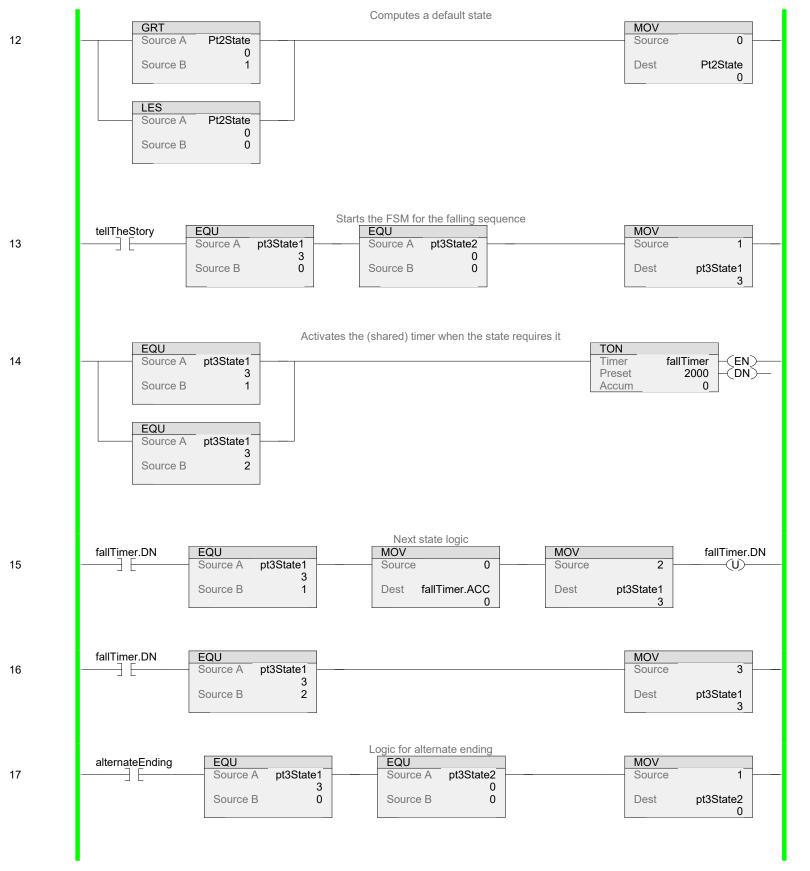# MainRoutine - Ladder Diagram

Page 1

PLC_Lab:MainTask:MainProgram

11/18/2020 2:17:09 PM

Total number of rungs in routine: 46

C:\Users\jmjohnson48\Downloads\Lab11.ACD

Enter all the logic to complete the lab below this rung. Remember to add rung comments!

0 ──[NOP]──

Transitions based off the current state

1 transition1 ─] [─ State1 ─] [─ .... State2 ─(L)─

2 transition2 ─] [─ State2 ─] [─ .... State3 ─(L)─

3 transition3 ─] [─ State3 ─] [─ .... State1 ─(L)─

uses Moore oputputs and unlateches so that the whole code is not completed with 2 states active

4 State1 ─] [─ .... Show_State1 ─( )─ State3 ─(U)─

5 State2 ─] [─ .... Show_State2 ─( )─ State1 ─(U)─

6 State3 ─] [─ .... Show_State3 ─( )─ State2 ─(U)─

7 State2 ─]/[─ State3 ─]/[─ .... State1 ─(L)─

Detects the state transition necessasary based off the current state

8 Change ─] [─

EQU	
Source A	Pt2State
	0
Source B	0

StateChanged ─]/[─ .... toggleMe ─(L)─ StateChanged ─(L)─

9 Change ─] [─

EQU	
Source A	Pt2State
	0
Source B	1

StateChanged ─]/[─ .... toggleMe ─(U)─ StateChanged ─(L)─

Changes the state under ecessasary conditions

10 StateChanged ─] [─ Change ─]/[─

EQU	
Source A	Pt2State
	0
Source B	0

MOV	
Source	1
Dest	Pt2State
	0

StateChanged ─(U)─

11 StateChanged ─] [─ Change ─]/[─

EQU	
Source A	Pt2State
	0
Source B	1

MOV	
Source	0
Dest	Pt2State
	0

StateChanged ─(U)─

Computes a default state

12
```
GRT
Source A    Pt2State
                   0
Source B           1

LES
Source A    Pt2State
                   0
Source B           0
```
```
MOV
Source             0

Dest        Pt2State
                   0
```

