
UNIVERSITETI I PRISHTINËS
“HASAN PRISHTINA”
FAKULTETI I INXHINIERISË MEKANIKE
PRISHTINË



PUNIM DIPLOME
(Master)

Mentori:
Prof. Dr. Shaban Shabani

Kandidati:
Bsc.Besim Ferataj

Prishtinë, 2016

UNIVERSITETI I PRISHTINËS
“HASAN PRISHTINA”
FAKULTETI I INXHINIERISË MEKANIKE
PRISHTINË



PUNIM DIPLOME
(Master)

Tema:

**Dirigjimi i procesit të prodhimit të tullave me ndihmën e PLC-së
STEP 7 në fabrikën e bllokave EL-Boksh**

**The conduction of bricks production using the PLC STEP-7 in the bricks
factory "EL-Boksh"**

Mentori:
Prof. Dr. Shaban Shabani

Kandidati:
Bsc. Besim Ferataj

Prishtinë, 2016

PËRMBAJTJA

HYRJA	5
1.0 AUTOMATIZIMI I PROCESIVE INDUSTRIALE	6
1.1Automatet e Fundëm.....	9
1.2SistemetNumerike	12
1.3Përdorimi i sistemeve tjera numerike dhe komandimi i tyre	12
2.0 SIMBOLET DHE VARIABLAT PËR PROGRAMIMIN E PLC-së	18
2.1 Urdhërat e përdorur për programim	18
2.2 Simbolet	19
2.2.1. Simboli “&” (dhe, AND).....	19
2.2.2. Simboli barazim (Assing)	20
2.2.3. Operacioni logjik OSE (≥ 1)	20
2.2.4. Vija e mesme dalëse “ # ” (Midline Output)	21
2.2.5. Detektimi i RLO-së pozitive (P).....	21
2.2.6. Starto prodhimin (vendos daljen) S Set Output	21
2.2.7. Ristarto prodhimin (rivendos daljen) R Reset Output	22
2.2.8. Starto_Ristarto Flip Flop (vendos_rivendos) SR Set_reset Flip Flop	22
2.2.9. Ristarto_Starto Flip Flop (rivendos_vendos) RS Reset_set Flip Flop	23
2.2.10. Krahasimi i dy vlerave	23
2.2.11. Lëvizja (Move): Transferimi i vlerave	24
2.2.12 Kalimi i pa kushtëzuar dhe kalimi i kushtëzuar nëbllok	24
3.0 KOMANDIMI LOGJIK	25
3.1 Konfigurimi i CPU-së.....	25

4.0 Dirigjimi i procesit të prodhimit të tullave me ndimën e PLC-së STEP 7 në fabriken e bllokave	
ELBOKSH.....	37
4.1 Ekuacionet logjike të sistemit tëderigjimit për perpunimin e blloqeve tëargjilës.....	39
4.2Blloku Funkzioni dhe Blloku i të dhënave (FB& DB1)	59
4.3 Caktimi vlerave fillestare të parametrave formale	61
5.0 PROGAMIMI DHE INTEGRIMI I NËNBLOQEVE FUNKSIONALE (FB1& DB1 NË FC1)	65
5.1 programimi dhe integrimi I nënbloqeve funksionale (FB2 & DB2 në FC2)	68
PËRFUNDIM	71
LITERATURA	72

HYRJE

Projektimi i sistemeve diskrete e posaçërisht i atyre digjitale të dirigjimit kërkon një njohje më të thellë të teorisë së sistemeve diskrete dhe të rezultateve të sintezës së tyre në regjimet e mundshme të punës dhe në kërkesa të ndryshme nga aspekti i kualitetit të sjelljes. Teoria e diskretizimit dhe e përpunimit digjital të sinjalit është relativisht e kohës së vjetër, mirë e njohur dhe, duhet theksuar, e zhvilluar kryesisht me qëllim të analizës dhe projektimit të sistemeve diskrete të dirigjimit. Këto teknika i kanë marrë dhe tani përdorin disiplinat e reja teknike, siç është përpunimi digjital i sinjalit dhe të ngjashme. Në shkallën e tanishme të zhvillimit, teknikat e sintezës së sistemeve diskrete me degë rivepruese vazhdimisht pasurohen me zgjidhje të reja, prandaj prej inxhinierit të dirigjimit i cili punon dhe studion në këtë lëmi kërkohet përcjellja e vazhdueshme e literaturës bashkëkohore shkencore të lëmis së caktuar.

Është me interes që të theksohet se hendeku ekzistues mes teorisë dhe praktikës të sistemit digjital të dirigjimit është më i madh se në disiplinat tjera teknike. Në fakt bazat teorike janë dhënë në formë analitike, kurse zgjidhjet për sintezën e sistemit – kryesisht në formë të rezultateve të projektimit analitik. Nga ana tjetër, realizimet praktike të dirigjimit në të shumtën e rasteve bazohen në mikrokompjuterët në rolin e procesit të informatave dhe në komponentet digjitale shtesë, teknologjia e të cilave vazhdimisht përsoset dhe pasurohet. Prej këndej, që hendeku i përmendur të kapërcehet, është e domosdoshme njohja e karakteristikave fizike dhe e parametrave të komponenteve të teknikave kompjuterike dhe procedura e shfrytëzimit të tyre në realizimin e dirigjimit digjital. Për këtë arsye janë shtjelluar principet, metodat dhe teknikat e analizës së karakteristikave statike dhe dinamike të objektit të dirigjimit, shndëruesit A/D dhe D/A etj.

1.0 AUTOMATIZIMI I PROCESEVE INDUSTRIALE

Automatizimi i proceseve industriale mund të përcaktohet si disiplinë që studion metodologjitë dhe teknologjitë që lejojnë kontrollin e fluksit të energjisë, të materialeve dhe të informacioneve të nevojshme për realizimin e proceseve prodhuese, pa ndërhyrjen e njeriut. Termi “automatizim” u përdor për herë të parë në vitin 1946 nga shoqëria “Ford Motor” në SHBA (USA) për të kualifikuar ciklin e vet të prodhimit të motorëve.

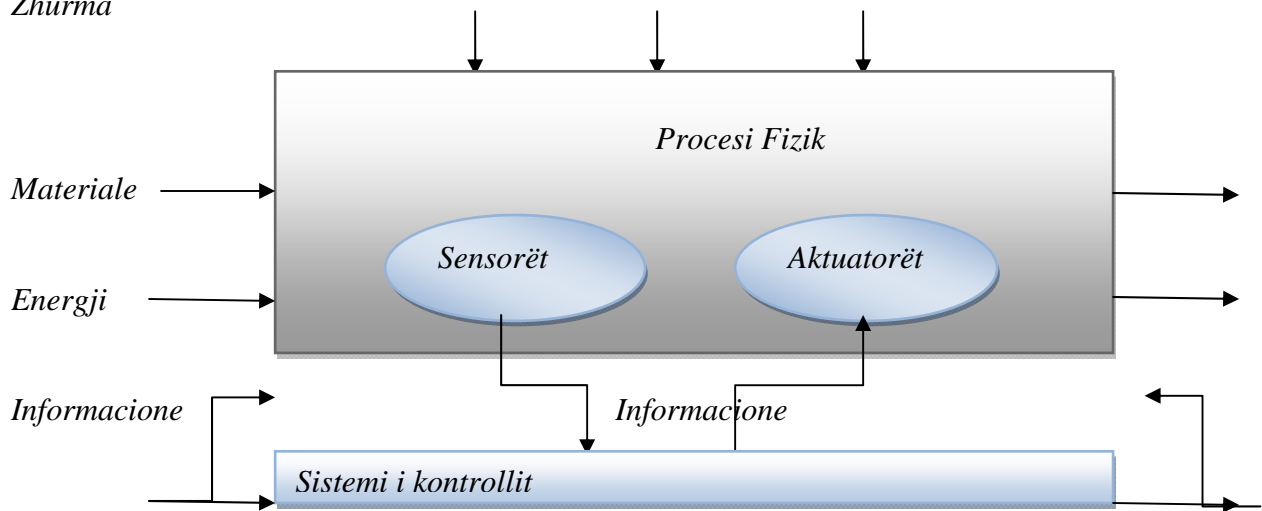
Për kushtet e sotme modern të prodhimit, është i nevojshëm një automatizim shumë fleksibil që mund të përcaktohet si “personalizimi i masës”. Përpara erës industriale, prodhimi ishte artizanal dhe, si rrjedhojë, rezultonte në personalizim të lartë të prodhimeve. Industrializimi dhe prodhimi në masë solli edhe njëtrajtësimin e prodhimit. Më pas u kalua në një fazë ku prodhimit në masë iu bashkëngjiti një varietet më i madh karakteristikash të prodhimeve, për tu arritur në fazën e sotme në të cilën është rritur shumë ky varietet, duke realizuar pikërisht atë që përcaktohet si “personalizim i masës”.

Rëndësia e automatizimit të proceseve prodhuese moderne rrjedh nga një tërësi faktorësh, jo vetëm ekonomike, ku do të përmendim:

- Përmirësimin e cilësive të prodhimit;
- Mundësia e përdorimit të një impianti të vetëm për shumë lloje prodhimesh;
- Shkurtimin e kohëve të prodhimit;
- Mundësinë e reduktimit të magazinave për hyrje dhe dalje;
- Zvogëlimin e jashtëzakonshëm të skarciteteve të prodhimeve;
- Koston më të vogël të prodhimit;
- Nevojën për t’u përshtatur me rregullat dhe ligjet që pengojnë punën me dorë në disa drejtime të industrisë (p.sh., industria farmaceutike dhe ushqimore);
- Mundësinë e reduktimit të impaktit ambiental dhe kursimin e energjisë.
- Në një sistem industrial dallojmë procesin fizik dhe sistemin e kontrollit (fig.1.1.1).

Procesi fizik mund të përcaktohet si kombinim i veprimeve që veprojnë mbi entitete të botës fizike, duke i ndryshuar disa karakteristika. Veprimet e lëvizjes, përpunimet mekanike, reaksionet kimike dhe rrjedhja e flukseve energjetike janë disa nga veprimet që e kënaqin këtë përcaktim dhe si rrjedhojë, mundë të konsiderohen procese fizike; si të tilla janë edhe objekt i automatizimit. Nga ana tjetër, trajtimi i pastër i informacioneve nuk sjell ndryshime në botën reale dhe nuk mund të konsiderohet proces fizik.

Zhurma



Një proces fizik merr në hyrje materiale në formën e prodhimeve të përpunuara dhe energji. Ai gjithashtu merr edhe nga sistemi i kontrollit informacione në forma të ndryshme, si vlera tensionesh ose rrymash elektrike, presion të një lëngu apo skena vlerash binare të koduara.

Procesi prodhon në dalje material në formën e prodhimeve të varura edhe nga energjia. Ai, gjithashtu, dërgon informacione drejt sistemit të kontrollit. Edhe zhurmat që vijnë nga mjedisi dhe që veprojnë mbi proces mund të konsiderohen si hyrje të tij, por si të pamodifikueshme sipas dëshirës.

Informacionet në dalje sigurohen nga pajisje të caktuara që përbëhen nga sensorë, të cilët transformojnë variablin që duhet marrë në tipin e madhësisë që përshtatet për matjen, dhe nga një përbërës të quajtur transduktor, i cili pranon informacione në formë të variablave fizike apo kimike dhe e konverton në madhësi të një natyre tjetër, kryesisht elektrike, e përshtatshme për t'u transmetuar.

Shumë shpesh, sensorët dhe transduktorët ndodhen në të njëjtin element, dhe kjo është arsyeja se pse përgjithësisht me sensor (ose transduktor) i referohemi një pajisje që është në gjendje të masë një madhësi duke dhënë në dalje një sinjal kryesisht të tipit elektrik, që lidhet me të.

Informacionet në hyrje përdoren nga aktuatorët për të alternuar vlerën e variablave të kontrollit të procesit. Zakonisht, aktuatorët, që siç quhen ndryshe *elementët e fundmë të kontrollit*, paraprihen nga pre-aktuatorët, të cilët realizojnë kthimin e informacionit dhe amplifikimin e fuqisë. Për shembull, një valvul është pajisja e fundme e kontrollit për rrjedhjen e një lëngu në një tub, ndërsa motori elektrik që e vë në lëvizje bashkë me funksionimin e tij është pre-aktuatori.

Sensorët, aktuatorët dhe pre-aktuatorët mund të konsiderohen si pjesët realizuese të një procesi fizik dhe përbëjnë ndërfaqen kundrejt sistemit të kontrollit.

Sistemi i kontrollit, si rrjedhojë, merr informacion mbi gjendjen e procesit nëpërmjet sensorëve, i përpunon nëpërmjet algoritmeve të specifikuar dhe u dërgon aktuatorëve informacionet në lidhje me veprimet që duhet të kryhen për të realizuar kontrollin e procesit fizik.

Për këtë qëllim, ai merr formën edhe nga një ose më shumë entitete të jashtme, të cilat mund të jenë operatorë njerëzorë ose sisteme të tjera kontrolli hierarkis më të lartë.

Ky sistem gjithashtu, është në gjendje t'u dërgojë këtyre entitete të jashtme informacione mbi gjendjen dhe mbi procesin e kontrolluar. Nga paraqitja e mësipërme e sistemit të kontrollit rezulton se ai merr, përpunon dhe dërgon informacione, ndaj dhe duhet trajtuar si një sistem që trajton informacionet, pra, një sistem informatik. Një sistem kontrolli është një aplikim i asaj që sot përcaktohet si teknologjia e informacionit dhe komunikimit (ICT, Information and Communication Technology).



1.1 AUTOMATET E FUNDMË

Sjella e automateve me numër të fundmë të gjendjeve të brendshme përshkruhet me sekuencën e ngjarjeve të cilat lajmërohen në momente kohore diskrete $t = 1, 2, 3, \dots$ etj. Supozojmë se automati A pas pranimit të sinjaleve hyrëse, i përgjigjet serisë së gjeneruar të sinjaleve dalëse. Nëse në kohën t automati A pranon sinjalin hyrës $x(t)$, atëherë dalja $y(t)$ e tij varet si nga $x(t)$ ashtu edhe nga gjendja paraprake e sinjalit hyrës. Automati mund të ketë numër të madh të ngjarjeve paraprake, për mbajtjen në mend të të cilave i nevojitet kapaciteti pakufi i memorieve. Pasi që në praktik nuk është e mundur të ndërtohet automati me një numër pakufi të memorieve, ne do të përqendrohemi në automate me numër të caktuar të veprimeve. Do të shqyrtohen automatet të cilët do të kenë mundësi t'i njohin numër të fundmë të klasave paraprake hyrëse të cilave do t'i bashkëngjiten gjendjet e saja të brendshme.

Në aspektin teorik automatet e fundme njihen si pajisje që pasqyrojnë shkronjat e alfabetit, ku secilës shkronjë që sillet në hyrje të tij, i korrespondohet shkronja e caktuar në dalje, ose si quhet ndryshe shkronja e hyrjes pasqyrohet në atë të daljes. Vargu i shkronjave të pranuar në hyrje në mënyrë sekuenciale, njëra pas tjetrës, e formon fjalën hyrëse e cila në dalje gjenerohet në fjalën e caktuar. Nëse shkronjat ose fjalët hyrëse nënkuptohen si informatë mbi programin e ndonjë procesi, ndërsa shkronjat ose fjalët dalëse do të nënkuptohen si komanda për kryerjen e atij programi, atëherë supozimi i dhënë në Teorinë e Automateve të Fundme, e cila shfrytëzohet në teknikën llogaritëse, me më pak modifikime, mund të përdoren në kryerjen e automatizimit në industri.

Para një kohe të afërtë, sistemet e tilla janë ndërtuar në mënyrë intuitive, edhe atë sipas parashikimit dhe eksperiencës së projektantit dhe në bazë të skemave të ngjashme të zgjidhura, një ndërtim i tillë ka kërkuar shumë kohë të gjatë.

Automatet e fundme në mënyrë grafike mund të tregohen si në fig. 1, në formë të një blloku A (sistemi digjital) me n hyrje dhe m kanale dalje. Blloku A përmban pjesën funksionale, ku si operator pasqyron sinjalin hyrës në atë dalës. Në secilin kanal është prezent nga një sinjal diskret i cili shërben për bartjen e informatave. Këto sinjale diskrete mund të kenë numër të caktuar të vlerës, dhe sistemet ku në strukturën e tij kanë elemente me mundësi të pranimit të sinjaleve diskrete, atëherë sistemet e tilla quhen **sisteme digjitale**.

Në kushtet e tanishme në teknikën digjitale shfrytëzohen sinjalet binare, çka do të thotë se

posedojnë vetëm dy vlera që kanë domethënie për ngjarjen se ekziston (1) ose nuk ekziston (0) (p.sh. tensioni, rrymimi, shtypja, rrjedha etj.).

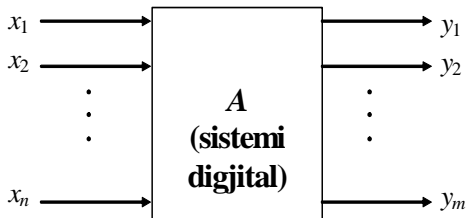


Fig. 1.1

Automati përmban kanalet në hyrje x_1, x_2, \dots, x_n , në të cilët janë prezentë sinjalet binare dhe në dalje kanalet y_1, y_2, \dots, y_m , në këto kanale gjithashtu formohen sinjalet binare. Në hyrje

bashkësia e rregulluar nga vlerat e sinjaleve, formojnë shkronjën hyrëse e që shprehet përmes vektorit \underline{h} me komponentë që kanë vlerë variable $x_i (0 < i \leq n)$:

$$\underline{h} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

Gjithashtu edhe variablat $y_j (0 < j \leq m)$ do të jenë komponentët e shkronjave dalje gjegjësisht vektori \underline{d} :

$$\underline{d} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}.$$

Numri i komponenteve në secilin vektor tregon gjatësinë e sinjalit hyrës dhe dalës. Shkronja \underline{h} dhe \underline{d} i përgjigjen shkronjave edhe atë shkronja e hyrjes $H = \{h_1, h_2, \dots, h_n\}$, e në dalje shkronja $D = \{d_1, d_2, \dots, d_m\}$.

Puna e automatit kryhet atëherë kur në hyrje të tij lajmërohet shkronja \underline{h}_i dhe në dalje krijohet shkronja \underline{d}_j . Automati është i caktuar nëse është e vendosur korrespondenca e shkronjës hyrëse me atë të daljes dhe shenohet:

$$\underline{h}_i \rightarrow \underline{d}_j.$$

Automati që ka numër të caktuar të kanaleve të hyrjes dhe të daljes ku sinjalet do të marrin vlerën e bashkësisë $\{0, 1\}$ do të ketë numër të caktuar të kombinimeve hyrëse dhe dalëse.

Kombinimet e hyrjeve dhe të daljeve shprehen në formë të bashkësisë së rregulluar të vlerave të sinjaleve

përkatësisht $(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$ dhe $(\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_m)$, ku simboli “ \sim ” tregon vlerën e sinjalit të shkronjave alfabetike të hyrjes dhe daljes me gjatësi konstante. Të gjitha këto shkronja përmes sinjaleve të koduara me n gjegjësisht m shifra binare $\{0, 1\}$ i emërtojnë hyrjet gjegjësisht daljet. Pasi që ka numër të caktuar të shkronjave hyrëse dhe dalëse, atëherë automatet e tilla i quajm **të caktuar – fundmë**.

Nëse gjatë punës është siguruar korrespondimi i sigurt në mes shkronjave hyrëse dhe dalëse, ashtu që secilës shkronjë hyrëse (kombinimi i vlerave të sinjaleve hyrëse), pavarësisht nga koha kur ajo lajmërohet, gjithmon i përgjigjet vetëm një shkronjë e caktuar e daljes, që do të thotë se ekziston pasqyrimi njëvlefshëm, nga kjo, këto automate quhen **kombinuese** ose automate **pa memorie**.

Nëse kemi që nga një shkronjë hyrëse në kohë të ndryshme gjenerohet në shkronja të ndryshme dalëse, që do të thotë ekziston pasqyrim shumëvlefshëm, nga kjo, këto automate quhen **sekuenciale** ose automate **me memorie**. Te këto automate shkronja dalëse nuk varet vetëm nga shkronja hyrëse në kohën e caktuar, por edhe nga shkronjat hyrse paraprake.

1.2 SISTEMET NUMERIKE

Për shfrytëzim të sistemeve digjitale (kompjuterit) është e domosdoshme të njihet principi i punës së tyre, e para së gjithash të sistemit numerik, shndërrimin e vlerave numerike prej një sistemi në tjetrin dhe të veprimeve aritmetike brenda sistemit të dhënë. Është e qartë pse sistemi numerik me bazë 10 (dekad) konsiderohet si "natyral", por prapëseprap, për shkak të realizimit të tyre fizik sistemet digjitale (kompjuterët) punojnë me sistem binar, baza numerike të cilëve është 2.

Sistemet digjitale, të cilët përdoren si organe për drejtimin automatik, janë të komponuar nga elementet vetëm me dy gjendje dhe operojnë me sinjale binare {0, 1}. Gjithashtu edhe informatat të cilat lajmërohen e që shërbejnë në përpunimin e karakterit diskret, përdorin sinjalet binare. Për përpunimin dhe kodimin e tyre rol të rëndësishëm luan sistemi numerik binar. Andaj është e nevojshme të njihemi mirë për sistemet numerike, të cilët mund të parashihen si sisteme për kodim.

1.3 PËRDORIMI I SISTEMEVE TJERA NUMERIKE DHE KODIMI I TYRE

E metë e sistemit numrik dysh është numri i madh i rendeve të ndryshme për paraqitjen e një numri; kështu, për shembull, numri trerendësh $197_{(10)}$ në sistemin dysh paraqitet me tetë rende; numrat dhjetorë me pesë rende që përdoren gjerësisht në dirigjimin e makinave metalpunuese, në sistemin dysh shprehen me 16 rende. Meqenëse çdo rendi i përgjigjet një element elektronik ose elektrik ose pneumatik, etj., për shprehjen e tij, kuptohet se në makinë do të kemi një numër të madh elementesh të tilla.

Përveç sistemit tetësh, njëfarë përhapjeje ka edhe sistemi gjashtëmbëdhjetësh, por ai që përdoret më shumë është sistemi dyshodhjetor, ku kombinohet përshtatshmëria e përdorimit të sistemit dysh dhe ekonomia e sistemit dhjetësh. Me këtë rast humbet mundësia e përparësisë së treguar të numrave në në formë binare, por shumë mangësi (pengesa) largohen gjatë përpunimit të të dhënave me makinë, sepse madhësit e hyra dhe madhësit e dala duhet të jenë dekadë, do të thotë se secili numër në hyrje dhe në dalje duhet të kodohet. Pajisja për kodimin e numrave si një tërësi nga forma dekadë në formën

binare është shumë e komplikuar dhe e shtrenjët, prandaj kodimi i secilës dekad në veti kodohet. Atëherë kodimi (C) binar (B) dekad (D) ose në gjuhën ang. (Binary Coded Decimal–BCD) dhe kështu janë formuar kode të ndryshme BCD.

Në këtë sistem, çdo shifër e sistemit dhjetësh (0, 1, ..., 9) shprehet me një numër katërrendësh të sistemit dysh, që quhet tetradë.

Tetrada formohet sipas një kodi të caktuar, kështu me tetradën 8421, e formuar nga numrat 2^3 , 2^2 , 2^1 dhe 2^0 të sistemit dysh, çdo shifër e sistemit dhjetësh paraqitet si shumë e numrave të këtij kodi (tabela 7).

Tabela 1.1.

