საქართველოს ტექნიკური უნივერსიტეტი

ჯ. გრიგალაშვილი

სასმელებისა და პროდუქტების წარმოების ტექნოლოგიური პროცესების ავტომატიზაციის შესახებ

ტომი 4

(მეთოდური მითითებები ლაბორატორიულ-პრაქტიკული სამუშაოების შესასრულებლად)

რეგისტრირებულია სტუ-ს
სარედაქციო-საგამომცემლო
საბჭოს მიერ

თბილისი 2015

უაკ 62-52 (075.8); 681.5 (075.32)

მეთოდური მითითებები დამუშავებულია ”მიკროპროცესორები და სამრეწველი მიკროკონტროლერები”–ს კურსის სამუშაო პროგრამის საფუძველზე, რომელიც თვალისწინებს Simatic S7 პროგრამირებადი ლოგიკური კონტროლერის შასწავლას. მეთოდური მითითებები უზრუნველყოფს თეორიული ცოდნის მასალის ათვისებას, Simatic S7–300 პროგრამირებად კონტროლერებზე აგებული მართვის ავტომატიზირებული სისტემების ტექნიკური მომსახურებისა და თანხლებისათვის აუცილებელი უნარების გამომუშავებას და საქართველოს საწარმოების მოთხოვნილებების რეალიზაციას მინიმუმალური შემადგენლობისა და დონის სპეციალისტების პრაქტიკული მომზადებების საქმიანობაში.

რეცენზენტი აკადემოკოსი რ. ხუროძე

თავი 1

1.1. შესავალი

ამ თავში განხილული ლაბორატორიული პრაქტიკუმი (შემდეგში ლაბორატორიული სამუშაოები) განკუთვნილია ფირმა Siemens - ის პროგრამირებადი ლოგიკურ კონტროლერების (პლკ) გამოყენებისათვის არსებული თეორიული მასალის ათვისებაში, ავტომატიზირებული მართვის სისტემების მოწყობილობებისა და ელემენტების ძირითადი მახასიათებლების შესწავლაში, ტექნოლოგიური პროცესების მართვის პროგრამების შექმნის უნარ–ჩვევების ჩამოყალიბებაში.

ლაბორატორიული სამუშაოების შესრულების შედეგად სტუდენტმა უნდა იცოდეს:

- სხვადასხვა ტიპის პლკ-ს არქიტექტურული თავისებურებები;
- პლკ-ს დაპროგრამირების ენის STEP 7- ის ძირითადი ინსტრუქციები.

უნდა შეძლოს:

- დაადგინოს ავტომატიზირებული მართვის სისტემების ელემენტების პარამტრები;
- შეადგინოს უმარტივესი პროგრამები STEP 7- ის ენაზე და დატესტოს ისინი.

ლაბორატორიული საუშაოები სრულდება სტუდენტების მიერ თეორიული მასალის (ლექციების) გავლის შემდეგ, იმისათვის, რომ პრაქტიკულად აითვისონ და გაიმყარონ თეორიული მასალებით მიღებული ცოდნა.

1.2. მეთოდური რეკომენდაციები ლაბორატორიული სამუშაოების შესასრულებლად

ლაბორატორიული სამუშაოები სრულდება, სპეციალურ ლაბორატორიებში კონკრეტულ სტენდებზე ან კომპიუტერებზე STEP 7-ის პროგრამული პაკეტის გამოყენებით.

სამუშაო ადგილების ნომრები მითითებულ უნდა იქნას მასწავლებლის მიერ. ლაბორატორიული სამუშაოს შესრულების დროს სტუდენტებმა უნდა იმოქმედონ შრომის უსაფრთხოების ტექნიკის წესების დაცვით.

ლაბორატორიული სამუშაოს შესრულება რეკომენდირებულია შემდეგი თანმიმდევრობით:

I ეტაპი

თეორიული მომზადება სამუშაოს შესრულებისათვის, რისთვისაც აუცილებელია წინამდებარე მეთოდური მითითებებისა და თეორიული მასალის შეწავლა (ლექციები).

თეორიული მასალის შესწავლის შედეგად :

სტუდენტმა უნდა იცოდეს:

- შესასწავლი მოწყობილობის ფიზიკური საფუძვლები და მუშაობის პრინციპი;
- მოწყობილობის სქემის ძირითადი ელემენტების დანიშნულება და მათი გავლენა მის მახასიათებლებზე;
- STEP 7 - პლკ–ს პროგრამირების ენის ძირითადი ინსტრუქციები.

უნდა შეეძლოს:

- მართვის ავტომატიზირებული სისტემების ელემენტების მახასიათებლების დადგენა;
- შეადგინოს და დატესტოს უმარტივესი პროგრამები STEP 7 ენაზე.

II ეტაპი

სამუშაოების ჩატარებისათვის უნდა მომზადდეს სამუშაო რვეული (გამოყენებულ იქნას მზა სახის, დანართი 2, ანდა მომზადდეს დამოუკიდებლად).

III ეტაპი

უშუალოდ სამუშაოს შესრულება.

პირველი და მეორე ეტაპების გამოყენება ხდება სამუშაოს შესრულების წინ, ლაბორატორიული სამუშაოს უშუალო შესრულება თავის თავში შეიცავს:

1. მონაცემების შეგროვებას ან პროგრამების შექმნას დანადგარის მუშაობის მოცემული ალგორითმის მიხედვით;
2. ანგარიშის გაფორმებას;
3. ლაბორატორიული სამუშაოს დაცვას.

ა. მონაცემების შეგროვება (სამუშაოს შესრულების ინსტრუქციის მიხედვით)

სამუშაოს შესრულების ინსტრუქცია შეიცავს შემდეგ განყოფილებებს: ლაბორატორიული სამუშაოს ნომერს, სახელს, მიზანას, აღჭურვილობს, შესრულების თანმიმდევრობას.

მონაცემების შეგროვება ხდება შემდეგი თანმიმდევრობით:

- ვსწავლობთ სამუშაოს შესრულების ინსტრუქციას;
- ვიგებთ მუშაობის მიზანს და ქმედებათა თანმიმდევრობას;
- გაუგებარ მომენტებს ვაზუსტებთ მასწავლებლთან;
- ვამზადებთ აუცილებელ ცხრილებს.
- ვასრულებთ ქმედებებს "შესრულების თანმიმდევრობის" პუნქტის მიხედვით.

ამის შემდეგ უნდა მომზადდეს ძირითადი ქმედებებისა და დასკვნების კონსპექტები. შემდეგ კი მონაცემები უნდა დაკონსპექტდს და შეტანილ იქნას ანგარიშში.

ბ. ანგარიშის გაფორმება

ყველა სტუდენტი ინდივიდუალურად აფორმებს ანგარიშს A4 ფორმატის ფურცელზე სამუშაო რვეულის გამოყენებით. დასაშვებია აღნიშნული სამუშაოს შესრულება ხელნაწერით.

ყველა სამუშაოს ანგარიში უნდა შეიცავდეს:

1) ლაბორატორიული სამუშაოს ნომერს (იხ. სამუშაოს შესრულების ინსტუქცია);
2) სამუშაოს სახელწოდებას (იხ. სამუშაოს შესრულების ინსტუქცია);
3) სამუშაოს მიზანს (იხ. სამუშაოს შესრულების ინსტუქცია);
4) აღჭურვილობას (რაც გამოიყენება მოცემულ სამუშაოში);
5) სამუშაოს მსვლელობას (სამუშაოს შესრულების მიმდინარეობის მოწესრიგებული აღწერა, დასკვნები და მონაცემები პუნქტებად, ცხრილის შევსება).

გ. ლაბორატორიული სამუშაოს დაცვა

ლაბორატორიული სამუშაოს დაცვისათვის სტუდენტმა უნდა:

- წარმოადგინოს სამუშაოს შესრულების მიზანი და თანმიმდევრობა;
- შეისწავლოს პრაქტიკული და თეორიული მასალა კითხვების მიხედვით;

- უპასუხოს საკონტროლო კითხვებს ლაბორატორიული სამუშაოს შესახებ, რომელიც შეესაბამება ცოდნას "დამაკმაყოფილებელ ნიშანზე" და დამატებით კითხვების მოცემული თემის მიხედვით უფრო მაღალი შეფასების მიღებისათვის.

დაცულ ლაბორატორიულ სამუშაოზე ხელს აწერს მასწავლებელი და უთითებს თარიღს.

ლაბორატორიული სამუშაოს სრულად შესრულების შემთხვევაში სტუდენტი დაიშვება გამოცდაზე. სტუდენტები, რომლებიც ვერ/არ დაიცავენ ლაბორატორიულ სამუშაოებს, გამოცდებზე არ დაიშვებიან.

1.3. მითითებები უსაფრთხოების ტექნიკის დაცვისათვის

ლაბორატორიული სამუშაოს შესრულებისას სტუდენტი ვალდებულია დაიცვას დაწესებული უსაფრთხოების ტექნიკის წესები.

სტენდის ჩართვა შეძლება მხოლოდ მასწავლებლის ნებართვის შედეგად, აწყობილი სქემის სისწორის შემოწმების შემდეგ. არ შეიძლება სტენდის დატოვება ჩართულ მდგომარეობაში ყურადღების გარეშე.

კომპიუტერულ ლაბორატორიაში მუშაობისას სტუდენტი მოვალეა მკაცრად დაიცავას ჰიგიენური მოთხოვნილებები ვიდეოდისპლეურ ტერმინალებთან, პერსონალურ კომპიუტერებთან და მათთან მუშაობის ორგანიზაციასთან დაკავშირებით (CanPiH 2.2.2.542-96).

1.4. თეორიული ცნობარი

პროგრამირებადი კონტროლერი SIMATIC S7-300

Simatic S7-300კონტროლერს აქვს აგების მოდულური პრინციპი. მოდულები, რომლებისგანაც ის შედგება შეიძლება იყოს როგორც ცენტრალური (განლაგებულნი ცენტრალური პროცესორული მოწყობილობის (ცპმ) გვერდით) ასევე განაწილებული (განლაგებულნი მართვის ობიექტების უშუალო სიახლოვეს) სპეციალური დაყენებებისა ან დანიშნული პარამეტრების აუცილებლობის გარეშე.

S7-300 შედგება შემდეგი კომპონენტებისაგან:

- *კარკასებისაგან* (Racks) - კარკასების დანიშნულებაა მათზე მოდულების განთავსება და მათი ერთმანეთთან შეერთების უზრუნველყოფა;
- *ელექტროკვებისაგან* (Power Supply, PS) - ეს ბლოკი უზრუნველყოფს ელექტროენერგიის მიწოდებას კონტროლერის შიგა მოწყობილობებზე;
- *ცენტრალურ პროცესორულ მოწყობილობისაგან* (Central Processing Unit, CPU) - ამ ბლოკის დანიშნულებაა სამომხმარებლო პროგრამის შენახვა და დამუშავება;
- *ინტერფეისული მოდულებისაგან* (Interface Modules, IM) - მათი დანიშნულებაა კარკასის შეერთება სხვა კარკასთან;
- *სასიგნალო მოდულებისაგან* (Signal Modules, SM) – ამ მოდულების დანიშნულებაა სისტემური სიგნალების ადაპტაცია პროცესორის შიგა დონეებთან, ანდა ამძრავების მართვა ციფრული ანდა ანალოგიური სიგნალების საშუალებით;

- *ფუნქციონალური მოდულებისაგან* (Function Modules, FM) - ეს მოდულები CPU–საგან დამოუკიდებლად ასრულებენ რთულ ანდა დროის მიხედვით კრიტიკულ დამუშავებებს;
- *კომუნიკაციური პროცესორებისაგან* (Communication Processors, CP) - ეს მოდულები ამყარებენ კავშირს დამხმარე ქსელეთან (ქვექსელებთან);
- *ქვექსელებისაგან* (Subnets) - ეს ქვექსელები აერთიანებს პროგრამულ კონტროლერებს ერთმანეთთან და სხვა მოწყობილობებთან.

CPU-ს მეხსიერების არეები

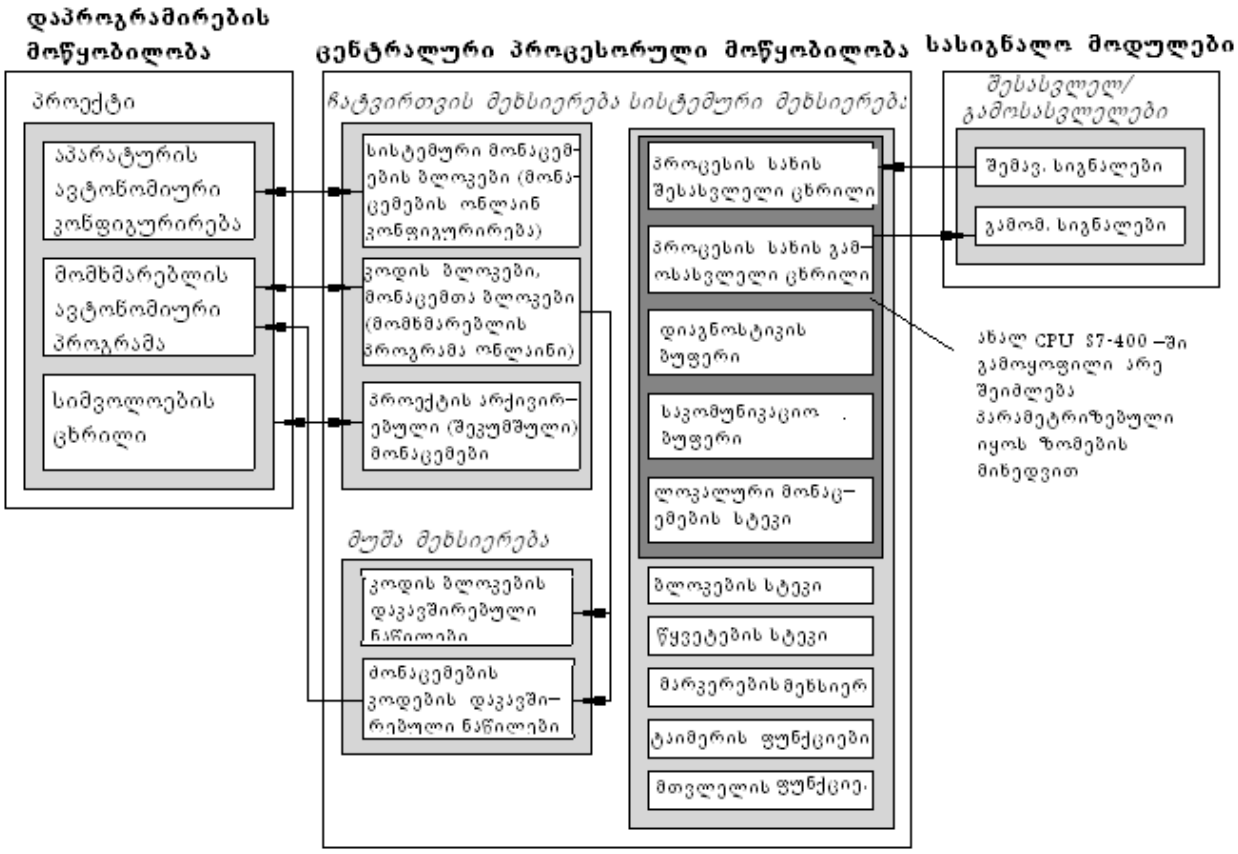
Simatic S7 სისტემებში განაწილებული შეყვანა/გამოყვანა (I/O) წარმოადგენს სისტემის ინტეგრირებულ ნაწილს. CPU თავისი მეხსიერების სხვადასხვა არეებით ქმნის აპარატურულ საფუძველს მომხმარებლის პროგრამების დამუშავებისათვის.

მომხმარებლის მეხსიერება

ნახ. 1.1. – ზე ნაჩვენებია CPU–ს მეხსიერების ის არეები, რომლებსაც პროგრამებისათვის არსებითი მნიშვნელობა აქვთ. თვითონ მომხმარებლის პროგრამა განთავსდება ორ არეში, რომელთაც ეწოდებათ ჩატვირთვის მეხსიერება და მუშა მეხსიერება.

ჩატვირთვის მეხსიერება შეიძლება ინტეგრირებული იყოს CPU–ში ანდა იგი შეიძლება იყოს მიერთებადი (Plugin) მეხსიერების ბარათის საშუალებით. მომხმარებლის მთელი პროგრამა, თავის საკონფიგურაციო მონაცემებთან ერთად განთავსდება ჩატვირთვის მეხსიერებაში.

მუშა მეხსიერება წარმოადგენს მაღალსიხშირულ RAM-ს, რომელიც მთლიანად ჩაშენებულია CPU–ში. მუშა მეხსიერება შეიცავს მომხმარებლის პროგრამის ერთმანეთთან დაკავშირებულ ნაწილებს. ძირითადათ ესენია პროგრამული კოდი და მომხმარებლის მონაცემები.



ნახ. 1.1 CPU-ს მეხსიერების არეები

დამაპროგრამირებელ მოწყობილობას (პროგრამატორს) მთელი პროგრამა, კონფიგურაციების ჩათვლთ გადააქვს ჩატვირთვის მეხსიერებაში. შემდეგ CPU-ს ოპერაციული სისტემა აკოპირებს პროგრამულ კოდსა და სამომხმარებლო მონაცემებს მუშა მეხსიერებაში. ამ პროგრამის უკან, პროგრამატორში გადატვირთვის შემთხვევაში, ჩატვირთვის მეხსიერებიდან არჩეული ბლოკები შეივსება მუშა მეხსიერებაში არსებული მონაცემთა მისამართების მიმდინარე მნიშვნელობებით.

თუ კი ჩატვირთვის მეხსიერება შედგება RAM-ისაგან, მაშინ მომხმარებლის პროგრამის შენახვისათვის საჭირო იქნება სარეზერვო ბატარეიის გამოყენება. იქ კი, სადაც ჩატვირთვის მეხსიერება რეალიზებულია როგორც ინტეგრირებული EEPROM, ანდა როგორც მისაერთებელი EPROM ფლეშ მეხსიერების ბარათი, ამ შემთხვევაში CPU–მ შეიძლება იმუშაოს სარეზერვო ბატარეიის გარეშე.

CPU 3xxIFM–ის ჩატვირთვის მეხსიერების შემადგენლობაში შედის RAM და EEPROM კომპონენტები. პროგრამა გადაიტანება და იტესტება RAM მეხსიერებაში, რის შემდეგაც მენიუს ბრძანების დახმარებით ეს დატესტილი პროგრამა შეიძლება შენახულ იყოს კვების შეშფოთებებისაგან დაცულ ინტეგრირებულ (ჩაშენებულ) EEPROM მეხსიერებაში.

CPU S7-300 – ები (CPU 318–საგან გამოკლებით) აღჭურვილია ჩაშენებული ჩატვირთვის მეხსიერებით RAM, რომელსაც შეეძლება მთელი პროგრამის დატევა. EPROM ფლეშ მეხსიერების ბარათი შეიძლება გამოყენებულ იქნას როგორც მონაცემთა მტარებელი, ანდა როგორც კვების შეშფოთებებისაგან დაცული, მომხმარებლის პროგრამების შენახვის გარემო.

CPU S7-300–ებში მომხმარებლის მეხსიერების (მონაცემთა ბლოკები – data blocks) და სისტემური მეხსიერების (მერკერების მეხსიერება, ტაიმერები, მთვლელები) ცალკეული ნაწილების მიმდინარე მნიშვნელობები შეიძლება შენახულ იყოს მეხსიერების ენერგოდამოუკიდებელ ელემენტზე. ამრიგად, კვების შეშფოთების შემთხვევაში შეიძლება დავიცვათ და უსაფრთხო გავხადოთ მონაცემები, სარეზერვო კვების არარსებობის შემთხვევაშიც კი.

სისტემური მეხსიერება

CPU S7-300 – ები შეიცავს აგრეთვე სისტემურ მეხსიერებას (system memory), რაც პრინციპში წარმოადგენს ცვლადების მისამართებს და რომლებზედაც ხდება მიმართვა პროგრამის მუშაობის დროს. მისამართები გაერთიანებულია არეებად (მისამართების არეები), რომლებიც შეიცავს ცენტრალური პროცესორული მოწყობილობის მიერ განსაზღვრულ მისამართების გარკვეულ რაოდენობას. მისამართები შეიძლება იყოს მაგ. შესასვლელები, რომლების გამოიყენება, მაგალითად, გადამრთველის მყისიერი კონტაქტის (ღილაკის) ანდა შეზღუდვის გადამრთველების (საბოლოო გამომრთველების) სასიგნალო მდგომარეობის სკანირებისათვის; ასევე გამოსასვლელები, რომლებიც შეიძლება გამოყენებულ იქნას კონტაქტორებისა და ნათურების მართვისთვის.

სისტემური მეხსიერებაში შედის მისამართების შემდეგი არეები:

- შესასვლელების არე (Inputs, I) – იგი წარმოადგენს ციფრული შესასვლელების მოდულების სახეს (”პროცესის სახე”);
- გამოსასვლელების არე (Outputs, Q) - იგი წარმოადგენს ციფრული გამოსასვლელების მოდულების სახეს (”პროცესის სახე”);
- მერკერების მეხსიერებისა არე (Bit memory, M) - ეს არე ინახავს ისეთ ინფორმაციას, რომელიც ხელმისაწვდომია მთელი პროგრამისათვის;
- ტაიმერების არე (Timers, T) - ეს მეხსიერების ის უჯრედებია, რომლებიც გამოიყენება დროის დაყოვნებებისა და მონიტორინგის ინტერვალების რეალიზაციისათვის;
- მთვლელების არე (Counters, C) - ეს პროგრამულ დონეზე ორგანიზებული მეხსიერების უჯრედებია, რომლებიც გამოიყენება გადათვლების საწარმოებლად ზრდადობის ან კლებადობის მიხედვით.
- დროებითი ლოკალური მონაცემების არე (Temporary Local Data, L). მეხსიერების ის უჯრედებია, რომლებიც გამოიყენება დინამიური შუალედური ბუფერების სახით ბლოკების დამუშავების დროს. დროებითი ლოკალური მონაცემები

განლაგებულია L-სტეკში. რომლებიც დინამიურად გამოიყენება CPU-ს მიერ პროგრამის შესრულების დროს.

განასხვავებენ ბინარულ (დისკრეტულ) სიგნალებს და ანალოგურ სიგნალებს:

ბინარული (დისკრეტული) სიგნალები

ბინარული სიგნალი (binary signal) შეიცავს ერთ ბიტ ინფორმაციას. დისკრეტული სიგნალების მაგალითია CPU-ს შესასვლელი სიგნალები, რომლებიც მას მიეწოდება ციფრული შესასვლელი მოდულების გავლით საბოლოო გამომრთველებიდან, მყისიერი კონტაქტის გამომრთველებიდან და ა. შ., აგრეთვე გამომავალი სიგნალები, რომლებიც ციფრული გამომავალი მოდულების გავლით ანთებენ ან აქრობენ ნათურებს, მართავენ (ჩართავენ ან გამორთავენ) კონტაქტორებს და სხვ.

ანალოგური სიგნალები

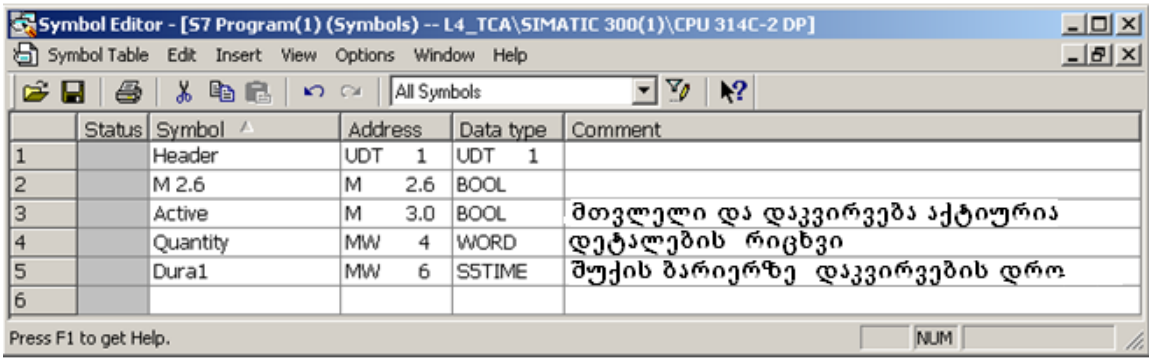
ანალოგური სიგნალი (analog signal), ციფრულში გარდაქმნის შემდეგ, შეიცავს 16 ბიტ ინფორმაციას. ანალოგური სიგნალი შეესაბამება "არხს", რომელიც აისახება კონტროლერში მანქანური სიტყვის ფორმით (word), ორი ბაიტის სახით. ანალოგური შესასვლელი სიგნალები (მაგ. ძაბვა თერმორეზისტორიდან) შედის ანალოგურ შესასვლელ მოდულებში, გარდაიქმნება ციფრულ კოდში და ამის შემდეგ წარმოიშვება მათი დამუშავების შესაძლებლობა კონტროლერში როგორც 16–თანრიგიანი სიგნალისა (16 საინფორმაციო ბიტი). მეორე მხრივ კი 16–თანრიგიანმა სიგნალმა ანალოგური გამომავალი მოდულის საშუალებით შეიძლება მართოს ანალოგური ინდიკატორი, რისთვისაც ეს 16 თანრიგიანი ინფორმაცია გარდაიქმნება ანალოგურ სიდიდეში (მაგ. დენის ძალაში).

სიმბოლოთა ცხრილები

ეს ცხრილები მუშაობენ მმართველ პროგრამაში მისამართებთან. ესენია შესასვლელების, გამოსასვლელების, ტაიმერების, ბლოკების და ა. შ. მისამართები. მათ შეიძლება მივაკუთნოთ აბსოლუტური მისამართი (მაგ. I1.0) ანდა სიბვოლური მისამართი (მაგ. Start signal). სიბვოლური დამისამართება აბსოლუტური მისამართის ნაცვლად გამოიყენებს სახელებს (სახელწოდებებს). თუ კი გამოყენებულ იქნება გააზრებული სახელები, მაშინ პროგრამა გახდება უფრო მეტად გასაგები.

სიმბოლურ დამისამართებაში განირჩევა *ლოკალური (local)* და *გლობალური (global)* სიმბოლოები. ლოკალური სიმბოლო ცნობილია მხოლოდ იმ ბლოკში, რომელშიც იგია განსაზღვრული. შეიძლება გამოყენებულ იქნას ერთდაიგივე ლოკალური სიმბოლო სხვადასხვა ბლოკში სხვადასხვა დანიშნულების მიზნით. გლობალური სიმბოლოები ცნობილია მთელი პროგრამისთვის და აქვთ ერთიდაიგივე მნიშველობა ყველა ბლოკისათვის. გლობალური სიმბოლოები განისაზღვრება სიმბოლოთა ცხრილში ((ობიექტი *Simbols* (*სიმბოლოები*) *S7 Program – ის* კონტეინერში *(S7* – პროგრამა)).

ნახ. 1.2 – ზე წარმოდგენილია სიმბოლოთა ცხრილის მაგალითი.



ნახ. 1.2 სიმბოლოთა ცხრილის მაგალითი

სიმბოლოთა ცხრილში სახელები შეიძლება მივაკუთნოთ შემდეგ მისამართებს და ობიექტებს:

- შესასვლელებს I, გამოსასვლელებს Q, პერიფერიულ შესასვლელებს PI და პერიფერიულ გამოსასვლელებს PQ;
- მერკერებს M, ტაიმერებს T და მთვლელებს C;
- კოდურ ბლოკებს OB, FB, FC, SFC, SFB და მონაცემთა ბლოკებს DB;
- მომხმარებლის მიერ განსაზღვრულ მონაცემთა ტიპებს UDT;
- ცვლადების ცხრილს (variable table) VAT.

მონაცემთა მისამართები და მონაცემთა ბლოკები ჩართულია ლოკალური მისამართების რიცხვში; ასოცირებული სიმვოლოები გლობალური მონაცემების ბლოკების შემთხვევაში განისაზღვრება მონაცემთა ბლოკის აღწერის განყოფილებაში, ხოლო ეგზემპლიარების მონაცემთა ბლოკების შემთხვევაში – ფუნქციონალური ბლოკის აღწერის განყოფილებაში.

პროგრამების რედაქტორი

სამომხმარებლო პროგრამების შექმნის საჭიროებებისთვის ბაზური პაკეტი STEP 7 შეიცავს პროგრამების რედაქტორს (Program Editor) პროგრამირების შემდეგი სახის ენებისათვის:

- LAD (*ladder Logic*ან *ladder diagram* – კონტაქტური გეგმა, რელეური ლოგიკის დიაგრამების მსგავსი წარმოდგენა, მრავალსაფეხუროვანი სქემა;
- FBD (*function block diagram* - ფუნქციონალური ბლოკების დიაგრამა ან ფუნქციონალური გეგმა);
- STL (*statement list* - ოპერატორების ან მნემონიკების სია, ასემბლერის მაგვარი ენა.

LAD და FBD ენებზე პროგრამირება ხდება არსებული ბიბლიოთეკიდან ბლოკების არჩევითა და მათი შეყვანით რედაქტორში და მათივე შესასვლელებსა და გამოსასვლელებზე შესაბამისი მისამართების დაყენებით.

თუ კი გლობალური მისამართებისთვის გამოიყენება სიმბოლური დამისამართება, მაშინ ეს სიმბოლოები უკვე უნდა ქონდეს მინიჭებული აბსოლუტურ მისამართებს; მიუხედავად ამისა, შეიძლება ახალი სიმბოლოების შეყვანა ანდა მათი შეცვლა პროგრამის შეყვანის დროს.

ბლოკები

პროგრამა, მისი მოსახერხებლად წაკითხვისა და ადვილად გაგების მიზნით შეიძლება დაყოფილ იქნას ნებისმიერი რაოდენობის ნაწილებად. STEP 7 – ის პროგრამირების ყველა ენა მხარს უჭერს ამ კონცეფციას და ამისათვის ხელთ გვეძლევა გამოსაყენებლად აუცილებელი ფუნქციების ნაკრები. პროგრამის თვითოეული ნაწილი უნდა იყოს დამოუკიდებელი და უნდა აღჭურვილ იყოს ტექნოლოგიური ან ფუნქციონალური ბაზისით. პროგრამის ამ ნაწილებს ეწოდებათ "ბლოკები" ("Blocks"). ამრიგად, ბლოკი ეს პროგრამის ნაწილია, რომელიც განსაზღვრულია თავისი ფუნქციონალობით, სტრუქტურით ანდა ამოსახსნელი ამოცანით.

ბლოკის ტიპები

STEP 7 სხვადასხვა ამოცანებისათვის წარმოადგენს ბლოკების სხვადასხვა ტიპს.

- *სამომხმარებლო ბლოკებს* (user blocks) - შეიცავს სამომხმარებლო პროგრამას და მომხმარებლის მონაცემებს;
- *სისტემურ ბლოკებს* (system blosks) – შეიცავს სისტემურ პროგრამასა და სისტემურ მონაცემებს;
- *სტანდარტულ ბლოკებს* (standart blocks) – უშუალო გამოყენებისათვის (წინასწარ შექმნილი) მზა ბლოკები, ისეთები მაგალითად როგორიცაა ფუნქციონალური მოდულებისა (FM) და კომუნიკაციური პროცესორების (CP) დრაივერები.

სამომხმარებლო ბლოკები

დიდი და რთული პროგრამებისათვის რეკომედირებულია და ზოგჯერ აუცილებელიც კი არის პროგრამის "სტრუქტურირება" (დაყოფა) საკმარისი რაოდენობის ბლოკების გამოყოფით. დანართისაგან (სამომხმარებლო პროგრამა) დამოკიდებულებით შეიძლება გამოყენებულ იქნას სხვადასხვა ტიპის ბლოკები:

ორგანიზაციული ბლოკები (Organization blocks – OB) – ეს ბლოკები გამოიყენება როგორც ინტერფეისი ოპერაციულ სისტემასა და მომხმარებლის პროგრამას შორის. CPU-ს ოპერაციული სისტემა ახდენს ორგანიზაციული ბლოკის გამოძახებას რაიმე ხდომილების წარმოშობის შემთხვევაში, მაგ. აპარატურული წყვეტების ანდა დროის მიხედვით დღეღამური წყვეტების შემთხვევაში. მთავარი პროგრამა იმყოფება ორგანიზაციულ ბლოკში OB1–ში. დანარჩენ ორგანიზაციულ ბლოკებს გააჩნიათ მუდმივი ნომრები, დანიშნულებები და დამოკიდებულებები იმ ხდომილებებისაგან, რომელთა დამუშავებებისთვისაც ისინი გამოიძახება.

ფუნქციონალური ბლოკები (Function blocks – FB) – ეს ბლოკები წარმოადგენს პროგრამის შემადგენელ ნაწილებს, რომელთა გამოძახება შეიძლება დაპროგრამირებულ იყოს ბლოკის პარამეტრების დახმარებით. მათ აქვთ მეხსიერების არე ცვლადებისათვის (variable memory), რომელიც განლაგებულია მონაცემთა ბლოკში. ეს უკანასკნელი ბლოკი კი მუდმივადაა დანიშნული (განკუთვნილი) ფუნქციონალური ბლოკისთვის, ან უფრო სწორად, ფუნქციონალური ბლოკის *გამოძახებისთვის (call).*

გარდა ამისა, ფუნქციონალური ბლოკის თვითოეული გამოძახებისათვის შეიძლება დანიშნულ იყოს სხვა მონაცემთა ბლოკი (მონაცემთა ისეთივე სტრუქტურით, მაგრამ სხვა მნიშვნელობების შემცველობით). ასეთ, მუდმივად დანიშნულ ბლოკს, ეწოდება მონაცემთა ეგზემპლიარული ბლოკი ანდა მონაცემთა ბლოკის ეგზემპლიარი (instance data block), ხოლო ფუნქციონალური ბლოკისა და მონაცემთა ბლოკის ეგზემპლიარის ერთობლივ გამოძახებას ეწოდება გამოძახების ეკზემპლიარი (call instance) ან უბრალოდ "ეგზემპლიარი" ("instance"). ფუნქციონალურ ბლოკებს შეუძლიათ ასევე შეინახონ თავიანთი ცვლადები გამომძახებელი ფუნქციონალური ბლოკის მონაცემთა ეგზემპლიარულ ბლოკში; ასეთი ეგზემპლიარს ეწოდება "ლოკალური ეგზემპლიარი" ("local instance").

ფუნქციები (*Functions – FC*) - ეს ბლოკები გამოიყენება ავტომატიზაციის ხშირად განმეორებადი ანდა რთული ფუნქციების რეალიზაციისათვის. მათთვის შესაძლებელია დანიშნულ იყოს პარამეტრები. ფუნქციებს შეუძლიათ დააბრუნონ მნიშვნელობები (ეწოდება ფუნქციის მნიშვნელობები) გამომძახებელ ბლოკში. ფუნქციის მნიშვნელობა წარმოადგენს არააუცილებელ პარამეტრს. გარდა ამისა ფუნქციას შეიძლება ჰქონდეს სხვა გამომავალი პარამეტრი. ფუნქციები არ ინახას ინფორმაციას და მათ არა აქვს დანიშნული მონაცემთა ბლოკი.

მონაცემთა ბლოკები (Data blocks – DB) – ეს ბლოკები შეიცავს პროგრამის მონაცემებს. მონაცემთა ბლოკების დაპროგრამირებისას განსაზღვრავენ მონაცემთა შენახვის ფორმას (რომელ ბლოკში, რა რიგითობით და ამასთან როგორი მონაცემთა ტიპია გამოყენებული). მონაცემთა ბლოკები გამოიყენება ორგვარი ხერხით:

1) როგორც გლობალური მონაცემთა ბლოკები (global data blocks);
2) როგორც მონაცემთა ბლოკების ეგზემპლიარები (instance data blocks).

მონაცემთა გლობალური ბლოკი მომხმარებლის პროგრამაში არის ასე ვთქვათ, "თავისუფალი" მონაცემთა ბლოკი და იგი არ დაენიშნება კოდურ ბლოკს. მაგრამ ეგზემპლიარული მონაცემთა ბლოკი დაენიშნება ფუნქციონალურ ბლოკს და ინახავს ამ ბლოკის ლოკალური მონაცემების ნაწილს.

განსაზღვრული ბლოკური ტიპის ბლოკების რაოდენობა და ბლოკების ზომები დამოკიდებულია CPU–ს ტიპზე. საორგანიზაციო ბლოკების რიცხვი და მათი ნომრები ფიქსირებულია; ისინი დაენიშნება მათ CPU–ს ოპერაციულ სისტემის მიერ. სხვა ტიპის ბლოკებს შეიძლება დამოუკიდებლად დაენიშნოს ნომრები განსაზღვრულული დიაპაზონიდან. აგრეთვე შეიძლება სიმბოლოების ცხრილის დახმარებით დაენიშნოს

თვითოეულ ბლოკს სახელი (სიმბოლო) და შემდეგ მიკითხულ იქნას ამ ბლოკებთან ამ მიკუთვნებული სახელებით.

სისტემური ბლოკები

სისტემური ბლოკები იოპერაციული სისტემის კომპონენტებია. ისინი შეიძლება შეიცავდეს პროგრამებს (სისტემური ფუნქციები – SFC, ან სისტემური ფუნქციონალური ბლოკები – SFB) ანდა მონაცემებს (სისტემური მონაცემთა ბლოკები – SDB). სისტემური ბლოკები საშუალებას გვაძლევს მივაკითხოთ მნიშვნელოვან სისტემურ ფუნქციებს, ისეთებს როგორიცაა CPU–ს შიგა ტაიმერების მართვა, ანდა სხვადასხვა საკომუნიკაციო ფუნქციებს.

შეიძლება გამოძახებულ იქნან SFC და SFB, მაგრამ არ შეიძლება მათი არც ცვლილება არც თვითნებური დაპროგრამირება. თვითონ ბლოკები არ იჭერენ ადგილებს მომხმარებლის მეხსიერებაში (user memory); ამ ბლოკების მხოლოდ გამოძახებები და აგრეთვე SFB ბლოკების ეგზემპლიარული მონაცემთა ბლოკები განლაგდებიან სამომხმარებლო მეხსიერებაში.

SDB ბლოკები შეიცავს ინფორმაციას ისეთ ქმედებებზე, როგორიცაა მაგალითად ავტომატიზაციის სისტემის ანდა მოდულების პარამეტრების კონფიგურაცია. STEP 7 სისტემა თვითონ გენერირებს ამ ბლოკებს და მართავს მათ. მიუხედავათ ამისა შეიძლება განისაზღვროს მათი შემცველობა, მაგ. სადგურის კონფიგურირების დროს. ჩვეულებრივად SDB ბლოკები განთავსდება ჩატვირთვის მეხსიერებაში (load memory). მომხმარებლის პროგრამიდან მათთან წვდომის მიღება შეუძლებელია.

სტანდარტული ბლოკები

შესაქმნელ ფუნქციებთან და ფუნქციონალურ ბლოკებთან ერთად დამატებით შეიძლება გამოყენებულ იქნას მზა ბლოკები (ეგრეთწოდებული "სტანდარტული ბლოკები").

ისინი შეიძლება მოწოდებულ იქნას მონაცემთა მატარებლებზე, ან მოთავსებულ იქნას STEP 7 პაკეტის შემადგენლობაში შემავალ ბიბლიოთეკებში (მაგ. IEC – ფუნქციები ანდა ფუნქციები S5/S7 გარდაქმნებისათვის).

1.5. ლაბორატორიულ-პრაქტიკული სამუშაო №1

SPET 7 პროგრამულ პაკეტთან გაცნობა, პროექტის შექმნა

სამუშაოს მიზანი: STEP 7 Simatic Manager პროგრამული პაკეტის გაცნობა, უკონტაქტო გეგმით წარმოდგენილი პროგრამების შედგენის მეთოდებისა რედაქტირების ათვისება.

აღჭურვილობა: STEP 7 Simatic Manager პროგრამული პაკეტი.

თეორიული მონაცემები

ამოსახსნელი ამოცანის სირთულის მიხედვით S7-300-ში კონტროლერებში შეიძლება გამოიყენებულ იყოს 8 ტიპის ცენტრალური პროცესორი:

- CPU 312 IFM: კომპაქტური ცენტრალური პროცესორი დისკერტული სიგნალებისათვის გათვალისწინებული ჩაშენებული შემყვანებისა და გამომყვანების ნაკრებით, რომლის გამოყენება შეიძლებაავტომატიზაციის ავტონომიური სისტემების;
- CPU 313: იაფი ცენტრალური პროცესორი, ავტომატიზაციის მცირე მოდულური სისტემების აგებისათვის;.
- CPU 314 IFM: კომპაქტური ცენტრული პროცესორი ჩაშენებული შემყვანებისა და გამომყვანების ნაკრებით როგორც დისკრეტული, ასევე ანალოგური სიგნალებისათვის, რომლის გამოიყენება შეიძლება ავტომატიზაციის ავტონომიური სისტემებისათვის, რომელთაც ახასიათებთ ხდომილებებზე რეაქციის მცირე დრო და სიგნალების დამუშავების სპეციალური ფუნქციების შესრულების უნარი;
- CPU 314: ცენტრალური პროცესორი ავტომატიზაციის მოდულური სისტემების აგებისათვის, რომლებსაც ახასიათებს მონაცემების დამუშავების მაღალი სიჩქარე.
- CPU 315/CPU 315 2 DP: ცენტრალური პროცესორი ავტომატიზაციის საშუალო ან დიდი მოცულობის პროგრამების მქონე სისტემების შექმნისათვის, რომლებიც ემსახურება PROFIBUS DP სალტესთან მიერთებულ ლოკალურ და განაწილებულ შეყვანა–გამოყვანის სისტემებს;
- CPU 316-2DP: ცენტრალური პროცესორი ინფორმაციის დამუშავების რთული ალგორითმების მქონე ავტომატიზაციის მოდულური სისტემის შექმნისათვის, რომლებიც გამოიყენებენ PROFIBUS DP სალტესთან მიერთებულ ლოკალურ და განაწილებულ შეყვანა–გამოყვანის სისტემებს;
- CPU 318–2 DP: ცენტარლური პროცესორები დიდი მოცულობის მქონე პროგრამების შესრულებისათვის და აგრეთვე PROFIBUS - DP ქსელს დახმარებით განშტოებული კონფიგურაციის მქონე განაწილებული შეყვანა–გამოყვანების მომსახურებისათვის.

ცენტრული პროცესორების ძირითადი ტექნიკური მახასიათებელი

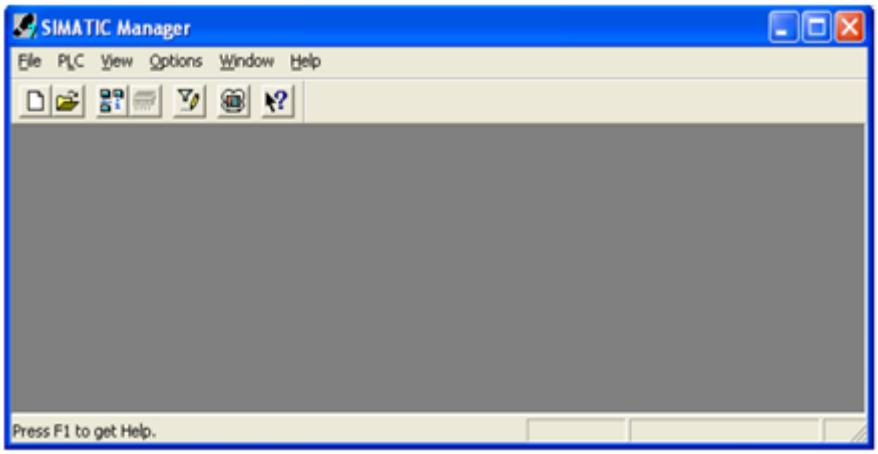
- მაღალი სწრაფქმედება. ლოგიკური ბრძანებების შესრულების დრო CPU 312 IFM და CPU 313 პროცესორებისათვის შეადგენს 600ნწმ, CPU 314 IFM…. CPU 316 პროცესორებისათვის – 300ნწმ , CPU 318-2 პროცესორისათვის - 100ნწმ;
- ოპერატიული მეხსიერების მოცულობა, რომლებიც შეესაბამება ამოსასხსნელი ამოცანების კლასს: 6კბაიტდან CPU 312 IFM-პროცესორისთვის 512 კბაიტადე CPU 318-2 პროსესორისთვის;
- გაფართოვების მოქნილი შესაძლებლობები. CPU 312 IFM და CPU 313 პროცესორები იძლევა ერთკარკასიანი კონფიგურაციის შექმნის შესაძლებლობას. დასაშვებია 8-მდე შეყვანა–გამოყვანის მოდულის შეერთების შესაძლებლობა. დანარჩენი ცენტრალური პროცესორი კი იძლევა 32 მოდულამდე შეერთების შესაძლებლობას ოთხკარკასიანი კონფიგურაციის გათვალისწინებით.

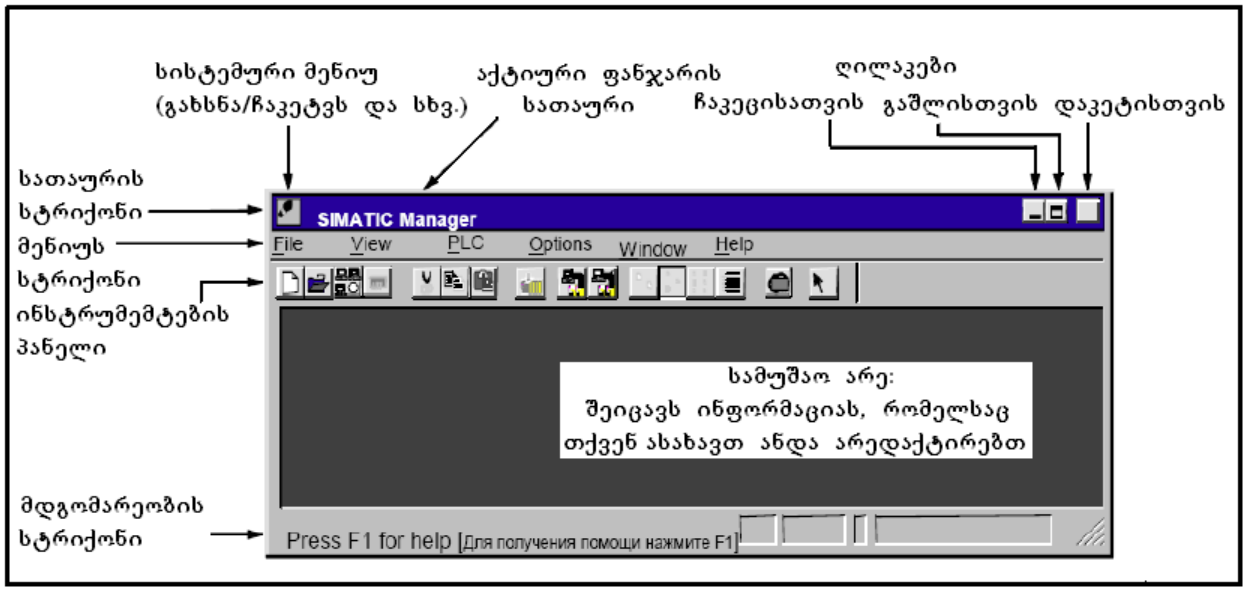
S7-300 კონტროლერების პროგრამის შექმნისათვის გამოიყენება STEP 7 Simatic Manager –ის პაკეტი.

STEP 7 Simatic Manager –ის პაკეტის ძირითადი უტილიტები, რომლებიც ხელმისაწვდომია SIMATIC -STEP 7 -ის საქაღალდისგან შემდეგია:

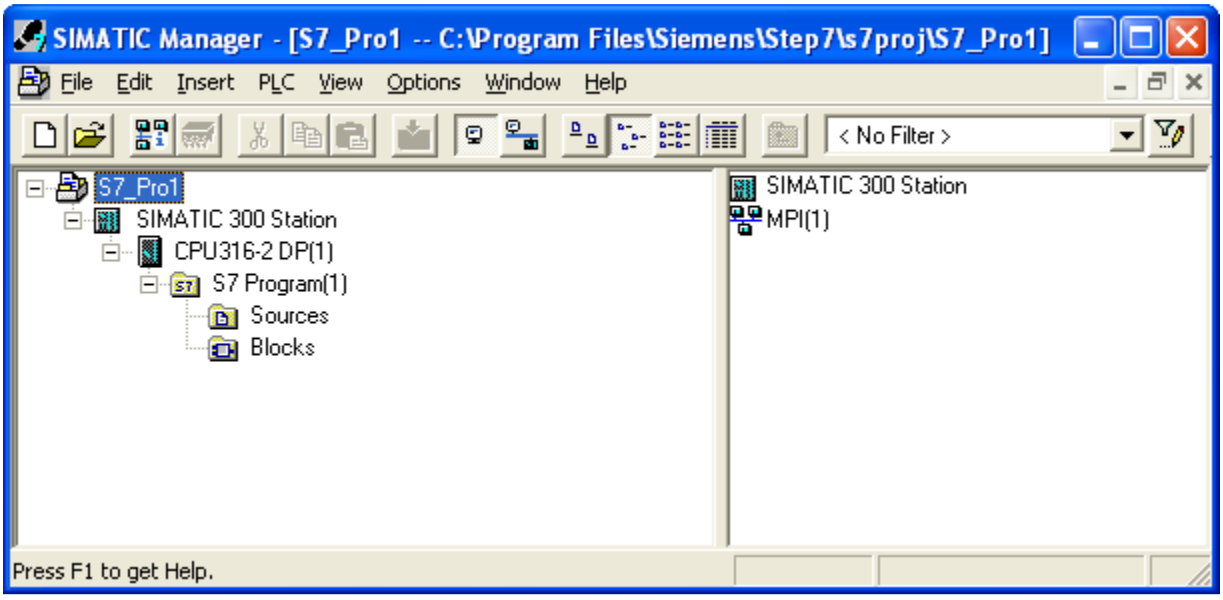
- SIMATIC Manager;
- LAD, STL, FDB –programming S7;
- Memory Card Parameter Assignment;
- Netpro – Configuring Networks;
- PID Control Parameter Assignment;
- S7 SCL – programming S7 Blocks;
- S7-GRAPH – programming Sequential Control System;
- S7 – PDIAG – Configuring Process Diagnostic;
- S7-PLCSM SimulatingModules;
- Setting the PG-PC Interface;

SIMATIC Manager - ეს S7–ის ობიექტების რედაქტირების გრაფიკული ინტერფეისია (პროექტები, სამომხმარებლო პროგრამების ფაილები, სადგურების აღჭურვილობა და ინსტრუმენტები). ამ საშუალებების ძირითადი ფანჯარა ნაჩვენებია ნახ. 1–ზე.

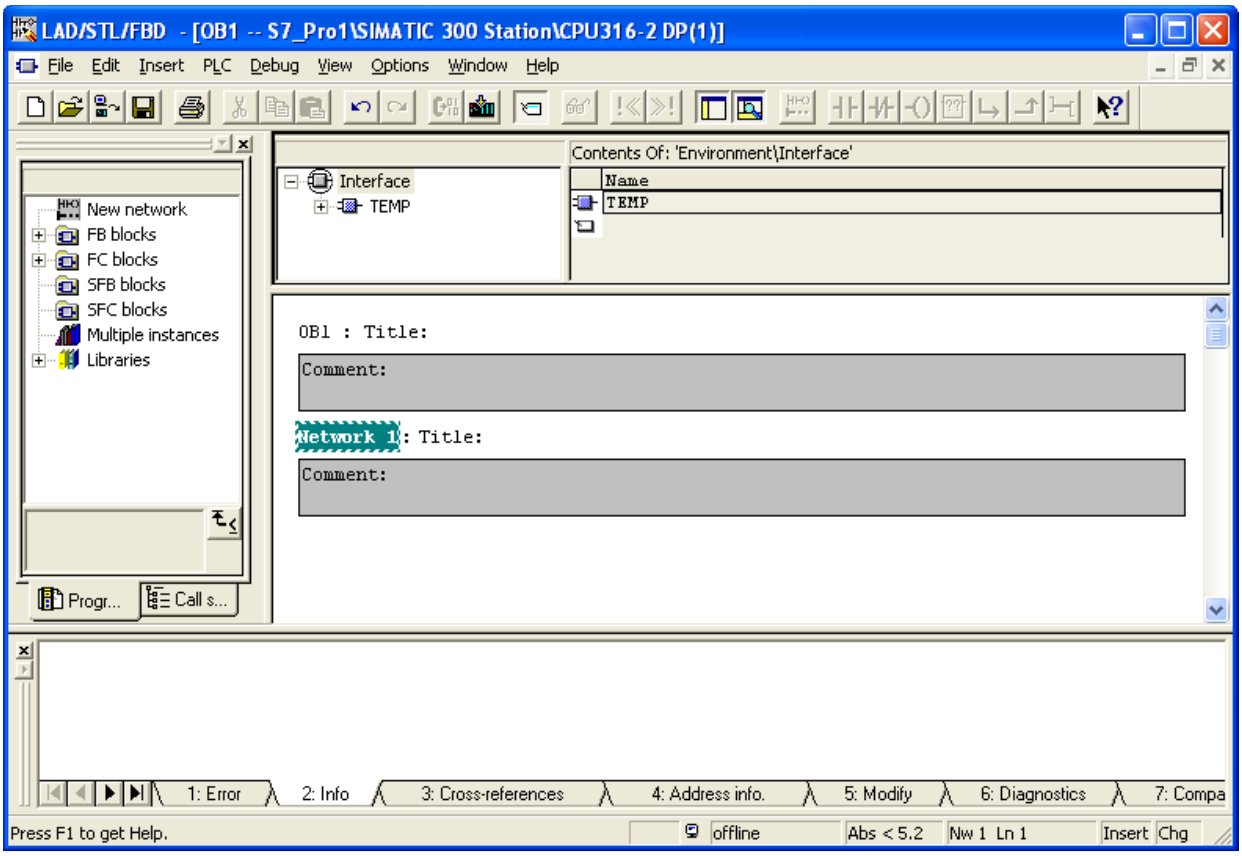




ნახ. 1SIMATIC Manager -ის მენიუ და ინსტრუმენტების პანელი



ნახ. 2 პროექტის სტრუქტურა SIMATIC Manager - ში

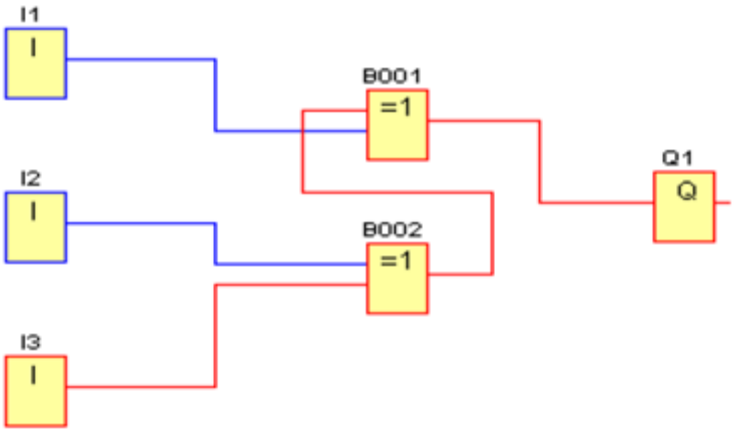


ნახ.3 LAD/STL/FBD რედაქტორის ფანჯარა

1. **სამუშაოს შესრულების თანმიმდევრობა:**

1.1 გაუშვით პროგრამა Simatic Manager, შექმენით პროექტი CPU 315 – 2DP -სთვის ერთი OB1 ბლოკით და გაეცანით პროგრამის ფანჯრის ძირითად ელემენტებს;

1.2 პროგრამის საშუალებების გამოყენებით შეიყვანეთ შემდეგი პროგრამა:



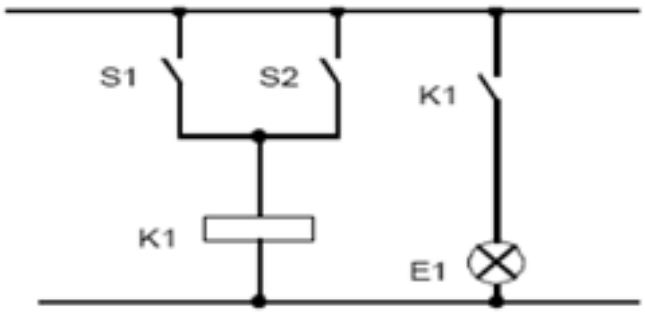
სიმულაციის ღილაკის დაჭერის შედეგად შეამოწმეთ პროგრამის მუშაობა და შეადგინეთ ჭეშმარიტების ცხრილი ამ სქემისთვის.

B 001 და B 002 ელემენტები შეცვალეთ ელემენტით **და**.

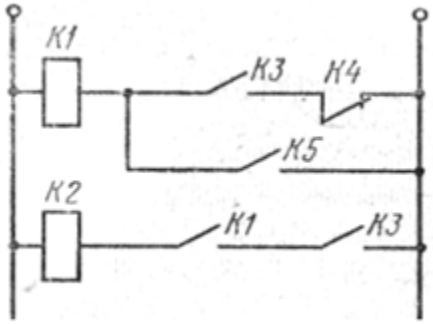
სიმულაციის ღილაკის დაჭერის შედეგად შეამოწმეთ პროგრამის მუშაობა და შვადგინეთ ჭეშმარიტების ცხრილი ამ სქემისთვის;

1.3 პროგრამის საშუალებების გამოყენებით შევადგინეთ პროგრამა უკონტაქტო ელემენტების გამოყენებით, რომელიც ასრულებს მუშაობას შემდეგი სახის ალგორითმების შესაბამისად.

1.3.1 E1 ნათურა უნდა აინთოს ნებისმიერი S1 და S2 გამომრთველების ჩართვისას



1.3.2



1.4 პროგრამის საშუალებების გამოყენებით შეადგინეთ პროგრამა, რომელიც შეძლებს შემდეგ ფუნქციას რეალიზაციას.

$$Fz=[(a+d)*b+z]*\bar{c}$$

შეამოწმეთ ამ პროგრამის მუშაობა ემულიაციის რეჟიმში.

2. ანგარიში უნდა შეიცავდეს

2.1. სამუშაოს სახელს;

2.2. მიზანს;

2.3. აღჭურვილობის ჩამონათვალს;

2.4. ფანჯრის ელემენტებსა და მათ დანიშნულებას;

2.5. სქემებსა და პროგრამებს, რომლებიც რეალიზაციას გაუწევს ამ ფუნქციებს;
2.6. ჭეშმარიტების ცხრილებს 1.2 პუნქტისათვის;
2.7. დასკვნას სამუშაოს შესახებ.

3. საკონტროლო საკითხები და დავალებები

3.1. ჩამოთვალეთ პლკ S7-300-ის ძირითადი პარამეტრები;
3.2. როგორია პლკ S7-300-ის გამოყენების არე;
3.3. პროგრამის ფანჯრების რომელი ძირითადი ელემენტებია თქვენთვის ცნობილი და რა არის მათი დანიშნულება.

1.6. ლაბორატორიულ-პრაქტიკული სამუშაო №2

ცენტრალური თაროს კონფიგურირება

სამუშაოს მიზანი: ცენტრალური კარკასის კონფიგურირების მეთოდების ათვისება ტექნოლოგიური პროცესების (ტპ) მართვის პროგრამების შექმნისას.

მოწყობილობა:STEP 7 პროგრამული პაკეტი.

