

UNIVERSITATEA "POLITEHNICA"  
TIMIȘOARA  
BIBLIOTECA CENTRALĂ

Nr. Inv. \_\_\_\_\_

Dulap \_\_\_\_\_ Lit. \_\_\_\_\_

\_\_\_\_\_

UNIVERSITATEA  
"POLITEHNICA" TIMIȘOARA  
FACULTATEA DE ELECTROTEHNICĂ  
DOMENIUL DE DOCTORAT:  
INGINERIE ELECTRICĂ

Teza de doctorat

**Asupra implementării echipamentelor cu  
logică programată pentru comanda unei stații  
de etalonat contoare de energie electrică cu  
achiziția erorii**

BIBLIOTECA CENTRALĂ  
UNIVERSITATEA "POLITEHNICA"  
TIMIȘOARA



00129098

Inginer Doru Leșan

**Conducător științific  
Prof. Dr. Ing. Alexandru Vasilevici**

- 2000 -

## Cuvânt înainte

Lucrarea "Asupra implementării echipamentelor cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii" a fost elaborată în perioada 1993-1998, având la bază brevetul de invenție pentru "Sistem numeric de calcul și afișare a erorii contoarelor de energie electrică cu achiziția erorii, comandat de un calculator", brevet numărul 112544 din 1997 și cererea de brevet cu titlul "Sistem numeric de comandă, reglare și măsură a generatoarelor polifazate de energie electrică pentru etalonarea contoarelor de energie electrică" depusă la OSIM cu numărul C1665 din 1998 la care subsemnatul este coautor.

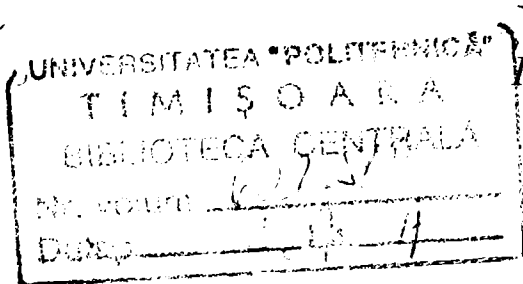
Sprijinul științific și moral acordat de către conducătorul științific, profesor doctor inginer Alexandru Vasilevici, competența și continua îndrumare acordată pe parcurs, au contribuit decisiv la reușita lucrării.

Sprijinul competent și amabil al colectivului catedrei de electroenergetică din cadrul Facultății de Electrotehnică al Universității "Politehnica" din Timișoara și în special al domnilor ș.l. dr. ing. Silvian Năstăsie și ș.l. dr. ing. Doru Vătău a contribuit esențial la cristalizarea lucrării.

Pe parcursul elaborării tezei, dificultățile interne au putut fi depășite numai cu sprijinul competent al colegilor din cadrul A.E.M. S.A. și în special al doamnei inginer Bechira Marcela.

Realizarea practică a echipamentului este rodul colaborării permanente cu beneficiarii acestor instalații, în principal CONEL ELECTRICA, a căror observații și sugestii competente au dus la îmbunătățirea softului și a întregului echipament.

Tuturor celor amintiți țin să le transmit cele mai respectuoase mulțumiri și să-mi exprim întreaga recunoștință pentru sprijinul acordat.



Ing. Doru Leșan

## CUPRINS

<b>CAPITOLUL 1.....</b>	<b>1</b>
<b>INTRODUCERE.....</b>	<b>1</b>
1.1 STAȚII DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ.....	2
1.2 OBIECTIVELE TEZEI.....	5
1.3 STRUCTURA TEZEI.....	5
<b>CAPITOLUL 2.....</b>	<b>8</b>
<b>EVOLUȚIA CONCEPȚIEI DIN PUNCT DE VEDERE TEHNIC A INSTALAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ.....</b>	<b>8</b>
2.1 ASUPRA NECESITĂȚII VERIFICĂRII CONTOARELOR DE ENERGIE ELECTRICĂ.....	9
2.2 METODE DE VERIFICARE A CONTOARELOR DE ENERGIE ELECTRICĂ.....	15
2.3 EVOLUȚIA INSTALAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ PE PLAN MONDIAL.....	17
2.4 SCURT ISTORIC AL EVOLUȚIEI DEZVOLTĂRII INSTALAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ LA L.L.C. S.C. A.E.M. S.A.....	18
<b>CAPITOLUL 3.....</b>	<b>21</b>
<b>STAȚII DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ - MOD DE REALIZARE.....</b>	<b>21</b>
3.1 CERINȚE FUNCȚIONALE ALE STAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ.....	22
3.2 SCHEMA BLOC ȘI ELEMENTELE COMPONENTE ALE STAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ.....	23
3.2.1 Sistemul de generare al tensiunilor.....	23
3.2.2 Sistemul de generare al curentului.....	25
3.2.3 Descrierea constructivă și funcțională a stației SE 04.....	26
<b>CAPITOLUL 4.....</b>	<b>40</b>
<b>ELEMENTE ALE AUTOMATIZĂRII NECESARE ÎN FUNCȚIONAREA STAȚIEI DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ.....</b>	<b>40</b>
4.1 MODERNIZAREA UNEI STAȚII DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ EXISTENTĂ, PRIN IMPLEMENTAREA ECHIPAMENTELOR CU LOGICĂ PROGRAMATĂ.....	41
4.2 DOTAREA UNEI STAȚII DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ CU ECHIPAMENTE DE CALCUL ȘI AFIȘARE A ERORII CONTOARELOR.....	44
4.3 DESCRIEREA FUNCȚIONĂRII ELEMENTELOR DE AUTOMATIZARE ALE STAȚIEI DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ.....	47
4.3.1 Funcționarea elementelor de automatizare ale stației de etalonat contoare de energie electrică cu autotransformatoare, cu reglaj prin intermediul servomotoarelor al mărimilor prescrise.....	47
4.3.2 Implementarea logicii programate pe o instalație cu autotransformatoare cu reglajul mărimilor prescrise prin intermediul servomotoarelor.....	48
4.3.3 Funcționarea elementelor de automatizare ale stației de etalonat contoare de energie electrică cu reglaj electronic al mărimilor prescrise.....	58
4.3.4 Implementarea logicii programate pe o instalație cu generator electronic.....	59
4.3.5 Implementarea programului pe calculatorul de proces în vederea etalonării contoarelor de energie electrică.....	64
<b>CAPITOLUL 5.....</b>	<b>68</b>

<b>REALIZAREA INSTALAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ CU ACHIZIȚIA ERORII.....</b>	<b>68</b>
5.1 CLASIFICAREA INSTALAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ.....	69
5.1.1 După criteriul constructiv:.....	69
5.1.2 După metoda de lucru utilizată pentru etalonarea contoarelor de energie electrică:.....	69
5.2 SISTEM DE CALCUL ȘI AFIȘAREA ERORII IMPLEMENTAT PE INSTALAȚIILE DE ETALONAT ȘI/SAU VERIFICAT CONTOARE DE ENERGIE ELECTRICĂ.....	69
5.2.1 Descrierea sistemului de calcul și afișarea erorii "ERRORREX M" din dotarea instalațiilor de etalonat contoare de energie electrică.....	70
5.2.2 Principiul de funcționare:.....	71
5.2.3. Descriere constructivă și funcțională.....	74
5.2.4. Caracteristici tehnice principale.....	104
5.3 SCHEMA BLOC ȘI ELEMENTELE COMPONENTE ALE SISTEMULUI DE CALCUL AFERENT STAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ.....	105
<b>CAPITOLUL 6.....</b>	<b>110</b>
<b>PREZENT ȘI VIITOR ÎN REALIZAREA INSTALAȚIILOR DE TESTARE A CONTOARELOR DE ENERGIE ELECTRICĂ.....</b>	<b>110</b>
6.1 SISTEME ELECTRONICE DE ALIMENTARE CU PUTERE.....	111
6.2 TEHNOLOGIA AMPLIFICATORULUI ANALOGIC.....	112
6.3 TEHNOLOGIA AMPLIFICATORULUI ÎN COMUTAȚIE.....	113
6.4 TENDINȚE NOI ÎN DOMENIUL UNITĂȚILOR ELECTRONICE DE ALIMENTARE CU PUTERE.....	115
6.5 CONTOARE ETALON DE REFERINȚĂ DE PRECIZIE.....	116
6.6 PROGRAMUL DE CONTROL.....	117
<b>CAPITOLUL 7.....</b>	<b>118</b>
<b>MODALITĂȚI DE VERIFICARE ȘI REZULTATELE TESTĂRIILOR INSTALAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ CU ACHIZIȚIA ERORII..</b>	<b>118</b>
7.1 VERIFICAREA STAȚIEI SE 04.....	119
7.2 REZULTATELE VERIFICĂRII STAȚIEI SE 04.....	119
7.3 VERIFICAREA SISTEMULUI DE CALCUL ERRORREX M.....	119
7.3.1 Condiții electrice.....	121
7.3.2 Condiții climatice.....	125
7.3.3 Condiții mecanice.....	125
7.3.4 Solicitări de durabilitate.....	126
7.4 REZULTATELE VERIFICĂRII SISTEMULUI DE CALCUL ERRORREX M.....	126
<b>CAPITOLUL 8.....</b>	<b>130</b>
<b>CONCLUZII.....</b>	<b>130</b>
<b>REFERINȚE BIBLIOGRAFICE.....</b>	<b>136</b>
BREVETE.....	141
<b>ANEXE.....</b>	<b>142</b>
ANEXA 1 ORDINOGRAME ERRORREX M.....	A1
ANEXA 2 PROGRAM POST CALCUL ERRORREX M.....	A7
ANEXA 3 PROGRAM INTREFAȚĂ ACHIZIȚIE ERRORREX M (1/3).....	A18
PROGRAM INTREFAȚĂ ACHIZIȚIE ERRORREX M (2/3).....	A21
PROGRAM INTREFAȚĂ ACHIZIȚIE ERRORREX M (3/3).....	A23
ANEXA 4 PROGRAM BLOC DE MĂSURĂ ERRORREX M.....	A29
ANEXA 5 PROGRAM BLOC DE COMANDĂ SE 04.....	A40
ANEXA 6 PROGRAM CALCULATOR.....	A54
ANEXA 7 SCHEMA ELECTRICĂ A STAȚIEI DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ SE 04.....	A117

## **CAPITOLUL 1**

### **INTRODUCERE**

## ***1.1 Stații de etalonat contoare de energie electrică***

Pentru etalonarea și/sau verificarea contorului de energie electrică conform normelor în vigoare, este necesar ca acesta să fie alimentat cu un sistem de tensiuni trifazat de amplitudine prescrisă, aplicat circuitului de tensiune al contorului și un sistem de curenți trifazat de amplitudine prescrisă, aplicat circuitului de curent al contorului. Defazajul dintre cele două sisteme trifazate trebuie să poată fi prescris, pentru a putea controla factorul de putere.

Stațiile de etalonat contoare asigură generarea celor două sisteme (de tensiune, respectiv de curent) trifazate, programabile în amplitudine și fază unul față de altul. Cele două sisteme sunt izolate galvanic între ele pentru a permite conectarea mai multor contoare de energie electrică concomitent. Circuitul de tensiune alimentează toate bobinele de tensiune ale contoarelor pe fiecare fază legate în paralel, iar circuitul de curent alimentează toate bobinele de curent ale contoarelor pe fiecare fază legate în serie.

Instalațiile de etalonat contoare de energie electrică se clasifică după cum urmează:

- după criteriul constructiv [3], [4]:
  - instalații cu autotransformatoare cu reglaj manual al mărimilor prescrise (manuale);
  - instalații cu autotransformatoare cu reglaj prin intermediul servomotoarelor al mărimilor prescrise (semiautomate);
  - instalații cu generator electronic (automate).
- după metoda de lucru utilizată pentru etalonarea contoarelor de energie electrică [4], [52], [63]:
  - instalații fără contor electronic etalon (specifice metodei watmetru - cronometru);
  - instalații cu contor electronic etalon (specifice metodei contorului etalon).

Implementarea logicii programate se poate face pe instalații cu autotransformatoare cu reglaj al mărimilor prescrise prin intermediul servomotoarelor și pe instalații cu generator electronic.

În momentul actual, pe plan mondial, se fabrică instalații de etalonare a contoarelor de energie electrică manuale, semiautomate și automate.

Instalațiile *automate* sunt dotate aproape în exclusivitate cu generator electronic. Datorită reducerii costurilor componentelor electronice și a îmbunătățirii permanente a performanțelor acestora, s-au putut obține generatoare de puteri considerabile cu randament ridicat și un grad înalt de automatizare la un preț din ce în ce mai scăzut. Aceste generatoare funcționează în comutație și sunt controlate cu un sistem echipat cu microcontroler. Prezența sistemului face posibilă comunicarea cu un calculator de proces, de la care se prescriu mărimile de lucru.

Firmele consacrate în fabricarea de astfel de echipamente, (ZERA GmbH, SCHLUMBERGER), produc instalații de acest gen, fiecare după un concept propriu. Instalațiile cele mai complexe sunt dotate cu mai multe monitoare pentru afișarea parametrilor de funcționare a instalației (tensiuni, curenți, decalaj și puteri) și a probelor în curs de desfășurare. Aceste instalații au un preț relativ ridicat și sunt concepute pentru utilizarea lor de un personal cu înaltă specializare; sunt instalații universale cu ajutorul cărora se pot executa atât lucrări de verificare și/sau etalonare a contoarelor de energie electrică, cât și lucrări de calibrare a altor aparate (traductoare, relee de precizie). Aceste posibilități multiple le fac să fie greu accesibile (din punct de vedere financiar și al personalului necesar pentru deservire) utilizatorilor din țară.

Instalațiile *semiautomate* sunt instalații care au în dotarea lor elemente de comandă automatizate (autotransformatoare comandate de servomotoare, decalor rotativ comandat de servomotor, etc) și care sunt dotate de obicei cu facilități auxiliare de reglare a mărimilor prescrise. Una dintre aceste instalații este stația trifazată METRA P038 produsă de Metra Blansco din Cehia.

Instalațiile *manuale* au o pondere foarte mică, ele fiind de construcție mai veche, realizarea acestora implicând cheltuieli de

manoperă relativ mari. Pe plan mondial s-a renunțat aproape complet la producerea de astfel de instalații.

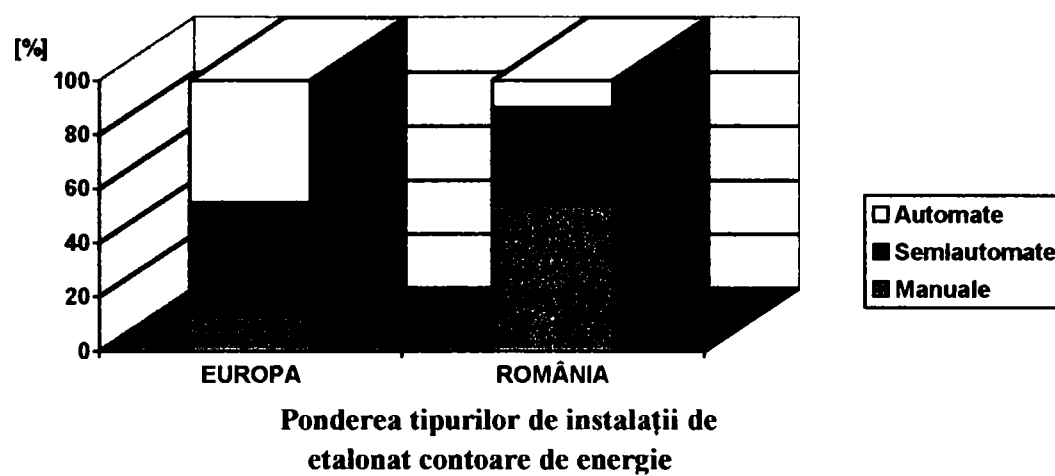


FIGURA 1.1

Așa cum rezultă și din graficul prezentat în figura 1.1 [42], pe plan european ponderea cea mai mare o au instalațiile automate, urmate de cele semiautomate. Fabricanții de contoare de energie electrică au în dotare instalații complexe cu generatoare electronice. Instalațiile manuale sunt cele mai puțin răspândite și sunt de obicei cele rămase încă în exploatare fiind la limita de casare, sau cele artisanale, realizate prin autodotări.

La noi în țară, instalațiile manuale de construcție mai veche aflate în exploatare constituie dotarea majoritară, actuală a verificatorilor de contoare de energie electrică. Pe lângă acestea, sunt în funcțiune un număr remarcabil de instalații semiautomate și câteva instalații automate.

Având în vedere structura tipurilor de instalații existente, s-a considerat de un interes deosebit posibilitatea automatizării instalațiilor semiautomate și realizarea unui model de instalație automată de etalonat contoare de energie electrică la un preț scăzut, cu performanțe suficiente pentru etalonarea și/sau verificarea parcului existent de contoare. La realizarea tipului nou de instalație automatizată s-a avut în vedere și dotarea în viitorul apropiat cu noile tipuri de contoare multitarif, clasice și electronice. Performanțele acestor instalații satisfac, conform normativelor în vigoare, necesitățile impuse pentru verificarea contoarelor de energie electrică.



Sistemul de calcul al erorii contoarelor de energie electrică cu care se dotează aceste instalații este realizat cu un înalt grad de automatizare [41], permițând calculul erorii cu ajutorul unei metode specifice care face ca eroarea de calcul să fie dată practic de eroarea convertorului etalon al instalației.

## ***1.2 Obiectivele tezei***

Lucrarea își propune drept obiectiv principal realizarea unei stații de etalonat contoare de energie electrică dotată cu un sistem numeric de calcul și afișare a erorii contoarelor, cu achiziția acestora la un calculator compatibil IBM PC.

În acest scop lucrarea abordează următoarele aspecte:

- a) evoluția concepției din punct de vedere tehnic a instalațiilor de etalonat contoare de energie electrică cu achiziția erorii;
- b) modul practic de realizare a stațiilor de etalonat contoare de energie electrică și a echipamentelor de automatizare;
- c) modalități de verificare și rezultatele testărilor în laborator și la beneficiari a stațiilor de etalonat contoare de energie electrică cu achiziția erorii.

La baza stabilirii obiectivelor au stat considerente legate de necesitățile practice ale laboratoarelor de etalonare și/sau verificare a contoarelor de energie electrică ale fabricanților de contoare cât și a beneficiarilor acestora, în principal filialele CONEL Electrica.

## ***1.3 Structura tezei***

Lucrarea este structurată pe opt capitole, la care se adaugă cuprinsul, referințele bibliografice și anexele.

Capitolul 2 face o trecere în revistă a evoluției concepției din punct de vedere tehnic a instalațiilor de etalonat contoare de energie

electrică. Se fac referiri la situația pe plan mondial și național, cu particularizare pentru L.L.C. S.C. A.E.M. S.A. Timișoara.

Următorul capitol este dedicat prezentării cerințelor funcționale ale stațiilor de etalonat contoare de energie electrică. De asemenea se prezintă modul concret de realizare al unei stații de etalonat contoare de energie electrică, prezentându-se din punct de vedere constructiv și funcțional stația SE 04.

Capitolul 4 face o prezentare a elementelor de automatizare necesare în funcționarea stației de etalonat contoare de energie electrică. Pornind de la analiza necesității de modernizare a unei stații de etalonat contoare de energie electrică existente, s-au stabilit direcțiile de dezvoltare a echipamentelor necesare dotării unei astfel de instalații. S-a prezentat soluția propusă de autor în vederea implementării echipamentelor cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii la un calculator compatibil IBM PC. De asemenea s-a prezentat implementarea programului pe calculatorul de proces în vederea etalonării contoarelor de energie electrică.

Următorul capitol descrie în mod concret soluția propusă de autor, prezentând pe larg realizarea constructivă și modul de funcționare al sistemului de calcul și afișarea erorii ERRORREX M, cât și caracteristicile tehnice principale. Sunt prezentate de asemenea ordinogramele de funcționare, ANEXA 1, pentru toate blocurile componente ale sistemului. Pe baza acestora s-a realizat pachetul de programe soft ce deservește sistemul, cuprinse în ANEXA 2 până la ANEXA 6. În ANEXA 7 este prezentată schema electrică desfășurată a stației SE 04.

Capitolul 6 face o trecere în revistă a situației actuale și a tendințelor de dezvoltare a instalațiilor de etalonat contoare de energie electrică.

Capitolul următor prezintă modalitățile de verificare și rezultatele concrete ale testării instalațiilor de etalonat contoare de energie electrică și anume a stației SE 04 și a sistemului ERRORREX M care reprezintă soluția practică concepută de autor. În cadrul capitolului sunt atașate buletinele de verificare corespunzătoare.

Ultimul capitol prezintă concluziile generale la lucrare, cu indicarea contribuțiilor originale ale autorului.

## **CAPITOLUL 2**

### **EVOLUȚIA CONCEPȚIEI DIN PUNCT DE VEDERE TEHNIC A INSTALAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ**

## 2.1 Asupra necesității verificării contoarelor de energie electrică

Măsurarea energiei electrice este cea mai frecventă măsurătoare care se realizează în practică, intervenind la toți consumatorii electrici racordați la rețelele de distribuție. Energia activă este o mărime integrală, fiind integrala puterii active în intervalul de timp considerat:

$$W = \int_{t_1}^{t_2} P dt \quad (2.1)$$

Din formula 2.1 rezultă că nu se pot utiliza instrumente obișnuite care efectuează media pe o perioadă a mărimii de măsurat.

Pentru măsurarea energiei sunt necesare instrumente integratoare și conform teoriei generale a instrumentelor cu citire directă aceste instrumente nu au culuri antagoniste. Ecuația diferențială a mișcării acestor instrumente este:

$$J \frac{d^2 \alpha}{dt^2} + A \frac{d\alpha}{dt} = M \quad (2.2)$$

Integrând în intervalul de timp  $t_2-t_1$  se obține:

$$J \left| \frac{d\alpha}{dt} \right|_{t_1}^{t_2} + A \alpha \Big|_{t_1}^{t_2} = \int_{t_1}^{t_2} M dt \quad (2.3)$$

În regim de viteză constantă  $d\alpha/dt = \text{constant}$ , după terminarea regimului tranzitoriu, integrala devine:

$$A(\alpha_2 - \alpha_1) = \int_{t_1}^{t_2} M dt \quad (2.4)$$

Neglijarea regimului tranzitoriu conduce la erori. Factorul  $\alpha_2 - \alpha_1$  este unghiul total de rotație al echipajului mobil în intervalul de timp  $t_2-t_1$ , care se poate scrie sub forma:

$$\alpha_2 - \alpha_1 = 2\pi N \quad (2.5)$$

unde  $N$  este numărul de rotații efectuate de echipajul mobil în intervalul de timp  $t_2-t_1$  (poate fi și număr fracționar). Cu acestea, se obține:

$$A2\pi N = \int_{t_1}^{t_2} M dt \quad (2.6)$$

$$N = \frac{1}{2\pi A} \int_{t_1}^{t_2} M dt \quad (2.7)$$

Deci există posibilitatea măsurării unor integrale ale cuplurilor dacă coeficientul de amortizare  $A$  este o mărime constantă. Deoarece

$$W = \int_{t_1}^{t_2} P dt \quad (2.8)$$

înseamnă că în expresia cuplului activ trebuie să intre puterea, adică:

$$M = KP \quad (2.9)$$

Unde  $K$  este o mărime constantă. În acest caz:

$$M = \frac{K}{2\pi A} \int_{t_1}^{t_2} P dt = \frac{K}{2\pi A} W \quad (2.10)$$

De aici rezultă:

$$W = CM \quad (2.11)$$

unde  $C = \frac{2\pi A}{K}$  se numește raport de transmisie.

Contorul este un instrument ce măsoară mărimi integrale. Contorul de energie activă conține un dispozitiv de tip wattmetric pentru producerea lui  $M$ , este lipsit de resoarte spirale și are un cuplu de amortizare constant și reglabil  $A$ . Contorul de energie reactivă are cuplul activ realizat printr-un dispozitiv de tip varmetric, indicația lui fiind:

$$W_r = \int_{t_1}^{t_2} Q dt \quad (2.12)$$

**Contorul de energie** este un aparat de măsurat utilizat în mod continuu, de un număr mare de consumatori de energie electrică. Din acest motiv, fabricarea contoarelor de energie are caracterul unei producții de masă. Aceasta reclamă metode și instalații pentru etalonarea și verificarea contoarelor cu următoarele caracteristici:

- Precizie ridicată;
- Productivitate foarte mare;
- Grad înalt de automatizare.

Datorită operațiilor complexe pe care le implică etalonarea și verificarea, atât metodele cât și instalațiile, s-au apropiat relativ lent de aceste necesități ale producției.

Principala caracteristică a determinării erorii de măsurare a unui contor este faptul că indicația acestuia nu este o valoare determinată instantaneu, ci o înregistrare în timp, ceea ce conduce la un timp de testare mult mai lung decât în cazul celorlalte aparate de măsurat. Din

acest motiv, productivitatea ridicată necesară se realizează prin testarea simultană a unui număr cât mai mare de contoare.

**Funcționarea contorului de inducție în regim deformant** introduce o eroare suplimentară în precizia de măsurare a energiei electrice active, eroare a cărei estimare se impune a fi făcută pe baza unui model teoretic [17], [21], [22], [26]. Eroarea de măsurare a energiei în regim deformant depinde de mărimile externe impuse de rețea, de caracteristicile consumatorului, de anumiți parametri ai contorului, precum și de interacțiunea rețea-sistem de măsurare. Stabilirea modelului de calcul al contorului în regim deformant este dificilă, având în vedere multitudinea parametrilor care intervin, varietatea configurațiilor geometrice, nelinearitatea sistemului de măsură. Pentru stabilirea expresiilor momentului cuplurilor activ și de amortizare care acționează asupra discului contorului în regim deformant se consideră inducțiile magnetice în întrefierul electromagneților de tensiune și de curent electric de variație nesinusoidală în timp, dezvoltabile în serie Fourier.

Determinarea densității curentului electric în disc și a inducției magnetice se face prin integrarea ecuațiilor lui Maxwell, scrise sub forma generală. Într-un referențial imobil în raport cu polii magnetici, legea inducției electromagnetice pentru corpurile în mișcare, scrisă relativ la armonica de rang  $n$  rezultă:

$$\operatorname{rot} \bar{E}_n = -\frac{\partial \bar{B}_n}{\partial t} + \operatorname{rot}(\bar{v} \times \bar{B}_n) \quad (2.13)$$

în care  $\bar{E}_n$  este armonica de rang  $n$  a intensității câmpului electric indus în disc, iar  $\bar{v}$  este viteza benzii care modelează coroana circulară de sub poli.

Câmpul electric indus determină în disc un curent electric de inducție cu o densitate locală  $\bar{J}_n$ , ce rezultă din legea conducerii electrice:

$$\bar{E}_n = \rho \bar{J}_n = \frac{1}{\sigma} \bar{J}_n \quad (2.14)$$

În relația (2.14) s-a notat cu  $\rho$  rezistivitatea și cu  $\sigma$  conductivitatea materialului discului. Se consideră că materialul discului este omogen, izotrop și la temperatură uniformă.

Din legea circuitului magnetic rezultă dependența câmpului magnetic de reacție de curenții electrici induși în disc:

$$\operatorname{rot} \vec{B}_n = \mu_0 \frac{\delta}{\delta_1} \underline{J}_n \quad (2.15)$$

Din legea conservării sarcinii electrice adevărate rezultă:

$$\operatorname{div} \vec{J}_n = 0 \quad (2.16)$$

iar din legea conducției electrice se obține:

$$\operatorname{div} \vec{E}_n = 0 \quad (2.17)$$

Relațiile (2.14) - (2.17) ca și relația:

$$\operatorname{div} \vec{B}_n = 0 \quad (2.18)$$

scrise pentru domeniile I și II, constituie sistemul de ecuații diferențiale prin integrarea cărora se determină densitatea curentului electric  $\vec{J}_n$  în disc și inducția magnetică  $\vec{B}_n$ .

Din relația (2.17) rezultă că poate fi definit un potențial electric vector pe baza relației:

$$\vec{E}_n = \operatorname{rot} \vec{A}_n \quad (2.19)$$

În acest fel ecuațiile de tip Laplace și Poisson care permit stabilirea potențialului electric vector  $\vec{A}_n$  în zonele I și II rezultă:

$$\frac{\partial^2 \underline{A}_{nI}}{\partial x^2} + \frac{\partial^2 \underline{A}_{nI}}{\partial y^2} - \frac{\alpha_n^2 v}{\omega_n} \frac{\partial \underline{A}_{nI}}{\partial x} - j \alpha_n^2 \underline{A}_{nI} = 0, \quad (2.20)$$

$$\frac{\partial^2 \underline{A}_{nII}}{\partial x^2} + \frac{\partial^2 \underline{A}_{nII}}{\partial y^2} - \frac{\alpha_n^2 v}{\omega_n} \frac{\partial \underline{A}_{nII}}{\partial x} - j \alpha_n^2 \underline{A}_{nII} = j \omega_n \underline{B}_n^0 + v \frac{\partial \underline{B}_n^0}{\partial x}$$



În relațiile (2.20) s-au utilizat notațiile:

$$\alpha_n^2 = \omega_n \mu_0 \sigma \frac{\delta}{\delta_1} \quad (2.21)$$

$$\omega_n = 2\pi f n$$

Inducțiile magnetice de excitație din întrefierul electromagneților de tensiune și de curent electric care intervin în relațiile (2.20) se dezvoltă în serie Fourier spațială în raport cu coordonata  $x$ :

$$\underline{B}_n^0(x) = \underline{B}_n^0 e^{\lambda x} \left[ \frac{C_0}{2} + \sum_{k=1}^{\infty} S_k \sin k \frac{2\pi}{\tau} x + \sum_{k=1}^{\infty} C_k \cos k \frac{2\pi}{\tau} x \right] \quad (2.22)$$

unde s-a notat:

$$\lambda = \frac{\alpha_n^2 v}{2\omega_n} = \mu_0 \sigma \frac{\delta v}{\delta_1 2} \quad (2.23)$$

Componentele densităților de curent electric orientate în direcția axelor OX și OY în zonele I și II sunt determinate din expresiile potențialului electric vector pe baza relațiilor:

$$\underline{J}_{nx} = \sigma \frac{\partial A_n}{\partial y} \quad (2.24)$$

$$\underline{J}_{ny} = -\sigma \frac{\partial A_n}{\partial x}$$

Constantele de integrare din soluțiile ecuațiilor se determină cu ajutorul condițiilor de frontieră și de racordare:

- pe suprafețele  $y=l+l_1$  și  $y=-l$  care măginesc domeniile conductoare I și II densitatea curentului trebuie să aibă componenta normală nulă:

$$\underline{J}_n \cdot \underline{n} \Big|_{y=l+l_1} = 0$$

(2.25)

$$J_{ny} \Big|_{II} = 0$$

- pe suprafețele de separare a celor două zone I și II componentele tangențiale și normale ale densității curentului electric trebuie să se conserve:

$$\underline{J}_{nx} \Big|_{y=l} = \underline{J}_{nx} \Big|_{y=l} \quad (2.26)$$

$$\underline{J}_{ny} \Big|_{y=l} = \underline{J}_{ny} \Big|_{y=l}$$

Ipotezele simplificatoare adoptate pentru determinarea relațiilor de calcul al momentelor cuplurilor activ și de amortizare permit obținerea unor rezultate suficient de apropiate de valorile obținute prin determinări experimentale.

**Clasa de precizie** a contoarelor de energie se stabilește în funcție de erorile relative de măsurare, determinate pentru diverse regimuri de funcționare. Eroarea relativă de măsurare este definită conform relației:

$$\varepsilon_r = \frac{W_{masurat} - W_{etalon}}{W_{etalon}} * 100 \quad [\%] \quad (2.27)$$

unde  $W_{masurat}$  este energia înregistrată de contorul testat, iar  $W_{etalon}$  este energia măsurată de un contor considerat etalon.

Probele metrologice la care este supus un contor includ variația curentului de sarcină în intervalul 5...400% din curentul de bază și variația defazajului în intervalul 0,8 capacitiv...0,5 inductiv. Contoarele trifazate sunt testate atât în regim de sarcină echilibrată, cât și cu încărcare pe o singură fază. Pentru fiecare probă, contorul trebuie să aibă o eroare de măsură mai mică decât limitele stabilite în documentele legale.

Astfel, procedura de determinare a clasei de precizie a contoarelor de energie este laborioasă, cuprinzând mai multe probe, fiecare necesitând un timp de măsurare ce poate fi de ordinul minutelor.

Tot în cadrul verificării performanțelor metrologice sunt incluse teste care nu necesită determinarea erorii de măsurare. Pragul de sensibilitate este dat de curentul minim la care începe să se rotească discul contorului și nu trebuie să depășească 0,5% din curentul de bază, în condițiile nominale de tensiune și defazaj. Verificarea

mersului în gol constă în absența rotației dicului, în condițiile alimentării contorului doar în tensiune, în intervalul 80...110% din tensiunea nominală. Verificarea mecanismului totalizator se realizează prin contorizarea unei cantități de energie de valoare mare într-un timp de 15...20 minute.

În afara probelor metrologice prezentate, contoarele mai sunt verificate din punct de vedere al performanțelor mecanice și al stabilității în funcționare.

**Etalonarea** este etapa tehnologică în care, asupra contorului asamblat și în regim de funcționare, se efectuează reglaje pentru asigurarea preciziei necesare în fiecare probă. Reglajele se efectuează manual pe baza erorii de măsurare medii, într-un interval de timp. Contoarele sunt prevăzute cu trei reglaje pe fiecare fază, cele trifazate fiind etalonate atât în regim trifazat echilibrat, cât și pe fiecare fază în parte. Din acest motiv, operația de etalonare este cea mai lungă în procesul de producție a contoarelor și necesită personalul cel mai calificat. Pentru o reglare cât mai eficientă, în timp real, operatorul trebuie să determine cât mai rapid și cât mai exact eroarea medie într-un interval de timp a contorului ce se etalonează.

**Verificarea metrologică** este o etapă legală ce se efectuează asupra contorului fabricat și sigilat. Ea reprezintă totalitatea operațiilor metrologice care se execută cu scopul constatării dacă performanțele și caracteristicile aparatului se încadrează sau nu în limitele impuse de normativele în vigoare. Operația necesită determinarea erorii de măsurare a contorului în fiecare probă în parte și acordarea calificativului admis sau respins. Eficiența este dată de viteza de determinare a erorilor de măsurare, verificarea făcându-se printr-o măsurare unică, la fiecare probă.

Complexitatea acestor operații conduce la necesitatea realizării unor instalații de testare industrială a contoarelor deosebit de eficiente.

## ***2.2 Metode de verificare a contoarelor de energie electrică***

Necesitatea asigurării furnizorilor de energie electrică și a abonaților cu contoare de bună calitate, cu erori cât mai mici, a determinat în timp găsirea și dezvoltarea unor soluții de verificare a

încadrării contoarelor în limitele de precizie prevăzute de actele normative în vigoare.

Astfel funcție de principiile și metodele aplicate, de la verificările manuale până la cele folosind linii automatizate, metodele de verificare se pot clasifica în câteva grupe distincte:

- Metoda wattmetru-cronometru: măsurarea energiei prin măsurarea concomitentă a puterii și a timpului;
- Metoda comparării directe: compararea vitezei discului contorului cu a unui contor etalon;
- Metoda comparării stroboscopice: compararea vitezei discului contorului cu a unui contor etalon folosind efectul stroboscopic;
- Metoda contorului etalon: compararea energiei electrice înregistrate de contor cu energie înregistrată de un contor etalon.

Metoda *wattmetru-cronometru* se bazează pe măsurarea unui anumit număr de rotații ale discului sistemului mobil al contorului de verificat și pe măsurarea timpului utilizat pentru aceasta, presupunându-se că puterea reglată și urmărită rămâne constantă pe timpul măsurării. La verificarea contorului sub sarcină diferită de cea nominală, numărul de rotații calculate se micșorează sau se mărește de atâtea ori, de câte ori această sarcină este mai mică sau mai mare decât cea nominală; datorită acestui fapt timpul rămâne același pentru toate sarcinile, ceea ce simplifică calculul erorilor. Tendința de a perfecționa această metodă și de a-i mării precizia a condus la apariția dispozitivelor de numărare automată a rotațiilor efectuate de discul contorului verificat.

Metoda *comparării directe* constă în compararea vitezei discului contorului cu cea a unui contor considerat etalon. Metoda, ca și toate metodele care folosesc contor etalon, prezintă avantajul că nu necesită stabilizarea sarcinii, fiind justificată pentru reglarea concomitentă a opt până la zece contoare.

Metoda *comparării stroboscopice* constă în compararea valorii instantanee a vitezei unghiulare a discului contorului cu o numită viteză unghiulară. Această comparație se efectuează pe cale optică.

Metoda *contorului etalon* constă în compararea indicației mecanismului de înregistrare de la contorul care se verifică cu indicațiile contorului etalon. Această metodă și-a găsit o largă aplicare, utilizându-se pentru verificarea simultană a unui număr de până la 200

bucăți contoare. Cu acest procedeu se verifică nu numai justetea reglării vitezei de rotație a discului contorului de verificat, ci se controlează în același timp și justetea de funcționare a mecanismului său de integrare, care în cazul tuturor celorlate procedee de verificare necesită o operație suplimentară.

### ***2.3 Evoluția instalațiilor de etalonat contoare de energie electrică pe plan mondial***

Primele instalații de etalonat și/sau verificat contoare de energie electrică erau construite în scopul verificării contoarelor prin metoda comparației cu un contor a cărui eroare era cunoscută. Instalațiile de acest gen nu s-au răspândit datorită ineficienței lor și a preciziei scăzute.

Următoarele tipuri de instalații de etalonat și/sau verificat contoare de energie electrică au fost cele care au utilizat metoda wattmetru - cronometru. Pentru utilizarea metodei respective dotarea acestora era minimală: wattmetre și cronometre.

Metoda necesită un număr mare de operatori, timp îndelungat de lucru, precizia de verificare fiind influențată negativ de subiectivismul uman.

Datorită acestor neajunsuri s-a impus în mod firesc perfecționarea instalațiilor.

Pe baza metodelor de etalonare prezentate s-au realizat instalații pentru etalonarea și verificarea contoarelor electrice care erau practic doar surse de curent alternativ cu posibilități de reglare a curentului, tensiunii și decalajului între curent și tensiune. Aceste instalații prezintă avantajul că sunt robuste și se pot utiliza pentru toate metodele de etalonare verificare a contoarelor.

În prezent, firmele consacrate în fabricarea de astfel de echipamente, (ZERA GmbH, SCHLUMBERGER) [3], [25], produc instalații de acest gen, fiecare după un concept propriu. Instalațiile cele mai complexe sunt dotate cu mai multe monitoare pentru afișarea parametrilor de funcționare a instalației (tensiuni, curenți, decalaj și puteri) și a probelor în curs de desfășurare. Aceste instalații au un preț relativ ridicat și sunt concepute pentru utilizarea lor de un personal cu înaltă specializare; sunt instalații universale cu ajutorul cărora se pot executa atât lucrări de verificare și/sau etalonare a contoarelor de energie electrică, cât și lucrări de calibrare a altor aparate (traductoare,

relee de precizie). Aceste posibilități multiple le fac să fie greu accesibile (din punct de vedere financiar și al personalului necesar pentru deservire) utilizatorilor din țară.

#### **2.4 Scurt istoric al evoluției dezvoltării instalațiilor de etalonat contoare de energie electrică la L.L.C. S.C. A.E.M. S.A.**

Datorită creșterii producției de contoare de energie electrică și măririi exigenței la recepția la beneficiari, A.E.M. S.A. a fost nevoită să doteze liniile de fabricație cu instalații de etalonat contoare de energie electrică cu productivitate mărită și performanțe ridicate. În figura 2.1 este prezentată evoluția sistemelor de etalonat contoare de energie electrică la L.L.C. S.C. A.E.M. S.A. [36].

Primele astfel de instalații care au dotat liniile de fabricație în anul 1981 au fost SETAC 16.

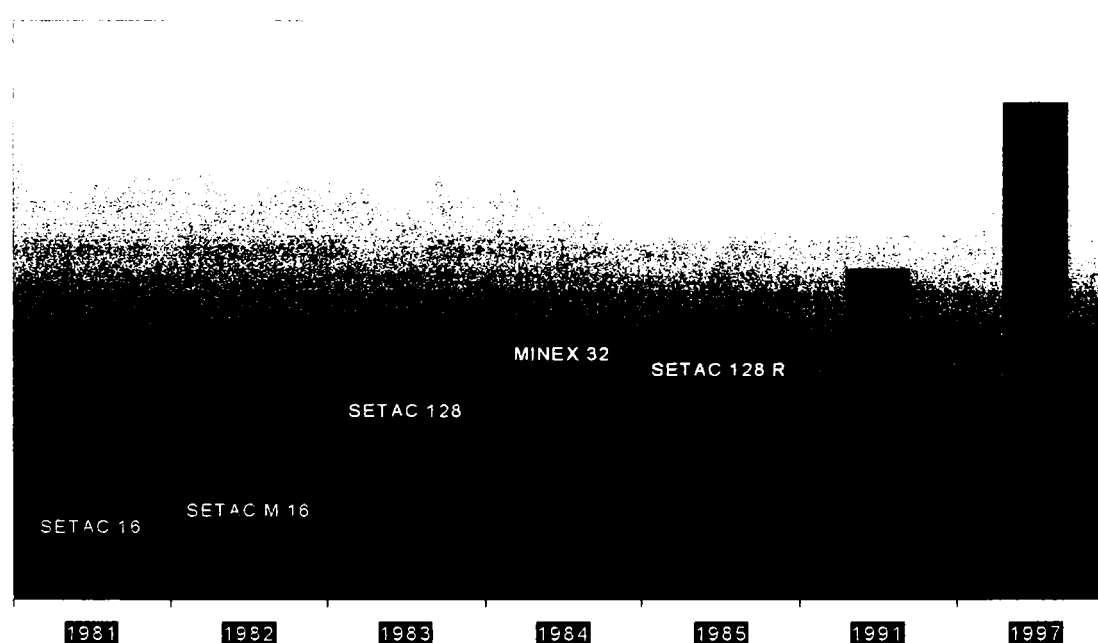


FIGURA 2.1

Aceste instalații puteau etalona concomitent 16 contoare monofazate de energie electrică cu aceeași constantă, utilizând un contor etalon specific, cu conectare prin transformatoare de măsură. Eroarea era calculată și afișată la unitatea centrală, pe rând pentru fiecare contor în parte. Ulterior acest sistem a fost îmbunătățit (SETAC M 16) prin adăugarea posibilității de autotest și contorizarea

unui kWh pentru proba de mecanism înregistrator [27], [31], [39], [55].

Datorită necesității creșterii ritmului de etalonare a contoarelor de energie electrică a fost necesară dezvoltarea unei instalații care să poată etalona mai multe contoare concomitent. În acest scop a fost pus în fabricație sistemul SETAC 128, care era capabil să etaloneze concomitent maximum 128 contoare de energie electrică cu aceeași constantă, utilizând un contor etalon specific, cu conectare prin transformatoare de măsură. Numărul constantelor contoarelor de verificat era limitat la cele fabricate la A.E.M. S.A. Calculul și afișarea erorii erau făcute în posturile de calcul asociate fiecărui contor, prin aceasta mărindu-se productivitatea operației de etalonare.

În paralel cu sistemul SETAC 128 a fost dezvoltat și sistemul MINEX 32 care a fost primul sistem cu microprocesor pentru etalonarea contoarelor de energie electrică produs la A.E.M. S.A. Acesta era capabil să etaloneze 32 contoare de energie electrică cu aceeași constantă concomitent, utilizând mai multe tipuri de contoare etalon, cu conectare prin transformatoare de măsură. Calculul erorii se făcea în unitatea centrală, iar valoarea erorii era afișată pe afișajele specifice fiecărui contor. Datorită procesoarelor lente existente în acea perioadă și imposibilității extinderii lui, sistemul a fost abandonat.

Ulterior a fost dezvoltat sistemul SETAC 128 R adăugându-se un calculator de proces care era capabil să achiziționeze valoarea erorii afișate la posturile de calcul și să genereze un protocol la imprimanta sistemului. Calculatorul era un sistem realizat cu procesorul Z80 de tip TIM S, iar posturile de calcul erau realizate cu componente discrete. Datorită numărului mare de componente, fiabilitate sistemului era redusă.

Anul 1991 a reprezentat un salt important în evoluția sistemelor de calcul a erorii contoarelor de energie electrică prin punerea în fabricație a sistemului ERRORREX [38], [40]. Sistemul a fost primul la care posturile de calcul au fost realizate cu microcontroler, ceea ce a făcut posibil calculul erorii cu o precizie ridicată, ajungându-se la afișarea cu două zecimale a erorii procentuale a contoarelor de etalonat. Sistemul permite etalonarea concomitent a maximum 128 contoare de energie electrică cu aceeași constantă, utilizând mai multe tipuri de contoare etalon, cu conectare prin transformatoare de măsură. Calculatorul cu care este dotat sistemul este un calculator compatibil PC, comunicarea făcându-se pe comunicatorul serial cu protocol RS 232C. Prin utilizarea unui număr redus de componente fiabilitatea sistemului a crescut considerabil.

Datorită reacțiilor de la beneficiari și în scopul dezvoltării primelor instalații automate de verificat contoare, în anul 1997 s-a pus în fabricație sistemul ERRORREX M, care este rezultatul experienței acumulate și a colaborării cu beneficiarii. Sistemul este un sistem "deschis", adică poate fi extins ulterior cu diverse periferice în funcție de opțiunile beneficiarului și de necesitățile modernizării și extinderii instalației. Sistemul permite etalonarea concomitent a maximum 250 contoare de energie electrică cu constante diferite, utilizând orice tip de contor etalon, atât cu conectare directă cât și cu conectare prin transformatoare de măsură. Mai mult chiar, sistemul poate utiliza trei contoare etalon monofazate câte unul pe fiecare fază a instalației, acestea putând fi tratate independent pentru etalonarea simultană a contoarelor monofazate de energie electrică repartizate pe faze diferite ale unei instalații trifazate, sau însumate pentru etalonarea contoarelor trifazate de energie electrică. Sistemul poate măsura valorile mărimilor electrice din circuitele contoarelor prin intermediul blocului de măsură și permite afișarea acestora opțional pe monitorul calculatorului de proces. Sistemul poate deconecta stația după contorizarea unui număr prescris de kWh pentru proba de mecanism înregistrator. Ca prim pas spre realizarea unei instalații complet automate sistemul poate comanda stația în vederea unei succesiuni de probe pentru verificarea automată a contoarelor de energie electrică.



## **CAPITOLUL 3**

### **STAȚII DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ - MOD DE REALIZARE**

### 3.1 Cerințe funcționale ale stațiilor de etalonat contoare de energie electrică

Pentru etalonarea și/sau verificarea contorului de energie electrică conform normelor în vigoare, este necesar ca acesta să fie alimentat cu un sistem de tensiuni trifazat de amplitudine prescrisă, aplicat circuitului de tensiune al contorului și un sistem de curenți trifazat de amplitudine prescrisă, aplicat circuitului de curent al contorului. Defazajul dintre cele două sisteme trifazate trebuie să poată fi prescris, pentru a putea controla factorul de putere.

Stațiile de etalonat contoare asigură generarea celor două sisteme (de tensiune respectiv de curent) trifazate, programabile în amplitudine și fază unul față de altul. Cele două sisteme sunt izolate galvanic între ele pentru a permite conectarea mai multor contoare de energie electrică concomitent. Circuitul de tensiune alimentează toate

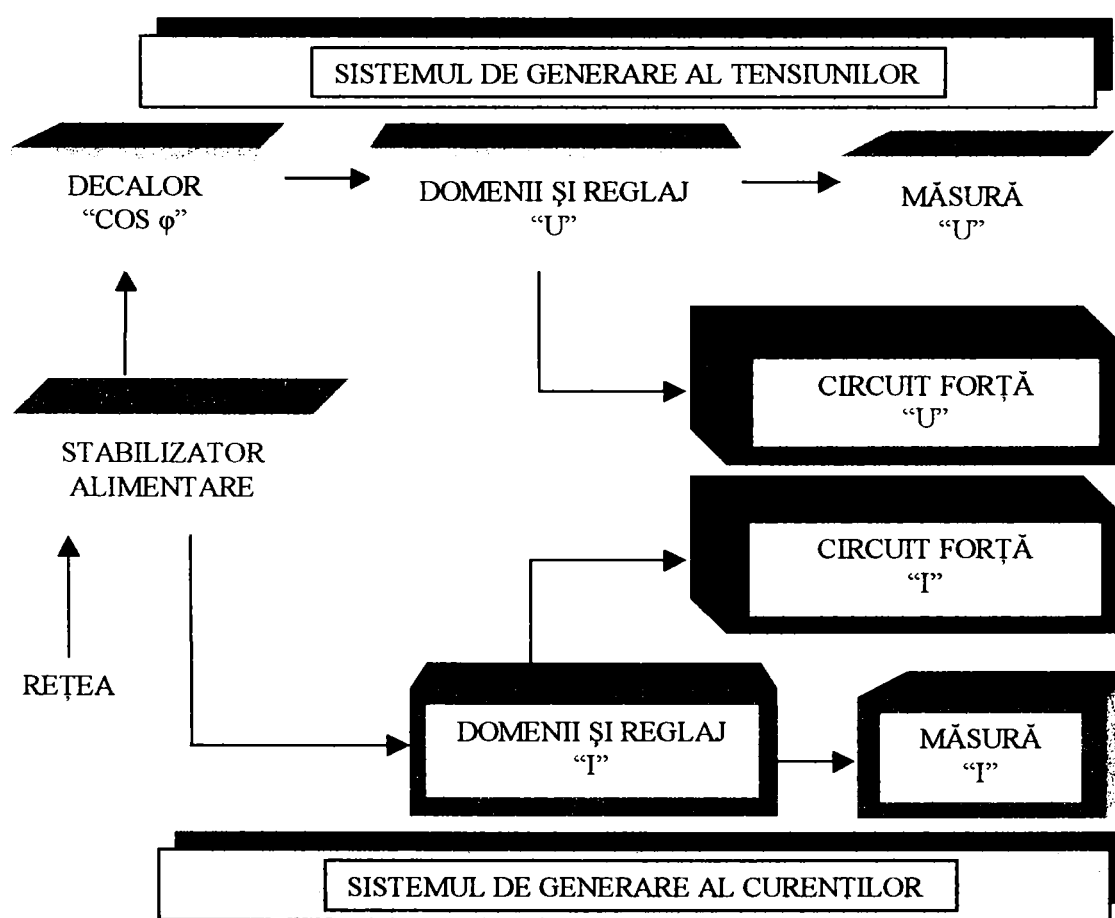


FIGURA 3.1

bobinele de tensiune ale contoarelor pe fiecare fază legate în paralel, iar circuitul de curent alimentează toate bobinele de curent ale contoarelor pe fiecare fază legate în serie.

### 3.2 Schema bloc și elementele componente ale stațiilor de etalonat contoare de energie electrică

În figura 3.1 este indicată schema bloc a stației de etalonat contoare de energie electrică.

Se disting circuitele de generare a tensiunii și circuitele de generare a curentului [58], [59]. Cele două circuite sunt alimentate de la rețeaua de curent alternativ prin intermediul unui stabilizator, sau în cazul când se dorește lucrul la o frecvență diferită de frecvența rețelei, prin intermediul unui grup motor - generator.

#### 3.2.1 Sistemul de generare al tensiunilor

Sistemul de generare al tensiunilor este una din părțile importante ale stației și are rolul de a genera un sistem trifazat de tensiuni reglabile în amplitudine și fază, respectiv de a alimenta circuitele de tensiune ale sarcinii de probă și măsurare. Sistemul este funcțional divizat în părți bine distincte: defazare, simetrizare,

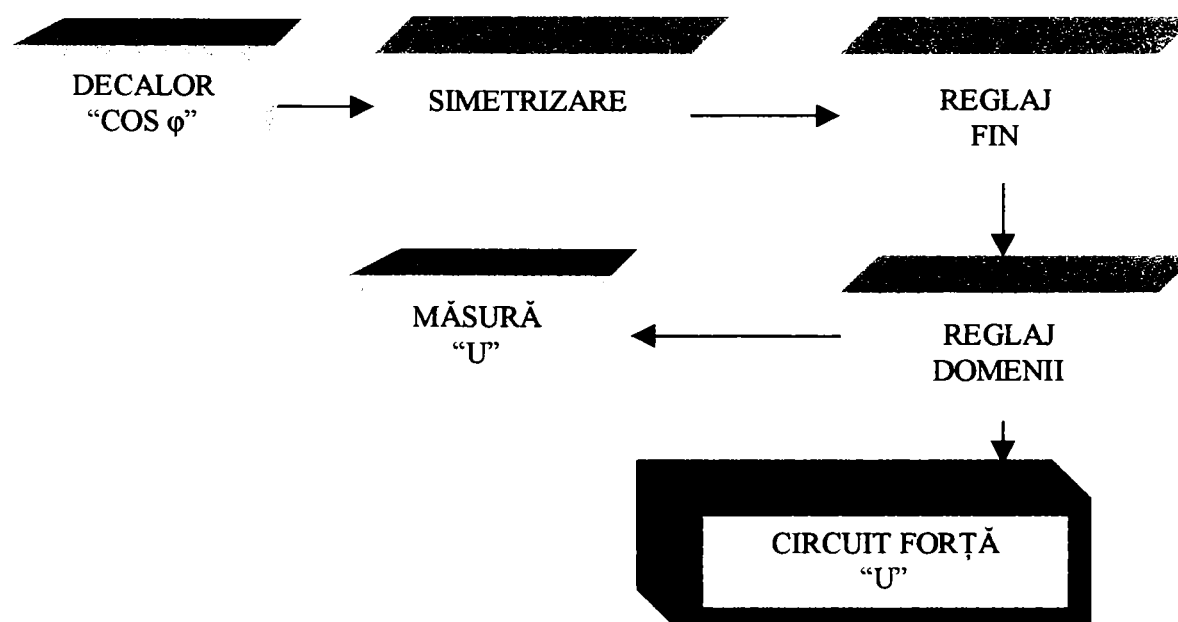


FIGURA 3.2

transformare, selectare domeniului și circuit de măsură.

Circuitul de tensiune al stației reprezentat în figura 3.2 este format din:

- circuit de decalare (decalor);
- circuite de simetrizare a sistemului trifazat de tensiuni;
- circuitul de reglaj fin al tensiunii;
- circuitul de selectare al domeniilor de tensiune;
- circuitul de măsură al valorii tensiunii;
- circuitul de forță.

*Circuitul de decalare* este constituit dintr-un decalor trifazat al cărui rotor poate fi rotit cu ajutorul unui servomotor pentru realizarea decalajului dorit între tensiunea de alimentare din primar și tensiunea din secundar (tensiunea aplicată circuitului de forță).

*Circuitul de simetrizare al sistemului trifazat de tensiuni* este constituit dintr-un grup de două perechi de autotransformatoare reglabile, acționate de servomotoare, care au rolul de a realiza o corecție a amplitudinii și fazei tensiunilor de pe două faze ale sistemului trifazat în așa fel încât să rezulte în final un sistem trifazat simetric cu:

$$\begin{aligned} |U_R| &= |U_S| = |U_T|; \\ \angle U_R, U_S &= \angle U_S, U_T = \angle U_T, U_R = 120^\circ. \end{aligned} \quad (3.1)$$

*Circuitul de reglaj fin al tensiunii* este constituit dintr-un grup de trei autotransformatoare reglabile, acționate de servomotoare, care au rolul de a realiza o corecție a amplitudinii tensiunilor de pe fiecare fază în parte.

*Circuitul de selectare al domeniilor de tensiune* este constituit din trei transformatoare care permit alegerea domeniului de tensiune dorit și a treptelor procentuale de tensiune din cadrul domeniilor.

*Circuitul de măsură al valorii tensiunii* este constituit din cele trei transformatoare de măsură de tensiune și instrumentele indicatoare aferente și are ca scop măsurarea și afișarea amplitudinii tensiunii din circuitul de forță.

*Circuitul de forță* al sistemului de generare al tensiunilor este constituit din elementele de comutare a sarcinii (contactoare pentru tipurile de conexiuni), elementele de conectare a sarcinii (panouri de fixare a contoarelor) și contoarele de verificat.

### 3.2.2 Sistemul de generare al curentului

Sistemul de generare al curentului are rolul de a genera un sistem trifazat de curenți reglabili în amplitudine, respectiv de a alimenta circuitele de curent ale sarcinii de probă și măsurare. Sistemul este funcțional divizat în părți distincte: transformare, selectare domenii, circuit de măsură.

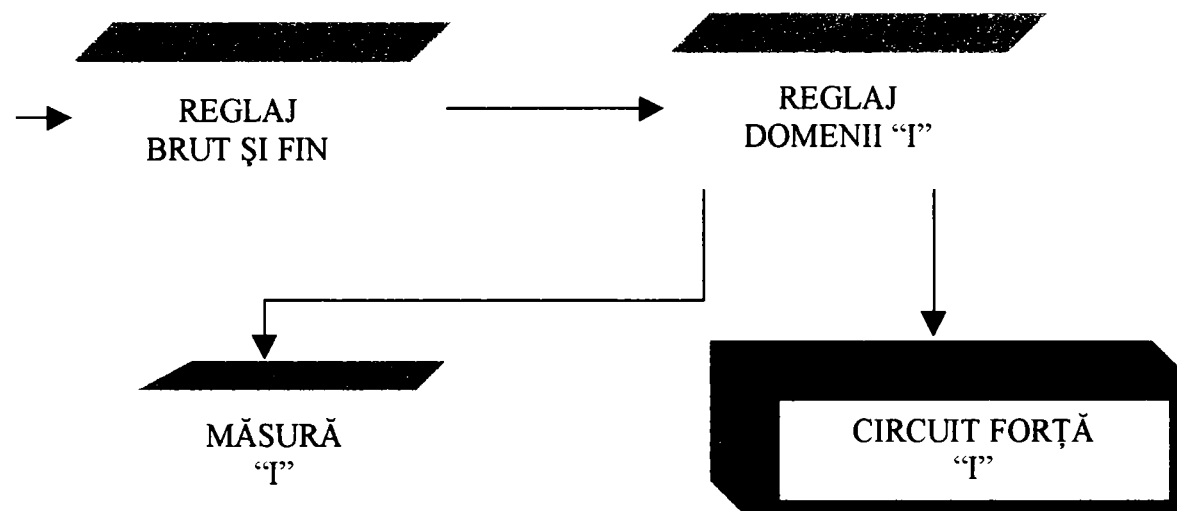


FIGURA 3.3

Circuitul de curent al stației reprezentat în figura 3.3 este format din:

- circuitul de reglaj brut și fin al amplitudinii curentului;
- circuitul de selectare al domeniilor de curent;
- circuitul de măsură al amplitudinii curentului;
- circuitul de forță.

*Circuitul de reglaj brut și fin al amplitudinii curentului este constituit dintr-un grup de șase autotransformatoare reglabile, acționate de servomotoare, câte două pe fiecare fază (unul pentru reglaj brut, iar celălalt pentru reglaj fin), care au rolul de a realiza variația amplitudinii curentului pe fiecare fază în parte, de la zero la valoarea admisă de domeniul selectat.*

*Circuitul de selectare al domeniilor de curent este constituit din trei transformatoare de curent care permit alegerea domeniului de curent dorit.*

*Circuitul de măsură al amplitudinii curentului* este constituit din trei transformatoare de măsură de curent și instrumentele indicatoare aferente și are ca scop măsurarea și afișarea amplitudinii curentului din circuitul de forță.

*Circuitul de forță* al sistemului de generare al curenților este constituit din elementele de comutare a sarcinii (contactoare pentru tipurile de conexiuni), elementele de conectare a sarcinii (panouri de fixare a contoarelor) și contoarele de verificat.

### **3.2.3 Descrierea constructivă și funcțională a stației SE 04**

#### **3.2.3.1 Descrierea constructivă**

Descrierea se va face prin referire la schema electrică atașată la anexă.

Cu ajutorul acestei stații se realizează reglarea și măsurarea puterii electrice echivalentă cu care se încarcă contoarele de verificat. Stația SE 04 de etalonat și/sau verificat contoare de energie electrică mono și trifazate, este o instalație complexă cu un grad înalt de flexibilitate în funcționare. Ea este structurată modular cu desfășurare pe verticală. Instalația se compune din două elemente de bază: stația propriu zisă, care constituie partea de generare și masa de etalonat care deține spațiul de lucru al ansamblului descris în continuare. Notațiile corespund schemelor electrice prezentate în ANEXA 7. Stația și masa de etalonat formează un ansamblu cu un grad ușor de îmbinare, ele putând fi separate ușor în cazul unor intervenții ocazionale.

Stația SE 04 este dezvoltată vertical pe șapte nivele materializate prin sertare ușor demontabile protejate spre exterior de panouri laterale de protecție detașabile.

Pe primul nivel, se află repartizat pe două panouri verticale blocul de servicii, acesta constituie etajul de intrare în stație și realizează racordul acestuia cu rețeaua trifazată de alimentare. Tot pe acest nivel se află și blocul de ieșire din stație, prin el se face conexiunea cu masa de etalonat aflată în apropiere. La al doilea nivel se află blocul de rele montate pe un panou de bază rigidizat prin îmbinare cu filet de cadrul suport existent la nivelul anterior. Al treilea nivel conține blocul de corecție și reglaj tensiune, componentele lui fiind distribuite pe un sertar evacuabil prin translatare spre partea

posterioară a stației. Această zonă este spațiul de evacuare și pentru celălalte blocuri care sunt amplasate pe sertare. Pe nivelul patru se află blocul de reglaj curent dezvoltat pe un sertar similar cu cel precedent putând fi evacuat prin același procedeu descris mai sus. Pe al cincilea nivel se află blocul de sarcină furnizorul propriuzis către circuitele exterioare respectiv cele de măsură a energiei electrice necesare conform cerințelor impuse de proba ce urmează a fi efectuată. Pe al șaselea nivel se află amplasat blocul contactoarelor, componentele lui asigurând execuția selectărilor pe partea de forță. Pe al șaptelea nivel se află blocul de măsură ce adaptează parametrii mășurați la cerințele aparatelor de măsură aflate în structura stației. În dreptul nivelelor șase și șapte, pe considerente ergonomice amplasat în fața blocurilor de sarcină și a contactoarelor se află amplasat blocul de comandă și semnalizare.

Blocul de servicii și ieșire este plasat pe nivelul I. De la separatorul de intrare FG primul element component este disjunctorul trifazat 1AT cu rolul de protecție generală fiind echipat cu elemente termooactive a căror acționare provoacă decuplarea totală a stației de sursa sa de alimentare. La borna de intrare a acestui aparat rezervat fazei "R", se alimentează un circuit monofazat ce asigură serviciile în instalație și va fi descris în rândurile ce urmează. De la bornele de ieșire ale autotransformatorului 1AT, circuitul trifazat ajunge la contactorul principal al stației CO. De la bornele de ieșire a contactorului principal, prin nodurile 7, 8, 9, circuitul de intrare se ramifică spre două grupuri de siguranțe fuzibile, F1...F3, F4...F6 ele constituind protecția la scurtcircuit a instalației. Aceste aparate delimitează calea comună a stației. Prin bornele lor de ieșire circuitul de forță se ramifică pe cele două căi de bază, calea de tensiune respectiv calea de curent. De la bornele de ieșire a celor două grupuri de siguranțe fuzibile circuitul trifazat de forță se bifurcă și continuă prin două grupuri de disjunctoare monopolare IA1...IA3 și IA4...IA6, ele asigură ca și aparatele anterioare selectiv protecția pe cele două căi principale a stației.

Circuitele descrise până acum constituie partea de intrare respectiv de protecție a stației și sunt circuitele principale a blocului de servicii. Aceste aparate sunt amplasate pe nivelul I a stației și pot fi accesate prin zona posterioară respectiv cele laterale, cu îndepărtarea prealabilă a capacelor de protecție. Cum s-a amintit în rândurile de mai sus, de la bornele de intrare a primului aparat (IAT) de pe faza "R" (nodul 1), se dezvoltă circuitele de servicii propriuzise destinate elementelor de comandă, de semnalizare și altor componente

secundare. Prin fuzibilul F7 se asigură un circuit protejat având ca terminal o priză monofazată Pr1 montată pe masa de etalonat, cu tensiunea permanentă a rețelei necondiționată de funcționarea stației, având rol de tensiune de lucru în apropierea instalației pentru diverse activități utilitare; depanare, reparație, iluminat local, etc. De pe acelaș nod se ramifică un al doilea circuit protejat prin fuzibilul F8. Circuitul alimentează transformatorul TR18 care prin puntea redresoare P1 asigură alimentarea dispozitivelor de protecție DPEU și DPEI de pe calea de tensiune, respectiv cea de curent a stației. Tot din nodul unu se desprinde circuitul principal de servicii protejat în amonte de fuzibilul F9. Circuitul alimentează bobina contactorului de serviciu C43, respectiv contactele principale ale acestuia. Este un circuit cu funcționare condiționată de starea inițială a stației; orice avarie, sau defecțiune, blochează activarea sa. De la bornele principale ale contactorului C43, se alimentează mai multe circuite protejate:

- prin fuzibilul F11 transformatorul TR19, prin secundarul lui puntea redresoare P2 care asigură tensiunea în circuitele servomotoarelor; acest circuit este cu polaritatea comutabilă și posedă două trepte de tensiuni selectabile;
- comun cu circuitul primar o altă secțiune a secundarului transformatorului T19, furnizează două tensiuni redresate, necesare circuitelor de semnalizare, de comandă, respectiv circuitului electronic;
- prin fuzibilele F13, F14 se asigură alimentarea circuitelor de forță a stației cu tensiunea rețelei;
- prin fuzibilul F15 se asigură alimentarea traductoarelor de frecvență, de defazaj, respectiv contoarele etalon existente în circuitele de măsură ale stației.

Tot pe nivelul I se află montat și regulatorul de fază RF cu servoacționare demultiplicată printr-un ansamblu cinematic dublu melcat ce asigură autofrânarea sistemului. El furnizează pe calea de tensiune, un sistem de tensiuni trifazate și realizează decalajul acestora față de cel furnizat în circuitele căii de curent a stației. Pe acest nivel amplasat pe fața anterioară a stației, se mai află și placa bornelor de ieșire, aceasta constituie un ansamblu terminal ce asigură legătura cu masa de etalonat. Toate aceste elemente descrise până în prezent sunt amplasate pe două cadre verticale.

Blocul de relee este un ansamblu care conține toate elementele de comutație care au rol de acționare în schema generală a stației. Ele sunt montate pe un panou orizontal amplasat pe nivelul II al instalației. Panoul și aparatele de pe el sunt accesibile din partea



laterală a stației, în stânga calea de tensiune, iar în dreapta calea de curent. Releele, montate în două grupuri au întările respectiv ieșirile asigurate prin 8 șiruri de cleme și 4 conectori tip fișă. Un grup de rele intermediare asigură acționările de reglaj, celălalt grup asigură acționările de selectare, ambele grupuri având roluri bine delimitate prin logica controlerului ce le comandă. Primul grup, care conține relele Rel52...Rel55 este dezvoltat în zona anterioară a panoului dinspre dreapta spre stânga stației. Al doilea grup este un ansamblu de rele Ri1...Ri53 dispuse etajat, este divizat pe două subgrupe amplasate dinspre zona anterioară spre zona posterioară pe partea din dreapta a panoului suport. Pe părțile laterale a suportului sunt montate șirurile de cleme pentru curenți tari respectiv conectorii pentru curenți slabi, pe stânga cele destinate căii de tensiune, iar pe dreapta pentru calea de curent.

Blocul de corecție și reglaj tensiune este un subansamblu component al căii de tensiune, în montaj de tip sertar este amplasat pe nivelul III al stației deasupra blocurilor descrise anterior. Elementele prezentului bloc sunt fixate pe o ramă suport translatabilă spre partea posterioară a stației pe unde este și posibilă evacuarea sa în caz de întreținere sau depanare. Aparatajul principal al acestui bloc este format din cinci autotransformatoare servocomandate trei ATR1, ATR2, ATR3 situate pe partea dreapta a sertarului, desfășurate în profunzime și două ATR4, ATR5, situate pe partea stânga. Primele au rolul de a ajusta tensiunea trifazată de pe calea de tensiune, schema de corecție având posibilitatea de a regla tensiunile individuale pe fază cu  $\pm 20\%$  în jurul valorii lor nominale, următoarele ajustează defazajul tensiunilor de fază. Între cele două grupuri de autotransformatoare sunt montate transformatoarele TR1...TR5.

Blocul de reglaj curent este o componentă a căii de curent, și este amplasat pe nivelul IV al stației cu rolul de a regla parametrul curent în stație. Acest bloc este constituit din șase autotransformatoare servocomandate, trei autotransformatoare ATR6, ATR7, ATR8 asigurând reglajul brut al curentului fiind capabile să modifice tensiunea la bornele transformatoarelor de sarcină curent, de la valoarea zero la valoarea necesară a domeniului maxim de curent de etalonare/verificare. Alte trei autotransformatoare ATR9, ATR10, ATR11 montate pe partea din stânga a sertarului asigură un reglaj fin, o corecție a curentului de sarcină prin intermediul a trei transformatoare TR12, TR13, TR14, tip serie, conectate în primarul transformatoarelor de sarcină curent. Autotransformatoarele ambelor blocuri descrise mai sus, sunt aparate de inducție construite pe circuite

magnetice toroidale cu solicitare magnetică redusă și au cursoarele concepute pe principiul rularii. Elementele de rulare sunt argintate cu scopul reducerii rezistenței de contact, respectiv de a diminua la maximum efectul de prăfuire a zonei de contact prezent la alte soluții. Reglarea presiunii de contact se face cu ușurință prin torsionarea axului suport a ansamblului cursorului. Aparatele de mai sus transmit semnale logice la blocul de comandă despre stările lor extreme; sunt acționate prin servomotoare având două viteze de regim.

Blocul de sarcină de tip sertar este suportul fizic a șase transformatoare de sarcină: trei din ele, TR6, TR7, TR8 montate pe partea din stânga a sertarului furnizează parametrul tensiune spre circuitele exterioare și celelalte trei TR15, TR16, TR17 plasate pe dreapta sertarului asigură curentul aceluiași circuite. În spațiul liber dintre aceste transformatoare sunt montate opt contactoare pe suport elastic cu scopul diminuării zgomotului la manevre de tip cuplări/decuplări; aceștia sunt contactori tripolari cu acționare la curenti mici pentru reducerea eventualelor câmpuri magnetice induse în spațiile adiacente. Trei dintre aceste contactoare C1, C2, C3 în conexiune interblocață, controlează circuitul primar al transformatoarelor de sarcină tensiune TR6, TR7, TR8. Ele conectează secțiuni diferite ale bobinajelor primare, astfel ca la o tensiune de alimentare stabilită să se obțină în circuitul de sarcină fracțiuni de tensiuni nominale. Celelalte cinci contactoare C4...C8 execută selectările, în conexiune interblocață, a gameilor de tensiune în circuitul secundar a transformatoarelor de sarcină tensiune. În spațiul superior contactoarelor este montat blocul de comandă. Blocul asigură comanda aparatajului de acționare din stație respectiv gestionează comenzile și semnalizările în timpul funcționării acestora; fiind descris în cadrul blocului de comandă.

Blocul contactoarelor este ansamblul ce conține contactoarele rezervate căii de curent, mai precis gameilor de curent C27...C33, cele pentru gameile de tensiune orientate către circuitele de măsură, C9...C13 și grupul de comutație REL1...REL8. Aparatele sunt plasate pe suporturi orizontali, montați pe amortizoare spre a îmbunătății condițiile de lucru în timpul comutației acestora. Ieșirea din acest nivel este făcută grupat, pe partea stânga este cea de tensiune, iar pe partea dreaptă cea de curent.

Blocul de măsură este pe nivelul VII al instalației și conține transformatorii de măsură curenti TM4...TM6 pentru gameile 0,2A până la 10A și TM4/...TM6/ pentru gameile curentilor mari 20A...100A, respectiv divizoarele 1/100 TM7...TM9. Tot pe acest

nivel sunt montați contactorii C34, C35 pentru gamele de 50A, 100A și contactorii ce deservește circuitele din secundarul transformatorilor de măsură precum și circuitele divizoarelor de curent.

Panoul de comandă și semnalizare este interfața cu utilizatorul stației. Este montat pe considerente ergonomice pe un panou vertical. Pe aliniamentul superior sunt plasate cele trei contoare etalon CER, CES, CET de mare precizie tip RM10 și sunt conectate în circuit prin bare rigide de curent respectiv de tensiune. În zona din mijlocul panoului sunt montate pe o linie: voltmetrele VR, VS, VT și ampermetrele AR, AS, AT. Sub aparatele de măsură sunt grupate pe trei nivele butoanele de comandă; în zona din stânga sunt cele ce acționează asupra parametrului “tensiune”, iar în zona din dreapta cele destinate parametrului “curent”. Butoanele de comandă au atașate imediat pe un nivel superior elementele de semnalizare, cele pentru tensiune de culoare roșie, cele pentru curent de culoare galbenă. Face excepție de la acest mod de așezare colțul din stânga panoului unde sunt amplasate elementele comenzilor primare, acestea dispunând de semnalizări duble. Funcțiile acestora sunt explicitate prin simbolizări plasate în dreptul lor pe un nivel imediat inferior.

Masa de etalonat formată din trei corpuri distincte, este îmbinată ușor de stația propriu-zisă și are posibilitate de a fi detașată de aceasta prin manevre minime. Pe corpurile laterale se găsesc câte două sisteme de fixare a contoarelor, fiecare cu două posturi de lucru, dintre care unul echipat cu componente aparținând sistemului ERRORREX M, adică sesizorul optic și postul de calcul și afișare a erorii. Sub fiecare post de lucru se găsește blocul bornelor cu câte trei borne de curent și patru borne de tensiune. Bornele de curent prezintă două tipuri de contacte: un tip pentru curenți mari și unul pentru curenți mici, interschimbabile prin îmbinare cu filet având suprafețe de contact cu duritate redusă spre a micșora rezistența de contact. Terminalele de curent sunt executate din conductori flexibili pentru a ușura manevrarea lor în timpul conectării la contorul de etalonat, cele de tensiune sunt echipate cu elemente de contact tip crocodil având izolație suplimentară, pentru a reduce posibilitatea de atingere directă, accidentală a conductoarelor. Pe corpul din mijloc al mesei de etalonat se găsesc anexele de lucru: un frecvențmetru digital DCFD pentru controlul rapid al frecvenței din rețeaua de alimentare, două prize bipolare Pr.1, Pr.2 și un comutator voltmetric CV. Pr.2 prezintă la bornele sale tensiuni de fază sau de linie, selectabile prin comutatorul voltmetric CV, această priză are rolul de a furniza toate tensiunile unui sistem trifazat în cazul când există opțiunea operatorului, dependent

de exigența lucrării curente, de a verifica simetria internă a tensiunilor debitate spre posturile de lucru. Sub panoul principal al mesei, pe ambele părți ale ei, se găsesc spațiile tehnice care conțin circuitele de ieșire curent, tensiune, celelalte circuite și constituie datorită accesului la ele, o zonă cu destinație exclusiv tehnică.

### 3.2.3.2 *Descrierea funcțională*

Referirile sunt făcute la schema electrică și urmăresc detalierea funcțională explicită a schemelor: bloc, de sistem și cea desfășurată SE 04 anexată la finele lucrării; ANEXA 7. Schema desfășurată a stației după care se efectuează descrierea funcțională este dezvoltată pe orizontală dinspre alimentare spre ieșire având elementele sale plasate pe trei nivele grafice ce sunt aliniate pe direcția verticală, exprimând pe diversele nivele același funcții. Pe partea superioară a schemei este dezvoltată calea de tensiune, se extinde până la prima manșeta de orientare. Sub această parte a schemei sunt dezvoltate în zona de mijloc, circuitele blocului de servicii. Pe partea inferioară a schemei este amplasată calea de curent ce se termină cu manșeta de orientare aferentă funcțiilor conținute. Nivelele logice după care s-a dezvoltat schema sunt:

- execuția care este constituită din aparatele comutației de forță ce acționează direct asupra parametrilor electrici furnizași de stație;
- acționarea care cuprinde ansamblul elementelor ce modifică opțional, selectiv, starea aparatului de pe primul nivel;
- comanda și semnalizarea care este ansamblul elementelor aflate la dispoziția utilizatorului spre a intervenii în modificarea stării stației.

Funcțiile descrise în continuare vor fi simbolizate prin FK,  $K \in [0...56]$  urmând a fi utilizate în continuare în această formă prescurtată. Funcțiile care au forma complementară explicită vor fi simbolizate prin FK0 și vor fi descrise imediat în continuare.

**F0** - *racordarea stației la circuitele exterioare* de alimentare care se realizează prin cuplarea fișei trifazate FG la sursa externă de alimentare. Prin această acțiune se pune sub tensiune nodul 1 al stației și astfel ajunge potențialul rețelei la bornele: contactorului C43, transformatorului TR18, prizei PR1 cu tensiune permanentă montată pe masa de etalonat a stației. În această stare, dispozitivele de protecție DPEU, DPEI sunt active și controlează circuitele de forță a

stației. Un semnal luminos verde plasat sub butonul de pornire “P” indică starea de așteptare a stației și realizarea acestei funcții. În caz contrar, de inactivare a acestor elemente de protecție, stația sau chiar aceste elemente sunt avariate și se emite selectiv prin două surse luminoase un semnal roșu în colțul stâng-jos, deasupra sau sub butonul “O”. Ca efect funcția nu este realizată, pornirea stației numai este posibilă și se necesită o intervenție din partea personalului desemnat pentru acest scop.

**F1 - pornirea stației** – această funcție de bază se realizează prin acționarea butonului de comandă “P”, plasat pe panoul de comandă în colțul din stânga-jos a acestuia. Prin acționarea acestui buton: se stinge semnalul luminos verde a stării anterioare de așteptare și se aprinde un semnal roșu deasupra lui, prin care se indică starea pornită a stației. Ca efect se activează contactorul C43 care se automenține printr-una din contactele sale ND. Închiderea contactelor sale realizează activarea circuitelor de serviciu, totodată prin nodul 800 se închide și contactorul principal “CO”. Contactele sale R-A, S-B, T-C se închid și tensiunea trifazată de lucru este distribuită pe calea de tensiune la elementele de corecție și reglaj până la nodurile 43, 49, 51, în amonte de contactele principale a grupului contactoarelor C1, C2, C3, respectiv pe calea de curent până la contactele principale ale grupului de contactoare C23, C24 pe faza R și S, iar pe faza T până la transformatorul de sarcină TR17. În circuitele auxiliare de comandă, semnalizare, acționare, prin punctele de alimentare 1-2, 3-4, 5-6, 7-8 desprinse din blocul de servicii, potențialele sunt distribuite pe barele comune ale aparatelor și elementelor prezente în circuite. Stația este gata de a primi și a executa comenzile lansate de la panoul de comandă.

**F10 - oprirea stației** - această funcție complementară lui F1 se realizează prin acționarea butonului de oprire “O”, prin aceasta se aduce stația la starea de așteptare obținut prin realizarea funcției FO. Contactorul C43 se dezactivează prin întreruperea automenținerii sale și astfel toate potențialele prezente pe diverse circuite ale stației vor fi retrase pe nodurile 1, 2, 3 în zona aval de fișa generală FG.

Funcțiile ce urmează a fi descrise în continuare sunt de selectare realizate în zona amonte transformatorilor de sarcină TR6, TR7, TR8, ele acționând direct asupra înfășurărilor primare modificând astfel prin acțiunea lor, raportul de transformare a acestora.

**F2** - *selectarea tensiunii nominale* - această funcție este realizată în calea de tensiune prin acționarea butonului de comandă "1Un". Prin acționarea acestui buton, releul Ri2 activat își comută contactul ND din circuitul de alimentare a contactorului C2 și acesta prin închiderea contactelor sale face posibil avansarea potențialelor trifazate prezente deja pe nodurile 43, 49, 51, pe înfășurările primare ale transformatoarelor de sarcină TR6, TR7, TR8. La bornele secundare a acestora, în amonte de bornele contactoarelor C4, C5, C6, C7, C8 vor fi prezente cele cinci tensiuni nominale pe care stația este capabilă să le furnizeze spre circuitele exterioare de sarcină, în cadrul gamelor pe care le asigură. Această funcție ca și următoarele două, sunt funcții de selectare, interdependente una de alta între ele existând relații de interblocaj la nivel de contactoare și de excludere la nivel de bloc de comandă.

**F3** - *selectarea fracțiunii 0,8 din tensiunea nominală* - este o funcție realizată în calea de tensiune prin acționarea butonului "0,8Un". Prin această acționare releul Ri1 își comută contactul ND din circuitul de alimentare al contactorului C1 și acesta, prin închiderea contactelor sale transmite potențialele trifazate prezente pe nodurile 43, 49, 51 la înfășurările primare, pe sectoare suplimentare de bobine ce modifică corespunzător raportul de transformare a acestora.

În cadrul unui regim de funcționare modificat transformatoarele de mai sus vor prezenta la bornele lor secundare, tensiuni trifazate de 80% din valoarea nominală. Această funcție are aceeași interdependență ca și funcția anterioară.

**F4** - *selectarea fracțiunii 1,1 din tensiunea nominală* - este o funcție realizată tot în calea de tensiune prin acționarea butonului "1,1Un". Prin această acțiune, releul Ri3 activat își comută contactul ND din circuitul de alimentare al contactorului C3 și acesta prin închiderea contactelor sale transmite potențialele trifazate de pe nodurile amintite la descrierea funcțiilor anterioare, pe alte sectoare ale înfășurărilor primare ale transformatoarelor de sarcină TR6, TR7, TR8. Prin acest regim de funcționare, la bornele secundare ale acestora se vor induce tensiuni majorate cu 10% din tensiunea nominală asigurată prin funcția F2.

Selectările realizate prin funcțiile F2, F3, F4 sunt semnalizate prin surse de culoare roșie amplasate imediat deasupra butoanelor cu care se apelează.

Urmează un grup de funcții tot de selectare, dar care sunt realizate după înfășurările secundare ale transformatoarelor TR6, TR7, TR8. Aceste funcții au rolul de a furniza la bornele circuitelor de sarcină și la cele de măsură, tensiunile trifazate conform gamelor prevăzute a fi asigurate în circuitele stației SE 04.

**F5** – *selectarea gamei de tensiune 100 v3* - este o funcție de selectare realizată în aval de transformatoarele de sarcină. Prin acționarea butonului "100/√3" se activează releul Ri4 care își comută contactul ND din circuitul de alimentare al grupului de contactoare C4, C9. Aceste contactoare transmit tensiunea către sarcinile de probă respectiv cele de măsură a tensiunii corespunzătoare primei game de lucru. Contactele auxiliare ale contactorului C4 asigură blocajul spre gamele superioare, iar la nivelul blocului electronic de comandă se asigură excluderea realizării unei duble selectări accidentale. Această funcție ca și celelalte din acest grup sunt semnalizate prin surse luminoase de culoare roșie.

**F6** - *selectarea gamei de tensiune 110 v3* - este o funcție similară cu cea precedentă din punct de vedere al scopului și a modului de realizare a ei. Această funcție este apelată prin acționarea butonului de comandă "110/√3". Acționarea butonului are ca urmare activarea releului Ri5 care prin închiderea contactelor sale ND alimentează bobina contactoarelor C5 și C10. Acestea prin contactele lor auxiliare blochează celelalte aparate de comutație cu funcțiuni similare și prin cele principale cuplează circuitele exterioare și cele de măsură la tensiunea trifazată de 110/√3V.

**F7** - *selectarea gamei de tensiune 208 v3*;

**F8** - *selectarea gamei de tensiune 380 v3*;

**F9** - *selectarea gamei de tensiune 480 v3* - sunt similare cu cele precedente. Aparatajul implicat în realizarea acestor funcții rezultă cu claritate din schema electrică a stației SE 04.

**F10** - *selectarea regimului 3W* - face parte dintr-un grup de patru funcții care au rolul de a distribui în mod corespunzător potențialele pe bornele de intrare a wattmetrelor spre a fi în concordanță cu schemele de conectare a contoarelor supuse verificărilor. Această funcție distribuie tensiunea la bornele contoarelor etalon conform schemelor de conectare utilizate în sistemele trifazate cu patru conductori. Funcția se apelează prin

butonul de comandă "3W". Prin comanda lansată se activează releul Ri13 care acționează asupra circuitului de alimentare a releelor REL1, REL2. Aceste rele tripolare închid circuitele de intrare spre contoarele etalon și asigură o distribuția potențialelor la bornele de intrare a contoarelor etalon.

**F11** - *selectarea regimului 2W*;

**F12** - *selectarea regimului 3var*;

**F13** - *selectarea regimului 2var* – sunt similare cu cele precedente. Aparatajul implicat în realizarea acestor funcții rezultă cu claritate din schema electrică a stației SE 04. Comanda este asigurată de butoanele de comandă "2W", "3var", "2var". Ele acționează relele Ri14, Ri15, respectiv Ri34 care la rândul lor închid circuitele de alimentare a releelor tripolare Rel3,4 ; Rel5,6 respectiv Rel7,8 a căror contacte transmit tensiunile trifazate corespunzător la bornele contoarelor etalon CER, CES, CET. Circuitul activ este semnalizat prin semnalizare luminoasă roșie.

**F14** - *selectarea succesiunii directe RST* - este o funcție de ordonare a celor trei faze, în succesiunea directă RST la bornele de ieșire sarcină. Această funcție este implicită și are numai rol de semnalizare. Prin butonul de comandă "RST" se anulează funcția F140, cea complementară descris mai jos.

**F140** - *selectarea succesiunii inverse SRT* - este funcția complementară a celei precedente. Această funcție este apelată prin butonul de comandă "SRT". Prin comanda SRT releul Ri9 este activat și astfel contactorul C15 își comută contactele. Ca urmare alimentarea lui C14 este întreruptă, succesiunea directă este anulată, iar cele trei faze trec pe succesiune inversă datorită contactelor principale ale contactorului C15. Ambele stări de ieșire pe calea de tensiune respectiv de curent spre circuitele de sarcină sunt semnalizate prin semnal luminos roșu.

**F15** - *închiderea fazei R pe circuitele de sarcină*;

**F16** - *închiderea fazei S pe circuitele de sarcină*;

**F17** - *închiderea fazei T pe circuitele de sarcină* - sunt comenzi importante, prin ele individual pe faze, se cuplează circuitele de sarcină ale stației. Prin butoanele "UR", "US", "UT" se închid circuitele de alimentare a releelor Ri10, Ri11, Ri12 care la rândul lor prin contactele lor ND acționează asupra contactorilor C16, C17, C18,



iar ele prin contactele lor principale închid pe calea de tensiune circuitele din aval de transformatorii de sarcină. Stările se comandă individual astfel se pot obține pe circuitele exterioare tensiuni monofazate, bifazate, trifazate.

**F18** - *anularea închiderii circuitelor de sarcină* - este complementar funcțiilor F15, F16, F17 și se apelează prin acționarea butonului de comandă "U=0. Acest buton acționat prin blocul de comandă întrerupe alimentarea releelor de acționare și contactorii C16, C17, C18 se deconectează.

**F19** - *corecția tensiunii nominale pe faza R;*

**F20** - *corecția tensiunii nominale pe faza S;*

**F21** - *corecția tensiunii nominale pe faza T* - au rolul de a corecta tensiunile de fază în jurul valorii nominale cari datorită diferitelor sarcini respectiv a regimurilor diferite de funcționare a transformatoarelor de sarcină necesită mici corecții. Aceste corecții se execută cu autotransformatoarele ATR1, ATR2, ATR3 prin ajustarea cursorului față de priza mediană a acestora. Prin această acțiune se induce în transformatorii serie TR1, TR2, TR3 componente în fază sau în antifază ce se compun. Astfel la bornele transformatoarelor de sarcină va ajunge o tensiune de alimentare ajustabilă. Aceste funcții se obțin prin acționarea servomotoarelor SA1, SA2, SA3, ce acționează la rândul lor asupra cursorului autotransformatoarelor. Servomotoarele sunt acționate de releele Ri29, Ri30, Ri31. La nivel de comandă funcțiile se obțin prin acționarea corelată a butoanelor de comandă "τUR", "URσ", "τUS", "σUS", "τUT", "σUT. Funcțiile de mai sus, reclamă o corelare cu informațiile transmise de cele trei voltmetre ale stației.

**F22** - *corecția polară a fazei S;*

**F23** - *corecția polară a fazei T* - sunt două funcții speciale cu utilizare impusă de NTM4-3-90 pentru stații având clasă de precizie ridicată. Se folosesc atunci când pe stație contoarele supuse verificărilor sunt de clasă egală sau mai mică de 1,5%. Aceste funcții se realizează în montajele ce conțin autotransformatoarele ATR5, ATR6, respectiv transformatorii TR5, TR6, prin activarea releelor Ri32, Ri33. Prin acționare butoanelor de comandă "US/R", "US/T", "UT/S", "UT/R" se schimbă defazajul în două sensuri, a fazelor S și T față de faza de referință R, se obține ca urmare printr-o procedură

multiplă, condiția de simetrie a fazelor. Semnificația butoanelor este următoarea:

“US/R” - defazarea tensiunii de fază S spre tensiunea R - sens pozitiv;

“US/T” - defazarea tensiunii de fază S spre tensiunea T - sens negativ;

“UT/S” – defazarea tensiunii de fază T spre tensiunea S - sens pozitiv;

“UT/R” - defazarea tensiunii de fază T spre tensiunea R - sens negativ, cu specificația că tensiunea de fază R este cea de referință.

**F24** - *reglarea defazajului dintre tensiune și curent* - realizează defazajul propriuzis dintre tensiunea și curentul de sarcină ce ajung la bornele contoarelor de verificat. Se realizează modificând poziția unghiulară a decalorului de fază RF acționat de servomotorul SA12, alimentat prin releele Rel52 și Rel53 în montaj inversor de sens comandate de releele Ri52 și Ri53. Releele sunt activate de butoanele de comandă “CAP” și “IND”. Cosfimetrul montat pe faza R va furniza starea decalajului dintre tensiune și curent, informație necesară pentru reglarea diverselor regimuri de lucru ale stației.

**F25...F39** - *selectarea gamelor de curent de la 0,002 la 100A* - au rolul de a selecta gamele de curent oferite de stație, în esență selectează diverse secțiuni din înfășurarea secundară a transformatoarelor de sarcină curent TR15, TR16, TR17, cu scopul de a obține regimuri optime de funcționare a stației pentru diferitele sarcini ce pot fi conectate la ea.

**F40...F45** - *reglarea brută spre valori minime respectiv maxime a curentului pe fazele R, S și T* - sunt funcții de reglaj brut prin care se obțin valori ale curentului de la valoarea zero la valoarea nominală a gamei la bornele de ieșire a stației.

**F46...F51** - *reglarea fină spre valori minime respectiv maxime a curentului pe fazele R, S și T* - sunt funcțiile de reglaj fin a curentului de fază. Aceste funcții se obțin prin acționarea butoanelor corespunzătoare care prin releele de comandă Ri39, Ri40 Ri41, acționează asupra elementelor de reglaj fin, autotransformatoarele ATR6, ATR7 ,ATR8. Acest reglaj poate produce o încărcare de până la 120% a sarcinii.

**F52** - *cuplarea sarcinii la calea de curent faza R;*

**F53** - *cuplarea sarcinii la calea de curent faza S;*

**F54** - *cuplarea sarcinii la calea de curent faza T* -aceste funcții închid calea de curent, conectând circuitele de sarcină la stație. Acesta se obține prin activarea individuală a contactorilor C39, C40, C41 datorită releelor de comandă Ri26, Ri27, Ri28, care își comută contactele ND la apelarea funcțiilor de mai sus, de la butoanele de comandă "IR", "IS", "IT".

**f55** - *comutarea circuitelor de divizare cu raportul 1/100* – este o funcție indirectă activată în cazul selectărilor pentru domeniile mici de curent caz în care se introduce suplimentar în circuitele de măsură transformatoarele de măsură TM7, TM8, TM9, activate de contactorii C36, C37 prin releul Ri25 comandat prin blocul de comandă când gama de curent selectat este mai mic decât 0,2A. Această adaptare se realizează pentru domeniile cuprinse între 0,2...0,002A.

**f56** - *comutarea circuitelor transformatoarelor de măsură* - este funcția prin care se conectează în circuitul de măsură selectiv, două grupuri de transformatoare de măsură, unul pentru curenți  $I \leq 10A$ , iar altul pentru  $I > 10A$ . Această modificare de stare se obține prin comutarea grupurilor de contactoare C44, C46 pentru curenți mici și grupul contactorilor C45, C47 pentru curenți mari.

\*\*\*

La acest capitol contribuțiile autorului sunt legate de modul de concepție a schemei electrice în vederea posibilității implementării blocului de comandă și blocului de măsură și pentru obținerea performanțelor cerute de beneficiari. Au fost eliminate interblocajele între domeniile de tensiune și curent și toate condițiile având rol de protecție. Funcțiile interblocajelor și protecțiilor au fost preluate de blocul de comandă, prin aceasta reducându-se complexitatea și costurile de realizare ale instalației.

## **CAPITOLUL 4**

### **ELEMENTE ALE AUTOMATIZĂRII NECESARE ÎN FUNCȚIONAREA STAȚIEI DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ**

#### 4.1 Modernizarea unei stații de etalonat contoare de energie electrică existentă, prin implementarea echipamentelor cu logică programată

În vederea implementării echipamentelor cu logică programată pentru comanda stației de etalonat contoare de energie electrică trebuie avute în vedere toate modurile de lucru posibile ale stației în regim manual și modurile specifice regimului automat.

Studiul a fost efectuat pe instalații cu autotransformatoare cu reglaj prin intermediul servomotoarelor al mărimilor prescrise, acestea constituind la nivelul actual marea majoritate a dotării principalilor deținători de astfel de instalații.

