

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



**PUNIM DIPLOME**  
**(Master)**

*Titulli i temës:*

**PROGRAMIMI I BLOQEVE FUNKSIONALE PËRMES SIMATIC S7-300 TE SISTEMI I SHIRITIT TRANSPORTUES DHE ROBOTIT SKARA**

*Titulli i temës në anglisht:*

**PROGRAMMING FUNCTIONAL BLOCKS THROUGH SIMATIC S7-300 ON CONVEYER BELT AND ROBOT SKARA SYSTEMS**

**Mentori:**  
**Prof. Dr. Agron Pajaziti**

**Kandidati:**  
**Bach. i mak. Muhamet Berisha**

**Prishtinë**

**2019**

**PËRMBAJTJA**

1.Hyrje .....	9
1.1. Përdorimi .....	9
1.2. Përdorimi special .....	10
1.3. Kostoja e ulët inxhinierike .....	10
1.4. Kostoja e ulët e operimit .....	10
3. SIMBOLET DHE VARIABLAT PËR PROGRAMIMIN E PLC-së .....	14
3.1. Urdhërat e përdorur për programimin .....	14
3.2. SIMBOLET .....	15
3.2.1. Simboli “&” (dhe, AND) .....	15
3.2.2. Simboli barazim ( Assing ) .....	15
3.2.3. Operacioni logjik OSE ( $\geq 1$ ) .....	16
3.2.4. Vija e mesme dalëse “ # ” ( Midline Output ) .....	16
3.2.5. Detektimi i RLO-së pozitive ( P ) .....	16
3.2.6. Starto prodhimin (vendos dalje) S Set Output .....	17
3.2.7. Ristarto prodhimin (rivendos dalje) R Reset Output .....	17
3.2.8. Starto_Ristarto Flip Flop (vendos_rivendos) SR Set_reset Flip Flop .....	17
3.2.9. Ristarto_Starto Flip Flop (rivendos_vendos) RS Reset_set Flip Flop .....	18
3.2.10. Krahasimi i dy vlerave .....	18
3.2.11. Lëvizja (Move): Transferimi i vlerave .....	18
3.2.12. Kalimi i pa kushtëzuar dhe kalimi i kushtëzuar në bllok .....	19
4.1. DIZAJNI DHE QARKU .....	20
4.2. Shembulli 1: Numërimi dhe Krahasimi i Urdhërave .....	25
4.2.1. Zona e Ruajtjes me numërues dhe krahasues .....	25
4.2.2. Programimi logjik që i aktivizon llambat e indikatorit në panelin tregues .....	26
4.3 Shembulli 2: Udhëzimet matematikore .....	28
4.3.1. Zgjedhja e problemeve matematikore .....	28
4.3.2. Programi logjik ”Ladder” .....	28
4.4. Shembulli 3: Udhëzimi i fjalës logjike .....	29
4.4.1. Sistemi i ngrohjes së furrës .....	29
4.4.2. Program logjik shkallë “Ladder Logic” .....	30

5. Zgjidhja e elementeve përbërëse të robotit .....	31
5.1 Elementet përbërëse të shiritit transportues dhe robotit skara.....	31
5.1.1 Aplikime praktike .....	31
5.1.2 Udhëzimet e përdorura .....	31
5.2. Elementet përbërëse të shiritit transportues dhe robotit skara.....	32
5.2.1 Shiritit transportues me elementet përbërëse.....	32
5.2.2 Roboti Skara me pjesët përbërëse të aksit të tretë .....	33
5.2.3 Roboti Skara me pjesët përbërëse të aksit dytë .....	34
5.2.4 Roboti Skara me pjesët përbërëse të aksit të parë dhe me komandat e lëvizjes.....	35
6. Programimi i blloqeve funksionale, me përshkrimin e sistemit dhe programimin e shiritit transportues përmes Simatic S7-300.....	38
6.1. Programimi i shiritit transportues .....	38
6.2. SHPJEGIMI I BLOQEVE FUNKSIONALE TË PROGRAMIMIT TË SHIRITIT TRANSPORTUES .....	39
6.2.1. OBI – Cikli i programimit dhe Organizimi i Blloqeve .....	39
6.2.2. Kuptimi i funksionimit të bllokut OBI.....	40
6.2.3. Të dhënat lokale për OBI .....	40
6.2.4. Programi kryesor me lëvizje të pandërprerë .....	42
6.3. PROGAMIMI I BLOQEVE FUNKSIONALE (FB).....	42
6.3.1. Bllok Funkcioni dhe Blloku i të Dhënave (FB dhe DB) .....	43
6.3.2. Variablat e të dhënave të tipit FB .....	43
6.3.3. Caktimi i parametrave aktual për parametra formal.....	43
6.3.4. Caktimi i vlerave fillestare te parametrat formal. ....	44
6.4. SHPJEGIMI I BLOKUT FUNKSIONAL FB1 .....	45
6.4.1. Simboli SR (Set Reset Flip Flop).....	45
6.4.2. Thirr FB nga kutia (CALL_FB) .....	46
6.4.3. Përshkrimi për thirrjen e bllokut FB.....	46
6.4.4. Thirr bllok/sistemin FB nga kutia (CALL_SFB).....	47
6.4.5. Përshkrimi për thirrjen SFB .....	47
6.4.6. Kohuesi (Timer) T1 .....	50
6.4.7. Udhëzimi i MOVE (lëviz) i integruar në Bllokun FB1, na lejon që të prodhojmë hapësirë kohore me frekuenca të ndryshme në daljen Q12.0 përmes Q13.7.....	52

6.4.8. Sensorët detektues të programuar në bllokun FB2 .....	53
6.4.9. Blloku funksional FB2 i integruar në bllokun e të dhënave DB2 në kuadër të bllokut kryesorë OB1 .....	54
6.4.10. Gjendja e sinjalit në motor.....	55
6.4.11. Numruesi (S_CU) i cili shërben për integrimin e sensorit numërues, si dhe qëndrimi në lëvizje të pandërprerë i shiritit transportues.....	56
6.5. PROGRAMIMI I SENSORIT INDUKTIV I INTEGRUAR NË BLOKUN FUNKSIONAL FB2. ..	57
6.5.1. Shembulli DB (bllok i të dhënave) .....	58
6.5.2. Krijimi themelorë i një blloku DB.....	58
6.6. BLOKU FUNKSIONAL FCs .....	59
6.6.1. Aplikimi.....	59
6.6.2. Dallime të rëndësishme në mes të parametrave të jashtëm FCs dhe FBs.....	59
6.6.3. Thirrja bllokut FC nga kutia, (CALL_FC).....	60
6.6.4. Thirr sistemin FC nga kutia (CALL_SFC).....	61
6.6.5. Blloku Funksional FC1 .....	62
6.7 Procesi i programimit të robotit Skara .....	63
6.8 OB1 Programi organizativ kryesor i funksionit të robotit Skara.....	64
6.8.1 Përshkrimi i ciklit të programit të bllokut organizativ OB1.....	64
6.9. OB100 Nënprogrami kryesor organizativ i robotit SKARA.....	64
6.9.1 Startimi i bllokut organizativ OB100.....	64
6.10. FB51 Blloku Funksional .....	65
6.10.1 Objektet dhe hierarkia e objekteve .....	65
6.10.2 DB Bllok i të dhënave.....	66
6.10.3 Krijimi themelor i një blloku DB .....	66
6.10.4 FC Blloku Funksional .....	66
6.11 SFB47 Sistemi i blloqeve funksionale .....	67
6.11.1 Kontrollimi i numëruesve me SFB47 “COUNT” .....	67
7. Programimi i lëvizjes të robotit skara përmes PLC Simatic S7-300 .....	68
7.1 OB1 Programi organizativ kryesor i funksionit të robotit Skara.....	69
7.1.1 OB1 “Qarku i programit kryesor (cikli)” .....	69
7.2 OB100 Nënprogrami kryesor organizativ i ROBOTIT SKARA .....	74
7.2.1 OB100 “Ristartimi i plotë i OB1” .....	74

<i>7.3 FB51 Blloku Funkcional</i> .....	75
<i>7.4 FC5: FUNKSIONI</i> .....	83
<i>7.5 FC10 Nën funksioni i programimit të lëvizjeve të robotit SKARA</i> .....	86
KONKLUSIONE DHE PËRFUNDIMI .....	93
LITERATURA E SHQYRTUAR.....	94

## Lista e figurave

<b>Figura 1.1</b> Linja e prodhimit në industrin automobilistike – automatizuar me S7-300.....	9
<b>Figura 1.2</b> Procesi fizik dhe sistemi i kontrollit.....	12
<b>Figura 4.1</b> Konfigurimi i S7-300:hapsirë-ruajtëse,modular dhe jashtëzakonisht i thjeshtë.....	20
<b>Figura 4.2</b> Ndërfaqja e integruar e S7-300 CPU për lidhje direkte në MPI dhe PROFIBUS DP .....	21
<b>Figura 4.3</b> CPU 314C-2 DP dhe CPU 315-2 DP (djathtas).....	22
<b>Figura 4.4</b> CPU 317-2 DP_ CPU standarde me dy DP Interfaces (ndërfaqje .....	24
<b>Figura 4.5</b> CPU 317T-2 DP_teknologjia CPU për funksionin e kontrollit të lëvizjeve.....	24
<b>Figura 4.6</b> Paneli tregues në shiritin transportues .....	26
<b>Figura 4.7</b> Tregon metoden manuale për startim, duke shtypur butonin me dorë.....	29
<b>Figura 5.1</b> Shiriti transportues) .....	32
<b>Figura 5.2</b> Roboti Skara aksi i tretë .....	33
<b>Figura 5.3</b> Roboti Skara aksi i dytë .....	34
<b>Figura 5.4</b> Roboti Skara aksi i parë.....	35
<b>Figura 5.5</b> Shiriti transportues .....	36
<b>Figura 5.6</b> Cilindri pneumatik.....	37
<b>Figura 6.1</b> Realizimi i punës së shiritit transportues .....	38
<b>Figura 6.2</b> Hiearkia e objekteve .....	66
<b>Figura 7.1</b> Roboti Skara.....	68
<b>Figura 7.2</b> Pamja e PLC68.....	69

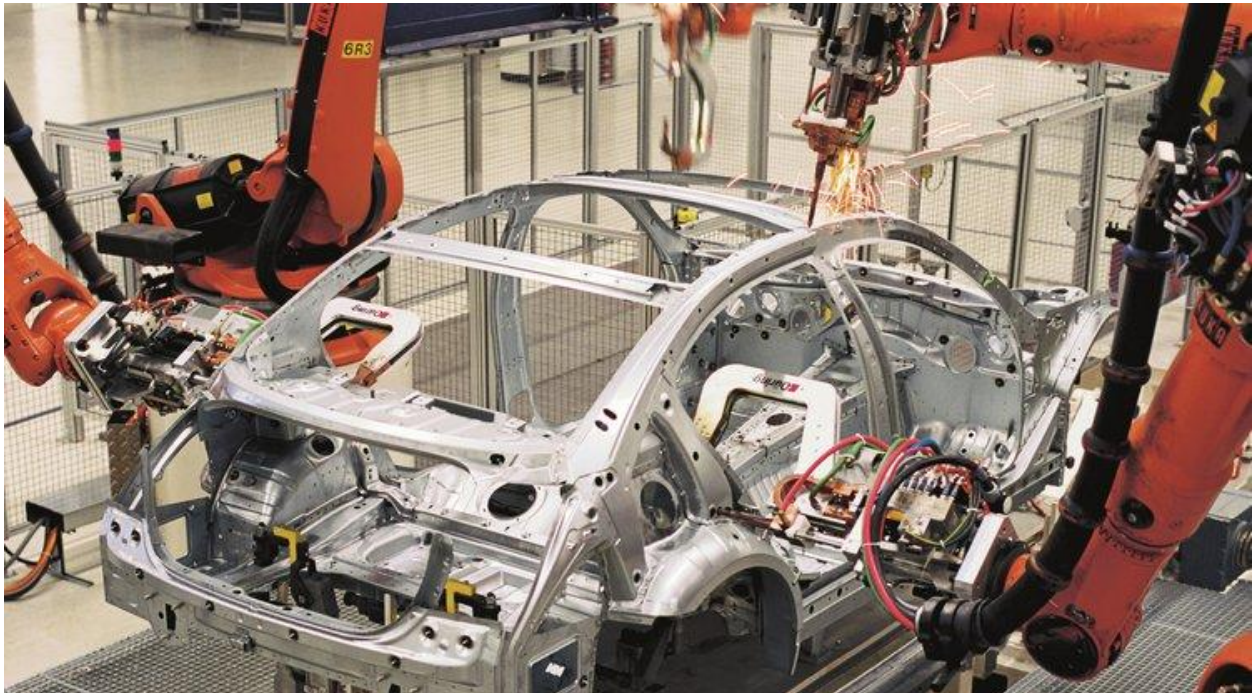
## Lista e tabelave

<b>Tabela 4.1</b> Të dhëna të përgjithshme të CPU-ve.....	23
<b>Tabela 4.2</b> Treguesi i komponenteve të sistemit dhe adresat absolute.....	29
<b>Tabela 5.1</b> Udhëzimet e përdorura.....	31
<b>Tabela 6.1</b> Organizimi i blloqeve, cikli i programimit për shiritin transportues.....	39
<b>Tabela 6.2</b> Struktura e blloqeve në programin burimorë.....	39
<b>Tabela 6.3</b> Variablat, tipet dhe përshkrimi për OB1.....	41
<b>Tabela 6.4</b> Caktimi i parametrave aktual, si dhe variablat hyrëse dhe dalëse në bllokun FB....	44
<b>Tabela 6.5</b> Caktimi i vlerës fillestare përmes variablave.....	44
<b>Tabela 6.6</b> Eelementet përbërëse në hyrje dhe dalje, parametrat, lloji i të dhënave dhe përshkrimi.....	45
<b>Tabela 6.7</b> Parametrat, lloji i të dhënave, zona e memories dhe përshkrimi për bllokun FB.....	46
<b>Tabela 6.8</b> Parametrat, lloji i të dhënave, zona e memories dhe përshkrimi për bllokun FB ...	47
<b>Tabela 6.9</b> Të dhënat programuese në hyrje të bllokut FB1.....	48
<b>Tabela 6.10</b> Të dhënat programuese në dalje të bllokut FB1.....	48
<b>Tabela 6.11</b> Tregon gjendjen e programuar të pandryshuar në numrim.....	48
<b>Tabela 6.12</b> Tregon gjendjen e pandryshuar të thirrjes së shiritit.....	48
<b>Tabela 6.13</b> Parametrat, lloji i të dhënave, zona e memories si dhe përshkrimi.....	50
<b>Tabela 6.14</b> Tregon gjendjen e dhënë në simbloin MOVE me të gjithë parametrat e dhënë.....	52
<b>Tabela 6.15</b> Paraqitja e të dhënave hyrëse për programimin e sensorit induktiv II.....	53
<b>Tabela 6.16</b> Paraqitja e të dhënave dalëse për programimin e sensorit induktiv II.....	53
<b>Tabela 6.17</b> Deklarimi, emri, adresa, tipi, vlera fillestare, komentimi.....	58
<b>Tabela 6.18</b> Deklarimi, emri, adresa, tipi, vlera fillestare, komentimi.....	58
<b>Tabela 6.19</b> Parametrat, lloji i të dhënave, zona e memories, si dhe përshkrimi për bllokun FC.....	60
<b>Tabela 6.20</b> Parametrat, lloji i të dhënave, zona e memories, si dhe përshkrimi për bllokun FC i.....	61
<b>Tabela 6.21</b> Thirrjet e OB-ve.....	65

# KAPITULLI I

## 1.Hyrje

S7-300 është programi më i kërkuar i kontrollit SIMATIC i cili mundëson zgjidhjen e problemeve automatike me sukses të lartë. S7-300 për sistemin e zgjidhjeve me theks të veçantë në inxhinierinë e prodhimit ka dëshmuar një platformë universale automatike që do të thotë një zgjidhje optimale për problemin tuaj. Parametrat e integruar në të e bëjnë këtë platformë vazhdimisht më të kërkuar [5], [6].



*Fig. 1.1. Linja e prodhimit në industrinë automobilistike – automatizuar me S7-300*

### 1.1. Përdorimi

SIMATIC S7-300 ofron zgjidhje për problemet e ndryshme automatike në fushat si në vijim:

- Inxhinierinë e prodhimit
- Industrinë automobilistike
- Konstruksionet e përgjithshme makinerike
- Konstruksionin e makinave speciale
- Konstruktimin e makinave serike (të gjitha llojet e makinave prodhuese)
- Përpunimin e plastikës
- Industrinë e paketimit (Ambalazhimit)
- Industrinë e ushqimit dhe duhanit
- Procesin inxhinierik (p.sh furnizimi me ujë, inxhinierinë e ndërtimit).



### ***1.2. Përdorimi special***

Për përdorime special kemi produkte të veçanta të dizajnuara me ndihmën e S7-300 si në vijim:

- Aplikimi i ruajtjes nga dështimi me aplikacionin e ri S7-300F sikurse I/O ku këto aplikime tani janë të mundshme,
- Për aplikacionet e kërkuara teknologjike dhe funksionin e kontrollit të lëvizjeve, mund të përdoret teknologjia speciale CPU,
- Në të jejtën kohë njësitë e kompletuara SIMATIC C7 janë të lejuara në S7-300 CPU, me integrimin e HMI-së për konfigurimin dhe kontrollin makinerik ku hapsira është e kufizuar,
- Ky spektër është një çark i ndalur nga CPU në formën e konfigurimit ET 200S, e cila bënë shpërndarjen ku procesi intelgjent është i mundur.

### ***1.3. Kostoja e ulët inxhinierike***

S7-300 karakterizohet nga efiqienca e konfigurimit dhe programimit, e cila rezulton me kosto të ulët inxhinierike. Kualiteti i tërësishëm i strukturave të CPU-së e bënë S7-300 një platformë ideale për problemet e caktuara në STEP 7, që është një mjet inxhinierik në përputhshmëri me standardin IEC 61131-3, për shembull në nivele të larta të gjuhës.

Përveç kësaj, teknologjia orientohet nga kohëzgjatja e softverit, p.sh. kontrolli i lëvizjeve të lehta mundë të përdoret për detyrën e kontrollit të lëvizjes. Mjetet inxhinierike gjithashtu janë lehtësuar me programimin modular dhe riparimin e softverit ekzistues. Veglat inxhinierike nuk mbështesin vetëm zhvillimin por gjithashtu edhe riparimin, lehtësimin e mirëmbajtjes dhe dokumentimin e programit. Konfigurimi i procesit të diagnostifikimit për analizimin e dështimeve të mundshme, reduktimin e humbjeve kohore dhe rritjen e produktivitetit janë implementuar në koston inxhinierike.

### ***1.4. Kostoja e ulët e operimit***

Mikro karda e memorjes The Micro Card (MMC) mund të kryejë funksionin si vend ruajtje e të dhënave të programit, prandaj nuk është e nevojshme të ketë bateri rezervë dhe kostoja e mirëmbajtjes mund të ruhet pjesërisht. MMC gjithashtu mund të ruaj një projekt, duke përfshirë këtu simbolet dhe komentet të cilat e bëjnë më të lehtë përdorimin pasi që nuk ka të dhëna të nevojshme në mjetet përkatëse.

MMC lehtësisht freskon të dhënat. Kjo lejon leximin dhe shkrimin e hyrjes edhe përgjatë ekzekutimit, sikurse në shembujt e arkivimit të vlerave për matje ose përpunimi i mjeteve, në mënyrë që leximi dhe shkrimi të bëhet më i lehtë.

## KAPITULLI II

Automatizimi i proceseve industriale mund të përcaktohet si disciplina 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 kualifikur ciklin e vet të prodhimit të motorëve.

Për kushtet e sotme modern të prodhimit, është i nevojshëm një automatizim shumë fleksibil që mundë të përcaktohet si “personalizmi 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 bashkangjiti 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” [1], [2].

Rëndësia e automatizimit të proceseve prodhuese moderne rrjedh nga një tërsi 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;
- Zvoglimin e jashtëzakonshëm të skarciteteve të prodhimeve;
- Koston më të vogël të prodhimit;
- Nevojen 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.2.1.).

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.

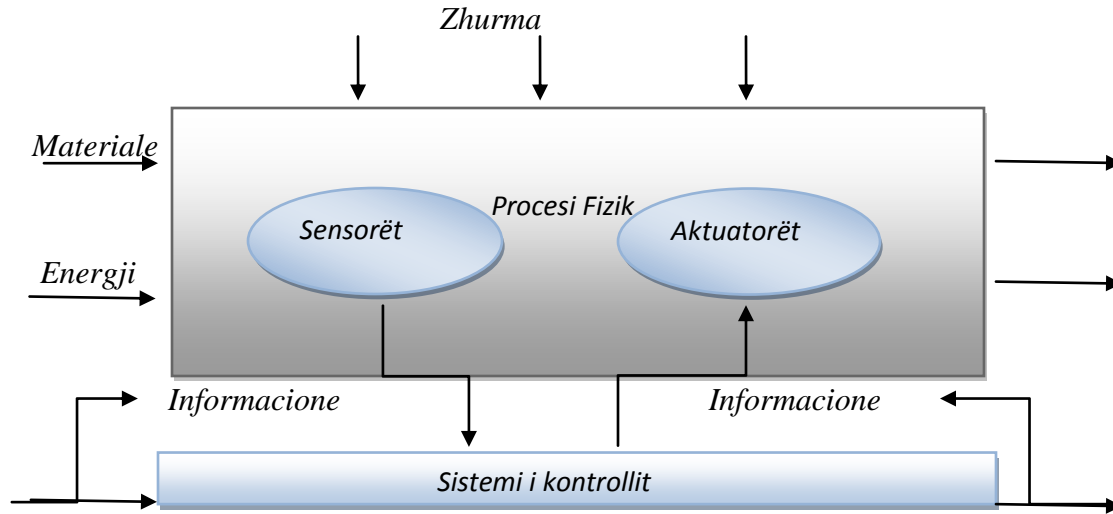


Fig. 1.2. Procesi fizik dhe sistemi i kontrollit

Një procesi 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 [2], [3].

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 fundëm të kontrollit*, paraprihen nga pre-aktuatorët, të cilët realizojnë kthimin e informacionit dhe amplifikimin e fuqisë.

Për shembull, një valvol 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 hierarkisht më të lartë.

Ky sistem gjithashtu, është në gjendje t'u dërgojë këtyre entiteteve 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).

## KAPITULLI III

### 3. SIMBOLET DHE VARIABLAT PËR PROGRAMIMIN E PLC-së

#### 3.1. Urdhërat e përdorur për programimin

Urdhërat e “Bit Logic” punojnë me dy shifra, 1 dhe 0. Këto dy shifra e formojnë bazën e një sistemi numrerik 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. [6], [7].

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

- ---|---            Kontakti normalisht i hapur (Address)
- ---| / |---        Kontakti normalisht i mbyllur
- ---(SAVE)        Ruajtja e RLO në kujtesën BR
- XOR            Bit ekzkluziv OR
- ---( )            Spirale 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).

### 3.2. SIMBOLET

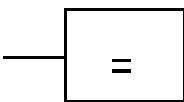
#### 3.2.1. Simboli “&” (dhe, AND)



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).

#### 3.2.2. Simboli barazim ( Assing )

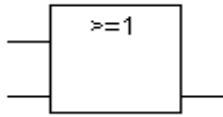


Ky simbol prodhon rezultatin e operacioneve logjike. Simboli në fund të operacionit logjik ka sinjalin “1” ose “0” bazuar në këto kritere:

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.

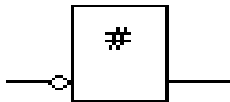
### 3.2.3. Operacioni logjik OSE ( $\geq 1$ )



Me operacionin logjik OSE ( $\geq 1$ ) 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.

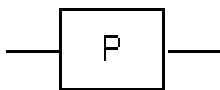
Secili operacion logjik OSE që nuk është operacioni i parë në varg kombinon rezultatin e sinjalit të kontrolluar të tij me vlerën e ruajtur nga biti RLO.

### 3.2.4. Vija e mesme dalëse “#” (Midline Output)



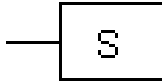
Vija e mesme dalëse është një element ndërmjetësues i cili e amortizon RLO-në. Më saktësisht ky element amortizon operimin e bitit logjik të fushës së fundit në mënyrë që të hapet para daljes së vijës së mesme. Ky operacion dirigjohet nga Master Contro Rele (MCR) Releja master kontrolluese.

### 3.2.5. Detektimi i RLO-së pozitive (P)



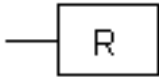
Operacioni i Detektimit të RLO-së Pozitive është operacioni i cili detekton ndryshimin nga “0” në “1” të adresës specifike dhe në të ndikon një RLO me vlerë “1” pas udhëzimit. Gjendja e sinjalit aktual të RLO-së është e kombinuar me gjendjen e sinjalit të adresës. Nëse gjendja e sinjalit të adresës është 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”.

### 3.2.6. Starto prodhimin (vendos dalje) S Set Output



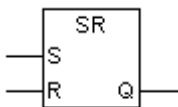
Udhëzuesi “Starto prodhimin” ekzekutohet vetëm atëherë kur RLO është “1”. Ky udhëzues vendos adresat specifike në “1”. Nëse RLO është “0” ky udhëzues nuk ka ndikim në adresën specifike, që do të thotë ska ndryshim. Udhëzuesi Start dirigjohet nga Master Kontroll Rele MCR.

### 3.2.7. Ristarto prodhimin (rivendos dalje) 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.

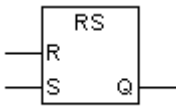
### 3.2.8. Starto\_Ristarto Flip Flop (vendos\_rivendos) SR Set\_reset Flip Flop



Udhëzuesi starto\_ristarto flip flop egzekuton 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.



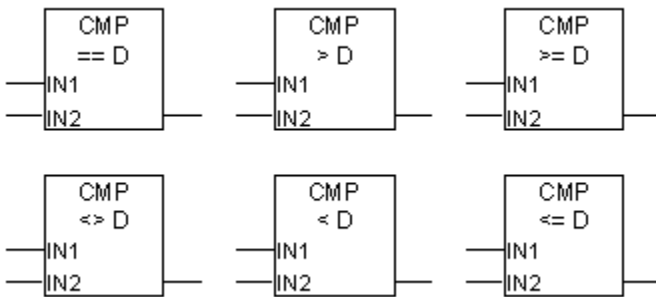
**3.2.9. Ristarto\_Starto Flip Flop (rivendos\_vendos) RS Reset\_set Flip Flop**



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”.

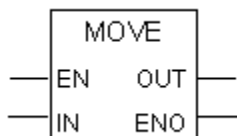
Nëse hyrja R është “0” dhe hyrja S është “1” flip flopi startohet. Nëse të dy hyrjet e RLO-së R dhe S janë “1” flip flopi startohet. Ky udhëzues dirigjohet nga Master Kontroll Rele MRC.