Starts the FSM for the falling sequence

13 tellTheStory
```
EQU
Source A    pt3State1
                   3
Source B           0
```
```
EQU
Source A    pt3State2
                   0
Source B           0
```
```
MOV
Source             1

Dest        pt3State1
                   3
```

Activates the (shared) timer when the state requires it

14
```
EQU
Source A    pt3State1
                   3
Source B           1
```
```
EQU
Source A    pt3State1
                   3
Source B           2
```
```
TON
Timer       fallTimer    (EN)
Preset      2000         (DN)
Accum          0
```

Next state logic

15 fallTimer.DN
```
EQU
Source A    pt3State1
                   3
Source B           1
```
```
MOV
Source             0

Dest     fallTimer.ACC
                   0
```
```
MOV
Source             2

Dest        pt3State1
                   3
```
fallTimer.DN
(U)

16 fallTimer.DN
```
EQU
Source A    pt3State1
                   3
Source B           2
```
```
MOV
Source             3

Dest        pt3State1
                   3
```

Logic for alternate ending

17 alternateEnding
```
EQU
Source A    pt3State1
                   3
Source B           0
```
```
EQU
Source A    pt3State2
                   0
Source B           0
```
```
MOV
Source             1

Dest        pt3State2
                   0
```

Does the (shares) timer for the states that require it

18

EQU
Source A　　pt3State2
　　　　　　　　0
Source B　　　　1

EQU
Source A　　pt3State2
　　　　　　　　0
Source B　　　　2

TON
Timer　　climbTimer　　(EN)
Preset　　　　1000　　(DN)
Accum　　　　　0

Does the next state logic

19

climbTimer.DN

EQU
Source A　　pt3State2
　　　　　　　　0
Source B　　　　1

MOV
Source　　　　　2
Dest　　pt3State2
　　　　　　　　0

MOV
Source　　　　　0
Dest　　climbTimer.ACC
　　　　　　　　0

climbTimer.DN
(U)

20

climbTimer.DN

EQU
Source A　　pt3State2
　　　　　　　　0
Source B　　　　2

MOV
Source　　　　　3
Dest　　pt3State2
　　　　　　　　0

Creates moore output based off the state

21

EQU
Source A　　pt3State1
　　　　　　　　3
Source B　　　　0

EQU
Source A　　pt3State2
　　　　　　　　0
Source B　　　　0

showNeutralSherlock
( )

22

EQU
Source A　　pt3State1
　　　　　　　　3
Source B　　　　1

showFall1
( )

23

EQU
Source A　　pt3State1
　　　　　　　　3
Source B　　　　2

showFall2
( )

24

EQU
Source A　　pt3State2
　　　　　　　　0
Source B　　　　1

showClimb1
( )

25

EQU
Source A　　pt3State2
　　　　　　　　0
Source B　　　　2

showClimb2
( )

Creates default state value

26

```
LES
Source A    pt3State1
                   3
Source B           0
```

```
GRT
Source A    pt3State1
                   3
Source B           3
```

```
LES
Source A    pt3State2
                   0
Source B           0
```

```
GRT
Source A    pt3State2
                   0
Source B           3
```

```
MOV
Source            0
Dest     pt3State1
                 3
```

```
MOV
Source            0
Dest     pt3State2
                 0
```

Resets the state machine and logic for initial state of both sequences

27

```
EQU
Source A    pt3State1
                   3
Source B           0
```

```
EQU
Source A    pt3State2
                   0
Source B           0
```

showNeutralSherlock
( )

