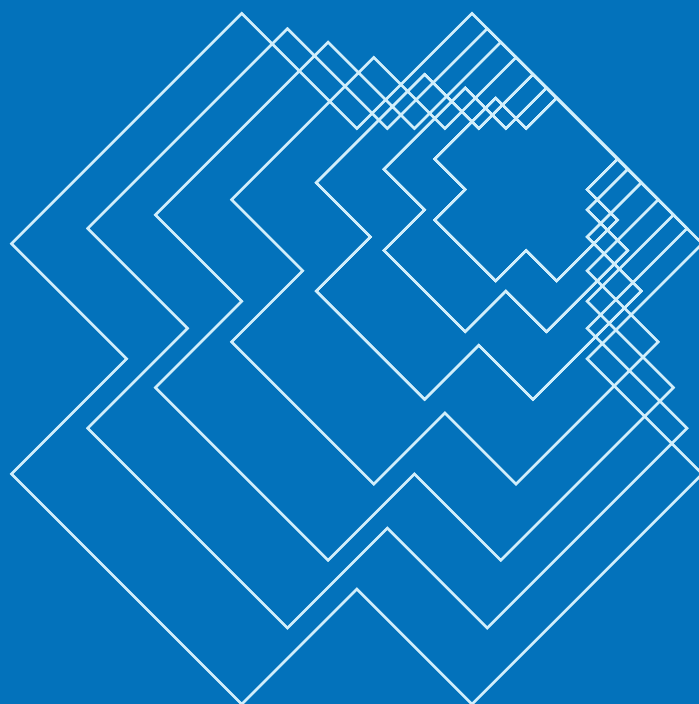
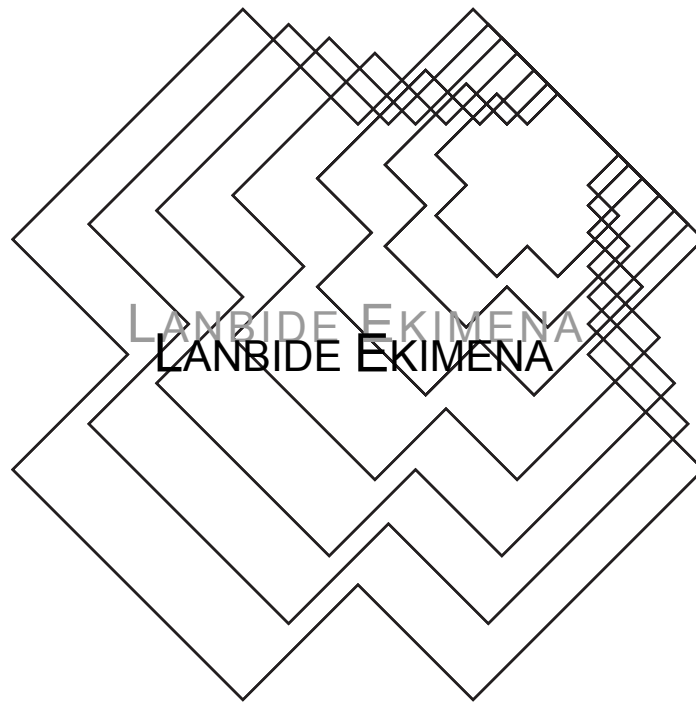




Automaten programazioa (OMRON)



LANBIDE
EKIMENA



▣ *Proiektuaren bultzatzaileak*



▣ *Laguntzaileak*



Gipuzkoako Foru Aldundia
Diputación Foral de Gipuzkoa
Gizarte eta Erakunde Harremanetarako
Departamentua

▣ *Hizkuntz koordinazioa*



hizkuntz
ELHUYAR
zerbitzuak

Eusko Jaurlaritza. 06-03-22ko onespena

Egilea(k): OSINALDE VIDAL ASIER

Zuzenketak: Elhuyar Hizkuntza Zerbitzuak

Maketa: Garbiñe Goikoetxea

Azalaren diseinua: Naiara Beasain

2006an prestatua



SARRERA.....	2
1. PLC-EN EGITURA.....	4
1.1. Egitura.....	4
1.2. Memoria.....	7
1.3. CPUa.....	9
1.4. Sarrera/Irteera unitateak.....	10
2. SARRERA/IRTEEREN KONEXIOA.....	14
2.1. Sarreren konexioak.....	14
2.2. Irteeren konexioak.....	15
3. PLC-EN PROGRAMAZIOA.....	19
3.1. Memoriaren guneak.....	19
3.2. Programaren arkitektura.....	22
3.3. Aginduak.....	22
3.4. Programazio-lengoiak.....	23
3.5. Oinarrizko aginduak.....	26
3.6. Programazio-teknikak.....	42
3.7. Egin beharreko araketak.....	43
3.8. Programazio-kontsolaren erabilera.....	43
3.9. Egin beharreko araketak.....	46

3.10.	Agindu-zerrenda.....	46
3.11.	Egin beharreko ariketak.....	64
4.	GRAFCEA	65
4.1.	<i>Grafce</i> terako sarrera.....	65
4.2.	<i>Grafce</i> a eraikitzeke kontzeptu nagusiak.....	65
4.3.	<i>Grafce</i> taren eboluzio-erregelak.....	68
4.4.	II. mailako <i>grafce</i> a.....	73
4.5.	Ekuazio logikoak.....	74
4.6.	Kontaktuen eskemak osatzea ekuazio logikoak oinarritzat hartuz.....	78
4.7.	<i>Grafce</i> t konplexuagoen ekuazio logikoak.....	81
4.8.	Egin beharreko ariketak.....	86
5.	ARIKETAK	87
5.1.	Oinarrizko ariketak.....	87
5.2.	Zenbait irteerako eskemei buruzko ariketak.....	90
5.3.	IL, ILC eta TRi buruzko ariketak.....	91
5.4.	Tenporizadoreen oinarrizko ariketak.....	93
5.5.	Kontadoreen oinarrizko ariketak.....	95
5.6.	Tenporizadoreen eta kontadoreen ariketak.....	96
5.7.	Sistema sekuentzialaren ariketak (<i>Grafce</i> -a erabiliz).....	101

AUTOMATEN PROGRAMAZIOA (OMRON)

SARRERA

Historian zehar gizakiak makina bat lan astun, errepikakor eta arriskutsu egin behar izan du. Lan horiek erraztu eta arintzeko sortu zen automatizazioa.

Pneumatika, hidraulika eta elektrizitatea izan ziren hasiera batean automatizazio-prozesuan gehien erabili ziren teknikak, bai aginte-zirkuituan, bai indar-zirkuituan. Ondoren, eta elektronika eta mikroprozesagailuen garapen teknologikoa dela medio, automata programagarriak sortu ziren, baina aginte-zirkuituan bakarrik erabiltzen ziren.

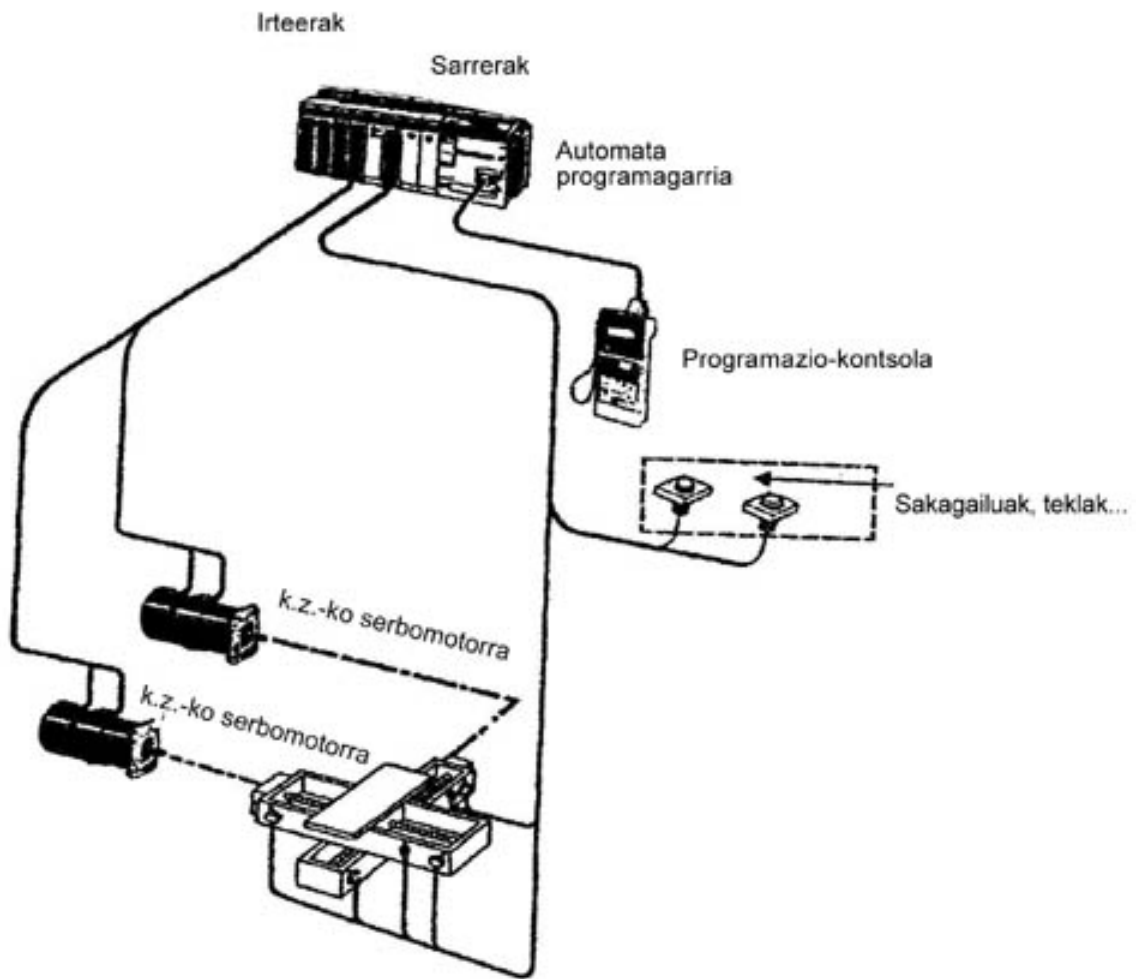
Automatak 1960. urtean hasi ziren industrian erabiltzen. Izan ere, Bedford Associates erakundeak *Modular Digital Controller* izeneko gailua eskaini zion auto-ekoizle handi bati. Gailu horren izena oinarritzat hartuz sortu zen historiako lehen automata, MODICON 084a, hain zuzen ere.

Automata programagarriak erabiltzen hasteko arrazoi nagusia zirkuituak aldatzeko garaian dituen abantailak izan ziren. Elektrizitatean, koadro elektrikoaren funtzioa aldatzeko, kableatu guztia (edo zati handi bat) aldatu beharra dago, eta, jakina, lan zail eta luzea da. Pneumatikan eta hidraulikan ere beste horrenbeste gertatzen da. Automata programagarriaren bitartez, ordea, nahikoa da programa aldatzea. Gainera, automatak beste teknologia kableatuek baino espazio gutxiago hartzen du.

Gaur egun, automata programagarria mikroprozesagailu bat oinarritzat duen sistema gisa defini daiteke. Hona hemen ezaugarri nagusiak:

- ✓ Gehienetan modulutan konfiguraturia daude
- ✓ Lengoia ez-informatikoan programa daitezke
- ✓ Denbora errealean eta industria-giroan prozesu sekuentzialak kontrolatzeko gai dira

Automata programagarria, laburtuta, PLC (*programmable logic controller*) siglen bitartez adierazi ohi da.



1 PLC-EN EGITURA

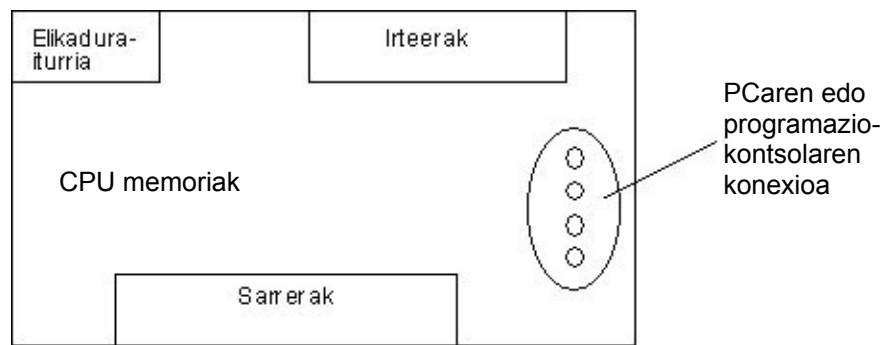
1.1 Egitura

Kanpo-egitura

Gaur egun erabiltzen diren egitura ohikoenak egitura trinkoa eta modulu-egitura dira.

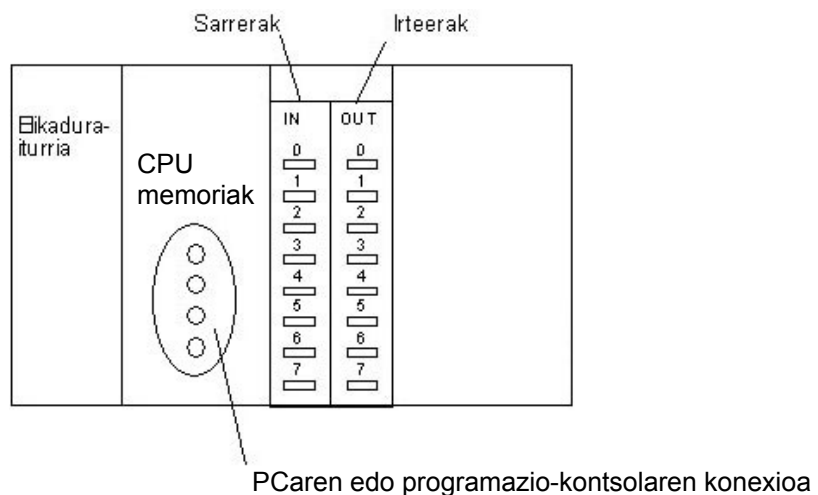
▶ Egitura trinkoa

Bloke bakar batean daude piloturik elementu guztiak, hots, elikadura-iturria (EI), mikroprozesagailua (CPU), memoriak, sarrerak/irteerak (input/output), etab.



▶ Modulu-egitura

Egitura osoa bloketan zatitua dago. Funtzio bakoitzari modulu bat egokitzen zaio: EI, CPU, I/O, etab.



■ Barne-egitura

Automata programagarriak sei elementu nagusiz osatuak daude:

- ✓ Sarrera-modulua (inputa)
- ✓ Prozesu-unitate zentrala (CPU)
- ✓ Irteera-modulua (outputa)
- ✓ Elikadura-iturria
- ✓ Programazio-kontsola
- ✓ Gailu periferikoak
- ✓ Interfazeak

▶ Sarrera-modulua

Kaptadoreen seinaleak (sakagailuenak, ibiltarte-mikroetengailuenak, sentsoreenak...) egokitu eta kodifikatu egiten ditu, CPUak uler ditzan. Gainera, PLCaren barne-zirkuitu elektronikoak babestu egiten ditu; izan ere, zirkuitu horiek eta kaptadoreak banatzen ditu.

▶ CPU (prozesu-unitate zentrala)

Hau da sistemaren adimena. Erabiltzaileak sortutako programari dagozkion aginduak interpretatu eta, sarrerako balioen arabera, dagozkion irteerak aktibatzen ditu.

▶ Irteera-modulua

Irteera-moduluak sarrerakoaren alderantzizko eginkizuna betetzen du. CPUtik datozen seinaleak deskodifikatu, amplifikatu eta eragingailuei bidaltzen dizkie: lanparei, erreleei, kontaktoreei, elektrobalbulei... Barne-zirkuitua ere babestua dago.

► **Elikadura-iturria**

Kanpoko sarearen tentsioa (normalean 230 V-eko korrante alternoa) PLCaren barneko elikadura-tentsiora (5 V) eta sarrerako gailuen tentsiora (24 V) egokitzen du.

► **Programazio unitatea**

Erabiltzailea CPUaren barnealdearekin komunikatu ahal izateko erabiltzen da. Adibidez, programa automataren memorian sartu eta gorde ahal izateko.

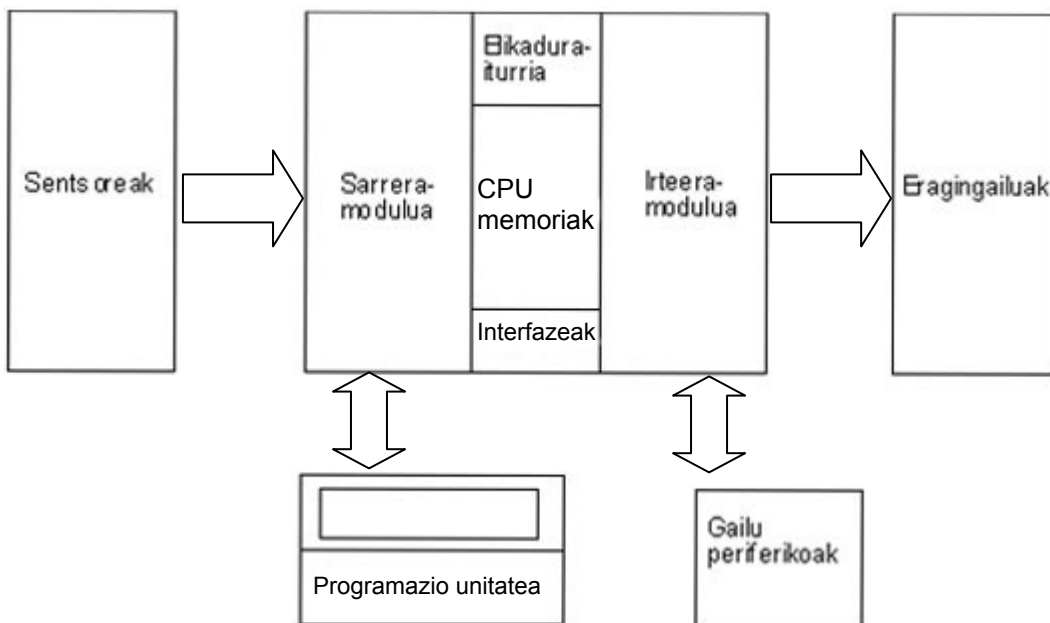
► **Gailu periferikoak**

Automatari lotzen zaizkion gailuak dira. Funtzio espezifiko bat egiteko edo automataren aplikazio-eremua handitzeko erabili ohi dira.

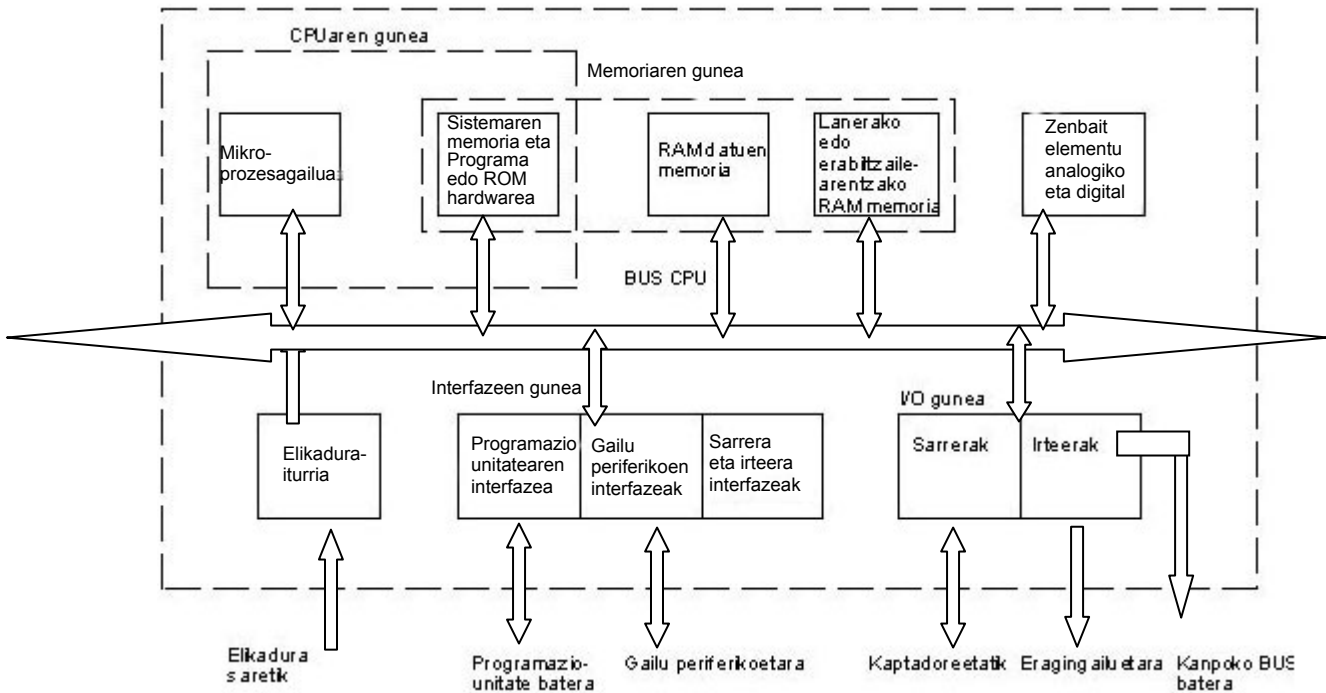
Gehienetan, ez dute ez programaren sorreran ezta egikaritzean parte hartzen.

► **Interfazeak**

CPUaren eta gailu periferikoen arteko lotura egin ahal izateko erabiltzen diren zirkuitu edo gailuak dira.



PLC baten arkitektura:



1.2 Memoria

Informazioa gordetzeko erabiltzen den gailua da.

Memoria motak

▶ RAM (random access memory)

Ausazko atzipeneko memoria edo irakurtzen eta idazten uzten duen memoria. Irakurketa eta idazketa-prozesuak modu elektrikoan egiten dira. Korrontea moztean, informazioa galdu egiten da.

▶ ROM (read only memory)

Irakurtzeko soilik balio duen memoria da. Memoriako edukia irakur daiteke, baina ezin da memorian bertan idatzi. Datuak edo aginduak automataren fabrikatzaileak grabatzen ditu, eta ezin dugu horien edukia aldatu.

Hona hemen beste zenbait memoria mota:

	PROGRAMAZIOA	EZABATZEA	KORRONTEA MOZTUZ GERO
PROM programable ROM	Elektrikoa	Ezin da	Datuek gordeta jarraitzen dute
EPROM electrically programable ROM	Elektrikoa	Izpi ultramoreen bidezkoa (UV)	Datuek gordeta jarraitzen dute
EEPROM electrically erasable programable ROM	Elektrikoa	Elektrikoa	Datuek gordeta jarraitzen dute

■ Memorien erabilera PLC batean

▶ Erabiltzailearen memoria

Erabiltzaileak egindako programa RAM memorian grabatu ohi da. Izan ere, mikroprozesagailuak programa irakurtzeaz gain, erabiltzaileak programa aldatzeko ahalmena izan behar du. Korrontea eteten denean, RAMean gordetako datuak galdu egiten dira. Horrelakorik gerta ez dadin, tarteko bateria bat jartzen da, elikadura konstantea izan dezan eta datuak gal ez daitezen.

▶ Datu-taula

Hor egon ohi dira sarrera-irteeren irudia, markak, kontagailu-balioak eta antzeko barne-aldagaiak, etab.

▶ Sistemaren memoria

Alde batetik, ROM edo EPROM memoria dago, martxan jartzeko CPUaren agindua gordetzen duena. Bestetik, RAM memoria dago, sistemaren konfigurazioa gordetzen duena.

▶ EPROM memoria eta EEPROM

RAMa ordezka dezakete programa behin betikoa denean.

1.3 Cpu

CPUa elementu hauek osatzen dute: prozesagailuak, memoriak eta zirkuitu osagarri elkartuek.

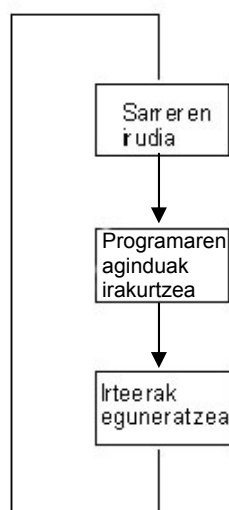
ROM memoriak zer egin behar duen esan behar dio prozesagailuari, bai sistema martxan jartzean, bai prozesuan zehar edota itzaltzean (sistemaren firmware edo softwarea).

Oro har, ekintza hauek betetzen ditu:

- ✓ Ziklo-denbora kontrolatu eta gainbegiratu (*watchdog*)
- ✓ Konexioan eta programaren egikaritzean autotesta egin
- ✓ Programaren eta multzoaren konfigurazioaren azterketa-zikloa hasi
- ✓ Zikloaren denbora-oinarria sortu
- ✓ Kanpoko elementuekin komunikatu
- ✓ Etab.

Hau da CPUaren oinarrizko lan-zikloa:

1. Sarreren irudia osatu. Irudi hori programaren egikaritzean zehar aldatzen ez diren barne-erregistro batzuetan gordetzen da.
2. Programaren aginduetara sartu eta agindu horiek exekutatu. Horrela, programaren erregistroak eguneratu egiten dira.
3. Erregistroen arabera irteerak eguneratu.



1.4 Sarrera/irteera (i/o) unitateak

- ▶ **Sarrera unitateak:** kaptadoreetako informazioa hartu eta CPUari ematen dioten gailuak dira.
- ▶ **Irteera unitateak:** CPUtik informazioa hartu eta eragingailuetara eramaten duten gailuak dira.

Hauek dira haien funtzio nagusiak:

1. Kaptadoreen eta eragingailuen laneko tentsioa nahiz intentsitatea automataren zirkuitu elektronikoetara egokitzen dituzte.
2. Zirkuitu logikoen eta potentzia-zirkuituen artean bereizketa elektrikoa egiten dute.
3. Helbide bat ematen diote kaptadore eta eragingailu bakoitzari, prozesagailuak uler dezan.

