PID - Rung #3:1 - PD9:0

Controller Gain, Kc: 3276.7	Setpoint: 2500
Reset Term, Ti: 0.00	Setpoint MAX(Smax): 16384
Rate Term, Td: 0.00	Setpoint MIN(Smin): 0
Loop Update Time: 1.00	Process Variable PV: 2383
Control Mode: E = SP - PV	Control Output CV (%): 100
PID Control: Auto	Output Max CV(%): 100
Time Mode: STI	Output Min CV(%): 0
Limit Output CV: No	Scaled Error: 117
Deadband: 0	Feed Forward Bias: 0

PID MAIN

The programs uses a PID to control a simulated temperature system that is modeled after Ron Beaufort's 'Hot Rod' system. The 'Hot Rod' System had two time constants of 1 and 3.41 minutes and a gain of 3.69 degrees/percent output. To experiment with the PID one only needs to change the Setpoint at N7:0 and view the response on the PID trend. Note, all the integers are in X10 integer format. 1000 = 100.0 degrees. There fore to set the set point to 200 degrees one must put 2000 in register N7:0. There isn't much in the main ladder as the PID and Simulator are in the STI subroutines. The interrupts occur every second.

Peter Nachtwey 20040914

0000

First Pass
S:1/15

Call this routine
during first scan.
PID_INIT

JSR
Jump To Subroutine
SBR File Number U:4

0001

END

PID Update Routine.

This routine should be called by a STI or activated using a timer. The purpose of this routine is to calculate an analog output that will control the temperature at the desired set point. This first rung conditionally activates the simulator. The simulator calculates PV0 in degrees in F8:0. The PV in degrees should then by copied or scaled into N7:1. In the case below the temperature is in integer *10 format.

