ტექნიკური უნივერსიტეტი

ჯ. გრიგალაშვილი



დაპროგრამება TIA Portal-ში

S7-1200/S7-1500

დამტკიცებულია სახელმძღვანელოდ სტუ-ს სარედაქციო-საგამომცემლო საბჭოს მიერ, ოქმი №

თბილისი 2021

უაკ 681.5

TIA Portal (Totally Integrated Automation Portal) - ტექნოლოგიური პროცესების მართვის ავტომატიზირებული სისტემების პროგრამული უზრუნველყოფის დამუშავების უახლესი ინტეგრირებული გარემოა კონტროლერებისა და ამმრავებისაგან დაწყებული, დამთავრებული ადამიანი-მანქანური ინტერფეისებით. იგი წარმოადგენს თანამედროვე კომპლექსური ავტომატიზაციის კონცეფციის ხორცშესხმას და Siemens AG კომპანიის ავტომატიზაციის სისტემების ოჯახის Simatic-ის ევოლუციურ განვითარებას.

ამ ნაშრომში განხილულია STEP 7 V13, რომელიც დაფუმნებულია TIA Portal-ის დაპროექტების ერთიანი სამუშაო გარემოს ფუნქციონალურ შესამლებლობებზე და აქვს ინტუიციურად გასაგები, ეფექტიური და მომავალზე ორიენტირებული, Simatic პროგრამირებად კონტროლერების ბაზაზე დაფუმნებული ავტომატიზაციის პროექტების დამუშავების პროგრამული უზრუნველყოფა. იგი წარმოდგენილია ორი ვერსიის სახით:

- SIMATIC STEP 7 Professional V13 რომელიც არის SIMATIC-ის S7-1200, S7-300, S7-400, WinAC პროგრამირებადი კონტროლერების დაპროექტების სისტემა. იგი ასევე ოპტიმალურად მხარს უჭერს აგრეთვე SIMATIC-ის ახალი სერიის S7-1500-ის ლოგიკურ პროგრამულ კონტროლერებს;
- SIMATIC STEP 7 Basic V13 რომელიც არის საკუთრივ SIMATIC-ის S7-1200 პროგრამირებადი კონტროლერებისათვის განკუთვნილი დაპროექტების სისტემა.

ნაშრომში გამოყენებულია შინაარსის გადმოცემის ორიგინალური მიდგომა. კონკრეტულად, აღებულია Siemens AG-ს ახალი თაობის კონტროლერებიდან ორი უახლესი მოდელი S7-1200/ S7-1500 ხაზგასმულია მათი უპირატესობანი ძველ მოდელებთან S7-300/S7-400 შედარებით. მოცემულია მათი გამოყენების რეკომენდაციები, მოყვანილია მაგალითები და მითითებულია ბმულები წარმოქმნილ შესაძლო კითხეებზე პასუხის გაცემისათვის.

ნაშრომში დიდი ადგილი ეთმობა S7-1200 პროგრამული ლოგიკური კონტროლერების პრაქტიკული გამოყენების საკითხებს, კერმოთ განხილულია: კონტროლერის Ethernet შეერთება კომპიუტერთან, კონტროლერის დაკავშირება პროგრამასთან, IP მისამართის დადგენა, ახალი პროექტის გახსნა, პროექტში ახალი მოწყობილობის დამატება, მოწყობილობის გამართვა, პროგრამის შექმნა ორი რიცხვის გამრავლების მაგალითზე, პროგრამის კომპილიაცია და კონტროლერში ჩატვირთვა, პროგრამის ტესტირებები.

სახელმძღვანელო განკუთვნილია ინფორმატიკისა და მართვის სისტემების ფაკულტეტის მაგისტრატურისა და დოქტორანტურის საფეხურის სტუდენტებისათვის.

რეცენზენტი მართვის სისტემების დეპარტამენტის ხელმძღვანელი პროფესორი ქეთევან კოტრიკაძე რეცენზენტი კომპიუტერული ინჟინერიის დეპარტამენტის პროფესორი ია მოსაშვილი

საგამომცემლო სახლი "ტექნიკური უნივერსიტეტი", 2021

სარჩევი

1	შესავ	ალი	7			
2	ინოვაციები S7-1200/S7-1500					
	2.1	შესავალი	8			
	2.2	ტერმინები	8			
	2.3	დაპროგრამების ენები	10			
	2.4.	ოპტიმიზირებული მანქანური კოდი	11			
	2.5.	ბლოკების შექმნა	12			
	2.6.	ოპტიმიზირებული ბლოკები	13			
	2.6.1	S7-1200: ოპტიმიზირებული ბლოკი	13			
	2.6.2	S7-1500: ოპტიმიზირებული ბლოკი	14			
	2.6.3	მონაცემთა შენახვის ყველაზე უკეთესი ვარიანტი S7-1500-ში	15			
	2.6.4	გარდაქმნა ოპტიმიზირებულ და არაოპტიმიზირებულ ტეგებს შორის	19			
	2.6.5	პარამეტრების გადაცემა ბლოკებს შორის ოპტიმიზირებული და				
		სტანდარტული ხელწვდომის ტიპებით	19			
	2.6.6	კომუნიკაცია ოპტიმიზირებულ მონაცემებით	20			
	2.7	ბლოკის თვისებები	21			
	2.7.1	ბლოკის ზომა	21			
	2.7.2	მაორგანიებელი ბლოკების რაოდენობა (OB)	21			
	2.8.	მონაცემთა ახალი ტიპები S7-1200/S7-1500-ში	22			
	2.8.1	მონაცემთა ელემენტარული ტიპები	23			
	2.8.2	მონაცემთა ტიპი Data_Time_Long	24			
	2.8.3	დამხმარე მონაცემთა ტიპები დროისათვის	24			
	2.8.4	მონაცემთა ტიპები უნიკოდთან მუშაობისათვის	24			
	2.8.5	მონაცემთა ტიპი VARIANT (S7-1500 და S7-1200 FW4.1-ით)	25			
	2.9	ინსტრუქციები	29			
	2.9.1	CALCULATE	29			
	2.9.2	ინსტრუქციები MOVE	29			
	2.9.3	ინსტრუქციები VARIANT-ით (S7-1500 და S7-1200 FW4.1-ით)	32			
	2.9.4	RUNTIME	32			
	2.10	სიმბოლიკა და კომენტარები	33			
	2.10.1	პროგრამის რედაქტორი	33			
	2.10.2	კომენტარები დაკვირვების ცხრილში	34			
	2.11	სისტემური კონსტანტები	35			
	2.12	სამომხმარებლო კონსტანტები	37			
	2.13	კონტროლერის ტეგებისა და HMI ტეგებისთვის შიგა ბმულების ID	38			
	2.14	რეჟიმი STOP შეცდომების წარმოშობის შემთხვევაში	40			
3	შესავა	ალი დაპროგრამებაში	41			
	3.1	ოპერაციული სისტემა და სამომცხმარებლო პროგრამა	41			
	3.1.1	ოპერაციული სისტემა და სამომხმარებლო პროგრამა	41			
	3.2	პროგრამული ბლოკები	42			

3.2.1	მაორგანიზებელი ბლოკები (OB)	43
3.2.2	ფუნქციები (FC)	44
3.2.3	ფუნქციონალური ბლოკები (FB)	. 47
3.2.4	ეკზემპლიარები	48
3.2.5	მულტიეკზემპლიარები	. 49
3.2.6	მონაცემთა გლობალური ბლოკები (DB)	51
3.2.7	ჩატვირთვა განმეორებითი ინციალიზაციის გარეშე	52
3.2.8	ბლოკების განმეორებითი გამოყენების შესაძლებლობა	55
3.2.9	ბლოკის ნომერის ავტომატური მიკუთვნება	. 57
3.3	ბლოკის ინტერფეისის ტიპები	58
3.3.1	ფაქტიური მნიშვნელობის მიკუთვნება შემავალ პარამეტრზე	58
3.3.2	ფაქტიური პარამეტრის მიკუთვნება გამავალ პარამეტრზე	58
3.3.3	პარამეტრის გადაცემის ვარიანტები	59
3.4	შენახვის კონცეფცია	59
3.4.1	ბლოკების ინტერფეისები მონაცემთა გაცვლისთვის	59
3.4.2	მეხსიერების გლობალური არე	60
3.4.3	მეხსიერების ლოკალური არე	61
3.4.4	ხელწვდომის სიჩქარე მეხსიერების არეებთან	62
3.5	შენახვითობა	63
3.6	სიმბოლური დამისამართება	66
3.6.1	სიმბოლური დამისამართება ნაცვლად აბსოლუტ. დამისამართებისა	66
3.6.2	ARRAY-ს მონაცემთა ტიპი და ირიბი ხელწვდომა ელემენტებზე	68
3.6.3	STRUCT დs PLC data type	. 70
3.6.4	შეყვანა/გამოყვანის არეებთან ხელწვდომა PLC data type-ის დახმარებ	. 73
3.6.5	შერჩევითი ხელწვდომა	74
3.7	ბიბლიოთეკები	. 75
3.7.1	ბიბლიოთეკების ტიპები და ბიბლიოთეკების ელემენტები	76
3.7.2	ტიპობრივი კონცრფცია	77
3.7.3	განსხვავებები ტიპიზირებულ ობიექტებს შორის SPU-სა და HMI-თვის	78
3.7.4	ბლოკის ვერსიის შექმნა	. 78
3.8	მწარმოებლობის გაზრდა აპარატული წყვეტების დახმარებით	. 82
3.9	დამატებითი რეკომენდაციები მწარმოებლობის გაზრდისთვის	84
3.10	დაპროგრამების ენა SCL: რჩევები და რეკომენდაციები	. 85
3.10.1	გამოძახების შაბლონის გამოყენება	85
3.10.2	ინსტრუქციების რომელი პარამეტრები არიან აუცილებელნი	. 87
3.10.3	ცვლადების სახელების გადატანა	. 87
3.10.4	FOR, REPEAT da WHILE ციკლების გამოყენება	87
3.10.5	CASE ინსტრუქციის გამოყენება	89
3.10.6	მთვლელის ყოფაქცევა ციკლისათვის FOR	. 89
3.10.7	FOR ციკლი უკუ მიმართულებით	90
3.10.8	ეკზემპლიარების მარტივი შექმნა გამოძახებისათვის	. 90

	3.10.9 ცვლადების დამუშავება Time (დრო) მონაცემთა ტიპით	90
4	აპარატულად დამოუკიდებელი დაპროგრამება	
	4.1 S7-300/S7-400 და S7-1200/S7-1500 მონაცემთა ტიპები	91
	4.2 გადასვლა მერკერებიდან გლობალურ მონაცემთა ბლოკებზე	
	4.3 სინქრობაიტის დაპროგრამება	
5	STEP 7 Safety TIA Portal-do	
	5.1 შესავალი	95
	5.2 ტერმინები	
	5.3 უსაფრთხოების პროგრამის კომპონენტები	
	5.4 ჯგუფი F runtime	
	5.5 F ხელმოწერა	
	5.6 მისამართის საწყისი PROFIsafe F-I/O-ზე	100
	5.7 F პერიფერიის შეფასება	101
	5.8 მნიშვნელობათა მდგომარეობები (S7-1200/S7-1500)	101
	5.9 მონაცემთა ტიპები	102
	5.10 PLC data type-ის შაბლონები F პროგრამისთვის	103
	5.11 TRU/FALSE	104
	5.12 ოპტიმიზირებული კომპილიაცია და შესრულების რეჟიმი	105
	5.12.1 უარის თქმა დროზე გავლენის მქონე ბლოკების გამოყენებაზე	106
	5.12.2 უარის თქმა ჩალაგებულ გამოძახებებზე	106
	5.12.3 სტანდარტული და უსაფრთხოების პროგრამების განცალება	107
	5.12.4 მულტიეკზემპლიარების გამოყენება	107
	5.12.5 უარის თქმა JMP/lable ინსტრუქციის გამოყენებაზე	107
	5.13 მონაცემთა გაცვლა სტანდარტულ და F პროგრამას შორის	108
	5.14 უსაფრთხოების პროგრამის ტესტირება	109
	5.15 STOP რეჯიმი F-შეცდომის წარმოშობისას	109
	5.16 უსაფრთხოების პროგრამის მიგრაცია	110
	5.17 ძირითადი რეკომენდაციები უსაფრთხოებისათვის	110
	5. 18 ყველაზე უფრო მეტად მნიშვნელოვანი რეკომენდაციები	110
6	პრაქტიკული სამუშაოები	112
	6.1_პრაქტიკული სამუშაო 1	112
	6.1.1 S7-1200 პლკ-ს პრაქტიკული გამოყენების საკითხები	112
	6.1.2 TIA Portal-ის ხელმისაწვდომი ფუნქციები	112
	6.1.3 ქსელური შეერთების გამართვა	116
	6.1.4 TIA Portal-ის ჩამოტვირთვა	120
	6.1.5 TIA Portal-ის ინსტალაცია	124
	6.2 პრაქტიკული სამუშაო 2	136
	6.2.1 TIA Portal-ის გაშვება	137
	6.2.2 IP მისამართის მინიჭება	138
	6.2.3 ახალი მოწყობილობის დამატება	142
	6.2.4 S7-1200-ის გამართვა და შორიდან მიერთება	144

6.2.5 პროგრამის შექმნა	
6.2.6 პროგრამის კომპილიაცია და ჩატვირთვა კონტროლერში	158
6.2.7 კონტროლერის ტესტირება	162
6.2.8 პროგრამის მუშაობის ვიზუალიზაცია	163
6.2.9 პროგრამის ბიჯობრივი გაწყობა	164
6.2.10 პროგრამის ვიზუალური ტესტირება	170
6.2.11 პროგრამის მუშაობა ონლაინ რეჟიმში	172
6.3. პრაქტიკული სამუშაო 3	174
6.4 გამოყენებული ლიტერატურა	

1 შესავალი

SIMATIC ახალი თაობის კონტროლერების დამუშავების მიზნები

- ყველა კომპონენტების დამუშავების ერთიან გარემოში მუშაობა (კონტროლერები, HMI მოწყობილობები, ამძრავები და სხვ.);
- ერთსახა დაპროგრამება;
- გაზრდილი მწარმოებლობა;
- ბრძანებათა სრული ნაკრები თვითოეული ენისათვის;
- მთელი პროგრამის წარმოდგენა სიმბოლური სახელწოდებებით;
- მონაცემთა დამუშავება მაჩვენებლის გამოყენების გარეშე;
- შექმნილი ბლოკების განმეორებით გამოყენების შესაძლებლობა.

სახელმძღვანელოს მიზანი

განახლებული იქნა SIMATIC-ის ახალი თაობის S7-1200 და S7-1500 კონტროლერების სისტემური არქიტექტურა და TIA Portal-ის გამოყენების შემთხვევაში აღნიშნული სიახლეები იძლევა უპირატესობას კონტროლერების დაპროგრამებისა და კონფიგურირების პროცესებში.

ამ დოკუმენტში აღწერილია რეკომენდაციები და რჩევები S7-1200 და S7-1500 კონტროლერების ეფექტური დაპროგრამებისათვის. ზოგიერთი სხვაობა S7-300 და S7-400 კონტროლერების სისტემურ არქიტექტურასთან, აგრეთვე ახალი შესაძლებლობები დაპროგრამების პროცესში აღწერილია მარტივ და გასაგებ ენაზე. ეს დაგვეხმარება შექმნათ ავტომატიზაციის ამოცანების სტანდარტული და ოპტიმალური გადაწყვეტები.

მოყვანილი მაგალითები შეიძლება გამოყენებულ იქნას როგორც S7-1200 ისე S7-1500 კონტროლერებისთვის.

ამ სახელმძღვანელოს საკვანძო მომენტები

ამ დოკუნმენტში აღწერილია TIA Portal-ის შემდეგი შესაძლებლობები:

- S7-1200/S7-1500-ის სიახლეები:
 - დაპროგრამების ენა;
 - ოპტიმიზირებული ბლოკები;
 - მონაცემთა ტიპები და ინსტრუქციები.
- რეკომენდაციები დაპროგრამებაში:
 - ოპერაციული სისტემა და სამომხმარებლო პროგრამა;
 - მონაცემთა შენახვის პრინციპი;
 - სიმბოლური დამისამართება;
 - ბიბლიოთეკები.
- რეკომენდაციები აპარატულ-დამოუკიდებელ დაპროგრამებაზე;
- რეკომენდაციები STEP 7 Safety-თვის TIA Portal-ში;
- ყველაზე მეტად მნიშვნელოვანი რეკომენდაციების მიმოხილვა.

უპირატესობანი და ახალი შესაძლებლობები

უპირატესობათა უმეტესობა მიიღწევა შემდეგი რეკომენდაციების გამოყენებით:

- მომხმარებლის მძლავრი პროგრამა;
- გასაგები პროგრამული სტრუქტურები;
- ინტუიციური და ეფექტიური პროგრამული გადაწყვეტილებები.

დამატებითი ინფორმაცია

SIMATIC კონტროლერების დაპროგრამების დროს, პროგრამისტის ამოცანაა შექმნას გასაგები და წაკითხვადი სამომხმარებლო პროგრამა. თვითოეული მომხმარებელი იყენებს თავის სტრატეგიას თუ როგორ შექმნას და დაასახელოს ტეგები, ბლოკები ანდა კომენტარები. სხვადასხვა მიდგომა სამომხმარებლო პროგრამის შექმნისათვის, რომელიც გასაგები იქნება მხოლოდ თვით პროგრამისტისთვის.

სახელმძღვანელო დაპროგრამებაში შესაძლებლობას გაძლევთ გამოიყენოთ განსაზღვრული სტილი პროგრამების შედგენის დროს. ასეთი მექანიზმი, მაგალითად აღწერს ტეგების დანიშნულებას და ბლოკების სახელწოდებებს, რაც ამარტივებს დამუშავების ამოცანას მაგალითად SCL ენაზე დაპროგრამებისას. თქვენ შეგიძლიათ გამოიყენოთ აღნიშნული წესები და რეკომენდაციები მომავალში: ისინი იძლევა (დაპროგრამების არა სტანდატი) კონსისტენტური დაპროგრამების წინაპირობას.

შენიშვნა

თქვენ შეგიძლიათ მოძებნოთ S7-1200 და S7-1500 კონტროლერების დაპროგრამების სახელმძღვანელო ბმულით: <u>https://support.industry.siemens.com/cs/ww/en/view/81318674</u>

2 ინოვაციები S7-1200/S7-1500-ში

2.1 შესავალი

მირითადათ, SIMATIC S7-1200/1500 კონტროლერების დაპროგრამება დარჩა ისეთივე, როგორც S7-300/400 კონტროლერების. არსებობს ერთნაირი დაპროგრამების ენები, როგორიცაა LAD, FBD, STL, SCL ანდა GRAPH. ისეთივე ბლოკების ტიპები, მაგალითად მაორგანიზებელი ბლოკები (OB), ფუნქციონალური ბლოკები (FB), ფუნქციები (FC) ანდა მონაცემთა ბლოკები (DB). ანუ S7-300/400-ისთვის უკვე შექმნილი პროგრამები შეიძლება გამოყენებულ იქნას S7-1500-ისთვსაც, ხოლო პროგრამები შექმნილი LAD, FBD, SCL ენებზე შეიძლება გამოყენებულ იქნას S7-1200ზეც. ამის გარდა, არსებობს კიდევ მრავალი სიახლეები, რომლებიც გაამარტივებენ დაპროგრამებას და რომლებსაც შეუძლიათ შექმნან ძლიერი და ეკონომიური კოდი, გამოყენებული მეხსიერების თვალსაზრისით.

ჩვენ არა მხოლოდ ვიძლევით რეკომენდაციას შვექმნათ პროგრამები, რომლებიც არა მხოლოდ გამოდგება ერთი-ერთზე S7-1200/1500 კონტროლერებზე, არამედ გამოვიყენოთ ახალი შესაძლებლობები და სადაც კი ეს შესაძლებელია, გამოვიყენოთ ისინი. ამით ჩვენ მიიღებთ პროგრამულ კოდს, რომელიც, მაგალითად იქნება:

- ოპტიმალური მეხსიერების გამოყენებაში CPU-სათვის;
- მარტივი გამოყენებისათვის;
- მარტივი პრეოგრამის შემდგომი ექსპლუატაციის თვალსაზრისით.

შენიშვნა

ინფორმაცია S7-300/400 მიგრაციისთვის S7-1500-ში ხელმისაწვდომია შემდეგი ბმულით:

https://support.industry.siemens.com/cs/ww/en/view/109478811

2.2. ტერმინები

ძირითადი ტერმინები TIA Portal-ში

ზოგიერთი ტერმინი შეცვლილი იქნა TIA Portal-ში მოხერხებული მუშაობისთვის (იხ. სურათი 2-1:).



ტერმინები ტეგებისა და პარამეტრებისათვის

ტეგებთან, ფუნქციებთან და ფუნქციონალურ ბლოკებთან მუშაობის დროს მათი უმეტესობის გამოყენება ხდება არასწორად ანდა არაკორექტულად. შემდეგ გამოსახულებაზე ნაჩვენებია აღნიშნული განსხვავება. (იხ. სურათი 2-2:, ცხრილი 2-1:)



სურათი 2-2: ტერმინი ტეგებისა და პარამეტრებისათვის

	ტერმინები	აღწერები
1.	ტეგი	ტეგების ასახვა ხდება სახელ/იდენტიფიკატორით და დანიშნული მისამართი კონტროლერის მეხსიერებაში. (ტეგები ყოველთვის განისაზღვრება მონაცემთა ტიპოებით (Bool, Integer და ა.შ.შ. - PLC ტეგები; - ტეგები მონაცემთა ბლოკებში; - მონაცემთა მთლიანი ბლოკები.
2.	ტეგის მნიშვნელობა	ტეგის მნიშვნელობა შენახულია თვითონ ტეგში (მაგალითად, 15 -ეს მნიშვნელობაა მთელრიცხვა ტეგისა)
3.	ფაქტიური პარამეტრი	ფაქტიური პარამეტრები, ეს ტეგებია, რომლებიც დაკავშირებულია ინტერფეისთან ანდა ინსტრუქციებთან, ფუნქციებთან ანდა ფუნქციონალურ ბლოკებთან.
4.	ფორმალური პარამეტრი (მონაცემთა გადაცემისთვის ბლოკის პარამეტრი	ფორმალური პარამეტრები, ეს ინტერფეისული პარამეტრებია ფუნქციებისა და ფუნქციონალური ბლოკების (შესასვლელები, გამოსასვლელები, გადასასვლელები, და Ret_Val).

ცხრილი 2-1: ტერმინები ტეგებისა და პარამეტრებისათვის

შენიშვნა

თქვენ შეგიძლიათ მონახოთ დამატებითი ინფორმაცია შემდეგი ბმულების დახმარებით:

რამდენი ინფორმაციაა ხელმისაწვდომი ინტერნეტში STEP 7-ის მიგრაციისთვის (TIA Portal)-სა და WinCC (TIA Portal)-ში?

https://support.industry.siemens.com/cs/ww/en/view/56314851

რა მოთხოვნილებებს უნდა აკმაყოფილებდეს STEP7 V5.x პროექტი STEP 7 Professional (TIA Portal)-ში მიგრაციისათვის?

https://support.industry.siemens.com/cs/ww/en/view/62100731

PLC-ს მიგრაცია S7-1500-ისთვის STEP 7 (TIA Portal)-ით

https://support.industry.siemens.com/cs/ww/en/view/67858106

რამდენად რაციონალურად და ეფექტიურად აპროგრამირებთ S7-1200/S7-1500-ს STEP 7 (TIA Portal)-ში?

https://support.industry.siemens.com/cs/ww/en/view/67582299

რატომ არ შეიძლება სხვადასხვა რეგისტრების გადაცემა და პარამეტრების პირდაპირი გადაცემა S7-1500-სთვის STEP 7 (TIA Portal)-ში?

ამასთანავე STL პროგრამების მიგრაცია S7-1500-სთვის აღწერილია ამ ბმულში https://support.industry.siemens.com/cs/ww/en/view/67655405

2.3. დაპროგრამების ენები

სამომხმარებლო პროგრამების შექმნისათვის გამოიყენება დაპროგრამების რამოდენიმე ენა. თვითოეულ ენას აქვს თავისი უპირატესობა, რომლებიც შეიძლება შემდეგში იყოს გამოყენებული სხვადასხვა დროს. თვითოეული ბლოკი სამომხმარებლო პროგრამაში შეიძლება შექმნილ იყოს დაპროგრამების ნებისმიერ ენაზე (იხ. ცხრილი 2-2:).

ცხრილი 2-2: დაპროგრამების ენები

დაპროგრამების ენები	S7-1200	S7-1500
Ladder (LAD)	*	*
Function block diagram (FBD)	*	*
Structured control language (SCL)	~	~
Graph	×	*
Statement list (STL)	×	✓

შენიშვნა

თქვენ შეგიძლიათ მოძებნოთ ინფორმაცია შემდეგი საკითხების ირეგვლივ : დაპროგრამების ენების შედარება SIMATIC S7-1200 / S7-1500

https://support.industry.siemens.com/cs/ww/en/view/86630375

რა სახის მოთხოვნილებები უნდა იქნას გათვალისწინებული S7-SCL პროგრამების მიგრაციის დროს STEP 7 (TIA Portal)-ში?;

https://support.industry.siemens.com/cs/ww/en/view/59784005

რომელი ინსტრუქციები არ არის მხარდაჭერილი STEP 7 (TIA Portal)-ში SCL პროგრამებში?;

https://support.industry.siemens.com/cs/ww/en/view/58002709

რა გზით განისაზღვრება კონსტანტები STEP 7 (TIA Portal) SCL პროგრამებში?. <u>https://support.industry.siemens.com/cs/ww/en/view/52258437</u>

2.4. ოპტიმიზირებლი მანქანური კოდი

TIA Portal და S7-1200/1500 შესაძლებლობას გვაძლევენ მივიღოთ ოპტიმიზირებული მწარმოებლობა შესრულების რეჟიმში, დაპროგრამების ყველა ენაზე. ამასთან, ყველა ენა ერთნაირად კომპილირდება მაშინვე მანქანურ კოდებში.

უპირატესობანი

- დაპროგრამების ყველა ენას აქვთ მწარმოებლობის ერთნაირი მაღალი დონე (ხელწვდომის ერთნაირი ტიპების შემთხვევაში);
- მწარმოებლობა არ დაბლდება შუალდური ბიჯით კომპილიაციის
 შემთხვევაში STL ენაზე;

თვისებები

შემდეგ სურათზე ნაჩვენებია განსხვავებები S7 პროგრამების კომპილიაციის შემთხვევში მანქანურ კოდებში (იხ. სურათი 2-3:).





- S7-300/400/WinAC კონტროლერებისათვის პროგრამები LAD და FBD ენებზე დასაწყისში კომპილირდება STL-ში, ხოლო შემდეგ მანქანურ კოდში;
- S7-1200/1500 კონტროლერებისათვის ყველა დაპროგრამების ენა კომპილირდება მაშინვე მანქანურ კოდებში.

2.5 ბლოკების შექმნა

ყველა ბლოკი, ისეთები მაგალითად როგორიცაა OB, FB და FC შეიძლება დაპროგრამირდეს ზემოთ აღწერილ ნებისმიერ დაპროგრამების ენაზე. ამგვარად, საწყისი ტექსტი SCL-ზე დაპროგრამებისათვის არ შეიქმნება. თქვენ შეგიძლიათ შეირჩიოთ SCL ენა, როდესაც ქმნით ბლოკს, ამის შემდეგ ბლოკი შესაძლებელია მაშინვე დაპროგრამირდეს (იხ. სურათი 2-4:).



სურათი 2-4: ფანჯარა "Add new block" (ახალი ბლოკის შექმნა)

2.6 ოპტიმიზირებული ბლოკები

S7-1200/1500 კონტროლერებს აქვთ მონაცემთა ოპტიმიზირებული შენახვის შესაძლებლობლობა. ოპტიმიზირებულ ბლოკებში ყველა ტეგი ავტომატურად ლაგდება მონაცემთა ტიპის მიხედვით. აღნიშნული მეთოდი შესაძლებლობას გვაძლევს შევამციროთ მონაკვეთები ტეგებს შორის, ამრიგად ასეთი ტეგები ოპტიმიზირებულნი არიან პროცესორისათვის ხელწვდომის დროის მიხედვით. არაოპტიმიზირებული ბლოკები არსებობენ S7-1200/1500-ებთან შეთავსებადობის მიზნისთვის.

უპირატესობანი

- ხელწვდომა ყოველთვის მიმდინარეობს სწრაფად, რადგანაც ბაზა სადაც არის შენახული ინფორმაცია არის ოპტიმიზირებული სისტემის მიერ და დამოუკიდებელია მისი აღწერისაგან;
- არ არსებობს შეცდომის შესაძლებლობა აბსოლუტირი დამისამართების გამოყენებისას მონაცემებთან მიკითხვის დროს, რომელმაც შეცვალა სიმბოლური დამისამართება;
- ცვლადების აღწერის ცვლილებები აღარ გამოიწვევს ხელწვდომის შეცდომას, რადგან მაგალითად, HMI იღებს სიმბოლურ ხელწვდომას;
- ცალკეული ტეგები შესაძლებელია აღწერილ იყოს როგორც"retain" (შენახვითობა);
- არავითარი გაწყობები მონაცემთა ეკზემპლიარულ ბლოკებში არ არის საჭირო.
 ყველაფერი გადაეცემა FB-ში (შენახვითობის ჩათვლით);
- მეხსიერების რეზერვი მონაცემთა ბლოკში შესაძლებლობას გვაძლევს ვცვალოთ ფაქტიური მნიშვნელობები მონაცემთა კარგვების გარეშე (იხ. თავი 3.2.7 ჩატვირთვა განმეორებითი ინციალიზაციის გარეშე).

2.6.1. S7-1200: ოპტიმიზირებული ბლოკი (იხ. სურათი 2-5:).



სურათი 2-5: სტანდარტული და ოპტიმიზირებული ბლოკები S7-1200-ში

თვისებები

- მონაკვეთები ტეგებს შორის არ არის, რადგანაც ტეგები ყველაზე მეტი მონაცემთა ტიპებით, ყველაზე დიდი ზომით, განლაგდება ბლოკის დასაწყისში, ხოლო ყველაზე მცირე ზომით-ბოლოში;
- ოპტიმიზირებული ბლოკებისათვის შესაძლებელია მხოლოდ სიმბოლური ხელწვდომა.

2.6.2 S7-1500: ოპტიმიზირებული ბლოკი (იხ. სურათი 2-6:, სურათი 2-7:).



სურსათი 2-6: ოპტიმიზირებული ბლოკი S7-1500



სურათი 2-7: მეხსიერების განაწილება მონაცემთა ოპტიმიზირებულ ბლოკებში

 შენახვის სტრუქტურა განცალკავებულია და შეიძლება კოპირებულ იქნას როგორც ერთი ბლოკი; შესანახი მონაცემები ჩაიწერება ცალკე არეში და შეიძლება კოპირებულ იქნას როგორც ერთი ბლოკი. კვების ბლოკის შეშფოთების შემთხვევაში, ეს ინფორმაცია შეინახება CPU-ს შიგა მეხსიერებაში. "MRES" შეასრულებს იმ ინფორმაციის ჩამოყრას საწყის მნიშვნელობებზე, რომლებიც შენახულია ჩამტვირთავ მეხსიერებაში.

თვისებები

- მონაცემთა შორის მონაკვეთები არ არსებობს, რადგანაც ტეგები ყველაზე დიდი ზომით განლაგდება ბლოკის დასაწყისში, ხოლო ყველაზე მცირე ზომით-ბლოკის ბოლოში;
- სწრაფი ხელწვდომა, პროცესორში მონაცემთა შენახვის ასეთი სტრუქტურიდან გამომდინარე (ყველა ტეგი ჩაიწერება ისეთნაირად, რომ S7-1500-მა შესმლოს პირდაპირ ჩაწეროს და წაიკითხოს ტეგები ერთი მანქანური ბრმანებით).
- ლოგიკური ტიპის (Bool) ტეგები შეინახება როგორც ბაიტი უფრო სწრაფი ხელწვდომისათვის. ამ მიზეზით კონტროლერს არ ჭირდება ხელწვდომის მასკა.
- ოპტიმიზირებულ ბლოკებს აქვთ სარეზერვო მეხსიერება გადატვირთვის
 საჭიროებისათვის მუშაობის პროცესში (იხ. თავი 3.2.7 ჩატვირთვა
 განმეორებითი ინიციალიზაციის გარეშე);
- ოპტიმიზირებული ბლოკებისათვის შესაძლებელია მხოლოდ სიმბოლური ხელწვდომა (იხ. სურათი 2-7:).

2.6.3 მონაცემთა შენახვის ყველაზე უკეთესი ვარიანტი S7-1500 - ში

SIMATIC-ის პირველ კონტროლერებთან შეთავსებადობის მიზნით, მონაცემთა შენახვის "Big-Endian" პრინციპი გამოყენებულ იქნა S7-300/400 კონტროლერებში. S7-1500 კონტროლეტრების ახალი თაობა ყოველთვის იღებს ხელწვდომას 4 ბაიტთან (32 ბიტი) "Little-Endian" თანმიმდევრულობაში პროცესორის არქიტექტურის ცვლილებებისაგან გამომდინარე. ეს კი ზოგიერთ ჩემთხვევაში იძლევა გარკვეულ უპირატესობებს (იხ. სურათი 2-8:, ცხრილი 2-3:).



სურათი 2-8: ხელწვდომა მონაცემებთან S7-150

	სტანდარტული ბლოკი	ოპტიმიზირებული ბლოკი
1.	კონტროლერც ჭირდება მიიღოს	კონტროლერი ინახავს ტეგებს,
	ხელწვდომა 2x4 ბაიტთან, რათა	ხელწვდომა ოპტიმიზირებულია.
	შეძლოს მიღება 4 ბაიტური	სრულდება 32 ბიტური (REEAL)
	მონაცემისა (მაგ. REAL). ამასთან,	ხელწვდომა.
	ბაიტების თაწმიმდევრობა	ბაიტების თაწმიმდევრობის
	აუცილებელია შეიცვალოს.	ცვლილება არ მოითხოვება.
2.	წაიკითხება მთელი ბაიტი და	თვითოეულ ბიტს მიეკუთვწება
	დაედება მასკა.	ბაიტი. ხელწვდომის მიღების
	ბაიტი დაიბლოკება წებისმიერი	შემთხვევაში, კონტროლერი არ
	სხვა ხელწვდომისათვის.	ადებს მასკას ბაიტზე.
3.	ბლოკის მაქსიმალური ზომა 64კბ.	ბლოკის მაქსიმალური ზომა 16მბ.

ცხრილი 2-3: ხელწვდომა მონაცემებთან S7-1500 კონტროლერებში

რეკომენდაციები

- ყოველთვის გამოიყენეთ მხოლოდ ოპტიმიზირებული ბლოკები:
 - ისინი არ ითხოვენ აბსოლუტურ დამისამართებას, მათზედ ხელწვდომა შესაძლებელია მიღწეულ იქნას სიმბოლური დამისამართებით. ირიბი დამისამართებაც შესაძლებელია აგრეთვე სიმბოლური მონაცემების დახმარებით (იხ. თავი 3.6.2 ARRAYმონაცემთა ტიპი და ელემენტებთან ირიბი ხელწვდომა);
 - ოპტიმიზირებული ბლოკების დამუშავება კონტროლერში გაცილებით
 სწრაფად ხდება ვიდრე სტანდარტულ ბლოკებში.
- თავი აარიდეთ მონაცემთა კოპირებას ოპტიმიზირებულ და არაოპტიმიზირებულ ბლოკებს შორის. საწყის ფორმატიდან სასურველში აუცილებელმა გარდაქმნებმა შეიძლება დაიკავოს ბევრი საპროცესორო დრო.

მაგალითი : ოპტიმიზირებული ხელწვდომის დაყენება ბლოკთან

ოპტიმიირებული ხელწვდომა ბლოკთან სიჩუმით აქტივირებულია ყველა შექმნილი ბლოკისთვის S7-1200/1500-ში. ხელწვდომა ბლოკთან შეიძლება დაყენებულ იქნას OB, FB-სთვის და გლობალური DB-სთვის. ეკზემპლიარული DB-სთვის გაწყობა დამოკიდებულია შესაბამის FB-ზე. ბლოკის მიგრაციისათვის S7-300/400 კონტროლერებიდან S7-1200/1500 კონტროლერებში, ბლოკთან ხელწვდომა არ უქმდება ავტომატურად. თქვენ შეგიძლიათ შეცვალოთ ხელწვდომის ტიპი მოგვიანებით "optimized block access"-ით (ბლოკის ოპტიმიზირებული ხელწვდომა). თქვენ აუცილებლად დაგჭირდებათ პროგრამის კომპილირება ბლოკთან შეცვლილი ხელწვდომის შემდეგ. თუ თქვენ შეცვლით FB-ს "optimized block access"-ით (ბლოკის ოპტიმიზირებული ხელწვდომა), ამ შემთხვევაში დანიშნული ეკზემპლიარული მონაცემთა ბლოკები ავტომატურად განახლდება. მიჰყევით ინსტრუქციებს, რომლებიც აღწერილია ქვემოთ, ბლოკზე ოპტიმიზირებული ხელწვდომის დავალებისთვის (იხ. ცხრილი 2-4:).

ბიჯი	ინსტრუქცია					
1.	დააჭირეთ ღილაკს "Maximizes/minimizes Overview"					
	(გაშალოთ/დაახვიოთ ასახვა) პროექტის ნავიგატორში.					
	Project Léte View Intern Oxford Oxford Table Project See Project See					
2.	გადადით საქაღალდეში"Program blocks"(პროგრამული ბლოკები)					
3.	ოპტიმიზაციიც აწყობებს. ამ ფანჯარაში პარამეტრი "Optimized block access" (ოპტიმიზირებული ხელწვდომა ბლოკზე) შეიმლება აგრეთვე შეცვლილ იყოს.					
	შენიშვნა: ეკზემპლიარული მონაცემთა ბლოკები (აქ "Function_block_1_DB) მემკვიდრეობით იღებს "ოპტიმიზირებულის" მფდგომარეობას შესაბამისი FB-სგან. სწორედ ამიტომ თვისება "ოპტიმიზირებული" მეიცემა FB-ში. პროექტის კომპილიაციის შემდეგ DB-ს მიეკუთვნება თვისება,					

ცხრილი 2-4: ბლოკზე ოპტიმალური ხელწვდომის დავალება

ოპტიმიზირებული და არაოპტიმიზირებული ბლოკების ასახვა TIA Portal-ში

მომდევნო ორ გამოსახულებაზე შეიძლება დავინახოთ განსხვავება ოპტიმიზირებულ და არაოპტიმიზირებულ მონაცემთა DB ბლოკებს შორის (იხ. სურათი 2-9:, 2-10:). გლობალურ DB-ებს განეკუთვნება ზუსტად იგივე განსხვავებები (იხ. ცხრილი 2-5:).

	Tancaor_book_1_bb								
		Na	me	Data type	Start value	Retain	Visible in	Setpoint	
1	•	•	Input						
2	-0	•	Input_bool_1	Bool	false	1	1		
3	-	٠	input_byte_1	Byte	16#0		~		
4	•	٠	input_bool_2	Bool	false	v	v		
5	•	•	Input_word	Word	16#0		1		
6	-0	٠	Input_byte_2	Byte	16#0		4		
7		•	Output						
8	•	•	Output_bool_1	Bool	false		1		
9	-		InOut						
10			Static						

სურათი 2-9: მონაცემთა ოპტიმიზირებული ბლოკი (წანაცვლების გარეშე)

	Function_Block_1_DB									
-		Na	me	Data type	_	Offset	Start value	Retain	Visible in	Setpoint
1	-	٠	Input							
2	•	٠	Input_bool_1	Bool		0.0	false		4	
3	•	٠	Input_byte_1	Byte		1.0	16#0		4	
4	-	٠	Input_bool_2	Bool		2.0	false		4	
5	•	٠	Input_word	Word		4.0	16#0		4	
6		٠	input_byte_2	Byte		6.0	16#0		4	
7	-	٠	Output							
8	-	٠	Output_bool_1	Bool		8.0	false		4	
9			InOut							
10	-0		Static							

სურათი 2-10: მონაცემთა არაოპტიმიზირებული ბლოკი (წანაცვლების გარეშე)

ცხრილი 2-5: განსხვავებანი: ოპტიმიზირებული და არაოპტიმიზირებული მონაცემთა ბლოკი

ოპტიმიზირებული	არაოპტიმიზირებული
მონაცემთა ბლოკი	მონაცემთა ბლოკი
ოპტიმიზირებული მონაცემთა	არაოპტიმიზირებულ ბლოკებში "წანაცვლ-
ბლოკი დამისამართდება სიმბო-	ება" აისახება და შეიძლება გამოყენებულ
ლურად. წანაცვლება არ აისახება.	იქნას დამისამართებისთვის.
ოპტიმიხზირებულ ბლოკებში ნები-	არაოპტიმიზირებულ ბლოკებში მხოლოდ
სმიერი ცალკეული ტეგი აღიწერება	ყველა ან არცერთ. ტეგს შეიძლება
როგორც "Retain" (შენახვითი).	ჰქონდეთ თვისება "Retain" (შენახვითი).

ტეგების შენახვითობა გლობალურ DB-ში განისაზღვრება თვითონ DB-ში. სიჩუმით იგი შენახვითია.

ეკზემპლიარული ბლოკის შენახვითობა განისაზღვრება ფუნქციონალურ ბლოკში (არა ეკზემპლიარულ DB-ში). ასეთი აწყობა შემდეგში გამოყენებული იქნება ყველა ეკზემპლიარულ ბლოკთან FB.

ხელწვდომის ტიპები ოპტიმიზირებული და არაოპტიმიზირებული ბლოკებისათვის

შემდეგ ცხრილში ნაჩვენებია ბლოკებთან ხელწვდომის ყველა ტიპი. (იხ. ცხრილი 2-6:) ცხრილი 2-6: ხელწვდომის ტიპები

ხელწვდომის ტიპი	ოპტიმიზირებული ბლოკი	არაოპტიმიზირებული ბლოკი
სიმბოლური	×	×
ინდექსირებული (ელემენტებთან)	~	×
არჩევითი (Slince) ხელწვდომა	~	~
AT ინსტრუქცია	× (ალტერნატივა: არჩევითი) ხელწვდომა)	Ý
აბსოლურური	× (ალტერნატივა: მასივი ინდექსით)	× .
ირიბი (მაჩვენებლით)	, ალტერნატივა: (VARIANT / მასივი ინდექსი)	✓
ჩატვირთვა განმეორებითი ინციალიზაციის გარეშე	✓	× .

შენიშვნა

თქვენ შეგიძლიათ მოძებნოთ დამატებითი ინფორმაცია შემდეგი საკითხების ირგვლივ:

რა განსხვავებებია საჭირო რომ ვიცოდეთ მონაცემთა შენახვის შესახებ ოპტიმიზირებულ და სტანდარტულ ხელწვდომებს შორის STEP 7-ში (TIA portal)?; https://support.industry.siemens.com/cs/ww/en/view/67655611 რა თვისებებზეა საჭირო ყურადღების გამახვილება STEP 7-ში (TIA portal) ინსტრუქციისათვის "READ_DBL" დაქ "WRIT_DBL", თუ DB-ს გამოყენება ხდება ოპტიმიზირებული ხელწვდომით? https://support.industry.siemens.com/cs/ww/en/view/51434747

2.6.4 გარდაქმნა ოპტიმიზირებულ და არაოპტიმიზირებულ ტეგებს შორის

მთავარი რეკომენდაცია - ეს მუშაობაა ოპტიმიზირებულ ტეგებთან. მიუხედავად ამისა, თუ ჩვენ გინდათ რაიმე შემთხვევებში მიმართოთ დაპროგრამების ძველ სტილს, ასეთ შემთხვევაში პროგრამაში იქნება ოპტიმიზირებული და არაოპტიმიზირებული ბლოკების ნარევი.

სისტემა განსაზღვრავს თვითოეული ტეგის ჩაწერის შიგა ადგილს, დამოუკიდებლად რთული (მომხმარებლის მიერ განსაზღვრული მონაცემთა ტიპით) თუ ელემენტარული მონაცემთა ტიპით (INT, REAL და სხვ.) იქნა იგი აღწერილი. ორ ტეგს შორის ურთიერთმოქმედების შემთხვევაში შენახვის სხვადასხვა პრინციპით, სისტემა ავტომატურად ასრულებს გარდაქმნას. სტრუქტურიზებული ტეგების შემთხვევაში აღნიშნული გარდაქმნა ითხოვს გაზრდილ მწარმოებლობას და შემლებისდაგვარად ასეთ სიტუაციას უკეთესია თავი ავარიდოთ.

2.6.5 პარამეტრების გადაცემა ბლოკებს შორის ოპტიმიზირებულ და სტანდარტულ ხელწვდომის ტიპებით

თუ ბლოკის გამოძახებისას, სტრუქტურები გადაეწოდება გამომძახებელ ბლოკში როგორც გამავალი (InOut) პარამეტრი, მაშინ ის იქნება გადაცემული ბმულით (იხ. თავი 3.3.2 გადაცემა გამავალი პარამეტრის ბმულით).

მაგრამ, ეს არ ეხებსა სიტრუაციას, თუ ერთი ბლოკი "Optimized access" თვისებითაა (ოპტიმიზირებული ხელწვდომა), ხოლო მეორე ბლოკს აქვს თვისება "Standard access" (სტანდარტული ხელწვდომა). პრინციპულად, ყველა პარამეტრი გადაიცემა როგორც ასლები (იხ. თავი 3.3.1 გადაცემა მნიშვნელობის მიხედვით შესასვლელი პარამეტრების გავლით).

ამ შემთხვევაში გამომძახველი ბლოკი ყოველთვის მუშაობს მონაცემთა ასლებთან. ბლოკის დამუშავების დროს მოცემული მნიშვნელობები შეიძლება შეცვლილ იყოს, და ამის შემდეგ კოპირებულ იქნას უკან საწყის ოპერანდში.

აღნიშნული მექანიზმი შეიძლება იყოს არაკორექტული ზოგიერთ შემთხვევაში, თუ კი საწყისი ოპერანდი იცვლება ასინხრონული პროცესით, მაგალითად ხელწვდომა HMIსთან ანდა შეცდომების დამუშავების OB-სთან. თუ კი ბლოკის დამუშავების შემდეგ ასლები გადაცემულ იქნება საწყის ოპერანდში, მაშინ მასში გადაწერილი იქნება ასინქრონული ცვლილებების მონაცემებიც.

