

Лабораторно упражнение

Програмируеми логически контролери (PLC). Ладер диаграми. Програмиране на PLC SIMATIC S7 1200

I. Цел на упражнението

Студентите да се запознаят с:

- ладер диаграмите като основен език за програмиране на PLC;
- програмируемия логически контролер SIMATIC S7 1200;
- средата за програмиране TIA PORTAL V13;
- технологията на програмиране при SIMATIC S7 1200.

II. Опитна постановка

Опитната постановка включва:

- програмируем логически контролер SIMATIC S7 1200, CPU 1214C/DC/DC/DC;
- платка с ключета, свързани към цифровите входове, имитиращи състоянието на реални технологични контакти;
- работна станция (персонален компютър);
- интерфейсен кабел за комуникация между PLC и работната станция.

III. Теоретична обосновка

1. Ладер диаграми

Ладер диаграмите са графичен език за програмиране на PLC, базиран на електрическите релейно-контактни схеми.

“Ладер програмата” се състои от определена последователност от логически “контактни” инструкции, чрез които се идентифицира състоянието на всеки един от елементите (контактите) на електромеханичната система, използвани за управлението на машината (процеса). Действителното състояние на контактите от електромеханичната система се заменя с логическото.

В релейно-контактните системи, управлението на всеки изпълнителен механизъм се извършва посредством отделна електрическа верига, завършваща с “бобина”. Контактите на всяко от устройствата участващи в управлението на бобината се свързват в съответствие с логическата функция на управлението. Когато състоянията на отделните контакти позволяват протичането на ток през бобината, изпълнителният механизъм се задейства.

Началото на всяко “стъпало”- логическо уравнение в Ладер диаграмата, започва с условна “линия на захранването”. Прието е състоянията на отделните “контакти” в логическото уравнение да се нарича “тестова зона”, а участващите инструкции “входни инструкции”. Последният елемент на всяко стъпало е условната “бобина”(изходна инструкция), която се активира в резултат на текущото изчисление на логическата комбинация от състоянието на входовете.

2. Програмируем логически контролер SIMATIC S7 1200, CPU 1214C/DC/DC/DC

Програмируемият логически контролер SIMATIC S7 1200 в конструктивно отношение е от тип “компактно устройство”, а различните му модификации се различават по: обема на паметта, предназначена за потребителската програма, типа на захранващото напрежение, типа на цифровите изходи, броя на разширителните модули

и др. CPU разполага с интегриран етернет интерфейс (PROFINET) за програмиране и връзка с други PLC и модули.

Програмируемият логически контролер SIMATIC S7 1200, CPU 1214C/DC/DC/DC има 14 цифрови входа, 10 цифрови изхода и два АЦП. Допълнително е добавен модул ЦАП.

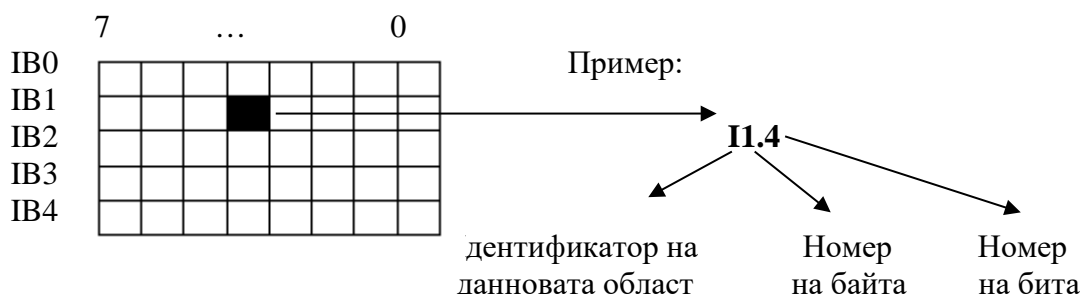
Цифровите входове са разделени на два байта и са означени съответно с **DIa.0** до **DIa.7**, и **DIb.0** до **DIb.5**. На всеки вход съответства светодиода, който свети, когато на съответния вход се подава логическа 1. Към входовете от **DIa.0** до **DIa.7** са свързани 8 ключета, с които се симулира затваряне и отваряне на контакти (нормално отворени или нормално затворени).

