Controller Simple_PackML_1Axis
Simplified version of PackML for one axis
Controller Fault Handler
Power-Up Handler

Tasks

MainTask
- **UN01_ExampleMachine**
  - **MainRoutine**
  - **UP01_PackML**
- **EM01_Axis01**
  - **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_1**

Ungrouped Axes

Add-On Instructions

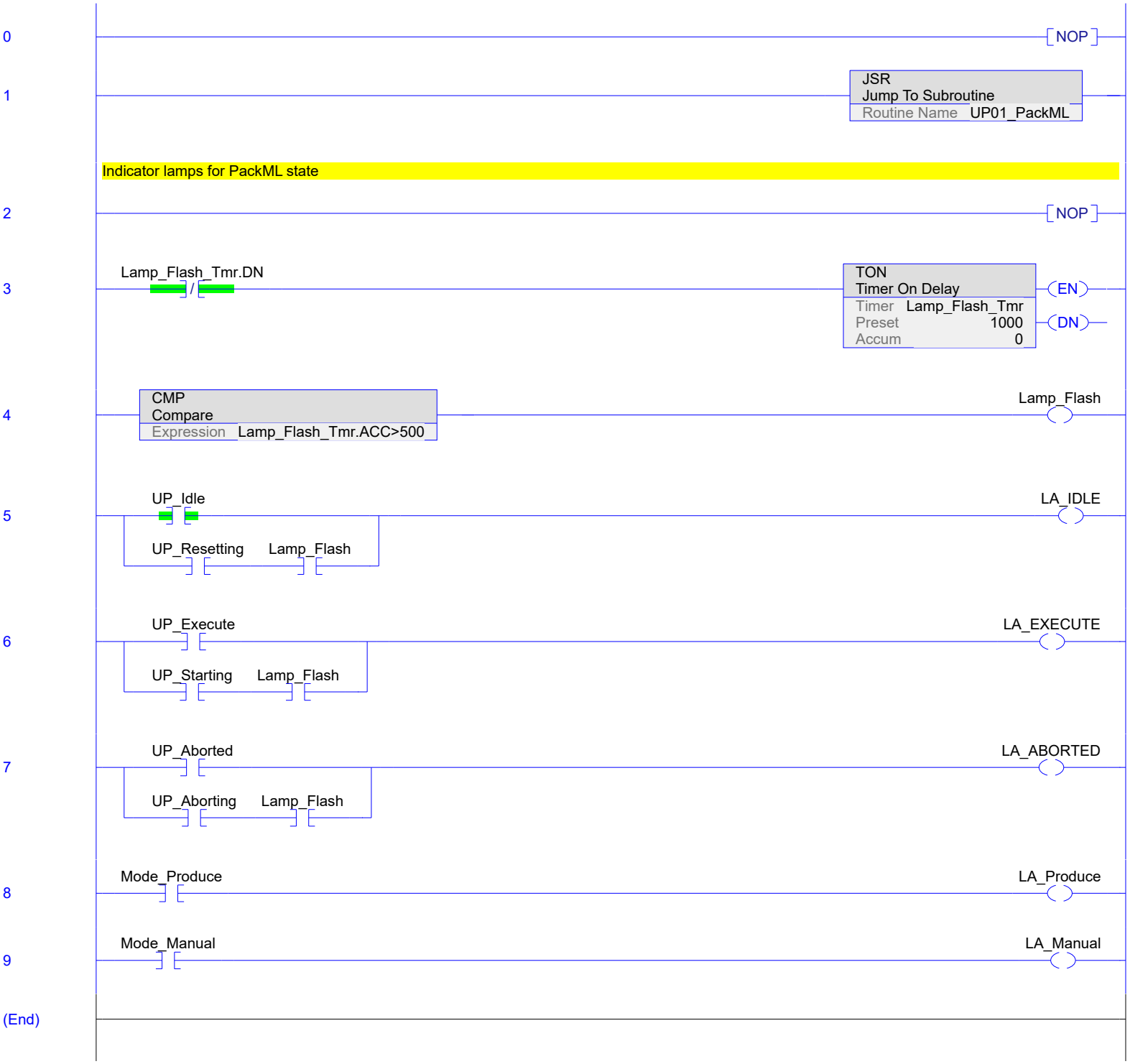
Data Types
- **User-Defined**
- **Strings**
- **Add-On-Defined**
- **Module-Defined**
  - **AB:1734_15SLOT:I:0**
  - **AB:1734_15SLOT:O:0**
  - **AB:1734_DI8:C:0**
  - **AB:1734_DOB8:C:0**
  - **AB:1734_IE8:C:0**
  - **AB:1734_IE8:I:0**
  - **AB:1734_OE4:C:0**
  - **AB:1734_OE4:I:0**
  - **AB:1734_OE4:O:0**
  - **AB:1756_DI:C:0**
  - **AB:1756_DI:C:1**
  - **AB:1756_DI:I:0**
  - **AB:1756_DO:C:0**
  - **AB:1756_DO:I:0**
  - **AB:1756_DO:O:0**
  - **AB:1769_HSC1_Range:C:0**
  - **AB:Embedded_AnalogIO1:C:0**
  - **AB:Embedded_AnalogIO1:I:0**
  - **AB:Embedded_AnalogIO1:O:0**
  - **AB:Embedded_DiscreteIO1:C:0**
  - **AB:Embedded_DiscreteIO1:I:0**
  - **AB:Embedded_DiscreteIO1:O:0**
  - **AB:Embedded_HSC1:C:0**
  - **AB:Embedded_HSC1:I:0**
  - **AB:Embedded_HSC1:O:0**
  - **AB:Embedded_HSC1_STRUCT_OUT1:O:0**
  - **AB:Motion_Diagnostics:S:1**