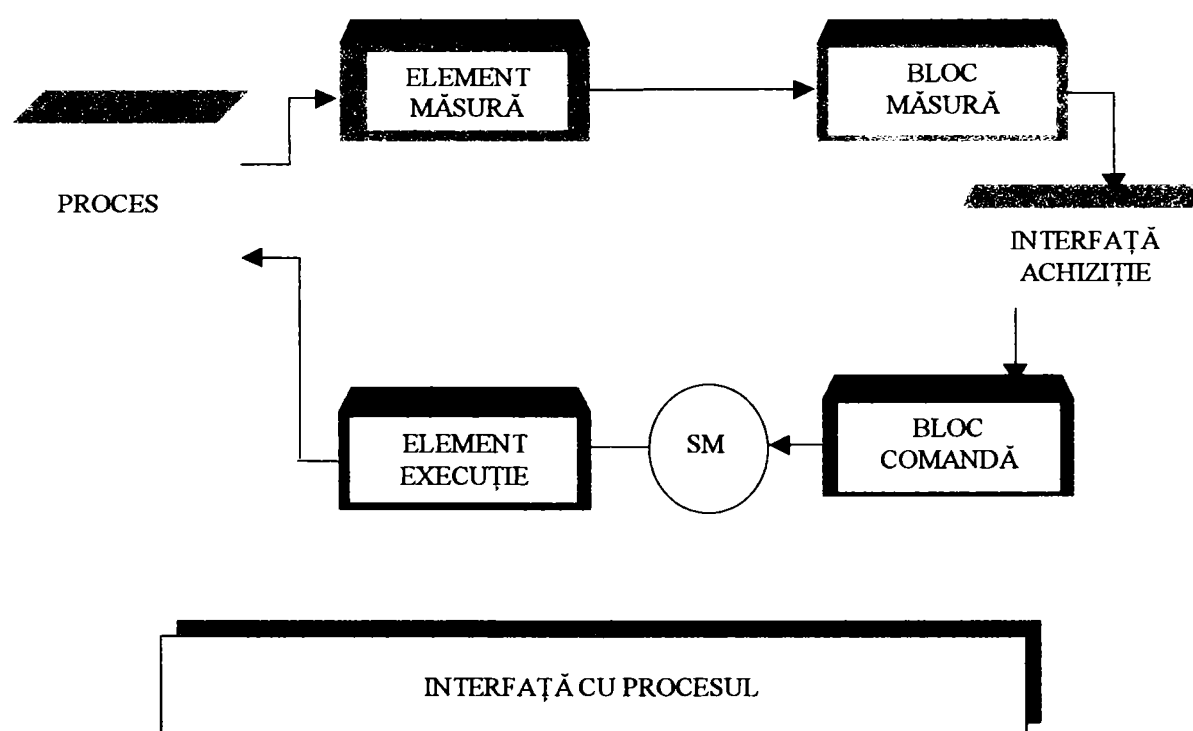


FIGURA 4.1

Principial, (figura 4.1) toate mărimile a căror reglaj se face cu un element de execuție comandabil prin intermediul unei comenzi electrice (servomotor, releu) și a căror valoare poate fi controlată de elemente de măsură cu ieșire electrică (semnal unificat, contact electric), pot fi incluse în bucle de reglaj controlate de calculatorul de proces [16], [32], [65], [68].

În *circuitul de tensiune* al stației reprezentat în figura 3.2, elementele care pot fi comandate de calculatorul de proces sunt:

- circuitul de decalare;
- circuitele de simetrizare a sistemului trifazat de tensiuni;
- circuitul de selectare al domeniilor de tensiune;
- circuitul de reglaj fin al tensiunii.

În *circuitul de decalare*, decalajul între sistemul de tensiuni și sistemul de curenți generat de instalație, respectiv factorul de putere, poate fi modificat prin acționarea asupra rotorului decalorului cu ajutorul unui servomotor comandat de blocul de comandă. Valoarea reglată poate fi citită cu ajutorul lanțului format din traductorul de factor de putere - semnal unificat, convertorul analog - digital și blocul de măsură. Servomotorul este acționat cu două trepte de viteză pentru micșorarea timpului de reglare și îmbunătățirea preciziei de reglare.

*Circuitul de simetrizare al sistemului trifazat de tensiuni* este necesar în special din cauza imperfecțiunilor constructive ale decalorului de tensiune.

Am conceput un algoritm de reducere a timpului de obținere a simetrizării tensiunilor, prin reducerea numărului de iterații la maximum două, de la patru până la șase câte erau necesare de regulă.

Simetrizarea tensiunii se face folosind un artificiu ilustrat în figura 4.2 [16], [54] și dezvoltat în paragraful 4.3.2.1.

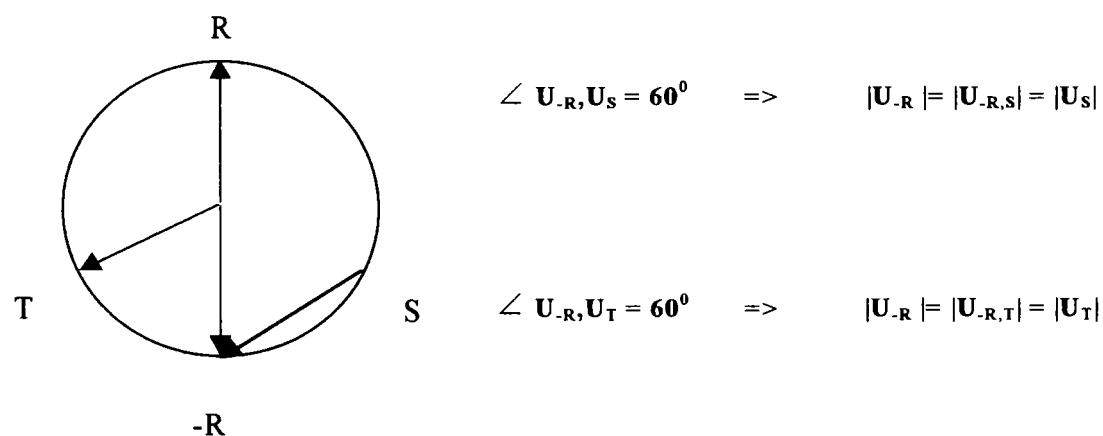


FIGURA 4.2

*Circuitul de selectare al domeniilor de tensiune* este comandat prin intermediul contactorilor de selecție a domeniilor, comandați la

rândul lor de relee intermediare și blocul de comandă al sistemului de calcul. Descrierea funcționării circuitului este dezvoltată în paragraful 4.3.2.1.

În *circuitul de reglaj fin al tensiunii* elementele de execuție sunt trei autotransformatoare care permit o corecție a amplitudinii tensiunilor în jurul valorii selectate (nominal și procentual) de pe fiecare fază în parte. Autotransformatoarele sunt acționate de servomotoare comandate de blocul de comandă. Valoarea reglată poate fi citită cu ajutorul lanțului format din traductorul de tensiune alternativă - semnal unificat, convertorul analog - digital și blocul de măsură. Servomotoarele sunt cu două trepte de viteză pentru micșorarea timpului de reglare și îmbunătățirea preciziei de reglare.

În *circuitul de curent* al stației reprezentat în figura 3.3, elementele care pot fi comandate de calculatorul de proces sunt:

- circuitul de selectare al domeniilor de curent;
- circuitul de reglaj brut și fin al amplitudinii curentului.

*Circuitul de selectare al domeniilor de curent* este comandat prin intermediul contactorilor de selecție a domeniilor, comandați la rândul lor de relee intermediare și blocul de comandă al sistemului de calcul. Contactorii de selecție a domeniilor de curent au și o blocare electrică pe partea de comandă de tip "unul din N", pentru a evita conectarea accidentală a două domenii concomitent.

O particularitate a funcționării circuitului de selectare a domeniilor de curent a stației rezultă din necesitatea impusă ca la verificarea contoarelor de energie electrică, în circuitul de sarcină variația curentului să nu fie bruscă, pentru evitarea magnetizării remanente a circuitului magnetic al acestora, impunându-se schimbarea domeniilor doar atunci când curentul din circuitul de sarcină este nul.

În *circuitul de reglaj brut și fin al curentului* elementele de execuție sunt șase autotransformatoare, trei pentru reglaj brut care permit reglarea amplitudinii curentului de la zero la o valoare apropiată de valoarea prescrisă și trei pentru reglaj fin care permit o corecție a amplitudinii curentului la valoarea prescrisă pe fiecare fază.

Autotransformatoarele sunt acționate de servomotoare comandate de blocul de comandă. Valoarea reglată poate fi citită cu ajutorul lanțului format din traductorul de curent alternativ - semnal

unificat, convertorul analog - digital și blocul de măsură. Servomotoarele sunt acționate cu o singură treaptă de viteză.

#### ***4.2 Dotarea unei stații de etalonat contoare de energie electrică cu echipamente de calcul și afișare a erorii contoarelor***

În vederea măririi vitezei de lucru și îmbunătățirii calității la etalonarea și verificarea contoarelor de energie electrică, stațiile de etalonare pot fi dotate cu echipamente de calcul și afișare a erorii.

Aceste echipamente calculează și afișează eroarea contorului la fiecare rotație a discului, respectiv la fiecare impuls al dispozitivului de afișare în cazul contoarelor electronice.

Funcționarea echipamentului este prezentată în figura 4.3.

Se folosesc curent două metode de determinare a erorii:

- cu determinarea erorii prin numărare locală;
- cu determinarea erorii prin calcul în fiecare post.

*Contorul etalon*, elementul de măsură de precizie, generează un număr de impulsuri proporțional cu energia electrică debitată spre sarcină (contoare) [10], [50], [51], [69].

*Interfața de achiziție* asigură prelucrarea impulsurilor de la contorul etalon prin multiplicare și/sau divizare și comunicarea bidirecțională cu posturile de calcul și calculatorul de proces.

*Sesizorii optici* vizează marca de pe discul contorului de etalonat, respectiv dispozitivul de afișare (diodă electroluminiscentă, afișaj cu cristale lichide) în cazul contoarelor electronice [5], [31].

În cazul determinării erorii prin *numărare locală*, interfața de achiziție asigură prelucrarea impulsurilor de la contorul etalon prin multiplicare și/sau divizare în așa fel încât la o rotație completă a discului contorului de etalonat, în cazul când acesta nu ar avea eroare, să fie transmise 10.000 de impulsuri. Valoarea de 10.000 s-a ales din cerința de calcul și afișare a erorii cu o precizie de 0,01%.



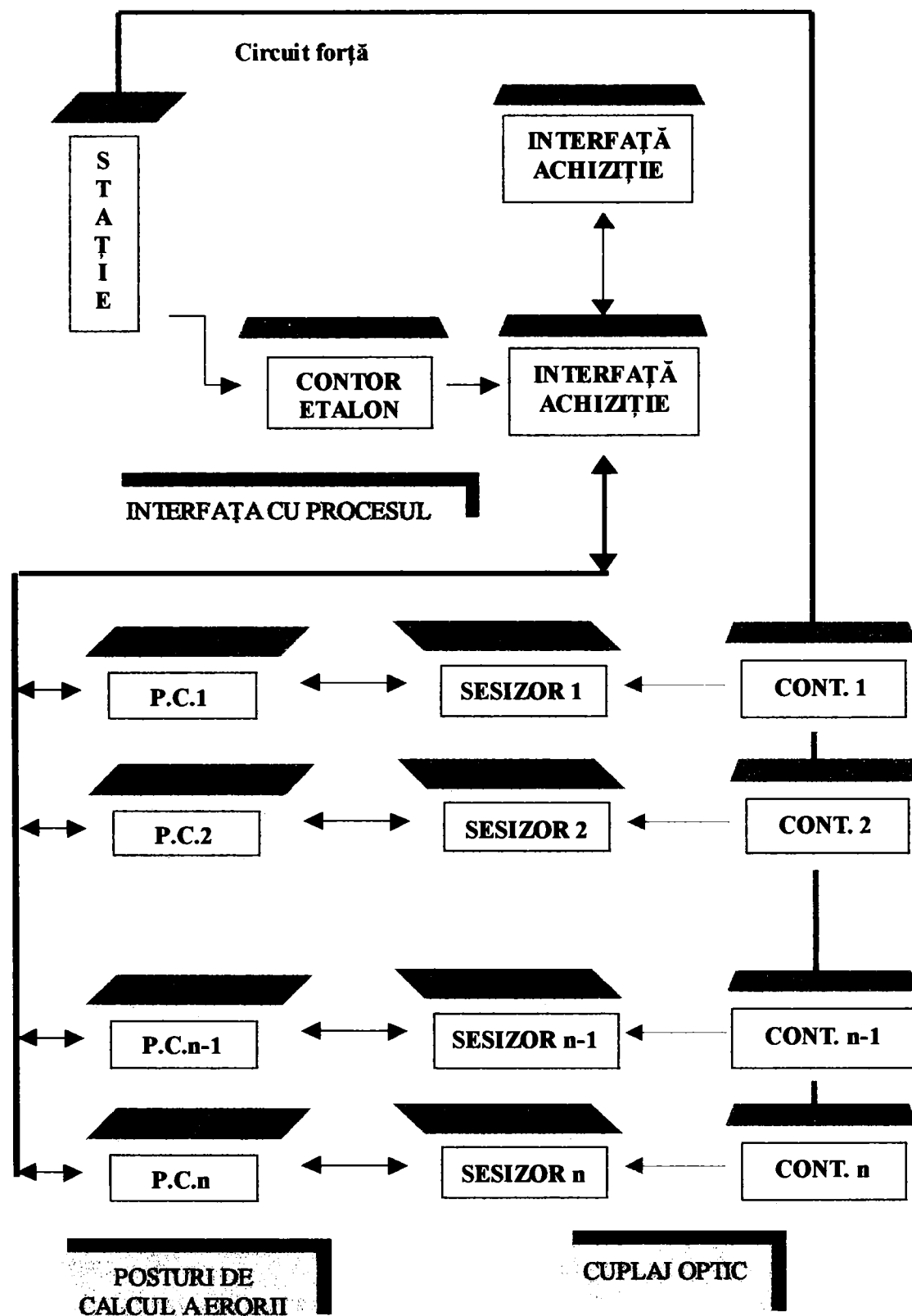


FIGURA 4.3

Postul de calcul, la sesizarea mărcii de la contorul de etalonat, încarcă un regisru cu 10000 și decrementează valoarea regisrului cu impulsurile primite de la interfața de achiziție. La apariția următoarei mărci, valoarea rămasă în regisrul respectiv este afișată și procesul se

repetă. Restul astfel obținut și afișat este egal cu eroarea contorului cu semn, calculată după relația:

$$\varepsilon_{[\%]} = \frac{(10.000 - N_{\text{numarat}})}{10.000} * 100 \quad (4.1)$$

Precizia de determinare a erorii este de  $10^{-4}$ , ceea ce reprezintă eroarea de numărare ( $\pm 1$  digit). În acest caz mai intervine eroare de aproximare a factorului de corecție a frecvenței generate de contorul etalon, în așa fel încât numărul de impulsuri să fie 10.000 la o rotație a discului contorului. Această eroare este de maximum  $10^{-5}$ .

Metoda are dezavantajul că permite numai etalonarea simultană a contoarelor cu aceeași constantă (rotații/kWh) și relația de calcul a erorii (4.1) nu este relația consacrată (4.2):

$$\varepsilon_{[\%]} = \frac{(N_{\text{calculat}} - N_{\text{numarat}})}{N_{\text{numarat}}} * 100 \quad (4.2)$$

Avantajul metodei constă în operații puține efectuate cu procesorul din post, permițând utilizarea procesoarelor lente, mai ieftine.

În cazul determinării erorii prin *calcul local în fiecare post*, interfața de achiziție asigură prelucrarea impulsurilor de la contorul etalon prin multiplicare sau divizare în așa fel încât pe linia de date spre posturile de calcul să fie transmisă o frecvență cât mai apropiată de frecvența maximă admisă de linie pentru un calcul cât mai precis al erorii.

Postul de calcul numără impulsurile primite de la interfața de achiziție într-un registru. La sesizarea mărcii de la contorul de etalonat, postul de calcul memorează valoarea din registru. La apariția următoarei mărci, postul calculează valoarea erorii făcând diferența între valoarea nouă și valoarea veche raportată la valoarea nouă:

$$\varepsilon_{[\%]} = \frac{(N_{\text{initial}} - N_{\text{final}})}{N_{\text{final}}} * 100 \quad (4.3)$$

Precizia de determinare a erorii crește, fiind practic limitată în intervalul  $10^{-4} \dots 2 * 10^{-5}$ , ceea ce reprezintă eroarea de numărare ( $\pm 1$  digit). În acest caz nu mai intervine eroarea de aproximare a factorului de corecție a frecvenței generate de contorul etalon, acesta fiind un raport de numere întregi. Relația de calcul a erorii (4.3) este relația consacrată (4.2).

Metoda are avantajul că permite etalonarea simultană a contoarelor cu constante (rotații/kWh sau Wh/rotație) diferite. La inițializarea sistemului fiecare post de calcul primește de la calculatorul de proces, prin intermediul interfeței de achiziție, valoarea

constantei contorului conectat la postul respectiv. Un alt avantaj al sistemului este posibilitatea implementării unui sistem de comunicare mai complex cu verificări ale corectitudinii transmisiei de date atât între posturile de calcul cât și între posturi și interfața de achiziție. Dezavantajul metodei de a necesita procesoare rapide în postul de calcul este practic eliminat de prețul scăzut la care se pot obține în momentul actual aceste procesoare.

### ***4.3 Descrierea funcționării elementelor de automatizare ale stației de etalonat contoare de energie electrică***

#### ***4.3.1 Funcționarea elementelor de automatizare ale stației de etalonat contoare de energie electrică cu autotransformatoare, cu reglaj prin intermediul servomotoarelor al mărimilor prescrise***

O instalație de etalonat contoare de energie electrică cu autotransformatoare, cu reglaj prin intermediul servomotoarelor al mărimilor prescrise, se pretează la implementarea echipamentelor cu logică programată pentru comandă și reglaj.

Calculatorul de proces poate executa o serie de operații care duc la creșterea vitezei de lucru cu stația, precum și la automatizarea aproape în întregime a operației de verificare a contoarelor de energie electrică [56], [61], [62], [67].

În procesul de verificare a contoarelor de energie se execută următoarele operații curente:

- selectarea conexiunii contoarelor;
- selectarea domeniului de tensiune;
- selectarea factorului de putere;
- reglarea tensiunii;
- simetrizarea tensiunii;
- selectarea domeniului de curent;
- reglarea curentului.

*Selectarea conexiunii contoarelor se face prin acționarea unuia din contactorii corespunzători conexiunii dorite (activ trei conductoare, activ două conductoare, reactiv trei conductoare, reactiv două conductoare).*

*Selectarea domeniului de tensiune* se face prin acționarea unuia din contactorii corespunzători domeniului de lucru și a celor corespunzători încărcării procentuale.

*Selectarea factorului de putere* se face prin comanda servomotorului de acționare asupra decalorului.

*Reglarea tensiunii* se face prin comanda servomotorului de acționare asupra autotransformatoarelor de reglaj fin al tensiunii.

*Simetrizarea tensiunii* se face folosind un subprogram iterativ în două până la patru iterații:

1. inversare fază "R";
2. acționare autotransformator pentru  $U_{-R} = U_S$ ;
3. acționare autotransformator pentru  $U_{-R} = U_{-R,S}$ ;
4. acționare autotransformator pentru  $U_{-R} = U_T$ ;
5. acționare autotransformator pentru  $U_{-R} = U_{-R,T}$ ;
6. revenire la 2 - 5 până la realizarea condițiilor de simetrie impuse;
7. inversare fază "R".

*Selectarea domeniului de curent* se face prin acționarea unuia din contactorii corespunzători domeniului de lucru.

*Reglarea curentului* se face prin comanda servomotoarelor de acționare asupra autotransformatoarelor de reglaj brut al curentului care măresc curentul din circuitul de curent al stației de la zero la o valoare apropiată de valoarea dorită. După aceasta se comanda servomotoarele de acționare asupra autotransformatoarelor de reglaj fin pentru stabilirea valorii prescrise cu precizia impusă.

Pentru schimbarea domeniului de curent este necesară aducerea la zero a curentului din circuitul de curent al stației.

#### ***4.3.2 Implementarea logicii programate pe o instalație cu autotransformatoare cu reglajul mărimilor prescrise prin intermediul servomotoarelor***

Schema bloc a unei astfel de instalații este prezentată în figura 4.4.

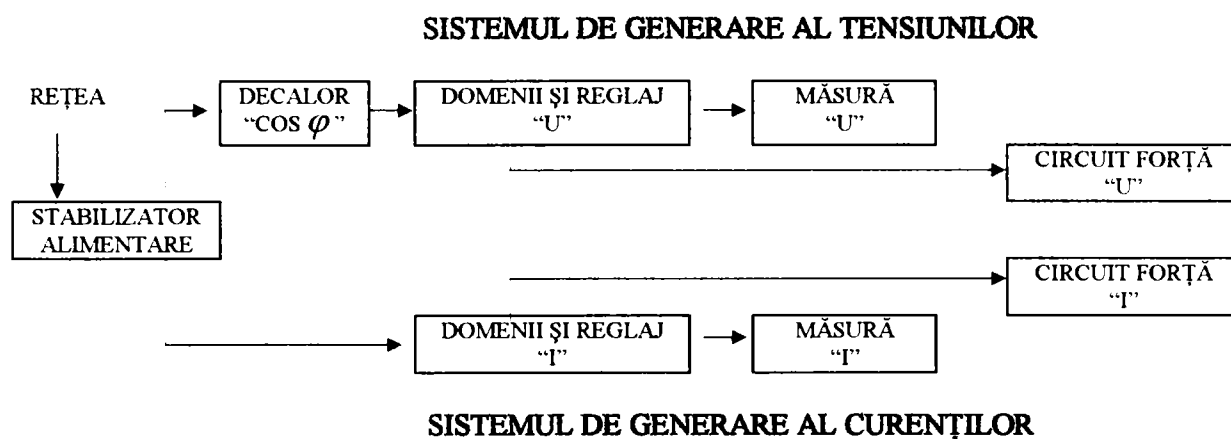


FIGURA 4.4

În vederea implementării echipamentelor cu logică programată pentru comanda stației de etalonat contoare de energie electrică trebuie avute în vedere toate modurile de lucru posibile ale stației în regim manual și modurile specifice regimului automat.

Implementarea echipamentelor cu logică programată a fost efectuată pe o instalație cu autotransformatoare cu reglajul mărimilor prescrise prin intermediul servomotoarelor, acestea constituind la nivelul actual marea majoritate a dotării principalilor deținători de astfel de instalații. O stație de acest tip este stația trifazată METRA P038 produsă de Metra Blansco din Cehia. Această stație a fost produsă în anii 1970 și constituie una din dotările cele mai răspândite ale societăților comerciale care etalonează și verifică contoare de energie electrică (în principal filialele ELECTRICA SA).

Instalația de etalonat contoare de energie electrică METRA P038 are clasa de precizie generală de 0,5 %. Modul de selectare a domeniilor de tensiune și în special a celor de curent are o particularitate. În cazul tensiunii, instalația este dotată cu un singur domeniu de tensiune, ceea ce o face utilizabilă numai pentru etalonarea și verificarea contoarelor alimentate direct de la rețea. Selectarea tensiunilor se face prin selectarea abaterilor procentuale admise la verificarea contorilor de energie electrică. În cazul domeniilor de curent, se selectează valoarea nominală a curentului contorului de etalonat și apoi procentul din acest curent la care se execută proba. Acest mod a dus la realizarea unui transformator de curent, atât cel de generare a curentului cât și cel de măsură, cu un număr mare de înfășurări, greoi de realizat în clasa de precizie cerută.

Prin concepția ei inițială, instalația de etalonat contoare de energie electrică METRA P038, a fost realizată numai pentru

etalonarea și verificarea contoarelor de energie electrică conectate direct la rețea, după normativele existente la vremea fabricației acestora.

La implementarea echipamentelor cu logică programată pe instalațiile METRA P038, s-au realizat două variante de instalații:

- cu păstrarea neschimbată a modului de selectare a domeniilor și a clasei de precizie generale (toate butoanele de comandă ale instalației au rămas aceleași);
- cu înlocuirea transformatoarelor de generare și măsură de curent, introducerea de domenii suplimentare de tensiune și îmbunătățirea corespunzătoare a clasei de precizie (0,2 % sau chiar 0,1 %).

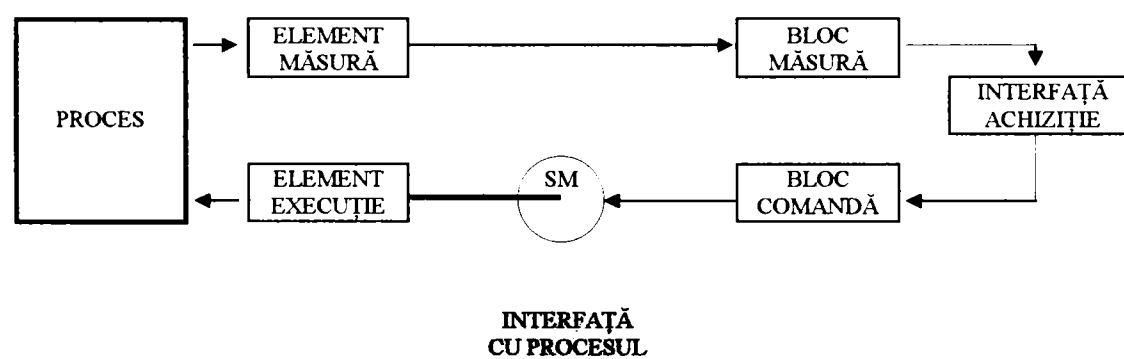


FIGURA 4.5

În cazul reglării tensiunii, a curentului (brut și fin) și a decalajului dintre tensiune și curent, *elementul de execuție* este un servomotor care acționează asupra cursorului unui autotransformator, respectiv asupra rotorului decalajului. *Elementul de măsură* este reprezentat de convertoarele de tensiune, curent, respectiv factor de putere.

Principial, (figura 4.5) toate mărimile a căror reglaj se face cu un element de execuție comandabil prin intermediul unei comenzi electrice (servomotor, releu) și a căror valoare poate fi controlată de elemente de măsură cu ieșire electrică (semnal unificat, contact electric), pot fi incluse în bucle de reglaj controlate de calculatorul de proces.

### 4.3.2.1 Circuitul de tensiune

În *circuitul de tensiune* al stației, figura 3.2, elementele care pot fi comandate de calculatorul de proces sunt:

- circuit de decalare;
- circuite de simetrizare a sistemului trifazat de tensiuni;
- circuitul de selectare al domeniilor de tensiune;
- circuitul de reglaj fin al tensiunii.

În *circuitul de decalare*, decalajul între sistemul de tensiuni și sistemul de curenți generat de instalație, respectiv factorul de putere poate fi modificat prin acționarea asupra rotorului decalorului cu ajutorul unui servomotor comandat de blocul de comandă. Valoarea reglată poate fi citită cu ajutorul lanțului format dintr-un traductor de factor de putere - semnal unificat, un convertor analog - digital și un bloc de măsură. Servomotorul este acționat cu două trepte de viteză pentru micșorarea timpului de reglare și îmbunătățirea preciziei de reglare.

În prima fază a realizării programului de prescriere a decalajului, respectiv a factorului de putere, s-a plecat de la ideea măsurării factorului de putere cu ajutorul lanțului prezentat anterior. În faza a doua, s-a constatat, cum era și de așteptat, că decalajul depinde practic numai de poziția relativă a rotorului față de statorul decalorului. Pentru un anumit decalor există o corespondență unic determinată între poziția relativă a rotorului față de stator și factorul de putere. Pentru a elimina din lanțul de reglare a factorului de putere traductorul electronic de factor de putere (element care nu funcționează decât dacă este străbătut în același timp și de tensiune și de curent), s-a introdus un artificiu. La prima instalare a sistemului, sau la înlocuirea decalorului, microcontrolerul execută un program de calibrare a decalorului; calibrarea constă într-o serie de măsurători, care se desfășoară transparent din punct de vedere al utilizatorului, în timpul cărora se generează o tabelă de corespondență biunivocă între poziția relativă a rotorului și factorul de putere. Valorile din această tabelă sunt salvate într-o memorie E<sup>2</sup>ROM, rămânând disponibile și după o deconectare a tensiunii de alimentare. În continuare instalația lucrează în regimul automat, folosind în primă aproximație de reglaj valorile din tabelă, apoi după stabilirea tensiunii și curentului, se trece la o corecție a factorului de putere cu ajutorul măsurătorii exacte, folosind traductorul.

Funcționarea în regim automat este prezentată simplificat în ordinograma din figura 4.6. Prima fază a reglării inițiale se execută cu treapta a doua de viteză (viteza mărită), iar a doua etapă de corecție se execută cu viteză redusă pentru o finețe mai mare a reglajului. În acest mod s-a obținut timpul minim de prescriere a factorului de putere dorit.

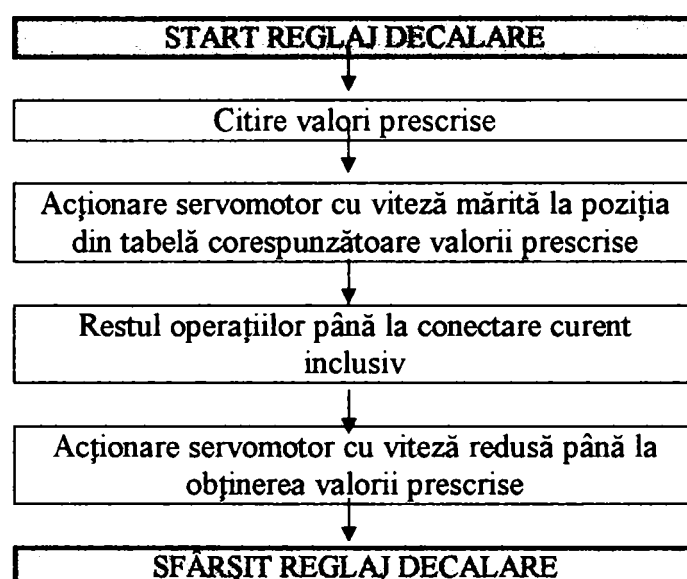


FIGURA 4.6

În circuitul de simetrizare al sistemului trifazat de tensiuni se folosește un artificiu prezentat în figura 4.2.

Se inversează cu  $180^\circ$  tensiunea  $U_R$  de pe faza "R" cu ajutorul unui transformator de tensiune de măsură cu raportul de transformare 1:1 și se compară cu ajutorul a două comparatoare electronice de tip fereastră cu ieșire contact, pe de o parte valorile

$$|U_{-R}| = |U_S| \quad (4.4)$$

și pe de altă parte valorile

$$|U_{-R} - U_S| = |U_{-R}| \quad (4.5)$$

În circuitul secundar al decalorului pe faza "S", sunt conectate două autotransformatoare reglabile, acționate de servomotoare care au rolul de a realiza o corecție a amplitudinii și fazei tensiunilor de pe



faza "S". Autotransformatorul de corecție a amplitudinii este alimentat de la bornele secundare a decalorului, în paralel cu faza "S", iar autotransformatorul de corecție a fazei este alimentat de la bornele secundare a decalorului între fazele "R" și "T".

Un circuit identic se află montat pe faza "T". După stabilirea egalităților menționate se trece pe faza "T" unde prin acționarea celeilalte perechi de autotransformatoare trebuie îndeplinite condițiile:

și

$$\begin{aligned} |U_{-R}| &= |U_T| \\ |U_{-R} - U_T| &= |U_{-R}| \end{aligned} \quad (4.6)$$

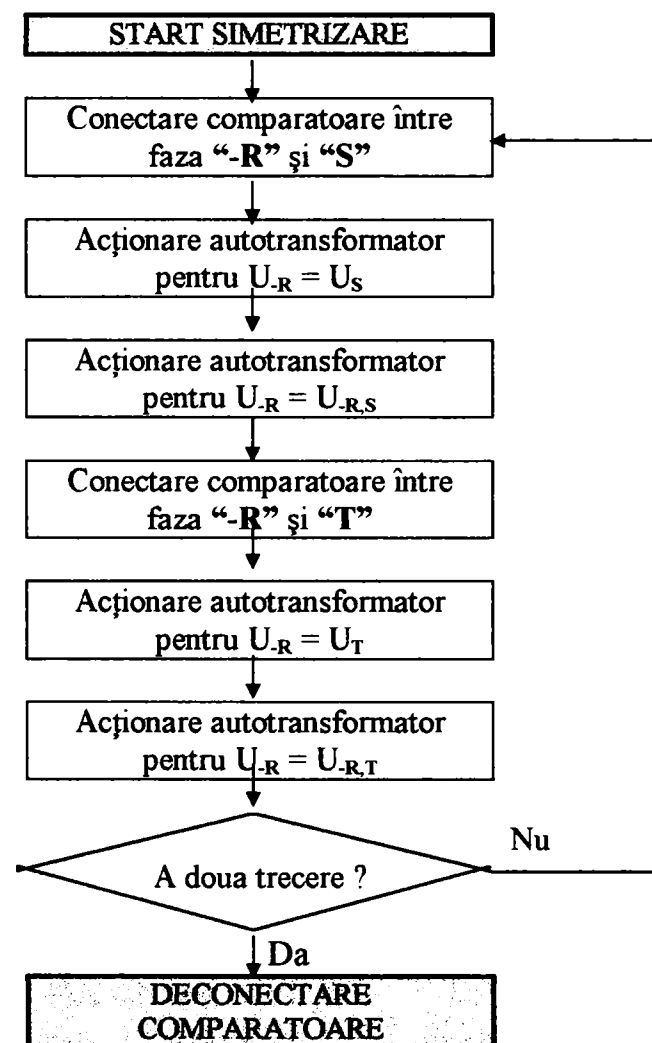


FIGURA 4.7

În final sunt îndeplinite condițiile de sistem trifazat simetric:

$$\begin{aligned} &|U_R| = |U_S| = |U_T| \\ \text{și} & \qquad \qquad \qquad (4.7) \\ &\angle U_R, U_S = \angle U_S, U_T = \angle U_T, U_R = 120^\circ \end{aligned}$$

Simetrizarea automată a tensiunii se face folosind un subprogram a cărui ordinogramă simplificată este prezentată în figura 4.7, cu următoarea ordine a operațiilor:

1. conectarea comparatoarelor între fazele “-R” și “S”;
2. acționarea autotransformatorului corespunzător pentru  $U_{-R} = U_S$ ;
3. acționarea autotransformatorului corespunzător pentru  $U_{-R} = U_{-R,S}$ ;
4. conectarea comparatoarelor între faza “-R” și “T”;
5. acționarea autotransformatorului corespunzător pentru  $U_{-R} = U_T$ ;
6. acționarea autotransformatorului corespunzător pentru  $U_{-R} = U_{-R,T}$ ;
7. revenirea la punctele “1” - “6” până la realizarea condițiilor de simetrie impuse (de obicei se execută o singură trecere pentru reglare, a doua fiind doar o verificare a simetriei);
8. deconectarea comparatoarelor.

Cele două subprograme descrise până în prezent, reglarea decalajului și reglarea simetrizării sistemului trifazat de tensiuni, sunt necesare, sub forma descrisă, din cauza imperfecțiunilor constructive ale decalajului de tensiune, decalaj care echipează în momentul de față marea majoritate a instalațiilor de etalonat contoare de energie electrică aflate în exploatare în țară.

În cazul lucrului manual cu instalația, se acționează tastele corespunzătoare (“+COS  $\varphi$ ” și “-COS  $\varphi$ ”), citindu-se valorile pe instrumentul indicator. La menținerea apăsată a tastei mai mult de două secunde, se trece automat la viteza mărită de acționare a servomotorului. După prescrierea valorii dorite se acționează butonul de corecție automată a simetriei, care acționează conform descrierii anterioare.

Prescrierea decalajului dorit în regim automat se face folosind un subprogram cu următoarea ordine a operațiilor:

1. acționare servomotor decalor cu viteză mărită la poziția corespunzătoare din tabelă pentru decalajul dorit;
2. simetrizare automată;
3. după conectarea curentului se revine la reglajul fin al decalării cu treapta redusă de viteză, funcție de valoarea măsurată de lanțul traductor - convertor;
4. refacere simetrizare automată.

*Circuitul de selectare al domeniilor de tensiune* este comandat prin intermediul contactorilor de selecție a domeniilor, comandați la rândul lor de contactoare statice (triac) și blocul de comandă al sistemului de calcul. Contactorii de selecție a domeniilor de tensiune au și o blocare electrică pe partea de comandă de tip “unul din N”, pentru a evita conectarea accidentală a două domenii concomitent. Alegerea treptelor procentuale de tensiune din cadrul domeniilor se face tot prin intermediul contactorilor de selecție a rapoartelor, comandați la rândul lor de contactori statici și blocul de comandă al sistemului de calcul. Și acești contactorii au o blocare electrică pe partea de comandă de același tip.

În *circuitul de reglaj fin al tensiunii* elementele de execuție sunt trei autotransformatoare care permit o corecție a amplitudinii tensiunilor în jurul valorii selectate (nominal și procentual) de pe fiecare fază în parte. Autotransformatoarele sunt acționate de servomotoare comandate de blocul de comandă. Valoarea reglată poate fi citită cu ajutorul lanțului format dintr-un traductor de tensiune alternativă - semnal unificat, un convertor analog - digital și un bloc de măsură. Servomotoarele sunt acționate cu două trepte de viteză pentru micșorarea timpului de reglare și îmbunătățirea preciziei de reglare. La menținerea apăsată a tastei mai mult de două secunde se trece automat la viteza mărită de acționare a servomotorului.

La selectarea manuală a unui domeniu de tensiune, respectiv a raportului procentual dorit, se acționează tasta corespunzătoare de pe panoul instalației, după care se acționează manual asupra butoanelor de corecție fină a tensiunii de pe fiecare fază, urmărindu-se valoarea indicată de instrumentele de măsură.

#### 4.3.2.2 Circuitul de curent

În *circuitul de curent* al stației reprezentat în figura 3.3, elementele care pot fi comandate de calculatorul de proces sunt:

- circuitul de reglaj brut și fin al amplitudinii curentului;
- circuitul de selectare al domeniilor de curent.

În *circuitul de reglaj brut și fin al amplitudinii curentului* elementele de execuție sunt șase autotransformatoare reglabile, acționate de servomotoare, câte două pe fiecare fază (unul pentru reglaj brut, iar celălalt pentru reglaj fin), care au rolul de a realiza variația amplitudinii curentului pe fiecare fază în parte, de la zero la valoarea admisă de domeniul selectat. Autotransformatoarele sunt acționate de servomotoare comandate de blocul de comandă. Valoarea reglată poate fi citită cu ajutorul lanțului format dintr-un traductor de curent alternativ - semnal unificat, un convertor analog - digital și un bloc de măsură. Servomotoarele sunt acționate cu două trepte de viteză pentru micșorarea timpului de reglare și îmbunătățire a preciziei de reglare. În cazul comenzii manuale, la menținerea apăsată a tastei mai mult de două secunde, se trece automat la viteza mărită de acționare a servomotorului.

*Circuitul de selectare al domeniilor de curent* este comandat prin intermediul contactorilor de selecție a domeniilor, comandați la rândul lor de contactoare statice (triac) și blocul de comandă al sistemului de calcul. Contactorii de selecție a domeniilor de curent au și o blocare electrică pe partea de comandă de tip "unul din N", pentru a evita conectarea accidentală a două domenii concomitent.

O particularitate a funcționării circuitului de selectare al domeniilor de curent a stației rezultă din necesitatea impusă la verificarea contoarelor ca în circuitul de sarcină, variația curentului să nu fie bruscă, pentru evitarea magnetizării remanente a circuitului magnetic al acestora, impunându-se schimbarea domeniilor doar atunci când curentul din circuitul de sarcină este nul. Această cerință a fost îndeplinită la comanda manuală, prin blocarea acționării tastelor corespunzătoare domeniilor de curent, atât timp cât valoarea curentului în circuitul de sarcină este diferită de zero. După aducerea la zero a valorii curentului din circuitul de sarcină, se permite comutarea domeniilor de curent. Orice manevră interzisă sau greșită

este semnalizată de avertizorul sonor al sistemului cu microcontroler din blocul de comandă.

În producția curentă la L.L.C. S.C. A.E.M. S.A. se află instalațiile de etalonat contoare de energie electrică SE 03.

Instalația se prezintă sub forma unei mese cu cinci blocuri funcționale distincte:

- blocul de reglaj tensiune;
- blocul de reglaj curent;
- blocul de comandă;
- blocul de servicii;
- blocul de măsură și protecție.

Instalația este prevăzută cu instrumente indicatoare analogice de clasă 1 sau 1,5 în funcție de clasa de precizie a stației (0,5 respectiv 0,2%) și cu maxim 5 domenii de tensiune, cuprinse între 56,74V și 400V (3x600VA) și domenii de curent în intervalul 0,2A la 100A, în trepte 1, 2, 5, cu o posibilitate de raport 1/100 până la gama de 10A. Decalajul tensiune - curent este reglabil continuu în intervalul 0,5 inductiv, până la 0,5 capacitiv. Instalația face parte din categoria instalațiilor semiautomate, cu reglajul mărimilor prescrise prin intermediul servomotoarelor.

În prezent s-a trecut la fabricarea unei noi stații de etalonat contoare de energie electrică, SE 04.

Instalația de etalonat contoare de energie electrică SE 04 prezintă față de familia anterioară SE 03 o serie de noutăți care o fac să fie situată în rândul produselor de vârf din această clasă.

Prescrierea și reglarea tuturor mărimilor de lucru se poate face, funcție de variantă, de la dulapul de forță sau de la o consolă.

Concepția părții de comandă a stației a fost complet modificată, trecându-se de la comenzile clasice cu tastatură cu reținere - relele intermediare, la comenzi cu taste profesionale - sistem de calcul cu microcontroler - contactori statici. Această soluție a dus la creșterea fiabilității ansamblului, precum și la posibilitatea de comandă a întregii stații de la distanță, prin intermediul unei legături seriale RS 232C, de la un calculator compatibil PC.

Mărimile specifice probelor, tensiune, curent, factor de putere, sunt afișate în funcție de varianta constructivă fie pe dulapul de forță sau consolă, fie direct pe ecranul calculatorului cu care este dotată instalația.

Instalația face parte din categoria instalațiilor automate, cu reglajul mărimilor prescrise prin intermediul servomotoarelor comandate de un sistem cu microcontroler.

### ***4.3.3 Funcționarea elementelor de automatizare ale stației de etalonat contoare de energie electrică cu reglaj electronic al mărimilor prescrise***

O instalație de etalonat contoare de energie electrică cu reglaj electronic al mărimilor prescrise este formată în principal din blocul de generare electronic care conține șase generatoare electronice grupate câte trei în două sisteme trifazate unul de generare al tensiunilor și celălalt de generare al curenților. Fiecare sistem este perfect echilibrat. Prescrierea tensiunii, curentului, decalajului și a frecvenței se face prin intermediul unei interfețe RS 232C standard. Măsurarea mărimilor electrice (tensiune, curent, factor de putere și energie) se face în blocul de generare, iar transmiterea rezultatului măsurătorii se face prin intermediul aceleiași interfețe RS 232C.

Timpii de reglare ai instalației s-au redus din cauza eliminării elementelor mecanice (autotransformatoare, contactoare), precum și a algoritmului de reglare a simetrizării și precizia de prescriere a mărimilor a crescut.

În procesul de verificare a contoarelor de energie electrică se execută următoarele operații curente:

- selectarea conexiunii contoarelor;
- selectarea factorului de putere;
- reglarea tensiunii;
- reglarea curentului.

*Selectarea conexiunii contoarelor* se face prin transmiterea spre blocul de generare a comenzii corespunzătoare.

*Selectarea factorului de putere* se face prin transmiterea spre blocul de generare a valorii dorite însoțită de codul de recunoaștere corespunzător pentru factor de putere.

*Reglarea tensiunii* se face prin transmiterea spre blocul de generare a valorii dorite însoțită de codul de recunoaștere corespunzător pentru tensiune.

*Reglarea curentului* se face prin transmiterea spre blocul de generare a valorii dorite însoțită de codul de recunoaștere corespunzător pentru curent. La recepționarea codului pentru curent blocul generează un curent crescător până la valoarea prescrisă. La schimbarea valorii prescrise variația curentului este continuă până la noua valoare. Prin aceasta se elimină timpul de comutare a domeniilor de curent.

#### 4.3.4 Implementarea logicii programate pe o instalație cu generator electronic

##### 4.3.4.1 Schemă de principiu

Schema bloc a unei instalații cu generarea electronică a sistemelor trifazate de tensiune și curent este prezentată în figura 4.8.

Spre deosebire de instalațiile clasice, la acest tip de instalații, dispăre stabilizatorul de tensiune, funcția lui fiind preluată de generatoarele electronice.

Decalajul dintre cele două sisteme trifazate se generează prin programarea decalajului de fază între cele două grupuri de generatoare.

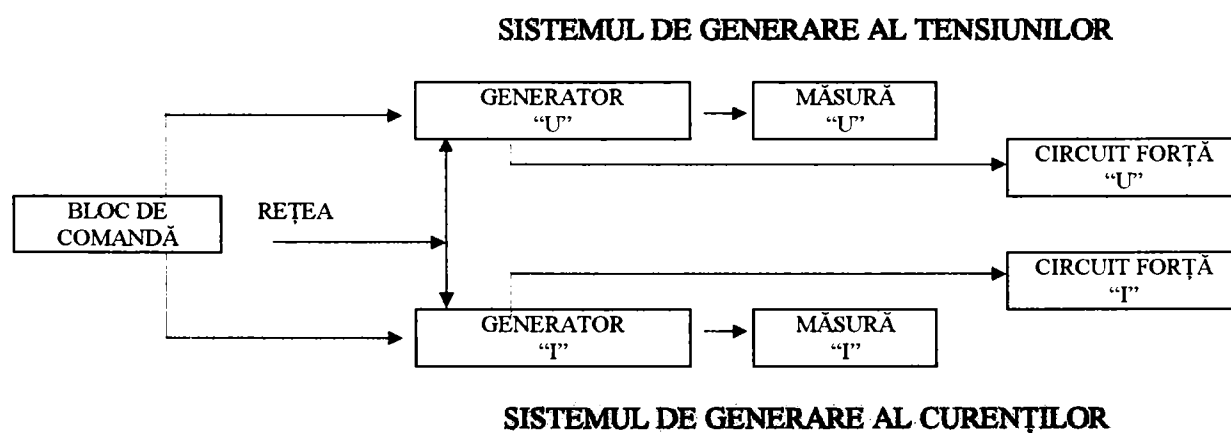


FIGURA 4.8

Fiecare grup de generatoare (tensiune și curent) este compus din trei generatoare identice comandate la un decalaj de  $120^\circ$  unul față de

celălalt. Păstrându-se decalajul de  $120^\circ$  între generatoarele din cadrul fiecărui grup, a dispărut necesitatea simetrizării sistemelor.

Prescriere mărimilor de lucru, tensiune, curent și decalaj se face prin intermediul blocului de comandă, fie prin comenzi directe de la panoul instalației, fie prin interfața RS 232C de la calculatorul de proces.

În cazul dotării instalației cu un contor etalon de fabricație modernă, figura 4.9, au dispărut și elementele discrete ale blocului de măsură (traductoare, convertoare), deoarece contoarele etalon din această familie măsoară și transmit, prin interfața din dotare, spre calculatorul de proces toate mărimile necesare (tensiuni, curenți și puteri pe fiecare fază, factorul de putere, frecvența tensiunii de ieșire).

Practic în cazul unei astfel de instalații, se impune doar realizarea unui program corespunzător pentru derularea într-o anumită

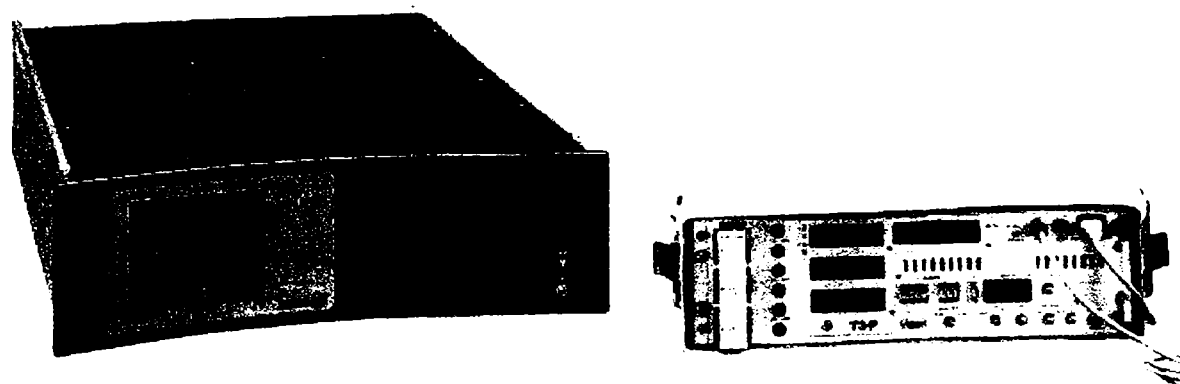


FIGURA 4.9

ordine a probelor de lucru, ordine specifică fiecărui tip de contor. Se poate crea o bază de date cu toate tipurile de contoare care se pot verifica pe instalația respectivă, cu toate succesiunile de probe specifice acestora.

#### ***4.3.4.2 Modul de comandă și funcționare***

Pentru automatizarea completă a operației de verificare a contoarelor de energie electrică, calculatorul de proces trebuie să execute o succesiune de probe într-o ordine prestabilită și să înregistreze rezultatele fiecărei probe.

În cazul contoarelor active cu trei sisteme, cele mai răspândite de altfel, este necesară efectuarea unei succesiuni de operații conform ordinogramei din figura 4.10.



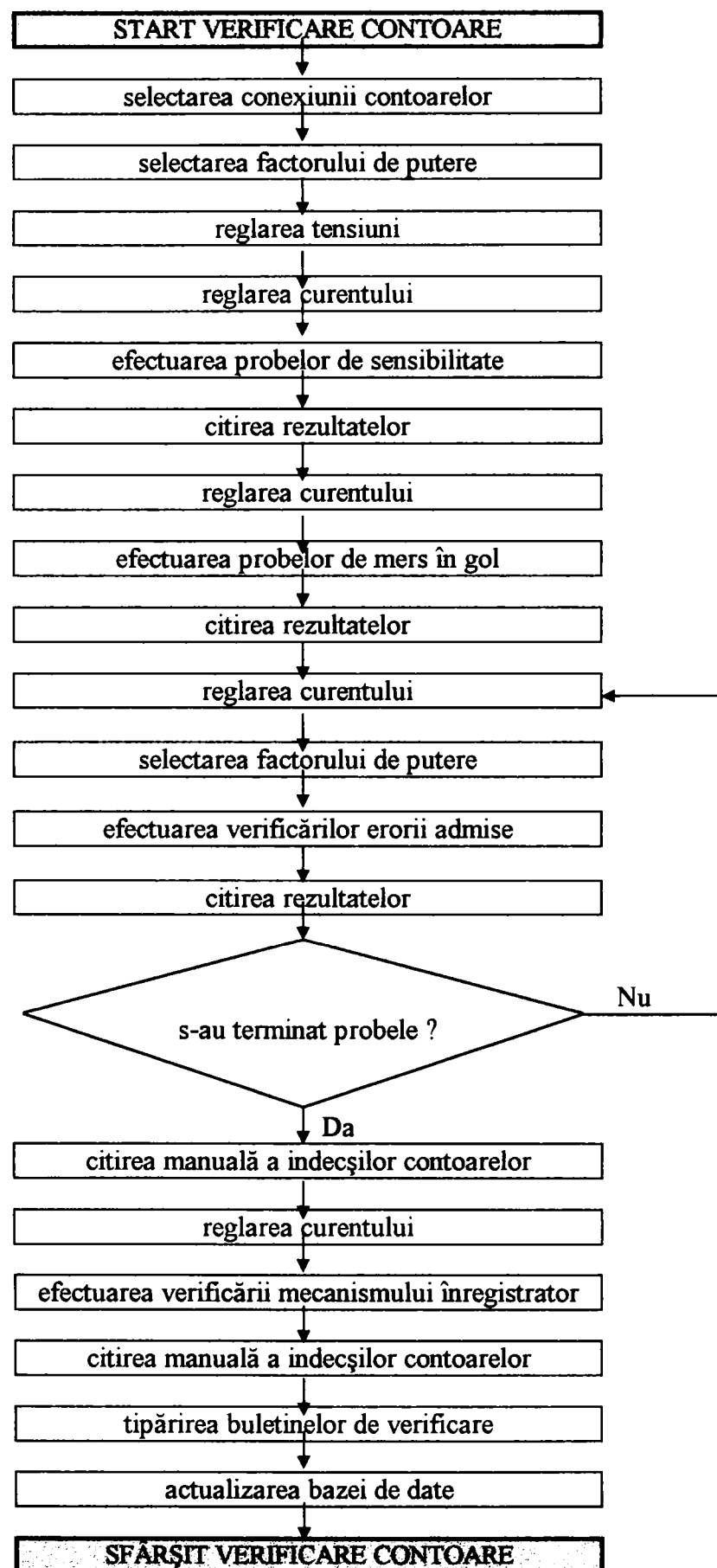


FIGURA 4.10

O particularitate a sistemului de calcul este modul de citire a rezultatelor în cazul probelor când nu se urmărește o valoare a erorii (valoare ușor de calculat în postul de calcul și afișare a erorii), ci rezultatul probei este de tip “admis” - “respins”. Astfel de probe sunt sensibilitatea și mersul în gol.

La proba de sensibilitate postul de calcul așteaptă un interval de timp prestabilit apariția a cel puțin două mărci de la contorul de etalonat. Dacă acestea apar, contorul se rotește la proba de sensibilitate și se consideră “admis”.

La proba de mers în gol postul de calcul așteaptă un interval de timp prestabilit apariția unei mărci de la contorul de etalonat. Dacă apare mai mult de o marcă, contorul se consideră “respins”.

La începutul și la sfârșitul efectuării verificării mecanismului înregistrator, operatorul introduce indexul contoarelor în terminalul portabil al sistemului. Aceasta este singura verificare la care intervine operatorul.

O instalație automată, pentru a avea o productivitate mare, trebuie să permită etalonarea simultană a cât mai multe contoare de energie electrică. Numărul de contoare care pot fi conectate la o stație

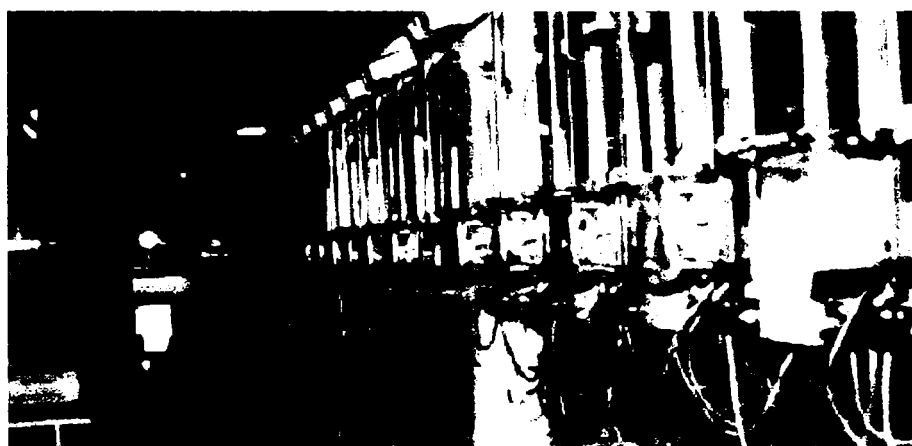


FIGURA 4.11

este funcție de puterea disponibilă a generatoarele de tensiune și curent. În mod curent se poate lucra cu panouri de 12 până la 120 de contoare (figura 4.11). Din punct de vedere al sistemului de calcul, acesta poate gestiona până la 250 de contoare fără scăderea performanțelor de viteză, singura limitare practică rămânând generatoarele. Avantajul lucrului cu mai multe contoare simultan la verificare, este acela că practic în intervalul de timp necesar etalonării unui singur contor, se poate realiza verificarea întregului panou, timpii

de efectuare ai citirilor erorilor de la contoare fiind total neglijabili față de durata probelor.

Operatorul intervine numai la montarea și demontarea contoarelor de pe panouri, la poziționarea sesizoarelor optice și la probele a căror rezultate nu pot fi citite direct de sistemul de calcul (efectuarea verificării mecanismului înregistrator);

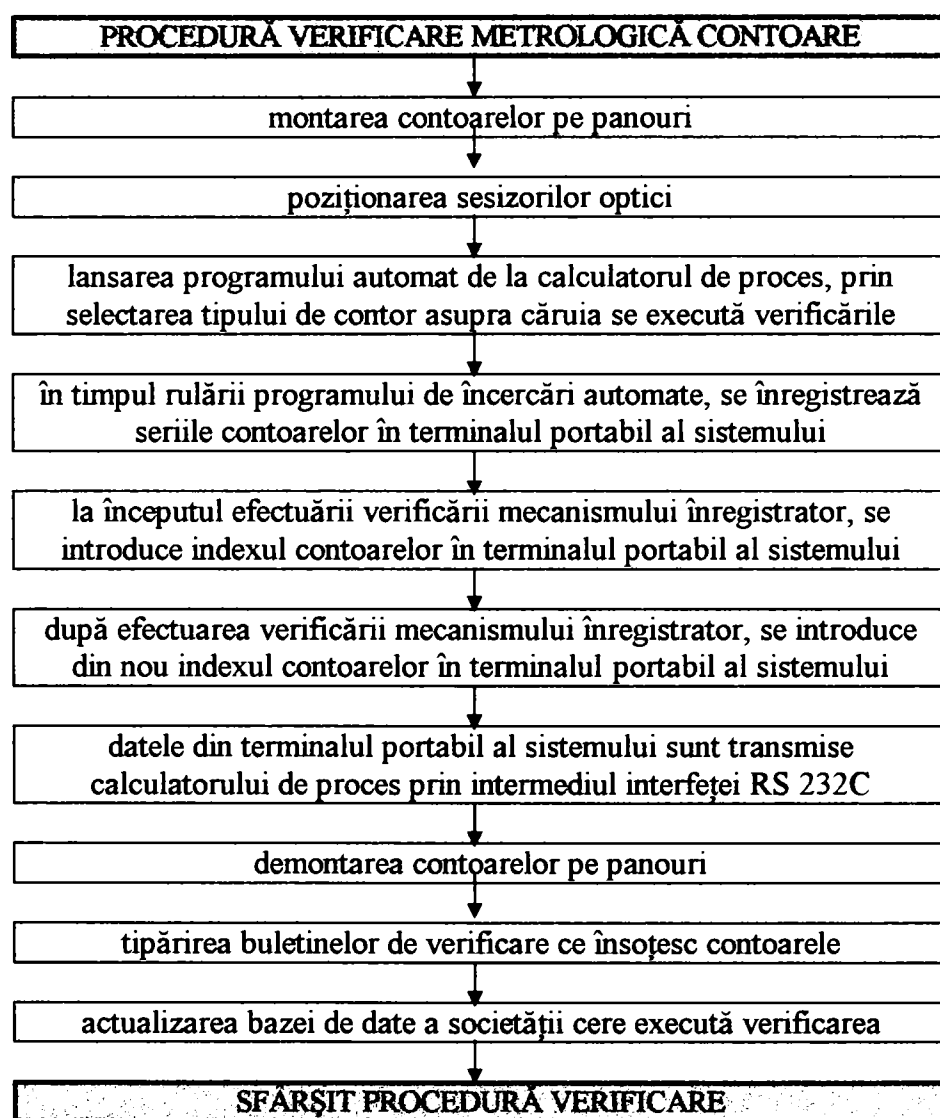


FIGURA 4.12

Sucesiunea operațiilor care trebuie efectuate pentru lucrul cu acest sistem este prezentată în ordinograma din figura 4.12.

În cazul utilizării instalațiilor cu autotransformatoare cu reglaj prin intermediul servomotoarelor al mărimilor prescrise dotate cu

contor electronic etalon, procedeul descris mai sus este ceva mai lent din cauza:

- necesității corecției simetriei sistemului trifazat din circuitul de tensiune după fiecare schimbare a valorii prescrise a factorului de putere;
- necesitatea aducerii la zero a curentului la fiecare schimbare a domeniului de curent al stației;
- timpi lungi de reglare a autotransformatoarelor și a decalorului de tensiune.

Fiabilitatea acestui sistem este mult mai redusă decât a sistemului cu generator electronic.

În cazul utilizării instalațiilor cu generator electronic dotate cu contor electronic etalon, procedeul descris mai sus este mai rapid fiind eliminate toate inconvenientele enumerate.

#### ***4.3.5 Implementarea programului pe calculatorul de proces în vederea etalonării contoarelor de energie electrică***

Pentru automatizarea completă a operației de verificare a contoarelor de energie electrică, calculatorul de proces trebuie să execute o succesiune de probe într-o ordine prestabilită și să înregistreze rezultatele fiecărei probe.

În cazul contoarelor active cu trei sisteme, cele mai răspândite de altfel, este necesară efectuarea următoarei succesiuni de operații:

1. selectarea conexiunii contoarelor;
2. selectarea factorului de putere;
3. reglarea tensiunii;
4. reglarea curentului;
5. efectuarea probelor de sensibilitate;
6. citirea rezultatelor;
7. reglarea curentului;
8. efectuarea probelor de mers în gol;
9. citirea rezultatelor;
10. reglarea curentului;
11. selectarea factorului de putere;

12. efectuarea verificării erorii admise;
13. citirea rezultatelor;
14. punctele 10 - 13 se repetă până la epuizarea tuturor valorilor impuse ale curentului și factorului de putere;
15. citirea manuală a indecșilor contoarelor;
16. reglarea curentului;
17. efectuarea verificării mecanismului înregistrator;
18. citirea manuală a indecșilor contoarelor;
19. tipărirea buletinelor de verificare;
20. actualizarea bazei de date a societății cere execută verificarea.

O particularitate a sistemului de calcul este modul de citire a rezultatelor în cazul probelor când nu se urmărește o valoare a erorii (valoare ușor de calculat în postul de calcul și afișare a erorii), ci rezultatul probei este de tip “admis” - “respins”. Astfel de probe sunt sensibilitatea și mersul în gol.

Operatorul intervine numai la montarea și demontarea contoarelor de pe panourile de etalonare (figura 4.13), la poziționarea



FIGURA 4.13

sesizoarelor optice și la probele a căror rezultate nu pot fi citite direct de sistemul de calcul (efectuarea verificării mecanismului înregistrator).

Sucesiunea operațiilor care trebuie efectuate pentru lucrul cu acest sistem sunt:

1. montarea contoarelor pe panouri;
2. poziționarea sesizorilor optici;

3. lansarea în execuție a programului automat de la calculatorul de proces, prin selectarea tipului de contor asupra căruia se execută verificările;
4. în timpul rulării programului de încercări automate, operatorul înregistrează seriile contoarelor în terminalul portabil al sistemului;
5. la începutul efectuării verificării mecanismului înregistrator, calculatorul de proces cere operatorului să introducă indexul contoarelor în terminalul portabil al sistemului;
6. după efectuarea verificării mecanismului înregistrator, calculatorul de proces cere operatorului să introducă din nou indexul contoarelor în terminalul portabil al sistemului (fiind ultima verificare indexul introdus este și indexul final cu care s-au încheiat verificările contorul, index menționat în buletinul de verificări);
7. datele din terminalul portabil al sistemului sunt transmise calculatorului de proces prin intermediul interfeței RS 232C;
8. demontarea contoarelor pe panouri;
9. tipărirea buletinelor de verificare ce însoțesc contoarele;
10. actualizarea bazei de date a societății care execută verificarea.

În cazul utilizării instalațiilor cu autotransformatoare cu reglaj prin intermediul servomotoarelor al mărimilor prescrise dotate cu contor electronic etalon, procedeul descris mai sus este ceva mai lent din cauza:

- necesității corecției simetriei sistemului trifazat din circuitul de tensiune după fiecare schimbare a valorii prescrise a factorului de putere;
- necesitatea aducerii la zero a curentului la fiecare schimbare a domeniului de curent al stației;
- timpi lungi de reglare a autotransformatoarelor și a decalorului de tensiune.

Fiabilitatea acestui sistem este mult mai redusă decât al sistemului cu generator electronic.

În cazul utilizării instalațiilor cu generator electronic dotate cu contor electronic etalon, procedeul descris mai sus este mai rapid fiind eliminate toate inconvenientele enumerate.

\*\*\*

La acest capitol contribuțiile autorului sunt legate de conceperea ideii de ansamblu asupra implementării echipamentelor cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii la un calculator compatibil IBM PC.

Am conceput și realizat:

- schema electrică și de automatizare a stației;
- schema electrică a blocului de comandă;
- schema electrică a blocului de măsură;
- schema electrică a sistemului ERRPRREX M pentru calculul și afișarea erorii contoarelor de energie electrică.

## **CAPITOLUL 5**

### **REALIZAREA INSTALAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ CU ACHIZIȚIA ERORII**



## ***5.1 Clasificarea instalațiilor de etalonat contoare de energie electrică***

### ***5.1.1 După criteriul constructiv:***

- a) instalații cu autotransformatoare cu reglaj manual al mărimilor prescrise (GANZ);
- b) instalații cu autotransformatoare cu reglaj prin intermediul servomotoarelor al mărimilor prescrise (METRA, ZERA, IRCOMON, SE04);
- c) instalații cu generator electronic (EDY).

### ***5.1.2 După metoda de lucru utilizată pentru etalonarea contoarelor de energie electrică:***

- a) instalații fără contor electronic etalon (specifice metodei watmetru - cronometru);
- b) instalații cu contor electronic etalon (specifice metodei contorului etalon);

Implementarea logicii programate se poate face pe instalații cu autotransformatoare cu reglaj prin intermediul servomotoarelor al mărimilor prescrise și pe instalații cu generator electronic.

## ***5.2 Sistem de calcul și afișarea erorii implementat pe instalațiile de etalonat și/sau verificat contoare de energie electrică***

În vederea automatizării procesului de verificare și/sau etalonare a contoarelor de energie electrică a fost realizat un sistem de calcul care operează împreună cu un calculator PC. Acest sistem, este destinat echipamentelor capabile să realizeze etalonarea sau verificarea unui număr de contoare de energie electrică determinat de posibilitățile instalației de etalonat.

### **5.2.1 Descrierea sistemului de calcul și afișarea erorii “ERRORREX M” din dotarea instalațiilor de etalonat contoare de energie electrică**

Sistemul ERRORREX M, este destinat dotării echipamentelor capabile să realizeze etalonarea și/sau verificarea contoarelor de energie electrică de clasă 0,1 sau chiar mai precise.

Datorită trecerii la o familie de procesoare mai evoluată și la afișajele cu cristale lichide în locul celor cu diode electroluminiscente, s-a obținut un consum mediu pe post mult mai redus față de versiunea existentă de ERRORREX, ceea ce a permis eliminarea necesității sectorizării alimentării posturilor de calcul și sesizoarelor, utilizarea unei singure surse de alimentare în comutație, care reușește să alimenteze un număr maxim de 250 de posturi.

Prin utilizarea senzorului HPJ - D21 produs de firma Honeywell, s-a obținut pe lângă reducerea consumului, o poziționare mai ușoară a în raport cu discul contorului, senzorul având suprafața de vizare mărită și reducând influența negativă a neuniformității (zgârieturilor) marginilor discului contorului în procesul de calcul a erorii.

Funcțiile interfeței de achiziție ale sistemului ERRORREX existent au fost preluate de procesoarele mai rapide din posturile de calcul, în așa fel încât noua interfață are gabaritul și consumul de energie mult redus.

Prin implementarea unui protocol de comunicare special pe interfața RS 232C [12], [48], atât între calculatorul sistemului și posturile de calcul cât și între posturile de calcul, s-a reușit eliminarea totală, transparent din punct de vedere al utilizatorului, e erorilor de transmisie. În cazul apariției unei erori de transmisie la orice nivel, protocolul este astfel conceput încât reia transfereul de date până la eliminarea erorii.

*Terminalul portabil* atașat sistemului [13], figura 5.1 facilitează introducerea datelor specifice fiecărui contor (constantă, serie, index de început și sfârșit la probele de mecanism înregistrator, index de ieșire, admis/respins la probele de sensibilitate și mers în gol, numărul de inventar, etc). Introducerea datelor se face de operator la terminalul portabil după care, acesta se conectează cu calculatorul PC, căruia îi transmite aceste date.

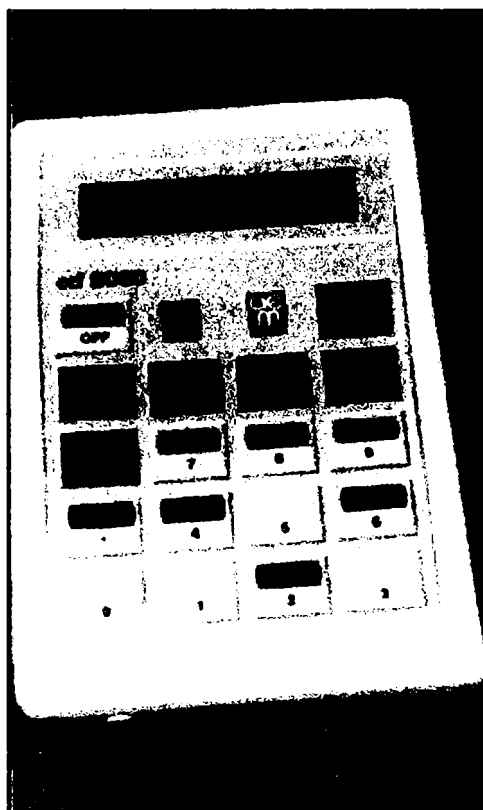


FIGURA 5.1

Conceput într-o construcție modulară, sistemul se poate realiza pentru un număr de maximum 250 de contoare.

### 5.2.2 Principiul de funcționare:

Eroarea se calculează pentru fiecare rotație a discului contorului de etalonat, la fiecare post de calcul și se afișează individual, prin modulele asociate fiecărui contor.

Pentru determinarea erorii se utilizează metoda "contorului etalon", ca element etalon utilizându-se un convertor putere - frecvență electronic de precizie corespunzătoare.

Calculul erorii se face pe baza formulei:

$$E_{\%} = \frac{N_{m\acute{a}s} - N_{calc}}{N_{m\acute{a}s}} \times 100 \quad [\%] \quad (5.1)$$

unde,  $N_{m\acute{a}s}$  reprezintă numărul de impulsuri ale contorului etalon măsurate între două ture de disc ale contorului de etalon, iar  $N_{calc}$  reprezintă numărul de impulsuri calculat a contorului etalon, în aceeași fereastră, pentru care eroarea contorului de etalonat este zero.

Posturile de calcul sunt interschimbabile, sistemul de calcul având posibilitatea de a atribui fiecărui post de calcul o adresă la care răspunde. Comenzile pot fi generate pentru toate posturile simultan, sau individual pentru fiecare post în parte. Datorită acestei facilități, toate modurile de lucru ale postului de calcul pot fi programate specific pentru fiecare post.

Fiecare post de calcul poate etalona și/sau verifica contoare cu constantă diferită, calculul erorii făcându-se pentru un număr programabil de ture ale discului contorului - între 1 și 255. De asemenea, există posibilitatea de a introduce un filtru soft programabil pentru eliminarea influenței negative a zgârieturilor de pe discul contorului.

Semnalul de marcă de pe discul contorului (sau ieșirea de impulsuri a contoarelor electronice) este detectat de sesizorul de marcă atașat fiecărui contor și transmis la postul de calcul. Sistemul are în dotare sesizoare cu vizare și prin geamul contorului, sau opțional alte tipuri de sesizoare adaptate la tipul de contor ce urmează a fi etalonat și/sau verificat.

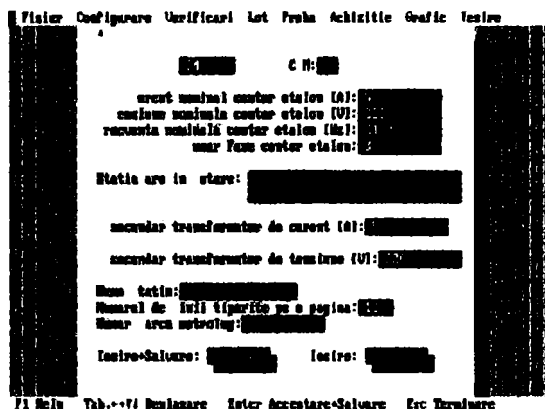
Postul de calcul este realizat pe baza unui microcontroler de opt biți [7], [8], [14], [43], [46], [57], [64] are afișaj cu cristale lichide de 3 1/2 digiți, 1/2" înălțime, ceea ce permite citirea erorii de pe afișaj și în prezența luminii ambiante puternice. El calculează și afișează eroarea contorului, valoarea erorii poate fi transmisă la comandă la calculator, pe linia de comunicație serială, de tip RS 232C.

Interfața de achiziție măsoară și multiplică sau divide semnalul de frecvență etalon generat de convertorul putere - frecvență, asigurând precizia de calcul a erorii contoarelor de energie.

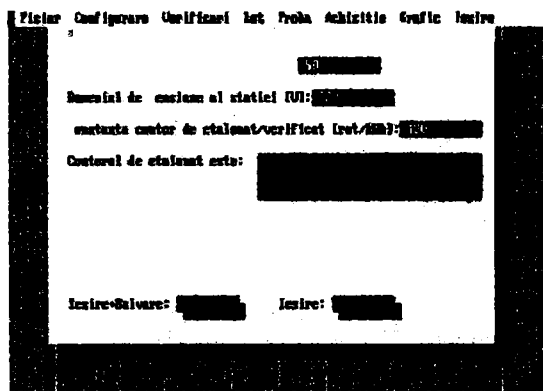
Programarea probelor de lucru, introducerea constantei contorului, transmiterea comenzii de achiziție a erorii contoarelor și a celorlalte comenzi, se face foarte ușor, datorită realizării unui program conversațional pe calculatorul compatibil PC care introduce operatorul într-un meniu în care îl conduce pas cu pas, arătându-i operațiile pe care trebuie să le execute. Programul este oferit în versiunea sub Windows. Câteva din meniurile de lucru ale programului sunt prezentate în figura 5.2 [37], [66].



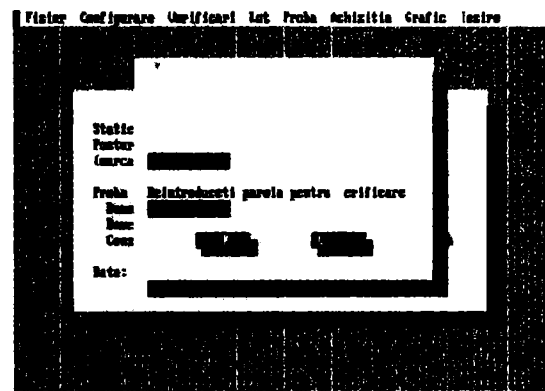
Ecran de intrare



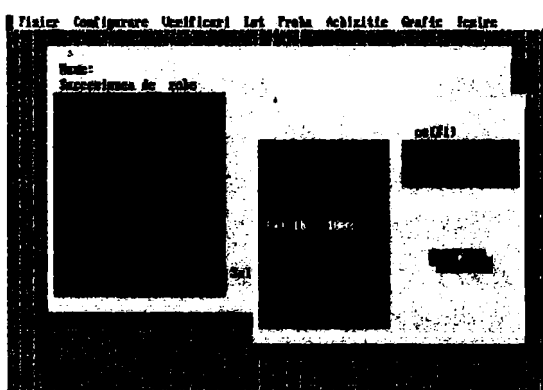
Meniu de configurare



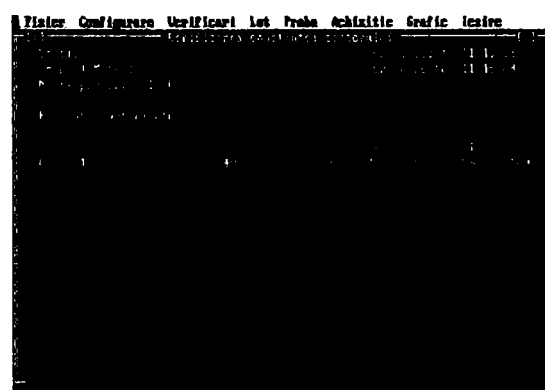
Meniu probă curentă



Activarea parolei



Succesiune de probe



Verificare constantă contor

FIGURA 5.2

Erorile achiziționate se pot protocola, la cerere, la imprimanta sistemului.

Stocarea erorilor pe calculator permite de asemenea formarea unei bănci de date pentru diverse prelucrări statistice.

### **5.2.3. Descriere constructivă și funcțională**

#### **5.2.3.1 Postul de calcul și afișarea erorii și interfața de achiziție la nivel de schemă bloc**

Sistemele numerice de calcul și afișare a erorii contoarelor de energie electrică utilizate înainte de realizarea sistemului ERRORREX M, prezintă patru mari dezavantaje descrise în continuare:

- nu pot etalona și/sau verifica orice tip de contor electric din cauză că nu se pot prescrie decât un număr limitat de constante ale contoarelor de etalonat/verificat;
- nu permit achiziția erorii determinate de blocurile de calcul la un calculator, pentru obținerea imediată a buletinelor de verificare;
- nu pot etalona și/sau verifica contoare de energie electrică cu clasă de precizie mai mică de unu, datorită metodei de determinare a erorii de măsură a contoarelor;
- datorită soluției tehnice adoptate, funcționează cu consum mare de energie electrică.

Prin realizarea sistemului ERRORREX M s-au înlăturat aceste dezavantaje, în plus se pot enumera următoarele avantaje obținute:

- sistemul permite etalonarea și/sau verificarea oricărui tip de contor, indiferent de constantă și de clasa de precizie, care poate fi mai mare de 0,1%;
- sistemul realizează achiziția erorilor determinate la un calculator PC, ceea ce permite creerea de buletine de verificare și creerea unei baze de date despre loturile de contoare etalonate/verificate care pot fi utilizate în calcule statistice de producătorul sau utilizatorul de contoare;
- datorită componentelor utilizate, sistemul funcționează cu consum redus de energie și asigură buna citire a erorii și la posturile de calcul care sunt amplasate în încăperi cu afișajul orientat spre o sursă puternică de lumină, de exemplu fereastra.

În figura 5.3 s-a prezentat o schemă bloc a unei instalații de etalonat/verificat contoare, unde cu linie întreruptă sunt marcate următoarele componente ale instalației:  $q$  contoare de etalonat/verificat  $Wh_i$ ,  $i=1\dots q$ , alimentate de la o stație de forță care

generează curent și tensiune în domenii bine definite SE, unu sau trei convertoare putere-frecvență de precizie CPF1, CPF2, CPF3, cu rolul de a genera un semnal de referință etalon, q sesizoare optoelectronice  $SO_i$ ,  $i=1 \dots q$ , cu rolul de a genera un impuls la fiecare trecere a mărcii de pe discul contorului de etalonat/verificat printr-o poziție fixă, un modul de alimentare MA, care are rolul de a genera tensiunea de alimentare a interfeței de achiziție, a sesizoarelor optoelectronice și a posturilor de calcul, un calculator PC cu imprimantă I, având rolul de a asigura prin pachetul de programe implementat pe PC transmiterea și execuția comenzilor pentru modulele de lucru ale sistemului, respectiv de a executa tipărirea buletinelor de verificări ale contoarelor verificate.

La nivelul schemei bloc, partea de noutate a sistemului, constă dintr-un număr de q posturi de calcul și afișare a erorii  $PC_i$ , unde  $i=1 \dots q$ , care funcționează total independent, în timp real, având posibilitatea de a calcula și afișa eroarea contoarelor asociate, determinată la un număr programabil de rotații complete ale discului contorului și o interfață de achiziție IA.

Fiecare post de calcul  $PC_i$  are câte trei intrări de semnal: SI, M, FeR (sau FeS, sau FeT) și două ieșiri de semnal: LD, SO. La intrarea SI a primului post de calcul se recepționează comenzile transmise de la un calculator compatibil PC prin interfața de achiziție IA, ieșirea SO, comenzi care se transmit la posturile următoare, prin ieșirea SO a postului de calcul conectată la intrarea SI a postului următor. La fiecare intrare M a posturilor de calcul se aplică un impuls separat, generat la fiecare rotație completă a discului contorului de sesizorul optoelectronic  $SO_i$  asociat postului  $PC_i$ . La intrările FeR, respectiv FeS sau FeT ale posturilor de calcul, se aplică un tren de impulsuri de la ieșirea FeR, respectiv FeS sau FeT, după caz, a interfeței de achiziție. Ieșirea SO a postului de calcul  $PC_i$  se conectează la intrarea SI a postului de calcul următor,  $PC_{i+1}$ . Ieșirea LD a tuturor posturilor de calcul este o linie comună care reprezintă o intrare pentru interfața de achiziție IA, linie pe care posturile de calcul comunică spre un calculator PC datele solicitate prin comenzile asociate modulelor de lucru.

Interfața de achiziție IA are rolul de a genera un semnal la ieșirea Fe, prin prelucrarea semnalului FinR (sau semnalelor FinR, FinS, FinT, funcție de modul de generare a frecvenței etalon, pe una sau trei faze separate) generat de convertorul putere-frecvență CPF1 (respectiv convertoarele CPF1, CPF2, CPF3) și prelucrat (multiplicat-demultiplicat) de interfața de achiziție IA. Interfața de

achiziție are cinci intrări: FinR, FinS, FinT, LD și Rx și șase ieșiri: Tx, FeR, FeS, FeT, SO. La intrările FinR, FinS, FinT se aplică impulsurile generate de convertoarele putere-frecvență  $CPF_i$ , cu  $i=1...3$ . La intrarea Rx, conectată cu ieșirea serială a calculatorului PC, sunt transmise cuvintele de comandă de la calculatorul PC către posturile de calcul  $PC_i$  și interfața de achiziție IA. La intrarea LD sunt recepționate informațiile transmise de posturile de calcul  $PC_i$  ca răspuns la comenzile primite, pentru a fi transmise mai departe spre calculatorul PC. La ieșirea Tx, conectată la intrarea serială a calculatorului PC, sunt transmise spre calculatorul PC răspunsurile corespunzătoare cuvintelor de comandă recepționate de posturile de calcul și de interfața de achiziție. La ieșirea Fe este generat un tren de impulsuri etalon către posturile de calcul  $PC_i$ . La ieșirea SO sunt transmise cuvintele de comandă recepționate de la calculatorul PC către posturile de calcul  $PC_i$ .

Funcționarea sistemului de calcul prezentată la nivelul schemei bloc este descrisă în cele ce urmează.

Modurile de lucru posibile pentru sistem sunt facilitate de un pachet de programe implementat pe calculatorul PC, în posturile de calcul și în interfața de achiziție. Datorită acestor programe și datorită configurației sistemului, acesta poate executa următoarele funcțiuni: învățare post, programare constantă, programare număr de rotații complete disc contor, autotest sistem, achiziția erorii de măsură a contoarelor pentru postul de calcul și programarea constantelor de multiplicare-divizare a frecvenței etalon, programarea modului de generare a frecvenței etalon către posturile de calcul și măsurarea energiei electrice consumată într-o anumită perioadă de timp pentru interfața de achiziție.

În modul de lucru *învățare post*, posturile sunt programate cu un anumit număr, începând de la 1 până la max. 255, numerele fiind consecutive. Fiecare post răspunde către calculatorul PC cu numărul învățat, astfel sistemul își determină singur numărul total de posturi și poate adresa în continuare oricare dintre posturi separat.

În modul de lucru *programare constantă* în fiecare post de calcul se programează o valoare determinată, diferită, funcție de constanta contorului de etalonat/verificat, de raportul de transformare al transformatoarelor etalon de tensiune și curent care intră în componența stației SE, în funcție de tipul contorului de etalonat/verificat (monofazat/trifazat), astfel încât să se obțină, în condiții de eroare zero a contoarelor față de CPF, un număr  $n$  (formula 5.1) de impulsuri.



În modul de lucru *programare număr de rotații complete de disc contor*, se poate programa numărul de ture la care se face calculul erorii contoarelor.

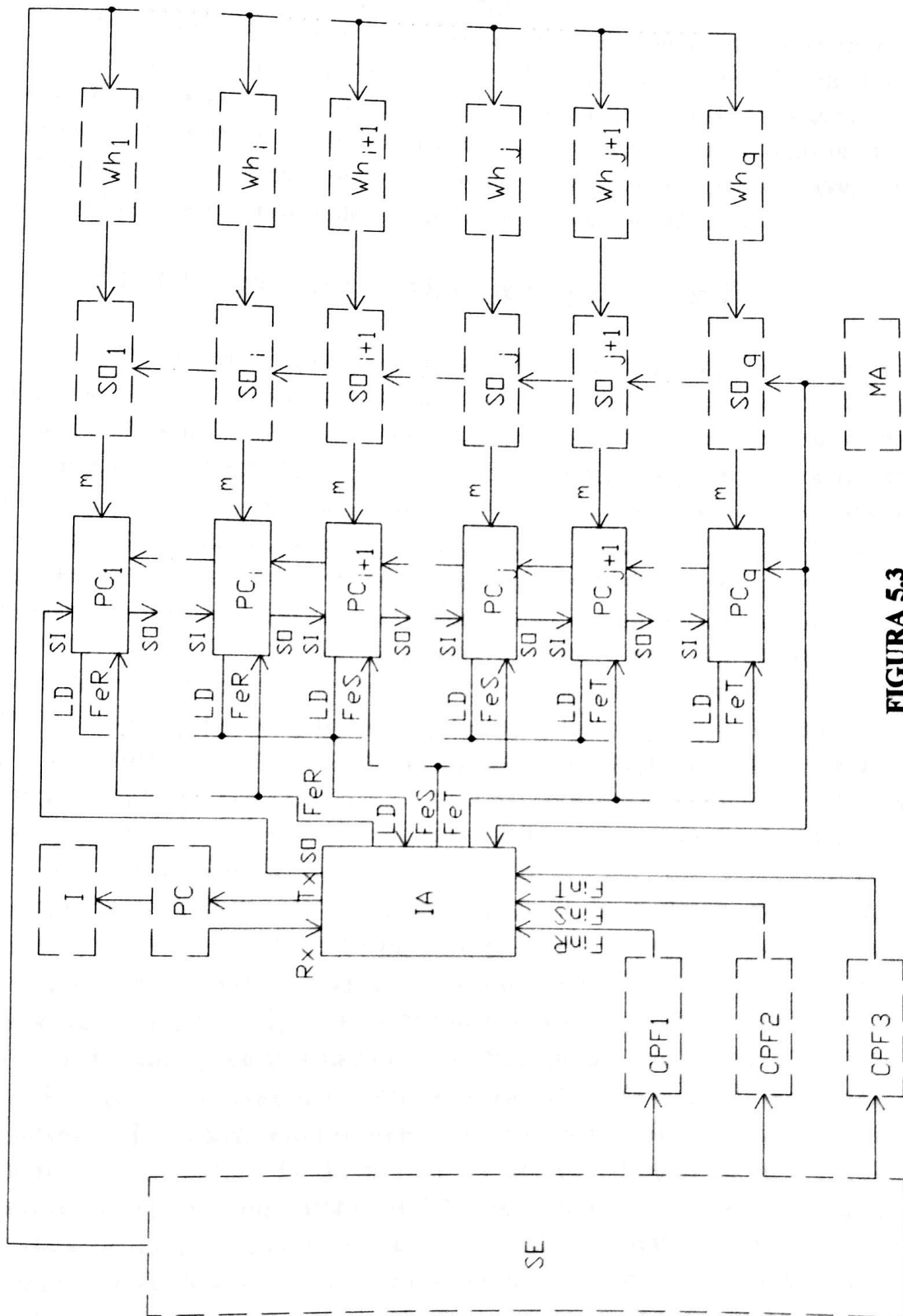


FIGURA 5.3

Modul de lucru *autotest sistem* are rolul de a declanșa afișarea unei secvențe cunoscute pe afișajul cu cristale lichide al postului de calcul. Urmărind succesiunea derulată pe afișaj se poate diagnostica corecta funcționare a postului de calcul.

În modul de lucru *achiziție eroare de măsură*, la comanda generată de la calculatorul PC, acesta va recepționa de la toate posturile de calcul sau numai de la cel solicitat, valoarea erorii de măsură determinată pentru contorul testat. Informațiile recepționate în diferite probe se constituie în baze de date utilizate pentru generarea automată a unor buletine de verificări ale contoarelor testate.

### **5.2.3.1.1 Postul de calcul și afișarea erorii**

Am conceput postul de calcul și afișare a erorii PC<sub>i</sub> care are aceeași schemă pentru fiecare post [1], [6], [34], schemă prezentată în figura 5.4, care constă într-un microcontroler M, în memoria căruia s-a implementat un set de programe care comandă și gestionează toate modurile de lucru ale postului, microcontroler prevăzut cu un circuit de reset care acționează la intrarea RST, format din rezistorul R1 și condensatorul C1, un oscilator cu un rezonator de cuarț RC și condensatoarele C2 și C3 care generează tactul microcontrolerului la intrările XTAL1 și XTAL2. Microcontrolerul este prevăzut cu zece linii de port: P1.0, P1.1, P1.2, P1.3, P1.4, P1.5, P1.6, P2.0, P2.1 și P2.2, cu o intrare de întrerupere externă INT0, o intrare de timer T0 și cu posibilități de comunicare serială: intrarea RxD și ieșirea TxD. Un circuit amplificator-convertor cu șase porți BC<sub>i</sub>, cu i=1...6, este prevăzut cu rol de convertor de nivel pentru circuitele BC1, BC2, BC3 a căror intrări sunt conectate prin rezistorii R8, R6, respectiv R7 la tensiunea de +5V și cu rol de amplificator BC4, BC5, BC6. Intrarea circuitului BC1 este linia SI pe care posturile de calcul comunică între ele și cu interfața de achiziție IA, intrarea circuitului BC2 este linia M, pe care postul de calcul primește impuls de la sesizorul optoelectronic SO<sub>i</sub> la fiecare trecere a mărcii prin dreptul său, iar intrarea circuitului BC3 este linia comună tuturor posturilor de calcul, pe care se transmite frecvența etalon FeR sau FeS sau FeT. Ieșirea circuitului BC6 este conectată la un montaj cu ieșire colector în gol, prevăzut pentru a realiza linia comună LD între cele q posturi de calcul și realizat dintr-un rezistor R1 care are un terminal conectat la ieșirea circuitului BC6 și celălalt terminal conectat în baza unui tranzistor T1 și la un terminal al unui rezistor R2 care are al doilea terminal conectat la tensiunea de +5V împreună cu un terminal al unui rezistor

R4 care are al doilea terminal conectat în emitorul tranzistorului T1. În colectorul lui T1 este conectată baza unui tranzistor T2 și un terminal al unui rezistor R3 care are al doilea terminal conectat împreună cu emitorul tranzistorului T2 la masa sistemului. Ieșirea LD este colectorul tranzistorului T2. Un circuit multiplexor-demultiplexor CH este prevăzut pentru realizarea celor două tipuri de conexiuni pentru posturile de calcul: în paralel și în cascadă. În funcție de nivelul aplicat la intrările de comandă A și B legate împreună la ieșirea microcontrolerului P2.1 prin amplificatorul BC4, este realizată alocarea semnalelor ax, ay și bx, by funcție de semnalele "ax sau ay", respectiv "bx sau by". Intrarea de comandă C, conectată la ieșirea P2.2 a microcontrolerului M, determină alocarea semnalului "cx sau cy" funcție de semnalele ce se aplică la intrările cx, cy. Un circuit driver DVR pentru un afișaj cu cristale lichide LCD este prevăzut pentru comanda afișajului. Datele de afișat sunt memorate în DRV pe liniile de intrare B0, B1, B2, B3 prin multiplexarea celor trei cifre jumătate pe liniile BP1, BP2, selecția circuitului făcându-se pe linia CS, toate aceste linii fiind conectate la portul P1.X, cu  $X=0\dots6$  al microcontrolerului M care realizează comanda driver-ului. Un afișaj cu cristale lichide LCD este prevăzut pentru afișarea erorii de măsură calculată de postul de calcul sau a unor informații suplimentare. Datele de intrare pentru LCD sunt furnizate de circuitul DRV, pe liniile comune: a1, b1, c1, d1, e1, f1, g1 pentru prima cifră, a2, b2, c2, d2, e2, f2, g2 pentru a doua cifră, a3, b3, c3, d3, e3, f3, g3 pentru a treia cifră, x, y, z, k pentru unitate-semn.

Funcționarea unui post de calcul  $PC_i$  este prezentată în continuare.

