Program File List

Name	Number	Type	Rungs	Debug	Bytes
[SYSTEM]	0	SYS	0	No	0
	1	SYS	0	No	0
CONTINUOUS	2	LADDER	10	No	352
2FAILSPLUS	252	LADDER	2	No	46
2FAILS5120	253	LADDER	2	No	74
2FAILS2560	254	LADDER	2	No	74
SYNMPULSES	255	LADDER	6	No	97

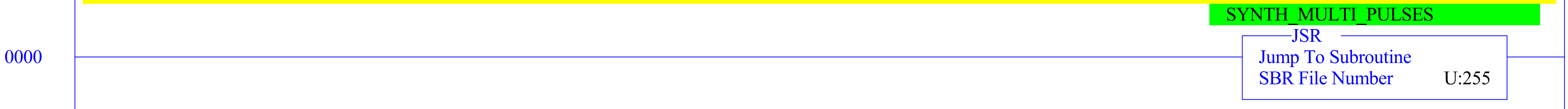
Data File List

Name	Number	Type	Scope	Debug	Words	Elements	Last
OUTPUT	0	O	Global	No	12	4	O:3
INPUT	1	I	Global	No	18	6	I:5
STATUS	2	S	Global	No	0	66	S:65
BINARY	3	B	Global	No	1	1	B3:0
TIMER	4	T	Global	No	12	4	T4:3
COUNTER	5	C	Global	No	3	1	C5:0
CONTROL	6	R	Global	No	3	1	R6:0
INTEGER	7	N	Global	No	1	1	N7:0
FLOAT	8	F	Global	No	2	1	F8:0
RJCTTIMERS	9	L	Global	No	2	1	L9:0
MISC_BITS	13	B	Global	No	6	6	B13:5
TMRBSYFLGS	103	B	Global	No	17	17	B103:16
RJCTTIMERS	104	T	Global	No	339	113	T104:112

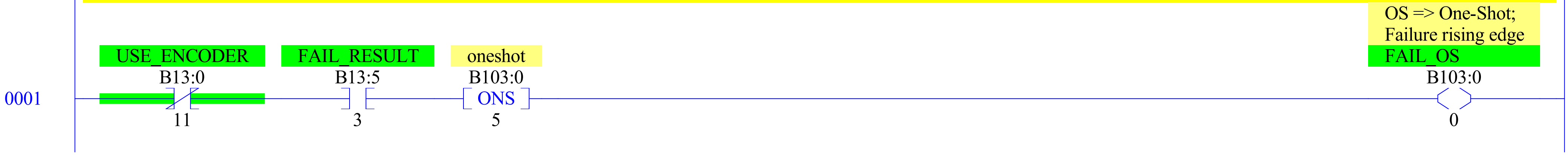
Model various multiple failed can timings; show their effect on the reject logic

1) Cans on a conveyor pass by an inspection station and continue on to a reject station
1.1) Cans that pass inspection continue on the conveyor past the reject station
1.2) Cans that fail inspection should be removed the conveyor by the pusher at the reject station
1.3) Camera(s) at the inspection station issue a pass or fail result for each can at time that can is at the inspection station
1.3.1) A fail result is modeled/emulated here via the [Stretched fail pulse] timer object in the routine LAD 255 SYNMPULSES
1.3.2) A pass result is not modeled in this test program program
2) On any scan cycle that detects the rising edge of the fail result,
2.1) Select the first of the three reject timer objects in Data File T104 that is not already selected by a previous fail event
2.1.1) Reject timer object REJECTn_TIMER is not selected when corresponding bit REJECTn is 0
2.1.2) Reject timer object REJECTn_TIMER is selected when corresponding bit REJECTn is 1
2.2) Start that reject timer object timing for 5120ms (5.12s) in a TON (Timer ON-delay) instruction
3) The reject timer object's increasing accumulation of time models the motion of the failed can
from the inspection station, along the conveyor, to the reject station
4) When a reject timer object expires (T104:x/DN bit will be 1),
4.1) The failed can, which selected that reject timer object, is assumed, in the model, to be
in front of the reject pusher solenoid.
4.2) Set the reject bit (TIME2REJECT) to 1 to trigger the solenoid and reject the failed can
3.2) Deselect that reject timer object (reset its reject bit to 0)
==

Rung 0000 calls the routine LAD 255 SYNMPULSES, which emulates particular timings
of failed cans detected by the camera(s) at the inspection station; refer to the comments in
routine SYNMPULSES for more detail.



Detect the rising edge (start) of a new failed can event



Multiple Overlapping Rejection Timers

Rungs 0002 and 0003 control the first reject timer object, REJECT1_TIMER T104:0

On any scan cycle when the failed can one-shot indicates the rising edge (beginning) of a failed can event

