

Seria MELSEC FX

Automate programabile

Manual introductiv

**FX1S, FX1N,
FX2N, FX2NC,
FX3U**

Despre acest manual

Textele, imaginile, diagramele și exemplele din acest manual sunt oferite doar în scop informativ. Acestea sunt destinate ca suport pentru explicarea instalării, operării, programării și utilizării automatelor programabile din seriile MELSEC FX1S, FX1N, FX2N,FX2NC și FX3U.

Dacă aveți întrebări privind instalarea și operarea oricăruia dintre produsele descrise în acest manual, vă rugăm contactați reprezentantul dumneavoastră de vânzări local sau distribuitorul local (a se vedea coperta din spate).

Puteți găsi cele mai recente informații și răspunsuri la întrebările frecvente pe site-ul nostru, la adresa www.mitsubishi-automation.com.

MITSUBISHI ELECTRIC EUROPE BV își rezervă dreptul de a efectua modificări în acest manual sau în specificațiile tehnice ale produselor sale în orice moment, fără nici o înștiințare prealabilă.

**Manual introductiv, pentru automatele programabile din seria MELSEC FX
FX1S, FX1N, FX2N, FX2NC și FX3U
Art. nr.: XXXXXXX**

Versiunea			Revizui/Compleări/Rectificări
A	01/2006	pdp-tr	Prima ediție
B	01/2007	pdp-dk	Adăugarea capitolului 7 A fost luat în considerare domeniului extins de produse pentru unitățile de bază din seria FX3U, din capitolele 2.3 și 2.4.

Instrucțiuni privind siguranța

A se utiliza doar de către personal calificat

Acest manual este destinat pentru uzul exclusiv al personalului instruit și calificat în domeniul electric, familiarizat cu toate standardele relevante privind siguranța tehnologică. Toate lucrările cu hardware-ul descris aici, inclusiv proiectarea sistemului, instalarea, configurarea, mentenanța, service-ul și testarea echipamentelor, pot fi efectuate doar de către personal cu calificare în domeniul electric și cu experiență, care să dețină diplome aprobate și să fie familiarizați cu toate standardele și regulamentele aplicabile privind siguranța tehnologică în automatizări. Orice operații sau modificări ale hardware-ului și/sau software-ului produselor noastre, ce nu sunt descrise în acest manual, pot fi efectuate doar de către personal autorizat al firmei Mitsubishi Electric.

Utilizarea corespunzătoare a produselor

Automatele programabile din seriile FX1S, FX1N, FX2N, FX2NC și FX3U sunt destinate doar pentru aplicațiile specifice descrise explicit în acest manual. Trebuie respectate toate setările și toți parametri specificați în acest manual. Toate produsele descrise au fost proiectate, fabricate, testate și documentate, respectându-se strict standardele relevante de siguranță. Modificarea hardware-ului sau software-ului de către personal necalificat sau nerespectarea avertismentelor înscrise pe produse și specificate în acest manual pot duce la rănirea gravă a personalului și/sau la daune materiale. Automatele programabile din seriile FX1S, FX1N, FX2N, FX2NC și FX3U pot fi utilizate numai cu perifericele și echipamentul de extensie recomandate în mod explicit și aprobate de către Mitsubishi Electric.

Oricare alte utilizări sau aplicații ale produselor vor fi considerate necorespunzătoare.

Regulamente relevante de siguranță

Toate regulamentele relevante de siguranță și de prevenire a accidentelor pentru aplicația dumneavoastră specifică trebuie respectate la proiectarea sistemului și la instalarea, configurarea, mentenanța, service-ul și testarea acestor produse. În această privință, regulamentele de mai jos sunt deosebit de importante. Totuși, această listă nu se pretinde a fi exhaustivă; dumneavoastră însă sunteți responsabil să vă familiarizați și să vă conformați regulamentelor aplicabile locației dumneavoastră.

- Standarde VDE
 - VDE 0100
Regulamente pentru ridicarea de instalații de putere cu tensiuni sub 1000 V
 - VDE 0105
Operarea instalațiilor de putere
 - VDE 0113
Instalații electrice cu echipament electronic
 - VDE 0160
Echipament electronic de utilizat în instalații de putere
 - VDE 0550/0551
Regulamente pentru transformatoare
 - VDE 0700
Siguranța aparatelor electrice de uz casnic și a aplicațiilor similare
 - VDE 0860
Regulamente de siguranță pentru aplicații electronice cu alimentare de la rețea și pentru accesoriile acestora pentru uz casnic și aplicații similare.
- Regulamente de asigurare împotriva incendiilor

- Regulamente de prevenire a accidentelor
 - VBG Nr.4
Sisteme și echipament electric

Sisteme și echipament electric

În acest manual, avertismentele privind siguranța sunt identificate astfel:



PERICOL:

Nerespectarea avertismentelor de siguranță identificate prin acest simbol poate duce la riscuri de sănătate și de rănire pentru utilizator.



AVERTISMENT:

Nerespectarea avertismentelor de siguranță identificate prin acest simbol poate duce la pagube materiale sau la avariarea echipamentelor și a altor bunuri.

Informații generale și măsuri de precauție privind siguranța

Următoarele măsuri de precauție privind siguranța sunt destinate a fi un ghid generic la utilizarea sistemelor de automate programabile în combinație cu alte echipamente. Aceste instrucțiuni trebuie întotdeauna respectate la proiectarea, instalarea și operarea tuturor sistemelor de control.

**PERICOL:**

- **Respectați toate regulamentele de siguranță și de prevenire a accidentelor valabile pentru aplicația dumneavoastră specifică. Întrerupeți întotdeauna alimentarea de la rețea înainte de a efectua lucrări de instalare și de cablare pentru ansambluri, componente și dispozitive.**
- **Ansamblurile, componentele și dispozitivele trebuie întotdeauna instalate în carcase rezistente la șocuri, la care sunt montate capace adecvate și siguranțe sau întrerupători de circuit.**
- **În instalațiile clădirii trebuie să fie integrate dispozitive cu conectare permanentă la alimentarea de rețea, cu un întreruptor pentru toți polii și o siguranță adecvată.**
- **Verificați regulat cablurile de alimentare și liniile de conectare ale echipamentelor, pentru a stabili dacă au apărut întreruperi și defecte de izolație. Dacă este descoperit un defect la un cablu, deconectați imediat echipamentul și cablurile de la rețeaua de alimentare cu energie și înlocuiți cablurile defecte.**
- **Înainte de a utiliza echipamentul pentru prima oară, verificați dacă valorile pentru puterea nominală corespund valorilor rețelei de alimentare locale.**
- **Luați măsurile adecvate pentru a vă asigura că defecțiunile apărute la cabluri sau deteriorările de miez din circuitele de transmisie nu duc la stări nedefinite ale echipamentului.**
- **Este responsabilitatea dumneavoastră să luați toate măsurile necesare pentru a vă asigura că programele întrerupte de căderi de tensiune vor putea fi repornite în mod adecvat și în siguranță. Trebuie să vă asigurați de asemenea de faptul că nu pot apărea condiții de utilizare periculoase în timpul funcționării, nici chiar pentru perioade scurte de timp.**
- **Unitățile de OPRIRE ÎN CAZ DE URGENȚĂ (EMERGENCY OFF) standardizate conform EN 60204/ICE 204 și VDE 0113 trebuie să fie funcționale în orice moment și în toate modurile de operare ale automatului programabil. Funcția de resetare a unității de OPRIRE ÎN CAZ DE URGENȚĂ (EMERGENCY OFF) trebuie proiectată astfel încât să nu poată niciodată să ducă la o repornire necontrolată sau nedefinită.**
- **Trebuie să implementați măsuri de precauție hardware și software, pentru a împiedica apariția de stări nedefinite ale sistemului de control, cauzate de cablurile circuitelor de transmisie sau de deteriorări ale miezului.**
- **La utilizarea modulelor, asigurați-vă întotdeauna că toate specificațiile și cerințele mecanice sunt respectate cu strictețe.**

Cuprins

1	Introducere	
1.1	Despre acest manual.....	1-1
1.2	Informații suplimentare	1-1
2	Automate programabile	
2.1	Ce este un automat programabil?	2-1
2.2	Cum procesează automatele programabile programele	2-2
2.3	Seria MELSEC FX.....	2-4
2.4	Alegerea controlerului potrivit	2-5
2.5	Părți componente	2-6
2.5.1	Circuite de intrare și de ieșire	2-6
2.5.2	Alcătuirea unităților de bază MELSEC FX1S.....	2-6
2.5.3	Alcătuirea unităților de bază MELSEC FX1N	2-7
2.5.4	Alcătuirea unităților de bază MELSEC FX2N	2-7
2.5.5	Alcătuirea unităților de bază MELSEC FX2NC.....	2-8
2.5.6	Alcătuirea unităților de bază MELSEC FX3U	2-8
2.5.7	Glosar de componente ale automatelor programabile.....	2-9
3	Introducere în programare	
3.1	Structura unei instrucțiuni de program	3-1
3.2	Biți, octeți și cuvinte	3-2
3.3	Sisteme de numerație.....	3-2
3.4	Setul de instrucțiuni de bază	3-5
3.4.1	Operații logice inițiale.....	3-6
3.4.2	Utilizarea rezultatului unei operații logice.....	3-6
3.4.3	Utilizarea comutatorilor și senzorilor.....	3-8
3.4.4	Operații AND (ȘI).....	3-9
3.4.5	Operații OR.....	3-11
3.4.6	Instrucțiuni pentru conectarea blocurilor de operare.....	3-12
3.4.7	Operații cu execuție pe front	3-14
3.4.8	Setarea și resetarea dispozitivelor	3-15
3.4.9	Stocarea, citirea și ștergerea rezultatelor operațiilor	3-17
3.4.10	Generarea de impulsuri	3-18
3.4.11	Funcția de control principal (instrucțiunile MC și MCR)	3-19
3.4.12	Inversarea rezultatului unei operații	3-20

3.5	Nu uitați măsurile de siguranță!	3-21
3.6	Exemple aplicații PLC	3-23
3.6.1	Un sistem de alarmă	3-23
3.6.2	O ușă automată de tip rulou	3-28

4 Prezentarea detaliată a dispozitivelor

4.1	Intrări și ieșiri	4-1
4.2	Relee.....	4-3
4.2.1	Relee speciale	4-3
4.3	Temporizatoare	4-4
4.4	Contoare	4-7
4.5	Regiștri	4-9
4.5.1	Regiștri de date	4-9
4.5.2	Regiștri speciali	4-10
4.5.3	Regiștri de fișiere	4-11
4.6	Tehnici de programare pentru temporizatoare și contoare	4-11
4.6.1	Specificarea indirectă a valorilor de referință pentru temporizatoare și contoare....	4-11
4.6.2	Întârziere la oprire	4-14
4.6.3	Întârziere la activare și la dezactivare	4-15
4.6.4	Generatoare de semnal de ceas	4-16

5 Tehnici avansate de programare

5.1	Listă instrucțiuni aplicate.	5-1
5.1.1	Introducerea instrucțiunilor aplicate.....	5-6
5.2	Instrucțiuni pentru mutarea datelor	5-7
5.2.1	Mutarea valorilor individuale cu instrucțiunea MOV	5-7
5.2.2	Mutarea grupurilor de dispozitive bit	5-9
5.2.3	Mutarea blocurilor de date cu instrucțiunea BMOV.....	5-10
5.2.4	Copierea unui dispozitiv sursă în mai multe destinații (FMOV)	5-11
5.2.5	Schimbul de date cu modulele de funcții speciale.....	5-12
5.3	Instrucțiuni de comparare	5-15
5.3.1	Instrucțiunea CMP	5-15
5.3.2	Comparații în operațiile logice	5-17
5.4	Instrucțiuni matematice	5-20
5.4.1	Adunarea.....	5-21
5.4.2	Scăderea	5-22

5.4.3	Înmulțirea	5-23
5.4.4	Împărțirea	5-24
5.4.5	Combinarea instrucțiunilor matematice	5-25

6 Opțiuni de extensie

6.1	Introducere.....	6-1
6.2	Module disponibile	6-1
6.2.1	Module pentru suplimentarea numărului de intrări și ieșiri digitale	6-1
6.2.2	Module de intrare/ieșire analogice.....	6-1
6.2.3	Module de comunicații.....	6-2
6.2.4	Module de poziționare	6-2
6.2.5	Terminale de operare HMI.....	6-2

7 Procesarea valorilor analogice

7.1	Module analogice	7-1
7.1.1	Criterii pentru selecția modulelor analogice.....	7-3
7.2	Lista modulelor analogice.....	7-5

1 Introducere

1.1 Despre acest manual

Acest manual vă va ajuta să vă familiarizați cu utilizarea seriei MELSEC FX de automate programabile. Manualul este destinat utilizatorilor care nu au încă experiența programării automatelor programabile (PLC).

Programatorii care au deja experiență în programarea automatelor programabile de la alți fabricanți pot de asemenea să utilizeze acest manual, pentru a se familiariza cu seria MELSEC FX.

Simbolul "□" este utilizat ca înlocuitor, pentru identificarea altor automate programabile din aceeași familie. De exemplu, denumirea "FX1S-10□-□□" este utilizată pentru a desemna toate automatele programabile al căror nume începe cu FX1S-10, adică FX1S-10 MR-DS, FX1S-10 MR-ES/UL, FX1S-10 MT-DSS și FX1S-10MT-ESS/UL.

1.2 Informații suplimentare

Informații detaliate privind produsele individuale dintr-o anumită serie pot fi obținute din manualele de operare și de instalare ale modulelor individuale.

Pentru o prezentare generală a tuturor controlerelor din seria MELSEC FX, consultați Catalogul de serie MELSEC FX, art. nr. 167840. Acest catalog conține de asemenea informații privind opțiunile de extensie și accesoriile disponibile.

Pentru o introducere în utilizarea pachetului software pentru programare, consultați Manualul pentru începători FX Developer FX, art. nr. 166391.

Puteți găsi informații detaliate despre toate instrucțiunile de programare în Manualul de programare pentru seria MELSEC FX, art. nr. 132738 și în Manualul de programare pentru seria FX3U, art. nr. 168591.

Capacitățile de comunicație și opțiunile controlerelor MELSEC FX sunt documentate în detaliu în Manualul de comunicații, art. nr. 070143.

Toate manualele și cataloagele Mitsubishi pot fi descărcate gratuit de pe site-ul web al Mitsubishi, www.mitsubishi-automation.com.

2 Automate programabile

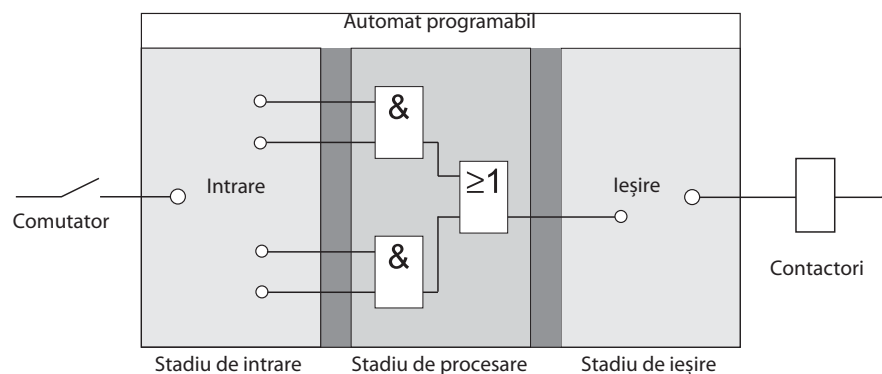
2.1 Ce este un automat programabil?

Spre deosebire de controlerele convenționale, ale căror funcții sunt determinate de cablarea fizică, funcțiile automatelor programabile (PLC) sunt definite de către un program.

Automatele programabile trebuie de asemenea să fie conectate la echipamente externe, prin cabluri, dar conținutul memoriei lor de program poate fi modificat în orice moment pentru ca programele să fie adaptate la diverse sarcini de control.

Automatele programabile primesc date, le procesează și transmit în afară rezultatele. Acest proces se desfășoară în trei stadii:

- un stadiu de intrare,
- un stadiu de procesare și
- un stadiu de ieșire



Stadiul de intrare

Stadiul de intrare transmite semnalele de control de la comutatori, butoane sau senzori către stadiul de procesare.

Semnalele provenite de la aceste componente sunt generate ca parte a procesului de control și sunt transmise în dispozitivele de intrare ca stări logice. Stadiul de intrare le transmite spre stadiul de procesare într-un format pre-procesat.

Stadiul de procesare

În stadiul de procesare, semnalele pre-procesate, venite din stadiul de intrare, sunt procesate și combinate cu ajutorul operațiilor logice și al altor funcții. Memoria de program a stadiului de procesare este integral programabilă. Ordinea de procesare poate fi schimbată în orice moment, prin modificarea sau înlocuirea programului stocat.

Stadiul de ieșire

Rezultatele procesării semnalelor de intrare de către program sunt transmise stadiului de ieșire, unde controlează elemente ce pot fi comutate, cum ar fi: contactori, lumini de semnalizare, ventile electromagnetice etc.

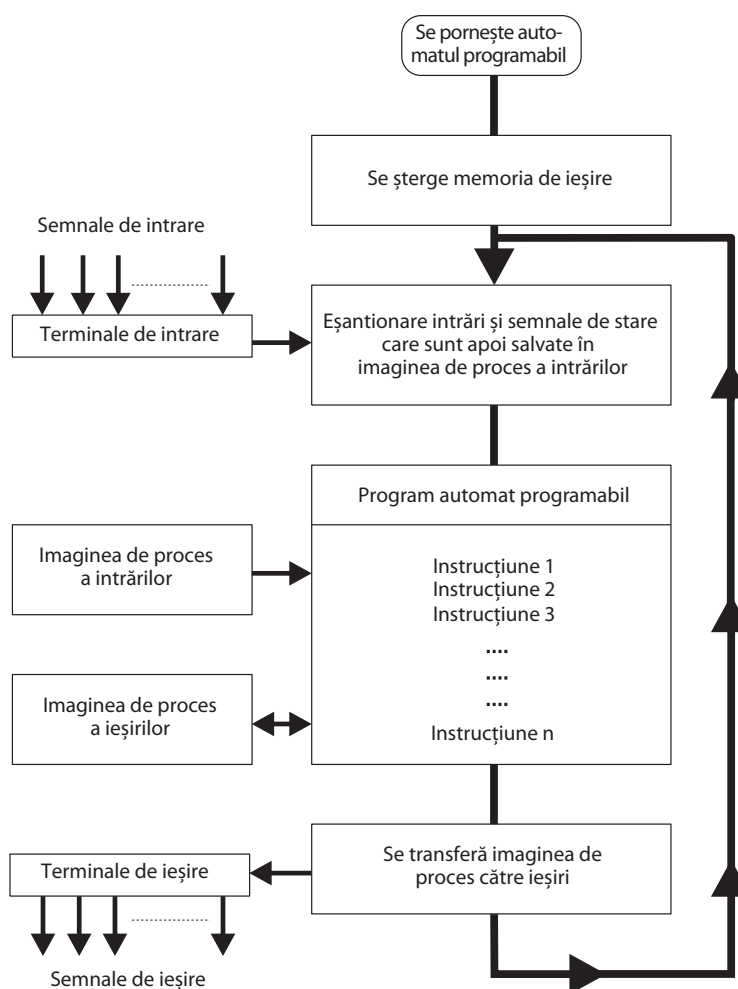
2.2 Cum procesează automatele programabile programele

Un automat programabil își efectuează sarcinile prin execuția unui program dezvoltat de obicei în exteriorul automatului și transferat ulterior în memoria de program a acestuia. Înainte de a începe să programați, este util să aveți noțiuni de bază privind modul în care automatele programabile procesează aceste programe.

Un program al unui automat programabil constă dintr-o secvență de instrucțiuni ce controlează funcțiile automatului. Automatul programabil execută aceste instrucțiuni de control secvențial, adică una după alta. Întreaga secvență de program este ciclică, ceea ce înseamnă că se va repeta într-o buclă continuă. Durata necesară unei singure repetiții de program este denumită durata sau perioada de ciclare a programului.

Procesarea imaginii de proces

Programul din automatul programabil nu este executat direct asupra intrărilor și ieșirilor, ci utilizează o "imagine de proces" a acestor intrări și ieșiri.



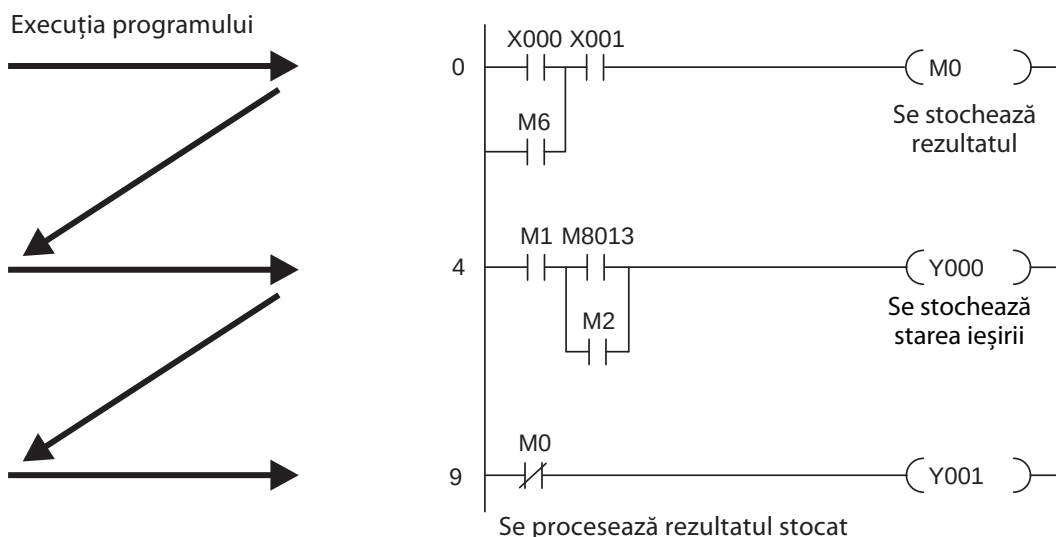
Imaginea de proces a intrărilor

La începutul fiecărui ciclu de program, sistemul interoghează stările semnalelor intrărilor și le stochează într-o memorie-tampon, creând o "imagine de proces" a intrărilor.

Procesarea programului

După secvența de procesare a intrărilor, automatul programabil accesează stările intrărilor stocate în imaginea de proces. Acest lucru înseamnă că orice modificări ulterioare ale stărilor de intrare vor fi înregistrate doar la **următorul** ciclu al programului !

Programul este executat de sus în jos, în ordinea în care au fost programate instrucțiunile. Rezultatele fiecăruia dintre pașii de program sunt stocate și pot fi utilizate în timpul ciclului de program curent.



Imaginea de proces a ieșirilor

Rezultatele operațiilor logice relevante pentru ieșiri sunt stocate într-o memorie-tampon de ieșire: aceasta este imaginea de proces a ieșirilor. Imaginea de proces a ieșirilor este stocată în memoria-tampon până la rescriere. După ce valorile au fost scrise în ieșiri, ciclul programului este repetat.

Diferențe între procesarea de semnal în automatele programabile și în controlere cablate

În controlerele cablate, programul este definit de elementele funcționale și conexiunile (cablurile) dintre ele. Toate operațiile de control sunt executate simultan (execuție în paralel). Orice modificare de stare a unui semnal de intrare va declanșa o modificare instantanee în starea semnalului corespondent de ieșire.

Într-un automat programabil, nu este posibil să se răspundă la modificările de stare ale semnalului de intrare decât în cursul următorului ciclu al programului, ce survine după apariția modificării. În prezent, acest dezavantaj este compensat în mare măsură de faptul că perioadele de scanare ale programelor sunt extrem de scurte. Durata unei perioade de scanare a unui program depinde de numărul și tipul de instrucțiuni executate.

2.3 Seria MELSEC FX

Micro automatele programabile compacte ale familiei MELSEC FX constituie baza dezvoltării de soluții pentru automatizări industriale de complexitate medie, cu un necesar de puncte pentru intrări/ieșiri de până la 256.

Cu excepția modelelor seriei FX1S, toate controlerele familiei FX pot fi prevăzute cu module de extensie pentru a putea răspunde tuturor cerințelor specifice ale aplicației.

Sunt de asemenea posibile conexiunile în rețea. Acest lucru face ca seria FX de controlere să poată comunica cu alte automate programabile și sisteme de control, precum și cu interfețe HMI (interfețe om-mașină și panouri de control). Sistemele de automat programabil pot fi integrate și în rețele MITSUBISHI, ca puncte de lucru locale, și ca puncte de lucru subordonate (de tip slave) în rețele deschise precum PROFIBUS/DP.

În plus, puteți de asemenea să creați rețele multidrop și peer-to-peer cu controlerele din seria MELSEC FX.

Modelele FX1N, FX2N și FX3U beneficiază de capacități de extensie modulare, ceea ce le face alegerea potrivită pentru aplicații complexe și sarcini ce necesită funcții speciale, cum ar fi conversii digital-analogic și viceversa, sau capacități de legare în rețea.

Toate controlerele acestor serii fac parte din familia MELSECFX și sunt perfect compatibile între ele.

Specificații	FX1S	FX1N	FX2N	FX2NC	FX3U
Nr. max. de puncte de intrare / ieșire integrate	30	60	128	96	128
Capacitate de extindere (nr. max. de intrări / ieșiri posibile)	34	132	256	256	384
Memorie de program (pași)	2000	8000	16000	16000	64000
Durată ciclu per instrucțiune logică (μs)	0,55 – 0,7	0,55 – 0,7	0,08	0,08	0,065
Nr. de instrucțiuni (standard / limbaj secvențial / funcții speciale)	27 / 2 / 85	27 / 2 / 89	27 / 2 / 107	27 / 2 / 107	27 / 2 / 209
Nr. max. de module pentru funcții speciale ce pot fi conectate	—	2	8	4	8 stânga 10 dreapta

2.4 Alegerea controlerului potrivit

Unitățile de bază din seria MELSEC FX sunt disponibile în mai multe versiuni, cu diferite posibilități de alimentare și tipuri de ieșiri. Puteți alege o unitate cu alimentare la 100-240 V AC, 24 V DC sau 12-24 V DC și cu ieșiri pe releu sau pe tranzistori.

Seria	Intrări / Ieșiri	Tip	Nr. de intrări	Nr. de ieșiri	Alimentare	Tip ieșire
FX1S	10	FX1S-10 M□-□□	6	8	24 V DC sau 100 – 240 V AC	Tranzistor sau releu
	14	FX1S-14 M□-□□	8	6		
	20	FX1S-20 M□-□□	12	8		
	30	FX1S-30 M□-□□	16	14		
FX1N	14	FX1N-14 M□-□□	8	6	12 – 24 V DC sau 100 – 240 V AC	Tranzistor sau releu
	24	FX1N-24 M□-□□	14	10		
	40	FX1N-40 M□-□□	24	16		
	60	FX1N-60 M□-□□	36	24		
FX2N	16	FX2N-16 M□-□□	8	8	24 V DC sau 100 – 240 V AC	Tranzistor sau releu
	32	FX2N-32 M□-□□	16	16		
	48	FX2N-48 M□-□□	24	24		
	64	FX2N-64 M□-□□	32	32		
	80	FX2N-80 M□-□□	40	40		
	128	FX2N-128 M□-□□	64	64		
FX2NC	16	FX2NC-16 M□-□□	8	8	24 V DC	Tranzistor sau releu
	32	FX2NC-32 M□-□□	16	16		
	64	FX2NC-64 M□-□□	32	32		
	96	FX2NC-96 M□-□□	48	48		
FX3U	16	FX3U-16 M□-□□	8	8	24 V DC sau 100 – 240 V AC	Tranzistor sau releu
	32	FX3U-32 M□-□□	16	16		
	48	FX3U-48 M□-□□	24	24		
	64	FX3U-64 M□-□□	32	32		
	80	FX3U-80 M□-□□	40	40		
	128	FX3U-128 M□-□□	64	64	100 – 240 V AC	Tranzistor sau releu

Pentru a alege controlerul potrivit aplicației dumneavoastră, trebuie să vă puneți următoarele întrebări:

- De câte semnale de intrare aveți nevoie? (acestea pot fi contacte de comutare externe, butoane sau senzori).
- Ce tipuri de funcții trebuie să utilizați și de câte ori?
- Ce opțiuni de alimentare cu energie electrică aveți la dispoziție?
- Cât de mari sunt sarcinile pe care trebuie să le comute ieșirile? Alegeți ieșiri pe releu pentru comutarea de sarcini mari și ieșiri pe tranzistor pentru comutări statice de mare frecvență.

2.5 Părți componente

Toate controlerele FX au același design de bază. Elementele funcționale și ansamblurile principale sunt descrise în glosarul din secțiunea 2.5.7.

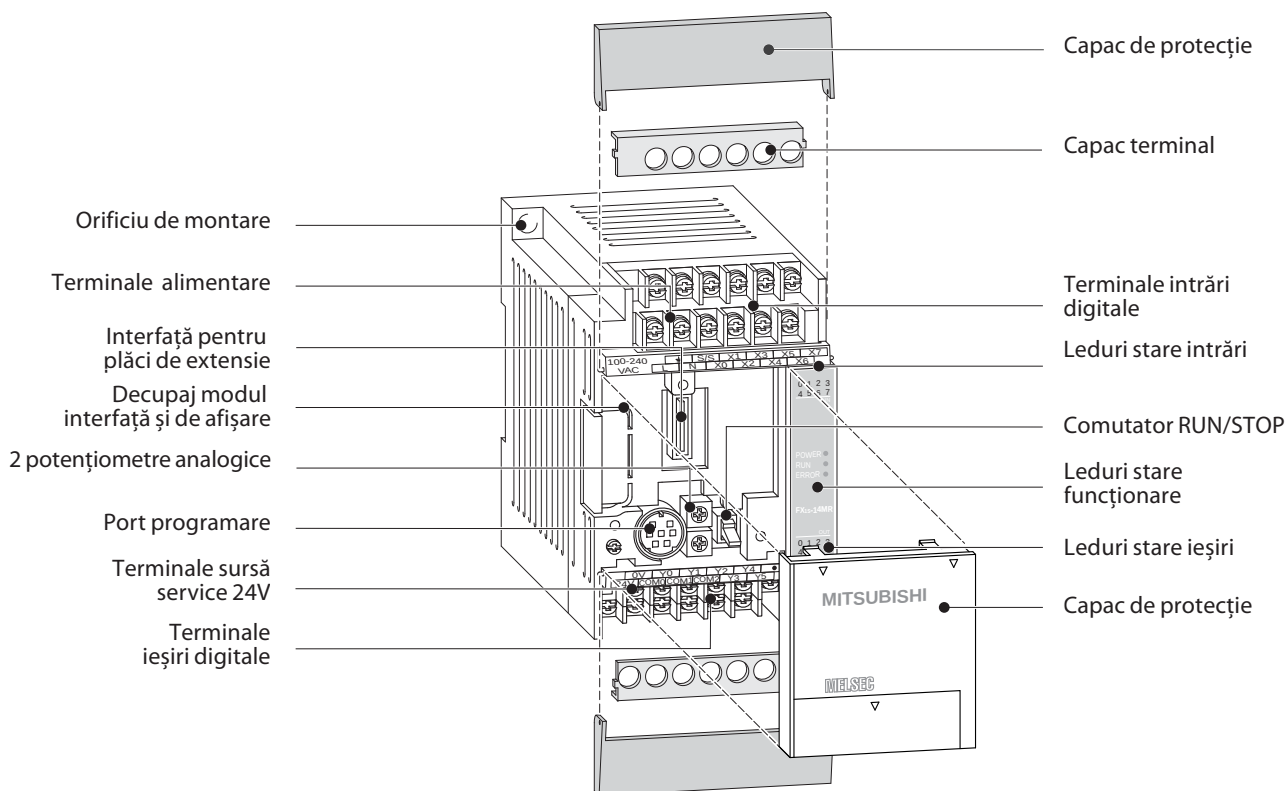
2.5.1 Circuite de intrare și de ieșire

Circuitele de intrare utilizează intrări flotante. Acestea sunt izolate electric de celelalte circuite ale automatului programabil, prin optocupluri. **Circuitele de ieșire** pot fi pe tranzistori sau pe rele. Ieșirile pe tranzistori sunt de asemenea izolate electric de celelalte circuite ale automatului programabil, prin optocupluri.

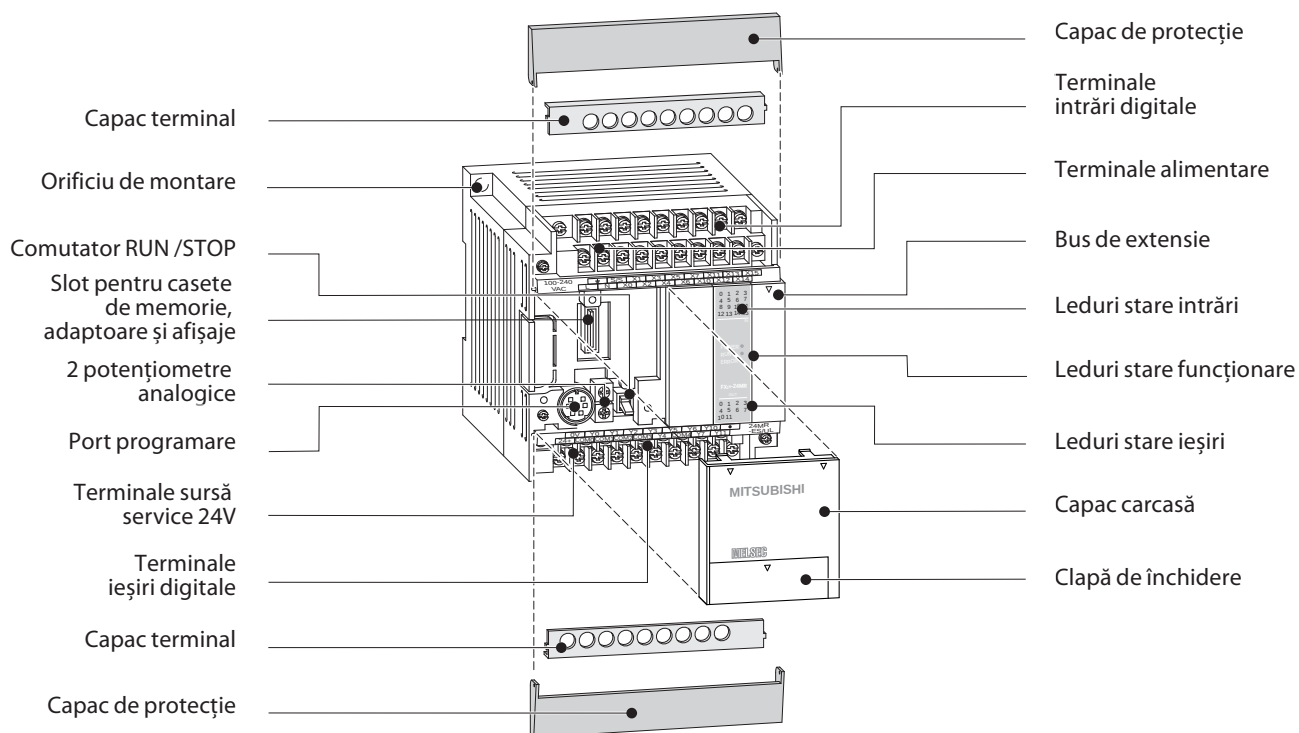
Tensiunea de comutare a tuturor intrărilor digitale trebuie să fie aceeași (de ex. 24 V DC). Această tensiune poate fi primită de la unitatea de alimentare integrată a automatului programabil. Dacă tensiunea de comutare la intrări este mai mică decât valoarea nominală (de exemplu, mai mică de 24 V DC), intrarea nu va fi procesată.

Curentul maxim de ieșire este de 2 A pentru modelele pe rele ce alimentează sarcini rezistive, 250 V AC trifazat și 0,5 A pentru sarcini alimentate la 24 V DC.