Postul de calcul funcționează corect dacă își cunoaște numărul  $i$ ,  $i=1\dots q$ , valoarea constantei  $n_0$  - vezi formula (5.1) și numărul de rotații după care trebuie să calculeze eroarea. La comanda de învățare post el răspunde prin afișarea pe afișajul cu cristale lichide LCD a mesajului "P xx", unde xx reprezintă numărul postului. Dacă acesta este mai mare decât 99, se va afișa alternativ "P" și "xxx", unde xxx este numărul postului, fiecare informație fiind menținută pe afișaj aproximativ 0,5s. De asemenea postul de calcul va trimite ca răspuns calculatorului PC numărul de post recepționat și memorat. La comanda de programare a numărului de rotații complete ale discului contorului după care se va face calculul erorii, pe afișaj nu apare nici un mesaj, dar spre calculatorul PC se transmite numărul recepționat și memorat. La comanda de programare constantă, pe afișajul postului de calcul apare mesajul "- - -", iar spre calculatorul PC se transmite

valoarea recepționată și memorată. După ce aceste valori au fost memorate prin recepționarea și decodificarea cuvintelor de comandă pe intrarea SI, care prin amplificatorul BC1 ajunge la intrarea serială RxD a microcontrolerului M, postul de calcul este pregătit să efectueze calculul erorii.

$$\varepsilon [ \% ] = \frac{n_0 - n}{n} \times 100 \quad (5.2)$$

La intrarea Fe se aplică un tren de impulsuri de frecvență etalon care sunt numărate în timer-ul T0 al microcontrolerului M. Fiecare impuls detectat incrementează un timer pe doi octeți T0, astfel încât dacă acesta ajunge să numere peste 65 536 s-a detectat situația de lipsă a semnalului de la contorul de etalonat/verificat aplicat la intrarea M, situație marcată pe afișajul LCD al postului de calcul prin afișarea mesajul “-LO”. În caz contrar, semnalul de marcă M determină o fereastră de măsură, funcție de numărul de rotații complete ale discului contorului programate, fereastră în care se numără un anumit număr de impulsuri de frecvență Fe, adică se obține numărul n - vezi formula (5.2). Dacă între două impulsuri venite la intrarea M numărul contorizat de timer-ul T0 este zero, atunci se sesizează situația de lipsă frecvență etalon, situație care se marchează prin afișarea mesajului “-PE”. Calculul erorii se face după formula (5.2), unde  $n_0$  este un număr programat în postul de calcul, în urma executării comenzii programare constantă. Transmiterea cuvântului de comandă de la calculatorul PC se face pe linia SI, cuvântul fiind preluat și prelucrat de circuitul serial al microcontrolerului M, intrarea RxD. În mod normal, după efectuarea reset-ului, posturile sunt conectate în paralel, ieșirea SO a postului de calcul este conectată la intrarea SI datorită faptului că ieșirea P2.2 a microcontrolerului M este programată la valoarea de potențial “0”, adică “bx sau by” este “0”, ceea ce înseamnă că “cx sau cy” este conectat la cx, adică la intrarea SI a postului de calcul, prin intermediul circuitului BC1. În această situație, ieșirea microcontrolerului P2.1 este programată la valoarea de potențial “0”, adică intrarea A a circuitului CH conectată la P2.1 prin amplificatorul BC4 este “0”, ceea ce înseamnă că semnalul “ax sau ay” este transmis la ieșirea ax a circuitului CH, respectiv la intrarea amplificatorului BC6, care prin montajul cu ieșire colector în gol va pune linia LD la valoarea potențialului de +5V. Montajul cu ieșire colector în gol are rol de amplificator de linie și face funcția de SAU

cablat pe linia LD între cele  $q$  posturi de calcul. Funcționarea montajului este următoarea: rezistoarele R1, R2 formează un divizor rezistiv care atacă tranzistorul T1 cu rol de generator de curent în baza tranzistorului T2. Valoarea curentului este dată de rezistorul R4. Rezistorul R3 preia curentul rezidual al tranzistorului T1. Conectarea posturilor de calcul în paralel permite accesul rapid al cuvintelor de comandă la fiecare post. În situația în care se transmit cuvinte de comandă numai pentru un anumit post, postul în cauză își recunoaște numărul, recepționează comanda și răspunde spre calculator din starea de funcționare numită transfer, după cum urmează: ieșirea P2.1 este programată la valoarea de potențial "1", ceea ce înseamnă că intrările de comandă A și B ale circuitului CH au valoarea "1", adică ieșirea microcontrolerului TxD este conectată prin circuitul BC1, intrarea "bx sau by", ieșirea by, circuitul BC6, montajul cu colector în gol, la linia LD. În această stare ieșirea P2.2 este programată cu valoarea de potențial "1", ceea ce înseamnă că intrarea de comandă C a circuitului CH este "1", adică valoarea de potențial de la intrarea "ax sau ay", +5V este transmisă la ieșirea SO a postului de calcul prin ay, cy, "cx sau cy", circuitul BC5. După efectuarea transmisiei către calculatorul PC, postul de calcul revine în modul de conectare paralel, prin reprogramarea ieșirilor P2.1, P2.2 la valoarea "1". În cazul în care postul de calcul recunoaște un cuvânt de comandă care a fost transmis pentru toate posturile de calcul, el trece din modul de conectare în paralel în modul de conectare în cascadă, după cum urmează: ieșirea P2.1 rămâne programată cu valoarea "0", ceea ce înseamnă că linia LD rămâne la valoarea de potențial de +5V, iar ieșirea P2.2 este programată cu valoarea de potențial "1", adică intrarea "ax sau ay" este "1" și intrarea de comandă C a circuitului CH este "1", ceea ce înseamnă că ieșirea TxD a microcontrolerului M este conectată la ieșirea SO a postului de calcul prin: "bx sau by", bx, cy, "cx sau cy", circuitul BC5, ceea ce echivalează cu desfacerea legăturii dintre SO și SI caracteristică modului de conectare în paralel a posturilor de calcul. El rămâne în această stare cât timp recepționează cuvântul de comandă, după care trece în starea de funcționare transfer, descrisă mai sus, unde trimite răspuns către calculatorul PC. După aceea, prin reprogramarea ieșirilor P2.1 și P2.2 postul de calcul revine în modul de conectare în cascadă, când execută transmiterea cuvântului de comandă recepționat la postul de calcul următor și prin extensie, la toate posturile de calcul. După ce a executat transferul către postul următor, tot prin reprogramarea ieșirilor P2.1 și P2.2, postul de calcul revine în modul de conectare în paralel. Același lucru se întâmplă în

situația în care postul de calcul nu recepționează succesiunea așteptată la cuvântul de comandă într-un interval de timp definit.

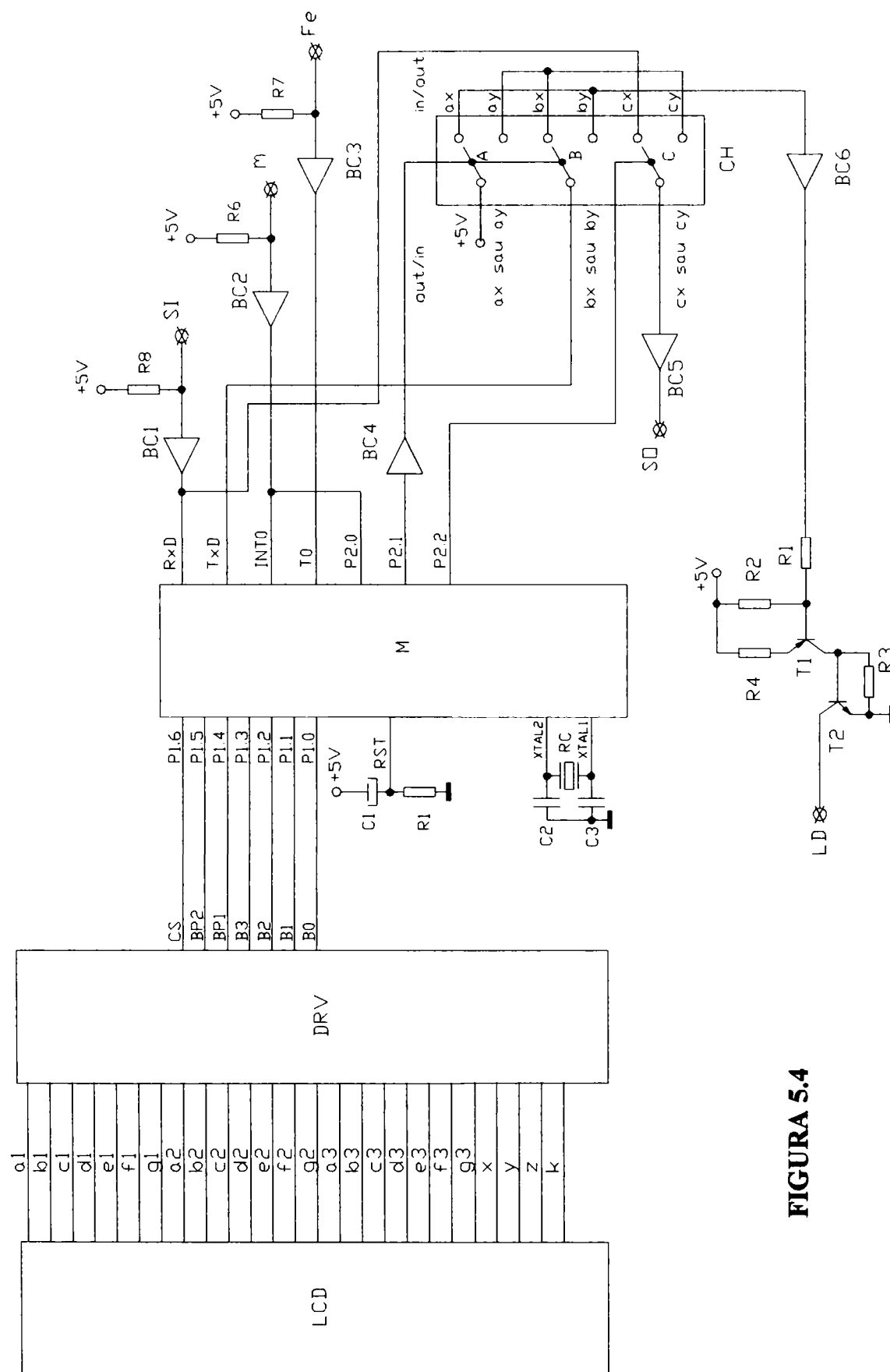


FIGURA 5.4

### 5.2.3.1.2 Interfața de achiziție

Am conceput interfața de achiziție, a cărei schemă este prezentată în figura 5.5, este alcătuită din cinci microcontrolere: M1, M2, M3, M4, M5 în memoriile cărora sunt implementate pachete de programe distincte cu care sunt stabilite următoarele moduri de lucru: programare constantă de multiplicare-divizare a semnalului de frecvență etalon, măsurarea energiei consumate într-un anumit interval de timp și programarea tipului de convertor putere-frecvență utilizat [33], [35], [47], [53]. Cele trei pachete de programe sunt distribuite după cum urmează: primul în microcontrolerul M1, al doilea în microcontrolerul M2 și al treilea în microcontrolerelor M3, M4, M5, care sunt înscrise cu programe identice. Un microcontroler M1 este prevăzut pentru a asigura legătura interfeței de achiziție cu un calculator PC prin intrarea RxD, respectiv ieșirea TxD. Comenzile sunt transmise de la calculatorul PC la intrarea Rx a interfeței izolată electric prin circuitul IZ1, la posturile de calcul PCi, prin circuitul amplificator-convertor BC11, la ieșirea SO, respectiv la interfața de achiziție IA prin circuitul amplificator-convertor BC10, la intrarea RxD a microcontrolerului. Răspunsurile se transmit la calculatorul PC, după recepționarea cuvintelor de comandă, la ieșirea Tx, prin circuitul de izolare electrică IZ2, de la ieșirea unei porți SAU. O intrare a porții este de la interfața de achiziție, ieșirea TxD a microcontrolerului M1, iar a doua intrare este legată la linia LD pe care sunt transmise răspunsurile de la posturile de calcul PCi. Microcontrolerul M1 are intrarea de reset RST conectată la un circuit de reset format din rezistorul R1 conectat la masă și condensatorul C3 conectat la +5V. La intrările XTAL1 și XTAL2 ale microcontrolerului M1 se aplică un circuit de tact format din rezonatorul de cuarț K și condensatoarele C1 și C2. Microcontrolerul este prevăzut cu trei linii de port P1.0, P1.1 și P1.2 conectate la aceleași linii ale microcontrolerelor M3, M4 și M5 cu rol de transmitere a constantei de multiplicare a frecvenței etalon și cu trei linii de port P1.4, P1.5, P1.6 cu rol de a transmite la intrările corespunzătoare A, B, C ale unor circuite multiplexor-demultiplexor AMD1, AMD2 și AMD3 constanta de divizare a frecvenței etalon aplicată la intrările FinR, FinS, FinT ale circuitelor amplificator-convertor BC7, BC8 și BC9. Trei grupe de circuite

formate din circuitul de divizare DIV1 și circuitul de multiplicare-demultiplicare AMD1, respectiv DIV2, AMD2 și DIV3, AMD3 sunt prevăzute pentru a realiza divizarea semnalului de frecvență etalon aplicat corepunzător la intrările FinR, FinS și FinT. Cele trei grupe de circuite sunt realizate identic și anume: ieșirile Q1, Q2, Q3, Q4, Q5, Q6 și Q7 ale circuitului DIV<sub>i</sub>, cu  $i=1...3$ , reprezentând trepte de divizare ale semnalului de frecvență aplicată la intrarea  $\phi_i$ , cu  $i=1...3$  a circuitului DIV<sub>i</sub>, respectiv intrarea 0 a circuitului AMD<sub>i</sub>, cu  $i=1...3$ , sunt conectate la intrările 1, 2, 3, 4, 5, 6, 7 ale circuitului AMD<sub>i</sub>. La ieșirea OUT<sub>i</sub> a circuitului AMD<sub>i</sub> se obține, funcție de constanta de divizare de la intrările A, B, C, o frecvență divizată care se aplică la intrarea de întrerupere externă INT0 a microcontrolerelor M3, respectiv M4 și M5 pentru operația de multiplicare. Frecvența multiplicată, în funcție de constanta de multiplicare, este obținută la ieșirile P2.0 ale microcontrolerelor M3, M4, M5. Microcontrolerele M3, M4, M5 sunt prevăzute cu rol de multiplicare a frecvenței etalon. Ele au același circuit de reset, aplicat la intrările RST, ca și M1, dar funcționează cu tact extern, aplicat la intrările XTAL1. Ieșirile P2.0 ale microcontrolerelor M3, M4 și M5 sunt conectate la intrările de întrerupere externă INT0, INT1, respectiv la intrarea de timer T0 ale unui microcontroler M2, prevăzut cu rol de a efectua însumarea celor trei frecvențe etalon, frecvența rezultată fiind obținută la ieșirea P2.3 a microcontrolerului M2. Intrarea de reset RST și cea de tact XTAL1 a microcontrolerului M2 sunt conectate cu intrările similare ale microcontrolerelor M3, M4 și M5. Ieșirea P2.3 a microcontrolerului M2 este conectată cu un circuit multiplexor-demultiplexor AMD4, prevăzut cu rol de a genera la ieșirile "ax sau ay", prin circuitul amplificator-convertoare BC12, "bx sau by", prin circuitul amplificator-convertoare BC13 și "cx sau cy", prin circuitul amplificator-convertoare BC14 semnale de frecvență etalon FeR, FeS, FeT. La intrările ay, by, cy se aplică semnalul de la ieșirile P2.0 ale microcontrolerelor M3, M4, M5, iar la intrările ax, bx, cx se aplică semnalul de la ieșirea P2.3 a microcontrolerului M2, astfel încât funcție de semnalul aplicat la intrările A, B, C ale AMD4, comandate de ieșirea P1.3 a microcontrolerului M1, la ieșirile "ax sau ay", "bx sau by", "cx sau cy" se obțin fie la toate același semnal (frecvența însumată), fie semnale diferite, frecvențele etalon pe cele trei faze FeR, FeS, FeT. Microcontrolerul M2 are opt linii de port comune cu microcontrolerul M1: P0.0, P0.1, P0.2, P0.3, P0.4, P0.5, P0.6, P0.7, prevăzute pentru a se transmite pe ele, de la M2 către M1, valoarea de energie măsurată pe fiecare fază R, S și T, la comanda



dată de M1 pe linia de port P2.3 legată la intrarea de timer T1 a microcontrolerului M2. Liniile de port comune P2.6 și P2.7 celor două microcontrolere M1 și M2 sunt prevăzute pentru a comanda transferul informației de energie măsurată în nouă pași, informația de transmis fiind de câte trei octeți pe fază.

Funcționarea interfeței de achiziție este prezentată în

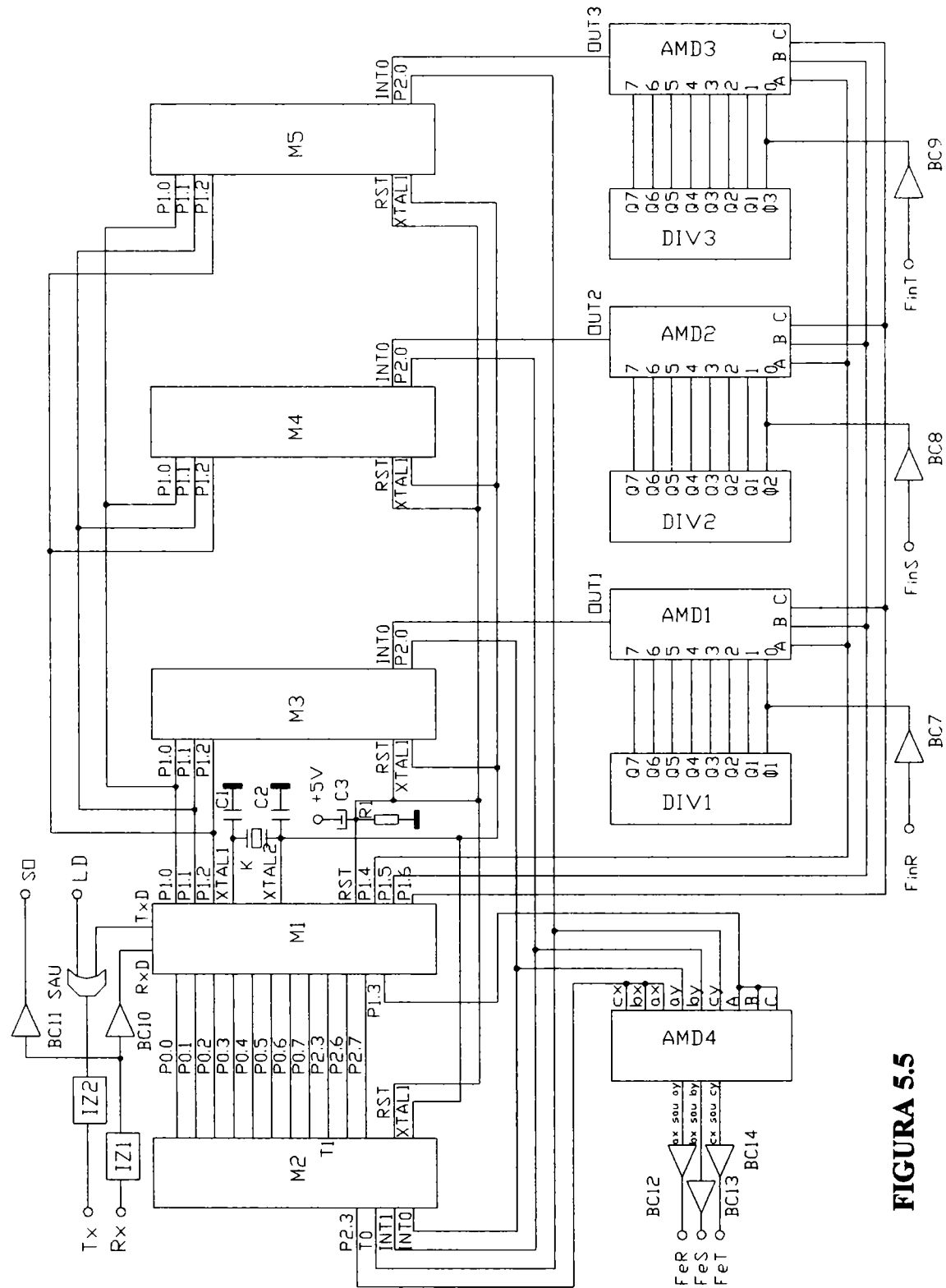


FIGURA 5.5

continuare.

Interfața de achiziție trebuie să fie programată de la un calculator PC cu constanta de multiplicare-divizare, care se transmite sub forma unui cuvânt de comandă la intrarea RxD a microcontrolerului M1 prin intrarea Rx și circuitul amplificator-convertoare BC10. După ce cuvântul de comandă este decodificat de interfață, aceasta memorează valoarea recepționată și transmite mesaj de răspuns spre calculatorul PC la ieșirea Tx prin ieșirea microcontrolerului TxD, poarta SAU și circuitul de izolare electrică IZ2. Valoarea constantei de multiplicare-divizare recepționată este separată în constanta de multiplicare care se transmite la microcontrolerul M3, M4 și M5 pe liniile comune P1.0, P1.1, P1.2 și constanta de divizare, care se transmite pe liniile P1.4, P1.5, P1.6 la intrările A, B, C ale circuitelor AMD1, AMD2, AMD3. Frecvențele etalon de intrare FeR, FeS, FsT se aplică prin circuitele BC7, BC8, BC9 la intrările corespunzătoare  $\phi 1$  și 0 ale circuitelor DIV1, respectiv AMD1,  $\phi 2$  și 0 ale circuitelor DIV2, respectiv AMD2 și  $\phi 3$  și 0 ale circuitelor DIV3, respectiv AMD3. În funcție de constanta de divizare, este selectată una din intrările 0, 1, 2, 3, 4, 5, 6, 7, care reprezintă trepte de divizare ale semnalului. Frecvența divizată este obținută la ieșirile OUT1, OUT2 și OUT3 ale circuitelor AMD1, respectiv AMD2 și AMD3 și este aplicată la intrările de întrerupere externă INT0 ale microcontrolerelor M3, M4 și M5. Microcontrolerul M3, M4, M5 au rolul de a multiplica aceste semnale, astfel încât la fiecare impuls de la intrările INT0 se generează la ieșirile P2.0 un tren de impulsuri, al cărui număr este dat de constanta de multiplicare. Frecvența multiplicată este aplicată atât la intrările ay, by, cy ale circuitului AMD4, cât și la intrările INT0, INT1, T0 ale microcontrolerului M2. M2 are rolul de a însuma cele trei frecvențe și de a măsura energia pe fiecare din cele trei intrări. Însumarea celor trei frecvențe se face prin generarea la fiecare impuls recepționat pe fiecare dintre cele trei intrări, a unui impuls la ieșirea P2.3, ieșire conectată la intrările ax, bx, cx, legate împreună, ale circuitului AMD4. Măsurarea energiei pe fiecare din cele trei intrări (faze) se face prin contorizarea frecvențelor corectate cu constantele de multiplicare-divizare, aplicate la intrările INT0, INT1 și T0 ale microcontrolerului M2 în trei numărătoare de câte trei octeți. La comanda de măsurare energie recepționată de microcontrolerul M1 acesta transmite pe linia P2.3 la intrarea de timer T1 a microcontrolerului M2 un semnal care determină citirea, transmiterea spre microcontrolerul M1 pe liniile P0.0, P0.1, P0.2, P0.3, P0.4, P0.5,

P0.6, P0.7, cu multiplexarea realizată pe liniile P2.6 și P2.7 și anularea conținutul numărătoarelor corespunzătoare celor trei faze. Din datele de configurare a instalației de etalonare/verificare se cunoaște tipul și numărul de convertoare putere-frecvență utilizate, informație ce se transmite printr-un cuvânt de comandă spre interfața de achiziție și care determină valoarea de potențial a ieșirii P1.3 a microcontrolerului M1. Dacă sunt trei convertoare putere-frecvență monofazate și se execută etalonarea/verificarea unor contoare monofazate, atunci P1.3 va fi programat la valoarea de potențial "1" astfel încât la ieșirile "ax sau ay", "bx sau by" și "cx sau cy" se vor obține prin circuitele BC12, BC13 și BC14 semnalele de frecvență etalon FeR, FeS și FeT, diferite pe fiecare ieșire. Dacă se etalonează/verifică contoare trifazate, atunci ieșirea P1.3 este programată cu valoarea de potențial "0", iar pe ieșirile "ax sau ay", "bx sau by" și "cx sau cy" va apărea frecvența etalon însumată, aceeași pe toate ieșirile.

### ***5.2.3.2 Ordinograme de funcționare pentru postul de calcul și afișarea erorii, interfața de achiziție și programul implementat pe calculatorul PC***

În ANEXA 1 este prezentată ordinograma de funcționare a postului de calcul și afișare a erorii (blocuri 1 până la 49), ordinograma de funcționare a microcontrolerelor care fac parte din interfața de achiziție (blocuri 50 până la 108) cât și ordinograma de funcționare a programului implementat pe calculatorul PC (blocurile 109 până la 125).

Pentru postul de calcul, programul principal prezentat în ANEXA 2, execută, bloc 1, apoi verifică dacă s-a recepționat un cuvânt de comandă de la calculator, bloc 2, 3, ceea ce ar duce la găsirea fanionului FAN pe "1" logic. Dacă nu s-a recepționat nici un cuvânt, se intră în bucla de calcul a erorii, verificându-se dacă contorul a făcut prima rotație completă (a venit marcă), bloc 4. Dacă a venit marcă de la contorul de etalonat/verificat, se verifică dacă sunt îndeplinite condițiile de calcul a erorii, bloc 5, adică existența tuturor datelor necesare în calculul erorii. În caz afirmativ, se trece la calculul erorii, bloc 6, și apoi afișarea ei, bloc 7, 8. Se așteaptă în continuare venirea unei mărci, bloc 9, 10. Dacă a venit marca, se determină modul de lucru programat, bloc 11, apoi se calculează eroarea, bloc 12 și se afișează, bloc 13. În continuare, bloc 14, se incrementează un contor de timp, bloc 17 și se verifică dacă contorul a măsurat deja 2

secunde, bloc 18. Dacă au trecut 2 secunde, fără să fi venit o nouă marcă, se pregătește un mesaj de lipsă marcă și se anulează contorul de 2 secunde, bloc 19, mesaj care se transmite spre afișaj, bloc 20. Programul se reia, așteptând venirea unui nou cuvânt de comandă, bloc 2 sau a unei mărci. Dacă s-a recepționat un cuvânt de comandă de la calculatorul PC, bloc 15, se determină modul de lucru transmis, se anulează fanionul FAN și se reia programul, bloc 2. Dacă nu a venit marcă, bloc 4, se așteaptă într-o buclă de timp de 2 secunde venirea mărcii. Dacă ea se înregistrează după trecerea timpului programat, bloc 21, se trece la calculul erorii, bloc 9. Dacă după trecerea acestui timp nu a venit marcă, se pregătește mesaj pentru situația de absență marcă, bloc 22 și programul continuă cu afișarea erorii, bloc 7, 8. Dacă există marcă, se va trece la calculul erorii, bloc 9, 10, 11. Dacă nu sunt prezente datele pentru calculul erorii, se pregătește mesaj pentru lipsă constantă, bloc 16, apoi acesta se afișează, bloc 7, 8. Când se recepționează un cuvânt de comandă de la calculatorul PC pe intrarea de întrerupere serială, se declanșează subrutina care deservește această situație, bloc 23. Se salvează cuvântul recepționat, bloc 24, se verifică dacă acesta este dedicat postului de calcul, bloc 25, iar în caz că se adresează postului de calcul, se verifică dacă este primul cuvânt din seria care se transmite, bloc 26. Dacă este primul cuvânt, înseamnă că s-a recepționat un cod ce determină modul de lucru programat. Se memorează codul, bloc 27 și se trece la determinarea modului de lucru. Dacă este comandat modul de lucru învățare, bloc 28, programare constantă, bloc 29, sau programarea numărului de ture, bloc 30, se știe că urmează un nou cuvânt, de aceea se incrementează numărătorul de cuvinte NRC, bloc 40, se anulează bucla de timp, bloc 33, 41, care gestionează așteptarea setului de cuvinte de comandă, funcție de modul de lucru programat, și apoi se iese din subrutina de tratare a întreruperii seriale, bloc 38. Dacă se decodifică comandă de achiziție erori, bloc 31, sau autotest, bloc 32, înseamnă că s-a terminat de transmis setul de cuvinte de comandă, se anulează numărătorul de cuvinte, NRC și se poziționează pe "1" logic fanionul FAN, bloc 42, apoi se revine din subrutina de tratare a întreruperii seriale, bloc 38. Dacă s-a recepționat comandă de tip învățare, transmitere constantă sau programare număr de ture, se vor transmite mai multe cuvinte. Dacă nu este primul cuvânt, bloc 34, se verifică dacă s-a recepționat al doilea cuvânt, bloc 43. Dacă este al doilea cuvânt și s-a transmis modul de lucru învățare, bloc 44, se salvează numărul postului inițial, bloc 45, după care se iese din program prin bloc 42, 38. Numărul transmis va fi numărul primului

post de calcul, următoarele stabilindu-se prin modul de lucru învățare. În caz că s-a transmis modul de lucru programare număr ture, bloc 46, se salvează numărul transmis și se pregătește ieșirea din subprogramul de tratare întrerupere, bloc 42. Dacă s-a transmis modul de lucru programare constantă, se salvează primul octet al constantei, bloc 48 și se pregătește recepția următorului cuvânt, bloc 40. Dacă s-a transmis al treilea cuvânt, aceasta se referă numai la modul de lucru programare constantă, memorându-se al doilea octet al constantei, bloc 49 și apoi pregătindu-se ieșirea din subprogram, bloc 42. Dacă cuvântul recepționat nu este pentru postul de calcul respectiv, se verifică dacă el s-a transmis pentru toate posturile, bloc 35. În caz că a fost transmis pentru toate posturile, se transmite spre celelalte posturi, bloc 36 și se inițializează o buclă de timp, bloc 37, apoi se pregătește ieșirea din subrutină, bloc 38, 39. Dacă se manifestă o eroare de transmisie, după trecerea timpului măsurat de bucla de timp, se pregătește recepția unei noi serii de cuvinte prin anularea numărătorului de cuvinte.

Programarea constantelor de multiplicare-divizare a frecvenței etalon pentru interfața de achiziție IA se face de la calculatorul PC, constanta fiind calculată din datele instalației de etalonat/verificat astfel încât prin multiplicarea-divizarea frecvenței etalon (a frecvențelor etalon) să se obțină, un număr  $n$  - formula (5.2) de impulsuri.

Programarea modului de generare a frecvenței etalon către posturile de calcul se face de la calculatorul PC către interfața de achiziție IA în scopul de a obține frecvențele etalon separate (pe fiecare linie de ieșire din IA frecvențele FeR, FeS, FeT sunt diferite) sau însumate (pe cele trei linii de ieșire din IA, frecvențele FeR, FeS, FeT sunt însumate, ieșirile fiind identice).

Modul de lucru comandat de la calculatorul PC care cere măsurarea energiei electrice consumată într-o anumită perioadă de timp, comandă citirea energiei măsurate de contor într-o anumită perioadă de timp și afișarea ei pe monitorul calculatorului PC.

În continuare este prezentată ordinograma de funcționare a microcontrolerelor care fac parte din interfața de achiziție (blocuri 50 până la 108). Softul aferent funcționării interfeței de achiziție este prezentat în ANEXA 3. Microcontrolerul M1 (blocuri 50 până la 62) are rolul să gestioneze transferul serial de date dintre interfață, respectiv posturile de calcul și calculatorul PC. După inițializările fanioanelor, bloc 50, se verifică recepționarea unui cuvânt, bloc 51, 52. Dacă s-a recepționat un cuvânt, fanionul FAN este găsit cu

valoarea logică "1" și se trece la determinarea cuvântului recepționat pentru interfața de achiziție. Dacă acesta este programarea constantelor de multiplicare-divizare, bloc 53, atunci după retransmisia cuvintelor la calculatorul PC și memorarea constantelor, bloc 55, se anulează fanionul FAN, bloc 56, 57, pregătindu-se în acest fel posibilitatea unei noi recepții. Programul se reia în bucla de așteptare cuvânt recepționat, bloc 51. Dacă s-a transmis comanda de programare a modului de generare a frecvenței etalon, pe o fiecare linie separat sau însumate (M/T), bloc 58, se trece la prelucrarea și execuția comenzii, se retransmite la calculatorul PC, bloc 59, după care se pregătesc condițiile unei noi recepții, bloc 56. Dacă s-a transmis comanda de măsurare energie, bloc 60, se startează măsurarea energiei pe fiecare fază, bloc 61, se retransmite la PC comanda împreună cu măsurătorile efectuate pe fiecare fază în parte, bloc 62, apoi se fac pregătiri pentru așteptarea unui nou cuvânt transmis de la calculatorul PC. Subrutina care tratează recepția cuvintelor la interfața de achiziție este similară cu cea prezentată la postul de calcul și afișarea erorii (blocuri 23 până la 49). Recepția unui cuvânt de comandă (set de octeți) este marcată prin poziționarea fanionului FAN pe "1" logic. Subrutina recepționează cele trei tipuri de comenzi care se pot adresa către interfața de achiziție, comenzi care determină modurile de lucru prezentate mai sus.

Microcontrolerul M2 (blocurile 63 până la 86) este prevăzut să însumeze cele trei frecvențe etalon și să le genereze. Fiecare frecvență etalon, după procesul de multiplicare-divizare, generează o întrerupere care este tratată cu câte o subrutină de felul următor: întreruperea sesizată, bloc 63, declanșează la ieșirea P2.3 a portului microcontrolerului M2 un semnal "1" logic, bloc 64, apoi incrementează un numărator de impulsuri, MEMi cu  $i=1...3$ , pentru fiecare fază separat, bloc 65, după care poziționează pe "0" logic aceeași ieșire de port P2.3, bloc 66, încheind generarea unui impuls de frecvență etalon și terminând astfel tratarea întreruperii, bloc 67. Dacă s-a recepționat o întrerupere care generează comanda de măsurare a energiei, bloc 68, se salvează conținutul numărătoarelor MEMi, în care s-au contorizat impulsurile de frecvență etalon, bloc 69, apoi acestea se anulează, bloc 70, se poziționează un fanion FAN pe "1" logic, bloc 71, ceea ce va indica microcontrolerului că este citită energia contorizată până în acel moment, și terminând astfel tratarea întreruperii, bloc 72. Programul principal implementat în microcontrolerul M2, după inițializări, bloc 73, urmărește transmisia unei comenzi de măsurare energie, urmărind conținutul fanionului

FAN, bloc 74, 75. Dacă acesta este "1", după anularea sa, bloc 76, se transmite energia măsurată spre calculatorul PC, bloc 74, apoi se reintră în bucla de așteptare comandă de măsurare energie, bloc 74. Dacă nu s-a primit comanda de măsurare energie, bloc 75, programul rămâne într-o buclă de așteptare a acesteia, bloc 74. Perioada la care se generează impulsurile de frecvență etalon este realizată cu ajutorul timer-ului TIMER 1 al microcontrolerului M2. Numărul de impulsuri care se vor genera se află memorate în contorul Nrimp, iar durata dintre două impulsuri este memorată în număratoarele TIMER-ului 1. În acest caz, întreruperea de overflow a număratorului TIMER 1 lansează în execuție subrutina de tratare a întreruperii, bloc 78. Se generează un semnal "1" logic la ieșirea portului P2.0 a microcontrolerului, bloc 79, se reîncarcă număratoarele timer-ului la valoarea inițială care dă durata de timp între două impulsuri, bloc 80, apoi se decrementează număratorul de impulsuri Nrimp, bloc 81. Se verifică dacă s-au generat toate impulsurile programate. Dacă s-au generat toate impulsurile, număratorul Nrimp este nul, bloc 82 și se dă comanda de stop numărare la timer, bloc 83, apoi se generează semnal "0" logic pe ieșirea P2.0 a portului microcontrolerului M2, bloc 84, 85 și se revine din subrutina de tratare a întreruperii date de TIMER 1, bloc 86. Aceiași pași sunt parcurși și în cazul în care nu s-a terminat de generat numărul de impulsuri programat, bloc 82, 84, 85, 86.

Microcontrolerul M3 (respectiv M4 și M5) au implementată aceeași structură soft (blocuri 87 până la 108), cu rol de a realiza multiplicarea semnalelor venite de la convertorul putere-frecvență. Programul principal, după inițializări, bloc 87 intră într-o buclă de așteptare a unei întreruperi generate de un impuls de frecvență etalon, bloc 88, 89. Când apare un impuls, se lansează în execuție subrutina care tratează întreruperea, bloc 90. Se citește și se salvează conținutul număratorului TIMER 0, bloc 91, cu care se determină distanța, ca durată, dintre două impulsuri etalon. Se anulează număratorul TIMER 0, bloc 92, pregătindu-l astfel pentru următoarea numărare. Se citește factorul de multiplicare, bloc 93, și se face o decodificare a lui. Dacă are valoarea 0, bloc 94, impulsul rămâne nemodificat, bloc 95, și se revine din subrutina de tratare a întreruperii, bloc 96. Dacă factorul de multiplicare ia una dintre valorile 1 bloc 97, 2 bloc 99, 3 bloc 101, 4 bloc 103, 5 bloc 105 sau 6 bloc 107, se procedează la memorarea numărului de impulsuri în locația Nrimp și se determină durata dintre două impulsuri care se vor genera (care se memorează în număratorul TIMER 1 al microcontrolerului M2), astfel încât impulsurile generate să fie uniform repartizate între două impulsuri venite de la convertorul

putere-frecvență, operație prezentată în blocurile 98, respectiv 100, 102, 104, 106 și 108. După aceste operații se face ieșirea din subrutina de tratare întrerupere, bloc 95.

În continuare este prezentată ordinograma de funcționare a programului implementat pe calculatorul PC (blocurile 109 până la 125), care dă posibilitatea operatorului să transmită comenzi sistemului. Softul aferent este prezentat în ANEXA 6 [44], [45]. După configurarea sistemului, bloc 109, unde sunt introduse date legate de caracteristicile tehnice ale convertorului putere-frecvență etalon utilizat, de tipul stației și de tipul contoarelor de etalonat/verificat, se introduc pe rând constantele contoarelor de etalonat/verificat, bloc 110, numărul de ture după care se va calcula eroarea, bloc 111, probele de curent și de tensiune la care se va supune contorul, bloc 112. Ținând cont de aceste date, programul determină constantele de multiplicare-divizare ale frecvenței etalon, modul de generare a frecvenței etalon, date pe care le transmite împreună cu numărul de ture către posturile de calcul și interfața de achiziție, operație marcată prin blocul 113. După acest transfer, programul rămâne în așteptarea unei noi comenzi, bloc 114, 115. Dacă nu se recepționează comandă, se continuă așteptarea, bloc 114. Dacă se recepționează o comandă, se trece la decodificarea și execuția ei care se termină cu transmisia unui cuvânt de comandă către posturile de calcul sau interfață. Astfel se pot transmite comenzile de învățare post, bloc 116, 117, achiziția erorii, bloc 118, 119, o nouă probă de curent și tensiune, bloc 120, 121, autotestul sistemului, bloc 122, 123 sau comanda de măsurare a energiei, bloc 124, 125. După execuția comenzilor, se trece din nou în bulca de așteptare a unei noi comenzi.

Orice comandă lansată de la calculatorul PC este recepționată la intrarea Rx a interfeței de achiziție IA și este preluată fie de interfața de achiziție, fie de posturile de calcul PCi, funcție de adresa de destinație a comenzii. Prin ieșirea SO a interfeței de achiziție IA, comanda este transmisă la primul post de calcul, la intrarea SI. Mecanismul de transmitere al informațiilor de la primul post de calcul la următorul este același pentru toate posturile de calcul PCi, adică de la ieșirea SO la intrarea SI a postului următor. Fiecare post răspunde la comenzile primite de la calculatorul PC, cu un anumit număr de octeți, pe linia comună LD, care prin interfața de achiziție, intrarea LD, este transmis la calculatorul PC, prin ieșirea Tx a interfeței de achiziție IA. După ce fiecare post de calcul PCi a fost învățat să se recunoască la un anumit număr, a fost programat cu constanta specifică, funcție de tipul contorului de etalonat/verificat și caracteristicile instalației de



etalonat/verificat și a fost programat cu un număr de rotații ale discului contorului după care să se realizeze calculul erorii, postul de calcul execută calculul erorii după formula (5.2), în felul următor: convertorul putere-frecvență CPF1 (sau convertoarele putere-frecvență CPF1, CPF2, CPF3) generează un tren (trenuri) de impulsuri care se aplică la intrarea FinR (respectiv la intrările FinR, FinS, FinT) interfeței de achiziție IA și suportă o prelucrare în cadrul interfeței, funcție de constanta de multiplicare-divizare determinată în funcție de datele de instalație, proba de etalonare/verificare la care este supus contorul și constanta acestuia. Semnalul prelucrat este transmis către posturile de calcul, intrările FeR, FeS, FeT prin ieșirile corespunzătoare FeR, FeS, FeT ale interfeței de achiziție IA. Sesizorul optoelectronic SOi transmite la fiecare rotație completă a discului contorului de etalonat/verificat un impuls la intrarea M a fiecărui post de calcul. În funcție de numărul de rotații complete ale discului contorului Whi programat pentru calculul erorii, este determinată o fereastră de măsură în care postul de calcul numără impulsurile etalon și execută calculul erorii conform formulei (5.2). Rezultatul este afișat pe afișajul cu cristale lichide al postului de calcul în domeniul -19,99...+19,99%. Dacă eroarea determinată nu aparține acestui domeniu, atunci se mai pot afișa câteva mesaje suplimentare: "+E." pentru eroare mai mare decât +19,99%; "-E." pentru eroare mai mare decât -19,99%; "-PE" dacă lipsește frecvența etalon; "-LO" dacă lipsește semnalul de la sesizorul optoelectronic SOi; "- - -" dacă eroarea este în curs de calcul.

Programele realizate pe baza acestor ordinograme sunt prezentate în ANEXA 2, ANEXA 3 și ANEXA 6.

### ***5.2.3.3 Blocul de comandă și blocul de măsură la nivel de schemă bloc***

Pentru realizarea operațiilor de comandă, reglare și măsură a generatoarelor polifazate de energie electrică integrate în instalațiile de etalonare/verificare a contoarelor de energie electrică, sunt cunoscute instalații care au blocul de comandă al generatoarelor realizat din relee și logică de interblocare cu relee, ca elemente de măsură a mărimilor electrice fiind dotate cu instrumente de panou, iar operațiile de reglare sunt executate manual, rezultatul reglării urmărindu-se pe instrumentele de măsură [9].

Metoda prezintă două mari dezavantaje: unul este acela că pentru creșterea preciziei de reglare s-a impus împărțirea în multe

domenii de curent și tensiune, ceea ce duce la un număr ridicat de componente (relee), deci la preț ridicat și fiabilitate scăzută, iar al doilea dezavantaj este acela că reglarea mărimilor electrice se realizează manual, fiind influențată de subiectivismul operatorului la citirea instrumentelor de măsură.

Sistemul pe care l-am conceput, realizat și implementat pe stația SE 04, înlătură aceste dezavantaje și putem spune că au rezultat următoarele avantaje:

- permite reducerea numărului de domenii de curent și tensiune, cu păstrarea preciziei de reglare, simplificarea transformatorului de curent, eliminându-se contactoarele și elementele corespunzătoare domeniilor respective, ceea ce duce la creșterea fiabilității și scăderea prețului de realizare a instalației;

- elimină subiectivismul operatorului în reglarea mărimilor prescrise;

- se poate implementa pe orice instalație de etalonat care are în dotare autotransformatoare cu servomotoare, pentru prescrierea mărimilor de lucru.

Sistemul este alcătuit din două blocuri, figura 5.6: un bloc de comandă, BC care comunică cu un calculator compatibil PC, recepționând comenzi și transmițând informații spre PC și un bloc de măsură, BM a parametrilor electrice ai instalației (curent, tensiune, defazaj) care comunică cu calculatorul PC, în sensul că primește comenzi de la calculator și transmite spre acesta datele măsurate. Comenzile recepționate de blocul de comandă se transmit spre sursele de tensiune și cele de curent, obținându-se valorilor prescrise ale acestora. Comenzile recepționate de blocul de măsură determină prelevarea anumitor mărimi electrice necesare funcționării instalației într-un anumit regim de lucru și transmiterea lor spre calculatorul PC.

În schema bloc din figura 5.6 s-au prezentat cu linie întreruptă următoarele componente ale instalației: un decalor, DEC cu rolul de a realiza o defazare între sistemul de tensiuni și curenți ai instalației, două surse: una de tensiune, SURSA U și una de curent, SURSA I cu rolul de a genera sistemul trifazat de tensiuni, respectiv de curenți necesar pentru etalonarea contoarelor de energie electrică, un bloc de contorizare a energiei, CTW cu rolul de a afișa puterea în circuitele de ieșire la comanda dată de BC și un bloc al tastaturii de pe panoul instalației, BT cu rol de a selecta parametrii de lucru ale instalației.

Calculatorul compatibil IBM-PC, PC cu monitor, M cu rolul de a transmite comenzile de lucru către blocurile de control și de măsură, fiind dotat cu un monitor auxiliar, MA cu rolul de a permite afișarea

mărimilor electrice prescrise. Programul soft implementat pe PC urmărește valorile măsurate și cele prescrise ale parametrilor instalației și realizează funcția de reglare a acestora.

Funcționarea sistemului la nivelul schemei bloc poate fi descrisă în felul următor: modurile de lucru posibile pentru sistem sunt facilitate de un pachet de programe implementat pe calculatorul PC, în blocul de comandă și în blocul de măsură. Datorită acestor programe și datorită configurației sistemului, acesta poate executa următoarele funcțiuni: selectare domenii de tensiune, selectare domenii de curent, selectare regimuri de măsură și succesiunea fazelor, comandă acționarea autotransformatoarelor și conectarea panourilor.

Funcțiunile sunt realizate de următoarele tipuri de taste:

- sunt taste de selectare domeniu de tensiune, de selectare procent din tensiune, de tipul o singură tastă acționată o dată, celelalte taste acționate nu se iau în considerare;
- sunt taste de conectare panouri la tensiune și curent, cu observația că la fiecare apăsare a unei taste de conectare tensiune sau curent pe cele trei faze separate, se schimbă starea releului corespunzător, iar dacă se acționează tastele deodată, se schimbă starea celor trei relee dacă cele trei comenzi de conectare pe faze sunt în aceeași stare, sau se deconectează în caz contrar;
- sunt taste de selectare domeniu de curent, de tipul o singură tastă acționată o dată din toate tastele asociate domeniilor de curent; la acționarea unei taste se comandă releele corespunzătoare domeniilor de curent până la condiția de curent zero (I0), după care se conectează domeniul corespunzător tastei acționate;
- sunt taste de acționare autotransformatoare pe curent, tensiune și decaljul dintre curent și tensiune, cu observația că se acceptă mai multe taste acționate simultan, comanda la relee dându-se simultan. La acest tip de taste există două viteze de acționare a autotransformatoarelor. Dacă acționarea unei taste durează mai mult de o secundă, se conectează automat a doua viteză de acționare. Viteza de acționare a autotransformatoarelor devine zero când se termină acționarea tastei;
- sunt taste de selectare regimuri de măsură, de tipul o singură tastă acționată o dată, cu rolul de a acționa automat releul corespunzător unui anumit regim;

- sunt taste de selectare succesiune faze, cu observația că la conectare se impune o anumită stare a releului corespunzător, stare care se modifică la fiecare acționare a sa.

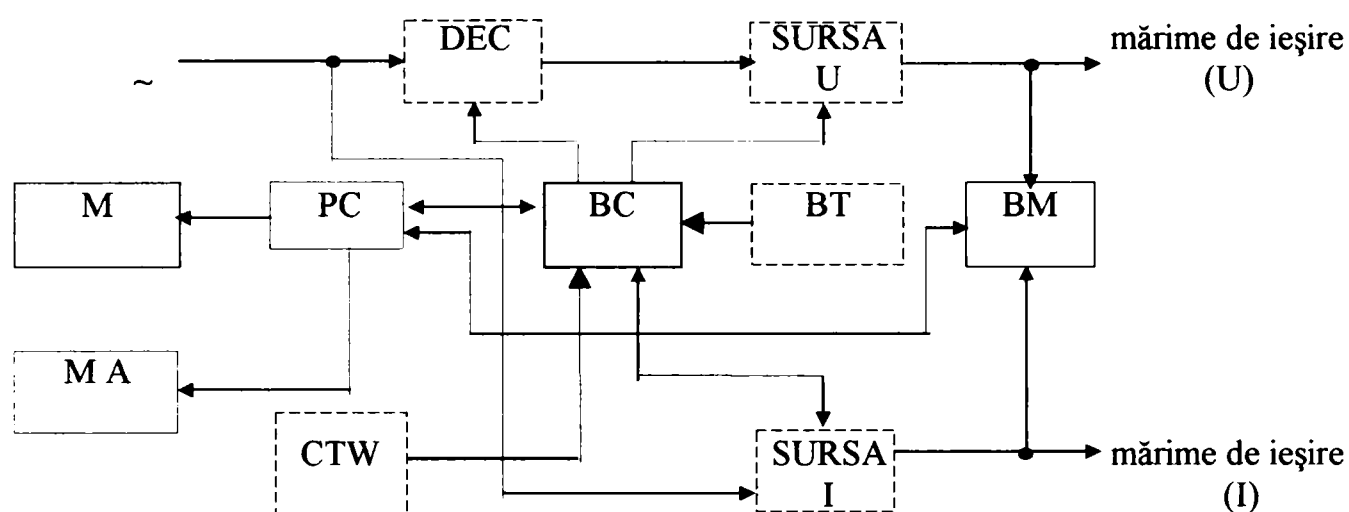


FIGURA 5.6

#### 5.2.3.3.1 Blocul de măsură

Am conceput blocul de măsură, prezentat în figura 5.7 [2], [11], [15], [20], alcătuit dintr-un circuit care conține șase convertoare analog-digitale de precizie CI1, cu trei intrări (I1, I2 și I3) conectate la transformatoarele de curent și trei intrări (U1, U2 și U3) conectate la divizoarele de tensiune. Printr-o intrare de selecție, CS1 circuitul este conectat la o poartă de tip ȘI, CI2 a cărei intrări sunt conectate la ieșirile INI și SOUT1 ale unui circuit decodor CI3, respectiv ale unui circuit de tip procesor numeric de semnal CI4. Procesorul numeric de semnal, CI4 este conectat prin patru intrări de comandă, SC și o intrare de date, DATA la circuitul CI1, iar cu ieșirile SOUT0 și SOUT1 la un circuit memorie de tip EEPROM, CI5 la intrările SK și D1. O intrare de selecție a circuitului CI5, CS5 este conectată la ieșirea INI a circuitului CI3, iar o intrare de date, D0 este conectată la intrarea/ieșirea de date a circuitului CI1. Circuitul CI4 are o magistrală de date (8 linii), DBUS0-7 prin care se conectează la un circuit de tip

memorie FIFO, CI6. Scrierea și citirea din memoria FIFO este comandată la o intrare W, intrare conectată la ieșirea unui circuit de tip poartă ȘI-NU, CI7. Intrările circuitului CI7 sunt conectate câte una la o ieșire ADDR1 a circuitului CI4, respectiv la o ieșire STROBE a circuitului CI4, trecută printr-un circuit de tip inversor, CI8. Aceeași ieșire STROBE este conectată și la o intrare, CL a circuitului CI3. Circuitele CI3 și CI4 sunt conectate și printr-o magistrală suplimentară de date, S/MBUS, de 8 linii. La o intrare G a circuitului CI3 se conectează ieșirea unui circuit de tip poartă ȘI-NU, CI9 care are intrările conectate la ieșirile ADDR0, respectiv RD/WR ale circuitului CI4, ieșirea RD/WR fiind trecută printr-un circuit de tip inversor, CI10. Circuitul CI3 are o ieșire DATRDY care se conectează la o intrare INT0 a unui circuit de tip microcontroler, CI11. Circuitul CI11 este conectat prin intrările P1.0 și P1.1 la ieșirile R, respectiv RS ale memoriei FIFO, CI6. Pe o magistrală, D0-7 se face legătura între circuitele CI6 și CI11, iar prin liniile P3.0 (RxD) și P3.1 (TxD) ale microcontrolerului CI11 se face legătura între blocul de măsură și blocul de control, printr-un circuit de translatare de nivel și separare galvanică, CI12 care se conectează la liniile SI și LD ale sistemului numeric de comandă, reglare și măsură a generatoarelor polifazate de energie.

Funcționarea blocului de măsură este prezentată mai jos.

La intrările de curent (I1, I2, I3) și la intrările de tensiune (U1, U2, U3) ale circuitului CI1 se aplică mărimile analogice de curent, respectiv de tensiune de la transformatoarele de curent, respectiv de la divizoarele de tensiune. Circuitul realizează conversia semnalelor analogice în semnale numerice, cu o precizie asigurată de niște constante programate în memoria EEPROM, CI5. Memoria este de tip serial, datele transmițându-se de la ieșirea DO a circuitului CI5 la intrarea/ieșirea DATA a circuitului CI1. Memoria EEPROM este programată pe liniile SK și D1 cu datele și semnalul de tact de la procesorul de semnal CI4. De asemenea, selecția memoriei se face la intrarea CS5, cu semnalul INI generat prin decodificarea magistralei suplimentare a procesorului de semnal de către circuitul decodor CI3. Citirea mărimilor numerice din CI1 se face în momentul în care circuitul este selectat pe intrarea de selecție CS1. Semnalul de selecție este obținut printr-o funcție logică ȘI între semnalul INI generat de CI3 și SOUT1 generat de procesorul numeric de semnal, CI4. Stabilirea transferului de date se face în anumite condiții gestionate de semnalele de comandă și control SC dintre circuitele CI1 și CI4. Procesorul numeric de semnal transferă la rândul său date recepționate

de la convertoare într-o memorie de tip FIFO, CI6 pe liniile de date DBUS0-7. Semnalul care comandă înscrierea în memoria FIFO, aplicat la intrarea W a circuitului CI6 se obține ca o funcție logică de tip ȘI-NU între un semnal de STROBE complementat prin circuitul inversor CI8 și semnalul ADDR1 generat de procesorul de semnal. Datele salvate în memoria FIFO sunt preluate de microcontrolerul CI11 pe liniile D0-7, în momentul determinat de semnalul DATRDY obținut prin decodificarea magistralei suplimentare a procesorului de semnal de către decodorul CI3. Semnalul apare când datele parametrilor mășurați sunt convertite și citite deja de lanțul de circuite CI1, CI4 și CI6.

Prin setul de programe implementat în microcontrolerul CI11 se

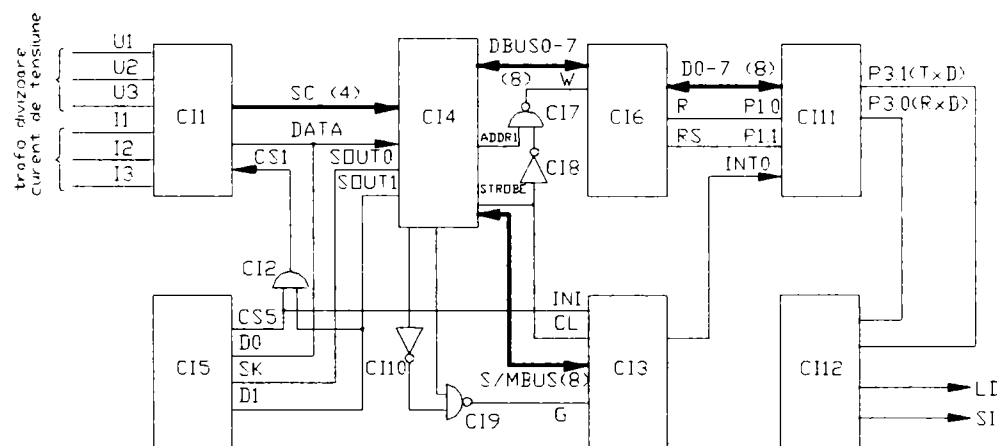


FIGURA 5.7

realizează prelucrarea valorilor numerice a mărimilor de intrare: curent, tensiune și decalajul dintre acestea. De asemenea se măsoară frecvența rețelei, puterea consumată activă și reactivă. Semnalele aplicate la intrările P1.0 și P1.1 ale microcontrolerului CI11 de la ieșirile R și RS ale memoriei FIFO, CI6 stabilesc condițiile de transfer a informațiilor între cele două circuite. La comanda transmisă de calculatorul PC pe linia SI a circuitului CI12, cu rol de adaptor de nivel de tensiune și separator galvanic, datele prelucrate se transmit spre calculator pe linia LD, tot prin circuitul CI12. Semnalele de recepție a comenzii, respectiv transmisie a datelor solicitate se aplică la intrarea P3.0 (RxD), respectiv la ieșirea P3.1 (TxD)

microcontrolerului CI11. Calculatorul PC recepționează datele și face o comparație între valorile programate și cele măsurate. Funcție de rezultatul comparației se decide asupra comenzii care se va transmite spre blocul de comandă. De asemenea, pe afișajul monitorului auxiliar se vor afișa valorile măsurate ale mărimilor curent, tensiune, defazaj, frecvența rețelei, putere activă și reactivă.

### 5.2.3.3.2 Blocul de comandă

Am conceput blocul de comandă, prezentat în figura 5.8 [9], bloc care este realizat pe baza unui sistem cu un microcontroler M1, prevăzut cu trei porturi de intrare/ieșire P1 (P1.0 până la P1.7), P0 (P0.0 până la P0.7) și P2 (P2.0 până la P2.7). Porturile P1, respectiv P0 sunt conectate prin circuitele de tip buffer B10 până la B17, respectiv B20 până la B27 la liniile TI0 până la TI7, respectiv TO0 până la TO7 ale matricii de taste funcționale. La fiecare intrare a circuitelor B20 până la B27 este conectat un rezistor R1 până la R8 cu al doilea terminal conectat la masă.

Pe patru linii ale portului P2 (P2.0 până la P2.3) microcontrolerul M1 este conectat la intrările A, B, C, D ale unui circuit decodor, DEC cu rolul de a selecta una din cele opt ieșiri Q0 până la Q7, ieșiri legate la câte un bloc de comandă BC0 până la BC7. Aceste blocuri au aceeași structură fiecare, ele fiind realizate cu un circuit de tip latch L0 până la L7, cu intrările D0 până la D7 conectate la liniile portului P1 al microcontrolerului M1. Ieșirile circuitului IL0 până la IL7 sunt conectate prin niște rezistori R19 până la R26 în baza unor tranzistori T3 până la T10; prin colectorul acestor tranzistori se transmit comenzile la elementele de execuție, adică relele intermediare și dispozitivele de afișare, emitorul fiind conectat la masă montajului. Alte trei linii ale portului P2 (P2.4 până la P2.6) sunt conectate la ieșirile circuitelor de tip buffer B31 până la B33. Prin fiecare intrare, conectată printr-un rezistor R10 până la R12 masa montajului, se realizează legătura la cele trei faze R, S respectiv T ale blocului de controlarea energiei CTW.

Comunicația cu calculatorul PC se face prin cele două linii seriale SI și LD. Pe intrarea SI sunt recepționate comenzile de la calculatorul PC care ajung la intrarea serială RxD a microcontrolerului M1 printr-un circuit de tip buffer B30 cu intrarea conectată printr-un rezistor R9 la potențialul pozitiv al sursei de tensiune Vcc. Transmisia informațiilor către calculatorul PC se face pe linia LD, de la ieșirea TxD a microcontrolerului M1 prin următorul lanț de circuit: ieșirea

TxD este conectată printr-un rezistor R13 în baza unui tranzistor T2, bază conectată la potențialul pozitiv al sursei de alimentare Vcc printr-un rezistor R14. Tranzistorul T2 are emitorul conectat la potențialul pozitiv al sursei de tensiune Vcc printr-un rezistor R16 și colectorul conectat la masă printr-un rezistor R15, rezistor care se conectează și în baza unui tranzistor T1. Tranzistorul T1 are emitorul conectat la masa sistemului, iar colectorul este linia LD de comunicație cu calculatorul PC. De asemenea, o ieșire de port P2.7 a microcontrolerului M1 este conectată în colectorul unui optocuplor OP care este conectat printr-un rezistor R17 la potențialul pozitiv al sursei de tensiune Vcc. Emitorul optocuplorului este conectat la masa sistemului, iar în partea de recepție, în paralel cu dioda optocuplorului OP este conectată o diodă D1, cu un terminal la potențialul pozitiv al sursei de tensiune Vcc și cu al doilea terminal printr-un rezistor R18, la o intrare CZ cu rol de a sesiza condiția de nul a autotransformatoarelor de curent.

În continuare este prezentată funcționarea blocului de comandă.

Microcontrolerul M1, prin cele două porturi de intrare/ieșire poate gestiona maximum 64 de taste funcționale, organizate într-o matrice de forma 8x8. După determinarea tastei acționată (prin citirea liniilor TI0 până la TI7, respectiv TO0 până la TO7 ale matricii de taste funcționale), funcție de tipul de funcție a tastei decodificată, pe cele patru linii ale portului P2 (P2.0 până la P2.3) microcontrolerul generează un număr de la 0 la 7 în binar, număr decodificat de decodorul DEC. La una dintre ieșirile Q0 până la Q7 va apărea un semnal cu valoarea "0" logic, celelalte ieșiri fiind la valoarea "1" logic, corespunzător cu tasta acționată. Ieșirea va activa blocul de comandă corespunzător, în sensul că circuitul Li cu  $i=0$  până la 7 va activa una dintre ieșirile IL0 până la IL7 funcție de tasta acționată, determinată prin valorile de potențial aflate la ieșirile portului P1, respectiv intrările D0 până la D7 ale blocurilor de comandă corespunzător. Ieșirea activată ILi cu  $i=0$  până la 7, acționează prin perechea rezistor-tranzistor asupra elementelor de execuție, adică relele intermediare și dispozitivele de afișare, în sensul de execuție a funcției tastei acționate.



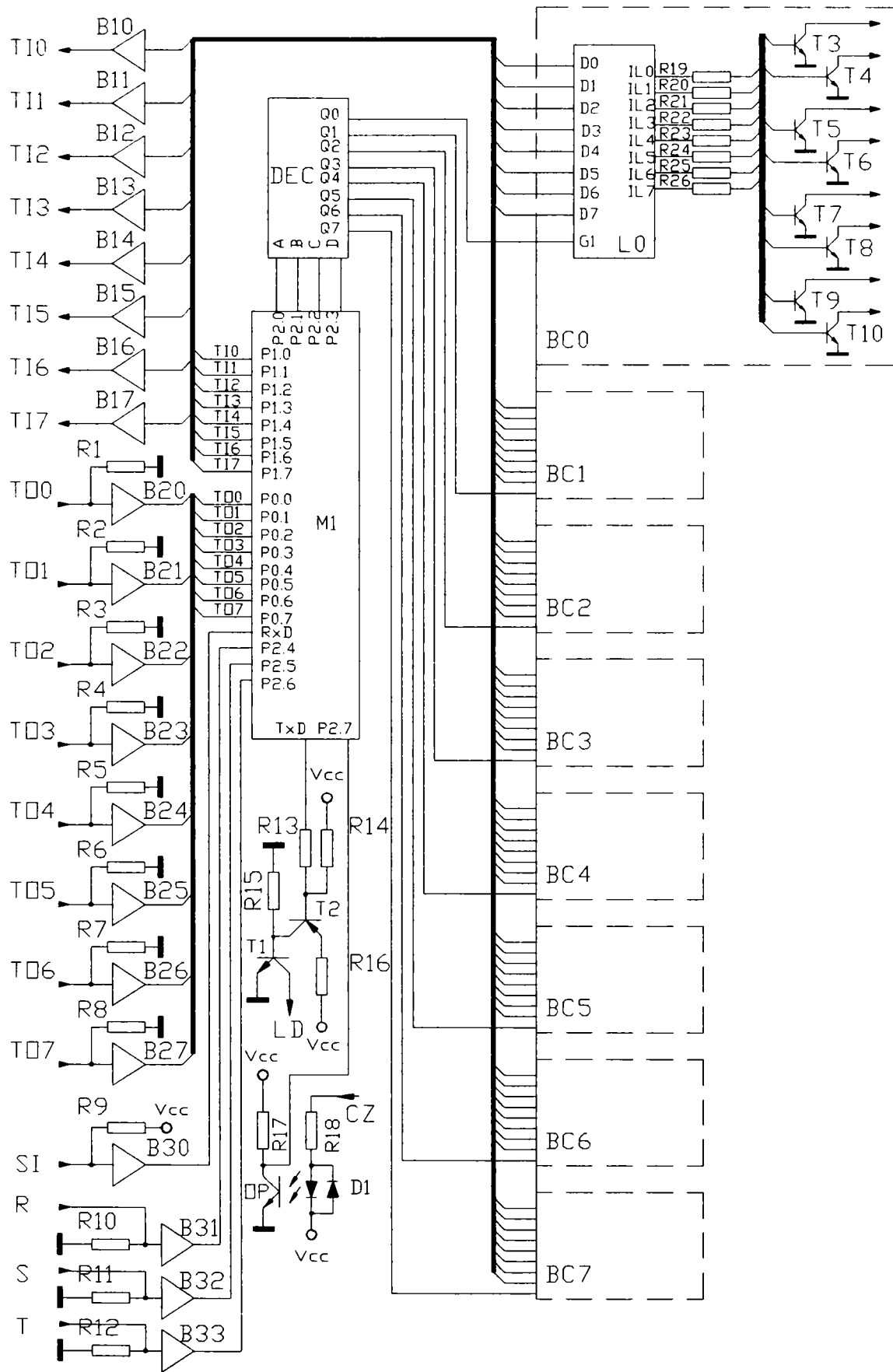


FIGURA 5.8

#### ***5.2.3.4 Ordinograme de funcționare pentru blocul de comandă, blocul de măsură și programul implementat pe calculatorul PC***

În figura 5.9 sunt prezentate ordinogramele de funcționare ale programelor implementate în procesoarele din blocul de comandă și blocul de măsură și reglare, cât și din calculatorul PC.

Microcontrolerul M1 din blocul de comandă are implementat un program conform ordinogramei (blocuri de la 1 la 7). După inițializările care se fac, bloc 1 se așteaptă acționarea unei taste, bloc 3. Dacă s-a acționat o tastă, bloc 5, se determină tipul tastei acționate, bloc 6 și apoi se lansează în execuție lanțul de operații specifice tastei respective, bloc 7. Dacă nu s-a detectat nici o tastă acționată, se verifică dacă nu s-a recepționat o comandă de la calculatorul PC, bloc 4. Dacă a venit o comandă de la PC, bloc 5, după executarea lanțului de operații specifice tastei acționate, blocul transmite un răspuns la PC în care îi comunică valoarea recepționată. Dacă nu a venit nici o comandă de la PC, bloc 2, se așteaptă în continuare acționarea unei taste.

Microcontrolerul M2 din blocul de măsură și reglare are implementat un program conform ordinogramei (blocuri de la 8 la 17). După inițializările care se fac, bloc 8 programul intră într-o buclă de așteptare de întreruperi, bloc 10, modalitate prin care comunică cu procesorul de semnal și cu calculatorul PC. Dacă a venit o întrerupere, aceasta se execută, bloc 11, după care se reia așteptarea unei noi întreruperi, bloc 9.

Dacă întreruperea a venit de la calculatorul PC, bloc 12, se determină și se execută comanda solicitată, bloc 13, după care se face revenirea în programul principal, bloc 14. Dacă întreruperea a fost generată de procesorul de semnal, bloc 15, se citește informația venită de la procesorul de semnal care cuprinde ultimul set de măsurători ale procesorului din memoria FIFO, bloc 16, pregătindu-se datele pentru transmis la calculatorul PC, apoi subrutina revine în programul principal, bloc 17.

Programele pentru blocul de măsură și blocul de comandă sunt prezentate în ANEXA 4 și ANEXA 5.

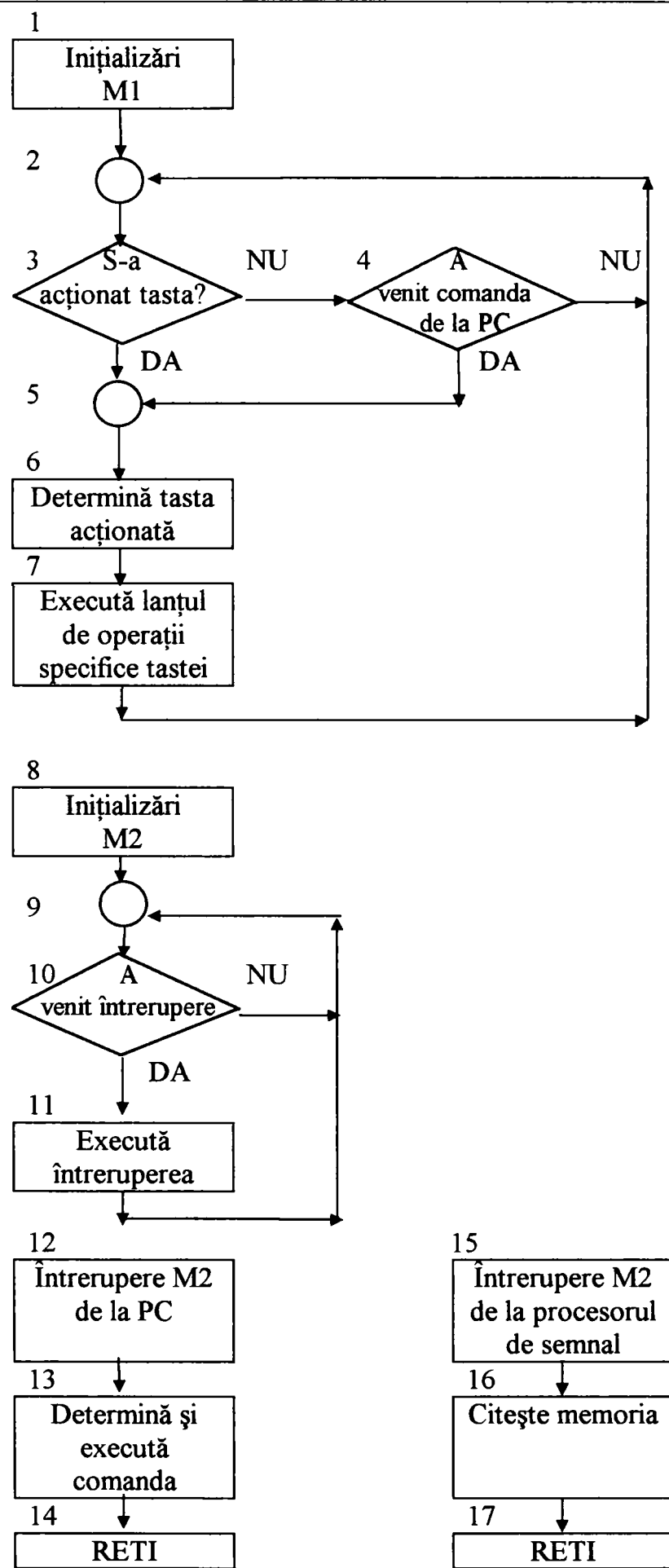


FIGURA 5.9

#### 5.2.4. Caracteristici tehnice principale

- numărul de posturi de lucru: max. 250;
- rezoluția de afișare:  $\pm 0,01\%$ ;
- eroarea suplimentară maximă introdusă de sistem:  $\pm 0,01\% \pm 1$  digit;
- domeniul erorilor afișabile:  $-19,99\%$  până la  $+19,99\%$ ;
- constanta contorului electronic etalon: oricare<sup>\*</sup>;
- rapoartele transformatorului de curent etalon: oricare<sup>\*</sup>;
- rapoartele transformatorului de tensiune etalon: oricare<sup>\*</sup>;
- constantele contoarelor de etalonat/verificat: oricare, diferite pentru fiecare post de lucru în parte<sup>\*</sup>;
- facilități auxiliare:
  - programare ghidată prin tehnica meniurilor;
  - autotest;
  - contorizarea unui număr programabil de kWh;
  - achiziția erorii la calculator cu posibilitatea de editare a unui protocol la imprimantă;
  - existența codurilor de eroare care semnaleză absența frecvenței etalon, a semnalului de marcă de la contor, sau erori mai mari, respectiv mai mici decât limita domeniului de afișare;
  - poate determina eroarea contorului după un număr programabil de rotații ale discului contorului sau de impulsuri ale contoarelor electronice (de la 1 la 255);
  - se poate introduce un filtru soft de timp pentru eliminarea influenței negative introdusă de zgârieturile de pe discul contorului;
- tensiunea de alimentare: 220V/50-60Hz;
- puterea consumată: max 0,5W pentru un post de calcul cu senzor;
- dimensiunile de gabarit ale postului de calcul și afișare a erorii: 135 x 85 x 45mm;

<sup>\*</sup> se definesc la instalare.

### **5.3 Schema bloc și elementele componente ale sistemului de calcul aferent stațiilor de etalonat contoare de energie electrică**

În figura 5.10 este indicată schema bloc a sistemului de calcul aferent stației de etalonat contoare de energie electrică.

Sistemul de calcul al stației este format din:

- calculator de proces;
- contorul (contoarele) etalon;
- blocul de comandă (inclus în SE 04);
- blocul de măsură;
- blocul de intrare (inclus în SE 04);
- interfață de achiziție;
- posturi de calcul;
- sesizoare optice.

*Calculatorul de proces*, figura 5.11, este elementul principal de conducere al procesului, care prin intermediul unei interfețe RS 232C standard realizează comunicarea cu interfața de achiziție de la care



**FIGURA 5.11**

primește date legate de desfășurarea procesului și trimite comenzi spre sistem.

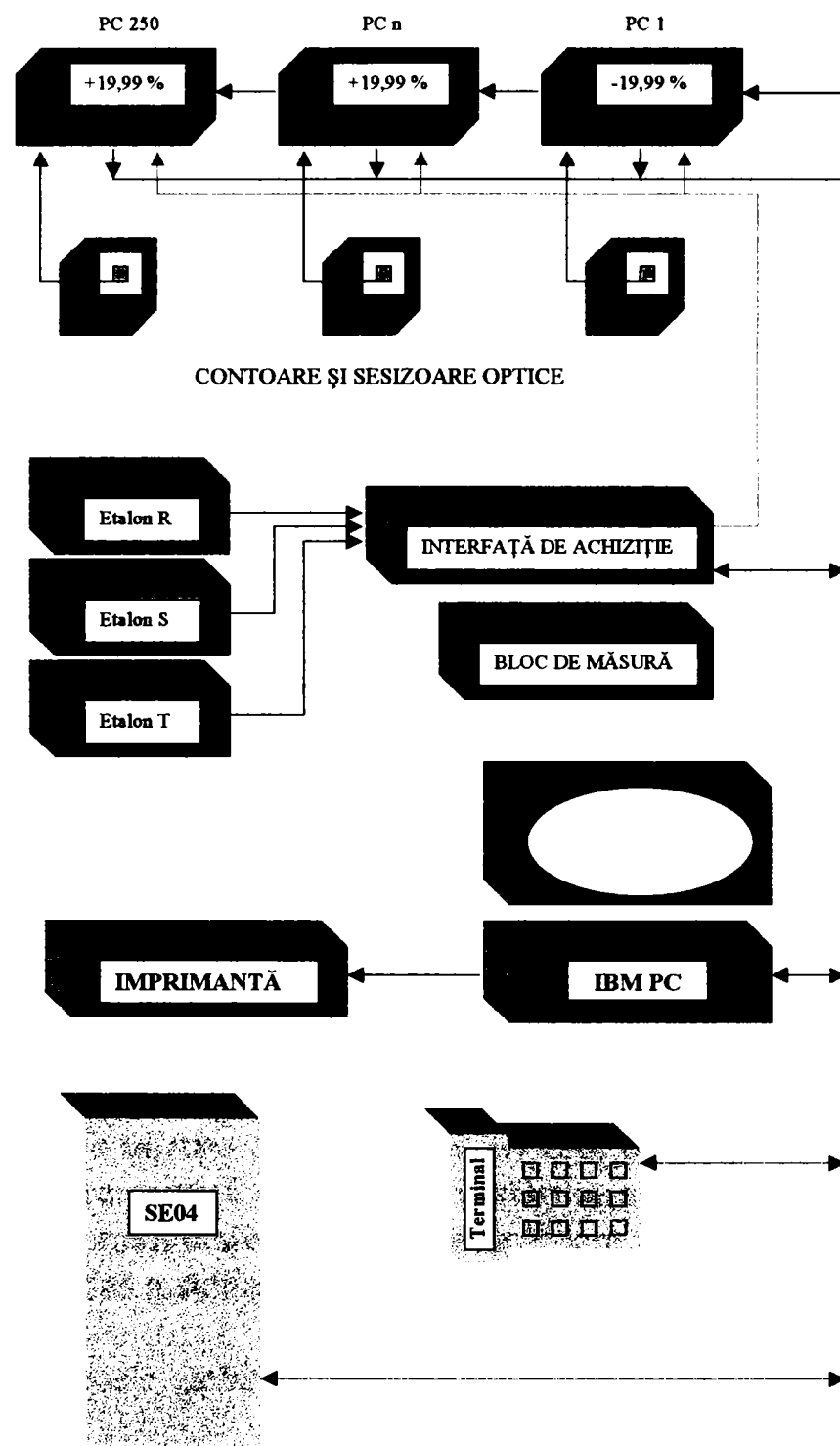


FIGURA 5.10

*Contorul etalon* figura 5.12, este elementul de măsură de precizie care generează un număr de impulsuri proporțional cu energia

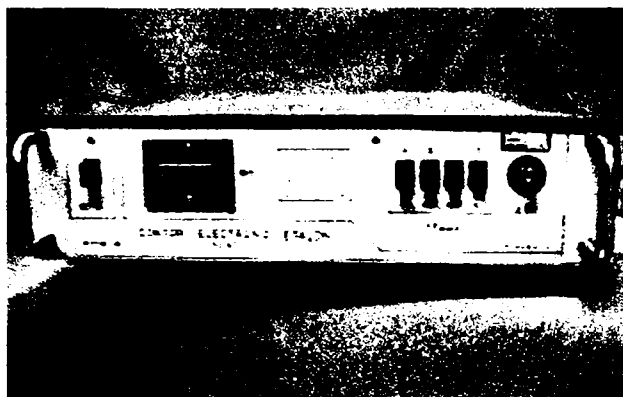


FIGURA 5.12

electrică debitată spre sarcină (contoare).

*Blocul de comandă (inclus în SE 04)* este constituit din totalitatea elementelor de comandă (relee, cuploare optice) care asigură transmiterea comenzilor către toate elementele de execuție ale stației (servomotoare, contactori, elemente de semnalizare) și izolarea galvanică a stației față de blocul de calcul.

*Blocul de măsură* este constituit din totalitatea elementelor de măsură a mărimilor analogice (traductoare de tensiune, curent, factor de putere, împreună cu convertoarele analog - digitale aferente).

*Blocul de intrare (inclus în SE 04)* este constituit din totalitatea elementelor care asigură citirea tuturor stărilor elementelor de execuție (senzori de proximitate, microîntrerupătoare, cuploare optice) și izolarea galvanică a stației față de blocul de calcul.

*Interfața de achiziție*, figura 5.13, este un sistem de calcul

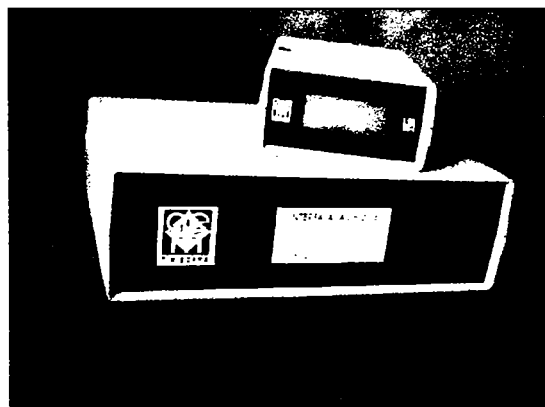


FIGURA 5.13

independent care asigură prelucrarea impulsurilor de la contorul etalon (multiplicare/divizare), transmiterea comenzilor de la calculatorul de proces spre blocul de comandă, citirea mărimilor de stare ale sistemului prin intermediul blocurilor de măsură și de intrare și comunicarea bidirecțională cu posturile de calcul și calculatorul de proces.

*Posturile de calcul* figura 5.14, calculează eroarea contoarelor aferente, pe baza semnalului de marcă primit de la contorul de etalonat

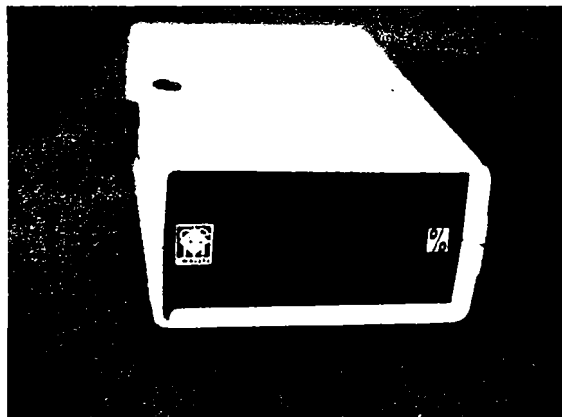


FIGURA 5.14

și a semnalului de frecvență etalon primit de la interfața de achiziție a sistemului. Posturile de calcul au și rolul de a afișa eroare contorului de etalonat.

*Sesizorii optici* figura 5.15 au rolul de a viza marca de pe discul contorului de etalonat, respectiv dispozitivului de afișare (diodă



FIGURĂ 5.15

electroluminiscentă, afișaj cu cristale lichide) în cazul contoarelor



electronice. Sesiunile constituie interfața postului de calcul și afișare a erorii cu contorul de etalonat.

\*\*\*

La acest capitol contribuțiile autorului sunt legate de conceperea schemelor electrice ale blocurilor de măsură și comandă, ale postului de calcul și afișare a erorii, cât și a interfeței de achiziție, ceea ce reprezintă concepția constructivă a întregului sistem ERRPRREX M pentru calculul și afișarea erorii contoarelor de energie electrică cu achiziția erorii la un calculator compatibil IBM PC.

Concepția funcțională a sistemului ERRPRREX M, materializată prin pachetul de programe soft realizat, reprezintă de asemenea contribuția originală a autorului.

Ca urmare a aspectului de noutate, soluțiile propuse s-au finalizat prin două brevete de invenție [B1], [B2] și un dosar pentru obținerea unui nou brevet de invenție de la OSIM [B3].

## **CAPITOLUL 6**

### **PREZENT ȘI VIITOR ÎN REALIZAREA INSTALAȚIILOR DE TESTARE A CONTOARELOR DE ENERGIE ELECTRICĂ**

Condițiile politice, economice și ecologice contribuie la acordarea unei importanțe crescânde preciziei măsurării energiei electrice în cea mai mare parte a lumii. Chiar și cel mai simplu contor poate fi etalonat într-un interval de timp din ce în ce mai scurt cu o precizie tot mai mare. Contorul modern multifuncțional, pe de altă parte, trebuie să fie testat pentru mai multe funcții.

### ***6.1 Sisteme electronice de alimentare cu putere***

Principalele părți componente ale unei instalații pentru testarea contoarelor sunt unitățile de alimentare cu putere care crează tensiunea și curentul furnizate contorului.

#### **De ce tocmai o unitate de alimentare cu putere?**

În mod convențional, unitățile cu autotransformatoare de alimentare cu putere au mai multe dezavantaje. Există o legătură directă între valorile puterii furnizate circuitelor de măsurare și cele ale alimentării publice cu putere, atât în ceea ce privește amplitudinea cât și defazajul. Variația alimentării de la rețea trebuie să fie compensată prin reglări ale circuitelor de măsurare. Într-un sistem de testare cu trei faze, această reglare permanentă poate deveni o adevărată încercare pentru răbdarea celor care o fac. O unitate de stabilizare a tensiunii poate reduce efectele datorate variațiilor, dar poate totodată influența în mod negativ simetria tensiunii.

Trebuie subliniat faptul că factorul de distorsiune în circuitul de măsurare depinde direct de rețeaua publică de alimentare. Transformatoarele stației de testare măresc factorul de distorsiune -se pot introduce nelinierități și semnale armonice.

Nu este posibilă decuplarea frecvenței circuitului de măsurare de frecvența rețelei de alimentare când se folosește o stație cu autotransformatoare pentru testarea contoarelor. Aceasta înseamnă că testarea contoarelor se poate face numai cu frecvența rețelei. În rețelele cu variații mari ale frecvenței are loc și o influențare a rezultatelor testării contoarelor, deoarece unele tipuri de contoare sunt dependente în mare măsură de frecvență.

O îmbunătățire reală a condițiilor de testare se poate realiza prin introducerea unui grup generator - motor. În acest caz influența tensiunii și frecvenței rețelei publice de alimentare este practic eliminată. Este chiar posibil, de exemplu, să fie testate contoare cu frecvența nominală de 60 Hz cu sisteme care au fost proiectate pentru testarea contoarelor cu frecvența nominală de 50 Hz.

Totuși, introducerea unui grup generator - motor înseamnă cerințe suplimentare care pot constitui adevărate probleme. În primul rând costurile pentru operare și service nu sunt mici. În al doilea rând generatorul trebuie să fie sincron cu un factor de distorsiune redus pentru a corespunde cerințelor standardelor contorului. Astfel de generatoare sunt greu de procurat, practic ele nefiind disponibile.

Cerințele IEC, referitoare la stabilitatea tensiunii la o valoare de 0,5...1%, simetria tensiunii și curentului de 0,5...2%, frecvența de 0,5...1% și factorul de distorsiune 2...3% -nu pot fi îndeplinite cu un astfel de echipament. Cerințele pot fi îndeplinite numai printr-o supraveghere permanentă a stabilizării, printr-o întreținere permanentă a transformatoarelor și a dispozitivelor de comutare, printr-o permanentă verificare a valorilor stabilite, toate acestea fiind executate de către personalul de operare, pentru a enumera doar câteva dintre aspecte.

Cea mai bună soluție este folosirea unei unități statice de alimentare cu putere.

## ***6.2 Tehnologia amplificatorului analogic***

Cea mai simplă metodă de introducere a unei unități statice de alimentare cu putere este aceea de folosire a unei unități cu amplificatori clasici analogici [30]. Astfel de amplificatori au fost mult timp folosiți în domeniul tehnologiei audio. Când privește testarea contoarelor, există niște limite. Pe de-o parte, domeniul de frecvență este de obicei cuprins între limitele 45...65 Hz, în timp ce pe de alta parte se cere o putere de ieșire foarte mare: aceasta s-ar putea situa în domeniul kilowaților. Amplificatoarele folosite la testarea contoarelor nu au o impedanță de ieșire constantă pe care să ne putem baza, deoarece atât rezistența circuitului de tensiune cât și rezistența circuitului de curent pot varia în funcție de efectele căldurii, de exemplu, sau chiar și în funcție de diversele tipuri de contoare care sunt testate.

O unitate statică de alimentare cu putere este alcătuită în esență din generatoare care crează semnale sinusoidale pentru circuitele de tensiune și curent și, de regulă, șase amplificatoare de putere, trei pentru tensiune și trei pentru curent.

Amplificatoarele analogice au câteva caracteristici care reprezintă avantaje și care contribuie la:

- marea stabilitate a valorilor de ieșire;
- linearitatea ridicată a transmisiei semnalului;
- factorul de distorsiune mică al valorilor de ieșire.

În plus acestea pot fi ușor reglate cu regulatoare convenționale, deoarece decalajul semnalului este mic. Pot fi create valori utile ale puterii de ieșire mari de câțiva kVA.

Totuși, amplificatoarele analogice au și câteva dezavantaje:

- randamentul slab, ceea ce conduce la pierderi mari prin disiparea căldurii reziduale;
- nevoia de eliminare a excesului de căldură, mai ales în țările tropicale;
- dimensiunile lor mari.

Deoarece este nevoie de spațiu pentru eliminarea căldurii, de multe ori este necesar ca instalarea unităților să se facă în camere separate. Necesitatea răcirii lor permanente adaugă un zgomot de fond mediului ambiant ceea ce deranjează personalul laboratorului și poate conduce la apariția unor erori de testare a contoarelor.

Amplificatoarele analogice au fost și sunt folosite pe scară largă cu mult succes. Dezavantajele lor au făcut necesară introducerea unor soluții noi pentru a le elimina.

### ***6.3 Tehnologia amplificatorului în comutație***

Aceste amplificatoare nu sunt atât de cunoscute ca și cele analogice și nu sunt atât de răspândite [19], [24]. Principalele motive sunt două; conceptul de comutare de dată recentă și componentele necesare, rapide și cu randament ridicat, comutatoarele electronice cu pierderi mici au fost scumpe și greu de achiziționat. În plus, datorită caracteristicilor lor de bază, dezavantajele acestora au făcut ca ele să fie dificil de utilizat. Există totuși și avantaje majore.

Cel mai important dintre acestea este randamentul deosebit de ridicat care nu este mai mic de 85%. Un amplificator analogic are în cel mai bun caz un randament de 50%, funcție de principiul

operational. Amplificatoarele în comutație se recomandă în mod special pentru utilizarea în țările tropicale pentru că nu necesită evacuarea unei cantități mari de căldură reziduală, un factor important al costului acestora. Deoarece nu mai sunt necesare elementele voluminoase de răcire, unitățile de ventilație și construcția specială pentru evacuarea aerului, aceste amplificatoare au o construcție mai compactă, permițând folosirea unor sisteme mai performante la testare. Cerințele de spațiu vor ține cont de mărimea standului de testare pentru contoare, și nu de mărimea unității de alimentare cu putere.

Zgomotul de fond, caracteristic sistemelor de ventilație forțată a amplificatoarelor analogice, este mult redus, deoarece sistemele de răcire nu trebuie să funcționeze tot timpul.

Datorită unei creșteri deosebite înregistrate de disponibilitatea componentelor electronice din domeniul comenzilor de motoare și al tehnologiei de alimentare cu putere de la rețea, sunt disponibile elemente de blocuri constructive cu performanțe ridicate pentru a fi folosite în tehnologia testării contoarelor.

O putere mare de ieșire de până la 5kVA și chiar mai mare poate fi generată fără costuri suplimentare mari de proiectare, deoarece numai ieșirea amplificatorului trebuie modificată.

Aceste avantaje importante sunt însoțite de câteva inconveniente care fac să fie mai dificilă folosirea acestui tip de amplificator.

Principiul de funcționare prin fragmentarea undei sinusoidale într-o serie de impulsuri scurte, impun utilizarea circuitelor de filtrare pentru valorile de ieșire pentru ca semnalele de ieșire să aibă forma pură de undă sinusoidală.

Datorită întârzierilor semnalului condiționat al sistemului, amplificatoarele se reglează greu cu regulatoarele convenționale. Aceasta înseamnă că un factor mic de distorsiune la nivelul valorilor de ieșire impuse mai mic de 0,8%, este posibil numai prin creșterea costurilor. În plus, aceste amplificatoare au o stabilitate scăzută la nivelul valorii de ieșire în comparație cu amplificatoarele analogice.

Avantajele amplificatoarelor în comutație pot fi evaluate când se va găsi o soluție pentru depășirea dezavantajelor.

## ***6.4 Tendințe noi în domeniul unităților electronice de alimentare cu putere***

În ultimii ani s-au înregistrat progrese considerabile pe piața componentelor. Componentele care se folosesc într-un amplificator în comutație se pot procura astăzi la prețuri rezonabile. Acest lucru se observă în special în cazul procesoarelor de semnal puternice și rapide și a convertoarelor analogice - digitale rapide de precizie. Folosirea acestor elemente face posibilă realizarea unui sistem de reglare digital în timp real [18], [23], [60].

Avantajele unor astfel de sisteme de reglare digitale sunt următoarele:

- măsurarea și reglarea cu precizie a valorilor efective și a defazajului, stabilite prin calcul și prin urmare reduc erorile mici care se găsesc în convertoarele analog - numerice. Nu este necesară multiplicarea analogică.
- reglarea cu precizie a tuturor armonicilor se poate face pentru un spectru larg de frecvență. Acest lucru are drept consecință un factor scăzut de distorsiune. Pe de altă parte, atunci când au loc operațiile speciale de testare a unui contor, se poate genera o reproducere exactă a unei forme de undă compusă ca semnal de ieșire, când este necesar să se suprapună armonici peste semnalele de bază de curent sau de tensiune sau peste ambele tipuri de semnale.
- este posibilă obținerea unor semnale stabile chiar cu amplificatoare cu întârzieri de semnal prin realizarea unor algoritmi de reglare complicați în diferite momente. Acesta este un avantaj deosebit în cazul amplificatoarelor în comutație.
- folosirea algoritmilor de reglare adaptivi, face posibilă eliminarea instabilității dependente de sarcină.

Aceste măsuri arată că toate dezavantajele tehnologiei amplificatorului în comutație pot fi depășite în totalitate.

Chiar și fără sistemul frecvent folosit de feedback al valorilor de ieșire aceste semnale pot fi generate cu mare precizie. Folosirea unui amplificator în comutație asociat cu tehnologia de reglare digitală face posibilă folosirea unei unități aproape perfecte pentru sursa de putere pentru testarea contoarelor.

Unitățile pentru furnizarea de putere de acest tip, în viitor, le vor înlocui pe cele care folosesc tehnologia analogică.

## ***6.5 Contoare etalon de referință de precizie***

Pe lângă unitatea pentru sursa de putere, contorul etalon de referință are rol decisiv în domeniul testării contoarelor [35]. În prezent, numai contoarele etalon cu domeniu larg, de obicei cu o precizie de 0,05%, sunt folosite pentru măsurări în domeniul 45...65Hz.

Cel mai mare avantaj al contoarelor etalon de referință digitale constă în faptul că acestea pot măsura toate valorile diferite aproape simultan. După transformarea semnalelor de intrare analogice de tensiune și curent într-o formă digitală, valorile dorite pot fi calculate folosindu-se un procesor digital de semnal care lucrează în timp real.

Când un contor etalon digital de referință este încorporat într-un sistem de testare a contoarelor, nu mai sunt necesare alte instrumente de măsurare, deoarece toate valorile, cum sunt, tensiunile, curenții, puterea fiecărei faze, frecvența, factorul de putere, și altele, pot fi transmise prin interfața contorului etalon de referință. Ele pot fi apoi afișate pe ecran, sau pot să fie folosite în operațiile de procesare ulterioară a datelor. Există contoare etalon de referință universale cu o precizie standard de 0,05 și 0,02%. Domeniul de măsurare al circuitelor de tensiune cuprind, în mod obișnuit, valori de la câțiva volți la 480V, între fază și neutru, și de la câțiva miliamperi la 120A în circuitele de curent. Un astfel de contor de referință nu necesită nici un transformator extern de curent sau tensiune. Acesta poate fi instalat în orice stație pentru testarea contoarelor fără cheltuieli suplimentare pentru modernizări.

Contorul standard de referință are o interfață de linie serială pentru operare software. Măsurarea erorii unui contor se poate face prin simpla conectare a unui cap de scanare la priza de intrare pentru impulsuri. O priză de ieșire pentru impulsuri face să fie disponibilă o frecvență proporțională cu puterea. Acesta mai poate fi folosit pentru verificarea contorului etalon de referință într-un institut de metrologie.

Așa-numitul comparator cu domeniu larg se folosește în prezent pentru măsurători speciale de mare precizie. Acesta are caracteristici similare cu cele ale contoarelor etalon, descrise mai sus, dar se livrează ca instrument independent, și nu ca o parte funcțională a unui sistem de testare pentru contoare. Domeniul de frecvență al acestor comparatoare este de la curent continuu și până la 2000Hz. Circuitele de intrare pentru curent ale comparatorului sunt echipate de preferință



cu șunturi, decât cu transformatoare convenționale. Această caracteristică arată că unitatea poate măsura și curentul continuu. Precizia comparatorului este aproape de 0,01%, ea fiind influențată de puterea aparentă.