შენიშვნა

თქვენ შეგიძლიათ მოძებნოთ დამატებითი ინფორმაცია შემდეგი საკითხების ირგვლივ: რატომ გადაიწერება HMI სისტემის მონაცემები ანდა Web სერვერის მონაცემები S7-1500-ში?

https://support.industry.siemens.com/cs/ww/en/view/109478253

რეკომენდაციები

 ყოველთვის დააყენეთ ხელწვდომის ერთიდაიგივე ტიპი იმ ორივე ბლოკისათვის, რომლებიც ურთიერთქმედებენ ერთმანეთთან.

2.6.6 კომუნიკაციები ოპტიმიზირებული მონაცემებით

ინტერფრეისი (CPU, CM) გადასცემს მონსაცემებს იმ თანმიმდევრობით, როგორადაც ისინი განლაგებულნი არიან (არა აქვს მნიშვნელობა ისინი ოპტიმიზირებულები არიან თუ არა), (იხ. სურათი 2-11:).



სურათი 2-11: კომუნიკაციები CPU-CPU

გასაგზავნი ინფორმაცია შეიძლება იყოს: მისაღები ინფორმაცია შეიძება იყოს:

- ოპტიმიზირებული;
- არაოპტიმიზირებული;
- ტეგი (ნებისმიერი ტიპის);
- ტეგი (ნებისმიერი ტიპის);
- ოპტიმიზირებული;
- არაოპტიმიზირებული;
- ტეგი (ნებისმიერი ტიპის);
- ბუფერი (ბაიტების მასივი).

მაგალითი

- ტეგი PLC-ის მონაცემების ტიპით (მონაცემთა ჩაწერა) გადაეცემა CPU-ს;
- გამგზავნელ CPU-ში, ტეგი მოცემულია როგორც კომუნიკაციური ბლოკის (TSEND_C) ფაქტიური პარამეტრი;
- მიმღებ CPU-ში, მისაღები ინფორმაცია გადაეცემა იგივე ტიპის ტეგში;
- ასეთ შემთხვევაში, მიღებული ინფორმაციით შეიძლება მუშაობის გაგრძელება სიმბოლური დამისამართებთ.

შენიშვნა

ნებისმიერი ტეგი ანდა მონაცემთა ბლოკი (რომლებიც წარმოებულია PLC data types მონაცემთა ტიპებისაგან) შეიძლება გამოყენებულ იქნას როგორც მონაცემთა ჩანაწერები.

შენიშვნა

ასევე შესაძლებელია, რომ გასაგზავნი და მისაღები მონაცემები არ იყოს იდენტიურები:

გასაგზავნი მონაცემები	მისაღები მონაცემები
ოპტიმიზირებულები ->	არაოპტიმიზირებულები
არაოპტიმიზირებულები ->	ოპტიმიზირებულები

კონტროლერი ავტომატურად ასრულებს მონაცემთა კორექტულ გადაცემას და მათ შენახვას.

2.7 ბლოკების თვისებები

2.7.1 ბლოკის ზომები

S7-1200/S7-1500 კონტროლერებისათვის ბლოკების მაქსიმალური ზომები მეხსიერების თვალსაზრისით არის მნიშვნელოვნად გაზრდილი (იხ. ცხრილი 2-7:).

			0	~ ~ 0 0
მაქსიმალე რაოდ (დამოკიდებულ მეხსიერების ზ	ური ზომა და დენობა ია მირითადი ომაზე)	S7 -300/400	S7-1200	S7-1500
DB	მაქს. ზომა	64 кб	64 кб	64 кნ ((არაოპტიმიზირებული) 10 Mნ (ოპტიმიზირებული) CPU1518)
	მაქს. რა-ბა	16.000	65.535	65.535
FC/FB	მაქს. ზომა	64 кб	<u>64</u> к б	512 кნ 3 Mნ (ოპტიმიზირებული) CPU1518)
	მაქს. რა-ბა	7.999	65.535	65.535
FC / FB / DB	მაქს. რა-ბა	4.096 (CPU319) 6.000 (CPU412)	1.024	10.000 (CPU1518)

რეკომენდაციები

- გამოიყენეთ DB S7-1500 კონტროლერებისათვის როგორც დიდი მოცულობის მონაცემების შენახვის ადგილი;
- მონაცემები, რომლებიც > 64 კბ-ზე, S7-1500 კონტროლერებში შეიძლება შენახულ იყოს ოპტიმიზირებულ DB-ში (მაქსიმალური ზომა 10 მბ).

2.7.1 მაორგანიზებელი ბლოკების რაოდენობა (OB)

OB შეიძლება გამოყენებულ იქნას სამომხმარებლო პროგრამის იერარქიის შექმნისათვის. ამ მიზნისათვის ხელმისაწვდომია სხვადასხვა OB-ები (იხ. ცხრილი 2-8:). ცხრილი 2-8: მაორგანიზებელი ბლოკების რაოდენობა

მაორგანიზ. ბლოკის ტიპი	S7-1200	S7-1500	მიზანი
<u>ციკლური და სასტარტო</u>	100	100	პროგრამის
OB	100		მოდულიარიზაცია
13161.8mmo 5000.81	50	50	ცალკეული ხდომილე-
222020 9600 633902	50		ბის დამუშავების OB
წყვეტა დროის	4*	20	პროგრამის
დაყოვნებით		20	მოდულიარიზაცია
4: ციკლური წყვეტა		20	პროგრამის
		20	მოდულიარიზაცია
დღის დრო	×	20	პროგრამის
			მოდულიარიზაცია

* დაწყებული ოპერაციული სისტემის V4 ვერსიიდან წყვეტები ხელმისაწვდომია დაყოვნებით, ხელმისაწვდომია აგრეთვე ოთხი "watchdog" წყვეტა.

რეკომენდაციები

- გამოიყენეთ OB სტრუქტურირებული სამომხმარებლო პროგრამების იერარქიის შესაქმნელად;
- უფრო დაწვრილებითი ინფორმაციისათვის OB-ის შესახებ, მიმართეთ თავს
 3.2.1 მაორგანიზებელი ბლოკი (OB).

2.8 მონაცემთა ახალი ტიპები S7-1200/1500-ში

S7-1200/1500 კონტროლერები მხარს უჭერს მონაცემთა ახალ ტიპებს, რაც შესაძლებლობას იძლევა შეიქმნას უფრო ეფექტიური პროგრამული კოდი. ახალი 64 ბიტიანი მონაცემების გამოყენებისას, შესაძლებელია ცვლადების გამოყენება მნიშვნელობათა დიდი დიაპაზონით და ასევე გამოთვლების სიზუსტის მნიშვნელოვანი გაზრდა.

შენიშვნა

თქვენ შეგიძლიათ მონახოთ დამატებითი ინფორმაცია შემდეგი საკითხების ირგვლივ: როგორ მოვახდინოთ მონაცემთა ტიპების გარდაქმნა TIA Portal-ში S7-1200/1500 კონტროლერებისათვის?

https://support.industry.siemens.com/cs/ww/en/view/48711306

2.8.1. ელემენტარული მონაცემთა ტიპები (იხ. ცხრილი 2-9:)

ტიპი	ზომა	Value range
USint	<mark>8 ბიტი</mark>	0255
SInt	8 ბიტი	-128 127
UInt	16 ბიტი	065535
UDInt	32 ბიტი	0 4.3 მილიოწი
ULInt*	64 ბიტი	0 18,4 'ტრილიონი (10 ¹⁸)
Lint*	64 ბიტი	-9,2 ტრილიონი _ 9,2 ტრილიონი
LWord	64 ბიტი	16#0000 0000 0000 0000 до 16# FFFF FFFF FFFF FFFF

ცხრილი 2-9: მთელრიცხვა მონაცემთა ტიპები

** მხოლოდ S7-1500-ისთვის

ცხრილი 2-10: ათობითი მონაცემთა ტიპები მცოცავი მძიმით

ტიპი	ზომა	მნიშვნელობათა დიაპაზონი
Real	32 ბიტი (1 ნიშნის ბიტი, 8 ექსპონენტის	-3.40e+38 3.40e+38
	ბიტი, 23 ბიტი (მანტისა), სიზუსტით 27	
	ნიშანი მძიმის შემდეგ	
L Deal	64 ბიტი (1 წიშნის ბიტი, 11 ბიტი	
LReal	ექსპონენტა, 52 ბიტი მანტისა)	-1.79e+308 1.79e+308
	სიზუსტით 15 ნიშანი მმიმის შემდეგ	

შენიშვნა

TIA Portal-ში არის გლობალური ბიბლიოთეკა "Long Fanctions" ინსტრუქციებით მონაცემთა დიდი ტიპებისთვის.



შენიშვნა

თქვენ შეგიძლიათ მონახოთ დამატებითი ინფორმაცია შემდეგი საკითხის ირგვლივ: STEP 7 (TIA Portal)-ში, DINT ტიპების შეკრების რეზულტატი SCL-ში რატომ აისახება არაკორექტულად?

https://support.industry.siemens.com/cs/ww/en/view/98278626

2.8.2 მონცემთა ტიპი Data_Time_Long (იხ. ცხრილი 2-11:)

ცხრილი 2-11: მონაცემთა ტიპი DTL (Data_Time_Long)

წელი თვე დღე კვირ. დღე საათი წუთი წამი | ნანოწამი

DTL ყოველთვის კითხულობს მიმდინარე სისტემურ დროს. ცალკეულ მნიშვნელობებთან ხელწვდომა სრულდება სიმბოლურად (მაგალითად, My_Timestamp.Hour)

უპირატესობა

• თვითოეულ მინდორთან (მაგალითად, Year, Month, ...) შეიძლია შესრულდეს სიმბოლური ხელწვდომა.

რეკომენდაცია

გამოიყენეთ ახალი მონაცემთა ტიპი DTL სიმბოლური ხელწვდომით (მაგალითად, My_Timestamp.Hour).

შენიშვნა

თქვენ შეგიძლიათ იპოვოთ დამატებითი ინფორმაცია შემდეგი საკითხების ირგვლივ: როგორ შეგეძლებათ დაავალოთ, წაიკითხოთ და დაარედაქტოროთ STEP 7 (TIA Portal)-ში თარიღი და დრო CPU S7-300/S7-400/S7-1200/S7-1500 მოდელებისთვის? <u>https://support.industry.siemens.com/cs/ww/en/view/43566349</u> როგორი ფუნქციებია ხელმისაწვდომი STEP 7 V5.5-სა და TIA Portal-ში DT და DTL მონაცემთა ტიპების დამუშავებისათვის? <u>https://support.industry.siemens.com/cs/ww/en/view/63900229</u>

2.8.3 მონაცემთა დამხმარე ტიპები დროისათვის (იხ. ცხრილი 2-12:)

ცხრილი 2-12: დროის მონაცემთა ტიპები (მხოლოდ S7-1500)

ტიპი	ზომა	მნიშვნელობათა დიაპაზონი
LTime	64 ბიტი	LT#-106751d23h47m16s854ms775us808ns LT#+106751d23h47m16s854ms775us807ns
LTIME_OF_DAY	64 ბიტი	LTOD#00:00:00.000000000 LTOD#23:59:59.9999999999

2.8.4 მონაცემთა ტიპები უნიკოდთან მუშაობისათვის

WCHAR და WSTRING მონაცემთა ტიპები შეიძლება გამოყენებულ უნიკოდის ფორმატში წარმოდგენილ სიმბოლოებთან მუშაობისათვის (იხ. ცხრილი 2-13:)

ცხრილი 2-13: მონაცემთა ტიპები უნიკოდთან სამუშაოდ (მხოლოდ S7-1500)

ტიპი	ზომა	მნიშვნელობათა დიაპაზონი
WCHAR	2 байта	-
WSTRING	(4 + 2*n) байт	წინასწარ დაყენებული მნიშვნელობა 0 254 სიმბოლო მაქს. მნიშვნელობა 0 16382

თვისეზები

- n = სიმბოლოთა თანმიმდევრობის სიგრძე
 - მაგალითად, სიმბოლოთა დამუშავება ლათინურ, ჩინურ ანდა სხვა ენებზე;
 - სტრიქონთა გაწყვეტები, გვერდების გადაშლა, ტაბულიაციის სიმბოლოები, პრობელის სიმბოლო;
 - სპეციალური სიმბოლოები: დოპლერის ნიშანი, ბრჭყალები.

მაგალითი

- WCHAR# '4'
- WSTRING# 'Hello World'

2.8.5 მონაცემთა ტიპი VARIANT (S7-1500 da S7 - 1200 4.1 ვერსიიდან)

VARIANT ტიპის პარამეტრი ეს მაჩვენებელია, რომელაც შეუძლიათ მიაკითხოს ტეგებს სხვადასხვა მონაცემთა ტიპებით. განსხვავებით ANI მაჩვენებლისაგან VARIANT მჩვენებელია ტიპის შემოწმებით. საწყისი და მიზნობრივი სტრუქტურები გამოსასვლელზე და შესასვლელზე მოწმდება შესრულების რეჟიმში და ისინი უნდა იყვნენ იდენტურები.

VARIANT გამოიყენება, მაგალითად შესასვლელი პარამეტრის სახით (TSEND_C) საკომუნიკაციო ბლოკებისათვის (იხ. სურათი 2-12:).



სურათი 2-12: VARIANT მონაცემთა ტიპი შესასვლელი პარამეტრის სახით ინსტრუქციისათვის TSEND_C.

უპირატესობანი

- ტიპის ჩაშენებული შემოწმება თავიდან გვაცილებს ხელწვდომის შეცდომას;
- სიმბოლური დამისამართებიდან გამომდინარე ასეთი ტიპის ტეგებთან, კოდის წაკითხვა ხდება უფრო ადვილად;
- კოდის დაწერა წარმოებს უფრო ეფექტურად და სწრაფად;
- Variant მაჩვენებლები ინტუიციურად უფრო გასაგებია, ვიდრე მაჩვენებლები ANY;
- Variant ტეგები შეიძლება პირდაპირ გამოყენებულ იქნას სისტემურ ფუნქციებში;
- შესძლებელია სხვადასხვა სტრუქტურიზებული მონაცემების უფრო მოქნილი და ეფექტური გადაცემა.

თვისებები

Variant და ANY მაჩვენებლების შედარება შესაძლებლობას გვაძლევს დავინახოთ განსხვავება ტიპების მონაცემების თვისებებს შორის (იხ. ცხრილი 2-14:).

ANY	Variant
ითხოვს 10კბაიტ მეხსიერებას გარკვეული სტრუქტურით	არ ითხოვს სამომხმარებლო მეხსიერებას
ინციალიზაცია ან მეხსიერების დანიშნული არეებით ანდა ANY სტრუქტურების შევსებით	ინციალიზაცია სისტემური ინსტრუქციების დანიშნული მეხსიერებათა არეებით
დაკავშირებული სტრუქტურების არატიპიზირებული ტიპი შეუმლებელია იყოს გარკვეული	დაკავშირებული სტრუქტურების ტიპიზირებული ტიპი შეიძლება განსაზღვრულ იქნას მასივის სიგრძით
ნაწილობრივ ტიპიზირებულია- შეიძლება განსაზღვრულ იყოს მასივის ზომა	VARIANT შეიძლება ასევე დამუშავებულ და იქნას სისტემური ინსტრუქციების დახმარებით

ცხრილი 2-14: Variant და ANY მაჩვენებლების შედარება

რეკომენდაციები

- გამოიყენეთ მხოლოდ VARIANT მონაცემთა ტიპი ირიბი დამისამართებით მუშაობის დროს, თუ კი მონაცემთა ტიპი უცნობია პროგრამის მუშაობის დაწყებამდე;
- შეამოწმეთ რისთვის იყენებთ თქვენ ANY მაჩვენებელს. უმეტეს შემთხვევაში მისი გამოყენება ზედმეტია (იხ. ცხრილი ქვემოთ);
- გამოიყენეთ მხოლოდ VARIANT მონაცემთა ტიპი ირიბი დამისამართებით მუშაობის დროს, თუ კი მონაცემთა ტიპები უცნობია პროგრამის მუშაობის დაწყებამდე.
 - გამოიყენეთ მონცემთა ტიპი VARIANT InOut პარამეტრის სახით იმ ბლოკების შესაქმნელად, რომლებიც უნდა იყვნენ დამოუკიდებელნი ფაქტიური პარამეტრების მონაცემთა ტიპებისაგან (იხ. მაგალითი ამ თავში);

- გამოიყენეთ მონცემთა ტიპი VARIANT ANY მაჩვენებლის ნაცვლად. ტიპის
 ჩაშენებული შემოწმებისგან გამომდინარე, შეცდომების აღმოჩენა ხდება
 წინასწარ. სიმბოლური დამისამართებიდან გამომდინარე, პროგრამული
 კოდი ხდება უფრო გასაგები;
- გამოიყენეთ ინსტრუქცია Variant, მაგალითად, ტიპის განსაზღვრისათვის (იხ. შემდეგი მაგალითი 2.9.3 თავში (ინსტრუქციები VARIANT);
- გამოიყენეთ ინდექსირებული მასივები (ARRAY) მაჩვენებელი ANY-ის ნაცვლად, მასივის ელემენტების მისამართზე (იხ. თავი 3.6.2 ARRAY მონაცემთა ტიპი და ირიბი ხელწვდომა ელემენტებთან) (იხ. ცხრილი 2-15:) ცხრილი 2-15: ANY მაჩვენებლის შედარება და მისი შეცვლა

სად გამოიყენება ANY?		შეცვლა S7-1200/1500
დაპროგრამების ფუნქციები, სადაც შეიძლება გამოყენებულ იქნას სხვადასხვა მონაცემთა ტიპები	^	ფუნქციები Variant მაჩვენრებლით InOut პარამეტრის სახით ბლოკებისათვის (იხ. შემდეგი მაგალითები)
მასივების დამუშავება - მაგ., წაკითხვა, ინციალიზაცია, ერთნაირი ტიპის ელემენტაბის კოპირება	→	სტანდარტული ფუნქციები მასივებისათვის - წაკითხვა და ჩაწერა #myArray[#index] (იხ. თავი 3.6.2 მონაცემთა ტიპი ARRAY და ირიბი ხელწვდომა ელემენტებთან) -კოპირება MOVE_BLK-ს დახმარებით (იხ. თავი 2.9.2 MOVE ინსიბრუქციები
სტრუქტურების გადაცემა და დამუშავება - მაგალითად, სტრუქტურის გადაცემა, განსაზღვრული მომხმარებლის მიერ ANY მაჩვენებლის დახმარებით ფუნქციებში	→	სტრუქტურების გადაცემა InOut პარამეტრების სახით - იხ, თავი 3.3.2 გადაცემა ბმულით InOut პარამეტრის გავლით

მაგალითი

სამომხმარებლო პროგრამაში VARIANT მონაცემთა ტიპის გამოყენებისას შესაძლებელია განისაზღვროს მონაცემთა ტიპი და შესრულდეს შესაბამისი დამუშავება. შემდეგ ფუნქციაში "MoveVariant" ნაჩვენებია ასეთი მიდგომა.

- ფორმალური პარამეტრი InOut "InVar" (მონაცემთა ტიპი VARIANT) გამოიყენება იმისათვის, რომ ნაჩვენები იქნას ტეგის დამოუკიდებლობა მონაცემთა ტიპისაგან;
- ფიზიკური პარამეტრის მონაცემთა ტიპი განისაზღვრება "Type_Of" ინსტრუქციის დახმარებით;

 "MOVE_BLK_VARIANT" ინსტრუქციის დახმარებით ტეგის მნიშვნელობა კოპირდება სხვა გამომავალ ფორმალურ პარამეტრებში, მონაცემთა ტიპისაგან დამოკიდებულებით (იხ. სურათი 2-13:).

```
MoveVariant
                                Default value Comment
                    Data type
 + Input
 - Output
 Outinteger
                    int
                                        Integer data
      OutReal
                    Real
 ....
                                         Real data
 • • OutMyType
                   "MyType"
                                        User defined PLC data type
 🕶 🕶 InOut
      InOutVariant.
                    Variant
                                         Variable data input
 - Fiemp
 - v Constant
      NO_CORRECT_DATA_TYPE Word
                                16#8084
 .
                             CASE TypeOf(#InOutVariant) OF // Check datatypes
Int: // Передача целого типа (Integer)
    #MoveVariant := MOVE_BLK_VARIANT (SRC := #InOutVariant,
                                           COUNT := 1,
                                           SRC INDEX := 0,
                                           DEST_INDEX := 0,
                                           DEST => #OutInteger);
Real: // Передача вещественного типа (Real)
    #MoveVariant := MOVE_BLK_VARIANT(SRC := #InOutVariant,
                                           COUNT := 1,
                                           SRC INDEX := 0,
                                           DEST INDEX := 0,
                                           DEST => #OutReal);
МуТуре: // Передача собственного типа (МуТуре)
      #MoveVariant := MOVE_BLK_VARIANT(SRC := #InOutVariant,
                                           COUNT := 1,
                                           SRC INDEX := 0,
                                           DEST INDEX := 0,
                                           DEST => #OutMyType);
ELSE // Ошибка, тип данных не определен
      #MoveVariant := WORD TO INT(#NO CORRECT DATA TYPE);
      // 80B4: Код ошибки MOVE_BLK_VARIANT: Тип данных не соответствует
END CASE;
```

სურათი 2-13: ფუნქცია "MoveVariant"-ის ფორმალური პარამეტრები

შენიშვნა

თუ თქვენ გჭირდებათ გადააკოპიროთ არასტრუქტურიზებული ტეგების VARIANT მნიშვნელობები, მაშინ შეგიძლიათ ასევე გამოიყენოთ VariantGet ნაცვლად MOVE_BLK_VARIANT-ისა (იხ. თავი 2.9.3 VARIANT ინსტრუქციები).

2.9 ინსტრუქციები

2.9.1 CALCULATE

CALCULATE ინსტრუქციის დახმარებით, თქვენ შეგიძლიათ შეასრულოთ მათემატიკური გამოთვლები (მაგალითად (1N1+1N2) * 1N3)), რომლებიც არ არიან დამოკიდებული მონაცემთა ტიპებისაგან. მათემატიკური ფორმულის ჩაწერა ხდება ინსტრუქციების რედაქტორში (იხ. სურათი 2-14:).

"Edit "Cal	culate" instruction X
Cour :=	(IN1 + IN2) * IN3
CALCULATE Real	Example: (N1 = N2) * (N1 - IN2)
	Possible instructions:
OUT = 0H1 + H(2) *H3 #Rea[Jn1 - H1 OUT - #Rea[Out - #Rea[Jn2 - H2	And, Or, XOr, Swap, Not/Imc, +, -, *, f, Nod, Abs, Neg, Exp, **, Prac, Ln, Sin, ASin, Cos, ACos, Tan, ATan, Sqr, Sqt, Round, Cell, Floor, Trunc
stealin3-143 #	OK Cancel

სურათი 2-14: ინსტრუქცია Calculate ფორმულის რედაქტორით

შენიშვნა

უფრო დაწვრილებითი ინფორმაციის სანახავად თქვენ შეგიძლიათ მიაკითხოთ TIA Portal-ის Online ცნობას "CALCULATE" ინსტრუქციის შესახებ.

უპირატესობანი

- მათემატიკური ფორმულა იყენებს მხოლოდ ერთ ინსტრუქციას;
- დროის ეკონომია, რომელიც მიიღწევა მარტივი აწყობებიდან გამომდინარე.

თვისეზები

- მხარს უჭერს ბიტების თანმიმდევრობას, მთელ და ნამდვილ რიცხვებს;
- მხარს უჭერს სხვადასხვა მთემატიკურ ფუნქციებს, (ყველა მირითადი არითმეტიკული ოპერაციები, ტრიგონომეტრიული ფუნქციები, დამრგვალება, ლოგარითმული ფუნქციები და სხვ.);
- შესასვლელი პარამეტრების ცვალებადი რაოდენობა.

რეკომენდაციები

 ყოველთვის გამოიყენეთ ფუნქცია CALCULATE მატემატიკური გამოთვლებისათვის ნაცვლად ისეთი ინსტრუქციების მრავლჯერადი გამოძახებისა, როგორიცაა ADD, SUB და სხვ.

2.9.2 ინსტრუქცია MOVE

STEP-7-ში (TIA) არის შემდეგი სახის MOVE ინსტრუქციები. აქედან ინსტრუქცია MOVE_BLK_VARIANT S7-1200/1500-სთვის არის ახალი (იხ. ცხრილი 2-16:).

ინსტრუქცია	გამოყენება	თვისებები
MOVE	მნიშვნელობათა კოპირება მასივის კოპირება	 შესასვლელი პარამეტრის შემცველობის კოპირება IN გამოსასვლელ პარამეტრში OUT; შესასვლელი და გამოსასვლელი პარამეტრები უნდა იყოს ერთი ტიპის; პარამეტრები შეიმლება იყოს აგრეთვე სტრუქტურირებული (PLC data types); მთელი მასივისა და სტრუქტურის კოპირება.
MOVE_BLK	რამდენიმე. არეს კოპირება	 მასივის შემცველობის კოპირება სხვა მასივში; მოწაცემთა საწყისი და მიზწობრივი ტიპები უწდა ემთხვეოდეს ერთმაწეთს; მთელი მასივისა და სტრუქტურების კოპირება; მასივის რამოდეწიმე ელემეწტის კოპირება სტრუქტურებით, დაწყებული გარკვეული ელემეწტიდაწ.
UMOVE_BLK	მასივის კოპირება უწყვეტად	 მასივის შემცველობის კონსიტეტური კოპი- რება გამომახების კოპირების შეწყვეტის გარეშე OB წყვეტის დამუშავებაში; საწყისი და მიზნობრივი მონაცემთა ტიპები უნდა ემთხვეოდეს ერთმანეთს.
MOVE_BLK_VARIANT (S7-1500 და S7-1200, დაწყებული : FW4.1-დან)	მასივის კოპირება	 ერთი ანდა რამდენიმე სტრუქტურირებული ტეგის (PLC data types) კოპირება; მონაცემთა ტიპების გარკვევა მუშაობის დროს; დეტალური ინფორმაცია შეცდომაზე; სასევე ცალკე მხარდაჭერილია ელემენტარული და სტრუქტურირებული ტიპები, PLC data types, მასივები და DB მასივები
გადასვლა მიმდევრობითში (S7-1500 და S7-1200, დაწყებული FW4.1)	სტრუქტურ. მონაცემების კოპირება ბაიტურ მასივში	 მოწაცემთა რამდენიმე ჩანაწერი შეიძლება ჩაწერილი იყოს ერთ ბაიტურ მასივში და გაგზავნილ იქნას სხვა მოწყობილობებზე; შესასვლელი და გამოსასვლელი პარამეტრები შეიძლება გადაცემულ იქნას როგორც Variant.
გამოსვლა მიმდევრობითიდან (S7-1500 და S7-1200, დაწყებული FW4.1-დან)	ბაიტიური მასივის კოპი- რება ერთ ან რამოდენიმე სტრუქტურაში	 - გამოყენება I-Device: I-Device იღებს რომელიღაც ჩანაწერებს შესასვლელ არეში, რომლებიც გადაკოპირებული იქნებიან სხვადასხვა სტრუქტურებში; - რამდენიმე ჩანაწერი შეიძლება იყოს კომბინირებული ბაიტების სხვადასხვა მასივში. გამოსასვლრელი თანმიმდევრობიდან შესაძლებლობას გვძლევს დავაკოპიროთ იგი სხვადასხვა, სარლქბური.

ცხრილი 2-16: გადადგილების ინსტრუქციები



სურათი 2-15: მიმდევრობითში გადატანისა და მიმდევრობითიდან გამოტანის ინსტრუქციები (S7-1500 და S7-1200, დაწყებული FW4.1-დან)

რეკომენდაციები

- თქვენ უნდა გაიაზროთ განსხვავება MOVE, MOVE_BLK და MOVE_BLK_VARIANT-ს შორის:
 - გამოიყენეთ ინსტრუქცია MOVE მთელი სტრუქტურის კოპირებისთვის;
 - გამოიყენეთ ინსტრუქცია MOVE_BLK (ARRAY) მასივის ნაწილების კოპირებისთაის მონაცემთა ცნობილი ტიპებით;
 - გამოიყენეთ ინსტრუქცისა MOVE_BLK_VARIANT, მხოლოდ მაშინ, თუ გინდათ კოპირება (ARRAY) მასივის ნაწილის, რომლის მონაცემთა ტიპი ცნობილია მხოლოდ პროგრამის მუშაობის პროცესში.

შენიშვნა

UMOVE_BLK: კოპირების პროცესი არ შეიძლება შეწყვეტილ იქნას ოპერაციული სისტემის სხვა ამოცანით. ამიტომ, CPU-ს რეაქციის დრო შეიძლება გაიზარდოს მომუშავე "Copy array without interruption" ინსტრუქციის დროს (მასივის კოპირება წყვეტის გარეშე);

MOVE ინსტრუქციების შესახებ უფრო დაწვრილებითი ინფორმაციის მიღებისათვის მიაკითხეთ Online დახმარებას TIA Portal-ში.

შენიშვნა

თქვენ შეგიძლიათ მოძებნოთ დამატებითი ინფორმაცია შემდეგ საკითხებზე: როგორ შევასრულოთ მეხსიერების არეების და სტრუქტურიზებული მონაცემების კოპირება ერთი მონაცემების ბლოკიდან მეორეში STEP 7 (TIA Portal)-ში? https://support.industry.siemens.com/cs/ww/en/view/42603881

2.9.3 VARIANT ინსტრუქციები (S7-1500 და S7-1200, დაწყებული FW4.1-დან) (იხ. ცხრილი 2-17:).

ინსტრუქციები	შენიშვნები	თვისებები						
ინსტრუქციები MOVE								
VariantGet	მნიშვნელობათა წაკითხვა	ასეთი ინსტრუქცია საშუალებას გამლევთ თქვენ წაიკითხოთ ტეგის მნიშვნელობა, რომელიც აკითხავს VARIANT-ს.						
VariantPut	მნიშვნელობათა ჩაწერა	ასეთი ინსტრუქცია საშუალებას გაძლევთ თქვენ ჩაწეროთ ტეგის მნიშვნელობა, რომელიც აკითხავს VARIANT-ს.						
სია								
CountOfElements	ელემენტების გადათვლა	ასეთი ინსტრუქციით თქვენ შეგიძლიათ მიიღოთ ელემენტების რაოდენობა მასივში VARIANT ტიპის ცვლადისგან						
შედარების ინსტრ	აუქ ციები							
TypeOf() მხოლოდ (SCL)	მონაცემთა ტიპის განსაზღვრა	გამოიყენეთ ეს ინსტრუქცია მონაცემთა ტიპის განსაზღვრისათვის ცვლადის ტიპიდან VARIANT						
TypeOfElements() მხოლოდ (SCL)	მასივის მონაცემთა ტიპის განსაზღვრა	გამოიყენეთ ეს ინსტრუქცია მასივის ელემენტების მონაცემთა ტიპის გარკვევისათვის VARIANT ტიპის ცვლადიდან						
გარდაქმნის ინსტ	რუქციები							
VARIANT_TO_DB_ANY მხოლოდ (SCL)	მონაცემთა ბლოკის ნომრის განსაზღვრა	ამ ინსტრუქციით თქვენ შეგიძლიათ მიიღოთ PLC data type ნომერი ეკზემპლიარული ბლოკისთვის, სისტემური მონაცემთა ტიპიდან ან DB - დან						
DB_ANY_TO_VARIANT მხოლოდ (SCL)	Variant ტეგის შექმნა მონაცემთა ბლოკისაგან	ამ ინსტრუქციით თქვენ შექმნით ტეგს VARIANT ეკზემპლიარული მონაცემთა ბლოკიდან PLC data type ტიპით, მონაცემთა სისტე- მური ტიპით ანდა DB მასივით.						

ცხრილი 2-17: ინსტრუქციები მონაცემთა ტიპისათვის VARIANT

შენიშვნა

უფრო დაწვრილებითი ინფორმაციის მიღებისათვის VARIANT ინსტრუქციების შესახებ, შეგიძლიათ მიაკითხოთ TIA Portal-ის online დახმარებას.

2.9.4 RUNTIME

"RUNTIME" ინსტრუქციის გამოყენებით, თქვენ შეგიძლიათ შეაფასოთ მთლიანი პროგრამის, ცალკეული ბლოკების ანდა ბრძანებების თანმიმდევრობების შესრულების რეჟიმი. თქვენ შეგიძლიათ გამოიძახოთ აღნიშნული ინსტრუქცია SCL - ში (S7-1200/S7-1500)-თვის და STL-ში (S7-1500)-თვის.

შენიშვნა

თქვენ შეგიძლიათ მონახოთ დამატებითი ინფორმაცია შემდეგი საკითხების ირგვლივ: როგორ განვსაზღვროთ პროგრამის ანდა მისი ცალკეული ზლოკების მუშაობის დრო (S7-1200/S7-1500)-ში?

https://support.industry.siemens.com/cs/ww/en/view/87668055

2.10 სიმბოლიკა და კომენტარები

2.10.1 პროგრამის რედაქტორი

უპირატესობები

სიმბოლიკისა და კომენტარების გამოყენების შემთხვევაში, პროგრამის კოდი გაგებისათვის თქვენს პროგრამაში იქნება გაცილებით უფრო მარტივი. მთელი სიმბოლიკა პროგრამულ კოდთან ერთად პროგრამის ჩაწერის პროცესში შეინახება კონტროლერში, რაც შესაძლებლობას გვაძლევს შევასრულოთ სისტემის მომსახურება offline პროექტებში.

რეკომენდაციები

- კოდის წაკითხვის გაუმჯებესების თვალსაზრისით ყოველტვის გამოიყენეთ კომენტარები პროგრამებში;
- დაამუშავეთ პროგრამები იმგვარად, რომ ისინი იყოს გასაგები თქვენი კოლეგებისათვისაც.

შემდეგ მაგალითში, თქვენ შეძლებთ გაეცნოთ დამხმარე ინსტრუმენტებს რედაქტორში პროგრამის კომენტირებისათვის.

A N -0 -0 - 0				1 0 75 111
нй нй 🕾 🔬 😝 🖽	2 2 2 4 C	🗏 🕼 😋 👀 ((프 9월 😵 📜	= 0, IB
Interface				
Name	Data type	Defaul	t value Retain	Accessible f
1 • Input		_		
2 (1)***		1		
Information:	(2)		
 Network 1: Star 		/		
- Hethory II Sta	it engine			
start the engine				
	50.1			500.0
"Activate_1 3	Activate_2*			"Start"
HH	4 H			()]
 Network 2: Sor 	ed control			
Call Average black to	ed_como			
Call function block to	set speed and acces	enation		
	#Engine s	peed.	SR by speed	1.
	instand	e	and	
	\$282		acceleration	\frown
	"Engine_sp	peed"		(4)
	EN	ENO		~
0.0	Speed			
0.0	Acceleration			
0.0	Accaleration			

სურათი 2-16: კომენტარები სამომხმარებლო პროგრამაში (LAD)

მაგალითი

შემდეგ გამოსახულებაში თქვენ შეგიძლიათ ნახოთ კომენტარიების შექმნის ვარიანტი LAD რედაქტორში (ისევე როგორც FBD-სთვის) (იხ. სურათი 2-16:). კომენტარიების შესაძლო ვარიანტები:

- 1. კომენტარები ბლოკთან მიმართებაში;
- 2. სეგმენტის სათაურის კომენტარები;
- 3. სეგმენტის კომენტარები;
- ინსტრუქციების, ბლოკების და ფუნქციების კომენტარები (გახსნა, დახურვა და სხვ.).

SCL და STL დაპროგრამების ენებზე ნიშნის // დახმრებით თქვენ შეგიძლიათ დააკოკომენტაროთ ერთი სტრიქონი.

მაგალითი

Filling level := Radius * Radius * PI * height; // **შევსების დონის გამოთვლა**

შენიშვნა

თქვენ შეგძლიათ მონახოთ დამატებითი ინფორმაცია შემდეგი საკითხების ირგვლივ: რატომ არ ხდება ასასახი ტექსტების, სათაურების და კომენტარების ასახვა პროექტის გახსნის შემდეგ STEP 7 (TIA Potrtal) რედაქტორში? <u>https://support.industry.siemens.com/cs/ww/en/view/41995518</u>

2.10.2 კომენტარები დაკვირვებათა ცხრილებში

უპირატესობები

 დაკვირვების ცხრილებშიც შესაძლებელია ასევე კომენტარების შექმნა უფრო მეტად სტრუქტურირებული სახის მისაღებად.



სურათი 2-17: Watch table with comment lines

რეკომენდაციები

- ყოველთვის გამოიყენეთ კომენტარების სიმბოლოები თქვენი დაკვირვების ცხრილის სტრუქტურირებისათვის;
- შეძლებისდაგვარად, გაუკეთეთ კომენტარები თვითოეულ ტეგს.

მაგალითი (იხ. სურათი 2-17:).

2.11 სისტემური კონსტანტები

S7/300/400 კონტროლერებში აპარატული და პროგრამული კომპონენტების იდენტიფიკაცია სრულდება ლოგიკური ანდა დიაგნოსტიკური მისამართების მიკუთვნებით.

S7-1200/1500 კონტროლერებში იდენტიფიკაცია სრულდება სისტემური კონსტანტების დახმარებით.

S7-1200/1500 კონტროლერების ყველა აპარატულ და პროგრამულ კომპონენტებს (მაგ. ინტერფეისები, მოდულები, OB,...) აქვთ თავიანთი სისტემური კონსტანტები. სისტემური კონსტანტები ავტომატურად იქმნება მოწყობილობების კონფიგურაციის პროექტირების დროს ცენტრალურ და განაწილებულ პერიფერიებში.

უპირატესობანი

 თქვენ შეგიძლიათ მიიღოთ ხელწვდომა სისტემურ კონსტანტებთან მუდულის სახელის გამოყენებით, ნაცვლად აპარატული იდენტიფიკაციისა.

რეკომენდაციები

 პროგრამის შექმნის გამარტივების მიზნით მიაკუთვნეთ სახელი მოდულს მისი დანიშნულების შესაბამისად.

მაგალითი

შემდეგ მაგალითში, თქვენ შეგიძლიათ დაინახოთ თუ როგორ გამოიყენება სისტემური კონსტანტები სამომხმარებლო პროგრამაში (იხ. სურათი 2-18:).



სურათი 2-18: "სისტემური კონსტანტები" სამომხმარებლო პროგრამაში

- 1. კონტროლერის სისტემური კონსტანტები შეიძლება ინახოს კატალოგში "PLC tags Default tag table" (ტეგები PLC-ტეგების ცხრილი სიჩუმით);
- სისტემური კონსტანტები მოთავსებულია ცალკე ჩანართში "Default tag table"-ში (ტეგების ცხრილი სიჩუმით);
- ამ მაგალითში სიმბოლური სახელი "Robot_arm_left" დანიშნული იყო მოდულისათვის DI. თქვენ შეგიძლიათ მოძებნოთ მოდული სისტემური კონსტანტების ცხრილში. სამომხმარებლო პოგრამაში "Robot_arm_left" ურთიერთკავშირშია დიაგნოსტიკური ბლოკის "GET_DIAG" შესასვლელთან.

შენიშვნა

ყოველი ცალკეული მოწყობილობაისთვის სისტემური კონსტანტების სწრაფი მონახისთვის უნდა გაიხსნას რედაქტორი "Device configuration" (მოწყობილობის კონფიგურაცია),

0 1	2 3	4 5	6	7	15	23			
Rail_0					:		R		
							24		
					16				
- ee i					:				
<		п					Ξ		
PLC_2 [CPU 1516-3 PN/	DP]								
General IO tags	System cor	istants	Texts						
Name	Туре		Hardwa	re ident	ifier	Comr	men		
PIP OB Servo	Pip		32768						
OB_Main	OB_PCYCLE		1						
OB_CyclicInterrupt	OB_Cyclic		30						
PLC_2[MC]	Hw_SubMod	Hw_SubModule			51				
PLC_2[Common]	Hw_SubMod	ule	50						
PLC_2[Display]	Hw_SubMod	Hw_SubModule							
PLC_2[Exec]	Hw_SubMod	Hw_SubModule			52				
PLC_2	Hw_SubMod	Hw_SubModule		49					
DP_interface_1	Hw_Interface	Hw_Interface			60				
PROFINET_interface_1	Hw_Interface	Hw_Interface		64					
PROFINET_interface_2	Hw_Interface	2	72						
Port_1[PN](1)	Hw_Interface	2	73						
Port_1[PN]	Hw_Interface		65						

შენიშვნა

თქვენ შეგიძლიათ იპოვოთ დამატებითი ინფორმაცია შემდეგ საკითხებზე: S7/1200/1500 კონტროლერებისთვის რისთვის არის საჭირო სისტემური კონსტანტები STEP 7 (TIA Portal)-ში?

https://support.industry.siemens.com/cs/ww/en/view/78782835
2.12 სამომხმარებლო კონსტანტები

სამომხმარებლო კონსტანტების დახმარებით თქვენ შეგიძლიათ შექმნათ თქვენი მუდმივი მნიშვნელობები. ძირითადად, სამომხმარებლო პროგრამაში გამოიყენება ლოკალური კონსტანტები OB, FC, FB ბლოკებისათვის და ასევე გლობალური კონსტანტები კონტროლერის მთელი პროგრამისათვის.

უპირატესობანი

- სამომხმარებლო პროგრამები შეიძლება გამოყენებულ იქნას მთელ პროგრამაში მუდმივი მნიშვნელობების შეცვლისათვის გლობალურად ანდა ლოკალურად;
- სამომხმარებლო კონსტანტების დახმარებით პროგრამა შეიძლება დაწერილ იქნას უფრო ადვილად წასაკითხი სახით.

თვისებები

- ლოკალური სამომხმარებლო კონსტანტების გამოცხადება ხდება ბლოკის ინტერფეისში;
- გლობალური სამომხმარებლო კონსტანტების გამოცხადება ხდება "PLC tags"-ში (ტეგები PLC);
- სამომხმარებლო კონსტანტები შეიძლება წაკითხულ იქნას მხოლოდ
 სამომხმარებლო პროგრამების მიერ;
- დაცული ბლოკებისათვის (know-how protected მინიჭებული თვისებით)
 სამომხმარებლო კონსტანტები დამალულია.

რეკომენდაციები

- გამოიყენეთ სამომხმარებლო კონსტანტები პროგრამის მოქნილობისა და კითხვადობის გაუმჯობესების მიზნით, კონკრეტულად:
 - შეცდომათა კოდების;
 - CASE ინსტრუქციების;
 - გარდაქმნის კოეფიციენტების;
 - ნატურალური კონსტანტების ...

ლოკალური სამომხმარებლო კონსტანტები CASE ინსტრუქციებისათვის (იხ. სურათი 2-19:); გლობალური სამომხმარებლო კონსტენტები (იხ. სურათი 2-20:).



სურათი 2-19: ლოკალური სამომხმარებლო კონსტანტები CASE ინსტრუქციებისათვის

		🚭 Tags 🛛 🛛	User constants	🖉 Syst	em co	nstants
\exists^0	6	B 5 👘		-		
PL	.C ta	igs				
		Name	Tag table	Data type	Value	Comment
1		GLOBAL_MN	Default tag table	Int	5	
2		GLOBAL_MAX	Default tag table	Int	100	
3		«Add news»				



შენიშვნა

უფრო დაწვრილებითი ინფორმაცია სამომხმარებლო კონსტანტების გამოყენების შესახებ აღწერილია FAQ-ში: როგორ შეიძლება გარდავქმნათ ტეგი STEP 7 (TIA Portal)-ში? https://support.industry.siemens.com/cs/ww/en/view/61928891

2.13 კონტროლერების ტეგებისა და HMI ტეგებისთვის შიგა ბმულების ID

STEP 7, WinCC, Startdrive, Satety და სხვა პროგრამები ინტრგრირებულნი არიან TIA Potrtal-ის დამუშავების გარემოს საერთო ბაზაში. მონაცემთა ცვლილებები ავტომატურად აისახება სამომხმარებლო პროგრამის დანარჩენ არეებში, მიუხედავად იმისა, თუ სად იყო შესრულებული ცვლილებები: კონტროლერში, პანელში თუ ამმრავში. ამით უზრუნველყოფილი ხდება მონაცემთა მთლიანობა.

ტეგის შექმნის დროს, TIA Potrtal-ი ავტომატურად ქმნის უნიკალურ ბმულისათვის განკუთვნილ ID-ს. თქვენ არ შეგეძლებათ დაინახოთ ანდა დააპროგრამოთ ეს ID, ეს პროცედურა არის შიგა პროცედურა. ტეგების ცვლილების დროს (მისამართი), ბმულისათვის განკუთვნილი ID რჩება უცვლელი. ქვემოთ გამოსახულებაზე სქემატურად ნაჩვენებია ბმულისათვის განკუთვნილი ID-ს შესაბამისობა რაიმე ინფორმაციასთან (იხ. სურათი 2.21:).

PLC_1 HMI_1						
PLC-ს სიმბ. სახელი	აბსოლ <mark>უტ.</mark> მისამართი	PLC-ის მიგა ID	HMI-ს შიგა ID	სიმბოლ უ რ სახელ. HMI	ხელწვდ(ტიპი	PLC-სთან შეერთება
Motor_1	10.0	000123	009876	Motor_1	<symbolic access></symbolic 	PLC_1
Valve_2	Q0.3	000138	000578	Valve_2	<symbolic access></symbolic 	PLC_1

სურათი 2-21: შიგა ბმულისათვის განკუთვნილი ID PLC-სა და HMI-ისთვის

შენიშვნები

ID შეცვლილ იქნება თუ ...

- შეცვლილი იქნება სახელი;
- შეცვლილ იქნება ტიპი;
- შეცვლილ იქნება ტეგი.

უპირატესობანი

- თქვენ შეგძლიათ დანიშნულება შეუცვალოთ ტეგებს შიგა კავშირების შეუცვლელად. კომუნიკაციები კონტროლერებს, HMI-ებს და ამძრავებს შორის რჩება შეუცვლელი;
- სიმბოლური სახელის სიგრძე არ მოქმედებს კომუნიკაციურ დატვირთვაზე კონტროლერსა და HMI-ს შორის.

თვისებები

თუ თქვენ შეცვლით PLC-ის ტეგების მისამართებს, თქვენ აუცილებლად დაგჭირდებათ კონტროლერის გადატვირთვა. არ არის აუცილებელი HMI მოწყობილობების გადატვირთვა, გამომდინარე შიგა სისტემური მისამართებისგან ბმულებისათვის განკუთვნილი ID-ებით. (იხ. სურათი 2-22).



