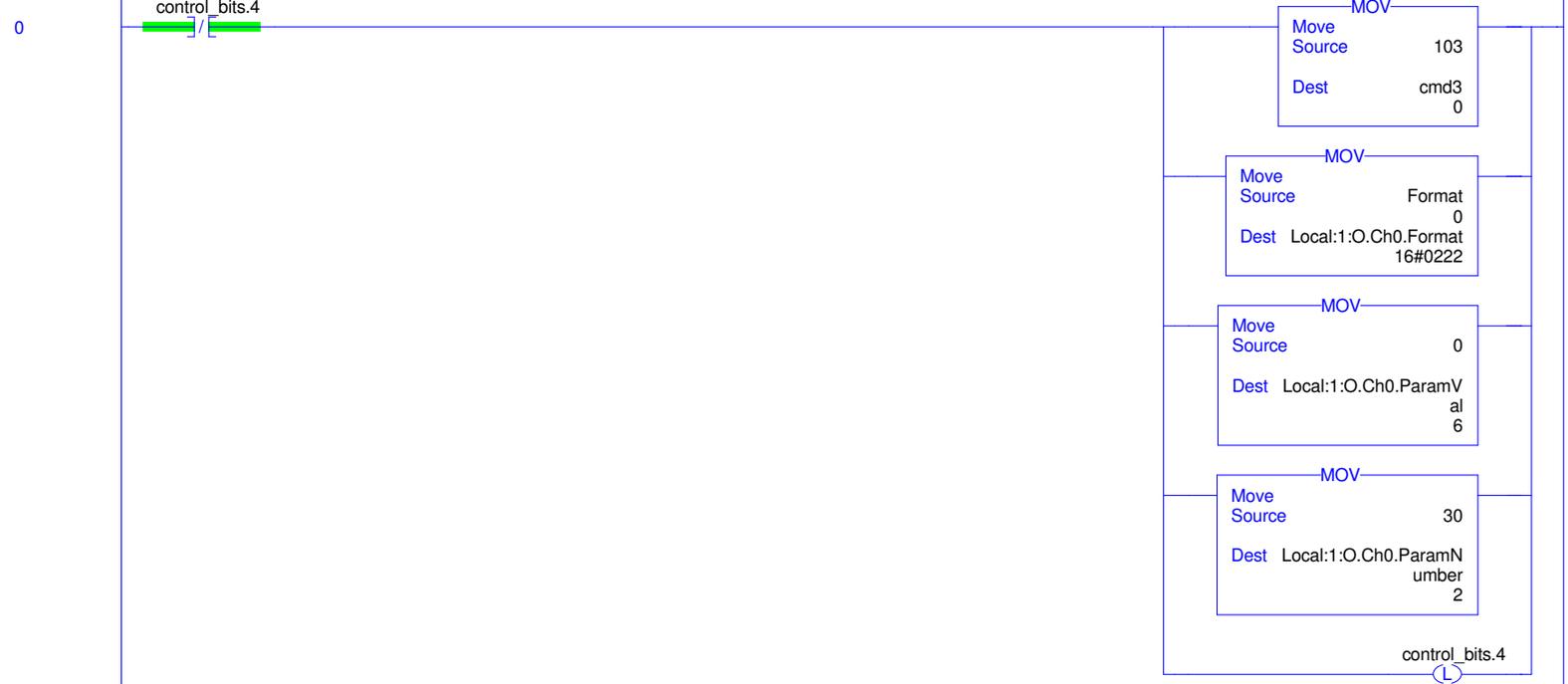
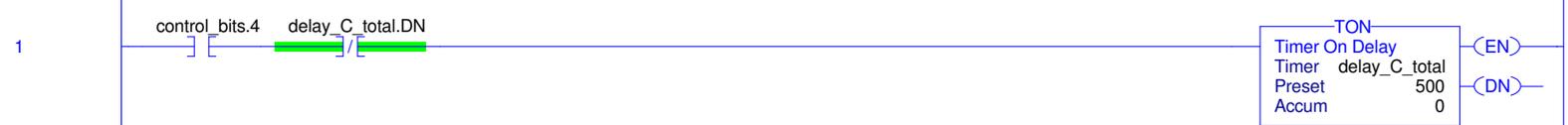


Routine to clear Total

When the subroutine is triggered, this rung will move the write command, 103, to a cmd3 tag. It also moves the format value, xyz hex, Total parameter ID, 30, and the new total value, 0, into the Local:x:O table. Once triggered, the rung is blocked to prevent running again.



The ladder logic seems to run very fast, so once the routine is triggered and we start moving data around, I placed a delay here to prevent commands being sent before all the data movement is done. I set for 1/2 second, but do not think it needs to be this long a delay. Just needs to be long enough for all data to be moved/copied before proceeding.



Once the delay time has elapsed, will latch a bit on to start next stage of command.



This rung will trigger the subroutine that actually runs the command and sends it to the module. This subroutine needs a command number as an input parameter. The command number in cmd3 tag will be given as the input parameter.



