CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF MECHANICAL ENGINEERING

Department of production machines and equipment

Bachelor’s thesis

Attachment 1 – Source Codes for TIA Portal

2019 **Nikita Kuprin**

Table of contents

a. OB Main – original

PLC-Robot_V15 / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

Main [OB1]

Main Properties							
General							
Name	Main	**Number**	1	**Type**	OB	**Language**	SCL
Numbering	Automatic						
Information							
Title	"Main Program Sweep (Cycle)"	**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Main			
Name	**Data type**	**Default value**	**Comment**
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
▼ Temp			
pomProm	Int		
Constant			

```
0001 (*
0002 Toto je hlavni program
0003 *)
0004 //souradnice poloh se definuji ve startupu
0005
0006 IF NOT "di_1_1" THEN            //Obsluha tlačítka "Total stop"
0007   "TS" := TRUE;
0008 ELSIF "pokracovat" THEN
0009   "TS" := FALSE;
0010 END_IF;
0011
0012 CASE "State" OF                      //Hlavní stavový automat
0013   0:  // Default
0014     IF "reference_1_exec" THEN
0015       "State" := 10;
0016     END_IF;
0017   10:  // Reference
0018     "FB_reference_1"(Execute:=TRUE,
0019          Switch:="di_0_0",
0020          Motor=>"do_0_0");
0021     "FB_reference_2"(Execute:=TRUE,
0022          Switch:="di_0_1",
0023          Motor=>"do_0_2");
0024     "FB_reference_3"(Execute:=TRUE,
0025          Switch:="di_0_2",
0026          Motor=>"do_0_4");
0027
0028     IF "FB_reference_1".Done AND "FB_reference_2".Done AND "FB_reference_3".Done THEN
0029       "State" := 20;
0030     END_IF;
0031   20: //Reset counterů
0032     "reset1" := TRUE;
0033     "reset2" := TRUE;
0034     "reset3" := TRUE;
0035     IF "High_Speed_Counter_1".CountValue = 0 AND "High_Speed_Counter_2".CountValue = 0 AND "High_Speed_Coun-
     ter_3".CountValue = 0 THEN
0036       "State" := 30;
0037     END_IF;
0038   30: //Normal mode
0039     "reset1" := FALSE;
0040     "reset2" := FALSE;
0041     "reset3" := FALSE;
0042     IF "reference_1_exec" THEN
0043       "State" := 10;
0044     END_IF;
0045                  //kopmpresor
0046     "FB_goto_1"(Execute := "JedNaExec1",
0047        "GoTo" := "JedNa1",
0048        Actual := "High_Speed_Counter_1".CountValue,
0049           Mplus => "do_0_1",
0050           Mminus => "do_0_0",
0051           State => "GoToState1");
0052     "FB_goto_2"(Execute:="JedNaExec2",
0053       "GoTo":="JedNa2",
0054       Actual:= "High_Speed_Counter_2".CountValue,
0055       Mplus=>"do_0_3",
0056       Mminus=>"do_0_2",
0057       State=>"GoToState2");
0058     "FB_goto_3"(Execute:="JedNaExec3",
0059       "GoTo":="JedNa3",
0060       Actual:= "High_Speed_Counter_3".CountValue,
0061       Mplus=>"do_0_4",
0062       Mminus=>"do_0_5",
0063       State=>"GoToState3");
```

```
0064
0065 IF NOT "moving" THEN                              //ovldádání tlačítek přesunů
0066   IF "PresunExec0" THEN
0067     "moving" := TRUE;
0068     "FB_presun_0"(Exec := TRUE,
0069             X := "pos_0_x",
0070             C := "pos_0_c");
0071   END_IF;
0072   IF "PresunExec1" THEN
0073     "moving" := TRUE;
0074     "FB_presun_0"(Exec := TRUE,
0075             X := "pos_1_x",
0076             C := "pos_1_c");
0077   END_IF;
0078   IF "PresunExec2" THEN
0079     "moving" := TRUE;
0080     "FB_presun_0"(Exec := TRUE,
0081             X := "pos_2_x",
0082             C := "pos_2_c");
0083   END_IF;
0084   IF "PresunExec3" THEN
0085     "moving" := TRUE;
0086     "FB_presun_0"(Exec := TRUE,
0087             X := "pos_3_x",
0088             C := "pos_3_c");
0089   END_IF;
0090   IF "PosunExec0" THEN                              //ovládání tlačítek posunů
0091     "moving" := TRUE;
0092     "FB_posun_0"(Exec := TRUE,
0093            X := "pos_0_x",
0094            C := "pos_0_c");
0095   END_IF;
0096   IF "PosunExec1" THEN
0097     "moving" := TRUE;
0098     "FB_posun_0"(Exec := TRUE,
0099            X := "pos_1_x",
0100            C := "pos_1_c");
0101   END_IF;
0102   IF "PosunExec2" THEN
0103     "moving" := TRUE;
0104     "FB_posun_0"(Exec := TRUE,
0105            X := "pos_2_x",
0106            C := "pos_2_c");
0107   END_IF;
0108   IF "PosunExec3" THEN
0109     "moving" := TRUE;
0110     "FB_posun_0"(Exec := TRUE,
0111            X := "pos_3_x",
0112            C := "pos_3_c");
0113   END_IF;
0114 END_IF;
0115 "FB_presun_0"();                                  //spuštění bloků posunů a přesunů
0116 "FB_posun_0"();
0117 IF "FB_presun_0".Done OR "FB_posun_0".Done THEN     //zjištění ukončení bloků posunů a přesunů
0118   "FB_presun_0"(Exec := FALSE);
0119   "FB_posun_0"(Exec := FALSE);
0120   "moving" := FALSE;
0121 END_IF;
0122 IF NOT "moving" AND "automaticky" THEN      //změna módu na automaticky
0123   "State" := 40;
0124 END_IF;
0125
0126 IF NOT "moving" AND "skorapky" THEN         //změna módu na skořápky
0127   "State" := 60;
0128 END_IF;
0129
0130 40:             //inicializační posun robota před automatickým módem
0131   "moving" := TRUE;
0132   "FB_posun_0"(Exec := TRUE,
0133          X := "pos_0_x",
0134          C := "pos_0_c");
0135   "State" := 50;
0136
0137 50:                              //Automatický mód
0138   IF NOT "moving" AND NOT "automaticky" THEN // přesun zpět do normálního módu
0139     "State" := 30;
0140   END_IF;
0141
0142   IF NOT "moving" THEN                     //pokud je nový obrobek na dopranvíků 0, zvol prázdný dopravník 1,2 nebo 3 a
      přesuň obrobek
0143     IF "Dopravnik0" THEN
0144       IF NOT "Dopravnik1" THEN
0145         "moving" := TRUE;
0146         "FB_presun_0"(Exec := TRUE,
0147                 X := "pos_1_x",
0148                 C := "pos_1_c");
0149         "Dopravnik0" := FALSE;
0150         "Dopravnik1" := TRUE;
```

```
0151        ELSIF NOT "Dopravnik2" THEN
0152          "moving" := TRUE;
0153          "FB_presun_0"(Exec := TRUE,
0154                  X := "pos_2_x",
0155                  C := "pos_2_c");
0156          "Dopravnik0" := FALSE;
0157          "Dopravnik2" := TRUE;
0158        ELSIF NOT "Dopravnik3" THEN
0159          "moving" := TRUE;
0160          "FB_presun_0"(Exec := TRUE,
0161                  X := "pos_3_x",
0162                  C := "pos_3_c");
0163          "Dopravnik0" := FALSE;
0164          "Dopravnik3" := TRUE;
0165        END_IF;
0166      END_IF;
0167    END_IF;
0168    IF "FB_presun_0".Done THEN                    //čekej dokud nebude hotov a se vrať do výchozí pozice
0169      "FB_presun_0"(Exec := FALSE);
0170      "FB_posun_0"(Exec := TRUE,
0171             X := "pos_0_x",
0172             C := "pos_0_c");
0173    END_IF;
0174    IF "FB_posun_0".Done THEN                //pokud je zpět ve výchozí pozici čekej na další pokyn
0175      "FB_posun_0"(Exec := FALSE);
0176      "moving" := FALSE;
0177    END_IF;
0178    "FB_goto_1"(Execute := "JedNaExec1",        //spuštění fb na pohyby os
0179          "GoTo" := "JedNa1",
0180          Actual := "High_Speed_Counter_1".CountValue,
0181          Mplus => "do_0_1",
0182          Mminus => "do_0_0",
0183          State => "GoToState1");
0184    "FB_goto_2"(Execute := "JedNaExec2",
0185          "GoTo" := "JedNa2",
0186          Actual := "High_Speed_Counter_2".CountValue,
0187          Mplus => "do_0_3",
0188          Mminus => "do_0_2",
0189          State => "GoToState1");
0190    "FB_goto_3"(Execute := "JedNaExec3",
0191          "GoTo" := "JedNa3",
0192          Actual := "High_Speed_Counter_3".CountValue,
0193          Mplus => "do_0_4",
0194          Mminus => "do_0_5",
0195          State => "GoToState1");
0196    "FB_presun_0"();
0197    "FB_posun_0"();
0198
0199  60:                  //Skorapky // 4 pozice(0,1,2,3), zacina na 0,1,2
0200    "X_uvnitr":=200;          //inicializace módu skořápky
0201    "X_vne":=1500;
0202    "C_start":=-2300;
0203    "C_inc":=-250;
0204    "prazdne" := 3;
0205    "counter" := 0;
0206    "counter_max" := 10;
0207    "Zbyva" := 0;
0208    "State" := 70;
0209
0210
0211  70:              //skořápky
0212    "Zbyva" := "counter_max" - "counter";        //výpočet "zbývá"
0213    IF "counter" = "counter_max" THEN            // pokud je již dostatečně zamícháno ukonči
0214      "skorapky" := FALSE;
0215      "State" := 30;
0216      "counter" := 0;
0217    ELSE
0218      "random" := "prazdne";
0219      //"random" := ("prazdne" + 1) MOD 3;
0220    WHILE ("random" = "prazdne") DO              //zvol náhodnou pozici
0221      "random"  :=  FLOOR(0.5 * (1 + SIN(DINT_TO_REAL(TIME_TO_DINT(TIME_TCK())) )*4);
0222      //"random" := FLOOR(SIN(DINT_TO_REAL(TIME_TO_DINT(TIME_TCK()))) * 4);
0223    END_WHILE;
0224      "FB_posun_0"(Exec := TRUE,                   //spusť posun
0225             X := "X_vne",
0226             C := "C_start" + "random" * "C_inc");
0227      "State" := 80;
0228    END_IF;
0229  80:
0230    "FB_goto_1"(Execute := "JedNaExec1",        //spuštění fb pro ovládání os
0231          "GoTo" := "JedNa1",
0232          Actual := "High_Speed_Counter_1".CountValue,
0233          Mplus => "do_0_1",
0234          Mminus => "do_0_0",
0235          State => "GoToState1");
0236    "FB_goto_2"(Execute := "JedNaExec2",
0237          "GoTo" := "JedNa2",
0238          Actual := "High_Speed_Counter_2".CountValue,
```

```
0239          Mplus => "do_0_3",
0240          Mminus => "do_0_2",
0241          State => "GoToState1");
0242    "FB_goto_3"(Execute := "JedNaExec3",
0243          "GoTo" := "JedNa3",
0244          Actual := "High_Speed_Counter_3".CountValue,
0245          Mplus => "do_0_4",
0246          Mminus => "do_0_5",
0247          State => "GoToState1");
0248    "FB_posun_0"();
0249    IF "FB_posun_0".Done THEN
0250      "FB_posun_0"(Exec := FALSE);
0251      "State" := 90;
0252    END_IF;
0253  90:
0254    "FB_soup_0"(Exec:=TRUE,
0255          C:="C_start" + "prazdne" * "C_inc");
0256
0257    "State" := 100;
0258  100:
0259    "FB_goto_1"(Execute := "JedNaExec1",
0260          "GoTo" := "JedNa1",
0261          Actual := "High_Speed_Counter_1".CountValue,
0262          Mplus => "do_0_1",
0263          Mminus => "do_0_0",
0264          State => "GoToState1");
0265    "FB_goto_2"(Execute := "JedNaExec2",
0266          "GoTo" := "JedNa2",
0267          Actual := "High_Speed_Counter_2".CountValue,
0268          Mplus => "do_0_3",
0269          Mminus => "do_0_2",
0270          State => "GoToState1");
0271    "FB_goto_3"(Execute := "JedNaExec3",
0272          "GoTo" := "JedNa3",
0273          Actual := "High_Speed_Counter_3".CountValue,
0274          Mplus => "do_0_4",
0275          Mminus => "do_0_5",
0276          State => "GoToState1");
0277    "FB_soup_0"();
0278    IF "FB_soup_0".Done THEN
0279      "FB_soup_0"(Exec := FALSE);
0280      "prazdne" := "random";
0281      "counter" := "counter" + 1;
0282      "State" := 70;
0283    END_IF;
0284
0285  ELSE  // Statement section ELSE
0286      ;
0287  END_CASE;
0288
0289  // realizace 2 citacu (enkodery)
0290  //
0291  "High_Speed_Counter_1"(SwGate:=TRUE, CaptureEnable:=TRUE, SetCountValue:= "reset1");
0292  //
0293  "High_Speed_Counter_2"(SwGate:=TRUE, CaptureEnable:=TRUE, SetCountValue:= "reset2");
0294
0295  "High_Speed_Counter_3"(SwGate:=TRUE, CaptureEnable:=TRUE, SetCountValue:= "reset3");
0296
0297
0298
```