0000

SBR
Subroutine

Heater simulation enable.
SIMULATE
B3/2

Temperature System Simulator.
SIMULATOR

JSR
Jump To Subroutine
SBR File Number U:5

PV in X10 integer format. 1000 = 100.0 degrees
PV

MUL
Multiply
Source A F8:0
237.9172<
Source B 10.0
10.0<
Dest N7:1
2383<

0001

Scaled Set Point in X10 integer format.
PID0.SPS

MOV
Move
Source N7:0
2500<
Dest PD9:0.SPS
2500<

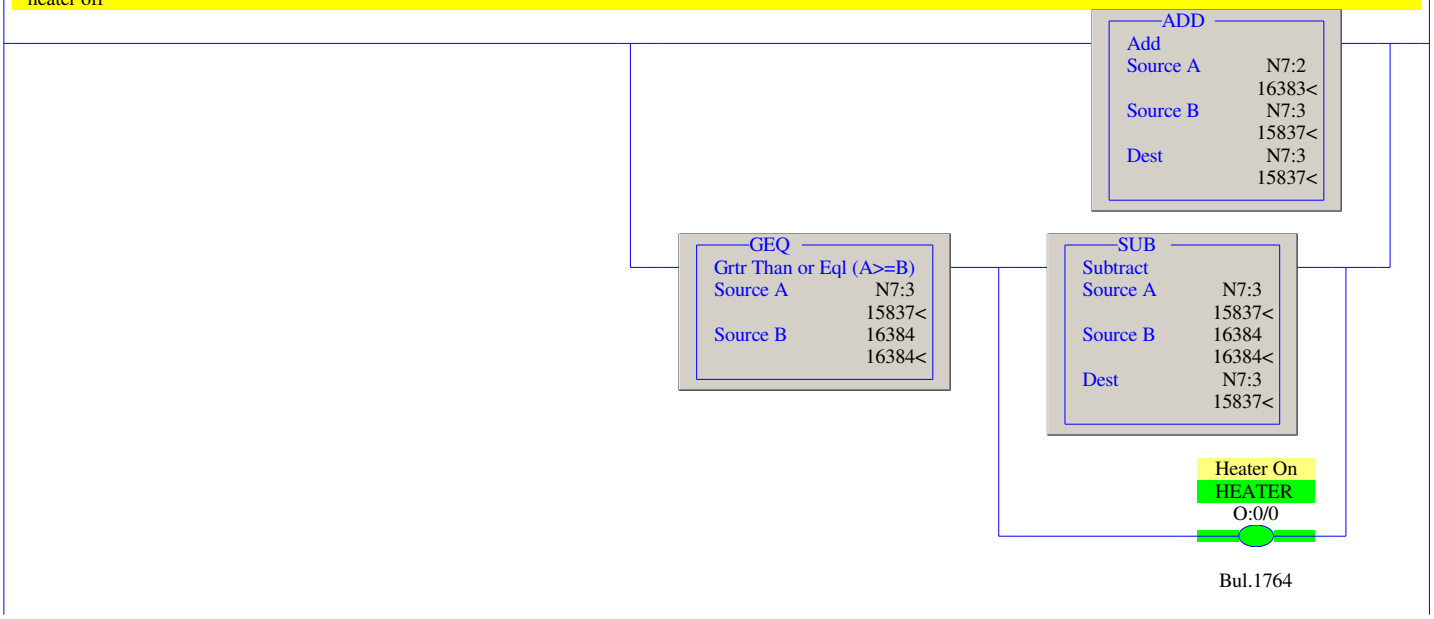
PID0

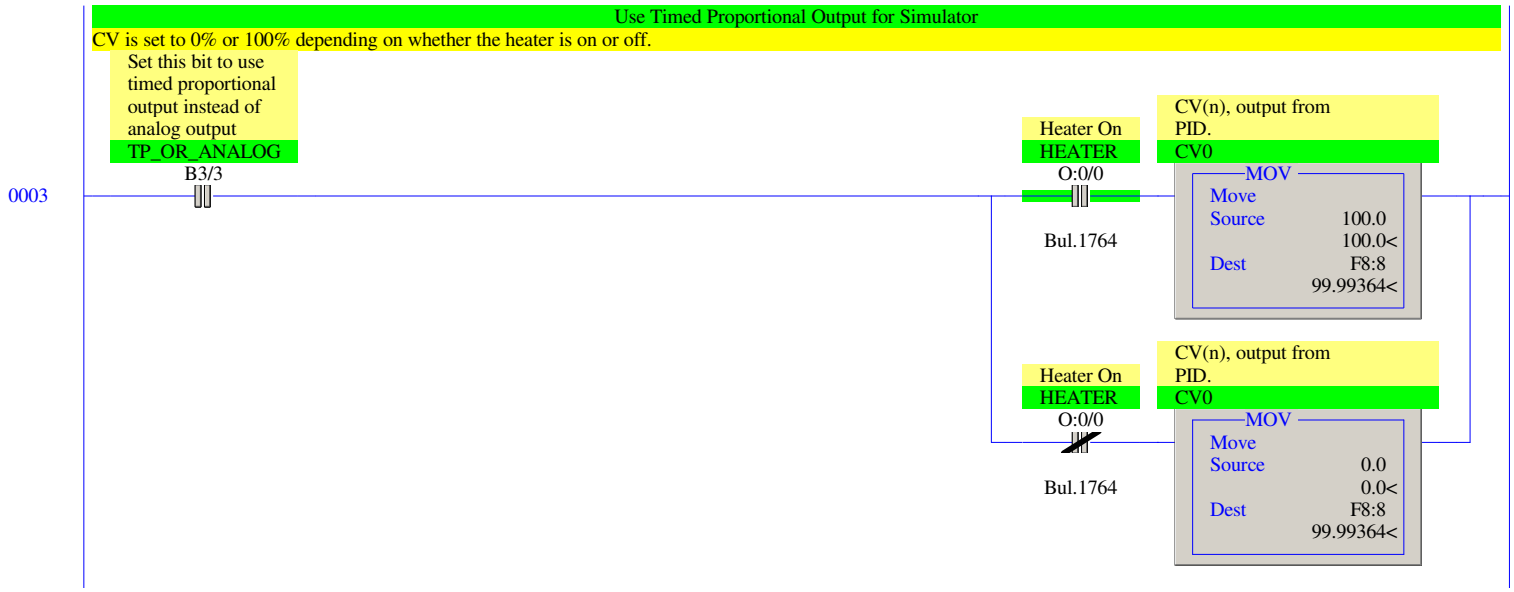
PID
PID
PID File PD9:0
Process Variable N7:1
Control Variable N7:2
Setup Screen