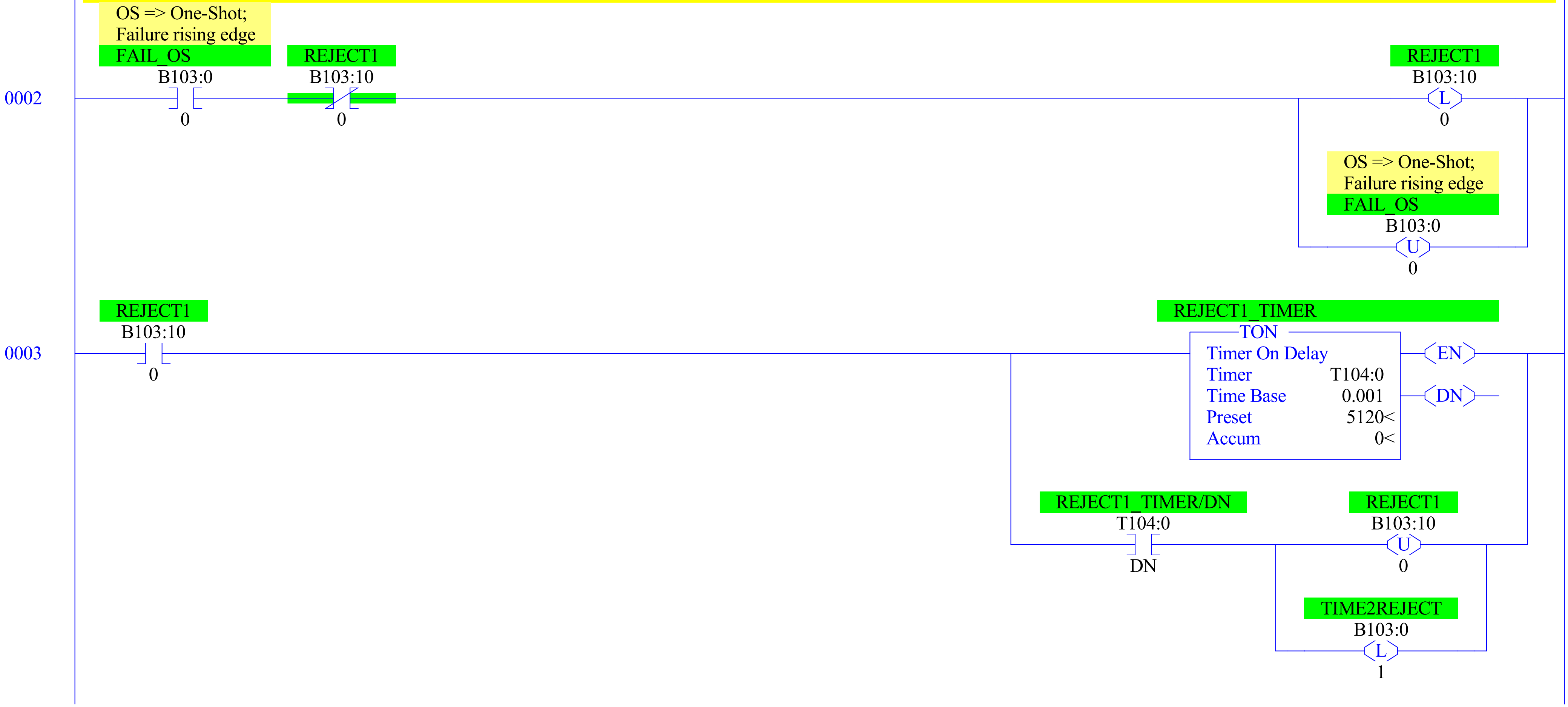
Rung 0002

- If the first reject timer object is not already selected, then
    - select it, and
    - unlatch the failed can one-shot so no other reject timer objects will be selected below
- Otherwise do nothing as it is already selected and timing a previous failed can event

Rung 0004

- If the first reject timer object has been selected, then start and continue its timing for 5120ms*
- When the first timer object expires, deselect it

* N.B. the timer object preset of 5.12s was chosen to make it easier to observe
the behavior of reject logic in this demonstration program, and is, and
is not meant to represent an actual conveyor line.



Rungs 0004 and 0005 control the second reject timer object, REJECT2_TIMER T104:1

N.B. These rungs are similar to those for the previous reject timer object. The primary difference is that the selection is dependent on the previous reject timer object being already selected (see the second instruction on Rung 0004, i.e. XIC REJECT1) which

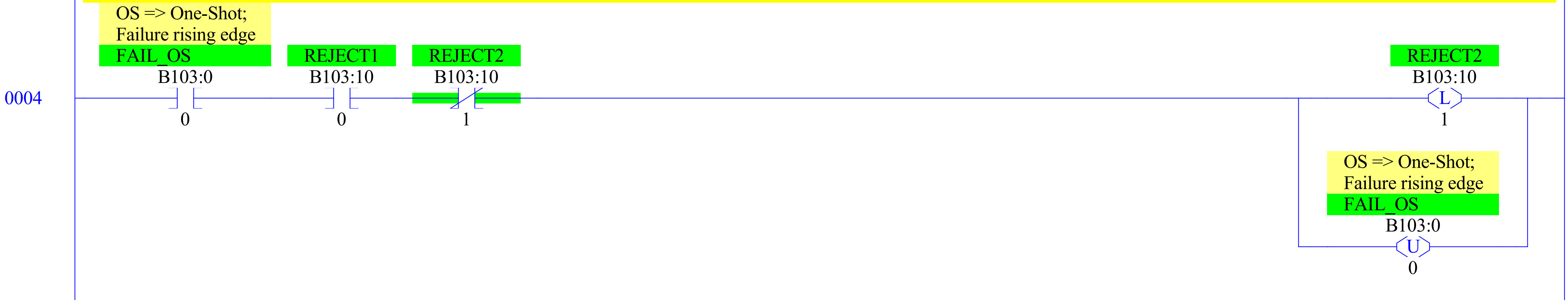
1) is unnecessary because if the previous reject timer object was not selected before, and became selected by Rung 0002 on, the current scan cycle, then Rung 0002 would have also unlatched the failed can one-shot, which will prevent Rung 0004 from selecting the second timer object for the same failed can,

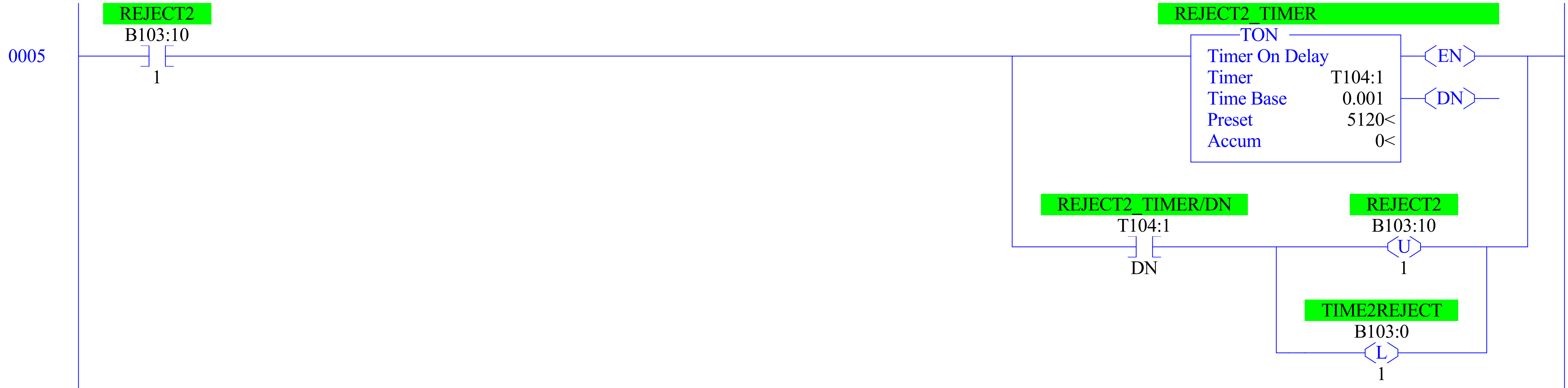
2) is also a design error, because if the first timer object above expired on the current scan cycle, then the first reject timer object selection bit (REJECT1)

