

-  **Controller Winder**
 - Chapter 10 Winder Example Simplified PackML
-  **Controller Fault Handler**
-  **Power-Up Handler**

Tasks

-  **MainTask**
 -  **UN01_Winder**
 -  **MainRoutine**
 -  **UP00_Procedure**
 -  **UP01_PackML**
 -  **UP05_Laser_Distance**
 -  **UP05_Laser_Distance_Moving_Average**
 -  **EM_Axis01_Left**
 -  **MainRoutine**
 -  **CM00_Procedure**
 -  **CM02_ServoAxisObject**
 -  **CM03_ServoAxisJog**
 - Manual Jog Control jogs the servo axis when the Unit is in Idle state
 -  **EM_Axis02_Line**
 -  **MainRoutine**
 -  **CM00_Procedure**
 -  **CM02_ServoAxisObject**
 -  **CM03_ServoAxisJog**
 - Manual Jog Control jogs the servo axis when the Unit is in Idle state
 -  **EM_Axis03_Right**
 -  **MainRoutine**
 -  **CM00_Procedure**
 -  **CM02_ServoAxisObject**
 -  **CM03_ServoAxisJog**
 - Manual Jog Control jogs the servo axis when the Unit is in Idle state
-  **Unscheduled**

Motion Groups

-  **Motion_Group**
 -  **Axis_01_Left**
 -  **Axis_02_Line**
 -  **Axis_03_Right**
-  **Ungrouped Axes**

Add-On Instructions

Data Types

-  **User-Defined**
 -  **Unit_Procedure_Steps**
-  **Strings**
-  **Add-On-Defined**
-  **Module-Defined**
 -  **AB:1734_15SLOT:I:0**
 -  **AB:1734_15SLOT:O:0**
 -  **AB:1734_DI8:C:0**
 -  **AB:1734_DOB8:C:0**
 -  **AB:1734_IE8:C:0**
 -  **AB:1734_IE8:I:0**
 -  **AB:1734_OE4:C:0**
 -  **AB:1734_OE4:I:0**
 -  **AB:1734_OE4:O:0**
 -  **AB:1756_DI:C:0**
 -  **AB:1756_DI:I:0**
 -  **AB:1756_IF8_Float:C:0**
 -  **AB:1756_IF8_Float:I:0**
 -  **AB:1756_NII_Struct:C:0**
 -  **AB:1769_HSC1_Range:C:0**
 -  **AB:Embedded_AnalogIO1:C:0**

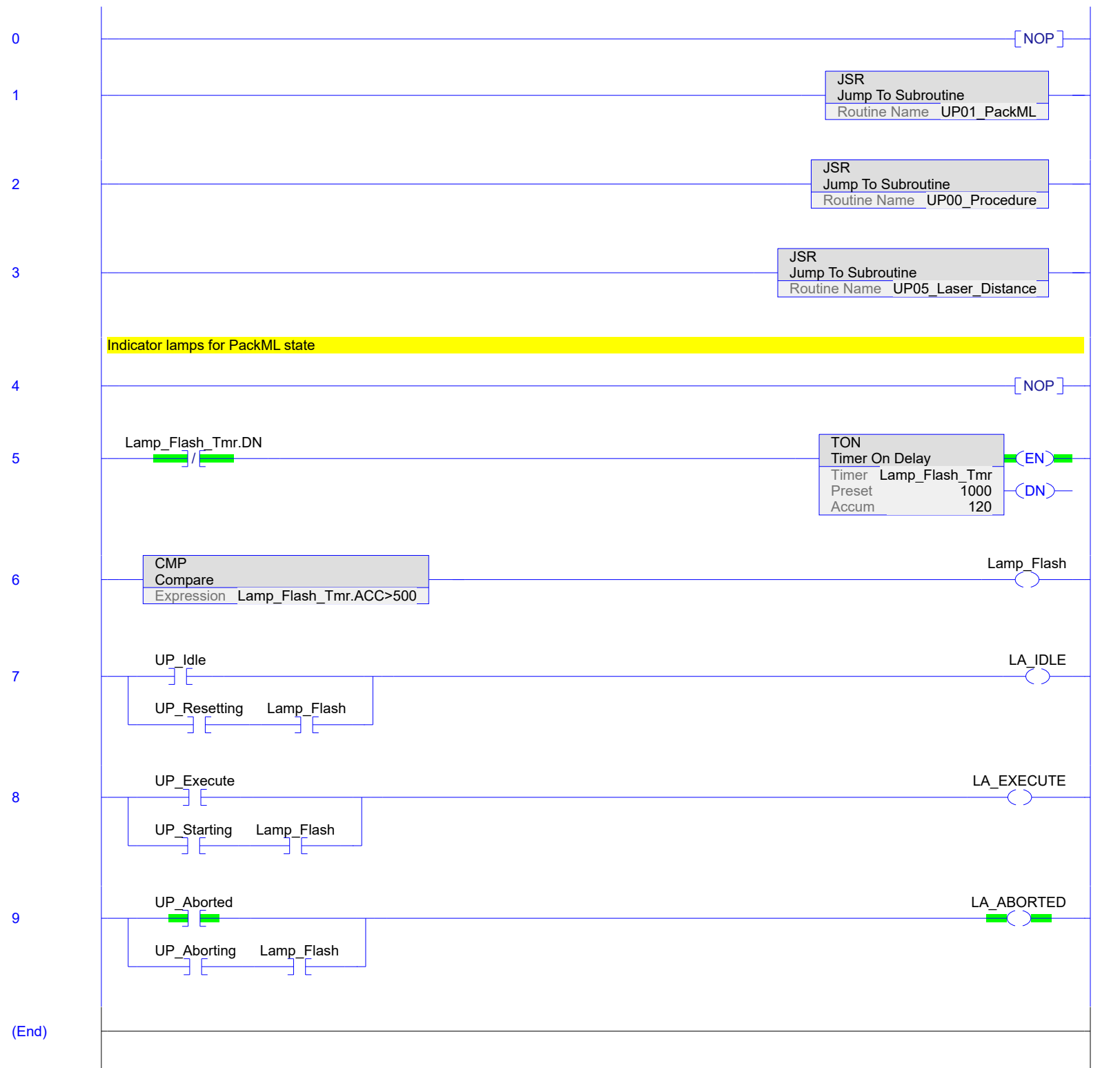
101 AB:Embedded_AnalogIO1:I:0
010
101 AB:Embedded_AnalogIO1:O:0
010
101 AB:Embedded_DiscreteIO1:C:0
010
101 AB:Embedded_DiscreteIO1:I:0
010
101 AB:Embedded_DiscreteIO1:O:0
010
101 AB:Embedded_HSC1:C:0
010
101 AB:Embedded_HSC1:I:0
010
101 AB:Embedded_HSC1:O:0
010
101 AB:Embedded_HSC1_STRUCT_OUT1:O:0
010
101 AB:Motion_Diagnostics:S:1
010

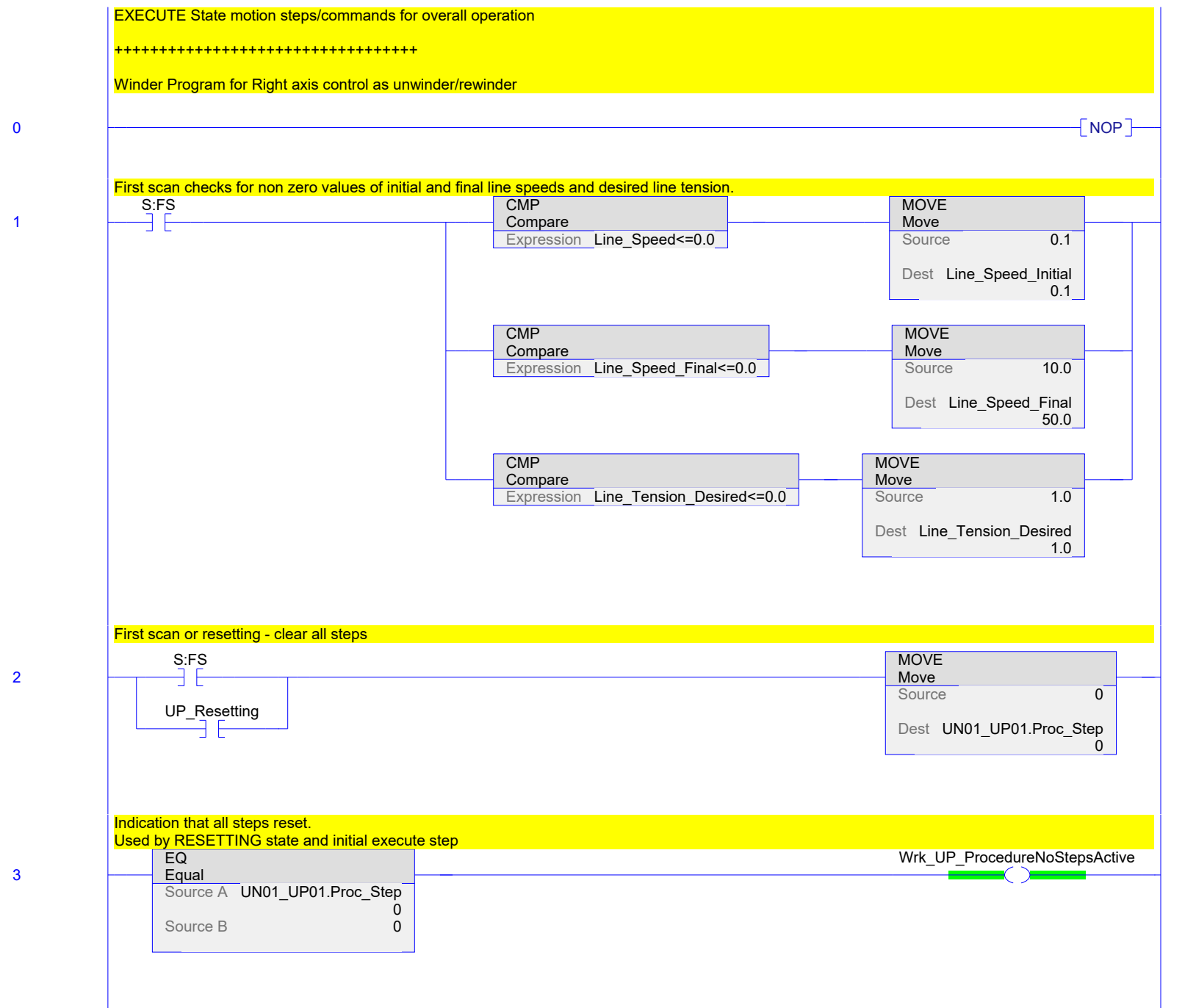
Trends

- ☒ Laser_Meas
Laser distance measurements
- ☒ Velocities

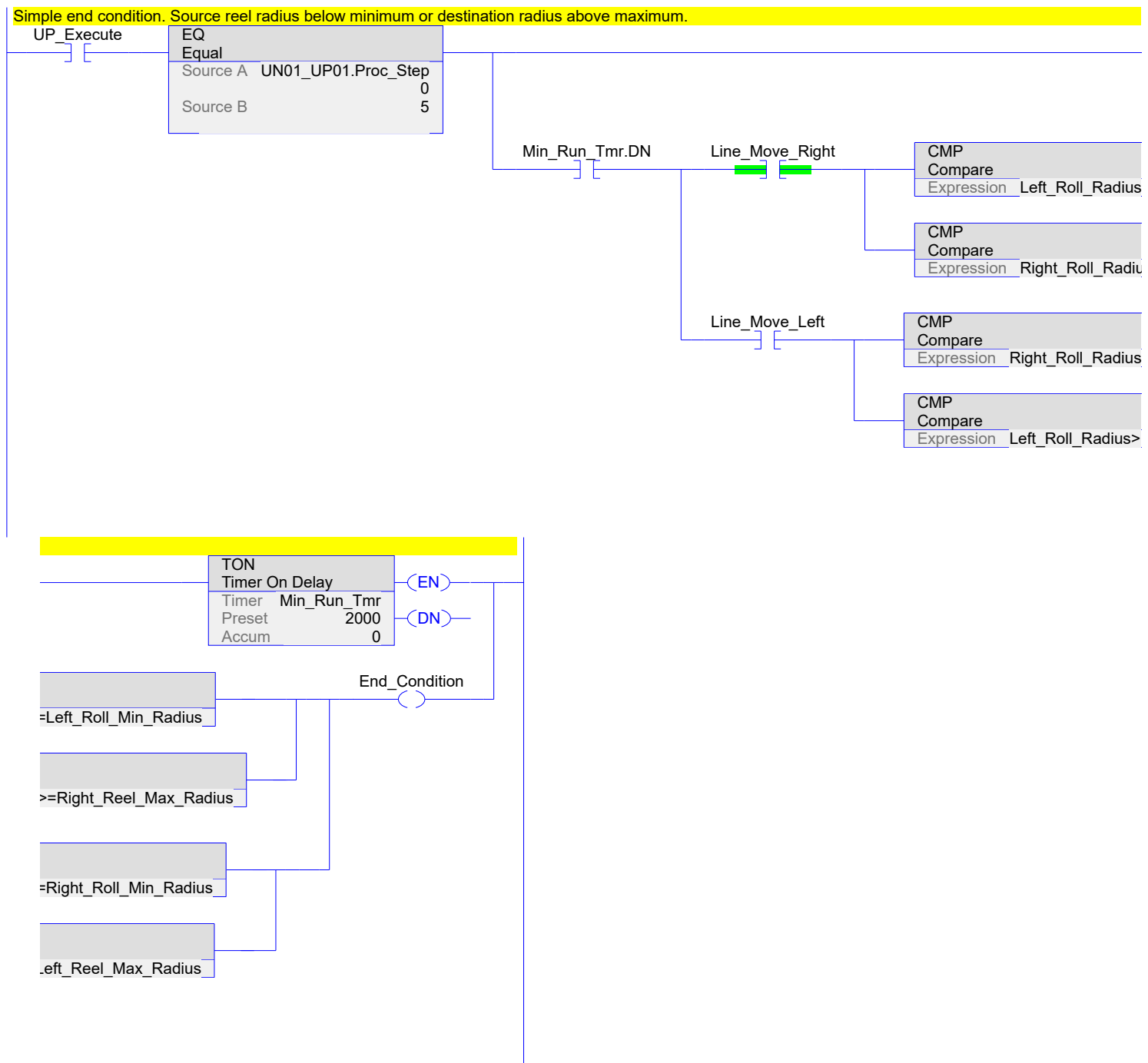
I/O Configuration

-  1756 Backplane, 1756-A10
 -  [0] 1756-L71 Winder
 -  [3] 1756-IF8/A analog_in
 -  [7] 1756-EN2TR en2tr
 -  Ethernet
 -  1756-EN2TR en2tr
 -  2198-C1004-ERS Left_Axis
 -  2198-C1004-ERS Right_Axis





4



EXECUTE -

If no steps active,
Line speed set to initial speed
Read direction switch and set/reset line move direction
Check that source reel is not below minimum radius
If checks pass, then start in step 1. Otherwise, to step 9 which goes back to idle.

UP_Execute Mode_Produce Wrk_UP_ProcedureNoStepsActive

MOVE
Move
Source Line_Speed_Initial
0.1
Dest Line_Speed
0.0

Direction switch.
0=move right (CW),
1= move left (CCW)

Directon_SW

Line_Move_Right

Line_Move_Left

Direction switch.
0=move right (CW),
1= move left (CCW)

Directon_SW

Line_Move_Right

Line_Move_Left

MOVE
Move
Source 1
Dest UN01_UP01.Proc_Step
0

Step 1 - Start rewinder axis

Line moving right, start the right axis. Moving left, start the left axis

UP_Execute

EQ
Equal
Source A UN01_UP01.Proc_Step
0
Source B 1

Line_Move_Right UN01_UP01.Proc_StepDn[1].[Axis_Right]

Line_Move_Left UN01_UP01.Proc_StepDn[1].[Axis_Left]

MOVE
Move
Source 2
Dest UN01_UP01.Proc_Step
0

Step 2 - Start line axis

Axis does not exist, so short the step complete indication from the axis

UP_Execute

EQ
Equal
Source A UN01_UP01.Proc_Step
0
Source B 2

UN01_UP01.Proc_StepDn[2].[Axis_Line]

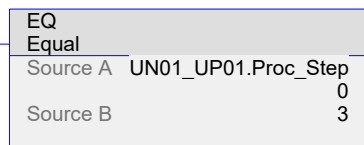
MOVE
Move
Source 3
Dest UN01_UP01.Proc_Step
0

Step 3 - Start unwinder axis

Line moving right, start the left axis. Moving left, start the right axis

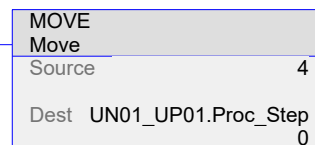
8

UP_Execute



Line_Move_Right UN01_UP01.Proc_StepDn[3].[Axis_Left]

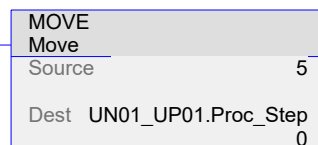
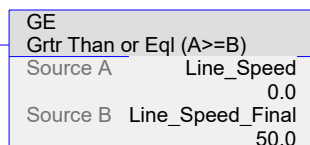
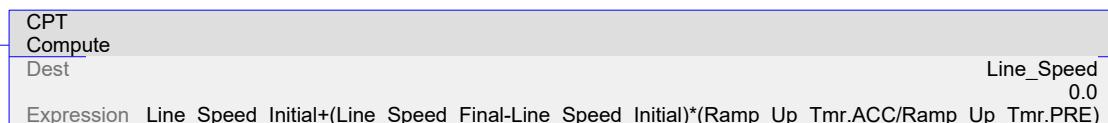
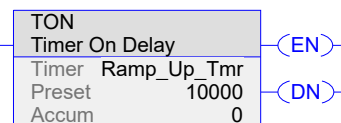
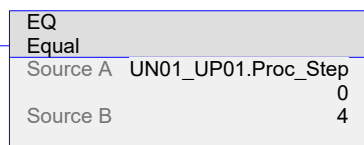
Line_Move_Left UN01_UP01.Proc_StepDn[3].[Axis_Right]



