











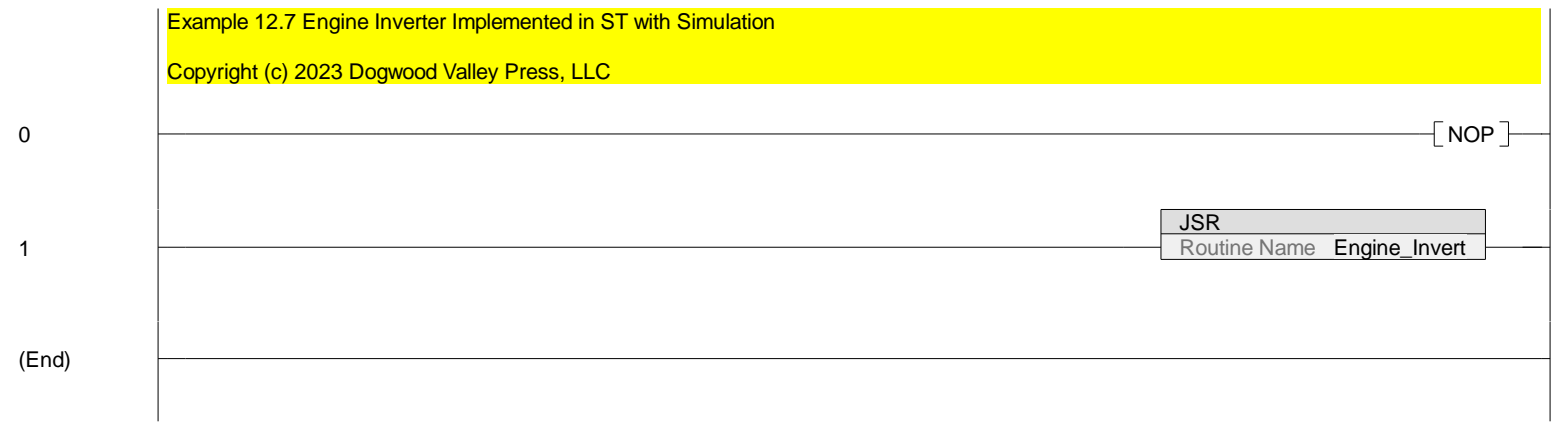
 **Controller Example_12_7** **Controller Fault Handler** **Power-Up Handler****Tasks** **MainTask** **MainProgram** **MainRoutine** **Engine_Invert** **Simulation** **Simulation** **Unscheduled****Motion Groups** **Ungrouped Axes****Add-On Instructions****Data Types** **User-Defined** **Seq_Type**