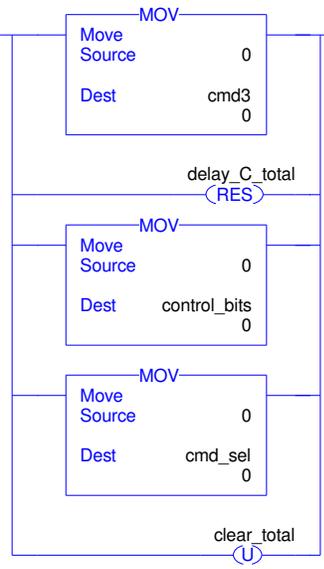
Subroutine called to send commands and values to the module.

Routine to clear Total

The subroutine will turn on a bit indicating when it is done. Once that bit, cmd_sel.15, is set, this rung will then clear tags, reset delay timers, and clear bits that triggered the routine.

4

cmd_sel.15

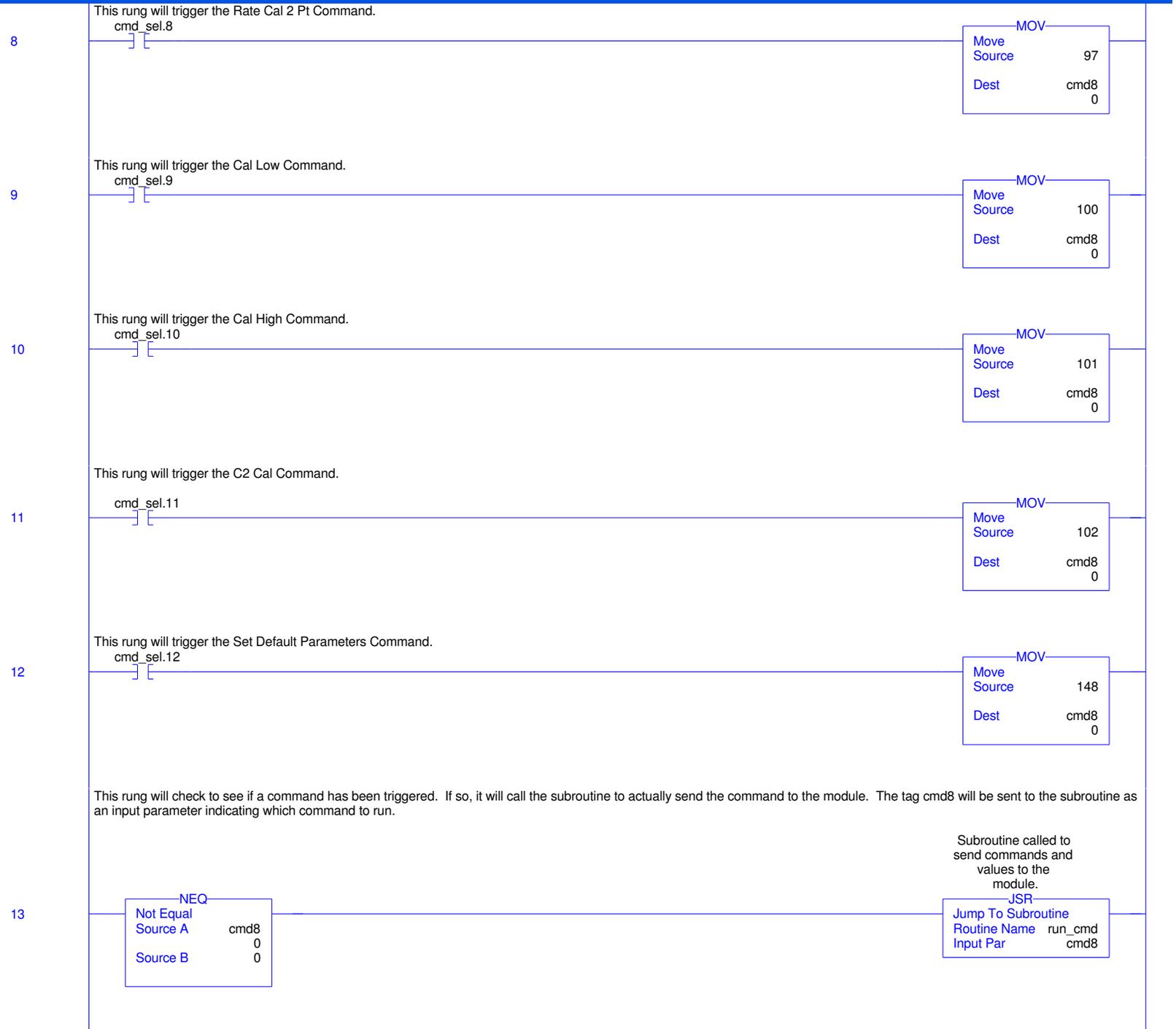


(End)