Цифровите изходи са разделени на два байта и са означени съответно с **DOa.0** до **DOa.7**, и **DOb.0**, **DOb.1**. На всеки изход съответства светодиода, който свети, когато на съответния изход се получава логическа 1.

Обменът на информация между различните задачи и структурни единици на потребителската програма се извършва посредством даннови области, поддържани от операционната система. В лабораторното упражнение ще бъдат разгледани следните даннови области:

- *образ на входния процес* -> В тази област операционната система записва състоянието на входните точки. Идентифицира се с "I".
- *образ на изходния процес* -> От тази област операционната система копира състоянията на изходните битове в изходните точки. Идентифицира се с "Q".
- *вътрешни релета* -> Тази област е предназначена за запазване състоянията на "оперативни релета", имащи отношение към управлението на обекта, но които нямат отношение към конкретен изход. Идентифицира се с "M".

Програмируемият логически контролер е с байтова организация на оперативната памет. При адресиране на битови операнди се посочва номерът на байта от съответната даннова област и номерът на бита от съответния байт.



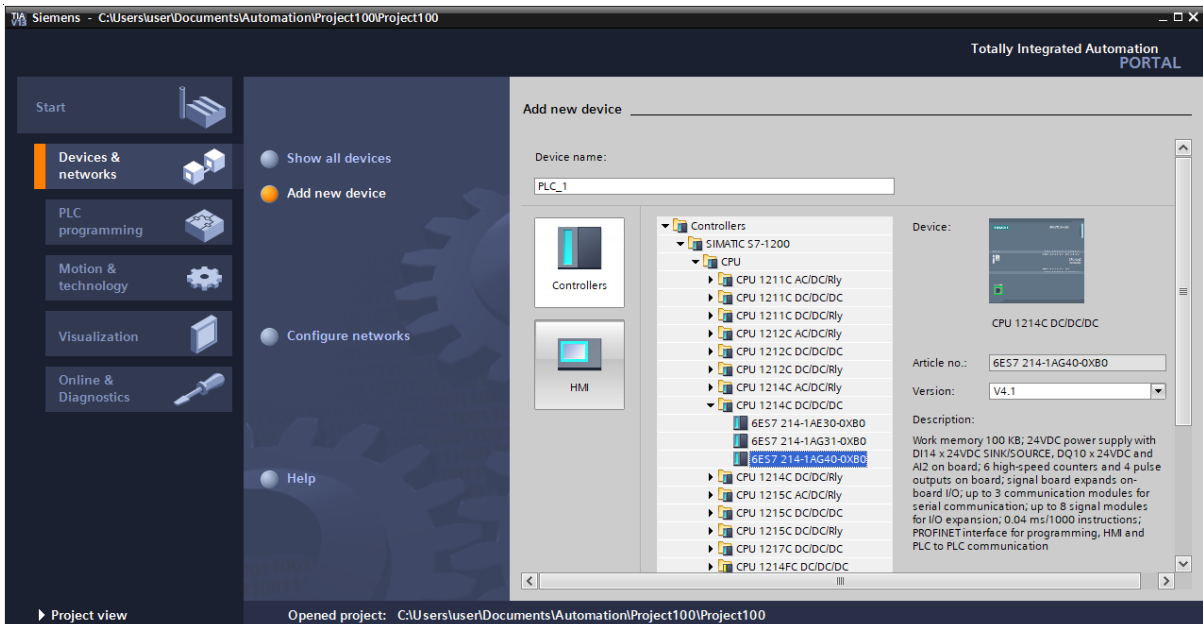
При адресиране на байтови операнди се посочва идентификаторът на областта, размерността на операнда – Byte (B) и неговият номер. Например IV3. Адресирането на операнди с по-голяма размерност (W – дума или D- двойна дума) обхваща област от няколко байта. Кой байтове са включени в адресирания операнд, зависи от посочения стартов адрес. Например в IW3 са включени 3 и 4 байт от адресируемата I даннова област.