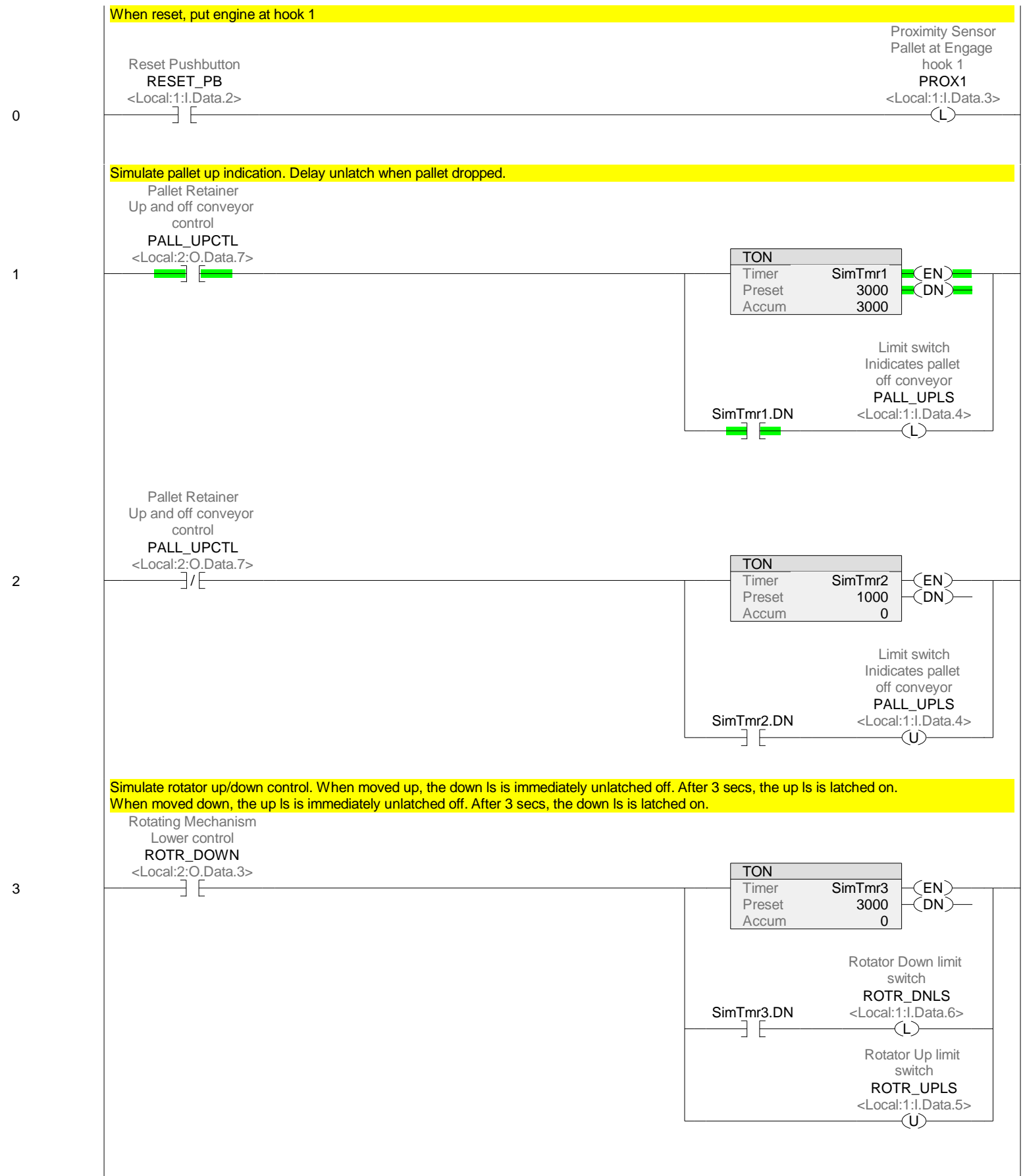
Counter-Based Sequencer

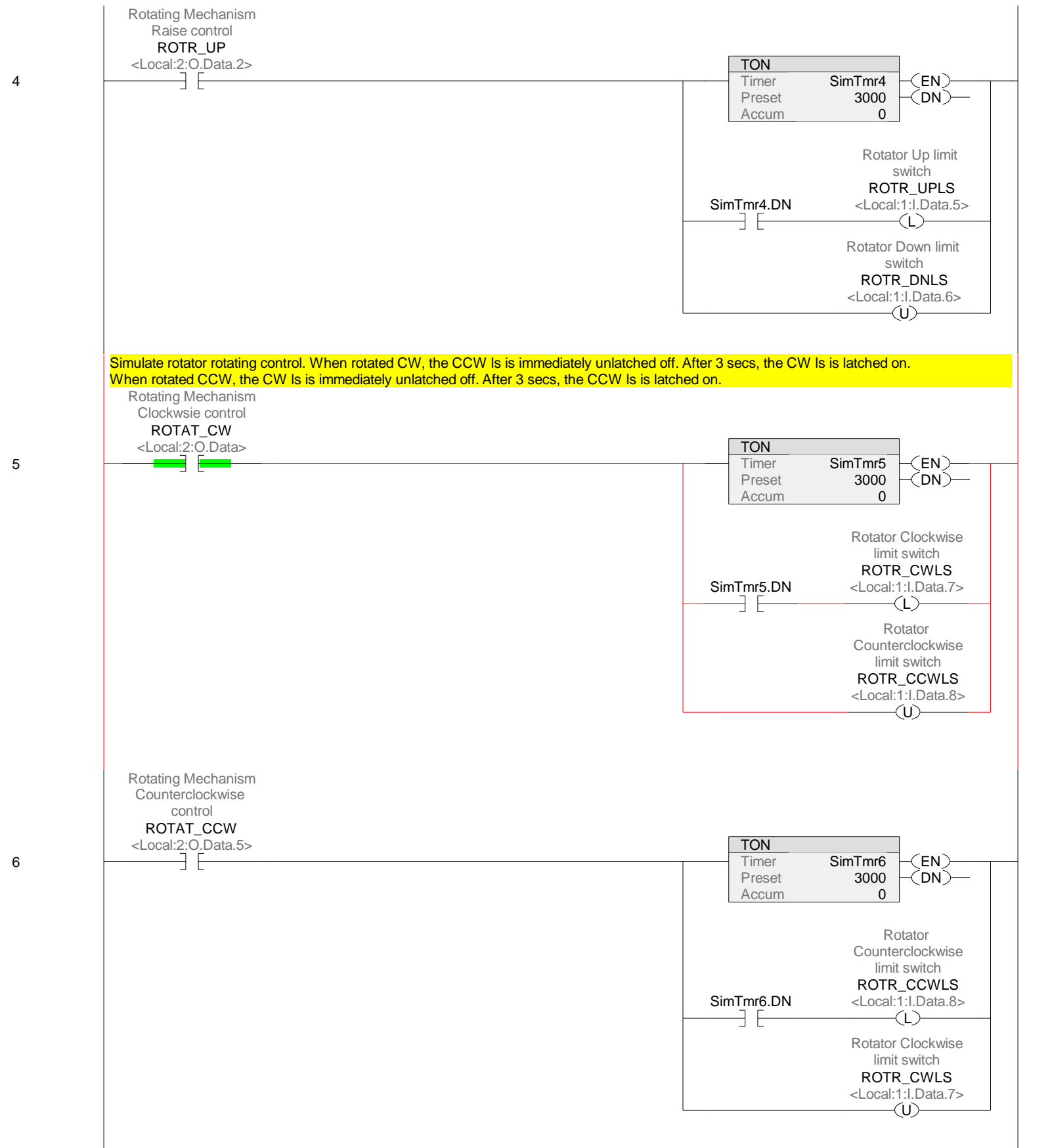
 **Strings** **Add-On-Defined** **Module-Defined** **AB:1756_DI:C:0** **AB:1756_DI:I:0** **AB:1756_DO:C:0** **AB:1756_DO:I:0** **AB:1756_DO:O:0****Trends****I/O Configuration** **1756 Backplane, 1756-A10** **[0] 1756-L71 Example_12_7** **[1] 1756-IB32/A Dig_In** **[2] 1756-OB16I Dig_Out**

```
1 // Engine Inverter in ST
2 //
3 // Start/stop of operation
4 IF (START_PB and NOT Int_Reset) THEN Run := 1; END_IF;
5 IF (Run AND NOT STOP_PB) THEN Run := 0; END_IF;
6 //
7 // Normal operation transition out of initial step
8 IF Run and (IStep = 0 ) THEN IStep := 1; END_IF;
9 //
10 // Transitions for normal operation
11 CASE IStep of
12     1: IF PROX1 and Run THEN IStep := 2; END_IF;
13     2: Eng1_Tmr.TimerEnable := 1;
14         Eng1_Tmr.PRE := 2000;
15         TONR(Eng1_Tmr);
16         IF Eng1_Tmr.DN THEN Eng1_Tmr.TimerEnable := 0; TONR(Eng1_Tmr); IStep := 3; END_IF;
17     3: IF PALL_UPLS and Run THEN IStep := 4; END_IF;
18     4: IF ROTR_DNLS and Run THEN IStep := 5; END_IF;
19     5: Clmp_Tmr.TimerEnable := 1;
20         Clmp_Tmr.PRE := 1500;
21         TONR(Clmp_Tmr);
22         IF Clmp_Tmr.DN THEN Clmp_Tmr.TimerEnable := 0; TONR(Clmp_Tmr); IStep := 6; END_IF;
23     6: IF ROTR_UPLS and Run THEN IStep := 7; END_IF;
24     7: IF ROTR_CWLS and Run THEN IStep := 8; END_IF;
25     8: IF ROTR_DNLS and Run THEN IStep := 9; END_IF;
26     9: UnClmp_Tmr.TimerEnable := 1;
27         UnClmp_Tmr.PRE := 1000;
28         TONR(UnClmp_Tmr);
29         IF UnClmp_Tmr.DN THEN UnClmp_Tmr.TimerEnable := 0; TONR(UnClmp_Tmr); IStep := 10; END_IF;
30     10: IF ROTR_UPLS and Run THEN IStep := 11; END_IF;
31     11: IF ROTR_CCWLS and Run THEN IStep := 12; END_IF;