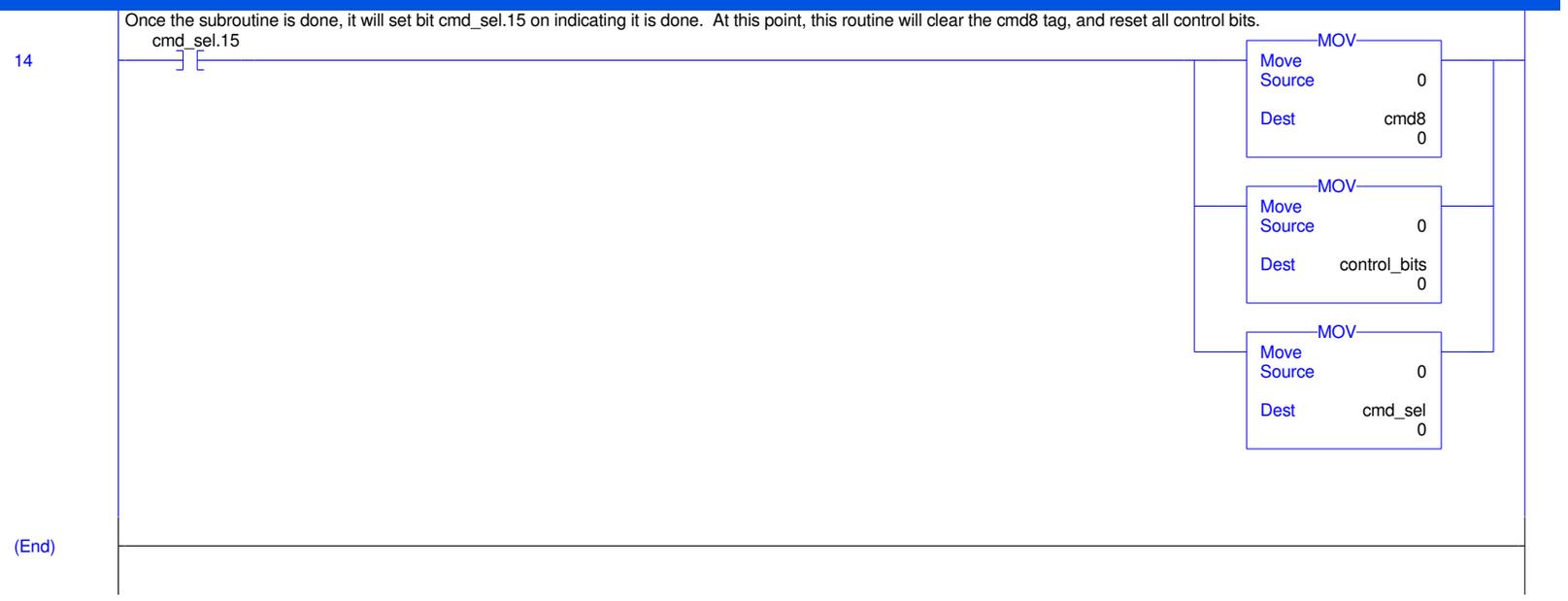
This routine will allow the user to trigger the commands to be sent to the module. Each of the first rungs, 0 – 12, will trigger a different command. Each rung will move a command number into tag cmd to run the command.



This routine will allow the user to trigger the commands to be sent to the module. Each of the first rungs, 0 – 12, will trigger a different command. Each rung will move a command number into tag cmd8 to run the command.



This routine will allow the user to trigger the commands to be sent to the module. Each of the first rungs, 0 – 12, will trigger a different command. Each rung will move a command number into tag cmd to run the command.





Vendor/ Address/ Phone

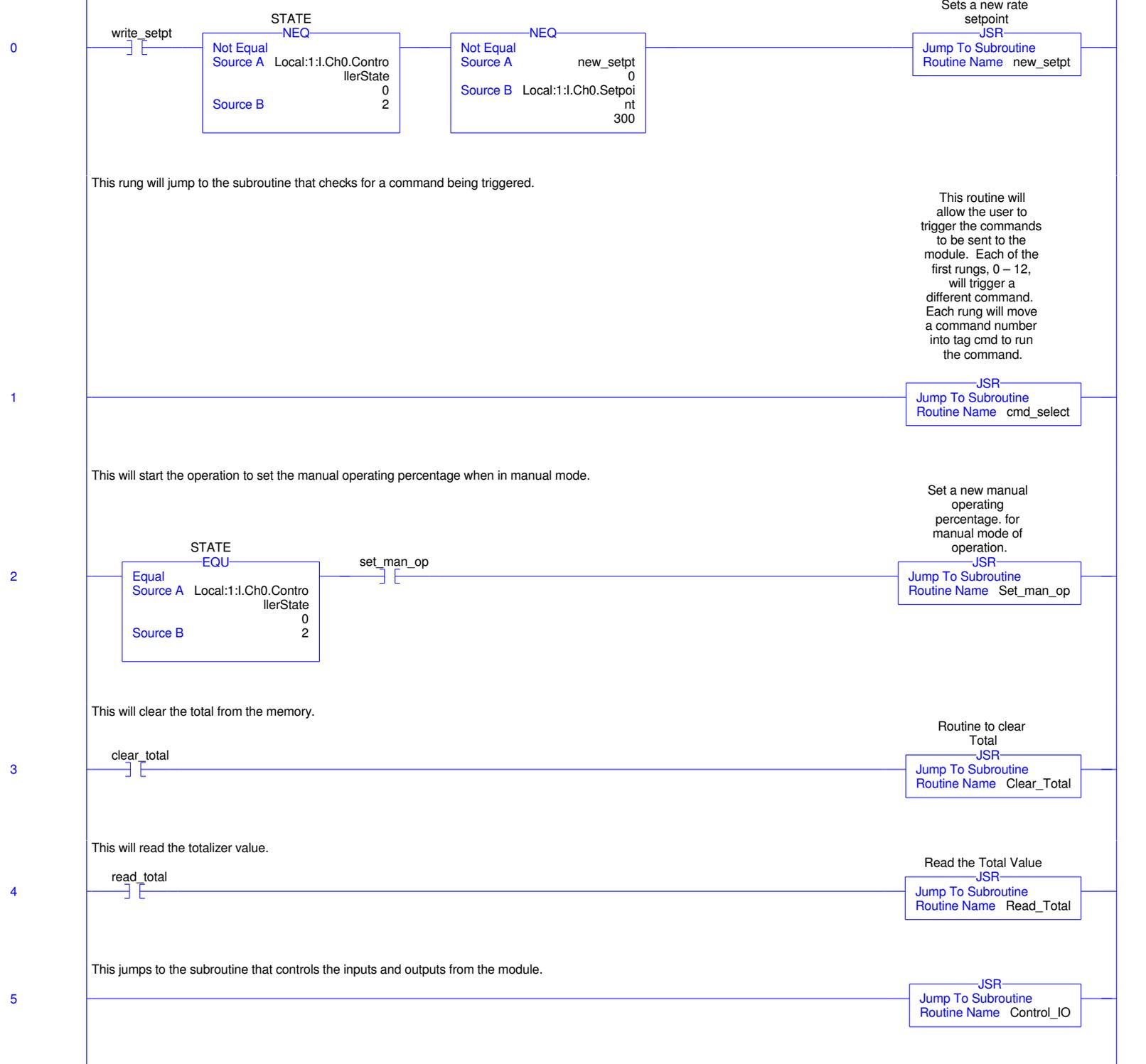
Hardy Instruments
 3860 Calle Fortunada, San Diego, CA , 92123-1825
 (800) 821-5831 USA

