

## Project Documentation

File:	Barman.project
Date:	07.05.2019
Profile:	CODESYS V3.5 SP9 Patch 7

## Table of Contents

1 Device: Device	3
1.1 Plc Logic: Plc Logic	3
1.1.1 Application: Application	3
1.2 Device: ASi_Master_1	97
1.2.1 Device: ASi_1_binaryIO	98
1.2.2 Device: ASi_1_analogIO	98
1.3 Device: ASi_Master_2	98
1.3.1 Device: ASi_2_binaryIO	98
1.3.2 Device: ASi_2_analogIO	99
1.4 Device: Fieldbus_Interface	99
1.4.1 Device: FieldBusData_	99
1.4.2 Device: ASi_1_binaryOut	99
1.4.3 Device: ASi_1_analogOut	100
1.4.4 Device: ASi_2_binaryOut	100
1.4.5 Device: ASi_2_analogOut	100
1.5 Device: Ethernet	100
2 : Project Settings	101

# 1 Device: Device

## **Users and Groups**

Users:

Groups:

---

## **Information**

Name: AC14  
Vendor: ifm electronic  
Groups: PLCs  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Order number: AC14  
Description: Programmable AS-Interface Gateway

---

## **Information**

Name: AC14  
Vendor: ifm electronic  
Groups: PLCs  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Order number: AC14  
Description: Programmable AS-Interface Gateway

## ***1.1 Plc Logic: Plc Logic***

### **1.1.1 Application: Application**

### 1.1.1.1 Folder: Funkce

#### 1.1.1.1.1 Folder: Manual

##### 1.1.1.1.1.1 POU: Dopravnik

```
1  FUNCTION_BLOCK Dopravnik
2  VAR_INPUT
3      Start : BOOL ;
4      Akt_pozice : INT ;           //aktualni pozice dopravniku
5      Pozice : INT ;             //vyzadana pozice napoje
6  END_VAR
7  VAR_OUTPUT
8      Start_right : BOOL ;
9      Start_left : BOOL ;
10     Complete : BOOL ;
11     Aborted : BOOL ;
12 END_VAR
13 VAR_CONSTANT
14     Idle : UINT := 0 ;
15     Pohyb : UINT := 10 ;
16     Ukonceni : UINT := 20 ;
17 END_VAR
18 VAR
19     Stav : UINT := 0 ;
20     Ukonceni_TON : TON ;
21     Complete_TON : TON ;
22     a : INT ;
23     Compltete_TON : INT ;
24 END_VAR
25
```

```
1  //0 - pocatek
2  //1 - prvni lahev
3  //2 - druha lahev
4  //3 - tretí lahev
5  //4 - ctvrta lahev
6  //5 - konec
7
8  Start_right := FALSE ;
9  Start_left := FALSE ;
10 Complete := FALSE ;
11
12
13 Complete_TON ( In := ( ( pozice * 2 ) = akt_pozice ) , PT := T#3S ) ;
14 Ukonceni_TON ( In := Stav = Ukonceni , PT := T#2S ) ;
15
16 a := ( Pozice * 2 ) - Akt_pozice ;
17
```

```
18      //Prechod mezi stavy
19      CASE Stav OF
20
21          Idle :
22
23              IF Start THEN
24                  Stav := Pohyb ;
25              END_IF
26
27          Pohyb :
28
29              IF Complete_TON . Q THEN
30                  Stav := Ukonceni ;
31              END_IF
32
33          Ukonceni :
34
35              IF Ukonceni_TON . Q OR NOT Start THEN
36                  Stav := Idle ;
37              END_IF
38
39      END_CASE
40
41
42      //Zapis na vystupy
43      CASE Stav OF
44
45          Idle :
46
47          Pohyb :
48
49              IF ( a < 0 ) THEN
50                  Start_left := TRUE ;
51              END_IF
52
53              IF ( a > 0 ) THEN
54                  Start_right := TRUE ;
55              END_IF
56
57          Ukonceni :
58
59              Complete := TRUE ;
60
61      END_CASE
62
63
64
65
66
67
68
```

### 1.1.1.1.2 POU: Init\_poloha

```
1  FUNCTION_BLOCK Init_poloha
2  VAR_INPUT
3      Akt_pozice : INT ;
4      TL_STOP : BOOL ;
5      Vytah_dole : BOOL ;
6  END_VAR
7  VAR_OUTPUT
8      Start_vytah_dolu : BOOL ;
9      Start_dopravnik_doleva : BOOL ;
10 END_VAR
11 VAR_CONSTANT
12     Idle : UINT := 0 ;
13     Vytah : UINT := 10 ;
14     Dopravnik : UINT := 20 ;
15     Ukonceni : UINT := 30 ;
16 END_VAR
17 VAR
18     Stav : UINT := 0 ;
19     Stop_TON : TON ;
20     Uvolneni_TON : TON ;
21     Ukonceni_TON : TON ;
22     Start : bool ;
23 END_VAR
24
```

---

```
1  Start_vytah_dolu := FALSE ;
2  Start_dopravnik_doleva := FALSE ;
3
4  //Pri drzeni tlacika STOP 5s se vytah vrati na puvodni pozici pro
   naskladneni sklenice
5  Stop_TON ( IN := ( TL_STOP AND NOT OPT_VYTAH ) , PT := T#3S , Q => , ET => ) ;
6
7  Ukonceni_TON ( IN := Stav = Ukonceni , PT := T#2S ) ;
8
9
10 //Prechod mezi stavy
11 CASE Stav OF
12
13     Idle :
14
15         IF Stop_TON . Q AND NOT Vytah_dole THEN
16             Stav := Vytah ;
17         ELSIF Stop_TON . Q AND Vytah_dole THEN
18             Stav := Dopravnik ;
19         END_IF
20
21     Vytah :
22
23         IF Vytah_dole THEN
24             Stav := Dopravnik ;
25         END_IF
```

```
26
27     Dopravnik :
28
29         IF Akt_Pozice = 0 THEN
30             Stav := Ukonceni ;
31         END_IF
32
33     Ukonceni :
34
35         IF Ukonceni_TON . Q THEN
36             Stav := Idle ;
37         END_IF
38
39     END_CASE
40
41
42     //Zapis na vystupy
43     CASE Stav OF
44
45         Idle :
46
47         Vytah :
48
49             Start_vytah_dolu := TRUE ;
50
51         Dopravnik :
52
53             Start_dopravnik_doleva := TRUE ;
54
55         Ukonceni :
56
57
58     END_CASE
59
60
61
62
```

### 1.1.1.1.3 POU: Podavac

```
1     FUNCTION_BLOCK Podavac
2     VAR_INPUT
3         Start : BOOL ;
4         Akt_pozice : INT ;
5     END_VAR
6     VAR_CONSTANT
7         Idle : UINT := 0 ;
8         Nahoru_1 : UINT := 10 ;
9         Dolu_1 : UINT := 20 ;
10        Nahoru_2 : UINT := 30 ;
11        Dolu_2 : UINT := 40 ;
12        Ukonceni : UINT := 50 ;
13    END_VAR
14    VAR_OUTPUT
15        Podl_Nahoru : BOOL ;
16        Podl_Dolu : BOOL ;
```

```
17     Pod2_Nahoru : BOOL ;
18     Pod2_Dolu : BOOL ;
19     Complete : BOOL ;
20     Aborted : BOOL ;
21 END_VAR
22 VAR
23     Ukonceni_TON : TON ;
24     Stav : UINT := 0 ;
25
26     //Simulace senzoru
27     Sim_Podavac_TON : TON ;
28     Sim_Vytah_TON : TON ;
29
30
31 END_VAR
32
```

---

```
1     Pod1_Nahoru := FALSE ;
2     Pod2_Nahoru := FALSE ;
3     Pod1_Dolu := FALSE ;
4     Pod2_Dolu := FALSE ;
5     Complete := FALSE ;
6
7     Ukonceni_TON ( In := ( Stav = Ukonceni ) , PT := T#2S ) ;
8     //Simulace
9     Sim_Podavac_TON ( IN := ( Stav = Nahoru_1 ) , PT := T#3S ) ;
10    Sim_Vytah_TON ( IN := ( Stav = Nahoru_2 ) , PT := T#3S ) ;
11
12    //Prechod mezi stavy
13    CASE Stav OF
14
15        Idle :
16
17            IF Start AND Akt_pozice = 0 THEN
18                Stav := Nahoru_1 ;
19            END_IF
20
21        Nahoru_1 :
22
23            IF POD_1_UP_OK AND ( OPT_PODAVAC OR Sim_Podavac_TON . Q ) THEN
24                Stav := Dolu_1 ;
25            END_IF
26
27        Dolu_1 :
28
29            IF POD_1_DOWN_OK THEN
30                Stav := Nahoru_2 ;
31            END_IF
32
33        Nahoru_2 :
34
35            IF POD_2_UP_OK AND ( OPT_Vytah OR Sim_Vytah_TON . Q ) THEN
36                Stav := Dolu_2 ;
37            END_IF
```



```
38
39     Dolu_2 :
40
41         IF POD_2_DOWN_OK THEN
42             Stav := Ukonceni ;
43         END_IF
44
45     Ukonceni :
46
47         IF Ukonceni_TON . Q THEN
48             Stav := Idle ;
49         END_IF
50
51     END_CASE
52
53
54     //Zapis na vystupy
55     CASE Stav OF
56
57         Idle :
58
59
60         Nahoru_1 :
61
62             Pod1_Nahoru := TRUE ;
63
64         Dolu_1 :
65
66             Pod1_Dolu := TRUE ;
67
68         Nahoru_2 :
69
70             Pod2_Nahoru := TRUE ;
71
72         Dolu_2 :
73
74             Pod2_Dolu := TRUE ;
75
76         Ukonceni :
77
78             Complete := TRUE ;
79
80     END_CASE
81
```

### 1.1.1.1.1.4 POU: Rizeni

```
1      FUNCTION_BLOCK Rizeni
2      VAR_INPUT
3          Akt_pozice : INT ;
4          Tlac_1 : BOOL ;
5          Tlac_2 : BOOL ;
6          Tlac_3 : BOOL ;
7          Tlac_4 : BOOL ;
8          TLac_Stop : BOOL ;
9          Sklenice_Complete : BOOL ;
10         Dopravnik_Complete : BOOL ;
11         Vytah_Complete : BOOL ;
12     END_VAR
13     VAR_OUTPUT
14         Pozice : INT ;
15         Start_Podavac : BOOL ;
16         Start_Vytah : BOOL ;
17         Start_Dopravnik : BOOL ;
18         Majak_Signalizace : INT ;
19     END_VAR
20     VAR_CONSTANT
21         Idle : UINT := 0 ;
22         Nulovani : UINT := 2 ;
23         Tlacitka : UINT := 3 ;
24         Lahve : UINT := 10 ;
25         Sklenice : UINT := 20 ;
26         Dopravnik : UINT := 30 ;
27         Vytah : UINT := 40 ;
28         Konec : UINT := 50 ;
29         Stop : UINT := 60 ;
30         sPorucha : UINT := 99 ;
31     END_VAR
32     VAR
33         Stav : UINT := 0 ;
34         Start_TRIG : R_TRIG ;
35         Konec_TON : TON ;
36         Stop_TON : TON ;
37         Running : BOOL ;
38         Start : BOOL ;
39         i : INT ;
40         Done : BOOL ;
41         Cnt : INT ;
42         K : INT ;
43     END_VAR
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

```

7
8  //Prechody mezi stavy
9  CASE Stav OF
10     Idle :
11
12         IF ( Tlac_1 OR Tlac_2 OR Tlac_3 OR Tlac_4 ) AND Akt_Pozice = 0
13     THEN
14         Stav := Nulovani ;
15         IF Tlac_1 THEN
16             ManDrink [ 1 ] := 1 ;
17         ELSIF Tlac_2 THEN
18             ManDrink [ 1 ] := 2 ;
19         ELSIF Tlac_3 THEN
20             ManDrink [ 1 ] := 3 ;
21         ELSIF Tlac_4 THEN
22             ManDrink [ 1 ] := 4 ;
23         END_IF
24     END_IF
25
26     //Nulovani pole pro drink
27     Nulovani :
28
29         IF Done THEN
30             Stav := Lahve ;
31         END_IF
32
33     //Zadavani lahvi pro vysledny drink
34     Lahve :
35
36         IF Cnt > 1 AND Tlac_Stop AND NOT OPT_Vytah THEN
37             Stav := Sklenice ;
38         ELSIF Cnt > 1 AND Tlac_Stop AND OPT_Vytah THEN
39             Stav := Dopravnik ;
40         ELSIF NOT ( ( OPT_1 AND BOTTLE_1 ) AND ( OPT_2 AND BOTTLE_2 ) AND
41             ( OPT_3 AND BOTTLE_3 ) AND ( OPT_4 AND BOTTLE_4 ) ) OR Simulace ) THEN
42             Stav := sPorucha ;
43         END_IF
44
45     Sklenice :
46
47         IF Sklenice_Complete THEN
48             Stav := Dopravnik ;
49         END_IF
50
51     Dopravnik :
52
53         IF Dopravnik_Complete AND i < Cnt THEN
54             Stav := Vytah ;
55         ELSIF Dopravnik_Complete AND i >= Cnt THEN
56             Stav := Konec ;
57         END_IF
58
59     Vytah :
60
61         IF Vytah_Complete THEN

```

```
61         Stav := Dopravnik ;
62         i := i + 1 ;
63     END_IF
64
65     Konec :
66
67     IF Konec_TON . Q THEN
68         Stav := Idle ;
69     END_IF
70
71     Stop :
72
73     IF Stop_TON . Q THEN
74         Stav := Idle ;
75     END_IF
76
77     sPorucha :
78
79     IF TLac_Stop THEN
80         Stav := Idle ;
81     END_IF ;
82
83 END_CASE
84
85
86 //Zapis na vystupy
87 CASE Stav OF
88
89     Idle :
90
91         Done := FALSE ;
92         Porucha := false ;
93
94         IF Akt_Pozice = 0 THEN
95
96             IF ( OPT_1 AND BOTTLE_1 ) OR Simulace THEN
97                 TLAC_1_LIGHT := TRUE ;
98             ELSE
99                 TLAC_1_LIGHT := FALSE ;
100             END_IF
101
102             IF ( OPT_2 AND BOTTLE_2 ) OR Simulace THEN
103                 TLAC_2_LIGHT := TRUE ;
104             ELSE
105                 TLAC_2_LIGHT := FALSE ;
106             END_IF
107
108             IF ( OPT_3 AND BOTTLE_3 ) OR Simulace THEN
109                 TLAC_3_LIGHT := TRUE ;
110             ELSE
111                 TLAC_3_LIGHT := FALSE ;
112             END_IF
113
114             IF ( OPT_4 AND BOTTLE_4 ) OR Simulace THEN
115                 TLAC_4_LIGHT := TRUE ;
116             ELSE
```

```
117         TLAC_4_LIGHT := FALSE ;
118     END_IF
119
120     TLAC_STOP_LIGHT := FALSE ;
121
122     ELSE
123
124         TLAC_1_LIGHT := FALSE ;
125         TLAC_2_LIGHT := FALSE ;
126         TLAC_3_LIGHT := FALSE ;
127         TLAC_4_LIGHT := FALSE ;
128         TLAC_STOP_LIGHT := TRUE ;
129
130     END_IF
131
132     //Nulovani pole pro drink
133     Nulovani :
134
135         FOR i := 2 TO 7 DO
136             ManDrink [ i ] := 0 ;
137         END_FOR ;
138
139         i := 1 ;
140         Cnt := 2 ;
141         Done := TRUE ;
142
143     //Zadavani lahvi pro vysledny drink
144     Lahve :
145
146         IF Cnt < 8 THEN
147             IF Tlac_1 AND ( ( OPT_1 AND BOTTLE_1 ) OR Simulace ) THEN
148                 ManDrink [ Cnt ] := 1 ;
149                 Cnt := Cnt + 1 ;
150             ELIF Tlac_2 AND ( ( OPT_2 AND BOTTLE_2 ) OR Simulace ) THEN
151                 ManDrink [ Cnt ] := 2 ;
152                 Cnt := Cnt + 1 ;
153             ELIF Tlac_3 AND ( ( OPT_3 AND BOTTLE_3 ) OR Simulace ) THEN
154                 ManDrink [ Cnt ] := 3 ;
155                 Cnt := Cnt + 1 ;
156             ELIF Tlac_4 AND ( ( OPT_4 AND BOTTLE_4 ) OR Simulace ) THEN
157                 ManDrink [ Cnt ] := 4 ;
158                 Cnt := Cnt + 1 ;
159             END_IF
160         ELSE
161             TLAC_1_LIGHT := FALSE ;
162             TLAC_2_LIGHT := FALSE ;
163             TLAC_3_LIGHT := FALSE ;
164             TLAC_4_LIGHT := FALSE ;
165         END_IF
166
167         IF Cnt > 1 THEN
168             TLAC_STOP_LIGHT := TRUE ;
169         END_IF
170
171     Sklenice :
172
```

```
173         TLAC_1_LIGHT := FALSE ;
174         TLAC_2_LIGHT := FALSE ;
175         TLAC_3_LIGHT := FALSE ;
176         TLAC_4_LIGHT := FALSE ;
177
178         Start_Podavac := TRUE ;
179
180     Dopravnik :
181
182         IF i >= Cnt THEN
183             Pozice := 5 ;
184         ELSE
185             Pozice := ManDrink [ i ] ;
186         END_IF
187
188         Start_Dopravnik := TRUE ;
189
190     Vytah :
191
192         Start_Vytah := TRUE ;
193
194     Konec :
195
196         Done := FALSE ;
197
198     Stop :
199
200         Porucha := TRUE ;
201
202     sPorucha :
203
204         Majak_Signalizace := 2 ;
205         TLAC_STOP_LIGHT := TRUE ;
206         TLAC_1_LIGHT := FALSE ;
207         TLAC_2_LIGHT := FALSE ;
208         TLAC_3_LIGHT := FALSE ;
209         TLAC_4_LIGHT := FALSE ;
210
211     END_CASE
212
213
214
215
```

### 1.1.1.1.5 POU: Vytah

```
1      FUNCTION_BLOCK Vytah
2      VAR_INPUT
3          Start : BOOL ;
4      END_VAR
5      VAR_OUTPUT
6          Start_nahoru : BOOL ;
7          Start_dolu : BOOL ;
8          Complete : BOOL ;
9          Aborted : BOOL ;
10     END_VAR
11     VAR_CONSTANT
12         Idle : UINT := 0 ;
13         Nahoru : UINT := 10 ;
14         Cekani : UINT := 20 ;
15         Dolu : UINT := 30 ;
16         Ukonceni : UINT := 40 ;
17     END_VAR
18     VAR
19         Complete_TON : TON ;
20         Cekani_TON : TON ;
21         Stav : UINT := 0 ;
22         Start_TLAC : R_TRIG ;
23     END_VAR
24
```

---

```
1      Start_nahoru := FALSE ;
2      Start_dolu := FALSE ;
3      Complete := FALSE ;
4
5      Cekani_TON ( In := Stav = Cekani , PT := T#5S ) ;
6      Complete_TON ( In := Stav = Ukonceni , PT := T#1S ) ;
7      Start_TLAC ( CLK := Start ) ;
8
9      //Prechod mezi stavy
10     CASE Stav OF
11
12         Idle :
13
14             IF Start_TLAC . Q THEN
15                 Stav := Nahoru ;
16             END_IF
17
18         Nahoru :
19
20             IF Vytah_UP_OK THEN
21                 Stav := Cekani ;
22             END_IF
23
24         Cekani :
25
26             IF Cekani_TON . Q THEN
```

```
27         Stav := Dolu ;
28     END_IF
29
30     Dolu :
31
32         IF Vytah_DOWN_OK THEN
33             Stav := Ukonceni ;
34         END_IF
35
36     Ukonceni :
37
38         IF Complete_TON . Q THEN
39             Stav := Idle ;
40         END_IF
41
42     END_CASE
43
44     //Zapis na vystupy
45     CASE Stav OF
46
47         Idle :
48
49         Nahoru :
50
51             Start_nahoru := TRUE ;
52
53         Cekani :
54
55             Start_nahoru := TRUE ;
56
57         Dolu :
58
59             Start_dolu := TRUE ;
60
61         Ukonceni :
62
63             Complete := TRUE ;
64
65     END_CASE
66
67
68
```

### 1.1.1.1.2 Folder: Operace



## 1.1.1.1.2.1 Folder: Dokonceni

### 1.1.1.1.2.1.1 POU: faze\_PocatecniPoloha

```
1  FUNCTION_BLOCK faze_PocatecniPoloha
2  VAR_INPUT
3      Poloha : BOOL ;
4      Start : BOOL ;
5      Resume : BOOL ;
6      Hold : BOOL ;
7      Abort : BOOL ;
8  END_VAR
9  VAR_OUTPUT
10     Run : BOOL ;
11     Running : BOOL ;
12     Held : BOOL ;
13     Aborted : BOOL ;
14     Complete : BOOL ;
15     Porucha : bool ;
16 END_VAR
17 VAR_CONSTANT
18     //Stavy faze
19     fIdle : UINT := 0 ;
20     fRunning : UINT := 1 ;
21     fHeld : UINT := 2 ;
22     fAborted : UINT := 3 ;
23     fComplete : UINT := 4 ;
24
25 END_VAR
26 VAR
27     Start_Trig : R_Trig ;
28     Resume_Trig : R_Trig ;
29     Abort_Trig : R_Trig ;
30     Hold_Trig : R_Trig ;
31     Stav_faze : UINT := 0 ;
32     Ton_Complete : TON ;
33     Ton_Idle : TON ;
34     fTimeOut : TON ;
35 END_VAR
36
```