სურათი 2-22. მისამართების ცვლილება ანდა სტრიქონის დამატება

2.14 რეჟიმი STOP შეცდომების წარმოშობის დროს

S7-300/400-გან განსხვავებით, S7-1200/1500 კონტროლერებში "STOP" რეჯიმში გადასვლა ხდება მხოლოდ ზოგიერთ შემთხვევაში.

TIA Portal-ში კონსისტენტული შემოწმებებისაგან გამომდინარე, "STOP" რეჯიმში გადასვლა S7-1200/1500 კონტროლერებისათვის უმეტეს შემთხვევაში შესამლებელია თავიდან იქნას აცილებული. კონსტისტენტულობაზე პროგრამული ბლოკების შემოწმება ხდება კომპილიაციის დროს TIA Portal-ში. აღნიშნული მექანიზმი თავიდან იცილებს S7-1200/1500 კონტროლერების გაჩერებას ზოგიერთი სახის შეცდომების აღმოჩენის დროს, განსხვავებით მათი წინამორბედებისაგან.

უპირატესობანი

S7-1200/1500 კონტროლერები "STOP" რეჟიმში გადადიან მხოლოდ სამ შემთხვევაში, რაც ამარტივებს შეცდომის დამუშავების დაპროგრამებას.

თვისებები (იხ. ცხრილი 2-18:)

ცხრილი 2-18: S7-1200/1500 კონტროლერების რეაქცია შეცდომებზე

	შეცდომა	S7-1200	S7-1500
1.	ერთმაგი გადაჭარბება ციკლის დროზე	RUN	STOP, თუ OB80 არ არის
2.	,ორმაგი გადაჭარბება ციკლის დროზე	STOP	STOP
3.	დაპროგრამების შეცდომა	RUN	STOP, თუ OB121 არის

OB შეცდომების დამუშავება

- OB80 "Time error interrupt" (ციკლის დროის გადამეტება) გამოიძახება ოპერაციული სისტემის მიერ, როდესაც გადაჭარბებულ იქნა ციკლის მაქიმალური დრო;
- OB121 "Programming error" (დაპროგრამების შეცდომა) გამოიძახება ოპერაციული სისტემის მიერ, პროგრამაში შეცდომების წარმოშობის დროს.

შეცდომების ყოველი წარმოშობის დროს ავტომატურად ხდება ჩაწერა დიაგნოსტიკურ ბუფერში.

შენიშვნა

S7-1200/1500 კონტროლერებში არის აგრეთვე სხვა OB შეცდომების დამუშავებისთვის (დიაგნოსტიკური მოვლენები, მოწყობილობის მწყობრიდან გამოსვლა და სხვ.). უფრო დაწვრილებითი ინფორმაცია S7-1200/1500 კონტროლერების რეაქციაზე შეცდომებზე, თქვენ შეგიძლიათ მოძებნოთ TIA Portal-ის online დახმარებაში განყოფილებაში "Events and OB" (ხდომილებები და OB).

3. შესავალი დაპროგრამებაში

დაპროგრამების ენები

სამომხმარებლო პროგრამების შექმნისათვის გამოიყენება რამდენიმე სახის დაპროგრამების ენა. თვითოეულ ენას აქვს თავისი უპირატესობა რომლებიც შესაძლებელია შემდეგ გამოყენებულ იქნას პროგრმებისაგან დამოკიდებულებით. თვითოეული ბლოკი სამომხმარებლო პროგრამაში შესაძლებელია შექმნილ იყოს დაპროგრამების ნებისმიერ ენაზე. ცხრილში 1.1: მოყვანილია კორპორაცია სიმენსის ახალი თაობის პროგრამირებადი ლოგიკური კონტროლერების S7-1200 და S7-1500 დაპროგრამებისთვის გამოსაყენებელი დაპროგრამების ენები:

	80000001-1.000	0.0180309000 90900
დაპროგრამების ენა	S7-1200	\$7-1 500
Ladder (LAD)	✓	\checkmark
Function block diagram (FBD)	✓	\checkmark
Structured control language (SCL)	✓	\checkmark
Graph	×	\checkmark
Statement list (STL)	×	\checkmark

ცხრილი 1-1: დაპროგრამების ენები

3.1.1 ოპერაციული სისტემა და სამომხმარებლო პროგრამა

SIMATIC ლოგიკური კონტროლერები შეიცავს ოპერაციულ სისტემას და სამომხმარებლო პროგრამას:

- ოპერაციული სისტემა ახორციელებს კონტროლერში ყველა იმ ფუნქციისა და პროცესის მართვას, რომლებიც არ არის დაკავშირებული მართვის გარკვეულ ამოცანებთან (მაგ. რეესტრის დამუშავება, პროცესის სახის განახლება, სამომხმარებლ პროგრამის დამუშავება, შეცდომათა დამუშავება, მეხსიერების მართვა და ა.შ.). ოპერაციული სისტემა ეს - კონტროლერის ჩაშენებული პროგრამული ნაწილია.
- სამომხმარებლო პროგრამა შედგება ბლოკებისაგან, რომლებიც საჭიროა ავტომატიზაციის ამოცანების გადასაწყვეტად. სამომხმარებლო პროგრამა აიგება ბლოკებისაგან და შემდგომში ჩაიტვირთვება კონტროლერში (იხ. სურათი 3-1:).



სურათი 3-1 : ოპერაციული სისტემა და სამომხმარებლო პროგრამა

SIMATIC კონტროლერებისათვის, სამომხმარებლო პროგრამა ყოველთვის სრულდება ციკლურად. ციკლური მაორგანიზებელი ბლოკი OB ("Main") უკვე იარსებებს "Program blocks" (პროგრამული ბლოკები) საქაღალდეში აღნიშნული კონტროლერის დამატების შემთხვევაში STEP-7-ში. ბლოკის დამუშავება და გამოძახება ხდება კონტროლერის მიერ უსასრულო ციკლში.

3.2. პროგრამული ბლოკები

STEP-7 (TIA Portal)-ში გამოყენებული ბლოკის ტიპები დარჩნენ STEP-7-ის წინა ვერსიიდან. აქ შედის:

- მაორგანიზებელი ბლოკები;
- ფუნქციონალური ბლოკები;
- ფუნქციები;
- მონაცემთა ბლოკები.

STEP-7-ის გამოცდილი მომხმარებლები ალბათ მათთან უკვე გაცნობილნი არიან, ხოლო ახალბედები ადვილად აითვისებენ მათ.

უპირატესობანი

- თქვენ შეგეძლებათ შექმნათ პროგრამის სტრუქტურა სხვადასხვა ტიპის ბლოკების გამოყენებით;
- სტრუქტურული დაპროგრამების დროს, თქვენ იღებთ შესაძლებლობას იმუშაოთ მრავალჯერადი გამოყენების ბლოკებთან ამ პროექტის შიგნით ანდა გარეთ. ასეთი ბლოკები განირჩევა მხოლოდ პარამეტრების ნაკრებებით (იხ. თავი 3.8.2 ბლოკების მანმეორებითი გამოყენება).
- თქვენი პროექტი ანდა დანადგარი გახდება უფრო გამჭირვალე. შეცდომები სამუშაო სადგურებში მარტივად აღმოსაჩენი, გასაანალიზებელი და გამოსწორებადი. თქვენი პროგრამის თანდევნა და კორექტირება ხდება უფრო იოლად საკეთებელი.

რეკომენდაციები

- მოახდინეთ თქვენი ავტომატიზაციის ამოცანის სტრუქტურირება;
- დაყავით ხაზის საერთო ფუნქციონალი უფრო მცირე პროცედურებად მანამ, სანამ უკვე ვეღარ შეძლებთ მრავალჯერადი გამოძახების ფუნქციის რეალიზებას სხადასხვა პარამეტრებით;
- განსაზღვრეთ ინტერფეისები ფუნქციონალურ კვანძებს შორის. განსაზღვრეთ უნიკალური ინტერფეისები იმ ფუნქციონალისთვის, რომელიც მოეწოდება "გარე კომპანიების მიერ".

ყველა მაორგანიზებელი ბლოკი, ფუნქციონალური ბლოკი და ფუნქცია შეიძლება დაპროგრამებულ იქნას შემდეგ ენებზე (იხ. ცხრილი 3-1:).

დაპროგრამების ენა	S7-1200	S7-1500
Ladder (LAD)	\checkmark	\checkmark
Function block diagram (FBD)	✓	\checkmark
Structured control language (SCL)	\checkmark	\checkmark
Graph	×	\checkmark
Statement list (STL)	*	\checkmark

ცხრილი 3-1: დაპროგრამების ენები

3.2.1 მაორგანიზებელი ბლოკები (OB) (იხ. სურათი 3-2:)



სურათი 3-2: ფანჯარა "Add new block" (ახალი ბლოკის დამატება) (OB)

მაორგანიზებელი ბლოკი (OB) წარმოადგენს ინტერფეისს ოპერაციულ სისტემასა და მომხმარებლის პროგრამას შორის. მისი გამოძახება ხდება ოპერაციული სისტემის მიერ და მართავს შემდეგ პროცესებს:

- კონტროლერის ქცევას გაშვებისას;
- პროგრამის ციკლური დამუშავებას;
- პროგრამის წყვეტების დამუშავებას;
- შეცდომების დამუშავებას.

კონტროლერის სხვადასხვა ტიპისაგან დამოკიდებულებით ხელმისაწვდომია სხვადასხვა რაოდენობის ტიპის OB.

თვისებები

- OB-ს გამოძახდება ხდება კონტროლერის ოპერაციული სისტემის მიერ;
- პროგრამაში შეიძლება შექმნილ იყოს რამდენიმე მთავარი OB (Main). ასეთი

OB-ების დამუშავება ხდება მიმდევრობით მათი OB ნომრების ზრდის შესაბამისად (იხ. სურათი 3-3:).



სურათი 3-3: რამდენიმე მთავარი OB (Main)-ის გამოყენება.

რეკომენდაციები

- დაყავით ქვეპროგრამების სხვადასხვა ნაწილების გამოძახებები, რომლებიც შეიძლება გადატანილ იქნას ერთი კონტროლერიდან მეორეზე, რამდენიმე მთავარ მაორგანიზებელ ბლოკად OB (Main).
- თავი აარიდეთ კავშირებს რამდენიმე მთავარ OB (Main)-ებს შორის. ისინი გამოყენებულ უნდა იქნას ერთმანეთისაგან დამოუკიდებლად. თუ თქვენ მაინც ახორციელებთ რამოდენიმე მთავარ OB-ს შორის ინფორმაციის გაცვლას, გამოიყენეთ ამისათვის გლობალური DB-ები (იხ. თავი 4.2 გადასვლა მერკერული არედან მონაცემთა გლობალურ ბლოკებზე).



სურათი 3-4: პროგრამის ნაწილების შენახვა პროექტის ბიბლიოთეკაში

 განაცალკავეთ პროგრამის ყველა ნაწილი, რომლებიც ურთიერთობენ ერთმანეთთან ცალკეულ საქაღალდეებად და შინახეთ ისინი განმეორებით გამოყენებისათვის საპროექტო ანდა გლობალურ ბიბლიოთეკაში (იხ. სურათი 3-4:).

შენიშვნა

თქვენ შეგიძლიათ მონახოთ დამატებითი ინფორმაცია შემდეგ საკითხზე: რომელი მაორგანიზებელი ბლოკები შეიძლება იქნას გამოყენებულ STEP-7 (TIA Portal)-ში? <u>https://support.industry.siemens.com/cs/ww/en/view/40654862</u>

3.2.2 ფუნქციები (FC) (იხ. სურათი 3-5:)

Add new block					×
Name: Block_1	_		_		
Corganization block	Language: Number:	LAD manual automatic	(*)]\$]		
Function block	Description: Functions are r	code blocks or subrout	ines without dedi	cated memory.	
Function					
Deta block	more_				
> Additional infor	mation				
Add new and gper	ŧ.			ок	Cancel

სურათი 3-5: "Add new block" ფანჯარა (ახალი ბლოკის დამატება) (FC)

ფუნქცია (FC) ეს ისეთი ბლოკია, რომელსაც არ გააჩნია მეხსიერება. სწორედ ამიტომ ასეთი ბლოკების პარამეტრების მნიშვნელობები შეინახება ბლოკის შემდგომ გამოძახებამდე.

თვისეზები

- FC ეს არის ციკლური შენახვის ბლოკები;
- დროითი ტეგები არ არის განსაზღვრულნი არაოპტიმიზირებული ბლოკების გამოძახების შემთხვევებში. ოპტიმიზირებულ ბლოკებში, ტეგები ყოველთვის იღებენ მნიშვნელობებს სიჩუმით (S7-1500 და S7-1200, დაწყებული Firmware V4). ამგვარად, ასეთი ქცევა თავიდან იცილებს შეცდომების წარმოშობას;
- FC-ში მონაცემთა შენახვისათვის, შესაძლებელია მონაცემთა გლობალური ბლოკების გამოყენება;
- FC-ებს შეიძლება ჰქონდეთ რამდენიმე გამოსასვლელი;
- ფუნქციის მნიშვნელობა შეიძლება ხელმეორედ გამოყენებულ იქნას SCL-ში როგორც ფორმულა.

რეკომენდაციები

- გამოიყენეთ ფუნქციები იმ დანართებში, რომლებიც რამდენჯერმე გამოიძახება პროგრამის სხვადასხვა ნაწილებში;
- ფუნქციის მნიშვნელობის განმეორებით გამოყენებისათვის, გამოიყენეთ SCL.
 <ოპერანდი> := <სახელი FC> (პარამეტრების სია);

მაგალითი

მომდევნო მაგალითში FC-ში დაპროგრამებულია მატემატიკური გამოსახულება. გამოთვლების რეზულტატი აღწერილია როგორც დაბრუნებითი მნიშვნელობა, რომელიც შემდგომში შეიძლება გამოყენებულ იქნას განმეორებით (იხ. ცხრილი 3-2:).



ცხრილი 3-2: ფუნქციის მნიშვნელობათა განმეორებითი გამოყენება

შენიშვნა

თქვენ შეგიძლიათ მოძებნოთ დამატებითი ინფორმაცია შემდეგი საკითხების შესახებ: რა რაოდენობის მაქსიმალური პარამეტრი შეიძლება განისაზღვროს ფუნქციისათვის STEP 7-ში (TIA Portal) S7-1200/S7-1500 CPU? https://support.industry.siemens.com/cs/ww/en/view/99412890 3.2.3 ფუნქციონალური ბლოკები (FB) (იხ. სურათი 3-6:)



სურათი 3-6: ფანჯარა "Add new block" (ახალი ბლოკის დამატება) (FB)

ფუნქციონალური ბლოკი (FB) - ეს არის მონაცემთა ციკლურად შემნახველი ბლოკი, რომლის მნიშვნელობები შეიძლება ხელმისაწვდომი იყოს ამ ბლოკის შემდგომი გამოძახების შემთხვევაში. მონაცემთა შენახვის რეალიზაცია ხორციელდება მონაცემთა ბლოკის ეკზემპლიარში.



თვისებები

- FB ეს ისეთი ბლოკია, რომელიც ახორციელებს მონაცემთა ციკლურ შენახვას;
- დროითი ტეგები არ არის განსაზღვრულნი არაოპტიმიზირებული ბლოკების გამოძახებისას. ოპტიმიზირებულ ბლოკებში ტეგები ყოველთვის იღებს მნიშვნელობებს სიჩუმით (S7-1500 და S7-1200, დაწყებული Firmware V4). ამგვარად, ასეთი მიდგომა თავიდან გვაცილებს შეცდომების წარმოშვებას;
- სტატიკური ცვლადები ინახავენ თავიანთ მნიშვნელობებს ბლოკის შემდეგ გამოძახებამდე.

რეკომენდაციები

 გამოიყენეთ ფუნქციონალური ბლოკები სტრუქტურიზებული სამომხმარებლო პროგრამების შესაქმნელად. ფუნქციონალური ბლოკი შეიძლება გამოძახებული იყოს რამდენჯერმე სამომხმარებლო პროგრამის სხვადასხვა ნაწილებში. ასეთი მექანიზმი ამარტივებს პროგრამის ხელმეორედ გამოსამახებელი ელემენტების პროგრამირებას;

 თუ კი ფუნქციონალური ბლოკების გამოძახება ხდება რამდენიმეჯერ პროგრამაში, მაშინ გამოიყენეთ ბლოკების ცალკეული ეკზემლიარები ანდა უმჯობესია მულტიეკზემპლიარები.

3.2.4. ეკზემპლიარები

ფუნქციონალური ბლოკის გამოძახების დროს აუცილებელია მითითებულ იქნას ეკზემპლიარული მონაცემთა ბლოკი. ის მონაცემები რომლებსაც ამუშავებს ფუნქციონალური ბლოკი შეინახება ეკზემპლიარულ DB-ში.

ეკზემპლიარული DB ყოველთვის შეიქმნება FB-ს აღწერილ ინტერფეისთან შესაბამისობაში და ცალკეული ცვლადები შეუძლებელია შეცვლილ იყოს თვითონ ეკზემპლიარულ DB-ში (იხ. სურათი 3-8:).



სურათი 3-8: FB-ს ინტერფეისის სტრუქტურა

ეკზემპლიარულ DB-ში ინახება შესასვლელები, გამოსასვლელები, შესასვლელ/გამოსასვლელები და სტატიკური ცვლადები. დროითი ცვლადები ინახება L სტეკში. L სტეკი აქტუალურია მხოლოდ მიმდინარე პროცესისათვის. ანუ დროითი ცვლადების ინციალიზაცია უნდა მოხდეს ყოველ ციკლში.

თვისებები

- FB-ს გამოძახების დროს აუცილებელია დაინიშნოს ეკზემპლიარული DB;
- ეკზემპლიარული DB იქმნება TIA Portal-ში ხელით კი არა, არამედ ავტომატურად FB-ს გამომახების დროს;
- ეკზემპლიარული DB-ს სტრუქტურა განისაზღვრება FB-ს ინტერფეისის შესაბამისად და შეიძლება შეცვლილ იყოს მხოლოთ მისი მეშვეობით.

რეკომენდაციები

შექმენით პროგრამები ისეთნაირად, რომ ეკზემპლიარული DB-ის მონაცემები
 შესაძლებელი იყოს შეიცვალოს მხოლოდ შესაბამისი FB-თი. ამის შესაბამისად,

თქვენ იძლევით გარანტიას, რომ ბლოკი გამოყენებული იქნება კონკრეტულად თავისი ამოცანის გადასაწყვეტად ნებისმიერ პროექტებში.

უფრო დაწვრილებითი ინფორმაციის მისაღებად მიაკითხეთ თავს 3.4 ბლოკის ინტერფეისი მონაცემთა გაცვლისათვის.

3.2.5 მულტიეკზემპლიარები

მულტიეკზემპლიარების დახმარებით გამოსამახებელ ფუნქციონალურ ბლოკებს შეუმლიათ შეინახონ თავიანთი ინფორმაცია, გამომმახებელი ფუნქციონალური ბლოკის ეკზემპლიარულ ბლოკში. ანუ, თუ ერთი ფუნქციონალური ბლოკის გამომახება ხდება სხვა ფუნქციონალურ ბლოკში, მაშინ ყველა მისი ინფორმაცია შეინახება უფრო მაღალი დონის FB-ს მონაცემთა ეკზემპლიარულ ბლოკში. გამოსამახებელი ბლოკის ფუნქციონალი რჩება უცვლელი.

შემდეგ გამოსახულებაზე ნაჩვენებია, თუ ერთ FB -ში როგორ გამოიყენება სხვა FB ("IEC Timer"). ყველა მონაცემი შეინახება მულტიეკზემპლიარულ DB-ში. ამგვარად, შეიძლება შეიქმნას ბლოკი დამოუკიდებელი საქმიანობისათვის დროის მიხედვით, მაგ. ტაქტური გენერატორი (იხ. სურათი 3-9:).



სურათი 3-9: მულტიეკზემპლიარი

უპირატესობანი

- განმეორებადი გამოყენების შესაძლებლობა;
- მრავალჯერადი გამოყენება;
- უფრო ადვილად წაკითხვადი პროგრამა უფრო მცირე რაოდენობის DB-თი;
- პროგრამის მარტივი კოპირება;
- პროგრამის სტრუქტურირების კარგი შესძლებლობა.

თვისებები

• მულტიეკზემპლიარები ეს მეხსიერების არეებია შიგნით ეკზემპლიარული DB-თი.

რეკომენდაციები

გამოიყენეთ მულტიეკზემპლიარები:

- DB ეკზემპლიარების შემცირების მიზნით;
- ადვილად წაკიკითხი პროგრამების შესაქმნელად, განმეორებითი გამოყენების შესამლებლობით;
- ლოკალური ფუნქციების დასაპროგრამებლად მაგალითად ტაიმერის, მთვლელის, ფრონტის გამოყოფის და სხვ;

მაგალითი

თუ თქვენ გჭირდებათ ტაიმერების ან მთვლელების ფუნქციები, მაშინ გამოიყენთ ბლოკები "IEC Timer" და "IEC Counter" ნაცვლათ SIMATIC-ის ტაიმერებისა და მთვლელებისა აბსოლუტური დამისამართებით. თუ არის ამის შესამლებლობა, მაშინ რეკომენდებულია გამოიყენოთ მულტიეკზემპლიარები. ეს მექანიზმი შესამლებლობას იმლევა მინიმუმამდე შევამციროთ მონაცემთა ბლოკების რაოდენობა სამომხმარებლო პროგრამაში (იხ. სურათი 3-10:).

🕶 🔄 Timer operations	
IEC Timers	
🛥 TP	Generate pulse
TON 🖶	Generate on-delay
TOF	Generate off-delay
TONR	Time accumulator
O -(TP)	Start pulse timer
0 -(TON)-	Start on-delay timer
() -(TOF)-	Start off-delay timer
O -(TONR)-	Time accumulator
-(RT)	Reset timer
SIMATIC Timers	
SIMATIC Timers	Assign pulse timer parameters and start
SIMATIC Timers S_PULSE S_PEXT	Assign pulse timer parameters and start Assign extended pulse timer parameters and start
SIMATIC Timers E S_PULSE E S_PEXT E S_ODT	Assign pulse timer parameters and start Assign extended pulse timer parameters and start Assign on-delay timer parameters and start
SIMATIC Timers S_PULSE S_PEXT S_ODT S_ODT S_ODTS	Assign pulse timer parameters and start Assign extended pulse timer parameters and start Assign on-delay timer parameters and start Assign retentive on-delay timer parameters and start
SIMATIC Timers S_PULSE S_PEXT S_ODT S_ODTS S_OFFDT	Assign pulse timer parameters and start Assign extended pulse timer parameters and start Assign on-delay timer parameters and start Assign retentive on-delay timer parameters and start Assign off-delay timer parameters and start
SIMATIC Timers	Assign pulse timer parameters and start Assign extended pulse timer parameters and start Assign on-delay timer parameters and start Assign retentive on-delay timer parameters and start Assign off-delay timer parameters and start Start pulse timer
SIMATIC Timers	Assign pulse timer parameters and start Assign extended pulse timer parameters and start Assign on-delay timer parameters and start Assign retentive on-delay timer parameters and start Assign off-delay timer parameters and start Start pulse timer Start extended pulse timer
SIMATIC Timers	Assign pulse timer parameters and start Assign extended pulse timer parameters and start Assign on-delay timer parameters and start Assign retentive on-delay timer parameters and start Assign off-delay timer parameters and start Start pulse timer Start extended pulse timer Start on-delay timer
SIMATIC Timers E S_PULSE E S_PEXT E S_ODT E S_ODTS E S_OFFDT O -(SP) O -(SE) O -(SD) O -(SS)	Assign pulse timer parameters and start Assign extended pulse timer parameters and start Assign on-delaytimer parameters and start Assign retentive on-delay timer parameters and start Assign off-delay timer parameters and start Start pulse timer Start extended pulse timer Start on-delay timer Start retentive on-delay timer

სურათი 3-10: IEC ტაიმერების ბიბლიოთეკა

შენიშვნა

თქვენ შეგიძლიათ მონახოთ დამატებითი ინფორმაცია შემდეგ საკითხებზე: როგორ გამოვაცხადოთ ტაიმერები და მთვლელები S7-1500-ის და S7-1200ისთვის (TIA Portal) ? <u>https://support.industry.siemens.com/cs/ww/en/view/67585220</u> **3.2.6. მონაცემთა გლობალური ბლოკები (DB)** (იხ. სურათი 3-11:)

Add new block			
Neme: Dete_block_1			
	type:	Giobal DS	
CB	Language:	(05	-
Organita Son	fismber		- [E]
-		() manual	
-		 automatic 	
Function block	Dest Splon.		
Function	Dere blocks (D Select one off - A global data - An instance d	85) we done areas in th to following types block ata block	er som program virkelt enträlle som dirte.
Data bizzà	-		
Additional infor	mation		
Add ones and gen			Dec. Decel

სურათი 3-11: "Add new block" ფანჯარა (ახალი ბლოკის დამატება) (DB)

მონაცემთა ბლოკებში შეიძლება მოთავსებულ იყოს სხვადასხვა სამომხმარებლო, ინფორმაცია, რომელიც შეიძლება გამოყენებულ იქნას მთელს პროგრამაში (იხ. სურათი 3-12:).



სურათი 3-12: გლობალური DB, როგორც მონაცემთა შენახვის ძირითადი ადგილი

უპირატესობანი

- მეხსიერების სტრუქტურირებული არე;
- წვდომის მაღალი სიჩქარე.

თვისებები

- სამომხმარებლო პროგრამის ყველა ბლოკს შეუძლია მიწვდეს გლობალურ DB-ს;
- გლობალური DB-ს სტრუქტურა შეიძლება შედგებოდეს მონაცემთა ნეისმიერი ტიპებისაგან;
- გლობალური DB-ების შექმნა შეიძლება პროგრამულ რედაქტორში ანდა ადრე შექმნილი "user-defind PLC data type"-ის შესაბამისად (მომხმარებლის მიერ განსაზღვრული მონაცემთა ტიპი) (იხ. თავი - მონაცემთა ტიპი STRUCT და PLC data types).

რეკომენდაციები

 გამოიყენეთ გლობალური DB, მაშინ როდესაც მონაცემების დამუშავება უნდა მოხდეს პროგრამის სხვა ნაწილში.

შენიშვნა

თქვენ შეგიძლიათ მონახოთ დამატებითი ინფორმაცია შემდეგი საკითხების შესახებ: რა ტიპის დაშვებები, სვეტების მნიშვნელობები და მუშაობის რეჟიმებია ხელმისაწვდომი გლობალური მონაცემთა ბლოკებისათვის STEP 7-ში. https://support.industry.siemens.com/cs/ww/en/view/68015630

3.2.7 ჩატვირთვა განმეორებითი ინციალიზაციის გარეშე

სამომხმარებლო პროგრამა იმისათვის, რომ შევცვალოთ მომუშავე **M330** კონტროლერზე, S7- 1200 კონტროლერებს (დაწყებული V4.0 ვერსიიდან) და S7-1500 კონტროლერებს აქვთ ოპტიმიზირებული ფუნქციების და/ან მონაცემთა ბლოკების გაფართოვების შესაძლებლობა მუშობის პროცესში. ინტერფეისეზის တ႑ဒ႑ာ შეგეძლებათ ჩატვირთოთ შეცვლილი ბლოკები CPU-ს STOP რეჯიმში გადასვლის გარეშე, ამასთან ადრე ჩატვირთული ცვლადების ფაქტიური მნიშვნელობები არ იქნება შეცვლილი (იხ. სურათი 3-13:).



სურათი 3-13: ჩატვირთვა განმეორებითი ინციალიზაციის გარეშე

თუ კონტროლერი იმყოფება RUN რეჯიმში, მაშინ უნდა შეასრულოთ შემდეგი პროცედურები:

- 1. გაააქტიურეთ "Downloading without reinitialization" (ჩათვირთვა განმეორებითი ინციალიზაციის გარეშე);
- 2. დაამატეთ ახალი ცვლადები მონაცემთა ბლოკში;
- 3. ჩატვირთვეთ ბლოკი კონტროლერში.

უპირატესობანი

ახალი ცვლადების ჩატვირთვა პროცესორის მუშაობის შეწყვეტის გარეშე.
 კონტროლერი რჩება "RUN" რეჯიმში.

თვისებები

- ჩატვირთვა განმეორებითი ინციალიზაციის გარეშე შესაძლებელია მხოლოდ ოპტომიზირებულ ბლოკებში;
- ახალი ცვლადები იქნება ინციალიზირებულები. დარჩენილი ცვლადების მნიშვნელობდები დარჩება უცვლელები;
- ბლოკს სარეზერვო მეხსიერებით ესაჭიროება მეტი მეხსიერება კონტროლერში;
- სარეზერვო მეხსიერება დამოკიდებულია კონტროლერის მუშა მეხსიერებისაგან, მიუხედავად ამისა ის შეიძლება იყოს არაუმეტეს 25 მგბ-ის.
- იგულისხმება, რომ ბლოკისათვის განსაზღვრულია მეხსიერების რეზერვი;
- სიჩუმით, მეხსიერების რეზერვი დაყენებულია 100 ბაიტის სიდიდის;
- მეხსიერების რეზერვი მიეცემა ინდივიდუალურად თვითოეული ბლოკისათვის ცალცალკე;
- ბლოკები შეიძლება იქნას გაფართოვებულნი.

რეკომენდაციები

 განსაზღვრეთ მეხსიერებს რეზერვი ბლოკებისათვის, რომლებიც შემდეგში იქნება გაფართოვებულნი ექსპლოატაციაში შეყვანის პროცესში (მაგ. ტესტური ბლოკები). ექსპლუატაციაში შეყვანის პროცესი არ იქნება შეწყვეტილი ახალი ცვლადების ჩატვირთვისას. უკვე შექმნილი ცვლადების მიმდინარე მნიშვნელობები არ იქნება შეცვლილნი.

მაგალითი: ბლოკისთვის მეხსიერების ზომის დაყენება

შემდეგ ცხრილში ნაჩვენებია მეხსიერების რეზერვის დანიშვნა მისი შემდგომი ჩატვირთვისათვის განმეორებადი ინციალიზაციის გარეშე (იხ. ცხრილი 3-3:).



ცხრილი 3-3: მეხსიერების რეზერვის დანიშვნა

შენიშვნა

თქვენ შეგიძლიათ დააყენოთ მეხსიერების რეზერვის სიდიდის მნიშვნელობა სიჩუმით ახალი ბლოკებისათვის TIA Portal-ში.

მენიუში, გადადით "Options – Settings"-ში, შემდეგ "PLC programming – General-Dounload without reinitialization"-ში.

მაგალითი: ჩატვირთვა მეორადი ინციალიზაციის გარეშე

შემდეგ მაგალითში ნაჩვენებია თუ როგორ უნდა შესრულდეს ჩატვირთვა განმეორებითი ინციალიზაციის გარეშე (იხ. ცხრილი 3-4:)

ცხრილი 3-4: ჩატვირთვა განმეორებითი ინციალიზაციის გარეშე

საფეხურ	ინსტრუქცია					
1.	მოთხოვნა: დაყენებულ უნდა იქნას მეხსიერების რეზერვი (იხ. ზევით)					
2.	გახსენით მონაცემთა ოპტიმიზირებული ბლოკი DB					
3.	დააჭირეთ ღილაკზე "Download without reintialization" (ჩატვირთვა განმეორებითი ინციალიზაციის გარეშე) და დააჭირეთ "OK" Global_D8_Re-init Hare Start value Recsin Start v					
4.	QSSGSSGSGGG SbSccn (GgndcgbS GgbSbgnon)					
5.	ჩატვირთეთ ბლოკი კონტროლერში					
6.	რეზულტატი: - ბლოკის ცვლადების ფაქტიური მნიშვნელობები დარჩნენ იგივენი					

შენიშვნა

უფრო დაწვრილებითი ინფორმაცია თქვენ შეგიძლიათ მოძებნოთ TIA Portal - ის ონლაინ დახმარებაში "Loading block extensions without reinitialization" (ბლოკის გაფართოვების ჩატვირთვა განმეორებითი ინციალიზაციის გარეშე).

თქვენ შეგიძლიათ მოძებნოთ დამატებითი ინფორმაცია შემდეგი საკითხების ირგვლივ: როგორია ჩატვირთვის ხერხები S7-1500-ში RUN რეჟიმში? https://support.industry.siemens.com/cs/ww/en/view/68015630

3.2.8 ბლოკების განმეორებითი გამოყენების შესაძლებლობები

ბლოკების შექნმის კონცეფციიდან გამომდინარე, თქვენ გეძლევათ შესაძლებლობა შექმნათ სტრუქტურირებული და ეფექტიური პროგრამები.

უპირატესობანი

- ბლოკები შეიძლება გამოყენებულ იქნას პროგრამის ნებისმიერ ადგილას;
- ბლოკები შეიძლება უნივერსალურად გამოყენებულ იქნას სხვა პროექტში;
- იმ შემთხვევაში თუ თვითოეული ბლოკი წყვიტავს თავის დამოუკიდებელ ამოცანას, მაშინ ასეთი მიდგომა წარმოადგენს სტრუქტურული დაპროგრამების საფუძველს;
- მცირდება შეცდომათა რაოდენობა;
- შესაძლებელია შეცდომათა მარტივი დიაგნოსტირება.

რეკომენდაციები

თუ თქვენ გინდათ გამოიყენოთ ბლოკი განმეორებით, მაშინ იმოქმედეთ შემდეგი რეკომენდაციების მიხედვით:

- აღიქვით ბლოკი როგორც ინკაფსულირებული ფუნქცია. ანუ, თვითოეული ბლოკი წარმოადგენს თავის თავად სამომხმარებლო პროგრამის ამა თუ იმ ამოცანის ამოხსნისათვის;
- გამოიყენეთ რამდენიმე მთავარი OB (Main), სისტემის წაწილების დაჯგუფებისათვის;
- ყოველთვის გამოიყენეთ მონაცემთა გაცვლა ბლოკებს შორის საკუთარი ინტერფეისების მეშვეობით და არა ეკზემპლიარების გავლით (იხ. თავი გავლით (იხ. თავი 3.4.1 ბლოკის ინტერფეისი მონაცემთა გაცვლისათვის);
- ბლოკთან მუშაობის დროს არ გამოიყენოთ საპროექტო სპეციფიკური მონაცემები და ასევე შემდეგი კომპონენტები:
 - ხელწვდომა გლობალურ DB-ებთან და კონკრეტული DB ეკზემპლიარების გამოყენება;
 - ხელწვდომა ტეგებთან;
 - ხელწვდომა გლობალურ კონსტანტებთან.
- იმ ბლოკებისათვის რომლებიც შესაძლებელია გამოყენებულ იქნას ხელმეორედ არის მოთხოვნათა ჩამონათვალი, ისევე როგორც დაცულ (know-hou-protectid) ბლოკებთან ბიბლიოთეკებში. ამიტომ, თქვენ აუცილებლად უნდა შეამოწმოთ ბლოკების იმ თვისებების არსებობა, რომლებიც იძლევა განმეორებითი გამომახების საშუალებას "Block can be used as know-how protected library element" (ბლოკი შეიძლება გამოყენებულ იქნას როგორც ბიბლიოთეკის ელემენტი knowhou-protectid) დაცვით. შემოწმებამდე გადაიღეთ ბლოკის ასლი (იხ. სურათი 3-14:).

Attributes		
EC chec	¢	
🗌 Handle e	rrors within block	
Block car	be used as know-how prot	ected library element

სურათი 3-14: ბლოკის ატრიბუტები

3.2.9. ბლოკებზე ნომრების ავტომატური მიკუთვნება

შიგა დამუშავებისათვის, ბლოკების ნომრების აუცილებელი მიკუთვნება ხდება ავტომატურად სისტემის მიერ (აწყობა ბლიკის თვისებებში) (იხ. სურათი 3-15:).



სურათი 3-15: ბლოკებზე ნომრების ავტომატური მიკუთვნება

უპირატესობანი

 ნომრებიანი ბლოკების კონფლიქტის თავის აიცილება ხდება ავტომატურად, მაგალითად კოპირების შემდეგ, TIA Portal-ში კომპილიაციის პროცესში.

რეკომენდაციები

• არ შეცვალოთ მიმდინარე აწყობა "automatic" (ავტომატიური) (სურათი 3-16:).

fb_KiteBoard [FB1]		×
General		
General Information Time stamps Compilation Protection Attributes Download without reinitia lization	General Nome: Type: Language: Number:	te_Kiteboard FB SCL ★ I ↓ Manual
	(¢] 11	3
		OK Cancel

სურათი 3-16: აწყობა ბლოკის თვისებებში

3.3. ბლოკის ინტერფეისის ტიპი

FB და FC ბლოკებს აქვთ სამი სხვადასხვა ტიპი ინტერფეისში: In (შესასვლელი), InOut (გამავალი) და Out (გამოსასვლელი). ინტერფეისის ასეთი ტიპების დახმარებით ბლოკები იძენენ პარამეტრებს. პარამეტრები დამუშავდება და რეზულტატები გადაეცემა გამომძახებელ ბლოკს. InOut პარამეტრები გამოიყენება როგორც მონაცემთა გადაცემისათვის გამომძახებელ ბლოკში, ასევე უკან რეზულტატის დაბრუნებისათვის. ასეთი პარამეტრებისათვის არსებობს გადაცემის ორი ხერხი.

3.3.1. შესასვლელ პარამეტრზე ფაქტიური მნიშვნელობების მიკუთვნება

ბლოკის გამოძახების დროს, ფაქტიური პარამეტრის მნიშვნელობა კოპირდება In ტიპის ბლოკის შესასვლელ პარამეტრში. ამისათვის ბლოკში გამოიყოფა მეხსიერების დამატებითი არე (იხ. სურათი 3-17:).



სურათი 3-17: შესასვლელ პარამეტრში მნიშვნელობების კოპირება

თვისებები

- თვითოეული ბლოკი ახდენს მოცემული პარამეტრების ერთნაირ დამუშავებას;
- მწიშვნელობების კოპირება ხდება ბლოკის გამოძახების დროს.

3.3.2 ფაქტიური მნიშვნელობის მიკუთვნება გამავალ პარამეტრზე

ბლოკის გამოძახების დროს, ფაქტიური პარამეტრის მისამართი მიეკუთვნება გარმავალ პარამეტრს. ამასთან, მეხსიერების დამატებითი გამოყოფა არ არის საჭირო (იხ. სურათი 3-18:).



სურათი 3-18: მნიშვნელობებზე გადაცემა ბმულით (მაჩვენებელი პარამეტრის შენახვის ადგილზეა)

თვისებები

- თვითოეული ბლოკი ახდენს მოცემული პარამეტრის ერთნაირ დამუშავებას;
- ფაქტიური პარამეტრები გადაეცემა ბმულით გამომმახებელ ბლოკს.

რეკომენდაციები

 მეხსიერების რაციონალური გამოყენებისათვის სტრუქტურირებელი ცვლადებისათვის (მაგ. ARRAY, STRUC, STRING, ...), გამოიყენეთ გამავალი პარამეტრები.

3.3.3 პარამეტრის გადაცემის ვარიანტები

შემდეგ ცხრილში ნაჩვენებია, თუ როგორ შეიძლება S7-1200/S7-1500-ში ბლოკის პარამეტრები გამოყენებულ იქნას ელემენტარული და რთული ტიპის ცვლადების გადაცემისათვის (იხ. ცხრილი 3-5:).

ბლოკის ტიპი/ფორმ. პარამეტრი		მონაცემთა ელემენტ. ტიპი	მონაცემთა რთული ტიპი	
FC	შესასვლელი	ასლი	ბმულით	
	გამოსასვლელი	ასლი	ზმულით	
	გადასასვლელი	ასლი	ბმულით	
FB	შესასვლელი	ასლი	ასლი	
	გამოსასვლელი	ასლი	ასლი	
	გადასასვლელი	ასლი	ბმულით	

ცხრილი 3-5: პარამეტრის გადაცემის ვარიანტები

შენიშვნა

თუ ბლოკის გამძახების დროს, ხდება ოპტიმიზირებული მონაცემების გადაცემა "standart access" (სტანდარტული ხელწვდომა) თვისებით, მაშინ ასეთი მონაცემები გადაცემა ასლების სახით. თუ ბლოკი შეიცავს რთული ტიპის ბევრ პარამეტრს, მაშინ შეიძლება წარმოიშვას დროებითი ცვლადების არეს გადავსება (ლოკალური სტეკი). ამის თავიდან აცილება შეიძლება, თუ შევქმნით ხელწვდომის ერთიდაიგივე ტიპს ორთავე ბლოკისათვის. (იხ. თავი 2.6.5 პარამეტრების გადაცემა ბლოკებს შორის ოპტიმიზირებული და სტანდარტული ხელწვდომით).

3.4 შენახვის პრინციპი

Step 7-ში არის განსხვავება გლობალურ და ლოკალურ მეხსიერებათა არეებს შორის. სამომხმარებლო პროგრამაში მეხსიერების გლობალური არე ხელმისაწვდომია ნებისმიერი ბლოკისათვის, ხოლო ლოკალური მეხსიერების არე ხელმისაწვდომია მხოლოდ შესაბამისი ბლოკისათვის.

3.4.1 ბლოკების ინტერფეისები მანაცემთა გაცვლისათვის

თუ თქვენ ახდენთ პროგრამისა და ფუნქციების "ინკაპსულირებას", მაშინ ბლოკებს შორის მონაცემთა გაცვლა შესაძლებელია შესრულებულ იქნას მხოლოდ მათი ინტერფეისების საშუალებით, რაც იძლევა უპირატესობას მათი გამოყენების დროს.

უპირატესობანი

- პროგრამა შეიძლება შედგენილ იყოს ცალკეული ბლოკებისაგან, რომელთაგან თვითოეული წყვეტს მის წინაშე დასმულ თავის ამოცანას;
- პროგრამა შეიძლება ადვილად იქნას გაფართოვებული და შეყვანილი მუშაობაში;
- პროგრამული კოდი ადვილად წაიკითხება, რადგანაც არსებობს დამალული ურთიერთგადამკვეთი ხელწვდომები.

რეკომენდაციები

- შემლებისდაგვარად გამოიყენეთ მხოლოდ ლოკალური ცვლადები. ასეთ შემთხვევაში ბლოკი შეიძლება იყოს უნივერსალური და გამოყენებული მრავალჯერ;
- მონაცემთა გაცვლები შეასრულეთ ინტერფეისული ბლოკების (In, Out, InOut) გამოყენებით, რათა შესაძლებელი იყოს მათი მრავალჯერადი გამოყენება;
- ლოკალური მეხსიერებისათვისათვის გამოიყენეთ მხოლოდ ეკზემპლიარული ბლოკები შესაბამისი ფუნქციონალური ბლოკებისათვის. დანარჩენმა ბლოკებმა არ უნდა ჩაწერონ რაიმე მონაცემები ეკზემპლიარულ ბლოკებში (იხ. სურათი 3-19:).



სურათი 3-19: ეკზემპლიარულ ბლოკებთან დაშვების თავიდან აცილება

თუ კი ინფორმაციის გაცვლისათვის გამოიყენება მხოლოდ ბლოკის ინტერფეისი, მაშინ ეს იძლევა გარანტიას, რომ ყველა ბლოკს შეეძლება ფუნქციონირება ერთმანეთისაგან დამოუკიდებლად (იხ. სურათი 3-20:).



სურათი 3-20: ბლოკის ინტერფეისი მონაცემთა გაცვლისათვის

3.4.2. მეხსიერების გლობალური არე

მეხსიერების არეს ეწოდება გლობალური თუ კი სამომხმარებლო პროგრამის ნებისმიერ ნაწილს შეემლება მიიღოს მასზედ ხელწვდომა. არსებობს აპარატულად დამოკიდებული მეხსიერების არეები (მაგალითად მერკერული მეხსიერება, ტაიმერები, მთვლელები და სხვ). და გლობალური DB. აპარატულად დამოკიდებულ მეხსიერებათა არეებთან მუშაობის დროს უნდა გვახსოვდეს, რომ პროგრამა შეიმლება არ აღმოჩნდეს შრომისუნარიანი სხვა კონტროლერებთან შედარებით, რადგან ზოგიერთი მეხსიერების არე შეიძლება უკვე დაკავებული იყოს. ამიტომ თქვენ უნდა გამოიყენოთ გლობალური DB აპარატულად დამოკიდებული მეხსიერების არეების ნაცვლად.

უპირატესობანი

- სამომხმარებლო პროგრამები შეიძლება გამოყენებულ იქნას უნივერსალურად და დამოუკიდებლად აპარატულურული კონფიგურაციისაგან;
- სამომხმარებლო პროგრამა შეიძლება იყოს სტრეუქტურიზებული მერკერული
 მეხსიერების დაყოფის გარეშე სხვადასხვა მომხმარებლებისათვის;
- ოპტიმიზირებული გლობალური DB გაცილებით უფრო ეფექტურია ვიდრე მერკერული მეხსიერების არე, რომელიც არ არის ოპტიმიზირებული მისი შეთავსებადობის აუცილებლობიდან გამომდინარე.

რეკომენდაციები

- არ გამოიყენოთ მერკერული მეხსიერება, იმუშავეთ გლობალურ DB-ებთან.
- არ იმუშაოთ აპარატულად დამოკიდებულ მეხსიერებასთან, მაგ.
 სინქრობაიტთან ანდა მთვლელთან. გამოიყენეთ IEC მთვლელები და ტაიმერები მულტიეკზემპლიარებით (იხ. თავი 3.2.5 მულტიეკზემპლიარები).
 IEC ტაიმერები შეგიძლიათ იპოვოთ "Instructions – Basic Instructions – Timer operations"-ში (ინსტრუქციები-ძირითადი ინსტრუქციები-ტაიმერები) (იხ. სურათი 3-21:).

 Timer operations 	
IEC Timers	
s TP	Generate pulse
= TON	Generate on-delay
TOF	Generate off-delay
TONR	Time accumulator
-(TP)-	Start pulse timer
-(TON)-	Start on-delay timer
() -(TOF)	Start off-delay timer
() -(TONR)-	Time accumulator
() -(RT)-	Resettimer
() -(PT)-	Load time duration

სურათი 3-21: IEC ტაიმერები

3.4.3. მეხსიერების ლოკალური არე

- სტატიკური ცვლადები;
- დროითი ცვლადები.

რეკომენდაციები

- გამოიყენეთ სტატიკური ცვლადები იმ მწიშვნელობებისათვის, რომლებიც დაგჭირდებათ მომდევნო ციკლში;
- გამოიყენეთ დროითი ცვლადები როგორც კეშ მეხსიერება მიმდინარე ციკლისათვის. ხელწვდომის დრო დროით ცვლადებთან უფო ნაკლებია, ვიდრე სტატიკურ ცვლადებთან.