Tetrada				Llojet e kodeve										
Numri i vendeve				Binare	8-4-2-1	Aiken	Exces-3	Këcim për 2	Këcim për 8	4-2-2-1	5-4-2-1	5-2-2-1	5-3-1-1	White
4	3	2	1											
0	0	0	0	0	0	0		0	0	0	0	0	0	0
0	0	0	1	1	1	1		1	1	1	1	1	1	1
0	0	1	0	2	2	2			2	2	2	2		
0	0	1	1	3	3	3	0		3	3	3	3	2	2
0	1	0	0	4	4	4	1		4		4		3	
0	1	0	1	5	5		2		5				4	3
0	1	1	0	6	6		3		6	4		4		
0	1	1	1	7	7		4		7	5				4
1	0	0	0	8	8		5	2			5	5	5	5
1	0	0	1	9	9		6	3			6	6	6	6
1	0	1	0	10			7	4			7	7		
1	0	1	1	11		5	8	5			8	8	7	7
1	1	0	0	12		6	9	6		6	9		8	
1	1	0	1	13		7		7		7			9	8
1	1	1	0	14		8		8	8	8		9		
1	1	1	1	15		9		9	9	9				9
Binar				decimal										

Për shembull, numri 3526 do të paraqitet në këtë mënyrë:

$$3526_{(10)} = 0011\ 0101\ 0010\ 0110_{(2)}$$

Meqenëse në llogaritjen për shndërrimin e numrit nga sistemi dhjetësh në sistemin dysh ndeshemi në disa vështirësi, por për kodimin e përmasave gjatë përpunimit në sistemet e dirigjimit të programuar, ngandonjëherë përdoret tetrada 2421 e formuar nga numrat dysh $2^1, 2^2, 2^1, 2^0$. Kodimi i çdo shifre në tetradën 2421 tregohet në tabelën 8.

Tabela 1.2.

Sistemi dhjetësh	Tetrada 2421
	$2^1 2^2 2^1 2^0$
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	0110
7	0111
8	1110
9	1111

Përveç kodeve 8421 dhe 2421, ka edhe kode të tjera të ndërtuara mbi sistemin dysh-dhjetë. Analiza e tyre tregon se më i përshtatshëm është kodi 2421.

Për kodimin binar, të numrit në dekadën decimale, duhet secilës dekad t'i ndahet 4 bita, ndërsa teorikisht minimumi është 3,3 bita, çka do të thotë se do të mbeten të pa shfrytëzuar për secilën dekad 0,7 bita, përkatësisht 19,7%.

Pasi që secila dekadë e përdorë minimumin prej 4 bita, kjo do të na japë 16 kombinime të ndryshme, d.m.th. në numrin dual prej 16 kombinimeve të mundshme do të shfrytëzohen cilido prej 10 kombinimeve.

Kombinimet e veçanta të 4 shenja binare ndryshe i quajmë **tetrade**, ndërsa tetradat e pa shfrytëzuara i quajmë **pseudotetradat**.

Tabela 1.3.

Shenja e kodeve		8 - 4 - 2 - 1				Aiken				Exces - 3 (Stibitz)			
Vlera		8	4	2	1	2	4	2	1	nuk ka			
Bashkagjitja e numrave	0	0	0	0	0	0	0	0	0	0	0	1	1
	1	0	0	0	1	0	0	0	1	0	1	0	0
	2	0	0	1	0	0	0	1	0	0	1	0	1
	3	0	0	1	1	0	0	1	1	0	1	1	0
	4	0	1	0	0	0	1	0	0	0	1	1	1
	5	0	1	0	1	1	0	1	1	1	0	0	0
	6	0	1	1	0	1	1	0	0	1	0	0	1
	7	0	1	1	1	1	1	0	1	1	0	1	0
	8	1	0	0	0	1	1	1	0	1	0	1	1
	9	1	0	0	1	1	1	1	1	1	1	0	0

Kodi 8 - 4 - 2 - 1 është i ngjashëm me kodin e pastër binar, mirëpo ky kod i shfrytëzon vetëm 10 tetradat e para, ndërsa 6 tetradat tjera mbeten të pa shfrytëzuara. Kodi i pastër binar i shfrytëzon të 16 tetrada. Kodi 8-4-2-1 është i përshtatshëm për numërim e jo për kalkulim. Ky kod e ka edhe një të metë lidhur me numrin **0** i cili në prezencën e vetë nuk e ka asnjë 1 do të thotë (0000). Gjithashtu ky kod nuk është simetrik, prandaj nuk kemi zbritje të njëvlershme.

Në kodin Stibitz është e nevojshme disa korrekture, të cilat nuk janë shumë dhe jo të përbërshme, e nejta vlen edhe për kodin Aiken.

Përpos kodit **BCD** 4-bitësh që është më i përshtatshëm për përdorim, ekzistojnë edhe kode me më shumë se 4-bita, si për shembull 5, 7, 10 e më tepër që kanë efekte të mira në ndonjë qëllim të caktuar.

Kodi $\binom{10}{1}$ është njëri prej 10 më të përshtatshmit si për madhësitë hyrëse ashtu edhe për madhësitë dalëse për përpunimin manual të të dhënave. Sepse për secilin numër dekad përdoret 10 bitëshi, prej tyre vetëm njëri e ka vlerën binare 1, ndërsa të tjerët e kanë 0. Në vendin e parë njëshi (1) e ka shenjën e dekadës 0, ndërsa njësit në vendet 2, 3, ..., 10 i kanë numrat 1, 2, ..., 9. Te ky kod do të kemi aq vende ose bita, për aq sa dëshirojmë të kodojmë, ku në rastin e tonë kemi 10. Nëse dëshirojmë të bëjmë kodimin m shenjave, do ta përdornim kodin të i cili secila fjalë do të ketë m bita, ku njëri prej m shenjave do ta ketë vlerën 1, ndërsa të tjerët 0.

Kodet e tilla janë paraqitur në tabelën 1.4.

Tabela 1.4.

Lloji i kodeve	$\binom{10}{1}$										8 - 4 - 2 - 1 - 0 kodi					Kodi numër 3				
Vendi	10	9	8	7	6	5	4	3	2	1	5	4	3	2	1	5	4	3	2	1
Vlera	9	8	7	6	5	4	3	2	1	0	8	4	2	1	0	Nuk ka				
Numrat decimal	0									1	1		1				1		1	1
	1								1					1	1	1	1	1		
	2							1					1		1	1	1		1	
	3							1					1	1		1	1			1
	4						1					1			1	1		1	1	
	5					1						1		1		1		1		1
	6				1							1	1			1			1	1
	7			1							1	1						1	1	1
	8		1								1				1		1	1	1	
	9	1									1			1			1	1		1

Kodi 8-4-2-1-0, ka pesë bita dhe me peshë 2 dhe shërben bartjen e numrave përmes teleks-kanalit.

Në tabelën 1.5 janë dhënë kodet me nga 7 bita.

Tabela 1.5.

Lloji i kodit	Kodi biginar							Kodi qibinar							Kodi biginar reflektues							
	Vendi	7	6	5	4	3	2	1	7	6	5	4	3	2	1	7	6	5	4	3	2	1
Vlera	5	0	4	3	2	1	0	8	6	4	2	0	1	0	Nuk ka							
Numrat decimal	0		1				1					1		1		1						1
	1		1				1					1	1			1					1	
	2		1			1					1			1		1			1			
	3		1		1						1		1			1		1				
	4		1	1						1				1		1	1					
	5	1						1		1			1		1		1					
	6	1					1			1				1	1				1			
	7	1				1				1				1		1				1		
	8	1			1				1					1	1						1	
	9	1		1					1					1		1						

Të tre llojet e kodeve me 7 bita të treguar në figurën 10 janë të përshtatshëm për zbulimin e gabimit. Ku të tre llojet peshën e kanë 2. Sot te repartet e automatizuara me shumicë përdoret leximi mekanik i gjatësisë dhe këndeve me anën e këtyre kodeve.

2.0 SIMBOLET DHE VARIABLAT PËR PROGRAMIMIN E PLC-së

2.1 URDHËRAT E PËRDORUR PËR PROGRAMIMIM

Përshkrimi

Urdhërat e “Bit Logic” punojnë me dy shifra, 1 dhe 0. Këto dy shifra e formojnë bazën e një sistemi numerik të quajtur sistem binar. Të dy shifrat 1 dhe 0 quhen shifra binarë apo bita. Në botën e kontakteve dhe mbështjelljeve, një “1” tregon që një gjendje është aktivizuar apo ka energji, dhe një “0” tregon se nuk shprehet aktivizim ose mund të themi se nuk ka energji. Udhëzimet “Bit Logic” tregojnë gjendjet e sinjalit “1” dhe “0”, të kombinuara sipas logjikës së Boolean. Këto kombinime që prodhojnë rezultat prej gjendjes “1” ose “0” quhen rezultat i operacionit të logjikës (RLO).

Operacionet logjike që janë shkaktuar nga udhëzimet Bit Logic kryejnë një sërë funksionesh.

Ekzistojnë shumë udhëzime Bit Logic për të kryer operacionet e mëposhtme:

- ---|--- Kontakti në gjendjen normale i hapur (Address)
- ---|/|--- Kontakti në gjendjen normalei mbyllur
- ---(SAVE) Ruajtja e RLO në kujtesën BR
- XOR Bit ekzkluziv OR
- ---() Spirala dalëse ose (prodhuese)
- ---(#)--- Mesvijë prodhuese
- ---|NOT|--- Energji rrjedhëse anasjelltas.

Udhëzimet e mëposhtme reagojnë ndaj një RLO nga gjendja “1”

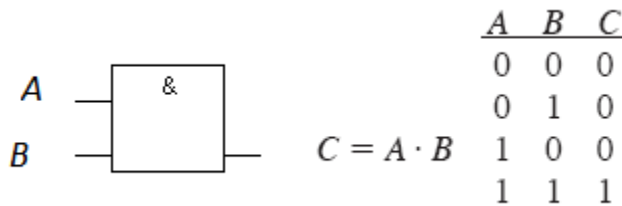
- ---(S) Spirale e vendosur (Set Coil)
- ---(R) Spirale e rivendosur (Reset Coil)
- SR Vendos Rivendos “Flip Flop” (Set Reset)
- RS Rivendos Vendos “Flip Flop” (Reset Set).

Udhëzimet tjera reagojnë në tranzicionin pozitiv ose negative, për të kryer funksionet e mëposhtme:

- ---(N)--- Zbulimi negativ RLO
- ---(P)--- Zbulimi pozitiv RLO
- NEG Buzë-zbulimi apo Adresimi ngative
- POS Buzë-zbulimi apo Adresimi pozitiv
- Immediate Read (Lexim i menjëhershëm)
- Immediate Write (Shkrim i menjëhershëm)

2.2 SIMBOLET

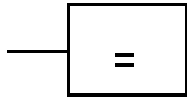
2.2.1 Simboli “&” (dhe, AND)



Me “&” (AND) mund të kontrollojmë sinjalin e “2” ose më shumë hyrjeve specifike. Nëse gjendja e sinjalit është “1” kushti është i mjaftueshëm dhe rezultati i daljes është i barabartë me “1” (në këtë rast njësia hyrëse është e barabartë me njësinë dalëse).

Nëse sinjali i hyrjes është “0” (Zero) kushti është i pa mjaftueshëm, dalja është “0”. Nëse blloku “&” është blloku i parë në operacionin logjik, ai ruan rezultatin nga R20 bit. Çdo bllok “&” i cili nuk është bllok i parë, kombinon rezultatin e sinjaleve hyrëse me vlerën e ruajtur nga biti RLO. Këto vlera kombinohen në bazë të tabelës së bllokut “&” (AND).

2.2.2 Simboli barazim (Assing)

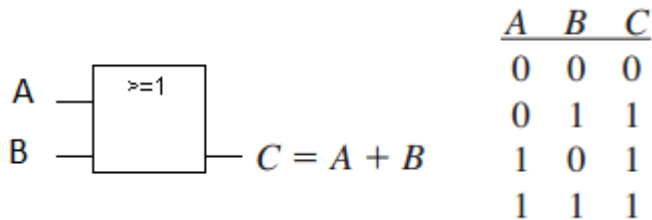


Ky simbol prodhon rezultatin e operacioneve logjike. Simboli në fund të operacionit logjik ka sinjalin “1” ose “0” bazuar në këto kritere:

Sinjali në dalje është “1” kur kushti i operacionit logjik para daljes në kuti është i plotësuar.

Sinjali është “0”, kur kushti nuk është plotësuar.

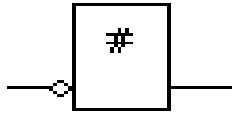
2.2.3 Operacioni logjik OSE (≥ 1)



Me operacionin logjik OSE (≥ 1) mund të kontrollojmë sinjalin hyrës të dy ose më shumë adresave hyrëse. Nëse gjendja e njërit nga sinjalet hyrëse është “1” kushti plotësohet, operacioni logjik OSE prodhon rezultatin në dalje 1. Nëse gjendja e sinjalit nga të gjitha adresat është “0” kushti nuk plotësohet, rezultati në dalje është “0”. Nëse operacioni logjik OSE është operacioni i parë në varg ai i ruan rezultatet e gjendjes së sinjalit në RLO bit.

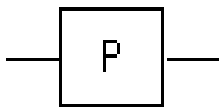
Secili operacion logjik OSE që nuk është operacion i parë në varg kombinon rezultatin e sinjalit të kontrolluar të tij me vlerën e ruajtur nga biti RLO.

2.2.4 Vija e mesme dalëse “#” (Midline Output)



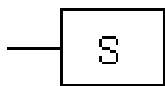
Vija e mesme dalëse është një element ndërmjetësues i cili e amortizon RLO-në. Më saktësisht ky element amortizon operimin e bitit logjik të fushës së fundit në mënyrë që të hapet para daljes së vijës së mesme. Ky operacion dirigjohet nga Master Contro Rele (MCR) Releja master kontrolluese.

2.2.5 Detektimi i RLO-së pozitive (P)



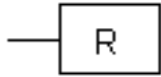
Operacioni i Detektimit të RLO-së Pozitive është operacion i cili detekton ndryshimin nga “0” në “1” të adresës specifike dhe në të ndikon një RLO me vlerë “1” pas udhëzimit. Gjendja e sinjalit aktual të RLO-së është e kombinuar me gjendjen e sinjalit të adresës. Nëse gjendja e sinjal të adresës është 0 dhe RLO ka vlerë “1” para udhëzimit të këtij operacioni, RLO do të jetë “1” (pulës) pas udhëzimit. Në të gjitha rastet e tjera RLO është “0”.

2.2.6 Starto prodhimin (vendos daljen) S Set Output



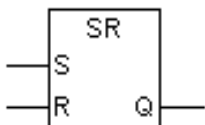
Udhëzuesi “Starto prodhimin” ekzekutohet vetëm atëherë kur RLO është “1”. Ky udhëzues vendos adresat specifike në “1”. Nëse RLO është “0” ky udhëzues nuk ka ndikim në adresën specifike, që do të thotë nuk ka ndryshim. Udhëzuesi Start dirigjohet nga Master Kontroll Rele MCR.

2.2.7 Ristarto prodhimin (rivendos daljen) R Reset Output



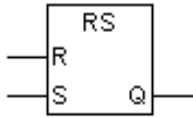
Udhëzuesi Ristarto prodhimin ekzekutohet vetëm atëherë kur RLO është “1”. Nëse RLO është “1” ky udhëzues ristarton adresat specifike në “0”. Nëse RLO është “0” udhëzuesi nuk ka efekt në adresat specifike që do të thotë nuk ka ndryshim. Ky udhëzues dirigjohet nga Master Kontroll Rele MRC.

2.2.8 Starto_Ristarto Flip Flop (vendos_rivendos) SR Set_reset Flip Flop



Udhëzuesi starto_ristarto flip flop ekzekuton startimin (S) ose ristartimin (R) vetëm kur RLO është “1”. Kur RLO është “0” ky udhëzues nuk ka efekt. Udhëzuesi SR startohet kur gjendja e sinjalit në hyrje S është “1” dhe gjendja e sinjalit në hyrje R është “0”. Kur gjendja e sinjalit në hyrjen S është “0” dhe në hyrjen R është “1”, Flip flopi ristartohet. Nëse një RLO në të dyja hyrjet S dhe R është “1” Flip Flopi Ristartohet. Ky udhëzues dirigjohet nga Master Kontroll Rele MRC.

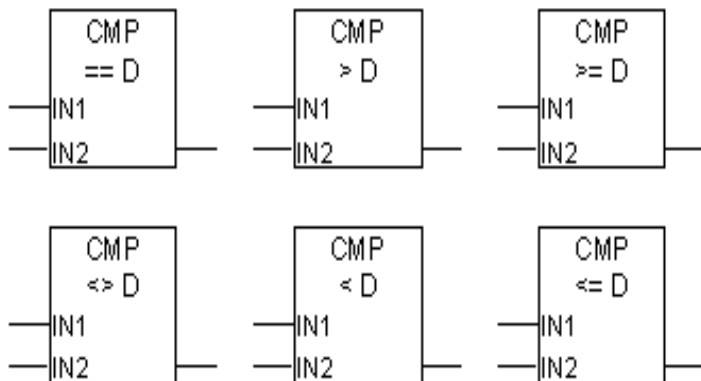
2.2.9 Ristarto_Starto Flip Flop (rivendos_vendos) RS Reset_set Flip Flop



Udhëzuesi Ristarto_Starto Flip Flop ekzekuton startimin (S) ose ristartimin (R) vetëm kur RLO është “1”. Një RLO me vlerë “0” nuk ka efekt në këtë udhëzues, ky udhëzues ristartohet kur gjendja e sinjalit në hyrjen R është “1” dhe në hyrjen S është “0”.

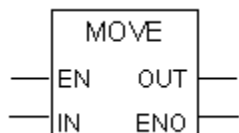
Nëse hyrja R është “0” dhe hyrja S është “1” flip flopi startohet. Nëse të dy hyrjet e RLO-së R dhe S janë “1” flip flopi startohet. Ky udhëzues dirigjohet nga Master Kontroll Rele MRC.

2.2.10. Krahasimi i dy vlerave



Udhëzuesi i krahasimit të dy vlerave krahason dy vlera nga baza 32 bit me pikën qarkulluese të numrave. Ky udhëzues krahason hyrjet IN1 dhe IN2 bazuar në llojin e krahasusit, për secilin rast konkret CMP == D, CMP > D, CMP >= D, CMP <> D, CMP < D, dhe CMP <= D.

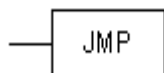
2.2.11 Lëvizja (Move): Transferimi i vlerave



Me udhëzuesin për transferimin e vlerave mund të transferojmë vlerat specifike në variabla. Vlera specifike e hyrjes IN kopjohet në adresën specifike në daljen OUT. ENO ka gjendjen e sinjalit të njëjtë sikurse EN. Me kutinë zhvendosësh udhëzuesi i transferimit të vlerave mund të kopjon të gjitha të dhënat elementare me 8, 16 ose 32 bit.

Mund të caktojmë llojin e të dhënave sikurse në grupe ose struktura të cilat duhet të kopjohen në sistemin e funksionit SFC 20 “BLKMOV”. Ky udhëzues dirigjohet nga Master Kontroll Rele MCR.

2.2.12 Kalimi i pa kushtëzuar dhe kalimi i kushtëzuar në bllok



Urdhër i kalimit të pa kushtëzuar në bllok korrespondon në urdhrin “shko tek emërtimi”, asnjëri nga urdhrat në mes të kalimit dhe emërtimit nuk ekzekutohet. Ky urdhër mund të përdoret nga të gjitha blloqet logjike p.sh blloqet organizative (OB-të), blloqet funksionale (FB-të), dhe funksionet (FC-të). Nuk duhet të ketë asnjë operacion logjik para kutisë së kalimit të pa kushtëzuar.



Për dallim nga kalimi i pa kushtëzuar në bllok, te kalimi i kushtëzuar urdhri ekzekutohet vetëm në rastin kur RLO është “1”.

3.0 KOMANDIMI LOGJIK

3.1 KONFIGURIMI I CPU-së

Të dhënat e nevojshme paraprake të PLC-së ekzistuese, për konfigurim dhe programim të CPU-së me të dhëna ekzakte të PLC-së.

Pasi që fillimisht kemi instaluar programin SIMATIC S7 300 në kompjuterin personal (PC), hapim programin dhe fillojmë konfigurimin e CPU-së me të dhënat ekzistuese. Fillimisht hapim një dritare të re për të filluar një projekt të ri, si dhe duke e emërtuar projektin p.sh., Tema Master. Ky projekt do të shërbejë për fillimin e konfigurimit të CPU-së, dhe më vonë për programimin e nevojshëm në PLC. Duhet theksuar se është e pamundur që të programohet një PLC, në qoftë se paraprakisht nuk është bërë konfigurimi i CPU-së.

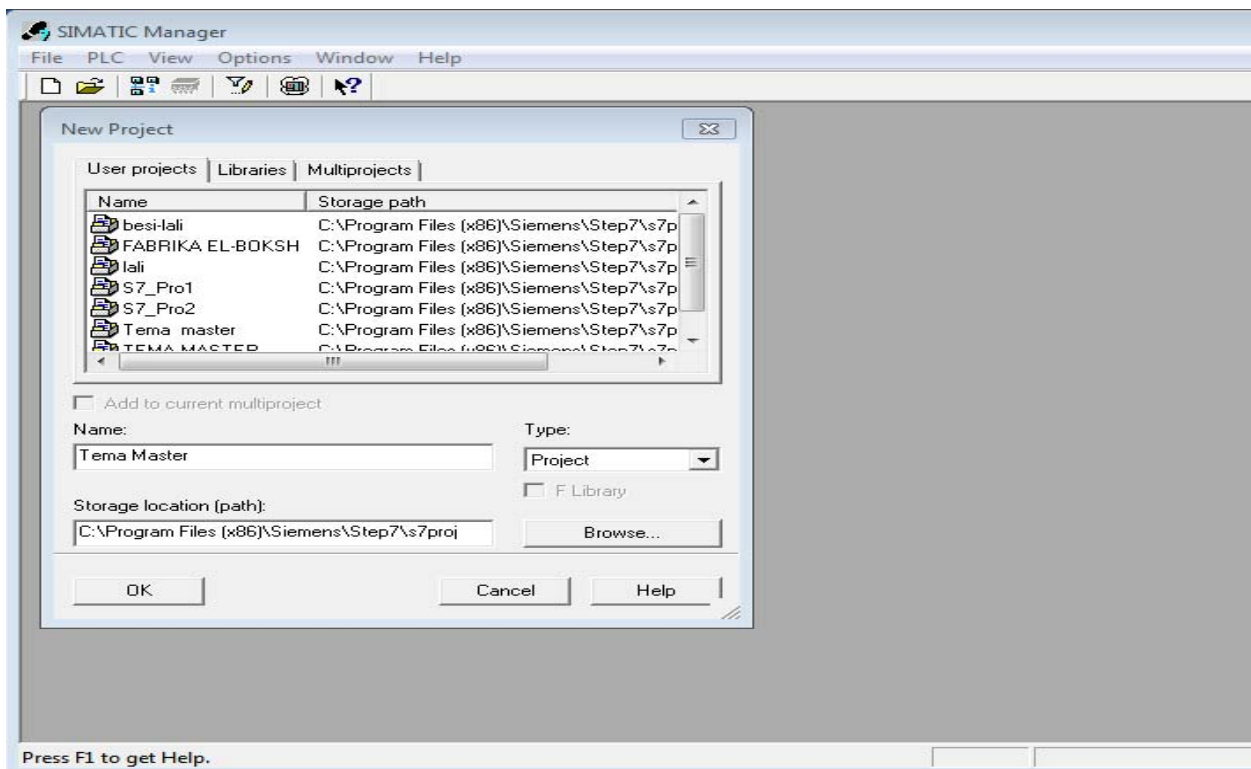


Fig.3.1 Të dhënat e nevojshme paraprake të PLC-së ekzistuese, për konfigurim dhe programim të CPU-së me të dhëna ekzakte të PLC-së.

