	ControlLogix Multizone Temperature Control Application Note – Plastics Industry	Creation Date: ***9/12/02***
	Performing Multizone Temperature Control with the Function Block PIDE in a ControlLogix Processor	

Software Used
All examples shown were developed using RSLogix 5000 Version 10 or higher.

Document Purpose
This document is not a manual or training material, but an Application Note, which could be useful in helping a Rockwell Automation customer who wishes to perform control on a multi-zone temperat ure device within a ControlLogix processor.

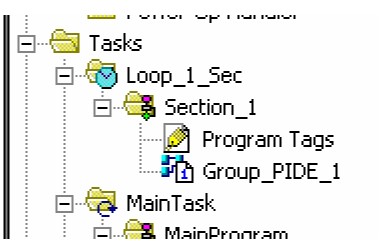
Intended Audience
This document is to be used by Rockwell Automation employees/customers supporting and selling ControlLogix systems.

Concept of Application Note
In some application scenarios it is desirable to

- Group relevant temperature zones together such that new sections can be easily added.
- Use Time Proportioned Outputs so that discrete actuators can be used.
- Program autotunes such that simultaneous autotunes occur within a sectional group.
- Configure PIDEs appropriately for high lag systems.

Program Structure

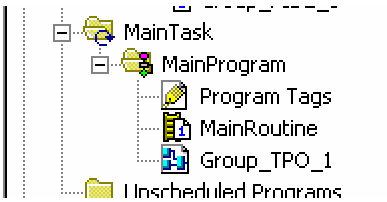
The individual PIDE instructions are executed from a periodic task scheduled to be executed at one second intervals. Typically, closed loop heat (cool) control does not need to be executed any faster than this as the deadtime and time constants of these systems are much greater. In extrusion or injection temperature control applications it is very rare that a regulatory temperature loop needs to be executed faster than this.



The one second task executes the 'Section_1' program every second which in turn executes the 'Group_PIDE_1' routine. The routine has multiple sheets which contain the PIDEs necessary to regulate the individual zones. Within the routine there is one PIDE (representing one zone) per sheet. There are as many sheets in the routine as required to regulate the zones of the section of the machine.
Note that the PIDE backing tags must be Controller Scoped so that the accompanying SRTP instructions can access the PIDE output.

If another section of the machine is added (i.e. a second extruder barrel) one can simply copy the 'Section_1' program and paste another section into the 'Loop_1_Sec' task.

The Time Proportioned Outputs are accomplished through the use of the SRTP instruction in function block. Since the discrete actuation must be executed fast enough to have sufficient resolution to control, these SRTP instructions have been executed in a routine which is called (via JSR) by the main routine of the continuous task.



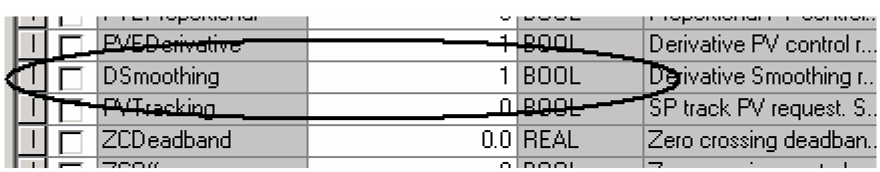
Each sheet of the 'Group_TPO_1' routine contains the logic necessary for one zone of the machine. If another section of the machine is added (i.e. a second extruder barrel) one can simply copy the 'Group_TPO_1' routine and paste another section into the 'MainProgram' program. It would be necessary to include another JSR to this new routine in the 'MainRoutine.'
Note that the Time Proportioned Output related programming could be included in another periodic task. However, that task would have to be scheduled to run at a rate faster than 100 milliseconds to maintain the resolution required for appropriate control.

PIDE Configuration

There are two configurations typically used in these applications: heat or cool only (where only heating or cooling is used) and heat / cool (where heating and cooling are both used on a single zone).
In a cool only scenario the same general configuration applies with the action of the PIDE changed to calculate error as Error = PV-SP.

In the example program:
The first and fourth zones (PIDEs) are configured for heat only operation.
The second and third zones (PIDEs) are configured for heat /cool operation.

In all PIDEs used in this type of an application the 'DSmoothing' parameter of the PIDE should be set to one (1).



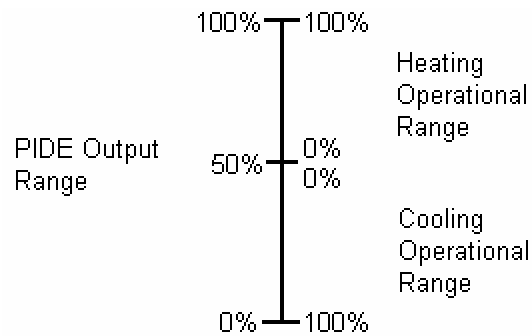
This parameter determines if the derivative action will have a first order filter applied to its computation. Typically high derivative gains are used to compensate for large dead times in the system. As such, it needs to be filtered so that the output does not jump due to the large derivative component.

Heat (or Cool) Only

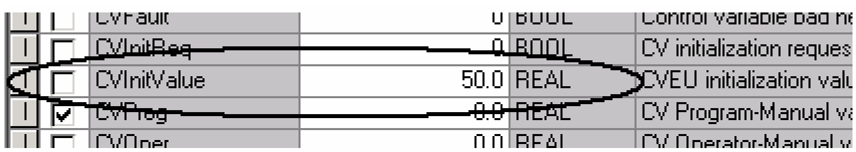
Heat only loops use the entire range of the output for the heat actuation, therefore no further configuration is required.
If a cool only loop is required, set the 'ControlAction' parameter to one (1).