2.1) would have been latched when evaluating Rung 0002, which would prevent the first timer object from being selected by the current failed can event, and

2.2) would have been unlatched after evaluating Rung 0003, because the first timer object expired,

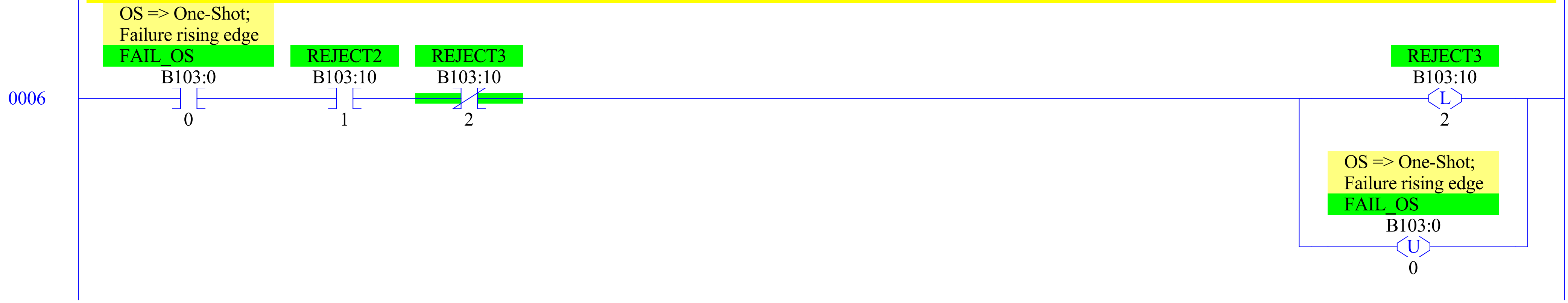
2.3) and finally, on this Rung 0004, that unlatched state of that first reject timer object selection bit would prevent selection of the second reject timer object, even though that second reject timer object was not selected.

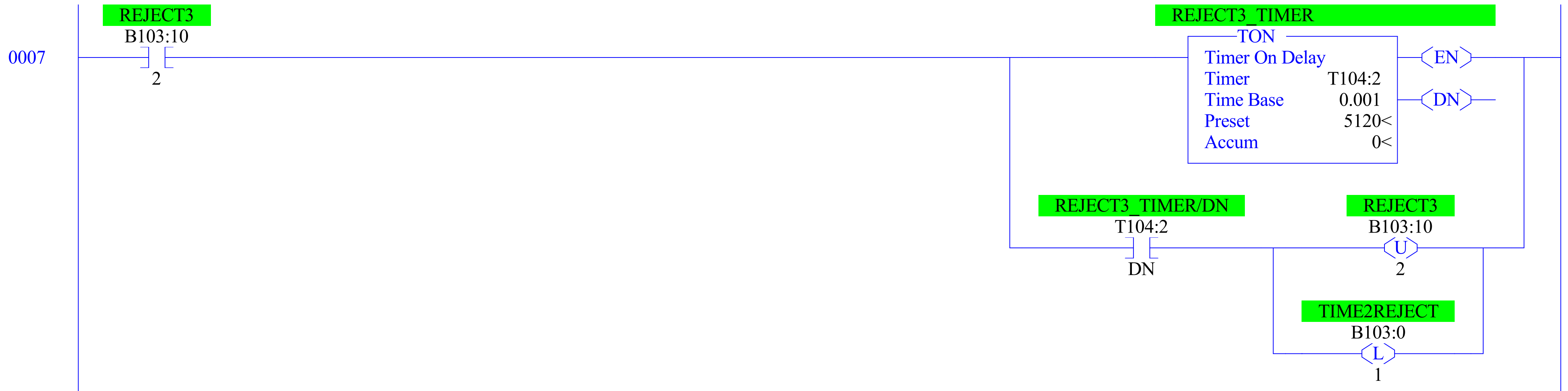




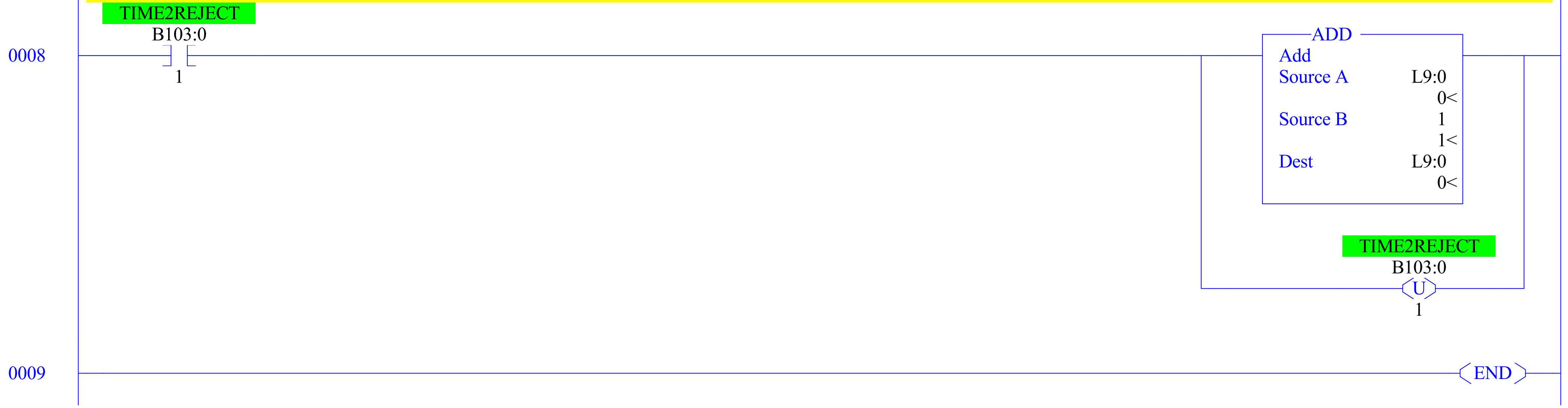
Rungs 0006 and 0007 control the second reject timer object, REJECT2_TIMER T104:1

N.B. There rungs are essentially identical to those for the previous timer object; refer to the N.B. for Rung 0003 for details.