Pasi që kemi emërtuar projektin e ri dhe është ruajtur në program, Tema Master klikojmë (Insert Station) zgjedhim modelin e PLC-së që ne e kemi në disponim (SIMATIC 300 Station)

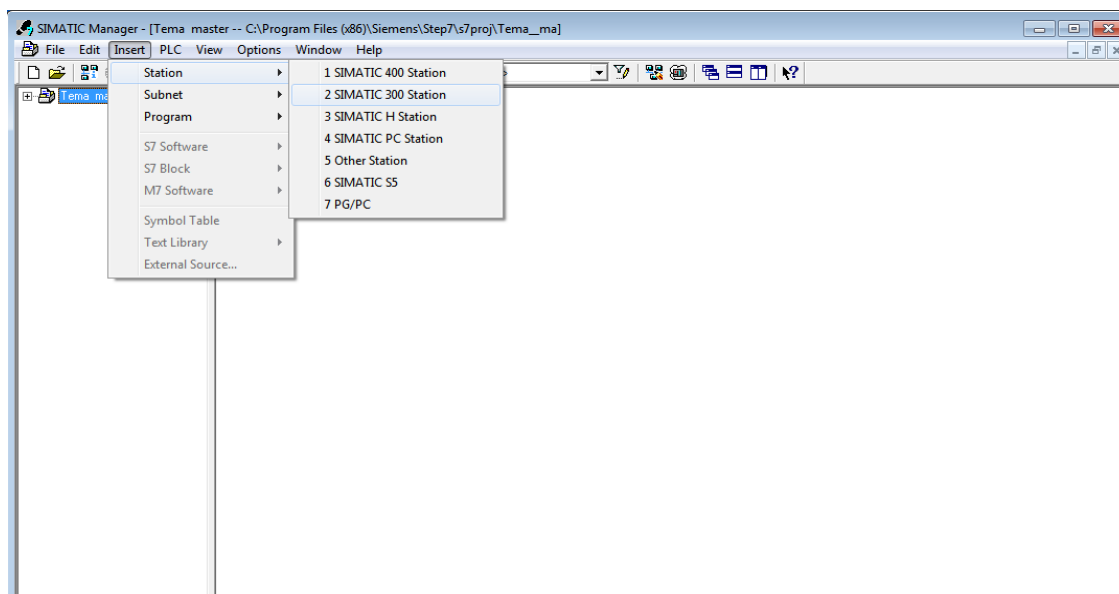
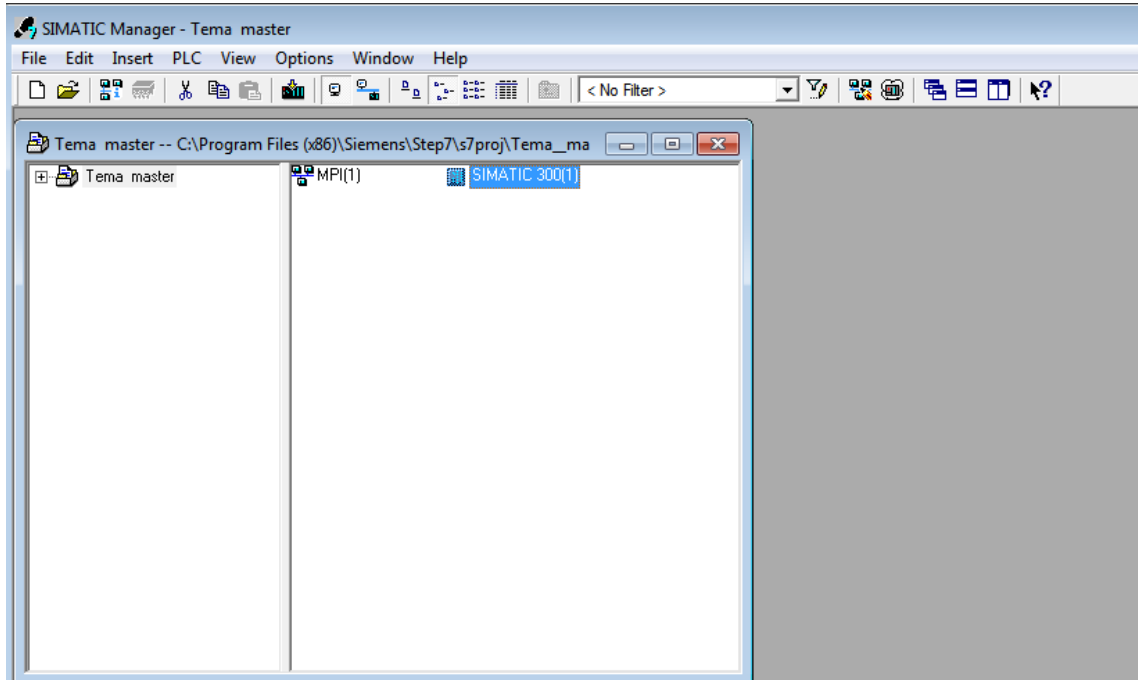


Fig.3.2 Zgjedhim modelin e PLC-së që ne e kemi në disponim SIMATIC 300 .

Zgjedhim modelin SIMATIC 300(1), brenda programit është krijuar një folder për konfigurim me po këtë emër.



Dhe duke klikuar me tastin e majtë mbi SIMATIK 300(1) të, formohet një dritare e re me emrin (Hardware).

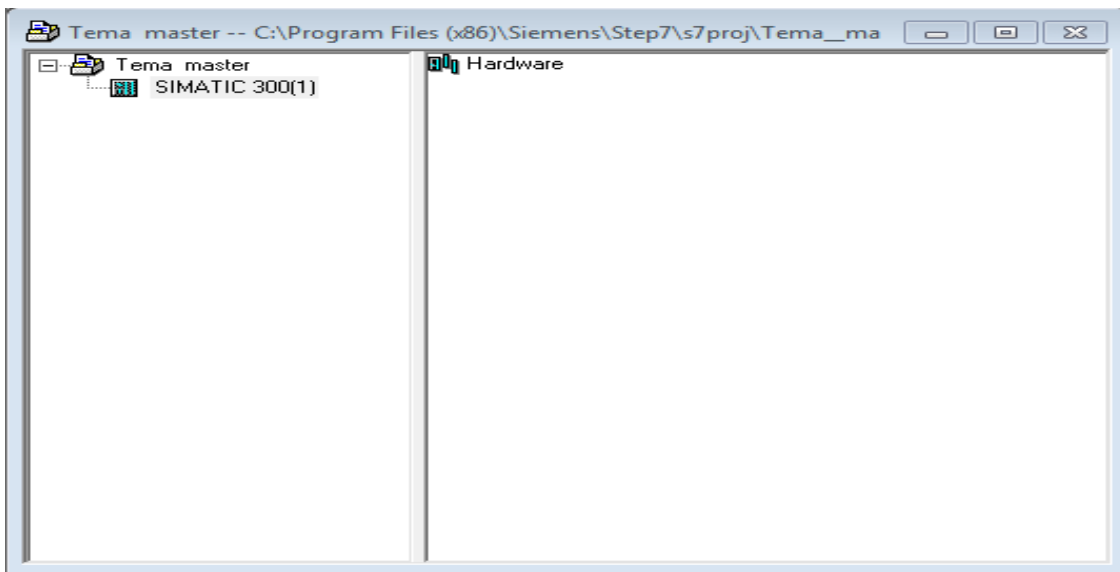


Fig.3.3. Formimi i dritares me emrin Hardware

Bëjmë hapjen e dritares së krijuar më parë (Hardware) dhe mbi të klikojmë, ku hapet dritarja e re me modele të ndryshme të PLC-ve, dhe ne zgjedhim PLC-në tonë **SIMATIC 300**.

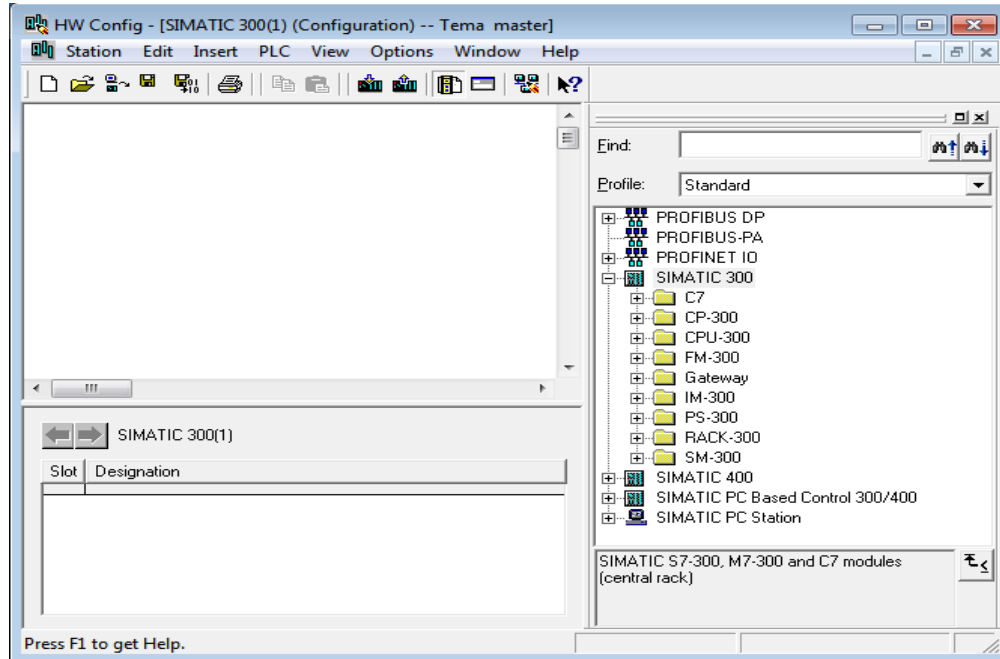


Fig.3.4 Zgjedhim PLC-në tonë SIMATIC 300

Zgjedhim folderin **Rack 300**, klikojmë mbi të dhe zgjedhim **Rail**, ku me këtë rast në anën e majtë të dritares hapën dy tabela për ruajtjen e të dhënave të PLC-së për konfigurim.

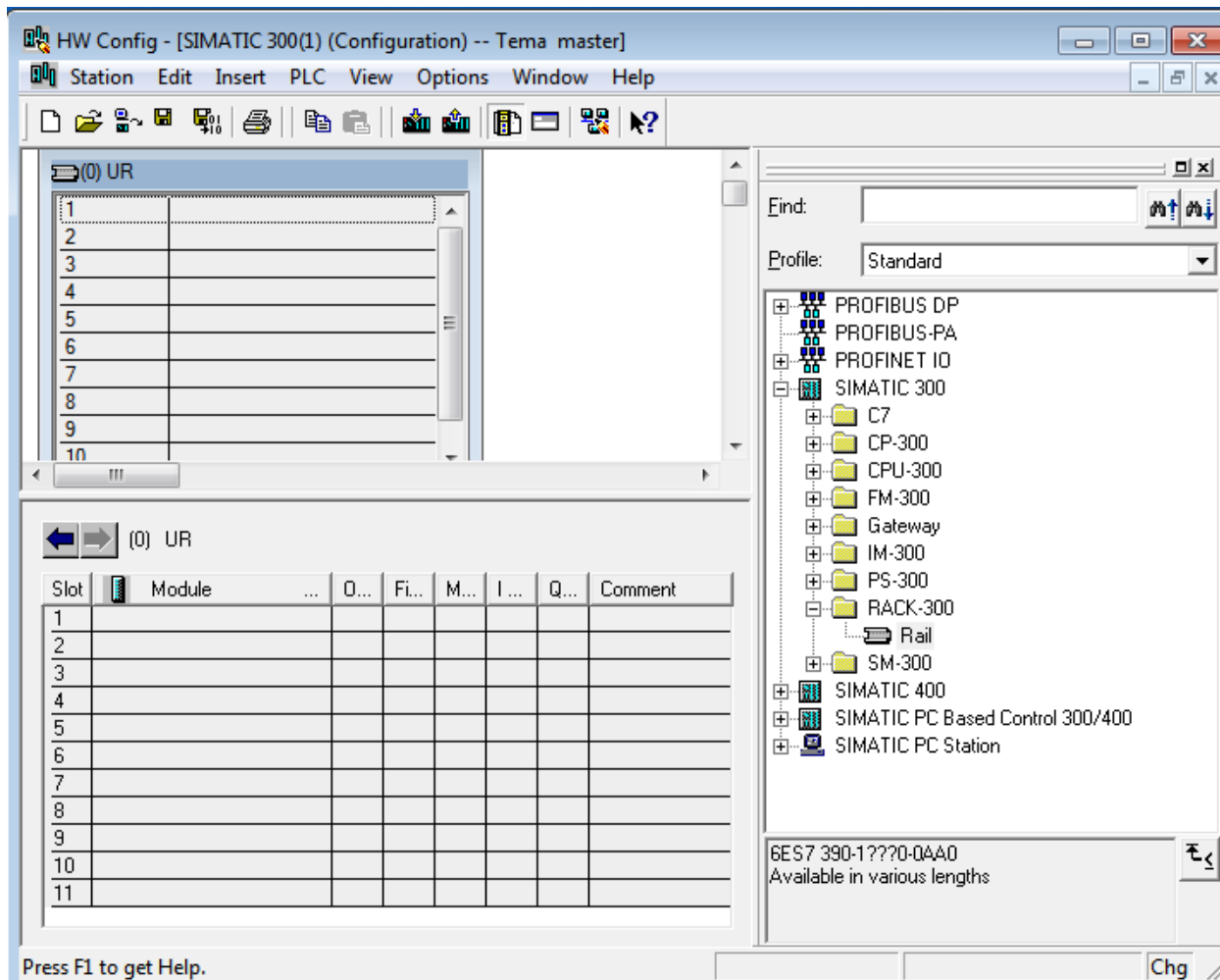


Fig. 3.5 Tabela për zgjedhjen e të dhënave përkatse të PLC

Folderi RACK-300 përkatësisht Rail është tabela ku në të cilën tabelë vendosen të gjitha hyrjet edhe daljet DI/DO e PLC-së ekzistuese leximi i vlerave hyrëse, dalëse edhe hyrëse/dalëse bëhet të modulet e të cilat module vendosen pran PLC-së, selektojm rreshtin e parë me numër rendor një, dhe pas selektimit të rreshtit zgjedhim të dhënat përkatëse të PLC-së për amper (A), ku PLC-ja në rastin tonë punon me 5A, dhe bëhet zgjedhja e dhënë **PS 307 5A**.

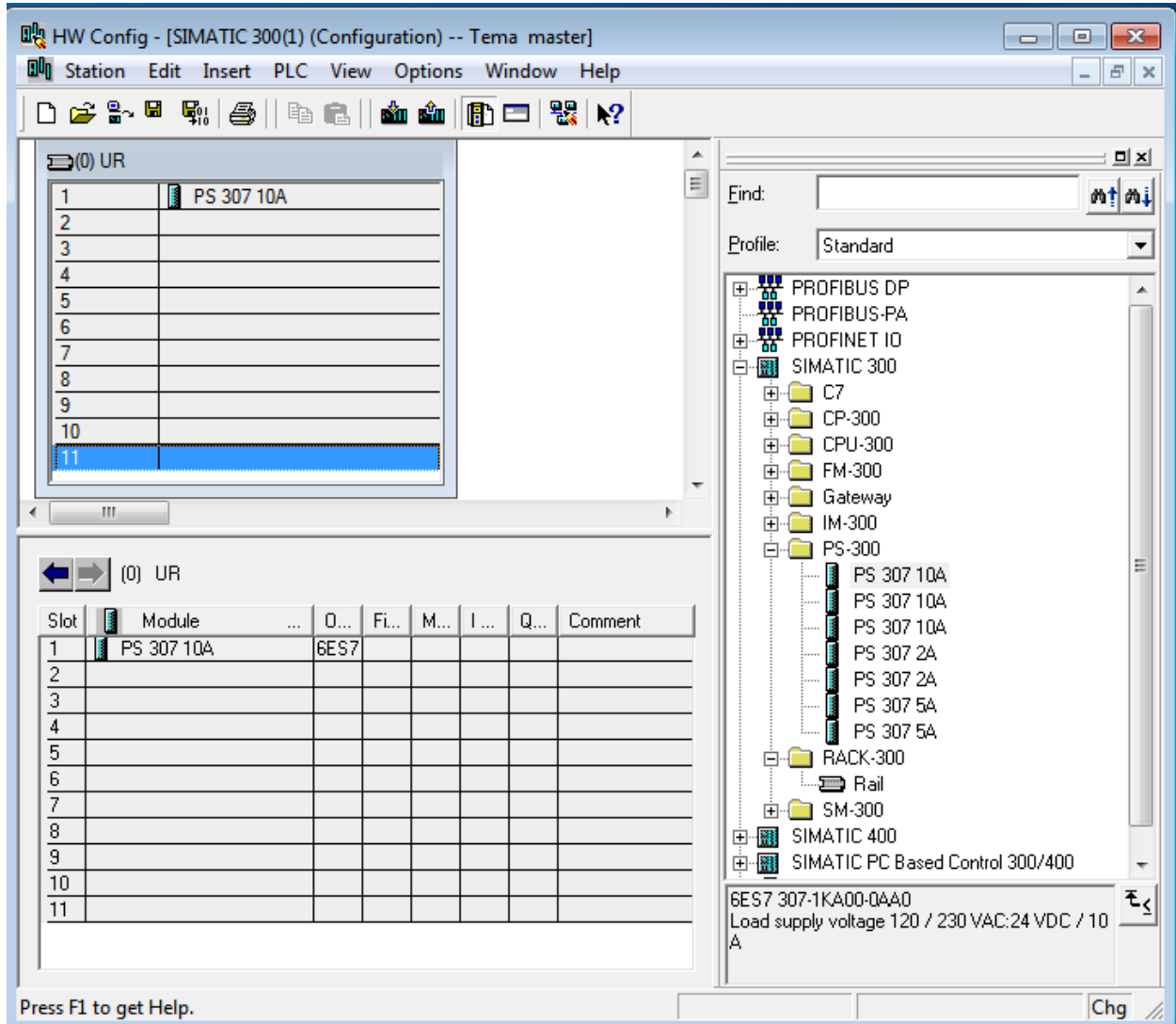


Fig.3.6 Zgjedhim të dhënat e CPU- së

Në folderat e hapur tek SIMATIC 300, pasi që kemi selektuar rreshtin e dytë në tabelën e poshtme, zgjedhim të dhënat e CPU-së ekzistuese CPU 314C-2 PN/DPe cila punon me 24.0V.

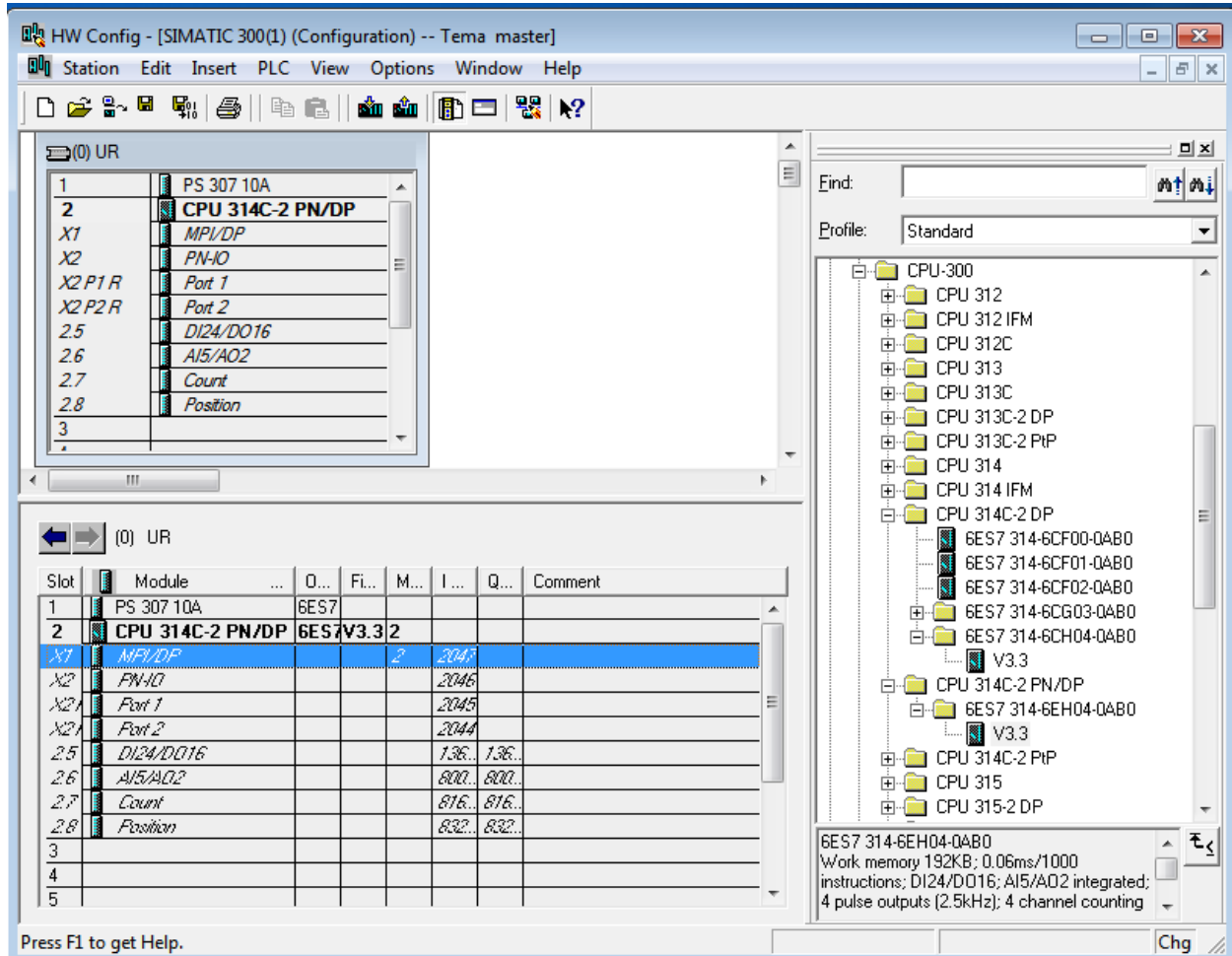


Fig 3.7Zgjedhim të dhënat e CPU-s

Vlen të theksohet se veçoria e këtij program është se në rreshtin e tretë informatat nuk mund të merren pasi që programi nuk e lejon një gjë të tillë, atëherë selektojmë rreshtin e katërt dhe i vendosim informatat hyrëse dhe dalëse te PLC-së.