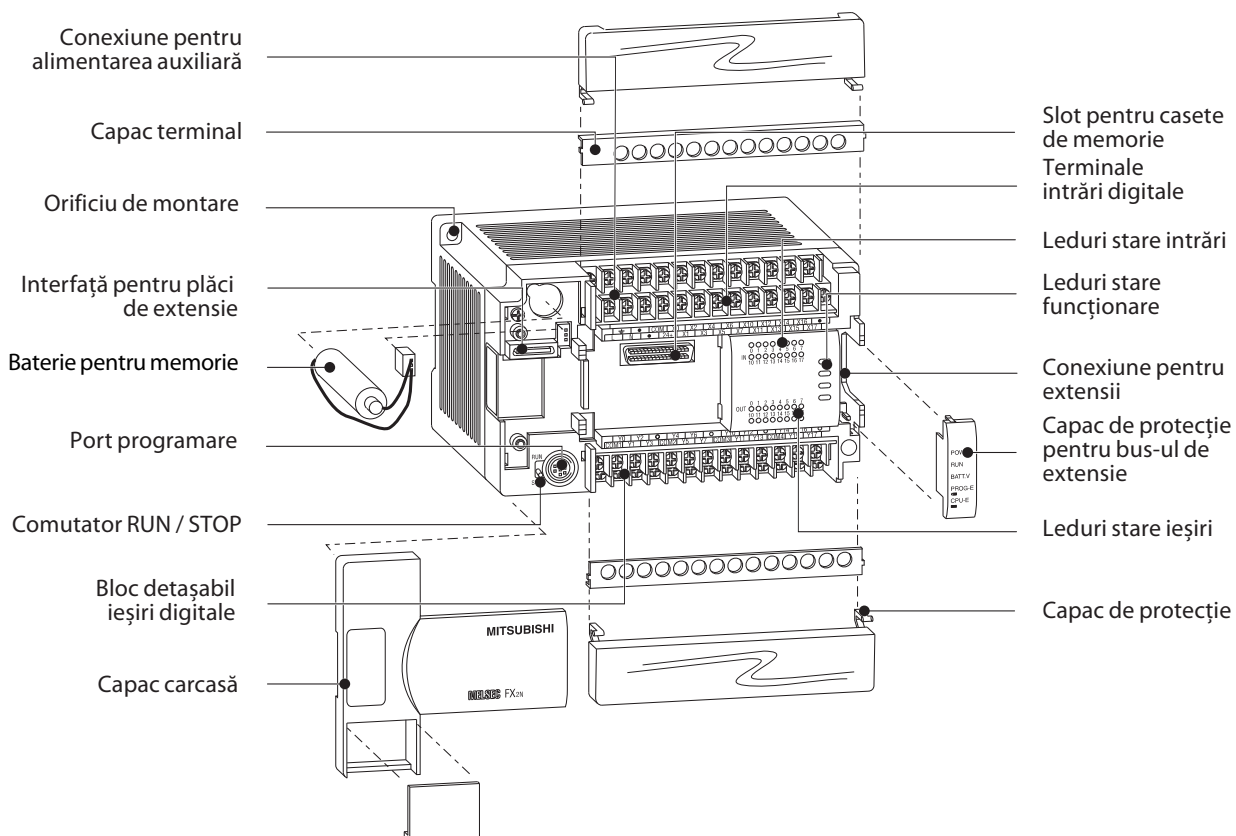
2.5.2 Alcătuirea unităților de bază MELSEC FX1S



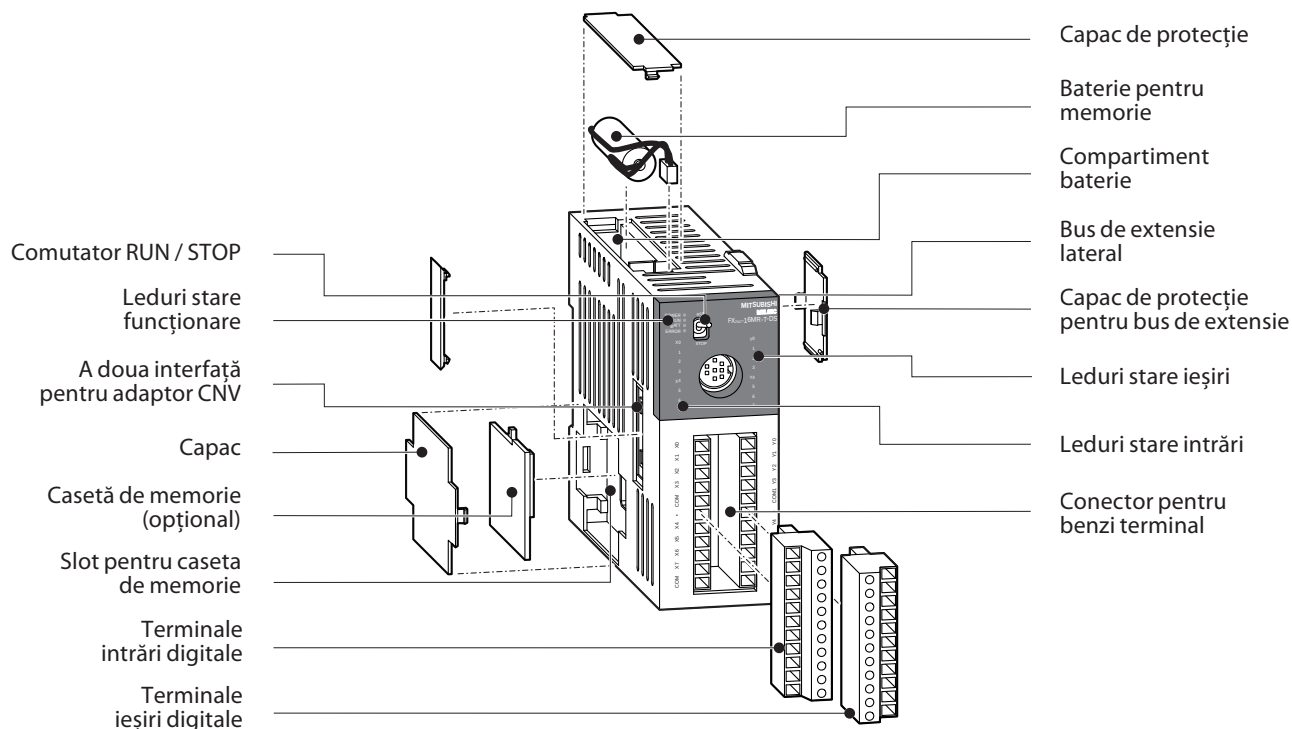
2.5.3 Alcătuirea unităților de bază MELSEC FX1N



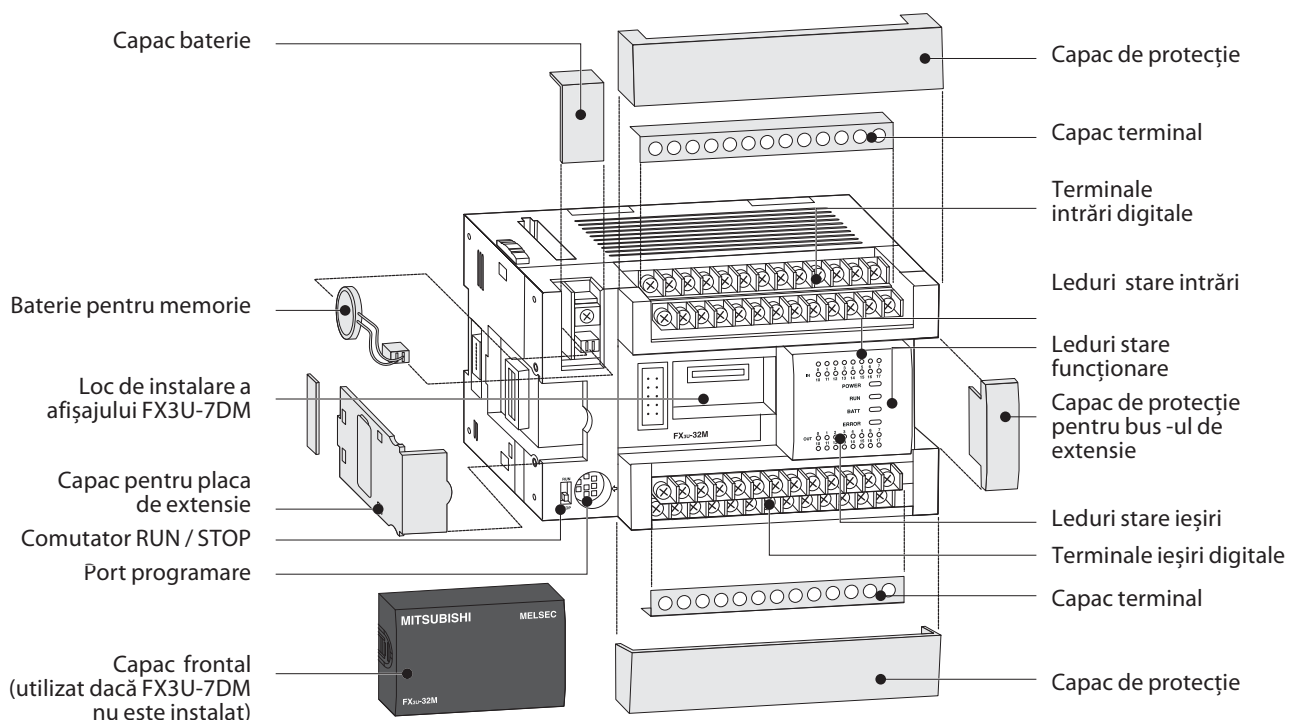
2.5.4 Alcătuirea unităților de bază MELSEC FX2N



2.5.5 Alcătuirea unităților de bază MELSEC FX2NC



2.5.6 Alcătuirea unităților de bază MELSEC FX3U



2.5.7 Glosar de componente ale automatelor programabile

Următorul tabel descrie semnificația și funcționalitatea componentelor și pieselor unui automat programabil Mitsubishi.

Componentă	Descriere
Conexiune pentru plăci de extensie	La această interfață pot fi conectate plăci adaptoare opționale. Sunt disponibile diverse adaptoare pentru toate seriile FX (cu excepția FX2NC). Aceste adaptoare extind capacitățile controlerelor cu funcții suplimentare sau interfețe de comunicații. Plăcile adaptoare sunt inserate direct în slot.
Port programare	Acest port poate fi utilizat pentru conectarea unității de programare mobile FX-20P-E sau a unui computer ori laptop extern ce conține un pachet de programare software (de ex. GX Developer FX)
EEPROM	Memorie de citire/scriere în care poate fi stocat programul automatului programabil, pentru a fi apoi citit cu software-ul de programare. Această memorie nevolatilă își păstrează conținutul atunci când alimentarea este întreruptă sau în situația unei căderi de tensiune fără a necesita o baterie.
Slot pentru caseta de memorie	Slot pentru casete de memorie opționale. Inserarea unei casete de memorie va dezactiva memoria internă a automatului – acesta va executa în acest caz doar programul stocat pe casetă.
Bus de extensie	Modulele de extensie suplimentare pentru intrări/ieșiri, precum și modulele pentru funcții speciale, ce adaugă capacități suplimentare sistemului automatului programabil, pot fi conectate aici. A se vedea Capitolul 6 pentru o prezentare generală a modulelor disponibile.
Potențiometre analogice	Potențiometrele analogice sunt utilizate pentru setarea unor valori de referință analogice. Setarea poate fi interogată de către programul automatului și utilizată pentru contoare, ieșiri pe puls și alte funcții (A se vedea Secțiunea 4.6.1)
Sursă auxiliară	Sursa de alimentare auxiliară (pentru toate modelele în afară de FX2NC) furnizează o tensiune stabilizată la 24 V DC, pentru semnalele de intrare și senzorii acestora. Capacitatea acestei surse de alimentare depinde de modelul automatului programabil (de ex. FX1S și FX1N: 400mA; FX2N-16M□-□□ la FX2N-32M□-□□: 250 mA, modelele de la FX2N-48M□-□□ la FX2N-64M□-□□: 460 mA)
Intrări digitale	Intrările digitale sunt utilizate pentru citirea semnalelor de control de la comutatorii, butoanele sau senzorii conectați. Aceste intrări pot citi valorile ON (prezență semnal) și OFF (fără semnal).
Ieșiri digitale	Puteți conecta diverse mecanisme de acționare și alte dispozitive la aceste ieșiri, în funcție de natura aplicației dvs. și de tipul ieșirii.
Leduri stare intrări	Aceste leduri arată ce intrări sunt active în momentul respectiv. Atunci când este aplicat un semnal unei intrări, ledul corespunzător se aprinde, indicând faptul că starea intrării este ON.
Leduri stare ieșiri	Aceste leduri arată stările curente ON/OFF ale ieșirilor digitale. Ieșirile pot comuta mai multe tensiuni și tipuri de curent electric, în funcție de modelul și tipul ieșirii.
Leduri stare funcționare	Ledurile RUN, POWER și ERROR afișează starea curentă a automatului programabil. POWER indică faptul că automatul este alimentat, RUN se aprinde atunci când se execută programul automatului, iar ERROR se aprinde atunci când se înregistrează o eroare sau o problemă de funcționare.
Baterie pentru memorie	Bateria protejează conținutul memoriei RAM volatile a automatului programabil MELSEC, în cazul unei căderi de tensiune (doar pentru FX2N, FX2NC și FX3U). Aceasta protejează zona de memorie latch a temporizatoarelor, contorilor și releelor intermediare. În plus, alimentează ceasul în timp real, atunci când automatul programabil este deconectat de la sursa de alimentare.
Comutator RUN / STOP	Automatele programabile au două moduri de operare, RUN și STOP. Comutatorul RUN / STOP vă permite comutarea manuală de la un mod la altul. În modul RUN, automatul programabil execută programul stocat în memorie. În modul STOP, execuția programului este oprită, iar automatul poate fi programat.

3 Introducere în programare

Un program constă dintr-o secvență de instrucțiuni de program. Aceste instrucțiuni determină funcționalitatea automatului programabil și sunt procesate secvențial, în ordinea în care au fost introduse de programator. Pentru a crea un program de automat programabil trebuie deci să analizați procesul, astfel încât să îl puteți controla și segmenta în pași ce pot fi reprezentați prin instrucțiuni. O instrucțiune de program, reprezentată de o linie sau o "treaptă" (în formatul bazat pe scheme cu contacte), este cea mai mică unitate a unui program de automat programabil.

3.1 Structura unei instrucțiuni de program

O instrucțiune de program constă din instrucțiunea însăși (uneori denumită comandă) și unul sau mai mulți operanzi (mai mulți în cazul instrucțiunilor aplicate), care într-un automat programabil sunt referințe către dispozitive. Unele instrucțiuni se introduc ca atare, fără specificarea nici unui operand – acestea sunt instrucțiunile care controlează execuția programului în automatul programabil.

Fiecărei instrucțiuni pe care o introduceți i se atribuie automat un număr de pas ce identifică în mod unic poziția acelei instrucțiuni în program. Acest lucru este important deoarece face posibilă introducerea aceleiași instrucțiuni, care să se refere la același dispozitiv, în mai multe locuri în program.

Ilustrațiile de mai jos arată modul în care sunt reprezentate instrucțiunile de program în formatele a două limbaje de programare: Schema cu contacte (LD, ladder diagram, stânga) și în Listă de instrucțiuni (IL, dreapta).



Instrucțiunea descrie ce trebuie făcut, adică funcția pe care doriți să o îndeplinească automatul programabil. Operandul sau dispozitivul este obiectul asupra căruia doriți să fie efectuată funcția. Acesta este alcătuit din două părți, numele și adresa dispozitivului:



Exemple de dispozitive:

Nume dispozitiv	Tip	Funcție
X	Intrare	Terminal de intrare aflat pe automatul programabil (de ex. conectat la un comutator)
Y	Ieșire	Terminal de ieșire aflat pe automatul programabil (de ex. pentru un contactor sau bec)
M	Relev	O memorie-tampon din automatul programabil, ce poate avea două stări, ON sau OFF.
T	Temporizator	Un "relev de timp", ce poate fi utilizat pentru programarea funcțiilor de temporizare.
C	Contor	Un contor
D	Registru de date	Pentru stocarea datelor în automatul programabil; acest registru permite stocarea valorilor măsurate, a rezultatelor calculelor etc.

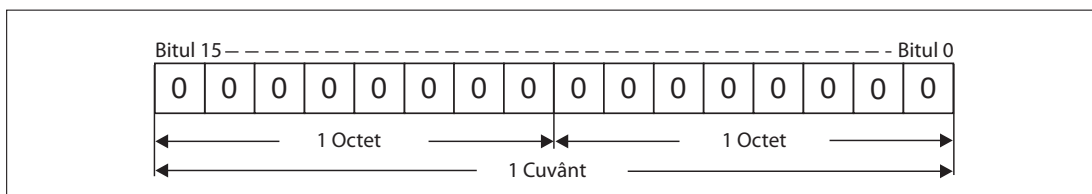
Pentru o descriere detaliată a dispozitivelor disponibile, consultați capitolul 4.

Fiecare dispozitiv este identificat după adresa sa. De exemplu, deoarece fiecare automat programabil are mai multe intrări, pentru a se citi o anumită intrare, trebuie să specificați și numele și adresa dispozitivului.

3.2 Biți, octeți și cuvinte

La fel ca peste tot în tehnologia digitală, cea mai mică unitate de informație dintr-un automat programabil este un "bit". Un bit poate avea doar două stări: "0" (OFF sau FALS) și "1" (ON sau ADEVĂRAT). Automatele programabile au mai multe așa-numite **dispozitive bit**, ce pot avea doar două stări; de exemplu: intrările, ieșirile și releele sunt astfel de dispozitive.

Următoarea unitate de măsură pentru informație este "octetul", care constă în 8 biți, iar "cuvântul" constă în doi octeți. În automatele programabile din seriile FX MELSEC, regiștrii de date sunt "dispozitive cuvânt", ceea ce înseamnă că pot stoca valori pe 16 biți.



Deoarece un registru de date are o capacitate de 16 biți, acesta poate stoca valori cuprinse între -32.768 și +32.767 (a se vedea capitolul 3.3). Atunci când trebuie stocate valori mai mari, sunt combinate două cuvinte pentru a se forma un cuvânt de 32 de biți, ce poate stoca valori cuprinse între -2.147.483.648 și +2.147.483.647. Contoarele, de exemplu, utilizează această capacitate.

3.3 Sisteme de numerație

Automatele programabile din seria FX MELSEC utilizează câteva sisteme de numerație diferite, pentru introducerea și afișarea valorilor, precum și pentru specificarea adreselor de dispozitive.

Sistemul zecimal

Sistemul de numerație zecimal este cel mai des utilizat în viața de zi cu zi. Este un sistem de numerație "pozițional în baza 10", în care fiecare cifră (poziție) dintr-un număr are o valoare de 10 ori mai mare decât cifra din dreapta sa. După ce se ajunge la 9 în fiecare poziție, valoarea acesteia redevine 0, iar următoarea poziție este incrementată cu 1, pentru a indica următoarea serie de 10 cifre (9 -> 10, 99 -> 100, 999 -> 1,000 etc).

- Baza: 10
- Cifre: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

În automatele programabile din seria FX MELSEC, numerele în baza 10 sunt utilizate pentru introducerea de constante și de valori de referință pentru temporizatoare și contoare. Adresele dispozitivelor sunt de asemenea introduse în sistem zecimal, cu excepția adreselor intrărilor și ieșirilor.

Sistemul binar

Ca orice alt computer, un automat programabil poate de fapt să distingă doar două stări, ON/OFF sau 0/1. Aceste "stări binare" sunt stocate în biți individuali. Atunci când trebuie introduse numere, sau când este necesară afișarea acestora în alte formate, software-ul de programare convertește automat numerele din sistemul binar în alte sisteme de numerație.

- Baza: 2
- Cifre: 0 și 1

Atunci când numerele din sistem binar sunt stocate într-un cuvânt (așa cum s-a arătat mai sus), valoarea fiecărei cifre (poziții) din cuvânt este o putere a numărului 2 mai mare decât cea a cifrei din dreapta sa. Principiul este exact același ca în reprezentarea în sistem zecimal, dar incrementul este 2, nu 10 (a se vedea ilustrația grafică):

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Notăție în Baza 2	Valoare zecimală	Notăție în Baza 2	Valoare în Baza 10
2^0	1	2^8	256
2^1	2	2^9	512
2^2	4	2^{10}	1024
2^3	8	2^{11}	2048
2^4	16	2^{12}	4096
2^5	32	2^{13}	8192
2^6	64	2^{14}	16384
2^7	128	2^{15}	32768*

* În valorile binare, bitul 15 este utilizat pentru a reprezenta semnul (dacă bitul 15 este egal cu zero, valoarea este pozitivă, iar dacă este egal cu unu, valoarea este negativă)

Pentru a converti o valoare binară într-o valoare din sistemul zecimal, trebuie doar să înmulțiți fiecare cifră ce are valoarea 1 cu puterea corespundătoare a lui 2, apoi să calculați suma rezultatelor.

Exemplu

00000010 00011001 (binar)

00000010 00011001 (binar) = $1 \times 2^9 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^0$

00000010 00011001 (binar) = $512 + 16 + 8 + 1$

00000010 00011001 (binar) = 537 (zecimal)

Sistemul hexazecimal

Numerele în sistem hexazecimal sunt mai ușor de tratat decât cele din sistemul binar, fiind și foarte ușor de convertit din sistemul binar. De aceea, numerele în hexazecimal sunt utilizate adesea în tehnologia digitală și în automatele programabile. În controlerile din seria FX MELSEC, numerele hexazecimale sunt utilizate pentru reprezentarea constantelor. În manualul de programare și alte manuale, numerele hexazecimale sunt întotdeauna identificate printr-un H plasat imediat după număr, pentru a se evita confuzia cu numerele zecimale (de ex. 12345H).

- Baza: 16
- Cifre: 0,1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (literele A, B, C, D, E și F reprezintă valorile zecimale 10, 11, 12, 13, 14 și 15)

Sistemul hexazecimal funcționează la fel ca sistemul de numere în baza 10, singura diferență este că trebuie să numărați până la F în baza 16, în loc de a număra până la 9, înainte de a reporni număratoarea de la 0 și de a incrementa următoarea cifră (FH → 10H, 1FH → 20H, 2FH → 30H, FFH → 100H etc). Valoarea unei cifre este o putere a numărului 16 și nu o putere a lui 10.

1A7FH

$16^0 = 1$	(În acest exemplu: 15×1	= 15)
$16^1 = 16$	(În acest exemplu: 7×16	= 112)
$16^2 = 256$	(În acest exemplu: 10×256	= 2560)
$16^3 = 4096$	(În acest exemplu: 1×4096	= 4096)
		<u>6783</u> (Zecimal)

Următorul exemplu ilustrează de ce este atât de simplu să se convertească valori binare în valori hexazecimale:

1	1	1	1	0	1	1	0	1	0	1	1	1	0	0	1	Binar
15				5				11				9				Zecimal*
F				5				B				9				Hexazecimal

* Conversia blocurilor de 4 biți în valori zecimale nu generează în mod direct o valoare care să corespundă exact valorii binare complete pe 16 biți! Prin contrast, valoarea binară poate fi convertită direct în sistem hexazecimal, iar valoarea rezultată în hexazecimal va avea exact aceeași valoare ca cea binară.

Sistemul octal

Intrările X8 și X9 și ieșirile Y8 și Y9 nu există în unitățile de bază ale seriei FX MELSEC. Acest lucru se întâmplă deoarece intrările și ieșirile automatelor programabile MELSEC sunt numerotate utilizându-se sistemul de numerație octal, în care nu există cifrele 8 și 9. În acest sistem, cifra curentă este resetată la 0 în momentul în care numărătoarea ajunge la 7, iar cifra din poziția următoare este incrementată 7 (0 – 7, 10 – 17, 70 – 77, 100 – 107 etc).

- Baza: 8
- Cifre: 0, 1, 2, 3, 4, 5, 6, 7

Sumar

Următorul tabel oferă o prezentare generală a celor patru sisteme de numeratie:

Baza 10	Baza 8	Baza 16	Baza 2
0	0	0	0000 0000 0000 0000
1	1	1	0000 0000 0000 0001
2	2	2	0000 0000 0000 0010
3	3	3	0000 0000 0000 0011
4	4	4	0000 0000 0000 0100
5	5	5	0000 0000 0000 0101
6	6	6	0000 0000 0000 0110
7	7	7	0000 0000 0000 0111
8	10	8	0000 0000 0000 1000
9	11	9	0000 0000 0000 1001
10	12	A	0000 0000 0000 1010
11	13	B	0000 0000 0000 1011
12	14	C	0000 0000 0000 1100
13	15	D	0000 0000 0000 1101
14	16	E	0000 0000 0000 1110
15	17	F	0000 0000 0000 1111
16	20	10	0000 0000 0001 0000
:	:	:	:
99	143	63	0000 0000 0110 0011
:	:	:	:

3.4 Setul de instrucțiuni de bază

Instrucțiunile automatelor programabile din seria FX MELSEC pot fi clasificate în două categorii: instrucțiuni elementare și instrucțiuni aplicate, care sunt uneori denumite "instrucțiuni pentru aplicație".

Funcțiile efectuate de către instrucțiunile elementare sunt comparabile cu funcțiile îndeplinite de controlerele cablate. Toate automatele programabile din familia MELSEC FX suportă setul de instrucțiuni de bază, dar instrucțiunile de aplicații suportate diferă de la un model la altul (a se vedea capitolul 5).

Ghid de referință pentru setul de instrucțiuni elementare

Instrucțiune	Funcție	Descriere	Referință
LD	Încarcă	Operație logică inițială, interoghează starea "1" a semnalului (normal deschis)	capitolul 3.4.1
LDI	Încarcă invers	Operație logică inițială, interoghează starea "0" a semnalului (normal închis)	
OUT	Instrucțiune pt. ieșire	Atribuire rezultatul unei operații logice unui dispozitiv	capitolul 3.4.2
AND	AND logic	Operație logică AND (ȘI), interoghează starea "1" a semnalului	capitolul 3.4.4
ANI	AND NOT	Operație logică AND NOT (ȘI NU), interoghează starea "0" a semnalului	
OR	OR logic	Operație logică OR (SAU), interoghează starea "1" a semnalului	capitolul 3.4.5
ORI	OR NOT	Operație logică OR NOT (SAU NU), interoghează starea "0" a semnalului	
ANB	Bloc AND	Conectează în serie un bloc de circuit de pe o ramură paralelă cu blocul paralel precedent.	capitolul 3.4.6
ORB	Bloc OR	Conectează în paralel un bloc serial de circuite cu blocul serial precedent.	
LDP	Instrucțiuni cu execuție pe front	Încarcă Puls, se încarcă la detectarea unei unui front crescător al semnalului	capitolul 3.4.7
LDF		Încarcă Puls descrescător, se încarcă la detectarea unui front descrescător al semnalului	
ANDP		ȘI Puls, ȘI logic pe frontul crescător al semnalului	
ANDF		ȘI Puls descrescător, ȘI logic pe frontul descrescător al semnalului	
ORP		SAU Puls, SAU logic pe frontul crescător al semnalului	
ORF		SAU Puls descrescător, SAU logic pe frontul descrescător al semnalului	
SET	Setare dispozitiv	Atribuire o stare de semnal ce este păstrată chiar dacă, condiția nu mai este adevărată	capitolul 3.4.8
RST	Resetare dispozitiv		
MPS	Stocare, citire și ștergere rezultat intermediar	Stocare într-un punct de memorie, stocarea într-o stivă a rezultatului unei operații	capitolul 3.4.9
MRD		Citire memorie, citirea dintr-o stivă a rezultatului stocat al unei operații	
MPP		Ștergere din memorie, citirea rezultatului stocat al unei operații și ștergerea acestuia din stivă	
PLS	Instrucțiuni pe puls	Puls, setează un dispozitiv pentru un ciclu de scanare la detectarea frontului crescător al condiției (intrarea își schimbă starea la ON)	capitolul 3.4.10
PLF		Puls pe front descrescător, setează un dispozitiv* pentru un ciclu de scanare la detectarea frontului descrescător al condiției (intrarea își schimbă starea la OFF)	
MC	Control principal	Instrucțiuni pentru activarea sau dezactivarea execuției unor părți definite din program	capitolul 3.4.11
MCR	Control principal – Resetare		
INV	Inversare	Inversează rezultatul unei operații	capitolul 3.4.12

3.4.1 Operații logice inițiale

Instrucțiune	Funcție	Simbol	GX Developer FX
LD	Instrucțiunea Încarcă pornește o operație logică și verifică existența în dispozitiv a stării de semnal "1"		
LDI	Instrucțiunea Încarcă invers pornește o operație logică și verifică existența în dispozitiv a stării de semnal "0"		

Un circuit dintr-un program începe întotdeauna cu o instrucțiune LD sau LDI. Aceste instrucțiuni pot fi executate asupra intrărilor, releelor, temporizatoarelor și contoarelor.

Pentru exemple privind utilizarea acestor instrucțiuni, consultați descrierea instrucțiunii OUT din secțiunea următoare.

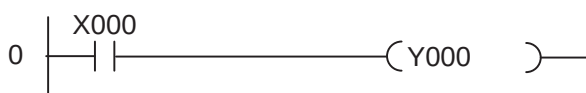
3.4.2 Utilizarea rezultatului unei operații logice

Instrucțiune	Funcție	Simbol	GX Developer FX
OUT	Instrucțiune de ieșire, atribuie rezultatul unei operații unui dispozitiv		

Instrucțiunea OUT poate fi utilizată pentru a întrerupe un circuit de program. Puteți de asemenea programa circuite care să utilizeze mai multe instrucțiuni OUT. Acesta nu este însă în mod necesar finalul programului. Dispozitivul setat cu rezultatul operației ce utilizează instrucțiunea OUT poate fi apoi utilizat ca stare de semnal de intrare în pașii ulteriori ai programului.

Exemplu (Instrucțiuni LD și OUT)

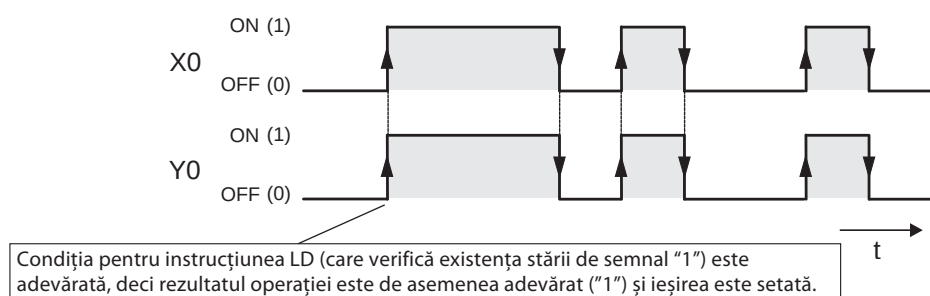
Schemă cu contacte



Listă de instrucțiuni

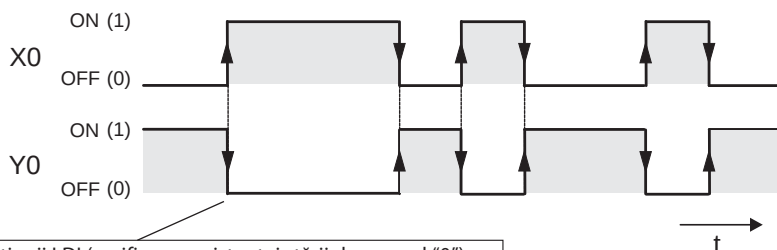
0	LD	X000
1	OUT	Y000

Aceste două instrucțiuni vor rezulta în următoarea secvență de semnale:



Exemplu (Instrucțiuni LDI și OUT)Schemă cu contacteListă de instrucțiuni

0	LDI	X000
1	OUT	Y000

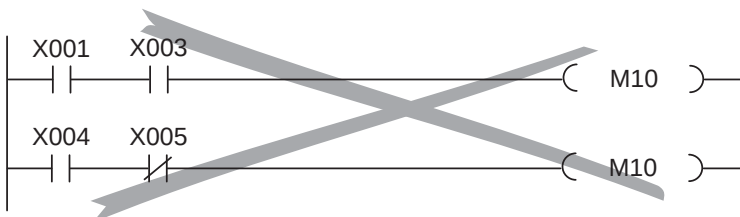


Condiția instrucțiunii LDI (verificarea existenței stării de semnal "0") nu mai este adevărată, astfel încât ieșirea este resetată.

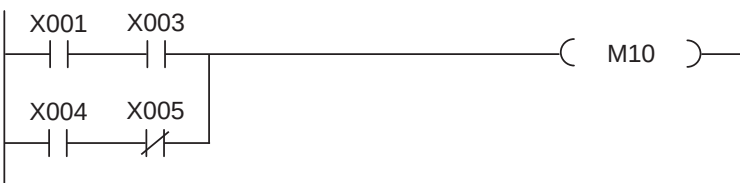
Dubla atribuire a releelor sau ieșirilor

Nu atribuiți niciodată rezultatul unei operații aceluiași dispozitiv în mai multe locuri în program!

Programul este executat secvențial de sus în jos, astfel încât în acest exemplu, a doua atribuire a M10 ar suprascrie pur și simplu rezultatul primei atribuiți.



Puteți rezolva această problemă cu ajutorul modificării afișate în dreapta. Acest lucru va lua în considerare toate condițiile de intrare necesare și va seta rezultatele corect.



3.4.3 Utilizarea comutatorilor și senzorilor

Înainte de a continua cu descrierea celorlalte instrucțiuni, vom descrie modul în care semnalele primite de la comutatori, senzori etc. pot fi utilizate în programele dumneavoastră.

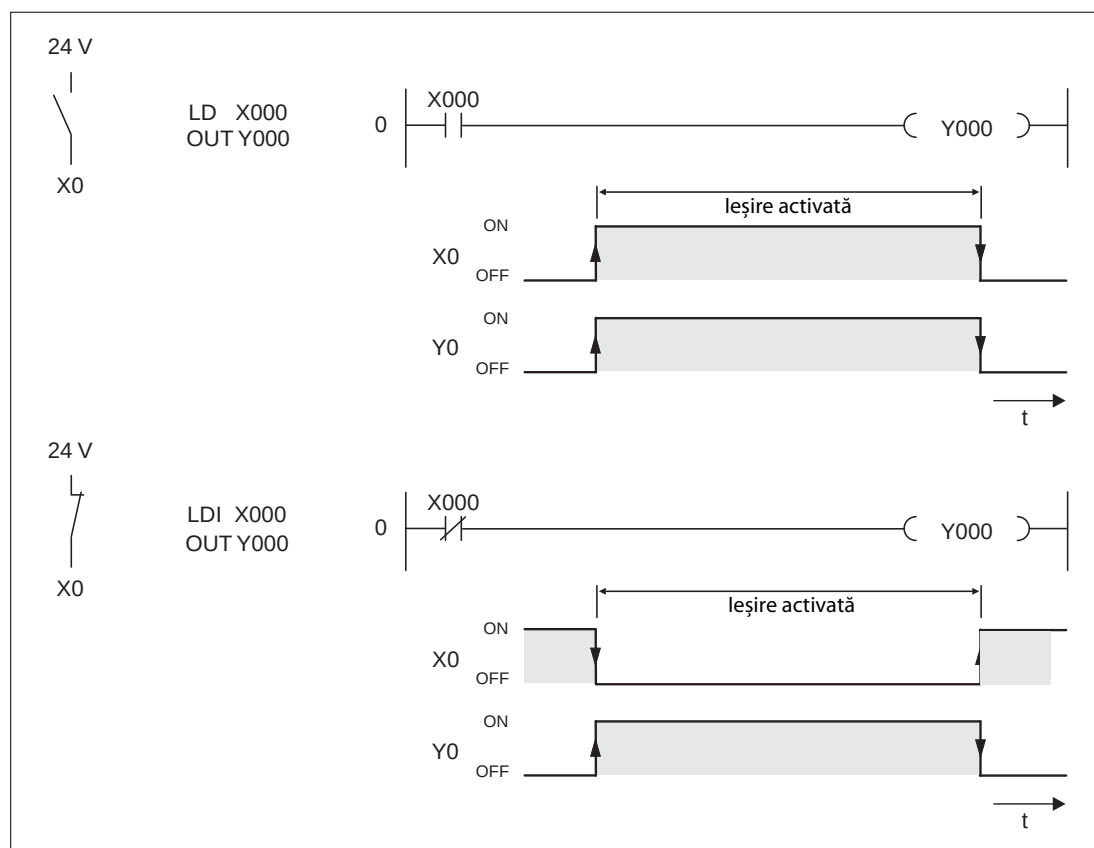
Programele din automatele programabile trebuie să poată răspunde la semnale de la comutatori, butoane și senzori, pentru o funcționare corectă. Este important să înțelegeți că instrucțiunile de program pot interoga doar starea binară a semnalului intrării specificate – indiferent de tipul intrării și de modul în care este controlată.

	Contact normal deschis	Atunci când este comandat un contact normal deschis, intrarea este setată (ON, stare "1" a semnalului)
	Contact normal închis	Atunci când este comandat un contact normal închis, intrarea este resetată (OFF, stare "0" a semnalului)

După cum vă puteți imagina, acest lucru înseamnă că atunci când scrieți propriul dumneavoastră program trebuie să știți dacă elementul conectat la intrarea automatului programabil este un contact normal deschis sau normal închis. O intrare conectată la un contact normal deschis trebuie tratată diferit de o intrare conectată la un contact normal închis. Următorul exemplu ilustrează acest lucru.

De obicei sunt utilizați comutatori cu contacte normal deschise. Uneori însă, se utilizează, din motive de securitate, contacte normal închise – de exemplu pentru deconectarea acționărilor (a se vedea secțiunea 3.5).

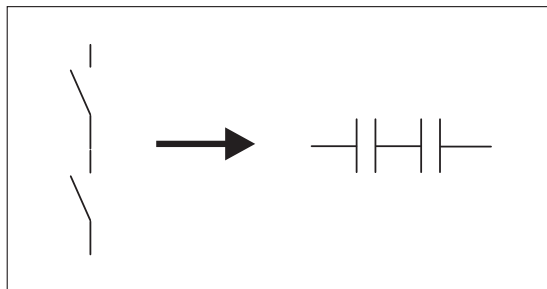
Ilustrația de mai jos arată două secvențe de program în care rezultatul este exact același, chiar dacă se utilizează tipuri diferite de comutatori. Când se operează comutatorul, ieșirea este setată (pornită).



3.4.4 Operații AND (ȘI)

Instrucțiune	Funcție	Simbol	GX Developer FX
AND	AND logic (operație AND cu interogarea stării semnalului pentru a verifica dacă este "1" sau ON)		
ANI	AND NOT logic (operație AND logică ce interoghează starea semnalului pentru a verifica dacă este "0" sau OFF)		

O operație AND (ȘI) este logic identică unei conexiuni seriale a doi sau mai mulți comutatori dintr-un circuit electric. Curentul va fi transmis doar dacă toți comutatorii sunt închiși. Dacă unul sau mai mulți comutatori sunt deschiși, curentul nu va fi transmis – condiția AND (ȘI) este falsă.

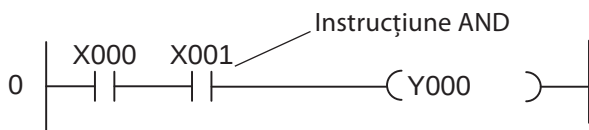


Rețineți că software-ul de programare utilizează aceleași pictograme și taste funcționale pentru instrucțiunile AND și ANI și pentru instrucțiunile LD și LDI. Când programați în limbajul Schemă cu contacte (Ladder Diagram), software-ul atribuie automat instrucțiunile corecte pe baza poziției de inserare.

Când programați în formatul Listă de instrucțiuni, rețineți că nu puteți utiliza instrucțiunile AND și ANI la începutul unui circuit (o linie de program în formatul Schemă cu contacte)! Circuitele trebuie să înceapă cu o instrucțiune LD sau LDI (a se vedea capitolul 3.4.1).