Sarrerak

Sarrerak (*input, eingabe, entrada*) honela sailka daitezke:

- ✓ Tentsioaren arabera
- ✓ Seinale motaren arabera
- ✓ Sarrera bereziak
- ▶ **Tentsioaren arabera**
 - ✓ Tentsio gabeak
 - ✓ Korrante zuzenekoak
 - ✓ Korrante alternokoak

▶ **Seinale motaren arabera**

a) *Sarrera digitalak*

Hauek dira erabilienak, eta *dena ala ezer ez* motakoak dira; alegia, tentsio-maila jakin bat dute edota ez dute tentsiorik.

Adibidez: sakagailuak, etengailuak, ibiltarte-mikroetengailuak, etab.

b) *Sarrera analogikoak*

Badira *dena ala ezer ez* balioa ez duten magnitude batzuk. Magnitude horiek lerrun bateko balioak bata bestearen atzetik hartzen dituzte jarraian. Kasu horietan, beharrezkoa da sarrera analogikoko modulu bat edukitzea. Sarrera analogikoen, beraz, tentsio- edo korrante-balio aldakorrek ematen dituzte.

Hauek dira ohikoenak:

- ✓ Tentsioa: 0 voltetik 10 voltera edo 10 voltetik 10 voltera
- ✓ Intentsitatea: 4 miliamperetik 20 miliamperera

Adibidez: likido-maila, tenperatura, motor-abiadura, etab. neurtzeko erabiltzen diren gailu elektrikoak.

CPUak balio analogikoa uler dezan, beharrezkoa da analogiko/digital bihurtgailua. Bihurtgailu horren zehaztasun-mailaren arabera izango da (bit kopuruaren arabera) sarrera analogikodun unitatearen zehaztasuna.

▶ **Sarrera bereziak**

PLC askok, orain arte ikusitako sarrerez gain, beste mota bateko sarrerak ere izan ditzakete, sarrera bereziak, hain zuzen ere.

a) *Kontaketa azkarreko sarrerak*

Barne-kontadore batekin zuzenean kableatuta dauden sarrerak dira, eta funtzionamendu independentea dute automataren ohiko zikloarekiko.

b) *Alarma-sarrerak*

Sarrera horiek PLCaren ohiko funtzionamendua eteten dute, beste eginbehar berezi batzuk egin ditzan.

 **Irteerak**

Sarreretan gertatzen den moduan, hemen ere sailkapen hau egiten da irteera motaren arabera:

- ✓ Analogikoak
- ✓ Digitalak

Irteera analogikoek tentsio- edo intentsitate-balio aldakorra dute. Irteera digitalek, berriz, funtziona dezaketen edo funtziona ez dezaketen elementuei (*dena* edo *ezer* ez motakoei) eragiteko balio dute.

Azken txartel horiek honela osatuta egon daitezke:

▶ **Errelerako irteera**

Korronte zuzenaren nahiz altxanoaren bidez funtziona dezaketen eragingailuekin erabiltzen dira. Jardute-abiadurak oso handiak ez direnean eta kontsumoak ertainak (2 ampere) direnean erabiltzen dira.

▶ **Transistorerako irteera**

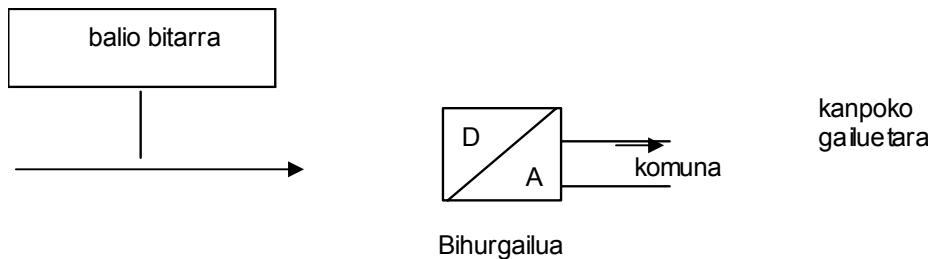
Hauek dira gaur egun gehien erabiltzen diren irteera-txartelak.

Korronte zuzen edo altxanoaren bidez funtziona dezakete. Hauek dira ezaugarri nagusiak: kontsumo baxuko (0,5-0,9 ampere) kargekin erabiltzea, erantzun azkarra edukitzea eta eragiketa kopuru handia egiteko ahalmena izatea. Hauen bizi-iraupena erreleena baino luzeagoa da.

▶ **TRIACerako irteera**

Irteera TRIAC motakoa bada, korronte altxano elementuak erabili behar dira. Kommutazio azkarretan erabiltzen dira, errelea geldoegia den kasuetan. 0,4 ampereko intentsitatea jasan dezakete. Hauen biziraupena erreleena baino luzeagoa da.

Sarrera analogikoekin gertatzen denaren alderantzizkoa gertatzen da irteera analogikoekin: digital/analogiko-bihurgailua behar da.



Kanpo-elementu hauek konekta daitezke:

- ✓ Motorren abiaduraren erregulatzailak
- ✓ Elektrobalbula proportzionalak

Sarrera analogikoetan gertatzen den bezala, zenbat eta zehaztasun handiagoa eduki D/A bihurgailuak, orduan eta balio analogiko zehatzagoak lortuko dira.

Irteera analogikoak honelakoak izan ohi dira:

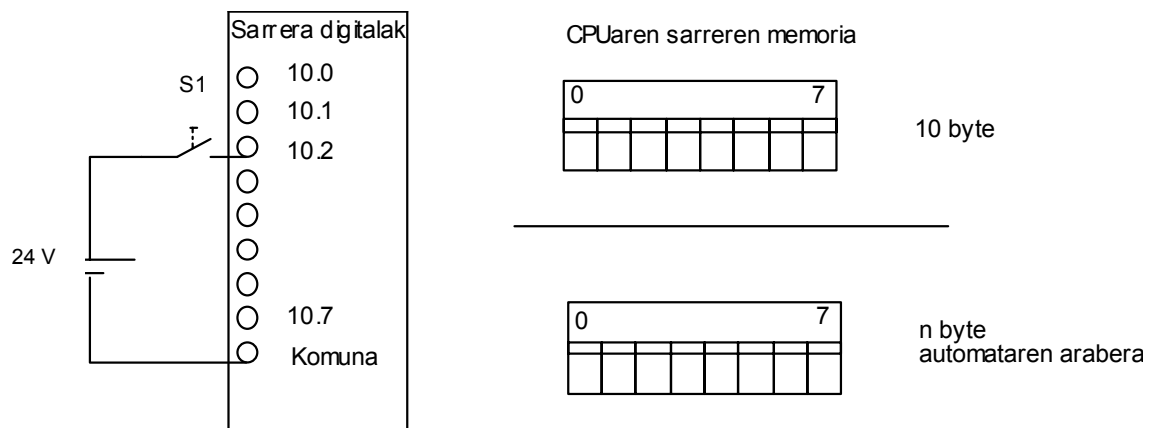
- ✓ Tentsioa: 0 voltetik 10 voltera edo 10 voltetik 10 voltera
- ✓ Intentsitatea: 4 miliamperetik 20 miliamperera

2 SARRERA/IRTEEREN KONEXIOA

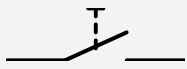
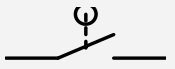
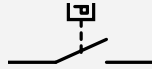

2.1 Sarreraren konexioa

Kaptadoreak:

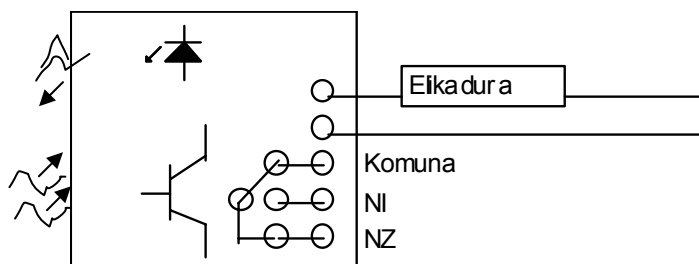
Sarrera digitalei *dena ala ezer* ez sarrera seinalea dagokie; hau da, edo tentsio-maila jakin bat dago edo ez dago tentsiorik. Kaptadoreak honela konektatzen dira:



Sakagailuez gain, badaude tentsio gabe funtzionatzen duten beste hainbat elementu:

KAPTADOREA	HAUTAGAILUA	IBILTARTE-MIKROETENGAILUAK	PRESOSTATOA	MAGNETOTERMIKOAK
IKURRA				

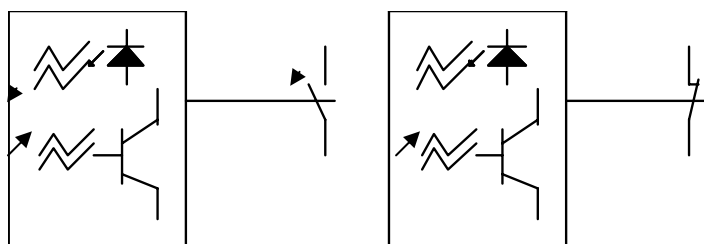
Badira elementu gehiago, errele-irteerako detektagailu fotoelektrikoa kasurako, baina elikatuak izan behar dute (DC edo AC).



Elementu hori konektatu ahal izateko, 24 V-eko komunarekin lotzen dugu, eta, ondoren, irteera-kontaktu baten sarrerarekin. Kontaktu hori normalean zabalik (NZ) edo normalean itxita (NI) motakoa izan ohi da:

- ✓ NZ: zerbait detektatzean emango du seinalea.
- ✓ NI: ezer detektatzen ez duen bitartean emango du seinalea.

Eskemetan honela marrazten dira:



Kaptadore gehiago ere badaude:

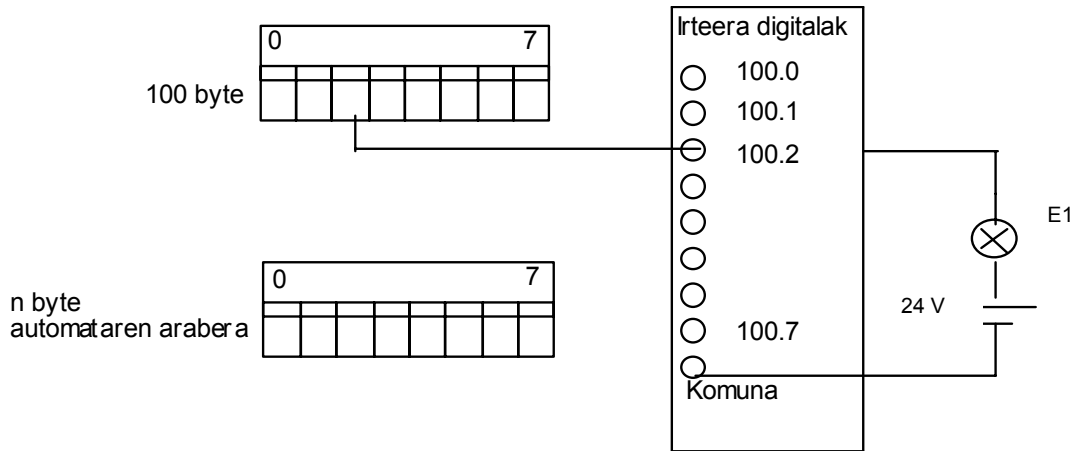
- ✓ Bi hariko detektagailu inдукtiboak eta kapazitiboak.
- ✓ Hiru hariko detektagailu inдукtiboak, kapazitiboak eta transistore-irteeradun fotoelektrikoak.

2.2 Irteeren konexioa

■ Irteera digitalak

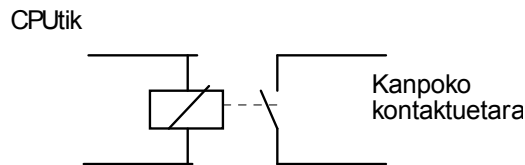
Automataren irteerako kontaktuetan eragingailuak konektatzen dira. Bi modutan egin ohi da hori: zuzenean edo zeharka (aginte-elementuak zubi gisa erabilia). Zeharkako konexioa izanez gero, kontaktoreen bobinak, erreleen bobinak, lanparak, etab erabiltzen dira.

Oro har, irteera bat honela adierazten da:



► **Errele-irteera**

Errele-irteerak honela adierazten dira laburtuta:

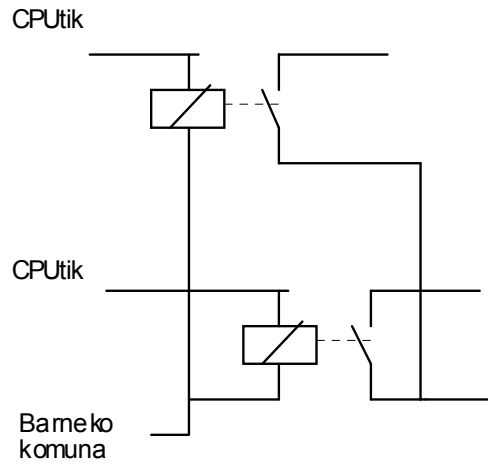


Errelearen hanketako bati komuna deritzo, eta besteari, berriz, irteera (*output, ausgabe, sortie, salida*). Irteera mota honekin korrante zuzeneko eta korrante alternoko eragingailuak erabiltzen dira.

Errele motako irteerak abantaila nagusi bat du: elikadura-tentsio ezberdina duten elementuei automata bera erabiliz eragin dakieke. Desabantaila nagusia, berriz, txartelen tamaina handia da.

Eragingailuak aukeratzean, kontuan hartu behar da eragingailuen intentsitate izendatua. Izan ere, barne-erreleak jasaten duena baino txikiagoa izan behar du.

Oro har, errele-irteera bakoitza indibiduala da, baina, batzuetan, irteera-komunak multzotan agertzen dira.



► **Transistorerako irteera**

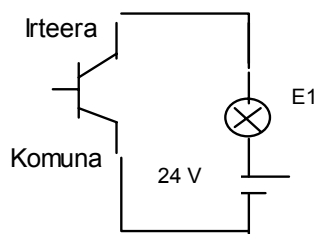
Irteera mota honetan, oktoakoplagailuak erabiltzen dira barne- eta kanpo-zirkuituak elkarren artean elektrikoki isolatzeko.

Transistorerako irteera egiteko, bi transistore mota erabili ohi dira:

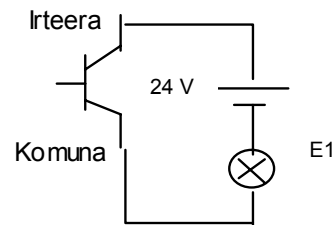
- ✓ NPN transistorea: txartel mota honetan, base komuna kanpoko elikadura-iturriko negatiboari lotua dago.
- ✓ PNP transistorea: txartel mota honetan, base komuna kanpoko elikadura-iturriko positiboari lotua dago.

Konexioen adibideak:

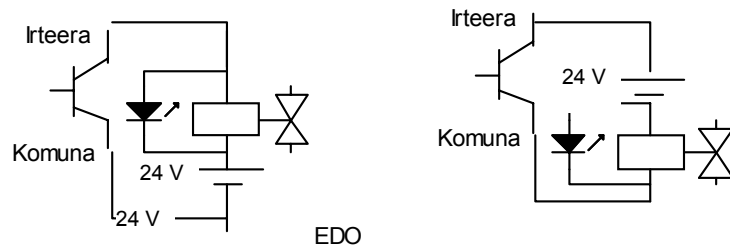
a) *Lanparak*



EDO

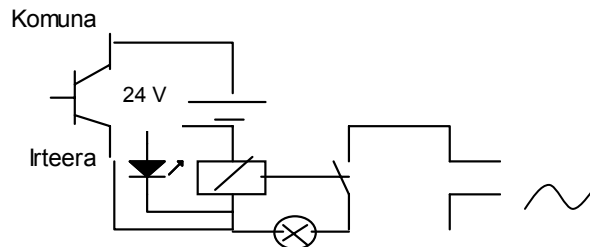


b) Bobinak



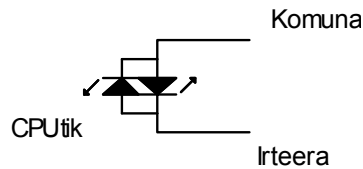
c) Transistoreak:

Transistoreek intentsitate baxua jasaten dutenez, errele osagarriak erabiltzen dira eragingailuei eragiteko.

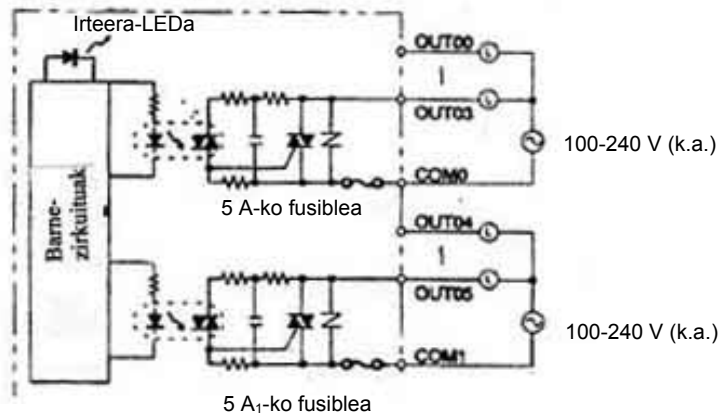


■ Irteera analogikoak

Irteera mota honetan, bi kasu bereizten dira: errele-irteerak eta TRIAC-irteerak. Lehenengoak irteera digitalen atalean aztertu ditugu; eta bigarrenak honela adierazten dira laburtuta:



Irteera ugariz osatutako taldeetan antolatu ohi dira:



3 PLC-EN PROGRAMAZIOA

3.1 Memoriaren guneak

PLCaren memoria hainbat gunetan dago antolatua. Gune bakoitzak bere eginkizunak eta ezau-garriak ditu.

Programaren gunea

PLCaren programa gordetzen den gunea (*Ladder*, lengoaia mnemonikoan).

Datuen gunea

Balio jakin batzuk gordetzeko edo PLCaren egoerari buruzko informazioa jasotzeko erabiltzen da gune hau. Funtzioen arabera, gune hauetan banatzen da: IR, SR, AR, HR, LR, DM, TR eta T/C.

▶ I/O gunea eta barne-gunea (IR00-IR15)

Memoriaren gune honetan gordetzen da I/O erreleen moduan kudeatutako kanalei dagokien informazioa: alde batetik, kanpo-terminalei lotutako kanalei (sarrerei eta irteerei) dagokiena, eta bestetik, barne-erreleei dagokiena (ez dagokie kanpo-terminalik).

▶ Gune berezia (SR244-SR255)

Seinaleztapen-erreleen informazioa gordetzen da hemen. Errele horiek funtzio jakin batzuk dituzte, adibidez:

- ✓ Zerbitzua (beti ON edo OFF)
- ✓ Diagnostika (seinaleztapenak edo anomaliak)
- ✓ Tenporizazioak (maiztasun anitzeko ordulariak)
- ✓ Kalkuluak (<, >, =)
- ✓ Komunikazioak

▶ **Gune osagarria (AR00-AR27)**

PLCaren baliabideen kontrola eta informazioa gordetzen dituzten bitak ageri dira, adibidez, ataka periferikoena, memoria-kaseteena, etab.

Gune horiek bi bloketan banaturik daude: alde batetik, seinaleztapena, non konfigurazio-erroreak eta sistemaren datuak irakur daitezkeen, eta bestetik, datuen memorizazioa eta kudeaketa, zeinak atxikipengune batek bezala jokatzen duen.

▶ **Loturagunea (LR00-LR63)**

Gune horiek PC Link 1:1 moduan lotuta dauden PLCen arteko datu-trukaketa egiteko erabiltzen dira; hots, gune horiek, LR moduan erabiltzen ez badira, IR moduan erabil daitezke.

IR, SR, AR, eta LR memoriaren guneen ezaugarri nagusiak hauek dira:

- ✓ Eskuragarriak dira bit edo kanal moduan
- ✓ Unean erabili gabe dauden I/O erreleak IR moduan erabil daitezke
- ✓ Elikadura eten edo eragiketa-modua aldatzen denean ez dute egoera mantentzen

▶ **Atxikipengunea (HR00-HR99)**

Gune hauek beren egoera mantentzen dute elikadura eteten denean edo PLCaren modua aldatzen denean. IR moduan kudeatzen dira.

▶ **Datu-memoria (DM00-DM6143)**

Memoria hauek *hitza* motakoak dira, alegia, 16 bitekoak. Gainera, zenbaki-balioak kudeatzeko erabil daitezke, eta ez dute beren egoera galtzen elikadura eten edo PLCaren egoera aldatzen bada.

Gune honetan gorde ohi dira PLCaren konfigurazioan erabiltzen diren parametroak, *Setupa*, alegia.

▶ **Tenporizadoreak eta kontadoreak (TIM00-TIM511 eta CNT00-CNT511)**

Gailu horien funtzionamendua simulatzen duen gunea da. Atzerapenak eta zenbaketak programatzeko erabiltzen ditu PLCak.

Memoriaren mapa

Memoriaren guneak analizatu ostean, mapa hau da:

DATUEN GUNEA		TAMAINA	HITZAK
IR gunea	Sarrerren gunea	256 bit	IR000tik IR015era
	Irteeren gunea	256 bit	IR100etik IR115era
	Lanerako gunea	Gutxienez 256 bit	IR016tik IR089ra
			IR116tik IR189ra
			IR216tik IR219ra
			IR224tik IR229ra
Controller Link-aren egoeragunea		96 bit	IR090etik IR095era
		96 bit	IR190etik IR195era
Makrogunea	Sarrerren gunea	64 bit	IR096tik IR099ra
	Irteeren gunea	64 bit	IR196tik IR199ra
Balizko txartelerako gunea. 1. <i>slot</i> a		256 bit	IR200etik IR215era
Analogikoen konfigurazioaren gunea		64 bit	IR220tik IR223ra
Abiadura handiko kontadorearen uneko balioa		32 bit	IR230etik IR231ra
Balizko txartelerako gunea 2. <i>slot</i> a		192 bit	IR232tik IR243ra
	SR gunea	184 bit	SR244tik SR255era
	HR gunea	1600 bit	HR00tik HR99ra
	AR gunea	448 bit	AR00tik AR27ra
	TR gunea	8 bit	---
	LR gunea	1024 bit	LR00tik LR63ra
Kontadorearen/tenporizadore aren gunea		512 bit	TIM/CNT 00tik TIM/CNT 511ra
DM gunea	Irakurri/idatzi	3072 hitz	DM0000tik DM3071ra
		3072 hitz	DM0000tik DM6143ra
	Irakurketa soilik	425 hitz	DM0000tik DM6568ra
	Log errorearen gunea	31 hitz	DM0000tik DM6599ra
	PLCaren <i>setupa</i>	56 hitz	DM0000tik DM6655era
EM gunea		6144 hitz	EM0000tik EM6143ra

3.2 Programaren arkitektura

Edozein programa diseinatzean, komeni da pauso jakin batzuk jarraitzea:

1. PLCa aplikatzen den sistemaren ezaugarriak eta eskakizunak definitzea.
2. I/O gailuak identifikatu eta elementu errealekin erlazionatzea esleipen-taula baten bidez. Tauletan informazio hau jaso behar da:
3. Laneko kanalak eta bitak.
4. Tenporizadoreak, kontagailuak eta jauziak.
5. Programatzea (aukeran dauden lengoaietako bat erabiliz).
6. Programa CPUra transferitzea. Programazio-kontsola erabiltzen bada, nahitaezkoa da programa lengoia mnemonikoan idatzia egotea.
7. Programaren funtzionamendu egokia baieztatzea simulazioaren bidez.
8. Behin betiko programa memorian gordetzea.

3.3 Aginduak

Mugimenduak modu logikoan egin ahal izateko aukera ematen diguten parametroak dira aginduak. Hiru atal nagusitan banatzen dira: helbidea, agindua eta parametroak.

1. Helbidea: agindu bakoitzak programan duen posizioa adierazten du, zein lerrotan dagoen, alegia.
2. Aginduak: hainbat eragiketa egiteko erabiltzen diren instrukzioak dira.
3. Parametroak: eragiketa logikoari (aginduari) dagozkion datuak dira. Parametro horiek bi eratakoak izan daitezke: motakoak eta baliokoak.

Adibidez:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	H0501

Hauek dira gehien erabiltzen diren aginduak:

- ✓ LD: zirkuitu-adar bat edo bloke logiko bat irekitzeko behar den agindua da.
- ✓ OUT: irteerako bobina aktibatzen du. Zirkuitu baten amaiera adierazten du.
- ✓ AND: bi kontaktu edo gehiago seriean konektatzeko erabiltzen da. Pneumatikan “ETA” balbulak egiten duen funtzio bera betetzen du.
- ✓ OR: bi kontaktu edo gehiago paraleloan konektatzeko erabiltzen da. Pneumatikan “EDO” balbulak egiten duen funtzio bera betetzen du.
- ✓ NOT: aurretik duena alderantzikatzen du. Gehienbat NI dauden kontaktuak adierazteko erabiltzen da.
- ✓ END: programaren amaiera adierazten du. Nahitaez agertu behar du; bestela, programa exekutatzen.

NOT agindua LD, AND, OR, eta OUT aginduekin konbina daiteke.

Aurreko atalean ikusitako memoriaren gune guztiak erabil daitezke parametro moduan: IR, SR, AR, LR, HR, DM eta TIM/CNT.

3.4 Programazio-lengoiak

PLCa programatzean, lau lengoia erabil daitezke programa idazteko.

Lengoaia mnemonikoa

CPUaren aginduek osatzen dute lengoia hau. Kontrol-funtzioak modu laburtuan adierazita daude. Ez da oso erraza eskema elektrikoaren eta aginduen zerrenda modu intuitiboan erlazionatzea. Hala ere, programatzeko modurik ahaltsuena eta azkarrena da.