Pas selektimit të rreshtit të katërt të tabelës zgjedhim inputet hyrëse të cilat janë të vendosura tek folderi **SM-300** hapim folderin me emrin **DI-300** dhe bëjmë zgjedhjen e inputëve hyrëse sipas të dhënave të moduleve të PLC-së e që është **SM321-DI32XDC24V**

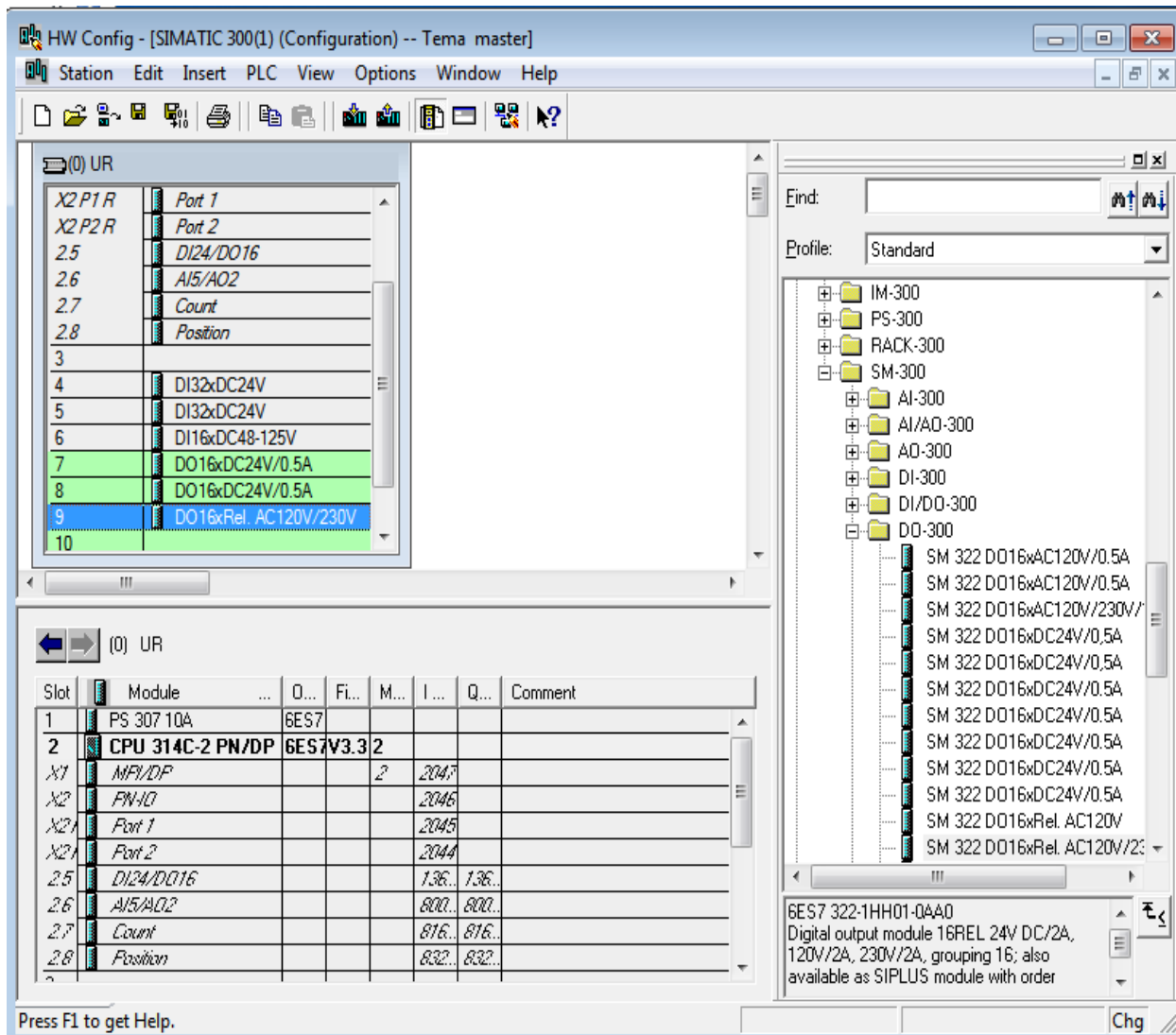


Fig. 3.8 Zgjedhja e inputet hyrse dhe outputet

Sipas tabelës shihet se inputet janë të vendosura në rreshtin e 4,5,6 e që si hyrje apo inpute lexohen vlerat prej I address 0...3 apo adresa hyrëse I 0...3 e deri me 8...9.

Pas vendosjes së inputeve në rreshtin katërt,tani bëhet vendosja e outputeve në rreshtin e 7,8,9 me të dhënave e moduleve të PLC-së, mbesim tek folder SM-300 klikojmë folderin me emrin **DO-300** pastaj klikojmë kodin : **SM322 DO 32xDC24V/0.5A**selektojm me tastin e majtë dy here mbi këtë kod dhe të dhënat barten në rreshtin e 7pastaj ne te njëjtën mënyrë i zgjedhjen edhe dy module tjera automatikisht.

Sipas tabelës shihet se outputet janë të vendosura në rreshtin e pestë e që si dalje output mund të lexohen vlerat prej Q address 12deri 21

Të gjitha këto të dhëna për konfigurimin e PLC-së ruhen duke shtypur ikonin save-compile

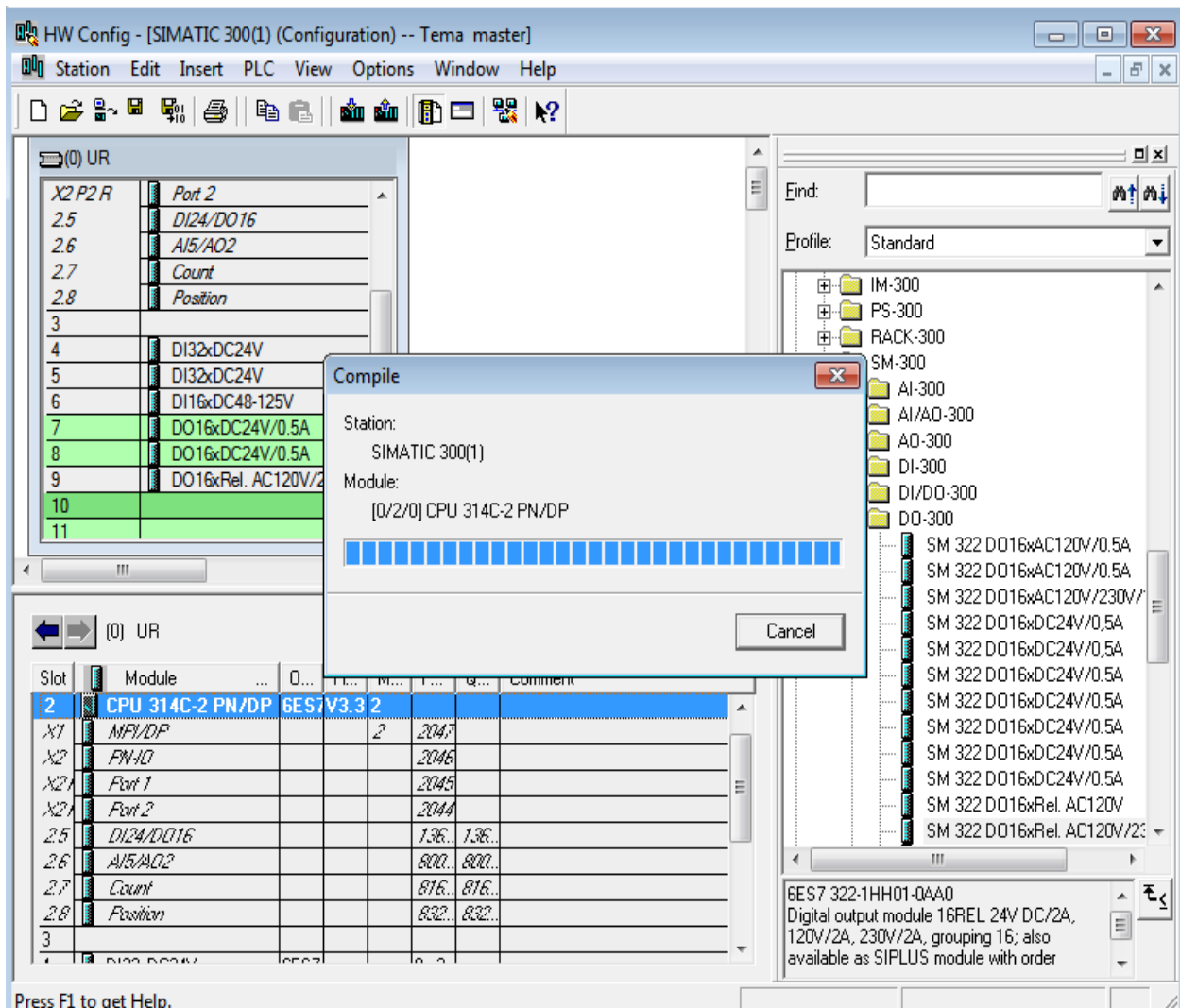


Fig. 3.9 Konfigurimi i CPU-s

Pas përfundimit me të dhënat konfiguruese të CPU-s kemi krijuar mundësinë për programimin e PLC-së, ku në mënyrë automatike duke zbatuar hapat paraprak krijohet blloku kryesor i programimit OB1. Duke pasur parasysh gjithë këtë punë për konfigurim të CPU-s me të dhënat ekzakte, kemi arritur në përfundim që për çfarëdo programimi që neve na nevojitet për realizim në praktikë duhet gjithsesi ti ndjekim hapat paraprak, në këtë rast për programimin e SHIRITIT TRANSPORTUES. Me konfigurimin e CPU-së, janë krijuar të gjitha parakushtet për fillimin e programimit të PLC-së.

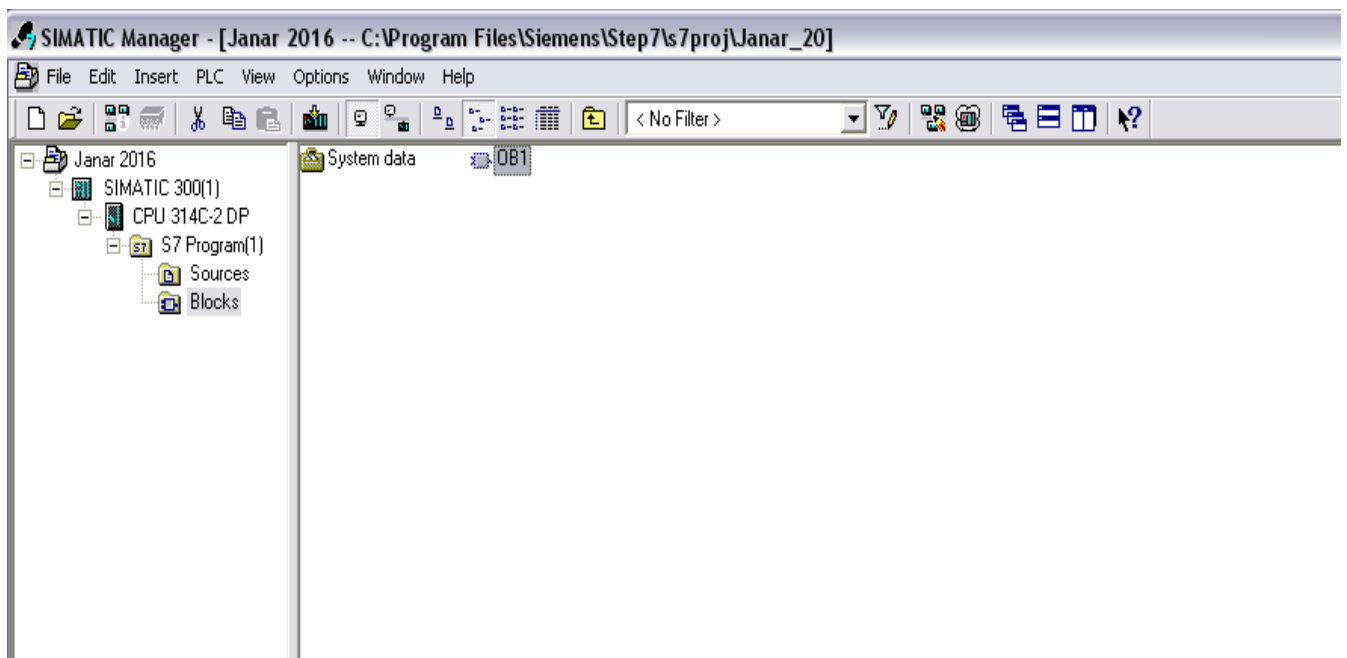


Fig. 3.10 Blloku kryesor i programimit OB1.

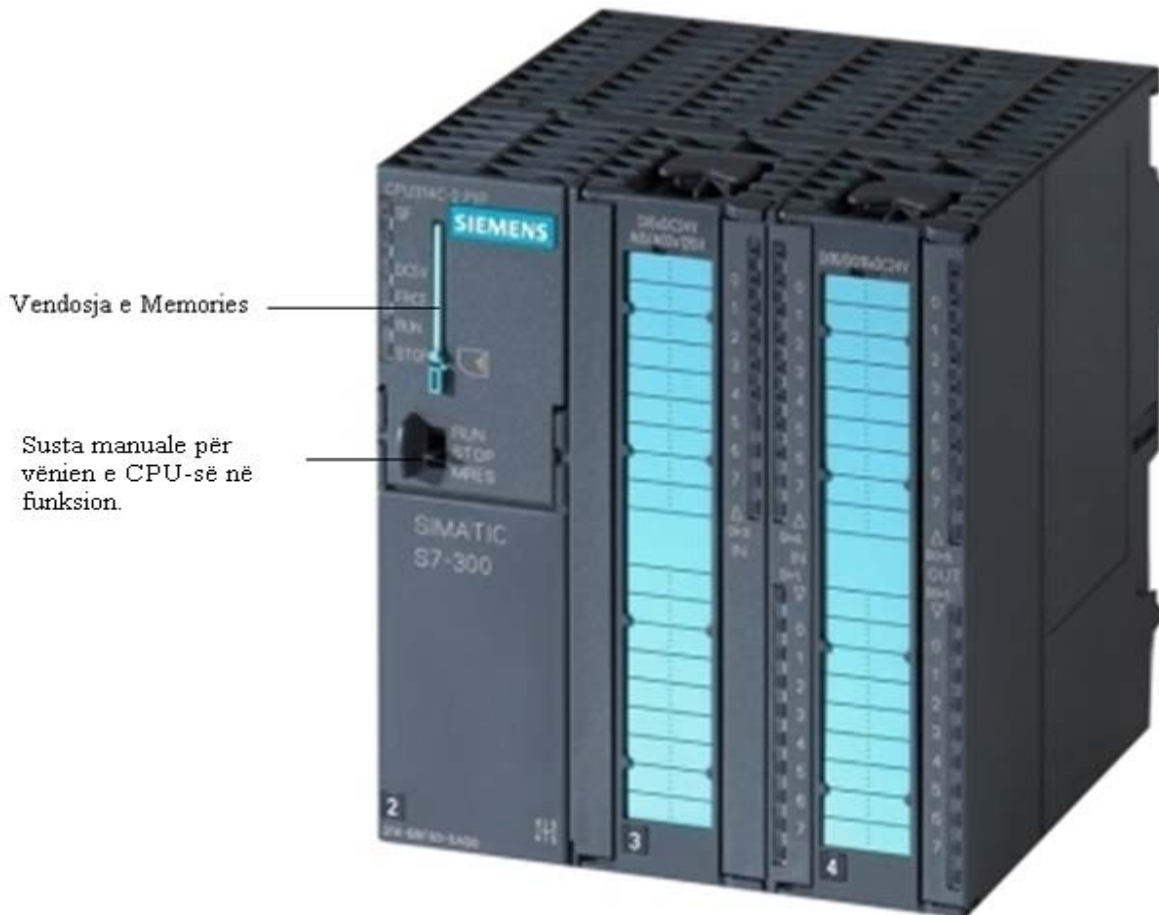


Fig. 3.11 Figura në vazhdim paraqet integrimin e CPU-së dhe PLC-ve, përmes së cilave bëhet komandimi logjik.

Figura në vazhdim paraqet integrimin e CPU-së dhe PLC-ve, përmes së cilave bëhet komandimi logjik, si dhe funksionalizimi i programimit të PLC-së p.sh., në rastin tonë konkret mikserit,presës, shiritave transportues ,prerësit dhe pirunëve .

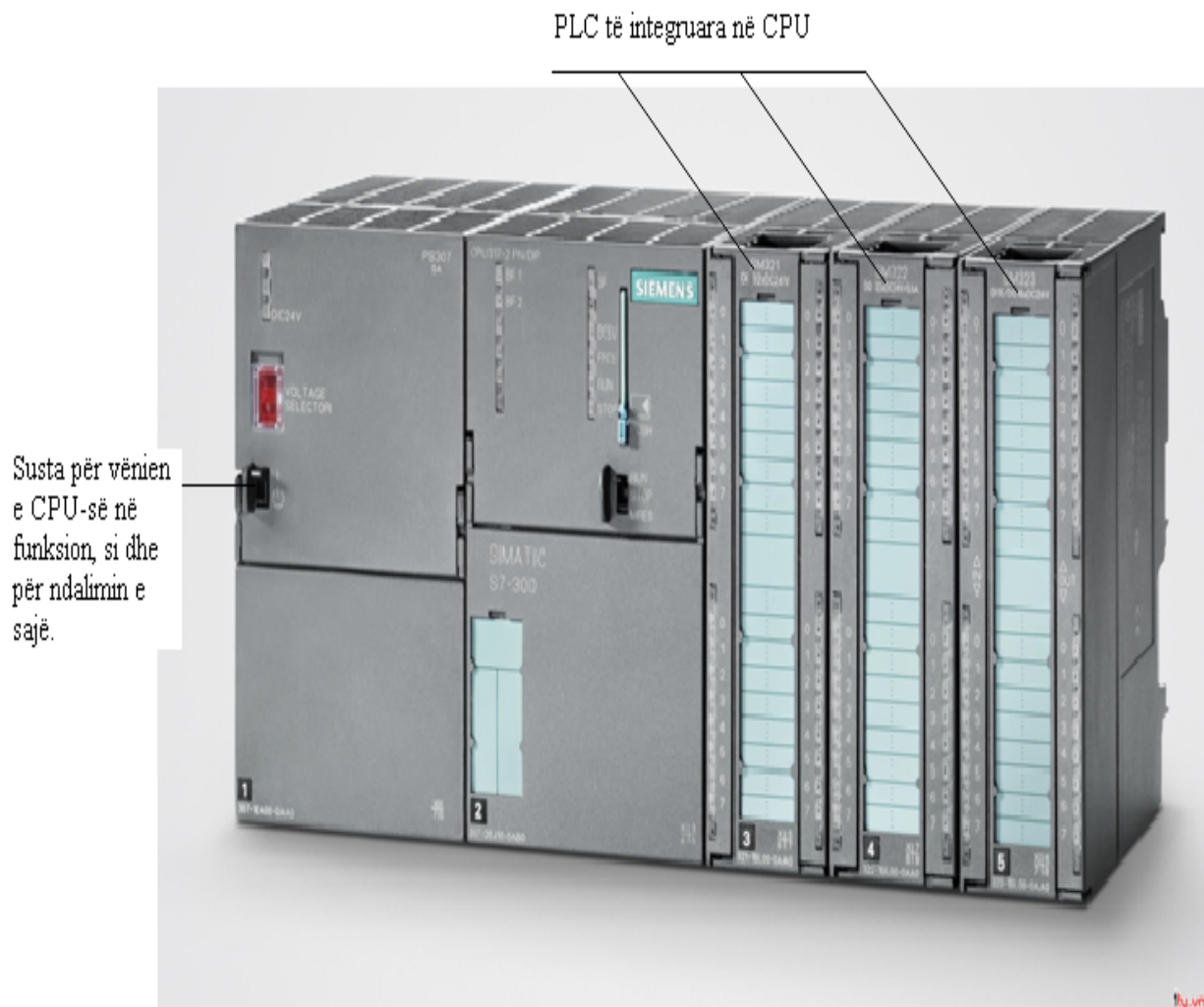


Fig. 3.12 Integrimi i CPU-së dhe PLC-ve, për funksionimin normal të komandimit logjik.

Pajisja e cila shërben për nxjerrjen e programit nga kompjuteri, përpunimin e tij, si dhe futjen në kartë memoriën e cila është e vendosur në CPU. Pra kjo pajisje lë të kuptohet se është mjaft me rëndësi për realizimin e punës programuese.

4.0 Dirigjimi i procesit të prodhimit të blloqeve me ndihmën e PLC-së STEP 7 në fabrikën e bllokove EL-Boksh

Dirigjimi i procesit të prodhimit ka të bëjë me prodhimin e blloqeve nga operacioni i renjes së dheut në shiritin e pare deri te vendosja e bllokave në rrjeta (vagona).

Para se të fillojmë programimin e një PLC duhet ta kemi të qartë se ekzistojnë disa rregulla programore paraprake, ku në qoftë se nuk i zbatojmë ato rregulla pothuajse është e pamundur që të programohet një PLC.

Duke u nisur nga rregullat fillestare për programim, si dhe zbatimi i tyre në përpikmëri, janë krijuar disa blloqe programuese, blloqe të cilat në vete kanë nënbloqe të programuara dhe të integruara njëra me tjetrën, në mënyrë që makinerit të programohet sipas nevojë.

Cikli i programimit është i përbërë nga:

OB1 – Programi kryesor, me nënprograme;FB1,DB1, FB2,DB2,

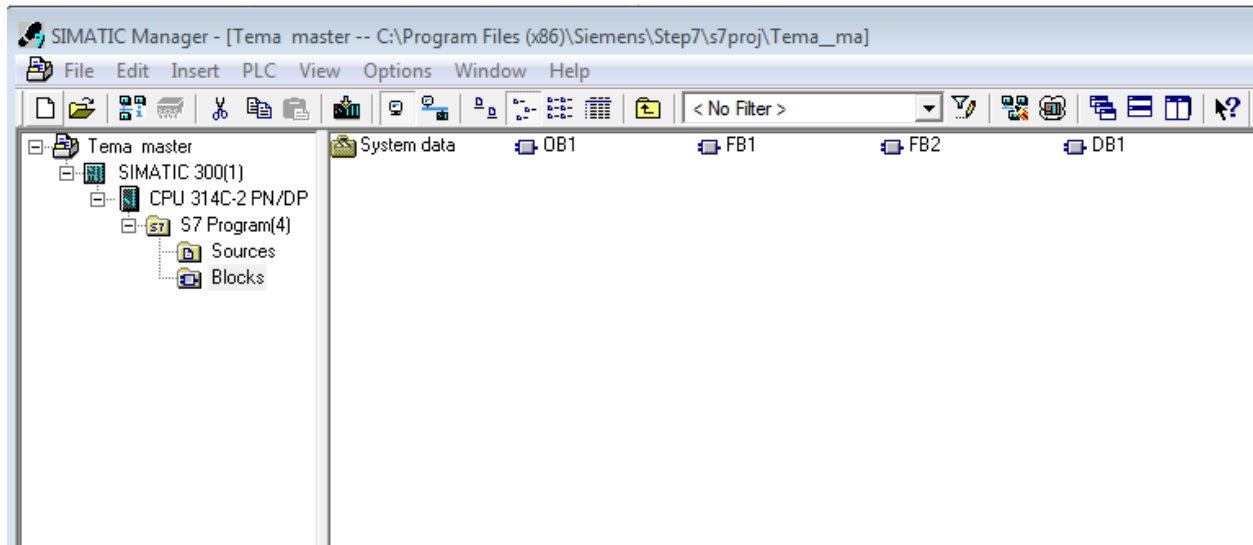


Fig. 4.1 Organizimi i blloqeve, cikli i programimit për makinat prodhuese.

Para se të fillojmë programimin se pare duhet te behet organizimi i sinjaleve hyrëse (digital input) dhe sinjaleve dalëse (digjital output).