Exemplu de instrucțiune AND

Schemă cu contacte



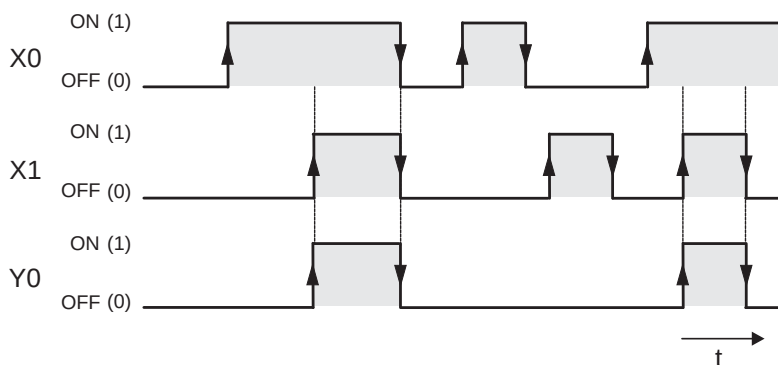
Listă de instrucțiuni

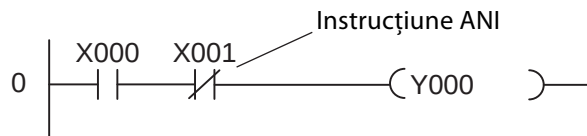
```

0  LD    X000
1  AND   X001
2  OUT   Y000

```

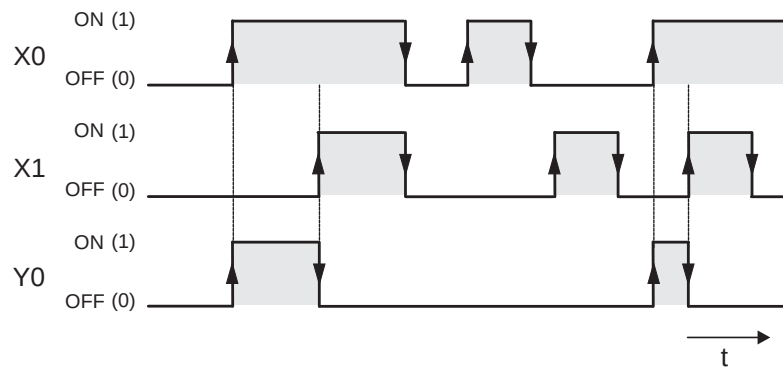
În acest exemplu, ieșirea Y0 este comutată pe pornit doar atunci când X0 și X1 sunt **ambele** pornite:



Example of an ANI instructionSchemă cu contacteListă de instrucțiuni

0	LD	X000
1	ANI	X001
2	OUT	Y000

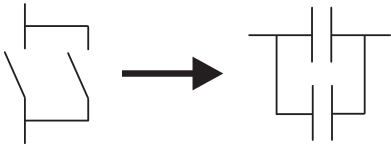
În exemplu, ieșirea Y0 este comutată pe pornit atunci când intrarea X0 este activată, iar intrarea X1 este dezactivată.



3.4.5 Operații OR

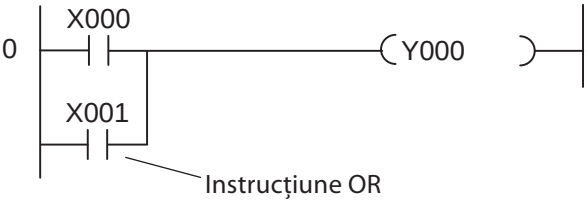
Instrucțiune	Funcție	Simbol	GX Developer FX
OR	OR logic (operație OR cu interogarea stării semnalului pentru a se verifica dacă este "1" sau ON)		
ORI	OR NOT logic (operație OR cu interogarea stării semnalului pentru a se verifica dacă este "0" sau OFF)		

O operație OR (SAU) este logic echivalentă unei conexiuni în paralel a mai multor comutatori dintr-un circuit electric. Curentul va intra în momentul în care se închide unul dintre comutatori. Fluxul de curent se va opri doar dacă toți comutatorii sunt închși.



Exemplu de instrucțiune OR

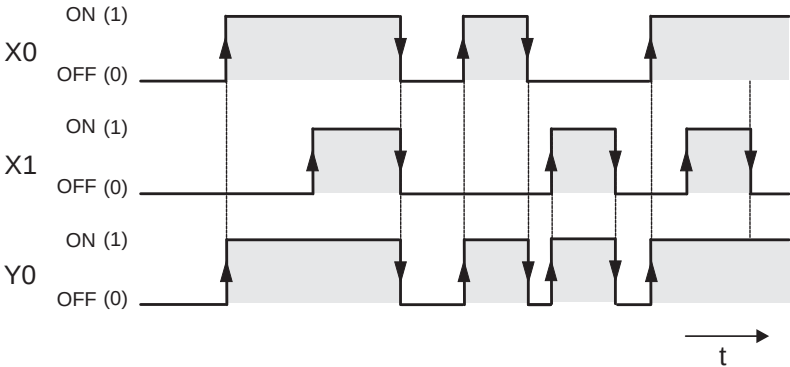
Schemă cu contacte

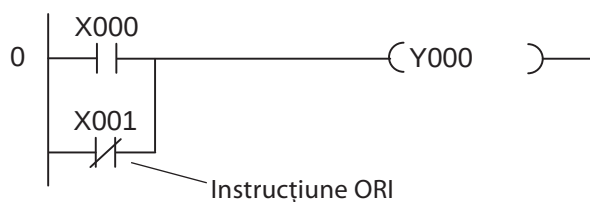


Listă de instrucțiuni

0	LD	X000
1	OR	X001
2	OUT	Y000

În exemplu, ieșirea Y0 este comutată pe pornit când fie intrarea X0, fie intrarea X1 sunt pornite:



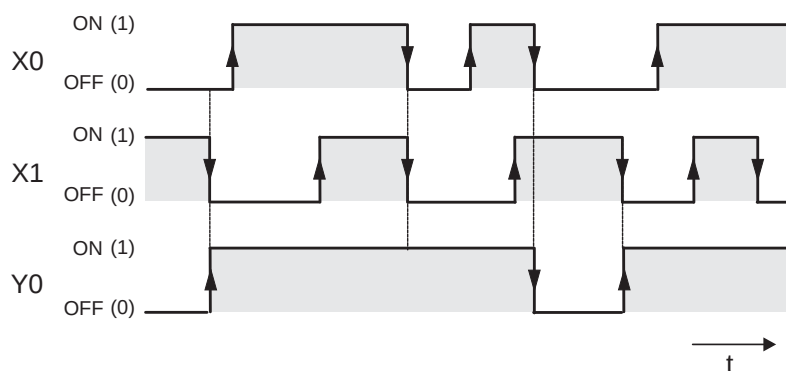
Exemplu de instrucțiune ORISchemă cu contacteListă de instrucțiuni

```

0  LD    X000
1  ORI    X001
2  OUT   Y000

```

În acest exemplu, ieșirea Y0 este comutată pe pornit doar atunci când fie X0 este pornit, fie X1 este oprit:

**3.4.6****Instrucțiuni pentru conectarea blocurilor de operare**

Instrucțiune	Funcție	Simbol	GX Developer FX
ANB	Bloc AND (conexiune serială de blocuri sau de operații/circuite paralele)	—	
ORB	Bloc OR (conexiune în paralel de blocuri sau de operații/circuite în serie)		

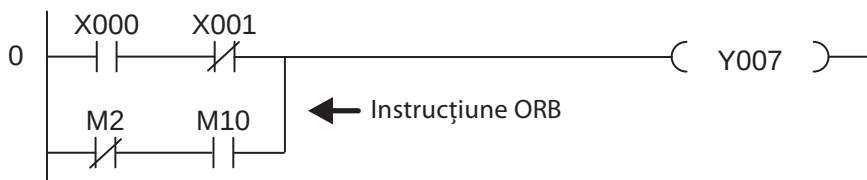
Deși instrucțiunile ANB și ORB sunt instrucțiuni pentru automatul programabil, în modul de programare Schemă cu contacte, ele sunt afișate și introduse doar ca linii de conectare. În formatul Listă de instrucțiuni, sunt afișate ca instrucțiuni și trebuie introduse cu acronimele corespunzătoare, ANB și ORB.

Ambele instrucțiuni se introduc fără dispozitive și pot fi utilizate cât de des doriți într-un program. Numărul maxim de instrucțiuni LD și LDI este însă restricționat la 8 pe circuit, ceea ce limitează de asemenea la 8 numărul de instrucțiuni ORB sau ANB pe care puteți utiliza înainte de o instrucțiune pentru o ieșire.

Exemplu de instrucțiune ANBSchemă cu contacteListă de instrucțiuni

0	LD	X000	
1	ORI	M2	← Prima conexiune paralelă (operație OR)
2	LDI	X001	
3	OR	M10	← A doua conexiune paralelă (operație OR)
4	ANB		← Instrucțiune ANB ce conectează cele două operații OR
5	OUT	Y007	

În acest exemplu, ieșirea Y07 este comutată pe pornit dacă intrarea X00 are starea "1", sau dacă releul M2 are starea "0" și intrarea X01 are starea "0", sau dacă releul M10 are starea "1".

Exemplu de instrucțiune ORBSchemă cu contacteListă de instrucțiuni

0	LD	X000	
1	ANI	X001	← Prima conexiune serială (operație AND)
2	LDI	M2	
3	AND	M10	← A doua conexiune serială (operație AND)
4	ORB		← Instrucțiune ORB ce conectează cele două operații AND
5	OUT	Y007	

În acest exemplu, ieșirea Y07 este comutată pe pornit dacă intrarea X00 are starea "1" și intrarea X01 are starea "0", sau dacă releul M2 are starea "0" și releul M10 are starea "1".

3.4.7 Operații cu execuție pe front

Instrucțiune	Funcție	Simbol	GX Developer FX
LDP	Încarcă pe front, se încarcă pe frontul crescător al semnalului din dispozitiv		
LDF	Încarcă pe front descrescător, se încarcă pe frontul descrescător al semnalului din dispozitiv		
ANDP	Puls AND, operație logică AND pe front crescător al semnalului din dispozitiv		
ANDF	Puls AND pe front descrescător, operație logică AND pe front descrescător al semnalului din dispozitiv		
ORP	Puls OR, operație logică OR pe front crescător al semnalului din dispozitiv		
ORF	Puls OR descrescător, operație logică OR pe front descrescător al semnalului din dispozitiv		

În programele pentru automate programabile, va fi adesea necesar să detectați și să creați un răspuns pentru frontul crescător sau descrescător al semnalului unui dispozitiv bit. Un front de impuls crescător va indica o comutare a dispozitivului de la "0" la "1", iar un front de impuls descrescător va indica o comutare de la "1" la "0".

Pe parcursul execuției, operațiile cu acțiune pe front furnizează valoarea "1" când semnalul de stare al dispozitivului respectiv se modifică.

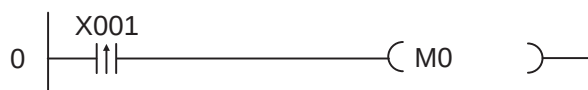
Când este necesar să utilizați aceste operații? De exemplu, să presupunem că aveți o bandă transportoare cu un senzor ce se activează pentru a incrementa un contor de fiecare dată când pe bandă trece un pachet. Dacă nu utilizați o funcție cu acțiune pe front, veți avea rezultate incorecte, deoarece contorul va fi incrementat cu 1 la fiecare ciclu de program în care comutatorul este înregistrat ca setat. Dacă înregistrați doar impulsul crescător al semnalului comutatorului, contorul va fi incrementat corect, cu 1, la fiecare pachet ce ajunge pe bandă.

Notă

Majoritatea instrucțiunilor aplicate pot fi de asemenea executate pe frontul semnalului. Pentru detalii, a se vedea capitolul 5).

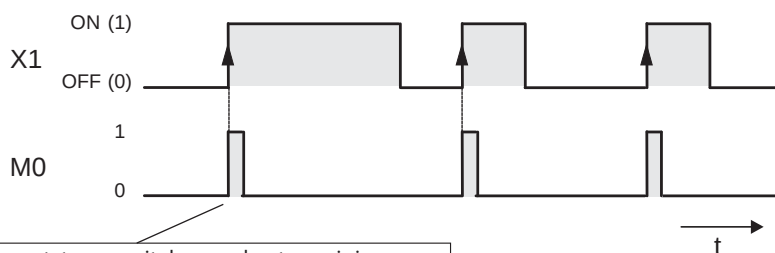
Evaluarea unui impuls de semnal crescător

Schemă cu contacte



Listă de instrucțiuni

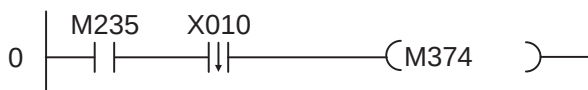
0	LDP	X001
1	OUT	M0



Releul M0 este comutat pe pornit doar pe durata unui singur ciclu de program.

Evaluarea unui impuls de semnal descrescător

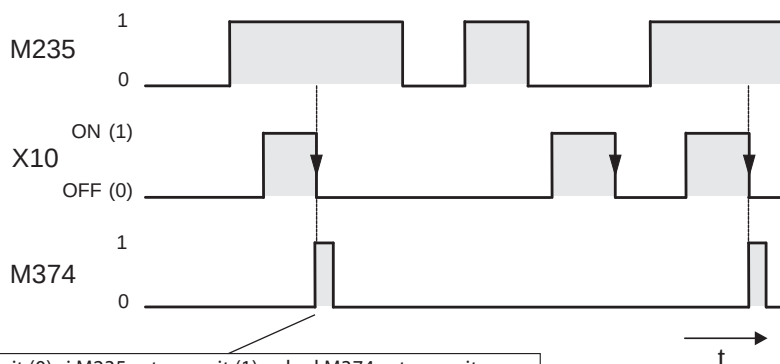
Schemă cu contacte



Listă de instrucțiuni

```

0 LD    M235
1 ANDF  X010
2 OUT   M374
  
```



Dacă X0 este oprit (0) și M235 este pornit (1), releul M374 este pornit pentru un singur ciclu de program

Cu excepția caracteristicii de declanșare pe front, funcțiile instrucțiunilor LDP, LDF, ANDP, ANDF, ORP și ORF sunt identice cu cele ale instrucțiunilor LD, AND și OR. Aceasta înseamnă că puteți utiliza operații declanșate pe front în programele dvs. în același mod în care utilizați versiunile lor convenționale.

3.4.8 Setarea și resetarea dispozitivelor

Instrucțiune	Funcție	Simbol	GX Developer FX
SET	Setarea unui dispozitiv® (atribuirea stării de semnal "1")		
RST	Resetarea unui dispozitiv® (atribuirea stării de semnal "0")		

- ① Instrucțiunea SET poate fi utilizată pentru setarea ieșirilor (Y), a releelor (M) și a releelor de stare (S).
- ② Instrucțiunea RST poate fi utilizată pentru resetarea ieșirilor (Y), a releelor (M), a releelor de stare (S), a temporizatoarelor (T), a contoarelor (C) și a regiștrilor (D, V, Z).

În mod normal, starea semnalului unei instrucțiuni OUT va rămâne "1" doar atâta timp cât rezultatul operației conectate la instrucțiunea OUT este tot "1". De exemplu, dacă veți conecta un buton care se apasă la o intrare, iar la ieșirea corespundentă veți conecta un bec, apoi le veți conecta cu o instrucțiune LD și o instrucțiune OUT, becul va rămâne aprins doar atâta vreme cât butonul rămâne apăsat.

Instrucțiunea SET poate fi utilizată pentru a folosi un impuls scurt de comutare în scopul de a porni (seta) o ieșire sau un releu și a le lăsa în starea pornit. Dispozitivul va rămâne apoi pornit până îl opriți (resetati) cu o instrucțiune RST. Acest lucru vă permite să implementați "funcții de memorare" sau să comutați acționările între stările pornit și oprit cu ajutorul butoanelor. (Ieșirile sunt de obicei oprite de asemenea atunci când automatul programabil este oprit sau este întreruptă alimentarea cu energie. Unele releu însă își păstrează starea ultimului semnal în aceste condiții – de exemplu, un releu pornit va rămâne pornit în acest caz.)

Pentru a introduce o instrucțiune SET sau RST în formatul Schemă cu contacte, faceți clic pe pictograma afișată în tabelul de mai sus, în GX Developer, sau apăsați tasta F8. Apoi introduceți instrucțiunea și numele dispozitivului pe care doriți să îl setați sau resetati, de exemplu SET Y1.

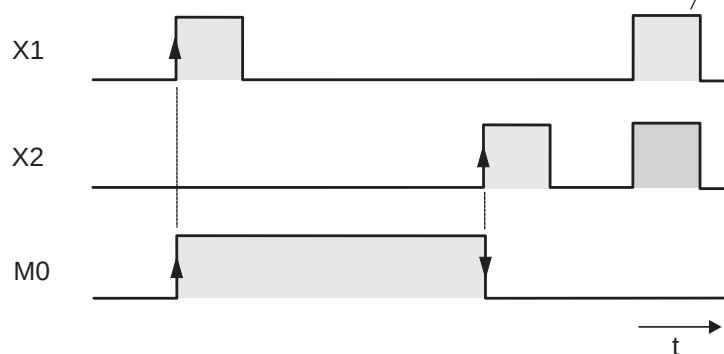
Schemă cu contacte



Listă de instrucțiuni

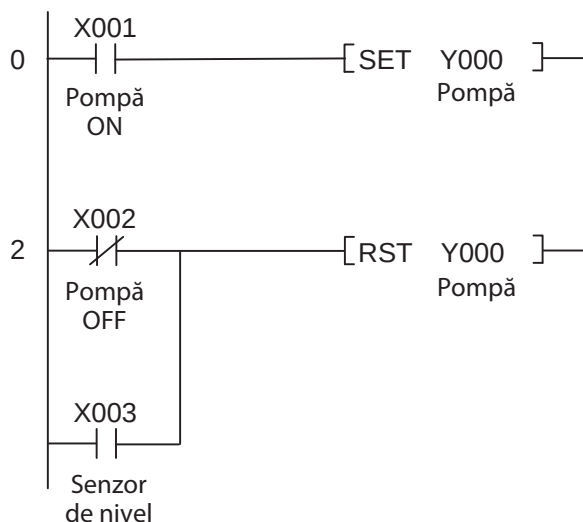
0	LD	X001
1	SET	M0
2	LD	X002
3	RST	M0

Dacă instrucțiunile de setare și resetare pentru același dispozitiv au amândouă rezultatul "1", prioritatea este a ultimei operații executate. În acest exemplu, ultima operație executată este instrucțiunea RST, deci M0 rămâne oprit.



Acest exemplu este un program pentru controlarea unei pompe ce umple un container. Pompa este controlată manual cu două butoane, ON și OFF. Din motive de siguranță, pentru funcția OFF se utilizează un contact normal închis. Atunci când containerul s-a umplut, un senzor de nivel oprește automat pompa.

Schemă cu contacte



Listă de instrucțiuni

0	LD	X001
1	SET	Y000
2	LDI	X002
3	OR	X003
4	RST	Y000

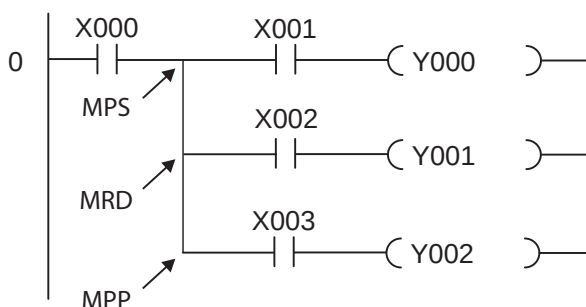
3.4.9 Stocarea, citirea și ștergerea rezultatelor operațiilor

Instrucțiune	Funcție	Simbol	GX Developer FX
MPS	Stocare punct de memorie, stochează rezultatul unei operații	—	—
MRD	Citire memorie, citește rezultatul unei operații	—	—
MPP	Aducere din memorie, citește rezultatul stocat al unei operații și îl șterge	—	—

Instrucțiunile MPS, MRD și MPP sunt utilizate pentru stocarea rezultatelor operațiilor și a valorilor intermediare într-o memorie numită “stivă”. Aceste instrucțiuni permit programarea de operații pe mai multe niveluri, ceea ce face ca programele să fie mai ușor de citit și de gestionat.

Atunci când introduceți programe în formatul Schemă cu contacte, aceste instrucțiuni sunt inserate automat de către software-ul de programare. Instrucțiunile MPS, MRD și MPP sunt afișate efectiv doar în formatul Listă de instrucțiuni, iar când programați în acest format, ele trebuie introduse manual.

Schemă cu contacte



Listă de instrucțiuni

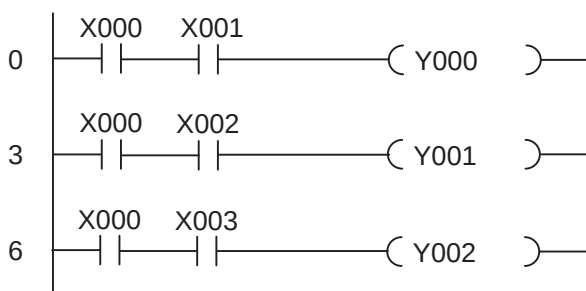
```

0  LD      X000
1  MPS
2  AND     X001
3  OUT     Y000
4  MRD
5  AND     X002
6  OUT     Y001
7  MPP
8  AND     X003
9  OUT     Y002

```

Pentru ca avantajul acestor instrucțiuni să fie mai evident, exemplul de mai jos arată aceeași secvență de program, programată fără MPS, MRD și MPP:

Schemă cu contacte



Listă de instrucțiuni

```

0  LD      X000
1  AND     X001
2  OUT     Y000
3  LD      X000
4  AND     X002
5  OUT     Y001
6  LD      X000
7  AND     X003
8  OUT     Y002

```

Atunci când utilizați această abordare, trebuie să programați dispozitivele (X0 în acest exemplu) de mai multe ori. Aceasta duce la mai multă muncă de programare, iar aceasta ce poate fi o diferență majoră în cazul programelor lungi și al construcțiilor de circuite complexe.

În ultima instrucțiune pentru ieșire, pentru ștergerea stivei trebuie să utilizați MPP în loc de MRD. Puteți utiliza mai multe instrucțiuni MPS pentru a crea operații cu maximum 11 niveluri. Pentru mai multe exemple de utilizare a instrucțiunilor MPS, MRD și MPP, consultați Manualul de programare pentru seria FX.

3.4.10 Generarea de impulsuri

Instrucțiune	Funcție	Simbol	GX Developer FX
PLS	Puls, setează un dispozitiv* pe durata unui singur ciclu de program la detectarea frontului crescător al condiției sau dispozitivului de intrare		
PLF	Puls descrescător, setează un dispozitiv* pe durata unui singur ciclu de program la detectarea frontului descrescător al condiției sau dispozitivului de intrare		

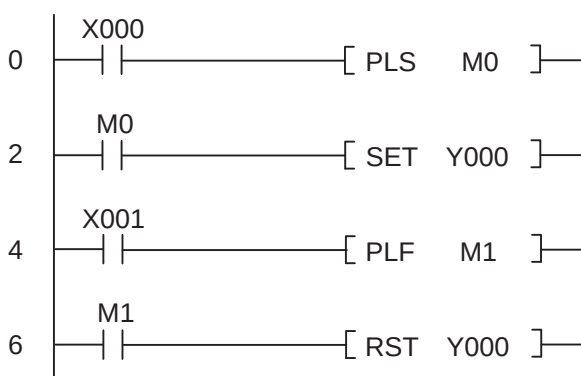
* Instrucțiunile PLS și PLF pot fi utilizate pentru setarea ieșirilor (Y) și a releelor (M).

Aceste instrucțiuni convertesc efectiv un semnal static într-un impuls scurt, a cărui durată depinde de lungimea ciclului de program. Dacă utilizați instrucțiunea PLS în locul unei instrucțiuni OUT, starea semnalului din dispozitivul specificat va fi setată la "1" doar pentru un singur ciclu de program, mai exact în timpul ciclului în care starea semnalului din dispozitivul de dinaintea instrucțiunii PLS din circuit comută de la "0" la "1" (impuls crescător).

Instrucțiunea PLF răspunde unui puls de semnal descrescător și setează dispozitivul specificat la "1" pentru un singur ciclu de program, în timpul ciclului în care starea semnalului de comandă comută de la "1" la "0" (impuls descrescător).

Pentru a introduce o instrucțiune PLS sau PLF în formatul Schemă cu contacte, faceți clic în bara de stare GX Developer, de pe pictograma cu instrumente afișată mai sus sau apăsați F8. Apoi introduceți în caseta de dialog instrucțiunea și dispozitivul corespunzător ce trebuie setat, de exemplu PLS Y2.

Schemă cu contacte

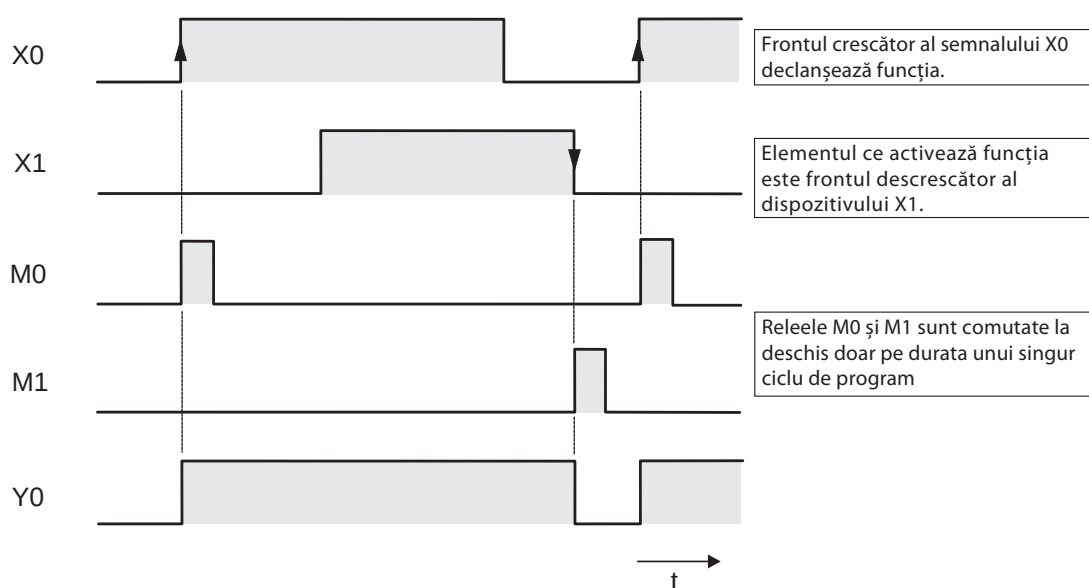


Listă de instrucțiuni

```

0 LD      X000
1 PLS     M0
2 LD      M0
3 SET     Y000
4 LD      X001
5 PLF     M1
6 LD      M1
7 RST     Y000

```



3.4.11 Funcția de control principal (instrucțiunile MC și MCR)

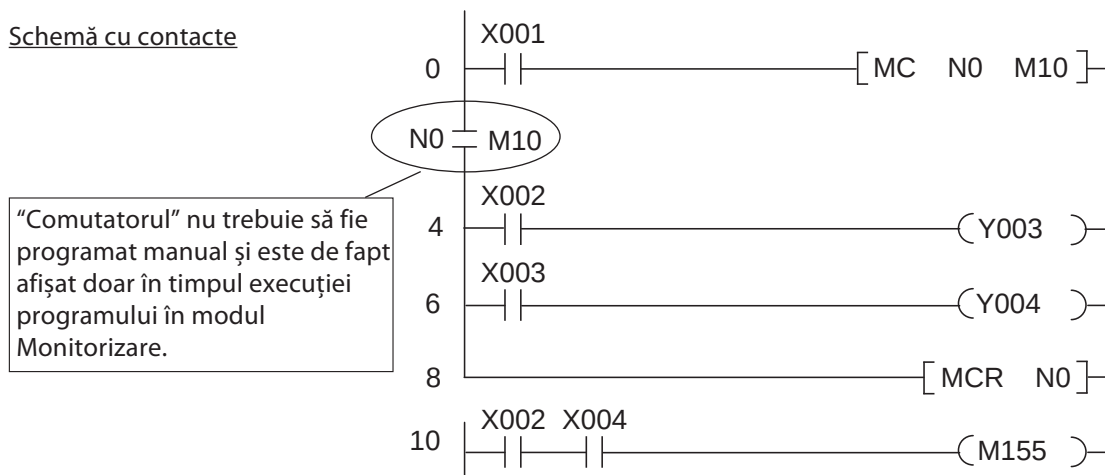
Instrucțiune	Funcție	Simbol	GX Developer FX
MC	Control principal, setează o condiție de control principal, marcând începutul unui bloc ^① de program	MC n □	
MCR	Resetare control principal, resetează o condiție de control principal, marcând finalul unui bloc ^② de program	MCR n	

① Instrucțiunea MC poate fi utilizată asupra ieșirilor (Y) și a releelor (M). n: N0 până la N7

② n: N0 până la N7

Instrucțiunile MC (setare control principal) și MCR (resetare) pot fi utilizate pentru setarea condițiilor pe baza cărora blocurile de program individuale pot fi activate sau dezactivate. În formatul Schemă cu contacte, o instrucțiune Control principal funcționează ca un comutator în bara de bus din stânga, ce trebuie închis pentru a fi executat următorul bloc de program.

Schemă cu contacte



Listă de instrucțiuni

0	LD	X001	
1	MC	N0	M10
4	LD	X002	
5	OUT	Y003	
6	LD	X003	
7	OUT	Y004	
8	MCR	N0	
10	LD	X002	
11	AND	X004	
12	OUT	M155	

În exemplul de mai sus, liniile de program dintre instrucțiunile MC și MCR sunt executate doar atunci când intrarea X001 este pornită.

Secțiunea de program ce trebuie executată poate fi specificată cu adresa de imbricare N0 – N7, ceea ce vă permite să introduceți mai multe instrucțiuni MC înainte de instrucțiunea MCR de închidere. (Pentru un exemplu de imbricare, a se vedea Manualul de programare FX). Adresarea unui dispozitiv Y sau M specifică un contact de închidere. Acest contact va activa secțiunea de program atunci când condiția de intrare pentru instrucțiunea MC are valoarea Adevărat.

În cazul în care condiția de intrare a instrucțiunii MC va avea valoarea Fals, stările dispozitivelor dintre instrucțiunile MC și MCR se vor modifica după cum urmează:

- Temporizatoarele cu memorare, contoarele și dispozitivele ce sunt controlate cu instrucțiuni SET și RST își vor păstra starea curentă.
- Temporizatoarele fără memorare și dispozitivele ce sunt controlate cu instrucțiuni OUT sunt resetate.

(Pentru detalii despre temporizatoare și contoare, a se vedea capitolul 4).

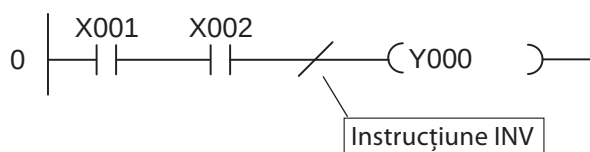
3.4.12 Inversarea rezultatului unei operații

Instrucțiune	Funcție	Simbol	GX Developer FX
INV	Inversare, inversează rezultatul unei operații		

Instrucțiunea INV este utilizată simplu, fără alți operanzi. Ea inversează rezultatul operației care vine chiar înaintea sa.

- Dacă rezultatul acelei operații era "1", acesta va fi inversat în "0"
- Dacă rezultatul acelei operații era "0", acesta va fi inversat în "1".

Schemă cu contacte



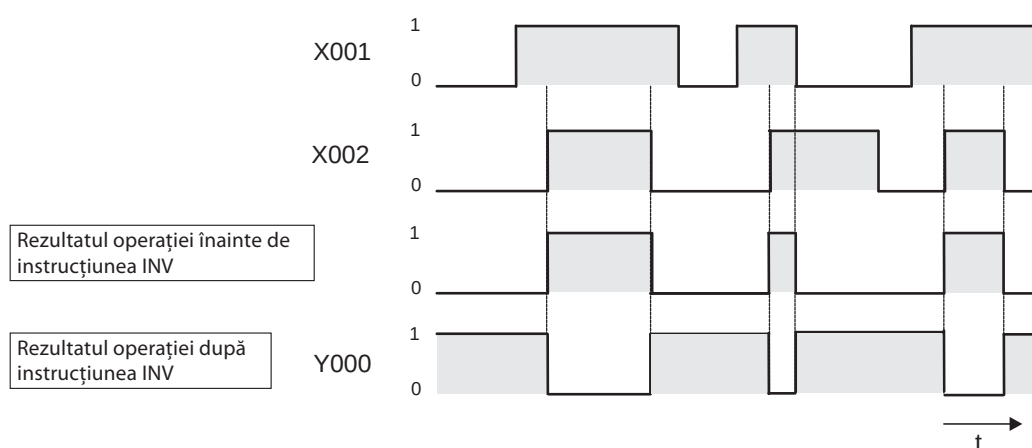
Listă de instrucțiuni

```

0 LD X001
1 AND X002
2 INV
3 OUT Y000

```

Exemplul de mai sus generează următoarea secvență de semnale:



Instrucțiunea INV poate fi utilizată atunci când trebuie să inversați rezultatul unei operații complexe. Aceasta poate fi utilizată în aceeași poziție ca instrucțiunile AND și ANI.

Instrucțiunea INV nu poate fi utilizată la începutul unei operații (unui circuit), în felul în care pot fi utilizate instrucțiunile LD, LDI, LDP sau LDF.

3.5 Nu uitați măsurile de siguranță!

Automatele programabile au multe avantaje față de controlerele cablate. Dar când vine vorba de siguranță, este important să înțelegeți că nu puteți avea încredere oarbă într-un automat programabil.

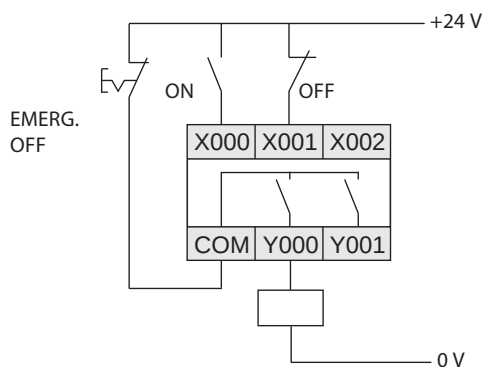
Dispozitive de oprire de urgență (STOP)

Este esențial să vă asigurați că erorile din sistemul sau programul de control nu pot duce la riscuri pentru personal sau utilaje. Dispozitivele de OPRIRE DE URGENȚĂ trebuie să rămână complet funcționale chiar și atunci când automatul programabil nu funcționează cum trebuie – de exemplu, pentru a întrerupe, dacă este necesar, alimentarea cu curent a automatului programabil.

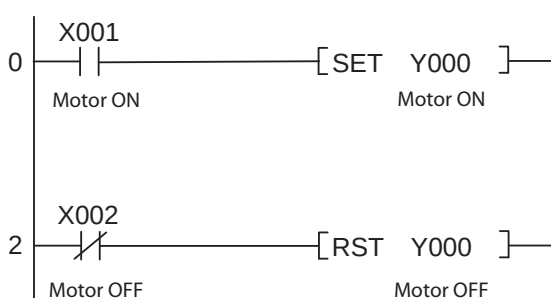
Nu implementați niciodată un comutator STOP de oprire de urgență doar ca intrare procesată de automatul programabil, cu închiderea activată de către programul automatului. Acest lucru ar fi mult prea riscant.

Măsurile de siguranță în cazul întreruperilor cablurilor

Trebuie de asemenea să luați măsuri pentru a asigura siguranța în cazul în care transmisia semnalelor de la comutatori la automatul programabil este întreruptă de defecțiuni ale cablurilor. Atunci când un echipament este pornit și oprit prin intermediul unui automat programabil, utilizați întotdeauna comutatori sau butoane cu contacte normal deschise pentru pornire și contacte normal închise pentru oprire.



În acest exemplu, contactorul pentru un sistem de acționare poate de asemenea să fie comutat manual în poziția închis, cu ajutorul unui comutator de Opre de urgență.



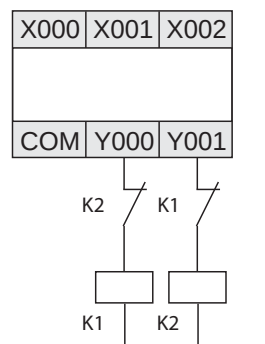
În programul pentru această instalație, contactul normal deschis de pe comutatorul ON este interogată cu o instrucțiune LD, contactul normal închis de pe comutatorul OFF este interogată cu o instrucțiune LDI. Ieșirea, și deci și acționarea, este închisă când intrarea X002 are starea de semnal "0". Aceasta este situația în cazul în care este operat comutatorul OFF sau când conexiunea dintre comutator și intrarea X002 este întreruptă.

Acest lucru asigură faptul că, în cazul unei defecțiuni a unui cablu, acționarea este închisă automat și nu poate fi activată. În plus, închiderea are prioritate deoarece este procesată de către program după instrucțiunea de pornire.

Contacte de interblocare

Dacă aveți două contacte care nu trebuie să fie niciodată pornite simultan – de exemplu ieșiri pentru selectarea operării înainte sau înapoi a unui motor – trebuie implementată o interblocare pentru ieșiri, cu contacte fizice în contactorii controlați de automatul programabil. Acest lucru este necesar deoarece în program poate fi utilizată o singură interblocare internă, iar o eroare în automatul programabil poate duce la activarea simultană a ambelor ieșiri.

Exemplul din dreapta arată o astfel de interblocaje cu contacte de contactori. Aici, este fizic imposibil ca cei doi contactori K1 și K2 să fie comutați pe pornit simultan.



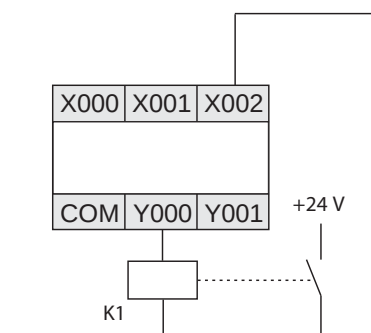
Închiderea automată

Atunci când un automat programabil este utilizat pentru a controla secvențe de mișcare în care pot apărea riscuri dacă respectivele componente se mișcă dincolo de anumite puncte, trebuie instalați comutatori de limite de cursă suplimentari, pentru a întrerupe automat mișcarea. Acești comutatori trebuie să funcționeze direct și independent de automatul programabil. A se vedea capitolul 3.6.2 pentru un exemplu al unei astfel de instalări de închidere automată.

Feedback-ul semnalului de ieșire

În general, ieșirile automatelor programabile nu sunt monitorizate. Atunci când este activată o ieșire, programul presupune că răspunsul corect a avut loc în afara automatului programabil. În majoritatea cazurilor, nu sunt necesare instalații suplimentare. Însă în cazul aplicațiilor critice, trebuie să monitorizați semnalele de ieșire și cu automatul programabil – de exemplu atunci când erorile din circuitul de ieșire (cabluri rupte, contacte defecte) ar putea avea consecințe grave asupra siguranței sau funcționării sistemului.

În exemplul din dreapta, un contact normal deschis din contactorul K1 comută intrarea X002 la pornit atunci când ieșirea Y000 este activată. Acest lucru permite programului să monitorizeze funcționarea corectă a ieșirii și a contactorului conectat. Rețineți că această soluție simplă nu verifică dacă echipamentul comutat funcționează corect (de exemplu, dacă un motor chiar se rotește în realitate). Pentru a verifica acest lucru, ar fi necesare funcții suplimentare, de exemplu un senzor de viteză sau un traductor de tensiune pentru sarcină.



3.6 Exemple aplicații PLC

Automatele programabile oferă un număr aproape nelimitat de moduri de legare a intrărilor cu ieșirile. Sarcina dumneavoastră este să alegeți instrucțiunile potrivite din cele suportate de automatele programabile ale familiei MELSEC FX, pentru a programa o soluție adecvată aplicației dumneavoastră.