თეორიული ცნობები

აპარატურული ნაწილის შექმნა იწყება კარკასის დამატებით (Rack), რომელიც იმყოფება შესაბამის კატალოგში. მაგალითად, SIMATIC 300 სადგურის შექმნისას უნდა გაიხსნას SIMATIC 300 ელემენტების კატალოგი და Rack-300 საქაღალდედან დაემატოს Rail ელემენტი. დამატება შეიძლება ან ორმაგი დაწკაპებით ანდა "drag &drop" ტექნოლოგის გამოყენებით მისი გადათრევით.

ძირითადი წესი

მოდულები უნდა განლაგდეს ერთმანეთის მიყოლებით გამოტოვებების გარეშე.

თარო 0:

- სლოტი 1: მხოლოდ კვების ბლოკი (მაგ 6E57307-..) ან ცარიელი;
- სლოტი 2: მხოლოდ CP11 (მაგ, 6E57314-..);
- სლოტი 3: ინტერფეისური მოდული (მაგ, 6E57 360-361…) ან ცარიელი;
- სლოტები 4-დან 11-მდე: სასიგნალო ან ფუნქციონალური მოდულები, კომუნიკაციური პროცესორები ან ცარიელები.

1. სამუშაოს შესრულების თანმიმდევრობა

1.1 გაუშვით პროგრამა SIMATIC Manager, შექმენით პროექტი CPU 315-2 DP – ის, შეყვანა-გამოყვანის ბლოკებისა და OB1 – ს ერთი ბლოკის გამოყენებით;

1.2 Hardwar პროგრამის გამოყენებით შეასრულეთ პროექტის აპარატული ნაწილის კონფიგურირება შემდეგი მოდულების დაყენებით;

№ п/п	მოდულის ტიპი	მარკირება	საიდენტიფიკაციო ნომერი
1	კვების ბლოკი	PS 307 10A	6ES7 307-1KA00-0AA0
2	დისკრეტული შეყვანის ბლოკი	SM321 DI16xDC24V	6ES7 321-1BH81-0AA0
3	ანალოგური შეყვანის ბლოკი	SM331 AI2x12Bit	6ES7 331-7KB01-0AB0
4	დისკრეტული გამოყვანის ბლოკი	SM322 DO16xDC24V/0.5A	6ES7 322-1BH00-0AA0
5	ანალოგური გამოყვანის ბლოკი	SM332 AO2x12Bit	6ES7 332-5HB00-0AB0

1.3 გამოიყენეთ რა პროექტის შექმნილი კონფიგურაცია, შეასრულეთ პარამეტრების აწყობა შემდეგი სახით:

- CPU მისამართი MPI ქსელში უნდა იყოს 4;
- აკრძალეთ CPU –ს სტარტი მიმდინარე (განისაზღვრება ОС–ით) და მოცემული კონფიგურაციების განსხვავებისას;
- დააყენეთ მაქსიმალური დრო შეტყობინების მიღებისათვის მოდულების მზადყოფნის შესახებ და პარამეტრების გადაცემის შესახებ მოდულებში 700 მწმ–ის ტოლი;
- დააყენეთ ციკლის მაქსიმალური დრო 250 მწმ–ის ტოლი;
- დააყენეთ ციკლის დროის გაზრდის კოეფიციენტი (K) საკომუნიკაციო დატვირთვის ხარჯზე 30%-ის ტოლი;
- გააქტიურერთ და მიაკუთვნეთ სინქრონიზაციის ბაიტს მისამართი MB 10 - ს ტოლი;
- რემანენტური მეხსიერებისათვის დააყენეთ შემდეგი სახის მნიშვნელობები:
  . MBO - MB10 მარკერები;
  . მთვლელები CO-C10;
  . ტაიმერები TO – T4.
- შეასრულეთ სისტემური საათის დღეღამეური ჩამორჩენის კონპენსაცია 4 წამის სიდიდით;
- ნება დართეთ პარამეტრებთან მხოლოდ წაკითხვის ხელწვდომას;
- შეცვალეთ შეყვანისა და გამოყვანის მოდულების მისამართები ისეთნაირად, რომ მისამართები იწყებოდეს 124-დან და იყოს უწყვეტი ყველა მოდულის გათვალისწინებით.

1.4 შეამოწმეთ შექმნილი კონფიგურაცია კორექტულობაზე;

1.5 ჩატვირთეთ კონტროლერის სიმულატორში კონფიგურაცია;

1.6 შეამოწმეთ სინქრონიზაციის ბაიტის მუშაობა, რისთვისაც დაწერეთ პროგრამა, რომელიც ასრულებს საკონტროლო ნათურის ციმციმს 0,5 ჰც, 1ჰც და 2ჰც სიხშირით შემავალი სიგნალისაგან (1, 2, 3) დამოკიდებულებით;

1.7 დანადგარის კონტროლერთან მიერთებისას წაიკითხეთ მისი კონფიგურაცია და შეამოწმეთ მისი კატალოგთან შესაბამისობა.

2. ანგარიში უნდა შეიცავდეს:

2.1. სამუშაოს დასახელებას;
2.2. სამუშაოს მიზანს;
2.3. ხელსაწყოთა ჩამონათვალს;
2.4. შექმნილ კონფიგურაციას;
2.5. დანადგარის პლკ–ისგან წაკითხულ კონფიგურაციას;
2.6. დასკვნას სამუშაოს შესახებ.

3. საკონტროლო საკითხები და დავალებები:

3.1. ჩამოთვალეთ პლკ S7-300-ის ძირითადი პარამეტრები;

3.2. როგორია პლკ 300-ის გამოყენების არე?

3.3. პლკ–ს კონფიგურირებისას როგორია სამუშაოთა თანმიმდევრობა?

3.4. როგორ უნდა შესრულდეს MPI-ის მისამართის და შეყვანისა და გამოყვანის მოდულების მისამართის შეცვლა?

3.5. შეიძლება თუ არა შეყვანისა და გამოყვანის მოდულებს ერთნაირი მისამართები ჰქონდეთ?

1.7. ლაბორატორიულ – პრაქტიკული სამუშაო №3

დეცენტრალიზებული პერიფერიის კონფიგურირებაPROFIBUS-ისათვის

სამუშაოს მიზანი: ტიპური ლოგიკური ელემენტების გამოყენებით მართვის პროგრამების შედგენის და რედაქტირების მეთოდების ათვისება.

მოწყობილობა: STEP7 პროგრამული პაკეტი.

თეორიული ცნობები

ავტომატიზაციის სისტემებს ჩვეულებრივი კონფიგურაციით აქვთ კაბელური შეერთებები გადამწოდებთან და შემსრულებელ მოწყობილობებთან. კაბელები უშუალოდ არის ჩასმული ცენტრალური პროგრამული ლოგიკური კონტროლერის შეყვანა/გამოყვანის მოდულებში. ხშირად ეს გულისხმობს გამტართა მნიშვნელოვანი რაოდენობის გამოყენებას.

ვიყენებთ რა დეცენტრალიზებულ კონფიგურაციას, თუ მოვათავსებთ შეყვანა/გამოყვანის მოდულებს გადამწოდებსა და შესრულებულ მოწყობილობებთან სიახლოვეს, შეგვიძლია მნიშვნელოვნად შევამციროთ გამოყენებულ კაბელთა რაოდენობა.

კავშირები პროგრამულ ლოგიკურ კონტროლერს, შეყვანა/გამოყვანის მოდულებსა და საველე მოწყობილობებს შორის წარმოებს PROFIBUS DP-ს დახმარებით.

1. **სამუშაოს შესრულების თანმიმდევრობა:**

   1.1 გაუშვით SIMATIC Manager პროგრამა, შექმენით პროექტი CPU 315-2 DP - ის PROFIBUS ქსელისა და შეყვანა/გამოყვანის ბლოკების და OB1 – ის ერთი ბლოკის გამოყენებით;

   1.2 Hardwar პროგრამის გამოყენებით შეასრულეთ DP მასტერ – სისტემის კონფიგურირება;

   1.3 Hardwar პროგრამის გამოყენებით შეასრულეთ პროექტის აპარატურული ნაწილის კონფიგურირება, ამასთან დააყენეთ დეცენტრალიზებული პერიფერიის შემდეგი ტიპის მოდულები;

№ п/п	მოდულის ტიპი	მარკირება	საიდენტიფიკაციო ნომერი
2	დისკრეტული შეყვანის ბლოკი	SM321 DI16xDC24V	6ES7 321-1BH81-0AA0
4	დისკრეტული გამოყვანის ბლოკი	SM322 DO16xDC24V/0.5A	6ES7 322-1BH00-0AA0
2	დისკრეტული DP შეყვანის ბლოკი	B-16DI	6ES7 131-0BH00-0XB0
4	დისკრეტული DP გამოყვანის ბლოკი	B-16DO	6ES7 132-0BH0-0XB0
4	ინტერფისული მოდული DP	IM153	6ES7 153-IAA00-0XB0
2	დისკრეტული შეყვანის ბლოკი	SM321 DI16xDC24V	6ES7 321-1BH81-0AA0
4	დისკრეტული გამოყვანის ბლოკი	SM322 DO16xDC24V/0.5A	6ES7 322-1BH00-0AA0

1.4 პროექტის შექმნილი კონფიგურაციის გამოყენებით შეასრულეთ პარამეტრების შემდეგი სახის აწყობები;

- შეასრულეთ შემავალი და გამომავალი მოდულების ყველა მისამართის შეცვლა ისეთნაირად, რომ მისამართები იწყებოდეს 124-დან და იყოს უწყვეტი ყველა DP მოდულისათვის;

1.5 შეამოწმეთ შექმნილი კონფიგურაცია კორექტულობაზე და შეინახეთ;

1.6 ჩატვირთეთ კონფიგურაცია კონტროლერის სიმულიატორში;

1.7 დაწერეთ პროგრამა, რომელიც შეასრულებს დისკრეტული DP შეყვანის მოდულის B-16D1 – ის შესასვლელის მდგომარეობის წაკითხვას და SM322 დისკრეტული გამოყვანის მოდულზე მიერთებულ საკონტროლო ნათურების ციმციმს იმ სიხშირით, რაც დამოკიდებული უნდა იყოს შემავალი სიგნალისგან (1,2,3) სიხშირეებთ 0,5 ჰც, 1ჰც და 2ჰც, შესაბამისასდ;

1.8 დანადგარის კონტროლერის მიერთებით დეცენტრალიზებული პერიფერიასთან წაიკითხეთ მისი კონფიგურაცია და განსაზღვრეთ ყველა მოდულის მისამართები.

2. **ანგარიში უნდა შეიცავდეს:**
   2.1. სამუშაოს სახელწოდებას;
   2.2. სამუშაოს მიზანს;
   2.3. ხელსაწყოების ჩამონათვალს;
   2.4. შექმნილ კონფიგურაციას;
   2.5. მოწყობილობის პლკ–სგან წაკითხულ კონფიგურაციას;
   2.6. დასკვნას სამუშაოზე.

3. **საკონტროლო საკითხვები და დვალებები:**
   3.1. ჩამოთვალეთ პლკ S7-300 – ს ძირითადი პარამეტრები;
   3.2. რა შემთხვევაშია გამართლებული დეცენტრალიზებული პერიფერიის გამოყენება?
   3.3. როგორია სამუშაოს თანმიმდევრობა პლკ–ს კონფიგურაციის დროს დეცენტრრალიზებული პერიფერიის გამოყენებისას ?
   3.4. როგორ უნდა შესრულდეს MPI მისამართის და შეყვანა–გამოყვანის DP მოდულების მისამართების შეცვლა?
   3.5. შესაძლებელია თუ არა შეყვანა–გამოყვანის DP მოდულებს ჰქონდეთ ერთნაირი მისამართები?

1.8.ლაბორატორიულ – პრაქტიკული სამუშაო №4

ტექნოლოგიური პროცესების მართვის მარტივი პროგრამების შედგენა S7-300 კონტროლერის გამოყენებით

სამუშაოს მიზანი:ტიპური ლოგიკური ელემენტების გამოყენებით მართვის პროგრამების შედგენის და რედაქტირების მეთოდების ათვისება.

მოწყობილობა: STEP 7 პროგრამების პაკეტი.

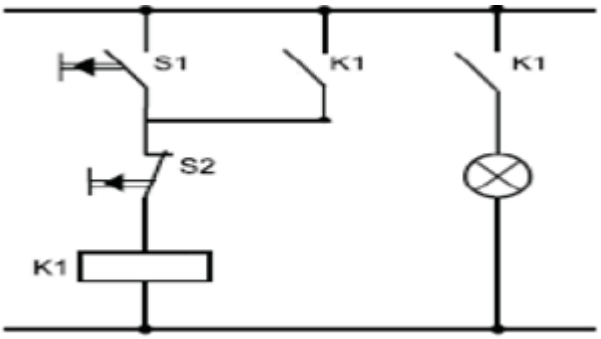
სამუშაოს შესრულების თანმიმდევრობა:

1.1. გაუშვით SIMATIC Manager პროგრამა, შექმენით პროექტი CPU 315-2 DP პროცესორის, შეყვანის გამოყვანის ბლოკების და ერთი OB1 ბლოკის გამოყენებით;

1.2 1-6 დავალებების მიხედვით შეადგინეთ პროგრამა და ემულაციის რეჟიმში გაშვებით შეამოწმეთ პროგრამის შრომისუნარიანობა და მოცემული მუშაობის ალგორითმის შესრულების სისწორე.

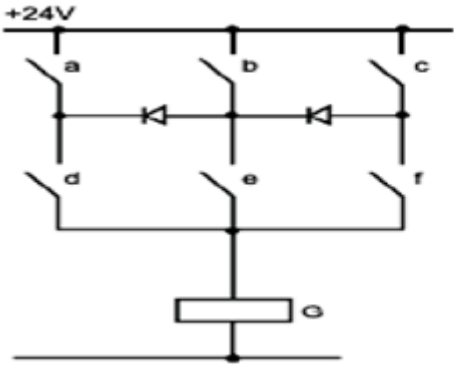
დავალებები:

1. შექმენით პროგრამა, რომელიც განახორციელებს ნახ. 1–ზე ნაჩვენები სქემის მუშაობის ალგორითმს, რომელიც მართავს განათებას ხელით მართვის რეჟიმში;

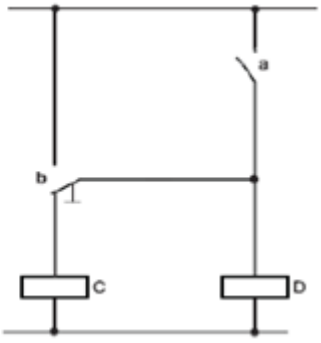


ნახ. 1

2. შექმენით პროგრამა, რომელიც რეალიზებას გაუკეთებს ნახ. 2 და ნახ. 3 სქემების მუშაობის ალგორითმებს და შეცვლის მათ;

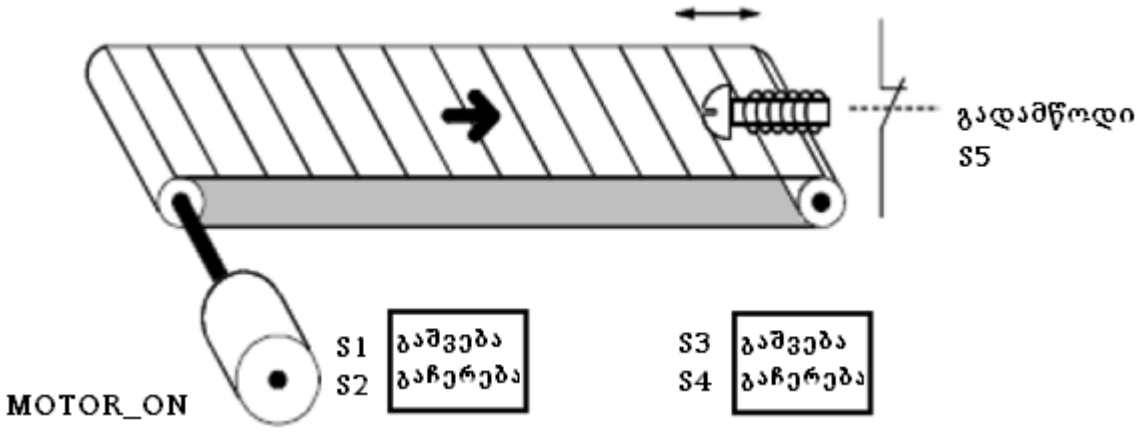


ნახ. 2



ნახ. 3

3. გრძელი გამავალი დერეფნის შესასვლელებთან დაყენებულია B1 და B2 გამომრთველელები, რომლებიც უზრუნველჰყოფს დერეფნის განათების ჩართვას L1 ნათურით. შეადგინეთ პლკ S7-300–ის მართვის პროგრამა. პროგრამამ უნდა უზრუნველყოს გამომრთველების მდგომარეობისგან დამოუკიდებლად შუქის ჩართვა- გამორთვა ნებისმიერი გამოსართავიდან;
4. დაამუშავეთ წყალსაქაჩი ტუმბოს მართვის პროგრამა ავზში წყლის მიწოდების უზრუნველყოფისათვის. ბაკში დაყენებულია ქვედა და ზედა დონეების გადამწოდები. პროგრამამ უნდა უზრუნველყოს ავზში ქვედა და ზედა დონეებს შორის წყლის დონის ავტომატური დაცვა;
5. დაამუშავეთ ტრანსპორტერის ლენტის მართვის პროგრამა, რომელიც შეძლება მოძრაობში მოდიოდეს ელექტროძრავის საშუალებით. ტრანსპორტერის თავში 2 ღილაკია: S1 გაშვებისთვის და S2 გაჩერებისათვის. ტრანსპორტერის ბოლოშიც ორი ღილაკია S3 ამოქმედებისათვის და S4 გაჩერებისთვის. ტრანსპორტერის გაშვება და გაჩერება უნდა ხდებოდეს ნებისმიერი ბოლოდან. აგრათვე, გადამწოდი S5 აჩერებს ტრანსპორტერს, როდესაც ლენტაზე მყოფი საგანი აღწევს ბოლოს.



ნახ. 4

2. ანგარიშიუნდა შეიცავდეს:

2.1. სამუშოს დასახელებას;

2.2. სამუშაოს მიზანს;

2.3. ხელსაწყოების ჩამონათვალს;

2.4. დამუშავებულ სქემებს უკონტაქტო გეგმაში;

2.5. ფარდობითი დამისამართების ცხრილებს თვითოეული პროგრამისათვის;

2.6. დასკვნას სამუშაოს შესახებ.

3. საკონტროლო საკითხები და დავალებები.

3.1. ჩამოთვალეთ პლკ S7-300-ის ძირითადი პარამეტრები;

3.2. როგორია პლკ S7-300-ის გამოყენების არეალი?

3.3. როგორია სამუშაოს თანმიმდევრობა პლკ - სათვის პროგრამების შედგენისას?

1.9. ლაბორატორიულ – პრაქტიკული სამუშაო №5

ტექნოლოგიური პროცესების მართვის პროგრამების შედგენა S7-300 კონტროლერისათვის დროითი დაყოვნებების გამოყენებით

სამუშაოს მიზანი: დროის დაყოვნებების (ტაიმერების) გამოყენებით პროგრამების შედგენის და რედაქტირების მეთოდების ათვისება.
მოწყობილობა: STEP 7პროგრამების პაკეტი.

თეორიული ცნობები

ოპერატორების თანმიმდევრობა ტაიმერის გაშვებისათვის

ტაიმერის ფუნქცია:	მაგალითები:
ტაიმერის განბლოკირება (Enable timer)	A I 16.5 FR T 5
ტაიმერის გაშვება (Start timer)	A I 17.5 L S5T#1s SP T 5
ტაიმერის ჩამოგდება (Reset timer)	A I 18.0 R T 5
ტაიმერის რიცხვითი მნიშვნელობის ამოკითხვა (Dijital timer check)	L T 5 T MW20 LC T 5 T MW22
ტაიმერის ორობითი ამოკითხვა (Binary timer check)	A T 5 = Q 2.0

1. სამუშაოს შესრულების თანმიმდევრობა:

1.1. გაუშვით SIMATIC Manager პროგრამა, შექმენით პროექტი CPU 315-2 DP-ს, შეყვანა გამოყვანის ბლოკებისა და OB1-ს ერთი ბლოკის გამოყენებით;

1.2. 1–4 დავალებების მიხედვით შეადგინეთ პროგრამები და მათი ემულიაციის რეჟიმში გაშვებით შეამოწმეთ პროგრამის შრომისუნარიანობა და დაადგინეთ ასრულებს თუ არა სწორად მუშაობის მოცემულ ალგორითმს.

დავალებები:

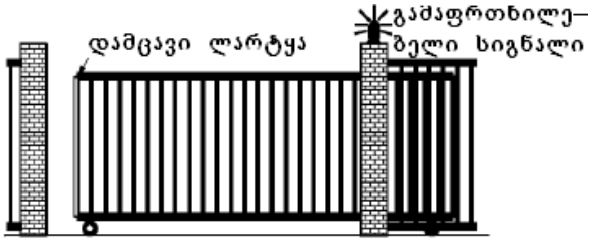
1. შექმენით პროგრამა, რომელიც მმართველი ბრძანების მოსვლის შემდეგ 80 წამის განმავლობაში უზრუნველყოფს სიგნალის გამოცემას;
2. დაამუშავეთ პროგრამა ქარხნის ტერიტორიაზე შესასვლელი კარების მართვისთვის. კარები იმართება ხელით.

მოთხოვნილებები კარების მართვის სისტემისათვის:

- კარები იღება და იკეტება ვახტის სადგომში არსებულ ღილაკებზე თითის დაჭერით. ერთდროულად ვახტიორს შეუძლია გააკონტროლოს კარების მუშაობა სასიგნალო ნათურების მიხედვით. ღილაკებზე ერთდროული დაჭერა არ ცვლის კარების მდგომარეობას;
- კარების გადაადგილების შეწყვეტა შესაძლებელია ნებისმიერ დროს;
- მოციმციმე გამაფრთხილებელი სიგნალი ჩაირთვება 5 წამით ადრე კარების გადაადგილების დაწყებამდე და ჩართული იქნება მანამ, სანამ

კარები იმყოფება მოძრაობაში (საბოლოო მდგომარეობის გადამწოდის ამოქმედებამდე);

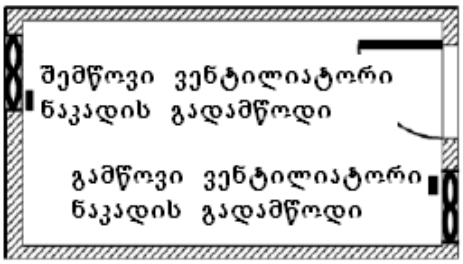
- დამცავი ლარტყა გარანტიას იძლევა, რომ კარების ჩაკეტვისას არავინ არ მიიღებს ტრამვას, არაფერი არ იქნება ჩაჭერილი კარების მიერ ანდა დაზიანებული. მისი კონტაქტების ამუშავების დროს აუცილებლად მოხდება კარების მოძრაობის შეწყვეტა.



ნახ. 1 კარების გარე ხედი

3. დაამუშავეთ პროგრამა საამქროს სავენტილიაციო სისტემის მართვისათვის.
სავენტილიაციო სისტემის დანიშნულება მდგომარეობს იმაში, რომ მიაწოდოს სუფთა ჰაერი საამქროს სათავსში და გაიწოვოს ნაგებობიდან დაგუბებული ჰაერი. მოთხოვნილებები ვენტილიაციის სისტემის მიმართ:

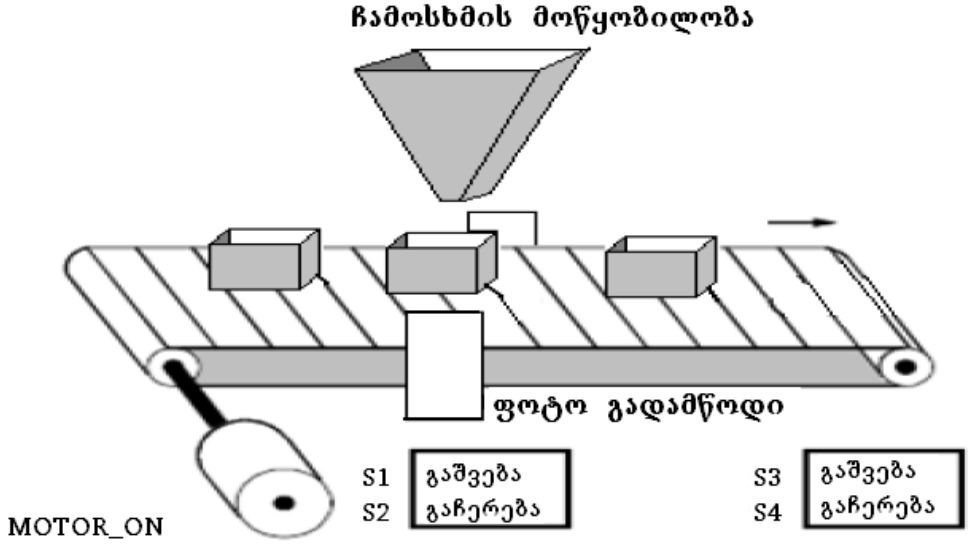
ნაგებობაში არსებობს უვარგისი ჰაერის გამწოვი ვენტილიატორი და სუფთა ჰაერის შემწოვი ვენტილიატორი ნახ. 2. ორივე ვენტილიატორის მუშაობა კონტროლირდება ნაკადის გადამწოდების საშუალებით. ნაგებობაში არასდროს არ უნდა წარმოიშვას ჭარბი წნევა. შემწოვი ვენტილიატორი უნდა ჩაირთოს იმ შემთხვევაში, თუ ნაკადის გადამწოდი იძლევა სიგნალს, რომ გამწოვი ვენტილიატორი მუშაობს საიმედოდ. თუ კი მცირე დაყოვნების შემდეგ ჰაერის ნაკადი არ რეგისტრირდება, მაშინ სისტემა გამოირთვება და გაიცემა შეტყობინება უწესრიგობის შესახებ. თუ კი ერთერთი ვენტილიატორი გამოვა მწყობრიდან მაშინ უნდა ჩაირთოს გამაფრთხილებელი ნათურა.



ნახ. 2 სამქროს ვენტილიაციის სისტემის გარე ხედი

4. დაამუშავეთ პროგრამა უწყვეტი ჩამოსხმის დანადგარის მართვისთვის (ნახ. 3). დანადგარი წარმოადგენს ტრანსპორტერის ლენტას მაზე დამაგრებული ფორმებით. ჩამოსხმის ზონაში ფორმის აღმოჩენისას 60 წამის განმავლობაში წარმოებს გათხევადებული მასის ჩასხმა ფორმებში.
მოთხოვნილებები დანადგარის მართვის სისტემის მიმართ:

- ტრანსპორტერის თავში დაყენებულია ორი ღილაკი: S1 გაშვებისათვის და S2 გაჩერებისათვის. ტრანსპორტერის ბოლოში აგრეთვე დაყენებულია ორი ღილაკი: S3 გაშვებისათვის და S4 გაჩერებისათვის. ტრანსპორტერი შეიძლება გაუშვათ ან გავაჩეროთ ნებისმიერი ბოლოდან. პროგრამის ეს ბლოკი გააფორმეთ როგორც ქვეპროგრამა;
- ფორმის შემოსვლა ჩამოსხმის ზონაში ფიქსირდება ფოტოელექტრული გადამწოდის მიერ;
- ჩამოსხმის პროცესის დამთავრების შემდეგ ტრანსპორტერის მოძრაობა გრძელდება შემდეგი ფორმის შემოსვლამდე;
- ჩამოსხმის ზონაში ფორმის შემოსვლის შეწყვეტის შემთხვევაში 40 წამზე მეტი ხნის განმავლობაში ტრანსპორტერი ჩერდება და გამოსცემს განგაშის სიგნალს.



ნახ. 3 უწყვეტი ჩამოსხმის მოწყობილობა

2. ანგარიში უნდა შეიცავდეს:

2.1. სამუშოს დასახელებას;
2.2. სამუშაოს მიზანს;
2.3. ხელსაწყოების ჩამონათვალს;
2.4. დამუშავებულ პროგრამებსუკონტაქტო გეგმაში;
2.5. ფარდობითი დამისამართების ცხრილებს თვითოეული პროგრამისათვის;
2.6. დასკვნას სამუშაოს შესახებ.

3. საკონტროლო საკითხები და დავალებები:

3.1. რა ალგორითმით ხდება ტაიმერის გამოყენება;
3.2. რა ალგორითმით ხდება მთვლელის გამოყენება;
3.3. პლკ–ს პროგრამის შედგენის დროს სამუშაოთა რა თანმიმდევრობებია შესასრულებელი?

1.10.ლაბორატორიულ – პრაქტიკული სამუშაო №6

ტექნოლოგიური პროცესების მართვის პროგრამების შედგენა S7-300 კონტროლერებისათვის მთვლელების გამოყენებით

სამუშაოს მიზანი: პროგრამების შედგენის და რედაქტირების მეთოდების ათვისება მთვლელების გამოყენებით.

მოწყობილობა: STEP 7პროგრამების პაკეტი.

თეორიული ცნობები

ოპერატორების თანმიმდევრობა მთვლელის გაშვებისათვის

მთვლელის ფუნქცია:	მაგალითები
მთვლელის განბლოკირება (Enable counter)	A I 22.0 FR C 17
პირდაპირი თვლის ფუნქცია (Counter up)	A I 22.1 CU C 17
უკუ თვლის ფუნქცია (counter down)	A I 22.2 CD C 17
მთვლელის დაყენება (Set counter)	A I 22.3 L C#500 S C 17
მთვლელის ჩამოგდება (Reset counter)	A I 22.4 R C 17
მთვლელის რიცხვობრივი მნიშვნელობის შემოწმება (Digital counter check)	L C 17 T MW 30 LC C 17 T MW 32
მთვლელის ორობითი წაკითხვა (Binary counter check)	A C 17 = O 13.0

თვითოეული მთვლელი ხასიათდება შემდეგი პარამეტრებით:

თვლის სიდიდით. თვითოეული მთვლელისთვის დარეზერვირებულია 16 ბიტიანი სიტყვა მონაცემთა სისტემური მეხსიერების სპეციალურ არეში. იგი გამოიყენება ორობით კოდში თვლის სიდიდის შესანახად 0 ... 999 დიაპაზონში.

თვლა მატებაზე. როდესაც CU–ს შესასვლელზე RLO იცვლება ”0”– დან ”1”– ზე, მთვლელის მიმდინარე მნიშვნელობა იზრდება ერთით (ზედა ზღვარი = 999).

თვლა კლებაზე. როდესაც CD–ს შესასვლელზე RLO იცვლება ”0”– დან ”1”– ზე, მთვლელის მიმდინარე მნიშვნელობა მცირდება ერთით (ქვედა ზღვარი = 0).

მთვლელის დაყენება. როდესაც “S” შესასვლელზე RLO იცვლება ”0”– დან ”1”– ზე, მთვლელი იღებს იმ მნიშვნელობას რაც მოცემულია PV შესასვლელზე.

მთვლელის ჩამოგდება. როდესაც სიგნალი ”R” შესასვლელზე ტოლია ”1”–ის მთვლელი დგება ნულოვან მდგომარეობაში. როდესაც მთვლელის ნულზე დაყენების (ჩამოგდების) პირობა სრულდება, მაშინ შეუძლებელია მთვლელის როგორც დაყენება, ასევე თვლის პროცედურა.

RV მთვლელის მნიშვნელობა (0 ... 999) მიეწოდება PV შესასვლელზე:

- როგორც კონსტანტა (C#...)
- როგორც ცვლადი BCD ფორმატში.

B1/BCD მთვლელის მნიშვნელობა შეიძლება ჩატვირთულ იყოს აკუმულიატორში როგორც ორობითი, ასევე BCD რიცხვი, შემდეგ კი გადაცემულ იქნას სხვა მისამართზე.

Q მთვლელის მდგომარეობა შეიძლება შემოწმებულ იყოს ”Q”–ს გამოსასვლელზე:

- მთვლელი = 0 → Q = 0
- მთვლელი > 0 → Q = 1

მთვლელის ტიპები

- C_CU = დამაგროვებელი (თვლა მხოლოდ ”ზევით”)
- C_CD = მაკლებელი (თვლა მხოლოდ ”ქვევით”)
- C_CUD = რევერსიული მთვლელი.

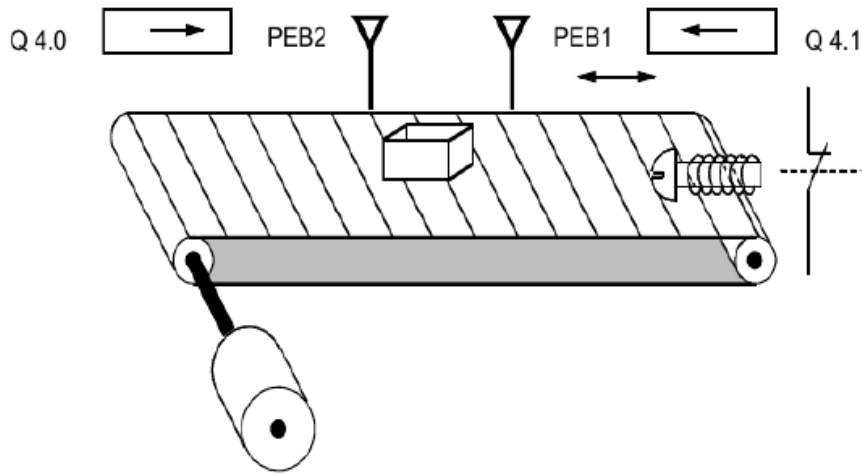
1. სამუშაოს შესრულების თანმიმდევრობა:

1.1. გაუშვით SIMATIC Manager პროგრამა, შექმენით პროექტი CPU 315-2 DP-ს, შეყვანა გამოყვანის ბლოკებისა და OB1-ს ერთი ბლოკის გამოყენებით;

1.2. 1–4 დავალებების მიხედვით შეადგინეთ პროგრამები და მათი ემულიაციის რეჟიმში გაშვებით შეამოწმეთ პროგრამის შრომისუნარიანობა, და დაადგინეთ ასრულებს თუ არა მუშაობის მოცემულ ალგორითმს სწორად.

დავალებები:

1. დაამუშავეთ პროგრამა, რომელიც უზრუნველყოფს ავტომობილების რაოდენობის დათვლას სადგომზე. ავტომობილების გავლა ფიქსირდება ოპტიკური გადამწოდების სიგნალების მიხედვით (შესვლა და გასვლა ცალცალკე).
2. ნახაზზე ნაჩვენებია ტრანსპორტერის ლენტა, რომელიც აღჭურვილია ორი ფოტოელექტრული გადამწოდით (PEB1 და PEB2), ისინი დაპროექტებულია ისეთნაირად, რომ შეუძლია იმ მიმართულების განსაზღვრა საითაც გადაადგილდება პაკეტი ლენტაზე. თვითოეული ფოტოგადამწოდი მუშაობს როგორც ნორმალურად ღია კონტაქტი.



ნახ. 1 ტრანსპორტერის სქემა

3. დაამუშავეთ პროგრამა, რომელიც უზრუნველყოფს ავტომობილების რაოდენობის დათვლას სადგომზე. ავტომობილების გავლა ფიქსირდება ოპტიკური გადამწოდების სიგნალების მიხედვით, რომლებიც განლაგებულია შესასვლელ კარებთან რაღაც მანძილზე. (ავტომობილის გავლის მიმართულება განისაზღვრება ფრონტების გამოჩენის რიგითობით გამომავალ გადამწოდებზე).

4. დაამუშავეთ მოწყობილობის პროგრამა, რომელიც უზრუნველყოფს დეტალების რაოდენობის დათვლას, რომლებმაც გაიარეს დამუშავება ორ სხვადასხვა სამუშაო ადგილზე. გავლა ფიქსირდება ოპტიკური გადამწოდებით. უზრუნველყავით მონაცემთა შენახვის კვების გამორთვის შემთხვევაშიც კი ბრძანების "განულება" მიღებამდე.

2. ანგარიშიუნდა შეიცავდეს:

2.1. სამუშოს დასახელებას;

2.2. სამუშაოს მიზანს;

2.3. ხელსაწყოების ჩამონათვალს;

2.4. დამუშავებულ პროგრამებს;

2.5. ფარდობითი დამისამართების ცხრილებს თვითოეული პროგრამისათვის;

2.6. დასკვნას სამუშაოს შესახებ.

3. საკონტროლო საკითხები და დავალებები:

3.1. რა ალგორითმით ხდება მთვლელის გამოყენება;

3.2. რა ტიპის მთვლელები გამოიყენება და რა არის მათი თავისებურებები?

3.3. პლკ–ს კვების გამორთვის შემთხვევაში როგორ შეინახავთ მთვლელის მონაცემებს?

3.4. პლკ–ს პროგრამის შედგენის დროს რა სახის სამუშაოების თანმიმდევრობაა შესასრულებელი?

1.11. ლაბორატორიულ – პრაქტიკული სამუშაო№7

ტექნოლოგიური პროცესების მართვის პროგრამების შედგენა S7-300 კონტროლერებისათვის შედარებისა და ართმეტიკული ოპერაციების ბრძანებების გამოყენებით

სამუშაოს მიზანი: პროგრამების შედგენისა და რედაქტირების მეთოდების ათვისება შედარებისა და ართმეტიკული ოპერაციების გამოყენებით.

მოწყობილობა: STEP 7 პროგრამების პაკეტი.

თეორიული ცნობები

შედარების ფუნქცია პროგრამირდება შემდეგი სქემის შესაბამისად:

```
მისამართის ჩატვირთვა   Address1;
მისამართის ჩატვირთვა   Address2;
შედარების ფუნქცია;
რეზულტატის მინიჭება    Rezult;

მაგალითად:

L   MW   120;
L   512;
>I;
A   Intut;
=   Output1;
```

აქ შეიძლება გამოიყენებულ იქნას შედარების ინსტრუქცია იმისათვის, რომ შეადაროთ ორი რიცხვი შემდეგ ფორმატებში:

I ორი რიცხვის შედარება Integer;

D ორი რიცხვის შედარება double integer;

R ორი რიცხვის შედარება real (IEEE – ფორმატი მცოცავი მძიმით, 32 ბიტი).

თუ კი შედარების რეზულტატი ჭეშმარიტია ”True”, მაშინ RLO ტოლია “1”. თუ არა, “0”.

IN1 და IN2 შესასვლელებზე მოდებული სიდიდეები შეიძლება შედარდეს ერთმანეთთან შემდეგ პირობებთან შესაბამისობაზე:

==	INI1 ტოლია INI2
<>	INI1 ტოლი არ არის INI2
>	INI1 მეტია INI2
<	INI1 ნაკლებია INI2
>=	INI1 მეტია ან ტოლია IN2
<=	INI1 ნაკლებია ან ტოლია IN2

ართმეტიკული ფუნქციების პროგრამირება ხდება შემდეგი საერთო სქემის მიხედვით:

```
ჩატვირთვა (load)  მისამართის Address1;
ჩატვირთვა (Load)  მისამართის Address2;
ართმეტიკული ფუნქცია;
გადაცემა (transfer)  რეზულტატის Resalt;
```

ოპერაციის რეზულტატი შეინახება აკუმულიატორში Accumulator 1.
მაგალითი: გამოთვალეთ მნიშვნელობა Resultl :=Value1 + Value2 - Value3

```
L       Value1;
L       Value2;
+I;                         //ჯამი  Value1 + Value2
L       Value3;
-I;                         //ჯამი – Value3
T       Result1;
```

1. სამუშაოს შესრულების თანმიმდევრობა:

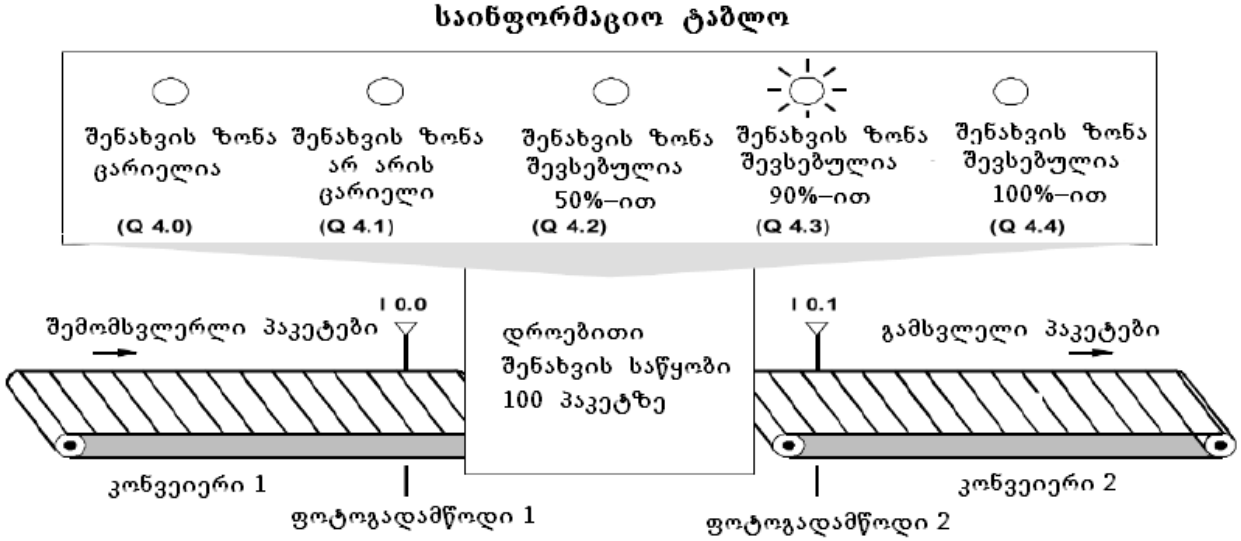
1.1. გაუშვით SIMATIC Manager პროგრამა, შექმენით პროექტი CPU 315-2 DP-ს, შეყვანა გამოყვანის ბლოკებისა და OB1-ს ერთი ბლოკის გამოყენებით;

1.2. 1–4 დავალებების მიხედვით შეადგინეთ პროგრამები და მათი ემულიაციის რეჟიმში გაშვებით შეამოწმეთ პროგრამის შრომისუნარიანობა და დაადგინეთ ალგორითმის შესრულების სისწორე.

დავალებები:

1. შეადგინეთ პროგრამა, რომელიც ახდენს შემდეგი სახის განტოლების ამოხსნას
   MD4 = ((10+25)x15)/5
2. შეადგინეთ პროგრამა, რომელიც ახდენს გადამწოდიდან მიღებული მონაცემების საშუალო მნიშვნელობის გამოთვლას, ტემპერატურის ოცჯერ თანმიმდევრობით გაზომვის შემთხვევაში, რომლებიც განლაგებულნი არიან მეხსიერების თანმიმდევრობით განლაგებულ უჯრედებში დაწყებული MD 12 – დან, მონაცემთა ფორმატია ორმაგი სიტყვა (D).
3. შეადგინეთ პროგრამა, რომელიც ახდენს წნევის გადამწოდიდან მიღებული მონაცემების მასშტაბირებას. მასშტაბირება ხდება მიღებული მონაცემების (რომლებიც განლაგებულია MD 30 – ში) გამრავლებით მასშტაბურ კოეფიციენტ (MD 20) - ზედა მიღებული ნამრავლიდან ნულის დრეიფის სიდიდის გამოკლებით (MD 10).
4. შეადგინეთ პროგრამა, ნახაზზე წარმოდგენილი შენახვის ზონისათვის, რომელიც შედგება სისტემისაგან ორი კონვეიერისა და მათ შორის არსებული დროებითი შენახვის ზონისაგან. კონვეიერი 1 გადაადგილებს პაკეტებს შენახვის ზონისაკენ. ფოტოგადამწოდი, რომელიც განთავსებულია პირველი კონვეიერის ბოლოში შენახვის ზონის მახლობლად განსაზღვრავს, რამდენი პაკეტია შემოსული შენახვის ზონაში.
   კონვეიერი 2 გადაადგილებს პაკეტებს დროებითი შენახვის ზონისაგან ჩასატვირთავი მოედნისაკენ, საიდანაც სატვირთო ავტომობილებს მიაქვთ პაკეტები კლიენტებთან მიწოდებისათვის.
   ფოტოგადამწოდი მე–2 კონტეინერის ბოლოში დროებითი გადაადგილების ზონის მახლობლად განსაზღვრავს, რამდენმა პაკეტმა დატოვა შენახვის ზონა ჩასატვირთ მოედანზე გადატანისთვის.

საინფორმაციო ტაბლო ხუთი ნათურით უჩვენებს დროებითი შენახვის ზონის შევსების დონეს.



ნახ. 2ტექნოლოგიური დანადგარი

2. ანგარიშიუნდა შეიცავდეს:

2.1. სამუშოს დასახელებას;

2.2. სამუშაოს მიზანს;

2.3. ხელსაწყოების ჩამონათვალს;

2.4. დამუშავებულ პროგრამებს;

2.5. ფარდობითი დამისამართების ცხრილებს თვითოეული პროგრამისათვის;

2.6. დასკვნას სამუშაოს შესახებ.

3. საკონტროლო საკითხები და დავალებები:

3.1. რა ალგორითმი გამოიყენება შედარების ოპერაციის დროს?

3.2. რა ალგორითმი გამოიყენება მათემატიკური ოპერაციების დროს?

3.3. რა ტიპის მონაცემები გამოიყენება დასამუშავებელ პროგრამაში?

3.4. როგორ შეინახავთ მონაცემებს პლკ–ს კვების გამორთვის შემთხვევაში ?

3.5. პლკ–ს პროგრამის შედგენის დროს რა სახის სამუშაოების თანმიმდევრობაა შესასრულებელი?

1.12. ლაბორატორიულ – პრაქტიკული სამუშაო №8

ტექნოლოგიური პროცესების მართვის პროგრამების შედგენა S7-300 კონტროლერებისათვის წანაცვლების ბრძანებების გამოყენებით

სამუშაოს მიზანი: წანაცვლების ბრძანებების გამოყენებით პროგრამების შედგენის და რედაქტირების მეთოდების ათვისება

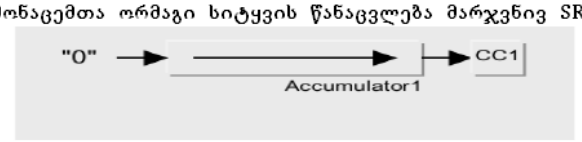
მოწყობილობა: STEP 7 პროგრამების პაკეტი.

თეორიული ცნობები

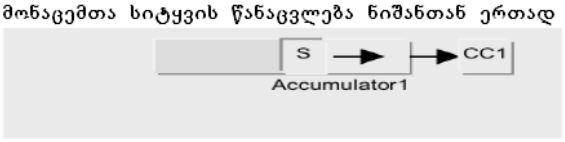
წანაცვლების ფუნქცია:

მონაცემთა სიტყვის წანაცვლება მარჯვნივ SRW

მონაცემთა ორმაგი სიტყვის წანაცვლება მარჯვნივ SRW

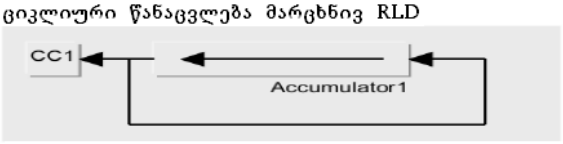


მონაცემთა სიტყვის წანაცვლება ნიშანთან ერთად CC1

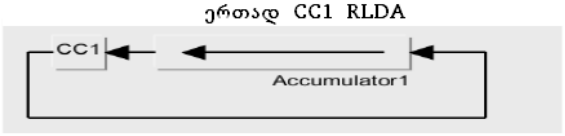


მონაცემთა ორმაგი სიტყვის წანაცვლება ნიშანთან ერთად SSD

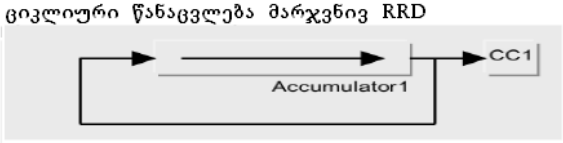
ციკლიური წანაცვლება მარცხნივ RLD



ციკლ. წანაცვლება მარცხნივ მდგომარეობის ბიტთან ერთად CC1 RLDA



ციკლიური წანაცვლება მარჯვნივ RRD



ციკლ. წანაცვლება მარჯვნივ მდგომარეობის ბიტთან ერთად CC1 RRDA

მონაცემთა წანაცვლება Wotrd ფორმატის (სიტყვა)	მნიშვნელობა სიტყვაში MW 130 წანაცვლდება 4 პოზიციით მარცხნივ და შეინახება სიტყვაში MW 132. აქ პოზიციის რიცხვი მოცემულია პარამეტრით ინსტრუქციაში	`L MW 130;`
		`SLW 4;`
		`T MW 132;`

მონაცემთა წანაცვლება DuobleWord ფორმატის (ორმაგი სიტყვა)	მნიშვნელობა ცვლადში "ShiftOn" წანაცვლდება მარჯვნივ "ShiftPos" პოზიციაზე და შეინახება "ShiftOf" ცვლადში. აქ პოზიციის რიცხვი მოცემულია აკუმულიატორით accumulator 2.	`L "Global_DB".ShiftPos;`
		`L "Global_DB".ShiftOn;`
		`SRD ;`
		`T "Global_DB".ShiftOff;`

წანაცვლება ნიშნით	ცვლადის #Actval ნიშნით მარჯვნივ 2 პოზიციაზე და გადაცემა ცვლადში #Display.	`L #Actval;`
		`SSI 2;`
		`T #Display;`

1. სამუშაოს შესრულების თანმიმდევრობა:

1.1. გაუშვით SIMATIC Manager პროგრამა, შექმენით პროექტი CPU 315-2 DP-ს, შეყვანა გამოყვანის ბლოკებისა და OB1-ს ერთი ბლოკის გამოყენებით;

1.2. 1–3 დავალებების მიხედვით შეადგინეთ პროგრამები და მათი ემულიაციის რეჟიმში გაშვებით შეამოწმეთ პროგრამის შრომისუნარიანობა და დაადგინეთ ალგორითმის შესრულების სისწორე.

დავალებები:

1. დაამუშავეთ პროგრამა, რომელიც ასრულებს IB 10 მისამართისგან მიღებული მონაცემების ციკლურ წანაცვლებას.
2. დაამუშავეთ პროგრამა, რომელიც ასრულებს IB 10 მისამართისგან მიღებული მონაცემების წანაცვლებას 5 პოზიციით მარჯვნივ და გადასცემს მიღებულ რეზულტატს MB 20-ში.
3. დაამუშავეთ პროგრამა, რომელიც ასრულებს მორბენალი ნათების ფუნქციას. გაითვალისწინეთ მორბენალი ნათების სამი სხვადასხვა კომბინაციის რეალიზაციის შესაძლებლობა.

2. ანგარიშიუნდა შეიცავდეს:

2.1. სამუშოს დასახელებას;

2.2. სამუშაოს მიზანს;

2.3. ხელსაწყოების ჩამონათვალს;

2.4. დამუშავებულ პროგრამებს;

2.5. ფარდობითი დამისამართების ცხრილებს თვითოეული პროგრამისათვის;

2.6. დასკვნას სამუშაოს შესახებ.

3. საკონტროლო საკითხები და დავალებები:

3.1. რა ალგორითმი გამოიყენება წანაცვლების ოპერაციის დროს?

3.2. რა ტიპის მონაცემები გამოიყენება დასამუშავებელ პროგრამაში?

3.3. პროგორ შეინახავთ მონაცემებს პლკ–ს კვების გამორთვის შემთხვევაში ?

3.4. პლკ–ს პროგრამის შედგენის დროს რა სახის სამუშაოების თანმიმდევრობაა შესასრულებელი?

1.13.ლაბორატორიულ – პრაქტიკული სამუშაო №9

ტექნოლოგიური პროცესების მართვის პროგრამების შედგენა S7-300 კონტროლერებისათვის ანალოგური მმართველი სიგნალების გამოყენებით

სამუშაოს მიზანი: ანალოგური მმართველი სიგნალების გამოყენებით პროგრამების შედგენის და რედაქტირების მეთოდების ათვისება.

მოწყობილობა: STEP 7 პროგრამების პაკეტი.

თეორიული ცნობები

ანალოგური სიგნალების გამოყვანის მოდულების დანიშნულებაა კონტროლერის შიგა ციფრული სიდიდეების ციფრულ–ანალოგურ ფორმებში გარდაქმნა და შესაბამისი ანალოგური გამოსასვლელი სიგნალების ფორმირება.

ანალოგიური სიდიდე დენი/ძაბვა ნომინალური დიაპაზონით (იგი აირჩევა მოდულის პარამეტრირების დროს უტილიტით HW config) გარდაიქმნება მოდულში რიცხვად 0–დან +27648 დიაპაზონში (დენი/ძაბვის სიმეტრიული მნიშვნელობებისათვის, მაგ. ±10 ვოლტისთვის ეს დიაპაზონი ტოლი იქნება მინუს 27648 – დან პლიუს 27648 – მდე). თუ კი ანალოგიური სიდიდე აჭარბებს ნომინალურ მნიშვნელობას, მაშინ ანალოგურმა მოდულმა შეიძლება გამოსცეს მნიშვნელობა 27648 – დან 32767–მდე. ამ დროს მიღებულია, რომ 32767–ის მნიშვნელობის დროს ადგილი აქვს გადავსებას.

მასშტაბირება

მასშტაბირება ეს რიცხვითი მნიშვნელობის მათემატიკური გარდაქმნაა ფიზიკური დიაპაზონის მნიშვნელობაში. უნიპოლიარული სიგნალებისთვის მასშტაბირება ხდება ფორმულით:

OUT = [(FLOAT(IN)/27648*(HI_LIM-LO_LIM)] + LO_LIM,

სადაც: **IN** - შესასვლელი ანალოგიური სიდიდის რიცხვითი მნიჭვნელობაა;

HI_LIM-LO_LIM – ზედა და ქვედა ზღვრებია ფიზიკური დიაპაზონისათვის.

<u>აუცილებელი სიზუსტის უზრუნველსაყოფად ყველა გამოთვლები უნდა შესრულდეს რიცხვებში ფორმატით Real.</u>

1. სამუშაოს შესრულების თანმიმდევრობა:

1.1. გაუშვით SIMATIC Manager პროგრამა, შექმენით პროექტი CPU 315-2 DP-ს, ანალოგიური შეყვანის SM331, ანალოგიური გამოყვანის SM332 ბლოკებისა და OB1-ს ერთი ბლოკის გამოყენებით;

1.2. 1–5 დავალებების მიხედვით შეადგინეთ პროგრამები და მათი ემულიაციის რეჟიმში გაშვებით შეამოწმეთ პროგრამის შრომისუნარიანობა და დაადგინეთ ალგორითმის შესრულების სისწორე.

დავალებები:

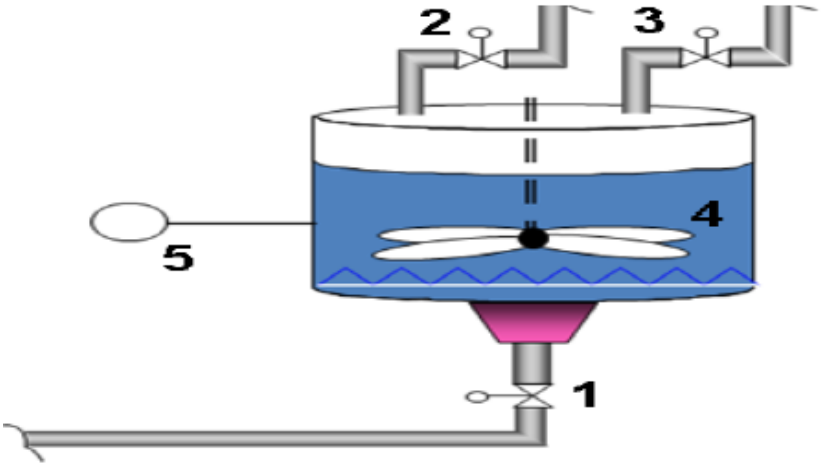
1. დაამუშავეთ პროგრამა, რომელიც ასრულებს ხსნარის მომზადებას ორი სითხისაგან მოცემული პროცენტული თანაფარდობის მიხედვით.

დანადგარის მუშაობის ალგორითმი:

- იღება მე–2 ონკანი, იწყება ავზის შევსება სითხით 1. სითხე 1 ავსებს ავზს 30%–ით;
- იკეტება მე–2 ონკანი;

- იღება მე–3 ონკანი, იწყება ავზის შევსება სითხით 2. სითხე 2 ავსებს ბაკს 70%–მდე;
- იკეტება მე–3 ონკანი;
- 30 წამით ჩაირთვება ფრიალა 4 და ნარევი გადაერევა;
- გაიღება 1–ლი ონკანი და ხდება ნარევის გადაცლა. გადაცლა წარმოებს ავზის სრულ დაცლამდე;
- ჩაიკეტება 1–ლი ონკანი;
- პროცესი განმეორდება.

სიხის დონის გაზომვა ბაკში ხდება დონის ანალოგური გადამწოდით.



ნახ. 3ტექნოლოგიური დანადგარი

2. დაამუშავეთ პროგრამა იმ მოწყობილობისათვის, რომელიც ახდენს გადამწოდიდან მიღებული სიგნალის გადამეტების შესახებ სიგნალიზაციას იმ მნიშვნელობაზე, რომელიც დაყენებულია ნიშნულზე. სიგნალისა და ნიშნულის სიგნალების ცვლილების დიაპაზონია 0 – 10ვ. სხვაობის სიდიდე უნდა შეინახებოდეს MD10-ში.

3. დაამუშავეთ პროგრამა იმ მოწყობილობისათვის, რომელიც უზრუნველყოფს საცავში სითხის დონის ინდიკაციას. ინდიკაცია ხდება დისკრეტული 5 თანრიგა ინდიკატორით. სიგნალი სითხის დონის გამზომიდან მიეწოდება 0–10ვ სიგნალის სახით.

4. დაამუშავეთ პროგრამა იმ მოწყობილობისათვის, რომელიც უზრუნველყოფს თერმოსტატის გამაცხელებელი ელემენტის მართვას. თერმოსტატი არეგულირებს სითხის ტემპერატურას +10°–იდან +90°–მდე. ინფორმაცია ტემპერატურის გადამწოდიდან გამოდის ძაბვის სახით, გადამწოდის მგრძნობიარობაა S=2ვ/გრადუსზე. გაითვალისწინეთ ხელით მართვის შესაძლებლობა.

5. დაამუშავეთ კომპონენტების ერთმანეთში შერევის სისტემის მართვის პროგრამა. სისტემამ უნდა იმუშაოს შემდეგნაირად:

- 1500 ლიტრის საერთო მოცულობის საცავი 40%–ით ივსება 1–ლი კომპონენტით (ბრძანება 1), სათავსის დარჩენილი ნაწილი ივსება მე–2 კომპონენტით (ბრძანება 2). ინფორმაცია სათავსის შევსების შესახებ გამოდის ძაბვის სახით გადამწოდიდან;
- შემდეგ ნარევი გადაერევა ერთმანეთში 30 წამის განმავლობაში ფრიალას საშუალებით (ბრძანება 3);

- შემდეგ ნარევი გადმოიცლება ბუნკერიდან (ბრძანება 4);
- გაითვალისწინეთ ხელით მართვის შესაძლებლობა.

2. ანგარიშიუნდა შეიცავდეს:

2.1. სამუშოს დასახელებას;

2.2. სამუშაოს მიზანს;

2.3. ხელსაწყოების ჩამონათვალს;

2.4. დამუშავებულ პროგრამებს;

2.5. ფარდობითი დამისამართების ცხრილებს თვითოეული პროგრამისათვის;

2.6. დასკვნას სამუშაოს შესახებ.