Symbol	Address	Type	Comment
"automaticky"	%M15.1	Bool	
"C_inc"	%MD52	DInt	
"C_start"	%MD48	DInt	
"counter"	%MD60	DInt	
"counter_max"	%MD72	DInt	
"di_0_0"	%I0.0	Bool	
"di_0_1"	%I0.1	Bool	
"di_0_2"	%I0.2	Bool	
"di_1_1"	%I1.1	Bool	
"do_0_0"	%Q0.0	Bool	
"do_0_1"	%Q0.1	Bool	
"do_0_2"	%Q0.2	Bool	
"do_0_3"	%Q0.3	Bool	
"do_0_4"	%Q0.4	Bool	
"do_0_5"	%Q0.5	Bool	
"Dopravnik0"	%M15.2	Bool	
"Dopravnik1"	%M15.3	Bool	
"Dopravnik2"	%M15.4	Bool	
"Dopravnik3"	%M15.5	Bool	
"FB_posun_0".Done		Bool	
"FB_presun_0".Done		Bool	
"FB_reference_1".Done		Bool	
"FB_reference_2".Done		Bool	
"FB_reference_3".Done		Bool	
"FB_soup_0".Done		Bool	
"GoToState1"	%MW13	Int	

Symbol	Address	Type	Comment
"GoToState2"	%MW16	Int	
"GoToState3"	%MW18	Int	
"High_Speed_Counter_1".CountValue		DInt	Current counter value
"High_Speed_Counter_2".CountValue		DInt	Current counter value
"High_Speed_Counter_3".CountValue		DInt	Current counter value
"JedNa1"	%MD20	DInt	
"JedNa2"	%MD24	DInt	
"JedNa3"	%MD28	DInt	
"JedNaExec1"	%M4.4	Bool	
"JedNaExec2"	%M4.5	Bool	
"JedNaExec3"	%M4.6	Bool	
"moving"	%M15.0	Bool	
"pokracovat"	%M34.1	Bool	
"pos_0_c"	%MD80	DInt	
"pos_0_x"	%MD76	DInt	
"pos_1_c"	%MD88	DInt	
"pos_1_x"	%MD84	DInt	
"pos_2_c"	%MD96	DInt	
"pos_2_x"	%MD92	DInt	
"pos_3_c"	%MD104	DInt	
"pos_3_x"	%MD100	DInt	
"PosunExec0"	%M12.6	Bool	
"PosunExec1"	%M12.2	Bool	
"PosunExec2"	%M12.3	Bool	
"PosunExec3"	%M12.4	Bool	
"prazdne"	%MD64	DInt	
"PresunExec0"	%M12.5	Bool	
"PresunExec1"	%M4.7	Bool	
"PresunExec2"	%M12.0	Bool	
"PresunExec3"	%M12.1	Bool	
"random"	%MD56	DInt	
"reference_1_exec"	%M4.0	Bool	
"reset1"	%M4.1	Bool	
"reset2"	%M4.2	Bool	
"reset3"	%M4.3	Bool	
"skorapky"	%M15.6	Bool	
"State"	%MD5	DInt	
"TS"	%M34.0	Bool	
"X_uvnitr"	%MD40	DInt	
"X_vne"	%MD44	DInt	
"Zbyva"	%MD68	DInt	

b. OB Startup – original

PLC-Robot_V15 / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

Startup [OB100]