## ***6.6 Programul de control***

La testarea contoarelor moderne, în special a celor electronice, folosirea ultimelor metode este obligatorie. Pentru a se asigura calitatea testării contoarelor în conformitate cu prevederile ISO 9002, sistemele de testare a contoarelor trebuie la rândul lor să poată fi verificate. Trebuie să se garanteze trasabilitatea relativ la standardul national.

Administrarea situației cu datele testării care se cumulează rapid este de mare importanță. În plus, verificarea datelor măsurării trebuie pregătită conform recomandărilor. Pentru garantarea calității testărilor este necesară o procedură de testare automată.

Având în vedere cele de mai sus, o instalație modernă pentru contoare trebuie să aibă un software “user friendly” și pentru control universal care să facă testări și etalonări automate ale tuturor tipurilor de contoare posibile. Măsurătorile sunt executate în paralel pentru toate contoarele conectate printr-un set de capuri de scanare. Aceasta se aplică atât la contoarele de tip Ferraris cât și la cele electronice. În general, un astfel de sistem de testare pentru contoare trebuie să fie prevăzut cu 10, 20, 40, 60, sau 80 de poziții pentru contoare. Programul de control răspunde de punctele de încărcare individuale în ordinea dorită, comutând contoarele de testat pentru testarea automată. La sfârșitul testării rezultatele apar tipărite sub forma unui raport de testare.

Este evident faptul că a avut loc schimbarea unei generații de la sistemele de testare pentru contoarele clasice la cele pentru testarea contoarelor electronice moderne. Tendințele actuale sunt o dovadă a faptului că are loc o permanentă îmbunătățire a caracteristicilor sistemelor electronice. Sistemele de tip modular pot fi utilizate în toate regiunile lumii și atât calitatea, costurile și cerințele de service pot fi optimizate.

## CAPITOLUL 7

### MODALITĂȚI DE VERIFICARE ȘI REZULTATELE TESTĂRIILOR INSTALAȚIILOR DE ETALONAT CONTOARE DE ENERGIE ELECTRICĂ CU ACHIZIȚIA ERORII

### **7.1 Verificarea stației SE 04**

Verificarea metrologică a stațiilor de verificat contoare de energie electrică se face după NTM 4-39-90 aprobat de Comisia Națională pentru Standarde, Metrologie și Calitate. Acest normativ a devenit obligatoriu pentru toate unitățile din activitatea economică și socială din anul 1993.

### **7.2 Rezultatele verificării stației SE 04**

Verificarea stației SE 04 a fost făcută în laboratoare de specialitate autorizate pe plan național, iar buletinul emis este prezentat în figura 7.1. Rezultatul verificării este foarte bun pentru soluția aleasă, putându-se realiza stații cu precizia de 0,2% fără utilizarea unor componente cu costuri ridicate.

### **7.3 Verificarea sistemului de calcul ERRORREX M**

Metoda de verificare a sistemului de calcul al erorii contoarelor de energie electrică ERRORREX M utilizat pentru etalonarea și verificarea contoarelor de energie electrică este prevăzută în standardul de firmă al produsului (SF 38/97) care stabilește condițiile tehnice de calitate și modul de verificare al sistemului.

La elaborarea standardului de firmă s-au folosit următoarele standarde:

- STAS 10000/6-83 Principiile și metodologia standardizării. Reguli de redactare a standardelor;
- STAS 10000/7-74 Principiile și metodologia standardizării. Reguli pentru executarea și prezentarea materialelor grafice din standarde.
- NTM 4-39-90 Verificarea instalațiilor de verificat contoare de energie electrică;
- STAS 10327/1-87 Aparat electronice de măsurat. Prescripții generale de securitate.



TIMIȘOARA  
SC AEM SA

Laboratorul de metrologie  
Entitatea tehnica nr. 1 -incercari de tip si fiabilitate

CERTIFICAT  
ISO 9001  
Nr. 9 / 1996



NR.7/14.04.1997

## BULETIN DE INCERCARE

01. PRODUSUL SUPUS INCERCARII: Instalatie de etalonat contoare de energie electrica tip SE03RR-ERRORREX
02. CARACTERISTICI :Un=220/380 V; 50-60 Hz; I=0,5 A; 1 A; 2 A; 5 A; 10 A; 20 A; 50 A; 100 A; cl.0,5%
03. PROVENIENTA : SC AEM-SA
04. SOLICITANT / COMANDA : DCD-DIECA
05. CANTITATEA : 1 buc; seria 001
06. SUBIECTUL INCERCARILOR : Incercari pentru aprobare de model
07. METODA DE INCERCARE: Conform NTM 4-39-90 "Verificarea metrologica a instalatiilor de verificat contoare de energie electrica"
08. APARATURA INTREBUINTATA : convertor putere-frecventa trifazat CET-01, seria 801/89; convertor putere-frecventa trifazat CET-01, seria 49; numarator universal E 206, nr. 894; distorsiometru BKF6-205913; aparat pentru verificarea succesiunii fazelor ZN1; stand rigiditate 2 kV
09. ERORI POSIBILE : +/-0, 2%
10. DATA PRIMIRII PRODUSULUI LA INCERCARI : 27.03.1997  
DATA INCEPERII INCERCARILOR : 27.03.1997  
DATA FINALIZARII INCERCARILOR : 7.04.1997
11. REZULTATELE INCERCARILOR :
  - verificarea conditiilor tehnice de constructie-NTM 4-39-90, pct. 3.1. -corespunde
  - verificarea conditiilor tehnice de functionare-NTM 4-39-90, pct. 3.2. -corespunde
  - verificarea preciziei generale de masurare a instalatiei-NTM 4-39-90, pct. 3.2.19. -corespunde

Tab. 1

Gama curent	cos fi	Trifazat	R	S	T
0,5 A	1 A	0,02	0,01	0,04	0,02
	0,5	0,18	0,10	0,26	0,20
1 A	1	0,05	0,04	0,04	0,08
	0,5	0,22	0,23	0,22	0,33
2 A	1	0,09	0,05	0,11	0,11
	0,5	0,29	0,19	0,50	0,41
5 A	1	0,15	0,11	0,16	0,15
	0,5	-0,01	-0,02	0	-0,05
10 A	1	0,08	0,04	0,14	0,04
	0,5	0,25	0,21	0,31	0,21
20 A	1	0,09	0	0,20	-0,01
	0,5	0,28	0,15	0,38	0,13
50 A	1	0,08	0,07	0,20	-0,03
	0,5	0,28	0,14	0,43	0,16
100 A	1	0,07	0	0,13	-0,03
	0,5	0,18	0,05	0,29	0,10

12. CONCLUZII : Instalatia de etalonare contoare de energie electrica tip SE03RR-ERRORREX a corespuns la incercarile efectuate..
13. OBSERVATII : Nu sunt.

### Buletin 7.1

Pentru verificarea sistemului trebuie efectuate cel puțin următoarele probe:

### 7.3.1 Condiții electrice

- Rezistența de izolație a interfeței de achiziție și a postului de calcul și afișare a erorii măsurată între toți pinii tuturor conectorilor legați între ei pe de o parte și carcasa aparatului pe de altă parte trebuie să fie de minimum  $20M\Omega$  în condiții de referință și  $2M\Omega$  după ciclul climatic de la punctul 2.5.

Notă: Carcasa aparatului fiind din material electroizolant, prin legarea la carcasă se înțelege legarea la o folie conductoare în care este învelit aparatul. Distanța dintre folie și conductoare trebuie să fie mai mare de 10mm, dar să nu depășească 20mm.

- Rezistența de descărcare a sarcinilor electrostatice a interfeței de achiziție, măsurată între toți pinii conectorului C1 (figura 7.1) legați între ei pe de o parte și toți pinii conectorului C2 legați între ei pe de altă parte trebuie să fie de minimum  $1M\Omega$ . Rezistența de descărcare a sarcinilor electrostatice a interfeței de achiziție, măsurată între pinii 6, 7, 8, 9 din conectorul C3 legați între ei pe de o parte (restul pinilor fiind lăsați în gol) și toți pinii din conectorul C2 legați între ei pe de altă parte trebuie să fie de minimum  $1M\Omega$ .
- Interfața de achiziție și postul de calcul și afișare a erorii trebuie să reziste fără să apară conturnări sau străpungeri la aplicarea unei tensiuni sinusoidale de 500V, 50Hz, timp de un minut, aplicarea și revenirea realizându-se progresiv, cu 100V/s, între toți pinii tuturor conectorilor legați între ei pe de o parte și carcasa aparatului pe de altă parte.

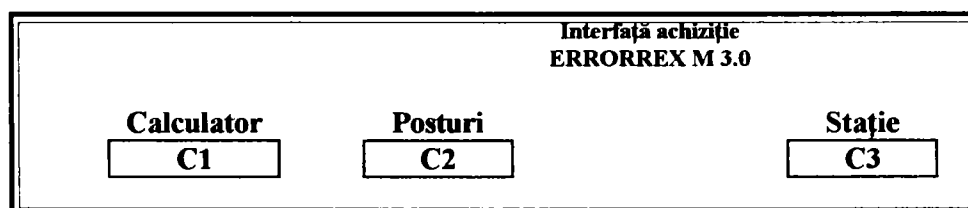


FIGURA 7.1

- Interfața de achiziție trebuie să reziste fără să apară conturnări sau străpungeri, la aplicarea unei tensiuni sinusoidale de 500V, 50Hz, timp de un minut, aplicarea și revenirea realizându-se progresiv, cu

100V/s, între toți pinii conectorului C1 (figura 7.1) legați între ei pe de o parte și toți pinii conectorului C2 legați între ei pe de altă parte. Aceeași tensiune se va aplica și între pinii 6, 7, 8, 9 din conectorul C3 legați între ei pe de o parte și toți pinii din conectorul C2 legați între ei pe de altă parte.

- Tensiunea de alimentare  $U_n$  a interfeței de achiziție și a postului de calcul și afișare a erorii este de  $12 \pm 4V$ .
- Puterea consumată în condiții de referință este de maxim 2,5W pentru interfața de achiziție și de maxim 0,4W pentru postul de calcul și afișare a erorii.
- La punerea sub tensiune, postul de calcul și afișare a erorii trebuie să afișeze caracterele: P - -.
- La comanda învățare posturi pe afișajul postului de calcul și afișare a erorii apare un mesaj de forma P nn unde nn reprezintă poziția în șina din instalație a postului de calcul și afișare a erorii. La valori ale lui nn mai mari de 99, pe afișaj apare alternativ P și nnn unde nnn reprezintă poziția în șina din instalație a postului de calcul și afișare a erorii.
- La comanda Autotest afișaj pe afișajul postului de calcul și afișare a erorii apar succesiv, cu durata de afișare de aproximativ 0,3 secunde, următoarele semne:
  - 11.11
  - + 2.22
  - 333
  - + 4.44
  - 555
  - + 6.66
  - 777
  - + 8.88
  - 999
  - + 0.00
  - - -
- Secvența se repetă până la primirea unei alte comenzi.
- La transmiterea constantei de calcul a erorii pe afișajul postului de calcul și afișare a erorii apar trei linii ( - - - ) care semnifică faptul că este în curs calculul erorii.
- Rezoluția de afișare a postului de calcul și afișare a erorii este de 0,01%.
- Domeniul erorilor care se afișează la postul de calcul și afișare a erorii este: -19,99...+19,99%.

- La postul de calcul și afișare a erorii se afișează următoarele informații suplimentare:
  - +E. semnifică depășirea erorii de +19,99%;
  - E. semnifică depășirea erorii de -19,99%;
  - LO semnifică lipsa impulsurilor de la sesizorul optic;
  - PE semnifică lipsa impulsurilor de la contorul etalon.
- Pentru verificarea interfeței de achiziție se utilizează montajul din figura 7.2. Semnificația semnalelor din conectorii C1, C2 și C3 ai interfeței de achiziție și conectorul posturilor de calcul și afișare a erorii este prezentată în tabelul 1.
- În modul de lucru cu Ieșiri separate, raportul dintre frecvența de intrare generată de contorul etalon FiR, FiS, FiT și frecvența ieșirii corespunzătoare FoR, FoS, FoT trebuie să fie o valoare din șirul: 1/64; 1/32; 1/16; 1/8; 1/4; 1/2; 1; 2; 4; 8; 16; 32; 64; 128, în domeniul de lucru al frecvențelor de intrare 15...1 500 000Hz și cu frecvența de ieșire mai mică de 25 000Hz.

TABEL 1

Pin	Interfață achiziție						Post calcul și afișare a erorii	
	C1 Calculator		C2 Posturi		C3 Stație			
1.	-	Neconectat	0V	Masă	-	Neconectat	-	Neconectat
2.	Rx	Recepție	LD	Recepție	LD	Recepție	M	Marcă
3.	Tx	Emisie	SO	Emisie	SO	Emisie	Fe	Frecvență etalon
4.	-	Neconectat	-	Neconectat	-	Neconectat	SI	Recepție
5.	GND	Masă	-	Neconectat	GND	Masă	LD	Linie date
6.	-	Neconectat	-	Neconectat	STOP <sub>R</sub>	Comandă R	-	Neconectat
7.	-	Neconectat	0V	Masă	STOP <sub>S</sub>	Comandă S	SO	Emisie
8.	-	Neconectat	-	Neconectat	STOP <sub>T</sub>	Comandă T	+12V	Alimentare
9.	-	Neconectat	-	Neconectat	COMM.	Comun +	0V	Masă
10.			-	Neconectat				
11.			+12V	Alimentare				
12.			+12V	Alimentare				
13.			+12V	Alimentare				
14.			-	Neconectat				
15.			0V	Masă				
16.			0V	Masă				
17.			-	Neconectat				
18.			-	Neconectat				
19.			Fo <sub>R</sub>	Ieșire R				
20.			-	Neconectat				
21.			-	Neconectat				
22.			Fo <sub>S</sub>	Ieșire S				
23.			-	Neconectat				
24.			-	Neconectat				
25.			Fo <sub>T</sub>	Ieșire T				

- În modul de lucru cu ieșiri însumate, raportul dintre suma frecvențelor de intrare generată de contoarele etalon  $F_{iR}$ ,  $F_{iS}$ ,  $F_{iT}$  și frecvența oricărei ieșiri  $F_{oR}$ ,  $F_{oS}$ ,  $F_{oT}$  trebuie să fie o valoare din șirul:  $1/64$ ;  $1/32$ ;  $1/16$ ;  $1/8$ ;  $1/4$ ;  $1/2$ ;  $1$ ;  $2$ ;  $4$ ;  $8$ ;  $16$ ;  $32$ ;  $64$ ;  $128$ , în domeniul de lucru al frecvențelor de intrare  $15 \dots 1\,500\,000\text{Hz}$  și cu frecvența de ieșire mai mică de  $25\,000\text{Hz}$ .
- Eroarea introdusă de sistemul de calcul în condiții de referință trebuie să fie de  $\max. 1 \times 10^{-4} \pm 1$  digit.
- Sistemul de calcul nu introduce eroare suplimentară în întreaga gamă a temperaturilor de funcționare  $+5 \dots 40^\circ\text{C}$ .
- Sistemul de calcul nu introduce eroare suplimentară în întreaga gamă a tensiunilor de alimentare.
- Blocul de măsură al sistemului de calcul, dacă este prezent, trebuie să permită afișarea pe monitorul calculatorului compatibil PC a mărimilor selectate din meniul de configurare. Tensiunile de linie sunt tensiuni calculate în cazul unui sistem trifazat.

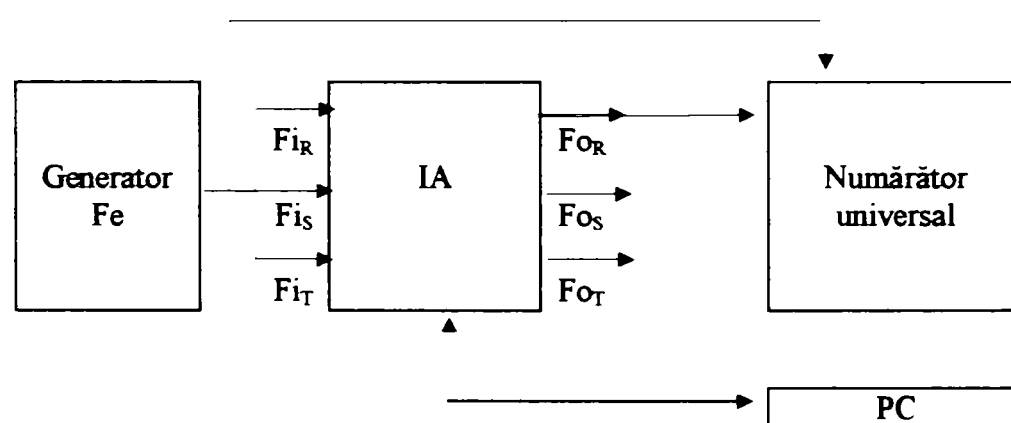


FIGURA 7.2

- Condiții de compatibilitate electromagnetică
- Sistemul de calcul trebuie să funcționeze corect în prezența unui câmp electromagnetic de înaltă frecvență.
- Sistemul de calcul trebuie să se încadreze în limitele normale ale perturbațiilor radioelectrice generate conform STAS 6048/1-87.



### 7.3.2 Condiții climatice

- Sistemul de calcul trebuie să funcționeze corect după ce a fost supus următorului ciclu climatic:
  - stocare la frig conform STAS 8393/2-77, metoda Ab, în următoarele condiții: 4h la  $-25^{\circ}\text{C}$ , în stare de nefuncționare;
  - revenire în mediu normal: 2h;
  - stocare la căldură uscată conform STAS 8393/3-78, metoda Bc, în următoarele condiții: 4h la  $+55^{\circ}\text{C}$ , în stare de nefuncționare;
  - revenire în mediu normal: 2h;
  - încercare la căldură umedă, timp de 96h la  $40^{\circ}\text{C}$ , umiditate relativă 93%, conform STAS 8393/4-81 metoda Ca, în stare de nefuncționare;
  - revenire în mediu normal: 2h.
- După ciclul climatic interfața de achiziție și postul de calcul și afișare a erorii trebuie să corespundă condițiilor tehnice prevăzute. De asemenea după aceste încercări climatice sistemul de calcul nu trebuie să prezinte deformări ale suprafețelor datorate umidității, sau temperaturii din timpul încercărilor.

### 7.3.3 Condiții mecanice

- Interfața de achiziție și postul de calcul și afișare a erorii se supun unei încercări de vibrații sinusoidale conform STAS 8393/19-89 cu următorii parametri:
  - amplitudinea vibrațiilor: 0,35mm;
  - frecvența vibrațiilor: 10...55Hz;
  - durata încercării: 30min.
- După încercarea la vibrații, interfața de achiziție și postul de calcul și afișare a erorii trebuie să corespundă condițiilor tehnice și să nu prezinte deteriorări.
- Interfața de achiziție și postul de calcul și afișare a erorii se supun, în ambalaj pentru transport, conform STAS 8393/18-90, unei probe de zdruncinări cu următorii parametri:
  - accelerația de vârf: 10g;
  - numărul de impulsuri de zdruncinare:  $1000 \pm 10$ ;
  - durata impulsului de zdruncinare: 16ms;
  - frecvența zdruncinărilor: 1...2Hz.

- După încercarea la zdruncinări, interfața de achiziție și postul de calcul și afișare a erorii trebuie să corespundă condițiilor tehnice și să nu prezinte deteriorări.

#### **7.3.4 Solicitări de durabilitate**

- Sistemul de calcul trebuie să suporte o probă de durabilitate de 96 ore în cazul probelor de tip. În cazul probelor de lot durata probei de durabilitate este de 21 ore, probă ce poate fi efectuată și sub forma a 3 x 7 ore. După efectuarea probelor de durabilitate sistemul de calcul trebuie să funcționeze corect.

#### **7.4 Rezultatele verificării sistemului de calcul ERRORREX M**

Verificările sistemului de calcul ERRORREX M au fost făcute în laboratoare autorizate, iar buletinele emise sunt prezentate în buletinul 7.2 și buletinul 7.3. Rezultatul verificărilor confirmă soluția originală creată, prin aceea că eroarea sistemului de calcul este de 0,01%.

\*\*\*

La acest capitol contribuțiile autorului sunt legate de conceperea standardului de firmă a sistemului numeric pentru calculul și afișarea erorii contoarelor de energie electrică ERRPRREX M.



TIMIȘOARA  
SC AEM SA

Laborator Metrologie

Entitatea tehnica nr. 1-incercari de tip si fiabilitate



NR. 47/20.08.1998

## BULETIN DE INCERCARE

01. PRODUSUL SŪPUS INCERCĂRII: **SISTEM DE CALCUL AL ERORII CONTOARELOR DE ENERGIE ELECTRICA, tip ERRORREX-M**

02. CARACTERISTICI :domeniul de lucru al frecventelor contorului etalon:15...1 500 000 Hz, limita superioara a domeniului frecventei de iesire dupa interfata de achiziție:cel puțin 25 000 Hz; numarul posturilor de calcul și afisare a erorii: maximum 250; constantele contoarelor de etalonat/verificat:oricare; posibilitatea contorizării unui numar programabil de Kwh:bloc de masura

03. PROVENIENTA:SC AEM-SA

04. SOLICITANT / COMANDA :Sectia a-III-a/FDEE Rm. Vâlcea

05. CANTITATEA :1 buc.interfata ,seria 120; 68 posturi de calcul, seriile conform anexa 1

06. SUBIECTUL INCERCĂRIILOR :Incercari de lot

07. METODA DE INCERCARE:Conform SF nr.38/1997 și NI 10760/20.08.1998 emisa de DCD-DIECA

08. APARATURA INTREBUINTATA:Numarator universal E 206, seria 894/85; generator de frecventa E 502, seria 7029/87; MAVO-35, seria 786/78; megohmetru de 500 V, seria 0022; statie rigiditate dielectrica, nr.0426/74;sursa I 4102, seria 3018/1986.

09. ERORI POSIBILE :+/- 1 ppm

10. DATA PRIMIRII PRODUSULUI LA INCERCĂRI :5.08.1997

DATA INCEPERII INCERCĂRIILOR :5.08.1997

DATA FINALIZĂRII INCERCĂRIILOR:20.08.1998

11. REZULTATELE INCERCĂRIILOR :

-verificare conditii constructive	pct.2.1.1	-corespunde
-verificarea functionarii	pct.2.1.3	-corespunde
-verificarea asamblării	pct.2.1.6	-corespunde
-verificarea rezistentei de izolare	pct.2.2.1	-corespunde
-verificarea rezistentei de descarcare a sarcinilor electrostatice	pct.2.2.2	-corespunde
-verificarea la strapungere	pct.2.2.3	-corespunde
-verificarea la strapungere a iesirilor izolate	pct.2.2.4	-corespunde
-verificarea domeniului tensiunii Un	pct.2.2.5	-corespunde
-verificarea la punere sub tensiune	pct.2.3.1.1	-corespunde
-invatare posturi	pct.2.3.1.2	-corespunde
-autotest post calcul și afisare erori	pct.2.3.1.3	-corespunde
-eroare in curs de calcul	pct.2.3.1.4	-corespunde
-verificarea domeniului erorilor afisate	pct.2.3.1.6	-corespunde
-verificarea afisării informatiilor suplimentare	pct.2.3.1.7	-corespunde
-verificarea raportului pentru iesiri separate	pct.2.3.2.1	-corespunde
-verificarea raportului pentru iesiri insumate	pct.2.3.2.2	-corespunde
-verificarea liniei de comunicare cu posturile	pct.2.3.3.1	-corespunde
-proba de anduranta	pct.2.7	-corespunde

12. CONCLUZII :Produsul "Sistem de calcul al erorii contoarelor de energie electrica tip ERRORREX-M" a fost corespuns la incercările efectuate.

13. OBSERVĂTII :Nu sint.

**Buletin 7.2**



MINISTERUL COMUNICATIILOR  
INSPECTORATUL GENERAL AL  
COMUNICATIILOR  
DIRECTIA TERITORIALA TIMISOARA

**BULETIN**  
**de măsurare a perturbatiilor radioelectrice**  
**Nr. 12 / 1997**

1. Denumirea..... **SISTEM DE CALCUL AL ERORII CONTOARELOR DE**  
..... **ENERGIE ELECTRICĂ tip ERRORREX-M** .....
2. Categoria..... **CLASA B** .....
3. Solicitantul măsurătorii ..... **S.C. A.E.M. S.A. TIMIȘOARA** .....
4. Scopul măsurătorii ..... **omologare serie zero** .....
5. Dispozitive și măsuri de antiparazitare prevăzute la sursa încercată .....
6. Număr de exemplare încercate ..... **=3=** .....
7. Măsurătorile au fost efectuate conform ..... **STAS 6048/1-80 și SR CISPR 22** .....
8. Locul măsurătorii : **Laborator I.G.C. - Timișoara**
  - a) Tensiuni perturbatoare .....
  - b) Câmp perturbator.....
9. Aparată de măsură și dispozitive folosite .....
- ..... **Analizor spectru TEKTRONIX 2710** .....
10. Rezultatele măsurătorilor sunt cuprinse în ANEXA 1 care face parte integrantă din prezentul Buletin.

**PRODUSUL SE INCADREAZA IN LIMITELE ADMISE IN ROMANIA DIN PUNCT  
DE VEDERE AL COMPATIBILITATII ELECTROMAGNETICE**

**NOTĂ :** Prezentul Buletin nu reprezintă un certificat de calitate în privința fiabilității produsului.

II. Produsele vor purta în mod obligatoriu inscripția :

**ANTIPARAZITAT conform STAS...../.....**

și

**ANTIPARAZITAT conform aviz MC nr...../.....**

**Buletin 7.3 (1/2)**

**REZULTATUL MĂSURĂTORILOR DE..... TENSIUNE PERTURBATOARE ȘI CÂMP PERTURBATOR.....[ dB ]**  
Determinarea nivelului perturbator caracteristic la produsul ... **SISTEM DE CALCUL AL ERORII CONTOARELOR DE ENERGIE ELECTRICALĂ tip ERORREX M**

$P = X_n + K_n * S_n < L$        $X_n = 1/n \sum X_i$        $S_n = 1 / (n-1)^{1/2} ( \sum (X_i - X_n)^2 )^{1/2}$        $n: 3 \quad 4 \quad 5 \quad 6$   
K: 2.04 1.69 1.52 1.42

Frecv. măsurare (MHz)	Armonica nr.	Frecv. (canal) recepție (MHz)	1	2	Xi; Ex: seria 3	4	5	Xn	Sn	KnSn	P[dB]	L[dB] SR CISPR 22 (clasa B)
0	1	2	3	4	5	6	7	8	9	10	11	12
NOTĂ 1 : Produsul nu prezintă tensiuni perturbatoare în gama de frecvențe 150KHz - 30MHz.												
NOTĂ 2 : Produsul nu prezintă câmp perturbator măsurabil la distanța de 10 m.												
												150KHz-500KHz 56-46 dBμV 500KHz-5MHz 46 dBμV 5MHz-30MHz 50 dBμV
												30MHz-230MHz 30 dBμV/m 230MHz-1000MHz 37 dBμV/m

**Buletin 7.3 (2/2)**

## **CAPITOLUL 8**

### **CONCLUZII**

Teza și-a propus drept obiectiv principal conceperea și realizarea (figura 8.1 și 8.2) unui echipament cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii la un calculator compatibil IBM PC.

În acest scop lucrarea a analizat în amănunțime:

- a) evoluția concepției din punct de vedere tehnic a instalațiilor de etalonat contoare de energie electrică cu achiziția erorii;
- b) modul practic de realizare a stațiilor de etalonat contoare de energie electrică;
- c) modul practic de realizare a echipamentelor de automatizare;
- d) modalități de verificare și rezultatele testărilor în laborator și la beneficiarii a stațiilor de etalonat contoare de energie electrică cu achiziția erorii.

La baza stabilirii obiectivelor au stat considerente legate de necesitățile practice ale laboratoarelor de etalonare și/sau verificare a contoarelor de energie electrică ale fabricanților de contoare cât și a beneficiarilor acestora, în principal societățile de distribuție ale ELECTRICA.

- primele instalații de etalonat și/sau verificat contoare de energie electrică erau construite în scopul verificării contoarelor prin metoda comparației cu un contor a cărui eroare era cunoscută. Instalațiile de acest gen nu s-au răspândit datorită ineficienței lor și a preciziei scăzute;
- următoarele tipuri de instalații de etalonat și/sau verificat contoare de energie electrică au fost cele care au utilizat metoda wattmetru - cronometru. Pentru utilizarea metodei respective dotarea acestora era minimală: wattmetre și cronometre;
- metoda necesita un număr mare de operatori, timp îndelungat de lucru, precizia de verificare fiind influențată negativ de subiectivismul uman;

- dezvoltarea și maturizarea cercetării în domeniul tehnologiei digitale a condus la posibilitatea introducerii microcontrolerelor în acest tip de aplicații care necesită urmărirea în timp real;
- necesitatea perfecționării instalațiilor pentru creerea unor baze de date care să asigure informații reale și obiective pentru urmărirea contoarelor de energie electrică;
- necesitatea retehnologizării instalațiilor existente cu effort financiar minim;
- necesitatea actuală de integrare a informaticii de proces în contextul sistemului informatic managerial al societăților de distribuție ale ELECTRICA.

Cele mai importante rezultate și contribuții originale prezentate de autor în legătură cu problemele teoretice și aplicative în prezenta teză sunt:

- Modul specific de realizare al schemei electrice a stației SE 04 în vederea posibilității implementării blocului de comandă și blocului de măsură și pentru obținerea performanțelor cerute de beneficiari (figura 8.1).
- Ideea de ansamblu asupra implementării echipamentelor cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii la un calculator compatibil IBM PC.
- Conceperea sistemului format din blocul de comandă și blocul de măsură al stației de etalonat contoare de energie electrică.
- Conceperea sistemului ERRORREX M pentru calculul și afișarea erorii contoarelor de energie electrică (figura 8.2).
- Realizarea schemelor electrice ale blocurilor de măsură și comandă, ale postului de calcul și afișare a erorii, cât și a interfeței de achiziție, ceea ce reprezintă concepția constructivă a întregului sistem ERRORREX M pentru



calculul și afișarea erorii contoarelor de energie electrică cu achiziția erorii la un calculator compatibil IBM PC.

- Concepția funcțională a sistemului ERRPOREX M, materializată prin pachetul de programe soft realizat.
- Conceperea standardului de firmă a sistemului numeric pentru calculul și afișarea erorii contoarelor de energie electrică ERRPRREX M.

Ca urmare a aspectului de noutate, soluțiile aplicate s-au finalizat prin două brevete de invenție [B1], [B2] și un dosar pentru obținerea unui nou brevet de invenție de la OSIM [B3].

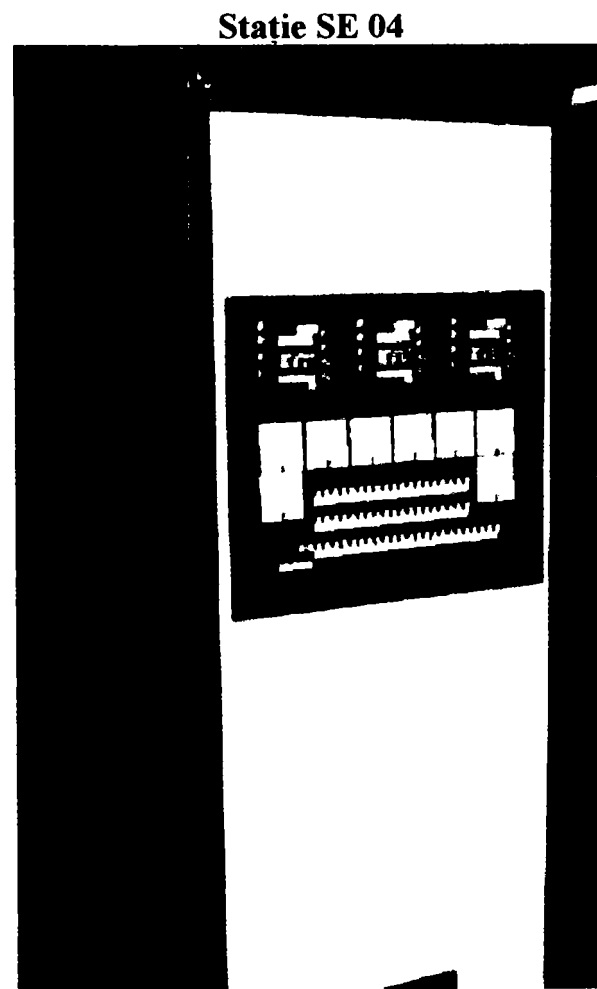


FIGURA 8.1

Sistem ERRORREX M



FIGURA 8.2

Analizând rezultatele testelor efectuate asupra echipamentului, precum și caracteristicile tehnice ale unor produse de pe plan internațional, se pot formula următoarele observații:

- firmele de prestigiu pe plan mondial au în fabricația curentă stații automate de etalonat contoare.
- s-a rezolvat modernizarea stațiilor de etalonat contoare de energie electrică care era necesară datorită momentului actual de dezvoltare tehnologică;
- soluția realizată și propusă pe plan local are avantajul costurilor reduse (aproximativ 1/4) față de un echipament din import.

Utilizând un contor electronic etalon trifazat de clasă 0,2 (EHF33) se pot verifica contoare de clasă mai mare decât unu. Pentru verificarea contoarelor de clasă mai mare decât 0,2 este nevoie de un contor electronic etalon trifazat de clasă 0,05 (DHG) cu prețul de aproximativ șapte ori mai mare (figura 8.3).

Prin utilizarea a trei contoare electronice de precizie etalon (RM 10 de clasă 0,05) corelat cu blocul de măsură, în loc de unul trifazat de aceeași clasă de precizie, (DHG) s-au redus substanțial costurile instalației fără a diminua performanțele (costurile cresc de

aproximativ două ori față de contorul de clasă 0,2). Mai mult, prin

Prețul contoarelor etalon [mii US\$]

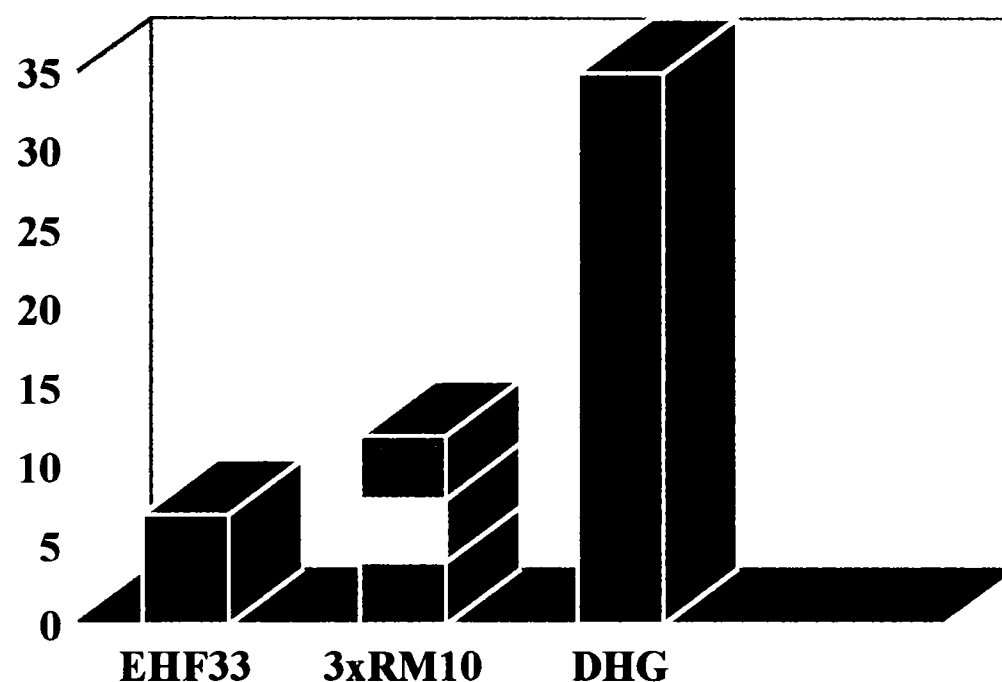


FIGURA 8.3

utilizarea celor trei contoare etalon distribuite fiecare pe câte o fază a instalației, s-a mărit productivitatea operației de etalonare de trei ori, prin posibilitatea etalonării simultan a trei grupuri de contoare monofazate conectate pe faze diferite ale instalației de etalonat, mod de lucru nerealizat de instalațiile oferite pe plan mondial.

În încheiere se menționează că prin punerea la dispoziția laboratoarelor de verificat contoare de energie electrică a echipamentului realizat, se rezolvă un aspect important al activității de verificare metrologică a contoarelor de energie electrică.

## REFERINȚE BIBLIOGRAFICE

1. \*\*\* Digital Equipment Corporation: Microcomputer Interfaces Handbook, S.U.A., 1980
2. \*\*\* Energy Measurement IC Data Book, 1996
3. \*\*\* Ganz: Pupitre d'etalonage pour ordinateurs, Ungaria, 1980
4. \*\*\* Ganz: Measuring Instruments Works, Basic Production Program, Ungaria, 1980
5. \*\*\* Plate-forme d'automatisme TSX Premium, Groupe Schneider, 1997
6. \*\*\* Industrial programmable controllers, Groupe Schneider, 1998
7. \*\*\* Programmable Microcontroller Peripherals, Philips Semiconductors, 1993
8. \*\*\* I<sup>2</sup>C Peripherals for Microcontrollers, Philips Semiconductors, 1992
9. \*\*\* Components for process automation, Rose, Phoenix Mecano Company, 1998
10. \*\*\* Radian Research, Inc. Portable Metronic Standards, Operation Manual, 1996
11. \*\*\* Nonvolatile Memory Data Book, Atmel, 1996
12. \*\*\* Texas Instruments, Data Transmission Design Seminar, 1998
13. \*\*\* Portable Data Terminals, BCP Series, TYSSO, 1998
14. \*\*\* Microcontroller, Data Book Atmel, 1997
15. \*\*\* AVR, Enhanced Risc Microcontroller, Data Book, Atmel, 1997
16. Antoniu, I. S. *Bazele electrotehnicii*, Editura Didactică și Pedagogică, București, 1974
17. Arie, A.  
Golovanov, C.  
Golovanov, Carmen  
Golovanov, N. *Poluarea cu armonici a sistemelor electroenergetice funcționând în regim permanent simetric*, Editura Academiei Române, București, 1994
18. Baggot, A. J. *The effect of wave distortion on the measurement of energy tariff meters*, IEE Conference Publication No156, Metering Apparatus and Tariffs for Electricity

- Supply, Lodon, Nov. 15-17, 1977, pag 280-284
19. Benmouyal, G. *Removal of DC-offset in current waveforms using digital mimic filtering*, IEEE Transaction on Power Delivery vol. 11, no 3, 1996, pag. 1164-1168
20. Bodea, I. *Circuite integrate liniare*, Editura Tehnică, București 1985
21. Buta, A. Milea, L. Pană, A. *Impedanța armonică a rețelelor sistemelor electroenergetice*, Editura Tehnică, București, 2000
22. Buta, A. Ivașcu, C. Milea, L. *Efectele regimului deformant asupra instalațiilor și aparatelor electrice din stațiile sistemului energetic național*, Energetica vol. 36, septembrie, București 1988, pag. 404-411
23. Czarnecki, L. S. *Comments on active power flow and energy accounts in electrical systems with nonsinusoidal waves forms and asymetry*, IEEE Transactions PWRD, vol. 11, no 3, 1996, pag. 1244-1250
24. Dekold, D. *Integrated multiplier simplifies wattmeter design*, Electronics Designer a Casebook, McGraw-Hill, S.U.A., 1976
25. Detlef, E. Schneider, G. *Metering test procedures for today and the future*, METERING INTERNATIONAL no.2, 1998, pag. 159-163.
26. Dugan, R. C. *Electrical power Systems Anality*, New York, Mc. Brow-Hill, 1996
27. Faniciu, A. Popa, M. Leșan, D. *Aparat pentru calculul și afișarea numerică a erorii contoarelor electrice de energie*, Brevet de invenție numărul 86071 din 1984
28. Faniciu, A. Stoian, A. Popa, M. *Echipament pentru calculul și afișarea numerică a erorii contoarelor electrice de energie*, Brevet de invenție numărul 83844 din 1986
29. Giacoletto, L. J. *Electronics Designers Handbook*, McGraw-Hill, S.U.A., 1977
30. Goodenough F. *Industrial electronical generator*, IEEE Transaction on Power Delivery vol. 9, no 1, 1996, pag. 814-821
31. Graeme J. *Phase compennsation on the data transmission buss*, IEEE Transaction on Power Delivery vol. 10, no 2, 1996, pag. 944-950
32. Gubisch, A. Gualgi, P. L. Miljanic, P. N. *Power calibrator using sampled feedback for curent and voltage*, IEEE Transaction on Instrumentation and Measurement, vol. 46,

- West, J. L. 1997, pag. 403-407
33. Hnatiuc, P. *Aspecte moderne ale construcției aparatelor electrice*, Conferința națională OPTIM'94 Brașov
34. Intel Intel386™EX, Embedded Microprocessor User's Manual 1995
35. Jurcă, T. *Generator numeric de tensiune alternativă*, Brevet de invenție numărul 93768 din 1987
36. Leșan D. *Modernizarea instalațiilor de etalonat contoare utilizând sistemul ERRORREX M*, Simpozion RENEL, Cluj, 1997, pag. 86-90
37. Leșan, D. *Prezentarea pachetelor de programe soft ale sistemului de etalonat contoare*, ERRORREX M, Al II-lea Simpozion de Informatică în Energetică, Deva, SIE 1998
- Pescaru, L. *Sistem numeric de comandă, reglare și măsură a generatoarelor polifazate de energie electrică pentru etalonarea contoarelor de energie electrică*, Cerere de brevet de invenție numărul C1665 din 1998
38. Leșan, D. *Sistem de calcul pentru etalonarea contoarelor*, ERRORREX, Primul Simpozion de Informatică în Energetică, Suceava, SIE 1996
- Popa, M. *Sistem de calcul a erorii contoarelor de energie electrică cu afișarea erorii și achiziție de date la un calculator PC*, Simpozion RENEL, Olimp-Neptun, 1997
39. Leșan, D. *Implementarea sistemelor automate de etalonat contoare de energie electrică*, Simpozion AEM-RENEL, Timișoara, 1997
- Popa, M. *Modernizarea instalațiilor de etalonat contoare ale F.R.E.-urilor, cu costuri minime*, Simpozion AEM-RENEL, 1998
40. Leșan, D. *Microprocessors Interfacing Techniques*, S.U.A., 1988
- Popa, M. *Internet pentru toți*, Editura Teora, 1996
41. Leșan, D. *MORE Internet For Dummies*, IDG Books, 1994
42. Leșan, D. *Microprocesoare Circuite – Proiectare*, Editura Militară, București 1986
43. Lesea, A. *Sisteme numerice cu circuite integrate*, Editura Tehnică, București 1980
- Yaks, R. *Manualul inginerului electronist vol II "Transmisii de date"*, Editura Tehnică, București, 1984
44. Levine, R. J. *Electrotehnica de la A la Z*, 1985
- Carol Baroudi
45. Levine, R. J. Margaret Lenine Young
46. Lupu, C. Stăncescu, Ș.
47. Maican, S.
48. Mateescu, A. Bănică, I. ș.a.
49. Micu, E.

50. Millea, A.  
Brătulescu, T.  
ș.a. *Contor electronic trifazat de energie electrică*, Lucrările simpozionului național de metrologie, București, 29-31 octombrie 1981
51. Millea, A. *Contoare electronice de energie electrică*, Energetica vol. 34, vol. 3, București 1987, pag. 258-263
52. Millea, A. *Măsurări electrice-principii și metode*, București, 1982
53. PHILIPS *IC20 80C51-based 8-bit microcontrollers* 1993
54. Pop, E.  
ș.a. *Tehnici moderne de măsurare*, Editura Facla 1983
55. Popa, M.  
Leșan, D. *Sistem numeric de calcul și afișare a erorii contoarelor de energie electrică cu achiziția erorii, comandat de un calculator*, Brevet de invenție numărul 112544 din 1997
56. Ramm, G.  
Moser, H.  
Braun, A. *A new scheme for generating and measuring active, reactive and apparent power at power frequencies with uncertainties of  $2.5 \cdot 10^6$* , IEEE Transaction on Instrumentation and Measurement, vol. 48, no. 2, 1999, pag. 422-426
57. Rony, P. R.  
Larse, D.  
ș. a. *Microcomputer Interfacing and Programing*, S.U.A., 1977
58. Ruegger, M. *Stations d'etalonage ETALOGYR pour energie active et reactive*, Revue Landis-Gyr, nr.1/1988
59. Sava, M.  
Toma, M. *Instalație de generat tensiuni și curenți trifazici cu defazaj programabil și independent pe fiecare fază*, Brevet de invenție numărul 102614 din 1988
60. Simonson, P.  
Svenson, S.  
Rydler, K. E. *A comparison of power measuring systems*, IEEE Transaction on Instrumentation and Measurement, vol. 46, apr. 1997, pag. 423-425
61. Șora, C. *Bazele electrotehnicii*, Editura Didactică și Pedagogică, București, 1982
62. Șora, I. *Aționări electrice și automatizări*, Editura IPT, 1979
63. Stoian, A.  
Popa, M.  
Faniciu, A. *Etalonarea și verificarea numerică a contoarelor electrice cu inducție*, Sesiune de comunicări București, 1982
64. Strugaru, C.  
Popa, M. *Microprocesoare pe 16 biți*, Editura TM 1992
65. Trifu, V.  
Vasilievici, Al. *Exemple de implementare a automatelor programabile în instalațiile electrice.*

- Realizarea unei comenzi pentru pornirea motoarelor asincrone cu rotorul bobinat, Analele Universității Oradea, 2000, vol 1, pag.75-82*
66. Turturea, D. *Programarea aplicațiilor Windows în limbajul C*, Editura tehnică, 1995
67. Vasilevici, Al. Andea, P. *Aparate și echipamente electrice*, Editura ~~Orizont~~ *Orizont* uniuersitară, Timișoara 2000
68. Vasilevici, Al. Frigură, F. *Exemple de implementare a automatelor programabile în instalațiile de comandă ale marilor consumatori industriali*, Analele Universității Oradea, 2000, vol 1, pag.56-64
69. ZERA ELECTRIC Catalog 1995

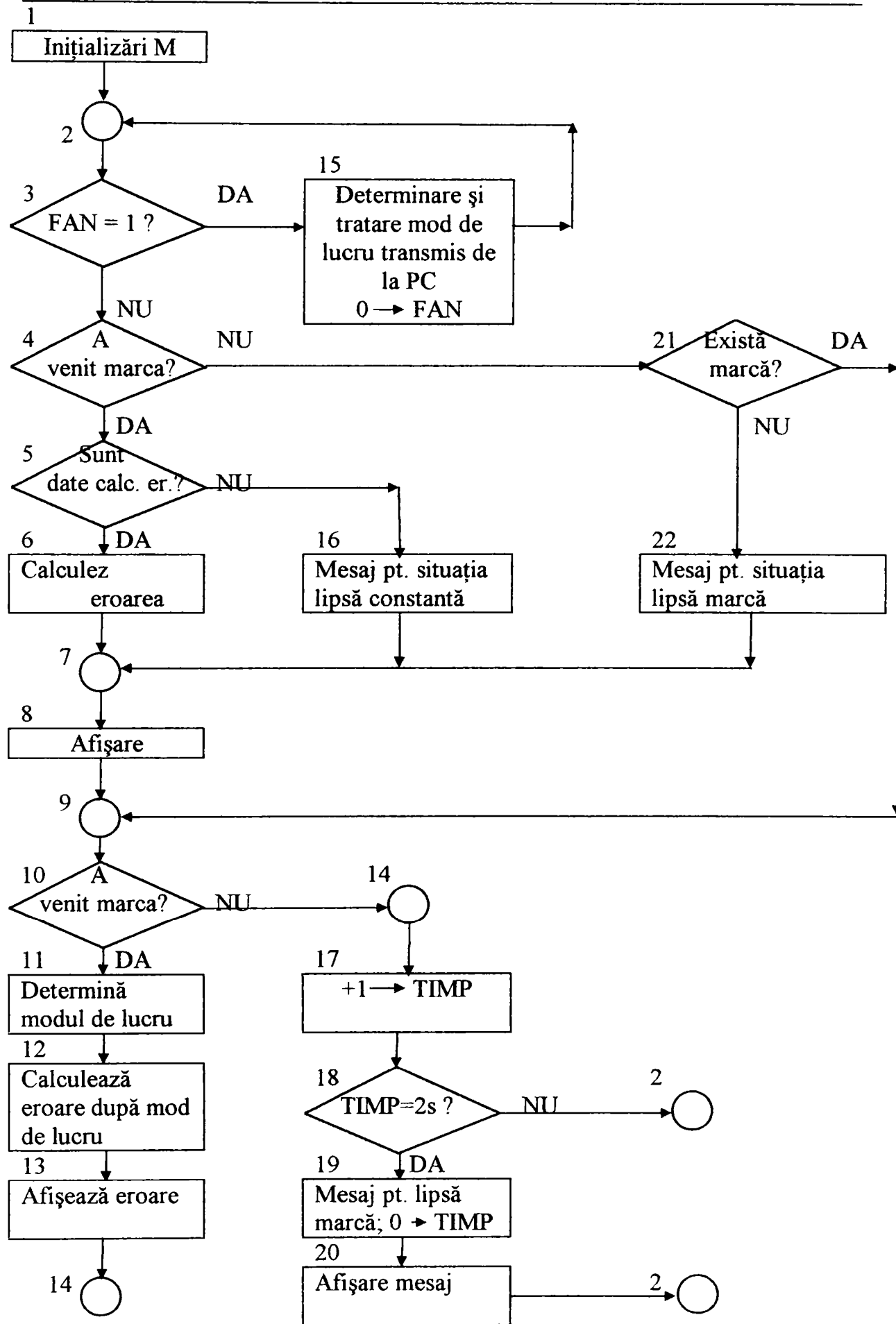


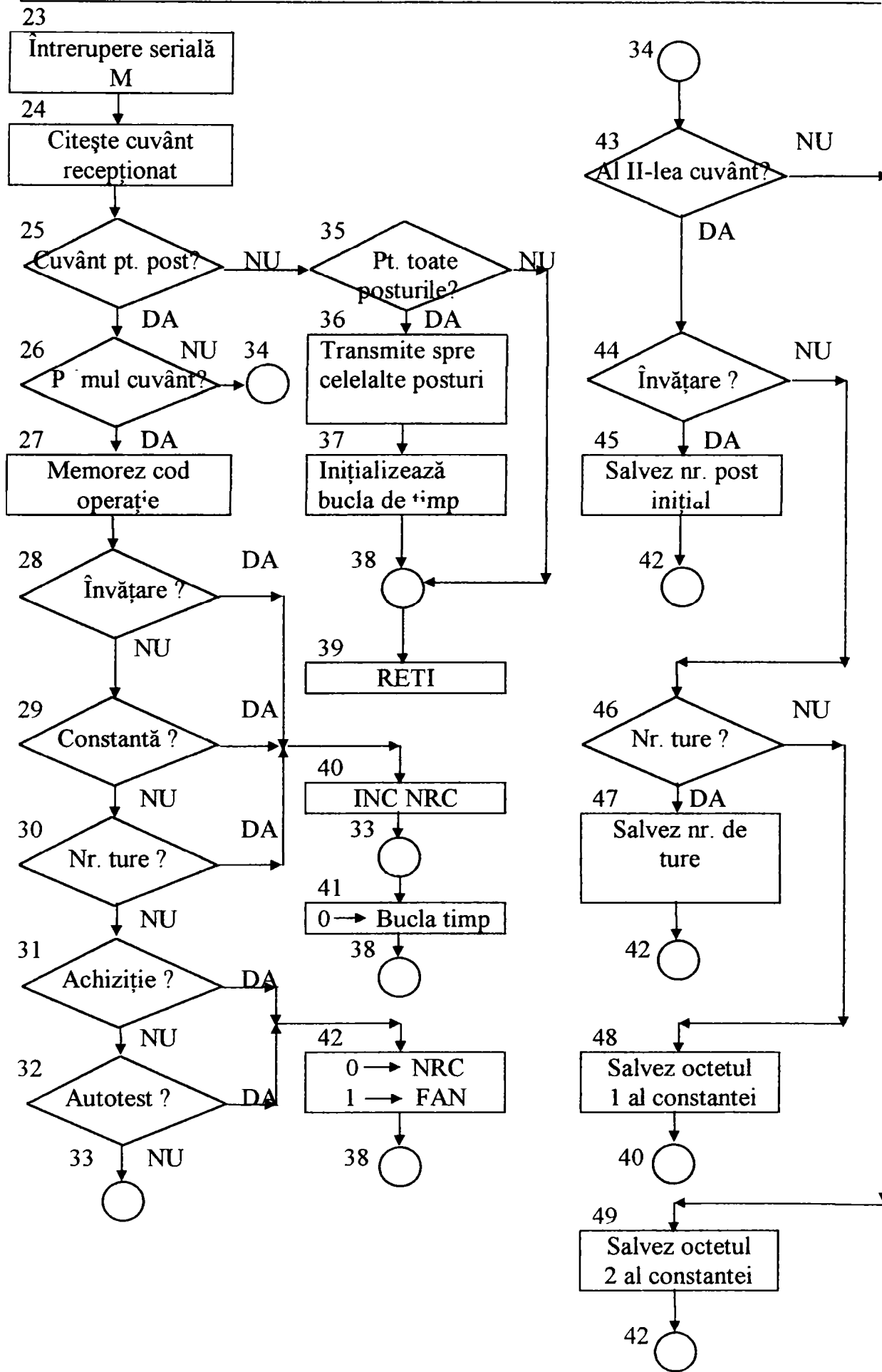
### ***Brevete***

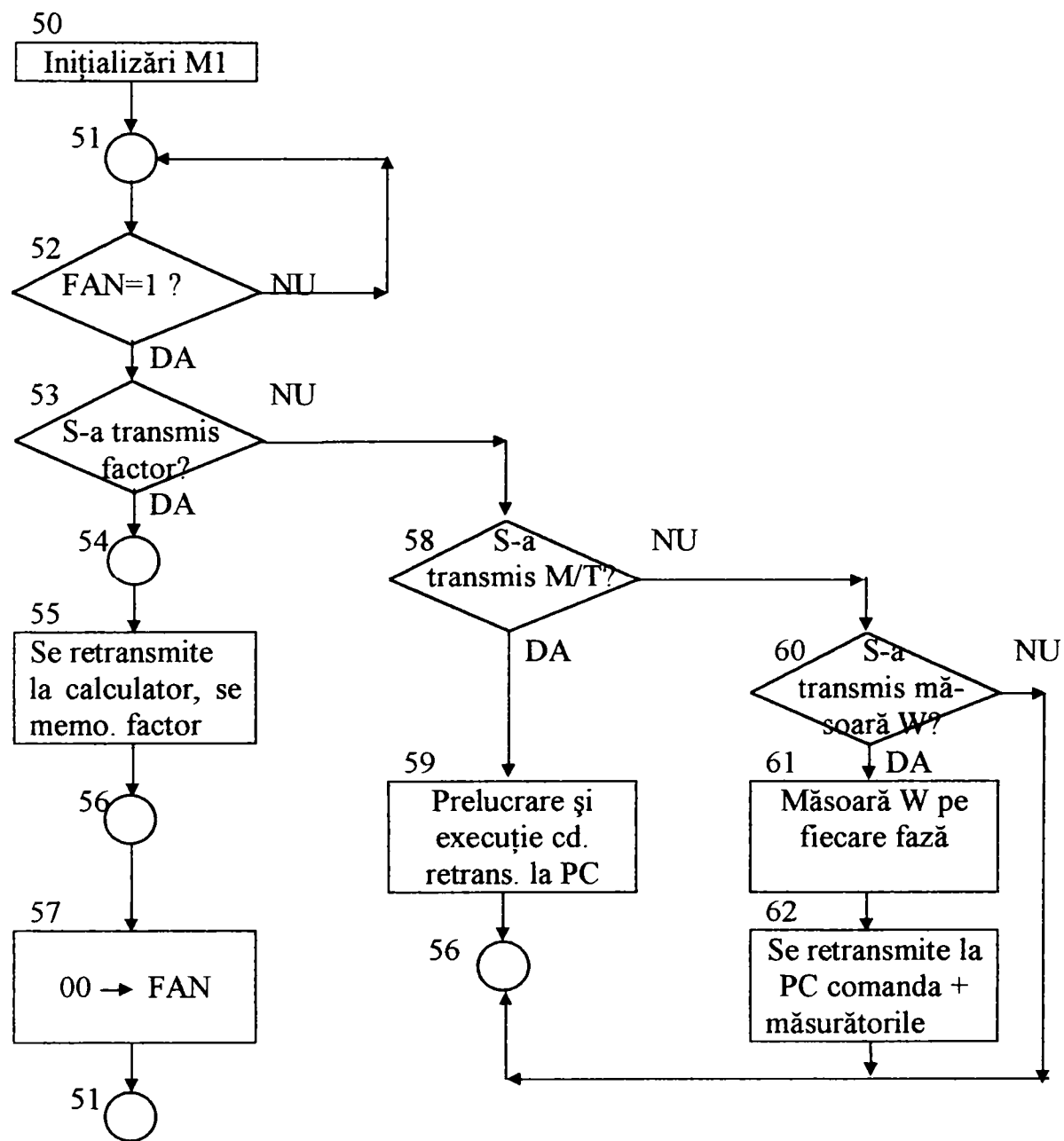
- |    |  |   |
|----|--|---|
| B1 | Aristide Faniciu,<br>Marcela Popa,<br>Doru Leșan | "Aparat pentru calculul și afișarea numerică a erorii contoarelor electrice de energie"<br>Brevet de invenție numărul 86071 din 1984  |
| B2 | Marcela Popa,<br>Doru Leșan                      | "Sistem numeric de calcul și afișare a erorii contoarelor de energie electrică cu achiziția erorii, comandat de un calculator"<br>Brevet de invenție numărul 112544 din 1997                              |
| B3 | Doru Leșan,<br>Marcela Popa                      | "Sistem numeric de comandă, reglare și măsură a generatoarelor polifazate de energie electrică pentru etalonarea contoarelor de energie electrică"<br>Cerere de brevet de invenție numărul C1665 din 1998 |

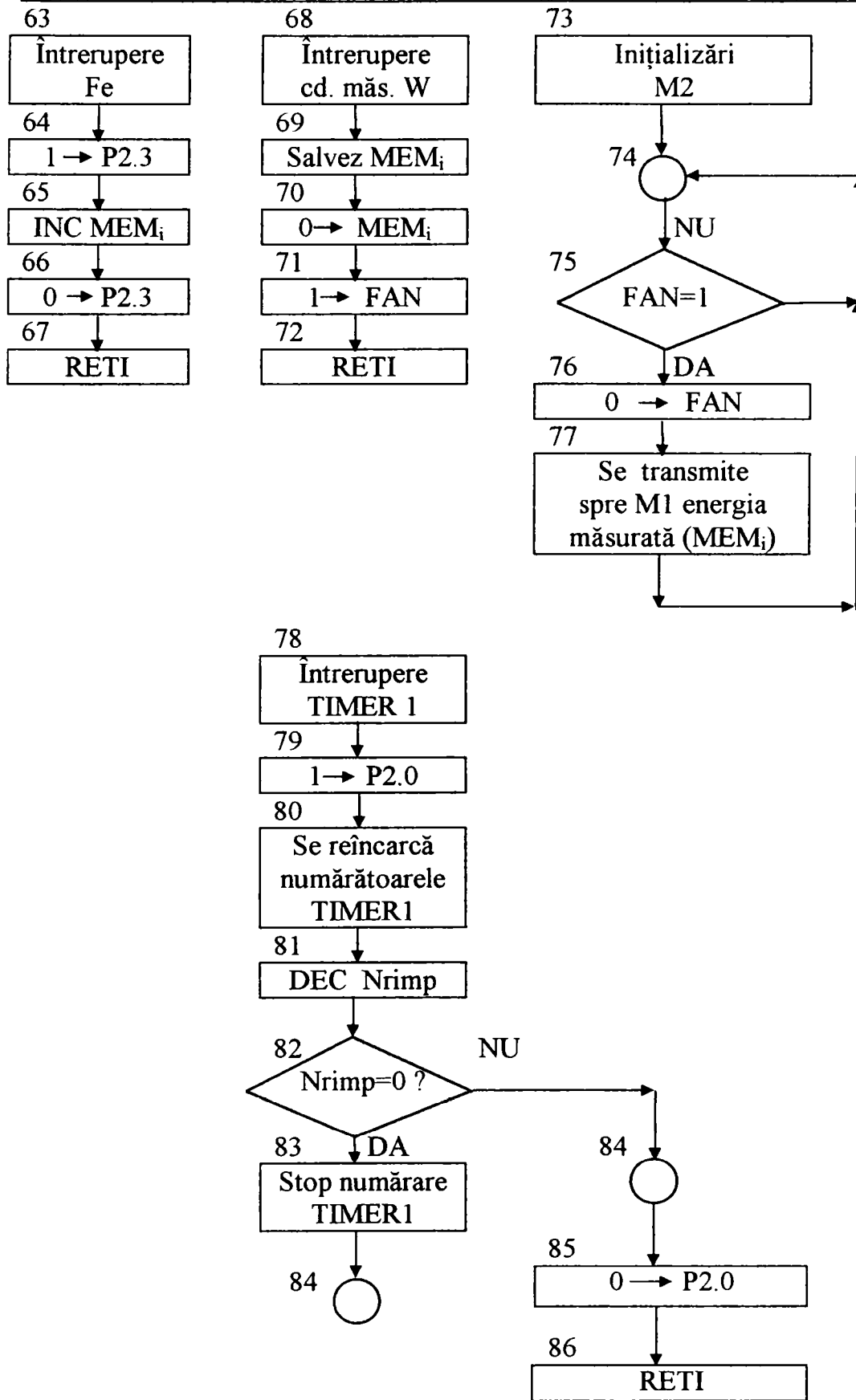
## ANEXE

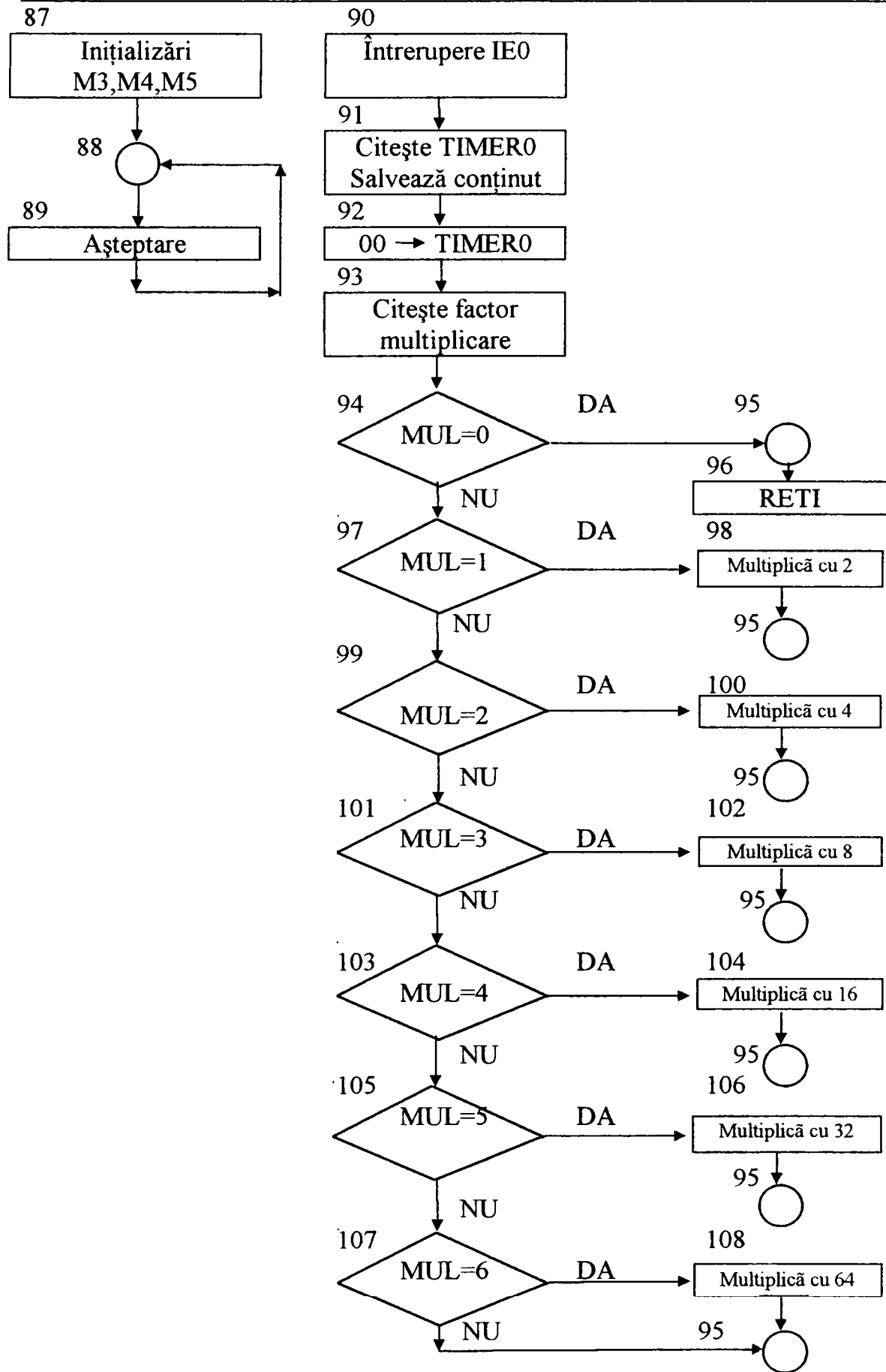
<i>ANEXA 1</i> Ordinograme ERRORREX M	A1
<i>ANEXA 2</i> Program post calcul ERRORREX M	A7
<i>ANEXA 3</i> Program intrefață achiziție ERRORREX M (1/3)	A18
Program intrefață achiziție ERRORREX M (2/3)	A21
Program intrefață achiziție ERRORREX M (3/3)	A23
<i>ANEXA 4</i> Program BLOC DE MĂSURĂ ERRORREX M	A29
<i>ANEXA 5</i> Program BLOC DE COMANDĂ SE 04	A40
<i>ANEXA 6</i> Program calculator	A54
<i>ANEXA 7</i> Schema electrică a stației de etalonat contoare de energie electrică SE 04	A117

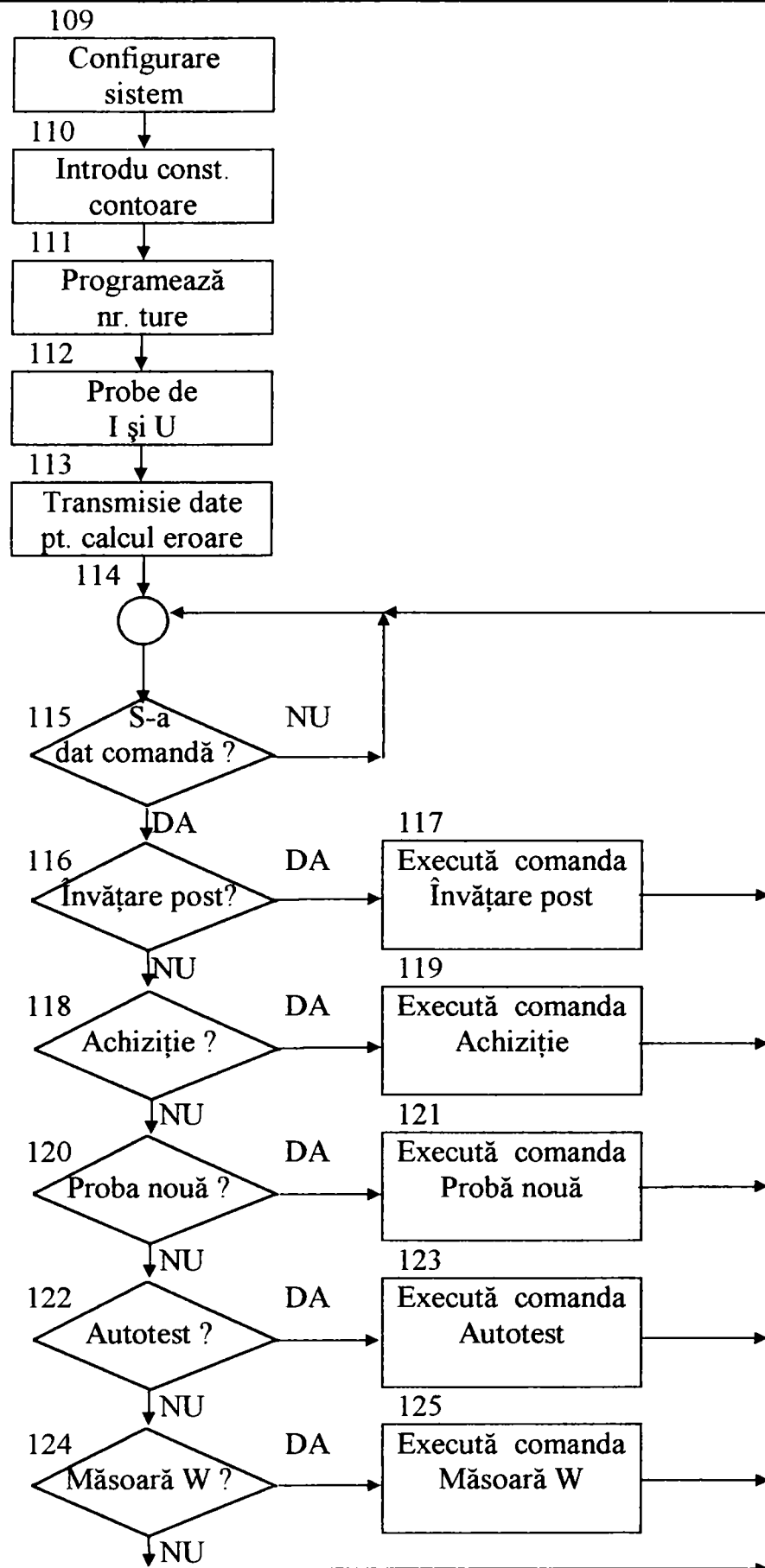














			CONT9	EQU	062H
			CONT10	EQU	063H
; Program post calcul					
ERRORREX-M					
; Tabela cu adrese utilizate					
CIT1	EQU	030H	INVAT	EQU	011H
CIT2	EQU	031H	ACHIZ	EQU	022H
CONT0	EQU	032H	CONST	EQU	033H
CONT1	EQU	033H	AFIS	EQU	044H
CONT2	EQU	034H	MONIT	EQU	055H
CONT3	EQU	035H	TURE	EQU	066H
CONT4	EQU	036H	FILTRU	EQU	077H
CONT5	EQU	037H	NRMARCI	EQU	088H
CONT6	EQU	02EH			
FAN	EQU	02FH	ORG	0	
AdrSemn	EQU	038H	LJMP	INCEP	
AK1	EQU	039H	ORG	3	
AK2	EQU	03AH	LJMP	INEXT	
CIT1S	EQU	03BH	ORG	0BH	
CIT2S	EQU	03CH	LJMP	INTIM0	
AdrNP	EQU	03DH	ORG	023H	
NRC	EQU	03EH	LJMP	INSER	
CUV1	EQU	03FH	; program principal		
CUV2	EQU	040H	INCEP:		
CUV4	EQU	041H	MOV	SP,#20H	
CUV5	EQU	042H	; programare stiva		
CUV	EQU	043H	MOV	IE,#10H	
TAB1	EQU	044H	; prog. întreruperi IEO și RI&TI		
TAB2	EQU	045H	MOV	IP,#01H	
TAB3	EQU	046H	; prog. priorități întreruperi		
TAB4	EQU	047H	MOV	TCON,#71H	
CONTPC	EQU	048H	; într. IEO activă pe frontul căzător		
CONTSR	EQU	049H	MOV	TMOD,#25H	
CONSEC	EQU	04AH	; programare timer		
CONAFI	EQU	04BH	MOV	TL1,#0FDH	
Adr1	EQU	04CH	; TIMER1-timer, mod 2 (2400		
Adr2	EQU	04DH	MOV	TH1,#0FDH	
Adr3	EQU	04EH	; TIMER0-counter, mod 1		
Adr4	EQU	04FH	MOV	SCON,#50H	
Adr5	EQU	050H	; programare port serial în mod 1		
Adr6	EQU	051H	; începe zona de inițializare a mem. pt.		
Adr7	EQU	052H	necesitățile programului		
Adr8	EQU	053H	MOV	00CH,#00	
TAB5	EQU	054H	MOV	00DH,#00	
TAB6	EQU	055H	MOV	CIT1,#00	
NUM	EQU	056H	MOV	CIT2,#00	
AFIS1	EQU	057H	MOV	CONT0,#00	
AFIS2	EQU	058H	; indică venirea unei mărci		
AdrTU	EQU	059H	MOV	CONT1,#00	
IMP	EQU	05AH	; numărător de două marci		
CUV6	EQU	05BH	MOV	CONT2,#00	
AdrTUI	EQU	05CH	; pt. K2,K1=0 (#01), lipsă Fe (#02),		
AdrFIL	EQU	05DH	lipsă marcă (#03)		
AdrMARC	EQU	05EH	MOV	CONT5,#00	
CONT7	EQU	05FH	; contoare pt. 0,5s la afișare		
BAK	EQU	060H	MOV	CONT6,#00	
CONT8	EQU	061H	; adresă Bit Adressable		

<pre> MOV CONT7,#00 ; contoriz. timp determinare marcă MOV CONT8,#00 MOV CONT9,#00 ; contoriz. 2s MOV CONT10,#00 ; contoriz. 2ms CLR P2.2 MOV AdrMARC,#00 MOV FAN,#00 ; adresă Bit Adressable MOV AdrSemn,#00 MOV AK1,#00 ; mem. constanta probei MOV AK2,#00 MOV CIT1S,#00 ; mem. info. pt. transfer serial MOV CIT2S,#00 MOV AdrNP,#00 ; mem. nr. post MOV NRC,#00 ; contor de cuvinte recepționate MOV CUV1,#00 ; mem. cuvinte pt. transm. spre MOV CUV2,#00 ; PC și spre postul următor MOV CUV4,#00 MOV CUV5,#00 MOV CUV,#00 ; mem. cuvânt recepționat MOV CONTPC,#00 ; nr. cuvinte transmise la PC MOV CONTSR,#00 ; nr. cuvinte transmise la postul următ. MOV CONSECR,#00 ; contori de timp pt. afișaj MOV CONAFI,#00 MOV NUM,#00 MOV AdrTU,#01H ; mem. nr. ture la care se calc. eroarea MOV AdrTUI,#01H MOV AdrFIL,#OFFH ; determină timp pt. citire marcă MOV AFIS2,#0FEH ; pe afișaj, inițial - _ _ _ MOV AFIS1,#0AAH MOV TAB2,#0FEH ; mem. info. pt. afișaj MOV TAB1,#0AAH MOV TAB4,#0FEH MOV TAB3,#0AAH MOV TAB6,#0FEH MOV TAB5,#0AAH ACALL AFISAJ SETB IE.7 </pre>	<pre> E9: MOV A,CONT0 INC A JZ E8 ; dacă a venit marca, CONT0=FF MOV A,CONT7 ; a venit marca ? CLR C SUBB A,AdrFIL JZ E8M MOV CONT7,#00 ; a trecut timpul MOV CONT0,#00 JB P2.0,E8 ; a venit marca? MOV CONT1,#00 ; anulez contor de mărci INC AdrMARC ; da, a venit marca DJNZ AdrTU,E8 MOV AdrTU,AdrTUI ; reinit. nr. TURE SETB PSW.3 ; calculez delta T-&gt;R7,6 CLR C MOV A,R0 SUBB A,R4 MOV R6,A MOV A,R6 SUBB A,R5 MOV R7,A MOV A,CONT1 JC IN1 CJNE A,#01,IN1 MOV A,CONT2 CJNE A,#04,IN2S AJMP IN2 IN2S: MOV CONT2,#05 AJMP IN2 IN1: CJNE A,#02,IN2 MOV A,CONT2 CJNE A,#04,IN1S AJMP IN2 IN1S: MOV CONT2,#05H IN2: MOV A,R0 ; reactualizez val. veche în R5,4 MOV R4,A MOV A,R6 MOV R5,A CLR PSW.3 ; începe bucla de calcul a erorii în următoarele etape: K-DELTAT=X--&gt;R3,R2 ; X*100=Y ; Y/DELTAT cu rotunjire MOV A,CONT2 CJNE A,#04,IN5 AJMP ES2 IN5: CJNE A,#05,IN3 </pre>
--	---

	AJMP ES2	ACALL SUB10
IN3:	CJNE A,#06,IN4	; calc. (K-DeltaT)*10000
	AJMP ES2	ACALL SUB10
IN4:	MOV A,AK1	; Algoritmul de înmulțire:
	; pt.K=0, nu se calc. eroarea	ACALL SUB10
	ORL A,AK2	;
	MOV CONT2,#01	[(a*2*2+a)*2*2*2+(a*2*2+a)*2]*2
	JZ X3	ACALL SUB10
	AJMP ES2	; împărțirea unui nr. pe 3 octeți (Adr3,2,1) la
E8M:	INC CONT7	un nr. pe 2 octeți (R7,R6) cu rez. la
E8:	MOV A,#03	CIT2,CIT1
	ANL A,CONT1	MOV IMP,#00
	CJNE A,#03,X4	; dacă IMPARTITORUL > 32768,
	; dacă CONT1=02-->lipsă marcă	se
	MOV CONT2,#03	MOV A,R7
	MOV CONT1,#00	; împarte la 2 și se memorează FF
	AJMP E11	RLC A
X4:	AJMP E52	; la adresa IMP, urmând ca la
X3:	MOV CONT2,#00	sfârșitul
	ACALL ADUDT	JNC ES1
	MOV A,R7	; împărțirii să se împartă câtul cu 2
	ORL A,R6	CLR C
	JZ E10	MOV A,R7
	MOV CONT2,#02	RRC A
	; indică lipsa Fe	MOV R7,A
	AJMP ES2	MOV A,R6
E10:	MOV AdrSemn,#00	RRC A
	; începe calcul eroare	MOV R6,A
	MOV CIT1,#00	MOV IMP,#0FFH
	MOV CIT2,#00	ES1: MOV CONT3,#10H
	CLR C	; calculează Y/DT
	; calculează K - DeltaT (DT)	E14: MOV R3,Adr4
	MOV A,AK1	MOV R2,Adr3
	; cu rezultat la R3,R2	E13: ACALL INCCIT
	SUBB A,R6	ACALL DIF
	MOV R2,A	JNC E13
	MOV A,AK2	ACALL SUMA
	SUBB A,R7	ACALL DECCIT
	MOV R3,A	MOV Adr4,R3
	JNC E12	MOV Adr3,R2
	MOV AdrSemn,#0FFH	ACALL ROTL
	; dacă diferența <0, rezultatul se	ACALL RCITL
	complementează (c2) MOV A,R2	DJNZ CONT3,E14
	; și se memorează FF la	MOV R3,Adr4
	AdrSemn	MOV R2,Adr3
	CPL A	SUP1: ACALL INCCIT
	ADD A,#01	; dacă restul e > ca împărțitorul
	MOV R2,A	ACALL DIF
	MOV A,R3	JNC SUP1
	CPL A	ACALL SUMA
	ADDC A,#00	ACALL DECCIT
	MOV R3,A	CLR C
E12:	MOV Adr4,#00	; în R3,R2 rest, în R7,R6 împărțitor
	MOV Adr3,#00	MOV A,R7
	MOV Adr2,R3	RRC A
	MOV Adr1,R2	; se execută R7,R6-R3,R2

<pre> XCH  A,R3 ; dacă împ/2 &gt; rest (C=0) nu se face rotunjirea MOV  R7,A MOV  A,R6 RRC  A XCH  A,R2 MOV  R6,A ACALL DIF JNC  E11 MOV  A,#01 ; se execută rotunjirea rez. +1 la CIT2,CIT1 ADD  A,CIT1 MOV  CIT1,A MOV  A,#00 ADDC A,CIT2 MOV  CIT2,A E11: MOV  A,IMP       CJNE A,#0FFH,ES2       CLR  C ; împărțitorul &gt; 32768, se face corecția câtului MOV  A,CIT2 ; (împărțire cu 2) RRC  A MOV  CIT2,A MOV  A,CIT1 RRC  A MOV  CIT1,A ES2: ACALL CONVBCD ; eroarea la CIT2,CIT1 se convertește în binar ACALL AFISAJ E15: AJMP  E52 ; Subrutina tratează întreruperea externă (marca este și la intrarea P2.0) INEXT: PUSH 0E0H       PUSH 0D0H       MOV  CONT0,#0FFH       SETB PSW.3 E1:  MOV  R0,TL0       ; citesc val. TIMER0       MOV  R1,TH0       MOV  R2,TL0       MOV  R3,TH0       CLR  C       ; verifică citire corectă       MOV  A,R0       SUBB A,R2       JZ   E1       CLR  C       MOV  A,R6       SUBB A,R3       JZ   E1       CLR  PSW.3 </pre>	<pre> POP  0D0H POP  0E0H RETI ; Subrutina tratează întreruperea serială INSER: JBC  SCON.0,ER       RETI ER:    ACALL RECEPT       RETI ; Subrutina transmite serial la PC și postul următor cuvinte de la CIT1S,CIT2S TRANSE:       PUSH 0E0H       PUSH 0D0H       MOV  SBUF,CIT1S E5:   JBC  SCON.1,E4       AJMP E5 E4:   MOV  A,CONTSR       CJNE A,#03,E6       MOV  CONTSR,#01       AJMP E7 E6:   MOV  SBUF,CIT2S X8:   JBC  SCON.1,E7       AJMP X8 E7:   POP  0D0H       POP  0E0H       RET ; Subrutina tratează întreruperea de la TIMER0 la overflow INTIM0:       INC  CONT1       RETI ; Subrutina realizează rotirea la stânga cu 1 rang a unui nr. de 4 octeți aflat la Adr4,3,2,1 ROTL: CLR  C       MOV  A,Adr1       RLC  A       MOV  Adr1,A       MOV  A,Adr2       RLC  A       MOV  Adr2,A       MOV  A,Adr3       RLC  A       MOV  Adr3,A       MOV  A,Adr4       RLC  A       MOV  Adr4,A       RET ; Subrutina efectuează înmulțirea cu 10 a unui nr. de la Adr4,3,2,1 cu rezultat la Adr4,3,2,1 SUB10: MOV  Adr5,Adr1       ; nr. se salvează la Adr8,7,6,5 </pre>
--	--

```

MOV  Adr6,Adr2          RET
MOV  Adr7,Adr3
MOV  Adr8,Adr4          ; Subrutina efectuează +1 la CIT2,CIT1
ACALL ROTL              INCCIT:  MOV  A,CIT1
ACALL ROTL              ADD   A,#01
MOV  A,Adr1             MOV  CIT1,A
ADD  A,Adr5             MOV  A,CIT2
MOV  Adr1,A             ADDC  A,#00
MOV  A,Adr2             MOV  CIT2,A
ADDC A,Adr6             RET
MOV  Adr2,A
MOV  A,Adr3             ; Subrutina efectuează -1 la CIT2,CIT1
ADDC A,Adr7             DECCIT:
MOV  Adr3,A             CLR   C
MOV  A,Adr4             MOV  A,CIT1
ADDC A,Adr8             SUBB  A,#01
MOV  Adr4,A             MOV  CIT1,A
ACALL ROTL              MOV  A,CIT2
RET                     SUBB  A,#00
                        MOV  CIT2,A
                        RET

; Subrutina realizează o buclă de timp de
10ms
MS10: MOV  CONT3,#0AH   ; Subrutina adună împărțitorul la deîmpărțit
E16:  ACALL MS1         R3,R2+R7,R6
      DJNZ  CONT3,E16   SUMA: MOV  A,R2
      RET              ADD  A,R6
                        MOV  R2,A
                        MOV  A,R3
                        ADDC  A,R7
                        MOV  R3,A
                        RET

; Subrutina realizează o buclă de timp de
1ms
MS1:  MOV  CONT4,#0F9H
E17:  NOP
      NOP
      DJNZ  CONT4,E17
      RET

; Subrutina transferă info. din al II-lea set de
regiștri în I set
ADUDT:
      SETB  PSW.3
      MOV  A,R7
      CLR  PSW.3
      MOV  R7,A
      SETB  PSW.3
      MOV  A,R6
      CLR  PSW.3
      MOV  R6,A
      RET

; Subrutina realizează scăderea numerelor
din R3,R2-R7,R6
DIF:  CLR  C
      MOV  A,R2
      SUBB  A,R6
      MOV  R2,A
      MOV  A,R3
      SUBB  A,R7
      MOV  R3,A

; Subrutina efectuează rotirea câtlui cu 1
rang la stânga
RCITL: CLR  C
      MOV  A,CIT1
      RLC  A
      MOV  CIT1,A
      MOV  A,CIT2
      RLC  A
      MOV  CIT2,A
      JNC  RC1
      MOV  CONT2,#06H
      ; eroari pozitive mari +E.
RC1:  RET

; Subrutina realiz.afișarea erorii de la
AFIS2,AFIS1, semnul este pus pe c.m.s. bit
al AFIS2, 1 pt. <0
AFISAJ: MOV  A,CUV6
      CJNE  A,#AFIS,S1
      ; dacă cd. AFIS se afișează
următoarea secvență:
      ACALL AFTEST
S1:   MOV  A,AFIS1
      ; -11.11

```

Asupra implementării echipamentelor cu logică programată pentru comanda unei stații de  
 etalonat contoare de energie electrică cu achiziția erorii - ANEXA 2  
 Program post calcul și afișarea erorii ERRORREX M

ANL	A,#0FH	CPL	P1.6
; + 2.22		RET	
MOV	P1,A		
; - 3 33		AFTEST:	
MOV	P1,A	MOV	A,NUM
; + 4.44		CJNE	A,#00,S2
NOP		S14:	MOV AFIS1,#11H
CPL	P1.6	MOV	AFIS2,#31H
; - 5 55		S4:	MOV TAB2,AFIS2
NOP		MOV	TAB4,AFIS2
; + 6.66		MOV	TAB6,AFIS2
NOP		MOV	TAB1,AFIS1
; - 7 77		MOV	TAB3,AFIS1
CPL	P1.6	MOV	TAB5,AFIS1
; + 8.88		INC	NUM
MOV	A,AFIS1	RET	
; - 9 99		S2:	CJNE A,#01,S3
ANL	A,#0F0H	MOV	AFIS1,#22H
; + 0.00		MOV	AFIS2,#02H
CLR	C	AJMP	S4
; --		S3:	CJNE A,#02,S5
RRC	A	MOV	AFIS1,#33H
RRC	A	MOV	AFIS2,#13H
RRC	A	AJMP	S4
RRC	A	S5:	CJNE A,#03,S6
MOV	P1,A	MOV	AFIS1,#44H
ORL	A,#10H	MOV	AFIS2,#04H
MOV	P1,A	AJMP	S4
NOP		S6:	CJNE A,#04,S7
CPL	P1.6	MOV	AFIS1,#55H
NOP		MOV	AFIS2,#15H
NOP		AJMP	S4
CPL	P1.6	S7:	CJNE A,#05,S8
MOV	A,AFIS2	MOV	AFIS1,#66H
ANL	A,#0FH	MOV	AFIS2,#06H
MOV	P1,A	AJMP	S4
ORL	A,#20H	S8:	CJNE A,#06,S9
MOV	P1,A	MOV	AFIS1,#77H
NOP		MOV	AFIS2,#17H
CPL	P1.6	AJMP	S4
NOP		S9:	CJNE A,#07,S10
NOP		MOV	AFIS1,#88H
CPL	P1.6	MOV	AFIS2,#08H
MOV	A,AFIS2	AJMP	S4
ANL	A,#0F0H	S10:	CJNE A,#08,S11
CLR	C	MOV	AFIS1,#99H
RRC	A	MOV	AFIS2,#19H
RRC	A	AJMP	S4
RRC	A	S11:	CJNE A,#09,S12
RRC	A	MOV	AFIS1,#00H
MOV	P1,A	MOV	AFIS2,#00H
ORL	A,#30H	AJMP	S4
MOV	P1,A	S12:	CJNE A,#0AH,S13
NOP		MOV	AFIS1,#0AAH
CPL	P1.6	MOV	AFIS2,#0FAH
NOP		AJMP	S4
NOP		S13:	MOV NUM,#00H

```

AJMP S14
; Subrutina conv. eroarea de la CIT2.CIT1,
cu semn la AdrSemn în BCD
; rezultatul este la adr. CIT2,CIT1, cu semn
pe bit c.m.s
; face pregătirea info. de transmis la
comanda de achiziție
; eroare de transmis la CIT2S,CIT1S,
bit16=1 pt. er.<0
;
bit15=1 pt. transm. cod eroare
; (Fe,Co)
bit14=1 pt. transm. er.>0
; codurile de eroare sunt: -Fe--> - PE
; -Co--> - L0
; -Er--> -E__
; +Er--> +E__
; măsurare în curs--> _ - - -

CONVBBCD:
MOV A,CONT2
JZ E22
CJNE A,#01,E23
AJMP E24
; afișaj neschimbat (K1,2=0)
E23: CJNE A,#02,E25
MOV AFIS2,#1FH
; lipsește Fe
MOV AFIS1,#0EBH
MOV CIT2S,#40H
; pt. achiziție
MOV CIT1S,#01H
AJMP E24X
E25: CJNE A,#03,E26
MOV AFIS2,#1FH
; lipsește marca
MOV AFIS1,#0D0H
MOV CIT2S,#40H
; pt. achiziție
MOV CIT1S,#02H
AJMP E24X
E26: CJNE A,#04,X5
MOV CONT2,#00
MOV AFIS2,#0FAH
; măsurare în curs
MOV AFIS1,#0AAH
MOV CIT2S,#40H
; pt. achiziție
MOV CIT1S,#05H
AJMP E24X
X5: CJNE A,#05,X5S
MOV CONT2,#00
MOV CIT2S,#0E0H
; TIMER0 s-a dat peste cap de 2 ori
MOV CIT1S,#06H
MOV AFIS2,#7BH
MOV AFIS1,#0FFH
AJMP E24X
E22: MOV R3,CIT2
; eroarea la CIT2,1 cu semn la
AdrSemn
MOV R2,CIT1
; nr. de convertit la R3,R2
MOV R7,#1FH
; in R7,6 8000=1F.40
MOV R6,#40H
ACALL DIF ; er-8000
JC E28S
MOV A,AdrSemn
; er>80.00%
CJNE A,#0FFH,E29S
MOV CIT2S,#0E0H
; er>-80.00%
MOV CIT1S,#04H
MOV AFIS2,#7BH
MOV AFIS1,#0FFH
AJMP E24X
E29S: MOV CIT2S,#60H
; er>+80.00%
MOV CIT1S,#03H
MOV AFIS2,#0BH
MOV AFIS1,#0FFH
AJMP E24X
E28S: ACALL SUMA
; refac er in R3,2
MOV R7,#07H
; în R7,6=2000 (07.D0)
MOV R6,#0D0H
ACALL DIF
; efectuez eroare-2000
JC E28
MOV A,AdrSemn
; C=0, er>=2000
CJNE A,#0FFH,E29
MOV A,CIT2
; er>-19,99
ORL A,#0A0H
MOV CIT2S,A
; info. pt. achiziție
MOV A,CIT1
MOV CIT1S,A
MOV AFIS2,#7BH
; info. pt. afișaj
MOV AFIS1,#0FFH
AJMP E24X

```

Program post calcul și afișarea erorii ERRORREX M

E29:	MOV A,CIT2 ; er>19,99 ORL A,#20H MOV CIT2S,A ; info. pt. achiziție MOV A,CIT1 MOV CIT1S,A MOV AFIS2,#0BH ; info. pt. afisaj MOV AFIS1,#0FFH AJMP E24X	MOV A,CIT1 CLR C RLC A RLC A RLC A ORL A,R2 MOV CIT1,A ; rezultat conv. la CIT2,CIT1 MOV A,CIT2 ANL A,#0F0H JZ ETI MOV A,AdrSemn JZ ETI3 ; eroare +_x.xx ORL CIT2,#70H ; eroare -_x.xx ORL CIT2S,#80H AJMP ETI3
E28:	MOV CIT2,#00 MOV CIT1,#00 ACALL SUMA ; C=1, er<2000, reface er. în R3.2 MOV CIT2S,R3 ; er. se transm în binar MOV CIT1S,R2 MOV R7,#03H ; 1000 (03.E8) în R7,6 MOV R6,#0E8H	ETI: MOV A,AdrSemn JZ ETI2 ORL CIT2,#30H ; eroare -1x.xx ORL CIT2S,#80H AJMP ETI3
E31:	INC CIT2 ACALL DIF ; execută er.-1000 JC E30 AJMP E31	ETI2: ANL CIT2,#0FH ORL CIT2,#80H ; eroare +1x.xx
E30:	DEC CIT2 ; er<1000 (C=1) ACALL SUMA MOV R7,#00H ; 100 (00.64) în R7,6 MOV R6,#64H MOV A,CIT2 CLR C RLC A RLC A RLC A RLC A MOV CIT2,A ; la CIT2,1 avem X0.00	ETI3: MOV AFIS1,CIT1 MOV AFIS2,CIT2 E24X: MOV TAB2,AFIS2 MOV TAB4,AFIS2 MOV TAB6,AFIS2 MOV TAB1,AFIS1 MOV TAB3,AFIS1 MOV TAB5,AFIS1 E24: RET
E33:	INC CIT2 ACALL DIF ; efectuez er-100 JC E32 AJMP E33	; Subrutina recepționează caracter serial RECEPT: PUSH 0E0H ; salvez conținut A PUSH 0D0H MOV CUV,SBUF ; citesc cuvânt recepționat JB FAN.1,E36 JB FAN.2,E37 ; primul cuvânt ? MOV A,AdrNP ; cuvânt I = NP ? CJNE A,CUV,E38 MOV CUV1,#00
E32:	DEC CIT2 ACALL SUMA MOV R6,#0AH ; 10 (00.0A) în R7,6 MOV R7,#00H	E41: SETB FAN.2 ; da, cuvânt = NP
E35:	INC CIT1 ACALL DIF ; efectuez er-10 JC E34 AJMP E35	E44: INC NRC
E34:	DEC CIT1 ACALL SUMA	E40: MOV CONT5,#00 ; inițializare buclă de timp