Sample is for demonstration purposes only, showing an example of reading and writing commands and values to and from the module.

Main routine with bits to trigger the start of all other routines.

The operator will enter a new setpoint into the new_setpt tag. If the new setpoint is not equal to the current setpoint, and the module is not in manual mode, then it will jump to the new_setpt subroutine. This subroutine will download the new setpoint value into the module. This routine can be trigger by the write_setpt bit or if you wish it to run automatically you can remove the write_setpt bit and it will run whenever the new_setpt does not equal the current setpt.

NOTE: If you have this run automatically and the setting of the new setpoint fails for some reason, it will run the routine again. If the reason the setting failed, this will continuously run.

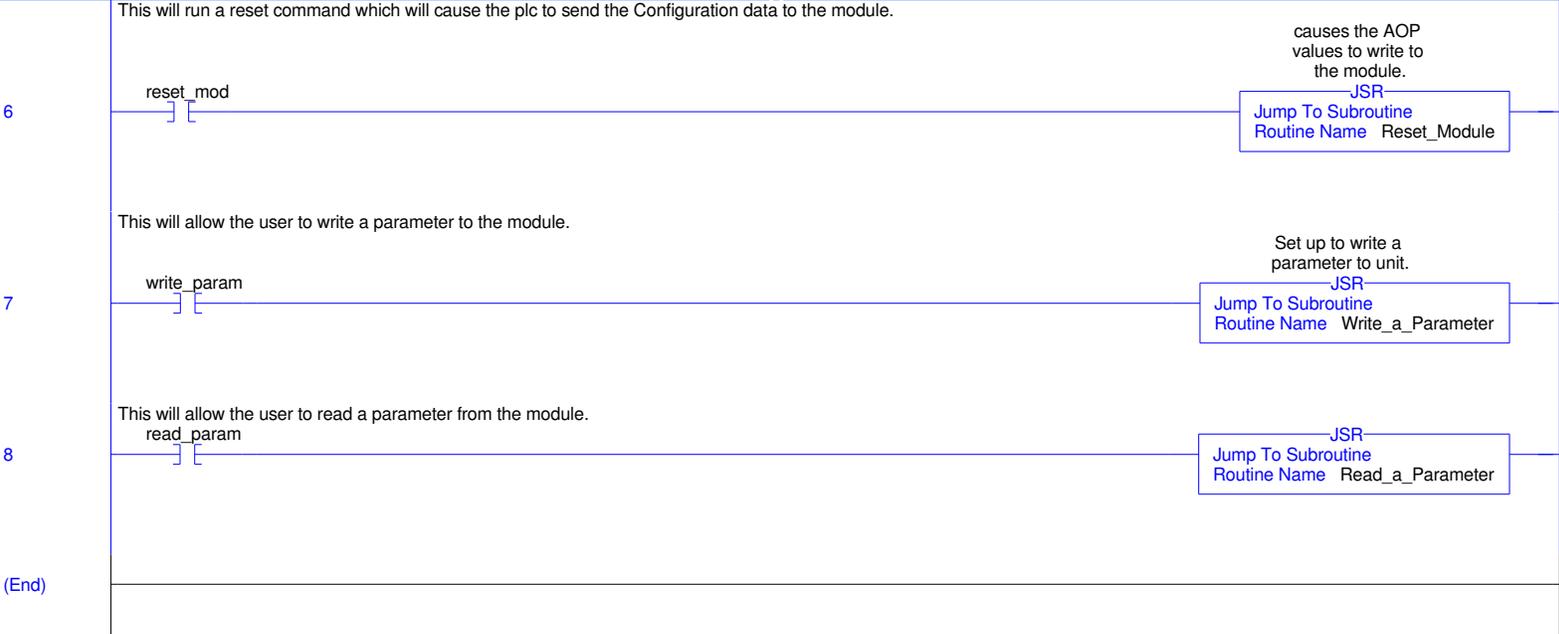


Vendor/ Address/ Phone

Hardy Instruments
3860 Calle Fortunada, San Diego, CA , 92123-1825
(800) 821-5831 USA

Sample is for demonstration purposes only, showing an example of reading and writing commands and values to and from the module.