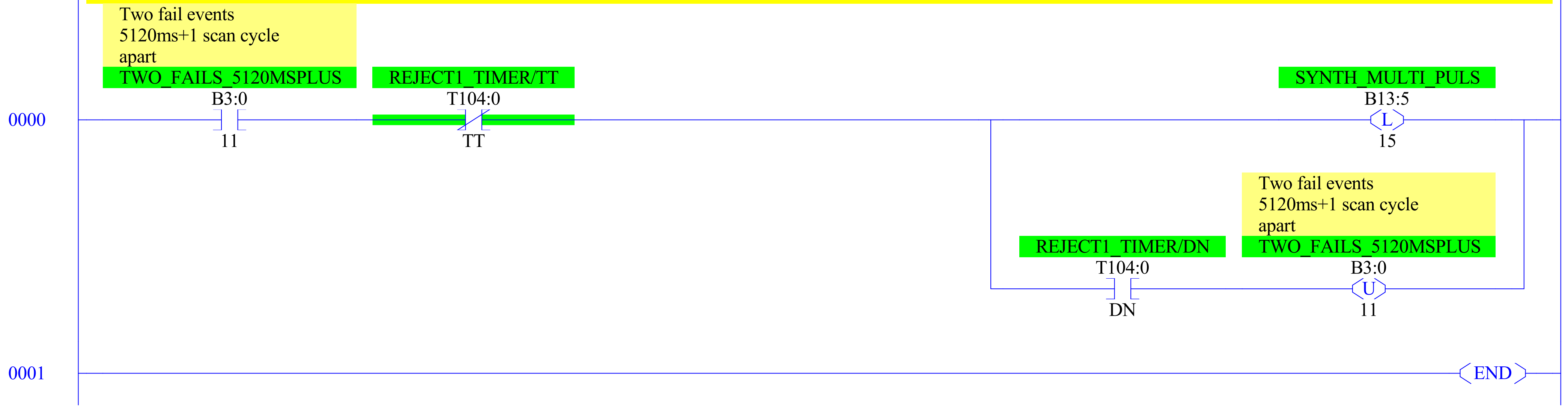


Count, and then unlatch, each reject event i.e. when the logic models a failed can arriving at the reject station



Toggling bit TWO_FAILS_5120MS_PLUS B3:0/11 will trigger two failed can events 5.12s, plus one scan cycle, apart, which means the second failed can event will occur on the scan cycle immediately after the reject timer object will expired from the first failed can event.

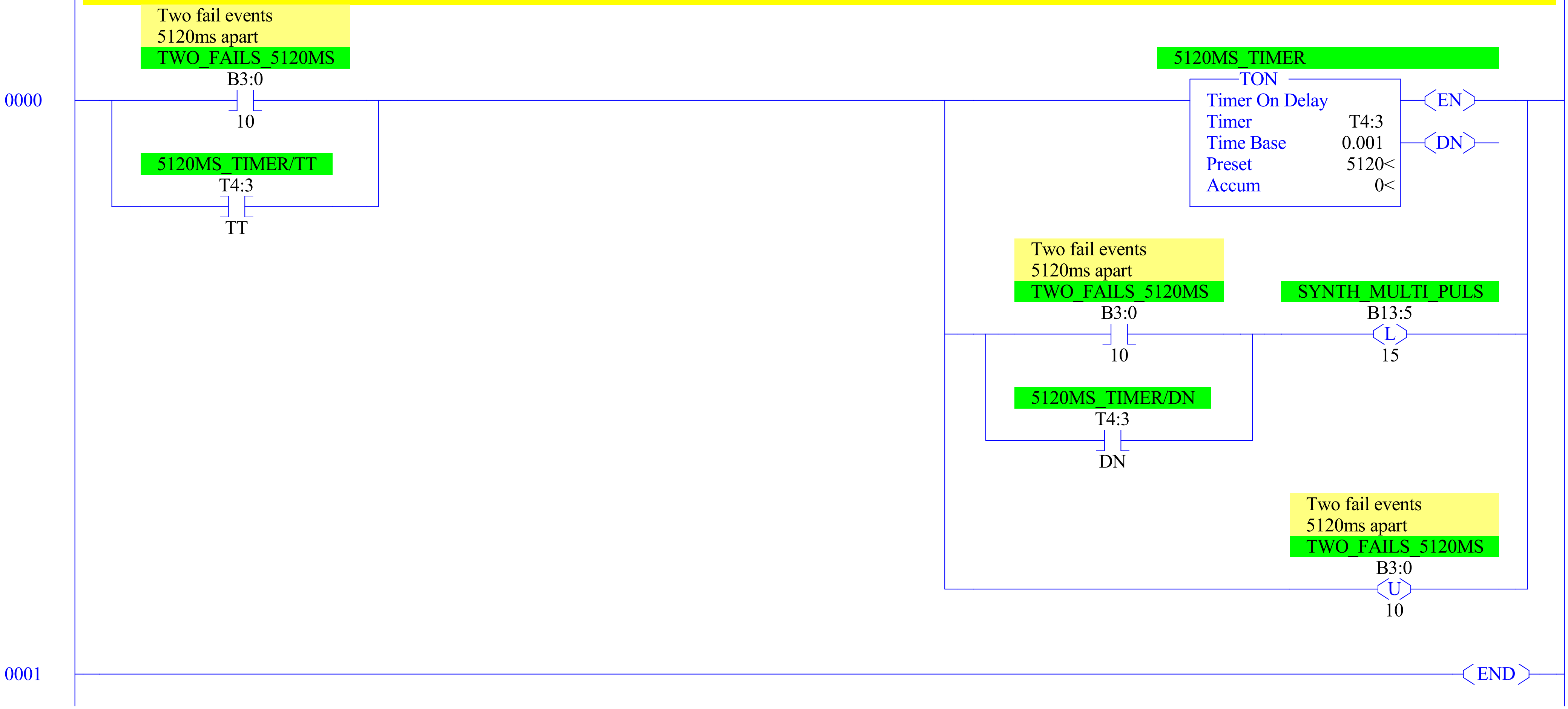
For more detail, refer to the comments in routine SYNMPULSES on the rung of the JSR call of this routine.



