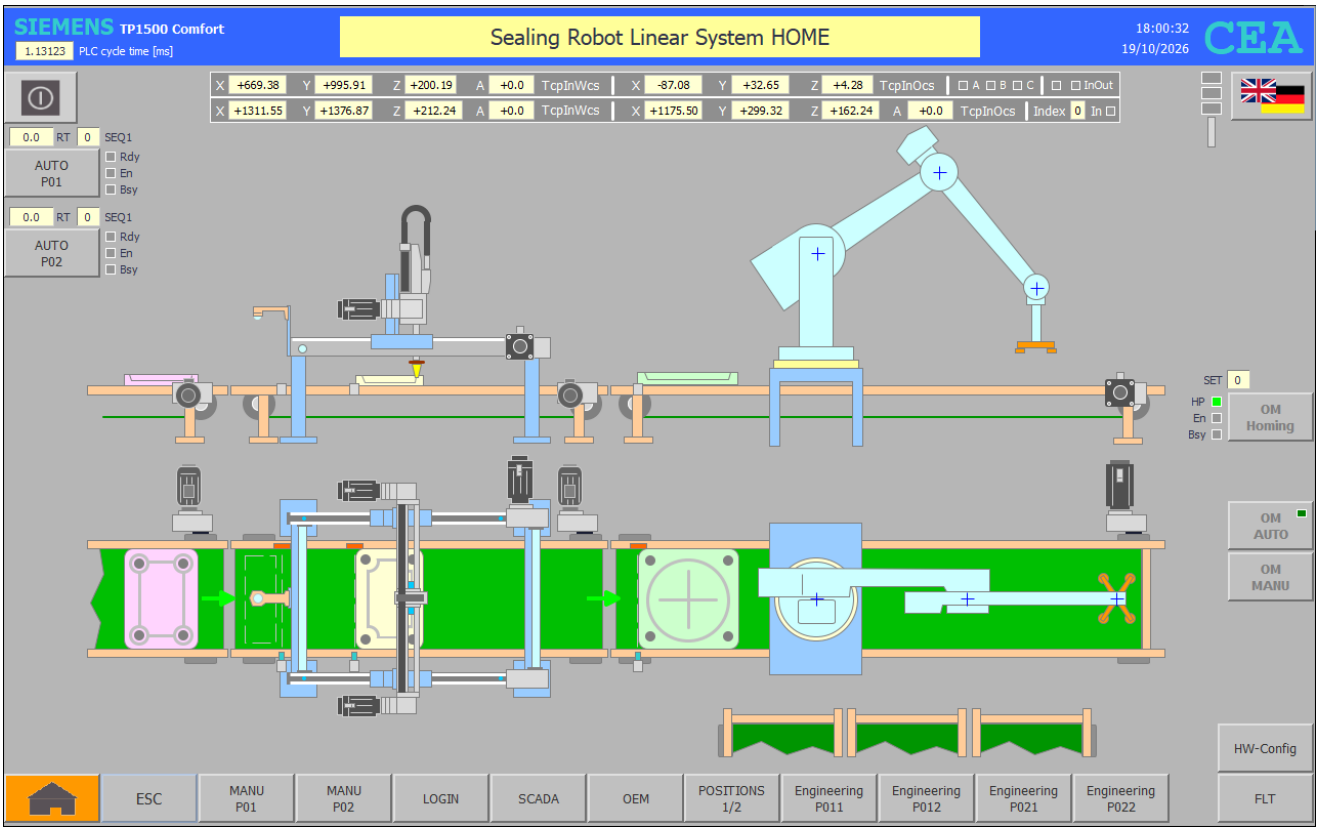
Sealing Robot Linear System



Contents

Sealing Robot Linear System 1
1. Revision History 3
2. Purpose of this document 4
3. Main pre-setting 4
4. Main objectives 4
5. Velocity handling 4
6. Process module P01 - Sealing Robot 4
7. Process module P02 - Pick and Place Robot 4
8. General disclaimer 5
9. Preamble of Manual Mode 5
10. Detailed Description 7
11. Programming 8
12. Operating Mode Automatic of P01 and P02 9
13. FB SEQ1 P01 Sealing Robot Programme Automatic Mode 10
14. FB SEQ1 P02 Pick & Place Robot Linear Programme Automatic Mode 11
15. Track Conveyor explained 11
16. Sub Function SEQ331 14
17. Sub Function SEQ332 15
18. Sub Function SEQ333 16
19. Learning from - and about 17
20. Project Videos 17

1. **Revision History**

Date:	Name:	Comment:
13.10.2026	mc	Texting basic objectives
19.10.2026	mc	Screenshot introduction page

2. **Purpose of this document**

Sealing surface technology's main task is to exclude external substances such as fluid or dust that could contaminate a sealing system.
CEA describes in this document the digital twin which was the fundament of a real project, i.e., sealing-linear-system.

3. **Main pre-setting**

- OEM Setting (Original Equipment Manufacturer)
- SCADA Setting (System Control And Data Acquisition)

4. **Main objectives**

- Enable different dynamics of equipment whilst running
- Camera detection of 3-types of protective covers
- Covers picking during non-stop conveying
- Speedy transmission between two robot-systems
- Energy efficiency e.g. automatically stop due to supply shortfall

5. **Velocity handling**

It enables speed adaption during production. The given SCADA velocity can be changed via HMI during production without losing production quality.

6. **Process module P01 - Sealing Robot**

A camera informs the robot of the current protective cover which is about to arrive while crossing a light barrier, and thus enables the robot to move towards the required starting position. The robot controls a sealing compound valve that is to close in case of moving from one starting point to another.

7. **Process module P02 - Pick and Place Robot**

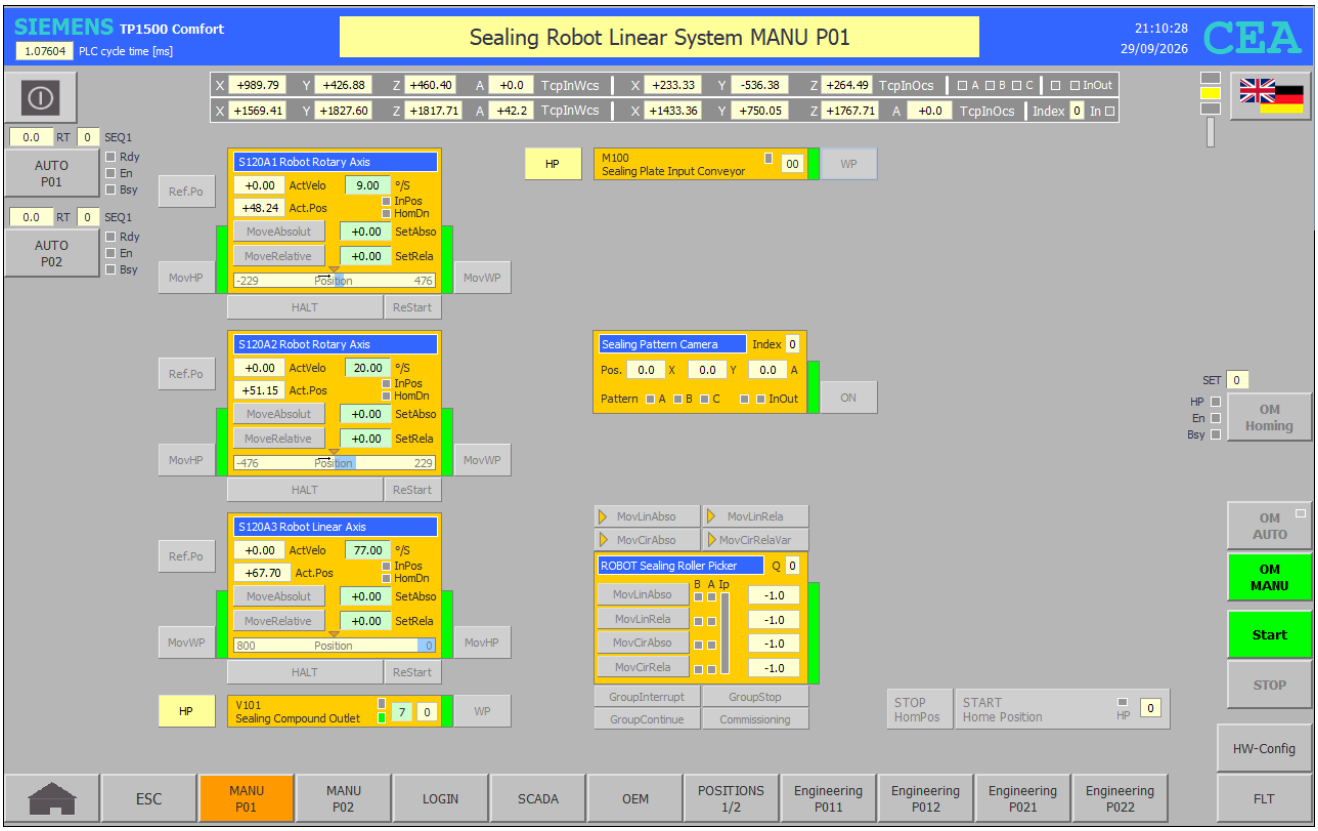
As soon as the sealing process is finished the robot is then enabled to move to the next calculated starting position, and wait for the protective cover to come. The robot picks the cover during moving using the DOWN>FOLLOW>UP principle.