3. Средата за програмиране на PLC SIMATIC S7 1200

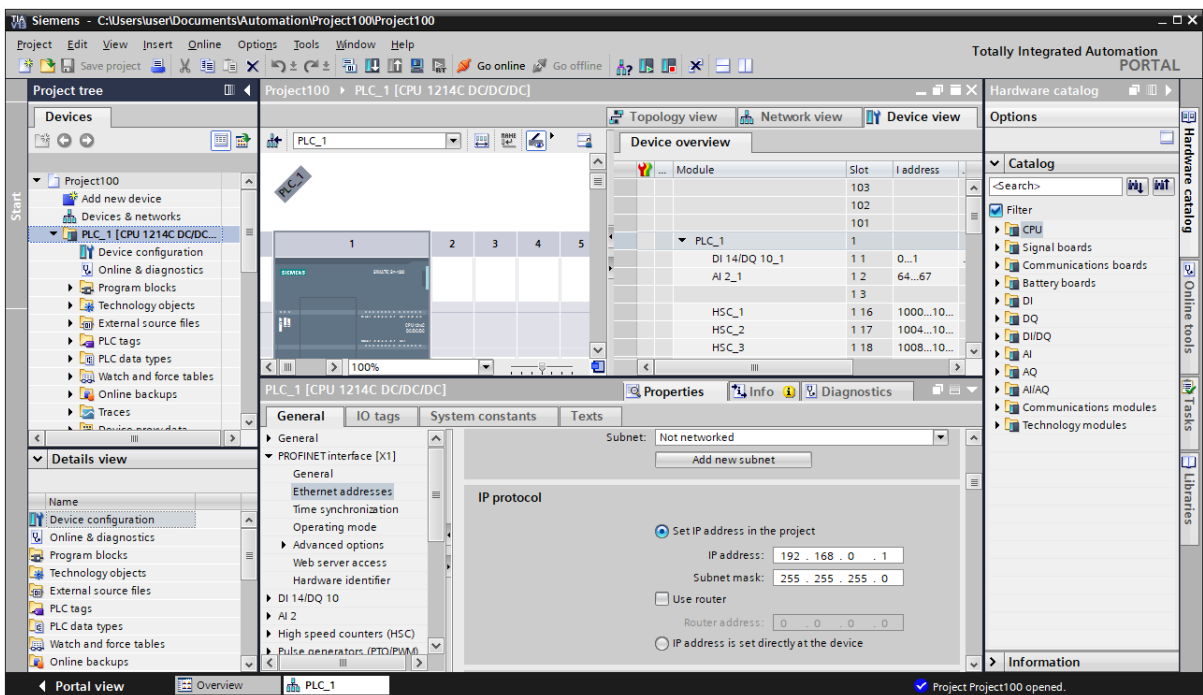
Програмирането на контролера SIMATIC S7 1200 се извършва с помощта на специализирана програма TIA PORTAL V13, работеща под WINDOWS.


IV. Задачи за изпълнение

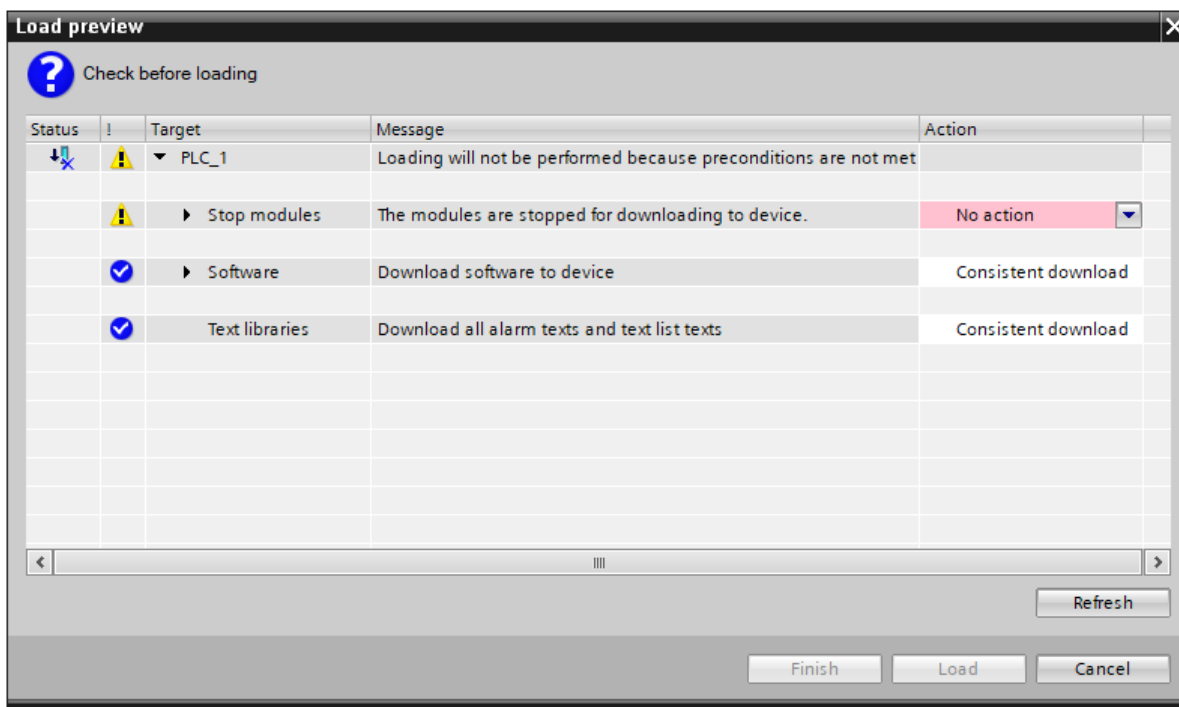
1. Да се реализира проект за програмиране на контролера SIMATIC S7 1200.
За тази цел да се извърши следното:
 - стартира се програмата TIA PORTAL V13;
 - от менюто се избира "create new project" със съответното име;
 - избира се "Configure a device";
 - избира се "Add new device";