3. საკონტროლო საკითხები და დავალებები:

3.1. როგორ სრულდება მონაცემთა მასშტაბირება?

3.2. როგორ ხდება ანალოგიური შესასვლელებისა და გამოსასვლელების კონფიგურირება?

3.3. როგორ ხდება ანალოგიური შესასვლელისგან მონაცემთა წაკითხვა?

3.4. როგორ ხდება მონაცემთა გაცემა ანალოგიური გამოსასვლელის გამოყენებით?

1.14. ლაბორატორიულ – პრაქტიკული სამუშაო №10

S7-300 კონტროლერებისათვის ტექნოლოგიური პროცესების მართვის პროგრამების შედგენა ცვლადების შენახვებისათვის მონაცემთა ბლოკების გამოყენებით

სამუშაოს მიზანი: ცვლადების შენახვისათვის მონაცემთა ბლოკების გამოყენებით პროგრამების შედგენის და რედაქტირების მეთოდების ათვისება.
მოწყობილობა: STEP 7 პროგრამების პაკეტი.

თეორიული ცნობები

ბლოკის გამოყენებამდე საჭიროა "გავაღოთ მონაცემთა ბლოკი" (OPN) ბრძანების შესრულებით, როგორც ერთობლივად გამოსაყენებელი (გლობალური) მონაცემთა ბლოკისთვის, ასევე მონაცემთა ბლოკის ეკზემპლიარისათვის.

მონაცემთა ბლოკებთან მუშაობისათვის გამოიყენება შემდეგი სახის ინსტრუქციები:

- OPN გააღეთ მონაცემთა ბლოკი;
- CDB DB და DI მონაცემთა ბლოკების რეგისტრების გაცვლა;
- LDBLG გლობალური მონაცემთა ბლოკის სიგრძის ჩატვირთვა ACCU 1-ში;
- LDBNO გლობალური მონაცემთა ბლოკის ნომრის ჩატვირთვა ACCU 1-ში;
- LGILG ეკზეპლიარული მონაცემთა ბლოკის სიგრძის ჩატვირთვა ACCU 1-ში;
- LDINO ეკზეპლიარული მონაცემთა ბლოკის ნომრის ჩატვირთვა ACCU 1-ში.

1. სამუშაოს შესრულების თანმიმდევრობა:

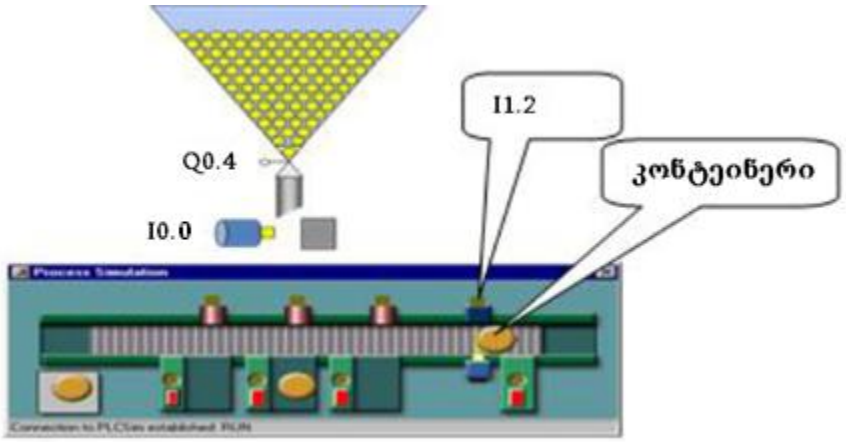
1.1. გაუშვით SIMATIC Manager პროგრამა, შექმენით პროექტი CPU 315-2 DP-ის, ანალოგიური შეყვანისა SM331, ანალოგიური გამოყვანის SM332 ბლოკებისა და OB1-ს ერთი ბლოკის გამოყენებით;

1.2. 1–5 დავალებების მიხედვით შეადგინეთ პროგრამები რომლებშიც მონაცემები შენახულია DB90 ბლოკში და მათი ემულიაციის რეჟიმში გაშვებით შეამოწმეთ პროგრამის შრომისუნარიანობა და დაადგინეთ მოცემული ალგორითმის შესრულების სისწორე.

დავალებები:

1. დაამუშავეთ პროგრამა, რომელიც ასრულებს DB10 ბლოკის გახსნას, მე – 14 ბაიტის წაკითხვას, მის ციკლურ წანაცვლებას 3 თანრიგზე მარცხნივ და რეზულტატის მოთავსებას DB12–ის მე – 4 ბაიტში. (ბაიტ 14–ის მნიშვნელობა DB10-ში ჩაწერეთ წინასწარ).
2. შექმენით პროგრამა, რომელიც შეასრულებს ორმაგი სიტყვის წაკითხვას DB12–დან (წინასწარ შექმენით ეს სიტყვა) Real ფორმატში, გაამრავლეთ იგი 10,3 და მოახდინეთ ორმაგი სიტყვის გამოყვანა QD0-ში.
3. დაამუშავეთ პროგრამა შემფუთავი კონვეიერული ხაზის მართვისათვის (იხ. ნახაზი 4). I 1.2 გადამწოდიდან სიგნალის დადებითი ფრონტის შემთხვევაში კონვეიერი უნდა გაჩერდეს და უნდა გაიღოს ქურო Q 0.4. დაიწყება ტაბლეტების ცვენა მილიდან

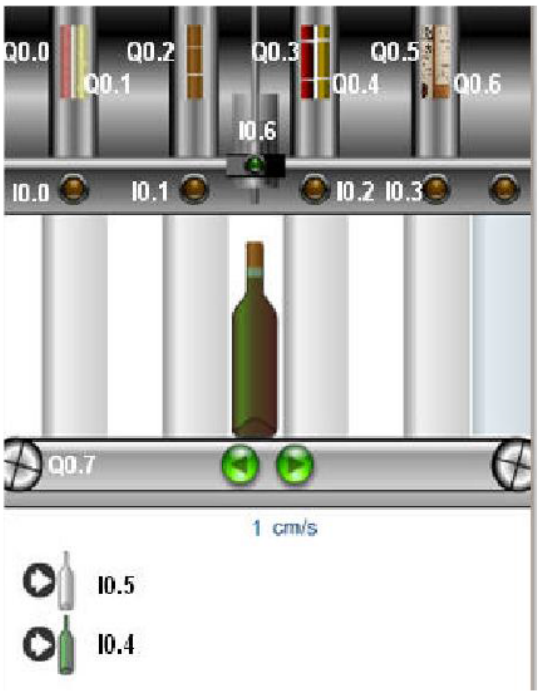
კონტეინერში, მათი დათვლა წარმოებს I 0.0 ფოტოელემენტით. 50 ტაბლეტის დათვლის შემდეგ Q 0.4 ქურო უნდა დაიკეტოს და კონვეიერმა ისევ უნდა გააგრძელოს მოძრაობა. პროგრამით უზრუნველყავით მონაცემთა ბლოკში ინფორმაციის შენახვა კონვეიერის მუშაობის რეჟიმის შესახებ (ჩართვა–გამორთვა), აგრეთვე შეფუთული ტაბლეტებისა და კონტეინერების საერთო რაოდენობის შესახებ.



ნახ. 4 ტექნოლოგიური დანადგარი

4. დაამუშავეთ პროგრამა ღვინის ჩამოსხმის დანადგარისთვის (ნახ. 5). ჩამოსხმისათვის მიეწოდება ნათელი და მუქი ბოთლები. არჩევა განისაზღვრება შესაბამისი გენერატორის მიერ ანდა თაგვით ერთერთ ღილაკზე დაჭერით ქვევიდან.
ბოთლის შემოსვლას იმ პოზიციაში, სადაც უნდა მოხდეს 4–დან ერთერთი ტექნოლოგიური ოპერაცია (ჩამოსხმა, მოხუფვა, ეტიკეტის მიწებება ყელზე, ეტიკეტის მიწებება ბოთლზე), თანა სდევს შესაბამისი გადამწოდების გააქტიურება (I0.0 – I0.3). თვითოეული ტექნოლოგიური ოპერაციის დამთავრების სიგნალიზაცია ხდება ერთი საერთო გადამწოდით I0.6.
მოთხოვნილებები:

- ნათელი ბოთლი უნდა აივსოს თეთრი ღვინით, უნდა დაიხუფოს საცობით და მიეწებოს ეტიკეტები თეთრი ღვინისათვის;
- მუქი ბოთლი უნდა შეივსოს წითელი ღვინით, უნდა დაიხუფოს საცობით და მიეწებოს ეტიკეტები წითელი ღვინისათვის;
- ტექნოლოგიური ოპერაციის შესრულების დროს კონვეიერი უნდა იქნას გაჩერებული;
- ახალი ოპერაცია შეიძლება დაწყებულ იქნას მხოლოდ წინა ოპერაციის დამთავრების შემდეგ;
- დამატებით: გაითვალისწინეთ თეთრი და წითელი ღვინის ბოთლების გადათვლა;
- უზრუნველყავით ინფორმაციის შენახვა დანადგარის მუშაობის რეჟიმის შესახებ, სითხის რაოდენობის შესახებ ავზში.



შესასვლელები

I0.0	ბოთლი სადგურში: შევსება
I0.1	ბოთლი სადგურში: საცობი
I0.2	ბოთლი სადგურში: ეთიკეთი ყელზე
I0.3	ბოთლი სადგურში: ეთიკეტი
I0.4	მუქი ბოთლის გადამწოდი
I0.5	ღია ბოთლის გადამწოდი
I0.6	მიმდინარე ოპერაცია შესრულებულია

გამოსასვლელები

Q0.0	შეავსეთ წითელი ღვინით
Q0.1	შეავსეთ თეთრი ღვინით
Q0.2	დახურეთ საცობით
Q0.3	დააწებეთ ეთიკეთი ბოთლის ყელზე წითელი ღვინით
Q0.4	დააწებეთ ეთიკეტი ბოთლის ყელზე თეთრი ღვინით
Q0.5	დააწებეთ ეთიკეტი ბოთლზე წითელი ღვინით
Q0.6	დააწებეთ ეთიკეტი ბოთლზე თეთრი ღვინით
Q0.7	გამორთეთ კონვეიერი

ნახ. 5 ტექნოლოგიური დანადგარი

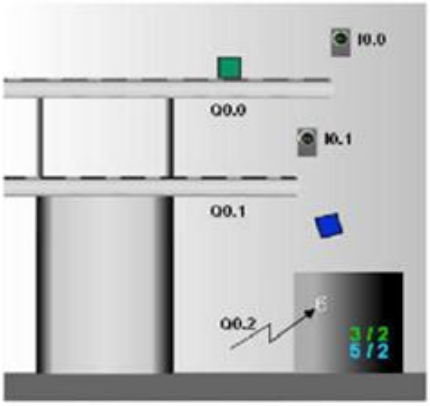
5. შემფუთი დანადგარი (ნახ. 6) ფუთავს მწვანე და ლურჯ კუბიკებს ყუთებად. მწვანე კუბიკები ავტომატურად მიეწოდება ზედა კონვეიერს თუ კი ის ჩართულია. ლურჯი კუბიკები ავტომატურად მიეწოდება ქვედა კონვეიერს, თუ ისიც ჩართულია. თვითოეულ ყუთში აუცილებელია განთავსდეს თვითოეული ფერის მკაცრად განსაზღვრული კუბიკების რაოდენობა. თუ კი ეს მოთხოვნა არ სრულდება, მაშინ ზედმეტი კუბიკები ვარდება ყუთიდან. აუცილებელია აგრეთვე, რომ ისე მოხდეს კუბიკების შეფუთვა, რომ არცერთი მათგანი არ დაიკარგოს, ანუ არ გამოვარდეს ყუთიდან.

მოთხოვნები:

- თვითოეულ ყუთში უნდა მოხვდეს 5 ლურჯი და 3 მწვანე კუბიკი;
- როგორც კი გარკვეული ფერის კუბიკების აუცილებელი რაოდნობა შემოვა ყუთში, მაშინ მისი შესაბამისი კონვეიერი ჩერდება;
- მორიგი ყუთის შევსების შემდეგ შემოდის სიგნალი (Q0.2) ახალი ყუთის დაყენებაზე და ორივე კონტეიერი ჩაირთვება თავიდან (Q0.0, Q0.1).

დამატებით:

- გაითვალისწინეთ თვითოეული ფერის კუბიკის გადათვლა და ინფორმაციის განლაგება DB –ში;
- ყუთების რაოდენობის მოცემული რიცხვის მიღწევისას, მაგ. 12, შეწყვიტეტთ კუბიკების შეფუთვა.



გამოსასვლელები
Q0.0 ჩართერთ ზედა კონვეიერი
Q0.1 ჩართეთ ქვედა კონვეიერი
Q0.2 დააყენეთ ახალი ყუთი

შესასვლელები
I0.0 კუბიკი (მწვანე) ზედა კონვეიერზე
I0,1 კუბიკი (ლურჯი) ქვედა კონვეიერზე
I0.2 ყუთი არ არის შევსებული

ნახ. 6 შემფუთი დანადგარი

2. ანგარიშიუნდა შეიცავდეს:

2.1. სამუშოს დასახელებას;

2.2. სამუშაოს მიზანს;

2.3. ხელსაწყოების ჩამონათვალს;

2.4. დამუშავებულ პროგრამებს;

2.5. ფარდობითი დამისამართების ცხრილებს თვითოეული პროგრამისათვის;

2.6. დასკვნას სამუშაოს შესახებ.

3. საკონტროლო საკითხები და დავალებები:

3.1. რისთვის გამოიყენება მონაცემთა ბლოკები?

3.2. რა თამიმდევრობით ხდება მონაცემთა ბლოკების შექმნა?

3.3. როგორ ხდება ბლოკის მონაცემების დამისამართება?

3.4. როგორი თანმიმდევრობით უნდა იქნას გამოყენებული პროგრამაში მონაცემები ბლოკებიდან ?

1.14. ლაბორატორიულ – პრაქტიკული სამუშაო №11
ტპ–ის მართვის პროგრამების შედგენა S7-300 კონტროლერებისათვის FC და FB ბლოკების გამოყენებით

სამუშაოს მიზანი: FC და FB ბლოკების გამოყენებით პროგრამების შედგენის და რედაქტირების მეთოდების ათვისება.
მოწყობილობა: STEP 7პროგრამების პაკეტი.

თეორიული ცნობები

დიდ სისტემებში ხშირად გამოყენებული ფუნქციებისათვის იქმნება უნივერსალური ბლოკები FC და FB წინასწარ დანიშნული პარამეტრებით (პარამეტრირებადი ბლოკები). მათ აქვთ ფორმალური შემავალი და გამომავალი პარამეტრები, რომლებსაც დაენიშნებათ ფაქტიური პარამეტრები ბლოკის გამოძახების დროს. პროგრამული ბლოკის ადაპტაცია მართვად მოწყობილობასთან ამ შემთხვევაში შედგება პარამეტრების მიკუთვნებასთან ბლოკის გამოძახების დროს, ამასთან ამ დროს ბლოკის პროგრამა არ იცვლება.

პარამეტრირებადი ბლოკები გამოიყენება პროგრამის ცალკეული ფრაგმენტების ხშირად გამოყენების აუცილებლობის შემთხვევაში. ასეთი ბლოკების გამოყენება იძლევა შემდეგ უპირატესობას:

- პროგრამის ფრაგმენტის დაწერა ხდება მხოლოდ ერთჯერ;
- ასეთი ბლოკი იკავებს მხოლოდ მისთვის ერთჯერადად განსაზღვრულ მონაკვეთს მომხმარებლის მეხსიერებაში, თუმცა იგი პროგრამაში შეიძლება გამოყენებულ იქნას მრავალჯერ;
- ბლოკი პროგრამირდება ფორმალური პარამეტრებით (ატრიბუტებით input (შემავალი), output (გამომავალი) ან in/out (შემ/გამ)), რომლებსაც მიენიჭებათ ფაქტიური მისამართები (ფაქტიური პარამეტრები) მხოლოდ და მხოლოდ ბლოკის გამოძახების შემთხვევაში.

ფუნქციებისაგან (FC), განსხვავებით ფუნქციონალურ ბლოკებს (FB) აქვთ აგრეთვე მეხსიერება. ეს ნიშნავს იმას, რომ ამ ფუნქციონალურ ბლოკს დაენიშნება ლოკალური მონაცემების განსაკუთრებული ბლოკი (**ეკზემპლიარი DB**). FB ბლოკის გამოძახების შემთხვევაში ავტომატურად გაიხსნება მონაცემთა ბლოკი განსაზღვრული ნომრით (DB-ს ეკზეპლიარი).

DB ეკზეპლიარი გამოიყენება **სტატიკური ცვლადების** შენახვისათვის. ეს ლოკალური ცვლადები შეიძლება გამოყენებულ იქნას მხოლოდ FB-ში, ცვლადების გამოცხადების ცხრილში, სადაც ისინი გამოცხადებულია. ბლოკის დამუშავების დამთავრების შემდეგ ეს ცვლადები შეინახება.

1. სამუშაოს შესრულების თანმიმდევრობა:

1.1. გაუშვით SIMATIC Manager პროგრამა, შექმენით პროექტი CPU 315-2 DP-ს, ანალოგიური შეყვანის SM331, ანალოგიური გამოყვანის SM332 ბლოკებისა და OB1-ს ერთი ბლოკის გამოყენებით;

1.2. 1–3 დავალებების მიხედვით შეადგინეთ პროგრამები FC და FB ბლოკების გამოყენებით და მათი ემულიაციის რეჟიმში გაშვებით შეამოწმეთ პროგრამის შრომისუნარიანობა და დაადგინეთ მოცემული ალგორითმის შესრულების სისწორე.

დავალებები:

1. FC-ს გამოყენებით დაამუშავეთ პროგრამა, რომელიც გამოთვლის შემდეგ ფუნქციას მონაცემთა სამი სხვადასხვა ვარიანტისათვის:

$Y=(X_1 + X_2 + X_3)/3$

მონაცემები განლაგებულია მერკერულ მეხსიერებაში დაწყებული MW20–დან, რეზულტატი თავსდება იმავე მერკერულ მეხსიერებაში დაწყებული MW40.

2. დაამუშავეთ პროგრამა, რომელიც შეასრულებს მორბენალი ცეცხლის ფუნქციას. გაითვალისწინეთ მორბენალი ცეცხლის სამი სხვადასხვა კომბიმაციის ავტომატური მონაცვლეობა.

3. დაამუშავეთ პროგრამა, რომელიც ასრულებს ხსნარის მომზადებას ორი სხვადასხვა სითხისგან, მოცემული პროცენტული თანაფარდობის მქონე ორი სხვადასხვა რეცეპტით. დანადგარის მუშაობის ალგორითმი:

- იღება ონკანი 2, იწყება ავზის შევსება სითხით 1. ეს სითხე ავსებს ავზს პირველი რეცეპტისათვის 30%–ით, მეორე რეცეპტისათვის 55%–ით;
- ონკანი 2 იკეტება;
- იღება ონკანი 3, იწყება ავზის შევსება სითხით 2. ეს სითხე ავსებს ავზს პირველი რეცეპტისათვის 70%–ით, მეორე რეცეპტისათვის 55%–ით;
- ონკანი 3 იკეტება;
- ჩაირთვება ფრიალა 4 და იწყება ნარევის გადარევა პირველი რეცეპტისათვის – 40 წმ–ის განმავლობაში, მეორე რეცეპტისათვის 10 წამის განმავლობაში;
- იღება ონკანი 1 და იწყება ნარევის გადმოსხმა. გადმოსხმა წარმოებს ავზის სრულ დაცლამდე;
- ონკანი 1 იკეტება;
- პროცესი მეორდება;
- სითხის დონის გაზომვა ავზში, რომლის მოცულობაა 5000 ლიტრი წარმოებს დონის ანალოგიური გადამწოდის დახმარებით;
- გაითვალისწინეთ რეცეპტების ხელით გადართვის შესაძლებლობა.
-

2. ანგარიშიუნდა შეიცავდეს:

2.1. სამუშოს დასახელებას;

2.2. სამუშაოს მიზანს;

2.3. ხელსაწყოების ჩამონათვალს;

2.4. დამუშავებულ პროგრამებს;

2.5. ფარდობითი დამისამართების ცხრილებს თვითოეული პროგრამისათვის;

2.6. დასკვნას სამუშაოს შესახებ.

3. საკონტროლო საკითხები და დავალებები:

3.1. რისთვის გამოიყენება FC და FB ბლოკები;

3.2. რა თამიმდევრობით ხდება FC და FB ბლოკების შექმნა?

3.3. როგორ ხდება აღნიშნული ბლოკების დამისამართება?

3.4. როგორი თანმიმდევრობით გამოიყენება FC და FB ბლოკები პროგრამაში?

თავი 2

ამ თავში განხილული მეთოდური მითითებები გათვალისწინებულია იმ სტუდენტებისათვის, რომლებიც გადიან მომზადებას ხელსაწყოთამშენებლობის, ავტომატიზაციის საშუალებებისა და მართვის სისტემების შესაბამისი საგანმანათლებლო პროგრამებით. იგი დიდ დახმარებას გაუწევს სტუდენტებს საგნების "ტექნოლოგიური პროცესების და წარმოებების ავტომატიზაცია", "საინფორმაციო სისტემები და ტექნოლოგიები", "ელექტროამძრავები და სამრეწველო დაბადგარებისა და ტექნოლოგიური კომპლექსების ავტომატიკა", ინფორმაციის დამუშავებისა და მართვის ავტომატიზირებული სისტემები" ათვისებაში.

2.1. შესავალი

ტექნოლოგიური პროცესებისა და წარმოებების ავტომატიზაცია წარმოადგენს წარმოების ეფექტურობის გაზრდის, ეკოლოგიური უსაფრთხოების და წარმოების უავარიობის უზრუნველყოფისათვის ერთერთ ძირითად მიმართულებას.

ტექნოლოგიური პროცესების მართვის ავტომატიზირებული სისტემები (ტპ მას) უზრუნველყოფს ნედლეულისა და ენერგიის ხვედრითი ხარჯების შემცირებას, ტექნოლოგიური მოწყობილობების რესურსის გაზრდას სამრეწველო რესურსების ოპტრიმიზაციის ხარჯზე, ავარიული სიტუაციების წინასწარ შეტყობინებას, აგრეგატების რაციონალურ დატვირთვას.

თანამედროვე ტპ მას წარმოადგენს განაწილებულ, იერარქიულ მართვის სისტემას. ასეთი სისტემების საფუძვლად, მართვის იერარქიის დაბალ საფეხურებზე გამოიყენება მიკროპროცესორული მოწყობილობები, რომლებიც ახორციელებს ინფორმაციის შეგროვებასა და დამუშავებას, ხოლო მართვის იერარქიის მაღალ საფეხურებზე – სამუშაო სადგურები მათზედ დაყენებული SCADA სისტემებით .

წინამდებარე ლაბორატორული პრაქტიკუმი შეიცავს სამუშაოთა ციკლს ელექტრული გამაცხელებელი ღუმელის ავტომატური მართვის სისტემის აგებისათვის სამრეწველო მიკროპროცესორული კონტროლერების ბაზაზე.

№1 ლაბორატორიული სამუშაოს შესრულების პროცესში სტუდენტები აითვისებენ Simatic S7-300 კონტროლერის სამუშაოდ მომზადებისა და მისი დაპროგრამებისათვის აუცილებელ ყველა საჭირო პროცედურასა და ოპერაციას. №2 ლაბორატორიულ სამუშაოში სტუდენტები გაეცნობიან Simatic S7-300 კონტროლერის ინტეგრაციას SCADA სისტემა ProTool-თან. №3 ლაბორატორიულ სამუშაოში სტუდენტები შეისწავლიან STEP 7-ის ბაზური ინსტრუქციების პროგრამირებას და გაეცნობიან ტაიმერების ფუნქციონირებას ანალოგიური სიგნალის განივ–იმპულსური მოდულაციის რეალიზაციის მაგალითზე. №4 სამუშაოში სწარმოებს მართვის ობიექტის ექსპერიმენტალური კვლევა და მისდი მათემატიკური მოდელის აგება. №5 სამუშაოში განისაზღვრება ოპტიმალური პრამეტრები პი – რეგულატორისათვის. №6 სამუშაოში მოხდება ტემპერატურული რეჟიმის მართვის სისტემის აგება S7-300 კონტროლერის საფუძველზე და მისი ფუნქციონირების ხარისხის ექსპერიმენტალური შემოწმება.

თვითოეული ამ ლაბორატორიული სამუშაოსთვის სტუდენტებმა უნდა გააფორმონ ინდივიდუალურ ანგარიში. ანგარიშის ჩათვლის პროცესში სტუდენტმა უნდა უპასუხოს კითხვებს და შეარულოს პრაქტიკული დავალებები.

2.2. ლაბორატორიული სამუშაო №1

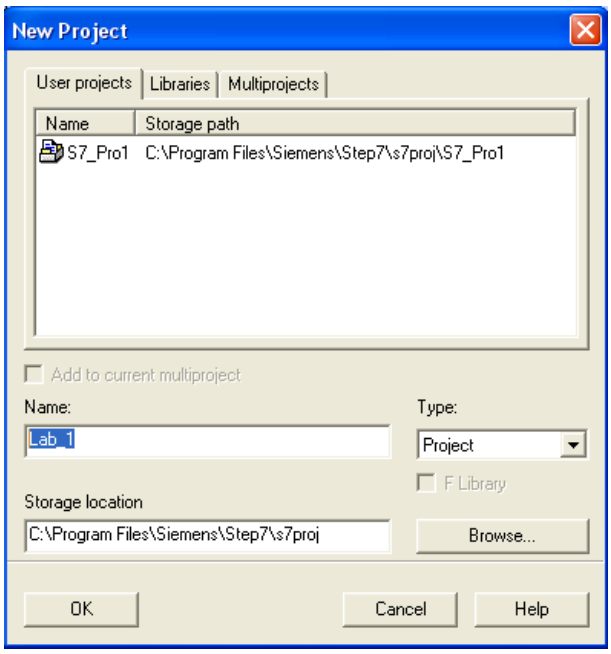
ანალოგიური სიგნალების შეყვანა Simatic S7-300 – ში

1.1 სამუშაოს მიზანი: Simatic S7-300კონტროლერში ანალოგიური სიგნალების შეყვანისა და დაკალიბრების უნარების შეძენა.

1.2. სამუშაოს შესრულების თანმიმდევრობა

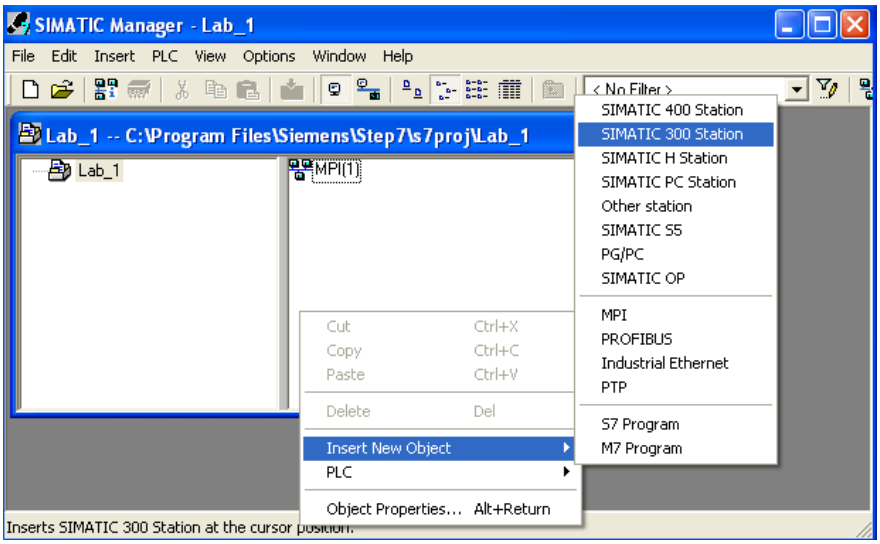
1.2.1. გაუშვით პროგრამა SIMATIC Manager;

1.2.2. შექმენით ახალი პროექტი, რისთვისაც შეასრულეთ File/New … გამოჩნდება ფანჯარა New project. შეიყვანეთ პროექტის დასახელება (იხ. ნახ. 1.3).



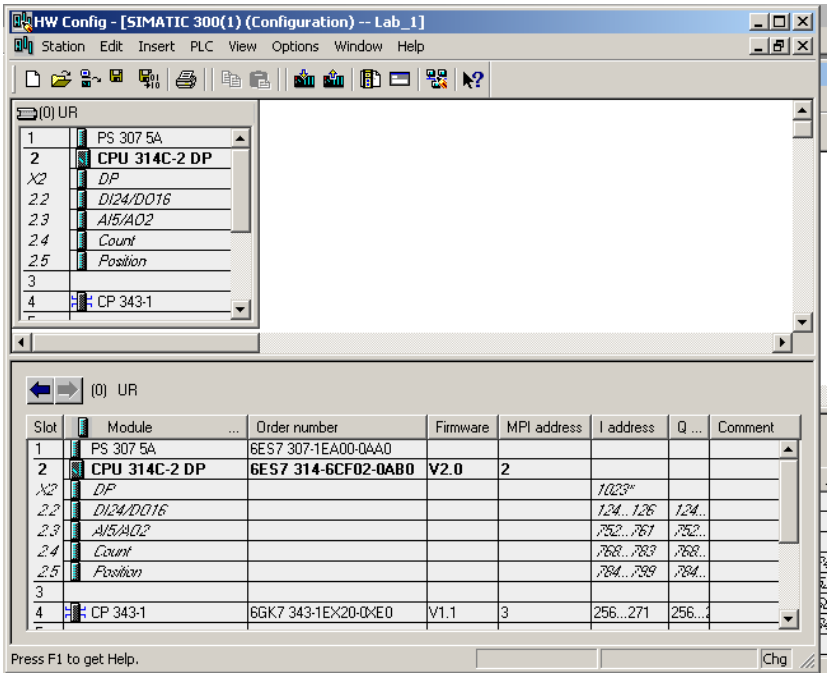
ნახ. 1.3 ფანჯარა ახალი პროექტის შესაქმნელად

1.2.3. შექმნილი პროექტის მარჯვენა ნაწილში დააწკაპეთ თაგვის მარჯვენა ღილაკზე და გამოჩენილ კონტექსტურ მენიუში აირჩიეთ Insert New Object/SIMATIC 300 Station (იხ. ნახ. 1.4).



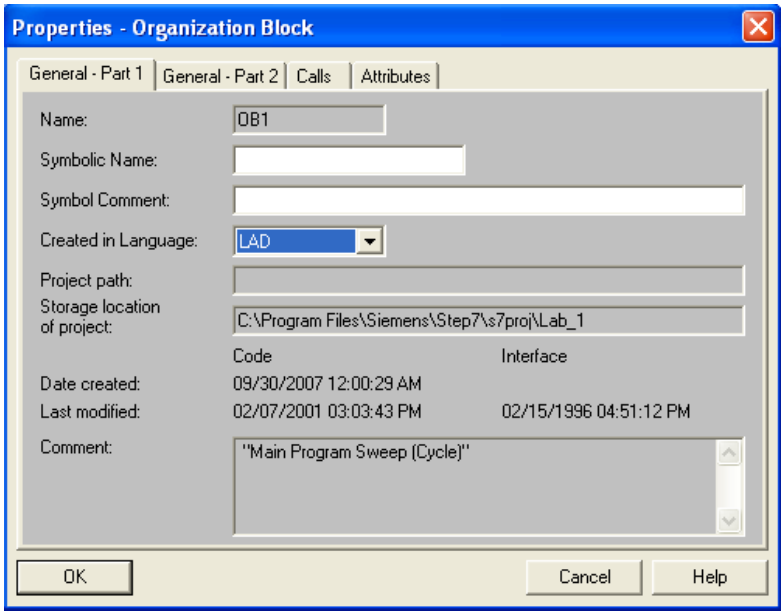
ნახ. 1.4 კონტროლერის შერჩევა

1.2.4. გახსენით შექმნილი SIMATIC 300(1), შემდგ გახსენით Hardware. გამოჩენილი ფანჯრის მარჯვენა ნაწილში გახსენით ხე შემდეგნაირად: SIMATIC 300/RACK-300. გადაიტანეთ ელემენტი Rail ეკრანის მარცხენა ნაწილში. ანალოგიურად, გადაიტანეთ მარჯვენა სტრიქონში ელემენტი PS 307 5A PS-300-სგან; მეორე სტრიქონში ელემენტი V2.0 CPU-300/CPU – სგან 314C-2DP; მეოთხეში – V1.1 CP-300/Industrial Ethernet/CP-314-1EX20-0XE0 (იხ. ნახ. 1.5). დახურეთ ფანჯარა HW Config. შემდეგში დაეთანხმეთ შენახვას, რისთვისაც დააჭირეთ ღილაკს Yes.



ნახ. 1.5 კონფიგურაცია SIMATIC 300(1)

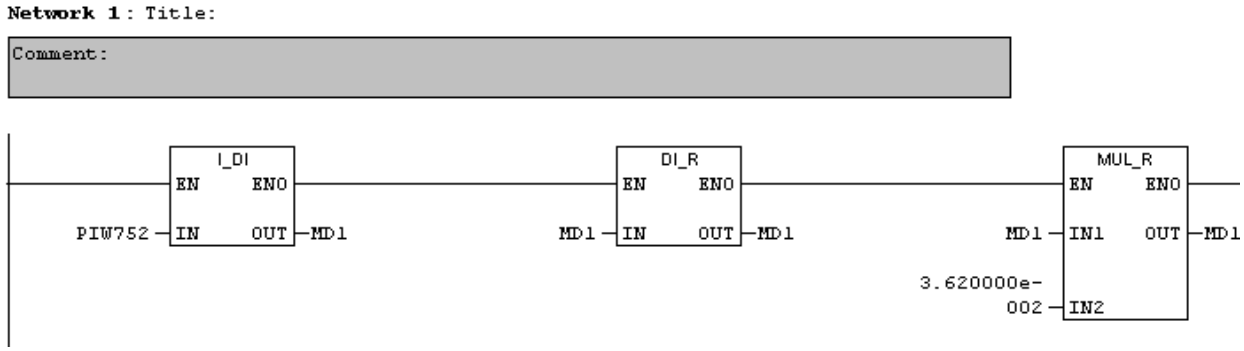
1.2.5. პროექტის ფანჯრის მარცხენა ნაწილში გახსენით ხე Blocks - ამდე. გახსენით საოტრგანიზაციო ბლოკი OB1. გამოჩენილ Properties ფანჯარაში აირჩიეთ ენა, რომელზედაც უნდა დაწეროთ პროგრამა: LAD (საკონტაქტო–სარელეო სქემები) (იხ. ნახ. 1.6).



ნახ. 1.6OB1 საორგანიზაციო ბლოკის პარამეტრების ფანჯარა

1.2.6. გამოჩენილ ფანჯარაში დაწერეთ დასმული ამოცანის შესრულების პროგრამა (ღუმელის ტემპერატურის მნიშვნელობის გადაღება) ორი ხერხით: 1) ქვემოთ შემოთავაზებული ალგორითმის გამოყენებით; 2) საბიბლიოთეკო ფუნქციის SCALE-ს გამოყენებით.

1.2.7. მთელი რიცხვის გარდაქმნის ალგორითმი Real–ის ტიპის რიცხვში, ანალოგიური შესასვლელისათვის:



ნახ. 1.7 პროგრამის შესრულება პირველი ხერხით

1.2.7.1. Converter-ისაგან გამოყავით შტო ფანჯრის მარცხენა ნაწილში, აიჩიეთ ელემენტი I_DI (ბლოკი, რომელიც გარდაქმნის Int მუდმივ რიცხვს ფიქსირებულ მძიმიან რიცხვად Dint), მასზედ ორჯერ დაწკაპებით. მიეცით ანალოგური შესასვლელი PIW754 და გამოსასვლელი MD1. (იხ. ნახ. 1. 7).

1.2.7.2. ანალოგიურად ამისა, გამოყავით შტო, Convert–ისგან (ეს ბლოკი ახდენს ფიქსირებულმძიმიანი რიცხვის Dint–ის გარდაქმნას ნამდვილ რიცხვში Real) დაამატეთ DI_R ბლოკი. მიეცით შესასვლელი MD1და MD1გამოსასვლელი (იხ. ნახ. 1.7).

1.2.7.3. გარდამქმნელისგან გამოდის სიგნალი 0 – 10 ვ, რაც შეესაბამება გასაზომ ტემპერატურას 0–1000°C. კონტროლერის აცპ–ში ეს სიგნალი 0 – 10 ვ გარდაიქმნება ციფრულში, რომლის გაზომვის დიაპაზონია 0–27648. იმისათვის, რომ მივიღოთ ტემპერატურის მნიშვნელობა, აუცილებელია, რომ სიგნალი აცპ–დან გარდავქმნათ შემდეგი დამოკიდებულებით:

$$\theta = k * აცპ + b,$$

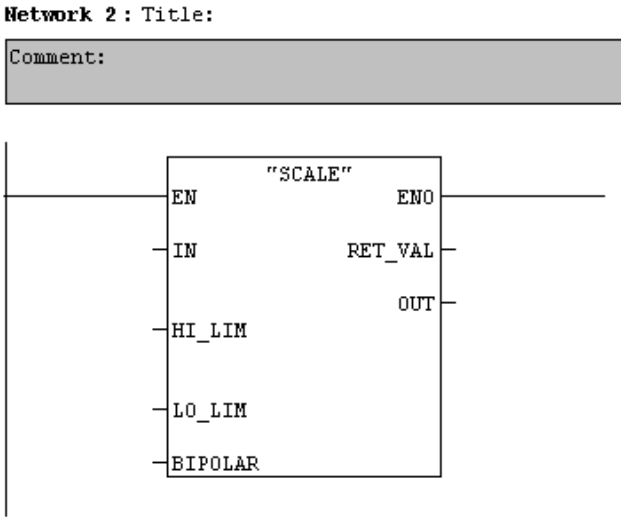
სადაც θ – ტემპერატურაა, °C; *აცპ* – აცპ–ს კოდია მთელრიცხვა წარმოდგენით.

ამ დამოკიდებულების რეალიზაციისათვის STEP 7–ის საშუალებების გამოყენებით ვამატებთ გამრავლების ბლოკს MUL_R Flouting-point fct - ისაგან. გამოსასვლელად მიეცით MD1, პირველ შესასვლელად – MD1, ხოლო მეორე შესასვლელად – კოეფიციენტ k-ს მნიშვნელობა (იხ. ნახ. 2.7).

კოეფიციენტების k-ს და b განსაზღვრისათვის აუცილებელია განტოლებათა სისტემის ამოხსნა:

$$\begin{cases} \theta_{min} = k \cdot აცპ + b \\ \theta_{max} = k \cdot აცპ + b \end{cases}$$

1.2.8. მეორე ხერხი. გამოყავით შტო Libraries/Standart Library/TI-S7 Converting Blocks ფანჯრის მარცხენა მხარეს და აირჩიეთ FC 105 SCALE Convert (იხ. ნახ. 1.8).



ნახ. 1.8 პროგრამის შესრულების მეორე წესი

შესასვლელებისა და გამოსასვლელების დანიშვნა:

IN – დასამასშტაბირებელი სიდიდის შეასვლელი სიგნალი;

HI_LIM – დასაკალიბრებელი სიდიდის ზედა ზღვარი;

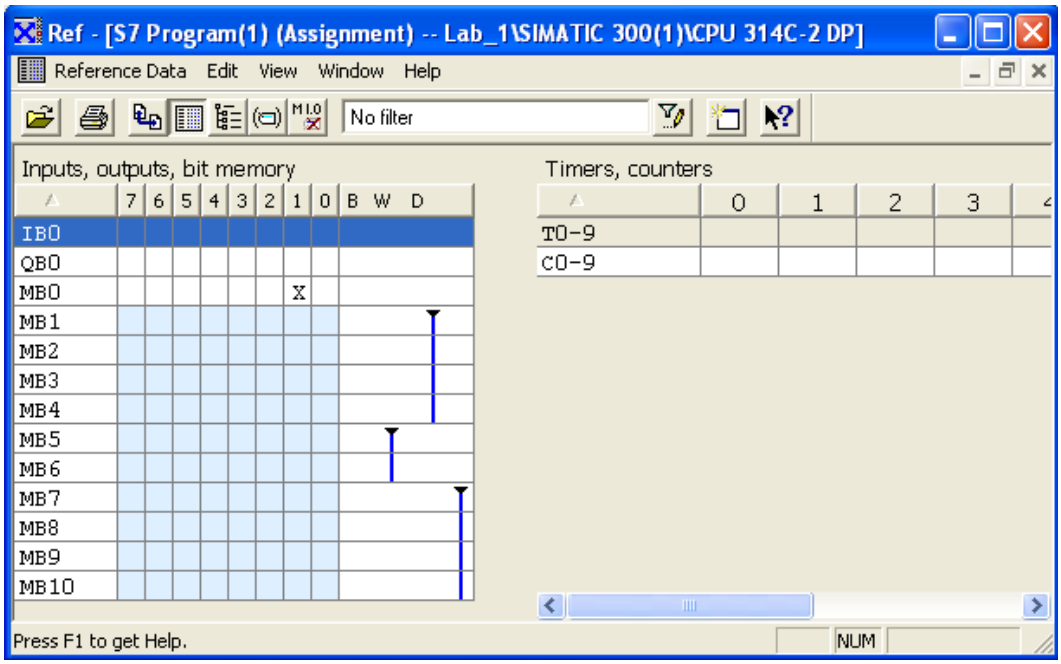
L0_LIM – დასაკალიბრებელი სიდიდის ქვედა ზღვარი;

BIPOLAR – შესასვლელი, რომელიც ადგენს სიგბალის პოლაროზას (0 – უნიპოლიარული; 1 – ბიპოლიარული);

RET_VAL – აბრუნებს 0 – ს, თუ შესრულების დროს არ იყო შეცდომები;

OUT –მასშტაბირებული გამოსასვლელი.

1.2.9. იმისათვის, რომ შეხედოთ ამ მომენტისათვის დაკავებულ მერკერულ მეხსიერებას, მთავარ მენიუში აირჩიეთ Options/Referance Data/Display და Assignment (Input, Output, Bit Memory, Timers, Counters) (იხ. ნახ. 1.9).



ნახ. 1.9 მეხსიერების ფანჯარა

1.2.10 შეინახეთ ცვლილებები OB1-ში, მთავარ მენიუში File/Save - ს არჩევით.

1.2.11 პროექტის ფანჯრის მარცხენა ნაწილში ჩაკეცევთ ყველა ”–” SIMATIC 300 (1) – მდე. ინსტრუმენტების პანელზე დააჭირეთ ღილაკს Download  და შემდეგ დაეთანხმოთ ყველაფერზე.

1.2.12 გაუშვით კონტროლერი (დააყენეთ ტუმბლერი Run – ში).

1.2.13 დაბრუნდით OB1 ბლოკში და ინსტრუმენტების პანელზე დააჭირეთ ღილაკს Monitor (on/off)  . თუ ყველაფერი შესრულებულია სწორად, მაშინ OB1 ფანჯრის

მარჯვენა ნაწილში გამონათდება ღუმელის ტემპერატურის გაზომვის რეზულტატი. განსაზღვრის ორთავე წესისათვის რეზულტატები უნდა ეთანხმებოდნენ ერთმანეთს.

1.2.14 თუ წარმოიშვება პროგრამის გადარედაქტირებისა საჭიროება, უნდა აუშვათ ღილაკს Monitor (on/off)  . და გამორთოთ კონტროლერი (ტუმბლერი Stop - ში. გადაარედაქტორეთ პროგრამა 1.3.10 – 1.3.13 პუნქტების გამოყენებით.

1.2.15 გააკეთეთ დასკვნები შესრულებულ სამუშაოზე.

2.1.4 საკონტროლო კითხვები და დავალებები

1.3.1 რა როლს თამაშობენ კონტროლერები ავტომატიზაციის სისტემებში?

1.3.2 განმარტეთ პროექტის შექმნის თანმიმდევრობები STEP 7 – ში.

1.3.3 რა ენებზეა შესაძლებელი პროგრამების შექმნა STEP 7 – ში და რაში მდგომარეობს მათი თავისებურებები?

1.3.4 რა ფუნქციებს ასრულებს ბლოკი ”scale”?

1.3.5 რა არეებისაგან შედგება SIMATIC S7-300 კონტროლერის მეხსიერება?

1.3.6 რა ტიპის ბლოკები არსებობს STEP 7 – ში და რა არის მათი ფუნქციონალური დანიშნულება?

ანგარიშის აუცილებელი შემადგენლები

1. ლაბორატორიული სტენდის კონფიგურირებული სადგურის S-300-ის სკრინშოტი.

2. პროგრამა STEP 7 – ზე.

2.3. ლაბორატორიული სამუშაო № 2

ანალოგური სიგნალის ვიზუალიზაცია და დაარქივება ProTool სისტემის SCADA სისტემის გამოყენებით

2.1 სამუშაოს მიზანი:

ProTool სისტემის SCADA–სთან მუშაობის გამოცდილების მიღება.

2.2 თეორიული შესავალი

რა არის ProTool?

ProTool – ესSCADA (Supervisory Control And Data Acquisition - დისპეჩერული მართვა და მონაცემთა შეგროვება) სისტემაა. SCADA – ეს სპეციალიზირებული პროგრამული უზრუნველყოფაა, რომელიც ორიენტირებულია დისპეტჩერსა და მართვის სისტემას შორის ინტერფეისის უზრუნველყოფაზე, აგრეთვე კომუნიკაციაზე გარე სამყაროსთან.

ProTool/ Pro შედგება კონფიგურირების სისტემისაგან ProTool/Pro CS (Configuration System) და ტექნოლოგიური პროცესის ვიზუალიზაციის პაკეტისაგან ProTool/Pro RT (Runtime).

ProTool/ Pro გვთავაზობს შემდეგ შესაძლებლობებს:

- პროცესის ვიზუალიზაციის მოსახერხებულ სისტემას სტანდარტული ელემენტების დიდი არჩევანით, როგორიცაა: შყვანა/გამოყვანის ველები, ჰისტოგრამები, ტრენდების გრაფიკები, რასტრული და ვექტორული გრაფიკა და დინამიური ატრიბუტები;
- შეტყობინებების ინტეგრირებულ სისტემას;
- პროცესისა და გზავნილის ცვლადების არხივიზაციას;
- სამომხმარებლო ფუნქციებს Visual Basic Script® –ის საფუძველზე;
- დრაივერებს SIMATIC S5, SIMATIC S7 და სხვა მწარმოებლების PLC – ებთან კავშირებისათვის.

STEP 7 – ში ინტეგრირებულ ProTool–ში კონფიგურირება

ProTool შეიძლება იყოს ინტეგრირებული SIMATIC STEP 7 – ის პროგრამულ უზრუნველყოფასთან, რაც საშუალებას იძლევა გამოყენებულ იქნას სიმვოლოები და მონაცემთა ბლოკები როგორც ProTool – ის ტეგები. ეს არა მხოლოდ დროის ეკონომიას იძლევა, არამედ ამცირებს შეცდომების შესაძლებლობას ერთი და იგივე მონაცემების განმეორებით შეყვანის შემთხვევაში.

ProTool **– ის ინტეგრაცია STEP 7** – ში იძლევა შემდეგი სახის ***უპირატესობებს:***

- შეიძლება ProTool პროექტების მართვა SIMATIC Manager – ის გამოყენებით (ანუ იგივე პროგრამის მეშვეობით, რაც გამოიყენება STEP 7 პროექტების მართვისათვის);

- შეიძლება STEP 7–ის სიმბოლოებისა და მონაცემთა ბლოკების გამოყენება S7 სიმბოლოთა ცხრილიდან როგორც ტეგებისა. მონაცემთა ტიპი და მისამართი ამ შემთხვევაში დაიტანება ავტომატურად;
- ProTool ასახავს ამ შემთხვევაში PLC – ის მთელ სიას STEP 7–ის პროექტში და შემდეგ, PLC – ის არჩევის შემდეგ განსაზღვრავს მისამართის შესაბამის პარამეტრებს;
- STEP 7–ში შეიძლება ALARM_S – ის შეტყობინების კონფიგურირება და მათი გაყვანა ოპერატორის ტერმინალზე;
- სიმბოლური სახელი დანიშნულ უნდა იქნას მხოლოდ ერთჯერ, ხოლო შემდეგ შეიძლება მისი გამოყენება საჭიროებისდა მიხედვით.

ProTool პროექტის ფანჯარა.

ახალი ან არსებული პროექტის გახსნისას, გამოჩნდება *პროექტის ფანჯარა.*

პროექტის ფანჯარაში ობიექტის ის ტიპები, რომელთა კონფიგარაციაცაა საჭირო იმყოფებიან მარცხნივ, ოლო თვითონ ობიექტები მარჯვნივ. თუ როგორი ობიექტებია შესაძლებელი კონფიგურირებისათვის დამოკიდებულია კონკრეტული ოპერატორული ტერმინალისაგან.

სხვადასხვა ობიექტები ProTool–ში უშუალოდ დაკავშირებულია პროგრამებთან მათი რედაქტირებისათვის.

პროექტის მონაცემები ProTool – ში შენახულნი არიან *ობიექტების* სახით. ობიექტები პროექტში ორგანიზებულნი არიან *ხისმაგვარი სტრუქტურის* სახით.

პროექტის ფანჯარა ასახავს ობიექტების ტიპს, რომლებიც განეკუთვნება პროექტს, ისინი შეიძლება დარედაქტირდეს მოცემულ ოპერატორულ ტერმინალზე. ობიექტთა კლასები შეიცავენ ისეთ ობიექტებს, რომელთა თვისებებიც შეიძლება შეცვლილ იყოს.

პროექტის ფანჯარა ორგანიზებულია შემდეგი სახით (იხ. ნახ. 2.1):

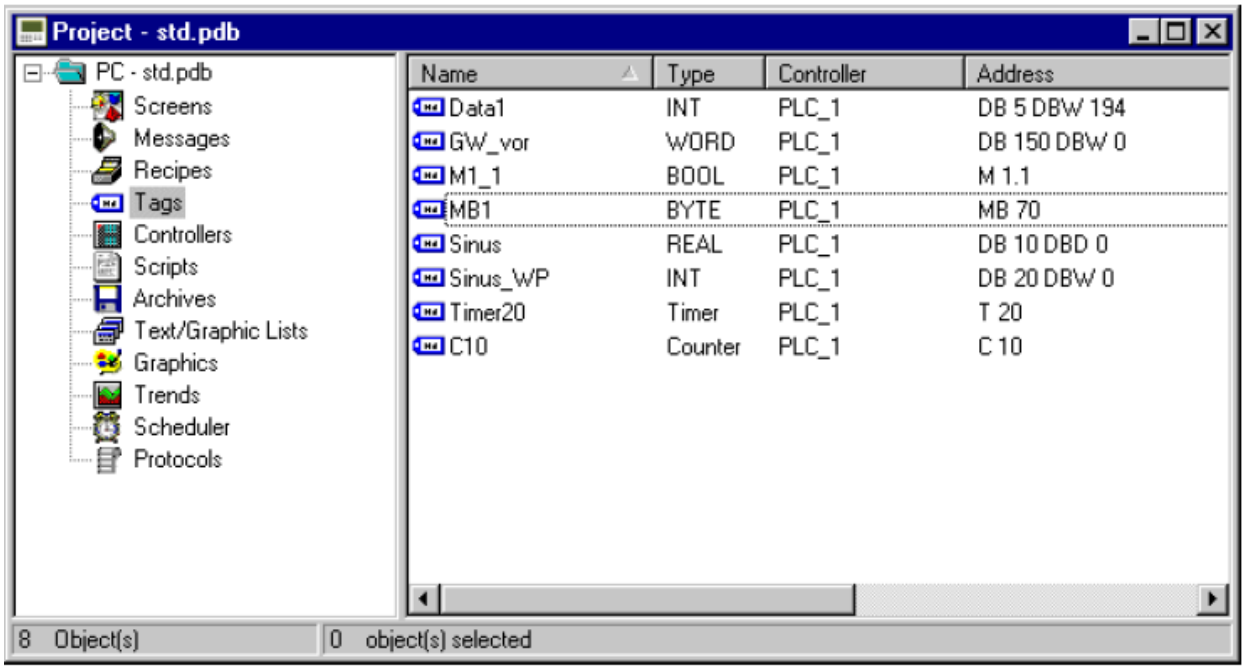
- პროექტის სათაური შეიცავს *პროექტის სახელს*;
- ეკრანის მარცხენა ნახევარი ასახავს *ობიექტთა ტიპებს*, რომლებიც შეიძლება კონფიგურირებულ იქნას, ხოლო მასში შემავალი ობიექტები აისახება ფანჯრის მარჯვენა ნახევარში.

ტეგების გამოყენება

ტეგ – ეს SCADA სისტემის ცვლადია, რომელსაც გააჩნია სიმბოლური სახელი და მონაცემთა გარკვეული ტიპი. ტეგ–ის მნიშვნელობა იცვლება PLC - ის პროგრამის შესრულების დროს.

ტეგებს, რომლებიც დაკავშირებულია PLC - სთან ეწოდებათ ***გლობალური.*** ხოლო ის ტეგები, რომლებიც PLC - სთან არ არიან დაკავშირებული ეწოდებათ ***ლოკალური.***

გლობალური ტეგი იკავებს PLC - ში გარკვეულ სამისამართო სივრცეს, რომელიც ხელმისაწვდომია ჩაწერისა და წაკითხვისათვის როგორც ოპერატორული ტერმინალისგან ასევე PLC–ისგან.



ნახ. 2.1 პროექტის ფანჯრის მაგალითი ტეგებით.

ლოკალური ტეგები ხელმისაწვდომია მხოლოდ ოპერატორული ტერმინალის საძღვრებში. ლოკალური ტეგები შეიძლება შეიქმნას, მაგალითად, იმისათვის, რომ ოპერატორს შესაძლებლობა ჰქონდეს მოახდინოს საჭირო დაყენებები ოპერატორულ ტერმინალზე.

ProTool–ს შეუძლია შემდეგი ტიპის ***ტეგების გამოცნობა*** (მაგრამ, მათგან ყველანი არ არის ხელმისაწვდომი ცალკეული PLC–ისათვის).

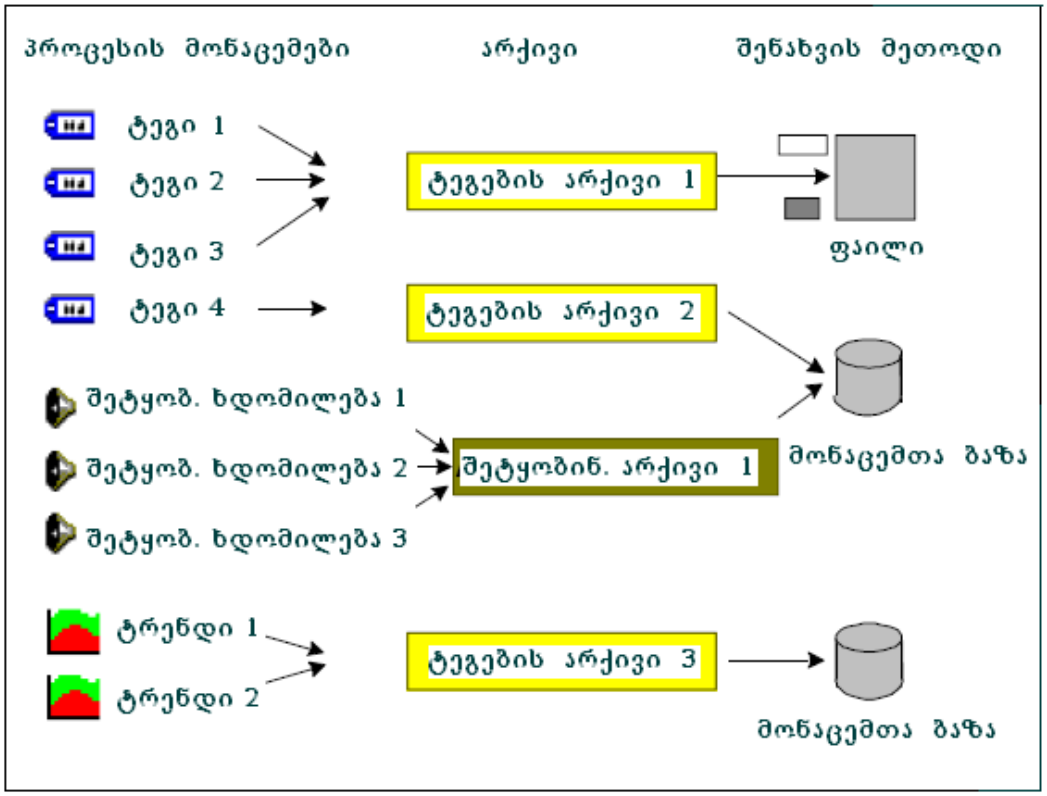
მონაცემთა ტიპები	თანრი–გები	მნიშვნელობათა დიაპაზონი
INT	16 ბიტი	–32768–დან 32767–მდე
UINT	16 ბიტი	0–დან 65535–მდე
LONG	32 ბიტი	–2147483648–დან 2147483647
ULONG	32 ბიტი	0–დან 4294967295
FLOAT	32 ბიტი	ზედა ზღვარი; ± 3.402823 ე+38 ქვედა ზღვარი: ± 1.175495 ე–38
BOOL	–	true (1), false (0)
STRING	–	1–დან 255 ბაიტამდე
DATETIME	64 ბიტი	მნიშვნელობა თარიღი/დრო
ტეგების მასივი	ასეთი ტიპი აერთიანებს ერთიდაიგივე ტიპის ნებისმიერი რაოდენობის ტეგს ერთ მთლიანობაში, რომელთანაც მიკითხვა შეიძლება დამოუკიდებელ ობიექტთან	

არქივის შექმნა

სისტემები Windows–ის ბაზაზე შესაძლებლობას იძლევა დავაარქივოთ პროცესის მონაცემები (ანუ, შევინახოთ ისინი დიდი ხნის განმავლობაში და შემდეგ ვაანალიზოთ იგი).