28   reset
—] [—

```
MOV
Source            0
Dest     fallTimer.ACC
                 0
```

```
MOV
Source         2000
Dest     fallTimer.PRE
              2000
```

```
MOV
Source            0
Dest     climbTimer.ACC
                 0
```

```
MOV
Source         1000
Dest     climbTimer.PRE
              1000
```

```
MOV
Source            0
Dest     pt3State1
                 3
```

```
MOV
Source            0
Dest     pt3State2
                 0
```

Stoplight FSM
1 = Greem
2 = Yellow
3 = Red
Next state logic below

29   EWSensor
—] [—

```
EQU
Source A          NS
                   1
Source B           1
```

```
TON
Timer       SensorT      (EN)
Preset         1000      (DN)
Accum             0
```

NSSensor
—] [—

```
EQU
Source A          EW
                   3
Source B           1
```

30

SensorT.DN
┤ ├

EQU
Source A   NS
                1
Source B   1

MOV
Source   0
Dest   RedWaitT.ACC
                0

MOV
Source   0
Dest   SensorT.ACC
                0

MOV
Source   2
Dest   NS
                1

RedWaitT.DN
—(U)—

SensorT.DN
—(U)—

31

EQU
Source A   NS
                1
Source B   2

EQU
Source A   EW
                3
Source B   2

TON
Timer   YellowT   —(EN)—
Preset   2000   —(DN)—
Accum   0

32

YellowT.DN
┤ ├

EQU
Source A   NS
                1
Source B   2

MOV
Source   3
Dest   NS
                1

MOV
Source   0
Dest   RedWaitT.ACC
                0

MOV
Source   0
Dest   YellowT.ACC
                0

RedWaitT.DN
—(U)—

YellowT.DN
—(U)—

33

EQU
Source A   EW
                3
Source B   3

EQU
Source A   NS
                1
Source B   3

TON
Timer   RedWaitT   —(EN)—
Preset   2000   —(DN)—
Accum   0

34

SensorT.DN
┤ ├

EQU
Source A   EW
                3
Source B   1

MOV
Source   0
Dest   RedWaitT.ACC
                0

MOV
Source   0
Dest   SensorT.ACC
                0

MOV
Source   2
Dest   EW
                3

RedWaitT.DN
—(U)—

SensorT.DN
—(U)—

35

YellowT.DN
┤ ├

EQU
Source A   EW
                3
Source B   2

MOV
Source   3
Dest   EW
                3

MOV
Source   0
Dest   YellowT.ACC
                0

YellowT.DN
—(U)—

Does error detection

36
```
GRT
Source A      NS
              1
Source B      3

LES
Source A      NS
              1
Source B      1
```
```
MOV
Source        1

Dest          NS
              1
```

37
```
GRT
Source A      EW
              3
Source B      3

LES
Source A      EW
              3
Source B      1
```
```
MOV
Source        3

Dest          EW
              3
```

Does moore output based on the state

38
```
EQU
Source A      EW
              3
Source B      1
```
EWGreen
( )

39
```
EQU
Source A      EW
              3
Source B      2
```
ToEW   EWYellow
(U)   ( )

40
```
EQU
Source A      EW
              3
Source B      3
```
EWRed
( )

41
```
EQU
Source A      NS
              1
Source B      1
```
NSGreen
( )

42
```
EQU
Source A      NS
              1
Source B      2
```
ToEW   NSYellow
(L)   ( )

43
```
EQU
Source A      NS
              1
Source B      3
```
NSRed
( )

Does logic once the stop light is double red to see whuch state is next

44

ToEW	RedWaitT.DN	EQU		EQU		MOV	
		Source A	NS 1	Source A	EW 3	Source	1
		Source B	3	Source B	3	Dest	EW 3

45

ToEW	RedWaitT.DN	EQU		EQU		MOV	
		Source A	NS 1	Source A	EW 3	Source	1
		Source B	3	Source B	3	Dest	NS 1

(End)