- задава се име на контролера – например PLC_1;
- избира се тип на контролера - CPU 1214C/DC/DC/DC 6ES7 214-1AG40-0XB0;
- на екрана се визуализира проекта с избрания контролер;



- от менюто "Ethernet addresses" трябва да се зададе съответен IP адрес (от 10.33.0.11 до 10.33.0.15) от вътрешната мрежа на лабораторията и адреса на рутера 10.33.0.1;
- така създадения проект се зарежда в контролера – с мишката се маркира полето PLC_1(CPU 1214C) от дървото на проекта и проектът се зарежда с помощта на бутон  (Download to device) от лентата с инструменти;
- стартира се търсенето на съответни устройства със "Start search";
- конфигурацията на проекта се компилира и преди да се зареди се избира опцията "Stop all" от следния прозорец и се зарежда конфигурацията в контролера.





2. Да се разучи и експериментира работата на контактните инструкции.




Контактните инструкции описват състоянието на отделни "контакти"- битове от съответната потребителска област и се кодират директно в "Ладер диаграмата". В качеството на отделен "контакт" от системата за управление могат да се използват всички битове от адресируемите даннови области на PLC.


- Нормално отворен контакт -> затваря (включва) се, когато стойността на адресирания бит е 1 (табл.1).
- Нормално затворен контакт -> затворен е когато стойността на адресирания бит е 0.

Табл.1

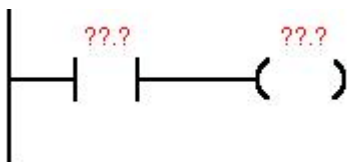
Наименование на контактната инструкция	Ладер символ
Нормално отворен контакт	
Нормално затворен контакт	

2.1. Да се експериментира работата на нормално отворения контакт

За тази цел от дървото на проекта, намиращо се в лявата част на екрана, се отваря папката с название Program blocks и се стартира  Main [OB1]. От полето Basic instruction, намиращо се в дясната част на екрана, се отваря папката Bit logic operations. Маркира се символът на нормално отворения контакт . С помощта на мишката символът се предвижва в полето  Main [OB1] за въвеждане на Ладер уравненията.

Както се забелязва в началото на това поле има плътна вертикална линия, която всъщност представлява линията на захранването (началото на всяко стъпало в Ладер диаграмата). Зоната в дясно от линията на захранването се нарича “тестова зона”, а участващите инструкции “входни инструкции”. Последният елемент на всяко Ладер уравнение е условната “бобина”. Ладер символът на бобината  също се намира в папката Bit logic operations. Маркира се символът на бобината и с помощта на мишката се предвижва вдясно от символа на нормално отворения контакт.