E51:	MOV CONT6,#00	AJMP E48
	POP OD0H	E40V: MOV CUV2,#12H
	POP OE0H	AJMP E48
	RET	E42: CJNE A,#02,E49
E38:	MOV A, CUV	MOV A, CUV2
	; nu, cuvântul nu e NP	; cuvânt III
	INC A	CJNE A,#INVAT,E50X
	JZ E39	MOV AdrNP, CUV
	; cuvânt = FF ?	; cuvânt III este NP
	SETB FAN.1	AJMP E48
	; nu, cuvânt nu e FF, se transmite	E50X: CJNE A,#TURE,E50S
E36:	MOV CONT10,#00	; cuvânt III este nr. TURE
	; anulez contor de 2ms	MOV AdrTU,#01H
	AJMP E51	; CEEE
E39:	MOV A, CONT8	MOV AdrTUI, CUV
	; posturi legate în paralel ?	AJMP E48
	CJNE A,#00,E39S	E50S: CJNE A,#FILTRU,E50
	SETB P2.2	; cuvânt III este nr. pt. FILTRU
	MOV CONT8,#01	MOV AdrFIL, CUV
	MOV CONT9,#00	AJMP E48
	; init. bucla de 2s	E50: MOV AK1, CUV
	AJMP E51	; cuvânt III este K1
E39S:	MOV CUV1,#0FFH	AJMP E44
	; da, cuvânt = FF	E49: MOV AK2, CUV
	AJMP E41	; al IV-lea cuvânt este K2
E37:	MOV A, NRC	AJMP E48
	; al II-lea cuvânt ?	; continuare program principal
	CJNE A,#01,E42	E52: JB FAN.3,E64
	MOV CUV2, CUV	AJMP E53
	; cuvânt II	E64: MOV A, CUV2
	MOV A, CUV	CJNE A,#INVAT,X6
	CJNE A,#INVAT,E43	CLR IE.0
	AJMP E44	CLR IE.1
	; dacă cod INVAT	MOV CUV6,#00
E43:	CJNE A,#ACHIZ,E45	; cd. INVATARE POZITIE
E48:	MOV FAN,#08	MOV A, AdrNP
	; dacă cod ACHIZ	; pregătește transmisia la PC: NP
	MOV NRC,#00	COD
	AJMP E51	INC A
E45:	CJNE A,#CONST,E46	; și la postul următor
	AJMP E44	CJNE A,#0FFH,E64S
	; dacă cod CONST	MOV A,#01
E46:	CJNE A,#AFIS,E47	E64S: MOV CUV4,A
	MOV NUM,#00H	; FF COD NP+1
	AJMP E48	MOV CIT2,#00
	; dacă cod AFIS	; pregătește NP pt. afișare
E47:	CJNE A,#MONIT,E40X	MOV CIT1, AdrNP
	AJMP E44	MOV AdrSemn,#00
	; dacă cod MONIT	ACALL E22
E40X:	CJNE A,#TURE,E40S	; nr. în CIT2, CIT1
	; dacă cod TURE	MOV A, AFIS2
	AJMP E44	ORL A,#0F0H
E40S:	CJNE A,#FILTRU,E40W	MOV AFIS2,A
	; dacă cod FILTRU	ANL A,#0FH
	AJMP E44	JZ E55
E40W:	CJNE A,#NRMARCI,E40V	ORL AFIS2,#0EH
	; dacă cod NRMARCI	MOV A, AFIS1

	ANL	A,#0F0H		CLR	FAN.1
	JZ	E56	E200:	MOV	A,CONT6
	ORL	AFIS1,#0F0H		CJNE	A,#1CH,E61
E56:	MOV	TAB6,AFIS2		MOV	CONT5,#00
	MOV	TAB5,AFIS1		MOV	CONT6,#00
E56X:	MOV	TAB1,AFIS1		CLR	FAN.3
	MOV	TAB3,AFIS1		CLR	FAN.1
	MOV	TAB2,AFIS2		MOV	NRC,#00
	MOV	TAB4,AFIS2		CLR	FAN.2
	MOV	CIT1S,AdrNP		INC	CONSEC
	MOV	CIT2S,CUV2		MOV	A,CONSEC
	MOV	CONTPC,#01H		CJNE	A,#03,X7
	MOV	CONTSR,#04H		INC	CONAFI
E66:	SETB	P2.1		MOV	CONSEC,#00
E58:	MOV	A,CONTPC		MOV	A,CONAFI
	JZ	E57		CJNE	A,#01,X11
	ACALL	TRANSE		MOV	AFIS2,TAB6
	DEC	CONTPC		MOV	AFIS1,TAB5
	MOV	CIT1S,CUV4		AJMP	E63
	MOV	CIT2S,CUV5	E201:	MOV	A,CONT9
	AJMP	E58		CJNE	A,#02,E200
E55:	MOV	TAB6,#0FEH		MOV	CONT9,#00
	MOV	TAB5,#0FFH		CLR	P2.2
	AJMP	E56X		MOV	CONT8,#00
X6:	AJMP	E54		AJMP	E200
E57:	MOV	A,CUV1	X11:	CJNE	A,#02,E62
	INC	A		MOV	AFIS2,TAB2
	JZ	E59		MOV	AFIS1,TAB1
	MOV	CIT1S,CUV1	E63:	ACALL	AFISAJ
	MOV	CIT2S,CUV2	X7:	AJMP	E9
	CLR	P2.1	E62:	CJNE	A,#03,E63
E60:	MOV	A,CONTSR		MOV	AFIS2,TAB4
	; prog. frecv. transm. post (600)			MOV	AFIS1,TAB3
	JZ	E59		MOV	CONAFI,#00
	ACALL	TRANSE		INC	CONT9
	DEC	CONTSR		AJMP	E63
	; prog. frecv. transm. PC (2400)		E61:	INC	CONT10
	MOV	CIT1S,CUV4		; contoriz. bucla de 2ms	
	MOV	CIT2S,CUV5		MOV	A,CONT5
	AJMP	E60		ADD	A,#01
E59:	CLR	FAN.3		MOV	CONT5,A
	CLR	P2.2		MOV	A,CONT6
	; legatura pt. posturi în paralel			ADDC	A,#00
	MOV	CONT8,#00		MOV	CONT6,A
	MOV	CONT5,#00		JB	FAN.3,X9
	MOV	CONT6,#00		AJMP	E9
	ACALL	AFISAJ	X9:	AJMP	E64
E53:	MOV	A,CONT8	E54:	CJNE	A,#ACHIZ,E65
	; dacă posturi în paralel (CONT8=0)			MOV	CUV4,CIT1S
	CJNE	A,#00,E201		; cd. ACHIZITIE	
	MOV	A,CONT10		MOV	CUV5,CIT2S
	; au trecut 2ms?			; pregătește transm. la PC	
	CJNE	A,#50H,E200		MOV	CIT1S,AdrNP
	; nu			; NP COD E1 E2	
	MOV	CONT10,#00		MOV	CIT2S,CUV2
	; da, anulez CONT8 și FAN1			; și la postul următor: FF(NP) COD	
				MOV	CONTPC,#02H

<p>E65: MOV CONTSR,#01H          AJMP E66          CJNE A,#CONST,E67          MOV CUV6,#00          MOV CONT2,#04H          MOV AFIS2,#0FAH          ; cd. CONST          MOV AFIS1,#0AAH          ; pregătește transm. la PC          MOV TAB2,#0FAH          ; NP(FF) COD K1 K2          MOV TAB1,#0AAH          ; și la postul următor: FF COD K1</p> <p>K2          MOV TAB4,#0FAH          MOV TAB3,#0AAH          MOV TAB6,#0FAH          MOV TAB5,#0AAH          SETB IE.0          ; activează întreruperi generate          SETB IE.1          ; de MARCA și TIMER0 (de la Fe)</p> <p>E69: MOV CIT1S,AdrNP          MOV CIT2S,CUV2          MOV CUV4,AK1          MOV CUV5,AK2          MOV CONTPC,#02          MOV CONTSR,#02          AJMP E66</p> <p>E67: CJNE A,#AFIS,E68          ; cd. AUTOTEST AFISAJ          CLR IE.0          CLR IE.1          MOV CUV6,#44H          MOV CIT1S,CUV1          ; preg. transm. la post următor</p> <p>FF(NP) COD          MOV CIT2S,CUV2          MOV CONTSR,#01          AJMP E57</p> <p>E68: CJNE A,#MONIT,E68X          CLR IE.0          CLR IE.1          MOV CUV6,#00          MOV AFIS1,AK1          ; cd. MONITOR          MOV AFIS2,AK2          MOV AdrSemn,#00          MOV TAB6,AFIS2          MOV TAB5,AFIS1          MOV TAB2,AFIS2          MOV TAB1,AFIS1          MOV TAB4,AFIS2          MOV TAB3,AFIS1          AJMP E69</p> <p>E68X: CJNE A,#TURE,E70</p>	<p>MOV CUV4,AdrTUI          ; cd. TURE          MOV CUV5,#00          ; pregătește transm. la PC          MOV CIT1S,AdrNP          ; NP COD TURE          MOV CIT2S,CUV2          ; și la postul următor FF(NP) COD          MOV CONTPC,#02H          MOV CONTSR,#04H          AJMP E66</p> <p>E70: CJNE A,#FILTRU,E71          MOV CUV4,AdrFIL          ; cd. FILTRU          MOV CUV5,#00          ; pregătește transm. la PC          MOV CIT1S,AdrNP          ; NP COD FILTRU          MOV CIT2S,CUV2          ; și la postul următor FF(NP) COD          MOV CONTPC,#02H          MOV CONTSR,#04H          AJMP E66</p> <p>E71: CJNE A,#NRMARCI,E71S          ; cd. Nr.MARCI          MOV CUV6,#00          MOV CONT2,#04H          MOV A,AdrMARC          JZ E100          MOV BAK,A</p> <p>E100: MOV CUV4,AdrMARC          ; pregătește transm. la PC          MOV AdrMARC,#00          MOV CUV5,BAK          ; NP COD NR.MARCI BAK          MOV CIT1S,AdrNP          ; și la postul următor FF(NP) COD          MOV CIT2S,CUV2          MOV CONTPC,#02H          MOV CONTSR,#02H          AJMP E66</p> <p>E71S: MOV CIT2S,CUV2          ; pt. cod eronat transm. la PC NP</p> <p>COD          MOV CIT1S,AdrNP          MOV CONTPC,#01H          MOV CONTSR,#00          AJMP E6</p>
--	---

```

; Program intrefață achiziție
ERRORREX M (1/3)
; procesor M1 (comunicare serială)

; Tabela cu adrese utilizate
STATIE      EQU 02EH
FAN         EQU 02FH
NRC         EQU 030H
CUV         EQU 031H
COD         EQU 032H
CONT5       EQU 033H
CONT6       EQU 034H
CONT10      EQU 035H
FACTOR      EQU 036H
S1          EQU 037H
S2          EQU 038H
S3          EQU 039H
S4          EQU 03AH
S5          EQU 03BH
S6          EQU 03CH
S7          EQU 03DH
S8          EQU 03EH
S9          EQU 03FH
S10         EQU 040H
S11         EQU 041H
CUCU        EQU 042H

FACT        EQU 01AH
RELEU       EQU 01BH
ENERGIE     EQU 01CH

ORG 0
LJMP E150
ORG 23H
LJMP INSER

; program principal
E150: MOV SP,#20H
      ; programare stivă
      MOV IE,#90H
      ; prog.întreruperi
      MOV IP,#10H
      ; prog. priorități întreruperi
      MOV TCON,#40H
      MOV TMOD,#20H
      ; programare timer
      MOV SCON,#50H
      MOV TL1,#0FDH
      ; TIMER1-tact serial (9600 baud)
      MOV TH1,#0FDH
      MOV NRC,#00
      MOV FAN,#00
      ; locație adresabilă pe bit
      MOV P1,#00
      MOV CONT10,#00

E22:  MOV CONT5,#00
      MOV CONT6,#00
E11:  JB FAN.7,E12
      INC CONT10
      MOV A,CONT10
      CJNE A,#0FAH,E11
      CLR FAN.1
      MOV CONT10,#00
      MOV A,CONT5
      ADD A,#01
      MOV CONT5,A
      MOV A,CONT6
      ADDC A,#00
      MOV CONT6,A
      CJNE A,#04,E11
      CLR FAN.1
      CLR FAN.2
      CLR FAN.7
      MOV NRC,#00
      MOV CONT5,#00
      MOV CONT6,#00
      AJMP E11
E12:  MOV A,COD
      MOV S1,#0FEH
      CJNE A,#FACT,E13
      MOV S2,#1AH
      MOV S3,FACTOR
E24:  MOV SBUF,S1
E15:  JBC SCON.1,E14
      AJMP E15
E14:  MOV SBUF,S2
E17:  JBC SCON.1,E16
      AJMP E17
E16:  MOV SBUF,S3
E19:  JBC SCON.1,E20
      AJMP E19
E20:  CLR FAN.7
      CLR FAN.1
      CLR FAN.2
      AJMP E22
E13:  CJNE A,#RELEU,E23
      MOV A,STATIE
      ANL A,#07H
      MOV CUCU,A
      MOV A,P2
      ANL A,#0F8H
      ORL A,CUCU
      MOV P2,A
      MOV S2,#1BH
      MOV S3,STATIE
      AJMP E24
E23:  CJNE A,#ENERGIE,E20
      CLR P2.3
      MOV S2,#1CH
      CLR P2.7
E102: JB P2.6,E102
      SETB P2.3

```

	MOV	S3,P0		AJMP	E129
	SETB	P2.7	E128:	MOV	SBUF,S6
E103:	JNB	P2.6,E103	E131:	JBC	SCON.1,E130
	CLR	P2.7		AJMP	E131
E104:	JB	P2.6,E104	E130:	MOV	SBUF,S7
	MOV	S4,P0	E133:	JBC	SCON.1,E132
	SETB	P2.7		AJMP	E133
E105:	JNB	P2.6,E105	E132:	MOV	SBUF,S8
	CLR	P2.7	E135:	JBC	SCON.1,E134
E106:	JB	P2.6,E106		AJMP	E135
	MOV	S5,P0	E134:	MOV	SBUF,S9
	SETB	P2.7	E137:	JBC	SCON.1,E136
E107:	JNB	P2.6,E107		AJMP	E137
	CLR	P2.7	E136:	MOV	SBUF,S10
E108:	JB	P2.6,E108	E139:	JBC	SCON.1,E138
	MOV	S6,P0		AJMP	E139
	SETB	P2.7	E138:	MOV	SBUF,S11
E109:	JNB	P2.6,E109	E141:	JBC	SCON.1,E140
	CLR	P2.7		AJMP	E141
E110:	JB	P2.6,E110	E140:	CLR	FAN.7
	MOV	S7,P0		CLR	FAN.1
	SETB	P2.7		CLR	FAN.2
E111:	JNB	P2.6,E111		AJMP	E22
	CLR	P2.7			
E112:	JB	P2.6,E112			
	MOV	S8,P0			
	SETB	P2.7			
E113:	JNB	P2.6,E113			
	CLR	P2.7			
E114:	JB	P2.6,E114			
	MOV	S9,P0			
	SETB	P2.7			
E115:	JNB	P2.6,E115			
	CLR	P2.7			
E116:	JB	P2.6,E116			
	MOV	S10,P0			
	SETB	P2.7			
E117:	JNB	P2.6,E117			
	CLR	P2.7			
E118:	JB	P2.6,E118			
	MOV	S11,P0			
	SETB	P2.7			
E119:	JNB	P2.6,E119			
	MOV	SBUF,S1			
E121:	JBC	SCON.1,E120			
	AJMP	E121			
E120:	MOV	SBUF,S2	E8:	INC	NRC
E123:	JBC	SCON.1,E122		MOV	CONT5,#00
	AJMP	E123			
E122:	MOV	SBUF,S3			
E125:	JBC	SCON.1,E124	E5:	POP	ACC
	AJMP	E125		RET	
E124:	MOV	SBUF,S4	E4:	SETB	FAN.1
E127:	JBC	SCON.1,E126			
	AJMP	E127	E2:	MOV	CONT10,#00
E126:	MOV	SBUF,S5			
E129:	JBC	SCON.1,E128			

; Subrutina tratează întreruperea serială

INSER: JBC SCON.0,E1

RETI

E1: ACALL RECEPTIE

; recepționează caracter serial de

forma FE.11.nn

RETI

; Subrutina recepționează caracter serial

RECEPTIE:

PUSH ACC

; salvez conținut A

MOV CUV,SBUF

; citesc cuvânt recepționat

JB FAN.1,E2

JB FAN.2,E3

; primul cuvânt ?

MOV A,#0FEH

; este cuvântul pentru interfață ?

CJNE A,CUV,E4

SETB FAN.2

; da, cuvânt = NP

E8: INC NRC

MOV CONT5,#00

; inițializare bucla de timp

MOV CONT6,#00

E5: POP ACC

RET

E4: SETB FAN.1

; nu e cuvânt pt. interfață

E2: MOV CONT10,#00

; anulez contor de 2ms

AJMP E5

E3: MOV A,NRC  
CJNE A,#01,E9  
MOV A,CUV  
; al II-lea cuvânt  
MOV COD,A  
CJNE A,#ENERGIE,E6  
E7: SETB FAN.7  
; dacă cod ENERGIE  
MOV NRC,#00  
AJMP E5  
E6: CJNE A,#RELEU,E10  
AJMP E8  
; dacă cod MONO/TRI  
E10: CJNE A,#FACT,E7  
AJMP E8  
; dacă cod MONIT  
E9: MOV A,COD  
CJNE A,#FACT,E30  
MOV FACTOR,CUV  
; al III-lea cuvânt  
JB P2.4,EX  
JB P2.5,EX  
EY: MOV P1,FACTOR  
AJMP E7  
EX: ORL FACTOR,#80H  
AJMP EY  
E30: CJNE A,#RELEU,E7  
MOV STATIE,CUV  
AJMP E7

; Program interfață achiziție  
**ERRORREX M (2/3)**  
 ; procesor M2 (însușire frecvență etalon)

; Tabela cu adrese utilizate

FAN	EQU	02FH
S1	EQU	030H
S2	EQU	031H
S3	EQU	032H
S4	EQU	033H
S5	EQU	034H
S6	EQU	035H
Adr1	EQU	036H
S10	EQU	037H
S20	EQU	038H
S30	EQU	039H
S40	EQU	03AH
S50	EQU	03BH
S60	EQU	03CH
S100	EQU	03DH
S200	EQU	03EH
S300	EQU	03FH
S400	EQU	040H
S500	EQU	041H
S600	EQU	042H

ORG 0  
 LJMP E1  
 ORG 03H  
 LJMP E9  
 ORG 0BH  
 LJMP E90  
 ORG 013H  
 LJMP E900  
 ORG 01BH  
 LJMP E11

; program principal

E1: MOV SP,#20H  
 ; programare stiva  
 MOV IE,#8FH  
 ; prog.întreruperi  
 MOV IP,#07H  
 ; prog. priorități întreruperi  
 MOV TCON,#55H  
 MOV TMOD,#66H  
 ; programare timer  
 MOV S1,#00  
 MOV S2,#00  
 MOV S3,#00  
 MOV S10,#00  
 MOV S20,#00  
 MOV S30,#00  
 MOV S100,#00

MOV S200,#00  
 MOV S300,#00  
 MOV TL0,#0FFH  
 MOV TH0,#0FFH  
 MOV TL1,#0FFH  
 MOV TH1,#0FFH  
 E2: JNB FAN.7,E2  
 CLR FAN.7  
 E3: JB P2.7,E3  
 MOV P1,S4  
 CLR P2.6  
 E4: JNB P2.7,E4  
 SETB P2.6  
 E5: JB P2.7,E5  
 MOV P1,S5  
 CLR P2.6  
 E6: JNB P2.7,E6  
 SETB P2.6  
 E7: JB P2.7,E7  
 MOV P1,S6  
 CLR P2.6  
 E8: JNB P2.7,E8  
 SETB P2.6  
 E13: JB P2.7,E13  
 MOV P1,S40  
 CLR P2.6  
 E14: JNB P2.7,E14  
 SETB P2.6  
 E15: JB P2.7,E15  
 MOV P1,S50  
 CLR P2.6  
 E16: JNB P2.7,E16  
 SETB P2.6  
 E17: JB P2.7,E17  
 MOV P1,S60  
 CLR P2.6  
 E18: JNB P2.7,E18  
 SETB P2.6  
 E19: JB P2.7,E19  
 MOV P1,S400  
 CLR P2.6  
 E20: JNB P2.7,E20  
 SETB P2.6  
 E21: JB P2.7,E21  
 MOV P1,S500  
 CLR P2.6  
 E22: JNB P2.7,E22  
 SETB P2.6  
 E23: JB P2.7,E23  
 MOV P1,S600  
 CLR P2.6  
 E24: JNB P2.7,E24  
 SETB P2.6  
 AJMP E2

```
; Subrutina tratează întreruperea externă
INT0
; generează impuls la P2.3 cu durata de 11
microcilcii
E9:   SETB  P2.3
      XCH  A,Adr1
      MOV  A,S1
      ADD  A,#01
      MOV  S1,A
      JNC  E10
      MOV  A,S2
      ADDC A,#00
      MOV  S2,A
      MOV  A,S3
      ADDC A,#00
      MOV  S3,A
E10:  CLR  P2.3
      XCH  A,Adr1
      RETI

; Subrutina tratează întreruperea externă
INT1
; generează impuls la P2.3 cu durata de 11
microcilcii
E90:  SETB  P2.3
      XCH  A,Adr1
      MOV  A,S10
      ADD  A,#01
      MOV  S10,A
      JNC  E100
      MOV  A,S20
      ADDC A,#00
      MOV  S20,A
      MOV  A,S30
      ADDC A,#00
      MOV  S30,A
E100: CLR  P2.3
      XCH  A,Adr1
      RETI

; Subrutina tratează întreruperea de overflow
data de TIMER0
; generează impuls la P2.3 cu durata de 11
microcilcii
E900: SETB  P2.3
      XCH  A,Adr1
      MOV  A,S100
      ADD  A,#01
      MOV  S100,A
      JNC  E1000
      MOV  A,S200
      ADDC A,#00
      MOV  S200,A
      MOV  A,S300
      ADDC A,#00
      MOV  S300,A
E1000: CLR  P2.3

; Subrutina tratează întreruperea de overflow
data de TIMER1 când se transmite comanda
de măsurare energie
E11:  CLR  IE.7
      MOV  S4,S1
      MOV  S5,S2
      MOV  S6,S3
      MOV  S1,#00
      MOV  S2,#00
      MOV  S3,#00
      MOV  S40,S10
      MOV  S50,S20
      MOV  S60,S30
      MOV  S10,#00
      MOV  S20,#00
      MOV  S30,#00
      MOV  S400,S100
      MOV  S500,S200
      MOV  S600,S300
      MOV  S100,#00
      MOV  S200,#00
      MOV  S300,#00
      SETB IE.7
      SETB FAN.7
      RETI
```



```

; Program interfață achiziție
ERRORREX M (3/3)
; procesor M3, M4, M5 (măsurare și
generare)

Adr1      EQU    030H
Adr2      EQU    031H
Adr3      EQU    032H
Adr4      EQU    033H
Nrimp     EQU    037H

ORG 0
LJMP M1
ORG 3
LJMP M3
ORG 01BH
LJMP M7

; program principal

M1:  MOV  SP,#20H
      ; programare stiva
      MOV  IE,#89H
      ; prog.întreruperi
      MOV  IP,#09H
      ; prog. priorități întreruperi (08)
      MOV  TCON,#11H
      MOV  TMOD,#11H
      ; programare timer
      CLR  P2.1
      ; există microcontroler
M2:  NOP
      AJMP M2

; Subrutina tratează întreruperea externă IE0
M3:  SETB P2.0
      CLR  TCON.4
      ; citește TIMER0
      MOV  Adr1,TL0
      MOV  Adr2,TH0
      MOV  TL0,#00
      MOV  TH0,#00
      SETB TCON.4
      MOV  A,P1
      CLR  P2.0
      JZ   M4
      JB   TCON.5,E1
      CJNE A,#01,M5
      MOV  Nrimp,#01H
      ; se multiplică cu 2
      CLR  C
      MOV  A,Adr2
      RRC  A
      CPL  A
      MOV  Adr4,A

MOV  A,Adr1
RRC  A
CPL  A
ADD  A,#24H
MOV  TL1,A
MOV  A,Adr4
ADDC A,#00
JC   M15
MOV  TH1,A
SETB TCON.6
M4:  RETI
M15: MOV  TL1,#0FFH
      MOV  TH1,#0FFH
      SETB TCON.6
      RETI
M5:  CJNE A,#02,M6
      MOV  Nrimp,#03H
      CLR  C
      ; multipl. cu 4
      MOV  A,Adr2
      RRC  A
      MOV  Adr2,A
      MOV  A,Adr1
      RRC  A
      MOV  Adr1,A
      CLR  C
      MOV  A,Adr2
      RRC  A
      CPL  A
      MOV  Adr4,A
      MOV  A,Adr1
      RRC  A
      CPL  A
      MOV  Adr3,A
      MOV  A,Adr4
      CJNE A,#0FFH,E110
      MOV  A,Adr3
      RR   A
      RR   A
      RR   A
      RR   A
      ANL  A,#0FH
      ADD  A,Adr3
      ADD  A,#0BH
      MOV  Adr3,A
      ADD  A,#0FH
      MOV  TL1,A
      MOV  TH1,#0FFH
      JC   E180
      SETB TCON.6
      RETI
E180: MOV  TL1,#0FFH
      SETB TCON.6
      RETI
E110: MOV  A,Adr3
      ADD  A,#11H
      MOV  Adr3,A

```

Program interfață achiziție (M3, M4, M5) ERRORREX M

	MOV	A,Adr4		ADDC	A,#00
	ADDC	A,#00		MOV	Adr4,A
	MOV	Adr4,A		SETB	TCON.6
	SETB	TCON.6		RETI	
	RETI		M9:	CJNE	A,#04,M10
E1:	AJMP	E11			; se înmulțește cu 16
M6:	CJNE	A,#03,M9		MOV	Nrimp,#0FH
	MOV	Nrimp,#07H		CLR	C
		; se multiplică cu 8		MOV	A,Adr2
	CLR	C		RRC	A
	MOV	A,Adr2		MOV	Adr2,A
	RRC	A		MOV	A,Adr1
	MOV	Adr2,A		RRC	A
	MOV	A,Adr1		MOV	Adr1,A
	RRC	A		CLR	C
	MOV	Adr1,A		MOV	A,Adr2
	CLR	C		RRC	A
	MOV	A,Adr2		MOV	Adr2,A
	RRC	A		MOV	A,Adr1
	MOV	Adr2,A		RRC	A
	MOV	A,Adr1		MOV	Adr1,A
	RRC	A		CLR	C
	MOV	Adr1,A		MOV	A,Adr2
	CLR	C		RRC	A
	MOV	A,Adr2		MOV	Adr2,A
	RRC	A		MOV	A,Adr1
	MOV	Adr2,A		RRC	A
	MOV	A,Adr1		MOV	Adr1,A
	RRC	A		CLR	C
	MOV	Adr1,A		MOV	A,Adr2
	CLR	C		RRC	A
	MOV	A,Adr2		MOV	Adr2,A
	RRC	A		MOV	A,Adr1
	CPL	A		RRC	A
	MOV	Adr4,A		MOV	Adr1,A
		; la Adr4,3 durata dintre impulsuri		CLR	C
	MOV	A,Adr1		MOV	A,Adr2
	RRC	A		RRC	A
	CPL	A		CPL	A
	MOV	Adr3,A		MOV	Adr4,A
	MOV	A,Adr4		MOV	A,Adr1
	CJNE	A,#0FFH,M110		RRC	A
	MOV	A,Adr3		CPL	A
	RR	A		MOV	Adr3,A
	RR	A		MOV	A,Adr4
	RR	A		CJNE	A,#0FFH,M210
	RR	A		MOV	A,Adr3
	ANL	A,#0FH		RR	A
	ADD	A,Adr3		RR	A
	ADD	A,#0BH		RR	A
	MOV	Adr3,A		RR	A
	ADD	A,#0FH		ANL	A,#0FH
	MOV	TL1,A		ADD	A,Adr3
	MOV	TH1,#0FFH		ADD	A,#0AH
	JC	M180		MOV	Adr3,A
	SETB	TCON.6		ADD	A,#0CH
	RETI			MOV	TL1,A
M180:	MOV	TL1,#0FFH		MOV	TH1,#0FFH
	SETB	TCON.6		JC	M280
	RETI			SETB	TCON.6
M110:	MOV	A,Adr3		RETI	
	ADD	A,#11H			
	MOV	Adr3,A	M280:	MOV	TL1,#0FFH
	MOV	A,Adr4		SETB	TCON.6

	RETI		RR	A	
M210:	MOV	A,Adr3	ANL	A,#03H	
	ADD	A,#0EH	ADD	A,Adr3	
	MOV	Adr3,A	ADD	A,#0DH	
	MOV	A,Adr4	MOV	Adr3,A	
	ADDC	A,#00	ADD	A,#15H	
	MOV	Adr4,A	MOV	TL1,A	
	SETB	TCON.6	MOV	TH1,#0FFH	
	RETI		JC	M171	
M10:	CJNE	A,#05,M11	SETB	TCON.6	
	; se înmulțește cu 32		RETI		
	MOV	Nrimp,#1FH	M171:	MOV	TL1,#0FFH
	CLR	C		SETB	TCON.6
	MOV	A,Adr2		RETI	
	RRC	A	M101:	MOV	A,Adr3
	MOV	Adr2,A		ADD	A,#0EH
	MOV	A,Adr1		MOV	Adr3,A
	RRC	A		MOV	A,Adr4
	MOV	Adr1,A		ADDC	A,#00
	CLR	C		MOV	Adr4,A
	MOV	A,Adr2		SETB	TCON.6
	RRC	A	M41:	RETI	
	MOV	Adr2,A	M11:	CJNE	A,#06,M41
	MOV	A,Adr1		MOV	Nrimp,#3FH
	RRC	A		; se înmulțește cu 64	
	MOV	Adr1,A		CLR	C
	CLR	C		MOV	A,Adr2
	MOV	A,Adr2		RRC	A
	RRC	A		MOV	Adr2,A
	MOV	Adr2,A		MOV	A,Adr1
	MOV	A,Adr1		RRC	A
	RRC	A		MOV	Adr1,A
	MOV	Adr1,A		CLR	C
	CLR	C		MOV	A,Adr2
	MOV	A,Adr2		RRC	A
	RRC	A		MOV	Adr2,A
	CPL	A		MOV	A,Adr1
	MOV	Adr4,A		RRC	A
	MOV	A,Adr1		MOV	Adr1,A
	RRC	A		CLR	C
	CPL	A		MOV	A,Adr2
	MOV	Adr3,A		RRC	A
	MOV	A,Adr4		MOV	Adr2,A
	CJNE	A,#0FFH,M101		MOV	A,Adr1
	MOV	A,Adr3		RRC	A
	RR	A		MOV	Adr1,A
	RR	A		CLR	C
	RR	A		MOV	A,Adr2
	RR	A		RRC	A
	RR	A		MOV	Adr2,A

	MOV	A,Adr1		SETB	TCON.6
	RRC	A		RETI	
	MOV	Adr1,A	E2:	CJNE	A,#02,E3
	CLR	C		MOV	Nrimp,#03H
	MOV	A,Adr2		SETB	C
	RRC	A			; multipl. cu 4
	CPL	A		MOV	A,Adr2
	MOV	Adr4,A		RRC	A
	MOV	A,Adr1		MOV	Adr2,A
	RRC	A		MOV	A,Adr1
	CPL	A		RRC	A
	MOV	Adr3,A		MOV	Adr1,A
	MOV	A,Adr4		CLR	C
	CJNE	A,#0FFH,M102		MOV	A,Adr2
	MOV	A,Adr3		RRC	A
	RR	A		CPL	A
	RR	A		MOV	Adr4,A
	RR	A		MOV	A,Adr1
	RR	A		RRC	A
	RR	A		CPL	A
	RR	A		MOV	Adr3,A
	ANL	A,#03H		SETB	TCON.6
	ADD	A,Adr3		RETI	
	ADD	A,#0EH	E3:	CJNE	A,#03,E4
	MOV	Adr3,A		MOV	Nrimp,#07H
	ADD	A,#14H			; se multiplică cu 8
	MOV	TL1,A		SETB	C
	MOV	TH1,#0FFH		MOV	A,Adr2
	JC	M172		RRC	A
	SETB	TCON.6		MOV	Adr2,A
	RETI			MOV	A,Adr1
M172:	MOV	TL1,#0FFH		RRC	A
	SETB	TCON.6		MOV	Adr1,A
	RETI			CLR	C
M102:	MOV	A,Adr3		MOV	A,Adr2
	ADD	A,#0EH		RRC	A
	MOV	Adr3,A		MOV	Adr2,A
	MOV	A,Adr4		MOV	A,Adr1
	ADDC	A,#00		RRC	A
	MOV	Adr4,A		MOV	Adr1,A
	SETB	TCON.6		CLR	C
	RETI			MOV	A,Adr2
				RRC	A
				CPL	A
				MOV	Adr4,A
					; la Adr4,3 durata dintre impulsuri
				MOV	A,Adr1
				RRC	A
				CPL	A
				MOV	Adr3,A
				SETB	TCON.6
				RETI	
			E4:	CJNE	A,#04,E5
					; se înmulțește cu 16
				MOV	Nrimp,#0FH
				SETB	C
				MOV	A,Adr2

; se tratează Fe mică

E11: CLR TCON.5  
 CJNE A,#01,E2  
 MOV Nrimp,#01H  
 ; se multiplică cu 2  
 SETB C  
 MOV A,Adr2  
 RRC A  
 CPL A  
 MOV Adr4,A  
 MOV A,Adr1  
 RRC A  
 CPL A  
 MOV Adr3,A

Program interfață achiziție (M3, M4, M5) ERRORREX M

RRC	A		MOV	Adr2,A
MOV	Adr2,A		MOV	A,Adr1
MOV	A,Adr1		RRC	A
RRC	A		MOV	Adr1,A
MOV	Adr1,A		CLR	C
CLR	C		MOV	A,Adr2
MOV	A,Adr2		RRC	A
RRC	A		CPL	A
MOV	Adr2,A		MOV	Adr4,A
MOV	A,Adr1		MOV	A,Adr1
RRC	A		RRC	A
MOV	Adr1,A		CPL	A
CLR	C		ADD	A,#0EH
MOV	A,Adr2		MOV	Adr3,A
RRC	A		MOV	A,Adr4
MOV	Adr2,A		ADDC	A,#00
MOV	A,Adr1		MOV	Adr4,A
RRC	A		SETB	TCON.6
MOV	Adr1,A	E7:	RETI	
CLR	C	E6:	CJNE	A,#06,E7
MOV	A,Adr2		MOV	Nrimp,#3FH
RRC	A			; se înmulțește cu 64
CPL	A		SETB	C
MOV	Adr4,A		MOV	A,Adr2
MOV	A,Adr1		RRC	A
RRC	A		MOV	Adr2,A
CPL	A		MOV	A,Adr1
MOV	Adr3,A		RRC	A
SETB	TCON.6		MOV	Adr1,A
RETI			CLR	C
E5:	CJNE	A,#05,E6	MOV	A,Adr2
		; se înmulțește cu 32	RRC	A
	MOV	Nrimp,#1FH	MOV	Adr2,A
	SETB	C	MOV	A,Adr1
	MOV	A,Adr2	RRC	A
	RRC	A	MOV	Adr1,A
	MOV	Adr2,A	CLR	C
	MOV	A,Adr1	MOV	A,Adr2
	RRC	A	RRC	A
	MOV	Adr1,A	MOV	Adr2,A
	CLR	C	MOV	A,Adr1
	MOV	A,Adr2	RRC	A
	RRC	A	MOV	Adr1,A
	MOV	Adr2,A	CLR	C
	MOV	A,Adr1	MOV	A,Adr2
	RRC	A	RRC	A
	MOV	Adr1,A	MOV	Adr2,A
	CLR	C	MOV	A,Adr1
	MOV	A,Adr2	RRC	A
	RRC	A	MOV	Adr1,A

```
CLR C  
MOV A,Adr2  
RRC A  
CPL A  
MOV Adr4,A  
MOV A,Adr1  
RRC A  
CPL A  
ADD A,#0EH  
MOV Adr3,A  
MOV A,Adr4  
ADDC A,#00  
MOV Adr4,A  
SETB TCON.6  
RETI
```

; Subrutina tratează întreruperea dată de  
TIMER1 care generează Nrimp, cu durata  
dintre impulsuri ; programată la Adr4,3

```
M7: SETB P2.0  
CLR TCON.6  
MOV TL1,Adr3  
MOV TH1,Adr4  
SETB TCON.6  
DJNZ Nrimp,M8  
CLR TCON.6  
M8: NOP  
CLR P2.0  
RETI
```

			PR2	EQU	032H
			PR1	EQU	033H
; Program BLOC DE MĂSURĂ			PS4	EQU	034H
ERRORREX M			PS3	EQU	035H
			PS2	EQU	036H
; Tabela cu adrese utilizate			PS1	EQU	037H
DOM	EQU	P0.0	PT4	EQU	038H
DIM	EQU	P0.1	PT3	EQU	039H
CSM	EQU	P0.2	PT2	EQU	03AH
SKM	EQU	P0.3	PTT1	EQU	03BH
		; pinii 8052 alocați memoriei E2	PRA4	EQU	03CH
DSP			PRA3	EQU	03DH
MBUS7	EQU	P1.7	PRA2	EQU	03EH
RSMET	EQU	P2.0	PRA1	EQU	03FH
		; RESET convertoare	PSA4	EQU	040H
SCL	EQU	P2.1	PSA3	EQU	041H
		; liniile magistralei I2C	PSA2	EQU	042H
SDA	EQU	P2.2	PSA1	EQU	043H
RSWD	EQU	P2.3	PTA4	EQU	044H
ELCD	EQU	P2.4	PTA3	EQU	045H
RESFIFO	EQU	P2.6	PTA2	EQU	046H
		; linii comandă FIFO	PTA1	EQU	047H
RWLCD	EQU	P3.6	DFR2	EQU	048H
RDFIFOEQU	P3.7		DFR1	EQU	049H
			DFS2	EQU	04AH
FAN	EQU	020H	DFS1	EQU	04BH
;FAN.0			DFT2	EQU	04CH
marchează decalaj <0			DFT1	EQU	04DH
;FAN.1		; pt.	FR2	EQU	04EH
bucla de recepție cuvinte			FR1	EQU	04FH
;FAN.2			UR2	EQU	050H
;FAN.7			UR1	EQU	051H
			US2	EQU	052H
CONT9	EQU	021H	US1	EQU	053H
		; contor la împărțire	UT2	EQU	054H
CONT10	EQU	022H	UT1	EQU	055H
		; contor bucla de timp recepție	IR2	EQU	056H
CONT5	EQU	023H	IR1	EQU	057H
CONT6	EQU	024H	IS2	EQU	058H
NRC	EQU	025H	IS1	EQU	059H
		; contor nr. cuvinte recepționate	IT2	EQU	05AH
COD	EQU	026H	ITT1	EQU	05BH
CUV	EQU	027H	AUX4	EQU	05CH
S1	EQU	028H	AUX3	EQU	05DH
		; adrese pt. cuvinte de transmis	AUX2	EQU	05EH
S2	EQU	029H	AUX1	EQU	05FH
S3	EQU	02AH			
S4	EQU	02BH			; la 060H se înscrie pachetul de 16 octeți
S5	EQU	02CH			recep de la DSP
S6	EQU	02DH	CIT4	EQU	070H
S7	EQU	02EH			; adrese folosite de subrut.
S8	EQU	02FH	IMPART		
			CIT3	EQU	071H
PR4	EQU	030H	CIT2	EQU	072H
		; zona pt. valori citite din DSP pt.	CIT1	EQU	073H
PR3	EQU	031H	Adr0	EQU	074H
			Adr1	EQU	075H
			Adr2	EQU	076H
Pa,Pr,DECALAJ,FRECVENTA,U,I					

Adr3	EQU	077H	CLR	RESFIFO
Adr4	EQU	078H	SETB	RESFIFO
Adr5	EQU	079H	SETB	RDFIFO ; FIFO
Adr6	EQU	07AH	dezactivat	
; stiva este la E0H			CLR	PSW.5
; cod. cuvinte de comandă ce pot fi			MOV	RCAP2H,#0FFH ;9600
recepționate de BLOC MĂSURĂ CURENT			BAUD;TIMER 2	GENERATOR DE BAUD
	EQU	01DH	MOV	RCAP2L,#0B2H ;PT
TENSIUNE	EQU	02DH	EMISIE&RECEPTIE	
DECALAJ	EQU	03DH	SETB	RCLK
PAC	EQU	04DH	;ATIT LA EMISIE CIT SI LA	
PREAC	EQU	05DH	RECEPTIE	
FRECV	EQU	06AH	SETB	TCLCK
IDENT	EQU	08AH	CLR	EXEN2
			CLR	CT2
RCAP2H	EQU	0CBH	SETB	TR2
;RELATIVE LA TIMER2			;pornește TIMER2	
RCAP2L	EQU	0CAH	CLR	RSMET
T2CON	EQU	0C8H	CLR	P2.3
TF2	EQU	T2CON.7	SETB	P2.3
EXF2	EQU	T2CON.6	LCALL	D1S
RCLK	EQU	T2CON.5	CLR	CSM
TCLCK	EQU	T2CON.4	; low	
EXEN2	EQU	T2CON.3	CLR	SKM
TR2	EQU	T2CON.2	; low	
CT2	EQU	T2CON.1	SETB	DIM
CPRL2	EQU	T2CON.0	; high	
			SETB	DOM
			; high	
ORG	0		CLR	PSW.5
LJMP	PPP		LCALL	D1S
ORG	13H		SETB	IE.7
LJMP	INTEX1		E22:	CLR P2.3
ORG	23H		SETB	P2.3
LJMP	INSER		MOV	CONT5,#00
			MOV	CONT6,#00
; program principal			E11:	JB FAN.7,E12
PPP:	CLR	RSWD	MUL	AB
	SETB	RSWD	INC	CONT10
	MOV	SP,#0E0H	MOV	A,CONT10
; programare stiva			CJNE	A,#0FBH,E11
	MOV	IE,#14h	CLR	FAN.1
	MOV	TMOD,#16H	; au trecut 2ms	
; TIMER0=counter, TIMER1=timer			MOV	CONT10,#00
	MOV	TCON,#55H	MOV	A,CONT5
	MOV	SCON,#50H	ADD	A,#01
; port serial cu viteza dată de			MOV	CONT5,A
	MOV	IP,#00H ;	MOV	A,CONT6
TIMER2 format de 8 biți			ADDC	A,#00
	MOV	NRC,#00	MOV	CONT6,A
	MOV	FAN,#00	CJNE	A,#04H,E11
; locație adresabilă pe bit			CLR	FAN.1
	MOV	P1,#00	CLR	FAN.2
	MOV	CONT10,#00	CLR	FAN.7
	SETB	PSW.3	MOV	NRC,#00
	MOV	R7,#0CH	MOV	CONT5,#00
	CLR	PSW.3	MOV	CONT6,#00
	SETB	RESFIFO		



ET6:	AJMP E11	MOV SBUF,S5
	MOV S3,#00H	E129: JBC SCON.1,E128
	MOV S4,#00H	AJMP E129
	AJMP ET9	E128: CLR P2.3
ET7:	MOV S5,#00H	SETB P2.3
	MOV S6,#00H	MOV SBUF,S6
	AJMP ET10	E131: JBC SCON.1,E130
ET8:	MOV S7,#00H	AJMP E131
	MOV S8,#00H	E130: CLR P2.3
	AJMP E24	SETB P2.3
E13X:	AJMP E13	MOV SBUF,S7
E12:	MOV A,COD	E133: JBC SCON.1,E132
	MOV S1,#0FDH	AJMP E133
	CJNE A,#CURENT,E13X	E132: CLR P2.3
	; se cere transmiterea valorilor de curent	SETB P2.3
	MOV S2,#1DH	MOV SBUF,S8
	MOV A,PR4	E135: JBC SCON.1,E140
	JB ACC.7,ET6	AJMP E135
	MOV AUX2,IR2	E140: CLR FAN.7
	MOV AUX1,IR1	CLR FAN.1
	ACALL DETI	CLR FAN.2
	MOV S3,CIT1	AJMP E22
	MOV S4,CIT2	E13: CJNE A,#TENSIUNE,E23
ET9:	MOV A,PS4	; se cere transmiterea valorilor de U
	JB ACC.7,ET7	MOV S2,#2DH
	MOV AUX2,IS2	MOV AUX2,UR2
	MOV AUX1,IS1	MOV AUX1,UR1
	ACALL DETI	ACALL DETU
	MOV S5,CIT1	MOV S3,CIT1
	MOV S6,CIT2	MOV S4,CIT2
ET10:	MOV A,PT4	MOV AUX2,US2
	JB ACC.7,ET8	MOV AUX1,US1
	MOV AUX2,IT2	ACALL DETU
	MOV AUX1,ITT1	MOV S5,CIT1
	ACALL DETI	MOV S6,CIT2
	MOV S7,CIT1	MOV AUX2,UT2
	MOV S8,CIT2	MOV AUX1,UT1
E24:	MOV SBUF,S1	ACALL DETU
E121:	JBC SCON.1,E120	MOV S7,CIT1
	AJMP E121	MOV S8,CIT2
E120:	CLR P2.3	AJMP E24
	SETB P2.3	SET1: MOV S3,#00H
	MOV SBUF,S2	MOV S4,#00H
E123:	JBC SCON.1,E122	AJMP SET2
	AJMP E123	SET3: MOV S5,#00H
E122:	CLR P2.3	MOV S6,#00H
	SETB P2.3	AJMP SET4
	MOV SBUF,S3	SET5: MOV S7,#00H
E125:	JBC SCON.1,E124	MOV S8,#00H
	AJMP E125	AJMP E24
E124:	CLR P2.3	E23: CJNE A,#DECALAJ,E20
	SETB P2.3	; se cere transmiterea valorilor decalajului pe
	MOV SBUF,S4	faze
E127:	JBC SCON.1,E126	MOV S2,#3DH
	AJMP E127	MOV A,PR4
E126:	CLR P2.3	JB ACC.7,SET1
	SETB P2.3	MOV AUX2,DFR2

	MOV AUX1,DFR1		MOV S6,#00H
	ACALL DETDF		AJMP ET4
	MOV S3,CIT1	ET5:	MOV S7,#00H
	MOV S4,CIT2		MOV S8,#00H
SET2:	MOV A,PS4		AJMP E24
	JB ACC.7,SET3	SE10:	CJNE A,#PREAC,SE11
	MOV AUX2,DFS2		; se cere transmiterea valorilor de P reactivă
	MOV AUX1,DFS1		CLR FAN.3
	ACALL DETDF		MOV S2,#5DH
	MOV S5,CIT1		MOV Adr4,#00H
	MOV S6,CIT2		MOV Adr3,PRA4
SET4:	MOV A,PT4		MOV Adr2,PRA3
	JB ACC.7,SET5		MOV Adr1,PRA2
	MOV AUX2,DFT2		MOV Adr0,PRA1
	MOV AUX1,DFT1		MOV A,PRA4
	ACALL DETDF		JNB ACC.7,ET11
	MOV S7,CIT1		ACALL COMPL
	MOV S8,CIT2		SETB FAN.3
	AJMP E24	ET11:	ACALL DETPA
E20:	CJNE A,#PAC,SE10		JNB FAN.3,ET12
	; se cere transmiterea valorilor de P activă		CLR FAN.3
	MOV S2,#4DH		ACALL COMPL1
	MOV A,PR4	ET12:	MOV S3,CIT1
	JB ACC.7,EST1		MOV S4,CIT2
	MOV Adr4,#00H		MOV Adr4,#00H
	MOV Adr3,PR4		MOV Adr3,PSA4
	MOV Adr2,PR3		MOV Adr2,PSA3
	MOV Adr1,PR2		MOV Adr1,PSA2
	MOV Adr0,PR1		MOV Adr0,PSA1
	ACALL DETP		MOV A,PSA4
	MOV S3,CIT1		JNB ACC.7,ET13
	MOV S4,CIT2		ACALL COMPL
ET2:	MOV A,PS4		SETB FAN.3
	JB ACC.7,ET3	ET13:	ACALL DETPA
	MOV Adr4,#00H		JNB FAN.3,ET14
	MOV Adr3,PS4		CLR FAN.3
	MOV Adr2,PS3		ACALL COMPL1
	MOV Adr1,PS2	ET14:	MOV S5,CIT1
	MOV Adr0,PS1		MOV S6,CIT2
	ACALL DETP		MOV Adr4,#00H
	MOV S5,CIT1		MOV Adr3,PTA4
	MOV S6,CIT2		MOV Adr2,PTA3
ET4:	MOV A,PT4		MOV Adr1,PTA2
	JB ACC.7,ET5		MOV Adr0,PTA1
	MOV Adr4,#00H		MOV A,PTA4
	MOV Adr3,PT4		JNB ACC.7,ET15
	MOV Adr2,PT3		ACALL COMPL
	MOV Adr1,PT2		SETB FAN.3
	MOV Adr0,PTT1	ET15:	ACALL DETPA
	ACALL DETP		JNB FAN.3,ET16
	MOV S7,CIT1		ACALL COMPL1
	MOV S8,CIT2	ET16:	MOV S7,CIT1
	AJMP E24		MOV S8,CIT2
EST1:	MOV S3,#00H		AJMP E24
	MOV S4,#00H	SE11:	CJNE A,#FREC,SE12
	AJMP ET2		; se cere transmiterea valorilor de F
ET3:	MOV S5,#00H		MOV S2,#6AH

	ACALL DETFR	MOV A,#0FDH
	MOV S3,CIT1	; este cuvântul pentru BLOC
	MOV S4,CIT2	MĂSURĂ?
	MOV SBUF,S1	CJNE A,CUV,E4
SE13:	JBC SCON.1,SE14	CLR P0.6
	AJMP SE13	SETB FAN.2
SE14:	CLR P2.3	; da, cuvânt = NP
	SETB P2.3	E8: INC NRC
	MOV SBUF,S2	MOV CONT5,#00
SE17:	JBC SCON.1,SE16	; inițializare bucla de timp
	AJMP SE17	MOV CONT6,#00
SE16:	CLR P2.3	E5: POP ACC
	SETB P2.3	RET
	MOV SBUF,S3	E4: CLR P0.4
SE18:	JBC SCON.1,SE19	SETB FAN.1
	AJMP SE18	; nu e cuvânt pt. blocul de măsură
SE19:	CLR P2.3	RE2: MOV CONT10,#00
	SETB P2.3	; anulez contor de 2ms
	MOV SBUF,S4	AJMP E5
SE20:	JBC SCON.1,SE21	RE3: MOV A,CUV
	AJMP SE20	; al II-lea cuvânt
SE21:	CLR FAN.7	CLR P0.5
	CLR FAN.1	MOV COD,A
	CLR FAN.2	CJNE A,#CURENT,E6
	AJMP E22	E7: SETB FAN.7
SE12:	CJNE A,#IDENT,SE21	; dacă cod MASURARE CURENT
; se cere transmiterea codului de identificare	MOV S2,#0A8H	MOV NRC,#00
	MOV SBUF,S1	AJMP E5
SE23:	JBC SCON.1,SE22	E6: CJNE A,#TENSIUNE,RE10
	AJMP SE23	AJMP E7
SE22:	CLR P2.3	; dacă cod MASURARE
	SETB P2.3	TENSIUNE
	MOV SBUF,S2	RE10: CJNE A,#DECALAJ,ES1
SE24:	JBC SCON.1,SE21	AJMP E7
	AJMP SE24	; dacă cod MASURARE
		DECALAJ
		ES1: CJNE A,#PAC,ES2
		AJMP E7
		; dacă cod MASURARE P
		ACTIVA
		ES2: CJNE A,#PREAC,ES3
		AJMP E7
		; dacă cod MASURARE P
		REACTIVA
		ES3: CJNE A,#FRECV,ES4
		AJMP E7
		; dacă cod MASURARE
		FRECVENTA
		ES4: CJNE A,#IDENT,E7
		AJMP E7
		; dacă cod IDENTIFICARE
		; Subrutina tratează întreruperea pt. citire
		pachete de la DSP
		INTEX1:
		CLR P2.3

; Subrutina tratează întreruperea serială  
INSER:

JBC SCON.0,E1  
RETI

E1: ACALL RECEPTIE  
; recep.carac.serial de forma

FD.ND  
RETI

; Subrutina recepționează caracter serial  
RECEPTIE:

PUSH ACC  
; salvez conținut A  
MOV CUV,SBUF  
; citesc cuvânt recepționat  
CLR P0.7  
JB FAN.1,RE2  
JB FAN.2,RE3  
; primul cuvânt ?

SETB P2.3		AJMP INT4
; citește FIFO la 20 ms		INT5: CJNE A,#02H,INT6
SETB PSW.3		; citește valori decalaj pe cele trei faze
; lucrează în setul II de regiștrii		MOV DFR2,6BH
PUSH ACC		MOV DFR1,6AH
MOV A,R7		MOV DFS2,6DH
JZ INT1X		MOV DFS1,6CH
DEC R7		MOV DFT2,6FH
CJNE R7,#00H,INT4		MOV DFT1,6EH
INT1X:		AJMP INT4
MOV R0,#10H		INT6: CJNE A,#04H,INT7
MOV R1,#60H		; citește frecvența
INT2: CLR RDFIFO		MOV FR2,6DH
MOV @R1,P1		MOV FR1,6CH
SETB RDFIFO		AJMP INT4
INC R1		INT7: CJNE A,#05H,INT4
DJNZ R0,INT2		; citește valori pentru U și I pe cele trei faze
MOV A,60H		MOV UR2,65H
CJNE A,#00H,INT3		MOV UR1,64H
;citește valori de P activă pe cele trei faze		MOV US2,67H
MOV PR4,65H		MOV US1,66H
;octetul c.m.s.		MOV UT2,69H
MOV PR3,64H		MOV UT1,68H
MOV PR2,67H		MOV IR2,6BH
MOV PR1,66H		MOV IR1,6AH
MOV PS4,69H ;octetul		MOV IS2,6DH
c.m.s.		MOV IS1,6CH
MOV PS3,68H		MOV IT2,6FH
MOV PS2,6BH		MOV ITT1,6EH
MOV PS1,6AH		AJMP INT4
MOV PT4,6DH		
;octetul c.m.s.		
MOV PT3,6CH		; calc.cpl.de 2 al unui nr. pe 4 octeți de la
MOV PT2,6FH		adresele Adr3...Adr0
MOV PTT1,6EH		COMPL:
INT4: CLR RESFIFO		MOV A,Adr0
SETB RESFIFO		CPL A
POP ACC		MOV Adr0,A
CLR PSW.3		MOV A,Adr1
RETI		CPL A
INT3: CJNE A,#01H,INT5		MOV Adr1,A
; citește valori de P reactivă pe cele trei faze		MOV A,Adr2
MOV PRA4,65H		CPL A
;octetul c.m.s.		MOV Adr2,A
MOV PRA3,64H		MOV A,Adr3
MOV PRA2,67H		CPL A
MOV PRA1,66H		MOV Adr3,A
MOV PSA4,69H		MOV A,Adr4
;octetul c.m.s.		CPL A
MOV PSA3,68H		MOV Adr4,A
MOV PSA2,6BH		MOV A,Adr0
MOV PSA1,6AH		ADD A,#01H
MOV PTA4,6DH		MOV Adr0,A
;octetul c.m.s.		MOV A,Adr1
MOV PTA3,6CH		ADDC A,#00H
MOV PTA2,6FH		MOV Adr1,A
MOV PTA1,6EH		MOV A,Adr2
		ADDC A,#00H

```

MOV  Adr2,A
MOV  A,Adr3
ADDC A,#00H
MOV  Adr3,A
MOV  A,Adr4
ADDC A,#00H
MOV  Adr4,A
RET

; cpl.de 2 al unui nr. pe 2 octeți de la
adresele CIT2,CIT1
COMPL1:
MOV  A,CIT1
CPL  A
MOV  CIT1,A
MOV  A,CIT2
CPL  A
MOV  CIT2,A
MOV  A,CIT1
ADD  A,#01H
MOV  CIT1,A
MOV  A,CIT2
ADDC A,#00H
MOV  CIT2,A
RET

; calc.F cu formula 4000000/FR2,1 cu
rezultat pe 2 octeți CIT2,1
DETFR:
MOV  Adr0,#00H
MOV  Adr1,#09H
MOV  Adr2,#3DH
MOV  Adr3,#00H
MOV  Adr4,#00H
MOV  Adr5,FR1
MOV  Adr6,FR2
MOV  A,FR2
ORL  A,FR1
JZ   DETFR1
MOV  CIT1,#00H
MOV  CIT2,#00H
RET

DETFR1:
ACALL IMPART
RET

; calc.U după formula U*129/1000 cu
rezultat pe 2 octeți
;
;                               U*86/1000
(U*56/3E8)
; inițial se încarcă în AUX2,1 valorile
corespunzătoare ale U
DETU:
MOV  A,AUX1
MOV  B,#56H
MUL  AB
MOV  Adr0,A
MOV  Adr1,B
MOV  A,AUX2
MOV  B,#56H
MUL  AB
ADD  A,Adr1
MOV  Adr1,A
MOV  A,B
ADDC A,#00
MOV  Adr2,A
MOV  A,AUX3
MOV  B,#4BH
MUL  AB
ADD  A,Adr2
MOV  Adr2,A
MUL  AB
MOV  A,AUX1
MOV  B,#4BH
MUL  AB
ADD  A,Adr1
MOV  Adr1,A
MOV  A,B
ADDC A,#00
MOV  Adr2,A
MOV  A,AUX3
MOV  B,#4BH
MUL  AB
ADD  A,Adr2
MOV  Adr2,A
MUL  AB
MOV  A,AUX1
MOV  B,#05H
MUL  AB
MOV  Adr0,A
MOV  Adr1,B
MOV  A,AUX2
MOV  B,#05H
MUL  AB
ADD  A,Adr1
MOV  Adr1,A
MOV  A,B
ADDC A,#00
MOV  Adr2,A
MOV  AUX1,Adr0
MOV  AUX2,Adr1
MOV  AUX3,Adr2
MOV  A,AUX1
MOV  B,#4BH
MUL  AB
MOV  Adr0,A
MOV  Adr1,B
MOV  A,AUX2
MOV  B,#4BH
MUL  AB
ADD  A,Adr1
MOV  Adr1,A
MOV  A,B
ADDC A,#00
MOV  Adr2,A
MOV  A,AUX3
MOV  B,#4BH
MUL  AB
ADD  A,Adr2
MOV  Adr2,A
MUL  AB
MOV  A,AUX1
MOV  B,#03H
ACALL IMPART
RET

; calc.I cu formula I*375/1000
(U*5*4B/3E8)cu rezultat pe 2 octeți
; inițial se încarcă în AUX2,1 valorile
corespunzătoare ale I
DETI:
MOV  A,AUX1
MOV  B,#05H
MUL  AB
MOV  Adr0,A
MOV  Adr1,B
MOV  A,AUX2
MOV  B,#05H
MUL  AB
ADD  A,Adr1
MOV  Adr1,A
MOV  A,B
ADDC A,#00
MOV  Adr2,A
MOV  AUX1,Adr0
MOV  AUX2,Adr1
MOV  AUX3,Adr2
MOV  A,AUX1
MOV  B,#4BH
MUL  AB
MOV  Adr0,A
MOV  Adr1,B
MOV  A,AUX2
MOV  B,#4BH
MUL  AB
ADD  A,Adr1
MOV  Adr1,A
MOV  A,B
ADDC A,#00
MOV  Adr2,A
MOV  AUX1,Adr0
MOV  AUX2,Adr1
MOV  AUX3,Adr2
MOV  A,AUX1
MOV  B,#4BH
MUL  AB
ADD  A,Adr2
MOV  Adr2,A
MUL  AB
MOV  A,AUX1
MOV  B,#03H
ACALL IMPART
RET

```

<pre> MOV A,B ADDC A,#00H MOV Adr3,A MOV Adr4,#00H MOV Adr5,#0E8H MOV Adr6,#03H ACALL IMPART RET  ; calculează P activă după formula (P/10000)*137*225/40000 ; (P/10000)*137*15/4000 ; cu rez. pe 2 octeți ; inițial se încarcă în Adr3,2,1,0 valorile corespunzătoare ale P DETP: MOV Adr5,#10H MOV Adr6,#27H ACALL IMPART MOV A,CIT1 MOV B,#89H MUL AB MOV Adr0,A MOV Adr1,B MOV A,CIT2 MOV B,#89H MUL AB ADD A,Adr1 MOV Adr1,A MOV A,B ADDC A,#00 MOV Adr2,A MOV AUX1,Adr0 MOV AUX2,Adr1 MOV AUX3,Adr2 MOV A,AUX1 MOV B,#0FH MUL AB MOV Adr0,A MOV Adr1,B MOV A,AUX2 MOV B,#0FH MUL AB ADD A,Adr1 MOV Adr1,A MOV A,B ADDC A,#00H MOV Adr2,A MOV A,AUX3 MOV B,#0FH MUL AB ADD A,Adr2 MOV Adr2,A MOV A,B ADDC A,#00H MOV Adr3,A </pre>	<pre> MOV Adr4,#00H MOV Adr5,#0A0H MOV Adr6,#0FH ACALL IMPART RET  ; calc.Pr după formula (PA/1000)*63*43/64000 ( din cauza P&lt;0) ; (PA/1000)*43*21/32000 ;(P/3E8)*2B*3F/FA.00)cu rezultat pe 2 octeti ; inițial se încarcă în Adr3,2,1,0 valorile corespunzătoare ale P DETPA: MOV Adr5,#0E8H MOV Adr6,#03H ACALL IMPART MOV A,CIT1 MOV B,#2BH MUL AB MOV Adr0,A MOV Adr1,B MOV A,CIT2 MOV B,#2BH MUL AB ADD A,Adr1 MOV Adr1,A MOV A,B ADDC A,#00H MOV Adr2,A MOV A,CIT3 MOV B,#2BH MUL AB ADD A,Adr2 MOV Adr2,A MOV A,B ADDC A,#00H MOV Adr3,A MOV AUX1,Adr0 MOV AUX2,Adr1 MOV AUX3,Adr2 MOV AUX4,Adr3 MOV A,AUX1 MOV B,#15H MUL AB MOV Adr0,A MOV Adr1,B MOV A,AUX2 MOV B,#15H MUL AB ADD A,Adr1 MOV Adr1,A MOV A,B ADDC A,#00 MOV Adr2,A MOV A,AUX3 </pre>
---	--

Program bloc de măsură ERRORREX M

MOV B,#15H	MOV AUX3,Adr2
MUL AB	MOV A,AUX1
ADD A,Adr2	MOV B,#3DH
MOV Adr2,A	MUL AB
MOV A,B	MOV Adr0,A
ADDC A,#00H	MOV Adr1,B
MOV Adr3,A	MOV A,AUX2
MOV A,AUX4	MOV B,#3DH
MOV B,#15H	MUL AB
MUL AB	ADD A,Adr1
ADD A,Adr3	MOV Adr1,A
MOV Adr3,A	MOV A,B
MOV A,B	ADDC A,#00
ADDC A,#00H	MOV Adr2,A
MOV Adr4,A	MOV A,AUX3
MOV Adr5,#00H	MOV B,#3DH
MOV Adr6,#7DH	MUL AB
ACALL IMPART	ADD A,Adr2
RET	MOV Adr2,A
	MOV A,B
	ADDC A,#00H
	MOV Adr3,A
	MOV Adr4,#00H
	MOV Adr5,#10H
	MOV Adr6,#27H
	ACALL IMPART
	JNB FAN.0,DETDF2
	CLR FAN.0
	MOV A,CIT1
	CPL A
	MOV CIT1,A
	MOV A,CIT2
	CPL A
	MOV CIT2,A
	MOV A,CIT1
	ADD A,#01H
	MOV CIT1,A
	MOV A,CIT2
	ADDC A,#00H
	MOV CIT2,A
	DETDF2:
	RET
	; Subrutina împărțire 4 octeți la 2 octeți cu
	rezultat pe 3 octeți
	;
	(Adr4,3,2,1,0:Adr6,5-->CIT4,3,2,1)
	IMPART:
	MOV CIT1,#00
	MOV CIT2,#00
	MOV CIT3,#00
	MOV CIT4,#00
	MOV R6,Adr6
	MOV R5,Adr5
	MOV CONT9,#10H
	E2: MOV R3,Adr4
	MOV R2,Adr3

; calculează DECALAJ FAZE după formula  
(DF\*5\*61/10000)  
;(P\*5\*3D/27.10)cu rezultat pe 2 octeți  
; inițial se încarcă în AUX2,1 valorile  
corespunzătoare ale DF

DETDF:

```

MOV A,AUX2
JNB ACC.7,DETDF1
MOV A,AUX1
CPL A
MOV AUX1,A
MOV A,AUX2
CPL A
MOV AUX2,A
MOV A,AUX1
ADD A,#01H
MOV AUX1,A
MOV A,AUX2
ADDC A,#00H
MOV AUX2,A
SETB FAN.0

```

DETDF1:

```

MOV A,AUX1
MOV B,#05H
MUL AB
MOV Adr0,A
MOV Adr1,B
MOV A,AUX2
MOV B,#05H
MUL AB
ADD A,Adr1
MOV Adr1,A
MOV A,B
ADDC A,#00
MOV Adr2,A
MOV AUX1,Adr0
MOV AUX2,Adr1

```

<pre> E10: MOV R1,Adr2       LCALL INCIT       LCALL DIF       JNC E10       LCALL SUMA       LCALL DECIT       MOV Adr4,R3       MOV Adr3,R2       MOV Adr2,R1       LCALL ROTL       LCALL RCITL       DJNZ CONT9,E2       MOV R3,Adr4       MOV R2,Adr3       MOV R1,Adr2       LCALL INCIT       LCALL DIF       JNC E3       LCALL SUMA       LCALL DECIT E3:   RET  ; Subrutina incrementează câțul INCIT:       MOV A,CIT1       ADD A,#01       MOV CIT1,A       MOV A,CIT2       ADDC A,#00       MOV CIT2,A       MOV A,CIT3       ADDC A,#00       MOV CIT3,A       MOV A,CIT4       ADDC A,#00       MOV CIT4,A       RET  ; subrutina realizează decrementarea câțului DECIT:       CLR C       MOV A,CIT1       SUBB A,#01       MOV CIT1,A       MOV A,CIT2       SUBB A,#00       MOV CIT2,A       MOV A,CIT3       SUBB A,#00       MOV CIT3,A       MOV A,CIT4       SUBB A,#00       MOV CIT4,A       RET  ; Subrutina efectuează dif. dintre R3,2,1- R6,5--&gt;R3,2,1 </pre>	<pre> DIF:  CLR C       MOV A,R6       SUBB A,R5       MOV R1,A       MOV A,R2       SUBB A,R6       MOV R2,A       MOV A,R3       SUBB A,#00       MOV R3,A       RLC A       RET  ; Subrutina efectuează suma dintre R3,2,1+R6,5--&gt;R3,2,1 SUMA:       MOV A,R6       ADD A,R5       MOV R1,A       MOV A,R2       ADDC A,R6       MOV R2,A       MOV A,R3       ADDC A,#00       MOV R3,A       RET  ; Subrutina realiz.rot.la stânga cu 1 bit a unui nr. de la ; Adr4,3,2,1,0 ROTL:       CLR C       MOV A,Adr0       RLC A       MOV Adr0,A       MOV A,Adr1       RLC A       MOV Adr1,A       MOV A,Adr2       RLC A       MOV Adr2,A       MOV A,Adr3       RLC A       MOV Adr3,A       MOV A,Adr4       RLC A       MOV Adr4,A       RET  ; Subrutina rotește câțul cu 1 rang la stânga RCITL:       CLR C       MOV A,CIT1       RLC A       MOV CIT1,A       MOV A,CIT2       RLC A </pre>
--	---



```
MOV CIT2.A
MOV A,CIT3
RLC A
MOV CIT3.A
MOV A,CIT4
RLC A
MOV CIT4.A
RET

D1S: MOV R4,#04H
D12: MOV R5,#0D2H
D11: LCALL D1MS
      DJNZ R5,D11
      DJNZ R4,D12
      RET

D05S: MOV R4,#02H
D052: MOV R5,#0D2H
D051: LCALL D1MS
      DJNZ R5,D051
      DJNZ R4,D052
      RET

D10MS:
      ACALL D1MS
      ACALL D1MS
      ACALL D1MS
      ACALL D1MS
      ACALL D1MS
      ACALL D1MS
      ACALL D1MS
      ACALL D1MS
      ACALL D1MS
      ACALL D1MS
      ACALL D1MS
      ACALL D1MS
      RET

D1MS: CLR P2.3
      SETB P2.3
      MOV A,#0B6H
M1S:  LCALL D4US
      DEC A
      JZ M1S
      RET

D4US: LJMP D41
D41:  AJMP D42
D42:  RET

D2US: RET
```

<pre> ; Program boc de comandă SE 04  ORG 0 LJMP PRGPR ORG 0Bh LJMP TIMERO  PRGPR MOV P0,#0h MOV P2,#0F0h PIX:  INC P2       MOV R0,P2       CJNE R0,#0F7h,PIX       SETB P2.3       MOV P0,#0FDh       MOV P2,#0F4h       SETB P2.3 FIX:  JB P2.4,FLX       MOV P0,#0h       MOV P2,#0F4h       SETB P2.3 MOV IE,#82h       MOV TMOD,#1h       MOV DPTR,#90h ;adresa de inceput a subrutinelor pt. butoane       MOV R1,#30h ;adresa de la care incep sa memorez comanda releelor IETX: MOV @R1,#0h ;* aici initializez zona de memorie unde memorez       INC R1 ;* comanda releelor       CJNE R1,#38h,IETX ;*       MOV R1,#30h       ;restabilesc adresa       MOV R5,#8h ;R5 realizeaza bucla pt. citirea tastaturii       MOV P0,#1h       CLR A TAST: JB P1.0,TEMP FT1:  ADD A,#3h       JB P1.1,TEMP ET2:  ADD A,#3h       JB P1.2,TEMP ET3:  ADD A,#3h       JB P1.3,TEMP ET4:  ADD A,#3h       JB P1.4,TEMP ET5:  ADD A,#3h       JB P1.5,TEMP ET6:  ADD A,#3h       JB P1.6,TEMP ET7:  ADD A,#3h       JB P1.7,TEMP ET8:  ADD A,#3h       PUSH ACC </pre>	<pre> MOV A,P0 RL A INC R1 MOV P0,A POP ACC DJNZ R5,ETI CLR A MOV R1,#30h MOV R5,#8h ETI:  AJMP TAST TEMP: ACALL SRTEMP       JMP @A+DPTR SRTEMP: MOV R6,#32h ;* temporizare de 10ms pentru eliminarea ETI2: MOV R7,#0C8h ;* vibratiei tastelor ETI1: DJNZ R7,ETI1 ;*       DJNZ R6,ETI2 ;*       RET ;* LJMP B1 LJMP B2 LJMP B3 LJMP B4 LJMP B5 LJMP B6 LJMP B7 LJMP B8 LJMP B9 LJMP B10 LJMP B11 LJMP B12 LJMP B13 LJMP B14 LJMP B15 LJMP B16 LJMP B64 LJMP B17 LJMP B18 LJMP B19 LJMP B20 LJMP B21 LJMP B22 LJMP B23 LJMP B24 LJMP B25 LJMP B26 LJMP B27 LJMP B28 LJMP B29 LJMP B30 LJMP B31 LJMP B32 LJMP B33 LJMP B34 LJMP B35 LJMP B36 </pre>
---	---

LJMP B37	LOOP1: MOV R0,P1	;	*
LJMP B38	astcapta eliberarea tastei (a tuturor tastelor)		
LJMP B39	CJNE R0,#0h,LOOP1		
LJMP B40	LJMP FT1	;	se
LJMP B41	intoarce in bucla de citire a tastelor (va citi		
LJMP B42	tasta B2)		
LJMP B43	ER2: LJMP ET2		
LJMP B44	B2: MOV R0,P1	;	B2
LJMP B45	ESTE PENTRU Un		
LJMP B46	CJNE R0,#2h,ER2		
LJMP B47	PUSH P0		
LJMP B48	MOV P0,@R1		
LJMP B49	ANL P0,#0F8h		
LJMP B50	LCALL SRTMP		
LJMP B51	SETB P0.1		
LJMP B52	MOV @R1,P0		
LJMP B53	MOV P2,#0F0h		
LJMP B54	SETB P2.3		
LJMP B55	POP P0		
LJMP B56	NOP		
LJMP B57	LOOP2: MOV R0,P1		
LJMP B58	CJNE R0,#0h,LOOP2		
LJMP B59	AJMP ET2		
LJMP B60	ER3: LJMP ET3		
LJMP B61	B3: MOV R0,P1	;	B3
LJMP B62	ESTE PENTRU 1.1Un		
LJMP B63	CJNE R0,#4h,ER3		
ESTE PENTRU 0.8Un	PUSH P0		
ER1: LJMP FT1	;	daca B1	
nu mai este apasata (in acest moment) se			
trece la B2			
B1: MOV R0,P1	;	*	
verifica daca mai este apasata tasta B1 si			
daca nu			
	CJNE R0,#1h,ER1	;	*(si
daca sint apasate mai multe taste) trece la B2			
PUSH P0	NOP		
	MOV P0,@R1	;	aduce
din memorie setarile anterioare pentru			
primele 8 relec			
	ANL P0,#0F8h		
;reseteaza comenzile incompatibile			
cu apasarea tastei B1			
	LCALL SRTMP		
	SETB P0.0	;	seteaza
comanda corespunzatoare apasarii lui B1			
	MOV @R1,P0	;	salveaza
in memorie comanda corespunzatoare			
apasarii lui B1			
	MOV P2,#0F0h	;	#0F0h
este adresa primelor 8 relec			
	SETB P2.3		
;invalidarea setarii releelor prin			
blocarea scrierii in 74LS373			
	POP P0		
	NOP		
	LOOP3: MOV R0,P1		
	CJNE R0,#0h,LOOP3		
	AJMP ET3		
ER4: LJMP ET4			
B4: MOV R0,P1	;	AICI	
INCEP CELE 5 GAME DE TENSIUNI			
	CJNE R0,#8h,ER4		
	;(380/1.41,100/1.41,etc)		
	PUSH P0		
	MOV P0,@R1		
	ANL P0,#7h		
	LCALL SRTMP		
	SETB P0.3		
	MOV @R1,P0		
	MOV P2,#0F0h		
	SETB P2.3		
	POP P0		
	NOP		
	LOOP4: MOV R0,P1		

	CJNE R0,#0h,LOOP4	SETB P0.7
	AJMP ET4	MOV @R1,P0
ER5:	LJMP ET5	MOV P2,#0F0h
B5:	MOV R0,P1	SETB P2.3
	CJNE R0,#10h,ER5	POP P0
	PUSH P0	NOP
	MOV P0,@R1	LOOP8: MOV R0,P1
	ANL P0,#7h	CJNE R0,#0h,LOOP8
	LCALL SRTMP	AJMP ET8
	SETB P0.4	ER9: LJMP FT1
	MOV @R1,P0	B9: MOV R0,P1
	MOV P2,#0F0h	;SUCESIUNEA DIRECTA A
	SETB P2.3	FAZELOR RST
	POP P0	CJNE R0,#1h,ER9
	NOP	PUSH P0
LOOP5:	MOV R0,P1	MOV P0,@R1
	CJNE R0,#0h,LOOP5	CLR P0.1
	AJMP ET5	LCALL SRTMP
ER:	LJMP ET6	MOV @R1,P0
B6:	MOV R0,P1	MOV P2,#0F1h
	CJNE R0,#20h,ER6	SETB P2.3
	PUSH P0	POP P0
	MOV P0,@R1	NOP
	ANL P0,#7h	LOOP9: MOV R0,P1
	LCALL SRTMP	CJNE R0,#0h,LOOP9
	SETB P0.5	AJMP FT1
	MOV @R1,P0	ER10: LJMP ET2
	MOV P2,#0F0h	B10: MOV R0,P1
	SETB P2.3	;SUCESIUNEA INVERSA A
	POP P0	FAZELOR SRT
	NOP	CJNE R0,#2h,ER10
LOOP6:	MOV R0,P1	PUSH P0
	CJNE R0,#0h,LOOP6	MOV P0,@R1
	AJMP ET6	SETB P0.1
ER7:	LJMP ET7	LCALL SRTMP
B7:	MOV R0,P1	MOV @R1,P0
	CJNE R0,#40h,ER7	MOV P2,#0F1h
	PUSH P0	SETB P2.3
	MOV P0,@R1	POP P0
	ANL P0,#7h	NOP
	LCALL SRTMP	LOOP10:
	SETB P0.6	MOV R0,P1
	MOV @R1,P0	CJNE R0,#0h,LOOP10
	MOV P2,#0F0h	AJMP ET2
	SETB P2.3	ER11: LJMP ET3
	POP P0	;DE AICI INCEP COMENZILE DE
	NOP	SELECTARE A REGIMULUI DE
LOOP7:	MOV R0,P1	MASURA
	CJNE R0,#0h,LOOP7	B11: MOV R0,P1
	AJMP ET7	;REGIMUL ACTIV CU 4 FIRE
ER8:	LJMP ET8	"3W"
B8:	MOV R0,P1	CJNE R0,#4h,ER11
	CJNE R0,#80h,ER8	PUSH P0
	PUSH P0	MOV P0,@R1
	MOV P0,@R1	ANL P0,#0C3h
	ANL P0,#7h	LCALL SRTMP
	LCALL SRTMP	SETB P0.5

MOV @R1,P0	MOV @R1,P0
MOV P2,#0F1h	MOV P2,#0F1h
SETB P2.3	SETB P2.3
POP P0	POP P0
NOP	NOP
LOOP11:	LOOP14:
MOV R0,P1	MOV R0,P1
CJNE R0,#0h,LOOP11	CJNE R0,#0h,LOOP14
AJMP ET3	LJMP ET6
ER12: LJMP ET4	B15: LJMP ET7
B12: MOV R0,P1	B16: LJMP ET8
;REGIMUL ACTIV CU 3 FIRE	B64: LJMP FT1
"2W"	ER17: LJMP ET2 ;DE
CJNE R0,#8h,ER12	AICI INCEP DOMENIILE DE CURENTI
PUSH P0	B17: MOV R0,P1 ;0,002A
MOV P0,@R1	CJNE R0,#2h,ER17
ANL P0,#0C3h	PUSH P0
LCALL SRTMP	MOV P0,#0FDh ;aici va
SETB P0.4	fi cuplat si releul pentru viteza mare
MOV @R1,P0	MOV P2,#0F4h
MOV P2,#0F1h	SETB P2.3
SETB P2.3	EJ: JB P2.4,EJ ;aici
POP P0	asteapta ajungerea la zero a celor 6 motoare
NOP	MOV P0,#0h ;* aici se
LOOP12:	sterg toate celelalte domenii de curenti
MOV R0,P1	MOV P2,#0F2h ;*
CJNE R0,#0h,LOOP12	(pentru ca nu pot fi apasate doua domenii in
LJMP ET4	acelasi
ER13: LJMP ET5	MOV P2,#0F3h ;* timp)
B13: MOV R0,P1	MOV P2,#0F4h ;* aici
;REGIMUL REACTIV CU 4 FIRE	deconecteaza relele ce comanda cele 6
"3VAR"	motoare
CJNE R0,#10h,ER13	SETB P2.3 ;* aici
PUSH P0	blocheaza scrierea in latch-uri
MOV P0,@R1	MOV P0,#1h ;* aici se
ANL P0,#0C3h	cupleaza releul pentru domenii mai mici sau
LCALL SRTMP	MOV P2,#0F3h ;* egale
SETB P0.3	cu 5A
MOV @R1,P0	SETB P2.3 ;*
MOV P2,#0F1h	MOV P0,#3h ;* aici va
SETB P2.3	fi cuplat si releul de 1/100
POP P0	MOV P2,#0F2h ;*
NOP	SETB P2.3
LOOP13:	POP P0
MOV R0,P1	NOP
CJNE R0,#0h,LOOP13	LOOP17:
LJMP ET5	MOV R0,P1
ER14: LJMP ET6	CJNE R0,#0h,LOOP17
B14: MOV R0,P1	LJMP ET2
;REGIMUL REACTIV CU 3 FIRE	ER18: LJMP ET3
"2VAR"	B18: MOV R0,P1 ;0,005A
CJNE R0,#20h,ER14	CJNE R0,#4h,ER18
PUSH P0	PUSH P0
MOV P0,@R1	MOV P0,#0FDh
ANL P0,#0C3h	MOV P2,#0F4h
LCALL SRTMP	SETB P2.3
SETB P0.2	EJ1: JB P2.4,EJ1

MOV P0,#0h	MOV P0,#1h
MOV P2,#0F2h	MOV P2,#0F3h
MOV P2,#0F3h	SETB P2.3
MOV P2,#0F4h	MOV P0,#11h
SETB P2.3	MOV P2,#0F2h
MOV P0,#1h	SETB P2.3
MOV P2,#0F3h	POP P0
SETB P2.3	NOP
MOV P0,#5h	LOOP20:
MOV P2,#0F2h	MOV R0,P1
SETB P2.3	CJNE R0,#0h,LOOP20
POP P0	LJMP ET5
NOP	ER21: LJMP ET6
LOOP18:	B21: MOV R0,P1
MOV R0,P1	;0,05A
CJNE R0,#0h,LOOP18	CJNE R0,#20h,ER21
LJMP ET3	PUSH P0
ER19: LJMP ET4	MOV P0,#0FDh
B19: MOV R0,P1	MOV P2,#0F4h
;0,01A	SETB P2.3
CJNE R0,#8h,ER19	EJ4: JB P2.4,EJ4
PUSH P0	MOV P0,#0h
MOV P0,#0FDh	MOV P2,#0F2h
MOV P2,#0F4h	MOV P2,#0F3h
SETB P2.3	MOV P2,#0F4h
EJ2: JB P2.4,EJ2	SETB P2.3
MOV P0,#0h	MOV P0,#1h
MOV P2,#0F2h	MOV P2,#0F3h
MOV P2,#0F3h	SETB P2.3
MOV P2,#0F4h	MOV P0,#21h
SETB P2.3	MOV P2,#0F2h
MOV P0,#1h	SETB P2.3
MOV P2,#0F3h	POP P0
SETB P2.3	NOP
MOV P0,#9h	LOOP21:
MOV P2,#0F2h	MOV R0,P1
SETB P2.3	CJNE R0,#0h,LOOP21
POP P0	LJMP ET6
NOP	ER22: LJMP ET7
LOOP19:	B22: MOV R0,P1
MOV R0,P1	;0,1A
CJNE R0,#0h,LOOP19	CJNE R0,#40h,ER22
LJMP ET4	PUSH P0
ER20: LJMP ET5	MOV P0,#0FDh
B20: MOV R0,P1	MOV P2,#0F4h
;0,02A	SETB P2.3
CJNE R0,#10h,ER20	EJ5: JB P2.4,EJ5
PUSH P0	MOV P0,#0h
MOV P0,#0FDh	MOV P2,#0F2h
MOV P2,#0F4h	MOV P2,#0F3h
SETB P2.3	MOV P2,#0F4h
EJ3: JB P2.4,EJ3	SETB P2.3
MOV P0,#0h	MOV P0,#1h
MOV P2,#0F2h	MOV P2,#0F3h
MOV P2,#0F3h	SETB P2.3
MOV P2,#0F4h	MOV P0,#41h
SETB P2.3	MOV P2,#0F2h

SETB P2.3	CJNE R0,#0h,LOOP24
POP P0	LJMP FT1
NOP	ER25: JMP ET2
LOOP22:	B25: MOV R0,P1
MOV R0,P1	;1A
CJNE R0,#0h,LOOP22	CJNE R0,#2h,ER25
LJMP ET7	PUSH P0
ER23: LJMP ET8	MOV P0,#0FDh
B23: MOV R0,P1	MOV P2,#0F4h
;0,2A	SETB P2.3
CJNE R0,#80h,ER23	EJ8: JB P2.4,EJ8
PUSH P0	MOV P0,#0h
MOV P0,#0FDh	MOV P2,#0F2h
MOV P2,#0F4h	MOV P2,#0F3h
SETB P2.3	MOV P2,#0F4h
EJ6: JB P2.4,EJ6	SETB P2.3
MOV P0,#0h	MOV P0,#1h
MOV P2,#0F2h	MOV P2,#0F3h
MOV P2,#0F3h	SETB P2.3
MOV P2,#0F4h	MOV P0,#8h
SETB P2.3	MOV P2,#0F2h
MOV P0,#1h	SETB P2.3
MOV P2,#0F3h	POP P0
SETB P2.3	NOP
MOV P0,#2h	LOOP25:
MOV P2,#0F2h	MOV R0,P1
SETB P2.3	CJNE R0,#0h,LOOP25
POP P0	LJMP ET2
NOP	ER26: LJMP ET3
LOOP23:	B26: MOV R0,P1
MOV R0,P1	;2A
CJNE R0,#0h,LOOP23	CJNE R0,#4h,ER26
LJMP ET8	PUSH P0
ER24: LJMP FT1	MOV P0,#0FDh
B24: MOV R0,P1	MOV P2,#0F4h
;0,5A	SETB P2.3
CJNE R0,#1h,ER24	EJ9: JB P2.4,EJ9
PUSH P0	MOV P0,#0h
MOV P0,#0FDh	MOV P2,#0F2h
MOV P2,#0F4h	MOV P2,#0F3h
SETB P2.3	MOV P2,#0F4h
EJ7: JB P2.4,EJ7	SETB P2.3
MOV P0,#0h	MOV P0,#1h
MOV P2,#0F2h	MOV P2,#0F3h
MOV P2,#0F3h	SETB P2.3
MOV P2,#0F4h	MOV P0,#10h
SETB P2.3	MOV P2,#0F2h
MOV P0,#1h	SETB P2.3
MOV P2,#0F3h	POP P0
SETB P2.3	NOP
MOV P0,#4h	LOOP26:
MOV P2,#0F2h	MOV R0,P1
SETB P2.3	CJNE R0,#0h,LOOP26
POP P0	LJMP ET3
NOP	ER27: JMP ET4
LOOP24:	B27: MOV R0,P1
MOV R0,P1	;5A

```

CJNE R0,#8h,ER27
PUSH P0
MOV P0,#0FDh
MOV P2,#0F4h
SETB P2.3
EJ10: JB P2.4,EJ10
MOV P0,#0h
MOV P2,#0F2h
MOV P2,#0F3h
MOV P2,#0F4h
SETB P2.3
MOV P0,#1h
MOV P2,#0F3h
SETB P2.3
MOV P0,#20h
MOV P2,#0F2h
SETB P2.3
POP P0
NOP
LOOP27:
MOV R0,P1
CJNE R0,#0h,LOOP27
LJMP ET4
ER28: LJMP ET5
B28: MOV R0,P1
;10A
CJNE R0,#10h,ER28
PUSH P0
MOV P0,#0FDh
MOV P2,#0F4h
SETB P2.3
EJ11: JB P2.4,EJ11
MOV P0,#0h
MOV P2,#0F2h
MOV P2,#0F3h
MOV P2,#0F4h
SETB P2.3
MOV P0,#2h
MOV P2,#0F3h
SETB P2.3
MOV P0,#40h
MOV P2,#0F2h
SETB P2.3
POP P0
NOP
LOOP28:
MOV R0,P1
CJNE R0,#0h,LOOP28
LJMP ET5
ER29: LJMP ET6
B29: MOV R0,P1
;20A
CJNE R0,#20h,ER29
PUSH P0
MOV P0,#0FDh
MOV P2,#0F4h
SETB P2.3
EJ12: JB P2.4,EJ12
MOV P0,#0h
MOV P2,#0F2h
MOV P2,#0F3h
MOV P2,#0F4h
SETB P2.3
MOV P0,#2h
MOV P2,#0F3h
SETB P2.3
MOV P0,#80h
MOV P2,#0F2h
SETB P2.3
POP P0
NOP
LOOP29:
MOV R0,P1
CJNE R0,#0h,LOOP29
LJMP ET6
ER30: LJMP ET7
B30: MOV R0,P1
;50A
CJNE R0,#40h,ER30
PUSH P0
MOV P0,#0FDh
MOV P2,#0F4h
SETB P2.3
EJ13: JB P2.4,EJ13
MOV P0,#0h
MOV P2,#0F2h
MOV P2,#0F3h
MOV P2,#0F4h
SETB P2.3
MOV P0,#6h
MOV P2,#0F3h
SETB P2.3
POP P0
NOP
LOOP30:
MOV R0,P1
CJNE R0,#0h,LOOP30
LJMP ET7
ER31: LJMP ET8
B31: MOV R0,P1
;100A
CJNE R0,#80h,ER31
PUSH P0
MOV P0,#0FDh
MOV P2,#0F4h
SETB P2.3
EJ14: JB P2.4,EJ14
MOV P0,#0h
MOV P2,#0F2h
MOV P2,#0F3h
MOV P2,#0F4h
SETB P2.3
MOV P0,#0Ah
MOV P2,#0F3h

```



<pre> SETB P2.3 POP P0 NOP LOOP31: MOV R0,P1     CJNE R0,#0h,LOOP31     LJMP ET8 ER32:  LJMP FT1          ;DE AICI INCEP COMENZILE DE REGLARE CURENTI B32:   MOV R0,P1         CJNE R0,#1h,ER32         PUSH P0         MOV TL0,#0AFh         MOV TH0,#3Ch         MOV R2,#20h         SETB TR0         MOV P0,#2h         MOV P2,#0F4h         SETB P2.3         LCALL SRTMP         MOV P0,#6h      ;sens crescator         MOV R3,P0         MOV P2,#0F4h         SETB P2.3         POP P0         NOP LOOP32: MOV R0,P1     CJNE R0,#0h,LOOP32     CLR TR0     CLR P0.0     PUSH P0     MOV P0,#0h     MOV P2,#0F4h     SETB P2.3     POP P0     LJMP FT1 ER33:  LJMP ET2 B33:   MOV R0,P1         CJNE R0,#2h,ER33         PUSH P0         MOV TL0,#0AFh         MOV TH0,#3Ch         MOV R2,#20h         SETB TR0         MOV P0,#0h         MOV P2,#0F4h         SETB P2.3         LCALL SRTMP         MOV P0,#4h      ;scns descrescator         MOV R3,P0         MOV P2,#0F4h         SETB P2.3         POP P0         NOP </pre>	<pre> LOOP33: MOV R0,P1     CJNE R0,#0h,LOOP33     CLR TR0     CLR P0.0     PUSH P0     MOV P0,#0h     MOV P2,#0F4h     SETB P2.3     POP P0     LJMP ET2 ER34:  LJMP ET3 B34:   MOV R0,P1         CJNE R0,#4h,ER34         PUSH P0         MOV TL0,#0AFh         MOV TH0,#3Ch         MOV R2,#20h         SETB TR0         MOV P0,#2h         MOV P2,#0F4h         SETB P2.3         LCALL SRTMP         MOV P0,#0Ah         MOV R3,P0         MOV P2,#0F4h         SETB P2.3         POP P0         NOP LOOP34: MOV R0,P1     CJNE R0,#0h,LOOP34     CLR TR0     CLR P0.0     PUSH P0     MOV P0,#0h     MOV P2,#0F4h     SETB P2.3     POP P0     LJMP ET3 ER35:  LJMP ET4 B35:   MOV R0,P1         CJNE R0,#8h,ER35         PUSH P0         MOV TL0,#0AFh         MOV TH0,#3Ch         MOV R2,#20h         SETB TR0         MOV P0,#0h         MOV P2,#0F4h         SETB P2.3         LCALL SRTMP         MOV P0,#8h         MOV R3,P0         MOV P2,#0F4h         SETB P2.3         POP P0 </pre>
---	---

<pre> NOP LOOP35: MOV R0,P1     CJNE R0,#0h,LOOP35     CLR TR0     CLR P0.0     PUSH P0     MOV P0,#0h     MOV P2,#0F4h     SETB P2.3     POP P0     LJMP ET4 ER36:  LJMP ET5 B36:  MOV R0,P1     CJNE R0,#10h,ER36     PUSH P0     MOV TL0,#0AFh     MOV TH0,#3Ch     MOV R2,#20h     SETB TR0     MOV P0,#2h     MOV P2,#0F4h     SETB P2.3     LCALL SRTMP     MOV P0,#12h     MOV R3,P0     MOV P2,#0F4h     SETB P2.3     POP P0     NOP LOOP36: MOV R0,P1     CJNE R0,#0h,LOOP36     CLR TR0     CLR P0.0     PUSH P0     MOV P0,#0h     MOV P2,#0F4h     SETB P2.3     POP P0     LJMP ET5 ER37:  LJMP ET6 B37:  MOV R0,P1     CJNE R0,#20h,ER37     PUSH P0     MOV TL0,#0AFh     MOV TH0,#3Ch     MOV R2,#20h     SETB TR0     MOV P0,#0h     MOV P2,#0F4h     SETB P2.3     LCALL SRTMP     MOV P0,#10h     MOV R3,P0     MOV P2,#0F4h     SETB P2.3 </pre>	<pre> POP P0 NOP LOOP37: MOV R0,P1     CJNE R0,#0h,LOOP37     CLR TR0     CLR P0.0     PUSH P0     MOV P0,#0h     MOV P2,#0F4h     SETB P2.3     POP P0     LJMP ET6 ER38:  LJMP ET7 B38:  MOV R0,P1     CJNE R0,#40h,ER38     PUSH P0     MOV TL0,#0AFh     MOV TH0,#3Ch     MOV R2,#20h     SETB TR0     MOV P0,#2h     MOV P2,#0F4h     SETB P2.3     LCALL SRTMP     MOV P0,#22h     MOV R3,P0     MOV P2,#0F4h     SETB P2.3     POP P0     NOP LOOP38: MOV R0,P1     CJNE R0,#0h,LOOP38     CLR TR0     CLR P0.0     PUSH P0     MOV P0,#0h     MOV P2,#0F4h     SETB P2.3     POP P0     LJMP ET7 ER39:  LJMP ET8 B39:  MOV R0,P1     CJNE R0,#80h,ER39     PUSH P0     MOV TL0,#0AFh     MOV TH0,#3Ch     MOV R2,#20h     SETB TR0     MOV P0,#0h     MOV P2,#0F4h     SETB P2.3     LCALL SRTMP     MOV P0,#20h     MOV R3,P0     MOV P2,#0F4h </pre>
---	--