```
1  //Uvolneni vystupu
2  Running := FALSE ;
3  Complete := FALSE ;
4  Aborted := FALSE ;
5  Held := FALSE ;
6  Run := FALSE ;
7
8
9
10 //Poozadavek na spusteni faze
11 Start_Trig ( CLK := Start ) ;
```

```
12      //Pozadavek na pokracovani faze
13      Resume_Trig ( CLK := Resume ) ;
14      //Pozadavek na preruseni faze
15      Abort_Trig ( CLK := Abort ) ;
16      //Pozadavek pro pozastaveni faze
17      Hold_Trig ( CLK := Hold ) ;
18
19      //Spravna poloha - Ukonceni faze
20      Ton_Complete ( In := Poloha AND ( Stav_faze = fRunning ) , PT := T#3S ) ;
21      //Ukonceni faze
22      Ton_Idle ( In := ( Stav_faze = fComplete ) OR ( Stav_faze = fAborted ) , PT :=
23      T#5S ) ;
24      //TimeOut faze
25      fTimeout ( IN := ( Stav_faze = fRunning ) , PT := T#15S ) ;
26
27
28
29      //Stavac pro faze - prechody
30      CASE Stav_faze OF
31          fIdle :
32              IF Start_Trig . Q THEN
33                  Stav_faze := fRunning ;
34              END_IF
35
36          fRunning :
37              IF Ton_Complete . Q THEN
38                  Stav_faze := fComplete ;
39              END_IF
40
41              IF Hold_Trig . Q THEN
42                  Stav_faze := fHeld ;
43              END_IF
44
45              IF Abort_Trig . Q THEN
46                  Stav_faze := fAborted ;
47              END_IF
48
49              IF fTimeout . Q THEN
50                  Porucha := TRUE ;
51              END_IF ;
52
53          fAborted :
54              IF Ton_Idle . Q THEN
55                  Stav_faze := fIdle ;
56              END_IF
57
58          fComplete :
59              IF Ton_Idle . Q THEN
60                  Stav_faze := fIdle ;
61              END_IF
62
63          fHeld :
64              IF Resume_Trig . Q THEN
65                  Stav_faze := fRunning ;
66              END_IF
```

```
67
68         IF Abort_Trig . Q THEN
69             Stav_faze := fAborted ;
70         END_IF
71
72     END_CASE
73
74     //Stavac pro faze - zapisy
75     CASE Stav_faze OF
76         fIdle :
77
78         fRunning :
79
80             Running := TRUE ;
81
82             IF NOT Poloha THEN
83                 Run := TRUE ;
84             END_IF ;
85
86         fHeld :
87             Held := TRUE ;
88
89         fAborted :
90             Aborted := TRUE ;
91
92         fComplete :
93             Complete := TRUE ;
94
95     END_CASE
96
```

#### 1.1.1.1.2.1.2 POU: faze\_SenzorVytah

```
1     FUNCTION_BLOCK faze_SenzorVytah
2     VAR_INPUT
3         Cidlo : BOOL ;
4         Start : BOOL ;
5         Resume : BOOL ;
6         Hold : BOOL ;
7         Abort : BOOL ;
8     END_VAR
9     VAR_OUTPUT
10        Running : BOOL ;
11        Held : BOOL ;
12        Aborted : BOOL ;
13        Complete : BOOL ;
14        Porucha : bool ;
15    END_VAR
16    VAR CONSTANT
17        //Stavy faze
18        fIdle : UINT := 0 ;
19        fRunning : UINT := 1 ;
20        fHeld : UINT := 2 ;
21        fAborted : UINT := 3 ;
22        fComplete : UINT := 4 ;
23
```

```
24     END_VAR
25     VAR
26         Start_Trig : R_Trig ;
27         Resume_Trig : R_Trig ;
28         Abort_Trig : R_Trig ;
29         Hold_Trig : R_Trig ;
30         Stav_faze : UINT := 0 ;
31         Ton_Complete : TON ;
32         Ton_Idle : TON ;
33         fTimeOut : TON ;
34     END_VAR
35
36
```

---

```
1     //Uvolneni vystupu
2     Running := FALSE ;
3     Complete := FALSE ;
4     Aborted := FALSE ;
5     Held := FALSE ;
6
7
8     //Pozadavek na spusteni faze
9     Start_Trig ( CLK := Start ) ;
10    //Pozadavek na pokracovani faze
11    Resume_Trig ( CLK := Resume ) ;
12    //Pozadavek na preruseni faze
13    Abort_Trig ( CLK := Abort ) ;
14    //Pozadavek pro pozastaveni faze
15    Hold_Trig ( CLK := Hold ) ;
16
17    //Spravna poloha - Ukonceni faze
18    Ton_Complete ( In := Cidlo AND ( Stav_faze = fRunning ) , PT := T#2S ) ;
19    //Ukonceni faze
20    Ton_Idle ( In := ( Stav_faze = fComplete ) OR ( Stav_faze = fAborted ) , PT :=
    T#5S ) ;
21    //TimeOut faze
22    fTimeOut ( IN := ( NOT cidlo AND ( Stav_faze = fRunning ) ) , PT := T#5S ) ;
23
24    //Stavac pro faze - prechody
25    CASE Stav_faze OF
26        fIdle :
27            IF Start_Trig . Q THEN
28                Stav_faze := fRunning ;
29            END_IF
30
31        fRunning :
32            IF Ton_Complete . Q THEN
33                Stav_faze := fComplete ;
34            END_IF
35
36            IF Hold_Trig . Q THEN
37                Stav_faze := fHeld ;
38            END_IF
39
40            IF Abort_Trig . Q THEN
```

```
41         Stav_faze := fAborted ;
42     END_IF
43
44     IF fTimeOut . Q THEN
45         Porucha := TRUE ;
46     END_IF ;
47
48     fAborted :
49         IF Ton_Idle . Q THEN
50             Stav_faze := fIdle ;
51         END_IF
52
53     fComplete :
54         IF Ton_Idle . Q THEN
55             Stav_faze := fIdle ;
56         END_IF
57
58     fHeld :
59         IF Resume_Trig . Q THEN
60             Stav_faze := fRunning ;
61         END_IF
62
63         IF Abort_Trig . Q THEN
64             Stav_faze := fAborted ;
65         END_IF
66
67     END_CASE
68
69     //Stavac pro faze - zapisy
70     CASE Stav_faze OF
71         fIdle :
72
73         fRunning :
74
75             Running := TRUE ;
76
77         fHeld :
78             Held := TRUE ;
79
80         fAborted :
81             Aborted := TRUE ;
82
83         fComplete :
84             Complete := TRUE ;
85
86     END_CASE
87
```

### 1.1.1.1.2.1.3 POU: faze\_TL\_Stop

```
1      FUNCTION_BLOCK faze_TL_Stop
2      VAR_INPUT
3          Tlacitko : bool ;
4          Start : BOOL ;
5          Resume : BOOL ;
6          Hold : BOOL ;
7          Abort : BOOL ;
8      END_VAR
9      VAR_OUTPUT
10         Running : BOOL ;
11         Held : BOOL ;
12         Aborted : BOOL ;
13         Complete : BOOL ;
14     END_VAR
15     VAR CONSTANT
16         //Stavy faze
17         fIdle : UINT := 0 ;
18         fRunning : UINT := 1 ;
19         fHeld : UINT := 2 ;
20         fAborted : UINT := 3 ;
21         fComplete : UINT := 4 ;
22
23     END_VAR
24     VAR
25         Start_Trig : R_Trig ;
26         Resume_Trig : R_Trig ;
27         Abort_Trig : R_Trig ;
28         Hold_Trig : R_Trig ;
29         Stav_faze : UINT := 0 ;
30         Ton_Complete : TON ;
31         Ton_Idle : TON ;
32
33     END_VAR
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
263
```

```
17      //Spravna poloha - Ukonceni faze
18      Ton_Complete ( In := Tlacitko AND ( Stav_faze = fRunning ) , PT := T#3S ) ;
19      //Ukonceni faze
20      Ton_Idle ( In := ( Stav_faze = fComplete ) OR ( Stav_faze = fAborted ) , PT :=
21      T#5S ) ;
22
23      //Stavac pro faze - prechody
24      CASE Stav_faze OF
25          fIdle :
26              IF Start_Trig . Q THEN
27                  Stav_faze := fRunning ;
28              END_IF
29
30          fRunning :
31              IF Ton_Complete . Q THEN
32                  Stav_faze := fComplete ;
33              END_IF
34
35              IF Hold_Trig . Q THEN
36                  Stav_faze := fHeld ;
37              END_IF
38
39              IF Abort_Trig . Q THEN
40                  Stav_faze := fAborted ;
41              END_IF
42
43          fAborted :
44              IF Ton_Idle . Q THEN
45                  Stav_faze := fIdle ;
46              END_IF
47
48          fComplete :
49              IF Ton_Idle . Q THEN
50                  Stav_faze := fIdle ;
51              END_IF
52
53          fHeld :
54              IF Resume_Trig . Q THEN
55                  Stav_faze := fRunning ;
56              END_IF
57
58              IF Abort_Trig . Q THEN
59                  Stav_faze := fAborted ;
60              END_IF
61
62      END_CASE
63
64      //Stavac pro faze - zapisy
65      CASE Stav_faze OF
66          fIdle :
67
68          fRunning :
69
70              Running := TRUE ;
71
```

```
72         fHeld :
73             Held := TRUE ;
74
75         fAborted :
76             Aborted := TRUE ;
77
78         fComplete :
79             Complete := TRUE ;
80
81     END_CASE
82
```

## 1.1.1.1.2.2 Folder: Lahve

### 1.1.1.1.2.2.1 POU: faze\_Dopravnik

```
1     FUNCTION_BLOCK faze_Dopravnik
2     VAR_IN_OUT
3         Aktualni_pozice : INT ;
4     END_VAR
5     VAR_INPUT
6         Pozice : INT ;
7         Start : BOOL ;
8         Resume : BOOL ;
9         Hold : BOOL ;
10        Abort : BOOL ;
11    END_VAR
12    VAR_OUTPUT
13        Left : BOOL ;
14        Right : BOOL ;
15        Running : BOOL ;
16        Held : BOOL ;
17        Aborted : BOOL ;
18        Complete : BOOL ;
19        Porucha : BOOL ;
20    END_VAR
21    VAR_CONSTANT
22        //Stavy faze
23        fIdle : UINT := 0 ;
24        fRunning : UINT := 1 ;
25        fHeld : UINT := 2 ;
26        fAborted : UINT := 3 ;
27        fComplete : UINT := 4 ;
28
29    END_VAR
30    VAR
31        Start_Trig : R_Trig ;
32        Resume_Trig : R_Trig ;
33        Abort_Trig : R_Trig ;
34        Hold_Trig : R_Trig ;
35        Stav_faze : UINT := 0 ;
36        Ton_Complete : TON ;
37        Ton_Idle : TON ;
38        fTimeOut : TON ;
```



Barman.project 07.05.2019 20:45 Page 25 of 101

```
50         IF Ton_Idle . Q THEN
51             Stav_faze := fIdle ;
52         END_IF
53
54     fComplete :
55         IF Ton_Idle . Q THEN
56             Stav_faze := fIdle ;
57         END_IF
58
59     fHeld :
60         IF Resume_Trig . Q THEN
61             Stav_faze := fRunning ;
62         END_IF
63
64         IF Abort_Trig . Q THEN
65             Stav_faze := fAborted ;
66         END_IF
67
68     END_CASE
69
70     //Stavac pro faze - zapisy
71     CASE Stav_faze OF
72         fIdle :
73
74         fRunning :
75
76             Running := TRUE ;
77
78             //Pozadovany smer pohybu
79             Pozad_Pozice := ( pozice * 2 ) - Aktualni_pozice ;
80
81             IF ( Pozad_Pozice < 0 ) THEN
82                 Left := TRUE ;
83             ELSIF ( Pozad_Pozice > 0 ) THEN
84                 Right := TRUE ;
85             END_IF
86
87
88         fHeld :
89             Held := TRUE ;
90
91         fAborted :
92             Aborted := TRUE ;
93
94         fComplete :
95             Complete := TRUE ;
96
97     END_CASE
98
```

### 1.1.1.1.2.2.2 POU: faze\_Vytah

```
1      FUNCTION_BLOCK faze_Vytah
2      VAR_INPUT
3          Cidlo : BOOL ;
4          Start : BOOL ;
5          Resume : BOOL ;
6          Hold : BOOL ;
7          Abort : BOOL ;
8      END_VAR
9      VAR_OUTPUT
10         Run : BOOL ;
11         Running : BOOL ;
12         Held : BOOL ;
13         Aborted : BOOL ;
14         Complete : BOOL ;
15         Porucha : BOOL ;
16     END_VAR
17     VAR_CONSTANT
18         //Stavy faze
19         fIdle : UINT := 0 ;
20         fRunning : UINT := 1 ;
21         fHeld : UINT := 2 ;
22         fAborted : UINT := 3 ;
23         fComplete : UINT := 4 ;
24
25     END_VAR
26     VAR
27         Start_Trig : R_Trig ;
28         Resume_Trig : R_Trig ;
29         Abort_Trig : R_Trig ;
30         Hold_Trig : R_Trig ;
31         Stav_faze : UINT := 0 ;
32         Ton_Complete : TON ;
33         Ton_Idle : TON ;
34         fTimeOut : TON ;
35     END_VAR
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628

```

```
16 Hold_Trig ( CLK := Hold ) ;
17
18 //Spravna poloha - Ukonceni faze
19 Ton_Complete ( In := Cidlo AND ( Stav_faze = fRunning ) , PT := T#3S ) ;
20 //Ukonceni faze
21 Ton_Idle ( In := ( Stav_faze = fComplete ) OR ( Stav_faze = fAborted ) , PT :=
T#5S ) ;
22 //Timeout faze
23 fTimeout ( IN := ( Stav_faze = fRunning ) , PT := T#15S ) ;
24
25
26 //Stavac pro faze - prechody
27 CASE Stav_faze OF
28   fIdle :
29     IF Start_Trig . Q THEN
30       Stav_faze := fRunning ;
31     END_IF
32
33   fRunning :
34     IF Ton_Complete . Q THEN
35       Stav_faze := fComplete ;
36     END_IF
37
38     IF Hold_Trig . Q THEN
39       Stav_faze := fHeld ;
40     END_IF
41
42     IF Abort_Trig . Q THEN
43       Stav_faze := fAborted ;
44     END_IF
45
46     IF fTimeout . Q THEN
47       Porucha := TRUE ;
48     END_IF ;
49
50   fAborted :
51     IF Ton_Idle . Q THEN
52       Stav_faze := fIdle ;
53     END_IF
54
55   fComplete :
56     IF Ton_Idle . Q THEN
57       Stav_faze := fIdle ;
58     END_IF
59
60   fHeld :
61     IF Resume_Trig . Q THEN
62       Stav_faze := fRunning ;
63     END_IF
64
65     IF Abort_Trig . Q THEN
66       Stav_faze := fAborted ;
67     END_IF
68
69 END_CASE
70
```

```
71      //Stavac pro faze - zapisy
72      CASE Stav_faze OF
73          fIdle :
74
75          fRunning :
76
77              Running := TRUE ;
78              Run := TRUE ;
79
80          fHeld :
81              Held := TRUE ;
82
83          fAborted :
84              Aborted := TRUE ;
85
86          fComplete :
87              Complete := TRUE ;
88
89      END_CASE
90
```

### 1.1.1.1.2.3 Folder: Sklenice

#### 1.1.1.1.2.3.1 POU: faze\_PlnyVytah

```
1      FUNCTION_BLOCK faze_PlnyVytah
2      VAR_INPUT
3          Cidlo : BOOL ;
4          Start : BOOL ;
5          Resume : BOOL ;
6          Hold : BOOL ;
7          Abort : BOOL ;
8      END_VAR
9      VAR_OUTPUT
10         Run : BOOL ;
11         Running : BOOL ;
12         Held : BOOL ;
13         Aborted : BOOL ;
14         Complete : BOOL ;
15         Porucha : bool ;
16     END_VAR
17     VAR CONSTANT
18         //Stavy faze
19         fIdle : UINT := 0 ;
20         fRunning : UINT := 1 ;
21         fHeld : UINT := 2 ;
22         fAborted : UINT := 3 ;
23         fComplete : UINT := 4 ;
24     END_VAR
25     VAR
26
27         Start_Trig : R_Trig ;
28         Resume_Trig : R_Trig ;
29         Abort_Trig : R_Trig ;
```

```
30      Hold_Trig : R_Trig ;
31      Stav_faze : UINT := 0 ;
32      Ton_Complete : TON ;
33      Ton_Idle : TON ;
34      fTimeOut : TON ;
35      Pozad_Pozice : INT ;
36 END_VAR
37
```

---

```
1      //Uvolneni vystupu
2      Running := FALSE ;
3      Complete := FALSE ;
4      Aborted := FALSE ;
5      Held := FALSE ;
6      Run := FALSE ;
7
8
9      //Pozadavek na spusteni faze
10     Start_Trig ( CLK := Start ) ;
11     //Pozadavek na pokracovani faze
12     Resume_Trig ( CLK := Resume ) ;
13     //Pozadavek na preruseni faze
14     Abort_Trig ( CLK := Abort ) ;
15     //Pozadavek pro pozastaveni faze
16     Hold_Trig ( CLK := Hold ) ;
17
18     //Spravna poloha - Ukonceni faze
19     Ton_Complete ( In := Cidlo AND ( Stav_faze = fRunning ) , PT := T#1S ) ;
20     //Ukonceni faze
21     Ton_Idle ( In := ( Stav_faze = fComplete ) OR ( Stav_faze = fAborted ) , PT :=
T#5S ) ;
22     //TimeOut faze
23     fTimeOut ( IN := ( NOT cidlo AND ( Stav_faze = fRunning ) ) , PT := T#5S ) ;
24
25     //Stavac pro faze - prechody
26     CASE Stav_faze OF
27         fIdle :
28             IF Start_Trig . Q THEN
29                 Stav_faze := fRunning ;
30             END_IF
31
32         fRunning :
33             IF Ton_Complete . Q THEN
34                 Stav_faze := fComplete ;
35             END_IF
36
37             IF Hold_Trig . Q THEN
38                 Stav_faze := fHeld ;
39             END_IF
40
41             IF Abort_Trig . Q THEN
42                 Stav_faze := fAborted ;
43             END_IF
44
45             IF fTimeOut . Q THEN
```

```
46         Porucha := TRUE ;
47     END_IF ;
48
49     fAborted :
50         IF Ton_Idle . Q THEN
51             Stav_faze := fIdle ;
52         END_IF
53
54     fComplete :
55         IF Ton_Idle . Q THEN
56             Stav_faze := fIdle ;
57         END_IF
58
59     fHeld :
60         IF Resume_Trig . Q THEN
61             Stav_faze := fRunning ;
62         END_IF
63
64         IF Abort_Trig . Q THEN
65             Stav_faze := fAborted ;
66         END_IF
67
68     END_CASE
69
70     //Stavac pro faze - zapisy
71     CASE Stav_faze OF
72         fIdle :
73
74         fRunning :
75
76             Running := TRUE ;
77             Run := TRUE ;
78
79         fHeld :
80             Held := TRUE ;
81
82         fAborted :
83             Aborted := TRUE ;
84
85         fComplete :
86             Complete := TRUE ;
87
88     END_CASE
89
```

### 1.1.1.1.2.3.2 POU: faze\_Podavac

```
1      FUNCTION_BLOCK faze_Podavac
2      VAR_INPUT
3          Cidlo : BOOL ;
4          Start : BOOL ;
5          Resume : BOOL ;
6          Hold : BOOL ;
7          Abort : BOOL ;
8      END_VAR
9      VAR_OUTPUT
10         Run : BOOL ;
11         Running : BOOL ;
12         Held : BOOL ;
13         Aborted : BOOL ;
14         Complete : BOOL ;
15         Porucha : bool ;
16     END_VAR
17     VAR CONSTANT
18         //Stavy faze
19         fIdle : UINT := 0 ;
20         fRunning : UINT := 1 ;
21         fHeld : UINT := 2 ;
22         fAborted : UINT := 3 ;
23         fComplete : UINT := 4 ;
24
25     END_VAR
26     VAR
27         Start_Trig : R_Trig ;
28         Resume_Trig : R_Trig ;
29         Abort_Trig : R_Trig ;
30         Hold_Trig : R_Trig ;
31         Stav_faze : UINT := 0 ;
32         Ton_Complete : TON ;
33         Ton_Idle : TON ;
34         fTimeout : TON ;
35         Pozad_Pozice : INT ;
36     END_VAR
37
38
39     //Uvolneni vystupu
40     Running := FALSE ;
41     Complete := FALSE ;
42     Aborted := FALSE ;
43     Held := FALSE ;
44     Run := FALSE ;
45
46
47     //Pozadavek na spusteni faze
48     Start_Trig ( CLK := Start ) ;
49     //Pozadavek na pokracovani faze
50     Resume_Trig ( CLK := Resume ) ;
51     //Pozadavek na preruseni faze
52     Abort_Trig ( CLK := Abort ) ;
```

---