Adibidez:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0100
0001	OR	0000
0002	AND NOT	0101
0003	OUT	1000
0004	END	-

Errele-diagrama

Hau errazen ulertzen den lengoia. Arrazoi horregatik da, hain zuzen ere, OMRON PLCetan erabiliena. Gainera, esan behar da beste fabrikatzaileen PLCak errele-diagrama erabiliz programa daitezkeela; esaterako, oso erabilia den SIEMENS PLCa. Izan ere, lengoia mnemonikoa erabili nahi izanez gero, SIEMENSEn agindu espezifikoak ikasi beharko lirateke.

Hona hemen programazioko oinarritzko ikurrak:

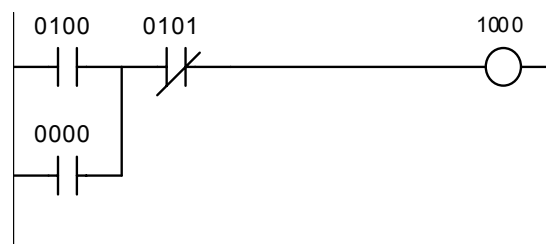


Oharra:

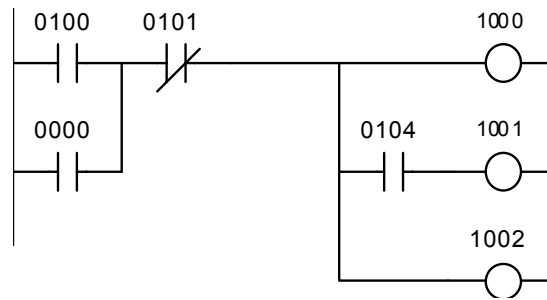
NZ=normalean zabalik eta NI=normalean itxita

Errele-diagrama kontaktu-eskemarekin aldera daiteke. Gainera, kontaktu-eskema eskema elektrikoak marrazteko erabiltzen den kontrol-logikaren oso antzekoa da.

Lengoia horretan idatzitako programa zirkuitu bat osatzen duen adar-seriea litzateke. Adar edo *network* bakoitza kontaktu batzuek osatua dago. Kontaktu horiek seriean edo paraleloan egon daitezke. Kontaktu horien konbinazioaren emaitza irteera bat da. Irteera bobina edo haril baten aktibazioa izan daiteke, edota funtzio berezi baten aktibazioa.

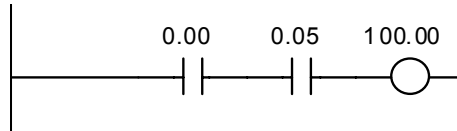


Zirkuituen adar guztiak diagramaren ezker aldean dagoen barra bertikal batean hasten dira. Seinalea ezker-eskuin eta goitik behera irakurri behar da.



Errele-diagramaren zirkuitu-adar bakoitzari lengoia mnemonikoan idatzitako agindu-sekuentzia bat dagokio.

Zirkuitu-adar guztiak LOAD aginduarekin hasten dira.

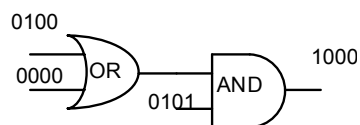


Lengoia mnemonikoan adierazita:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	AND	0.05
0003	OUT	100.00
0004	END	-

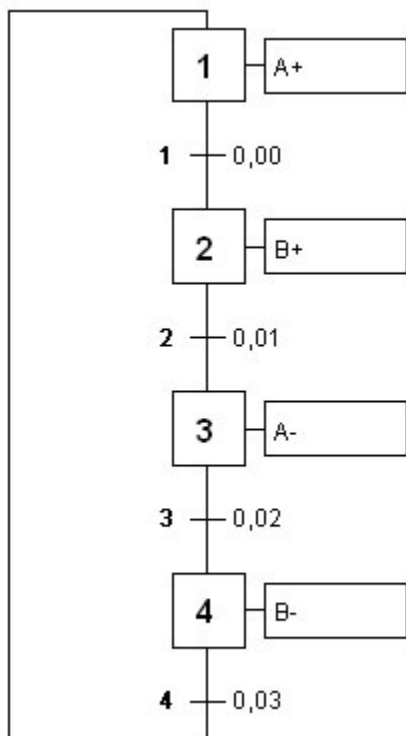
Funtzio-eskema

Funtzio-eskeman, funtzio logiko bakoitzari bloke logiko bat dagokio, eta bloke logiko horrek dagokion funtzioa egiten du. Lengoia honek gertutasun matematiko eta logiko handiagoa behar du.



 **Graficeta**

Lengoaia hau prozesu sekuentzialetan, errepikakorretan edo ziklikoetan erabiltzen da. Egoerak eta trantsizioak automataren funtzioekin batzen dira. Trantsizioak egoeren arteko pauso gisa defini daitezke.

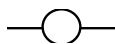


3.5 Oinarrizko aginduak

 **Bit baten egoera begiratzea**

- ▶ **Errelearen irteeraren bobina.**

Ikurra:



- ✓ Ikur horrek zirkuitu-eskemetako bobinek bezala jokatzen du.
- ✓ Korronea bobinaraino iristen bada, irteeraren edo bitaren balioa "1" izango da.
- ✓ Korronea ez bada bobinaraino iristen, irteeraren edo bitaren balioa "0" izango da.
- ✓ Bobinak zirkuituaren eskuineko aldean azaldu behar du nahitaez; ez du azaldu behar tarteko posizioetan.

Irteera ezeztatu bat sor daiteke bobina ezeztatu bat erabiliz.

Ikurra: 

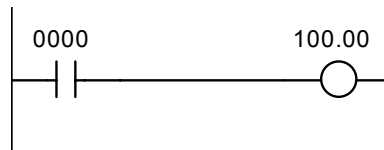
► **Bitak “1” balioa duen begiratzea.**

Bit jakin batek, alegia, sarrera, irteera, marka, tenporizadore, kontadore, etab. batek, “1” balio logikoa duen ala ez jakiteko, ikur hau erabili behar da:

Ikurra: 

- ✓ Ikur hori ixten bada, haren bitaren balio logikoa “1” izango da. Ondorioz, korronteak zeharkatu egingo du eta irteerak, bobinak kasu, “1” balioa hartuko du.
- ✓ Alderantziz, ikur hori irekita badago, haren bitaren balio logikoa “0” izango da. Ondorioz, korronteak ez du zeharkatuko eta irteerak, bobinak kasu, “0” balioa hartuko du.

Kontaktu bidezko adibidea:



0000 kontaktua ixtean, 24 V-eko korrante zuzena igaro eta 100.00 irteerara iritsiko da. Alegia, 0000 sarrerak “1” balio logikoa hartzen duenean, 100.00 irteerak “1” balio logikoa hartuko du, eta alderantziz.

Lengoaia mnemonikoan da:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0000
0001	OUT	100.00
0002	END	-

LD aginduak lerro-hasiera esan nahi du, eta parametro moduan duen bitak “1” balioa hartzen duen egiaztatzen du. Hala bada, ondorengo lerrora pasatu eta hark agintzen diona egingo du. 0000 parametroak ez badu “1” balioa hartzen, alegia, “0” hartzen badu, ez da hurrengo lerrora igaroko.

Eragiketak seriean egin ahal izateko, AND agindua jarri behar da jarraian, eta eragiketak paraleloan egiten jarraitzeko, OR agindua jarri behar da jarraian.

► **Bitak “0” balioa duen begiratzea.**

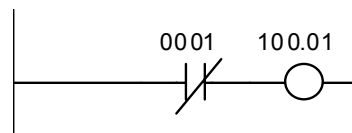
Bit jakin batek, alegia, sarrera, irteera, marka, tenporizadore, kontadore, etab. batek, “0” balio logikoa duen ala ez jakiteko, ikur hau erabili behar da:

Ikurra: 

Ikur hori zabaltzen bada, haren bitaren balio logikoa “1” izango da. Ondorioz, korronteak ez du zeharkatuko eta irteerak bobinak kasu, “0” balioa hartuko du.

Alderantziz, ikur hori itxita badago, haren bitaren balio logikoa “0” izango da. Ondorioz, korronteak zeharkatu egingo du eta irteerak bobinak kasu, “1” balioa hartuko du.

Kontaktu bidezko adibidea:



0001 kontaktua ixtean, 24 V-eko korronte zuzena igaro eta 100.01 irteerara iritsiko da. Alegia, 0001 sarrerak “0” balio logikoa hartzen duenean, 100.01 irteerak “1” balio logikoa hartuko du, eta alderantziz.

Lengoaia mnemonikoan:

<i>HELBIDEA</i>	<i>AGINDUA</i>	<i>PARAMETROA</i>
0000	LD NOT	0001
0001	OUT	100.01
0002	END	-

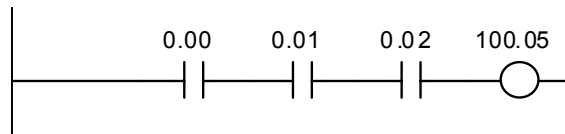
Seriean ezeztatutako kontaktuen eragiketak egin ahal izateko, AND NOT agindua jarri behar da jarraian, eta paralelokoetan, berriz, OR NOT agindua jarri behar da jarraian.

■ And konbinazioa

Ingeleseko AND aginduak euskarazko *eta* ordezkaten du. Agindu hori erabiliz gero, beharrezkoa da une berean zenbait baldintza betetzea irteera aktibatzeko.

► 1. adibidea:

Kontaktuak erabiliz:



Kasu honetan, beharrezkoa da 0.00, *eta*, 0.01, *eta*, 0.02 une berean sakatzea edo ixtea, irteerara korrontea iritsi ahal izateko; alegia, beharrezkoa da hiru kontaktuek une berean "1" balio logikoa edukitzea irteerak "1" balio logikoa izan dezan.

Eta, logika bitarrean, biderketa gisa adierazten da. Beraz:

$$0.00 * 0.01 * 0.02 = 100.05$$

edo balio logikoak ordezkaturaz

$$1 * 1 * 1 = 1$$

Kontaktuen bat bakarrik egonda zabalik, alegia, "0" balio logikoa edukita, bobinara ez da korronterik iritsiko, eta ondorioz, "0" balio logikoa edukiko du. Demagun 0.00 kontaktua ez dagoela sakatua, orduan:

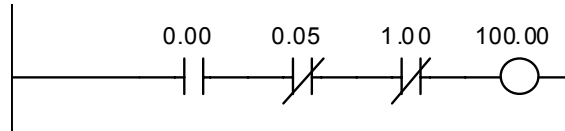
$$0 * 1 * 1 = 0$$

Lengoaia mnemonikoan:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	AND	0.01
0002	AND	0.02
0003	OUT	100.05
0004	END	-

► 2. adibidea:

Kontaktuak erabiliz:



Kasu honetan, beharrezkoa da une berean 0.00 sakatzea, eta 0.05 ez sakatzea, eta 1.00 ez sakatzea bobinara korrontea irits dadin; alegia, beharrezkoa da 0.00 kontaktuak “1” balio logikoa edukitzea, eta 0.05ak “0” edukitzea, eta 1.00k “0” edukitzea 100.00k “1” balio logikoa izan dezan.

Beraz:

$$\overline{0.00} * 0.05 * 1.00 = 100.00$$

edo balio logikoak ordezkaturaz

$$\overline{1} * 0 * 0 = 1$$

edo beste era batera adierazita

$$1 * 1 * 1 = 1$$

Izan ere, “0”ren balio ezeztatu edo alderantzizkoa “1” da, eta “1” balioarena, “0”.

Lengoaia mnemonikoan:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	AND NOT	0.05
0002	AND NOT	1.00
0003	OUT	100.00
0004	END	-

Or konbinazioa

Ingeleseko OR aginduak euskarazko *edo* ordezkaten du. Agindu hori erabiliz gero, beharrezkoa da baldintzaren bat betetzea irteera eduki ahal izateko.

► 1. adibidea:

Kontaktuak erabiliz:



Kasu honetan, nahikoa da 0.00 *edo* 0.01 *edo* 0.02 sakatzea edo ixtea irteerara korronea iritsi ahal izateko; alegia, nahikoa da hiru kontaktuetako batek “1” balio logikoa edukitzea irteerak “1” balio logikoa izan dezan.

Edo, logika bitarrean, batuketa gisa adierazten da. Beraz:

$$0.00+0.01+0.02=100.05$$

edo balio logikoak ordezkaturaz

$$1+1+1=1$$

edo beste konbinazio hauetako bat:

$$1+1+0=1, 1+0+0=1, 1+0+1=1, 0+1+0=1, 0+1+1=1, 0+0+1=1.$$

Kontaktu guztiak zabalik egonda soilik, alegia, “0” balio logikoa edukita, bobinara ez da korronea iritsiko, eta ondorioz, “0” balio logikoa edukiko du. Demagun, 0.00, 0.01 eta 0.02 kontaktuak ez daudela sakatuta. Orduan, hau izango dugu:

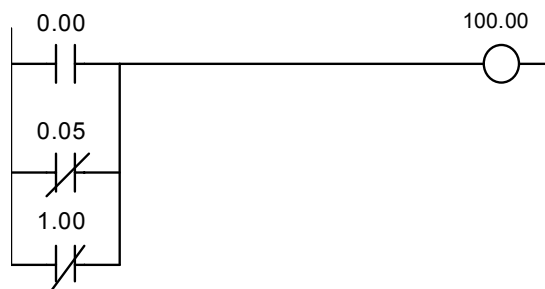
$$0+0+0=0$$

Lengoia mnemonikoan:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	OR	0.01
0002	OR	0.02
0003	OUT	100.05
0004	END	-

► 2. adibidea:

Kontaktuak erabiliz:



Kasu honetan, beharrezkoa da 0.00 sakatzea, edo 0.05 ez sakatzea, edo 1.00 ez sakatzea bobinara korronea irits dadin; alegia, beharrezkoa da 0.00 kontaktuak "1" balio logikoa edukitzea, edo 0.05ak "0" edukitzea, edo 1.00k "0" edukitzea 100.00k "1" balio logikoa izan dezan.

Beraz:

$$\overline{0.00} + 0.05 + 1.00 = 100.00$$

edo balio logikoak ordezkaturaz

$$\overline{1} + 0 + 0 = 1$$

edo beste konbinazio hauetako bat:

$$\overline{1} + 0 + 1 = 1, 1 + 1 + 1 = 1, 1 + 1 + 0 = 1, 0 + 0 + 1 = 1, 0 + 0 + 0 = 1, 0 + 1 + 0 = 1.$$

Izan ere, "0"ren balio ezeztatu edo alderantzizkoa "1" da, eta "1"ena, "0".

Bobinara korronterik ez iristeko modu bakarra konbinazio hau litzateke:

$$\overline{0+1} + 1=0$$

Lengoaia mnemonikoan:

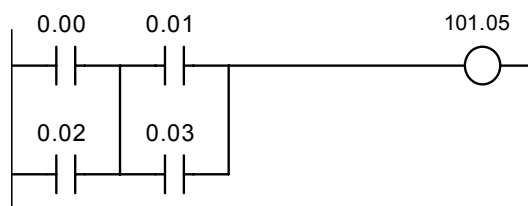
HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	OR NOT	0.05
0002	OR NOT	1.00
0003	OUT	100.00
0004	END	-

AND-LD (OR konbinazioa AND konbinazioaren aurretik doanean)

AND eta LD aginduen arteko konbinazioa da. Bi bloke seriean konektatzeko erabiltzen da.

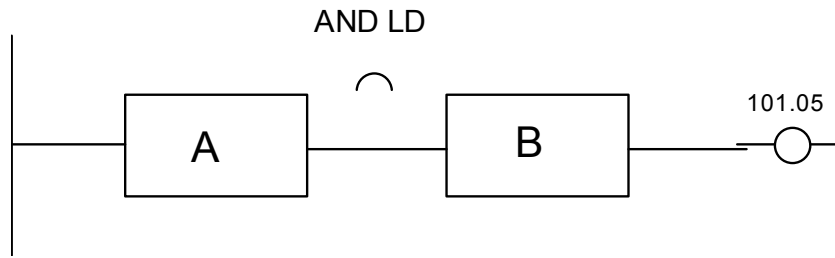
► 1. adibidea:

Kontaktuak erabiliz:



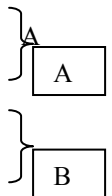
Kasu honetan, korrontea 101.05 bobinara iristeko, gutxienez 0.00 edo 0.02, eta 0.01 edo 0.03 itxi behar dira.

0.00 eta 0.02 kontaktuek bloke bat osatzen dute, eta 0.01 eta 0.03 kontaktuek beste bloke bat.



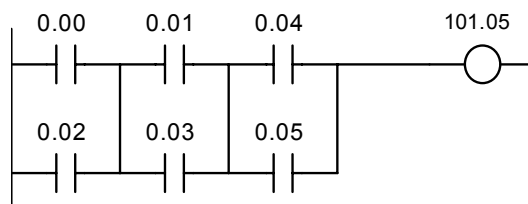
Lengoia mnemonikoan adierazteko garaian, bloke bakoitzak LD aginduarekin hasi behar du, eta ondoren, haien arteko lotura adierazten da. Kasu honetan bi blokeak seriean daudenez, AND-LD agindua erabili behar da seriean daudela adierazteko.

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	OR	0.02
0002	LD	0.01
0003	OR	0.03
0004	AND LD	-
0005	OUT	101.05
0006	END	-



► 2. adibidea:

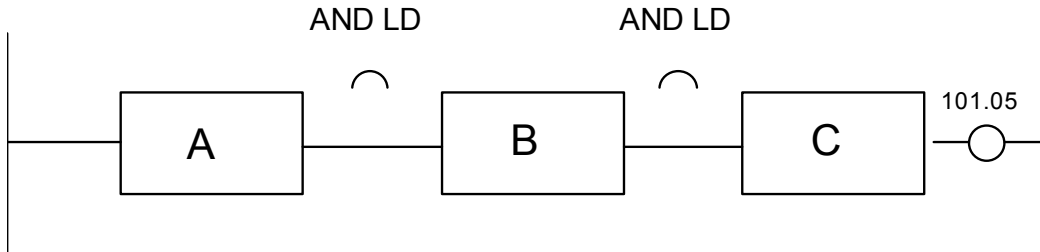
Kontaktuak erabiliz:



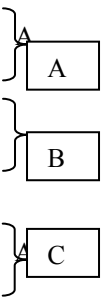
Kasu honetan, korronea 101.05 bobinara iristeko, gutxienez 0.00 edo 0.02 eta 0.01 edo 0.03 eta 0.04 edo 0.05 itxi behar dira.

a) *Metodoa*

Hiru blokeak definitu behar dira lehenengo, eta ondoren, haien arteko lotura adierazi.



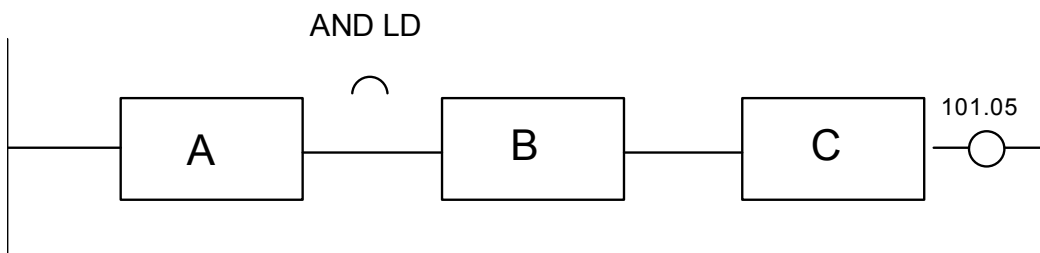
HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	OR	0.02
0002	LD	0.01
0003	OR	0.03
0004	LD	0.04
0005	OR	0.05
0006	AND LD	-
0007	AND LD	-
0008	OUT	101.05
0009	END	-



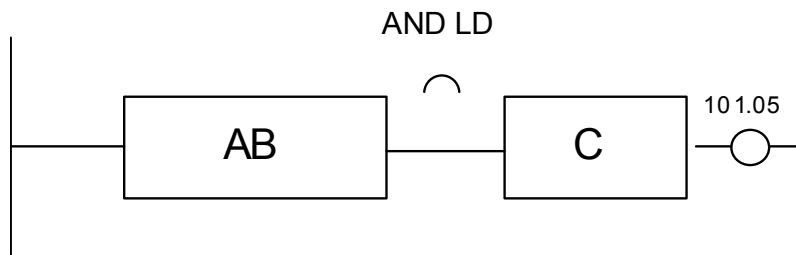
Metodo honen erabilera, ordea, mugatua dago. Izan ere, gehienez 8 bloke defini daitezke jarraian haien arteko lotura adierazi gabe.

b) *Metodoa*

Bi blokeak definitu behar dira lehenengo, eta ondoren, haien arteko lotura adierazi.

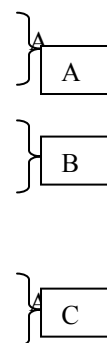


Gero, hirugarren blokea definituko dugu, eta aurrekoekin duen lotura adierazi du.



Lengoia mnemonikoan:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	OR	0.02
0002	LD	0.01
0003	OR	0.03
0004	AND LD	-
0005	LD	0.04
0005	OR	0.05
0007	AND LD	-
0008	OUT	101.05
0009	END	-



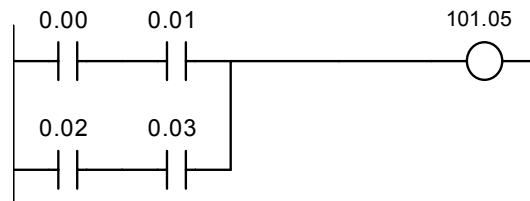
A metodoaren aldean abantaila nagusi bat du, hots, bloke kopuruaren muga noiz gainditzen den kontrolatu behar ez izatea.

■ OR-LD (AND konbinazioa OR konbinazioaren aurretik doanean)

OR eta LD aginduen arteko konbinazioa da. Bi bloke paraleloan konektatzeko erabiltzen da.

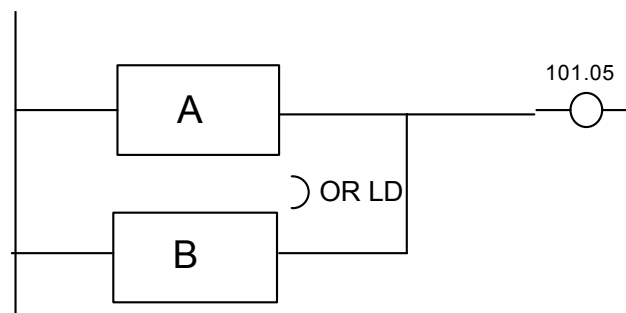
► 1. adibidea:

Kontaktuak erabiliz:



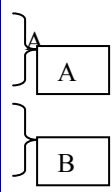
Kasu honetan, korronea 101.05 bobinara iristeko, gutxienez 0.00 eta 0.01 edo 0.02 eta 0.03 itxi behar dira.

0.00 eta 0.01 kontatuek bloke bat osatzen dute, eta 0.02 eta 0.03 kontaktuek, beste bloke bat.



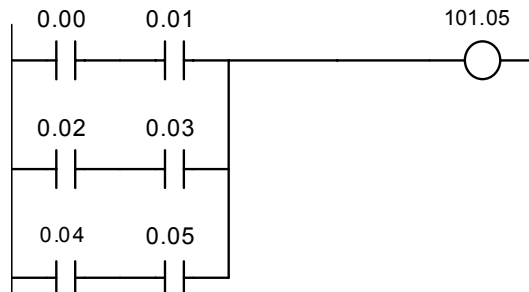
Lengoaia mnemonikoan adierazteko garaian, bloke bakoitzak LD aginduarekin hasi behar du, eta ondoren, haien arteko lotura adierazten da. Kasu honetan bi blokeak paraleloan daudenez, OR-LD agindua erabili beharko da seriean daudela adierazteko.

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	AND	0.01
0002	LD	0.02
0003	AND	0.03
0004	OR LD	-
0005	OUT	101.05
0006	END	-



► **2. adibidea:**

Kontaktuak erabiliz:



Kasu horretan, korrontea 101.05 bobinara iristeko, gutxienez 0.00 eta 0.01 edo 0.02 eta 0.03 edo 0.04 eta 0.05 itxi behar dira.

Hiru blokeak definitu behar dira, eta ondoren, haien arteko lotura adierazi.

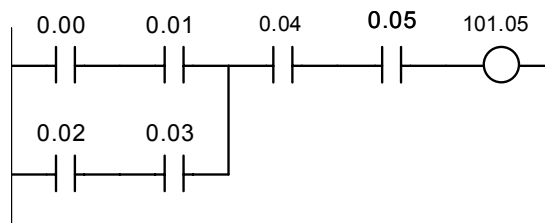
Horretarako, AND LD puntuaren 2. adibidean aztertutako bi metodoak erabil daitezke. Ez ahaztu A metodoa erabiliz gero 8 bloketan dagoela muga.

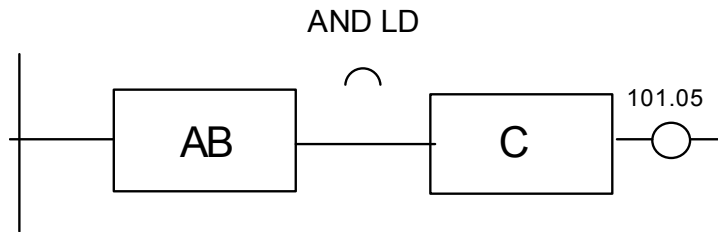
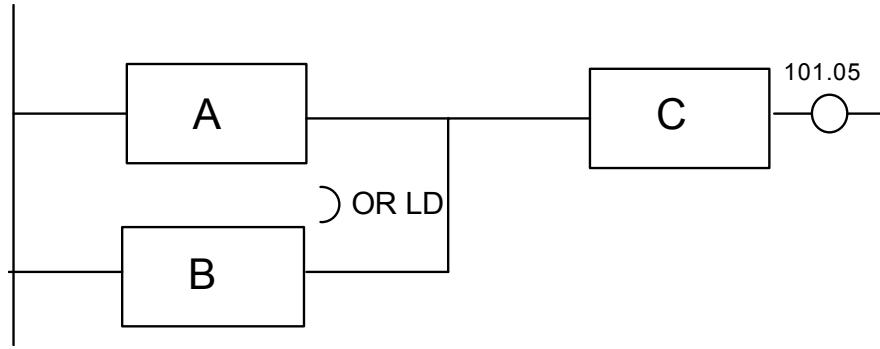
■ **AND eta OR konbinazioak**

AND eta OR hainbat modutan konbina daitezke. Hala ere, hiru konbinazio hauek laburbiltzen dituzte beste guztiak.