**3.2.10. 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.

**3.2.11. Lëvizja (Move): Transferimi i vlerave**



Me udhëzuesin për transferimin e vlerave mund të transferojm 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 zhvendosëse udhëzuesi i transferimit të vlerave mund të kopjon të gjitha të dhënat elementare me 8, 16 ose 32 bit.

Mund të caktojm 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.

### 3.2.12. Kalimi i pa kushtëzuar dhe kalimi i kushtëzuar në bllok



Urdhëri i kalimit të pa kushtëzuar në bllok korrespondon në urdhërin “shko tek emertimi”, asnjëri nga urdhërat në mes të kalimit dhe emertimit 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 urdhëri ekzekutohet vetëm në rastin kur RLO është “1”. Parkim, ku kamionët marrin pakot aty për aty, për dërgim tek konsumatorët. Një pengesë fotoelektrike (SENSOR) i vendosur në fund të shiritit transportues 2 afër zonës së magazinimit, përcakton se sa paketa do të largohen nga zona e magazinimit, për të shkuar në vend-ngarkim.

# KAPITULLI IV

## 4.1. DIZAJNI DHE QARKU

### 4.1.1. Dizajnimi

S7-300 mundëson një hapësirë për ruajtjen e konfigurimit modular të makinave kontrolluese, për adaptimin e mundësisë së zgjidhjeve të problemeve me dorë, pa marrë para sysh rezultatet e caktuara.

Një freskues nuk është i nevojshëm gjatë operimit. Kur është dizajnuar modelimi është bazuar në standardin gjerman DIN. Konfigurimi është i mundshëm dhe i fuqizuar në pajtueshmëri elektromagnetike.

Moduli i shumëllojshmërisë së spektrit S7-300 mund të përdoret për zgjerimet e centralizuara sikurse për konfigurimin e thjeshtë në strukturën e ET 200M; kjo rezulton në kosto, efikasitet, kursim dhe mirëmbajtje [4], [5].

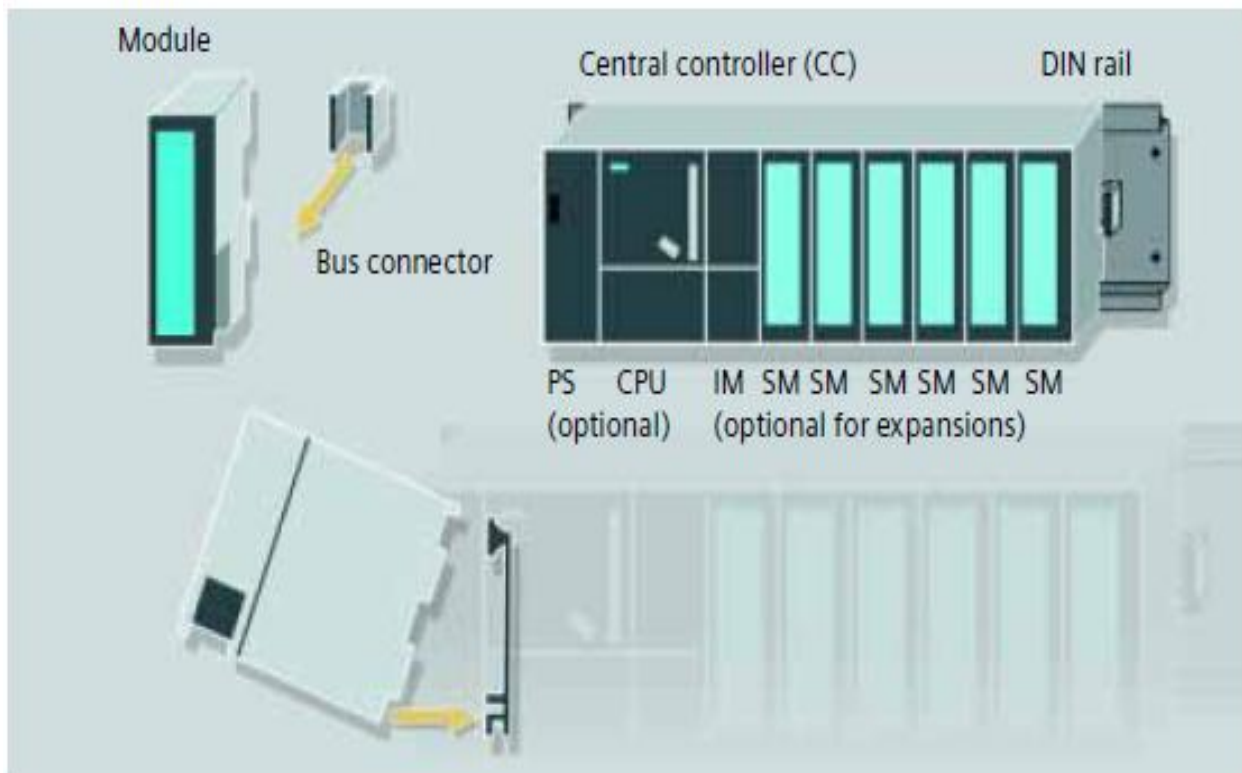


Fig. 4.1. Konfigurimi i S7-300: hapësirë-ruajtëse, modular dhe jashtëzakonisht i thjeshtë

#### 4.1.2. Fuqia, rrjeti fleksibil

Ndërfaqja e integruar direkt në CPUs lejon konfigurimin e pamjeve që mundëson instalimin e standardeve të teknologjisë, dhe pajisjet e programimit p.sh., për HMI. Ka lidhje të mjaftueshme të një numri të paisjeve HMI. Kursi i këti funksioni të pajisjeve programuese mundëson lidhjet në cilëndo pikë të qarkut, dhe mundëson adresimin e të gjitha nyjeve në qark.

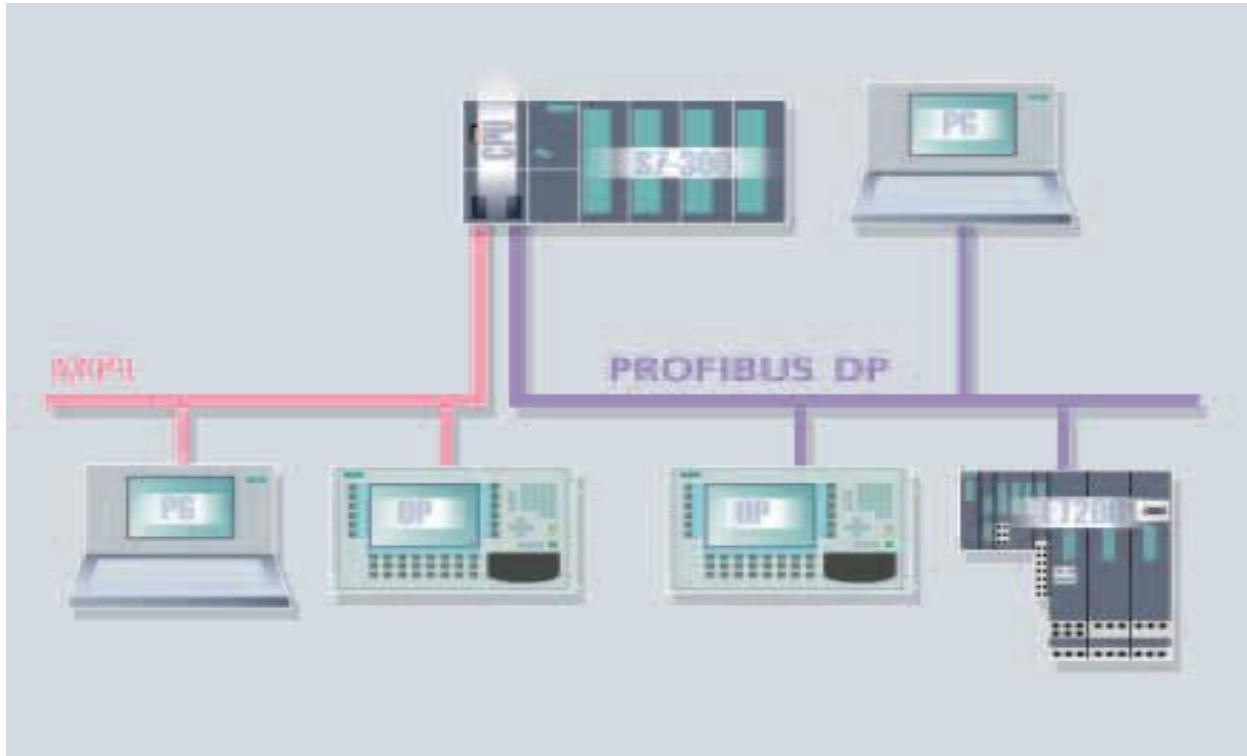


Fig. 4.2. Ndërfaqja e integruar e S7-300 CPU për lidhje direkte në MPI dhe PROFIBUS DP.

#### 4.1.3. Pika e shumfishtë e ndërfaqies – MPI

MPI është zgjidhje me kosto të ulët për komunikim me veglat programuese dhe PC-së (kompjuterit), sistemit HMI (R), dhe kontrolluesit tjetër SIMATIC S7/C7/W në AC. Ndalesat totale 125 MPI mund të lidhen në transmisionin e shkallës 187.5 kbit/s, p.sh., për shkëmbimin e të dhënave në mes të kontrollerëve të ndryshëm ose funksioneve HMI pa pasur nevojë për ndonjë programim të lartë. Me CPU 317 dhe 318-2 DP, MPI gjithashtu mund të përdoret si PROFIBUS DP dhe lejon konfigurimin e dy linjave DP.

#### 4.1.4. Profibus DP

Për konfigurimin ekonomik të rrjeteve të madha shpërndarëse, SIMATIC S7-300 mund të lidhet me PROFIBUS DP (bazuar në EN 50170). PROFIBUS DP mundëson komunikim të shpejtë me partner të shumëllojshëm, pra komunikimin nga kontrolleri SIMATIC në ekipin e veglave të fushës së tretë.

Gjithashtu komunikimi me SIMATIC S5 ose SIMATIC 505 ku sistemet ekzistuese mund të konfigurohen.

Shperndarja I/Os mund të konfigurohet me STEP 7 si I/O të përqëndruara e cila ruan shumë fletëngarkesa inxhinierike. S7-300 përdoret si kontrollor dhe mbikqyrës.

Sistemi është i hapur që nga mbështetja e funksionalizimit të DP V1. Bën caktimin dhe diagnostifikimin e parametrave për sistemet e rrjetit të lidhur me PROFIBUS në fushën e tretë.

Për përdorim në të ardhmen CPU-ja e re që ka të integruar Ethernet/PROFINet, ndërfaqje për komponentet e bazave automatike, kontrollin e programimit, operimit si dhe monitorimin përmes internetit do të jetë i mundshëm.

#### 4.1.5. Spektri CPU

Për ndërtimin e një sistemi të programimit kontrollues, përdoruesi duhet të zgjedhë një nga shkallët e spektrit të CPU-së, nga modeli startues e deri te performancat e larta të CPU-së. CPU-ja mundëson cikle punuese të shkurtëra të makinave duke u bazuar në shpejtsinë e performancës së saj. Moduli i ngushtë mundëson kontrollimin kompakt të konfigurimit dhe kontrollimin e dimensioneve të vogla të kabinetit kontrollues. Të ashtuquajturat standardet e CPU-së kanë një gjërësi prej vetëm 40 mm, dhe gjithashtu mundësojnë dizajnimin e dy ruajtësve nga dështimi. Komponentet e ndryshme CPU me gjërësi 80 deri 120 mm gjithashtu mund të integrojnë I/O dhe funksionet e tjera teknologjike [3], [7].



Fig.4.3. CPU 314C-2 DP dhe CPU 315-2 DP (djathtas)

Në Tabelën në vazhdim janë paraqitur të dhëna të përgjithshme, mbi CPU të llojeve të ndryshme.




Dizajni	CPU	Fillimi i dërgesës	Ndërpamja e integruar	Integrimi I/O	Integrimi i funksioneve teknologjike	Specifikimi teknologjikal: shiko faqen
Stabdardet e CPUs	CPU 312	3ti vendosur në 03	MPI			6
	CPU 314		MPI			6
	CPU 315-2 DP		DP, MPI			6
	CPU 317-2 DP 		DP, DP/MPI			6
	CPU 318-2 DP		DP, DP/MPI			7
Ruajtja e dështimit CPU	CPU 315F-2 DP	3ti vendosur në 03	DP, MPI		Fail-safety	7
	CPU 317F-2 DP 		DP, DP/MPI		Fail-safety	7
Kompakti CPU	CPU 312C	4ti vendosur në 03	MPI	Digital	Numërimi	8
	CPU 313C		MPI	Digital, analog	Numërimi	8
	CPU 313C-2 PtP		PtP, MPI	Digital	Numërimi	8
	CPU 313C-2 DP		DP, MPI	Digital	Numërimi	8
	CPU 314C-2 PtP		PtP, MPI	Digital, analog	<ul style="list-style-type: none"> <li>▪ Numërimi</li> <li>▪ Pozicionimi</li> </ul>	9
	CPU 314C-2 DP		DP, MPI	Digital, analog	<ul style="list-style-type: none"> <li>▪ Numërimi</li> <li>▪ Pozicionimi</li> </ul>	9
Teknologjia CPU	CPU 317T-2 DP 	4ti vendosur në 03	DP, DP/MPI	Digital	<ul style="list-style-type: none"> <li>▪ Sinkronizim,</li> <li>▪ Rrugëtimi për gjetjenë stopit</li> <li>▪ Shtyp shenjën e kontrollit</li> <li>▪ Ndërrimi kompjuter ik</li> <li>▪ Kontrollimi i pozicionimit</li> </ul>	9

Tabela 4.1. Të dhëna të përgjithshme të CPU-ve.

#### 4.1.6. Të rejat në CPU

Të gjitha CPU-të 317 kanë 512 KB për memorien e punimit dhe mundësitë gjithëpërfshirëse të pordorimit të veglave Inxhinierike të STEP 7, si dhe teknologjisë softverike të orientuar në bazë të jetëgjatësisë së softverit.

Gjithashtu ofrojnë përmirësime dhe qark punues më fleksibil, duke filluar që nga 32 lidhje aktive me nyjet e tjera, p.sh., veglat e programimit dhe panelet operatore, ku gjithashtu mund të ndërtohen njëkohësisht.

Më shumë se në tri versione të larta CPU-t janë në dispozicion për aplikacione të ndryshme:

- 317-2 DP standardi CPU, është i përshtatshëm për detyrat më të shpeshta të kontrollimit me shpërndarje të lartë të funksioneve komunikative. Njësoj DP ndërfaqet janë në dispozicion si kombinim i ndërfaqeve DP/MPI, dhe mund të konfigurohen me cilindo nga PROFIBUS-ët të larta, ose PROFIBUS-ët e ulëta,

- Fail-Safe (dështim-siguri) 317F-2 DP CPU ofron sistemin e sigurisë së dështimit në zgjerim në kompromis me standardet CPU, dhe PROFIsafe për komunikim të sigurt. Kompletohet me specifikimin e standardeve të fuqishme apo të mëdha:
  - IEC 61508 (SIL 3),
  - EN 954 (kategoria 4) dhe
  - NFPA 79, NFPA 85.

Ka aprovimin nga Inspeksioni Teknologjik Gjerman (TUV). Programet e përshtatshme dhe të sigurta mund të programohen duke përdorur STEP 7 LAD dhe FBD, ku këto janë gjuhët dhe certifikimi i shembujve të programimit nga libreria F e Shpërndarjes së sigurtë. Për zgjerim të mëtutjeshëm janë në dispozicion fail-safe ET 200S dhe ET 200M [4].

- Teknologjia CPU 317T-2 DP ngërthen në vete teknologji të fuqishme, dhe funksionet e kontrollimit të lëvizjeve në CPU. Është dizajnuar për kontrollimin e lëvizjeve dinamike dhe akseve të shumëllojshme. Para-programimi i PLC-së bën hapjen e butë, funksionin i kontrollit të lëvizjeve, integrimin diskret I/O dhe barazlargimin e lëvizjeve të PROFIBUS DP. Mundëson kontrollin fleksibil të lëvizjeve dhe akseve të veçanta, p.sh., kontrollimi i pozicionimit, sinkronizimi dhe/ose ndryshimi përmes kompjuterit. STEP 7 mundëson konfigurimin dhe parametrizimin e akseseve.



*Fig.4.4. CPU 317-2 DP\_ CPU standarde me dy DP Interfaces (ndërfaqie)*



*Fig.4.5. CPU 317T-2 DP\_teknologjia CPU për funksionin e kontrollit të lëvizjeve*

#### **4.1.7. Gjashtë kompaktësitë e CPU-së**

- CPU 312C
- CPU 313C
- CPU 313C-2 PtP
- CPU 313C-2 DP
- CPU 314C-2 PtP
- CPU 314C-2 DP

Me integrimin e I/O dhe funksionet teknologjike, programi vazhdimisht është i ripërtirë dhe ka treguar funksionalitet dhe performancë të lartë si:

- Përvetësimi i shpejtësisë së vlerave aktuale për numërimin dhe matjen periodike, mundëson përmirësimin e shkëlqyeshëm dinamik. P.sh për pozicionimin e problemeve. Procesimi i trefishtë i shpejtësisë për pikën qarkulluese operon me shumë shpejtësi ekzekutimin e programit.
- Mbrojtja e kopjes së përdorimit të softverit mund të implementohet duke e skenuar numrin serial të MMC gjatë operimit.
- Një numër i rritur i funksioneve të thirrjes dhe blloqeve të të dhënave, lehtëson bartjen e programeve të përdorura të cilat janë krijuar për CPU “standard”, dhe tani përdoren si CPU kompakte.

## **4.2 Shembulli 1: Numërimi dhe Krahasimi i Urdhërave**

### **4.2.1. Zona e Ruajtjes me numërues dhe krahasues**

Figura në vazhdim paraqet një sistem me dy shirita transportues dhe një zonë të magazinimit apo ruajtjes së përkohëshme në mes tyre. Shiriti transportues “1” dërgon pakot në zonën e magazinimit. Një pengesë fotoelektrike (SENSOR) në fund të shiritit transportues 1 në afërsi të zonës së magazinimit, bën përcaktimin se sa pako janë dorëzuar (zbrazur) në zonën e magazinimit. Shiriti transportues 2, transporton pakot nga zona e magazinimit në një vend ng i tregues që përfshin 5 ndriçues, përcjellë mbushjen e pakove të caktuara në zonën e depos së përkohëshme. [6], [7].



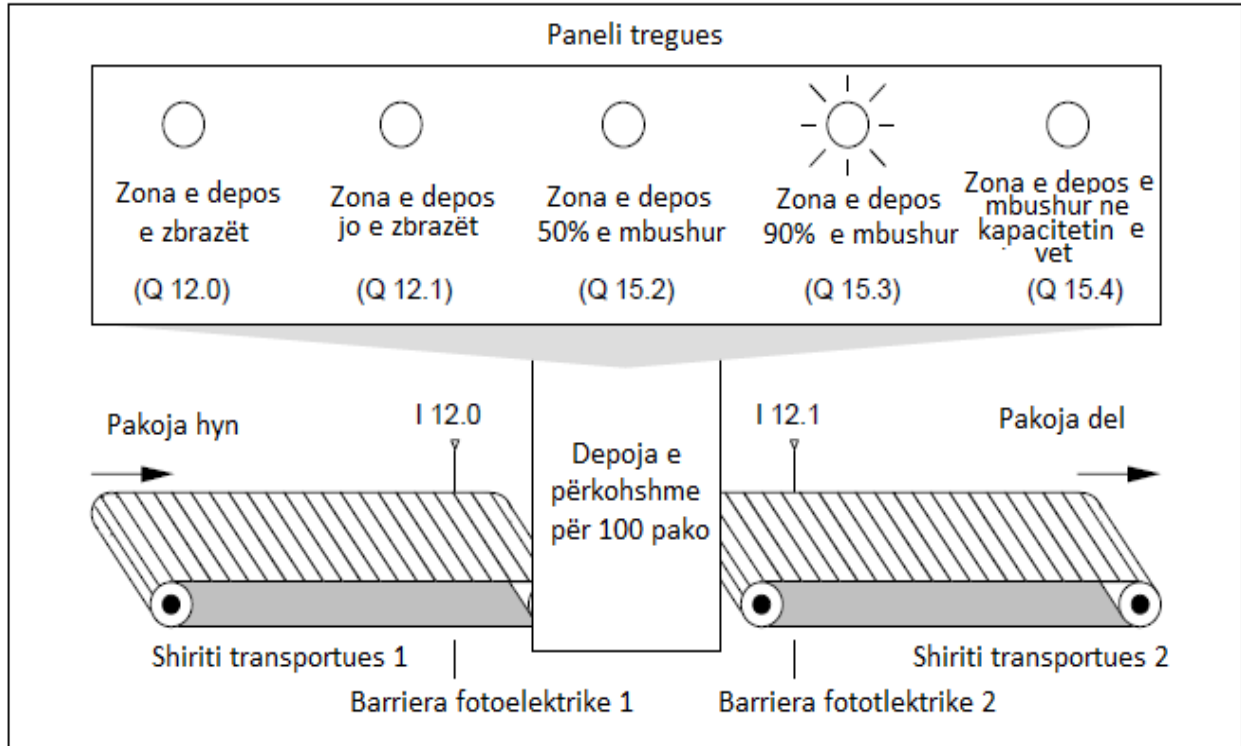
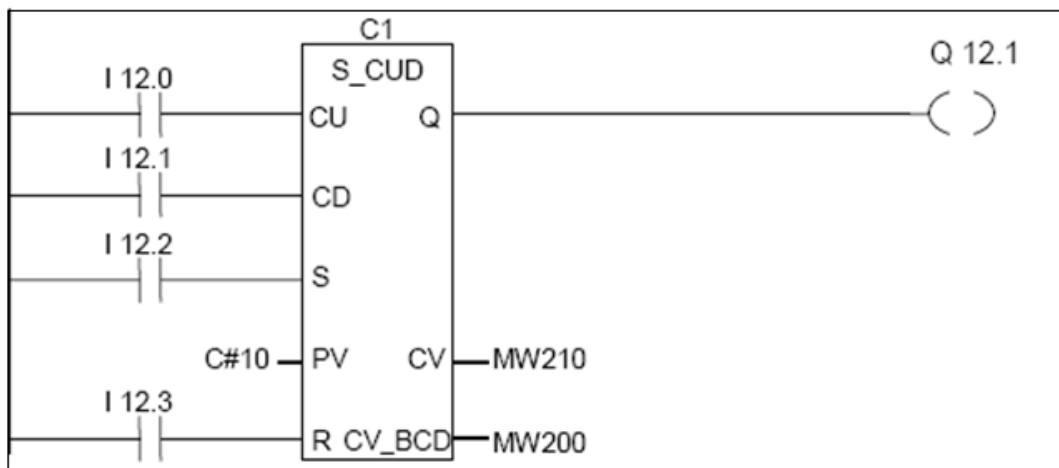


Figura 4.6. Paneli tregues në shiritin transportues.

**4.2.2. Programimi logjik që i aktivizon llambat e indikatorit në panelin tregues**

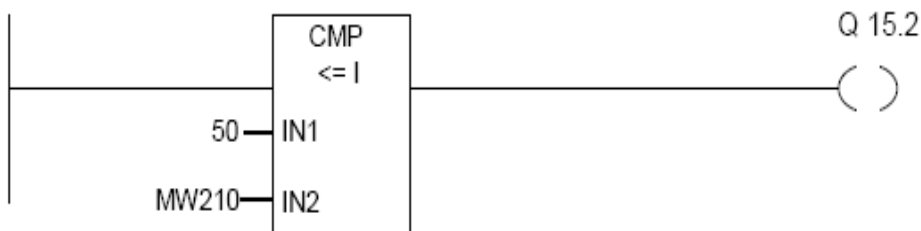
Qarku 1: Numëruesi (Counter) C1 numëron çdo ndryshim sinjali nga “0” në “1” në hyrjen CU, dhe numëron secilin ndryshim-zbritje të sinjalit nga “0” në “1” në hyrje të CD. Me një ndryshim të gjendjes nga “0” në “1” në hyrje të S, vlera e numëruesit është vendosur në vlerën PV. Një ndryshim sinjali në gjendjen nga “0” në “1” në hyrjen R, rivendosë vlera në numërues në gjendjen “0”. MW200 përmban vlerën aktuale për numëruesin C1. Q12.1 tregon “zonë magazinimi jo bosh”.



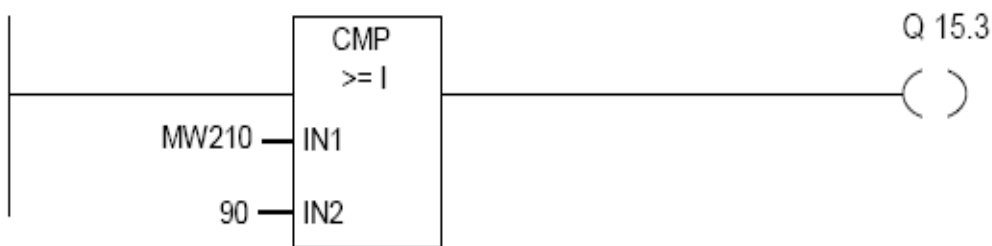
Qarku 2: Q12.0 tregon “zonën e magazinimit të hapur”.



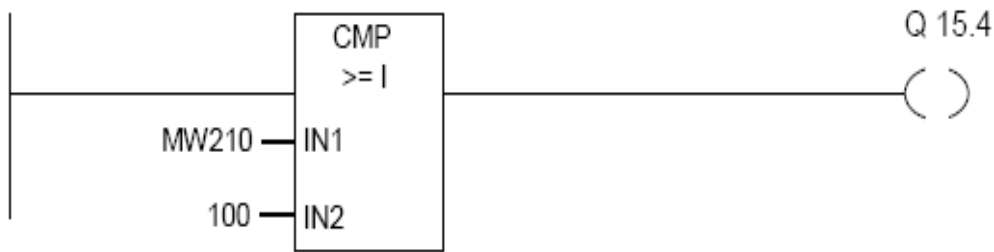
Qarku 3: Nëse vlera e numëruar është më e vogël ose e barabartë me 50 (me fjal të tjera nëse vlera e numëruar aktuale është më e madhe ose e barabart me 50), llamba e indikatorit për “zonën e depos 50% të mbushur” është ndezur.



Qarku 4: Nëse vlera e numruar është më e madhe ose e barabart me 90, llamba e indikatorit për “zonën e mbushur 90%” është ndezur.



Qarku 5: Nëse vlera në numruar është më e madhe ose e barabart me 100, llamba e indikatorit për “zonën e magazinimit të mbushur (plotësuar)” është ndezur.