```
15      //Pozadavek pro pozastaveni faze
16      Hold_Trig ( CLK := Hold ) ;
17
18      //Spravna poloha - Ukonceni faze
19      Ton_Complete ( In := Cidlo AND ( Stav_faze = fRunning ) , PT := T#1S ) ;
20      //Ukonceni faze
21      Ton_Idle ( In := ( Stav_faze = fComplete ) OR ( Stav_faze = fAborted ) , PT :=
22      T#5S ) ;
23      //Timeout faze
24      fTimeout ( IN := ( Stav_faze = fRunning AND ( Stav_faze = fRunning ) ) , PT :=
25      T#10S ) ;
26
27      //Stavac pro faze - prechody
28      CASE Stav_faze OF
29          fIdle :
30              IF Start_Trig . Q THEN
31                  Stav_faze := fRunning ;
32              END_IF
33
34          fRunning :
35              IF Ton_Complete . Q THEN
36                  Stav_faze := fComplete ;
37              END_IF
38
39              IF Hold_Trig . Q THEN
40                  Stav_faze := fHeld ;
41              END_IF
42
43              IF Abort_Trig . Q THEN
44                  Stav_faze := fAborted ;
45              END_IF
46
47              IF fTimeout . Q THEN
48                  Porucha := TRUE ;
49              END_IF ;
50
51          fAborted :
52              IF Ton_Idle . Q THEN
53                  Stav_faze := fIdle ;
54              END_IF
55
56          fComplete :
57              IF Ton_Idle . Q THEN
58                  Stav_faze := fIdle ;
59              END_IF
60
61          fHeld :
62              IF Resume_Trig . Q THEN
63                  Stav_faze := fRunning ;
64              END_IF
65
66              IF Abort_Trig . Q THEN
67                  Stav_faze := fAborted ;
68              END_IF
69      END_CASE
```

```
69
70 //Stavac pro faze - zapisy
71 CASE Stav_faze OF
72     fIdle :
73
74     fRunning :
75
76         Running := TRUE ;
77         Run := TRUE ;
78
79     fHeld :
80         Held := TRUE ;
81
82     fAborted :
83         Aborted := TRUE ;
84
85     fComplete :
86         Complete := TRUE ;
87
88 END_CASE
89
```

### 1.1.1.1.2.3.3 POU: faze\_Zasobnik

```
1 FUNCTION_BLOCK faze_Zasobnik
2 VAR_INPUT
3     Cidlo : BOOL ;
4     Start : BOOL ;
5     Resume : BOOL ;
6     Hold : BOOL ;
7     Abort : BOOL ;
8 END_VAR
9 VAR_OUTPUT
10    Run : BOOL ;
11    Running : BOOL ;
12    Held : BOOL ;
13    Aborted : BOOL ;
14    Complete : BOOL ;
15    Porucha : bool ;
16 END_VAR
17 VAR CONSTANT
18     //Stavy faze
19     fIdle : UINT := 0 ;
20     fRunning : UINT := 1 ;
21     fHeld : UINT := 2 ;
22     fAborted : UINT := 3 ;
23     fComplete : UINT := 4 ;
24
25 END_VAR
26 VAR
27     Start_Trig : R_Trig ;
28     Resume_Trig : R_Trig ;
29     Abort_Trig : R_Trig ;
30     Hold_Trig : R_Trig ;
31     Stav_faze : UINT := 0 ;
32     Ton_Complete : TON ;
```

```
33     Ton_Idle : TON ;
34     fTimeOut : TON ;
35
36     Pozad_Pozice : INT ;
37 END_VAR
38
```

---

```
1     //Uvolneni vystupu
2     Running := FALSE ;
3     Complete := FALSE ;
4     Aborted := FALSE ;
5     Held := FALSE ;
6     Run := FALSE ;
7
8
9     //Pozadavek na spusteni faze
10    Start_Trig ( CLK := Start ) ;
11    //Pozadavek na pokracovani faze
12    Resume_Trig ( CLK := Resume ) ;
13    //Pozadavek na preruseni faze
14    Abort_Trig ( CLK := Abort ) ;
15    //Pozadavek pro pozastaveni faze
16    Hold_Trig ( CLK := Hold ) ;
17
18    //Spravna poloha - Ukonceni faze
19    Ton_Complete ( In := Cidlo AND ( Stav_faze = fRunning ) , PT := T#1S ) ;
20    //Ukonceni faze
21    Ton_Idle ( In := ( Stav_faze = fComplete ) OR ( Stav_faze = fAborted ) , PT :=
T#5S ) ;
22    //TimeOut faze
23    fTimeOut ( IN := ( Stav_faze = fRunning ) , PT := T#5s ) ;
24
25    //Stavac pro faze - prechody
26    CASE Stav_faze OF
27        fIdle :
28            IF Start_Trig . Q THEN
29                Stav_faze := fRunning ;
30            END_IF
31
32        fRunning :
33            IF Ton_Complete . Q THEN
34                Stav_faze := fComplete ;
35            END_IF
36
37            IF Hold_Trig . Q THEN
38                Stav_faze := fHeld ;
39            END_IF
40
41            IF Abort_Trig . Q THEN
42                Stav_faze := fAborted ;
43            END_IF
44
45            IF fTimeOut . Q THEN
46                Porucha := TRUE ;
47            END_IF ;
```

```
48
49     fAborted :
50         IF Ton_Idle . Q THEN
51             Stav_faze := fIdle ;
52         END_IF
53
54     fComplete :
55         IF Ton_Idle . Q THEN
56             Stav_faze := fIdle ;
57         END_IF
58
59     fHeld :
60         IF Resume_Trig . Q THEN
61             Stav_faze := fRunning ;
62         END_IF
63
64         IF Abort_Trig . Q THEN
65             Stav_faze := fAborted ;
66         END_IF
67
68     END_CASE
69
70     //Stavac pro faze - zapisy
71     CASE Stav_faze OF
72         fIdle :
73
74         fRunning :
75
76             Running := TRUE ;
77             Run := TRUE ;
78
79         fHeld :
80             Held := TRUE ;
81
82         fAborted :
83             Aborted := TRUE ;
84
85         fComplete :
86             Complete := TRUE ;
87
88     END_CASE
89
```

#### 1.1.1.1.2.4 Folder: VstupniKontrola

### 1.1.1.1.2.4.1 POU: faze\_KontrolaVytahu

```
1  FUNCTION_BLOCK faze_KontrolaVytahu
2  VAR_INPUT
3      Cidlo : BOOL ;
4      Start : BOOL ;
5      Resume : BOOL ;
6      Hold : BOOL ;
7      Abort : BOOL ;
8  END_VAR
9  VAR_OUTPUT
10     Running : BOOL ;
11     Held : BOOL ;
12     Aborted : BOOL ;
13     Complete : BOOL ;
14     Porucha : BOOL ;
15 END_VAR
16 VAR CONSTANT
17     //Stavy faze
18     fIdle : UINT := 0 ;
19     fRunning : UINT := 1 ;
20     fHeld : UINT := 2 ;
21     fAborted : UINT := 3 ;
22     fComplete : UINT := 4 ;
23
24 END_VAR
25 VAR
26     Start_Trig : R_Trig ;
27     Resume_Trig : R_Trig ;
28     Abort_Trig : R_Trig ;
29     Hold_Trig : R_Trig ;
30     Stav_faze : UINT := 0 ;
31     Ton_Complete : TON ;
32     Ton_Idle : TON ;
33     fTimeOut : TON ;
34 END_VAR
35
```

```
1  //Uvolneni vystupu
2  Running := FALSE ;
3  Complete := FALSE ;
4  Aborted := FALSE ;
5  Held := FALSE ;
6
7
8  //Pozadavek na spusteni faze
9  Start_Trig ( CLK := Start ) ;
10 //Pozadavek na pokracovani faze
11 Resume_Trig ( CLK := Resume ) ;
12 //Pozadavek na preruseni faze
13 Abort_Trig ( CLK := Abort ) ;
14 //Pozadavek pro pozastaveni faze
15 Hold_Trig ( CLK := Hold ) ;
16
```

```
17      //Spravna poloha - Ukonceni faze
18      Ton_Complete ( In := Cidlo AND ( Stav_faze = fRunning ) , PT := T#2S ) ;
19      //Ukonceni faze
20      Ton_Idle ( In := ( Stav_faze = fComplete ) OR ( Stav_faze = fAborted ) , PT :=
21      T#5S ) ;
22      //Timeout faze
23      fTimeout ( IN := ( NOT cidlo AND ( Stav_faze = fRunning ) ) , PT := T#5S ) ;
24
25      //Stavac pro faze - prechody
26      CASE Stav_faze OF
27          fIdle :
28              IF Start_Trig . Q THEN
29                  Stav_faze := fRunning ;
30              END_IF
31
32          fRunning :
33              IF Ton_Complete . Q THEN
34                  Stav_faze := fComplete ;
35              END_IF
36
37              IF Hold_Trig . Q THEN
38                  Stav_faze := fHeld ;
39              END_IF
40
41              IF Abort_Trig . Q THEN
42                  Stav_faze := fAborted ;
43              END_IF
44
45              IF NOT Cidlo THEN
46                  Porucha := TRUE ;
47              END_IF
48
49              IF fTimeout . Q THEN
50                  Porucha := TRUE ;
51              END_IF ;
52
53          fAborted :
54              IF Ton_Idle . Q THEN
55                  Stav_faze := fIdle ;
56              END_IF
57
58          fComplete :
59              IF Ton_Idle . Q THEN
60                  Stav_faze := fIdle ;
61              END_IF
62
63          fHeld :
64              IF Resume_Trig . Q THEN
65                  Stav_faze := fRunning ;
66              END_IF
67
68              IF Abort_Trig . Q THEN
69                  Stav_faze := fAborted ;
70              END_IF
71      END_CASE
```

```
72
73 //Stavac pro faze - zapisy
74 CASE Stav_faze OF
75     fIdle :
76
77     fRunning :
78
79         Running := TRUE ;
80
81     fHeld :
82         Held := TRUE ;
83         Running := FALSE ;
84
85     fAborted :
86         Aborted := TRUE ;
87         Running := FALSE ;
88
89     fComplete :
90         Complete := TRUE ;
91         Running := FALSE ;
92
93 END_CASE
94
```

#### 1.1.1.1.2.4.2 POU: faze\_Material

```
1 FUNCTION_BLOCK faze_Material
2 VAR_INPUT
3     Lahve : BOOL ;
4     Start : BOOL ;
5     Resume : BOOL ;
6     Hold : BOOL ;
7     Abort : BOOL ;
8 END_VAR
9 VAR_OUTPUT
10    Running : BOOL ;
11    Held : BOOL ;
12    Aborted : BOOL ;
13    Complete : BOOL ;
14    Porucha : BOOL ;
15 END_VAR
16 VAR CONSTANT
17     //Stavy faze
18     fIdle : UINT := 0 ;
19     fRunning : UINT := 1 ;
20     fHeld : UINT := 2 ;
21     fAborted : UINT := 3 ;
22     fComplete : UINT := 4 ;
23
24 END_VAR
25 VAR
26     Start_Trig : R_Trig ;
27     Resume_Trig : R_Trig ;
28     Abort_Trig : R_Trig ;
29     Hold_Trig : R_Trig ;
30     Stav_faze : UINT := 0 ;
```

```
31     Ton_Complete : TON ;
32     Ton_Idle : TON ;
33     fTimeOut : TON ;
34 END_VAR
35
36
```

---

```
1      //Uvolneni vystupu
2      Running := FALSE ;
3      Complete := FALSE ;
4      Aborted := FALSE ;
5      Held := FALSE ;
6
7
8
9      //Pozadavek na spusteni faze
10     Start_Trig ( CLK := Start ) ;
11     //Pozadavek na pokracovani faze
12     Resume_Trig ( CLK := Resume ) ;
13     //Pozadavek na preruseni faze
14     Abort_Trig ( CLK := Abort ) ;
15     //Pozadavek pro pozastaveni faze
16     Hold_Trig ( CLK := Hold ) ;
17
18     //Spravna poloha - Ukonceni faze
19     Ton_Complete ( In := Lahve AND ( Stav_faze = fRunning ) , PT := T#3S ) ;
20     //Ukonceni faze
21     Ton_Idle ( In := ( Stav_faze = fComplete ) OR ( Stav_faze = fAborted ) , PT :=
22     T#5S ) ;
23     //TimeOut faze
24     fTimeOut ( IN := ( NOT lahve AND ( Stav_faze = fRunning ) ) , PT := T#5S ) ;
25
26     //Stavac pro faze - prechody
27     CASE Stav_faze OF
28         fIdle :
29             IF Start_Trig . Q THEN
30                 Stav_faze := fRunning ;
31             END_IF
32         fRunning :
33             IF Ton_Complete . Q THEN
34                 Stav_faze := fComplete ;
35             END_IF
36
37             IF Hold_Trig . Q THEN
38                 Stav_faze := fHeld ;
39             END_IF
40
41             IF Abort_Trig . Q THEN
42                 Stav_faze := fAborted ;
43             END_IF
44
45             IF NOT LAHVE THEN
46                 Porucha := TRUE ;
47             END_IF
```



```
48
49     IF fTimeOut . Q THEN
50         Porucha := TRUE ;
51     END_IF ;
52
53     fAborted :
54         IF Ton_Idle . Q THEN
55             Stav_faze := fIdle ;
56         END_IF
57
58     fComplete :
59         IF Ton_Idle . Q THEN
60             Stav_faze := fIdle ;
61         END_IF
62
63     fHeld :
64         IF Resume_Trig . Q THEN
65             Stav_faze := fRunning ;
66         END_IF
67
68         IF Abort_Trig . Q THEN
69             Stav_faze := fAborted ;
70         END_IF
71
72     END_CASE
73
74     //Stavac pro faze - zapisy
75     CASE Stav_faze OF
76         fIdle :
77
78         fRunning :
79
80             Running := TRUE ;
81
82         fHeld :
83             Held := TRUE ;
84
85         fAborted :
86             Aborted := TRUE ;
87
88         fComplete :
89             Complete := TRUE ;
90
91     END_CASE
92
```

### 1.1.1.1.3 POU: Dopravnik\_pohyb

```
1      FUNCTION_BLOCK Dopravnik_pohyb
2      VAR_INPUT
3          Start1 : BOOL ;
4          Start2 : BOOL ;
5      END_VAR
6      VAR_OUTPUT
7          Run1 : BOOL ;
8          Run2 : BOOL ;
9      END_VAR
10     VAR
11     END_VAR
12
13
```

---

```
1      IF Start1 AND NOT Start2 THEN
2          Run1 := TRUE ;
3      ELSE
4          Run1 := FALSE ;
5      END_IF
6
7      IF Start2 THEN
8          Run2 := TRUE ;
9      ELSE
10         Run2 := FALSE ;
11     END_IF
12
13
```

### 1.1.1.1.4 POU: Errs

```
1      FUNCTION_BLOCK Errs
2      VAR_INPUT
3          TLak : BOOL ;
4          Komunikace : BOOL ;
5          KLic : BOOL ;
6          TL_STOP : BOOL ;
7      END_VAR
8      VAR_OUTPUT
9      END_VAR
10     VAR
11         fTRIG_STOP : R_TRIG ;
12     END_VAR
13
```

---

```
1      fTRIG_STOP ( Clk := TL_STOP ) ;
2
3      IF TLak THEN
4          Batch_Porucha . Nizky_Tlak := TRUE ;
5      END_IF
6
```

```
7      IF Komunikace THEN
8          Batch_Porucha . Komunikace := TRUE ;
9      END_IF
10
11     IF fTRIG_STOP . Q THEN
12         Batch_Porucha . Komunikace := FALSE ;
13         Batch_Porucha . Nizky_Tlak := FALSE ;
14         Batch_Porucha . Lahev_1 := FALSE ;
15         Batch_Porucha . Lahev_2 := FALSE ;
16         Batch_Porucha . Lahev_3 := FALSE ;
17         Batch_Porucha . Lahev_4 := FALSE ;
18
19     END_IF
20
21
```

#### 1.1.1.1.5 POU: Komunikace

```
1      FUNCTION_BLOCK Komunikace
2      VAR_INPUT
3          Comm_CNT : INT ;
4      END_VAR
5      VAR_OUTPUT
6          Comm_OK : BOOL ;
7      END_VAR
8      VAR
9          Val : INT ;
10         PrevVal : INT ;
11         TON_Comm : TON ;
12         TOF_Comm : TOF ;
13     END_VAR
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
```

### 1.1.1.1.6 POU: Majak

```
1      FUNCTION_BLOCK Majak
2      VAR_INPUT
3          Comm_Ok : BOOL ;
4      END_VAR
5      VAR_OUTPUT
6          Out : Bool ;
7          Green : BOOL ;
8          Orange : BOOL ;
9          Red : BOOL ;
10     END_VAR
11     VAR
12     END_VAR
13
```

---

```
1      Green := FALSE ;
2      Orange := FALSE ;
3      Red := FALSE ;
4
5
6      IF NOT Porucha THEN
7          IF Key AND Comm_Ok THEN
8              Green := TRUE ;
9          ELSIF Key AND NOT Comm_Ok THEN
10             Orange := TRUE ;
11          ELSIF NOT KEy THEN
12             Red := TRUE ;
13          END_IF
14      ELSE
15          Red := TRUE ;
16      END_IF
17
18
19
20      Porucha := FALSE ;
21
```

### 1.1.1.1.7 POU: Podavac\_pohyb

```
1      FUNCTION_BLOCK Podavac_pohyb
2      VAR_INPUT
3          Start1 : BOOL ;
4          Start2 : BOOL ;
5      END_VAR
6      VAR_OUTPUT
7          Run1 : BOOL ;
8          Run2 : BOOL ;
9      END_VAR
10     VAR
11     END_VAR
12
```

---

```
1      IF Start1 AND NOT Start2 THEN
2          Run1 := TRUE ;
3      ELSE
4          Run1 := FALSE ;
5      END_IF
6
7      IF Start2 THEN
8          Run2 := TRUE ;
9      ELSE
10         Run2 := FALSE ;
11     END_IF
12
```

### 1.1.1.1.8 POU: Pozice

```
1      FUNCTION_BLOCK Pozice
2      VAR_INPUT
3          Pozice0 : BOOL ; Pozice1 : BOOL ; Pozice2 : BOOL ; Pozice3 : BOOL ; Pozice4 :
4          BOOL ; Pozice5 : BOOL ;
5          Dopr_vlevo : BOOL ; Dopr_vpravo : BOOL ;
6      END_VAR
7      VAR_OUTPUT
8          Out : BOOL ;
9          Vystup : INT ;
10     END_VAR
11     VAR
12         Akt_pozice : INT := 0 ;           //aktualní pozice
13         Smer : INT ;
14     END_VAR
```

---

```
1      //0 - počáteční pozice
2      //1 - pozice mezi počátkem a prvním senzorem
3      //2 - pozice prvního senzoru
4      //3 - pozice mezi prvním senzorem a druhým senzorem
5      //4 - pozice druhého senzoru
6      //5 - pozice mezi druhým senzorem a třetím senzorem
7      //6 - pozice třetího senzoru
8      //7 - pozice mezi třetím senzorem a čtvrtým senzorem
9      //8 - pozice čtvrtého senzoru
10     //9 - pozice mezi čtvrtým senzorem a druhým senzorem
11     //10 - konečná pozice
12
13     IF Dopr_vlevo THEN
14         smer := 1 ;
15     ELSIF Dopr_vpravo THEN
16         smer := 2 ;
17     END_IF
18
19
20     IF pozice0 THEN
21         akt_pozice := 0 ;
22     END_IF
23
```

```
24     IF pozice1 THEN
25         akt_pozice := 2 ;
26     END_IF
27
28     IF pozice2 THEN
29         akt_pozice := 4 ;
30     END_IF
31
32     IF pozice3 THEN
33         akt_pozice := 6 ;
34     END_IF
35
36     IF pozice4 THEN
37         akt_pozice := 8 ;
38     END_IF
39
40     IF pozice5 THEN
41         akt_pozice := 10 ;
42     END_IF
43
44     (*
45     IF NOT Pozice0 AND NOT Pozice1 AND NOT Pozice2 AND NOT Pozice3 AND NOT
46     Pozice4 AND NOT Pozice5 AND Akt_pozice = 0 THEN
47         Akt_Pozice:=99;
48     END_IF
49     *)
50     IF ( NOT pozice0 AND NOT pozice1 ) AND ( ( akt_pozice = 0 AND smer = 2 )
51     OR ( akt_pozice = 2 AND smer = 1 ) ) THEN
52         akt_pozice := 1 ;
53     END_IF
54     IF ( NOT pozice1 AND NOT pozice2 ) AND ( ( akt_pozice = 2 AND smer = 2 )
55     OR ( akt_pozice = 4 AND smer = 1 ) ) THEN
56         akt_pozice := 3 ;
57     END_IF
58     IF ( NOT pozice2 AND NOT pozice3 ) AND ( ( akt_pozice = 4 AND smer = 2 )
59     OR ( akt_pozice = 6 AND smer = 1 ) ) THEN
60         akt_pozice := 5 ;
61     END_IF
62     IF ( NOT pozice3 AND NOT pozice4 ) AND ( ( akt_pozice = 6 AND smer = 2 )
63     OR ( akt_pozice = 8 AND smer = 1 ) ) THEN
64         akt_pozice := 7 ;
65     END_IF
66     IF ( NOT pozice4 AND NOT pozice5 ) AND ( ( akt_pozice = 8 AND smer = 2 )
67     OR ( akt_pozice = 10 AND smer = 1 ) ) THEN
68         akt_pozice := 9 ;
69     END_IF
70     Vystup := akt_pozice ;
71
72
73
```

### 1.1.1.1.9 POU: StateOfBatch

```
1  FUNCTION_BLOCK StateOfBatch
2  VAR_INPUT
3  END_VAR
4  VAR_OUTPUT
5      Out : BOOL ;
6      Running : BOOL ;
7      Err : BOOL ;
8  END_VAR
9  VAR
10     fErr : BOOL ;
11     fRunning : BOOL ;
12     fTRIG_Err : R_TRIG ;
13     fTRIG_Key : R_TRIG ;
14 END_VAR
15
```

---