► **Paralelo-serie zirkuitua**

Lehenengo pausoa kontaktu-eskema bloketan zatitzea izango da, eta bigarrena, berriz, programa lengoia mnemonikoan eraikitzea; horretarako, kontuan hartu behar dugu blokeak ezker-eskuin eta goitik behera programatzen direla.



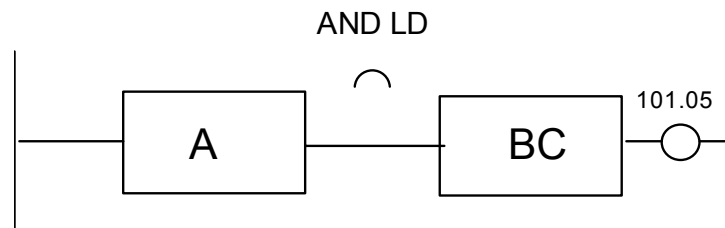
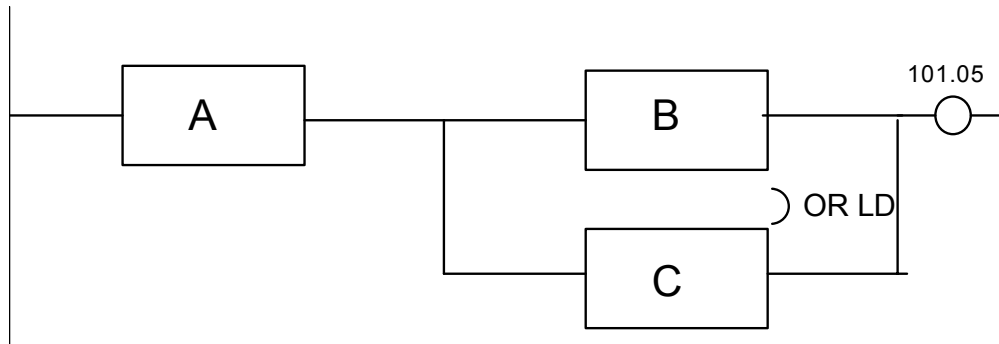
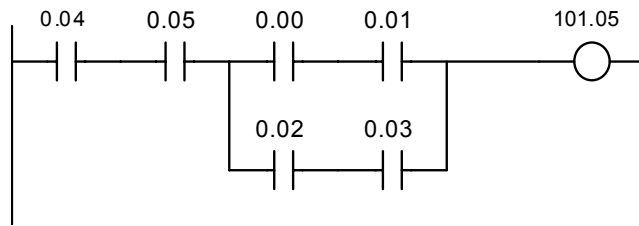


HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	AND	0.01
0002	LD	0.02
0003	AND	0.03
0004	OR LD	-
0005	LD	0.04
0005	AND	0.05
0007	AND LD	-
0008	OUT	101.05
0009	END	-

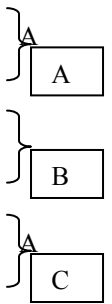
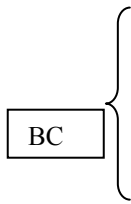
AB

A
A
B
C

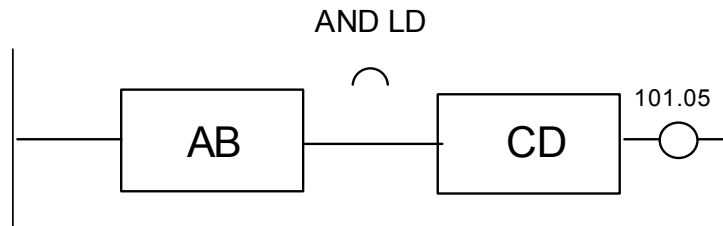
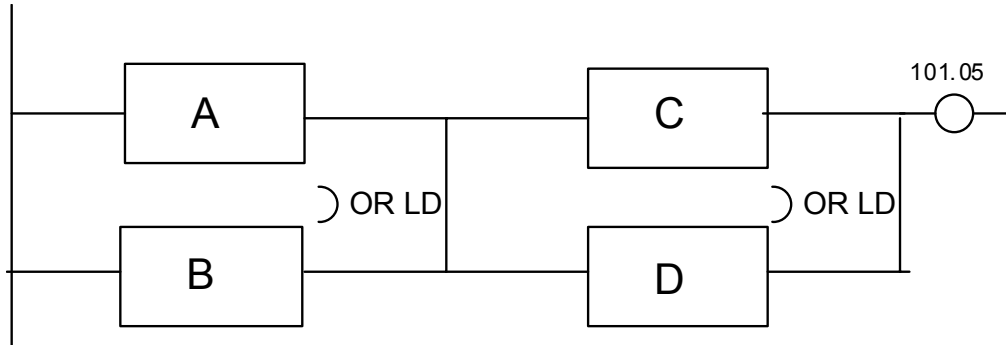
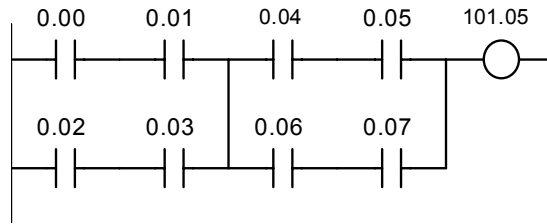
► Serie-paralelo zirkuitua



HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.04
0001	AND	0.05
0002	LD	0.00
0003	AND	0.01
0004	LD	0.02
0005	AND	0.03
0005	OR LD	-
0007	AND LD	-
0008	OUT	101.05
0009	END	-



► Zirkuitu paraleloen serie-konexioa



	HELBIDEA	AGINDUA	PARAMETROA	
AB	0000	LD	0.00	A
	0001	AND	0.01	
	0002	LD	0.02	B
	0003	AND	0.03	
CD	0004	OR LD	-	C
	0005	LD	0.04	
	0006	AND	0.05	D
	0007	LD	0.06	
	0008	AND	0.07	
	0009	OR LD	-	
	0010	AND LD	-	
	0011	OUT	101.05	
0012	END	-		

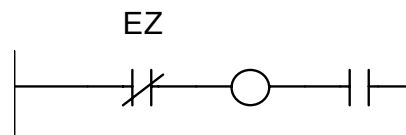
3.6 Programazio-teknikak

Hona hemen programatzeko garaian kontaktu-diagrametan kontuan hartu beharreko zenbait ohar:

1. Erabil daitezkeen kontaktu kopurua mugagabea denez, zirkuitua ahalik eta modurik argi eta sinpleenean eraiki.
2. Kontaktuen kopurua gutxitzen saiatzeak zirkuituaren konplexutasuna areagotu ohi du.
3. Seinalea ezker-eskuin eta goitik behera zabaltzen da.
4. Ezin da zuzenean irteerarik konektatu ezkerreko lerro bertikalean.



5. Seriean edota paraleloan konekta daitezkeen kontaktuen kopurua mugagabea da.
6. Irteera guztiek kontaktu osagarriak dituzte. Kontaktuek autoelikadura sortzeko programa daitezke.
7. Irteera bat programatu ostean, ezin da kontakturik marraztu.



8. Bobina baten zenbakia ezin da zirkuitu berean behin baino gehiagotan erabili; alegia, ezin dira irteerak bikoiztu.
9. Hainbat bobina edo irteera paraleloan programa daitezke.

3.7 Egin beharreko ariketak

Egin itzazu ariketa hauek: 5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.5, 5.1.6, 5.1.7, 5.1.8, 5.2.1, 5.2.2 eta 5.2.3.

Lehenengo pausoa blokeak eta azpiblokeak definitzea da. Bigarren pausoa, berriz, azpiblokeak, haien arteko lotura, blokeak eta haien arteko lotura programatzea izango da.

3.8 Programazio-kontsolaren erabilera

Pasahitza sartzea

Automata programagarriak hiru egoera hauetan lan egiten du:

- ✓ *program* egoeran
- ✓ *monitor* egoeran
- ✓ *run* egoeran.

Lehenik eta behin, automata programagarrian konektatu behar da kontsola. Ondoren, automata piztu, eta kontsolaren pantailan *password* hitza ageriko da.

Programatzen hasi ahal izateko, blokeo hori kendu beharra dago. Horretarako, tekla-segida hau sakatu behar da:

CLR + MONTR

Memoria eta datuak ezabatzea

▶ Ezabaketa osoa

Programa berri bat idatzi baino lehen, nahitaezkoa da lehendik memorian gordeta dagoen programa ezabatzea.

Horretarako, ondoko tekla-segida hau sakatu behar da:

CLR + PLAY/SET + NOT + REC/RESET + MONTR


Oharra

Tekla-segida sakatzen ari garela tekla okerren bat sakatuz gero, CLR tekla sakatu behar da, eta hasieratik sakatu beharko dugu berriz segida.

► **Ezabaketa partziala**

Demagun 0123 helbidean dagoen agindua ezabatu nahi dugula. Horretarako, erabiltzaileak 0123 lerroan kokatu behar du lehenengo, eta ondoren, tekla-segida hau sakatu:

edo

 + 
 + 

 **Helbide jakin bat aukeratzea**

Programaren lerro jakin batera sartzeko, esaterako 0123 helbidera, tekla hauek sakatu behar dira:

 +  +  + 

Aurreko edo hurrengo lerroko agindua ikusteko, goranzko eta beheranzko geziak dituzten teklak erabiltzen dira:

 eta 

 **Programa idaztea**

Hiru pauso hauek jarraitu behar dira:

1. Automata *program* egoeran ipini.
2. Zenbakiak eta instrukzioak dituzten teklak erabili agindua idazteko.
3. Hurrengo agindua idazteko, WRITE tekla sakatu; bestela, agindua ez litzateke memorian gordeta geratuko.

Programa irakurketa

Ikusi nahi den lerroaren helbidea jarri, eta gezien bidez, lerro horretatik gora edo behera mugitzeko aukera dago.

Bada aipatzea merezi duen beste zerbaitekin: monitorizazioa.

Bi aukera daude automatikaren programa exekutatzeko edo simulatzeko: *run* egoera edo *monitor* egoera.

Programa exekutatzen ari dela aldagairen baten egoera ikusi nahi bada, PLCak *monitor* egoeran behar du. Kasu horretan, aldagaiak monitorizatzen ari garela esan ohi da.

PLCa *run* egoeran izanez gero, zenbait aldagai ezingo dira pantailan ikusi.

Programaren azterketa

Behin programa idatzita dagoela, egoki edo arau jakin batzuen arabera idatzia dagoela frogatu dezakegu. Horretarako, nahikoa da tekla-segida hau sakatzea.

CLR + SRCH

Agindu edo instrukzioak bilatzea

Demagun "LD 0002" agindua bilatu nahi dugula. Horretarako, tekla-segida hau sakatu behar dugu.

CLR + LD + SRCH

Agindu-bilaketa egitean ez du garrantzirik PLCak une horretan duen egoerak.

Tekla-segida sakatu ondoren SRCH tekla behin eta berriz sakatzen bada, programan dauden LD guztiak agertuko dira bata bestearen atzetik.

■ Agindu edo instrukzioak txertatzea

Pauso hauek jarraitu behar dira:

1. Automata *program* egoeran jarri.
2. Agindua txertatu nahi den helbidera joan.
3. INS tekla sakatu.
4. Goranzko edo beheranzko gezia duen tekla sakatu.

3.9 Egin beharreko ariketak

5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.5, 5.1.6, 5.1.7, 5.1.8, 5.2.1, 5.2.2 eta 5.2.3. ariketetako programa hartu eta automata programagarrian sartu, programaren funtzionamendu egokia baieztatzeko.

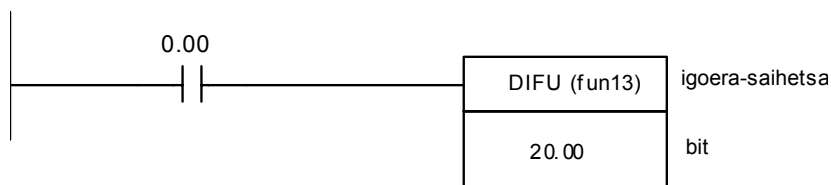
3.10 Agindu-zerrenda

■ Saihets positiboa (DIFU (fun13))

Agindu honen bitartez, kontrolatu nahi den seinalearen saihetsaren igoera atzematen da. Saihetsa igo egingo da seinalea ez-aktibotik aktibora igarotzen denean.

Aginduaren parametroa memoriaren mapan aztertutako edozein bit izan daiteke, hau da, IR, SR, AR, HR edo LR.

Agindu hori honela adierazten da kontaktuak erabiliz:



Kasu horretan, kontrolatu nahi den seinalea 0.00 sarrera da, eta saihetsen bat gertatu den adierazten duena, berriz, 20.00 marka edo bita.

Beraz, 0.00 sarrera aktibatzean, "0" logikotik "1" logikora igaroko da, eta ondorioz, 20.00k ere "1" balioa bereganatuko du. Gainera, 20.00k balio bera mantenduko du *Scan* ziklo osoan.

Lengoia mnemonikoan honela programatzen da saihets baten detekzioa:

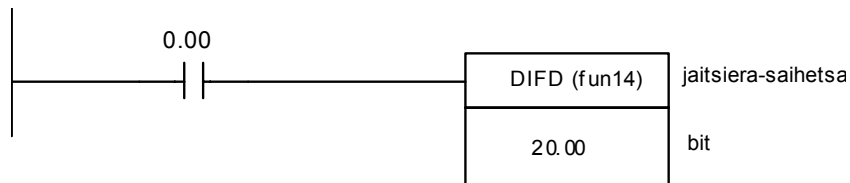
HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	DIFU(fun13)	20.00
0002	END	-

Saihets negatiboa (DIFD (fun14))

Agindu honen bitartez, kontrolatu nahi den seinalearen saihetsaren jaitsiera atzematen da. Saihetsa jaitsi egingo da seinalea aktibotik ez-aktibora igarotzen denean.

Aginduaren parametroa memoriaren mapan aztertutako edozein bit izan daiteke, hau da, IR, SR, AR, HR edo LR.

Agindu hori honela adierazten da kontaktuak erabiliz:



0.00 sarrera desaktibatzean, "1" logikotik "0" logikora igarotzen da, eta ondorioz, 20.00k "1" balioa bereganatuko du. Gainera, 20.00k balio hori mantenduko du *Scan* ziklo oso batean.

Lengoia mnemonikoan honela programatzen da saihets baten detekzioa:

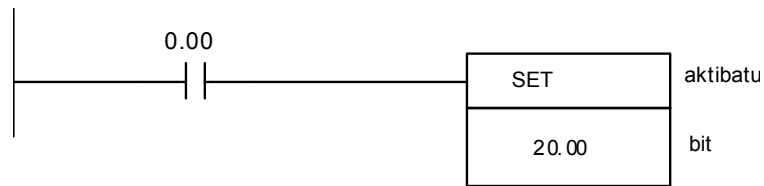
HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	DIFD(fun14)	20.00
0002	END	-

Memoria-eragiketak

Eragiketa hauen bidez, konbinazio jakin baten emaitza memorian gordetzen da.

Aginduaren parametroa memoriaren mapan aztertutako edozein bit izan daiteke, alegia, IR, SR, AR, HR edo LR.

► SET funtzioa



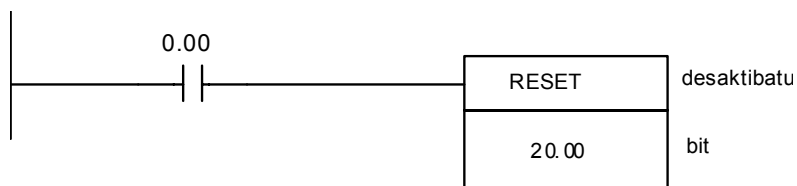
0.00 sarrera aktibatzen denean, haren balioa “0” logikotik “1” logikora igarotzen da. Ondorioz, 20.00 irteera aktibatu edo seteatu egingo da, eta “1” balio logikoa bereganatuko du.

Irteerak “1” balioa mantentzen du, nahiz eta 0.00 sarrera desaktibatu. Irteera horrek “0” balioa berreskuratzeko modu bakarra Set aginduaren kontrakoa jasotzea da, hain zuzen ere, Reset agindua jasotzea.

Lengoaia mnemonikoa erabiliz:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	SET	20.00
0002	END	-

► RESET funtzioa



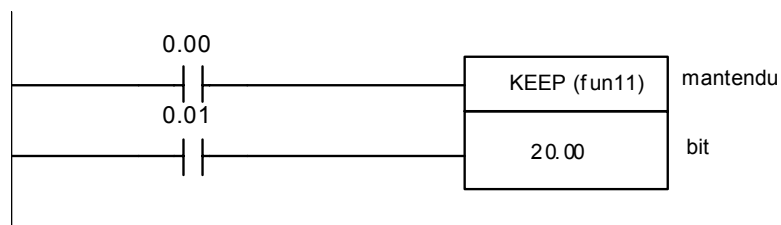
0.00 sarrera aktibatzen denean, haren balioa “0” logikotik “1” logikora igarotzen da. Ondorioz, 20.00 irteera desaktibatu edo reseteatu egingo da, eta “0” balio logikoa bereganatuko du.

Irteerak “0” balioa mantenduko du, nahiz eta 0.00 sarrera desaktibatu. Irteera horrek “1” balioa berreskuratzeko modu bakarra *Reset* aginduaren kontrakoa jasotzea da, hain zuzen ere, *Set* agindua jasotzea.

Lengoaia mnemonikoa erabiliz:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	RESET	20.00
0002	END	-

► **KEEP(fun11) funtzioa**



KEEP agindua *Set* eta *Reset* aginduen arteko konbinazioa da.

0.00 sarrera aktibatzen denean, haren balioa “0” logikotik “1” logikora igarotzen da. Ondorioz, 20.00 irteera aktibatu edo *seteatu* egingo da, eta “1” balio logikoa bereganatuko du.

0.01 sarrera aktibatzen denean, berriz, haren balioa “0” logikotik “1” logikora igarotzen da. Ondorioz, 20.00 irteera desaktibatu edo *reseteatu* egingo da, “0” balio logikoa bereganatuko du.

0.00 eta 0.01 sarrerak une berean aktibatzen badira, *Reset* funtzioak izango du lehentasuna.

Lengoaia mnemonikoa erabiliz:

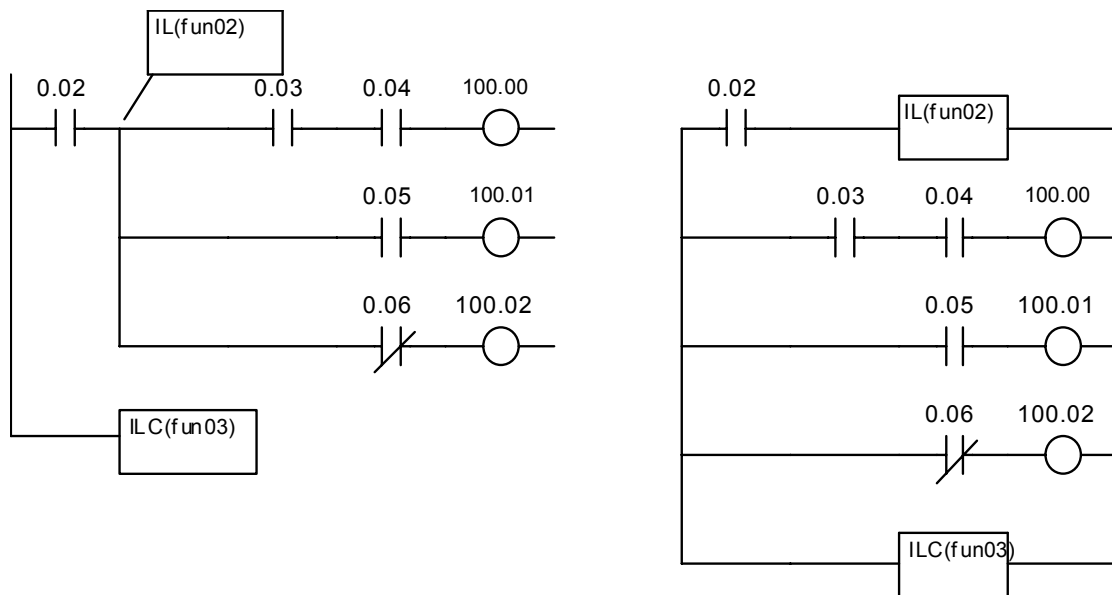
HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	LD	0.01
0002	KEEP (fun11)	20.00
0003	END	-

IL(fun02), ILC(fun03) eta TR aginduak

► IL eta ILC

IL agindua beti erabiltzen da ILC aginduarekin batera. ILren eta ILCren artean ageri diren irteera guztiak desaktibatuta mantenduko dira ILren aurretik dauden baldintzak betetzen ez badira.

Kontaktuak erabiliz, bi era hauetan adieraz daiteke:



Kasu honetan, IL aginduaren aurretik 0.02 kontaktua dago, eta hori sakatu ezean, ez da aktibatuko ILren eta ILCren artean kokatutako irteeretako bat bera ere.

Beraz, IL aginduak baldintzatutako blokearen hasiera adierazten du, eta ILC aginduak, berriz, bloke horren amaiera.

Lengoaia mnemonikoan honela adierazten da:

<i>HELBIDEA</i>	<i>AGINDUA</i>	<i>PARAMETROA</i>
0000	LD	0.02
0001	IL (fun02)	-
0002	LD	0.03
0003	AND	0.04
0004	OUT	100.00
0005	LD	0.05
0006	OUT	100.01
0007	LD NOT	0.06
0008	OUT	100.02
0009	ILC(fun03)	-
0010	END	-

Adierazitako agindu-konbinazio hori oso erabilgarria da lengoaia mnemonikoan. Izan ere, IL eta ILC konbinazioa erabiltzen ez bada, 0.02 izeneko kontaktua behin eta berriz programatu behar da irteera bakoitzeko.

<i>HELBIDEA</i>	<i>AGINDUA</i>	<i>PARAMETROA</i>
<i>0000</i>	<i>LD</i>	<i>0.02</i>
0001	AND	0.03
0002	AND	0.04
0003	OUT	100.00
<i>0004</i>	<i>LD</i>	<i>0.02</i>
0005	AND	0.05
0006	OUT	100.01
<i>0007</i>	<i>LD</i>	<i>0.02</i>
0008	AND NOT	0.06
0009	OUT	100.02
0010	END	-

Argi ikusten denez, kontaktu berbera hiru aldiz programatu behar da.

Adibide hori erreferentziatzen hartuz, IL eta ILC aginduak erabiliz lortutakoarekin alderatuta, pentsa daiteke lerro kopurua ez dela asko gutxitu.

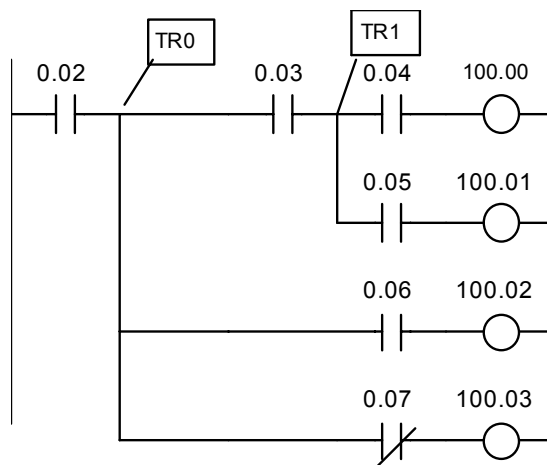
Izan ere, hala da adibide horretan. Baina demagun IL aurretik 0.02 izeneko kontaktu bakarra egon beharrean hamahiru kontaktuz osatutako kontaktu-katea dagoela. Kasu horretan, IL eta ILC aginduak erabilita, hogeita sei lerro gutxiago erabiltzen dira programa berbera idazteko.

Horra hor bi agindu hauek erabiltzearen abantaila nagusia.

► **TR edo behin-behineko memoriako errelea**

Behin-behineko memoriako erreleak irteera bat baino gehiagoko zirkuituetan erabili ohi dira.

LD edo OUT aginduekin batera agertzen da beti. LD aginduak lagunduta, TR memoriak gordea duen informazioa kargatzen da. OUT aginduak lagunduta, berriz, TR memoriaren aurretik doan informazioa gordetzen da du. Kontaktuak erabiliz:



Lengoia mnemonikoa erabiliz:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.02
0001	OUT	TR0
0002	LD	TR0
0003	AND	0.03
0004	OUT	TR1
0005	LD	TR1
0006	AND	0.04
0007	OUT	100.00
0008	LD	TR1
0009	AND	0.05
0010	OUT	100.01
0011	LD	TR0
0012	AND	0.06
0013	OUT	100.02
0014	LD	TR0
0015	AND NOT	0.07
0016	OUT	100.03
0017	END	-

Gorde TR0n

Gorde TR1n

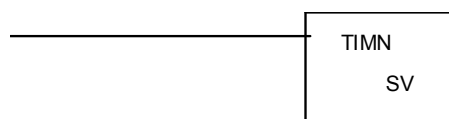
Egin beharreko ariketak

Egin itzazu ariketa hauek: 5.3.1, 5.3.2 eta 5.3.3.

Tenporizadoreak

TIM agindua sarrerako seinalearekiko atzerapen bat sortzeko erabiltzen da.

Ikurra:



TIM: tenporizadorea adierazteko era.

N: tenporizadorearen izena adierazten du; 000tik hasi eta, automataren arabera, 511raino irits daiteke.