Main routine with bits to trigger the start of all other routines.



Sets a new rate setpoint

When the subroutine is triggered, this rung will move the write command, 103, to a com2 tag. It also moves the format value, xyz hex, setpoint parameter ID, 35, and the new setpoint value into the Local:x:O table. Once triggered, the rung is blocked to prevent running again.



The ladder logic seems to run very fast, so once the routine is triggered and we start moving data around, I placed a delay here to prevent commands being sent before all the data movement is done. I set for 1/2 second, but do not think it needs to be this long a delay. Just needs to be long enough for all data to be moved/copied before proceeding.



Once the delay time has elapsed, will latch a bit on to start next stage of command.



This rung will trigger the subroutine that actually runs the command and sends it to the module. This subroutine needs a command number as an input parameter. The command number in cmd2 tag will be given as the input parameter.



Subroutine called to send commands and values to the module.

Sets a new rate setpoint

The subroutine will turn on a bit indicating when it is done. Once that bit, cmd_sel.15, is set, this rung will then clear tags, reset delay timers, and clear bits that triggered the routine. This will also copy the new setpoint value to the configuration table, so if the unit is reset, it will load the new value from the AOP file. If you do not wish to have the AOP file updated with the change, you would need to remove or disable the COP command that copies the value into the configuration table.

4

cmd_sel.15

MOV

Move	Source	0
Dest	cmd2	0

COP

Copy File	Source	Local:1:I.Ch0.ParamV
Dest	al	Local:1:C.Ch0RateSet
Length	Point	1

setpt_delay
(RES)

MOV

Move	Source	0
Dest	control_bits	0

MOV

Move	Source	0
Dest	cmd_sel	0

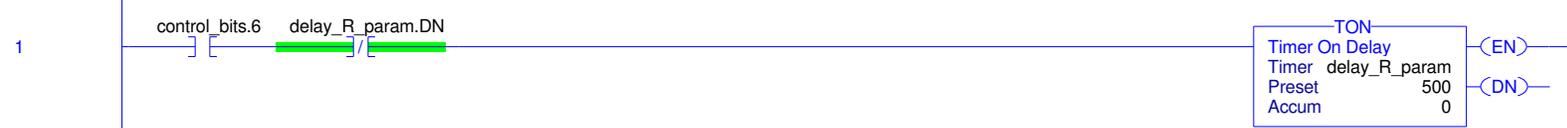
write_setpt
(U)

(End)

When the subroutine is triggered, this rung will move the write command, 105, to a cmd4 tag. It also moves the format value, xyz hex, parameter ID, found in tag param_ID, into the Local:x:O table. Once triggered, the rung is blocked to prevent running again. Prior to running this routine, you would insure the parameter ID number you wish to read is written into param_ID tag.



The ladder logic seems to run very fast, so once the routine is triggered and we start moving data around, I placed a delay here to prevent commands being sent before all the data movement is done. I set for 1/2 second, but do not think it needs to be this long a delay. Just needs to be long enough for all data to be moved/copied before proceeding.



Once the delay time has elapsed, will latch a bit on to start next stage of command.



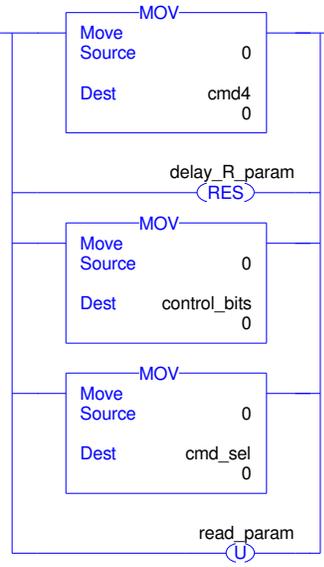
This rung will trigger the subroutine that actually runs the command and sends it to the module. This subroutine needs a command number as an input parameter. The command number in cmd4 tag will be given as the input parameter.