```
1
2     fRunning := FALSE ;
3     fErr := FALSE ;
4
5
6     IF NOT KEY THEN
7         Rizeni . Maintenance := TRUE ;
8         Rizeni . Batch := FALSE ;
9         Rizeni . Man := FALSE ;
10    ELSIF KEY AND NOT Comm_Ok THEN
11        Rizeni . Maintenance := FALSE ;
12        Rizeni . Batch := FALSE ;
13        Rizeni . Man := TRUE ;
14    ELSIF KEY AND Comm_Ok THEN
15        Rizeni . Maintenance := FALSE ;
16        Rizeni . Batch := TRUE ;
17        Rizeni . Man := FALSE ;
18    END_IF
19
20    fTRIG_Err ( CLK := Batch_Porucha . Komunikace OR Batch_Porucha . Lahev_1 OR
Batch_Porucha . Lahev_2 OR Batch_Porucha . Lahev_3 OR Batch_Porucha . Lahev_4
OR Batch_Porucha . Nizky_Tlak OR Batch_Porucha . Podavac_1 OR Batch_Porucha
. Podavac_2 OR
21        Batch_Porucha . Vytah_Sklenice OR Batch_Porucha . TimeOut OR
Batch_Porucha . Zasobnik ) ;
22
23    fTRIG_Key ( CLK := NOT Key ) ;
24
25    IF fKontrolaVytahu . RUNNING OR fLahev_1_Kontrola . RUNNING OR
fLahev_2_Kontrola . RUNNING OR fLahev_3_Kontrola . RUNNING OR
fLahev_4_Kontrola . RUNNING OR fSenzorZasobnik . RUNNING OR fSenzorVytah .
RUNNING OR
26        fPodavac_1_nahoru . RUNNING OR fPodavac_1_dolu . RUNNING OR
fPodavac_2_nahoru . RUNNING OR fPodavac_2_dolu . RUNNING OR fDopravnik .
RUNNING OR fVytah_nahoru . RUNNING OR fVytah_dolu . RUNNING OR
```

```
27      fPritomnost_sklenice . RUNNING OR fTlacitko_STOP . RUNNING OR
fPocatecni_poloha . RUNNING THEN
28
29      fRunning := TRUE ;
30  END_IF
31
32  IF Batch_Porucha . Komunikace OR Batch_Porucha . Lahev_1 OR Batch_Porucha .
Lahev_2 OR Batch_Porucha . Lahev_3 OR Batch_Porucha . Lahev_4 OR
Batch_Porucha . Nizky_Tlak OR Batch_Porucha . Podavac_1 OR Batch_Porucha .
Podavac_2 OR
33      Batch_Porucha . Vytah_Sklenice OR Batch_Porucha . TimeOut OR
Batch_Porucha . Zasobnik THEN
34
35      Porucha := TRUE ;
36      fErr := TRUE ;
37  END_IF
38
39
40
41
42  //celkove poruchy
43  IF fTRIG_Err . Q THEN
44      Poruchy_Pocet := Poruchy_Pocet + 1 ;
45  END_IF
46
47  //Davkovac porucha
48  IF fTRIG_Err . Q AND ( Batch_Porucha . Podavac_1 OR Batch_Porucha . Podavac_2
OR Batch_Porucha . Vytah_Sklenice OR Batch_Porucha . Zasobnik ) THEN
49      Batch_Porucha . Podavac_Pocet := Batch_Porucha . Podavac_Pocet + 1 ;
50  END_IF
51
52  //Podavac porucha
53  IF fTRIG_Err . Q AND ( Batch_Porucha . Lahev_1 OR Batch_Porucha . Lahev_2 OR
Batch_Porucha . Lahev_3 OR Batch_Porucha . Lahev_4 ) THEN
54      Batch_Porucha . Davkovac_Pocet := Batch_Porucha . Davkovac_Pocet + 1 ;
55  END_IF
56
57  //Ostatni poruchy
58  IF fTRIG_Err . Q AND ( Batch_Porucha . TimeOut OR Batch_Porucha . Komunikace
OR Batch_Porucha . Nizky_Tlak ) THEN
59      Batch_Porucha . Ostatni_Pocet := Batch_Porucha . Ostatni_Pocet + 1 ;
60  END_IF
61
62
63  IF fTRIG_Key . Q THEN
64      Opravy_Pocet := Opravy_Pocet + 1 ;
65  END_IF
66
67
68  Running := fRunning ;
69  Err := fErr ;
70
71
72
73
74
```



### 1.1.1.1.10 POU: Vytah\_pohyb

```
1      FUNCTION_BLOCK Vytah_pohyb
2      VAR_INPUT
3          Start1 : BOOL ;
4          Start2 : BOOL ;
5      END_VAR
6      VAR_OUTPUT
7          Run1 : BOOL ;
8          Run2 : BOOL ;
9      END_VAR
10     VAR
11     END_VAR
12
13
```

---

```
1      IF Start1 AND NOT Start2 THEN
2          Run1 := TRUE ;
3      ELSE
4          Run1 := FALSE ;
5      END_IF
6
7      IF Start2 THEN
8          Run2 := TRUE ;
9      ELSE
10         Run2 := FALSE ;
11     END_IF
12
```

### 1.1.1.2 Folder: Main Function

#### 1.1.1.2.1 POU: Batch

```
1      FUNCTION_BLOCK Batch
2      VAR_INPUT
3      END_VAR
4      VAR_OUTPUT
5      END_VAR
6      VAR
7          //Operace Vstupni Kontrola
8          Vytah_Poloha : faze_KontrolaVytahu ;
9          Lahve_1_kontrola : faze_Material ;
10         Lahve_2_kontrola : faze_Material ;
11         Lahve_3_kontrola : faze_Material ;
12         Lahve_4_kontrola : faze_Material ;
13
14         //Operace Sklenice
15         Zasobnik_NeniPrazdny : faze_Zasobnik ;
16         SkleniceVeVytahu : faze_PlnyVytah ;
17         Podavac_1_Nahoru : faze_Podavac ;
18         Podavac_1_Dolu : faze_Podavac ;

```

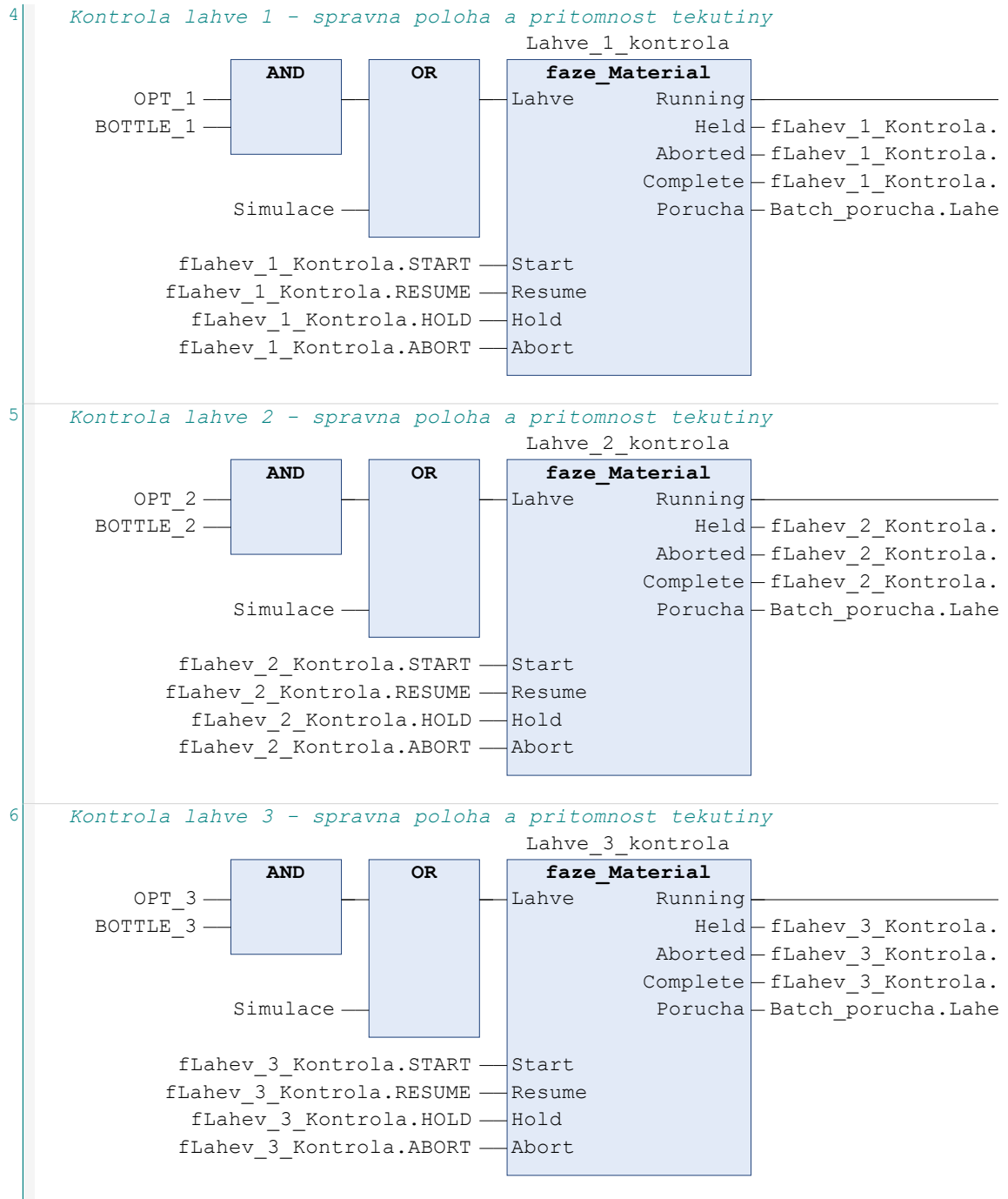
### 1.1.1.2.1 POU: Batch

```
19      Podavac_2_Nahoru : faze_Podavac ;
20      Podavac_2_Dolu : faze_Podavac ;
21
22      //Operace lahve
23      Dopravnik : faze_Dopravnik ;
24      Vytah_Nahoru : faze_Vytah ;
25      Vytah_Dolu : faze_Vytah ;
26
27      //Operace Dokonceni
28      Pritomnost_Sklenice : faze_SenzorVytah ;
29      Tlacitko_STOP : faze_TL_Stop ;
30      Pocatecni_Poloha : faze_PocatecniPoloha ;
31
32      Dopravnik_smer : Dopravnik_pohyb ;
33      Podavac1_smer : Podavac_pohyb ;
34      Podavac2_smer : Podavac_pohyb ;
35      Vytah_smer : Vytah_pohyb ;
36
37      //Propojky
38      Podavac_1_Nahoru_Start : BOOL ;
39      Podavac_1_Dolu_Start : BOOL ;
40      Podavac_2_Nahoru_Start : BOOL ;
41      Podavac_2_Dolu_Start : BOOL ;
42      Vytah_Nahoru_Start : BOOL ;
43      Vytah_Dolu_Start : BOOL ;
44      Dopravnik_Vlevo_Start_1 : BOOL ;
45      Dopravnik_Vlevo_Start_2 : BOOL ;
46      Dopravnik_Vpravo_Start : BOOL ;
47
48
49      END_VAR
50
```



G  
—— fKontrolaVytahu.HELD  
D  
TE

---



————— fLahev\_1\_Kontrola.RUNNING  
HELD  
ABORTED  
COMPLETE  
·v\_1

---

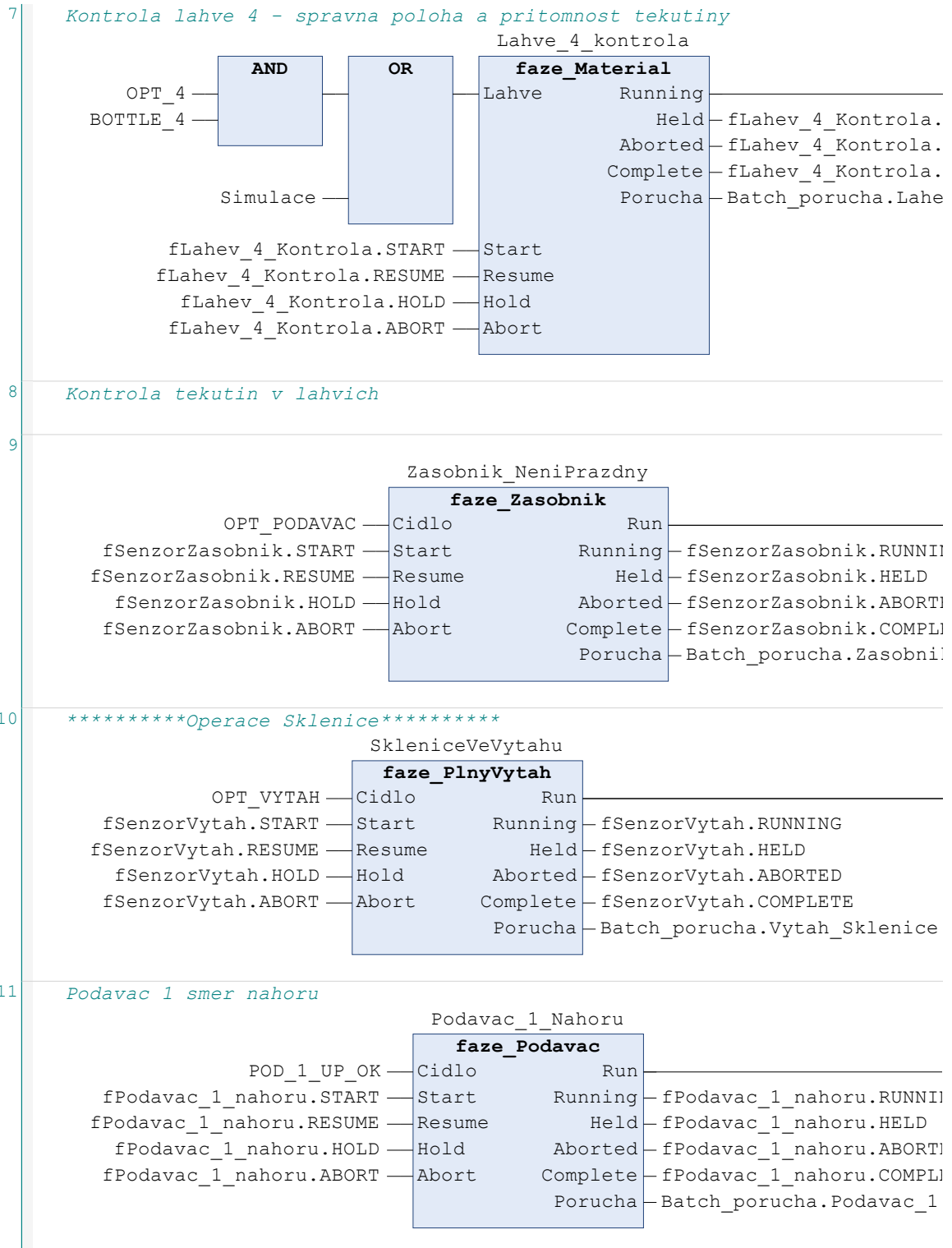
————— fLahev\_2\_Kontrola.RUNNING  
HELD  
ABORTED  
COMPLETE  
·v\_2

---

————— fLahev\_3\_Kontrola.RUNNING  
HELD  
ABORTED  
COMPLETE  
·v\_3

---

### 1.1.1.2.1 POU: Batch



----- fLahev\_4\_Kontrola.RUNNING  
HELD  
ABORTED  
COMPLETE  
v\_4

---

-----  
NG

ED  
ETE  
k

---

-

-

----- Podavac\_1\_Nahoru\_Start  
NG  
  
ED  
ETE

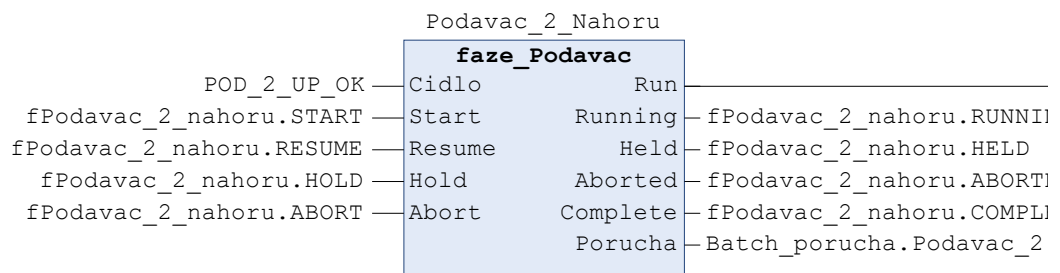
---

### 1.1.1.2.1 POU: Batch

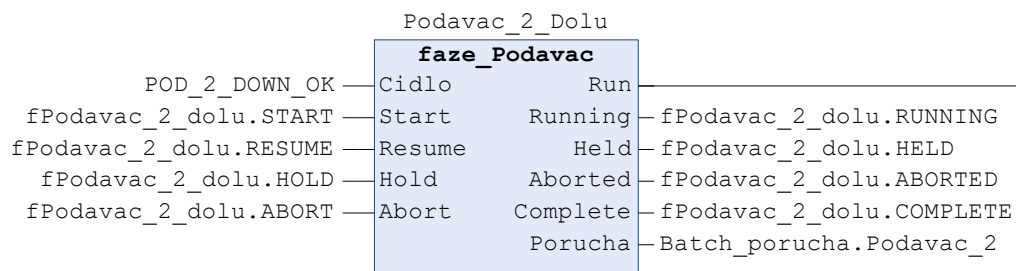
12 Podavac 1 smer dolu



13 Podavac 2 smer nahoru

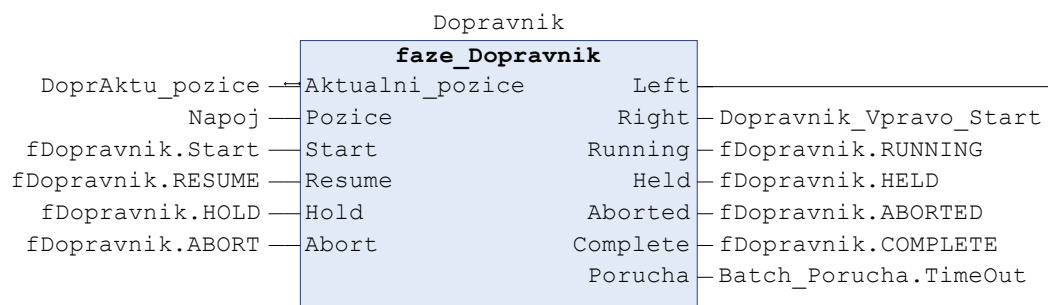


14 Podavac 2 smer dolu



15 \*\*\*\*\*Operace lahve\*\*\*\*\*

16





-Podavac\_1\_Dolu\_Start

---

———— Podavac\_2\_Nahoru\_Start  
NG

ED  
ETE

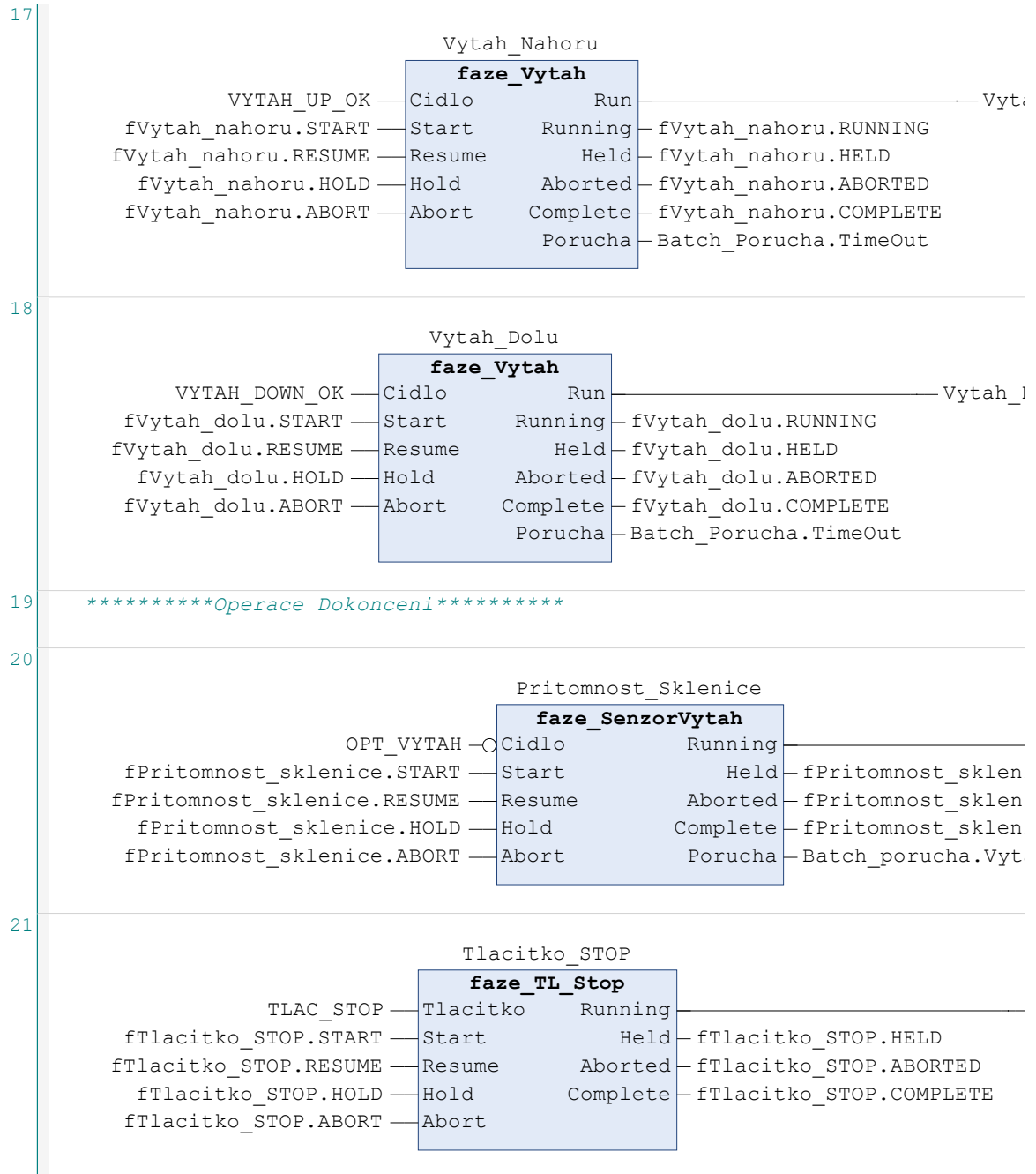
---

-Podavac\_2\_Dolu\_Start

---

——Dopravnik\_Vlevo\_Start\_1

### 1.1.1.2.1 POU: Batch



ah\_Nahoru\_Start

---

oolu\_Start

---

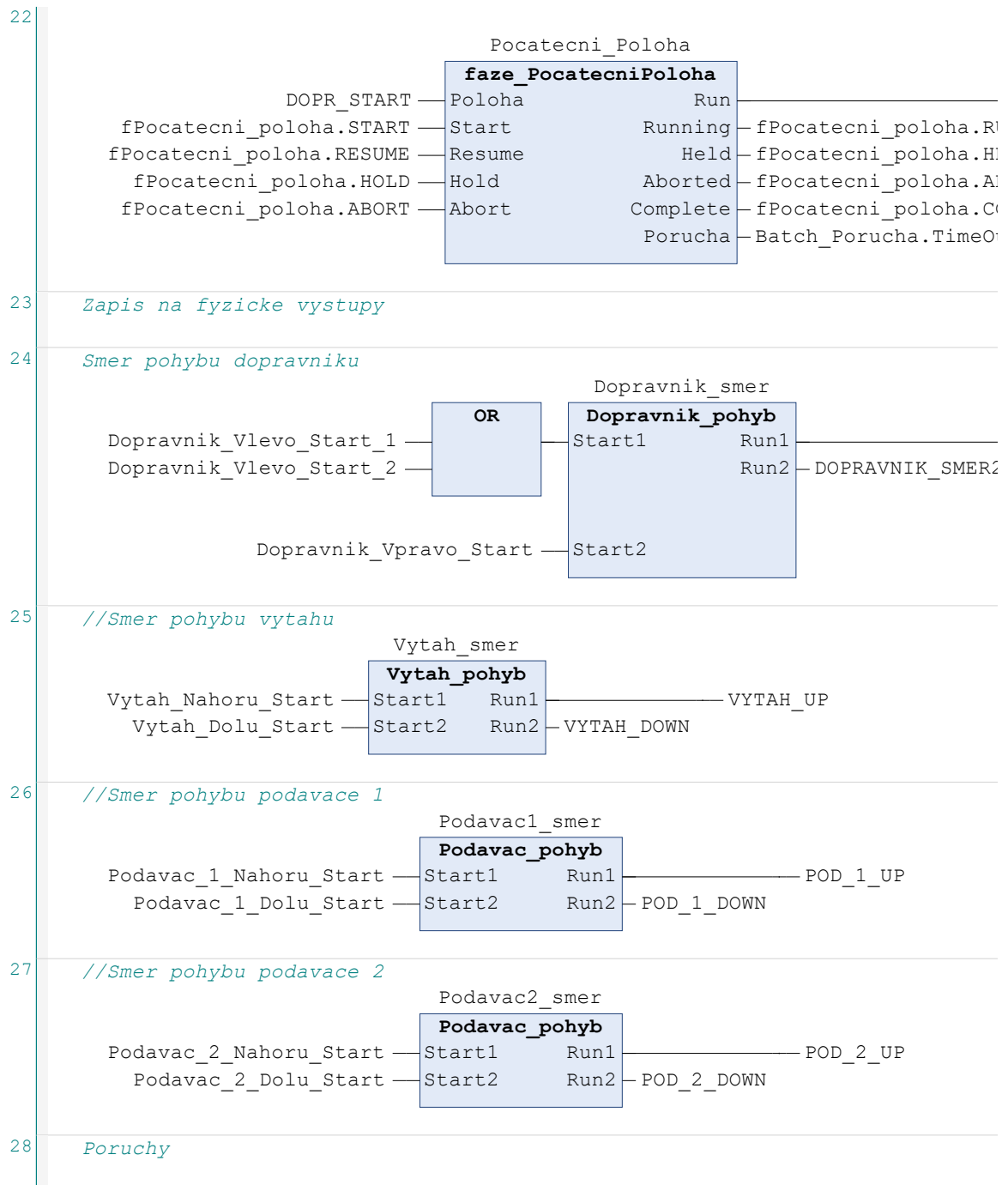
fPritomnost\_sklenice.RUNNING  
ice.HELD  
ice.ABORTED  
ice.COMPLETE  
ah\_Sklenice

---

-fTlacitko\_STOP.RUNNING

---

### 1.1.1.2.1 POU: Batch



————Dopravnik\_Vlevo\_Start\_2  
UNNING  
ELD  
BORTED  
OMplete  
ut

---

———DOPRAVNIK\_SMER1  
?

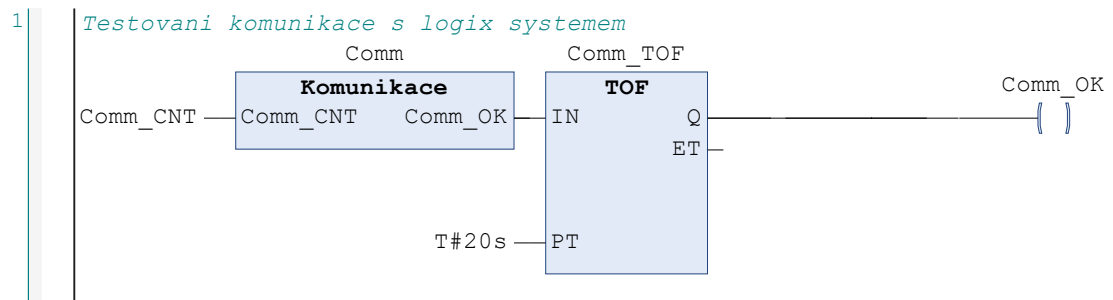
---

## 1.1.1.2.2 POU: Main

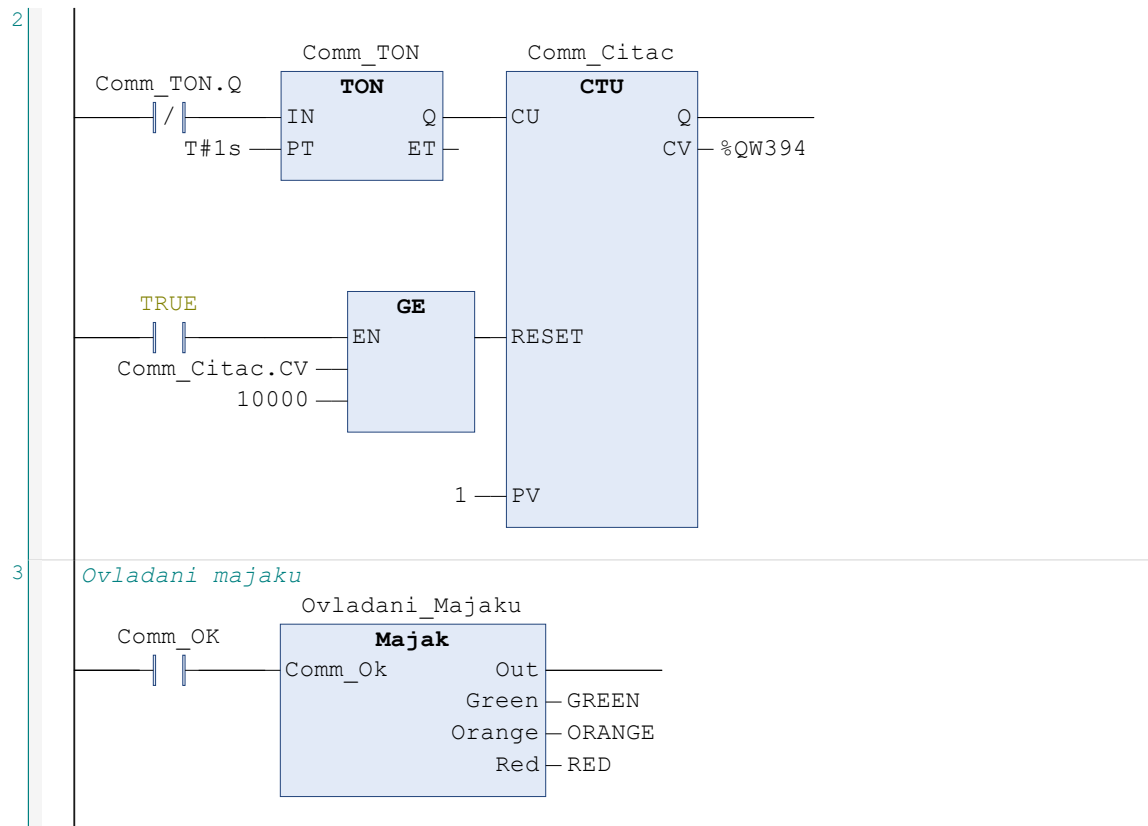
```

