Checks to see if last current value sent is greater or less than the new value being sent.
If the last value sent is greater than the new value (current will be decreasing) then send target current to PSU.
If the last value is less than the new value (current will be increasing) then toggle boolean bit to allow for the hysteresis value.

0

Regulation_Send

GRT
Greater Than (A>B)
Source A I_coarse 0.0
Source B Last_I_Sent 0.0

cont3

LES
Less Than (A<B)
Source A I_coarse 0.0
Source B Last_I_Sent 0.0

MUL
Multiply
Source A Q1_I_MOL 0.0
Source B 799.9992
Dest Set_current_value_Percent_MOL 0

DTOS
DINT to String
Source Set_current_value_Percent_MOL 0
Dest Current_Output '000000'

reg_cont

Stop_Regulation_q1

DTOS
DINT to String
Source MPS_Wa_Value 0
Dest Current_Output '000000'

hold_current

This command is used to write a value between 0 to 999999 to the regulation module. Leading Zero format must be used.
Six numbers are to be used.

1

reg_cont

hold_current

LEQ
Less Than or Eql (A<=B)
Source A Current_Output.LEN 6
Source B 6

MOV
Move
Source Current_Output.LEN 6
Dest Current_length_hold 1

SUB
Subtract
Source A 7
Source B Current_Output.LEN 6
Dest start_position 6

INSERT
Insert String
Source A zero_6x '000000'
Source B Current_Output '000000'
Start start_position 6
Dest Current_Output '000000'

cont2

Add the value to the command Wa, then add carriage return and send it out.
Copying to the send string clears any previous stored command.
HMI commands go here

2

cont2

DELETE
String Delete
Source Current_Output '000000'
Qty Current_length_hold 1
Start 7
Dest Current_Output '000000'

CONCAT
String Concatenate
Source A MPS_Wa_cmd 'WA '
Source B Current_Output '000000'
Dest MPS_Wa_cmd_string 'WA 000000$r'

CONCAT
String Concatenate
Source A MPS_Wa_cmd_string 'WA 000000$r'
Source B CR '$r'
Dest MPS_Wa_cmd_string 'WA 000000$r'

COP
Copy File
Source MPS_Wa_cmd_string
Dest SString[1]
Length 1

No_Reply3
ONS

MOV
Move
Source 0
Dest count 0

3

Check ADC readback to see if it is equal to the PSU hysteresis value

Reg_On_constant

MUL
Multiply
Source A Q1_I_MOL 0.0
Source B 0.02
Dest MPS_Wa_value_hyst_prec 0.0

ADD
Add
Source A Q1_I_MOL 0.0
Source B MPS_Wa_value_hyst_prec 0.0
Dest MPS_Wa_value_hyst_upper 0.0

SUB
Subtract
Source A Q1_I_MOL 0.0
Source B MPS_Wa_value_hyst_prec 0.0
Dest MPS_Wa_value_hyst_lower 0.0

LIM
Limit Test (CIRC)
Low Limit MPS_Wa_value_hyst_lower 0.0
Test I_coarse 0.0
High Limit MPS_Wa_value_hyst_upper 0.0

Enable_cont3 ONS — cont3

4

Set PSU value equal to previously requested current value

cont3

DTOS
DINT to String
Source MPS_Wa_Value 0
Dest Current_Output '000000'

LEQ
Less Than or Eql (A<=B)
Source A Current_Output.LEN 6
Source B 6

MOV
Move
Source Current_Output.LEN 6
Dest Current_length_hold 1

SUB
Subtract
Source A 7
Source B Current_Output.LEN 6
Dest start_position 6

INSERT
Insert String
Source A zero_6x '000000'
Source B Current_Output '000000'
Start start_position 6
Dest Current_Output '000000'

cont4

5

Add the value to the command Wa, then add carriage return and send it out.
Copying to the send string clears any previous stored command.

cont4

DELETE
String Delete
Source Current_Output '000000'
Qty Current_length_hold 1
Start 7
Dest Current_Output '000000'

CONCAT
String Concatenate
Source A MPS_Wa_cmd 'WA '
Source B Current_Output '000000'
Dest MPS_Wa_cmd_string 'WA 000000$r'

CONCAT
String Concatenate
Source A MPS_Wa_cmd_string 'WA 000000$r'
Source B CR '$r'
Dest MPS_Wa_cmd_string 'WA 000000$r'

HMI commands go here

COP
Copy File
Source MPS_Wa_cmd_string
Dest SString[1]
Length 1

No_Reply_2 ONS — Loop L

MOV
Move
Source 0
Dest count 0

6

"Loop" indicator is latched during ramp-down

Loop

MUL
Multiply
Source A Q1_I_Target 0.0
Source B 0.02
Dest MPS_Wa_value_hyst_prec 0.0

ADD
Add
Source A Q1_I_Target 0.0
Source B MPS_Wa_value_hyst_prec 0.0
Dest MPS_Wa_value_hyst_upper 0.0

SUB
Subtract
Source A Q1_I_Target 0.0
Source B MPS_Wa_value_hyst_prec 0.0
Dest MPS_Wa_value_hyst_lower 0.0

LIM
Limit Test (CIRC)
Low Limit MPS_Wa_value_hyst_lower 0.0
Test I_coarse 0.0
High Limit MPS_Wa_value_hyst_upper 0.0

Loop U

(End)