等于 0 时则表示不使用微分作用时间的控制 ( Derivative Smoothing ) ·
<b>PID_EQ</b>	BOOL	PID 计算式选择	TRUE : Dependent Formula FALSE : Independent Formula	
<b>PID_DE</b>	BOOL	PID 微分项误差计算选择	TRUE : 使用现在值 ( PV ) 的变化量来计算微分项的控制值 ( Derivative of PV ) FALSE : 使用偏差量 ( E ) 的变化量来计算微分项的控制值 ( Derivative of E )	
<b>PID_DIR</b>	BOOL	PID 正反向	TRUE : 反向动作 ( E=SV-PV ) FALSE : 正向动作 ( E=PV-SV )	

装置编号	数据类型	功能	设定范围	说明
ERR_DBW	REAL	偏差量 ( E ) 不作用范围	单精度浮点数范围	偏差量 ( E ) 等于 SV-PV 或 PV-SV · 当 ERR_DBW 设定为 0 时即表示不启动此功能 · 否则主机会去检查这次的 E 值是否小于 ERR_DBW 的绝对值且是否符合 Cross 状态转换条件 · 若都有则将 E 值视为 0 之后进行 PID 计算 · 否则依照正常处理将 E 值代入 PID 计算。
MV_MAX	REAL	输出值 ( MV ) 饱和上限	单精度浮点数范围	例：设定 1,000 · 则输出值 ( MV ) 大于 1,000 时将以 1,000 输出。 此设定值需大于 MV_MIN · 否则主机会自动将 MV_MAX 与 MV_MIN 进行互换。
MV_MIN	REAL	输出值 ( MV ) 饱和下限	单精度浮点数范围	例：设定 -1,000 · 则输出值 ( MV ) 小于 -1,000 时将以 -1,000 输出。 此设定值需小于 MV_MAX · 否则主机会自动将 MV_MAX 与 MV_MIN 进行互换。
MOUT	REAL	MV 手动值	单精度浮点数范围	配合 PID_MAN 模式使用 · 当 PID 设定在手动模式 ( PID_MAN=True ) 时 · 则输出值 ( MV ) 会依照手动设定值 ( MOUT ) 输出 · 但是仍然在 MV_MAX 与 MV_MIN 之间。

装置编号	数据类型	功能		设定范围	说明	
<b>BIAS</b>	REAL	前馈控制输出值		单精度浮点数范围	使用于 PID 前馈控制。	
<b>I_MV</b> ( 连续占用 10 个 DWord 装置 )	REAL	I_MV	暂存累积的积分值	单精度浮点数范围	为累积的积分值，通常只供参考用，但是使用者还是可以依需求清除或修改，不过须以单精度浮点数修改之，当 MV 超出 MV_MAX 或 MV_MIN 时，I_MV 值不会再改变	
		I_MV+1	暂存前次 E 值	系统纪录前次偏差量。		
		I_MV+2 ~ I_MV+5	系统用参数，使用者请勿使用。			
		I_MV+6	系统纪录前次 PV 值。			
		I_MV+7 ~ I_MV+9	系统用参数，使用者请勿使用。			
<b>MV</b>	REAL	MV 输出值	MV 值会介于 MV_MAX 与 MV_MIN 之间。			

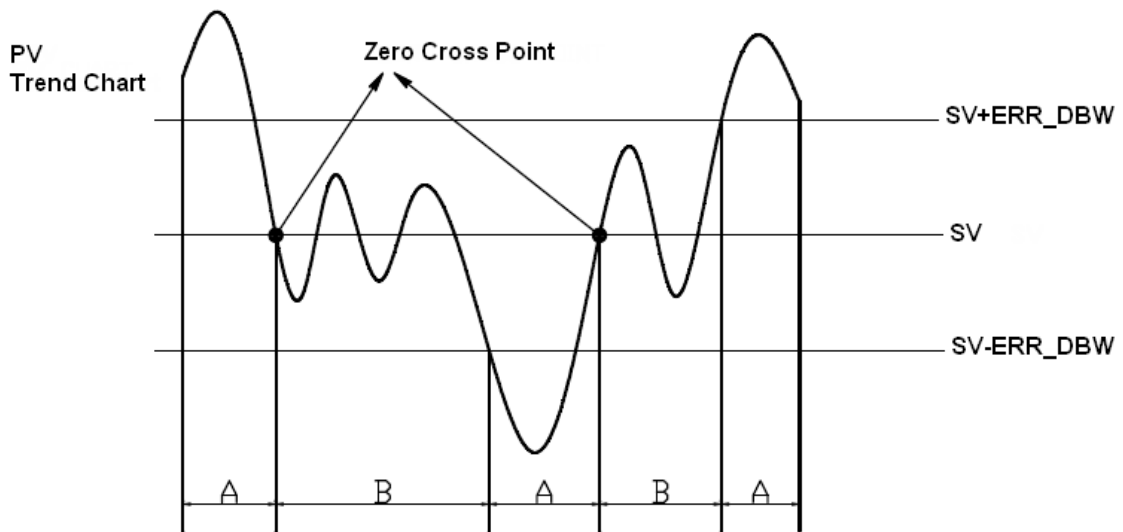
**程序范例：**

1. 执行 DPIDE 指令前先将参数设定完成。
2. X0.0=ON 的时候指令被执行，M1=ON 的时候 DPIDE 指令才开始进行运算，M1=OFF 时 MV 值为 0，MV 值存于 D200 中。X0.0 变成 OFF 时，指令不被执行，之前的数据没有变化。


**补充说明：**

1. 指令无使用次数之限制，但是 **I\_MV~I\_MV+9** 所指定的缓存器号码不可被其它程序重复使用。
2. **I\_MV** 占用 20 个 Word 缓存器，于上述程序例当中 **I\_MV** 所占用的设定区域为 D126~D145。
3. DPIDE 指令只能使用在周期性工作 (Cyclic Task) 与时间中断当中，当指令用于时间中断时，取样时间 (CYCLE) 会自动以时间中断的时间为主。
4. DPIDE 只要被扫描到都会以取样时间 (CYCLE) 所设定的时间来做 PID 运算并直接输出 MV 值，并不会自动计算扫描时间是否有到达取样时间 (CYCLE) 才输出，但如果本指令使用在时间中断程序内，取样时间 (CYCLE) 将与时间中断程序的间隔时间相同，DPIDE 指令会以时间中断程序的中断间隔时间来做 PID 的运算。
5. DPIDE 的现在值 (PV) 在 PID 执行运算动作前必须是一个稳定值。如果要抓取特殊模块的输入值作 PID 运算时，请注意这些模块的 A/D 转换时间。
6. 当 PV 值进入 ERR\_DBW 的范围时，一开始主机仍会依照 E 值进行 PID 计算，直到 PV 穿过 SV 值时 (Zero Cross Point) 代表 Cross Status 成立，此时会将 E 值视为 0 代入 PID 计算，一直到 PV 值超出 ERR\_DBW 的范围时才会恢复将 E 值代入 PID 计算，若 PID\_DE=True 则表示使用 PV 值来进行微分项的计算，则在 Cross Status 条件成立后，主机会将 Delta PV 视为 0 进行 PID 微分项的计算。(Delta PV=当前 PV-前次 PV)  
 例如以下的 PV 趋势图中，A 的区段主机会依照正常的 PID 进行计算，而 B 的区段主机会将 E 或 Delta PV 视为 0 进行 PID 计算。





### PID 计算公式 :

当 **PID\_MODE** 控制模式设定为 0 时 , 为自动控制模式。

#### 1. Independent Formula & Derivative of E ( PID\_EQ=False & PID\_DE=False )

$$MV = K_p E + K_i \int_0^t E dt + K_d \frac{dE}{dt} + BIAS$$

$$E = SV - PV \quad \text{or} \quad E = PV - SV$$

#### 2. Independent Formula & Derivative of PV ( PID\_EQ=False & PID\_DE=True )

$$MV = K_p E + K_i \int_0^t E dt - K_d \frac{dPV}{dt} + BIAS$$

$$E = SV - PV$$

or

$$MV = K_p E + K_i \int_0^t E dt + K_d \frac{dPV}{dt} + BIAS$$

$$E = PV - SV$$

#### 3. Dependent Formula & Derivative of E ( PID\_EQ=True & PID\_DE=False )

$$MV = K_c \left[ E + \frac{1}{T_i} \int_0^t E dt + T_d \frac{dE}{dt} \right] + BIAS$$

$$E = SV - PV \quad \text{or} \quad E = PV - SV$$

**4. Dependent Formula & Derivative of PV ( PID\_EQ=True & PID\_DE=True )**

$$MV = K_c \left[ E + \frac{1}{T_i} \int_0^t E dt - T_d \frac{dPV}{dt} \right] + BIAS$$