Startup Properties							
General							
Name	Startup	**Number**	100	**Type**	OB	**Language**	SCL
Numbering	Automatic						
Information							
Title	"Complete Restart"	**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Startup			
Name	**Data type**	**Default value**	**Comment**
▼ Input			
LostRetentive	Bool		True if retentive data are lost
LostRTC	Bool		True if date and time are lost
Temp			
Constant			

```
0001 (*
0002 inicializace
0003 -------------
0004 tento program je spusten pouze 1x pri startu PLC
0005 *)
0006 //definice souradnic poloh
0007 "pos_0_x" := 923;
0008 "pos_0_c" := -1764;
0009 "pos_1_x" := 923;
0010 "pos_1_c" := -2419;
0011 "pos_2_x" := 1362;
0012 "pos_2_c" := -2778;
0013 "pos_3_x" := 2221;
0014 "pos_3_c" := -3073;
0015
0016 // definice počátečních hodnot proměnných
0017 "web_prodleva" := T#100ms;
0018
0019 "do_0_0" := 0;
0020 "do_0_1" := 0;
0021 "do_0_2" := 0;
0022 "do_0_3" := 0;
0023 "do_0_4" := 0;
0024 "do_0_5" := 0;
0025 "do_0_6" := 0;
0026 "do_0_7" := 0;
0027 "do_1_0" := 0;
0028 "do_1_1" := 0;
0029 "do_1_2" := 0;
0030 "do_1_3" := 0;
0031 "do_1_4" := 0;
0032 "do_1_5" := 0;
0033 "do_1_6" := 0;
0034 "do_1_7" := 0;
0035
0036 "pom_do_0_0" := 0;
0037 "pom_do_0_1" := 0;
0038 "pom_do_0_2" := 0;
0039 "pom_do_0_3" := 0;
0040 "pom_do_0_4" := 0;
0041 "pom_do_0_5" := 0;
0042 "pom_do_0_6" := 0;
0043 "pom_do_0_7" := 0;
0044 "pom_do_1_0" := 0;
0045 "pom_do_1_1" := 0;
0046 "pom_do_1_2" := 0;
0047 "pom_do_1_3" := 0;
0048 "pom_do_1_4" := 0;
0049 "pom_do_1_5" := 0;
0050 "pom_do_1_6" := 0;
0051 "pom_do_1_7" := 0;
0052
```

Symbol	Address	Type	Comment
"do_0_0"	%Q0.0	Bool	
"do_0_1"	%Q0.1	Bool	
"do_0_2"	%Q0.2	Bool	
"do_0_3"	%Q0.3	Bool	
"do_0_4"	%Q0.4	Bool	
"do_0_5"	%Q0.5	Bool	
"do_0_6"	%Q0.6	Bool	
"do_0_7"	%Q0.7	Bool	
"do_1_0"	%Q1.0	Bool	
"do_1_1"	%Q1.1	Bool	
"do_1_2"	%Q1.2	Bool	
"do_1_3"	%Q1.3	Bool	

Symbol	Address	Type	Comment
"do_1_4"	%Q1.4	Bool	
"do_1_5"	%Q1.5	Bool	
"do_1_6"	%Q1.6	Bool	
"do_1_7"	%Q1.7	Bool	
"pom_do_0_0"	%M2.0	Bool	
"pom_do_0_1"	%M2.1	Bool	
"pom_do_0_2"	%M2.2	Bool	
"pom_do_0_3"	%M2.3	Bool	
"pom_do_0_4"	%M2.4	Bool	
"pom_do_0_5"	%M2.5	Bool	
"pom_do_0_6"	%M2.6	Bool	
"pom_do_0_7"	%M2.7	Bool	
"pom_do_1_0"	%M3.0	Bool	
"pom_do_1_1"	%M3.1	Bool	
"pom_do_1_2"	%M3.2	Bool	
"pom_do_1_3"	%M3.3	Bool	
"pom_do_1_4"	%M3.4	Bool	
"pom_do_1_5"	%M3.5	Bool	
"pom_do_1_6"	%M3.6	Bool	
"pom_do_1_7"	%M3.7	Bool	
"pos_0_c"	%MD80	DInt	
"pos_0_x"	%MD76	DInt	
"pos_1_c"	%MD88	DInt	
"pos_1_x"	%MD84	DInt	
"pos_2_c"	%MD96	DInt	
"pos_2_x"	%MD92	DInt	
"pos_3_c"	%MD104	DInt	
"pos_3_x"	%MD100	DInt	
"web_prodleva"	%MD8	Time	

c. prg_obsluha_DO

Robot_pick_&_place / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

prg_obsluha_DO [OB123]

prg_obsluha_DO Properties							
General							
Name	prg_obsluha_DO	**Number**	123	**Type**	OB	**Language**	SCL
Numbering	Automatic						
Information							
Title	"Main Program Sweep (Cycle)"	**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