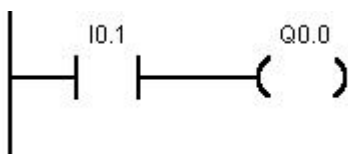
Ладер уравнението, което е въведено до тук, трябва да има вида показан на фиг.1.



Фиг.1


Следва да зададем битовете от паметта, в които се записва състоянието на входните и изходните точки. За целта е необходимо да се кликне с мишката в полето ??? , намиращо се над самите Ладер символи и да се въведат адресите на входната и изходната точка (начинът на задаване е описан в Теоретичната обосновка т.III.2). На фиг.2 е даден конкретен пример за задаване на адресируемите данни области.

Network 1:



Фиг.2

С десния бутон на мишката може да се изтриват инструкции, да се вмъкват или изтриват редове и колони в даден Network.

Следва да се зареди така написаната програма в паметта на PLC. Това става с помощта на бутон  от лентата с инструменти. Забележете, че по време на запис на потребителската програма в паметта на PLC в долната част на екрана, компилаторът извежда съобщение за протичане на процеса на компилация и зареждане на програмата.

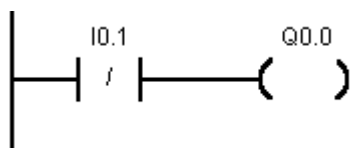
Проверката за работата на програмата се осъществява чрез превключване на ключето S1 от платката с ключетата. Информация за състоянието на входовете и изходите на PLC можем да получим директно от светодиодите на лицевата му страна.

Като се използва бутона  от лентите с инструменти, да се наблюдава работата на програмата в реално време.

2.2. Да се експериментира работата на нормално затворения контакт.



Следвайте описанието на работа дадено в т.2.1. Ладер уравнението, което трябва да напишете има следния вид:

Network 1:

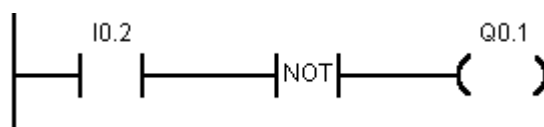


Заредете новата потребителска програма в PLC и наблюдавайте разликите спрямо нормално отворения ключ в реално време.

3. Логическа инверсия “NOT”

Използва се за извеждане на инверсната стойност на изход. Ладер символът на логическата инверсия ( NOT ) намира също в папката Bit logic operations от дървото с инструкциите. На фиг.3 е даден пример за използването ѝ.

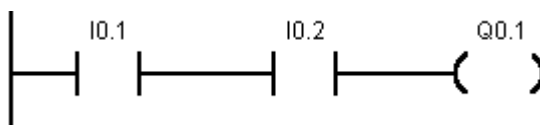
Network 1:



Фиг. 3

4. Запознаване с операцията логическо “И”

Графичното изображение на операцията логическо “И” със средствата на релейно-контактните схеми представлява съвкупност от последователно свързани нормално отворени контакти. Бобината ще бъде активна (през нея ще протича ток) само ако нормално отворените контакти са включени. Примерно ладер уравнение с операцията логическо “И” е показано на фиг.4.

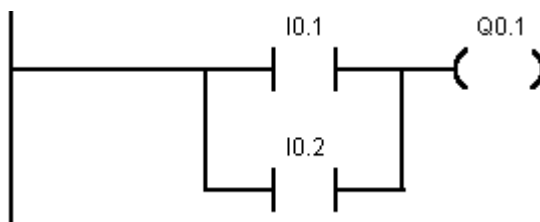


Фиг. 4



Следвайте описанието дадено в т.2.1, постройте ладер уравнението с операцията лог. “И”, заредете програмата и експериментирайте нейната работа.

5. Запознаване с операцията логическо “ИЛИ”

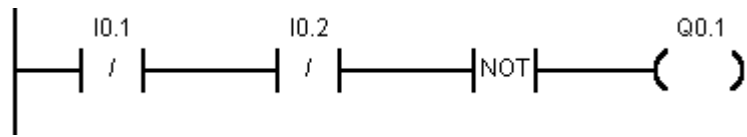
Графичното изображение на операцията логическо “ИЛИ” със средствата на релейно-контактните схеми представлява съвкупност от паралелно свързани нормално отворени контакти. Бобината ще бъде активна (през нея ще протича ток), ако поне един (или двата) от нормално отворените контакти е включен(и). Примерно ладер уравнение с операцията логическо “ИЛИ” е показано на фиг.5.