### 4.3 Shembulli 2: Udhëzimet matematikore

#### 4.3.1. Zgjedhja e problemeve matematikore

Një program i thjeshtë na tregon se si mund të përdorim udhëzimet matematikore për të përfituar rezultatin e ekuacionit në vazhdim:

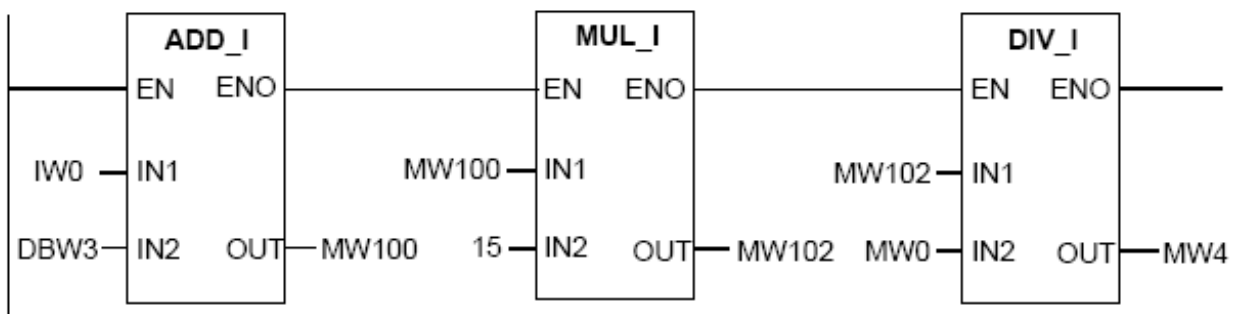
$$MW4 = ((IW0 + DBW3) \times 15) / MW0$$

#### 4.3.2. Programi logjik "Ladder"

Çarku 1: Hapja e bllokut të të dhënave DB1



Çarku 2: Fjala hyrëse IW0 është përcaktuar për të shpërndarë fjalën e përbashkët të të dhënave DBW3 (blloku i të dhënave duhet të përcaktohet dhe të hapet), dhe shuma është ngarkuar në fjalën e memorien MW 100. MW 100 pastaj është shumëzuar me 15, dhe përgjigjëjet ruhen në fjalën MW102 të kujtesës. MW 102 është e ndarë nga MW0 me rezultatin e ruajtur në MW4.



### 4.4. Shembulli 3: Udhëzimi i fjalës logjike

#### 4.4.1. Sistemi i ngrohjes së furrës

Operatori për ngrohjen e furrës fillon duke e shtyrë butonin e shtytjes Start. Ky operator mund ta ketë përcaktuar gjatësinë e kohës për ngrohje, duke përdorur butonin në formë timoni të treguar në figurë. Vlerën që përcakton operatori tregon sekondat e përcaktuar në binarë (BCD) në formë decimale [3], [5].

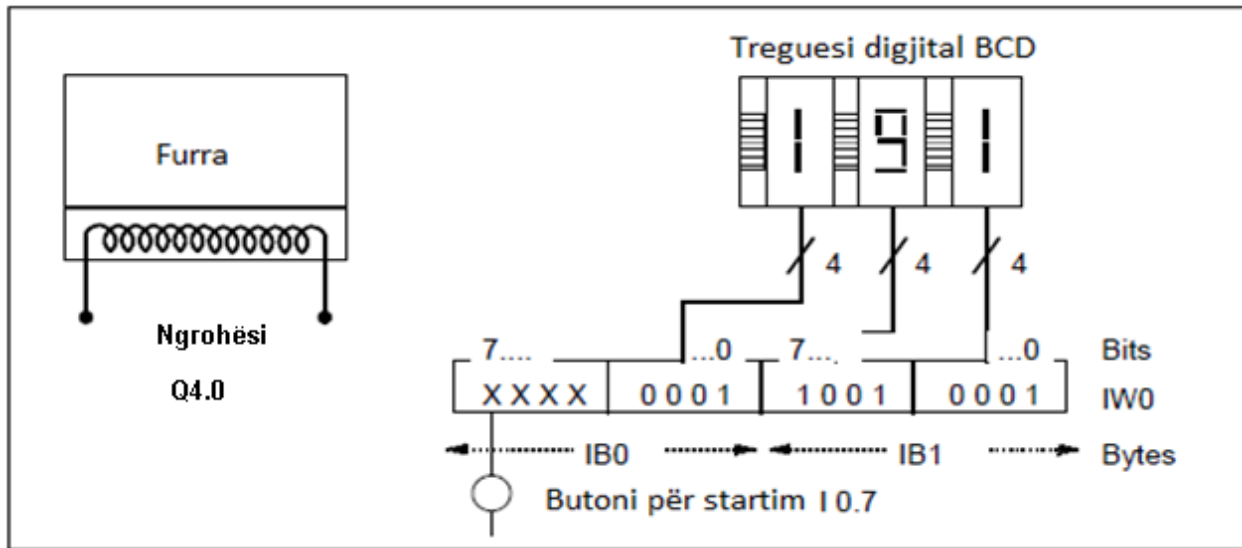


Figura 4.7. Tregon metoden manuale për startim, duke shtypur butonin me dorë.

Komponentet e sistemit	Adresa absolute
Shtyp butonin start	I 0.7
Treguesi për njëshe	I 1.0 to I 1.3
Treguesi më i lartë	I 1.4 to I 1.7
Treguesi për qindra	I 0.0 to I 0.3
Fillimi i ngrohjes	Q 4.0

Tabela 4.2. Treguesi i komponenteve të sistemit dhe adresat absolute.

**4.4.2. Program logjik shkallë “Ladder Logic”**

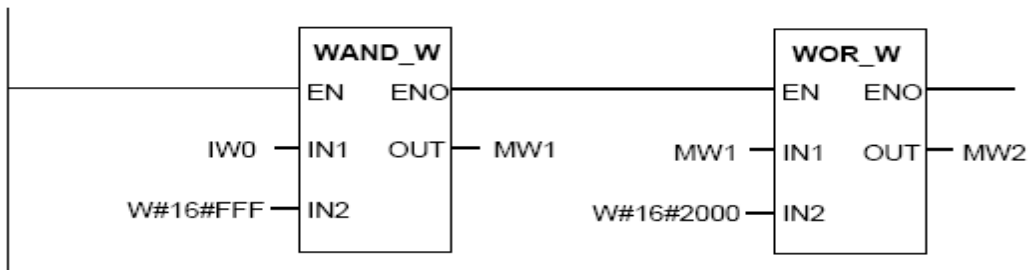
Qarku 1: Nëse kohuesi është në funksion, atëher vëhet në funksion ngrohësi.



Qarku 2: Në qoftë se kohuesi është duke punuar, udhëzimi rikthehu përfundon procedimin këtu.



Qarku 3: Biti i maskës hyrëse I0.4 përmes I0.7 (rikthen atë në gjendjen 0). Këto bite të treguesit hyrës në mënyrë rotative duke startur me anë të gishtit tregues, nuk janë përdorur. Të 16 pjesët (bitët) të hyrjeve janë kombinuar me W#16#0FFF në bazë të udhëzimit (Word) dhe përshkrimit të wordit. Rezultati është ruajtur në memorien me fjalë MW1. Me qëllim për të caktuar kohën bazë në sekonda, vlera prezente është kombinuar me W#16#2000 në bazë të fjalës (Word), ose udhëzuesit të fjalëve, rregullimin e bitëve nga 13 në 1, dhe rivendosjen e bitëve 13 drejt 0.



Qarku 4: Starto kohuesin T1 si pulsant kohor i zgjëruar, nëse butoni start është shtypur dhe ruhet si vlerë prezente në memorien me fjalë MW2 (rrjedhë nga logjika më lart).



# KAPITULLI V

## 5. Zgjidhja e elementeve përbërëse të robotit

### 5.1 Elementet përbërëse të shiritit transportues dhe robotit skara

#### 5.1.1 Aplikime praktike

Secili udhëzues logjik përshkruhet në këtë manual me një operacion specifik. Nëse kombinojmë këto udhëzime në program, mund të realizojmë një shumëllojshmëri detyrash të automatizimit. Në këtë kapitull shqyrtohen shembuj të aplikimit praktik të shënuar më poshtë:

Kontrollimi i shiritit transportues duke përdorur udhëzuesin e bitit logjik.

Detektimi e drejtimit të lëvizjeve në shiritin transportues duke përdorur udhëzuesin e bitit logjik.

Gjenerimi i pulseve të orës duke përdorur udhëzimet kohore [3], [7].

Mbajtja e rrugës, drejtimit nga depoja duke përdorur udhëzimet e numërusit dhe krahasuesit.

Zgjedhja e një problemi duke përdorur udhëzimet matematikore.

Caktimi i gjatësisë së kohës për nxehjen e një furrë.

#### 5.1.2 Udhëzimet e përdorura

Tab. 5.1. Udhëzimet e përdorura

Memorja	Elementet e programit	Pershkrimi i katalogut
WAND_W	Udhëzimi i fjalëve logjike	(fjalë) dhe fjalë
WOR_W	Udhëzimi i fjalëve logjike	(fjalë) ose fjalë
---(CD)	Numërues	Unaza numëruse e poshtme
---(CU)	Numërues	Unaza numëruse e epërme
---(R)	Udhëzimet e bitit logjik	Rivendosje e unazës
---(S)	Udhëzimet e bitit logjik	Vendosje e unazës
---(P)	Udhëzimet e bitit logjik	RLO kufiri pozitiv
ADD_I	Udhëzimet e pikes qarkulluse	Vendos kufirin (integrimin)
DIV_I	Udhëzimet e pikes qarkulluse	Ndaje kufirin (integrimin)
MUL_I	Udhëzimet e pikes qarkulluse	Kufiri i shumëllojshëm
CMP <=	Krahason	Krahason kufijt (integrimin)
—   —	Udhëzimet e bitit logjik	Kontaktet e hapura normale
—  /  —	Udhëzimet e bitit logjik	Kontaktet e mbyllura normale
—()	Udhëzimet e bitit logjik	Unaza e prodhimit
---( JMPN )	Kalimi	Kalo-nëse-jo
---( RET )	Kontrollimi i programit	Rikthim
MOVE	Lëviz	Cakto vlerën e lëvizjes
--- ( SE )	Kohor	Unaza e zhgjeruar kohore

## 5.2. Elementet përbërëse të shiritit transportues dhe robotit skara

### 5.2.1 Shiritit transportues me elementet përbërëse

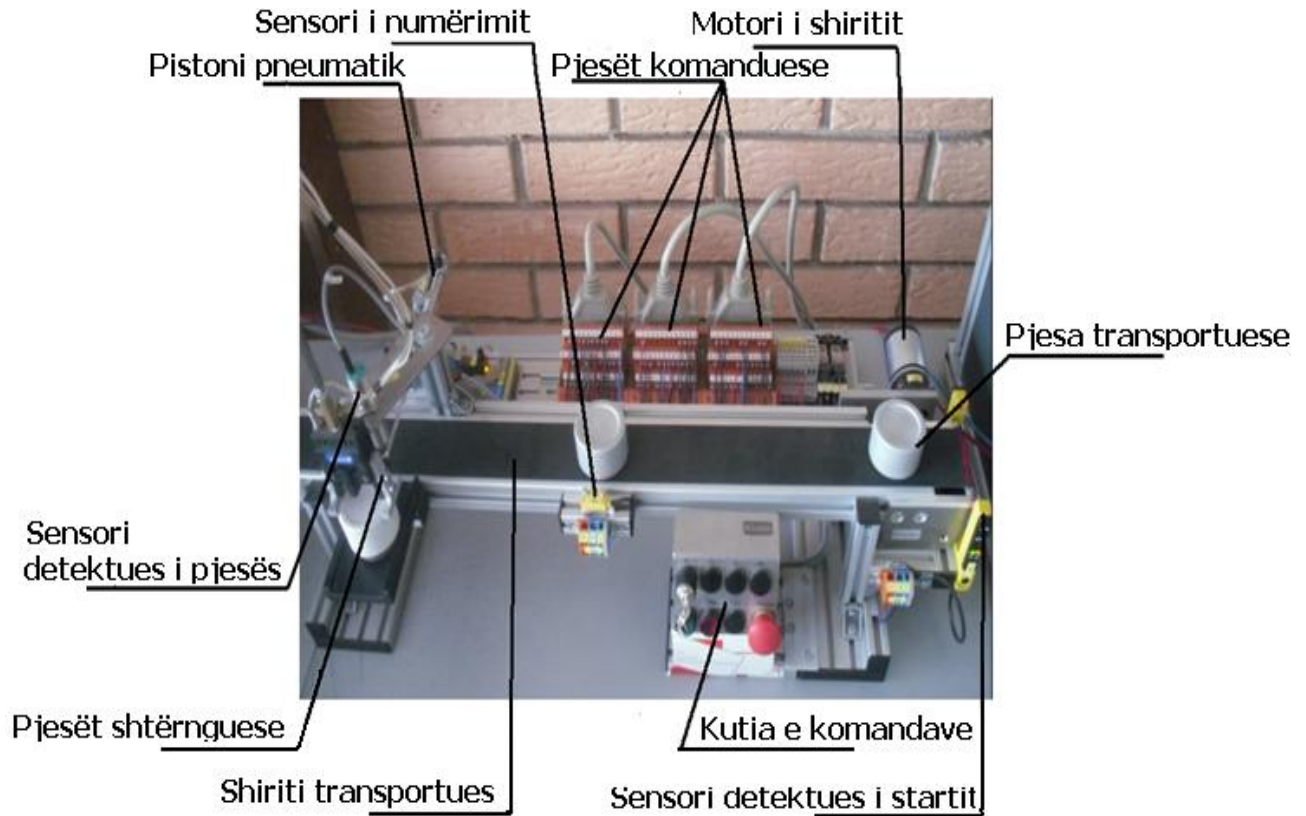


Fig. 5.1. Shiriti transportues.

### 5.2.2 Roboti Skara me pjesët përbërëse të aksit të tretë

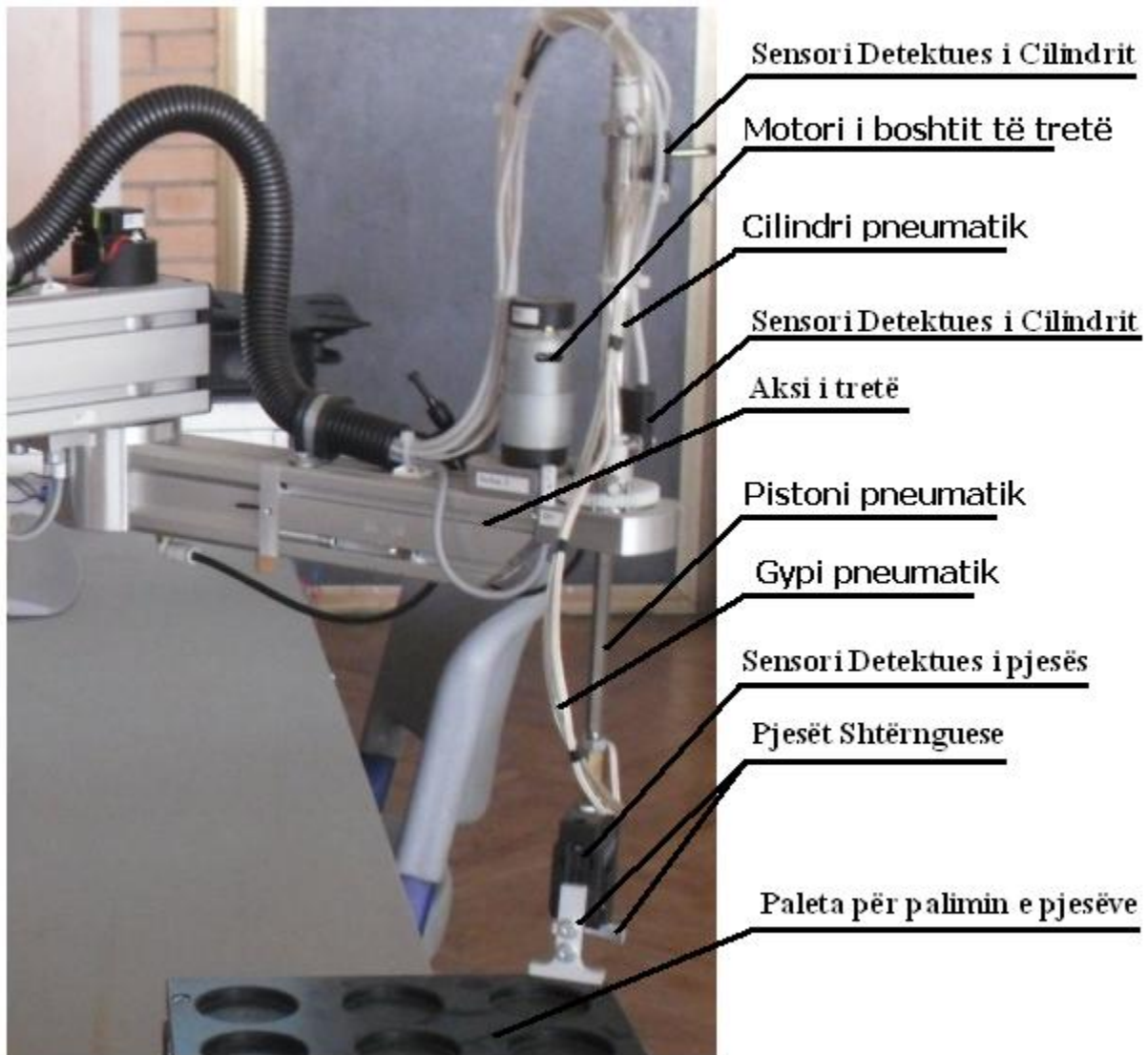
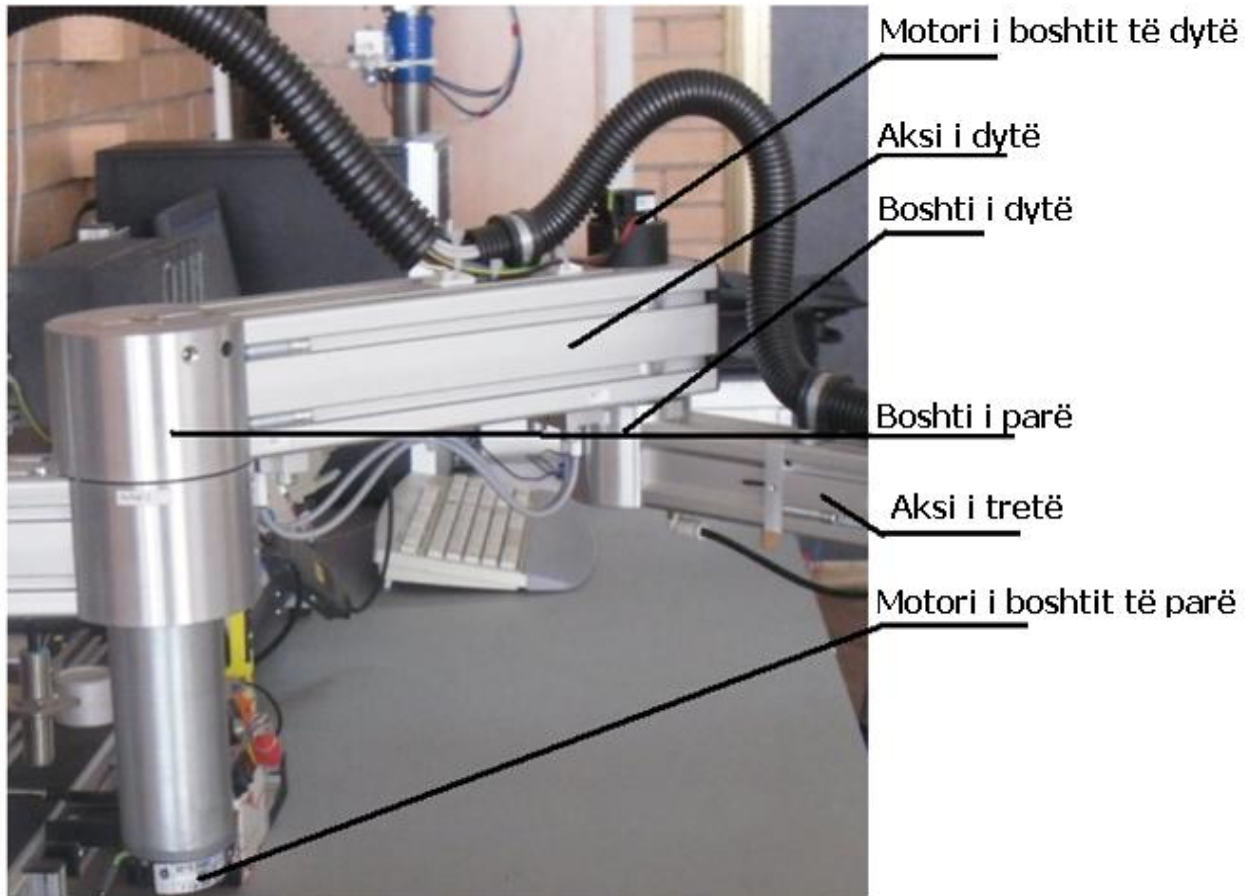


Fig. 5.2. Roboti Skara aksi i tretë.



### 5.2.3 Roboti Skara me pjesët përbërëse të aksit dytë



*Fig. 5.3. Roboti Skara aksi i dytë.*

### 5.2.4 Roboti Skara me pjesët përbërëse të aksit të parë dhe me komandat e lëvizjes

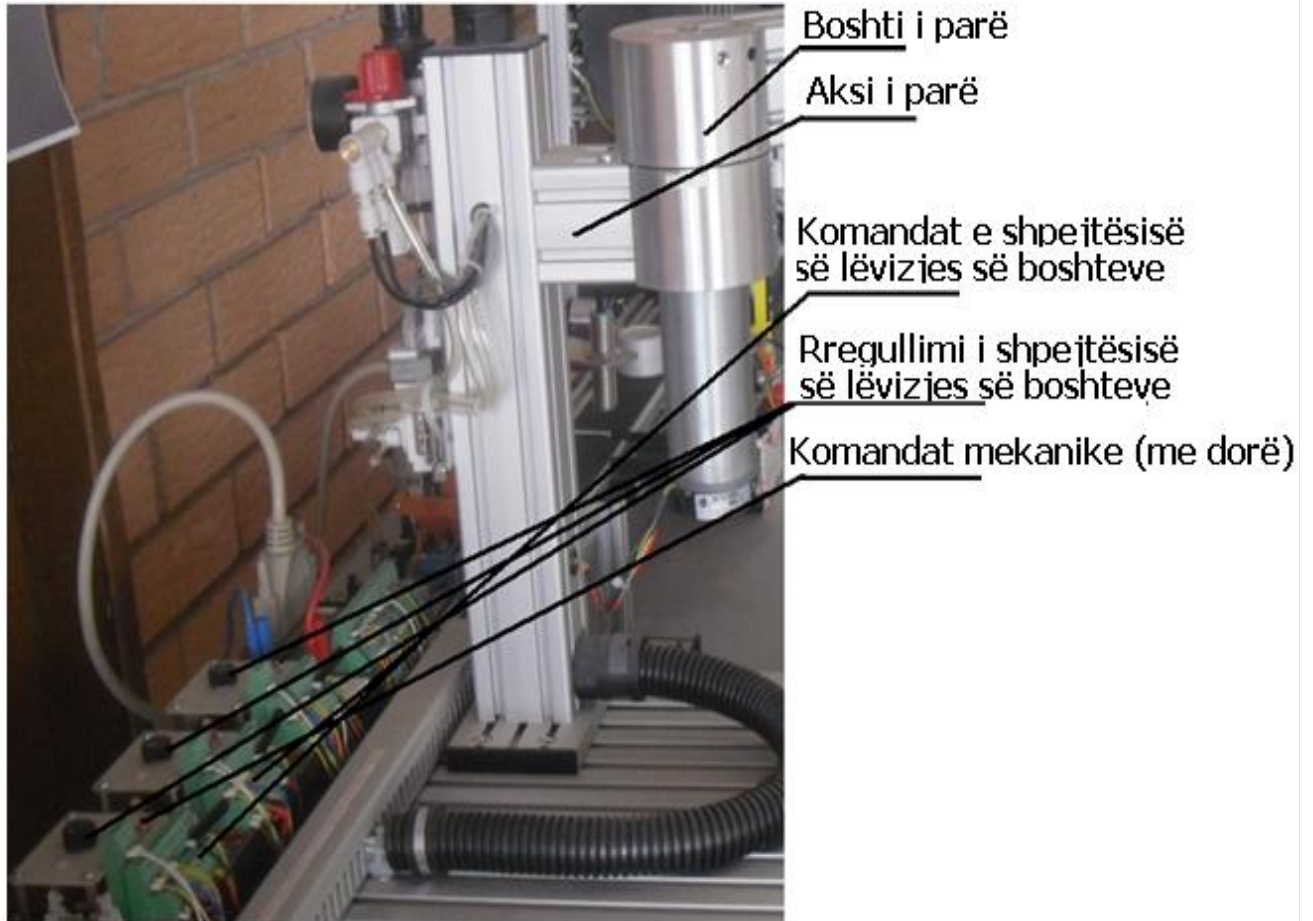
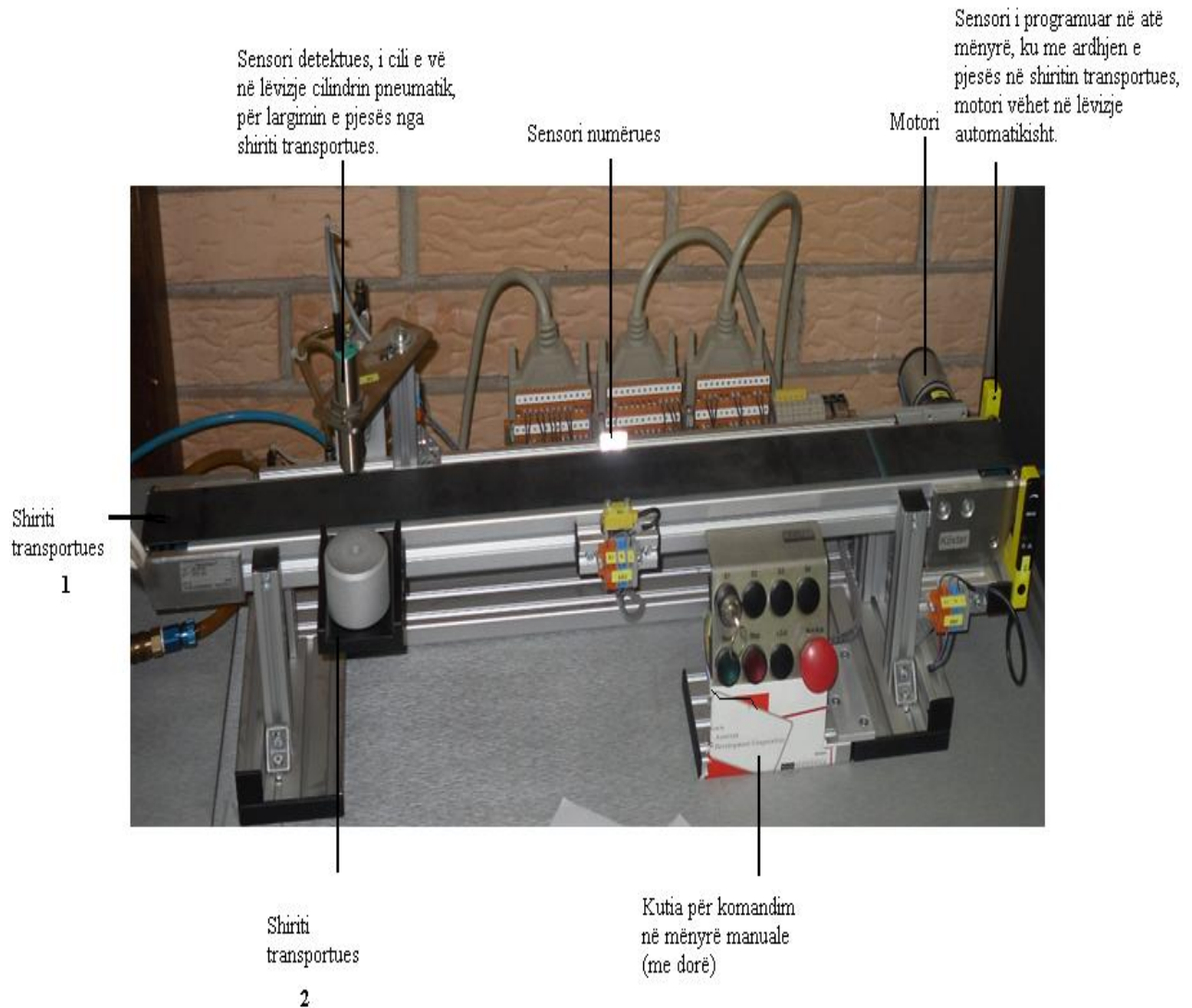


Fig. 5.4. Roboti Skara aksi i parë.