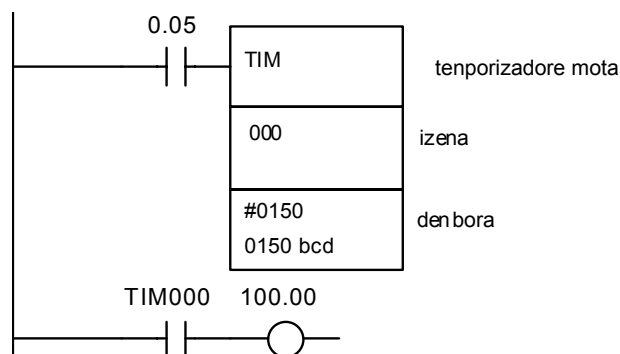
SV edo *start value*: tenporizadoreak hasieran deskontatu beharreko denbora adierazten du. Denbora-tarte horrek 0 segundo eta 999,9 segundo bitartekoa izan beharko du.

Deskontatu beharreko denbora programatzean, kontuan hartu behar da zein tenporizadore mota erabiltzen ari garen. Bi motatakoak izan daitezke:

1. TIM: tenporizadore erabiliena da, eta 0,1 segundoko denbora-oinarria du.
2. TIMH: ez da hain erabilia; abiadura handiko tenporizadore deritzo. 0,01 segundoko denbora-oinarria du.

Beraz, 15 segundo programatu nahi badira eta tenporizadorea TIM motakoa bada, 150 zenbakia teklatu behar da; tenporizadorea TIMH motakoa bada, berriz, 1500 zenbakia teklatu behar da.

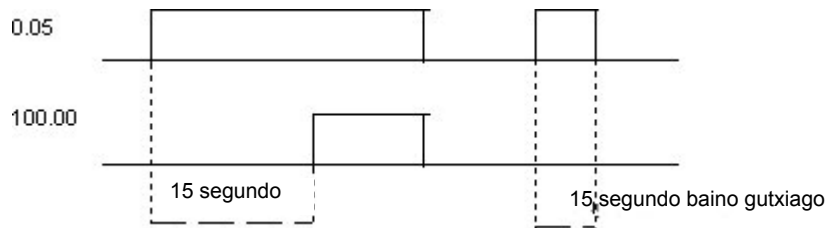
Tenporizadorea honela adierazten da kontaktuen bidez:



Kasu honetan, 0.05 kontaktua ixten denean, korrontea iritsiko da TIM000 tenporizadorera, eta programatuta dagoen denbora —15 segundo alegia— deskontatzen hasiko da. Kontaketa bukatzen denean, tenporizadorearen kontaktuak ireki edo itxi egingo dira, hasieran duten posizioaren arabera. Kasu horretan, TIM000 kontaktua itxi eta 100.00 irteera aktibatu egingo dira.

0.05 kontaktua zabalduz gero, ez da korronterik iritsiko TIM000 tenporizadorera; ondorioz, tenporizadorea *reseteatu* egingo da, eta 15 segundoak hartuko ditu berriz ere hasierako baliotzat. Gainera, TIM000 kontaktua zabalduko da, eta 100.00 irteera desaktibatu egingo da.

Beraz, TIM000 tenporizadoreak deskontaketa buka dezan, beharrezkoa da 0.05 kontaktua 15 segundo baino gehiagoz itxita egotea. Bestela, ez luke deskontatuko eta tenporizadoreak berriz hasierako balioa hartuko luke, 15 segundokoa, alegia.



Lengoaia mnemonikoa erabiliz:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.05
0001	TIM	000
0002	-	#0150
0003	LD	TIM00
0004	OUT	100.00
0005	END	-

Oharra

Tenporizadoreari eta kontadoreari ez zaie izen berdina eman behar; alegia, programa berean ez da agertu behar TIM000 eta CNT000.

 **Egin beharreko ariketak**

Egin itzazu ariketa hauek: 5.4.1, 5.4.2, 5.4.3 eta 5.4.4.

Kontadoreak

Kontadoreen seinale bat zenbat aldiz aktibatu den zenbatzen dute.

Kontuan hartu behar dago programa berean bi kontadorek ez dutela izen berdina eduki behar.

Kontadoreek 0000tik 9999ra bitartean zenbatzen dute, eta haien kontaktuak *normalean zabalik* edo *normalean itxita* motakoak izan daitezke.

Kontadoreen artean bi mota bereizten dira:

- ✓ CNT edo goitik beherako kontadorea
- ✓ CNTR edo kontadore itzulgarria

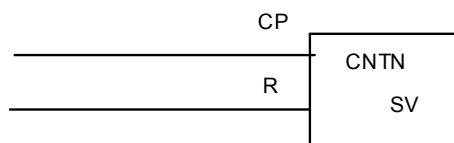
▶ CNT edo goitik beherako kontadorea

Kontadore mota hau balio batekin programatzen da. Sarrera aktibatzen den bakoitzean, kontadoreak une horretan gordeta duen balioari bat kentzen dio. Gainera, kontadorea *reseteatzen* denean, hasieran programatutako balioa berreskuratzen du.

Oharra

Reseta aktibatua baldin badago, zenbaketaren balioa ez da aldatzen, nahiz eta sarrera behin eta berriz aktibatu.

Ikurra:



N: kontadorearen izena adierazten du; 000tik hasi eta, automataren arabera, 511raino irits daiteke.

SV edo *start value*: kontadoreak hasieran izango duen balioa adierazten du. Zenbaketa-tarte hori 0000tik 9999ra bitartekoa izango da.

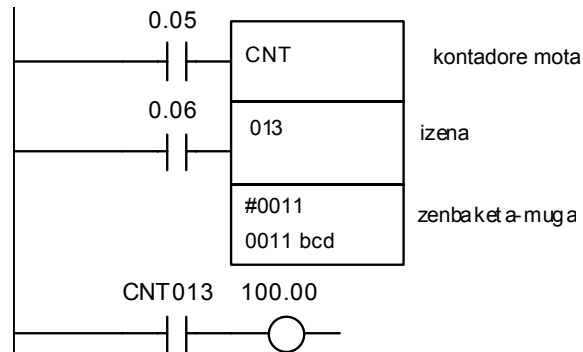
Adibidea:

013 izeneko kontadoreak 11tik 0raino deskonta dezake.

0.05 sarrera aktibatzen den bakoitzean, kontadoreak gordeta duen balioari 1 kenduko dio.

0.06 sarrera aktibatzean kontadorearen *resefeatu* egingo da eta hasierako balioa hartuko du, 11, alegia.

Kontaktuen bidez honela adierazten da:



Lengoia mnemonikoa erabiliz:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.05
0001	LD	0.06
0002	CNT	013
0003	-	#0011
0004	LD	CNT013
0005	OUT	100.00
0006	END	-

► CNTR edo kontadore itzulgarria

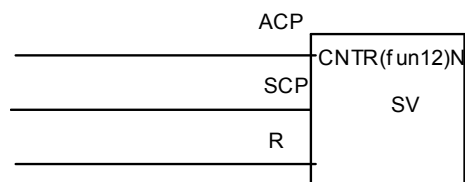
Kontadore mota hau mugatu egin behar da programatzen denean. Konta dezakeen balio maximoa adierazten du muga horrek.

Beste kontadorearen aldean, kontadore itzulgarriak sarrera bat gehiago du. Izan ere, sarrera bat aktibatzen denean, kontadoreak gordetzen duen balioari bat gehituko dio, eta beste sarrera aktibatzen bada, berriz, bat kendu. Gainera, kontadorea *reseteatzen* denean, zero balioa hartuko du.

Oharra

Reseta aktibatua badago, zenbaketa-aren balioa ez da aldatzen, nahiz eta sarrerak behin eta berriz aktibatu.

Ikurra:



N: kontadorearen izena adierazten du; 000tik hasi eta, automataren arabera, 511raino irits daiteke.

SV edo *start value*: kontadoreak izango duen zenbaketa-mugaren balioa adierazten du. Zenbaketa-tarte hori 0000tik 9999ra bitartekoa izango da.

Adibidea:

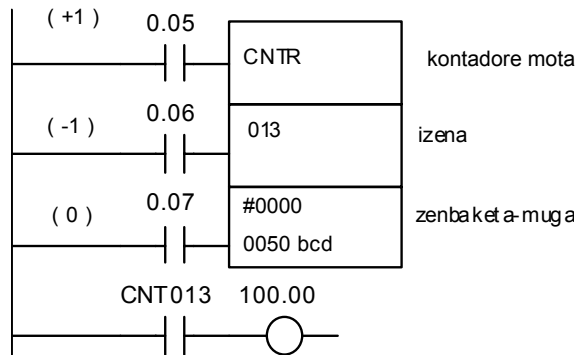
013 izeneko kontadoreak 0tik 50era kontatu dezake.

0.05 sarrera aktibatzen den bakoitzean, kontadoreak gordeta duen balioari 1 gehituko dio.

0.06 sarrera aktibatzen den bakoitzean, kontadoreak gordeta duen balioari 1 kenduko dio.

0.07 sarrera aktibatzean kontadorea *reseteatu* egingo da eta 0 balioa hartuko du.

Kontaktuen bidez, honela adierazten da:

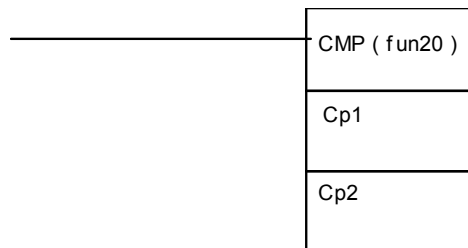


Lengoaia mnemonikoa erabiliz:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.05
0001	LD	0.06
0002	LD	0.07
0003	CNTR	013
0004	-	#0050
0005	LD	CNT013
0006	OUT	100.00
0007	END	-

Konparadoreak

Ikurra:



Agindu horren bidez, bi balio edo datu konpara daitezke, Cp1 eta Cp2, hain zuzen ere.

Hala ere, konparaketa gauzatu ahal izateko, hiru kontaktu berezi behar dira. Hona zein diren kontaktu horiek:

C20, C20K AUTOMATAK	C200H, C1000H, C2000H, C2000, CQM1H AUTOMATAK	C500 AUTOMATA	ESANAHIA
1905	25505	6305	>
1906	25506	6306	=
1907	25507	6307	<

Oharra

Konparadorea ez da zuzenean ezkerreko lerroa lotu behar. Nahitaezkoa da gutxienez kontaktu bat tartean jartzea. Kontaktu horrek NI motakoa izan behar du da.

Adibidea:

Demagun kontadore baten balioa 4 zenbakiarekin konparatu nahi dela. Gainera, kontadorearen balioa 4 baino handiagoa denean, 100.00 LEDa piztuko da.

Kontaktuen bidez honela adierazten da:

Lengoaia mnemonikoa erabiliz:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.05
0001	CMP	-
0002	-	CNT013
0003	-	#0004
0004	LD	25505
0005	OUT	100.00
0006	END	-

2. eta 3. lerroetan balioak gorde egiten ditu, eta 4. lerroan, berriz, konparatu.

Egin beharreko ariketak

Egin itzazu ariketa hauek: 5.5.1 eta 5.5.2.

Datuen desplazamendua

MOV aginduaren bitartez, erregistro batean dagoena beste erregistro batera mugitzen da edo beste erregistro batean kopiatzen da.

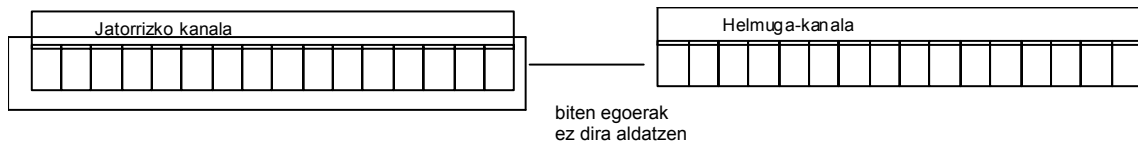
Datuen guneak hauek izango dira:

1. Jatorrizko kanala: IR, SR, AR, DM, HR, TC, LR eta #.
2. Helmuga-kanala: IR, SR, AR, DM, HR eta LR.

Kontadoreak ezin dira helmuga-kanal izan.

Egikaritze-baldintza ON egoeran dagoenean, jatorrizko kanalean dagoena helmuga-kanalean kopiatuko du MOV aginduak; alegia, S kanalean dagoena D kanalean kopiatuko du.

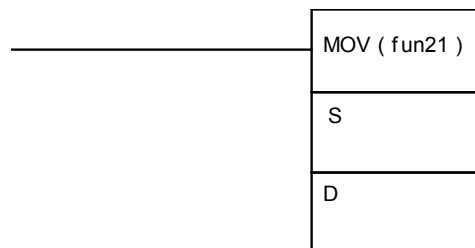
Agindu honen bitartez, datuen transferentzia 16 bit erabilita egingo da.



Datuen transferentzia ondoko bi modu hauetan egin daiteke:

MOVen bitartez.

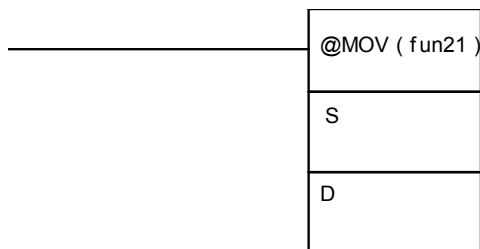
Ikurra:



Egikaritze-baldintza ON egoeran dagoenean, datuak behin eta berriz transferitzen dira. Beraz, transferitutako bitetan aldaketarik bada, aldaketa hori berehala egingo da.

@MOVen bitartez.

Ikurra:

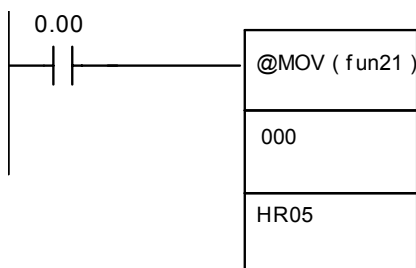


Kasu honetan, berriz, egikaritze-baldintza OFF egoeratik ON egoerara igarotzean soilik egingo da datu-transferentzia. Beraz, transferitutako bitetan aldaketarik bada, agindua beste behin eman behar da.

Adibidea:

IR000n dagoena HR05era transferitu behar da.

Kontaktuen bidez honela adierazten da:



Lengoaia mnemonikoa erabiliz:

HELBIDEA	AGINDUA	PARAMETROA
0000	LD	0.00
0001	@MOV (fun21)	-
0002	-	000
0003	-	HR05
0004	END	-

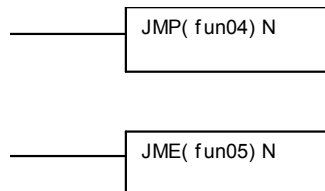
Beraz, biten transferentzia hau izango da:



Jauziak

Agindu hau programaren puntu batetik bestera jauzi egiteko erabiltzen da.

Ikurra:

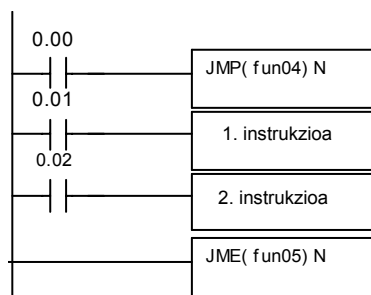


JMP aginduak jauziaren abiapuntua adierazten du, eta JME aginduak, berriz, jauziaren bukaera.

Agindu bat erabiltzeak bestea erabiltzea eskatzen du nahitaez. Jauzi-zenbakiak (N) 01etik 99rakoak izan behar du, baina ezin da errepikatu; alegia, jauzi bakoitzak bere zenbakia izango du. Hala ere, badago salbuespen bat: 00 jauzia nahi den adina erabil baitaiteke.

Jauzia egiteko JMP baldintzatzen duen instrukzioa ON denean, ez da jauzirik gertatuko. Instrukzioa OFF denean, berriz, programak jauzi egingo du zenbaki bera duen JME aginduraino. Orduan, bi agindu horien artean dauden instrukzioak ez dira exekutatu, eta programak bere ibilbidea jarraituko du JMEtik aurrera.

Kontaktuen bidez honela adierazten da:



Lengoaia mnemonikoa erabiliz:

<i>HELBIDEA</i>	<i>AGINDUA</i>	<i>PARAMETROA</i>
0000	LD	0.00
0001	JMP(fun04)	01
0002	LD	0.01
0003	1.Instrukzioa	
0004	LD	0.02
0005	2.Instrukzioa	
0006	JME(fun05)	01
0007	END	-

Aztertu berri dugun aginduak aurretik aztertutako beste baten antzeko funtzioa du; ILren eta ILCrenaren antzekoa, hain zuzen ere.

3.11 Egin beharreko ariketak

Egin itzazu ariketa hauek: 5.6.1, 5.6.2, 5.6.3, 5.6.4 eta 5.6.5.

4 GRAFCETAK

4.1 *Grafceterako sarrera*

Grafceta automatismo sekuentzial batek izan ditzakeen jokabide askotarikoak grafikoki adierazteko erabiltzen den diagrama da.

Grafceta fase, giltza, etiketa eta arkuz osatutako adierazpen grafikoa da.

Kontaktuen eskema lortzeko jarraitu beharreko pausoetako bat da *grafceta*.

Eskema hori lortzeko, pauso hauek jarraitu behar dira, bata bestearen atzetik:

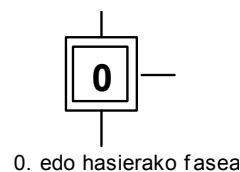
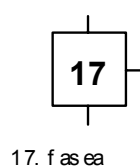
- ▶ **1. pausoa:** indar-zirkuitua eraiki. Zirkuitu hori elektro-pneumatikoa izan ohi da.
- ▶ **2. pausoa:** I. mailako *grafceta* osatu.
- ▶ **3. pausoa:** izendatzeak egin.
- ▶ **4. pausoa:** II. mailako *grafceta* osatu.
- ▶ **5. pausoa:** ekuazio logikoak ebatzi.
- ▶ **6. pausoa:** ekuazio logikoak oinarritzat hartuz, kontaktuen eskema osatu.

4.2 *Grafceta* eraikitzeko kontzeptu nagusiak

Fasea

Funtzionamendu-zikloan zehar automatismoaren jokabidea konstante mantentzen den egoera da.

Faseak karratu bat eta zenbaki bat erabiliz adierazten dira. Hasierako edo lehen fasea adierazteko, berriz, karratu bikoitza erabiltzen da.



Fase bati lotutako ekintzak

Fase bakoitzari hainbat ekintza lot dakizkioke. Akzio horiek laukizuzen baten barnean adierazten dira; laukizuzen hori fasearen ikurraren eskuinaldean marrazten da, eta etiketa izena hartzen du.

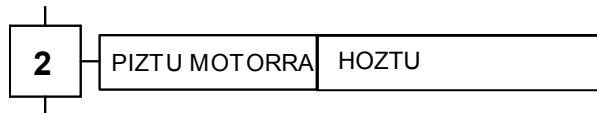
Adibidez:



Fase bati loturiko ekintza bat gertatzen denean, fase hori aktibatua dagoela esaten da. Hori grafikoki adierazteko, fasearen karratuaren barnean, fasearen zenbakiaren azpian, puntu beltz bat marrazten da.

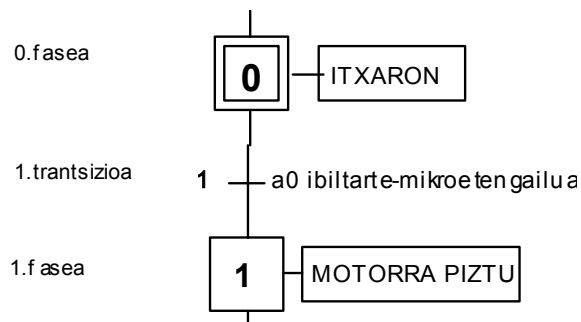


Kasu batzuetan, fase bati ekintza bat baino gehiago dagozkio. Honela adierazten da:



Giltza edo trantsizioa

Giltza edo trantsizioa jarraian dauden bi fase bereizten dituen barrera da. Hesi horrek trantsizioa gainditzeko bete behar den baldintza logikoa adierazten du. Hala ere, baldintza hori betetzea ez da beti nahikoa izango trantsizioa gainditzeko. Izan ere, kasu askotan aurreko faseak aktibatuz egon beharko du.



Arkuak

Arkuak fasea eta trantsizioa lotzen dituen lerro zuzenari deritzo.

Adibidea:

Demagun efektu bikoitzeko bi zilindroren —A eta B— aginte-zirkuitua automata baten bitartez egin nahi dugula. Horretarako, kontaktuen eskema lortzea izango dugu helburu.

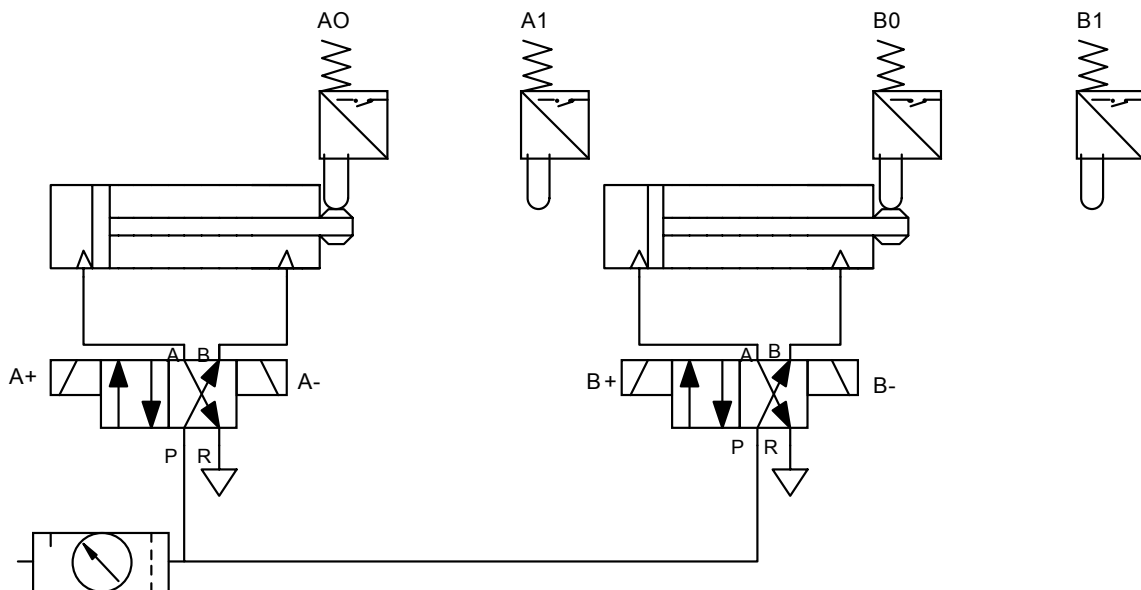
1. pausoa eskema elektro-pneumatikoa osatzea da.

Hau izango da mugimenduen sekuentzia:

- ✓ A zilindroaren zurtoina atera
- ✓ B zilindroaren zurtoina atera
- ✓ A zilindroaren zurtoina sartu
- ✓ B zilindroaren zurtoina sartu

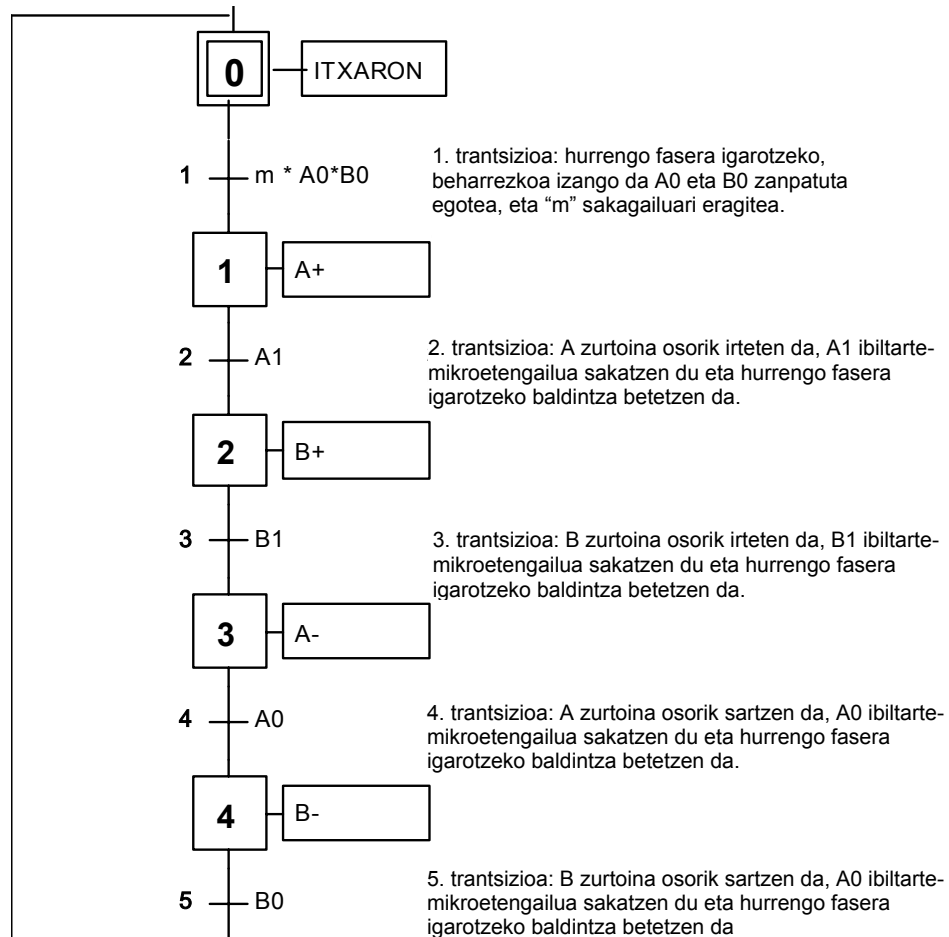
Honela adierazten da hori era laburtuan: A+, B+, A-, B-.

