```
1        PROGRAM  Weighing_Station
2        VAR
3            Ons6_Front_Entry              : F_TRIG ;
4            Ons7_Left_Entry               : F_TRIG ;
5            Ons8_Right_Entry              : F_TRIG ;
6        END_VAR
7
```

```
1        (*
2        * File: Weighing Station
3        * Author: Jaime Calvente Mieres
4        * Date: 14-08-2022
5        * Description: Program to control the weight management process of boxes
6        * and paht selection depending on size.
7        *)
8
9
10       (*
11       * If system active and there is a box at scale sensor then start weighing
12       * the boxes and set a weighin time. When the time has passed, the output of
13       * the timer will free up the path selection to exit the boxes.
14       *)
15       IF  System_Active  THEN
16           // One shot instances
17           Ons6_Front_Entry ( CLK  :=  At_Forward_Entry ) ;
18           Ons7_Left_Entry ( CLK  :=  At_Left_Entry ) ;
19           Ons8_Right_Entry ( CLK :=  At_Right_Entry ) ;
20
21           // Timer instantiation
22           Scale_Timer ( IN  :=  ( At_Scale  AND  NOT  Soft_Stop_Active ) ,  PT  :=
         Weighin_Time ,  ET  =>  Time_Scale_Elap ) ;
23
24           (* Box Path selection by weight when timer out is "On" *)
25           IF  Scale_Timer . Q  THEN
26               // Large box size to Front
27               IF  ( Weight  >=  5.0 )  AND  NOT  ( M_Box_To_Right  OR  M_Box_To_Left )
         THEN
28                   M_Box_To_Front  :=  TRUE ;
29               // Medium size box to Right
30               ELSIF  ( Weight  >=  3.5  AND  Weight  <  5.0 )  AND  NOT  ( M_Box_To_Front
          OR  M_Box_To_Left )  THEN
31                   M_Box_To_Right  :=  TRUE ;
32               // Small size box to Left
33               ELSIF  ( Weight  >=  1.5  AND  Weight  <  3.5 )  AND  NOT  ( M_Box_To_Front
          OR  M_Box_To_Right )  THEN
34                   M_Box_To_Left  :=  TRUE ;
35               END_IF
36           END_IF
37
38           (* Box Path selector Deactivations *)
39           // Front path reset
40           IF  Ons6_Front_Entry . Q  THEN
41               M_Box_To_Front  :=  FALSE ;
42           END_IF
43           // Letf path reset
44           IF  Ons7_Left_Entry . Q  THEN
```

```
45              M_Box_To_Left  := FALSE ;
46          END_IF
47          // Right path reset
48          IF Ons8_Right_Entry . Q THEN
49              M_Box_To_Right  := FALSE ;
50          END_IF
51
52      (* If not system active due to an emergency stop
53      or a soft stop, deactivate the box selector and
54      reset the timer. *)
55      ELSE
56          Scale_Timer . IN  := FALSE ;
57          M_Box_To_Front  := FALSE ;
58          M_Box_To_Left  := FALSE ;
59          M_Box_To_Right  := FALSE ;
60      END_IF
61
```