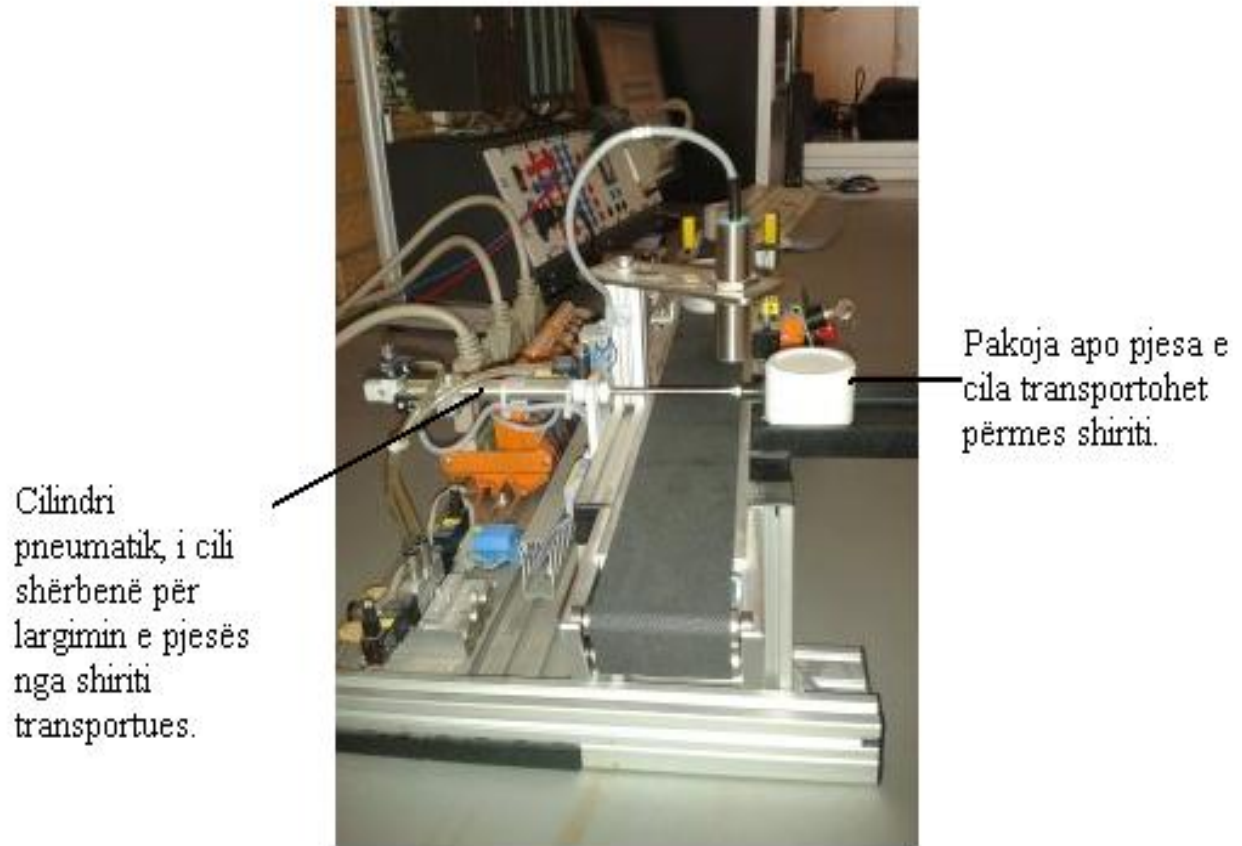
Shiriti transportues i cili gjindet në laboratorin e FIM-së me pajisjet përcjellëse: motori, sensorët, shiriti, kutia komanduese në mënyrë manuale, si dhe pjesa e cila transportohet dhe komandohet përmes programimit logjik SIMATIC S7-300.



*Fig. 5.5. Shiriti transportues.*

Cilindri pneumatik i lidhur me sensorë, i cili shërbenë për largimin e pjesës nga shiriti transportues në mënyrë të programuar, ku me ardhjen e pjesës tek sensor i detektues, motori ndalet, largohet pjesa me ndihmën e cilindrit pneumatik, në shirit tjetër transportues apo në

ndonjë pako, dhe cilindri kthehet në pozitën normale pra në pikën zero, dhe në mënyrë automatike përsëri fillon punën motori. Ky cikël përsëritet çdo herë.



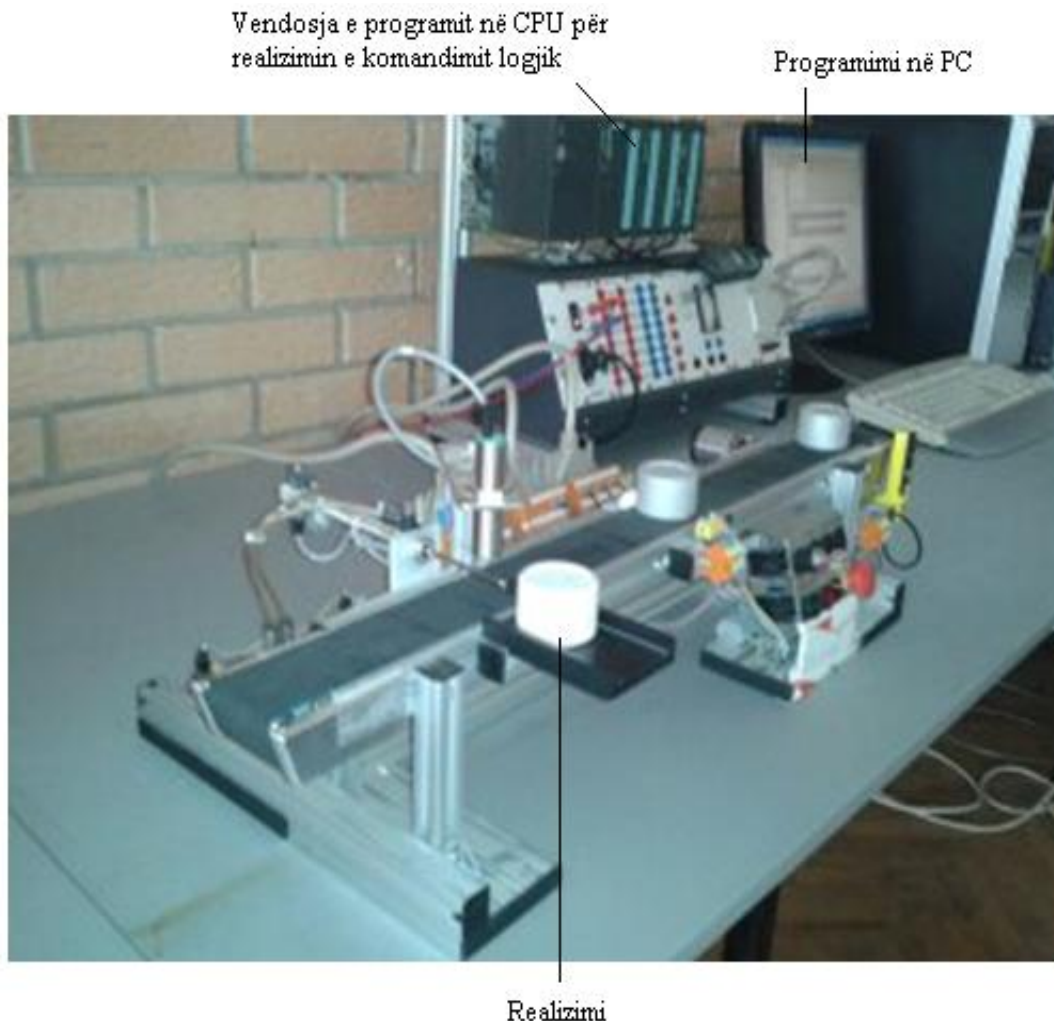
*Fig. 5.6. Cilindri pneumatik.*

## KAPITULLI VI

### ***6. Programimi i blloqeve funksionale, me përshkrimin e sistemit dhe programimin e shiritit transportues përmes Simatic S7-300***

#### ***6.1. Programimi i shiritit transportues***

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 [4], [5], [6], [7].



*Fig. 6.1. Realizimi i punës së shiritit transportues.*



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 nënblloqe të programuara dhe të integruara njëra me tjetrën, në mënyrë që shiriti transportues të programohet sipas nevojë.

Cikli i programimit është i përbërë nga:

OB1 – Programi kryesor, me nënprograme; FB1, FB2, FC1, DB1, DB2, si dhe blloqe funksionale: FB60, FB62, FB80, FB81, FC60, FC61, FC62, FC63, SFB52.

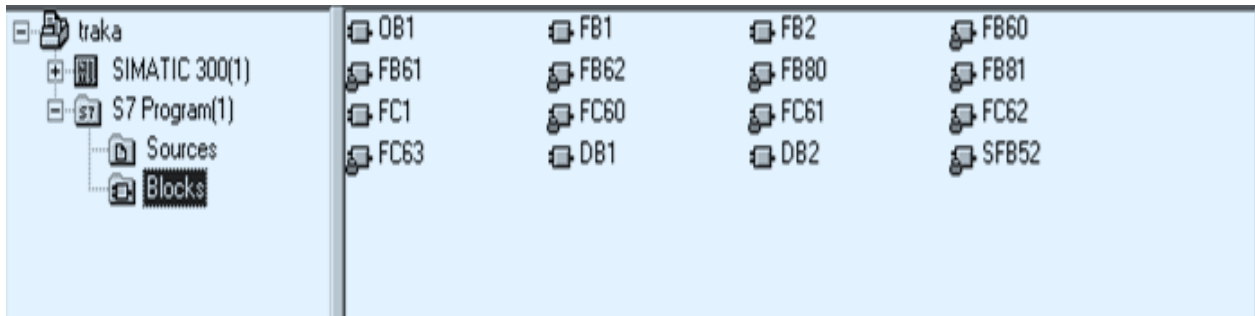


Tabela 6.1. Organizimi i blloqeve, cikli i programimit për shiritin transportues.

Interface	Name	Data Type	Address	Comment
TEMP	OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
	OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
	OB1_PRIORITY	Byte	2.0	Priority of OB Execution
	OB1_OE_NUMBER	Byte	3.0	1 (Organization block 1, OB1)
	OB1_RESERVED_1	Byte	4.0	Reserved for system
	OB1_RESERVED_2	Byte	5.0	Reserved for system
	OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
	OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
	OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
	OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started
	A	Word	20.0	
	B	Word	22.0	

Tabela 6.2. Struktura e blloqeve në programin burimorë.

## 6.2. SHPJEGIMI I BLOQEVË FUNKSIONALE TË PROGRAMIMIT TË SHIRITIT TRANSPORTUES

### 6.2.1. OB1 – Cikli i programimit dhe Organizimi i Blloqeve

Sistemi operativ i CPU S7 ekzekuton organizimin e bllokut OB1 periodikisht. Kur blloku i organizuar OB1 është ekzekutuar, sistemi operativ fillon përsëri realizimin. Ekzekutimi ciklik OB1 është nisur pas fillimi, dhe cikli përfundon komplet. Në gjithashtu mund të krijohet blloqe të tjera funksionale (FBS, SFBs) ose funksionet (FCS, SFCs) në organizimin e blloqeve OB1.

### 6.2.2. Kuptimi i funksionimit të bllokut OB1

OB1 ka prioritet më të vogël ndaj të gjithë OBs-ve në zhvillim nga një cikël i monitoruar, me fjalë të tjera, të gjitha OBs tjera përveç OB90 mund ta ndërpresin ekzekutimin e OB1. Zhvillimet e mëposhtme të shkaktuara nga sistemi operativ për të thirrur OB1:

- Fillimi i përfunduar,
- Ekzekutimi OB1 (cikli i mëparshëm) ka përfunduar.

Kurë OB1 është ekzekutuar, sistemi operativ i dërgon të dhënat e përgjithshme. Para se të ristartohet OB1, sistemi operativ shkruan tabelën përfundimtare të procesit të pasqyrimin dhe të moduleve prodhuese-dalëse për realizime të ndryshme programuese, dhe kjo përditson tabelën e hyrjeve, dhe procesit të pasqyrimin që gjithashtu merr të dhëna të përgjithshme për një CPU.

S7 monitoron kohën maksimale, duke siguruar një kohë maksimale për përgjigjëje. Vlera maksimale për kohën e pasqyruar paraprakisht është 150 ms. Ne mund të vendosim një vlerë të re, ose mund të rrisim monitorimin e kohës kudo në programin tonë me SFC43"RE\_TRIGR". Nëse programi kalon kohën maksimale të ciklit OB1, sistemi operativ e thërret bllokun OB80 (OB kohë e gabuar); nëse OB80 nuk është programuar, CPU ndryshon me mënyrën STOP.

Përveç monitorimit maksimal të kohës, gjithashtu është e mundur për tu garantuar edhe një kohë minimale. Sistemi operativ do të shtyj fillimin e një cikli të ri (shkrim i tabelës prodhuese në procesin e imazhit të moduleve të prodhimit), derisa të arrihet koha minimale.

### 6.2.3. Të dhënat lokale për OB1

Në Tabelën 6.3. janë paraqitur variablat, tipet si dhe përshkrimi për OB1.

Variabli	Tipi	Përshkrimi
OB1_EV_CLASS	BYTE	Klasë-veprimtaria dhe identifikuesit: B#16#11: OB1 aktiv
OB1_SCAN_1	BYTE	<ul style="list-style-type: none"> <li>• B#16#01 përfundim i një ristarti të ngrohtë</li> <li>• B#16#02 përf. i një ris. Të nxehtë</li> <li>• B#16#03 përf. i ciklit kryesorë</li> <li>• B#16#04përf. i një</li> </ul>

		cikli të ftohët B#16#05 cikli i parë OB1 në një CPU pas kalimit në gjendjen STOP
OB1_PRIORITY	BYTE	Klasë prioriteti 1
OB1_OB_NUMBR	BYTE	Numri OB1 (01)
OB1_RESERVED_1	BYTE	E rezervuar
OB1_RESERVED_2	BYTE	E rezervuar
OB1_PREV_CYCLE	INT	Koha e mëparshme e skanimit ( ms )
OB1_MIN_CYCLE	INT	Koha minimale e ciklit (ms) që nga fillimi i fundit
OB1_MAX_CYCLE	INT	Koha maksimale e ciklit (ms) që nga fillimi i fundit
OB1_DATE_TIME	DATE AND TYPE	DATA_DHE_KOHA e ditës kur OB1 është thirrur

*Tabela 6.3. Variablat, tipet dhe përshkrimi për OB1.*



#### 6.2.4. Programi kryesor me lëvizje të pandërprerë

**OB1** : Programi kryesor me lëvizje të pandërprerë

Blloku FunkSIONAL FC1

Qarku i bllokut funksional FC1



*Blloku funksional FC1 i programuar dhe i organizuar në bllokun OB1*

#### 6.3. PROGRAMIMI I BLOQEVE FUNKSIONALE (FB)

Bllonet funksionale FBs iu përkasin atyre blloqeve që ne mund t'i programojmë vet. Blloku i të dhënave është caktuar si një memorie ( shembull i bllokut të të dhënave). Parametrat që janë transferuar në FB dhe variablat statike janë ruajtur në shembullin DB. Variablat e përkohëshme janë ruajtur në ram-memorien e të dhënave lokale.

Të dhënat ruajtura në shembullin DB nuk mund të humbin, kur ekzekutimi i bllokut FB është i plotë. Të dhënat e ruajtura në RAM memorien e të dhënave, megjithatë humbin kur ekzekutimi i bllokut FB ka përfunduar. [4], [5], [6], [7].

Një bllok FB, përmban një program që gjithmonë është ekzekutuar kur FB është thirrur nga një bllok tjetër logjik. Bllonet funksionale shpesh e bëjnë më të lehtë programimin me funksione komplekse.

### 6.3.1. Bllok Funkcioni dhe Blloku 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 motora 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.

### 6.3.2. Variablat e të dhënave të tipit FB

Nëse programi përdorues është i strukturuar në mënyrë që një FB përmban thirrje për blloqe më tej, tashmë funksioni ekzistues mund të përfshijë FBs blloqe, që do të quhen si ndryshore statike të tipit të të dhënave të bllokut FB në tabelën e deklarimit të ndryshueshëm të thirrjes FB.

### 6.3.3. Caktimi i parametrave aktual për parametra formal

Në përgjithësi nuk është e domosdoshme në STEP 7 caktimi i parametrave aktual për parametra formal në bllokun FB. Megjithatë ka përjashtime për këtë. Parametrat aktuale duhet të caktohen në rastet e mëposhtme:

- Për një (in/out) parametër të një lloji të ndërlikuar i të dhënave ( për shembull, STRING, ARRAY ose DATE\_AND\_TIME)
- Për të gjitha llojet e parametrave (për shembull, TIMER, COUNTER ose POINTER)

STEP 7 cakton parametrat aktuale në parametrat formalë të një blloku FB si në vazhdim:

- *Kurë ne saktësojmë parametrat aktualë në formulimin e thirrjes:* udhëzimet e bllokut FB, përdorin parametrat e ofruar aktualë
- *Kur ne nuk saktësojmë parametrat aktualë në formulimin e thirrjes:* udhëzimet e bllokut FB përdorin vlerën e ruajtur në shembullin e bllokut DB.

Në tabelën e mëposhtme janë dhënë: lloji i të dhënave fillestare, lloji i të dhënave komplekse si dhe lloji i parametrit.	<b>Lloji i të dhënave</b>		
<b>Variablat</b>	<b>Lloji i të dhënave fillestare</b>	<b>Lloji i të dhënave komplekse</b>	<b>Lloji i parametrit</b>
Hyrjet	Jo parametër i duhur	Jo parametër i duhur	Parametër aktual i duhur
Daljet	Jo parametër i duhur	Jo parametër i duhur	Parametër aktual i

			duhur
Hyrje/Dalje (in/out)	Jo parametër i duhur	Parametër aktual i duhur	--

*Tabela 6.4. Caktimi i parametrave aktual, si dhe variablat hyrëse dhe dalëse në bllokun FB.*

#### **6.3.4. Caktimi i vlerave fillestare te parametrat formal.**

Ne mund të caktojmë vlerat fillestare të parametrave formale në pjesën e deklaramit 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.

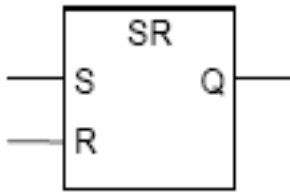
Tabela e mëposhtme tregon variablat të cilat mund të caktojnë një vlerë fillestar. Nëse të dhënat e përkohëshme janë humbur pasi që blloku është ekzekutar, atëherë nuk mund t'u caktohet asnjë vlerë atyre të dhënave. [4], [5], [6], [7].

	<b>Lloji i të dhënave</b>		
<b>Variablat</b>	<b>Lloji i të dhënave fillestare</b>	<b>Lloji i të dhënave komplekse</b>	<b>Lloji i parametrat</b>
Hyrje	Vlera fillestare e lejuar	Vlera fillestare e lejuar	--
Dalje	Vlera fillestare e lejuar	Vlera fillestare e lejuar	--
Hyrje/Dalje (in/out)	Vlera fillestare e lejuar	--	--
Statik	Vlera fillestare e lejuar	Vlera fillestare e lejuar	--
I përkohshëm	--	--	--

*Tabela 6.5. Caktimi i vlerës fillestare përmes variablave.*

## 6.4. SHPJEGIMI I BLOKUT FUNKSIONAL FBI

### 6.4.1. Simboli SR (Set Reset Flip Flop)



Prametrat	Lloji i të dhënave	Zona e kujtesës	Përshkrimi
<address>	BOOL	I, Q, M, L, D	Vendos ose Rivendos "bit"
S	BOOL	I, Q, M, L, D	Aktivizo udhëzimin e vendosur
Q	BOOL	I, Q, M, L, D	Aktivizo udhëzimin e rivendosur
R	BOOL	I, Q, M, L, D	Gjendja e sinjalit në <address>

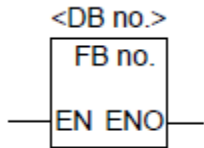
Tabela 6.6. Eelementet përbërëse në hyrje dhe dalje, parametrat, lloji i të dhënave dhe përshkrimi

‘SR flip flop’ është vendosur nëse gjendja e sinjalit është “1” në hyrje të S, dhe “0” në hyrje të R. Ndryshe, nëse gjendja e sinjalit është “0” në hyrje të S, dhe “1” në hyrje të R, atëherë ‘flip flop’ është rivendosur. Nëse RLO është “1” në të dy hyrjet mënyrë kjo më e veçantë, atëherë kjo mënyrë është me rëndësi primare. ‘SR flip flop’ së pari bën ekzekutimin e udhëzimit të vendosur, pastaj ky udhëzim rivendoset në <address> të caktuar, në mënyrë që kjo adresë të mbetet e rivendosur për pjesën e mbetur në skanimin e programit [5], [6].

SR tregon udhëzimet që janë realizuar vetëm kur RLO është në gjendjen e sinjalit “1”. RLO në gjendjen “0” nuk ka efekt në këto udhëzime, dhe adresa e përcaktuar në këtë udhëzim mbetet e pandryshuar.

**6.4.2. Thirr FB nga kutia (CALL\_FB)**

**Simboli**



Simboli varet nga FB (nëse ka parametra dhe sa janë ata). Duhet patjetër të ketë EN, ENO dhe emrin apo numrin e FB.

Parametrat	Lloji i të dhënave	Zona e memories	Përshkrimi
EN	BOOL	I,Q,M,L,D	Hyrja e lejushme
ENO	BOOL	I,Q,M,L,D	Dalja e lejushme
FB no.	BLOCK_FB	-	Numri i FB/DB; rresht i cili varet në CPU
DB no.	BLOCK_DB	-	

*Tabela 6.7. Parametrat, lloji i të dhënave, zona e memories dhe përshkrimi për bllokun FB.*

**6.4.3. Përshkrimi për thirrjen e bllokut FB**

**CALL\_FB** (Thirr bllokun e funksionit nga një kuti) ekzekutohet nëse EN është "1". Nëse CALL\_FB është e ekzekutuar:

- Adresa e rikthimit nga blloku i thirrjes është ruajtur,
- Të dhënat e selektuara për dy blloqet aktuale (DB dhe shembulli DB) janë ruajtur,
- Zona e të dhënave është rivendosur nga zona lokale aktuale,
- Biti MA (biti aktivë MCR) është zhvendos në raftin B,
- Zona e re lokale për funksionin e thirrjes është krijuar.

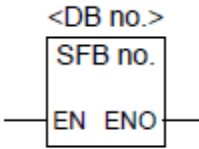
Pas kësaj, programi vazhdon mbrenda funksionit të thirrjes bllok. Biti BR është skenuar për të gjetur ENO. Përdoruesi duhet të caktoj gjendjen që kërkohet në bitin BR në thirrjen e bllokut aktual duke përdorur ---(SAVE).

**Gjendje e dhënë**

		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
Të pakufizuar	Shkrimet	x	-	-	-	0	0	X	x	x
Të kufizuar	Shkrimet	-	-	-	-	0	0	X	x	x

**6.4.4. Thirr bllok/sistemin FB nga kutia (CALL\_SFB)**

**Simboli**



Simboli varet nga SFB (nese ka parametra dhe sa janë ata). Duhet gjithsesi të ketë EN, ENO dhe emrin ose numrin e SFB.

Parametrat	Lloji i të dhënave	Zona e memorjes	Përshkrimi
EN	BOOL	I,Q,M,L,D	Hyrja e lejushme
ENO	BOOL	I,Q,M,L,D	Dalja e lejushme
SFB no.	BLOCK_SFB	-	Numri i SFB; rresht i cili
DB no.	BLOCK_DB	-	varet në CPU

*Tabela 6.8. Parametrat, lloji i të dhënave, zona e memories dhe përshkrimi për bllokun FB.*

**6.4.5. Përshkrimi për thirrjen SFB**

**CALL\_SFB** (Thirr sistemin e bllok funksionit nga kutia) Thirrja ekzekutohet nëse EN është “1”.

Nëse CALL\_SFB ekzekutohet,

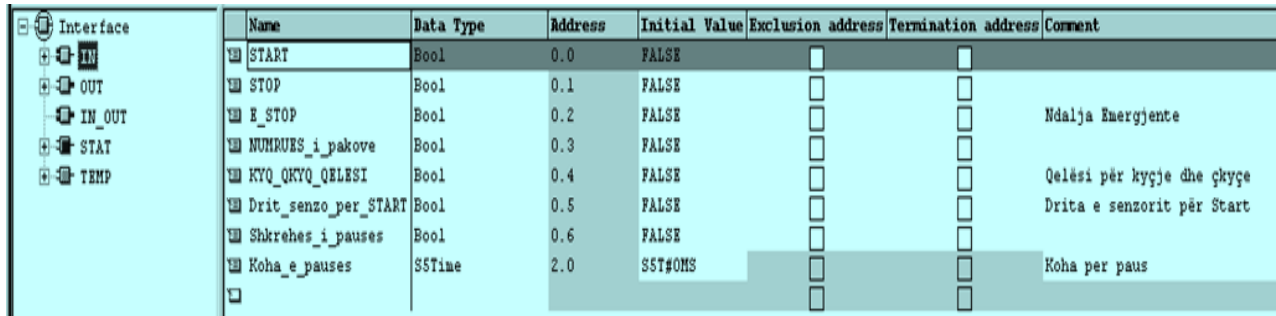
- Adresa e rikthimit të thirrjes së bllokut ruhet,
- Të dhënat e selektuara për dy blloqet e të dhënave (DB dhe shembulli i DB) janë të ruajtura,
- Zona e të dhënave të mëhershme zëvendësohet me zonen e të dhënave aktuale,
- Biti MA (biti MCR aktiv) zhvendoset në raftet B,
- Zona e re e të dhënave për thirrje të funksionit është krijuar.