Gainera, zilindro bakoitzak bi ibiltarte-mikroetengailu izango ditu. Banaketa-balbula, berriz, eragite bikoitzekoa izango da.



Mugimenduen sekuentzia hasteko, beharrezkoa izango da zilindro bakoitzaren zurtoina barruan egotea, alegia, A0 eta B0 zanpatuta egotea, eta sakagailu bati eragitea; esaterako, m izeneko sakagailuari. Beraz, mugimendu-katea hasi baino lehen, sistemak pausagune-egoeran egon behar du.

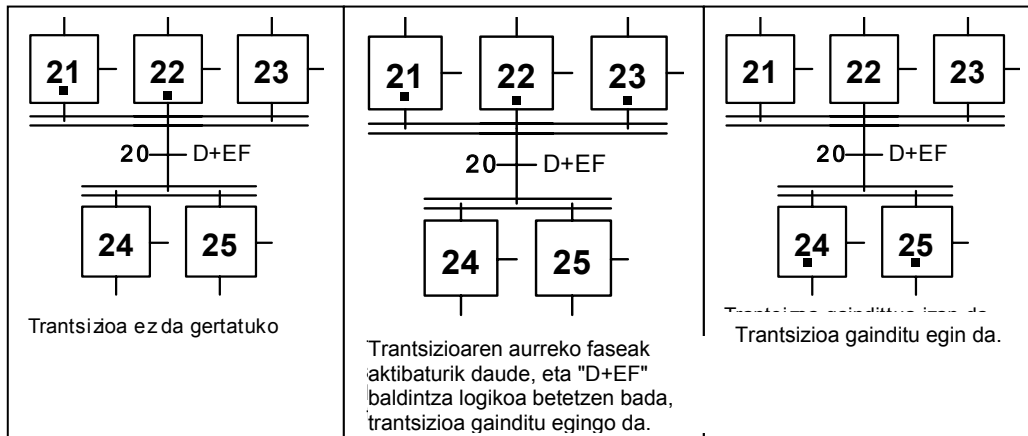
Eskema elektro-pneumatikoa osatu denean, **2. pausoa graficeta** eraikitzea izango da:



4.3 Graficetaren eboluzio-erregelak

Graficetako desplazamenduak garbi ulertzeko, komeni da hiru eboluzio-erregela hauek ezagutzea:

Aurretik duen fasea aktibatua badago eta baldintza logikoa betetzen bada soilik gainditu daiteke trantsizioa. Adibidez:



Trantsizioa gainditzen denean, jarraian dagoen hurrengo fasea aktibatu egingo da eta aurrekoa desaktibatu.

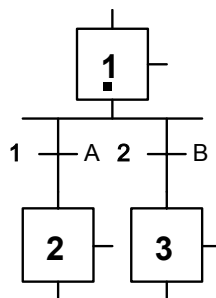
Fase batek aldi berean aktibatuzeko eta desaktibatuzeko agindua jasotzen badu, fase hori aktibatuz geratuko da.

Eboluzio-erregelak ohikoak diren egituretan aplikatzen badira, hau gertatzen da:

Edo DIBERGENTZIA

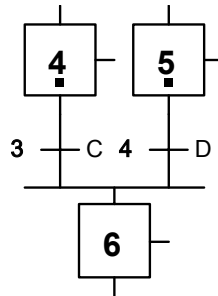
2. fasea aktibatuzeko, beharrezkoa izango da 1. fasea aktibatuta egotea eta A baldintza logikoa betetzea.

3. fasea aktibatuzeko, beharrezkoa izango da 1. fasea aktibatuta egotea eta B baldintza logikoa betetzea.



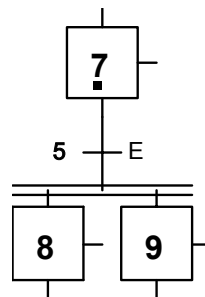
Edo KONBERGENTZIA

6. fasea aktibatzeko, beharrezkoa izango da 4. fasea aktibatuta egotea *eta* C baldintza logikoa betetzea, *edo* 5. fasea aktibatuta egotea *eta* D baldintza logikoa betetzea.



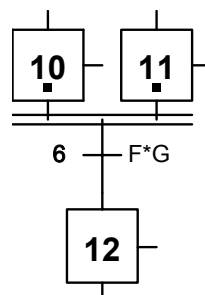
Eta DIBERGENTZIA

8. fasea *eta* 9. fasea une berean aktibatzeko, beharrezkoa izango da 7. fasea aktibatuta egotea *eta* E baldintza logikoa betetzea.



Eta KONBERGENTZIA

12. fasea aktibatzeko, beharrezkoa izango da 10. fasea *eta* 11. fasea aktibatuta egotea *eta* F*G baldintza logikoa betetzea.

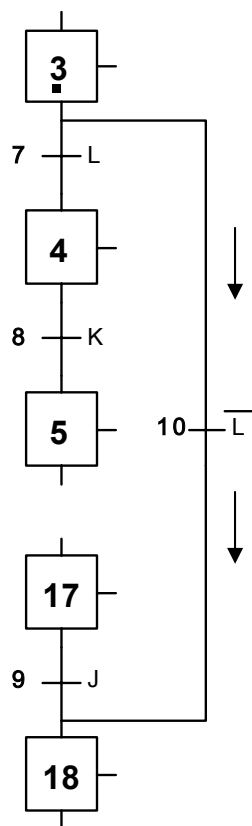


■ Zenbait faseko jauzi baldintzatuak

► **Jauzi baldintzatuak:**

L baldintza logikoa desaktibatua badago, alegia, $L=0$ baldin bada, 3. faseetik 18. faserako jauzia gertatuko da.

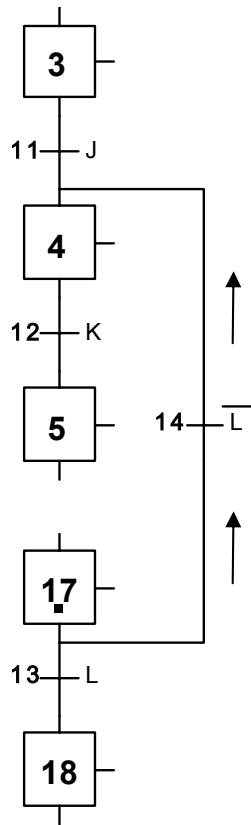
Aldiz, L baldintza logikoa aktibatua badago, alegia, $L=1$, sekuentziak bere ohiko eboluzioa jarraituko du: 3. faseetik 4. fasera igaroko da, 4. faseetik 5. fasera, 5. faseetik 6. fasera, etab.



► **Errepikapen baldintzatuak:**

L baldintza logikoa desaktibatua badago, alegia, $L=0$ baldin bada, 4. faseetik 17. faserako sekuentzia errepikatu egingo da.

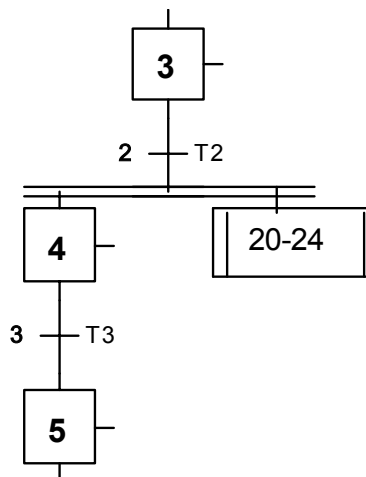
Aldiz, L baldintza logikoa aktibatua badago, alegia, $L=1$, sekuentziak bere ohiko eboluzioa jarraituko du: 17. faseetik 18. fasera igaroko da, eta fase gehiago badaude, aurrera jarraituko du.



Sekuentzia errepikatua

Kasu batzuetan, programaren hasieran agertutako fase-sekuentzia bat berriz errepikatzen da programan zehar. Kasu horietan, nahikoa da oso-osorik idaztea programaren hasieran, eta ondoren, sekuentzia agertzen den bakoitzean, haren hasierako eta bukaerako fasearen zenbakiak bi lerro bertikal dituen karratu baten barnean adieraztea.

Adibidez, demagun 20. fasetik 24. faserara sekuentzia errepikatzen dela 3. fasea bukatzean. Honela adierazten da:



4.4 II. Mailako *grafceta*

Orain arte aztertutako guztia I. mailako *grafcetari* zegokion. Hala ere, eta sarreran aipatu dugun moduan helburua kontaktuen eskema lortzea denez, hona hemen 3. eta 4. pausoak adibide bera oinarritzat hartuta:

3.pausoa: izendatzeak.

II. mailako *grafceta* egiteko, lehenik eta behin, I. mailako *grafcetean* aginduak, faseak eta giltzak izendatzeko erabili diren izenak automatan erabiliko direnekin ordezkatu behar dira. Gure kasuan hauek izango dira izendatzeak :

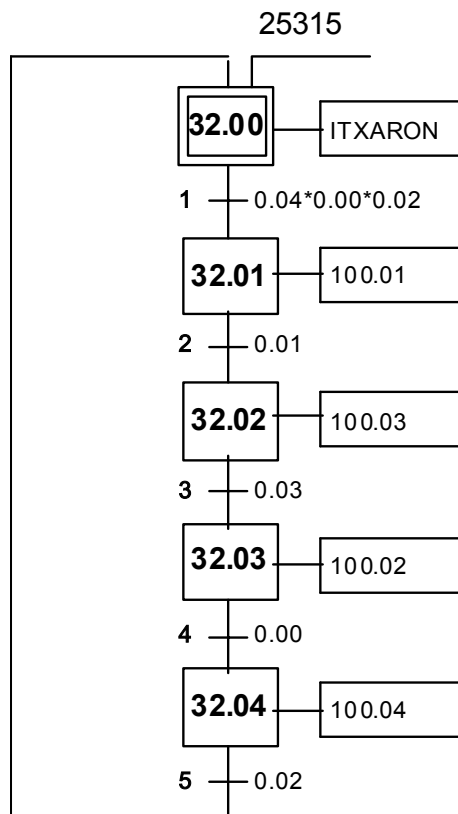
SARRERAK		IRTEERAK		FASEAK	
A0	0.00	A+	100.00	0	32.00
A1	0.01	A-	100.01	1	32.01
B0	0.02	B+	100.02	2	32.02
B1	0.03	B-	100.03	3	32.03
M	0.04			4	32.04

Oharra:

32.00, 32.01, 32.02, 32.03 eta 32.04 automataren barne-erreleak dira.

4. pausoa: II. mailako *grafceta*.

Izendatzeak erabakita daudenean, izendapen berriein II. mailako *Grafceta* osatuko dugu:

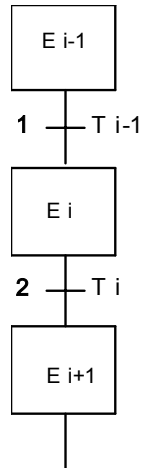


Oharra: 25315 automataren barne-errelea da. Haren funtzioa da automata RUN egoerara pasatzean 32.00 seinalea edo korrontea helaraztea. 25315 barne-errelea erabiliko ez bagenu, ez litzateke sekula korronterik iritsiko 32.00ra, eta ondorioz, programa ez litzateke inoiz ere martxan jarriko.

4.5 Ekuazio logikoak

II. mailako *grafceta* osatua dagoenean, ekuazio logikoak lor daitezke eta, horien bitartez, kontaktuen eskema.

Hurrengo *grafceta* oinarritzat hartuz, Ei fasearen aktibazioa adierazten duen ekuazio logikoa lortu nahi da.



Bi aukera hauek ditugu:

- **Kitzikapenik ezak lehentasuna duen memoriarena.**

Fi fasea gertatzeko, beharrezkoa da aurreko fasea *eta* aurreko trantsizioa aktibatua egotea, *edo* Fi fasea bera aktibatua egotea.

$$F_{i-1} * T_{i-1} + F_i$$

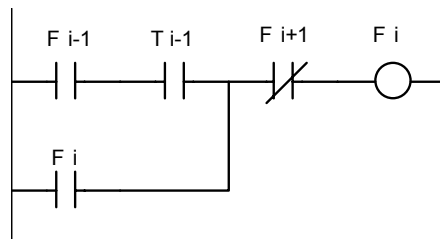
Gainera, Fi fasea gertatzen bada, ezin da une berean hurrengo fasea gertatu.

$$\overline{F_{i+1}}$$

Ondorioz, Fi fasearen ekuazio logikoa hau da:

$$F_i = (F_{i-1} * T_{i-1} + F_i) * \overline{F_{i+1}}$$

Kontaktuak erabiliz, honela adierazten da:



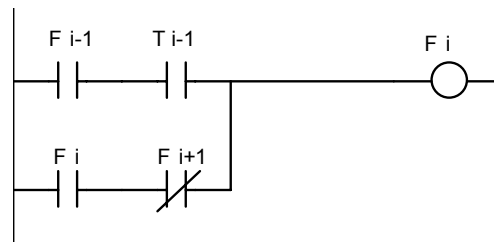
► **Kitzikapenak lehentasuna duen memoriarena.**

Fi fasea gertatzeko, beharrezkoa da aurreko fasea *eta* aurreko trantsizioa aktibatuta egotea, *edo* Fi fasea bera aktibatua egotea *eta* hurrengo fasea, Fi+1, ez aktibatzea.

Ondorioz, Fi fasearen ekuazio logikoa hau da:

$$F_i = F_{i-1} * T_{i-1} + F_i * \overline{F_{i+1}}$$

Kontaktuak erabiliz, honela adierazten da:



Oharra

Ekuazio logikoak ebaztean bi aukera horiek erabil daitezkeen arren, bietako bat aukeratzea eta ahal den guztietan bera erabiltzea gomendatzen da.

Gure adibidearekin jarraituz, ebatz ditzagun ekuazio logikoak II. mailako *grafcetetik*, bi aukerak erabiliz:

■ **5. pausoa: ekuazio logikoak**

► **Kitzikapenik ezak lehentasuna duen memoriarena.**

32.00 fasearen ekuazio logikoa:

32.00 fasea gertatzeko, beharrezkoa da aurreko fasea, 32.04, *eta* aurreko trantsizioa, 0.02, aktibatua egotea, *edo* 25315 errelea aktibatuta egotea, *edo* 32.00 fasea bera aktibatua egotea.

$$32.00 = 32.04 * 0.02 + 25315 + 32.00$$

Gainera, 32.00 fasea gertatzen bada, ezin da une berean hurrengo fasea, 32.01, gertatu.

$$\overline{32.01}$$

Ondorioz, 32.00 fasearen ekuazio logikoa hau da:

$$\overline{32.00} = (32.04 * 0.02 + 25315 + 32.00) * \overline{32.01}$$

32.01 fasearen ekuazio logikoa:

32.01 fasea gertatzeko, beharrezkoa da aurreko fasea, 32.00, eta aurreko trantsizioa, 0.00, 0.02 eta 0.04, aktibatua egotea, edo 32.01 fasea bera aktibatua egotea.

$$32.01 = 32.00 * 0.00 * 0.02 * 0.04 + 32.01$$

Gainera, 32.01 fasea gertatzen bada, ezin da une berean hurrengo fasea, 32.02, gertatu.

$$\overline{32.02}$$

Ondorioz, 32.01 fasearen ekuazio logikoa hau da:

$$32.01 = (32.00 * 0.00 * 0.02 * 0.04 + 32.01) * \overline{32.02}$$

Hori guztia **beste faseei** aplikatuz gero, hauek izango dira ekuazio logikoak:

$$32.02 = (32.01 * 0.01 + 32.02) * \overline{32.03}$$

$$32.03 = (32.02 * 0.03 + 32.03) * \overline{32.04}$$

$$32.04 = (32.03 * 0.00 + 32.04) * \overline{32.00}$$

- ▶ **Kitzikapenak lehentasuna duen memoriarena.**

32.00 fasearen ekuazio logikoa:

32.00 fasea gertatzeko, beharrezkoa da aurreko fasea, 32.04, eta aurreko trantsizioa, 0.02, aktibatua egotea, edo 25315 errelea aktibatuta egotea, edo 32.00 fasea bera aktibatua egotea eta hurrengo fasea, 32.01, ez aktibatzea.

Ondorioz, 32.00 fasearen ekuazio logikoa hau da:

$$\overline{32.00} = 32.04 * 0.02 + 25315 + 32.00 * \overline{32.01}$$

Hori guztia **beste faseei** aplikatuz gero, hauek izango dira ekuazio logikoak:

$$32.01 = 32.00 \cdot 0.00 + 0.02 \cdot 0.04 + 32.01 \cdot 32.02$$

$$32.02 = 32.01 \cdot 0.01 + 32.02 \cdot 32.03$$

$$32.03 = 32.02 \cdot 0.03 + 32.03 \cdot 32.04$$

$$32.04 = 32.03 \cdot 0.00 + 32.04 \cdot 32.00$$

4.6 Kontaktuen eskemak osatzea ekuazio logikoak oinarritzat hartuz

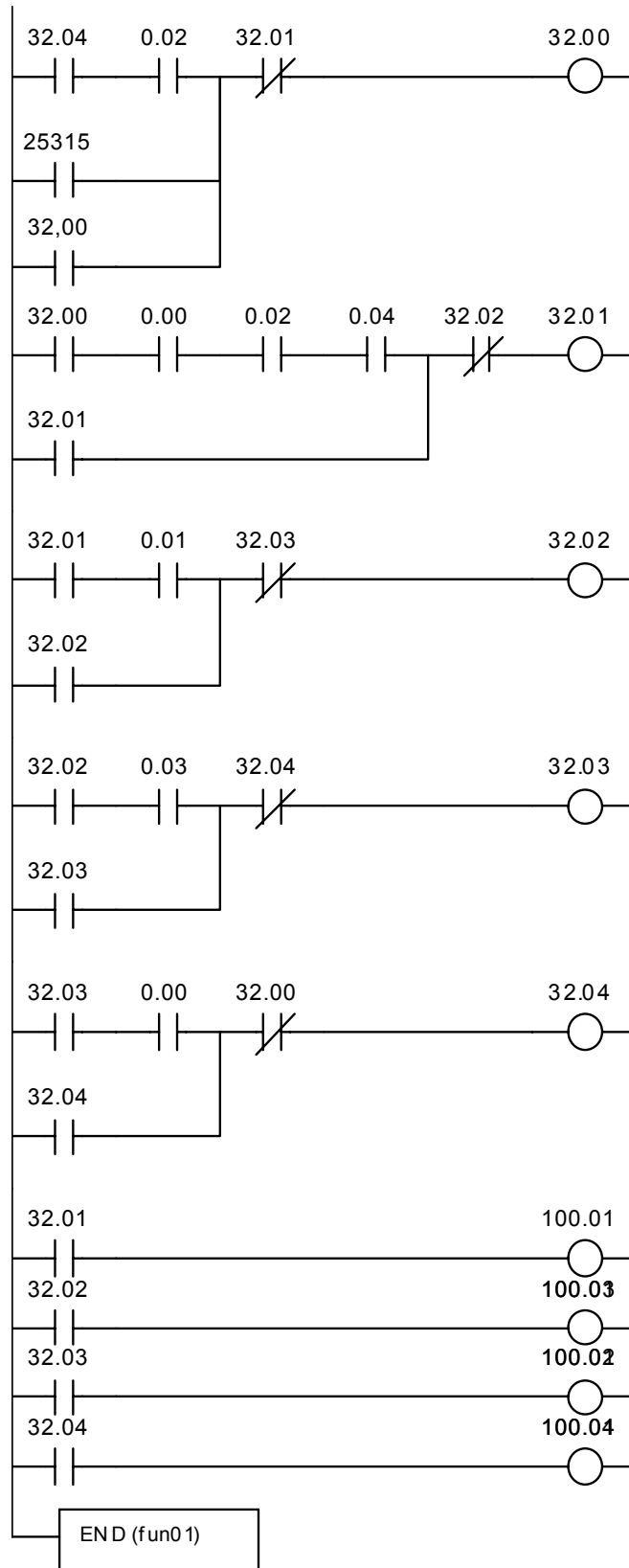
Hau da sistema sekuentzial bat programatu ahal izateko eman behar den **6. pausoa** eta azkena. Kontuan hartu behar da logika bitarrean zeinu hauek duten esanahia:

1. kontaktuen arteko biderketak bi kontaktu horiek seriean daudela adierazten du
2. kontaktuen arteko batuketak, berriz, bi kontaktu horiek paraleloan daudela adierazten du

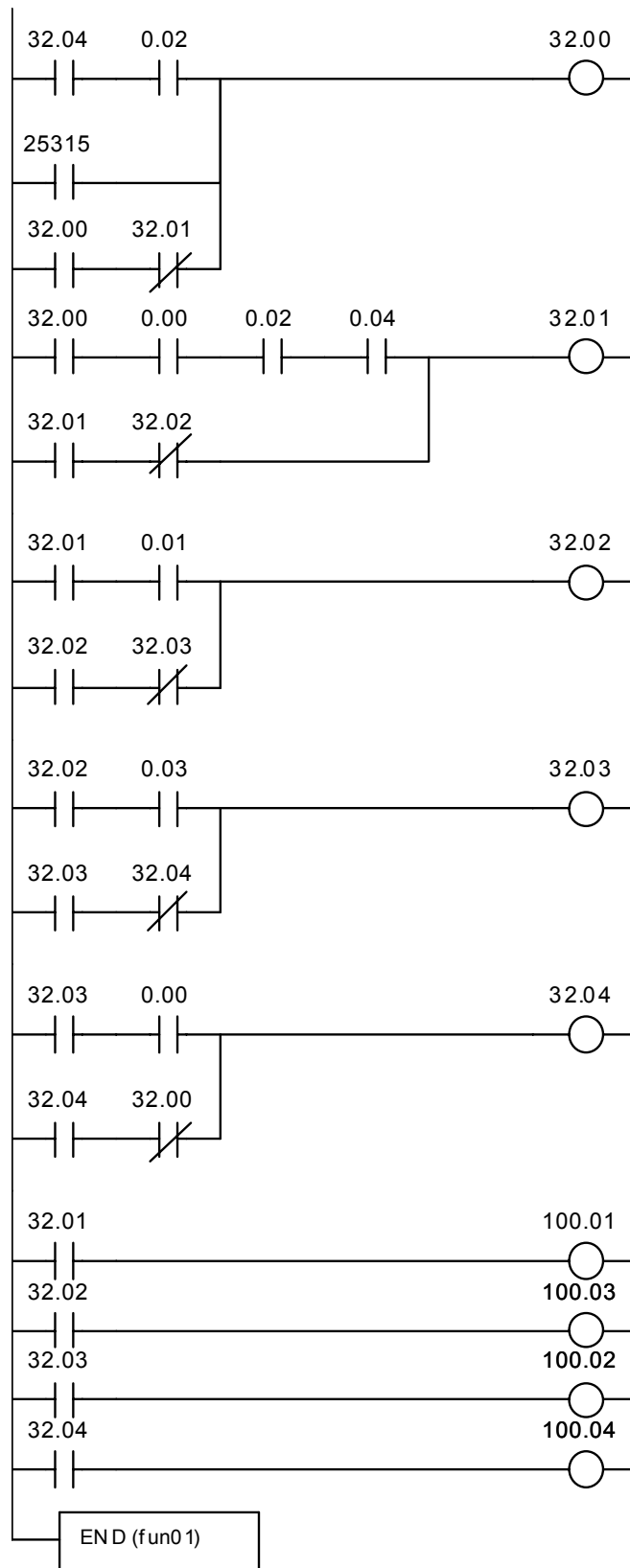
Kontaktuen eskema eratzean, ekuazio logikoak hartzen dira oinarritzat. Ekuazio logikoak, ordea, bi modutan ebatzi ditugu lehentasunaren arabera. Beraz, haietako bat aukeratu beharko da kontaktu-eskema eraikitzeko.

Hala ere, kasu honetan, bi aukeren eskemak osatu ditugu eta ezberdintasunak agerian gelditu dira.

► Kitzikapenik ezak lehenetsuna duen memoriarena

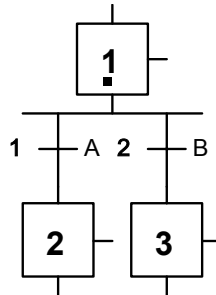


► Kitzikapenak lehenetsuna duen memoriarena.