Trends

I/O Configuration

1756 Backplane, 1756-A10
- **[0] 1756-L71 Simple_PackML_1Axis**
- **[7] 1756-EN2TR en2tr**
  - **Ethernet**
    - **1756-EN2TR en2tr**

2198-C1004-ERS Axis_Servo



Implements simplified version of PackML States

++

0 NOP

Handle Mode

Only two possible modes - Manual and Produce

HMI_ModeChg changes mode - toggles between manual and produce. Only allowed if in aborted state.

On first scan - set to manual mode

1 s:fs — Mode_Manual (L)

2 HMI_I_ModeChgReq, UP_Aborted, ModeChgONS (ONS), Mode_Manual, Mode_Produce (/) — Mode_Manual (U)

3 HMI_I_ModeChgReq, UP_Aborted, ModeChg2ONS (ONS), Mode_Manual (/), Mode_Produce — Mode_Manual (L)

4 Mode_Manual (/) — Mode_Produce ( )

5 Mode_Manual — MOVE Move: Source 3, Dest HMI_I_ModeCmd 0

6 Mode_Produce — MOVE Move: Source 1, Dest HMI_I_ModeCmd 0

First scan - clear all states and enter aborted

7 s:fs — UP_Idle (U), UP_Starting (U), UP_Execute (U), UP_Aborting (U), UP_Aborted (L), UP_Resetting (U)

If no PackML states active, set to IDLE.

8 UP_Idle (/), UP_Starting (/), UP_Execute (/), UP_Aborting (/), UP_Aborted (/), UP_Resetting (/) — UP_Aborted (L)

IDLE -
Wait for start pushbutton press.

Next state is STARTING

9 UP_Idle, Start push button START_PB — UP_Idle (U), UP_Starting (L)

STARTING -

Exit when all axes started.

Add more done Booleans as appropriate

10 UP_Starting, EM_Starting_Done[1] — UP_Starting (U), UP_Execute (L)

Transition for EXECUTE to IDLE

Add more done Booleans as appropriate

11 UP_Execute EM_Execute_Done[1] UP_Execute(U) UP_Idle(L)

ABORTING -

From any state if stop or error in motion instruction or axis fault

Exit when all axes are off. Add more done Booleans as appropriate
Next step is ABORTED

12 Stop push button STOP_PB / Reset push button RESET_PB / UP_Idle(U) UP_Starting(U) UP_Resetting(U) UP_Aborted(U) UP_Execute(U) UP_Aborting(L)
EM_MotionInstrError[1]

13 UP_Aborting EM_Aborting_Done[1] UP_Aborting(U) UP_Aborted(L)

ABORTED -

Stay in state until reset pressed.