SETB P2.3	MOV P2,#0F4h
POP P0	SETB P2.3
NOP	POP P0
LOOP39:	NOP
MOV R0,P1	LOOP41:
CJNE R0,#0h,LOOP39	MOV R0,P1
CLR TR0	CJNE R0,#0h,LOOP41
CLR P0.0	CLR TR0
PUSH P0	CLR P0.0
MOV P0,#0h	PUSH P0
MOV P2,#0F4h	MOV P0,#0h
SETB P2.3	MOV P2,#0F4h
POP P0	SETB P2.3
LJMP ET8	POP P0
ER40: LJMP FT1	LJMP ET2
B40: MOV R0,P1	ER42: LJMP ET3
CJNE R0,#1h,ER40	B42: MOV R0,P1
PUSH P0	CJNE R0,#4h,ER42
MOV TL0,#0AFh	PUSH P0
MOV TH0,#3Ch	MOV TL0,#0AFh
MOV R2,#20h	MOV TH0,#3Ch
SETB TR0	MOV R2,#20h
MOV P0,#2h	SETB TR0
MOV P2,#0F4h	MOV P0,#2h
SETB P2.3	MOV P2,#0F4h
LCALL SRTMP	SETB P2.3
MOV P0,#42h	LCALL SRTMP
MOV R3,P0	MOV P0,#82h
MOV P2,#0F4h	MOV R3,P0
SETB P2.3	MOV P2,#0F4h
POP P0	SETB P2.3
NOP	POP P0
LOOP40:	NOP
MOV R0,P1	LOOP42:
CJNE R0,#0h,LOOP40	MOV R0,P1
CLR TR0	CJNE R0,#0h,LOOP42
CLR P0.0	CLR TR0
PUSH P0	CLR P0.0
MOV P0,#0h	PUSH P0
MOV P2,#0F4h	MOV P0,#0h
SETB P2.3	MOV P2,#0F4h
POP P0	SETB P2.3
LJMP FT1	POP P0
ER41: LJMP ET2	LJMP ET3
B41: MOV R0,P1	ER43: LJMP ET4
CJNE R0,#2h,ER41	B43: MOV R0,P1
PUSH P0	CJNE R0,#8h,ER43
MOV TL0,#0AFh	PUSH P0
MOV TH0,#3Ch	MOV TL0,#0AFh
MOV R2,#20h	MOV TH0,#3Ch
SETB TR0	MOV R2,#20h
MOV P0,#0h	SETB TR0
MOV P2,#0F4h	MOV P0,#0h
SETB P2.3	MOV P2,#0F4h
LCALL SRTMP	SETB P2.3
MOV P0,#40h	LCALL SRTMP
MOV R3,P0	MOV P0,#80h

MOV R3,P0	PUSH P0
MOV P2,#0F4h	MOV P0,#0h
SETB P2.3	MOV P2,#0F4h
POP P0	MOV P2,#0F5h
NOP	SETB P2.3
LOOP43:	POP P0
MOV R0,P1	LJMP ET6
CJNE R0,#0h,LOOP43	ER46: LJMP ET7
CLR TR0	B46: MOV R0,P1
CLR P0.0	CJNE R0,#40h,ER46
PUSH P0	PUSH P0
MOV P0,#0h	MOV P0,#2h
MOV P2,#0F4h	MOV P2,#0F4h
SETB P2.3	SETB P2.3
POP P0	LCALL SRTMP
LJMP ET4	MOV P0,#2h
ER44: LJMP ET5 ;DE AICI	MOV P2,#0F5h
INCEP COMENZILE DE REGLARE	SETB P2.3
TENSIUNI	POP P0
B44: MOV R0,P1	NOP
CJNE R0,#10h,ER44	LOOP46:
PUSH P0	MOV R0,P1
MOV P0,#2h	CJNE R0,#0h,LOOP46
MOV P2,#0F4h	PUSH P0
SETB P2.3	MOV P0,#0h
LCALL SRTMP	MOV P2,#0F4h
MOV P0,#1h	MOV P2,#0F5h
MOV P2,#0F5h	SETB P2.3
SETB P2.3	POP P0
POP P0	LJMP ET7
NOP	ER47: LJMP ET8
LOOP44:	B47: MOV R0,P1
MOV R0,P1	CJNE R0,#80h,ER47
CJNE R0,#0h,LOOP44	PUSH P0
PUSH P0	MOV P0,#0h
MOV P0,#0h	MOV P2,#0F4h
MOV P2,#0F4h	SETB P2.3
MOV P2,#0F5h	LCALL SRTMP
SETB P2.3	MOV P0,#2h
POP P0	MOV P2,#0F5h
LJMP ET5	SETB P2.3
ER45: LJMP ET6	POP P0
B45: MOV R0,P1	NOP
CJNE R0,#20h,ER45	LOOP47:
PUSH P0	MOV R0,P1
MOV P0,#0h	CJNE R0,#0h,LOOP47
MOV P2,#0F4h	PUSH P0
SETB P2.3	MOV P0,#0h
LCALL SRTMP	MOV P2,#0F4h
MOV P0,#1h	MOV P2,#0F5h
MOV P2,#0F5h	SETB P2.3
SETB P2.3	POP P0
POP P0	LJMP ET8
NOP	ER48: LJMP FT1
LOOP45:	B48: MOV R0,P1
MOV R0,P1	CJNE R0,#1h,ER48
CJNE R0,#0h,LOOP45	PUSH P0

MOV P0,#2h	CJNE R0,#0h,LOOP50
MOV P2,#0F4h	PUSH P0
SETB P2.3	MOV P0,#0h
LCALL SRTMP	MOV P2,#0F4h
MOV P0,#4h	MOV P2,#0F5h
MOV P2,#0F5h	SETB P2.3
SETB P2.3	POP P0
POP P0	LJMP ET3
NOP	ER51: LJMP ET4
LOOP48:	B51: MOV R0,P1
MOV R0,P1	CJNE R0,#8h,ER51
CJNE R0,#0h,LOOP48	PUSH P0
PUSH P0	MOV P0,#0h
MOV P0,#0h	MOV P2,#0F4h
MOV P2,#0F4h	SETB P2.3
MOV P2,#0F5h	LCALL SRTMP
SETB P2.3	MOV P0,#8h
POP P0	MOV P2,#0F5h
LJMP FT1	SETB P2.3
ER49: LJMP ET2	POP P0
B49: MOV R0,P1	NOP
CJNE R0,#2h,ER49	LOOP51:
PUSH P0	MOV R0,P1
MOV P0,#0h	CJNE R0,#0h,LOOP51
MOV P2,#0F4h	PUSH P0
SETB P2.3	MOV P0,#0h
LCALL SRTMP	MOV P2,#0F4h
MOV P0,#4h	MOV P2,#0F5h
MOV P2,#0F5h	SETB P2.3
SETB P2.3	POP P0
POP P0	LJMP ET4
NOP	ER52: LJMP ET5
LOOP49:	B52: MOV R0,P1
MOV R0,P1	CJNE R0,#10h,ER52
CJNE R0,#0h,LOOP49	PUSH P0
PUSH P0	MOV P0,#2h
MOV P0,#0h	MOV P2,#0F4h
MOV P2,#0F4h	SETB P2.3
MOV P2,#0F5h	LCALL SRTMP
SETB P2.3	MOV P0,#10h
POP P0	MOV P2,#0F5h
LJMP ET2	SETB P2.3
ER50: LJMP ET3	POP P0
B50: MOV R0,P1	NOP
CJNE R0,#4h,ER50	LOOP52:
PUSH P0	MOV R0,P1
MOV P0,#2h	CJNE R0,#0h,LOOP52
MOV P2,#0F4h	PUSH P0
SETB P2.3	MOV P0,#0h
LCALL SRTMP	MOV P2,#0F4h
MOV P0,#8h	MOV P2,#0F5h
MOV P2,#0F5h	SETB P2.3
SETB P2.3	POP P0
POP P0	LJMP ET5
NOP	ER53: LJMP ET6
LOOP50:	B53: MOV R0,P1
MOV R0,P1	CJNE R0,#20h,ER53

PUSH P0			LJMP ET8
MOV P0,#0h		ER56:	LJMP FT1
MOV P2,#0F4h		B56:	MOV R0,P1
SETB P2.3			;cuplarea/decuplarea sarcinii la faza
LCALL SRTMP		R	
MOV P0,#10h			CJNE R0,#1h,ER56
MOV P2,#0F5h			PUSH P0
SETB P2.3			MOV P0,@R1
POP P0			CPL P0.0
NOP			MOV @R1,P0
LOOP53:			MOV P2,#0F6h
MOV R0,P1			SETB P2.3
CJNE R0,#0h,LOOP53			POP P0
PUSH P0			NOP
MOV P0,#0h		LOOP56:	
MOV P2,#0F4h			MOV R0,P1
MOV P2,#0F5h			CJNE R0,#0h,LOOP56
SETB P2.3			LJMP FT1
POP P0		ER57:	LJMP ET2
LJMP ET6		B57:	MOV R0,P1
ER54: LJMP ET7			;cuplarea/decuplarea sarcinii la faza
B54: MOV R0,P1	;reglare	S	
cosFI in sens capacitiv			CJNE R0,#2h,ER57
CJNE R0,#40h,ER54			PUSH P0
PUSH P0			MOV P0,@R1
MOV P0,#20h			CPL P0.1
MOV P2,#0F5h			MOV @R1,P0
SETB P2.3			MOV P2,#0F6h
POP P0			SETB P2.3
NOP			POP P0
LOOP54:			NOP
MOV R0,P1		LOOP57:	
CJNE R0,#0h,LOOP54			MOV R0,P1
PUSH P0			CJNE R0,#0h,LOOP57
MOV P0,#0h			LJMP ET2
MOV P2,#0F5h		ER58:	LJMP ET3
SETB P2.3		B58:	MOV R0,P1
POP P0			;cuplarea/decuplarea sarcinii la faza
LJMP ET7		T	
ER55: LJMP ET8			CJNE R0,#4h,ER58
B55: MOV R0,P1	;reglare		PUSH P0
cosFI in sens inductiv			MOV P0,@R1
CJNE R0,#80h,ER55			CPL P0.2
PUSH P0			MOV @R1,P0
MOV P0,#40h			MOV P2,#0F6h
MOV P2,#0F5h			SETB P2.3
SETB P2.3			POP P0
POP P0			NOP
NOP		LOOP58:	
LOOP55:			MOV R0,P1
MOV R0,P1			CJNE R0,#0h,LOOP58
CJNE R0,#0h,LOOP55			LJMP ET3
PUSH P0		ER59:	LJMP ET4
MOV P0,#0h		B59:	MOV R0,P1
MOV P2,#0F5h			;decuplarea trifazata a sarcinii
SETB P2.3			CJNE R0,#8h,ER59
POP P0			PUSH P0

```

MOV P0,@R1
ANL P0,#0F8h
;TREBUIE TEMPORIZARI ?
MOV @R1,P0
MOV P2,#0F6h
SETB P2.3
POP P0
NOP
LOOP59:
MOV R0,P1
CJNE R0,#0h,LOOP59
LJMP ET4
ER60: LJMP ET5
;cuplarea/decuplarea caii de
tensiune la faza R
B60: MOV R0,P1
CJNE R0,#10h,ER60
PUSH P0
MOV P0,@R1
CPL P0.3
MOV @R1,P0
MOV P2,#0F6h
SETB P2.3
POP P0
NOP
LOOP60:
MOV R0,P1
CJNE R0,#0h,LOOP60
LJMP ET5
ER61: LJMP ET6
;decuplarea /decuplarea caii de
tensiune la faza S
B61: MOV R0,P1
CJNE R0,#20h,ER61
PUSH P0
MOV P0,@R1
CPL P0.4
MOV @R1,P0
MOV P2,#0F6h
SETB P2.3
POP P0
NOP
LOOP61:
MOV R0,P1
CJNE R0,#0h,LOOP61
LJMP ET6
ER62: LJMP ET7
;cuplarea/decuplarea caii de
tensiune la faza T
B62: MOV R0,P1
CJNE R0,#40h,ER62
PUSH P0
MOV P0,@R1
CPL P0.5
MOV @R1,P0
MOV P2,#0F6h
SETB P2.3

POP P0
NOP
LOOP62:
MOV R0,P1
CJNE R0,#0h,LOOP62
LJMP ET7
ER63: LJMP ET8
;decuplarea trifazata a caii de
tensiune
B63: MOV R0,P1
CJNE R0,#80h,ER63
PUSH P0
MOV P0,@R1
ANL P0,#0C7h
MOV @R1,P0
MOV P2,#0F6h
SETB P2.3
POP P0
NOP
LOOP63:
MOV R0,P1
CJNE R0,#0h,LOOP63
LJMP ET8

SRTMP:
MOV R4,#5h ;temporizare de
50ms
STEMP:LCALL SRTEMP
DEC R4
CJNE R4,#0h,STEMP
RET

TIMER0:
DJNZ R2,ETX
PUSH P0
MOV P0,R3
SETB P0.0
MOV P2,#0F4h
SETB P2.3
CLR TR0
POP P0
ETX: RETI

```

; Program calculator PC pentru  
 ERRORREX M

```

/* Subrutina Achbulet */
*****
#define WIN31
#include <stdio.h>
#include "contor.h"
#include "functii.h"
#include "achbulet.h"

extern int com1,com2,com3,com4,com;
extern int nr_posturi;

void TMenWindow::Ach_bul()
{
    GetApplication()->ExecDialog(new
    Probe(this, "PROBE"));
}

Probe::Probe(PTWindowsObject Owner, LPSTR
DName)
    :TDialog(Owner,DName)
{
    memset(&PR,0x0,sizeof PR);
    i1 = new TRadioButton(this,IDD_1,NULL);
    i5 = new TRadioButton(this,IDD_5,NULL);
    i10 = new TRadioButton(this,IDD_10,NULL);
    i20 = new TRadioButton(this,IDD_20,NULL);
    i50 = new TRadioButton(this,IDD_50,NULL);
    i100 = new
    TRadioButton(this,IDD_100,NULL);
    i200 = new
    TRadioButton(this,IDD_200,NULL);
    i300 = new
    TRadioButton(this,IDD_300,NULL);
    i400 = new
    TRadioButton(this,IDD_400,NULL);
    i600 = new
    TRadioButton(this,IDD_600,NULL);
    i1R = new
    TRadioButton(this,IDD_100R,NULL);
    i1S = new
    TRadioButton(this,IDD_100S,NULL);
    i1T = new
    TRadioButton(this,IDD_100T,NULL);
    f1 = new TRadioButton(this,IDD_F11,NULL);
    f5 = new TRadioButton(this,IDD_F15,NULL);
    f8 = new TRadioButton(this,IDD_F18,NULL);
    nfis = new TEdit(this,IDD_NFIS,7);
    PR.i100 = BF_CHECKED;
    PR.f1 = BF_CHECKED;
    TransferBuffer=(void far *)&PR;
}

void Probe::Ok(RTMessage)
{
    char numef[40],pmf[8];
    unsigned char post[1000];

    SetCursor(NULL);
    memset(PR.nfis,0x0,sizeof(PR.nfis));
    
```

```

SendDlgItemMsg(IDD_NFIS,WM_GETTEXT,7,
LPARAM(PR.nfis));
strcpy(pmf,PR.nfis);

if(SendDlgItemMessage(HWindow,IDD_1,BM_
GETCHECK,0,0)==0) PR.i1=0; else PR.i1=1;

if(SendDlgItemMessage(HWindow,IDD_5,BM_
GETCHECK,0,0)==0) PR.i5=0; else PR.i5=1;

if(SendDlgItemMessage(HWindow,IDD_10,BM_
GETCHECK,0,0)==0) PR.i10=0;else PR.i10=1;

if(SendDlgItemMessage(HWindow,IDD_20,BM_
GETCHECK,0,0)==0) PR.i20=0;else PR.i20=1;

if(SendDlgItemMessage(HWindow,IDD_50,BM_
GETCHECK,0,0)==0) PR.i50=0;else PR.i50=1;

if(SendDlgItemMessage(HWindow,IDD_100,B
M_GETCHECK,0,0)==0) PR.i100=0;else
PR.i100=1;

if(SendDlgItemMessage(HWindow,IDD_200,B
M_GETCHECK,0,0)==0) PR.i200=0;else
PR.i200=1;

if(SendDlgItemMessage(HWindow,IDD_300,B
M_GETCHECK,0,0)==0) PR.i300=0;else
PR.i300=1;

if(SendDlgItemMessage(HWindow,IDD_400,B
M_GETCHECK,0,0)==0) PR.i400=0;else
PR.i400=1;

if(SendDlgItemMessage(HWindow,IDD_600,B
M_GETCHECK,0,0)==0) PR.i600=0;else
PR.i600=1;

if(SendDlgItemMessage(HWindow,IDD_100R,B
M_GETCHECK,0,0)==0) PR.i1R=0;else
PR.i1R=1;

if(SendDlgItemMessage(HWindow,IDD_100S,B
M_GETCHECK,0,0)==0) PR.i1S=0;else
PR.i1S=1;

if(SendDlgItemMessage(HWindow,IDD_100T,B
M_GETCHECK,0,0)==0) PR.i1T=0;else
PR.i1T=1;

if(SendDlgItemMessage(HWindow,IDD_F11,BM_
GETCHECK,0,0)==0) PR.f1=0;else PR.f1=1;

if(SendDlgItemMessage(HWindow,IDD_F15,BM_
GETCHECK,0,0)==0) PR.f5=0;else PR.f5=1;

if(SendDlgItemMessage(HWindow,IDD_F18,BM_
GETCHECK,0,0)==0) PR.f8=0;else PR.f8=1;

if(PR.nfis[0]!='\0')
    {MessageBeep(0);
    
```



```

    MessageBox(NULL,"Introduceti
totusi un nume de fisier !","Nume fisier
lipsa",MB_OK|MB_ICONEXCLAMATION);
    return;
}
strcpy(numef,PR.nfis);
if(PR.i1R==1) strcat(numef,"r");
if(PR.i1S==1) strcat(numef,"s");
if(PR.i1T==1) strcat(numef,"t");
else if(PR.i1R==0 && PR.i1S==0 &&
PR.i1T==0) strcat(numef,"m");
if(PR.f1==1)
    strcat(numef,"1.");
if(PR.f5==1)
    strcat(numef,"5.");
if(PR.f8==1)
    strcat(numef,"8.");

if(PR.i1==1) strcat(numef,"001");
if(PR.i5==1) strcat(numef,"005");
if(PR.i10==1)strcat(numef,"010");
if(PR.i20==1)strcat(numef,"020");
if(PR.i50==1)strcat(numef,"050");
if(PR.i100==1)strcat(numef,"100");
if(PR.i200==1)strcat(numef,"200");
if(PR.i300==1)strcat(numef,"300");
if(PR.i400==1)strcat(numef,"400");
if(PR.i600==1)strcat(numef,"600");
if(PR.i1R==1)strcat(numef,"100");
if(PR.i1S==1)strcat(numef,"100");
if(PR.i1T==1)strcat(numef,"100");

TDialog::CloseWindow();

memset(post,'\0',sizeof post);
unsigned char ox[2];
unsigned int zz;
float fl;
int i;
FILE *fis;

if(com==com1)
{
    transmitere(1,3,0xff,0xff,0x22);
    receptie(1,0);
}
if(com==com2)
{
    transmitere(2,3,0xff,0xff,0x22);
    receptie(2,0);
}
if(com==com3)
{
    transmitere(3,3,0xff,0xff,0x22);
    receptie(3,0);
}
if(com==com4)
{
    transmitere(4,3,0xff,0xff,0x22);
    receptie(4,0);
}

fis=fopen(numef,"w+");
for(i=0;i<nr_posturi;i+=4)
{
    ox[0]=post[i+3];
    ox[1]=post[i+2];
    if((ox[0]&0x80)!=0x80)
    {
        zz=(int)ox[0]*256+(int)ox[1];
        fl=(float)zz/100;
    }
    else
    {
        ox[0]=ox[0]&0x7f;
        zz=(int)ox[0]*256+(int)ox[1];
        fl=-
        (float)zz/100;
    }
    fprintf(fis,"%d %.2f\n",post[i],fl);
} //for
if(PR.i1==1) fprintf(fis,"%s
","Ib=1%"); //la sf. fis
if(PR.i5==1) fprintf(fis,"%s
","Ib=5%");
if(PR.i10==1) fprintf(fis,"%s
","Ib=10%");
if(PR.i20==1) fprintf(fis,"%s
","Ib=20%");
if(PR.i50==1) fprintf(fis,"%s
","Ib=50%");
if(PR.i100==1)
    fprintf(fis,"%s ","Ib=100%");
if(PR.i200==1)
    fprintf(fis,"%s ","Ib=200%");
if(PR.i300==1)
    fprintf(fis,"%s ","Ib=300%");
if(PR.i400==1)
    fprintf(fis,"%s ","Ib=400%");
if(PR.i600==1)
    fprintf(fis,"%s ","Ib=600%");
if(PR.i1R==1) fprintf(fis,"%s
","Ib=100R%");
if(PR.i1S==1) fprintf(fis,"%s
","Ib=100S%");
if(PR.i1T==1) fprintf(fis,"%s
","Ib=100T%");
if(PR.f1==1) fprintf(fis,"%s
"," Cos(Fi)=1");
if(PR.f5==1) fprintf(fis,"%s
"," Cos(Fi)=0.5");
if(PR.f8==1) fprintf(fis,"%s
"," Cos(Fi)=0.8");

fclose(fis);
memset(&PR,0x0,sizeof PR );
strcpy(PR.nfis,prnf);
}

```

```

/* Subrutina Buletin* /
*****
#define WIN31
#include <stdio.h>
#include "contor.h"
#include "constant.h"
#include "buletin.h"
#include "functii.h"
#include "achbulet.h"

extern RECT r;
extern int total;
extern char path1[40];
HWND hw,hwn;
char
cap1[12][7]={"Nr.", "Tip", "Serie", "Un", "In", "Con
st.", "Sen-", "Mers", "Index", "Index", "Cls
", "Index"};
char
cap2[12][7]={"crt.", "Cnt.", "Cnt.", "[V]", "[A]", "co
ntor", "sib.", "gol", "start", "stop", "prec.", "iesire"};
int lung[12]={3,10,9,3,3,5,1,1,7,7,3,7};

void TMenWindow::BuletinL()
{
if(total!=0)
{
hw=GetActiveWindow();
GetApplication()->MakeWindow(new
TBul( this,"Verificari\\Buletin\\Caracteristici
contoare" ));
hwn=GetLastActivePopup(hw);
}
}

TBul::TBul(PtWindowsObject AParent, LPSTR
ATitle)

TWindow(AParent, ATitle)
{
Attr.Style=Attr.Style|WS_POPUP|WS_CAPTIO
N|WS_OVERLAPPEDWINDOW|WS_HSCROL
L|WS_VSCROLL;
Attr.X=7;
Attr.Y=105;
Attr.W=r.right-40;
Attr.H=300;
rinduri=0;k1=0;k2=0;k3=0;

Scroller= new TScroller(this,10,10,50,50);
EnableKBHandler();

int k=20;
for(int i=0;i<12;i++)
{
new TStatic(this,-1,cap1[i],k,18,85,25,0);
kk[i]=k;
y=lung[i]-strlen(cap1[i]);
if(y>0) y=15*y+2;
k=k+12*strlen(cap1[i])+y;

Edit[0]= new
TEdit(this,IDD_CRT,"",kk[0],100,13*lung[0],30,
lung[0]+1,FALSE);
Edit[1]= new
TEdit(this,IDD_TIPC,"",kk[1]+7,100,12*lung[1],
30,lung[1]+1,FALSE);
Edit[2]= new
TEdit(this,IDD_SERIE,"",kk[2],100,12*lung[2],3
0,lung[2]+1,FALSE);
Edit[3]= new
TEdit(this,IDD_UN,"",kk[3],100,13*lung[3],30,l
ung[3]+1,FALSE);
Edit[4]= new
TEdit(this,IDD_IN,"",kk[4],100,13*lung[4],30,l
ung[4]+1,FALSE);
Edit[5]= new
TEdit(this,IDD_CON,"",kk[5],100,12*lung[5],30
,lung[5]+1,FALSE);

Edit[6]= new
TEdit(this,IDD_SENS,"",kk[6],100,25*lung[6],3
0,lung[6]+1,FALSE);
Edit[7]= new
TEdit(this,IDD_MERS,"",kk[7],100,25*lung[7],3
0,lung[7]+1,FALSE);
Edit[8]= new
TEdit(this,IDD_ISTART,"",kk[8],100,12*lung[8]
,30,lung[8]+1,FALSE);
Edit[9]= new
TEdit(this,IDD_ISTOP,"",kk[9],100,12*lung[9],3
0,lung[9]+1,FALSE);
Edit[10]= new
TEdit(this,IDD_PREC,"",kk[10],100,13*lung[10]
,30,lung[10]+1,FALSE);
Edit[11]= new
TEdit(this,IDD_IIES,"",kk[11],100,12*lung[11]
,30,lung[11]+1,FALSE);

new
TButton(this,IDOK1,"OK",200,180,70,30,TRUE)
;
new
TButton(this,IDCANCEL1,"Cancel",400,180,70,
30,FALSE);

in=fopen("lumi.tmp","w+");

ShowWindow (hw,SW_SHOW);
UpdateWindow(hw);
}
}

```

```

void TBul::IOk(RTMessage)
{
    char buf[7];

    SendDlgItemMessage(hwn,IDD_CRT,WM_GETTEXT,4,LPARAM(CT.crt));

    SendDlgItemMessage(hwn,IDD_TIPC,WM_GETTEXT,11,LPARAM(CT.tipc));

    SendDlgItemMessage(hwn,IDD_SERIE,WM_GETTEXT,10,LPARAM(CT.serie));

    SendDlgItemMessage(hwn,IDD_UN,WM_GETTEXT,4,LPARAM(CT.un));

    SendDlgItemMessage(hwn,IDD_IN,WM_GETTEXT,4,LPARAM(CT.in));

    SendDlgItemMessage(hwn,IDD_CON,WM_GETTEXT,6,LPARAM(CT.con));

    SendDlgItemMessage(hwn,IDD_SENS,WM_GETTEXT,2,LPARAM(CT.sens));

    SendDlgItemMessage(hwn,IDD_MERS,WM_GETTEXT,2,LPARAM(CT.mers));

    SendDlgItemMessage(hwn,IDD_ISTART,WM_GETTEXT,8,LPARAM(CT.istart));

    SendDlgItemMessage(hwn,IDD_ISTOP,WM_GETTEXT,8,LPARAM(CT.istop));

    SendDlgItemMessage(hwn,IDD_PREC,WM_GETTEXT,4,LPARAM(CT.prec));

    SendDlgItemMessage(hwn,IDD_IIES,WM_GETTEXT,8,LPARAM(CT.iies));

    fprintf(in,"%s %s %s %s %s %s %s %s %s %s %s %s\n",CT.crt,CT.tipc,CT.serie,CT.un,CT.in,CT.con,CT.sens,CT.mers,CT.istart,CT.istop,CT.prec,CT.iies);
    rinduri++;

    crt=atoi(CT.crt);
    crt++;

    strcpy(buf,"Post");
    strcat(buf,inttostring(crt));

    GetPrivateProfileString("Constanta",buf,"",constanta,6,path1);

    SendDlgItemMessage(hwn,IDD_CRT,WM_SETTEXT,0,LPARAM(inttostring(crt)));

    SendDlgItemMessage(hwn,IDD_SERIE,WM_SETTEXT,0,LPARAM(""));

    SendDlgItemMessage(hwn,IDD_CON,WM_SETTEXT,0,LPARAM(constanta));

    SendDlgItemMessage(hwn,IDD_SENS,WM_SETTEXT,0,LPARAM(""));

    SendDlgItemMessage(hwn,IDD_MERS,WM_SETTEXT,0,LPARAM(""));

    SendDlgItemMessage(hwn,IDD_ISTART,WM_SETTEXT,0,LPARAM(""));

    SendDlgItemMessage(hwn,IDD_ISTOP,WM_SETTEXT,0,LPARAM(""));

    SendDlgItemMessage(hwn,IDD_IIES,WM_SETTEXT,0,LPARAM(""));

    SetActiveWindow(hwn);

    if(rinduri==total)
    { DestroyWindow(hwn);
      DestroyWindow(HWindow);
      fclose(in);
      if(GetApplication()->ExecDialog(obbul = new Buletin(this,"BULETIN"))!=IDOK)
      {
          cfisiere();
          citirefis();
          fclose(out);
          fclose(in);
      }
    }

    void TBul::Ecart(RTMessage Msg)
    {
        if(danu==0)
        {
            if(Msg.LP.Hi==EN_SETFOCUS)
            {
                Edit[0]->SetText("1");
                CT.crt[0]='1';
            }
            danu=1;

            GetPrivateProfileString("Constanta","Post1","",constanta,6,path1);
            Edit[5]->SetText(constanta);
        }
    }

    void TBul::ICancel(RTMessage)
    {
        DestroyWindow(hwn);
        DestroyWindow(HWindow);
        fclose(in);
    }

```

```

if(GetApplication()->ExecDialog(obbul = new
Bulletin(this,"BULETIN"))==IDOK)
{
    cfisiere();
    citirefis();
    fclose(in);
}

void TBul::citirefis()
{
    int len,i,j,t=0;
    char sir[12][11];
    char sire[8];
    FILE *fis[36];
    struct time tm;
    struct date d;

    if((obbul->BUL).nfis[0]!='\0')
    {
        MessageBeep(0);
        MessageBox(HWindow,"Introduceti un nume
de fisier", "Mesaj de
atentionare",MB_ICONEXCLAMATION|MB_O
K);
        return;
    }
    out=fopen(strcat((obbul-
>BUL).nfis,"bul"),"w+");
    fprintf(out,"%s","\n\n\n");
    getdate(&d);
    fprintf(out,"%2d.%02d.%4d\n",d.da_day,
d.da_mon,d.da_year);
    gettime(&tm);
    fprintf(out,"%2d:%02d\n",tm.ti_hour,
tm.ti_min);
    fprintf(out,"%s", "          B U
L E T I N   D E   V E R I F I C A R E \n\n");
    fprintf(out,"%s", "Nr. Tip Serie Un In
Const Sen- Mers Index Index Cls. Index");
    fprintf(out,"%s", " Cos(Fi)=1");
    for(i=0;i<k1+k2+4;i++) fprintf(out,"%s", " ");
    fprintf(out,"%s", "Cos(Fi)=0.5");
    for(i=0;i<k2+k3+4;i++) fprintf(out,"%s", " ");
    fprintf(out,"%s", "Cos(Fi)=0.8");
    fprintf(out,"%s","\n crt. Contor Cont. [V]
[A] cont. sib. gol start stop prec. iesire ");
    if(j1[0]==0&&kl!=0) fprintf(out,"%s ", "1%");
    for(i=0;i<36;i++) //pt cap de tabel
    switch(j1[i])
    {
        //aranjare in antet
        case 1: fprintf(out," %s ", "5%");
        break;
        case 2: fprintf(out,"%s ", "10%");
        break;
        case 3: fprintf(out,"%s ", "20%");
        break;
        case 4: fprintf(out,"%s ", "50%");
        break;
        case 5: fprintf(out,"%s ", "100%");
        break;
        case 6: fprintf(out,"%s ", "200%");
        break;
        case 7: fprintf(out,"%s ", "300%");
        break;
        case 8: fprintf(out,"%s ", "400%");
        break;
        case 9: fprintf(out,"%s ", "600%");
        break;
        case 10: fprintf(out,"%s ", "100R%");
        break;
        case 11: fprintf(out,"%s ", "100S%");
        break;
        case 12: fprintf(out,"%s ", "100T%");
        break;
        case 13: fprintf(out," %s ", "1%");
        break;
        case 14: fprintf(out," %s ", "5%");
        break;
        case 15: fprintf(out,"%s ", "10%");
        break;
        case 16: fprintf(out,"%s ", "20%");
        break;
        case 17: fprintf(out,"%s ", "50%");
        break;
        case 18: fprintf(out,"%s ", "100%");
        break;
        case 19: fprintf(out,"%s ", "200%");
        break;
        case 20: fprintf(out,"%s ", "300%");
        break;
        case 21: fprintf(out,"%s ", "400%");
        break;
        case 22: fprintf(out,"%s ", "600%");
        break;
        case 23: fprintf(out,"%s ", "100R%");
        break;
        case 24: fprintf(out,"%s ", "100S%");
        break;
        case 25: fprintf(out,"%s ", "100T%");
        break;
        case 26: fprintf(out," %s ", "1%");
        break;
        case 27: fprintf(out," %s ", "5%");
        break;
        case 28: fprintf(out,"%s ", "10%");
        break;
        case 29: fprintf(out,"%s ", "20%");
        break;
        case 30: fprintf(out,"%s ", "50%");
        break;
        case 31: fprintf(out,"%s ", "100%");
        break;
        case 32: fprintf(out,"%s ", "200%");
        break;
        case 33: fprintf(out,"%s ", "300%");
        break;
        case 34: fprintf(out,"%s ", "400%");
        break;
        case 35: fprintf(out,"%s ", "600%");
        break;
    }
}

```

```

fprintf(out,"%s","\n
_____
_____
_____
\n");

char ch;
in=fopen("lumi.tmp","r+");
long cpos = ftell(in);
fseek(in, 0L, SEEK_END);
long length = ftell(in);
fseek(in, cpos, SEEK_SET);
if(length==0) {fclose(in); fclose(out); return;}

for(j=0;j<j2;j++) //j2 =nr de fisiere care exista
fis[j]=fopen(numef[j1[j]],"r+"); //j1[j]=indicii
fisierelor care exista

for(i=0;i<rinduri;i++)
{j=0;
do
{
fscanf(in,"%s",sir[j]);
int k=0;
do{
fscanf(in,"%c",&ch);
if(ch==' ') k++;
}while(ch==' ');
fseek(in,-1,SEEK_CUR);
j++;t=j;
while(k>1&&j<t+k-1)
{ sir[j][0]='\0';
j++;
}

}while(j<12);

len=lung[0]-strlen(sir[0]);
fprintf(out," %s",sir[0]);
for(j=0;j<len+2;j++) fprintf(out,"%s"," ");
len=lung[1]-strlen(sir[1]);
fprintf(out,"%s",sir[1]);
for(j=0;j<len+1;j++) fprintf(out,"%s"," ");
len=lung[2]-strlen(sir[2]);
fprintf(out,"%s",sir[2]);
for(j=0;j<len+2;j++) fprintf(out,"%s"," ");
len=lung[3]-strlen(sir[3]);
fprintf(out,"%s",sir[3]);
for(j=0;j<len+2;j++) fprintf(out,"%s"," ");
len=lung[4]-strlen(sir[4]);
fprintf(out,"%s",sir[4]);
for(j=0;j<len+2;j++) fprintf(out,"%s"," ");
len=lung[5]-strlen(sir[5]);
fprintf(out,"%s",sir[5]);
for(j=0;j<len+3;j++) fprintf(out,"%s"," ");
len=lung[6]-strlen(sir[6]);
fprintf(out,"%s",sir[6]);
for(j=0;j<len+2;j++) fprintf(out,"%s"," ");
len=lung[7]-strlen(sir[7]);
fprintf(out,"%s",sir[7]);
for(j=0;j<len+4;j++) fprintf(out,"%s"," ");
len=lung[8]-strlen(sir[8]);

fprintf(out,"%s",sir[8]);
for(j=0;j<len+2;j++) fprintf(out,"%s"," ");
len=lung[9]-strlen(sir[9]);
fprintf(out,"%s",sir[9]);
for(j=0;j<len+2;j++) fprintf(out,"%s"," ");
len=lung[10]-strlen(sir[10]);
fprintf(out,"%s",sir[10]);
for(j=0;j<len+3;j++) fprintf(out,"%s"," ");
len=lung[11]-strlen(sir[11]);
fprintf(out,"%s",sir[11]);
for(j=0;j<len+1;j++) fprintf(out,"%s"," ");

for(j=0;j<j2;j++)
{
fscanf(fis[j],"%s",sire);
fscanf(fis[j],"%s",sire);
fprintf(out,"%s",sire);
len=6-strlen(sire);
for(int k=0;k<len;k++) fprintf(out,"%s"," ");
}
fprintf(out,"%c","\n");
}
fclose(out);
memset(&(obbul->BUL),0x0,sizeof obbul->BUL);
}

void TBul::cfisiere()
{
for(int i=0;i<36;i++)
strcpy(numef[i],(obbul->BUL).nfis);
if((obbul->BUL).i1a==1)
{strcat(numef[0],"m1.001");k1++;} else
numef[0][0]='\0';
if((obbul->BUL).i5a==1)
{strcat(numef[1],"m1.005");k1++;} else
numef[1][0]='\0';
if((obbul->BUL).i10a==1){strcat(numef[2],"m1.010");k1+
+;} else numef[2][0]='\0';
if((obbul->BUL).i20a==1){strcat(numef[3],"m1.020");k1+
+;} else numef[3][0]='\0';
if((obbul->BUL).i50a==1){strcat(numef[4],"m1.050");k1+
+;} else numef[4][0]='\0';
if((obbul->BUL).i100a==1)
{strcat(numef[5],"m1.100");k1++;} else
numef[5][0]='\0';
if((obbul->BUL).i200a==1)
{strcat(numef[6],"m1.200");k1++;} else
numef[6][0]='\0';
if((obbul->BUL).i300a==1)
{strcat(numef[7],"m1.300");k1++;} else
numef[7][0]='\0';
if((obbul->BUL).i400a==1)
{strcat(numef[8],"m1.400");k1++;} else
numef[8][0]='\0';
if((obbul->BUL).i600a==1)
{strcat(numef[9],"m1.600");k1++;} else
numef[9][0]='\0';
}

```

```

if((obbul->BUL).i1Ra==1)
{strcat(numef[10],"r1.100");k1++;} else
numef[10][0]='\0';
if((obbul->BUL).i1Sa==1)
{strcat(numef[11],"s1.100");k1++;} else
numef[11][0]='\0';
if((obbul->BUL).i1Ta==1)
{strcat(numef[12],"t1.100");k1++;} else
numef[12][0]='\0';
if((obbul->BUL).i1b==1)
{strcat(numef[13],"m1.001");k2++;} else
numef[13][0]='\0';
if((obbul->BUL).i5b==1)
{strcat(numef[14],"m5.005");k2++;} else
numef[14][0]='\0';
if((obbul->BUL).i10b==1)
{strcat(numef[15],"m5.010");k2++;} else
numef[15][0]='\0';
if((obbul->BUL).i20b==1)
{strcat(numef[16],"m5.020");k2++;} else
numef[16][0]='\0';
if((obbul->BUL).i50b==1)
{strcat(numef[17],"m5.050");k2++;} else
numef[17][0]='\0';
if((obbul->BUL).i100b==1)
{strcat(numef[18],"m5.100");k2++;} else
numef[18][0]='\0';
if((obbul->BUL).i200b==1)
{strcat(numef[19],"m5.200");k2++;} else
numef[19][0]='\0';
if((obbul->BUL).i300b==1)
{strcat(numef[20],"m5.300");k2++;} else
numef[20][0]='\0';
if((obbul->BUL).i400b==1)
{strcat(numef[21],"m5.400");k2++;} else
numef[21][0]='\0';
if((obbul->BUL).i600b==1)
{strcat(numef[22],"m5.600");k2++;} else
numef[22][0]='\0';
if((obbul->BUL).i1Rb==1)
{strcat(numef[23],"r5.100");k2++;} else
numef[23][0]='\0';
if((obbul->BUL).i1Sb==1)
{strcat(numef[24],"s5.100");k2++;} else
numef[24][0]='\0';
if((obbul->BUL).i1Tb==1)
{strcat(numef[25],"t5.100");k2++;} else
numef[25][0]='\0';
if((obbul->BUL).i1c==1)
{strcat(numef[26],"m8.001");k3++;} else
numef[26][0]='\0';
if((obbul->BUL).i5c==1)
{strcat(numef[27],"m8.005");k3++;} else
numef[27][0]='\0';
if((obbul->BUL).i10c==1)
{strcat(numef[28],"m8.010");k3++;} else
numef[28][0]='\0';
if((obbul->BUL).i20c==1)
{strcat(numef[29],"m8.020");k3++;} else
numef[29][0]='\0';
if((obbul->BUL).i50c==1)
{strcat(numef[30],"m8.050");k3++;} else
numef[30][0]='\0';
if((obbul->BUL).i100c==1)
{strcat(numef[31],"m8.100");k3++;} else
numef[31][0]='\0';
if((obbul->BUL).i200c==1)
{strcat(numef[32],"m8.200");k3++;} else
numef[32][0]='\0';
if((obbul->BUL).i300c==1)
{strcat(numef[33],"m8.300");k3++;} else
numef[33][0]='\0';
if((obbul->BUL).i400c==1)
{strcat(numef[34],"m8.400");k3++;} else
numef[34][0]='\0';
if((obbul->BUL).i600c==1)
{strcat(numef[35],"m8.600");k3++;} else
numef[35][0]='\0';

OFSTRUCT ofs;
int ng;

memset(j1,0,sizeof(j1));
for(i=0,j2=0;i<36;i++)
{
ng=OpenFile(numef[i],&ofs,OF_EXIST);
if(ng!=-1)
{
j1[j2]=i;
j2++;
}
else
if(numef[i][0]!='\0')
{
MessageBox(HWindow,"Fisierul nu
exista",numef[i],MB_OK);
GetApplication()->ExecDialog(new
Probe(this,"PROBE"));
i--;
}
}

Buletin::Buletin(PTWindowsObject Owner,
LPSTR DName)
:DeBaza(Owner,DName)
{
memset(&BUL,0x0, sizeof BUL);
i1a = new TBCheckBox(this,IDD_1a,NULL);
i5a = new TBCheckBox(this,IDD_5a,NULL);
i10a = new TBCheckBox(this,IDD_10a,NULL);
i20a = new TBCheckBox(this,IDD_20a,NULL);
i50a = new TBCheckBox(this,IDD_50a,NULL);
i100a = new
TBCheckBox(this,IDD_100a,NULL);
i200a = new
TBCheckBox(this,IDD_200a,NULL);
i300a = new
TBCheckBox(this,IDD_300a,NULL);
i400a = new
TBCheckBox(this,IDD_400a,NULL);
i600a = new
TBCheckBox(this,IDD_600a,NULL);

```

```

i1Ra = new extern int com;
TBCheckBox(this,IDD_100Ra,NULL); extern unsigned char post[1000];
i1Sa = new extern char name[40];
TBCheckBox(this,IDD_100Sa,NULL);
i1Ta = new
TBCheckBox(this,IDD_100Ta,NULL);

Com::Com( PTWindowsObject Owner , LPSTR
DName )
:TDialog(Owner,DName)
{
i1b = new TBCheckBox(this,IDD_1b,NULL);
i5b = new TBCheckBox(this,IDD_5b,NULL);
i10b = new
TBCheckBox(this,IDD_10b,NULL);
i20b = new
TBCheckBox(this,IDD_20b,NULL);
i50b = new
TBCheckBox(this,IDD_50b,NULL);
i100b = new
TBCheckBox(this,IDD_100b,NULL);
i200b = new
TBCheckBox(this,IDD_200b,NULL);
i300b = new
TBCheckBox(this,IDD_300b,NULL);
i400b = new
TBCheckBox(this,IDD_400b,NULL);
i600b = new
TBCheckBox(this,IDD_600b,NULL);
i1Rb = new
TBCheckBox(this,IDD_100Rb,NULL);
i1Sb = new
TBCheckBox(this,IDD_100Sb,NULL);
i1Tb = new
TBCheckBox(this,IDD_100Tb,NULL);

ComUL.c5 = BF_CHECKED;

TransferBuffer=(void far *)&COMUL;
}

void Com :: Ok(RTMessage )
{
if(SendDlgItemMessage(HWindow,IDD_CO1,B
M_GETCHECK,0,0)==0) COMUL.c1=0; else
COMUL.c1=1;

if(SendDlgItemMessage(HWindow,IDD_CO2,B
M_GETCHECK,0,0)==0) COMUL.c2=0; else
COMUL.c2=1;

if(SendDlgItemMessage(HWindow,IDD_CO3,B
M_GETCHECK,0,0)==0) COMUL.c3=0; else
COMUL.c3=1;

if(SendDlgItemMessage(HWindow,IDD_CO4,B
M_GETCHECK,0,0)==0) COMUL.c4=0; else
COMUL.c4=1;

if(SendDlgItemMessage(HWindow,IDD_CAUT,
BM_GETCHECK,0,0)==0) COMUL.c5=0; else
COMUL.c5=1;

if(COMUL.c1==1 )

if(OpenComm("com1",1024,128)!=IE_HARDW
ARE &&
OpenComm("com1",1024,128)!=IE_BADID)
{ com=1;

WritePrivateProfileString("Configurare
","Com","1",name);
setare_com();

TDialog::CloseWindow();

transmitere(com,3,0xfe,0x1b,0x07);
receptie(com,3);
}

i1c = new TBCheckBox(this,IDD_1c,NULL);
i5c = new TBCheckBox(this,IDD_5c,NULL);
i10c = new TBCheckBox(this,IDD_10c,NULL);
i20c = new TBCheckBox(this,IDD_20c,NULL);
i50c = new TBCheckBox(this,IDD_50c,NULL);
i100c = new
TBCheckBox(this,IDD_100c,NULL);
i200c = new
TBCheckBox(this,IDD_200c,NULL);
i300c = new
TBCheckBox(this,IDD_300c,NULL);
i400c = new
TBCheckBox(this,IDD_400c,NULL);
i600c = new
TBCheckBox(this,IDD_600c,NULL);
Const[0]= new TEdit(this,IDD_ERR,4);
Const[1]= new TEdit(this,IDD_BULETIN,6);
Const[2]= new TEdit(this,IDD_FISI,14);
Const[3]= new TEdit(this,IDD_METR,22);

TransferBuffer=(void far *)&BUL;
}

/* Subrutina Com */
*****
#define WIN31
#include "contor.h"
#include "com.h"
#include "functii.h"

```

```

        if(post[0]!=0xfe ||
post[1]!=0x1b || post[2]!=0x07)
        if(OpenComm("com4",1024,128)!=IE_
        HARDWARE &&
        OpenComm("com4",1024,128)!=IE_BADID)
        { com=4;
                WritePrivateProfileString("Configurare
        ", "Com", "4", name);
                setare_com();
                TDialog::CloseWindow();
                transmitere(com,3,0xfe,0x1b,0x07);
                receptie(com,3);
                if(post[0]!=0xfe ||
post[1]!=0x1b || post[2]!=0x07)
                MessageBox(HWindow,"Interfata nu este
        conectata","Mesaj de
        eroare",MB_OK|MB_ICONHAND);
        }
        else {
                MessageBox(HWindow,"Com-
        ul este ocupat sau nu exista","Eroare setare
        com",MB_OK|MB_ICONEXCLAMATION);
                return;}
        if(COMUL.c2==1)
        if(
        OpenComm("com2",1024,128)!=IE_HARDWA
        RE &&
        OpenComm("com2",1024,128)!=IE_BADID)
        { com=2;
                WritePrivateProfileString("Configurare
        ", "Com", "2", name);
                setare_com();
                TDialog::CloseWindow();
                transmitere(com,3,0xfe,0x1b,0x07);
                receptie(com,3);
                if(post[0]!=0xfe ||
post[1]!=0x1b || post[2]!=0x07)
                MessageBox(HWindow,"Interfata nu este
        conectata","Mesaj de
        eroare",MB_OK|MB_ICONHAND);
        }
        else {
                MessageBox(HWindow,"Com-
        ul este ocupat sau nu exista","Eroare setare
        com",MB_OK|MB_ICONEXCLAMATION);
                return;}
        if(COMUL.c3==1)
        if(
        OpenComm("com3",1024,128)!=IE_HARDWA
        RE &&
        OpenComm("com3",1024,128)!=IE_BADID)
        { com=3;
                WritePrivateProfileString("Configurare
        ", "Com", "3", name);
                setare_com();
                TDialog::CloseWindow();
                transmitere(com,3,0xfe,0x1b,0x07);
                receptie(com,3);
                if(post[0]!=0xfe ||
post[1]!=0x1b || post[2]!=0x07)
                MessageBox(HWindow,"Interfata nu este
        conectata","Mesaj de
        eroare",MB_OK|MB_ICONHAND);
        }
        else {
                MessageBox(HWindow,"Com-
        ul este ocupat sau nu exista","Eroare setare
        com",MB_OK|MB_ICONEXCLAMATION);
                return;}
        if(COMUL.c4==1)
        if(OpenComm("com4",1024,128)!=IE_
        HARDWARE &&
        OpenComm("com4",1024,128)!=IE_BADID)
        { com=4;
                WritePrivateProfileString("Configurare
        ", "Com", "4", name);
                setare_com();
                TDialog::CloseWindow();
                transmitere(com,3,0xfe,0x1b,0x07);
                receptie(com,3);
                if(post[0]!=0xfe ||
post[1]!=0x1b || post[2]!=0x07)
                MessageBox(HWindow,"Interfata nu este
        conectata","Mesaj de
        eroare",MB_OK|MB_ICONHAND);
        }
        else {
                MessageBox(HWindow,"Com-
        ul este ocupat sau nu exista","Eroare setare
        com",MB_OK|MB_ICONEXCLAMATION);
                return;}
        if(COMUL.c5==1)
        for(int i=1;i<5;i++) //cautare
        automata de com
        {
                char
        comul[5]="\0"; wsprintf(comul,"com%d",i);
        if(OpenComm(comul,1024,128)!=IE_HARDWA
        RE) //noul com e liber
        { com=i;
                setare_com();
                transmitere(com,3,0xfe,0x1b,0x07);
                receptie(com,3);
                if(post[0]==0xfe && post[1]==0x1b &&
post[2]==0x07)
                { com=i;
                        WritePrivateProfileString("Configurare", "Com", i
        ntostring(i), name);
                        break;}
                else{
                        TDialog::CloseWindow();
                                MessageBox(HWindow,"Interfata nu
        este conectata","Mesaj de eroare",MB_OK);
                                WritePrivateProfileString("Configurare
        ", "Com", intostring(i), name);
                                break;
                }
        }
    
```



```

    }
    } //if
    } //for
    TDialog::CloseWindow();
}

/* Subrutina Comunic */
*****
#define WIN31

#include <stdio.h>
#include <filedial.h>

#include "contor.h"
#include "comunic.h" //clasa afisare
#include "functii.h"
#include "com.h"

extern int com1,com2,com3,com4,com;
extern char name[40],path[40];
extern int no;
extern RECT r;
extern HANDLE hInst;
extern char Device[40],Driver[40],Port[15];
char FileName[50];
extern unsigned char post[1000];
int este=0,total;

void TMenWindow::Invatare()
{int nr_def;
char comi[5]="\0";

SetCursor(NULL);

strcpy(comi,"com");strcat(comi,inttostring(com));
int co=OpenComm(comi,1024,128);
if(co==IE_HARDWARE || co==IE_BADID)
//nu exista com-ul din fisier sau e ocupat
{
    MessageBox(HWindow,"Eroare
    configurare COM
    !","Mesaj",MB_OK|MB_ICONEXCLAMATION
    );
    GetApplication()->ExecDialog(new
    Com(this,"SCH_COM"));
}
else //e bun com-ul
{ setare_com();
  transmitere(com,3,0xfe,0x1b,0x07);
//testare interfata
  receptie(com,3);

  if(post[0]!=0xfe || post[1]!=0x1b ||
  post[2]!=0x07)
    }

    MessageBox(HWindow,"Interfata
    neconectata !","Mesaj de eroare",MB_OK);
    }

    if(com==com1)
    {
        transmitere(1,4,0xff,0xff,0x11,0x01);
        receptie(1,0);
    }
    if(com==com2) //com2
    {
        transmitere(2,4,0xff,0xff,0x11,0x01);
        receptie(2,0);
    }
    if(com==com3)
    {
        transmitere(3,4,0xff,0xff,0x11,0x01);
        receptie(3,0);
    }
    if(com==com4)
    {
        transmitere(4,4,0xff,0xff,0x11,0x01);
        receptie(4,0);
    }

    for(int i=0;i<no;i+=2) //no=nr de
    octeti primiti
        if(post[i+2]-post[i]>1)
            {
                nr_def=(i+2)/2;
                conversie(nr_def,"Eroare la
                postul ", "");
            }

    /* FILE *stream;
    stream=fopen("raoul.gab","w+");
    for(int k=0;k<no;k++)

    fprintf(stream,"%x\n",post[k]);
    fprintf(stream,"nr de
    posturi:%d\n",no/2);
    fclose(stream);*/
    no/=2;
    total=no;
    conversie(no,"Instalatia are "," posturi
    ");
}

void TMenWindow::About()
{
    GetApplication()->ExecDialog(new
    TDialog(this,"About"));
}

void TMenWindow::Iesire()
{
    if(MessageBox(HWindow,"Doriti iesirea
    ?","Abandonare

```

```

program",MB_YESNO|MB_ICONQUESTION)=
=IDYES)
    { WinHelp( HWindow,
"CONTOR.HLP", HELP_QUIT,0L );
    DestroyWindow(HWindow);
    }

void TMenWindow::Com1() //setez comul
{
    int co=OpenComm("COM1",1024,128);
    if(co==IE_HARDWARE || co==IE_BADID)
//nu exista com-ul din fisier sau e ocupat
        MessageBox(HWindow,"Comul este
ocupat sau nu exista !","Mesaj de
eroare",MB_OK|MB_ICONEXCLAMATION);
    else
    {
        com=com1;
        outportb(0x3fb,0x80); //pun DLAB pe
1
        outportb(0x3f8,0xc); //rata de transfer
outportb(0x3f9,0x0);
        outportb(0x3fb,0x3); //paritate,stop
outportb(0x3f9,0x0); //validez
intreruperea

        WritePrivateProfileString("Configurare","Com","
1",name);
    }
}

void TMenWindow::Com2()
{
    int co=OpenComm("COM2",1024,128);
    if(co==IE_HARDWARE || co==IE_BADID)
//nu exista com-ul din fisier sau e ocupat
        MessageBox(HWindow,"Comul este
ocupat sau nu exista !","Mesaj de
eroare",MB_OK|MB_ICONEXCLAMATION);
    else
    {
        com=com2;
        outportb(0x2fb,0x80); //pun DLAB pe
1
        outportb(0x2f8,0xc); //rata de transfer
outportb(0x2f9,0x0);
        outportb(0x2fb,0x3); //paritate,stop
outportb(0x2f9,0x0); //validez
intreruperea

        WritePrivateProfileString("Configurare","Com","
2",name);
    }
}

void TMenWindow::Com3() //setez comul
{
    int co=OpenComm("COM3",1024,128);
    if(co==IE_HARDWARE || co==IE_BADID)
//nu exista com-ul din fisier sau e ocupat
        MessageBox(HWindow,"Comul este
ocupat sau nu exista !","Mesaj de
eroare",MB_OK|MB_ICONEXCLAMATION);
    else
    {
        com=com3;
        outportb(0x3eb,0x80); //pun DLAB pe
1
        outportb(0x3e8,0xc); //rata de transfer
outportb(0x3e9,0x0);
        outportb(0x3eb,0x3); //paritate,stop
outportb(0x3e9,0x0); //validez
intreruperea

        WritePrivateProfileString("Configurare","Com","
3",name);
    }
}

void TMenWindow::Com4()
{
    int co=OpenComm("COM4",1024,128);
    if(co==IE_HARDWARE || co==IE_BADID)
//nu exista com-ul din fisier sau e ocupat
        MessageBox(HWindow,"Comul este
ocupat sau nu exista !","Mesaj de
eroare",MB_OK|MB_ICONEXCLAMATION);
    else
    {
        com=com4;
        outportb(0x2eb,0x80); //pun DLAB pe
1
        outportb(0x2e8,0xc); //rata de transfer
outportb(0x2e9,0x0);
        outportb(0x2eb,0x3); //paritate,stop
outportb(0x2e9,0x0); //validez
intreruperea

        WritePrivateProfileString("Configurare","Com","
4",name);
    }
}

void TMenWindow::Autotest()
{ // port 0: 3f8-com1, port 1: 2f8-com2
    if(com==com1)
        transmitere(1);
    if(com==com2)
        transmitere(2);
    if(com==com3)
        transmitere(3);
    if(com==com4)
        transmitere(4);
}

void TMenWindow::Achizitie()
{unsigned char ox[2];
    unsigned int zz;
    float fl;
    int i,k;
}

```

```

FILE *fis;
SetCursor(NULL);
memset(post,'\0',sizeof post);
if(com==com1)
{
    transmitere(1,3,0xff,0xff,0x22);
    receptie(1,0);
}
if(com==com2)
{
    transmitere(2,3,0xff,0xff,0x22);
    receptie(2,0);
}
if(com==com3)
{
    transmitere(3,3,0xff,0xff,0x22);
    receptie(3,0);
}
if(com==com4)
{
    transmitere(4,3,0xff,0xff,0x22);
    receptie(4,0);
}

fis=fopen("achiz.tst","w+");
for(i=0;i<4*total;i+=4)
{
    ox[0]=post[i+3];
    ox[1]=post[i+2];
    if((ox[0]&0x80)!=0x80)
    {
        zz=(int)ox[0]*256+(int)ox[1];
        fl=(float)zz/100;
    }
    else
    {
        ox[0]=ox[0]&0x7f;

        zz=(int)ox[0]*256+(int)ox[1];
        fl=-(float)zz/100;
    }
    if(fl==163.86)
        fprintf(fis,"%d
%s\n",(int)post[i],"Lipsa marca");
    else
        fprintf(fis,"%d
%.2fn",(int)post[i],fl);
}
fclose(fis);

strcpy(FileName1,path);
strcat(FileName1,"\\achiz.tst");
if(total!=0)
    GetApplication()->MakeWindow(new
afisare(this,FileName1));
}

char ChildName[50];
PTWindowsObject p;

if(este==0)
{
    wsprintf(ChildName,"Window %d ",
win_no++);
if(GetApplication()->ExecDialog(new
TFileDialog(this, SD_FILEOPEN,
strcpy(FileName1, "*. *"))) == IDOK)
{
    strcat(ChildName,FileName1);
    GetApplication()->MakeWindow(new
afisare(this,ChildName));
//GetApplication()-
>MakeWindow(new
afis_edit(this,ChildName,FileName1));
}
}

afis_edit::afis_edit(PTWindowsObject AParent,
LPSTR ATitle,LPSTR AFileName)
:TFileDialog(AParent,ATitle,AFileName)
{
    Attr.Style=Attr.Style|WS_POPUP|WS_OVERLA
PPEDWINDOW;
    Attr.X=10;
    Attr.Y=105;
    Attr.W=r.right-40;
    Attr.H=300;
}

void afis_edit::SetupWindow(){
    TFileDialog::SetupWindow();
}

/*void TMenWindow::FisOpen(RTMessage)
{
    char ChildName[50];
    PTWindowsObject p;

    if(este==0)
    {
        wsprintf(ChildName,"Window %d ",
win_no++);
        afis_edit *obct;
        obct= new afis_edit(this,ChildName,NULL);

        // obct->SetupWindow();
        obct->Open();
    }
}

void TMenWindow::FisOpen(RTMessage)

```

```

if(GetApplication()->ExecDialog(new
TFileDialog(this, SD_FILEOPEN,
strcpy(fileName1, "*. *")) == IDOK)
{
    strcat(ChildName, fileName1);
    //p=GetApplication()->MakeWindow(obct);
}
}
} */

void TMenWindow::PrintIt()
{
    if(este==0)
        MessageBox(HWindow, "Nu exista
        fisier deschis", "Mesaj de
        atentionare", MB_ICONEXCLAMATION |
        MB_OK);
    else
        GetApplication()->ExecDialog(new
        Tiparire_imp(this, "Tiparire"));
}

void TMenWindow::PrnSetup(RTMessage)
{
    HINSTANCE DevDriver,
    char TmpStr[20];
    void(far pascal
    *DevMode)(HWND, HANDLE, LPSTR, LPSTR);
    lstrcpy(TmpStr, Driver);
    lstrcat(TmpStr, ".DRV");
    DevDriver=LoadLibrary(TmpStr);
    if((int)DevDriver>=32)
    {
        (FARPROC)DevMode=GetProcAddress(DevDri
        ver, "DEVICEMODE");

        (*DevMode)(HWindow, DevDriver, Device, Port);
        FreeLibrary(DevDriver);
    }
}

Tiparire_imp::Tiparire_imp(PTWindowsObject
Owner, LPSTR DName)
: TDialog(Owner, DName)
{
    rind = new TEdit(this, IDD_RND, 3);
    p1 = new TEdit(this, IDD_PAG1, 3);
    p2 = new TEdit(this, IDD_PAG2, 4);
    toate = new TBRadioButton( this,
    IDD_TOATE, NULL );
    delapo = new TBRadioButton( this,
    IDD_DELAPO, NULL );
}

void Tiparire_imp::SetupWindow()
{
    TDialog::SetupWindow();
}

```

```

care=0;
SetDlgItemInt(HWindow, IDD_RND, 60, FALSE);
SetDlgItemInt(HWindow, IDD_PAG1, 1, FALSE);
SetDlgItemInt(HWindow, IDD_PAG2, total, FALS
E);
SendDlgItemMsg(IDD_TOATE, BM_SETCHEC
K, 1, 0); //marcat

SendDlgItemMsg(IDD_DELAPO, BM_SETCHE
CK, 0, 0);

EnableWindow(GetDlgItem(HWindow, IDD_PA
G1), FALSE);

EnableWindow(GetDlgItem(HWindow, IDD_PA
G2), FALSE);
}

void Tiparire_imp::Ok(RTMessage)
{
    SendDlgItemMsg(IDD_RND, WM_GETTEXT, 3,
    LPARAM(strind));

    SendDlgItemMsg(IDD_PAG1, WM_GETTEXT, 3
    ,LPARAM(strp1));

    SendDlgItemMsg(IDD_PAG2, WM_GETTEXT, 4
    ,LPARAM(strp2));

    if(SendDlgItemMessage(HWindow, IDD_TOAT
    E, BM_GETCHECK, 0, 0)==1)
        care=1;
    else

    if(SendDlgItemMessage(HWindow, IDD_DELAP
    O, BM_GETCHECK, 0, 0)==1)
        care=2;

    nrrind = atoi(strind);
    nrpag1 = atoi(strp1);
    nrpag2 = atoi(strp2);
    if(nrrind>60)
    {
        MessageBeep(MB_ICONHAND);

        MessageBox(NULL, "Valoarea trebuie sa fie mai
        mica de 60", "Mesaj de
        atentionare", MB_ICONHAND);
        return;
    }
    TDialog::CloseWindow();
    Tiparire();
}

```

```

BOOL FAR PASCAL AbortProc(HDC,short)
{
    MSG msg;

    while(PeekMessage(&msg,NULL,0,0,PM_REMOVE))
        DispatchMessage(&msg);
    return TRUE;
}

void Tiparire_imp::Tiparire()
{
    int Height,Width;
    int LineCount,i,j,k=0,flag=0;
    char TextLine[200],ch;
    int nrr;

    FARPROC AbortInst;
    PrintedHeight=0;
    PrmDC=CreateDC(Driver,Device,Port,NULL);
    Height=GetDeviceCaps(PrmDC,VERTRES);
    Width=GetDeviceCaps(PrmDC,HORZRES);
    EnableWindow(HWindow,FALSE);

    AbortInst=MakeProcInstance((FARPROC)AbortProc,hInst);

    Escape(PrmDC,SETABORTPROC,0,(LPSTR)AbortInst,NULL);

    Escape(PrmDC,STARTDOC,12,"PrintEditor",NULL);

    fil=fopen(FileName1,"r");

    HFONT
    hf1=CreateFont(23,14,0,0,FW_BOLD,FALSE,FALSE,FALSE,0,0,0,(BYTE)VARIABLE_PITCH|0x04|FF_MODERN,"modern");
    SelectObject(PrmDC,hf1);

    for(i=0;!feof(fil);i++)
        {
            for(j=0;j<200;j++)
                {
                    fread(&ch,sizeof(char),1,fil);
                    TextLine[j]=ch;

                    if(ch=='\n'){TextLine[j+1]='\0'; break;}
                }
            LineLen= strlen(TextLine);

            Extent=GetTextExtent(PrmDC,TextLine,LineLen); //calc dim textului cu fontul folosit
            while(Ex.LineWidth>Width)
                {
                    TextLine[--LineLen]='\0';
                }
            Extent=GetTextExtent(PrmDC,TextLine,LineLen);
        }
}

if(PrintedHeight+Ex.LineHeight>Height)
{
    Escape(PrmDC,NEWFRAME,0,NULL,NULL);
    PrintedHeight=0;
}

if((i%nrrind==0 && care==1)|| i%60==0)
    if(flag==1)
        {
            i=0;flag=0;

            Escape(PrmDC,NEWFRAME,0,NULL,NULL);

            cap_tabel();
        }
    if(care==2)
        if(i<10+nrrpag1 || i>10+nrrpag2 )
            continue;
        else
            {
                k++;

                if(k>=60)
                    {
                        k=0;

                        Escape(PrmDC,NEWFRAME,0,NULL,NULL);
                        cap_tabel();
                    }
            }
        if(LineLen)
            TextOut(PrmDC,0,PrintedHeight,TextLine,LineLen-1);
            flag=1;
            PrintedHeight+=Ex.LineHeight;
        } //for

    Escape(PrmDC,NEWFRAME,0,NULL,NULL);
    Escape(PrmDC,ENDDOC,0,NULL,NULL);

    fclose(fil);
    FreeProcInstance(AbortInst);
    DeleteObject(hf1);
    EnableWindow(HWindow,TRUE);
    DeleteDC(PrmDC);
}

void Tiparire_imp::aleg_dclapo()
{
}

```

```

}
SendDlgItemMsg(IDD_RND,WM_SETTEXT,0,
LPARAM("60"));

EnableWindow(GetDlgItem(HWindow,IDD_PA
G1),TRUE);

EnableWindow(GetDlgItem(HWindow,IDD_PA
G2),TRUE);
}

void Tiparire_imp :: aleg_toate()
{
SetDlgItemInt(HWindow,IDD_PAG1,1,FALSE);

SetDlgItemInt(HWindow,IDD_PAG2,total,FALS
E);

EnableWindow(GetDlgItem(HWindow,IDD_PA
G1),FALSE);

EnableWindow(GetDlgItem(HWindow,IDD_PA
G2),FALSE);
}

void Tiparire_imp :: cap_tabel()
{
char TextLinec[200],chr;

for(int j=0;j<46;j++)
if(FileName1[j]!='' && FileName1[j+1]=='b'
&& FileName1[j+2]=='u' &&
FileName1[j+3]=='T' && FileName1[j+4]=='\0')
{
long curpos = ftell(fil);
fseek(fil, 0L, SEEK_SET);
//pun pointerul la inceputul fis
for(int i=0;i<10;i++)
//tiparesc capul de tabel
{
for(j=0;j<200 ;j++)
{
fread(&chr,sizeof(char),1,fil);
TextLinec[j]=chr;

if(chr=='\n'){TextLinec[j+1]='\0'; break;}
}

Extent=GetTextExtent(PmDC,TextLinec,LineLe
n);

TextOut(PmDC,0,0,TextLinec,lstrlen(TextLinec)
-1);
}

fseek(fil,curpos,SEEK_CUR);
break;
} //if
}

afisare::afisare(PtWindowsObject AParent,
LPSTR ATitle)
:TWindow(AParent,ATitle)
{
Attr.Style= WS_CAPTION | WS_SYSMENU |
WS_POPUP|WS_THICKFRAME|WS_VISIBLE
| WS_VSCROLL |WS_HSCROLL;
Attr.X=15;
Attr.Y=105;
Attr.W=r.right-30;
Attr.H=400;
este++;
Scroller= new TScroller(this,30,30,60,190);
}

void afisare::SetupWindow()
{
TWindow::SetupWindow();
GetClientRect(HWindow,&rr);

hf=CreateFont(18,0,0,0,FW_BOLD,FALSE,FAL
SE,FALSE,0,0,0,0,(BYTE)VARIABLE_PITCH|
0x02|FF_MODERN,"modern");
}

void afisare::Paint(HDC hdc,PAINTSTRUCT &)
{
FILE *fil;
int i,j;
char buf[200],ch;

fil=fopen(FileName1,"r");
SetTextColor(hdc,RGB(255,0,0));

SetBkColor(hdc,GetSysColor(COLOR_WINDO
W));
SelectObject(hdc,hf);

for(i=0;!feof(fil);i++)
{
for(j=0;j<200 ;j++)
{
fread(&ch,sizeof(char),1,fil);
buf[j]=ch;
if(ch=='\n'){buf[j+1]='\0'; break;}
}
TextOut(hdc,5,20*i+30,buf,strlen(buf)-1);
}
fclose(fil);
}

void afisare::~afisare()
{
este--;
DeleteObject(hf);
}

```

```

/* Subrutina Constant */
*****
#define WIN31
#include <stdio.h>
#include <inputdia.h>
#include <math.h>
#include <bradio.h>
#include "contor.h"
#include "functii.h"
#include "mydiag.h"
#include "constant.h"

extern char path1[40],path2[40],name[40];
extern int com1,com2,com3,com4,com,total;
extern RECT r;
extern CONF Config;
short int OOkk=0;
int hello,hello1,hello2;
extern unsigned char post[1000];
char bufer1[6],bufer2[6];
int probac=0;
float K_et;
int mult,w1=0,w2=0;
char indice;

void TMenWindow::Const_cont()
{
    GetApplication()->ExecDialog(new
    TTont(this,"TURE_DISK"));
    if (OOkk==1){
        ProbaCurenta();
        OOkk=0;
    }
}

CCont::CCont(PTWindowsObject Owner,LPSTR
DName)
:DeBaza(Owner,DName)
{ char buf[7], *buffer;
  int n1,n2;

  memset(&CCC,0x0,sizeof(CCC));
  aceeași = new TBRadioButton( this , IDD_AC ,
  NULL );
  diferit = new TBRadioButton( this , IDD_DIF ,
  NULL );

  Const[0]= new TEdit(this,IDD_CONST1,9);
  Const[1]= new TEdit(this,IDD_CONST2,9);
  Const[2]= new TEdit(this,IDD_CONST3,9);
  Const[3]= new TEdit(this,IDD_CONST4,9);
  Const[4]= new TEdit(this,IDD_CONST5,9);

  Posturi[0]= new TEdit(this,IDD_POST1,30);
  Posturi[1]= new TEdit(this,IDD_POST2,30);
  Posturi[2]= new TEdit(this,IDD_POST3,30);
  Posturi[3]= new TEdit(this,IDD_POST4,30);
  Posturi[4]= new TEdit(this,IDD_POST5,30);

  buffer=inttostring(total);
  if(total!=0) {strcpy(buf,"1-");
                strcat(buf,buffer);
                strcpy(CCC.posttab1,buf);
                }
  CCC.aceeasi = BF_CHECKED;
  CCC.diferit = BF_UNCHECKED;

  char yes[15];
  if(total!=0)
  {
      wsprintf(yes,"Post%d",total);
      n1 =
      GetPrivateProfileInt("Constanta",yes,0,path1);
      n2 =
      GetPrivateProfileInt("Constanta","Post1",0,path1
      );

      if(n1==n2 && n1!=0 )
          strcpy(CCC.consttab1,inttostring(n1));
  }
  TransferBuffer=(void far *)&CCC;
}

void CCont::numar()
{
  for(int k=IDD_POST1;k<=IDD_POST5;k++)
      EnableWindow(GetDlgItem(HWindow
      ,k),FALSE);
  adev=1;
}

void CCont::numar1()
{
  for(int k=IDD_POST1;k<=IDD_POST5;k++)
      EnableWindow(GetDlgItem(HWindow
      ,k),TRUE);
  adev=0;
}

void CCont::Ok(RTMessage)
{
  char p1[30],p2[30],p3[30],p4[30],p5[30];
  char buf[7];

  for(int i=1;i<=250;i++)
  { char *yes=inttostring(i);
    strcpy(buf,"Post");
    strcat(buf,yes);

    WritePrivateProfileString("Constanta",buf,"",path
    1);
  }

  SendDlgItemMsg(IDD_CONST1,WM_GETTEX
  T,9,LPARAM(CCC.consttab1));
}

```

```

SendDlgItemMsg(IDD_CONST2,WM_GETTEXT,9,LPARAM(CCC.consttab2));
SendDlgItemMsg(IDD_CONST3,WM_GETTEXT,9,LPARAM(CCC.consttab3));
SendDlgItemMsg(IDD_CONST4,WM_GETTEXT,9,LPARAM(CCC.consttab4));
SendDlgItemMsg(IDD_CONST5,WM_GETTEXT,9,LPARAM(CCC.consttab5));
SendDlgItemMsg(IDD_POST1,WM_GETTEXT,30,LPARAM(CCC.posttab1));
SendDlgItemMsg(IDD_POST2,WM_GETTEXT,30,LPARAM(CCC.posttab2));
SendDlgItemMsg(IDD_POST3,WM_GETTEXT,30,LPARAM(CCC.posttab3));
SendDlgItemMsg(IDD_POST4,WM_GETTEXT,30,LPARAM(CCC.posttab4));
SendDlgItemMsg(IDD_POST5,WM_GETTEXT,30,LPARAM(CCC.posttab5));

hello1=0;
strcpy(p1,CCC.posttab1);
strcpy(p2,CCC.posttab2);
strcpy(p3,CCC.posttab3);
strcpy(p4,CCC.posttab4);
strcpy(p5,CCC.posttab5);

posturi(p1,CCC.consttab1,"Constanta",path1);
posturi(p2,CCC.consttab2,"Constanta",path1);
posturi(p3,CCC.consttab3,"Constanta",path1);
posturi(p4,CCC.consttab4,"Constanta",path1);
posturi(p5,CCC.consttab5,"Constanta",path1);

if(hello1==1) {
    MessageBeep(0);
    MessageBox(HWindow,"Date invalide !!!","Eroare de domeniu",MB_OK|MB_ICONINFORMATION);
    return;
}
hello1=0;
if(CCC.consttab1[0]!='0' || CCC.consttab1[0]!='0') hello=0;
else {hello=1;hello1=hello;}
if(CCC.consttab2[0]!='0' || CCC.consttab2[0]!='0') hello=0;
else {hello=1;hello1=hello;}
if(CCC.consttab3[0]!='0' || CCC.consttab3[0]!='0') hello=0;
else {hello=1;hello1=hello;}

if(CCC.consttab4[0]!='0' || CCC.consttab4[0]!='0') hello=0;
else {hello=1;hello1=hello;}
if(CCC.consttab5[0]!='0' || CCC.consttab5[0]!='0') hello=0;
else {hello=1;hello1=hello;}

{
    MessageBeep(0);
    MessageBox(HWindow,"Valoarea trebuie sa fie > 0","Eroare de domeniu",MB_OK|MB_ICONINFORMATION);
    return;
}

for(i=0;i<total;i++)
{
    char *yes=IntToString(i+1);
    strcpy(buf,"Post");
    strcat(buf,yes);

if(GetPrivateProfileInt("Constanta",buf,6,path1)==0)
{
    MessageBeep(0);

    MessageBox(HWindow,"Toate constantele trebuie sa fie definite !","Eroare de domeniu",MB_OK|MB_ICONINFORMATION);
    return;
}
}
TDialog::CloseWindow();

//((TMenuItem*)(TMenuItem*)Parent)->Parent->ProbaCurenta();
OOKk=1;
}

TFont::TFont(PWindowsObject Owner,LPSTR DName)
:DeBaza(Owner,DName)
{ char buf[7];
memset(&TTT,0,sizeof(TTT));
aceeasi = new TRadioButton( this , IDD_AC1 , NULL );
diferit = new TRadioButton( this , IDD_DIF1 , NULL );

Const[0]= new TEdit(this,IDD_C1,4);
Const[1]= new TEdit(this,IDD_C2,4);
Const[2]= new TEdit(this,IDD_C3,4);
Const[3]= new TEdit(this,IDD_C4,4);
Const[4]= new TEdit(this,IDD_C5,4);

Posturi[0]= new TEdit(this,IDD_P1,30);
Posturi[1]= new TEdit(this,IDD_P2,30);
Posturi[2]= new TEdit(this,IDD_P3,30);

```



```

Posturi[3]= new TEdit(this,IDD_P4,30);
Posturi[4]= new TEdit(this,IDD_P5,30);
strcpy(TTT.c1,"1");
char *buffer=inttostring(total);
if(total!=0) {strcpy(buf,"1-");

        strcat(buf,buffer);

        strcpy(TTT.p1,buf);
    }
TTT.aceeasi = BF_CHECKED;
TTT.diferit = BF_UNCHECKED;
TransferBuffer=(void far *)&TTT;
}

void TFont::numar()
{
for(int k=IDD_P1;k<=IDD_P5;k++)

        EnableWindow(GetDlgItem(HWindow
,k),FALSE);
adev=1;
}

void TFont::numar1()
{
for(int k=IDD_P1;k<=IDD_P5;k++)

        EnableWindow(GetDlgItem(HWindow
,k),TRUE);
adev=0;
}

void TFont::Ok(RTMessage)
{
char p1[30],p2[30],p3[30],p4[30],p5[30];
char buf[7];
SetCursor(NULL);
for(int i=1;i<=250;i++)
{ char *yes=inttostring(i);
strcpy(buf,"Post");
strcat(buf,yes);

WritePrivateProfileString("Ture_disc",buf,"",path
2);
}

SendDlgItemMsg(IDD_C1,WM_GETTEXT,4,L
PARAM(TTT.c1));

SendDlgItemMsg(IDD_C2,WM_GETTEXT,4,L
PARAM(TTT.c2));

SendDlgItemMsg(IDD_C3,WM_GETTEXT,4,L
PARAM(TTT.c3));

SendDlgItemMsg(IDD_C4,WM_GETTEXT,4,L
PARAM(TTT.c4));

SendDlgItemMsg(IDD_C5,WM_GETTEXT,4,L
PARAM(TTT.c5));

SendDlgItemMsg(IDD_P1,WM_GETTEXT,30,L
PARAM(TTT.p1));

SendDlgItemMsg(IDD_P2,WM_GETTEXT,30,L
PARAM(TTT.p2));

SendDlgItemMsg(IDD_P3,WM_GETTEXT,30,L
PARAM(TTT.p3));

SendDlgItemMsg(IDD_P4,WM_GETTEXT,30,L
PARAM(TTT.p4));

SendDlgItemMsg(IDD_P5,WM_GETTEXT,30,L
PARAM(TTT.p5));

hello1=0;
strcpy(p1,TTT.p1); strcpy(p2,TTT.p2);
strcpy(p3,TTT.p3); strcpy(p4,TTT.p4);
strcpy(p5,TTT.p5);
posturi(p1,TTT.c1,"Ture_disc",path2);
posturi(p2,TTT.c2,"Ture_disc",path2);
posturi(p3,TTT.c3,"Ture_disc",path2);
posturi(p4,TTT.c4,"Ture_disc",path2);
posturi(p5,TTT.c5,"Ture_disc",path2);

if(hello1==1)
{
MessageBeep(0);
MessageBox(HWindow,"Date
invalide !!!", "Eroare de
domeniu",MB_OK|MB_ICONINFORMATION);
return;
}
hello1=0;
if(TTT.c1[0]!='\0' || TTT.c1[0]!='0')
hello=0;
else {hello=1;hello1=hello;}
if(TTT.c2[0]!='\0' || TTT.c2[0]!='0')
hello=0;
else {hello=1;hello1=hello;}
if(TTT.c3[0]!='\0' || TTT.c3[0]!='0')
hello=0;
else {hello=1;hello1=hello;}
if(TTT.c4[0]!='\0' || TTT.c4[0]!='0')
hello=0;
else {hello=1;hello1=hello;}
if(TTT.c5[0]!='\0' || TTT.c5[0]!='0')
hello=0;
else {hello=1;hello1=hello;}
if(hello1==0)
{
MessageBeep(0);
MessageBox(HWindow,"Valoarea
trebuie sa fie in intervalul 1-255", "Eroare de
domeniu",MB_OK|MB_ICONINFORMATION);
return;
}
TDialog::CloseWindow();

int ok=0;

//FILE *fis;
    
```

Asupra implementării echipamentelor cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii - ANEXA 6  
Program calculator PC pentru ERRORREX M

```

//fis=fopen("turedisc.tst","w+");
for(i=0;i<total;i++)
{
    SetCursor(NULL);
    char *yes=IntToString(i+1);
    strcpy(buf,"Post");
    strcat(buf,yes);
    unsigned int
nr=GetPrivateProfileInt("Ture_disc",buf,6,path2);

    for(int k=0;k<3;k++)
    {
        if(com==com1)

transmitere(1,3,(char)(i+1),0x66,(char)nr);
        receptie(1,4);
    }
    if(com==com2)

transmitere(2,3,(char)(i+1),0x66,(char)nr);
        receptie(2,4);
    }
    if(com==com3)

transmitere(3,3,(char)(i+1),0x66,(char)nr);
        receptie(3,4);
    }
    if(com==com4)

transmitere(4,3,(char)(i+1),0x66,(char)nr);
        receptie(4,4);
    }

    if(post[0]==(char)(i+1) &&
post[1]==0x66 && post[2]==(char)nr)
    {ok=0; break;}
    else
        ok=1;
    }
    if(ok==1)
        conversie(i+1,"Eroare la
postul", "");
    //fprintf(fis,"%d %x %x
%x\n",post[0],post[1],post[2],post[3]);
}
//fclose(fis);
GetApplication()->ExecDialog(new
CCont(this,"CONST_CONT"));
}

void DeBaza::validez_domeniu(RTMessage
Msg,int indice)
{
char Txt[9],*p;
float intreg;

Const[indice]->GetText(Txt,9);

if (Msg.LP.Hi==EN_UPDATE)
{
for(p=Txt,*p&&(isdigit(*p)||*p==' ');p++);
if(*p)
{ MessageBeep(0);
Const[indice]-
>SetText(val[indice]);
}
else
lstrcpy(val[indice],Txt);
}

void DeBaza::validez_posturi(RTMessage
Msg,int indice)
{
char Txt[9],*p;
float intreg;

Posturi[indice]->GetText(Txt,9);

if (Msg.LP.Hi==EN_UPDATE)
{
for(p=Txt,*p&&(isdigit(*p)||*p=='-
||*p==' ');p++);
if(*p)
{ MessageBeep(0);
Posturi[indice]-
>SetText(val[indice]);
}
else
lstrcpy(val[indice],Txt);
}

void posturi(char *tablou,char *tablou1,char
*ent,char *fis)
{
int i,k,t;
char tab[30],*buffer,buf[7],szbuf[8];
int valnum[250];

memset(valnum,'\0',250);
memset(tab,'\0',30);

i=0;t=0;
do
{
if(tablou[i]>='0'&&tablou[i]<='9')

if(tablou[i+1]==' '||
tablou[i+1]=='-'||tablou[i+1]=='\0')
{

valnum[t]=atoi(tablou);

for(k=0;k<30;k++)

```

```

        tab[k]=tablou[k+i+1];
        strcpy(tablou,tab);
        i=0;
        }
        if(tablou[i]!=',')
if(tablou[i+1]!='|'||tablou[i+1]!='-')
        {for(k=0;k<30;k++)
tab[k]=tablou[k+i+1];
        strcpy(tablou,tab);i=-1;
        }
        else {t++;
        for(k=0;k<30;k++)
tab[k]=tablou[k+i+1];
strcpy(tablou,tab);
        valnum[t]=atoi(tablou);
        i=-1;
        }
        if(tablou[i]!='-')
if(tablou[i+1]!='|'||tablou[i+1]!='-')
        {for(k=0;k<30;k++)
tab[k]=tablou[k+i+1];
        strcpy(tablou,tab);i=-1;
        }
        else {t++;
        for(k=0;k<30;k++)
tab[k]=tablou[k+i+1];
        int yy=atoi(tablou)-valnum[t-1];
        for(k=t;k<t+yy-1;k++)
        valnum[k]=valnum[k-
1]+1;
        t+=yy-1;
        i=-1;
        }
        i++;
        }while(*tablou!='\0');

for(i=0;valnum[i]!='\0'&& i<250;i++)
        {char *ttt=inttostring(valnum[i]);

        strcpy(buf,"Post");
        strcat(buf,ttt);

GetPrivateProfileString(ent,buf,"",szbuf,8,fis);
        if(szbuf[0]!='\0'){
WritePrivateProfileString(ent,buf,tablou1,fis);
        hello=0;
        }
        else
        {
        char *sz=(char *)malloc(8);
        for(int k=0;k<8;k++)
sz[k]=szbuf[k];
        if(strcmp(sz,tablou1)==0)
WritePrivateProfileString(ent,buf,tablou1,fis);
        else
        {
        hello=1;
        hello1=hello;
        break;
        }
        } //for
        }

void TMenWindow::ProbaCurenta()
{
        probac=1;

        memset(bufe1,0x0,sizeof bufe1);
        memset(bufe2,0x0,sizeof bufe2);

GetPrivateProfileString("Domenii","Curent","",b
ufer1,6,"proba.ini");

GetPrivateProfileString("Domenii","Tensiune","",
bufe2,6,"proba.ini");

do{
        if (
GetPrivateProfileInt("Configurare","Transfcur",0,
name)==1)
        { w1=1;
        if(GetApplication()-
>ExecDialog(new TInputDialog(this,"Domeniul
de curent al statiei","Introduceti curentul:
",bufe1,6))==IDOK)
        if(atoi(bufe1)>150 ||
        atof(bufe1)<=0)
        {
MessageBcep(MB_ICONHAND);

MessageBox(HWindow,"Domeniul curentului
este (0 , 150).","Date invalide
!",MB_OK|MB_ICONHAND);
        }
        else

WritePrivateProfileString("Domenii","Curent",bu
fe1,"proba.ini");
        }
        else break;

        }while(atoi(bufe1)>150 || atof(bufe1)<=0);

do{
        if (
GetPrivateProfileInt("Configurare","Transften",0,
name)==1)
        { w2=1;

```

```

        if(GetApplication()-
>ExecDialog(new TInputDialog(this,"Domeniul
de tensiune ", "Introduceti tensiunea:
", bufer2,6))!=IDOK)
        if(atof(bufer2)>500
|| atof(bufer2)<=0)
        {
        MessageBeep(MB_ICONHAND);
        MessageBox(HWindow,"Domeniul tensiunii este
(0 , 500).","Date invalide
!",MB_OK|MB_ICONHAND);
        }
        else
        WritePrivateProfileString("Domenii","Tensiune",
bufer2,"proba.ini");
        }
        else break;
    }while(atof(bufer2)>500 || atof(bufer2)<=0);
    ProbaTransmit();
}

void TMenWindow::ProbaTransmit()
{
    FILE *fis,*fil;
    char buff[7];
    int ok=1,okk=0;
    float Rc,R_Prod=1;
    unsigned char jq[2];
    char ind;

        probac=1;
        RECT prrect;
        prrect.left = r.right-400 ; prrect.top = 1 ;
        prrect.right = r.right ; prrect.bottom = 40 ;
        InvalidateRect(HWindow,&prrect,TRUE); //fortez apelarea lui WM_PAINT
        mult=0;
        if (
        GetPrivateProfileInt("Configurare","Constcont",0
,name)!=2)
            K_et=atof(Config.CurentContorEtalon);
        else
            K_et=3600*atof(Config.CurentContorEtalon)/(at
of(Config.Curent)*atof(Config.Tensiune)*atof(C
onfig.Nrfaze));

        if(total==0) return;
        OFSTRUCT ofs;
        int ng = OpenFile(path1,&ofs,OF_EXIST); int
ng1 = OpenFile(path2,&ofs,OF_EXIST);
        int cons =
        GetPrivateProfileInt("Constanta","Post1",6,path1
);
    int pos =
    GetPrivateProfileInt("Ture_disc","Post1",6,path2
);
    if(cons==0 || pos==0 || ng==-1 || ng1==-1)
        { MessageBeep(MB_ICONHAND);
        MessageBox(NULL,"Nu ati introdus
constantele si turele pentru contori","Mesaj de
attentionare",MB_OK|MB_ICONEXCLAMATIO
N);
        return;
        }
    calc_max();
    Rc=20*(float)constanta/(tura*K_et);
    if(w1==0 && w2==1) //fara curent
    {
        Prod= atof(bufer2)/atof(Config.TensiuneSec);
        Rc=Rc*Prod;
    }
    if(w2==0 && w1==1) //fara tensiune
    {
        Prod= atof(bufer1)/atof(Config.CurentSec);
        Rc=Rc*Prod;
    }
    if(w1==1 && w2==1)
    {
        Prod=
(atof(bufer1)/atof(Config.CurentSec))*(atof(bufer
2)/atof(Config.TensiuneSec));
        Rc=Rc*Prod;
    }
    if(Rc<1)
    {
        Rc=1/Rc;
        mult=1; //diviz
    }
    for(int i=0;;i++)
        if(pow(2,i)<=Rc && Rc<=pow(2,i+1))
            if(Rc<=(pow(2,i)+pow(2,i+1))/2)
            {
                R=pow(2,i);
                ind=i;
                break;
            }
            else
            {
                R=pow(2,i+1);
                ind=i+1;
                break;
            }
    if(mult==0 && ind>6) {ind=6; R=pow(2,6);}
    if(mult==1 && ind>7) {ind=7; R=pow(2,7);}
    if(mult==1) {ind*=16; R=1/R;}

    for(int k=0;k<3;k++)
    {
        if(com==com1)
        {

```

<pre> transmitere(1,3,0xfe,0x1a,ind); receptie(1,0); } if(com==com2) {     transmitere(2,3,0xfe,0x1a,ind);     receptie(2,0); } if(com==com3) {     transmitere(3,3,0xfe,0x1a,ind);     receptie(3,0); } if(com==com4) {     transmitere(4,3,0xfe,0x1a,ind);     receptie(4,0); } if(post[0]==0xfe &amp;&amp; post[1]==0x1a &amp;&amp; post[2]==(char)ind) {okk=0; break;} else okk=1; } if(okk==1)     MessageBox(HWindow,"Eroare de comunicare", "Mesaj de eroare",MB_ICONEXCLAMATION MB_OK); indice=ind;  fil=fopen("proba.txt","w+"); fprintf(fil,"K_et=%f , Rc=%f , R=%f , n=%d \n",K_et,Rc,R,ind); fprintf(fil,"Tura pt produs maxim: %d \n",(int)tura); fprintf(fil,"Constanta pt produs max: %d \n",(int)constant); if(mult==1)     fprintf(fil,"Divizare\n",mult); else     fprintf(fil,"Multiplicare\n",mult); fclose(fil);  fis=fopen("posturi.tst","w+");  for(i=0;i&lt;total;i++) { char *yes=inttostring(i+1); strcpy(buf,"Post"); strcat(buf,yes); unsigned char hxx=(i+1)-256*((i+1)/256);  int nr = GetPrivateProfileInt("Constanta",buf,6,path1); int tr = GetPrivateProfileInt("Ture_disc",buf,6,path2);  if(nr==0) {     MessageBox(HWindow,"Nu ati introdus constantele pentru contori", "Mesaj de </pre>	<pre> attentionare",MB_OK MB_ICONEXCLAMATIO N); break; } } unsigned int X=(1000*R*tr*K_et/nr)*((float)1/Prod); jq[0]= X/256; jq[1]=X- 256*jq[0]; for(int k=0;k&lt;3;k++) { if(com==com1) { transmitere(1,4,hxx,0x33,jq[1],jq[0]); receptie(1,4); } if(com==com2) { transmitere(2,4,hxx,0x33,jq[1],jq[0]); receptie(2,4); } if(com==com3) { transmitere(3,4,hxx,0x33,jq[1],jq[0]); receptie(3,4); } if(com==com4) { transmitere(4,4,hxx,0x33,jq[1],jq[0]); receptie(4,4); } if(post[0]==hxx &amp;&amp; post[1]==0x33 &amp;&amp; post[2]==jq[1] &amp;&amp; post[3]==jq[0]) {ok=1; break;} else ok=0; } </pre>
--	---

```

        if(ok==0)
conversie(i+1,"Postul ", " este defect sau
lipseste");
        fprintf(fis,"%d
%x %x %x jq[0]=%x jq[1]=%x
\n",post[0],post[1],post[2],post[3],jq[0],jq[1]);
        } //for
fclose(fis);
}

void TMenWindow::calc_max()
{
char buf[7],nume[8];
float *ture,*constante;
int ind=0;

if(total!=0)
{
ture = (float *) malloc(4*total);
constante = (float *) malloc(4*total);
memset(ture,0,sizeof ture);
memset(constante,0,sizeof constante);

for(int i=0;i<total;i++)
{ char *yes=inttostring(i+1);
strcpy(buf,"Post");
strcat(buf,yes);

GetPrivateProfileString("Constanta",buf,"",nume,
8,path1);
constante[i]=atol(nume);

ture[i]=GetPrivateProfileInt("Ture_disc",buf,6,pa
th2);
}

max=(1/constante[0])*ture[0];
for(i=0;i<total-1;i++)

if((1/constante[i])*ture[i]<(1/constante[i+1])*ture
[i+1])
{max=(1/constante[i+1])*ture[i+1];
ind=i+1;
}
else
if((1/constante[i])*ture[i]>(1/constante[i+1])*ture
[i+1])
{max=(1/constante[i])*ture[i];
ind=i;
}

tura=ture[ind];
constanta=constante[ind];
free(ture);
}
}

Metra::Metra(PWindowsObject Owner,LPSTR
DName)
:TDialog(Owner,DName)
{
i5 = new TBRadioButton( this , IDD_5i , NULL
);
i10 = new TBRadioButton( this , IDD_10i ,
NULL );
i15 = new TBRadioButton( this , IDD_15i ,
NULL );
i20 = new TBRadioButton( this , IDD_20i ,
NULL );
i30 = new TBRadioButton( this , IDD_30i ,
NULL );
p5 = new TBRadioButton( this , IDD_5p ,
NULL );
p10 = new TBRadioButton( this , IDD_10p ,
NULL );
p50 = new TBRadioButton( this , IDD_50p ,
NULL );
p100 = new TBRadioButton( this , IDD_100p ,
NULL );
p200 = new TBRadioButton( this , IDD_200p ,
NULL );
p300 = new TBRadioButton( this , IDD_300p ,
NULL );
p400 = new TBRadioButton( this , IDD_400p ,
NULL );
p500 = new TBRadioButton( this , IDD_500p ,
NULL );
p600 = new TBRadioButton( this , IDD_600p ,
NULL );
}

void Metra :: SetupWindow()
{
TDialog::SetupWindow();
int ln = GetPrivateProfileInt("Metra","I
nominal",0,"metra.ini");
if(ln==1)
SendDlgItemMsg(IDD_5i,BM_SETCHECK,1,0)
;
if(ln==2)
SendDlgItemMsg(IDD_10i,BM_SETCHECK,1,0
);
if(ln==3)
SendDlgItemMsg(IDD_15i,BM_SETCHECK,1,0
);
if(ln==4)
SendDlgItemMsg(IDD_20i,BM_SETCHECK,1,0
);
if(ln==5)
SendDlgItemMsg(IDD_30i,BM_SETCHECK,1,0
);
int lp = GetPrivateProfileInt("Metra","I
procent",0,"metra.ini");
if(lp==1)
SendDlgItemMsg(IDD_5p,BM_SETCHECK,1,0)
;
}