prg_obsluha_DO			
Name	**Data type**	**Default value**	**Comment**
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

```
0001 // reseni impulzniho spinani digitalnich vystupu z HMI panelu - vyhodnocovací část
0002 "IEC_Timer_DQ0-0_DB".TP(IN:="HMI_Data_block".web_di_0_0,
0003             PT:="web_prodleva");
0004  "IEC_Timer_DQ0-1_DB".TP(IN := "HMI_Data_block".web_di_0_1,
0005             PT := "web_prodleva");
0006 "IEC_Timer_DQ0-2_DB".TP(IN := "HMI_Data_block".web_di_0_2,
0007             PT := "web_prodleva");
0008 "IEC_Timer_DQ0-3_DB".TP(IN := "HMI_Data_block".web_di_0_3,
0009             PT := "web_prodleva");
0010 "IEC_Timer_DQ0-4_DB".TP(IN := "HMI_Data_block".web_di_0_4,
0011             PT := "web_prodleva");
0012 "IEC_Timer_DQ0-5_DB".TP(IN := "HMI_Data_block".web_di_0_5,
0013             PT := "web_prodleva");
0014 "IEC_Timer_DQ0-6_DB".TP(IN := "HMI_Data_block".web_di_0_6,
0015             PT := "web_prodleva");
0016 "IEC_Timer_DQ0-7_DB".TP(IN := "HMI_Data_block".web_di_0_7,
0017             PT := "web_prodleva");
0018 "IEC_Timer_DQ1-0_DB".TP(IN := "HMI_Data_block".web_di_1_0,
0019             PT := "web_prodleva");
0020 "IEC_Timer_DQ1-1_DB".TP(IN := "HMI_Data_block".web_di_1_1,
0021             PT := "web_prodleva");
0022 "IEC_Timer_DQ1-2_DB".TP(IN := "HMI_Data_block".web_di_1_2,
0023             PT := "web_prodleva");
0024 "IEC_Timer_DQ1-3_DB".TP(IN := "HMI_Data_block".web_di_1_3,
0025             PT := "web_prodleva");
0026 "IEC_Timer_DQ1-4_DB".TP(IN := "HMI_Data_block".web_di_1_4,
0027             PT := "web_prodleva");
0028 "IEC_Timer_DQ1-5_DB".TP(IN := "HMI_Data_block".web_di_1_5,
0029             PT := "web_prodleva");
0030 "IEC_Timer_DQ1-6_DB".TP(IN := "HMI_Data_block".web_di_1_6,
0031             PT := "web_prodleva");
0032 "IEC_Timer_DQ1-7_DB".TP(IN := "HMI_Data_block".web_di_1_7,
0033             PT := "web_prodleva");
0034
0035
0036 //trvale sepnuti digitalnich vystupu z HMI panelu - vyhodnocovací část, detekce náběžné hrany
0037 "R_TRIG_DB_DO_0_0"(CLK := "HMI_Data_block".web2_di_0_0);   // detekce náběžné hrany tlačítka
0038 "R_TRIG_DB_DO_0_1"(CLK := "HMI_Data_block".web2_di_0_1);
0039 "R_TRIG_DB_DO_0_2"(CLK := "HMI_Data_block".web2_di_0_2);
0040 "R_TRIG_DB_DO_0_3"(CLK := "HMI_Data_block".web2_di_0_3);
0041 "R_TRIG_DB_DO_0_4"(CLK := "HMI_Data_block".web2_di_0_4);
0042 "R_TRIG_DB_DO_0_5"(CLK := "HMI_Data_block".web2_di_0_5);
0043 "R_TRIG_DB_DO_0_6"(CLK := "HMI_Data_block".web2_di_0_6);
0044 "R_TRIG_DB_DO_0_7"(CLK := "HMI_Data_block".web2_di_0_7);
0045 "R_TRIG_DB_DO_1_0"(CLK := "HMI_Data_block".web2_di_1_0);
0046 "R_TRIG_DB_DO_1_1"(CLK := "HMI_Data_block".web2_di_1_1);
0047 "R_TRIG_DB_DO_1_2"(CLK := "HMI_Data_block".web2_di_1_2);
0048 "R_TRIG_DB_DO_1_3"(CLK := "HMI_Data_block".web2_di_1_3);
0049 "R_TRIG_DB_DO_1_4"(CLK := "HMI_Data_block".web2_di_1_4);
0050 "R_TRIG_DB_DO_1_5"(CLK := "HMI_Data_block".web2_di_1_5);
0051 "R_TRIG_DB_DO_1_6"(CLK := "HMI_Data_block".web2_di_1_6);
0052 "R_TRIG_DB_DO_1_7"(CLK := "HMI_Data_block".web2_di_1_7);
0053
0054
0055 // pokud je náběžná hrana tlačítka prepinaciho dig.vystup, tak neguje stav pomocné proměnné výstupu
0056  IF "R_TRIG_DB_DO_0_0".Q THEN
0057    "pom_do_0_0":= NOT "pom_do_0_0";
0058 END_IF;
0059 IF "R_TRIG_DB_DO_0_1".Q THEN
0060   "pom_do_0_1" := NOT "pom_do_0_1";
0061 END_IF;
0062 IF "R_TRIG_DB_DO_0_2".Q THEN
0063   "pom_do_0_2" := NOT "pom_do_0_2";
0064 END_IF;
0065 IF "R_TRIG_DB_DO_0_3".Q THEN
```

```
0066   "pom_do_0_3" := NOT "pom_do_0_3";
0067 END_IF;
0068 IF "R_TRIG_DB_DO_0_4".Q THEN
0069   "pom_do_0_4" := NOT "pom_do_0_4";
0070 END_IF;
0071 IF "R_TRIG_DB_DO_0_5".Q THEN
0072   "pom_do_0_5" := NOT "pom_do_0_5";
0073 END_IF;
0074 IF "R_TRIG_DB_DO_0_6".Q THEN
0075   "pom_do_0_6" := NOT "pom_do_0_6";
0076 END_IF;
0077 IF "R_TRIG_DB_DO_0_7".Q THEN
0078   "pom_do_0_7" := NOT "pom_do_0_7";
0079 END_IF;
0080 IF "R_TRIG_DB_DO_1_0".Q THEN
0081   "pom_do_1_0" := NOT "pom_do_1_0";
0082 END_IF;
0083 IF "R_TRIG_DB_DO_1_1".Q THEN
0084   "pom_do_1_1" := NOT "pom_do_1_1";
0085 END_IF;
0086 IF "R_TRIG_DB_DO_1_2".Q THEN
0087   "pom_do_1_2" := NOT "pom_do_1_2";
0088 END_IF;
0089 IF "R_TRIG_DB_DO_1_3".Q THEN
0090   "pom_do_1_3" := NOT "pom_do_1_3";
0091 END_IF;
0092 IF "R_TRIG_DB_DO_1_4".Q THEN
0093   "pom_do_1_4" := NOT "pom_do_1_4";
0094 END_IF;
0095 IF "R_TRIG_DB_DO_1_5".Q THEN
0096   "pom_do_1_5" := NOT "pom_do_1_5";
0097 END_IF;
0098 IF "R_TRIG_DB_DO_1_6".Q THEN
0099   "pom_do_1_6" := NOT "pom_do_1_6";
0100 END_IF;
0101 IF "R_TRIG_DB_DO_1_7".Q THEN
0102   "pom_do_1_7" := NOT "pom_do_1_7";
0103 END_IF;
0104
0105 //zápis stavu na výstupy
0106 (*
0107 "do_0_0" := "IEC_Timer_DQ0-0_DB".Q OR "pom_do_0_0";
0108 "do_0_1" := "IEC_Timer_DQ0-1_DB".Q OR "pom_do_0_1";
0109 "do_0_2" := "IEC_Timer_DQ0-2_DB".Q OR "pom_do_0_2";
0110 "do_0_3" := "IEC_Timer_DQ0-3_DB".Q OR "pom_do_0_3";
0111 "do_0_4" := "IEC_Timer_DQ0-4_DB".Q OR "pom_do_0_4";
0112 "do_0_5" := "IEC_Timer_DQ0-5_DB".Q OR "pom_do_0_5";*)
0113
0114 "do_kompresor" := "IEC_Timer_DQ0-6_DB".Q OR "pom_do_0_6";
0115 (*
0116 "do_0_7" := "IEC_Timer_DQ0-7_DB".Q OR "pom_do_0_7";
0117 "do_1_0" := "IEC_Timer_DQ1-0_DB".Q OR "pom_do_1_0";
0118 "do_1_1" := "IEC_Timer_DQ1-1_DB".Q OR "pom_do_1_1";
0119 "do_1_2" := "IEC_Timer_DQ1-2_DB".Q OR "pom_do_1_2";
0120 "do_1_3" := "IEC_Timer_DQ1-3_DB".Q OR "pom_do_1_3";
0121 "do_1_4" := "IEC_Timer_DQ1-4_DB".Q OR "pom_do_1_4";
0122 "do_1_5" := "IEC_Timer_DQ1-5_DB".Q OR "pom_do_1_5";
0123 "do_1_6" := "IEC_Timer_DQ1-6_DB".Q OR "pom_do_1_6";
0124 "do_1_7" := "IEC_Timer_DQ1-7_DB".Q OR "pom_do_1_7";
0125 *)
0126
0127
0128
0129
0130
```

Symbol	Address	Type	Comment
"do_kompresor"	%Q0.6	Bool	
"HMI_Data_block".web2_di_0_0		Bool	
"HMI_Data_block".web2_di_0_1		Bool	
"HMI_Data_block".web2_di_0_2		Bool	
"HMI_Data_block".web2_di_0_3		Bool	
"HMI_Data_block".web2_di_0_4		Bool	
"HMI_Data_block".web2_di_0_5		Bool	
"HMI_Data_block".web2_di_0_6		Bool	
"HMI_Data_block".web2_di_0_7		Bool	
"HMI_Data_block".web2_di_1_0		Bool	
"HMI_Data_block".web2_di_1_1		Bool	
"HMI_Data_block".web2_di_1_2		Bool	
"HMI_Data_block".web2_di_1_3		Bool	
"HMI_Data_block".web2_di_1_4		Bool	
"HMI_Data_block".web2_di_1_5		Bool	
"HMI_Data_block".web2_di_1_6		Bool	
"HMI_Data_block".web2_di_1_7		Bool	
"HMI_Data_block".web_di_0_0		Bool	
"HMI_Data_block".web_di_0_1		Bool	
"HMI_Data_block".web_di_0_2		Bool	
"HMI_Data_block".web_di_0_3		Bool	

Symbol	Address	Type	Comment
"HMI_Data_block".web_di_0_4		Bool	
"HMI_Data_block".web_di_0_5		Bool	
"HMI_Data_block".web_di_0_6		Bool	
"HMI_Data_block".web_di_0_7		Bool	
"HMI_Data_block".web_di_1_0		Bool	
"HMI_Data_block".web_di_1_1		Bool	
"HMI_Data_block".web_di_1_2		Bool	
"HMI_Data_block".web_di_1_3		Bool	
"HMI_Data_block".web_di_1_4		Bool	
"HMI_Data_block".web_di_1_5		Bool	
"HMI_Data_block".web_di_1_6		Bool	
"HMI_Data_block".web_di_1_7		Bool	
"IEC_Timer_DQ0-6_DB".Q		Bool	
"pom_do_0_0"	%M2.0	Bool	
"pom_do_0_1"	%M2.1	Bool	
"pom_do_0_2"	%M2.2	Bool	
"pom_do_0_3"	%M2.3	Bool	
"pom_do_0_4"	%M2.4	Bool	
"pom_do_0_5"	%M2.5	Bool	
"pom_do_0_6"	%M2.6	Bool	
"pom_do_0_7"	%M2.7	Bool	
"pom_do_1_0"	%M3.0	Bool	
"pom_do_1_1"	%M3.1	Bool	
"pom_do_1_2"	%M3.2	Bool	
"pom_do_1_3"	%M3.3	Bool	
"pom_do_1_4"	%M3.4	Bool	
"pom_do_1_5"	%M3.5	Bool	
"pom_do_1_6"	%M3.6	Bool	
"pom_do_1_7"	%M3.7	Bool	
"R_TRIG_DB_DO_0_0".Q		Bool	
"R_TRIG_DB_DO_0_1".Q		Bool	
"R_TRIG_DB_DO_0_2".Q		Bool	
"R_TRIG_DB_DO_0_3".Q		Bool	
"R_TRIG_DB_DO_0_4".Q		Bool	
"R_TRIG_DB_DO_0_5".Q		Bool	
"R_TRIG_DB_DO_0_6".Q		Bool	
"R_TRIG_DB_DO_0_7".Q		Bool	
"R_TRIG_DB_DO_1_0".Q		Bool	
"R_TRIG_DB_DO_1_1".Q		Bool	
"R_TRIG_DB_DO_1_2".Q		Bool	
"R_TRIG_DB_DO_1_3".Q		Bool	
"R_TRIG_DB_DO_1_4".Q		Bool	
"R_TRIG_DB_DO_1_5".Q		Bool	
"R_TRIG_DB_DO_1_6".Q		Bool	
"R_TRIG_DB_DO_1_7".Q		Bool	
"web_prodleva"	%MD8	Time	

d. FB_posun – original

PLC-Robot_V15 / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

FB_posun [FB4]