Acest capitol oferă două exemple simple ce demonstrează modul de dezvoltare al unei aplicații pentru un automat programabil, de la definirea sarcinii la programul complet.

3.6.1 Un sistem de alarmă

Primul pas este dezvoltarea unui concept clar a ceea ce doriți să obțineți. Aceasta înseamnă că trebuie să abordați problema ascendent, de jos în sus, și să descrieți cât mai detaliat ce doriți să facă automatul programabil.

Descrierea sarcinii

Obiectivul este de a crea un sistem de alarmă cu mai multe circuite de alarmă și o funcție de întârziere pentru activarea și dezactivarea sistemului.

- Sistemul va fi activat de un comutator principal, cu o întârziere de 20 de secunde între acționarea comutatorului și activare. Acest lucru va oferi destul timp utilizatorilor pentru a părăsi casa fără a declanșa alarma. În timpul acestei perioade de întârziere, un afișaj va arăta dacă circuitele de alarmă sunt închise.
- Dacă unul din circuite este întrerupt, se va declanșa o alarmă (sistem cu circuit închis, alarma este declanșată și dacă circuitul este sabotat). În plus, dorim să știm ce circuit a declanșat alarma.
- Atunci când este declanșată o alarmă, se va activa o lumină intermitentă și o sirenă, după un interval de 10 secunde. (Alarmerle acustice și vizuale sunt activate după un interval de timp pentru ca sistemul să poată fi dezarmat la intrarea în casă. Tot din acest motiv, dorim să avem un indicator luminos special care să arate că sistemul este armat).
- Sirena va suna doar timp de 30 de secunde, dar lumina intermitentă va rămâne activă până la dezarmarea sistemului.
- Pentru dezactivarea sistemului de alarmă, se va putea utiliza și un comutator operat de o cheie.

Atribuirea intrărilor și ieșirilor

Următorul pas este definirea semnalelor de intrare și de ieșire pe care trebuie să le procesăm. Pe baza specificațiilor, știm că vom avea nevoie de un comutator operat de o cheie și de 4 lumini de alarmă. În plus, vom avea nevoie de cel puțin 3 intrări pentru circuitele de alarmă și 2 ieșiri pentru sirenă și lumina intermitentă de alarmă. Aceasta rezultă într-un total de 4 intrări și 6 ieșiri. Apoi vom atribui aceste semnale intrărilor și ieșirilor automatului programabil:

Funcție		Nume	Adresă	Observații
Intrare	Armare sistem	S1	X1	Contact normal deschis (operat de o cheie)
	Circuit de alarmă 1	S11, S12	X2	Contacte normal închise (este declanșată o alarmă atunci când intrarea are starea de semnal "0")
	Circuit de alarmă 2	S21, S22	X3	
	Circuit de alarmă 3	S31, S32	X4	
Ieșire	Afișarea mesajului "sistem armat"	H0	Y0	Funcțiile de ieșire sunt activate atunci când ieșirile corespunzătoare sunt activate. De exemplu, dacă este setată Y1, va suna semnalul de alarmă
	Semnal de alarmă (sirenă)	E1	Y1	
	Alarmă optică (lumină ce se rotește)	H1	Y2	
	Afișare circuit 1 alarmă	H2	Y3	
	Afișare circuit 2 alarmă	H3	Y4	
	Afișare circuit 3 alarmă	H4	Y5	

Programarea

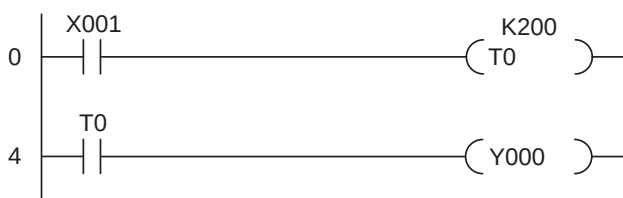
Acum putem începe să scriem programul. De obicei, numai după ce veți începe programarea efectivă vă veți da seama dacă vor fi necesare dispozitive cu relee și dacă da, câte vor fi necesare. Ceea ce este sigur în acest proiect este faptul că vom avea nevoie de trei temporizatoare pentru funcțiile importante. Dacă am fi utilizat un controler cablat, am fi utilizat relee cu temporizator pentru aceste funcții. Într-un automat programabil, aveți temporizatoare electronice programabile (a se vedea secțiunea 4.3). Aceste temporizatoare pot de asemenea să fie definite înainte să începem să programăm:

Funcție	Adresă	Observații
Temporizator	Întârziere armare	T0
	Întârziere declanșare alarmă	T1
	Durată de activare sirenă	T2

În continuare, putem programa sarcinile de control individuale .

● Armarea cu întârziere a sistemului de alarmă

Schemă cu contacte



Listă de instrucțiuni

```

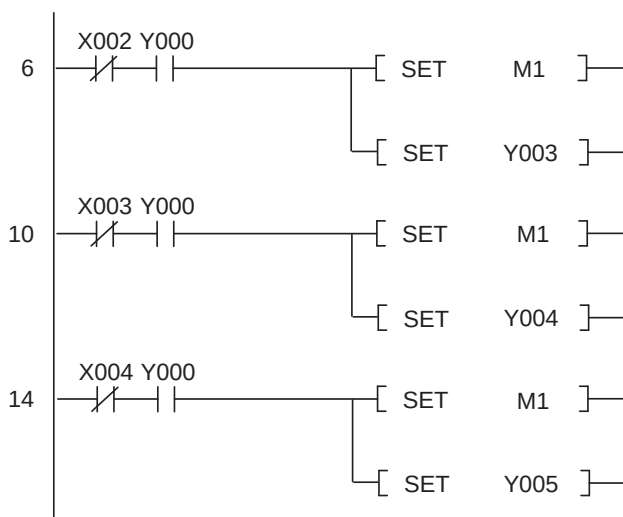
0  LD    X001
1  OUT   T0      K200
4  LD    T0
5  OUT   Y000

```

Atunci când comutatorul operat de cheie trece în poziția ON, începe să se scurgă timpul de întârziere, implementat cu temporizatorul T0. După 20 de secunde ($K200 = 200 \times 0.1 \text{ s} = 20 \text{ sec.}$), indicatorul luminos conectat la ieșirea Y000 se va aprinde, ceea ce va indica faptul că sistemul e armat.

● Monitorizarea circuitelor de alarmă și declanșarea semnalului de alarmă

Schemă cu contacte



Listă de instrucțiuni

```

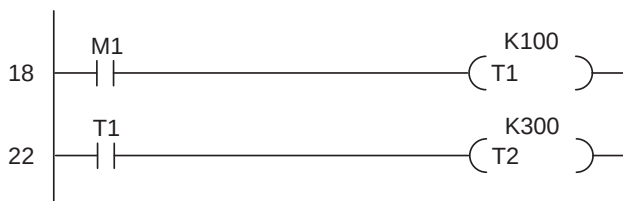
6  LDI   X002
7  AND   Y000
8  SET   M1
9  SET   Y003
10 LDI   X003
11 AND   Y000
12 SET   M1
13 SET   Y004
14 LDI   X004
15 AND   Y000
16 SET   M1
17 SET   Y005

```

Ieșirea Y000 este interogată în această rutină pentru a se verifica dacă sistemul de alarmă este armat. Puteți de asemenea să utilizați un releu aici, care ar fi apoi setat și resetat simultan cu ieșirea Y000. O întrerupere a circuitului de alarmă va seta releul M1 (ceea ce va indica faptul că alarma a fost declanșată) doar dacă sistemul de alarmă este armat efectiv. În plus, ieșirile Y003 – Y005 sunt utilizate pentru a indica ce circuit de alarmă a declanșat alarma. Releul M1 și ieșirea corespundentă a circuitului de alarmă vor rămâne setate chiar și când circuitul de alarmă este închis la loc.

● **Întârzierea activării alarmei**

Schemă cu contacte



Listă de instrucțiuni

```

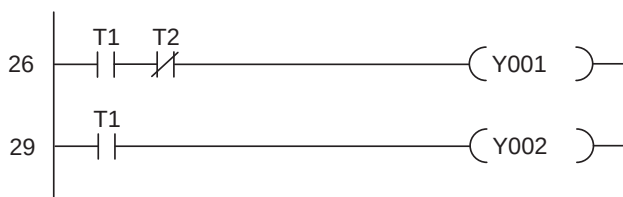
18 LD    M1
19 OUT   T1    K100
22 LD    T1
23 OUT   T2    K300

```

Atunci când este declanșată o alarmă (M1 este comutat la starea "1"), începe temporizatorul de întârziere de 10 secunde. După cele 10 secunde, T1 va porni temporizatorul T2, care este setat la 30 de secunde, iar apoi va începe durata de activare a sirenei.

● **Afișajul alarmei (activarea sirenei și a lumii intermitente)**

Schemă cu contacte



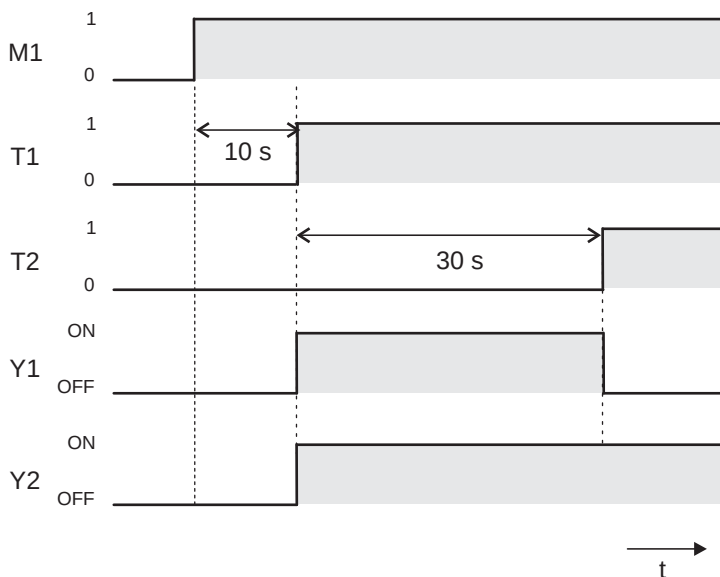
Listă de instrucțiuni

```

26 LD    T1
27 ANI   T2
28 OUT   Y001
29 LD    T1
30 OUT   Y002

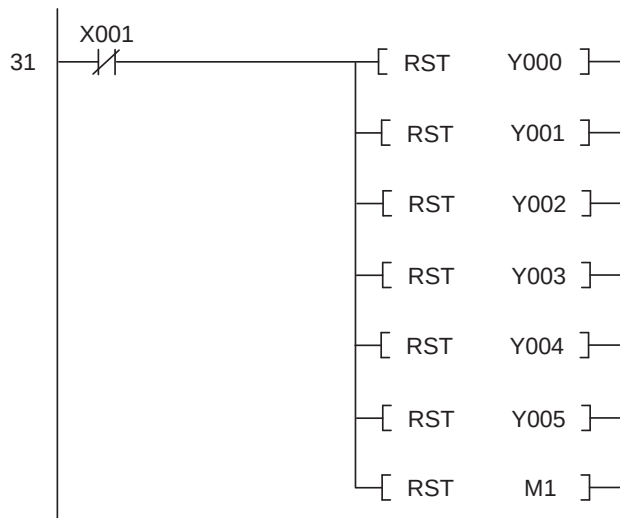
```

Sirena va fi activată după intervalul de întârziere de 10 secunde (T1) și va rămâne activată până la activarea temporizatorului T2. La finalul perioadei de activare de 30 de secunde (T2), sirena se va dezactiva. Lumina intermitentă este de asemenea pornită după întârzierea de 10 secunde. Următoarea ilustrație arată secvența semnalelor generate de această secțiune a programului:



● Resetarea tuturor ieșirilor și a releului

Schemă cu contacte



Listă de instrucțiuni

```

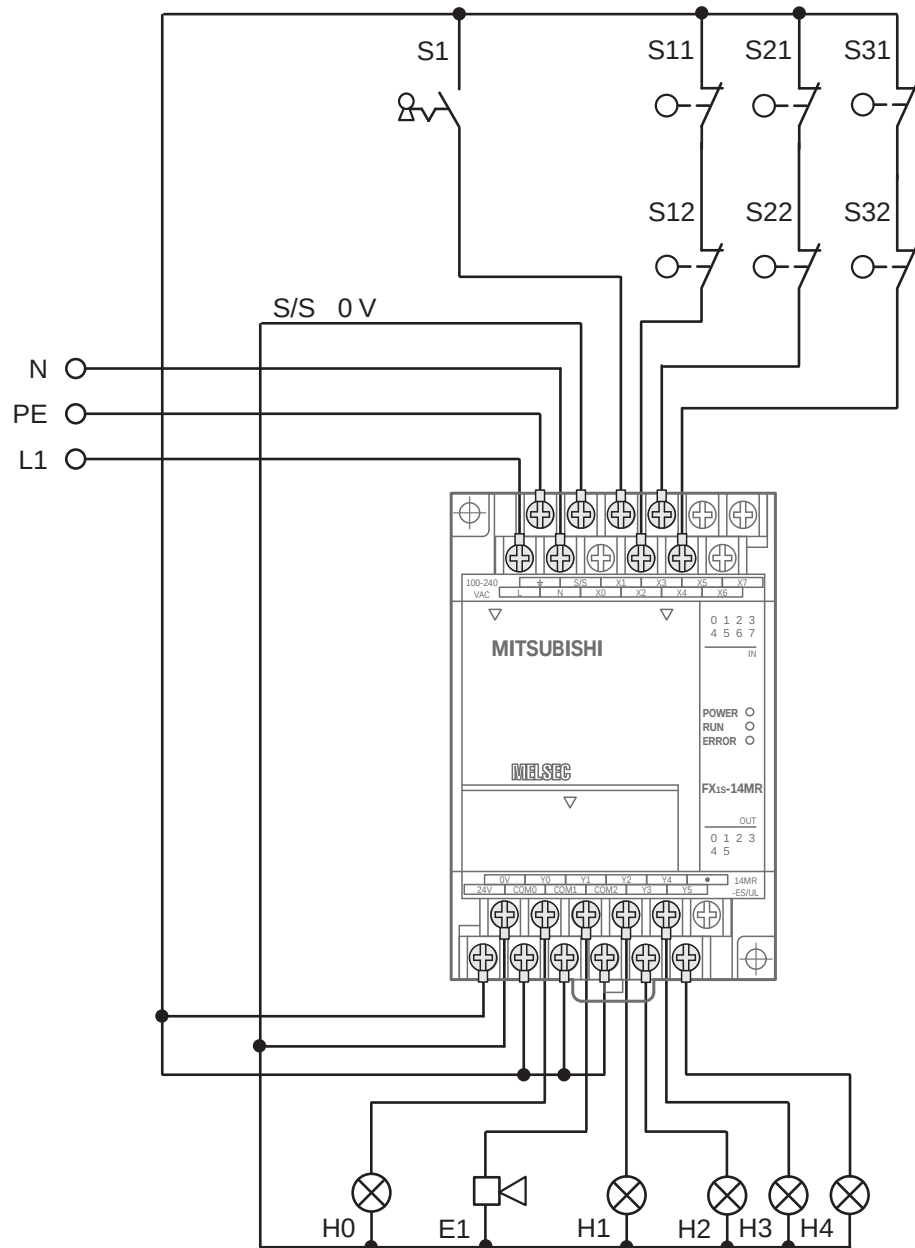
31 LDI    X001
32 RST    Y000
33 RST    Y001
34 RST    Y002
35 RST    Y003
36 RST    Y004
37 RST    Y005
38 RST    M1

```

Atunci când sistemul de alarmă este dezactivat cu comutatorul cu cheie, toate ieșirile utilizate de către program și de către releul M1 sunt resetate. Dacă a fost declanșată o alarmă, va fi afișat circuitul de alarmă întrerupt care a fost eliberat până la decuplarea sistemului.

Conectarea automatului programabil

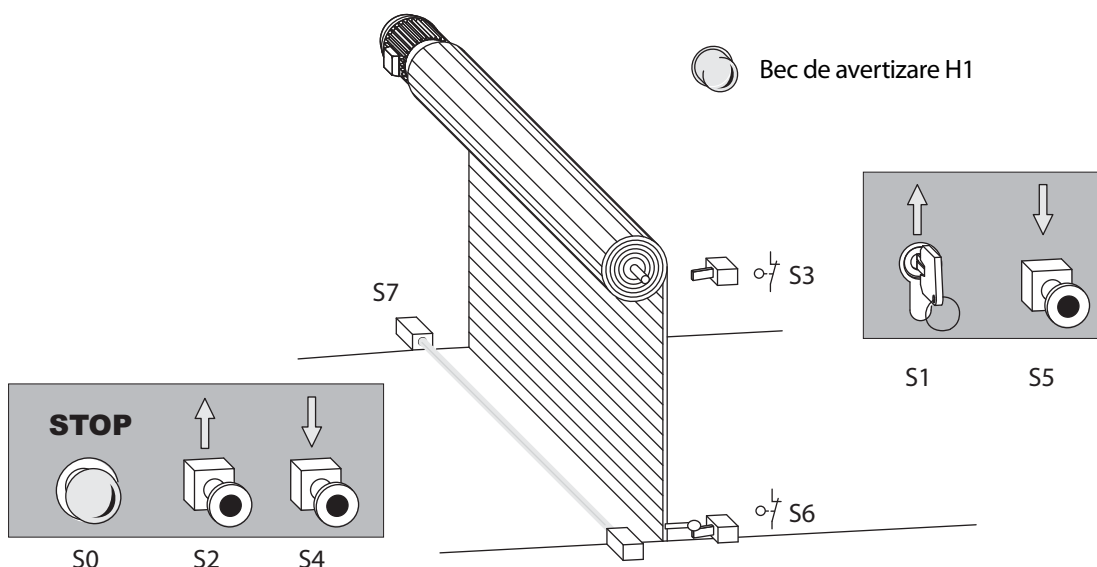
Schița de mai jos arată cât de ușor de implementat este acest sistem de alarmă, cu un automat programabil din seria FX. Acest exemplu este creat utilizându-se modelul FX1N-14MR.



3.6.2 O ușă automată de tip rulou

Descrierea sarcinii

Dorim să implementăm un sistem de control pentru ușa unui spațiu de depozitare, care să fie de tip rulou, automată, și să permită operarea simplă dinăuntru și din afara depozitului. În sistem trebuie de asemenea integrate dispozitive de siguranță.



● Funcționare

- Trebuie să fie posibilă deschiderea ușii din afară, cu comutatorul cu cheie S1, precum și închiderea acesteia cu butonul S5. Înăuntru depozitului, trebuie să fie posibilă deschiderea ușii cu butonul S2 și închiderea acesteia cu S4.
- Un comutator suplimentar cu temporizator trebuie să închidă poarta automat dacă aceasta rămâne deschisă mai mult de 20 de secunde.
- Stările "poartă în mișcare" și "poartă în poziție nedefinită" trebuie să fie indicate de o lumină intermitentă de avertizare.

● Instalații de siguranță

- Trebuie instalat un buton de oprire (S0) care să poată opri imediat mișcarea ușii, oprind ușa în poziția curentă. Comutatorul de oprire nu este însă o funcție de oprire de urgență! Semnalul de comutare este doar procesat de către automatul programabil și nu comută nici una din conexiunile externe la rețeaua de alimentare.
- Trebuie instalată o barieră fotoelectrică (S7) care să identifice eventualele obstacole din calea ușii. Dacă bariera va înregistra un obstacol în timp ce ușa se închide, ușa trebuie să se deschidă automat.
- Trebuie instalați doi comutatoare de limitare, pentru a opri motorul ușii atunci când ajunge în pozițiile "complet deschisă" (S3) și "complet închisă" (S6).

Atribuirea intrărilor și ieșirilor

Descrierea sarcinii definește clar numărul de intrări și de ieșiri necesare. Motorul de acționare al porții este controlat cu ajutorul a două ieșiri. Semnalele necesare sunt atribuite astfel intrărilor și ieșirilor automatului programabil:

Funcție		Nume	Adresă	Observații
Intrări	Buton STOP	S0	X0	Contact normal închis (când comutatorul este operat, X0 este "0" și ușa se oprește)
	Comutator Deschidere operat cu cheie (afară)	S1	X1	Contacte normal deschise
	Buton Deschidere (înăuntru)	S2	X2	
	Comutator limită superioară (ușă deschisă)	S3	X3	Contact normal închis (X2 este 0 când poarta este sus și S3 este activat)
	Buton Închidere (înăuntru)	S4	X4	Contacte normal deschise
	Buton Închidere (afară)	S5	X5	
	Comutator limită inferioară (ușă închisă)	S6	X6	Contact normal închis (X6 este "0" când ușa este jos și S6 este activat)
	Barieră fotoelectrică	S7	X7	X7 este setat la "1" când se detectează un obstacol
Ieșiri	Lumină de avertizare	H1	Y0	—
	Contactor motor (motor înapoi)	K1	Y1	Înapoi – Deschidere ușă
	Contactor motor (motor înainte)	K2	Y2	Înainte – Închidere ușă
Temporizator	Întârziere pentru închiderea automată	—	T0	Durată: 20 secunde

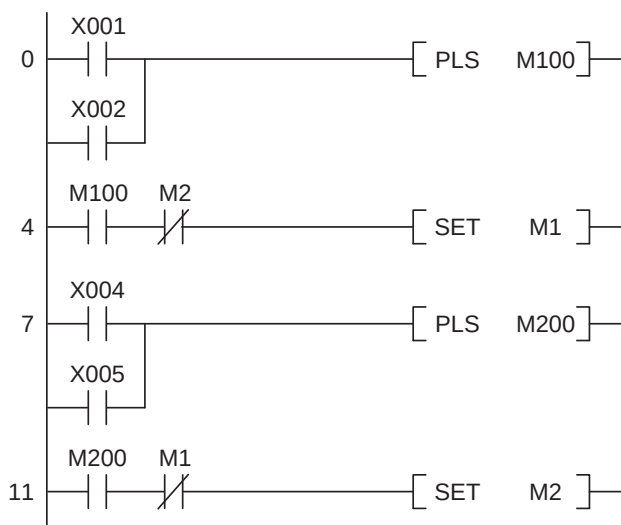
Componentele programului

- Operarea ușii automate cu role cu ajutorul butoanelor

Programul trebuie să convertească semnalele de intrare pentru operarea porții în două comenzi pentru motorul de acționare: "Deschidere poartă" și "Închidere poartă". Deoarece acestea sunt semnale de la butoane ce sunt disponibile doar pentru scurt timp la intrări, ele trebuie stocate. În acest scop, vom utiliza două relee pentru a reprezenta intrările din program și pentru a le seta și reseta după necesități:

- M1: deschiderea porții
- M2: închiderea porții

Schemă cu contacte



Listă de instrucțiuni

```

0 LD    X001
1 OR    X002
2 PLS   M100
4 LD    M100
5 ANI   M2
6 SET   M1
7 LD    X004
8 OR    X005
9 PLS   M200
11 LD   M200
12 ANI  M1
13 SET  M2
  
```

Semnalele pentru deschiderea porții sunt procesate primele: Atunci când sunt operate comutatorul cu cheie S1 sau butonul S2, este generat un semnal și releul M001 este setat la starea de semnal "1"

pe durata unui singur ciclu de program. Acest lucru asigură faptul că poarta nu va putea fi blocată dacă butonul se înțepenește sau dacă operatorul nu îl eliberează imediat.

Trebuie să ne asigurăm de faptul că motorul de acționare poate fi comutat în poziția pornit doar atunci când nu se rotește deja în sensul opus.

Acest lucru este implementat prin programarea automatului astfel încât M1 să poată fi setat doar atunci când M2 nu este setat.

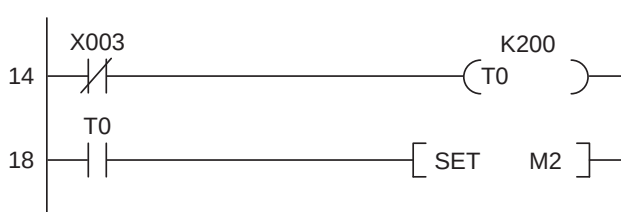
NOTĂ

Dispozitivul de interblocare al sensului de rotire al motorului trebuie de asemenea suplimentat de un dispozitiv de interblocare fizic suplimentar, cu contactori fizici, aflat în afara automatului programabil (a se vedea diagrama de cablare).

O abordare similară este utilizată pentru procesarea semnalelor de la butoanele S4 și S5, pentru închiderea porții. Aici, releul M1 este interogată pentru a se vedea dacă are starea de semnal "0", pentru a se asigura faptul că M1 și M2 nu pot fi ambele setate simultan.

● Închidere automată după 20 de secunde

Schemă cu contacte



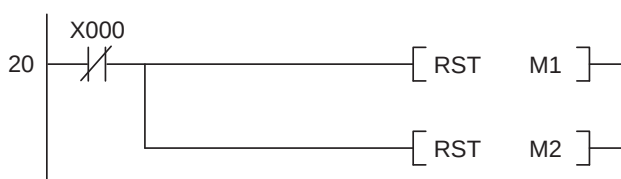
Listă de instrucțiuni

14	LDI	X003	
15	OUT	T0	K200
18	LD	T0	
19	SET	M2	

Atunci când poarta este deschisă, comutatorul de limitare S3 se activează, iar intrarea X3 este comutată în poziția închis. (Din motive de siguranță, S3 este un contact normal închis). Atunci când se întâmplă acest lucru, temporizatorul T0 va porni cronometrarea perioadei de întârziere de 20 de secunde ($K200 = 200 \times 0.1 \text{ s} = 20\text{s}$). Atunci când temporizatorul ajunge la 20 de secunde, este setat releul M2 și poarta este închisă.

● Oprirea porții cu comutatorul de oprire

Schemă cu contacte



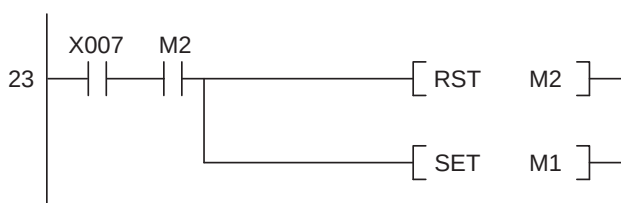
Listă de instrucțiuni

20	LDI	X000
21	RST	M1
22	RST	M2

Apăsarea butonului STOP (S0) va reseta relele M1 și M2, oprind motorul porții.

● Identificarea obstacolelor cu ajutorul barierei fotoelectrice

Schemă cu contacte



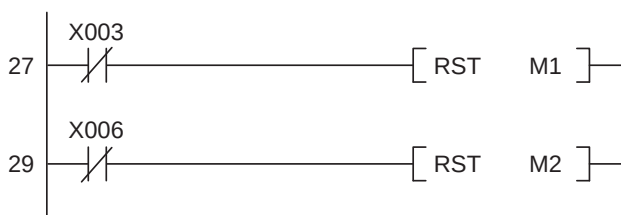
Listă de instrucțiuni

23	LD	X007
24	AND	M2
25	RST	M2
26	SET	M1

Dacă bariera fotoelectrică înregistrează un obstacol în timp ce poarta se închide, releul M2 este resetat și operația de închidere este întreruptă. După aceea, este setat releul M1, iar poarta va începe să se deschidă din nou.

● Deconectarea motorului cu comutatorii de limitare

Schemă cu contacte



Listă de instrucțiuni

```

27 LDI    X003
28 RST    M1
29 LDI    X006
30 RST    M2

```

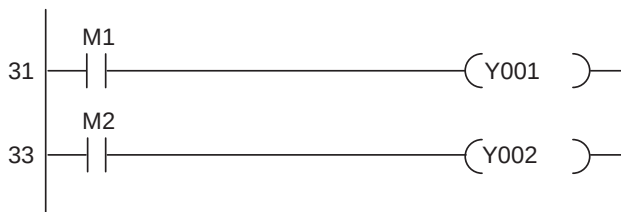
Atunci când poarta este deschisă, comutatorul de limitare S3 este activat, iar intrarea X3 este comutată în poziția închis. Aceasta va duce la resetarea releului M1, ceea ce va opri motorul. Atunci când poarta este complet închisă, S6 este activat, X6 este comutat în poziția oprit, iar M2 este resetat, oprind motorul. Din motive de siguranță, comutatorii de limitare sunt contacte normal închise. Acest lucru asigură faptul că motorul este de asemenea oprit automat (sau că nu poate fi pornit) dacă este întreruptă conexiunea dintre comutator și intrare.

NOTĂ

Comutatorii de limitare trebuie să fie cablați astfel încât să oprească motorul automat chiar și fără automatul programabil.

● Controlul motorului

Schemă cu contacte



Listă de instrucțiuni

```

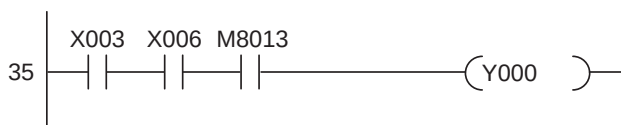
31 LD     M1
32 OUT    Y001
33 LD     M2
34 OUT    Y002

```

La finalul programului, stările de semnal ale releelor M1 și M2 sunt transferate ieșirilor Y001 și Y002.

● Lumina de avertizare: "Poartă în mișcare" și "Poartă în poziție nedefinită"

Schemă cu contacte



Listă de instrucțiuni

```

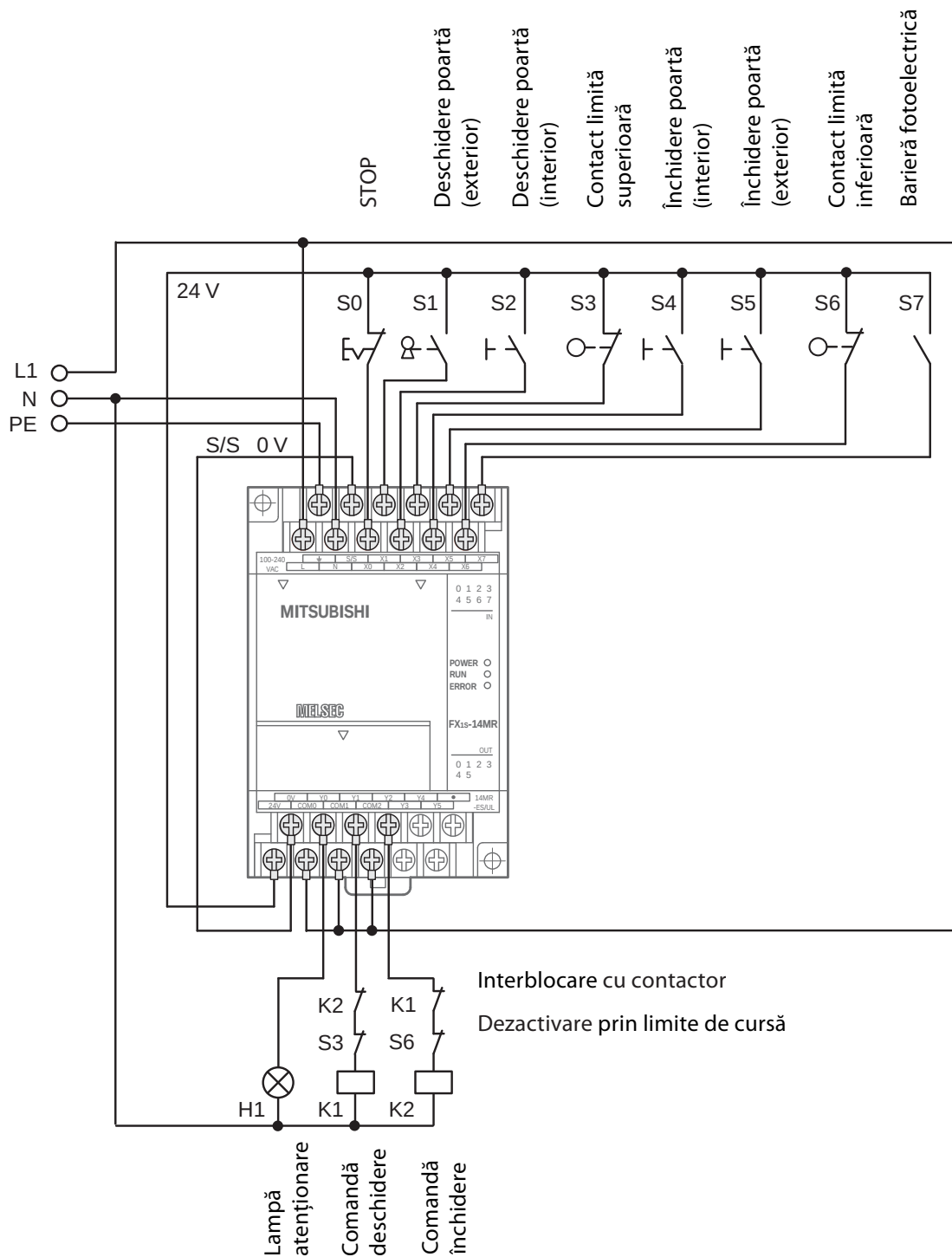
35 LD     X003
36 AND    X006
37 AND    M8013
38 OUT    Y000

```

Dacă nu este activat nici unul din comutatorii de limitare, acest lucru înseamnă că poarta este în curs de a fi deschisă sau închisă, sau că s-a oprit într-o poziție intermediară. În oricare dintre aceste situații, lumina de avertizare va clipi intermitent. Viteza clipirii intermitente este controlată de releul special M8013, care este setat și resetat automat la intervale de 1 secundă (a se vedea capitolul 4.2).

Conectarea automatului programabil

Sistemul de control al ușii automate cu role poate fi implementat cu un automat programabil de tipul FX1N-14MR.



4 Prezentarea detaliată a dispozitivelor

Dispozitivele din automatele programabile sunt utilizate pentru controlul direct al instrucțiunilor din program. Stările lor de semnal pot fi citite și modificate de către programul din automatul programabil. Un dispozitiv are două părți:

- numele dispozitivului și
- adresa dispozitivului.

Exemplu de dispozitiv (de ex. intrare 0):

Nume dispozitiv  Adresă dispozitiv

4.1 Intrări și ieșiri

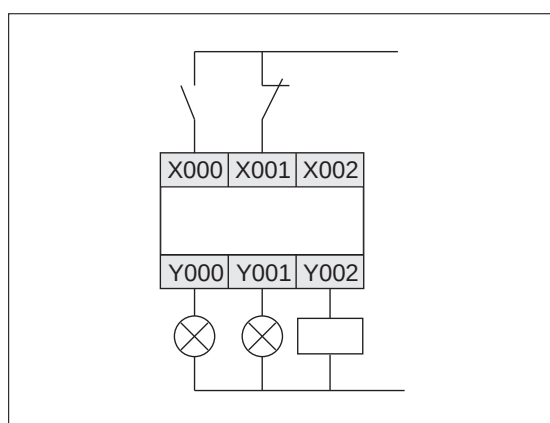
Intrările și ieșirile automatului programabil îl conectează pe acesta la procesul controlat. Atunci când programul automatului interoghează o intrare, este măsurată tensiunea din terminalul intrării, de pe automat. Deoarece aceste intrări sunt digitale, ele pot avea doar două stări de semnal: ON și OFF. Atunci când tensiunea la terminalul intrării atinge 24V, intrarea este deschisă, adică ON (starea "1"). Dacă tensiunea este sub 24V, intrarea este evaluată ca fiind închisă (stare de semnal "0").

În automatele programabile MELSEC, pentru intrări se utilizează identificatorul "X". Aceeași intrare poate fi interogată de câte ori este necesar, în cadrul aceluiași program.

NOTĂ

Automatul programabil nu poate modifica starea intrărilor. De exemplu, nu este posibilă execuția unei instrucțiuni OUT asupra unui dispozitiv de intrare.

Dacă este executată o instrucțiune de ieșire asupra unei ieșiri, rezultatul operației curente (starea de semnal) este aplicat terminalului de ieșire al automatului programabil. Dacă aceasta este o ieșire pe releu, atunci releul se închide (toate releele au contacte de închidere). Dacă este o ieșire pe tranzistor, tranzistorul este comandat și se activează astfel circuitul conectat.



Ilustrația din stânga arată un exemplu al modului în care puteți conecta comutatori la intrări și becuri și contactori la ieșirile unui automat programabil MELSEC.

Identificatorul pentru dispozitivele de ieșire este "Y". Ieșirile pot fi utilizate în instrucțiuni cu operații logice și în instrucțiuni de ieșire. Este însă important să rețineți că nu puteți niciodată utiliza o instrucțiune de ieșire asupra aceleiași ieșiri decât o singură dată (a se vedea și secțiunea 3.4.2).

Următorul tabel oferă o prezentare generală a intrărilor și ieșirilor automatelor programabile din seria MELSEC.