32     12: IF (NOT PALL_UPLS) and Run THEN IStep := 13; END_IF;
33     13: Eng2_Tmr.TimerEnable := 1;
34         Eng2_Tmr.PRE := 3000;
35         TONR(Eng2_Tmr);
36         IF Eng2_Tmr.DN THEN Eng2_Tmr.TimerEnable := 0; TONR(Eng2_Tmr); IStep := 1; END_IF;
37 ELSE
38     IStep := 0;
39 END_CASE;
40 //
41 // Start/stop for reset operation
42 IF (RESET_PB and NOT Run) THEN Int_Reset := 1; END_IF;
43 IF (Int_Reset and RStep = 4) THEN Int_Reset := 0; END_IF;
44 //
45 // Reset initial transition and clear normal step number
46 IF Int_Reset and RStep = 0 THEN RStep := 1; IStep := 0; END_IF;
47 //
48 // Reset operation transitions
49 CASE RStep of
50     1: RUnClmp_Tmr.TimerEnable := 1;
51         RUnClmp_Tmr.PRE := 1000;
52         TONR(RUnClmp_Tmr);
53         IF RUnClmp_Tmr.DN THEN RUnClmp_Tmr.TimerEnable := 0; TONR(RUnClmp_Tmr); RStep := 2; END_IF;
54     2: IF ROTR_UPLS THEN RStep := 3; END_IF;
55     3: IF ROTR_CCWLS THEN RStep := 4; END_IF;
56     4: IF (NOT Int_Reset) THEN RStep := 0; END_IF;
57 ELSE
58     RStep := 0;
59 END_CASE;
```

```
60 //
61 // Control of outputs;
62 IF (IStep = 2) THEN ENG1_RET := 1; ELSE ENG1_RET := 0; END_IF;
63 IF (IStep = 13) THEN ENG2_RET := 1; ELSE ENG2_RET := 0; END_IF;
64 IF ((IStep = 6) OR (IStep = 10) AND Run) OR (RStep = 2) THEN ROTR_UP := 1; ELSE ROTR_UP := 0; END_IF;
65 IF ((IStep = 4) OR (IStep = 8) AND Run) THEN ROTR_DOWN := 1; ELSE ROTR_DOWN := 0; END_IF;
66 IF ((IStep = 7) AND Run) THEN ROTAT_CW := 1; ELSE ROTAT_CW := 0; END_IF;
67 IF ((IStep = 11) AND Run) OR (RStep = 3) THEN ROTAT_CCW := 1; ELSE ROTAT_CCW := 0; END_IF;
68 IF (IStep >= 5) AND (IStep <=8) THEN GRIP_CLOS := 1; ELSE GRIP_CLOS := 0; END_IF;
69 IF (IStep >= 3) AND (IStep <=11) THEN PALL_UPCTL := 1; ELSE PALL_UPCTL := 0; END_IF;
70
71
72
```