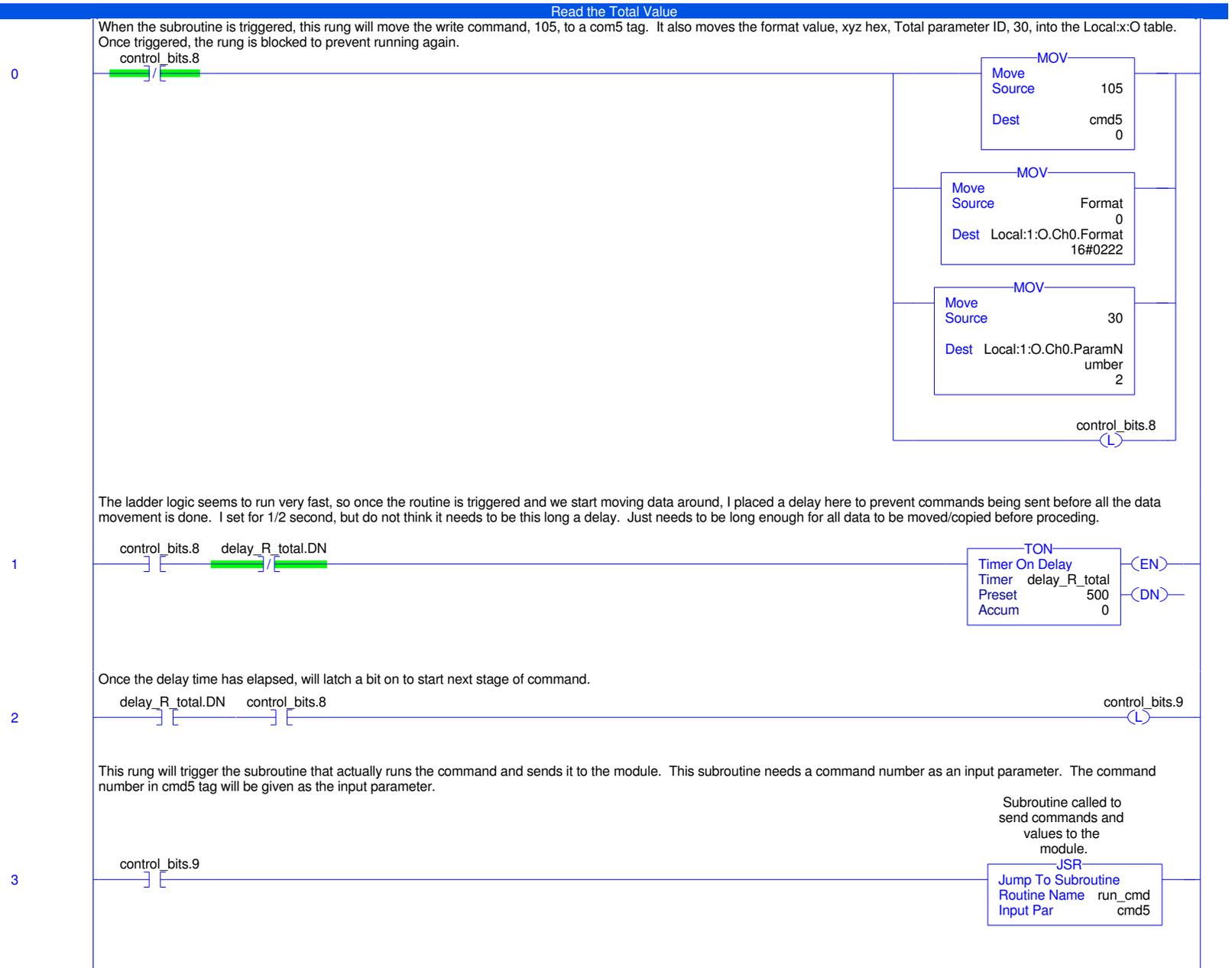
The subroutine will turn on a bit indicating when it is done. Once that bit, cmd_sel.15, is set, this rung will then clear tags, reset delay timers, copy status and clear bits that triggered the routine.

4

cmd_sel.15



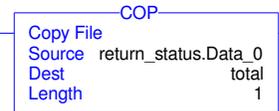
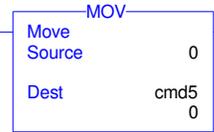
(End)



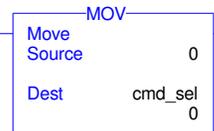
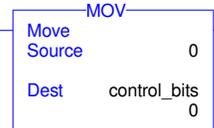
Read the Total Value

The subroutine will turn on a bit indicating when it is done. Once that bit, cmd_sel.15, is set, this rung will then clear tags, reset delay timers, copy status and clear bits that triggered the routine.