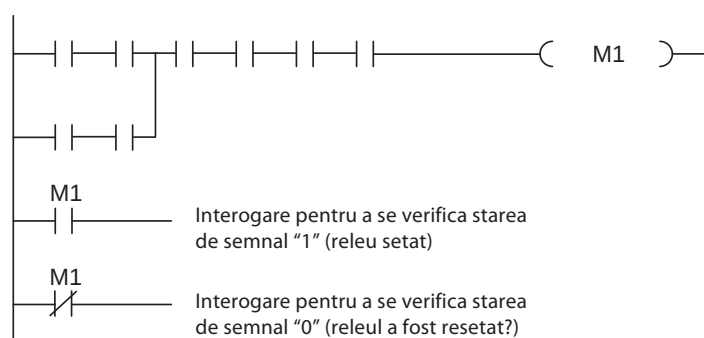
Dispozitiv		Intrări	Ieșiri
Identificator dispozitiv		X	Y
Tip dispozitiv		Dispozitiv bit	
Valori posibile		0 sau 1	
Formatul adresei dispozitivului		Octal	
Nr. de dispozitive și de adrese. (în funcție de tipul unității de bază a automatului programabil)	FX1S	6 (X00–X05) 8 (X00–X07) 12 (X00–X07, X10, X11, X12, X13) 16 (X00–X07, X10–X17)	4 (Y00–Y03) 6 (Y00–Y05) 8 (Y00–Y07) 14 (Y00–Y07, Y10–Y15)
	FX1N	8 (X00–X07) 14 (X00–X07, X10–X15) 24 (X00–X07, X10–X17, X20–X27) 36 (X00–X07, X10–X17, X20–X27, X30–X37, X40, X41, X42, X43) Numărul total de intrări poate fi crescut la maximum 84 (X123), cu ajutorul modulelor de extensie. Cu toate acestea, suma totală a intrărilor și ieșirilor nu poate depăși 128.	6 (Y00–Y05) 10 (Y00–Y07, Y10, Y11) 16 (Y00–Y07, Y10–Y17) 24 (Y00–Y07, Y10–Y17, Y20–Y27) Numărul total de intrări poate fi crescut la maximum 64 (Y77), cu ajutorul modulelor de extensie. Cu toate acestea, suma totală a intrărilor și ieșirilor nu poate depăși 128.
	FX2N	8 (X00–X07) 16 (X00–X07, X10–X17) 24 (X00–X07, X10–X17, X20–X27) 32 (X00–X07, X10–X17, X20–X27, X30–X37) 40 (X00–X07, X10–X17, X20–X27, X30–X37, X40–X47) 64 (X00–X07, X10–X17, X20–X27, X30–X37, X40–X47, X50–X57, X60–X67, X70–X77)	8 (Y00–Y07) 16 (Y00–Y07, Y10–Y17) 24 (Y00–Y07, Y10–Y17, Y20–Y27) 32 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37) 40 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37, Y40–Y47) 64 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37, Y40–Y47, Y50–Y57, Y60–Y67, Y70–Y77)
	FX2NC	8 (X00–X07) 16 (X00–X07, X10–X17) 32 (X00–X07, X10–X17, X20–X27, X30–X37) 48 (X00–X07, X10–X17, X20–X27, X30–X37, X40–X47, X50–X57)	8 (Y00–Y07) 16 (Y00–Y07, Y10–Y17) 32 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37) 48 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37, Y40–Y47, X50–X57)
	FX3U	8 (X00–X07) 16 (X00–X07, X10–X17) 24 (X00–X07, X10–X17, X20–X27) 32 (X00–X07, X10–X17, X20–X27, X30–X37) 40 (X00–X07, X10–X17, X20–X27, X30–X37, X40–X47) 64 (X00–X07, X10–X17, X20–X27, X30–X37, X40–X47, X50–X57, X60–X67, X70–X77)	8 (Y00–Y07) 16 (Y00–Y07, Y10–Y17) 24 (Y00–Y07, Y10–Y17, Y20–Y27) 32 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37) 40 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37, Y40–Y47) 64 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37, Y40–Y47, Y50–Y57, Y60–Y67, Y70–Y77)

* Numărul total de intrări poate fi crescut la maximum 248 (X367), cu ajutorul modulelor de extensie. Cu toate acestea, suma totală a intrărilor și ieșirilor nu poate depăși 256.

4.2 Relee

În programele pentru automatul programabil, va fi adesea necesar să stocați temporar rezultate binare intermediare (o stare de semnal, "0" sau "1"), pentru a face referință la ele mai târziu. Automatul programabil are celule de memorie speciale, disponibile în acest scop și denumite "relee auxiliare", sau "relee" pe scurt (dispozitive identificate prin: "**M**").

Puteți stoca rezultatul binar al unei operații într-un releu, de exemplu cu o instrucțiune de ieșire, iar apoi puteți utiliza acel rezultat în operații ulterioare. Releele ajută la simplificarea citirii programelor și reduc numărul pașilor de program: Puteți stoca într-un releu rezultatele operațiilor ce trebuie utilizate de mai multe ori, iar apoi îl puteți interoga cât de des este necesar, din alte părți din program.



În plus față de releele obișnuite, controlerele FX au de asemenea și relee cu memorare sau "latch". Releele normale, non latch, sunt toate resetate la starea de semnal "0" atunci când automatul programabil este deconectat de la sursa de alimentare, aceasta fiind și starea lor standard atunci când automatul este pornit. Spre deosebire de acestea, releele latch își păstrează stările curente atunci când automatul este deconectat de la sursa de alimentare și când acesta este conectat din nou la rețea.

Dispozitiv		Tipuri de rele	
		Relee non latch	Relee latch
Identificator dispozitiv		M	
Tip dispozitiv		Dispozitiv bit	
Valori posibile pentru un dispozitiv		0 sau 1	
Formatul adresei dispozitivului		Zecimal	
Numărul de dispozitive și de adrese	FX1S	384 (M0–M383)	128 (M384–M511)
	FX1N	384 (M0–M383)	1152 (M384–M1535)
	FX2N	500 (M0–M499) ^①	524 (M500–M1023) ^②
	FX2NC		2048 (M1024–M3071)
	FX3U	500 (M0–M499) ^①	524 (M500–M1023) ^② 6656 (M1024–M7679)

① Puteți configura aceste relele ca rele latch utilizând parametrii automatului programabil.

② Puteți configura aceste relele ca rele non latch utilizând parametrii automatului programabil.

4.2.1 Relee speciale

În plus față de releele pe care le puteți închide sau deschide cu programul automatului, există și o altă clasă de rele, cunoscute ca rele speciale, sau de diagnosticare. Aceste rele utilizează intervalul de adrese ce începe cu M8000. Unele dintre ele conțin informații privind starea sistemului, iar altele pot fi utilizate pentru a influența execuția programului. Următorul tabel arată câteva exemple de rele speciale disponibile.

Releu special	Funcție	Opțiuni de procesare din program
M8000	Atunci când automatul programabil este în modul RUN (pornit), acest releu este întotdeauna setat la "1".	Interogarea stării semnalului
M8001	Atunci când automatul programabil este în modul RUN (pornit), acest releu este întotdeauna setat la "0".	
M8002	Impuls de inițializare (după activarea modului RUN, acest releu este setat la "1" pe durata unui singur ciclu de program.	
M8004	Eroare automat programabil	
M8005	Tensiune scăzută baterie	
M8013	Impuls de semnal ceas: 1 secundă	
M8031	Resetează toate dispozitivele (cu excepția regiștrilor de date, D) ce nu sunt înregistrate ca latch (cu memorare).	Interogarea stării semnalului
M8034	Dezactivează ieșirile – ieșirile automatului programabil rămân închise, dar execuția programului este continuată.	Setarea stării semnalului

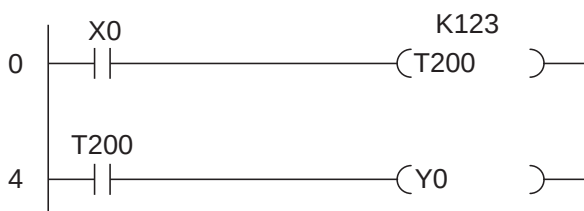
4.3 Temporizatoare

Atunci când controlați procese, veți dori adesea să programați o anumită întârziere înainte de a porni și a opri anumite operații. În controlerele cablate, acest lucru poate fi îndeplinit cu ajutorul releelor cu temporizator. În automatele programabile, acest lucru se efectuează cu ajutorul temporizatoarelor interne programabile.

Temporizatoarele sunt de fapt doar contoare ce numără semnalele ceasului intern al automatului programabil (de exemplu, impulsuri de 0,1 secunde). Atunci când valoarea contorului atinge valoarea de referință, ieșirea temporizatorului este comutată la starea pornit.

Toate temporizatoarele funcționează drept comutatori cu întârziere și sunt activate cu un semnal de stare "1". Pentru a porni și reseta temporizatoarele, trebuie să le programați în același mod ca și ieșirile. Puteți interoga ieșirile temporizatoarelor oricât de des doriți, pe parcursul programului.

Schemă cu contacte

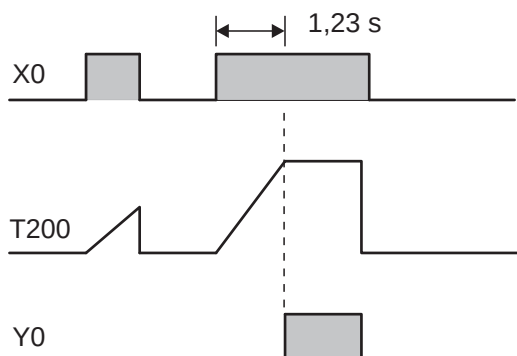


Listă de instrucțiuni

```

0 LD X0
1 OUT T200 K123
4 LD T200
5 OUT Y0
  
```

În exemplul de mai sus, temporizatorul T200 este pornit atunci când este pornită ieșirea X0. Valoarea de referință este $123 \times 10\text{ms} = 1,23$ secunde, astfel încât temporizatorul T200 va porni ieșirea Y0 după o întârziere de 1,23 secunde. Secvența semnalelor generate de următorul exemplu de programare este următoarea:



Temporizatorul continuă să numere impulsurile interne (10ms) atâta vreme cât X0 rămâne deschisă. Atunci când este atinsă valoarea de referință, este pornită ieșirea controlată de T200.

Dacă intrarea X0 sau sursa de alimentare a automatului programabil sunt oprite, temporizatorul este resetat, iar ieșirea controlată de acesta este de asemenea închisă.

Puteți de asemenea specifica valoarea de referință a temporizatorului în mod indirect, printr-o valoare în sistem zecimal stocată într-un registru de date. Pentru detalii, consultați secțiunea 4.6.1.

Temporizatoare cu memorare

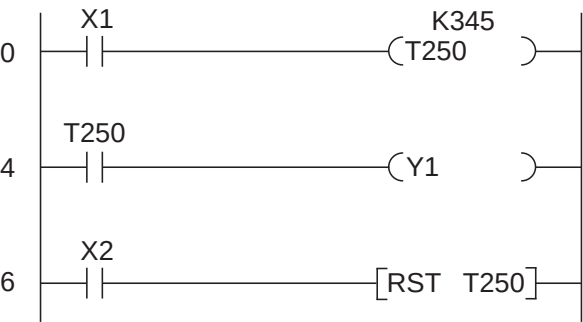
În plus față de temporizatoarele obișnuite descrise mai sus, controlerele din seria FX1N, FX2N, FX2NC și FX3U au și temporizatoare cu memorare, ce păstrează valoarea curentă a contorului de timp chiar dacă dispozitivul ce le controlează este oprit.

Valoarea curentă a contorului temporizatorului este stocată într-o memorie ce este păstrată chiar și în cazul unei căderi de tensiune.

Exemplu de program ce utilizează un temporizator cu memorie:

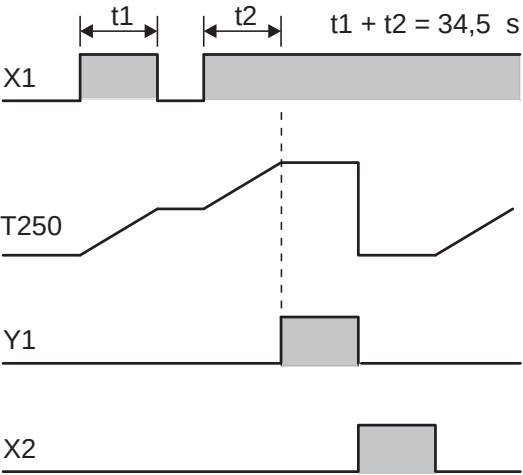
Schemă cu contacte

Listă de instrucțiuni



0	LD	X0
1	OUT	T250 K345
4	LD	T250
5	OUT	Y1
6	LD	X2
7	RST	T250

Temporizatorul T250 este pornit atunci când este închisă intrarea X0. Valoarea de referință este 345 x 0,1 sec = 34,5sec. Atunci când este atinsă valoarea de referință, T250 comută ieșirea Y1 la închis. Intrarea X2 resetează temporizatorul și deschide ieșirea controlată de acesta.



Atunci când X1 este închisă, temporizatorul numără impulsurile interne (100ms). Atunci când X1 se deschide, valoarea curentă a contorului de timp este păstrată. Ieșirea controlată de temporizator se închide atunci când valoarea curentă atinge valoarea de referință a temporizatorului.

Trebuie programată o instrucțiune separată pentru resetarea temporizatorului, deoarece acesta nu este resetat la deschiderea intrării X1 sau la deconectarea de la sursa de alimentare a automatului programabil. Intrarea X2 resetează temporizatorul T250 și deschide ieșirea controlată de acesta.

Temporizatoarele unităților de bază ale seriei MELSEC FX

Dispozitiv			Tipuri de temporizatoare	
			Temporizatoare obișnuite	Temporizatoare cu memorie
Identificator dispozitiv			T	
Tip dispozitiv (pentru setare și interogare)			Dispozitiv bit	
Valori posibile (ieșire temporizator)			0 sau 1	
Formatul adresei dispozitivului			Zecimal	
Introducerea valorii de referință a temporizatorului			Se introduce ca număr întreg în baza 10. Valoarea de referință poate fi setată fie direct, în instrucțiune, fie indirect, într-un registru de date.	
Numărul de dispozitive și adrese	FX1S	100 ms (Interval între 0,1 și 3276,7 s)	63 (T0–T62)	—
		10 ms (Interval între 0,01 și 327,67 s)	31 (T32–T62)*	—
		1 ms (Interval între 0,001 și 32,767 s)	1 (T63)	—
	FX1N	100 ms (Interval între 0,1 și 3276,7 s)	200 (T0–T199)	6 (T250–T255)
		10 ms (Interval între 0,01 și 327,67 s)	46 (T200–T245)	—
		1 ms (Interval între 0,001 și 32,767 s)	4 (T246–T249)	—
	FX2N FX2NC	100 ms (Interval între 0,1 și 3276,7 s)	200 (T0–T199)	6 (T250–T255)
		10 ms (Interval între 0,01 și 327,67 s)	46 (T200–T245)	—
		1 ms (Interval între 0,001 și 32,767 s)	—	4 (T246–T249)
	FX3U	100 ms (Interval între 0,1 și 3276,7 s)	200 (T0–T199)	6 (T250–T255)
		10 ms (Interval între 0,01 și 327,67 s)	46 (T200–T245)	—
		1 ms (Interval între 0,001 și 32,767 s)	256 (T256–T511)	4 (T246–T249)

* Aceste temporizatoare sunt disponibile doar atunci când este setat releul special M8028. Numărul total al temporizatoarelor pentru 100ms este în acest caz redus la 32 (T0–T31).

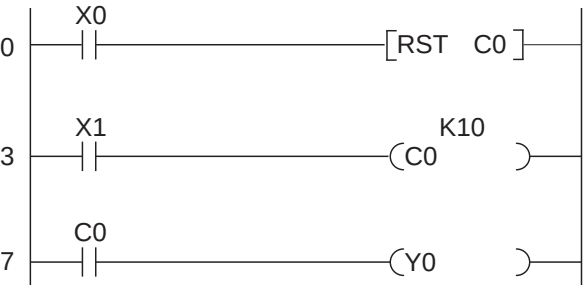
4.4 Contoare

Automatele din seria FX au de asemenea contoare interne pe care le puteți utiliza pentru programarea operațiilor de numărare.

Contoarele numără impulsurile de semnal aplicate de către program intrărilor corespundente. Ieșirea controlată de un contor se închide atunci când valoarea curentă a contorului atinge valoarea de referință definită în program. La fel ca temporizatoarele, ieșirile contoarelor pot de asemenea fi interogate ori de câte ori este necesar pe parcursul programului.

Exemplu de program ce utilizează un contor:

Schemă cu contacte

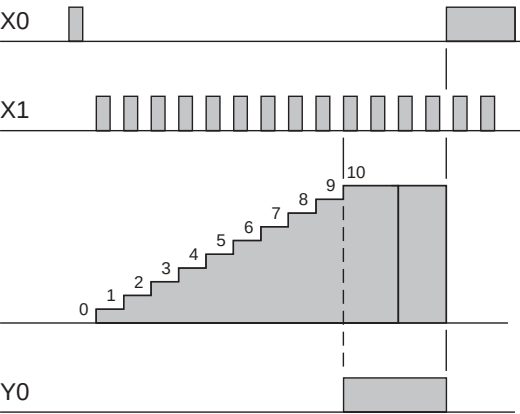


Listă de instrucțiuni

0	LD	X0
1	RST	C0
3	LD	X1
4	OUT	C0
7	LD	C0
8	OUT	Y0

Atunci când intrarea X1 este închisă, valoarea contorului C0 este incrementată cu 1. Ieșirea Y0 este setată după ce X1 a fost închisă și deschisă de 10 ori (valoarea de referință a contorului este K10).

Secvența semnalelor generate de acest program este următoarea:



Mai întâi, contorul este resetat cu intrarea X0 și o instrucțiune RST. Astfel, valoarea contorului este resetată la 0, ceea ce determină deschiderea ieșirii contorului.

După ce valoarea contorului a atins valoarea de referință, nici unul dintre impulsurile suplimentare din intrarea X1 nu va mai avea nici un efect asupra contorului.

Există două tipuri de contoare, pe 16 și pe 32 de biți. După cum le arată și numele, acestea pot stoca valori de până la 16 sau 32 de biți. Următorul tabel arată caracteristicile importante ale acestor contoare.

Caracteristică	Contoare pe 16 biți	Contoare pe 32 de biți
Sensul de numărare	Crescător	Crescător sau descrescător (sensul de numărare este determinat de starea unui releu special)
Intervalul de valori de referință	1 – 32767	-2 147 483 648 la 2 147 483 647
Introducerea valorii de referință	Direct, drept constantă în baza 10 (K), în instrucțiune, sau indirect, într-un registru de date	Direct, drept constantă în baza 10 (K), în instrucțiune, sau direct, printr-o pereche de regiștri
Comportament la depășirea valorii maxime a contorului	Numără până la 32.767, apoi valoarea contorului nu se mai modifică.	Contor de tip inel: după atingerea valorii de 2,147,483,647 următoarea valoare este -2,147,483,647 (în cazul numărării inverse după -2,147,483,647 urmează 2,147,483,647)
Ieșire contor	După ce a fost atinsă valoarea de referință, ieșirea rămâne închisă.	La incrementare ieșirea rămâne activată odată ce valoarea setată a fost atinsă. La decrementare ieșirea este resetată dacă valoarea curentă a contorului scade sub valoarea setată.
Resetare	Se utilizează o instrucțiune RST pentru a șterge valoarea curentă a contorului și a închide ieșirea acestuia.	

În plus față de contoarele normale, controlerele din seria MELSEC FX au de asemenea și contoare de mare viteză. Acestea sunt contoare pe 32 de biți ce pot procesa semnale de mare viteză citite de intrările X0 – X7. În combinație cu unele instrucțiuni speciale, este foarte simplu să utilizați aceste contoare pentru automatizarea sarcinilor de poziționare și a altor funcții.

Contoarele de mare viteză utilizează un principiu de întrerupere: Programul automatului este întrerupt și răspunde imediat la semnalul contorului. Pentru o descriere detaliată a contoarelor de mare viteză, consultați Manualul de programare pentru seria MELSEC FX.

Prezentare generală a contoarelor

Dispozitiv			Tip de contor	
			Contoare normale	Contoare cu memorare
Identificator dispozitiv			C	
Tip dispozitiv (pentru setare și interogare)			Dispozitiv bit	
Valori posibile ale dispozitivului (ieșire contor)			0 sau 1	
Formatul adresei dispozitivului			Zecimal	
Introducerea valorii de referință a contorului			Constantă număr întreg, în baza 10. Valoarea de referință poate fi setată fie direct în instrucțiune, fie indirect într-un registru de date (doi regiștri de date pentru contoarele pe 32 de biți).	
Numărul de dispozitive și de adrese	FX1S	Contor pe 16 biți	16 (C0–C15)	16 (C16–C31)
		Contor pe 32 de biți	—	—
		Contor de mare viteză pe 32 de biți	—	21 (C235–C255)
	FX1N	Contor pe 16 biți	16 (C0–C15)	184 (C16–C199)
		Contor pe 32 de biți	20 (C200–C219)	15 (C220–C234)
		Contor de mare viteză pe 32 de biți	—	21 (C235–C255)
	FX2N FX2NC	Contor pe 16 biți	100 (C0–C99) ^①	100 (C100–C199) ^①
		Contor pe 32 de biți	20 (C200–C219) ^②	15 (C220–C234) ^②
		Contor de mare viteză pe 32 de biți	21 (C235–C255) ^②	
	FX3U	Contor pe 16 biți	100 (C0–C99) ^①	100 (C100–C199) ^①
		Contor pe 32 de biți	20 (C200–C219) ^②	15 (C220–C234) ^②
		Contor de mare viteză pe 32 de biți	21 (C235–C255) ^②	

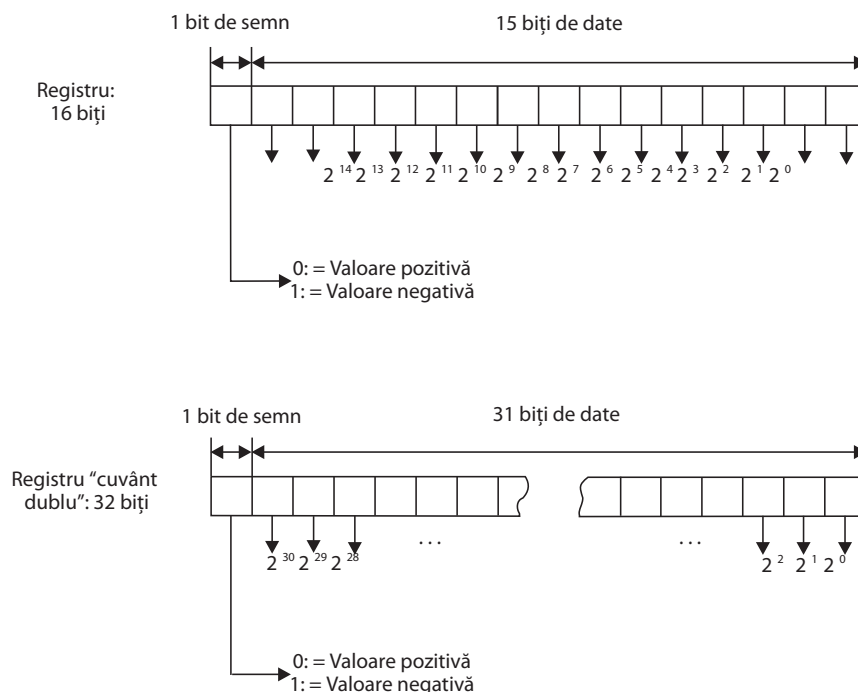
① Valorile curente ale contoarelor cu memorie sunt păstrate la întreruperea alimentării cu energie.

② Puteți seta parametrii automatului programabil astfel încât să configurați păstrarea acestor valori la întreruperea alimentării cu energie sau resetarea lor.

4.5 Regiștri

Releele automatelor programabile sunt utilizate pentru a stoca temporar rezultatele operațiilor. Totuși, releele pot stoca doar valori On/Off sau 1/0, ceea ce înseamnă că nu sunt adecvate pentru stocarea măsurătorilor sau a rezultatelor calculelor. Valori de acest tip pot fi stocate în "regiștrii" controlerelor din seria FX.

Regiștrii au 16 biți (un cuvânt) de informație (a se vedea și secțiunea 3.2). Puteți crea regiștri "cuvânt dublu", ce pot stoca valori de până la 32 de biți, prin combinarea a doi regiștri de date consecutivi.



Un registru obișnuit poate stoca valori între 0000H și FFFFh (-32.768 – 32.767). Regiștrii "cuvânt dublu" pot stoca valori între 00000000H și FFFFFFFFH (-2.147.483.648 – 2.147.483.647).

Controlerelor din seria FX au un număr mare de instrucțiuni pentru utilizarea și manipularea regiștrilor. Puteți scrie și citi valori în și din regiștri, puteți copia conținutul acestora, le puteți compara și puteți executa operații și funcții matematice asupra conținutului acestora (a se vedea capitolul 5).

4.5.1 Regiștri de date

Regiștrii de date pot fi utilizați ca memorie în programele dvs. pentru automate programabile. O valoare pe care un program o scrie într-un registru de date rămâne stocată în acesta până când programul o suprascrie cu o altă valoare.

Atunci când utilizați instrucțiuni pentru manipularea datelor pe 32 de biți, trebuie să specificați doar adresa unui registru pe 16 biți. Partea cea mai semnificativă a datelor pe 32 de biți este scrisă automat în următorul registru. De exemplu, dacă specificați registrul D0 ca destinație pentru stocarea unei valori pe 32 de biți, D0 va conține biții 0 - 15, iar D1 va conține biții 16 - 31.

Ce se întâmplă la deconectarea sau oprirea automatului programabil

În plus față de registrul obișnuit, al căror conținut se pierde atunci când automatul programabil este oprit sau când este întreruptă alimentarea, automatele programabile din seria FX au și registre cu memorare, al căror conținut este păstrat în astfel de situații.

NOTĂ

Dacă este setat releul special M8033, la oprirea automatului programabil nu este șters nici conținutul registrilor de date fără memorare.

Prezentare generală a registrilor de date

Dispozitiv		Tipuri de registre de date	
		Registri obișnuiți	Registri cu memorare
Identificator dispozitiv		D	
Tip dispozitiv (pentru setare și interogare)		Dispozitiv cuvânt (pot fi combinați doi registre pentru a stoca valori în dublu cuvânt)	
Valori posibile dispozitiv		Registri pe 16 biți: 0000h - FFFFh (de la -32768 la 32767)	
		Registri pe 32 de biți: 00000000h - FFFFFFFFh (de la -2 147 483 648 la 2 147 483 647)	
Formatul adresei dispozitivului		Zecimal	
Numărul de dispozitive și adrese	FX1S	128 (D0–D127)	128 (D128–D255)
	FX1N	128 (D0–D127)	7872 (D128–D7999)
	FX2N	200 (D0–D199) ^①	312 (D200–D511) ^②
	FX2NC		7488 (D512–D7999)
	FX3U	200 (D0–D199) ^①	524 (D500–D1023) ^②
			6656 (D1024–D7679)

① Puteți configura acești registre ca registre cu memorare utilizând parametrii automatului programabil.

② Puteți configura acești registre ca registre normali utilizând parametrii automatului programabil.

4.5.2

Registri speciali

La fel ca relele speciale (Capitolul 4.2.1) care pornesc de la adresa M8000, controlerile FX au de asemenea și registre speciali, sau de diagnosticare, ale căror adrese încep de la D8000. Adesea, există și o conexiune directă între relele speciale și registrele speciali. De exemplu, releul special M8005 arată că tensiunea din bateria automatului programabil este prea scăzută, iar valoarea corespunzătoare a tensiunii este stocată în registrul special D8005. Următorul tabel conține câțiva dintre registrele speciali disponibili, ca exemple.

Registru special	Funcție	Opțiuni de procesare din program
D8004	Adresă a releului de eroare (arată care dintre relele de eroare sunt setate)	Citirea conținutului registrului
D8005	Tensiune baterie (de ex. valoarea "36" înseamnă 3.6V)	
D8010	Durata ciclului curent de program	
D8013–D8019	Ora și data ceasului de timp real integrat	Citirea conținutului registrului Modificarea conținutului registrului
D8030	Valoare citită de la potențiometru VR1 (0 - 255)	Citirea conținutului registrului (doar FX1S și FX1N)
D8031	Valoare citită de la potențiometru VR2 (0 - 255)	

Registri cu conținut modificabil din exterior

Controlerile din seriile FX1S și FX1N au două potențiometre integrate, cu care puteți ajusta conținutul registrilor speciali D8030 și D8031 în intervalul 0 – 255 (a se vedea secțiunea 4.6.1). Aceste potențiometre pot fi utilizate pentru diverse sarcini, de exemplu pentru ajustarea valorii de referință a temporizatoarelor și contoarelor fără a mai fi necesară conectarea la controler a unei unități de programare.

4.5.3 Regiștri de fișiere

Conținutul regiștrilor de fișiere este de asemenea memorat la întreruperea alimentării cu energie. Regiștrii de fișiere pot fi astfel utilizați pentru stocarea de valori pe care trebuie să le transferați în regiștrii de date la pornirea automatului programabil, astfel încât să poată fi utilizate de către program pentru calcule, comparații sau ca valori de referință pentru temporizatoare.

Regiștrii de fișiere au aceeași structură ca regiștrii de date. De fapt, aceștia sunt regiștri de date, grupați în blocuri de câte 500 de adrese, fiecare bloc putând fi setat în intervalul de adrese D1000 – D7999.

Dispozitiv		Regiștri de fișiere
Identificator dispozitiv		D
Tip dispozitiv (pentru setare și interogare)		Dispozitiv cuvânt (pot fi combinați doi regiștri pentru a stoca valori în cuvânt dublu)
Valori posibile dispozitiv		Registru pe 16 biți: 0000h - FFFFh (-32768 - 32767) Registru pe 32 biți: 00000000H - FFFFFFFFh (-2 147 483 648 – 2 147 483 647)
Formatul adresei dispozitivului		Zecimal
Numărul de dispozitive și de adrese	FX1S	1500 (D1000–D2499) Se pot defini, prin intermediul parametrilor automatului programabil, maximum 3 blocuri de câte 500 de regiștri de fișiere.
	FX1N	7000 (D1000–D7999) Se pot defini, prin intermediul parametrilor automatului programabil, maximum 14 blocuri de câte 500 de regiștri de fișiere.
	FX2N	
	FX2NC	
	FX3U	

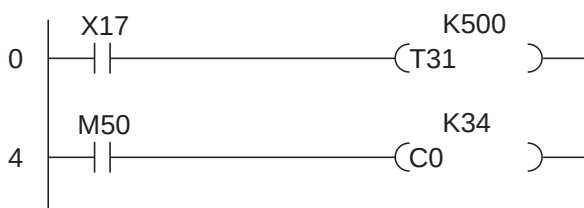
Pentru o descriere detaliată a regiștrilor de fișiere, consultați Manualul de programare pentru seria MELSEC FX.

4.6 Tehnici de programare pentru temporizatoare și contoare

4.6.1 Specificarea indirectă a valorilor de referință pentru temporizatoare și contoare

Cel mai obișnuit mod de a specifica valori de referință pentru temporizatoare și contoare este direct, în instrucțiunea de ieșire:

Schemă cu contacte



Listă de instrucțiuni

0	LD	X17	
1	OUT	T31	K500
4	LD	M50	
5	OUT	C0	K34

În exemplul de mai sus, T31 este un temporizator cu baza de timp de 100ms. Constanta K500 setează întârzierea la $500 \times 0,1 \text{ sec} = 50 \text{ sec}$. Valoarea de referință pentru contorul C0 este de asemenea setată direct, la valoarea 34, cu ajutorul constantei K34.

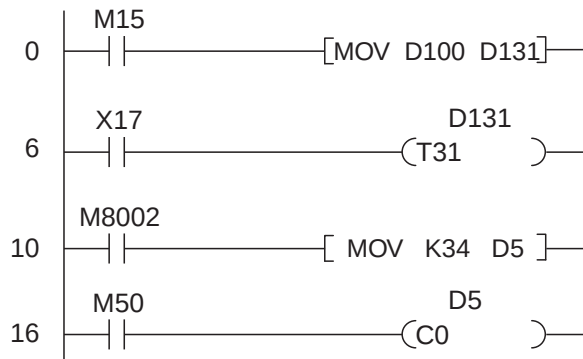
Avantajul specificării de valori de referință în acest mod este acela că nu trebuie să vă faceți griji privind valoarea de referință, după ce ați setat-o. Valorile pe care le utilizați în program sunt întotdeauna valide, chiar și după căderi de tensiune și pornirea directă a controlerului. Există însă și un dezavantaj: Dacă doriți să modificați valoarea de referință, trebuie să editați programul. Acest lucru este valabil mai ales pentru valorile de referință ale temporizatorilor, care sunt adesea ajustate la configurarea controlerelor și în timpul testelor programului.

Puteți de asemenea stoca valori de referință pentru temporizatoare și contoare în regiștrii de date și puteți instrui programul să le citească din regiștri. Ulterior, puteți modifica valorile rapid cu o unitate

de programare, dacă este necesar, sau puteți specifica valori de referință cu ajutorul comutatorilor de pe o consolă de control sau prin intermediul unui panou de control HMI.

Următoarea listă arată un exemplu al modului în care puteți specifica valori de referință indirect:

Schemă cu contacte



Listă de instrucțiuni

```

0  LD      M15
1  MOV     D100  D131
6  LD      X17
7  OUT     T31   D131
10 LD      M8002
11 MOV     K34   D5
16 LD      M50
17 OUT     C0    D5

```

- Atunci când releul M15 este setat, conținutul registrului D100 este copiat în D131. Acest registru conține valoarea de referință pentru T131. Puteți utiliza o unitate de programare sau de control pentru a ajusta conținutul registrului D100.
- Releul special M8002 este setat doar pe durata primului ciclu de program. Acesta este utilizat pentru a copia constanta 34 în registrul de date D5, care este apoi utilizat ca referință pentru contorul C0.

Nu trebuie să scrieți instrucțiuni de program pentru a copia valorile punctelor de referință în regiștri de date. Puteți de asemenea să utilizați o unitate de programare pentru a seta aceste valori, de exemplu după pornirea programului.



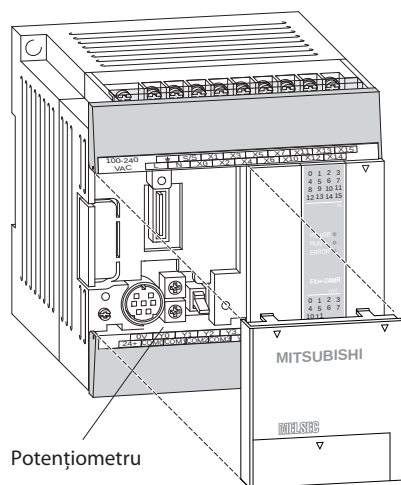
AVERTISMENT:

Dacă utilizați regiștri normali, valorile de referință vor fi pierdute atunci când se întrerupe alimentarea cu energie electrică sau când comutatorul RUN/STOP este comutat în poziția STOP. Dacă se întâmplă acest lucru, data viitoare când se conectează automatul la sursa de energie electrică sau când acesta este pornit din nou, ar putea fi create condiții de risc, deoarece toate valorile de referință vor avea valoarea "0".

Dacă nu configurați programul astfel încât să copieze automat valorile, trebuie să utilizați întotdeauna regiștri de date cu memorare pentru stocarea valorilor de referință ale temporizatoarelor și contoarelor. De asemenea, nu uitați: chiar și conținutul acestor regiștri se va pierde atunci când automatul programabil este oprit și valoarea tensiunii bateriei de rezervă a scăzut sub limita minimă.

Setarea valorilor de referință cu ajutorul potențioametrelor integrate

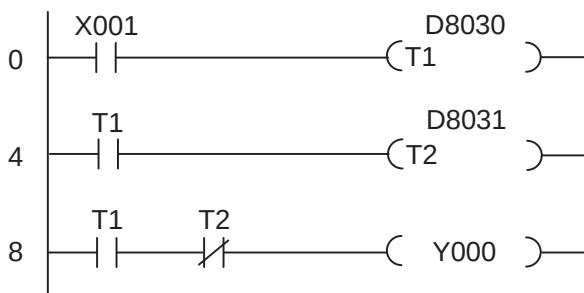
Controlerele din seriile FX1S și FX1N au două potențioametre analogice integrate, cu care puteți ajusta valorile de referință ale contoarelor și ale altor funcții, ușor și rapid.



Valoarea potențiometrului din partea de sus (VR1) poate fi citită din registrul de date special D8030, valoarea potențiometrului din partea de jos (VR2) poate fi citită din registrul de date special D8031. Pentru a utiliza unul dintre potențioametre ca sursă pentru valoarea de referință a unui temporizator, trebuie doar să specificați registrul corespundent în programul dvs., în loc de a specifica o constantă.

Valoarea din registru poate fi ajustată între 0 și 255 prin rotirea potențiometrului.

Schemă cu contacte

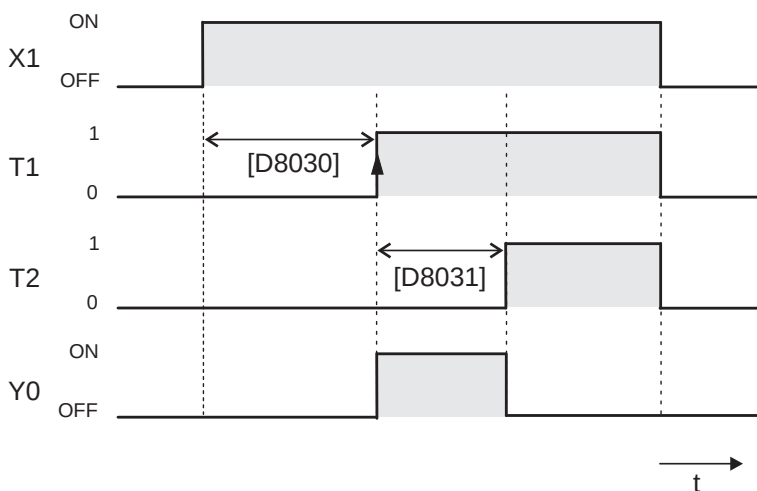


Listă de instrucțiuni

0	LD	X001	
1	OUT	T1	D8030
4	LD	T1	
5	OUT	T2	D8031
8	LD	T1	
8	ANI	T2	
10	OUT	Y000	

În exemplul de programare de mai sus, Y0 este comutat pe închis după întârzierea specificată pentru temporizatorul T1, pentru durata specificată de temporizatorul T2 (generare întârziată a impulsului).

Secvența de semnale

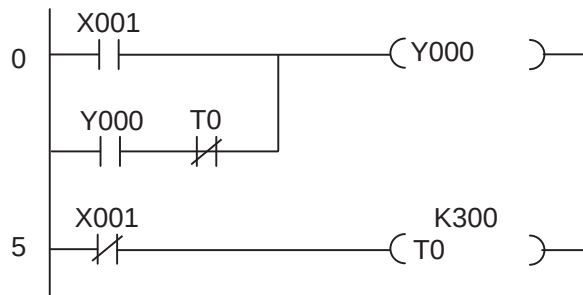


4.6.2 Întârziere la deconectare

În mod predefinit, toate temporizatoarele din automatele programabile MELSEC sunt temporizatoare cu întârziere la conectare, cu alte cuvinte ieșirea este închisă după perioada de întârziere definită. Va apărea însă adesea necesitatea de a programa o operație de întrerupere a unui semnal cu o anumită întârziere față de momentul deconectării (comutarea la OFF după o întârziere). Un exemplu tipic ar fi un ventilator dintr-o baie, care trebuie să continue să meargă câteva minute după stingerea luminilor.

