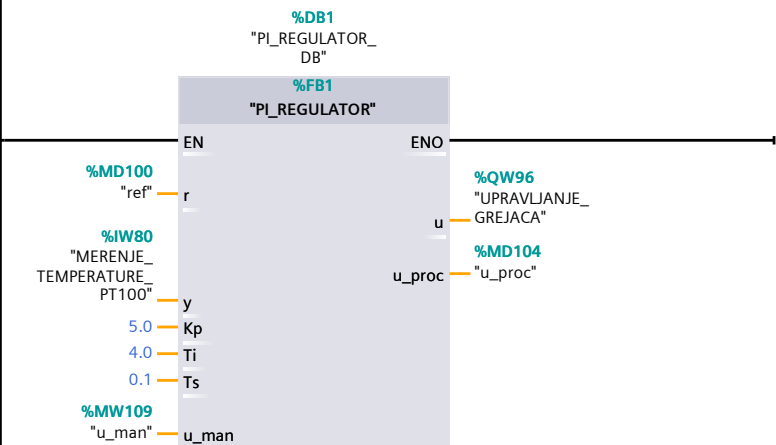


Cyclic interrupt [OB30]

Cyclic interrupt Properties							
General							
Name	Cyclic interrupt	Number	30	Type	OB	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Event_Count	Int		Events discarded
Temp			
Constant			

Network 1:



Totally Integrated Automation Portal

Main [OB1]

Main Properties

General

Name	Main	Number	1	Type	OB	Language	LAD
Numbering	Automatic						

Information

Title	"Main Program Sweep (Cycle)"	Author		Comment		Family	
Version	0.1	User-defined ID					

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			

Totally Integrated Automation Portal

Startup [OB100]

Startup Properties

General

Name	Startup	Number	100	Type	OB	Language	LAD
Numbering	Automatic						

Information

Title	"Complete Restart"	Author		Comment		Family	
Version	0.1	User-defined ID					

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			

Network 1:

%M108.1
"auto_previous"

(R)

%M108.0
"auto_current"

(R)

Totally Integrated Automation Portal

PI_REGULATOR [FB1]

PI_REGULATOR Properties

General

Name	PI_REGULATOR	Number	1	Type	FB	Language	SCL
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA	Writ-able from HMI/OPC UA	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment
▼ Input									
r	Real	0.0	Retain	True	True	True	False		
y	Word	16#0	Retain	True	True	True	False		
Kp	Real	0.0	Retain	True	True	True	False		
Ti	Real	0.0	Retain	True	True	True	False		
Ts	Real	0.0	Retain	True	True	True	False		
u_man	Word	16#0	Retain	True	True	True	False		
▼ Output									
u	Word	16#0	Non-retain	True	True	True	False		
u_proc	Real	0.0	Non-retain	True	True	True	False		
InOut									
▼ Static									
e	Real	0.0	Non-retain	True	True	True	False		
up	Real	0.0	Non-retain	True	True	True	False		
ui	Real	0.0	Non-retain	True	True	True	False		
u_PI	Real	0.0	Non-retain	True	True	True	False		
y_float	Real	0.0	Non-retain	True	True	True	False		
u_man_float	Real	0.0	Non-retain	True	True	True	False		
Temp									
Constant									

```
0001 // PI regulator - P (Proportional Gain), I (Integral Gain)
0002
0003 // Na P (Proportional Gain) utice parametar Kp.
0004 // Sto je vece Kp, to sistem brze reaguje na gresku (razliku izmedju trenutne izmerene temperature i zeljene, tj.
0005 // reference -> e = r - y_float), tj. agresivnije ce da menja temperaturu.
0006 // Tada sistem brze postize zeljenu temperaturu, ali postoji rizik od "overshoot-a", tj da se zeljena temperatura
0007 // premasi, pa da je potrebno da se postepeno vrati temperatura na zeljenu, kada sistem ulazi u stacionarno stanje
0008 // kada temperatura oscilira oko zeljene.
0009 // Takodje, sto je vece Kp, to je veca verovatnoca da budu manje oscilacije nakon sto sistem udje u stacionarno
0010 // stanje, tj. veca je preciznost u stacionarnom stanju.
0011
0012 // Na I (Integral Gain) utice parametar Ti.
0013 // Sto je vece Ti, to sistem blaze i postepenije ispravlja akumulirane greske iz proslosti kako bi se postiglo
0014 // sto preciznije stacionarno stanje, tj. da se odrzava temperatura sto blize zeljenoj.
0015
0016 #y_float := #y / 10.0;
0017 #u_man_float := #u_man / 27700.0 * 100.0;
0018
0019 IF "auto_current" THEN
0020     // Obezbedjivanje Bumpless prelaza izmedju rucnog i automatskog rezima rada.
0021     IF NOT "auto_previous" THEN
0022         #r := #y_float;
0023         #ui := #u_man_float;
0024     END_IF;
0025     #e := #r - #y_float;
0026     #up := #Kp * #e;
0027     #ui := #ui + #Kp / #Ti * #Ts * #e;
0028     #u_PI := #up + #ui;
0029     // Obezbedjivanje Antiwindup mehanizma u okviru automatskog rezima rada.
0030     IF #ui > 100.0 THEN
0031         #ui := 100.0;
0032     END_IF;
0033     IF #ui < 0.0 THEN
0034         #ui := 0.0;
0035     END_IF;
0036     IF #u_PI > 100.0 THEN
0037         #u_PI := 100.0;
0038     END_IF;
0039     IF #u_PI < 0.0 THEN
0040         #u_PI := 0.0;
0041     END_IF;
0042     #u_proc := #u_PI;
0043     #u := REAL_TO_INT(#u_PI / 100.0 * 27700.0);
0044     "auto_previous" := true;
0045 ELSE
0046     IF #u_man > 27700 THEN
0047         #u := 27700;
```

Totally Integrated Automation Portal

```
0048 ELSE
0049     IF #u_man < 0 THEN
0050         #u := 0;
0051     ELSE
0052         #u := #u_man;
0053     END_IF;
0054 END_IF;
0055 // Obezbedjivanje Bumpless prelaza izmedju rucnog i automatskog rezima rada.
0056 #r := #y_float;
0057 "auto_previous" := false;
0058 END_IF;
```

Symbol	Address	Type	Comment
"auto_current"	%M108.0	Bool	
"auto_previous"	%M108.1	Bool	
#e		Real	
#Kp		Real	
#r		Real	
#Ti		Real	
#Ts		Real	
#u		Word	
#u_man		Word	
#u_man_float		Real	
#u_Pl		Real	
#u_proc		Real	
#ui		Real	
#up		Real	
#y		Word	
#y_float		Real	