$$E = SV - PV$$

or

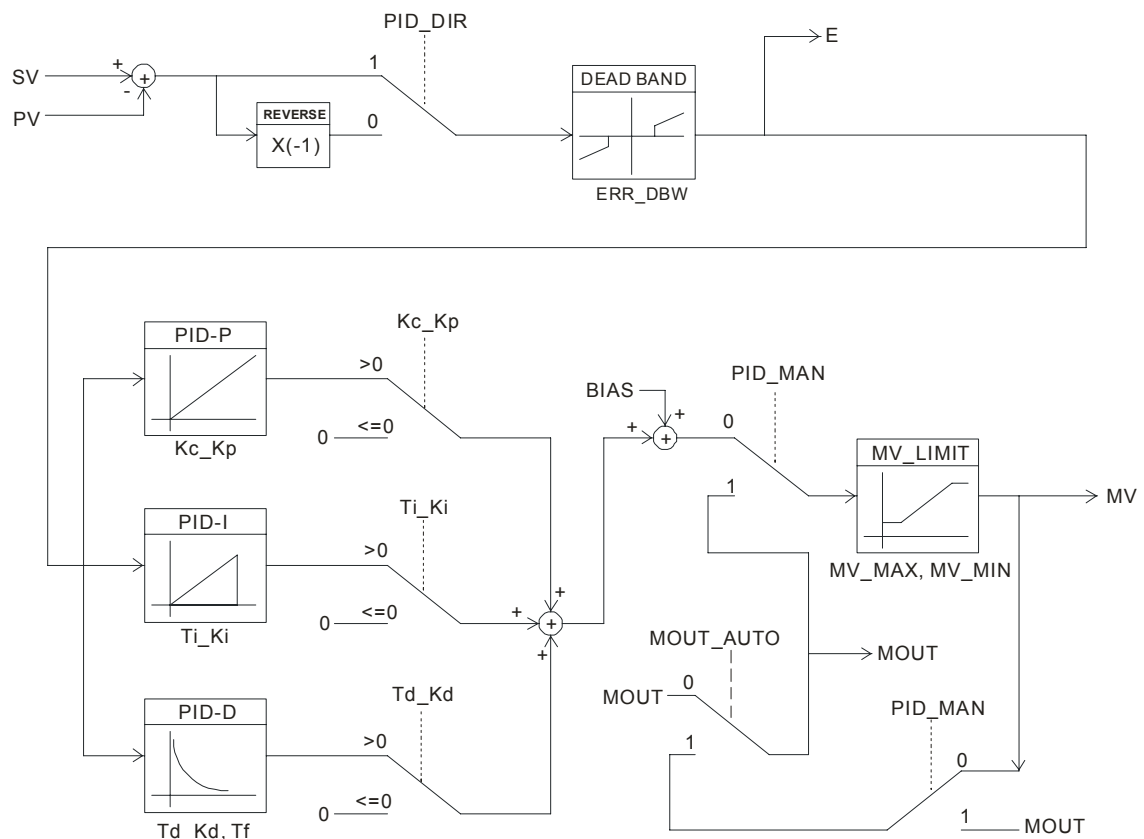
$$MV = K_c \left[ E + \frac{1}{T_i} \int_0^t E dt + T_d \frac{dPV}{dt} \right] + BIAS$$

$$E = PV - SV$$

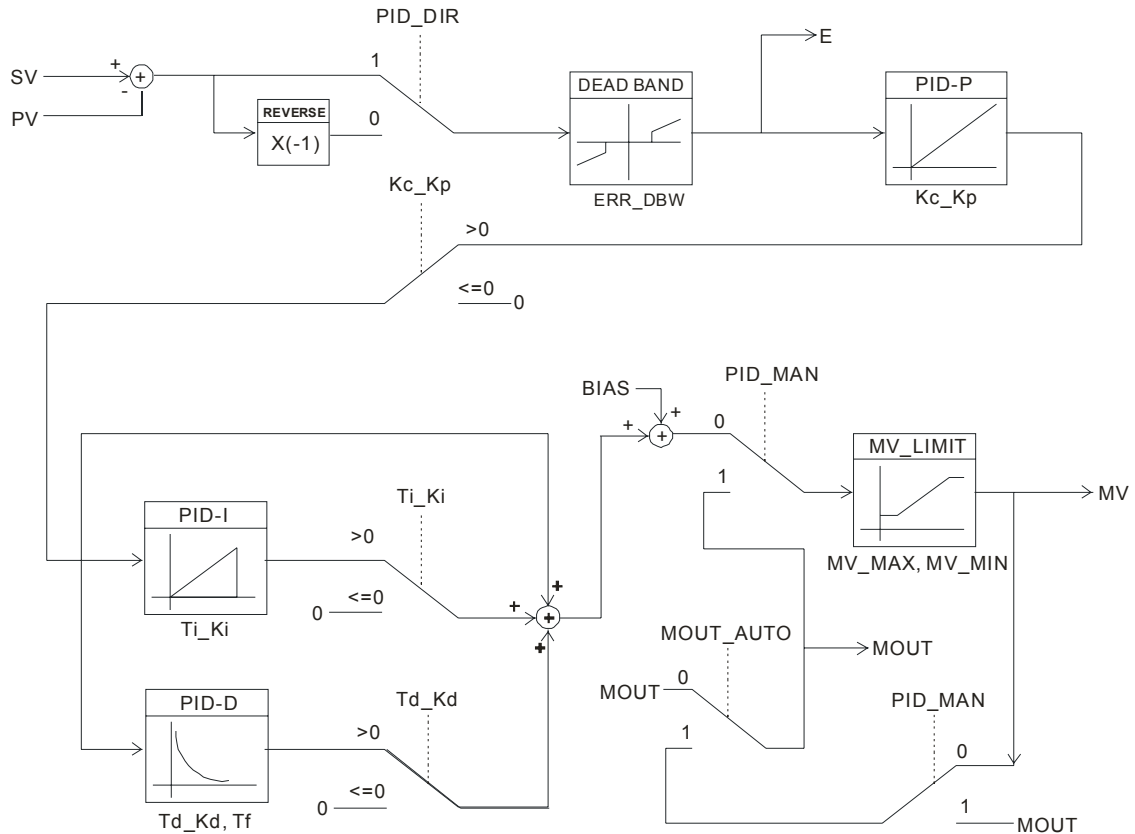
当 **PID\_MODE** 控制模式选择为 1，为自动调整模式，当自动调整完成后，**PID\_MODE** 会自动变成 0 转换为自动控制模式。

**PID 控制方块图：**

PID Block Diagram (Independent)



## PID Block Diagram (Dependent)



## 注意事项和建议：

- 由于可使用 DPIDE 指令的控制环境很多，因此请适当的选取控制功能，例如：当选择自动调整参数 (PID\_MODE=1) 功能时，MV 值会在 MAX/MIN 两个值之间做切换，所以请尽量勿使用于快速反应的马达控制环境中，以免造成在自动调适时系统因快速的剧烈变化，而造成对人员危险或损坏系统的现象发生。
- 使用者于调整 Kc\_Kp、Ti\_Ki 及 Td\_Kd 三个主要参数时 (PID\_MODE=0)，请先调整 Kc\_Kp 值 (依经验值设定)，而 Ti\_Ki 及 Td\_Kd 值先设定为 0，等到调整到大致上可控制时，再依序调整 Ti\_Ki 值 (由小到大) 以及 Td\_Kd 值 (由小到大)。其中 Kc\_Kp 值为 1 则表示 100%，即对偏差值的增益为 1，小于 100%将对偏差值衰减，大于 100%将对偏差值放大。
- 当使用者选用 (PID\_MODE=1) 时，建议请使用在停电保持区之 D 缓存器来储存参数，以免自动调整过的参数因停电后而消失。经过自动调整过的参数，并不能保证一定适用于每个控制的环境，因此使用者当然可自行修改调整过的参数，不过建议最好只修改 Ti\_Ki 或 Td\_Kd 数值就好。
- 本指令动作须配合许多参数值控制，因此请勿随意设定参数值，以免造成无法控制之现象。

#### 4. 主机 DPID/DPIDE 参数手动调整技巧

假设控制系统之受控体  $G(s)$  的转移函数为一阶的函数  $G(s) = \frac{b}{s+a}$  (一般马达的模型均为此函数) , 命令值  $SV$  为 1 , 取样时间  $T_s$  为 10ms 。建议调整步骤如下 :

步骤 1 : 首先将  $K_I$  及  $K_D$  值设为 0 , 接着先后分别设定  $K_P$  为 5 、 10 、 20 及 40 , 并分别记录其  $SV$  及  $PV$  状态 , 其结果如图 1 所示。