Фиг. 5

Използвайте бутоните  и  от лентата с инструменти за да прекарате вертикалните връзки между контактите.

Операцията логическо “ИЛИ” може да бъде представена като съвкупност от последователно свързани нормално затворени контакти и логическа инверсия, използвайки законите на Булевата алгебра (фиг.6).



Фиг. 6

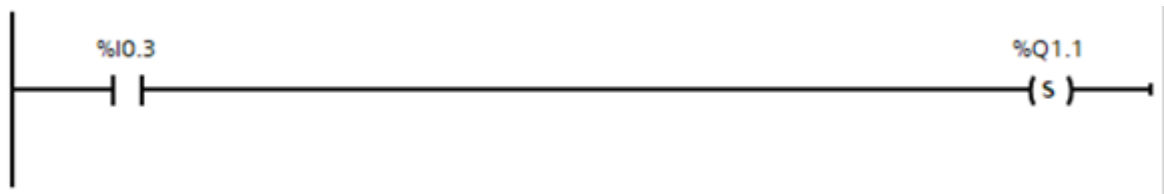
Следвайте описанието дадено в т.2.1, постройте ладер уравнението с операцията лог. “ИЛИ”, заредете програмата и експериментирайте нейната работа.

6. Запознаване с изходни инструкции за нулиране и установяване на даден изход.

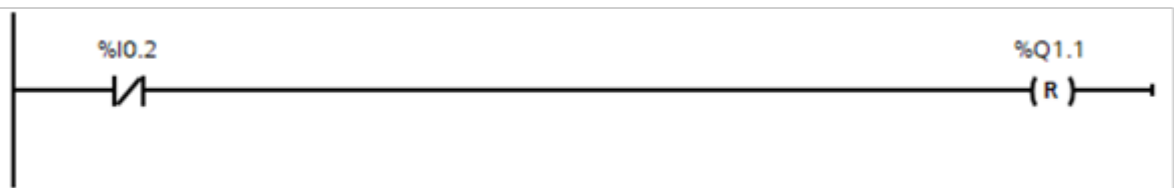
6.1.Нулиране на даден изход.

Заредете следната програма и обяснете нейната работа.

Network 1



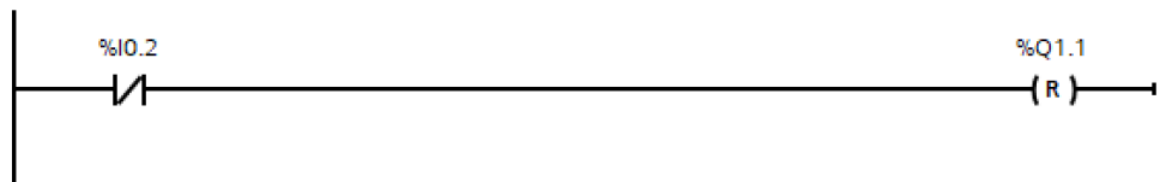
Network 2



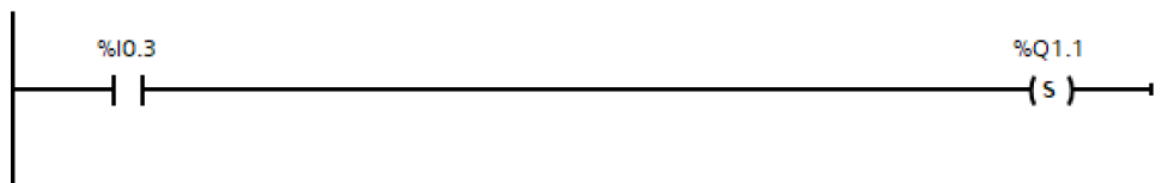
6.2.Установяване на даден изход.

Заредете следната програма и обяснете нейната работа.