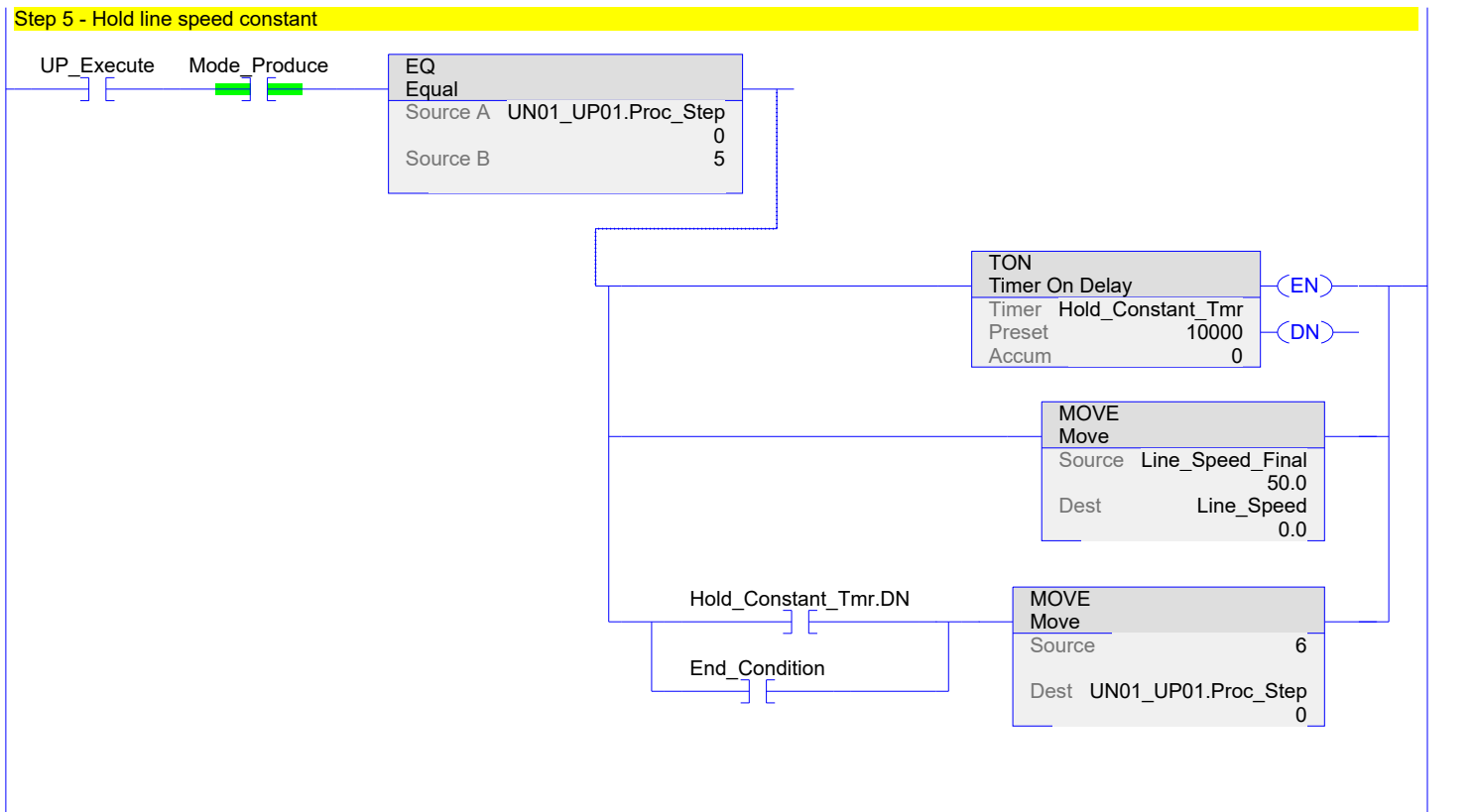
Step 4 - Ramp up to Final line speed

9

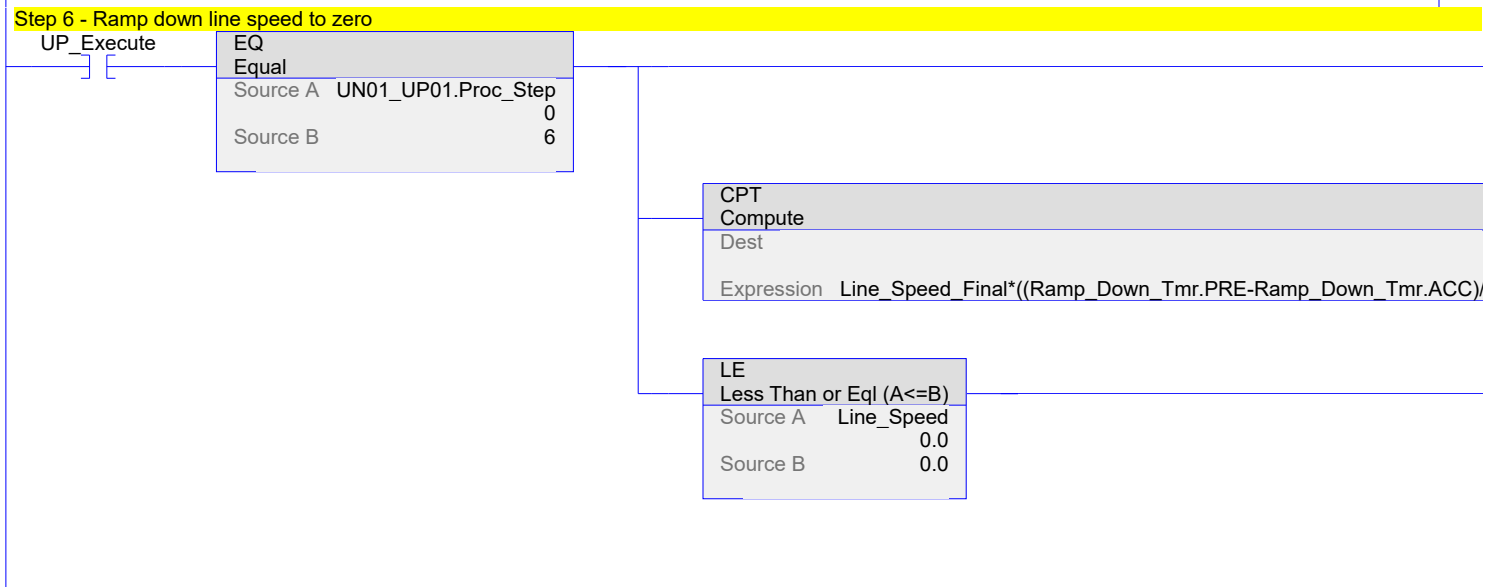
UP_Execute

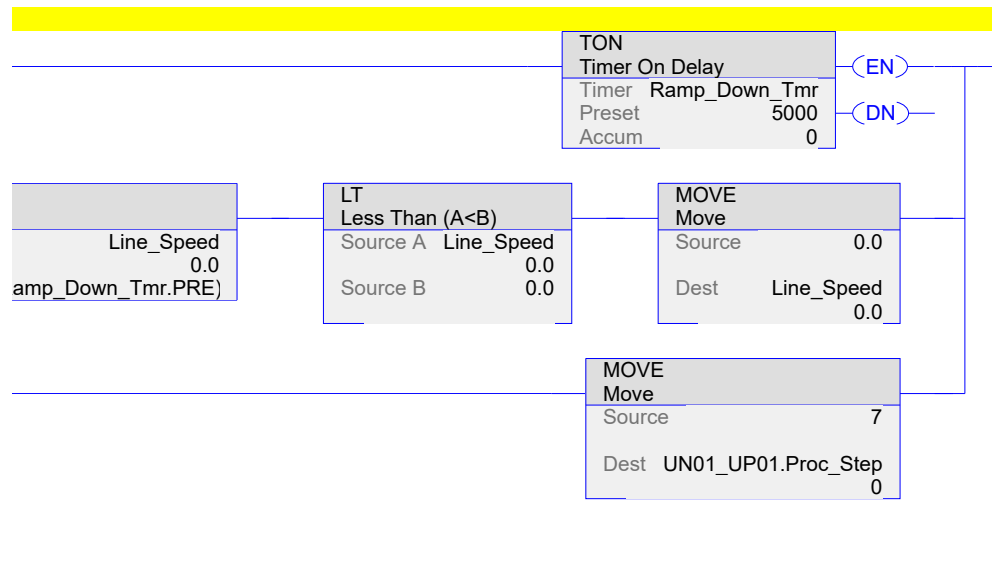


10



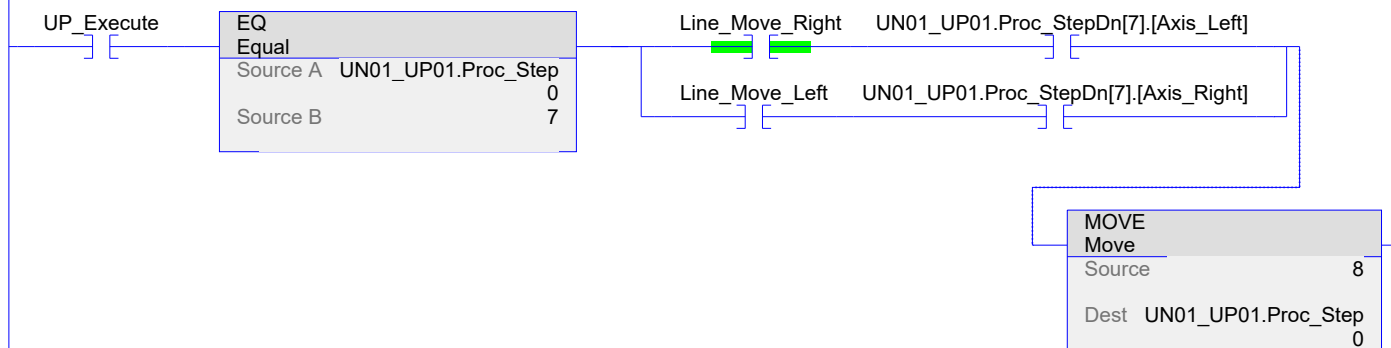
11





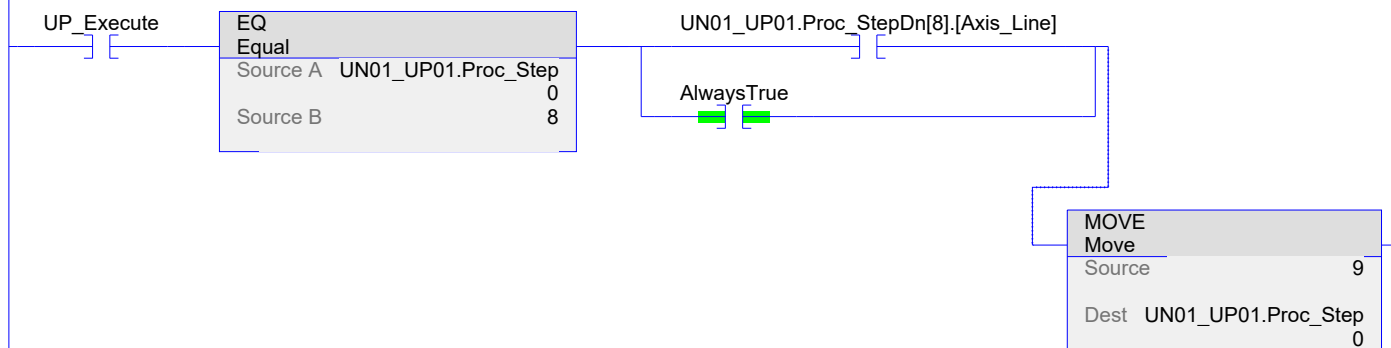
Step 7 - Stop unwinder axis
Line moving right, stop the left axis. Moving left, stop the right axis

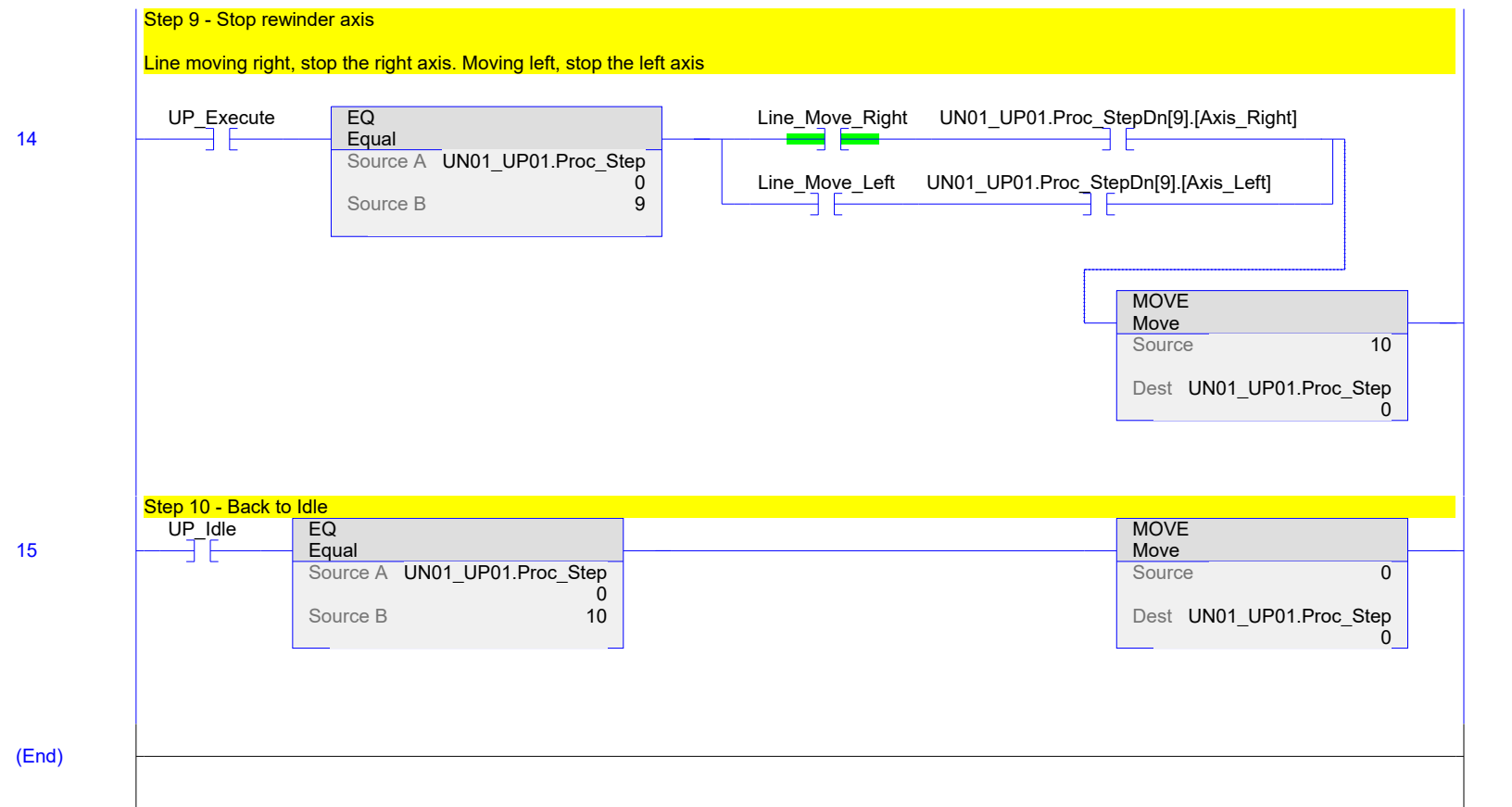
12

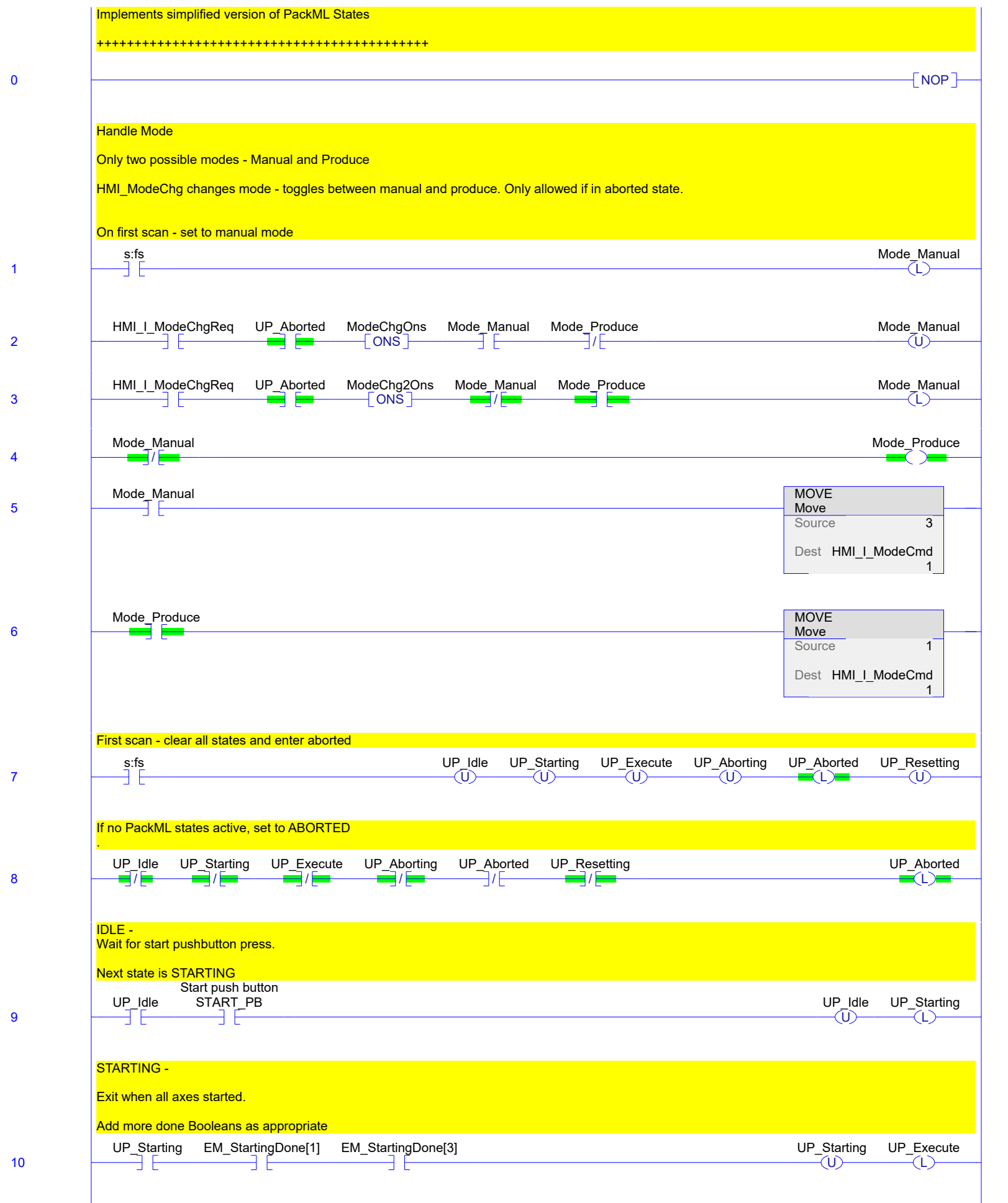


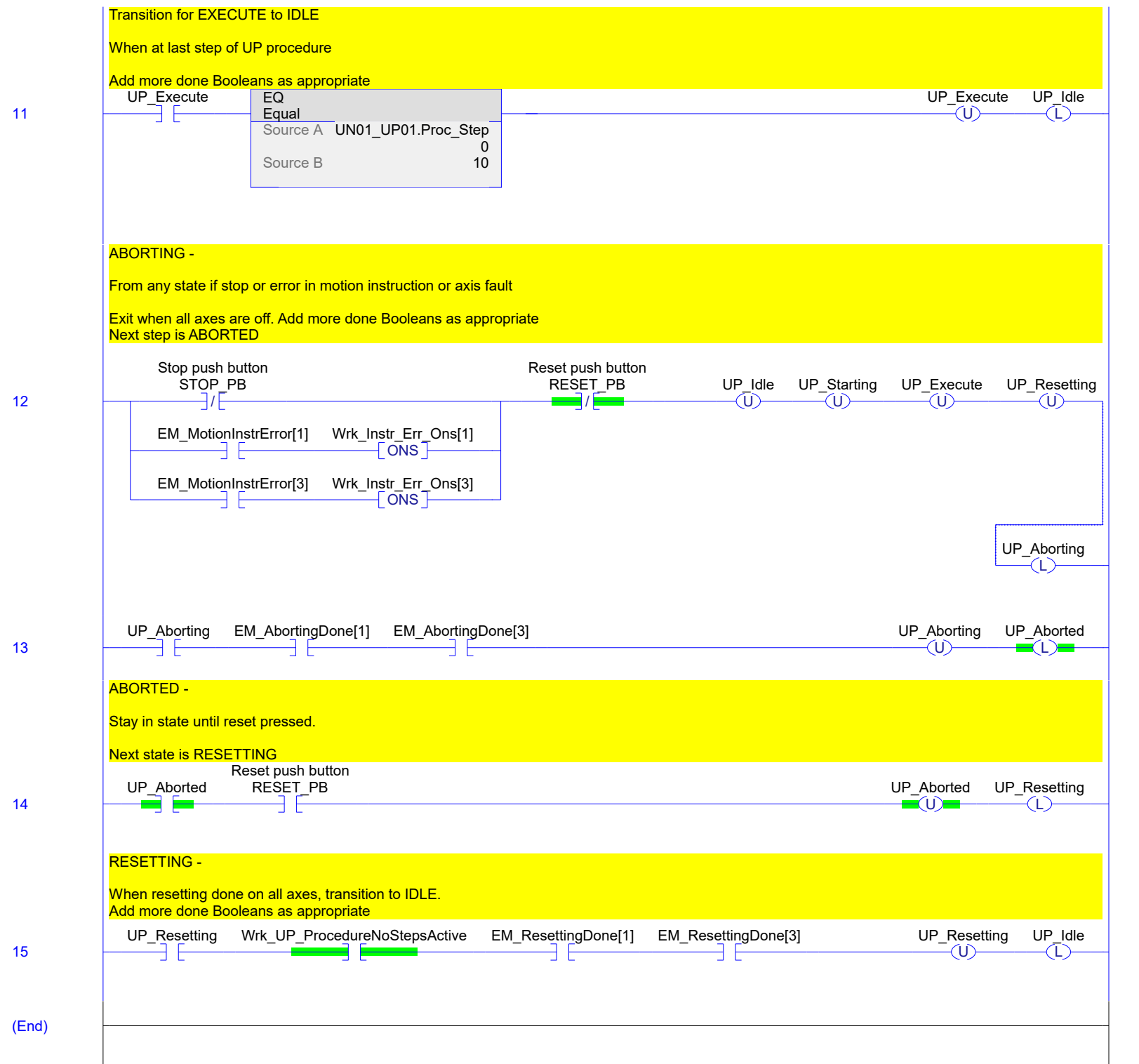
Step 8 - Stop Line axis
Axis does not exist, so short the step complete indication from the axis

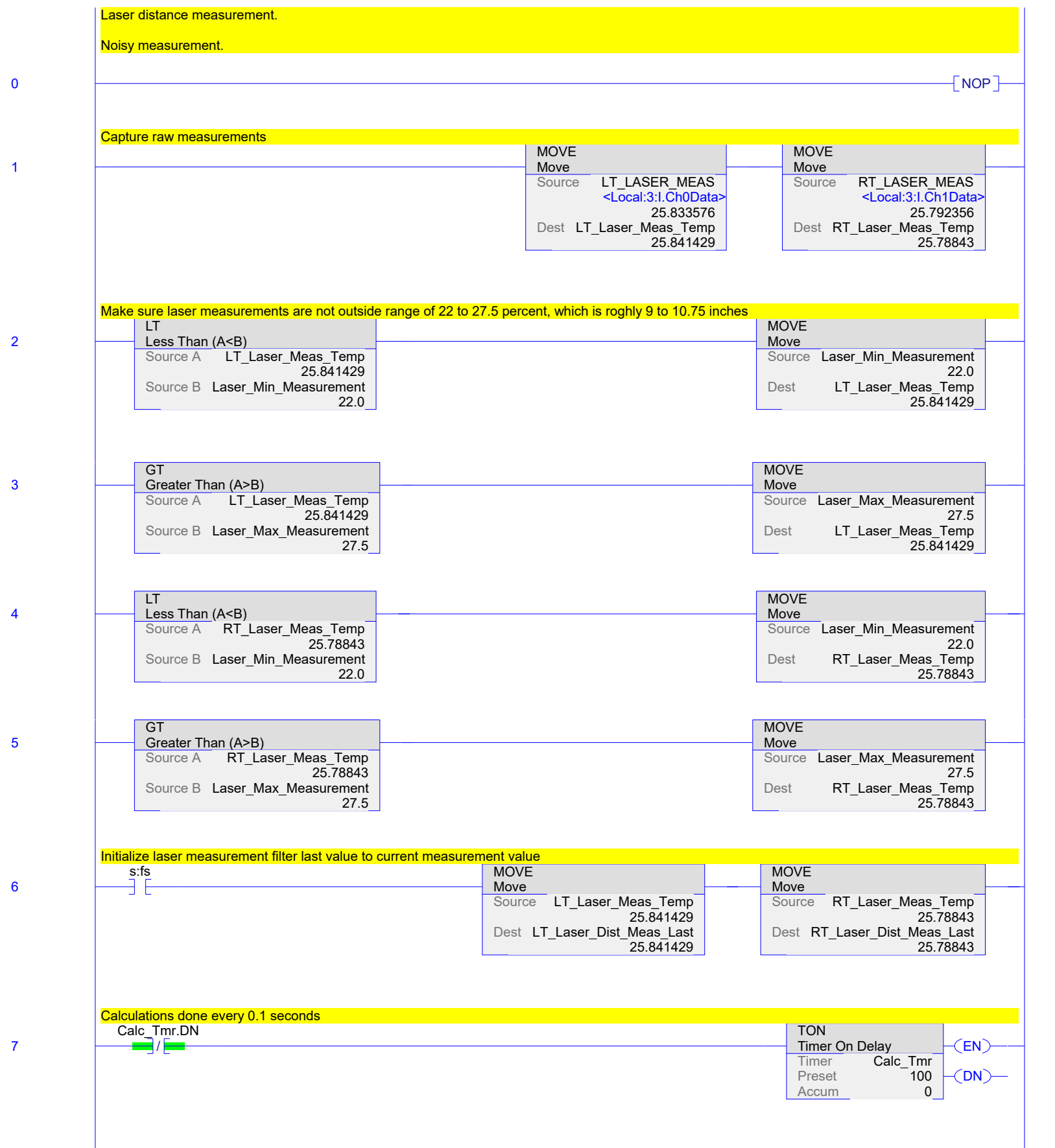
13

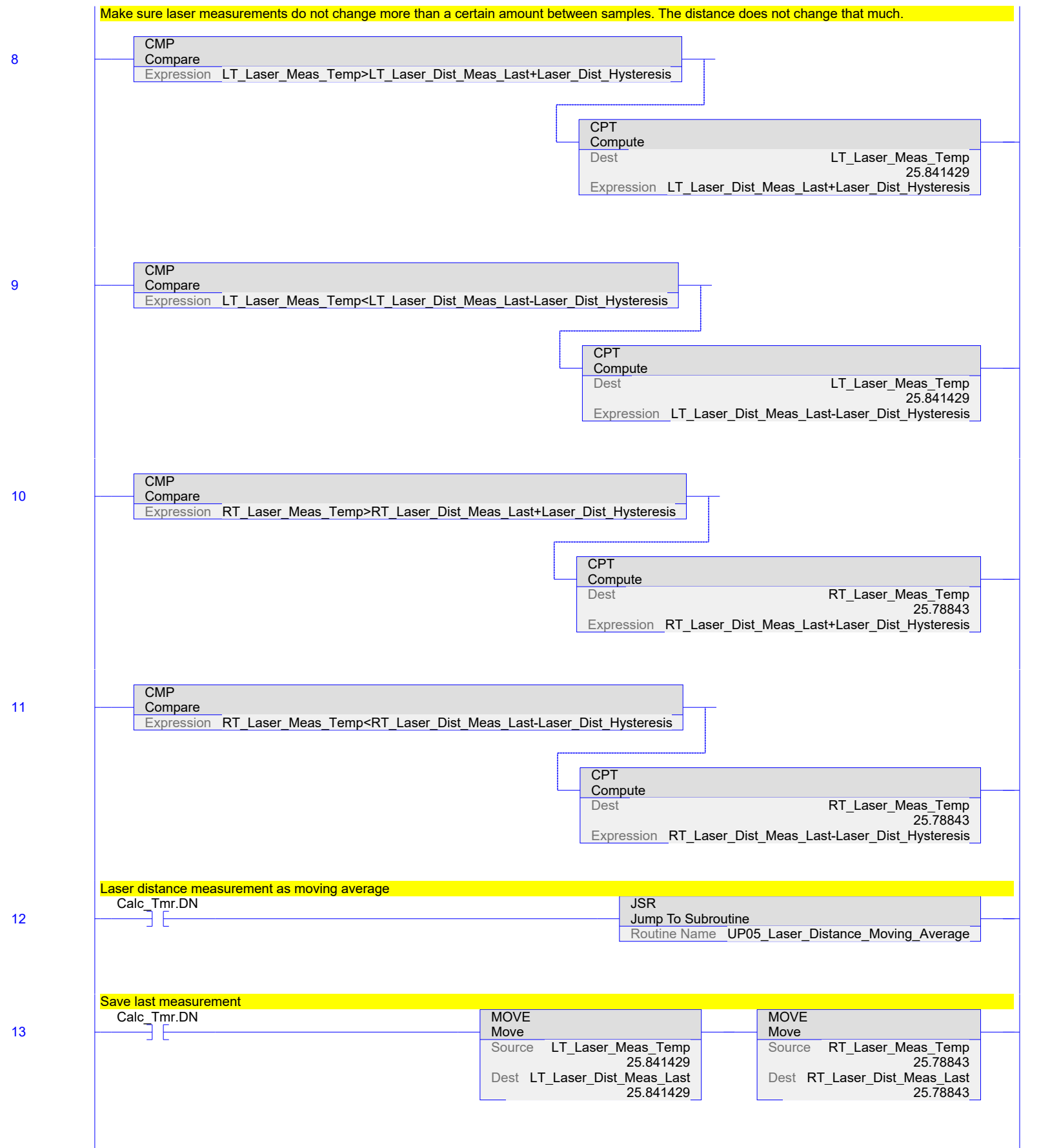












14

Left laser distance measurement

Distance measurement
of left laser, in mm

CPT	
Compute	
Dest	Left_Laser_Dist 258.84418
Expression	(LT_Laser_Dist_Filtered/100.0)*1000.0