Versiune 1 programare (cu automenținere)

Schemă cu contacte

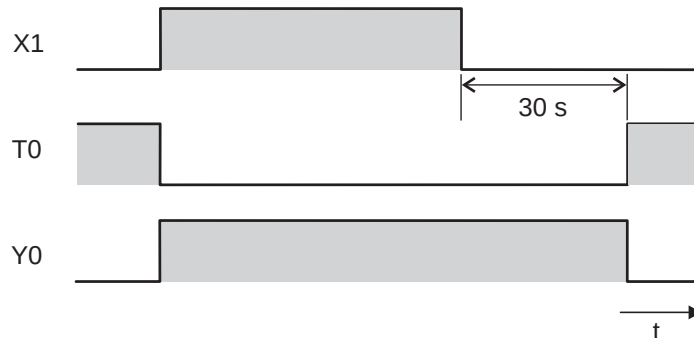


Listă de instrucțiuni

0	LD	X001
1	LD	Y000
2	ANI	T0
3	ORB	
4	OUT	Y000
5	LDI	X001
6	OUT	T0 K300

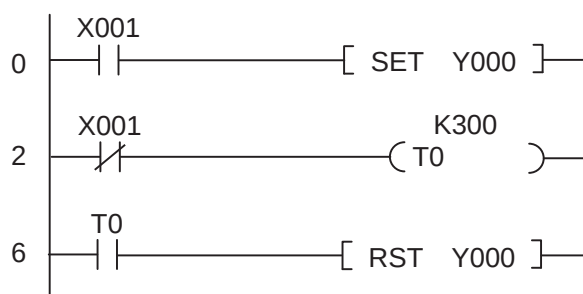
Atâta vreme cât intrarea X1 (de exemplu un comutator de iluminat) este închisă, ieșirea Y0 (ventilatorul) este de asemenea închisă. Funcția de automenținere asigură însă faptul că Y0 va rămâne închisă după ce X1 a fost deschisă, deoarece temporizatorul T0 încă mai cronometrează. T0 este pornit la deschiderea intrării X1. La finalul perioadei de întârziere (300x0,1sec = 30sec în acest exemplu), T0 întrerupe automenținerea și deschide ieșirea.

Secvența de semnale



Versiunea 2 de programare (setare/resetare)

Schemă cu contacte



Listă de instrucțiuni

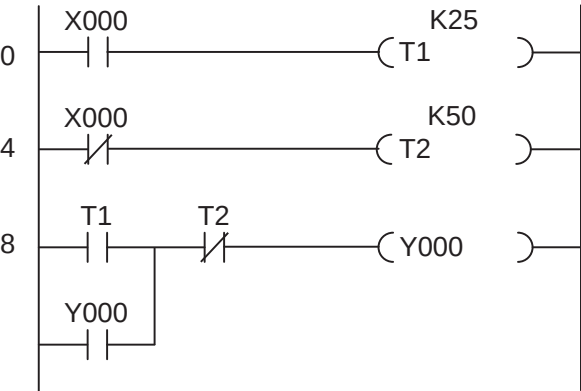
0	LD	X001
1	SET	Y000
2	LDI	X001
3	OUT	T0 K300
6	LD	T0
7	RST	Y000

Atunci când X1 este închisă, este setată ieșirea Y0 (pornită). Atunci când X1 este deschisă, este pornit temporizatorul T0. După perioada de întârziere, T0 va reseta ieșirea Y0. Secvența de semnale rezultată este identică celei generate de versiunea 1 de programare.

4.6.3 **Întârziere la activare și la dezactivare**

Uneori, veți dori să deschideți o ieșire cu o anumită întârziere, iar apoi, după o altă întârziere, să o închideți la loc. Acest lucru este foarte simplu de implementat cu ajutorul instrucțiunilor logice elementare ale controlerului.

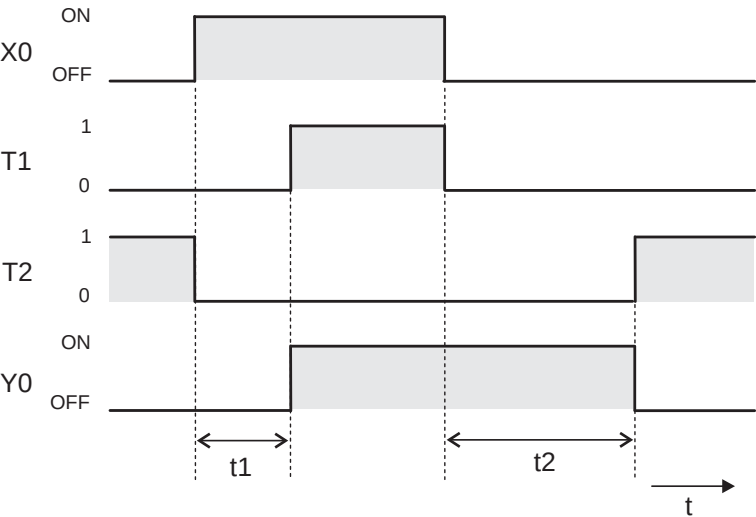
Schemă cu contacte



Listă de instrucțiuni

0	LD	X000	
1	OUT	T1	K25
4	LDI	X000	
5	OUT	T2	K50
8	LD	T1	
9	OR	Y000	
10	ANI	T2	
11	OUT	Y000	

Secvența de semnale



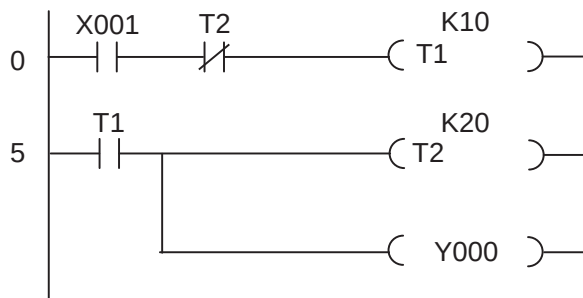
După activarea cu ajutorul temporizatorului T1, ieșirea Y000 se automenține până la expirarea temporizării la deconectare T2.

4.6.4 Generatoare de semnal de ceas

Controlerele au relee speciale care simplifică foarte mult sarcinile ce necesită un semnal de ceas regulat (de exemplu pentru a controla un semnal luminos intermitent ce semnalizează o eroare). De exemplu, releul M8013 este comutat pe pornit și pe oprit, alternativ, la intervale de 1 secundă. Pentru detalii complete privind toate relele speciale, consultați Manualul de programare pentru seria FX.

Dacă aveți nevoie de o altă frecvență de ceas sau dacă este necesar ca durata ON și durata OFF să fie diferite, puteți programa propriul dvs. generator de semnal de ceas, cu două temporizatoare, astfel:

Schemă cu contacte



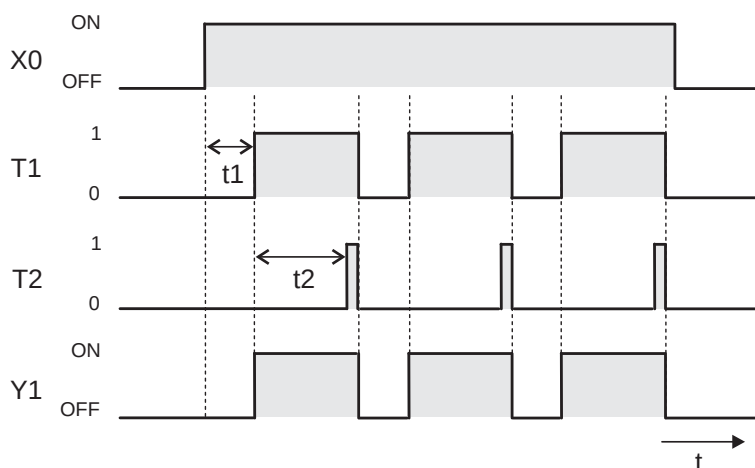
Listă de instrucțiuni

0	LD	X001	
1	ANI	T2	
2	OUT	T1	K10
5	LD	T1	
6	OUT	T2	K20
9	OUT	Y000	

Intrarea X1 pornește generatorul de ceas. Dacă doriți, puteți omite această intrare, pentru ca generatorul de ceas să fie mereu pornit. În program, puteți utiliza ieșirea temporizatorului T1 pentru a controla o lumină de avertizare intermitentă. Perioada ON este determinată de T2, perioada OFF de către T1.

Ieșirea temporizatorului T2 este comutată pe ON doar pentru un singur ciclu de program. Această durată este arătată ca fiind mult mai lungă decât este în realitate în ilustrația de secvență de semnale de mai jos. T2 determină comutarea pe OFF a temporizatorului T1, iar imediat după aceea T2 însuși este de asemenea comutat pe OFF. Acest lucru înseamnă de fapt că durata perioadei ON este prelungită cu durata de execuție a unui ciclu de program. Deoarece ciclul de program durează însă doar câteva milisecunde, acesta poate fi de obicei ignorat.

Secvența de semnale



5 Tehnici avansate de programare

Instrucțiunile logice elementare listate în Capitolul 3 pot fi utilizate pentru ca automatul programabil să poată îndeplini funcțiile unui controler cablat. Acest lucru însă este doar suprafața în ceea ce privește capacitățile automatelor programabile moderne. Deoarece orice automat programabil este construit pornind de la un microprocesor, acestea pot de asemenea efectua cu ușurință operații precum calcule matematice, compararea de numere, conversia dintr-o bază numerică în alta sau procesarea de valori analogice.

Funcții ca acestea, ce merg dincolo de capacitățile operatorilor logici, sunt executate cu ajutorul instrucțiunilor speciale, ce sunt denumite instrucțiuni aplicate sau de aplicație.

5.1 Listă instrucțiuni aplicate

Instrucțiunile aplicate au denumiri scurte bazate pe denumirile în engleză ale funcțiilor lor. De exemplu, instrucțiunea pentru compararea a două numere pe 16 biți și pe 32 de biți este numită CMP, ceea ce este prescurtarea de la "compare" (a compara).

Atunci când programați o instrucțiune aplicată, introduceți numele instrucțiunii urmat de numele dispozitivului. Următorul tabel arată toate instrucțiunile aplicate suportate în prezent de seria de controlere MELSEC FX. Această listă poate părea prea lungă la început, dar nu trebuie să memorați toate instrucțiunile. Atunci când programați, puteți utiliza funcțiile performante de Help ale GX Developer și GX IEC Developer pentru a găsi instrucțiunile de care aveți nevoie.

În acest capitol, vom acoperi doar instrucțiunile cel mai des utilizate, care sunt afișate cu un fundal gri în tabel. Pentru documentația completă privind toate instrucțiunile, cu exemple, vă rugăm consultați Manualul de programare pentru seria FX.

Categorie	Instrucțiune	Funcție	Controler				
			FX1S	FX1N	FX2N	FX2NC	FX3U
Instrucțiuni de control al programului	CJ	Salt condițional la o poziție din program					
	CALL	Apelează (execută) o subrutină					
	SRET	Retur subrutină, marchează sfârșitul unei subrutine					
	IRET	Retur întrerupere, marchează sfârșitul unei rutine de întrerupere					
	EI	Permitere întrerupere, permite procesarea rutinelor de întrerupere					
	DI	Interzicere întrerupere, interzice procesarea rutinelor de întrerupere	●	●	●	●	●
	FEND	Final program, marchează finalul programului principal					
	WDT	Resetarea temporizatorului "watch dog"					
	FOR	Marchează începutul unei bucle de program					
	NEXT	Marchează sfârșitul unei bucle de program					
Funcții de mutare și comparare	CMP	Compară valori numerice	●	●	●	●	●
	ZCP	Comparare domeniu, compară intervale numerice	●	●	●	●	●
	MOV	Mută datele dintr-o zonă de stocare în alta	●	●	●	●	●
	SMOV	Mutare prin deplasare			●	●	●
	CML	Negație (complementar), copiază și inversează			●	●	●
	BMOV	Mutare bloc	●	●	●	●	●
	FMOV	Mutare cu umplere, scriere valoare într-un bloc de dispozitive			●	●	●
	XCH	Schimb de date în dispozitivele specificate			●	●	●
	BCD	Conversie BCD	●	●	●	●	●
	BIN	Conversie binară	●	●	●	●	●

Categorie	Instrucțiune	Funcție	Controler				
			FX1S	FX1N	FX2N	FX2NC	FX3U
Funcții matematice și logice	ADD	Adună valori numerice	●	●	●	●	●
	SUB	Scade valori numerice	●	●	●	●	●
	MUL	Înmulțește valori numerice	●	●	●	●	●
	DIV	Împarte valori numerice	●	●	●	●	●
	INC	Incrementare	●	●	●	●	●
	DEC	Decrementare	●	●	●	●	●
	AND	AND (ȘI) logic	●	●	●	●	●
	OR	OR (SAU) logic	●	●	●	●	●
	XOR	OR (SAU) logic exclusiv	●	●	●	●	●
	NEG	Negație, inversarea logică a conținutului dispozitivului			●	●	●
Funcții de rotire și deplasare	ROR	Rotire la dreapta			●	●	●
	ROL	Rotire la stânga			●	●	●
	RCR	Rotire bit de transport la dreapta, rotire la dreapta cu bit de transport			●	●	●
	RCL	Rotire bit de transport la stânga, rotire la stânga cu bit de transport			●	●	●
	SFTR	Deplasare spre dreapta, deplasare bit spre dreapta	●	●	●	●	●
	SFTL	Deplasare spre stânga, deplasare bit spre stânga	●	●	●	●	●
	WSFR	Deplasare cuvânt spre dreapta			●	●	●
	WSFL	Deplasare cuvânt spre stânga			●	●	●
	SFWR	Scrisoare cu deplasare în registru, scrie într-o stivă FIFO	●	●	●	●	●
Funcții operare date	SFRD	Citire cu deplasare în registru, citește dintr-o stivă FIFO	●	●	●	●	●
	ZRST	Resetare domeniu, resetează domenii de dispozitive similare	●	●	●	●	●
	DECO	Decodificare date	●	●	●	●	●
	ENCO	Codificare date	●	●	●	●	●
	SUM	Suma (numărul) biților activi			●	●	●
	BON	Bit ON, verifică starea unui bit			●	●	●
	MEAN	Calculează valorile medii			●	●	●
	ANS	Setare indicator de temporizare, pornește un interval al unui temporizator			●	●	●
	ANR	Resetare indicator de temporizare			●	●	●
	SQR	Radical			●	●	●
Instrucțiuni de mare viteză	FLT	Virgulă mobilă, convertește date			●	●	●
	REF	Reactualizarea intrărilor și ieșirilor	●	●	●	●	●
	REFF	Reactualizarea intrări și ajustare filtru intrări			●	●	●
	MTR	Matrice de intrări, citește o matrice (MTR)			●	●	●
	DHSCS	Setare contor de mare viteză	●	●	●	●	●
	DHSCR	Resetare contor de mare viteză	●	●	●	●	●
	DHSZ	Comparare domenii de mare viteză			●	●	●
	SPD	Detectare viteză	●	●	●	●	●
	PLSY	Ieșire în impulsuri (frecvență)	●	●	●	●	●
	PWM	Ieșire în impulsuri cu modulație PWM	●	●	●	●	●
Instrucțiuni aplicate	PLSR	Ieșire în impulsuri de tip rampă (setare accelerație/decelerație)	●	●	●	●	●
	IST	Stare inițială, configurare sistem STL multi-mod	●	●	●	●	●
	SER	Căutare în stivă de date			●	●	●
	ABSD	Comparare contor absolut	●	●	●	●	●
	INCD	Comparare contor incremental	●	●	●	●	●
	TTMR	Temporizator cu învățare			●	●	●
	STMR	Temporizator special			●	●	●
	ALT	Stare alternativă, funcție bistabil	●	●	●	●	●
	RAMP	Funcție de timp rampă	●	●	●	●	●
	ROTC	Control masă rotativă			●	●	●
Instrucțiuni aplicate	SORT	Sortare date din tabel după câmpurile selectate			●	●	●

Categorie	Instrucțiune	Funcție	Controler				
			FX1S	FX1N	FX2N	FX2NC	FX3U
Instrucțiuni pentru dispozitive externe intrare/ieșire	TKY	Intrări în baza 10			●	●	●
	HKY	Intrări în baza 16			●	●	●
	DSW	Comutator digital	●	●	●	●	●
	SEGD	Decoder cu afișare 7 segmente			●	●	●
	SEGL	Afișare 7 segmente, cu memorare	●	●	●	●	●
	ARWS	Comutator săgeată			●	●	●
	ASC	Conversie ASCII			●	●	●
	PR	Tipărire, furnizarea datelor prin ieșiri			●	●	●
	FROM	Citirea datelor dintr-un modul de funcție specială		●	●	●	●
	TO	Scrierea datelor într-un modul de funcție specială		●	●	●	●
Instrucțiuni pentru dispozitive seriale externe	RS	Comunicații seriale RS	●	●	●	●	●
	PRUN	Rulare în paralel (mod octal)	●	●	●	●	●
	ASCI	Conversie la caracter ASCII	●	●	●	●	●
	HEX	Conversie la caracter hexazecimal	●	●	●	●	●
	CCD	Verificare cod, verificare sumă și paritate	●	●	●	●	●
	VRRD	Citire valori de referință de la FX1N-8AV-BD și FX2N-8AV-BD	●	●	●	●	●
	VRSC	Citire setări comutator de la FX1N-8AV-BD și FX2N-8AV-BD	●	●	●	●	●
	RS2	Comunicații seriale RS (2)					●
	PID	Programarea unei bucle de control PID	●	●	●	●	●
Stocare/Recuperare regiștri indecși	ZPUSH	Memorare în stivă, stocare valori regiștri index					●
	ZPOP	Recuperare din stivă, citește valoarea regiștrilor index					●
Operații în virgulă mobilă	DECOMP	Comparare valori în virgulă mobilă			●	●	●
	DEZCP	Comparare valori în virgulă mobilă (domeniu)			●	●	●
	DEMOV	Mutare valori în virgulă mobilă					●
	DESTR	Conversie valoare în virgulă mobilă într-un șir					●
	DEVAL	Conversie șir la valoare în virgulă mobilă					●
	DEBCD	Conversie valoare în virgulă mobilă la notație științifică			●	●	●
	DEBIN	Conversie notație științifică la valoare în virgulă mobilă			●	●	●
	DEADD	Adunare numere în virgulă mobilă			●	●	●
	DESUB	Scădere numere în virgulă mobilă			●	●	●
	DEMUL	Înmulțire numere în virgulă mobilă			●	●	●
	DEDIV	Împărțire numere în virgulă mobilă			●	●	●
	DEXP	Exponent virgulă mobilă					●
	DLOGE	Calculare logaritm natural					●
	DLOG10	Calculare logaritm în baza 10					●
	DESQR	Rădăcină pătrată din numere în virgulă mobilă			●	●	●
	DENEG	Inversarea semnului numerelor în virgulă mobilă					●
	INT	Conversie numere în virgulă mobilă la întregi			●	●	●
Instrucțiuni trigonometrice pentru numerele în virgulă mobilă	SIN	Calculare sinus			●	●	●
	COS	Calculare cosinus			●	●	●
	TAN	Calculare tangentă			●	●	●
	ASIN	Calculare arcsinus					●
	ACOS	Calculare arccosinus					●
	ATAN	Calculare arctangentă					●
	RAD	Conversie grade în radiani					●
	DEG	Conversie radiani în grade					●
Operații cu date	WSUM	Suma conținutului dispozitivelor cuvânt					●
	WTOB	Cuvânt la octet, divizare cuvinte în octeți					●
	BTOW	Octet la cuvânt, formare cuvinte din octeți individuali					●
	UNI	Combinare grupuri de 4 biți pentru a forma cuvinte					●
	DIS	Divizare cuvinte în grupuri de 4 biți					●
	SWAP	Inversarea bitului cel mai semnificativ cu cel mai puțin semnificativ			●	●	●
	SORT	Sortarea datelor dintr-un tabel					●

Categorie	Instrucțiune	Funcție	Controler				
			FX1S	FX1N	FX2N	FX2NC	FX3U
Instrucțiuni de poziționare	DSZR	Revenire în origine (cu comutator de proximitate)					●
	DVIT	Poziționare cu întrerupere					●
	TBL	Poziționare cu tabel de date					●
	DABS	Citirea poziției curente absolute	●	●	●	●	●
	ZRN	Revenire în origine	●	●			●
	PLSV	Impulsuri de ieșire cu frecvență variabilă	●	●			●
	DRVI	Poziționare la o valoare incrementală	●	●			●
	DRVA	Poziționare la o valoare absolută	●	●			●
Operații cu ceasul intern al automatului programabil	TCMP	Comparare date ceas	●	●	●	●	●
	TZCP	Comparare date ceas cu un domeniu (interval)	●	●	●	●	●
	TADD	Adunare date ceas	●	●	●	●	●
	TSUB	Scădere date ceas	●	●	●	●	●
	HTOS	Conversia valorii temporale din ore/minute/secunde în secunde					●
	STOH	Conversia valorii temporale din secunde în ore/minute/secunde					●
	TRD	Citirea datei și orei ceasului	●	●	●	●	●
	TWR	Scrierea datei și orei în ceasul automatului programabil	●	●	●	●	●
	HOUR	Operarea contorului pentru ore	●	●	●	●	●
Conversie din și în cod Gray	GRY	Conversie cod Gray în zecimal			●	●	●
	GBIN	Conversie număr zecimal în cod Gray					
Schimb de date cu modulele analogice	RD3A	Citire valori de intrare analogice		●	●	●	●
	WR3A	Scriere valori de intrare analogice					
Instrucțiuni în memoria externă	EXTR	Execuție comandă stocată în memorie ROM externă			●	●	
Instrucțiuni diverse	COMRD	Citire comentariu dispozitiv					
	RND	Generare număr aleator					
	DUTY	Generare impuls cu o anumită durată					●
	CRC	Verificare date (verificarea CRC)					
	HCMOV	Mutarea valorii curente a unui contor de mare viteză					
Instrucțiuni pentru datele stocate în dispozitive consecutive (blocuri de date)	BK+	Adunarea datelor dintr-un bloc de date					
	BK-	Scăderea datelor dintr-un bloc de date					
	BKCMPE	Compararea datelor din blocuri de date					●
	BKCMPE>						
	BKCMPE<						
	BKCMPE<>						
	BKCMPE<=						
	BKCMPE>=						
Operații cu șiruri	STR	Conversie date binare în șir					
	VAL	Conversie șir în date binare					
	\$+	Concatenare șiruri					
	LEN	Returnează lungimea unui șir					
	RIGHT	Extragere subșir din dreapta					●
	LEFT	Extragere subșir din stânga					
	MIDR	Selectarea unui șir de caractere					
	MIDW	Înlocuirea șirurilor de caractere					
	INSTR	Căutarea unui șir de caractere					
	\$MOV	Mutarea unui șir de caractere					
Operații cu tabele de date	FDEL	Ștergerea datelor dintr-un tabel					
	FINS	Inserarea datelor într-un tabel					
	POP	Citirea ultimelor date introduse într-un tabel					●
	SFR	Deplasare la dreapta a unui cuvânt de date pe 16 biți					
	SFL	Deplasare la stânga a unui cuvânt de date pe 16 biți					

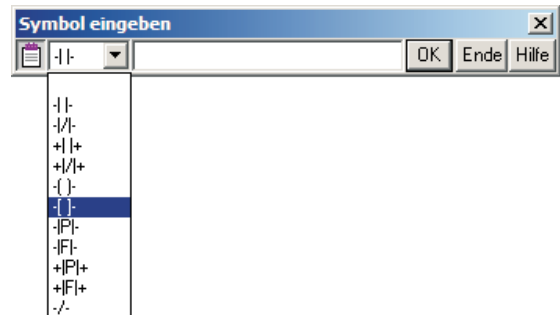
Categorie	Instrucțiune	Funcție	Controler				
			FX1S	FX1N	FX2N	FX2NC	FX3U
Operații de comparare	LD=	Comparare date	●	●	●	●	●
	LD>						
	LD<						
	LD<>						
	LD<=						
	LD>=						
	AND=						
	AND>						
	AND<						
	AND>=						
	OR=						
	OR>						
	OR<						
	OR<>						
	OR<=						
	OR>=						
Instrucțiuni pentru controlul datelor	LIMIT	Limitarea domeniului de ieșire al valorilor					●
	BAND	Definire decalare intrare					
	ZONE	Definire decalare ieșire					
	SCL	Scalare valori					
	DABIN	Conversie număr ASCII în valoare binară					
	BINDA	Conversie valoare binară în cod ASCII					
	SCL2	Scalare valori (structură diferită a tabelului de valori de scalat)					
Instrucțiuni pentru comunicarea cu convertizoarele de frecvență	IVCK	Verificare stare convertizor de frecvență					●
	IVDR	Control convertizor de frecvență					
	IVRD	Citire parametru convertizor de frecvență					
	IVWR	Scriere parametru în convertizor de frecvență					
	IVBWR	Scriere parametri în convertizor de frecvență, în blocuri					
Schimb de date cu modulele cu funcții speciale	RBFM	Citire din memoria-tampon a modului					●
	WBFM	Scriere în memoria-tampon a modului					
Instrucțiuni pentru contoarele de mare viteză	HSCT	Comparare valoare curentă a unui contor de mare viteză cu datele din tabelele de date					●
Instrucțiuni pentru regiștrii de fișiere de extensie	LOADR	Citire date din regiștrii de fișiere de extensie					●
	SAVER	Scriere date în regiștrii de fișiere de extensie					
	INITR	Inițializare regiștri de extensie și regiștri de fișiere de extensie					
	LOGR	Citire valori din dispozitive în regiștrii de extensie și în regiștrii de fișiere de extensie					
	RWER	Scriere date din regiștrii de extensie în regiștrii de fișiere de extensie					
	INITER	Inițializare regiștri de fișiere de extensie					

5.1.1 Introducerea instrucțiunilor aplicate

Programarea instrucțiunilor aplicate în GX Developer FX este foarte simplă. Trebuie doar să poziționați cursorul în locul din linia de program unde doriți să inserați instrucțiunea și să introduceți abrevierile pentru instrucțiune și operandul sau operanzii acesteia. GX Developer va înregistra automat faptul că introduceți o instrucțiune și va deschide dialogul de introducere (ca în imaginea de mai jos). Ca metodă alternativă, puteți de asemenea să poziționați cursorul și apoi să faceți clic pe caseta de inserare a instrucțiunilor din bara de instrumente .



Puteți de asemenea să selectați instrucțiunea din lista derulantă, pe care o puteți afișa printr-un clic pe semnul "▼".



Apoi, introduceți abrevierea instrucțiunii dorite și operandul sau operanzii în câmpul de intrare, separându-i prin spații.

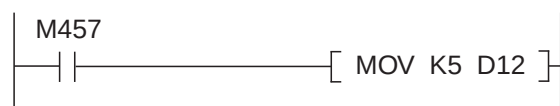
Toate numerele trebuie să fie precedate de o literă, care fie identifică tipul dispozitivului fie, în cazul constantelor, specifică formatul numărului. Litera "K" identifică acele constante ce sunt în sistem zecimal, iar litera "H" identifică acele constante ce sunt în hexazecimal.



În exemplul din stânga, este utilizată o instrucțiune MOV pentru a scrie valoarea 5 în registrul de date D12.

Butonul **Help** va deschide un dialog în care puteți căuta o instrucțiune adecvată pentru funcția pe care o doriți. Helpul conține de asemenea informații despre modul în care operează funcțiile și despre tipul și numărul de dispozitive pe care le pot accepta ca operanzi.

Apoi, faceți doar clic pe OK pentru a insera instrucțiunea aplicată în program.



Da a programa i  n formatul List  cu instruc iuni, introduce i instruc iunea  i operandul sau operanzii acesteia pe o singur  linie, separa i prin spa ii.

5.2 Instrucțiuni pentru mutarea datelor

Automatul programabil utilizează regiștri de date pentru a stoca măsurătorile, valorile, rezultatele intermediare ale operațiilor și valorile din tabele. Instrucțiunile pentru operații matematice ale controlerului pot citi operanzii de care au nevoie direct din regiștrii de date și pot de asemenea să scrie rezultatele în acești regiștri. Aceste instrucțiuni sunt însă suportate și de către instrucțiuni suplimentare de "mutare", cu care puteți copia datele dintr-un registru în altul sau puteți scrie constante în regiștrii de date.

5.2.1 Mutarea valorilor individuale cu instrucțiunea MOV

Instrucțiunea MOV "mută" datele din sursa specificată în destinația specificată.

NOTĂ

Rețineți că în ciuda numelui său, acesta este de fapt un proces de copiere: nu șterge datele din locația sursă.

Schemă cu contacte



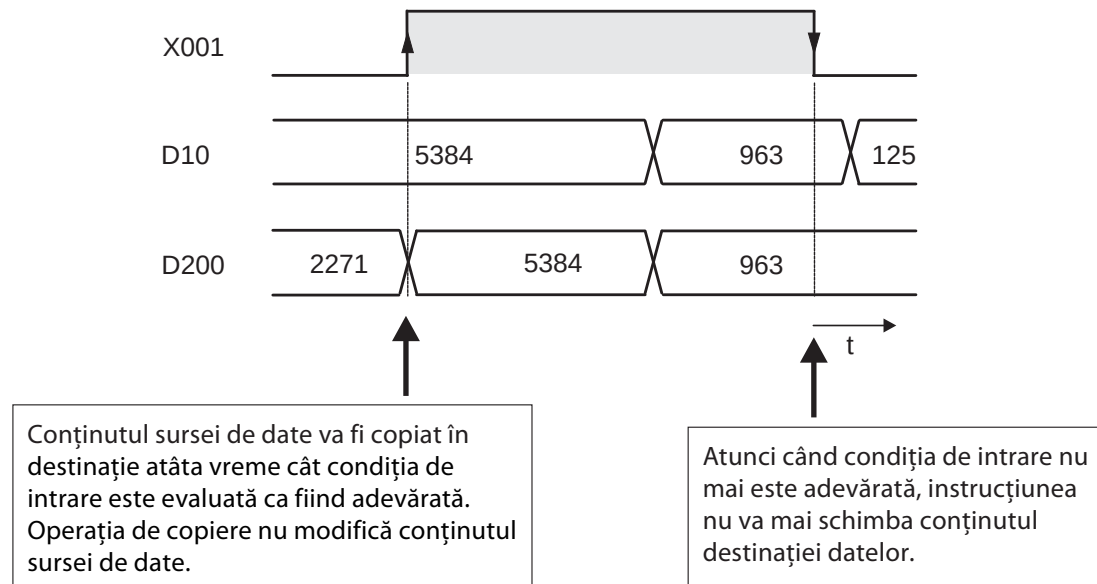
Listă de instrucțiuni

0 MOV D10 D200
① ②

① Sursa de date (poate fi și o constantă)

② destinația datelor

În exemplul de mai sus, valoarea din registrul de date D10 va fi copiată în registrul D200 atunci când intrarea X1 este închisă. Acest lucru va rezulta în următoarea secvență de semnale:



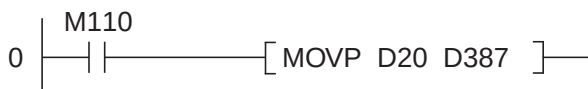
Execuția pe front a instrucțiunii MOV

În unele aplicații, este mai bine dacă valoarea este scrisă în destinație doar într-un singur ciclu de program. De exemplu, ați putea dori să faceți asta dacă alte instrucțiuni din program scriu în aceeași destinație sau dacă operația de mutare trebuie executată la un anumit moment.

Dacă adăugați un "P" instrucțiunii MOV (aceasta devenind astfel MOVP), ea va fi executată o singură dată pe frontul crescător al semnalului generat de condiția de intrare.

În exemplul de mai jos, conținutul din D20 este scris în registrul de date D387 atunci când starea lui M110 se modifică de la "0" la "1".

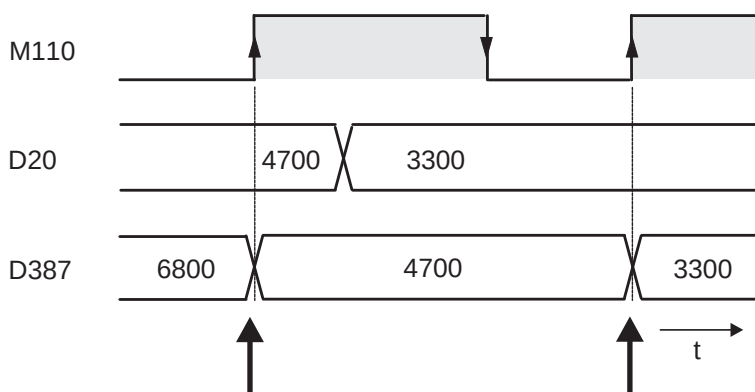
Schemă cu contacte



Listă de instrucțiuni

0	LD	M110
1	MOVP	D20 D387

După ce a fost executată această operație o singură dată, chiar dacă releul M110 rămâne setat, copierea în registrul D387 se oprește. Secvența semnalelor ilustrează acest lucru:

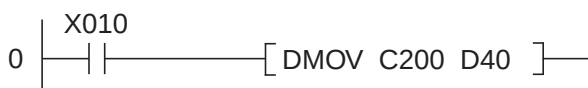


Conținutul sursei de date este copiat în destinație doar pe frontul crescător al semnalului condiției de intrare.

Mutarea datelor pe 32 de biți

Pentru a muta date pe 32 de biți, trebuie doar să adăugați prefixul D la instrucțiunea MOV (DMOV):

Schemă cu contacte



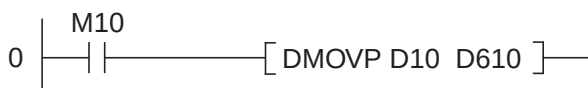
Listă de instrucțiuni

0	LD	X010
1	DMOV	C200 D40

Atunci când intrarea X010 este închisă, valoarea curentă a contorului pe 32 de biți, C200, este scrisă în regiștrii de date D40 și D41. D40 conține biții cei mai puțin semnificativi.

Așa cum vă puteți aștepta, există și o versiune declanșată de impuls a instrucțiunii DMOV pe 32 de biți:

Schemă cu contacte



Listă de instrucțiuni

0	LD	M10
1	DMOVP	D10 D610

Atunci când releul M10 este setat, conținutul regiștrilor D10 și D11 este scris în regiștrii D610 și D611.

5.2.2 Mutarea grupurilor de dispozitive bit

Secțiunea precedentă a arătat cum puteți utiliza instrucțiunea MOV pentru a scrie constante sau conținutul regiștrilor de date în alți regiștri de date. Secvențele consecutive de relee și alte dispozitive bit pot fi de asemenea utilizate atunci când se dorește stocarea valorilor numerice, putând fi copiate ca grupuri cu instrucțiuni aplicate. Pentru aceasta, trebuie să prefixați factorul "K" la adresa primului dispozitiv bit, specificând numărul de dispozitive pe care doriți să le copiați cu operația respectivă.

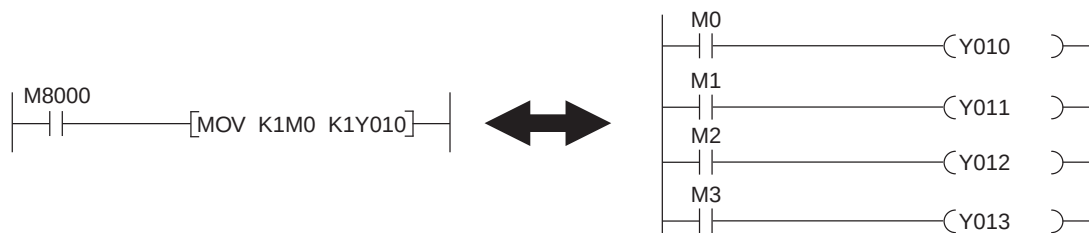
Dispozitivele bit sunt numărate în grupuri de câte 4, astfel încât factorul K specifică numărul de grupuri de 4. K1 = 4 dispozitive, K2 = 8 dispozitive, K3 = 12 dispozitive și așa mai departe.

De exemplu, K2M0 specifică cele 8 relee de la M0 la M7. Domeniul permis este de la K1 (4 dispozitive) la K8 (32 de dispozitive).

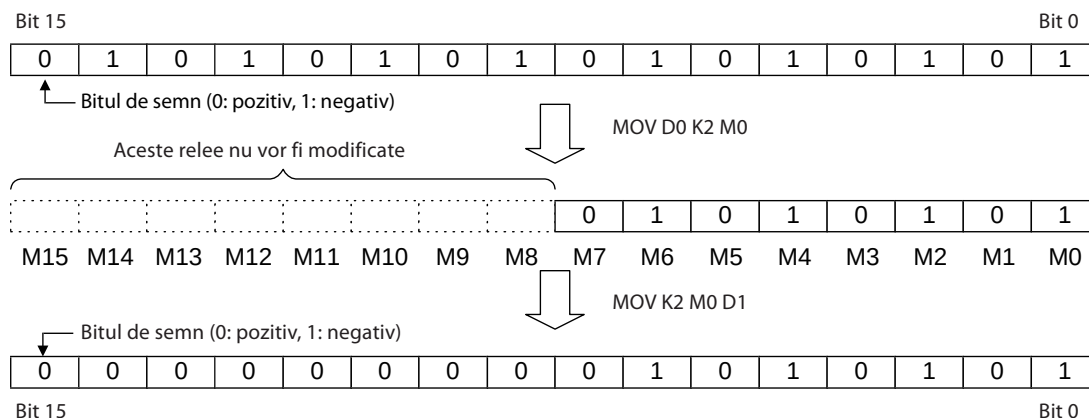
Exemple de adresare a grupurilor de dispozitive bit:

- K1X0: 4 intrări, începe la X0 (X0 la X3)
- K2X4: 8 intrări, începe la X4 (X4 la X13, notație în octal)
- K4M16: 16 relee, începe la M16 (M16 la M31)
- K3Y0: 12 ieșiri, începe la Y0 (Y0 la Y13, notație în octal)
- K8M0: 32 de relee, începând cu M0 (M0 la M31)

Adresarea mai multor dispozitive bit cu o singură instrucțiune face ca programarea să fie mai rapidă și permite generarea unor programe mai compacte. Următoarele două exemple transferă stările de semnal al releelor de la M0 la M3 în ieșirile Y10 – Y13:



Dacă domeniul destinație este mai mic decât domeniul sursă, biții excedentari sunt pur și simplu ignorați (a se vedea ilustrația următoare, exemplul de sus). Dacă destinația este mai mare decât sursa, dispozitivele în exces sunt umplute cu "0". Rețineți că atunci când se întâmplă acest lucru, rezultatul va fi întotdeauna pozitiv, deoarece bitul 15 este interpretat ca bitul de semn (exemplul de jos din ilustrația următoare).



5.2.3 Mutarea blocurilor de date cu instrucțiunea BMOV

Instrucțiunea MOV descrisă în secțiunea 5.2.1 poate scrie doar valori pe 16 sau 32 de biți într-o destinație. Dacă doriți, puteți programa mai multe secvențe de instrucțiuni MOV pentru a muta blocuri adiacente de date. Este însă mai eficient să utilizați instrucțiunea BMOV (B de la bloc și MOV de la move, mutare), care este furnizată exact în acest scop.