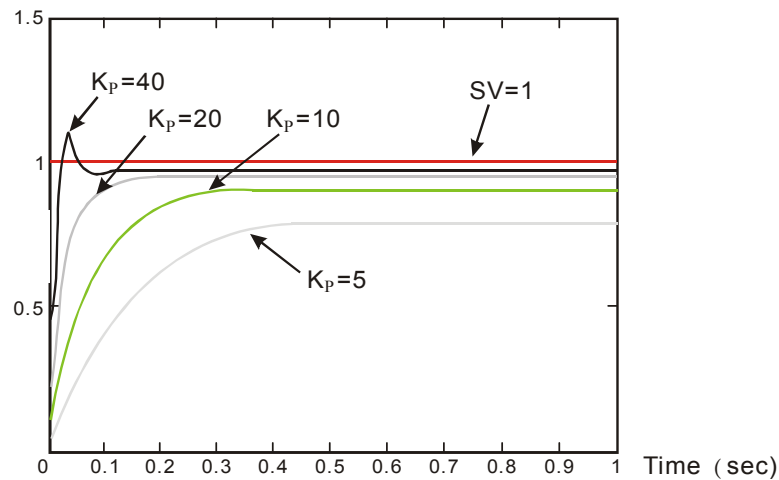


图 1

步骤 2 : 观察上图后得知  $K_P$  为 40 时 , 其反应会有过冲现象 , 因此不选用 ; 而  $K_P$  为 20 时 , 其  $PV$  反应曲线接近  $SV$  值且不会有过冲现象 , 但是由于启动过快 , 因此输出值  $MV$  瞬间值会很大 , 所以考虑暂不选用 ; 接着  $K_P$  为 10 时 , 其  $PV$  反应曲线接近  $SV$  值并且是比较平滑接近 , 因此考虑使用此值 ; 最后  $K_P$  为 5 时 , 其反应过慢 , 因此也暂不考虑使用。

步骤 3 : 选定  $K_P$  为 10 后 , 先调整  $K_I$  值由小到大 ( 如 1 、 2 、 4 至 8 ) , 以不超过  $K_P$  值为原则 ; 然后再调整  $K_D$  由小到大 ( 如 0.01 、 0.05 、 0.1 及 0.2 ) , 如果有启用  $T_f$  参数 , 则可以同时设定  $T_f = T_d/8$  来调整 , 以不超过  $K_P$  的 10% 为原则 ; 最后可得如图 2 之  $PV$  与  $SV$  的关系图。