FB_posun Properties							
General							
Name	FB_posun	**Number**	4	**Type**	FB	**Language**	SCL
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

FB_posun

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
Input									
Exec	Bool	false	Non-retain	True	True	True	False		
Z_Up	DInt	3000	Non-retain	True	True	True	False		
Z_Down	DInt	3758	Non-retain	True	True	True	False		
X	DInt	923	Non-retain	True	True	True	False		
C	DInt	-2419	Non-retain	True	True	True	False		
Output									
Done	Bool	false	Non-retain	True	True	True	False		
State	Int	0	Non-retain	True	True	True	False		
InOut									
Static									
Temp									
Constant									

```
0001 CASE #State OF //pro pohyb třemi osami
0002   0:  // Statement section case 1
0003     #Done := FALSE;
0004     IF #Exec THEN
0005       #State:=30;
0006     END_IF;
0007   30:
0008     "JedNa1" := #Z_Up;              // Z
0009     "JedNa2" := #X;                 //XC
0010     "JedNa3" := #C;
0011     "JedNaExec1" := TRUE;
0012     "JedNaExec2" := TRUE;
0013     "JedNaExec3" := TRUE;
0014     #State := 40;
0015   40:
0016     IF "FB_goto_1".Done AND "FB_goto_2".Done AND "FB_goto_3".Done THEN
0017       "JedNaExec1" := FALSE;
0018       "JedNaExec2" := FALSE;
0019       "JedNaExec3" := FALSE;
0020       #State := 70;
0021     END_IF;
0022   70:
0023     #Done := TRUE;
0024     IF NOT #Exec THEN
0025       #State := 0;
0026     END_IF;
0027 END_CASE;
0028
```

Symbol	Address	Type	Comment
"FB_goto_1".Done		Bool	
"FB_goto_2".Done		Bool	
"FB_goto_3".Done		Bool	
"JedNa1"	%MD20	DInt	
"JedNa2"	%MD24	DInt	
"JedNa3"	%MD28	DInt	
"JedNaExec1"	%M4.4	Bool	
"JedNaExec2"	%M4.5	Bool	
"JedNaExec3"	%M4.6	Bool	
#C		DInt	
#Done		Bool	
#Exec		Bool	
#State		Int	
#X		DInt	
#Z_Up		DInt	

e. FB_presun – original

PLC-Robot_V15 / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

FB_presun [FB3]

FB_presun Properties							
General							
Name	FB_presun	**Number**	3	**Type**	FB	**Language**	SCL
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

FB_presun

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
Exec	Bool	false	Non-retain	True	True	True	False		
Z_Up	DInt	3000	Non-retain	True	True	True	False		
Z_Down	DInt	3758	Non-retain	True	True	True	False		
X	DInt	923	Non-retain	True	True	True	False		
C	DInt	-2419	Non-retain	True	True	True	False		
fb_posun	Block_FB	FB 0	Non-retain	True	True	True	False		
▼ Output									
Done	Bool	false	Non-retain	True	True	True	False		
State	Int	0	Non-retain	True	True	True	False		
InOut									
Static									
Temp									
Constant									

```
0001 CASE #State OF  //pro přesunutí obrobku z aktuální pozice na zvolenou pozici
0002   0:  // Statement section case 1
0003     #Done := FALSE;
0004     IF #Exec THEN
0005       #State:=2;
0006     END_IF;
0007   2:
0008     "pom_do_0_6" := TRUE;
0009     "JedNa1" := #Z_Down;           // DOWN
0010     "JedNaExec1" := TRUE;
0011     #State := 4;
0012   4:
0013     IF "FB_goto_1".Done THEN
0014       "JedNaExec1" := FALSE;
0015       #State := 10;
0016     END_IF;
0017   10:
0018     "do_0_7" := TRUE;    //prisavka ON
0019     #State:=20;
0020   20:
0021     "JedNa1" := #Z_Up;           // UP
0022     "JedNaExec1" := TRUE;
0023     #State := 30;
0024   30:
0025         IF "FB_goto_1".Done THEN
0026           "JedNaExec1" := FALSE;
0027           #State := 40;
0028         END_IF;
0029       40:
0030         "JedNa2" := #X;                  // pohyb os XC
0031         "JedNa3" := #C;
0032         "JedNaExec2" := TRUE;
0033         "JedNaExec3" := TRUE;
0034         #State := 50;
0035       50:
0036         IF "FB_goto_2".Done AND "FB_goto_3".Done THEN
0037           "JedNaExec2" := FALSE;
0038           "JedNaExec3" := FALSE;
0039           #State := 60;
0040         END_IF;
0041       60:
0042         "JedNa1" := #Z_Down;         //DOWN
0043         "JedNaExec1" := TRUE;
0044         #State := 70;
0045       70:
0046         IF "FB_goto_1".Done THEN
0047           "JedNaExec1" := FALSE;
0048           #State := 80;
0049         END_IF;
0050   80:
0051     "do_0_7" := FALSE;               //prisavka off
0052     "pom_do_0_6" := FALSE;
```

```
0053      "IEC_Timer_0_DB".TON(IN:=TRUE,
0054                   PT:= t#1s);
0055      //pockej chvilu
0056      IF "IEC_Timer_0_DB".Q THEN
0057        #State := 90;
0058      END_IF;
0059    90:
0060      "IEC_Timer_0_DB".TON(IN := FALSE,
0061                   PT := t#1s);
0062      "JedNa1" := #Z_Up;           // UP
0063      "JedNaExec1" := TRUE;
0064      #State := 100;
0065    100:
0066      IF "FB_goto_1".Done THEN
0067        "JedNaExec1" := FALSE;
0068        #State := 110;
0069      END_IF;
0070    110:                    //done
0071      #Done := TRUE;
0072      IF NOT #Exec THEN
0073        #State := 0;
0074      END_IF;
0075  END_CASE;
0076
```

Symbol	Address	Type	Comment
"do_0_7"	%Q0.7	Bool	
"FB_goto_1".Done		Bool	
"FB_goto_2".Done		Bool	
"FB_goto_3".Done		Bool	
"IEC_Timer_0_DB".Q		Bool	
"JedNa1"	%MD20	DInt	
"JedNa2"	%MD24	DInt	
"JedNa3"	%MD28	DInt	
"JedNaExec1"	%M4.4	Bool	
"JedNaExec2"	%M4.5	Bool	
"JedNaExec3"	%M4.6	Bool	
"pom_do_0_6"	%M2.6	Bool	
#C		DInt	
#Done		Bool	
#Exec		Bool	
#State		Int	
#X		DInt	
#Z_Down		DInt	
#Z_Up		DInt	

f. FB_goto – original

PLC-Robot_V15 / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

FB_goto [FB2]

FB_goto Properties							
General							
Name	FB_goto	**Number**	2	**Type**	FB	**Language**	SCL
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