შენიშვნა

ოპტიმიზირებული ბლოკები: დროითი ცვლადების ინციალიზაცია ხდება ბლოკის "მნიშვნელობა სიჩუმით" ყოველი გამოძახების შემთხვევებში (S7-1500 და S7-1200 დაწყებული v4-დან).

არაოპტიმიზირებული ბლოკები: დროით ცვლადებს აქვთ გაურკვეველი მნიშვნელობები ბლოკის ყოველი გამოძახებისას.

3.4.4. წვდომის სიჩქარე მეხსიერების არეებთან

STEP 7-ში მეხსიერებასთან ხელწვდომის ორი ვარიანტი არსებობს. სისტემური მიზეზების გამო ხელწვდომის ერთი რომელიმე ტიპი შეიძლება იყოს უფრო სწრაფი ვიდრე მეორე (იხ. სურათი 3-22:).



სურათი 3-22: მეხსიერებასთან წვდომის სხვადასხვა ხერხი

ყველაზე უფრო სწრაფი ხელწვდომის ტიპი S7-1200/1500-ში ხელწვდომის დროის ზრდის მიხედვით:

- ოპტიმიზირებული ბლოკები: დროითი ცვლადები, FC და FB პარამეტრები, არაშენახვითი სტატიკური ცვლადები;
- 2. ოპტიმიზირებული ბლოკები ხელწვდომით:
- FB-ს შენახვით ცვლადით;
- ოპტიმიზირებულ გლობალურ DB-ებით;
- 3. ხელწვდომა არაოპტიმიზირებულ ბლოკებთან;
- 4. ინდექსირებული ხელწვდომა, ინდექსით, რომელიც გაიანგარიშება შესრულების რეჯიმის დროს (მაგალითად, Motor [i]);
- 5. ხელწვდომა შემოწმებით შესრულების რეჯიმში:
- ხელწვდომა DB-ებთან, რომლებიც იქმნება შესრულების რეჟიმის მუშაობის
 პროცესში ანდა იხსნება ირიბი დამისამართებით (მაგ. OPN DB[I];
- ხელწვდომა რეგისტრებთან, ანდა ირიბი ხელწვდომა მეხსიერებასთან;
- სტრუქტურების კოპირება ოტომიზირებულ და არაოპტიმიზირებულ ბლოკებს შორის (გარდა ბაიტური მასივისა).

3.5. შენახვითობა

კვების წყაროს შეშფოთების შემთხვევაში, კონტროლერი მისი სარეზერვო ენერგიის დახმარებით ახდენს შესანახი მონაცემების კოპირებას მუშა მეხსიერებიდან ენერგოდამოუკიდებელ მეხსიერებაში. კონტროლერის რესტარტის შემდეგ, პროგრამის მუშაობა გრმელდება შენახული მონაცემებისაგან. კონტროლერის მოდელისაგან დამოკიდებულებით ხელმისაწვდომია შენახვისათვის გათვალისწინებული მეხსიერების სხვადასხვა მოცულობები.

კონტროლერი	გამოყენებადი შენახვითი მეხსიერება მერკერების, ტაიმერების, მთვლელების DB და ტექნოლოგიური ობიექტებისათვის
CPU 1211C,1212C, 1214C, 1215C, 1217C	10კბაიტი
CPU 1511-1 PN	88 კბაიტი
CPU 1513-1 PN	88კბაიტი
CPU 1515-2 PN, 1516-3 PN/DP	472კბაიტი
CPU 1518-4 PN/DP	768კბაიტი

ცხრილი 3-6: შენახვისათვის განკუთვნილი მეხსიერება

ცხრილი 3-7: განსხვავება S7-1200 და S7-1500 შორის

\$7-1200	S7-1500
შენახვითობა შეიძლება დადგენილ	შენახვითობა შეიძლება დადგენილ
იქნას მხოლოდ მერკერებისათვის	იქნას მერკერების მთვლელების_და
ტაიმერებისათვის	ტაიმერებისათვის

უპირატესობანი

 შესანახი მონაცემები იმახსოვრებს თავიანთ მნიშვნელობებს კონტროლერის STOP რეჟიმში გადასვლის შემთვევაში და უკან RUN-ში ანდა კვების წყაროს შეშფოთებისა და კონტროლერის რესტარტის დროს.

თვისეზეზი

ოპტიმიზირებული DB-ს მონაცემების ელემენტარული ტიპების ცვლადებისათვის შენახვითობა შეიძლება აწყობილ იქნას ინდივიდუალურად თვითოეული ცვლადისათვის ცალცალკე. არაოპტიმიზირებული მონაცემთა ბლოკები შეიძლება იყოს ან მთლიანათ შენახვითები ანდა მთლანად არაშენახვითები.

შენახვითი მონაცემები შეიძლება მოშორებულ იქნას "memory reset"-ის დახმარებით (მეხსიერების ჩამოყრა) ანდა "Reset to factory settings" (ჩამოყრა ქარხნულ ანაწყობებზე):

- რეჟიმის გადართვა კონტროლერზე (MRES);
- კონტროლერის დისპლეი;
- Online STEP 7 (TIA Portal)-ის დახმარებით.

რეკომენდაცია

 არ გამოიყენოთ თვისება "Set in IDB" (დაინიშნება მონაცემთა ეკზემპლიარულ ბლოკში). ყოველთვის დანიშნეთ შენახვითობა ფუნქციონალურ ბლოკში და არა მონაცემთა ეკზემპლიარულ ბლოკში.

ø	2		🖻 👟 🕾 🔝 e	° €o "⊞ :	🦻 🖛 🗉 E	≣ '= '= ∦≎ ∮
	Int	erf	ace			
		Ne	me	Data type	Default value	Retain A
1	-	¥	Input			
2		٠	<trina.ri0gen></trina.ri0gen>			
3	-0	-	Output			
4		•	-dHinzufügen>-			
5	-	•	InOut			1
6		٠	<hinaufügen></hinaufügen>			
7	-	¥	Static			
8	-	٠	area_1	LReal 📃	0.0	Non-retain 💌
9	-0	٠	area_2	LReal	0.0	Non-retain
10	-	٠	height	LReal	0.0	Set in IDB
11	-0	•	radius	LReal	0.0	Jet million
12	-	•	return	LReal	0.0	Non-retain

სურათი 3-23: პროგრამული რედაქტორი (ფუნქციონალური ბლოკის ინტერფეისი)

	DB	Col	ntrolPart				
		Ne	me	Data type	Start value	Retain	Accessible from
1		÷	Static				
2	-	•	WriteDBL	Bool	ta ise		
5	-0	٠	WriteDBL_F	Bool	false		
ŧ.	-0		WRIT_RET_VAL	Int	0		
5	-0		WRIT_Busy	Bool	false		
5	-0	•	ReadDBL	Bool	false		. 🗹 .
7	•	٠	ReadDBL_P	Bool	fabe		
3	-	٠	READ_RET_VAL	Int	0		Image: 1
	-		READ_Busy	Bool	ta ise		I = 1

სურათი 3-24: პროგრამული რედაქტორი (მონაცემთა ბლოკი)

 თვისება "Set in IDB" (დაინიშნება მონაცემთა ეკზემპლიარულ ბლოკში) ახანგრძლივებს პროგრამის დამუშავების დროს. FB-ს ინტერფეისისათვის ყოველთვის აირჩიეთ "Non-retain" (არა შენახვითობა) ანდა "Retain" (შენახვითობა) (იხ. სურათი 3-23:, სურათი 3-24:).

მაგალითი: PLC-ის ტეგების შენახვითობა

მონაცემთა შენახვითობის აწყობა სრულდება PLC-ის ტეგების ცხრილში, ფუნქციონალურ ბლოკებში ანდა მონაცემთა ბლოკებში (იხ. სურათი 3-25:).

Prøject1 → PLC_2	(CPU 1214C DC/DC/D	C] → PLC_Z → PL	.C tags → I	Default tag				
			📹 Tags	🔳 User	B			
🥩 🗗 📆 🗰				-				
Default tag ta	ble			í l	k i i			
Name	Data type	Address Re	tain Visibl.	Canb Ç				
1 🚾 Start_1_97	-1200 Bool	%i0.0			E			
2 💶 Stop_1_57	-1200 Bool	%0.1		Image: A start of the start				
B 🚳 clock_bit_f	Bool	%M10.0	×		Ē.			
4 🚾 clock_bit_3	2 Bool	🗄 %M10.1 💌 🛛	/	Image: A start of the start	E			
5 😡 clock_bit_3 6 🔕 clock_bit_4	Retain memory	BHIA3		>	£			
7 🔤 clock_bit_s ای سندر ای میراند.	5 Number of	memory bytes starti	ng at MBO:					
	Number	of SIMATIC timers star	rting at TO:		· · · · · ·			
	Number of S	IMATIC counters star	ting at CO:	•	შენ ,და	ნახვით ყენებუ	ობა შეიძღ ლ იქნას	' 0
	Aveil	sble retentive memo	ory (Bytes):	484000	ნუ მაე	ლოვან გ., MBO	ი მისამარი , TO და CO	הת)–ו
			ок	Cencel	. —			

სურათი 3-25: ცვლადების შენახვითობის აწყობა PLC-ის ტეგების ცხრილში

	Inte	erfi	ace	1				
		Name				Data type	Default value	Retain
1	-0	٠	Sta	tic				
8	-0	•	Ŧ	IEC_Counter	Instance	IEC_COUNTER]	Retain 🔫
9	-01		•	CU		Bool	false	Non-retain
10	-0		•	œ		Bool	false	Rata in Set in IDS
11	-0		•	R		Bool	false	Ketam
12	-0		•	LD		Bool	false	Retain
13	-0		•	QU		Bool	false	Retain
14	-01		•	QD		Bool	false	Retain
15	-0		•	PV		Int	0	Retain
16	-0		•	CV		Int	0	Retain
	<							
	Blo	ck	title	e:				
·								
•		Net	two	VNK 1:				
	C	om	mer	nt				
					#IEC_Counter_ Instance			
					сти			
					Int			
		4	Col	untSignal —	CU			
					-	ALC: NOT THE REPORT OF A		

სურათი 3-26: შენახვითი მთვლელი როგორც მულტიეკზემპლიარი

მაგალითი: შენახვითი მთვლელი

თქვენ შეგიძლიათ ასევე გამოაცხადოთ ეკზემპლიარები ანდა ფუნქციები (ტაიმერი, მთვლელი და სხვ.), როგორც შენახვითები. როგორც ეს იყო ადრე აღწერილი 3.2.5-ში მულტიეკზემპლიარები (იხ. სურათი 26:).

შენიშვნები

თუ PLC-ის შენახვითი მეხსიერება არ არის საკმარისი, მაშინ აზრი აქვს შენახულ იქნას მონაცემები მონაცემთა ბლოკების სახით, რომლებიც იმყოფებიან მხოლოდ PLC-ის ჩატვირთვის მეხსიერებაში. აღნიშნული ხერხი აღწერილია მაგალითის სახით S7-1200-სთვის. იგი ასევე შეიძლება გამოყენებულ იქნას S7-1500-ისთვისაც. თქვენ შეგიძლიათ მონახოთ დამატებითი ინფორმაცია შემდეგი საკითხისათვის: როგორ მოვახდინოთ მონაცემთა ბლოკის კონფიგურირება ატრიბუტით "Only store in load memory" (შეინახება მხოლოდ ჩატვირთვის მეხსიერებაში) STEP 7 (TIA Portal) S7-1200-ისთვის.

https://support.industry.siemens.com/cs/ww/en/view/53034113

3.6. სიმბოლური დამისამართება

3.6.1 სიმბოლური დამისამართება აბსოლუტური დამისამართების ნაცვლად

TIA პორტალი ოპტიმიზირებულია სიმბოლური დამისამართების გამოყენებისთვის. ამასთან თქვენ იღებთ მრავალ უპირატესობას. სიმბოლური დამისამართებით მუშაობის დროს თქვენ შეგიძლიათ შექმნათ პროგრამები, ისე რომ არ მიაქცოთ ყურადღება ბლოკების შინაგან სტრუქტურებს. კონტროლერი თვითონ განსაზღვრავს მონაცემთა შენახვის ოპტიმალურ ვარიანტს. ამით თქვენ შეგეძლებად მთლიანად იფიქროთ დასმული ამოცანის გადაწყვეტაზე.

უპირატესობანი

- პროგრამა ხდება ადვილად წასაკითხი, ცვლადების სიმბოლური სახელწოდებების გამოყენებით;
- ცვლადების სახელების ავტომატური განახლება პროგრამის ყველა ადგილას;
- შესანახი მონაცემების მართვა პროგრამაში არ ითხოვს ხელით მართვას (აბსოლუტური დამისამართება);
- მონაცემებთან ხელწვდომის მძლავრი მექანიზმი;
- სწრაფქმედების გაზრდისთვის ანდა პროგრამის ზომის შემცირებისთვის არ მოითხოვება ხელით ოპტიმიზაცია:
- IntelliSense გვეხმარება შევასრულოთ ცვლადების სწრაფი შეყვანა;
- პროგრამული შეცდომების ნაკლები რაოდენობა, რაც მიიღწევა ტიპის შემოწმების გამოყენებით. (მონაცემთა ტიპების შემოწმება სრულდება ყველა ტიპის ხელწვდომისათვის).

რეკომენდაციები

• "არ მოითხოვება მონაცემთა შენახვის სტრუქტურის ცოდნა"

მაგალითი



ცხრილი 3-8: სიმბოლური ცვლადების შექმნის მაგალითი

 "იფიქრეთ სიმბოლურად. განსაზღვრეთ "შესაბამისი" სახელი თვითოეული ფუნქციისათვის, ცვლადებისათვის, მაგალითად, Pump_boiler_1, heater_room_4 და სხვ. ამგვარად პროგრამა იქნება წაკითხვადი კომენტარების გარეშეც კი; დაუნიშნეთ ყველა გამოყენებულ ცვლადს სიმბოლური სახელი თაგვის მარჯვენა ღილაკის დახმარებით.არსებობს ძალზედ ეფექტური ხერხი, რომელიც დაგიზოგავთ დროს, თუკი თქვენ გინდათ განსაზღვროთ რამდენიმე ცვლადი სეგმენტში. პირველ რიგში დაუნიშნეთ ყველა ცვლადს სახელები. ამის შემდეგ, განსაზღვრეთ ყველა ცვლადი ერთდროულად, მე-4 ბიჯის დახმარებით.

შენიშვნა

თქვენ შეგიძლიათ ნახოთ დამატებითი ინფორმაცია შემდეგი საკითხების ირგვლივ: რატომაა სიმბოლური სახელების მიკუთვნება და გამოყენება STEP-7-ში (TIA Portal) აუცილებელი S7-1500-ისთვის? https://support.industry.siemens.com/cs/ww/en/view/67598995

3.6.2 ARRAY-ს მონაცემთა ტიპი და ირიბი ხელწვდომა ელემენტებთან

ARRAY (მასივი) მონაცემთა ტიპი წარმოადგენს მონაცემთა სტრუქტურას, რომელიც შედგება ერთიდაიგივე ტიპის ელემენტებისაგან. ARRAY-ს მონაცემთა ტიპი მაგალითად რეცეპტების შენახვისათვის, მასალების გადაადგილების თვალყურის დევნებისათვის, დამუშავების ციკლური თანმიმდევრობისათვის, პროტოკოლებისათვის და სხვ. (იხ. სურათი 3-27:)

f V	Vrite → PLC_1 →	Program	blocks	: •	Field	<u> </u>								
*	👼 🛼 📫 📐													
Fie	d_Data													
	Nome		Dote ty	pe										
1	👻 Static													
2^{-}	Counter		Dint											
3	👻 Numbers		array (C	19]	ofint	7								
4	Numbers[0]		Int											
5	Numbers[1]		Int											
6	Numbers[2]		Int											
7	Numbers[3]		Int											
8	Numbers[4]		Int											
9	Numbers[5]		Int											
10	Numbers[6]		Int											
11	Numbers[7]		Int											
12	Numbers[8]		Int											
13	Numbers[9]		Int											
	4 III													

სურათი 3-27: ციკლური (INT) ტიპის მონაცემებს მასივი 10 ელემენტისთვის

თქვენ შეგიძლიათ მიიღოთ ირიბი ხელწვდომა მასივის ცალკეულ ელემენტებთან (array ["index"]) ცვლადის დახმარებით.

0)	PL	C_1 [CPU 1516-3 PN	/DP] 🕨 Pro	ogram blov
				en i na i e	
1	Field	LDa	ata		
	N	ame		Data type	5
1	•	St	atic		
2	-0	Ŧ	Numbers	Array [09]	1 - •
З	-0		Numbers[0]	DInt	
4	-		Numbers[1]	Dint	LAD / FBD: MOVE
5	-0		Numbers[2]	Dint	MOVE
6	-0		Numbers[3]	Dint	EN ENO
7	-		Numbers [4]	Dint	"Field_Data".Numbers[#i] — N 🔅 OUT1 — #Field_Number
8	-0		Numbers[5]	Dint	
9	-		Numbers[6]	Dint	and the second
10	-		Numbers[7]	Dint	and the state of t
11	•		Numbers[8]	Dint	<pre>SCL: 1 #Field_Number := "Field_Data".Numbers[#i];</pre>
12	-		Numbers [9]	Dint	JCL. 2
1.2			Alexandre and an and	and a strengthered	

ნახ. 3-28: ირიბი ხელწვდომა ელემენტებთან

უპირატრესობანი

- მარტივი ხელწვდომა, რადგანაც ARRAY-ის ყველა ელემენტის ტიპი ერთნაირია;
- არ არის საჭიროება რთული მაჩვენებლის შექმნისა;
- შესაძლებელია სწრაფი შექმნა და გაფართოვება;
- გამოიყენება დაპროგრამების ყველა ენაზე.

თვისებები

- მონაცემთა სტრუქტურირებული ტიპი;
- მონაცემთა სტრუქტურა შედგება ერთიდაიგივე ტიპის გარკვეული რაოდენობისგან;
- შესაძლებელია მრავალგანზომილებიანი მასივების შექმნა;
- შესაძლებელია ირიბი ხელწვდომა ცვლადის დახმარებით, რომელიც დინამიურად იცვლება პროგრამის მუშაობის დროს.

რეკომენდაციები

- გამოიყენეთ ARRAY მასივი ინდექსირებული ხელწვდომისთვის ნაცვლად მაჩვენებლისა (მაგ. მაჩვენებელი ANY). პროგრამა ხდება უფრო თვალნათელი, რადგანაც მასივის აღქმა უფრო გასაგებია სიმბოლური სახელების გამოყენებით, ნაცვლად მაჩვენებლისა;
- უფრო სრაფი ხელწვდომის მიღებისათვს, მასივის შენახვისათვის გამოიყენეთ DINT ტიპის მონაცემები დროითი ცვლადის სახით;
- გამოიყენეთ "MOVE_BLK" ინსტრუქცია იმისათვის, რომ გადააკოპიროთ ერთი მასივის ნაწილი მეორეში;
- გამოიყენეთ "GET_ERR_ID", იმისათვის, რომ მიიღოთ შეცდომების იდენტიფიკატორი მასივთან ხელწვდომის დროს.

შენიშვნა

 თქვენ შეგიძლიათ ნახოთ დამატებითი ინფორმაცია შემდეგი საკითხების ირგვლივ: როგორ შიძლება შევასრულოთ მასივთან ხელწვდომა ცვალებადი ინდექსით S7-1500-ში?; https://support.industry.siemens.com/cs/ww/en/view/67598676

- როგორ შეიძლება უსაფრთხოთ იქნას გამოყენებული ირიბი დამისამართება S7-1500-ში (TIA Portal)?; https://support.industry.siemens.com/cs/ww/en/view/97552147
- S7-1500-ში (TIA Portal) როგორ შეიძლება შევასრულოთ მონაცემთა გადაცემა ორ ცვლადს შორის "Array of Bool" და "Word" მონაცემთა ტიპებისათვის; <u>https://support.industry.siemens.com/cs/ww/en/view/108999241</u>

3.6.3. STRUCT და PLC data type მონაცემთა ტიპები

მონაცემთა ტიპი STRUST წარმოადგენს მონაცემთა სტრუქტურას, რომელიც შედგება სხვადასხვა ტიპის ელემენტებისაგან. სტრუქტურის აღწერა წარმოებს შესაბამის ბლოკში (იხ. სურათი 3-29:).

	Datablock												
		Na	me		Datentyp	Startwei							
1	-	•	Sta	atic									
2	-01	•	Ŧ	my_tst	Struct 🔳								
З	-0		•	InterfaceId	UInt	64							
4	-0		•	ID	UInt	1							
5	-		•	ConnectionType	Byte	16#0							
6	-0		•	ActiveEstablished	Bool	true							
7	-0		•	RemoteAddress	IP_V4								
8	-		•	RemotePort	UInt	2000							
9	-0		•	Loca Port	UInt	0							

სურათი 3-29: სტრუქტურა მონაცემთა სხვადასხვა ტიპის ელემენტებით

<u>1</u> 00	1	🛫 🛫 💺 🖽 💆 📾 🚍 🔢 🖤								
		Engine_Data								
 Project1 	-		No	me	Data typ	e	Default v	Visible in	Setting val.	
Add new device	1	-0	٠	Power	Struct					
📩 Devices & networks	Z	-0	٠	MaxPower	Int		1000			
PLC_1 [CPU 1516-3 PN/DP]	з	-0	•	cosfi	Real		0.89			
T Device configuration	4	-01	Ŧ	ElValues	Struct					
🐁 Online & diagnostics	5	-0	٠	U	Int		10000			
Program blocks	6	-0	•	1	Int		335			
Technology objects	7	-0		f	Int		50			
External source files	8		•	<add new=""></add>						
PLC tags	9	-21		n	Int		1480			
 PLC data types 	10			-cAdd news-						
Add new data type										
Engine_Data	J.,									

სურათი 3-30: PLC Data types

სტრუქტურებთან შედარებით შაბლონი PLC data type განისაზღვრება კონტროლერის გარეთ TIA Portal-ში სადაც შესაძლებელია მისი ცენტრალიზებული ცვლილებები. პროგრამის ყველა ადგილი, სადაც გამოიყენება ასეთი ტიპი იქნება ავტომატურად განახლებული.

PLC data type აღიწერება განყოფილებაში "PLC data types" პროექტის ნავიგატორში მის გამოყენებამდე (იხ. სურათი 3-30:).

უპირატესობანი

- PLC data type-ში ცვლილება ავტომატურად აისახება პროგრამის ყველა გამოყენებულ ადგილებში;
- ბლოკებს შორის მონაცემთა მარტივი გაცვლა მათი ინტერფეისების გავლით.

თვისებები

- PLC data type ყოველთვის მთავრდება სიტყვის საზღვარზე (იხ. სურათი ქვევით).
- მხედველობაში მიიღეთ სისტემის თვისებები, როდესაც:
 იყენებთ შეყვანა/გამოყვანის სხვადასხვა არეებს (იხ. თავი <u>3.6.4 ხელწვდომა</u> <u>შეყვანა/გამოყვანის არეებთან PLC data types-ის დახმარებით</u>);
 - PLC data types-ის კომუტაციის დროს გამოიყენეთ ფრეიმები;
 - პერიფერიებისთვის გამოიყენეთ პარამეტრების ჩაწერები PLC data types-გან;
 - არაოპტიმიზირებულ ბლოკებში გამოიყენეთ აბსოლუტური დამისამართება (იხ. სურათი 3-31:, სურათი 3-32:).





4 ბაიტი

სურათი 3-31: PLC data types ყოველთვის მთავრდება სიტყვის საზღვარზე



სურათი 3-32: PLC data type შეყვანა/გამოყვანის არესთან მუშაობის დროს

რეკომენდაციები

- გამოიყენეთ PLC data type-ის შაბლონები რამდენიმე მნიშვნელობების გაერთიანებისათვის, მაგალითად, ფრეიმები ანდა მრავის მონაცემები (სიჩქარის დაყენება, ბრუნვის მიმართულება, ტემპერატურა, და სხვ);
- ყოველთვის გამოიყენეთ PLC data type შაბლონები სტრუქტურების ნაცვლად მათი პროგრამაში მრავალჯერადი გამოყენების მიზნით;
- გამოიყენეთ PLC data type-ის შაბლონები მონაცემთა ბლოკებში სტრუქტურირებისათვის;
- გამოიყენეთ PLC data type-ის შაბლონები მონაცემთა ბლოკისთვის სტრუქტურის დანიშვნის მიზნით. PLC data type შეიძლება გამოყენებულ იქნას ნებისმიერი რაოდენობის DB-სათვის. თქვენ შეგიძლიათ მარტივად შექმნათ DB-ს ნებისმიერი რაოდენობა ერთი სტრუქტურით და შემდეგში ააწყოთ ისინი ცენტრალიზებულად PLC data type-ის ერთნაირი შაბლონით.

შენიშვნები

თქვენ შეგიძლიათ მიიღოთ დაწვრილებითი ინფორმაცია შემდეგი საკითხების ირგვლივ:

როგორ ხდება სტრუქტურების ინციალიზაცია მეხსიერების ოპტიმიზირებულ არეებში S7-1500 STEP 7 (TIA Portal)-ში?;

https://support.industry.siemens.com/cs/ww/en/view/78678760

როგორ შევქმნათ PLC data type-ის შაბლონები S7-1500 კონტროლერისათვის?; <u>https://support.industry.siemens.com/cs/ww/en/view/67599090</u>

როგორ გამოვიყენოთ სამომხმარებლო ტიპის მონაცემთა ბლოკები (UTD) S7-1500 STEP 7 (TIA Portal)-ში?;

https://support.industry.siemens.com/cs/ww/en/view/67582844

S7-1500-თვის ბლოკის გამოძახების დროს რატომ უნდა იყოს გადაცემული მთელი სტრუქტურები, თვითოეული ელემენტების ნაცვლად? https://support.industry.siemens.com/cs/ww/de/view/67585079
3.6.4 შეყვანა/გამოყვანის არეებთან ხელწვდომა PLC data types-ის დახმარებით

S7-1500 კონტროლერებში თქვენ შეგიძლიათ შექმნათ PLC data types შაბლონები და გამოიყენოთ ისინი შესასვლელებთან და გამოსასვლელებთან სტრუქტურირებული და სიმბოლური ხელწვდომისათვის (იხ. სურათი 3-33:).



სურათი 3-33: შეყვანა/გამოყვანის არეებთან ხელწვდომა PLC data types-ის დახმარებით

- 1. PLC data type-ის შექმნა მონაცემთა საჭირო სტრუქტურით;
- 2. PLC-ის ტეგის შქმნა PLC data type-ის შაბლონისა და შესასვლელ/გამოსასვლელის არეს საწყისი მისამართით (%Ix.0 ანდა %Qx.0, მაგალითად, %I0.0, %12.0, ...);
- 3. PLC-ის ტეგის გადაცემა როგორც ფაქტიური პარამტრის, ფუნქციონალურ ბლოკში;
- 4. ფუნქციონალური ბლოკის გამომავალი პარამტრი PLC data type-ის შექმნილი შაბლონის ტიპით.

უპირატესობანი

- მაღალი ეფექტურობა დაპროგრამების დროს;
- მარტივი მრავალჯერადი გამოყენება, PLC data type შაბლონის დახმარებით.

რეკომენდაციები

 გამოიყენეთ PLC data type შესასვლელ/გამოსასვლელებთან ხელწვდომის მისაღებად, მაგ. ტელეგრამების სიმბოლური მიღებისა და გაგზავნისათვის;

შენიშვნა

PLC data type-ის შაბლონის თვითოეულ ელემენტთან ხელწვდომა შეიძლება შესულებულ იქნას პირდაპირ სამომხმარებლო პროგრამაში:



3.6.5 არჩევითი ხელწვდომა

S7-1200/1500 კონტროლერებისათვის, თქვენ შეგიძლიათ მიიღოთ Byte, Word, DWord ან LWord ტიპის მონაცემთა ტეგებთან ხელწვდომა. მეხსიერების არეების დაყოფას (მაგ. ბაიტი ან სიტყვა) უფრო მცირე ზომის მეხსიერების არეებად (მაგ. Bool) ასევე ეწოდება არჩევა. ქვემოთ მოყვანილ სურათზე, ნაჩვენებია ხელწვდომა ცვლადებთან მონაცემთა ტიპებისათვის ბიტი, ბაიტი და სიტყვა (იხ. სურსათი 3-34:).



სურათი 3-34: არჩევითი ხელწვდომა

უპირატესობანი

- მაღალი ეფექტურობა დაპროგრამების დროს;
- ცვლადების გამოცხადების დროს არ არის საჭირო დამატებითი
 სტრუქტურირება;
- მარტივი ხელწვდომა (მაგალითად მართვის ბიტები).

რეკომენდაციები

 AT ასახვების ნაცვლად გამოიყენეთ არჩევითი ხელწვდომა მონაცემთა კონკრეტულ არეებთან ხელწვდომის დროს ცვლადების შემადგენლობაში.

შენიშვნები

თქვენ შეგიძლიათ მიიღოთ დაწვრილებითი ინფორმაცია შემდეგი საკითხების ირგვლივ: როგორ უნდა მივიღოთ სიმბოლური ხელწვდომა STEP-7 (TIA Portal)-ში არასტრუქტურირებულ მონაცემთა ტიპებთან: ბიტ-ბითად, ბაიტ-ბაიტად, სიტყვასიტყვად, ანდა სიმბოლო-სიმბოლოდ? <u>https://support.industry.siemens.com/cs/ww/en/view/57374718</u>

3.7. ბიბლიოთეკები

TIA Portal-ში თქვენ შეგიძლიათ შექმნათ დამოუკიდებელი ბიბლიოთეკები პროექტის სხადასხვა კომპონენტებისაგან, რომლებიც შემდეგში შეიძლება იყოს გამოყენებული განმეორებით.

უპირატესობანი

- კონფიგურირებული კომპონენტების მარტივი შენახვა TIA Portal-ში:
 - მოწყობილობების (კონტროლერი, HMI, ამძრავი და სხვ.);
 - პროგრამების, ბლოკების, ცვლადების, დაკვირვების ცხრილების;
 - HMI-ს სახის, HMI-ს ტეგების, სკრიპტების, და სხვ.
- პროექტებს შორის გაცვლა ბიბლიოთეკის გავლით;
- ბიბლიოთეკის ელემენტების ცენტრალიზებული განახლების ფუნქცია;
- ბიბლიოთეკის კომპონენტების ვერსიების მართვა;
- შესაძლებელი შეცდომების წყაროების რაოდენობის შემცირება.

რეკომენდაციები

- შექმენით მასტერ ასლები ბლოკების, აპარატურის კონფიგურაციის, HMI გამოსახულებების და სხვათა განმეორებითი გამოყენების გამარტივებისათვის;
- შექმენით ტიპები მხარდაჭერილი სისტემისათვის ბიბლიოთეკის კომპონენტების განმეორებთი გამოყენებისათვის:
 - ბლოკების ვერსიების მართვისთვის;
 - პროგრამის ყველა ელემენტის ცენტრალიზებული განახლების ფუნქციისათვის.
- გამოიყნეთ გლობალური ბიბლიოთეკა სხვა მომხმარებლებთან გაცვლისათვის ანდა როგორც ცენტრალური ბაზა, რამდენიმე მომხმარებლის ერთდროული მუშაობისათვის;
- დააკონფიგურირეთ თქვენი გლობალური ბიბლიოთეკის შენახვის ადგილი ისეთნაირად, რომ ის ავტომატურად გაიღოს ხოლმე TIA Portal-ის გაშვების შემთხვევაში.

უფრო დაწვრილებითი ინფორმაცია ხელმისაწვდომია ბმულით: https://support.industry.siemens.com/cs/ww/en/view/100451450

შენიშვნა

თქვენ შეგიძლიათ მიიღოთ დაწვრილებითი ინფორმაცია შემდეგი საკითხების ირგვლივ:

STEP 7-ის (TIA Portal) და WinCC (TIA Portal)-ის რომელი ელემენტები შეგიძლიათ შეინახოთ ბიბლიოთეკაში როგორც ტიპი ანდა მასტერ ასლი? https://support.industry.siemens.com/cs/ww/en/view/109476862 როგორ გავხსნათ გლობალური ბიბლიოთეკა STEP 7 TIA Portal -ში ჩაწერის ხელწვდომის უფლებით? https://support.industry.siemens.com/cs/ww/en/view/37364723

3.7.1 ბიბლიოთეკის ტიპები და ბიბლიოთეკის ელემენტები

არსებობს ბიბლიოთეკის ორი ტიპი:

- "Project libtrary" (საპროექტო ბიბლიოთეკა);
- "Global library' (გლობალური ბიბლიოთეკა).

თვითოეული მათგანი შედგბა:

- "Types" (ტიპებისაგან);
- "Master Copies" (მასტერ ასლებისაგან) (იხ. სურათი 3-35:)

Pergant Belle blaus beamt Derlime Derlime Dank Hinderss Halle	Libraries	
POLICE STATES STATES	J Garder J Garder 加速度	POSTAL
Populate 14	Options	Laster 25.5
Benkers		Dolom
300 *	Library view	H step da
300 2		
+ 1 17/100 Linute VG 091	Project library	the second se
Add reservative Col	· moject monary	
A revises & seconds		- 1 (and a local state of the lo
 B RC, 1 (29) 91011 PMD9 		his eres tar
Device configuration	w 🛄 Resignst liber of	- 2111
Codina A diag tantita	• 12 Project library	EV103
- Stephen Marks		6,2103
Patrice interest	- in the state of	• X Max
THE PRESENT VIAN		67705
* Norma 1		Excess .
 Network interrupt 	• • ••••••••••••••••••••••••••••••••••	- 0.0000000
a 🔒 Technology shipe to		the cost of the co
 B and External cource their. 	-III) V 1.0.1	Contraction of the second s
 ACtion 	F 1/100	PONT DEPA
 A.L data rgan 	-EX A 1.0.0	 L. Butters and Contains.
P La Tana and an area area	T B Mator	- III Cong Factoria
a Teor		
Cl A.Coloma	E-V101	a Total and
kadin		• 12 har tolk services
a land mobiles	lii V 1.0.0	- Trans
 Do Historico 		and the second s
 Model (on state on) 	 Master copies 	- 4711
* 🔐 Earroran data		@V101
* E Zoczener leiten zeitengs	Copyorruc	BLV104
a 70 Paulien generaten		1 Down
The first decision	2 V Global libraries	a variante das
and the second second	 Global libraries 	
Le Le	- Draw inference	nd E
1.00	Buttons-and-Switches	
6	Allerey out devel.	
	P Carcong r bircours	
* Details view	Monitoring-and-control-objects	
See.	Documentation templates	
A left new deline	E Woac MP	
A Devices & messades		
4 M.J.	▼ JU User_Lib_Versions	2 Into (hope: Ibung)
4 Postal view III dvendese	Types	Uteres Son Lib. Version on sorred.
	Add new type	
		S.
	4Ey V 1.0.1	
	=E ₂ V 1.0.0	
	■ Mator	5
	M. Info (Glob all) both right and	and the set of the later

სურათი 3-35: ბიბლიოთეკა TIA Portal-ში

- (1) "Project libtrary" (საპროექტო ბიბლიოთეკა);
 - ჩაშენებულია და იმართება პროექტში;
 - შესაძლებლობას გვაძლევს განმეორებით გამოვიყენოთ კომპონენტები პროექტში.
- (2) "Global library' (გლობალური ბიბლიოთეკა):
 - დამოუკიდებელი ბიბლიოთეკა;
 - შესაძლებელია გამოყენება სხვადასხვა პროექტებში.

ბიბლიოთეკა შედგება საბიბლიოთეკო ელემენტების ორი ტიპისაგან:

- (3) "Master Copies" (მასტერ ასლებისაგან):
 - საკონფიგურიაციო ელემენტების ასლები ბიბლიოთეკაში (მაგ. ბლოკები, აპარატული კონფიგურაცია, PLC-ს ცვლადების ცხრილები და სხვ);
 - ასლები, რომლებიც არ არიან კავშირში ელემენტებთან პროექტში;

- მასტერ ასლები შეიძლება შედგებოდეს კიდევ რამოდენიმე
 საკონფიგურიაციო ელემეტებისაგან.
- (4) "Types" (ტიპებისაგან)
 - ტიპები დაკავშირებულია თქვენს პროექტის კომპონენტების გამოყენების ადგილებთან. ტიპების ცვლილებისას, მათი გამოყენების ყველა ადგილი პროექტში შეიძლება იყოს ავტომატურად განახლებული;
 - მხარდაჭერილი ტიპები: ბლოკები (FC, FB), PLC data types, HMI-ს სახეები, HMI-ს წინა პანელები, HMI UDT, სკრიპტები);
 - დაქვედებარებული ელემენტები ავტომატურად განიცდიან ტიპიზაციას;
 - ყოველი ტიპი იღებს თავის ვერსიას: ცვლილებები შეიძლება მოხდეს
 მხოლოდ ახალი ვერსიების შექმნის დროს;
 - კონტროლერში შეიძლება გამოყენებულ იქნას ტიპის მხოლოდ ერთი ვერსია.

3.7.2. ტიპობრივი კონცეფცია

ეს კონცეფცია შესაძლებლობას გაძლევთ თქვენ შექმნათ ავტომატიზსაციის სტანდარტიზირებული ფუნქციები, რომლებიც შეგეძლებათ გამოიყენოთ რამოდენიმე სხვადასხვა მოწყობილობაში. ტიპიური კონცეფშია შესაძლებლობას გაძლევთ აგრეთვე შექმნათ ფუნქციების ახალი ვერსიები და განახლებები. თქვენ შეგიძლიათ გამოიყენოთ ტიპები სამომხმარებლო პროგრამაში, ამასთან თქვენ იღებთ შემდეგ უპირატესობებს:

უპირატესობები

- ყველა კომპონენტის ცენტრალიზებული განახლება პროექტში;
- შეუძლებელია არასასურველი ცვლილებები ტიპების გამოყენების ადგილებში;
- სისტემა გარანტიას იძლევა ტიპების კონსისტენტობისა, არასასურველი
 ოპერაციების თავიდან მოშორების გაძნელების გზით;
- ტიპის თავიდან მოშორების საჭიროებისას, ის თავიდან იქნება მოშორებული ყველა იმ ადგილისაგან, სადაც იყო გამოყენებული პროგრამაში მანამდე.

თვისებები

ტიპის გამოყენებისას, თქვენ შეგიძლიათ ცვალოთ აღნიშნული ელემენტი ცენტრალიზებულად და ეს ცვლილებები გამოყენებული იქნება პროექტის მთელ პროგრამაში (იხ. სურათი 3-36:).



სურათი 3-36: ტიპიზაცია სამომხმარებლო ბიბლიოთეკების დახმარებით

 უკეთესი იდენტიფიკაციის საჭიროებისათვის ტიპები ყოველთვის მოინიშნება პროექტში.

3.7.3 განსხვავება ტიპიზირებულ ობიექტებს შორის CPU-სა და HMI-სთვის

არსებობს სისტემური განსხვავებები ტიპიზირებულ ობიექტებს შორის კონტროლერებისა და HMI-ებისათვის (იხ. ცხრილი 3-9:)

ცხრილი 3-9: ტიპების განსხვავებულობა კონტროლერისა და HMI-სათვის

კონტროლერი	HMI
მართვის დაქვემდებარებული	დაქვემდებარეული ელემენტები
ელემრენტები ტიპიზირდებიან	·HMI-ის არ ტიპიზირდებიახ
მართვის დაქვმდებარებული	დაქვემდებარებული ელემენტები
ელემენტები იყენებენ ეკზემპლიარებს	HMI არ გამოიყენებენ ეკზემპლიარებს
მართვის ელემნტები	HMI გამოსახულებები და სკრიპტები
რედაქტირებიან ტესტურ რეჟიმში	რედაქტირდებიან გაწყობის
	გარემოში. წინა პანელები და HMI -
	UDT-ები რედაქტირდება პირდაპირ
	ბიბიოთეკაში არა ტესტურ რეჯიმში.

3.7.4 ბლოკის ვერსიის შექმნა

მაგალითი: ტიპის შექმნა

შემდეგ მაგალითში ნაჩვენებია ძირითადი ფუნქციების გამოყენება ბიბლიოთეკების ტიპების გამოყენებისას (იხ. ცხრილი 3-10:).

ცხრილი 3-10: ტიპის შექმნა



ბიჯი	ინსტრუქცია
3.	განსაზღვრეთ ახალი ცვლადი შექმნილი მონაცენთა ტიპის მიხედვით. ამასთან PLC data Type გამოყენებულ იქნება როგორც დაქვემდებური ფუნქციონალური ბლოკის მიმართ.
	Image: Solution of the set
4.	გადაიტანეთ ფუნქციონალური ბლოკი drag & drop-ის დახმარებით, საპროქტო ბიბლიოთეკის "Types" (ტიპი) განყოფილებაში.
5.	ນ ຫຼິດການ ນາຍິດການ ນາຍິດການ ນາຍິດ ນາຍ
ბიჯი	ინსტრუქცია
6.	PLC data type შაბლონის გამოყენება ავტომატურად შენახული იქნება ბიბლიოთეკაში Project library Project library Add new type * Block_1 Full Vec_deta_type_1 * West copts

მაგალითი: ტიპის ცვლილება (იხ. ცხრილი 3-11:).

ცხრილი 3-11 : ტიპის ცვლილება

ბიჯი	ინსტრუქიცია
1.	დააჭირეთ თაგვის მარჯვნა ღილაკით ბლოკზე "Projt Library" (საპროექტო ბიბლიოთეკა) და აირჩიეთ "Edit Type" Ubraries Options El Ubraries
	Project library Proj
2.	აირჩიეთ თუ რომლი კონტროლერი იქნება გამოყენებული როგორც ტესტური, ფუნქიონირების შესამოწმებლად და "OK"
	თუ პროექტში ერთდროულად რამოდენიმე კონტროლერი იყენებს არჩეულ ბლოკს, მაშინ აუცილებელია არჩეულ იქნას საჭირო კონტროლერი როგორც ტესტური

ბიჯი	იწსტრ	უქცია		
3	ბიბლიოთკის ფანჯრის გაღბისას, შიქმწება ბლოკის ახალი ვერსია			
0.		test" (ambamaa)	,	0
	ଷ୍ଟ୍ର ଚନ୍ଦ୍ରର ପର୍କପରପର୍ବାଳିକ ଚନ୍ଦ୍ରପର୍ବର ଆ ।	test (ປຕາຖິບ(ບົງບຣ).		
	Project East view insert Online Options loois window	Help III III 🖳 🛄 🍠 Go online 🖉 Go off	ine 🛵 🖪 🖪 😽 F	e ni na na
	Ubraries	E 4 \$7-1500_Libraries_V12_SP	+ PLC_1 [CPU 1516-	3 PN/DP] + Progr
	Options			
	Dany view	🖻 से से 🕫 👻 🍆 🚍 🚍	9-21-2 E (io 68 원 😵
	Project library	Interface		
		1 🚭 🗕 Input	Data type	Default value
		2 💶 🔹 💌 N_1	"User_data_type_1"	
	Add new type	3 CI • DN_DPP 4 CI • Speed	Bool	false
	Filosk_1	5 💶 · Position	Real	0.0
	W V0.0.1	6 1 h_2	Word	1640
	- BUser_dots_type_1	8 🕣 - Output		
	Image: Second	a cidd news		
		⊣⊢⊣⊩-0- ഈ ษ ·	-	
	Global libraries	▼ Block title:		
	Duttons-and Switches	Comment		
	Long Functions	 Network 1: 		
	Monitoring-end-control-objects	Comment		
	► U WINKC_MP			
	► User_Lib_Versions			
	NR white P ← → → → → → → → → → → → → → → → → → →	Anterian Norreson Nor	იებები ბლი	ოკში,
	დაამთავრებთ ბლოკის ტესტი	რებას, გადადით შ	ემდეგ საფ	ენაც თე ეხურზე
5.	დააჭირეთ ღილაკს "Release ver	sion" (გამოუშვით შ	ბლოკის ვერ	რსია).
	S7-1500_Libraries_V12_SP1 → PLC_1 (CPU 1516-3 PWOP)	E Program bits		

ბიჯი	ინსტრუქცია
6.	みらの空りなら დიალოგური ფანჯარა. აქ თქვნ შეგიმლიათ დაწეროთ კომენტარები მიმდინარე ვერსიის შესახებ და დააჭიროთ "OK". Intervention will be released type version. A new version will be released for the selected types. Assign there common properties to credit the the commended properties. Name of type: Block_1 Version: 00/2 Author: User
	Options Update instances in the project Delete unused type versions from the library
	თუ კი ბლოკი გამოიყნება პროგრამის რამდენიმე ადგილას პროქტის სხვაასხვა კონტროლერებში, მაშინ თქვენ შგეძლებათ ერთდროულად განაახლოთ ყველა ბლოკი: "Update instances in the projet" (განაახლეთ ეკზემპლიარები პროექტში). თუ ელემენტის ძველი ვერსიები უკვე აღარ არიან საჭირო, თქვენ შეგიძლიათ მოაშოროთ ისინი "Delete unused type version from library" (მოაშორეთ გამოუყენებელი ტიპები
7.	လွှေသင်္က လွှေသင်္ငတွေ တရားလေး စက်က္လာလာတ္လေးလေး စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှေလေးသို့ စက်က္လာတာလွှဲကို စက်က်က္လာတာလွှဲကို စက်က်က္လာတာလ စက်က္လာတာလ စက်က်က်က်က်က်က်က စက်က်က်က်က်က်က်က်က်က်က

3.8 მწარმოებლობის გაზრდა აპარატული წყვეტების დახმარებით

სამომხმარებლო პროგრამის შესრულების პროცესი შეიძლება შეწყვიტოს რაიმე მოვლენამ, მაგალითად აპარატულმა წყვეტამ. როდესაც თქვენთვის აუცილებელია პროცესორის სწრაფი გამოხმაურება აპარატულ წყვეტაზე (მაგალითად, სიგნალის დადებითი ფრონტის გამოჩენა შესასვლელ დისკრეტულ არხზე), მაშინ აუცილებელია მისი კონფიგურირება.

თვითოეული ასეთი წყვეტისთვის შეიძლება დაპროგრამებულ იქნას თავისი ქვეპროგრამა (OB). აღნიშნული OB გამოძახებულ იქნება პროცესორის მიერ ასეთი მოვლენის მოხდენის დროს. ამასთან ერთად, კონტროლერის ნორმალური მუშა ციკლი შეჩერდება და გაგრძელდება შემდგომში ამ აპარატული წყვეტის დამუშავების შემდეგ (იხ. სურათი 3-37:).