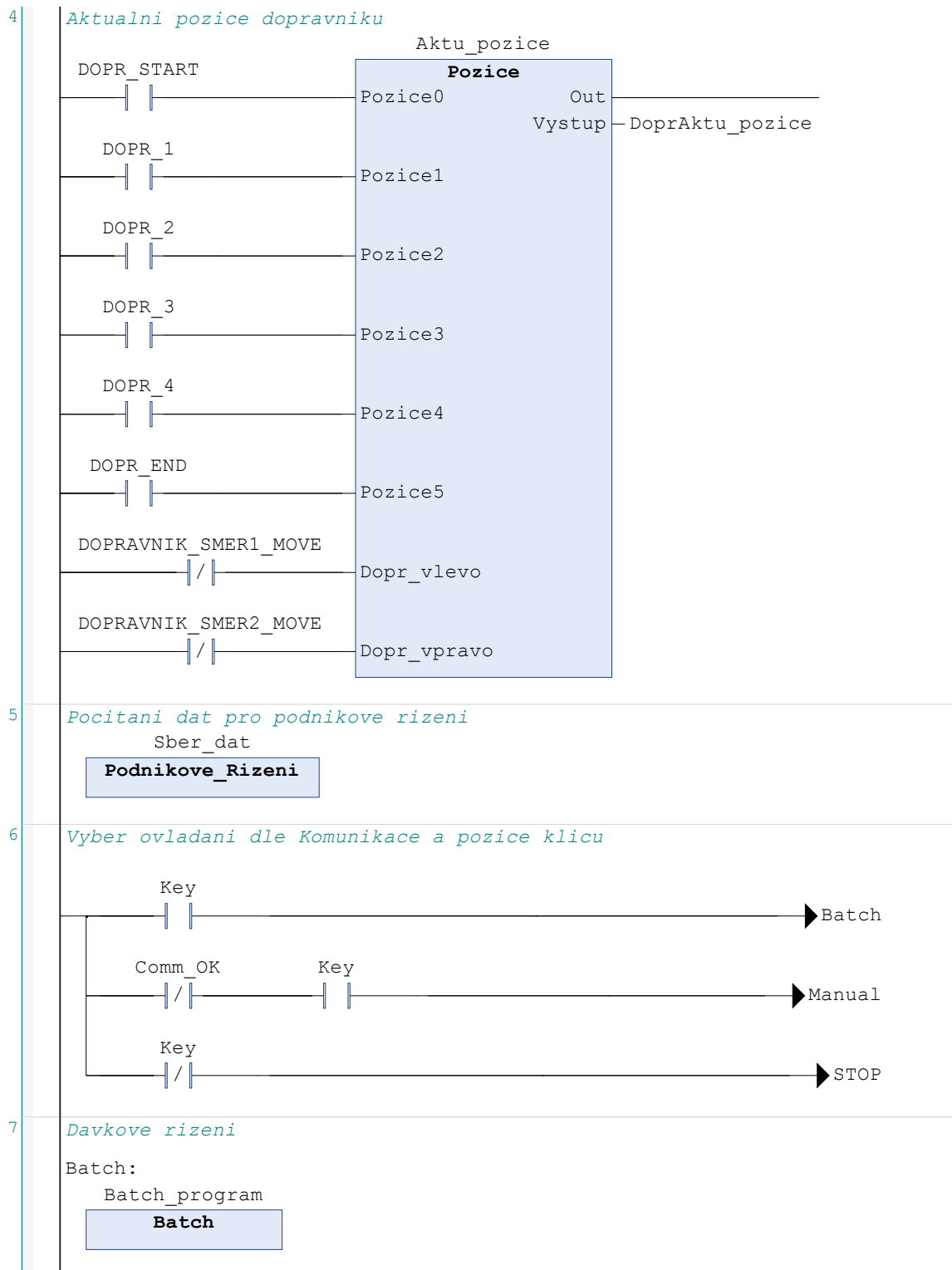
1  PROGRAM Main
2  VAR
3      //Kontora komunikace
4      Comm : komunikace ;
5
6      //Batch rizeni
7      Batch_program : Batch ;
8      //Manualni rizeni
9      Manual_program : Manual ;
10
11     //Aktualni pozice dopravniku
12     Aktu_pozice : Pozice ;
13
14     //Ovladani svetel majaku
15     Ovladani_Majaku : Majak ;
16
17     //Komunikacni casovac
18     Comm_TOF : TOF ;
19
20     Comm_TON : TON ;
21     Comm_Citac : CTU ;
22
23     //Sber dat pro podnikove rizeni
24     Sber_dat : Podnikove_Rizeni ;
25
26     //Zastaveni programu
27     Stop_program : Stop ;
28 END_VAR
29

```

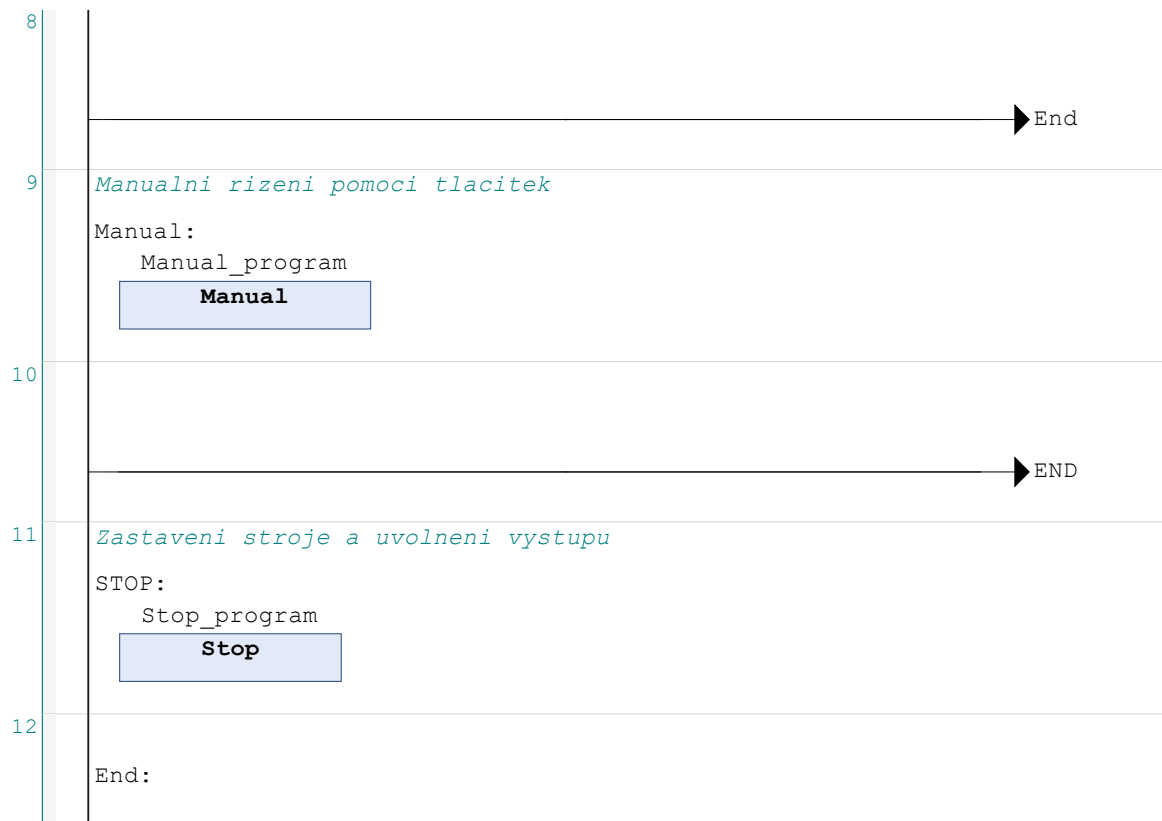


### 1.1.1.2.2 POU: Main







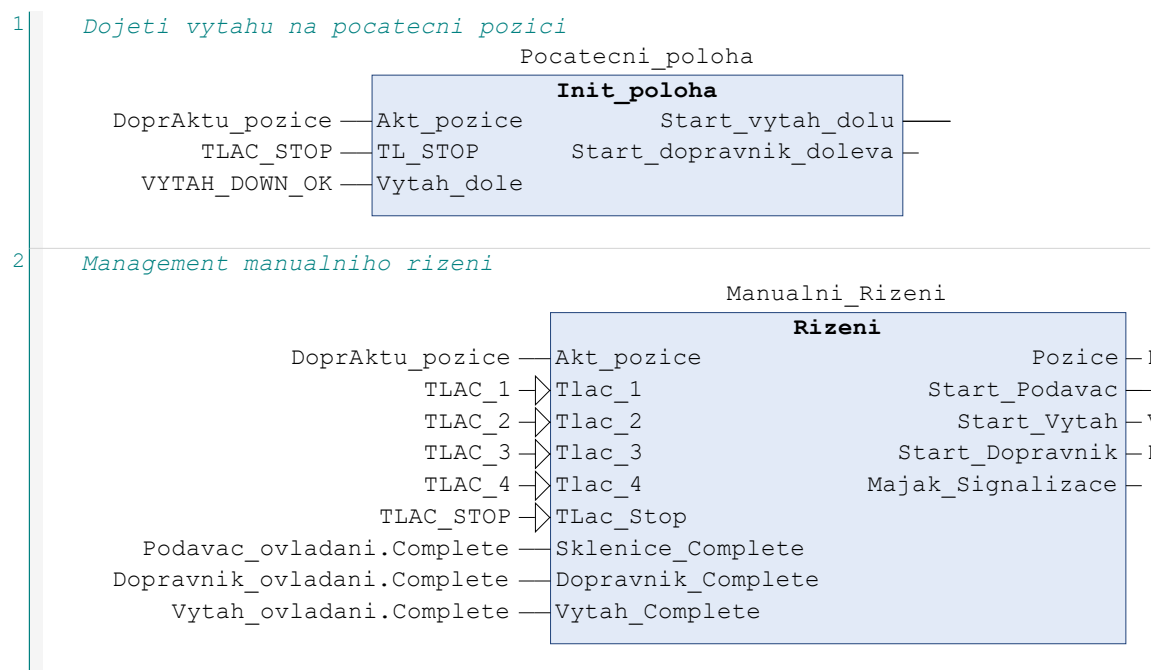


## 1.1.1.2.3 POU: Manual