Network 1

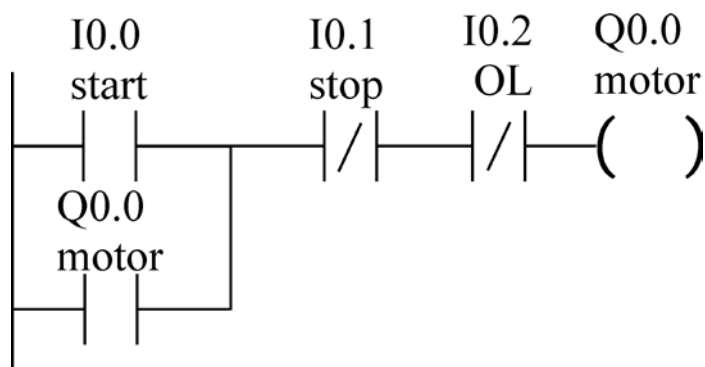


Network 2



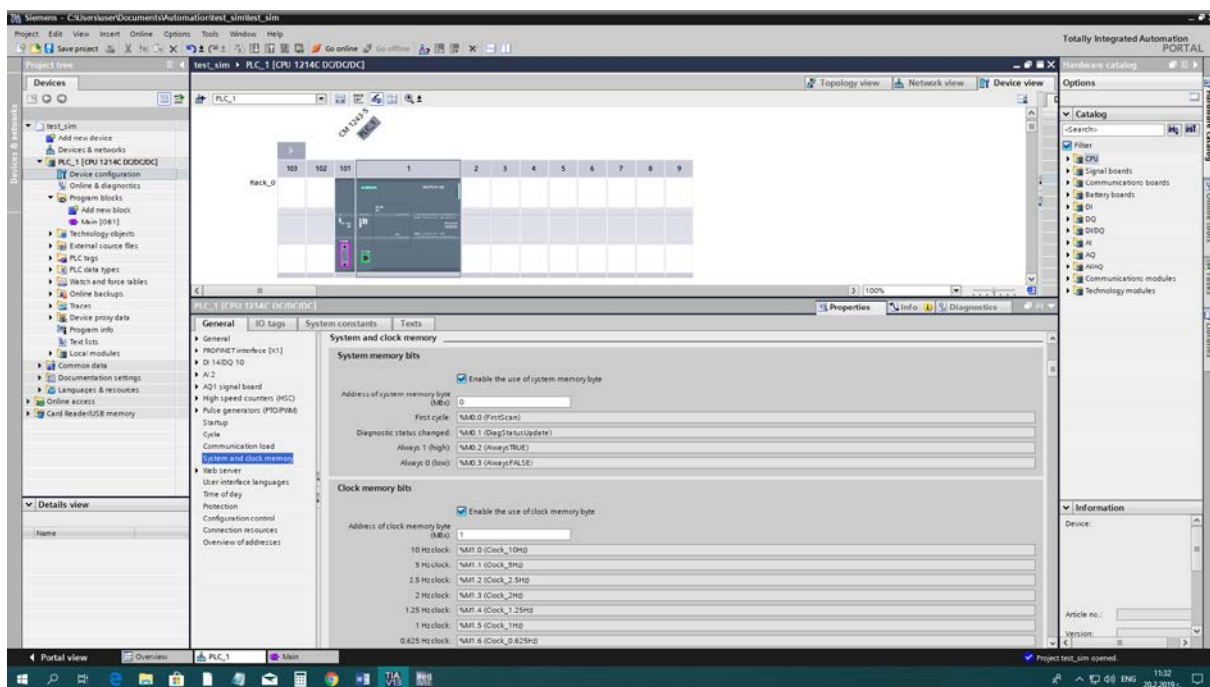
7. Заредете следната програма и обяснете нейната работа:

Network 1



С помощта на бутоните, свързани към входовете на контролера се проверява работоспособността на схемата.

8. От менюто Device configuration/Properties/System and clock memory да се активират system memory byte и clock memory byte.



9. Като се знае, че системен бит M1.5 реализира импулсна поредица с период 1 sec., то:

9.1. В Network 1 да се напише програма за включване/изключване на изход Q0.4 на всяка секунда при натиснат бутон I0.2.

9.2. В Network 2 да се напише аналогична програма за изход Q1.0 при ненатиснат бутон I0.3.

9.3. В Network 3 да се напише програма за логическото уравнение:
 $Q0.0 = Q0.4 \& Q1.0$.

9.4. Компилираната програма да се зареди в контролера и да се провери работоспособността ѝ с помощта на бутоните.