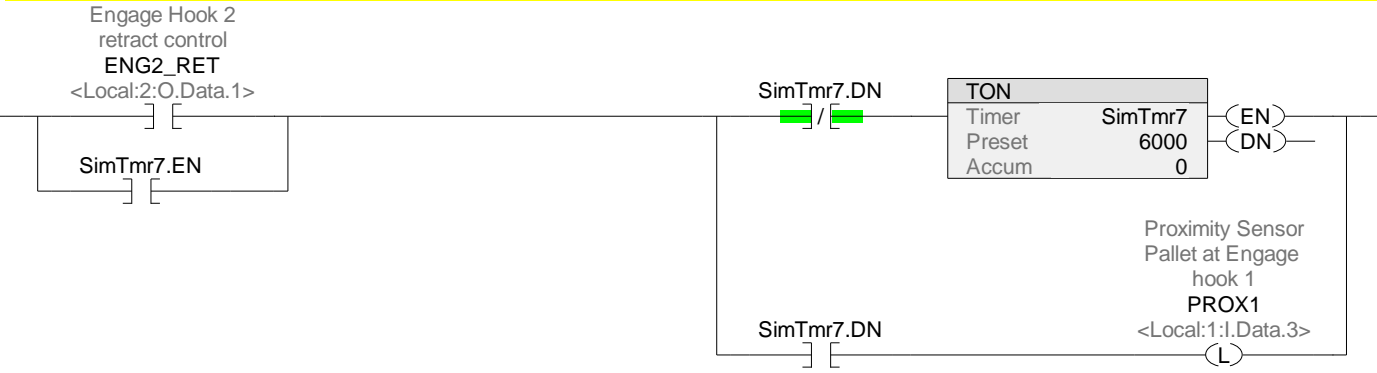






Simulate Pallet Prox
Latch it on 6 seconds after one has left the station.
Latch it off 3 second after new one retained.

7



8



(End)

Example_12_7	
Label does not exist	1
MainTask	
MainProgram	
Engine_Invert	
Structured Text	2
MainRoutine	
Ladder Diagram	4
Simulation	
Simulation	
Ladder Diagram	5