სურათი 3-37: წყვეტის გამოჩენისას ხდება ქვეპროგრამის გამოძახება (OB)

შემდეგ გამოსახულებაზე თქვენ შეგიძლიათ ნახოთ "აპარატული წყვეტის" კონფიგურაცია ციფრული შესასვლელი მოდულისათვის აპარატული კონფიგურაციის რედაქტორში (იხ. სურათი 3-38:).



სურათი 3-38: აპარატული წყვეტის კონფიგურაცია

უპირატესობანი

- მოვლენებზე სისტემის სწრაფი რეაქცია (გასვლა ზედა ან ქვედა ზღვრებიდან, სიგნალის ფრონტის გამოჩენა და სხვ.);
- თვითოეული ამ მოვლენისათვის შეიძლება გამოძახილ იქნას თავისი საკუთარი ქვეპროგრამა (OB).
- •

რეკომენდაციები

- გამოიყენეთ პროცესის წყვეტები პროგრამაში აპარატულ წყვეტებზე სწრაფი რეაგირებისათვის;
- თუ კი სისტემის გამოხმაურება არასაკმარისად სწრაფია, მიუხედავად აპარატული წყვეტის დაპროგრამირებისა, თქვენ აგრეთვი შეგიძლიათ გააუმჯობესოთ სისტემის რეაქცია. ამისათვის დააყენეთ საუკეთესო დაყოვნება "Input delay" მოდულისათვის. ხდომილებაზე გამოძახილი შესრულებული იქნება მხოლოდ მოცემული დროის დაყოვნების შემდეგ. აღნიშნული პარამეტრი გამოიყენება შესასვლელი სიგნალის ფილტრაციისათვის, მაგალითად, ისეთი შეცდომების თავიდან აცილების მიზნით, როგორიცაა კონტაქტების ყანყალი (იხ. სურათი 3-39:).



სურათი 3-39: დაყოვნების დაყენება შესასვლელისათვის

3.9 დამატებითი რეკომენდაციები მწარმოებლობის გაზრდისათვის

აქ თქვენ შეგიძლიათ ნახოთ რამდენიმე სასარგებლო რეკომენდაცია, რომლებიც გაზრდიან თქვენი კონტროლერის მწარმოებლობას.

რეკომენდაციები

S7-1200/1500 კონტროლერების მწარმოებლობის გაზრდისათვის გამოიყენეთ შემდეგი რჩევები:

- LAD/FBD: გამორთეთ "generate ENO" ბლოკებში. შესრულების აქტიური რეჟიმისათვის დეაქტივირებული იქნება შემოწმება;
- STL: არ გამოიყენოთ რეგისტრები, რადგან სამისამართო რეგისტრები და მონაცემთა რეგისტრები ემულირდებიან S7-1500-ში, შეთავსებადობის მიზნებისთვის.

შენიშვნა

თქვენ შეგიძლია მოძებნოთ დამატებითი ინფორმაცია შემდეგი საკითხების ირგვლივ: რა გზით შეიძლება მოახდინოთ ENO-ს გამოსასვლელის დეაქტივირების მართვა ინსტრუქციებისათვის?

https://support.industry.siemens.com/cs/ww/en/view/67797146

რა გზითაა შესაძლებელი მწარმოებლობის გაზრდა STEP 7-ში (TIA Portal) S7-1200/ S7-1500-ის CPU-ებისათვის?

https://support.industry.siemens.com/cs/ww/en/view/37571372

დაპროგრამების ენა SCL: რჩევები და რეკომენდაციები

3.10.1 გამოძახების შაბლონების გამოყენება

დაპროგრამების ენის მრავალი ინსტრუქცია შეიძლება წარმოდგენილ იყოს როგორც უკვე დაწერილი ინსტრუქციების გამოძახების შაბლონის სახით, რომელსაც გააჩნია ფორმალური პარამეტრები.

მაგალითი (იხ. ცხრილი 3-12:)

ბიჯი	ინსტრუქცია			
1.	Jogon Acid Acid Acid Acid Acid Acid Acid Acid			
2.	ახლა აუცილებელია შეავსოთ აუცილებელი პარამეტრები და ამის შემდეგ დააჭიროთ ღილაკს "Return" (დაბრუნება). Image: Control of the type Image: Contype Image: Control of the type			
3.	რედაქტორი ავტომატურად მოაშორებს ზედმეტობებს გამოზახების შაბლონისაგან.			
4.	თუ თქვენ გსურთ დააბრუნოთ ინსტრუქციის სრული გამომახება, შეასრულეთ შემდეგი პროცედურა. მათავსეთ თაგვის კურსორი გამოზახებაზე და დააჭირეთ "CTRL+SHIFT+SPACE". ახლა თქვენ იმყოფებით რეჟიმში Call Template. რედაქტორი განმეორებით გააფართოვებს გამომახებას. ღილაკით "TAB" თქვენ შემლებთ შეასრულოთ გადასვლა პარამეტრებს შორის			
5.	შენიშვნა: "CALL Template" რეჯიმში შეყვანა სრულდება კურსივით.			

ცხრილი 3-12: შაბლონის მარტივი გაფართოვება

3.10.2 ინსტრუქციის რომელი პარამეტრებია აუცილებელი?

თუ კი თქვენ გააფართოვეთ გამოძახების შაბლონი, მაშინ პარამეტრების ფერის მიხედვით, თქვენ უნდა მიხვდეთ თუ რომელი ფორმალური პარამეტრია აუცილებელი ფაქტიური პარამეტრის მიცემისათვის და რომელი არა. აუცილებელი პარამეტრები გამორჩეულნი იქნებიან მუქი ფერით.

3.10.3 ცვლადების სახელების გადატანა

SCL რედაქტორში, თქვენ შეგიძლიათ გამოიყენოთ გადაცოცების ფუნქციით drag&drop, ასევე შეიძლება გადატანილ იყოს ცვლადების სახელებიც. ერთი ცვლადის შეცვლისათვის მეორე ცვლადით, შეასრულეთ შემდეგი სახის პროცედურა (იხ. ცხრილი 3-13:)



ცხრილი 3-13: ცვლადების გადატანა SCL-ში.

3.10.4 ციკლების FOR, REPEAT და WHILE გამოყენება

ციკლებთან მუშაობის დროს არსებობს სამი სახის ინსტრუქცია. შემდეგ მაგალითებზე ნაჩვენებია მირითადი განსხვავებები.

თვისება: ციკლი FOR

ციკლი FOR სრულდება განსაზღვრული რაოდენობის ჯერადად. დასაწყისში მთვლელი დგება საწყის მდგომარეობაში. შემდეგ, ციკლის ყოველი გატარების დროს მთვლელი იზრდება მოცემული ბიჯით მანამ, სანამ არ მიაღწევს საბოლოო მნიშვნელობას.

მაღალი მწარმოებლობის მიღწევის მიზნისთვის საწყისი და საბოლოო მნიშვნელობების გაანგარიშება ხდება ერთხელ ციკლის დასაწყისში. ამის შემდეგ მთვლელის მნიშვნელობა უკვე აღარ ახდენს ზეგავლენას კოდზე ციკლის სხეულში.

სინტაქსისი

FOR counter :=start_count TO end_count DO // ციკლის სხეული ;

END_FOR;

EXIT ბრძანების გამოყენებით ციკლი შეიძლება შეწყვეტილ იქნას დროის ნებისმიერ მომენტში.

თვისება: ციკლი WHILE

ციკლი წინაპირობით WHILE, წყვეტს თავის მუშაობას მაშინვე როცა ციკლის პირობამ შეწყვიტა შესრულება. ციკლის დამთავრების პირობა მოწმდება ციკლის სხეულამდე. ამგვარად, ციკლის შესრულება არ მოხდება თუ კი პირობა არ კმაყოფილდება. ციკლის სხეულში ნებისმიერი ცვლადი შეიძლება მომზადებულ იქნას შემდეგი ციკლისათვის.

სინტაქსისი

WHILE condition DO // ციკლის სხეული ; END_WHILE;

თვისება: ციკლი REPEAT

ციკლი პოსტპირობით REPEAT, წყვეტს თავის მუშაობას მაშინვე როდესაც ციკლის პირობამ შეწყვითა შესრულება. ციკლის დამთავრების პირობა მოწმდება ციკლის სხეულის შემდეგ. ამგვარად, მოცემული ციკლი შესრულდება მინიმუმ ერთჯერ. ციკლის სხეულში თვითოეული ცვლადი შეიძლება მომზადდეს შემდეგი ციკლისათვის.

სინტაქსისი

REPEAT

// ციკლის სხეული ; ANTIL condition END_REPEAT;

რეკომენდაციები

- თუ კი მთვლელის ტეგის მნიშვნელობა წინასწარაა ცნობილი, მაშინ გამოიყენეთ FOR;
- თუ კი მთვლელის ტეგი ანდა ციკლის გაგრძელების პირობა შეიძლება კორექტირებულ იქნას ციკლის შესრულების პროცესში, მაშინ გამოიყენეთ WHILE ანდა REPEAT.

3.10.5 CASE ინსტრუქციის გამოყენება

CASE ინსტრუქციის დახმარებით SCL ენაში, პროგრამას შეუძლია გადასვლა პროგრამის ერთერთ მონაკვეთზე თუ შესრულდება შესაბამისი პირობა. ამის შემდეგ CASE ისტრუქციის შესრულება დამთავრებული იქნება. ეს მექანიზმი, მაგალითად, შესაძლებლობას გაძლევთ თქვენ ხშირად შეამოწმოთ მნიშვნელობათა აუცილებელი დიაპაზონი.

მაგალითი

CASE #myVar OF 5: FC5 (#myParam) ; 10, 12: FC10 (#myParam) ; 15: FC15 (#myParam) ; 0 ..20: FCGlobal (#myParam) ; // FCGlobal არასდროს არ გამოიძახება 5, 10, 12 ან 15 ის მნიშვნელობებისათვის ! ELSE END_CASE ;

შენიშვნა

CASE ინსტრუქციები ასევე მუშაობენ CHAR, STRING მონაცემთა ტიპებთან და მათ ელემენტებთან (იხ. მაგალითი 2.8.5 თავში. მონაცემთა ტიპი VARIANT).

3.10.6 მთვლელის ქცევა ციკლებისათვის FOR

FOR ციკლები SCL ენაზე მუშაობენ მხოლოდ და მხოლოდ მთვლელების დახმარებით, ანუ იტერაციის რაოდენობა ფიქსირებულია. ციკლში FOR მთვლელის მნიშვნელობა შეუძლებელია შეცვლილ იქნას. EXIT ინსტრუქციის გამოყენებით ციკლი შეიძლება შეწყვეტილ იქნას დროის ნებისმიერ მომენტში.

უპირატესობანი

 პროგრამის ოპტიმიზაცია კომპილიატორით სრულდება უფრო ეფექტურად, როდესაც მთვლელის მნიშვნელობა წინასწარაა ცნობილი.

მაგალითი

FOR # var := #lower TO #upper DO #var := #var + 1; //არაკორექტულია, კომპილიატორი -> გაფრთხილება END_FOR;

3.10.7 ციკლი FOR უკუ მიმართულებით

SCL ენაზე თქვენ შეგიძლიათ გაზარდოთ მთვლელი უკუ მიმართულებით სხვა ბიჯით. ამისათვის გამოიყენეთ საკვანმო სიტყვა "BY" ციკლის პირობაში.

მაგალითი

FOR #var :=upper TO # lower BY -2 DO

END_FOR;

თუ თქვენ განსაზღვრეთ "BY" როგორც "-2", ისე, როგორც ეს ნაჩვენებია მაგალითში, მაშინ მთვლელი შემცირდება ორით ყოველი ციკლის გავლის შემთხვევაში. თუ კი თქვენ არ გამოიყენებთ საკვანძო სიტყვას "BY", მაშინ სიჩუმით "BY" მიცემული იქნება 1.

3.10.8 ეკზემპლიარების მარტივი შექმნა გამოძახებებისთვის

თუ თქვენ უპირატესობას ანიჭებთ კლავიატურასთან მუშაობას, მაშინ არის მარტივი ხერხი ეკზემპლიარების შექმნისა ბლოკებისათვის SCL ენაზე.

მაგალითი (იხ. ცხრილი 3-14:)

ცხრილი 3-14: ეკზემპლიარების მარტივი შექმნა

ბიჯი	ინსტრუქცია				
1.	მიეცით ბლოკს სახელი: "." (წერტილი)-ის შემდეგ. თქვენ დაინახავთ შემდეგს.				
	4 "Blook_1". Mileck_1_DB" Single instance Mame villeck No. ox-Open "Call option" dialo Dipen "Call option" dialo Name villeck Name villeck Name villeck Name villeck Name villeck Name villeck				
2.	ზევიდან ნაჩვენებია უკვე არსებული ეკზემპლიარები. მათზე დამატებით თქვენ შეგიძლიათ შექმნათ ახალი ეკზემპლიარი ანდა მულტიეკზემპლიარი. იმისათვის, რომ გადახვიდეთ შესაბამის ჩანაწერებში კომპილიაციის ფანჯარაში გამოიყენეთ *s* ან *m*.				

3.10.9 ცვლადების დამუშავება Time (დრო) მონაცემთა ტიპით

თქვენ შეგიძლიათ იმუშავოთ Time მონაცემთა ტიპის ცვლადებით SCL ენაში, ისევე როგორც ჩვეულებრივი მნიშვნელობების შემთხვევაში. თქვენ არ დაგჭირებათ დამატებითი ფუნქციები, ისეთები როგორიცაა მაგალითად T_COMBINE, მაგრამ თქვენ შეგიძლიათ გამოიყენოთ მარტივი არითმეტიკული ოპერაციები. ასეთ მექანიზმს ეწოდება "ოპერატორების გადატვირთვა". SCL კომპილიატორი ავტომატურად იყენებს საჭირო ფუნქციებს. თქვენ შეგიძლიათ გამოიყენოთ აუცილებელი არითმეტიკული

ინსტრუქციები Time (დრო) მონაცემთა ტიპისათვის, ამასთან პროგრამირების ეფექტურობა იზრდება.

მაგალითი

TimeDifference := timeStamp1 - timeStamp2;

შემდეგ ცხრილში ნაჩვენებია გადატვირთული ოპერატორების შესაბამისობა და ის ფუნქციები, რომლებიც სრულდება ფაქტით (იხ. ცხრილი 3-15:):

გადატვირთული ოპრერანდი	ოპერაცია
Itime + time	T_ADD LTime
Itime + time	T_SUB LTime
Itime + lint	T_ADD LTime
Itime + lint	T_SUB LTime
time + time	T_ADD Time
time + time	T_SUB Time
time + dint	T_ADD Time
time + dint	T_SUB Time
Itod + time	T_ADD LTOD / Time
Itod + time	T_SUB LTOD / Time
tod + time	T_ADD TOD / Time
tod + time	T_SUB TOD / Time
tod + dint	T_ADD TOD / Time
tod + dint	T_SUB TOD / Time
dt + time	T_ADD DT / Time
dt + time	T_SUB DT / Time
ldt – ldt	T_DIFF LDT
dtl – dtl	T_DIFF DTL
dt – dt	T_DIFF DT
date – date	T_DIFF DATE
Itod – Itod	T_DIFF LTOD
date + Itod	T_COMBINE DATE / LTOD
date + tod	T_COMBINE DATE / TOD

ცხრილი 3-15: გადატვირთული ოპერანდები SCL-ში.

4. აპარატულად-დამოუკიდებელი დაპროგრამება

იმისათვის, რომ დარწმუნებული ვიყოთ, რომ ბლოკის გამოყენება შეიძლება ნებისმიერი კონტროლერის შემთხვევაში, აუცილებელია უარი ვთქვათ ეგრეთ წოდებულ აპარატულად დამოკიდებულ ფუნქციაზე და თვისებებზე.

4.1 მონაცემთა ტიპები S7-300/400-სა და S7-1200/1500-ში

ქვემოთ ნაჩვენებია ელემენტარული მონაცემთა ტიპებისა და მონაცემთა ჯგუფების სია.

რეკომენდაცია

- გამოიყენეთ მხოლოდ ის მონაცემთა ტიპი, რომლებსაც მხარს უჭერს ყველა ტიპის კონტროლერები (იხ. ცხრილი 4-1:).
- •

	. აღწერა	S7 - 300/400	S7-1200	S7-1500
მონაცემთა ბიტური ტიპები	 BOOL BYTE WORD DWORD 	~	~	~
	LWORD	×	×	~
სიმვოლ. ტიპი	 CHAR (8 бит) 	~	✓	~
რიცხვითი მონაცემთა ტიპები	 INT (16 бит) DINT (32 бита) REAL (32 бита) 	~	~	~
	 SINT (8 бит) USINT (8 бит) UINT (16 бит) UDINT (32 бит)а LREAL (64 бита) 	×	~	*
	 LINT (64 бита) ULINT (64 бита) 	×	×	~
მოწაცემთა ტიპები, რომლებიც	TIME DATE TIME_OF_DAY	~	~	~
ასახავენ	S5TIME	✓	×	✓
დროს	LTIME		~	1

ცხრილი 4-1: EN 61131-3 სტანდარტთან შესაბამისი მონაცემთა ელემენტარული ტიპები

ცხრილი 4-2: მონაცემთა ჯგუფები, რომლებიც შედგება სხვა ტიპის მონაცემებისაგან

L_TIME_OF_DAY

٠

	აღწერა	S7 - 300/400	S7-1200	S7-1500
მონაცემთა ხიპიზი	DT (DATE_AND_TIME)	~	×	~
Grogood,	• DTL	×	~	~
ოოძლების აღნიშნავენ	LDT (L_DATE_AND_TIME)	×	×	*
დროს	STRING	✓	~	~
მასივი	ARRAY	✓	✓	~
სტრუქტურა	STRUCT	✓	✓	~

	აღწერა	S7 - 300/400	S7-1200	S7-1500
მაჩვენებე- ლე	POINTER ANY	~	×	✓ ¹⁾
	VARIANT	×	~	×
ბლოკები	TIMER COUNTER	~	✓ ²⁾	~
	BLOCK_FBBLOCK_FC	~	×	~
	BLOCK_DBBLOCK_SDB	~	×	×
	VOID	✓	✓	~
მომხმარებ-	PLC Data Type	~	~	~
ლის				
შაბლონი				

ცხრილი 4-3: ფორმალური პარამეტრების ტიპები, რომლებიც გამოიყენება ბლოკების მიერ

- ოპტიმიზირებული ხელწვდომის დროს შესაძლებელია მხოლოდ სიმვოლური დამისამარება;
- 2) S7-1200 და S7-1500 -ისთვის მონაცემთა ტიპები TIMER და COUNTER წარმოდგენილია როგორც ICE_TIMER და ICE_COUNTER.

4.2 მარკერებიდან მონაცემთა გლობალურ ბლოკებზე გადასვლა

უპირატესობანი

 ოპტიმიზირებული გლობალური DB-ები გაცილებით უფრო ეფექტურნი არიან, ვიდრე მერკერული მეხსიერება, რომელიც არ არის ოპტიმიზირებული თავსებადობის მიზნებისათვის.

რეკომენდაციები

მერკერული მეხსიერების გამოყენება (ასევე სისტემური ბიტებისა და სინქრობაიტების) შეიძლება იყოს შეცდომის წარმოშობის მიზეზი პროგრამაში, რადგანაც ამ ტიპის მეხსიერების განზომილება სხვადასხვა კონტროლერისთვის სხვადასხვაა. პროგრამის დაწერის დროს არ გამოიყენოთ მერკერული მეხსიერება, ნაცვლად ამისა იმუშავეთ მონაცემთა გლობალურ ბლოკებთან. ასეთ შემთხვევაში პროგრამები იქნება გაცილებით უნივერსალურნი.

4.3 "სინქრობაიტის" დაპროგრამება

რეკომენდაციები

სინქრობაიტთან მუშაობისათვის აუცილებელია შეასრულოთ მისი გააქტიურება კონტროლერის აპარატულ კონფიგურაციაში. გამოიყენეთ პროგრამული ბლოკი ტაქტური გენერატორის სახით. ქვემოთ თქვენ შეგიძლიათ ნახოთ მაგალითი სადაც დაპროგრამებულია ტაქტური გენერატორი დაპროგრამების SCL ენაზე.

მაგალითი

შექმნილ ბლოკს გააჩნია შემდეგი ფუნქციონალი. დაუდგინდება აუცილებელი სიხშირე. გამოსასვლელ "Q"-ს აქვს მონაცემთა ტიპი Bool, რომელიც შეიცვლება მიცემული სიხშირის მიხედვით. გამოსასვლელზე "Countdown" გამოვა მიმდინარე მდგომარეობის "Q" დარჩენილი დრო.

თუ მიცემული სიხშირე ნაკლებია ან ტოლია 0.0-ზე, მაშინ გამოსასვლელი Q=FALCE და Countdown=0.0



შენიშვნა

ეს მაგალითი თქვენ შეგიძლიათ გადმოქაჩოთ შემდეგი ბმულიდან: https://support.industry.siemens.com/cs/ww/en/wiew/87507915

5 STEP 7 Safety TIA Portal-do

5.1 შესავალი

უსაფრთხოების სისტემებისა და ხანმარსაწინააღმდეგო დაცვის კონტროლერები (Failsafe) S7-1200F და S7-1500F მხარდაჭერილია TIA Portal-ის V13 SP1-ის ვერსიაში. ასეთ კონტროლერებში, უსაფრთხოების პროგრამის დაპროგრამებასთან ერთად ხელმისაწვდომია აგრეთვე სტანდარტული ფუნქციები. უსაფრთხოების პროგრამის შექმნისათვის გამოიყენება პაკეტი SIMATIC STEP 7 Safety (TIA Portal) (იხ. სურათი 5-1:)



სურათი 5-1: სტანდარტული პროგრამა და უსაფრთხოების პროგრამა

უპირატესობანი

- უნივერსალური ინსტრუმენტი, როგორც სტანდარტული ასევე უსაფრთხოების პროგრამების შექმნისათვის არის TIA Portal-ი;
- დაპროგრამება LAD და FBD ენებზე;
- უნივერსალური ინსტრუმენტები დიაგნოსტიკისა და online ფუნქციებისათვის.

შენიშვნა

უსაფრთხოების პროგრამები არ იძლევა შეცდომების არარსებობის გარანტიას. პროგრამისტი აგებს პასუხს პროგრამირების სწორ ლოგიკაზე. უსაფრთხოების სისტემები და ავარიის საწინააღმდეგო დაცვა ნიშნავს, რომ კონტროლერში უზრუნველყოფილი იქნება სამომხმარებლო უსაფრთხოების პროგრამის სწორი დამუშავება.

შენიშვნა

დამატებითი ინფორმაცია უსაფრთხოების თემის შესახებ, მოთხოვნილებები უსაფრთხოების შესახებ ანდა უსაფრთხოების პროგრამის პრინციპები ხელმისაწვდომია:

TIA Portal- ყველაზე მეტად მნიშვნელოვანი დოკუმენტებისა და ბმულების მიმოხილვა;

<u>https://support.industry.siemens.com/cs/ww/en/view/90939626</u> Applications & Tools – Safety Integrated; <u>https://support.industry.siemens.com/cs/ww/en/ps/14675/ae</u> STEP 7 Safety (TIA Portal) - სახელმძღვანელო <u>https://support.industry.siemens.com/cs/ww/en/ps/14675/man</u>

5.2 ტერმინები

ამ დოკუმენტში გამოიყენება შემდეგი სახის ტერმინები (იხ. ცხრილი 5-1:).

ტერმინი	აღწერა
სტანდარტული სამომხ-	სტანდარტული სამომხმარებლო
მარებლო პროგრამა	პროგრამა, სადაც არ გამოიყენება
	პროგრამის ჩ ფუნქციები
უსაფრთნოების	უსაფრთხოვების სამომხმარებლო
პროგრამა (F პროგრამა)	პროგრამის დამუშავება ხდება ცალკე,
	კონტროლერისაგან დამოუკიდებლად.
	უსაფრთხოების ყველა ბლოკი და
	ინსტრუქციები აღნიშნულია ყვითელი
	ფერით სამომხმარებლო რედაქტორის
	ინტერფეისში (მაგ. პროექტის
	ნავიგატორში), სტანდარტული
	ბლოკებისა და უსაფრთხოების ბლოკების
	უფრო ადვილი გარჩევისათვის
	ერთმანეთისაგან.
	უსაფრთხოების პარამეტრები F-CPU და
	F-I/O მონიშნულია ყვითელი ფერით
	აპარატული კონფიგურაციის
	რედაქტორში.

ცხრილი 5-1: უსაფრთხოების ტერმინები

5.3 უსაფრთხოების პროგრამის ელემენტები

უსაფრთხოების პროგრამა ყოველთვის შედგება სამომხმარებლო ბლოკების ანდა სისტემის მიერ გენერირებული F ბლოკებისაგან და "Safety Integration"რედაქტორიგან (უსაფრთხოების მართვა) (იხ. ცხრილი 5-2:).

ცხრილი 5-2:



5.4. ჯგუფი F runtime

უსაფრთხოების პროგრამა ყოველთვის სრულდება "F-runtime" ჯგუფში დაწესებული ციკლით. "F-runtime" ჯგუფი შედგება "Fail-safe organization block" (უსაფრთხოების მაორგანიზებელი ბლოკი), რომელსაც ეწოდება "Main safety block" (პროგრამის უსაფრთხოების მირითადი ბლოკი). უსაფრთხოების ყველა სამომხმარებლო ბლოკის გამომახება ხდება "Main safety block"-ისაგან (იხ. სურათი 5-2:).

F-runtime group 1 [RTG1	1				
Fail-safe organization	block P	D-		Main safety block	•
Name	F08_1			Main_Safety [FB1]	
Event class	Cyclic interrupt	1.00			
Number	123	Φ			
Cycle time	100000	μs			
Phase shift	0	μs		I-DB	
Priority	12	٥		Main_Safety_D8 (D81)	•
F-runtime group					
	Warn cycle time	of the ru	intime group	120000	μs
	Maximum cycle time of the runtime group				μs
	DB for runtime	group cor	nmunication	(None)	
	Fruntime	group int	formation DB	RTG1SysInfo	

სურათი 5-2: "F-runtime" ჯგუფი "Safety administration" რადაქტორში (უსაფრთხოების მართვა)

უპირატესობანი

- Runtime ჯგუფები შესაძლებელია იყოს შექმნილი და კონფიგურირებული "Safety Administrator"-ში (უსაფრთხოების მართვა);
- F- ბლოკები Runtime ჯგუფებში იქმნება ავტომატურად.

თვისებები

• შესაძლებელია შეიქმნას ორამდე Runtime ჯგუფი.

5.5 F ხელმოწერა (F სიგნატურა)

ყველა ცალკეულ F კომპონენტს (სადგური, პერიფერია, ბლოკები) აქვს უნიკალური F ხელმოწერა. F ხელმოწერის დახმარებით შესაძლებელია ადვილად იპოვოთ მოწყობილობის F კონფიგურაცია, F ბლოკები ანდა მთელი სადგური, რომლებიც შეესაბამება ორიგინალურ სადგურს ანდა პროგრამას.

უპირატესობანი

• F ბლოკებისა და F კონფიგურაციების მარტივი და სწრაფი შედარება.

თვისებები

- F პარამეტრების ხელმოწერა (F პერიფერიის მისამართების გარეშე) ...
 - ცვლილებები მხოლოდ დანიშნული პარამეტრების გავლით;
 - რჩება უცვლელი PROFIsafe მისამართის ცვლილებისას. მიუხედავად ამისა,
 საერთო F ხელმოწერა მთელი სადგურისათვის განიცდის ცვლილებას;
- F ბლოკის ხელმოწერა იცვლება F ბლოკში ლოგიკის ცვლილებისას.
- F ბლოკის ხელმოწერა რჩება უცვლელი თუ შეიცვლება (იხ. სურათი 5-3:):
 - ბლოკის ნომერი;
 - ბლოკის ინტერფეისი;
 - ბლოკის ვერსია.

მაგალითი (იხ. სურათი 5-3:)

scription	Offline signature Tim	e stamp		
ellective F-signature	675C8803 7/29	9/2014 4:20:41 PM (UTC +2:00))	
olocks			<u> </u>	
Y All F-blocks	•	(2)	
Description	Used and compiled	Function in safety program	Offline signature	Time
🕶 🛃 Program blocks				1
5 FOB_1 [OB123]	Yes	F-OB	0x84427972	7/29/
5 FOB_2 [OB124]	Yes	F-OB	0xF6658D19	7/29/
Main_Safety_1 [FB1]	Yes	F-FB	0x61F8DE42	7/29/
Alain Safety 2 (EBO)	Vee	6.69	0x65ED5CB2	7/29/
wain_salety_z [roo]	Tes	P*PB	GND3ED3CD2	
Main_Safety_2 (180)	Yes	I-DB for F-FB	0x27E959F6	7/29
Main_Safety_D8_1 (DB1) Main_Safety_D8_1 (DB1) Main_Safety_D8_1 (DB3) Marameter	Yes Yes Manual assignment o	FFB FDB for F-FB	0x27E959F6	7/29/ 7/29
Main_Safety_DB_1 (DB1) Main_Safety_DB_1 (DB1) Main_Safety_DB_1 (DB1) F-monitoring time:	Yes Yes Manual assignment o	I-DB for F-FB	0x27E959F6	7/29/ 7/29/
Main_Safety_D8_1 (D81) Main_Safety_D8_1 (D81) Marameter F-monitoring time: F-source address:	Yes Yes Manual assignment of 150	If F-monitoring time	0x27E959F6	7/29/
Main_Safety_D8_1 (DB1) Main_Safety_D8_1 (DB1) Darameter F-monitoring time: F-source address: F-destination address:	Yes Yes Manual assignment of 150 1 65532	of F-monitoring time	0x27E959F6	7/29/
F-monitoring time: [F-source address: [F-parameter (without addresses); [Yes Yes Manual assignment of 150 1 65532 18133	If F-monitoring time	0x27E959F6	7/29/. 7/29
F-monitoring time: F-monitoring time: F-source address: F-barameter (without addresses): Behavior after channel fault:	Yes Yes Manual assignment of 150 1 65532 18133 Passivate channel	If F-monitoring time	0x27E959F6	7/29/

სურათი 5-3: F ხელმოწერების მაგალითები

- 1. სადგურის კოლექტიური F ხელმოწერა "Safety administration" რედაქტორში (უსაფრთხოების მართვა);
- ბლოკების F ხელმოწერა "Safety administration" რედაქტორში (უსაფრთხოების მართვა) (შესაძლებელია აგრეთვე გაიგოთ ბლოკის თვისებებიდან);
- F პარამეტრების ხელმოწერა "Device view" (მოწყობილობის ასახვა) "Device & Networks"-ში.

შენიშვნა

S7-1500F კონტროლერებისათვის შეიძლება მიღებულ იქნას კოლექტიური F ხელმოწერა პირდაპირ დისპლეიზე ან Web სერვერის გავლით.

5.6 PROFIsafe მისამართის დანიშვნა F-I/O-ზე

ნებისმიერ F-I/O მოწყობილობას აქვს PROFIsafe მისამართი F კონტროლერებთან იდენტიფიკაციისა და კომუტაციისათვის. PROFIsafe მისამართის დანიშვნის დროს შესაძლებელია ორი სხვადასხვა კონფიგურაცია (იხ. ცხრილი 5-3:).

	ცხრილი 5-3: F მისამართის დანიშვნა
ET 200M / ET 200S	ET 200MP / ET 200SP
(მისამართის ტიპი PROFIsafe 1)	(მისამართის ტიპი PROFIsafe 2)
PROFIsafe მისამართის დანიშვნა	PROFIsafe მისამართების
უშუალოდ მოდულზე DIL	დანიშვნა უნდა მოხდეს მხოლოდ
გადამრთველის დახმარებით	TIA Portal-ის გავლით
TIA Portal-ის მოწყობილობის	კონფიგურირებული PROFIsafe
კონფიგურაციაში და DIL	მისამართი ჩაიტვირთება
გადამრთველზე პერიფერიაში	ინტელექტუალურ მოდულში
PROFIsafe მისამართები უნდა იყოს	\$
ერთნაირები	

უპირატესობანი

- F მოდულების შეცვლა შესაძლებელია PROFIsafe მისამართის მიცემის გარეშე ET 200MP და ET 200SP მოწყობილობებზე. ინტელექტუალური კოდური მოდული რჩება ბაზურ მოწყობილობაში (BaseUnit), მოდულის შეცვლის დროს;
- მარტივი კონფიგურაცია, TIA Portal-ი შეგატყობინებთ PROFIsafe მისამართის არასწორ მიკუთვნებაზე;
- ყველა F მოდულისათვის PROFIsafe მისამართები შეიძლება დამიშნულ იქნას ერთდროულად ET 200SP-თან.

შენიშვნა

უფრო დაწვრილებითი ინფორმაცია PROFIsafe მისამართების დანიშვნის შესახებ F-I/O მოწყობილობებისთვის ხელმისაწვდომია:

SIMATIC Industrial Software SIMATIC Safety-ში - კონფიგურაცია და დაპროგრამება https://support.industry.siemens.com/cs/ww/en/view/54110126

5.7 F - პერიფერიის შეფასება

ცვლადები F-I/O DB-ში ანდა მნიშვნელობათა მდგომარეობები PAE-ში	F-I/O S7-300/400-დან	F-I/O S7-1200/1500F-დაర
ACK_NEC	✓	✓
QBAD	✓	✓
PASS_OUT	~	✓
QBAD_I_xx *	✓	×
QBAD_O_xx *	✓	×
მნიშვნელობათა მდგომარეობები	x	~

ცხრილი 5-4: ცვლადები F-I/O DB-ში S7-300F/400F DA S7-1500F

F-I/O მოწყობილობების ყველა მდგომარეობა შეინახება F-I/O ბლოკებში. უსაფრთხოების პროგრამაში მდგომარეობები შეიძლება გაანალიზებულ იქნას და დამუშავებული. არის შემდეგი სახის განსხვავევები S7-1200F/1500F და S7-300F/400F-ს შორის (იხ. ცხრილი 5-4:).

* Q BAD_I_xx DA QBAD_O_xx გატყობინებენ თქვენ არხის მნიშვნელობის სისწორეზე და მიუთითებენ S7-1200F/1500F-ის ინვერტირებულ მნიშვნელობებზე (უფრო დაწვრილებითი ინფორმაცია ხელმისაწვდომია შემდეგ თავში).

5.8. მნიშვნელობათა მდგომარეობები (S7-1200F/1500F)

დიაგნოსტიკურ გზავნილებთან, მდგომარეობებთან და შეცდომების ასახვებთან ერთად F მოდული წარმოგვიდგენს ასევე ინფორმაციას თვითოეული შესასვლელი და გამოსასვლელი სიგნალის ნამდვილობის შესახებ- მნიშვნელობების მდგომარეობები. მნიშვნელობის მდგომარეობები შეინახება იგივენაირად, როგორც შესასვლელი სიგნალი-ასახვის არეში: მნიშვნელობათა მდგომარეობა გვეუბნება არხის მნიშვნელობების კორექტულობაზე (იხ. ცხრილი 5-5:):

- 1: არხში იმყოფება კორექტული მნიშვნელობა;
- 0: არხში იმყოფება არაკორექტული მნიშვნელობა.

ცხრილი 5-5: განსხვავებები Q_BAD (S7-300F/400F)-ებსა და (S7-1200F/1500F)-ებს შორის

სცენარი	QBAD (S7-300F/400F)	მწიშვნელობათა მდგომარეობა (S7-1200/1500F)
კორექტული მნიშვნელობა F-I/O (შეცდომები არ არის)	FALSE	TRUE
არხზე შეცდომებია	TRUE	FALSE
შეცდომის წასვლის ინფორ- მაცია არხიდან (ACK_REQ)	TRUE	FALSE
არხის (ACK_REI) შეცდომის კვიტირება	FALSE	TRUE

თვისებები

 მნიშვნელობების მდგომარეობა ჩაიწერება შესასვლელებისა და გამოსასვლელების ასახვის არეში;
 ხელწვდომა არხის მნიშვნელობასა და მნიშვნელობათა მდგომარეობებს F-I/O შორის უნდა შესრულებულ იყოს ერთი F runtime ჯგუფით.

რეკომენდაციები

 პროგრამის კიდევ უფრო ადვილი წაკითხვისათვის, ცვლადის ბოლოში უნდა დაამატოთ "VS", მაგალითად "Tag_In_1_VS" სიმბოლური სახელის სახით მნიშვნელობათა მდგომარეობებისთვის (Value Status).

მაგალითი

ბიტური მდგომარეობების მნიშვნელობების მდგომარეობა ასახვის არეში F-DI 8x24VDS HF მოდულის მაგალითზე (იხ. ცხრილი 5-6:).

ცხრილი 5-6: მნიშვნელობათა მდგომარეობების ბიტები ასახვის არეში	F-DI 8x24VDS-ს
მსაგალითზე	

ბაიტი	ბიტების დანიშვნა F-CPU-ში							
F-CPU	7	6	5	4	3	2	1	0
x + 0	DI ₇	DI ₆	DI ₅	DI ₄	DI ₃	Dl ₂	DI ₁	DI
x +1	მნიშვნე- ლობათა მდგომა- რეობა DI7- სთვის	მნიშვნე- ლობათა მდგომა- რეობა DI6- სთვის	'მნიშვნე- ლობათა მდგომა- რეობა DI5- სთვის	მნიშვნე- ლობათა მდგომა- რეობა DI4- სთვის	მნიშვნე- ლობათა მდგომა- რეობა DI3- სთვის	მნიშვნე- ლობათა მდგომა- რეობა DI2- სთვის	მნიშვნე- ლობათა მდგომა- რეობა DI1- სთვის	მნიშვნე- ლობათა მდგომა- რეობა DIo- სთვის

x = მოდულის საწყისი მისამართია

შენიშვნა

უფრო დაწვრილებითი ინფორმაცია ET 200SR სადგურის ყველა მოდულის მდგომარეობათა შესახებ ხელმისაწვდომია: სახელმძღვანელოში F-CPU-ს შესახებ <u>https://support.industry.siemens.com/cs/ww/en/ps/13719/man</u> სახელმძღვანელოში პერიფერიული მოდულების შესახებ F I/O <u>https://support.industry.siemens.com/cs/ww/en/ps/14059/man</u>

5.9. მონაცემთა ტიპები

უსაფრთხოების პროგრამებში S7-1200F/S71500F -ისთვის შეიძლება გამოყენებულ იქნას შემდეგი ტიპის მონაცემები (იხ. ცხრილი 5-7:).

ტიპი	ზომა	მნიშვ. დიაპაზონი
BOOL	1 ბიტი	01
INT	16 ბიტი	-32.768 32.767
WORD	16 ბიტი	-32.768 65.535
DINT	32 ბიტი	-2.14 2.14 მილიონი
TIME	32 ბიტი	T#-24d20h31m23s648ms T#+24d20h31m23s647ms- მდე

ცხრილი 5-7: მონაცემთა მთელრიცხვა ტიპები

5.10 PLC data type შაბლონები F-პროგრამებისთვის

უსაფრთხოების პროგრამებისთვის ასევე ხელმისაწვდომია სტრუქტურირებული მონაცემთა ტიპები PLC data types.

უპირატესობანი

 ცვლილებები PLC data type-ში ავტომატურად შედის ძალაში მომხმარებელთა პროგრამის ყველა ადგილებში.

თვისებები

- შაბლონები F-PLC data types გამოცხადდება და გამოიყენება ისევე, როგორც PLC data types;
- შაბლონებს F-PLC data types შეუძლიათ იმუშავონ ყველა ტიპის მონაცემებთან, რომლებიც შესაძლებელია გამოყენებულ იქნას უსაფრთხოების პროგრამაში;
- F-PLC data types შეუძლებელია გამოყენებულ იქნას სხვა F-PLC data types შაბლონების შიგნით;
- სტანდარტული სამომხმარებლო პროგრამების PLC data types შაბლონები
 შეიძლება გამოყენებულ იქნას უსაფრთხოების პროგრამაში ისევე, როგორც
 სტანდარტულ პროგრამაში.

რეკომენდაციები

- შეყვანა/გამოყვანის არეებში დაშვებისათვის ხელწვდომისათვის, აუცილებელია F-PLC data types შაბლონების გამოყენება (როგორც 3.6.4 თავში);
- დაცული უნდა იყოს შემდეგი სახის წესები:
- F-PLC data types შაბლონის ცვლადების სტრუქტურა უნდა ემთხვეოდეს F-I/O არხის სტრუქტურას;
- F-PLC data types შაბლონი 8 არხიანი F-I/O -სთვის, მაგალითად:
 - 8 BOOL ტიპის ცვლადებისთვის (არხის მნიშვნელობა);
 - 16 BOOL ტიპის ცვლადებისთვის (არხის მნიშვნელობა + მნიშვნელობათა მდგომარეობა).
- F-I/O თან ხელწვდომა შეიძლება შესრულებულ იქნას მხოლოდ
 აქტივირებულ არხებთან. 1002 (2v2) შეფასების კონფიგურირებისას, ზედა
 არხი ყოველთვის დეაქტივირდება.

მაგალითი (იხ. სურათი 5-4:).

F-PLC Datatype			F-I/O	0 1	2 3 4 5 6 2
F-DI8x24VDCHF			Raci		
reame	Data type	Default value			
1 C F_Input_Ch_0	Bool	false			
2 d F_Input_Ch_1	Bool	false			
3 💶 F_Input_Ch_2	Bool	false			<u></u>
4 💶 F_Input_Ch_3	Bool	false		to and a second	11
5 😋 F_Input_Ch_4	Bool	false		0.0	0.0
6 💶 F_Input_Ch_5	Bool	false		0.0	0.0
7 💶 F_Input_Ch_6	Bool	false		1	88
8 😋 F_Input_Ch_7	Bool	false		88	88
9 -CI F_InputCh_0_VS	Bool	false		88	88
10 - F_InputCh_1_VS	Bool	falze		00	00
11 C F_InputCh_2_VS	Bool	fals:e			
12 💶 F_InputCh_3_VS	Bool	false	<		1
13 💶 F_InputCh_4_VS	Bool	false	F-DI 8x24VDC HF	1 [F-DI8x24VDC]	
14 💶 F_InputCh_5_VS	Bool	false			
15 - F_InputCh_6_VS	Bool	false	General IO	tags System cons	tants Texts
16 - F_InputCh_7_VS	Bool	false	Name	Type A	ddress Tag table
12 gidt w			Eliput_1	"F-DI8x24VDCHF" D	111.0 Default tag table
	1		F_input_1	"F-DI8x24VDCHF" D	111.1 Default tag table
PLC Gambo	00		F_input_1	"F-DI8x24VDCHF" D	111.2 Default tag table
002-			F_input_1	"F-DI8x24VDCHF" D	111.3 Default tag table
PLC tags	1		F_input_1	"F-DI8x24VDCHF" D	111.4 Default tag table
Name Tao	table Data type	Address	F_input_1	"F-DI8x24VDCHF" D	111.5 Default tag table
1 E F_input_1 Def	ault tag table *F-DIBx24VDCHP	* %111.0	* F_input_1	"F-DI8x24VDCHF" D	111.6 Default tag table
2 «Add new»		<u>u</u>	F_input_1	"F-DI8x24VDCHF" D	111.7 Default tag table
and the second second second			F_input_1	"F-DI8x24VDCHF" D	112.0 Default tag table
			F_input_1	"F-DI8x24VDCHF" D	112.1 Default tag table
			F_input_1	"F-DI8x24VDCHF" D	112.2 Default tag table
			F_input_1	"F-DI8x24VDCHF" D	112.3 Default tag table
			F_input_1	"F-DIBx24VDCHF" D	112.4 Default tag table
			F_input_1	"F-DIBx24VDCHF" D	112.5 Default tag table
			F_input_1	"F-DIB:24VDCHF" D	112.6 Default tag table
			di * E leout 1	"F-DIRv24VDCHF" D	112.7 Default tag table

სურათი 5-4: შესავალ/გამოსავალ არეებთან ხელწვდომა F-PLC data types-ის დახმარებით

5.11 TRUE/FALSE

თუ თქვენთვის უსაფრთხოების პროგრამაში აუცილებელია სიგნალები "TRUE" და "FALSE", მაშინ ამისთვის არსებობს ორი სხვადასხვა ხერხი:

- ბლოკების ფაქტიური პარამეტრების სახით;
- ოპერანდების სახით ინსტრუქციებში.

ბლოკის ფაქტიური პარამეტრი

S7-1200/S7-1500F კონტროლერებისათვის თქვენ შეგიძლიათ გამოიყენოთ ბიტური კონსტანტები (bool) "FALSE" 0-სათვის და "TRUE" 1-სათვის, ფაქტიური პარამეტრების სახით ფორმალური პარამეტრების დანიშნისათვის, უსაფრთხოების პროგრამაში ბლოკის მიერ გამოძახების პროცესისას შემდგომი გამოყენებისათვის. ფორმალურ პარამეტრში ჩაიწერება მხოლოდე საკვანძო სიტყვა "FALSE" ან "TRUE" (იხ.

სურათი 5-6:).



სურათი 5-5: სიგნალები "FALSE" ან "TRUE" ფაქტიური პარამეტრების სახით

გამოიყენებები ინსტრუქციებში

თუ თქვენთვის აუცილებელია დანიშნოთ სიგნალები "FALSE" ან "TRUE" ინსტრუქციებისთვის, მაშინ თქვენ უნდა აფორმიროთ იგი, ისე როგორც ნაჩვენებია საურათზე ქვემოთ:

- გადართეთ დაპროგრამირების ენა FBD-ზე;
- შექმენით bool ტიპის ორი სტატისტიკური ანდა დროებითი ცვლადი: "alwaysTrue", "alwaysFalse";
- შექმენით სეგმენტი 1, როგორც სურათზეა ნაჩვენები ქვევით;
- ცვლადები შეიძლება გამოყენებულ იქნას მთელი ბლოკისათვის როგორც "True" და "False" (იხილეთ სურათი 5-6:).



სურათი 5-6: სიგნალები "True" და "False"

5.12 ოპტიმიზირებული კომპილიაცია და პროგრამის შესრულების რეჟიმი

ამ თავში თქვენ გაიგებთ იმას, თუ როგორ შეამციროთ კომპილიაციისა და სამომხმარებლო პროგრამის მუშაობის დრო.