შესაძლებელია დაარქივდეს პროცესის შემდეგი ტიპები მონაცემებისა:

- ***ტეგები.*** *Tags* (*ტეგები*) დიალოგურ ფანჯარაში განსაზღვრავენ აქტივიზაციის პირობებსა და მნიშვნელობათა დიაპაზონს ტეგების აქტივიზაციისათვის;
- ***შეტყობინებები.*** ამოვირჩეთ რა მენიუში *System→Mesages→Settings* (*სისტემა→შეტყობინება→პარამეტრები*) შეიძლება გაირკვეს, თუ რომელი შეტყობინებები იქნებიან დაარქივებული;
- ***ტრენდები.*** არქივის დაყენებისათვის, საიდანაც წაიკითხებიან ტეგები ტრენდის ასახვისათვის, გამოიყენებენ ჩანართს Data Source *(მონაცემთა წყაროები)* დიალოგიურ ფანჯარაში *Trend* (*ტრენდი*);
- არქივის თვისებები, ისეთები, როგორც შენახვის ადგილი და სხვ. განისაზღვრება დიალოგურ ფანჯარაში Trend (*ტრენდი*);
- დიაგრამა, რომელიც წარმოდგენილია ნახ. 2.2–ზე გვაჩვენებს ***არქივირების მოდელს.***



ნახ. 2.2 არქივირების მოდელი

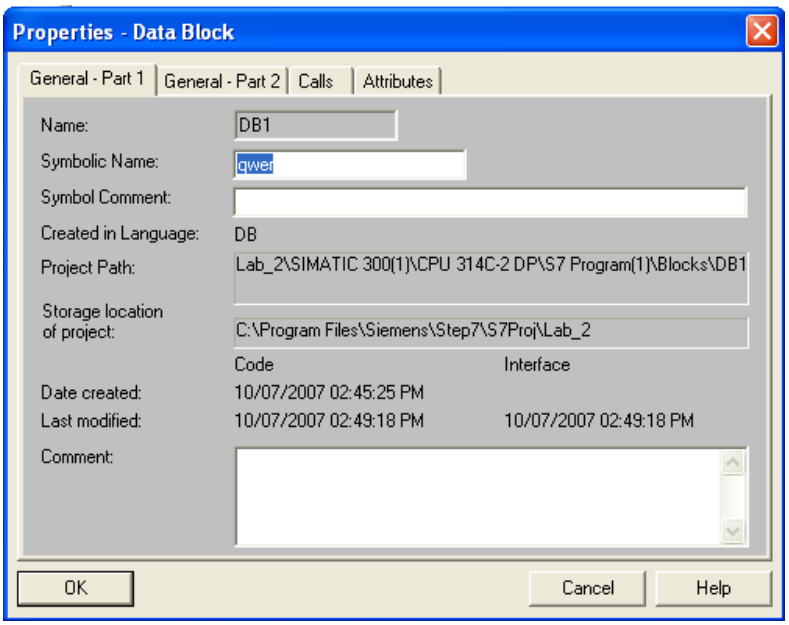
არქივის თვისებები. არქივი შეიძლება იყოს ორი ტიპის:

1) **Shot-term archive (მოკლეხნოვანი არქივი) –** FIFO ბუფერი, ეს ნიშნავს იმას, რომ თუ მაგალითად ბუფერს აქვს ტევადობა 100, მაშინ შეინახება მხოლოდ ბოლო 100 მნიშვნელობა. ძველ ჩანაწერებზე კი ხდება ახლების გადაწერა;
2) **Sequence archive (მიმდევრობითი არქივი) –** ეს არქივი ივსება განსაზღვრულ სიდიდემდე. შემდგომ კი მუშაობის გაგრძელებისთვის აუცილებელია არქივის აწყობის შეცვლა. მიმდევრობითი არქივებისათვის შეიძლება ავიჩიოთ ერთერთი შემდეგი ოფციათაგანი:
   - *Automatically Create Sequence Archive* (*მიმდევრობითი არქივის ავტომატური შექმნა*). არქივები იღებენ გარკვეულ სახელებს 1 დან n - მდე ნომრების დამატებით. არქივების რაოდენობა შეიძლება კონფიგურირებულ იქნას. როგორც კი საბოლოო არქივის შევსება დამთავრდება, დაიწყება პირველი არქივის თავიდან შევსება.
   - *Output System Message When* (*როდესაც გამოდის სისტემური შეტყობინება*). თუ არქივი (მაგ. დისკეტა) შეივსება, გამოვა სისტემური შეტყობინება. შეიძლება იმ დონის განსაზღვრა პროცენტებში, რა დროსაც უნდა გამოვიდეს შეტყობინება.
   - *Trigger Function (ფუნქციის აქტივიზაცია*). როდესაც არქივი შეივსება, გააქტიურდებიან ფუნქციები, რომლებიც ჩართავენ მიმდევრობითი არქივის სპეციალურ გადამუშავებას.

2.3. სამუშაოს შესრულების თანმიმდევრობა

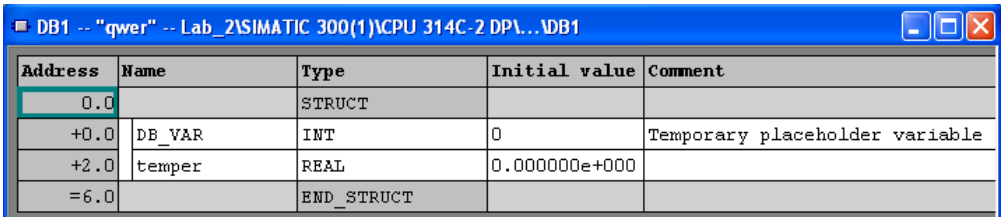
2.3.1 გაუშვით პროგრამა ***SIMATIC Manajer.***

2.3.2 გახსენით პროექტი, რომელიც შექმნილი იყო ლაბორატორულ სამუშაო №1–ში. ფანჯრის მარცხენა ნაწილში გახსენით ხე Bloks-ებამდე. შექმენით მონაცემთა ბლოკი DB, რისთვისაც დააწკაპეთ ფანჯრის მარჯვენა ნაწილში თაგვის მარჯვენა ღილაკით. გამოჩენილ კონტექსტურ მენიუში აირჩიეთ Insert New Object/Data Bloks. ახლად გამოჩენილ ფანჯარაში Properties (იხ. ნახ. 2.3) ჩაწერეთ ბლოკის სახელი (qwer). ამ შემთხვევაში DB1 ბლოკს გამოვიყენებთ როგორც მეხსიერების არედ, სადაც შესაძლებლობა იქნება ცვლადების შენახვისა.



ნახ. 2.3 მონაცემთა ბლოკის DB1-ის პარამეტრების ფანჯარა

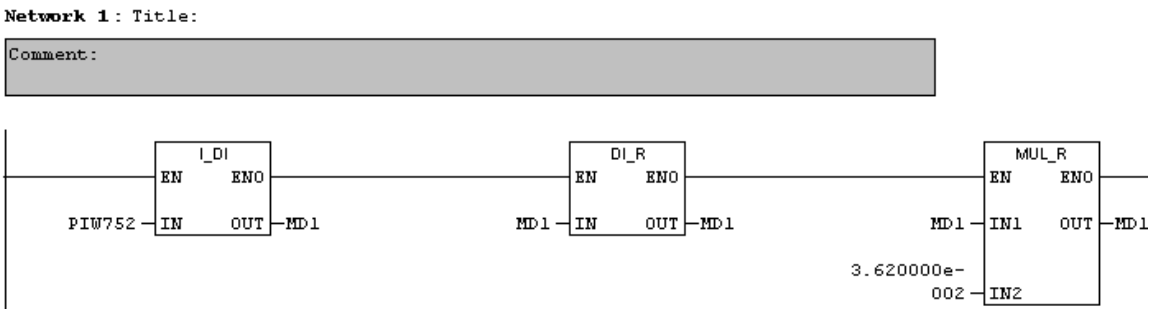
2.3.3. ორმაგი დაწკაპებით გახსენით DB1, სადაც შექმენით ახალი ცვლადი Temper (იხ. ნახ. 2.4). შეინახეთ ცვლილებები File/Save არჩევით.



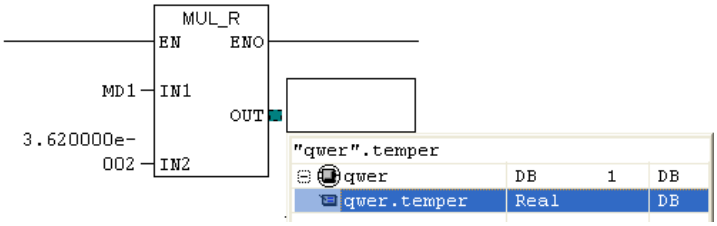
Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	DB_VAR	INT	0	Temporary placeholder variable
+2.0	temper	REAL	0.000000e+000	
=6.0		END_STRUCT		

ნახ. 2.4 DB1 მონაცემთა ბლოკის ფანჯარა

2.3.4 გახსენით OB1; მზა პროგრამაში (იხ. ნახ. 2.5) MUL_R ბლოკში შეცვალეთ გამოსასვლელი მნიშვნელობა (OUT) MD1 "qwer". temper -ზე შემდეგნაირად (იხ. ნახ. 2.6): დააყენეთ კურსორი MD1–ზე, დააჭირეთ "პრობელ", გახსენით "+" qwer–ისთვის და აარჩიეთ *"qwer". temper*. შეინახეთ ცვლილებები File/Save არჩევით.

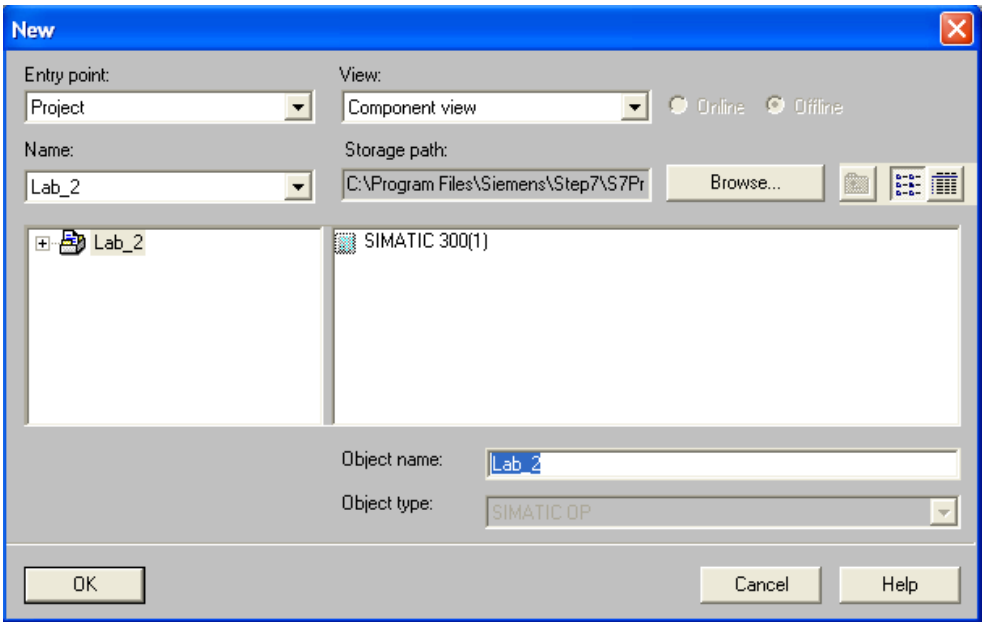


ნახ. 2.5 პროგრამის საწყისი სახე



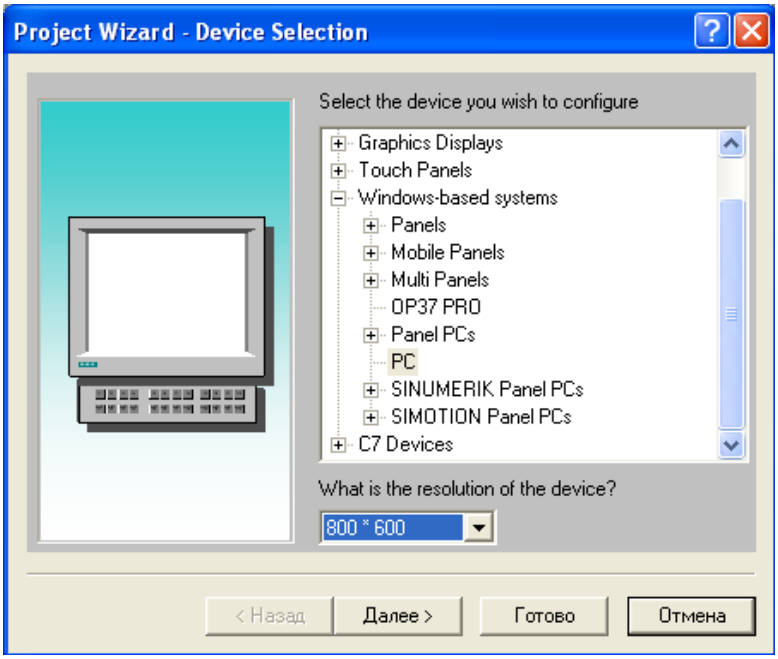
ნახ. 2.6 ბლოკის გამოსასვლელის ცვლადის დანიშვნა

2.3.5 გაუშვით პროგრამა ***ProTool*** SIMATIC/ProTool Pro CS – ისგან. შექმენით ახალი პროექტი File/New…, გამონათებულ ფანჯარაში აირჩიეთ პროექტი, რომელიც შექმნილია STEP 7-ში, შეიყვანეთ Object Name; აირჩიეთ OK (იხ. ნახ. 2.7).



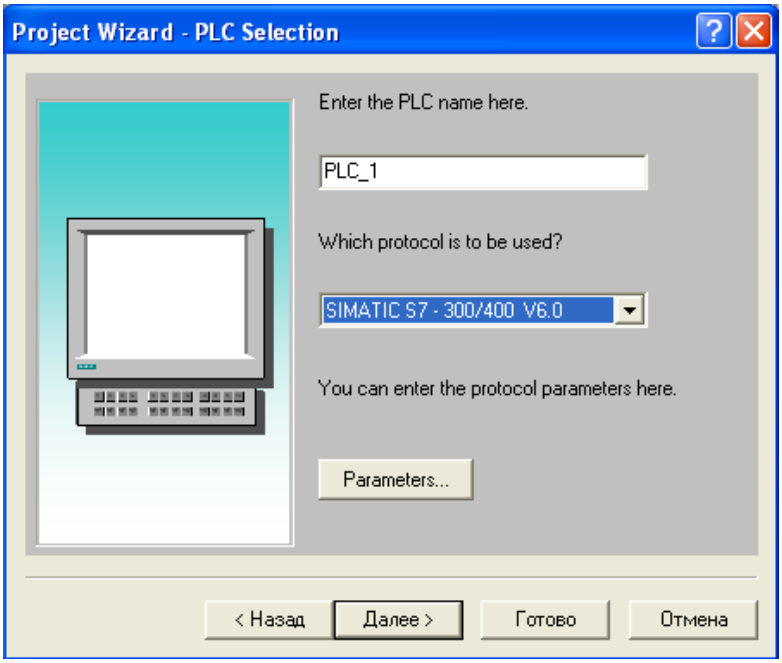
ნახ. 2.7 ახალი პროექტის შექმნის ფანჯარა *ProTool* –ში

3.3.6 ფანჯარაში Project Wizard გახსენით "+" Windows-bazed systems, ხოლო შემდეგ კი აირჩიეთ PC და მიუთითეთ გარჩევადობის აუცილებელი მნიშვნელობა. (იხ. ნახ. 2.8); შემდეგ დააჭირეთ ">" – ს .



ნახ. 2.8 *Project Wizard - Device Selection* - ის ფანჯარა

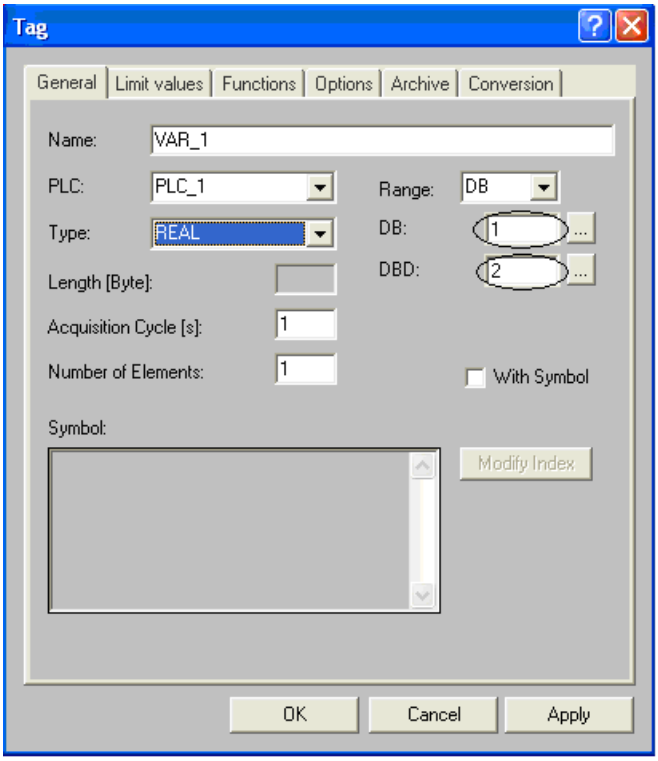
2.3.7 გამოჩენილ ფანჯარაში აირჩიეთ SIMATIC S7-300/400 V6.0 (იხ. ნახ. 2.9). შემდეგ დააჭირეთ ”>” – ს და მზადაა.



ნახ. 2.9 *Project Wizard - PLC Selection* - ის ფანჯარა

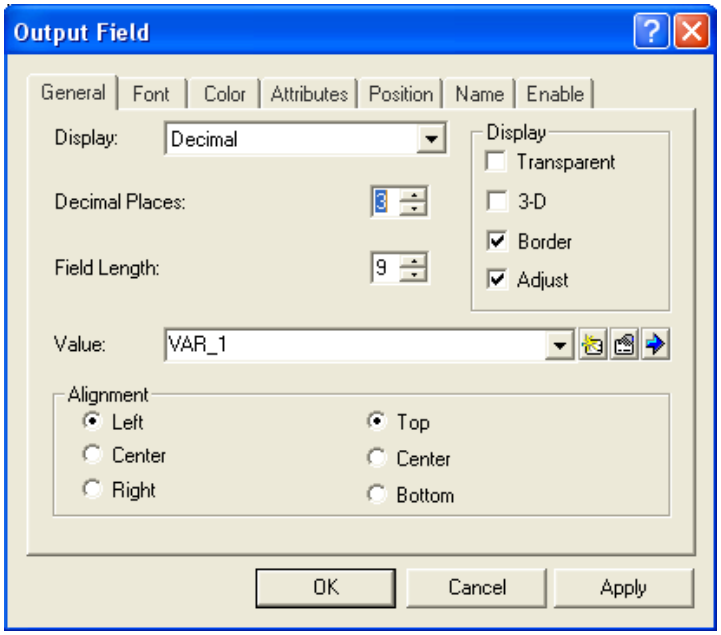
2.3.8 შექმენით ტეგი, ამისათვის პროექტის Lab_2 - ის მარცხენა ნაწილში დააწკაპეთ თაგვის მარჯვენა ღილაკით Tag – ზე, აირჩიეთ Tag insert … და გამოჩენილ ფანჯარაში შექმენით ტეგ–ის პარამეტრები (იხ. ნახ. 2.10). აქ DB ველში მოთავსებულია *DB*

მონაცემთა ბლოკის სახელი პროგრამა Step 7 – ში, ხოლო DBD – ველში კი *DB* მონაცემთა ბლოკის მისამართი იმავე პროგრამა Step 7 – ში. მისამართი შეიძლება ინახოს 2.2.3 პუნქტში.

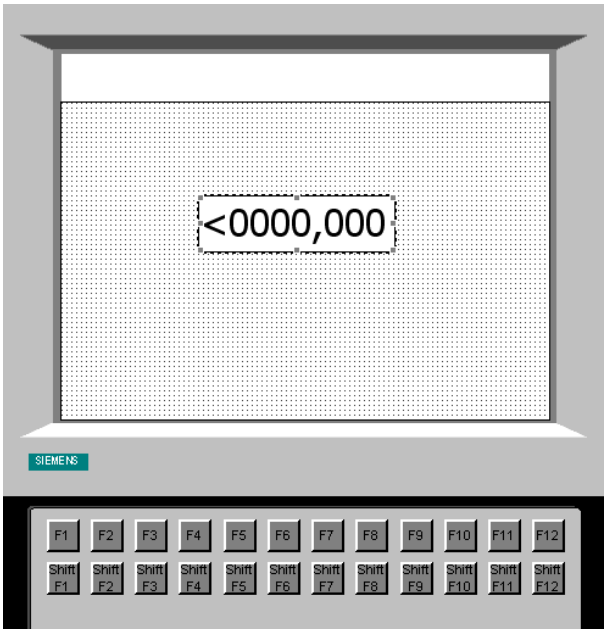


ნახ. 2.10 *Tag* პარამეტრების ფანჯარა

2.3.9 Lab_2 პროექტის მარცხენა ნაწილში ორმაგი დაწკაპებით გახსენით Screens. ინსტრუმენტების პანელზე აირჩიეთ Output Field  (*გამოსასვლელის ველი* – მისი დანიშნულებაა ოპერატორის ტერმინალზე სხვადასხვა ფორმატის მნიშვნელობების გამოყვანა). ობიექტის პარამეტრების გამოჩენილ ფანჯარაში Output Field მიეცით შემდეგი: Decimal Places: 3 (ნიშნების რიცხვი მძიმის შემდეგ); Field Length: 9 (ველის ნიშნების რიცხვი); Value: VAR_1; OK (იხ. ნახ. 2.11). ამ პროცედურების შედეგად მუშა არეში გამოჩნდება ღუმელის ტემპერატურის გაზომვის შედეგის ასახვის მასკა (იხ. ნახ. 2.12). შეინახეთ ცვლილებები File/Save არჩევით.

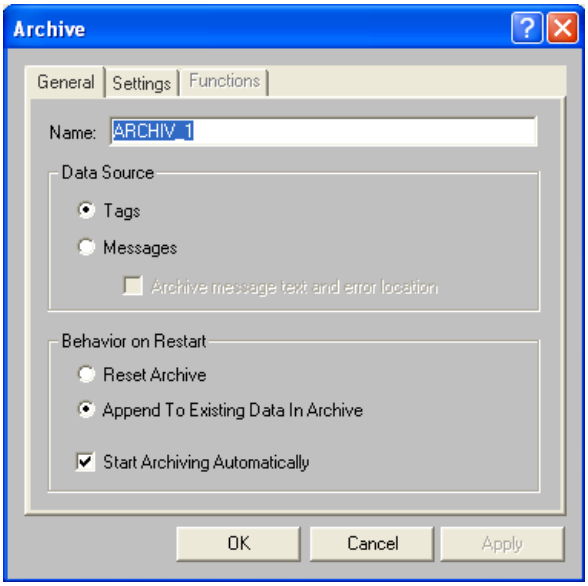


ნახ. 2.11 *Output Field* – ის პარამეტრების ფანჯარა



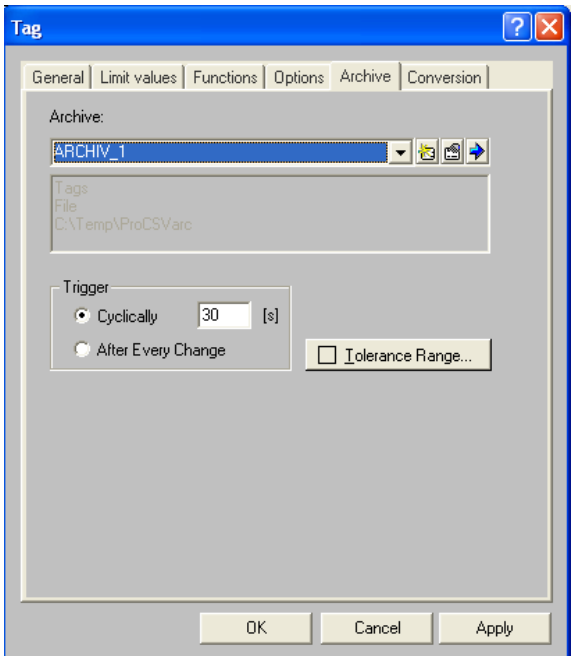
ნახ. 2.12 სამუშაო არე

2.3.10 შექმენით არქივი, სადაც მოხდება ღუმელის ტემპერატურის გაზომვის რეზულტატის შენახვა. ამისათვის, Lab_2 პროექტის მარცხენა ნაწილში დააწკაპეთ თაგვის მარჯვენა ღილაკით Archives - ზე, აირჩიეთ Archive insert … და გამოჩენილ ფანჯარაში მიეცით არქივის პარამეტრები (იხ. ნახ. 2.13).



ნახ. 2.13 *Archive* პარამეტრების ფანჯარა

2.3.11 გახსენით VAR_1 ტეგის ფანჯარა Properties, სადაც აირჩიეთ ჩანართი Archive და მიუთითეთ არქივის სახელი, რომელშიც უნდა ჩაიწეროს ტეგის მნიშვნელობა (ჩვენ შემთხვევაში (ARCHIVE_1) და დააყენეთ Ciclycally ტოლი 30s - ის, რაც ნიშნავს ტეგის მონაცემთა ჩაწერის დისკრეტობას არქივში.



ნახ. 2.14 *Tag* პარამეტრების ფანჯარა

3.3.12შეამოწმეთ პროექტის მუშაობის რეზულტატი სიმულიაციის რეჟიმში. ამისათვის პანელზე ავირჩიოთ *StartPro/Tool/ProSimulation* . გამონათებულ ფანჯარაში შეიყვანეთ მონაცემები შემდეგი თანმიმდევრობით:

VAR_1	REAL	0	Dec	1.0	Random		7,5	10		✓

2.3.13 შეამოწმეთ პროექტის მუშაობის რეზულტატი რეალურ კონტროლერზე. ***STEP7*** - ში პროექტის ფანჯრის მარცხენა ნაწილში ჩაკეცეთ ყველა ”–” SIMATIC 300 (1) - მდე. ინსტრუმენტების პანელზე დააჭირეთ ღილაკს Download და შემდეგ დაეთანხმეთ ყველაფერში. ჩატვირთეთ კონტროლერი (დააყენეთ ტუმბლერი Run – ში). ***ProTool*** - ში ინსტრუმენტების პანელზე დააჭირეთ ღილაკს *Start ProTool/Pro RT* . თუ კი ყველაფერი ეს სწორედ იქნა შესრულებული, მაშინ გამოჩნდება ფანჯარა, სადაც გამონათებული იქნება ღუმელის ტემპერატურის გაზომვის რეზულტატი.

2.3.14 ნახეთ ღუმელის ტემპერატურის გაზომვის რეზულტატი Excel - ში. ამისათვის პაკეტში ProCSVate ირჩიეთ დოკუმენტი ადრე შექმნილი არქივის სახელწოდებით (ამ შემთხვევაში ARCHIV_10, რადგან ProTool ამატებს ფაილის სახელს სიმბოლოს –0–). რეზულტატი წარმოდგენილ იქნება პროგრამა Excel - ში. (იხ. ნახ. 2.15):

Microsoft Excel - ARCHIV_10

Файл Правка Вид Вставка Формат Сервис Данные Окно Справка

Arial Cyr 10

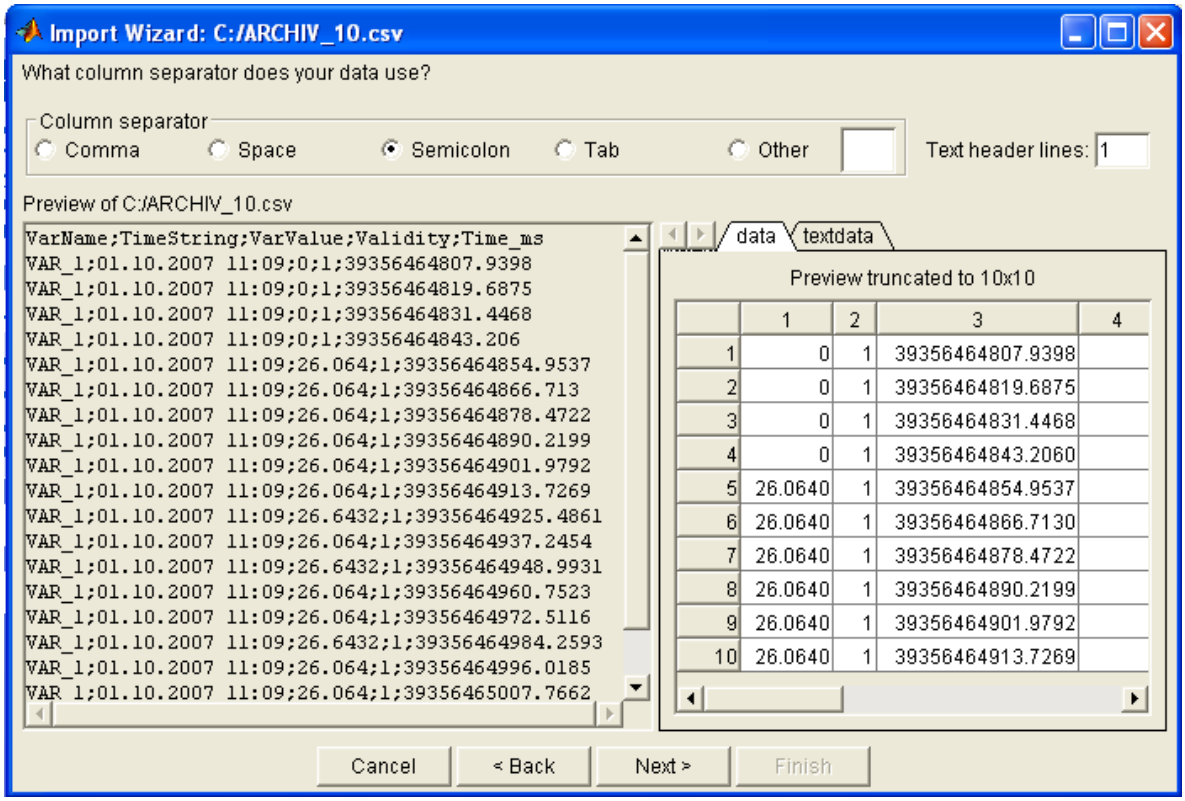
A22

	A	B	C	D	E	F
1	VarName	TimeString	VarValue	Validity	Time_ms	
2	VAR_1	01.10.2007 11:09	0	1	39356464807.9398	
3	VAR_1	01.10.2007 11:09	0	1	39356464819.6875	
4	VAR_1	01.10.2007 11:09	0	1	39356464831.4468	
5	VAR_1	01.10.2007 11:09	0	1	39356464843.206	
6	VAR_1	01.10.2007 11:09	26.064	1	39356464854.9537	
7	VAR_1	01.10.2007 11:09	26.064	1	39356464866.713	
8	VAR_1	01.10.2007 11:09	26.064	1	39356464878.4722	
9	VAR_1	01.10.2007 11:09	26.064	1	39356464890.2199	
10	VAR_1	01.10.2007 11:09	26.064	1	39356464901.9792	
11	VAR_1	01.10.2007 11:09	26.064	1	39356464913.7269	
12	VAR_1	01.10.2007 11:09	26.6432	1	39356464925.4861	
13	VAR_1	01.10.2007 11:09	26.064	1	39356464937.2454	
14	VAR_1	01.10.2007 11:09	26.6432	1	39356464948.9931	
15	VAR_1	01.10.2007 11:09	26.064	1	39356464960.7523	
16	VAR_1	01.10.2007 11:09	26.064	1	39356464972.5116	
17	VAR_1	01.10.2007 11:09	26.6432	1	39356464984.2593	
18	VAR_1	01.10.2007 11:09	26.064	1	39356464996.0185	
19	VAR_1	01.10.2007 11:09	26.064	1	39356465007.7662	

ARCHIV_10

ნახ. 2.15 ღუმელის მუშაობის ასახვა პროგრამა Excel - ში

2.3.15 ნახეთ ღუმელის ტემპერატურის გაზომვის რეზულტატი MatLab - ში. ამისათვის უნდა გაუშვათ MatLab - ი და გახსნათ დოკუმენტი ARCHIV_10, File/Open არჩევით … რეზულტატი წარმოდგენილ იქნება პროგრამა MatLab - ში (იხ. ნახ. 2.16):



ნახ. 2.16 ღუმელის მუშაობის ასახვა პროგრამა MatLab - ში

3.3.16 გამოიტანეთ დასკვნები გაწეული მუშაობის შესახებ.

2.4 საკონტროლო კითხვები და დავალებები

3.4.1 რა არის SCADA სისტემა?

3.4.2 განმარტეთ ტეგის მისამართის კონფიგურირების პრინციპი;

3.4.3 რა ტიპის არქივებს უჭერს მხარს Protool პაკეტის SCADA სისტემა?

3.4.4 რა არის ტეგები და რა ტიპის ტეგებს იცნობთ?

3.4.5 როგორ ხდება არქივების კონფიგურირება Protool პაკეტის SCADA სისტემაში?

<u>ანგარიშის აუცილებელი შემადგენლები</u>

1. პროგრამა STEP 7 – ზე.

2. პროექტის დამუშავების სკრინშოტი Protool პაკეტის SCADA სისტემაში?

2.4. ლაბორატორიული სამუშაო № 3

განივიმპულსური მოდულიაციის რეალიზაცია STEP 7- ში

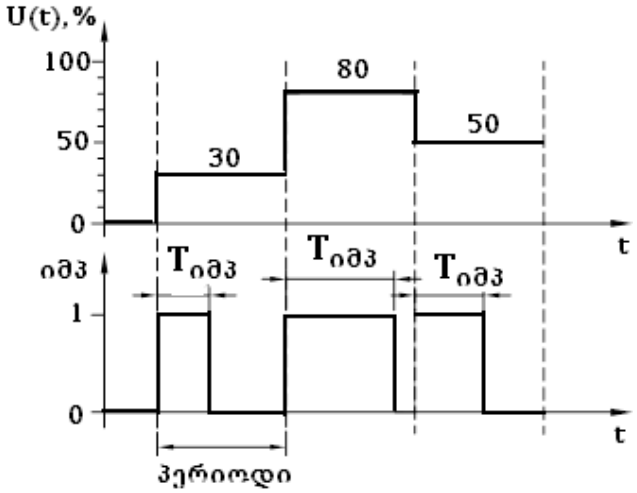
ლაბორატორიული ღუმელის გარდამავალი მახასიათებლის გადაღების ორგანიზაცია

3.1 სამუშაოს მიზანი: გიმ – ის რეალიზაციის გაცნობა STEP 7- ში.

3.2. თეორიული შესავალი

განივ – იმპულსური მოდულიაცია (გიმ)

განივ–იმპულსური მოდულიაციის გამოყენებით ხდება მმართველი $u(t)$ცვლადის ანალოგიური მნიშვნელობის გარდაქმნა იმპულსების თანმიმსდევრობაში "იმპ", პერიოდით $T_{პერიოდი}$. (იხ. ნახ 3.1).



ნახ. 3.1 გიმ–ის დროის დიაგრამა

ინსტრუქციების მიმოხილვა

ბიტური ლოგიკური ინსტრუქციები მუშაობენ ორ რიცხვთან – 1 და 0 – თან. ეს ორი რიცხვი ქმნის აღრიცხვის სისტემის ბაზისს, რომელსაც ეწოდება აღრიცხვის (ათვლის) ორობითი სისტემა. ციფრები 1 – სა და 0 – ს ეწოდებათ ორობითი ციფრები (bimary digits) ანუ უბრალოდ ბიტები. იმ სქემებთან მუშაობისას, რომლებიც გამოიყენებენ კონტაქტებსა და კოჭებს 1 – ის მნიშვნელობა შეესაბამება აქტიურ მდგომარეობას ანუ დენის გატარებას, ხოლო 0 – არააქტიურ მდგომარეობას ანუ დენის არ გატარებას.

ბიტური ლოგიკური ინსტრუქციები ინტერპრეტირებენ სიგნალებს 1 –ს და 0 – ს და კომბინირებენ მათზე ბულის ლოგიკის კანონებით. ეს კომბინაციები იძლევა რეზულტატს 1 – ს ან 0 – ს, რომელსაც ეწოდება ლოგიკური ოპერაციის რეზულტატი (RLO).

ნორმალურად ღია კონტაქტი (მისამართი)

აღნიშვნა:

<адрес>

---| |---

პარამეტრი	*მონაცემთა ტიპი*	*მეხსიერების არე*	*აღწერა*
<მისამართი>	BOOL	I,Q,M,L,D,T,C	ამოკითხვის ბიტის მისამართი

აღწერა:

ნორმალურად ღია კონტაქტი შეიკვრება მაშინ, როცა <მისამარი> – ით მითითებული ბიტის მდგომარეობა არის 1. თუ სიგნალის მდგომარეობა, რომელიც ამ მისამართითაა მითითებული ტოლია 1 – ის, მაშინ კონტაქტი შეკრულია და ლოგიკური ოპერაციის რეზულტატი ტოლია 1 – ის. თუ კი სიგნალის მდგომარეობა ამ მითითებული მისამართით ტოლია 0 – ის, მაშინ კონტაქტი გახსნილია და ბრძანება იძლევა ლოგიკური ოპერაციის რეზულტატს 0 – ს.

ნორმალურად შეკრული კონტაქტი (მისამართი)

აღნიშვნა:

<адрес>

---| / |---

პარამეტრი	*მონაცემთა ტიპი*	*მეხსიერების არე*	*აღწერა*
<მისამართი>	BOOL	I,Q,M,L,D,T,C	ამოკითხვის ბიტის მისამართი

აღწერა:

ნორმალურად შერკრული კონტაქტი შეკრავს წრედს მაშინ, როდესაც იმ ბიტის მდგომარეობა რომელიც "მისამართი" – თაა მითითებული, ტოლია 0 – ის. თუ სიგნალის მდგომარეობა, რომელიც ამ მისამართითაა მითითებული ტოლოა 0 – ის, მაშინ კონტაქტი შეკრულია და ლოგიკური ოპერაციის რეზულტატი ტოლია 1 – ის. თუ კი სიგნალის მდგომარეობა ამ მითითებული მისამართით ტოლია 1 – ის, მაშინ კონტაქტი გახსნილია და ბრძანება იძლევა ლოგიკური ოპერაციის რეზულტატს 0 – ის ტოლს.

ლოგიკური ოპერაციის რეზულტატის ინვერსია

აღნიშვნა:

---|NOT|---

აღწერა:

ლოგიკური ოპერაციის რეზულტატის ინვერსია ასრულებს ლოგიკური ოპერაციის რეზულტატისშეცვლას მისი საწინააღმდეგო მნიშვნელობით.

გამომავალი კოჭა

აღნიშვნა:

<адрес>

---( )

პარამეტრი	*მონაცემთა ტიპი*	*მეხსიერების არე*	*აღწერა*
<მისამართი>	BOOL	I,Q,M,L,D	მართული ბიტი

აღწერა:

გამომავალი კოჭა მუშაობს ისევე, როგორც ის კოჭა, რომელიც ჩართულია რელეურ კოტაქტური სქემის მართვის წრედში. თუ კი კოჭას მიეწოდება დენი, მაშინ ბიტი <მისამართი> ყენდება “1” – ში. თუ კი კოჭაზე არ მიეწოდება დენი, მაშინ ბიტი <მისამართი> ყენდება “0” – ში. გამომავალი კოჭა შეიძლება დაყენებულ იქნას მხოლოდ ლოგიკური წრედის მარჯვენა მხარეს. შეიძლება გამოყენებულ იქნას რამდენიმე გამომავალი კოჭა (მაქსიმუმ 16).

ლოგიკური ოპერაციის უარყოფითი ფრონტის გამოყოფა

აღნიშვნა:

<адрес>

---(N)---

პარამეტრი	*მონაცემთა ტიპი*	*მეხსიერების არე*	*აღწერა*
<მისამართი>	BOOL	I,Q,M,L,D	მისამართი უჩვენებს, თუ მეხსიერების რომელი ბიტი შეინახავს წინა ციკლის RLO – ს

აღწერა:

RLO - ს უარყოფითი ფრონტის გამოყოფა ინსტრუქციით ხდება მითითებულ მისამართზე 1 – დან 0 – ზე ცვლილების (უკანა ფრონტის) აღმოჩენა და მისი ასახვა RLO – ს დაყენებით 1 – ში ინსტრუქციის შესრულების შემდეგ. RLO – ს მიმდინარე მდგომარეობა დარდება ოპერანდის (ფრონტის მეხსიერების ბიტი) სიგნალის მდგომარეობას. თუ ოპერანდის სიგნალის მდგომარეობა ტოლია 1 – ის, ხოლო RLO ინსტრუქციის შესრულების წინ ტოლია 0 – ს, მაშინ RLO ინსტრუქციის შესრულების შემდეგ ტოლი იქნება 1 – ის (იმპულსი). ყველა სხვა შემთხვევაში კი RLO ტოლია 0 – ის. შესასვლელი RLO შემდეგ შეინახება მეხსიერების მითითებულ ბიტში.

ლოგიკური ოპერაციის დადებითი ფრონტის გამოყოფა

აღნიშვნა:

<адрес>

---(P)---

პარამეტრი	*მონაცემთა ტიპი*	*მეხსიერების არე*	*აღწერა*
<მისამართი>	BOOL	I,Q,M,L,D	მისამართი უჩვენებს, თუ მეხსიერების რომელი ბიტი შეინახავს წინა ციკლის RLO – ს

აღწერა:

RLO - ს დადებითი ფრონტის გამოყოფა ინსტრუქციით ხდება მითითებულ მისამართზე 0 – დან 1 – ზე ცვლილების (წინა ფრონტის) აღმოჩენა და მისი ასახვა RLO – ს დაყენებით 1 – ში ინსტრუქციის შესრულების შემდეგ. RLO – ს მიმდინარე მდგომარეობა დარდება ოპერანდის (ფრონტის მეხსიერების ბიტი) სიგნალის მდგომარეობას. თუ ოპერანდის სიგნალის მდგომარეობა ტოლია 0 –ს, ხოლო RLO ინსტრუქციის შესრულების წინ ტოლია 1 – ს, მაშინ RLO ინსტრუქციის შესრულების შემდეგ ტოლი იქნება 1 – ის (იმპულსი). ყველა სხვა შემთხვევაში კი RLO ტოლია 0 – ს. შესასვლელი RLO შემდეგ შეინახება მეხსიერების მითითებულ ბიტში.

ტაიმერები

ძირითადი ცნებები

ტაიმერები (Timers) - ეს მეხსიერების უჯრედებია, რომლებიც გამოიყენება ლოდინისა და მონიტორინგის ინტერვალების რეალიზაციისათვის.

ტაიმერები საშუალებას იძლევა პროგრამულად განხორციელდეს სინხრონიზაციის თანმიმდევრობები, ისეთები როგორიცაა ლოდინისა და მიყურადების ინტერვალები, ინტერვალების გაზომვები ანდა იმპულსების გენერირება.

არსებობს შემდეგი სახის ***ტაიმერების ტიპები:***

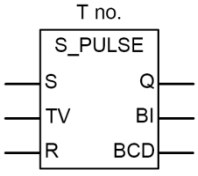
1) იმპულსური ტაიმერები (Puls timers);
2) იმპულსური ტაიმერები მეხსიერებით (Extendend puls timers);
3) ტაიმერები ჩართვის დაყოვნებით (On-delay timers);
4) ტაიმერები ჩართვის დაყოვნებით და დამახსოვრებით (Retentive on-delay timers);
5) ტაიმერები გამორთვის დაყოვნებებით (Off-delay timers).

ტაიმერი	*აღწერა*
S_PULSE ტაიმერი ”იმპულსი”	ის მაქსიმალური დრო, რომლის განმავლობაშიც გამომავალი სიგნალი რჩება 1 – ის ტოლი, ემთხვევა დაპროგრამებულ დროს T - ს. გამოსასვლელი ვარდება უფრო ადრე, თუ კი შესასვლელი სიგნალი იცვლის მდგომარეობას 0 – ზე.
S_PEXT ტაიმერი ”იმპულსი მეხსიერებით”	გამოსასვლელი სიგნალი ტოლია 1 – ის დაპროგრამებული დროის T - ს განმავლობაში, იმისდა მიუხედავად, თუ რა ხნის განმავლობაში რჩება 1 – ში შესასვლელი სიგნალი.
S_ODT ტაიმერი ”ჩართვის დაყოვნებით”	გამოსასვლელი სიგნალი დგება 1 – ში მხოლოდ დაპროგრამებული დროის T - ს გავლის შემდეგ, ამასთან, შესასვლელი სიგნალი ჯერ კიდევ უნდა იყოს 1 – ს ტოლი.
S_ODTS ტაიმერი ”ჩართვის დაყოვნებით და მეხსიერებით”	გამოსასვლელი სიგნალი დგება 1 –ში მხოლოდ დაპროგრამებული დროის T - ს გავლის შემდეგ, იმისდა მიუხედავად, თუ რა ხნის განმავლობაში რჩება 1 – ში შესასვლელი სიგნალი.
S_OFFDT ტაიმერი ”გამორთვის დაყოვნებით”	გამოსასვლელი სიგნალი ყენდება 1 –ში როდესაც დგება 1 – ში შესასვლელი სიგნალი და რჩება 1 – ის ტოლი ვიდრე ტაიმერი მუშაობს. დაპროგრამებული T დროის ათვლა იწყება მაშინ როდესაც შესასვლელი სიგნალი იცვლება 1 – დან 0 – ზე.

შემდეგში განვიხილავთ მხოლოდ S_PULSE და S_PEXT, რომლებიც გამოიყენება ლაბორატორული სამუშაოების შესრულების დროს.

S_PULSE : ტაიმერ ”იმპულსი”– ის პარამეტრების მიცემა და გაშვება

აღნიშვნა:



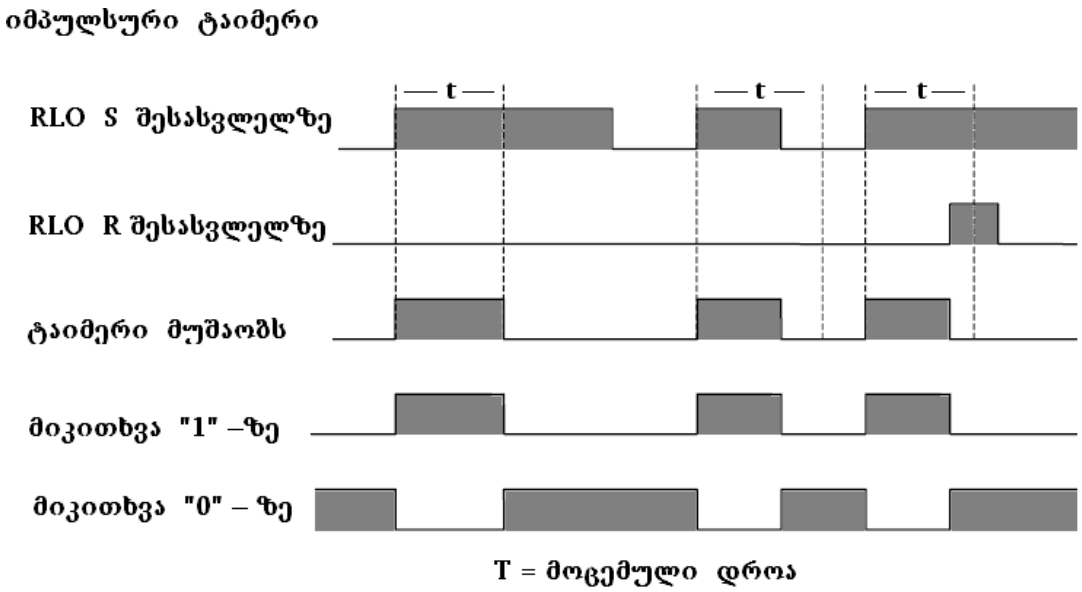
პარამეტრი	*მონაცემთა ტიპი*	*მეხსიერების არე*	*აღწერა*
T no.	TIMER	T	ტაიმერის ნომერი. ნომრების დიაპაზონი დამოკიდებულია CPU – ზე
S	BOOL	I,Q,M,D,L	გაშვების შესასვლელი
TV	S5TIME	I,Q,M,D,L	დროის დაყენება (0 – დან 9990 – მდე)
R	BOOL	I,Q,M,D,L	ჩამოგდების შესასვლელი
BI	WORD	I,Q,M,D,L	მონარჩენი დრო (მნიშვნელობა ორობით კოდში)
BCD	WORD	I,Q,M,D,L	მონარჩენი დრო (მნიშვნელობა BCD ფორმატში)
Q	BOOL	I,Q,M,D,L	ტაიმერის მდგომარეობა

აღწერა:

S_PULSE : (S5 ტაიმერი ”იმპულსი”) გაუშვებს მოცემულ ტაიმერს წინა ფრონტით (სიგნალის მდგომარეობის ცვლილება 0 – დან 1 – ზე) გაშვების შესასვლელზე (S). ტაიმერის გაშვებისათვის ყოველთვის აუცილებელია სიგნალის ცვლილება. ტაიმერი აგრძელებს მუშაობას იმ დროის განმავლობაში, რომელიც მოცემულია TV შესასვლელზე, მანამ, სანამ სიგნალის მდგომარეობა S შესასვლელზე რჩება 1 – იანის მდგომარეობაში. სანამ ტაიმერი მუშაობს, Q გამოსასვლელზე მიკითხვა მაღალ დონეზე იძლევა ლოგიკური ოპერაციის რეზულტატს 1 – ს. თუ S შესასვლელზე სიგნალი იცვლება 1 – დან 0 – ზე მოცემული დროის დამთავრებამდე, ტაიმერი ჩერდება, ხოლო Q გამოსასვლელზე მიკითხვა მაღალ დონეზე იძლევა ლოგიკური ოპერაციის რეზულტატს 0 – ს. თუ კი ტაიმერის მუშაობის დროს ჩამოყრის შესასვლელზე (R) - ზე არსებული სიგნალი შეიცვლის მდგომარეობას 1 – დან 0 – ზე, მაშინ ტაიმერი ვარდება. ეს ცვლილება ჩამოაგდებს დროს ნულში დროის ბაზაშიც კი. ერთიანი R შესასვლელზე არ ახდენს არავითარ გავლენას თუ ტაიმერი არ მუშაობს.

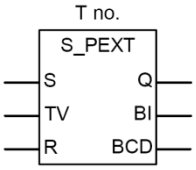
დროის მიმდინარე მნიშვნელობა შეიძლება წაკითხულ იქნას BI და BCD გამოსასვლელებზე. დროის მნიშვნელობა BI გამოსასვლელზე წარმოდგენილია ორობით ფორმატში, ხოლო BCD გამოსასვლელებზე – ორობით–ათობით ფორმატში. მიმდინარე დრო ტოლია TV შესასვლელზე მიცემული საწყისი მნიშვნელობისა და ტაიმერის გაშვების მომენტიდან გასული დროის შორის სხვაობისა. დროითი დიაგრამები იხ. ნახ. 3.2 – ზე

დროითი დიაგრამები



ნახ. 3.2 S_PULSE ტაიმერის მუშაობის დროის დიაგრამები

S_PEXT: ტაიმერ ”იმპულსი მეხსიერებით”– ის პარამეტრების მიცემა და გაშვება
აღნიშვნა:



პარამეტრი	*მონაცემთა ტიპი*	*მეხსიერების არე*	*აღწერა*
T no.	TIMER	T	ტაიმერის ნომერი. ნომრების დიაპაზონი დამოკიდებულია CPU – ზე
S	BOOL	I,Q,M,D,L	გაშვების შესასვლელი
TV	S5TIME	I,Q,M,D,L	დროის დაყენება (0 – დან 9990 – მდე)
R	BOOL	I,Q,M,D,L	ჩამოგდების შესასვლელი
BI	WORD	I,Q,M,D,L	მონარჩენი დრო (მნიშვნელობა

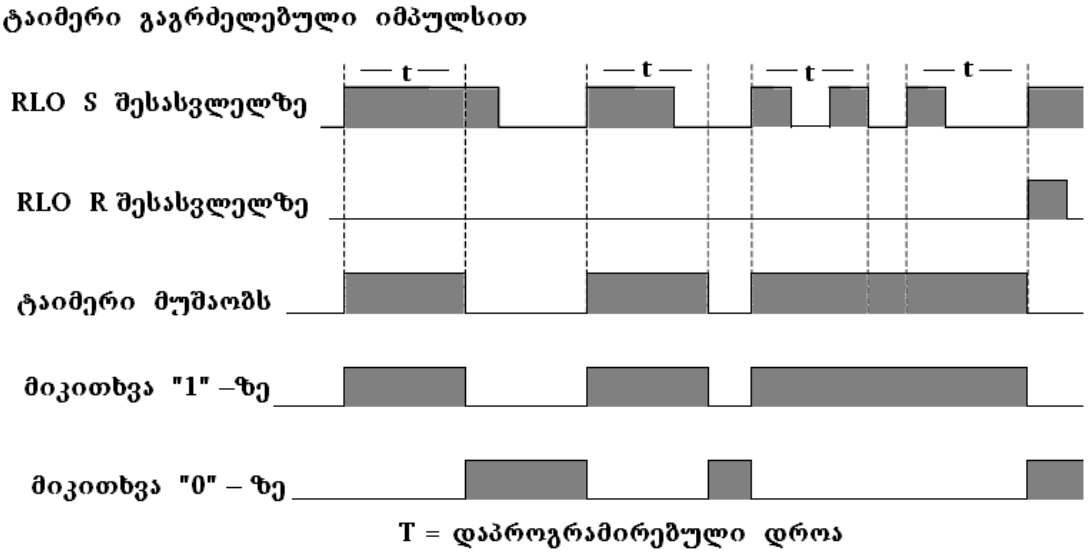
			ოროზით კოდში)
BCD	WORD	I,Q,M,D,L	მონარჩენი დრო (მნიშვნელობა BCD ფორმატში)
Q	BOOL	I,Q,M,D,L	ტაიმერის მდგომარეობა

აღწერა:

S_PEXT: (S5 ტაიმერი "გაგრძელებული იმპულსი") გაუშვებს მოცემულ ტაიმერს (სიგნალის მდგომარეობის ცვლილება 0 – დან 1 – ზე) გაშვების შესასვლელზე (S). ტაიმერის გაშვებისათვის ყოველთვის აუცილებელია სიგნალის ცვლილება. ტაიმერი აგრძელებს მუშაობას იმ დროის განმავლობაში, რომელიც მოცემულია TV შესასვლელზე, იმ შემთხვევაშიც კი, თუ სიგნალის მდგომარეობა S შესასვლელზე გადადის 0 – ის მდგომარეობაში მოცემული დროის გასვლემდე. სანამ ტაიმერი მუშაობს, Q გამოსასვლელზე მიკითხვა იძლევა ლოგიკური ოპერაციის რეზულტატს 1 – ს. ტაიმერი თავიდან გაიშვება მოცემული დროით თუ S შესასვლელზე სიგნალის მდგომარეობა იცვლება 0 – დან 1 – ზე ტაიმერის მუშაობის დროს. თუ კი ტაიმერის მუშაობის დროს ჩამოყრის შესასვლელზე (R) - ზე არსებული სიგნალი შეიცვლის მდგომარეობას 1 – დან 0 – ზე, მაშინ ტაიმერი ვარდება. ეს ცვლილება ჩამოაგდებს დროს ნულში დროის ბაზაშიც კი. 1–ი R შესასვლელზე არ ახდენს არავითარ გავლენას თუ ტაიმერი არ მუშაობს.

დროის მიმდინარე მნიშვნელობა შეიძლება წაკითხულ იქნას BI და BCD გამოსასვლელებზე. დროის მნიშვნელობა BI გამოსასვლელზე წარმოდგენილია ოროზით ფორმატში, ხოლო BCD გამოსასვლელებზე – ოროზით–ათობით ფორმატში. მიმდინარე დრო ტოლია TV შესასვლელზე მიცემული საწყისი მნიშვნელობისა და ტაიმერის გაშვების მომენტიდან გასული დროის შორის სხვაობისა. დროითი დიაგრამები იხ. ნახ. 3.1 – ზე.

დროითი დიაგრამები



ნახ. 3.3 S_PEXT ტაიმერის მუშაობის დროის დიაგრამები

ტაიმერის კომპონენტები

დროის მნიშვნლობა:

0 – დან მე 9 – ს ჩათვლით ბიტები შეიცავს დროის მნიშვნელობას ორობით კოდში. დროის მნიშვნელობა მიუთითებს დროითი მონაკვეთების რაოდენობას. როდესაც ტაიმერი გააქტიურდება, დროის მნიშვნელობა მცირდება ერთი ერთეულით, იმ ინტერვალით, რომელიც დაყენებულია დროის ბაზით. დროის მნიშვნელობა მცირდება მანამ, სანამ იგი არ გახდება ნულის ტოლი. დროის მნიშვნელობა შეიძლება მიცემულ იქნას ორობით, თექვსმეტობით ან ორობით–ათობთ კოდებით (BCD).

დროის მნიშვნელობის ჩატვირთვა შეიძლება შემდეგი სინტაქსისის გამოყენებით:

- S5T#aH_bM_cS_dMS,

სადაც: a = საათებია;

b = წუთებია:

c = წამებია;

d = მილიწამები.

მაქსიმალური დრო, რომელიც შეიძლება შეყვანილ იქნას, შეადგენს 9 990 წამს ან 2H_46M_30S.

S5TIME#4S = 4 წამს

s5t#2h_15m = 2 საათსა და 15 წუთს;

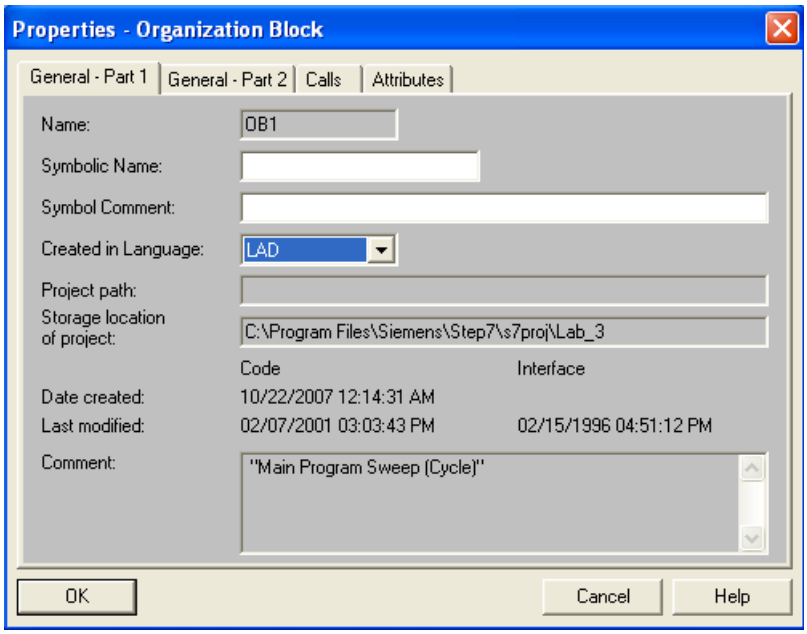
S5T#1H_12M_18S = 1 საათს, 12 წუთს და 18 წამს.

3.3 სამუშაოს შესრულების თანმიმდევრობა

3.3.1 გაუშვით პროგრამა SIMATIC Manager;

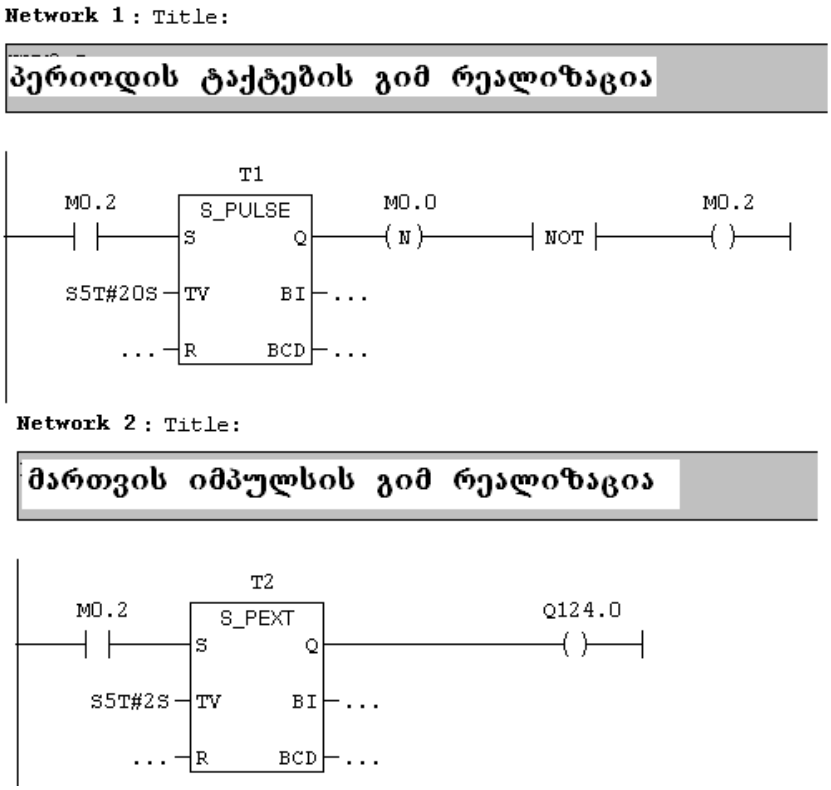
3.3.2 შექმენით ახალი პროექტი და დააკონფიგურირეთ კონტროლერი STEP 7 – ში;

3.3.3 პროექტის ფანჯრის მარცხენა ნაწილში გახსენით ხე Blocks. გახსენით საორგანიზაციო ბლოკი OB1. გამოჩენილ ფანჯარაში Properties აირჩიეთ ენა, რომელზდაც დავწერთ პროგრამას: LAD (კონტაქტურ–სარელეო სქემა) (იხ. ნახ. 3.4).