```

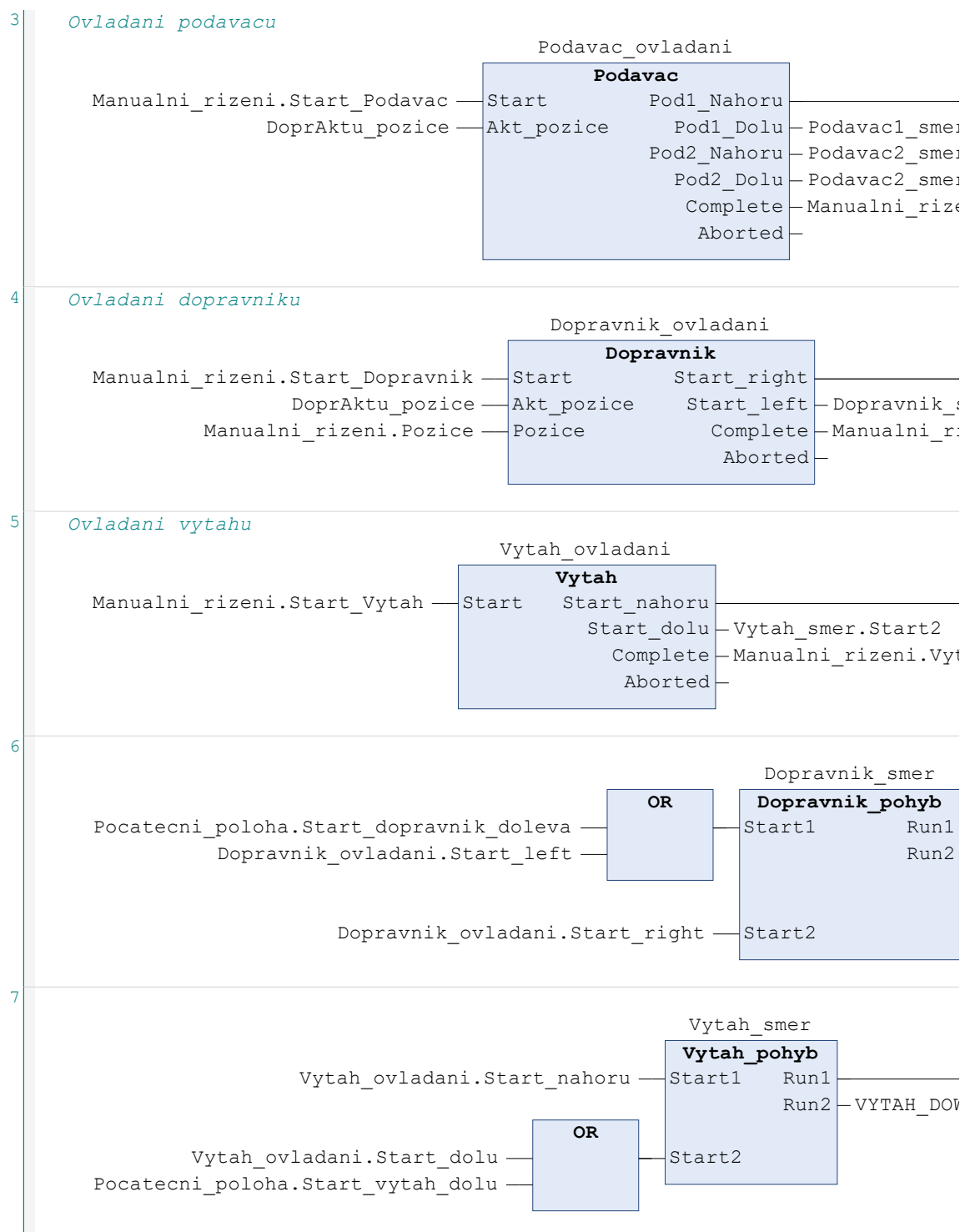
1  FUNCTION_BLOCK Manual
2  VAR_INPUT
3  END_VAR
4  VAR_OUTPUT
5  END_VAR
6  VAR
7      //Pocatecni poloha vytahu
8      Pocatecni_poloha : Init_poloha ;
9
10     //Zapis na fyzicke vystupy
11     Dopravnik_smer : Dopravnik_pohyb ;
12     Podavac1_smer : Podavac_pohyb ;
13     Podavac2_smer : Podavac_pohyb ;
14     Vytah_smer : Vytah_pohyb ;
15
16     //Manualni ovladani
17     Manualni_Rizeni : Rizeni ;
18     Podavac_ovladani : Podavac ;
19     Dopravnik_ovladani : Dopravnik ;
20     Vytah_ovladani : Vytah ;
21
22 END_VAR
23

```



Dopravnik\_ovladani.Pozice  
----- Podavac\_ovladani.Start  
Vytah\_ovladani.Start  
Dopravnik\_ovladani.Start

---



---

Podavac1\_smer.Start1  
r.Start2  
r.Start1  
r.Start2  
eni.Sklenice\_Complete

---

---

smer.Start2  
izeni.Dopravnik\_Complete

---

---

Vytah\_smer.Start1  
tah\_Complete

---

---

DOPRAVNIK\_SMER1  
DOPRAVNIK\_SMER2

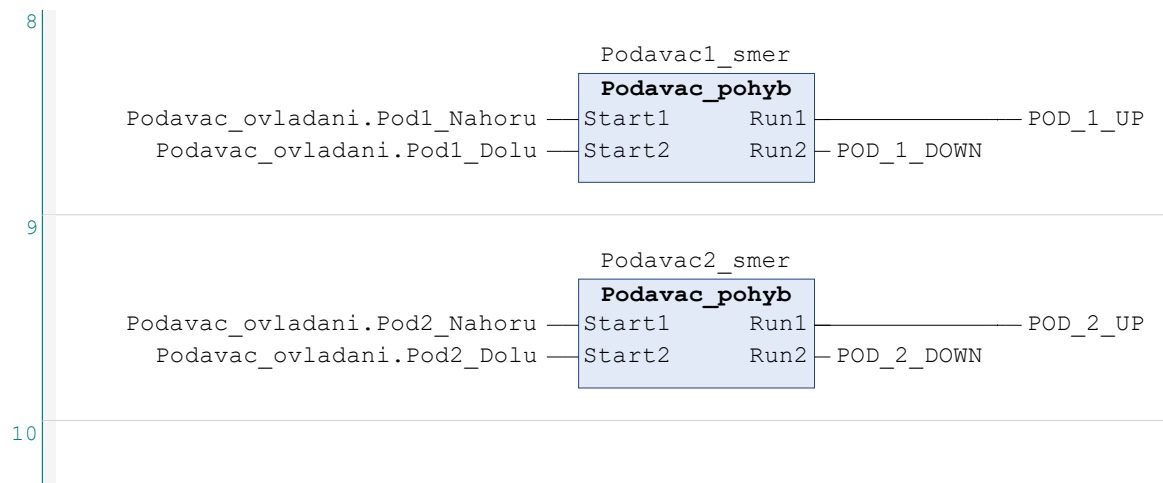
---

---

VYTAH\_UP  
VN

---

### 1.1.1.2.3 POU: Manual





### 1.1.1.2.4 POU: Mapovani

```
1      PROGRAM Mapovani
2      VAR
3
4          REAL_VALUE AT %MW200 : REAL ;
5          WORD_VALUE AT %MW200 : ARRAY [ 0 .. 1 ] OF WORD ;
6
7
8          //Prikazy fazi
9          //Dopravnik_CMD AT %IW372 :WORD;
10         KontrolaVytahu_CMD AT %IW373 : WORD ;
11         Lahev_1_Kontrola_CMD AT %IW374 : WORD ;
12         Lahev_2_Kontrola_CMD AT %IW375 : WORD ;
13         Lahev_3_Kontrola_CMD AT %IW376 : WORD ;
14         Lahev_4_Kontrola_CMD AT %IW377 : WORD ;
15         Podavac_1_nahoru_CMD AT %IW378 : WORD ;
16         Podavac_1_dolu_CMD AT %IW379 : WORD ;
17         Podavac_2_nahoru_CMD AT %IW380 : WORD ;
18         Podavac_2_dolu_CMD AT %IW381 : WORD ;
19         Dopravnik_CMD AT %IW382 : WORD ;
20         Vytah_nahoru_CMD AT %IW383 : WORD ;
21         Vytah_dolu_CMD AT %IW384 : WORD ;
22         Pritomnost_sklenice_CMD AT %IW385 : WORD ;
23         Tlacitko_STOP_CMD AT %IW386 : WORD ;
24         Pocatecni_poloha_CMD AT %IW387 : WORD ;
25         Zasobnik_Senzor_CMD AT %IW388 : WORD ;
26         Vytah_Senzor_CMD AT %IW389 : WORD ;
27
28         Comm AT %IW400 : WORD ;
29         Napoj_CMD AT %IW401 : WORD ;
30
31         //Stavy fazi
32         //Dopravnik_STA AT %QW372 :WORD;
33         KontrolaVytahu_STA AT %QW373 : WORD ;
34         Lahev_1_Kontrola_STA AT %QW374 : WORD ;
35         Lahev_2_Kontrola_STA AT %QW375 : WORD ;
36         Lahev_3_Kontrola_STA AT %QW376 : WORD ;
37         Lahev_4_Kontrola_STA AT %QW377 : WORD ;
38         Podavac_1_nahoru_STA AT %QW378 : WORD ;
39         Podavac_1_dolu_STA AT %QW379 : WORD ;
40         Podavac_2_nahoru_STA AT %QW380 : WORD ;
41         Podavac_2_dolu_STA AT %QW381 : WORD ;
42         Dopravnik_STA AT %QW382 : WORD ;
43         Vytah_nahoru_STA AT %QW383 : WORD ;
44         Vytah_dolu_STA AT %QW384 : WORD ;
45         Pritomnost_sklenice_STA AT %QW385 : WORD ;
46         Tlacitko_STOP_STA AT %QW386 : WORD ;
47         Pocatecni_poloha_STA AT %QW387 : WORD ;
48         Zasobnik_Senzor_STA AT %QW388 : WORD ;
49         Vytah_Senzor_STA AT %QW389 : WORD ;
50
51         //QW394 - Komunikace
52         Porucha_Sta AT %QW395 : WORD ;
53
```



```
54
55      Chod_celkem_STA  AT  %QW405  : WORD ;
56      Porucha_Celkem_STA  AT  %QW406  : WORD ;
57      Oprava_Celkem_STA  AT  %QW407  : WORD ;
58      Poruchy_Pocet_STA  AT  %QW408  : WORD ;
59      Opravy_Pocet_STA  AT  %QW409  : WORD ;
60      Porucha_Podavac_STA  AT  %QW410  : WORD ;
61      Porucha_Davkovac_STA  AT  %QW411  : WORD ;
62      Porucha_Ostatni_Sta  AT  %QW412  : WORD ;
63
64      MTBF_STA  AT  %QW413  : WORD ;
65      MTTR_STA  AT  %QW414  : WORD ;
66
67      Davky_Dobre_STA  AT  %QW415  : WORD ;
68      Davky_Celkem_STA  AT  %QW416  : WORD ;
69      Davky_Lahve_1_STA  AT  %QW417  : WORD ;
70      Davky_Lahve_2_STA  AT  %QW418  : WORD ;
71      Davky_Lahve_3_STA  AT  %QW419  : WORD ;
72      Davky_Lahve_4_STA  AT  %QW420  : WORD ;
73      Davky_Lahve_Celkem_STA  AT  %QW421  : WORD ;
74      Lahev_1_ml_STA  AT  %QW422  : WORD ;
75      Lahev_2_ml_STA  AT  %QW423  : WORD ;
76      Lahev_3_ml_STA  AT  %QW424  : WORD ;
77      Lahev_4_ml_STA  AT  %QW425  : WORD ;
78
79      Kvalita_STA_1  AT  %QW426  : WORD ;
80      Kvalita_STA_2  AT  %QW427  : WORD ;
81
82      Kvalita_Spojeni_STA_1  AT  %QW430  : WORD ;
83      Kvalita_Spojeni_STA_2  AT  %QW431  : WORD ;
84      Pocet_NotCOMM_STA  AT  %QW432  : WORD ;
85
86      END_VAR
87
```

---

```
1
2      IF key THEN
3
4          ////**FAZE PRIKAZY**//
5          //Operace Vstupni kontrola
6          //fKontrolaVytahu.START:= KontrolaVytahu_CMD.0;
7          fKontrolaVytahu . RESUME := KontrolaVytahu_CMD . 1 ;
8          fKontrolaVytahu . HOLD := KontrolaVytahu_CMD . 2 ;
9          fKontrolaVytahu . ABORT := KontrolaVytahu_CMD . 3 ;
10
11          fLahev_1_Kontrola . START := Lahev_1_Kontrola_CMD . 0 ;
12          fLahev_1_Kontrola . RESUME := Lahev_1_Kontrola_CMD . 1 ;
13          fLahev_1_Kontrola . HOLD := Lahev_1_Kontrola_CMD . 2 ;
14          fLahev_1_Kontrola . ABORT := Lahev_1_Kontrola_CMD . 3 ;
15
16          fLahev_2_Kontrola . START := Lahev_2_Kontrola_CMD . 0 ;
17          fLahev_2_Kontrola . RESUME := Lahev_2_Kontrola_CMD . 1 ;
18          fLahev_2_Kontrola . HOLD := Lahev_2_Kontrola_CMD . 2 ;
19          fLahev_2_Kontrola . ABORT := Lahev_2_Kontrola_CMD . 3 ;
```

```
20
21 fLahev_3_Kontrola . START := Lahev_3_Kontrola_CMD . 0 ;
22 fLahev_3_Kontrola . RESUME := Lahev_3_Kontrola_CMD . 1 ;
23 fLahev_3_Kontrola . HOLD := Lahev_3_Kontrola_CMD . 2 ;
24 fLahev_3_Kontrola . ABORT := Lahev_3_Kontrola_CMD . 3 ;
25
26 fLahev_4_Kontrola . START := Lahev_4_Kontrola_CMD . 0 ;
27 fLahev_4_Kontrola . RESUME := Lahev_4_Kontrola_CMD . 1 ;
28 fLahev_4_Kontrola . HOLD := Lahev_4_Kontrola_CMD . 2 ;
29 fLahev_4_Kontrola . ABORT := Lahev_4_Kontrola_CMD . 3 ;
30
31 //Operace Sklenice
32 fPodavac_1_nahoru . START := Podavac_1_nahoru_CMD . 0 ;
33 fPodavac_1_nahoru . RESUME := Podavac_1_nahoru_CMD . 1 ;
34 fPodavac_1_nahoru . HOLD := Podavac_1_nahoru_CMD . 2 ;
35 fPodavac_1_nahoru . ABORT := Podavac_1_nahoru_CMD . 3 ;
36
37 fPodavac_1_dolu . START := Podavac_1_dolu_CMD . 0 ;
38 fPodavac_1_dolu . RESUME := Podavac_1_dolu_CMD . 1 ;
39 fPodavac_1_dolu . HOLD := Podavac_1_dolu_CMD . 2 ;
40 fPodavac_1_dolu . ABORT := Podavac_1_dolu_CMD . 3 ;
41
42 fSenzorZasobnik . START := Zasobnik_Senzor_CMD . 0 ;
43 fSenzorZasobnik . RESUME := Zasobnik_Senzor_CMD . 1 ;
44 fSenzorZasobnik . HOLD := Zasobnik_Senzor_CMD . 2 ;
45 fSenzorZasobnik . ABORT := Zasobnik_Senzor_CMD . 3 ;
46
47 fPodavac_2_nahoru . START := Podavac_2_nahoru_CMD . 0 ;
48 fPodavac_2_nahoru . RESUME := Podavac_2_nahoru_CMD . 1 ;
49 fPodavac_2_nahoru . HOLD := Podavac_2_nahoru_CMD . 2 ;
50 fPodavac_2_nahoru . ABORT := Podavac_2_nahoru_CMD . 3 ;
51
52 fPodavac_2_dolu . START := Podavac_2_dolu_CMD . 0 ;
53 fPodavac_2_dolu . RESUME := Podavac_2_dolu_CMD . 1 ;
54 fPodavac_2_dolu . HOLD := Podavac_2_dolu_CMD . 2 ;
55 fPodavac_2_dolu . ABORT := Podavac_2_dolu_CMD . 3 ;
56
57 fSenzorVytah . START := Vytah_Senzor_CMD . 0 ;
58 fSenzorVytah . RESUME := Vytah_Senzor_CMD . 1 ;
59 fSenzorVytah . HOLD := Vytah_Senzor_CMD . 2 ;
60 fSenzorVytah . ABORT := Vytah_Senzor_CMD . 3 ;
61
62 //Operace Lahve
63 fDopravnik . START := Dopravnik_CMD . 0 ;
64 fDopravnik . RESUME := Dopravnik_CMD . 1 ;
65 fDopravnik . HOLD := Dopravnik_CMD . 2 ;
66 fDopravnik . ABORT := Dopravnik_CMD . 3 ;
67
68 fVytah_nahoru . START := Vytah_nahoru_CMD . 0 ;
69 fVytah_nahoru . RESUME := Vytah_nahoru_CMD . 1 ;
70 fVytah_nahoru . HOLD := Vytah_nahoru_CMD . 2 ;
71 fVytah_nahoru . ABORT := Vytah_nahoru_CMD . 3 ;
72
73 fVytah_dolu . START := Vytah_dolu_CMD . 0 ;
74 fVytah_dolu . RESUME := Vytah_dolu_CMD . 1 ;
75 fVytah_dolu . HOLD := Vytah_dolu_CMD . 2 ;
```

```
76   fVytah_dolu . ABORT := Vytah_dolu_CMD . 3 ;
77
78   //Operace Dokonceni
79   fPritomnost_sklenice . START := Pritomnost_sklenice_CMD . 0 ;
80   fPritomnost_sklenice . RESUME := Pritomnost_sklenice_CMD . 1 ;
81   fPritomnost_sklenice . HOLD := Pritomnost_sklenice_CMD . 2 ;
82   fPritomnost_sklenice . ABORT := Pritomnost_sklenice_CMD . 3 ;
83
84   fTlacitko_STOP . START := Tlacitko_STOP_CMD . 0 ;
85   fTlacitko_STOP . RESUME := Tlacitko_STOP_CMD . 1 ;
86   fTlacitko_STOP . HOLD := Tlacitko_STOP_CMD . 2 ;
87   fTlacitko_STOP . ABORT := Tlacitko_STOP_CMD . 3 ;
88
89   fPocatecni_poloha . START := Pocatecni_poloha_CMD . 0 ;
90   fPocatecni_poloha . RESUME := Pocatecni_poloha_CMD . 1 ;
91   fPocatecni_poloha . HOLD := Pocatecni_poloha_CMD . 2 ;
92   fPocatecni_poloha . ABORT := Pocatecni_poloha_CMD . 3 ;
93
94   //prikaz napoj
95   napoj := WORD_TO_INT ( Napoj_CMD ) ;
96
97   END_IF ;
98
99   //komunikacni citac
100  Comm_CNT := WORD_TO_INT ( Comm ) ;
101
102
103  //***FAZE STAVY*****
104  //Operace Vstupni kontrola
105  KontrolaVytahu_STA . 0 := fKontrolaVytahu . RUNNING ;
106  KontrolaVytahu_STA . 1 := fKontrolaVytahu . HELD ;
107  KontrolaVytahu_STA . 2 := fKontrolaVytahu . ABORTED ;
108  KontrolaVytahu_STA . 3 := fKontrolaVytahu . COMPLETE ;
109
110  Lahev_1_Kontrola_STA . 0 := fLahev_1_Kontrola . RUNNING ;
111  Lahev_1_Kontrola_STA . 1 := fLahev_1_Kontrola . HELD ;
112  Lahev_1_Kontrola_STA . 2 := fLahev_1_Kontrola . ABORTED ;
113  Lahev_1_Kontrola_STA . 3 := fLahev_1_Kontrola . COMPLETE ;
114
115  Lahev_2_Kontrola_STA . 0 := fLahev_2_Kontrola . RUNNING ;
116  Lahev_2_Kontrola_STA . 1 := fLahev_2_Kontrola . HELD ;
117  Lahev_2_Kontrola_STA . 2 := fLahev_2_Kontrola . ABORTED ;
118  Lahev_2_Kontrola_STA . 3 := fLahev_2_Kontrola . COMPLETE ;
119
120  Lahev_3_Kontrola_STA . 0 := fLahev_3_Kontrola . RUNNING ;
121  Lahev_3_Kontrola_STA . 1 := fLahev_3_Kontrola . HELD ;
122  Lahev_3_Kontrola_STA . 2 := fLahev_3_Kontrola . ABORTED ;
123  Lahev_3_Kontrola_STA . 3 := fLahev_3_Kontrola . COMPLETE ;
124
125  Lahev_4_Kontrola_STA . 0 := fLahev_4_Kontrola . RUNNING ;
126  Lahev_4_Kontrola_STA . 1 := fLahev_4_Kontrola . HELD ;
127  Lahev_4_Kontrola_STA . 2 := fLahev_4_Kontrola . ABORTED ;
128  Lahev_4_Kontrola_STA . 3 := fLahev_4_Kontrola . COMPLETE ;
129
130  //Operace Sklenice
131  Podavac_1_nahoru_STA . 0 := fPodavac_1_nahoru . RUNNING ;
```

#### 1.1.1.2.4 POU: Mapovani

---

```
132 Podavac_1_nahoru_STA . 1 := fPodavac_1_nahoru . HELD ;
133 Podavac_1_nahoru_STA . 2 := fPodavac_1_nahoru . ABORTED ;
134 Podavac_1_nahoru_STA . 3 := fPodavac_1_nahoru . COMPLETE ;
135
136 Podavac_1_dolu_STA . 0 := fPodavac_1_dolu . RUNNING ;
137 Podavac_1_dolu_STA . 1 := fPodavac_1_dolu . HELD ;
138 Podavac_1_dolu_STA . 2 := fPodavac_1_dolu . ABORTED ;
139 Podavac_1_dolu_STA . 3 := fPodavac_1_dolu . COMPLETE ;
140
141 Zasobnik_Senzor_STA . 0 := fSenzorZasobnik . RUNNING ;
142 Zasobnik_Senzor_STA . 1 := fSenzorZasobnik . HELD ;
143 Zasobnik_Senzor_STA . 2 := fSenzorZasobnik . ABORTED ;
144 Zasobnik_Senzor_STA . 3 := fSenzorZasobnik . COMPLETE ;
145
146 Podavac_2_nahoru_STA . 0 := fPodavac_2_nahoru . RUNNING ;
147 Podavac_2_nahoru_STA . 1 := fPodavac_2_nahoru . HELD ;
148 Podavac_2_nahoru_STA . 2 := fPodavac_2_nahoru . ABORTED ;
149 Podavac_2_nahoru_STA . 3 := fPodavac_2_nahoru . COMPLETE ;
150
151 Podavac_2_dolu_STA . 0 := fPodavac_2_dolu . RUNNING ;
152 Podavac_2_dolu_STA . 1 := fPodavac_2_dolu . HELD ;
153 Podavac_2_dolu_STA . 2 := fPodavac_2_dolu . ABORTED ;
154 Podavac_2_dolu_STA . 3 := fPodavac_2_dolu . COMPLETE ;
155
156 Vytah_Senzor_STA . 0 := fSenzorVytah . RUNNING ;
157 Vytah_Senzor_STA . 1 := fSenzorVytah . HELD ;
158 Vytah_Senzor_STA . 2 := fSenzorVytah . ABORTED ;
159 Vytah_Senzor_STA . 3 := fSenzorVytah . COMPLETE ;
160
161 //Operace Lahve
162 Dopravnik_STA . 0 := fDopravnik . RUNNING ;
163 Dopravnik_STA . 1 := fDopravnik . HELD ;
164 Dopravnik_STA . 2 := fDopravnik . ABORTED ;
165 Dopravnik_STA . 3 := fDopravnik . COMPLETE ;
166
167 Vytah_nahoru_STA . 0 := fVytah_nahoru . RUNNING ;
168 Vytah_nahoru_STA . 1 := fVytah_nahoru . HELD ;
169 Vytah_nahoru_STA . 2 := fVytah_nahoru . ABORTED ;
170 Vytah_nahoru_STA . 3 := fVytah_nahoru . COMPLETE ;
171
172 Vytah_dolu_STA . 0 := fVytah_dolu . RUNNING ;
173 Vytah_dolu_STA . 1 := fVytah_dolu . HELD ;
174 Vytah_dolu_STA . 2 := fVytah_dolu . ABORTED ;
175 Vytah_dolu_STA . 3 := fVytah_dolu . COMPLETE ;
176
177 //Operace Dokonceni
178 Pritomnost_sklenice_STA . 0 := fPritomnost_sklenice . RUNNING ;
179 Pritomnost_sklenice_STA . 1 := fPritomnost_sklenice . HELD ;
180 Pritomnost_sklenice_STA . 2 := fPritomnost_sklenice . ABORTED ;
181 Pritomnost_sklenice_STA . 3 := fPritomnost_sklenice . COMPLETE ;
182
183 Tlacidko_STOP_STA . 0 := fTlacidko_STOP . RUNNING ;
184 Tlacidko_STOP_STA . 1 := fTlacidko_STOP . HELD ;
185 Tlacidko_STOP_STA . 2 := fTlacidko_STOP . ABORTED ;
186 Tlacidko_STOP_STA . 3 := fTlacidko_STOP . COMPLETE ;
187
```