Tabela 4.1.Adresat dhe Simbolet e programimit të

Status	Symbol /	Address	Data type	Comment
1	Aktivizimi automatik	I 0.2	BOOL	
2	aktivizimi gjuntos se mi	FC 1	FC 1	
3	aktivizimi me dor	I 3.0	BOOL	
4	dalja	Q 13.0	BOOL	
5	Emergenca	I 0.3	BOOL	
6	FC2	FC 2	FC 2	
7	kyqja e motorrit te prese	Q 13.6	BOOL	
8	kyqja e gjuntos se mixer	Q 13.3	BOOL	
9	kyqja e gjuntos se prese	I 2.6	BOOL	
10	kyqja e motorrit te mixe	I 3.1	BOOL	
11	kyqja e valvules se mixe	I 3.2	BOOL	
12	kyqja e valvules se pres	Q 13.5	BOOL	
13	levizja e pirunave mrapa	Q 13.2	BOOL	
14	levizja e pirunave para	Q 13.1	BOOL	
15	mixeri	Q 12.4	BOOL	
16	paisjet pneumatike	FB 1	FB 1	
17	pistoni i hapur	I 3.4	BOOL	
18	pistoni i mbyllur	I 3.3	BOOL	
19	pjata	Q 12.5	BOOL	
20	prersi	Q 12.7	BOOL	
21	presa	Q 12.6	BOOL	
22	qkuqja e gjuntos se pres	Q 13.4	BOOL	
23	s. ngritjes se pirunave	I 2.0	BOOL	
24	s. i numrimit te bl	I 2.3	BOOL	
25	s. i preses	I 1.3	BOOL	
26	s. i presjonit	I 1.5	BOOL	
27	s. i uljes se pirunave	I 2.1	BOOL	
28	s. minimal	I 1.0	BOOL	
29	s. per aktivizimin e shi	I 2.4	BOOL	
30	s. per ndalimin e preses	I 2.2	BOOL	
31	s.i pjates	I 1.2	BOOL	
32	s.i preses	I 1.4	BOOL	
33	s.i shiritir 1	I 0.4	BOOL	
34	s.i shiritir 2	I 0.5	BOOL	
35	s.i shiritir 3	I 0.6	BOOL	
36	s.i shiritir 4	I 0.7	BOOL	
37	s.i tregimit te bllokave	I 1.7	BOOL	
38	s.i vakumit	I 1.6	BOOL	
39	s.maximal	I 1.1	BOOL	
40	s.mixerit	I 2.7	BOOL	
41	shirti 1	Q 12.0	BOOL	
42	shirti 2	Q 12.1	BOOL	
43	shirti 3	Q 12.2	BOOL	
44	shirti 4	Q 12.3	BOOL	
45	stari 1	I 3.5	BOOL	
46	stari	I 0.0	BOOL	

4.1 EKUACIONET LOGJIKE TË SISTEMIT TË DIRIGJIMIT PËR PËRPUNIMIN E BLOQEVE TË ARGJILËS

Procesi teknologjik për përpunimin e blloqeve të argjilës është si vijon:

Fillon prej rënjes së dheut në shiritin e parë (I) dhe bartet në mikser në të cilin bëhet lagja dhe përzierja. Gjatë rrotullimit të mikserit materiali i përzier gjuhet në pjatë, nga pjata kalon në shiritin transportues II, shiriti transportues II e bartë në presë, ku bëhet përzierja dhe nxjerrja e vakumit. Pas nxjerrjes së vakumit fillon me dalë brumi i argjilës nëpërmes kallupit në shiritin para prerësit. Në pajisjen para prerësit është i vendosur enkoderi që bënë matjen e gjatësisë, e cila gjatësi i përgjigjet dy blloqeve. Tani fillon operacioni i prerjes nga dy blloqe. Blloqet e prerë kalonë në shiritin III në të cilin bëhet rrotullimi, dhe pastaj kalojnë në shiritin IV. Procesi i tillë vazhdon në shiritin IV duke i formu dy rende nga 6 (gjashtë) blloqe. Pasë këtij operacioni me ndihmën e aktuatorit ngriten këto dy rende në lartësinë e caktuar dhe pastaj bëhet lëvizja e pirunjeve para dhe i lënë blloqet në rrjetë (vagon) dhe pajisja kthehet mbrapa. Procesi i tillë vazhdon deri sa mbush vagonin.

Në bazë të këtij procesi teknologjik kemi nxjerr ekuacionet logjike si në vijim

$$Q_{13.0} = (I_{0.0} + I_{0.1}) \times (\bar{I}_{0.4} \times \bar{I}_{2.7} \times \bar{I}_{0.5} \times \bar{I}_{1.4} \times \bar{I}_{1.5} \times \bar{I}_{1.6} \times \bar{I}_{0.3})$$

$$Q_{13.6} = Q_{13.0} \times I_{3.0\text{manual}} + I_{0.2\text{automatic}} \times I_{0.3\text{emergjenca}} \times Q_{13.0}$$

$$Q_{12.1} = (I_{3.0\text{manual}} + I_{0.2\text{automatic}}) \times \bar{I}_{0.1\text{stopi}}$$

$$Q_{12.5} = (I_{3.0manual} + I_{0.2automatik}) \times \bar{I}_{0.1stopi}$$

$$Q_{13.4} = (\bar{I}_{3.0manual} \times Q_{13.0} + I_{0.2automatik} \times Q_{13.0} \times I_{emergjenca}) \times \bar{I}_{0.1stopi}$$

$$Q_{12.0} = (I_{3.0manual} \times Q_{13.0} + I_{0.2automatik}) \times \bar{I}_{0.1stopi} \times (I_{1.1s.max} + I_{1.0s.min})$$

$$Q_{12.7} = I_{3.0manual} \times Q_{13.0} \times \bar{I}_{0.1stopi} + I_{0.2automatik} \times Q_{13.0} \times M_{0.0}$$

$$Q_{12.2}^+ = I_{3.0manual} \times Q_{13.0} + I_{0.2automatik} \times M_{0.0}$$

$$Q_{13.1} = (I_{3.0manual} \times Q_{13.0}) + (I_{0.2automatik} \times Q_{13.0} \times I_{2.3} \times \bar{I}_{0.1stopi})$$

$$Q_{13.2} = (I_{3.0manual} \times Q_{13.0}) + (I_{0.2automatik} \times Q_{13.0} \times I_{3.4} \times \bar{I}_{0.1stopi})$$

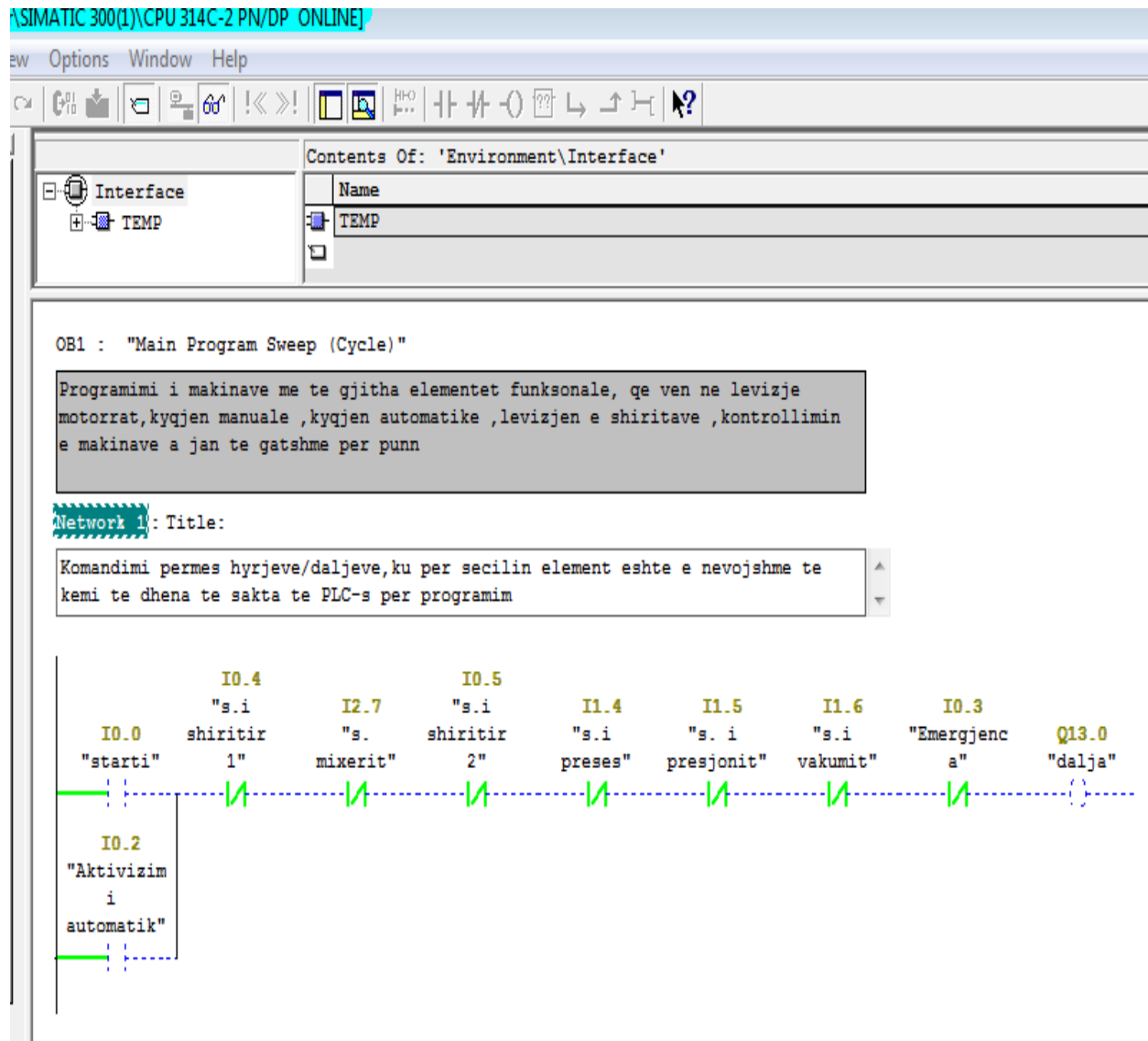


Fig. 4.2 Programimi i makinave me të gjitha elementet funksionale në gjendje të qetë

Në figurë kemi paraqitur skemen loghike të makinave, startin, sensorin e shiritit 1, sensorin e shiritit 2, sensorin e preses, sensorin e presit, sensorin e vakumit dhe daljen, ku me aktivizimin e imputit I 0.0 është aktivizu dalja .

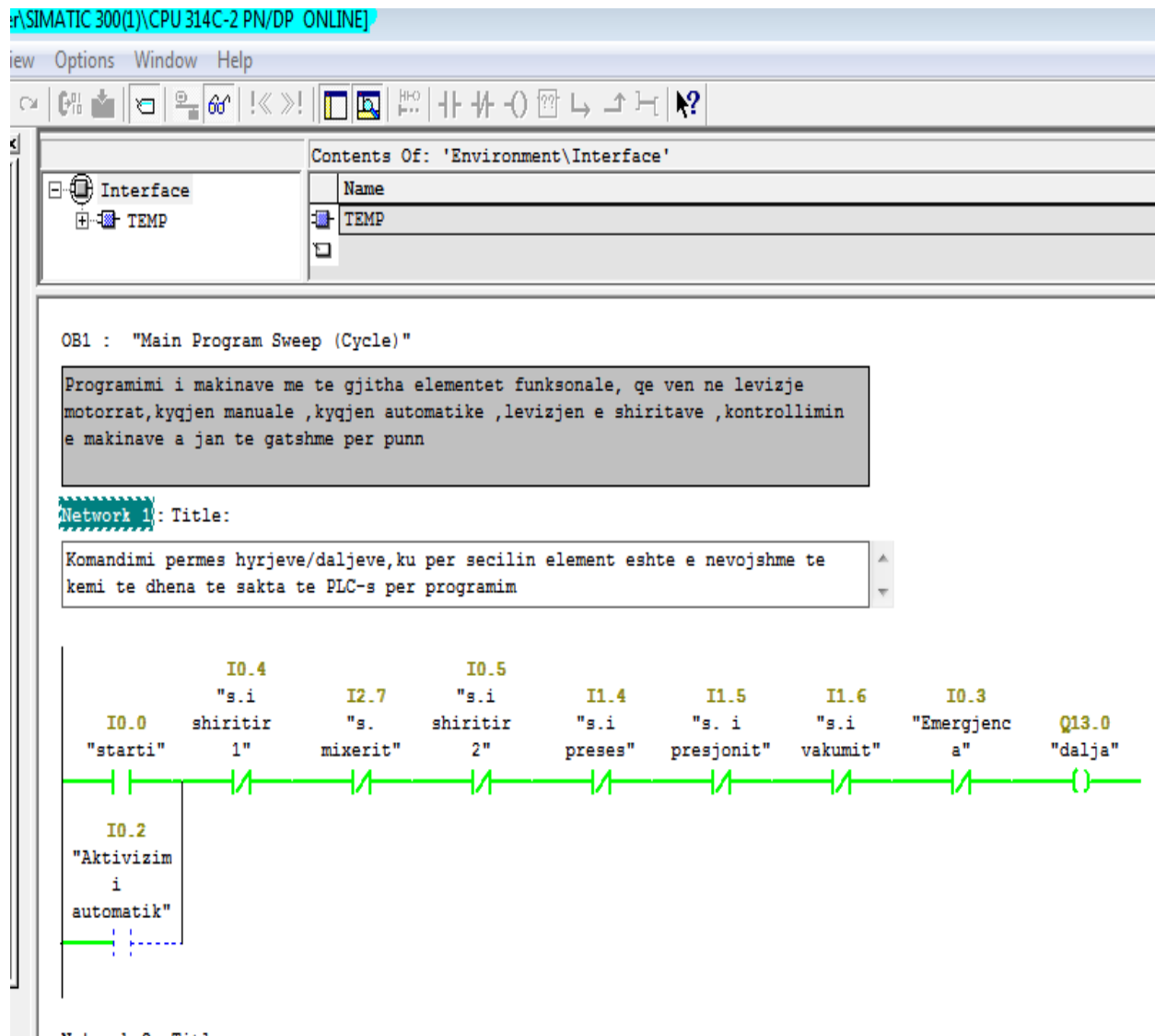
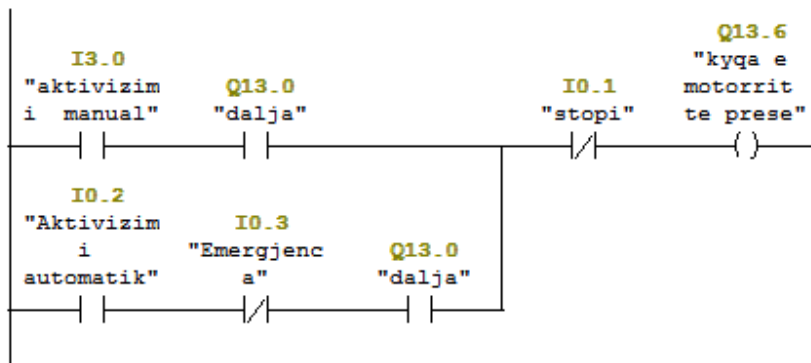
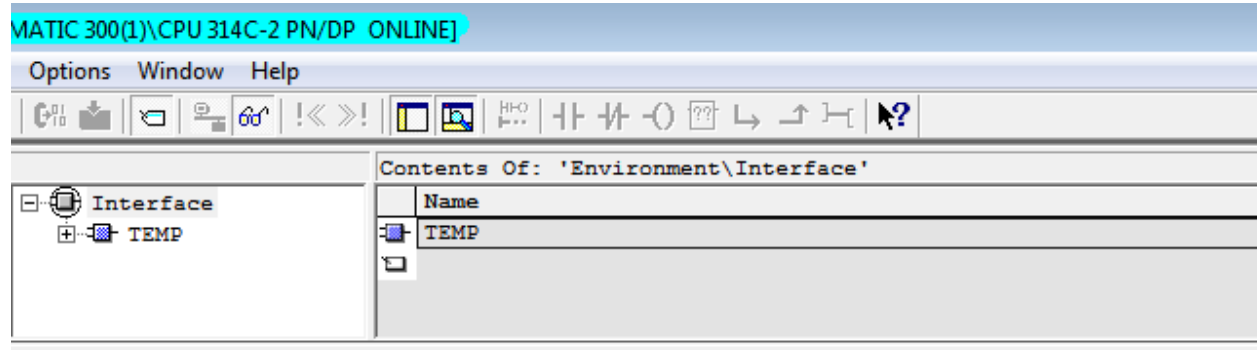


Fig. 4.3 Programimi i makinave me të gjitha elementet funksionale, me aktivizimin e imputit I0.0 ku është aktivizuar i gjith sistemi makinerik



Network 3: Title:

Comment:

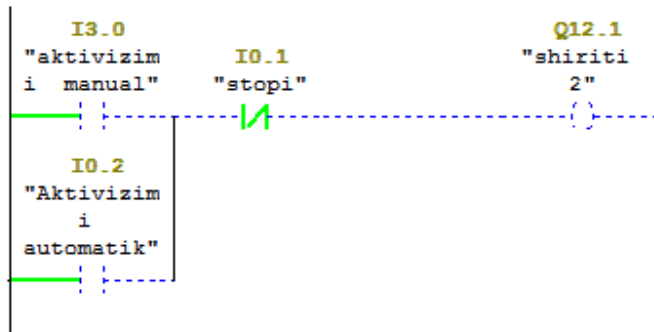


Fig. 4.4 Në figurë kemi paraqitur sinjalet për startimin e motorit të preses dhe të motorit të shiritit të parë

Në figurë kemi paraqit mikserin ku përmban dy mostra në të cilën njeri shërben përpirjen e argjilës dhe tjetri për bartjen e argjilës në shiritin transportues, mikseri ka sensorin minimal dhe maksimal qëshërben për tregimin e gjendjes së mikserit .



Fig.4.5 Mikser për perzirjen e argjillës

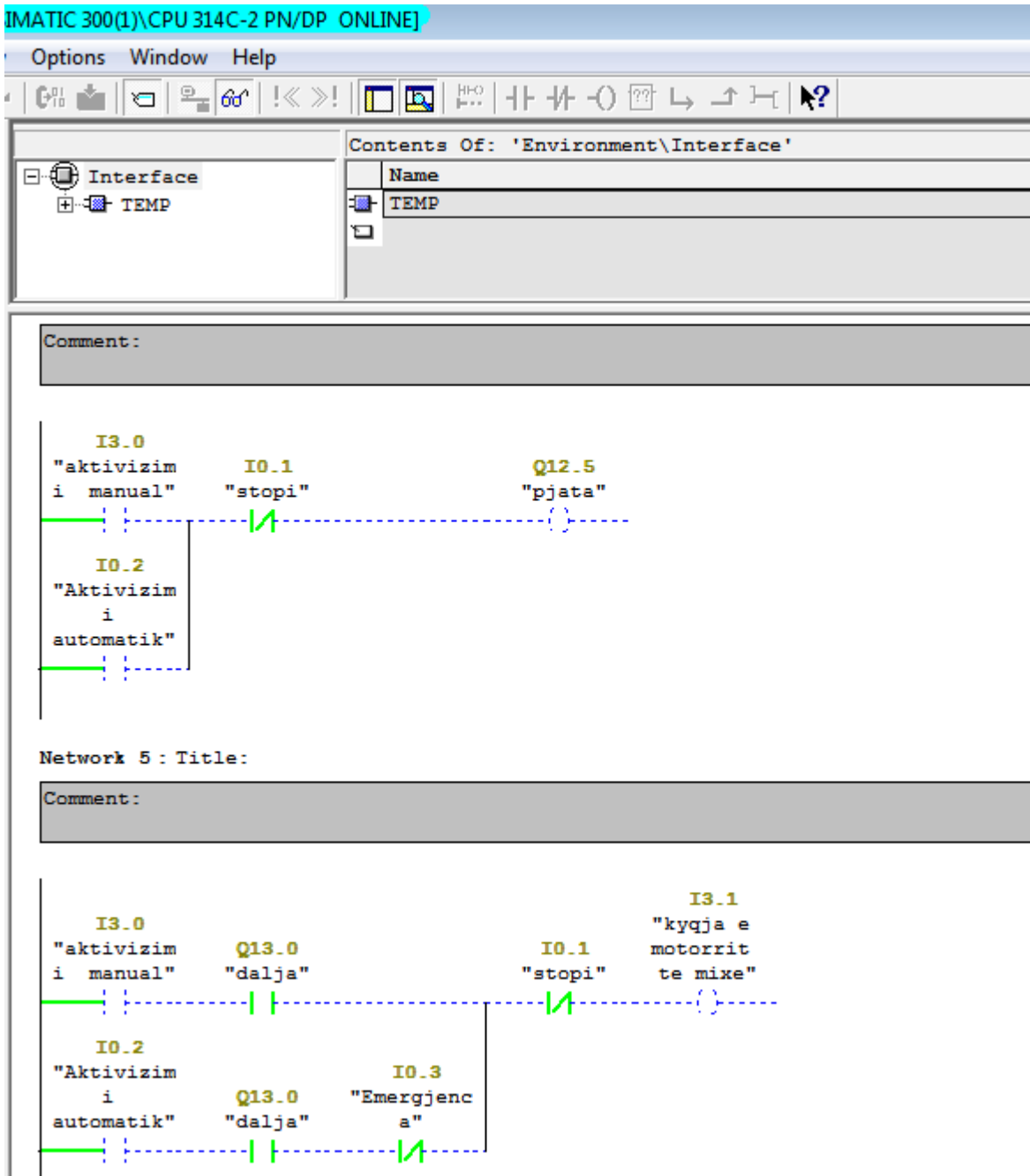


Fig. 4.6 Në figurë kemi paraqitur sinjalet për startimin e motorit të pjatës dhe të motorit të mikserit.