ნახ. 3.4 OB1 საორგანიზაციო ბლოკის პარამეტრების ფანჯარა

3.3.4 გამონთებულ ფანჯარაში დაწერეთ დასმული ამოცანის შესრულების პროგრამა (გარდამავალი მახასიათებლების გადაღების რეალიზაცია) (იხ. ნახ. 3.5):



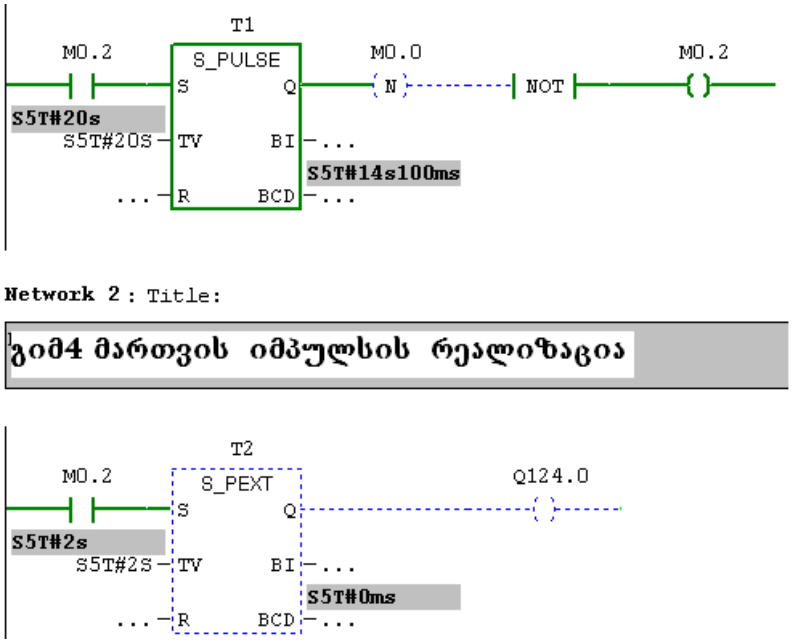
ნახ. 3.5 დასმული ამოცანის შესრულების პროგრამა

3.3.5 შეინახეთ ცვლილებები OB1 - ში, მთავარ მენიუში File/Save – ის არჩევით.

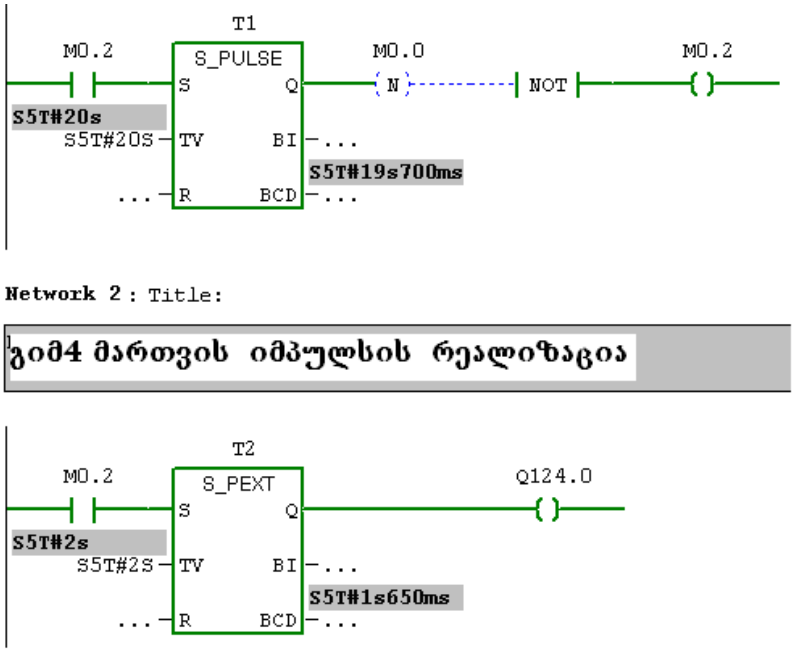
3.3.6 პროექტის ფანჯრის მარცხენა ნაწილში ჩაკეცეთ ყველა "–" SIMATIC 300(1) – მდე. ინსტრუმენტების პანელზე დააჭირეთ ღილაკზე Download  და შემდეგ დაეთანხმეთ ყველაფერზე.

3.3.7 ჩატვირთვეთ კონტროლერი ან Simulating (დაყაენეთ ტუმბლერი Ran - ში)

3.3.8 დაბრუნდით OB1 ბლოკში და ინსტრუმენტების პანელზე დააჭირეთ ღილაკს Monitor (on/off)  . თუ ყველაფერი სწორად იქნა შესრულებული, მაშინ პროგრამა იმუშავებს ისე, როგორც ნაჩვენებია ნახაზებზე 3.6 და 3.7.



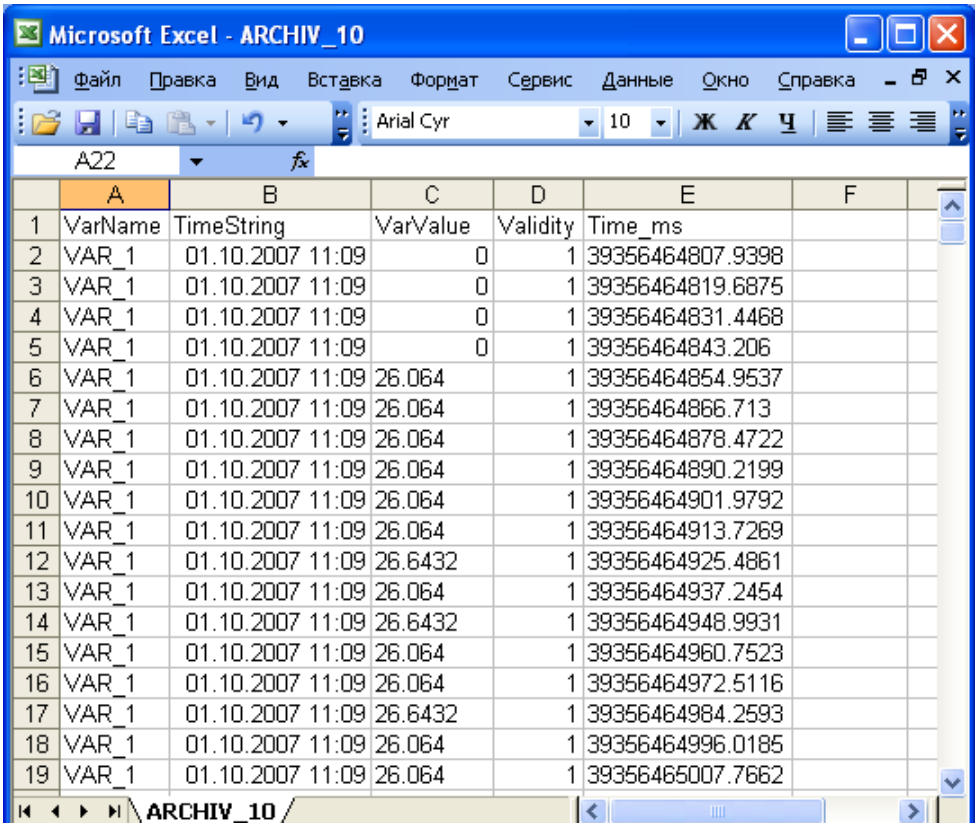
ნახ. 3.6 ითვლის ტაიმერი T1



ნახ. 3.7 ითვლან ტაიმერიები T1 და T2

3.3.9 თუ გვინდა პროგრამის გადარედაქტირება უნდა ავუშვათ ღილაკს Monitor (on/off)  და გამორთეთ კონტროლერი ან Simulating – ი (დააყენეთ ტუმბლერი STOP-ში). გადარედაქტირების შემდეგ გაიმეორეთ პუნქტები 3.3.8 – 3.3.10.

3.3.10 შექმენით არქივი ProTool – ის გამოყენებით (იხ. ნახ. 3.8). არქივის ფორმირების პრინციპი აღწერილია ლაბორატორულ სამუშაო № 2 –ში.



Microsoft Excel - ARCHIV_10

Файл Правка Вид Вставка Формат Сервис Данные Окно Справка

Arial Cyr 10

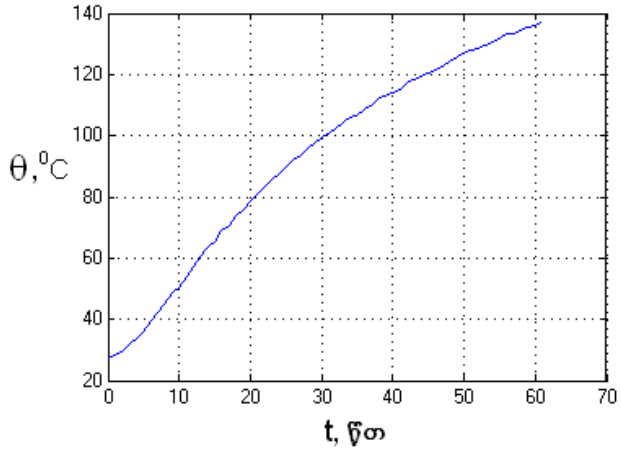
A22

	A	B	C	D	E	F
1	VarName	TimeString	VarValue	Validity	Time_ms	
2	VAR_1	01.10.2007 11:09	0	1	39356464807.9398	
3	VAR_1	01.10.2007 11:09	0	1	39356464819.6875	
4	VAR_1	01.10.2007 11:09	0	1	39356464831.4468	
5	VAR_1	01.10.2007 11:09	0	1	39356464843.206	
6	VAR_1	01.10.2007 11:09	26.064	1	39356464854.9537	
7	VAR_1	01.10.2007 11:09	26.064	1	39356464866.713	
8	VAR_1	01.10.2007 11:09	26.064	1	39356464878.4722	
9	VAR_1	01.10.2007 11:09	26.064	1	39356464890.2199	
10	VAR_1	01.10.2007 11:09	26.064	1	39356464901.9792	
11	VAR_1	01.10.2007 11:09	26.064	1	39356464913.7269	
12	VAR_1	01.10.2007 11:09	26.6432	1	39356464925.4861	
13	VAR_1	01.10.2007 11:09	26.064	1	39356464937.2454	
14	VAR_1	01.10.2007 11:09	26.6432	1	39356464948.9931	
15	VAR_1	01.10.2007 11:09	26.064	1	39356464960.7523	
16	VAR_1	01.10.2007 11:09	26.064	1	39356464972.5116	
17	VAR_1	01.10.2007 11:09	26.6432	1	39356464984.2593	
18	VAR_1	01.10.2007 11:09	26.064	1	39356464996.0185	
19	VAR_1	01.10.2007 11:09	26.064	1	39356465007.7662	

ARCHIV_10

ნახ. 3.8რეზულტატების ასახვა პროგრამა Excel – ში.

3.3.11 ააგეთ გარდამავალი პროცესის მრუდის გრაფიკი (ნახ. 3.9). ეს შეიძლება რეალიზებულ ინას Microsoft Excel ანდა Matlab გამოყენებითი პროგრამების პაკეტების გამოყენებით.



ნახ. 3.9 გარდამავალი პროცესის მრუდის გრაფიკი

3.3.12 ჩამოაყალიბეთ დასკვნები ჩატარებული სამუშაოს შესახებ.

3.4. საკონტროლო კითხვები და დავალებები

3.4.1 რა არის გიმ და რა არის უწყვეტი სიგნალის ამ სახით მოდულიაციის პრინციპი?

3.4.2 რისთვის გამოიყენება ინსტრუქცია ”გამომავალი კოჭა” STEP 7 – ში;

3.4.3 ახსენით ”ლოგიკური ოპერაციის დადებითი ფრონტის გამოყოფა” – ს ინსტრუქციის მუშაობის პრინციპი.

3.3.4 ახსენით ”S_PULSE””S_PEXT”და ტაიმერების მოქმედების პრინციპი.

<u>ანგარიშის აუცილებელი შემადგენლები</u>

1. პროგრამა გიმSTEP 7 – ზე.
2. გარდამავალი პროცესის მნიშვნელობების არქივი.
3. გარდამავალი პროცესის გრაფიკი.

2.5. ლაბორატორიული სამუშაო №4

მართვის ობიექტის იდენტიფიკაცია

4.1 სამუშაოს მიზანი: მართვის ობიექტის იდენტიფიკაციის მეთოდიკის ათვისება.

4.2 თეორეიული შესავალი

ობიექტის იდენტიფიკაციის ამოცანა მდგომარეობს მისი იმ მათემატიკური მოდელის მოძებნაში, რომელიც ყველაზე უკეთ არწერს მის დინამიურ თვისებებს. თუ კი საბოლოო მიზანს წარმოადგენს ოპყიმალური მართვის მოძებნა, მაშინ საკმარისია, როდესაც მოცემული შესასვლელის შემთხვევაში მოდელის გამოსასვლელი რომ ექვივალენტური იყოს სისტემის გამოსასვლელის. ასეთ შემთხვევაში არაა აუცილებლობა იმაში, რომ მოდელის სტრუქტურა და პარამეტრები ემთხვეოდეს ფიზიკური სისტემის სტრუქტურასა და პარამეტრებს.

ზოგადად კი მათემატიკური მოდელის აგება მდგომარეობს იმაში, რომ განსაზღვრულ იყოს ობიექტის სტრუქტურა, მოძებნილ იყოს პარამეტრების მნიშვნელობები და თუ ეს აუცილებლობას წარმოადგენს – მოძებნილ იყოს დამოკიდებული პარამეტრების მნივნელობები, მაგ. მდგომარეობათა ცვლადები.

უმეტეს შემთხვევებში, იდენტიფიკაციის მეთოდები მოცემული შეზღუდული სიზუსტის პირობებში არ გვაძლევენ რთული მოდელის აგების შესაძლებლობას, რომელიც სტრუქტურისა და პარემეტრების მიხედვით ექვივალენტური იქნება რეალურ ობიექტთან. მიუხედავად ამისა, ეს ფაქტი სრულიადაც არ გვიშლის ხელს გამოვიყენოთ შემდგომში ასეთი მოდელი, თუ კი იგი ასახავს ობიექტის არსებით მხარეებს. უფრო მეტიც, მისი სიმარტივიდან გამომდინარე, ასეთი მოდელი უფრო მეტად გამოსადეგია შემდგომი გამოყენებებისათვის.

იდენტიფიკაცია შეიძლება განხორციელდეს ანდა ფიზიკურ–მათემატიკური ანალიზის ანდა ექსპერიმენტალური მეთოდით ანალიზის გამოყენების გზით.

ფიზიკურ–მათემატიკური ანალიზის მეთოდით გამოყენებისას გამოდიან მოდელის კონსტრუქციული მონაცემებითა და ძირითადი პროცესების მათემატიკური აღწერით, რომლებსაც ადგილი აქვთ შესასწავლ ობიექტში. ამ შემთხვევაში იღებენ ალგებრული და დიფერენციალური განტოლებების სისტემას, რომლებიც შეიცავენ როგორც შესასვლელ და გამოსასვლელ ცვლადებს, ასევე მდგომარეობათა ცვლადებს. ამ განტოლებებში ზოგჯერ რთავენ ობიექტის ჭარბ შიგა ცვლადებს, რომლებიც პრინციპში შეიძლება არ იქნას მხედველობაში მიღებული.

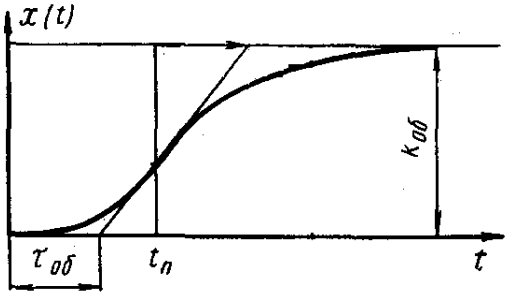
ექსპერიმენტალური მეთოდით ანალიზის გამოყენების დროს ჩვეულებრივად ძებნიან მდგრად მდგომარეობაში მყოფი ობიექტის მათემატიკურ მოდელს, მისი შესასვლელი და გამოსასვლელი სიდიდეების გაზომვების გზით.

სამუშაოში ხდება გამოკვლევა იმ სისტემისა, რომელიც შედგება ორი თბური ობიექტისაგან – ღუმელისა და თერმოწყვილისაგან. თვითოეული ამ ობიექტთაგანი შეიძლება აღიწეროს მათემატიკურად პირველი ხარისხის აპერიოდული კვანძითა და

დაგვიანების კვანძით, შესაბამისად მთელი სისტემის საერთო მოდელი შეიძლება მოძებნილ იყოს ქვემოთ წარმოდგენილი სახით:

$$W_{об}(s)=\frac{1}{T_1\cdot s+1}\cdot\frac{K_{об}}{T_2\cdot s+1}\cdot e^{(-\tau_{об}\cdot s)}$$

ასეთი ობიექტებისათვის, მათში შეშფოთებების შეტანით ერთეულოვანი ნახტომის სახით, შეიძლება მოღებულ იქნას გადასვლის ფუნქცია, როგორც ნაჩვენებია ნახ. 4.1–ზე. იგი წარმოადგენს მონოტორულ მრუდს, რომლის მახასიათებელი წერტილია გადაღუნვის წერტილი, რომელიც შეესაბამება მეორე წარმოებულის ნიშნის ცვლილებას.



ნახ. 4.1 მეორე ხარისხის სტატიური ობიექტის გადასვლის მახასიათებელი

ასეთი ობიექტის გადაცემის ფუნქცია შეიძლება განხილულ იქნას როგორც ორი აპერიოდული ობიექტის გადაცემის ფუნქციების ნამრავლი, რომელთა დროის მუდმივები T_1 და T_2 – ია შესაბამისად. გადასვლის ფუნქცია განისაზღვრება გამოსახულებით:

$$Y_{мод}(t_i) = y(0)+K_{об}\left(1-\frac{T_1}{T_1-T_2}\cdot e^{\frac{-(t_i-\tau_{об})}{T_1}}+\frac{T_2}{T_1-T_2}\cdot e^{\frac{-(t_i-\tau_{об})}{T_2}}\right)$$

სტატიკური ობიექტის დინამიური პარამეტრების მოიახლოვებითი განსაზღვრისათვის (დაგვიანება τоб, გადაცემის კოეფიციენტი Коб), გამომავალი სიდიდის გადაღუნვის წერტილში ავლებენ მხებს და აგრძელებენ მას გამომავალი სიდიდის საწყისი მნიშვნელობის ხაზთან გადაკვეთამდე (აბსცისების ღერძი). დროის მონაკვეთი შეშფოთების შეტანიდან მხების გადაკვეთამდე ღერძთან მიმართებაში, განსაზღვრავს ობიექტის დაგვიანებას.

სტატიკური ობიექტის გადაცემის კოეფიციენტი წარმოადგენს ობიექტის გამომავალი სიდიდის ცვლილებას საწყისი მდგომარეობიდან ახალ დამყარებულ მდგომარეობაში გადასვლის დროს, შესასვლელზე შეშფოთების შეტანასთან მიმართებაში:

$$K_{об} = \frac{x_{\infty} - x_0}{\Delta x_{вх}},$$

სადაც x_0 –გამომავალი სიდიდის მნიშვნელობაა საწყის დამყარებულ მდგომარეობაში;

x_{∞} – იგივე, ახალი დამყარებული მნიშვნელობისათვის;

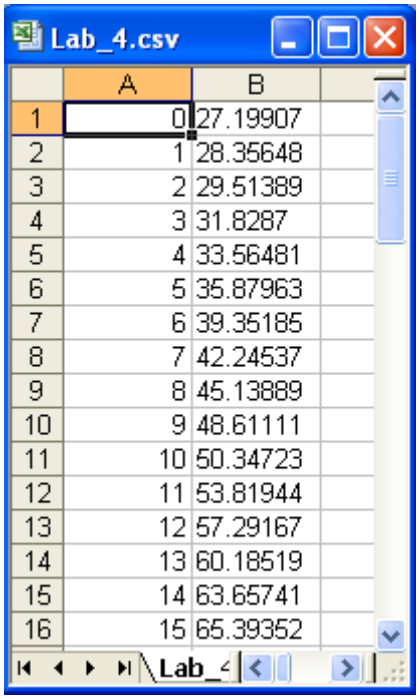
$\Delta x_{вх}$ – შესატანი შეშფოტების სიდიდეა.

დროის მუდმივების მნიშვნელობების განსაზღვრისათვის შეიძლება გამოყენებულ იქნას ოპტიმიზაციის სხვადასხვა მეთოდები, როგორიცაა გრადიენტული მეთოდით, მნიშვნელობების უბრალო გადარჩევით ფუნქციონალის მაქსიმუმთან მიღწევის გზით, რომელიც წარმოადგენს ცდომილებათა კვადრატების ჯამს:

$$F = \sum_{i=1}^{N} (Y_{мод}(t_i) - Y_{эксп}(t_i))^2$$

4.3 სამუშაოს შესრულების თანმიმდევრობა

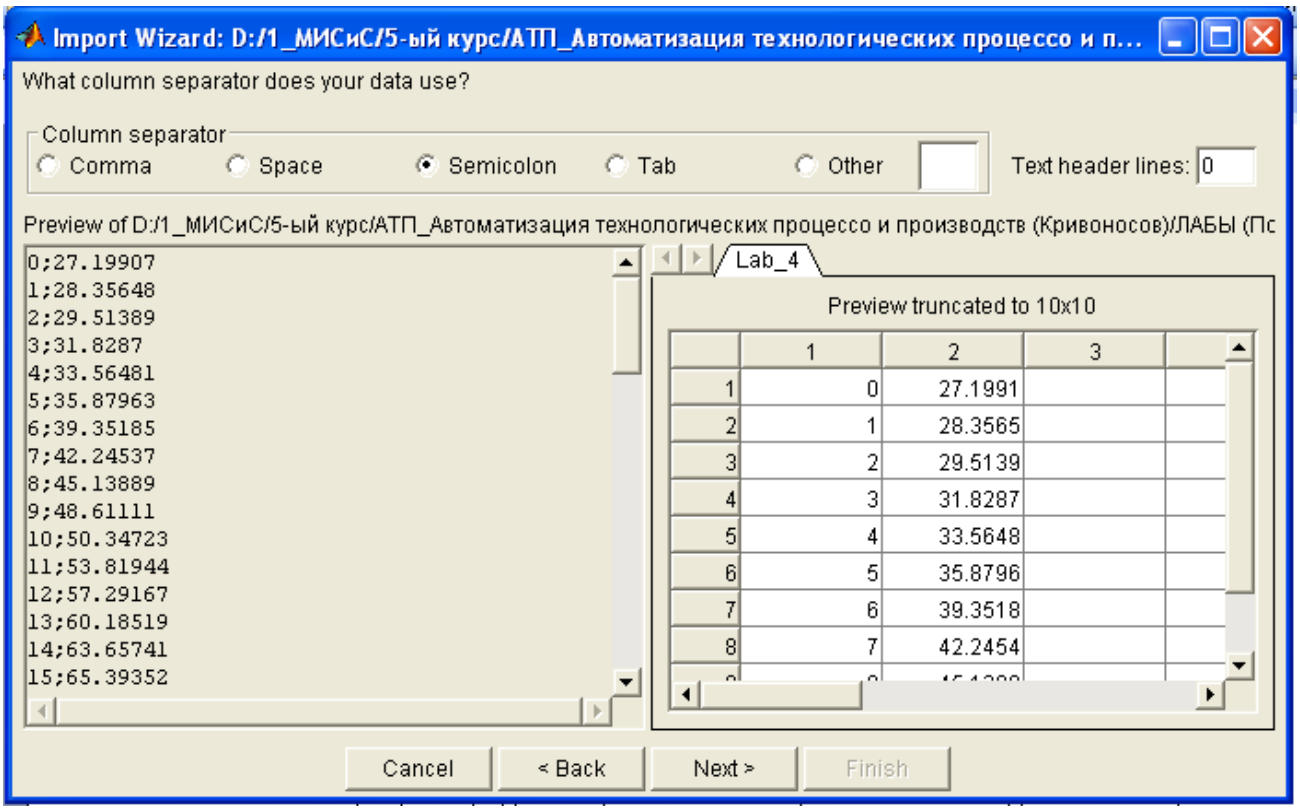
4.3.1 გახსენით №3 ლაბორატორიულ სამუშაოში შექმნილი არქივიMicrosoft Office Excelპროგრამული პაკეტის საშუალებით. მიიყვანეთ დოკუმენტი იმ სახემდე, რაც გამოსახულია ნახ. 4.2 – ზე.



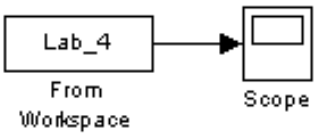
ნახ. 4.2Microsoft Office Excelდოკუმენტი. ღუმელის ტემპერატურის დამოკიდებულება დროში.

4.3.2 გახსენით შეცვლილი არქივი Matlab –ში File/Open ოპერაციის შესულებით. გაღებულ ფანჯარაში (ნახ. 4.3) აირჩიეთ Next/Finish.

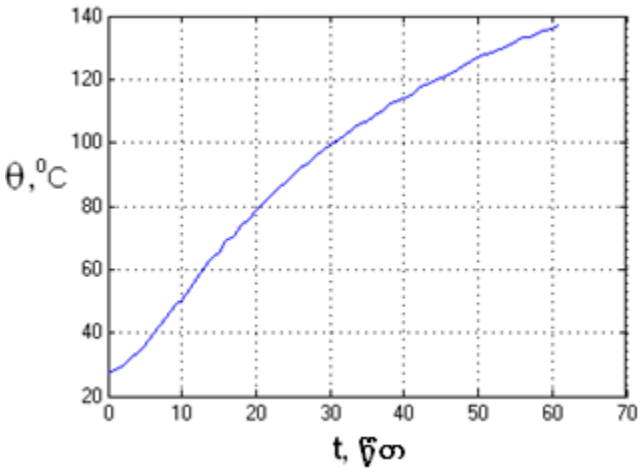
4.3.3 Matlab –ში გაუშვით Simulink, ინსტრუმენტების პანელზე  პიქტოგრამის დაჭერით. შექმენით ახალი დოკუმენტი File/New/Model ოპერაციის შესრულებით.
ააწყეთ სქემა, რომელიც გამოსახულია ნახ. 4.4 – ზე. From Workspace ბლოკში Data ველში შეიყვანეთ არქივის სახელი “Lab_4”. გაუშვით პროექტი, ინსტრუმენტების პანელზე Start Simulation  ღილაკზე თითის დაჭერით. Scope – ის გაღებით გაანალიზეთ გარდამავალი პროცესის გრაფიკი (იხ. ნახ. 4.5).



ნახ. 4.3 ღუმელის ტემპერატურის დამოკიდებულება დროშიMatlab – ით.

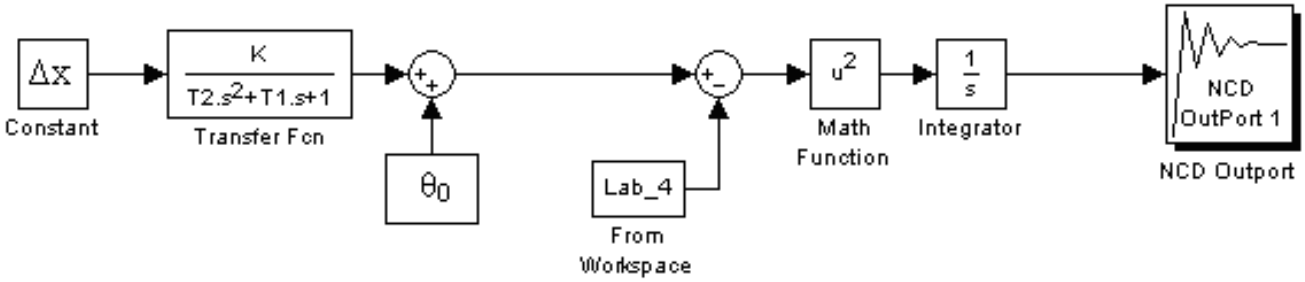


ნახ. 4.4 გარდამავალი პროცესის გრაფიკის აგების სქემა

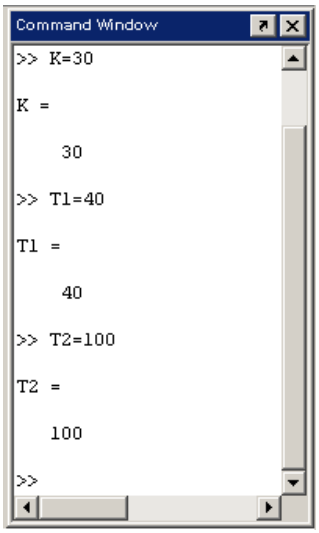


ნახ. 4.5 გარდამავალი პროცესის გრაფიკი

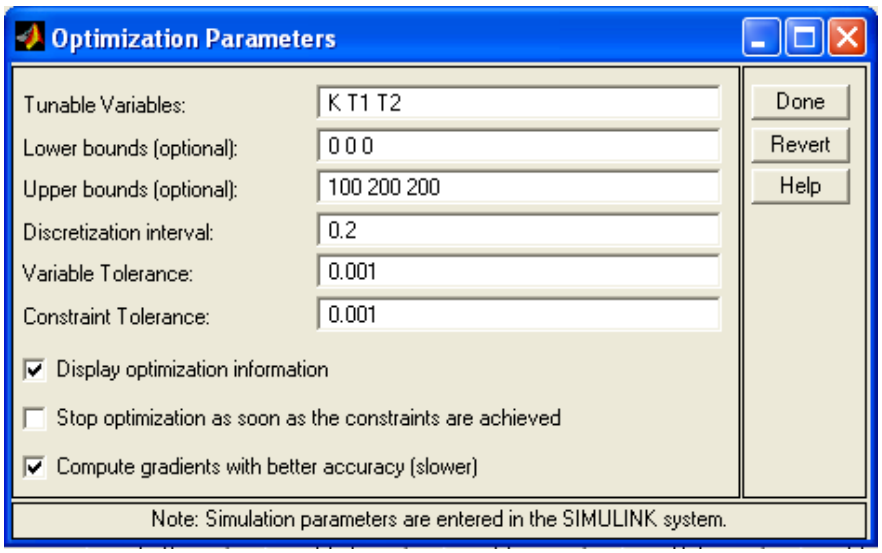
4.3.4 ობიექტის იდენტიფიკაციისათვის საჭირო იქნება ნახ. 4.6 – ზე ნაჩვენები სქემის აწყობა. აქ Δx – მართვის სიგნალია % – ებში, ϑ_0 -ტემპერატურის საწყისი მნიშვნელობა oC – ებში. ამ სქემისათვის დააყენეთ მამოდულირებელი დროის დისკრეტობა მონაცემთა მოხსნის დისკრეტობის ტოლი (ჩანართი Simulation → Simulation parameters → Solver option → Fixed step + Ode 5 (მეზობელ მენიუში)) გარდამავალი მახასიათებლის მოხსნისათვის.შემდეგ Matlab –ის ფანჯარაში Command Window K, T1 и T2 შეიყვანეთ პარამეტრების მნიშვნელობები (იხ. ნახ. 4.7). ამის შემდეგ გახსენით ბლოკი NCD Outport,აირჩიეთ მენიუში Optimization/Parameteres, გამჩენილ ფანჯარაში (იხ. 4.8) შეიყვანეთ მაოპტიმიზირებელი პარამეტრები, მათი ქვედა და ზედა ოპტიმიზაციის ზღურბლები. გაუშვით პროექტი, ინსტრუმენტების პანელზე ღილაკ Start –ზე დაჭერით (იხ. ნახ. 4.9). შემდეგში Matlab –ში ფანჯარაში Command Window ნახეთ K, T1 и T2 პარამეტრების მნიშვნელობები (იხ. ნახ. 4.10).



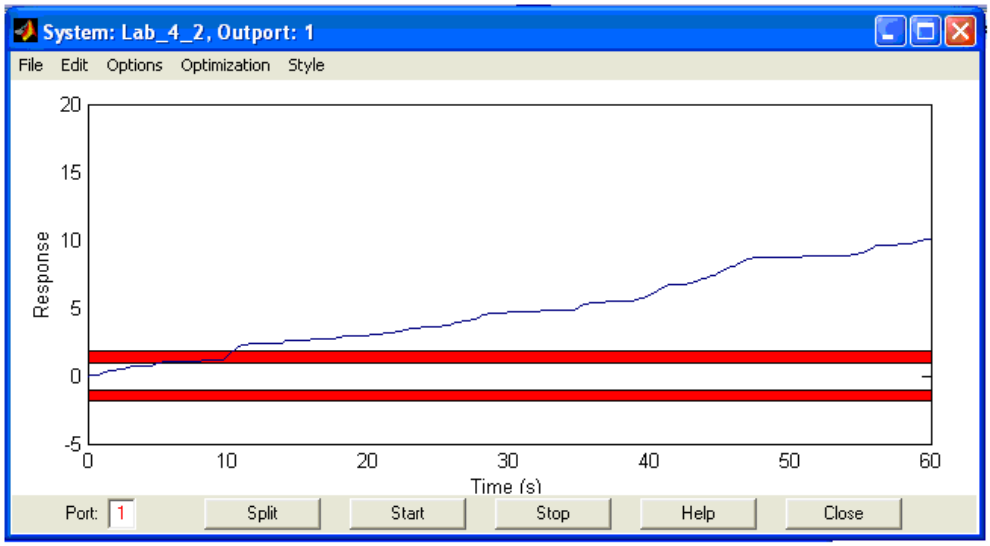
ნახ. 4.6 ობიექტის იდენტიფიკაციის სქემა



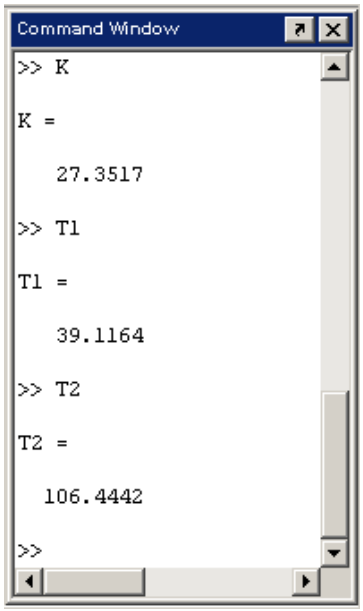
ნახ. 4.7 K, T1 и T2 პარამეტრების შეყვანა



ნახ. 4.8 K, T1 и T2 ოპტიმიზაციუს პარამეტრების განსაზღვრა

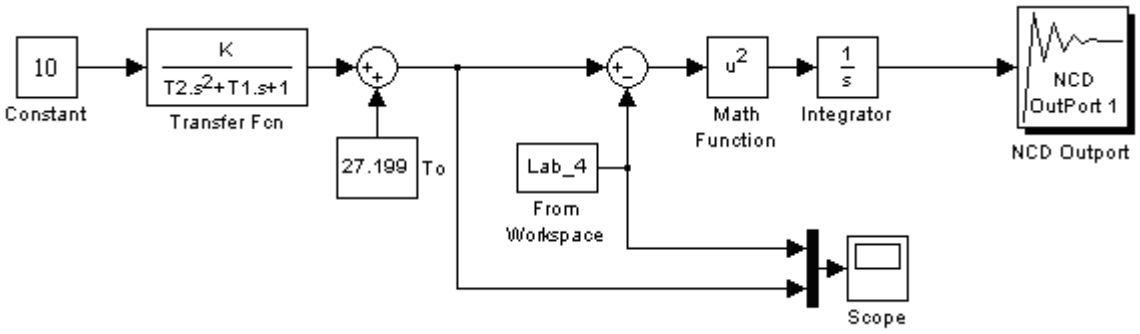


ნახ. 4.9NCD Outport ბლოკის მუშაობის რეზულტატი

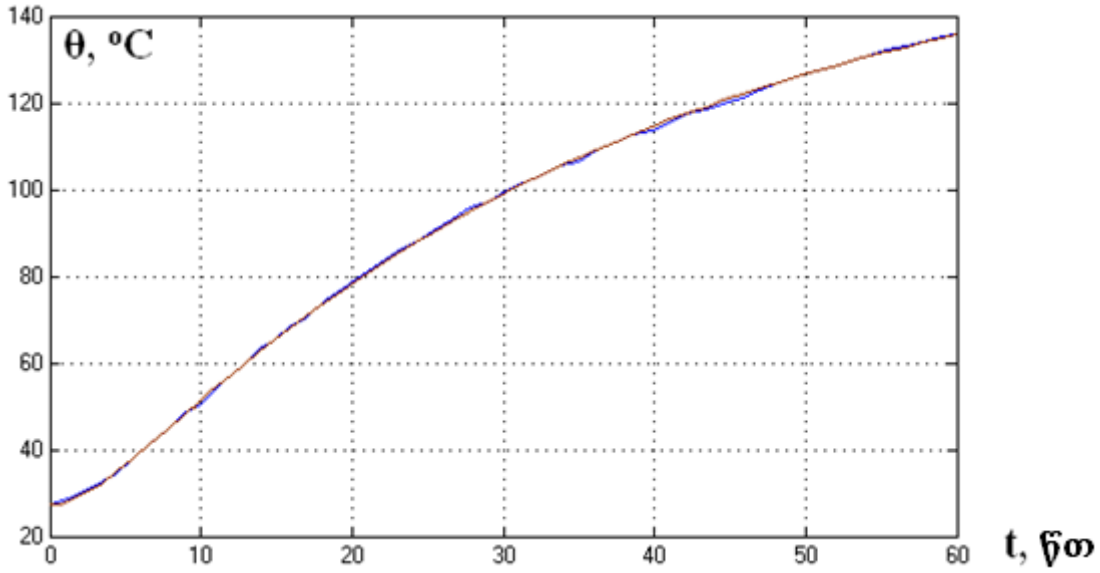


ნახ. 4.10 K, T1 и T2 ოპტიმიზირებული პარამეტრების ნახვა

4.3.5 Scope–ის გამოყენებით ნახეთ გარდამაქვალი პროცესის გრაფიკები, რომლებიც აგებულია (ლაბორატორიულ სამუშაო №3–ში) შესრულებული გაზომვებისა და მოდელური მნიშვნელობების მიხედვით. (იხ. ნახ. 4.111). გაანალიზეთ მიღებული რეზულტატები (იხ. ნახ. 4.12).



ნახ. 4.11 გარდამავალი პროცესის გრაფიკების ნახვის სქემა, გაზომვილი და მოდელური მნიშვნელობების მიხედვით



ნახ. 4.12 გარდამავალი პროცესის გრაფიკები, რომლებიც აგებულია გაზომვილი და მოდელური მნიშვნელობების მიხედვით

4.3.6 გამოიტანეთ დასკვნები შესრულებული სამუშაოს მიხედვით.

4.4. საკონტროლო კითხვები და დავალებები

4.4.1 რას ეწოდება ობიექტის იდენტიფიკაცია?

4.4.2 როგორ განისაზღვრება დაგვიანება?

4.4.3როგორი სახე აქვს მეორე რიგის აპერიოდული კვანძის გარდამავალ მახასიათებელს?

<u>ანგარიშის აუცილებელი შემადგენლები</u>

1. მართვის ობიექტის გარდამავალი მახასიათებელი.

2. პროგრამის ტექსტი ანდა ოპტიმიზაციის სქემა.

3.მართვის ობიექტისექსპერიმენტალური და მოდელური გარდამავალი ფუნქციების შედარების გრაფიკი.

2.6. ლაბორატორიული სამუშაო №5

პი რეგულატორის ოპტიმალური პარამეტრების განსაზღვრა

5.1. სამუშაოს მიზანი: პი რეგულირების კანონის რეალიზაციის გაცნობა კონტროლერის გამოყენებით. რეგულიატორის კოეფიციენტების ოპტიმიზაციის ხერხების შესწავლა.

5.2. თეორიული შესავალი

კონტროლერის პიდ რეგულიატორი ასრულებს სიგნალების პროპორციულ–ინტეგრალურ–დიფერენციალურ გარდაქმნას და აღიწერება შემდეგი სახის გარდამავალი მახასიათებლით:

$$u(t) = K_{\text{у}} \cdot \varepsilon(0) \cdot \left(1 + \frac{1}{\text{Т}_{\text{И}}} \cdot t + K_{Д} \cdot e^{\frac{-t}{T_{\text{Д}} / K_{\text{Д}}}} \right)$$

სადაც u(t) – რეგულიატორის მმართველი ცვლადია ავტონომიურ რეჟიმში;

ε(0) – შეცდომის სიგნალის ნორმალიზებული სიგნალია;

$K_{\text{у}}$ – რეგულიატორის გაძლიერებაა;

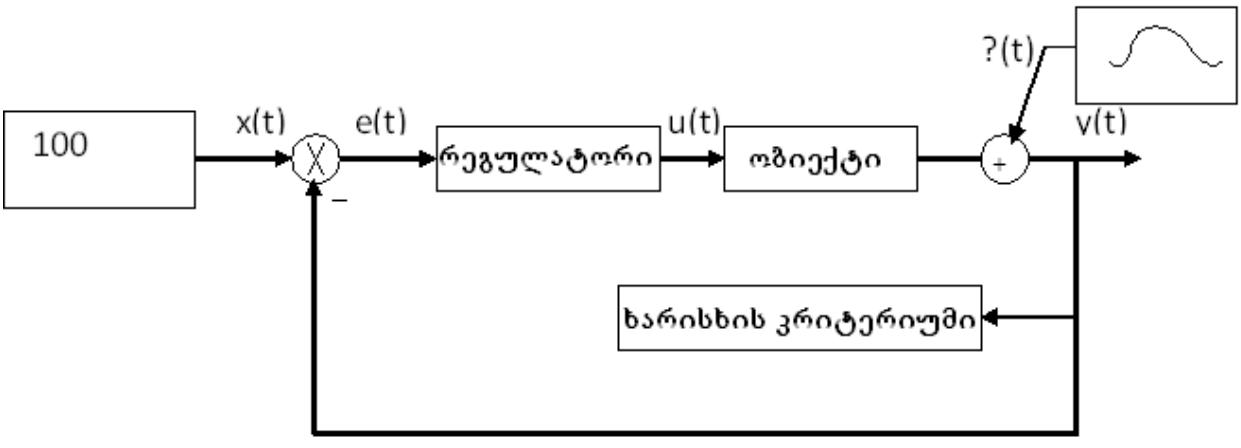
ТИ – გაინტეგრალების დროის;

$T_{\text{д}}$ – გადიფერენცირების დროა;

$K_{\text{д}}$ – გადიფერენცირების კოეფიციენტია.

განიხილება პი რეგულიატორის კოეფიციენტების ოპტიმიზაციის ამოცანა. არსებობს ამ ამოცანის ამოხცნის მრავალი ხერხი. მაგ. შეიძლება Siam, Matlab პაკეტების გამოყენება. აგრეთვე შეიძლება გამოყენებულ იქნას სტრუქტურული მოდელირების მეთოდი ხარისხის ფუნქციონალის უმცირეს კვადრატების მეთოდთან კომპლექსში.

პი რეგულატორის კოეფიციენტების ოპტიმიზაცია ტარდება ნახ. 5.1–ზე მოცემული სქემის მიხედვით.



ნახ. 5.1 პი რეგულატორის პარამეტრების ოპტიმიზაციის სქემა

სადაც x(t) – დავალების სიგნალია;

ε(**t**) – შეცდომის სიგნალი;

u(t) – მართვის სიგნალი;

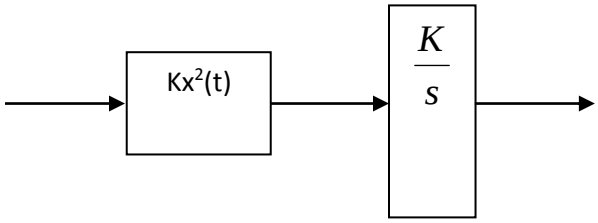
η(t) – გაზომვის ხელშეშლა;

y(t) – ობიექტის გამოსასვლელი;

Siam პაკეტის გამოყენების შემთხვევაში ხარისხის კრიტერიუმი გამოიყურება შემდეგნაირად:

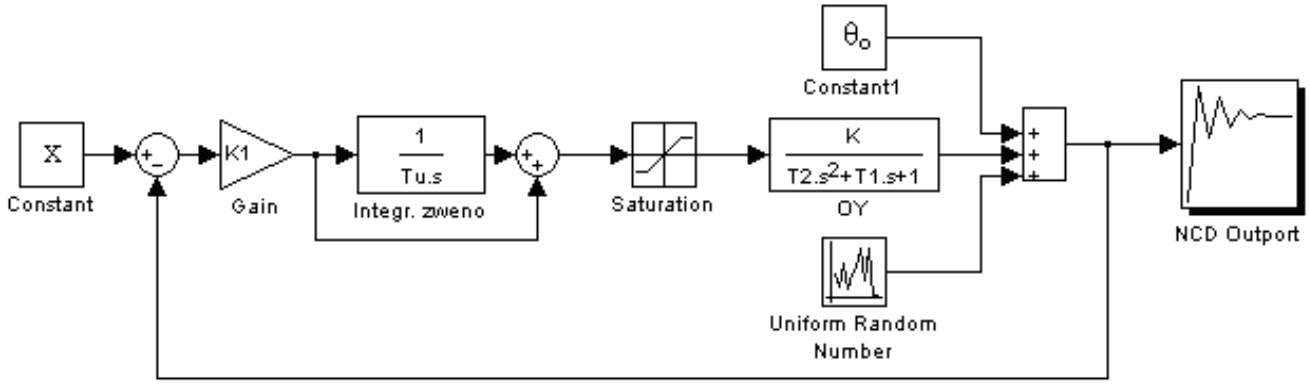
$$F = \int_0^T \varepsilon(t)^2$$

და აქვს სტრუქტურა პაკეტში აწყობისათვის:



Matlabპაკეტის გამოყენების შემთხვევაშიოპტიმიზაციისათვის შეიძლება გამოყენებულ იქნას NCD Outport ბლოკით Simulinkგარემოში (იხ. ნახ. 5.2). აღნიშნულმა ბლოკმა შეიძლება შეასრულოს შემდეგი ოპერაცია:

1. საოპტიმიზაციო სისტემის ნებისმიერ სიგნალზე მოთხოვნილი შეზღუდვის მიცემა;
2. პარამეტრების ჩვენება, რომლებზეც საჭიროა ოპტიმიზაცია;
3. გაურკვეველი პარამეტრების მითითება;
4. სისტემის პარამეტრული ოპტიმიზაციის ჩატარება მოცემული შეზღუდვების გათვალისწინებით.



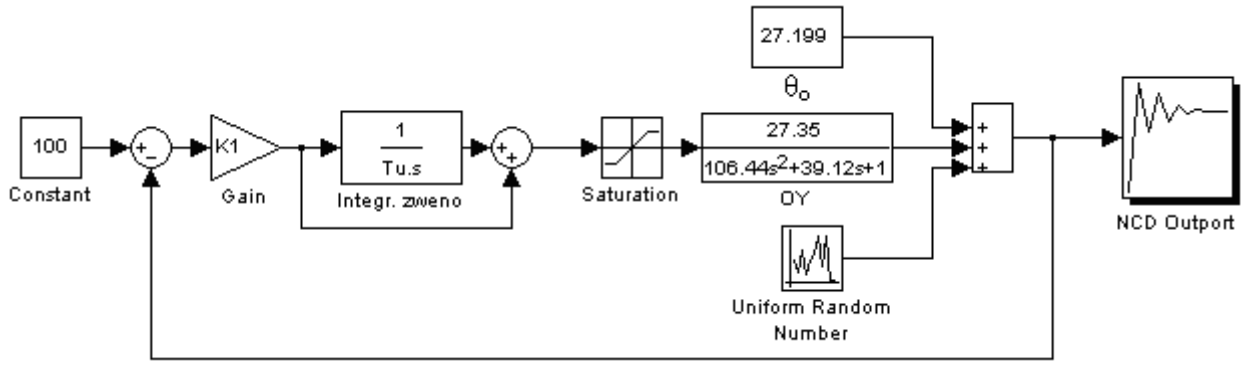
ნახ. 5.1 Matlabპაკერტის გარემოში რეალიზებული ოპტიმიზაციის სქემის მაგალითი

5.3. სამუშაოს შესრულების თანმიმდევრობა

5.3.1. შექმენით სქემა პი კოეფიციენტების ოპტიმიზაციისათვის ერთერთი ზემოთ შემოთავაზებული მეთოდით

გამოიძახეთ Simulink რისთვისაც Matlab –ის ინსტრუმენტების პანელში დააჭირეთ პიქტოგრამას . შექმენით ახალი დოკუმენტი რისთვისაც შეასრულეთ მოქმედებას File/New/Modelააგეთ სხემა, რომელიც ნაჩვენებია ნახ. 5.2–ზე.

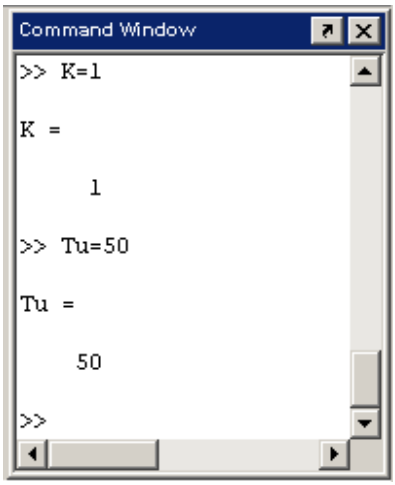
Saturation ბლოკში შეიტანეთ საჭირო მნიშვნელობები Upper limit ველში (მაგალითად, 100)და Lower limitველში მნიშვნელობა (მაგალითად, 0). დაუშვათ რომ ადგილი აქვს ხმაურს განაწილების თანაბარი კანონითდიაპაზონში [–0,1; 0,1], ამიტომ ბლოკში Uniform Random Number ველში Minimum შეიყვანით –0,1, ხოპლო ველში Maximum შეიტანეთ 0.1..



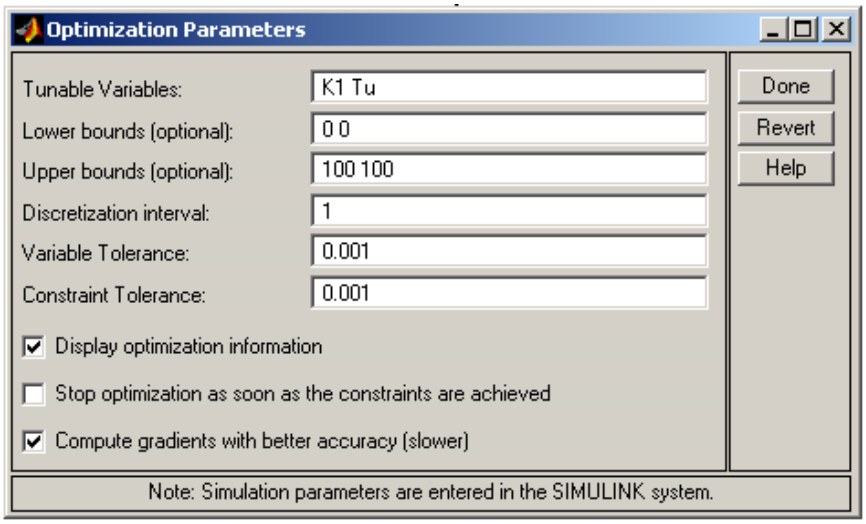
ნახ. 5.3 პი კოეფიციენტების ოპტიმიზაციის სქემა

5.3.2 მოახსინეთ კოეფიციენტების ოპტიმიზაცია

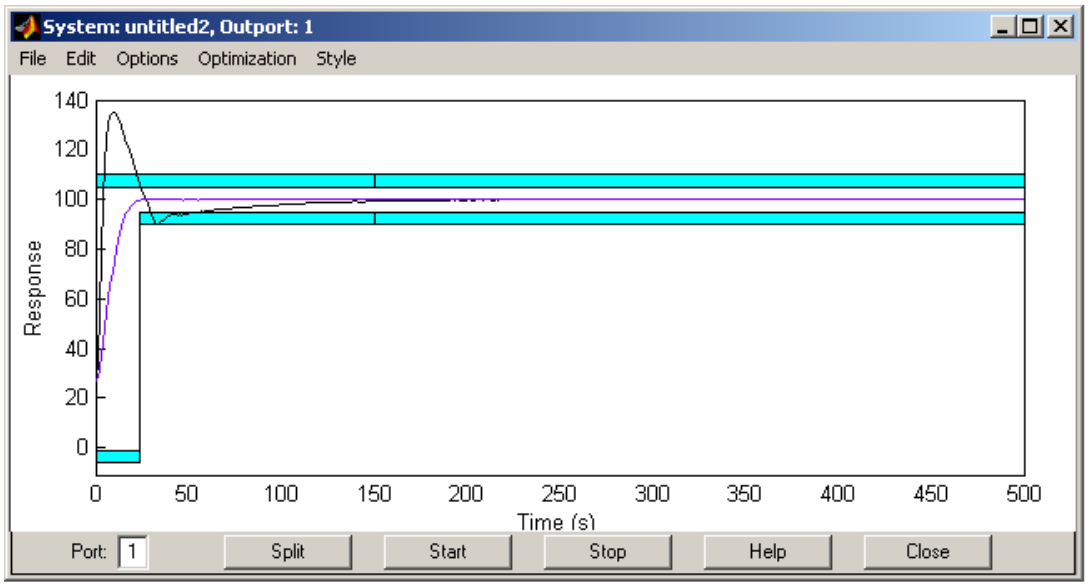
Matlab –ში ფანჯარაში Command Window შეიყვანეთ K1, Tu პარამეტრების მნიშვნელობები (იხ. ნახ. 5.4). ამის შემდეგ გახსენით ბლოკი NCD Outport,მენიუში Optimization/Parameteres –ის არჩევით, წარმოქმნილ ფანჯარაში (იხ. ნახ. 5,5) შეიტანეთ საოპტიმიზაციო კოეფიციენტები, მათი ზედა და ქვედა ოპტიმიზაციის ზღვრები. ინსტრუმენტების პანელზე Startღილაკზე თითის დაჭერით გაუშვით პროექტი (იხ. ნახ. 5.6). ამის შემდეგ Matlab –ში Command Windowფანჯარაში დაათვალიერეთ K1 და Tuპარამეტრების მნიშვნელობები (იხ. ნახ. 5.7).



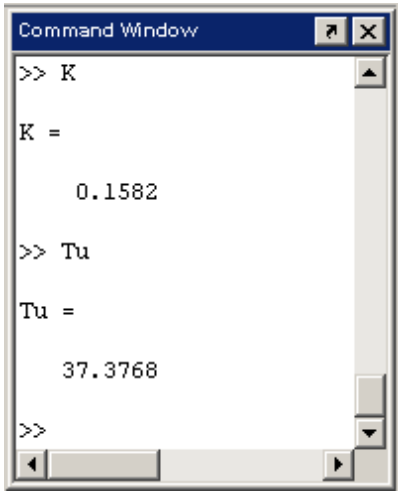
ნახ. 5.4 ***K1, Tu*** პარამეტრების შეყვანა



ნახ. 5.5 ***K1, Tu***ოპტიმიზაციის პარამეტრების განსაზღვრა



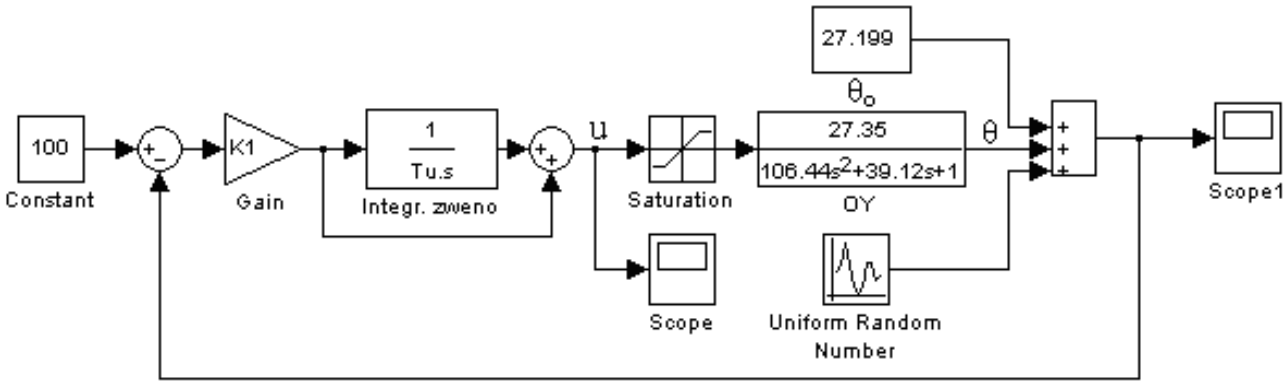
ნახ. 5.6 **NCD Outport** ბლოკის მუშაობის შედეგები



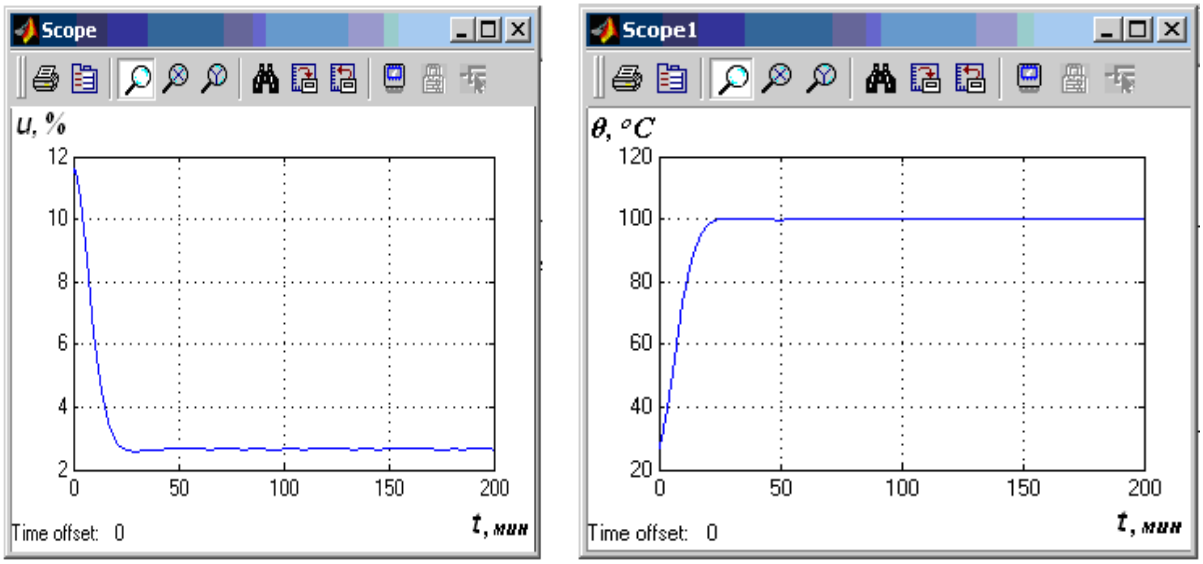
ნახ. 5.5 ***K1, Tu*** ოპტიმიზაციის პარამეტრების დათვალიერება

5.3.3 მართვის ობიექტის გამოსასვლელის გარდამავალი პროცესის გრაფიკის აგება შეყვანილი ოპტიმალური კოეფიციენტების შემთხვევაში

Scope – ის გამოყენებით დაათვალიერეთ გარდამავალი პროცესის გარაფიკები (ნახ. 5.8). გაანალიზეთ მიღებული შედეგები (ნახ. 5.9).