FB_goto

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
Execute	Bool	false	Non-retain	True	True	True	False		
GoTo	DInt	0	Non-retain	True	True	True	False		
Actual	DInt	0	Non-retain	True	True	True	False		
Limit	DInt	10	Non-retain	True	True	True	False		
EndMinus	DInt	0	Non-retain	True	True	True	False		
EndPlus	DInt	0	Non-retain	True	True	True	False		
▼ Output									
Mplus	Bool	false	Non-retain	True	True	True	False		
Mminus	Bool	false	Non-retain	True	True	True	False		
Done	Bool	false	Non-retain	True	True	True	False		
State	Int	0	Non-retain	True	True	True	False		
InOut									
Static									
▼ Temp									
Diff	DInt								
Constant									

```
0001 IF "TS" THEN //Bezpečnost
0002   #Mminus := FALSE;
0003   #Mplus := FALSE;
0004   #State := 0;
0005 ELSE
0006   IF #GoTo < #EndPlus AND #GoTo > #EndMinus THEN          // jsme v soft limitech?
0007
0008   #Diff := #GoTo - #Actual; //výpočet rozdílu dráhy
0009   CASE #State OF
0010     0:                            //INIT
0011       #Mplus := FALSE;
0012       #Mminus := FALSE;
0013       #Done := FALSE;
0014       IF #Execute THEN
0015         #State := 10;
0016       END_IF;
0017     10:                                    //MOVE
0018       IF NOT #Execute THEN
0019         #State := 0;
0020       END_IF;
0021       IF #Diff > #Limit THEN                  //Smer plus
0022         #Mplus := TRUE;
0023         #Mminus := FALSE;
0024       ELSIF #Diff < - #Limit THEN              //Smer minus
0025         #Mplus := FALSE;
0026         #Mminus := TRUE;
0027       ELSE                                    //Hotov
0028         #State := 20;
0029       END_IF;
0030     20:                                    //DONE
0031       #Mplus := FALSE;
0032       #Mminus := FALSE;
0033       #Done := TRUE;
0034       IF NOT #Execute THEN
0035         #State := 0;
0036       END_IF;
0037   END_CASE;
0038
0039 ELSE    //konec
0040   #Mplus := FALSE;
0041   #Mminus := FALSE;
0042 END_IF;
0043
0044 END_IF;
0045 (*
0046 #Diff := #GoTo - #Actual;
0047 CASE #State OF
0048  0: // Statement section case 1
```

```
0049  #Done := FALSE;
0050  IF #Execute THEN
0051  IF #Diff > 0 THEN
0052  #State := 10; //Smer plus
0053  ELSE
0054  #State := 20; //Smer minus
0055  END_IF;
0056  END_IF;
0057  10: // Smer plus
0058  #Mplus := TRUE;
0059  IF #Diff <= 0 THEN
0060  #Mplus := FALSE;
0061  #State := 30;
0062  END_IF;
0063  20:
0064  #Mminus := TRUE;
0065  IF #Diff >= 0 THEN
0066  #Mplus := FALSE;
0067  #State := 30;
0068  END_IF;
0069  30:
0070  #Mplus := FALSE;
0071  #Mminus := FALSE;
0072  #Done := TRUE;
0073  #State := 40;
0074  40:
0075  IF NOT #Execute THEN
0076  #State := 0;
0077  END_IF;
0078
0079  END_CASE;
0080  *)
```

Symbol	Address	Type	Comment
"TS"	%M34.0	Bool	
#Actual		DInt	
#Diff		DInt	
#Done		Bool	
#EndMinus		DInt	
#EndPlus		DInt	
#Execute		Bool	
#GoTo		DInt	
#Limit		DInt	
#Mminus		Bool	
#Mplus		Bool	
#State		Int	

g. OB Main – modified

Robot_pick_&_place / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

Main [OB1]

Main Properties					
General					
Name	Main	**Number**	1	**Type**	OB
Language	SCL	**Numbering**	Automatic		
Information					
Title	"Main Program Sweep (Cycle)"	**Author**		**Comment**	
Family		**Version**	0.1	**User-defined ID**	

Main		
Name	**Data type**	**Default value**
▼ Input		
Initial_Call	Bool	
Remanence	Bool	
▼ Temp		
pomProm	Int	
Constant		

```
0001 (*
0002 This is the main program
0003 *)
0004
0005 IF NOT "di_central_stop" THEN //Total stop
0006   "TS" := TRUE;
0007 ELSIF "pokracovat" THEN //Resume after Total stop
0008   "TS" := FALSE;
0009 END_IF;
0010
0011 //Main state machine
0012 CASE "State" OF
0013   0:  // Default mode
0014     IF "reference_1_exec" THEN //Reference button
0015       "State" := 10;
0016     END_IF;
0017
0018
0019   10:  // Reference mode
0020     "FB_reference_1"(Execute := TRUE, //referencing z-axis (vertical)
0021              Switch := "di_snimac_osa_z",
0022              Motor => "do_osa_z_nahoru");
0023     "FB_reference_2"(Execute := TRUE, //referencing y-axis (horizontal)
0024              Switch := "di_snimac_osa_y",
0025              Motor => "do_osa_y_vzad");
0026     "FB_reference_3"(Execute := TRUE, //referencing x-axis (rotation)
0027              Switch := "di_snimac_osa_x",
0028              Motor => "do_osa_x_smer");
0029
0030     IF "FB_reference_1".Done AND "FB_reference_2".Done AND "FB_refer-
     ence_3".Done THEN
0031       "State" := 20;
0032     END_IF;
0033
```

```
0034    20: //Resetting the counters
0035      "reset1" := TRUE;
0036      "reset2" := TRUE;
0037      "reset3" := TRUE;
0038      IF "High_Speed_Counter_1".CountValue = 0 AND "High_Speed_Counter_2".Count-
      Value = 0 AND "High_Speed_Counter_3".CountValue = 0 THEN
0039        "State" := 30;
0040      END_IF;
0041
0042    30: //Normal mode
0043      "moving" := FALSE; //default values of the tags
0044      "locate" := FALSE;
0045      "found" := FALSE;
0046      "reset1" := FALSE;
0047      "reset2" := FALSE;
0048      "reset3" := FALSE;
0049
0050      IF "reference_1_exec" THEN //Reference button action
0051        "State" := 10;
0052      END_IF;
0053
0054      //Operation of manual mode
0055      "FB_manual_1"();
0056
0057      //Locate the part
0058      IF NOT "moving" AND "LocateExec" THEN
0059        "State" := 40;
0060      END_IF;
0061
0062    40: //Initialize p&p mode (to initiate detection on myRIO)
0063      "locate" := TRUE;
0064
0065      IF "found" THEN
0066        "locate" := FALSE;
0067        "State" := 50;
0068      END_IF;
0069
0070      IF "ResetExec" THEN //Back to Normal mode (press of Reset button -> man-
      ual reset if object hasn't been found)
0071        "State" := 30;
0072      END_IF;
0073
0074    50: //P&P mode
0075      IF "ResetExec" THEN //Back to Refernce mode (manual reset in case of er-
      ror)
0076        "State" := 10;
0077      END_IF;
0078
0079      //Go to part
0080      IF "p&pExec" AND NOT "moving" THEN //sending manipulator to the part
0081        "moving" := TRUE;
0082        "FB_posun_1"(Exec := TRUE,
0083                     X := "pos_x",
0084                     C := "pos_c");
0085      END_IF;
0086
0087      //Pick & place the part
0088      IF "FB_posun_1".Done THEN //sending manipulator to pick the part and
      place it to storage
```

```
0089        "FB_posun_1"(Exec := FALSE);
0090        IF "Colour" >= 0 AND "Colour" < 1 AND "Shape" >= 1 AND "Shape" < 2
      THEN //red square
0091          "FB_presun_0"(Exec := TRUE,
0092                  X := "pos_01_x",
0093                  C := "pos_01_c");
0094        ELSIF "Colour" >= 0 AND "Colour" < 1 AND "Shape" >= 2 THEN //red circle
0095          "FB_presun_0"(Exec := TRUE,
0096                  X := "pos_02_x",
0097                  C := "pos_02_c");
0098        ELSIF "Colour" >= 1 AND "Colour" < 2 AND "Shape" >= 1 AND "Shape" < 2
      THEN //blue circle
0099          "FB_presun_0"(Exec := TRUE,
0100                  X := "pos_12_x",
0101                  C := "pos_12_c");
0102        ELSIF "Colour" >= 1 AND "Colour" < 2 AND "Shape" >= 2 THEN //blue square
0103          "FB_presun_0"(Exec := TRUE,
0104                  X := "pos_11_x",
0105                  C := "pos_11_c");
0106        ELSIF "Colour" >= 2 AND "Shape" >= 1 AND "Shape" < 2 THEN //green square
0107          "FB_presun_0"(Exec := TRUE,
0108                  X := "pos_21_x",
0109                  C := "pos_21_c");
0110        ELSIF "Colour" >= 2 AND "Shape" >= 2 THEN //green circle
0111          "FB_presun_0"(Exec := TRUE,
0112                  X := "pos_22_x",
0113                  C := "pos_22_c");
0114        END_IF;
0115      END_IF;
0116
0117      //Reference when done
0118      IF "FB_presun_0".Done THEN
0119        "FB_presun_0"(Exec := FALSE);
0120        "moving" := FALSE;
0121        "State" := 10;
0122      END_IF;
0123
0124      "FB_goto_1"(Execute := "JedNaExec1", //FBs for moving along the axis
0125            "GoTo" := "JedNa1",
0126            Actual := "High_Speed_Counter_1".CountValue,
0127            Mplus => "do_osa_z_dolu",
0128            Mminus => "do_osa_z_nahoru",
0129            State => "GoToState1");
0130      "FB_goto_2"(Execute := "JedNaExec2",
0131            "GoTo" := "JedNa2",
0132            Actual := "High_Speed_Counter_2".CountValue,
0133            Mplus => "do_osa_y_vpred",
0134            Mminus => "do_osa_y_vzad",
0135            State => "GoToState1");
0136      "FB_goto_3"(Execute := "JedNaExec3",
0137            "GoTo" := "JedNa3",
0138            Actual := "High_Speed_Counter_3".CountValue,
0139            Mplus => "do_osa_x_smer",
0140            Mminus => "do_osa_x_protismer",
0141            State => "GoToState1");
0142
0143      "FB_presun_0"(); //calling required FBs
0144      "FB_posun_1"();
0145  END_CASE;
```

```
0146
0147 //Encoders
0148 "High_Speed_Counter_1"(SwGate := TRUE,
0149                 CaptureEnable := TRUE,
0150                 SetCountValue := "reset1");
0151 //
0152 "High_Speed_Counter_2"(SwGate := TRUE,
0153                 CaptureEnable := TRUE,
0154                 SetCountValue := "reset2");
0155
0156 "High_Speed_Counter_3"(SwGate := TRUE,
0157                 CaptureEnable := TRUE,
0158                 SetCountValue := "reset3");
0159
0160 // Only for PLSSIM!!!
0161 (*"High_Speed_Counter_1".CountValue := "High_Speed_Counter_1".MeasuredValue;
0162 "High_Speed_Counter_2".CountValue := "High_Speed_Counter_2".MeasuredValue;
0163 "High_Speed_Counter_3".CountValue := "High_Speed_Counter_3".MeasuredValue;*)
0164
0165
0166
0167
```