```
188 Pocatecni_poloha_STA . 0 := fPocatecni_poloha . RUNNING ;
189 Pocatecni_poloha_STA . 1 := fPocatecni_poloha . HELD ;
190 Pocatecni_poloha_STA . 2 := fPocatecni_poloha . ABORTED ;
191 Pocatecni_poloha_STA . 3 := fPocatecni_poloha . COMPLETE ;
192
193 //Porucha davkovani
194 Porucha_Sta . 0 := Batch_Porucha . Komunikace ;
195 Porucha_Sta . 1 := Batch_Porucha . Nizky_Tlak ;
196 Porucha_Sta . 2 := Batch_Porucha . TimeOut ;
197 Porucha_Sta . 3 := Batch_Porucha . Podavac_1 ;
198 Porucha_Sta . 4 := Batch_Porucha . Podavac_2 ;
199 Porucha_Sta . 5 := Batch_Porucha . Zasobnik ;
200 Porucha_Sta . 6 := Batch_Porucha . Vytah_Sklenice ;
201 Porucha_Sta . 7 := Batch_Porucha . Lahev_1 ;
202 Porucha_Sta . 8 := Batch_Porucha . Lahev_2 ;
203 Porucha_Sta . 9 := Batch_Porucha . Lahev_3 ;
204 Porucha_Sta . 10 := Batch_Porucha . Lahev_4 ;
205
206 //Celkove bilance
207 Chod_celkem_STA := DINT_TO_WORD ( Chod_celkem ) ;
208 Porucha_Celkem_STA := DINT_TO_WORD ( Porucha_Celkem ) ;
209 Oprava_Celkem_STA := DINT_TO_WORD ( Oprava_Celkem ) ;
210 Poruchy_Pocet_STA := Poruchy_Pocet ;
211 Opravy_Pocet_STA := Opravy_Pocet ;
212
213 //MTBF A MTTR
214 MTBF_STA := DINT_TO_WORD ( MTBF ) ;
215 MTTR_STA := DINT_TO_WORD ( MTTR ) ;
216
217 //Dobre davky
218 Davky_Dobre_STA := Davky_DSC . Davky_Dobre ;
219 //Celkove davky
220 Davky_Celkem_STA := Davky_DSC . Davky_Celkem ;
221 //Jednotlive lahve
222 Davky_Lahve_1_STA := Davky_DSC . Lahev_1 ;
223 Davky_Lahve_2_STA := Davky_DSC . Lahev_2 ;
224 Davky_Lahve_3_STA := Davky_DSC . Lahev_3 ;
225 Davky_Lahve_4_STA := Davky_DSC . Lahev_4 ;
226 //Celkove lahve
227 Davky_Lahve_Celkem_STA := Davky_DSC . Lahev_Celkem ;
228 //Orientacni obsah lahve
229 Lahev_1_ml_STA := Davky_DSC . Zbyva_Lahve_1 ;
230 Lahev_2_ml_STA := Davky_DSC . Zbyva_Lahve_2 ;
231 Lahev_3_ml_STA := Davky_DSC . Zbyva_Lahve_3 ;
232 Lahev_4_ml_STA := Davky_DSC . Zbyva_Lahve_4 ;
233
234 //Kvalita vyroby
235 REAL_VALUE := Davky_DSC . Kvalita ;
236 Kvalita_STA_1 := WORD_VALUE [ 1 ] ;
237 Kvalita_STA_2 := WORD_VALUE [ 0 ] ;
238
239 //kvalita spojeni
240 REAL_VALUE := Kvalita_Comm ;
241 Kvalita_Spojeni_STA_1 := WORD_VALUE [ 1 ] ;
242 Kvalita_Spojeni_STA_2 := WORD_VALUE [ 0 ] ;
243
```

```
244     Pocet_NotCOMM_STA := Pocet_NotComm ;
245
246
247
248
249
250
251
252
```

#### 1.1.1.2.5 POU: Stop

```
1     FUNCTION_BLOCK Stop
2     VAR_INPUT
3     END_VAR
4     VAR_OUTPUT
5     END_VAR
6     VAR
7     END_VAR
8
```

```
1
2     VYTAH_UP := FALSE ;
3     VYTAH_Down := FALSE ;
4     POD_1_UP := FALSE ;
5     POD_1_Down := FALSE ;
6     POD_2_UP := FALSE ;
7     POD_2_Down := FALSE ;
8     DOPRAVNIK_SMER1 := FALSE ;
9     DOPRAVNIK_SMER2 := FALSE ;
10
11
```

#### 1.1.1.3 Folder: Podnikove Rizeni

##### 1.1.1.3.1 POU: Celkem\_Davek

```
1     FUNCTION_BLOCK Celkem_Davek
2     VAR_INPUT
3         Porucha : BOOL ;
4         Reset : BOOL ;
5     END_VAR
6     VAR_OUTPUT
7         Out : BOOL ;
8         Davky_Celkem : UINT ;
9         Davky_Dobre : UINT ;
10        Kvalita : REAL ;
11        Lahev_1 : UINT ;
12        Lahev_2 : UINT ;
13        Lahev_3 : UINT ;
14        Lahev_4 : UINT ;
15        Lahev_Celkem : UINT ;
```

```
16  END_VAR
17  VAR
18      fTRIG_Davka : R_TRIG ;
19      fTRIG_Ukonceni : R_TRIG ;
20      fTRIG_Porucha : R_TRIG ;
21      fLahev_1 : UINT ;
22      fLahev_2 : UINT ;
23      fLahev_3 : UINT ;
24      fLahev_4 : UINT ;
25      fLahev_Celkem : UINT ;
26      fReset_Trig : R_TRIG ;
27      fDavky_Spatne : UINT ;
28      fDavky_Dobre : UINT ;
29      fKvalita : REAL ;
30  END_VAR
31
```

---

```
1
2  fTRIG_Davka ( Clk := fVytah_nahoru . COMPLETE AND ( napoj = 1 OR napoj = 2
OR napoj = 3 OR napoj = 4 ) ) ;
3  fTRIG_Ukonceni ( Clk := ( fDopravnik . COMPLETE ) AND ( Napoj = 0 ) ) ;
4  fTRIG_Porucha ( Clk := Porucha ) ;
5  fReset_Trig ( clk := reset ) ;
6
7
8  IF fReset_Trig . q THEN
9      fLahev_1 := fLahev_2 := fLahev_3 := fLahev_4 := 0 ;
10 END_IF
11
12 IF fTRIG_Ukonceni . Q THEN
13     fDavky_Dobre := fDavky_Dobre + 1 ;
14 END_IF
15
16 IF fTRIG_Porucha . Q THEN
17     fDavky_Spatne := fDavky_Spatne + 1 ;
18 END_IF
19
20
21 IF fTRIG_Davka . Q THEN
22     IF napoj = 1 THEN
23         fLahev_1 := fLahev_1 + 1 ;
24     ELSIF napoj = 2 THEN
25         fLahev_2 := fLahev_2 + 1 ;
26     ELSIF napoj = 3 THEN
27         fLahev_3 := fLahev_3 + 1 ;
28     ELSIF napoj = 4 THEN
29         fLahev_4 := fLahev_4 + 1 ;
30     END_IF ;
31 END_IF
32
33 IF ( fDavky_Dobre + fDavky_Spatne ) > 0 THEN
34     fKvalita := ( fDavky_Dobre / ( fDavky_Dobre + fDavky_Spatne ) ) * 100 ;
35 END_IF ;
36
```

```

37     fLahev_celkem := fLahev_1 + fLahev_2 + fLahev_3 + fLahev_4 ;
38
39     //vystupy
40     Lahev_1 := fLahev_1 ;
41     Lahev_2 := fLahev_2 ;
42     Lahev_3 := fLahev_3 ;
43     Lahev_4 := fLahev_4 ;
44     Lahev_Celkem := fLahev_Celkem ;
45     Davky_Dobre := fDavky_Dobre ;
46     Davky_Celkem := fDavky_Dobre + fDavky_Spatne ;
47     Kvalita := fKvalita ;
48

```

### 1.1.1.3.2 POU: Celkova\_Doba

```

1     FUNCTION_BLOCK Celkova_Doba
2     VAR_INPUT
3         Enable : BOOL ;
4         Pulse_Sec : BOOL ;
5         Reset : BOOL ;
6     END_VAR
7     VAR_IN_OUT
8         Init : DINT ;
9     END_VAR
10    VAR_OUTPUT
11        Overall : DINT ;
12    END_VAR
13    VAR
14        fOverall : DINT ;
15    END_VAR
16

```

```

1
2     IF Reset THEN
3         fOverall := 0 ;
4     END_IF
5
6     IF Enable AND Pulse_Sec THEN
7         fOverall := fOverall + 1 ;
8     END_IF
9
10    IF Init <> 0 THEN
11        fOverall := Init ;
12        Init := 0 ;
13    END_IF
14
15
16    Overall := fOverall ;
17
18
19
20

```



### 1.1.1.3.3 POU: MeanTime

```
1  FUNCTION_BLOCK MeanTime
2  VAR_INPUT
3      Celkova_Doba : DINT ;
4      Pocet : UINT ;
5  END_VAR
6  VAR_OUTPUT
7      MT : DINT ;
8  END_VAR
9  VAR
10     fMT : Real ;
11 END_VAR
12
```

---

```
1  IF pocet > 0 THEN
2      fMT := DINT_TO_REAL ( Celkova_Doba ) / Pocet ;
3  ELSE
4      fMT := 0 ;
5  END_IF
6
7  MT := Real_TO_DINT ( fMT ) ;
8
```

### 1.1.1.3.4 POU: Podnikove\_Rizeni

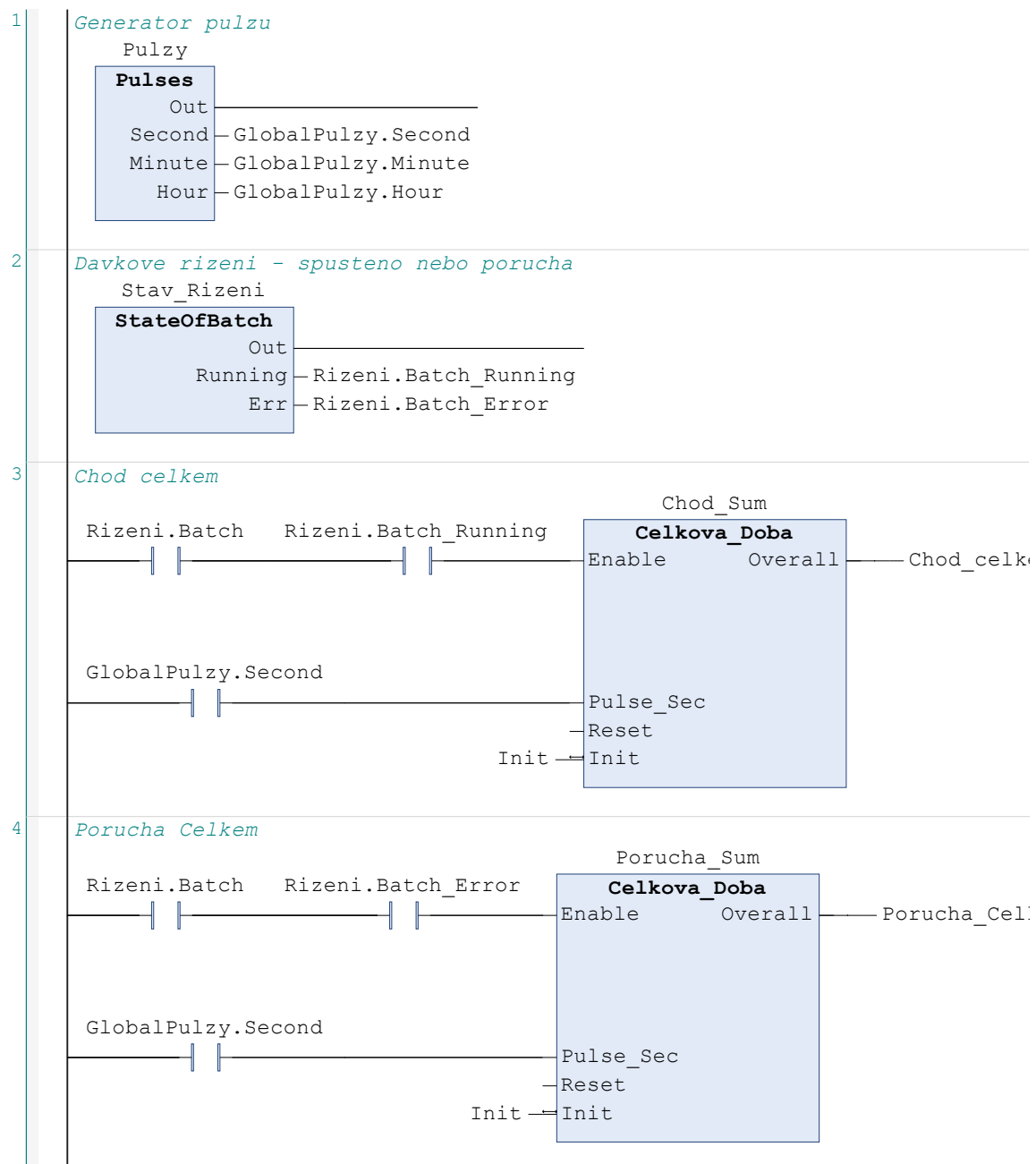
```
1  FUNCTION_BLOCK Podnikove_Rizeni
2  VAR_INPUT
3  END_VAR
4  VAR_OUTPUT
5  END_VAR
6  VAR
7      //Generator pulzu (sekundy,minuty,hodiny)
8      Pulzy : Pulses ;
9
10     //pocitani bilanci
11     Chod_Sum : Celkova_Doba ;
12     Porucha_Sum : Celkova_Doba ;
13     Oprava_Sum : Celkova_Doba ;
14     Davky : Celkem_Davek ;
15
16     //Stav rizeni
17     Stav_Rizeni : StateOfBatch ;
18     //MTBF
19     MeanTimeErr : MeanTime ;
20     //MTTF
21     MeanTimeRepair : MeanTime ;
22     //Kvalita Spojeni
23     Kvalita_Spojeni : Spojeni ;
24     //Prumerna doba na vyrobu davky
25     Prumerna_DobaVyroby : Prumer_DD ;
26     //Orientacni stav tekutin
27     Zbyva_Tekutiny : Zbyva_Davek ;
```

### 1.1.1.3.4 POU: Podnikove\_Rizeni

```

28
29      //Lokalni promenne
30      Init : DINT ;
31      Reset : bool ;
32      END_VAR
33

```



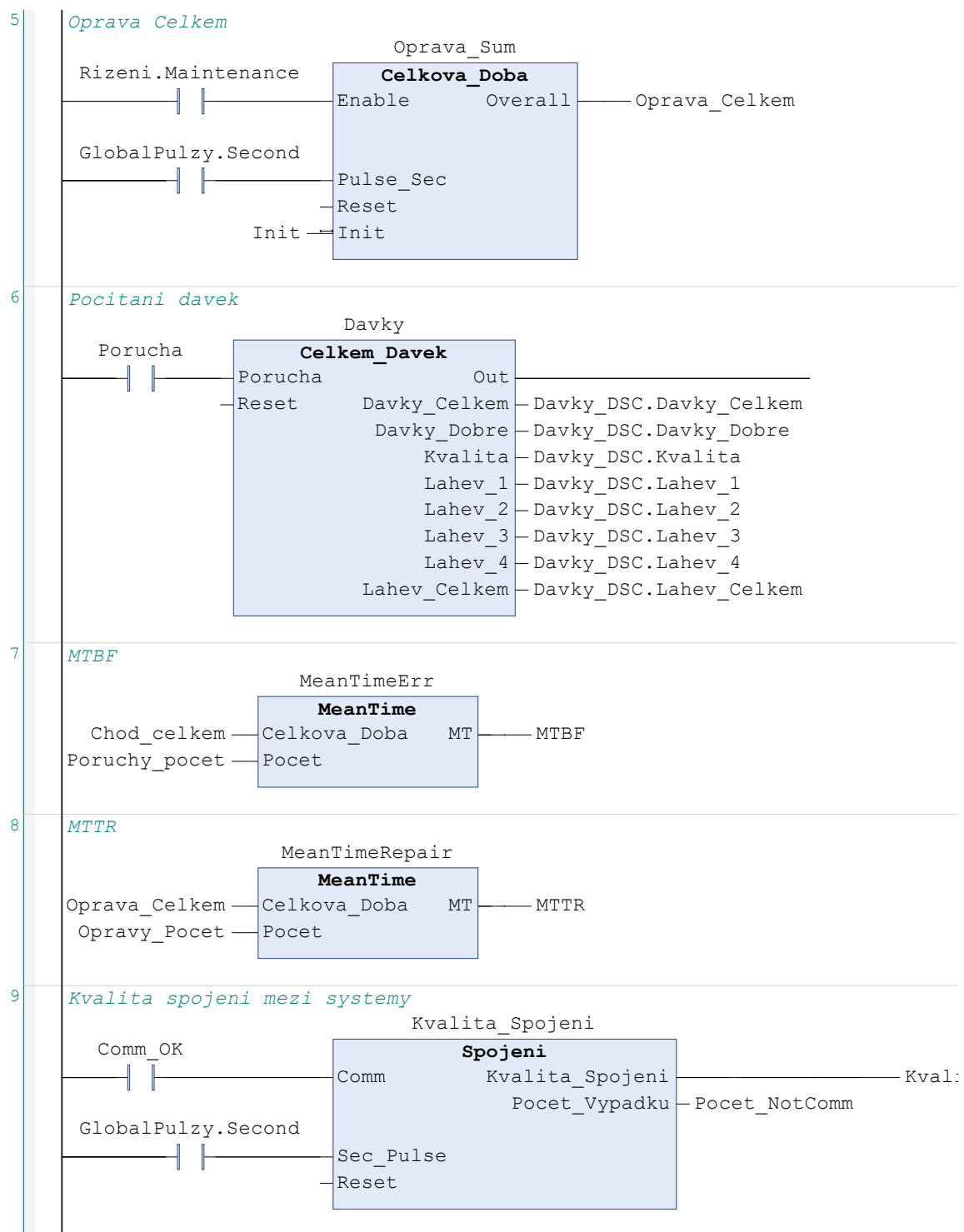
em

---

kem

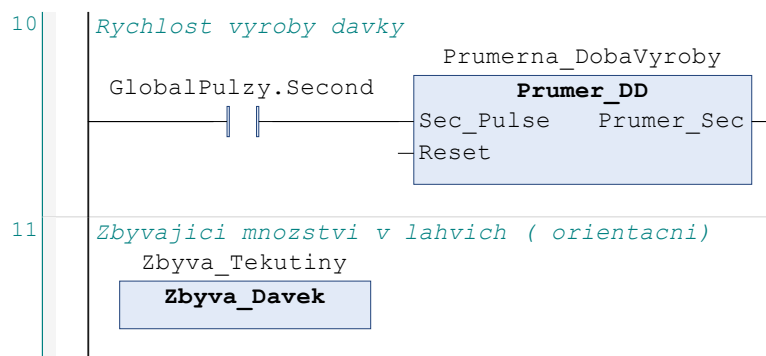
---

### 1.1.1.3.4 POU: Podnikove\_Rizeni



ita\_Comm

---





### 1.1.1.3.5 POU: Prumer\_DD

```
1      //prumerna dobra vyroby davky
2      FUNCTION_BLOCK Prumer_DD
3      VAR_INPUT
4          Sec_Pulse : BOOL ;
5          Reset : BOOL ;
6      END_VAR
7      VAR_OUTPUT
8          Prumer_Sec : UINT ;
9      END_VAR
10     VAR
11         fTRIG_Ukonceni : R_TRIG ;
12         Davky : ARRAY [ 1 .. 10 ] OF UINT ;
13     END_VAR
14
```

---

```
1
2      fTRIG_Ukonceni ( Clk := ( fDopravnik . COMPLETE ) AND ( Napoj = 0 ) ) ;
3
4
5
```

### 1.1.1.3.6 POU: Pulses

```
1      FUNCTION_BLOCK Pulses
2      VAR_INPUT
3      END_VAR
4      VAR_OUTPUT
5          Out : bool ;
6          Second : BOOL ;
7          Minute : BOOL ;
8          Hour : BOOL ;
9      END_VAR
10     VAR
11         DateTime : DT ;
12         RtcClock : RTC ;
13         Unix : DINT ;
14         MinutePulses : DINT ;
15         HourPulses : DINT ;
16
17         LastValue : DINT ;
18         fPulseS : BOOL ;
19         fPulseM : BOOL ;
20         fPulseH : BOOL ;
21
22
23     END_VAR
24
```

---

```
1      fPulseS := FALSE ;
2      fPulseM := FALSE ;
3      fPulseH := false ;
```