Next state is RESETTING

14 UP_Aborted Reset push button RESET_PB UP_Aborted(U) UP_Resetting(L)

RESETTING -

When resetting done on all axes, transition to IDLE.
Add more done Booleans as appropriate

15 UP_Resetting EM_Resetting_Done[1] UP_Resetting(U) UP_Idle(L)

(End)

EXECUTE State motion steps/commands

++++++++++++++++++++++++++++++++++

0 NOP

Initial Scan - Initialize motion control tags. Will also clear .ER bits.

1 s:fs, UP_Resetting

COP Copy File: Source MI_Init, Dest Wrk_MI_Move2, Length 1

COP Copy File: Source MI_Init, Dest Wrk_MI_Move1, Length 1

First scan or resetting - clear all steps

2 s:fs, UP_Resetting, UP_Idle → Step_1 (U), Step_2 (U), Step_3 (U), Step_4 (U), Step_5 (U), Step_6 (U), Step_7 (U), Step_8 (U), Step_9 (U), Step_10 (U)

Indication that all steps reset.
Used by RESETTING state and initial execute step

3 Step_1, Step_2, Step_3, Step_4, Step_5, Step_6, Step_7, Step_8, Step_9, Step_10 → No_Steps_Active

```
**********************************************************************************************************************
**                                                                                                                    **
**   Add your steps/blocks dealing with the axis motion here.                                                         **
**   Remember  the "Axis" tag for the motion blocks should be "Ref_ServoAxis"                                         **
**   Also, the "Motion Control" tag for the motion blocks should be a program tag, not a controller tag.              **
**                                                                                                                    **
**********************************************************************************************************************
```

4 NOP

EXECUTE -

Do two rotary profile moves

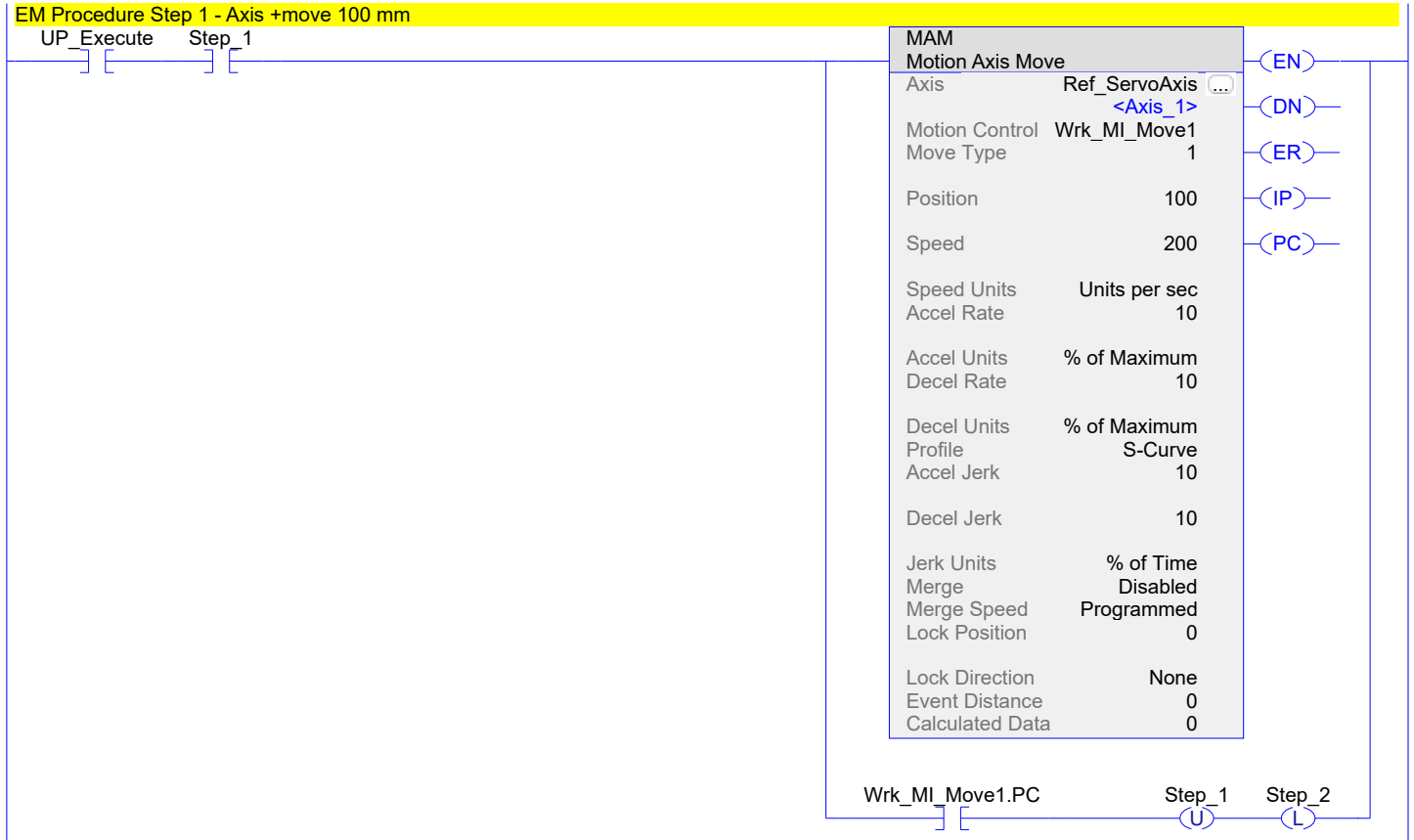
Next step - IDLE (if no errors)