cmd_sel.15

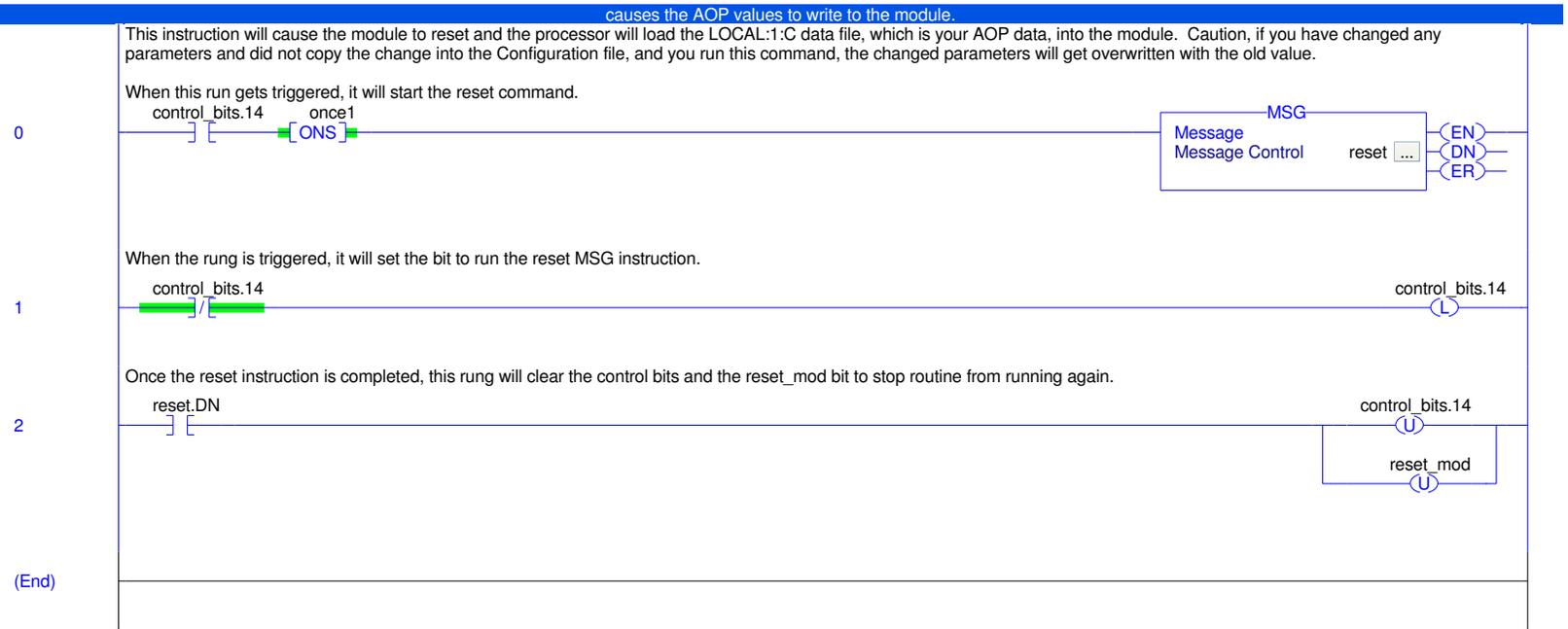


delay_R_total
(RES)



read_total
(U)

(End)



Subroutine called to send commands and values to the module.

This routine is receiving the command number as an input parameter.

This routine will get command numbers from other routines and actually send it to the module, gather return status, and let calling routines know when it is done.
 This rung will move the input command number to the Local:x:O data table to trigger the command.



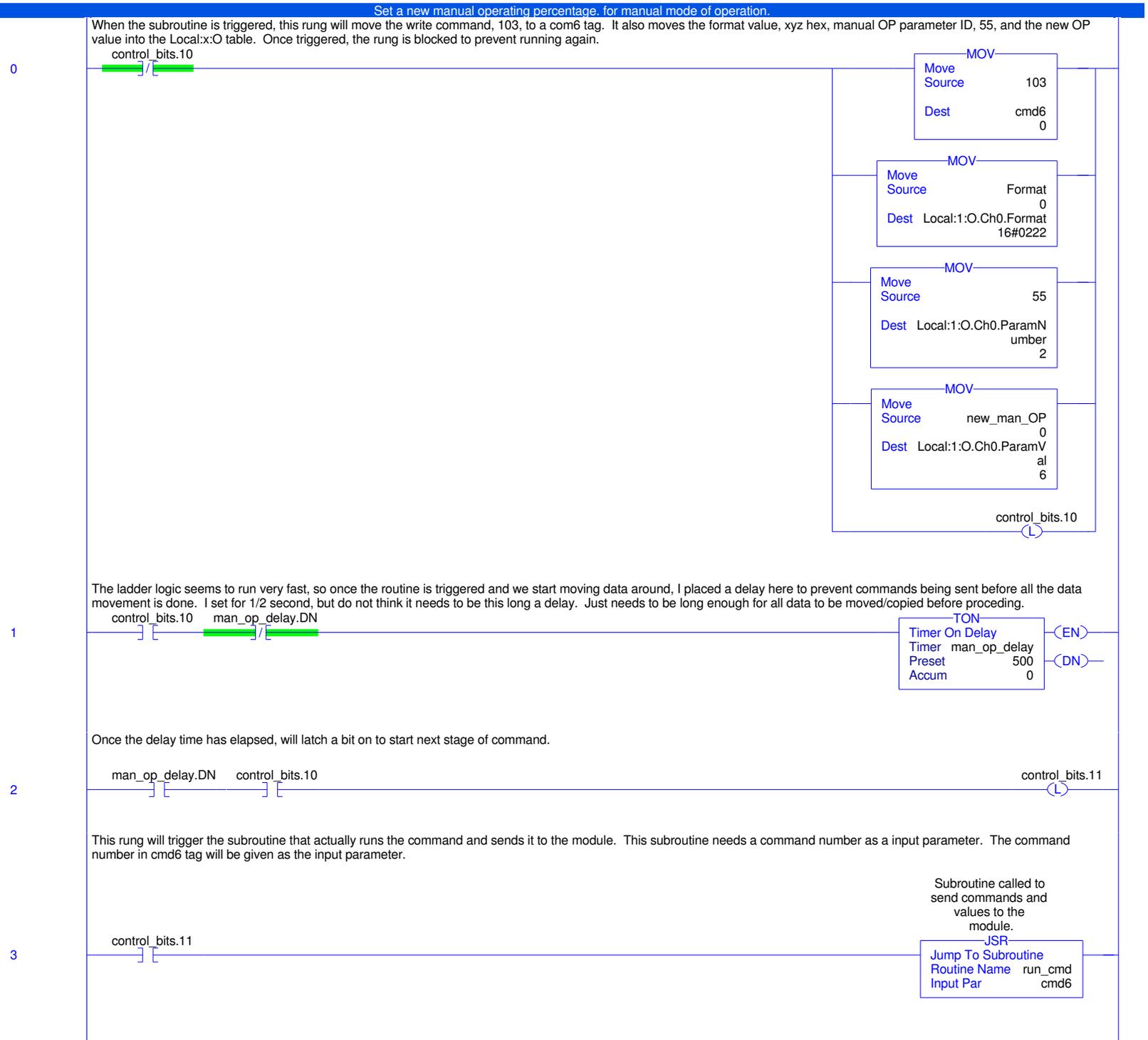
Once it has written the command number out to the module in the output table, it will monitor the input table Local:x:I for a reflection of the command number sent. This would indicate the command is completed. Once completed, the rung will copy the command status information into the "return_status" tag. This is a location where the user can refer to for status and/or error codes.