Programi vazhdon më pas në thirrjen SFB. ENO është “1” nëse SFB është thirrur (EN=”1”) dhe asnjë gabim nuk ndodh.

**Gjendje e dhënë**

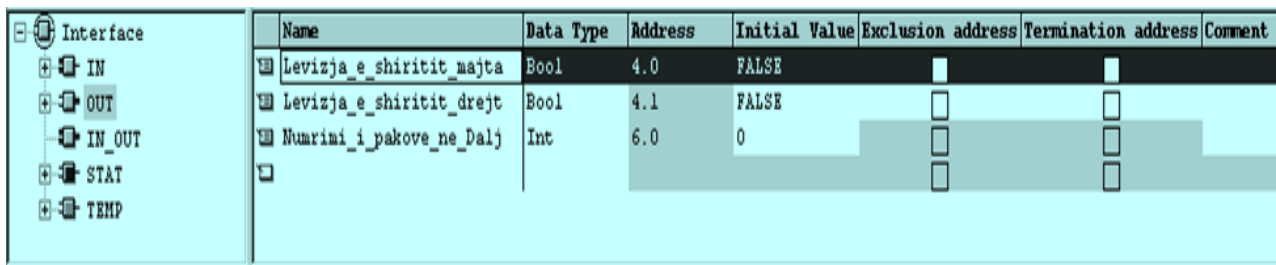
		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
Të pakufizuar	Shkrimet	x	-	-	-	0	0	x	x	x
Të kufizuar	Shkrimet	-	-	-	-	0	0	x	x	x

Tabelat në vazhdim tregojnë hyrjet dhe daljet në PLC, të cilat janë përdorur për programimin e sensorëve, si dhe programimin dhe komandimin e kontrollit manual.



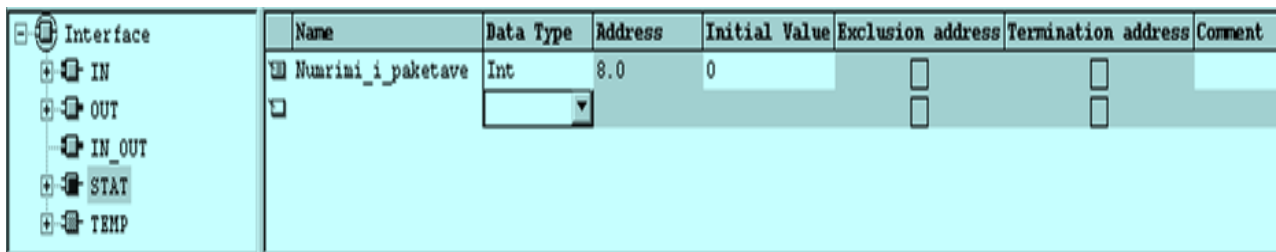
Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
START	Bool	0.0	FALSE			
STOP	Bool	0.1	FALSE			
E_STOP	Bool	0.2	FALSE			Ndalja Emergjente
NUMRUES_i_pakove	Bool	0.3	FALSE			
KYO_QKYO_QELESI	Bool	0.4	FALSE			Qelësi për kyçje dhe çkyçje
Drit_senzo_per_START	Bool	0.5	FALSE			Drita e sensorit për Start
Shkrehes_i_pauses	Bool	0.6	FALSE			
Koha_e_pauses	S5Time	2.0	S5T#0MS			Koha per paus

Tabela 6.9. Të dhënat programuese në hyrje të bllokut FBI.



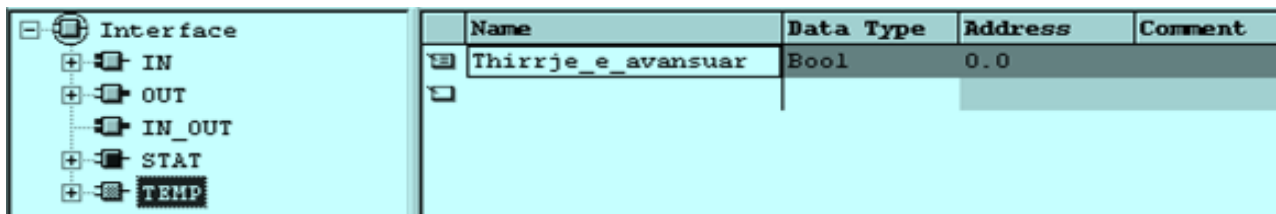
Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
Levizja_e_shiritit_majta	Bool	4.0	FALSE			
Levizja_e_shiritit_drejt	Bool	4.1	FALSE			
Numrimi_i_pakove_ne_Dalj	Int	6.0	0			

Tabela 6.10. Të dhënat programuese në dalje të bllokut FBI.



Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
Numrimi_i_paketave	Int	8.0	0			

Tabela 6.11. Tregon gjendjen e programuar të pandryshuar në numrim.



Name	Data Type	Address	Comment
Thirrje_e_avansuar	Bool	0.0	

Tabela 6.12. Tregon gjendjen e pandryshuar të thirrjes së shiritit.

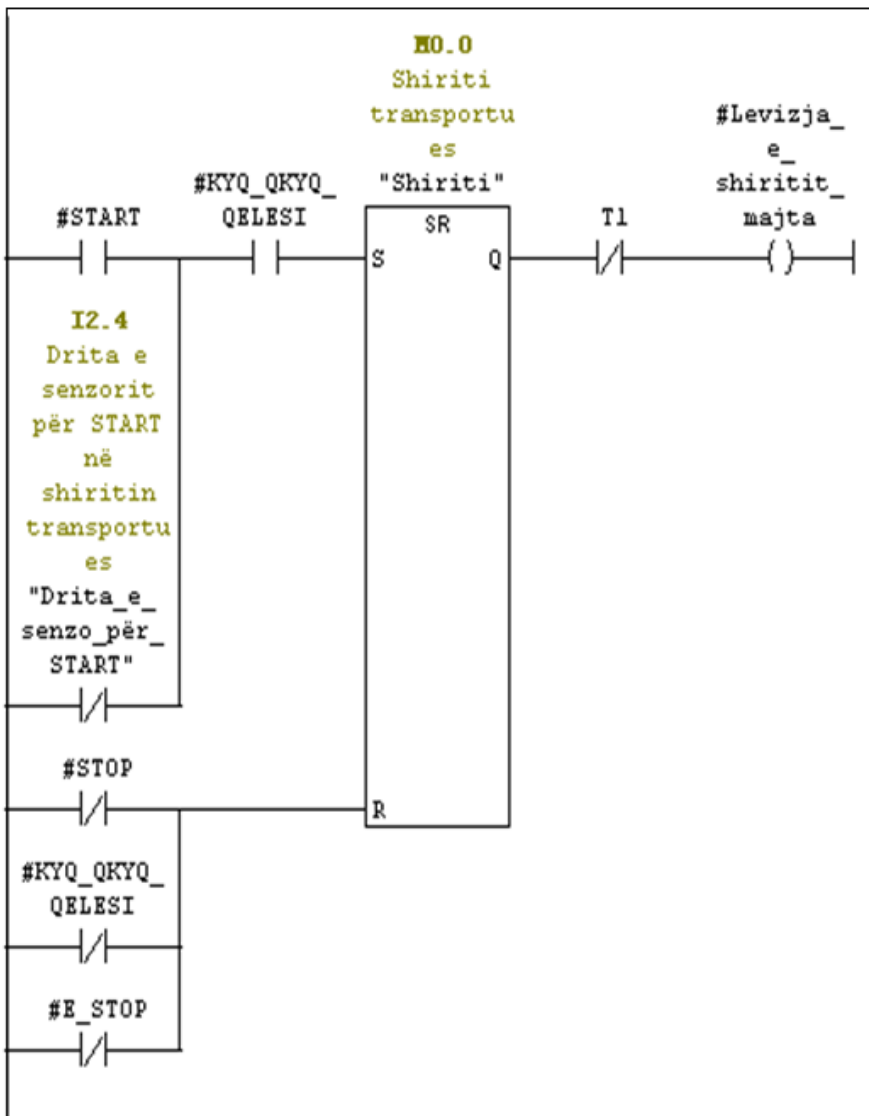
Nënprogrami FB1, i programuar në kuadër të programit kryesor OB1.

FB1 : Nënprogrami FB1

Programimi i shiritit transportues me të gjitha elementet funksionale, vënia në lëvizje e motorit, kyqja manuale përmes çelësit special si dhe butonit, sensorët, kohuesit etj.

**Qarku 1: Shiriti transportues**

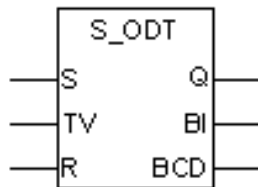
Komandimi përmes hyrje/daljeve, ku për secilin element është e nevojshme të kemi të dhëna të sakta të PLC-s për programim





**6.4.6. Kohuesi (Timer) T1**

Simboli



Parametrat	Lloji i të dhënave	Zona e memories	Përshkrimi
T no.	TIMER (KOHUES)	T	Identifikimi i numrit në kohues; rend i pavarur në CPU
S	BOOL	I,Q,M,L,D	Fillo hyrjen
TV	S5TIME	I,Q,M,L,D	Vlera në kohën paraprake
R	BOOL	I,Q,M,L,D	Rivendos hyrje
BI	WORD	I,Q,M,L,D	Vlera e mbetur në kohë, përmasa e numrit të plotë
BCD	WORD	I,Q,M,L,D	Vlera e mbetur në kohë, përmasat BCD
Q	BOOL	I,Q,M,L,D	Gjendja në kohues

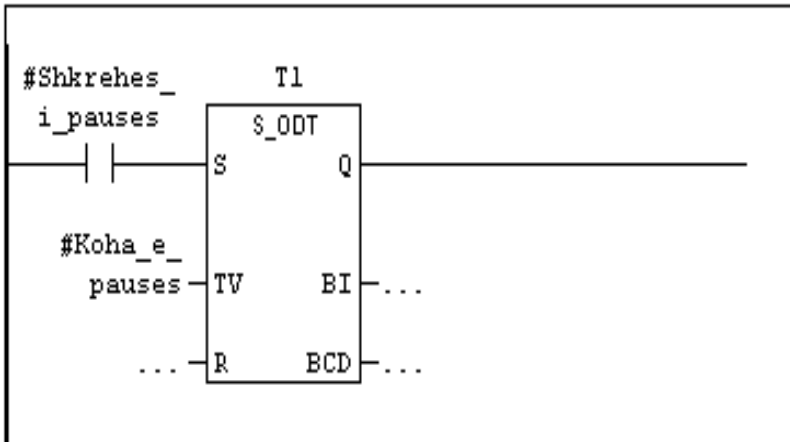
*Tabela 6.13. Parametrat, lloji i të dhënave, zona e memories si dhe përshkrimi*

**S\_ODT** (Në vonesën kohore S5 TIMER) përshkruan fillimin kohor të përcaktuar më lartë nëse ka përparësi pozitive në fillimin e hyrjes (S). Një ndryshim sinjali është gjithmonë i nevojshëm, për të mundësuar funksionimin me një kohues. Kohuesi vazhdon nëpër intervalin kohor të përcaktuar në hyrjen TV, për aq kohë sa gjendja në hyrje të sinjalit S është pozitive. Gjendja e sinjalit në daljen Q është “1”, kur kohuesi të ketë kaluar pa gabime dhe gjendja e sinjalit të dhënë S është ende “1”. Kur gjendja e sinjalit në hyrjen S ndryshon nga “1” në “0”, ndërsa kohuesi vazhdon sinjalin, kohuesi ndalet. Në këtë rast gjendja e sinjalit në daljen Q është “0”. [4], [5].

Kohuesi është rivendosur, në qoftë se elementi logjik rivendos (R), ndryshon në hyrje nga “0” në “1”, ndërsa kohuesi vazhdon drejtimin. Koha aktuale dhe koha bazë janë zero, ndërsa gjendja e sinjalit në daljen Q është “0”. Gjithashtu kohuesi është rivendosur nëse ka një logjikë “1” në hyrje të “R”, ndërsa kohuesi nuk ka vazhdim funksioni, dhe RLO në hyrje të S është “1”. Vlera e kohës aktuale mund të skanohet në daljet BI dhe BCD. Vlera kohore në BI është koduar si binar në kodimin BCD. Vlera e kohës aktuale është vlera fillestare në inicialet TV minus periudha kohore që ka filluar në kohues.

**Qarku 2: Kohuesi T1**

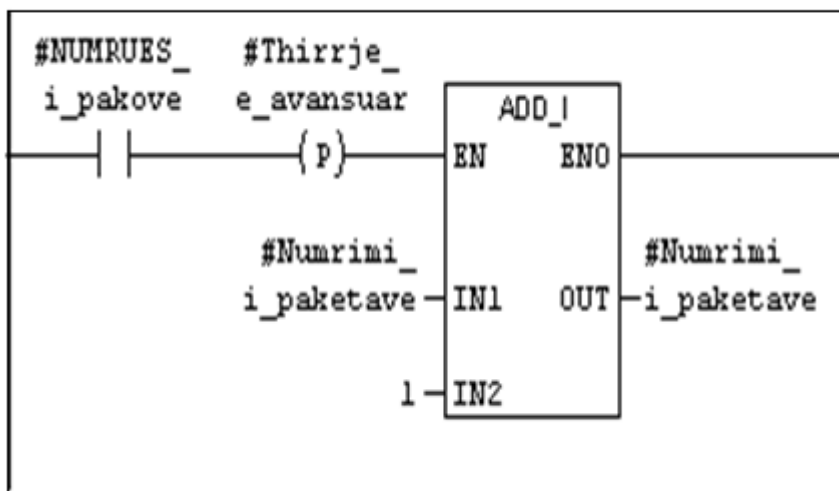
I cili shërben për programimin e pausave të nevojshme gjatë punës së shiritit transportues



Bloku i integruar funksional, kur kohuesi T1 është shuar.

**Qarku 3: Funksion i integruar**

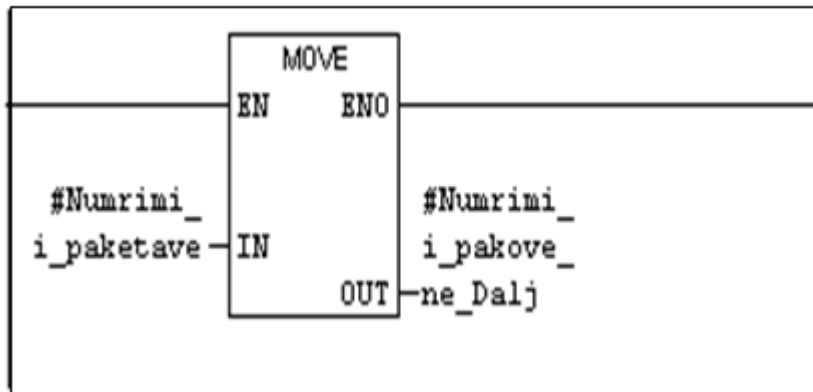
Funksioni i integruar, ku në të janë të instaluar: sensori për numërimin e pakove, si dhe thirrja e avansuar (P), për zbulimin e hershëm fillestar të pjesës.



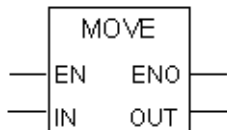
**6.4.7. Udhëzimi i MOVE (lëviz) i integruar në Blllokun FB1, na lejon që të prodhojmë hapësirë kohore me frekuenca të ndryshme në daljen Q12.0 përmes Q13.7.**

**Qarku 4: LËVIZ (Move), vlerë e caktuar**

Funksioni i cili ka një hyrje dhe një dalje, dhe i cili shërben për numërimin e pakove në hyrje dhe dalje të këtij funksioni.



Simboli



Parametrat	Lloji i të dhënave	Zona e memories	Përshkrimi
EN	BOOL	I,Q,M,L,D	Mundëso hyrjet
ENO	BOOL	I,Q,M,L,D	Mundëso daljet
IN	Të gjitha llojet elementare të të dhënave me gjatësi prej: 8,16 ose 32 bit	I,Q,M,L,D ose	Burimi i vlerës
OUT	Të gjitha llojet elementare të të dhënave me gjatësi prej: 8,16 ose 32 bit	Konstant	Përcaktim i adresës

*Tabela 6.14. Tregon gjendjen e dhënë në simbloin MOVE me të gjith parametrat e dhënë.*

**MOVE** (Vlerë e caktuar) është aktivizuar nga (Enable EN Input). Vlera e përcaktuar në hyrjen IN, kopjohet në adresën e përcaktuar në daljen OUT. ENO ka të njëjtën logjikë sikurse gjendja EN [6],[7].

Me simbolin MOVE mund të kopjojm vetëm BYTE, ËORD, DËORD apo objekte të dhënash. Përdoruesi i definuar për llojin e të dhënave si varg apo vektor, duhet të jet e kopjuar me funksionin e sistemit “BLKM” (SFC 20).

Gjendja e dhënë

	BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC
Shkrimet	1	-	-	-	-	0	1	1	1

#### 6.4.8. Sensorët detektues të programuar në bllokun FB2

Paraqitja e të dhënave në hyrje dhe dalje të PLC-s për funksionimin e sensorit detektues, i cili shërben për detektimin e pjesëve që kalojnë nëpër shiritin transportues, dhe njëkohësisht shërben për startimin e motorit në mënyrë automatike.

Të dhënat hyrëse programuese të sensorit induktiv IN:

Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
Starti_shtytes_Cil	Bool	0.0	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
Dalja_e_Cil_jasht	Bool	0.1	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
Futja_e_Cil_brenda	Bool	0.2	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	

Tabela 6.15. Paraqitja e të dhënave hyrëse për programimin e sensorit induktiv II.

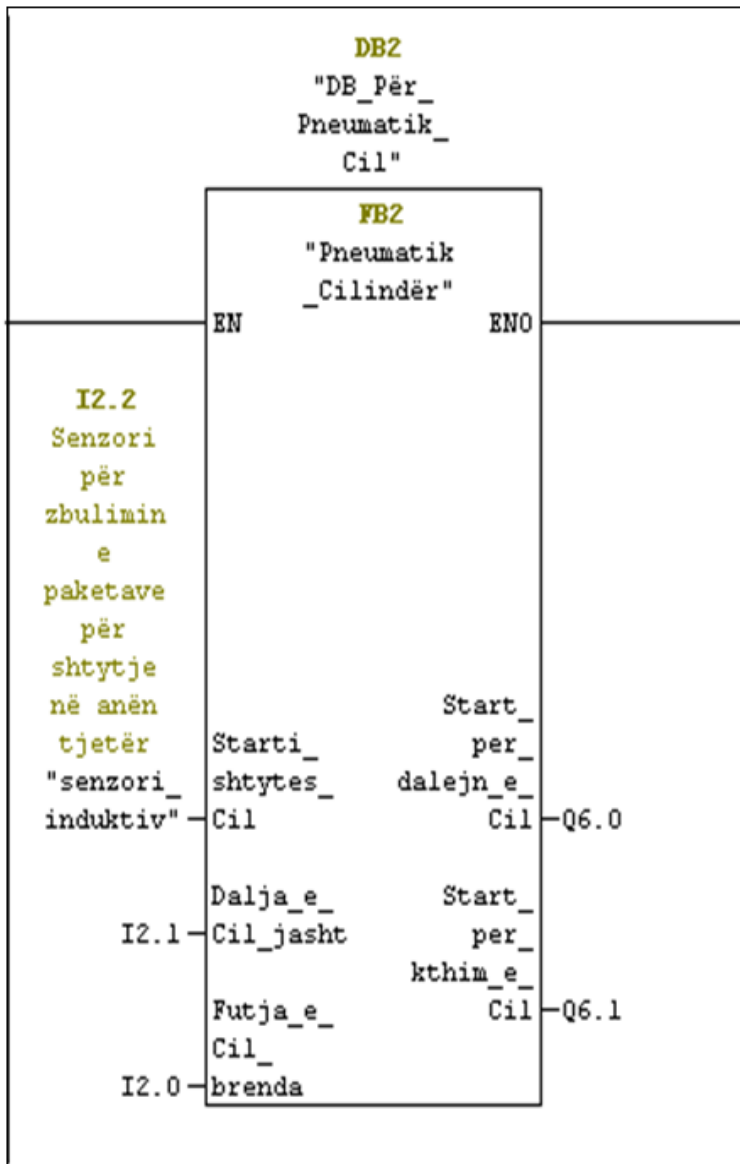
Të dhënat dalëse programuese të sensorit induktiv Out:

Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
Start_per_dalejn_e_Cil	Bool	2.0	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
Start_per_kthin_e_Cil	Bool	2.1	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	

Tabela 6.16. Paraqitja e të dhënave dalëse për programimin e sensorit induktiv II.

**6.4.9. Blloku funksional FB2 i integruar në bllokun e të dhënave DB2 në kuadër të bllokut kryesorë OBI**

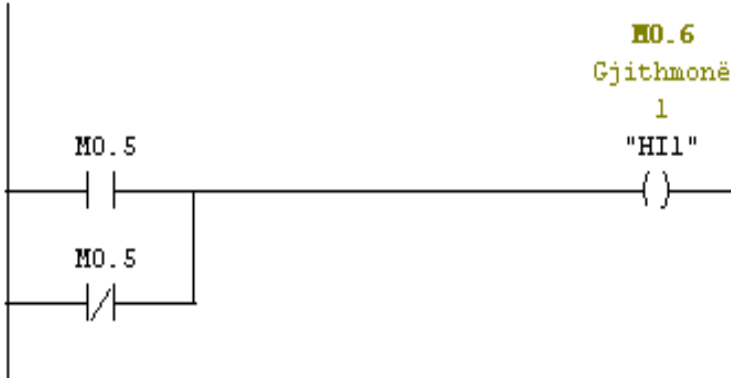
**Qarku 2: Blloku i të dhënave DB2, ku është i integruar edhe blloku FB**



**6.4.10. Gjendja e sinjalit në motor**

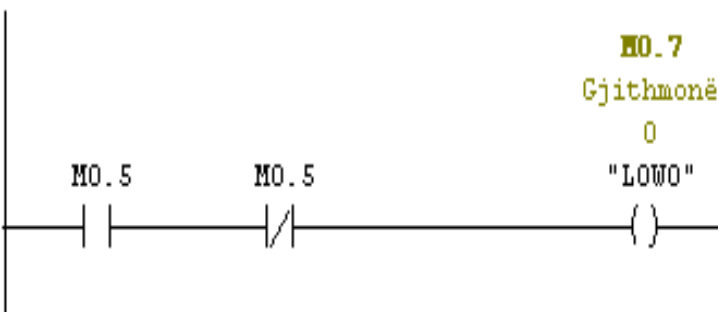
**Qarku 3:** Sinjali gjithmonë 1

Gjendja e sinjalit në motor gjithmonë 1.



**Qarku 4:** Sinjali gjithmonë 0

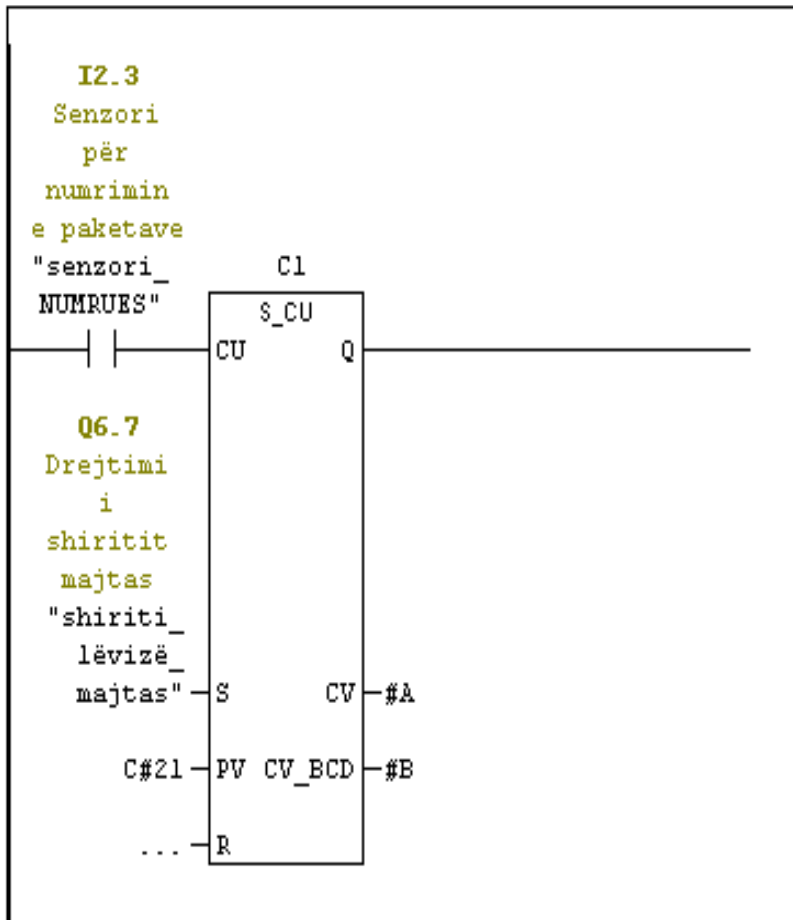
Gjendja e sinjalit në motor gjithmonë 0.



**6.4.11. Numruesi (S\_CU) i cili shërben për integrimin e sensorit numërues, si dhe qëndrimi në lëvizje të pandërprerë i shiritit transportues.**

**Qarku 5: Numruesi**

**Numëruesi ka për detyrë që përmes sensorit të bëjë numërimin e pjesëve që kalojnë Nëpër shiritin transportues**



**6.5. PROGRAMIMI I SENSORIT INDUKTIV I INTEGRUAR NË BLOKUN FUNKSIONAL FB2.**

FB2 : Blloku Funkcional FB2 për funksionimin e sensorëve për detektim

**Detektimi i detalit me sensorin induktiv 2.**

**Qarku 1: Sensori I**

Detektimi i detalit nga senzori induktiv II, ku përveç detektimit të detaleve në shirit përmes sensorit futet në funksion edhe cilindri pneumatik për largimin e pjesës nga SHIRITI TRANSPORTUES dhe vendosja e pjesëve në paketim ose shirit tjetër pune.



**Qarku 2: Sensori II**

Sensori që shërben për startimin e cilindrit pneumatik, shtytjen e cilindrit për daljen jashtë të tij, largimin e pjesës, si dhe kthimi mbrapa i cilindrit.





**6.5.1. Shembulli DB (bllok i të dhënave)**

Një shembull i bllokut të të dhënave DB është caktuar për secilin bllok funksion që transferon parametra. Parametrat aktualë dhe të dhënat e pandryshuara në bllokun FB, janë ruajtur në shembullin DB. Variablat e deklaruara në FB determinojnë ose përcaktojnë në shembullin e bllokut të të dhënave apo DB. Shembulli nënkupton një funksion bllok të thirrur. Nëse për shembull, një funksion bllok thirret pes herë në programin e përdoruesit S7, ka pes raste në këtë bllok. [4], [6], [7].