If no steps active, start in step 1

5 Mode_Produce, UP_Execute, No_Steps_Active → Step_1 (L)

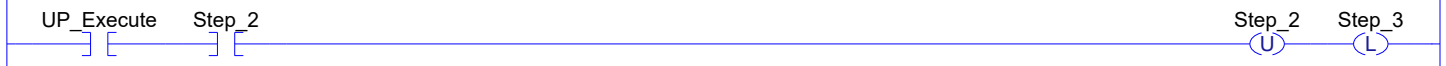
EM Procedure Step 1 - Axis +move 100 mm

6



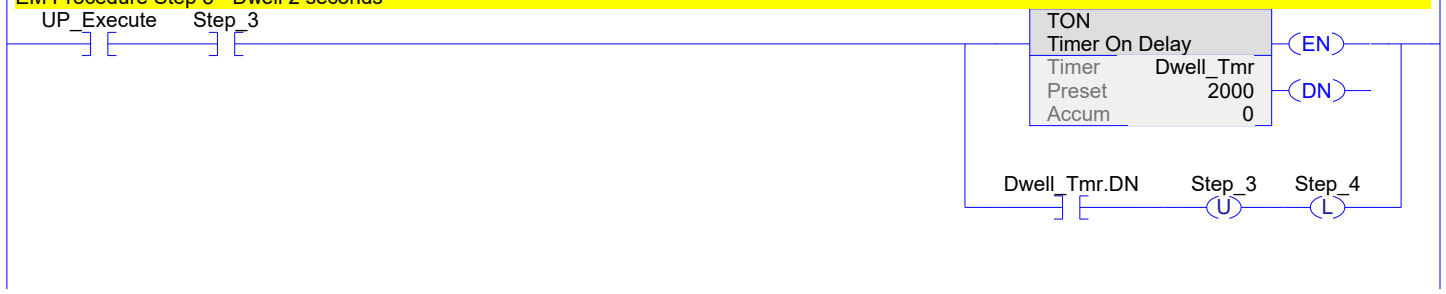
EM Procedure Step 2 - Spare step

7



EM Procedure Step 3 - Dwell 2 seconds

8



EM Procedure Step 4 - Axis -move 100 mm

9

UP_Execute Step_4

MAM
Motion Axis Move

Axis	Ref_ServoAxis <Axis_1>
Motion Control	Wrk_MI_Move2
Move Type	1
Position	-100
Speed	200
Speed Units	Units per sec
Accel Rate	10
Accel Units	% of Maximum
Decel Rate	10
Decel Units	% of Maximum
Profile	S-Curve
Accel Jerk	10
Decel Jerk	10
Jerk Units	% of Time
Merge	Disabled
Merge Speed	Programmed
Lock Position	0
Lock Direction	None
Event Distance	0
Calculated Data	0

EN DN ER IP PC

Wrk_MI_Move2.PC Step_4 (U) Step_10 (L)

Last step - turn on Execute_done Boolean for this axis.

