

```
1  PROGRAM MAIN
2  VAR
3      iCycleCount : INT ;
4      cycleTrigger : Tc2_Standard . R_TRIG ;
5
6      shiftRegister : WORD := 0 ;
7      shiftInput : BOOL ;
8
9      coverFrontEdge : Tc2_Standard . R_TRIG ;
10     coverBackEdge : Tc2_Standard . F_TRIG ;
11
12     camFrontEdge : Tc2_Standard . R_TRIG ;
13     camBackEdge : Tc2_Standard . F_TRIG ;
14
15     state : StateType := StateType . S0_SHEET_NOT_EXPECTED ;
16     nextState : StateType := StateType . S0_SHEET_NOT_EXPECTED ;
17     first : BOOL := TRUE ;
18
19     resetTimer : TON ;
20     fault : BOOL := FALSE ;
21
22     flasher : TON ;
23
24     i : INT := 0 ;
25 END_VAR
26
27
28     // Every cycle we need to see if there is a cover sheet being loaded
29     // It takes 3 cycles for a cover sheet to get to the scorer
30     // If there is a cover sheet in the scorer we need to check
31     //         - that there is a sheet there during the first pulse
32     //         - that there isn't a sheet present during the second pulse
33     // At all other times we shouldn't see a sheet in the scorer for either
34     pulse
35
36     Global . bCoverSheetDetected := NOT Global . bCoverFeedNoSheet ;
37
38     cycleTrigger ( CLK := Global . bCycleTrigger ) ;
39     coverFrontEdge ( CLK := Global . bScorerSheetDetected ) ;
40     coverBackEdge ( CLK := Global . bScorerSheetDetected ) ;
41     camFrontEdge ( CLK := Global . bScorerCam ) ;
42     camBackEdge ( CLK := Global . bScorerCam ) ;
43
44     resetTimer ( IN := fault , PT := T#1S ) ;
45     flasher ( IN := NOT flasher . Q , PT := T#250MS ) ;
46
47     IF cycleTrigger . Q THEN
48         iCycleCount := iCycleCount + 1 ;
49         shiftRegister := SHL ( shiftRegister , 1 ) ;
50         shiftRegister . 0 := Global . bCoverSheetDetected ;
51     END_IF ;
52
53     CASE state OF
54         StateType . S0_SHEET_NOT_EXPECTED :
55             IF first THEN
56                 fault := FALSE ;
```

```
29         END_IF
30
31         IF shiftRegister . 1 THEN
32             nextState := StateType . S1_SHEET_SEARCH ;
33             ADSLOGSTR ( msgCtrlMask := ADSLOG_MSGTYPE_STRING , msgFmtStr :=
'S0 -> S1' , strArg := '' ) ;
34             ELIF coverFrontEdge . Q AND NOT shiftRegister . 0 THEN
35                 nextState := StateType . S8_FAULT ;
36                 ADSLOGSTR ( msgCtrlMask := ADSLOG_MSGTYPE_STRING , msgFmtStr :=
'S0 -> S8' , strArg := '' ) ;
37             END_IF
38
39         StateType . S1_SHEET_SEARCH :
40             IF camFrontEdge . Q THEN
41                 IF Global . bScorerSheetDetected THEN
42                     nextState := StateType . S2_BREAK_SEARCH ;
43                     ADSLOGSTR ( msgCtrlMask := ADSLOG_MSGTYPE_STRING , msgFmtStr
:= 'S1 -> S2' , strArg := '' ) ;
44                 ELSE
45                     nextState := StateType . S8_FAULT ;
46                     ADSLOGSTR ( msgCtrlMask := ADSLOG_MSGTYPE_STRING , msgFmtStr
:= 'S1 -> S8' , strArg := '' ) ;
47                 END_IF
48             END_IF
49
50         StateType . S2_BREAK_SEARCH :
51             IF camFrontEdge . Q THEN
52                 IF NOT Global . bScorerSheetDetected THEN
53                     nextState := StateType . S3_BREAK_COMPLETE ;
54                     ADSLOGSTR ( msgCtrlMask := ADSLOG_MSGTYPE_STRING , msgFmtStr
:= 'S2 -> S3' , strArg := '' ) ;
55                 ELSE
56                     nextState := StateType . S8_FAULT ;
57                     ADSLOGSTR ( msgCtrlMask := ADSLOG_MSGTYPE_STRING , msgFmtStr
:= 'S2 -> S8' , strArg := '' ) ;
58                 END_IF
59             END_IF
60
61         StateType . S3_BREAK_COMPLETE :
62             IF camBackEdge . Q THEN
63                 IF NOT Global . bScorerSheetDetected THEN
64                     nextState := StateType . S4_CYCLE_COMPLETE ;
65                     ADSLOGSTR ( msgCtrlMask := ADSLOG_MSGTYPE_STRING , msgFmtStr
:= 'S3 -> S4' , strArg := '' ) ;
66                 ELSE
67                     nextState := StateType . S8_FAULT ;
68                     ADSLOGSTR ( msgCtrlMask := ADSLOG_MSGTYPE_STRING , msgFmtStr
:= 'S3 -> S8' , strArg := '' ) ;
69                 END_IF
70             END_IF
71
72         StateType . S4_CYCLE_COMPLETE :
73             IF cycleTrigger . Q THEN
74                 nextState := StateType . S0_SHEET_NOT_EXPECTED ;
75                 ADSLOGSTR ( msgCtrlMask := ADSLOG_MSGTYPE_STRING , msgFmtStr :=
'S4 -> S0' , strArg := '' ) ;
```

```
76         END_IF
77
78     StateType . S8_FAULT :
79         Global . bMisfeedAlarm := TRUE ;
80         fault := TRUE ;
81         IF flasher . Q THEN
82             Global . bMisfeedAlarm2 := TRUE ;
83         ELSE
84             Global . bMisfeedAlarm2 := FALSE ;
85         END_IF
86         IF resetTimer . Q THEN
87             Global . bMisfeedAlarm := FALSE ;
88             Global . bMisfeedAlarm2 := FALSE ;
89             nextState := StateType . S4_CYCLE_COMPLETE ;
90             ADSLOGSTR ( msgCtrlMask := ADSLOG_MSGTYPE_STRING , msgFmtStr :=
'S8 -> S4' , strArg := '' ) ;
91         END_IF
92     END_CASE
93
94     IF state <> nextState THEN
95         first := TRUE ;
96         state := nextState ;
97         FOR i := 32 TO 2 BY -1 DO
98             Global . stateHistory [ i ] := Global . stateHistory [ i - 1 ] ;
99         END_FOR
100         Global . stateHistory [ 1 ] := state ;
101     ELSE
102         first := FALSE ;
103     END_IF
104
```