8. **General disclaimer**

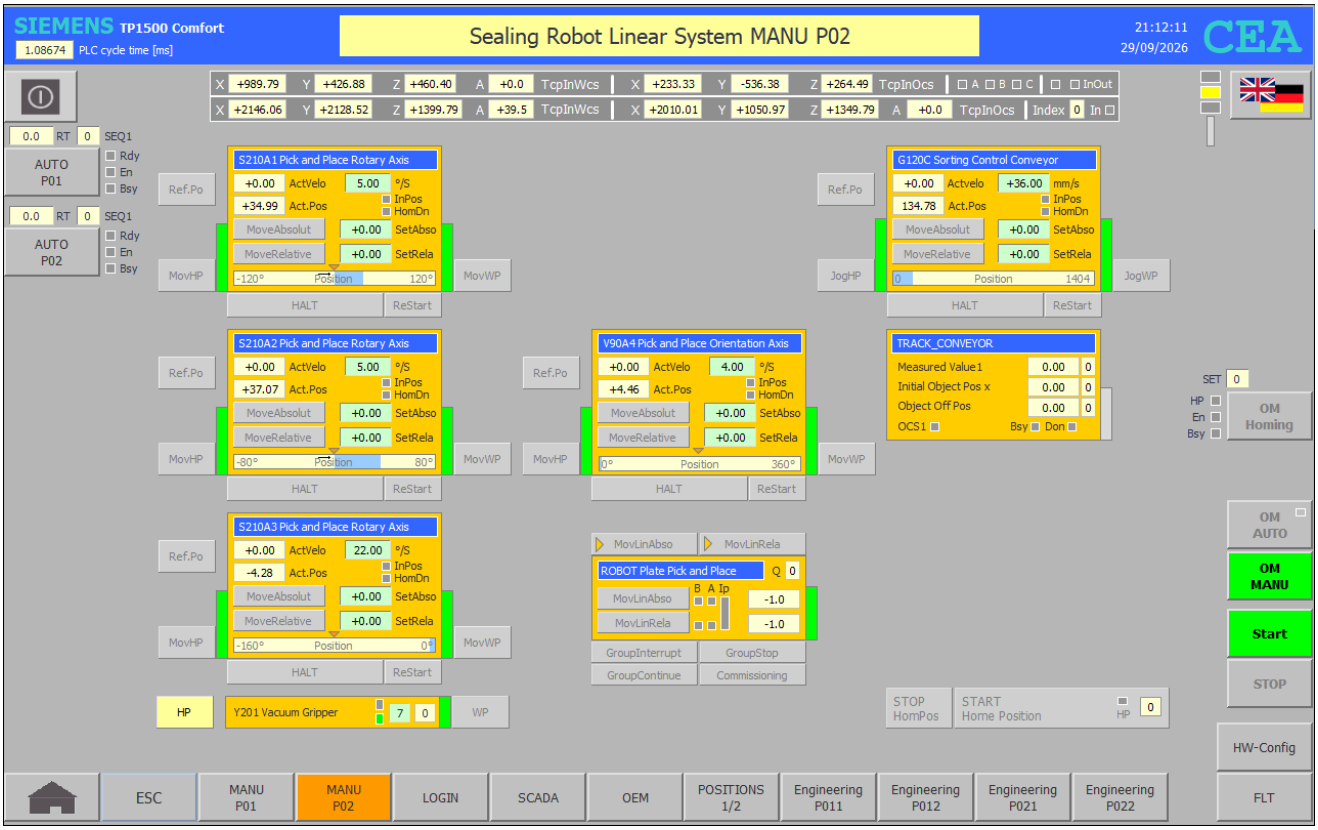
CEA will in no event be liable for any loss or damage including without limitation, indirect or consequential loss or damage to PLC data, or any damage whatsoever arising from culpable loss of life, bodily injury or damage to health arising out of, or in connection with the use of this documentation or PLC-programme.

9. **Preamble of Manual Mode**

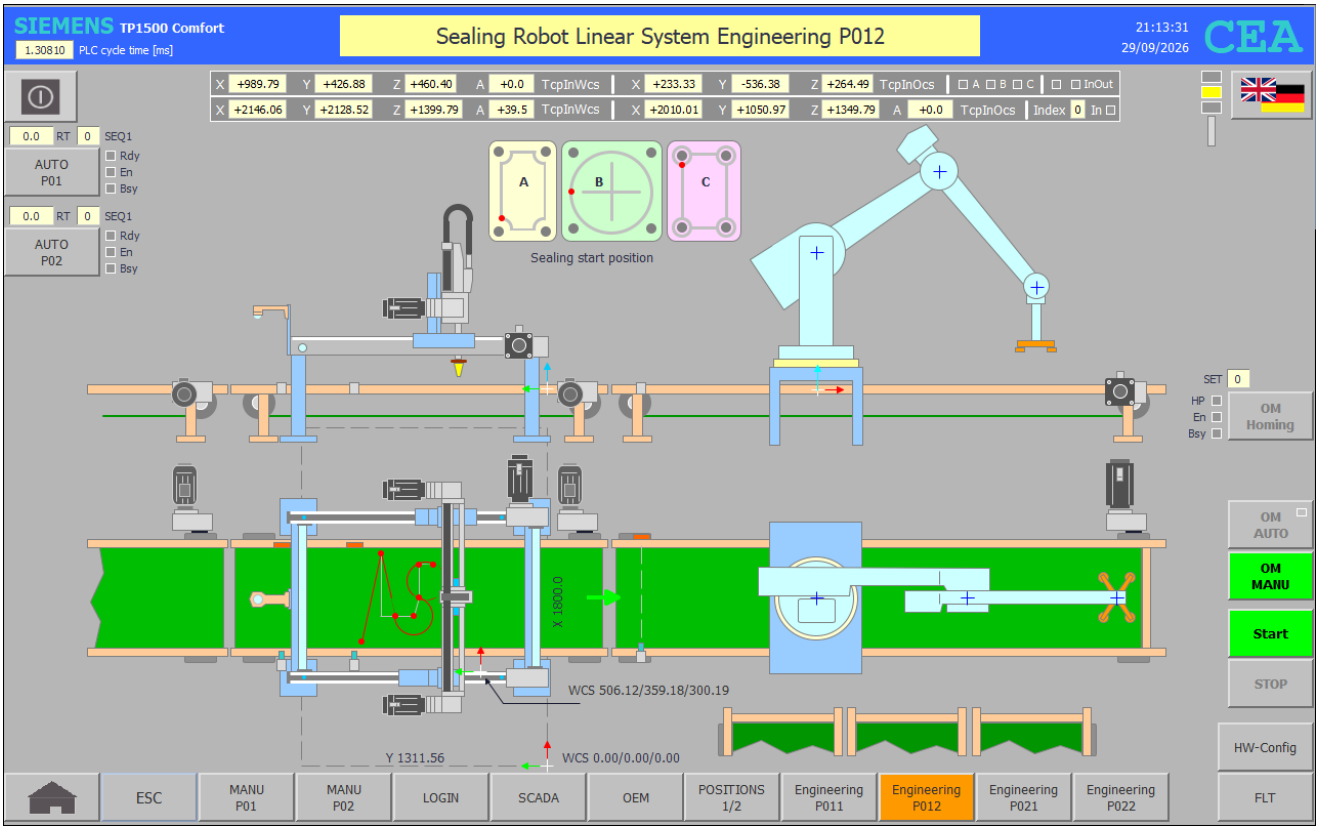
There have been no changes taken place in this presented CEA Manual Mode application. Manual Mode has been conceptualized to operate assembly units during commissioning of the plant regardless of any faults, unready units or of any process modules. Furthermore, it can interrupt Automatic Mode for any necessary correction and then back to continue the automatic process.



Human Machine Interface (HMI) of all available actuators presented in process-module P01 for the commissioning, and later for need of optimisation.



Human Machine Interface (HMI) of all available actuators presented in process-module P02 for the commissioning, and later for need of optimisation.



10. **Detailed Description**

A Preinstalled Conveyor delivers protective covers considered as a separate entity. The covers are delivered without any certain order i.e. only type A or B or C but randomly assigned, C, A or B, etc.