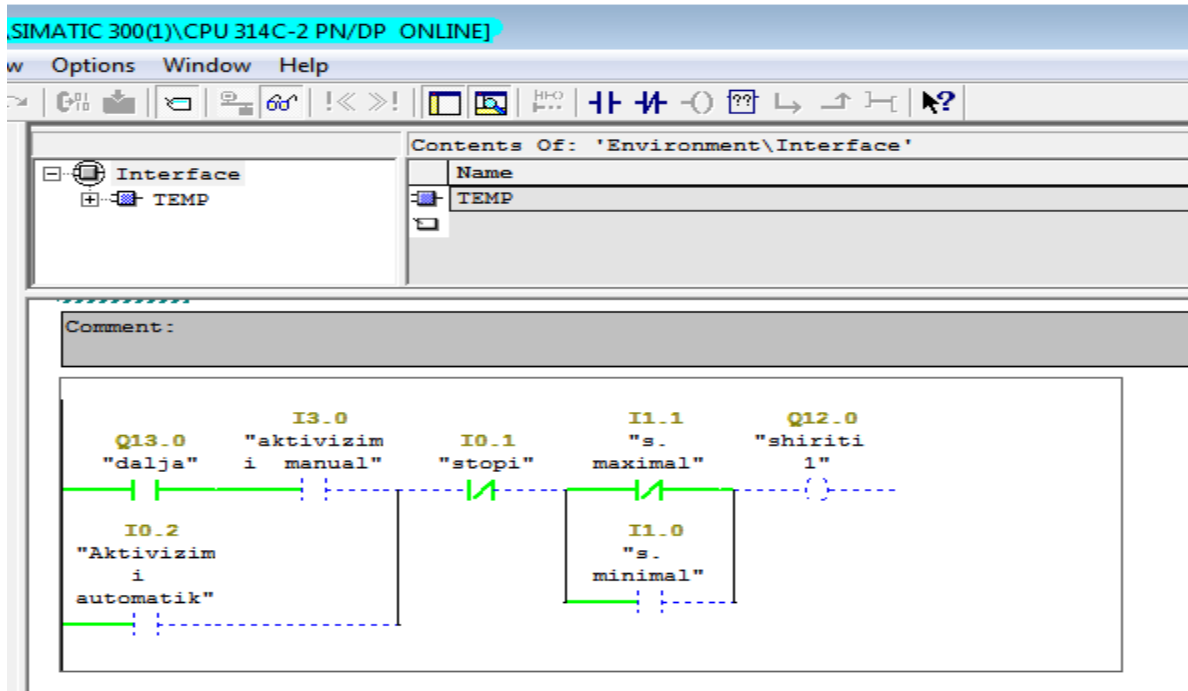


Fig. 4.7 Shiriti transportus në qetsi

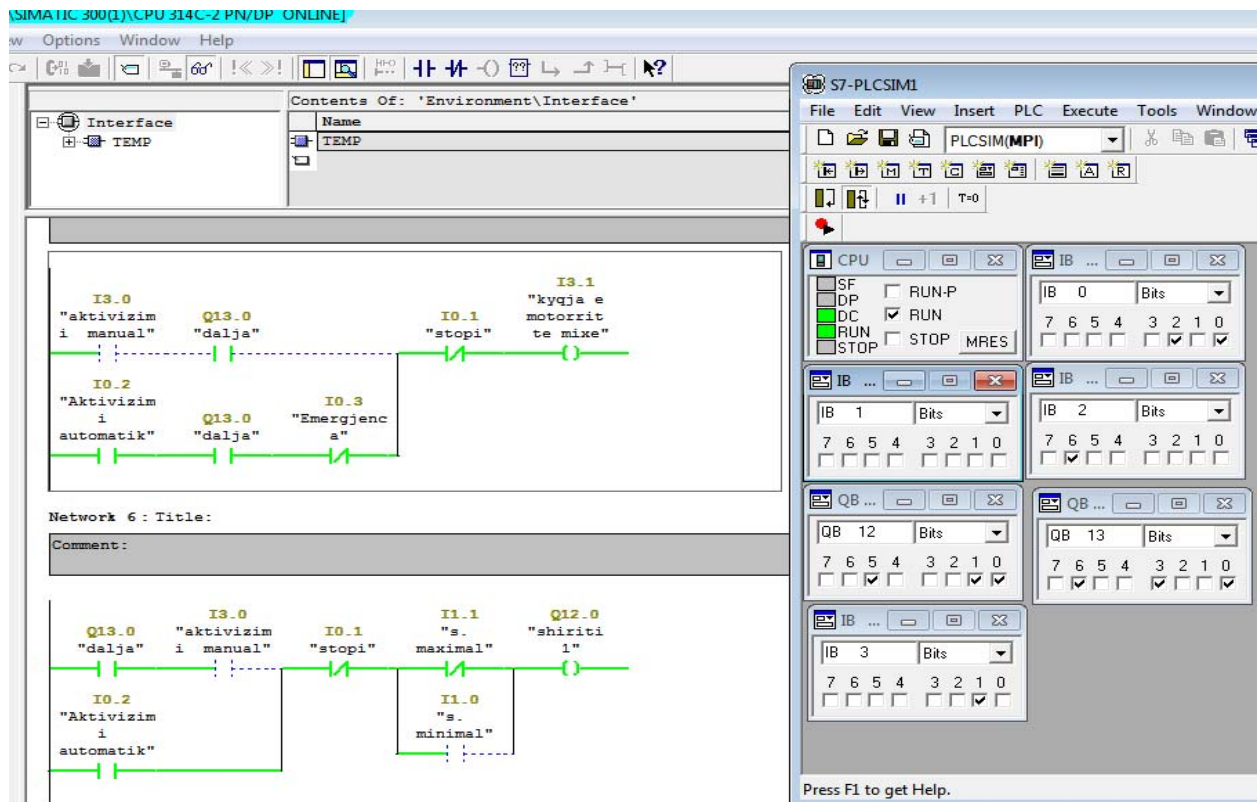


Fig. 4.8 Kyqja e motorit të mikserit dhe shiritit transportues

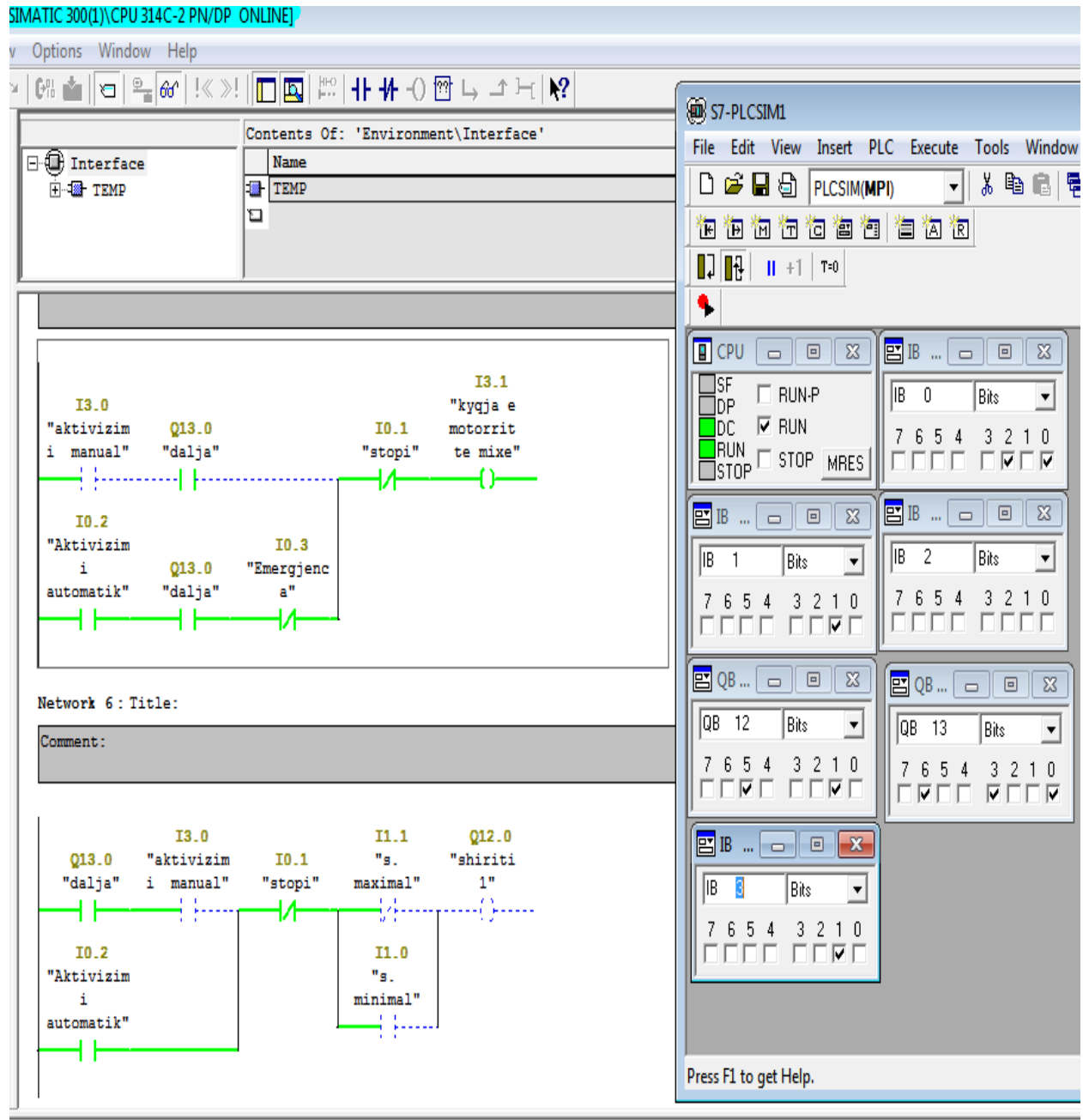


Fig.4.9 Në skemen logjike shihet mikseri në punë dhe shiriti i ndalur për shkak se ka reaguar senzori maksimal.

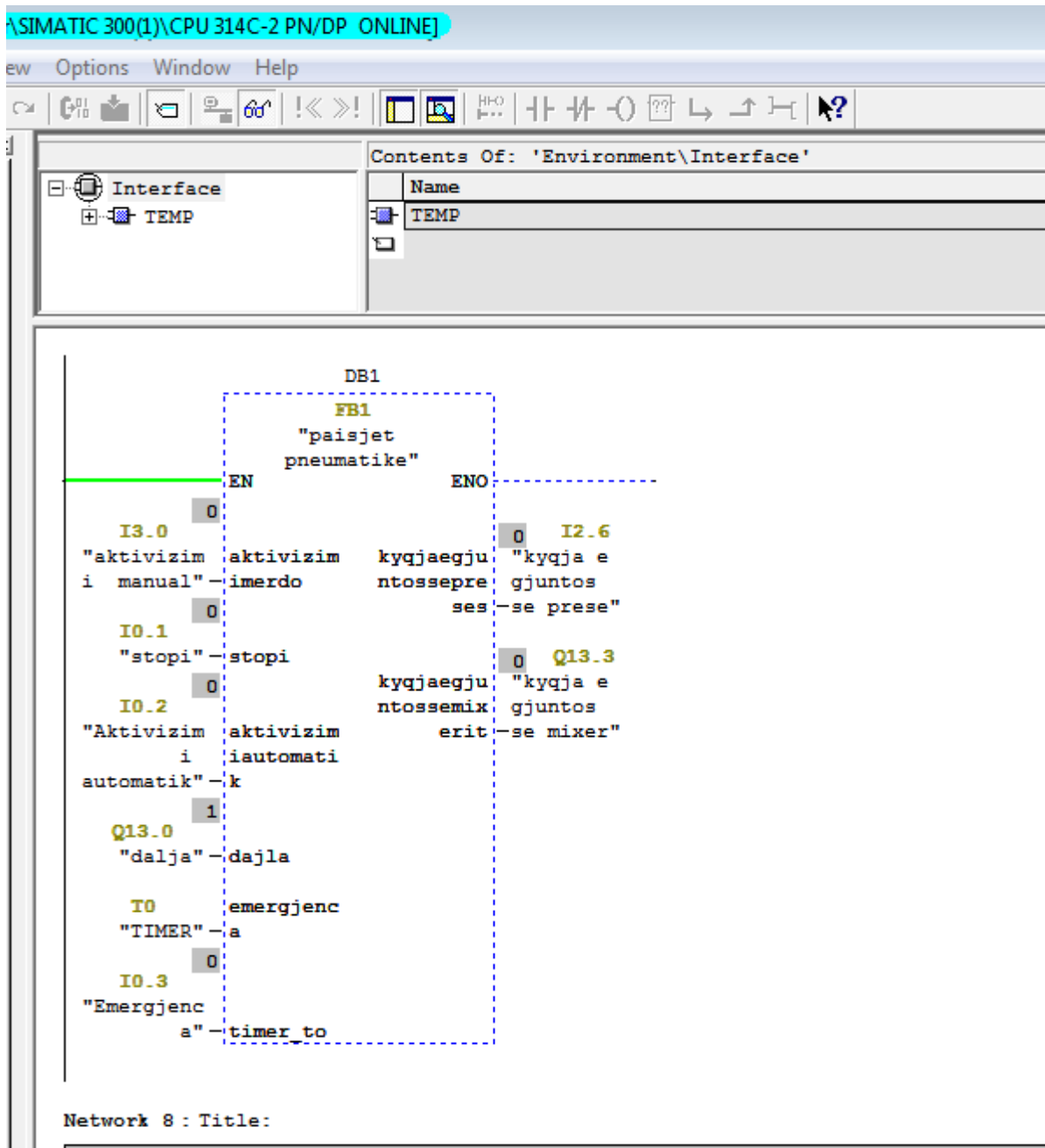


Fig.4.10 Blloku FB1 për aktivizimin e gjuntos të mikserit dhe presës

Në figur kemi parqitur prersi i cili bën prerjen e bllokut dhe motorri i prersit është servomotoor që komandohet nga simodraivi, në bazë të impulseve që lëshon enkoderi prerësi pret blloka përmes mekanizmit manivel.



Fig. 4.11 Prersi për prerjen e Bllokave

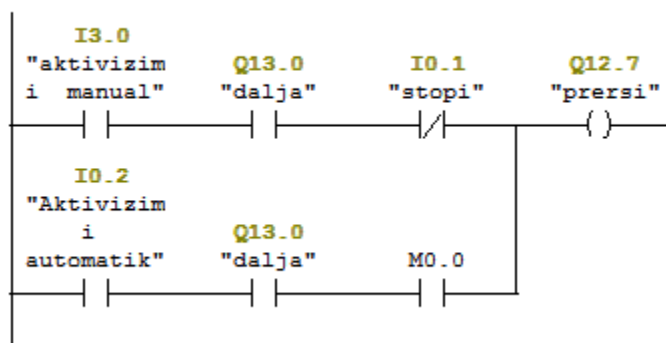


Fig. 4.12 Skema e programimit të prersit për prerjen e bllokove

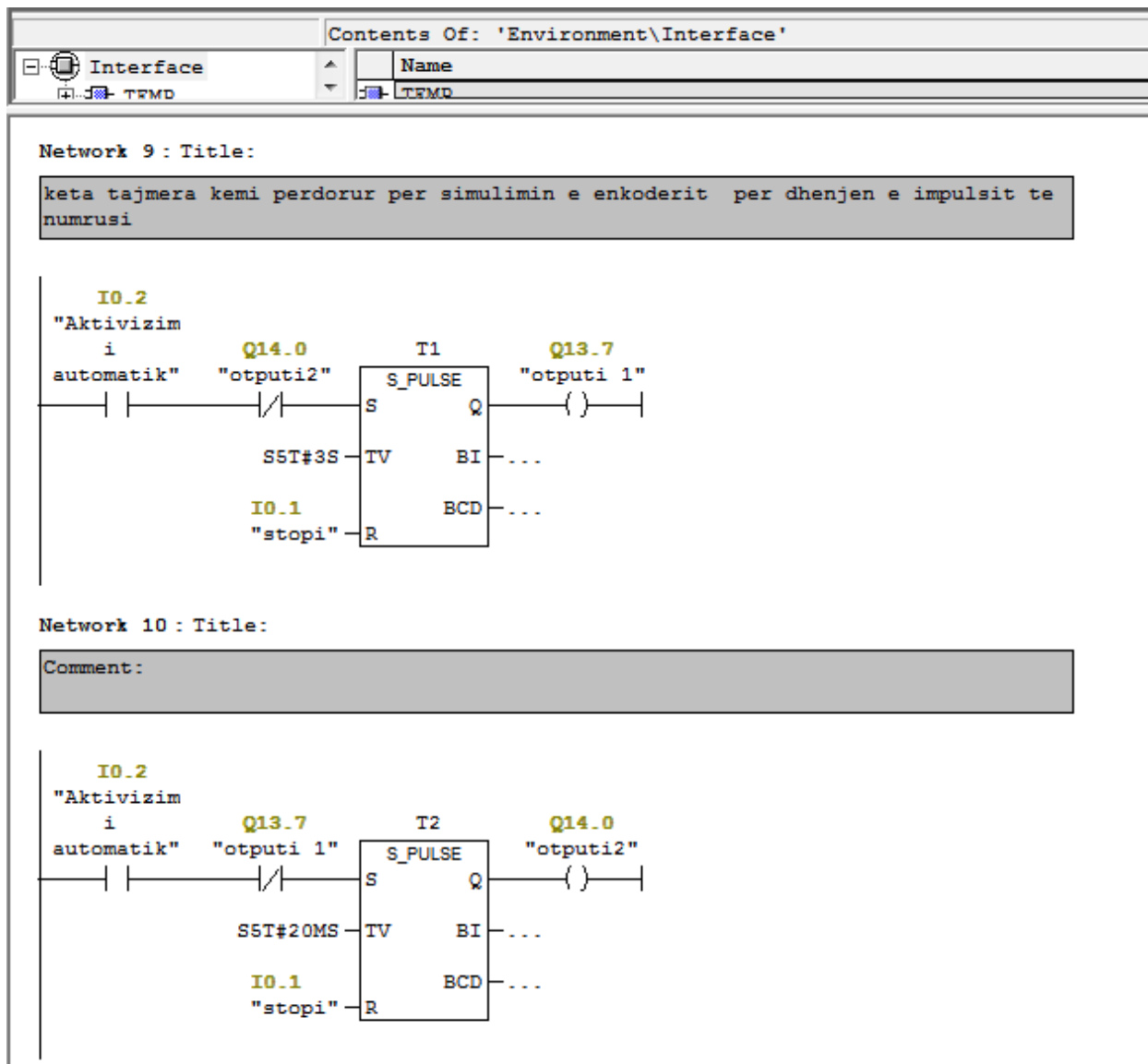


Fig. 4.13 Tajmerat për simulimin e enkoderit

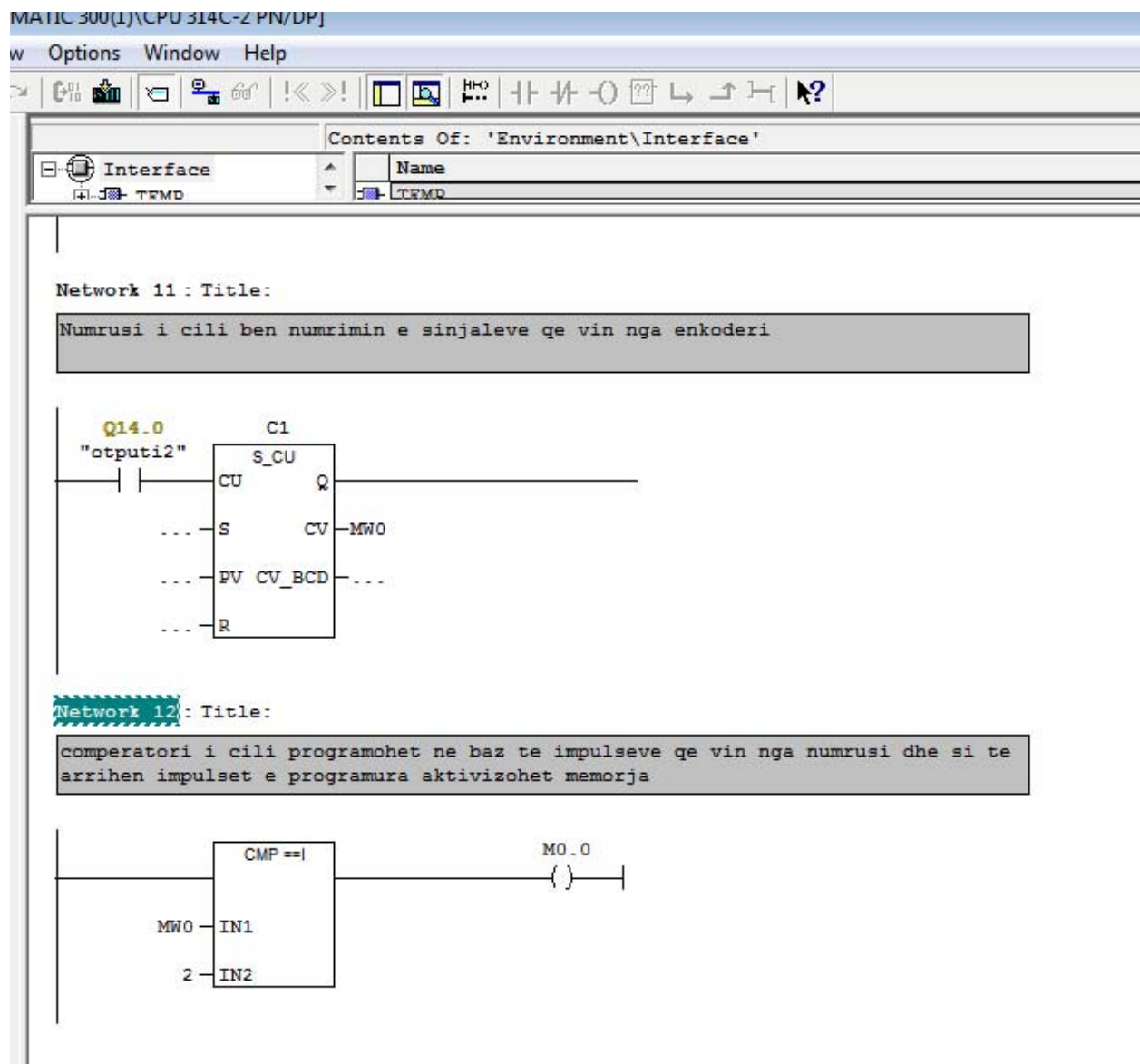


Fig. 4.14 Numrusi dhe memorja për aktivizimin e prerësit

Ne figurë kemi paraqitur presën që e bënë presimin e argjilës dhe e bartë deri te prerësi pastaj prerësi e bënë prerjen e bllokove dhe i kalon në shiritin e transportues



Fig. 4.15 Shiriti transportues për bartjen e bllokove

Network 13 : Title:

Comment:

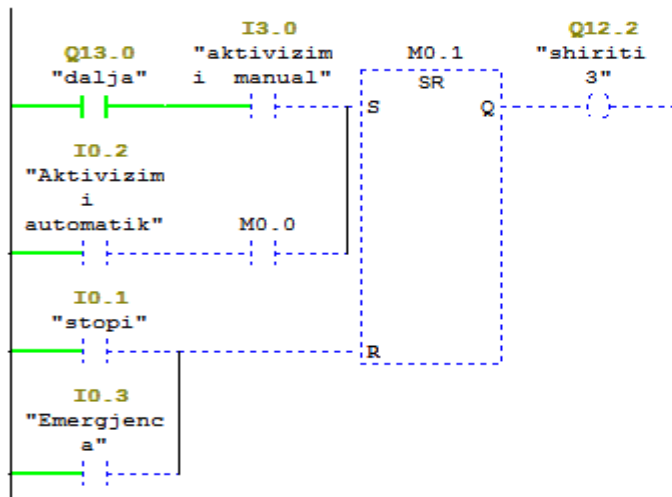


Fig. 4.16 Shiriti transportues në qetësi

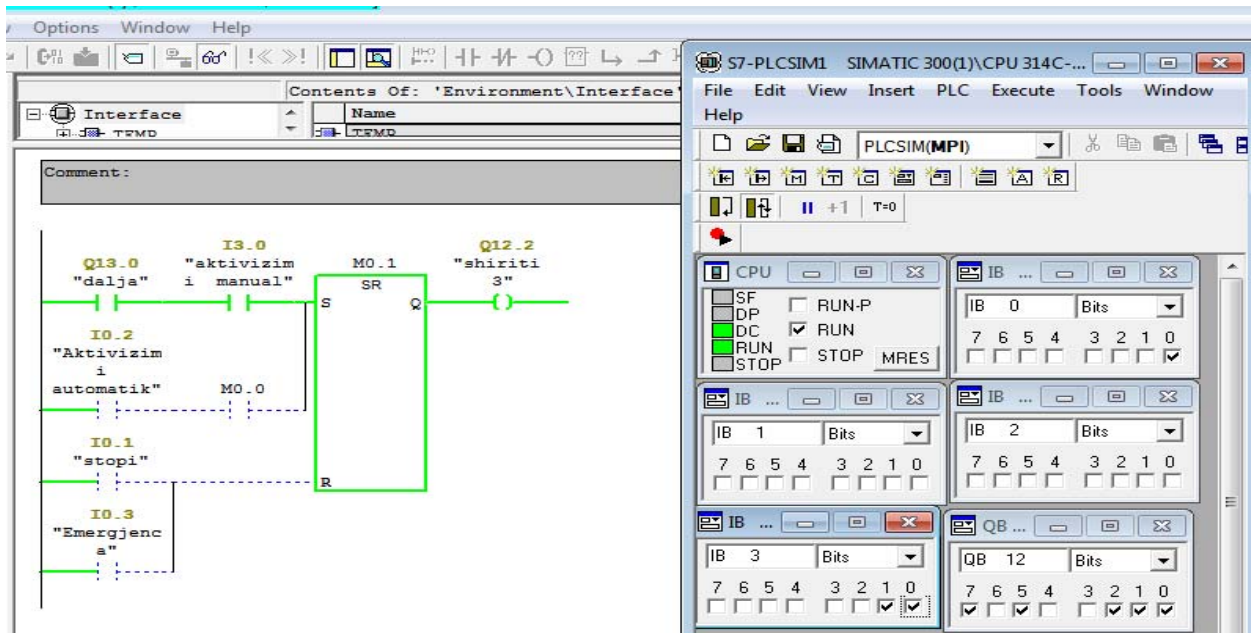


Fig. 4.17 Aktivizimi i shiritit transportues manual

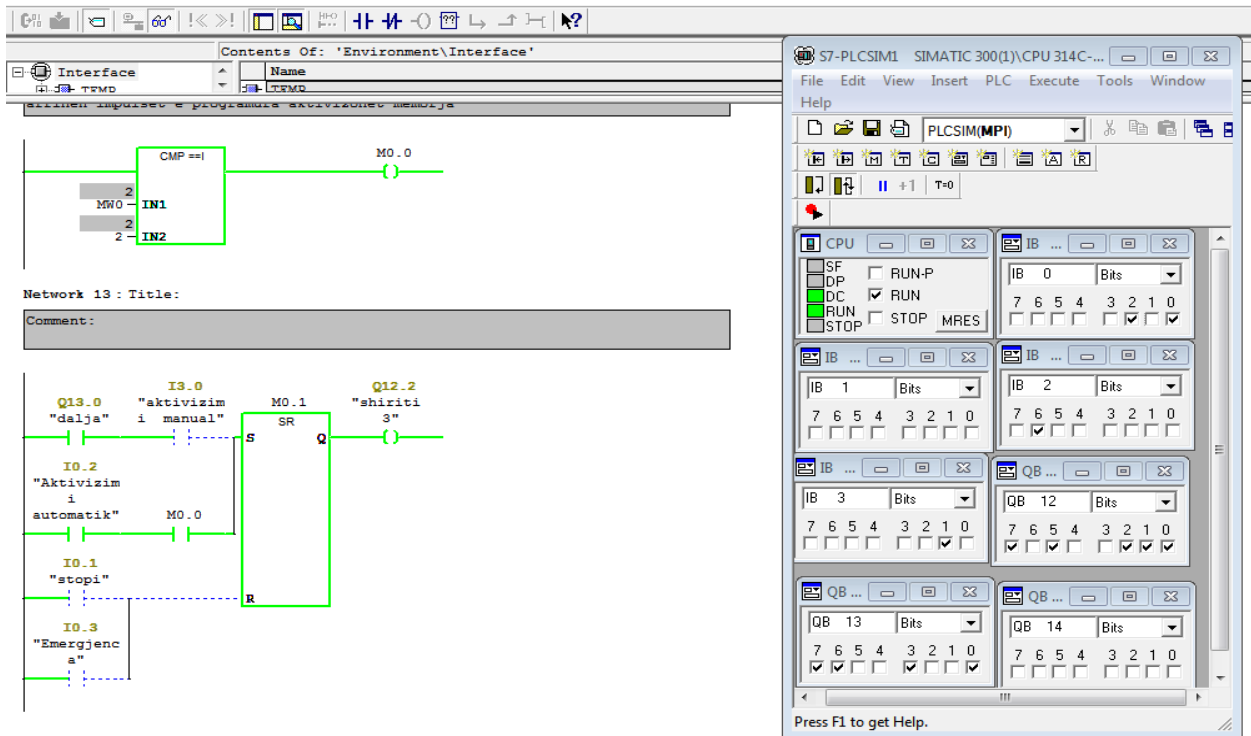


Fig. 4.18 Aktivizimi i shiritit transportues automatik

Në figurë kemi paraqitur shiritin e katërt që bën bartjen e bllokove deri të senzori që është në fund të transportierit .