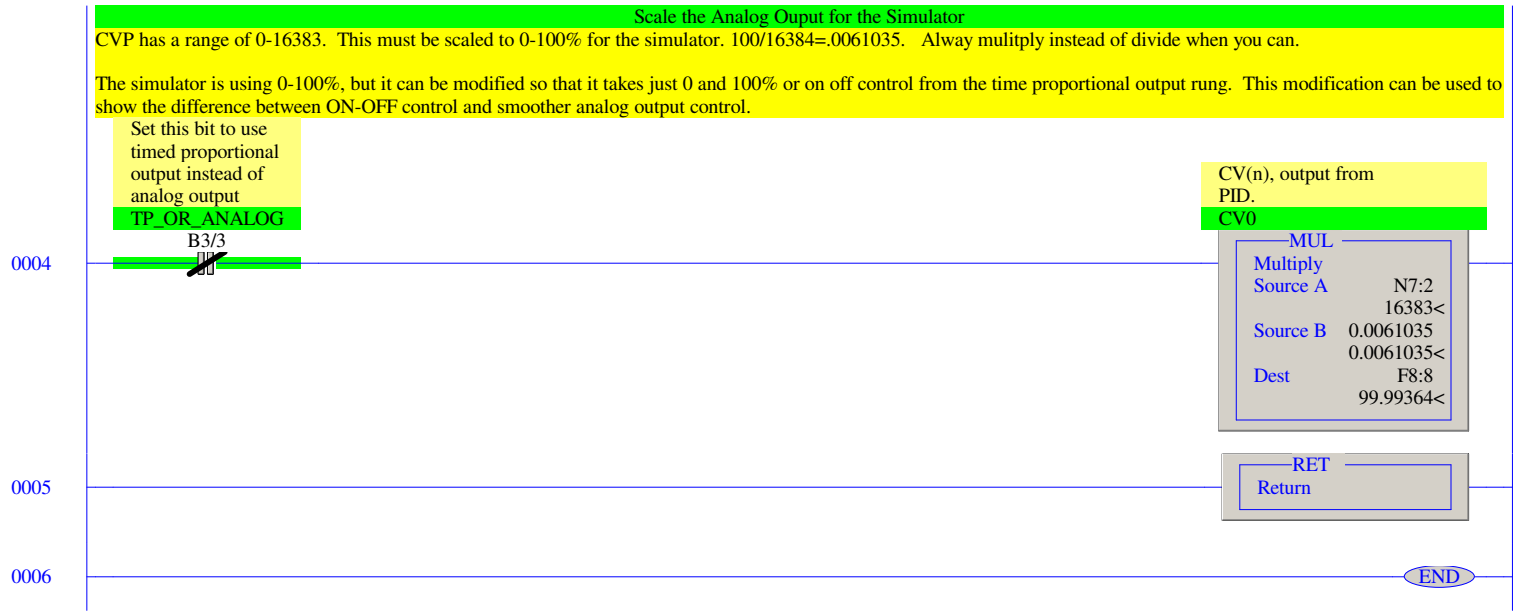
Convert Analog Output to ON-OFF Control

Use the time proportion technique.
accum = accum + cv
if accum >= 16384 then
  heater on
  accum = accum - 16384
else
  heater off