4.7 Grafcet konplexuagoen ekuazio logikoak

Edo DIBERGENTZIA



- **Kitzikapenik ezak lehentasuna duen kasuan:**

$$F1 = (\text{Aurreko fasea} * \text{trantsizioa} + F1) * (F2 + F3) = (\text{Aurreko fasea} * \text{trantsizioa} + F1) * \overline{F2} * \overline{F3}$$

$$F2 = (F1 * A + F2) * \text{Hurrengo fasea ezeztatua}$$

$$F3 = (F1 * B + F3) * \text{Hurrengo fasea ezeztatua}$$

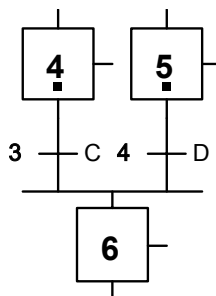
- **Kitzikapenak lehentasuna duen kasuan:**

$$F1 = \text{Aurreko fasea} * \text{trantsizioa} + F1 * (F2 + F3) = \text{Aurreko fasea} * \text{trantsizioa} + F1 * \overline{F2} * \overline{F3}$$

$$F2 = F1 * A + F2 * \text{Hurrengo fasea ezeztatua}$$

$$F3 = F1 * B + F3 * \text{Hurrengo fasea ezeztatua}$$

Edo KONBERGENTZIA



► **Kitzikapenik ezak lehentasuna duen kasuan:**

$$F4 = (\text{Aurreko fasea} * \text{trantsizioa} + F4) * F6$$

$$F5 = (\text{Aurreko fasea} * \text{trantsizioa} + F5) * F6$$

$$F6 = (F4 * C + F5 * D + F6) * \text{Hurrengo fasea ezeztatua}$$

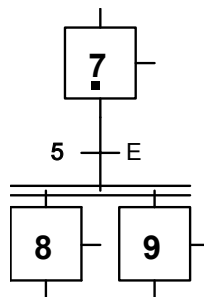
► **Kitzikapenak lehentasuna duen kasuan:**

$$F4 = \text{Aurreko fasea} * \text{trantsizioa} + F4 * F6$$

$$F5 = \text{Aurreko fasea} * \text{trantsizioa} + F5 * F6$$

$$F6 = F4 * C + F5 * D + F6 * \text{Hurrengo fasea ezeztatua}$$

■ **Eta DIBERGENTZIA**



► **Kitzikapenik ezak lehentasuna duen kasuan:**

$$F7 = (\text{Aurreko fasea} * \text{trantsizioa} + F7) * (F8 * F9) = (\text{Aurreko fasea} * \text{trantsizioa} + F7) * (F8 + F9)$$

$$F8 = (F7 * E + F8) * \text{Hurrengo fasea ezeztatua}$$

$$F9 = (F7 * E + F9) * \text{Hurrengo fasea ezeztatua}$$

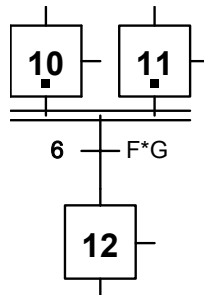
► **Kitzikapenak lehentasuna duen kasuan:**

$$F7 = \text{Aurreko fasea} * \text{trantsizioa} + \overline{F7 * (F8 * F9)} = \text{Aurreko fasea} * \text{trantsizioa} + F7 * (\overline{F8} + \overline{F9})$$

$$F8 = F7 * E + F8 * \text{Hurrengo fasea ezeztatua}$$

$$F9 = F7 * E + F9 * \text{Hurrengo fasea ezeztatua}$$

■ **Eta KONBERGENTZIA**



► **Kitzikapenik ezak lehentasuna duen kasuan:**

$$F10 = (\text{Aurreko fasea} * \text{trantsizioa} + F10) * \overline{F12}$$

$$F11 = (\text{Aurreko fasea} * \text{trantsizioa} + F11) * \overline{F12}$$

$$F12 = (F10 * F11 * F * G + F12) * \text{Hurrengo fasea ezeztatua}$$

► **Kitzikapenak lehentasuna duen kasuan:**

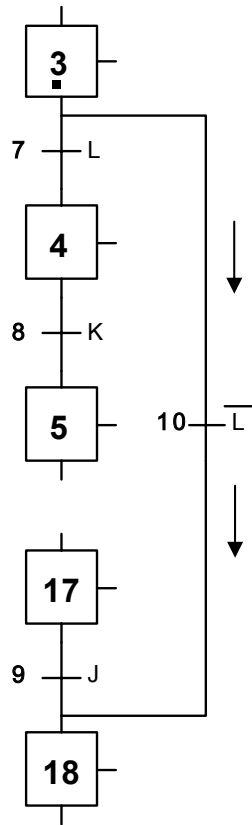
$$F10 = \text{Aurreko fasea} * \text{trantsizioa} + F10 * \overline{F12}$$

$$F11 = \text{Aurreko fasea} * \text{trantsizioa} + F11 * \overline{F12}$$

$$F12 = F10 * F11 * F * G + F12 * \text{Hurrengo fasea ezeztatua}$$

■ Zenbait faseko jauzi baldintzatuak

JAUZI BALDINTZATUA:



► Kitzikapenik ezak lehentasuna duen kasuan:

$$F3 = (\text{Aurreko fasea} * \text{trantsizioa} + F3) * (F4 + F18) = (\text{Aurreko fasea} * \text{trantsizioa} + F3) * (F4 * F18)$$

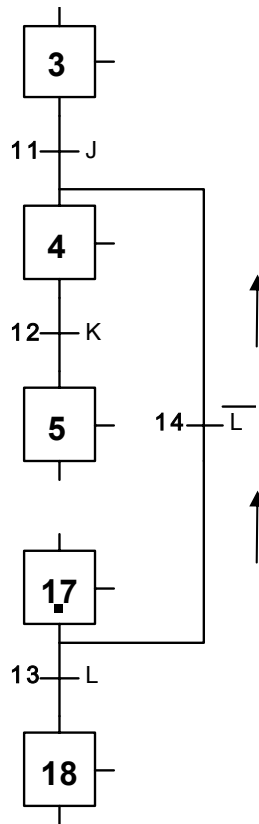
$$F18 = (F3 * L + F17 * J + F18) * \text{Hurrengo fasea ezeztatua}$$

► Kitzikapenak lehentasuna duen kasuan:

$$F3 = \text{Aurreko fasea} * \text{trantsizioa} + F3 * (F4 + F18) = \text{Aurreko fasea} * \text{trantsizioa} + F3 * (F4 * F18)$$

$$F18 = F3 * L + F17 * J + F18 * \text{Hurrengo fasea ezeztatua}$$

ERREPIKAPEN BALDINTZATUA:



- Kitzikapenik ezak lehentasuna duen kasuan:

$$F4 = (F3 \cdot J + F17 \cdot L + F4) \cdot F5$$

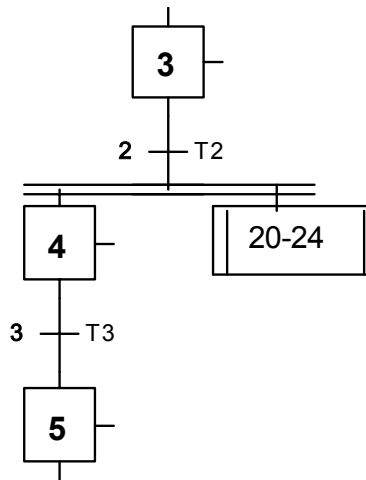
$$F17 = (\text{Aurreko fasea} \cdot \text{trantsizioa} + F17) \cdot (F4 + F18) = (\text{Aurreko fasea} \cdot \text{trantsizioa} + F17) \cdot (F4 \cdot F18)$$

- Kitzikapenak lehentasuna duen kasuan:

$$F4 = F3 \cdot J + F17 \cdot L + F4 \cdot F5$$

$$F17 = \text{Aurreko fasea} \cdot \text{trantsizioa} + F17 \cdot (F4 + F18) = \text{Aurreko fasea} \cdot \text{trantsizioa} + F17 \cdot (F4 \cdot F18)$$

■ Sekuentzia errepikatua.



► Kitzikapenik ezak lehentasuna duen kasuan:

$$F_{20} = (F_3 \cdot T_2 + F_{20}) \cdot \overline{F_{21}}$$

.
.

.

$$F_{24} = (F_{23} \cdot T_{23} + F_{24}) \cdot \text{Hurrengo fasea ezeztatua}$$

► Kitzikapenak lehentasuna duen kasuan:

$$F_{20} = F_3 \cdot T_2 + F_{20} \cdot \overline{F_{21}}$$

.
.

.

$$F_{24} = F_{23} \cdot T_{23} + F_{24} \cdot \text{Hurrengo fasea ezeztatua}$$

4.8 Egin beharreko ariketak

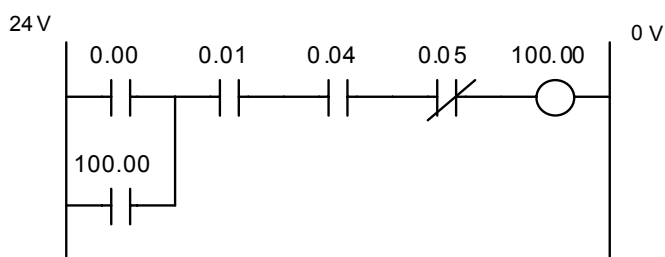
Egin itzazu ariketa hauek: 5.7.1, 5.7.2, 5.7.3, 5.7.4, 5.7.5, 5.7.6, 5.7.7 eta 5.7.8.

5 ARIKETAK

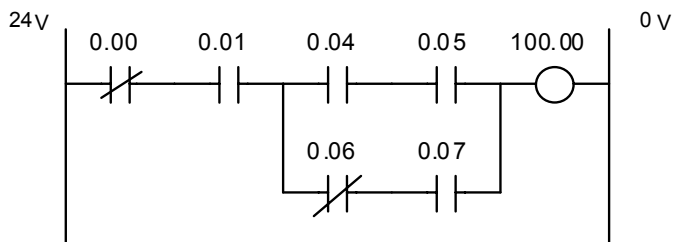
5.1 Oinarrizko ariketak

Kontaktu-eskema hauetan oinarrituz, lor ezazu programa lengoia mnemonikoan:

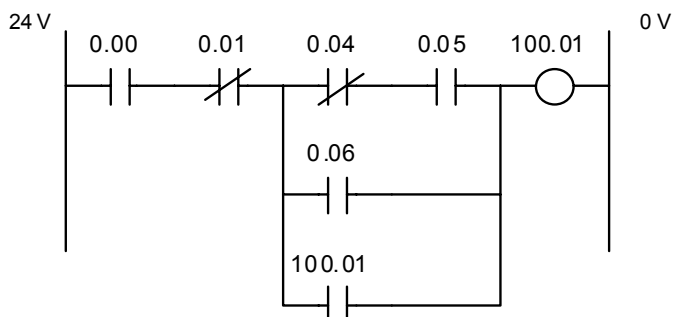
1. ariketa.



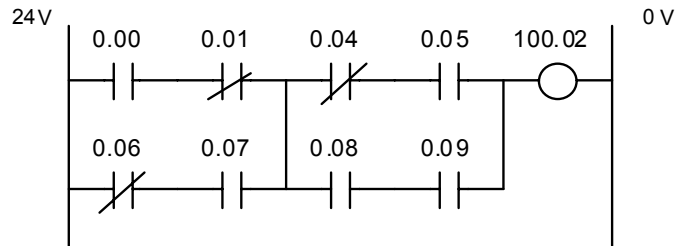
2. ariketa.



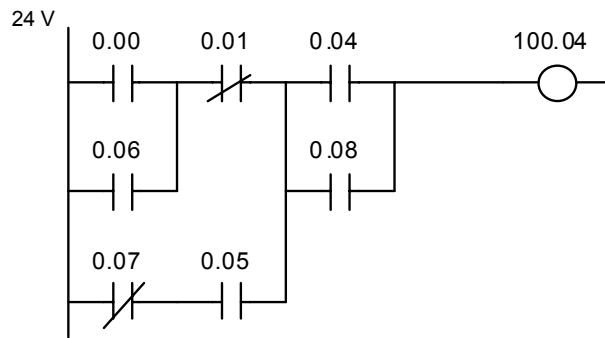
3. ariketa.



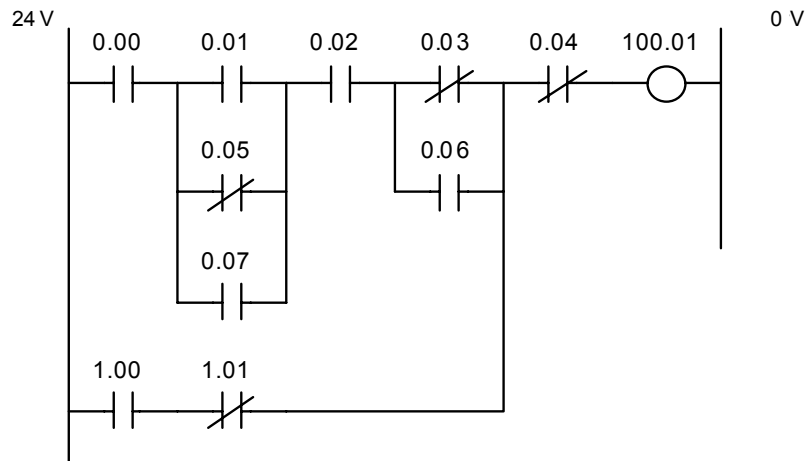
4. ariketa.



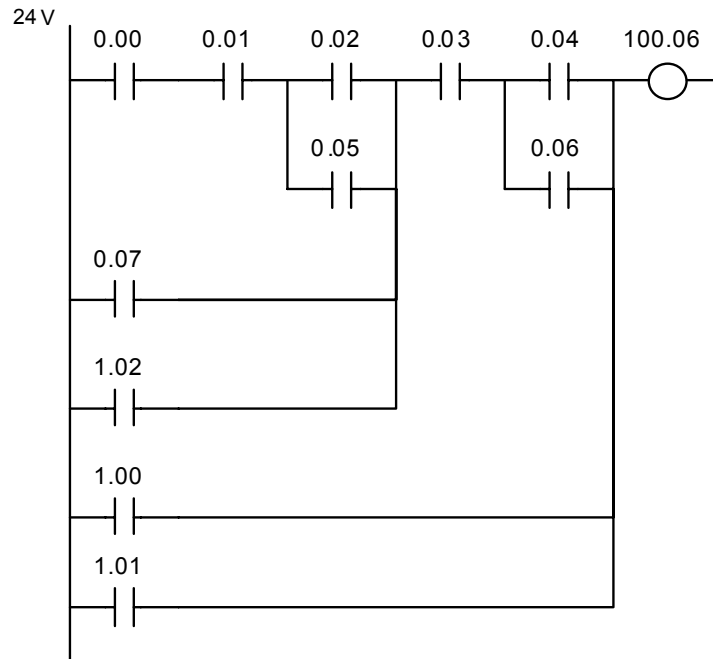
5. ariketa.



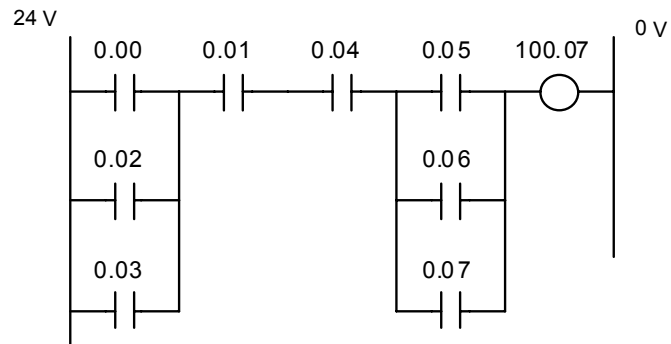
6. ariketa.



7. ariketa.



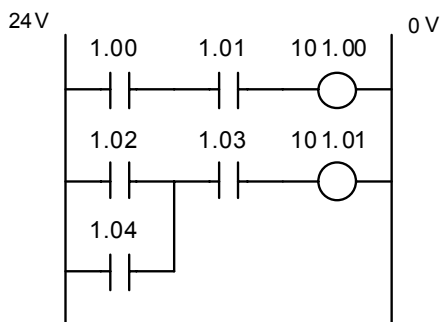
8. ariketa.



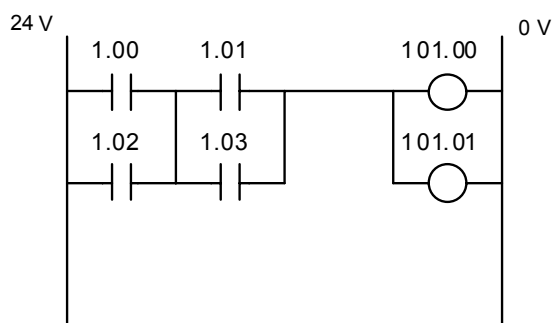
5.2 Zenbait irteerako eskemei buruzko ariketak

Kontaktu-eskema hauetan oinarrituz, lor ezazu programa lengoia mnemonikoan:

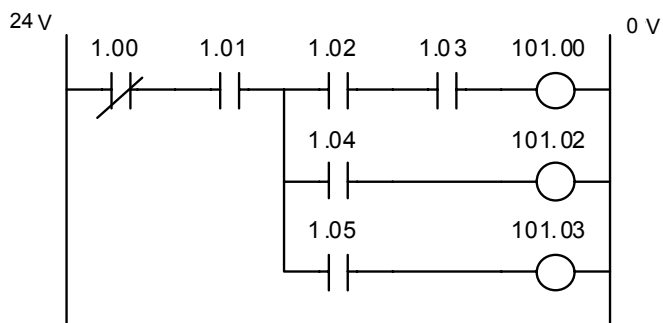
1. ariketa.



2. ariketa.



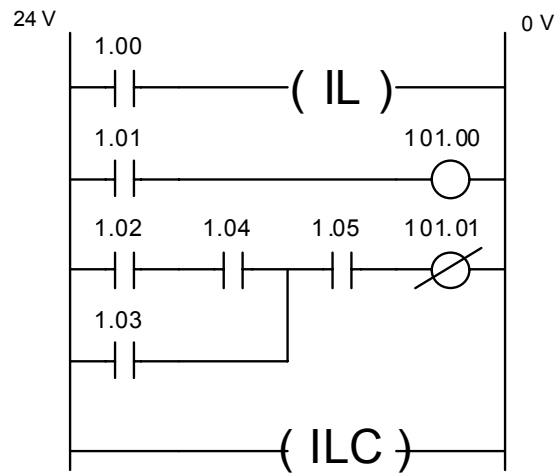
3. ariketa.



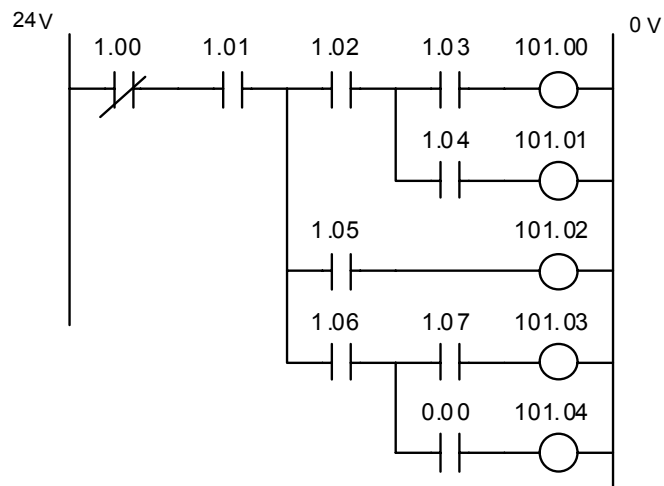
5.3 IL, ILC eta TR-ri buruzko ariketak

Kontaktu-eskema hauetan oinarrituz, lor ezazu programa lengoia mnemonikoan:

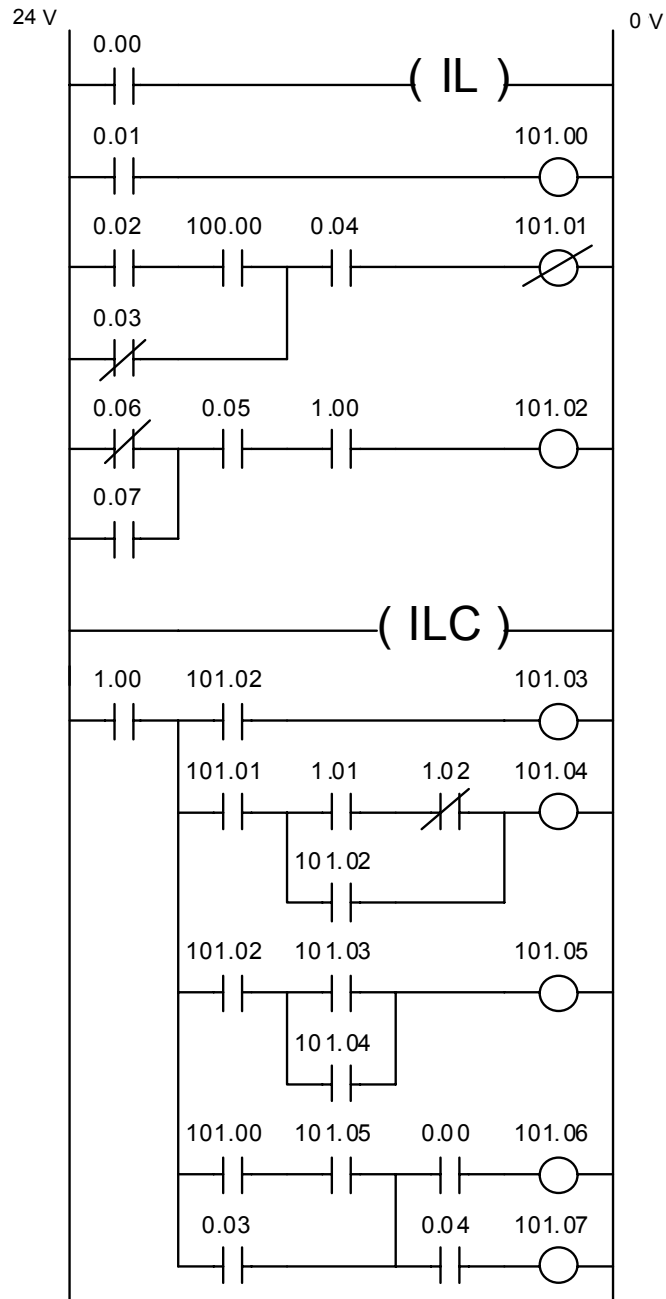
1. ariketa.



2. ariketa.



3. ariketa.

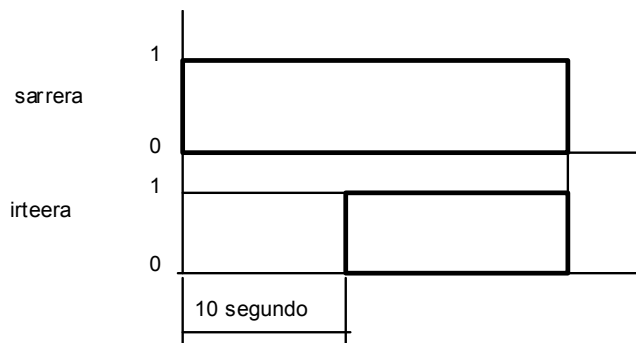


5.4 Tenporizadoreen oinarriko ariketak

1. ariketa.

Gela bateko argiak etengailua sakatu eta 10 segundora piztu behar dira. Egin beharrekoa:

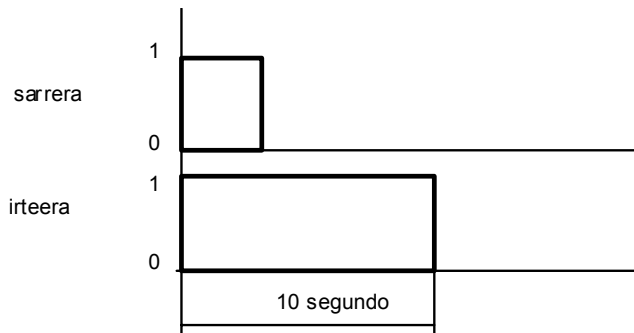
1. Sarrera eta irteera bakoitzari dagokion izendatzea egin.
2. Kontaktu-eskema osatu.
3. Programa idatzi lengoia mnemonikoa erabiliz.



2. ariketa.

Eraikin bateko atarian sartzean argia pizten da, eta 10 segundo igarotzean automatikoki itzaltzen da. Egin beharrekoa:

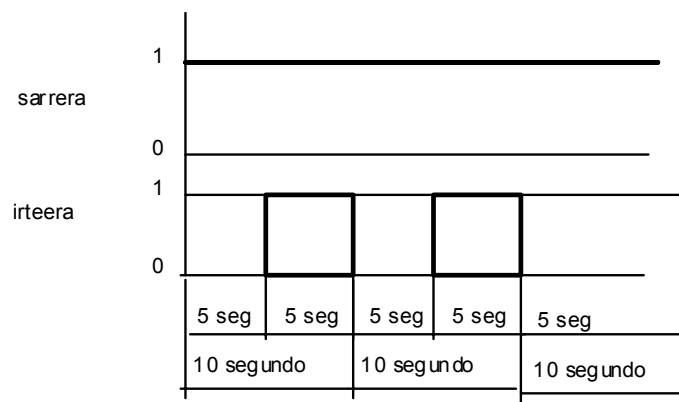
1. Sarrera eta irteera bakoitzari dagokion izendatzea egin.
2. Kontaktu-eskema osatu.
3. Programa idatzi lengoia mnemonikoa erabiliz.



3. ariketa.

Bonbilla baten aldizkakotasuna lortu behar da. Etengailua sakatzean, bonbillak 5 segundo eman behar ditu itzalita, eta ondoren, 5 segundo piztuta. Ziklo horrek etengabea izan behar du. Egin beharrekoa:

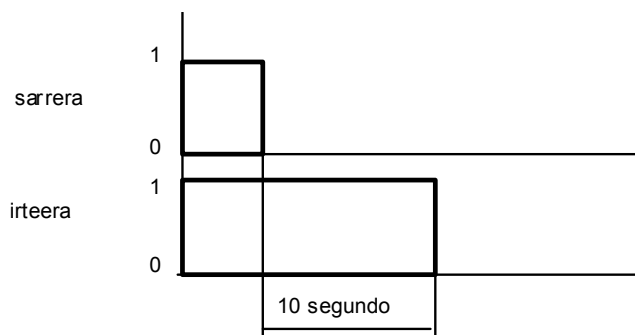
1. Sarrera eta irteera bakoitzari dagokion izendatzea egin.
2. kontaktu-eskema osatu.
3. Programa idatzi lengoia mnemonikoa erabiliz.



4. ariketa.

Gela bateko argiak sakagailu bati eragitean pizten dira. Sakagailua askatzean, argiak 10 segundoz egon behar du pizturik. Egin beharrekoa:

1. Sarrera eta irteera bakoitzari dagokion izendatzea egin.
2. kontaktu-eskema osatu.
3. Programa idatzi lengoia mnemonikoa erabiliz.



5.5 Kontadoreen oinarritzko ariketak

1. ariketa.

▶ Telefono mugikor baten dei kopuruaren kontrola

Telefono mugikor baten eguneko dei kopurua 4ra mugatu behar da.

Dei kopurua 4 edo handiagoa bada, LED edo bonbilla batek horren berri eman behar du. Egin beharrekoa:

1. Sarrera eta irteera bakoitzari dagokion izendatzea egin.
2. kontaktu-eskema osatu.
3. Programa idatzi lengoaia mnemonikoa erabiliz.

Oharra

Soilik 10 dei egin daitezke eguneko..

2. ariketa.