Schemă cu contacte

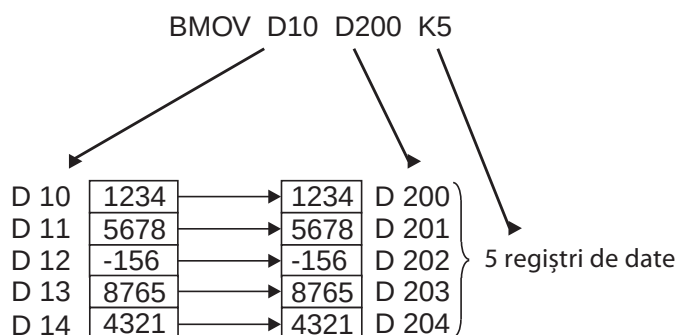


Listă de instrucțiuni



- ❶ Sursa de date (dispozitiv pe 16 biți, primul dispozitiv din domeniul sursă)
- ❷ Destinația datelor (dispozitiv pe 16 biți, primul dispozitiv din domeniul destinație)
- ❸ Numărul de elemente de mutat (max. 512)

Exemplul de mai sus funcționează astfel:

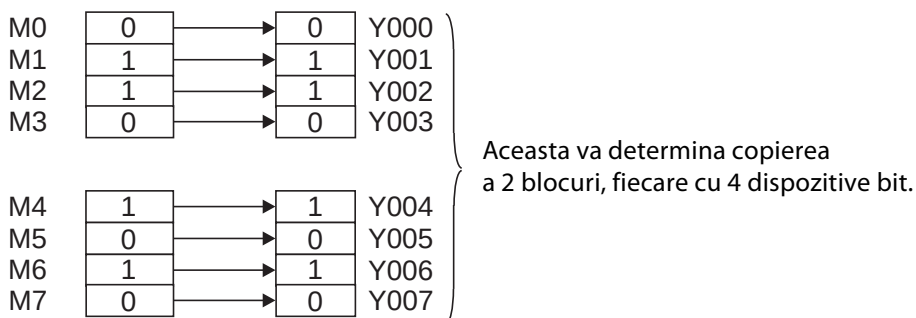


BMOV are de asemenea o versiune declanșată de impuls, BMOVP (a se vedea Secțiunea 5.1.2 pentru detalii privind execuția declanșată de impuls).

Blocurile de dispozitive bit: Atunci când mutați blocuri de dispozitive bit cu BMOV, factorii K ai sursei de date și ai destinației datelor trebuie să fie întotdeauna identici.

Exemplu

BMOV K1M0 K1Y0 K2



5.2.4 Copierea unui dispozitiv sursă în mai multe destinații (FMOV)

Instrucțiunea FMOV (F de la fill, umplere, și MOV de la move, mutare) copiază conținutul unui dispozitiv cuvânt sau al unui dispozitiv cuvânt dublu în mai multe dispozitive cuvânt sau cuvânt dublu, consecutive. Este utilizat în general pentru ștergerea datelor din tabele și pentru setarea datelor înregistrate la o valoare predefinită de pornire.

Schemă cu contacte

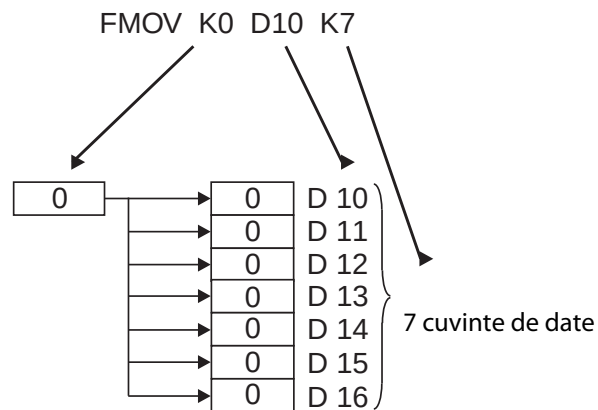


Listă de instrucțiuni



- ❶ Datele de scris în dispozitivele destinație (se pot utiliza și constante)
- ❷ Destinația datelor (primul dispozitiv al domeniului destinație)
- ❸ Numărul de elemente de scris în domeniul destinație (max. 512)

Următorul exemplu scrie valoarea "0" în 7 elemente:



FMOV are de asemenea o versiune declanșată de impuls, FMOV_P (a se vedea secțiunea 5.1.2 pentru detalii privind execuția declanșată de puls).

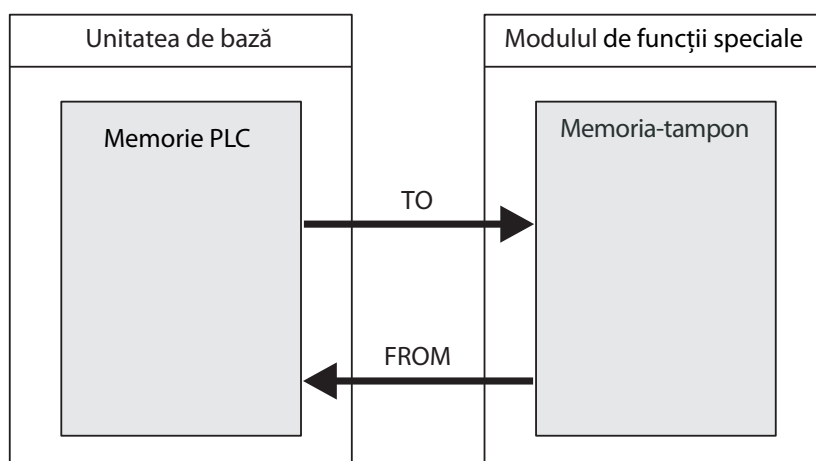
Puteți de asemenea să transferați date pe 32 de biți prin prefixarea instrucțiunii cu "D" (DFMOV și DFMOV_P).

5.2.5 Schimbul de date cu modulele de funcții speciale

Puteți adăuga module de funcții speciale pentru a crește numărul de intrări și ieșiri disponibile. În toate unitățile de bază ale seriei MELSEC FX, cu excepția modelelor FX1S. În plus, puteți de asemenea să suplimentați funcțiile controlerului prin adăugarea așa-numitelor "module de funcții speciale", de exemplu pentru citirea semnalelor analogice pentru curenți și tensiuni, pentru controlul temperaturii și pentru comunicațiile cu echipamente externe.

Modulele de extensie digitale de intrare/ieșire nu necesită instrucțiuni speciale; intrările și ieșirile suplimentare sunt tratate exact în același mod ca cele de pe unitatea de bază. Comunicațiile dintre unitatea de bază și modulele cu funcții speciale sunt efectuate cu ajutorul a două instrucțiuni aplicate specifice: instrucțiunile FROM și TO.

Fiecare modul de funcții speciale are un domeniu de memorie atribuit ca memorie-tampon pentru stocarea temporară a datelor, cum ar fi valorile măsurătorilor analogice sau datele primite. Unitatea de bază accesează această memorie-tampon, putând să citească valorile stocate în ea și să scrie noi valori în memoria-tampon, valori pe care modulul le poate apoi procesa (setările pentru funcțiile modulelor, datele pentru transmisii etc.)



Memoria-tampon poate avea maximum 23.767 celule individuale de memorie adresabile, iar fiecare celulă poate stoca 16 biți de date. Funcțiile celulelor memoriei-tampon depind de modulul respectiv - pentru detalii, consultați documentația modulului.

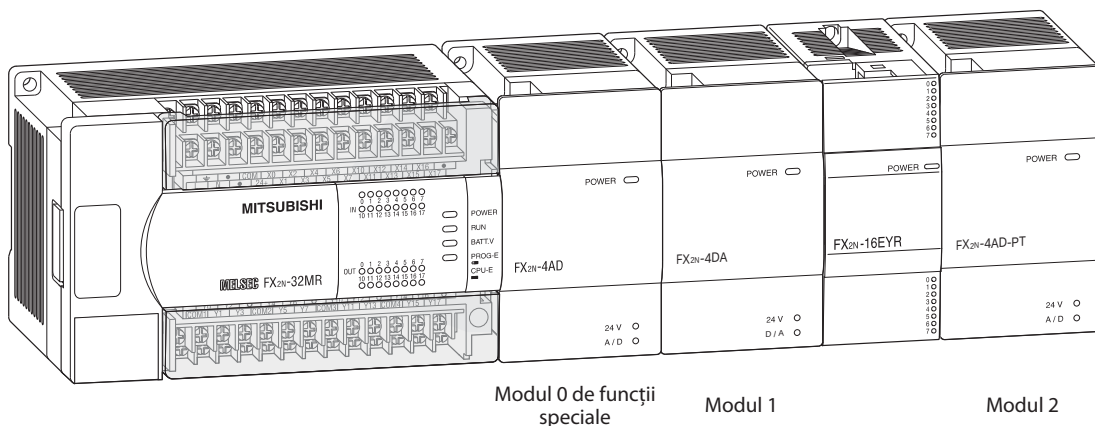
Adresă 0 memorie-tampon
Adresă 1 memorie-tampon
Adresă 2 memorie-tampon
:
:
Adresă n-1 memorie-tampon
Adresă n memorie-tampon

Atunci când utilizați instrucțiunile FROM și TO, sunt necesare următoarele informații:

- Modulul cu funcție specială din care se citește sau în care se scrie
- Adresa primei celule din memoria-tampon care trebuie citită sau în care se va scrie
- Numărul de celule din memoria-tampon care vor fi citite sau în care se va scrie
- Locația, din unitatea de bază, în care datele din modul vor fi stocate sau care conține datele ce vor fi scrise în modul.

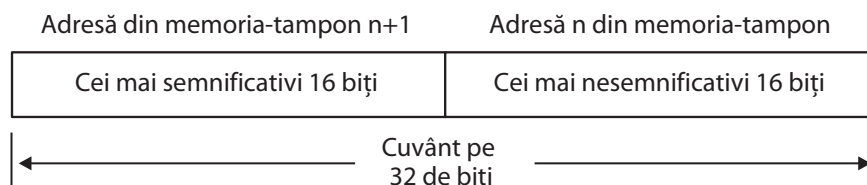
Adresa modulului de funcții speciale

Deoarece puteți atașa mai multe module de funcții speciale unui singur controler, fiecare modul trebuie să aibă un identificator unic, astfel încât să îl puteți adresa pentru a transfera datele în și din acesta. Fiecărui modul îi este atribuit automat un ID numeric între 0 și 7 (puteți conecta maximum 8 module de funcții speciale). Numerele sunt atribuite consecutiv, în ordinea în care sunt conectate modulele la automatul programabil.



Adresa de început din memoria-tampon

Fiecare dintre cele 32.767 de adrese din memoria-tampon poate fi adresată direct, în notație zecimală, în intervalul 0 – 32.767 (FX1N: 0 - 31). Atunci când accesați date pe 32 de biți, trebuie să rețineți că celulele de memorie cu adrese mai mici stochează cei 16 biți mai puțin semnificativi, iar celulele cu adrese mai mari stochează biții mai semnificativi.



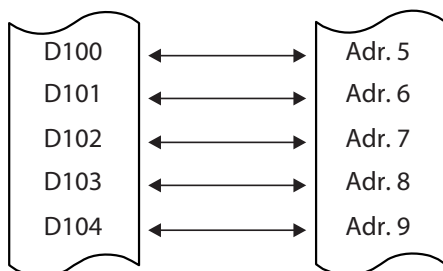
Aceasta înseamnă că adresa de început pentru datele pe 32 de biți este întotdeauna adresa ce conține cei mai puțin semnificativi 16 biți ai cuvântului dublu.

Numărul de unități de date de transferat

Cantitatea de date este definită de numărul de unități de date de transferat. Atunci când executați o instrucțiune FROM sau TO ca instrucțiune pe 16 biți, acest parametru este numărul de cuvinte de transferat. În cazul versiunilor pe 32 de biți, parametrul instrucțiunilor DFROM sau DTO specifică numărul de cuvinte duble de transferat.

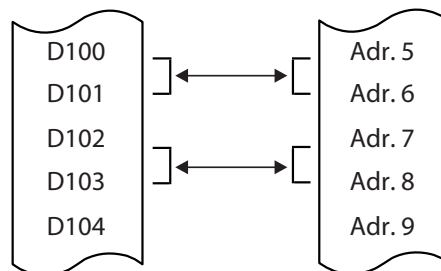
Instrucțiune pe 16 biți

Unități de date: 5



Instrucțiune pe 32 biți

Unități de date: 2



Valoarea pe care o puteți introduce pentru numărul de unități de date depinde de modelul de automat programabil pe care îl utilizați și de versiunea instrucțiunii FROM utilizate, pe 16 sau pe 32 de biți:

Model automat programabil	Domeniu valabil pentru numărul de unități de date de transferat	
	Instrucțiune pe 16 biți (FROM, TO)	Instrucțiune pe 32 de biți (DFROM, DTO)
FX2N	1 – 32	1 – 16
FX2NC	1 – 32	1 – 16
FX3U	1 – 32767	1 – 16383

Destinația / sursa datelor din unitatea de bază

În majoritatea cazurilor, va trebui să citiți date din regiștri și să le scrieți într-un modul de funcții speciale sau să copiați date din memoria-tampon a modului în regiștrii de date din unitatea de bază. Puteți însă de asemenea să utilizați ieșiri, relee și valorile curente ale temporizatoarelor și contoarelor ca surse și destinații de date.

Execuția pe front a instrucțiunilor

Dacă adăugați sufixul P la instrucțiuni, transferul de date va fi inițiat de declanșarea unui impuls (pentru detalii, a se vedea descrierea instrucțiunii MOV, din secțiunea 5.2.1).

Modul de utilizare al instrucțiunii FROM

Instrucțiunea FROM este utilizată pentru transferul datelor din memoria-tampon a unui modul de funcții speciale în unitatea de bază a controlerului. Rețineți că aceasta este o operație de copiere: conținutul datelor din memoria-tampon a modului nu va fi modificat.

Schemă cu contacte



Listă de instrucțiuni



- ❶ Adresă modul de funcții speciale (între 0 și 7)
- ❷ Adresa de început din memoria-tampon (FX1N: 0 - 31, FX2N, FX2NC și FX3U: 0 - 32,766). Puteți utiliza o constantă sau un registru de date care să conțină valoarea.
- ❸ Destinația datelor în unitatea de bază
- ❹ Numărul de unități de date de transferat

Exemplul de mai sus utilizează instrucțiunea FROM pentru a transfera date din modulul de intrări analogice FX2N-4AD, cu adresa 9. Instrucțiunea citește valoarea curentă a canalului 1 de la adresa 9 din memoria-tampon și o scrie în registrul de date D0.

Următorul exemplu arată modul în care versiunea pe 32 de biți a instrucțiunii este utilizată pentru a citi datele din adresa 2 din modulul de funcții speciale. Instrucțiunea citește 4 cuvinte duble începând de la adresa 8 din memoria-tampon și le scrie în regiștrii de date D8 – D15.



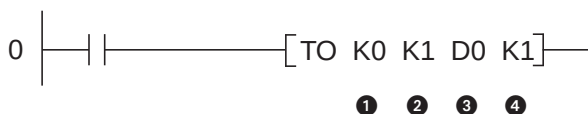
Următorul exemplu ilustrează utilizarea versiunii cu acțiune pe front, FROMP. În acest caz, conținutul adreselor 0 - 3 din memoria-tampon, este transferat în regiștrii de date D10 – D13 doar atunci când starea semnalului condiției de intrare se schimbă de la "0" la "1".



Modul de utilizare al instrucțiunii TO

Instrucțiunea TO transferă date din unitatea de bază a controlerului în memoria-tampon a unui modul cu funcție specială. Rețineți că aceasta este o operație de copiere, care nu alterează datele din locația sursă.

Schemă cu contacte



Listă de instrucțiuni

0	TO	K0	K1	D0	K1
		①	②	③	④

- ① Adresa modulului de funcții speciale (între 0 și 7)
- ② Adresa de început din memoria-tampon (FX1N: 0 - 31, FX2N, FX2NC și FX3U: 0 - 32,766). Puteți utiliza o constantă sau un registru de date care să conțină valoarea.
- ③ Sursa de date din unitatea de bază a controlerului
- ④ Numărul de unități de date de transferat

În exemplul de mai sus, conținutul registrului de date D0 este copiat la adresa 1 din memoria-tampon a modulului 0 de funcții speciale.

5.3 Instrucțiuni de comparare

Verificarea stării dispozitivelor bit, cum ar fi intrările și releele, poate fi efectuată cu instrucțiuni logice elementare, deoarece aceste dispozitive pot avea doar două stări: "0" și "1". Cu toate acestea, adesea va fi necesar să verificați conținutul dispozitivelor cuvânt înainte de a executa o acțiune, de exemplu pornirea unui ventilator de răcire ar trebui să aibă loc atunci când este depășită temperatura de referință. Controlerele din seria MELSEC FX oferă mai multe moduri de comparare a datelor.

5.3.1 Instrucțiunea CMP

CMP compară două valori numerice, care pot fi constantele sau conținutul regiștrilor de date. Puteți de asemenea compara valorile curente ale temporizatoarelor și contoarelor. În funcție de rezultatul comparației (mai mare decât, mai mic decât sau egal cu), este setat unul din cele trei dispozitive bit.

Schemă cu contacte



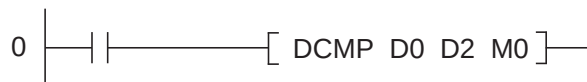
Listă de instrucțiuni

0	LD	..①		
1	CMP	D0	K100	M0
		②	③	④

- ① Condiția de intrare
- ② Prima valoare de comparat
- ③ A doua valoare de comparat
- ④ Unul dintre cele trei relee de ieșire consecutive, va fi setat (starea de semnal "1"), în funcție de rezultatul comparației:
 1. Dispozitiv 1: ON dacă Valoarea 1 > Valoarea 2
 2. Dispozitiv 2: ON dacă Valoarea 1 = Valoarea 2
 3. Dispozitiv 3: ON dacă Valoarea 1 < Valoarea 2

În acest exemplu, instrucțiunea CMP controlează releele M0, M1 și M2. M0 este "1" în cazul în care conținutul registrului D0 este mai mare decât 100; M1 este "1" în cazul în care conținutul registrului D0 este exact 100, iar M2 este "1" dacă D0 are un conținut a cărui valoare este mai mică decât 100. Starea celor trei dispozitive de tip bit este păstrată chiar și după dezactivarea condiției de intrare, deoarece a fost stocată ultima lor stare.

Pentru a compara date pe 32 de biți, trebuie să utilizați DCMP în loc de CMP:

Schemă cu contacteListă de instrucțiuni

```

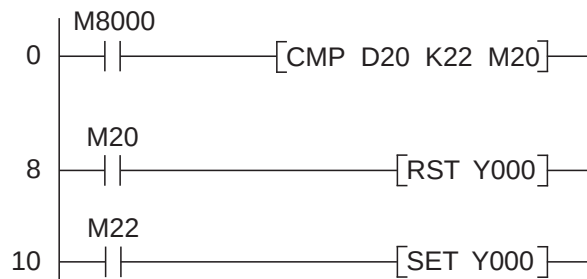
0 LD      ....
1 DCMP    D0      D2      M0

```

În exemplul de mai sus, conținutul regiștrilor D0 și D1 este comparat cu conținutul regiștrilor D2 și D3. Tratarea acestor trei dispozitive bit ce indică rezultatul comparației este exact aceeași ca pentru versiunea pe 16 biți a instrucțiunii.

Exemplu de aplicație

Este simplu de creat o buclă de control în două puncte, cu ajutorul instrucțiunii CMP:

Schemă cu contacteListă de instrucțiuni

```

0 LD      M8000
1 CMP     D20      K22      M20
8 LD      M20
9 RST     Y000
10 LD     M22
11 SET    Y0001

```

În acest exemplu, instrucțiunea CMP este executată ciclic. M8000 este întotdeauna "1" atunci când automatul programabil execută programul. Registrul D20 conține valoarea temperaturii curente a camerei. Constanta K22 este valoarea de referință, 22°C. Releele M20 și M22 arată când temperatura urcă sau coboară față de valoarea de referință. Dacă în cameră este prea cald, ieșirea Y0 este închisă. Dacă temperatura este prea scăzută, M22 va comuta ieșirea Y0 la loc în poziția deschisă. Această ieșire poate fi utilizată, de exemplu, pentru controlul unei pompe care să pompeze apă fierbinte.

5.3.2 Comparații în operațiile logice

În instrucțiunea CMP descrisă în secțiunea precedentă, rezultatul comparației este stocat în trei dispozitive bit. Adesea însă veți dori doar să executați o instrucțiune de ieșire sau o operație logică pe baza rezultatului unei comparații, și în general nu veți dori să utilizați trei dispozitive bit pentru acest lucru. Puteți face acest lucru cu instrucțiunile "încărcare comparație" și cu comparațiile logice, AND și OR.

Compararea la începutul unei operații logice

Schemă cu contacte



Listă de instrucțiuni

0 LD>= D40 D50
1 2 3

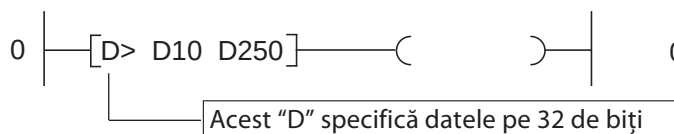
- ① Condiție de comparare
- ② Prima valoare de comparat
- ③ A doua valoare de comparat

În cazul în care condiția este evaluată ca fiind adevărată, după efectuarea comparației, starea semnalului este setată la "1". O stare de semnal "0" arată faptul că în urma comparației, condiția a fost evaluată ca fiind falsă. Sunt posibile următoarele comparații:

- Compararea pentru a se stabili egalitatea: = (valoare 1 = valoare 2)
leșirea din instrucțiune este setată la "1" doar dacă valorile din ambele dispozitive sunt egale.
- Comparare pentru a se stabili dacă o valoare este mai mare decât alta: > (valoare 1 > valoare 2)
leșirea din instrucțiune este setată la "1" doar dacă prima valoare este mai mare decât cea de-a doua valoare.
- Comparare pentru a se stabili dacă o valoare este mai mică decât alta: < (valoare 1 < valoare 2)
leșirea din instrucțiune este setată la "1" doar dacă prima valoare este mai mică decât cea de-a doua valoare.
- Comparare pentru a se stabili dacă valorile sunt diferite: <> (valoare 1 <> valoare 2)
leșirea din instrucțiune este setată la "1" doar dacă cele două valori sunt diferite.
- Compararea pentru a se stabili dacă o valoare este mai mică sau egală cu alta: <= (valoare 1 < valoare 2)
leșirea din instrucțiune este setată la "1" doar dacă prima valoare este mai mică sau egală decât a doua valoare.
- Compararea pentru a se stabili dacă o valoare este mai mare sau egală cu alta: >= (valoare 1 > valoare 2)
leșirea din instrucțiune este setată la "1" doar dacă prima valoare este mai mare sau egală decât cea de-a doua valoare.

Pentru a compara date pe 32 de biți, adăugați prefixul D (de la cuvânt dublu) la condiția de comparare):

Schemă cu contacte



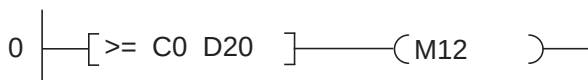
Listă de instrucțiuni

0 LDD> D10 D250

Exemplul de mai sus verifică dacă conținutul regiștrilor de date D10 și D11 este mai mare decât conținutul regiștrilor D250 și D251.

Alte exemple:

Schemă cu contacte

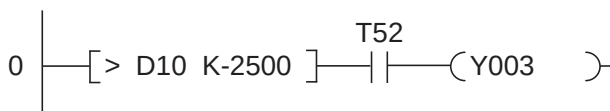


Listă de instrucțiuni

```
0 LD>= C0 D20
5 OUT M12
```

Releul M12 este setat la "1" atunci când valoarea contorului C0 este egală sau mai mare decât conținutul registrului D20.

Schemă cu contacte

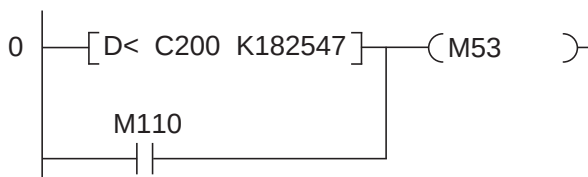


Listă de instrucțiuni

```
0 LD> D10 K-2500
5 AND T52
6 OUT Y003
```

Ieșirea Y003 este comutată la închis când conținutul registrului D10 este mai mare decât -2500, iar temporizatorul T52 și-a atins valoarea setată.

Schemă cu contacte



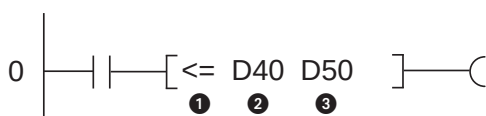
Listă de instrucțiuni

```
0 LDD< C200 K182547
9 OR M110
10 OUT M53
```

Releul M53 este setat la "1" fie dacă valoarea contorului C200 este mai mică decât 182.547, fie dacă releul M110 este setat la "1".

Comparația ca operație logică AND

Schemă cu contacte



Listă de instrucțiuni

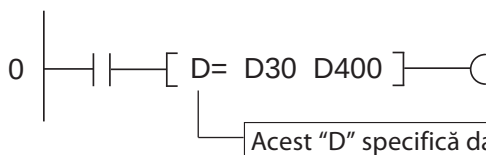
```
0 LD ...
1 AND<= D40 D50
1 2 3
```

- ❶ Condiția de comparat
- ❷ Prima valoare de comparare
- ❸ A doua valoare de comparare

O comparație AND poate fi utilizată ca o instrucțiune AND obișnuită (a se vedea capitolul 3).

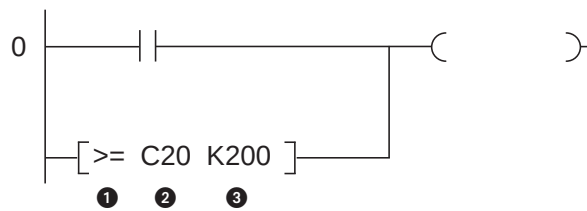
Opțiunile de comparare sunt aceleași ca cele descrise mai sus pentru o comparație la începutul unei operații. Puteți și în acest caz să comparați valori pe 32 de biți, printr-o operație AND:

Schemă cu contacte



Listă de instrucțiuni

```
0 ANDD= D30 D400
```

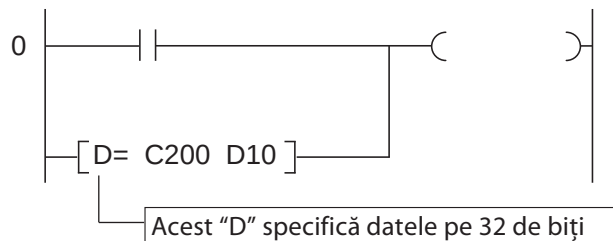
Comparația ca operație logică ORSchemă cu contacteListă de instrucțiuni

0	LD	...	
1	OR>=	C20	K200
	①	②	③

- ① Condiția de comparat
- ② Prima valoare de comparare
- ③ A doua valoare de comparare

O comparație OR poate fi utilizată ca o instrucțiune OR obișnuită (a se vedea capitolul 3).

Opțiunile de comparare sunt aceleași ca cele descrise mai sus pentru o comparație la începutul unei operații. Puteți și aici compara valori pe 32 de biți printr-o operație OR:

Schemă cu contacteListă de instrucțiuni

0	LD	...	
1	ORD=	C200	D10

5.4 Instrucțiuni matematice

Toate controlerele din seria MELSEC FX pot efectua cele patru operații aritmetice elementare și pot aduna, scădea, înmulți și împărți numere întregi (adică numere ce nu sunt în virgulă mobilă). Aceste instrucțiuni sunt descrise în secțiunea de față.

Unitățile de bază ale controlerelor din seriile FX2N, FX2NC și FX3U pot să proceseze și numere în virgulă mobilă. Acest lucru este efectuat prin intermediul instrucțiunilor speciale documentate în detaliu în Manualul de programare al seriei MELSEC FX.

După fiecare adunare sau scădere, trebuie să programați întotdeauna instrucțiunile din program să verifice stările releelor speciale listate mai jos, pentru a vedea dacă rezultatul este 0 sau a depășit intervalul de valori permis.

- **M8020**

Acest releu special este setat la "1" dacă rezultatul unei adunări sau scăderi este 0.

- **M8021**

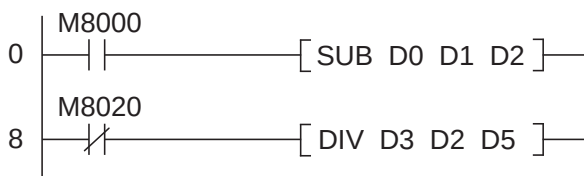
Releul special M8021 este setat la "1" dacă rezultatul unei adunări sau scăderi este mai mic decât -32.767 (operații pe 16 biți) sau -2.147.483.648 (operații pe 32 de biți).

- **M8022**

Releul special M8022 este setat la "1" dacă rezultatul unei adunări sau scăderi este mai mare decât +32.767 (operații pe 16 biți) sau +2.147.483.647 (operații pe 32 de biți).

Aceste releee speciale pot fi utilizate ca marcaje de permisiune pentru a se putea continua cu alte operații matematice. În exemplul următor, rezultatul operației de scădere din D2 este utilizat ca divizor. Deoarece împărțirea la 0 este imposibilă și duce la eroare, împărțirea este efectuată doar dacă divizorul este diferit de 0.

Schemă cu contacte



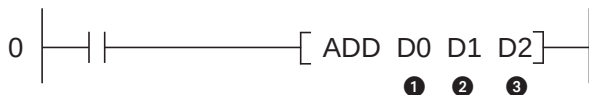
Listă de instrucțiuni

0	LD	M8000		
1	SUB	D0	D1	D2
8	LDI	M8020		
9	DIV	D3	D2	D5

5.4.1 Adunarea

Instrucțiunea ADD calculează suma a două valori pe 16 sau pe 32 de biți și scrie rezultatul în alt dispozitiv.

Schemă cu contacte



Listă de instrucțiuni

0 ADD D0 D1 D2
 ① ② ③

- ① Primul dispozitiv sursă sau prima constantă
- ② Al doilea dispozitiv sursă sau a doua constantă
- ③ Dispozitiv în care este stocat rezultatul adunării

Exemplul de mai sus adună conținutul lui D0 și D1 și scrie rezultatul în D2.

Exemple

Adăugarea constantei 1000 la conținutul registrului de date D100:

ADD K1000 D100 D102 $\rightarrow$ 1000 + $\boxed{\begin{smallmatrix} \text{D 100} \\ 53 \end{smallmatrix}}$ $\rightarrow$ $\boxed{\begin{smallmatrix} \text{D 102} \\ 1053 \end{smallmatrix}}$

Instrucțiunea ADD ia în considerare semnele valorilor adunate:

ADD D10 D11 D12 $\rightarrow$ $\boxed{\begin{smallmatrix} \text{D 10} \\ 5 \end{smallmatrix}}$ + $\boxed{\begin{smallmatrix} \text{D 11} \\ -8 \end{smallmatrix}}$ $\rightarrow$ $\boxed{\begin{smallmatrix} \text{D 12} \\ -3 \end{smallmatrix}}$

Puteți de asemenea să adunați valori pe 32 de biți, prin adăugarea prefixului "D" la instrucțiunea ADD (DADD):

DADD D0 D2 D4 $\rightarrow$ $\boxed{\begin{smallmatrix} \text{D 1} & \text{D 0} \\ 65238 \end{smallmatrix}}$ + $\boxed{\begin{smallmatrix} \text{D 3} & \text{D 2} \\ 27643 \end{smallmatrix}}$ $\rightarrow$ $\boxed{\begin{smallmatrix} \text{D 5} & \text{D 4} \\ 92881 \end{smallmatrix}}$

Dacă doriți, puteți să și scrieți rezultatul într-unul din dispozitivele sursă. Dacă însă faceți acest lucru, nu uitați că rezultatul se va modifica în fiecare ciclu de program dacă instrucțiunea ADD este executată ciclic!

ADD D0 K25 D0 $\rightarrow$ $\boxed{\begin{smallmatrix} \text{D 0} \\ 18 \end{smallmatrix}}$ + 25 $\rightarrow$ $\boxed{\begin{smallmatrix} \text{D 0} \\ 43 \end{smallmatrix}}$

Instrucțiunea ADD poate fi executată și pe front. În acest caz, este executată doar atunci când starea semnalului condiției de intrare se schimbă de la "0" la "1". Pentru a utiliza acest mod, adăugați sufixul "P" la instrucțiunile ADD (ADDP, DADDP).

În exemplul următor, valoarea constantă 27 este adunată doar o singură dată la conținutul D47, în ciclul de program în care starea semnalului releului M47 se schimbă de la "0" la "1":

Schemă cu contacte



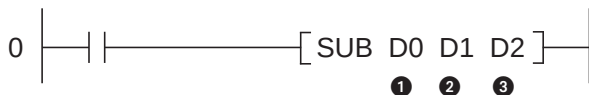
Listă de instrucțiuni

0 LD M47
 1 ADDP D47 K27 D51

5.4.2 Scăderea

Instrucțiunea SUB calculează diferența dintre două valori numerice (conținuturi ale unor dispozitive pe 16 sau 32 de biți sau ale unor constante). Rezultatul scăderii este scris într-un al treilea dispozitiv.

Schemă cu contacte



Listă de instrucțiuni



❶ Descăzut (aceasta este valoarea din care se scade scăzătorul)

❷ Scăzător (aceasta este valoarea care se scade din scăzut)

❸ Diferența (rezultatul scăderii)

În exemplul de mai sus, conținutul lui D1 este scăzut din conținutul lui D0, iar diferența este scrisă în D2.

Exemple

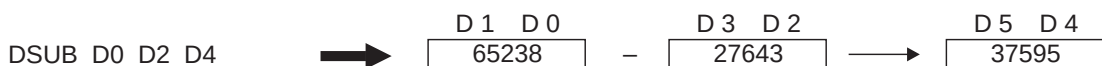
Se scade 100 din conținutul registrului de date D100, iar rezultatul este scris în D101:



Instrucțiunea SUB ia în calcul semnele valorilor:



Puteți de asemenea scădea valori pe 32 de biți prin adăugarea prefixului "D" la instrucțiunea SUB (DSUB):



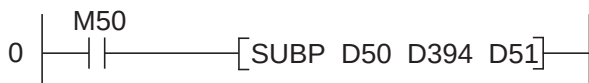
Dacă doriți, puteți să scrieți rezultatul și într-unul din dispozitivele sursă. Dacă faceți însă acest lucru, nu uitați că, în acest caz, rezultatul se va schimba la fiecare ciclu de program, dacă instrucțiunea SUB este executată ciclic!



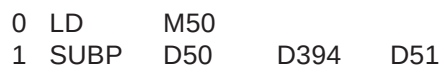
Instrucțiunea SUB poate fi de asemenea executată **pe front**. În acest caz, va fi executată doar atunci când starea semnalului condiției de intrare se va schimba de la "0" la "1". Pentru a utiliza acest mod, adăugați sufixul "P" la instrucțiunile SUB (SUBP, DSUBP).

În exemplul următor, conținutul registrului D394 este scăzut din conținutul lui D50 o singură dată, în ciclul de program în care starea semnalului releului M50 se schimbă de la "0" la "1":

Schemă cu contacte



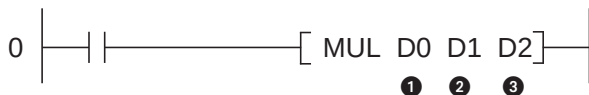
Listă de instrucțiuni



5.4.3 Înmulțirea

Instrucțiunea MUL din controlerele FX înmulțește două valori pe 16 sau 32 de biți și scrie rezultatul într-un al treilea dispozitiv.

Schemă cu contacte



Listă de instrucțiuni



- ❶ Deînmulțitul
- ❷ Înmulțitorul
- ❸ Dispozitiv în care este stocat rezultatul adunării

Exemplul de mai sus înmulțește conținutul lui D0 și D1 și scrie rezultatul în D2.

NOTĂ

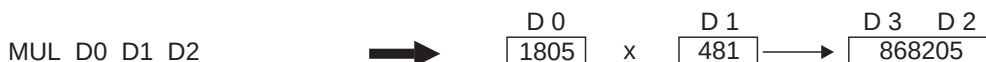
Atunci când înmulțiți două valori pe 16 biți, rezultatul poate cu ușurință să depășească domeniul ce poate fi reprezentat pe 16 biți. Din această cauză, produsul înmulțirii este întotdeauna scris în două dispozitive pe 16 biți consecutive (adică un cuvânt dublu pe 32 de biți).

Atunci când înmulțiți două valori pe 32 de biți, produsul este scris în patru dispozitive pe 16 biți (64 de biți, două cuvinte duble).

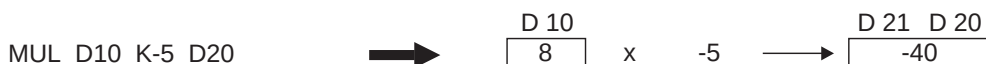
Luați întotdeauna în considerare dimensiunea intervalelor acestor dispozitive atunci când scrieți programul și aveți grijă să nu suprapuneți intervalele unele cu altele, prin utilizarea de dispozitive din intervalul în care sunt scrise produsele.

Exemple

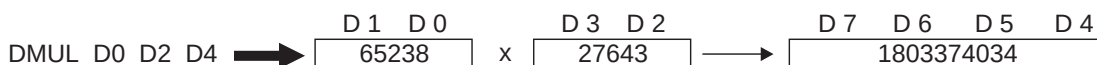
Înmulțirea conținutului lui D0 și D1 și stocarea produsului în D3 și D2:



Instrucțiunea MUL ia în considerare semnele valorilor. În acest exemplu, valoarea din D10 este înmulțită cu constanta -5:



Puteți de asemenea să înmulțiți valori pe 32 de biți, prin adăugarea prefixului "D" la instrucțiunea MUL (DMUL):

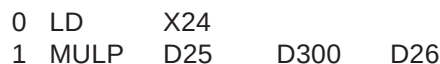


Instrucțiunea MUL poate fi executată și pe front, prin adăugarea sufixului "P" la instrucțiunile MUL (MULP, DMULP). Următoarea înmulțire este executată doar atunci când intrarea X24 este comutată de la "0" la "1":

Schemă cu contacte



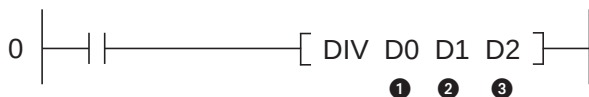
Listă de instrucțiuni



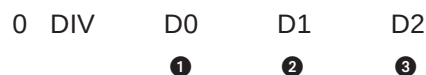
5.4.4 Împărțirea

Instrucțiunea DIV a seriei MELSEC FX divide un număr prin altul (conținutul a două dispozitive pe 16 sau 32 de biți sau a două constante). Aceasta este o operație cu numere întregi, nu se pot procesa valori în virgulă mobilă. Rezultatul este întotdeauna un număr întreg, iar restul este stocat separat.