Fig. 4.19 Shiriti i katërt për bartjen e bllokove

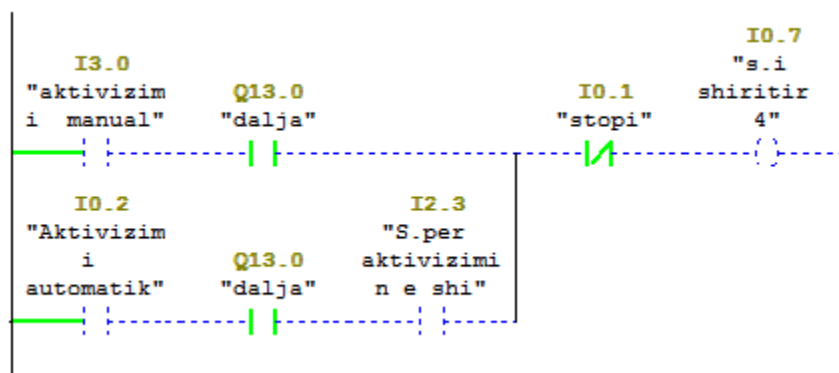


Fig.4.20 Shiriti transportues në qetësi

Skema logjike e shiritit transportues ku është bërë aktivizimi në mënyrë manuale

Network 14 : Title:

Comment:

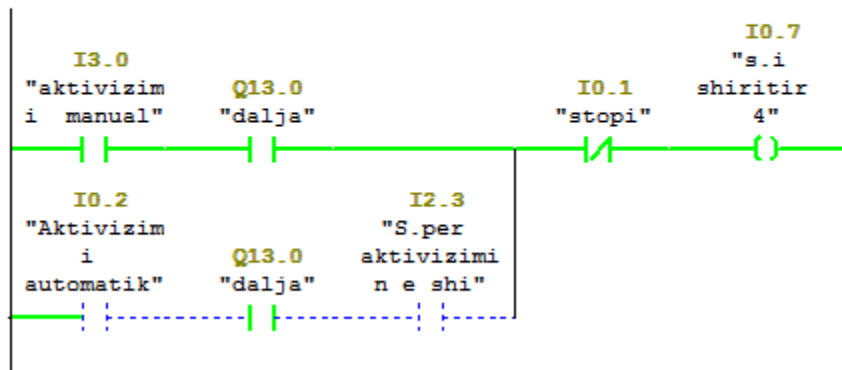


Fig. 4.21 Skema për aktivizimin manual

The screenshot shows the SIMATIC Manager interface. On the left, the 'Environment\Interface' tree is visible. The main window displays the same ladder logic diagram as in Fig. 4.21, but with a different network title: 'Network 15 : Title:'. On the right, the 'S7-PLCSIM1' status window is open, showing the CPU status. The CPU is in 'RUN' mode, indicated by a green bar. The 'IB' (Input Bits) window shows bit 4 is active (checked), which corresponds to the 's.i shiritir 4' output in the diagram.

Fig. 4.22 Skema logjike për aktivizimin e shiritit në mënyrë automatike

Skema paraqet lëvizjen e pirunëve para ku e bën vendosjen e bllokove në rrjet



Fig.4.23 Pirunët për vendosjen e bllokove në rrjet

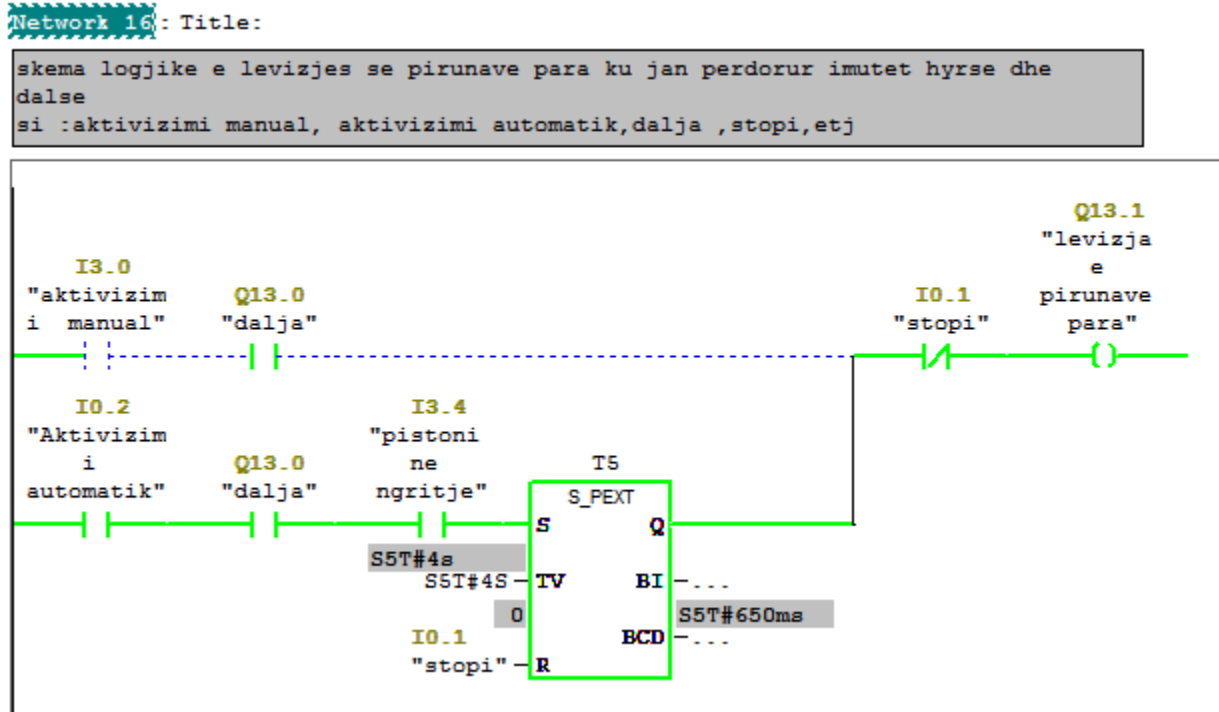


Fig.4.24 Skema logjike për lëvizjen e pirunëve para



Fig.4.25 Ulja e pirunëve dhe lëvizja mbrapa

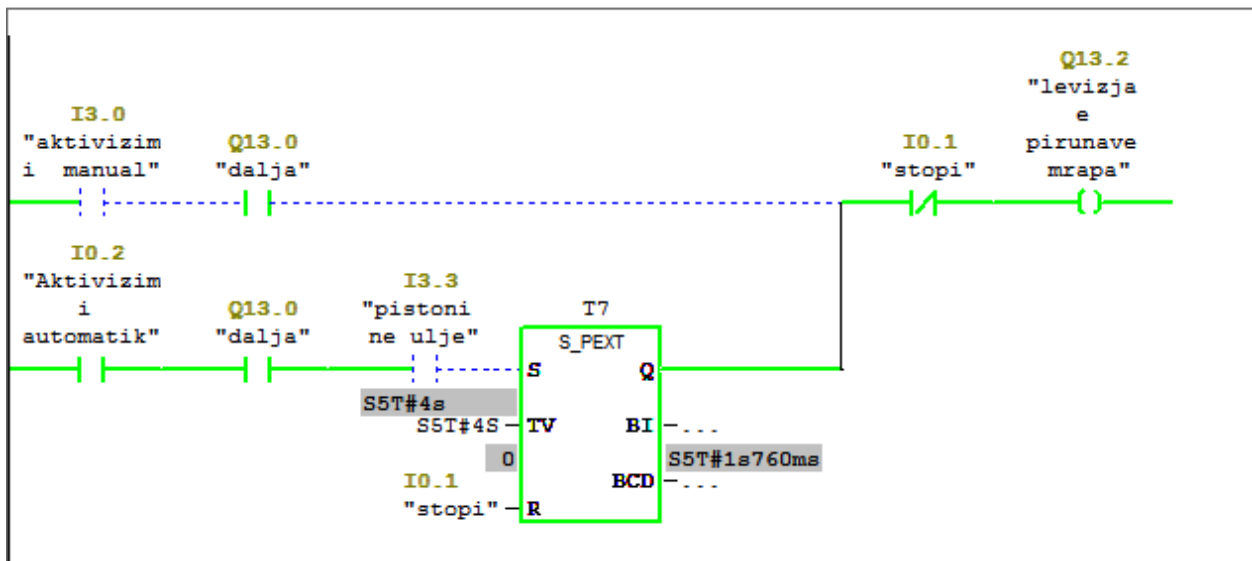


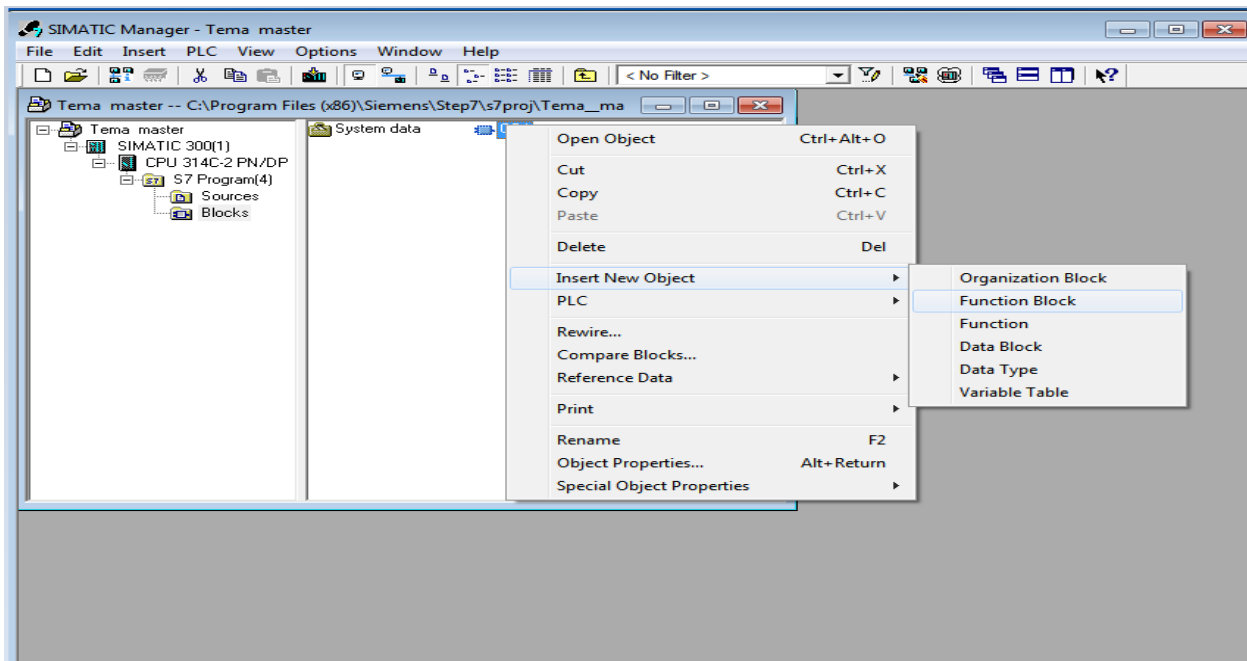
Fig. 4.26 Skema logjike e lëvizjes së pirunëve mbrapa

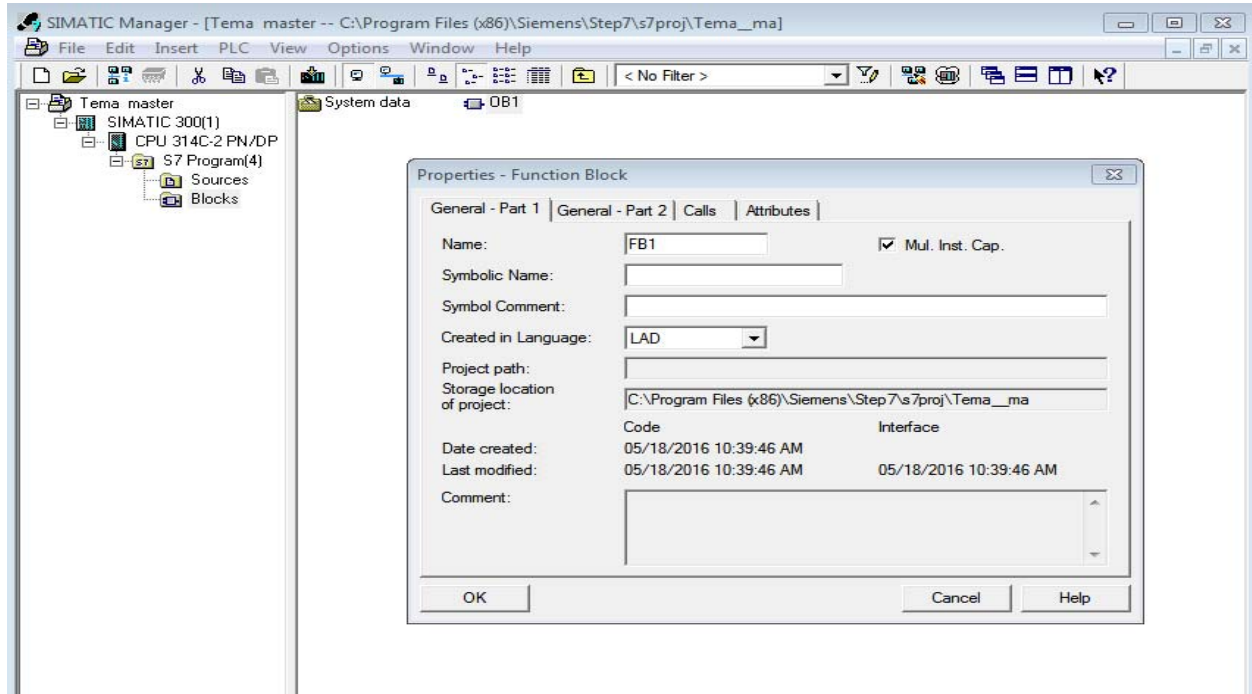


4.2 BLOK FUNKSIONI DHE BLOKU I TË DHËNAVE (FB DHE DB)

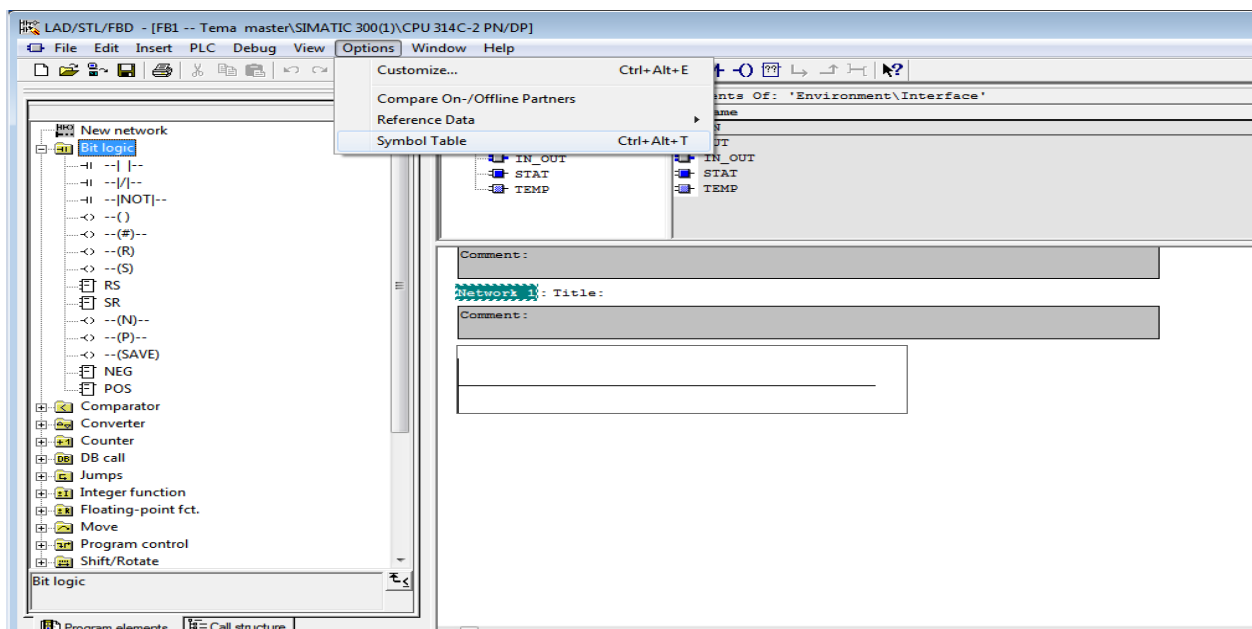
Një shembull i bllokut të dhënave është caktuar për secilin bllok të thirrjes së funksionit, që transferon parametrat. Duke e quajtur më shumë se një shembull të një blloku FB, ne mund të kontrollojmë më shumë se një pajisje me një FB. Një bllok FB për një lloj motori, p.sh., mund të kontrollojë motorë të ndryshëm duke përdorur grupe të ndryshme të të dhënave të shembujve të ndryshëm për çdo motor. Të dhënat për secilin motor (p.sh., shpejtësia, akumulatori për kohë operuese etj.) mund të ruhen në një ose më shumë blloqe të DBs.

Në udhëzimet e më poshtme është bërë krijimi i bllok-ut FB1 shtypim tastin e djathtë të mausit pastaj shkojmë Insert New Object zgjedhim Function Block hapët dritarja Properties Function Blockpastaj zgjedhim gjuhën programuese LAD.





Regjistrimi i vlerave të (input-ve dhe output-ve. kohëmatësit...etj) të gjitha mundësit që ka në dëshirojmë të programojmë dhe i regjistrojmë sikur vlera hyrëse apo dalëse e cila behët në atë mënyrë që çdo element i cili hynë në gjuhën programuese të regjistrohet në tabelënsiç shihet tani në këtë tabelëregjistrueseklikojmë të menya e shiritit lart Options>Symbol Table.



4.3 CAKTIMI I VLERAVE FILLESTARE TË PARAMETRAVE FORMALE.

Ne mund të caktojmë vlerat fillestare të parametrave formale në pjesën e deklarimit të bllokut FB. Këto vlera janë të shkruara në shembullin DB të ruajtura në bllokun FB. Nëse nuk i caktojmë parametrat aktualë në parametrat formal për formulimin e thirrjes, STEP 7 përdorë vlerat e ruajtura në shembullin DB. Këto vlera mund të jenë vlera fillestare që janë future në tabelën e variablave të deklaruara në një FB.

Në kuadër të block-ut FB1 është bërë programimi për kyçen dhe qkyqjen e gjuntos te presës dhe te mikserit me gjuhen programuese LAD

The screenshot shows the Siemens STEP 7 software interface for editing a function block (FB1). The title bar indicates the project path: LAD/STL/FBD - [FB1 -- "paisjet pneumatike" -- Tema master\SIMATIC 300(1)\CPU 314C-2 PN/DP\...\FB1].

The parameter table at the top lists the following variables:

Name	Data Type	Address	Initial Value	Exclusion add
aktiviz...	Bool	0.0	FALSE	<input type="checkbox"/>
stopi	Bool	0.1	FALSE	<input type="checkbox"/>

FB1 : Title:
Në gjuhen programuse LAD kemi paraqitur kyqjen dhe qkyqjen e gjuntos se preses dhe mikserit .