**6.5.2. Krijimi themelorë i një blloku DB**

Para se të krijohet një bllok të të dhënave DB, duhet të ekzistohet korrespondimi FB.

Gjithashtu duhet specifikuar numrin FB kur të krijohet një bllok i të dhënave.

Të dhënat për bllokun DB1: adresat, deklaratimet, emrat, tipet, vlerat fillestare dhe komentimi.

Address	Declaration	Name	Type	Initial value	Comment
0.0	in	START	BOOL	FALSE	
0.1	in	STOP	BOOL	FALSE	
0.2	in	E_STOP	BOOL	FALSE	Ndalja Emergjente
0.3	in	NUMRUES_i_pakove	BOOL	FALSE	
0.4	in	KYQ_QKYQ_QELESI	BOOL	FALSE	Qelësi për kyçje dhe çkyçje
0.5	in	Drit_senzo_per_START	BOOL	FALSE	Drita e sensorit për Start
0.6	in	Shkrehes_i_pauses	BOOL	FALSE	
2.0	in	Koha_e_pauses	S5TIME	S5T#0MS	Koha per paus
4.0	out	Levizja_e_shiritit_majta	BOOL	FALSE	
4.1	out	Levizja_e_shiritit_drejt	BOOL	FALSE	
6.0	out	Numrimi_i_pakove_ne_Dalj	INT	0	
8.0	stat	Numrimi_i_paketave	INT	0	

Tabela 6.17. Deklarimi, emri, adresa, tipi, vlera fillestare, komentimi.

Tabela e të dhënave në bllokun DB1, hyrjet dhe daljet.

Address	Declaration	Name	Type	Initial value	Comment
0.0	in	Starti_shtytes_Cil	BOOL	FALSE	
0.1	in	Dalja_e_Cil_jasht	BOOL	FALSE	
0.2	in	Futja_e_Cil_brenda	BOOL	FALSE	
2.0	out	Start_per_dalejn_e_Cil	BOOL	FALSE	
2.1	out	Start_per_kthim_e_Cil	BOOL	FALSE	

Tabela 6.18. Deklarimi, emri, adresa, tipi, vlera fillestare, komentimi.

## **6.6. BLOKU FUNKSIONAL FCs**

Funksionet (FCs) iu përkasin blloqeve që mund t'i programojmë vet. Një funksion është një bllok logjik “pa memorie”. Variablat e përkohëshme që i përkasin një blloku FC, janë të ruajtura në tabelën e të dhënave programuese. Këto të dhëna pastaj janë humbur kur blloku FC ka qenë i ekzekutuar. Për t'i ruajtur të dhënat e përgjithshme, funksionet mund të përdoren gjithashtu edhe në blloqe të përbashkëta.

Kurë një bllok FC nuk ka një memorie të veten, gjithmonë duhet t'i specifikojmë parametrat aktualë për të. Gjithashtu vlen të theksohet se nuk mund të caktohen vlera fillestare për të dhënat lokale të një blloku FC.

### **6.6.1. Aplikimi**

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 (shembll: 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, jan ruajtur si tregues në parametrat aktual të bllokut logjik që quhet FC.

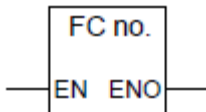
### **6.6.2. Dallime të rëndësishme në mes të parametrave të jashtëm FCs dhe FBs**

Në blloqet e funksionit FB një kopje e parametrave aktual në momentin e DB është përdorur gjatë qasjes së parametrave. Në qoftë se një parametër hyrës nuk është transferuar, ose një parametër i prodhimit nuk është shkruar në hyrje, kur një FB është thirrur dhe vlerat më të vjetra janë ende të ruajtura në qasjen DB, atëherë do të përdoret memoria FBs. Vlenë të theksohet gjithashtu, se funksionet FC nuk kan kujtesë. Në kundërshtim me FBs, për këtë arsye caktimi i parametrave formal të këtyre FCs, nuk është opsional por është thelbësor. Parametrat FC janë të arritshme nëpërmjet adresave ( treguesve, ose objekteve në të gjithë kufijt e zonës).

Kur një adresë e gjendjes së të dhënave (bllok i të dhënave), ose një variabel i quajtur bllok, është përdorur si një parametër aktual, një kopje e parametrin aktual është ruajtur përkohësisht në zonën lokale të të dhënave të bllokut, duke bërë thirrje për transferimin e parametrave.

### 6.6.3. Thirrja bllokut FC nga kutia, (CALL\_FC)

#### Simboli



Simboli varet nga FC (nese ka parametra dhe sa janë ata). Duhet gjithsesi të ketë EN, ENO dhe emrin apo numrin e FC-së.

Parametrat	Lloji i të dhënave	Zona e memorjes	Përshkrimi
EN	BOOL	I,Q,M,L,D	Hyrja e lejushme
ENO	BOOL	I,Q,M,L,D	Dalja e lejushme
FC no.	BLOCK_FC	-	Numri i FC; rresht i cili varet në CPU

Tabela 6.19. Parametrat, lloji i të dhënave, zona e memories, si dhe përshkrimi për bllokun FC.

#### Përshkrimi

**CALL\_FC** (Thirr një funksion nga kutia) përdoret për të thirrur një funksion (FC). Thirrja ekzekutohet nëse EN është "1".

Nëse CALL\_FC ekzekutohet:

- Adresa e rikthimit të thirrjes së bllokut ruhet,
- Zona e të dhënave të mëhershme zëvendësohet me zonën e të dhënave aktuale,
- Biti MA (biti MCR aktiv) zhvendoset në raftet B,
- Zona e re e të dhënave për thirrje të funksionit është krijuar.

Pas kësaj, programi vazhdon në funksionin e thirrjes.

Biti BR është skanuar me qëllim për të gjetur ENO. Përdoruesi duhet të caktojë gjendjen që kërkohet në bitin BR në thirrjen e bllokut duke përdorur --- (**SAVE**).

Nëse e thirr një funksion dhe variablat e deklaruara në tabelën e bllokut të thirrjeve ka IN, OUT dhe IN\_OUT, këto variabla janë paraqitur në funksion për thirrjen e bllokut si një list e parametrave formal.

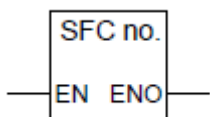
Kur thirret funksioni, duhet të caktuar parametrat aktual në parametrat formal në lokacionin e thirrjeve. Ndonjë vlerë fillestare në deklarimin e funksionit nuk ka domethënje.

**Gjendje e dhënë**

		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
Të pakufizuar	Shkrimet	x	-	-	-	0	0	x	x	x
Të kufizuar	Shkrimet	-	-	-	-	0	0	x	x	x

**6.6.4. Thirr sistemin FC nga kutia (CALL\_SFC)**

**Simboli**



Sistemi varet nga SFC (nëse ka parametra dhe sa janë ata). Duhet gjithësesi të ketë EN, ENO dhe emrin ose numrin e SFC.

Parametrat	Lloji i të dhënave	Zona e memorjes	Përshkrimi
EN	BOOL	-	Hyrja e lejushme
ENO	BOOL	-	Dalja e lejushme
SFC no.	BLOCK_SFB	-	Numri i SFC; rresht i cili varet në CPU

*Tabela 6.20. Parametrat, lloji i të dhënave, zona e memories, si dhe përshkrimi për bllokun FC.*

**Përshkrimi**

**CALL\_SFC** (Thirr sistemin e funksionit nga kutia), përdoret për të thirrur një SFC. Thirrja realizohet nëse EN është “1”.

Nëse thirrja CALL\_SFC ekzekutohet:

- Adresa e rikthimit të thirrjes së bllokut ruhet,
- Zona e të dhënave të mëhershme zëvendësohet me zonen e të dhënave aktuale,
- Biti MA (biti MCR aktiv) zhvendoset në raftet B,
- Zona e re e të dhënave për thirrje të funksionit është krijuar.

Pas kësaj, program vazhdon në thirrjen SFC. ENO është “1” nëse SFC është thirrur (EN = ”1”) dhe asnjë gabim nuk mund të ndodh.

**Fjalët e gjendjes**

		BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
Të pakufizuar	Shkrimet	x	-	-	-	0	0	x	x	x
Të kufizuar	Shkrimet	-	-	-	-	0	0	x	x	x

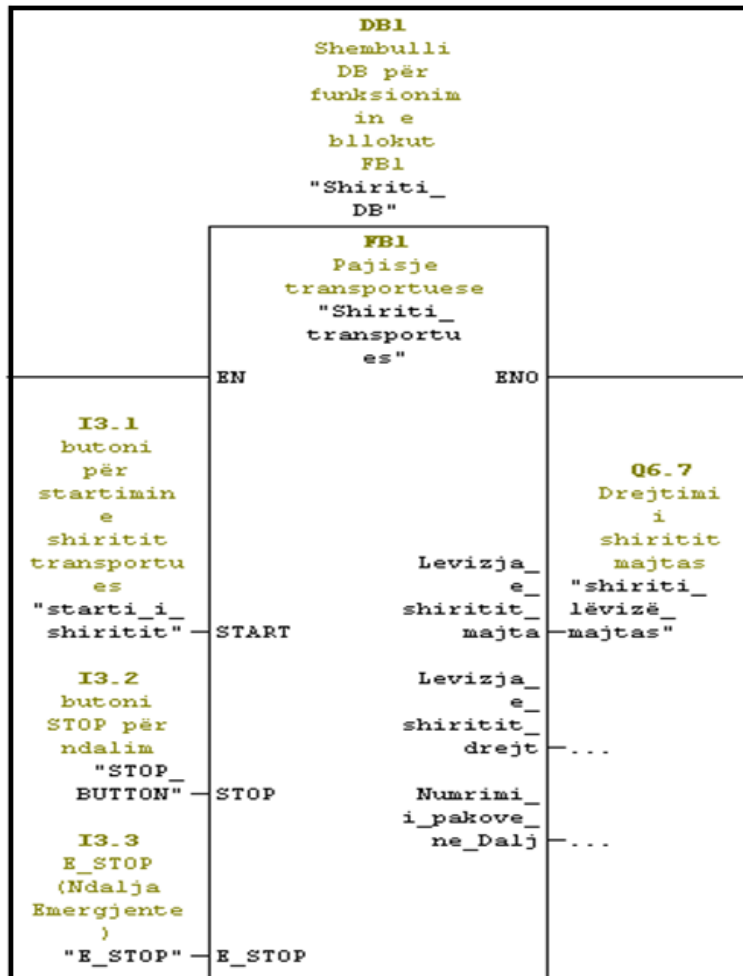
**6.6.5. Blloku Funkcional FC1**

FC1 : Blloku Funkcional FC1

FC1 ka të integruar brenda vehtës "bllokun e të dhënave DB1"

**Qarku 1: Integrimi i Bllokut Funkcional FB1 përmes bllokut të të dhënave DB1**

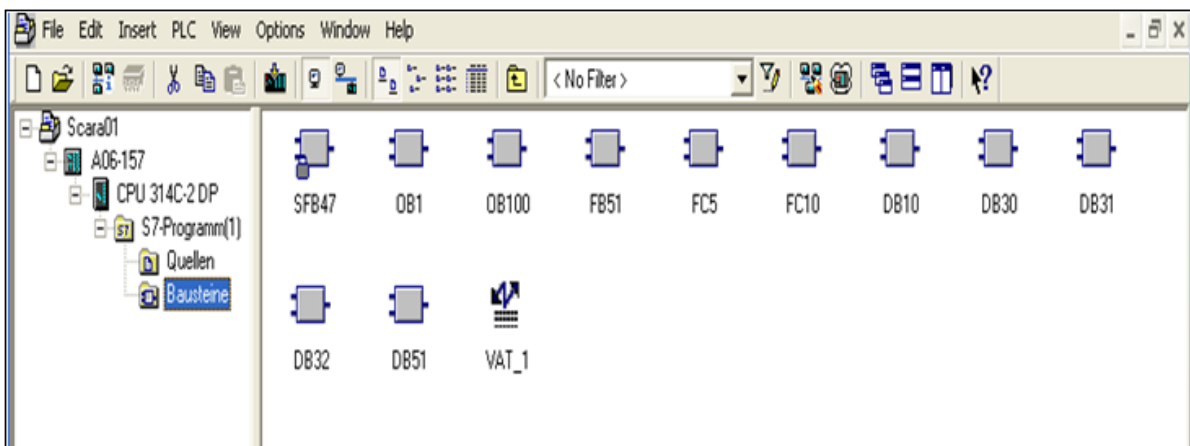
Shiriti transportues duke përdorur integrimin e blloqeve funksionale.



<p><b>I2.3</b> Senzori për numrimin e paketave "senzori_ NUMRUES"</p>	<p>NUMRUES_ i_pakove</p>
<p><b>I3.0</b> çelësi_i_s hiritit për START dhe STOP "çelësi_i_ shiritit_ transp"</p>	<p>KYQ_QKYQ_ QELESI</p>
<p><b>I2.4</b> Drita e senzorit për START në shiritin transportu es "Drita_e_ senzo_për_ START"</p>	<p>Drit_ senzo_ per_START</p>
<p><b>I2.2</b> Senzori për zbulimin e paketave për shtytje në anën tjetër "senzori_ induktiv"</p>	<p>Shkrehes_ i_pauses</p>
<p>SST#200MS</p>	<p>Koha_e_ pauses</p>

### 6.7 Procesi i programimit të robotit Skara

Programimi i robotit skara përmes programit SIMATIC S7-300 nga Logo Siemens



**Njësitë kryesore përbërëse të programit për robotin SKARA janë:**

**OB1 Programi organizativ kryesor i funksionit të robotit Skara**

**OB100 Nënprogrami organizativ kryesor i robotit Skara**

**FB51 Blloku Funkcional**

**FC5 Funksioni**

**FC10 Nën Funksioni**

**DB10 Blloku i të dhënave**

**DB30 Nënblloku i data bllokut**

**DB31 Nënblloku i data bllokut**

**DB32 Nënblloku i data bllokut**

**DB51 Nënblloku i data bllokut**

**SFB47 Sistemi i blloqeve funksionale**

**VAT\_1 Tabela e variablave**

## **6.8 OB1 Programi organizativ kryesor i funksionit të robotit Skara**

### **6.8.1 Përshkrimi i ciklit të programit të bllokut organizativ OB1**

Sistemi operativ i S7 CPU ekzekuton OB1 periodikisht, kur OB1 është ekzekutuar sistemi operativ starton atë përsëri. Cikli i ekzekutimit të OB1 starton pasi që set up-i të kompletohet. Mund të thërrasim gjithashtu bllokun funksional (FB, SFB) ose funksionet (FC, SFC) nga OB1.

Kuptimi i operacionit OB1:

OB1 ka prioritet më të ulët nga të gjitha OB-të tjerë, koha e punës së të cilëve është e monitoruar. Me fjalë të tjera, të gjitha OB-të e tjera përveç OB90 mund të ndërprejnë ekzekutimin e OB1.

## **6.9. OB100 Nënprogrami kryesor organizativ i robotit SKARA**

### **6.9.1 Startimi i bllokut organizativ OB100**

Mundësitë e startimit- dallohen këto tipe të startimit:

- Startimi i nxehtë (jo me S7- 300 dhe S7-400 H)

- Startimi mesatarisht ngrohtë
- Startimi i ftohtë.

Në tabelën në vijim, mund të shihet cili OB thirret nga sistemi operativ gjatë fillimit të startimit të robotit Skara S7-300:

*Tab. 6.21. Thirrjet e OB-ve*

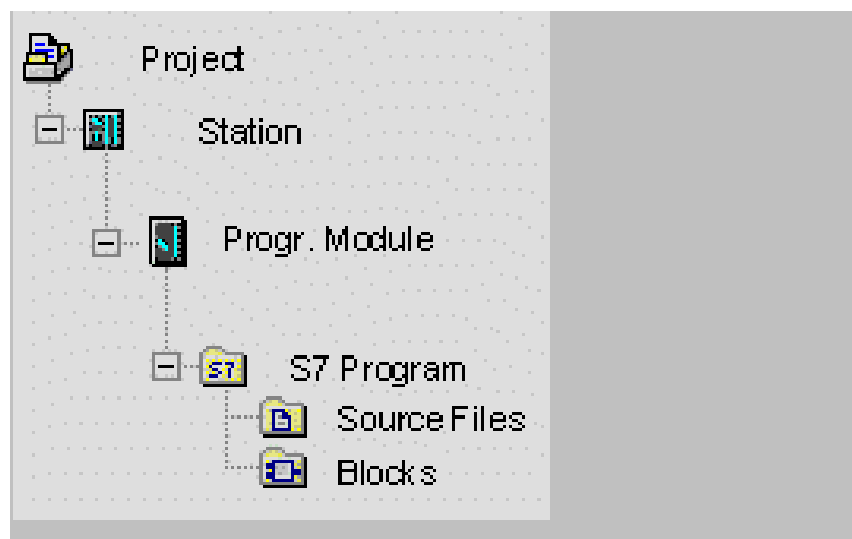
Tipi i startimit	OB përkatëse
Startimi i nxehtë	OB 101
Startimi mesatarisht i ngrohtë	OB 100
Startimi i ftohtë	OB 102

## **6.10. FB51 Blloku FunkSIONAL**

### **6.10.1 Objektet dhe hierarkia e objekteve**

Në të njejtën mënyrë sikurse Windows Explorer që tregon strukturën e folderave dhe fajlave, këtu SIMATIC MENAGER tregon hierarkinë për projektet dhe libraritë në STEP 7.

P.sh., një hierarki e objekteve tregohet në figurë:



*Fig. 6.2. Hierarkia e objekteve.*



Objektet kanë këto funksione:

1. Bartjen (transportin) e objekteve ekzistuese
  2. Folderët
  3. Bartjen e funksioneve (p.sh., startimin e një aplikacioni të pjesëshëm)
- 1) Përmes bartjes së objekteve ekzistuese mund të barten gjithashtu edhe funksionet. Kur klikojmë një objekt, mund të kryejmë këto funksione:
    - (Edit) Redaktimin e një objekti duke përdorur **Edit > open object**
    - Hapjen e një kutie në të cilën mund të shkruajmë opsionet specifike, kjo realizohet përmes **Edit > Object properties**
  - 2) Objektet si folderë- Një folder ( kryesor ) mund të përmbajë folderë ose objekte tjera të cilat mund të shihen kur hapet folderi kryesor.
  - 3) Startimi i pjesëshëm i një aplikacioni mund të bëhet përmes bartjes së funksioneve.

### 6.10.2 DB Bllok i të dhënave

Një shembulli i bllokut të të dhënave DB është caktuar për secilin bllok funksion që transferon parametra. Parametrat aktualë dhe të dhënat e pandryshuara në bllokun FB, janë ruajtur në shembullin DB. Variablat e deklaruara në FB determinojnë ose përcaktojnë në shembullin e bllokut të të dhënave apo DB. Shembulli nënkupton një funksion bllok të thirrur. Nëse për shembull, një funksion bllok thirret pesë herë në programin e përdoruesit S7, ka pesë raste në këtë bllok.

### 6.10.3 Krijimi themelor i një blloku DB

Para se të krijohet një bllok të të dhënave DB, duhet të ekzistojë korrespondimi FB. Gjithashtu duhet specifikuar numrin FB kur të krijohet një bllok i të dhënave.

### 6.10.4 FC Blloku Funksional

Funksionet (FCs) iu përkasin blloqeve që mund t'i programojmë vetë. Një funksion është një bllok logjik "pa memorie". Variablat e përkohëshme që i përkasin një blloku FC, janë të ruajtura në tabelën e të dhënave programuese. Këto të dhëna pastaj janë humbur kur blloku FC ka qenë i ekzekutuar. Për t'i ruajtur të dhënat e përgjithshme, funksionet mund të përdoren gjithashtu edhe në blloqe të përbashkëta.

Kur një bllok FC nuk ka memorie të vetën, gjithmonë duhet t'i specifikojmë parametrat aktualë për të. Gjithashtu vlen të theksohet se nuk mund të caktohen vlera fillestare për të dhënat lokale të një blloku FC.

## **6.11 SFB47 Sistemi i blloqeve funksionale**

### **6.11.1 Kontrollimi i numëruesve me SFB47 “COUNT”**

Përdoret për të kontrolluar funksionet e pozicionuara duke e përdorur programin e përdoruesit, me SFB COUNT (SFB 47).

Kemi këto operacione:

- Startimi/ndalimi i numërimit në hyrjen e softwerit SW\_GATE
- Mundësia (lejimi)/kontrollimi i daljeve DO
- Korigjimi i gjendjes së bitëve STS\_CMP, STS\_OFLW, STS\_UFLW dhe STS\_ZP
- Korigjimi i vlerës aktuale të numëruesit COUNT VAL
- Puna për leximin/shkrimin e regjistrave numërues të brendshëm
- Korigjimi i periodës aktuale (kohëzgjatja) TIME VAL.

## KAPITULLI VII

### ***7. Programimi i lëvizjes të robotit skara përmes PLC Simatic S7-300***

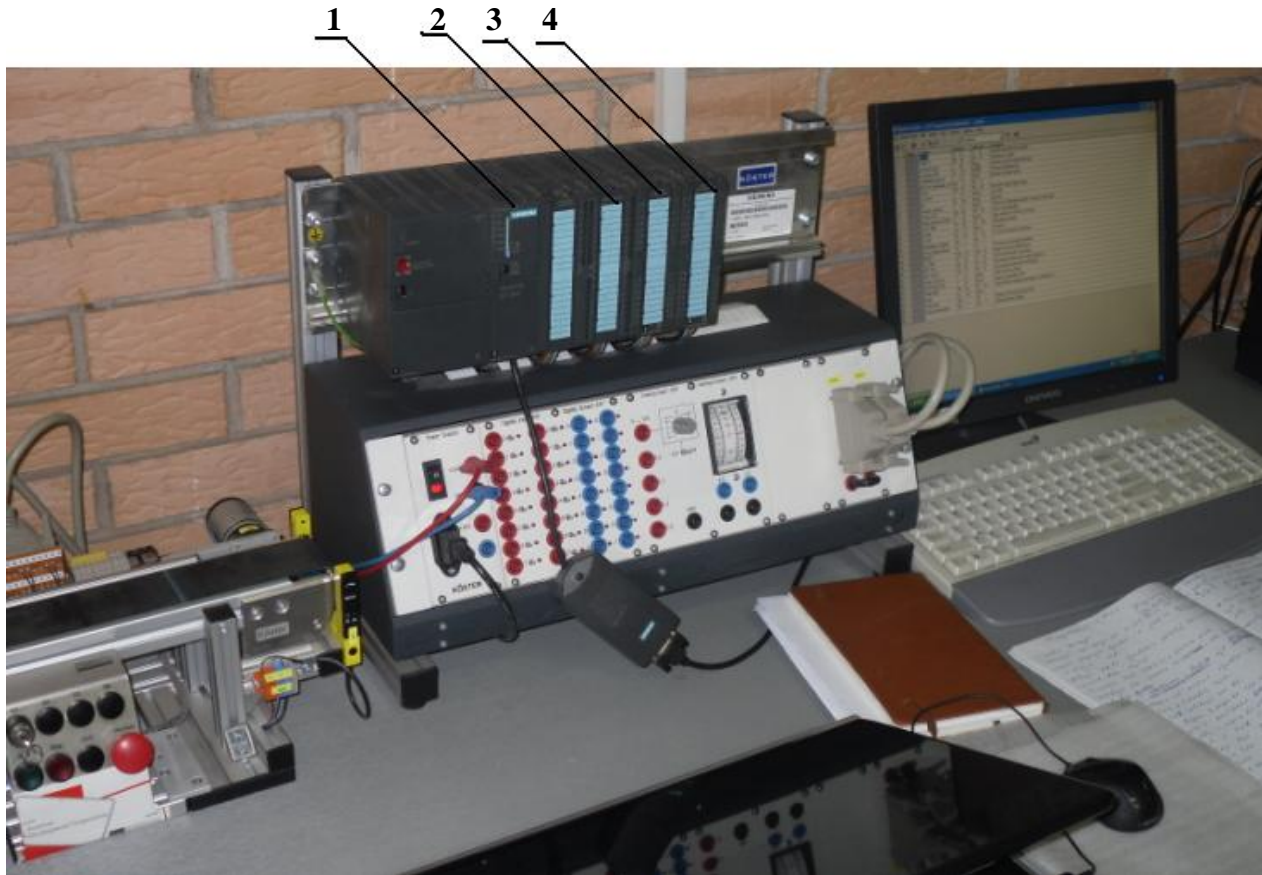
Instalimi mund të shërbejë si krahë robotik me një elektromagnet, i cili përdoret për kalimin e pjesëve dhe të elementeve metalike nga një shirit transportues në një paletë depozituese, e cila mundëson ngarkimin më të lehtë dhe më të shpejtë të pjesëve apo elementeve në një kamion transportues. Sistemi është vendosur të funksionojë automatikisht dhe të ngarkojë plotësisht paletën vetëm me 7 kalime në 3 pozicione të hedhjes ( shkarkimit ), shkarkimi duhet të bëhet tek secili nga këto pozicione, për të shpërndarë ngarkesën në paletën depozituese dhe pastaj të bëhet ngarkimi më i lehtë dhe më i shpejtë në kamion për t'u transportuar deri në vendin e caktuar për treg [4],[6].

Roboti SKARA punon me sistemin elektro pneumatik, lëvizja e shiritit bëhet me sistem elektrik, largimi i pjesës nga shiriti bëhet me sistem pneumatik, ndërsa largimi i pjesës nga baza e shiritit deri në paletën depozituese kryhet me ndihmën e krahut robotik i cili lëvizjet i kryen me sistemin elektrik, ndërsa shtërngimi bëhet me sistem pneumatik.



*Fig.7.1 Roboti Skara.*

*Në Figurën 7.2. kemi paraqitur PLC-të që kemi përdorur për hyrje në funksion (punë) të Shiritit Transportues dhe Robotit Skara, PLC-ja në figurë përbëhet nga 4 PLC.*



*Fig. 7.2. Pamja e PLC*

## ***7.1 OB1 Programi organizativ kryesor i funksionit të robotit Skara***

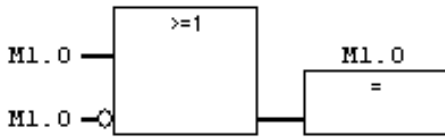
### ***7.1.1 OB1 “Qarku i programit kryesor (cikli)”***

Hapi 1: Startimi i faqes kryesore me çelësin S1.

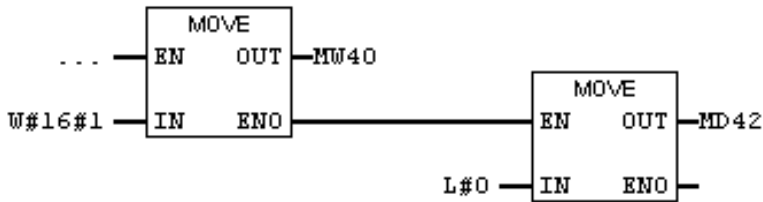
Hapi 2: Shtype butonin Start dhe mbaje të shtypur 1 sek, boshti është në lëvizje, bëj kujdes.