დანართისგან დამოკიდებულებით ყოველთვის არ არის შესაძლებელი რომ გამოიყენოთ ზოგიერთი რეკომენდაციები. მიუხედავად ამისა ახსნადია თუ რატომ გვაძლევს ზოგიერთი მეთოდი პროგრამირებაში შესაძლებლობას შევამციროთ კომპილიაციისა და პროგრამის მუშაობის დრო.

სამომხმარებლო ბლოკების გარდა, არსებობს აგრეთვე ავტომატურად შექმნილი შიდა სისტემური ბლოკები. ასეთი ბლოკები გარანტიას იძლევა უსაფრთხოების პროგრამების საიმედოდ მუშაობაზე. შიგა სისტემური ბლოკები ზრდიან პროგრამის კომპილიაციისა და მუშაობის დროს (იხ. ცურათი 5-7:).

მაგალითი



სურათი 5-7: მომხმარებლისა და სისტემის მიერ შექმნილი F-ბლოკები

5.12.1 შესრულების დროზე გავლენის მომხდენ: TP,TON, TOF ბლოკებზე უარის თქმა

თვითოეული ასეთი ბლოკი (TP,TON, TOF) ითხოვს ბლოკებსა და გლობალურ მონაცემებში დამატებით კორექტირებას უსაფრთხოების პროგრამის პროგრამულ კოდში.

რეკომენდაციები

შეეცადეთ შეძლებისდამაგვარად იშვიათად გამოიყენოთ ეს ბლოკები.

5.12.2 ჩასმულ გამოძახებებზე უარის თქმა

ჩასმული გამოძახებები ზრდიან სისტემის მიერ გენერირებული F ბლოკების მოცულობას, რადგანაც წარმოიშობა უსაფრთხოების ფუნქციის პროცედურის შემოწმების აუცილებლობა. თუ კი ჩასმების სიღრმემ გადააჭარბა რვას მაშინ თქვენ დაინახავთ გაფრთხილებას TIA Portal-ში კომპილიაციის ეტაპის დროს.

რეკომენდაციები

ჩამოაყალიბეთ თქვენი პროგრამის სტრუქტურა ისეთნაირად, რომ თქვენთვის საჭირო არ გახდეს ჩასმული გამოძახებების გამოიყენება.

5.12.3 გამოყოფა სტანდარტული პროგრამისა უსაფრთხოების პროგრამისაგან

რთულ პროექტებში ხშირად საჭირო ხდება მონაცემთა გაცვლების საჭიროება სტანდარტულ პროგრამასა და უსაფრთხოების პროგრამას შორის. თუ კი გადაცვლა რეალიზებულია ცვლადების დახმარებით (მაგალითად, მერკერებით), მაშინ სტანდარტულ პროგრამაში ცვლილებებმა შეიძლება მოითხოვოს უსაფრთხოვების პროგრამის კომპილიაცია. ცვლილებების ჩათვირთვისათვის აუცილებლია გადავიყვანოთ CPU რეჯიმში STOP-ში.

რეკომენდაციები

გამოიყენეთ სტანდარტული DB (იხ. თავი 5.13 მონაცემთა გადაცვლა სტანდარტულ და F – პროგრამას შორის). ცვლილებები სტანდარტულ პროგრამაში არ შეეხება უსაფრთხოების პროგრამას. სტანდარტული პროგრამის ჩატვირთვისათვის არ გახდება საჭირო კონტროლერის გადაყვანისა STOP რეჯიმში.

5.12.4 მულტიეკზემპლიარების გამოყენება

ერთი ეკზემპლიარული DB-ს გამოძახებისას უსაფრთხოების რამოდენიმე ადგილობრივ პროგრამაში, საჭირო ხდება ასეთი DB უნდა დამუშავებულ იყოს ერთზე მეტჯერ ერთი ციკლის განმავლობაში. ასეთი დამუშავება ითხოვს შიგა სისტემური F ბლოკების დიდ რაოდენობას.

რეკომენდაციები

შეძლებისდაგვარად ყოველთვის გამოიყენეთ მულტიეკზემპლიარები. ამით თქვენ შეამცირებთ შიგა სისტემური F ბლოკების რაოდენობას.

5.12.5 უარის თქმა JMP/Label ინსტრუქციების გამოყენებაზე

თუ კი ბლოკის გამოძახება ხდება (JMP/LABEL) ნიშნულზე გადასვლის გავლით, ეს ითხოვს დამატებით დამუშავებას შიგა სისტემურ F ბლოკებში. ამ შემთხვევაში, გამოძახებულ ბლოკზე გადასვლისას, გაშვებულ უნდა იქნას კოდის კორექტირება. აღნიშნული ინსტრუქციები ამცირებენ მწარმოებლობას და ზრდიან პროგრამის კომპილიაციის დროს.

რეკომენდაციები

შეეცადეთ არ გამოიყენოთ JMP/Lable ინსტრუქციები, რათა შეამციროთ შიგა სისტემური F ბლოკების რაოდენობა.

5.13 მონაცემთა გაცვლა სტანდარტულ და F პროგრამებს შორის

ზოგიერთ შემთხვევაში საჭირო ხდება შეასრულოთ მონაცემთა გაცვლა სტანდარტულ პროგრამასა და უსაფრთხოების პროგრამას შორის. ამასთან, დაცული უნდა იყოს შემდეგი სახის რეკომენდაციები, რათა გარანტირებულ იყოს მონაცემთა კონსისტენტობა როგორც სტანდარტული, ასევე უსაფრთხოების პროგრამისათვის.



სურათი 5-8: მონაცემთა გაცვლა სტანდარტულ პროგრამასა და უსაფრთხოების პროგრამებს შორის

რეკომენდაციები

- არ მოახდინოთ მონაცემთა გაცვლა მერკერების გამოყენებით (იხ. თავი 4.2. გადასვლა მერკერილი მეხსიერების გამოყენებიდან მონაცემთა გლობალურ ბლოკებზე);
- თავი მოუყარეთ უსაფრთხოების პროგრამასა და სტანდარტულ პროგრამას შორის ხელწვდომას ორ სტანდარტულ DB-ში (იხ. სურათი 5-8:).

5.14 უსაფრთხოების პროგრამის ტესტირება

დამატებით, მუდმივად კონტროლირებად მონაცემებთან სტანდარტული სამომხმარებლო პროგრამაში, თქვენ შეგიძლიათ შეცვალოთ შემდეგი მონაცემები უსაფრთხოების პროგრამისა დაცვის დეაქტივირულ რეჯიმში:

- I/O-ს ასახვის არეში;
- F-DB (DB გამოკლებით კომუნიკაციური ჯგუფის F-runtime-ისათვის);
DB F-FB ეკზემპლიარები;

• F- I/O DB.

თვისებები

- F-I/O-სთან ხელწვდომა შესაძლებელია მხოლოდ, როცა F-CPU იმყოფება RUN რეჯიმში;
- დაკვირვების ცხრილში, თქვენ შეგიძლიათ თვალყური ადევნოთ მაქსიმუმ 5 შესასვლელ/გამოსასვლელს უსაფრთხოების პროგრამაში;
- თქვენ შეგიძლიათ გამოიყენოთ რამდენიმე დაკვირვების ცხრილი;
- ტრიგერი აუცილებელად უნდა დააყენოთ "permanent" (მუდმი) ან "once" (ერთჯერადი) "cycle start"-ისთვის (ციკლის დასაწყისი) ანდა "cycle end"ისთვის (ციკლის დასასრული);
- ფორსირება ხელმიუწვდომელია F-I/O-სათვის;
- წყვეტების დაყენება სტანდარტულ პროგრამაში იწვევს თავის მხრივ
 შეცდომების წარმოშობას უსფრთხოების პროგრამაში:
 - F ციკლის საკონტროლო დროის გადამეტება;
 - შეცდომა კომუნიკაციის პროცესში F-I/O-სთან;
 - შეცდომა კომუნიკაციის პროცესში F-CPU-CPU-სთან;
 - CPU-ს შიგა შეცდომები.
- თუ კი თქვენ მაინც დაგჭირდებათ გამოიყენოთ წყვეტის წერტილები ტესტირებისათვის, მაშინ თქვენ უნდა დეაქტივირება გაუკეთოთ დაცვის რეჯიმს, რაც გამოიწვევს შემდეგ შეცდომებს:
 - შეცდომები კომუნიკაციის პროცესში F-I/O-სთან;
 - შეცდომები კომუნიკაციის პროცესში F-CPU-CPU-სთან;

5.15 STOP რეჟიმი F შეცდომების წარმოშობის დროს

F-CPU გადადის STOP რეჟიმში შემდეგ შემთხვევებში:

- საქაღალდეში "System blocks" (სისტემური ბლოკები), თქვენ არ უნდა დაამატოთ, შეცვალოთ ანდა მოაშოროთ არანაირი ბლოკი;
- თუ კი ინსტრუქციის შესრულების რეზულტატი გამოდის მონაცემთა ტიპის დასაშვები ზღვარიდან (გადავსება), მაშინ დიაგნოსტიკური მოვლენის მიზეზი ჩაიწერება F CPU-ს დიაგნოსტიკურ ბუფერში;
- პროგრამაში არ უნდა არსებობდეს მიმართვა DB F-FB-ს ნებისმიერ ეკზემპლიართან, რომლებიც არ გამოიძახება უსაფრთხოების პროგრამაში;
- თუ კი "Maximum cycle time of the F run-time group" (F run-time ჯგუფი)-ს ციკლის მაქსიმალური დრო იქნა გადაჭარბებული, მაშინ F-CPU გადავა STOP რეჟიმში. მიეცით დასაშვები მაქსიმალური დრო "Maximum cycle time der F run-Time group" (ციკლის მაქსიმალური დრო F run-time ჯგუფისთვის) (მაქსიმალური 20,000 მწმ);

- თუ კი F run-time ჯგუფი, საიდანაც უნდა იქნას წაკითხულ DB-ს ტეგები F-runtime ჯგუფისთვის (მთავარი ბლოკი F run-time ჯგუფის), მაშინ F-CPU გადავა STOP რეჯიმში;
- სტარტული მნიშვნელობების ცვლილება ეკზემპლიარულ DB F-FB აკრძალულია online და offline, რადგან შეიძლება გამოიწვიოს F-CPU გადასვლა STOP რეჯიმში;
- უსაფრთხოების პროგრამის მირითადი ბლოკი არ უნდა შეიცავდეს გამოუცხადებელ პარამეტრებს;
- F-FC ს გამოსასვლელები უნდა იყვნენ ყოველთვის ინციალიზირებული.

5.16 უსაფრთხოების პროგრამის მიგრაცია

ინფორმაცია უსაფრთხოების პროგრამის მიგრაციის შესახებ ხელმისაწვდომია ბმულით:

https://support.industry.siemens.com/cs/ww/en/view/109475826

5.17. ძირითადი რეკომენდაციები უსაფრთხოების მიმართებაში

STEP 7 Safety და F მოდულებთან მუშაობის დროს აუცილებლად უნდა მიდევდეთ შემდეგ რეკომენდაციებს:

- შეძლებისდამაგვარად, ყოველთვის გამოიყენეთ F კონტროლერები. შემდგომში უსაფრთხოების ფუნქციების გაფართოვება შეიძლება რეალიზებულ იქნას განსაკუთრებული შრომის გარეშე;
- ყოველთვის გამოიყენეთ ერთი პაროლი უსაფრთხოების პროგრამისათვის, რომ თავიდან იქნას აცილებული არაავტორიზებული ხელწვდომა. პაროლის მიცემა ხდება "Safety administracion" რედაქტორში (უსაფრთხოების მართვა).

5.18 ყველაზე უფრო მეტად მნიშვნელოვანი რეკომენდაციები

- ოპტიმიზირებული ბლოკების გამოყენება;
 თავი 2.6 ოპტიმიზირებული ბლოკები
- VARIANT მონაცემთა ტიპის გამოყენება ნაცვლად ANY ტიპისა;
 თავი 2.8.5 მონაცემთა ტიპი VARIANT
- პროგრამის სტრუქტურირება;
 თავი 3.2 მაორგანიზებელი ბლოკები (OB)
- ინსტრუქციის დამატება როგორც მულტიეკზემპლიარის (TON, TOF ..);
 თავი 3.2.5 მულტიეკზემპლიარები
- ბლოკების განმეორებითი გამოყენება;
 თავი 3.2.8 ბლოკების განმეორებით გამოყენების შესაძლებლობა
- სიმბოლური დაპროგრამება;
 თავი 3.6 სიმბოლური დამისამართება
- მონაცემებთან მუშაობის დროს გამოიყენეთ მასივები;
 თავი 3.6.2 ARRAY მონაცემთა ტიპი და ირიბი ხელწვდომა ელემენტებთან

- PLC data types-ის შექმნა;
 თავი 3.6.4 შეყვანა/გამოყვანის არეებთან ხელწვდომა PLC data types-ის დახმარებით
- ბიბლიოთეკების გამოყენება პროგრამების შენახვისათვის;
 თავი 3.7 ბიბლიოთეკები
- მერკერული მეხსიერებიდან გადასვლა მონაცემთა გლობალურ ბლოკებზე;
 თავი 4.2 მერკერული მეხსიერებიდან გადასვლა მონაცემთა

გლობალურ ბლოკებზე.

6 პრაქტიკული სამუშაოები

6.1. პრაქტიკული სამუშაო 1

6.1.1 S7-1200 პლკ-ს მომზადება სამუშაოდ

ქვემოთ განხილულია S7-1200 პროგრამირებადი ლოგიკური კონტროლერების პრაქტიკული გამოყენების საკითხები, კერძოთ აქ შევეხებით:

- პროგრამული უზრუნველყოფის მიმოხილვას;
- პროგრამირების ინსტრუქციების მიმოხილვას;
- ბიბლიოთეკის ბლოკების მიმოხილვას;
- კონტროლერის ჩართვასა და კონფიგურაციის ჩატვირთვას;
- პროგრამის შექმნას, კომპილიაციას, ჩატვირთვას და ტესტირებას.

6.1.2 TIA Portal-ის ხელმისაწვდომი ფუნქციები

ვნახოთ ახლა ის ფუნქციები, რომლებიც ჩევეთვის ხელმისაწვდომია პროგრამების დამუშავების დროს TIA Portal-ში ამ პროგრამის ბიბლიოთეკიდან. მთავარი (ბაზური) ფუნქციების სანახავად უნდა შევიდეთ Basic instruction განყოფილებაში (იხ. სურათი 6-1:) და მასზედ დაწკაპუნებით გამოვიძახოთ ბაზისური ფუნქციების ჩამონათვალი: (იხ. სურათი 6-2:).



სურათი 6-1: პროგრამული უზრუნველყოფის შემადგენელი ნაწილები; სურათი 6-2: ბაზისური ფუნქციების სია

აქ არის შემდეგი დასახელების საქაღალდეები:

1. Bit Logic Operation-ლოგიკა ბიტებზე, რომელიც შეიცავს ლოგიკურ ფუნქციებ: "და", "ან", "არა" და სხვადასხვა დანიშნულების ტრიგერებს (იხ. სურათი 6-3:).;

- 2. Timer Operation (ტაიმერები), სადაც შედის დროის სხვადასხვა მთვლელები (იხ. სურათი 6-4:);
- 3. Counter Operation სხვადასხვა სახის ხდომილებათა მთვლელები (იხ. სურათი 6-5:);
- 4. Comparator Oparation (შედარების ოპერაციები) (იხ. სურათი 6-6:);
- 5. Math fanctions (მათემატიკური ფუნქციები), აქ ხელმისაწვდომია როგორც ოთხი მირითადი არითმეტიკული ოპერაცია: შეკრება, გამოკლება, გამრავლება, გაყოფა, ასევე ტრიგონომეტრიული ფუნქციები, მაგ. სინუსი და კოსინუსი, ასევე არკსინუსი და არკკოსინუსი და სხვ. ფაქტიურად პროგრამაში წარმოდგენილია სრული მათემატიკა (იხ. სურათი 6-7:);
- 6. Move Operation (გადაადგილების ოპერაციები), რომლებიც ძირითადათ გამოიყენება ცვლადებთან მუშაობისათვის (იხ. სურათი 6-8:);
- Conversions operations (გარდაქმნის ოპერაციები), რომლებიც გამოიყენება გაზომვის შედეგების სხვადასხვა ერთეულებს შორის გარდაქმნებისათვის (იხ. სურათი 6-9:),;
- 8. Program control operations (პროგრამული მართვის ოპერაციები), რომლებიც ასრულებენ სხვადასხვა სახის გადასვლებს ქვეპროგრამებს შორის (იხ. სურათი 6-10),;
- 9. Word logic operations (ლოგიკური ოპერაციები სიტყვებზე), რომლებიც გამოიყენება ლოგიკური ოპერაციების შესრულებისათვის მთლიან სიტყვებზე (იხ. სურათი 6-11:);
- 10. Shift and rotate (ბიტებისა და ბაიტების წანაცვლებისა და ბრუნვის ოპერაციები) (იხ. სურათი 6-12:).

Basic instructions		Basic instructions	
Name	1	Name	Des
General	-	General	
➡ Hit logic operations	T	Bit logic operations	
 - -	1 40	Timer operations	
HI -1/1-	1	IEC Timers	
-ILONI-	1	TP	Ger
0 -()-		TON	Ger
-(/)-	- (-)	TOF	Ger
-(R)	1	TONR	Tim
-(S)	1	-(TP)-	Star
O SET_BF		() -(TON)-	Star
O RESET_BF	1	() -(TOF)-	Star
E SR		(TONR)-	Tim
E RS	1	-(RT)-	Res
-IPI-	:	-(PT)-	Loa
HI - N -		Fi Counter operations	

სურათი 6-3: ლოგიკა ბიტებზე; სურათი 6-4: ტაიმერები

Basic instructions		Basic instructions	
Name	Des	Name	Des
General		General	
Bit logic operations		Bit logic operations	
Timer operations		Timer operations	
▼ +1 Counter operations		Counter operations	
IEC Counters		 Comparator operations 	
= CTU	Cou	HI CMP ===	Equ
= СТО	Cou	HI CMP ↔	Not
= стир	Cou	HI CMP >=	Gre
Compare Count down		HI CMP <=	Les
Math functions		HI CMP >	Gre
Move operations		HI CMR<	Les
Conversion operations		HI IN RÂNGE	Valu
Program control operation	5	HI OUT_RANGE	Valu
Word logic operations		HI -OKI-	Che
🕨 🗮 Shift and rotate	-	HI -INOT_OKI-	Che

სურათი 6-5: სხვადასხვა ხდომილებათა მთვლელები; სურათი 6-6: კომპარატორები

 Basic instructions 		Basic instructions			
Name	[Name	Des		
Bit logic operations	~	General			
Timer operations		Bit logic operations			
Counter operations	6	G Timer operations			
Comparator operations	4	Counter operations			
	T	Comparator operations			
E CALCULATE	- (Math functions			
ADD ADD	4	✓ → Move operations			
E SUB	1	MOVE	Mov		
E MUL	1	FieldRead	Rea		
E DIV	1	FieldWrite	Whit		
COM E	1	MOVE_BLK	Mov		
E NEG		UMOVE_BLK	Mov		
INC INC	1	FILL_BLK	Fill		
E DEC	1	UFILL_BLK	Fill		
ABS	1	E SWAP	Swa		

სურათი 6-7: მათემატიკური ფუნქციები; სურათი 6-8: გადატანის ოპერაციები

Basic instructions		Basic instructions	
Name	Des	Name	Des
General		General	
Bit logic operations		Bit logic operations	
Timer operations		Timer operations	
Counter operations		Fil Counter operations	
Comparator operations		Comparator operations	
Math functions		Math functions	
Move operations		Move operations	
Conversion operations		Conversion operations	
CONVERT	Con	Program control operations	
E ROUND	Rou	(JMP)	Jum
E CEIL	Ger	(JMPN)	Jum
E FLOOR	Ger	LABEL	Jum
TRUNC	Trur	IMP_LIST	Defi
E SCALE_X	Sca	E SWITCH	Jum
NORM_X	Nor	-(RET)	Ret

სურათი 6-9: გარდაქმნის ოპერაციები, პროგრამული კონტროლის ოპერაციები



სურათი 6-11: ლოგიკური ოპერაციები სიტყვებზე; სურათი 6-12: ბიტებისა და ბაიტების წანაცვლებისა და ბრუნვის ოპერაციები კონტროლერებზე მუშაობის დაწყებისათვის უპირველეს ყოვლისა ჩვენ უნდა მივწოდოთ პლკ-ზე კვება. ჩვენს ხელთ არსებული მოდელის შემთხვევაში საჭიროა 220 ვოლტის მიწოდება, რის შედეგადაც მიკროკონტროლერი დადგება "Stop"-ის რეჯიმში, ანუ ანთებული იქნება მწვანე შუქდიოდი, პროგრამა კი ჯერ კიდევ არ არის დამუშავებული და მით უმეტეს ჩატვირთული. სამუშაოს დაწყებისათვის ჩვენ უნდა განვახორციელოთ კონტროლერის Ethernet შეერთება კომპიუტერთან, ამისათვის საჭიროა გამოვიყენოთ სპეციალური Ethernet კაბელი, რომლის ერთი ბოლო უნდა მივაერთოთ ჩვენი პროგრამირებადი ლოგიკური კონტროლერის შესაბამის გასართს, ხოლო მეორე ბოლო კი ჩვენი პერსონალური კომპიუტერის შესაბამის გასართს.

6.1.3 ქსელური შეერთების გამართვა

ამის შემდეგ საჭირო იქნება, რომ პერსონალურ კომპიუტერზე ჩავრთოთ ქსელური შეერთება, რისთვისაც უნდა მოვახდინოთ შემდეგი სახის აწყობები: დავაწკაპუნოთ ეკრანის ქვედა მარჯვენა კუთხეში არსებულ ნიშანზე SE Connected და გამონათებულ ფანჯარაში დავაწკაპუნოთ Change adapter options (იხ. სურათი 6-13:), მიღებულ ფანჯარაში (იხ. სურათი 6-14:), დავაწკაპუნოთ Lokal Area Connection და შევიდეთ კონტროლერის ქსელური შეერთების ფანჯარაში (იხ. სურათი 6-15:). Windows 10 ოპერაციული სისტემისათვის იგივე პროცედურა სრულდება შემდეგნაირად, შევდივართ ეკრანის მარჯვენა ქვედა კუთხეში არსებულ ნიშანზე

Unidentified network , შემდეგ ვაწკაპუნებთ ნიშანზე ' Unidentified network, გამოსულ ფანჯარაში ვაწკაპუნებთ Change adapter options, შედეგად გამოსულ ფანჯარაში ვაწკაპუნებთ ნიშანზე Ethernet Unidentified network და შევალთ კონტროლერის ქსელური შეერთების იმავე (იხ. სურათი 6-15:) ფანჯარაში.



სურათი 6-13: კონტროლერის ქსელური შეერთება მე-1 ეტაპი



სურათი 6-14: კონტროლერის ქსელური შეერთება მე-2 ეტაპი

Local Area Con	nection Status	
General		
Connection		
IPv4 Connectivi	ty:	Internet
IPv6 Connectivi	ty:	No network access
Media State:		Enabled
Duration:		00:04:42
Speed:		100.0 Mbps
Details Activity		
	Sent —	Received
Bytes:	161 559	550 626
Properties	Disable	Diagnose

სურათი 6-15: კონტროლერის ქსელური შეერთება მე-3 ეტაპი

ამ ბოლო ფანჯარაში ვაწკაპუნებთ ღილაკზე Properties, ახლად გამოსულ ფანჯარაში კი მოვნიშნავთ Internet Protokol Version 4 (TCP/IPv4) პროტოკოლს (იხ. სურათი 6-16:). შემდეგ, აქაც დავაჭერთ ღილაკს Properties, რის შედეგადაც გამოსულ ფანჯარაში (იხ. სურათი 6-17:), ვწერთ ჩვენი ქსელის IP მისამართს (იხ. სურათი 6-18:).

Local Area Connection Properties
Networking Sharing
Connect using:
Intel(R) 82579LM Gigabit Network Connection
Configure
This connection uses the following items:
File and Printer Sharing for Microsoft Networks QoS Packet Scheduler
✓ Internet Protocol Version 4 (TCP/IPv4)
Microsoft Network Adapter Multiplexor Protocol
Microsoft LLDP Protocol Driver
Internet Protocol Version 6 (TCP/IPv6)
✓ Link-Layer Topology Discovery Responder
< >
Install Uninstall Properties
Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
OK Cancel

სურათი 6-16: კონტროლერის ქსელური შეერთება მე-4 ეტაპი

Internet Protocol Version 4 (TCP/IP	v4) Prope	erties			\times
General					
You can get IP settings assigned au this capability. Otherwise, you need for the appropriate IP settings. Obtain an IP address automat	utomaticall d to ask yo ically	y if y our n	our n etwor	etwork sup k administ	ports rator
IP address:					
Subnet mask:					
Default gateway:					

სურათი 6-17: კონტროლერის ქსელური შეერთება მე-5 ეტაპი

Internet Protocol Version 4 (TCP/IPv4)	Properties	Х
General		
You can get IP settings assigned autom this capability. Otherwise, you need to for the appropriate IP settings.	natically if your network supports ask your network administrator	
Obtain an IP address automatical	у	
• Use the following IP address:		
IP address:	192 . 168 . 0 . 250	
Subnet mask:	255 . 255 . 255 . 0	
Default gateway:		
Obtain DNS server address autom	natically	
Use the following DNS server addresses	resses:	
Preferred DNS server:		
Alternate DNS server:		
Validate settings upon exit	Advanced	
	OK Cance	I

სურათი 6-18: კონტროლერის ქსელური შეერთება მე-6 ეტაპი

აქ ჩვენ შეგვიძლია ჩავწეროთ აბსოლუტურად ნებისმიერი მისამართი ჩვენი ქსელისათვის, მაგრამ შევეცადოთ დავიცვათ სტანდარტი. ამისათვის, ამ მისამართის პირველი რიცხვებად ავიღოთ 192 და 168, ამის შემდეგ კი, თუ დავუშვებთ, რომ ქსელი იყოს ნულოვანი, კომპიუტერის ნომერი კი მაგალითად 250, მაშინ მივიღებთ 192 168 0 250 (იხ. სურათი 6-18:).

შემდეგ გადავდივართ მასკას არჩევაზე. მასკად შეგვიძლია ავირჩიოთ 255, 255, 255, 0, რაც უფლებას მოგვცემს დავამისამართოთ 255-მდე მოწყობილობა. შემდეგ ვაჭერთ ღილაკს OK და შემდეგ კი აუცილებლად ღილაკს Close (იხ. სურათი 6-19:), რის შედეგადაც პლკ-ს ქსელური შეერთება კომპიუტერთან დასრულებული იქნება.

🏺 Local Area Conr	nection Status			×
General				
Connection				-
IPv4 Connectivit	ty:		Internet	
IPv6 Connectivit	ty:	No netw	ork access	
Media State:			Enabled	
Duration:			00:19:33	
Speed:		1	100.0 Mbps	
Details				
Activity				
	Sent —	-	Received	
Bytes:	186 396	ľ	633 564	
Properties	Disable	Diagnose		
			Close	

სურათი 6-19: კონტროლერის ქსელური შეერთება მე-7 ეტაპი

6.1.4 TIA Portal-ის ჩამოტვირთვა

პროგრამის დაწერისა და მისი პლკ-ში ჩატვირთვისათვის საჭიროა პლკ-ში დავაყენოთ პროგრამული უზრუნველყოფა TIA Portal v13, ან TIA Portal v14. ეს არის ფირმა სიმენსის პროგრამების დამუშავების გარემო TIA Portal-ი. აქ ჩვენ ვაჩვენებთ მთელს პროცესს დაწყებული TIA Portal-ის ჩამოტვირთვით დამთავრებული მისი დაყენებით კომპიუტერზე. ჩვენ გამოვიყენებთ უფასო ვერსიას გათვლილს 21 დღეზე. დავიმახსოვროთ ეს ინტერნეტ მისამართი <u>https://support.industry.siemens.com</u> თუ ჩვენ ვიმუშავებთ ფირმა სიმენსის პროდუქციასთან. იგი ჩვენ მალიან დაგვეხმარება მომავალში (იხ. სურათი 6-20:).

დასაწყისში ჩვენ დაგვჭირდება რეგისტრაცია ამ საიტზე. ამისათვის აქ არის რეგისტრაციის ღილაკი Register, თქვენ უნდა დავაჭიროთ თაგვის ღილაკით ამ ღილაკს და დაიწყოთ რეგისტრაციის პროცესი. რეგისტრაციის გარეშე თქვენ ვერ ჩამოთვირთავთ TIA Portal-ს. რეგისტრაციის ღილაკზე დაჭერის შემდეგ ჩვენ მოვხვდებით ფირმა Siemens-ის რეგისტრაციის გვერდზე (იხ. სურათი 6-21:), რომლის ველებიც უნდა შევავსოთ უშეცდომოდ, შემდეგ ბოლოში ველში Please enter trhe displayed code და დავაჭიროთ ღილაკს Continue.

ამის შემდეგ მოგვიწევს რამდენიმე დღის ლოდინი ვიდრე სიმენსი დაამუშავებს თქვენს განაცხადს და შემდეგ მოგივათ შეტყობინება თქვენს ელექტრულ მისამართზე.



სწურათი 6-20: ფირმა სიმენსის მხარდაჭერის support.industry საიტი

აქ სიმენსმა შეიძლება მოგთხოვოთ ზოგიერთი ჩანაწერის კორექტირება, ასეთ შემთხვევაში დაგჭირდებათ კიდევ ლოდინი რამდენმე დღე და ბოლოს სიმენსი მოგცევთ ხელწვდომას თქვენთვის საჭირო ფაილების გადმოტვირთვაზე. თუ კი მოხდა ისე, რომ ჩვენ დავრეგისტრირდით, მაგრამ საიტი არ გიშვებთ, მაშინ უნდა გამოიყენოთ სხვა ბრაუზერი.

ამის შემდეგ უკვე უნდა გადავიდეთ ჩვენთვის საჭირო ფაილების გადმოტვირთვაზე, ამისათვის შევდივართ იგივე მისამართზე <u>https://support.industry.siemens.com</u> და ჩვენთვის საჭირო პროდუქტის სახელი, მაგალითად TIA Portal v14 ჩავწეროთ ველში Search in Online Support (იხ. სურათი 6-21:). აქ სასურველია არ ჩავწეროთ ყველაზე ბოლო ვერსია, რადგან ამ ვერსიაში შეიძლება აღმოჩნდეს სხვადასხვა სახის შეცდომები და მოულოდნელობები.



სურათი 6-21: ჩვენთვის საჭირო პროგრამული პროდუქტის არჩევა

ამის შესაბამისად, ჩამოშლიდი სიიდან ვირჩევთ იგივე tia portal v14, შედეგად სისტემა გადადის ძებნის პროცესში და შემდეგ ეკრანზე აისახება ძებნის შედეგები (იხ. სურათი 6-22:). აქ უნდა დავაჭიროთ ღილაკს ALL და ჩამოშლილ ფანჯარაში მოვნიშნოთ მხოლოდ Download (იხ. სურათი 6-23:), რითაც ჩვენ ავირჩევთ ძებნის არეს.



სურათი 6-22: ჩვენთვის საჭირო პროდუქტის ძებნის შედეგის ასახვა

Программирование ПЛК. 7. Уст 🗙 Search - tia	portal v14 - Product5 × +			- (Ð
→ C ^a support.industry.siemens.com/cs	/search?search=tia%20portal%20v14&ty	/pe=ProductSupport%2CExampleOf	Use%2CCatalog%2CCertificate%2CCharacteristic%2CDownload%2CFaq%2CManual 🛠 📲	💁 🛸 🗐 🚺 Pau	used
SIEMENS			● Internet > Register > Log in		
Industry Online Support International Langua 	ge ► Contact ► Help	 Support Request 	▶ Site Explorer		
Home > Search			🗗 Product Support 🍽 Services 🎄 Forum 🖻 mySupport		
Search term					
tia portal v14	Q ×		Clear all filters		
Product Support Application example Catalog/Brochure					
Caracteristic Characteristic Download FAQ Product note Software archive	IC STEP 7 Professional V14 SP1 IC WINCC Comfort V14 SP1 ICS STARTDRIVE V14 SP1 DOWNLOAD IC WINCC Professional Trial V14 SP1 IC WINCC Advanced V14 SP1	more *	Results in areas Product Support > STEP 7 Professional (TIA Portal) (648) > STEP 7 Basic (TIA Portal) (610) > SIMATC WINCC (TIA Portal Engineering (389) > Failsafe CPUs (279) > Stander-CPUs (279)		
5 Services	age: 20 50 100	« < 1 2 3 4 5 > »	STEP 7 (TIA Portal) (1852)		
OK		Relevance descending 💌	 SIMATIC WinCC (TIA Portal) (1196) Communication - Networks (289) DIMATIC Device (400) 		
> Download SIMATIC STEP 7 (TIA Portal) As a registered customer you can downlo Brotectional M44 and test 3 for 31 days	V14 TRIAL Download ad the Trial software for SIMATIC STEP 7 Bas	09/30/2016 ic and ID: 109740158	> SINANCS (70) Additional areas Services		

სურათი 6-23: Download-ის მონიშვნა

შემდეგ ვაჭერთ ღილაკს OK და ეკრანი მიიღებს სახეს (იხ. სურათი 6-24:) და შევდივართ ჩვენთვის სხვადასხვა ვერსიის პროგრამული უზრუნველყოფების ჩამონათვალში.

Industry Online Support International	Language	▶ Contact ▶ Help ▶	Support Request	▶ Site Explorer
Home > Search				🗗 Product Support 🌁 Services 🎄 Forum 🖻 mySup
Search term tia portal v14		Q ×		Clear all filter
Search area Download	▼ X			
roduct links			more 🔻	
6ES7822-1AA04-0YA5 6AV2101-0AA04-0AA5 6SL3072-4EA02-0XG0 6AV2103-0AA04-0AA7 6AV2102-0AA04-0AA5	SIMATIC STEP 7 Professi SIMATIC WinCC Comfort SINAMICS STARTDRIVE SIMATIC WinCC Professio SIMATIC WinCC Advance	onal V14 SP1 V14 SP1 V14 SP1 DOWNLOAD Inal Trial V14 SP1 d V14 SP1		Results in areas Product Support > STEP 7 Professional (TIA Portal) (17) > SIMATIC WinCC (TIA Portal) Engineering (16) > Standard CPUs (15) > SIPLUS Standard CPUs (12) > SIPLUS Standard CPUs (11)
78 Entries Filtered by 'tia portal v14' 'Download' Ent	 ries per page: 20 50 100	«	< 1 2 3 4 5 > ≫ Relevance descending ▼	Additional areas Services > Further Information > TIA Portal Cloud > Service Platforms > Industry Premium Portal
> Download SIMATIC STEP 7 (TL As a registered customer you ca Professional V14 and test it for 2 are described in the delivery rele can also be ordered on DVD S	A Portal) V14 TRIAL Downlow n download the Trial software 1 days New features and cl ase for STEP 7 V14 The Tri- STEP 7 Professional V14	ad for SIMATIC STEP 7 Basic and hanges compared to earlier ver al version of SIMATIC STEP 7	09/30/2016 i ID: 109740158 sions ★★★☆☆(668) V14	Product Conferences > Communication - Networks Product Conferences > Decentral Peripherie Product Conferences > Drive Tools Product Conferences > Motion Control System SIMOTIOI Product Conferences > SINAMICS
For products:6ES7822-0AA04-01	YA7, 6ES7822-1AA04-0YA7			
Download Support Packages for Support Packages for TIA Portal V14includes only the current in them "Last change" The "TIA_I relevant Support Packages for Th includes the readme files for the permit you in the hardware cal procedure for updating the hardw Portal V16'.	or the hardware catalog in the V14 includes all available S lew Support Package files for Portal V14 The "TIA_Pon Support Packages for TIA Pon Support Packages for TIA Ponta lalog of your TIA Porta install vare catalog of your TIA Porta	he TIA Portal (HSP) support Package files for TIA P TIA Portal V14 which are lister wholad includes a list of the i tal_V14_Readmes.zip" down tal V14 The Support Packag aton 54163658 includes the I installation with this download	01/27/2021 ID: 72341852 3 at ★★☆☆☆(200) n the coad les will J TIA	Topic pages for these entries > TIA Portal - An Overview of the Most Important Documents and Links - Industrial Controls > PC-based automation: An overview of the most important documents and links (ASIS - After Sales Information system, Service & Support) > SIMATIC Technology - Motion Control: Overview and Important
For products:6ES7138-6DB00-0	BB1 All products of following attachments			Links > TIA Portal - An Overview of the Most Important Documents and Links - Controller > DIMATIO Foregraphic is the Optime Overset



აქ კი ვირჩევთ ტრიალ ვერსიას <Download SIMATIC STEP 7 (TIA Portal) V14 TRIAL Download და ეკრანზე გამოვა ჩვენთვის საჭირო ფაილების ჩამონათვალი (იხ. სურათი 6-25:). ჩვენ უნდა გადმოვტვირთოთ ჩვენი TIA Portal-ის ხუთივე ფაილი, შევქმნათ დესკტოპზე რაიმე დასახელების საქაღალდე და მოვათავსოთ ეს ფაილები ამ საქაღალდეში (იხ. 6-26:).

▶ Industry Online Support International → Language	▶ Contact ▶ Help ▶ Support Request	▶ Site Explorer	Search in Online Support	Q
> Home > Product Support		Product Sup	oport 🌶 Services 🏟 Forum 🗎	mySupport
Entry type: Download Entry ID: 109740158. Entry date: 09/30/2016				
	★★☆☆☆(669 > Rate)	1 5	s. ⊕
SIMATIC STEP 7 (TIA Portal) V14 TRIAL Downlo	ad	Share this Page	e: 🖂 🔚 🕨	
Entry Associated product(s)		mySupport	Cockpit	
SIMATIC STEP 7 (TIA Portal) V14 TRIAL Download				
		> Add to mySup > Add to mySup	port favorites	
As a registered customer you can download the Trial software for SIMATIC STEP 7 changes compared to earlier versions are described in the delivery release for STEI	Basic and Professional V14 and test it for 21 days. New features and P 7 V14; ≥ 109740340	> Favorites	port documentation	
Download		> My requests		
The software is subject to export restrictions; the download is only available to regis registration for export-restricted downloads can take several days. The Trial version of SIMATIC STEP 7 V14 can also be ordered on DVD:	lered users. Please bear in mind that because of strong demand, the	 CAx download My Products / 	ts Clipboard	
 Basic: D6ES7822-0AA04-0YA7 		Product info	ormation	
Professional: D6ES7822-1AA04-0YA7		Presales info		
		 Catalog and o 	rdering system online	
The download is divided into several files. Please download all the files into the sam combined and you can execute the setup.	te directory and execute the file ending with .exe. The files will now be	Technical info		
		Support		
STEP 7 Professional V14		 Service offer 		
CIMATIC STEP 7 Professional V14 001 (2.0 OP)		 Training 		
		 Contact & part 	tners	
LI o" SIMATIC_STEP_7_Professional_V14.002 (2,0 GB)				
of SIMATIC_STEP_7_Professional_V14.003 (2,0 GB)		Related link	(S	
D σ [*] SIMATIC_STEP_7_Professional_V14.004 (1019,8 MB)		> Compatibility T	iool	
SIMATIC_STEP_7_Professional_V14.exe (2,8 MB)		> Register your S > Visit our Conve	SINAMICS Drive arting Community	
SHA-256 checksum: D of SIMATIC_STEP_7_Professional_V14.bt (1 KB) 🔊 Info	ormation about SHA-256	> Get help with the set of the	he Digital Assistant e chatbot	
STEP 7 Basic V14:				

სურათი 6-25: ჩვენთვის საჭირო ფაილების სია

	Name	Date modified	Туре	Size
📌 Quick access		2 (0 (2021 7 45 PM		665 600 KD
Deskton	SIMATIC_S7_PLCSIM_V14.001	3/9/2021 7:46 PIVI	WinKAK archive	665,600 KB
	SIMATIC_S7_PLCSIM_V14.002	3/9/2021 7:44 PM	002 File	525,037 KB
🕂 Downloads 刘	SIMATIC_S7_PLCSIM_V14.exe	3/9/2021 7:50 PM	Application	2,719 KB
🖆 Documents 刘	SIMATIC_STEP_7_Professional_V14.001	3/8/2021 8:25 AM	WinRAR archive	2,097,152 KB
Fictures 🖉	SIMATIC_STEP_7_Professional_V14.002	3/8/2021 8:49 AM	002 File	2,097,152 KB
Docs	SIMATIC_STEP_7_Professional_V14.003	3/8/2021 9:04 AM	Adobe Acrobat D	2,097,152 KB
E_HDD (E:)	SIMATIC_STEP_7_Professional_V14.004	3/8/2021 9:23 AM	004 File	1,044,228 KB
Siemens V14 უთვა	မ SIMATIC_STEP_7_Professional_V14.exe	3/8/2021 9:27 AM	Application	2,832 KB
სახელმძღვანელ				

სურათი 6-26: სიმენსის საიტიდან გადმოტვირთული TIA Portal v14-ის საინსტალაციო ფაილები

აქ უნდა აღინიშნოს, რომ ჩვენ ავირჩიეთ TIA Portal v14-ის პროფესიონალური ვერსია SIMATIC_STEP_7_Professional, მაგრამ ასეთივე წარმატებით შეგვეძლო აგვერჩია SIMATIC_STEP_7_Basic ვერესია და აგრეთვე SIMATIC_STEP_PLCSIM სიმულიატორის ვერსია იმ შემთხვევაში თუ ჩვენ არ გვექნებოდა კონტროლერი ხელთ.

6.1.5 TIA PORTAL-ის ინსტალაცია

ახლა უნდა გადავიდეთ ჩვენს მიერ გადმოტვირთული TIA Portal v14-ის პროფესიონალური ვერსიის კომპიუტერში დაყენებაზე. ამისათვის საჭირო ინება გავუშვათ SIMATIC_STEP_7_Professional_v14.exe ფაილი (იხ. წინა სურათი). ეკრანზე გამოვა საწყისი გამოსახულება (იხ. სურათი 6-27:). აქ კი უკვე უნდა დააწვეთ ღილაკს <u>N</u>ext>.



სურათი 6-27: საწყისი გამოსახულება

ამის შემდეგ, ამ გამოსახულებაში (იხ. სურათი 6-28:) უნდა აირჩიოთ ენა - English ჩვენს შემთხვევაში და ისევ დავაწვეთ ღილაკს <u>N</u>ext>. მივიღებთ სურთს (იხ. სურათი 6-29:), სადაც ყოველგვარი მოქმედების გარეშე ვაჭერთ <u>N</u>ext>, რითაც დაიწყება პაკეტის ექსტრაქციის პროცესი (იხ. 6-30:).



სურათი 6-28: ენის არჩევა

🕼 STEP 7 Professional V14	l.0	—		\times					
SIEMENS									
Please en setup files created a	ter the folder where you want t s. If the folder does not exist, it utomatically.	o extrac : will be	t the						
The web package will b the package.	e extracted. You can only extract or e	extract an	d install						
Extract the package to	folder:								
Jemali AppData Local T	emp\SIMATIC STEP 7 Professional V14.0	B <u>r</u> ow	se						
 Overwrite files witho Extract the setup file 	 Overwrite files without prompting. Extract the setup files without being installed. 								
	< <u>B</u> ack <u>N</u> ext >	<u> </u>	ancel						

სურათი 6-29: პაკეტის ექსტრაქციის საქაღალდეს მითითება



სურათი 6-30: პაკეტის ექსტრაციის პროცესი

აქ შეიძლება შეწყდეს პროცესი და გამოგვიყანოს შეტყობინება (იხ. სურათი 6-31:), სადაც უნდა დავაჭიროთ ღილაკს <u>Y</u>es to All, რის შემდეგ ექსტრაქციის პროცესი ისევ გაგრძელდება (იხ. სურათი 6-31:).

STEP 7 Pr	ofessional V14.0		_		\times
SIEMI	INS				
STEP 7 P	rofessional V14.0				×
	The following file already exists:				
	Autorun.inf				
	Do you want to overwrite this file?				
<u>Y</u> es	Y <u>e</u> s to All <u>N</u> o	N <u>o</u> to All		<u>C</u> ancel	
					_
			Can	cel	
			<u>_</u>		

სურათი 6-31: ინსტალაციის პროცესის შეტყობინება



სურათი 6-32: პაკეტის ექსტრაქციის პროცესის გაგრძელება

ექსტრაქციის პროცესის დამთავრების შემდეგ ეკრანზე გამოვა გამოსახულება (იხ. სურათი 6-33:), სადაც პროგრამა გვთხოვს კომპიუტერის რესტარტს. ამას უნდა დავეთანხმოთ, რის შედეგად ინსტალაციის პროცესი ისევ გაგრმელდება.



სურათი 6-33: კომპიუტერის რესტარტის მოთხოვნა



სურათი 6-34: კომპიუტერის რესტარტის შემდეგ ეკრანზე გამოსახული ფანჯარა

კომპიუტერის რესტარტის შემდეგ ეკრანზე გამოისახება ფანჯარა (იხ. სურათი 6-34:), სადაც პირდაპირ უნდა დავაჭიროთ ღილაკს <u>N</u>ext>, შემდეგ გამოსულ ფანჯარაში (იხ. სურათი 6-35:) ისევ ვაჭერთ ღილაკს <u>N</u>ext> (იხ. სურათი 6-36:), შემდეგ ისევ <u>N</u>ext> (იხ. სურათი 6-37:).



სურათი 6-35: ინსტალაცის შუალედური ფანჯარა



სურათი 6-36: ინსტალაცის შუალედური ფანჯარა

```
SIMATIC STEP 7 Professional - Setup
```

SIEMENS	Totally Integrated Automation								
Welcome to STEP 7 Professional V14.0	P c	lease select th omponents wh	e product feat ich are alread	tures you wan ly installed, the	t to install. If ey will be un	you deselect installed.			
General settings	i Mi	nimal	1 📲 T	ypical	. II	User-defined			
 Configuration Install Overview Modify system System configuration Summary 	Engineeri	SIMATIC STE SIMATIC Wind Tools Migration Project m Automation Li Totally Integra TIA Portal Clo	P 7 Profession CC Basic V14.0 igration for STI igration for Wir icense Manage ited Automation ud Connector ramming S7-30	al V14.0 D EP 7 V5.4 SP5 nCC flexible 20 ar n UPDATER 00/400/1200/11	or higher 08 SP2/SP3, 500 controlle	SQL installation			
					6	Create desktor	icon		
	Hard Disk I	Drive space:							
	Drive	Size	Available	Required	Temp	Remaining	^		
	≪ C:\	115.1GB	28.8GB	4.4GB	110.0MB	24.4GB			
	Sec. (107.4GB	60.4GB	0.0B	0.0B	60.4GB	~		
	Target directory:								
	C:\Program Files\Siemens\Automation\ Browse								
				< Back	Next >	> G	ancel		

სურათი 6-37: პროგრამის შენახვის საქაღალდეს არ

ჩევა

აქ ინსტალაციის პროცესი შეგვეკითხება, თუ სად უნდა იყოს შენახული დაყენებული პროგრამა, უმჯობესია დავტოვოთ ისე, როგორც მითითებულია სიჩუმით და დავაჭიროთ ღილაკს <u>N</u>ext>. გამოჩენილ ფანჯარაში დავეთანხმოთ ლიცენზირების პირობებს (იხ. სურათი 6-38:) შესაბამისი ალმების აღნიშვნით და ისევ დავაჭიროთ ღილაკს <u>N</u>ext>, გამოსულ ფანჯარაში უნდა დავეთანხმოთ ახლა უკვე უსაფრთხოების კონტროლის პირობებს (იხ. სურათი 6-39:) და შემდეგ კიდევ უნდა დავაჭიროთ ღილაკს <u>N</u>ext>.