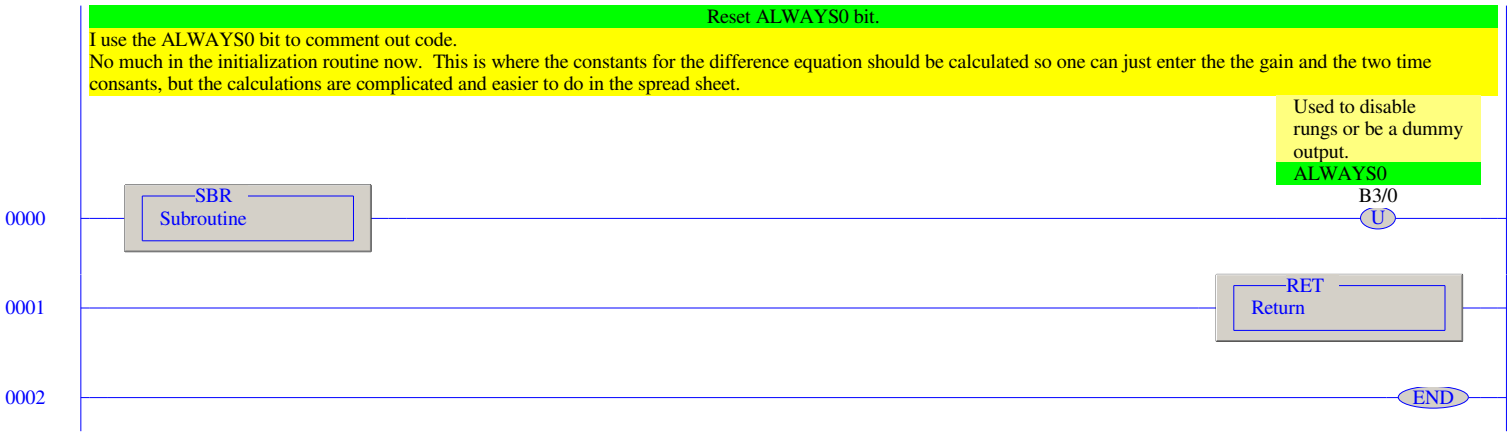
0002




Use Timed Proportional Output for Simulator
CV is set to 0% or 100% depending on whether the heater is on or off.
Set this bit to use timed proportional output instead of analog output
TP_OR_ANALOG
B3/3
0003
Heater On
HEATER
O:0/0
Bul.1764
CV(n), output from PID.
CV0
MOV
Move
Source
100.0
100.0<
Dest
F8:8
99.99364<
Heater On
HEATER
O:0/0
Bul.1764
CV(n), output from PID.
CV0
MOV
Move
Source
0.0
0.0<
Dest
F8:8
99.99364<


Scale the Analog Ouput for the Simulator
CVP has a range of 0-16383. This must be scaled to 0-100% for the simulator. 100/16384=.0061035. Alway mulitply instead of divide when you can.
The simulator is using 0-100%, but it can be modified so that it takes just 0 and 100% or on off control from the time proportional output rung. This modification can be used to show the difference between ON-OFF control and smoother analog output control.
Set this bit to use timed proportional output instead of analog output
TP_OR_ANALOG
B3/3
0004
CV(n), output from PID.
CV0
MUL
Multiply
Source A
N7:2
16383<
Source B
0.0061035
0.0061035<
Dest
F8:8
99.99364<
0005
RET
Return
0006
END

LAD 4 - PID_INIT - PID INITIALIZATION --- Total Rungs in File = 3



Temperaure System Simulator.

This subroutine computes the measured temperature of the plant PV using the formula G/(tau0*s+1)*(tau1*s+1) in the s domain.
This is a second order lag system implemented as a difference equation.
It uses the difference equation PV(n) = A1*PV(n-1)*A2*PV(n-2)+B1*CV(n-1)+B2*CV(n-2).
The values of the coefficents are documented below and in the description.
PV(n) is a floating point value in degrees.
CV(n) is from the PID or Manual control in the range of 0-16383 DAC counts or 0-100%