P01 Sealing-Robot on the left side is the module main actuator. A camera is setup at entrance of the process module to picture the coming cover, and records it in a shift register array of 5, each contain 3-bits: Product A, B or C. The module's conveyor transports the covers to a starting sealing position crossing two light barriers to ensure that only one cover is presented at working zone, and then stop. By ending the sealing sequence P01.to_P02 sends via his interface an EnableToRun TRUE-Signal.
P01 switches on the conveyor again to allow transfer P01 to P02.

P02 Pick and Place Robot on the right side is the module main actuator. A light barrier confirms that a sealing cover has entered the module. The module's conveyor is steadily on hence travels towards the calculated picking position. The robot waits for the cover to come at that position in order to be picked up throughout the transport. The robot arm moves down the cover centre point, activates his vacuum gripper during following the cover movement to then pick

it up, and place it on the due output conveyor outside the machine. At this time, the next sealing cover pickup-position is already calculated. The robot arm moves to the new calculated position, and waits for the cover to come.

11. Programming

The Sealing Robot Linear System includes two process modules P01 and P02. Both have robot kinematics on board. On the left side a Roller-Picker-3D which is the Sealing Robot of P01, on the right side an Articulated-Arm 3D with orientation which is the Pick and Place Robot of P02.

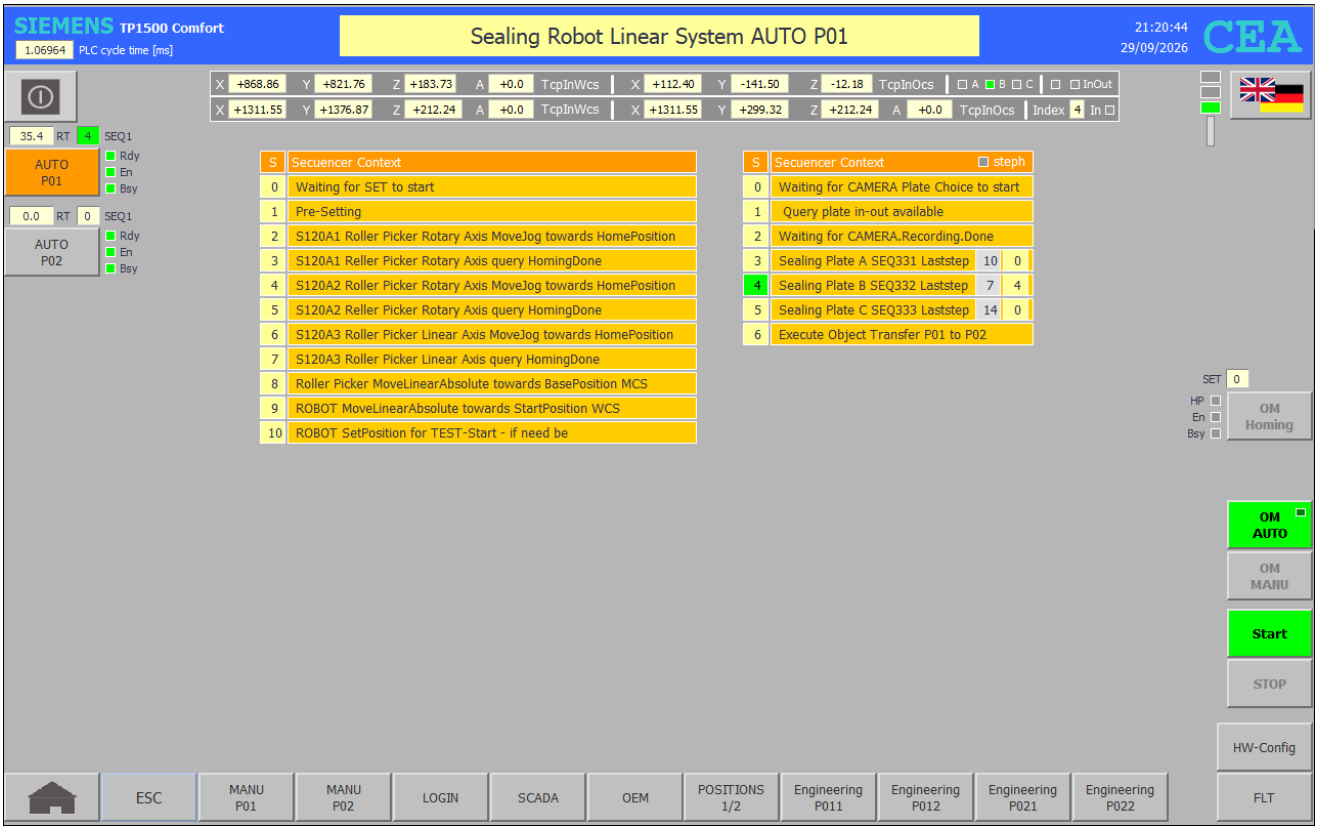
Network 6: ROBOT Sealing Robot - Roller Picker 3D (P01)

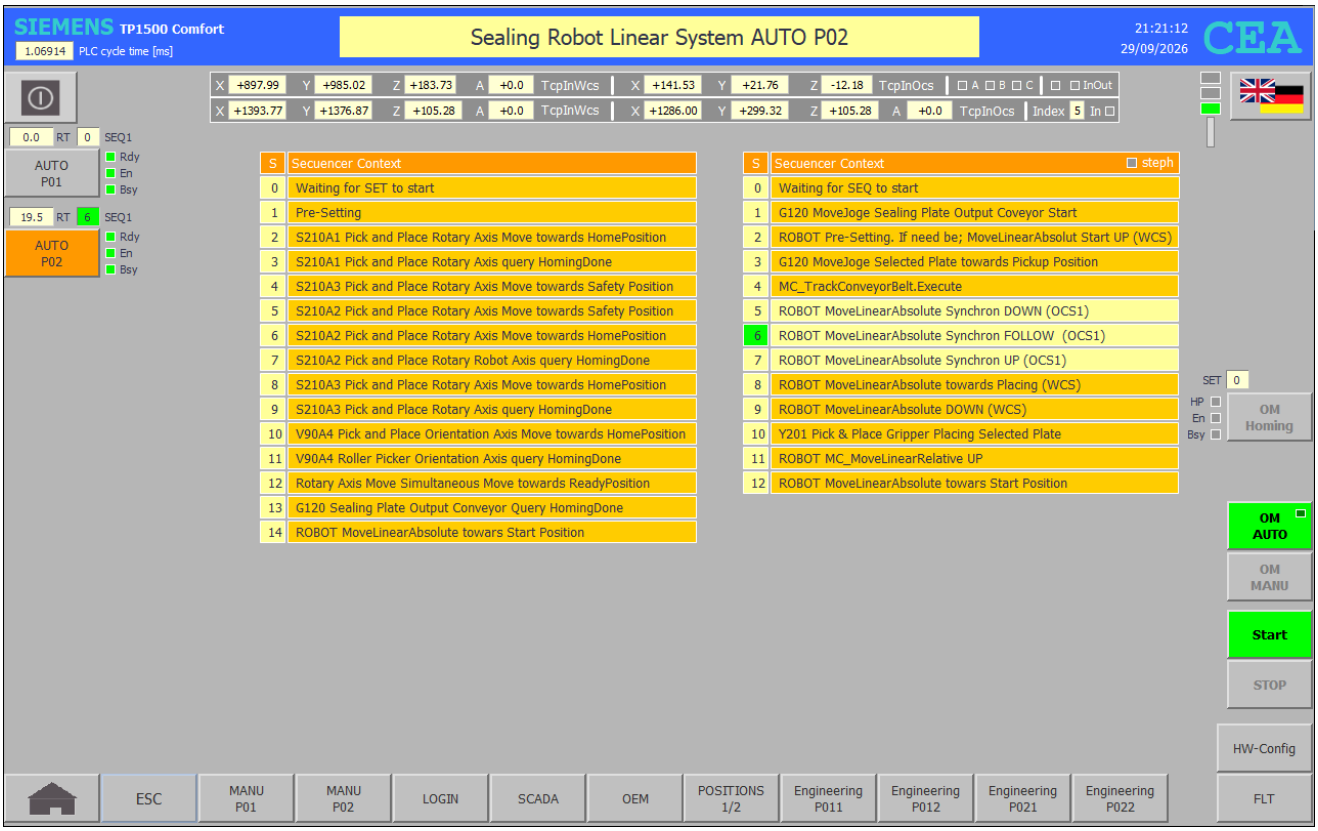
```
IF #ROBOT.DRV.MC_GroupStop.Done THEN
    "DB HMI KEY".P01.ROBOT.F5_GroupStop := FALSE;
END_IF;
// Enable disable Kinematics movement
#ROBOT.DRV.Support.KINEMATICS_Enabled := #S120A1.DRV.Support.Epos_Enabled AND #S120A2.DRV.Support.Epos_Enabled AND #S120A3.DRV.Support.Epos_Enabled;
// Assigning InOperation to enable ROBOT
#ROBOT.DRV.Support.InOperation := #S120A1.DRV.Support.InOperation AND #S120A2.DRV.Support.InOperation AND #S120A3.DRV.Support.InOperation;
#ROBOT(ENABLE := #S120A1.DRV.MC_Power.Busy AND #S120A2.DRV.MC_Power.Busy AND S120A3.DRV.MC_Power.Busy,
       GROUP_STOP := #ROBOT.DRV.Support.GroupStop AND #tmp_operation_mode OR "DB HM IEY".P01.ROBOT.F5_GroupStop AND #ROBOT.DRV.Support.Busy,
       GROUP_INTERRUPT := #Manual.#ROBOT.GroupInterrupt,
       GROUP_CONTINUE := "DI MAIN GLB".M00.EntirePlant_Run_Impulse AND #tmp_operation_mode OR #Manual.ROBOT.GroupContinue,
       MCB := "E00 ROBOT_01 CB",
       MOVE_LINEAR_ABSOLUTE := #ROBOT.DRV.Support.MoveLinearAbsolute AND #tmp_operation_mode OR #Manual.ROBOT.MoveLinearAbsolute,
       MOVE_LINEAR_RELATIVE := #ROBOT.DRV.Support.MoveLinearRelative AND #tmp_operation_mode OR #Manual.ROBOT.MoveLinearRelative,
       MOVE_CIRCULAR_ABSOLUTE := #ROBOT.DRV.Support.MoveCircularAbsolute AND #tmp_operation_mode OR #Manual.ROBOT.MoveCircularAbsolute,
       MOVE_CIRCULAR_RELATIVE := #ROBOT.DRV.Support.MoveCircularRelative AND #tmp_operation_mode OR #Manual.ROBOT.MoveCircularRelative,
       ACKNOWLEDGE_FLT := "DI MAIN GLB".M00.FLT_EntirePlant_ACK_REg,
       KINEMATICS := "+1-ROBOT");
```