```

```

if(Ip==2)
SendDlgItemMsg(IDD_10p,BM_SETCHECK,1,0);
if(Ip==3)
SendDlgItemMsg(IDD_50p,BM_SETCHECK,1,0);
if(Ip==4)
SendDlgItemMsg(IDD_100p,BM_SETCHECK,1,0);
if(Ip==5)
SendDlgItemMsg(IDD_200p,BM_SETCHECK,1,0);
if(Ip==6)
SendDlgItemMsg(IDD_300p,BM_SETCHECK,1,0);
if(Ip==7)
SendDlgItemMsg(IDD_400p,BM_SETCHECK,1,0);
if(Ip==8)
SendDlgItemMsg(IDD_500p,BM_SETCHECK,1,0);
if(Ip==9)
SendDlgItemMsg(IDD_600p,BM_SETCHECK,1,0);
}

void Metra::Ok(RTMessage)
{
int In=0,Ip=0;

if(SendDlgItemMessage(HWindow,IDD_5i,BM_GETCHECK,0,0)==1)//marcat
In=1;

if(SendDlgItemMessage(HWindow,IDD_10i,BM_GETCHECK,0,0)==1)//marcat
In=2;

if(SendDlgItemMessage(HWindow,IDD_15i,BM_GETCHECK,0,0)==1)//marcat
In=3;

if(SendDlgItemMessage(HWindow,IDD_20i,BM_GETCHECK,0,0)==1)//marcat
In=4;

if(SendDlgItemMessage(HWindow,IDD_30i,BM_GETCHECK,0,0)==1)//marcat
In=5;

if(SendDlgItemMessage(HWindow,IDD_5p,BM_GETCHECK,0,0)==1)//marcat
Ip=1;

if(SendDlgItemMessage(HWindow,IDD_10p,BM_GETCHECK,0,0)==1)//marcat
Ip=2;

if(SendDlgItemMessage(HWindow,IDD_50p,BM_GETCHECK,0,0)==1)//marcat
Ip=3;

if(SendDlgItemMessage(HWindow,IDD_100p,BM_GETCHECK,0,0)==1)//marcat
Ip=4;

if(SendDlgItemMessage(HWindow,IDD_200p,BM_GETCHECK,0,0)==1)//marcat
Ip=5;

if(SendDlgItemMessage(HWindow,IDD_300p,BM_GETCHECK,0,0)==1)//marcat
Ip=6;

if(SendDlgItemMessage(HWindow,IDD_400p,BM_GETCHECK,0,0)==1)//marcat
Ip=7;

if(SendDlgItemMessage(HWindow,IDD_500p,BM_GETCHECK,0,0)==1)//marcat
Ip=8;

if(SendDlgItemMessage(HWindow,IDD_600p,BM_GETCHECK,0,0)==1)//marcat
Ip=9;

GetPrivateProfileString("Domenii","Tensiune","",bufer2,6,"proba.ini");
switch(In)
{
case 1 :
{ switch(Ip)
{
case 1 :
strcpy(bufer1,"0.25"); break;
case 2 :
strcpy(bufer1,"0.5"); break;
case 3 :
strcpy(bufer1,"2.5"); break;
case 4 :
strcpy(bufer1,"5"); break;
case 5 :
strcpy(bufer1,"10"); break;
case 6 :
strcpy(bufer1,"15"); break;
case 7 :
strcpy(bufer1,"20"); break;
case 8 :
strcpy(bufer1,"25"); break;
case 9 :
strcpy(bufer1,"30"); break;
} break;
}
case 2 :
{ switch(Ip)
{
case 1 :
strcpy(bufer1,"0.5"); break;
case 2 :
strcpy(bufer1,"1"); break;
case 3 :
strcpy(bufer1,"5"); break;
}
}
}
}

```

```

case 4 :
strcpy(bufer1,"10"); break;
case 5 :
strcpy(bufer1,"20"); break;
case 6 :
strcpy(bufer1,"30"); break;
case 7 :
strcpy(bufer1,"40"); break;
case 8 :
strcpy(bufer1,"50"); break;
case 9 :
strcpy(bufer1,"60"); break;
} break;
}
case 3 :
{ switch(Ip)
{
case 1 :
strcpy(bufer1,"0.75"); break;
case 2 :
strcpy(bufer1,"1.5"); break;
case 3 :
strcpy(bufer1,"7.5"); break;
case 4 :
strcpy(bufer1,"15"); break;
case 5 :
strcpy(bufer1,"30"); break;
case 6 :
strcpy(bufer1,"45"); break;
case 7 :
strcpy(bufer1,"60"); break;
case 8 :
strcpy(bufer1,"75"); break;
case 9 :
strcpy(bufer1,"90"); break;
} break;
}
case 4 :
{ switch(Ip)
{
case 1 :
strcpy(bufer1,"1"); break;
case 2 :
strcpy(bufer1,"2"); break;
case 3 :
strcpy(bufer1,"10"); break;
case 4 :
strcpy(bufer1,"20"); break;
case 5 :
strcpy(bufer1,"40"); break;
case 6 :
strcpy(bufer1,"60"); break;
case 7 :
strcpy(bufer1,"80"); break;
case 8 :
strcpy(bufer1,"100"); break;
case 9 :
strcpy(bufer1,"120"); break;
} break;
}
case 5 :
{ switch(Ip)
{
case 1 :
strcpy(bufer1,"1.5"); break;
case 2 :
strcpy(bufer1,"3"); break;
case 3 :
strcpy(bufer1,"15"); break;
case 4 :
strcpy(bufer1,"30"); break;
case 5 :
strcpy(bufer1,"60"); break;
case 6 :
strcpy(bufer1,"90"); break;
case 7 :
strcpy(bufer1,"120"); break;
case 8 :
strcpy(bufer1,"150"); break;
case 9 :
strcpy(bufer1,"180"); break;
} break;
} //switch In
if(In==0 || Ip==0)
{
MessageBeep(MB_ICONHAND);
MessageBox(NULL,"Trebuie sa marcati si
procentele si curentii ", "Mesaj de
atentionare",MB_OK|MB_ICONHAND);
return;
}
if((In==3 && Ip==9) || (In==4 && (Ip==8 ||
Ip==9)) || (In==5 && (Ip==6 || Ip==7 ||Ip==8 ||
Ip==9) ))
{ MessageBeep(MB_ICONHAND);
MessageBox(NULL,"Valori de curent
nepermise ", "Mesaj de
atentionare",MB_OK|MB_ICONHAND);
}
else
{ char str[3];
gcvt(In,2,str);
WritePrivateProfileString("Metra","I
nominal",str,"metra.ini");
gcvt(Ip,2,str);
WritePrivateProfileString("Metra","I
procent",str,"metra.ini");
WritePrivateProfileString("Domenii","Curent",bu
fer1,"proba.ini");
if (
GetPrivateProfileInt("Configurare","Transfcur",0,
name)==1) w1=1;
else w1=0;
if (
GetPrivateProfileInt("Configurare","Transften",0,
name)==1) w2=1;
else w2=0;
((TMenWindow *)Parent)-
>ProbaTransmit();
}
}

```



```

SE04::SE04(PTWindowsObject Owner,LPSTR
DName)
:TDialog(Owner,DName)
{
ip100 = new TBCheckBox( this , IDD_IPER100
, NULL );
i02 = new TBRadioButton( this , IDD_02s ,
NULL );
i05 = new TBRadioButton( this , IDD_05s ,
NULL );
i1 = new TBRadioButton( this , IDD_1s ,
NULL );
i2 = new TBRadioButton( this , IDD_2s ,
NULL );
i5 = new TBRadioButton( this , IDD_5s ,
NULL );
i10 = new TBRadioButton( this , IDD_10s ,
NULL );
i20 = new TBRadioButton( this , IDD_20s ,
NULL );
i50 = new TBRadioButton( this , IDD_50s ,
NULL );
i100 = new TBRadioButton( this , IDD_100s ,
NULL );
u57 = new TBRadioButton( this , IDD_57s ,
NULL );
u63 = new TBRadioButton( this , IDD_63s ,
NULL );
u120 = new TBRadioButton( this , IDD_120s ,
NULL );
u220 = new TBRadioButton( this , IDD_220s ,
NULL );
u277 = new TBRadioButton( this , IDD_277s ,
NULL );
}

void SE04 :: SetupWindow()
{
TDialog::SetupWindow();
int In = GetPrivateProfileInt("SE04","I
nominal",0,"se04.ini");
if(In==1)
SendDlgItemMsg(IDD_02s,BM_SETCHECK,1,
0);
if(In==2)
SendDlgItemMsg(IDD_05s,BM_SETCHECK,1,
0);
if(In==3)
SendDlgItemMsg(IDD_1s,BM_SETCHECK,1,0)
;
if(In==4)
SendDlgItemMsg(IDD_2s,BM_SETCHECK,1,0)
;
if(In==5)
SendDlgItemMsg(IDD_5s,BM_SETCHECK,1,0)
;

if(In==6)
SendDlgItemMsg(IDD_10s,BM_SETCHECK,1,
0);
if(In==7)
SendDlgItemMsg(IDD_20s,BM_SETCHECK,1,
0);
if(In==8)
SendDlgItemMsg(IDD_50s,BM_SETCHECK,1,
0);
if(In==9)
SendDlgItemMsg(IDD_100s,BM_SETCHECK,1
,0);
int Uf =
GetPrivateProfileInt("SE04","Tensiune",0,"se04.i
ni");
if(Uf==1)
SendDlgItemMsg(IDD_57s,BM_SETCHECK,1,
0);
if(Uf==2)
SendDlgItemMsg(IDD_63s,BM_SETCHECK,1,
0);
if(Uf==3)
SendDlgItemMsg(IDD_120s,BM_SETCHECK,1
,0);
if(Uf==4)
SendDlgItemMsg(IDD_220s,BM_SETCHECK,1
,0);
if(Uf==5)
SendDlgItemMsg(IDD_277s,BM_SETCHECK,1
,0);
}

void SE04::Ok(RTMessage)
{
int In=0,Uf=0,IPER100=0;

if(SendDlgItemMessage(HWindow,IDD_IPER10
0,BM_GETCHECK,0,0)==1)//marcat
IPER100=1;

if(SendDlgItemMessage(HWindow,IDD_02s,BM
_GETCHECK,0,0)==1)//marcat
In=1;

if(SendDlgItemMessage(HWindow,IDD_05s,BM
_GETCHECK,0,0)==1)//marcat
In=2;

if(SendDlgItemMessage(HWindow,IDD_1s,BM_
GETCHECK,0,0)==1)//marcat
In=3;

if(SendDlgItemMessage(HWindow,IDD_2s,BM_
GETCHECK,0,0)==1)//marcat
In=4;

if(SendDlgItemMessage(HWindow,IDD_5s,BM_
GETCHECK,0,0)==1)//marcat
In=5;
}

```

```

if(SendDlgItemMessage(HWindow,IDD_10s,BM_GETCHECK,0,0)==1)//marcat
    In=6;
if(SendDlgItemMessage(HWindow,IDD_20s,BM_GETCHECK,0,0)==1)//marcat
    In=7;
if(SendDlgItemMessage(HWindow,IDD_50s,BM_GETCHECK,0,0)==1)//marcat
    In=8;
if(SendDlgItemMessage(HWindow,IDD_100s,BM_GETCHECK,0,0)==1)//marcat
    In=9;

if(SendDlgItemMessage(HWindow,IDD_57s,BM_GETCHECK,0,0)==1)//marcat
    Uf=1;
if(SendDlgItemMessage(HWindow,IDD_63s,BM_GETCHECK,0,0)==1)//marcat
    Uf=2;
if(SendDlgItemMessage(HWindow,IDD_120s,BM_GETCHECK,0,0)==1)//marcat
    Uf=3;
if(SendDlgItemMessage(HWindow,IDD_220s,BM_GETCHECK,0,0)==1)//marcat
    Uf=4;
if(SendDlgItemMessage(HWindow,IDD_277s,BM_GETCHECK,0,0)==1)//marcat
    Uf=5;

if(IPER100==0) //neapasat
{switch(In)
    {
    case 1 :
    strcpy(bufer1,"0.2"); break;
    case 2 :
    strcpy(bufer1,"0.5"); break;
    case 3 :
    strcpy(bufer1,"1"); break;
    case 4 :
    strcpy(bufer1,"2"); break;
    case 5 :
    strcpy(bufer1,"5"); break;
    case 6 :
    strcpy(bufer1,"10"); break;
    case 7 :
    strcpy(bufer1,"20"); break;
    case 8 :
    strcpy(bufer1,"50"); break;
    case 9 :
    strcpy(bufer1,"100"); break;
    }
}
else //apasat
{switch(In)
    {
    case 1 :
    strcpy(bufer1,"0.002"); break;
    case 2 :
    strcpy(bufer1,"0.005"); break;
    case 3 :
    strcpy(bufer1,"0.01"); break;
    case 4 :
    strcpy(bufer1,"0.02"); break;
    case 5 :
    strcpy(bufer1,"0.05"); break;
    case 6 :
    strcpy(bufer1,"0.1"); break;
    }
}

switch(Uf)
    {
    case 1 :
    strcpy(bufer2,"57.5"); break;
    case 2 :
    strcpy(bufer2,"63.5"); break;
    case 3 :
    strcpy(bufer2,"120"); break;
    case 4 :
    strcpy(bufer2,"220"); break;
    case 5 :
    strcpy(bufer2,"277"); break;
    }

if(In==0 || Uf==0)
    {
    MessageBeep(MB_ICONHAND);
    MessageBox(NULL,"Trebuie sa marcati si tensiunea si curentul","Mesaj de avertizare",MB_OK|MB_ICONHAND);
    return;
    }

if(IPER100==1 && (In==7 || In==8 || In==9))
    {
    MessageBeep(MB_ICONHAND);
    MessageBox(NULL,"Valori nepermise","Mesaj de avertizare",MB_OK|MB_ICONHAND);
    return;
    }

char str[3];
gcvt(In,2,str);
WritePrivateProfileString("SE04","I nominal",str,"se04.ini");
gcvt(Uf,2,str);
WritePrivateProfileString("SE04","Tensiune",str,"se04.ini");
WritePrivateProfileString("Domenii","Curent",bufer1,"proba.ini");
WritePrivateProfileString("Domenii","Tensiune",bufer2,"proba.ini");
}

```

```

if (
GetPrivateProfileInt("Configurare","Transfur",0,
name)==1) w1=1;
else w1=0;
if (
GetPrivateProfileInt("Configurare","Transften",0,
name)==1) w2=1;
else w2=0;
((TMenWindow *)Parent)->ProbaTransmit();
}

Altele::Altele(PTWindowsObject Owner,LPSTR
DName)
:TDialog(Owner,DName)
{
SetupWindow();
for(int i=0;i<nrcurent ;i++)
    al[i] = new TBRadioButton( this ,
440+i , NULL );
for(i=0;i<nrtensiune ; i++)
    u[i] = new TBRadioButton( this, 455+i ,
NULL);
}

void Altele::SetButtonText(void){
char tabl[7];
HWND h1;

for(int i=0;i<nrcurent ;i++)
    {
    gcvt(curenti[i],6,tabl);
    h1 = GetItemHandle(440+i);
    SetWindowText(h1,tabl);
    }
for(i=0;i<nrtensiune ; i++)
    {
    gcvt(tensiuni[i],6,tabl);
    h1 = GetItemHandle(455+i);
    SetWindowText(h1,tabl);
    }
}

void Altele :: SetupWindow()
{
char tab[100],tb[100];
int k,t=0,i=0;

TDialog::SetupWindow();

memset(curenti,'\0',sizeof(curenti));memset(tensi
uni,'\0',sizeof(tensiuni));
memset(tb,'\0',sizeof(tb));
memset(tab,'\0',sizeof(tab));
nrcurent=0;nrtensiune=0;

GetPrivateProfileString("Domenii alte
statii","Curent", "",tab,100,name);
do
    {
    if( tab[i]==' ' || tab[i]=='\0' ||
tab[i]=='\0')
        {
        curenti[t]=atof(tab);

for(k=0;tab[k+i+1]!='\0';k++)
            tb[k]=tab[k+i+1];
for(k=0;k<100;k++)
            tab[k]=tb[k];
            memset(tb,'\0',sizeof(tb));
            i=0;
            t++;
        }
    else i++;
    }while(tab[i]!='\0' || i!=0);
for(i=0;curenti[i]!='\0';i++) nrcurent=i+1;

memset(tb,'\0',sizeof(tb));
memset(tab,'\0',sizeof(tab));
GetPrivateProfileString("Domenii alte
statii","Tensiune", "",tab,100,name);
t=0;i=0;
do
    {
    if( tab[i]==' ' || tab[i]=='\0' ||
tab[i]=='\0')
        {
        tensiuni[t]=atof(tab);

for(k=0;tab[k+i+1]!='\0';k++)
            tb[k]=tab[k+i+1];
for(k=0;k<100;k++)
            tab[k]=tb[k];
            memset(tb,'\0',sizeof(tb));
            i=0;
            t++;
        }
    else i++;
    }while(tab[i]!='\0' || i!=0);
for(i=0;tensiuni[i]!='\0';i++) nrtensiune=i+1;

for(i=0; i<nrcurent && i<16 ;i++)

EnableWindow(GetDlgItem(HWindow,440+i),T
RUE);

for(i=0; i<nrtensiune && i<6 ;i++)

EnableWindow(GetDlgItem(HWindow,455+i),T
RUE);

int ln = GetPrivateProfileInt("Alte statii","I
nominal",0,"altele.ini");
if(ln==1)
SendDlgItemMsg(IDD_0002a1,BM_SETCHECK
,1,0);

```

```

if(In==2)
SendDlgItemMsg(IDD_0005a1,BM_SETCHECK,1,0);
if(In==3)
SendDlgItemMsg(IDD_001a1,BM_SETCHECK,1,0);
if(In==4)
SendDlgItemMsg(IDD_002a1,BM_SETCHECK,1,0);
if(In==5)
SendDlgItemMsg(IDD_005a1,BM_SETCHECK,1,0);
if(In==6)
SendDlgItemMsg(IDD_01a1,BM_SETCHECK,1,0);
if(In==7)
SendDlgItemMsg(IDD_02a1,BM_SETCHECK,1,0);
if(In==8)
SendDlgItemMsg(IDD_05a1,BM_SETCHECK,1,0);
if(In==9)
SendDlgItemMsg(IDD_1a1,BM_SETCHECK,1,0);
if(In==10)
SendDlgItemMsg(IDD_2a1,BM_SETCHECK,1,0);
if(In==11)
SendDlgItemMsg(IDD_5a1,BM_SETCHECK,1,0);
if(In==12)
SendDlgItemMsg(IDD_10a1,BM_SETCHECK,1,0);
if(In==13)
SendDlgItemMsg(IDD_20a1,BM_SETCHECK,1,0);
if(In==14)
SendDlgItemMsg(IDD_50a1,BM_SETCHECK,1,0);
if(In==15)
SendDlgItemMsg(IDD_100a1,BM_SETCHECK,1,0);
int Uf = GetPrivateProfileInt("Alte statii", "Tensiune", 0, "altele.ini");
if(Uf==1)
SendDlgItemMsg(IDD_575a1,BM_SETCHECK,1,0);
if(Uf==2)
SendDlgItemMsg(IDD_635a1,BM_SETCHECK,1,0);
if(Uf==3)
SendDlgItemMsg(IDD_120a1,BM_SETCHECK,1,0);
if(Uf==4)
SendDlgItemMsg(IDD_220a1,BM_SETCHECK,1,0);
if(Uf==5)
SendDlgItemMsg(IDD_277a1,BM_SETCHECK,1,0);
}

void Altele::Ok(RTMessage)

```

```

{
int In=0,Un=0;

if(SendDlgItemMessage(HWND,ID_0002a1,BM_GETCHECK,0,0)==1) In=1;

if(SendDlgItemMessage(HWND,ID_0005a1,BM_GETCHECK,0,0)==1) In=2;

if(SendDlgItemMessage(HWND,ID_001a1,BM_GETCHECK,0,0)==1) In=3;

if(SendDlgItemMessage(HWND,ID_002a1,BM_GETCHECK,0,0)==1) In=4;

if(SendDlgItemMessage(HWND,ID_005a1,BM_GETCHECK,0,0)==1) In=5;

if(SendDlgItemMessage(HWND,ID_01a1,BM_GETCHECK,0,0)==1) In=6;

if(SendDlgItemMessage(HWND,ID_02a1,BM_GETCHECK,0,0)==1) In=7;

if(SendDlgItemMessage(HWND,ID_05a1,BM_GETCHECK,0,0)==1) In=8;

if(SendDlgItemMessage(HWND,ID_1a1,BM_GETCHECK,0,0)==1) In=9;

if(SendDlgItemMessage(HWND,ID_2a1,BM_GETCHECK,0,0)==1) In=10;

if(SendDlgItemMessage(HWND,ID_5a1,BM_GETCHECK,0,0)==1) In=11;

if(SendDlgItemMessage(HWND,ID_10a1,BM_GETCHECK,0,0)==1) In=12;

if(SendDlgItemMessage(HWND,ID_20a1,BM_GETCHECK,0,0)==1) In=13;

if(SendDlgItemMessage(HWND,ID_50a1,BM_GETCHECK,0,0)==1) In=14;

if(SendDlgItemMessage(HWND,ID_100a1,BM_GETCHECK,0,0)==1) In=15;

if(SendDlgItemMessage(HWND,ID_575a1,BM_GETCHECK,0,0)==1) Un=1;

if(SendDlgItemMessage(HWND,ID_635a1,BM_GETCHECK,0,0)==1) Un=2;

if(SendDlgItemMessage(HWND,ID_120a1,BM_GETCHECK,0,0)==1) Un=3;

if(SendDlgItemMessage(HWND,ID_220a1,BM_GETCHECK,0,0)==1) Un=4;
}

```

```

return;
}
char str[3];
gcvt(In,2,str);

WritePrivateProfileString("Alte
statii","I nominal",str,"altele.ini");
gcvt(Un,2,str);

WritePrivateProfileString("Alte
statii","Tensiune",str,"altele.ini");
WritePrivateProfileString("Domenii","
Curent",bufer1,"proba.ini");

WritePrivateProfileString("Domenii","Tensiune",
bufer2,"proba.ini");
if (
GetPrivateProfileInt("Configurare","Transfur",0,
name)==1) w1=1;
else w1=0;
if (
GetPrivateProfileInt("Configurare","Transften",0,
name)==1) w2=1;
else w2=0;

((TMenWindow *)Parent)->ProbaTransmit();
}

/* Subrutina Contor */
*****
#define WIN31 1

#define MIN(a,b) ((a)<(b) ? (a):(b))
#include <windowsx.h>
#include <owl.h>
#include <button.h>
#include <dir.h>
#include <string.h>
#include "contorid.h"
#include "contor.h"
#include "parola.h"

BOOL b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12;
BOOL c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12;
int com1=1,com2=2,com3=3,com4=4,com;
char path[40],path1[40],path2[40],name[40];
HANDLE hInst;
RECT r;
char Device[40],Driver[40],Port[15];
extern char bufer1[6],bufer2[6];
extern int probac;

BYTE huge *TMenWindow :: ReadDib(char
*szFileName)
{
BITMAPFILEHEADER bmfh;
BYTE huge *lpDib;
DWORD dsize,dofset,dheadersize;
int hFile;

if(SendDlgItemMessage(HWindow,IDD_277a1,B
M_GETCHECK,0,0)==1) Un=5;

char strcurenti[30][7],strtensiuni[5][7];
for(int i=0;i<nrcurent;i++)
gcvt(curenti[i],6,strcurenti[i]);
for(i=0;i<nrtensiune;i++)
gcvt(tensiuni[i],6,strtensiuni[i]);

switch(In)
{
case 1 :
strcpy(bufer1,strcurenti[0]); break;
case 2 :
strcpy(bufer1,strcurenti[1]); break;
case 3 :
strcpy(bufer1,strcurenti[2]); break;
case 4 :
strcpy(bufer1,strcurenti[3]); break;
case 5 :
strcpy(bufer1,strcurenti[4]); break;
case 6 :
strcpy(bufer1,strcurenti[5]); break;
case 7 :
strcpy(bufer1,strcurenti[6]); break;
case 8 :
strcpy(bufer1,strcurenti[7]); break;
case 9 :
strcpy(bufer1,strcurenti[8]); break;
case 10 :
strcpy(bufer1,strcurenti[9]); break;
case 11 :
strcpy(bufer1,strcurenti[10]); break;
case 12 :
strcpy(bufer1,strcurenti[11]); break;
case 13 :
strcpy(bufer1,strcurenti[12]); break;
case 14 :
strcpy(bufer1,strcurenti[13]); break;
case 15 :
strcpy(bufer1,strcurenti[14]); break;
}
switch(Un)
{
case 1 :
strcpy(bufer2,strtensiuni[0]); break;
case 2 :
strcpy(bufer2,strtensiuni[1]); break;
case 3 :
strcpy(bufer2,strtensiuni[2]); break;
case 4 :
strcpy(bufer2,strtensiuni[3]); break;
case 5 :
strcpy(bufer2,strtensiuni[4]); break;
}
if(In==0 || Un==0)
{
MessageBeep(MB_ICONHAND);
MessageBox(NULL,"Trebuie sa
marcati si tensiunea si curentul","Mesaj de
avertizare",MB_OK|MB_ICONHAND);
}

```

```

WORD wread;

if(-1 == (hFile = _lopen( szFileName ,
OF_READ | OF_SHARE_DENY_WRITE )))
    return NULL;

if(!_read(hFile,(LPSTR)&bmfh,sizeof(BITMAPFILEHEADER))!=sizeof(BITMAPFILEHEADER))
{
    _lclose(hFile);
    return NULL;
}
if(bmfh.bfType!= *(WORD *)"BM")
{
    _lclose(hFile);
    return NULL;
}
dsize = bmfh.bfSize -
sizeof(BITMAPFILEHEADER);
lpDib = (BYTE huge
*)GlobalAllocPtr(GMEM_MOVEABLE,dsize);
if ( lpDib == NULL)
{
    _lclose(hFile);
    return NULL;
}
dofset=0;
while ( dsize>0)
{
    wread =
(WORD)MIN(32768ul,dsize);
    if(wread
!= _read(hFile,(LPSTR)(lpDib+dofset),wread))
    {
        _lclose(hFile);
        GlobalFreePtr(lpDib);
        return NULL;
    }
    dsize-=wread;
    dofset+=wread;
}
_lclose(hFile);
dheadersize =
GetDibInfoHeaderSize(lpDib);
if(dheadersize <12 ||(dheadersize >12
&& dheadersize <16))
{
    GlobalFreePtr(lpDib);
    return NULL;
}
return lpDib;
}

DWORD TMenWindow :: GetDibInfoHeaderSize(BYTE huge *lpDib)
{return ((BITMAPINFOHEADER huge
*)lpDib)->biSize;
}

WORD TMenWindow :: GetDibWidth(BYTE
huge *lpDib)
{
    if(GetDibInfoHeaderSize(lpDib) ==
sizeof(BITMAPCOREHEADER))
        return
(WORD)((BITMAPCOREHEADER huge
*)lpDib)->bcWidth);
    else
        return (WORD)((BITMAPINFOHEADER
huge *)lpDib)->biWidth);
}

WORD TMenWindow :: GetDibHeight(BYTE
huge *lpDib)
{
    if(GetDibInfoHeaderSize(lpDib) ==
sizeof(BITMAPCOREHEADER))
        return
(WORD)((BITMAPCOREHEADER huge
*)lpDib)->bcHeight);
    else
        return (WORD)((BITMAPINFOHEADER
huge *)lpDib)->biHeight);
}

BYTE huge *TMenWindow ::
GetDibBitsAddr(BYTE huge *lpDib)
{
    DWORD dwNumColors,dwColorTableSize;
    WORD wBitCount;

    if ( GetDibInfoHeaderSize(lpDib) ==
sizeof(BITMAPCOREHEADER))
    {
        wBitCount =
((BITMAPCOREHEADER huge *)lpDib)-
>bcBitCount;
        if(wBitCount!=24)
            dwNumColors=1L <<
wBitCount;
        else
            dwNumColors=0;
        dwColorTableSize =
dwNumColors*sizeof(RGBTRIPLE);
    }
    else
    {
        wBitCount =
((BITMAPINFOHEADER huge *)lpDib)-
>biBitCount;
        if(
GetDibInfoHeaderSize(lpDib)>=36)
            dwNumColors=
((BITMAPINFOHEADER huge *)lpDib)-
>biClrUsed;
        else
            dwNumColors=0;
        if(dwNumColors == 0)
        {
            if(wBitCount!=24)
                dwNumColors=1L<<wBitCount;
            else
                dwNumColors=0;
        }
    }
}

```

```

    }
dwColorTableSize=dwNumColors*sizeof(RGBQUAD);
    }
    return
lpDib+GetDibInfoHeaderSize(lpDib)+dwColorTableSize;
}

TMenuWindow::TMenuWindow(PWindowsObject AParent, LPSTR ATitle)

TWindow(AParent, ATitle)
{
    AssignMenu("Menu");
    LoadLibrary("bwcc.dll");
    Attr.Style=Attr.Style | WS_CAPTION | WS_SYSMENU;
    GetWindowRect(GetDesktopWindow(),&r);
    Attr.X=0;
    Attr.Y=0;
    Attr.W=r.right;//600
    Attr.H=600;
    flag = TRUE;
    win_no=1;

    getcwd(path,30);
    if(path[0]=='b' || path[0]=='B' || path[0]=='a' || path[0]=='A')
        {path[3]='\0';
        strcpy(name,path);
        strcat(name,"menu.ini");
        }
    else {
        strcpy(name,path);
        strcat(name,"\\menu.ini");
        }

    strcpy(path1,path);
    strcpy(path2,path);
    if(path[0]=='b' || path[0]=='B' || path[0]=='a' || path[0]=='A')
        { strcat(path1,"constcon.ini");
        strcat(path2,"turedisc.ini");}
    else
        {
            strcat(path1,"\\constcon.ini");
            strcat(path2,"\\turedisc.ini");
        }

    but1 = new
Buton(this,IDD_BUT1,"",5,3,34,34);
    but2 = new
Buton(this,IDD_BUT2,"",44,3,34,34);
    but3 = new
Buton(this,IDD_BUT3,"",83,3,34,34);
    but4 = new
Buton(this,IDD_BUT4,"",122,3,34,34);

    but5 = new
Buton(this,IDD_BUT5,"",161,3,34,34);
    but6 = new
Buton(this,IDD_BUT6,"",200,3,34,34);
    but7 = new
Buton(this,IDD_BUT7,"",239,3,34,34);
    but8 = new
Buton(this,IDD_BUT8,"",278,3,34,34);
    but9 = new
Buton(this,IDD_BUT9,"",317,3,34,34);
    but10 = new
Buton(this,IDD_BUT10,"",355,3,34,34);
    but11 = new
Buton(this,IDD_BUT11,"",394,3,34,34);
    but12 = new
Buton(this,IDD_BUT12,"",433,3,34,34);

    inf1=LoadBitmap(hInst,"PRBCRT");
    b1=TRUE;
    inf2=LoadBitmap(hInst,"CONSTANTA");
    b2=TRUE;
    inf3=LoadBitmap(hInst,"ACHIZ"); b3=TRUE;
    inf4=LoadBitmap(hInst,"G1"); b4=TRUE;
    inf5=LoadBitmap(hInst,"BULETIN");
    b5=TRUE;
    inf6=LoadBitmap(hInst,"G2"); b6=TRUE;
    inf7=LoadBitmap(hInst,"VERIF"); b7=TRUE;
    inf8=LoadBitmap(hInst,"ACHIZBUL");
    b8=TRUE;
    inf9=LoadBitmap(hInst,"PAROLA");
    b9=TRUE;
    inf10=LoadBitmap(hInst,"CONFIG");
    b10=TRUE;
    inf11=LoadBitmap(hInst,"INVATARE");
    b11=TRUE;
    inf12=LoadBitmap(hInst,"LISTARE");
    b12=TRUE;
    c1=TRUE; c2=TRUE; c3=TRUE;
    c4=TRUE; c5=TRUE; c6=TRUE; c7=TRUE;
    c8=TRUE;
    c9=TRUE; c10=TRUE; c11=TRUE;
    c12=TRUE;

    lpDib = ReadDib("c:\\windows\\atoll.bmp");
    if(lpDib==NULL)
        //MessageBox(HWND,"Fisierul
        pentru fond nu exista sau nu este DIB !", "Mesaj
        de
        eroare", MB_ICONEXCLAMATION|MB_OK);
    else
        {
            DWORD ncol;
            HANDLE handle;
            LLOGPALETTE pal;
            LPBITMAPINFO bmi;
            LPBITMAPCOREINFO bmci;

            ncol =
            IL<<((BITMAPINFOHEADER huge *)lpDib)-
            >biBitCount;

```

```

        handle =
GlobalAlloc(GMEM_ZEROINT |
GMEM_MOVEABLE,
        sizeof(LOGPALETTE) + ncol *
sizeof(PALETTEENTRY));
        if (handle != (HANDLE)NULL)
        {
            pal =
(LPLOGPALETTE)GlobalLock(handle);
            if (pal != NULL)
            {
                pal->palVersion = 0x0300;
                pal->palNumEntries = ncol;

                if
(GetDibInfoHeaderSize(lpDib) ==
sizeof(BITMAPCOREHEADER))
                {
                    bmci = ((BITMAPCOREINFO huge
*)lpDib);

                    for (int i = 0 ; i < ncol ; ++i)
                    {
                        pal->palPalEntry[i].peRed = bmci->
bmciColors[i].rgbtRed;

                        pal->palPalEntry[i].peGreen = bmci->
bmciColors[i].rgbtGreen;

                        pal->palPalEntry[i].peBlue = bmci->
bmciColors[i].rgbtBlue;

                        pal->palPalEntry[i].peFlags = 0;
                    }
                }
                else
                {
                    bmi =
((BITMAPINFO huge *)lpDib);

                    for (int i = 0 ; i < ncol ; ++i)
                    {
                        pal->palPalEntry[i].peRed = bmi->
bmiColors[i].rgbRed;

                        pal->palPalEntry[i].peGreen = bmi->
bmiColors[i].rgbGreen;

                        pal->palPalEntry[i].peBlue = bmi->
bmiColors[i].rgbBlue;

                        pal->palPalEntry[i].peFlags = 0;
                    }
                }
            }
        }
        palette =
CreatePalette(pal);

        GlobalUnlock(handle);
    }
    GlobalFree(handle);
} //if
} //else
}

void TMenWindow :: SetupWindow()
{
    char TmpStr[80];
    char *Src, *Dst;

    GetProfileString("windows","device","",TmpStr,
80);
    Src=TmpStr;
    Dst=Device;
    while(*Src && *Src!=';')
        *Dst++=*Src++;
    *Dst='\0';
    if(*Src)
        ++Src;
    Dst=Driver;
    while(*Src&&*Src!=';'&&*Src!=' ')
        *Dst++=*Src++;
    *Dst='\0';
    if(*Src)
        ++Src;
    Dst=Port;
    while(*Src)
        *Dst++=*Src++;
    *Dst='\0';

    com=GetPrivateProfileInt("Configurare","Com",
0,name);
    Invatare();
    memset(parola,'\0',sizeof parola);

    hfon =
CreateFont(18,0,0,0,FW_BOLD,FALSE,FALSE,
FALSE,0,0,0,(BYTE)VARIABLE_PITCH|0x0
2|FF_MODERN,"modern");
    brec.left=5; brec.top=37;
    brec.bottom=52; brec.right=r.right-100;

    TWindow::SetupWindow();
    memset( TxtParola1,'\0',10);
    SetTimer(HWindow,IDD_TIMP1,10,NULL);
    SetTimer(HWindow,IDD_TIMP2,1000,NULL);
}

TMenWindow::~TMenWindow()
{
    KillTimer(HWindow,IDD_TIMP);
    delete(but1);delete(but2);delete(but3);delete(but4
);
}
    
```



```

delete(but5);delete(but6);delete(but7);delete(but8
);
delete(but9);delete(but10);delete(but12);delete(bu
t12);
DeleteObject(inf1);DeleteObject(inf2);DeleteObj
ect(inf3);DeleteObject(inf4);
DeleteObject(inf5);DeleteObject(inf6);DeleteObj
ect(inf7);DeleteObject(inf8);
DeleteObject(inf9);DeleteObject(inf10);DeleteOb
ject(inf12);DeleteObject(inf12);
}

void TMenWindow::WMPaint(RTMessage)
{HDC hMemDC,hdc;
PAINTSTRUCT ps;
HBRUSH hbr,hbroid;
RECT r;

GetClientRect(HWindow,&r);
hdc=BeginPaint(HWindow, &ps);
hMemDC=CreateCompatibleDC(hdc);

if(palette != (HPALETTE)NULL)
{
    oldpal =
SelectPalette(hdc,palette,FALSE);
RealizePalette(hdc);
}
/*if(lpDib!=NULL)
{
    lpDibBits =GetDibBitsAddr(lpDib);
    cxDib = GetDibWidth(lpDib);
    cyDib = GetDibHeight(lpDib);
    SetStretchBltMode(hdc,COLORONCO
LOR);
    SetDIBitsToDevice(hdc,0,0,cxDib,cyDi
b,0,0,0,cyDib,(LPSTR)lpDibBits,(LPBITMAPIN
FO)lpDib,DIB_RGB_COLORS);
}
else
    StretchDIBits(hdc,0,38,640,480,0,38,64
0,480,(LPSTR)lpDibBits,(LPBITMAPINFO)lpDi
b,DIB_RGB_COLORS,SRCCOPY);
*/
if(oldpal != (HPALETTE)NULL)
{
    oldpal =
SelectPalette(hdc,oldpal,TRUE);
RealizePalette(hdc);
}

hbr=CreateSolidBrush(RGB(192,192,192));
hbroid=SelectObject(hdc,hbr);
Rectangle(hdc,0,0,r.right,38); //bara pentru
butoane
if (probac==1)
{
    char x1[40],x2[40];
    strcpy(x1,"Domeniul de curent :
");strcpy(x2,"Domeniu de tensiune : ");
    strcat(x1,buf1);strcat(x2,buf2);
    SetTextColor(hdc,RGB(255,0,0));
}

SetBkColor(hdc,RGB(192,192,192));
SelectObject(hdc,hfon);
TextOut(hdc,r.right-
300,2,x1,strlen(x1));
TextOut(hdc,r.right-
300,18,x2,strlen(x2));
}
SelectObject(hdc,hbroid);
DeleteObject(hbr);
DeleteDC(hMemDC);
ReleaseDC(HWindow,hdc);
}

void TMenWindow::WMTimer(RTMessage)
{
int n;
HDC dc,mdc;

if (flag==TRUE)
{
    flag=FALSE;

GetPrivateProfileString("Parola","Parola","",paro
la,8,name);
if(parola[0]!='\0')
{
    for(int k=0;k<10;k++)

if(parola[k]!='\0')
{ n=k;
//aflu dim parolei

break;}
for(k=0;k<n;k++)
parola[k]=parola[k]+37;
GetApplication()->ExecDialog(new
IParola(this,"Parola1"));

for(k=0;k<10;k++)
if(parola[k]!='\0')
{MessageBox(HWindow,"Nu
stiti parola,deci nu aveti acces in
program","Mesaj pentru
parola",MB_OK|MB_ICONEXCLAMATION);
exit(EXIT_SUCCESS);
}
}
KillTimer(HWindow,IDD_TIMP1);
Configurare();
}

dc=GetDC(HWindow); // pt bitmapurile de la
butoane
mdc=CreateCompatibleDC(dc);
if(b1==FALSE && c1==TRUE)
{
    c1=FALSE;
    harta=SelectObject(mdc,inf1
);
}
}

```

```

        BitBlt( dc, 5,37,75, 15, mdc,
0, 0, SRCCOPY );
    }
    if(b2==FALSE && c2==TRUE)
    {
        c2=FALSE;
        harta=SelectObject(mdc,inf2
);
        BitBlt( dc, 44,37,85, 15, mdc,
0, 0, SRCCOPY );
    }
    if(b3==FALSE && c3==TRUE)
    {
        c3=FALSE;
        harta=SelectObject(mdc,inf3
);
        BitBlt( dc, 83,37,65, 15, mdc,
0, 0, SRCCOPY );
    }
    if(b4==FALSE && c4==TRUE)
    {
        c4=FALSE;
        harta=SelectObject(mdc,inf4
);
        BitBlt( dc, 122,37,80, 15,
mdc, 0, 0, SRCCOPY );
    }
    if(b5==FALSE && c5==TRUE)
    {
        c5=FALSE;
        harta=SelectObject(mdc,inf5
);
        BitBlt( dc, 161,37,45, 15,
mdc, 0, 0, SRCCOPY );
    }
    if(b6==FALSE && c6==TRUE)
    {
        c6=FALSE;
        harta=SelectObject(mdc,inf6
);
        BitBlt( dc, 200,37,95, 15,
mdc, 0, 0, SRCCOPY );
    }
    if(b7==FALSE && c7==TRUE)
    {
        c7=FALSE;
        harta=SelectObject(mdc,inf7
);
        BitBlt( dc, 239,37,95, 15,
mdc, 0, 0, SRCCOPY );
    }
    if(b8==FALSE && c8==TRUE)
    {
        c8=FALSE;
        harta=SelectObject(mdc,inf8
);
        BitBlt( dc, 278,37,75, 15,
mdc, 0, 0, SRCCOPY );
    }
    if(b9==FALSE && c9==TRUE)
    {
        c9=FALSE;
        harta=SelectObject(mdc,inf9
);
        BitBlt( dc, 317,37,75, 15,
mdc, 0, 0, SRCCOPY );
    }
    if(b10==FALSE && c10==TRUE)
    {
        c10=FALSE;
        harta=SelectObject(mdc,inf10 );
        BitBlt( dc, 355,37,75, 15, mdc, 0, 0,
SRCCOPY );
    }
    if(b11==FALSE && c11==TRUE)
    {
        c11=FALSE;
        harta=SelectObject(mdc,inf11 );
        BitBlt( dc, 394,37,75, 15, mdc, 0, 0,
SRCCOPY );
    }
    if(b12==FALSE && c12==TRUE)
    {
        c12=FALSE;
        harta=SelectObject(mdc,inf12 );
        BitBlt( dc, 433,37,55, 15, mdc, 0, 0,
SRCCOPY );
    }
    SelectObject(mdc,harta);
    DeleteObject(harta);
    DeleteDC(mdc);
    ReleaseDC(HWindow,dc);
}

void TMenWindow ::
WMMouseMove(RTMessage )
{
    RECT dx;

    b1=TRUE;b2=TRUE;b3=TRUE ; b4=TRUE
;b5=TRUE ; b6=TRUE ; b7=TRUE ; b8=TRUE;
;b9=TRUE ; b10=TRUE ; b11=TRUE ;
b12=TRUE;

    if(b1==TRUE && c1==FALSE)
    {
        dx.left=5;dx.top=37;dx.bottom=52;dx.r
ight=80;
        c1=TRUE;
        InvalidateRect(HWindow,&dx,TRUE);
    }
    if(b2==TRUE && c2==FALSE)
    {
        dx.left=44;dx.top=37;dx.bottom=52;dx.
right=dx.left+85;
        c2=TRUE;
        InvalidateRect(HWindow,&dx,TRUE);
    }
    if(b3==TRUE && c3==FALSE)
    {
        dx.left=83;dx.top=37;dx.bottom=52;dx.
right=dx.left+65;
        c3=TRUE;
    }
}
    
```

<pre>         InvalidateRect(HWindow,&amp;dx,TRUE);     }     if(b4==TRUE &amp;&amp; c4==FALSE)     {         dx.left=122;dx.top=37;dx.bottom=52;d         x.right=dx.left+80;         c4=TRUE;         InvalidateRect(HWindow,&amp;dx,TRUE);     }     if(b5==TRUE &amp;&amp; c5==FALSE)     {         dx.left=161;dx.top=37;dx.bottom=52;d         x.right=dx.left+45;         c5=TRUE;         InvalidateRect(HWindow,&amp;dx,TRUE);     }     if(b6==TRUE &amp;&amp; c6==FALSE)     {         dx.left=200;dx.top=37;dx.bottom=52;d         x.right=dx.left+95;         c6=TRUE;         InvalidateRect(HWindow,&amp;dx,TRUE);     }     if(b7==TRUE &amp;&amp; c7==FALSE)     {         dx.left=239;dx.top=37;dx.bottom=52;d         x.right=dx.left+95;         c7=TRUE;         InvalidateRect(HWindow,&amp;dx,TRUE);     }     if(b8==TRUE &amp;&amp; c8==FALSE)     {         dx.left=278;dx.top=37;dx.bottom=52;d         x.right=dx.left+75;         c8=TRUE;         InvalidateRect(HWindow,&amp;dx,TRUE);     }     if(b9==TRUE &amp;&amp; c9==FALSE)     {         dx.left=317;dx.top=37;dx.bottom=52;d         x.right=dx.left+75;         c9=TRUE;         InvalidateRect(HWindow,&amp;dx,TRUE);     }     if(b10==TRUE &amp;&amp; c10==FALSE)     {         dx.left=355;dx.top=37;dx.bottom=52;d         x.right=dx.left+75;         c10=TRUE;         InvalidateRect(HWindow,&amp;dx,TRUE);     }     if(b11==TRUE &amp;&amp; c11==FALSE)     {         dx.left=394;dx.top=37;dx.bottom=52;d         x.right=dx.left+75;         c11=TRUE;         InvalidateRect(HWindow,&amp;dx,TRUE);     }     if(b12==TRUE &amp;&amp; c12==FALSE)     {         dx.left=433;dx.top=37;dx.bottom=52;d         x.right=dx.left+55;         c12=TRUE;     }     </pre>	<pre>         InvalidateRect(HWindow,&amp;dx,TRUE);     } }  void Buton :: WMMouseMove(RTMessage ) {     switch(Attr.Id)     {         case 390 :             b1=FALSE;b2=TRUE;b3=TRUE ; b4=TRUE             ;b5=TRUE ; b6=TRUE ; b7=TRUE ; b8=TRUE;             b9=TRUE ; b10=TRUE ;             b11=TRUE ; b12=TRUE;             break;         case 391 :             b1=TRUE;b2=FALSE;b3=TRUE ; b4=TRUE             ;b5=TRUE ; b6=TRUE ; b7=TRUE ; b8=TRUE;             b9=TRUE ; b10=TRUE ;             b11=TRUE ; b12=TRUE;             break;         case 392 :             b1=TRUE;b2=TRUE;b3=FALSE ; b4=TRUE             ;b5=TRUE ; b6=TRUE ; b7=TRUE ; b8=TRUE;             b9=TRUE ; b10=TRUE ;             b11=TRUE ; b12=TRUE;             break;         case 393 :             b1=TRUE;b2=TRUE;b3=TRUE ; b4=FALSE             ;b5=TRUE ; b6=TRUE ; b7=TRUE ; b8=TRUE;             b9=TRUE ; b10=TRUE ;             b11=TRUE ; b12=TRUE;             break;         case 394 :             b1=TRUE;b2=TRUE;b3=TRUE ; b4=TRUE             ;b5=FALSE ; b6=TRUE ; b7=TRUE ; b8=TRUE;             b9=TRUE ; b10=TRUE ;             b11=TRUE ; b12=TRUE;             break;         case 395 :             b1=TRUE;b2=TRUE;b3=TRUE ; b4=TRUE             ;b5=TRUE ; b6=FALSE ; b7=TRUE ; b8=TRUE;             b9=TRUE ; b10=TRUE ;             b11=TRUE ; b12=TRUE;             break;         case 396 :             b1=TRUE;b2=TRUE;b3=TRUE ; b4=TRUE             ;b5=TRUE ; b6=TRUE ; b7=FALSE ; b8=TRUE;             b9=TRUE ; b10=TRUE ;             b11=TRUE ; b12=TRUE;             break;         case 397 :     </pre>
---	---

```

DeleteObject(hbmp);
}

b1=TRUE;b2=TRUE;b3=TRUE ; b4=TRUE
;b5=TRUE ; b6=TRUE ; b7=TRUE ; b8=FALSE;
      b9=TRUE ; b10=TRUE ;
b11=TRUE ; b12=TRUE;
      break;
      case 398 :

b1=TRUE;b2=TRUE;b3=TRUE ; b4=TRUE
;b5=TRUE ; b6=TRUE ; b7=TRUE ; b8=TRUE;
      b9=FALSE ; b10=TRUE ;
b11=TRUE ; b12=TRUE;
      break;
      case 399 :

b1=TRUE;b2=TRUE;b3=TRUE ; b4=TRUE
;b5=TRUE ; b6=TRUE ; b7=TRUE ; b8=TRUE;
      b9=TRUE ; b10=FALSE ;
b11=TRUE ; b12=TRUE;
      break;
      case 400 :

b1=TRUE;b2=TRUE;b3=TRUE ; b4=TRUE
;b5=TRUE ; b6=TRUE ; b7=TRUE ; b8=TRUE;
      b9=TRUE ; b10=TRUE ;
b11=FALSE ; b12=TRUE;
      break;
      case 401 :

b1=TRUE;b2=TRUE;b3=TRUE ; b4=TRUE
;b5=TRUE ; b6=TRUE ; b7=TRUE ; b8=TRUE;
      b9=TRUE ; b10=TRUE ;
b11=TRUE ; b12=FALSE;
      break;
      }
}

Buton :: Buton(PTWindowsObject AParent, int
AnId, LPSTR AText, int X, int Y, int W, int
H,BOOL fl)

      :TButton(AParent,AnId,AText, X , Y ,
W , H,fl )
{
  Attr.Style = Attr.Style | BS_OWNERDRAW |
WS_VISIBLE;
  BtnUp =
LoadBitmap(hInst,MAKEINTRESOURCE(
UpNoFocus + AnId ));
  BtnDown =
LoadBitmap(hInst,MAKEINTRESOURCE(DnFo
cus + AnId));

  hbmp=BtnUp;
}

Buton :: ~ Buton()
{
DeleteObject(BtnUp);
DeleteObject(BtnDown);
}

void Buton::WMDrawItem(RTMessage Msg)
{
  HDC hMemDC,hdc;
  HBITMAP hb,bm;
  RECT rc;

  LPDRAWITEMSTRUCT lpdis =
(LPDRAWITEMSTRUCT)Msg.LParam;

  SetRect( &rc, lpdis->rcItem.left, lpdis-
>rcItem.top,
          lpdis->rcItem.left
+ Attr.W,
          lpdis->rcItem.top + Attr.H );
  FillRect( lpdis->hDC, &rc, GetStockObject(
( lpdis->itemState & ODS_SELECTED
)?
BLACK_BRUSH : LTGRAY_BRUSH ));

  hMemDC=CreateCompatibleDC(lpdis->hDC);
  SelectObject(hMemDC,hbmp );
  BitBlt( lpdis->hDC, lpdis->rcItem.left + 2, 0,
          32, 32, hMemDC,
          0, 0, SRCCOPY );
  DeleteDC( hMemDC );
}

void Buton :: WMLButtonDown(RTMessage
Msg)
{
  hbmp=BtnDown;
  DefWndProc(Msg);
}

void Buton :: WMLButtonUp(RTMessage Msg)
{
  hbmp=BtnUp;
  DefWndProc(Msg);
}

void TAplNoua::InitMainWindow()
{
  hInst=hInstance;
  MainWindow = new TMenWindow(
NULL,(LPSTR)"ERRORREX-M");
}

void TAplNoua::InitInstance()
{
  TApplication::InitInstance();
  if ( Status == 0 )

```

```

HAccTable =
LoadAccelerators(hInstance, "ACC1");
}

int PASCAL WinMain(HINSTANCE hInstance,
HINSTANCE hPrevInstance,
LPSTR lpCmdLine, int nCmdShow)
{
    TAplNoua MyApp("ERRORREX M",
hInstance, hPrevInstance,
                                lpCmdLine,
nCmdShow);
    MyApp.Run();
    return MyApp.Status;
}

/* Subrutina Functii */
*****
#define WIN31
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <dos.h>
#include <stdio.h>
#include "functii.h"
#include "contor.h"

unsigned char post[1000];
int no,nr_posturi;
char octeti[5];
extern int com1,com2,com3,com4,com;

void conversie(int numar,char *sir1,char *sir2)
{char *tab_dest[40];
int dec,sign;

if(numar>=0&&numar<=9)

tab=ecvt(numar,1,&dec,&sign);
if(numar>=10&&numar<=99)

tab=ecvt(numar,2,&dec,&sign);
if(numar>=100&&numar<=999)

tab=ecvt(numar,3,&dec,&sign);
strcpy(dest,sir1);
strcat(dest,tab);
strcat(dest,sir2);
    MessageBox(NULL,dest,"Mesaj",MB_
OK|MB_ICONINFORMATION);
}

char* inttostring(long numar)
{int dec,sign;
char *ecv;

if(numar>=0&&numar<=9)
    ecv = ecvt(numar,1,&dec,&sign);
if(numar>=10&&numar<=99)
    ecv = ecvt(numar,2,&dec,&sign);
if(numar>=100&&numar<=999)
    ecv = ecvt(numar,3,&dec,&sign);
if(numar>=1000&&numar<=9999)
    ecv = ecvt(numar,4,&dec,&sign);
if(numar>=10000&&numar<=99999)
    ecv = ecvt(numar,5,&dec,&sign);
if(numar>=100000&&numar<=999999)
    ecv = ecvt(numar,6,&dec,&sign);
if(numar>=1000000&&numar<=9999999)
    ecv = ecvt(numar,7,&dec,&sign);
if(numar>=10000000&&numar<=99999999)
    ecv = ecvt(numar,8,&dec,&sign);
return ecv;
}

char* intohexa(int numar)
{
int tri[10];
char tab[10],*tb;
int zecim,y=0,x;
int dec,sign;

memset(tab,'\0',10);
for(int i=0;numar!=0;i++)
    { zecim=numar-16*((int)numar/16);
numar/=16;
tri[i]=zecim;
}
y=i-1;
for(i=0;i<y;i++)
    {int x;
x=tri[i];
tri[i]=tri[y-i];
tri[y-i]=x;
}
for(i=0;i<=y;i++)
    {
switch(tri[i]){
case 10: tab[i]='a';break;
case 11: tab[i]='b';break;
case 12: tab[i]='c';break;
case 13: tab[i]='d';break;
case 14: tab[i]='e';break;
case 15: tab[i]='f';break;
default:
}
}
tb=ecvt(tri[i],1,&dec,&sign);
tab[i]=tb[0];
}
}for
char *hz=(char *)malloc(10);
for(i=0;i<10;i++) hz[i]=tab[i];
free(hz);
return(hz);
}

int hexatoint(unsigned char *n_hexa)
{
int i,len;
double numar=0;
char bb[1];
}

```

```

//n_hexa=(char *)malloc(4);
len=strlen(n_hexa);
for(i=0;i<len;i++)
{
    bb[0]=n_hexa[i];
    switch(n_hexa[i])
    {
        case 'a':
            numar+=10*pow(16,len-i-1);break;
        case 'b':
            numar+=11*pow(16,len-i-1);break;
        case 'c':
            numar+=12*pow(16,len-i-1);break;
        case 'd':
            numar+=13*pow(16,len-i-1);break;
        case 'e':
            numar+=14*pow(16,len-i-1);break;
        case 'f':
            numar+=15*pow(16,len-i-1);break;
        default:
            numar+=atoi(bb)*pow(16,len-i-1);
    }
}
return( numar);
}

void transmitere(int cecom,int nr,char oc0,char
oc1,char oc2,char oc3,char oc4)
{
    int i;

    setare(oc0,oc1,oc2,oc3,oc4);
    if(cecom==1) //portul COM1
    {
        for(i=0;i<nr;i++)
        { unsigned char ts0;
          outportb(0x3f8,octeti[i]);
          do{
              ts0=inportb(0x3fd);
              } while((ts0|0x9f)!=0xff);
          /*etic: asm{
              mov dx,3fdh
              in al,dx
              or al,9fh
              cmp al,0ffh
              jne etic
          } */
        }
    }
    if(cecom==2) //portul COM2
    {
        for(i=0;i<nr;i++)
        {unsigned char ts;
          outportb(0x2f8,octeti[i]);
          do{//if(i==nr) break;
              ts=inportb(0x2fd);
              } while((ts|0x9f)!=0xff);
          }
        }
    }
}

if(cecom==3) //portul COM2
{
    for(i=0;i<nr;i++)
    {unsigned char ts;
      outportb(0x3e8,octeti[i]);
      do{
          ts=inportb(0x3ed);
          } while((ts|0x9f)!=0xff);
      }
    }
}

if(cecom==4) //portul COM2
{
    for(i=0;i<nr;i++)
    {unsigned char ts;
      outportb(0x2e8,octeti[i]);
      do{
          ts=inportb(0x2ed);
          } while((ts|0x9f)!=0xff);
      }
    }
}

void setare(char oct0,char oct1,char oct2,char
oct3,char oct4 )
{ octeti[0]=oct0;
  octeti[1]=oct1;
  octeti[2]=oct2;
  octeti[3]=oct3;
  octeti[4]=oct4;
}

void receptie(int cecom,int nroct)
{
    unsigned char pp1;
    int sec,sec1,ss;
    struct time t;
    for(int i=0;i<20;i++) post[i]='\0';

    if(cecom==1) //COM1
    {
        for(int i=0;;i++)
        { if ( nroct==4 && i==4)
            break;

            gettime(&t);
            sec=(int)t.ti_sec;
            do {
                pp1=inportb(0x3fd);
                if((pp1&0x01)==0x01) break;
                gettime(&t);
                sec1=(int)t.ti_sec;
                ss=abs(sec1-sec);
                } while(ss<1);
                if((pp1&0x01)!=0x01 &&
                ss>=1)
                {no=i;
                  break;
                }
            }
    }
}

```

```

    }
    //receptionez octetii de la COM1
    post[i]=inportb(0x3f8);
    } //for
} //if
if(cecom==2) //COM2
{
    for(int i=0;;i++)
        { if ( nroct==4 && i==4)
        break;
        gettimeofday(&t);
        sec=(int)t.ti_sec;
        do
        {
            pp1=inportb(0x2fd);
            if((pp1&0x01)==0x01) break;
            gettimeofday(&t);
        } while(ss<1);
        ss=abs(sec1-sec);
        if((pp1&0x01)!=0x01 &&
        ss>=1)
        {no=i;
        break;
        }
        post[i]=inportb(0x2f8);
    //receptionez octetii de la COM2
    } //for
    FILE *stream;
    stream=fopen("raoul.gab", "w+");
    for(int k=0;k<no;k++)
        fprintf(stream, "%x\n", post[k]);
        fprintf(stream, "nr de
        posturi:%d\n", no/2);
        fclose(stream);
    } //else com
} //else com
if(cecom==3) //COM2
{
    for(int i=0;;i++)
        { if ( nroct==4 && i==4)
        break;
        gettimeofday(&t);
        sec=(int)t.ti_sec;
        do
        {
            pp1=inportb(0x3ed);
            if((pp1&0x01)==0x01) break;
            gettimeofday(&t);
        } while(ss<1);
        ss=abs(sec1-sec);
        if((pp1&0x01)!=0x01 &&
        ss>=1)
        {no=i;
        break;
        }
        post[i]=inportb(0x2e8);
    //receptionez octetii de la COM2
    } //for
    } //else com
    nr_posturi=no;
}
void setare_com()
{
    if(com==com1)
    {
        CloseComm(OpenComm("COM1",1024,128));
        outportb(0x3fb,0x80); //pun DLAB pe 1
        outportb(0x3f8,0xc); //rata de transfer
        outportb(0x3f9,0x0);
        outportb(0x3fb,0x3); //paritate, stop
        outportb(0x3f9,0x0); //validez intreruperea
    }
    if(com==com2) //com2

```

```

{
    CloseComm(OpenComm("COM2",1024,128));
    outportb(0x2fb,0x80); //pun DLAB pe 1
    outportb(0x2f8,0xc); //rata de transfer
    outportb(0x2f9,0x0);
    outportb(0x2fb,0x3); //paritate,stop
    outportb(0x2f9,0x0); //validez intreruperea
}
if(com==com3)
{
    CloseComm(OpenComm("COM3",1024,128));
    outportb(0x3eb,0x80); //pun DLAB
    pe 1
    outportb(0x3e8,0xc); //rata de transfer
    outportb(0x3e9,0x0);
    outportb(0x3eb,0x3); //paritate,stop
    outportb(0x3e9,0x0); //validez
    intreruperea
}
if(com==com4)
{
    CloseComm(OpenComm("COM4",1024,128));
    outportb(0x2eb,0x80); //pun DLAB
    pe 1
    outportb(0x2e8,0xc); //rata de transfer
    outportb(0x2e9,0x0);
    outportb(0x2eb,0x3); //paritate,stop
    outportb(0x2e9,0x0); //validez
    intreruperea
}
}

/* Subrutina Grafic */
*****
#define WIN31
#include "contor.h"
#include "functii.h"

void TMenWindow::CurbaContor()
{
    transmitere(2,1,0x2A);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 1",MB_OK);
    transmitere(2,1,0x2D);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 2",MB_OK);
    transmitere(2,1,0x30);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 3",MB_OK);
    transmitere(2,1,0x33);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 4",MB_OK);
    transmitere(2,1,0x36);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 5",MB_OK);
    transmitere(2,1,0x39);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 6",MB_OK);
    transmitere(2,1,0x3C);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 7",MB_OK);
    transmitere(2,1,0x3f);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 8",MB_OK);
    transmitere(2,1,0x42);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 9",MB_OK);
    transmitere(2,1,0x45);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 10",MB_OK);
    transmitere(2,1,0x48);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 11",MB_OK);
    transmitere(2,1,0x4B);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 12",MB_OK);
    transmitere(2,1,0x4E);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 13",MB_OK);
    transmitere(2,1,0x51);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 14",MB_OK);
    transmitere(2,1,0x54);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 15",MB_OK);
    transmitere(2,1,0x5F);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 16",MB_OK);
    transmitere(2,1,0x9F);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 17",MB_OK);
    transmitere(2,1,0xA2);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 18",MB_OK);
    transmitere(2,1,0xA5);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 19",MB_OK);
    transmitere(2,1,0xA8);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 20",MB_OK);
    transmitere(2,1,0xab);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 21",MB_OK);
    transmitere(2,1,0xae);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 22",MB_OK);
    transmitere(2,1,0xb1);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 23",MB_OK);
    transmitere(2,1,0xb1);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 24",MB_OK);
    transmitere(2,1,0xb4a);
    MessageBox(HWindow,"Nu este momentan
    implementata","Curba contor 25",MB_OK);
}

void TMenWindow::CurbaMedieLot()
{

```



```

    MessageBox(HWindow,"Nu este momentan
implementata","Curba medie lot",MB_OK);
}

/* Subrutina Help_com */
*****
#define WIN31
#include "help.h"
#include "contor.h"

void TMenWindow::Carte()
{
    MessageBox(HWindow,"Nu este momentan
implementata","Help\\Carte",MB_OK);
}

void TMenWindow::Instructiuni()
{
}

WinHelp(HWindow,"contor.hlp",HELP_CONTE
NTS,0L);
}

void TMenWindow::Alfabetic()
{
    MessageBox(HWindow,"Nu este momentan
implementata","Help\\Alfabetic",MB_OK);
}

/* Subrutina Mydiag */
*****
#define WIN31

#include "contor.h"
#include "mydiag.h"
#include "functii.h"
#include "constant.h"

extern char
path[40],path1[40],path2[40],name[40];
extern int com,com1,com2,com3,com4;
extern unsigned char post[1000];
CONF Config;
extern HANDLE hInst;
extern RECT r;
int flagok=0;
HWND handle1,handle2,handle3;

MyDiag::MyDiag(PWindowsObject
Owner,LPSTR DName)
:TDialog(Owner,DName)
{
    Campuri[0]= new TEdit(this,IDD_CUR,6);
    Campuri[1]= new TEdit(this,IDD_TEN,6);
    Campuri[2]= new TEdit(this,IDD_CCE,8);
    Campuri[3]= new TEdit(this,IDD_NR,2);
    Campuri[4]= new TEdit(this,IDD_CNS,6);
    Campuri[5]= new TEdit(this,IDD_TNS,6);

    hz = new TRadioButton( this , IDD_HZ ,
NULL );
    imp = new TRadioButton( this ,
IDD_IMP , NULL );
    tc = new TBCheckBox( this , IDD_TC ,
NULL );
    tt = new TBCheckBox( this , IDD_TT ,
NULL);
    s1 = new TRadioButton( this ,
IDD_S1 ,NULL );
    s2 = new TRadioButton( this ,
IDD_S2 , NULL );
    s3 = new TRadioButton( this ,
IDD_S3 , NULL );
    man = new TRadioButton( this ,
IDD_MAN , NULL );
    aut = new TRadioButton( this ,
IDD_AUT , NULL );
    sw=0;
    st1=0;st2=0;
    //TransferBuffer=(void far *)&Config
    // memset(&Config,0x0,sizeof Config);
}

void MyDiag::SetupWindow()
{
    TDialog::SetupWindow();

    GetPrivateProfileString("Configurare","Curent","
",Config.Curent,8,name);

    SendDlgItemMsg(IDD_CUR,WM_SETTEXT,0,
LPARAM(Config.Curent));

    GetPrivateProfileString("Configurare","Tensiune
","",Config.Tensiune,8,name);

    SendDlgItemMsg(IDD_TEN,WM_SETTEXT,0,
LPARAM(Config.Tensiune));

    GetPrivateProfileString("Configurare","Constcoe
t","",Config.CurentContorEtalon,8,name);

    SendDlgItemMsg(IDD_CCE,WM_SETTEXT,0,
LPARAM(Config.CurentContorEtalon));

    GetPrivateProfileString("Configurare","Nrfaze","
",Config.Nrfaze,5,name);

    SendDlgItemMsg(IDD_NR,WM_SETTEXT,0,L
PARAM(Config.Nrfaze));

    GetPrivateProfileString("Configurare","Curent_S
ec","",Config.CurentSec,6,name);

    SendDlgItemMsg(IDD_CNS,WM_SETTEXT,0,
LPARAM(Config.CurentSec));

    GetPrivateProfileString("Configurare","Tensiune
_Sec","",Config.TensiuneSec,6,name);

    SendDlgItemMsg(IDD_TNS,WM_SETTEXT,0,
LPARAM(Config.TensiuneSec));
}

```

```

n=
GetPrivateProfileInt("Configurare","Constcont",0
,name);

mc=GetPrivateProfileInt("Configurare","Transfc
ur",0,name);

mt=GetPrivateProfileInt("Configurare","Transfte
n",0,name);

ms=GetPrivateProfileInt("Configurare","Tipstatie
",0,name);

ma=GetPrivateProfileInt("Configurare","Tipachiz
",0,name);

if(n==1)
SendDlgItemMsg(IDD_HZ,BM_SETCHECK,1,0
);
if(n==2)
SendDlgItemMsg(IDD_IMP,BM_SETCHECK,1,
0);
if(mc==1)
SendDlgItemMsg(IDD_TC,BM_SETCHECK,1,0
);
if(mt==1)
SendDlgItemMsg(IDD_TT,BM_SETCHECK,1,0
);
if(ms==1)
SendDlgItemMsg(IDD_S1,BM_SETCHECK,1,0
);
if(ms==2)
SendDlgItemMsg(IDD_S2,BM_SETCHECK,1,0
);
if(ms==3)
SendDlgItemMsg(IDD_S3,BM_SETCHECK,1,0
);
if(ma==1)
SendDlgItemMsg(IDD_MAN,BM_SETCHECK,
1,0);
if(ma==2)
SendDlgItemMsg(IDD_AUT,BM_SETCHECK,1
,0);

if(n==1)
{
EnableWindow(GetDlgItem(HWindow,IDD_CU
R),TRUE);

EnableWindow(GetDlgItem(HWindow,IDD_TE
N),TRUE);
}
if(mc!=1)

{EnableWindow(GetDlgItem(HWindow,IDD_C
NS),FALSE);st1=0;}
else st1=1;
if(mt!=1)

{EnableWindow(GetDlgItem(HWindow,IDD_T
NS),FALSE);st2=0;}
else st2=1;
}

MyDiag::~MyDiag()
{
for(int i=0 ; i<6 ; i++) delete Campuri[i];
delete hz; delete imp;
delete tc; delete tt;
delete s1; delete s2; delete s3;
delete man; delete aut;
}

void MyDiag::verifica_domeniu(RTMessage
Msg,int indice,float capat1,float capat2)
{
char Txt[9],*p;
float intreg;
char SirMsg[40]="\0",sir[9];
Txt[0]='\0';

Campuri[indice]->GetText(Txt,9);
if(!sw)
{
lstrcpy(VechiulNr[indice],Txt);
sw=1;
}
if(Msg.LP.Hi==EN_UPDATE)
{
for(p=Txt;*p&&(isdigit(*p)||*p==' ');p
++);
if(*p)
{
MessageBeep(0);
Campuri[indice]-
>SetText(VechiulNumar[indice]);
}
else
lstrcpy(VechiulNumar[indice],Txt);
}
else if(Msg.LP.Hi==EN_KILLFOCUS)
{
intreg=atof(Txt);
sw=0;
if
(((intreg>capat2)||((intreg<capat1)))|| Txt[0]!='\0')
{
MessageBeep(0);

strcpy(SirMsg,"Intervalul corect este: ");
gcvt(capat1,4,sir);
strcat(SirMsg,sir); strcat(SirMsg," - ");
gcvt(capat2,4,sir);strcat(SirMsg,sir);
}
}
}

```

<pre> MessageBox(NULL,SirMsg,"Eroare de domeniu",MB_ICONEXCLAMATION   MB_OK);         Campuri[indice]- &gt;SetText(VechiulNr[indice]);         }     }  void MyDiag::Ok(RTMessage) { int ok=1;     SetCursor(NULL);  SendDlgItemMsg(IDD_CUR,WM_GETTEXT,6, LPARAM(Config.Curent));  SendDlgItemMsg(IDD_TEN,WM_GETTEXT,6, LPARAM(Config.Tensiune));  SendDlgItemMsg(IDD_CCE,WM_GETTEXT,8, LPARAM(Config.CurentContorEtalon));  SendDlgItemMsg(IDD_NR,WM_GETTEXT,2,L PARAM(Config.Nrfaze));  SendDlgItemMsg(IDD_CNS,WM_GETTEXT,6, LPARAM(Config.CurentSec));  SendDlgItemMsg(IDD_TNS,WM_GETTEXT,6, LPARAM(Config.TensiuneSec));  SendDlgItemMsg(IDD_CNS,WM_GETTEXT,6, LPARAM(Config.CurentSec));  SendDlgItemMsg(IDD_TNS,WM_GETTEXT,6, LPARAM(Config.TensiuneSec));  if(SendDlgItemMessage(HWindow,IDD_HZ,BM _GETCHECK,0,0)==0) Config.hz=0; else Config.hz=1;  if(SendDlgItemMessage(HWindow,IDD_IMP,B M_GETCHECK,0,0)==0) Config.imp=0; else Config.imp=1;  if(SendDlgItemMessage(HWindow,IDD_TC,BM _GETCHECK,0,0)==0) Config.tc=0; else Config.tc=1;  if(SendDlgItemMessage(HWindow,IDD_TT,BM _GETCHECK,0,0)==0) Config.tt=0; else Config.tt=1;  if(SendDlgItemMessage(HWindow,IDD_S1,BM _GETCHECK,0,0)==0) Config.s1=0; else Config.s1=1;  if(SendDlgItemMessage(HWindow,IDD_S2,BM </pre>	<pre> _GETCHECK,0,0)==0) Config.s2=0; else Config.s2=1;  if(SendDlgItemMessage(HWindow,IDD_S3,BM _GETCHECK,0,0)==0) Config.s3=0; else Config.s3=1;  if(SendDlgItemMessage(HWindow,IDD_MAN,B M_GETCHECK,0,0)==0)Config.man=0; else Config.man=1;  if(SendDlgItemMessage(HWindow,IDD_AUT,B M_GETCHECK,0,0)==0) Config.aut=0; else Config.aut=1;  WritePrivateProfileString("Configurare","Curent" ,Config.Curent,name);  WritePrivateProfileString("Configurare","Tensiu ne",Config.Tensiune,name);  WritePrivateProfileString("Configurare","Constc oet",Config.CurentContorEtalon,name);  WritePrivateProfileString("Configurare","Nrfaze" ,Config.Nrfaze,name);  WritePrivateProfileString("Configurare","Curent _sec",Config.CurentSec,name);  WritePrivateProfileString("Configurare","Tensiu ne_sec",Config.TensiuneSec,name);          if(Config.hz==1)                  WritePrivateProfileString("Configurare ","Constcont","1",name);                 if(Config.imp==1)                          WritePrivateProfileString("Configurare ","Constcont","2",name);                         if(Config.tc==1)                                  WritePrivateProfileString("Configurare ","Transcur","1",name);                                 else  WritePrivateProfileString("Configurare ","Transcur","",name);  if(Config.tt==1)  WritePrivateProfileString("Configurare ","Transften","1",name);   else  WritePrivateProfileString("Configurare ","Transften","",name);  if(Config.s1==1) </pre>
--	--

```

        WritePrivateProfileString("Configurare
", "Tipstatie", "1", name);
        if(Config.s2==1)

        WritePrivateProfileString("Configurare
", "Tipstatie", "2", name);
        if(Config.s3==1)

        WritePrivateProfileString("Configurare
", "Tipstatie", "3", name);
        if(Config.man==1)
        {

        WritePrivateProfileString("Configurare
", "Tipachiz", "1", name);
        for(int k=0;k<3;k++)
        { unsigned char oct=0x80;

                if(com==com1)
                {

        transmitere(1,3,0xfe,0x1a,oct);

        receptie(1,0);

                }
                if(com==com2)
                {

        transmitere(2,3,0xfe,0x1a,oct);

        receptie(2,0);

                }
                if(com==com3)
                {

        transmitere(3,3,0xfe,0x1a,oct);

        receptie(3,0);

                }
                if(com==com4)
                {

        transmitere(4,3,0xfe,0x1a,oct);

        receptie(4,0);

                }

                if(post[0]==0xfe &&
post[1]==0x1a && post[2]==oct)
                {
        ok=1;break;}
                else ok=0;
                }
                if(ok==0)
        MessageBox(HWindow,"Eroare de
comunicare","Mesaj de
eroare",MB_ICONEXCLAMATION|MB_OK);
        }
        if(Config.aut==1)
        {

        WritePrivateProfileString("Configurare
", "Tipachiz", "2", name);
        for(int k=0;k<3;k++)
        { unsigned
        {

        transmitere(1,3,0xfe,0x1a,oct);

        receptie(1,0);

                }
                if(com==com1)
                {

        transmitere(2,3,0xfe,0x1a,oct);

        receptie(2,0);

                }
                if(com==com2)
                {

        transmitere(3,3,0xfe,0x1a,oct);

        receptie(3,0);

                }
                if(com==com3)
                {

        transmitere(4,3,0xfe,0x1a,oct);

        receptie(4,0);

                }

                if(post[0]==0xfe
&& post[1]==0x1a && post[2]==oct)
                {
        ok=1;break;}
                else
        {
        ok=0;
        }
                if(ok==0)
        MessageBox(HWindow,"Eroare de
comunicare","Mesaj de
eroare",MB_ICONEXCLAMATION|MB_OK);
        }

        flagok=1;
        TDialog::CloseWindow();
        }

        void TMenWindow::Configurare()
        {
        HDC hdc = GetDC(HWindow);
        char x1[70] = "
";
        x2[70] = "
";
        SetBkColor(hdc,RGB(192,192,192));
        TextOut(hdc,r.right-310,2,x1,strlen(x1)-1);
        TextOut(hdc,r.right-310,17,x2,strlen(x2)-1);
    
```

```

ReleaseDC(HWindow,hdc);

GetApplication()->ExecDialog(new
MyDiag(this,"Configurare"));
if(flagok==1)
{
    DestroyWindow(handle1);
    DestroyWindow(handle2);
    DestroyWindow(handle3);
    if (
GetPrivateProfileInt("Configurare","Transfur",0,
name)==1)
    {
        if( Config.s1==1 )
        {
            GetApplication()-
>MakeWindow(new Metra(this,"METRA"));
            handle1 = GetLastChild()-
>HWindow;
        }
        if(Config.s2==1)
        {
            GetApplication()-
>MakeWindow(new SE04(this,"SE04"));
            handle2 = GetLastChild()-
>HWindow;
        }
        if(Config.s3==1)
        {
            Altele *p;
            GetApplication()-
>MakeWindow(p=new Altele(this,"ALTELE"));
            p->SetButtonText();
            handle3 = GetLastChild()-
>HWindow;
        }
    }
} //if

}

/*long dragos = GetFreeSpace(0);
char kkk[10];
gcvt(dragos,10,kkk);

MessageBox(HWindow,kkk,"Mesaj",MB_OK);*/

void MyDiag :: EditImp(RTMessage )
{
    SendDlgItemMessage(HWindow,IDD_CUR,WM_
_SETTEXT,0,LPARAM(""));

    SendDlgItemMessage(HWindow,IDD_TEN,WM_
_SETTEXT,0,LPARAM(""));

    EnableWindow(GetDlgItem(HWindow,IDD_CU
R),FALSE);
}

void MyDiag :: EditHz(RTMessage)
{
    SendDlgItemMessage(HWindow,IDD_CUR,WM_
_SETTEXT,0,LPARAM(Config.Curent));

    SendDlgItemMessage(HWindow,IDD_TEN,WM_
_SETTEXT,0,LPARAM(Config.Tensiune));

    EnableWindow(GetDlgItem(HWindow,IDD_CU
R),TRUE);

    EnableWindow(GetDlgItem(HWindow,IDD_TE
N),TRUE);
}

void MyDiag :: EditCu(RTMessage)
{
    if(st1%2==0)
    {
        SendDlgItemMessage(HWindow,IDD_
CNS,WM_SETTEXT,0,LPARAM(Config.Curen
tSec));

        EnableWindow(GetDlgItem(HWindow
,IDD_CNS),TRUE);
    }
    else
    {
        SendDlgItemMessage(HWindow,IDD_
CNS,WM_SETTEXT,0,LPARAM(""));

        EnableWindow(GetDlgItem(HWindow
,IDD_CNS),FALSE);
    }
    st1++;
}

void MyDiag :: EditTen(RTMessage)
{
    if(st2%2==0)
    {
        SendDlgItemMessage(HWindow,IDD_
TNS,WM_SETTEXT,0,LPARAM(Config.Tensiu
neSec));

        EnableWindow(GetDlgItem(HWindow
,IDD_TNS),TRUE);
    }
    else
    {
        SendDlgItemMessage(HWindow,IDD_
TNS,WM_SETTEXT,0,LPARAM(""));
    }
}

```

```

EnableWindow(GetDlgItem(HWindow
,IDD_TNS),FALSE);
}
st2++;
}

void MyDiag :: EditCurent(RTMessage Msg)
{
verifica_domeniu(Msg,0,0.01,50);
}

void MyDiag :: EditTensiune(RTMessage Msg)
{
verifica_domeniu(Msg,1,40,500);
}

void MyDiag :: EditCCE(RTMessage Msg)
{
verifica_domeniu(Msg,2,40,1000000);
}

void MyDiag :: EditNrFaze(RTMessage Msg)
{
verifica_domeniu(Msg,3,1,3);
}

void MyDiag :: CSec(RTMessage Msg)
{
verifica_domeniu(Msg,4,0.1,50);
}

void MyDiag :: TSec(RTMessage Msg)
{
verifica_domeniu(Msg,5,40,500);
}

/* Subrutina Noname */
*****
#include <stdio.h>
#include <string.h>

void main()
{
FILE *fis;
long pos,curpos;
char buffer[55];

strcpy(buffer,"c:\\users\\errorexm\\lumi.bul");
if((fis = fopen(buffer,"r"))!=NULL)
{
pos = ftell(fis);
fseek(fis,35,SEEK_CUR);
curpos = ftell(fis);
}
fclose(fis);
}

/* Subrutina Parola */
*****
#define WIN31
#include "contor.h"

```

```

#include "parola.h"

extern char name[40];

void TMenWindow::Parola()
{
TParola *obpar2;
SParola *obpar3;

char tab[10];
int n,i,k;
GetPrivateProfileString("Parola","Parola","",parola,8,name);
if(parola[0]!='\0')
{
for(k=0;k<10;k++)

if(parola[k]!='\0')
{ n=k;
//aflu dim parolei

break;}
for(k=0;k<n;k++)
parola[k]=parola[k]+37;
}

if(parola[0]!='\0') //se pune parola la
program
{
if( GetApplication()->ExecDialog(obpar2 = new
TParola(this,"Parola2"))==IDOK)
{
for(i=0;i<10;i++)

if((obpar2-
>TxtParola2).prima[i]!='\0')
{ n=i;

break;}
for(i=0;i<10&&(obpar2-
>TxtParola2).prima[0]!='\0';i++)
if((obpar2-
>TxtParola2).prima[i]==(obpar2-
>TxtParola2).doua[i])
{
tab[i] = (obpar2-
>TxtParola2).prima[i]-37;
if(i==9)
{
for(i=n;i<10;i++)

tab[i]='\0';

WritePrivateProfileString("Parola","Parola",tab,n
ame);

MessageBox(HWindow,"Parola a fost
actualizata","Mesaj pentru
parola",MB_OK|MB_ICONEXCLAMATION);
}
}
}
}
}
}

```

```

else {
    MessageBox(HWindow,"Confirmarea parolei
este gresita.\nProgramul ramane fara
parola!", "Mesaj pentru
parola",MB_OK|MB_ICONEXCLAMATION);
    break;
} //de la ok
memset(&(obpar2->TxtParola2),0x0,sizeof
obpar2->TxtParola2);
} // if(parola[0]!='\0')
else
    if( GetApplication()->ExecDialog(obpar3=new
SParola(this,"Parola3"))==IDOK)
    {
        for(i=0;i<10;i++)
            if((obpar3->TxtParola3).doua[i]!='\0')
                {n=i;
                break;
                }
        for(i=0;i<10;i++)
            if(parola[i]!=(obpar3-
>TxtParola3).prima[i])
                {MessageBox(HWindow,"Nu stiti parola,deci nu
o puteti schimba !", "Mesaj pentru
parola",MB_OK|MB_ICONEXCLAMATION);
                break;}
            else
                if((obpar3-
>TxtParola3).doua[i]!=(obpar3-
>TxtParola3).treia[i] & (obpar3-
>TxtParola3).doua[i]!='\0')
                    {
                        MessageBox(HWindow,"Confirmarea parolei e
gresita.\nRamane valabila parola anterioara!",
"Mesaj pentru
parola",MB_OK|MB_ICONEXCLAMATION);
                        break;}
                    else {
                        tab[i] =
(obpar3->TxtParola3).doua[i]-37;
                        if(i==9)
                            {
                                for(k=n;k<10;k++) tab[k]='\0';
                                if((obpar3->TxtParola3).doua[0]!='\0' & (obpar3-
>TxtParola3).treia[0]!='\0')
                                    {
                                        WritePrivateProfileString("Parola","Parola",tab,n
ame);
                                        MessageBox(HWindow,"Parola a
fost stearsa", "Mesaj pentru
parola",MB_OK|MB_ICONEXCLAMATION);
                                    }
                                else {
                                    WritePrivateProfileString("Parola","Parola",tab,name);
                                    MessageBox(HWindow,"Parola a fost
actualizata", "Mesaj pentru
parola",MB_OK|MB_ICONEXCLAMATION);
                                }
                            } //else
                            memset(&(obpar3->TxtParola3),0x0,sizeof
obpar3->TxtParola3);
                            } //if(ExecDialog)=0k
                            }
                            TParola::TParola(PtWindowsObject
Owner,LPSTR DName)
                                :TDialog(Owner,DName)
                                {
                                    memset(&TxtParola2,0x0, sizeof TxtParola2);
                                    intr[0]= new TEdit(this,IDD_PARA2,10);
                                    intr[1]= new TEdit(this,IDD_PARB2,10);
                                    TransferBuffer=(void far *)&TxtParola2;
                                }
                                IParola::IParola(PtWindowsObject
Owner,LPSTR DName)
                                    :TDialog(Owner,DName)
                                    {
                                        intrare= new TEdit(this,IDD_PAR1,10);
                                        TransferBuffer=(void far *)&((TMenWindow
*)Parent)->TxtParola1;
                                    }
                                    SParola::SParola(PtWindowsObject
Owner,LPSTR DName)
                                        :TDialog(Owner,DName)
                                        {
                                            memset(&TxtParola3,0x0, sizeof TxtParola3);
                                            schimb[0]= new TEdit(this,IDD_PARA3,10);
                                            schimb[1]= new TEdit(this,IDD_PARB3,10);
                                            schimb[2]= new TEdit(this,IDD_PARC3,10);
                                            TransferBuffer=(void far *)&TxtParola3;
                                        }
                                        /* Subrutina Runsucc */
                                        *****
                                        #define WIN31
                                        #include <filedial.h>
                                        #include <listbox.h>
                                        #include <io.h>
                                        #include <stdio.h>
                                        #include "contor.h"
                                        #include "runsucc.h"

```

Asupra implementării echipamentelor cu logică programată pentru comanda unei stații de  
etalonat contoare de energie electrică cu achiziția erorii - ANEXA 6  
Program calculator PC pentru ERRORREX M

```

#include "constant.h"
explicit[i]=ch;

        _lread(hFile,&ch,1);
        i++;
    }
    void TMenWindow :: Succlans()
    {
        //if(GetApplication()->ExecDialog(new
        RulareSucc(this,32500,"*.scc" )==IDOK)
        // Const_cont();
    }

    /*void RulareSucc :: Ok(RTMessage)
    {
        TDialog :: CloseWindow();
    } */

    void RulareSucc :: HandleFList(RTMessage
    Msg)
    {
        HFILE hFile;
        char path_name[50],explic[70];
        char slash[3] = "\\";
        int i;
        char ch;

        if ( Msg.LP.Hi == LBN_SELCHANGE )
        {
            idx = listb1->GetSelIndex();
            listb1-
            >GetString((LPSTR)tabl, idx);

            SendDlgItemMsg(ID_FNAME,WM_S
            ETTEXT,0,LPARAM(tabl));

            SendDlgItemMsg(IDD_DIREC,WM_G
            ETTEXT,30,LPARAM(FilePath));
            strcpy(path_name,FilePath);
            strcat(path_name,slash);
            strcat(path_name,tabl);

            listb2->ClearList();
            if(-1 == (hFile = _lopen( tabl
            , OF_READ | OF_SHARE_DENY_WRITE )))
            MessageBox(HWindow,"Eroare deschidere
            fisier","Mesaj",MB_OK);
            else
            {
                do
                {
                    i=0;

                    _lread(hFile,&ch,1);
                    memset(explic,0x0, sizeof explic);
                    while(ch!='\n')
                    {
                        _lread(hFile,&ch,1);
                        i++;
                    }
                    listb2-
                    >AddString(explic);

                    _lread(hFile,&ch,1);
                    if(ch!='\n' || ch!='
                    ') _llseek(hFile,-1,1);
                } while(ch!='\n' && ch!='
                ');
                _lclose(hFile);
            } //if
        }

        /* Subrutina Succespr */
        *****
        #define WIN31
        #include <listbox.h>
        #include <scrollba.h>
        #include <filedial.h>
        #include "contor.h"
        #include "functii.h"
        #include "succespr.h"

        Succ *ob;

        void TMenWindow::Succesiune()
        {
            //GetApplication()->ExecDialog(ob = new
            Succ(this,"SUCCES"));
        }

        Succ::Succ( PTWindowsObject Owner , LPSTR
        DName )
        :TDialog(Owner,DName)
        {
            listb1 = new TListBox(this,IDD_LIST1);
            listb2 = new TListBox(this,IDD_LIST2);
        }

        void Succ::SetupWindow()
        {
            char sir1[13][9]={" 1"," 5"," 10","
            20"," 50"," 100"," 200"," 300"," 400","
            600","100R","100S","100T"};
            char sir2[3][4]={"1","0.5","0.8"};
            char tabl[30];

            TDialog::SetupWindow();
            idx=0;len=0;
            for(int i=1;i<4;i++)
            for(int j=1;j<14;j++)
            {
                wsprintf(tabl,"I = %s%c , cos(fi) =
                %s",sir1[j-1],%',sir2[i-1]);
            }
        }
    }

```



Asupra implementării echipamentelor cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii - ANEXA 6  
 Program calculator PC pentru ERRORREX M

```

        if (listb1->AddString(tab1) < 0) break;
        if (i==3 && j==10) break;
    }
    for(i=0;i<36;i++) timpi[i]=70;
    strcpy(buf,"10");
}

Succ :: ~Succ()
{
    delete listb1;
    delete listb2;
    //delete ob;
}

void Succ :: WMSetFocus(RTMessage)
{
    SetFocus(listb1->HWindow);
}

void Succ :: Curent()
{
    GetApplication()->ExecDialog(new
    TCurent(this,"TCURRENT"));
}

void Succ :: Timp()
{
    GetApplication()->ExecDialog(new
    TimpProba(this,"TIMP"));
}

void Succ :: Add()
{ char buf[60];

    wsprintf(buf,"%s Timp maxim proba :
    %d",tab[idx],timpi[idx]);
    if (listb2->AddString(buf) < 0)
        MessageBox(NULL,"Memorie
    insuficienta","Mesaj de eroare",MB_OK |
    MB_ICONEXCLAMATION);
}

void Succ :: Remove()
{
    int idx2 = listb2->GetSelIndex();
    listb2->DeleteString(idx2);
}

void Succ :: Save()
{
    GetApplication()->ExecDialog(new
    Salvare(this,32513,"*.scc" ));
}

void Succ :: List1(RTMessage Msg)
{
    if ( Msg.LP.Hi == LBN_SELCHANGE )
    {
        idx = listb1->GetSelIndex();
        len = listb1-
        >GetStringLen(idx);
        listb1-
        >GetString((LPSTR)tab[idx], idx);
        //MessageBox(NULL, tab[idx], "You
        selected:", MB_OK);
    }
}

TimpProba :: TimpProba(PtWindowsObject
    AParent, LPSTR Name)
    : TDialog(AParent , Name)
{
    timp = new TEdit(this,IDD_TIMPSCR,4);
    scrollbar = new
    TScrollBar(this,IDD_SCROLL);
}

void TimpProba :: SetupWindow()
{
    TDialog :: SetupWindow();
    index = ((Succ *)Parent)->idx;
    ttt = ((Succ *)Parent)-> timpi[index]; //
    timpi[i] sunt initializati cu 70 in Succ

    SetDlgItemInt(HWindow,IDD_TIMPSCR,ttt,FA
    LSE);
}

void TimpProba :: Casuta()
{
    char sir[4];

    SendDlgItemMsg(IDD_TIMPSCR,WM_GETTE
    XT,4,LPARAM(sir));
    ttt = atoi(sir);
    strcpy(ob->buf,sir);
}

void TimpProba :: Ok(RTMessage)
{
    index = ((Succ *)Parent)->idx;
    ((Succ *)Parent)-> timpi[index] = ttt;

    TDialog :: CloseWindow();
}

```

Asupra implementării echipamentelor cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii - ANEXA 6  
Program calculator PC pentru ERRORREX M

```

void TimpProba :: WMVScroll(RTMessage msg)
{
switch(msg.WParam)
{
case SB_LINEDOWN :
ttt--;
if(ttt<0) ttt=0;

SendDlgItemMsg(IDD_TIMPSCR,WM_SETTE
XT,0,LPARAM(inttostring(ttt)));
break;

case SB_LINEUP :
ttt++;

SendDlgItemMsg(IDD_TIMPSCR,WM_SETTE
XT,0,LPARAM(inttostring(ttt)));
break;

}
}

void Salvare :: Ok(RTMessage)
{
int hfl;
OFSTRUCT ofs;
char sir[12];
char path_name[50];
char slash[3] = "\\";
char mesaj[35] = "Fisierul exista ! Srietii peste ?";

SendDlgItemMsg(ID_FNAME,WM_GETTEXT,
12,LPARAM(sir));

SendDlgItemMsg(IDD_DIREC,WM_GETTEXT,
30,LPARAM(FilePath));
strcpy(path_name, FilePath);
strcat(path_name,slash ); strcat(path_name,sir);

hfl = OpenFile(path_name, &ofs, OF_EXIST);
if(hfl != -1)
{
if(MessageBox(NULL,mesaj,"Mesaj
de eroare",MB_YESNO) == IDNO)
return;
}

int nr_lista = (ob->listb2)->GetCount(); //nr de
elem. in lista

if(HFILE_ERROR != (hfl =
_lcreat(path_name,0)))
{
for(int i=0; i<nr_lista; i++)
{
char strin[60];
(ob->listb2)->GetString(strin,i);
strcat(strin,"n");
_lwrite(hfl,strin,strlen(strin));
}
char explic[200],buf[4];

SendDlgItemMsg(IDD_EXPLIC,WM_GETTEX
T,200,LPARAM(explic));
strcat(explic,"n");
strcpy(buf,ob->buf);
_lwrite(hfl,"Curentul
nominal:",strlen("Curentul nominal:"));
_lwrite(hfl,buf,strlen(buf));
_lwrite(hfl,"n",1);
_lwrite(hfl,explic,strlen(explic));
}
_lclose(hfl);
TDialog :: CloseWindow();
}

TCurent :: TCurent(PTWindowsObject AParent,
LPSTR Name)
: TDialog(AParent , Name)
{
curent = new TEdit(this,IDD_VALCURE,4);
}

void TCurent :: SetupWindow()
{
TDialog :: SetupWindow();
int cure = 10;

SetDlgItemInt(HWindow,IDD_VALCURE,cure,
FALSE);
}

void TCurent :: Ok(RTMessage)
{
SendDlgItemMsg(IDD_VALCURE,WM_GETT
EXT,4,LPARAM(ob->buf));
TDialog::CloseWindow();
}

/* Subrutina Testare */
*****
#define WIN31
#include <stdio.h>
#include <inputdia.h>
#include "contor.h"
#include "parola.h"
#include "testare.h"
#include "functii.h"
#include "comunic.h"

extern int com1,com2,com3,com4,com,no;
extern unsigned char post[1000];
extern char path[40];
extern HANDLE hInst;
extern char FileName1[50];
extern RECT r;

void TMenWindow::Acceler()