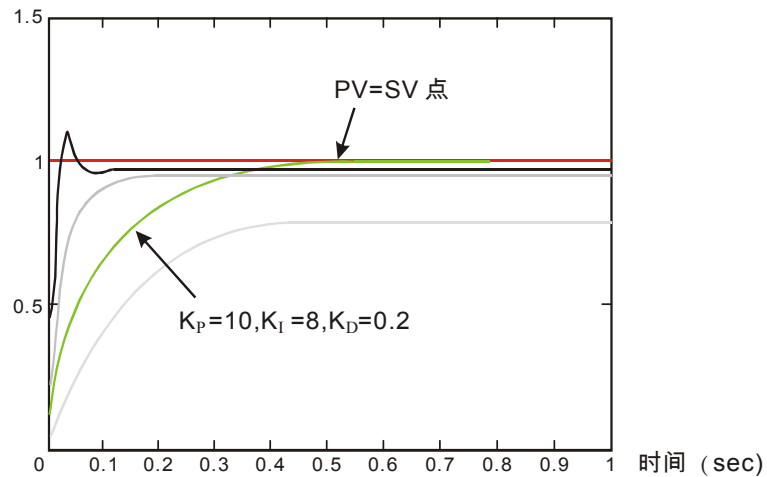


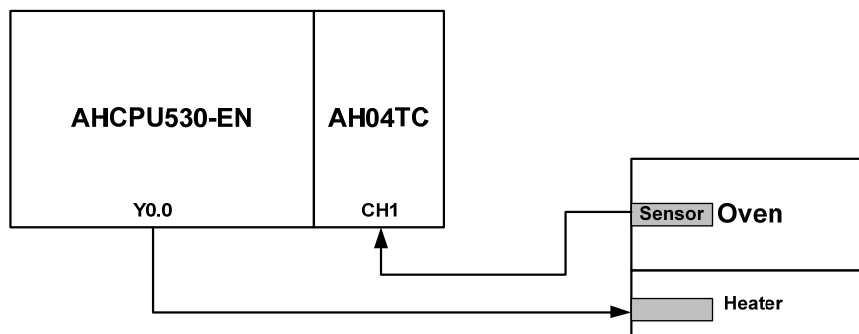
图 2

附注：本范例仅供参考，因此使用者还需依实际控制系统之状况，自行调整其适合之控制参数。

## 5. 范例说明

### 5.1 范例一：主机温度 DPID 控制 ( 主机输出控制小灯箱 )

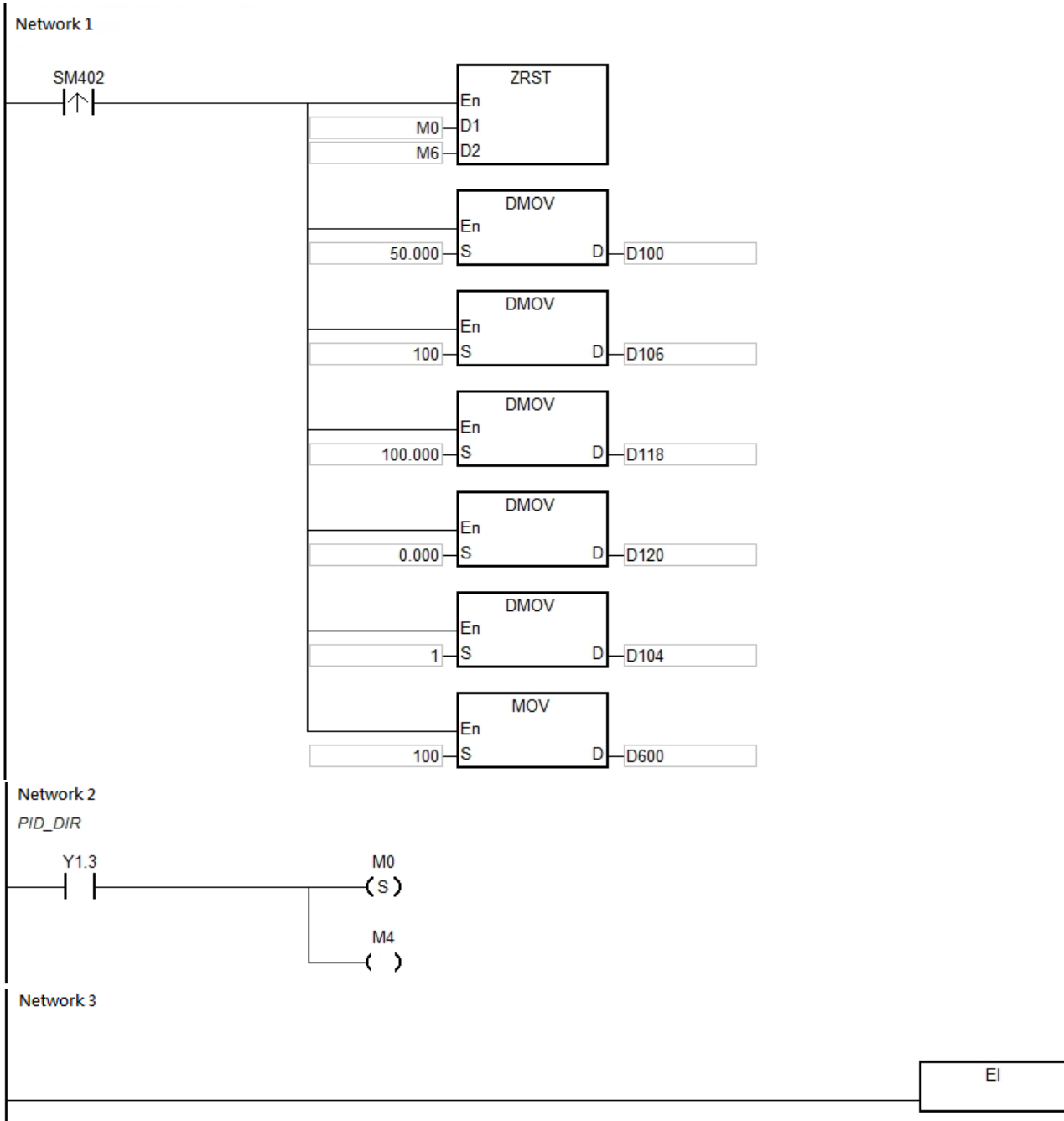
【系统架构】



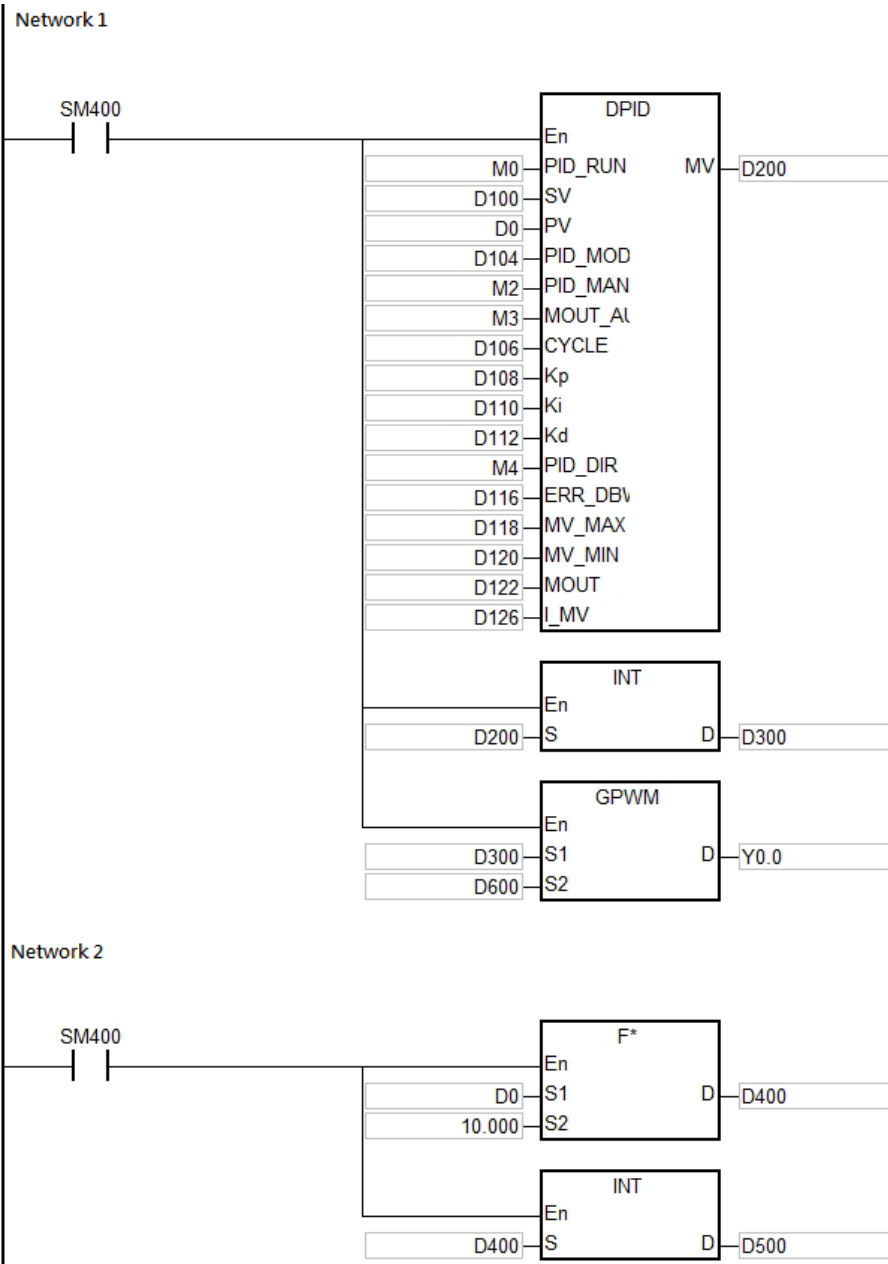
【控制要求】

本范例的控制环境为小灯箱，配置以AHCPU530-EN 主机搭配AH04TC模块K-type控制小灯箱，DPID控制为使用温度控制专用的自动调整参数功能 ( D104=0 ) 做初步调整，调整完毕时将自动改为1 ( D104 )，并且使用该自动计算出的参数 (  $K_P$ 、 $K_I$ 、 $K_D$  ) 实现对小灯箱的PID控制。

## 【控制程序】



DPID 功能写在时间中断 ( 1ms ) 子程序中

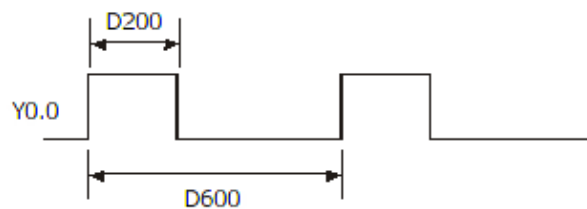


**【装置说明】**

PLC装置	控制说明
M0	启动控制模式
Y0.0	可调变脉冲宽度的脉冲输出
D200~D201	PID 运算输出结果
D100~D101	目标温度值
D600	GPWM 指令的运算周期
D106~D107	PID 取样时间
D104~D105	控制模式

**【程序说明】**

- 参数设定使用 ISPSOft 中之 HWCONFIG 设定，启动 04TC 模块 K-type。
- 启动 M0 及 M4，AH04TC 温度模块将小灯箱的现在值温度测得后传给 PLC 主机，主机控制模式使用温度自动调整参数功能 ( D104=0 ) 做初步调整，自动计算出最佳的 PID 温度控制参数，调整完毕后，自动修改动作方向为已调整过的温度控制专用功能 ( D104=1 )，并且使用该自动计算出的参数 ( D108  $K_p$ 、D110  $K_i$ 、D112  $K_D$  ) 实现对小灯箱温度的 PID 控制。
- 使用该温度自动调整参数功能进行 PID 运算，其输出结果 ( D200 ) 作为 GPWM 指令的输入，GPWM 指令执行后 Y0.0 输出可变宽度的脉冲控制加热器装置，从而自动实现对小灯箱温度的 PID 控制。





初步调整功能的实验结果如图 3 所示：

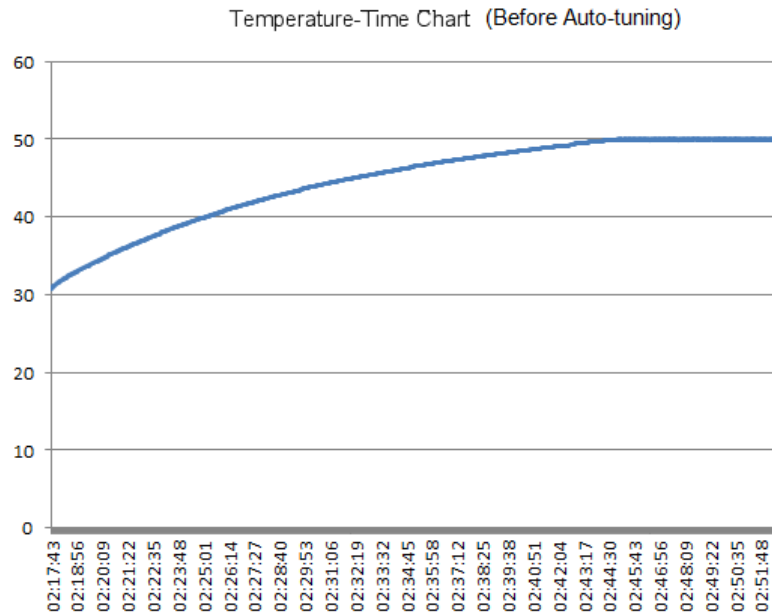


图 3

使用调整后参数做温度控制专用功能的实验结果如图 4 所示：

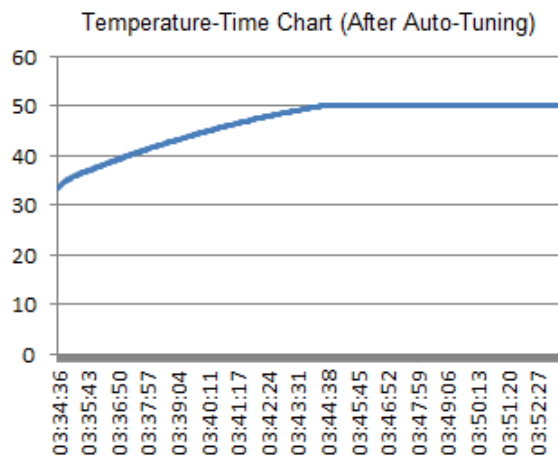
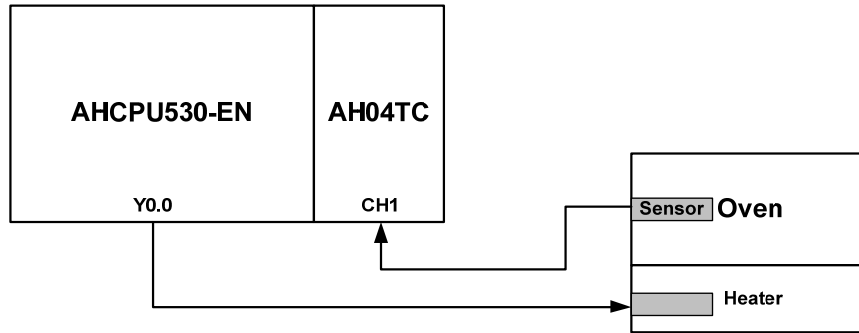