h. OB Startup – modified

Robot_pick_&_place / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

Startup [OB100]

Startup Properties					
General					
Name	Startup	**Number**	100	**Type**	OB
Language	SCL	**Numbering**	Automatic		
Information					
Title	"Complete Restart"	**Author**		**Comment**	
Family		**Version**	0.1	**User-defined ID**	

Startup		
Name	**Data type**	**Default value**
▼ Input		
LostRetentive	Bool	
LostRTC	Bool	
Temp		
Constant		

```
0001 (*
0002 initialization
0003 -------------
0004 this program is run only 1x during PLC startup
0005 *)
0006 //coordinates
0007 "pos_02_x" := 554; //red circle
0008 "pos_02_c" := -2327;
0009 "pos_01_x" := 1903; //red square
0010 "pos_01_c" := -2309;
0011 "pos_11_x" := 984; //green square
0012 "pos_11_c" := -1687;
0013 "pos_12_x" := 2272; //green circle
0014 "pos_12_c" := -1760;
0015 "pos_21_x" := 1942; //blue square
0016 "pos_21_c" := -2033;
0017 "pos_22_x" := 594; //blue circle
0018 "pos_22_c" := -1977;
0019
0020 // initial values
0021 "web_prodleva" := T#100ms;
0022
0023 "do_osa_z_nahoru" := 0;
0024 "do_osa_z_dolu" := 0;
0025 "do_osa_y_vzad" := 0;
0026 "do_osa_y_vpred" := 0;
0027 "do_osa_x_smer" := 0;
0028 "do_osa_x_protismer" := 0;
0029 "do_kompresor" := 0;
0030 "do_vacuum" := 0;
0031 "do_1_0" := 0;
0032 "do_1_1" := 0;
0033 "do_1_2" := 0;
0034 "do_1_3" := 0;
0035 "do_1_4" := 0;
0036 "do_1_5" := 0;
```

```
0037 "do_1_6" := 0;
0038 "do_1_7" := 0;
0039
0040 "pom_do_0_0" := 0;
0041 "pom_do_0_1" := 0;
0042 "pom_do_0_2" := 0;
0043 "pom_do_0_3" := 0;
0044 "pom_do_0_4" := 0;
0045 "pom_do_0_5" := 0;
0046 "pom_do_0_6" := 0;
0047 "pom_do_0_7" := 0;
0048 "pom_do_1_0" := 0;
0049 "pom_do_1_1" := 0;
0050 "pom_do_1_2" := 0;
0051 "pom_do_1_3" := 0;
0052 "pom_do_1_4" := 0;
0053 "pom_do_1_5" := 0;
0054 "pom_do_1_6" := 0;
0055 "pom_do_1_7" := 0;
0056
```

i. FB_manual

Robot_pick_&_place / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

FB_manual [FB6]

FB_manual Properties					
General					
Name	FB_manual	**Number**	6	**Type**	FB
Language	SCL	**Numbering**	Automatic		
Information					
Title		**Author**		**Comment**	
Family		**Version**	0.1	**User-defined ID**	

FB_manual			
Name	**Data type**	**Default value**	**Retain**
▼ Input			
limit1	DInt	0	Non-retain
limit2	DInt	0	Non-retain
limit3	DInt	0	Non-retain
Output			
InOut			
Static			
▼ Temp			
limit_3	DInt		
Temp_1	DInt		
limit_1	DInt		
limit_2	DInt		
Constant			

```
0001 IF "TS" THEN //Total stop
0002   "do_osa_x_protismer" := FALSE;
0003   "do_osa_x_smer" := FALSE;
0004   "do_osa_y_vpred" := FALSE;
0005   "do_osa_y_vzad" := FALSE;
0006   "do_osa_z_dolu" := FALSE;
0007   "do_osa_z_nahoru" := FALSE;
0008
0009 ELSE
0010   "do_osa_x_protismer" := "osa_x_protismer"; //anti-clockwise rotation
0011   "do_osa_x_smer" := "osa_x_smer"; //clockwise rotation
0012   "do_osa_y_vpred" := "osa_y_vpred"; //extend the arm
0013   "do_osa_y_vzad" := "osa_y_vzad"; //retract the arm
0014   "do_osa_z_dolu" := "osa_z_dolu"; //go down
0015   "do_osa_z_nahoru" := "osa_z_nahoru"; //go up
0016   "do_vacuum" := "vacuum"; //suction
0017   "pom_do_0_6" := "kompresor"; //compressor
0018
0019   //Axis limits
0020   IF "di_snimac_osa_x" THEN
0021     "do_osa_x_smer" := FALSE;
0022   END_IF;
0023   IF "High_Speed_Counter_3".CountValue <= #limit3 THEN
0024     "do_osa_x_smer" := FALSE;
0025   END_IF;
0026
0027   IF "di_snimac_osa_z" THEN
```

```
0028      "do_osa_z_nahoru" := FALSE;
0029    END_IF;
0030    IF "High_Speed_Counter_1".CountValue >= #limit1 THEN
0031      "do_osa_z_dolu" := FALSE;
0032    END_IF;
0033
0034    IF "di_snimac_osa_y" THEN
0035      "do_osa_y_vzad" := FALSE;
0036    END_IF;
0037    IF "High_Speed_Counter_2".CountValue >= #limit2 THEN
0038      "do_osa_y_vpred" := FALSE;
0039    END_IF;
0040  END_IF;
```

j. FB_posun – modified

Robot_pick_&_place / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

FB_posun [FB4]

FB_posun Properties					
General					
Name	FB_posun	**Number**	4	**Type**	FB
Language	SCL	**Numbering**	Automatic		
Information					
Title		**Author**		**Comment**	
Family		**Version**	0.1	**User-defined ID**	

FB_posun

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
Input									
Exec	Bool	false	Non-retain	True	True	True	False		
Z_Up	DInt	3000	Non-retain	True	True	True	False		
Z_Down	DInt	3758	Non-retain	True	True	True	False		
X	DInt	923	Non-retain	True	True	True	False		
C	DInt	-2419	Non-retain	True	True	True	False		
Output									
Done	Bool	false	Non-retain	True	True	True	False		
State	Int	0	Non-retain	True	True	True	False		
InOut									
Static									
Temp									
Constant									

```
0001 //State machine for navigation to the required location
0002
0003 CASE #State OF
0004   0: //default (waiting to be initiated)
0005     IF NOT #Exec THEN //default state of Done is FASLE! (otherwie FB_presun
     starts simultaneously...)
0006       #Done := FALSE;
0007     END_IF;
0008
0009     IF #Exec THEN
0010       #State := 30;
0011     END_IF;
0012
0013   30: //sending instructions to FB_goto
```

```
0014      "JedNa1" := #Z_Up; // Z
0015      "JedNa2" := #X; //pos_x
0016      "JedNa3" := #C; //pos_c
0017      "JedNaExec1" := TRUE;
0018      "JedNaExec2" := TRUE;
0019      "JedNaExec3" := TRUE;
0020      #State := 40;
0021
0022    40: //waiting till FB_goto is done
0023      IF "FB_goto_1".Done AND "FB_goto_2".Done AND "FB_goto_3".Done OR "ResetEx-
      ec" THEN //ResetExec -> manual reset (if FB_goto can't be completed)
0024        "JedNaExec1" := FALSE;
0025        "JedNaExec2" := FALSE;
0026        "JedNaExec3" := FALSE;
0027        #State := 70;
0028      END_IF;
0029
0030    70: //operation completed -> return to default
0031      #Done := TRUE;
0032      #State := 0;
0033  END_CASE;
0034
```