15

Right laser distance measurement

Distance measurement
of right laser, in
mm

CPT	
Compute	
Dest	Right_Laser_Dist 257.84503
Expression	(RT_Laser_Dist_Filtered/100.0)*1000.0

16

Left radius of reel

Radius of left reel,
in mm

CPT	
Compute	
Dest	Left_Roll_Radius 26.155823
Expression	Left_Laser_Dist_ToReelCenter-Left_Laser_Dist

17

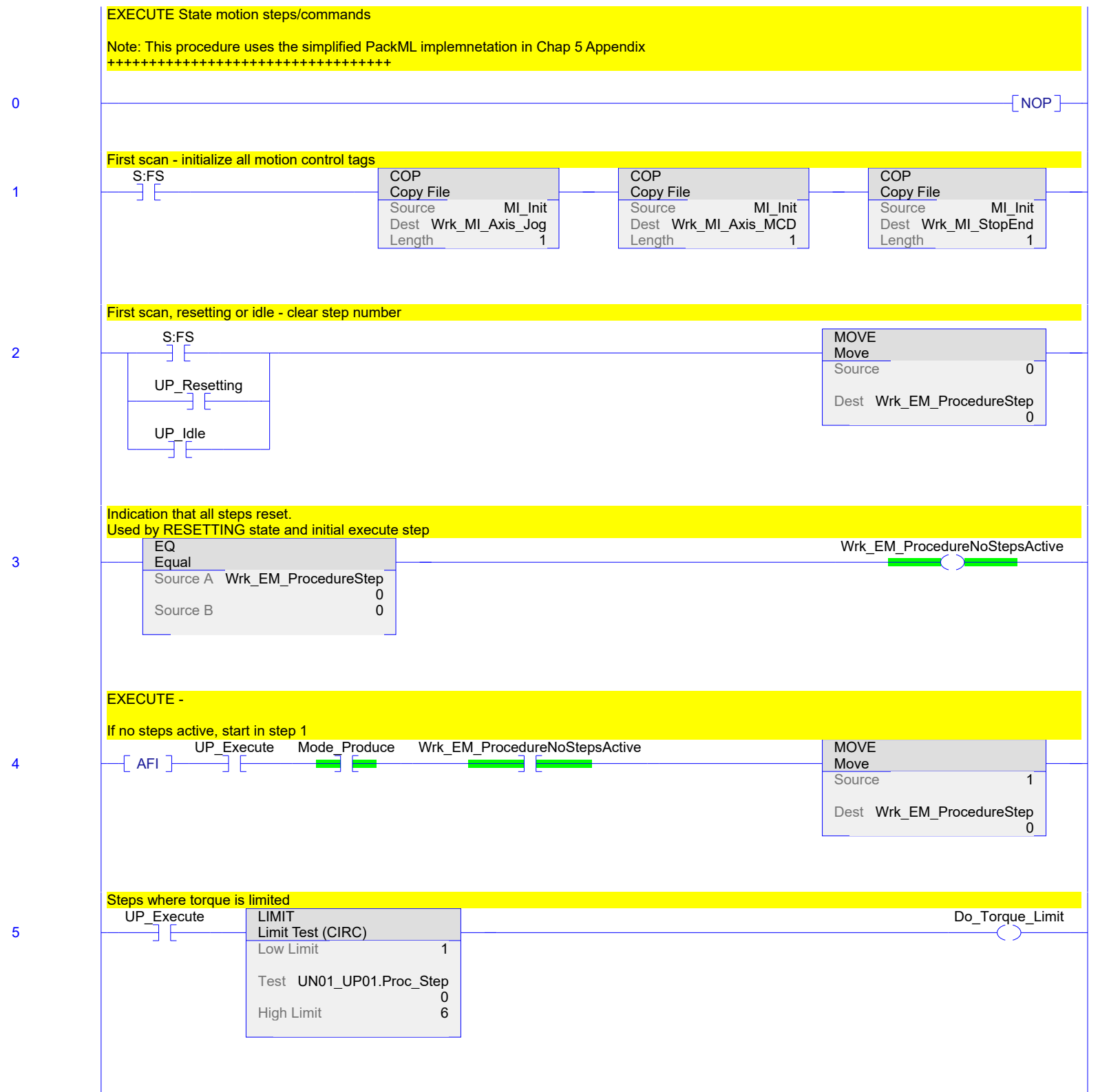
Right radius of reel

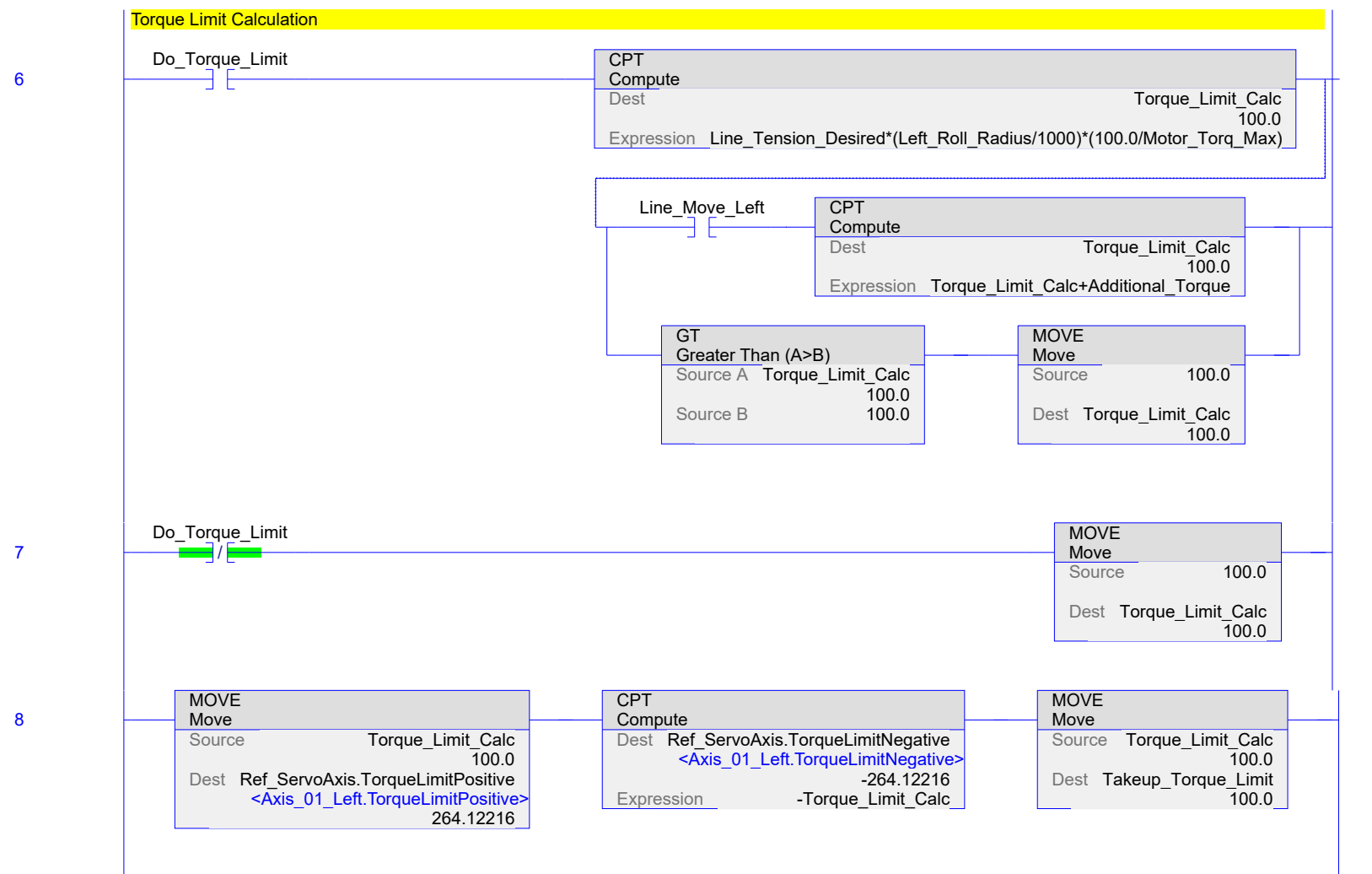
Radius of right
reel, in mm

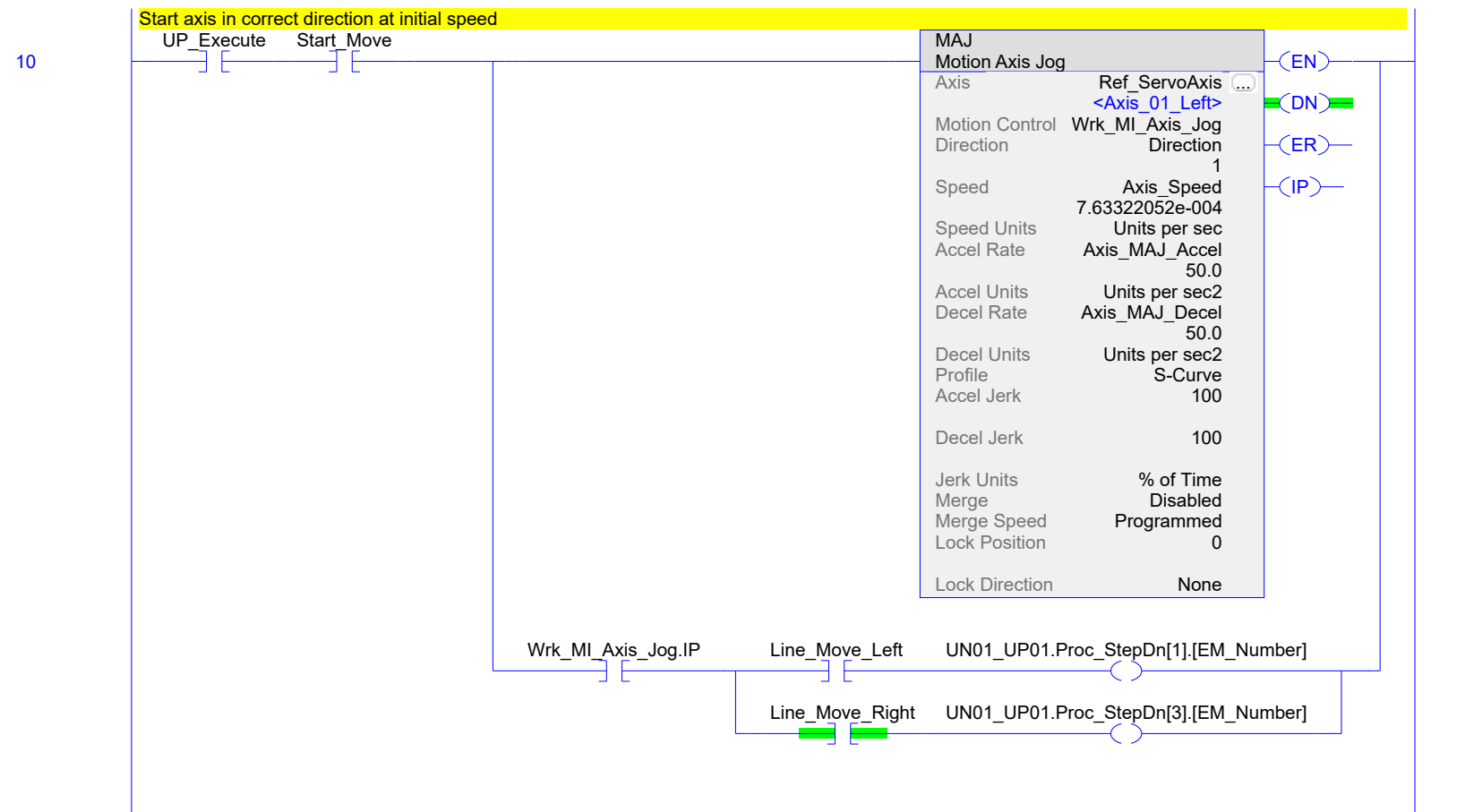
CPT	
Compute	
Dest	Right_Roll_Radius 27.154968
Expression	Right_Laser_Dist_ToReelCenter-Right_Laser_Dist

(End)

```
1 // Moving average filtering of laser measurement
2 // Values in current array are shifted up one index
3 // Current measurement placed in index 0
4 // Filtered measurement is average of elements in array
5 for N := NSamples-1
6 to 1 by -1
7 do
8     LT_Laser_Dist_Array[N] := LT_Laser_Dist_Array[N-1];
9     RT_Laser_Dist_Array[N] := RT_Laser_Dist_Array[N-1];
10 end_for;
11 LT_Laser_Dist_Array[0] := LT_Laser_Meas_Temp;
12 RT_Laser_Dist_Array[0] := RT_Laser_Meas_Temp;
13 //
14 LT_Laser_Dist_Filtered := 0.0;
15 RT_Laser_Dist_Filtered := 0.0;
16 for N := 0
17 to NSamples-1
18 do
19     LT_Laser_Dist_Filtered := LT_Laser_Dist_Filtered + LT_Laser_Dist_Array[N];
20     RT_Laser_Dist_Filtered := RT_Laser_Dist_Filtered + RT_Laser_Dist_Array[N];
21 end_for;
22 LT_Laser_Dist_Filtered := LT_Laser_Dist_Filtered / NSamples;
23 RT_Laser_Dist_Filtered := RT_Laser_Dist_Filtered / NSamples;
```

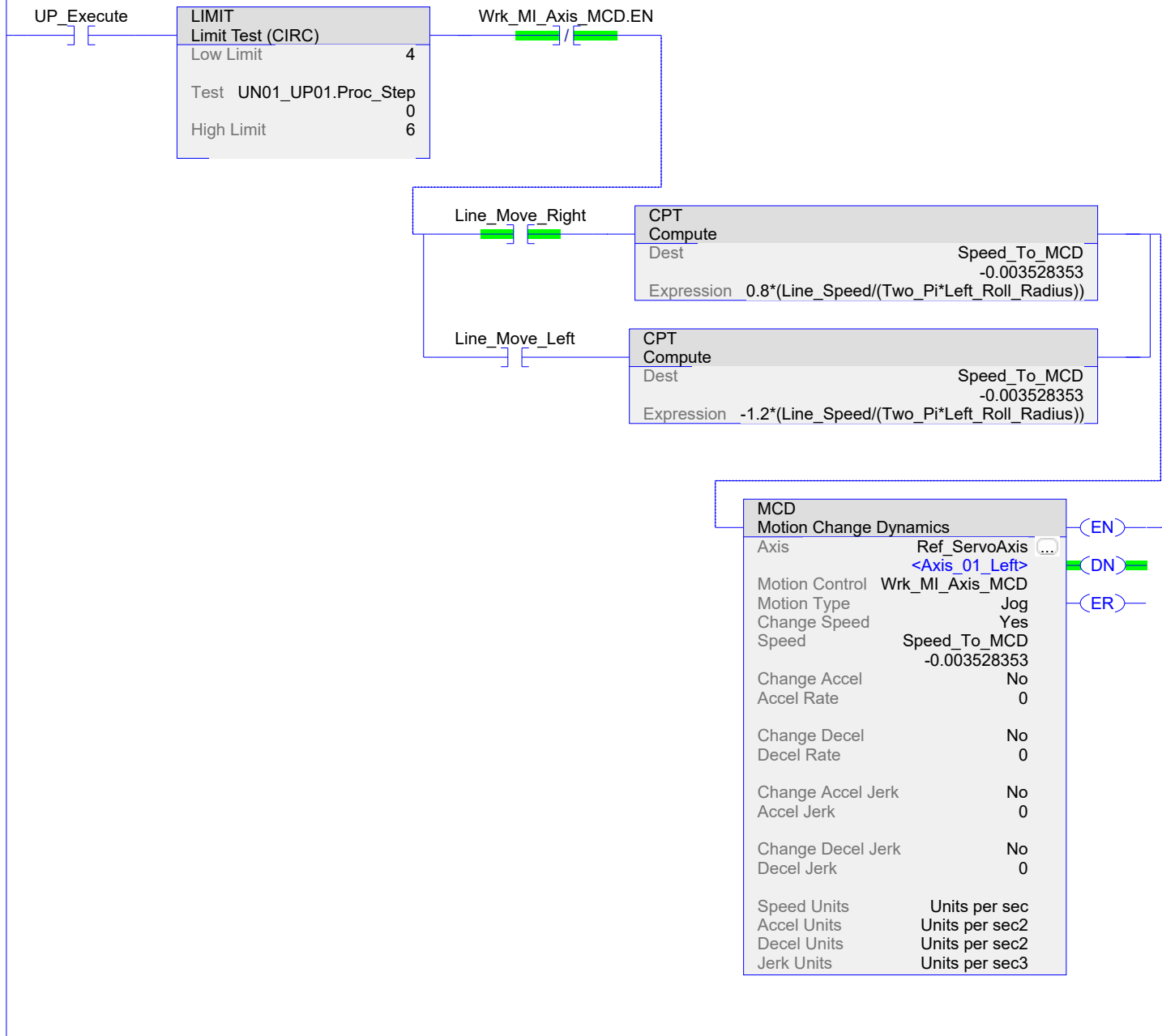




UP Procedure steps 4, 5, 6 - ramp up line speed, maintain, then decrease.

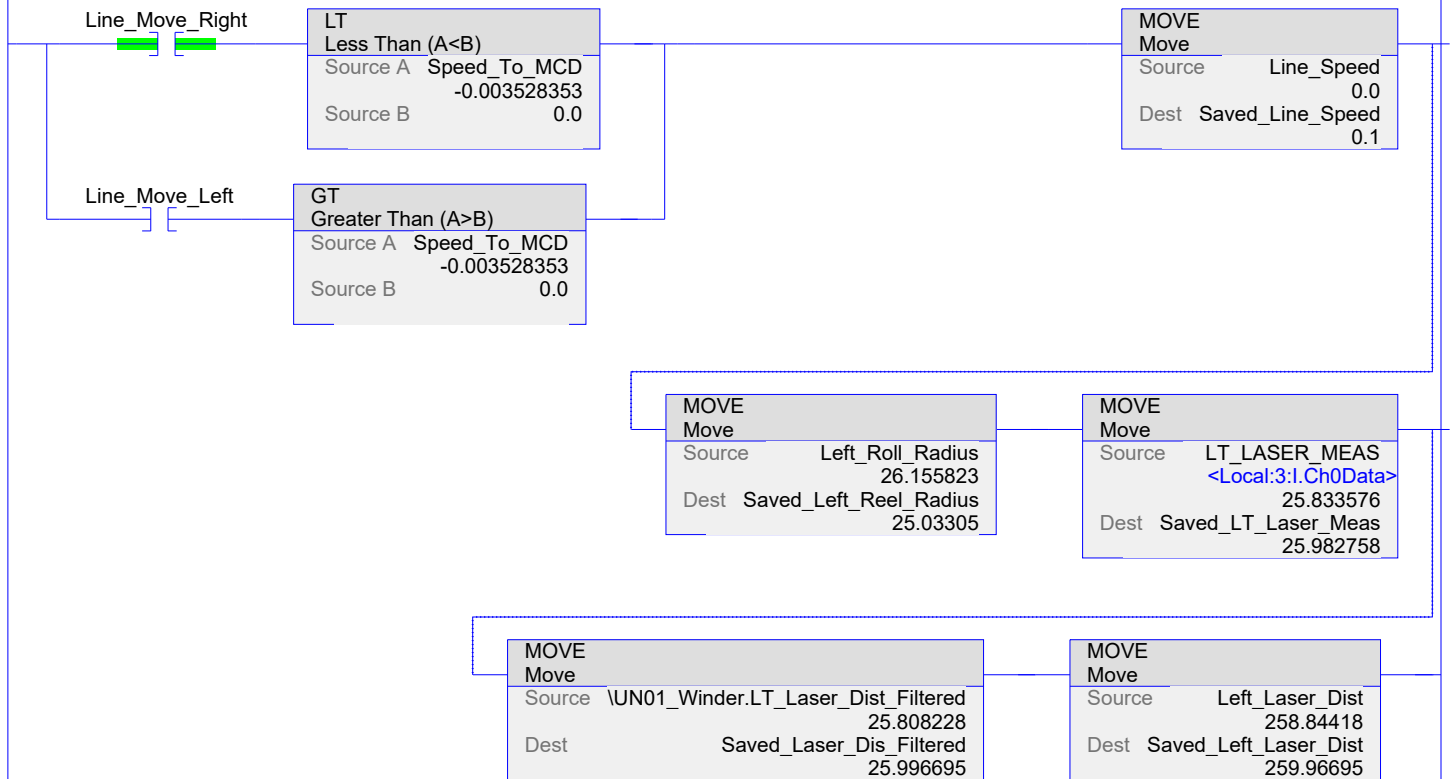
Change speed of motor to match desired line speed..