```

```

{
    memset(TxtParola1,0x00,sizeof TxtParola1);
    if( GetApplication()->ExecDialog(new
    IParola(this,"Parola1")==IDOK)
    {
        if(strcmp(TxtParola1,"aem")==0)
        {
            /*meni=GetMenu(HWindow);
            DestroyMenu(meni);
            meni=LoadMenu(hInst,"Menu2");
            SetMenu(HWindow,meni);
            SetCaption("Testare la Producator");*/
            GetApplication()->MakeWindow(new
            TFereastra( this,"Testare la producator" ) );
        }
        else
            MessageBox(HWindow,"Acces
            interzis !!!", "Parola gresita
            !",MB_OK|MB_ICONEXCLAMATION);
        } //if
        TxtParola1[0]='\0';
    }

TFereastra::TFereastra(PTWindowsObject
AParent, LPSTR ATitle)

TWindow(AParent,ATitle)
{
    AssignMenu("Menu2");
    Attr.Style=Attr.Style | WS_CAPTION |
    WS_SYSMENU| WS_POPUP
|WS_OVERLAPPEDWINDOW|WS_VISIBLE;
    Attr.X=5;
    Attr.Y=5;
    Attr.W=r.right-10;
    Attr.H=590;
}

Factor::Factor( PTWindowsObject Owner ,
LPSTR DName )
    :TDialog(Owner,DName)
{
    memset(&FA,0x0,sizeof(FA));
    sep= new
    TBRadioButton(this,IDD_SEP,NULL);
    ins= new TBRadioButton(this,IDD_INS,NULL);
    Const[0]= new TEdit(this,IDD_DIV,2);
    Const[1]= new TEdit(this,IDD_MUL,2);
    FA.ins = BF_CHECKED;
    strcpy(FA.div,"0");
    strcpy(FA.mul,"0");
    TransferBuffer=(void far *)&FA;
}

Releu::Releu( PTWindowsObject Owner ,
LPSTR DName )
    :TDialog(Owner,DName)
{
    memset(&RE,0x0,sizeof RE);
    on1= new
    TBRadioButton(this,IDD_ON1,NULL);
    off1= new
    TBRadioButton(this,IDD_OFF1,NULL);
    on2= new
    TBRadioButton(this,IDD_ON2,NULL);
    off2= new
    TBRadioButton(this,IDD_OFF2,NULL);
    on3= new
    TBRadioButton(this,IDD_ON3,NULL);
    off3= new
    TBRadioButton(this,IDD_OFF3,NULL);
    RE.off1 = BF_CHECKED;
    RE.off2 = BF_CHECKED;
    RE.off3 = BF_CHECKED;
    TransferBuffer=(void far *)&RE;
}

void TFereastra::fac_ies()
{
    GetApplication()->ExecDialog(new
    Factor(this,"FACTOR"));
}

void TFereastra::com_relee()
{
    unsigned char oct=0x00;
    int ok=1;

    if(GetApplication()->ExecDialog(new
    Releu(this,"RELEU"))==IDOK)
    {
        FILE *fis;
        fis=fopen("valid1.txt","w+");

        if(RE.off1==1) oct=oct|0x01; //conectat
        if(RE.off2==1) oct=oct|0x02;
        if(RE.off3==1) oct=oct|0x04;
        for(int k=0;k<3;k++)
        {
            if(com==com1)
            {
                transmitere(1,3,0xfe,0x1b,oct);
                receptie(1,0);
            }
            if(com==com2)
            {
                transmitere(2,3,0xfe,0x1b,oct);
                receptie(2,0);
            }
            if(com==com3)
            {
                transmitere(3,3,0xfe,0x1b,oct);
                receptie(3,0);
            }
            if(com==com4)
            {
                transmitere(4,3,0xfe,0x1b,oct);
                receptie(4,0);
            }
        }
    }
}

```

```

    }
    if(post[0]==0xfe && post[1]==0x1b
    && post[2]==oct)
        {ok=1;break;}
        else ok=0;

    if(ok==0)
    MessageBox(HWindow,"Eroare la
    interfata","Mesaj de eroare",MB_OK);
    }

    fprintf(fis,"%x %x %x
    ,oct=%x\n",post[0],post[1],post[2],oct);

    fprintf(fis,"on1:%d,on2:%d,on3:%d,off1=%d,off
    2=%d,off3=%d\n",RE.on1,RE.on2,RE.on3,RE.of
    f1,RE.off2,RE.off3);
    fclose(fis);
    } //if ok
    }

void TFereastra::mas_energ()
{
    GetApplication()->ExecDialog(new
    Energie(this,"ENERGIE"));
}

void TFereastra::ies_men()
{
    DestroyWindow(HWindow);
}

void Factor::Ok(RTMessage)
{
    unsigned char tt[2],oct;
    int ok=1;
    FILE *fis;

    SendDlgItemMsg(IDD_DIV,WM_GETTEXT,2,
    LPARAM(FA.div));

    SendDlgItemMsg(IDD_MUL,WM_GETTEXT,2,
    LPARAM(FA.mul));
    int nr1=atoi(FA.mul);
    int nr2=atoi(FA.div);

    if((nr1<0 || nr1>6) || (nr2<0 || nr2>7))
    { MessageBeep(0);
        MessageBox(HWindow,"Valoarea
        pentru divizare intre 0-7\nValoarea pentru
        multiplicare intre 0-6","Eroare de
        domeniu",MB_ICONEXCLAMATION|MB_OK)
        ;
        return;
    }
    TDialog::CloseWindow();
    tt[0]=nr1; //multipl
    tt[1]=nr2; //divizare
    oct=0x00;
}
    fis=fopen("valid.txt","w+");

    if (FA.sep==1)
    {
        for(int k=0;k<3;k++)
            {oct=(0x80|tt[0])|(tt[1]<<4);

                if(com==com1)
                {
                    transmite(1,3,0xfe,0x1a,oct);
                    receptie(1,0);
                }
                if(com==com2)
                {
                    transmite(2,3,0xfe,0x1a,oct);
                    receptie(2,0);
                }
                if(com==com3)
                {
                    transmite(3,3,0xfe,0x1a,oct);
                    receptie(3,0);
                }
                if(com==com4)
                {
                    transmite(4,3,0xfe,0x1a,oct);
                    receptie(4,0);
                }

                if(post[0]==0xfe && post[1]==0x1a &&
                post[2]==oct)
                    { ok=1;break;}
                    else ok=0;
                    }
                if(ok==0)
                MessageBox(HWindow,"Eroare de
                comunicare","Mesaj de
                eroare",MB_ICONEXCLAMATION|MB_OK);
                }
                else
                {
                    for(int k=0;k<3;k++)
                        {oct=tt[0]|(tt[1]<<4);

                            if(com==com1)
                            {
                                transmite(1,3,0xfe,0x1a,oct);
                                receptie(1,0);
                            }
                            if(com==com2)
                            {
                                transmite(2,3,0xfe,0x1a,oct);
                                receptie(2,0);
                            }
                            if(com==com3)
                            {
                                transmite(3,3,0xfe,0x1a,oct);
                                receptie(3,0);
                            }
                            if(com==com4)
                            {
                                transmite(4,3,0xfe,0x1a,oct);
                                receptie(4,0);
                            }
                        }
                    }
                }
    }
}

```

Asupra implementării echipamentelor cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii - ANEXA 6  
Program calculator PC pentru ERRORREX M

```

        if(post[0]==0xfe && post[1]==0x1a
&& post[2]==0x0c)
        { ok=1;break;}
        else ok=0;
        }
        if(ok==0)
        MessageBox(HWindow,"Eroare de
        comunicare", "Mesaj de
        eroare",MB_ICONEXCLAMATION|MB_OK);
        }

        fprintf(fis,"%x  %x
        %x\n",post[0],post[1],post[2]);
        fclose(fis);
        }

        {
        nn=nn1;
        MessageBeep(0);
        MessageBox(HWindow,"Interval ce
        iese din domeniu, sau eroare de
        introducere", "Eroare de
        introducere",MB_ICONHAND|MB_OK);
        strcpy(EN.mas,inttostring(nn));

        SendDlgItemMsg(IDD_MAS,WM_SETTEXT,0,
        (LPARAM)EN.mas);
        return;
        }

        nr=1;
        alarma=TRUE;
        Testare(nn);
        }

Energie::Energie( PTWindowsObject Owner ,
LPSTR DName )
        :TDialog(Owner,DName)
        {
        memset(&EN,0x0,sizeof EN);
        mas = new TEdit(this,IDD_MAS,3);
        Campuri[0]= new TStatic(this,IDD_FR,9);
        Campuri[1]= new TStatic(this,IDD_FS,9);
        Campuri[2]= new TStatic(this,IDD_FT,9);
        Campuri[3]= new TStatic(this,IDD_TOT,9);
        strcpy(EN.mas,"5");
        TransferBuffer=(void far *)&EN;
        nr=1;
        nn=5;
        }

void Energie::SetupWindow()
        {
        TDialog::SetupWindow();
        SetTimer(HWindow,IDD_T1,1000,NULL);
        alarma=TRUE;
        }

void Energie::WMTimer(RTMessage)
        {int j;

        if ((nr==1)&&(alarma==TRUE))
        Testare(nn);
        nr=nr+1;
        if ((nr==nn+1)|(nr>nn+1))
        nr=1;
        }

void Energie::IDReset(RTMessage)
        {
        int nn1;
        alarma=FALSE;
        nn1=nn;
        SendDlgItemMsg(IDD_MAS,WM_GETTEXT,3,
        (LPARAM)EN.mas);
        nn=atoi(EN.mas);
        if(nn<1 || nn>60 || !isdigit(nn))

void Energie::Testare(int nn)
        {
        /* gettimeofday(&t);
        sec=(int)t.ti_sec;
        do{
        gettimeofday(&t);
        sec1=(int)t.ti_sec;
        }while(abs(sec1-sec)<nn|| (sec1-
        sec<0&&60-sec+sec1<nn)); */
        SetCursor(NULL);
        FILE *fis;
        fis=fopen("valid3.txt","w+");
        jq=nn;

        if(com==com1)
        {
        transmitere(1,2,0xfe,0x1c);
        receptie(1,0);
        }
        if(com==com2)
        {
        transmitere(2,2,0xfe,0x1c);
        receptie(2,0);
        }
        if(com==com3)
        {
        transmitere(3,2,0xfe,0x1c);
        receptie(3,0);
        }
        if(com==com4)
        {
        transmitere(4,2,0xfe,0x1c);
        receptie(4,0);
        }

        octete1=((DWORD)post[4]*0xffff+(DWORD)po
        st[3]*0xff+(DWORD)post[2]);

        octete3=((DWORD)post[7]*0xffff+(DWORD)po
        st[6]*0xff+(DWORD)post[5]);
    
```

```

octete2=((DWORD)post[10]*0xffff+(DWORD)post[9]*0xff+(DWORD)post[8]);
octete4=octete1+octete2+octete3;

strcpy(tab1,inttostring(octete1));
strcpy(tab2,inttostring(octete2));
strcpy(tab3,inttostring(octete3));
strcpy(tab4,inttostring(octete4));

Campuri[0]->SetText(tab1);
Campuri[1]->SetText(tab2);
Campuri[2]->SetText(tab3);
Campuri[3]->SetText(tab4);

fprintf(fis,"%x %x %x %x %x %x %x %x %x %x\n",post[0],post[1],post[2],post[3],post[4],post[5],post[6],post[7],post[8],post[9],post[10],post[11]);
fprintf(fis,"%ld %ld %ld %ld\n",octete1,octete2,octete3,octete4);
fprintf(fis,"siruri: %s %s %s %s\n",tab1,tab2,tab3,tab4);
fclose(fis);
}

void Factor::valid_factor(RTMessage Msg,int indice)
{
char Txt[9],*p;

Const[indice]->GetText(Txt,9);

if (Msg.LP.Hi==EN_UPDATE)
{
for(p=Txt,*p&&(isdigit(*p));p++);
if(*p)
{ MessageBeep(0);
Const[indice]->SetText(val[indice]);
}
else
lstrcpy(val[indice],Txt);
}
}

void TFereastra::Achiz()
{
char buffer[4];
FILE *fis;

memset(buffer,'\0',sizeof buffer);
SetCursor(NULL);
if (GetApplication()->ExecDialog(new TInputDialog(this,"Introducere nr.post","Nr.post pentru achizitie: ",buffer,4))==IDOK)
{
int integ=atoi(buffer);
if(integ<1 || integ>255)
{ MessageBox(HWindow,"Nr.de post invalid","Mesaj de eroare",MB_OK|MB_ICONEXCLAMATION);
return;
}

if(com==com1)
{
transmitere(1,2,(char)integ,0x22);
receptie(1,4);
}
if(com==com2)
{
transmitere(2,2,(char)integ,0x22);
receptie(2,4);
}
if(com==com3)
{
transmitere(3,2,(char)integ,0x22);
receptie(3,4);
}
if(com==com4)
{
transmitere(4,2,(char)integ,0x22);
receptie(4,4);
}

fis=fopen("ach_test.tst","w+");
fprintf(fis,"\n %d %x %x %x\n",post[0],post[1],post[2],post[3]);
fclose(fis);
strcpy(fileName1,path);
strcat(fileName1,"\\ach_test.tst");
GetApplication()->MakeWindow(new afisare(this,fileName1));
}

void TFereastra::Invat()
{
char buffer[4];
FILE *fis;
int intr1,intr2;

memset(buffer,'\0',sizeof buffer);
SetCursor(NULL);
if(GetApplication()->ExecDialog(new TInputDialog(this,"Introducere nr.post","Nr.post existent pentru invatare: ",buffer,4))==IDOK)
{
intr1=atoi(buffer);
if(intr1<1 || (int)intr1>250 )
{ MessageBox(HWindow,"Nr.de post invalid","Mesaj de eroare",MB_OK|MB_ICONEXCLAMATION);
return;
}
}
}

```

Asupra implementării echipamentelor cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii - ANEXA 6  
Program calculator PC pentru ERRORREX M

```

if(GetApplication()->ExecDialog(new
TInputDialog(this,"Introducere nr.post", "Nr.post
nou pentru invatare: ",buffer,4))==IDOK)
{
    intr2=atoi(buffer);

    if(com==com1)
    {
        transmitere(1,3,(char)intr1,0x11,(char)intr2);
        receptie(1,2);
    }
    if(com==com2)
    {
        transmitere(2,3,(char)intr1,0x11,(char)intr2);
        receptie(2,2);
    }
    if(com==com3)
    {
        transmitere(3,3,(char)intr1,0x11,(char)intr2);
        receptie(3,2);
    }
    if(com==com4)
    {
        transmitere(4,3,(char)intr1,0x11,(char)intr2);
        receptie(4,2);
    }

    fis=fopen("inv_test.tst","w+");
    fprintf(fis,"n %d %x\n",post[0],post[1]);
    fclose(fis);
    strcpy(fileName1,path);
    strcat(fileName1,"\\inv_test.tst");
    GetApplication()->MakeWindow(new
afisare(this,fileName1));
}
}

void TFereastra::Constanta()
{
    char buffer[4],jq[2],tab[6];
    FILE *fis;

    memset(buffer,'\0',sizeof buffer);
    memset(tab,'\0',sizeof tab);
    SetCursor(NULL);
    if(GetApplication()->ExecDialog(new
TInputDialog(this,"Introducere nr.post", "Nr.post
pentru constanta: ",buffer,4))==IDOK)
    {
        int intreg=atoi(buffer);
        if(intreg<1 || intreg>255 )
            { MessageBox(HWindow,"Nr.de post
invalid", "Mesaj de
eroare",MB_OK|MB_ICONEXCLAMATION);
        }
    }

    return;
}

if(GetApplication()->ExecDialog(new
TInputDialog(this,"Introducere constanta
contor", "Introduceti constanta:",tab,6))==IDOK)
{
    unsigned int nr= atoi(tab);
    jq[0]=nr/256;
    jq[1]=nr-256*jq[0];
    if(com==com1)
    {
        transmitere(1,4,(char)intreg,0x33,jq[1],jq[0]);
        receptie(1,4);
    }
    if(com==com2)
    {
        transmitere(2,4,(char)intreg,0x33,jq[1],jq[0]);
        receptie(2,4);
    }
    if(com==com3)
    {
        transmitere(3,4,(char)intreg,0x33,jq[1],jq[0]);
        receptie(3,4);
    }
    if(com==com4)
    {
        transmitere(4,4,(char)intreg,0x33,jq[1],jq[0]);
        receptie(4,4);
    }

    fis=fopen("con_test.tst","w+");
    fprintf(fis,"n %d %x %x %x
%x ",post[0],post[1],post[2],post[3]);
    fclose(fis);
    strcpy(fileName1,path);

    strcat(fileName1,"\\con_test.tst");
    GetApplication()-
>MakeWindow(new afisare(this,fileName1));
}
}

void TFereastra::Auto_test()
{
    char buffer[4];
    FILE *fis;

    memset(buffer,'\0',sizeof buffer);
    SetCursor(NULL);
    if (GetApplication()->ExecDialog(new
TInputDialog(this,"Introducere nr.post", "Nr.post
pentru autotest: ",buffer,4))==IDOK)

```

```

{
    int intreg=atoi(buffer);
    if(intreg<1 || intreg>255 )
        { MessageBox(HWindow,"Nr. de post
invalid","Mesaj de
eroare",MB_OK|MB_ICONEXCLAMATION);
        return;
        }

    if(com==com1)
        transmitere(1,2,(char)intreg,0x44);
    if(com==com2)
        transmitere(2,2,(char)intreg,0x44);
    if(com==com3)
        transmitere(3,2,(char)intreg,0x44);
    if(com==com4)
        transmitere(4,2,(char)intreg,0x44);
}

void TFereastra::Ture()
{
    char buffer[4];
    FILE *fis;
    int intr1,intr2;

    memset(buffer,'\0',sizeof buffer);
    SetCursor(NULL);
    if(GetApplication()->ExecDialog(new
TInputDialog(this,"Introducere nr. post","Nr. post
pentru ture contor: ",buffer,4))==IDOK)
    {
        intr1=atoi(buffer);
        if(intr1<1 || intr1>255 )
            { MessageBox(HWindow,"Nr. de post
invalid","Mesaj de
eroare",MB_OK|MB_ICONEXCLAMATION);
            return;
            }
        buffer[0]='\0';
        if(GetApplication()->ExecDialog(new
TInputDialog(this,"Introducere nr. ture","Nr. ture
pentru contor : ",buffer,4))==IDOK)
        {
            intr2=atoi(buffer);
            if(intr2<1 || intr2>255)
                {
                    MessageBox(HWindow,"Nr. de ture trebuie sa fie
intre 1-255","Mesaj de
eroare",MB_OK|MB_ICONEXCLAMATION);
                    return;
                }
        }

        if(com==com1)
            {
                transmitere(1,3,(char)intr1,0x66,(char)intr2);
                receptie(1,4);
            }
    }

    if(com==com2)
        {
            transmitere(2,3,(char)intr1,0x66,(char)intr2);
            receptie(2,4);
        }
    if(com==com3)
        {
            transmitere(3,3,(char)intr1,0x66,(char)intr2);
            receptie(3,4);
        }
    if(com==com4)
        {
            transmitere(4,3,(char)intr1,0x66,(char)intr2);
            receptie(4,4);
        }

    fis=fopen("ture.tst","w+");
    fprintf(fis,"n %d %x
%d\n",post[0],post[1],post[2]);
    fclose(fis);
    strcpy(fileName,path);
    strcat(fileName,"\\ture.tst");
    GetApplication()->MakeWindow(new
afisare(this,fileName));
}

void TFereastra :: Filtru()
{
    char buffer[4];
    FILE *fis;
    int intr1,intr2;

    memset(buffer,'\0',sizeof buffer);
    SetCursor(NULL);
    if(GetApplication()->ExecDialog(new
TInputDialog(this,"Introducere nr. post","Nr. post
pentru filtru: ",buffer,4))==IDOK)
    {
        intr1=atoi(buffer);
        if(intr1<1 || intr1>255 )
            { MessageBox(HWindow,"Nr. de post
invalid !","Mesaj de
eroare",MB_OK|MB_ICONEXCLAMATION);
            return;
            }
        buffer[0]='\0';
        if(GetApplication()->ExecDialog(new
TInputDialog(this,"Introducere
nr. milisecunde","Nr. milisecunde pentru contor :
",buffer,3))==IDOK)
        {
            intr2=atoi(buffer);
            if(intr2<0 || intr2>10)
                {
                    MessageBox(HWindow,"Nr. de milisecunde

```



```

trebuie sa fie intre 0-10 !", "Mesaj de
eroare", MB_OK|MB_ICONEXCLAMATION);
        return;
    }

    char flt=(int)((float)intr2/10)*255);

    if(com==com1)
    {
        transmitere(1,3,(char)intr1,0x77,flt);
        receptie(1,4);
    }
    if(com==com2)
    {
        transmitere(2,3,(char)intr1,0x77,flt);
        receptie(2,4);
    }
    if(com==com3)
    {
        transmitere(3,3,(char)intr1,0x77,flt);
        receptie(3,4);
    }
    if(com==com4)
    {
        transmitere(4,3,(char)intr1,0x77,flt);
        receptie(4,4);
    }

    fis=fopen("filtru.tst","w+");
    fprintf(fis,"n %d %x
%d\n",post[0],post[1],post[2]);
    fclose(fis);
    strcpy(fileName1,path);
    strcat(fileName1,"\\filtru.tst");
    GetApplication()->MakeWindow(new
afisare(this,fileName1));
}
}
}

void TFereastra :: Marci()
{
    char buffer[4];
    FILE *fis;
    char intr1,intr2;

    memset(buffer,'\0',sizeof buffer);
    SetCursor(NULL);
    if(GetApplication()->ExecDialog(new
TInputDialog(this,"Introducere nr.post", "Nr.post
pentru numarare marci: ",buffer,4))==IDOK)
    {
        intr1=atoi(buffer);
        if(intr1<1 || intr1>no )
            { MessageBox(HWindow,"Nr.de post
invalid", "Mesaj de
eroare", MB_OK|MB_ICONEXCLAMATION);
                return;
            }
    }

    if(com==com1)
    {
        transmitere(1,2,intr1,0x88);
        receptie(1,4);
    }
    if(com==com2)
    {
        transmitere(2,2,intr1,0x88);
        receptie(2,4);
    }
    if(com==com3)
    {
        transmitere(3,2,intr1,0x88);
        receptie(3,4);
    }
    if(com==com4)
    {
        transmitere(4,2,intr1,0x88);
        receptie(4,4);
    }

    fis=fopen("marci.tst","w+");
    fprintf(fis,"n %d %x
%d\n",post[0],post[1],post[2]);
    fclose(fis);
    strcpy(fileName1,path);
    strcat(fileName1,"\\marci.tst");
    GetApplication()->MakeWindow(new
afisare(this,fileName1));
}
}

void TFereastra :: Monitor()
{
}

/* Subrutina Verifcon */
*****
#define WIN31
#include <inputdia.h>
#include <math.h>
#include <stdio.h>
#include "contor.h"
#include "verifcon.h"
#include "functii.h"
#include "mydiag.h"

extern int com1,com2,com3,com4,com;
extern int mult;
extern char name[40],indice;
extern char path1[40],path2[40];
extern RECT r;
long nt;
extern HWND hwn;
extern unsigned char post[100];
extern float K_et;
extern CONF Config;

```

Asupra implementării echipamentelor cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii - ANEXA 6

Program calculator PC pentru ERRORREX M

```

void TMenWindow::Verif_const()
{
    char buf[6]="1";
    char bufer1[6],bufer2[6];
    OFSTRUCT ofs;

    if(GetApplication()->ExecDialog(new
    TInputDialog(this,"Verificare constanta
    contor","Introduceti nr. de KW contorizati
    :",buf,6))==IDOK)
    {
        int cons =
        GetPrivateProfileInt("Constanta","Post1",6,path1
        );
        int pos =
        GetPrivateProfileInt("Ture_disc","Post1",6,path2
        );

        if(cons==0 ||pos==0 ||
        OpenFile(path1,&ofs,OF_EXIST)==-1 ||
        OpenFile(path2,&ofs,OF_EXIST)==-1)
        {
            MessageBeep(MB_ICONHAND);
            MessageBox(NULL,"Nu ati introdus
            constantele si turele pentru contori","Mesaj de
            atentionare",MB_OK|MB_ICONEXCLAMATIO
            N);
            return;
        }
        ProbaCurenta(); //de aici iau indice si
        mult
        nt = pow(10,3)*K_et*atol(buf);
        if (
        GetPrivateProfileInt("Configurare","Transfur",0,
        name)==1)
        {
            GetPrivateProfileString("Domenii","Curent","",b
            ufer1,6,"proba.ini");
            nt*=atof(Config.CurentSec)/atof(bufer1);
        }
        if (
        GetPrivateProfileInt("Configurare","Transften",0,
        name)==1)
        {
            GetPrivateProfileString("Domenii","Tensiune","",
            bufer2,6,"proba.ini");
            nt*=atof(Config.TensiuneSec)/atof(bufer2);
        }
        if(mult==0) nt*= pow(2,indice);
        //multiplicare
        if(mult==1) nt*= (float)1/pow(2,indice-
        8); //divizare
        MessageBox(HWindow,"Notati indexul
        contoarelor si conectati statia
        !","Avertisment",MB_OK|MB_ICONHAND);
        int panou =
        GetPrivateProfileInt("Configurare","Tipachiz",0,
        name);
        if(panou == 1)
            GetApplication()-
            >MakeWindow(new Procent( this,"Evidenta
            procente la faze" ) , //cele 3 faze cu cele 3 bare
            else
            GetApplication()-
            >MakeWindow(new Procent1(this,"Verificare
            mecanism registrator")); //o bara cu suma
            celor 3faze
            }
            }

void Procent1::Procent1(PtWindowsObject
    AParent, LPSTR ATitle)
    :TWindow(AParent,ATitle)
    {
        Attr.Style= Attr.Style| WS_POPUP|
        WS_CAPTION| WS_OVERLAPPEDWINDOW
        ;
        Attr.X=r.right/4;
        Attr.Y=r.bottom/2-80;
        Attr.W=r.right/3+20;
        Attr.H=190;
        EnableKBHandler();
        new
        TButton(this,IDCANCEL2,"Cancel",r.right/6-
        30,100,70,30,TRUE);
        new TStatic(this,-1,"Energie :",10,30,70,25,11);
        flag=0;
    }

void Procent1 :: SetupWindow()
{
    TWindow::SetupWindow();
    afisez_procent();
}

void Procent1 :: afisez_procent()
{
    struct time t;
    int sec,sec1;
    DWORD S=0;
    HDC hdc;
    int ok=0;
    HBRUSH hbr,hb1,hbroid;

    ShowWindow (HWindow,SW_SHOW);
    UpdateWindow(HWindow);
    hdc=GetDC(HWindow);
    Rectangle(hdc,89,24,r.right/3-29,56);

    hbr=CreateSolidBrush(RGB(0,0,255));
    hb1=CreateSolidBrush(RGB(255,0,0));
    hbroid=SelectObject(hdc,hbr);
    SetCursor(NULL);

    do{
        if (flag==1) break;
        if(com==com1)
        {
            transmitere(1,2,0xfc,0x1c);
        }
    }
}

```

Asupra implementării echipamentelor cu logică programată pentru comanda unei stații de etalonat contoare de energie electrică cu achiziția erorii - ANEXA 6  
Program calculator PC pentru ERRORREX M

```

receptie(1,0);
}
if(com==com2)
{
    transmitere(2,2,0xfe,0x1c);
    receptie(2,0);
}
if(com==com3)
{
    transmitere(3,2,0xfe,0x1c);
    receptie(3,0);
}
if(com==com4)
{
    transmitere(4,2,0xfe,0x1c);
    receptie(4,0); //am
    receptionat octetii pentru calcule de la R,S,T
}

S=S+((DWORD)post[4]*0xffff+(DWORD)post[3]*0xff+(DWORD)post[2]);

S=S+((DWORD)post[7]*0xffff+(DWORD)post[6]*0xff+(DWORD)post[5]);

S=S+((DWORD)post[10]*0xffff+(DWORD)post[9]*0xff+(DWORD)post[8]);

if(ok==0)
{ S=0; //la
inceput vin date eronate care nu le iau in calcul
ok=1; }

if(S>=nt)
{
    S=0;
    TextOut(hdc,180,60,"100%",strlen("100%"));
    SelectObject(hdc,hb1);
    Rectangle(hdc,90,25,r.right/3-30,55);
    SelectObject(hdc,hbr);
    if(com==com1)
    {
        transmitere(1,3,0xfe,0x1b,0x00); // aprind
        ledurile (conectat)
        receptie(1,0);
    }
    if(com==com2)
    {
        transmitere(2,3,0xfe,0x1b,0x00);
        receptie(2,0);
    }
    if(com==com3)
    {
        transmitere(3,3,0xfe,0x1b,0x00);
    }
}
receptie(3,0);
}
if(com==com4)
{
    if(MessageBox(HWindow,"Proba terminata
    !", "",MB_OK)==IDOK) ICanc();
    break;
} //if
int prok =
100*S/nt;
float val =
((float)prok/100);
val*=(r.right/3-
120);
Rectangle(hdc,90,25,90+val,55);
char buf[11]="\0";
strcpy(buf,inttostring(prok));
strcat(buf," % ");
TextOut(hdc,180,60,buf,strlen(buf));
if (flag==1) break;
MSG msg;
while(PeekMessage(&msg,hwn,0,0,PM_REMOVE))
if(!IsDialogMessage(hwn,&msg))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
if (flag==1) break;
gettime(&t);
sec=(int)t.ti_sec;
do{
    gettime(&t);
    sec1=(int)t.ti_sec;
    }while(abs(sec1-
sec)<2|| (sec1-sec<0 && 60-sec+sec1<2));
}while(S<nt);
SelectObject(hdc,hbroid);
DeleteObject(hbr);
SelectObject(hdc,hbroid);
DeleteObject(hbr);

```

```

DeleteObject(hb1);
ReleaseDC(hwn,hdc);
}

void Procent1::ICanc()
{
flag=1;
if(com==com1)
{
transmitere(1,3,0xfe,0x1b,0x07); // pun fazele
pe off
receptie(1,0);
}
if(com==com2)
{
transmitere(2,3,0xfe,0x1b,0x07);
receptie(2,0);
}
if(com==com3)
{
transmitere(3,3,0xfe,0x1b,0x07);
receptie(3,0);
}
if(com==com4)
{
transmitere(4,3,0xfe,0x1b,0x07);
receptie(4,0);
}
DestroyWindow(HWindow);
}

void Procent::afisez_procente()
{
DWORD SR=0,SS=0,ST=0;
struct time t;
int sec,sec1;
int ok=0,ok1=0,ok2=0,ok3=0;
unsigned char ox=0x07;
HDC hdc;
HBRUSH hbr,hbroid,hb1,hb2;
HPEN hpen,hpenOld;

ShowWindow
(HWindow,SW_SHOW);
UpdateWindow(HWindow);
hdc=GetDC(HWindow);
Rectangle(hdc,99,29,301,51);
Rectangle(hdc,99,99,301,121);
Rectangle(hdc,99,169,301,191);

hpen = CreatePen(PS_SOLID, 1,
RGB(0, 255, 255));
hpenOld = SelectObject(hdc, hpen);

hbr=CreateSolidBrush(RGB(0,255,255)
);
hb1=CreateSolidBrush(RGB(255,0,0));
hbroid=SelectObject(hdc,hbr);
SetCursor(NULL);

do{
if(flag==1) break;
MSG msg;

while(PeekMessage(&msg,hwn,0,0,PM_REMOVE))
if(!IsDialogMessage(hwn,&msg))
{
TranslateMessage(&msg);
DispatchMessage(&msg);
}
if(flag==1) break;

if(com==com1)
{
transmitere(1,2,0xfe,0x1c);
receptie(1,0);
}
if(com==com2)
{
transmitere(2,2,0xfe,0x1c);
receptie(2,0);
}
if(com==com3)
{
transmitere(3,2,0xfe,0x1c);
}
}

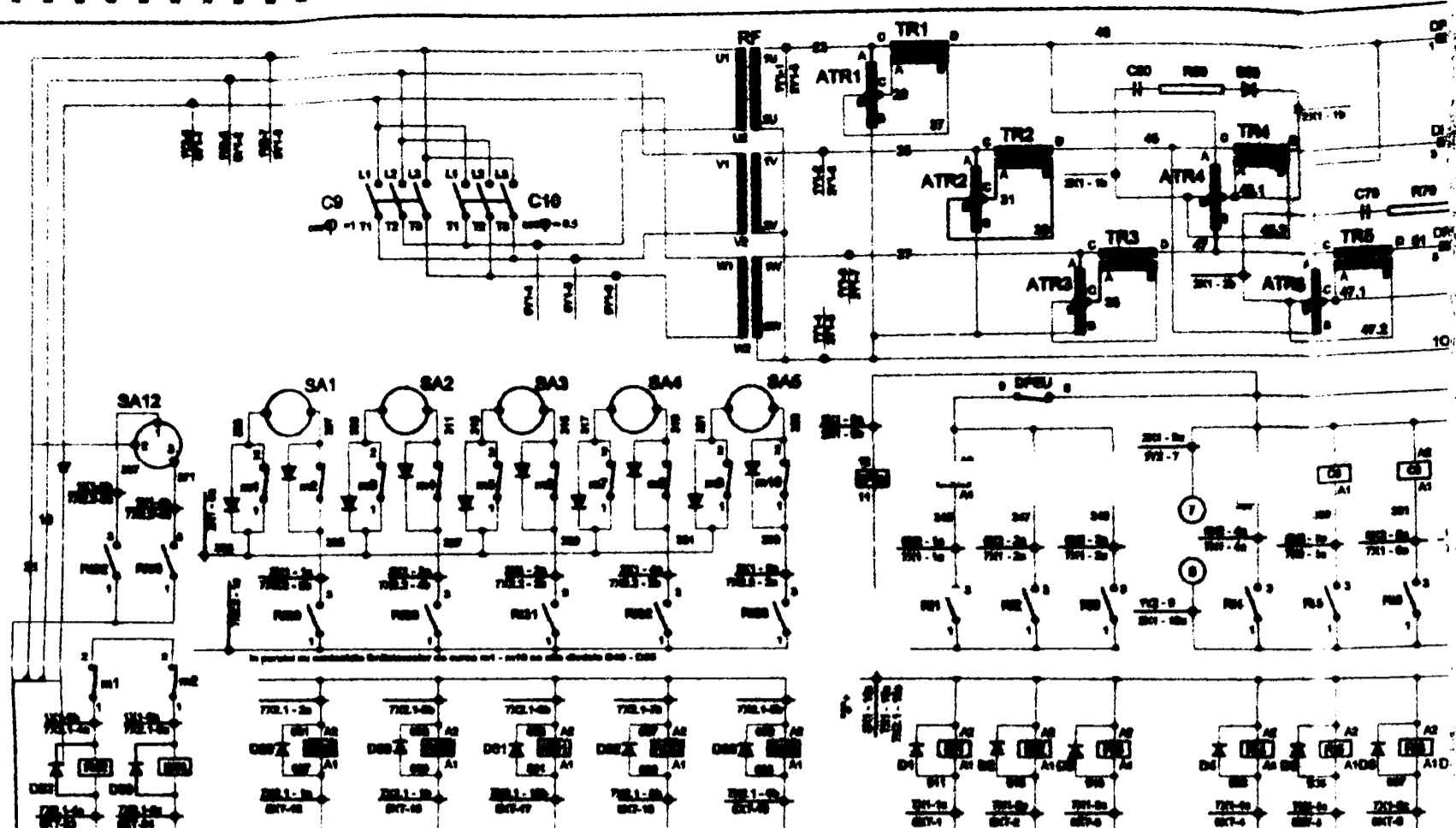
void Procent::Procent(PtWindowsObject
AParent, LPSTR ATitle)
:TWindow(AParent,ATitle)
{
Attr.Style= Attr.Style| WS_POPUP|
WS_CAPTION| WS_OVERLAPPEDWINDOW
;
Attr.X=r.right/4;
Attr.Y=r.bottom/2-80;
Attr.W=r.right/2;
Attr.H=300;
EnableKBHandler();
flag=0;
new
TButton(this,IDCANCEL2,"Cancel",r.right/4-
40,230,70,30,TRUE);
new TStatic(this,-1,"Faza R :",10,30,60,25,10);
new TStatic(this,-1,"Faza S :",10,100,60,25,10);
new TStatic(this,-1,"Faza T :",10,170,60,25,10);
}

void Procent :: SetupWindow()
{
TWindow::SetupWindow();
afisez_procente();
}

void Procent :: ~Procent()
{
}

```





REACTANZA CAPACITIVA		CONDICIA DE TRAZO DE FASE					PROTECCION ELECTROMAGNETICA EN CASO DE TEMPERATURA			SELECCION DE TENSIONES		
CAPACITIV	INDUCTIV	FAZA R	FAZA S	FAZA T	FAZA B	FAZA T	0.5A	1A	1.5A	277V	230V	135V

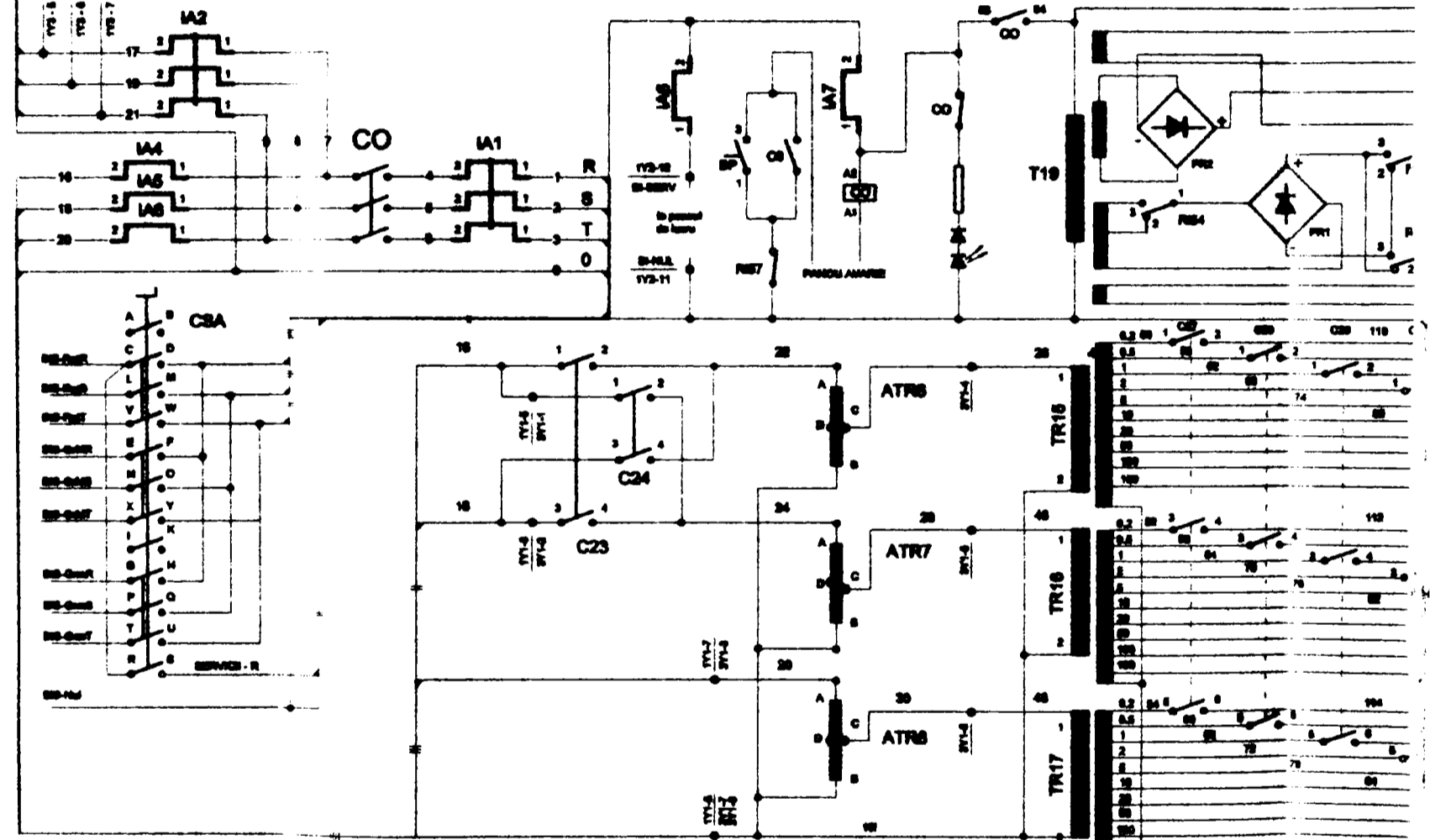
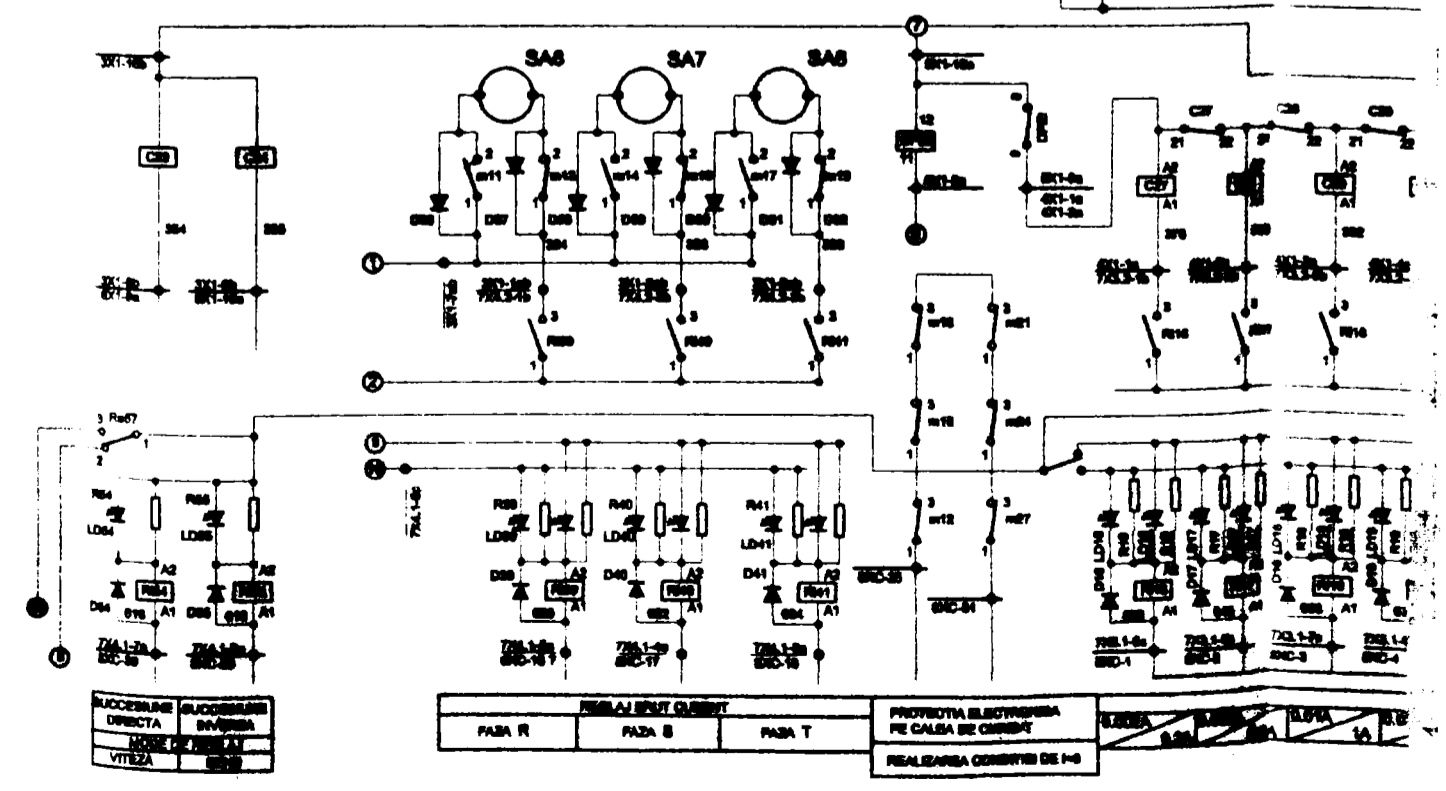
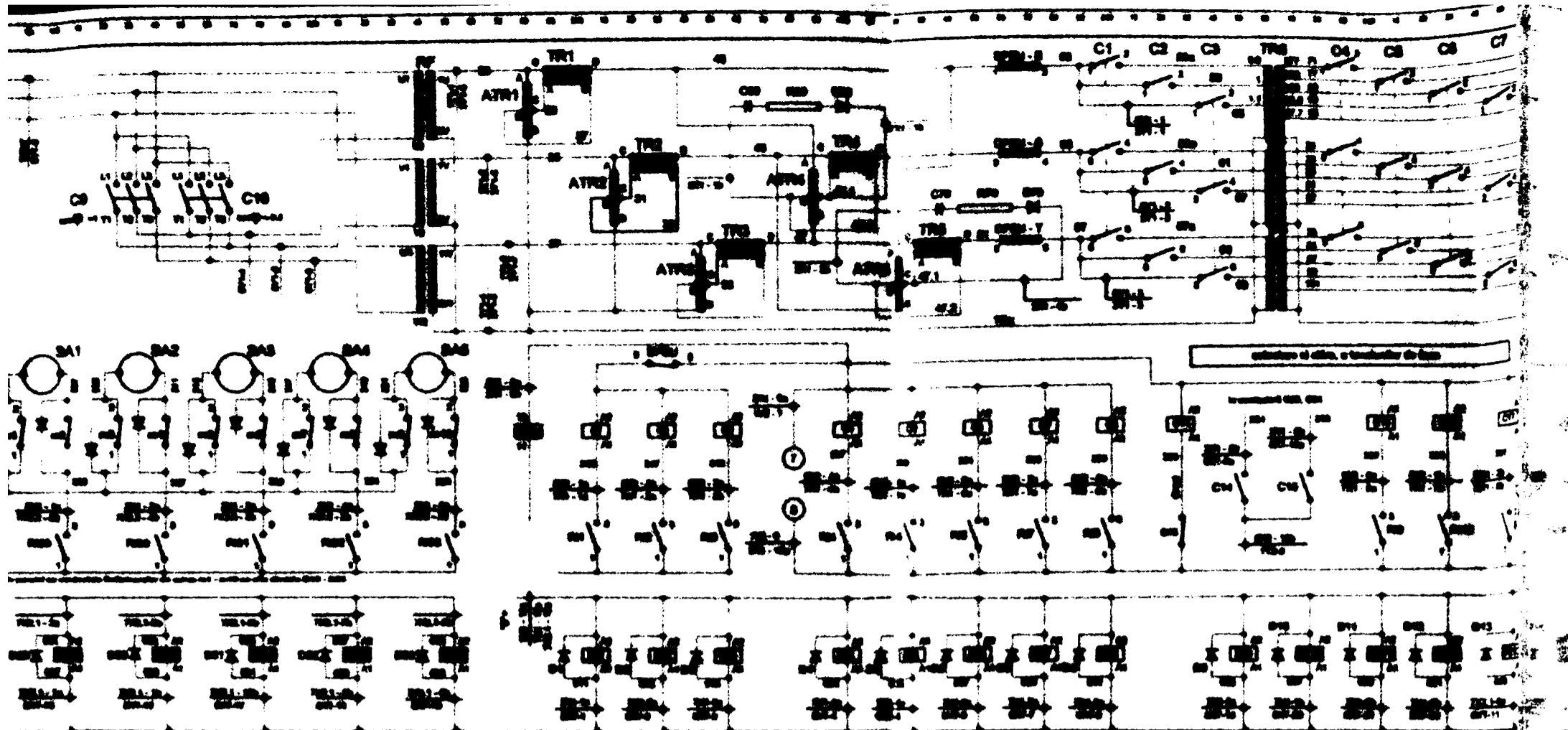


Diagrama CSA

	1	2	3	4
AB	X			
CB		X		
CA			X	
BA				X
LA	X			
LB		X		
MB			X	
PA				X
TA				X
VA	X			
XA				X

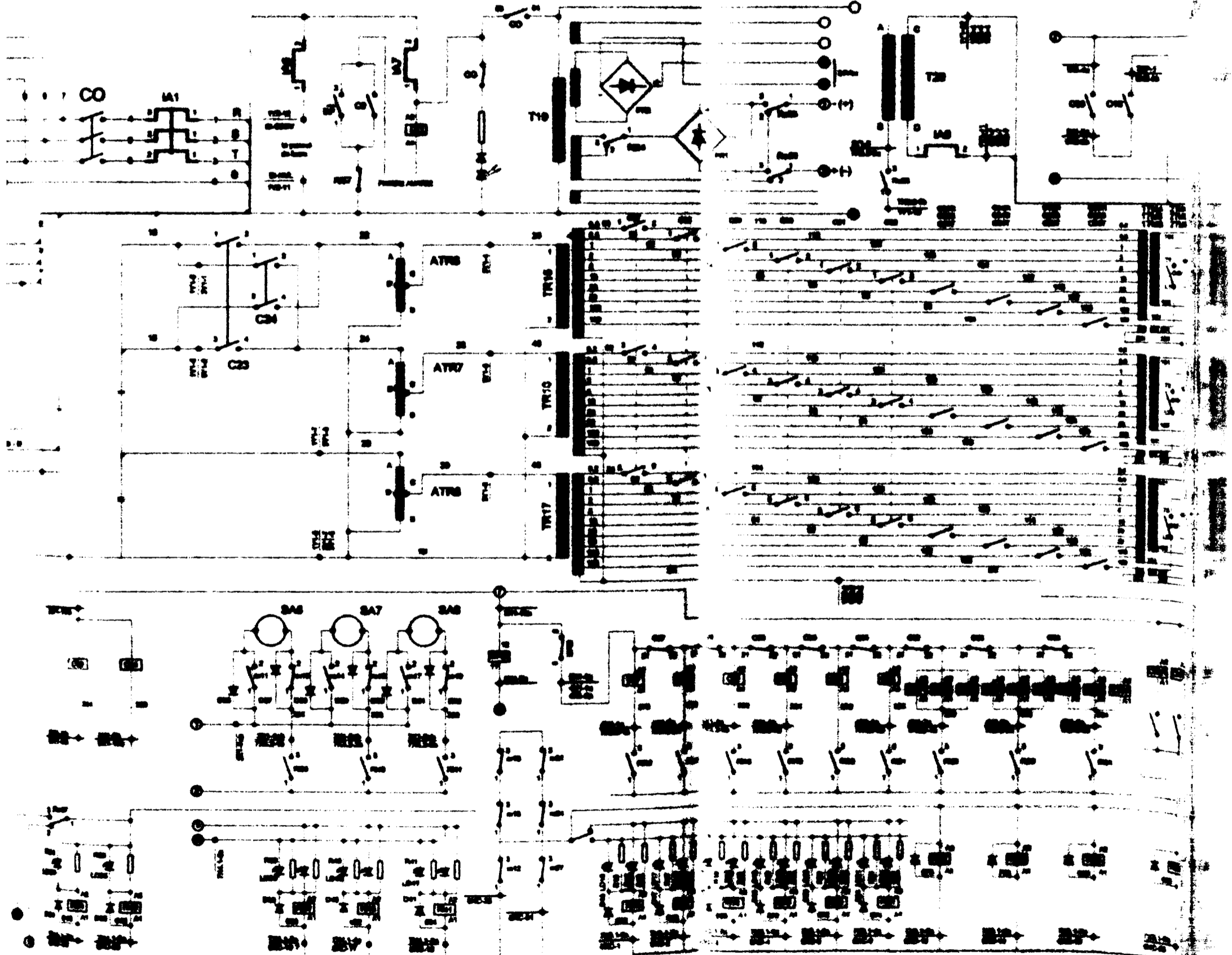


ACCESORIOS DIRECTA		REACTANZA INDUCTIVA			PROTECCION ELECTROMAGNETICA EN CASO DE CORRIENTE			SELECCION DE TENSIONES				
DIRECTA	INDIRECTA	INDUCTIVA	CAPACITIVA	FAZA R	FAZA S	FAZA T	0.5A	1A	1.5A	277V	230V	135V



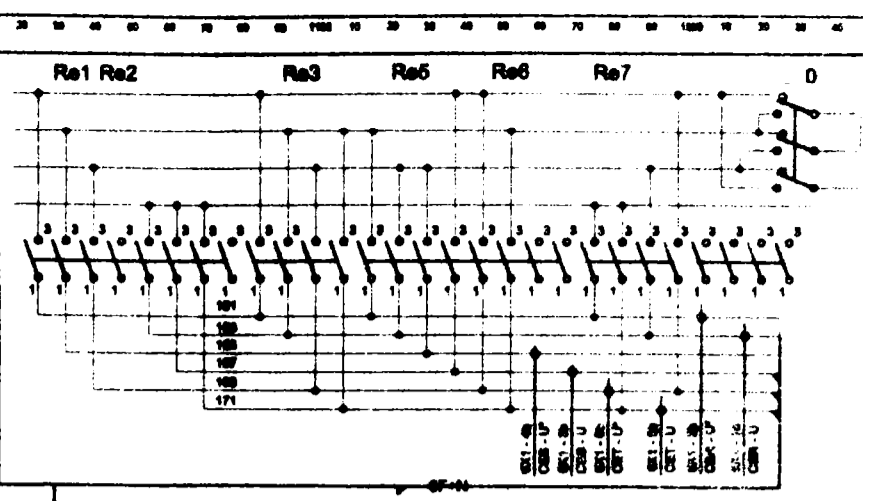
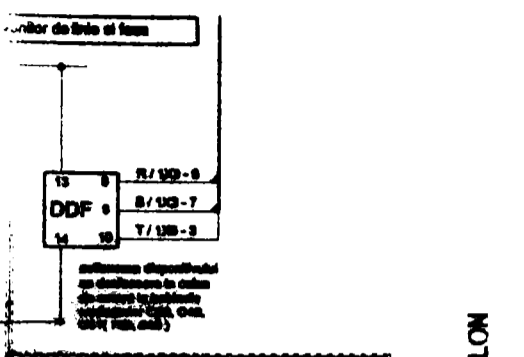
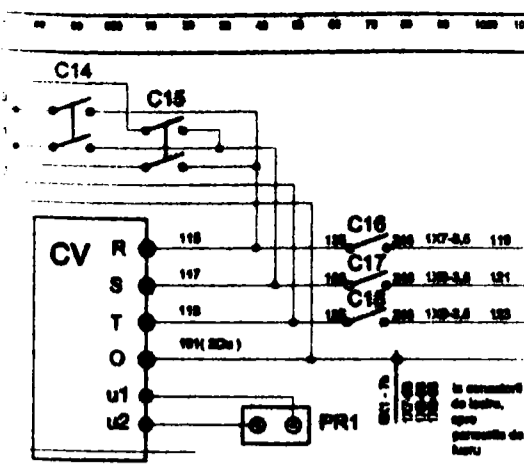
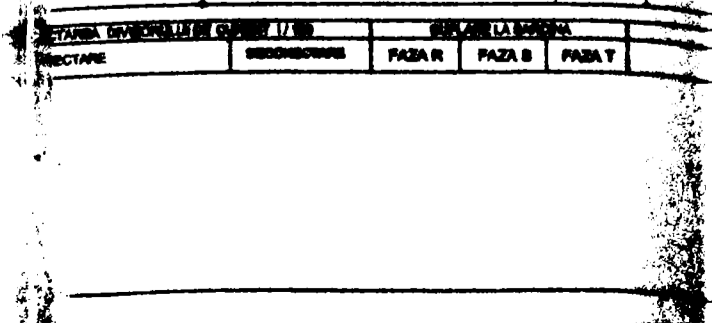
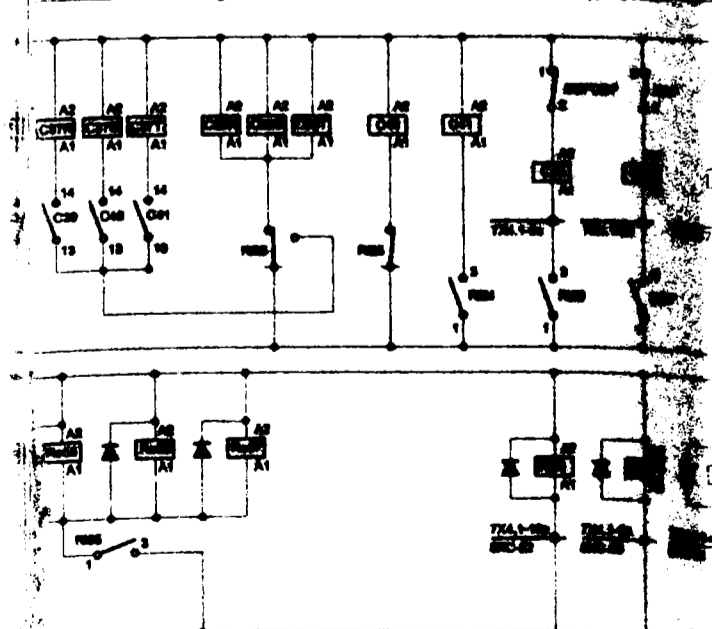
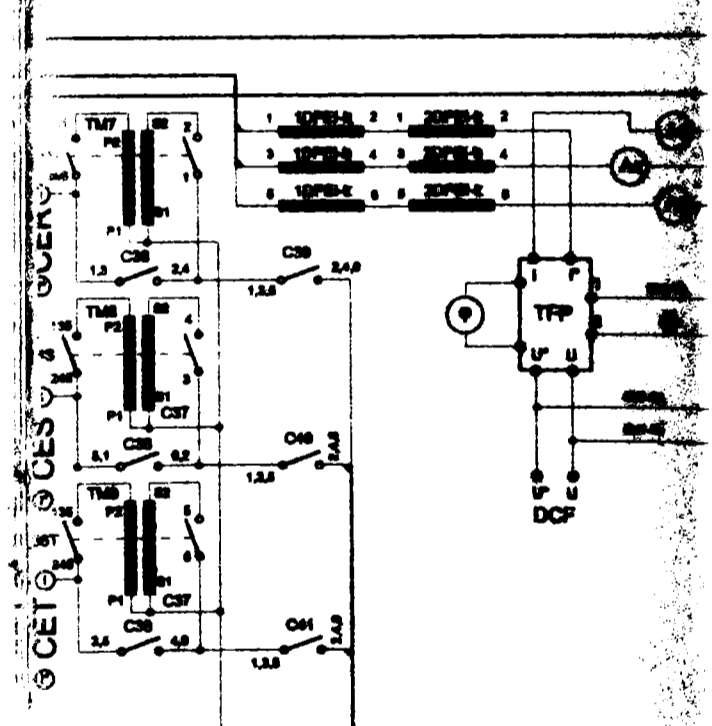
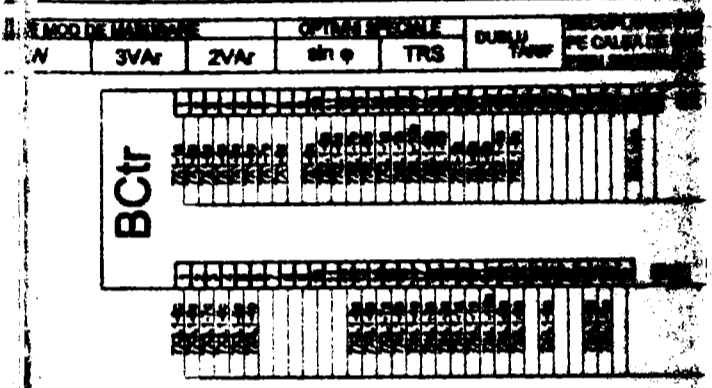
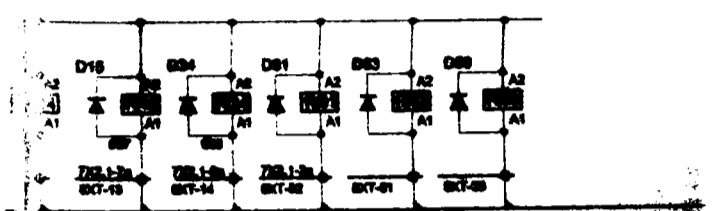
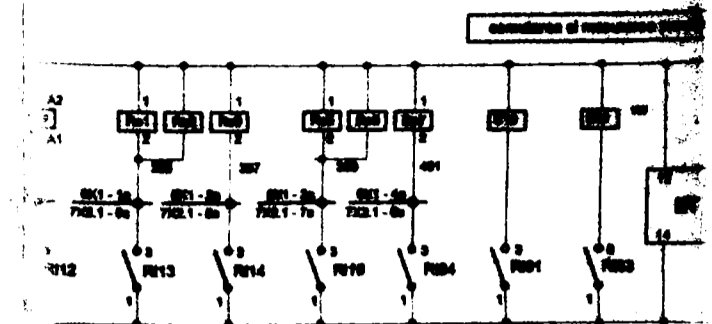
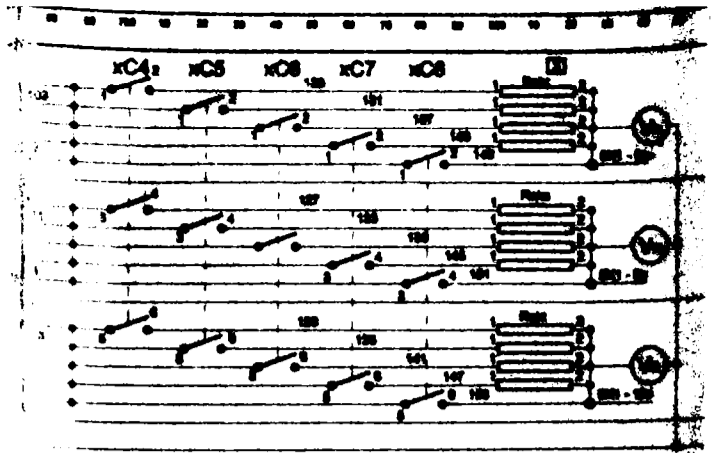
PAGE 2	PAGE 3	PAGE 7	PAGE 8	PAGE 9	PAGE 10	PAGE 11	PAGE 12	PAGE 13	PAGE 14	PAGE 15	PAGE 16	PAGE 17	PAGE 18	PAGE 19	PAGE 20	PAGE 21	PAGE 22	PAGE 23	PAGE 24	PAGE 25	PAGE 26	PAGE 27	PAGE 28	PAGE 29	PAGE 30
--------	--------	--------	--------	--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

PAGE 31	PAGE 32	PAGE 33	PAGE 34	PAGE 35	PAGE 36	PAGE 37	PAGE 38	PAGE 39	PAGE 40	PAGE 41	PAGE 42	PAGE 43	PAGE 44	PAGE 45	PAGE 46	PAGE 47	PAGE 48	PAGE 49	PAGE 50	PAGE 51	PAGE 52	PAGE 53	PAGE 54	PAGE 55	PAGE 56	PAGE 57	PAGE 58	PAGE 59	PAGE 60
---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

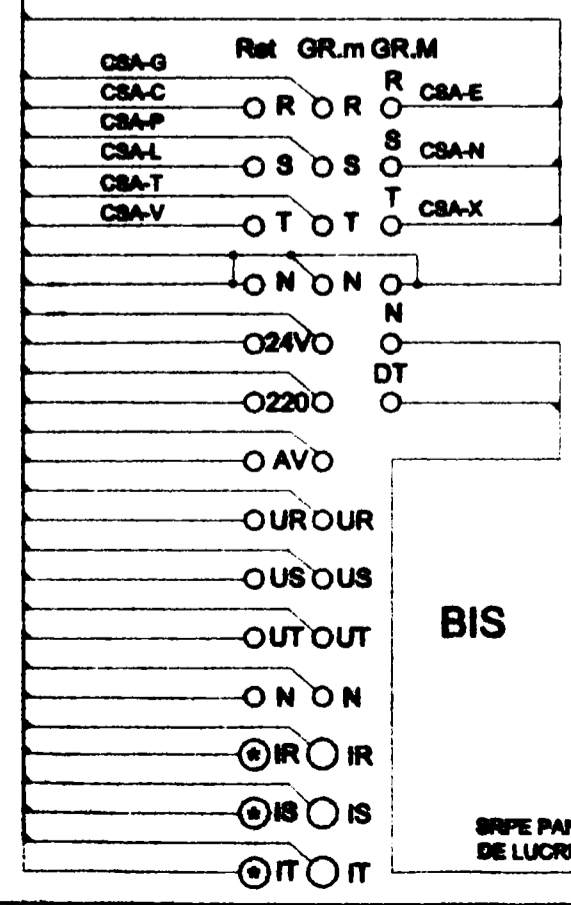
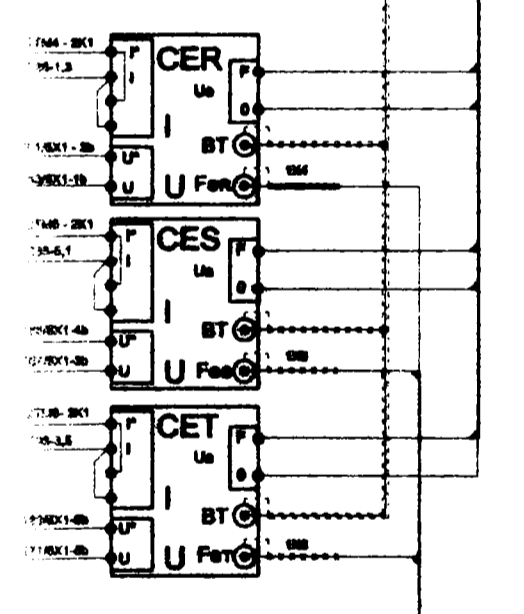
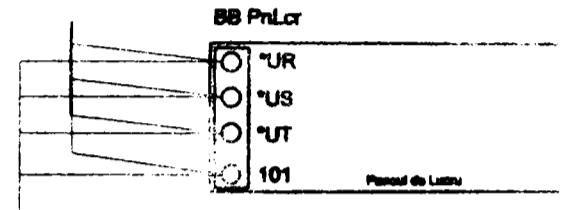


PAGE 2	PAGE 3	PAGE 7	PAGE 8	PAGE 9	PAGE 10	PAGE 11	PAGE 12	PAGE 13	PAGE 14	PAGE 15	PAGE 16	PAGE 17	PAGE 18	PAGE 19	PAGE 20	PAGE 21	PAGE 22	PAGE 23	PAGE 24	PAGE 25	PAGE 26	PAGE 27	PAGE 28	PAGE 29	PAGE 30	PAGE 31	PAGE 32	PAGE 33	PAGE 34	PAGE 35	PAGE 36	PAGE 37	PAGE 38	PAGE 39	PAGE 40	PAGE 41	PAGE 42	PAGE 43	PAGE 44	PAGE 45	PAGE 46	PAGE 47	PAGE 48	PAGE 49	PAGE 50	PAGE 51	PAGE 52	PAGE 53	PAGE 54	PAGE 55	PAGE 56	PAGE 57	PAGE 58	PAGE 59	PAGE 60
--------	--------	--------	--------	--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

PAGE 61	PAGE 62	PAGE 63	PAGE 64	PAGE 65	PAGE 66	PAGE 67	PAGE 68	PAGE 69	PAGE 70	PAGE 71	PAGE 72	PAGE 73	PAGE 74	PAGE 75	PAGE 76	PAGE 77	PAGE 78	PAGE 79	PAGE 80	PAGE 81	PAGE 82	PAGE 83	PAGE 84	PAGE 85	PAGE 86	PAGE 87	PAGE 88	PAGE 89	PAGE 90	PAGE 91	PAGE 92	PAGE 93	PAGE 94	PAGE 95	PAGE 96	PAGE 97	PAGE 98	PAGE 99	PAGE 100
---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	----------



TENSIUNEA DE SERVICIU CONTETALON



BIS

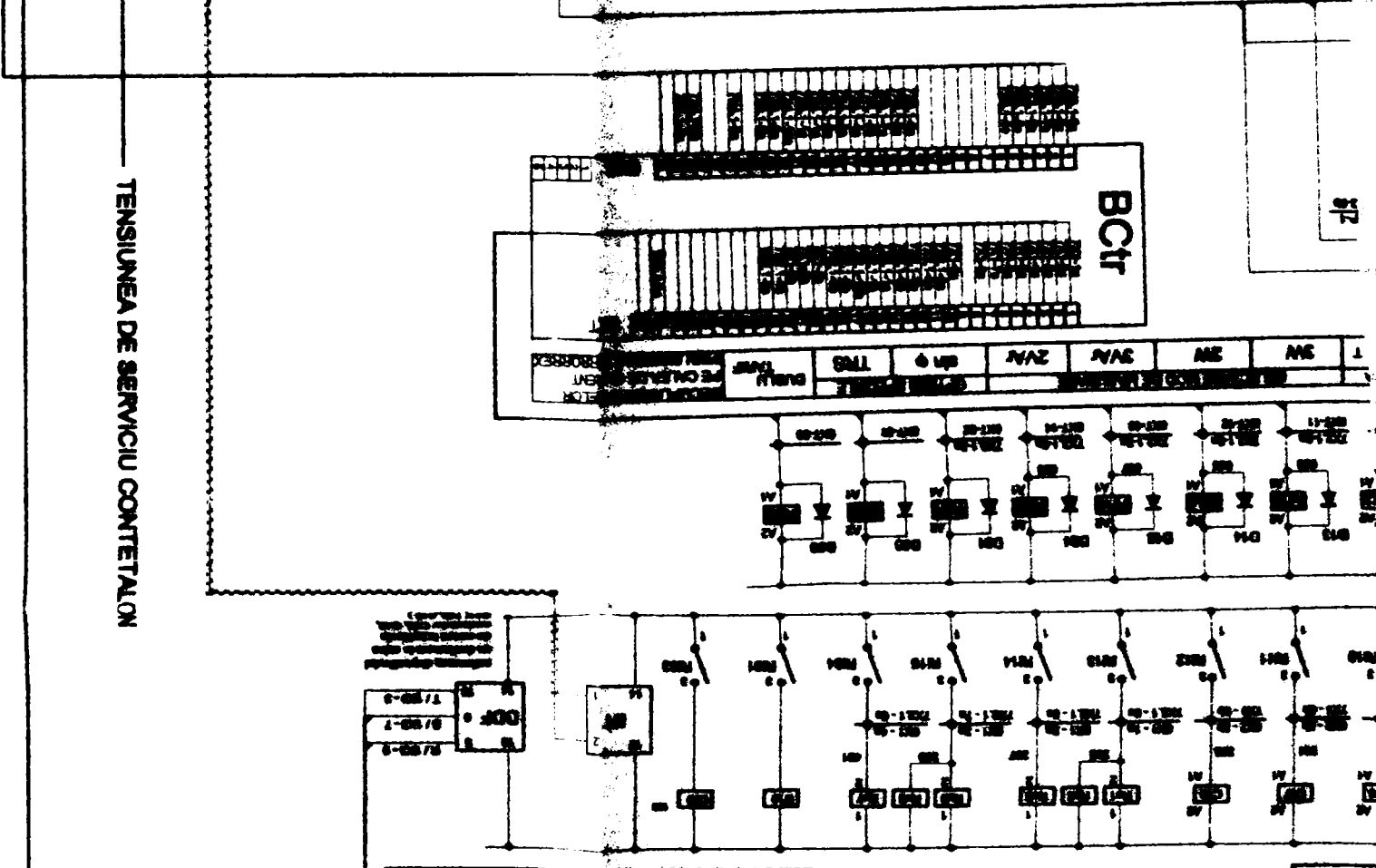
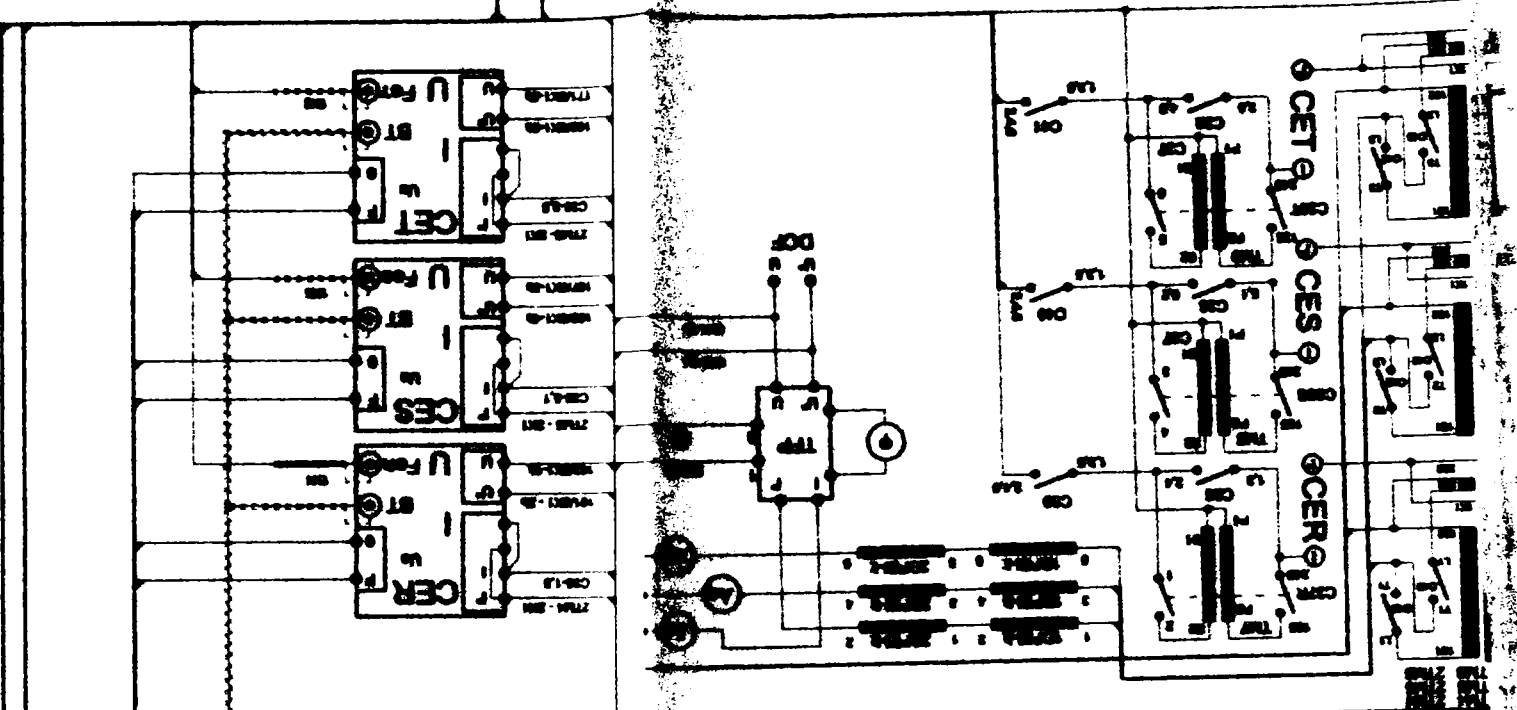
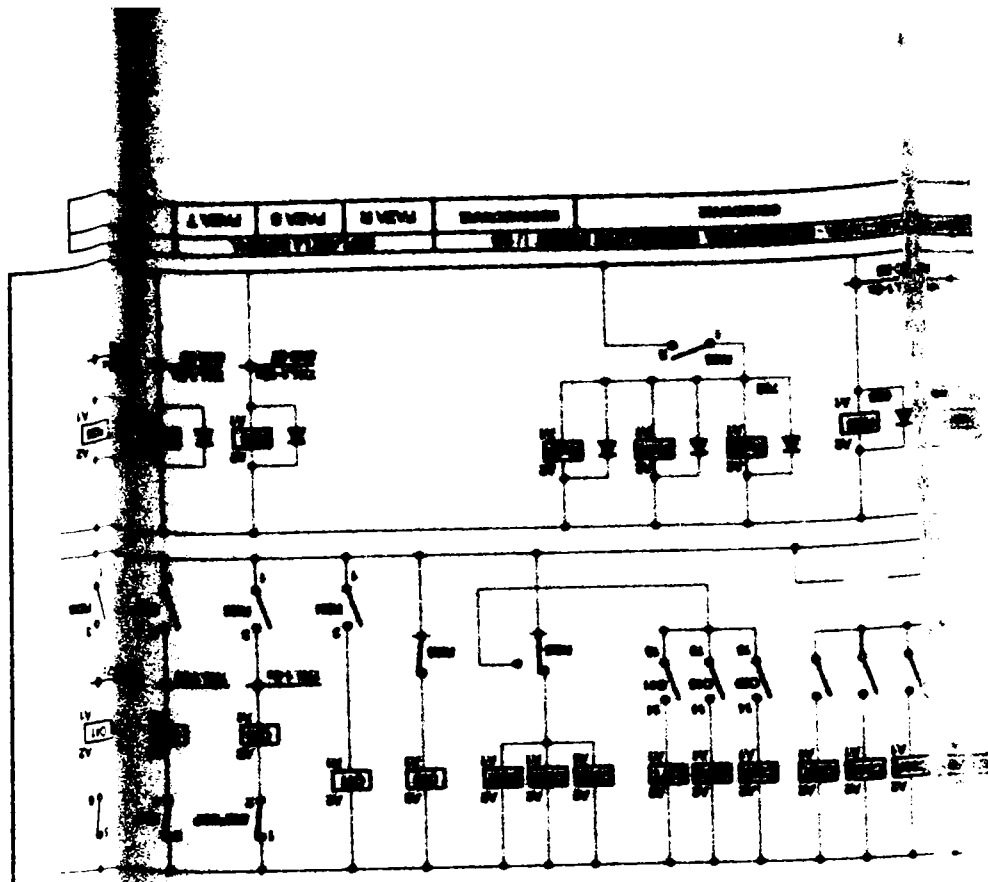
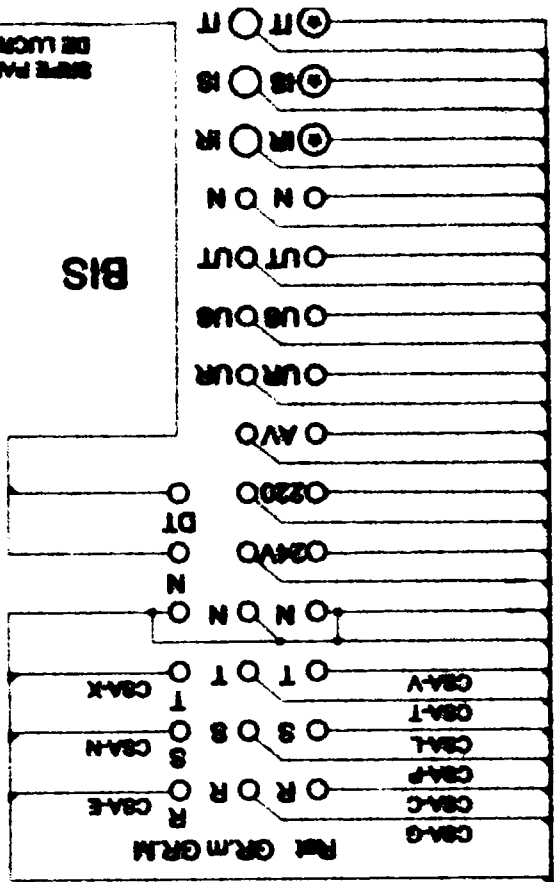
BIPE PANOURILE DE LUCRU

STATIE DE ETALONARE ANEXA 7

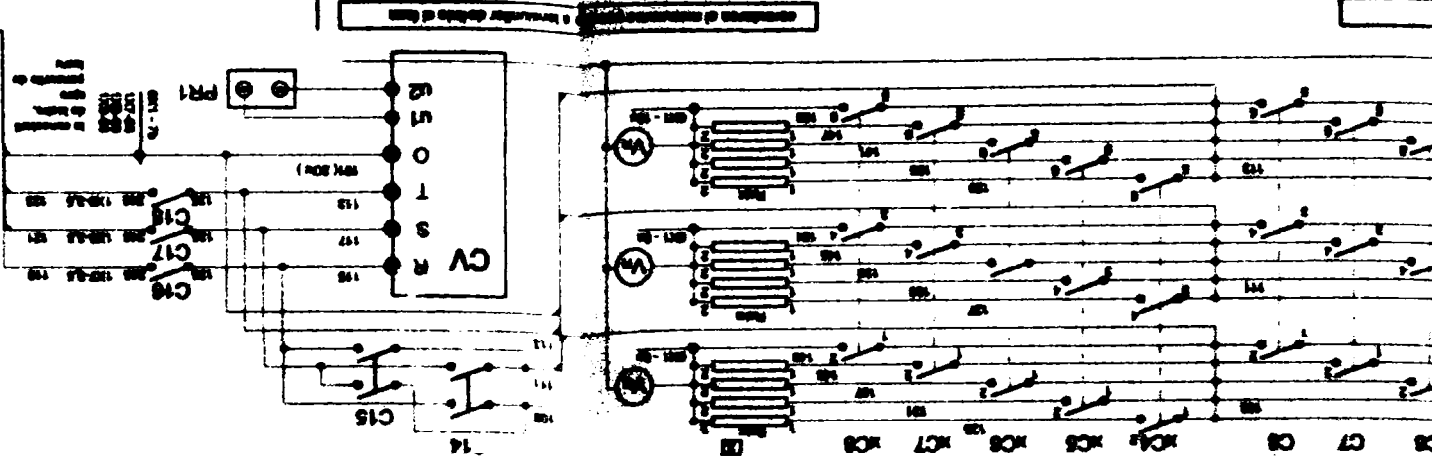
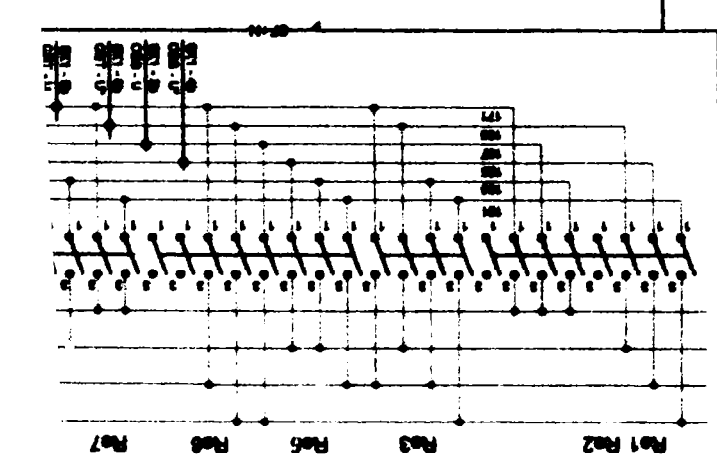
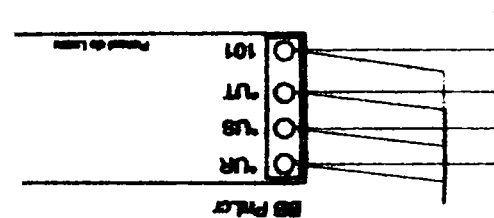


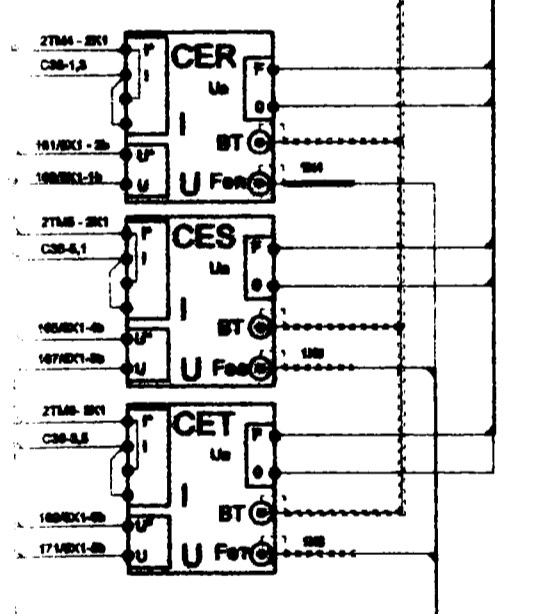
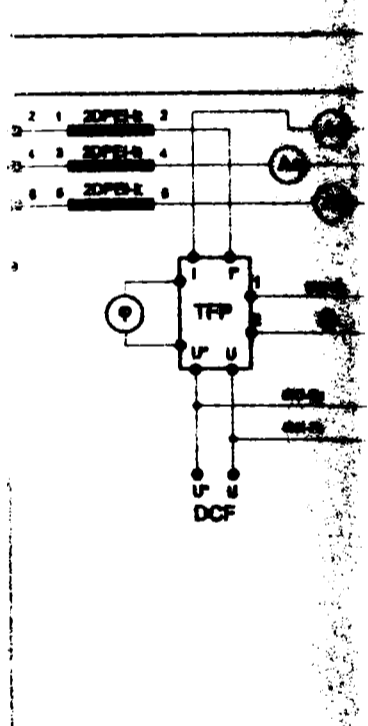
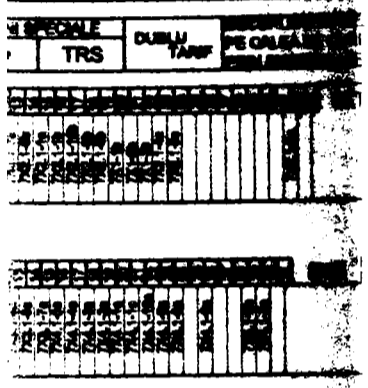
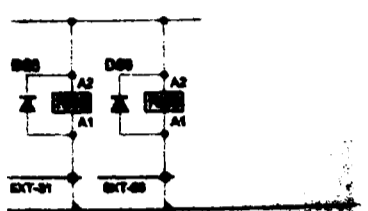
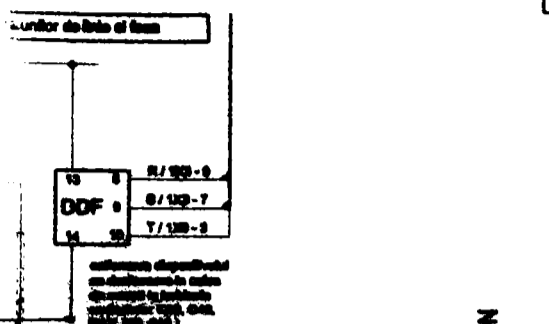
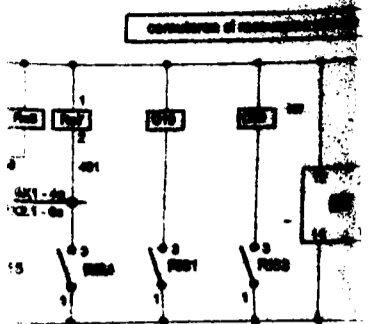
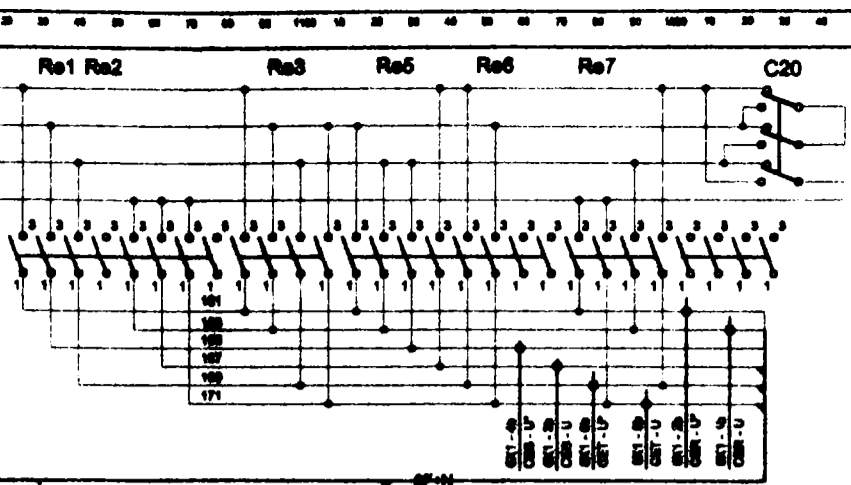
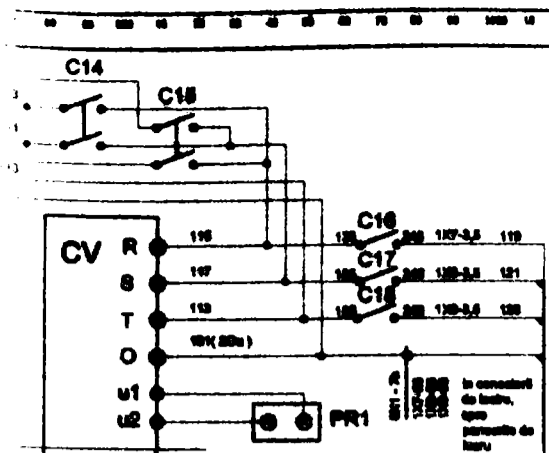
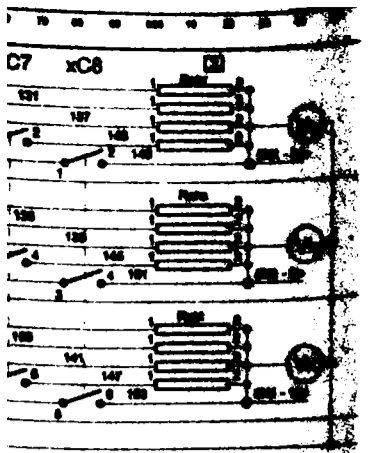
SEME PAVOULUI  
DE LUCRU

BIS

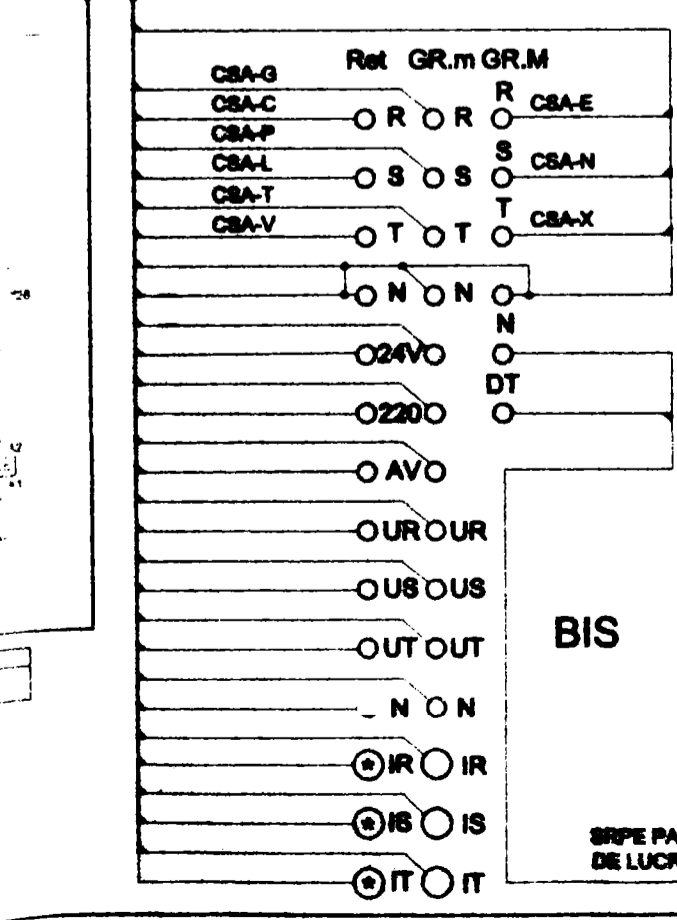
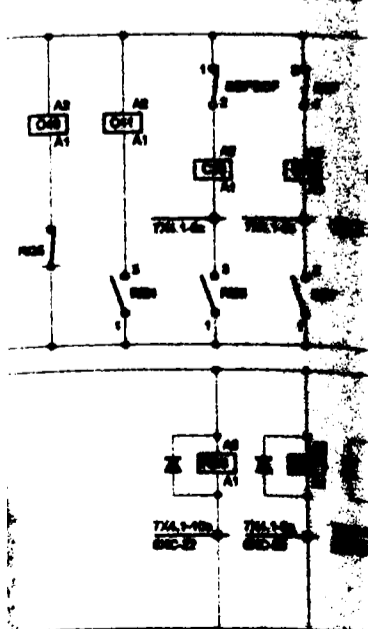
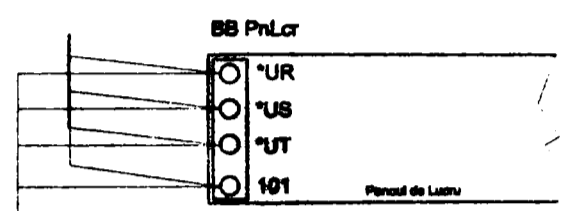


TENSIUNEA DE SERVICIU CONTEJALON





TENSIUNEA DE SERVICIU CONTETALON



BIS

BIPE PANOURILE DE LUCRU

STATIE DE ETALONARE SE 04 ANEXA 7