Symbol	Address	Type	Comment
"FB_goto_1".Done		Bool	
"FB_goto_2".Done		Bool	
"FB_goto_3".Done		Bool	
"JedNa1"	%MD20	DInt	
"JedNa2"	%MD24	DInt	
"JedNa3"	%MD28	DInt	
"JedNaExec1"	%M4.4	Bool	
"JedNaExec2"	%M4.5	Bool	
"JedNaExec3"	%M4.6	Bool	
"ResetExec"	%M0.4	Bool	
#C		DInt	
#Done		Bool	
#Exec		Bool	
#State		Int	
#X		DInt	
#Z_Up		DInt	

k. FB_presun – modified

Robot_pick_&_place / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

FB_presun [FB3]

FB_presun Properties					
General					
Name	FB_presun	**Number**	3	**Type**	FB
Language	SCL	**Numbering**	Automatic		
Information					
Title		**Author**		**Comment**	
Family		**Version**	0.1	**User-defined ID**	

FB_presun

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engineering	Set-point	Supervision	Comment
▼ Input									
Exec	Bool	false	Non-retain	True	True	True	False		
Z_Up	DInt	3000	Non-retain	True	True	True	False		
Z_Down	DInt	3758	Non-retain	True	True	True	False		
X	DInt	923	Non-retain	True	True	True	False		
C	DInt	-2419	Non-retain	True	True	True	False		
fb_posun	Block_FB	FB 0	Non-retain	True	True	True	False		
▼ Output									
Done	Bool	false	Non-retain	True	True	True	False		
State	Int	0	Non-retain	True	True	True	False		
InOut									
Static									
Temp									
Constant									

```
0001 //State machine for relocation of object to required location
0002
0003 CASE #State OF
0004   0:  //default
0005     #Done := FALSE;
0006     IF #Exec THEN
0007       #State := 2;
0008     END_IF;
0009
0010   2: //go DOWN
0011     "pom_do_0_6" := TRUE;
0012     "JedNa1" := #Z_Down;
```

```
0013      "JedNaExec1" := TRUE;
0014      #State := 4;
0015
0016    4: //are we down?
0017      IF "FB_goto_1".Done THEN
0018        "JedNaExec1" := FALSE;
0019        #State := 10;
0020      END_IF;
0021
0022    10: //suction & compressor ON
0023      "do_vacuum" := TRUE;
0024      #State := 20;
0025
0026    20: //bring the part UP
0027      "JedNa1" := #Z_Up;
0028      "JedNaExec1" := TRUE;
0029      #State := 30;
0030
0031    30: //are we up?
0032      IF "FB_goto_1".Done THEN
0033        "JedNaExec1" := FALSE;
0034        #State := 40;
0035      END_IF;
0036
0037    40: // movement along XC
0038      "JedNa2" := #X;
0039      "JedNa3" := #C;
0040      "JedNaExec2" := TRUE;
0041      "JedNaExec3" := TRUE;
0042      #State := 50;
0043
0044    50: //required location reached?
0045      IF "FB_goto_2".Done AND "FB_goto_3".Done THEN
0046        "JedNaExec2" := FALSE;
0047        "JedNaExec3" := FALSE;
0048        #State := 60;
0049      END_IF;
0050
0051    60: //lowereing the part
0052      "JedNa1" := #Z_Down;
0053      "JedNaExec1" := TRUE;
0054      #State := 70;
0055
0056    70: //is it down?
0057      IF "FB_goto_1".Done THEN
0058        "JedNaExec1" := FALSE;
0059        #State := 80;
0060      END_IF;
0061
0062    80: //suction off
0063      "do_vacuum" := FALSE;
0064      "pom_do_0_6" := FALSE;
0065      "IEC_Timer_0_DB".TON(IN := TRUE,
0066                           PT := t#1s);
0067      //wait
0068      IF "IEC_Timer_0_DB".Q THEN
0069        #State := 90;
0070      END_IF;
0071
```

```
0072    90: //go UP
0073      "IEC_Timer_0_DB".TON(IN := FALSE,
0074                           PT := t#1s);
0075      "JedNa1" := #Z_Up;
0076      "JedNaExec1" := TRUE;
0077      #State := 100;
0078
0079    100: //are we up?
0080      IF "FB_goto_1".Done THEN
0081        "JedNaExec1" := FALSE;
0082        #State := 110;
0083      END_IF;
0084
0085    110: //operation done -> default
0086      #Done := TRUE;
0087      IF NOT #Exec THEN
0088        #State := 0;
0089      END_IF;
0090  END_CASE;
0091
```

Symbol	Address	Type	Comment
"do_vacuum"	%Q0.7	Bool	
"FB_goto_1".Done		Bool	
"FB_goto_2".Done		Bool	
"FB_goto_3".Done		Bool	
"IEC_Timer_0_DB".Q		Bool	
"JedNa1"	%MD20	DInt	
"JedNa2"	%MD24	DInt	
"JedNa3"	%MD28	DInt	
"JedNaExec1"	%M4.4	Bool	
"JedNaExec2"	%M4.5	Bool	
"JedNaExec3"	%M4.6	Bool	
"pom_do_0_6"	%M2.6	Bool	
#C		DInt	
#Done		Bool	
#Exec		Bool	
#State		Int	
#X		DInt	
#Z_Down		DInt	
#Z_Up		DInt	

l. FB_goto – modified

Robot_pick_&_place / PLC_1 [CPU 1516-3 PN/DP] / Program blocks

FB_goto [FB2]

FB_goto Properties					
General					
Name	FB_goto	**Number**	2	**Type**	FB
Language	SCL	**Numbering**	Automatic		
Information					
Title		**Author**		**Comment**	
Family		**Version**	0.1	**User-defined ID**	

FB_goto			
Name	**Data type**	**Default value**	**Retain**
▼ Input			
Execute	Bool	false	Non-retain
GoTo	DInt	0	Non-retain
Actual	DInt	0	Non-retain
Limit	DInt	10	Non-retain
EndMinus	DInt	0	Non-retain
EndPlus	DInt	0	Non-retain
▼ Output			
Mplus	Bool	false	Non-retain
Mminus	Bool	false	Non-retain
Done	Bool	false	Non-retain
State	Int	0	Non-retain
InOut			
Static			
▼ Temp			
Diff	DInt		
Constant			

```
0001 IF "TS" THEN //Total stop
0002   #Mminus := FALSE;
0003   #Mplus := FALSE;
0004   #State := 0;
0005 ELSE
0006   IF #GoTo < #EndPlus AND #GoTo > #EndMinus THEN          // in soft limits?
0007
0008   #Diff := #GoTo - #Actual; //calculating distance
0009   CASE #State OF
0010     0:                              //INIT
0011       #Mplus := FALSE;
0012       #Mminus := FALSE;
0013       #Done := FALSE;
0014       IF #Execute THEN
0015         #State := 10;
0016       END_IF;
0017     10:                                        //MOVE
0018       IF NOT #Execute THEN
0019         #State := 0;
0020       END_IF;
0021       IF #Diff > #Limit THEN                    //plus direction
0022         #Mplus := TRUE;
```

```
0023          #Mminus := FALSE;
0024        ELSIF #Diff < - #Limit THEN              //minus direction
0025          #Mplus := FALSE;
0026          #Mminus := TRUE;
0027        ELSE                                    //no need to move -> done
0028          #State := 20;
0029        END_IF;
0030      20:                                      //DONE
0031        #Mplus := FALSE;
0032        #Mminus := FALSE;
0033        #Done := TRUE;
0034        IF NOT #Execute THEN
0035          #State := 0;
0036        END_IF;
0037    END_CASE;
0038
0039  ELSE    //don't move if not in soft limits
0040    #Mplus := FALSE;
0041    #Mminus := FALSE;
0042  END_IF;
0043
0044  END_IF;
```