ნახ. 5.8 გარდამავალი პროცესების გრაფიკების პარამეტრების დათვალიერების სქემა



ნახ. 5.9 გარდამავალი პროცესების გრაფიკები

5.3.4 გამოიტანეთ დასკვნები გაწეული მუშაობის შესახებ

5.4 საკონტროლო კითხვები და დავალებები

5.4.1შეაფასეთ მიხებული გარდამავალი პროცესის ხარისხი

5.4.2 რა სახის ოპტიმიზაციის მეთოდი იყო გამოყენებული სამუშაოში.

5.4.3 რა სახის გავლენას ახდენს პიდ რეგულატორის ასაწყობი კოეფიციენტები გარდამავალი პროცესის ხარისხზე?

5.4.4 რისთვის არის შეყვანილი ობიექტის მართვის მოდელის სქემაში введен блок «saturation» ბლოკი (ნახ.2)?

ანგარიშის აუცილებელი შემადგენელნი

1. პი ტეგულატორის პარამეტრების ოპტიმიზაციის სქემა.
2. პი ტეგულატორის პარამეტრების ოპტომალური პარამეტრების მნიშვნელობები.
3. გარდამავალი პროცესებისა და მმართველი ზემოქმედების გრაფიკები.

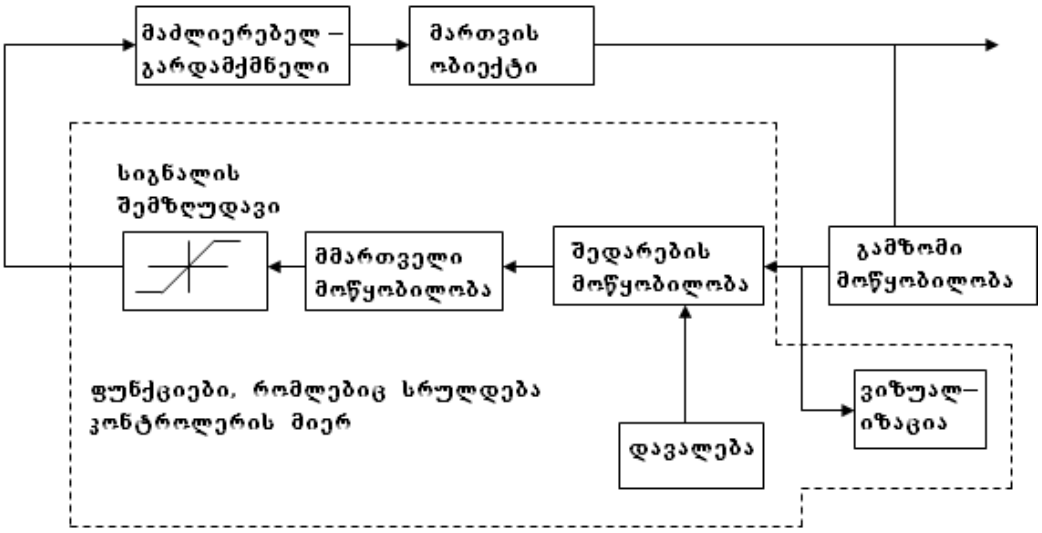
2.7. ლაბორატორიული სამუშაო №6

ტემპერატურული რეგულირების სისტემის აგება

6. 1. სამუშაოს მიზანი: მიკროკონტროლერის გამოყენებით ტიპიური მართვის კონტურის აგების მეთოდიკის შეთვისება .

6.2. თეორიული შესავალი

ტექნოლოგიური პროცესების მართვის ავტომატიზირებულ სისტემებში მართვის იერარქიის დაბალ დონეებზე იმყოფება ტექნოლოგიური აგრეგატებისა და ცალკეული პარამეტრების ავტომატური მართვისა და რეგულირების ქვესისტემები. ავტომატური მართვისა და რეგულირების ტიპიური კონტურის ფუნქციონალური სტრუქტურა ნაჩვენებია ნახ. 6.1 – ზე.



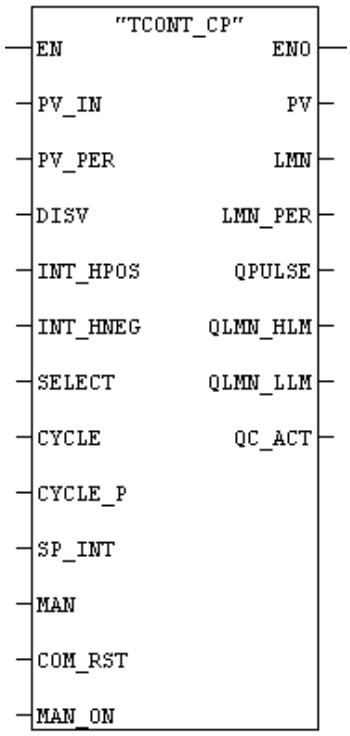
ნახ. 6.1 ტიპიური მართვის კონტურის ფუნქვიონალური სქემა

მოცემულ ლაბორატოტიულ სამუშაოში განიხილება ტემპერატურის მართვის კონტური ელექტრულ გამაცხელებელ ღუმელში. მართვის ობიექტია ღუმელი, რომლის გამოსავალი სიდიდე – ტემპერატურა – გაიზომება თერმოწყვილის ან თერმოწინააღმდეგობის დახმარებით.

ტემპერატურის სტაბილიზაციის მართვის კონტურის აგების დროს პროპორციულ/ინტეგრალურ/დიფერენციალური (პიდ), პროპორციულ/ინტეგრალური (პი) რეგულიატორების ბაზაზე, STEP 7 – ში, რეკომენდირებულია (TCONT_CP) ბლოკის გამოყენება, რომელიც უზრუნველყოფს მითითებული რეგულატორების ფუნქციების რეალიზებას.

რეგულატორების ბლოკის გამოძახება

ნახ. 6.2 – ზე გამოსახული სქემა გვიჩვენებს მართვის გამოძახებას FBD – ში. ხოლო, ცხრილ 1 – ში მოყვანილია (TCONT_CP) – ს პიდ რეგულირების ბლოკის ზოგიერთი პარამეტრის აღწერა.



ნახ. 6.2 (TCONT_CP) – ის სქემა

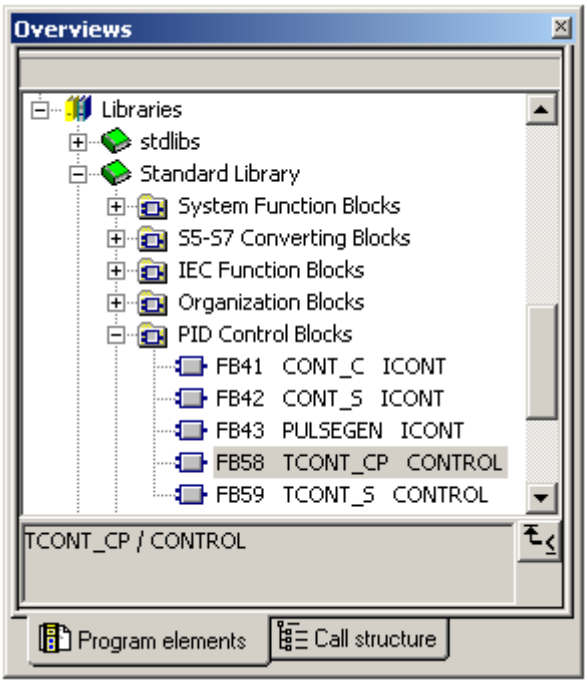
ცხრილი 1. სამუშაოში გამოყენებული (TCONT_CP) – ის ბლოკის პარამეტრების აღწერა

პარამეტრი	დანიშნულება	ტიპი	აღწერა
PV_IN	INPUT (შესასვლელი)	REAL	პროცესის ცვლადის შესასვლელი (“Process Variable In”). საწყისი მნიშვნელობა შეიძლება დაყენებულ იქნას ნაჩვენებ შესასვლელზე ანდა მასზევე შეიძლება მიერთებულ იქნას პროცესის გარე ცვლადი მცოცავ მძიმიანი რიცხვის ფორმატში.
SP_INT	INPUT (შესასვლელი)/ OUTPUT (გამოსასვლელი)	REAL	დაყენების სიგნალის შიგა მნიშვნელობა (“Internal Setpoint”). შესასვლელი “Internal Setpoint” გამოიყენება დაყენების სიგნალის დონის მიცემისათვის.
MAN	INPUT (შესასვლელი)/ OUTPUT (გამოსასვლელი)	REAL	მმართველი ცვლადი, რომელიც ხელით შეიყვანება (“Manual Value”). ავტომატური რეჟიმის დროს კოპრექტირდება მმართველი ცვლადის მნიშვნელობამდე.

MAN_ON	INPUT (შესასვლელი)/ OUTPUT (გამოსასვლელი)	BOOL	ხელის რეჟიმში გადართვის გადამრთველი ("Manual Operation On"). თუ პარამეტრი "Manual Operation On" დაყენებულია, მაშინ ხელით შეყვანილი მმართველი ცვლადი (MAN), დგინდება როგორც მმართველი ცვლადის მნიშვნელობა. 0 – ავტომატური რეჟიმია; 1 – ხელის რეჟიმი.
LMN	OUTPUT (გამოსასვლელი)	REAL	მმართველი ცვლადი ("Manipulated Variable"). მმართველი ცვლადის მოქმედი მნიშვნელობა მცოცავმძიმიან რიცხვის ფორმატში მიეწოდება ერთსახელა გამოსასვლელზე: Manipulated Variable.
QPULSE	OUTPUT (გამოსასვლელი)	BOOL	გამოსასვლელი იმპულსური სიგნალი ("Output Pulse"). გიმ-მოდულირებული იმპულსური წარმოდგენა მმართველი ცვლადისა გამოსასვლელზე: Output Pulse.

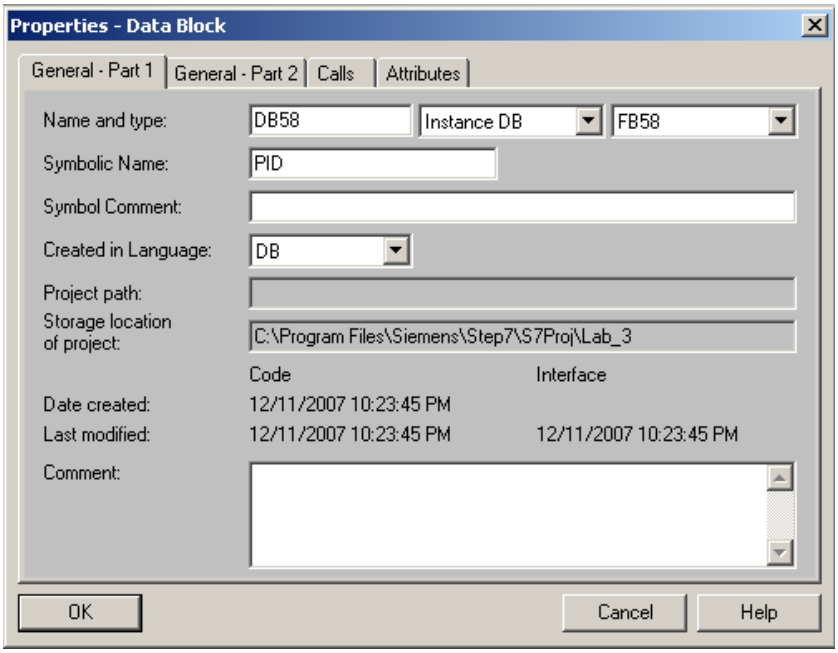
ბლოკის კონფიგურირების თანმიმდევრობა რეგულირების პი – კანონით რეალიზაციისას:

1) შექმენით საორგანიზაციო ბლოკი OB35; გახსენით იგი და Overviews ცნობარიდან აიღეთ კომპონენტი FB58 TCONT_CP CONTROL შტოთი Libraries / Standard Library / PID Control Blocks (იხ. ნახ. 6.3).



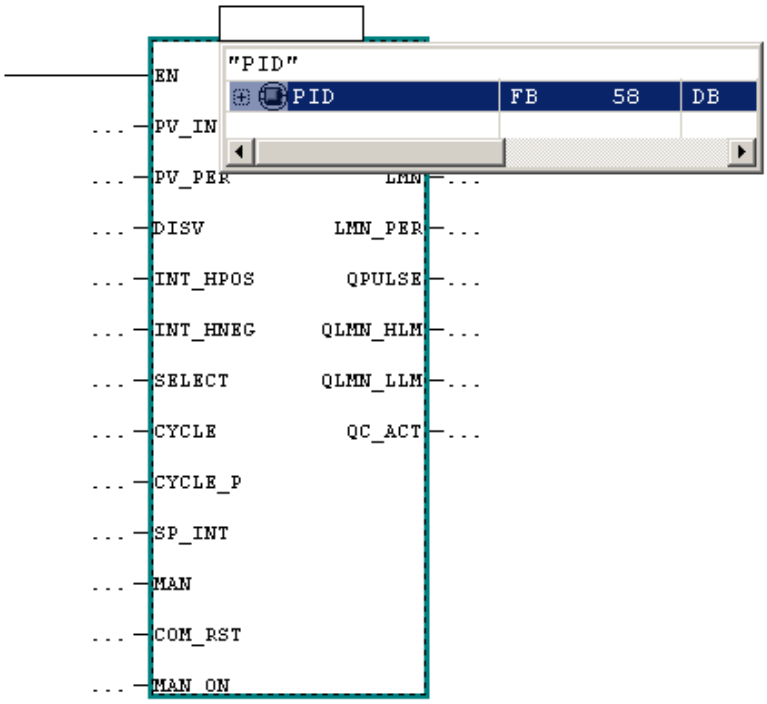
ნახ. 6.3 რეგულირების პი კანონის მარეალიზებელი ბლოკის არჩევა Overviews ცნობარიდან

2) კონტროლერში აუცილებელია პი რეგულატორისთვის მეხსიერების ადგილის გამოყოფა. ამისათვის საჭიროა მონაცემთა ბლოკის DB58 – ის შექმნა.



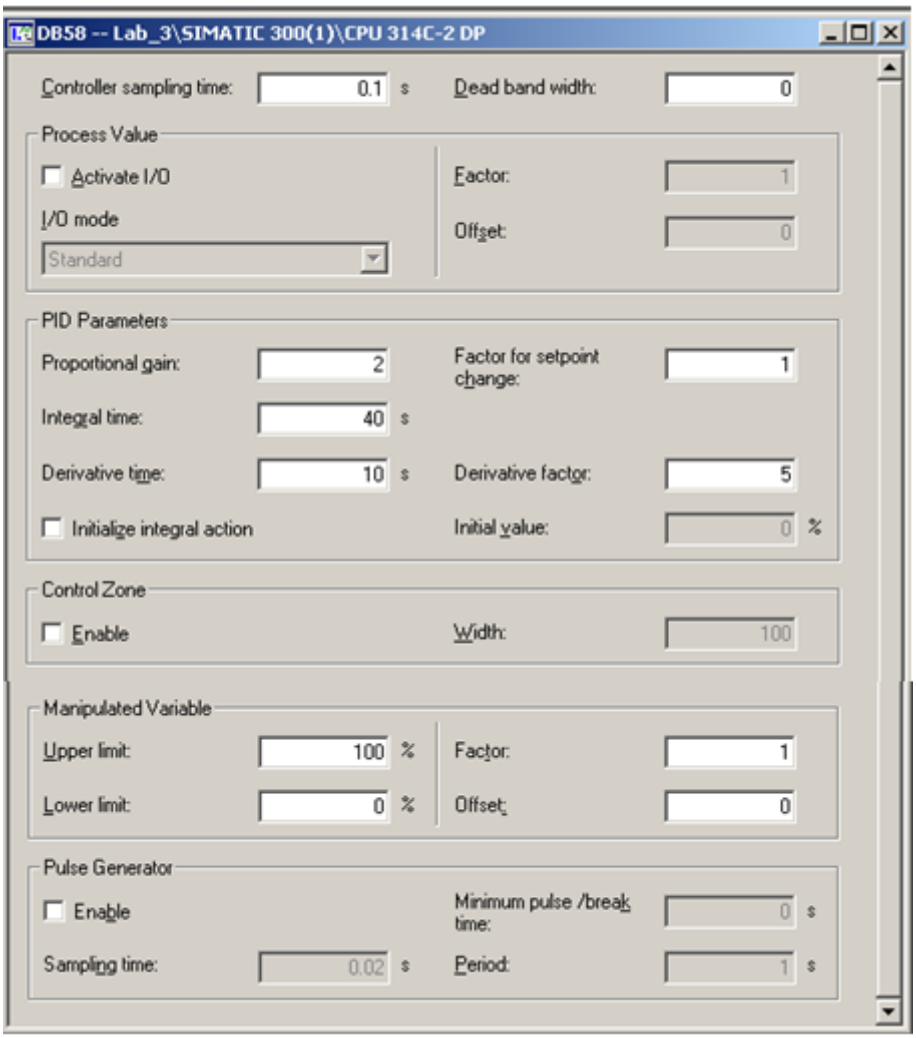
ნახ. 6.4 DB58 ბლოკის პარამეტრების ფანჯარა

3) მას შემდეგ, როდესაც შექმენით ბლოკი DB58, ბლოკ OB35 – ში TCONT_CP სახელწოდებაში აირ5ჩიეთ ”PID” (იხ. ნახ. 6.5).



ნახ. 6.5 პი – რეგულირების ბლოკის (TCONT_CP) მიკავშირება DB58 ბლოკთან

4) გახსენით DB58 ბლოკი და შეიყვანეთ პარამეტრების მოთხოვნილი მნიშვნელობები (ნახ. 6.6).



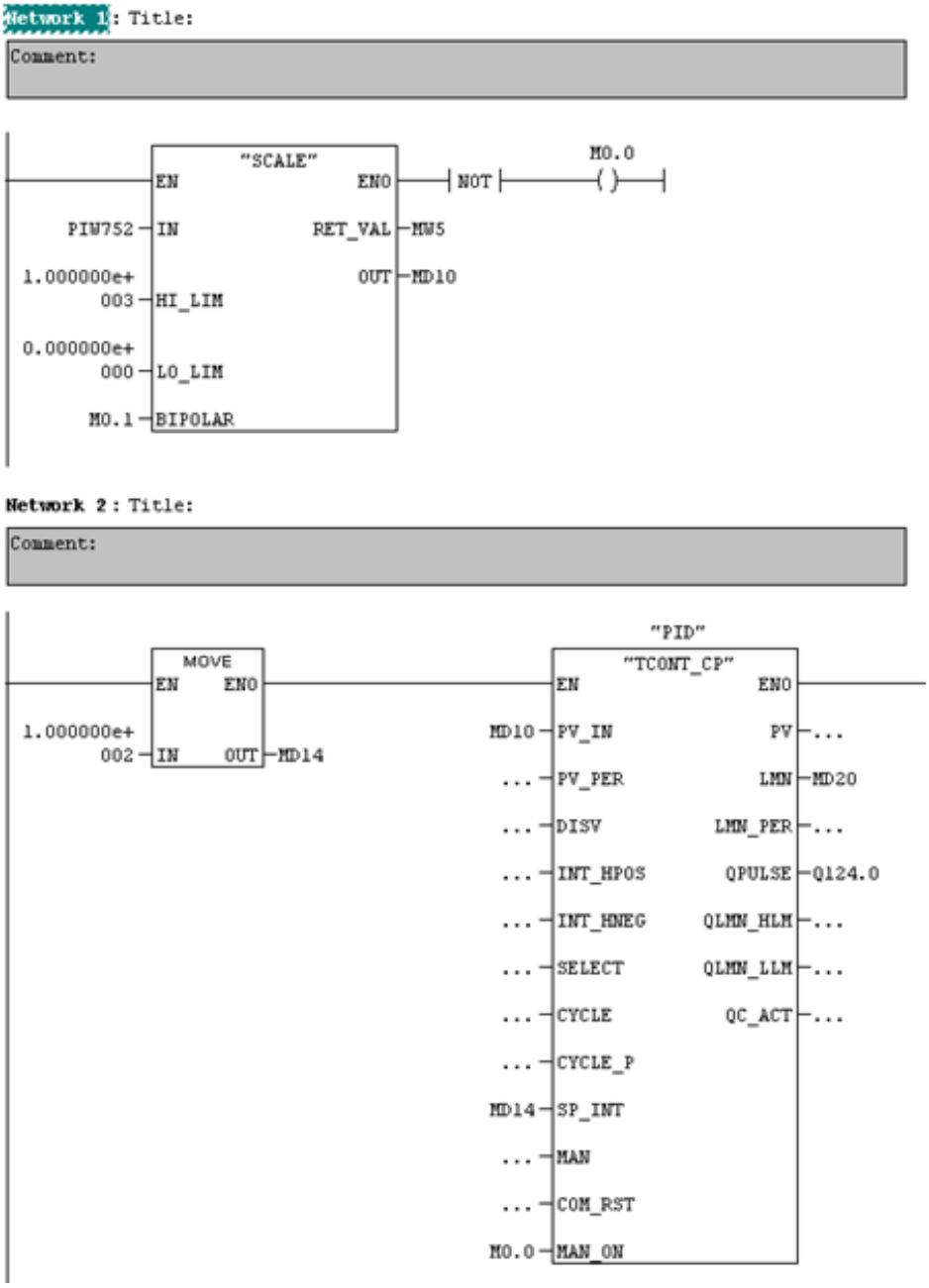
ნახ. 6.6 DB58 მონაცემთა ბლოკში პარამეტრის მნიშვნელობების შეყვანის ფანჯარა

6.3 სამუშაოს შესრულების თანმიმდევრობა

6.3.1 შექმენით ახალი პროექტი და დააკონფიგურირეთ კონტროლერი STEP 7 - ში.

6.3.2 პროექტის ფანჯრის მარცხენა ნაწილში გახსენით ხე Blocks - ამდე. შექმენით საორგანიზაციო ბლოკი OB35. გამოჩენილ ფანჯარაში Properties აირჩიეთ ენა, რომელზედაც დაწერთ პროგრამას: LAD (საკონტაქტო–სარელეო სქემები).

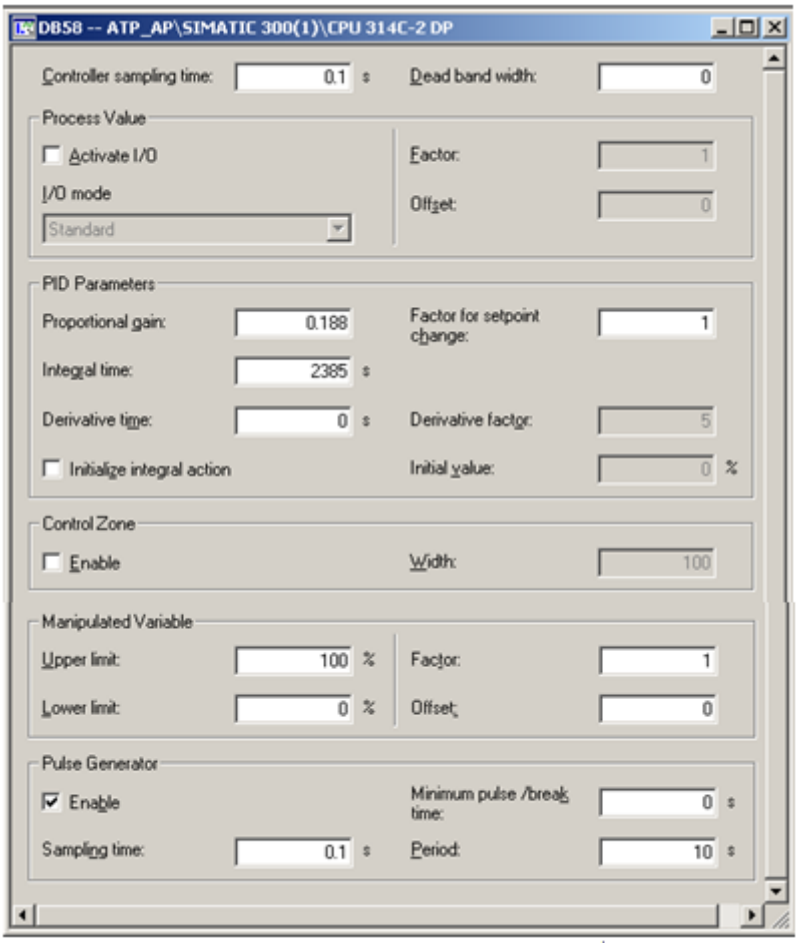
6.3.3 აღმოცენებულ ფანჯარაში დაწერეთ დასმული ამოცანის შესრულების პროგრამა (ტემპერეტურის რეგულირების სისტემის რეალიზაცია) (იხ. ნახ. 6.7):



ნახ. 6.7 დასმული ამოცანის შესრულების პროგრამა

6.3.4 შეინახეთ ცვლილებები OB35 – ში მთავარ მენიუში File/Save – ს არჩევით.

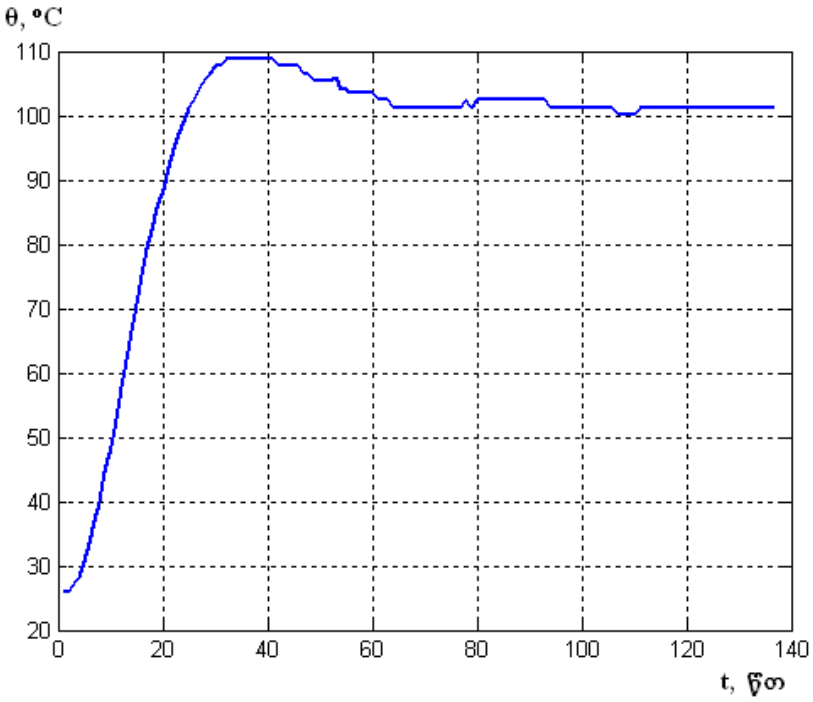
6.3.5 გახსენით მონაცემთა ბლოკი DB58, სადაც შეიყვანეთ პროპორციულობის კოეფიციენტის $K_З = 0.188$ და ინტეგრირების დროის $T_И = 2385$ (წამებში გადაყვანით), ასევე დააყენეთ Period – ი ტოლი *10* – ს, ანუ იმ მნიშვნელობის რაც მიღებულ იქნა ლაბორატორიულ სამუშაოში №3. პარამეტრი Sampling timeდააყენეთ ტოლი 0,1 წამის. დააყენეთ ალამი Pulse Generator(იხ. ნახ.6.8).



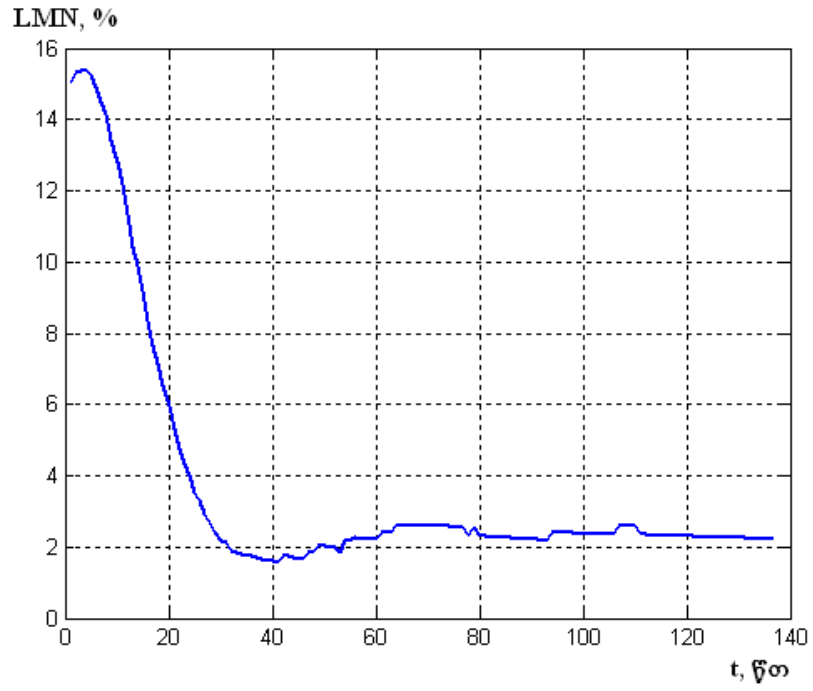
ნახ. 6.8 $K_з$–სა $T_о$–ს მნიშვნელობების შეყვანა DB58 ბლოკში

6.3.6 შეასრულეთ ტემპერატურისა და მართვის სიგნალებისა ვიზუალური ასახვა და დაარქივება (LMN) ProToo – ის საშუალებებით.

6.3. 7 ააგეთ გარდამავალი პროცესის მრუდის გრაფიკი (ნახ. 6.9). ეს შეიძლება გაკეთდეს გამოყენებითი პროგრამების პაკეტების Microsoft Excel ან Matlab –ის გამოყენებით.



ა)



ბ)

ნახ. 6.9. გარდამავალი პროცესების გრაფიკი, აგებულნი
ა) ტეგი VAR_1; ბ) ტეგი VAR_2 მნიშვნელობებით

6.3.8 გამოიტანეთ დასკვნები სამუშაოს შესრულების შესახებ.

6.4 საკონტროლო შეკითხვები და დავალებები

6.4.1 რა არის გადარეგულირება და როგორ გამოითვლება იგი?

6.4.2 განმართეთ ალგორითმების როლი სქემაში, რომელიც მოყვანილია ნახ. 5.7 – ზე.

6.4.3 განმარტეთ პი რეგულატორის არხების ფუნქციონირების პრინციპი.

6.4.5 რატომ არის რეკომენდირებული, რომ STEP 7 – ში რეგულირების პროგრამა დაიწეროს OB35 – ში და არა OB1 – ში?

ანგარიშის აუცილებელი შემადგენელი ნაწილი

1. პროგრამა დაწერილი STEP 7 – ში.
2. გარდამავალი პროცესის გრაფიკი.
3. გარდამავალი პროცესის ხარისხის მაჩვენებლების მნიშვნელობები.

თავი 3. პროგრამირებადი ლოგიკური კონტროლერის მუშაობის გამოკვლევა (SIMATIC S7-313C-2DP -ის ბაზაზე)

ამ თავში განხილულია SIMATIC S7-313C-2DP პროგრამირებადი ლოგიკური კონტროლერის თავისებურებები, შესასვლელი და გამოსასვლელი სიგნალების მიერთებები კონტროლერთან, აგრეთვე Step 7 პროგრამულ პაკეტთან მუშაობის პრინციპები და LAD ენაზე დაპროგრამირების ხერხები.

გათვალისწინებულია მართვისა და ავტომატიზაციის სტუდენტებისათვის, რომლებიც სწავლობენ „სამრეწველო მიკროკონტროლერები და პროგრამირებადი ლოგიკური კონტროლერების“ კურსს.

3.1. შესავალი

სამუშაოს მიზანია:

1. SIMATIC S7-313C-2DP პროგრამირებადი ლოგიკური კონტროლერის (პლკ) მოწყობილობისა და მისი მიერთების წესის შესწავლა;
2. პროგრამული პაკეტის Step-7-ის მეშვეობით პლკ-ს დაპროგრამირების საფუძვლების შესწავლა „კონტაქტური გეგმის“ ენის გამოყენებით.

დღეისათვის უკვე შესაძლებელია საკმაოდ რთული პროცესების ავტომატიზაციის რეალიზაციები, რაც განპირობებულია მიკროპროცესორული ტექნიკისა და პერსონალური კომპიუტერების განვითარების მაღალი დონით. ფირმა SIEMENS წარმოგვიდგენს სამრეწველო საწარმოების ავტომატიზაციისათვის მოწყობილობების ფართო ასორტიმენტს, დაწყებული მარტივი ლოგიკური რელედან LOGO და დამთავრებული თანამედროვე მოდულური სამრეწველო კომპიუტერებით რეალური დროის ოპერაციული სისტემებით. უფართოესი გავრცელება ჰპოვეს S7-300 სერიის პროგრამირებადმა ლოგიკურმა კონტროლერებმა. ისინი წარმოადგენს სწრაფ, მაღალმწარმოებლურ და მრავალფეროვან კომპაქტურ პლკ-ებს. ჩვენს სტენდზე დაყენებულია S7-313C-2DP ტიპის პლკ, რომლის დანიშნულებაა შედარებით მარტივი ტიპის ავტომატური მართვის ამოცანების გადაჭრა, სადაც აუცილებელია ინფორმაციის სწრაფი გადამუშავება და სისტემის რეაქციის მცერე დრო. კონტროლერს აქვს მოდულური სტრუქტურა სხვადასხვა ფუნქციონალიური მოდულების მასთან შეერთების გზით გაფართოების შესაძლებლობით.

S7-313C-2DP პლკ-ს ძირითადი განმასხვავებელი ნიშნებია:

- მიკროპროცესორი: 100-200 ნწ ბიტებზე ლოგიკური ოპერეციების შესრულებისათვის;
- ჩატვირთვის მეხსიერება NVFlash-EEPROM მეხსიერების მიკრობარათის სახით, ტევადობით 8 მბაიტამდე: პროგრამისა და მონაცემთა შენახვა, Step 7 პროექტის ოპციონალური შენახვა სიმბოლური ცხრილებითა და კომენტარებით, მონაცემთა არქივებისა და მონაცემების შენახვა;

- მონაცემთა სარეზერო ასლის არამომსახურებადი შენახვა: კვების წყაროს შეფერხებებისას მეხსიერების მიკრობარათში ჩაიწერება ალმების ტაიმერების მთვლელების მდგომარეობა და მონაცემთა ბლოკების შედგენილობა;
- ჩაშენებული MPI ინტერფეისი: დაპროგრამირება/ დიაგნოსტიკა/ მომსახურება/ უმარტივესი ქსელური სტრუქტურების აგება, მონაცემთა გადაცემის სიჩქარე 187.5 კბიტ/წმ. 16-მდე SIMATIC S7/C7 ცენტრალური პროცესორის გაერთიანება, მონაცემთა გლობალური გადაცემის მექანიზმის მხარდაჭერა;
- მუშაობის რეჟიმების (RUN/STOP/MRES) ჩაშენებული გადამრთველი;
- პაროლით დაცვა: უზრუნველყოფს პროგრამის დაცვას არასანქცირებული ხელწვდომისაგან;
- დიაგნოსტიკური ბუფერი: ბუფერში შეინახება 100 ბოლო შეტყობინება მტყუნებებზე და წყვეტებზე. ბუფერის შემცველობა შეიძლება გამოყენებულ იქნას ცენტრალური პროცესორის გაჩერების მიზეის ანალიზისთვის;
- რეალური დროის საათი: ყველა დიაგნოსტიკურ შეტყობინებაზე შეიძლება მიერთებულ იქნას თარიღისა და დროის აღნიშვნები;
- ჩაშენებული საკომუნიკაციო ფუნქციები: PG/OP კავშირების ფუნქციები, როგორც სტანდარტული ისე გაფართოებული (კლიენტი და სერვერი) S7 - კავშირის ფუნქციები;
- 16 ჩაშენებული დისკრეტული შესასვლელი =24ვ. ყველა შესასვლელი შეიძლება გამოყენებულ იქნას აპარატურული წყვეტების სიგნალების მიღებისათვის;
- 16 ჩაშენებული დისკრეტული გამოსასვლელი =24ვ./0.5ა.
- მოქნილი გაფართოება: ლოკალური შეყვანა-გამოყვანის სისტემა, რომელიც ემსახურება S7 300-ის 31-მდე სასიგნალო, ფუნქციონალურ და საკომუნიკაციო მოდულებს (4 მწკრივიანი კონფიგურაციის შემთხვევაში);
- პიდ-რეგულატორების აგების შესაძლებლობა იმპულსური ანდა ანალოგური გამომავალი სიგნალებისთვის;
- მუშა მეხსიერება: RAM ტევადობით 64 კბაიტი/ 20 K -ინსტრუქციისთვის;
- 8-მდე ლოგიკური შეერთება სამრეწველო ქსელებში, რომლებიც შედგება ლოგიკური კონტროლერებისაგან S7-300/S7-400/C7/პროგრამატორებისაგან/ კომპიუტერებისაგან/ოპერატორის პანელებისაგან. ერთი სტატიკური შეერთება დარეზერვირებულია პროგრამატორთან და ოპერატორის პანელთან კავშირისათვის;
- 12 დისკრეტული შესასვლელი 16-დან შეიძლება გამოყენებულ იქნას ჩაშენებული ტექნოლოგიური ფუნქციების მიერ;
- 3 დისკრეტული გამოსასვლელი 16-დან შეიძლება გამოყენებულ იქნას სიხშირულ ანდა განივ-იმპულსური მოდულიაციის რეჟიმში (2,5 კჰც-ამდე);

- სიხშირეების გამზომები: სამი არხი 30 კჰც-მდე სიხშირეების გაზომვისათვის. მოქმედების პრინციპი: იმპულსების რაოდენობის თვლა საყრდენი დროის მონაკვეთისათვის;
- ჩქარი თვლა: სამი მთვლელი (30 კჰც) ჩაშენებული კომპარატორებით, რომელთა შესასვლელები შეიძლება შეუერთდეს 24ვ ინკრიმენტალურ გადამწოდს;
- ჩაშენებული ინტერფეისი PROFIBUS-DP. წამყვანი ან მიმდევნი DP მოწყობილობების ინტერფეისი განაწილებული შეყვანა-გამოყვანის სისტემებში მუშაობისათვის. მომხმარებლის თვალთახედვით ლოკალური და განაწილებული შეყვანა-გამოყვანის სისტემები სრულიად იდენტურია, მათთვის გამოიყენება კონფიგურირების, დამისამართებისა და დაპროგრამირების ერთიდაიგივე მეთოდები.

3.2. უსაფრთხოების ტექნიკა ლაბორატორიული სამუშაოების შესრულების დროს.

სამუშაოები სრულდება ყოველთვის პედაგოგთან (ლაბორანტთან ერთად).

სამუშაოებზე დაიშვებიან ის სტუდენტები, რომლებსაც გავლილი აქვთ უსაფრთხოების ტექნიკის ინსტრუქტაჟი და გაცნობილნი არიან მეთოდურ სახელმძღვანელოს სტენდზე „SIMATIC S7-300 C".

ლაბორატორიული სამუშაოების შესრულება უნდა მოხდეს მეთოდური სამუშაოების მითითებების მკაცრი დაცვით. სამუშაოს პროცესში შემაერთებელი გამტარები და მომჭერები შეიძლება იმყოფებოდეს სიცოცხლისათვის სახიათო ძაბვის ქვეშ! გარე მოწყობილობების მიერთება სტენდის პანელთან წარმოებს მხოლოდ პედაგოგის (ლაბორანტის მიერ).

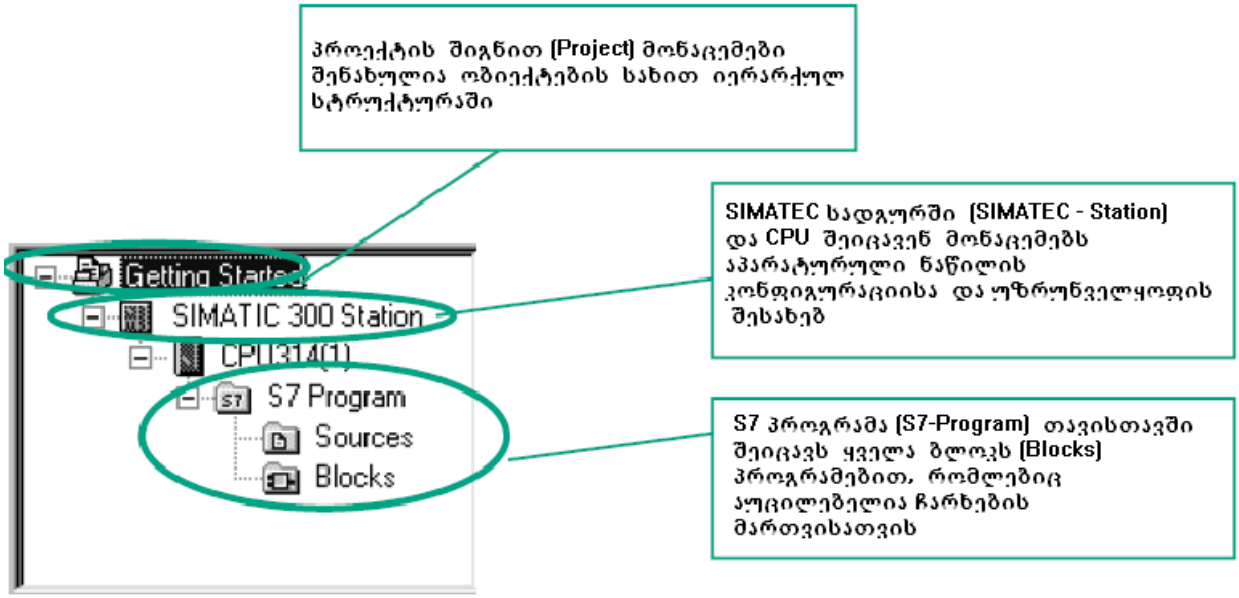
აკრძალულია მეხსიერების გარათის ამოღება კონტროლერიდან.

უწესრიგოათა აღმოჩენის დროს სამუშაოს მიმდინარეობა უნდა შეწყდეს და ეს ამბავი უნდა მოხსენდეს პედაგოგს.

3.3. პროგრამირებადი ლოგიკური კონტროლერის დაპროგრამირება

მეთოდური მითითებების ეს განყოფილება დაწერილია „STEP-7. პირველი ნაბიჯები" (Step 7 v.5.3. First steps. 3.н. 6ES7810-4CA07-8BW0) საფუძველზე.

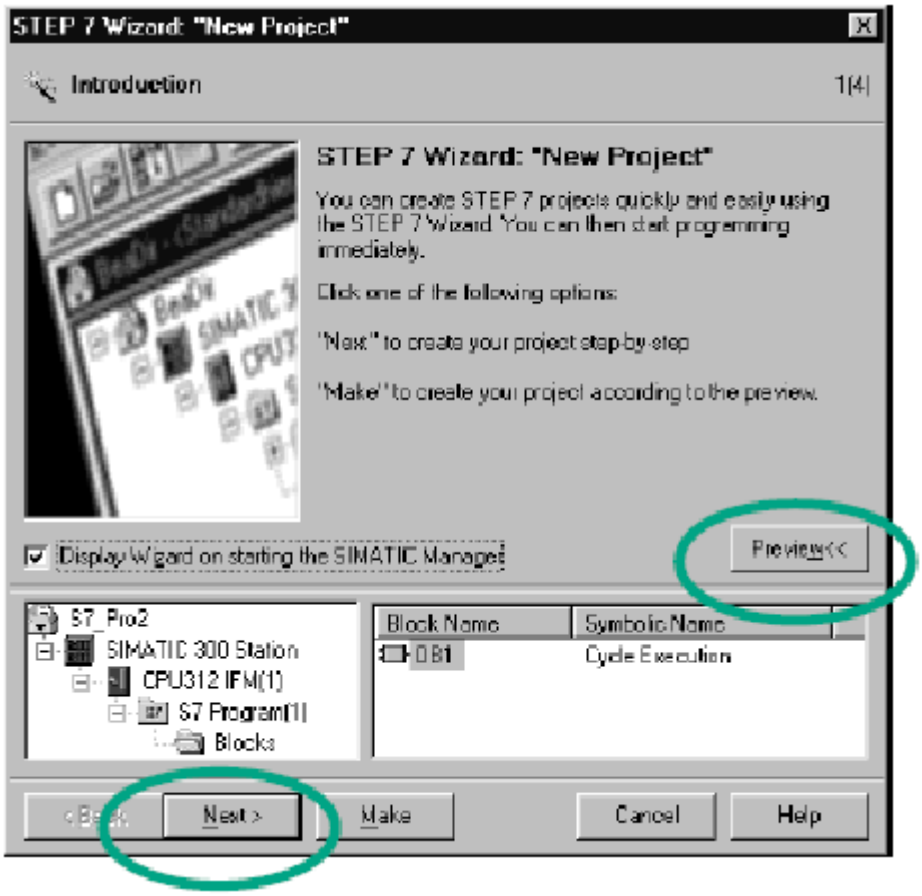
პლკ-ს დაპროგრამირება ხდება STEP-7 პროგრამულ პაკეტში. კომპიუტერზე დაყენებულია პაკეტი Step 7 v.5.2 Professional. აღნიშნული პაკეტი შესაძლებლობას იძლევა SIMATIC S7-300/400 სადგურებისასთვის დაიწეროს პროგრამები: კონტაქტური გეგმის, ფუნქციონალური გეგმის ან ოპერატორების სიის პროგრამირების ენებზე.



ნახ. 4 პროგრამის გაშვების შემდეგ გამოჩენილი ფანჯარა

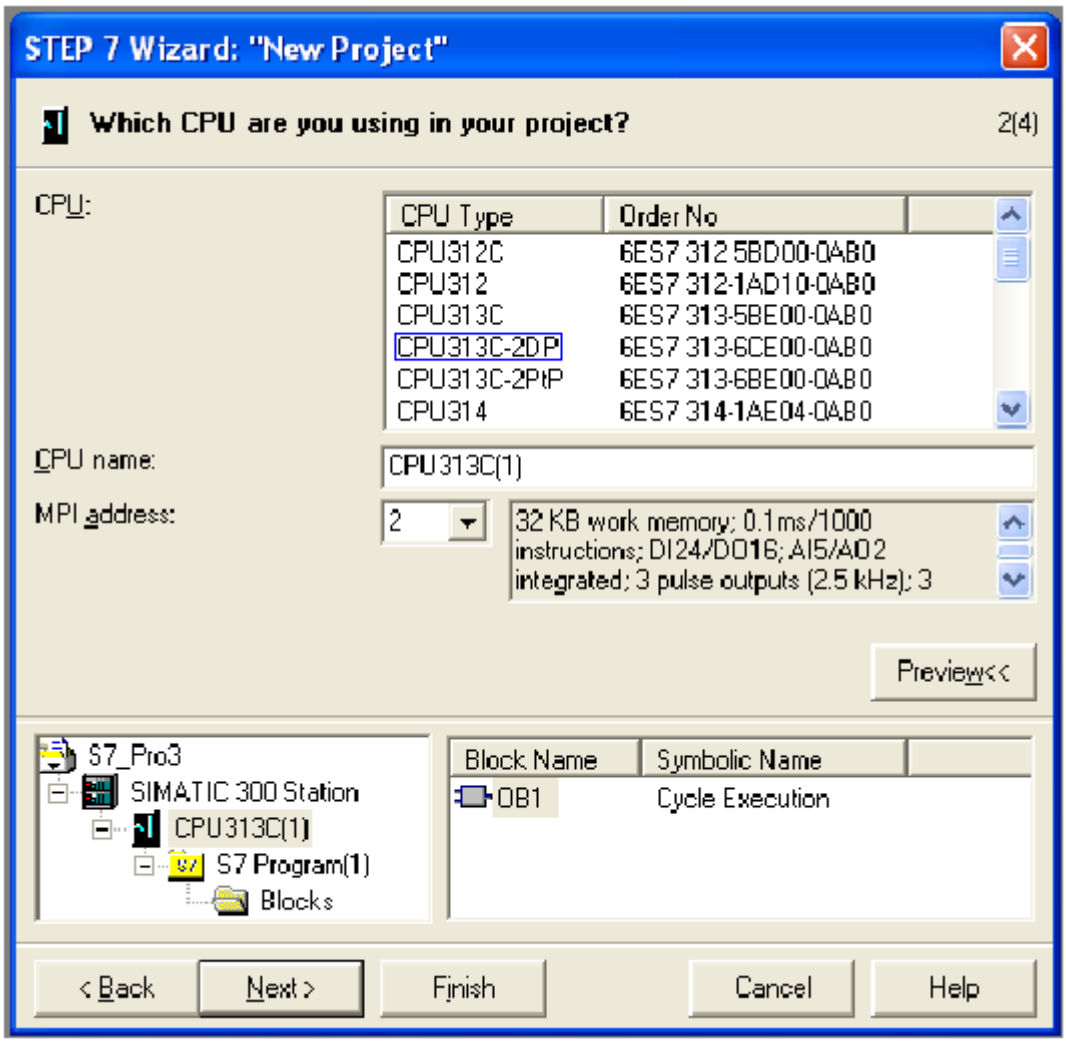
ძირითადი ფანჯრის გააქტიურებისათვის აუცილებელია „SIMATIC Manager» პროგრამის გაშვება. **SIMATIC Manager [SIMATIC ადმინისტრატორი]** - ეს ცენტრალური ფანჯარაა, რომელიც გააქტიურებელი ხდება მას შემდეგ, რაც გაიშვება STEP-7. სიჩუმით გაიშვება ოსტატი Step 7 Wizard, რომელიც დახმარებას გვიწევს პროექტის შექმნის საჭიროებისას. პროექტის სტრუქტურა გამოიყენება ყველა მონაცემისა და პროგრამების სათანადო განთავსებისა და შენახვისათვის.

შემდგომ დიალოგიურ ფანჯარაში გადასვლისათვის საჭიროა ღილაკზე **NEXT [შემდეგ]** თითის დაჭერა.

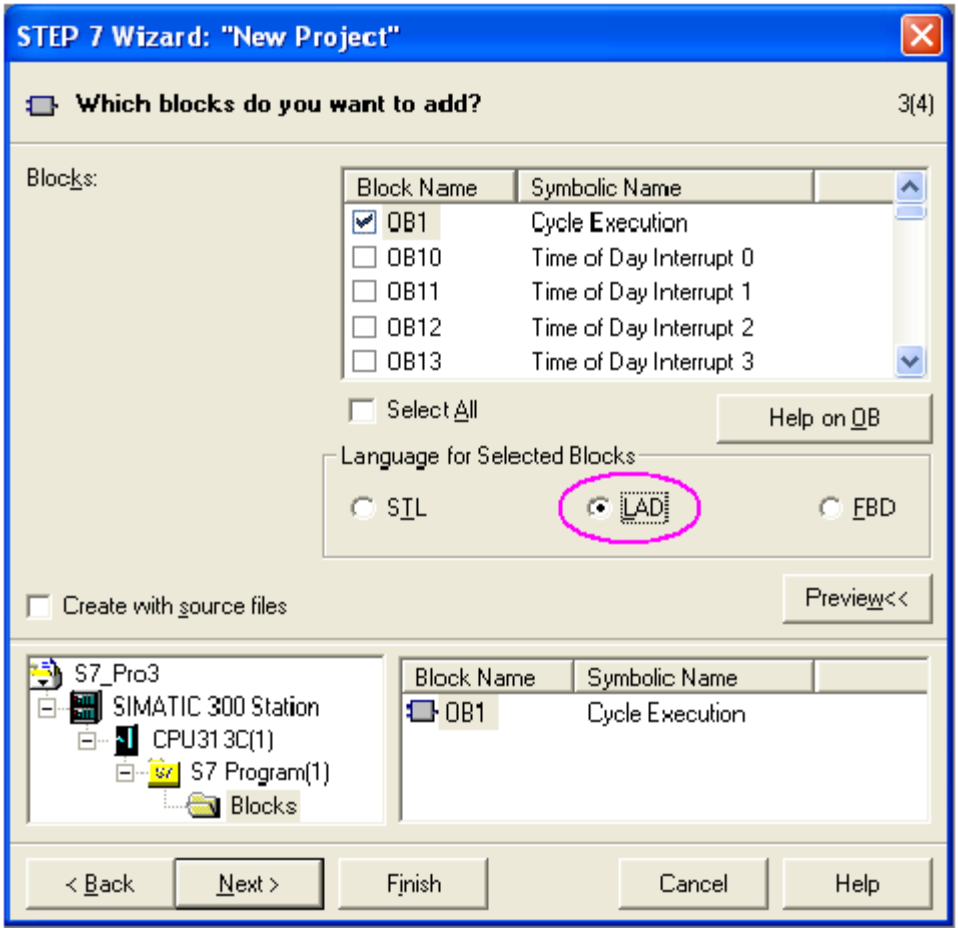


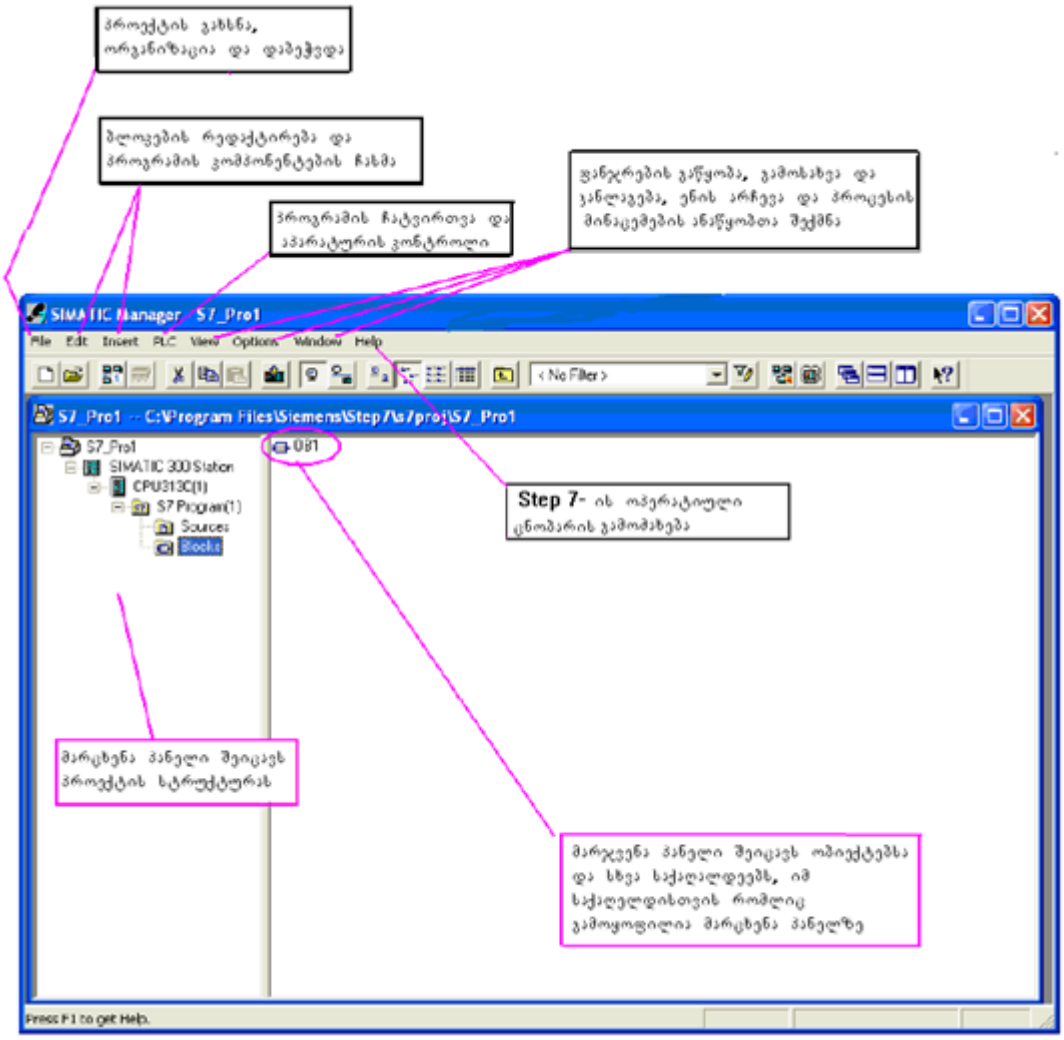
ნახ. 4 *Step 7 Wizard*

შემდეგ ეტაპზე აუცილებელია იმ პროცესორის (CPU) არჩევა, რომლისთვისაც იქმნება პროგრამა. აირჩიეთ იგი და დააჭირეთ **Next.**



ახლა აუცილებელია ენის არჩევა, რომელზედაც შეიქმნება პროგრამა, ჩვენ ვირჩევთ **LAD** ენას (სიტყვისაგან „ladder" - კიბე - კონტაქტური გეგმის ენა). ამ პროგრამულ უზრუნველყოფაში შეიძლება დაპროგრამირების განხორციელება აგრეთვე ორ სხვა ენაზეც, ესენია **STL**-გაფართოებული ასემბლერი და **FBD**-ფუნქციონალური გეგმა. ამ ენებს შორის გადართვა შეიძლება ნებისმიერ დროს, რის დროსაც პროგრამა გადაკონვერტირდება სხვა ენაში უმეტეს შემთხვევაში ავტომატურად. აირჩევთ რა ენას ისევე დააჭირეთ ღილაკს **Next.**


STEP 7 Wizard: "New Project"
Which blocks do you want to add?
3(4)
Blocks:
Block Name
Symbolic Name
OB1
Cycle Execution
OB10
Time of Day Interrupt 0
OB11
Time of Day Interrupt 1
OB12
Time of Day Interrupt 2
OB13
Time of Day Interrupt 3
Select All
Help on OB
Language for Selected Blocks
STL
LAD
FBD
Create with source files
Preview<<
S7_Pro3
SIMATIC 300 Station
CPU313C(1)
S7 Program(1)
Blocks
Block Name
Symbolic Name
OB1
Cycle Execution
< Back
Next >
Finish
Cancel
Help

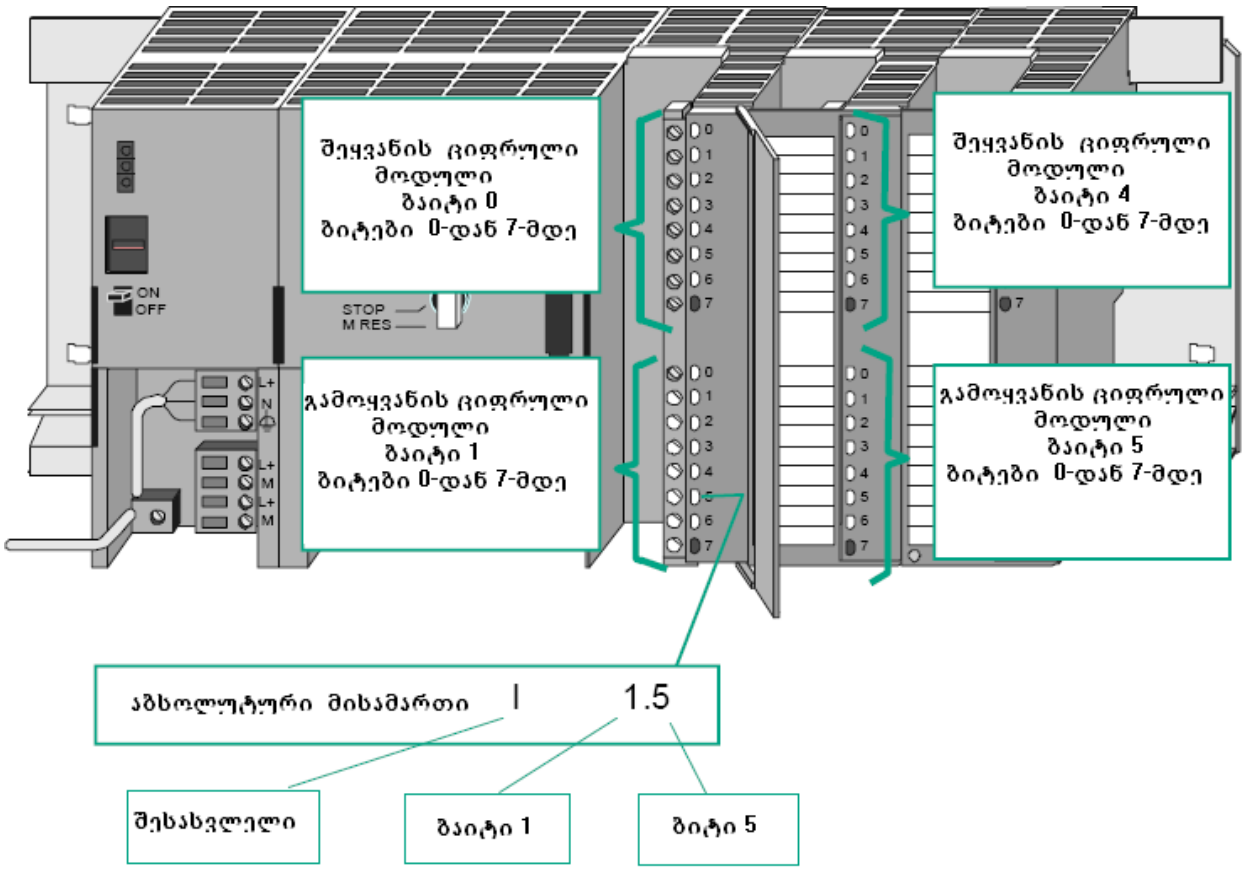


ნახ. 5 პროექტის გარე ხედი

ბოლო ეტაპზე აუცილებელია ამ თქვენი პროექტისათვის სახელწოდების მინიჭება, ამ მიზნით საჭიროა, რომ მისი სახელი ჩაიწეროს ველში **Project Name** და დაეჭიროს ღილაკს **Finish.** ამის შემდეგ Simatic Manager შექმნის პროექტს, რომლის გარე სახე ნაჩვენებია ნახ. 5-ზე.