Network 8: ROBOT Pick and Place Robot - Articulated Arm 3D (P02)

```
IF #ROBOT.DRV.MC_GroupStop.Done THEN
    "DB HMI KEY".P02.ROBOT.F5_GroupStop := FALSE;
END_IF;
// Enable disable Kinematics movement
#ROBOT.DRV.Support.KINEMATICS_Enabled := #S210A1.DRV.Support.Epos_Enabled AND #S210A2.DRV.Support.Epos_Enabled AND #S210A3.DRV.Support.Epos_Enabled AND
#V90A4.DRV.Support.Epos_Enabled;
// Assigning InOperation to enable ROBOT
#ROBOT.DRV.Support.InOperation := #S210A1.DRV.Support.InOperation AND #S210A2.DRV.Support.InOperation AND #S210A3.DRV.Support.InOperation AND
#V90A4.DRV.Support.InOperation;
#ROBOT(ENABLE := #S210A1.DRV.MC_Power.Busy AND #S210A2.DRV.MC_Power.Busy AND #S210A3.DRV.MC_Power.Busy AND #V90A4.DRV.MC_Power.Busy,
       GROUP_STOP := #ROBOT.DRV.Support.GroupStop AND #tmp_operation_mode OR "DB HMI KEY".P02.ROBOT.F5_GroupStop AND #ROBOT.DRV.Support.Busy,
       GROUP_INTERRUPT := #Manual.#ROBOT.GroupInterrupt,
       GROUP_CONTINUE := "DI MAIN GLB".M00.EntirePlant_Run_Impulse AND #tmp_operation_mode OR #Manual.ROBOT.GroupContinue,
       MCB := "E00 ROBOT_02 CB",
       MOVE_LINEAR_ABSOLUTE := #ROBOT.DRV.Support.MoveLinearAbsolute AND #tmp_operation_mode OR #Manual.ROBOT.MoveLinearAbsolute,
       MOVE_LINEAR_RELATIVE := #ROBOT.DRV.Support.MoveLinearRelative AND #tmp_operation_mode OR #Manual.ROBOT.MoveLinearRelative,
       ACKNOWLEDGE_FLT := "DI MAIN GLB".M00.FLT_EntirePlant_ACK_REg,
       KINEMATICS := "+2-ROBOT");
```

12. Operating Mode Automatic of P01 and P02





CEA standard sequencers are in Operation Mode Homing, and in Operation Mode Automatic. It appears when activating the button [OM Homing] for homing of the entire plant or in Manual Mode [OM MANU] for a local homing via [START Home Position] of each Process Module P01 or P02. In these cases we use our default programming called SSM (Step-Switching-Mechanism).
Note: There is no way to start automatic or semi-automatic without going through homing of the entire plant.

13. FB SEQ1 P01 Sealing Robot Programme Automatic Mode

Note: P01 has a main SSM function block which calls three sub SSM function block - i.e. SEQ331, SEQ332, SEQ333. To visit the sub blocks code click the corresponding underlined function.

Network 1: Pre-setting
```
"DI MAIN P01".M00.ReadyToStart := NOT "DI MAIN P01".M00.HomePositionRunning AND "DI MAIN GLB".M00.AutomaticEntirePlant_run;
#ProductionDone(IN := NOT "E00 M100 Plate Input" AND "DI MAIN GLB".M00.AutomaticEntirePlant_run,
                PT := "DB HMI KEY".Engineering_P01.Set_Stop_pulse_duration);
"DB HMI KEY".Engineering_P01.Actual_pulse_duration := #ProductionDone.ET;
IF "DI MAIN GLB".M00.AutomaticEntirePlant_run AND #ProductionDone.Q AND "DI MAIN P02".to_P01.Done AND "DI OUT P01".M100.FEEDBACK_WP THEN
    "DI MAIN GLB".M00.AutomaticEntirePlant_run := FALSE;
END_IF;
```