Bëj kujdes boshti krijon një hapësirë të gjerë të lëvizjes, dhe kjo është vetëm një test i programit.

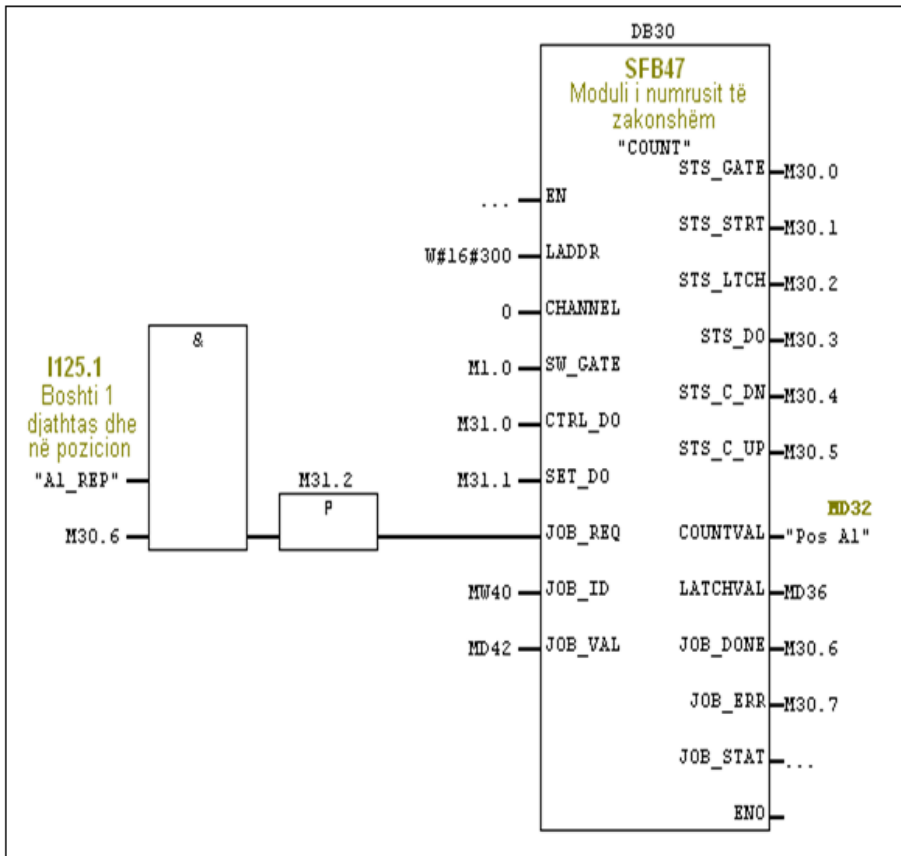
**Qarku 1:** Boshti statik



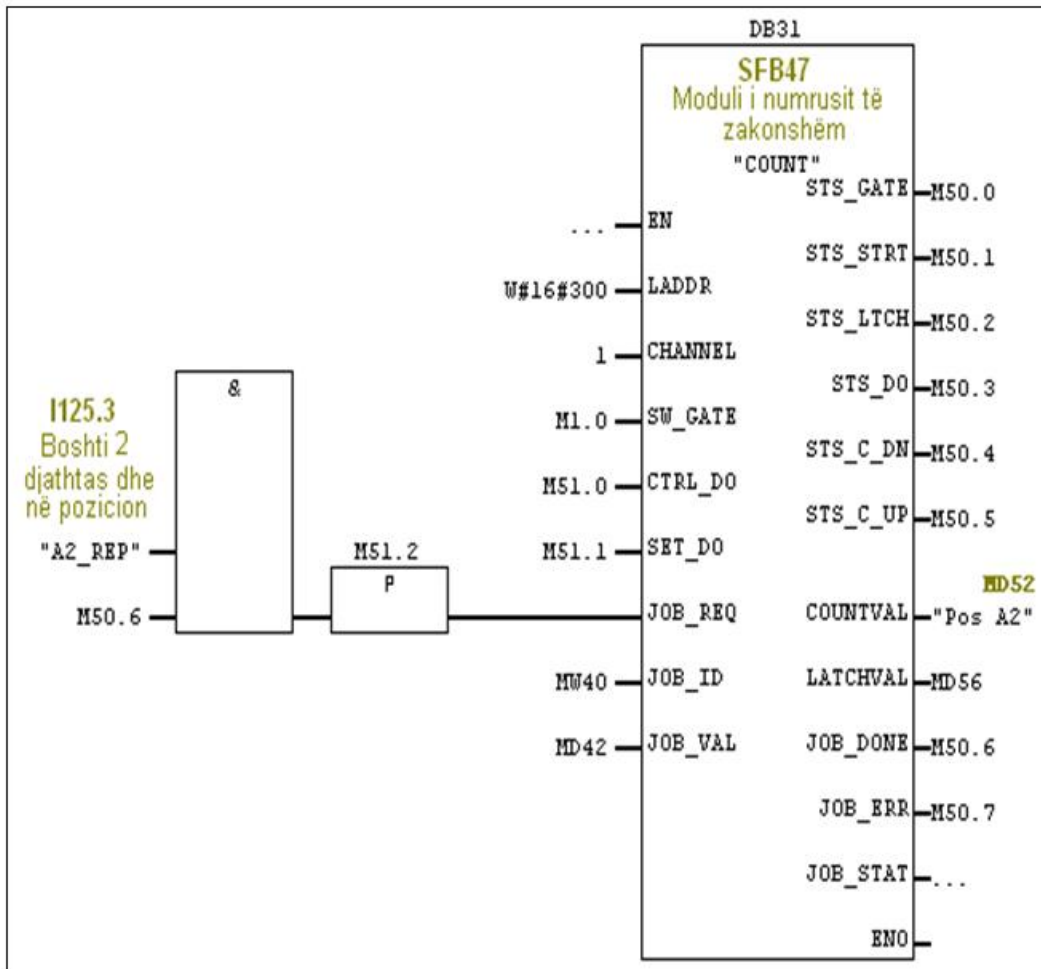
**Qarku 2:** Veprimi 1 numëruesi duke filluar nga vlera referente 0



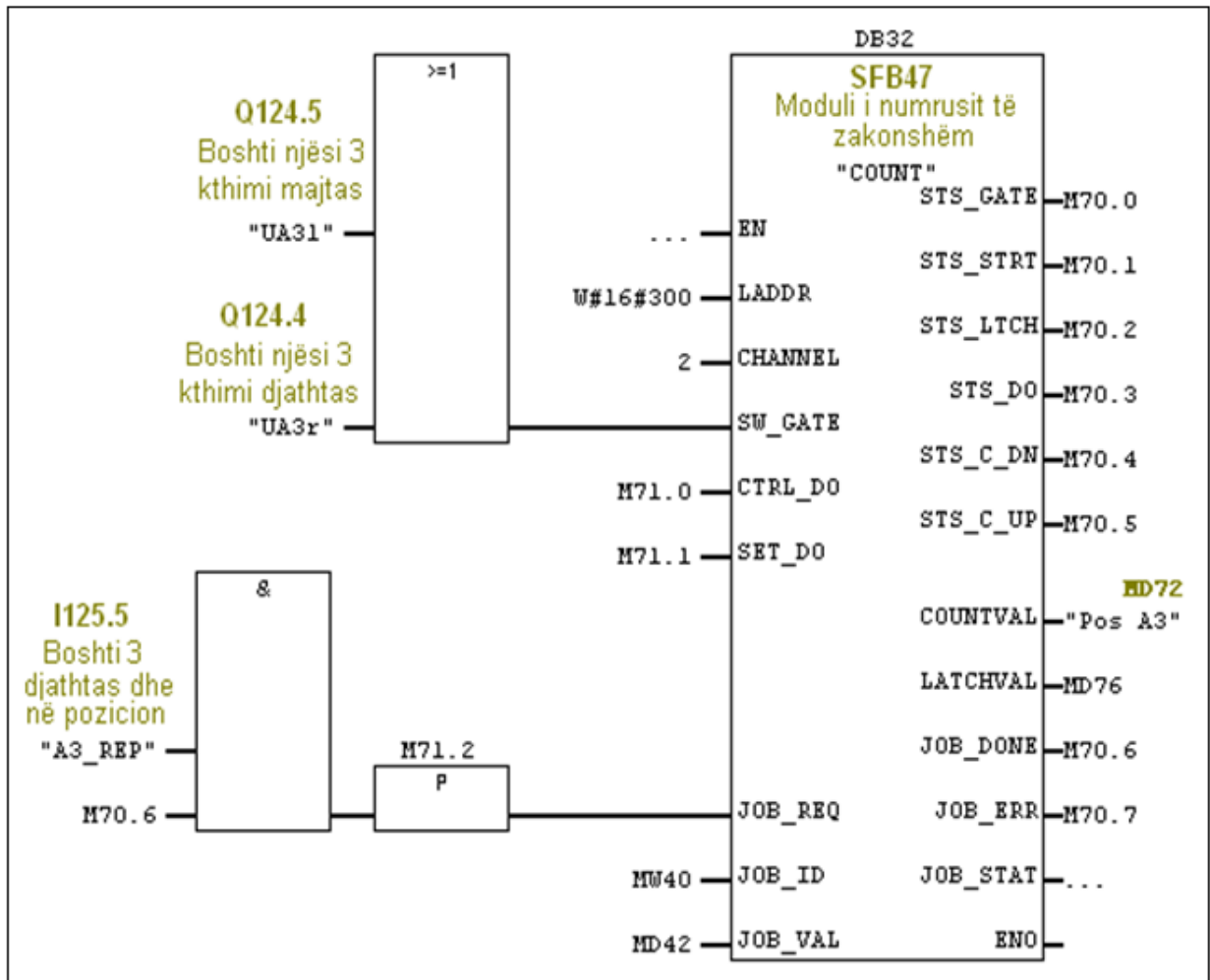
**Qarku 3:** Kontrollimi i lëvizjes së boshtit numërues 1 nga nënblloku i data bllokut DB30.



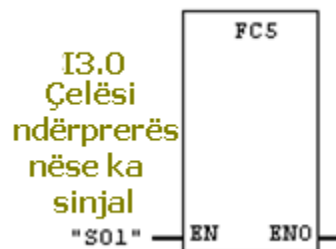
**Qarku 4:** Kontrollimi i lëvizjes së boshtit numërues 2 nga nënblloku i data bllokut DB31.



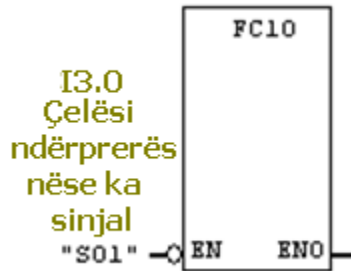
**Qarku 5:** Kontrollimi i lëvizjes së boshtit numërues 3 nga nënblloku i data bllokut DB32.



**Qarku 6:** Ndërprerësi me çelës nëse ka sinjal për STOP apo START në FC5



**Qarku 7:** Ndërprerësi me çelës nëse ka sinjal për STOP apo START në FC10



**Njësit përbërse programuese të programit kryesor të OB1**

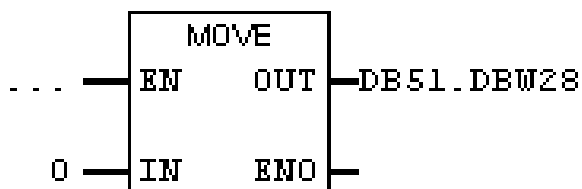
Name	Data Type	Address	Comment
<input checked="" type="checkbox"/> OB1_PRIORITY	Byte	0.0	Priority of OB Execution
<input checked="" type="checkbox"/> OB1_OB_NUMBR	Byte	1.0	1 (Organization block 1, OB1)
<input checked="" type="checkbox"/> OB1_RESERVED_1	Byte	2.0	Reserved for system
<input checked="" type="checkbox"/> OB1_RESERVED_2	Byte	3.0	Reserved for system
<input checked="" type="checkbox"/> OB1_PREV_CYCLE	Int	4.0	Cycle time of previous OB1 scan (milliseconds)
<input checked="" type="checkbox"/> OB1_MIN_CYCLE	Int	6.0	Minimum cycle time of OB1 (milliseconds)
<input checked="" type="checkbox"/> OB1_MAX_CYCLE	Int	8.0	Maximum cycle time of OB1 (milliseconds)
<input checked="" type="checkbox"/> OB1_DATE_TIME	Date_And_Time	10.0	Date and time OB1 started
<input checked="" type="checkbox"/> OB1_EV_CLASS	Byte	18.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
<input checked="" type="checkbox"/> OB1_SCAN_1	Byte	19.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
<input type="checkbox"/>			



## 7.2 OB100 Nënprogrami kryesor organizativ i ROBOTIT SKARA

### 7.2.1 OB100 “Ristartimi i plotë i OB1”

**Qarku 1:** Rivendosja e pjesës statike në lëvizje, përmes nënblokut të data bllokut DB51



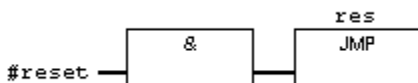
*Njësitë përbërëse programuese të nën programit kryesor në OB100*

Name	Data Type	Address	Comment
<input checked="" type="checkbox"/> OB100_EV_CLASS	Byte	0.0	16#13, Event class 1, Entering event state, Event logged in diagnostic buffer
<input checked="" type="checkbox"/> OB100_STARTUP	Byte	1.0	16#81/82/83/84 Method of startup
<input checked="" type="checkbox"/> OB100_PRIORITY	Byte	2.0	Priority of OB Execution
<input checked="" type="checkbox"/> OB100_OB_NUMBER	Byte	3.0	100 (Organization block 100, OB100)
<input checked="" type="checkbox"/> OB100_RESERVED_1	Byte	4.0	Reserved for system
<input checked="" type="checkbox"/> OB100_RESERVED_2	Byte	5.0	Reserved for system
<input checked="" type="checkbox"/> OB100_STOP	Word	6.0	Event that caused CPU to stop (16#4xxx)
<input checked="" type="checkbox"/> OB100_STRT_INFO	DWord	8.0	Information on how system started
<input checked="" type="checkbox"/> OB100_DATE_TIME	Date_and_Time	12.0	Date and time OB100 started
<input type="checkbox"/>			

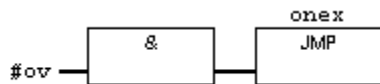
### 7.3 FB51 Blloku FunkSIONal

**FB51:** Dirigjimi i bllokut funksional, duke Startuar dhe Ristartuar lëvizjet e

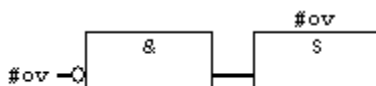
**Qarku 1:** Kalimi në rivendosje i lëvizjes



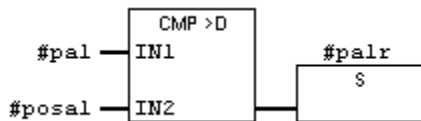
**Qarku 2:** Kalimi në vendosje i lëvizjes



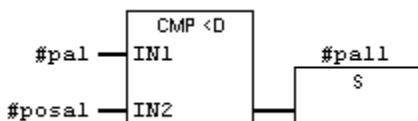
**Qarku 3:** Startimi i lëvizjeve



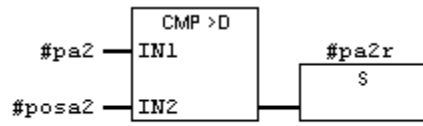
**Qarku 4:** Kontrolli i lëvizjes së boshtit 1, kthimi djathtas



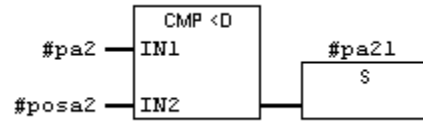
**Qarku 5:** Kontrolli i lëvizjes së boshtit 1 kthimi majtas



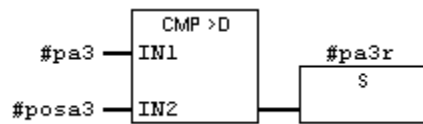
**Qarku 6:** Kontrolli i lëvizjes së boshtit 2 kthimi djathtas



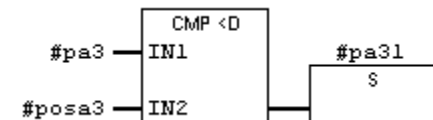
**Qarku 7:** Kontrolli i lëvizjes së boshtit 2 kthimi majtas



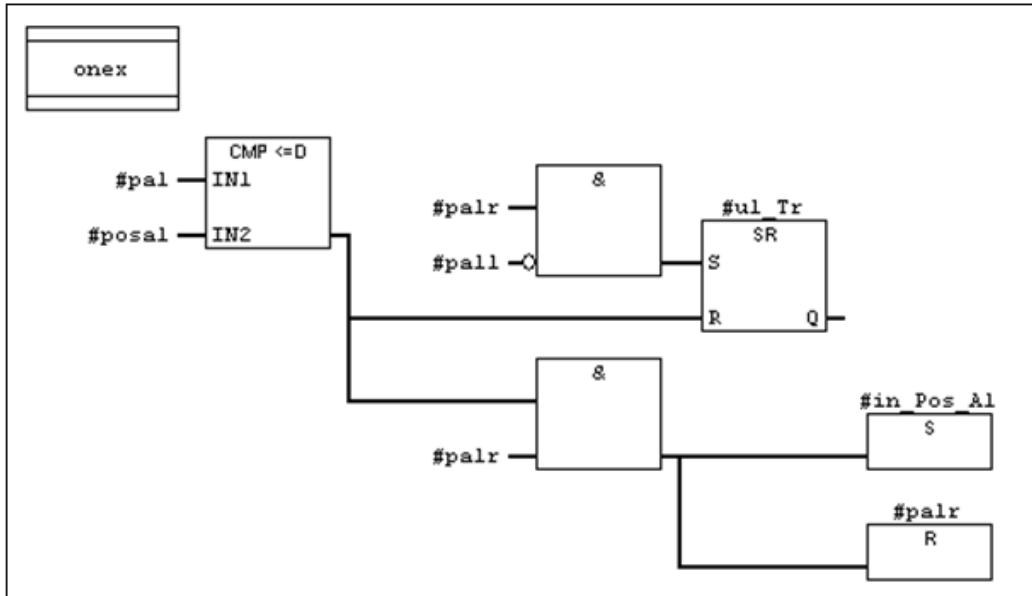
**Qarku 8:** Kontrolli i lëvizjes së boshtit 3 kthimi djathtas



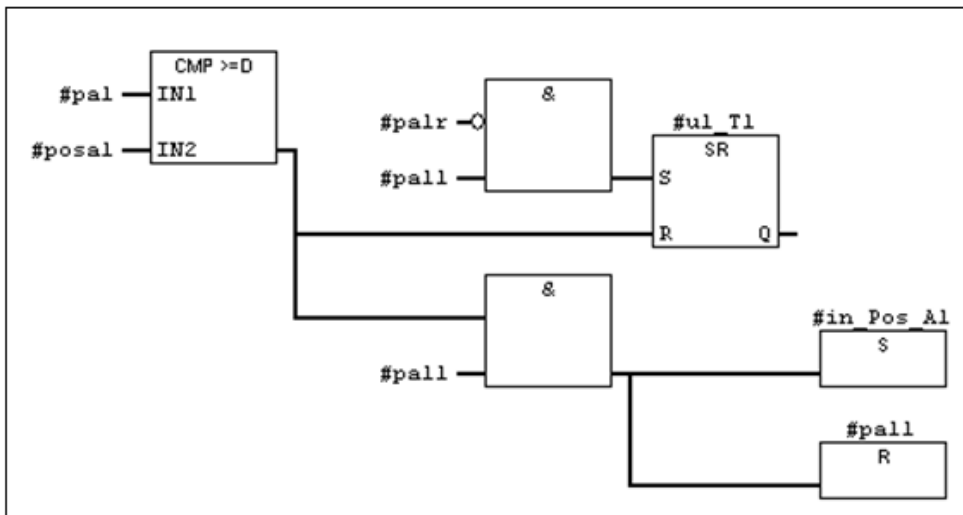
**Qarku 9:** Kontrolli i lëvizjes së boshtit 3 kthimi majtas