11



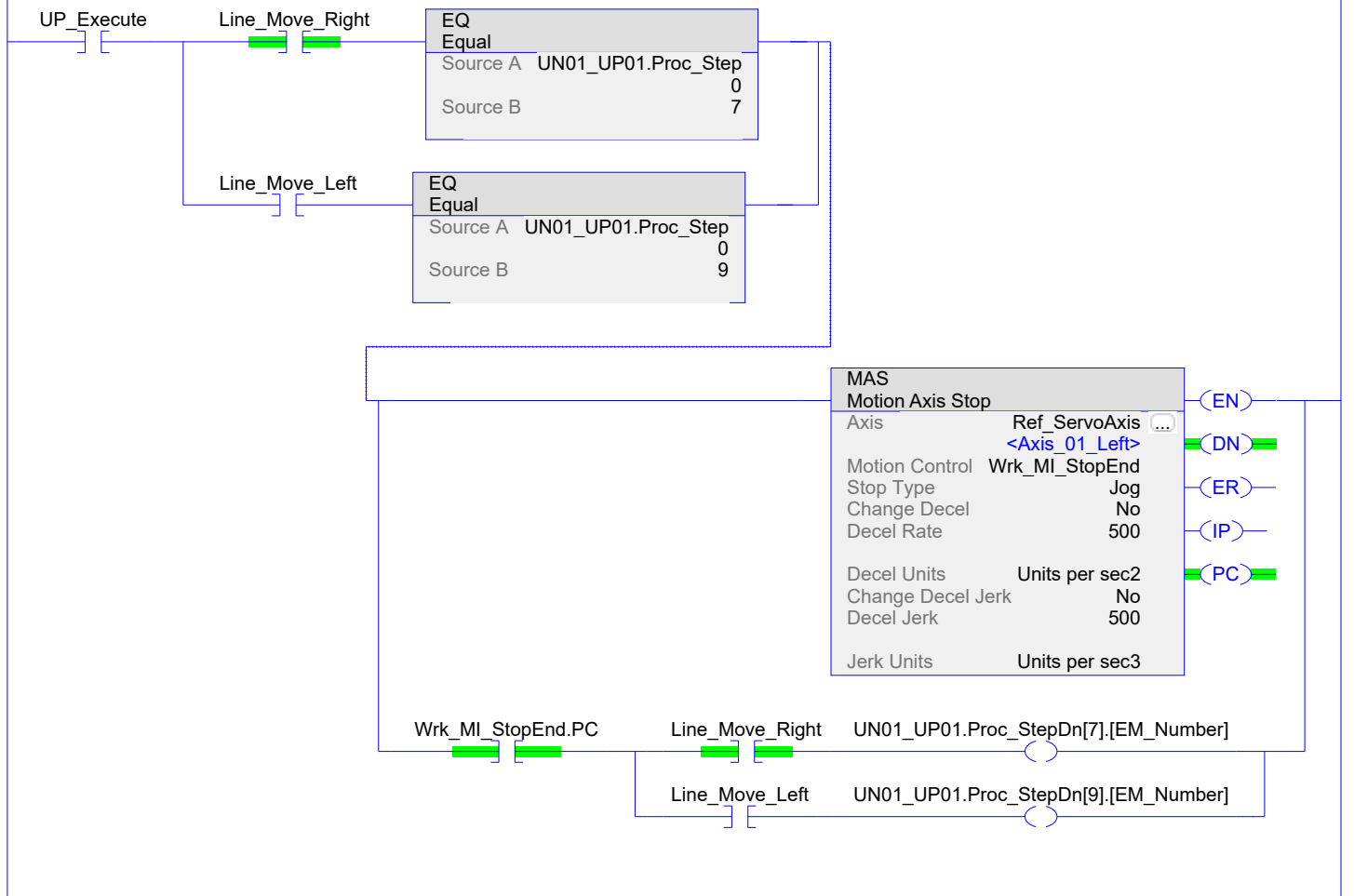
If sign of Speed_to_MCD is opposite of what it should be, trap intermediate calculation results.

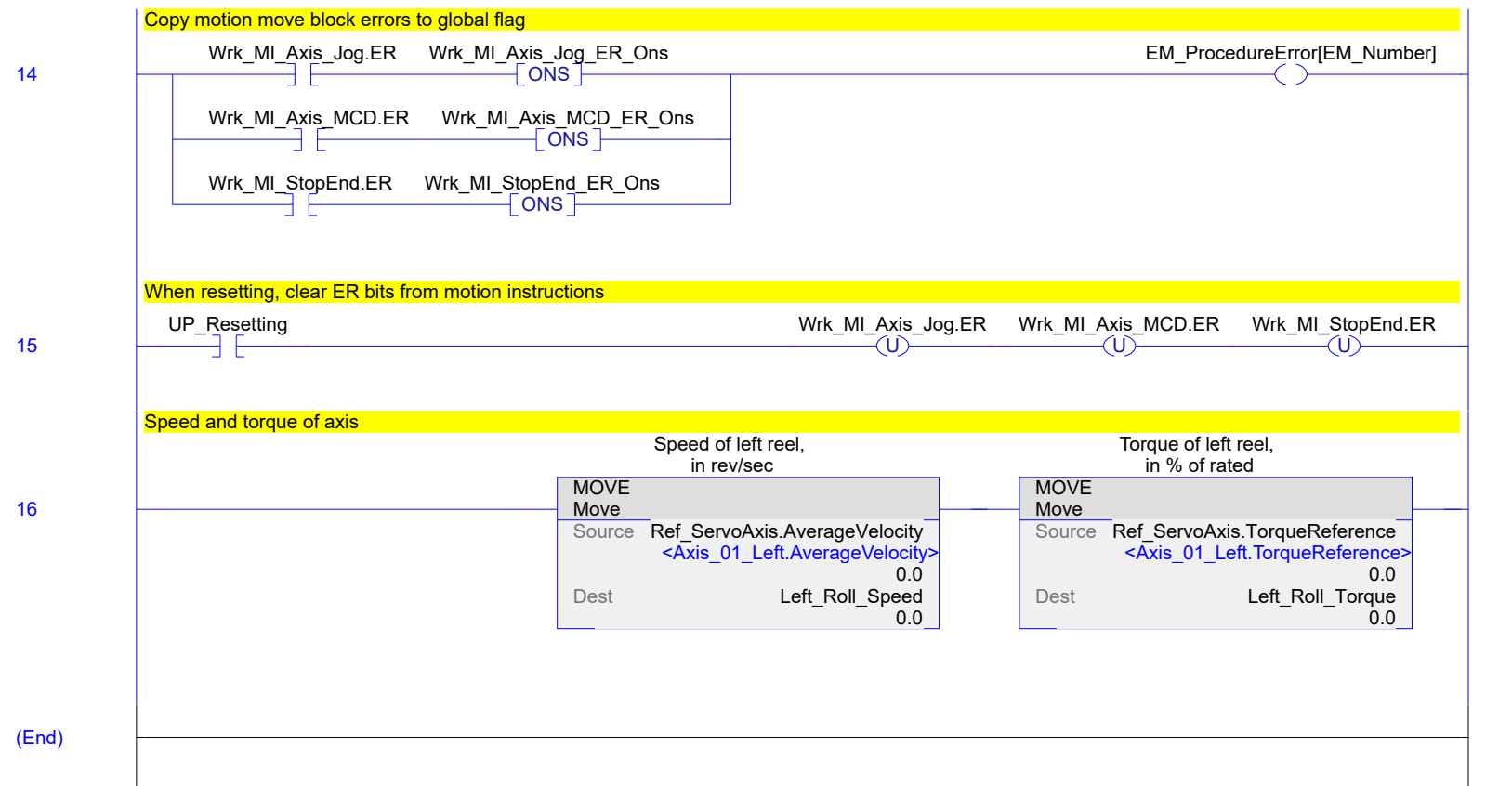
12



Stop axis

13





Axis-related blocks for PackML States

Note that EM_Number program tag contains the axis number. Index into Boolean arrays that communicate state "done" to main PackML state machine.

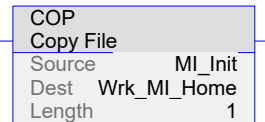
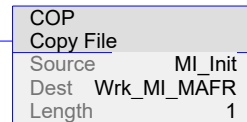
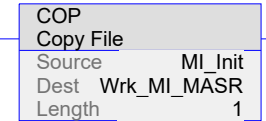
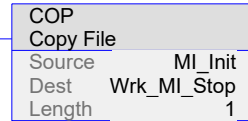
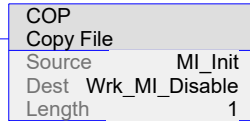
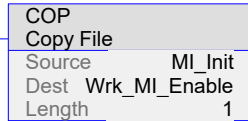
Ref_ServoAxis is aliased to the real Axis tag in the controller tags.

+++++

[NOP]

First scan - initialize all motion control tags

s:fs



STARTING -

Shutdown Reset

Fault Reset

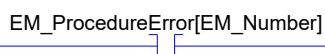
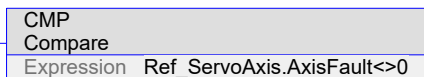
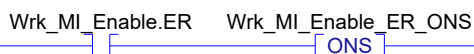
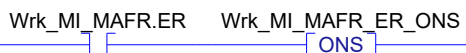
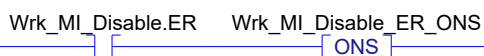
Servo On

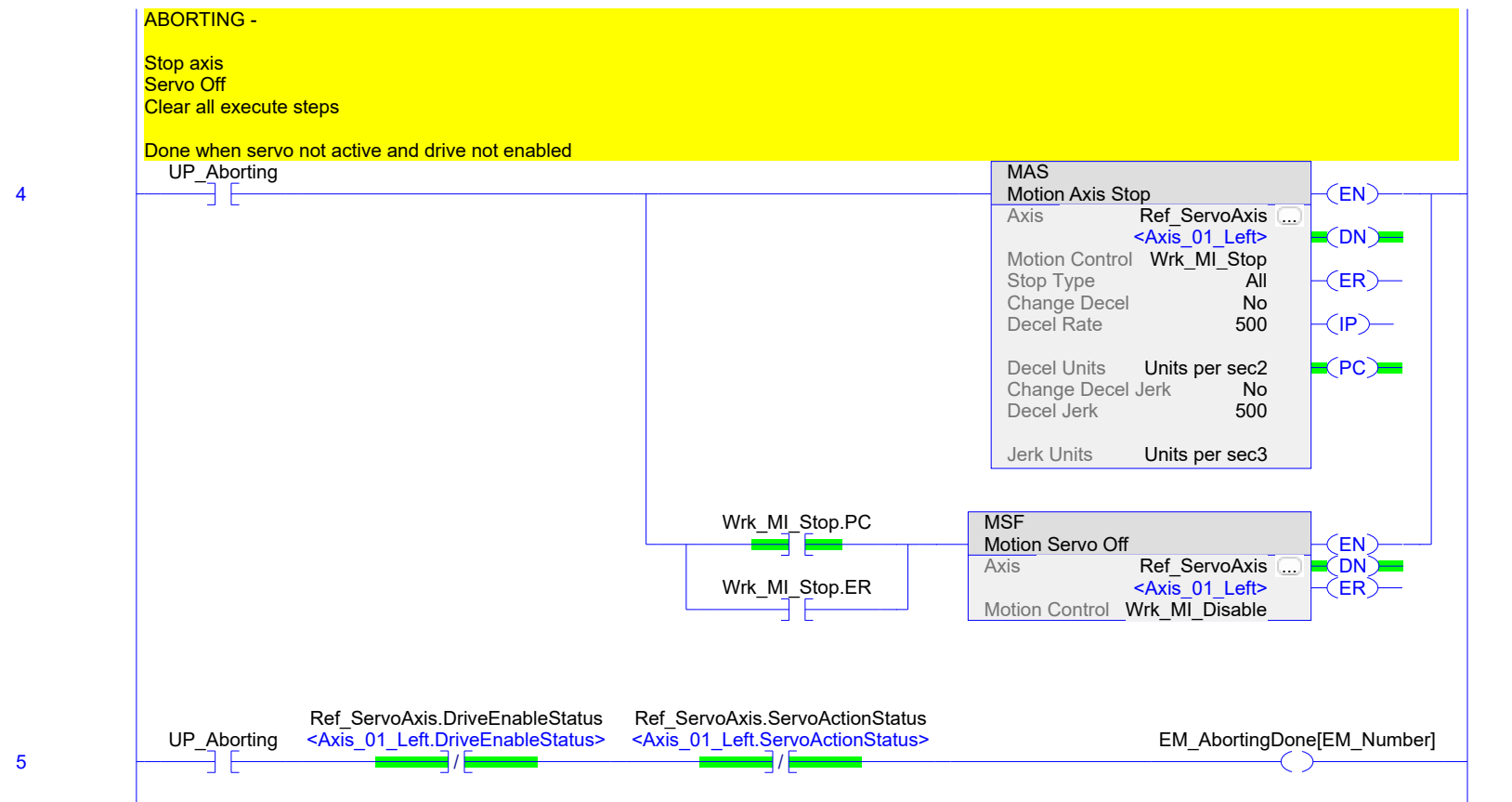
Done when servo active and drive enabled

Skip if servo active and drive enable status



Copy any motion instruction errors or axis fault to axis error Boolean





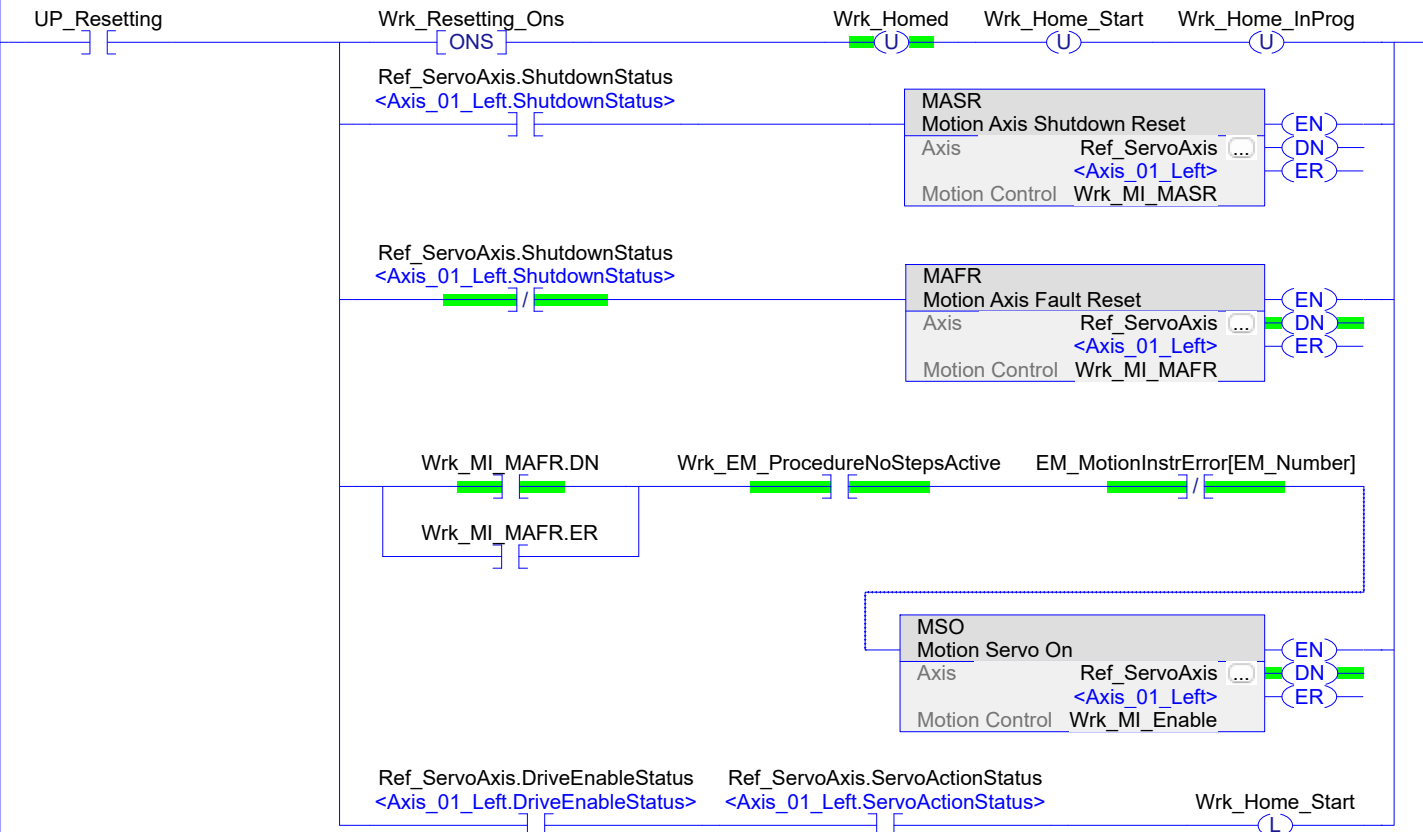
RESETTING -

Clear all faults on axis, clear all instruction ER bits.

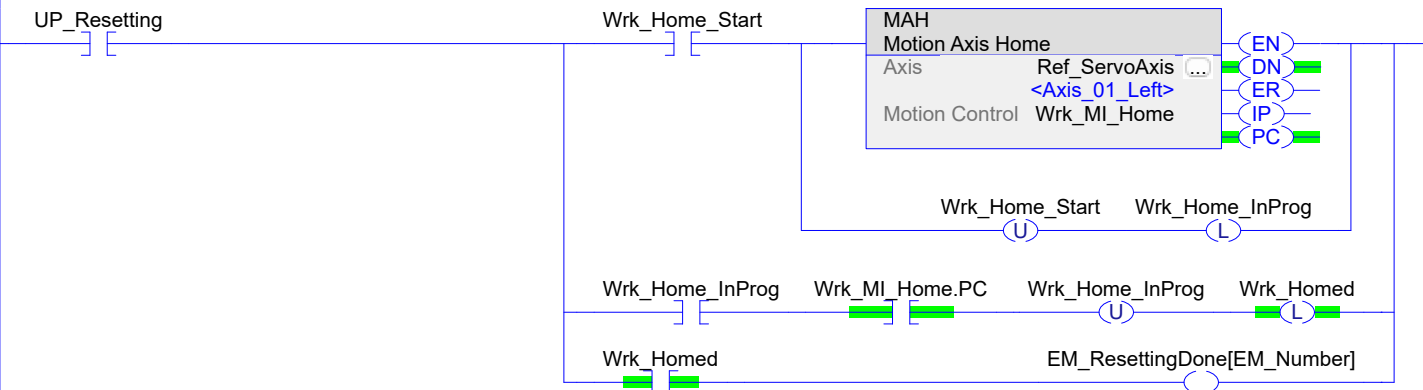
Shutdown Reset
Fault Reset
Servo On
Home Axis