图 4

## 5.2 范例二：主机温度 DPIDE 控制 ( 主机输出控制小灯箱 )

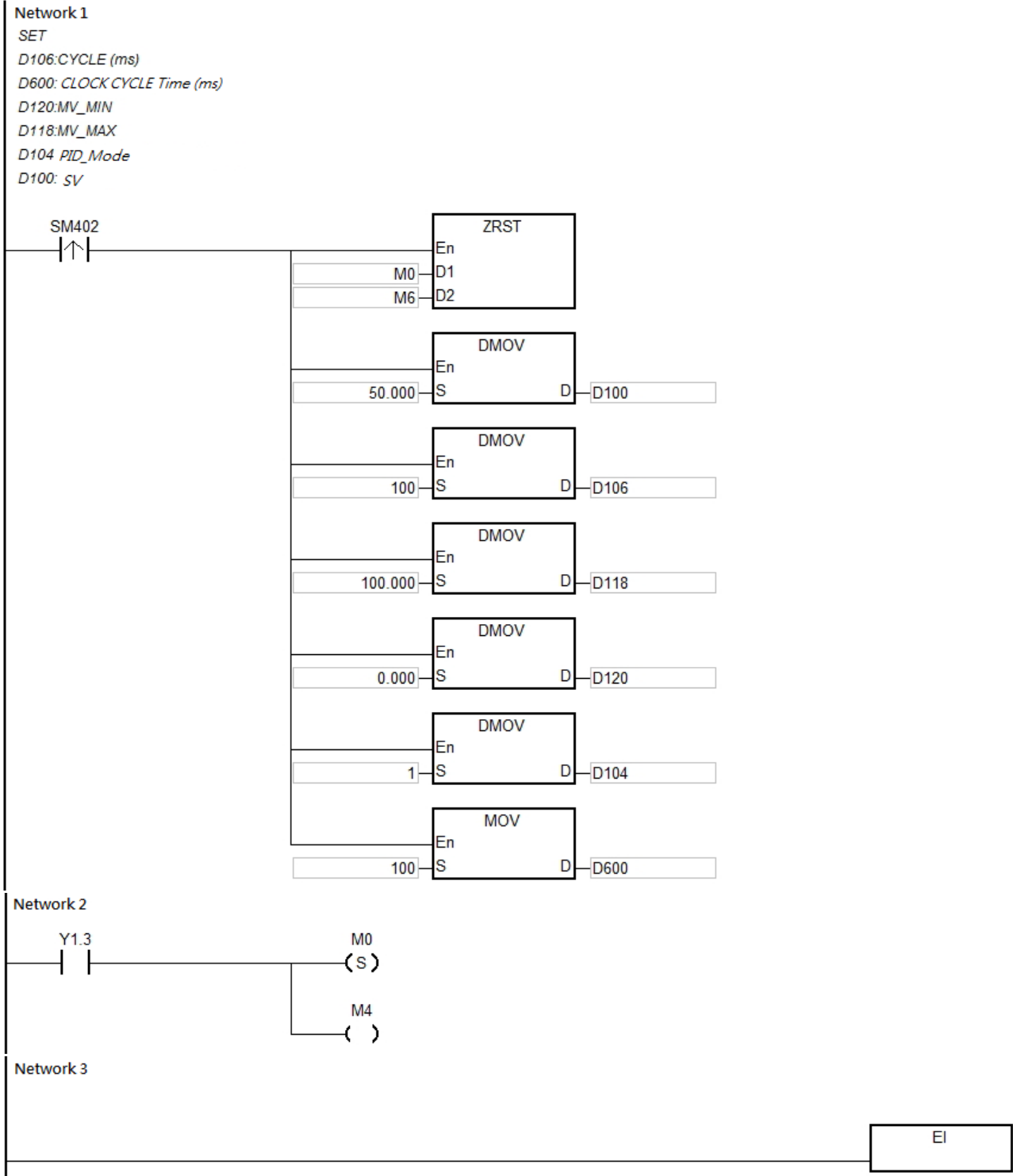
### 【系统架构】



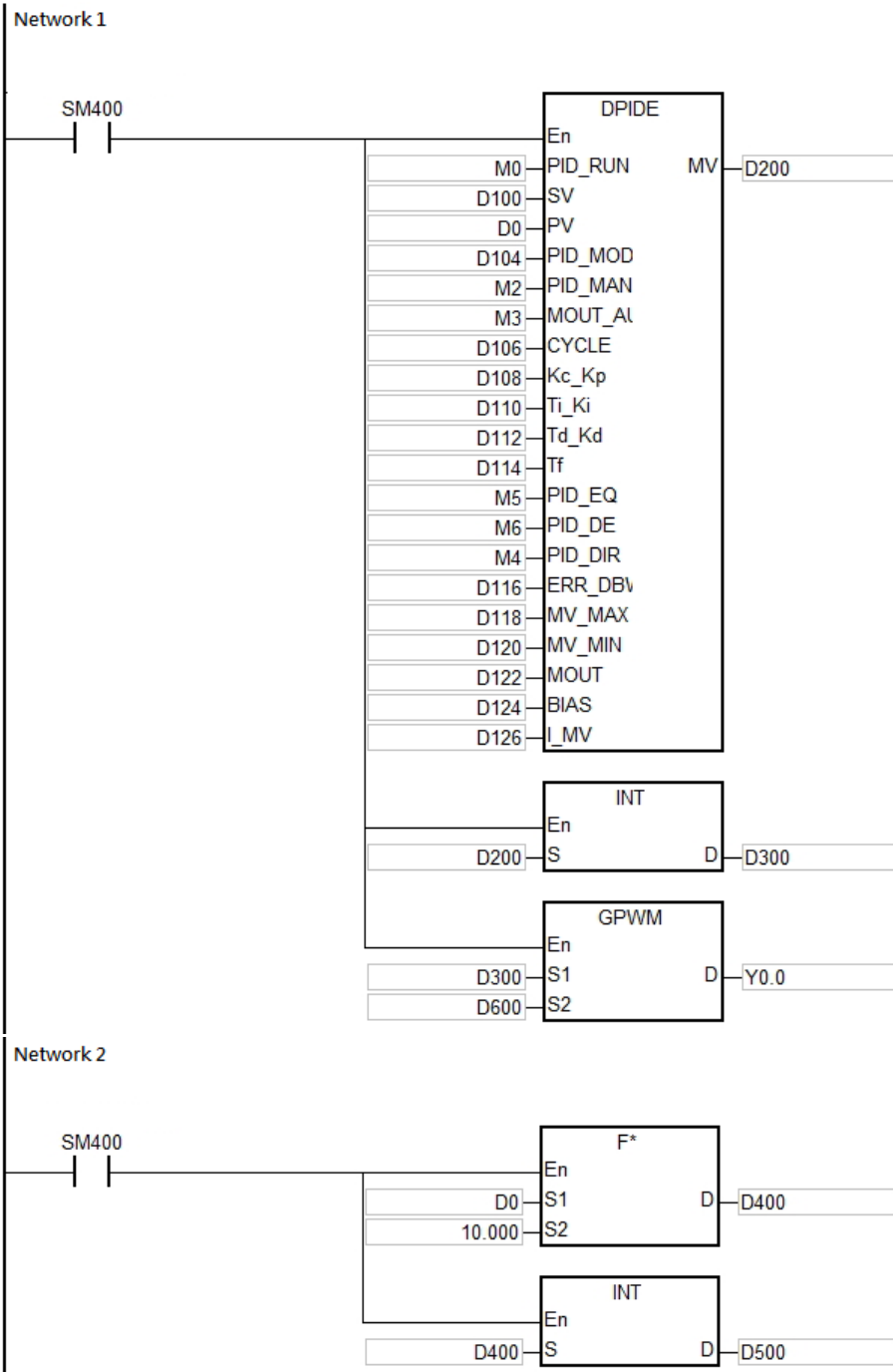
### 【控制要求】

本范例的控制环境为小灯箱，配置以AHCPU530-EN 主机搭配AH04TC模块K-Type控制小灯箱，DPIDE控制为使用温度控制专用的自动调整参数功能 ( D104=0 ) 做初步调整，调整完毕时将自动改为1 ( D104 )，并且使用该自动计算出的参数 (  $K_P$ 、 $K_I$ 、 $K_D$ 、 $T_f$  ) 实现对小灯箱的DPIDE控制。

【控制程序】



DPIDE 功能写在时间中断 ( 1ms ) 子程序中



**【装置说明】**

PLC装置	控制说明
M0	启动控制模式
Y0.0	可调变脉冲宽度的脉冲输出
D200~D201	PID 运算输出结果
D100~D101	目标温度值
D600	GPWM 指令的运算周期
D106~D107	PID 取样时间
D104~D105	控制模式

**【程序说明】**

- 参数设定使用 ISPSOft 中之 HWCONFIG 设定，启动 04TC 模块 K-type。
- DPIDE 运算后，其输出结果 ( D200 ) 作为 GPWM 指令的输入，GPWM 指令执行后 Y0.0 输出可变宽度的脉冲控制加热器装置，从而自动实现对小灯箱温度的 DPIDE 控制。
- 当温度无波动且持续维持 50°C 时，为自动计算出最佳的 DPIDE 温度控制参数 ( D108  $K_P$ 、D110  $K_I$ 、D112  $K_D$ 、D 114  $T_f$  )。实验结果之温度曲线如图 5。
- 使用 ISPSOft 读出模块 PID 温度控制参数 (  $K_P$ 、 $K_I$ 、 $K_D$ 、 $T_f$  )，将程序中模块对应之装置写入 (  $K_P$ 、 $K_I$ 、 $K_D$ 、 $T_f$  ) 之值，PID\_MODE ( D104 ) 设为 1 ( 手动模式 )。
- 利用计算出的参数  $K_P$ 、 $K_I$ 、 $K_D$ 、 $T_f$  数值，重新启动 M0，实验结果之温度曲线如图 6。