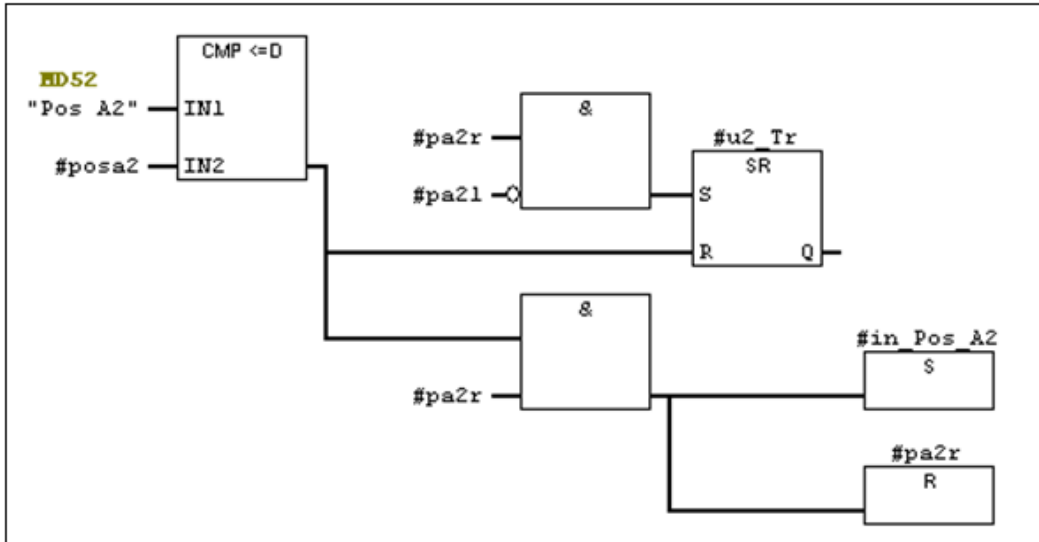
**Qarku 10:** A1 kthimi djathtas



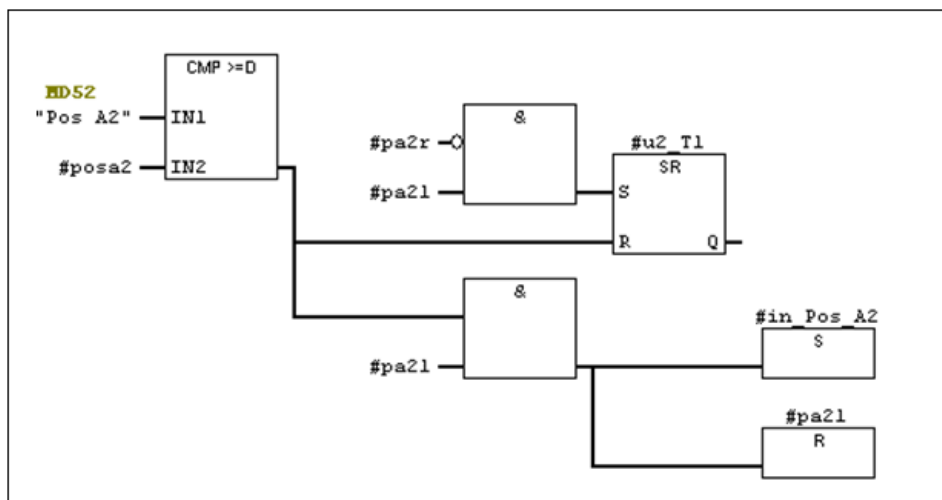
**Qarku 11:** A1 kthimi majtas



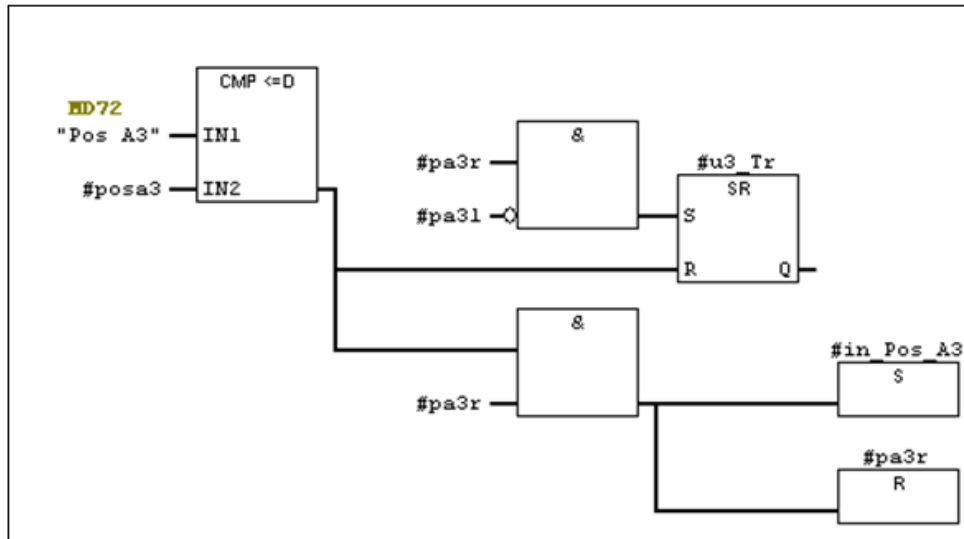
**Qarku 12:** A2 kthimi djathtas



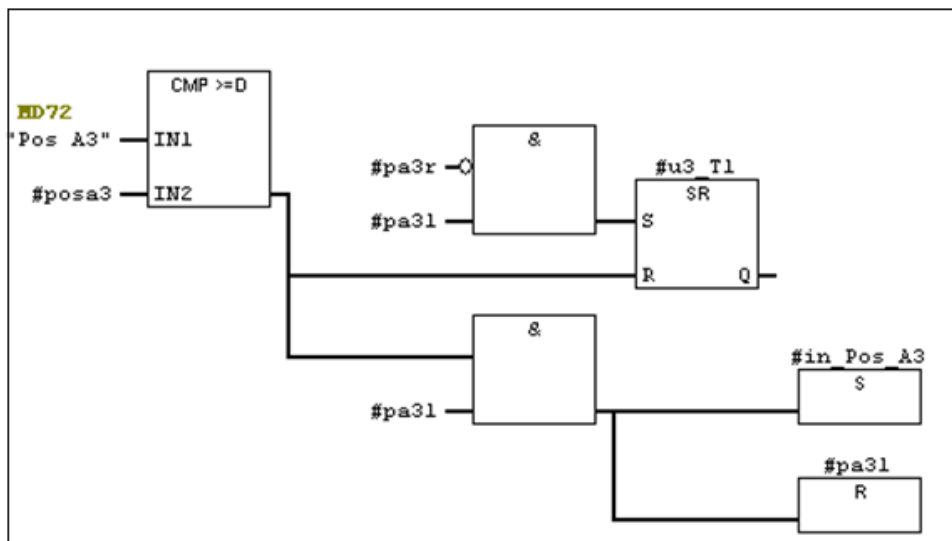
**Qarku 13:** A2 kthimi majtas



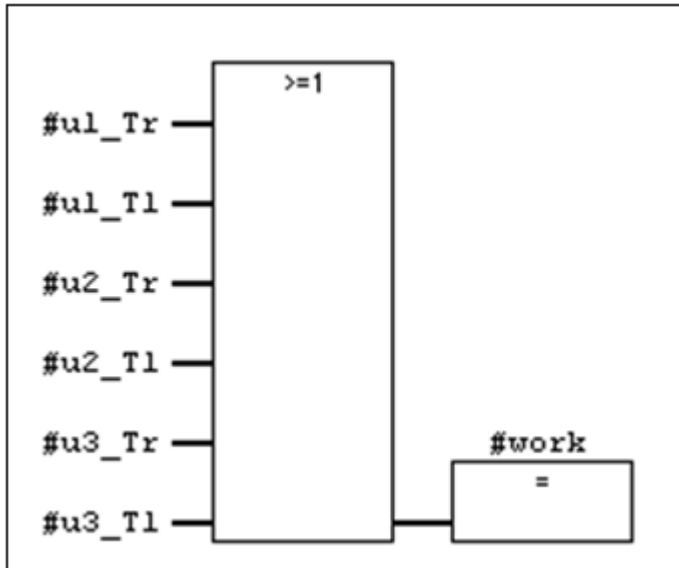
Qarku 14: A3 kthimi djathtas



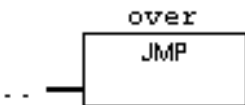
Qarku 15: A3 kthimi majtas



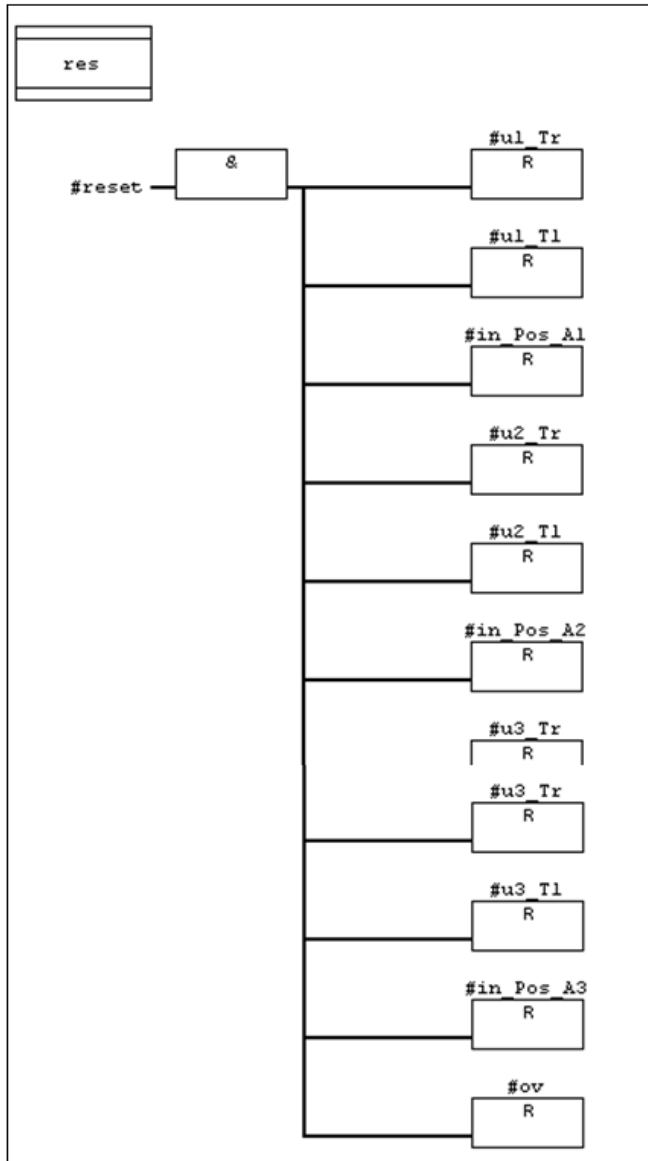
**Qarku 16:** Kontrollimi i sinjaleve hyrëse



**Qarku 17:** Kalimi i pakushtëzuar i sinjaleve



**Qarku 18:** Rivendosja e të gjitha boshteve në pozitën fillestare, si dhe kontrollimi i lëvizjeve të tyre.





***Njësitë përbërëse të programimit të lëvizjeve, njësitë hyrëse (Inputet) të Bllokut Funksional FB51***

Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
pal	DInt	0.0	L#0	<input type="checkbox"/>	<input type="checkbox"/>	
posal	DInt	4.0	L#0	<input type="checkbox"/>	<input type="checkbox"/>	
pa2	DInt	8.0	L#0	<input type="checkbox"/>	<input type="checkbox"/>	
posa2	DInt	12.0	L#0	<input type="checkbox"/>	<input type="checkbox"/>	
pa3	DInt	16.0	L#0	<input type="checkbox"/>	<input type="checkbox"/>	
posa3	DInt	20.0	L#0	<input type="checkbox"/>	<input type="checkbox"/>	
reset	Bool	24.0	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	

**Njësitë përbërëse të daljës (Outputet) të Bllokut Funksional FB51**

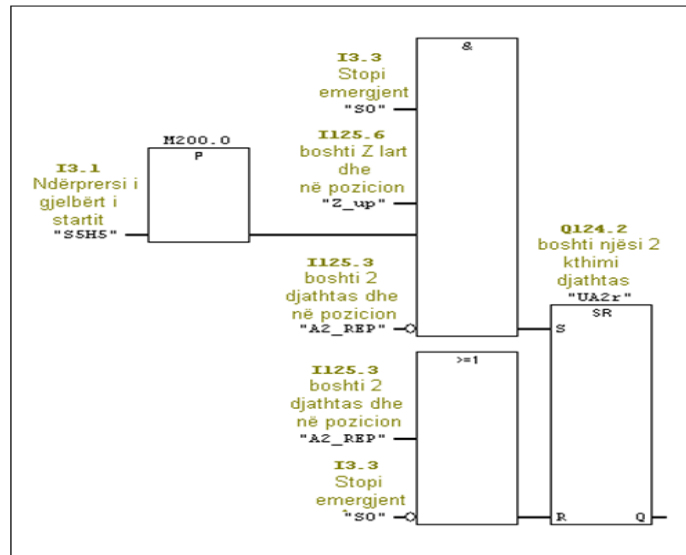
Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
ul_Tr	Bool	26.0	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
ul_Tl	Bool	26.1	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
in_Pos_A1	Bool	26.2	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
u2_Tr	Bool	26.3	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
u2_Tl	Bool	26.4	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
in_Pos_A2	Bool	26.5	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
u3_Tr	Bool	26.6	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
u3_Tl	Bool	26.7	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
in_Pos_A3	Bool	27.0	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
work	Bool	27.1	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	

**Njësitë përbërëse të Startit të Bllokut Funksional FB51**

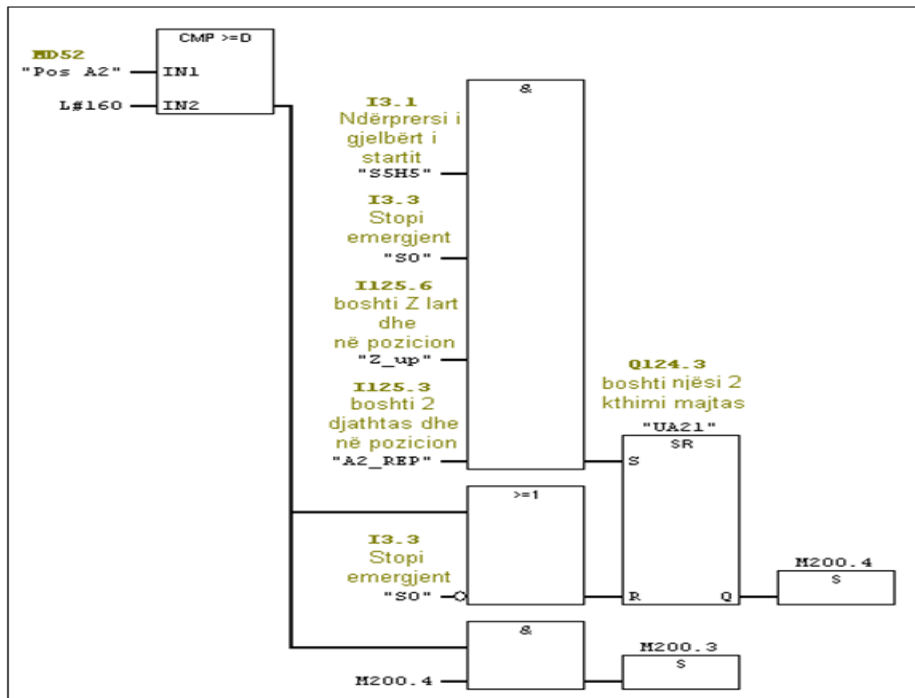
Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
palr	Bool	28.0	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
pal1	Bool	28.1	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
pa2r	Bool	28.2	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
pa2l	Bool	28.3	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
pa3r	Bool	28.4	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
pa3l	Bool	28.5	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
p1	Bool	28.6	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
p2	Bool	28.7	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
p3	Bool	29.0	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
p4	Bool	29.1	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
p5	Bool	29.2	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
p6	Bool	29.3	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	
ov	Bool	29.4	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	



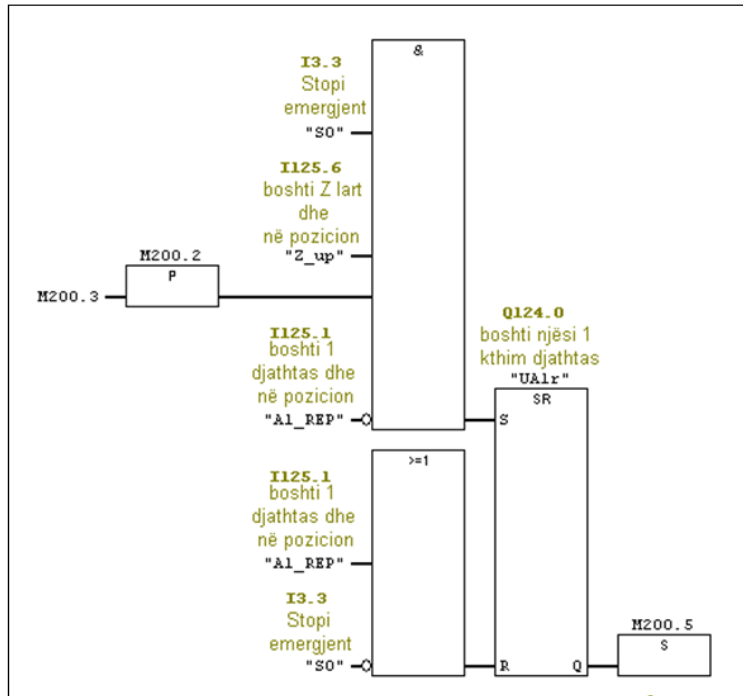
**Qarku 3:** Boshti njësi 2 kthimi djathtas, djathtas dhe në pozicion



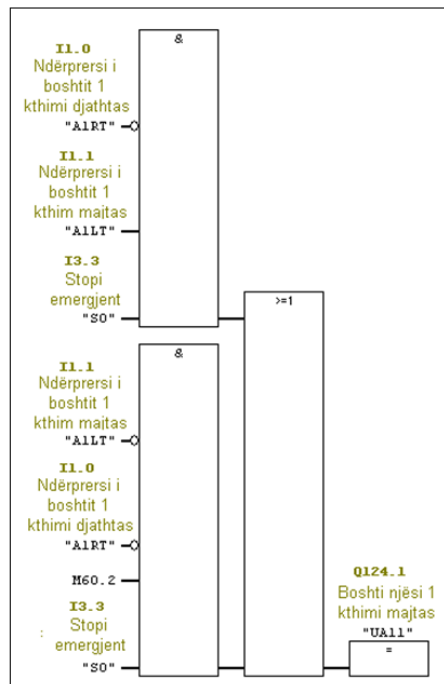
**Qarku 4:** Boshti njësi 2 kthim majtas deri në 90 pulse, pozicioni referent



**Qarku 5:** Boshti njësi 1 kthim djathtas, djathtas dhe në pozicion

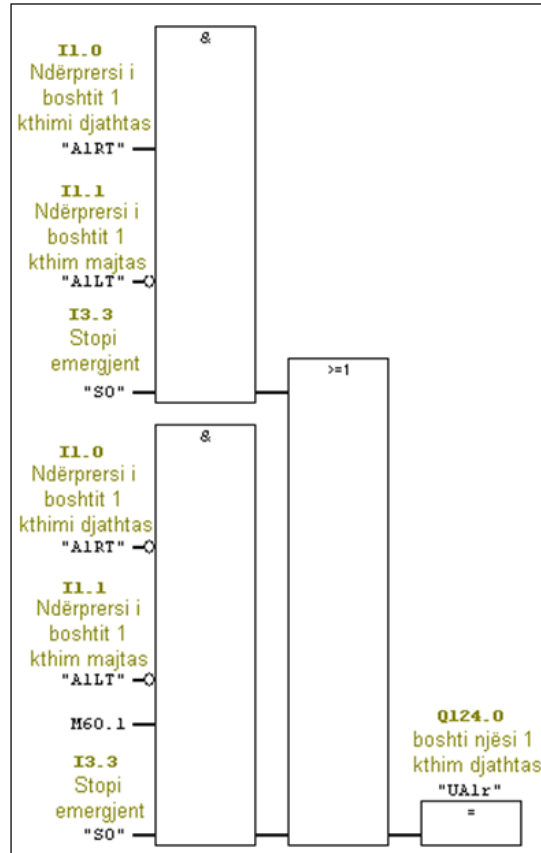


**Qarku 6:** Boshti njësi 1 kthim majtas deri në 600 pulse, pozicioni referent

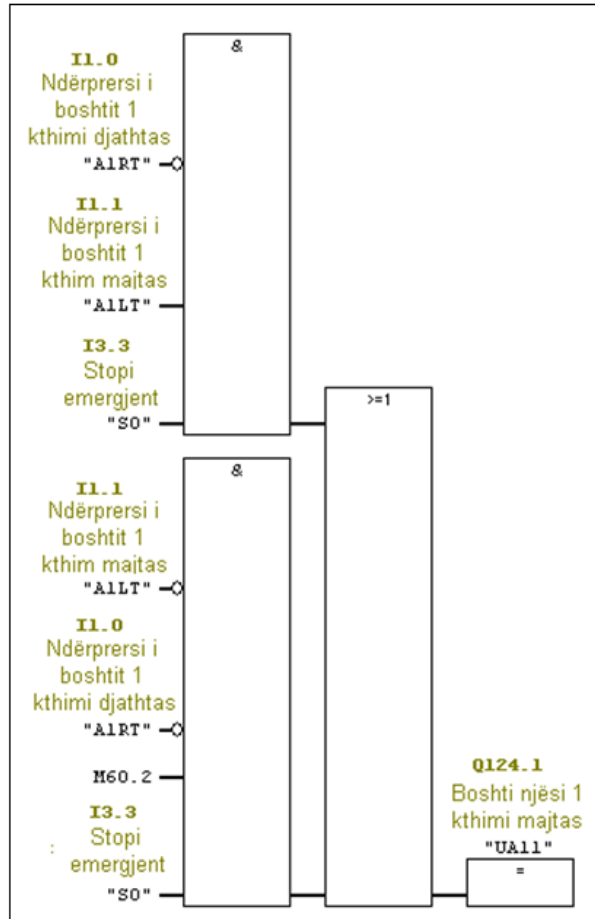


### 7.5 FC10 Nën funksioni i programimit të lëvizjeve të robotit SKARA

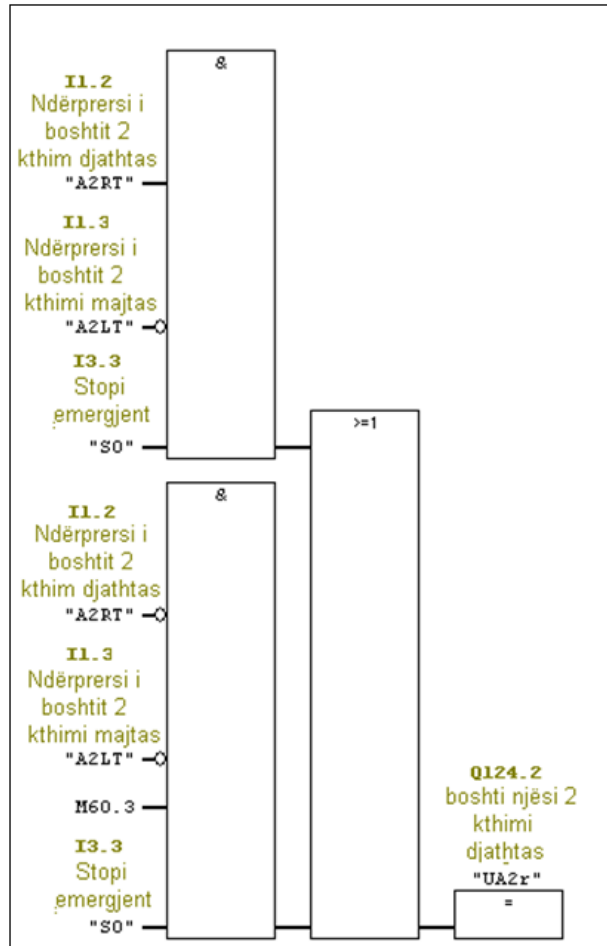
**Qarku 1:** A1 lëvizja manuale dhe automatike djathtas



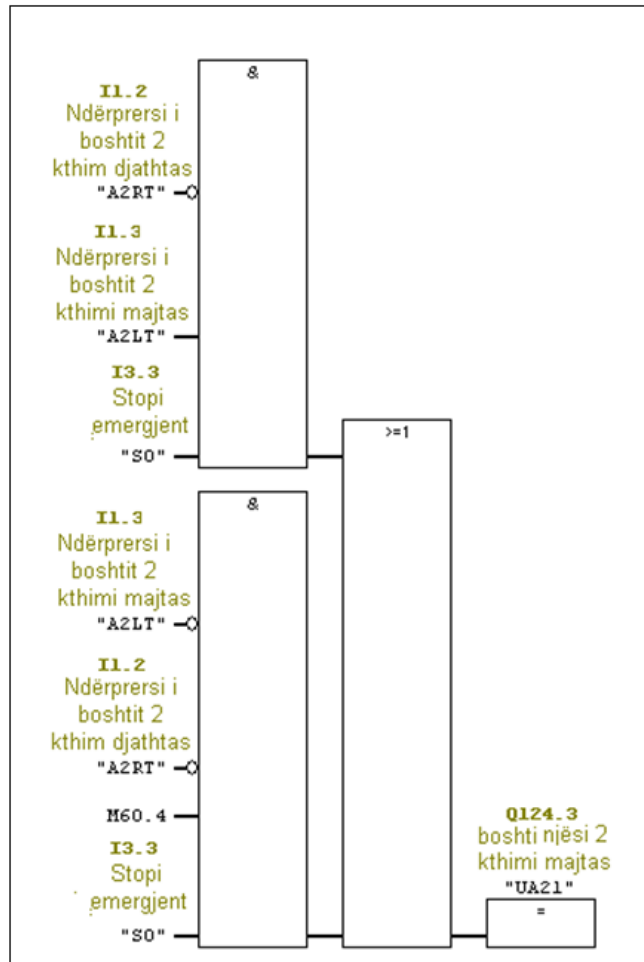
**Qarku 2:** A1 lëvizja manuale dhe automatike majtas



**Qarku 3:** A2 lëvizja manuale dhe automatike djathtas

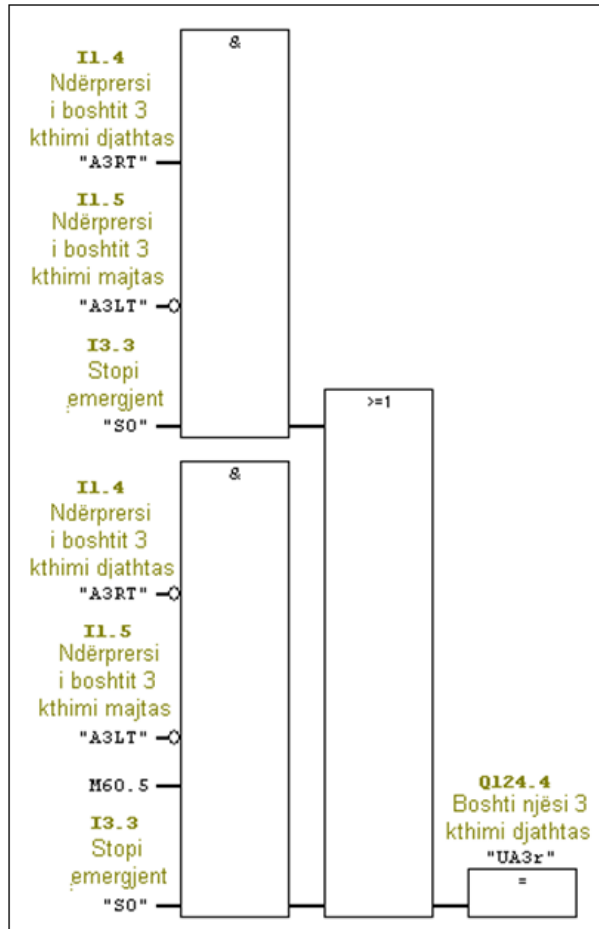


**Qarku 4:** A2 lëvizja manuale dhe automatike majtas



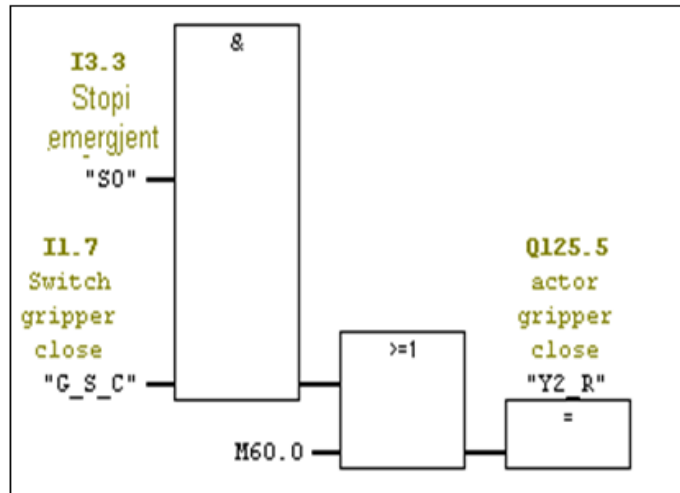


**Qarku 5:** A3 lëvizja manuale dhe automatike djathtas





**Qarku 8:** Boshti Z lartë/poshtë



## KONKLUSIONE DHE PËRFUNDIMI

Në bazë të literaturës së shqyrtuar, materialeve të shkarkuara nga interneti dhe dokumentacionit që posedon softueri i instaluar për programimin e PLC në SIMATIC S7-300 është vërejtur se ky softuer ka mundësi të shumta për programe të ndryshme dhe se kërkon një qasje përmbajtësore për të bërë programin konkret për robotin SKARA. Firma e njohur SIEMENS ka prodhuar një PLC për automatizim të linjave prodhuese i cili sistem pothuajse çdo vit pëson modifikime dhe avansime të konsiderueshme.

Nga literatura si dhe nga web faqet e vizituara shihet se programimit të PLC-ve i kushtohet rëndësi e veçantë. Ekzistojnë shumë tipe të tyre por SIMATIC S7-300 mundëson që me një qasje shumë të lehtë dhe praktike mund të shkruhet programi në sekuenca dhe të punohet me urdhra logjik adekuatë.

Një hapësirë të gjerë në programimin e PLC-ve padyshim zë SIMATIC S7-300 si kontrollues i zgjidhjeve adekuate për programimin e detyrave të ndryshme të sistemeve, i cili ofron një kontrollë të përsosur ndër softuerët logjik të programueshëm.

Nëse detyrat e kontrollit, integrimi i teknologjisë dhe arkivimi i të dhënave në një nivel adekuat bëhet përmes një kontrolluesi softuerik, atëherë softueri SIMATIC është mjaft i përsosur për realizimin e programimit dhe ruajtjen e të dhënave gjatë programimit të një sistemi automatik shumë të ndërlikuar.

Një kontrollues PLC i programuar me SIMATIC përfaqëson një investim të sigurtë për të ardhmen, dhe lejon për t'iu përgjigjur sfidave të reja në mënyrë të shpejtë, saktë, fleksibile dhe me kosto efektive.

Nevoja për efikasitet më të madh në programimin e PLC-ve me Simatic S7-300 ka shtyrë që shumë inxhinierë të krijojnë programe të veçanta softuerike për të lehtësuar punët gjatë automatizimit të centralizuar apo të decentralizuar për zbatim në degët e ndryshme të industrisë.

Në këtë punim është shkruar programi për punën e PLC-së për dirigjim të robotit Skara me ndihmën e instruksioneve bit logjike, instruksioneve kohore, instruksioneve të krahasimit dhe të numërimit, instruksioneve matematikore si dhe instruksioneve logjike dha ato të shenimeve në blloqe.

Teknikat e hulumtimit të cilat janë përdorur jenë ato të instruksioneve që u cekën më lartë si dhe janë përdorur instruksionet me PC (Program Control) të mundësisë së programimit të rregullimit për rastet speciale siç është kombinimi i punës së robotit dhe konvejerit.

Me programimin përmes softuerit SIMATIC nga kompania e njohur Siemens, arrijmë efikasitetin maksimal me softuerë të automatizimit i cili ofron punë më të mirë në këtë fushë. Softuerët e përkryer harmonizohen në industri për të optimizuar të gjitha detyrat softuerike të automatizimit në mbarë makineritë si dhe sistemi i cikleve të punës në industri. Përmes softuerit SIMATIC automatizimi arrin lartësi të reja në efikasitet që rezultoi nga uniformiteti i platformës së integruar dhe shfrytëzimi i ndërlidhjeve të hapura. Me përpilimin e këtij programi do të jemi në gjendje që të përgatisim bazën shumë të mirë për programimet e veçanta të komplikuar për rastet specifike të cilat mund të hasen në industri.

## LITERATURA E SHQYRTUAR

- [1] Kujtim Veisllari – “Automatët e Programueshëm – PLC”, Tiranë 2008
- [2] Agron Pajaziti – “Sisteme Hidraulike dhe Pneumatike”, Prishtinë 2005
- [3] [www.festo.com](http://www.festo.com)
- [4] [www.automation.siemens.com](http://www.automation.siemens.com)
- [5] [Kevin Collins](#) – “PLC Programming for industrial Automation”
- [6] Jürgen Kaftan – “PLC-Basic Course with SIMATIC S7”
- [7] Max Rabiee – “Programmable Logic Controllers texbook W/PLC Stimulation software”