Toggling bit TWO_FAILS_5120MS B3:0/10 will trigger two failed can events 5.12s apart, which means the second failed can event will occur* on the same scan cycle that the reject timer object will expire from the first failed can event.

* that is the nominal case, but because the timing is done here in parallel with the reject logic timing, the two timer objects' /DN events may not occur on the same scan cycle.

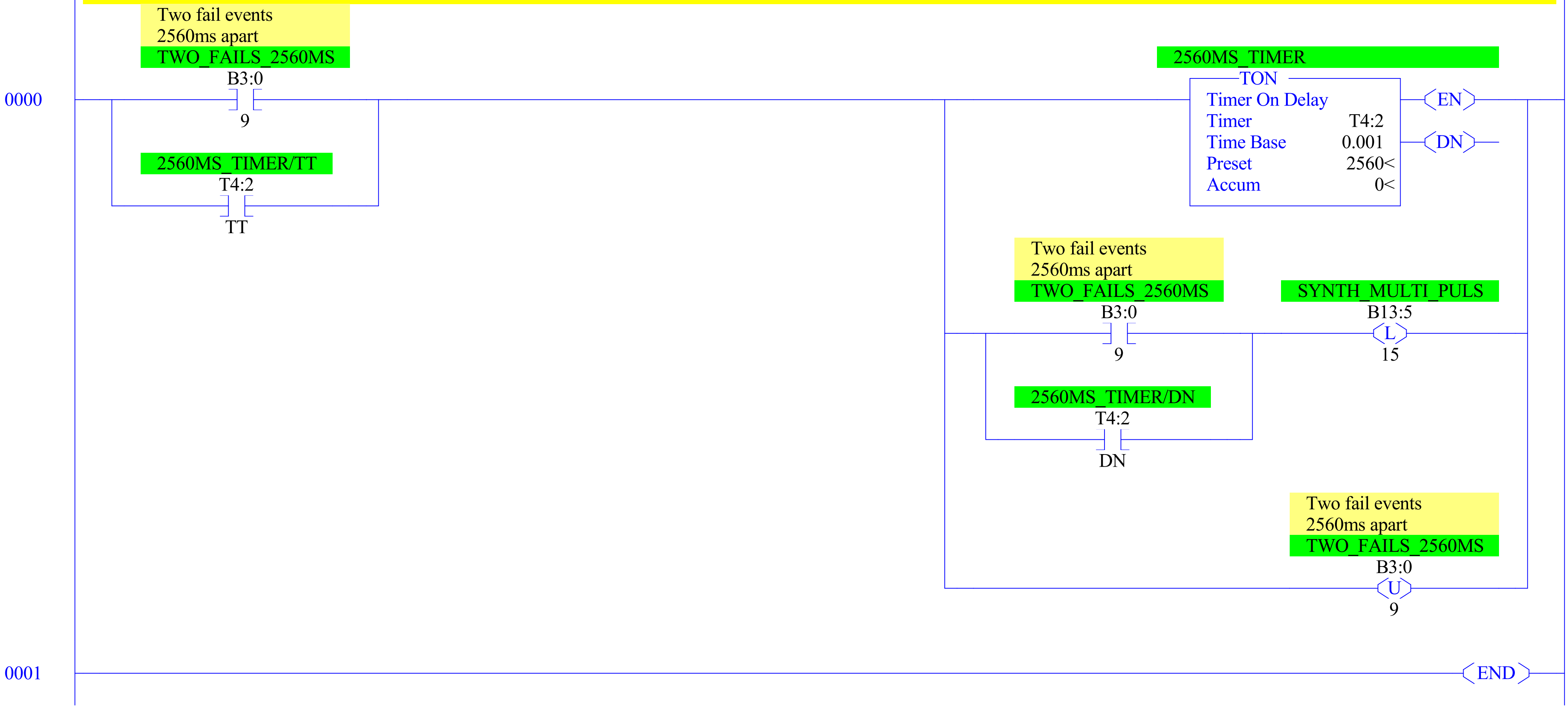
For more detail, refer to the comments in routine SYNMPULSES on the rung of the JSR call of this routine.



Toggling bit TWO_FAILS_2560MS B3:0/9 will trigger two failed can events 2.56s apart, which will cause the reject logic in the main routine LAD 2 CONTINUOUS, to select and start two overlapping reject timer objects, and finally issue two reject triggers, each 5.12s after its corresponding failed can event

* that is the nominal case, but because the timing is done here in parallel with the reject logic timing, the two timer objects' /DN events may not occur on the same scan cycle.

For more detail, refer to the comments in routine SYNMPULSES on the rung of the JSR call of this routine.



Generate synthetic failure pulse(s)

This routine LAD 255 SYNMPULSES allows modeling SYNthetic Multiple failed can event PULSES

Rungs 0000-0002 each call a routine (2FAIL_*) that models a pair of failed can events with various timings within the pair; all of those routines latch the SYNTH_MULTI_PULS bit's value to 1 twice, which triggers the pulse stretcher timer on Rung 0004, which in turn triggers a failed can event on Rung 0004.

==

Rung 0000 can trigger two failed can events 2.56s apart, which will cause the reject logic in the main routine LAD 2 CONTINUOUS, to select and start two overlapping reject timer objects, and finally issue two reject triggers, each 5.12s after its corresponding failed can event

This triggers correct behavior from the reject logic in main routine CONTINUOUS

2_FAILS_2560MS

0000 JSR
Jump To Subroutine
SBR File Number U:254

Rung 0001 can trigger two failed can events 5.12s apart, which means the second failed can event will occur* on the same scan cycle that the reject timer object will expire from the first failed can event.

