```
PROGRAM MAIN
VAR

    Timer1  : TON ;
    Timer2  :  TON ;
    PulseTimer1  :  TP ;
    PulseTimer2  :  TP ;

    CurrentState   :  States ;

    ServoRTrigY  :  R_Trig  ;
    ServoRTrigX  :  R_Trig  ;

    BusyFtrigX  :  F_Trig ;
    BusyFtrigY  :  F_Trig ;


    //Eigen trigger booleans
    InitYReady  :  BOOL ;
    InitXReady  :  BOOL ;

    ReadyX  :  BOOL ;
    ReadyY  :  BOOL ;

    Full  :  BOOL ;
    Busy  :  BOOL ;

END_VAR
```

```
PulseTimer1 ( IN := GVL_IO . Start_button ,  PT := T#150MS ) ;
PulseTimer2 ( IN := Busy ,  PT := T#150MS ) ;

Timer1 ( IN :=  GVL_IO . a_1_Vertical_Zaxis_downward ,  PT :=  T#10MS ) ;

CASE  CurrentState  OF
States . Init :
        //Aanzetten van servo's. Dit kan even duren GEEN HOMING
        GVL_IO . SVONX_SERVO_ON_tableX      :=  TRUE ;
        GVL_IO . SVONY_SERVO_ON_translY     :=  TRUE ;

        GVL_IO . SETUPX_ORIGIN_tableX  :=  PulseTimer1 . Q ;
        GVL_IO . SETUPY_ORIGIN_translY  :=  PulseTimer1 . Q ;

        //// Koppelen van outputs Busy aan var's. Triggered op een falling edge.
        IF  PulseTimer1 . Q  THEN
            ServoRTrigX  ( clk  :=  gvl_io . SETONX_ORIGIN_tableX  ) ;
            ServoRTrigY  ( clk  :=  gvl_io . SETONY_ORIGIN_translationY  ) ;
        END_IF

        // Bools setten aan de hand van triggers.
```

```
22              IF ServoRTrigX.Q THEN
23                  InitXReady := TRUE;
24              END_IF
25
26              IF ServoRTrigY.Q THEN
27                  InitYReady := TRUE;
28              END_IF
29
30              // Init voltooid ga naar idle
31              IF InitXReady AND InitYReady THEN
32                  CurrentState := States.Idle;
33              END_IF
34
35      States.Idle:
36              IF NOT GVL_IO.Stop_button AND NOT GVL_IO.dp2_Cover_detection THEN
37                  //SEND MESSAGEE TODO
38                  CurrentState := States.Idle;
39              END_IF
40
41              IF NOT GVL_IO.Stop_button AND GVL_IO.dp1_Can_detection AND GVL_IO.
        dp2_Cover_detection AND NOT Full THEN
42                  CurrentState := States.MoveToPickUp; //TODO stopbutton
43              END_IF
44
45              IF Full THEN
46                  CurrentState := States.PalletFull;
47              END_IF
48
49      States.MoveToPickUp:
50              Busy := TRUE;
51              GVL_IO.INX0_from_Input0_tableX := FALSE;
52              GVL_IO.INX1_from_Input1_tableX := FALSE;
53              GVL_IO.INX2_from_Input2_tableX := TRUE;
54              GVL_IO.INY0_from_Input0_translY := TRUE;
55              GVL_IO.INY1_from_Input1_translY := FALSE;
56              GVL_IO.INY2_from_Input2_translY := TRUE;
57              GVL_IO.DRIVEX_MOVE_tableX := PulseTimer2.Q;
58              GVL_IO.DRIVEY_MOVE_translY := PulseTimer2.Q;
59
60              //// Koppelen van outputs Busy aan var's. Triggered op een falling edge.
61              BusyFtrigX (clk := gvl_io.BUSYX_BUSY_tableX);
62              BusyFtrigY (clk := gvl_io.BUSYY_BUSY_traslationY);
63
64              // Bools setten aan de hand van triggers.
65              IF BusyFtrigX.Q THEN
66                  ReadyX := TRUE;
67              END_IF
68
69              IF BusyFtrigY.Q THEN
70                  ReadyY := TRUE;
71              END_IF
```

```
72
73              // Init voltooid ga naar idle
74              IF  ReadyY  AND  ReadyX  THEN
75                  GVL_IO . DRIVEX_MOVE_tableX  :=  FALSE ;
76                  GVL_IO . DRIVEY_MOVE_translY  :=  FALSE ;
77                  ReadyX :=  FALSE ;
78                  ReadyY :=  FALSE ;
79                  Busy :=  FALSE ;
80                  CurrentState  :=  States . PickUpDown ;
81              END_IF
82
83      States . PickUpDown :
84              IF  GVL_IO . a_0_Vertical_Zaxis_upward  THEN
85                  GVL_IO . Aplus_Vert_axis_downwards  :=  TRUE ;
86              END_IF
87
88              IF  GVL_IO . a_1_Vertical_Zaxis_downward  THEN
89                  GVL_IO . Vplus_Vacuum_start  :=  TRUE ;
90
91                  IF  Timer1 . Q  THEN
92                  CurrentState  :=  States . PickUpUp ;
93                  END_IF
94              END_IF
95
96      States . PickUpUp :
97              GVL_IO . Aplus_Vert_axis_downwards  :=  FALSE ;
98
99              IF  GVL_IO . a_0_Vertical_Zaxis_upward  THEN
100                 CurrentState  :=  States . MoveX ;
101             END_IF
102
103     States . MoveX :
104             CurrentState  :=  States . MoveY ;
105
106     States . MoveY :
107             CurrentState  :=  States . DropDown ;
108
109     States . DropDown :
110             IF  GVL_IO . a_0_Vertical_Zaxis_upward  THEN
111                 GVL_IO . Aplus_Vert_axis_downwards  :=  TRUE ;
112             END_IF
113
114             IF  GVL_IO . a_1_Vertical_Zaxis_downward  THEN
115                 GVL_IO . Vplus_Vacuum_start  :=  FALSE ;
116                 Timer2 ( IN := GVL_IO . a_1_Vertical_Zaxis_downward ,  PT := T#10MS ) ;
117                 IF  Timer2 . Q  THEN
118                     CurrentState  :=  States . DropUp ;
119                 END_IF
120             END_IF
121
122
```

```
123     States . DropUp :
124             IF  GVL_IO . a_1_Vertical_Zaxis_downward  THEN
125                 GVL_IO . Aplus_Vert_axis_downwards  :=  FALSE ;
126             END_IF
127
128             IF  GVL_IO . a_0_Vertical_Zaxis_upward  THEN
129                 CurrentState  :=  States . Idle ;
130             END_IF
131     END_CASE
132
```