Simatic-ში არსებობს საცნობარო ინფორმაციის შესანიშნავი სისტემა, მათ რიცხვში კონტექსტურ-დამოკიდებულიც. ცნობის მიღებისათვის აუცილებელია ობიექტის გამოყოფა და შემდეგ F1 ღილაკის დაჭერა.

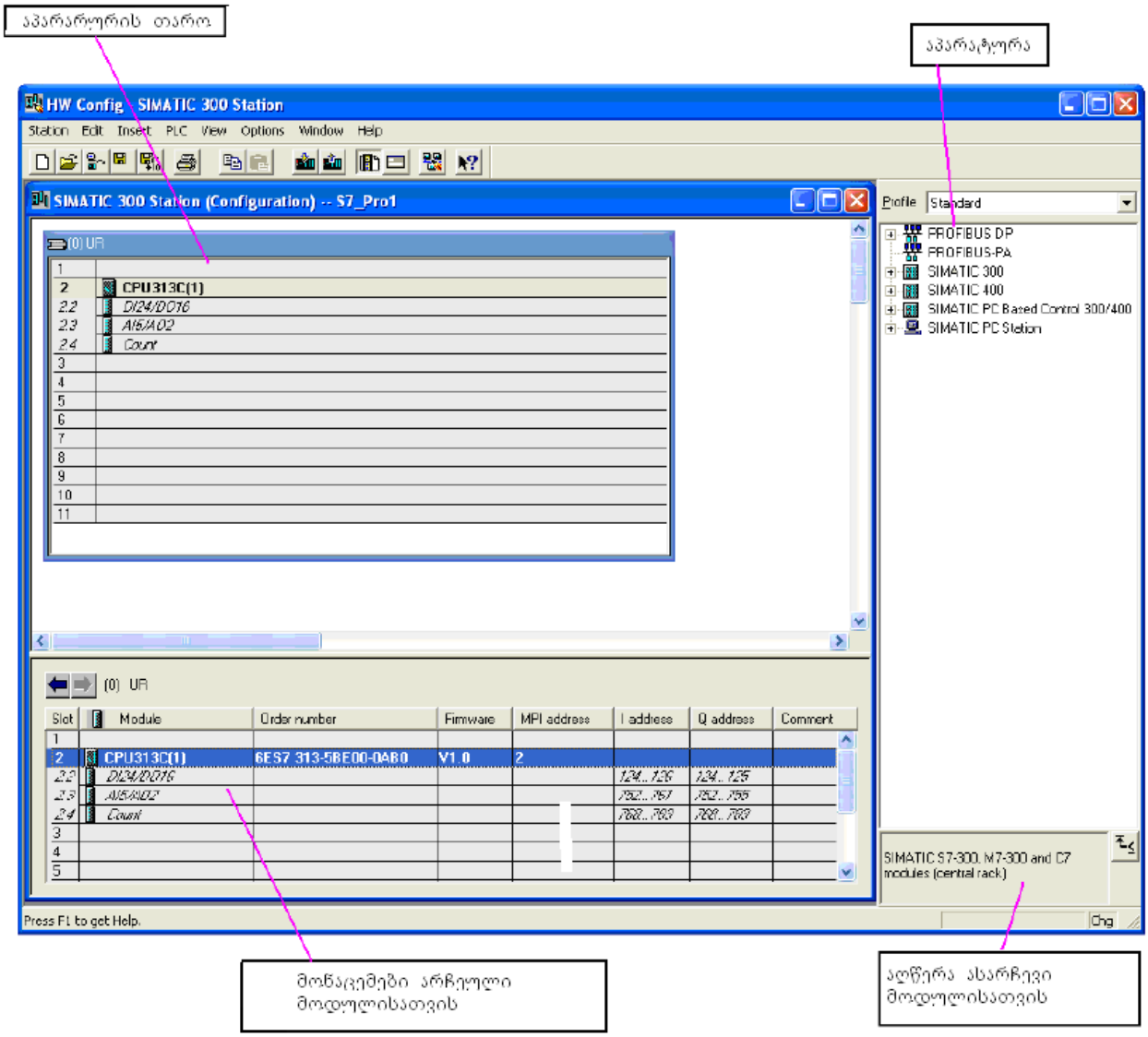
ახლა საჭიროა შევეხოთ შესასვლელ-გამოსასვლელების დამისამართების პრინციპს. დამისამართების სტრუქტურა ნაჩვენებია ნახ. 6-ზე.



ნახ. 6 დამისამართება Simatic-ში

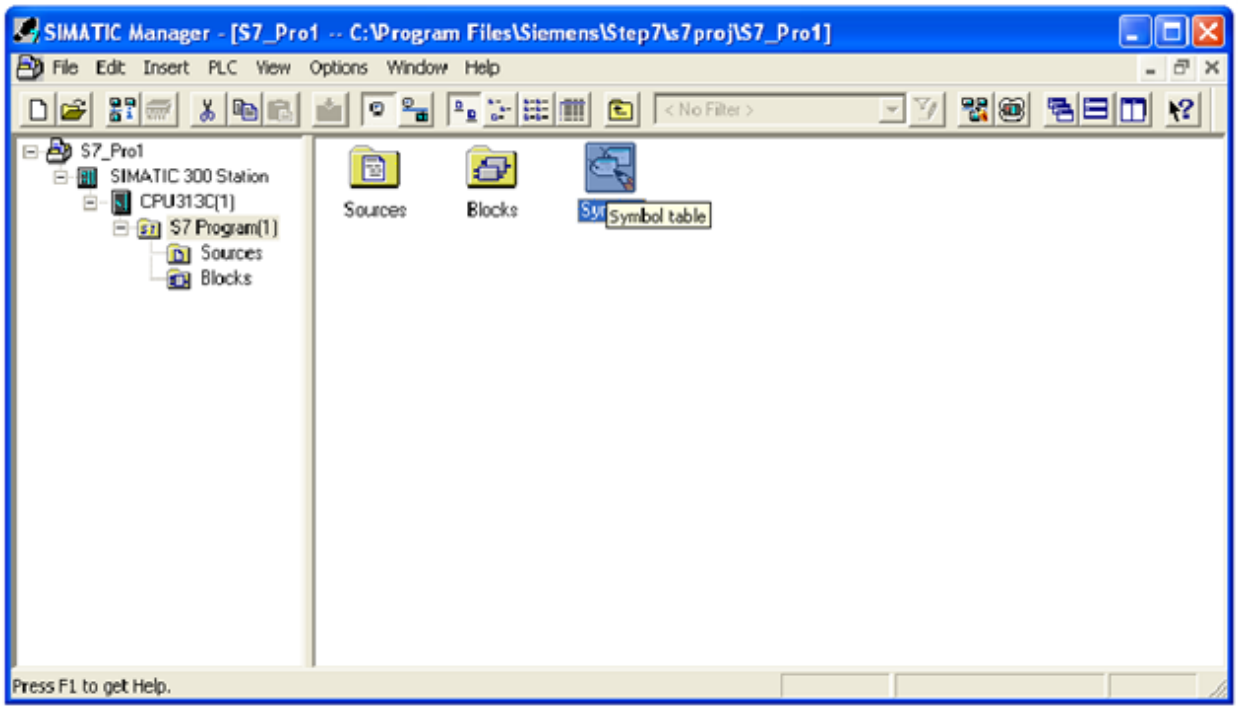
როგორც ჩანს, შესასვლელების ჯგუფზე შეიძლება მიკითხება როგორც ბაიტზე-მიკითხება მთლიანად შესასვლელების ჯგუფზე, გარდა ამისა შესაძლებელია მიკითხვა ცალკეულ შესასვლელებზეც-ბიტებზე. ანალოგური შესასვლელი წარმოდგენილნი იქნება როგორც 11-ბიტიანი სიტყვები.

იმისათვის, რომ შევხედოთ აბსოლუტურ მისამართებს, რომლებიც დასაშვებია კონკრეტული კონტროლერისათვის, აუცილებელია პროექტის სტრუქტურის **SIMATIC 300 Station** განყოფილებაზე თაგვის მარცხენა ღილაკით დაჭერა და ამის შემდეგ მარჯვენა პანელზე მოთავსებულ საქაღალდეზე Hardware ისევე მარცხენა ღილაკით ორჯერ დაჭერა. თქვენს წინაშე გაიღება ფანჯარა, რომელიც წარმოდგენილია ნახ. 7-ზე.

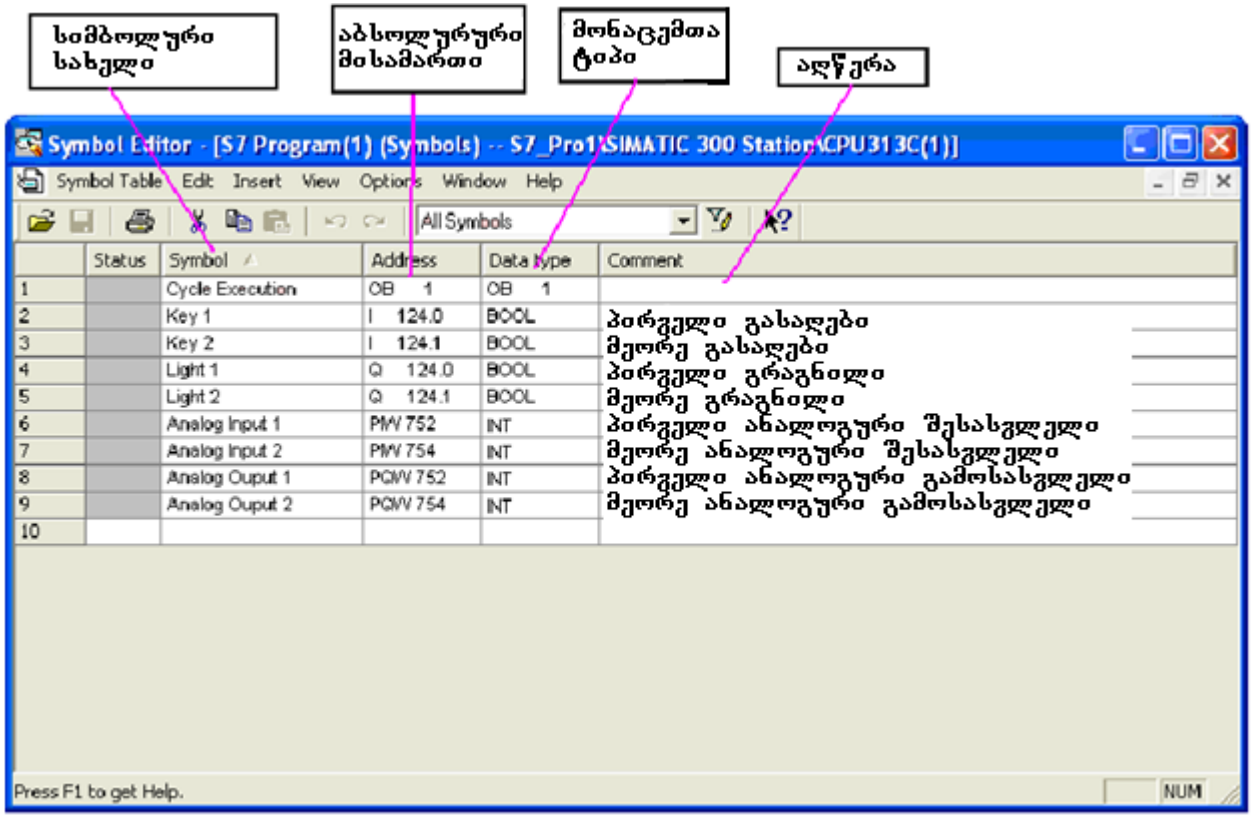


ნახ. 7 აპარატურის კონფიგურირება

პროგრამის დაწერისათვის უფრო მოსახერხებელია თუ ჩვენ აბსოლუტურ მისამართებს შევცვლით სიმვოლური სახელწოდებებით. მისამართებზე სიმვოლური სახელების მინიჭებისათვის აუცილებელია სიმვოლოთა ცხრილის გააქტიურება, რისთვისაც პროექტების ხეში საჭიროა თაგვის მარცხენა ღილაკით **S7 Program** საქაღალდეზე დაჭერა და შემდეგ მარჯვენა პანელში Symvols საქაღალდეზე ორჯერ მარცხენა ღილაკით დაჭერა (ნახ. 8). სიმვოლოთა ცხრილის გააქტიურების შემდეგ თქვენ დაინახავთ ფანჯარას, რომელიც ნაჩვენებია მე-9 ნახაზზე .



ნახ. 8 სიმვოლოთა ცხრილის გააქტიურება



ნახ. 9 სიმბოლოთა ცხრილი

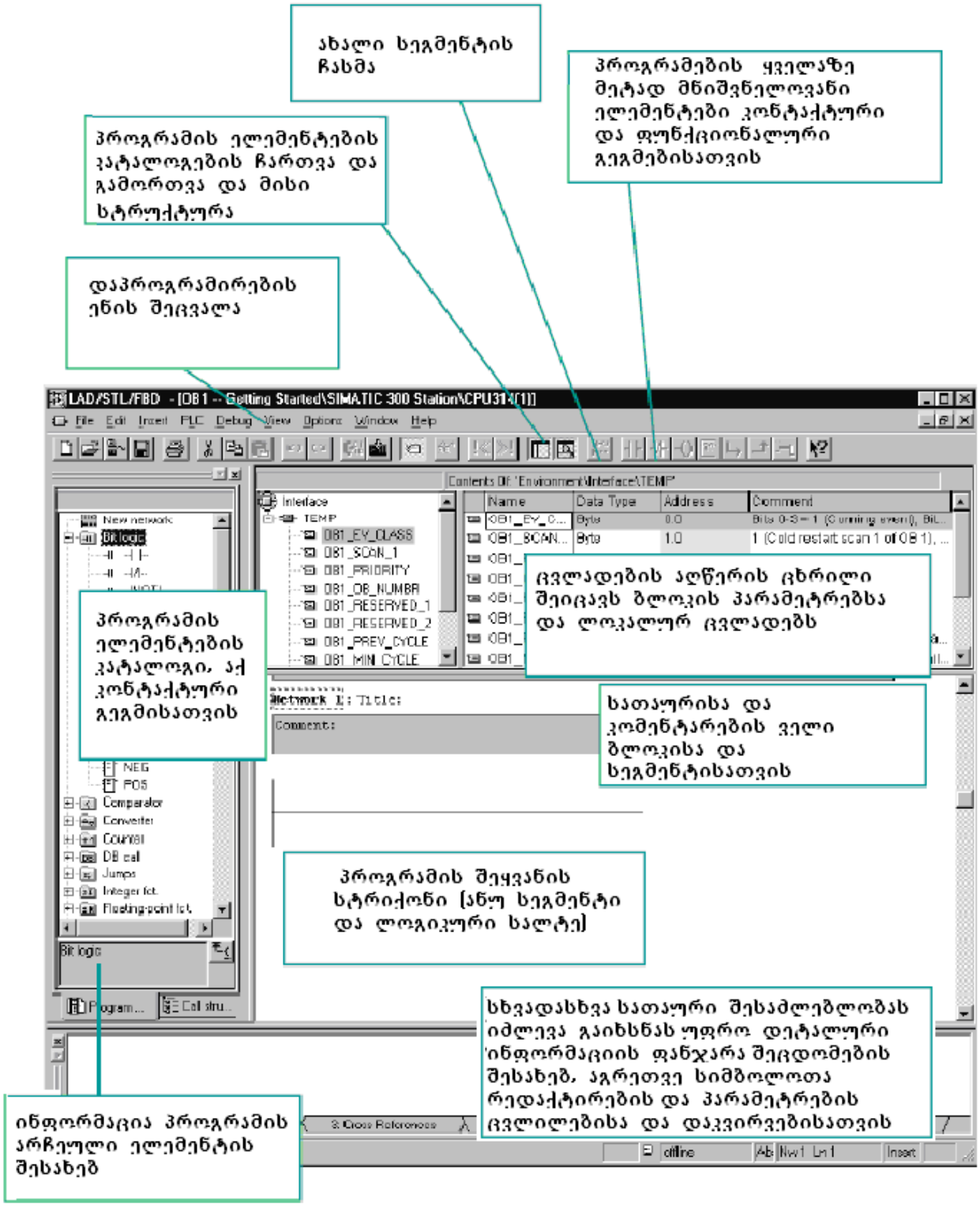
პირველად ცხრილში იქნება მხოლოდ ერთი ცვლადი OB1. შეავსეთ სიმვოლოთა ცხრილი ნახაზის მიხედვით და შეინახეთ ის. შეხედეთ თუ როგორაა აღნიშნული როგორც ციფრული შესასვლელები და გამოსასვლელები ასევე ანალოგურიც. უნდა აღინიშნოს, რომ მაქსიმალური მისამართი ერთი ჯგუფის შესასვლელისთვის ბუნებრივია იქნება XXX.7, ხოლო ანალოგურ შესასვლელებსა და გამოსასვლელებს აქვთ სიგრძე ორი სიტყვის ტოლი, ანუ პირველი ანალოგიური შესასვლელი იკავებს მისამართებს 752 და 753-ს.

შესაძლებელ მონაცემებად შეიძლება შეგვხვდეს:

BOOL BYTE WORD DWORD	ამ ტიპის მონაცემები ბიტების კომბინაციები არის დაწყებული 1 ბიტიდან დამთავრებული 32 ბიტამდე (DWORD)
CHAR	ამ ტიპის მონაცემები იკავებს ზუსტად ერთ სიმბოლოს სიმბოლოების ნაკრებიდან ASCII.
INT DINT REAL	ეს მონაცემები ხელმისაწვდომია რიცხვითი სიდიდეების დამუშავებისათვის (მაგალითად, ართმეტიკული გამოსახულებების გაანგარიშების დროს).
S5TIME TIME DATE TIME_OF_DAY	ამ ტიპის მონაცემები წარმოადგენს დროის სხვადასხვა მნიშვნელობებს Step 7-ის შიგნით (მაგალითად, იმისთვის, რომ დაყენდეს თარიღი, ანდა დროის მნიშვნელობის შესაყვანად ტაიმერისთვის).

უფრო დაწვრილებითი ცნობები მონაცემების შესახებ შეიძლება მონახულ იქნას პროგრამის საცნობარო სისტემაში.

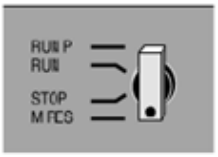
შევქმნათ უმარტივესი პროგრამა და დავაკვირდეთ, თუ როგორ იქმნება პროგრამები Step-7-ში. დასაწყისისთვის დავაწკაპოთ პროექტების ხეში **Blocks** საქაღალდეს, შემდეგ კი ორჯერ დავაწკაპოთ **OB1** ბლოკზე, ჩვენ თვალწინ გაიშლება ფანჯარა, რომელიც ნაჩვენებია ნახ. 10-ზე. პროგრამის ღილაკებს აქვთ ინტუიციურად ადვილად გასაგები მნიშვნელობები და ამოცურებადი მოკარნახეები. შექმენით პროგრამა რომლის გარე სახე ნაჩვენებია ნახ. 11-ზე. ღილაკები, რომლებიც ჩვენ დაგვჭირდება პროგრამის შედგენის დროს, შემოხაზულია. შეინახეთ პროგრამა და განახორციელეთ პროგრამის ჩატვირთვა.

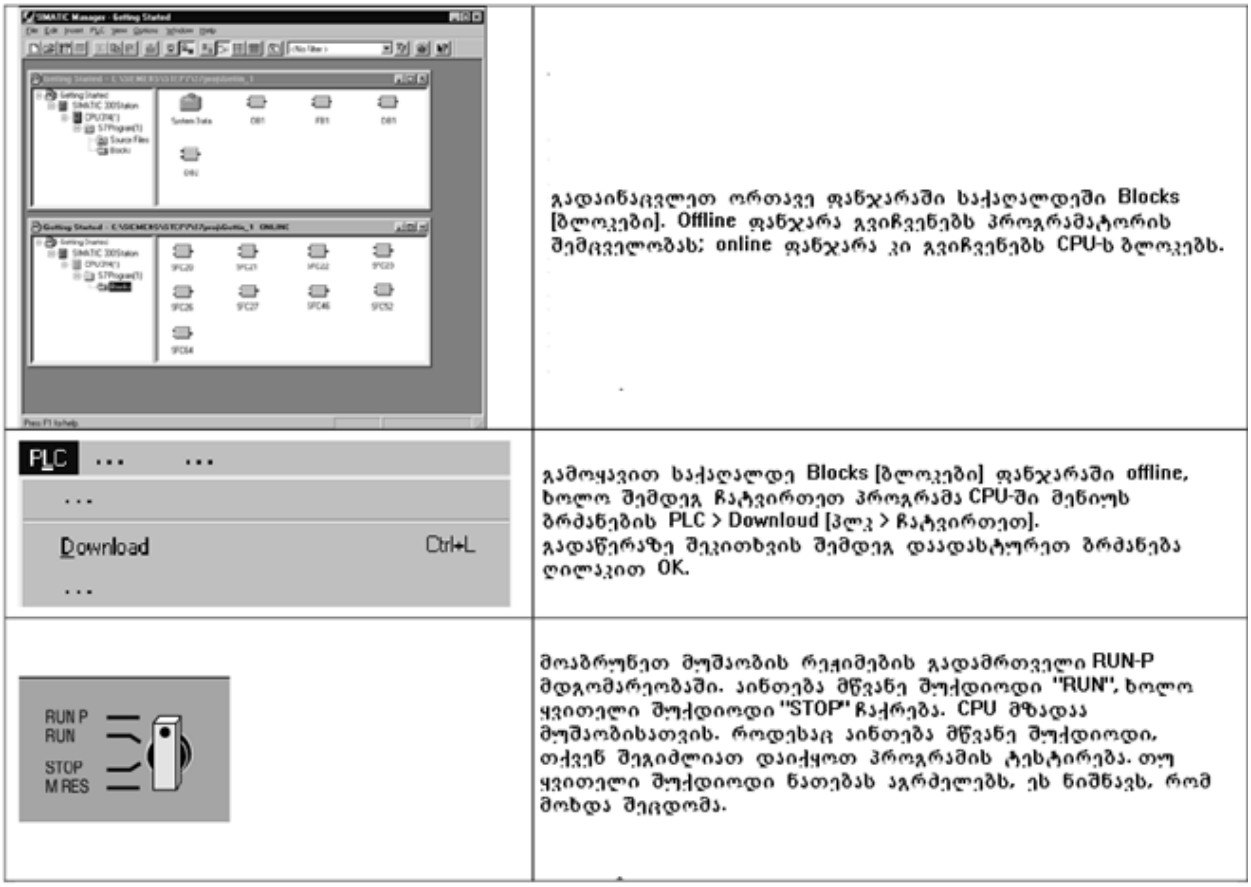


ნახ. 10 ბლოკების რედაქტორის ფანჯარა

3.4. პროგრამის ჩატვირთვა პროგრამირებად კონტროლერში

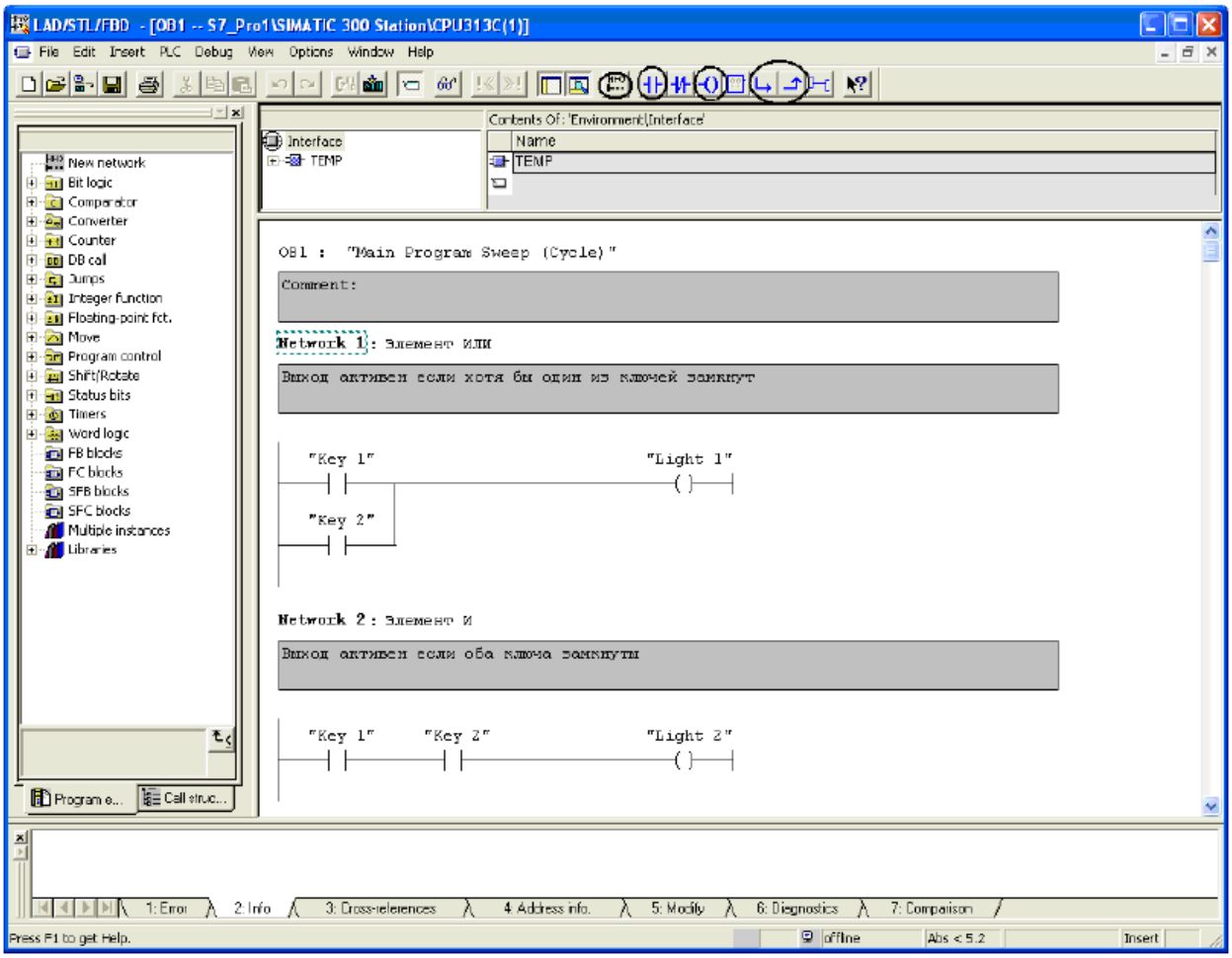
კონტროლერის ჩართვის თანმიმდევრობა და პროგრამის ჩატვირთვა ნაჩვენებია ქვემოთ.

	ჩართეთ სტენდის კვება. ჩართეთ გამომრთველი Power გამომავალი სიგნალების პანელზე (მარცხენა პანელი)
	ჩართეთ კვების ბლოკი ON/OFF (ჩართვა/გამორთვა) გადამრთველის საშუალებით. CPU - ზე აინთება დიოდი "DC 5V" (მუდმივი დენის 5 ვ)
	მუშაობის რეჟიმის გადამრთველი მოაბრუნეთ STOP მდგომარეობაში (თუ ის ჯერ კიდევ ამ მდგომარეობაში არ იმყოფება). აინთება წითელი შუქდიოდი "STOP" LED
	მოაბრუნეთ მოშაობის რეჟიმის გადამრთველი MRES მდგომარეობაში და დაყოვნეთ იგი არანაკლებ 3 წამის განმავლობაში, მანამ სანამ ყვითელი შუქდიოდი "STOP" არ დაიწყებს წელ ციმციმს. გაუშვით გადამრთველი და მანამ 3 წამი გავა იხევ დააბრუნეთ იგი MRES მდგომარეობაში. როდესაც შუქდიოდი "STOP" დაიწყებს სწრაფ ციმციმს, CPU ჩამოიყრება. თუ კი შუქდიოდი "STOP" არ დაიწყებს სწრაფად ციმციმს, გაიმეორეთ ეს პროცედურა (ამასთან თქვენ წაშალეთ CPU-ში არსებული პროგრამა).
	იმისათვის, რომ ჩატვირთოთ პროგრამა, ახლა იხევ მოაბრუნეთ მუშაობის რეჟიმის გადამრთველი "STOP" მდგომარეობაში.
	"თქვენი პროექტის" ფანჯრის გარდა გახსენით ფანჯარა "თქვენი პროექტი ONLINE". Online ან offline მდგომარეობა ნაჩვენები იქნება სხვადასხვა ფერის სათაურების დახმარებით.

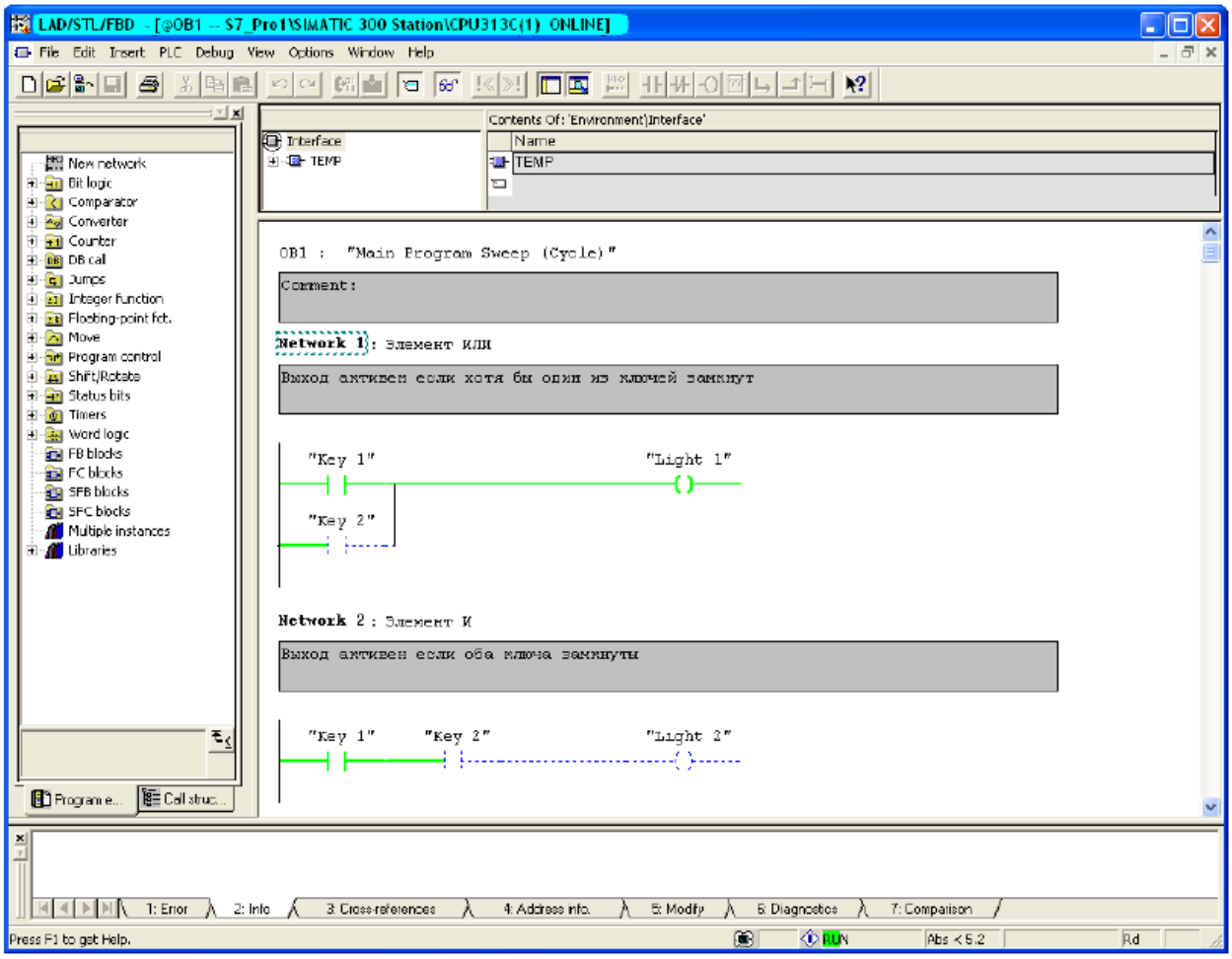


პროგრამის დათვალიერებისა და გაწყობისათვის დააჭირეთ ღილაკს  გახსნილი OB1 ბლოკის ფანჯარაში. ჩართეთ გასაღები I 124.0, და პროგრამის ეკრანზე თქვენ ნახავთ სურათს (იხ. ნახ. 12). დროის მოცემულ მომენტში ჩართულია ერთი გასაღები და პროგრამის ლოგიკის შესაბამისათ ანათებს ნათურა Q 124.0 გამოსასვლელზე. ჩაატარეთ ექსპერიმეტრები პროგრამაზე, რომ გაიგოთ თუ როგორ მუშაობს Step 7. უნდა აღინიშნოს, რომ სახელწოდებების მინიჭება შესასვლელ/გამოსასვლელებს სიმბოლოთა ცხრილში სრულებითაც არაა აუცილებელი, შეიძლება შეიყვანოთ უშუალოდ აბსოლუტური მისამართი.

დააჭირეთ ბლოკების რედაქტორში ღილაკს  , და სახელები შეიცვლება აბსოლუტური მისამართებით. და კიდევ ერთი მოსახერხებლობა, რაც დაგეხმარებათ თქვენ პროგრამების სწრაფად შექმნაში: თუ კი მისამართის შეყვანის დროს დააჭერთ **Ctrl+Space**, მაშინ Step 7 თვითონ შემოგთავაზებთ სახელების არჩევას, რომლებიც სიმბოლოთა ცხრილშია მოთავსებული (ამ დროს გაფილტრეთ ისინი, რომლებსაც ამ მნიშვნელობისთვის აზრი არაა აქვთ).



ნახ. 11 პროგრამის გარე სახე

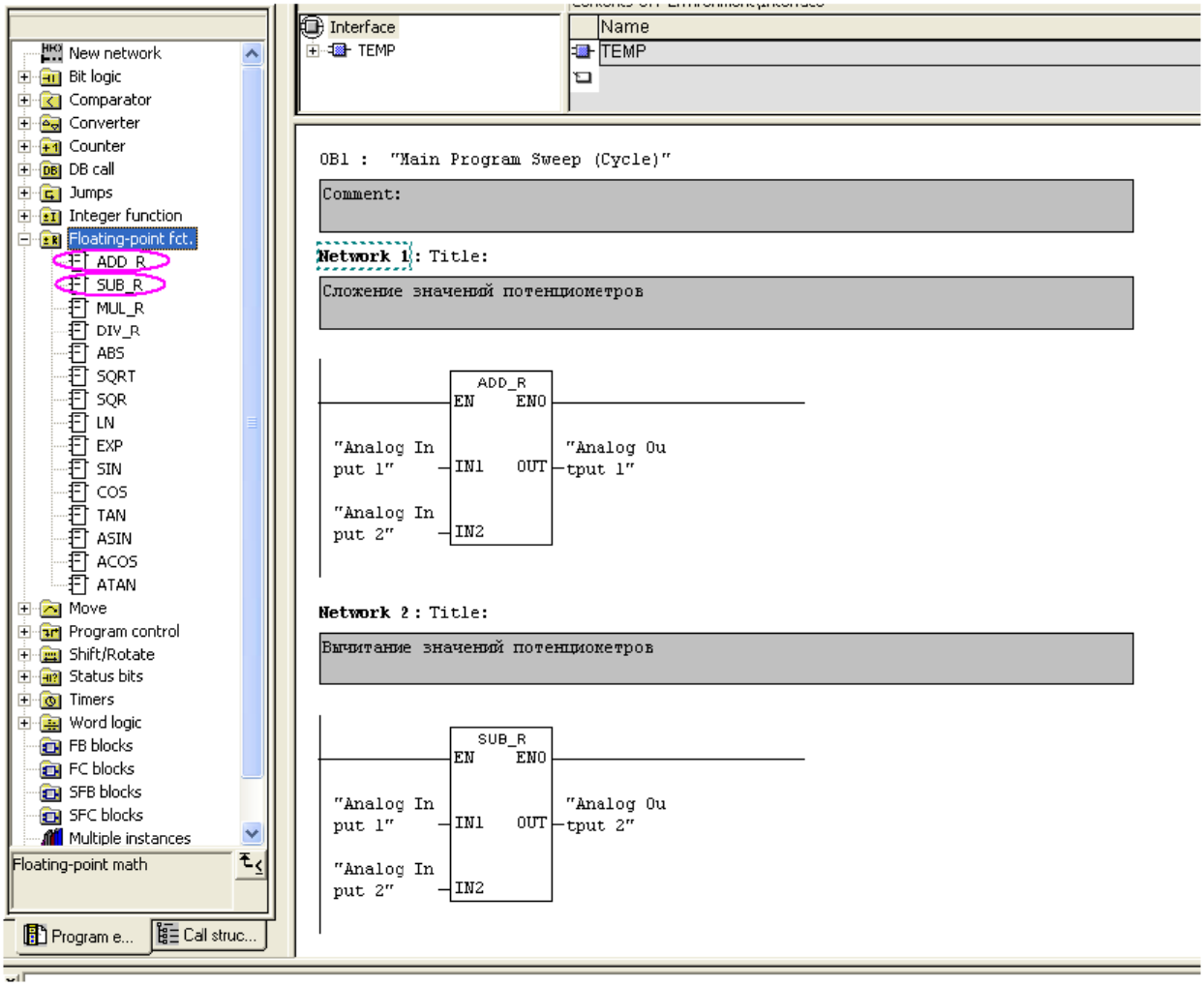


ნახ. 12 პროგრამის გაწყობა

Step 7 პროგრამას აქვს მდიდარი შესაძლებლობები პროგრამების შექმნისათვის, დაწყებული ოპერაციებით მცოცავმძიმიან რიცხვებზე და დამთავრებული სამრეწველო ქსელების აგებით, და მათი კონფიგურირებით (თუმც ეს დამოკიდებულია კიდევ თვითონ CPU-ს შესაძლებლობებზე). უფრო მოსახერხებელია პროგრამირების შესწავლა რაიმე პროექტის შექმნის დაწყებით და არა რაიმე მარტივი კონსულტაციებით, რითაც ჩვენ დავკავდებით შემდგომში.

ანალოგური შესასვლელებისა და გამოსასვლელების გამოყენება ანალოგიურია (ეს აგრეთვე დამოკიდებულია კიდევ თვით CPU-ს შესაძლებლობებზე). შექმენით პროგრამა, რომელიც ნაჩვენებია ნახ. 13-ზე, ანალოგური შესასვლელებისა და გამოსასვლელების კავშირისათვის (მაგალითი - ანალოგური კალკულიატორი). აღნიშნული პროგრამის შექმნისათვის აუცილებელი ბლოკები გამოყოფილია მასზედ. მიაქციეთ ყურადღება, რომ ანალოგიური შესასვლელი (10 ვ) წარმოგვიდგება უზარმაზარი რიცხვით, იმიტომ,

რომ შესასვლელი მრავალბიტურია, თუმც მისი გარჩევადობა მხოლოდ $2^{12}=4096$-ია, ანუ 10 ვ დაყოფილია 4096 ნაწილად. თუ კი თქვენ მუშაობთ უშუალოდ კონტროლერთან მაშინ აუცილებელია აღებულ იქნას ჩანართის ბლოკი Integer fanction. გაუშვით პროგრამა და პირველ ვოლტმეტრზე ნაჩვენები იქნება ძაბვის ვარდნების ჯამი პოტენციომეტრებზე, ხოლო მეორეზე - მათი სხვაობა.

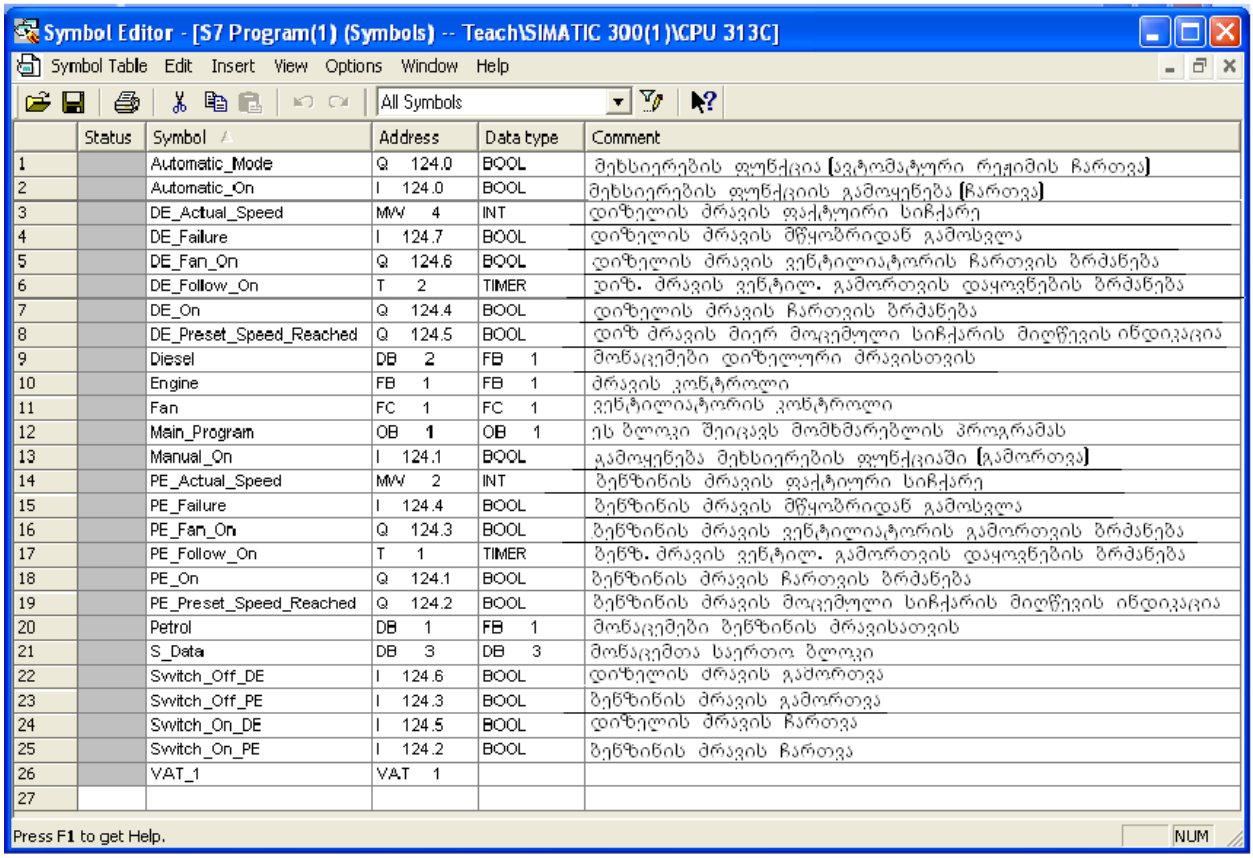


ნახ. 13 ანალოგიური კალკულიატორი

ჩვენს სამუშაოში შევეცდებით, არა მარტო გადმოვცეთ პროგრამის შედგენის საფუძვლები, არამედ მისი ოპტიმიზაციის ხერხებიც. შევქმნათ მარტივი პროგრამა ორი ძრავის მართვისათვის: ბენზინის და დიზელის ძრავებისთვის. ბუნებრივია, ნებისმიერი ძრავი უნდა ჩაირთოს და გამოირთოს. ასევე აუცილებელია ყურადღება მიექცეს ძრავის გამართულობას (გამოირთოს

იგი ავარიის დროს) და რათქმა უნდა აუცილებელია ყურადღება მიექცეს მის სიჩქარეს. აი ასეთი მართვის პროგრამას შევქმნით ჩვენ.

პროგრამის დაწერის მოხერხებულობისათვის საჭიროა ცხრილის შექმნა (იხ. ნახ. 14), ამასთან ერთად თქვენ შეგიძლიათ დატოვოთ უკვე შექმნილი პროექტი.



	Status	Symbol	Address	Data type	Comment
1		Automatic_Mode	Q 124.0	BOOL	მეხსიერების ფუნქცია (ავტომატური რეჟიმის ჩართვა)
2		Automatic_On	I 124.0	BOOL	მეხსიერების ფუნქციის გამოყენება (ჩართვა)
3		DE_Actual_Speed	MW 4	INT	დიზელის ძრავის ფაქტიური სიჩქარე
4		DE_Failure	I 124.7	BOOL	დიზელის ძრავის მწყობრიდან გამოსვლა
5		DE_Fan_On	Q 124.6	BOOL	დიზელის ძრავის ვენტილიატორის ჩართვის ბრძანება
6		DE_Follow_On	T 2	TIMER	დიზ. ძრავის ვენტილ. გამორთვის დაყოვნების ბრძანება
7		DE_On	Q 124.4	BOOL	დიზელის ძრავის ჩართვის ბრძანება
8		DE_Preset_Speed_Reached	Q 124.5	BOOL	დიზ ძრავის მიერ მოცემული სიჩქარის მიღწევის ინდიკაცია
9		Diesel	DB 2	FB 1	მონაცემები დიზელური ძრავისთვის
10		Engine	FB 1	FB 1	ძრავის კონტროლი
11		Fan	FC 1	FC 1	ვენტილიატორის კონტროლი
12		Main_Program	OB 1	OB 1	ეს ბლოკი შეიცავს მომხმარებლის პროგრამას
13		Manual_On	I 124.1	BOOL	გამოყენება მეხსიერების ფუნქციაში (გამორთვა)
14		PE_Actual_Speed	MW 2	INT	ბენზინის ძრავის ფაქტიური სიჩქარე
15		PE_Failure	I 124.4	BOOL	ბენზინის ძრავის მწყობრიდან გამოსვლა
16		PE_Fan_On	Q 124.3	BOOL	ბენზინის ძრავის ვენტილიატორის გამორთვის ბრძანება
17		PE_Follow_On	T 1	TIMER	ბენზ. ძრავის ვენტილ. გამორთვის დაყოვნების ბრძანება
18		PE_On	Q 124.1	BOOL	ბენზინის ძრავის ჩართვის ბრძანება
19		PE_Preset_Speed_Reached	Q 124.2	BOOL	ბენზინის ძრავის მოცემული სიჩქარის მიღწევის ინდიკაცია
20		Petrol	DB 1	FB 1	მონაცემები ბენზინის ძრავისათვის
21		S_Data	DB 3	DB 3	მონაცემთა საერთო ბლოკი
22		Switch_Off_DE	I 124.6	BOOL	დიზელის ძრავის გამორთვა
23		Switch_Off_PE	I 124.3	BOOL	ბენზინის ძრავის გამორთვა
24		Switch_On_DE	I 124.5	BOOL	დიზელის ძრავის ჩართვა
25		Switch_On_PE	I 124.2	BOOL	ბენზინის ძრავის ჩართვა
26		VAT_1	VAT 1		
27					

ნახ. 14 სიმბოლოების ცხრილი პროექტის შექმნისათვის

3.5. ფუნქციონალური ბლოკის შექმნა

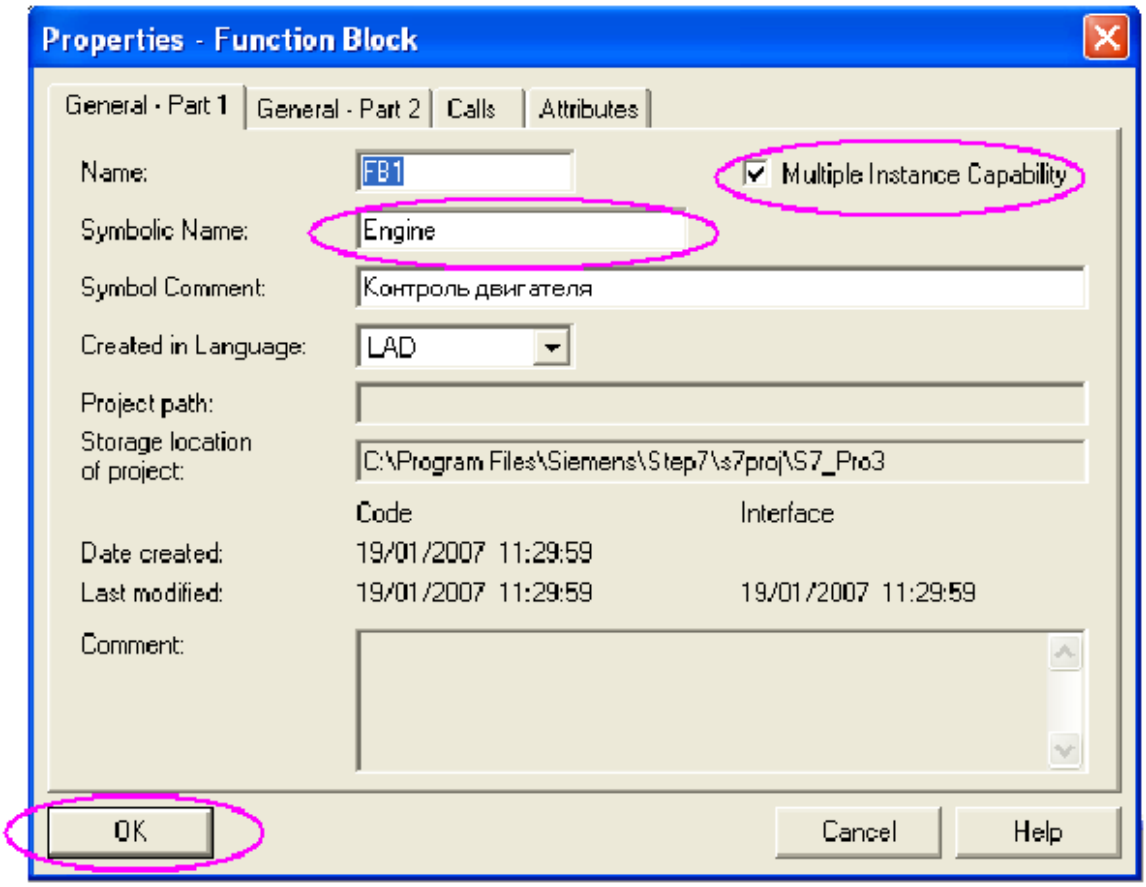
ფუნქციონალური ბლოკი (FB) მოთავსებულია პროგრამაში იერარქიულად ორგანიზაციული ბლოკის ქვემოთ. ის შეიცავს პროგრამის ნაწილს, რომელიც შეიძლება გამოძახებულ იქნას მრავალჯერ OB1 ბლოკში. ფუნქციონალური ბლოკის ყველა ფორმალური პარამეტრი და სტატისტიკური მონაცემები შეინახება მონაცემთა ცალკე ბლოკში (DB), რომელიც დაენიშნება ფუნქციონალურ ბლოკს. თქვენ დააპროგრამირებთ ფუნქციონალურ ბლოკს (FB1-ს სიმბოლური სახელით Engine [ძრავი]; იხ. სიმბოლოთა ცხრილი) ფანჯარაში LAD-ის პროგრამირებისათვის, რასაც თქვენ უკვე იცნობთ. ამისათვის თქვენ უნდა

გამოიყენოთ იგივე პროგრამირების ენა, რაც გამოიყენებოდა მე-4 თავში (OB1-ის პროგრამირება).

გახსენით პროექტში საქაღალდე **Blocks [ბლოკები]**. დააწკაპეთ ფანჯრის მარჯვენა ნახევარში თაგვის მარჯვენა ღილაკით. მარჯვენა ღილაკით ამოცურებული მენიუ შეიცავს უმნიშნელოვანეს ბრძანებებს მენიუს სტრიქონიდან. ჩასვით ახალი ობიექტის სახით **Function Block [ფუნქციონალური ბლოკი] (New Object – Function Block).**

Propertis – Fanctiun Block [თვისება - ფუნქციონალური ბლოკი] დიალოგიურ ფანჯარაში აირჩიეთ ის ენა, რომელზედაც თქვენ გინდათ ამ ბლოკის შექმნა, გააქტიურეთ ტრიგერული ღილაკი **Multiple instance FB** [მულტიეკზემპლიარული FB] და დაეთანხმეთ აწყობის დანარჩენ პარამეტრებს **OK** - ს დაწკაპებით. FB1 ფუნქციონალური ბლოკი ჩასმულ იქნება ბლოკების საქაღალდეში **(Blocks).** ორჯერ დააწკაპეთ FB1 ნიშანზე, რომ გახსნათ იგი.

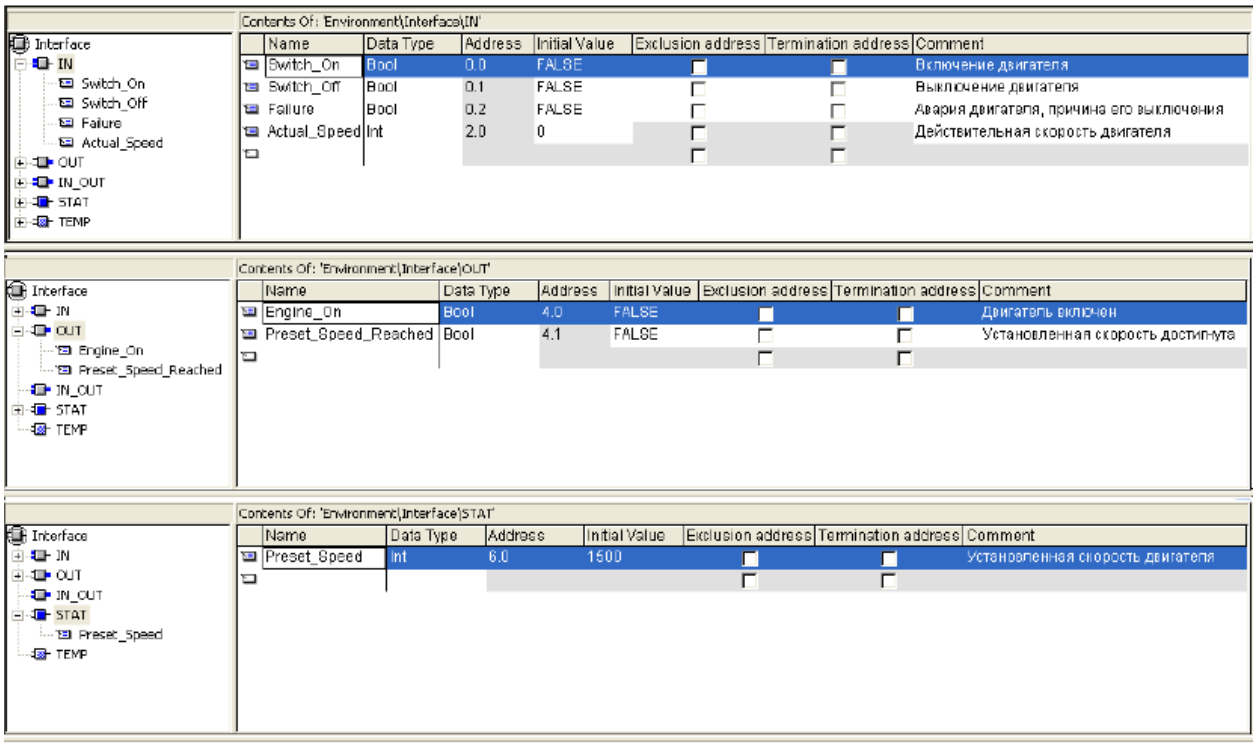
ახლა ვნახოთ თუ როგორ პროგრამირდება ფუნქციონალური ბლოკი. ძრავების მართვის ყველა მონაცემი უნდა გადაეცეც ფუნქციონალურ ბლოკს საორგანიზაციო ბლოკისაგან როგირც ბლოკის პარამეტრები და ამიტომ ცვლადების აღწერის ცხრილში განსაზღვრულნი უნდა იყვნენ როგორც შეშასვლელი და გამოსასვლელი ცვლადები (**in** და **out** აღწერა).