初步调整功能的实验结果如图 5 所示：

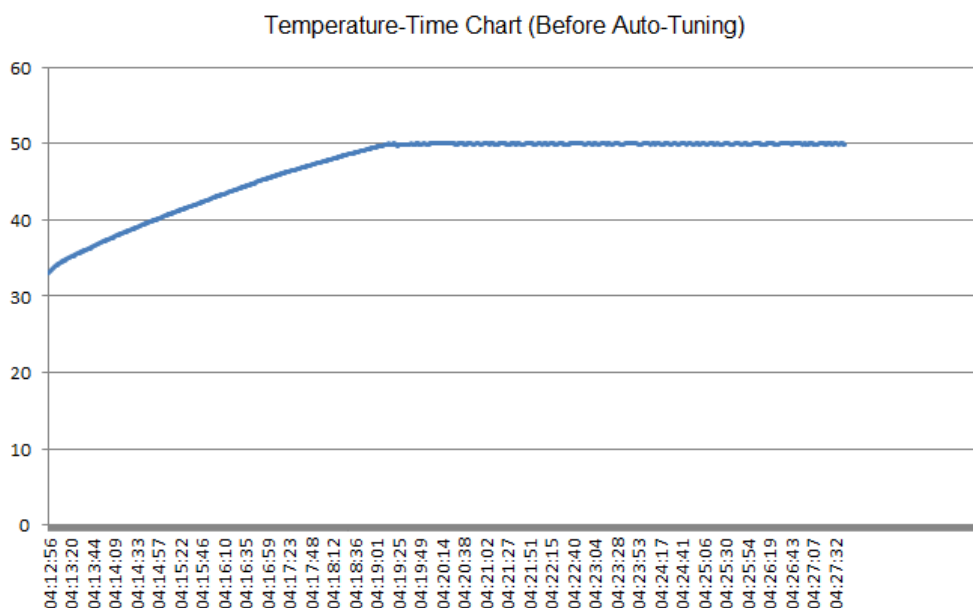


图 5

使用调整后参数做温度控制专用功能的实验结果如图 6 所示：

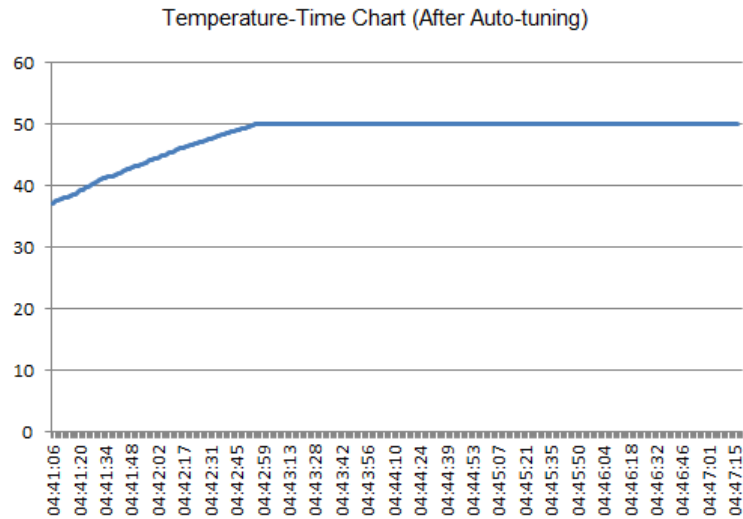
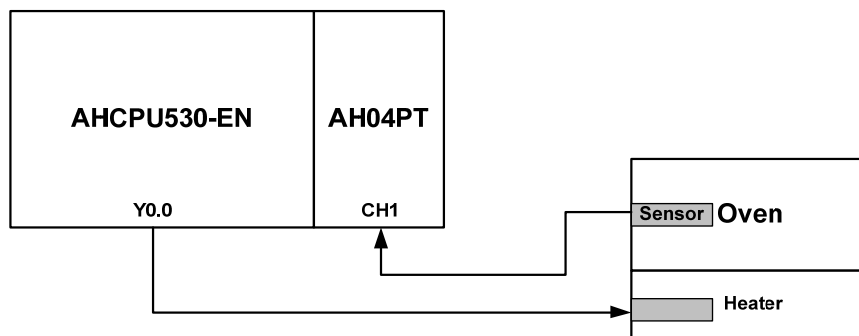


图 6

### 5.3 范例三：主机温度 DPID 控制（主机输出控制大温箱）

【系统架构】



【控制要求】

本范例的控制环境为大温箱，配置以AHCPU530-EN 主机搭配AH04PT模块PT100 模式控制大温箱，DPID控制为使用温度控制专用的自动调整参数功能（D104=0）做初步调整，调整完毕时将自动改为1（D104），并且使用该自动计算出的参数（ $K_P$ 、 $K_I$ 、 $K_D$ ）实现对大温箱的PID控制。

【控制程序】

参照5.1范例一之控制程序。

【装置说明】

参照5.1范例一之装置说明。

**【程序说明】**

- 参数设定使用 ISPSOft 中之 HWCONFIG 设定,启动 AH04PT 模块 PT100 模式。
- 其余说明参照 5.1 范例一之程序说明。

初步调整功能的实验结果如图 7 所示：

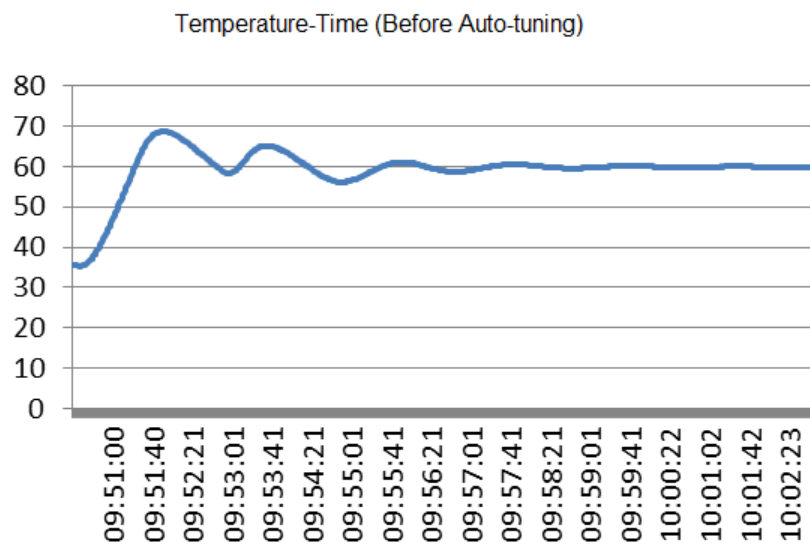


图 7

使用调整后参数做温度控制专用功能的实验结果如图 8 所示：

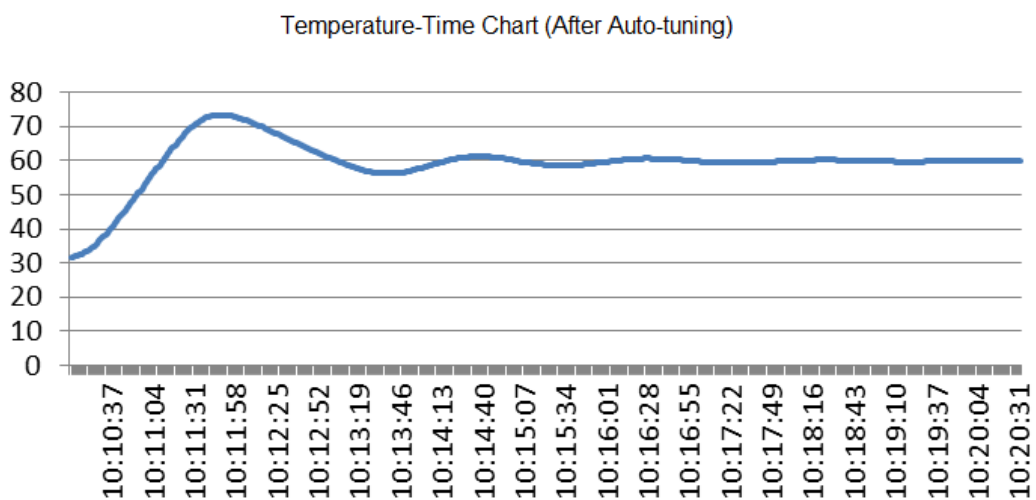
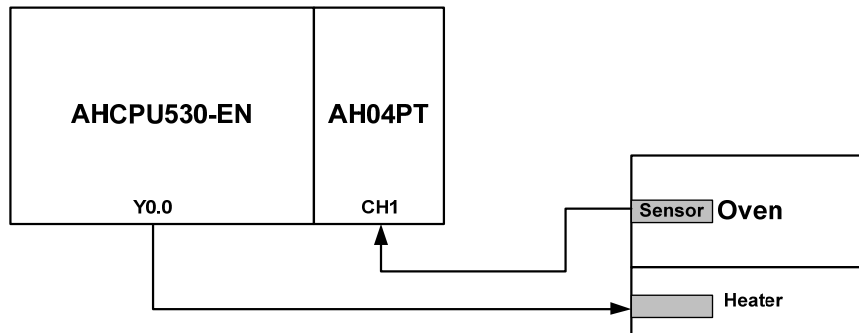