Network 2: Define last step
```
#SSM(LAST_STEP := 6);
```

Network 3: Step Switching Mechanism (SSM)
```
CASE #SSM.S.stepact OF
    0: // Waiting for SEQ to start
        #RunTimeCounter := 0;
        "DI OUT P01".CAMERA.Recording.Product_A := "DI OUT P01".CAMERA.Recording.Product_B := "DI OUT P01".CAMERA.Recording.Product_C :=
        "DI MAIN P02".to_P01.EnableToRun := "DI OUT P01".CAMERA.Recording.Done := "DI MAIN P01".M00.SEQ1_Busy := FALSE;
        "DI OUT P01".M100.DRV.Execute_WP := "DB HMI KEY".P01.M100.F12_Enable_WP AND "DI MAIN P01".M00.ReadyToStart AND #SSM.S.stepa;
        #SSM.S.stepd := #SSM.S.steps AND "DI MAIN P02".to_P01.Request AND "DI OUT P01".M100.FEEDBACK_WP;

    1: // Waiting for plate in-out available
        IF #SSM.S.stepfc THEN
            #SSM.S.c1.start := TRUE;
        ELSIF "DI OUT P01".M100.FEEDBACK_WP THEN
            IF #M100_Flag_12 AND #M100_Flag_22 AND NOT "E00 M100 Plate output" THEN
                "DI OUT P01".M100.DRV.Execute_WP := #M100_Flag_12 := #M100_Flag_22 := FALSE;
                #SSM.S.stepd := TRUE;
            ELSIF NOT "E00 M100 Plate output" AND "E00 M100 Plate Input" THEN
                #M100_Flag_12 := TRUE;
            ELSIF #M100_Flag_12 AND NOT "E00 M100 Plate Input" AND "E00 M100 Plate output" THEN
                #M100_Flag_22 := TRUE;
            END_IF;
        END_IF;
        #SSM.S.stepllo := "DI OUT P01".CAMERA.Recording.CEA_code <> 16#0000_0000 AND
        ("DI OUT P01".CAMERA.Recording.Product_A XOR "DI OUT P01".CAMERA.Recording.Product_B) XOR "DI OUT P01".CAMERA.Recording.Product_C;
        "DI OUT P01".CAMERA.Execute := NOT #SSM.S.stepd AND "DI OUT P01".M100.DRV.Lamp_ON AND "E00 M100 Plate Input";

    2: // Query CAMERA.Recording.Done
        IF #SSM.S.stepfc THEN
            "DI MAIN P01".M00.SEQ1_Busy := TRUE;
        ELSIF "DI OUT P01".CAMERA.Recording.Product_A THEN
            #SSM.S.stepd := "DI OUT P01".CAMERA.Recording.Done AND #SSM.S.stepllo;
            "DB HMI KEY".Engineering_P01.ROBOT.CEA_Code.Product_A := "DI OUT P01".CAMERA.Recording.CEA_code;
        ELSIF "DI OUT P01".CAMERA.Recording.Product_B THEN
            #SSM.S.stepnew := 4;
            #SSM.S.stepd := "DI OUT P01".CAMERA.Recording.Done AND #SSM.S.stepllo;
            "DB HMI KEY".Engineering_P01.ROBOT.CEA_Code.Product_B := "DI OUT P01".CAMERA.Recording.CEA_code;
        ELSIF "DI OUT P01".CAMERA.Recording.Product_C THEN
            #SSM.S.stepnew := 5;
            #SSM.S.stepd := "DI OUT P01".CAMERA.Recording.Done AND #SSM.S.stepllo;
            "DB HMI KEY".Engineering_P01.ROBOT.CEA_Code.Product_C := "DI OUT P01".CAMERA.Recording.CEA_code;
        END_IF;

    3: // Sealing cover A
        #SEQ331.SSM.S.steps := NOT ("DI MAIN P01".M00.HomePositionRunning OR #SEQ331.Done);
        #SEQ331.SSM.S.steph := "DI MAIN P01".SEQ1.SSM.S.steph;
        #SEQ331.SSM.S.stepa := "DI MAIN P01".M00.ReadyToStart;
        #SEQ331();
        IF #SEQ331.SSM.S.step0 AND NOT #SSM.S.steplo AND "DI MAIN P01".M00.ReadyToStart THEN
            #SSM.S.stepnew := 6;
            #SSM.S.stepd := TRUE;
            #SEQ331.Done := FALSE;
            "DI MAIN P01".to_P02.EnableToRun := TRUE;
        END_IF;

    4: // Sealing cover B
        #SEQ332.SSM.S.steps := NOT ("DI MAIN P01".M00.HomePositionRunning OR #SEQ332.Done);
        #SEQ332.SSM.S.steph := "DI MAIN P01".SEQ1.SSM.S.steph;
        #SEQ332.SSM.S.stepa := "DI MAIN P01".M00.ReadyToStart;
        #SEQ332();
        IF #SEQ332.SSM.S.step0 AND NOT #SSM.S.steplo AND "DI MAIN P01".M00.ReadyToStart THEN
            #SSM.S.stepnew := 6;
            #SSM.S.stepd := TRUE;
            #SEQ332.Done := FALSE;
            "DI MAIN P01".to_P02.EnableToRun := TRUE;
        END_IF;

    5: // Sealing cover C
        #SEQ333.SSM.S.steps := NOT ("DI MAIN P01".M00.HomePositionRunning OR #SEQ333.Done);
        #SEQ333.SSM.S.steph := "DI MAIN P01".SEQ1.SSM.S.steph;
        #SEQ333.SSM.S.stepa := "DI MAIN P01".M00.ReadyToStart;
        #SEQ333();
```

```
        IF #SEQ333.SSM.S.step0 AND NOT #SSM.S.steplo AND "DI MAIN P01".M00.ReadyToStart THEN
            #SSM.S.stepd := TRUE;
            #SEQ333.Done := FALSE;
            "DI MAIN P01".to_P02.EnableToRun := TRUE;
        END_IF;

    6: // Execute Object Transfer P01 to P02
        "DI OUT P01".M100.DRV.Execute_WP := "DB HMI KEY".P01.M100.F12_Enable_WP AND #SSM.S.stepa;
        IF "DI MAIN P02".to_P01.EnableToRun AND "DI OUT P02".G120.DRV.Support.Lamp_ON AND "DI OUT P01".M100.FEEDBACK_WP THEN
            #SSM.S.stepd := TRUE;
        END_IF;
END_CASE
```

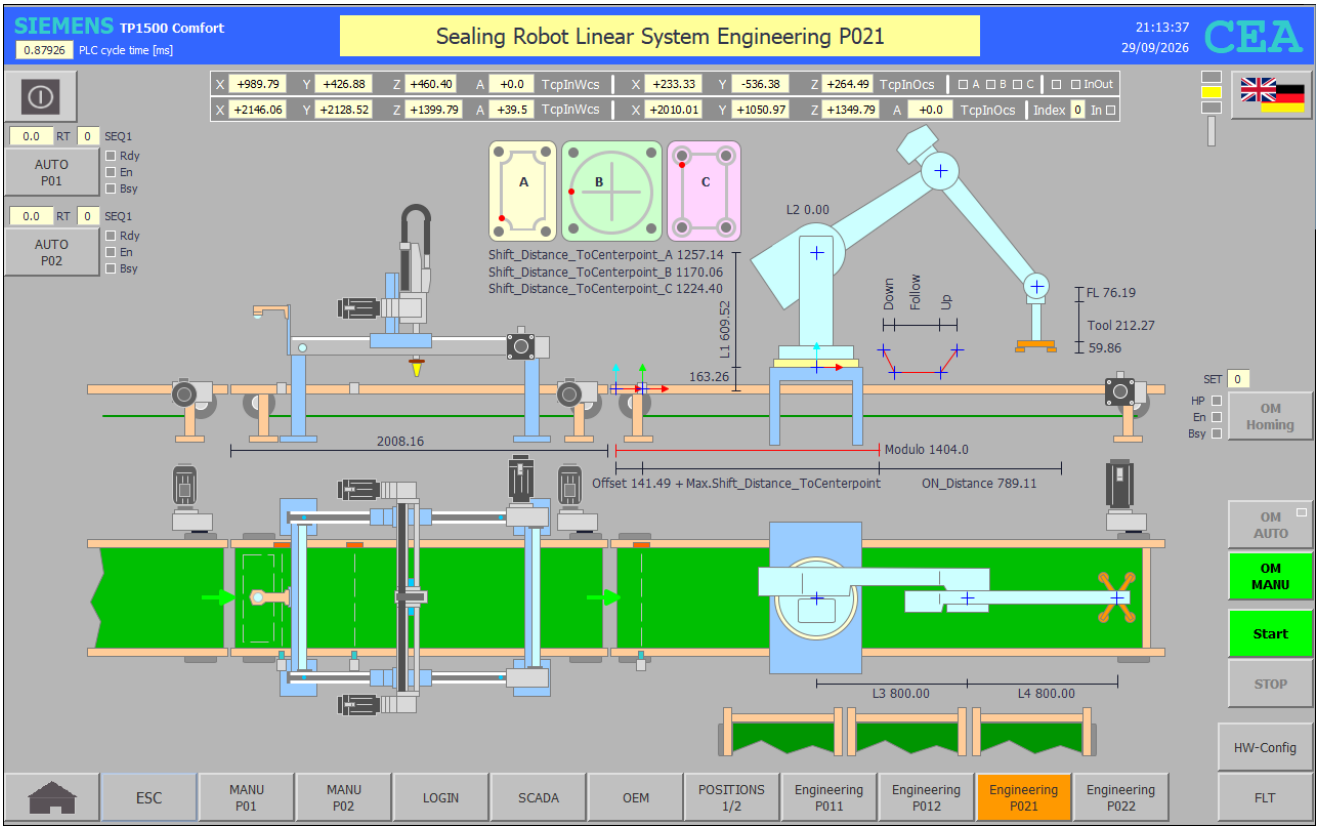
14. FB SEQ1 P02 Pick & Place Robot Linear Programme Automatic Mode

Depending on the camera recording result (product-type, A/B or C) the function block calculates distances the product has to travel as well as distances to the pick-up centre point of the coming protective cover.

15. Track Conveyor explained

Engineering P021 shows on the right side the calculated distance required to implement <Down-Follow-Up> motion of conveyor tracking. This distance must be within Object-Coordinate-System (OCS1), and is shorter or equal to the parameter modulo-length. The calculated distance involves Object-Offset + Shift-Distance to object central point. This can vary within OCS1.

During transport from one robot to another the protective covers must cross a light barrier. To ensure a complete transfer into the robot measuring sector the conveyor position is copied exactly after fully crossing the light barrier.