Axis_Move1 is a sample motion instruction tag.

10

UP_Execute Step_10 EM_Execute_Done[EM_Number]

All motion block errors in this routine should be in parallel and drive connected to global flag.

11

Wrk_MI_Move1.ER

Wrk_MI_Move2.ER

Wrk_MotionStepsError

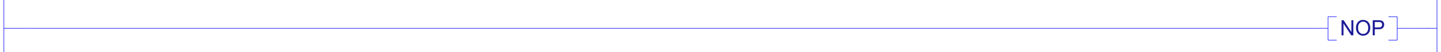
(End)

Axis-related blocks for PackML States

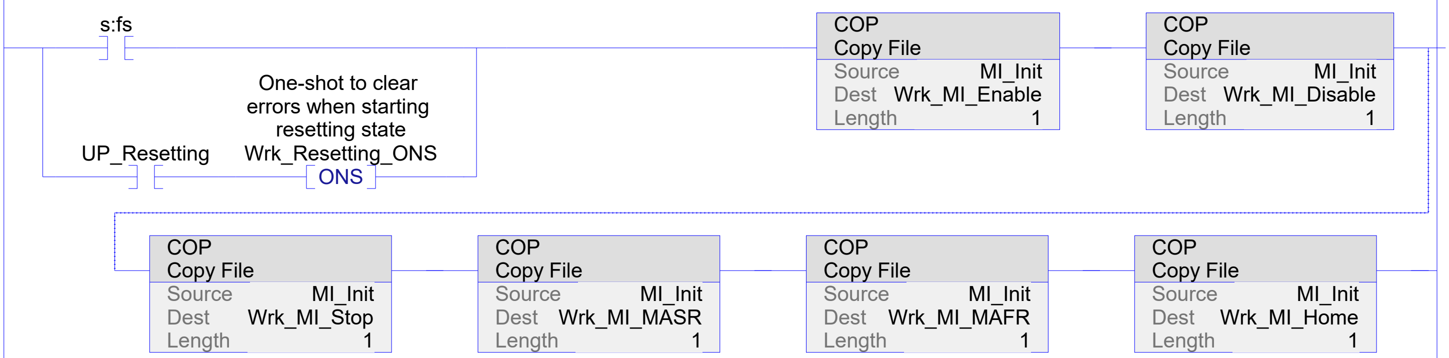
Note that EM_Number program tag contains the axis number. Index into Boolean arrays that communicate state "done" to main PackML state machine.