This results in incorrect behavior from the reject logic in main routine CONTINUOUS: the reject timer object from the first failed can event will still be selected on the scan cycle when when this new second failed can event is "looking" to select a new reject timer object (e.g. Rung 0002 in routine CONTINUOUS), so the logic will skip that already-selected reject timer object. However, that already-selected reject timer object will also expire on the next rung on that same scan cycle, which expiry will unlatch the selected bit (e.g. Rung 0003 in routine CONTINUOUS; bit REJECT1), and that unlatched bit will prevent the next reject timer object from being selected on the following rung (e.g. Rung 0004 in routine CONTINOUUS; see the instruction XIC REJECT1).

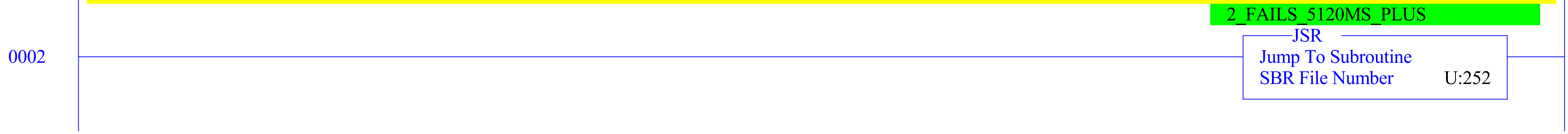
* that is the nominal case, but because the timing is done here in parallel with the reject logic timing, the two timer objects' /DN events may not occur on the same scan cycle.

2_FAILS_5120MS

0001 JSR
Jump To Subroutine
SBR File Number U:253

Rung 0002 can trigger two failed can events 5.12s, plus one scan cycle, apart, which means the second failed can event will occur on the scan cycle immediately after the reject timer object will expired from the first failed can event.

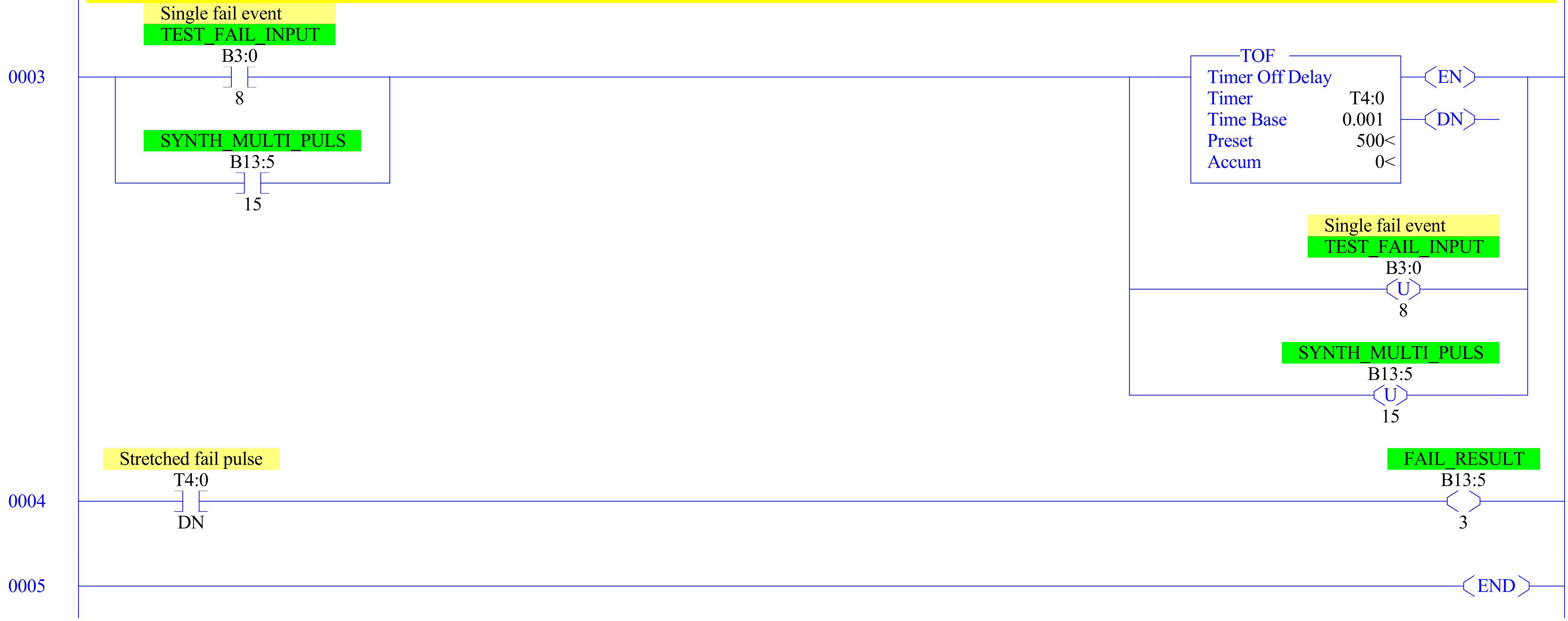
This results in incorrect behavior from the reject logic in main routine CONTINUOUS: the reject timer object from the first failed can event will be selected by this second failed can event, because that reject timer objects selected bit (e.g. REJECT1) will have been unlatched after the timer expired (e.g. Rung 0003 in main routine CONTINUOUS). So on this next scan cycle when the second failed can event triggers, the feed rung into the TON instruction will be true, and since it was also true on the previous scan cycle i.e. when it expired from the first failed can event, the reject timer object will still be enabled as well as expired, so it will immediately issue a second reject trigger by latching the value of TIMER2REJECT to 1 (e.g. Rung 0003 in main routine CONTINUOUS)..



Rungs 0004 and 0005 stretch a synthetic failed can event (TEST_FAIL_INPUT or SYNTH_MULTI_PULS) for half a second. This is longer than would be acceptable in an conveyor-reject system, but it is done here in the demo test program to make it easier to observe the event.