Complete when servo active and drive enabled and axis homed

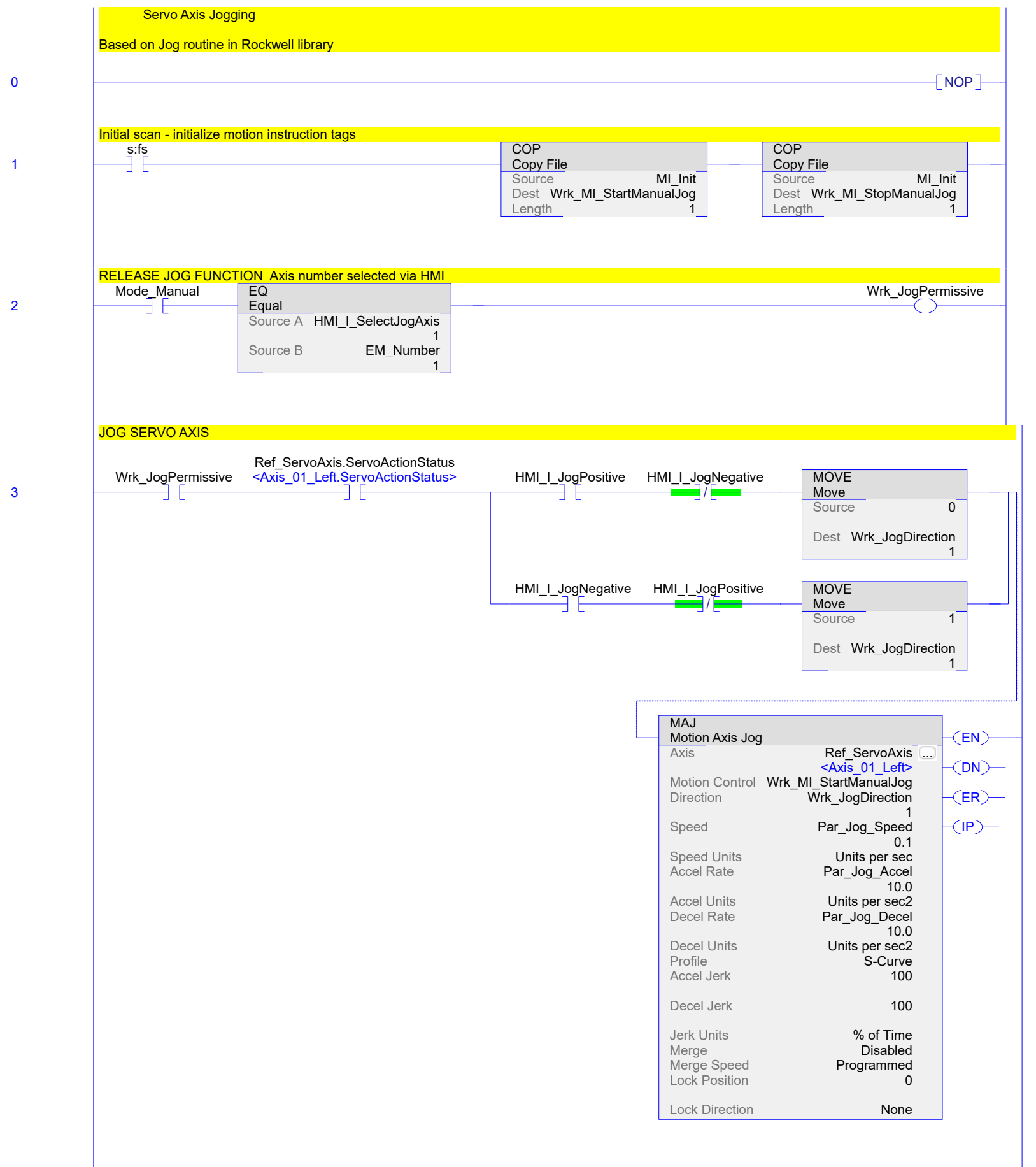
6

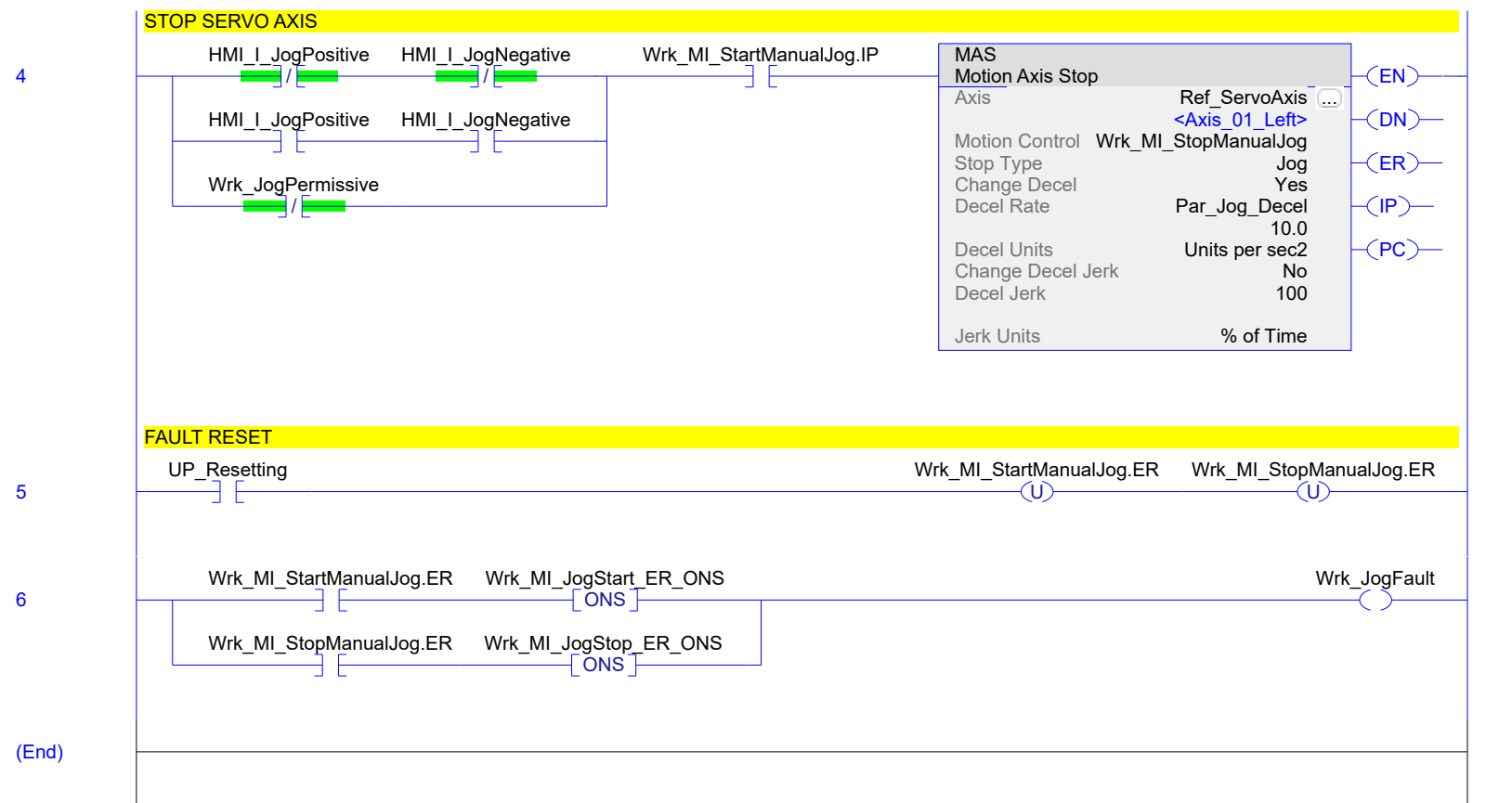


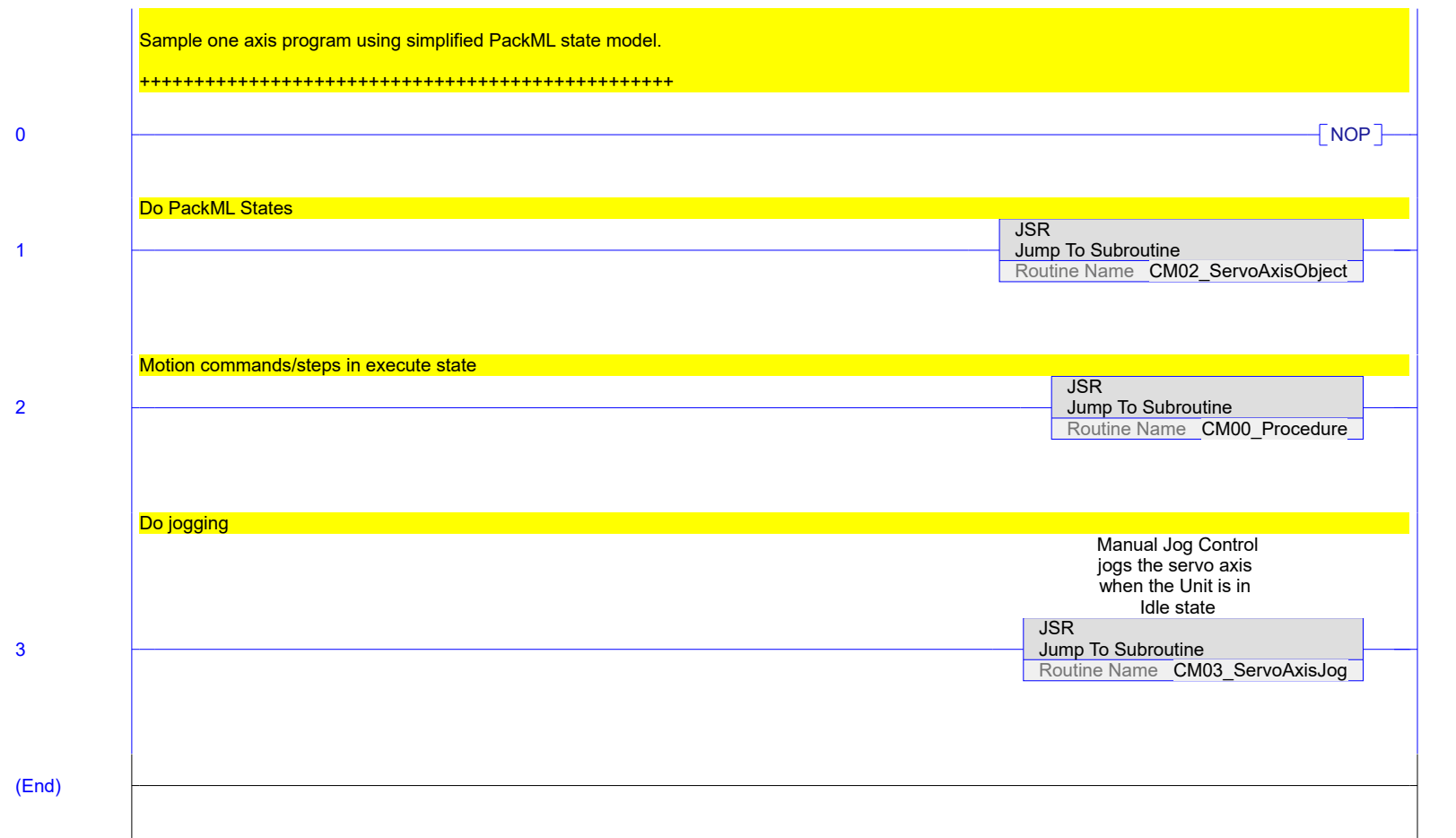
7

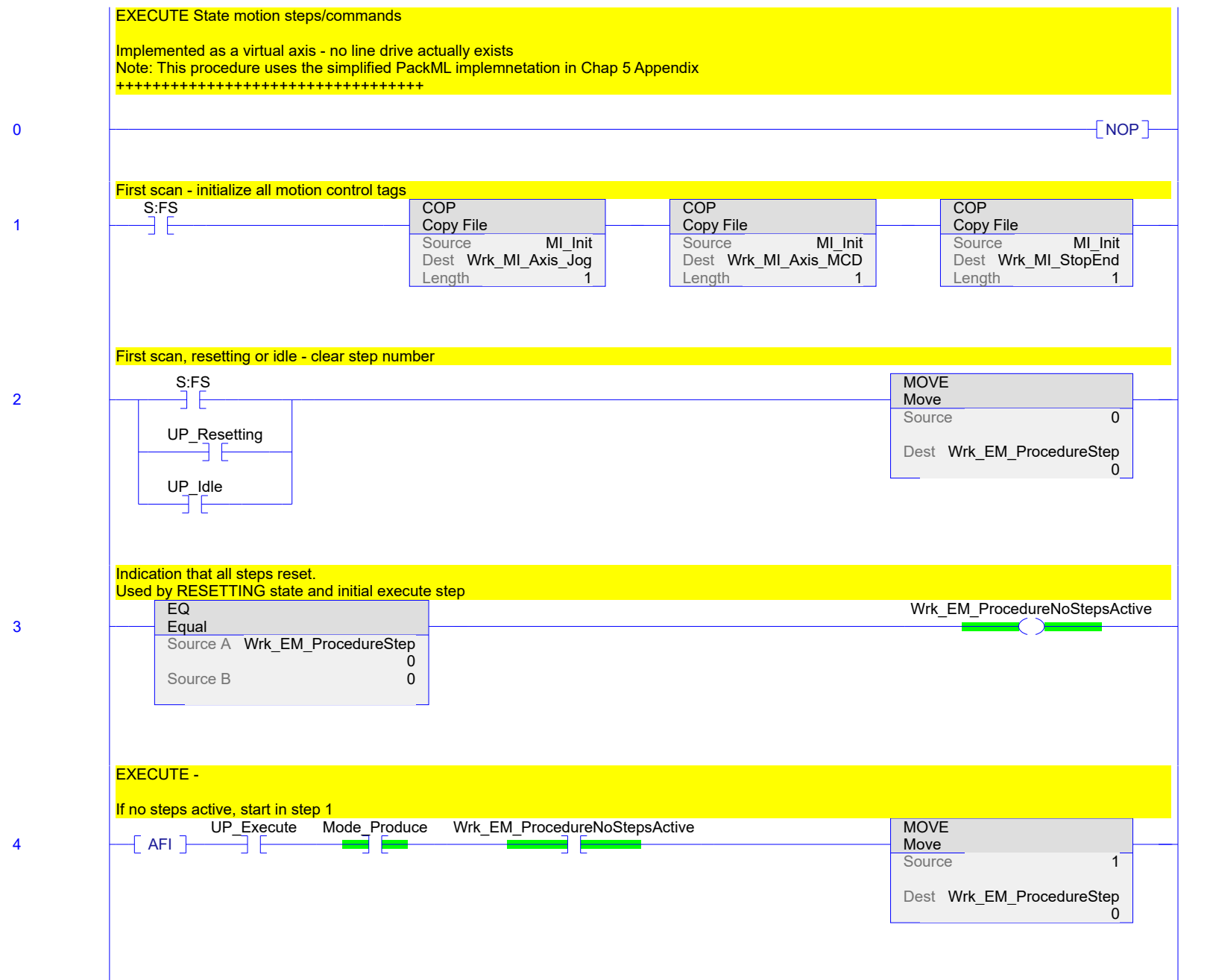


(End)



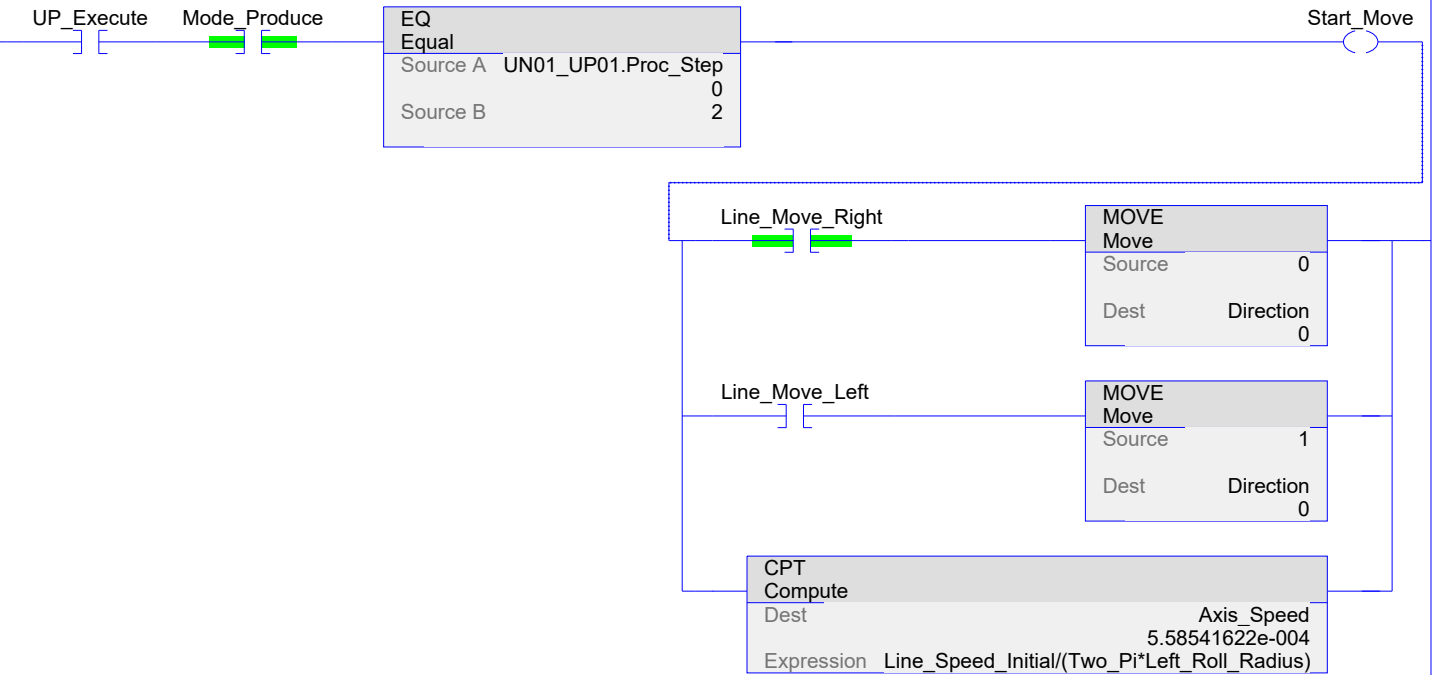






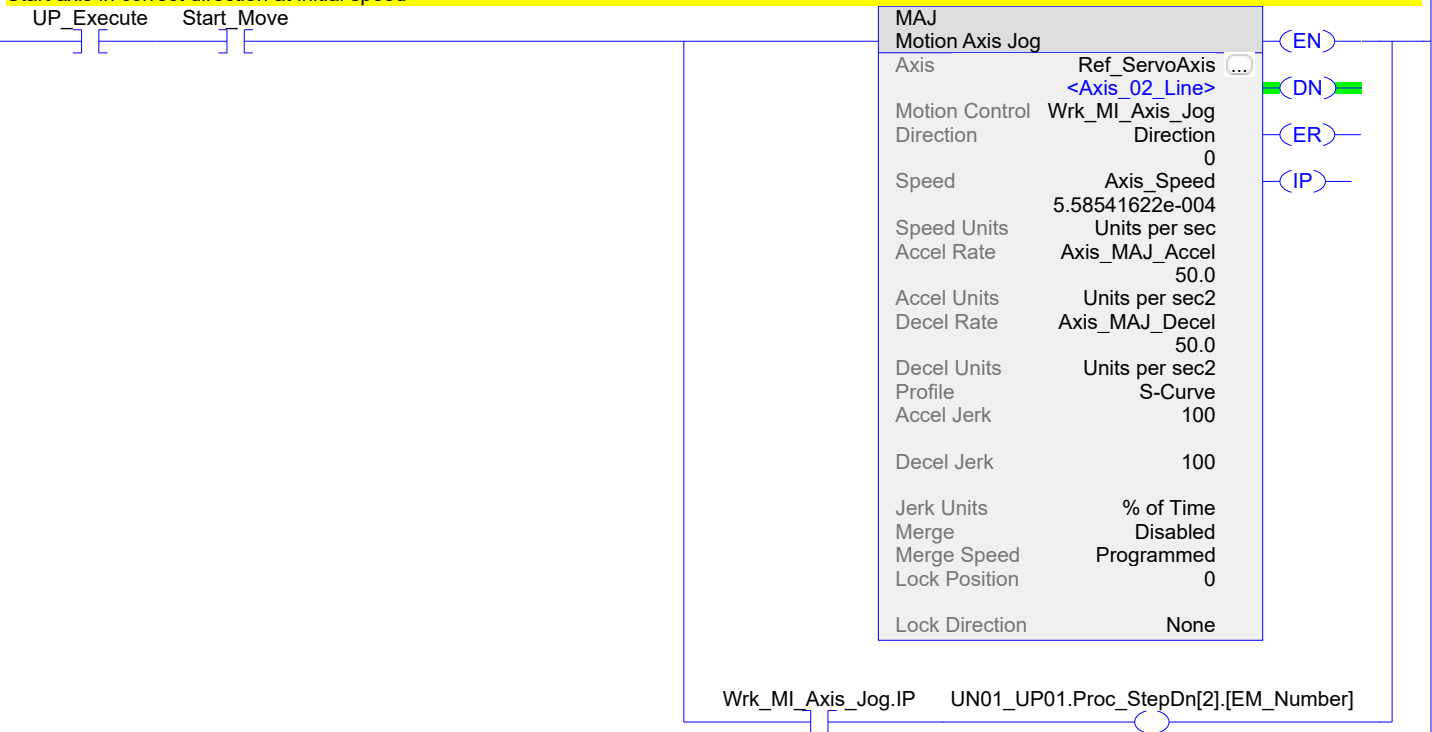
5

UP Procedure Step 2 - set direction and axis speed - uses left roll radius



6

Start axis in correct direction at initial speed



UP Procedure steps 4 to 6 - Change speed of motor to match desired line speed.
Speed is negative if moving left

UP_Execute

LIMIT	
Limit Test (CIRC)	
Low Limit	4
Test	UN01_UP01.Proc_Step
	0
High Limit	6

Wrk_Ml_Axis_MCD.EN

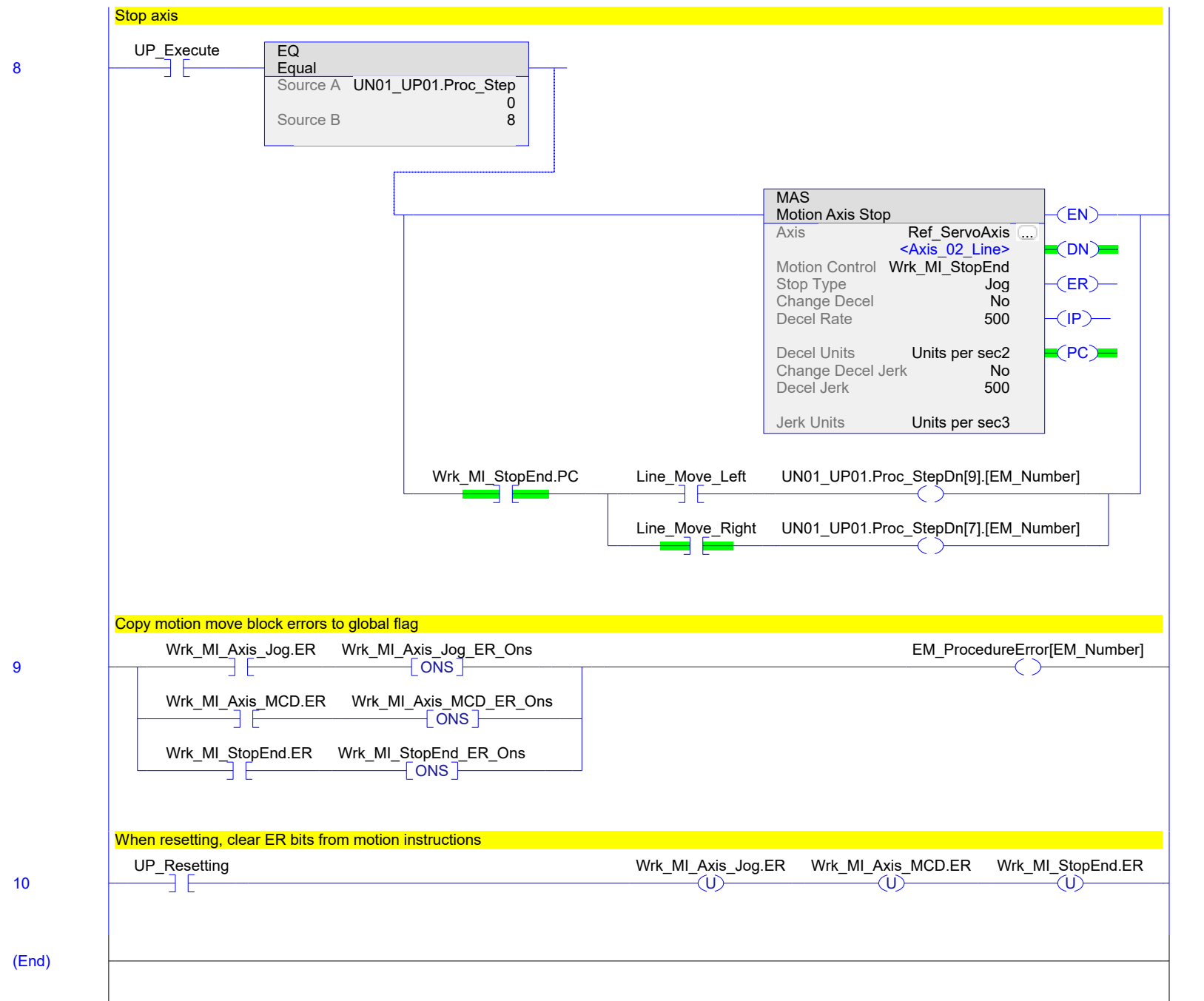
Line_Move_Right

Line_Move_Left

CPT	
Compute	
Dest	Speed_To_MCD
	1.18208339e-003
Expression	Line_Speed/(Two_Pi*Left_Roll_Radius)

CPT	
Compute	
Dest	Speed_To_MCD
	1.18208339e-003
Expression	-Line_Speed/(Two_Pi*Left_Roll_Radius)

MCD	
Motion Change Dynamics	
Axis	Ref_ServoAxis
	<Axis_02_Line>
Motion Control	Wrk_Ml_Axis_MCD
Motion Type	Jog
Change Speed	Yes
Speed	Speed_To_MCD
	1.18208339e-003
Change Accel	No
Accel Rate	0
Change Decel	No
Decel Rate	0
Change Accel Jerk	No
Accel Jerk	0
Change Decel Jerk	No
Decel Jerk	0
Speed Units	Units per sec
Accel Units	Units per sec ²
Decel Units	Units per sec ²
Jerk Units	Units per sec ³



Axis-related blocks for PackML States - VIRTUAL AXIS

Note that EM_Number program tag contains the axis number. Index into Boolean arrays that communicate state "done" to main PackML state machine.

Ref_ServoAxis is aliased to the real Axis tag in the controller tags.