ნახ. 15 ფუნქციონალური ბლოკის შექმნა

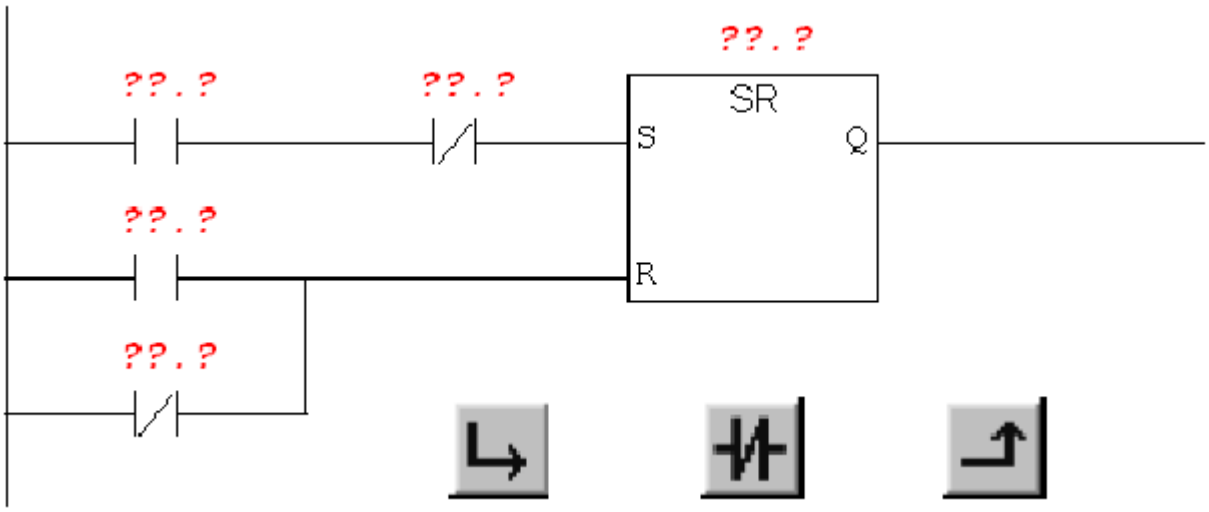
დეკლარაციის ცხრილი შედგება ცვლადების სახეობისაგან (მარცხენა ნაწილი) და გადაცემული პარამეტრების დეტალური ასახვისაგან (ცხრილის მარჯვენა ნაწილი). ავირჩევთ რა მარცხენა ნაწილში ცვლადების ტიპს IN, OUT, IN_OUT ან START, შეიტანეთ ცვლადების აუცილებელი სახელები, მონაცემთა ტიპები და აუცილებელი კომენტარები ცვლადების აღწერის ცხრილის მარჯვენა ნაწილში (კომენტარები უმჯობესია ჩავსვათ ბრჭყალებში). თქვენ შეიძლება გამოიყენოთ ჩამოშლადი მენიუ ცვლადებზე შესაბამისი ტიპის მინიჭებისათვის. საბოლოოდ თქვენ უნდა მიიღოთ დეკლარაციის ცხრილი, რაც ნახ. 16-ზე ნაჩვენები.

ცვლადების აღწერის ცხრილში ბლოკის პარამეტრების სახელებისათვის ნებადართულ სიმბოლოებათ შეიძლება გამოყენებულო იქნას მხოლოდ ასოები, ციფრები და ხაზგასმის ნიშნები, გარდა ამისა ნებადართულია მხოლოდ ლათინური ასოების გამოყენება ცვლადების აღწერის დროს.

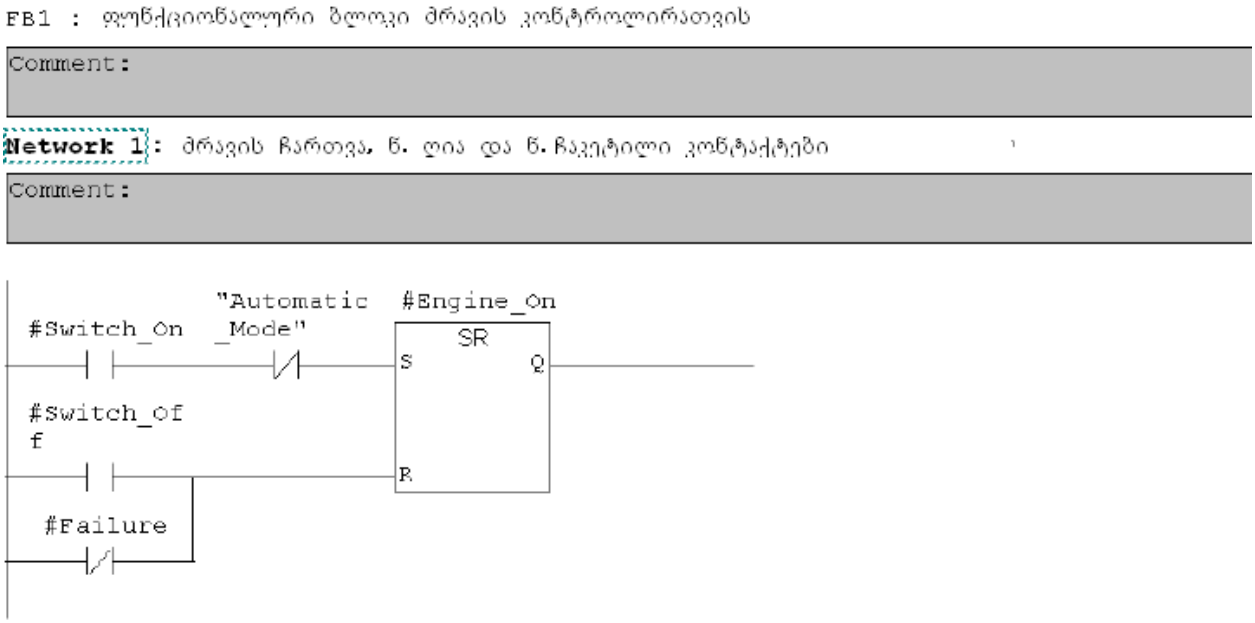


ნახ. 16 დეკლარაციების ცხრილი

ჩასვით ნორმალურად ღია კონტაქტი, ნორმალურად დაკეტილი კონტაქტი და SR ელემენტი 1 სეგმენტში (Network 1), რისთვისაც გამოიყენეთ ინსტრუმენტების პანელზე განლაგებული შესაბამისი ღილაკები ანდა პროგრამის ელემენტების კატალოგი. შემდეგ გამოჰყავით უშუალოდ R შესასვლელი. ჩასვით კიდევ ერთი ნორმალურად ღია კონტაქტი. გამოჰყავით სალტე უშუალოდ ამ კონტაქტის წინ. (იხ. ნახ. 17). ჩასვით ნორმალურად დაკეტილი კონტაქტი ნორმალურად ღია კონტაქტის პარარელურად. ყველა ამ ქმედების შედეგე ნაჩვენებია ნახ. 18-ზე.



ნახ. 17 მოქმედებათა თანმიმდევრობა

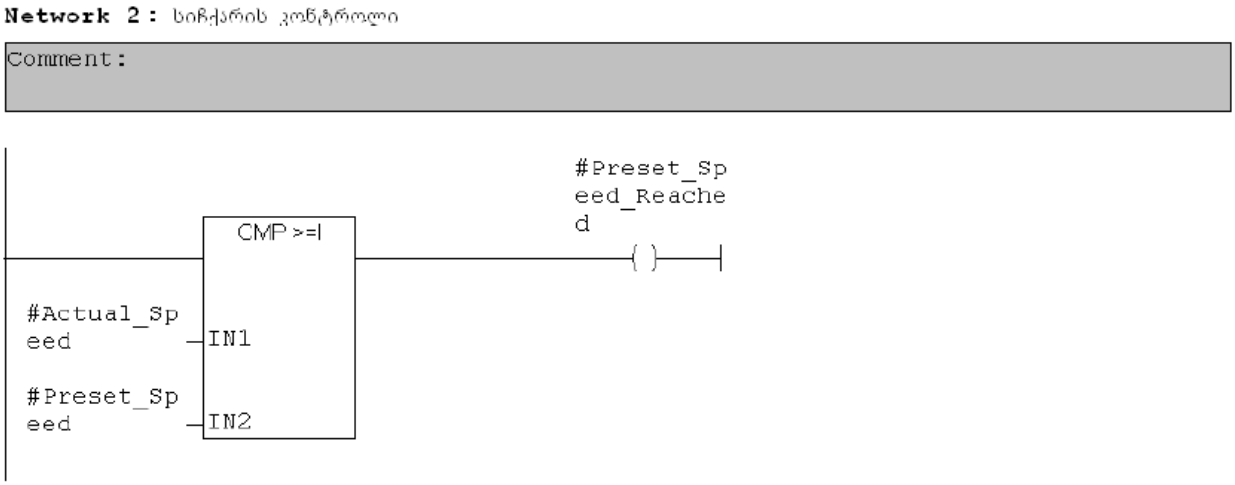


ნახ. 18 სეგმენტი

შეამოწმეთ გააქტიურებულია თუ არა სიმბოლური ასახვა (Symbolic Representation) (View -> Display with). გამოყავით კოთხვის ნიშნები და შეიყვანეთ შესაბამისი სახელები ცვლადების აღწერის ცხრილიდან (# სიმბოლო ისმება ავტომატურად. რეზულტატი ნაჩვენებია ნახ. 18-ზე.

ჩასვით ახალი სეგმენტი და გამოყავით სალტე. შემდეგ გადაინაცვლეთ პროგრამის ელემენტების კატალოგში, ვიდრე არ მიაღწევთ **Compare [შედარება]** ფუნქციას და ჩართეთ CMP>=1. შედარების ელემენტის გამოსასვლელზე

დააყენეთ გრაგნილი. ისევ გამოყევით კითხვითი ნიშნები და აღნიშნეთ გრაგნილი და შედარების ბლოკი სახელებით, რომლებსაც აიღებთ ცვლადების აღწერის ცხრილიდან. შემდეგ შეინახეთ თქვენი პროგრამა. მიღებული სეგმენტის სახე ნაჩვენებია ნახ. 19-ზე.



ნახ. 19 მიღებული სეგმენტის სახე

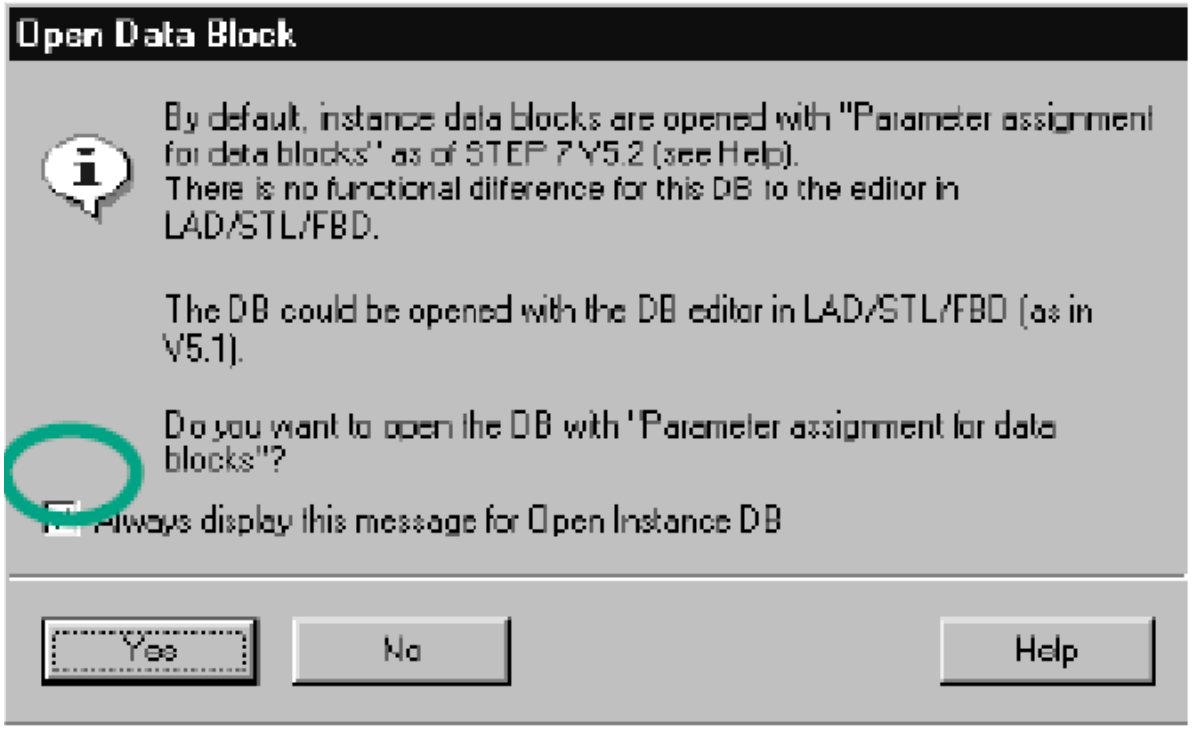
მაშასადამე რა მივიღეთ. როდესაც ცვლადს #Switch_On [ჩართეთ] აქვს მდგომარეობა „1“ და ცვლადს Automatic_Mode [ავტომატური_რეჟიმი] აწვს მდგომარეობა „0“, მაშინ ძრავი ჩაირთვება. ეს ფუნქცია დაშვებული არ არის, ვიდრე ავტომატური რეჟიმი არ გამოირთვება (უარყოფა Automatic_Mode, ჰორმალურად ჩართული კონტაქტი). როდესაც ცვლადს #Switch_Off [გამორთვა] ექნება მდგომარეობა „1“ ანდა ცვლადს #Fault [უწესრიგობა] აქვს მდგომარეობა „0“, მაშინ ძრავი გამოირთობა. ეს ფუნქცია ისევ სრულდება ცვლადი #Fault - ის უარყოფით (#Fault - ეს „ნულ-აქტიური სიგნალია, ის უდრის „1“-ს ნორმალურ მდგომარეობაში და „0“-ს თუ წარმოიქმნება უწესრიგობა). შედარების ბლოკი ადარებს ცვლადებს #Actual_Speed [ფაქტიური_სიჩქარე] და #Preset_Speed [დავალებული სიჩქარე] და შედარების რეზულტატს ადარებს #Preset_Speed_reached [დავალებული სიჩქარე მიღწეულია] (სიგნალის მდგომარეობა „1“).

3.6. მონაცემთა ეკზემპლიარების გენერაცია და ფაქტიური მნიშვნელობების ცვლილება

თქვენ ახლახანს შექმენით ფუნქციონალური ბლოკი FB1 (Engin [ძრავი] და განსაზღვრეთ სხვებთან ერთად ძრავის სპეციფიური პარამეტრები ცვლადების

აღწერის ცხრილში. იმისათვის, რომ მომავალში გქონდეთ OB1 ფუნქციონალური ბლოკის გამოძახების საშუალება, თქვენ უნდა მოახდინოთ შესაბამისი მონაცემთა ბლოკის გენერაცია. მონაცემთა ეკზემპლიარული ბლოკი (DB) ყოველთვის უნდა შეესაბამებოდეს ფუნქციონალურ ბლოკს. ფუნქციონალურმა ბლოკმა უნდა მართოს და აკონტროლოს ბენზინის ანდა დიზელის ძრავი. ძრავების სხვადასხვა მოთხოვნილი სიჩქარეები შენახულია ორ სხვადასხვა მონაცემთა ბლოკში, სადაც იცვლება მოთხოვნილი სიჩქარის მნიშვნელობები (#Preset_Speed [მოთხოვნილი სიჩქარე]. დააპროგრამირებთ რა ცენტრალიზებულად ფუნქციონალურ ბლოკს ერთჯერ, ამით თქვენ შეძლებთ შეამციროთ პროგრამირების მნიშვნელობა.

გადაინაცვლეთ საქაღალდეში **Blocks [ბლოკები]** და დააწკაპეთ ფანჯრის მარჯვენა ნახევარში თაგვის მარჯვენა ღილაკით. ჩასვით მონაცემთა ბლოკი (data block), ამოცურებადი მენიუს გამოყენებით. მიანიჭეთ სახელი ბლოკს DB1, აირჩიეთ ბლოკის ტიპი „Instance DB" (ბლოკ-ეკზემპლიარი) და დააყენეთ იგი FB1 ფუნქციონალური ბლოკის შესაბამისობაში (ნახ. 21).

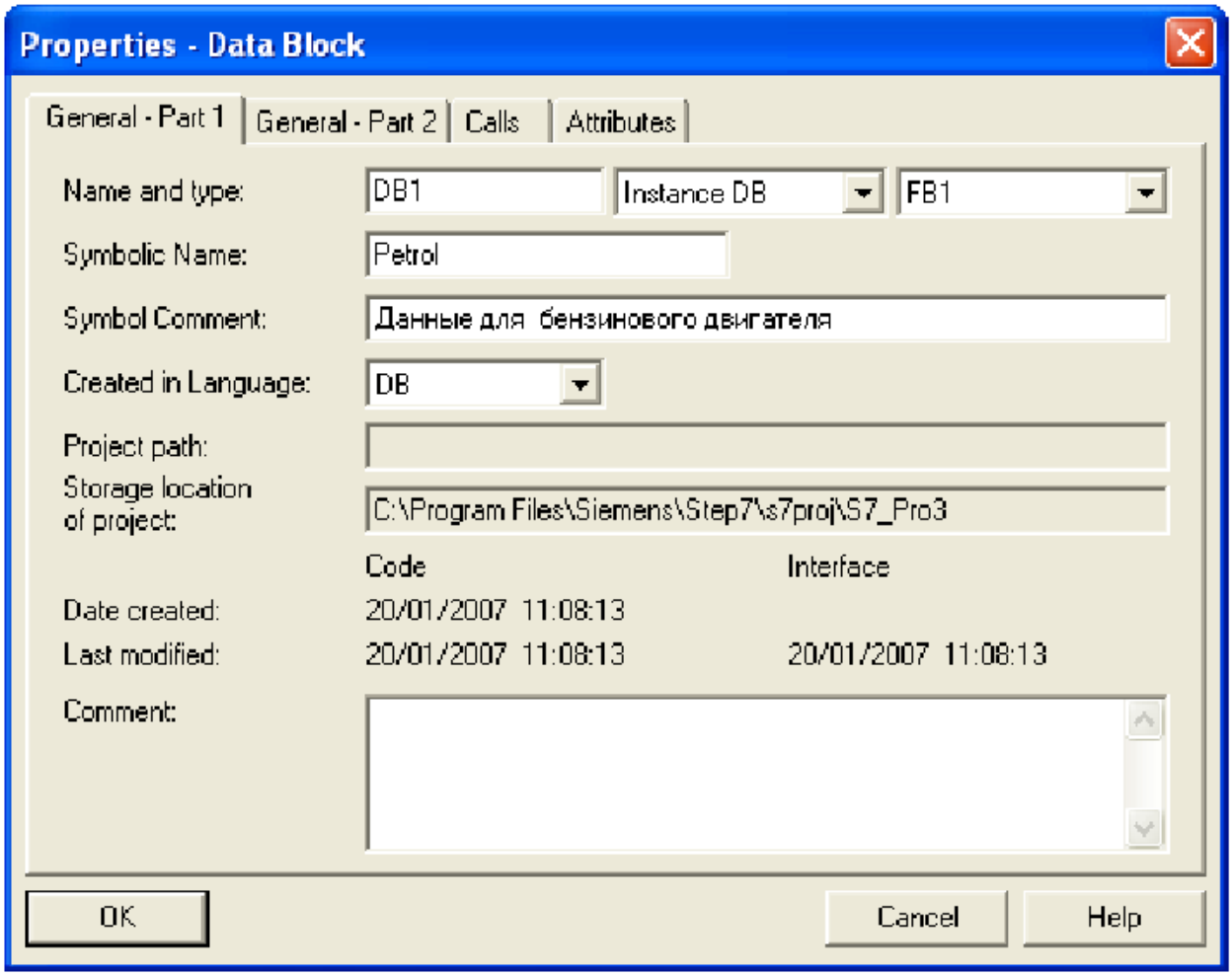


ნახ. 20 ბლოკ-ეკზემპლიარის დანიშვნა

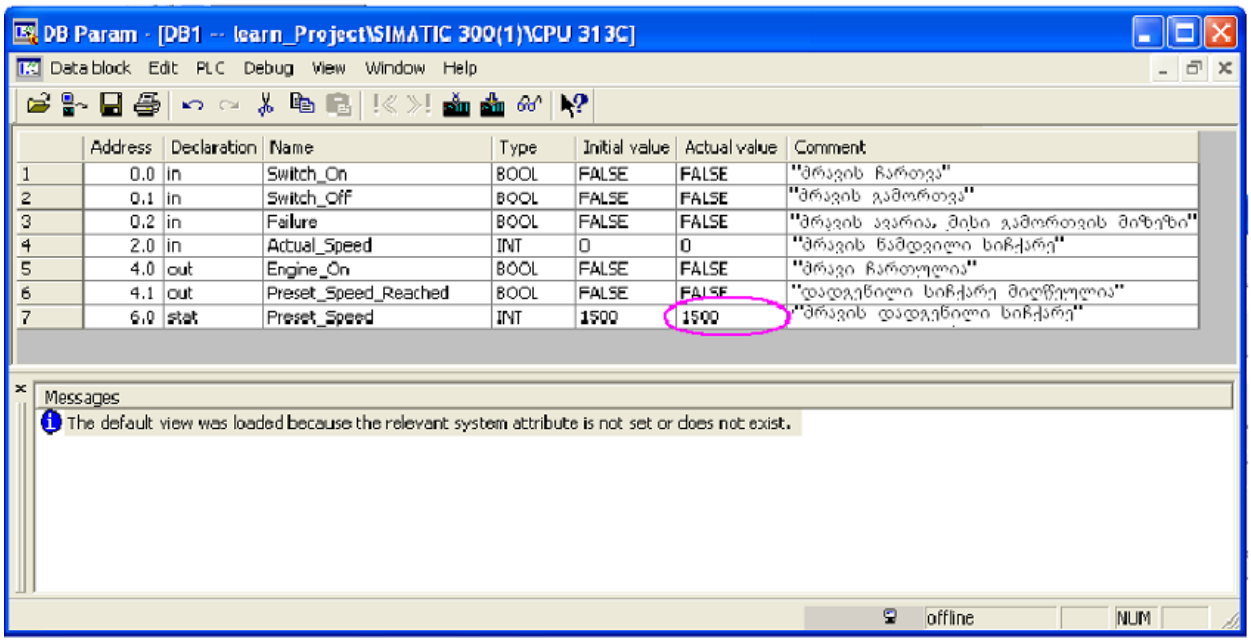
დაეთანხმეთ აწყობის ყველა პარამეტრს, რომლებიც აისახება დიალოგურ ფანჯარაში **Properties [თვისებები]** OK-ზე დაწკაპებით.

DB1 მონაცემთა ბლოკი ემატება პროექტს. ორჯერ დააწკაპეთ DB1 ნიშანზე, რომ გახსნათ იგი.

დაადასტურეთ FB1-ისთვის ბლოკ-ეკზემპლიარის დანიშვნა, **Engine [ძრავი]** OK –ს საშუალებით (ნახ. 20). შემდეგ შეიყვანეთ მნიშვნელობა „1500“ ბენზინის ძრავისთვის სვეტში Actual Value [ფაქტიური მნიშვნელობა] (სტრიქონში Preset_Speed) (ნახ. 22). ახლა თქვენ განსაზღვრეთ მაქსიმალური სიჩქარე ამ ძრავისათვის. შეინახეთ DB1 და დახურეთ ბლოკი. როგორც DB1-ისათვის, ასევე დააგენერირეთ კიდევ ერთი ბლოკი DB2, FB1-ისთვის. ახლა კი შეიყვანეთ მნიშვნელობა „1200“ ბენზინის ძრავისათვის.



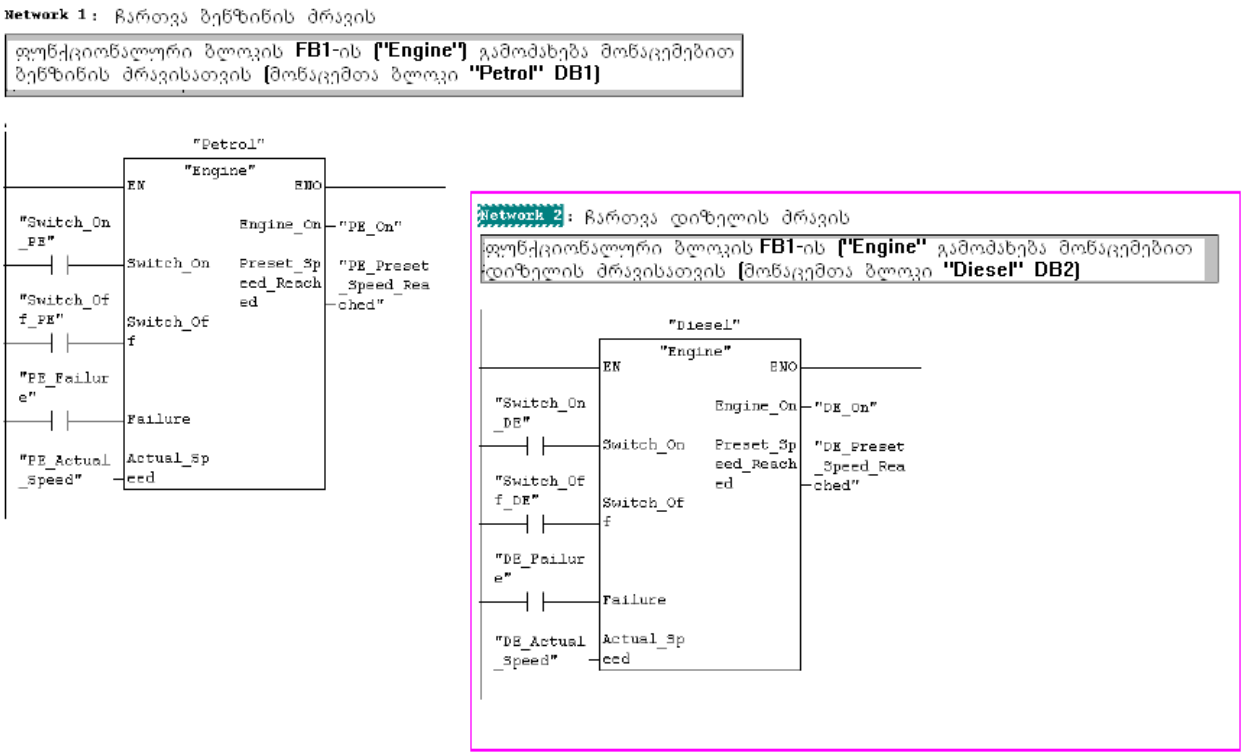
ნახ. 21 ბლოკის ტიპის არჩევა



ნახ. 22 ცვლადების დადგენა

ფაქტიური მნიშვნელობების ცვლილებით თქვენ დაამთავრეთ ორი ძრავის მართვისათვის მოსამზადებელი სამუშაოები მხოლოდ ერთი ფუნქციონალური ბლოკის დახმარებით. უფრო მეტი რაოდენობის ძრავების მართვისათვის ერთადერთი, რაც თქვენ უნდა გააკეთოთ ეს არის მონაცემთა დამატებითი ბლოკების გენერირება. შემდეგი ბიჯი, რაც თქვენ უნდა შეასრულოთ, ეს არის OB1 ფუნქციონალური ბლოკის გამოძახების დაპროგრამირება.

ორჯერ დააწკაპეთ OB1-ზე. შემდეგ გადაინაცვლეთ პროგრამის ელემენტების კატალოგში, ვიდრე არ მიაღწევთ ფუნქციონალური ბლოკების საქაღალდეს. გამოყავით FB1 და ჩასვით ეს ბლოკი OB1-ში. ჩასვით ნორმალურად ღია კონტაქტი თვითოეული შემდეგი შესასვლელების წინ: Switch_On [ჩართვა], Switch_Off [გამორთვა] და Fault [უწესრიგობა]. დააწკაპეთ ??? ნიშანზე, რომელიც მდებარეობს ბლოკის Engin [ძრავი] ზემოთ, ხოლო შემდეგ დატოვებთ რა კურსორს იმავე მდგომარეობაში, დააწკაპეთ თაგვის მარჯვენა ღილაკით შეყვანის ჩარჩოში. თაგვის მარცხენა ღილაკის გამოყენებით ამოცურებადი მენიუდან აირჩიეთ Insert Symbol [სიმბოლოს ჩასმა] ანდა დააჭირეთ **CTRL+SHIFT.** გამოჩნდება გადახვევადი სია. როდესაც ამას თქვენ პირველად აკეთებთ, ამ პროცედურამ შეიძლება დაიჭიროს რაღაც დრო. OB1 – გამოძახება DB1 - ის - ბენზინის ძრავის მონაცემები, DB2 – დიზელის ძრავის FB1 Engine მონაცემები. რეზულტატი ნაჩვენებია ნახ. 23-ზე.



ნახ. 23 მონაცემთა შეყვანის რეზულტატი

როდესაც თქვენ ქმნით პროგრამის სტრუქტურას ორგანიზაციული ბლოკების, ფუნქციონალური ბლოკების და მონაცემთა ბლოკების დახმარებით, თქვენ უნდა დააპროგრამიროთ გამოძახება დაქვემდებარებული ბლოკებისათვის (ისეთის, როგორც FB1-ია), იმ ბლოკში რომელიც იერარქიულად უფრო მაღლა დგას (მაგალითად, OB1). ეს პროცედურა ყოველთვის ერთნაირია. შეიძლება აგრეთვე მიენიჭოს სხვადასხვა ბლოკებს სიმბოლური სახელები სიმბოლოთა ცხრილში (მაგალითად, FB1-ს აქვს სახელი Engine [ძრავი], ხოლო DB1-ს სახელი Petrol [ბენზინის].

ჩატვირთეთ პროგრამა პლკ-ში განახორციელეთ პროგრამის ტესტირება, შეამოწმეთ ამუშავდებიან თუ არა ძრავები. თუ თქვენ ვერ იხსენებთ გადამრთველების მნიშვნელობებს (შესასვლელებსა და გამოსასვლელებს), მაშინ შეიძლება ნახოთ ისინი ცვლადების აღწერის ცხრილში.

3.7. პროგრამის ტესტირება

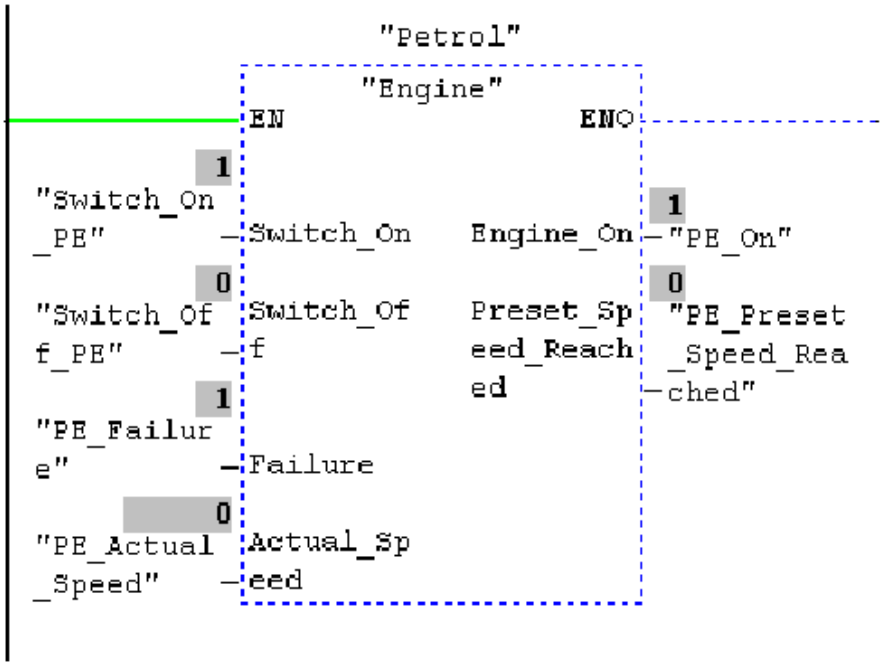
პროგრამის **სტატუსის** ფუნქციის გამოყენებით, თქვენ შეგიძლიათ დატესტოთ პროგრამა ბლოკში. ამის წინაპირობად საჭიროა CPU-სთან online კავშირის დამყარება. CPU უნდა იყოს RUN ანდა RUN-P რეჟიმში, ხოლო პროგრამა უნდა იყოს ჩატვირთული. გახსენით OB1 პროექტის ფანჯარაში **Getting Started OONLINE**. გაიხსნება პროგრამირების ფანჯარა LAD/STL/FBD. გააქტიურეთ ფუნქცია **Debag > Minitor [გაწყობა > დაკვირვება].** თქვენ დაინახავთ, „როგორ მიმოდიან სიგნალები“, მაგალითად ჩაკეტეთ გასაღები I 124.4 (ამ გასაღების ჩართვა შეესაბამება ბინზინის ძრავის გამართულ მუშაობას - ასეთი ფუნქციები ჩვეულებრივად ყოველთვის იყენებს ნორმალურად ღია კონტაქტს და მისი ჩაკეტვა უჩვენებს სისტემის გამართულობაზე, როგორც წესი ეს რამოდენიმე მიმდევრობით შეერთებული დაცვის კონტაქტია) და ჩართეთ კონტაქტი I 124.2 (ბენზინის ძრავის ჩართვა). შედეგად, თქვენ ნახავთ, რომ ზრავი ამუშავდა - ცვლადმა PE_On მიიღო ლოგიკური 1-ის მდგომარეობა (ნახ. 24).

OB1 : ციკლურად შესრულებადი მთავარი პროგრამა

Comment:

Network 1: ბენზინის ძრავის გაშვება

ფუნქციონალური ბლოკის **FB1**-ის **("Engine")** გამოძახება მონაცემებით ბენზინის ძრავისათვის **(**მონაცემთა ბლოკი **"Petrol" DB1)**



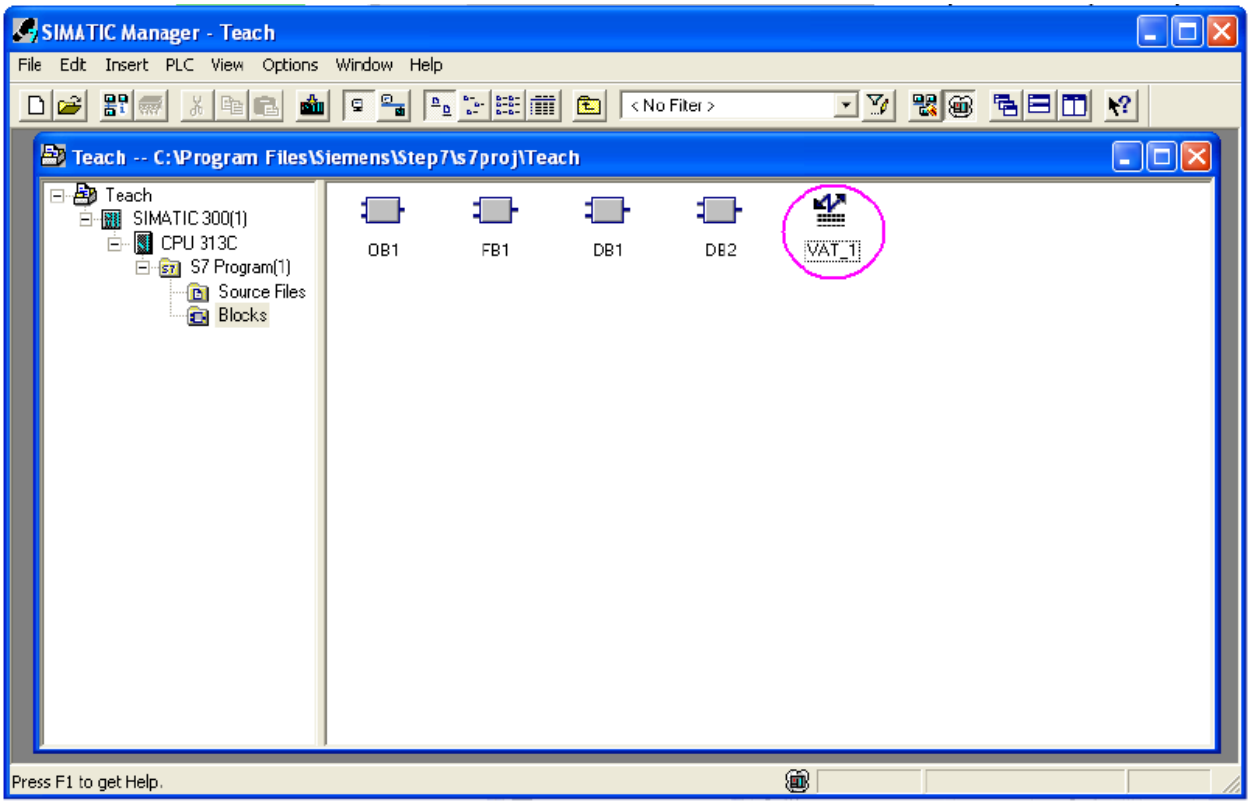
ნახ. 24 პროგრამა მუშაობაში

თქვენ შეგიძლიათ დატესტოთ პროგრამის ცალკეული ცვლადები მათი ცვლილებითა და თვალიერებით. ამის წინაპირობად საჭიროა CPU-სთან online კავშირის დამყარება. CPU უნდა იყოს RUN-P რეჟიმში, ხოლო პროგრამა უნდა იყოს ჩატვირთული. ისევე როგორც პროგრამის სტატუსის დახმარებით ტესტირებისას, თქვენ შეგიძლიათ ასახოთ შესასვლელები და გამოსასვლელები ცვლადების ცხრილებში. თქვენ შეგიძლიათ აგრეთვე დატესტოთ შედარების ბლოკი FB1-ში ძრავის სიჩქარისათვის, ფაქტიური სიჩქარის წინასწარი მინიჭების გზით.

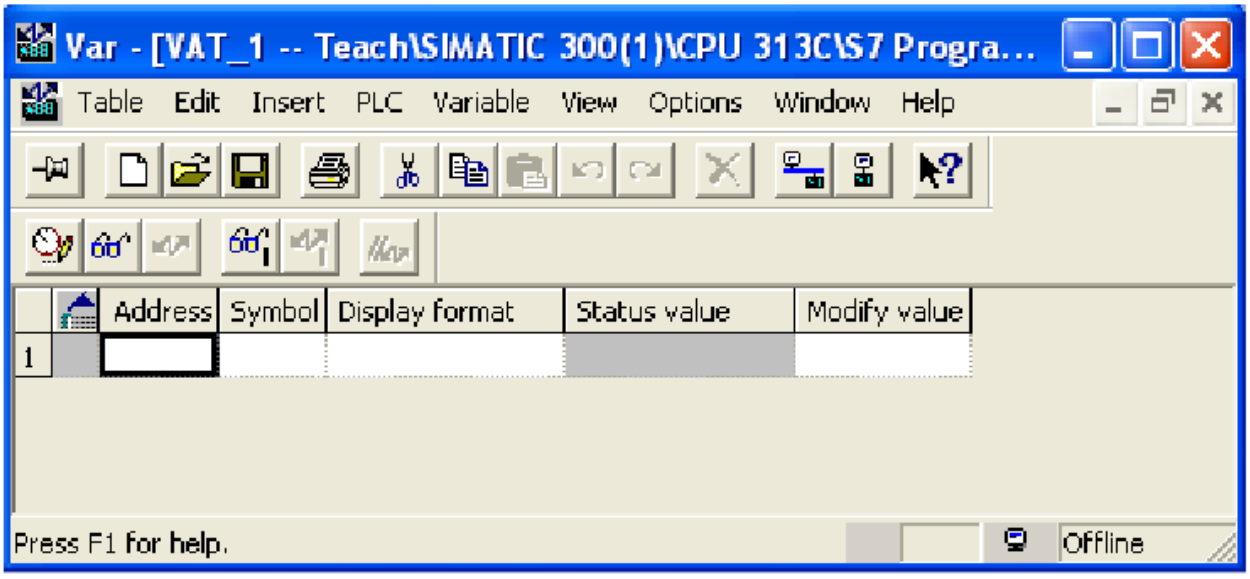
გადაინაცვლეთ საქაღალდეში Blocks [ბლოკები] და დააწკაპეთ თაგვის მარჯვენა ღილაკით ფანჯრის მარჯვენა ნახევარში. გამოიყენეთ თაგვის მარჯვენა

ღილაკი, რომ ჩავსვათ Variable Table [ცვლადების ცხრილი] ამოცურებადი მენიუდან. დაეთანხმეთ სიჩუმით მოცემულ აწყობის პარამეტრებს, დახურეთ დიალოგიური ფანჯარა **Properties [თვისებები]** OK-ზე დაწკაპებით.

VAT1 ცვლადების ცხრილი იქმნება ბლოკების საქაღალდეში (ნახ. 25). ორჯერ დააწკაპეთ VAT1-ზე, რომ გახსნათ ცხრილი; გაიხსნება ფანჯარა Monitoring and Moidifyng Variables [ცვლადების დათვალიერება და ცვლილება] (ნახ. 26).

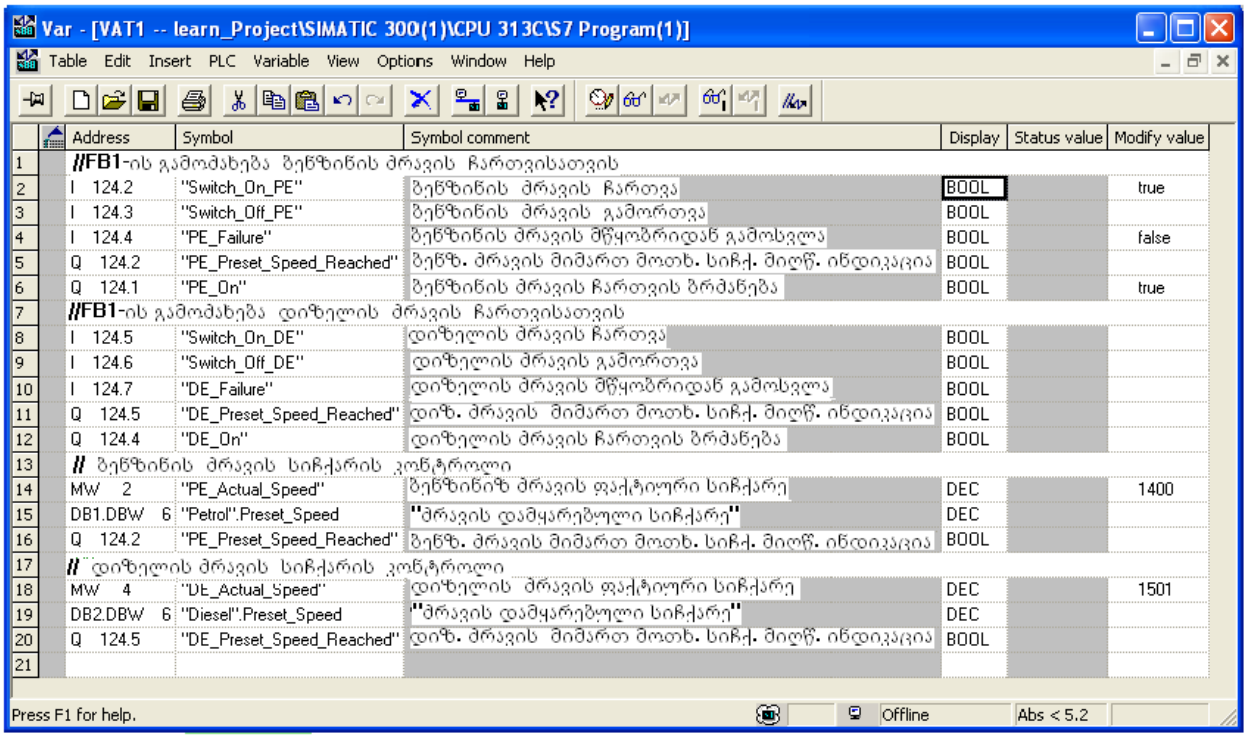


ნახ. 25 ცვლადების ცხრილის შექმნა



ნახ. 26 ცვლადების ცხრილის შექმნა

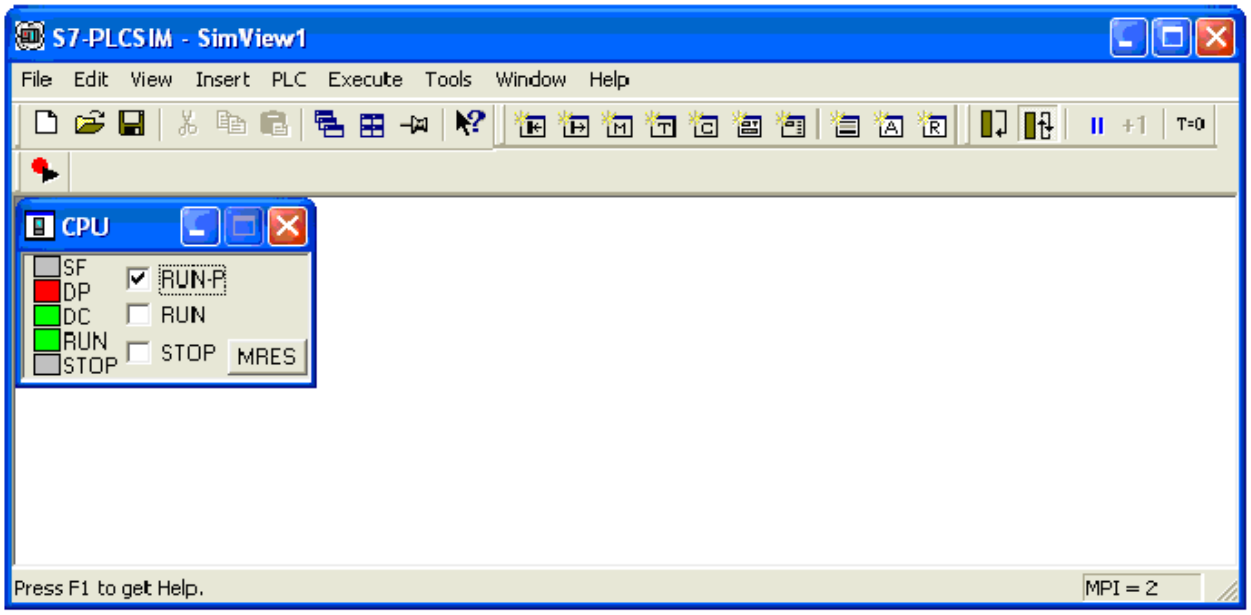
დასაწყისში ცვლადების ცხრილი ცარიელია. შეიტანეთ სიმბოლური სახელები ანდა მისამართები, როგორც ნაჩვენებია ნახ. 27-ზე.



ნახ. 27 ცვლადების მონიტორინგი

დანარჩენი ელემენტი იქნება დამატებულნი, როდესაც თქვენ დაამთავრებთ შეყვანას Enter-ზე დაჭერით. შეცვალეთ შემოწმების ფორმატი (Monitor Format) სიჩქარის ყველა მნიშვნელობისათვის DEC (ათობითი) ფორმატით. ამისათვის დააწკაპეთ სათაურში შესაბამის უჯრედზე (კურსორი გარდაისახება ისარად Monitor format -ის სვეტის ზემოთ) და თაგვის მარჯვენა ღილაკის გამოყენებით, აირჩიეთ DEC ფორმატი.

შეინახეთ თქვენი ცხრილი ცვლადებისა .



ნახ. 28 პროგრამის გამწყობის სახე

დააწკაპეთ ღილაკზე Monitor Variables ინსტრუმენტების პანელზე. დააჭირეთ გასაღებებს 1-ზე და 2-ზე თქვენს ტესტურ კონფიგურაციაზე და ყურადღება მიაქციეთ შედეგს ცვლადების ცხრილში (თუ კი კონტროლერი ჩართულია რათქმაუნდა). მდგომარეობების მნიშვნელობები ცვლადების ცხრილში იცვლება false - დან [მცდარი] tru- ზე {ჭეშმარიტი].

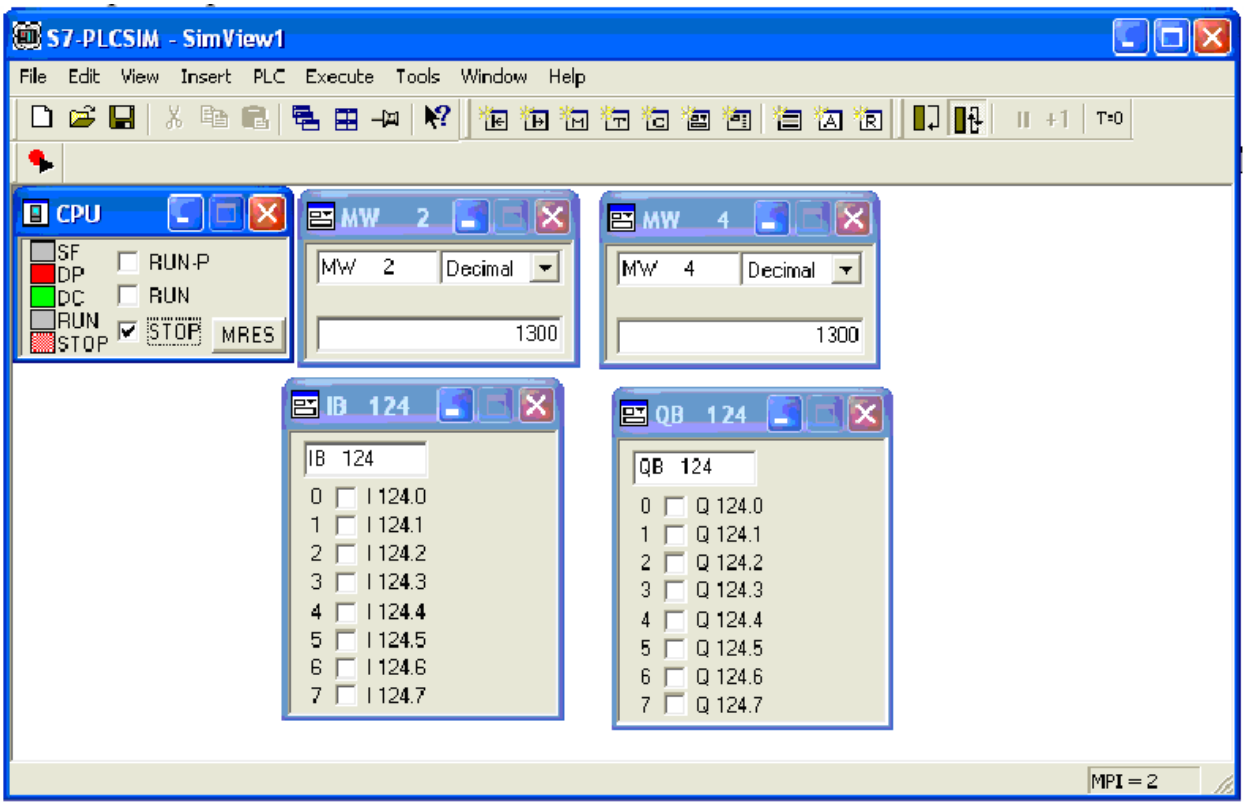
შეიტანეთ მნიშვნელობა „1500“ მისამართისთვის MW2 სვეტში Modify Value [მნიშვნელობის ცვლილება] და „1300“ სვეტში MW4. გადაეცით მნიშვნელობების ცვლილება თქვენს CPU-ში .

3.8. პროგრამის გაწყობა კონტროლერის გარეშე

პროგრამირების პროცესში პროგრამის გაწყობისათვის სრულიადასც არაა აუცილებელი კონტროლერის ქონა, პროგრამა შეიძლება გაწყობილ იყოს მის გარეშეც. პროგრამის გაწყობისათვის შეიძლება გამოყენებულ იქნას Step-7-ში არსებული სიმულიატორი.

სიმულიატორის გამოძახებისათვის Simatic Manager-ის მთავა ფანჯარაში საჭიროა Options->Simulate Modules - ის არჩევა, ამასთან სტენდის კონტროლერი უნდა იყოს გამორთული. გამოჩნდება ფანჯარა, რომელიც ნაჩვენებია ნახ. 28-ზე.

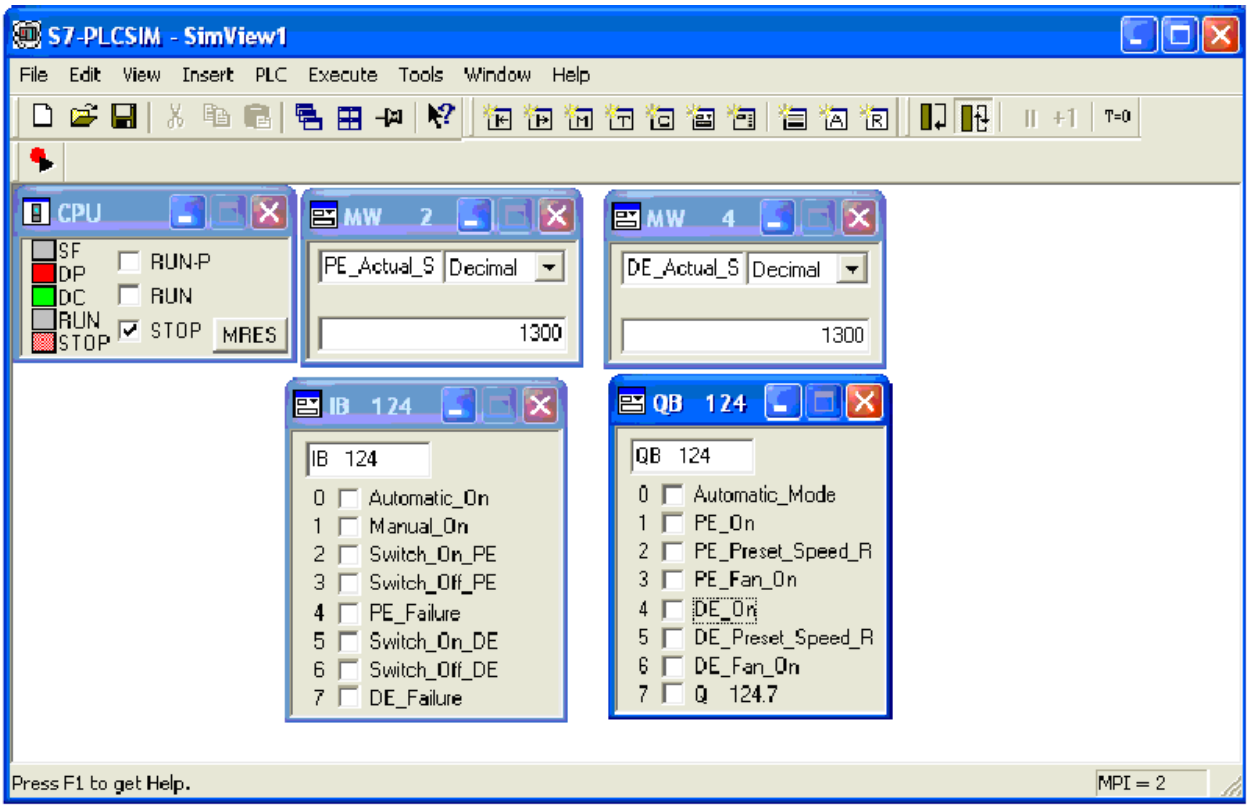
Insert-ის მენიუს გამოყენებით, ჩასვით Inputs Variable (ან დააჭირეთ F2-ს). ჩასვით კიდევ ერთი Inputs Variable და ჩასვით ორი ფანჯარა Vertical Bits. შესაბამის ველებში შეიტანეთ მნიშვნელობები, რომლებიც ნაჩვენებია ნახ. 29-ზე. ჩატვირთეთ პროგრამა სიმულიატორში, ზუსტად ისევე, როგორც თქვენ ჩატვირთეთ იგი რეალურ კონტროლერში.



ნახ. 29 პროგრამის გაწყობა

გაუშვით სიმულიატორი, ჩასვავთ რა გალოჩკას CPU-ს ფანჯრის RUN ან RUN-P ველში. ჩასვავთ რა გალოჩკას შესაბამის მინდვრებში IB (Input Bits), თქვენ შეგეძლებათ კონტროლერის შესასვლელი სიგნალების სიმულირება და გამოსასვლელი სიგნალების ნახვა QB-ში. უფრო მეტი მოხერხებულობისათვის

შეცვალეთ ნაკლებინფორმატიული აბსოლუტური მისამართები ფსევდონიმებით. ამისათვის მენიუში აირჩიეთ Tools->Options->Attach Symbols და აირჩიეთ შექმნილი პროექტის ცვლადების ცხრილი. აუცილებლობის შემთხვევაში მენიუს ამავე ქვეპუნქტში ჩასვით გალოჩკა Show Symbol-ში. რეზულტატი ნაჩვენებია ნახ. 30-ზე. სიმულიატორში შეიძლება შეიცვალოს კონტროლერის გამოსასვლელები შესასვლელების ცვლილების გარეშე, რაც შეუძლებელია რეალურ სისტემაში, მაგრამ ზოგზერ ეს აუცილებელია გაწყობის დროს. გაწყობის ყველა ხერხი, რომლებიც აღწერილ იყო ზემოთ, შეიძლება გამოყენებულ იქნას იმიტატორის გამოყენების დროს.



ნახ. 30 მინაწერების დამატება ცვლადებზე

ამით მეთოდური მითითებები მთავრდება. ლაბორატორიულში პრაქტიკული დავალების შესრულების დროს თქვენ აუცილებლად დაგჭირდებათ ცნობარი პროგრამირების შესახებ ანდა დოკუმენტაცია Step-7-ის შესახებ ქართულ ენაზე, რადგან ინფორმაციის მთელი მოცულობის მოცვა შეუძლებელია.

3.9. ანგარიშის წარდგენის ფორმა

წერილობითი ანგარიში ლაბორატორიული სამუშაოს შესახებ არ კეთდება. ლაბორატორიული სამუშაოს დაცვა მდგომარეობს ინდივიდუალური პრაქტიკული სამუშაოს შესრულებაში, რომელიც პლკ-ს გადაეცემა ხელმძღვანელს დაპროგრამირების ხელმძღვანელს.

ლიტერატურა

1. Анхимов В. Л., Опейко О. Ф., Михеев Н. Н. Теория автоматического управления. – М.: Дизайн ПРО, 2002.

2. Ганс Бергер Автоматизация посредством STEP 7 с использованием STL и SCL и программируемых контроллеров SIMATIC S7-300/400.

3. Шишмарев В. Ю. Автоматика: Учебник для сред. проф. образования –М.: Издательский центр «Академия», 2005.

4. Романов В. П. Основы языка программирования STEP7 и базового программного обеспечения промышленных контроллеров SIEMENS. Учебно-методическое пособие, Новокуз-нецк 2009.

5. Романов В. П. Сопровождение, диагностика и техническое обслуживание автоматизированных систем управления технологическими процессами на основе программируемых логических контроллеров S7 фирмы Siemens. Методическое пособие, Новокузнецк 2010.

6. Лазарев, Ю. Моделирование процессов и систем в MATLAB. [Текст]: Учебный курс. / Ю. Лазарев. - Спб.: Питер; Киев: Издательская группа BHV, 2005, 512с.;

7. Методы классической и современной теории автоматического управления: Учебник в 3-х т. - Т.3: Методы современной теории автоматического управления / Под ред. Н.Д. Егупова. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2000.-748 с.;

8. Методы классической и современной теории автоматического управления [Текст]. В 5-и т. Т.

9. Статистическая динамика и идентификация систем автоматического управления / Под ред. К.А. Пупкова, Н.Д. Егупова. – 2-е изд., перераб. и доп. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. - 640 с.;

10. Ганс, Бергер. Автоматизация с помощью Программ STEP7 LAD и FBD Программируемые контроллеры SIMATIC S7-300/400 Заказной номер: 6ES7810-4CA05-8AR0. Издание 2-е переработанное, 2001;

11. SIMATIC HMI. ProTool: Конфигурирование систем на базе Windows. Руководство пользователя. Заказной номер 6AV6594-1MA05-2AB0. Версия от 12/99;