Network 1: Pre-setting

```
"DI MAIN P02".M00.ReadyToStart := NOT "DI MAIN P02".M00.HomePositionRunning AND "DI MAIN GLB".M00.AutomaticEntirePlant_run;
```

Network 2: Define last step

```
#SSM(LAST_STEP := 12);
```

Network 3: Step Switching Mechanism (SSM)

```
CASE #SSM.S.stepact OF
    0: // Waiting for SEQ to start
        #num := #RunTimeCounter := 0;
        "DI MAIN P02".M00.SEQ1_Busy := FALSE;
        "DI MAIN P02".to_P01.Request := "DI MAIN P02".M00.ReadyToStart;
        #SSM.S.stepd := #SSM.S.steps AND "DI MAIN P01".to_P02.EnableToRun;

    1: // G120 MoveJoge Sealing Plate Output Conveyor Start
        IF #SSM.S.stepfc THEN
            "DI MAIN P02".to_P01.Done := FALSE;
            "DI OUT P02".TRACK_CONVEYOR.DRV.MC_MeasuringInput.Execute :=
            "DI OUT P02".TRACK_CONVEYOR.DRV.Support.Enable_ReMeasuring := "DI MAIN P02".M00.SEQ1_Busy := TRUE;
            "DI OUT P02".G120.DRV.Support.SetVelocity := "DB HMI KEY".Engineering_P02.ROBOT.Velocity_Middle;
        ELSIF #SSM.S.stepeg AND NOT "E00 G120 Plate input" THEN
            "DI MAIN P02".to_P01.EnableToRun :=
            #SSM.S.stepd := "DI MAIN P02".M00.SEQ1_Busy := "DI OUT P02".TRACK_CONVEYOR.DRV.MC_MeasuringInput.Done;
        END_IF;
        #SSM.S.stepeg := "E00 G120 Plate input" AND "DI OUT P02".G120.DRV.Support.Lamp_ON AND "DI OUT P01".M100.FEEDBACK_WP;
        "DI OUT P02".G120.DRV.Support.MoveJogWP := "DB HMI KEY".P02.G120.F12_Enable_WP AND #SSM.S.stepa;

    2: // ROBOT Pre-Setting. If need be, MoveLinearAbsolut to Start Position UP (WCS)
        IF #SSM.S.stepfc THEN
            #num := 1;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.DynamicAdaption := -1;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.BufferMode := 1;
            "DI OUT P02".TRACK_CONVEYOR.DRV.MC_MeasuringInput.Execute := "DI MAIN P02".to_P01.Request := FALSE;
            "DI OUT P02".TRACK_CONVEYOR.DRV.MC_TrackConveyorBelt.ConveyorBeltOrigin := "DB HMI KEY".Engineering_P02.Track_Conveyor.ConveyorBeltOrigin;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.CoordSystem := 0;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.DynamicAdaption := 1;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Velocity := "DB HMI KEY".Engineering_P02.ROBOT.Velocity_Fast;
            IF "DB HMI KEY".Engineering_P01.Object_ABC[#Index].Product_A THEN
                #PickUp.Product_A := TRUE;
                #PickUp.Product_B := #PickUp.Product_C := FALSE;
                "DB HMI KEY".Engineering_P02.Track_Conveyor.Distance_to_Belt_Down_z := "DB HMI KEY".Engineering_P02.ROBOT.PlacePosition_A[3];
                "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".ROBOT_MoveTo.Target_P02[#num].Position;
                "DI OUT P02".TRACK_CONVEYOR.DRV.Support.Measuring_Shift_Distance :=
                "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[1] :=
                "DB HMI KEY".Engineering_P02.Track_Conveyor.Shift_Distance_ToCenterpoint_A + "DB HMI KEY".Engineering_P02.Track_Conveyor.Object_Offset_x;
            ELSIF "DB HMI KEY".Engineering_P01.Object_ABC[#Index].Product_B THEN
                #PickUp.Product_B := TRUE;
                #PickUp.Product_A := #PickUp.Product_C := FALSE;
                "DB HMI KEY".Engineering_P02.Track_Conveyor.Distance_to_Belt_Down_z := "DB HMI KEY".Engineering_P02.ROBOT.PlacePosition_B[3];
                "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".ROBOT_MoveTo.Target_P02[#num].Position;
                "DI OUT P02".TRACK_CONVEYOR.DRV.Support.Measuring_Shift_Distance :=
                "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[1] :=
                "DB HMI KEY".Engineering_P02.Track_Conveyor.Shift_Distance_ToCenterpoint_B + "DB HMI KEY".Engineering_P02.Track_Conveyor.Object_Offset_x;
            ELSIF "DB HMI KEY".Engineering_P01.Object_ABC[#Index].Product_C THEN
                #PickUp.Product_C := TRUE;
                #PickUp.Product_A := #PickUp.Product_B := FALSE;
                "DB HMI KEY".Engineering_P02.Track_Conveyor.Distance_to_Belt_Down_z := "DB HMI KEY".Engineering_P02.ROBOT.PlacePosition_C[3];
                "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".ROBOT_MoveTo.Target_P02[#num].Position;
                "DI OUT P02".TRACK_CONVEYOR.DRV.Support.Measuring_Shift_Distance :=
                "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[1] :=
                "DB HMI KEY".Engineering_P02.Track_Conveyor.Shift_Distance_ToCenterpoint_C + "DB HMI KEY".Engineering_P02.Track_Conveyor.Object_Offset_x;
            END_IF;
            IF #Index < "DI OUT P01".CAMERA.Array_upper_bound THEN
                #Index += 1;
            ELSE
                #Index := "DI OUT P01".CAMERA.Array_lower_bound;
            END_IF;
        ELSE
            #SSM.S.stepd := "DI OUT P02".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P02".ROBOT.DRV.Support.MoveLinearAbsolute := "DI OUT P02".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    3: // G120 MoveJoge Selected Plate towards Pickup Position
        IF ABS("DI OUT P02".G120.DRV.Support.ActualPosition - "DI OUT P02".TRACK_CONVEYOR.DRV.Support.InitialObjectPosition_x) <
            "DB HMI EY".Engineering_P02.Track_Conveyor.Window THEN
            #SSM.S.stepd := TRUE;
        END_IF;

    4: // TRACK_CONVEYOR MC_TrackConveyorBelt.Execute
        IF #SSM.S.stepfc THEN
            "DI OUT P02".TRACK_CONVEYOR.DRV.MC_MeasuringInput.Execute := FALSE;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.CoordSystem := 1;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.DynamicAdaption := 0; // no adaption
            "DI OUT P02".TRACK_CONVEYOR.DRV.MC_TrackConveyorBelt.Execute := TRUE;
        ELSE
            #SSM.S.stepd := "DI OUT P02".TRACK_CONVEYOR.DRV.MC_TrackConveyorBelt.Done;
        END_IF;

    5: // ROBOT MoveLinearAbsolute.Active Synchron DOWN (OCS1)
        IF #SSM.S.stepfc THEN
            #num := 2;
            #SSM.S.c1.start := TRUE;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[2] := "DB HMI KEY".ROBOT_MoveTo.Target_P02[#num].Position[2];
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[3] := "DB HMI KEY".Engineering_P02.Track_Conveyor.Distance_to_Belt_Down_z;
        ELSE
            #SSM.S.stepd := "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Active AND NOT #SSM.S.steplo;
            "DI OUT P02".ROBOT.DRV.Support.MoveLinearAbsolute :=
            "DI OUT P02".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT #SSM.S.stepd AND #SSM.S.stepa AND #SSM.S.c1.done;
        END_IF;

    6: // ROBOT MoveLinearAbsolute.Active Synchron FOLLOW (OCS1)
        IF #SSM.S.stepfc THEN
            #SSM.S.c1.start := TRUE;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[1] += "DB HMI KEY".Engineering_P02.Track_Conveyor.Object_Follow_on_x;
        ELSE
            #SSM.S.stepd := "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Active AND NOT #SSM.S.steplo;
            "DI OUT P02".ROBOT.DRV.Support.MoveLinearAbsolute :=
            "DI OUT P02".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT #SSM.S.stepd AND #SSM.S.stepa AND #SSM.S.c1.done;
            "DI OUT P02".Y201.VLV_O.Execute_WP := "DB HMI KEY".P02.Y201.F12_Enable_WP;
        END_IF;

    7: // ROBOT MoveLinearAbsolute Synchron UP (OCS1) 1 synchronous UP
        IF #SSM.S.stepfc THEN
            #num := 3;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[1] += "DB HMI KEY".ROBOT_MoveTo.Target_P02[#num].Position[1];
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[3] := "DB HMI KEY".Engineering_P02.Track_Conveyor.Distance_to_Belt_Up_z;
        ELSE
            #SSM.S.stepd := "DI OUT P02".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P02".ROBOT.DRV.Support.MoveLinearAbsolute := "DI OUT P02".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    8: // ROBOT MoveLinearAbsolute Selected Plate towards Placing UP (WCS)
        IF #SSM.S.stepfc THEN
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.CoordSystem := 0;
            IF #PickUp.Product_A THEN
```

```
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".Engineering_P02.ROBOT.PlacePosition_A;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[3] := "DB HMI KEY".Engineering_P02.Track_Conveyor.Distance_to_Belt_Up_z;
        ELSIF #PickUp.Product_B THEN
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".Engineering_P02.ROBOT.PlacePosition_B;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[3] := "DB HMI KEY".Engineering_P02.Track_Conveyor.Distance_to_Belt_Up_z;
        ELSIF #PickUp.Product_C THEN
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".Engineering_P02.ROBOT.PlacePosition_C;
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[3] := "DB HMI KEY".Engineering_P02.Track_Conveyor.Distance_to_Belt_Up_z;
        END_IF;
    END_IF;
    #SSM.S.stepd := "DI OUT P02".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
    "DI OUT P02".ROBOT.DRV.Support.MoveLinearAbsolute := "DI OUT P02".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT #SSM.S.stepd AND #SSM.S.stepa;

9: // ROBOT MoveLinearAbsolute Selected Plate DOWN (WCS)
    IF #SSM.S.stepfc THEN
        "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Velocity := "DB HMI KEY".Engineering_P02.ROBOT.Velocity_Middle;
        IF #PickUp.Product_A THEN
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".Engineering_P02.ROBOT.PlacePosition_A;
        ELSIF #PickUp.Product_B THEN
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".Engineering_P02.ROBOT.PlacePosition_B;
        ELSIF #PickUp.Product_C THEN
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".Engineering_P02.ROBOT.PlacePosition_C;
        END_IF;
    END_IF;
    #SSM.S.stepd := "DI OUT P02".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
    "DI OUT P02".ROBOT.DRV.Support.MoveLinearAbsolute := "DI OUT P02".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT #SSM.S.stepd AND #SSM.S.stepa;

10: // Y201 Pick & Place Gripper Placing Selected Plate
    "DI OUT P02".Y201.VLV_O.Execute_WP := FALSE;
    #SSM.S.cl.set := 10;
    #SSM.S.cl.start := "DI OUT P02".Y201.FEEDBACK_HP;
    #SSM.S.stepd := #SSM.S.cl.done;

11: // ROBOT MoveLinearAbsolute Selected Plate UP (WCS)
    IF #SSM.S.stepfc THEN
        "DI MAIN P01".to_P02.EnableToRun := FALSE;
        "DI MAIN P02".to_P01.Request := TRUE;
        "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[3] := "DB HMI KEY".Engineering_P02.Track_Conveyor.Distance_to_Belt_Up_z;
        "DI MAIN P02".M00.ProductCounter.GoodParts += 1;
    ELSE
        #SSM.S.stepd := "DI OUT P02".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo AND
        ("DB HMI KEY".Engineering_P01.Object_ABC[#Index].Product_A OR "DB HMI KEY".Engineering_P01.Object_ABC[#Index].Product_B OR
        "DB HMI KEY".Engineering_P01.Object_ABC[#Index].Product_C);
        "DI OUT P02".ROBOT.DRV.Support.MoveLinearAbsolute := "DI OUT P02".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT #SSM.S.stepd AND #SSM.S.stepa;
    END_IF;

12: // ROBOT MoveLinearAbsolute towards Start Position UP
    IF #SSM.S.stepfc THEN
        #num := 4;
        "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Velocity := "DB HMI KEY".Engineering_P02.ROBOT.Velocity_Fast;
        "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".ROBOT_MoveTo.Target_P02[#num].Position;
        IF "DB HMI KEY".Engineering_P01.Object_ABC[#Index].Product_A THEN
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[1] :=
            "DB HMI KEY".Engineering_P02.Track_Conveyor.Shift_Distance_ToCenterpoint_A + "DB HMI KEY".Engineering_P02.Track_Conveyor.Object_Offset_x;
        ELSIF "DB HMI KEY".Engineering_P01.Object_ABC[#Index].Product_B THEN
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[1] :=
            "DB HMI KEY".Engineering_P02.Track_Conveyor.Shift_Distance_ToCenterpoint_B + "DB HMI KEY".Engineering_P02.Track_Conveyor.Object_Offset_x;
        ELSIF "DB HMI KEY".Engineering_P01.Object_ABC[#Index].Product_C THEN
            "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[1] :=
            "DB HMI KEY".Engineering_P02.Track_Conveyor.Shift_Distance_ToCenterpoint_C + "DB HMI KEY".Engineering_P02.Track_Conveyor.Object_Offset_x;
        ELSIF "DI OUT P02".ROBOT.DRV.MC_MoveLinearAbsolute.Position[1] = 0.0 THEN
            #SSM.S."1_FAULT_00_15".%X0 := TRUE;
        END_IF;
    ELSE
        "DI MAIN P02".to_P01.Done :=
        #SSM.S.stepd := "DI OUT P02".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
        "DI OUT P02".ROBOT.DRV.Support.MoveLinearAbsolute := "DI OUT P02".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT #SSM.S.stepd AND #SSM.S.stepa;
    END_IF;
END_CASE;
```