სურათი 6-38: ლიცენზირების პირობებზე დათანხმება

SIMATIC STEP 7 Professional - Setup

SIEMENS	Tota	lly Integrated	Automation					
Welcome to STEP 7 Professional V14.0	Security Control							
General settings	Correct functionality of STEP 7 Profession some of the security and permissions set accept these changes to continue the insta	al V14.0 requires tings on your syst allation.	changes to tem. You must					
 Configuration 	The following firewall settings will be modified Automation License Manager Service C:\Program Files\Common Files\Siemens\sws\almsrv	∆almsrv64x.exe	^					
Install	Network access range: Subnet							
 Overview 	The following file system rights will be set							
	C:\ProgramData\Siemens\Automation							
Modify system	+ Inherit from parent the permission entries that apply to child objects. Include these with							
 System configuration 	 Replace permission entries on all child objects wi child objects. 	th entries shown he	re that apply to					
O Summany	Rights for this folder, subfolders and files will be ad "Users"	Allow	Denv					
Junnary	Full access	-	-					
	Browse folders / execute file	+						
	List folder contents / read data	+	· .					
	Read attributes	+						
	Create files / write data	+						
	Create folders	+	-					
	Write attributes	+	-					
	Write extended attributes	+	· ·					
	Delete subfolders and files	+	-					
	Delete Read permissiona	+	-					
	Characterizations	•	- v					
	Save report	Print rep	ort					
	I accept the security and permissions settings on the security and permissions settings.	iis computer.						
	< Back	Next >	Cancel					

სურათი 6-39: უსაფრთხოების კონტროლის პირობებზე დათანხმება



სურათი 6-40: ინსტალირების დაწყების ფანჯარა

შედეგად მივიღებთ ინსტალირების დაწყების ფანჯარას (იხ. სურათი 6-40:), სადაც ვაჭრეთ ღილაკს Install და დაიწყება ინსტალაციის პროცესი, რომელიც გაგრძელდება რამდენიმე წუთის განმავლობაში (იხ. სურათი 6-41:).



სურათი 6-41: ინსტალირების პროცესი

ინსტალირების პროცესის დამთავრების შემდეგ ეკრანზე განმოვა ლიცენზიის გადაცემის ფანჯარა (იხ. 6-42:), სადაც, ვინაიდან 21 დღიან ტრაიალ ვერსიაზეა ლაპარაკი, ამიტომ უნდა დავაჭიროთ Skip license transfer ღილაკს თითი.



სურათი 6-42: ლიცენზიის გადაცემის ფანჯარა

აქ ინსტალირების პროცესი გაგრძელდება კიდევ რაღაც დროის განმავლობაში (იხ. სურათი 6-43:), რის შემდეგაც პროგრამა მოგვთხოვს კომპიუტერის გადატვირთთვას, რასაც უნდა დავეთანხმოთ და დავაჭიროთ ღილაკს Restart თითი (იხ. სურათი 6-44:).



სურათი 6-43: ინსტალირების პროცესის გაგრძელება

ამით TIA Portal-ის პროგრამის ინსტალაციის პროცესი დამთავრდება, ხოლო დესკტოპზე გამოვა პროგრამის ნიშნები (იხ. სურატი 6-45:), სადაც პირველი ნიშნის TIA Portal v14-ის დანიშნულებაა უკვე ინსტალირებული პროგრამის გაშვება პტროექტებთან მუშაობის უზრუნველსაყოფად.



სურათი 6-44: კომპიუერის გადატვირთვის მოთხოვნა



სურათი 6-45: პროგრამის დესკტოპის ნიშნები

კომპიუტერის გადატვირთვის შემდეგ ეკრანზე გამოვა გამოსახულება (იხ. სურათი 6-46:)



სურათი 6-46: TIA Portal-ის ნიშანი

ამის შემდეგ პროცესი გადადის TIA Portal-ის დაყენების დამამთავრებელ ეტპზე, ეკრანზე გამოვა ჯერ გამოსახულება (იხ. სურათი 6-47:),

SIMATIC STEP 7 Professional - Setup			
SIEMENS		1	otally Integrated Automation
Vielcome to STEP 7 Professional VI4.0 General settings Configuration Install Overview Modify system System configuration Summary	1 5	License Transfer License transfer could not be performe medium. Please transf the license key right cow, or do it a later time by sta application. Manual license transfer	d because of missing license key medium and retry license transfer ring Automation License Manager Retry license transfer Skip license transfer

სურათი 6-46: ლიცენზირების ფანჯარა

აქ პროგრამის დაყენება პრაქტიკულად დამთავრებყულია, რადგან ლიცენზირების პრეტენზია არა გვაქვს, ვაჭერთ ხელს Skip license transfer, შედეგად ეკრანზე გამოვა პროგრამის დამთავრების პროცესის ბოლო ეტაპი (იხ. სურათი 6-48:)



სურათი 6-48: TIA Portal-ის ინსტალაციის დამამთავრებელი ეტაპი

ამის შემდეგ, ჩვენ მაშინვე უნდა დავაყენოთ სიმულიაციის პროგრამა თუ არა გვაქვს კონტროლერი, ამისთვის უნდა გავუშვათ SIMATIC_PLCSIM_V14.exe ფაილი (იხ. სურათი 6-26:). ამ ფაილის დაყენება უმეტესწილად ანალოგიურია SIMATIC_STEP_7_Professional-ის დაყენებისა, ამიტომ მის დაყენებას აქ არ შევეხებით.

6.2 პრაქტიკული სამუშაო 2

6.2.1 TIA Portal-ის გაშვება

პროგრამის დაყენებისა და გაშვების შემთხვევაში გაიხსნება ჩვენი ახალი პროექტის სასტარტო ფანჯარა. პირველ რიგში საჭიროა შევქმნათ ახალი პროექტი (ან გამოვიძახოთ უკვე შექმნილი). ახალი პროექტის შექმნისათვის საჭიროა დავაწკაპუნოთ ღილაკზე Create new project (იხ. სურათი 6-49:), გამოსულ ფანჯარაში შევავსოთ ველები პროექტის სახელწოდების Project name, ჩვენს შემთხვევაში ვარქმევთ მას J. Grig_1, მეორე ველში ვუთითებთ და ვწერთ იმ საქაღალდეს სადაც ბუნდა განთავსდეს პროექტი, ჩვენს შემტხვევაში დესკტოპის Work საქაღალდეში, შემდეგ ველში ვუთითებთ ავტორის სახელს Jemal-ჩვენს შემთხვევაში და ვაჭერთ შექმნის ღილაკს Create. დაიწყება პროექტის შექმნის პროცესი, რომლის დამთავრების შემდეგაც ეკრანი მიიღებს სახეს (იხ. სურათი 6-50:).

ამის შემდეგ დავაჭიროთ თაგვის ღილაკით ეკრანის მარცხენა ქვედა კუთხეში Project View და გადავიდეთ პროექტის სამუშაო ფანჯარაში (იხ. სურათი 6-51:).

VIA Si	emens				_ ¤ X
					Totally Integrated Automation PORTAL
s	tart			Create new project	
			Onon existing project	Project name:	J.Grig_1
		1 1	Open existing project	Path:	C:IUsersUemalilDesktoplWork
			🥚 Create new project	Author:	Jemal
			Migrate project	Comment:	
					✓
		1997			Create
			Welcome Tour		
		15			
			First steps		

სურათი 6-49: ახალი პროექტის შექმნა

Via Siemens - C:\Users\Jema	ali\Desktop\W	orkV. Grig_1V. Grig_1		_ = X
				Totally Integrated Automation PORTAL
Start			First steps	
Devices & networks	*	Open existing project	Project: "J. Grig_1" was opened successfully. Please select the next step:	
PLC programming	٢	Create new project Migrate project	Start	
Motion & technology	*	Close project	Devices 8 Configure a device	
Visualization			PLC programming 🗇 Write PLC program	
Diagnostics	<u></u>	 Welcome Four First steps 	Motion & Configure technology	
			Visualization Configure an HM screen	
		Installed software		
		💮 Неір		
		🚯 User interface language	Project view Open the project view	
Project view		Opened project: C:\Users\Jemali\D	Desktop\Work\U. Grig_1\U. Grig_1	

სურათი 6-50: TIA Portal v13-ის სასტარტო ფანჯარა



სურათი 6-50: პროგრამის სამუშაო ფანჯარა

6.2.2 IP მისამართის მინიჭება

აქ ჩვენ დაგვჭირდება დაუკავშიროთ ჩვენი მოშორებული მოწყობილობა (პლკ) პროგრამას, მაგრამ მირითადი პრობლემა არის ის, რომ მას ჯერჯერობათ არ აქვს მინიჭებული IP მისამართი. მისამართის მინიჭებისათვის უნდა დავაწკაპუნოთ ღილაკზე Online Accses (ოპერატული ხელწვდომა), რითაც გაიხსნება შესაბამისი საქაღალდე, სადაც ვირჩევთ ჩვენს ქსელურ დაფას (იხ. სურათი 6-48:) (ჩვენი ლეპტოპისათვის (Lenovo ThinkPad T420) ეს არის Intel(R) 82579LM Gigabit Connection და განვაახლოთ ხელწვდომი მოწყობილობები Apdate Accessible devices ღილაკზე თაგვის მარცხენა ღილაკის დაწკაპუნებით.



სურათი 6-51: ქსელური დაფის არჩევა

როგორც სურათიდან ჩანს განახლების შემდეგ აქ გამოჩნდება ახალი მოწყობილობა CPUcommon [192.168.0.1] ან შეიძლება მისი ანალოგი (pic_1 [192.168 0 1]), რომელზედაც ორჯერ დაწკაპუნებით გამოჩნდება ჩამოშლადი სია (იხ სურათი 6-52:), მათ შორის გამოვა ნიშანი Online & Diagnostic, რომელზედაც ასევე ორჯერ დაწკაპუნებით დაიწყება ამ მოწყობილობის დიაგნოსტიკის პროცესი და გარკვეული დროის გავლის შემდეგ შემდეგ ეკრანი მიიღებს სახეს (იხ. სურათი 6-54:).



სურათი 6-52: Online & Diagnostic რეჟიმში გადასვლა

VA Siemens							_ •	2
Project Edit View Insert Online Options Too	ls Window Help					Totally Integra	ated Automation	
📑 🞦 🖬 Save project 🚢 🐰 🏥 油 🗙 🍤 🕯 (×±₩₩₩₩₩₩₩	Go online 🖉 Go offline 🛛 🛔 🖪	* 🖃 🛄			rouny megn	PORTAL	
Project tree	● Online access → Intel(R) 825	79LM Gigabit Network Connection	▶ CPU common [192.168.0.1]			_ # = ×	Online 🗊 🗉 🕨	
Devices							Options	_
2 B 0 0 I	Diagnostics General	General				1		online t
🗧 👻 🔚 Online access	Maximizes/minimizes the Overview.	Module						00
Display/hide interfaces	Diagnostics buffer						CPUcommon [192.1	్
🗟 🔸 🛄 USB (S7USB) 🕴	Cycle time		CPU 1212C AC/DC/Rly				RUN / STOP	1
Intel(R) 82579LM Gigabit Network Conn	PROFINET interface [X1]	Order number:	6ES7 212-18E40-0XB0				ERROR	B
C Update accessible devices	Functions	Hardware:	1				MAINT	sks
CPUcommon [192.168.0.1]		Firmware:	V 4.0.0				1	
Program blocks	_							Щ
Technology objects		Rack: 0						E
▶ C PLC data types		Slot 1						ari
Intel(R) Centrino(R) Ultimate-N 6300 AGN	9	-						Sa
Microsoft Wi-Fi Direct Virtual Adapter #3	9						< 11 >	÷
Microsoft Wi-Fi Direct Virtual Adapter #4	<u>64</u>						✓ Cycle time	
PC Adapter [MPI]	?	Module information						
PC Internal								
Card Reader/USB memory		Module name:	CPUcommon					
Card nesdenoso memory		Plant designation:						
		Location ID:					6 6	
		Installation date:				~		
				9 Properties	1 Info Diagnostics		Shortest: 0	
				Stroperaes	S mo Sugnostic.	, , , , , , , , , , , , , , , , , , , ,	Longest: 0	
	General Cross-reference	es compile						
			[c. m.]a	Deta Time				
<	Message Complete for devices on integration of the second seco	torface latel(0) 835701 M Ciga hit Naturak	Go to ?	2/2/2021 2:48:22 PM				
✓ Details view	Scanning for devices comp	leted for interface Intel(P) 825791 M Giga	hit Network Connection	3/2/2021 3:48:43 PM				
	Scanning for devices comp	ne teo tor interface inter(k) 6257 selvi diga	bit the throne connection	5/2/2021 5/40/451 M			1	

სურათი 6-53: Online & Diagnostic რეჟიმში გადასვლის რეზულტატი

აქ ჩვენთვის ყველაზე საინტერესოა ფუნქციებში Functions → Accain IP Adress მისამართის (IP) მიკუთვნება. ვაწკაპუნებთ ამ ფუნქციაზე და ვიღებთ შესაბამის ფანჯარას (იხ. სურათი 6-54:). გამოსულ ველებში ვწერთ ჩვენი მოწყობილობის (თუ უკვე ჩაწერილი არ არის ავტომატურად) IP მისამართს, მასკას და ვაჭერთ ღილაკს Assain IP Adress (IP მისამართის მიკუთვნება).

🕒 🗔 Save project 🔳 🐰 🗉 🗎 🗙 🏷	± (°		🖉 Goonline 🖉 Gooffline 🛔 🎚 🖟 🗶 📃 🛄	egrau	PORTAL	_
roject tree		Online access > Intel(R)	82579LM Gigabit Network Connection 🔸 plc_1 [192.168.0.1] 📃 🖬 🖬	iX	Online 🗊 🔟 🕨	
Devices					Options	0
3 O O 8	1	 Diagnostics General Functions 	Assign IP address		✓ CPU operator	inline to
Conline access		Assign IP address	MAC address: 28 -63 -36 -80 -25 -28 Accessible devices		Not supported	ŝ
U USB [S7USB] Update accessible devices Update & diagnostic Untent(R) Centino(R) Ultimate 4 6300 AGN Untent(R) U		, a sign name	IP address: 192.168.0.1 Subnet mask: 255.255.0 Use router Router address 0.0.0 Assign IP address			Tasks 📑 Libraries
Tage 2 - The Contraction of	n] 1999				Cycle time	

სურათი 6-54: IP მისამართის მიკუთვნება

ახლა უკვე ჩვენ მოწყობილობას ექნება თავისი IP მისამართი. ამის შემდეგ, ამავე ფანჯარაში თუ დავაწკაპუნებთ დიაგნოსტიკის ფუნქციას Diagnostic, მივიღებთ გამოსახულებას (იხ. სურათი 6-55:).

V13	Siemens	_											فكعد
Pr	oject Edit View Insert Online Options To		Window Help							1	Fotally Integr	ated Automation	
2	🛉 🎦 🔜 Save project 🔳 🐰 🗉 🛍 🗙 🍤 ±	Cai ±	▓▚ᡛᡦݠŖጶ‹	So online 🖉 Go offline 🛔 🖪 🖪	* 🗄 🛄							PORTA	AL
	Project tree										_ - •	Online 🗐 🔟	
	Devices											Options	
Online & Diagnostics	Constant Section 275 Constant Cons		 Diagnostics Functions Assign IP address Settime Firmware update Assign name Reset to factory settings 	Diagnostics General Module Order number: Hardware: Firmware: Rack: 0	CPU 1212C AC/DC/Rly 6657 212-18E40-0X80 1 V 4.0.0							CPU operator. CPUcommon [192 RUN/STOP ERROR MAINT	nline tools (III) Tasks Libraries
	Infelia Centinnolu Utimate 4 6300 AAN Merosoft WFI Direct Virtual Adapter #3 Acrosoft WFI Direct Virtual Adapter #4 Cr Adapter [Mm] PC Ficture [Mm] Ter Cinternal			Nodule information Module anne: Plant designation:	CPUcommon							< III ✓ Cycle time	> "
			General Cross-reference	es Compile				C. Properties	1 Info	U Diagnostics		Shortest: Current/last: Longest:	0
	< II Details view	>	Message Scanning for devices on init Scanning for devices comp Scanning for devices on init Scanning for devices comp The parameters were trans	terface Intel(R) 82579LM Gigabit Network leted for interface Intel(R) 82579LM Gigal terface Intel(R) 82579LM Gigabit Network leted for interface Intel(R) 82579LM Gigal ferred successfully.	Connection was started oit Network Connection Connection was started oit Network Connection	Go to ?	Date 3/2/2021 3/2/2021 3/2/2021 3/2/2021 3/2/2021	Time 3:48:33 PM 3:48:43 PM 4:08:19 PM 4:08:27 PM 4:09:58 PM					

სურათი 6-55: დიაგნოსტიკის ფანჯარა

თუ გადავალთ Diagnostics General-ში მაშინ მივიღებთ სურათს 6-56:

VA	Ma Siemens - Project2₽X									
Pr	oject Edit View Insert Online Op P 🎦 🗔 Save project 🔳 💥 🗐 🗎	tions Tools Window Help K ≌つ±C#± 🏹 🖥 🛄 🗓	🖳 🐺 🖉 Go online 🖉 Go offline 🛔 🖪 🖪 🗱 🗶 🖃 🛄	grated Automation PORTAL						
	Project tree 🔲 🖣	heros AR8121/AR811	/AR8114 PCI-E Ethernet Controller → PLC_1 [192.168.0.1] 🛛 💶 🖬 🖬 🗙 🥌	Online 🗊 🗉 🕨						
	Devices			Options						
Online & Diagnostics	Project tree ▲ Devices	Uneros AR8121/AR811 Diagnostics General Diagnostic status Diagnostics buffer Cycle time Memory PROFINET Interface [X1] Functions	AR8114 PCLE Ethernet Controller > PLC_1 [192.168.0.1]	Online I I I Online tools Options Options PLC_1 [192.168.0.1] RAINT Cycle time Cycle time Shortest: Currentilast: Longest:						
	<u> </u>		Manufacturer description: Steller NAG Serial number: S < < < > < < < < < < < < < < < < < < <							
	> Reference projects		🖳 Properties 🔂 Info 🔂 Diagnostics 🗖 🗏	<						
	> Details view	General Cross-refere	ces Compile	> Memory						
	Portal view Dverview	📡 Online & dia	1 Scanning for devic	es completed for int						
6			S W 3	▲ ♦) 😼 14:48 14.12.2018						

სურათი 6-56: Diagnostic General-ში გადასვლა

ამ ფანჯარაში ჩვენ შეგვიძლია ვნახოთ ინფორმაცია მიერთებული მოწყობილობის შესახებ, აგრეთვე მივანიჭოთ მათ სახელი (იხ. სურათი 6-57:) Functions Assign Name-ში გადასვლით.

roject Edit View Insert Online Options Tools Window Help Totally Integ						
Project tree	Online access + Intel(R) 82579LM Gigabit Network Connection + PLC [192.168.0.1]					
Devices		Options				
Image: Construction Image: Construction Image: Construction General	Assign name	▲ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■				
Online access Diagnostic status Diagnostic buffer Cycle time Cycle time Memory PROFINET interface [x1] Memory PROFINET interface [x1] Cycle time Memory PROFINET interface [x1] Cycle time Memory Rofine & Set time Technology Cycle data Cycle data Set time Sign name Cycle data Set time Sign name Sign name	Configured PROFINET device PROFINET device name: plc_1 Type: 57-1200	PLC_1 [192.168.0.1] RUN / STOP ERROR MAINT				
Cill Intel(R) Centrino	Device filter Only show devices of the same type Only show devices with bad parameter settings Only show devices without names Errsichbare Teinehmer im Netwerk:	 ≤ III > ∨ Cycle time 				
Caro ReadenUSB me	IP address MAC address Type Name Status					
	LED flashes Update Assign name	Shortest: 1 Currentilast: 1 Longest: 2				

სურათი 6-57: მოწყობილობზე სახელის დარქმევა

გამოსულ ფანჯარაში PROFINET device name სახელის ველის შევსების შემდეგ (pls_1 ჩვენ შემთხვევაში) უნდა დავაჭიროთ ღილაკს Assign Name (სახელის მიკუთვნება), იქვეა შუქდიოდების ციმციმის ღილაკი, რომელიც გამოიყენება იმისათვის, რომ აციმციმდეს შესაბამის პლკ-ს წინა ფრონტზე დაყენებული შუქდიოდები (იხ. სურათი 6-58:). თუ მოვნიშნავთ LED flashes ალამს მასზედ თაგვის ღილაკის დაწკაპუნებით, მაშინ ჩვენ კონტროლერის წინა პანელზე დავინახავთ შუქდიოდების ციმციმს. ეს ფუნქცია იმისათვისაა საჭირო, რომ თუ ჩვენ გვაქვს მრავალი პლკ, დავრწმუნდეთ, რომ ნამდვილად ვმუშაობთ იმ მოწყობილობასთან, რომელიც ჩვენ გვჭირდება. ამ ღილაკზე თითის დაჭერით პლკ-ს წინა პანელზე დაიწყებს შუქდიოდი ციმციმს, თანაც იმ პლკ-ზე, რომელთანაც მუშაობას ვაპირებთ და ამით ვრწმუნდებით, რომ ეს ის მოწყობილობაა, რომელიც ნამდვილად ჩვენ გვჭირდება.

Project Edit View Insert Online Options Tools Window Help Totally Integrated Aut								ated Automation PORTAL			
	Project tree	4	Online access Intel(R) 8257	line access → Intel(R) 82579LM Gigabit Network Connection → PLC_1 [192.168.0.1]							Online 🗊 🗉 🕨
	Devices										Options
	1 C O O 1	2	▼ Diagnostics	Assign name						^	
stic			General	Posigir name						=	✓ CPU operator
2	🕶 🙀 Online access		Diagnostic status								
ia,	🍸 Display/hide int		Diagnostics buffer		_						PLC_1 [192.168.0.1]
	USB [S7USB]	1	Cycle time		Cor	ifigured PROFINET de	evice				RUN / STOP
ē	🕶 🛄 Intel(R) 82579 L	1	Memory			PROFINET device name:	plc_1				ERROR
Ξ.	PUpdate acce		PROFINET interface [X1]			Type:	\$7,1200				
•	▼ 1 PLC_1 (192.16		Functions				37 1200				MANNI L
	😵 Online &		Assign IP address								
	🕨 🚘 Program		Set une								
	🕨 🙀 Technology		Firmware update								
	PLC data t		Assign name								
	Intel(R) Centrino	80	Reset to factory settings								
	🕨 🗎 Microsoft Wi-Fi Dir	80			Device filter						<
	🕨 🗎 Microsoft Wi-Fi Dir	80				Only show devices of	the same type				✓ Cycle time
	🛨 🎦 PC Adapter [MPI]	₩,									
	R Update acce										
	PC internal	1		Only show devices without names							
	TeleService [Auto	1	-								
	Card Reader/USB me			Erreichbare Teilnehmer im Netzverk:							-
				IP address	MAC address	Туре	Name	Status			ו ו
											Charles 1
											Current/last: 1
											Longest: 2
						D flashes	Update	Assign name			
				et listes opere roughtimme							

სურათი 6-58: შუქდიოდების აციმციმების დიოდი

შემდეგი მოქმედება, რაც ჩვენ შეგვიძლია გავაკეთოთ, ეს არის ჩამოყრა საქარხნო დაყენებებზე (იხ. სურათი 6-59:) Reset to factory settings ღილაკის არჩევით და შემდეგ Reset ღილაკის გამოყენებით. ამ ღილაკის გამოყენება ხშირად ხდება საჭირო იმ შემთხვევებში, თუ ჩვენ მრავალ ექსპერიმენტს ვატარებთ სხვადასხვა მოწყობილობასთან და ამ დროს იგი დაეკიდება. ამ ღილაკის გამოყენებით შესაძლებელი ხდება მისი მუშაობის აღდგენა პრაქტიკულად ნებისმიერი მდგომარეობიდან.

8	1 C O O	1	 Diagnostics 	Reset to factory settings	L		
÷.			General		~	CPU operator	.] #
Ê.	Online access		Diagnostic status				18
<u> </u>	Displayhide int		Diagnostics buffer		1	PLC_1 [192.168.0.1]	۱° ار
ē	USB [S7USB]	100	Cycle time		1.7	RUN / STOP	-
	▼ Intel(8) 825791		Memory	IP address: 192.168.0 .1		EPROP 0	
ų.	A? Update acce	1	 PROFINET interface [X1] 	PROFINET device name: plc.1			as i
ō	PLC 1 [192.16		 Functions 				ŝ
	V. Online &		Assign IP address	Detrain IP address	1		
	Program		Set time		1		
	Fechnology		Firmware update	O belete in address	1		E.
	PLC data t		Assign name	Reset	1		ari
	Intel(R) Centrino	*	Reset to factory settings		L		es
	Microsoft Wi-Fi Dir	80			<		4
	Microsoft Wi-Fi Dir	80			×	Cycle time	
	🕶 🛄 PC Adapter [MPI]	₩,					

სურათი 6-59: მოწყობილობის დაკიდებული მდგომარეობიდან გამოსვლის ღილაკი

6.2.3 ახალი მოწყობილობის დამატება

ამის შემდეგ, საჭირო იქნება ახალი მოწყობილობის დამატება ჩვენს პროექტში, ამისათვის შევდივართ ახალი მოწყობილობის არჩევის ფანჯარაში (იხ. სურათი 6-60:), სადაც ვირჩევთ Add New Device (ახალი მოწყობილობის დამატება) და ჩამოშლადი სიიდან ვირჩევთ ჩვენს ხელთ არსებული CPU-ს მოდელის ვერსიას, ვაწკაპუნებთ ღილაკს OK და მცირე დაყოვნების შემდეგ ვიღებთ სურათს რომელიც ნაჩვენებია სურათზე 6-61:.



სურათი 6-60: ახალი მოწყობილობის დამატება პროექტში



სურათი 6-61: ცენტრალური პროცესორის ვერსიის არჩევა

6.2.4 S7-1200-ის გამართვა და შორიდან მიერთება

ახლა საჭიროა ამ მოწყობილობის გამართვა, ამისათვის ორჯერ დავაწკაპუნოთ ამ ფიგურის მარჯვენა ქვედა კუთხეში და მივიღოთ გამოსახულება (იხ. სურათი 6-62:).





VA Siemens - Project1		_ a ×
Project Edit View Insert On 🌁 🎦 🔒 X	line Options Tools Window Help 🗄 🚡 🗙 🏷 🛨 (📲 🖫 🔛 🔝 🔛 📓 🖉 🎜 Go online 🖉 Go offline 🏭 🌆 🐺 📮 🛄 💿	Totally Integrated Automation PORTAL
•	Project1 PLC_1 [CPU 1212C DC/DC/DC]	_ 🖬 🖬 🗙 Hardw 🖷 💷 🕨
		🚰 Topology view 👗 Network view 📑 Device view Options 🕮
	# PLC_1 💌 🔛 🏹 🗄 🕰 ± 100% 💌	
×.		▲ Catalog
- two		Search> Mu MT
č		V Filter
	AC.	- o 🔽 🖓 🖓 🖓 🖉
	The second secon	Signal boards
		Communic
	Mack_0 strates source-su	DQ d
	103	
		Properties Info Diagnostics
General IO tags S	iystem constants Texts	ا الله المعادي المحافظ
Ethernet addresses	Ethemet addresses	<u>^</u>
Operating mode	Interface networked with	
Advanced options		bra
Hardware identifier	Subnet: Not networked	
DI8/DO6	Add new subnet	
High speed counters (HSC)		
Pulse generators (PTO/PWM)	IP protocol	
Startup	Set IP address in the project	
Cycle	IP address: 192.168.0.1	
System and clock memory	Subnet mask: 255 . 255 . 0	
 Web server 	Use router	
Time of day	Router address: 0 0 0 0	sing and black in will be used discussion also derive
User interface languages	IP address is set directly at the device	don enabled. If will by set directly on the device
Connection resources		
Overview of addresses	PROFINET	
		V Information
Portal view	th PLC_1	✓ Project Project1 opened.
🕙 🤶 🚞	K o 🔣 🖄 🧿 🛷 🚻	EN 🕐 🗭 🔺 🌒 🍡 16:35 14:12:2018

სურათი 6-63: მოწყობილობის გამართვის ფანჯარა
აქ შეიძლება დაგვჭირდეს შუა ქვედა ფანჯრის ამოწევა, რისთვისაც უნდა გამოვიყენოთ თაგვის პროცედურა. შედეგად გაიხსნება აწყობების ფანჯარა, სადაც IP მისამართი უკვე დაყენებულია და ჩვენ შეგვეძლება დავუკავშირდეთ მას შორიდან (იხ. სურათი 6-63:), ამისათვის უნდა დავაწვეთ ღილაკს Go online და ვღებულობთ გამოსახულებას (იხ. სურათი 6-64:).

M Siemens - Project1							_ • • >
Project Edit View Insert Online Options	Tools Window He	lp				Totally Integra	ated Automation
🕒 🕒 🖬 Seve project 📓 🐰 🗉 🕞 🗙 🎙	91 (P1 🖬 🗟 🗉	🗄 🔛 🔛 🍠 Co online 🖉 Co ottine					PORTAL
) 🗢 🛛 🛛 🖉	ect1 + PLC_1 (CPU 1	1212C DODODC]				_ # = X	Hardw 🗊 🗉 🕨
and a second contract of the second se				🚰 Topology view 🛛 📥	Network view	Device view	Options E
# D	C1	💌 🔛 🏹 🖏 ± 100%				E	
4						^	✓ Catalog
							dearcho Hig Hill
		A					Filter
5	4	P				12	• 📺 CPU 🕹
1							Signal boards
•	501	ect devices for opening the online co	nnection	×		1 2	Communic
		ume	Type	Goonline			Battery boards CI
	Rack_0 Pi	.c.1	CPU 1212C DC/DC/DC	8			00
							DIDQ
						×	• <u>a</u> 4
٤						2	AQ III
PLC_1 [CPU 1212C DC/DC/DC]					nfo 😩 🖳 Diagn	ostics	Communic_
General IO tags System const	tants Texts						• 📺 Technology
Ethernet addresses	a address of					A	
Time synchronization	Carcenter -						
Operating mode Interf	face networked v						
Hardware identifier						-	
DISDO6				Go online Cancel			ľ
► AI2				and a burn			
High speed counters (HSC) IP pro	stocol						
Pulse generators (PTO/PVM)							
Overview of addresses V PROF	INET						< = >
C = >						×	> Information
Portal view Overview	1 PLC_1					Project Project1 open	d.

სურათი 6-64: შორიდან მიერთება 1 ეტაპი

ამის შემდეგ ჩანართ ფანჯარაში ჯერ უნდა მოვნიშნავთ ალამი Go online შემდეგ კი მასთან ახლოს მდგომი მეორე ალამი. შემდეგ დავაწვებით ღილაკს Go online იხ. სურათი 6-65:), რის შედეგადაც მივიღებთ გამოსახულებას (იხ. სურათი 6-66:).

All memory , confects					
Project Edit View Insert Online Options	s Tools Window Help			Totally Integr	ated Automation
🕒 🕒 Save project 📓 🗶 🗷 🔍 💌	🐴 🗄 🖓 🔛 🔛 🔛 🗳 🖉 Co online 🖉 Co online				PORTAL
0 Proj	sject1 + PLC_1 [CPU 1212C DODODC]			_ # = ×	Hardw 🗊 🗉 🕨
			🖉 Topology view 🛛 🛔 Network view	n 🛐 Device view	Options 🛃
# E	LC_1 💌 🖬 🏹 🖬 🗔 ± 100%				
ett.				<u>^</u>	✓ Catalog
ŧ					deartha Ng HL
ē	*				Filter 8
5	B.C.			12	• (a) CPU 🗳
8	-		14		Signal boards
	Select devices for opening the online of	ennection	<u>^</u>	15	+ atteryboards
	Rack_0 Name	Type	Go online		 In the second sec
	R.C.1	CPU 1212C DODODC	×		P DQ
	103			-	
<				> 🗉	▶ 1 AQ
PLC_1 [CPU 1212C DC/DC/DC]			vio 🔒 😼 Di	agnostics 👘 👘 🔻	 AliAQ AliAQ
General IO tags System const	tants Texts				Technology
Ethernet addresses				A	
Time synchronization	et addresses				U
Operating mode Interf	rface networked v				
Hardware identifier					arie
+ DISDO6			Go gnine Cancel		-
► A2					
Prigh speed counters (PSC) IP pro Price generators (PSC)	rotocol				
Connection resources					2
Overview of addresses V PROF	FINET				> Information
Portal view Dveniew	th PLC 1			Project Project1 open	rd.

სურათი 6-65: შორიდან მიერთება 2 ეტაპი

With Stemens - Project1 Project Edit View Insert Online Options Too Image: State Sta	ls Window Help	Co oplina		n rm i .	v • •			Totally Integr	_ ■ X ated Automation
			2. go onnie 103 H					3 = 2	
t more the terms of terms	Go online	ionfigured access no levice LC_1	des of "PLC_1" Device type CPU 1212C DC/D	Slot 1 X1	Type PN/IE	Address Not configured	Subnet	w Device view	Options Hardware C
Devices & d			Type of the PG/PC inte PG/PC inte	erface:	Please select		•	Device data II ← I I ← I	Filter at a g Filter g Signal boards COmmunic Battery boards D D
▼ PLC_1 [CPU 1212C DC/DC/DC]	c	Con ompatible devices in	nection to interface/s 1st ga h target subnet: Device type	teway:		Show al	Il compatible devices	▼ > ♥ Niagnostics	DQ DI/DQ DI/DQ AI AQ AI/AQ A AC Communic T
General IO tags System constants Ethernet addresses Time synchronization Operating mode Advanced options Hardware identifier DIBIDO6	Flash LED		bene ype	Type	1	1001033			Fechnology Sks
Al2 High speed counters (HSC) Fulse generators (PTO/PWM) Startup Cycle Communication load	Online status information:						<u>R</u> efresh		
System and clock memory Veb server Time of day User interface languages Protection	Display only problem rep	ports				Go <u>o</u> nli	ne <u>C</u> ancel		
Connection resources Overview of addresses PROFINET								~	< III > > > > > > > > > > > > > > > > >
		Ø 🕴	A C					♥ Project Project1 open	ed. • (10) 16:40 • (14.12.2018

სურათი 6-66: შორიდან მიერთება 3 ეტაპი

Ma Siemens - Project1								- 7
Project Edit View Insert Online Options	ools Window Help						Totally Integ	rated Automation
📑 📑 🔚 Save project 📑 🐰 🏥 🗎 🗙 🍤 🕯	C*± 🖬 🖥 🛄 🔟	🖳 🔝 💋 Go online 🖇	🖉 Go offline 🛛 🛔 📗	■ × ⊟ 🔲			, ,	PORTAL
▶	Coopling						# = ×	Hardw 🔳 🔳 🕨
	do onine						ew 🛐 Device view	Options
Ht PLC_1	-	Configured access no	odes of "PLC_1"					
strain and		Device	Device type Slo	t Type	Address	Subnet	~	✓ Catalog
the state of the s		PLC_1	CPU 1212C DC/D 1	KT PN/IE	Not configured		=	<search> Mi MT</search>
								Filter
sa							= 0) The CPU
							evice	🕨 🛅 Signal boards
			Type of the PG/PC interfac	e: Please select		-		🕨 🛅 Communic
			PG/PC interfac	e: Please select		1 🔿 🔍	- 60	Battery boards
-				PN/IE				
				TeleService	2			> DI/DQ
			ist gotern				~	AI 🛛
<					Show all (compatible devices	> 🗉	AQ 📄
PLC_1 [CPU 1212C DC/DC/DC]		Compatible devices i	n target subnet:				liagnostics 🛛 🗆 = 🥆	AI/AQ
General IO tags System constant	s s	Device	Device type 1	ype A	Address Targ	et device		Technology
Ethernet addresses								
Time synchronization Ethernet a	ic realized							
Operating mode Interfac	±							1
Advanced options	Elash I ED							
Hardware identifier								
AI2		-						
High speed counters (HSC)						Refresh		
Pulse generators (PTO/PWM)	Online status informatio	on:						I
Startup =								I
Communication load								
System and clock memory	Display only probler	m reports						
Web server								
Time of day					Go online	Cancel		I
User interface languages								
Connection resources								
Overview of addresses Y PROFINE	т							< III >
							*	> Information
Portal view Overview	J PLC_1						Y Project Project1 oper	ied.
🔊 🤌 🚞 🔣 👩		🔿 🦪 🛛	IA.				EN 😰 🛱	 (i) 16:43 1413

სურათი 6-67: შორიდან მიერთება 4 ეტაპი

შემდეგ ჩამოვშლით ველს Type of the RG/RC interface და ავირჩევთ შეერთების ტიპს PN/FE (იხ. სურათი 6-67:). რაც გამოიწვევს კომპიუტერთან ჩვენი მოწყობილობის შეერთებას (იხ. სურათი 6-68:) და შეერთების წარმატებით განხორციელების შემთხვევაში ჩვენი ეკრანი მიიღებს საბოლოო სახეს (სურათი 6-69:).

Via Siemens - Project1										_	×
Project Edit View Insert Online Options Tools	s Window Help								Totally Integr	tod Automation	
📑 🔁 🔒 Save project 🚊 🐰 🗐 🗎 🗙 🍤 🛨 (*	4 🖬 🖥 🖽 🛍 😫	🔯 🧭 Go online 🖉	Go offline		× 🗆 💷				rotally integra	PORTAI	L
▶	DIC 1 [CDI 1212C DC	יוחכיוחכי							_ # = X	Hardw 🗊 🗉 🕨	
	30 online							×	Device view	Options	
AL DICL		Configured access node	es of "PLC_1"							-	E
2		Device	Device type	Slot	Туре	Address	Subnet			-	- 4
N 01		PLC_1	CPU 1212C DC/D	1 X1	PN/IE	Not configure	d		음	✓ Catalog	
1 et									_	<search> init init</search>	l ő
a										🛃 Filter	a
· · · · · · · · · · · · · · · · · · ·									= D	CPU	ß
									- 2	Signal boards	
		т	ype of the PG/PC inter	face:	PN/IE		•		data	Communic	8
	_ <u></u>		PG/PC inter	face:	Atheros Af	R8121/AR8113/AR8	114 PCI-E Ether 💌	1		Battery boards	9
	۵			bnet:	Direct at slot		v	٠			ine
							*			DI/DQ	to
								~	~	🕨 🛅 Al	-s
<									> 📃	🕨 🛄 AQ	
PLC 1 [CPU 1212C DC/DC/DC]		Compatible devices in t	arget subnet:			s	iow an compatible de	liac	inostics	AI/AQ	2
Conserved IO tages Supratem constants		Device	Device type	Туре		Address	Target device			Communic	ast
General To tags System constants	and and a			PN/IE		Access address				Technology	ŝ
Ethernet addresses A Ethernet addresses	P 122										
Operating mode Interface	7										E
Advanced options											bra
Hardware identifier	Flash LED										ries
► DI8/DO6											
► AI2							Refre	esh			
High speed counters (HSC) IP protoco											
Pulse generators (PTO/PWM)	Online status information:										
	Scan completed. 0 cor	npatible devices of 0 ac	cessible devices fou	nd.							
Communication load	Select Show all comp	atible devices to reach	i all compatible devic	es							
System and clock memory	Display only problem re	norts									
Web server											
Time of day						6	o online Car	ocel			
User interface languages											
Protection											
Connection resources PROFINET										<	
< III >									~	> Information	
Portal view Derview Portal view	.C_1			_					🗸 Project Project1 opene	d.	
🔊 🥖 😁 🕅 🔊		o 📝 🐺							EN 😨 🖡	16:48	

სურათი 6-68: კომპიუტერთან მოწყობილობის შეერთების პროცესი

Save project 💩 🗶	15 E × 101 (*1)	- 11日日日日 - 11 - 11 - 11 - 11 - 11 - 11	- bill x = 1			rotany integr	PORT
	B Projecti + PLC	1 (ONL 1212C DODODC)				_ # # X	Harmon #15
				Topology view	A Notaurk view	Device view	Options
	# R.C.1	· = 4 1 4 1 14 1 100%	•				
						0	~ Catalog
		55					dearcha ing
		W				110	Filter
						2	• (# CTU
	1000	and a second sec				1.1	Signal boar
		Concession of the local division of the loca					. Batteryboa
							• M Di
		No. of Concession, Name					1 00
		- I ^A					• (m A)
	<	and the second				2 🖸	• 34 AQ
				S Properties	Linfo Diag	nostics	 MARQ Marcal
Device Information	Connection Information	Alarm display					• Technology
No devices with problem	8	e ferningen er einen					
P. P	the Mercele	Ortain	- Help				

სურათი 6-69: კომპიუტერისა და პლკ-ს წარმატებული შეერთება

აქ ღილაკზე go online დაჭერით ვუერთდებით ქსელს (სურათი 6-69:). ამრიგად, ჩვენი მოწყობილობა ახლა უკვე ქსელშია ჩართული და შეიძლება დაიწეროს ჩვენი პირველი პროგრამა.

6.2.5 პროგრამის შექმნა

შევიდეთ პროგრამული ბლოკების საქაღალდეში (იხ. სურათი 6-70:) და გავხსნათ ძირითადი ბლოკი Main Blok (იხ. სურათი 6-71:), რომელიც ციკლურად მუშაობს პროცესორის გაშვების შემდეგ.



სურათი 6-70: პროგრამული ბლოკების საქაღალდეს ფანჯარა

ცალკეული ქვეამოცანების გადაწყვეტილებებისათვის უკეთესია შევქმნათ ჩვენი საკუთარი ბლოკები ამისათვის ვაწვებით Add new blok (იხ. სურათი 6-71:). რადგანაც ამ ჩვენმა ბლოკმა უნდა შეასრულოს ორი რიცხვის გამრავლება, ამიტომ დავარქვათ მას სახელს Mult. აქვე შეგვიძლია ავირჩიოთ დაპროგრამების ენა. ჩვენ შემთხვევაში ჯერჯერობით დავტოვებთ LAD ენას.

Siemens Sima	tic S7-1200 Part2/ # 6 1 # x = 1	PORTA
Constant and	The sum of the star screek - whereas - we built	Watch later Sr
Devices		Options
300	5 NYAAAA BERSES 6 26 6 7 9 4 1 1 1 1	
	Fickbacker	✓ Eavorites
1 STRee	⁹ Ar + + - + + + + →	anaous
A Tester & among		✓ Basic instructions
· ARCHICE LICE ACCORN	Bock SSec Nen hops in Seerp Cycle?	tana d
Cauce configuration	- Contest	• Central
S Online & diagnostics	w Network 1	 Stropt operations
· C Program blocks	O Contest	A California and a california
Add new block		· /2 Concentration
· New (DET)	•	+ 1 the tectors
 E System Particle 	•	+ Calibur spreations
· a fectualographers		+ 😼 Conversion operations
• an Exernal source firs		> Se Popen careal spendare
A THE REPORT OF	· Rebard 2	• 🙀 Word logic operations
1 To mark and here while	Comments	• E Shit and crase
Pa Pagan into		
R lecks		
+ g Looi modules		
• 🚽 Common data		
Alem cleans		
k tectos	×	
Details view		
Same		
E VIDEOS		
		2 Extended instructions
		> Lochestogy
15-40 / 27-47	Strepettes Viete Dispectice	2 de Veuluke

სურათი 6-71: Main Blok-ის ფანჯრა



სურათი 6-72: საკუთარი ბლოკების შექმნა

სურათი 6-72:-დან ჩანს, რომ ჩვენ შეგვიძლია შევქმნათ ოთხი სხვადასხვა სახის ბლოკი: პირველი (OB) - ეს მაორგანიზებელი ბლოკია, რომელიც შეიძლება ითქვას, რომ ძირითადი ბლოკია პროგრამაში, ამ ბლოკით შეიძლება შესრულდეს მაგ. წყვეტების დამუშავება, სასტარტო ფუნქციები და სხვ. მე-2 და მე-3 სახის ბლოკები ფუნქციონალური ბლოკებია (FB, FC), ანუ შეიძლება ითქვას რომ ისინი ჩვენი პროგრამის ქვეპროგრამებია. ბლოკები FB და FC ერთმანეთისაგან განსხვავდება იმით, რომ ბლოკ FB -ს აქვს თავისი მონაცემთა ბლოკი DB და შესაბამისად შეიძლება გამოყენებულ იქნას იმ შემთხვევებში, როდესაც ჩვენ გვაქვს რამოდენიმე ერთნაირი მოიწყობილობა, ერთნაირი მუშაობის ალგორითმებით, მაგრამ სხვადასხვა შესასვლელი მონაცემებით. ასეთ შემთხვევაში ჩვენ შეგვიძლია შევქმნათ სულ ერთი ბლოკი FB და ჩავტვირთოთ მასში სხვადასხვა შესასვლელი მონაცემები. მაგალითად, დაუშვათ რომ ჩვენ გვაქვს დანადგარი რამოდენიმე ტუმბოთი, ამ ტუმბოების მუშაობის ალგორითმი ერთნაირია, მაგრამ სხვადასხვა წნევებია, ტემპერატურები, სიჩქარეები და სხვ.

ჩვენ შევქმნით უბრალო ფუნქციონალურ ბლოკს FC ამ ბლოკზე დაწკაპებით. როგორც ვხედავთ იგი გამოჩნდება ეკრანის მარცხენა მხარეში პროგრამის ხეში (იხ. სურათი 6-73:).



