

```
1  PROGRAM PLC_PRG
2  VAR
3      state : UINT ;
4      // Timer used to simulate process
5      Timer_RFID : TON ;
6      time_from_RFID : TIME ;
7      // Timer for ramping up the conveyor
8      Timer_ramp : TON ;
9      RFID : dc_ecp.Dtm425 ;
10     data_RFID : dc_ecp.st_CPF_RfidData ;
11     // ID from RFID tag
12     CarrierID : UINT ;
13     TCPclient : dc_ecp.TcpClientByteStream ;
14     // Data string in from TCP server
15     dataIN : STRING ;
16     // Station ID
17     ST_ID : UINT := 9 ;
18     // Data string out to TCP server
19     StringData : String ( 109 ) ;
20     DateTime : DATE_AND_TIME ;
21     DT_FB : DTU.GetDateAndTime ;
22     // Date and time in string type
23     DateTimeString : STRING ;
24 END_VAR
25
```

```
1  // Course:      Software and Automation Framework
2  // Date:        2021-11-03
3  // Project:     Mini project SAF, TCP server communication with PLC
4  // Group:       563
5  // Authors:     Axel Villads Burford Toft      - atoft19@student.aau.dk
6  //              Jonathan Falk                  -
7  //              jfalk19@student.aau.dk
8  //              Simon Sunesen Gaasdal          -
9  //              sgaasd19@student.aau.dk
10 //              Vinh Quang Nguyen              -
11 //              vnguyel6@student.aau.dk
12 //              Vinicius Soares Matthiesen    -
13 //              vmatth19@student.aau.dk
14
15 CASE state OF
16
17     0 : // Start conveyor and reset piston
18         // Reset RFID reader and start TCP client
19         IO.xMB20 := FALSE ;
20         IO.xQA1_RIGHT := TRUE ;
21         RFID.ClearError ( ) ;
22         TCPclient.Connect ( sIP := '172.20.66.64' , uiPort := 8080 ) ;
23         state := 100 ;
24
```

```
22
23     100 : // Wait for RFID tag to clear error, then connect
24     IF RFID.xReady = TRUE THEN
25         RFID.Connect ( usiNodeId := 32 , usiNetworkId := 0 , usiChannel
:= 1 ) ;
26         state := 101 ;
27     ELSE
28         RFID.ClearError ( ) ;
29     END_IF
30
31     101 : // Wait for RFID tag to connect
32     IF TCPclient.xReady AND TCPclient.xConnected THEN
33         state := 1 ;
34     END_IF
35
36
37
38     1 : // Slow conveyor at 3. sensor
39     IF IO.xBG23 = TRUE THEN
40         IO.xQA1_SLOW := TRUE ;
41         state := 2 ;
42     END_IF
43
44     2 : // Reading RFID tag
45     IF IO.xBG21 = TRUE THEN
46         RFID.ReadTag ( uiStartAddress := 0 , uiDataLength := SIZEOF (
data_RFID ) , pData := ADR ( data_RFID ) ) ;
47         // Reading the Date and Time
48         DT_FB.xExecute := TRUE ;
49         state := 3 ;
50     END_IF
51
52     3 : // wait for the RFID msg to be recived, And check if the data and time
is done
53     IF RFID.xReady = TRUE AND DT_FB.xDone = TRUE THEN
54         CarrierID := dc_ecp.SwapWORD ( wWord := data_RFID.uiCarrierID ) ;
55         DateTime := DT_FB.dtDateAndTime ;
56         DateTimeString := DT_TO_STRING ( DateTime ) ;
57         // Generating "our" xml string
58         StringData := '<GROUP_563>' ;
59         StringData := CONCAT ( STR1 := StringData , STR2 := '<Station>' ) ;
60         StringData := CONCAT ( STR1 := StringData , STR2 := UINT_TO_STRING (
ST_ID ) ) ;
61         StringData := CONCAT ( STR1 := StringData , STR2 := '</Station>' ) ;
62         StringData := CONCAT ( STR1 := StringData , STR2 := '<Carrier>' ) ;
63         StringData := CONCAT ( STR1 := StringData , STR2 := UINT_TO_STRING (
CarrierID ) ) ;
64         StringData := CONCAT ( STR1 := StringData , STR2 := '</Carrier>' ) ;
65         StringData := CONCAT ( STR1 := StringData , STR2 := '<Date_Time>' ) ;
66         StringData := CONCAT ( STR1 := StringData , STR2 := DateTimeString ) ;
67         StringData := CONCAT ( STR1 := StringData , STR2 := '</Date_Time>' ) ;
```

```
68         StringData := CONCAT ( STR1 := StringData , STR2 := '</GROUP_563>' );
69
70         // Generating the data string to be send to the server
71         TCPclient . Send ( ptoSend := ADR ( StringData ) , uiSizeToSend := SIZEOF (
StringData ) ) ;
72         state := 4 ;
73     END_IF
74
75     4 : // wait for TCP to be ready to recieve data
76     IF TCPclient . xReady = TRUE THEN
77         TCPclient . Receive ( pToReceive := ADR ( dataIN ) , uiSizeToReceive :=
SIZEOF ( dataIN ) ) ;
78         state := 5 ;
79     END_IF
80
81     5 : // wait for TCP to receive data
82     IF TCPclient . xReady = TRUE THEN
83         // convert dataIN to Data Type "time":=time_from_RFID
84         time_from_RFID := INT_TO_TIME ( STRING_TO_INT ( DataIN ) ) ;
85         state := 6 ;
86     END_IF
87
88     6 : // Pause carrier for unknown time (time from server)
89     Timer_RFID ( IN := TRUE , PT := time_from_RFID ) ;
90     state := 7 ;
91
92     7 : // Proceed after time has ended
93     IF Timer_RFID . Q = TRUE THEN
94         IO . xMB20 := TRUE ;
95         Timer_RFID . IN := FALSE ;
96         state := 8 ;
97     END_IF
98
99     8 : // Start timer for ramp up speed
100    Timer_ramp ( IN := TRUE , PT := T#2S ) ;
101    state := 9 ;
102
103    9 : // Wait for ramp up to finish, andstart over
104    IF Timer_ramp . Q = TRUE THEN
105        IO . xQA1_SLOW := FALSE ;
106        IO . xMB20 := FALSE ;
107        Timer_ramp . IN := FALSE ;
108        state := 1 ;
109    END_IF
110
111    END_CASE
112
113    Timer_RFID ( ) ;
114    Timer_ramp ( ) ;
115    RFID ( ) ;
```

```
117     TCPclient ( ) ;  
118     DT_FB ( ) ;  
119  
120  
121  
122  
123
```