```
4
5   RtcClock ( EN := TRUE ) ;
6   DateTime := RtcClock . CDT ;
7
8   Unix := DT_TO_DINT ( DateTime ) ;
9
10  IF Unix <> LastValue THEN
11    LastValue := Unix ;
12    fPulseS := TRUE ;
13  END_IF
14
15
16  IF fpulseS THEN
17    MinutePulses := MinutePulses + 1 ;
18    HourPulses := HourPulses + 1 ;
19  END_IF
20
21  IF MinutePulses >= 60 THEN
22    MinutePulses := 0 ;
23    fPulseM := TRUE ;
24  END_IF
25
26  IF HourPulses >= 3600 THEN
27    HourPulses := 0 ;
28    fPulseH := TRUE ;
29  END_IF
30
31
32  //vystup
33  Second := fpulseS ;
34  Minute := fPulseM ;
35  Hour := fPulseH ;
36
37
```

### 1.1.1.3.7 POU: Spojeni

```
1  FUNCTION_BLOCK Spojeni
2  VAR_INPUT
3    Comm : BOOL ;
4    Sec_Pulse : BOOL ;
5    Reset : BOOL ;
6  END_VAR
7  VAR_OUTPUT
8    Kvalita_Spojeni : REAL ;
9    Pocet_Vypadku : UINT ;
10 END_VAR
11 VAR
12   fTRIG_Pocet : F_TRIG ;
13   fTRIG_Reset : R_TRIG ;
14   fSpatne : DINT ;
15   fDobre : DINT ;
16   fKvalita : REAL ;
17   fPocet : UINT ;
18 END_VAR
19
```

```
1
2   fTRIG_Pocet ( clk := Comm ) ;
3   fTRIG_Reset ( clk := Reset ) ;
4
5   IF fTRIG_Reset . Q THEN
6       fDobre := 0 ;
7       fSpatne := 0 ;
8   END_IF
9
10  IF Comm AND Sec_Pulse THEN
11      fDobre := fDobre + 1 ;
12  ELSIF NOT Comm AND Sec_Pulse THEN
13      fSpatne := fSpatne + 1 ;
14  END_IF
15
16  IF fTRIG_Pocet . Q THEN
17      fPocet := fPocet + 1 ;
18  END_IF
19
20  IF ( fDobre + fSpatne ) > 0 THEN
21      fKvalita := ( DINT_TO_REAL ( fDobre ) / DINT_TO_REAL ( fDobre + fSpatne ) ) *
22      100 ;
23  END_IF
24
25
26  //vystup
27  Pocet_Vypadku := fPocet ;
28  Kvalita_Spojeni := fKvalita ;
29
```

### 1.1.1.3.8 POU: Zbyva\_Davek

```
1   FUNCTION_BLOCK Zbyva_Davek
2   VAR_INPUT
3   END_VAR
4   VAR_OUTPUT
5   END_VAR
6   VAR
7   END_VAR
8
```

```
1   //Lahev 1
2   IF NOT OPT_1 AND NOT BOTTLE_1 THEN
3       Davky_DSC . Zbyva_Lahev_1 := 700 ;
4   END_IF
5   IF Napoj = 1 AND fVytah_nahoru . COMPLETE THEN
6       IF Davky_DSC . Zbyva_Lahev_1 > 0 THEN
7           Davky_DSC . Zbyva_Lahev_1 := Davky_DSC . Zbyva_Lahev_1 - 20 ;
8       ELSE
9           Davky_DSC . Zbyva_Lahev_1 := 0 ;
10      END_IF ;
11  END_IF
12
```

```
13      //Lahev 2
14      IF NOT OPT_2 AND NOT BOTTLE_2 THEN
15          Davky_DSC . Zbyva_Lahev_2 := 700 ;
16      END_IF
17      IF Napoj = 2 AND fVytah_nahoru . COMPLETE THEN
18          IF Davky_DSC . Zbyva_Lahev_2 > 0 THEN
19              Davky_DSC . Zbyva_Lahev_2 := Davky_DSC . Zbyva_Lahev_2 - 20 ;
20          ELSE
21              Davky_DSC . Zbyva_Lahev_2 := 0 ;
22          END_IF ;
23      END_IF
24
25
26      //Lahev 2
27      IF NOT OPT_3 AND NOT BOTTLE_3 THEN
28          Davky_DSC . Zbyva_Lahev_3 := 700 ;
29      END_IF
30      IF Napoj = 3 AND fVytah_nahoru . COMPLETE THEN
31          IF Davky_DSC . Zbyva_Lahev_3 > 0 THEN
32              Davky_DSC . Zbyva_Lahev_3 := Davky_DSC . Zbyva_Lahev_3 - 20 ;
33          ELSE
34              Davky_DSC . Zbyva_Lahev_3 := 0 ;
35          END_IF ;
36      END_IF
37
38      //Lahev 4
39      IF NOT OPT_4 AND NOT BOTTLE_4 THEN
40          Davky_DSC . Zbyva_Lahev_4 := 700 ;
41      END_IF
42      IF Napoj = 4 AND fVytah_nahoru . COMPLETE THEN
43          IF Davky_DSC . Zbyva_Lahev_4 > 0 THEN
44              Davky_DSC . Zbyva_Lahev_4 := Davky_DSC . Zbyva_Lahev_4 - 20 ;
45          ELSE
46              Davky_DSC . Zbyva_Lahev_4 := 0 ;
47          END_IF ;
48      END_IF
49
50
```

## 1.1.1.4 Folder: Struktury

### 1.1.1.4.1 DUT: Davky\_Pocet

```
1      TYPE Davky_Pocet :
2      STRUCT
3          Davky_Celkem : UINT ;
4          Davky_Dobre : UINT ;
5
6          Kvalita : REAL ;
7
8          Lahev_1 : UINT ;
9          Lahev_2 : UINT ;
10         Lahev_3 : UINT ;
11         Lahev_4 : UINT ;
```

```
12         Lahev_Celkem : UINT ;
13
14         Zbyva_Lahev_1  : UINT ;
15         Zbyva_Lahev_2  : UINT ;
16         Zbyva_Lahev_3  : UINT ;
17         Zbyva_Lahev_4  : UINT ;
18
19     END_STRUCT
20     END_TYPE
21
```

#### 1.1.1.4.2 DUT: Faze

```
1     TYPE Faze :
2     STRUCT
3         //stavy
4         COMPLETE : BOOL ;
5         RUNNING  : BOOL ;
6         HELD     : BOOL ;
7         ABORTED  : BOOL ;
8         //Povely
9         START    : BOOL ;
10        RESUME    : BOOL ;
11        HOLD     : BOOL ;
12        ABORT    : BOOL ;
13
14    END_STRUCT
15    END_TYPE
16
```

#### 1.1.1.4.3 DUT: GLocal\_Pulses

```
1     TYPE GLocal_Pulses :
2     STRUCT
3         Second : BOOL ;
4         Minute : BOOL ;
5         Hour   : bool ;
6     END_STRUCT
7     END_TYPE
8
```

#### 1.1.1.4.4 DUT: Poruchy

```
1     TYPE Poruchy :
2     STRUCT
3         Komunikace : BOOL ;           //100
4         Nizky_Tlak : BOOL ;           //101
5         TimeOut    : BOOL ;           //102
6         Podavac_1  : BOOL ;           //200
7         Podavac_2  : BOOL ;           //201
8         Zasobnik   : BOOL ;           //202
9         Vytah_Sklenice : BOOL ;       //203
10        Lahev_1    : BOOL ;           //300
11        Lahev_2    : BOOL ;           //301
12        Lahev_3    : BOOL ;           //302
13        Lahev_4    : BOOL ;           //303
```

```
14
15     //Pocet chyb na jednotku
16     Davkovac_Pocet : UINT ;
17     Podavac_Pocet : UINT ;
18     Ostatni_Pocet : UINT ;
19
20     END_STRUCT
21     END_TYPE
22
```

### 1.1.1.4.5 DUT: TypeofCTRL

```
1     TYPE TypeofCTRL :
2     STRUCT
3         Batch : BOOL ;
4         Man : BOOL ;
5         Maintenance : BOOL ;
6
7         //Batch
8         Batch_Running : BOOL ;
9         Batch_Error : bool ;
10
11
12     END_STRUCT
13     END_TYPE
14
```

### 1.1.1.5 Global Variable List: ASi

```
1     VAR_GLOBAL
2
3         TLAC_1 AT %IX10.0 : BOOL ;           //Tlačítko 1
4         TLAC_1_LIGHT AT %QX10.2 : BOOL ;     //Tlacitko 1 podsviceni
5         TLAC_1_LIGHT_OK AT %IX10.2 : BOOL ;
6         TLAC_2 AT %IX10.1 : BOOL ;           //Tlačítko 2
7         TLAC_2_LIGHT AT %QX10.3 : BOOL ;     //Tlacitko 2 podsviceni
8         TLAC_2_LIGHT_OK AT %IX10.3 : BOOL ;
9         TLAC_3 AT %IX9.0 : BOOL ;             //Tlačítko 3
10        TLAC_3_LIGHT AT %QX9.2 : BOOL ;       //Tlacitko 3 podsviceni
11        TLAC_3_LIGHT_OK AT %IX9.2 : BOOL ;
12        TLAC_4 AT %IX9.1 : BOOL ;             //Tlačítko 4
13        TLAC_4_LIGHT AT %QX9.3 : BOOL ;       //Tlacitko 4 podsviceni
14        TLAC_4_LIGHT_OK AT %IX9.3 : BOOL ;
15        TLAC_STOP AT %IX13.3 : BOOL ;         //Stop tlačítko
16        TLAC_STOP_LIGHT AT %QX13.1 : BOOL ;   //Stop tlacitko
17        podsviceni
18        TLAK_OK AT %IX3.0 : BOOL ;             //Tlakový senzor
19        KEY AT %IX13.2 : BOOL ;               //poloha klíče
20
21        DOPR_START AT %IX5.0 : BOOL ;         //magnetický senzor
22        nazačátku dopravníku
23        DOPR_1 AT %IX1.0 : BOOL ;             //magnetický senzor
24        dopravníku u 1. láhve
25        DOPR_2 AT %IX1.1 : BOOL ;             //magnetický senzor
26        dopravníku u 2. láhve
27        DOPR_3 AT %IX1.2 : BOOL ;             //magnetický senzor
28
```

### 1.1.1.5 Global Variable List: ASi

```
24      dopravniku u 3. láhve                                     //magnetický senzor
      DOPR_4 AT %IX1.3 : BOOL ;
      dopravniku u 4. láhve
25      DOPR_END AT %IX5.1 : BOOL ;                               //magnetický senzor nakonci
      dopravniku
26
27      OPT_PODAVAC AT %IX11.1 : BOOL ;                             //cidlo mezi podavaci
      (pritomnost sklenice)
28      OPT_VYTAH AT %IX11.0 : BOOL ;                               //cidlo na vytahu
      (pritomnost sklenice)
29      OPT_1 AT %IX11.3 : BOOL ;                                   //lahev 1 ve spravne
      poloze
30      OPT_2 AT %IX12.1 : BOOL ;                                   //lahev 2 ve spravne
      poloze
31      OPT_3 AT %IX12.2 : BOOL ;                                   //lahev 3 ve spravne
      poloze
32      OPT_4 AT %IX12.3 : BOOL ;                                   //lahev 4 ve spravne
      poloze
33      BOTTLE_1 AT %IX8.0 : BOOL ;                                 //přítomnost kapaliny v 1.
      láhvy
34      BOTTLE_2 AT %IX8.1 : BOOL ;                                 //přítomnost kapaliny v 2.
      láhvy
35      BOTTLE_3 AT %IX8.2 : BOOL ;                                 //přítomnost kapaliny v 3.
      láhvy
36      BOTTLE_4 AT %IX8.3 : BOOL ;                                 //přítomnost kapaliny v 4.
      láhvy
37      POD_1_DOWN AT %QX2.2 : BOOL ;                               //podavac 1 dolu
38      POD_1_DOWN_OK AT %IX2.0 : BOOL ;                           //spodni cidlo padavace
      1
39      POD_1_DOWN_MOVE AT %IX2.2 : BOOL ;                           //pohyb podavace 1 dolu
40      POD_1_UP AT %QX2.3 : BOOL ;                                 //podavac 1 nahoru
41      POD_1_UP_OK AT %IX2.1 : BOOL ;                               //horni cidlo padavace 1
      1
42      POD_1_UP_MOVE AT %IX2.3 : BOOL ;                             //pohyb podavace 1
      nahoru
43      POD_2_DOWN AT %QX4.2 : BOOL ;                               //podavac 2 dolu
44      POD_2_DOWN_OK AT %IX4.0 : BOOL ;                           //spodni cidlo padavace
      2
45      POD_2_DOWN_MOVE AT %IX4.2 : BOOL ;                           //pohyb podavace 2 dolu
46      POD_2_UP AT %QX4.3 : BOOL ;                                 //podavac 2 nahoru
47      POD_2_UP_OK AT %IX4.1 : BOOL ;                               //horni cidlo padavace 2
48      POD_2_UP_MOVE AT %IX4.3 : BOOL ;                             //pohyb podavace 2
      nahoru
49      VYTAH_UP AT %QX5.3 : BOOL ;                                 //vytah pohyb nahoru
50      VYTAH_DOWN AT %QX5.2 : BOOL ;                               //vytah pohyb dolu
51      VYTAH_UP_OK AT %IX6.1 : BOOL ;                             //horni cidlo vytahu
52      VYTAH_DOWN_OK AT %IX6.3 : BOOL ;                           //spodni cidlo vytahu
53      VYTAH_UP_MOVE AT %IX5.3 : BOOL ;                             //pohyb vytahu nahoru
      1
54      VYTAH_DOWN_MOVE AT %IX5.2 : BOOL ;                           //pohyb vytahu dolu
55      DOPRAVNIK_SMER1 AT %QX3.2 : BOOL ;                           //dopravník pohyb vlevo
56      DOPRAVNIK_SMER2 AT %QX3.3 : BOOL ;                           //dopravník pohyb vpravo
57      DOPRAVNIK_SMER1_MOVE AT %IX3.2 : BOOL ;                     //dopravník se pohybuje
      smerem vlevo
58      DOPRAVNIK_SMER2_MOVE AT %IX3.3 : BOOL ;                     //dopravník se pohybuje
      smerem vpravo
59      RED AT %QX7.0 : BOOL ;                                       //majak cervena
```

```
60     RED_OK  AT  %IX7.0 :  BOOL ;
61     ORANGE AT  %QX7.1 :  BOOL ;           //majak oranzova
62     ORANGE_OK AT  %IX7.1 :  BOOL ;
63     GREEN  AT  %QX7.2 :  BOOL ;           //majak zelena
64     GREEN_OK AT  %IX7.2 :  BOOL ;
65     END_VAR
66
```

### 1.1.1.6 Global Variable List: GLOBAL

```
1     VAR_GLOBAL
2
3     //typ rizeni
4     Rizeni : TypeOfCtrl ;
5
6
7     Porucha_En : BOOL ;
8
9     //****BATCH RIZENI****
10    //Struktury fazi
11    //***Operace Vstupni kontrola
12    fKontrolaVytahu : faze ;
13    fLahev_1_Kontrola : faze ;
14    fLahev_2_Kontrola : faze ;
15    fLahev_3_Kontrola : faze ;
16    fLahev_4_Kontrola : faze ;
17
18    //***Operace Sklenice
19    fSenzorZasobnik : Faze ;
20    fSenzorVytah : Faze ;
21    fPodavac_1_nahoru : faze ;
22    fPodavac_1_dolu : faze ;
23    fPodavac_2_nahoru : faze ;
24    fPodavac_2_dolu : faze ;
25
26    //***Operace Lahve
27    fDopravnik : faze ;
28    fVytah_nahoru : faze ;
29    fVytah_dolu : faze ;
30
31    //***Dokonceni
32    fPritomnost_sklenice : faze ;
33    fTlacitko_STOP : faze ;
34    fPocatecni_poloha : faze ;
35
36    //Kontrola komunikace
37    Comm_CNT : INT ;           //komunikacni citac
38    Comm_OK : BOOL ;          //komunikace OK
39
40    //Porucha batch
41    Batch_Porucha : Poruchy ;
42
43    napoj : INT ;              //vyber napoje 0 - pocatek,
                                // (1,2,3,4) - lahve, 5 - konec
44    DoprAktu_pozice : INT ;    //aktualni pozice dopravniku
```

```
45
46
47     ManDrink : ARRAY [ 1 .. 8 ] OF INT ;           //Manualni vyber lahvi
48     //Porucha manual
49     Porucha : BOOL ;
50
51     Simulace : BOOL := FALSE ;
52
53     //Kod chyby pro blikani majaku
54     Majak_kod : UINT ;
55
56     //Pulzy
57     GlobalPulzy : Global_Pulses ;
58
59     //*****
60     //Podnikove rizeni
61     Chod_celkem : DINT ;
62     Porucha_Celkem : DINT ;
63     Oprava_Celkem : DINT ;
64     Poruchy_Pocet : UINT ;
65     Opravy_Pocet : UINT ;
66
67     MTBF : DINT ;
68     MTTR : DINT ;
69
70     Kvalita_Comm : REAL ;
71     Pocet_NotComm : UINT ;
72     //Dobre, spatne a celkem davky
73     Davky_DSC : Davky_Pocet ;
74
75
76
77
78
79     END_VAR
80
```

### 1.1.1.7 Library Manager: Library Manager

```
#Standard
IoStandard, 3.5.4.0 (System)
    Base Interfaces, * (System)
    CmpErrors Interfaces, * (System)
    SysTypes2 Interfaces, * (System)
ACnnnnn_Utills, * (ifm electronic)
    #ACnnnnn_SYS_CMD
#System_VisuElems
#System_VisuElemMeter
#System_VisuElemsSpecialControls
#System_VisuElemsWinControls
#System_VisuElemTextEditor
#System_VisuElemTrace
#System_VisuNativeControl
#System_VisuElemsAlarm
#System_VisuElemCamDisplayer
#System_VisuElem3DPath
```



#system\_visuinputs  
#Standard64

### 1.1.1.8 Task Configuration: Task Configuration

Max. number of tasks: 100  
Max. number of cyclic tasks: 100  
Max. number of freewheeling tasks: 100  
Max. number of event tasks: 100  
Max. number of status tasks: 100

System Events:

#### 1.1.1.8.1 Task: MainTask

Priority: 1  
Type: Cyclic  
Interval: t#20ms  
Watchdog: Inactive  
POUs: Mapovani  
Main

##### 1.1.1.8.1.1 Program call: Mapovani

##### 1.1.1.8.1.2 Program call: Main

### 1.1.1.9 Trace: Trace

Settings:

Record 'Trace':  
Trigger variable:  
Trigger edge: Positive  
Post trigger (samples): 2  
Trigger level:  
Task: MainTask  
Measure in every: 1-th cycle  
Record condition:  
Buffer size: 101  
Comment:  
POU for visualisation: False  
Variables: Main.Manual\_program.Vytah\_ovladani.Complete

## ***1.2 Device: ASi\_Master\_1***

### **Information**

Name: ASi Master 1  
Vendor: ifm electronic  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: ASi Master 1 from ifm

### **1.2.1 Device: ASi\_1\_binaryIO**

#### **Information**

Name: ASi\_1\_binaryIO  
Vendor: ifm ecomatic gmbh  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: ASi Master 1 binaryIO from ifm

### **1.2.2 Device: ASi\_1\_analogIO**

#### **Information**

Name: ASi\_1\_analogIO  
Vendor: ifm ecomatic gmbh  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: ASi analog IO from ifm

## ***1.3 Device: ASi\_Master\_2***

### **Information**

Name: ASi Master 2  
Vendor: ifm electronic  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: ASi Master 2 from ifm

### 1.3.1 Device: ASi\_2\_binaryIO

#### **Information**

Name: ASi\_2\_binaryIO  
Vendor: ifm ecomatic gmbh  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: ASi binaryIO from ifm

### 1.3.2 Device: ASi\_2\_analogIO

#### **Information**

Name: ASi\_2\_analogIO  
Vendor: ifm ecomatic gmbh  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: ASi analog IO from ifm

## ***1.4 Device: Fieldbus\_Interface***

#### **Information**

Name: Fieldbus Interface  
Vendor: ifm electronic  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: Fieldbus Interface

### 1.4.1 Device: FieldBusData\_

#### **Information**

Name: FieldBusData  
Vendor: ifm ecomatic gmbh  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: FieldBus Data Words

## 1.4.2 Device: ASi\_1\_binaryOut

### **Information**

Name: ASi\_1\_binaryOut  
Vendor: ifm ecomatic gmbh  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: ASi Master 1 binary Outputs from Fieldbus

## 1.4.3 Device: ASi\_1\_analogOut

### **Information**

Name: ASi\_1\_analogOut  
Vendor: ifm ecomatic gmbh  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: ASi Master 1 analog Outputs from Fieldbus

## 1.4.4 Device: ASi\_2\_binaryOut

### **Information**

Name: ASi\_2\_binaryOut  
Vendor: ifm ecomatic gmbh  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: ASi Master 2 binary Outputs from Fieldbus

## 1.4.5 Device: ASi\_2\_analogOut

### **Information**

Name: ASi\_2\_analogOut  
Vendor: ifm ecomatic gmbh  
Groups:  
Type: 4096  
ID: 1063 0007  
Version: 3.5.4.5  
Description: ASi Master 2 analog Outputs from Fieldbus

## ***1.5 Device: Ethernet***

### **Information**

Name: Ethernet  
Vendor: ifm electronic  
Groups: Ethernet Adapter  
Type: 110  
ID: 1063 0001  
Version: 3.4.2.0  
Order number: -  
Description: Ethernet Link.

## **2 : Project Settings**

Static Analysis Light:

Unused variables (#33): 0  
Overlapping memory areas (#28): 0  
Concurrent access (#6): 0  
Multiple write access on output (#4): 0  
Multiple usage of name (#27): 0