```
1   FUNCTION_BLOCK FB_SCHMALZ
2   VAR_INPUT
3       st_pilz :   ST_PILZ ;
4       rMechanicalAngle    : REAL ;
5       iState  :   INT ;
6       i_xVacumOK  :   BOOL ;
7   END_VAR
8   VAR_OUTPUT
9       q_xVacum    :   BOOL ;
10      q_xBlowOff  :   BOOL ;
11  END_VAR
12  VAR
13      _StartBlowOffAngle  :   REAL ;
14      _StopBlowOffAngle   :   REAL ;
15      _StartVacumAngle    :   REAL ;
16      _StopVacumANgle     :   REAL ;
17      iSstate :  INT ;
18      _stPilz             : ST_PILZ ;
19      fbVacumComtron      :  FB_isInAngleRange ;  // VACUM CONTROL
20      bVacumInRange       :   BOOL ;
21      fbBlowOffControl    :  FB_isInAngleRange ;
22      bBlowOfInRange      :   BOOL ;
23  END_VAR
24
```

```
1   //Kopiraj vrednosti iz inputa
2   _stPilz  :=  st_pilz ;
3
4   (*
5   //vklop samo v state 30 ali state 35 -> ko je sinhronizirano
6   IF iState = 30 OR iSstate = 35 THEN
7                                                   //---------------TURN ON
    ANGLE------,------ TURN OFF ANGLE -----, MECHANICAL ANGLE---//
8           P_VacumOnOff        := M_AngleControlStatus(
    _stPilz.rTurnOnAngleVacum, _stPilz.rTurnOffAngleVacum, rMechanicalAngle);
9           P_BlowOffOnOff        :=
    M_AngleControlStatus(_stPilz.rTurnOnANgleBlowOff,
    _stPilz.rTurnOffAngleBlowOff, rMechanicalAngle);
10
11  ELSE
12          P_VacumOnOff        := FALSE;
13          P_BlowOffOnOff        := FALSE;
14  END_IF
15
16  *)
17  fbVacumComtron ( rOnAngle :=  _stPilz . rTurnOnAngleVacum  ,  rOffAngle :=
    _stPilz . rTurnOffAngleVacum  ,  rMechanicalAngle := rMechanicalAngle  ,
    xisInRange => bVacumInRange  ) ;
18  fbBlowOffControl ( rOnAngle :=  _stPilz . rTurnOnANgleBlowOff ,  rOffAngle :=
    _stPilz . rTurnOffAngleBlowOff  ,  rMechanicalAngle :=  rMechanicalAngle  ,
    xisInRange =>  bBlowOfInRange  ) ;
19
20
21  //********************
22  IF  iState  =  30  OR  iState  =  35  THEN
23      //Vakum
```

```
24          IF  ( rMechanicalAngle >  st_pilz . rTurnOnAngleVacum )  AND
25              ( rMechanicalAngle <  st_pilz . rTurnOffAngleVacum )  THEN
26
27              q_xVacum  :=  NOT  TRUE ;
28              q_xBlowOff  :=  FALSE ;
29
30          //IZPIH
31          ELSIF  ( rMechanicalAngle  <=  st_pilz . rTurnOnANgleBlowOff )  AND
32              ( rMechanicalAngle  >  st_pilz . rTurnOffAngleBlowOff )  AND
33              i_xVacumOK  THEN
34                  //q_xVacum := NOT FALSE;
35                  q_xBlowOff  :=  TRUE ;
36          ELSE
37                  q_xVacum  :=  NOT  FALSE ;
38                  q_xBlowOff  :=  FALSE ;
39          END_IF ;
40
41      ELSE
42          //Izklop vakuma in BlowOffa
43          q_xVacum     :=  NOT   FALSE ;  //Ngacija izhoda
44          q_xBlowOff   :=  FALSE ;
45      END_IF
46
47      //***************
48      (*
49      IF iState = 30 OR iState = 35 THEN
50              P_VacumOnOff := bVacumInRange;
51              P_BlowOffOnOff := bBlowOfInRange AND i_xVacumOK;
52      ELSE
53              P_VacumOnOff := FALSE;
54              P_BlowOffOnOff := FALSE;
55      END_IF
56      *)
57
```