Network 1 : Title:
Në këtë rrjet kemi bër programimin për startimin manual të gjuntos se preses ku është perdorur butoni për kyqje me manuale , dalja dhe stopi

```

#aktivizim      #kyqjaegju
  imanual      ntossepres
#aktivizim      es
  imanual      #kyqjaegju
#dajla          ntossepres
#dajla          es
#stopi          rit
  #stopi       rit
  )
    
```

Network 2 : Title:
Po të njeten rrjet për startimin manual të gjuntos së mikserit kemi bër programimin ku është perdorur butoni për kyqje me manuale , dalja dhe stopi

```

#aktivizim      #kyqjaegju
  imanual      ntossemixe
#aktivizim      rit
  imanual      #kyqjaegju
#dajla          ntossemixe
#dajla          rit
#stopi          rit
  #stopi       rit
  )
    
```

Fig. 4.27 Aktivizimi manual i gjuntos se presës dhe mikserit

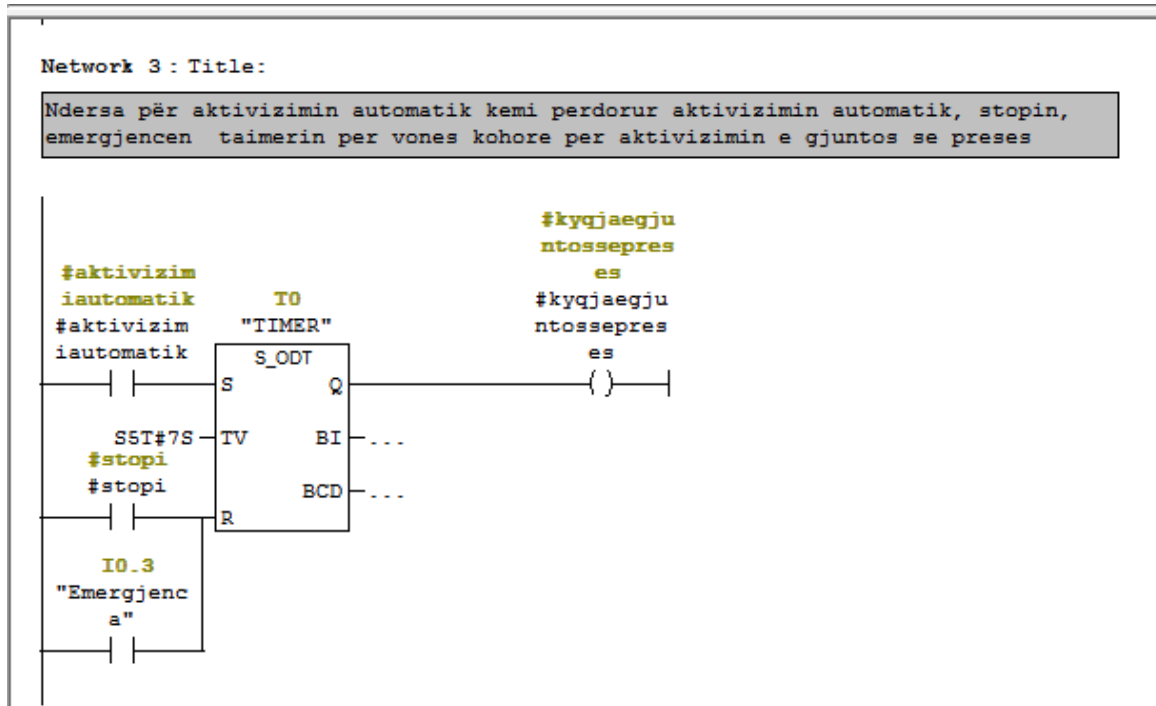


Fig.4.28 Aktivizimi automatik i gjuntos se presës

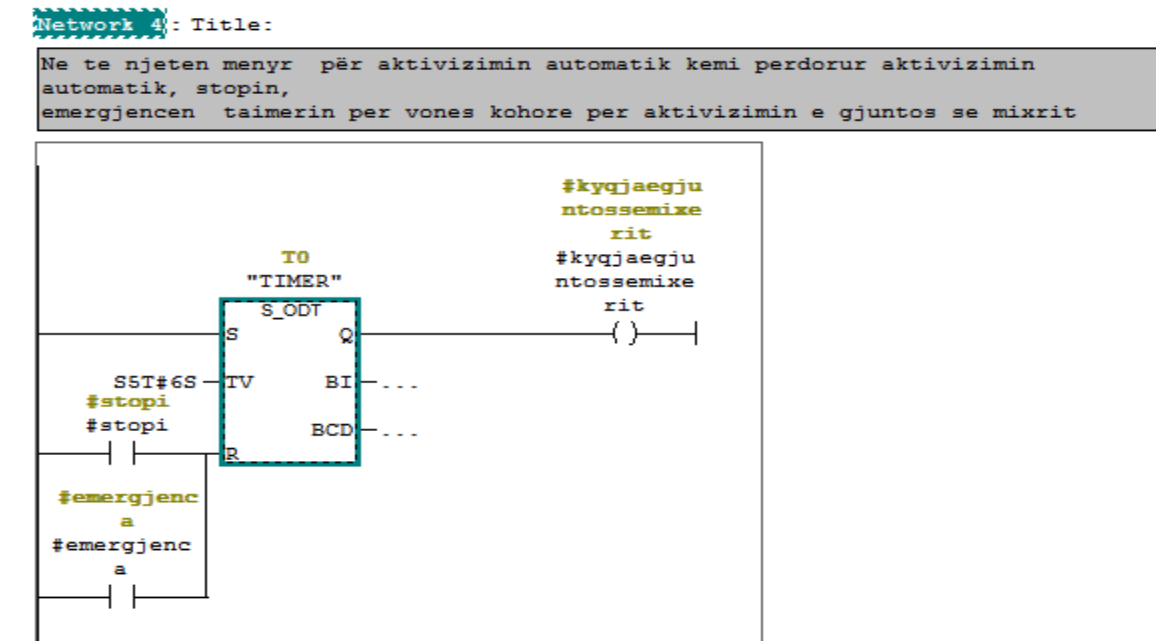


Fig.4.29 Aktivizimi automatik i gjuntos se presës

Tabela 4.1 Paraqitja e të dhënave hyrëse-IN

ke" -- Tema master\SIMATIC 300(1)\CPU 314C-2 PN/DP\...\FB1]

w Options Window Help

Contents Of: 'Environment\Interface\IN'

Name	Data Type	Address	Initial Value	Exclusion a
aktiviz...	Bool	0.0	FALSE	<input type="checkbox"/>
stopi	Bool	0.1	FALSE	<input type="checkbox"/>
aktiviz...	Bool	0.2	FALSE	<input type="checkbox"/>
dajla	Bool	0.3	FALSE	<input type="checkbox"/>
emergjenca	Timer	2.0		<input type="checkbox"/>
timer_to	Bool	4.0	FALSE	<input type="checkbox"/>

Tabela 4.2 Paraqitja e të dhënave dalëse-OUT

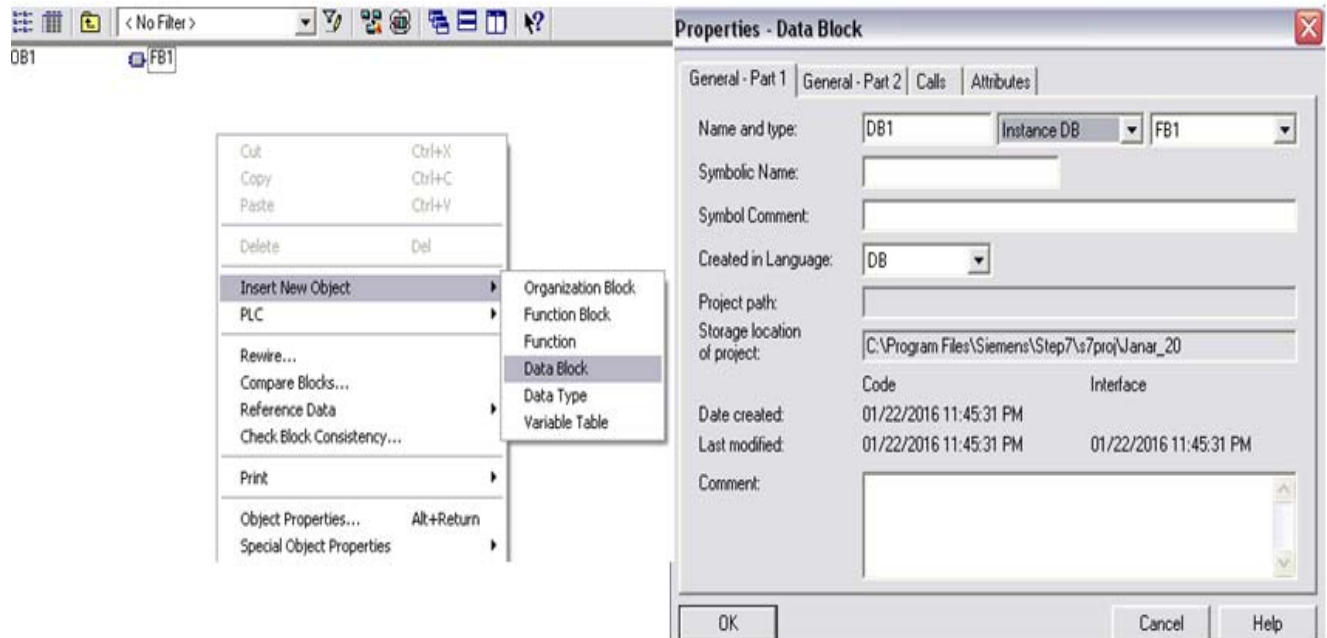
ke" -- Tema master\SIMATIC 300(1)\CPU 314C-2 PN/DP\...\FB1]

w Options Window Help

Contents Of: 'Environment\Interface\OUT'

Name	Data Type	Address	Initial Value	Exclusion address
kyqjaeg...	Bool	6.0	FALSE	<input type="checkbox"/>
kyqjaeg...	Bool	6.1	FALSE	<input type="checkbox"/>

Këto të dhëna nga Tab.4.1 dhe Tab.4.2 regjistrohen tek lidhja Interfacenë IN & OUT ku këto të dhëna regjistrohen apo barten tek OB1 n'kuadër të FB1.



DB1 (Data Block) funksioni i DB1 e luan rolin integruar apo ndërlidhës të FB-së pasi që elementet apo informatat të cilat gjenden në FB ato duhen të barten patjetër në DB kusht është që gjatë integrimit të nënbloqeve paraprakisht të kryhet ky veprim ndërmjet DB1 dhe FB1.

Tabela 4.3 DB1 bartja e të dhënave nga FB1 në DB1

The image shows the 'DB Param - DB1' window. The title bar reads 'DB1 -- Tema master\SIMATIC 300(1)\CPU 314C-2 PN/DP'. The window contains a table with the following data:

	Address	Declaration	Name	Type	Initial value	Actual valu	Comment
1	0.0	in	aktiviz...	BOOL	FALSE	FALSE	
2	0.1	in	stopi	BOOL	FALSE	FALSE	
3	0.2	in	aktiviz...	BOOL	FALSE	FALSE	
4	0.3	in	dajla	BOOL	FALSE	FALSE	
5	2.0	in	emerg...	TIMER	T 0	T 0	
6	4.0	in	timer_to	BOOL	FALSE	FALSE	
7	6.0	out	kyqja...	BOOL	FALSE	FALSE	
8	6.1	out	kyqja...	BOOL	FALSE	FALSE	

5.0 PROGAMIMI DHE INTEGRIMI I NËNBLLOQEVE FUNKSIONALE (FB1& DB1 NË FC1)

Block-u FC1 është block-u në të cilin është integruar block-u FB1 dhe me të dhënat të DB1 integrimi i blloqeve bëhet në atë mënyre që të minimizohen blloqet ku me pas të gjitha blloqet duhet të integrohen në Block-un kryesor të OB1.

Një bllok FC në vete përmban një sektor programi, që ekzekutohet gjithmonë kur FC thirret nga një bllok tjetër logjik.

Funksionet gjithashtu mund t'i përdorim për qëllimet e mëposhtme:

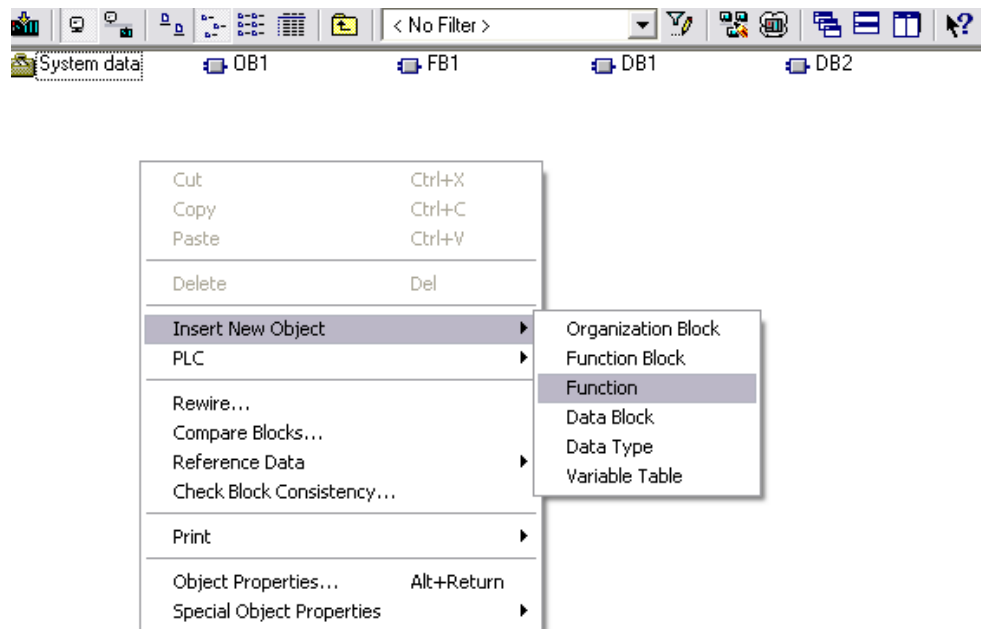
- Kthimin e një vlere funksioni të quajtur bllok (shembull: funksion matematik)
- Ekzekutimin e një funksioni teknologjik (shembull: funksioni i kontrollit të vetëm me një operacioni logjik).

Caktimi i parametrave aktual te parametrat formal.

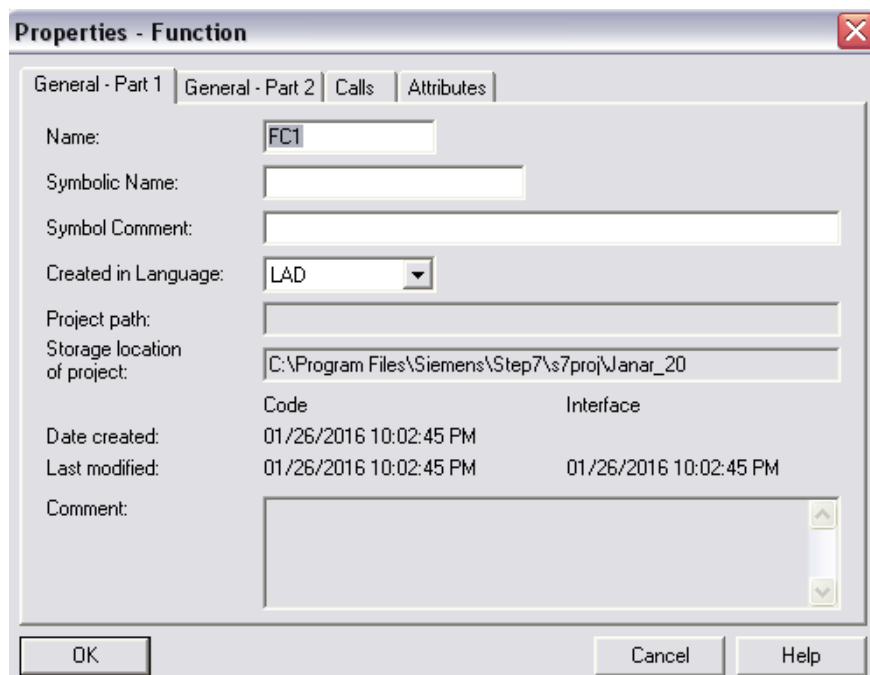
Një parametër formal është një maket për parametrin aktual. Parametrat aktual i zëvendësojnë parametrat formal kurë funksioni është thirrur. Gjithmonë duhet të caktohen parametrat aktual te parametrat formal të një funksioni FC (p.sh., një parametër aktual "I 3.6" te parametri formal "Start"). Hyrje, daljet dhe hyrje/daljet e parametrave të përdorura nga FC, janë ruajtur si tregues në parametrat aktual të bllokut logjik që quhet FC.

Krijimi i block-ut FC1 behët duke i respektuar rregullat sikur më poshtë sikur siç i kemi krijuar edhe blloqet më herët

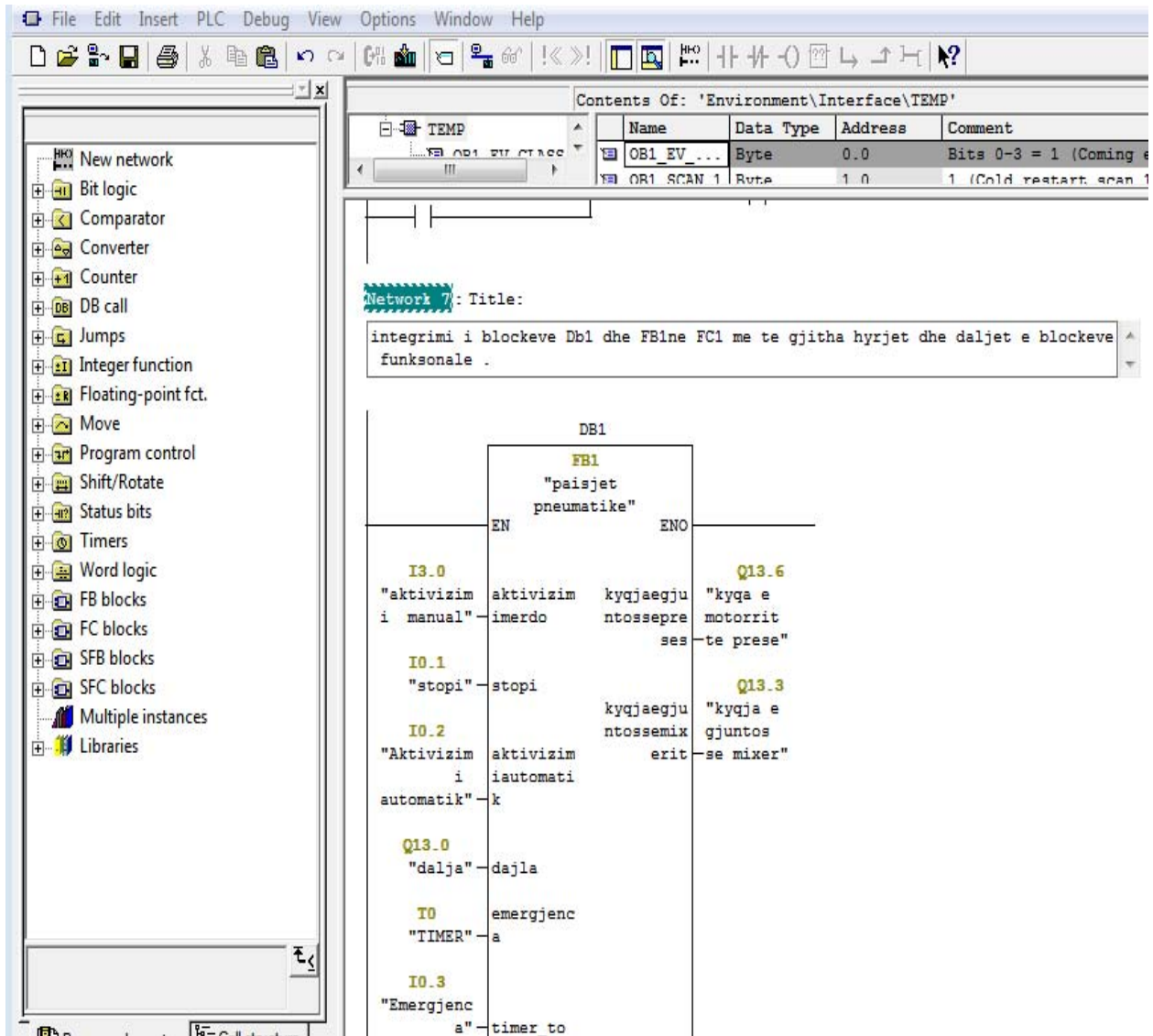
Selektojmë tastin e djathtë Insert New Object zgjedhim Function



Hapët dritarja Properties-Function me emrin FC1 zgjedhim gjuhën programuese LAD



Pas krijimi të block-ut FC1 klikojmë mbi atë bllok ku pasi të hapet bëjmë integrimi e blloqeve FB1si dhe të dhënat e block-ut DB1 sikur që shihet edhe në foto poshtë.

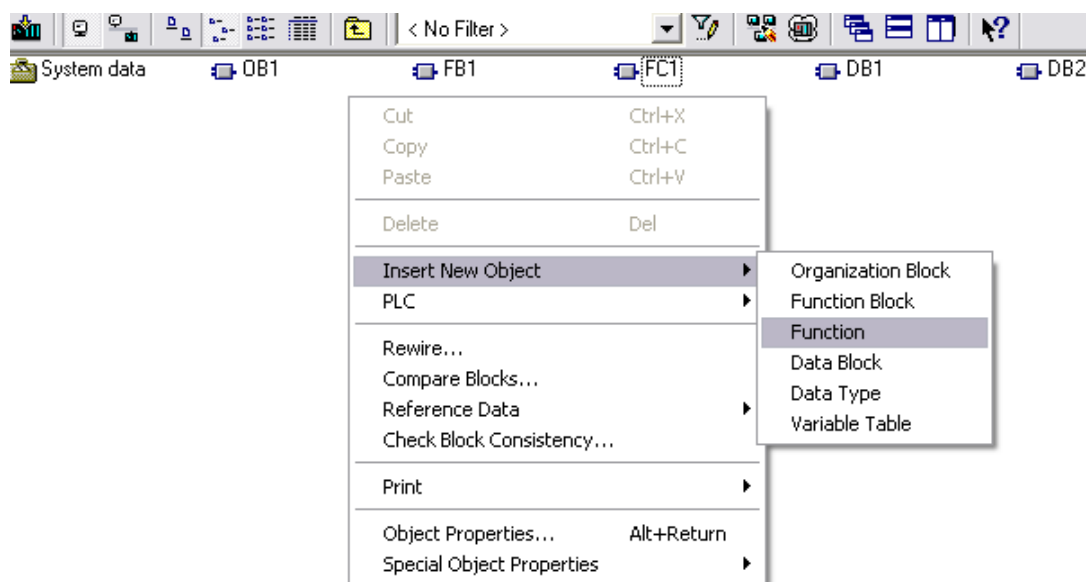


Pra këtu shihet se në block-u FC1 janë të integruar blloqet DB1 dhe FB1 me elementet hyrëse dhe dalëse.

5.1 PROGAMIMI DHE INTEGRIMI I NËNBLLOQEVE FUNKSIONALE (FB2 & DB2NË FC2)

Integrimi i block-ve FB2 dhe DB2 behët në FC2, bëhet sikur te integrimi i block-ut FC1 kushti që të bëhet është që së pari të krijohet block-u FC2. Krijimi i bllokut FC2 behët paraprakisht sikur blloku FC1.

Selektojmë tastin e djathtë Insert New Object zgjedhim Function



Hapët dritarja Properties-Function me emrin FC2 zgjedhim gjuhen programuese LAD

Properties - Function

General - Part 1 | General - Part 2 | Calls | Attributes

Name: FC2

Symbolic Name:

Symbol Comment:

Created in Language: LAD

Project path:

Storage location of project: C:\Program Files\Siemens\Step7\proj\Janar_20

Code	Interface
01/26/2016 10:47:32 PM	01/26/2016 10:47:32 PM

Date created: 01/26/2016 10:47:32 PM

Last modified: 01/26/2016 10:47:32 PM

Comment:

OK Cancel Help

Pasi që kemi krijuar block-un FC2 e hapim atë dhe pastaj bëjmë integrimin e blloqeve të DB2 dhe të FB2 në FC2 sikur që shihet më poshtë.

Pra këtu shihet se në block-u FC2 janë të integruar blloqet DB1 dhe FB2 me elementet hyrëse dhe dalëse që bëjnë ngritjen dhe uljen e pistonave varësisht prej elementeve logjike që i kemi përdorur për kushtet e caktuar.

Network 15 : Title:

Comment:

	DB2		
	EN	FB2	ENO
I3.0 "aktivizim i manual"	aktivizim i manual	pistonine ngritje	I3.4 "pistoni ne ngritje"
I0.2 "Aktivizim i automatik"	aktivizim iautomati k	pistonine ulje	I3.3 "pistoni ne ulje"
Q13.0 "dalja"	dalja		
I0.1 "stopi"	stopi		
I2.0 "senzori i pirunave"	senzori i pirunave		

PËRFUNDIMI

Punimi i diplomës ka përdor diagramin rrugë – kohë të procesit teknologjik për prodhimin e blloqeve (tullave të argjilës) në Fabrikën ELBOKSH në Klinë dhe metodën inxhinierike – Kaskade, përmes të cilave janë nxjerrë ekuacionet logjike me të cilat është realizuar sistemi digjital për kryerjen e procesit teknologjik.

Automati i programueshëm (Programmable Logic Controller PLC) është një prej komponentëve bazë të automatizimit industrial. Motivet kryesore të suksesit të tij duhet të kërkohen në besueshmërinë dhe thjeshtësinë e programimit.

Këto programe kryhen zakonisht nëpërmjet disa gjuhëve të thjeshta programuese dhe të riprogramueshme. Këto programe janë të lehtë për mirëmbajtje, mundësi për zgjerim si dhe punë në një sistem robust që mund të funksionojë me besueshmëri në mjedise të ndryshme industriale.

Me këtë temë tentojmë që të fitojmë rezultate të caktuara lidhur me programimin e blloqeve si dhe programimin detal të punës me PLC, dhe atë PLC Simatic S7 i cili ka aplikim shumë të madh në industri të llojeve të ndryshme.

Nga analiza teorike dhe pjesa eksperimentale është fituar një program detal i programimit të PLC-ve me anë të SIMATIC S7 i cili mundëson që të bëhet një model shumë i mirë për programimet e ngjashme si në SIMATIC.

Mendoj se kjo temë do të jetë një punim i mirëfilltë shkencor në lëmin e programimit të PLC-ve të cilët janë mjaftë të përhapur si të ne ashtu edhe në tërë vendet e zhvilluara industriale, dhe paraqesin një domosdoshmëri në punën me to.

LITERATURA

1. Kujtim Veisllari, "Automatët e Programueshëm" Tiranë 2008,
2. Prof. Dr. sc. Shaban shabani, "Dirigjimi dhe Rregullimi Automatik" Prishtinë 2002,
3. Milan Šodan, "Automatizacija Logičkim sklopovima", Zagreb 1973,
4. Kevin Colloins, " PLC Programming for industrial Automation",

Nga internet:

- Jürgen Kaftan - "PLC-Basic Course with SIMATIC S7 "
- Introduction to Mechatronics and Measurement Systems
- Ladder and Functional block Programming
- Basic Ladder Logic Programming
- Rationalization with Handling Technology
- PLC Programming for Industrial Automation
- Ladder Logic (LAD) for S7-300 and S7-400 Programming