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

+++

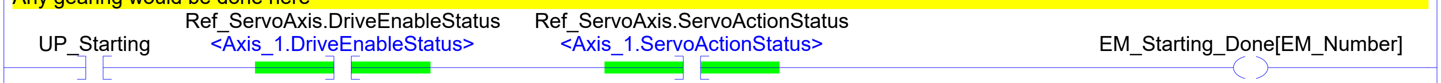
0 

First scan - initialize all motion control tags

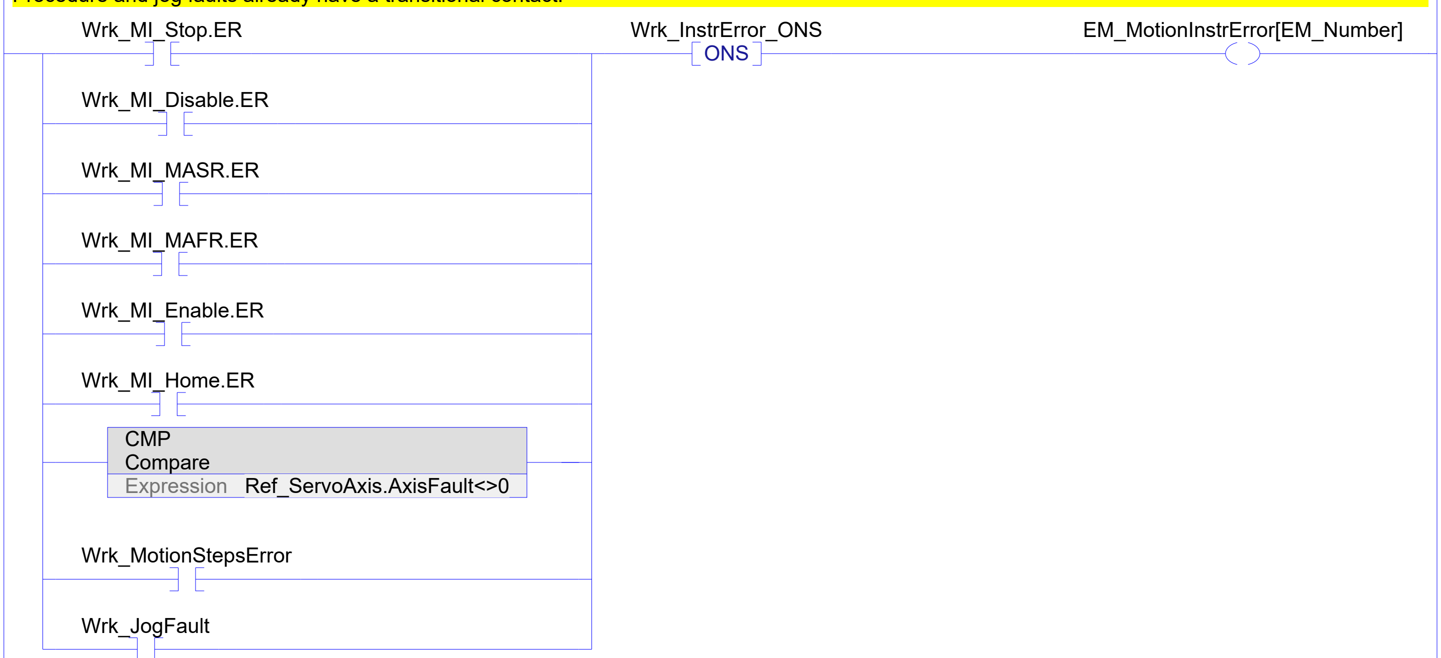
1 

STARTING -

Any gearing would be done here

2 

Copy any motion instruction errors or axis fault to axis error Boolean
Transitional contacts used do persistent error does not prevent leaving the Aborting state
Procedure and jog faults already have a transitional contact.

3 

ABORTING -

Stop axis
Servo Off
Clear all execute steps

Done when servo not active and drive not enabled

4

UP_Aborting

MAS
Motion Axis Stop
Axis Ref_ServoAxis <Axis_1>
Motion Control Wrk_MI_Stop
Stop Type All
Change Decel Yes
Decel Rate 50
Decel Units % of Maximum
Change Decel Jerk No
Decel Jerk 50
Jerk Units % of Maximum

EN DN ER IP PC

Wrk_MI_Stop.PC
Wrk_MI_Stop.ER

MSF
Motion Servo Off
Axis Ref_ServoAxis <Axis_1>
Motion Control Wrk_MI_Disable

EN DN ER

5

UP_Aborting
Ref_ServoAxis.DriveEnableStatus <Axis_1.DriveEnableStatus>
Ref_ServoAxis.ServoActionStatus <Axis_1.ServoActionStatus>
EM_Aborting_Done[EM_Number]

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

UP_Resetting
Wrk_ResetHome_ONS [ONS]
Wrk_MI_Enable.ER (U)
Wrk_Homed (U)
Wrk_Home_Start (U)
Wrk_Home_InProg (U)

Ref_ServoAxis.ShutdownStatus
<Axis_1.ShutdownStatus>

MASR
Motion Axis Shutdown Reset
Axis Ref_ServoAxis <Axis_1>
Motion Control Wrk_MI_MASR
EN DN ER

Ref_ServoAxis.ShutdownStatus
<Axis_1.ShutdownStatus> /

MAFR
Motion Axis Fault Reset
Axis Ref_ServoAxis <Axis_1>
Motion Control Wrk_MI_MAFR
EN DN ER

Wrk_MI_MAFR.DN
Wrk_MI_MAFR.ER
No_Steps_Active
EM_MotionInstrError[EM_Number] /

MSO
Motion Servo On
Axis Ref_ServoAxis <Axis_1>
Motion Control Wrk_MI_Enable
EN DN ER