Schema cu contacte



Listă de instrucțiuni



- ❶ Deîmpărțitul
- ❷ Împărțitor
- ❸ Câțul (rezultatul împărțirii, deîmpărțitul ÷ împărțitorul = câțul)

NOTĂ

Împărțitorul nu trebuie să fie niciodată 0. Împărțirea la 0 nu este posibilă și operația va genera o eroare.

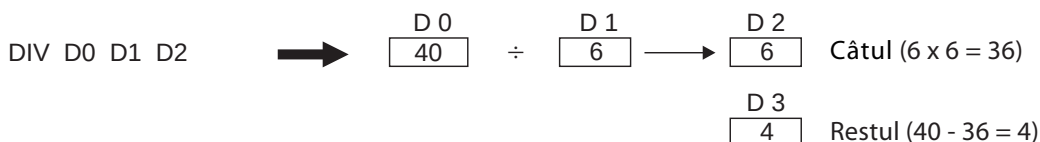
Atunci când se împart două valori pe 16 biți, câțul este scris într-un dispozitiv pe 16 biți, iar restul este scris în dispozitivul următor. Aceasta înseamnă că rezultatul unei împărțiri necesită întotdeauna două dispozitive consecutive pe 16 biți (= 32 de biți).

Atunci când împărțiți două valori pe 32 de biți, câțul este scris în două dispozitive pe 16 biți, iar restul este scris în următoarele două dispozitive pe 16 biți. Aceasta înseamnă că pentru rezultatul unei împărțiri pe 32 de biți, sunt necesare patru dispozitive consecutive pe 16 biți.

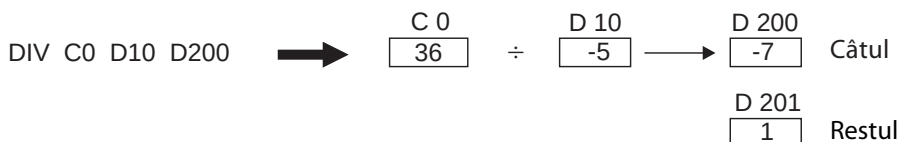
Luați întotdeauna în considerare dimensiunea acestor intervale de dispozitive atunci când scrieți un program și aveți grijă să nu creați suprapuneri de intervale prin utilizarea dispozitivelor din intervalele în care sunt scrise rezultatele calculelor.

Exemple

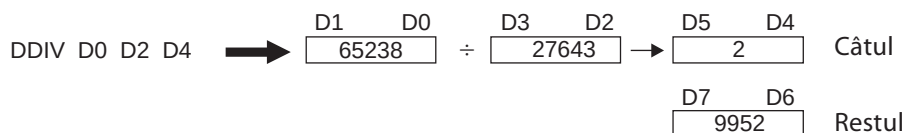
Se împarte conținutul lui D0 la conținutul lui D1 și se scrie rezultatul în D2 și D3.



Instrucțiunea DIV ia în calcul semnele valorilor. În acest exemplu, valoarea contorului din C0 este împărțită la valoarea din D10:



Împărțirea cu valori pe 32 de biți:



Adăugarea sufixului "P" la instrucțiunea DIV execută instrucțiunile pe front (DIV -> DIVP, DDIV -> DDIVP). În următorul exemplu, valoarea contorului C12 este împărțită la 4 doar în ciclul de program în care intrarea X30 este comutată pe închis:

Schemă cu contacte



Listă de instrucțiuni

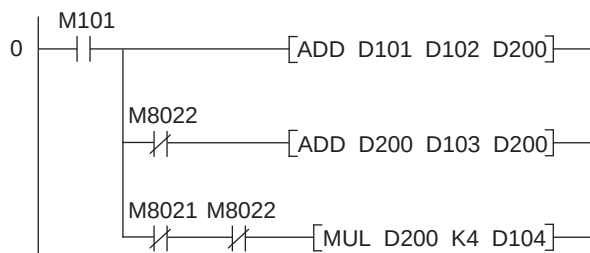
```
0 LD X30
1 DIVP C12 K4 D12
```

5.4.5 Combinarea instrucțiunilor matematice

În practică, adesea doriți să efectuați mai multe calcule. Controlerele FX vă permit să combinați instrucțiuni matematice pentru a putea rezolva calcule mai complexe. În funcție de natura calculului, s-ar putea să fie necesare dispozitive suplimentare care să stocheze rezultatele intermediare.

Exemplul următor arată cum puteți calcula suma valorilor din regiștrii de date D101, D102 și D103, iar apoi înmulți rezultatul cu factorul 4:

Schemă cu contacte



Listă de instrucțiuni

```
0 LD M101
1 ADD D101 D102 D200
8 MPS
9 ANI M8022
10 ADD D200 D103 D200
17 MPP
18 ANI M8021
19 ANI M8022
20 MUL D200 K4 D104
```

- Mai întâi sunt adunate conținutul lui D101 și al lui D102, iar rezultatul este stocat în D200.
- Dacă (și numai dacă) suma dintre conținutul lui D101 și al lui D102 nu depășește intervalul permis, este adunată cu valoarea din D103.
- Dacă suma regiștrilor D101 - D103 nu depășește intervalul permis, este apoi înmulțită cu factorul 4, iar rezultatul este scris în D104 și D105.

6 Opțiuni de extensie

6.1 Introducere

Puteți extinde unitățile de bază ale seriei MELSEC FX cu module de extensie și cu module de funcții speciale.

Aceste module se împart în trei categorii:

- Module ce ocupă intrări și ieșiri digitale (instalate pe partea dreaptă a controlerului). Acestea includ module I/O digitale compacte sau modulare și modulele de funcții speciale.
- Modulele ce nu ocupă intrări sau ieșiri digitale (instalate pe partea stângă a controlerului).
- Adaptoare de interfață și de comunicații, ce nu ocupă intrări și ieșiri digitale (instalate direct în unitatea controlerului).

6.2 Module disponibile

6.2.1 Module pentru suplimentarea numărului de intrări și ieșiri digitale

Sunt disponibile diverse module de extensie, compacte și modulare, pentru adăugarea de intrări și ieșiri la unitățile de bază ale modelelor MELSEC FX1N/FX2N/FX2NC și FX3U. În plus, intrările și ieșirile digitale pot fi de asemenea adăugate la controlerele din seriile FX1S, FX1N și FX3U cu adaptoare speciale ce se instalează direct în controler. Aceste adaptoare sunt o alegere foarte bună atunci când aveți nevoie doar de câteva intrări sau ieșiri suplimentare și nu aveți destul spațiu pentru a instala lateral module de extensie.

Unitățile de extensie “modulare” pot conține doar intrări și ieșiri digitale, nu au propriile surse de alimentare. Unitățile de extensie “compacte” au un număr mai mare de intrări și ieșiri și o sursă de alimentare integrată pentru bus-ul de sistem și pentru intrări.

Unitățile de bază pot fi combinate cu modulele de extensie disponibile într-o gamă variată de posibilități, ceea ce face posibilă configurarea sistemului dvs. într-un mod extrem de precis, adaptat necesităților aplicației.

6.2.2 Module de intrare/ieșire analogice

Modulele de intrare/ieșire analogice convertesc semnalele de intrare analogice în valori digitale sau valori numerice interne în semnale analogice.

Sunt disponibile mai multe module pentru semnale în curent sau tensiune și pentru monitorizarea temperaturilor, cu conectare directă la senzori Pt100 sau termo-elemente. A se vedea capitolul 7 pentru o introducere în procesarea semnalului analogic.

6.2.3 Module de comunicații

Mitsubishi Electric fabrică o gamă largă de module de interfață și adaptoare cu porturi seriale (RS-232, RS-422 și RS-485) pentru conectarea perifericelor sau a altor controlere.

Pentru integrarea modelelor MELSEC FX1N, FX2N, FX2NC și FX3U în diferite rețele, sunt disponibile mai multe module de comunicații speciale.

Modulele de interfață ENetwork sunt disponibile în prezent pentru Profibus/DP, AS-interface, DeviceNet, CANopen, CC-Link și pentru rețelele dezvoltate de Mitsubishi.

6.2.4 Module de poziționare

Puteți complementa contoarele interne de mare viteză ale controlerelor MELSEC FX cu module suplimentare de contorizare hardware, de mare viteză, externe, pe care le puteți utiliza pentru conectarea unor dispozitive cum ar fi traductorii de turație incrementală și modulele de poziționare pentru sistemele de acționare cu servomotor sau cu motoare pas cu pas.

Cu ajutorul seriei MELSEC FX și a modulelor de poziționare cu trenuri de impulsuri, puteți programa aplicații de poziționare de mare precizie. Aceste module pot fi utilizate și pentru controlul sistemelor de acționare cu servomotor sau cu motoare pas cu pas.

6.2.5 Terminale de operare HMI

Terminalele de operare Mitsubishi Electric oferă o interfațare eficientă și intuitivă cu controlerele din seria MELSEC FX. Unitățile de control HMI fac ca funcțiile aplicației controlate să fie transparente și ușor de înțeles.

Toate unitățile disponibile pot monitoriza și edita oricare dintre parametrii automatului programabil, cum ar fi valorile efective și de referință ale temporizatoarelor, contoarelor, regiștrilor de date, precum și instrucțiunile secvențiale.

Unitățile de control HMI sunt disponibile în variantele cu tastatură sau cu ecran senzorial. Taste funcționale complet programabile și ecrane senzoriale fac ca aceste unități să fie mult mai ușor de utilizat. Mediul de programare al acestor terminale este ușor de utilizat și intuitiv și rulează sub Windows.

Terminalele HMI comunică cu automatele programabile FX prin portul de programare și sunt conectate direct cu un cablu standard, nefiind necesare module suplimentare.

7 Procesarea valorilor analogice

7.1 Module analogice

Atunci când automatizați procese, va fi adesea necesar să obțineți sau să controlați valori analogice, cum ar fi temperaturi, presiuni sau nivele de umplere. Fără module suplimentare, unitățile de bază din seria MELSEC FX pot procesa doar semnale de intrare și ieșire digitale (adică date de tipul ON/OFF). De aceea, pentru intrări și ieșiri analogice sunt necesare module analogice suplimentare.

Practic, există două tipuri de module analogice:

- Module de intrare analogice și
- Module de ieșire analogice.

Modulele de intrare analogice pot achiziționa valori pentru semnale în curent, tensiune sau semnale speciale de temperaturi. Modulele de ieșire analogice furnizează semnale în curent sau tensiune. În plus, există de asemenea module combinate, care pot și să obțină, și să transmită semnale analogice.

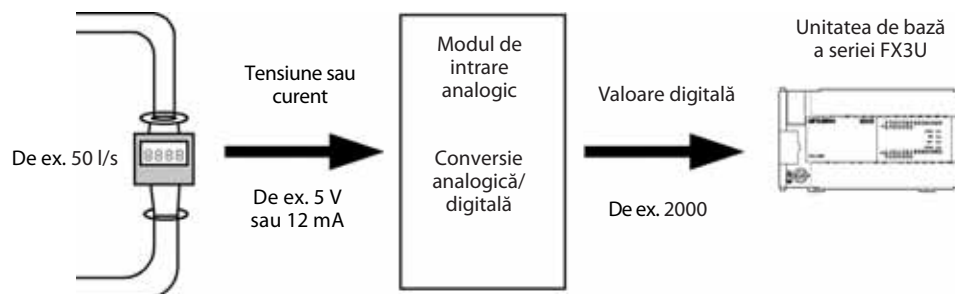
Module de intrare analogice

Modulele de intrare analogice convertesc o valoare analogică măsurată (de ex. 10V) într-o valoare digitală (de ex. 4000), ce poate fi procesată de către automatul programabil. Procesul de conversie este cunoscut drept conversie analogică/digitală sau, pe scurt, conversie A/D.

Temperaturile pot fi obținute direct de către modulele analogice din seria MELSEC FX, dar alte valori fizice cum ar fi presiunea sau viteza de curgere trebuie mai întâi convertite în valori de curent sau tensiune, înainte de a putea fi convertite în valori digitale care să poată fi procesate de automatul programabil. Această conversie este efectuată de senzori ce transmit semnale în intervale standardizate (de exemplu, între 0 și 10 V sau între 4 și 20 mA). Valoarea măsurată a unui semnal în curent prezintă o mai bună imunitate la perturbațiile generate de către lungimea cablurilor sau de către rezistențele de contact.

Următorul exemplu de obținere a unei valori analogice arată o soluție de măsurare a debitului cu un automat programabil din seria MELSEC FX.

Dispozitiv de măsurare a debitului
cu ieșire în tensiune sau curent



Module de intrare analogice pentru temperaturi

Valorile temperaturilor pot fi măsurate prin două metode: cu termocupluri sau senzori Pt100.

- Termometre cu rezistență Pt100

Aceste dispozitive măsoară rezistența unui element din platină, ce crește odată cu temperatura. La 0°C, elementul are o rezistență de 100 (de unde și denumirea Pt100). Senzorii de rezistență sunt conectați într-o configurație cu trei cabluri, ceea ce asigură faptul că rezistența cablurilor de conectare nu influențează rezultatul măsurătorii.

Domeniul cel mai extins de măsurare al termometrelor cu rezistență Pt100 este între -200°C și +600°C, dar în practică depinde și de capacitățile modulului de achiziție a datelor de temperatură utilizat.

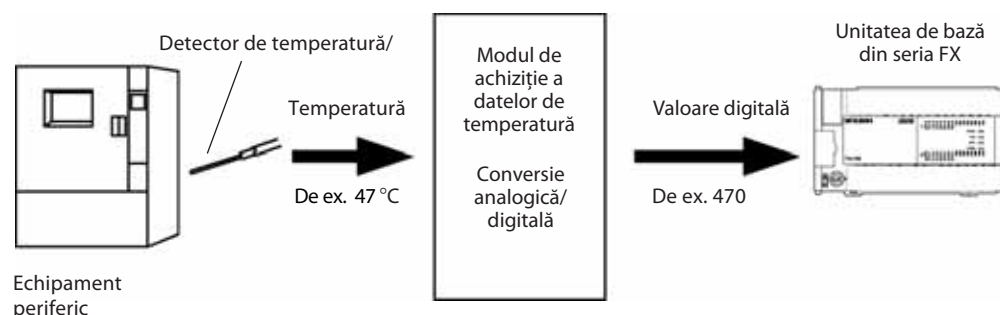
- Termocupluri

Aceste dispozitive de măsurare a temperaturii se bazează pe faptul că între două metale aflate în contact apare o tensiune proporțională cu temperatura. Această metodă măsoară deci temperatura prin intermediul unui semnal de tensiune.

Există mai multe tipuri de termocupluri. Ele diferă prin tensiunea termoelectrică și prin domeniul de temperaturi pe care le pot măsura. Combinațiile de materiale utilizate sunt standardizate și identificate printr-un cod specific. Tipurile J și K sunt foarte des utilizate. Termocuplurile de tip J utilizează fier (Fe) și un aliaj de cupru-nichel (CuNi), termocuplurile de tip K utilizează nichel-crom (NiCr) și nichel (Ni). În plus față de modul în care sunt construite, termocuplurile mai diferă și prin domeniile de temperatură pe care le pot măsura.

Termocuplurile pot fi utilizate pentru a măsura temperaturi între -200°C și +1.200°C.

Exemplu de măsurare a temperaturii:

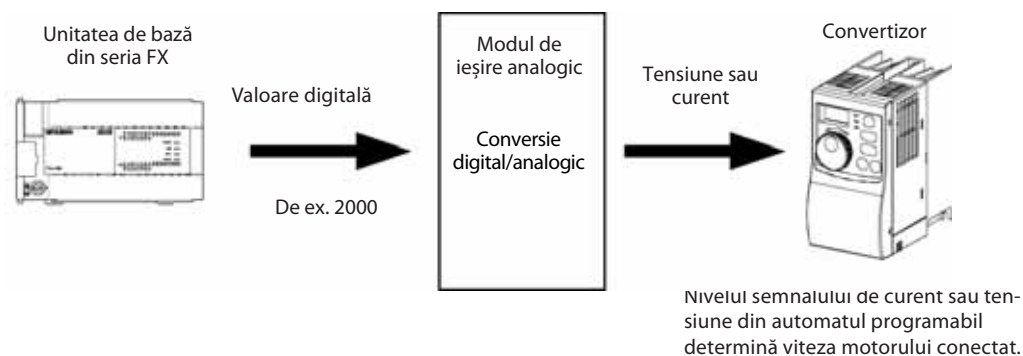


Module de ieșire analogice

Modulele de ieșire analogice convertesc o valoare digitală din unitatea de bază a automatului programabil într-o tensiune sau curent, ce pot fi apoi utilizate pentru a controla un dispozitiv extern (conversie digital/analogic sau D/A).

Semnalele de ieșire analogice generate de seria de controlere MELSEC FX sunt semnale unificate: 0-10 V și 4-20 mA.

Exemplul de pe pagina următoare arată utilizarea unui semnal analogic drept valoare de referință pentru acționarea unui convertizor de frecvență. În această aplicație, semnalul de curent sau tensiune din automatul programabil ajustează viteza motorului conectat la convertizorul de frecvență.



7.1.1 Criterii pentru selecția modulelor analogice

Modulele analogice pentru seria MELSEC FX sunt disponibile într-o gamă largă, pentru fiecare sarcină de automatizare trebuind să selectați modulul adecvat. Principalele criterii de selecție sunt următoarele:

- Compatibilitate cu unitatea de bază a automatului programabil

Modulul analogic trebuie să fie compatibil cu unitatea de bază a automatului programabil pe care îl utilizați. De exemplu, modulele analogice din seria FX3U nu pot fi conectate la o unitate de bază din seria FX1N.

- Rezoluția

Rezoluția este cea mai mică valoare fizică ce poate fi obținută sau transmisă de către modulul analogic.

În cazul modulelor cu intrare analogică, rezoluția este definită ca modificarea de tensiune, curent sau temperatură la intrare, ce duce la creșterea sau scăderea cu 1 a valorii ieșirii digitale.

În cazul modulelor cu ieșire analogică, rezoluția este modificarea de tensiune sau curent la ieșirea modulului, cauzată de creșterea sau scăderea cu 1 a valorii intrării digitale.

Rezoluția este restricționată de design-ul intern al modulelor analogice și depinde de numărul de biți necesari pentru stocarea valorii digitale. De exemplu, dacă se obține cu un convertor A/D pe 12 biți, o tensiune de 10 V, domeniul de tensiune este divizat în 4.096 de trepte ($2^{12} = 4096$, a se vedea Secțiunea 3.3). Aceasta va duce la o rezoluție de $10 \text{ V} / 4096 = 2,5 \text{ mV}$.

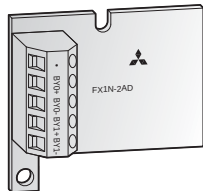
- Numărul de intrări sau ieșiri analogice

Intrările sau ieșirile modulelor analogice sunt denumite și canale. Puteți selecta module cu intrări analogice cu 2, 4 sau 8 canale, în funcție de numărul de canale de care aveți nevoie. Rețineți că există o limită de module cu funcții speciale pe care le puteți conecta la o unitate de bază a unui automat programabil (a se vedea secțiunea 7.1.2). Dacă știți că va trebui să instalați alte module cu funcții speciale, este deci mai bine să utilizați un singur modul cu patru canale, mai curând decât două module care au fiecare câte două canale, deoarece acest lucru vă permite conectarea mai multor module suplimentare.

Pentru seria de controlere MELSEC FX sunt disponibile mai multe tipuri de module analogice.

Plăci adaptoare

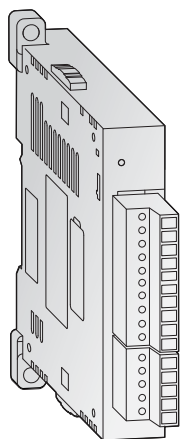
Plăcile adaptoare sunt mici plăci de circuite, ce se instalează direct în controlerele FX1S sau FX1N, ceea ce înseamnă că nu ocupă spațiu suplimentar în dulap.



Valorile digitale generate de semnalele ce vin din cele două canale de intrare ale adaptorului sunt scrise direct în regiștrii speciali D8112 și D8113, ceea ce face ca acestea să fie deosebit de ușor de procesat. Valoarea de ieșire pentru adaptorul de ieșire analogic este scrisă de program în regiștrul special D8114, iar apoi convertită de adaptor și trimisă la ieșire.

Adaptor special

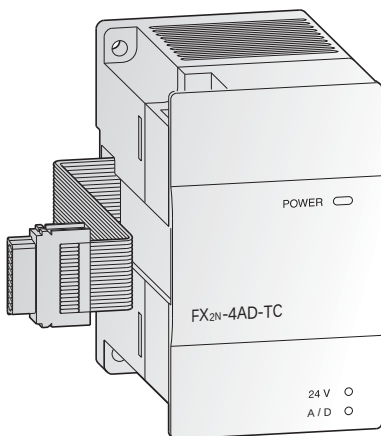
Adaptoarele speciale pot fi conectate doar pe partea stângă a unei unități de bază din seria MELSEC FX3U. Puteți instala maximum patru adaptoare speciale analogice.



Adaptoarele speciale nu utilizează puncte de intrare sau de ieșire din unitatea de bază. Ele comunică direct cu unitatea de bază, prin regiștri și relee speciale. Din acest motiv, nu sunt necesare în program instrucțiuni speciale de comunicare cu modulele de funcții speciale (a se vedea mai jos).

Modulele de funcții speciale

Pe partea dreaptă a fiecărei unități din seria MELSEC FX se pot conecta maximum opt module de funcții speciale.



În plus față de modulele analogice, modulele de funcții speciale disponibile includ module de comunicații, de poziționare și alte tipuri de module. Fiecare modul de funcții speciale ocupă opt puncte de intrare și opt puncte de ieșire pe unitatea de bază. Comunicația dintre modul și unitatea de bază a automatului programabil este efectuată prin memoria-tampon a modului cu instrucțiunile FROM și TO (a se vedea secțiunea 5.2.5).

7.2 Lista modulelor analogice

Tip modul		Specificație	Nr. de canale	Domeniu	Rezoluție	FX1S	FX1N	FX2N FX2NC	FX3U
Module de intrări analogice	Placă adaptoare	FX1N-2AD-BD	2	Tensiune: 0 V – 10 V DC	2.5 mV (12 biți)	●	●	○	○
				Curent: 4 mA – 20 mA DC	8 μA (11 biți)				
	Adaptor special	FX3U-4AD-ADP	4	Tensiune: 0 V – 10 V DC	2.5 mV (12 biți)	○	○	○	●
				Curent: 4 mA – 20 mA DC	10 μA (11 biți)				
	Bloc pentru funcție specială	FX2N-2AD	2	Tensiune: 0 V – 5 V DC 0 V – 10 V DC	2.5 mV (12 biți)	○	●	●	●
				Curent: 4 mA – 20 mA DC	4 μA (12 biți)				
		FX2N-4AD	4	Tensiune: -10 V – 10 V DC	5 mV (cu semn, 12 biți)	○	●	●	●
				Curent: 4 mA – 20 mA DC -20 mA – 20 mA DC	10 μA (cu semn, 11 biți)				
		FX2N-8AD*	8	Tensiune: -10 V – 10 V DC	0.63 mV (cu semn, 15 biți)	○	●	●	●
				Curent: 4 mA – 20 mA DC -20 mA – 20 mA DC	2.50 μA (cu semn, 14 biți)				
		FX3U-4AD	4	Tensiune: -10 V – 10 V DC	0.32 mV (cu semn, 16 biți)	○	○	○	●
				Curent: 4 mA – 20 mA DC -20 mA – 20 mA DC	1.25 μA (cu semn, 15 biți)				
Module de ieșiri analogice	Placă adaptoare	FX1N-1DA-BD	1	Tensiune: 0 V – 10 V DC	2,5 mV (12 biți)	●	●	○	○
				Curent: 4 mA – 20 mA DC	8 μA (11 biți)				
	Adaptor special	FX3U-4DA-ADP	4	Tensiune: 0 V – 10 V DC	2.5 mV (12 biți)	○	○	○	●
				Curent: 4 mA – 20 mA DC	4 μA (12 biți)				
	Bloc pentru funcție specială	FX2N-2DA	2	Tensiune: 0 V – 5 V DC 0 V – 10 V DC	2.5 mV (12 biți)	○	●	●	●
				Curent: 4 mA – 20 mA DC	4 μA, (12 biți)				
		FX2N-4DA	4	Voltage: -10 V – 10 V DC	5 mV (cu semn, 12 biți)	○	●	●	●
				Curent: 0 mA – 20 mA DC 4 mA – 20 mA DC	20 μA (10 biți)				
		FX3U-4DA	4	Tensiune: -10 V – 10 V DC	0.32 mV (cu semn, 16 biți)	○	○	○	●
				Curent: 0 mA – 20 mA DC 4 mA – 20 mA DC	0.63 μA (15 biți)				

* Modulul de funcții speciale FX2N-8AD poate măsura deopotrivă semnale de temperaturi, curenți sau tensiuni.

Tip modul		Specificație	Nr. de canale	Domeniu	Rezoluție	FX1S	FX1N	FX2N FX2NC	FX3U				
Module mixte pentru intrări/ieșiri	Bloc pentru funcție specială	FX0N-3A	2 intrări	Tensiune: 0 V – 5 V DC 0 V to 10 V DC	40 mV (8 biți)	○	●	●	●				
				Curent: 4 mA – 20 mA DC	64 μA (8 biți)								
			1 ieșire	Tensiune: 0 V – 5 V DC 0 V – 10 V DC	40 mV (8 biți)					○	●	●	●
				Curent: 4 mA – 20 mA DC	64 μA (8 biți)								
		FX2N-5A	4 intrări	Tensiune: -100 mV – 100 mV DC -10 V – 10 V DC	50 μV (cu semn, 12 biți) 0.312 mV (cu semn, 16 biți)	○	●	●	●				
				Curent: 4 mA – 20 mA DC -20 mA – 20 mA DC	10 μA/1,25 μA (cu semn, 15 biți)								
1 ieșire	Tensiune: -10 V – 10 V DC		5 mV (cu semn, 12 biți)	○	●					●	●		
	Curent: 0 mA – 20 mA DC		20 μA (10 biți)										
Module pentru temperaturi	Adaptor special	FX3U-4AD-PT-ADP	4	Termometru cu rezistență Pt100: De la : -50 °C – 250 °C	0.1 °C	○	○	○	●				
		FX3U-4AD-TC-ADP	4	Termocuplu tip K: De la -100 °C – 1000 °C	0.4 °C	○	○	○	●				
	Termocuplu tip J: De la -100 °C – 600 °C			0.3 °C									
	Bloc pentru funcție specială	FX2N-8AD*	8	Termocuplu tip K: De la -100 °C – 1200 °C	0.1 °C	○	●	●	●				
				Termocuplu tip J: De la -100 °C – 600 °C	0.1 °C								
				Termocuplu tip T: De la -100 °C – 350 °C	0.1 °C								
		FX2N-4AD-PT	4	Termometru cu rezistență Pt100: De la -100 °C – 600 °C	0.2 – 0., °C	○	●	●	●				
		FX2N-4AD-TC	4	Termocuplu tip K: De la -100 °C – 1200 °C	0.4 °C	○	●	●	●				
	Termocuplu tip J: De la -100 °C – 600 °C			0.3 °C									
Modul pt. control temperatură	FX2N-2LC	2	De exemplu, cu un termocuplu tip K: De la -100 °C – 1300 °C	0.1°C or 1 °C (în funcție de sonda de temperatură utilizată)	○	●	●	●					
			Termometru cu rezistență Pt100: De la -200 °C – 600 °C										

* Blocul cu funcție specială FX2N-8AD poate măsura deopotrivă semnale de temperaturi, curenți sau tensiuni.

● Placa adaptoare, adaptorul special sau blocul de funcții speciale pot fi utilizate cu o unitate de bază sau o unitate de extensie din această serie.

○ Placa adaptoare, adaptorul special sau blocul de funcții speciale nu pot fi utilizate cu această serie.

Index

A

Adapter boards (analog input/output)	7-4
ADD instruction	5-21
Analog input modules	
Function	7-1
Overview	7-5
Analog output modules	
Function	7-2
Overview	7-5
ANB instruction	3-12
AND instruction	3-9
ANDP/ANDF instruction	3-14
ANI instruction	3-9
Automatic shutdown	3-22

B

Binary number	3-3
MOV instruction	5-12
Buffer memory	5-12

C

CMPP instruction	5-5
Counter	
Functions	4-7
Specifying setpoints indirectly	4-11

D

Data registers	4-9
Device	
Address	3-1
Counter overview	4-8
Data register overview	4-10
File register overview	4-11
Inputs/outputs overview	4-2
Name	3-1
Relay overview	4-3
Timer overview	4-6
DIV instruction	5-24

E

EEPROM	2-9
Emergency STOP devices	3-21

Example of programming

A rolling shutter gate	3-28
An alarm system	3-23
Clock signal generator	4-16
Delay switch	4-4
Specifying timer and counter setpoints	4-11
Switch-off delay	4-14

F

Falling edge	3-14
FMOV instruction	5-11
FROM instruction	5-14

H

Hexadecimal numbers	3-3
-------------------------------	-----

I

Instruction	
ADD	5-21
ANB	3-12
AND	3-9
ANDF	3-14
ANDP	3-14
ANI	3-9
BMOV	5-10
CMP	5-15
DIV	5-24
FMOV	5-11
FROM	5-14
Interlock of contacts	3-21
INV	3-20
LD	3-6
LDF	3-14
LDI	3-6
LDP	3-14
MC	3-19
MCR	3-19
MOV	5-7
MPP	3-17
MPS	3-17
MRD	3-17
MUL	5-23

**will be updated,
before printing!**

Temperature acquisition modules	
Function	7-2
Overview	7-6
Temperature control module	7-5, 7-6
Thermocouples	7-2
Timers	4-4
TO instruction	5-15

HEADQUARTERS		EUROPEAN REPRESENTATIVES		EUROPEAN REPRESENTATIVES		EURASIAN REPRESENTATIVES	
MITSUBISHI ELECTRIC EUROPE B.V. German Branch Gothaer Straße 8 D-40880 Ratingen Phone: +49 (0) 2102 / 486-0 Fax: +49 (0) 2102 / 486-1120 e mail: megfamail@meg.mee.com		EUROPE		GEVA Wiener Straße 89 AT-2500 Baden Phone: +43 (0) 2252 / 85 55 20 Fax: +43 (0) 2252 / 488 60 e mail: office@geva.at		AUSTRIA	
MITSUBISHI ELECTRIC EUROPE B.V. French Branch 25, Boulevard des Bouvets F-92741 Nanterre Cedex Phone: +33 1 55 68 55 68 Fax: +33 1 55 68 56 85 e mail: factoryautomation@framee.com		FRANCE		TEHNIKON Oktjabrskaya 16/5, Ap 704 BY-220030 Minsk Phone: +375 (0)17 / 210 4626 Fax: +375 (0)17 / 210 4626 e mail: tehnikon@belsonet.net		BELARUS	
MITSUBISHI ELECTRIC EUROPE B.V. Irish Branch Westgate Business Park, Ballymount IRL-Dublin 24 Phone: +353 (0) 1 / 419 88 00 Fax: +353 (0) 1 / 419 88 90 e mail: sales.info@meir.mee.com		IRELAND		Koning & Hartman B.V. Researchpark Zellik, Pontbeeklaan 43 BE-1731 Brussels Phone: +32 (0)2 / 467 17 51 Fax: +32 (0)2 / 467 17 45 e mail: info@koningenhartman.com		BELGIUM	
MITSUBISHI ELECTRIC EUROPE B.V. Italian Branch Via Paracelso 12 I-20041 Agrate Brianza (MI) Phone: +39 039 6053 1 Fax: +39 039 6053 212 e mail: factoryautomation@it.mee.com		ITALY		AKNATHON Andrej Ljapchev Lbv. Pb 21 4 BG-1756 Sofia Phone: +359 (0) 2 / 97 44 05 8 Fax: +359 (0) 2 / 97 44 06 1 e mail: —		BULGARIA	
MITSUBISHI ELECTRIC EUROPE B.V. Spanish Branch Carretera de Rubí 76-80 E-08190 Sant Cugat de Valld Phone: +34 93 / 565 316 Fax: +34 93 / 589 1579 e mail: industrial@sp.mee.com		SPAIN		AutoCont Control Systems s.r.o. Nemocnicni 12 CZ-702 00 Ostrava 2 Phone: +420 59 / 6152 111 Fax: +420 59 / 6152 562 e mail: consys@autocont.cz		CZECH REPUBLIC	
MITSUBISHI ELECTRIC EUROPE B.V. UK Branch Travelle Lane GB-Hatfield Herts. AL10 8 XB Phone: +44 (0) 1707 / 27 61 00 Fax: +44 (0) 1707 / 27 86 95 e mail: automation@meuk.mee.com		UNITED KINGDOM		louis poulsen industri & automation Sensnivej 22 DK-2670 Greve Phone: +45 44 77 11 55 Fax: +45 (0) 43 / 9 52 11 e mail: lp@lplac.dk		DENMARK	
MITSUBISHI ELECTRIC CORPORATION Office Tower "Z" 14 F 8-12,1 chome, Harumi Chuo-Ku Tokyo 104-6212 Phone: +81 3 6221 6060 Fax: +81 3 6221 6075		JAPAN		UTU Elektrotehnika AS Pärnima mt.160i EE-13175 Vilnius Phone: +372 (0) 6 61 7 30 Fax: +372 (0) 6 572 8 e mail: utu@utu.ee		ESTONIA	
MITSUBISHI ELECTRIC AUTOMATION 500 Corporate Woods Parkway Vernon Hills, IL 60061 Phone: +1 847 / 478 21 00 Fax: +1 847 / 478 22 83		USA		Beijer Electronics OY Akatia 6a FIN-01740 Vantaa Phone: +358 (0) 9 / 886 77 500 Fax: +358 (0) 9 / 886 77 555 e mail: info@beijer.fi		FINLAND	
				UTECO A.B.E.E. 5, Mavrogenous Str. GR-18542 Piraeus Phone: +302 (0) 10 / 42 10 050 Fax: +302 (0) 10 / 42 12 033 e mail: sales@uteco.gr		GREECE	
				Meltrade Ltd. Fertő Utca 14. HU-1107 Budapest Phone: +36 (0)1 / 431-9726 Fax: +36 (0)1 / 431-9727 e mail: office@meltrade.hu		HUNGARY	
				SIA POWEL Lienes iela 28 LV-1009 Riga Phone: +371 784 / 22 80 Fax: +371 784 / 22 81 e mail: utu@utu.lv		LATVIA	
				UAB UTU POWEL Savanoriu pr. 187 LT-2053 Vilnius Phone: +370 (0) 52323-101 Fax: +370 (0) 52322-980 e mail: powel@utu.lt		LITHUANIA	
				ECONOTEC AG Postfach 282 CH-8309 Nürensdorf Phone: +41 (0) 1 / 838 48 11 Fax: +41 (0) 1 / 838 48 12 e mail: info@econotec.ch		SWITZERLAND	
				GTS Darülaceze Cad. No. 43 Kat. 2 TR-80270 Okmeydani-Istanbul Phone: +90 (0) 212 / 320 1640 Fax: +90 (0) 212 / 320 1649 e mail: gts@turk.net		TURKEY	
				CSC Automation Ltd. 15, M. Raskova St., Fl. 10, Office 1010 UA-02002 Kiev Phone: +380 (0) 44 / 494 3355 Fax: +380 (0) 44 / 494 3366 e mail: csc-a@csc-a.kiev.ua		UKRAINE	
				Consulting & Engineering d.o.o. Branka Krstanovica Str. 43-V SCG-18000 Niš Phone: +381 (0) 18 / 531 226 Fax: +381 (0) 18 / 532 31 e mail: info@bekeritene.rs		SERBIA AND MONTENEGRO	
				SCG-11300 Smederevo Phone: +381 (0)20 / 615 112 Fax: +381 (0)20 / 615 116 e mail: vladstodjovic@scg.rs		SLOVENIA	
				AutoCont Control s.r.o. Radlinského 47 SK-02601 Dolná Kubín Phone: +421 435868 210 Fax: +421 435868 210 e mail: info@autocontcontrol.sk		SLOVAKIA	
				INEA d.o.o. Stegne 11 SI-1000 Ljubljana Phone: +386 (0) 1-513 8100 Fax: +386 (0) 1-513 8170 e mail: inea@inea.si		SLOVENIA	
				Beijer Electronics AB Box 426 S-20124 Malmö Phone: +46 (0) 40 / 35 86 00 Fax: +46 (0) 40 / 35 86 02 e mail: info@beijer.se		SWEDEN	
				Kazpromautomatics Ltd. 2, Scladskaya Str. KAZ-470046 Karaganda Phone: +7 3212 50 11 50 Fax: +7 3212 50 11 50 e mail: info@kpakz.com		KAZAKHSTAN	
				Avtomatika Sever Ltd. Lva Tolstogo Str. 7, Off. 311 RU-197376 St Petersburg Phone: +7 812 1183 238 Fax: +7 812 1183 239 e mail: as@avtsev.spb.ru		RUSSIA	
				Consys Promyshlennaya St. 42 RU-198099 St Petersburg Phone: +7 812 325 3653 Fax: +7 812 147 2055 e mail: consys@consys.spb.ru		RUSSIA	
				Electrotechnical Systems Siberia Shetinkina St. 33, Office 116 RU-630088 Novosibirsk Phone: +7 3832 / 119598 Fax: +7 3832 / 119598 e mail: info@eltechsystems.ru		RUSSIA	
				Elektrostyle Poslannikov Per., 9, Str.1 RU-107005 Moscow Phone: +7 095 542 4323 Fax: +7 095 956 7526 e mail: info@est.ru		RUSSIA	
				Elektrostyle Krasnij Prospekt 220-1, Office No. 312 RU-630049 Novosibirsk Phone: +7 3832 / 106618 Fax: +7 3832 / 106626 e mail: info@est.ru		RUSSIA	
				ICOS Industrial Computer Systems Ltd Ryazanskij Prospekt, 8A, Off. 100 RU-100021 Moscow Phone: +7 05 232 0207 Fax: +7 095 332 0327 e mail: mail@icos.ru		RUSSIA	
				NPP Oralelektra Sverdlova 11A RU-620027 Ekaterinburg Phone: +7 34 32 / 532745 Fax: +7 34 32 / 532745 e mail: elektra@etel.ru		RUSSIA	
				STC Drive Technique Poslannikov Per., 9, Str.1 RU-107005 Moscow Phone: +7 095 790 7210 Fax: +7 095 790 7212 e mail: info@privod.ru		RUSSIA	

**will be updated,
before printing!**