▶ Futbol-zelai bateko ikusle kopuruaren kontrola

Futbol-zelai batera 18.000 ikusle joan ohi dira. Azken jardunaldiko ikusle kopurua 18.000 baino handiagoa, txikiagoa edo berdina izan zen azertu behar da.

Lehenik, ikusleen kopurua kontrolatuko duen sistema aktibatu behar da. Horretarako, 0.01 eten-gailua sakatu beharra dago.

Gero, norbaitek sarrerako hesia igarotzen badu, ikusle bat batuko zaio ikusle kopuruari. Sarrerako hesia 0.02 kontaktuarekin ordezkatu dugu. Aldiz, norbaitek irteerako hesia igarotzen badu, ikusle bat kenduko zaio ikusle kopuruari. Irteerako hesia 0.03 kontaktuarekin ordezkatu dugu.

Gainera, ikusleen kopurua zeroan jartzeko, 0.04 sakagailuari eragin beharko zaio

Ondorioz, 18.000 ikusle baino gutxiago badaude, 100.00 bonbilla piztu behar da. Ikusle kopurua, berriz, 18.000 edo 18.000 baino handiagoa bada, 100.01 bonbilla piztuko da.

Egin beharrekoa:

1. Sarrera eta irteera bakoitzari dagokion izendatzea egin.
2. kontaktu-eskema osatu.
3. Programa idatzi lengoia mnemonikoa erabiliz.

Oharra

Futbol-zelaiaren edukiera 29.000 ikuslekoa da. Programa automatikoki sartzean, komenigarria litzateke 18.000 zenbakia 18 zenbakiarekin ordezkatzeko eta 29.000, berriz, 29 zenbakiarekin.

Izan ere, horrela egingo ez bagenu, programaren funtzionamendu egokia baieztatzea oso luzea izango litzateke.

5.6 Tenporizadoreen eta kontadoreen araketak

1. ariketa.

► Berogailu baten kontrola

Berogailua pizteko, sakagailu bati eragin behar zaio. Sakagailuari eragitean, berogailua ordubetez arituko da eten gabe lanean gelditu arte.

Larrialdietan, NI motako sakagailu bat erabili beharko da berogailua gelditzeko.

Egin beharrekoa:

1. Sarrera eta irteera bakoitzari dagokion izendatzea egin.
2. kontaktu-eskema osatu.
3. Programa idatzi lengoia mnemonikoa erabiliz.

2. ariketa.

► Eskailera mekaniko baten kontrola

Eskailera mekanikoak erabilgarri egon behar du eskailera konektatzeko sakagailuari eragin eta gero. Eskailera, motorra, alegia, mugitzen hasiko da kontrol-panelean dagoen *eskailera erabilgarria* izeneko LEDa pizturik dagoenean eta norbaitek zelula fotoelektrikoa eteten duenean. Foto-zelula hori eteten den bakoitzean, eskailerak 40 segundoz funtzionatuko du.

Kontrol-sistema guztiaren deskonexioa hiru kasu hauetan gertatuko da:

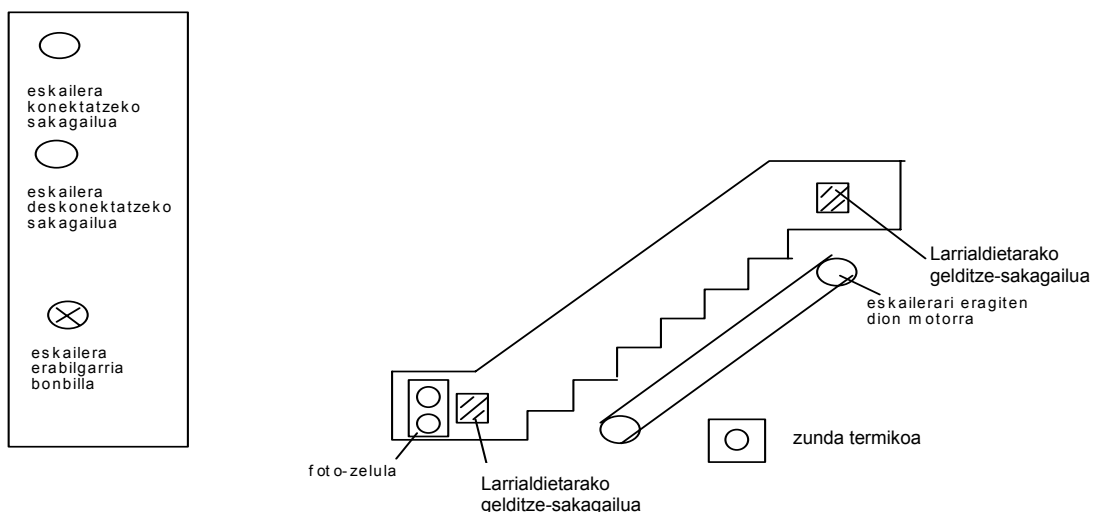
- ✓ Eskailera deskonektatzeko sakagailuari eragiten bazaio
- ✓ Zunda termikoa aktibatu bada
- ✓ Larrialdietarako gelditze-sakagailuari eragiten bazaio

Segurtasun-araudia aplikatuz, eskailera deskonektatzeko sakagailuak, zunda termikoak eta larrialdietarako gelditze-sakagailuak NI motako kontaktuak izan beharko dute.

Egin beharrekoa:

1. Sarrera eta irteera bakoitzari dagokion izendatzea egin.
2. Kontaktu-eskema osatu.
3. Programa idatzi lengoaia mnemonikoa erabiliz.

AGINTE-ETA KONTROL-PANELA



3. ariketa

▶ Aparkaleku baten kontrola

Helburua:

Aparkalekuaren egoera kontrolatu behar da une oro. Horretarako, bonbilla edo LED batzuen laguntza izango dugu.

Aparkalekuaren ezaugarriak:

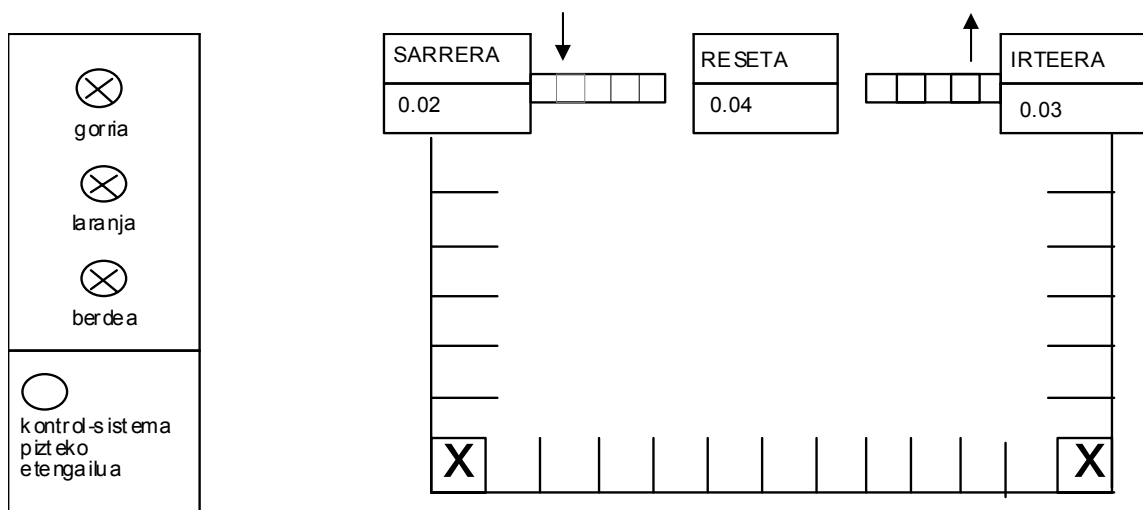
- ✓ 50 autorentzako lekua
- ✓ 50 auto badaude, led gorria pizten da
- ✓ Autoen kopurua 45 eta 50 bitartekoa bada, LED laranja pizten da

Funtzionamendua:

Aparkalekua kontrolatuko duen sistema pizteko, beharrezkoa izango da 0.07 etengailuari eragitea.

Egin beharrekoa:

1. Sarrera eta irteera bakoitzari dagokion izendatzea egin.
2. Kontaktu-eskema osatu.
3. Programa idatzi lengoia mnemonikoa erabiliz.



4. ariketa

▶ Auto baten karrozeriaren margotze automatikoaren kontrola

Margotze automatikoko instalazioa 1.00 sakagailuari eraginez pizten da. Instalazioa gelditzeko, berriz, 1.01 sakagailuari eragin behar zaio.

Instalazioa martxan dagoenean, 100.00 bonbilla piztuta izaten da.

Karrozeria batek 1.04 foto-zelula eteten duenean, 5 segundo igaro eta 100.01 atea itxi egiten dira.

Margotze-faseak, atea ixten direnetik berriz irekitzen diren arte, 20 segundoko iraupena du.

Atea irekitzen direnean, 1.05 foto-zelulak karrozeria detektatzen du, margotutzat jo eta zenbatu egiten du.

Autoaren karrozeria ez badago behar bezain margotua, langile batek 1.06 sakagailuari eragin, eta 1 kenduko zaio ordura arteko ondo pintatutako karrozerien kopuruari.

Lan-jardunaren ostean ondo margotutako karrozerien kopurua 5 baino txikiagoa bada, 100.02 bonbilla piztuko da.

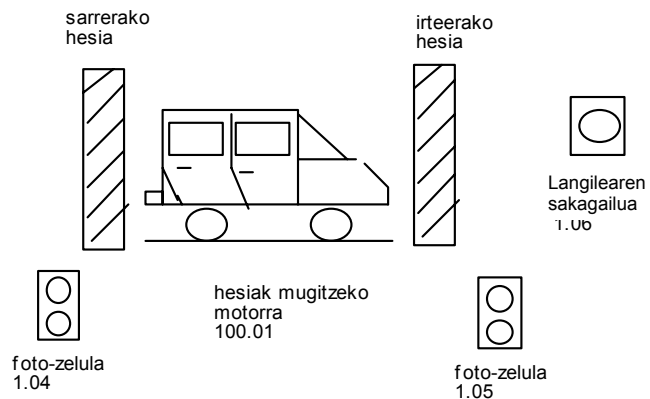
Lan-jardunaren ostean ondo margotutako karrozerien kopurua 5 edo 5 baino handiagoa bada, 100.03 bonbilla piztuko da.

Karrozeria kopurua reseteatzeko sakagailua 1.07 izango da.

Egin beharrekoa:

1. Sarrera eta irteera bakoitzari dagokion izendatzea egin.
2. kontaktu-eskema osatu.
3. Programa idatzi lengoia mnemonikoa erabiliz.

Instalazioa martxan jartzeko sakagailua 1.00	5 auto baino gutxiago 100.02
Instalazioa gelditze sakagailua 1.01	5 auto edo gehiago 100.03
Instalazioa martxan 100.00	



5. ariketa.

► Irteera baliogabeen kontrola

Munduko txapelketako 100 m-ko lasterketan 3 korrikalariren irteerak kontrolatu behar dira. Korrikalari bakoitzak kontadore bana du egiten dituen irteera baliogabeak zenbatzeko.

Funtzionamendua:

- ✓ Irteerak kontrolatzeko sistema 1.00 sakagailuari eragitean pizten da eta 1.01 sakagailuari eragitean itzali.
- ✓ Sistema piztua badago, hori adierazten duen 100.00 LEDa piztuko da.
- ✓ Semaforo batek adierazten du irteera noiz egin behar den. Hasieran, semaforoak 3 segundo ematen ditu gorri, laranja igaro baino lehen. Laranja 2 segundo emango ditu, eta ondoren, argia berdera igarotzen da, eta aldi berean txirrin baten soinua entzungo da, korrikalariak irten daitezen.
- ✓ Korrikalari bakoitzak sentsore bana du irteeran. Korrikalaria irten aurretik semaforoa berde ez badago edo txirrinaren soinua entzun ez bada, sentsoreak 1 batuko du irteera baliogabeak zenbatzen dituen kontadorean.
- ✓ Irteera baliogabeen kopurua 2 denean, korrikalariaren kalean hori adierazten duen argi bat piztuko da. Ondorioz, korrikalaria kanporatu egingo dute.

Egin beharrekoa:

1. Sarrera eta irteera bakoitzari dagokion izendatzea egin.
2. kontaktu-eskema osatu.
3. Programa idatzi lengoia mnemonikoa erabiliz.

5.7 Sistema sekuentzialen ariketak (*grafceta* erabiliz egin beharreko ariketak)

1. ariketa.

▶ **Kaxa-banaketa**

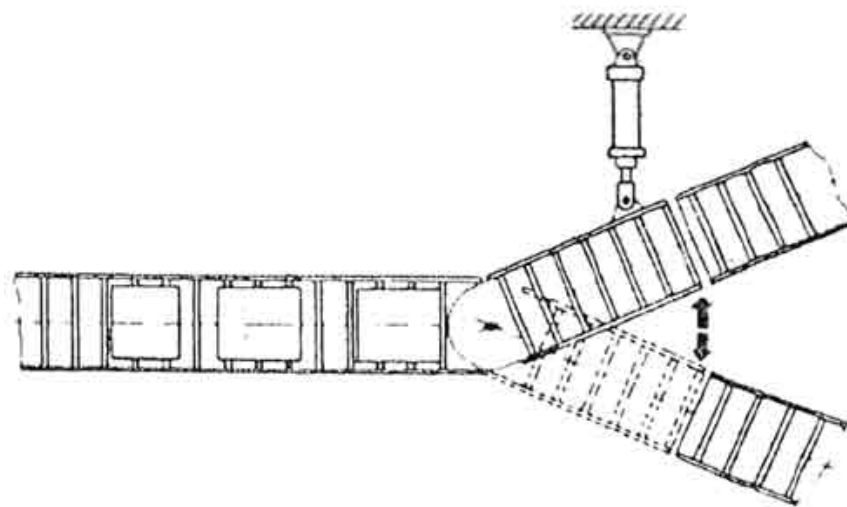
Sakagailu baten bitartez arraboldun zinta birarazi behar da. Zintak biratzen jarraitzeko, ez da beharrezkoa izango sakagailua zanpatuta mantentzea; une batez sakagailuari eraginda, zintak mugimendu osoa egin behar du. Zintak posizio berri hori mantenduko du bigarren sakagailu bati eragin arte.

Oharrak

1. oharra: sistemak funtziona dezan, ezer baino lehen, beharrezkoa izango da 100.00 LEDa piztuta egotea. LED hori 1.00 sakagailuari eraginez pizten da, eta 1.01 etengailuari eraginez itzali.
2. oharra: ezin da ibiltarte-mikroetengailurik erabili.

Egin beharrekoa:

1. Eskema elektro-pneumatikoa marraztu, dagozkion letra eta ikur guztiekin.
2. I. mailako *grafceta* eraiki.
3. Programa idatzi lengoia mnemonikoa erabiliz.
4. II. mailako *grafceta* eraiki.
5. Ekuazio logikoak ebatzi.
6. kontaktu-eskema osatu.



2. ariketa.

► Ate eta leihoen automatizazioa

Itsasontzi baten sotoko ate eta leihoak automatizatu behar dira, efektu bikoitzeko bi zilindro eta efektu bakuneko zilindro bat erabiliz. Ate eta leiho horiek sotoaren barnealdetik, sotoaren kanpoaldetik eta *aginte-zubitik* zabaldu eta itxi ahal izango dira. Horretarako, horietako leku bakoitzean bi sakagailu egongo dira; bata, atea eta leihoak zabaltzeko, eta bestea, ixteko.

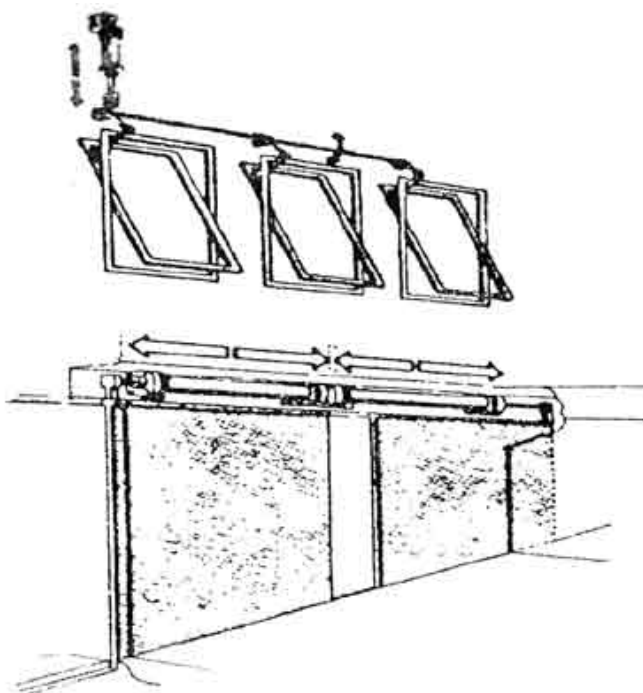
Oharrak

1. oharra: ezer baino lehen, sistemak funtziona dezan, beharrezkoa izango da 100.00 LEDa piztuta egotea. LED hori 1.00 sakagailuari eraginez pizten da, eta 1.01 etengailuari eraginez, itzali.

2. oharra: ezin da ibiltarte-mikroetengailurik erabili.

Egin beharrekoa:

1. Eskema elektro-pneumatikoa marraztu, dagozkion letra eta ikur guztiekin.
2. I. mailako *grafceta* eraiki.
3. Sarrera eta irteera bakoitzari dagokion izendatzea osatu.
4. II. mailako *grafceta* eraiki.
5. Ekuazio logikoak ebatzi.
6. kontaktu-eskema osatu.



■ 3. ariketa.

► **Kaxa-banaketa**

Egin ezazu 5.7.1. ariketa, baina, orainoan, erabili ibiltarte-mikroetengailuak.

■ 4. ariketa.

► **Garraio-mailaren aldaketa**

Arrabol multzo batean garraiatzen diren paketeak zilindro pneumatiko batek igo eta beste batek bultzatzen ditu, garraio-fasearen beste maila batera eramateko. B zilindroaren zurtoina ez da sartuko A zilindroaren zurtoina hasierako posiziora itzuli arte.

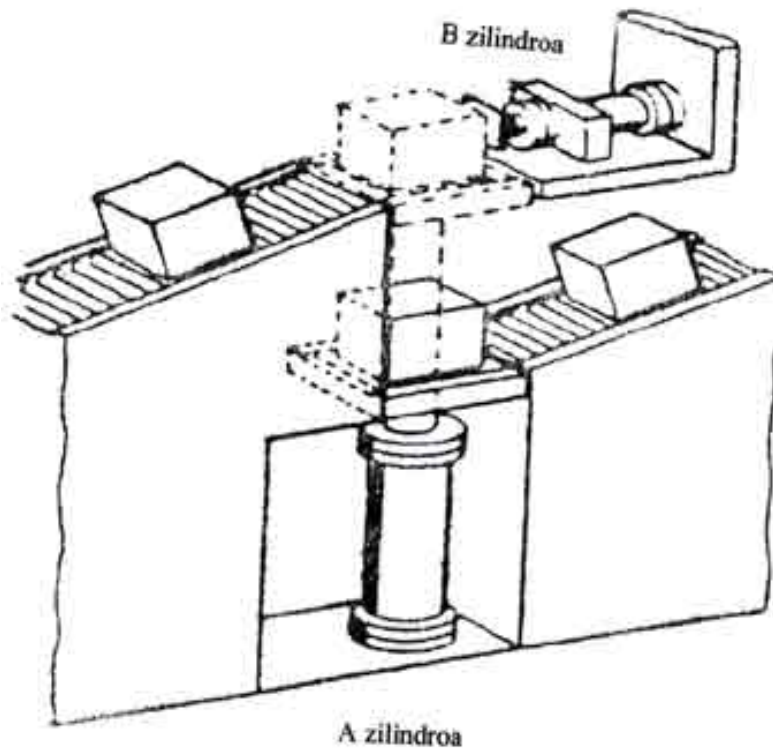
Mugimendu-sekuentzia hasteko, 1.00 sakagailuari eragin beharko zaio, eta gelditzeko, berriz, 1.01 sakagailuari.

Oharra

Ibiltarte-mikroetengailuak erabili behar dira.

Egin beharrekoa:

1. Eskema elektro-pneumatikoa marraztu, dagozkion letra eta ikur guztiekin.
2. I. mailako *grafceta* eraiki.
3. Sarrera eta irteera bakoitzari dagokion izendatzea osatu.
4. II. mailako *grafceta* eraiki.
5. Ekuazio logikoak ebatzi.
6. Kontaktu-eskema osatu.



6. ariketa.

► **Profil berezien estanzazioa**

Estanzazio-gailua automatikoki elikatu behar da profilez. Sakagailu bati eragitean, A zilindroaren zurtoinak profila estanzatuko du; ondoren, B zilindroaren zurtoinak gauza bera egingo du, eta azkenik, C zilindroaren zurtoinak. Zurtoin bakoitza bere hasierako posizioa itzuliko da egin beharreko lana egitean. Sekuentzia, beraz, hau izango da:

A+, A-, B+, B-, C+, C-.

Ziklo automatiko hori gelditzeko, bi aukera hauek eduki behar ditugu:

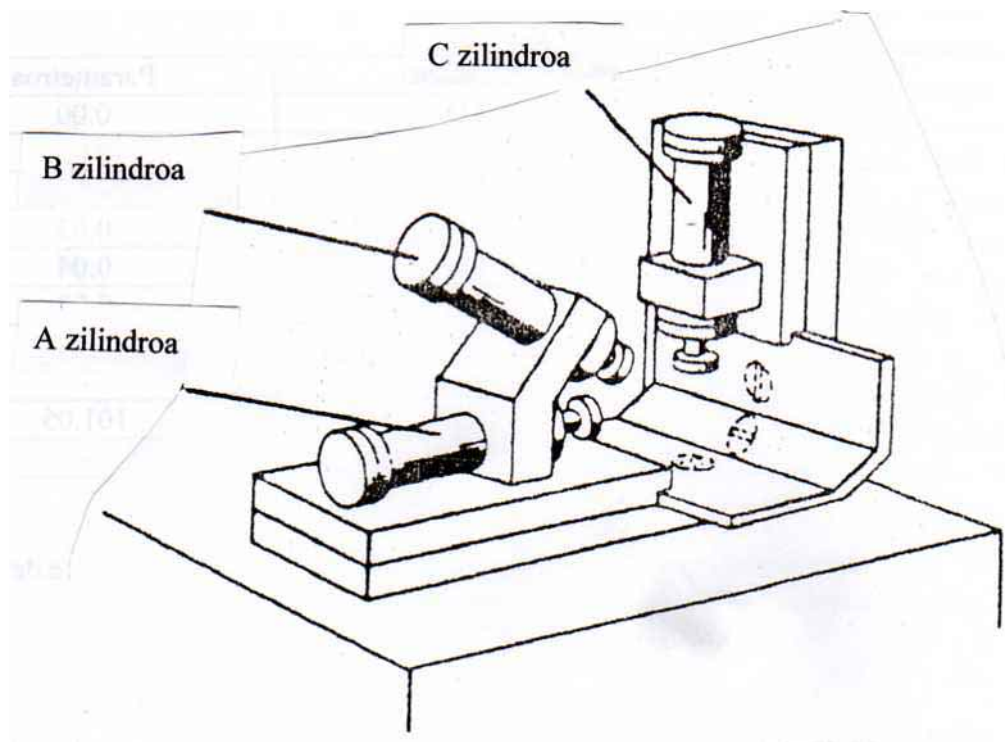
- ✓ 1.01 sakagailuari eragitean, zikloa gelditu egingo da, baina mugimendu-sekuentzia egin ostean.
- ✓ 1.08 sakagailuari eragiten zaion unean, zikloa gelditu egingo da, eta kanpoan dagoen zurtoina zilindro barrura itzuliko da. Sakagailu hori larrialdietarako erabiliko da.

Oharra

Ibiltarte-mikroetengailuak erabili behar dira.

Egin beharrekoa:

1. Eskema elektro-pneumatikoa marraztu, dagozkion letra eta ikur guztiekin.
2. I. mailako *grafceta* eraiki.
3. Sarrera eta irteera bakoitzari dagokion izendatzea osatu.
4. II. mailako *grafceta* eraiki.
5. Ekuazio logikoak ebatzi.
6. Kontaktu-eskema osatu.



7. ariketa.

► Profil berezien estapanazioa

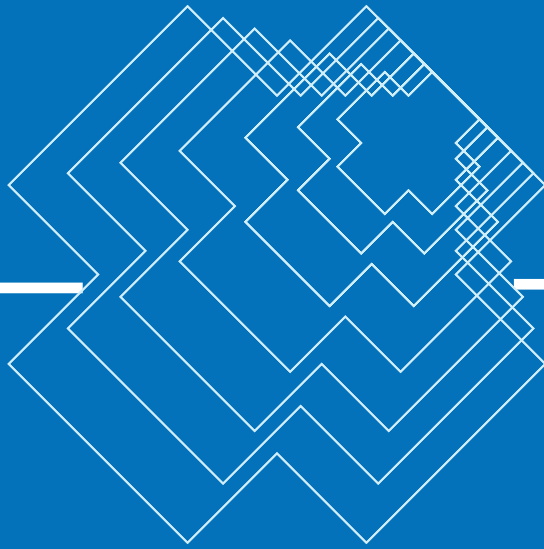
Egin ezazu 5.7.6. ariketa, baina, orainoan, mugimendu-sekuentzia honi jarraituz:

A+, A-, 2 segundo itxaron, B+, B-, 2 segundo itxaron, C+, C-, 2 segundo itxaron.

8. ariketa.

► Profil berezien estapanazioa

Egin itzazu beharrezko aldaketak 5.7.7. ariketako kontaktu-eskeman egin diren zikloen kopurua 5 baino txikiagoa denean 100.08 LEDa pitz dadin, eta zikloen kopurua 5 edo 5 baino handiagoa denean 100.09 LEDa pitz dadin.



LANBIDE
EKIMENA