სურათი 6-73: ფუნქციონალური ბლოკის შექმნა

იმისათვის რომ ეს FC ბლოკი გამოძახებულ იქნას ავტომატურად ყოველთვის, როცა კი მთავარი ბლოკი ამუშავდება, საჭიროა გადავიდეთ მთავარი ბლოკის ფანჯარაში ამ ბლოკზე Mine (OB1) დაწკაპუნებით და უბრალოდ გადავიტანოთ FC ბლოკი OB1 ბლოკზე (იხ. სურათი 6-74:). ამით მივიღებთ გამოსახულებას, რომელიც სურათი 6-75:-ზეა ასახული.

M Siemens - Project3						_ # X
Project Edit View Insert Online Opti	ions Tools Window Help				Totally Integr	notremotus here
🕒 🕒 🔄 Save project 📑 📈 🛤 🕞 🗙		📮 💋 Go online 🔊	Go offline 🔥 🚺		rotany integra	PORTAL
Project tree 🛛 🖬 🕯	Project3 + PLC_1 [CPU 12120	AC/DC/Rly] + Progra	m blocks ► Ma	in [081]	_ # = ×	Instruc 🗊 🛙 🕨
Devices						Options 🔡
1900 2	12 14 19 19 16 16 16 16 16 16 16 16 16 16 16 16 16	82:2:58	06.019	19 L L & C II	1	🗆 🛗 💀
	Main				_	> Favorites
 [1 Project3 	Name	Data type	Default value	Comment		A Racio Instant
Add new device	1 - input					• Dasic instruc
Devices & networks	2 - Initial Call	Bool		Initial call of this OB	1	Name
- RLC_1 [CRU 1212C ACIDCRIM]	3 - Remanence	Bool	-	=True, if remanent data are available		General
Device configuration				* 1		Timer operation
S Online & diagnostics						Counter operation
- 🕞 Program blocks	- March Maler - And Statement					Comparator con
Add new block	 Block title: "Nein Program Si 	veeb (cycie).				• 1 Math functions
🖝 Mein (081)	Comment					Move operation
Mult [FC1]	 Network 1: 					Conversion ope
Technology objects	Comment					Program control
External source files	Comment					Word logic open
FLC tags	1					Shift and rotate
PLC data types		_				F
Watch and force tables		Q-113				
Traces						< = > 5
Program info						✓ Extended ins
Device proxy data	 Network 2: 					Name
le Text lists	Comment					Date and tim A
Local modules						String + Chai II
Common data	11					Distributed II
Documentation settings						Interrupts
Languages & resources						Diagnostics
Doline access						🕨 🛄 Pulse 🛛 🔍
Card Reader/USB memory	1					< = >
	1					✓ Technology
	1					Name
	1					Counting

სურათი 6-74: OB1 ბლოკზე FC ბლოკის გადატანა

Siemens - Project3						-
ject Edit View Insert Online O	itions Tools Window Help				Totally Integrated Automatio	on
🔁 🛃 Save project 🚢 🐰 🧾 💽	K 🎝 ± (** 🖬 🖥 🖪 🖬 🖉 🎝	🖌 Go online 🖉 Go o	ittine 🔥 🖪		POR	RTA
	Project3 + PLC_1 [CPU 1212C AC/DC	(Rly] 🕨 Program t	locks + Mair	(OB1)	_ 🖉 🖬 🗙 Instruc 🖓 I	
Devices					Options	
900	k la ka ≈ e la	11 - 21 🖂 😥 (o 🐻 📾	♥ La La & ∞ B	E 0	1 1
	Main				> Favorites	
Project3	Name	lata type	Default value	Comment	V Basic Instr	ner.
Add new device	1 🚭 🕶 Input				A Huma	100
Devices & networks	2 😋 • Initial_Call 8	tool 🔳		Initial call of this OB	E Caparal	
· DLC_1 [CPU 1212C AC/DC/RM]	3 - Remanence 8	lool		=True, if remanent data are available		
Device configuration					h Timer one	eratio
S Online & diagnostics					h 🛃 Counter o	
🕶 🙀 Program blocks	- Marketter Inc	-1.38			Comparet	-
Add new block	 Block bble: Main Program Silveep (C) 	Acid).			 Nath funct 	ction
🖀 Main [081]	Comment				Move oper	erati
Mult [FC1]	 Network 1: 				Conversion	on of
Technology objects	Comment				Program c	cont
External source files	Contraction of the second seco				Word logic	ic op
PLC tags	9401				Shift and r	rota
PLC data types	"Mult"					
Watch and force tables	EN ENO					_
Traces					<	
Program info					 Extended in 	ins.
Device proxy data					Name	
Text lists	 Network 2: 				Date and t	tim
Local modules	Comment				String + C	Chai
Common data					Distributer	ed 1
Cocumentation settings					Interrupts	5
Languages & resources					🕨 🛅 Diagnostic	ics
Doline access					Pulse	
Card Reader/USB memory					< =	>
					✓ Technology	3Y
					Name	

სურათი 6-75: გამრავლების ბლოკის გამოტანა

ახლა ჩვენ უკვე უკვე შეგვიძლია დავიწყოთ ჩვენი პროგრამის შექმნა. ამისათვის პროგრამის ხეში ვაწკაპუნებთ ჩვენს მიერ შექმნილ ფუნქციონალურ ბლოკს Mult [FC1] (იხ. სურათი 6-76:).

M Siemens - Project3		_ # X
Project Edit View Insert Online C	iptions Tools Window Help	Totally Integrated Automation
🕒 🎦 🔜 Save project 🚢 🐰 🖽 🕞	🗙 🌇 🕇 (# ± 🙀 🔂 🗓 🔛 🔛 🖉 🌽 Go online 🖉 Go offline 🍶 🖪 📴 💥 🖃 🛄	PORTAL
Project tree 🛛	Project3 + PLC_1 [CPU 1212C AC/DC/Rby] + Program blocks + Mult [FC1]	💶 🖷 🖬 🗙 Instruc 🗗 🛛 🕨
Devices		Options
1900	👌 🚜 김 씨 원 씨 원 🗖 🗖 🗃 🗃 🗃 🗃 🖬 🕼 😢 📞 생 행 😵 님 님 요 안 100	III III 2
2	Mult	> Favorites
 Troject3 	Name Data type Default value Comment	Y Basic Instruc
Add new device	1 🗨 • Input	A Name
Devices & networks	2 • oldd news	E Canaral
PLC_1 [CPU 1212C AC/DC/Rly]	3 🕘 🕶 Output	V b California operati
Device configuration		M Timer operation
Se Online & diagnostics	·	Counter operation
 Program blocks 	w Black title:	> C Comparator ope
Add new block	slock title:	Math functions
🖀 Main [081]	Comment	Move operation
Ault (FC1)	Network 1:	Conversion ope 8
Technology objects	Company .	Program control
External source files	Comment	Word logic open
PLC tags		Shift and rotate
Eg PLC data types		5
Watch and force tables		10
Traces		< = > 5
Program info		✓ Extended ins
Device proxy data	Network 2:	Name
Text lists	Comment	Date and tim +
Local modules		String + Chai II
Common data		Distributed I
Documentation settings		Interrupts
Languages & resources		Diagnostics
Doline access		🕨 🛅 Pulse 🔍 🔍
Card Reader/USB memory		< = >
		✓ Technology

სურათი 6-76:

მაგალითისთვის ვიღებთ ასეთ ამოცანას: პირველი გადამრთველის გადართვა გადმორთვებს შემთხვევაში, რომელიც მერთებულია პლკ-ს ნოლოვან შესასვლელზე, ხორციელდება ამ გადართვების თვლა და მიღებული შედეგის ასახვა პლკ-ს გამოსასვლელზე. ამ მიზნის მისაღწევად გადამრთველების გამოსახულებიდან, (ამ ხუთიდან)

	⊣⊢	-1/F	-0-	??	↦	-1
--	----	------	-----	----	---	----

(ამ ხუთიდან) <??.♪

გადმოვიტანოთ პირველივე გამოსახულება

ჩვენი ფუნქციონალური ბლოკის პროგრამაში.

აქვე უნდა შევიტანოთ ამ გადამრთველის მნიშვნელობა. ეს მნიშვნელობა შეიძლება იყოს სამი სახის: ასო I ეს შესასვლელებია, ასო Q ეს გამოსასვლელებია და ასო M ეს მეხსიერების უჯრედია. ჩვენთან ეს გადამრთველი შესასვლელია, ამიტომ გავხსნით კითხვის ნიშნებს და ჩავწერთ ასო I-ს. აქვე მივუთითებთ პირველი ბაიტის ნომერს - ნულოვანს და პირველი ბიტის ნომერს ამ ბაიტში - ასევე ნულოვანს. საბოლოოდ მივიღებთ გამოსახულებას სადაც %I0.0 ჩვენს მიერ ჩაწერილი მნიშვნელობაა.



როგორც ვხედავთ კომპილიატორმა მაშინვე ავტომატურად მიანიჭა დასახელება ჩვენს ბაიტს Tag_1, იმისათვის, რომ მივანიჭოთ სხვა დასახელება უნდა შევიდეთ ტეგების საქაღალდეში (იხ. სურათი 6-77:) და გავხსნათ ცვლადების ცხრილი.

Siemens - Project3		_ •
roject Edit View Insert Online O	tions Tools Window Help	Totally Integrated Automation
9 🎦 🛃 Save project 📓 🗶 🐚 🕞	< 💫 🛨 (¹⁴ ± 📪 🐴 🔃 🔛 🔛 🔛 💋 Go online 🖉 Go offline 🗼 🖪 📑 🗙 🚍 📋	PORTAL
Project tree 🛛	Project3 + PLC_1 [CPU 1212C AC/DC/Rly] + Program blocks + Mult [FC1]	_ 🗗 🖬 🗙 Instruc 🗊 🛙 🕨
Devices		Options
1900	[영영하지 # F 문 문 문 문 문 명 # 명 # 문 문 한 6 6 6 6 6 6 6 6 7 9 7	· U I I I I I I I I I I I I I I I I I I
	Mult	> Favorites
▼ □ Project3	Name Data type Default value Comment	Y Basic Instruc
Add new device	1 🗨 = Input	A Name
devices & networks	2 • «Add newo»	I General
PLC_1 [CPU 1212C AC/DC/Rbj]	3 🗨 Vorput	V b Galleria operati
Device configuration		M Timer operation
Se Online & diagnostics		Counter operation
🕶 😹 Program blocks	- March Male	Comparator on
Add new block	• Block ube:	Math functions
🖀 Main [081]	Comment	Move operation
Mult [FC1]	Network 1:	Conversion ope
Technology objects	Command	Program control
External source files	Second Se	Word logic oper
🕶 🌄 FLC tags	30.0	Shift and rotate
a Show all tags	'Tag_1'	
Add new tag table		
📽 Default tag table [27]		¢
PLC data types		 Extended ins
Watch and force tables		Name
Traces	Network 2:	🕨 🛄 Date and tim 🛪
Program info	Comment	String + Chei III
Device proxy data		Distributed II
Text lists		Interrupts
Local modules		Diagnostics
Common data		Pulse v
 Documentation settings 		(III)
 Languages & resources 		✓ Technology
Gild Contractors		Name
Card Reader/USB memory		Counting
		PID Control

სურათი 6-77: ცვლადების ცხრილის ფანჯარა

ეკრანზე მივიღებთ გამოსახულებას (იხ. სურათი 6-78:) სადაც შევდივართ ტეგების ველში და Tag_1-ის ნაცვლად ჩავწერთ ნებისმიერ სხვა დასახელებას, მაგ. X. მივიღებთ ახალ ცვლადს X, აქვე დავინახავთ მის ტიპს - ანუ ორობით სიდიდეს (Bool), აქვეა მისი მისამართიც %I0.0.

M Siemens - Project3								_ 0
Project Edit View Insert Online O	rptions Tools Vändew Help ★ 51 (*1: 14 3) 10 10 1	🛔 📮 🍠 Go online y	a Go attine 🔥			10	Totally Integr	ated Automation PORTAL
Project bee D	Project3 > PLC_1 [CPU 1212	CAODORIA] + PLC	tags + Defau	t tag table [27]		_#=×	Tasks 💕 🗉 🕨
Devices				Q	Tags	B User constants	System constants	Options
1900	2 9 0 3 ° 2 2						3	
-	Default tag table							✓ Find and repl
• Troject3	Name	Data type	Address	Retain	Visibi	Acces. Comment		1
Add new device	1 -CD Teg_1	Bool	10.0			8		Find:
devices & networks	2 oldd neucs				9			3
								Thinkie words a
Device configuration	1000							Chamberra
Se Online & diagnostics	1000							Elverch cese
• g Program blocks	1000							E find in substru
Add new block								Find in hidden
Main (081)	1000							Clube whiteards
Mult (FC1)								Concentration
Technology objects	and a second sec							Elote repairr ex
 Beternal source files 	1000							O Whole docum
• 🚂 PLC tags	1000							Charman contracts
Show all tags	1000							Corrent compile
Add new tag table	and the second s							 Selection
S Default tag table [27]	and the second se							Down
 PLC data types 	and the second s							
 Watch and force tables 	and the second se							Oup
Traces								Fyn

სურათი 6-78: ტეგებზე სხვა სახელის მინიჭების ფანჯარა

აქ არის შესაძლებლობა წინასწარ ჩავწეროთ ყველა შემდეგში გამოსაყენებელი ცვლადი. ჩვენს შემთხვევაში ცვლადს შევქმნით Add New-ზე დაჭერით, რომელიც შეესაბამება გადართვების თვლას, დავარქმევთ მას XCount, ავირჩევთ მის ტიპს, ანუ კონკრეტულად, ჩამოშლადი სიიდან "მთელი" – int, მისამართის ჩასაწერად კომპილიატორი მაშინვე გააწითლებს მას, რადგან ამ ტიპის ცვლადისთვის არჩეულია არასწორი მისამართი. ჩვენ ავირჩევთ სწორ მისამართს M0 და OK.

ასევე ჩვენ უნდა შევქმნათ გამოსასვლელების ბაიტი. თუმც არის შესაძლებლობა მივაკითხოთ გამოსასვლელების თვითოეულ ბიტს ცალცალკე, მაგრამ ამ შემთხვევაში გამოვიყენებთ ინფორმაციას ერთბაშად მთელი ბაიტისთვის. ამისათვის დავამატებთ ახალ ველს ჩასაწერად და დავარქმევთ ახალ ცვლადს სახელწოდებას Out. ამ ცვლადის ტიპი იქნება ბაიტი, ხოლო მისამართი Q0, ანუ ნულოვანი ბაიტი (იხ. სურათი 6-79:).



სურათი 6-79: გამოსასვლელი ბაიტის შექმნა



სურათი 6-80: გადამრთველი X-ის დაპროგრამება

გავაგრძელოთ პროგრამის შექმნა (იხ. სურათი 6-80:). ჩვენ გვაქვს გადამრთველი X რომელიც მიერთებულია, რომელზედაც ხდება გადართვა/გადმორთვის მოქმედებების შესრულება თვლის ფუნქციის შესასრულებლად და რომლის გამოსასვლელიც მიერთებულია პლკ-ს ნულოვან შესასვლელზე, საიდანაც მოხვდება პროცესორში, რომელმაც უნდა განახორციელოს გადართვა/გადმორთვის თვლა.



სურათი 6-81: მთვლელების საქაღალდეში გადასვლა

ახლა ჩვენი მიზანია დავთვალოთ ამ გადართვების რაოდენობა. ამისათვის გადავდივართ მთვლელების საქაღალდეში (იხ. სურათი 6-81:) მარჯვნივ და გადმოვაცოცებთ იქიდან მთვლელის ბლოკს CTU-ს ჩვენს პროგრამაში და დავარქმევთ მათ რაიმე სახელს, მაგალითად XCounter. მივიღებთ გამოსახულებას, რომელიც ასახულია სურათზე 6-82:.



სურათი 6-82: მთვლელის გადმოტანა პროგრამის სამუშაო ფანჯარაში

მთვლელის RV შესასვლელზე უნდა ჩაიწეროს ის რიცხვი საიდანაც უნდა დაიწყოს მთვლელმა ათვლა, ჩვენ შემთხვევაში ჩავწეროთ 0 და CV გამოსასვლელზე ჩავწეროთ ცვლადი,

სადაც უნდა შევინახოთ ჩვენი მონაცემები. ჩვენ შემთხვევაში ვწერთ ცვლადს XCount. (იხ. სურათი 6-83:)



სურათი 6-83: მთვლელის დაპროგრამება

ამის შემდეგ, გადამრთველის ყოველ გადართვაზე, ჩვენ უნდა გავიტანოთ გამოსასვლელზე ის რიცხვი რომელიც მეხსიერებაში დაგროვდა. ამისათვის უნდა შევქმნათ განშტოება (სურათი 6-84:), (სურათი 6-85:).



სურათი 6-85: მთვლელის დაპროგრამება

შემდეგ ავიღოთ ბრძანება MOVE (იხ. სურათი 6-86:), ორჯერ დავაწკაპუნოთ მასზედ და მისი შესაბამისი ბლოკის გამომყვანებზე ჩავწეროთ ცვლადები XCount და Out. საბოლოოდ პროგრამა მიიღებს სახეს (იხ. სურათი 6-87:) და ამით პროგრამის შექმნა დამთავრებული იქნება. პროგრამა მზადაა კონტროლერში ჩატვირთვისათვის.



სურათი 6-86: MOVE ბრძანების გადმოტანა



სურათი 6-87: MOVE ბრძანების დაპროგრამება

6.2.6 პროგრამის კომპილიაცია და ჩატვირთვა კონტროლერში

ახლა გადავიდეთ ამ პროგრამის კომპილაციზე, რისთვისაც შევდივართ ჩვენ პლკ-ზე PLC_1 [CPU 1212AC/DC/Rly (იხ. სურათი 6-88:), ვაწკაპუნებთ მასზედ და ჩამოშლილ სიაში ვირჩევთ Compile -→ Software (Only Changes) (იხ. სურათი 6-89:) ეკრანზე გამოვა კომპილიაციის პროგრესის გამოსახულება (იხ. სურათი 6-90:).



სურათი 6-88: პროგრამის კომპილიაცია

			"XCou	inter"								
	% i0.0 "X"		C	U t								
	1	fe	CU alse — R 0 — PV	Q CV	%MW — "XCor	D unt"		-				
		%N *XCou	EN		%QB(~			
	_		_	_	_	100%	•					
						100%	▼ Second Second Secon	ties	🛄 Info	R D	iagnostics	18,
Gener	ral Cros	ss-references	Compile	Syr	ntax	100%	▼ Second Second Secon	ties	1. Info	U.D	iagnostics	Pe
Gener	ral Cros	ss-references I (errors: 0; warning	Compile	Syr	ntax	100%	Proper	ties	1 Info	<u>v</u> D	iagnostics	
Gener Compili ! Path	ral Cros	ss-references I (errors: 0; warning	Compile gs: 0) Description	Syı	ntax	100%	Proper	ties Go to	i Info	Errors	iagnostics Warnings	Time
Gener Compili ! Path	ral Cros ng completed PLC_1	ss-references I (errors: 0; warning	Compile gs: 0) Description	Syı	ntax	100%	▼ ■ Proper	ties Go to	2 Info	Errors 0	Warnings 0	Time 2:29:50 PM
Gener Compili ! Path ? T	ral Cros ng completed PLC_1 Program bl	ss-references I (errors: 0; warning locks	Compile gs: 0) Description	Syı	ntax	100%	▼ ■ Proper	Go to	i Info	Errors 0 0	Warnings 0 0	Time 2:29:50 PM 2:29:50 PM
Gener Compili Path	ral Cros ng completed PLC_1 ← Program bl XCount	is-references I (errors: 0; warning locks er (DB1)	Compile gs: 0) Description Block was suc	Syn	ntax / compile	100%	▼ Second Second Secon	Go to	2	Errors 0 0	iagnostics Warnings 0 0 0	Time 2:29:50 PM 2:29:50 PM 2:29:50 PM
Gener Compili ! Path ? • f	ral Cros ng completed PLC_1 ▼ Program bl XCount Gamr (f	ss-references I (errors: 0; warning locks er (DB1) FC1)	Compile gs: 0) Description Block was suc Block was suc	Syn	ntax / compil-	100% 	Proper	Go to	2 ?	Errors 0 0 0 0	Warnings 0 0 0 0 0	Time 2:29:50 PM 2:29:50 PM 2:29:50 PM 2:29:51 PM
Gener Compili ! Path ? • f	ral Cros ng completed PLC_1 • Program bl XCount Gamr (f Main (C	ss-references I (errors: 0; warning locks er (DB1) -C1) JB1)	Compile gs: 0) Description Block was suc Block was suc Block was suc	Syn cessfully cessfully cessfully	ntax / compil- / compil- / compil-	100% 	Proper	Go to	2 ?	Errors 0 0 0 0 0	Warnings 0 0 0 0 0 0 0	Time 2:29:50 PM 2:29:50 PM 2:29:50 PM 2:29:51 PM 2:30:00 PM

სურათი 6-91: კომპილიაციის პროცესის შედეგი

სურათი 6-90: პროგრამის კომპილიაციის პროცესი

%DB1

=



სურათი 6-89: პროგრამის კომპილიაციის სტარტი



კომპილაციის პროცესის დამთავრების შემდეგ ეკრანზე გამოჩნდება სურათი (იხ. სურათი 6-81:) საიდანაც ჩანს რომ კომპილაციამ ჩაიარა წარმატებით, არ არის შეცდომები და გაფრთხილებები. ამის შესაბამისად, ჩვენ უკვე შეგვიძლია ჩავტვირთოთ ჩვენს მიერ შექმნილი პროგრამა კონტროლერში, რისთვისაც შევდივართ ისევ PLC_1 [CPU1212AC/DC/Rly] ვაწკაპუნებთ მასზედ და ჩამოშლილი სიიდან ვირჩევთ Download to device \rightarrow Software (only change) (იხ. სურათი 6-82).



სურათი 6-92: პროექტის ჩატვირთვის დაწყება კონტროლერში

Load pro	eview				×
3	ompili	ing before downloadi	ng to device		
Status	1	Target	Message	Action	
↓ <mark>↓</mark>	8	▼ PLC_1	Loading will not be performed because preconditions are not m	et	
	4	Compile Compile	Compile before downloading to device	-	
	8	• E:	Compiling configuration Checking consistency for module S7-1200 station_1.		
	۲				
			Car	cel	
<			1111		>
					Refresh
			Finish	Load	Cancel

სურათი 6-93: პროექტის ჩატვირთვის პროცესი კონტროლერში

ეკრანზე გამოჩნდება ჩატვირთვის პროცესი (სურათი 6-90:) რომლის დამთავრების შემდეგაც მივიღებთ სურათს 6-94:, ვაწკაპებთ ღილაკს Refresh, ეკრანზე გამოჩნდება სურათი (იხ. სურათი 6-95:).

2 c	heck	before loading		
		-	Manage	A refer
atus 41	8	PLC_1	Loading will not be performed because preconditions are not met	Action
î				
	8	 Expand download 	It is not possible to download the complete PLC to the device.	
	8		You need to download the complete PLC to the device because there are dependencies between PLC program and hardware configuration. This is not possible as long as there is an online connection to the PLC. Go offline and download the PLC to the device again.	
	0	Device configurati	. Delete and replace system data in target	Download to device
	0	Software	Download software to device	Consistent download
	0	Text libraries	Download all alarm texts and text list texts	Consistent download
			111	:
				Refresh

სურათი 6-94: ჩატვირთვის პროცესის დამთავრება

itatus	E.	Target	Message	Action
+[]	0	· PLC_1	Ready for loading.	
	0	Device configurati.	Delete and replace system data in target	Download to device
	0	Software	Download software to device	Consistent download
	0	Text libraries	Download all alarm texts and text list texts	Consistent download
¢			10	

სურათი 6-95: პროცედურა Refresh

ამის შემდეგ ვაწკაპებთ ღილაკს Load, ეკრანი მიიღებს სახეს (იხ. სურათი 6-96:), აქ აუცილებლად უნდა მოვნიშნოთ ალამი Start all, რითაც მივიღებთ სურათს (სურათი 6-97:), საბოლოოდ კი ვაჭერთ ღილაკს Finish და გამოსახულება გადადის საწყის მდგომარეობაში, რაც იმას ნიშნავს, რომ კონტროლერის ჩატვირთვა დამთავრებულია და შეგვიძლია ჩავატაროთ ტესტირების პროცედურა.

Status	1	Target	Message	Action	
4	9	▼ PLC_1	Downloading to device completed without error.		
	A	Start modules	Start modules after downloading to device.	Start all	
<			11		

სურათი 6-96: პროგრამის ჩატვირთვის საფეხური Load

Status	1	Target	Message	Action
1	9	▼ PLC_1	Downloading to device completed without error.	
	4	Start modules	Start modules after downloading to device.	Start all
<			ш	

სურათი 6-97: პროგრამის ჩატვირთვის დამთავრების ბოლო საფეხური Finish

6.2.7 კონტროლერის ტესტირება

როგორც სურათიდან ვხედავთ (სურათი 6-98:) პლკ-ზე ზევით მიერთებულია ორ მწკრივად განლაგებული რვა გადამრთველი, რომელიც აღებულია S7-1200 კონტროლერის Starter Kit-იდან. თვითონ კონტროლერის კორპუსის წინა მხარეს მარჯვენა ნაწილში განლაგებულია ორი რვათანრიგიანი შუქდიოდების მწკრივი. ზედა მწკრივი მიერთებულია კონტროლერის შესასვლელებთან ხოლო ქვედა მწკრივი-გამოსასვლელებთან.

ზემოთნახსენები რვა გადამრთველი შეერთებულია შესასვლელებთან, ამიტომ ნებისმიერი მათგანის გადართვა იწვევს შესაბამისი შუქდიოდის ანთებას. შუქდიოდების ნუმერაცია იწყება 0-დან და მარჯვნიდან და მთავრდება 7-ით როგორც ზედა ასევე ქვედა მწკრივში.



სურათი 6-98: S7-1200 ტესტირებისათვის გამზადებული

ჩვენს შემთხვევაში, რადგან სურათზე მოყვანილი გადამრთველებიდან პირველია გადართული, ამიტომ მისი შესასბამისი შუქდიოდი ზედა მწკრივში, ანუ 0-ვანი შუქდიოდი ანთია. მოვახდინოთ ამ პირველი გადამრთველის გადართვა გადმორთვა ოთხჯერ. ჩვენს მიერ დამუშავებულ პროგრამაში არსებული მთვლელი ასრულებს ამ გადართვების რაოდენობის თვლას და მიღებული რეზულტატის ანუ ოთხის ასახვას შუქდიოდების მეორე მწკრივში, მაგრამ ეს ასახვა ხდება არა ათობით, არამედ ორობით კოდში, რომლის თანრიგიანობა იწყება არა მარჯვნიდან, არამედ მარცხნიდან და ამიტომაა, რომ ანთია მე-3 შუქდიოდი (00100000-ორობით კოდში, რომელიც ათობით კოდში უდრის 4-ს).

6.2.8 პროგრამის მუშაობის ვიზუალიზაცია

ჩვენ შეგვიძლია აგრეთვე პროგრამული ფანჯრიდან დავაკვირდეთ პროგრამის მუშაობას. ამისათვის მენიუს სტრიქონში უნდა დავაჭიროთ ღილაკს სათვალეებით, ეკრანზე მივიღებთ გამოსახულებას (სურათი 6-96:) რაც საშუალებას მოგვცემს რეალურ დროში ვნახოთ თუ როგორ მუშაობს ჩვენი პროგრამა. ამისათვის ჩავწეროთ მთვლელში რიცხვი 8.

💆 O O	💼 🖧 전 후 한 🖦 🗄 🗮 🗩 웹 ± 월 발 월 😢 다 다 🕮 🕸 한 것 👘	
	Block interface	> Favorites
👻 🛅 Project1	No condition defined.	Basic instructions
🗳 Add new device		Name Description
Devices & networks		General
▼ 1 PLC_1 [CPU 1212C A	Slock title:	Bit logic operations
Device configur	Comment	Imer operations
😼 Online & diagn	comment	Counter operations
🕶 🛃 Program blocks	Network 1:	Comparator operations
💕 Add new block	Comment	the second
Hain [OB1]		Move operations
📲 Gamr [FC1]		Conversion operations
System blocks	● %DB1	b d Program control operati
Technology objects	"XCounter"	Mord logic operations
External source fil	%0.0 CTU	Shift and rotate
PLC tags	● "X" Int	
PLC data types	cu o	
Watch and force t	false — R	< III
Traces	0 — PV 94400	✓ Extended instructions
Program info	cv — "XCount"	Name Description
Device proxy data		Date and time-of-day
Text lists	MOVE	String + Char
Local modules		Distributed I/O
🕨 🙀 Common data		Interrupts
Documentation setti	8 16#08 %/080	Diagnostics
Languages & resour	"XCount" IN * OUT	Pulse
Online access		

სურათი 6-99: პროგრამის მუშაობის ასახვა რეალურ დროში

გადამრთველი გადმორთულია, მთვლელში ჩაწერილია 8, ეკრანზე გამოჩნდება თუ როგორ იცვლება სიგნალების ასახვა გადამრთველის კიდევ ერთჯერ გადართვით, ხოლო მთვლელის მდგომარეობა უტოლდება ცხრას (სურათი: 6-100:).



სურათი 6-100: მთვლელის მუშაობის ვიზუალიზაცია

თუ კიდევ გამოვრთავთ და გადავრთავთ გადამრთველს, მაშინ მთვლელის მდგომარეობა გახდება 10 (იხ. სურათი 6-101:), შემდეგ 11, 12 და ა. შ. შ. როგორც ვხედავთ ეს ფუნქცია საკმაოდ მოსახერხებელია. ჩვენ შეგვიძლია ვუყუროთ პროგრამის მუშაობას online რეჯიმში. ამის შემდეგ თუ კი დავაჭერთ იგივე სათვალის ნიშანს, მაშინ დავაბრუნებთ პროგრამის ჩვეულებრივ მდგომარეობაში.

6.2.9 პროგრამის ბიჯობრივი გაწყობა

ახლა ვნახოთ თუ რითი განსხვავდება ერთმანეთისაგან დაპროგრამირების ორი ენა LAD და FBD ერთმანეთისაგან. როგორც აქამდე წარმოდგენილი ფიგურებიდან ჩანს LAD ენა უფრო ჰგავს ელექტრულ სქემას (იხ. სურათი 6-98:), თავისი გამომავალი მოწყობილობებით,

გადამრთველებით, რომლებიც შეიძლება შეერთებულ იქნას მიმდევრობით ანდა პარარელურად და სხვ.



სურათი 6-101: პროგრამა LAD ენაზე

თუ გადავრთავთ FBD ენაზე (სურათი 6-102:), მაშინ მივიღებთ გამოსახულებას, რომელიც ასახულია სურათზე 6-103:. აქ ვხედავთ, რომ იგი უფრო გავს ელექტრონულ სქემას სადაც გამოყენებულია ლოგიკური ფუნქციების შესაბამისი ბლოკები "და", "ან", "არა", ტრიგერები, მთვლელები და სხვ. ხოლო დაპროგრამება მსგავსია ლოგიკური სქემების ხაზვისა. კითხვაზე, თუ რომელი ენა უფრო მოსახერხებელია, შეგვიძლია ვუპასუხოთ, რომ ეს დამოკიდებულია თვითონ პროგრამისტზე, ანუ რომელი ენა უფრო ეხერხება მას გამოსაყენებლად. ჩვენის მხრივ დავამატებდით, რომ ორივე ვარიანტი საკმაოდ მარტივია და მათში დაპროგრამება დიდ ძალისხმევას არ საჭიროებს.



სურათი 6-102: დაპროგრამების ენის შერჩევა



სურათი 6-103: დაპროგრამება FBD ენაზე

გავაგრძელოთ ახლა ჩვენი პროგრამის განვითარება და გავიხსენოთ, რომ ჩვენ ვწერთ პროგრამას რომლის ძალითაც საბოლოო ჯამში უნდა მოხდეს ორი რიცხვის გამრავლება და შედეგის ასახვა გამოსასვლელ შუქდიოდებზე. ამისათვის, დასაწყისში საჭიროა შევქმნათ კიდევ ერთი მთვლელი Y ცვლადისათვის, რისთვისაც დაგვჭირდება ტეგების შემოტანა ამ მთვლელისათვისაც (იხ. სურათი 6-104:).

Ma Siemens -	Project5											_ • >
Project Edit	View Insert Online Optio	ns Tools Wind	low Help		Go onlir	ne 🔊 Go	offline 🛵 🖪 🖪 🗶 🖃 🛙				Totally Integrated Automation POR	n TAL
▶ Project5 ▶	PLC_1 [CPU 1212C AC/DC/R	lly] > PLC tags	Default tag	table [33]			_		_ # # ×	Tasks 📑 🛙	
							📶 Ta	as 🗉	User constants	. System constants	Ontions	÷
	00 <u>ñ</u> y						u .u	9º L				
Default	tag table									-4	M Find and replace	
Nar	me	Data type	Address	Retain	Visibl	Acces	Comment				• This and replace	
1 📲 X		Bool	%I0.0								Find:	
🖥 2 📲 XCr	ount	Int	%1/11/0									
🖣 3 🕢 Out	t	Byte	%QB0								Whole words only	
4 🕣 Y		Bool	%IO.1								Match case	, a
5 🔁 YCa	ount	Int	%MW2									
6 📲 Z		Int	%M/4								Find in substructures	
7 🔂 Res	setM	Bool 🔳	%M20.0	<u> </u>							Find in hidden texts	
8 <ad< td=""><td>dd new></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Use wildcards</td><td></td></ad<>	dd new>										Use wildcards	
											Use regular expressions	
											O Whole document	
											 From current position 	
											Selection	
											O Davia	
											O u	
											Up	
											Find	
											Poplace with:	
											Replace with	
											Replace Replace all	
											✓ Languages & resources	
											n divine la seconda	^
											Editing language:	
											English (United States)	
											Reference language:	=
											English (United States)	
							🔍 Prop	perties	🚺 Info 🚺 🗓	Diagnostics		
Device in	formation Connection	information	Alarm displa	v								~
Portal v	view 🔛 Overview	h PLC_1	🕮 Main		🔹 Gam	r	🎽 Default tag t	_		(The programming language of the sele	
? (🗧 📋 🔣 🛛	0 🔣			TIA V13						EN 🕐 🛱 🔺 🕪 隆 14: 26.12	:51 2.2018

სურათი 6-104: ტეგების შექმნა Y მთვლელისათვის

ამის შემდეგ ჩვენ დაგვჭირდება შევქმნათ ცვლადები Y, YCounter და Z გამრავლების რეზულტატისათვის, აგრეთვე Reset ცვლადი მთვლელების ნულზე დაყენებისა და რეზულტატის განულებისათვის, ეს მაშინ როცა გამრავლება უკვე დასრულებულია, შედეგი დაფიქსირებულია და პროგრამა უნდა მოემზადოს ახალი ოპერაციის ჩასატარებლად. ამის მისაღწევად გადავდივართ დაპროგრამებების გაგრმელებაზე. ჯერ ვაკოპირებთ პირველ

შტოში არსებულ პროგრამას და გადაგვაქვს იგი მეორე შტოში (სურათი 6-105:), შემდეგ, მეორე შტოში კოპირებულ X -ებს ვცვლი Y-ებით.



სურათი 6-105: Y ცვლადის შექმნა

აქ სასურველია, ასეთ შემთხვევებში დავამატოთ ასევე, გამორიცხვები (სურათი 6-106:) ანუ ჩვენ უნდა შევქმნათ პროგრამა ისე, რომ პირველი მთვლელი მუშაობდეს მაშინ, როცა მუშაობს პირველი გადამრთველი და არ მუშაობს მეორე გადამრთველი, ხოლო მეორე მთვლელი მუშაობდეს მაშინ, როდესაც მუშაობს მეორე გადამრთველი და არ მუშაობს პირველი გადამრთველი. ეს მიიღწევა X და Y ცვლადების შესაბამისი გადამრთველების სათანადო შეერთებებით.



სურათი 6-106: გამორიცხვების დამატება

ახლა დავწეროთ კოდი გამრავლებისათვის (იხ. სურათი 6-107:). გამრავლება უნდა მოხდეს მაშინ, როცა ჩართული იქნება ერთდროულად როგორც X ასევე Y გადამრთველები.

ამიტომ ეს გადამრთველები უნდა შევაერთოთ მიმდევრობით, შემდეგ მათემატიკური ფუნქციების საქაღალდედან გადმოვიტანოთ გამრავლების ბლოკი MUL და მივუერთოთ მიმდევრობით შეერთებულ გადამრთველებს.



სურათი 6-107: გამრავლების დაპროგრამება

აქ შესასვლელებზე უნდა ჩავუწეროთ გასამრავლებელი რიცხვები XCount და YCount ხოლო გამოსასვლელზე Z, რომლითაც აღნიშნულია გამრავლების რეზულტატი (ნამრავლი). გარდა ამისა, ჩვენ უნდა გამოვიყვანოთ ჩვენი რეზულტატი - ბაიტი პლკ-ს გამოსასვლელზე მისი შუქდიოდებზე ასახვისათვის. ამიტომ ვიყენებთ ჩვენთვის უკვე ნაცნობ ბლოკს MOVE და ვაერთებთ მას გამრავლების ბლოკის გამოსასვლელზე. მის შესასვლელზე კი ჩავწერთ ცვლადს Z, ხოლო გამოსასვლელზე ცვლადს Out (იხ. სურათი 6-108:).



სურათი 6-108: ტრიგერის დამატება პროგრამაში

აქ უკვე სასურველია გავაკეთოთ ასევე რეზულტატების ჩამოყრა (ნულზე დაყენება) მას შემდეგ, რაც გამრავლება უკვე მოხდა რეზულტატი კი უკვე ასახულია პლკ-ს გამოსასვლელზე, რადგან ჩვენთვის უკვე იგი ცნობილი იქნება და ნულზე ჩამოყრით შესაძლებელი გახდება სხვა რიცხვებზე მოქმედებები. შესაბამისად ამისა, ეს უნდა მოხდეს მას შემდეგ, როდესაც გამოვრთავთ ორთავე გადამრთველს X, Y. ამიტომ, დაგვჭირდება გამოვიყენოთ ტრიგერი, რომლის ამუშავებაც მოხდება უარყოფითი სიგნალით. ესაა N_TRIG ტრიგერი საქაღალდედან Bit Logic Operation (იხ. სურათი 6-108:).

ამის მისაღწევად ვასრულებთ განშტოებას მიმდევრობით შეერთებული გადამრთველებიდან, მივუერთებთ მას ჩვენს მიერ არჩეულ ტრიგერს. ამ ტრიგერს თავის მხრივ ჭირდება მეხსიერების უჯრედის ბიტი. მივცეთ მას M20.1 ბიტი, ტრიგერის გამოსასვლელზე იმავე საქაღალდედან დავსვათ ჩამოყრის ბიტი Reset. ჩვენ უნდა მივაერთოთ აგრეთვე ჩვენს მთვლელებს ეს Reset ბიტის გამოსასვლელი, ამიტომ მათს R შესასვლელებზე შევცვალოთ წარწერები და ჩავწეროთ Reset (იხ. სურათი 6-109:). სურათზე ნაჩვენებია ეს პროცედურა მხოლოდ YCounter-სთვის. XCounter-სთვის უნდა მოვიქცეთ ანალოგიურად.



სურათი 6-109: Reset სიგნალის მიერთება YCounter-ის R შესასვლელზე

იმისათვის, რომ ვნახოთ, თუ როგორ არიან განლაგებული ჩვენს მიერ გამოყენებული ბიტები და ბაიტები პლკ-ში შეგვიძლია გავხსნათ Program Info მიღებულ გამოსახულებაში შევიდეთ Assignment List –ში და მივიღოთ გამოსახულება (სურათი 6-110:). სადაც ვხედავთ ჩვენს მიერ დაკავებული მეხსიერების უჯრედებს, ვხედავთ მე-20 უჯრედს, სადაც გამოყენებული გვაქვს ორი ბიტი, ნულოვანი და პირველი, ვხედავთ სამ სიტყვას: MB0, MB1; MB2, MB3; MB4, MB5, ასევე გამოსასვლელების ბაიტს QB0 და შესასვლელებს IB0 და IB1,

	Project tree 🛛 🔳 🖣	Project1 > PLC_1 [CPU 1212C AC/DC/Rly] > Pro	gram info		_ = = ×
	Devices		E Call structure	Dependency structure	nent list 🔟 Resources
	🖻 O O 🖻	🔁 😓 🛨 🔜 no filter> 💌 🍸 📋 🗓 🕱			
B		Assignment list of PLC_1			
iii	🕶 🛅 Project1	Input, Output	å=	Bit memory	
La l	🚔 Add new device	Address 7 6 5 4 3 2 1 0 B W DWORD	LWORD	Address 7 6 5 4 3 2 1 0 B W	
60	📩 Devices & net	IRO A		MBO	
ā	▼ 1 PLC_1 [CPU 121	1864		MB1	
E	Device config	1865		MB2	
	😓 Online & di	1866		MB3	
	🕨 🔜 Program bl	IB67		MB4	
	Technology o	IB1000		MB5	
	External sour	IB1001		MB20 • •	
	PLC tags	IB1002			
	Show all	IB1003			
	Add new t	IB1004			
	Default t	IB1005			
		IB1006			
	Piccoata types	IB1007			
	• Ligg Watch and t	IB1008			
	Traces	IB1009			
	Program info	181010			
	Device proxy	181011			
	Text lists	181012			
	Local modules	IB1013			
	🕨 🙀 Common data	IB1015			
	Documentation	IB1016			
	🕨 📷 Languages & r	IB1017			
	Online access	IB1018			
	Card Reader/USB m	IB1019			
	_	IB1020			
		IB1021			
		IB1022			
		IB1023			
		QBO			
		QB1000			

სურათი 6-110: ჩვენს მიერ გამოყენებული ბიტებისა და ბაიტების განლაგება

ამის შემდეგ უნდა მოვახდინოთ ჩვენი პროექტის კომპილიაცია და შევამოწმოთ, ხომ არ არის შენიშვნები და შეცდომები მასში, შემდეგ ვახდენთ პროგრამის გადატვირთვას

კონტროლერში, რის შემდეგაც ჩვენმა პროგრამის ეკრანმა უნდა მიიღოს საბოლოო სახე სურათი 6-111:).



სურათი 6-111: პროგრამის კომპილიაცია და კონტროლერში ჩატვირთვა

6.2.10 პროგრამის ვიზუალური ტესტირება

პროგრამის ჩატვირთვის შემდეგ ჩვენ შეგიძლია გადავიდეთ პროგრამის ტესტირებაზე. პირველი გადამრთველის გამოყენებით ჩავწეროთ პირველ მთვლელში რიცხვი 4 (001000) (იხ. სურათი 6-112:). შემდეგ, მე-2 გადამრთველის გამოყენებით ჩავწეროთ მე-2 მთვლელში რიცხვი 5 (10100000) (სურათი 6-113:). ამის შემდეგ გადავრთოთ ორთავე გადამრთველი და პლკ-მ უნდა შეასრულოს ამ ორი რიცხვის გამრავლება (იხ. სურათი 6-114:) და მივიღოთ რიცხვი 20 (00101000).



სურათი 6-112: პირველ მთვლელში რიცხვი 4-ის (00100000) ჩაწერა



სურათი 6-113: მეორე მთვლელში რიცხვი 5-ის (10100000)-ის ჩაწერა



სურათი 6-114: ორი რიცხვის გამრავლება და რიცხვი 20-ის (00101000) მიღება

თუ კი ორთავე გადამრთველს გადმოვრთავთ საწინააღმდეგო მხარეს მაშინ რეზულტატი უნდა განულდეს და მივიღება შედეგი 0 (00000000).

6.2.11 პროგრამის მუშაობა ონლაინ რეჟიმში

ახლა ვნახოთ თუ როგორ გამოიყურება ეს ყველაფერი ონლაინ რეჟიმში. პირველ რიგში შეგვყავს რიცხვი 5 (იხ. სურათი 6-115:). ეს რეჟიმი მიიღწევა ნიშნების მენიუში სათვალის ნიშანზე დაჭერით.



სურათი 6-115: ონლაინ რეჟიმში შემოწმება 5-ის შეყვანა

სურათზე ჩანს, რომ მთვლელმა დაითვალა პირველი გადამრთველის ხუთჯერ გადართვა გადმორთვა. შემდეგ შევიყვანთ რიცხვს 4-ს მეორე მთვლელისათვის (იხ. სურათი 6-116:).



სურათი 6-116: ონლაინ რეჟიმში შემოწმება 4-ის შეყვანა

ამ რიცხვებს ჩვენ დავინახავთ გამრავლების ბლოკის შესასვლელებზე (იხ. სურათი 6-117:)



სურათი 6-117: რიცხვების 5 და 4-ის ასახვა ეკრანზე

ახლა კი ჩვენ გვჭირდება გადავამრავლოთ ეს რიცხვები პირველი გადამრთველის გადართვით (სურათი 6-118:). გამრავლების ბლოკის გამოსასვლელზე ვიღებთ შედეგს 20.



სურათი 6-118: რეზულტატის მიღება

ამ გადამრთველების გადმორთვით კი Reset გამოსასვლელზე მოხდება სიგნალის ფორმირება, რაც გამოიწვევს ჩვენი მთვლელების განულებას, გამრავლების ბლოკის შესასვლელებზე გაჩნდება ნულები, ხოლო შემდეგ მათი გამრავლების შედეგად ამ ბლოკის გამოსასვლელზეც გამრავლების შედეგი გაუტოლდება ნულს (სურათი 6-119:).



სურათი 6-119: რეზულტატის განულება

6.4. გამოყენებული ლიტერატურა:

- ჯ. გრიგალაშვილი. ტექნოლოგიური პროცესების მართვის ავტომატიზირებული სისტემების შესახებ, ტომი 2, პროგრამირებადი ლოგიკური კონტროლერები //დამხმარე სახელმძღვანელო, ბიბლიოთეკა სტუ, CD – 3156 // 2016 წ. 231 გვ.
- 2. Руководство по программированию S7-1200/S7 ... Siemens https://assets.new.siemens.com/siemens/assets/api/uuid:6f74436a53942bf8f1c7b6ed34d2ae72e92ed8a 8/programming-guideline-v14-rus.pdf
- 3. Simatik S7-1200 Школа http://plc4good.org.ua/view_post.php?id=133
- 4. TIA Portal Proffesional V14-ის ტრაიალ ვერსიის გადმოწერა და დაყენება https://www.youtube.com/watch?v=yIr_gEL108Y