Use the Excel spreadsheet PID-TEMP-ML.XLS to calculate Simulator Coefficients as follows:
T is the update time in seconds
G is the gain in degrees /% output G=3.755449
a=1/Tau0 Tau0 is time constant 0 in seconds. Tau0 = 70.658829 seconds
b=1/Tau1 Tau1 is time constant 1 in seconds. Tau1=145.668301 seconds
Q=exp(-a*T)
R=exp(-b*T)
A=(b*(1-Q)-a*(1-R))/(a*b*(b-a))
B=(a*Q*(1-R)+b*R*(1-Q))/(a*b*(b-a))
A1=-Q*R
A2=Q+R
B1=G*A*a*b
B2=G*B*a*b
C=(1-A1+A2)*Ta Ta is the ambient temperature.
The values above can't be easily calculated on a uLogix because it doesn't have a exp(routine) or CPT block.

The first rung shifts the PVs and CVs.

PV(n-2)=PV(n-1)
PV(n-1)=PV(n)

and CVs
CV(n-3) = CV(n-2)
CV(n-2) = CV(n-1)
CV(n-1) = CV(n)
The number of CVs to shift is dependent on the dead time.
Note. I had to use two COP blocks because one can't COP F8:8 F8:9 23 without copying F8:8 to all registers. You would think this can be done with one COP block.

0000

SBR
Subroutine

y(n-2)
PV2
MOV
Move
Source F8:1
237.5963<
Dest F8:2
237.2894<

y(n-1)
PV1
MOV
Move
Source F8:0
237.9172<
Dest F8:1
237.5963<

#TEMP10
COP
Copy File
Source #F8:8
Dest #F10:0
Length 23

CV(n-1). CV
calculated from
previous update
#CV1
COP
Copy File
Source #F10:0
Dest #F8:9
Length 23

Calculate PV(n)

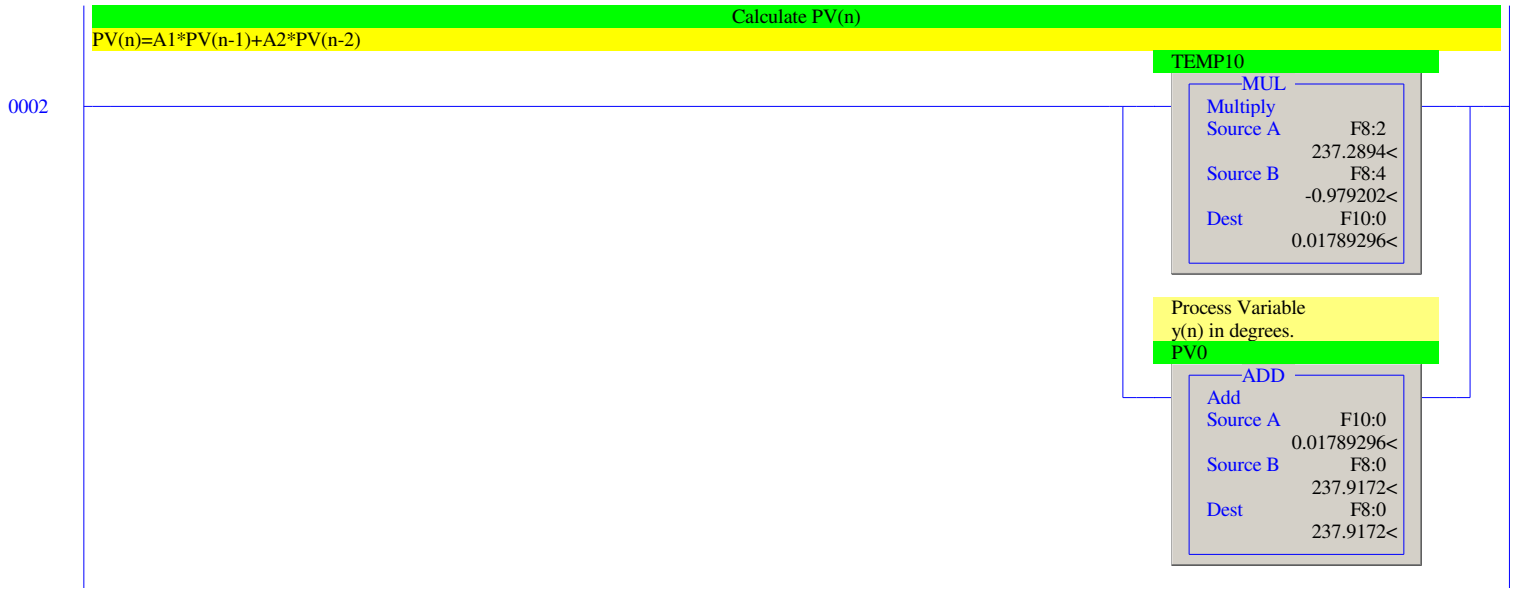
PV(n)=A1*PV(n-1)