16. Sub Function SEQ331

Network 1: Define last step

```
#SSM(LAST_STEP := 10);
```

Network 2: Step Switching Mechanism (SSM)

```
CASE #SSM.S.stepact OF
    0: // Waiting for SEQ to start
        "DI OUT P01".V101.VLV_O.Execute_WP := FALSE;
        #SSM.S.stepd := #SSM.S.steps AND "DI OUT P01".V101.FEEDBACK_HP;
        #num := -1;

    1: // ROBOT MoveLinearAbsolute towards OCS1 start position in WCS
        IF #SSM.S.stepfc THEN
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.BufferMode := 1;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.CoordSystem := 0;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.CoordSystem := "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.CoordSystem := 1;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".ROBOT_MoveTo.Target_P01[1].Position;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Velocity :=
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.Velocity := "DB HMI KEY".Engineering_P01.ROBOT.Velocity_Work;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.Velocity := "DB HMI KEY".Engineering_P01.ROBOT.Velocity_Work;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearAbsolute := "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    2: // 01 ROBOT MC_MoveLinearRelative in OCS1
        IF #SSM.S.stepfc THEN
            #num := 0;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_A.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".V101.VLV_O.Execute_WP := "DB HMI KEY".P01.V101.F12_Enable_WP AND #SSM.S.stepa;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_HP) AND #SSM.S.stepa;
        END_IF;

    3: // 12 ROBOT MC_MoveCircularRelative CircMode 2 + Radius in OCS1
        IF #SSM.S.stepfc THEN
            #num := 1;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.BufferMode := 1;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.CircMode := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_A.Target[#num].CircMode;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.PathChoice := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_A.Target[#num].PathChoice;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.Radius := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_A.Target[#num].Radius;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.EndPoint := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_A.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveCircularRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND "DI OUT P01".V101.FEEDBACK_WP AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    4: // 23 ROBOT MC_MoveLinearRelative in OCS1
        IF #SSM.S.stepfc THEN
            #num := 2;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_A.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND "DI OUT P01".V101.FEEDBACK_WP AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    5: // 34 ROBOT MC_MoveCircularRelative CircMode 2 + Radius in OCS1
        IF #SSM.S.stepfc THEN
            #num := 3;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.EndPoint := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_A.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveCircularRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND "DI OUT P01".V101.FEEDBACK_WP AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    6: // 45 ROBOT MC_MoveLinearRelative in OCS1
        IF #SSM.S.stepfc THEN
            #num := 4;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_A.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND "DI OUT P01".V101.FEEDBACK_WP AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    7: // 56 ROBOT MC_MoveCircularRelative CircMode 2 + Radius in OCS1
        IF #SSM.S.stepfc THEN
            #num := 5;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.EndPoint := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_A.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveCircularRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND "DI OUT P01".V101.FEEDBACK_WP AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    8: // 67 ROBOT MC_MoveLinearRelative in OCS1
        IF #SSM.S.stepfc THEN
            #num := 6;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_A.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND "DI OUT P01".V101.FEEDBACK_WP AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    9: // 70 ROBOT MC_MoveCircularRelative CircMode 2 + Radius in OCS1
        IF #SSM.S.stepfc THEN
            #num := 7;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.EndPoint := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_A.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveCircularRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND "DI OUT P01".V101.FEEDBACK_WP AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    10: // V101 Sealing Compound Outlet closed
        "DI OUT P01".V101.VLV_O.Execute_WP := FALSE;
        #Done := #SSM.S.stepd := "DI OUT P01".V101.FEEDBACK_HP;
END_CASE;
```