Heat / Cool

Heat / Cool loops are 'split-ranged' for operation with two outputs from a single PIDE.
50% to 100% PIDE output is mapped to 0% to 100% heating capacity.
50% to 0% PIDE output is mapped to 0% to 100% cooling capacity.



In order for proper initialization to occur in the PIDE the 'CVInitValue' parameter must be set to 50.0. Since 50.0% PIDE output represents 0% heat and 0% cool the PIDE must start at a value of 50.0% output so that no heat or cool actuation is occurring.



Setting this will initialize the PIDE to 50.0% output when the controller is placed in Run mode.

SRTP Configuration

Common Parameters

The SRTP instruction converts the supplied input value to a duty cycle for a discrete bit. As such, a maximum time needs to be entered into the 'CycleTime' parameter. In the example program this parameter has been set to five (5) seconds. Typical values for injection or extrusion temperature control applications range from one (1) second to ten (10) seconds.

Heat (or Cool) Only

The parameters relevant to immediate operation are as follows: MaxHeatIn, MinHeatIn, MaxHeatTime, MinHeatTime. These first two parameters define the span of the input signal which relates to the span of the CycleTime. The latter two parameters define the active span within the cycle range; a pulse will not occur if the pulse time calculated is either greater than the MaxHeatTime

or less than the MinHeatTime. Note: For cool only applications the Cool parameters may be used for clarity.
The example program heat-only zones are configured to map 0% to 100% input to 0 seconds to 5 seconds of output pulse.

I		CycleTime	5.0	REAL	The perioc
I		MaxHeatIn	100.0	REAL	Maximum I
I		MinHeatIn	0.0	REAL	Minimum h
I		MaxCoolIn	0.0	REAL	Maximum
I		MinCoolIn	0.0	REAL	Minimum c
I		MaxHeatTime	5.0	REAL	Maximum I
I		MinHeatTime	0.0	REAL	Minimum h
I		MaxCoolTime	0.0	REAL	Maximum
I		MinCoolTime	0.0	REAL	Minimum c

Heat / Cool

The parameters relevant to immediate operation are as follows: MaxHeatIn, MinHeatIn, MaxCoolIn, MinCoolIn, MaxHeatTime, MinHeatTime, MaxCoolTime, MinCoolTime. These first four parameters define the span of the input signal which relates to the span of the CycleTime. The latter four parameters define the active span within the cycle range; i.e. a pulse will not occur if the pulse time calculated is either greater than the MaxHeatTime or less than the MinHeatTime.
The example program heat / cool zones are configured to map 50% to 100% input to 0 seconds to 5 seconds of heat output pulse, and maps 50% to 0% input to 0 seconds to 5 seconds of cool output pulse.

I		CycleTime	5.0	REAL	The period of th
I		MaxHeatIn	100.0	REAL	Maximum heat i
I		MinHeatIn	50.0	REAL	Minimum heat ir
I		MaxCoolIn	0.0	REAL	Maximum cool i
I		MinCoolIn	50.0	REAL	Minimum cool ir
I		MaxHeatTime	5.0	REAL	Maximum heat
I		MinHeatTime	0.0	REAL	Minimum heat ti
I		MaxCoolTime	5.0	REAL	Maximum cool t
I		MinCoolTime	0.0	REAL	Minimum cool ti

AutoTune Management

The program to initiate AutoTunes on the individual zones is configured to either AutoTune all zones simultaneously as a group or individually as needed. The interface bits to request AutoTune are programmed to accept one-shot-latch writes and clear them upon completion of the request (one-shot-latch or retentative write operation is common of most HMI packages and hard operator interfaces).
The user interface bits for the AutoTune section are as follows:
'Group_x_Tune_Request' is the Boolean tag to request Autotune for group number 'x.' If it is desired to perform an Autotune on an entire group of zones then assert this Boolean.

‘Group_x_Zone_y_Tune_Request’ is the Boolean tag to request Autotune for zone number ‘y’ in group number ‘x.’ If it is desired to perform an Autotune on an individual zone then assert this Boolean.
The program will reset these bits when the requested Autotune has either completed or has aborted due to an error or a user abort request.

Specific operational details of this program section and configuration of individual Autotune parameters are covered in **Rockwell Automation KnowlegeBase document A20456899**. Please refer to this document for a detailed explanation of the operation of this program section.

The following configuration settings should be used to obtain the best Autotune results:
.ProcessType Non-Integrating
.PVTuneLimit Set to a relatively safe temperature for the specific system.
.StepSize Set to be as large an output (CV) step as possible for the best Autotune result
.ResponseSpeed The specific response speed setting used in determining the resulting gains is solely determined by the user. For these types of systems, the Fast setting is typically used.

Tuning a ‘group’ of related or adjacent temperature zones simultaneously tends to produce better results since the adjacent heating effects of one zone on another is accounted for using this methodology. Also, if only one ‘group’ of zones is tuned at a time, the Autotune tags can be reused for another group of zones. This is an effective way to keep the memory requirement for Autotune tags low as long as the groups do not need to be tuned simultaneously.

In the example program, the program section pertaining to the Autotune management is executed in the Continuous Task in the Main Program. It is not necessary for this code to be placed in the Continuous Task. It will work just as well if programmed in a periodic task running at a much slower rate.