Ref_ServoAxis.DriveEnableStatus
<Axis_1.DriveEnableStatus>
Ref_ServoAxis.ServoActionStatus
<Axis_1.ServoActionStatus>
Wrk_Home_Start (L)

7

UP_Resetting
Wrk_Home_Start

MAH
Motion Axis Home
Axis Ref_ServoAxis <Axis_1>
Motion Control Wrk_MI_Home
EN DN ER IP PC

Wrk_Home_Start (U)
Wrk_Home_InProg (L)

Wrk_Home_InProg
Wrk_MI_Home.PC
Wrk_Home_InProg (U)
Wrk_Homed (L)

Wrk_Homed
EM_Resetting_Done[EM_Number]

(End)

Servo Axis Jogging

Based on Jog routine in Rockwell library

0

NOP

Initial scan - initialize motion instruction tags

1

s:fs

COP
Copy File
Source MI_Init
Dest Wrk_MI_StartManualJog
Length 1

COP
Copy File
Source MI_Init
Dest Wrk_MI_StopManualJog
Length 1

RELEASE JOG FUNCTION Axis number selected via HMI

2

Mode_Manual UP_Execute

EQ
Equal
Source A HMI_I_Select_JogAxis
0
Source B EM_Number
1

Wrk_JogPermissive

JOG SERVO AXIS

3

Wrk_JogPermissive Ref_ServoAxis.ServoActionStatus <Axis_1.ServoActionStatus> HMI_I_JogPositive HMI_I_JogNegative

MOVE
Move
Source 0
Dest Wrk_JogDirection
0

HMI_I_JogNegative HMI_I_JogPositive

MOVE
Move
Source 1
Dest Wrk_JogDirection
0

MAJ
Motion Axis Jog
Axis Ref_ServoAxis <Axis_1>
Motion Control Wrk_MI_StartManualJog
Direction Wrk_JogDirection
0
Speed Par_Jog_Speed
0.0
Speed Units Units per sec
Accel Rate Par_Jog_Accel
0.0
Accel Units Units per sec2
Decel Rate Par_Jog_Decel
0.0
Decel Units Units per sec2
Profile S-Curve
Accel Jerk 100
Decel Jerk 100
Jerk Units % of Time
Merge Disabled
Merge Speed Programmed
Lock Position 0
Lock Direction None

EN DN ER IP

STOP SERVO AXIS

4

HMI_I_JogPositive HMI_I_JogNegative Wrk_MI_StartManualJog.IP

HMI_I_JogPositive HMI_I_JogNegative

Wrk_JogPermissive

MAS
Motion Axis Stop

Axis	Ref_ServoAxis <Axis_1>
Motion Control	Wrk_MI_StopManualJog
Stop Type	Jog
Change Decel	Yes
Decel Rate	Par_Jog_Decel 0.0
Decel Units	Units per sec2
Change Decel Jerk	No
Decel Jerk	100
Jerk Units	% of Time

EN DN ER IP PC

FAULT RESET

5

UP_Resetting

Wrk_MI_StartManualJog.ER (U) Wrk_MI_StopManualJog.ER (U)

6

Wrk_MI_StartManualJog.ER

Wrk_MI_StopManualJog.ER

Wrk_JogFault ( )

(End)

Sample one axis program using simplified PackML state model.

++

0 NOP

1 MOVE Move — Source 1 — Dest EM_Number 1

Do PackML States

2 JSR Jump To Subroutine — Routine Name CM02_ServoAxisObject

Motion commands/steps in execute state

3 JSR Jump To Subroutine — Routine Name CM00_Procedure

Do jogging

4 Manual Jog Control jogs the servo axis when the Unit is in Idle state
JSR Jump To Subroutine — Routine Name CM03_ServoAxisJog

(End)

Simple_PackML_1Axis
Label does not exist 1
MainTask
UN01_ExampleMachine
MainRoutine
Ladder Diagram 3
UP01_PackML
Ladder Diagram 4
EM01_Axis01
CM00_Procedure
Ladder Diagram 6
CM02_ServoAxisObject
Ladder Diagram 9
CM03_ServoAxisJog
Ladder Diagram 12
MainRoutine
Ladder Diagram 14