0001

Process Variable
y(n) in degrees.
PV0

MUL
Multiply

Source A	F8:1
	237.5963<
Source B	F8:3
	1.979106<
Dest	F8:0
	237.9172<


Calculate PV(n)
PV(n)=A1*PV(n-1)+A2*PV(n-2)
0002
TEMP10
MUL
Multiply
Source A F8:2
237.2894<
Source B F8:4
-0.979202<
Dest F10:0
0.01789296<
Process Variable
y(n) in degrees.
PV0
ADD
Add
Source A F10:0
0.01789296<
Source B F8:0
237.9172<
Dest F8:0
237.9172<

Calculate PV(n)

PV(n)=A1*PV(n-1)+A2*PV(n-2)+B1*CV(n-1)
CV(n-1) is set by PID and either the Timed Portional or Analog output. CV(n-1) has a range of 0 to 100%
B1 = G*A*a*b

0003

TEMP10

MUL
Multiply
Source A F8:21
99.99364<
Source B F8:5
0.0001811589<
Dest F10:0
0.01789296<

Process Variable
y(n) in degrees.

PV0

ADD
Add
Source A F10:0
0.01789296<
Source B F8:0
237.9172<
Dest F8:0
237.9172<

Calculate PV(n)

PV(n)=A1*PV(n-1)+A2*PV(n-2)+B1*CV(n-1)+B2*CV(n-2)
B2 = G*B*a*b

0004

TEMP10

MUL
Multiply
Source A F8:22
99.99364<
Source B F8:6
0.000178941<
Dest F10:0
0.01789296<

Process Variable
y(n) in degrees.

PV0

ADD
Add
Source A F10:0
0.01789296<
Source B F8:0
237.9172<
Dest F8:0
237.9172<

Calculate PV(n)

PV(n)=A1*PV(n-1)+A2*PV(n-2)+B1*CV(n-1)+B2*CV(n-2)+C
C is adjusted so PV goes to the ambient temperature when there is no control signal.
C=(1-A1+A2)*Ta

Process Variable
y(n) in degrees.
PV0

0005

ADD
Add
Source A F8:7
0.007349556<
Source B F8:0
237.9172<
Dest F8:0
237.9172<

0006

RET
Return

0007

END

Main

Processor Mode S:1/0 - S:1/4 = Remote Run
On Power up Go To Run (Mode Behavior) S:1/12 = 0
First Pass S:1/15 = No
Free Running Clock S:4 = 1110-1111-0100-0011

Proc

OS Catalog Number S:57 = 1510
OS Series S:58 = C
OS FRS S:59 =
Processor Catalog Number S:60 =
Processor Series S:61 = A
Processor FRN S:62 =
User Program Type S:63 = 818h
Compiler Revision Number S:64 =

Scan Times

Maximum (x10 ms) S:22 = 22
Watchdog (x10 ms) S:3 (high byte) = 10
Last 100 uSec Scan Time S:35 = 6
Scan Toggle Bit S:33/9 = 0