+++++

[NOP]

First scan - initialize all motion control tags

s:fs

COP Copy File		
Source	MI_Init	
Dest	Wrk_MI_Stop	
Length	1	

COP Copy File		
Source	MI_Init	
Dest	Wrk_MI_MASR	
Length	1	

COP Copy File		
Source	MI_Init	
Dest	Wrk_MI_MAFR	
Length	1	

STARTING -

Shutdown Reset
Fault Reset
Servo On

Done when servo active and drive enabled

Skip if servo active and drive enable status

UP_Starting Ref_ServoAxis.ServoActionStatus
<Axis_02_Line.ServoActionStatus> EM_StartingDone[EM_Number]

Copy any motion instruction errors or axis fault to axis error Boolean

Wrk_MI_Stop.ER Wrk_MI_Stop_ER_ONS

EM_MotionInstrError[EM_Number]

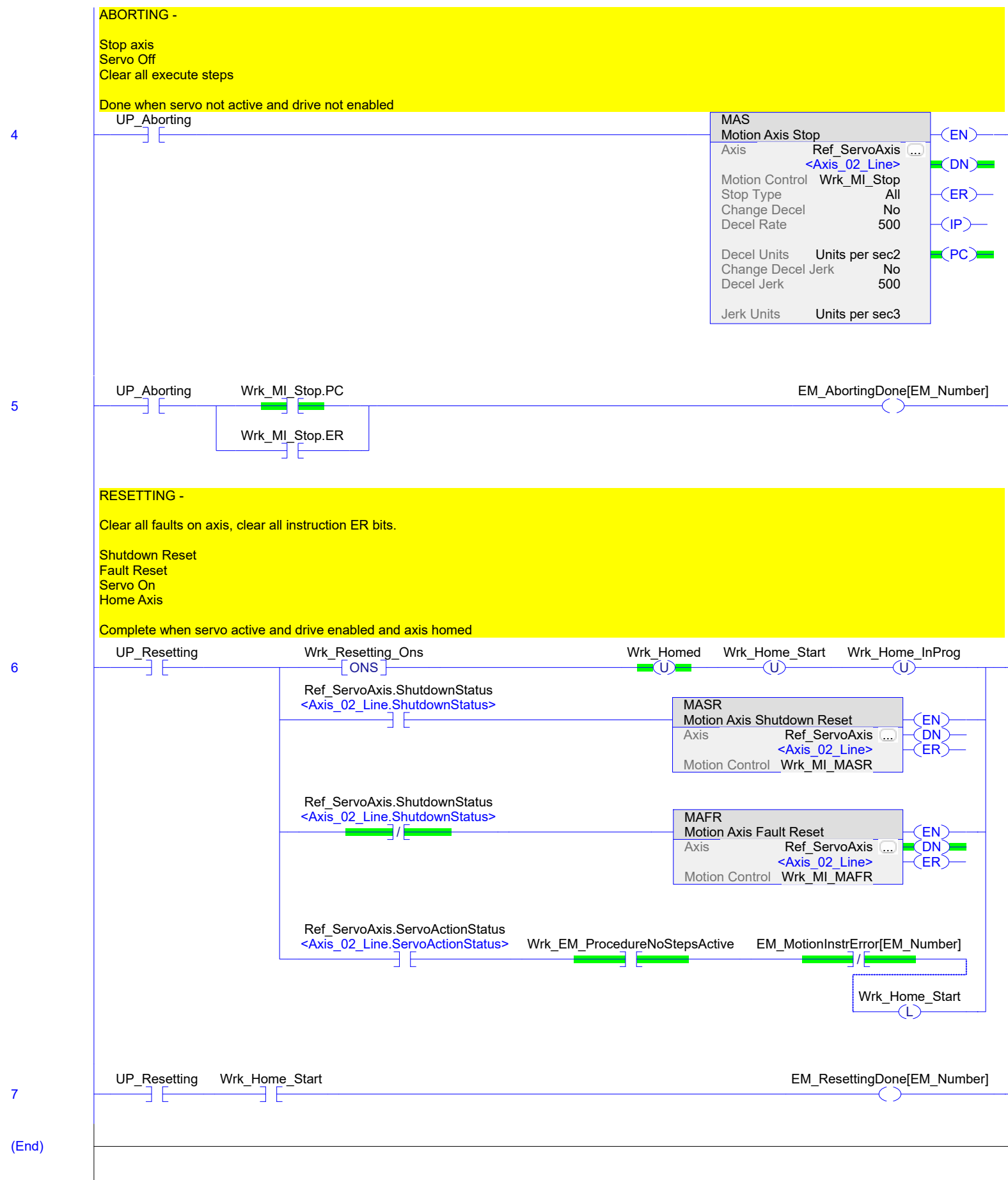
Wrk_MI_MASR.ER Wrk_MI_MASR_ER_ONS

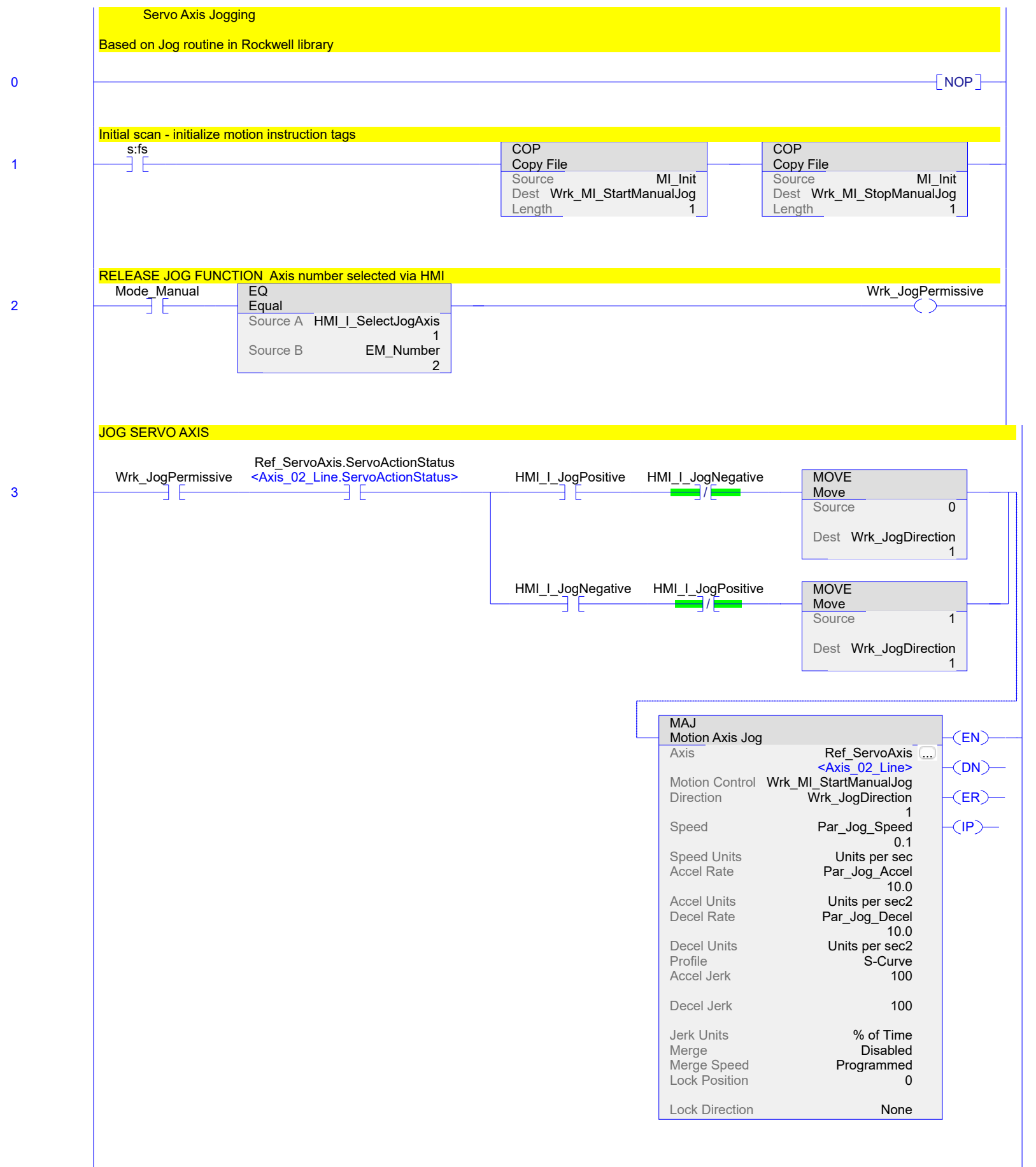
Wrk_MI_MAFR.ER Wrk_MI_MAFR_ER_ONS

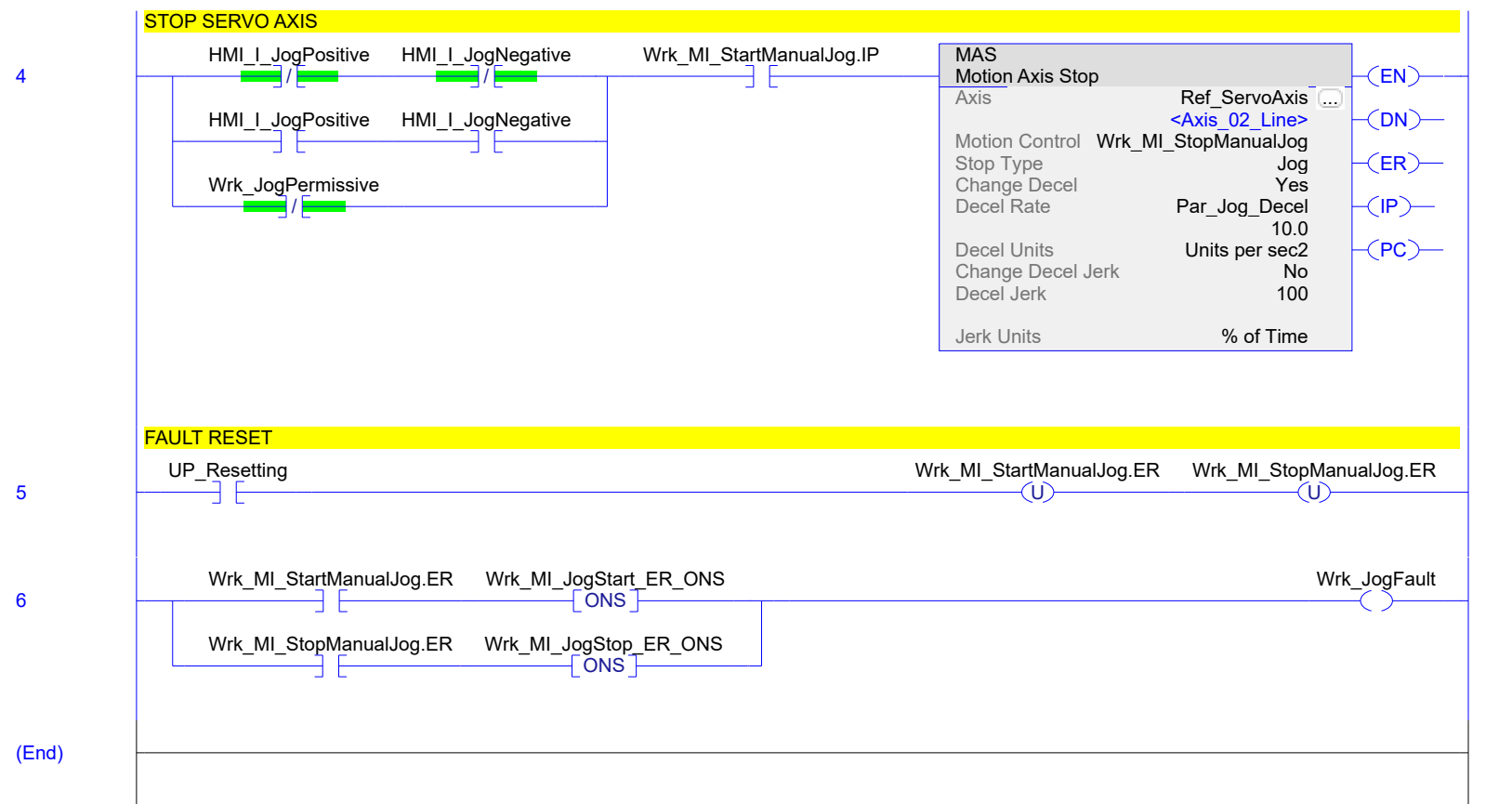
CMP Compare	
Expression	Ref_ServoAxis.AxisFault<>0

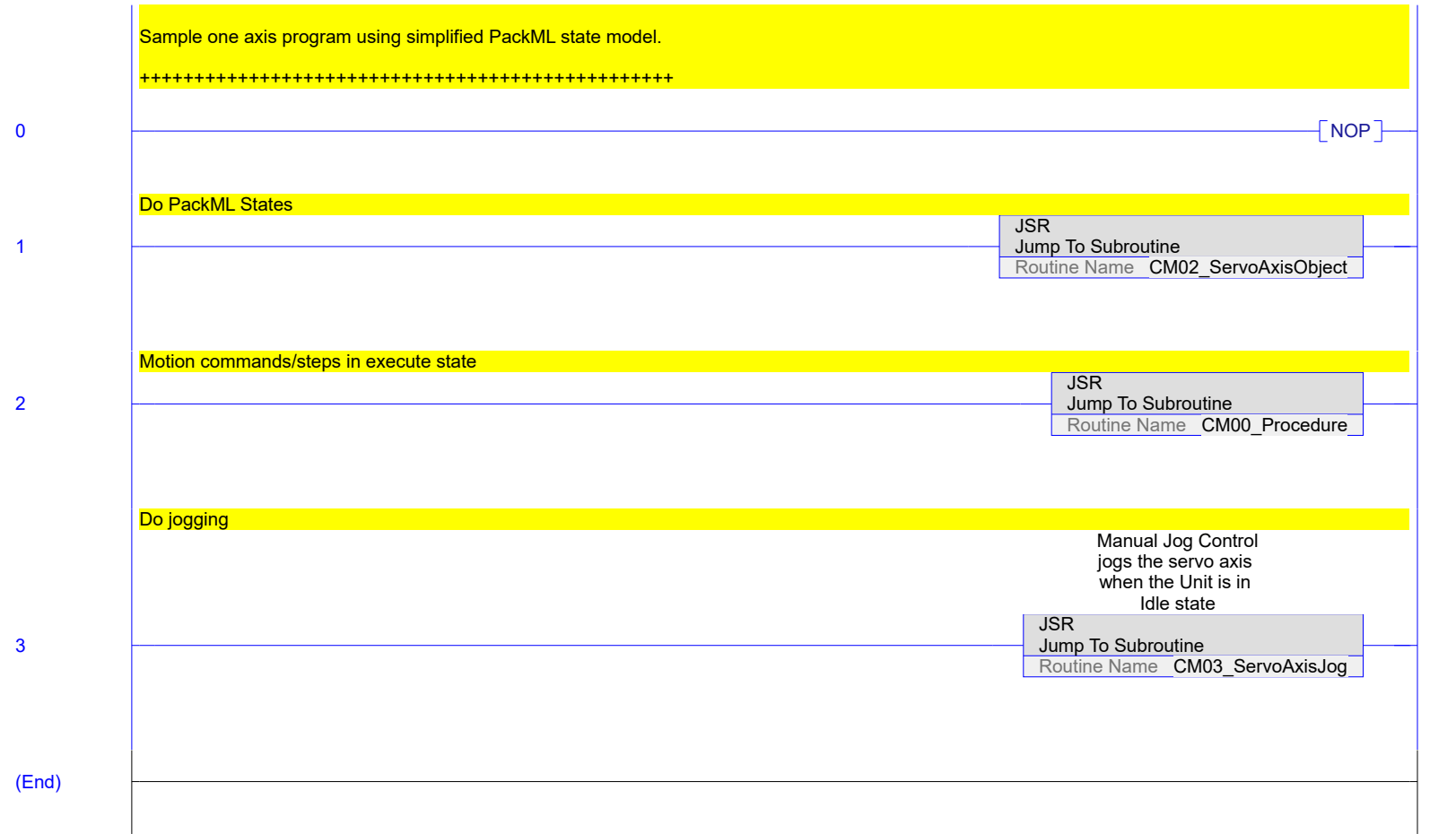
EM_ProcedureError[EM_Number]