17. Sub Function SEQ332

Network 1: Define last step

```
#SSM(LAST_STEP := 7);
```

Network 2: Step Switching Mechanism (SSM)

```
CASE #SSM.S.stepact OF
    0: // Waiting for SEQ to start
        "DI OUT P01".V101.VLV_O.Execute_WP := FALSE;
        #SSM.S.stepd := #SSM.S.steps AND "DI OUT P01".V101.FEEDBACK_HP;
        #num := -1;

    1: // ROBOT MoveLinearAbsolute towards OCS2 start position in WCS
        IF #SSM.S.stepfc THEN
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.BufferMode := 1;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.CoordSystem := 0;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.CoordSystem := "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.CoordSystem := 2;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".ROBOT_MoveTo.Target_P01[2].Position;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Velocity :=
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.Velocity := "DB HMI KEY".Engineering_P01.ROBOT.Velocity_Work;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.Velocity := "DB HMI KEY".Engineering_P01.ROBOT.Velocity_Work;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearAbsolute :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".M100.FEEDBACK) AND #SSM.S.stepa;
        END_IF;

    2: // 00 ROBOT MC_MoveCircularRelative CircMode 1 + Arc in OCS2
        IF #SSM.S.stepfc THEN
            #num := 0;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.BufferMode := 1;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.CircMode := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_B.Target[#num].CircMode;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.PathChoice := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_B.Target[#num].PathChoice;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.Arc := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_B.Target[#num].Arc; // 359.99°
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.AuxPoint := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_B.Target[#num].AuxPoint;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".V101.VLV_O.Execute_WP := "DB HMI KEY".P01.V101.F12_Enable_WP AND #SSM.S.stepa;
            "DI OUT P01".ROBOT.DRV.Support.MoveCircularRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_HP) AND #SSM.S.stepa;
        END_IF;

    3: // 01 ROBOT MC_MoveLinearRelative in OCS2
        IF #SSM.S.stepfc THEN
            #num := 1;
            "DI OUT P01".V101.VLV_O.Execute_WP := FALSE;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_B.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_WP) AND #SSM.S.stepa;
        END_IF;

    4: // 12 ROBOT MC_MoveLinearRelative in OCS2
        IF #SSM.S.stepfc THEN
            #num := 2;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_B.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".V101.VLV_O.Execute_WP := "DB HMI KEY".P01.V101.F12_Enable_WP AND #SSM.S.stepa;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_HP) AND #SSM.S.stepa;
        END_IF;

    5: // 23 ROBOT MC_MoveLinearRelative in OCS2
        IF #SSM.S.stepfc THEN
            #num := 3;
            "DI OUT P01".V101.VLV_O.Execute_WP := FALSE;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_B.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_WP) AND #SSM.S.stepa;
        END_IF;

    6: // 34 ROBOT MC_MoveLinearRelative in OCS2
        IF #SSM.S.stepfc THEN
            #num := 4;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_B.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".V101.VLV_O.Execute_WP := "DB HMI KEY".P01.V101.F12_Enable_WP AND #SSM.S.stepa;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_HP) AND #SSM.S.stepa;
        END_IF;

    7: // V101 Sealing Compound Outlet closed
        "DI OUT P01".V101.VLV_O.Execute_WP := FALSE;
        #Done := #SSM.S.stepd := "DI OUT P01".V101.FEEDBACK_HP;
END_CASE;
```

18. Sub Function SEQ333

Network 1: Define last step

```
#SSM(LAST_STEP := 14);
```

Network 2: Step Switching Mechanism (SSM)

```
CASE #SSM.S.stepact OF
    0: // Waiting for SEQ to start
        #SSM.S.stepd := #SSM.S.steps;
        #num := -1;

    1: // ROBOT MoveLinearAbsolute towards OCS3 start position in WCS
        IF #SSM.S.stepfc THEN
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.BufferMode := 1;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.CoordSystem := 0;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.CoordSystem := "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.CoordSystem := 3;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.Position := "DB HMI KEY".ROBOT_MoveTo.Target_P01[3].Position;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Velocity :=
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearAbsolute.Velocity := "DB HMI KEY".Engineering_P01.ROBOT.Velocity_Work;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.Velocity := "DB HMI KEY".Engineering_P01.ROBOT.Velocity_Work;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearAbsolute :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".M100.FEEDBACK) AND #SSM.S.stepa;
        END_IF;

    2: // 00 ROBOT MC_MoveCircularRelative CircMode 1 + Arc in OCS3
        IF #SSM.S.stepfc THEN
            #num := 0;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.BufferMode := 1;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.CircMode := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].CircMode;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.PathChoice := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].PathChoice;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.Arc := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].Arc;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.AuxPoint := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].AuxPoint;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".V101.VLV_O.Execute_WP := "DB HMI KEY".P01.V101.F12_Enable_WP AND #SSM.S.stepa;
            "DI OUT P01".ROBOT.DRV.Support.MoveCircularRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_HP) AND #SSM.S.stepa;
        END_IF;

    3: // 01 ROBOT MC_MoveLinearRelative in OCS3
        IF #SSM.S.stepfc THEN
            #num := 1;
            "DI OUT P01".V101.VLV_O.Execute_WP := FALSE;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_WP) AND #SSM.S.stepa;
        END_IF;

    4: // 12 ROBOT MC_MoveLinearRelative in OCS3
        IF #SSM.S.stepfc THEN
            #num := 2;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".V101.VLV_O.Execute_WP := "DB HMI KEY".P01.V101.F12_Enable_WP AND #SSM.S.stepa;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative := "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    5: // 22 ROBOT MC_MoveCircularRelative CircMode 1 + Arc in OCS3
        IF #SSM.S.stepfc THEN
            #num := 3;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.AuxPoint := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].AuxPoint;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveCircularRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND "DI OUT P01".V101.FEEDBACK_WP AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    6: // 23 ROBOT MC_MoveLinearRelative OCS3
        IF #SSM.S.stepfc THEN
            #num := 4;
            "DI OUT P01".V101.VLV_O.Execute_WP := FALSE;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_WP) AND #SSM.S.stepa;
        END_IF;

    7: // 34 ROBOT MC_MoveLinearRelative OCS3
        IF #SSM.S.stepfc THEN
            #num := 5;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".V101.VLV_O.Execute_WP := "DB HMI KEY".P01.V101.F12_Enable_WP AND #SSM.S.stepa;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_HP) AND #SSM.S.stepa;
        END_IF;

    8: // 44 ROBOT MC_MoveCircularRelative CircMode 1 + Arc in OCS3
        IF #SSM.S.stepfc THEN
            #num := 6;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.AuxPoint := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].AuxPoint;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveCircularRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND "DI OUT P01".V101.FEEDBACK_WP AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    9: // 45 ROBOT MC_MoveLinearRelative OCS3
        IF #SSM.S.stepfc THEN
            #num := 7;
            "DI OUT P01".V101.VLV_O.Execute_WP := FALSE;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_WP) AND #SSM.S.stepa;
        END_IF;

    10: // 56 ROBOT MC_MoveLinearRelative OCS3
        IF #SSM.S.stepfc THEN
            #num := 8;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".V101.VLV_O.Execute_WP := "DB HMI KEY".P01.V101.F12_Enable_WP AND #SSM.S.stepa;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
```

```
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_HP) AND #SSM.S.stepa;
        END_IF;

    11: // 66 ROBOT MC_MoveCircularRelative CircMode 1 + Arc in OCS3
        IF #SSM.S.stepfc THEN
            #num := 9;
            "DI OUT P01".ROBOT.DRV.MC_MoveCircularRelative.AuxPoint := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].AuxPoint;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveCircularRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND "DI OUT P01".V101.FEEDBACK_WP AND NOT #SSM.S.stepd AND #SSM.S.stepa;
        END_IF;

    12: // 67 ROBOT MC_MoveLinearRelative OCS3
        IF #SSM.S.stepfc THEN
            #num := 10;
            "DI OUT P01".V101.VLV_O.Execute_WP := FALSE;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_WP) AND #SSM.S.stepa;
        END_IF;

    13: // 70 ROBOT MC_MoveLinearRelative OCS3
        IF #SSM.S.stepfc THEN
            #num := 11;
            "DI OUT P01".ROBOT.DRV.MC_MoveLinearRelative.Distance := "DB HMI KEY".Engineering_P01.ROBOT.MoveTo_C.Target[#num].Position;
        ELSE
            #SSM.S.stepd := "DI OUT P01".ROBOT.DRV.Support.InPosition AND NOT #SSM.S.steplo;
            "DI OUT P01".V101.VLV_O.Execute_WP := "DB HMI KEY".P01.V101.F12_Enable_WP AND #SSM.S.stepa;
            "DI OUT P01".ROBOT.DRV.Support.MoveLinearRelative :=
            "DI OUT P01".ROBOT.DRV.Support.KINEMATICS_Enabled AND NOT (#SSM.S.stepd OR "DI OUT P01".V101.FEEDBACK_HP) AND #SSM.S.stepa;
        END_IF;

    14: // V101 Sealing Compound Outlet closed
        "DI OUT P01".V101.VLV_O.Execute_WP := FALSE;
        #Done := #SSM.S.stepd := "DI OUT P01".V101.FEEDBACK_HP;
END_CASE;
```

19. Learning from - and about

- S71500T Kinematics Function Manual V19
- S71500 S71500T Alarms Errors Diagnostics Manual V8
- S71500 S71500T Motion Control Overview Function Manual V19

20. Project Videos

Sealing_Robot_Linear_System_P01.mp4
Sealing_Robot_Linear_System_P02.mp4