Math

Math Overflow Selected S:2/14 = 0
Overflow Trap S:5/0 = 0
Carry S:0/0 = 0
Overflow S:0/1 = 0
Zero Bit S:0/2 = 0
Sign Bit S:0/3 = 0
Math Register (lo word) S:13 = 0
Math Register (high word) S:14-S:13 = 0
Math Register (32 Bit) S:14-S:13 = 0

Chan 0

Processor Mode S:1/0- S:1/4 = Remote Run
Node Address S:15 (low byte) = 0
Baud Rate S:15 (high byte) = ?
Channel Mode S:33/3 = 0
Comms Active S:33/4 = 0
Incoming Cmd Pending S:33/0 = 0
Msg Reply Pending S:33/1 = 0
Outgoing Msg Cmd Pending S:33/2 = 0

Debug

Suspend Code S:7 = 0
Suspend File S:8 = 0

Errors

Fault Override At Power Up S:1/8 = 0
Startup Protection Fault S:1/9 = 0
Major Error Halt S:1/13 = 0
Overflow Trap S:5/0 = 0
Control Register Error S:5/2 = 0
Major Error Executing User
Fault Rtn. S:5/3 = 0
Battery Low S:5/11 = 0
Input Filter Selection Modified S:5/13 = 0
ASCII String Manipulation error S:5/15 = 0
Fault Routine S:29 = 0
Major Error S:6 = 0h
Error Description:

Protection

Deny Future Access S:1/14 = No
Data File Overwrite Protection Lost S:36/10 = False

Mem Module

Memory Module Loaded On Boot S:5/8 = 0
Password Mismatch S:5/9 = 0
Load Memory Module On Memory Error S:1/10 = 0
Load Memory Module Always S:1/11 = 0
On Power up Go To Run (Mode Behavior) S:1/12 = 0
Program Compare S:2/9 = 0
Data File Overwrite Protection Lost S:36/10 = 0

Forces

Forces Enabled S:1/5 = Yes
Forces Installed S:1/6 = No

Data File B3 (bin) -- BINARY

Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	(Symbol) Description
B3:0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	

Data File N7 (dec) -- INTEGER

Offset	0	1	2	3	4	5	6	7	8	9
N7:0	2500	2383	16383	15837						

Data File F8 -- SIM DATA -- Simulator Data and Constants

Offset	0	1	2	3	4
F8:0	237.9172	237.5963	237.2894	1.979106	-0.979202
F8:5	0.0001811589	0.000178941	0.007349556	99.99364	99.99364
F8:10	99.99364	99.99364	99.99364	99.99364	99.99364
F8:15	99.99364	99.99364	99.99364	99.99364	99.99364
F8:20	99.99364	99.99364	99.99364	99.99364	99.99364
F8:25	99.99364	99.99364	99.99364	99.99364	99.99364
F8:30	99.99364	99.99364			

Data File PD9 -- PID CNTRL

Offset	TM	AM	CM	OL	RG	SC	TF	DA	DB	UL	LL	SP	PV	DN	EN	SPS	KC	Ti	TD	MAXS	MINS	ZCD	CVH	CVL	LUT
PD9:0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	2500	32767	0	0	16384	0	0	100	0	100

Address (Symbol) = Value [Description]

PD9:0.SPS = 2500 [Scaled Set Point inX10 integer format.]
PD9:0.SPV = 2383 [Set point]
PD9:0.CVP = 100 [Control Output]
PD9:0.KC = 32767 [Gain, unitless.]
PD9:0.Ti = 0 [Integrator TimeConstant in minutes]
PD9:0.TD = 0 [Differentiator TimeConstant in minutes.]

Address (Symbol) = Value [Description]

PD9:0.SPS = 2500 [Scaled Set Point inX10 integer format.]
PD9:0.SPV = 2383 [Set point]
F8:3 (A1) = 1.979106 [Output coeficientfor n-1]
PD9:0.KC = 32767 [Gain, unitless.]
PD9:0.Ti = 0 [Integrator TimeConstant in minutes]
PD9:0.TD = 0 [Differentiator TimeConstant in minutes.]