Once the command is complete, this rung will clear the output data location and set the bit indicating to the calling routine that it is done.



(End)

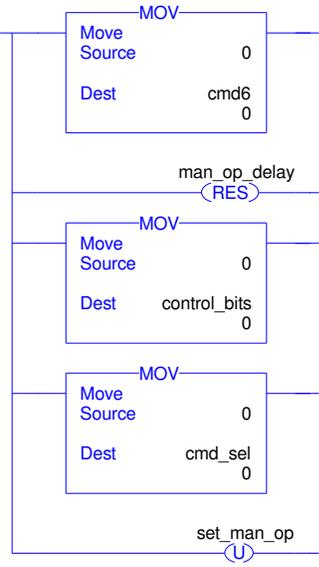


Set a new manual operating percentage. for manual mode of operation.

The subroutine will turn on a bit indicating when it is done. Once that bit, cmd_sel.15, is set, this rung will then clear tags, reset delay timers, and clear bits that triggered the routine.

4

cmd_sel.15



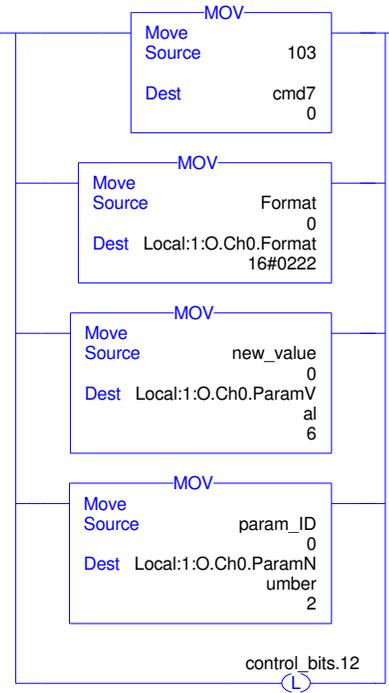
(End)

Set up to write a parameter to unit.

When the subroutine is triggered, this rung will move the write command, 103, to a com7 tag. It also moves the format value, xyz hex, parameter ID, in param_ID tag, and the new value, from new_value tag, into the Local:x:O table. Once triggered, the rung is blocked to prevent running again.

0

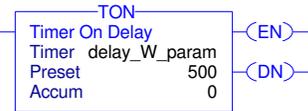
control_bits.12



The ladder logic seems to run very fast, so once the routine is triggered and we start moving data around, I placed a delay here to prevent commands being sent before all the data movement is done. I set for 1/2 second, but do not think it needs to be this long a delay. Just needs to be long enough for all data to be moved/copied before proceeding.

1

control_bits.12 delay_W_param.DN



Once the delay time has elapsed, will latch a bit on to start next stage of command.

2

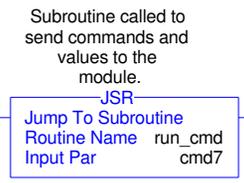
delay_W_param.DN control_bits.12

control_bits.13

This rung will trigger the subroutine that actually runs the command and sends it to the module. This subroutine needs a command number as an input parameter. The command number in cmd7 tag will be given as the input parameter.

3

control_bits.13

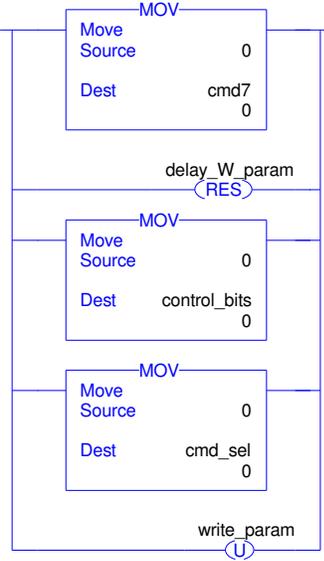


Set up to write a parameter to unit.

The subroutine will turn on a bit indicating when it is done. Once that bit, cmd_sel.15, is set, this rung will then clear tags, reset delay timers, copy status and clear bits that triggered the routine.

4

cmd_sel.15



(End)