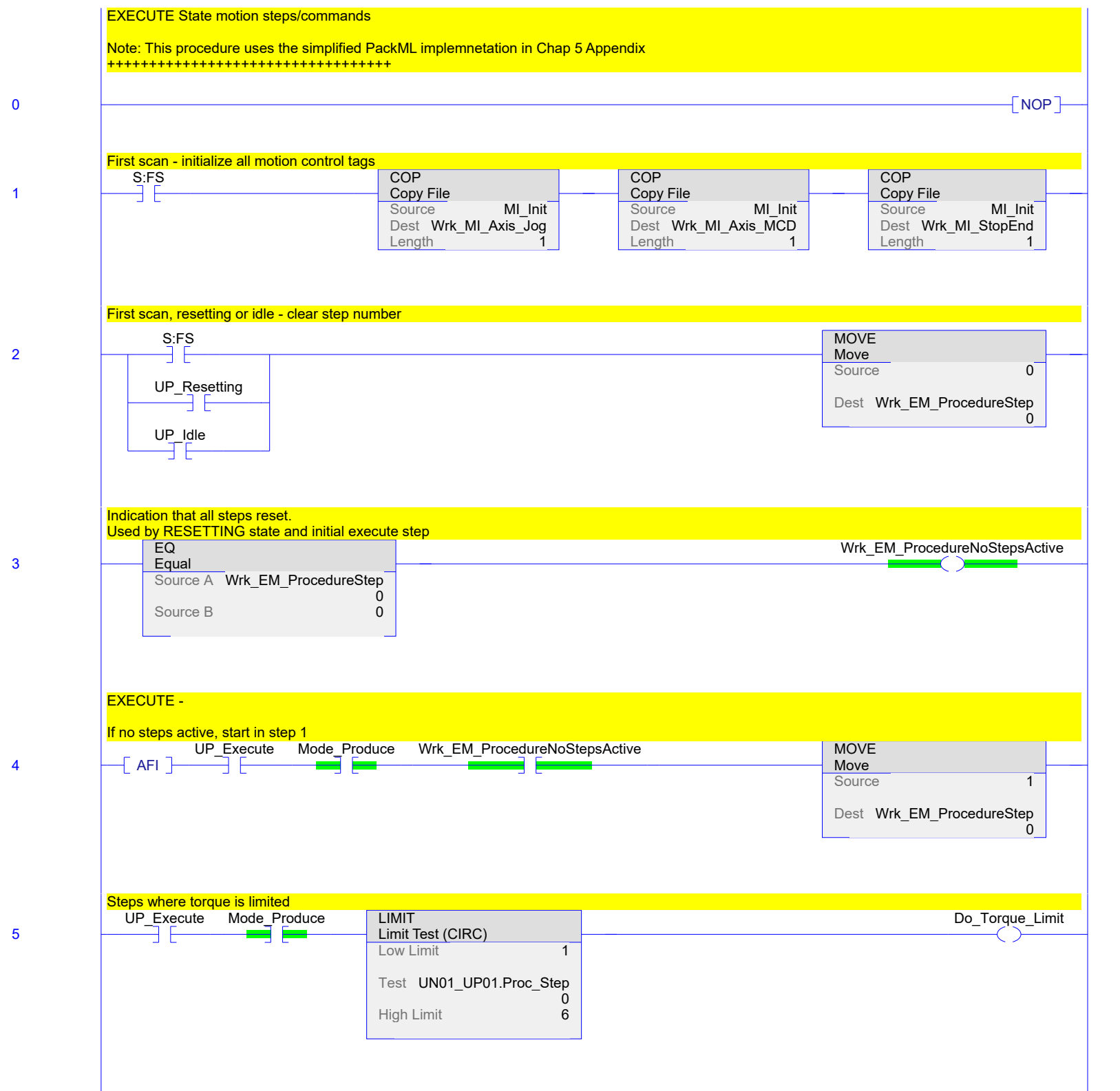
Wrk_JogFault

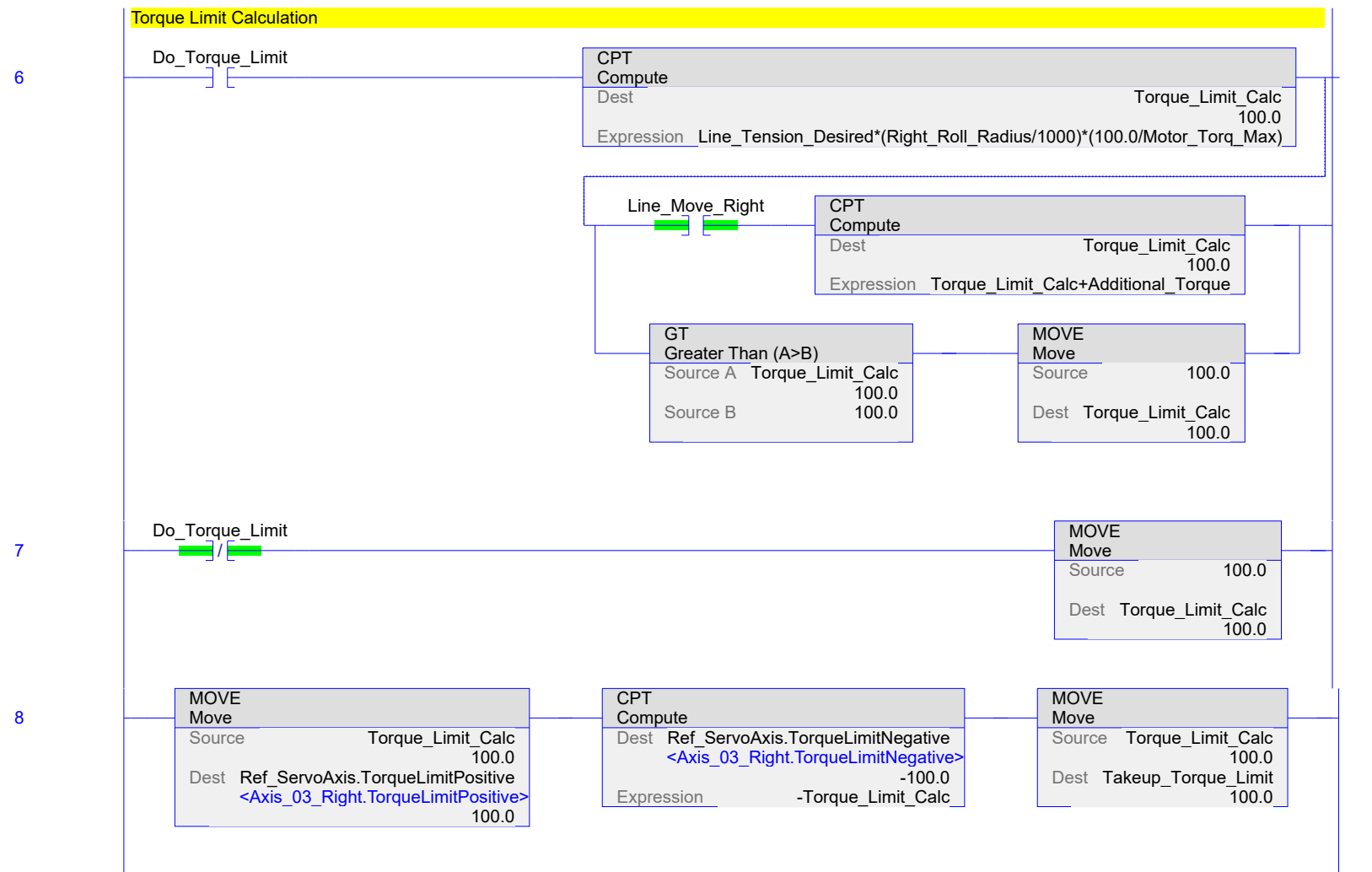




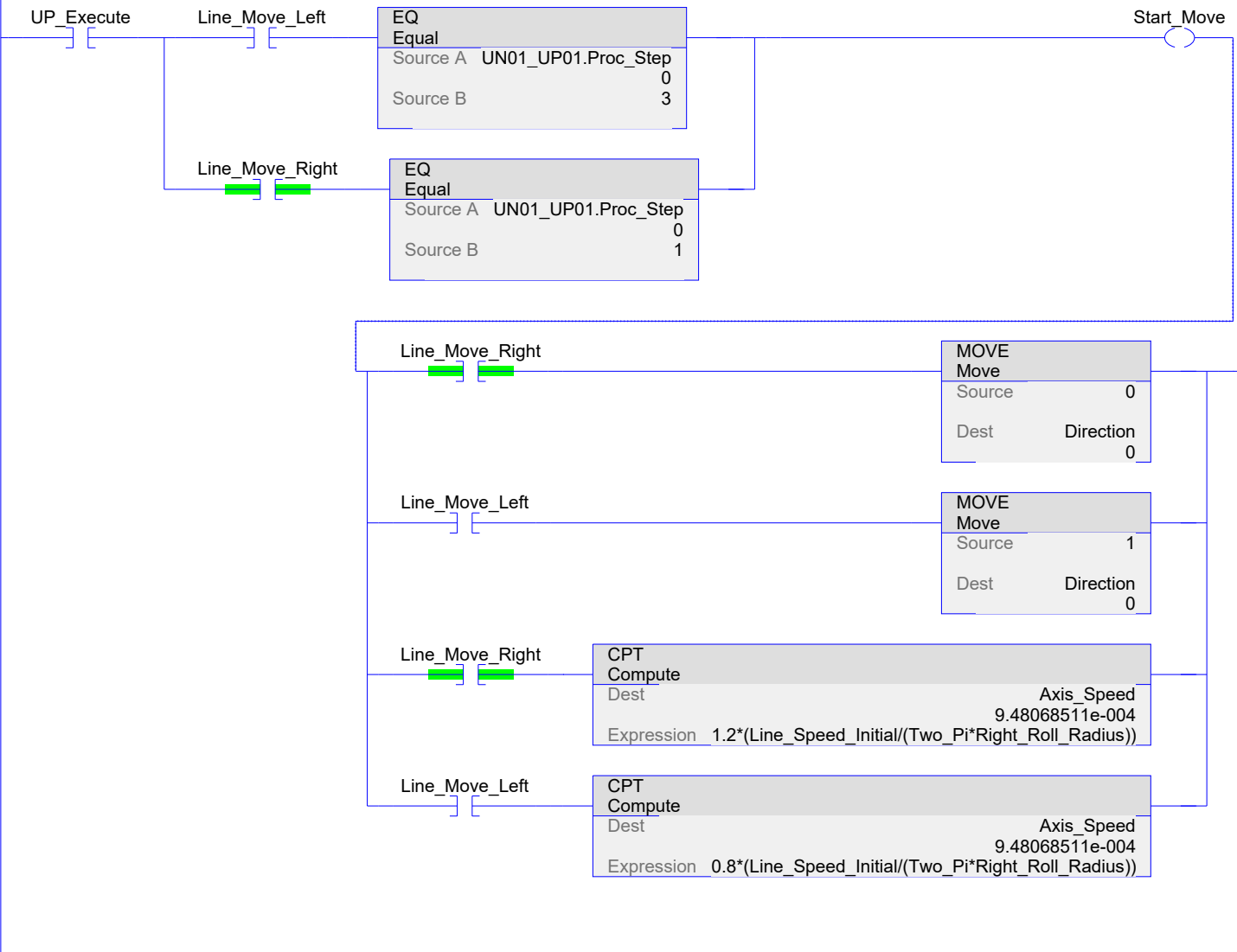




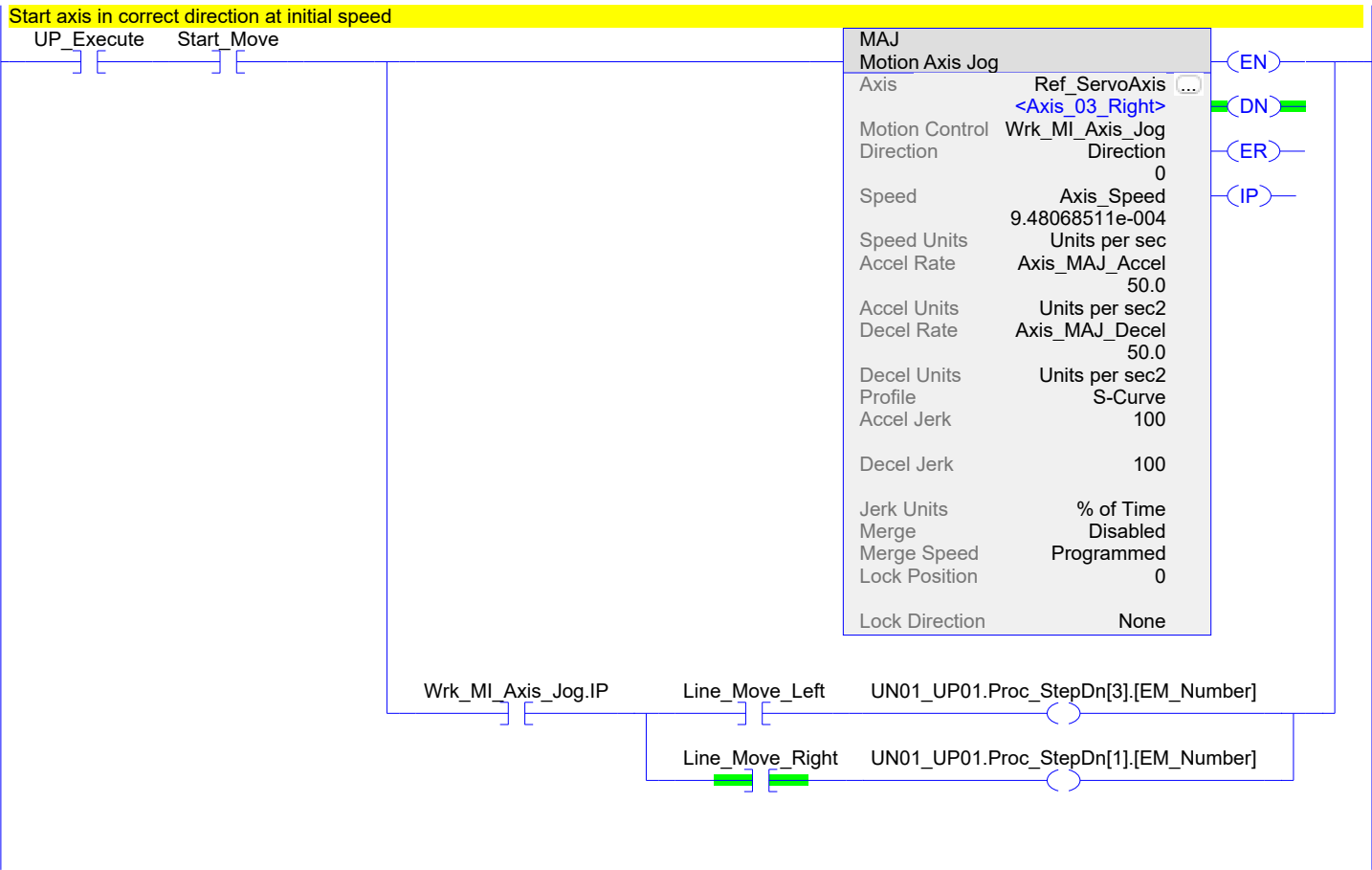




Starting step - Set direction tag for MAJ to value based on line direction
Set to be rewinder when moving right (CW) - set speed 20% faster
Set to be unwinder when moving left (CCW) - set speed 20% slower



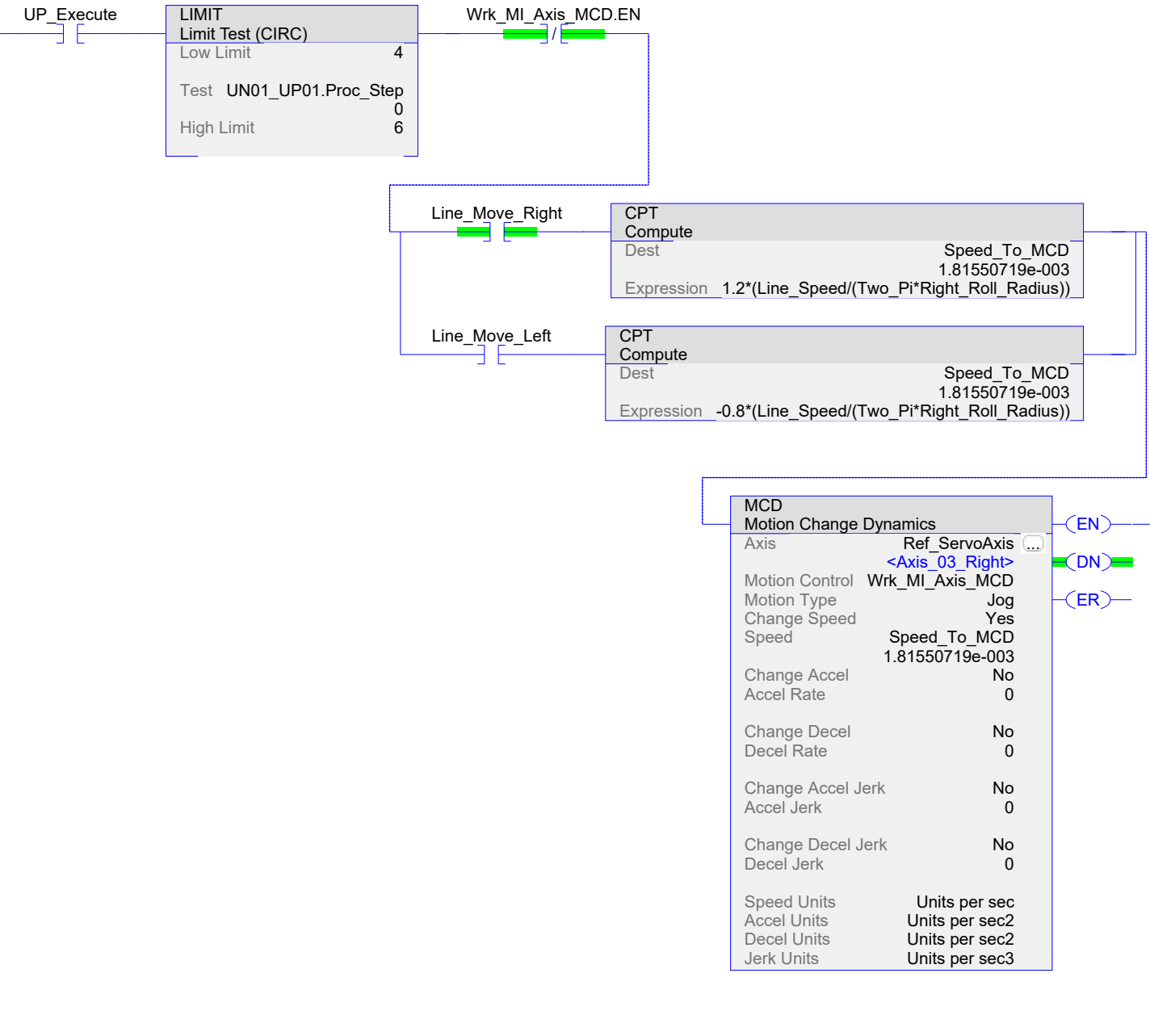
10



11

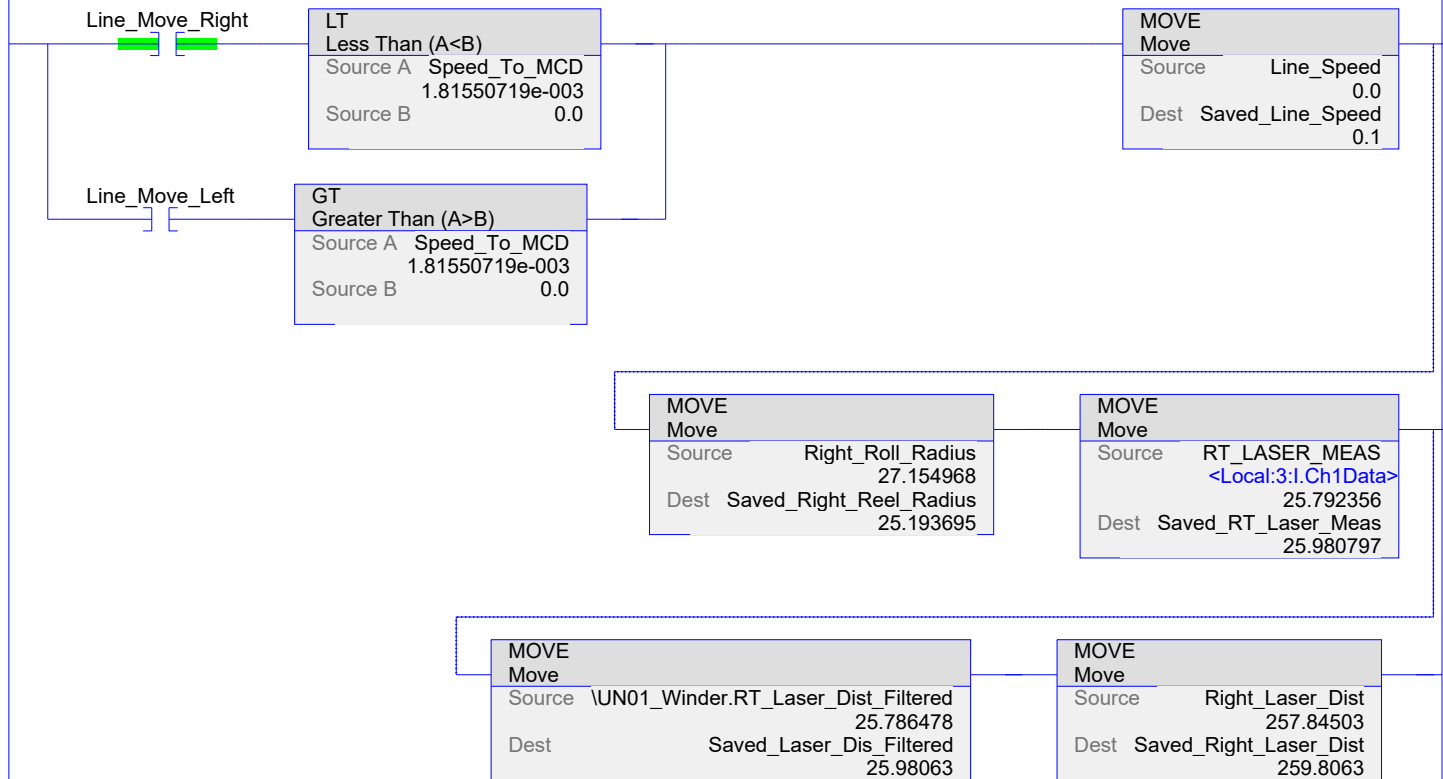
UP Procedure steps 4, 5, 6 - ramp up line speed, maintain, then decrease.

Change speed of motor to match desired line speed..



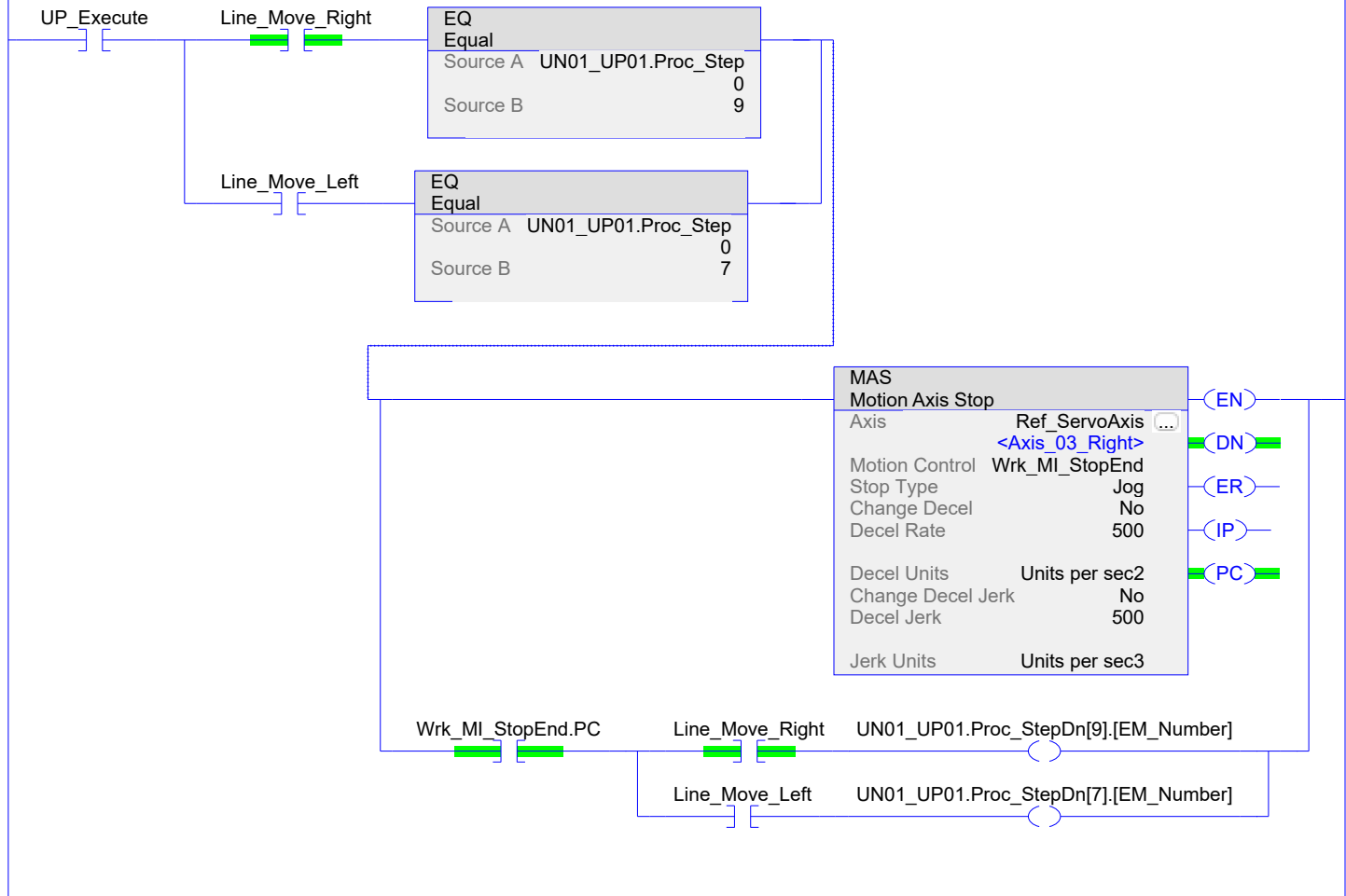
If sign of Speed_to_MCD is opposite of what it should be, trap intermediate calculation results.

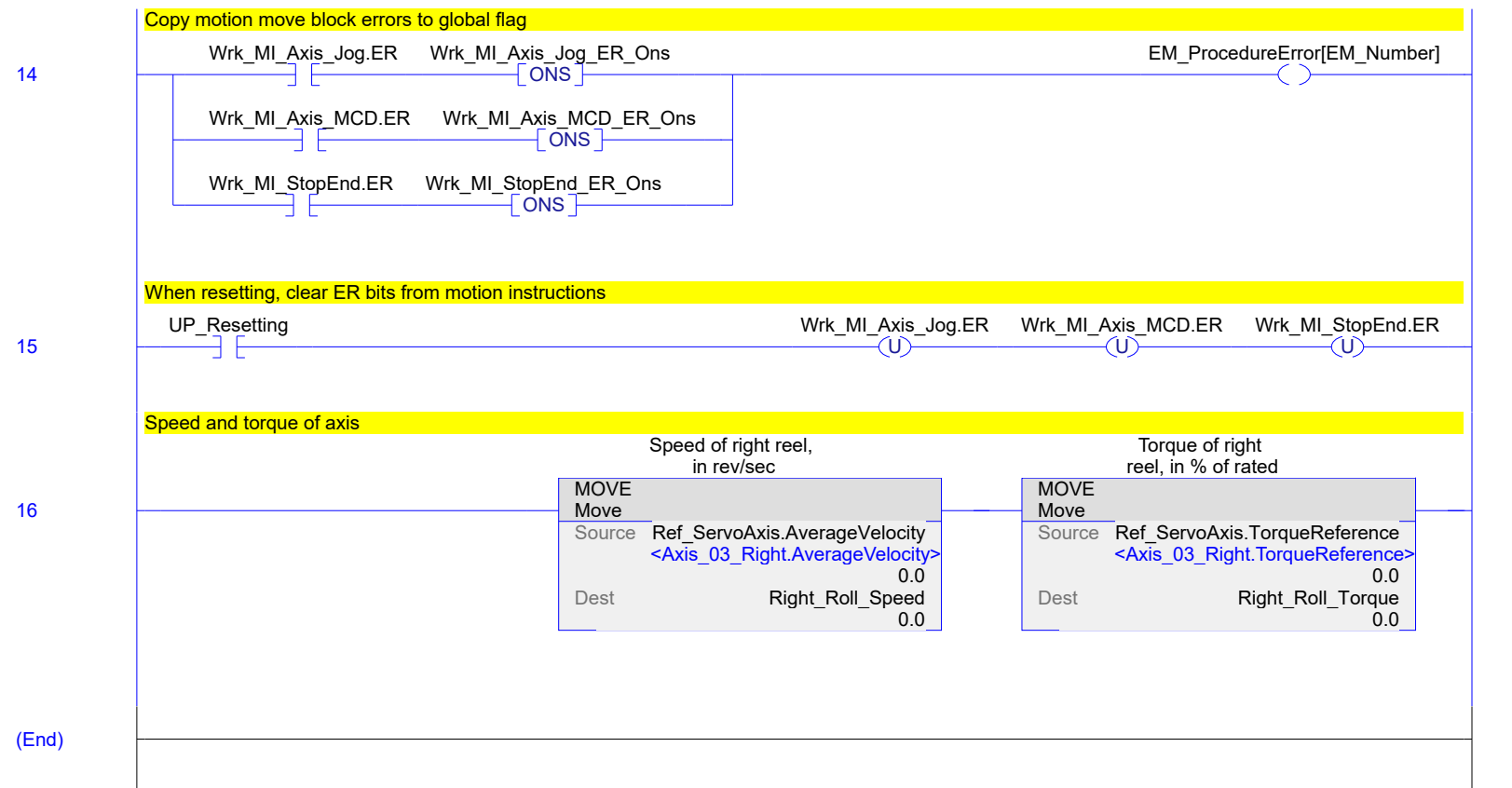
12



Stop axis

13





Axis-related blocks for PackML States

Note that EM_Number program tag contains the axis number. Index into Boolean arrays that communicate state "done" to main PackML state machine.

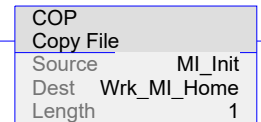
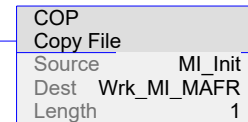
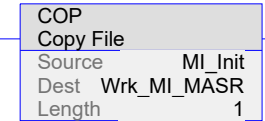
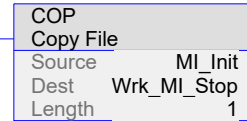
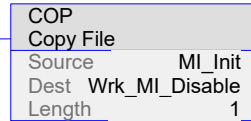
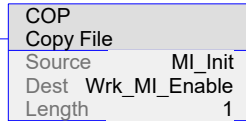
Ref_ServoAxis is aliased to the real Axis tag in the controller tags.