图 8

## 5.4 范例四：主机温度 DPIDE 控制 ( 主机输出控制大温箱 )

### 【系统架构】



### 【控制要求】

本范例的控制环境为大温箱，配置以AHCPU530-EN 主机搭配AH04PT模块PT100模式控制大温箱，DPIDE控制为使用温度控制专用的自动调整参数功能 ( D104=0 ) 做初步调整，调整完毕时将自动改为1 ( D104 )，并且使用该自动计算出的参数 (  $K_p$ 、 $K_i$ 、 $K_d$ 、 $T_f$  ) 实现对大温箱的DPIDE控制。

### 【控制程序】

参照5.2范例一之控制程序。

### 【装置说明】

参照5.2范例一之装置说明。

### 【程序说明】

- 参数设定使用 ISPSoft 中之 HWCONFIG 设定,启动 AH04PT 模块 PT100 模式。
- 其余说明参照 5.2 范例一之程序说明。



初步调整功能的实验结果如图 9 所示：

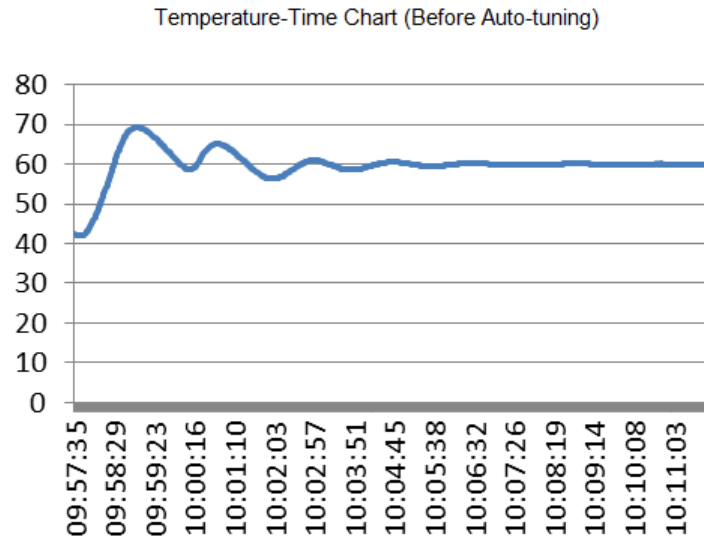


图 9

使用调整后参数做温度控制专用功能的实验结果如图 10 所示：

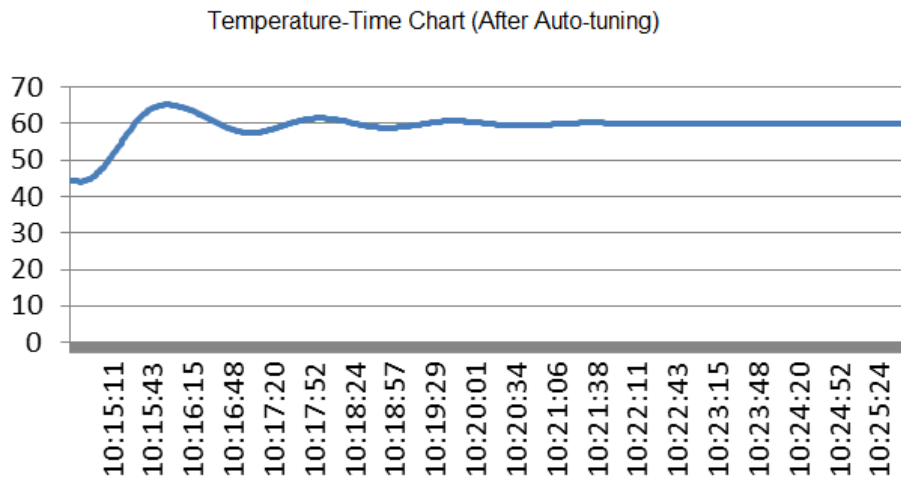
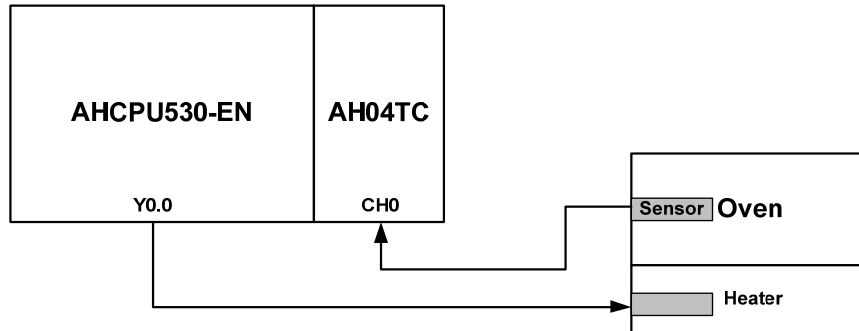


图 10

## 5.5 范例五：AH04TC 模块温度 DPID 控制

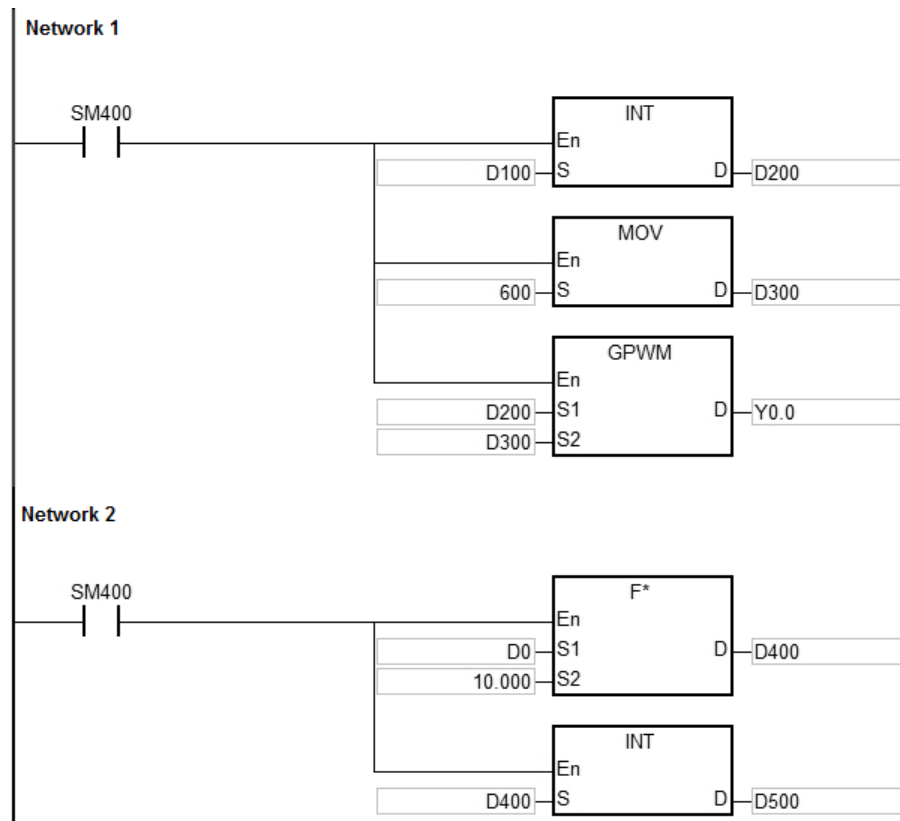
【系统架构】



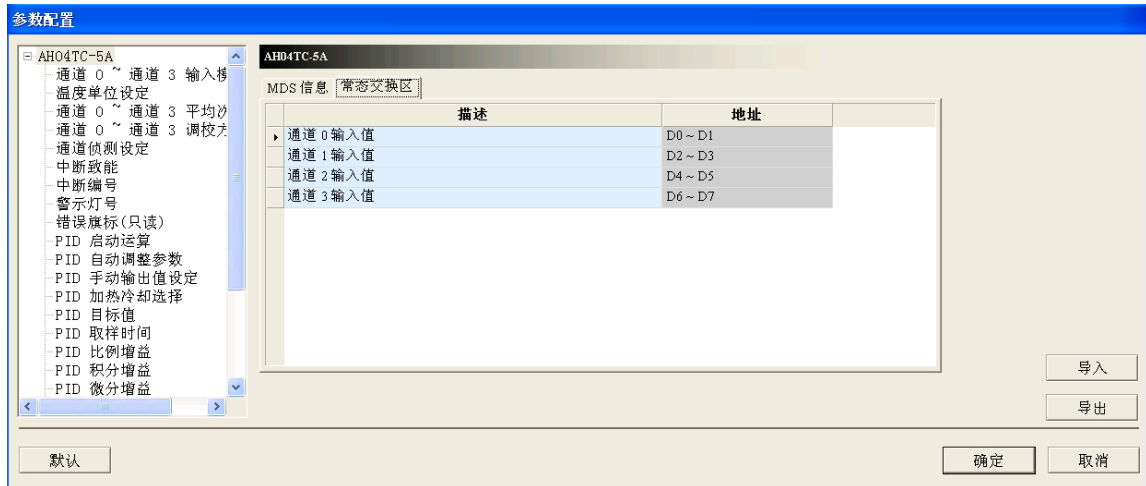
【控制要求】

本范例的控制环境为小灯箱，配置以AHCPU530-EN 主机搭配AH04TC模块控制小灯箱，PID控制为使用自动调整参数功能做调整，调整完毕后使用该自动计算出的参数 ( $K_P$ 、 $K_I$ 、 $K_D$ ) 实现对小灯箱的PID控制。