TEST_FAIL_INPUT is meant to be manually triggered (e.g. in RSLogix 5000) to synthetically emulate a single failed can event.

SYNTH_MULTI_PULS is the bit triggered twice by each of the routines 2_FAILS_* called above; each of those routines has an internal bit that can be triggered manually to trigger the two latches of SYNTH_MULTI_PULS at the timing apropos whichever 2_FAILS_* internal bit is manually triggered.



Data File O0 (bin) -- OUTPUT

```
Offset   15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0

O:0.0     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1763     MicroLogix 1100 Series B
O:0.1     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1763     MicroLogix 1100 Series B
O:0.2     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1763     MicroLogix 1100 Series B
O:0.3     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1763     MicroLogix 1100 Series B
```

Data File I1 (bin) -- INPUT

```
Offset   15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0

I:0.0     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1763     MicroLogix 1100 Series B
I:0.1     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1763     MicroLogix 1100 Series B
I:0.2     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1763     MicroLogix 1100 Series B
I:0.3     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1763     MicroLogix 1100 Series B
I:0.4     0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0   Bul.1763     MicroLogix 1100 Series B-Analog Inp 0
I:0.5     0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1   Bul.1763     MicroLogix 1100 Series B-Analog Inp 1
```

Data File S2 (hex) -- STATUS

Main

```
Processor Mode S:1/0 - S:1/4 = Remote Run
On Power up Go To Run (Mode Behavior) S:1/12 = 0
First Pass S:1/15 = No
Free Running Clock S:4 = 0001-1011-0011-0001
```

Proc

```
OS Catalog Number S:57 = 1100           User Program Type S:63 = 8108h
OS Series S:58 = B                      Compiler Revision Number S:64 =
OS FRS S:59 =
Processor Catalog Number S:60 =
Processor Series S:61 = A
Processor FRN S:62 =
```

Scan Times

```
Maximum (x10 ms) S:22 = 25
Watchdog (x10 ms) S:3 (high byte) = 10
Last 100 uSec Scan Time S:35 = 8
Scan Toggle Bit S:33/9 = 1
```

Math

```
Math Overflow Selected S:2/14 = 0            Math Register (lo word) S:13 = 0
Overflow Trap S:5/0 = 0                      Math Register (high word) S:14-S:13 = 0
Carry S:0/0 = 0                              Math Register (32 Bit) S:14-S:13 = 0
Overflow S:0/1 = 0
Zero Bit S:0/2 = 0
Sign Bit S:0/3 = 0
```

Chan 0

```
Processor Mode S:1/0- S:1/4 = Remote Run
Node Address S:15 (low byte) = 0             Outgoing Msg Cmd Pending S:33/2 = 0
Baud Rate S:15 (high byte) = ?
Channel Mode S:33/3 = 0
Comms Active S:33/4 = 0
Incoming Cmd Pending S:33/0 = 0
Msg Reply Pending S:33/1 = 0
```

Debug

```
Suspend Code S:7 = 0
Suspend File S:8 = 0
```

Data File S2 (hex) -- STATUS

Errors

```
Fault Override At Power Up S:1/8 = 0         Fault Routine S:29 = 0
Startup Protection Fault S:1/9 = 0           Major Error S:6 = 0h
Major Error Halt S:1/13 = 0
Overflow Trap S:5/0 = 0                      Error Description:
Control Register Error S:5/2 = 0
Major Error Executing User Fault Rtn. S:5/3 = 0
Battery Low S:5/11 = 0
Input Filter Selection Modified S:5/13 = 0
ASCII String Manipulation error S:5/15 = 0
```

Protection

```
Deny Future Access S:1/14 = No
Data File Overwrite Protection Lost S:36/10 = False
```

Mem Module

```
Memory Module Loaded On Boot S:5/8 = 0
Password Mismatch S:5/9 = 0
Load Memory Module On Memory Error S:1/10 = 0
Load Memory Module Always S:1/11 = 0
On Power up Go To Run (Mode Behavior) S:1/12 = 0
Program Compare S:2/9 = 0
Data File Overwrite Protection Lost S:36/10 = 0
```

Forces

```
Forces Enabled S:1/5 = Yes
Forces Installed S:1/6 = No
```

Data File B3 (bin) -- BINARY

```
Offset   15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0   (Symbol) Description

B3:0      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
Offset   EN TT DN     BASE    PRE    ACC  (Symbol) Description

T4:0      0  0  0 .001 sec    500      0
T4:1      0  0  0 .001 sec     51      0
T4:2      0  0  0 .001 sec   2560      0  (2560MS_TIMER)
T4:3      0  0  0 .001 sec   5120      0  (5120MS_TIMER)
```

Data File C5 -- COUNTER

```
Offset   CU CD DN OV UN UA    PRE    ACC  (Symbol) Description

C5:0      0  0  0  0  0  0   6001      0
```

Data File R6 -- CONTROL

```
Offset   EN EU DN EM ER UL IN FD    LEN    POS  (Symbol) Description

R6:0      0  0  0  0  0  0  0  0      0      0
```

Data File N7 (dec) -- INTEGER

Offset	0	1	2	3	4	5	6	7	8	9
N7:0	0									

Data File F8 -- FLOAT

Offset	0	1	2	3	4
F8:0	0				

Data File L9 (dec) -- RJCTTIMERS

```
Offset          0          1          2          3          4

L9:0            0
```

Data File B13 (bin) -- MISC_BITS

```
Offset   15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0   (Symbol) Description

B13:0     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B13:1     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B13:2     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B13:3     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B13:4     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B13:5     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Data File B103 (bin) -- TMRBSYFLGS

```
Offset   15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0   (Symbol) Description