+++++

[NOP]

First scan - initialize all motion control tags

s:fs



STARTING -

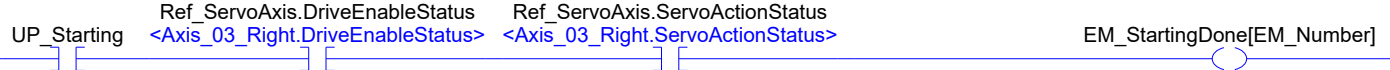
Shutdown Reset

Fault Reset

Servo On

Done when servo active and drive enabled

Skip if servo active and drive enable status



Copy any motion instruction errors or axis fault to axis error Boolean

Wrk_MI_Stop.ER Wrk_MI_Stop_ER_ONS

EM_MotionInstrError[EM_Number]

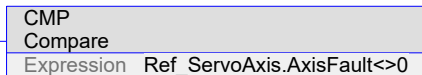
Wrk_MI_Disable.ER Wrk_MI_Disable_ER_ONS

Wrk_MI_MASR.ER Wrk_MI_MASR_ER_ONS

Wrk_MI_MAFR.ER Wrk_MI_MAFR_ER_ONS

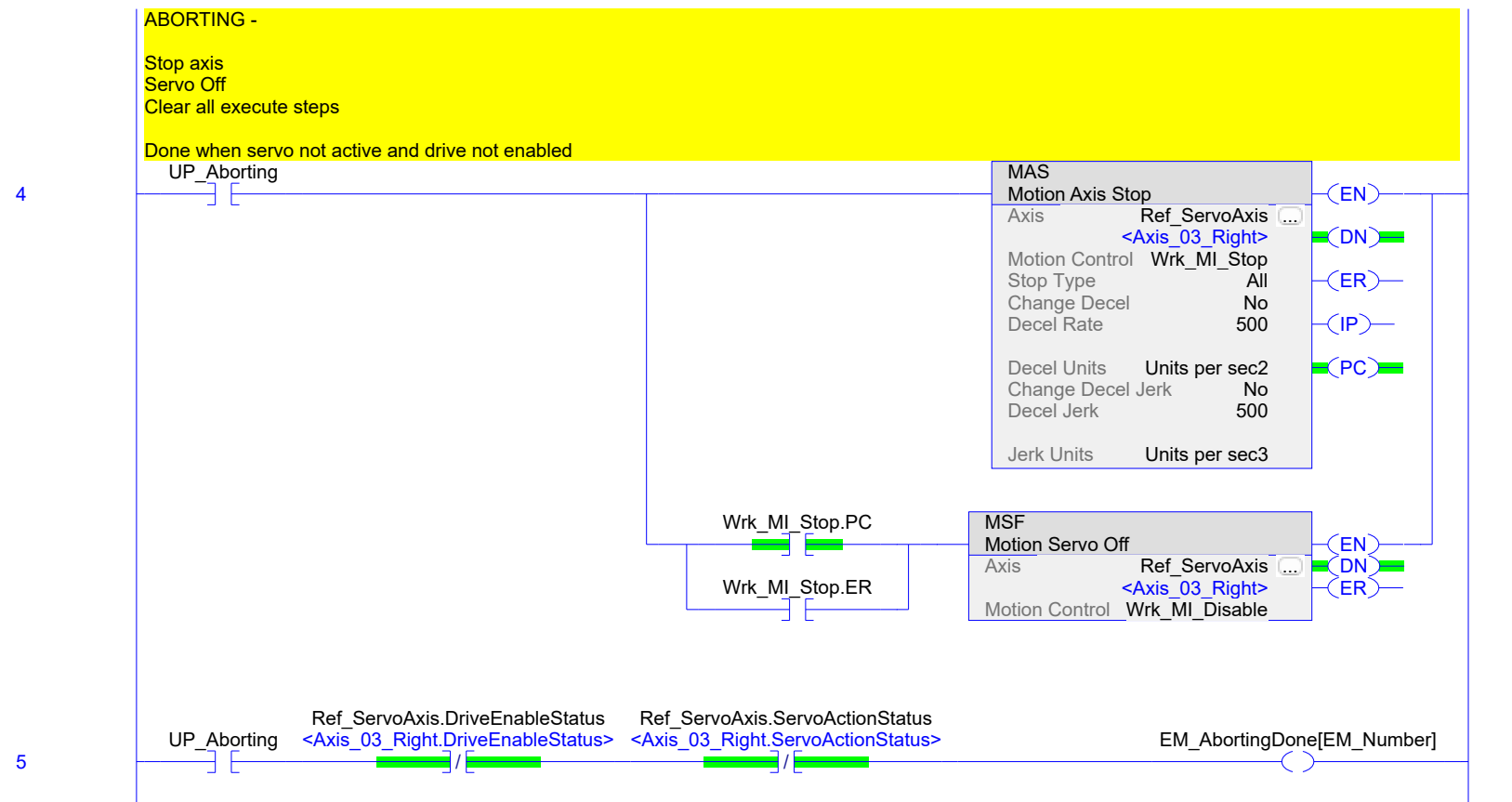
Wrk_MI_Enable.ER Wrk_MI_Enable_ER_ONS

Wrk_MI_Home.ER Wrk_MI_Home_ER_ONS



EM_ProcedureError[EM_Number]

Wrk_JogFault



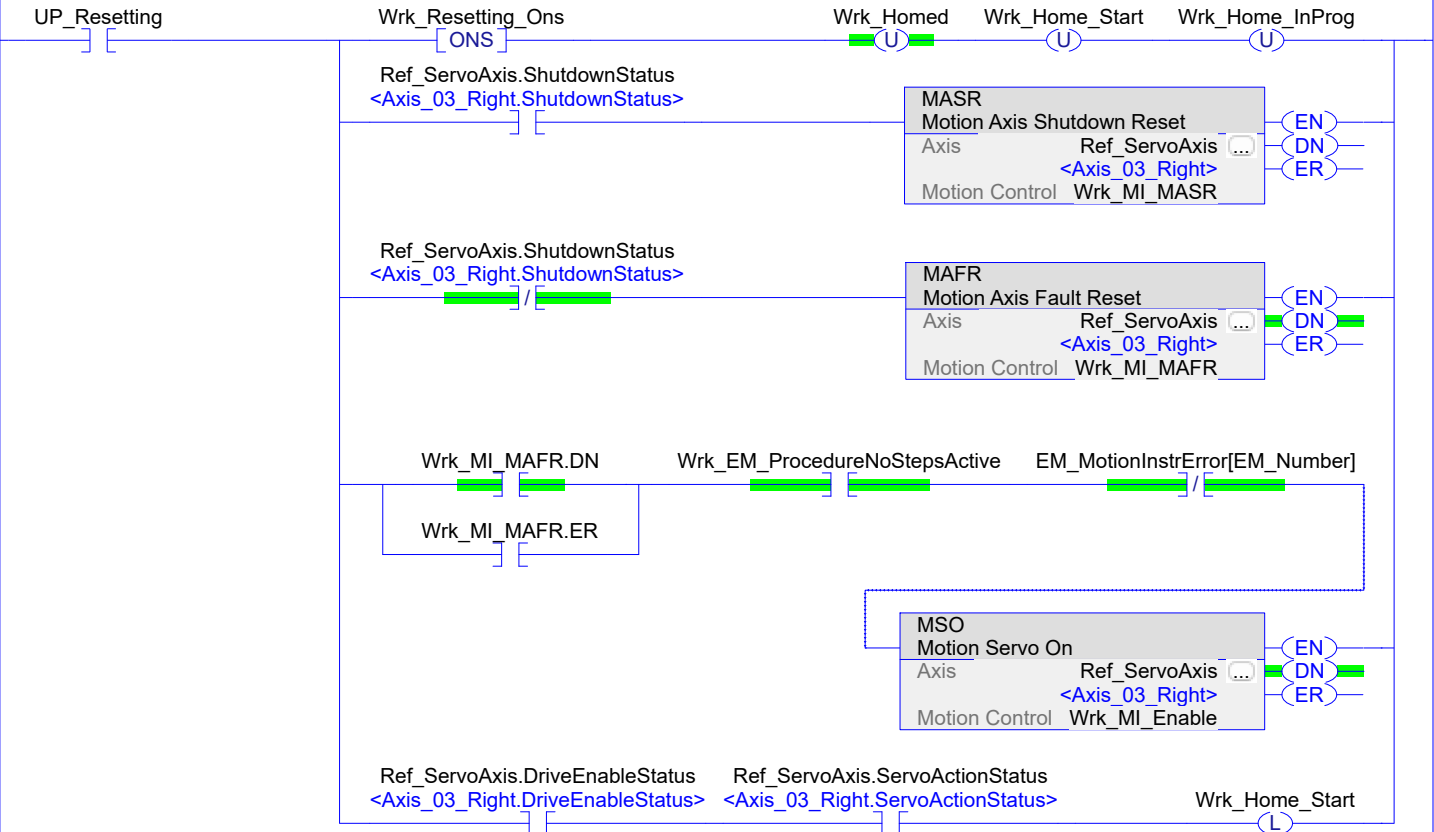
RESETTING -

Clear all faults on axis, clear all instruction ER bits.

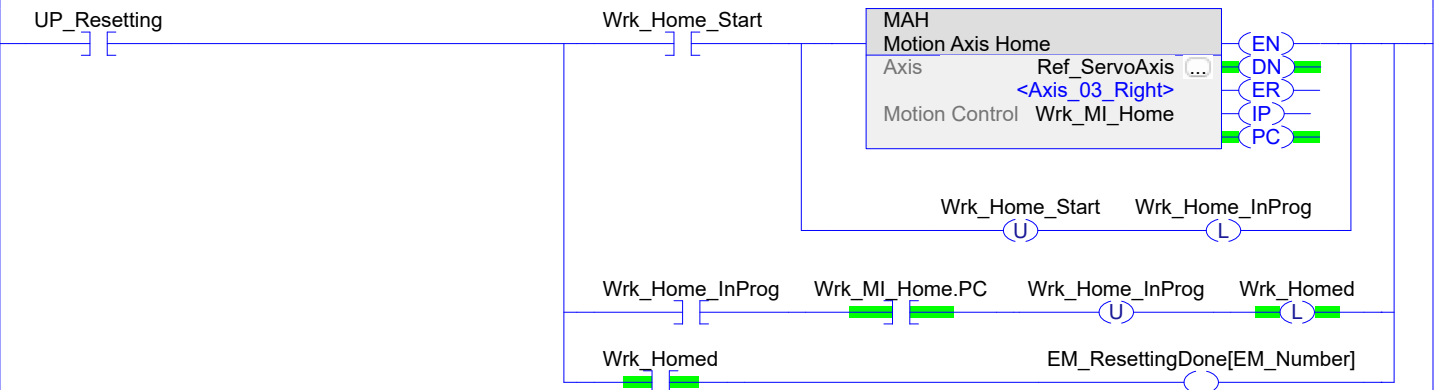
Shutdown Reset
Fault Reset
Servo On
Home Axis

Complete when servo active and drive enabled and axis homed

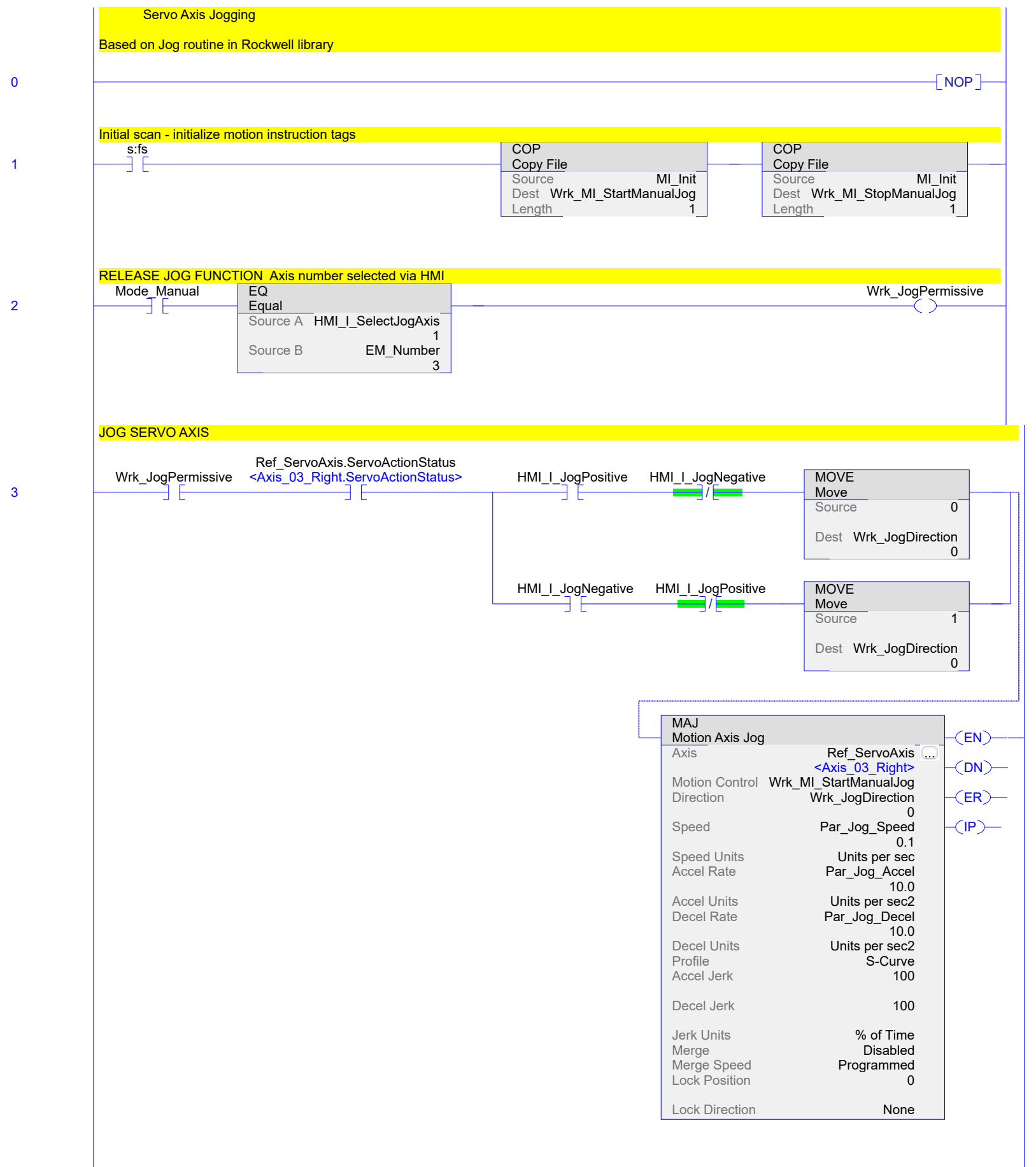
6

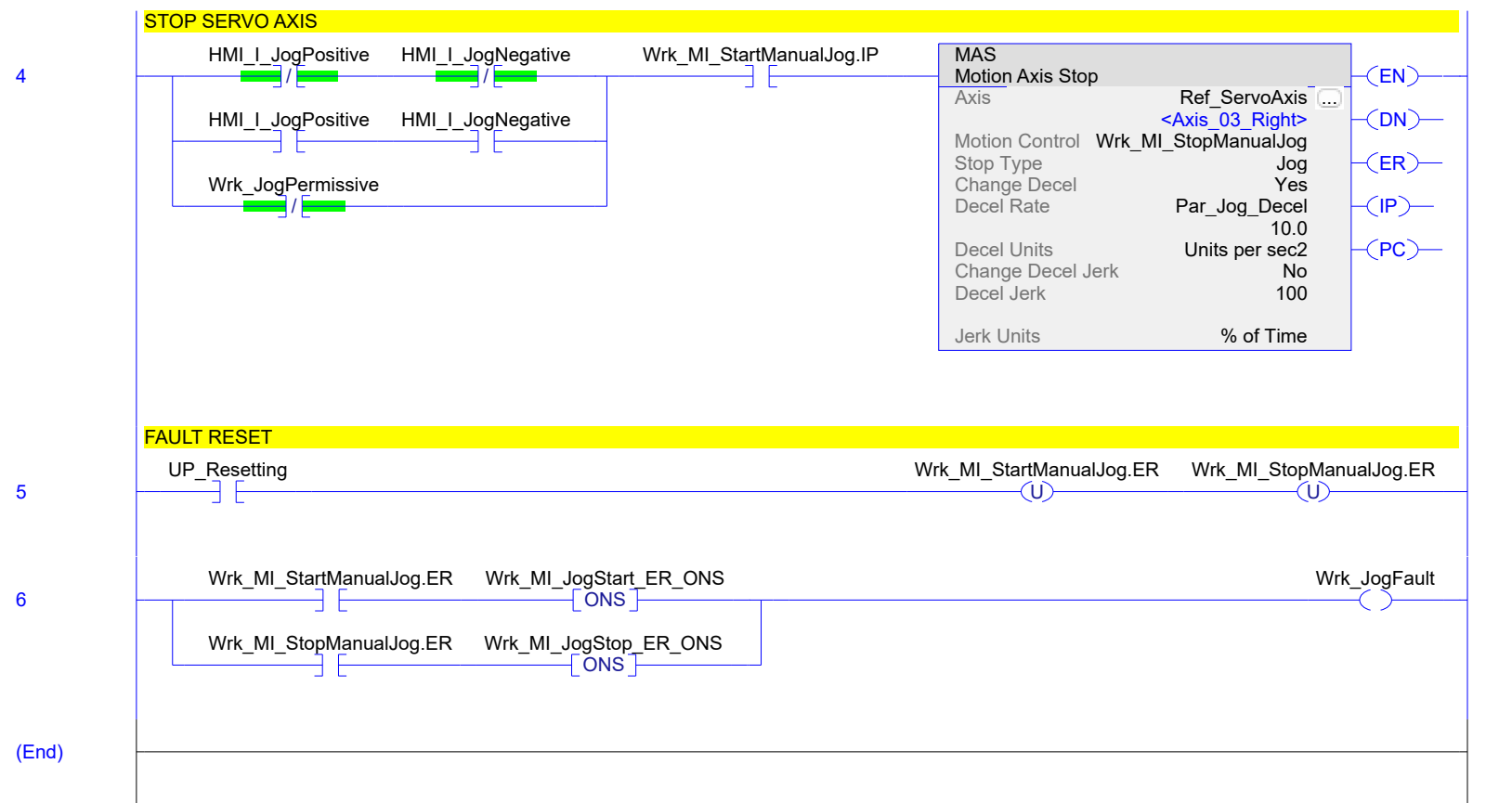


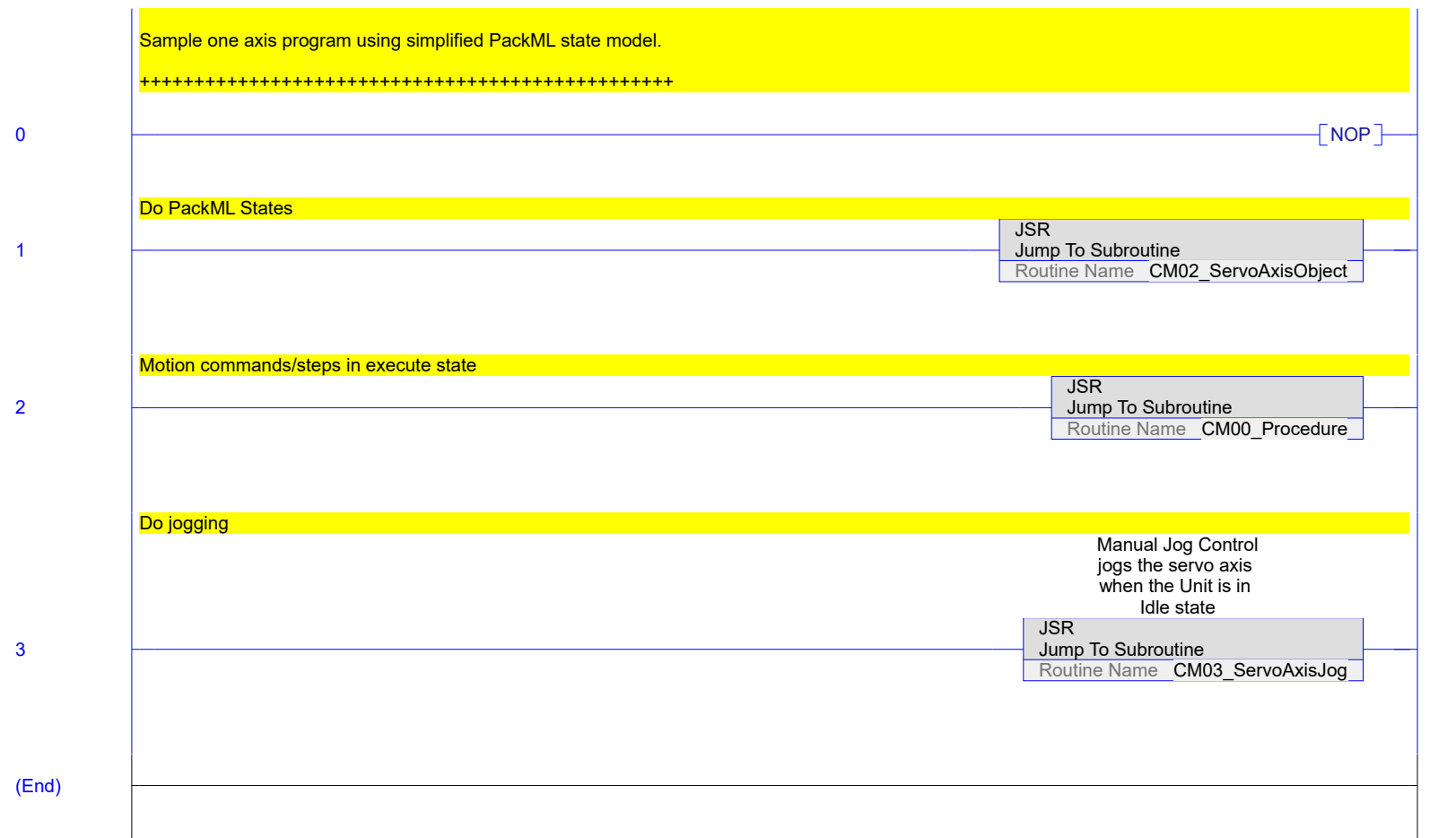
7



(End)







Winder	
Label does not exist	1
MainTask	
UN01_Winder	
MainRoutine	
Ladder Diagram	3
UP00_Procedure	
Ladder Diagram	4
UP01_PackML	
Ladder Diagram	11
UP05_Laser_Distance	
Ladder Diagram	13
UP05_Laser_Distance_Moving_Average	
Structured Text	16
EM_Axis01_Left	
CM00_Procedure	
Ladder Diagram	17
CM02_ServoAxisObject	
Ladder Diagram	24
CM03_ServoAxisJog	
Ladder Diagram	27
MainRoutine	
Ladder Diagram	29
EM_Axis02_Line	
CM00_Procedure	
Ladder Diagram	30
CM02_ServoAxisObject	
Ladder Diagram	34
CM03_ServoAxisJog	
Ladder Diagram	36
MainRoutine	
Ladder Diagram	38
EM_Axis03_Right	
CM00_Procedure	
Ladder Diagram	39
CM02_ServoAxisObject	
Ladder Diagram	46
CM03_ServoAxisJog	
Ladder Diagram	49
MainRoutine	
Ladder Diagram	51