【控制程序】



【参数设定】



1. HWCONFIG设定AH04TC通道0模式为K-type。
2. PID启动运算选取启动。
3. PID自动调整参数选取启动。
4. PID目标值设定50.0。
5. 输出上限值设定2000.0。
6. 其余皆为默认值。
7. 比例增益D110、积分增益D112、微分增益D114自行设定缓存器。
8. 设定输出值缓存器D100。

【装置说明】

PLC装置	控制说明
D0~D1	通道 0 现在值
D100~D101	PID 输出值 ( MV )
D110~D111	$K_P$ 比例增益
D112~D113	$K_I$ 积分增益
D114~D115	$K_D$ 微分增益
D300	GPWM 指令的运算周期

【使用说明】

- 参数设定均使用 ISPSOFT 中之 HWCONFIG 做设定。
- PID 运算后，其输出结果 ( D100~D101 ) 作为 GPWM 指令的输入，GPWM 指令执行后 Y0.0 输出可变宽度的脉冲控制加热器装置，从而自动实现对小灯箱温度的 PID 控制。
- 当温度无波动且持续维持 50°C 时，为自动计算出最佳的PID温度控制参数 ( D110~D111  $K_P$ 、D112~D113  $K_I$ 、D114~D115  $K_D$  )。实验结果之温度曲线如图 11。

- 使用ISPSOft 读出模块PID温度控制参数 (  $K_P$ 、 $K_I$ 、 $K_D$  )，将程序中模块对应之装置写入 (  $K_P$ 、 $K_I$ 、 $K_D$  ) 之值，自动调整参数取消勾选 ( 将 Auto Tuning改为不动作 )。
- 利用计算出的参数  $K_P$ 、 $K_I$ 、 $K_D$ 数值，重新实验结果之温度曲线如图 12。

初步调整 ( Auto Tuning ) 功能的实验结果如图 11 所示：

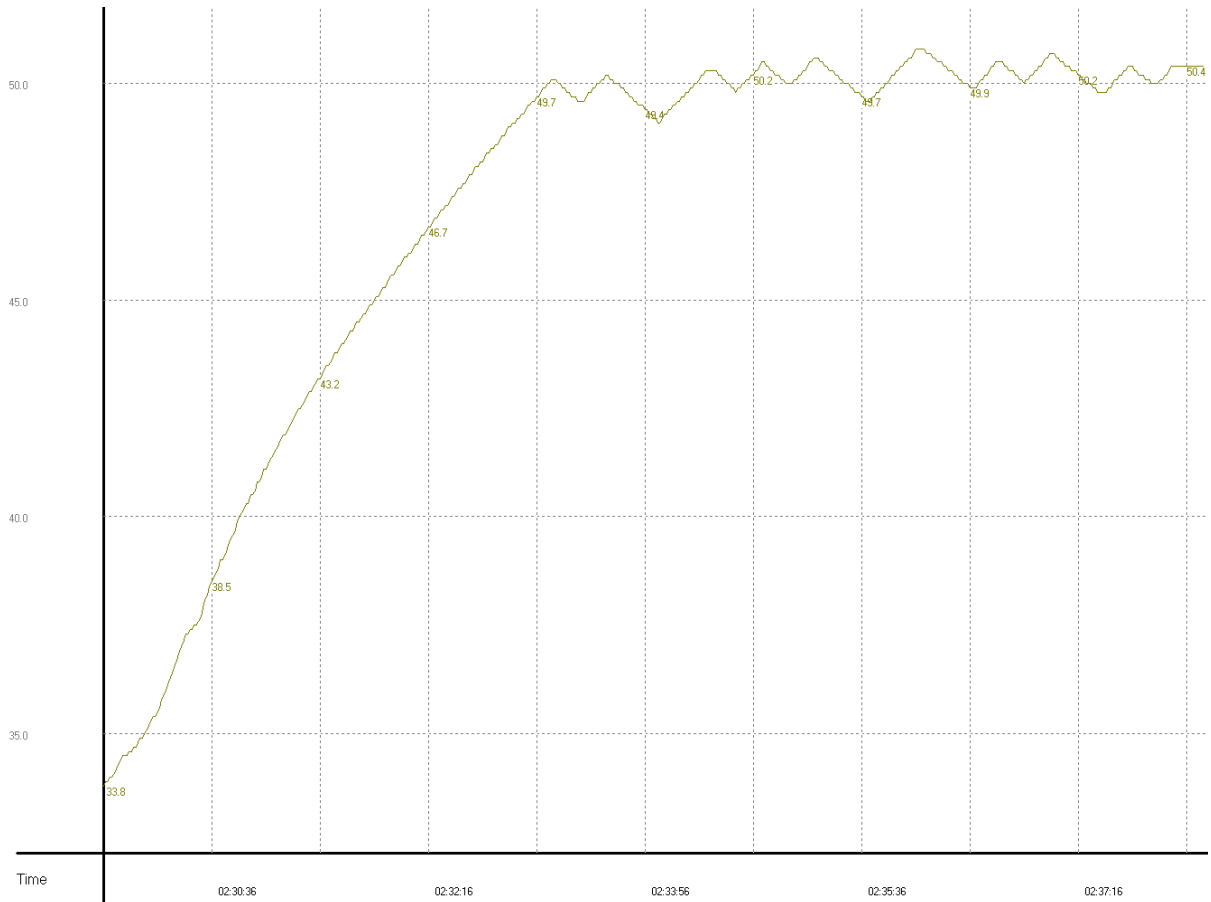


图 11

使用调整后参数 (  $K_P$ 、 $K_I$ 、 $K_D$  ) 实验结果如图 12 所示：

通道	起始温度	结束温度	$K_P$	$K_I$	$K_D$
CH0	33.8	50.0	102.836	6.170	2.57

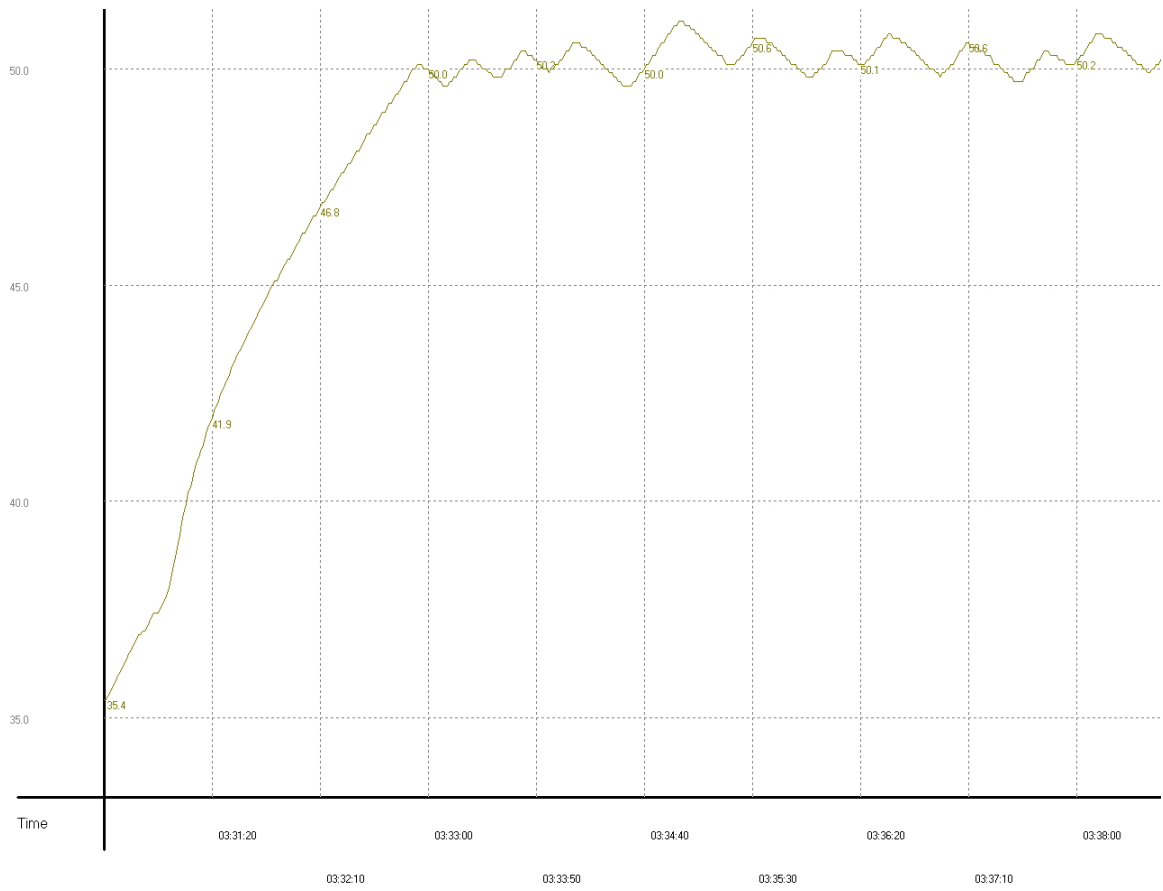


图 12