B103:0    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:2    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:3    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:4    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:5    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:6    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:7    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:8    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:9    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:10   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:11   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:12   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:13   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:14   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:15   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B103:16   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Data File T104 -- RJCTTIMERS

```
Offset   EN TT DN     BASE    PRE    ACC  (Symbol) Description

T104:0    0  0  0 .001 sec   5120      0  (REJECT1_TIMER)
T104:1    0  0  0 .001 sec   5120      0  (REJECT2_TIMER)
T104:2    0  0  0 .001 sec   5120      0  (REJECT3_TIMER)
T104:3    0  0  0  .01 sec      0      0
T104:4    0  0  0  .01 sec      0      0
T104:5    0  0  0  .01 sec      0      0
T104:6    0  0  0  .01 sec      0      0
T104:7    0  0  0  .01 sec      0      0
T104:8    0  0  0  .01 sec      0      0
T104:9    0  0  0  .01 sec      0      0
T104:10   0  0  0  .01 sec      0      0
T104:11   0  0  0  .01 sec      0      0
T104:12   0  0  0  .01 sec      0      0
T104:13   0  0  0  .01 sec      0      0
T104:14   0  0  0  .01 sec      0      0
T104:15   0  0  0  .01 sec      0      0
T104:16   0  0  0  .01 sec      0      0
T104:17   0  0  0  .01 sec      0      0
T104:18   0  0  0  .01 sec      0      0
T104:19   0  0  0  .01 sec      0      0
T104:20   0  0  0  .01 sec      0      0
T104:21   0  0  0  .01 sec      0      0
T104:22   0  0  0  .01 sec      0      0
T104:23   0  0  0  .01 sec      0      0
T104:24   0  0  0  .01 sec      0      0
T104:25   0  0  0  .01 sec      0      0
T104:26   0  0  0  .01 sec      0      0
T104:27   0  0  0  .01 sec      0      0
T104:28   0  0  0  .01 sec      0      0
T104:29   0  0  0  .01 sec      0      0
T104:30   0  0  0  .01 sec      0      0
T104:31   0  0  0  .01 sec      0      0
T104:32   0  0  0  .01 sec      0      0
T104:33   0  0  0  .01 sec      0      0
T104:34   0  0  0  .01 sec      0      0
T104:35   0  0  0  .01 sec      0      0
T104:36   0  0  0  .01 sec      0      0
T104:37   0  0  0  .01 sec      0      0
T104:38   0  0  0  .01 sec      0      0
T104:39   0  0  0  .01 sec      0      0
T104:40   0  0  0  .01 sec      0      0
T104:41   0  0  0  .01 sec      0      0
T104:42   0  0  0  .01 sec      0      0
T104:43   0  0  0  .01 sec      0      0
T104:44   0  0  0  .01 sec      0      0
T104:45   0  0  0  .01 sec      0      0
T104:46   0  0  0  .01 sec      0      0
T104:47   0  0  0  .01 sec      0      0
T104:48   0  0  0  .01 sec      0      0
T104:49   0  0  0  .01 sec      0      0
```

```
Offset   EN TT DN     BASE    PRE    ACC  (Symbol) Description

T104:50   0  0  0  .01 sec      0      0
T104:51   0  0  0  .01 sec      0      0
T104:52   0  0  0  .01 sec      0      0
T104:53   0  0  0  .01 sec      0      0
T104:54   0  0  0  .01 sec      0      0
T104:55   0  0  0  .01 sec      0      0
T104:56   0  0  0  .01 sec      0      0
T104:57   0  0  0  .01 sec      0      0
T104:58   0  0  0  .01 sec      0      0
T104:59   0  0  0  .01 sec      0      0
T104:60   0  0  0  .01 sec      0      0
T104:61   0  0  0  .01 sec      0      0
T104:62   0  0  0  .01 sec      0      0
T104:63   0  0  0  .01 sec      0      0
T104:64   0  0  0  .01 sec      0      0
T104:65   0  0  0  .01 sec      0      0
T104:66   0  0  0  .01 sec      0      0
T104:67   0  0  0  .01 sec      0      0
T104:68   0  0  0  .01 sec      0      0
T104:69   0  0  0  .01 sec      0      0
T104:70   0  0  0  .01 sec      0      0
T104:71   0  0  0  .01 sec      0      0
T104:72   0  0  0  .01 sec      0      0
T104:73   0  0  0  .01 sec      0      0
T104:74   0  0  0  .01 sec      0      0
T104:75   0  0  0  .01 sec      0      0
T104:76   0  0  0  .01 sec      0      0
T104:77   0  0  0  .01 sec      0      0
T104:78   0  0  0  .01 sec      0      0
T104:79   0  0  0  .01 sec      0      0
T104:80   0  0  0  .01 sec      0      0
T104:81   0  0  0  .01 sec      0      0
T104:82   0  0  0  .01 sec      0      0
T104:83   0  0  0  .01 sec      0      0
T104:84   0  0  0  .01 sec      0      0
T104:85   0  0  0  .01 sec      0      0
T104:86   0  0  0  .01 sec      0      0
T104:87   0  0  0  .01 sec      0      0
T104:88   0  0  0  .01 sec      0      0
T104:89   0  0  0  .01 sec      0      0
T104:90   0  0  0  .01 sec      0      0
T104:91   0  0  0  .01 sec      0      0
T104:92   0  0  0  .01 sec      0      0
T104:93   0  0  0  .01 sec      0      0
T104:94   0  0  0  .01 sec      0      0
T104:95   0  0  0  .01 sec      0      0
T104:96   0  0  0  .01 sec      0      0
T104:97   0  0  0  .01 sec      0      0
T104:98   0  0  0  .01 sec      0      0
T104:99   0  0  0  .01 sec      0      0
```

Data File T104 -- RJCTTIMERS

Offset	EN	TT	DN	BASE	PRE	ACC	(Symbol) Description
T104:100	0	0	0	.01 sec	0	0	
T104:101	0	0	0	.01 sec	0	0	
T104:102	0	0	0	.01 sec	0	0	
T104:103	0	0	0	.01 sec	0	0	
T104:104	0	0	0	.01 sec	0	0	
T104:105	0	0	0	.01 sec	0	0	
T104:106	0	0	0	.01 sec	0	0	
T104:107	0	0	0	.01 sec	0	0	
T104:108	0	0	0	.01 sec	0	0	
T104:109	0	0	0	.01 sec	0	0	
T104:110	0	0	0	.01 sec	0	0	
T104:111	0	0	0	.01 sec	0	0	
T104:112	0	0	0	.01 sec	0	0	