

# 6750

**PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW**  
**MERA-PIAP**  
**Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81**

**ZESPÓŁ AUTOMATYKI ELEKTRONICZNEJ**  
**PRACOWNIA REGULATORÓW ELEKTRONICZNYCH**

074

A

**Główny wykonawca** mgr inż. Grzegorz Kazimierski

**Wykonawcy** doc. dr inż. Jacek Korytkowski  
techn. Krystyna Miedzierska  
techn. Zenon Wieteska

**Konsultant**

**Nr zlecenia** S1245

Utrzymanie Laboratorium Badań Funkcjonalnych i Atestacji Środków Automatyki Elektrycznej i Elektronicznej. Etap 3. Opracowanie i wykonanie zestawu do 6-cio kanałowej rejestracji analogowych sygnałów wolnozmiennych oraz oprogramowanie do określania parametrów dynamicznych regulatorów PID zgodnie z normą IEC 546.

**Zleceniodawca**

**Pracę rozpoczęto dnia**

07.91

**zakończono dnia**

15.12.91

**Kierownik Pracowni**

*Pietrusiński*

Z-ca Dyr. d/s

**Kierownik Zespołu**

Bad.-Rozwojowych

*Korytkowski*

mgr inż. Z. Pietrusiński

doc. dr inż. J. Korytkowski

dr inż. J. Jabłkowski

**Praca zawiera:**

**Rozdzielnik - ilość egz:**

stron

Egz. 1

rysunków

Egz. 2

fotografii

Egz. 3

tabel

Egz. 4

tablic

Egz. 5

załączników

Egz. 6

**Nr rejestr.**

6750

1

## **Analiza deskrytorowa**

BADANIA : ANALIZA + OSPRZĘT + OPROGRAMOWANIE.

## **Analiza dokumentacyjna**

Praca zawiera projekt rozwiązania konstrukcyjnego zestawu 6-cio kanałowej rejestracji wolnozmiennych sygnałów analogowych.

## **Tytuły poprzednich sprawozdań**

1. Utrzymanie Laboratorium Badań Funkcjonalnych i Atestacji Środków Automatyki Elektrycznej i Elektronicznej.  
Etap 1. Stan obecny oraz potrzeby i perspektywy Laboratorium Badań Funkcjonalnych i Atestacji Środków Automatyki Elektrycznej i Elektronicznej.  
Nr rejestr. 6672.
2. Utrzymanie Laboratorium Badań Funkcjonalnych i Atestacji Środków Automatyki Elektrycznej i Elektronicznej.  
Etap 2. Projekt koncepcyjny modernizacji i wyposażenia Laboratorium w 1991r.  
Nr rejestr. 6700.

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW  
ZESPÓŁ AUTOMATYKI ELEKTRONICZNEJ

Utrzymanie Laboratorium Badań Funkcjonalnych i Atestacji  
Środków Automatyki Elektrycznej i Elektronicznej

Etap 3. Opracowanie i wykonanie zestawu do 6-cio kanałowej rejestracji analogowych sygnałów wolnozmiennych oraz oprogramowanie do określania parametrów dynamicznych regulatorów PID zgodnie z normą IEC 546.

Opracowali :

doc.dr inż.Jacek Korytkowski

mgr inż.Grzegorz Kazimierski

WARSZAWA. GRUDZIEŃ 1991 r.

Spis treści :

1. Wstęp
2. Rozwiązanie konstrukcyjne zestawu do 6-cio kanałowej rejestracji analogowych sygnałów wolnozmiennych.
3. Oprogramowanie zestawu i instrukcja posługiwania się programem.
4. Wnioski.

## 1. Wstęp.

W ramach etapu 3 pracy "Utrzymanie Laboratorium Badań Funkcjonalnych i Atestacji Środków Automatyki Elektrycznej i Elektronicznej" wykonano zestaw do 6-cio kanałowej rejestracji analogowych sygnałów wolnozmiennych, napisano i przetestowano oprogramowanie zestawu we współpracy z komputerem IBM PC.

## 2. Rozwiązanie konstrukcyjne zestawu do 6-cio kanałowej rejestracji analogowych sygnałów wolnozmiennych.

Zestaw komputerowej rejestracji sygnałów analogowych składa się z komputera IBM XT/AT oraz sprzężonej z nim kasety. Schemat blokowy zestawu przedstawiono na rys.1. Pakiety zestawu zostały podzielone na dwie części:

- pakiety części cyfrowej w skład których wchodzi pakiety przetworników A/C, C/A, wejść i wyjść dyskretnych sprzężone za pomocą pakietu interfejsu GIM-BUSEXT z komputerem,
- pakiety analogowe tzn. pakiety separatorów oraz pakiet przetwornika ADF sygnałów analogowych z filtracją.

Obie części zestawu posiadają własne zasilacze i są w pełni od siebie rozdzielone galwanicznie.

Poniżej omówiono poszczególne pakiety wchodzące w skład zestawu:

### Pakiety części cyfrowej:

#### - GIM - BUSEXT

Jest to pakiet adaptera pozwalającego na dołączenie do magistrali płyty głównej komputera IBM XT/AT magistralę BUSMAT II. Pakiet montowany jest w gnieździe płyty głównej komputera i łączony kablem z kasetą zawierającą magistralę BUSMAT II. W ten sposób pakiety zamontowane na magistrali BUSMAT II mogą być obsługiwane programowo przez komputer w taki sposób jakby były zamontowane bezpośrednio na magistrali komputera.

- ADO 16/12 jest pakietem 16 wejść analogowych o rozdzielczości 13 bitów. Część analogowa jest odizolowana od części cyfrowej systemu za pomocą szybkich transoptorów. Izolowana część analogowa wymaga osobnego zasilania napięciami  $\pm 15V$ . Czas przetwarzania jednego kanału wynosi ok.  $75\mu s$ .

- DAO 4/10. Jest to pakiet czterokanałowego przetwornika C/A o rozdzielczości 11 bitów (10 bitów + znak). Część analogowa jest odizolowana od części cyfrowej i wymaga osobnego zasilania napięciami  $\pm 15V$ . Zakres napięć wyjściowych  $-10V \dots +10V$ . Wyjścia zabezpieczone są przed przeciążeniem. Czas ustalania napięcia wyjściowego  $2\mu s$ .

- GXM - IPS jest zasilaczem przeznaczonym do zasilania zestawu pakietów części cyfrowej. Zasilacz wytwarza następujące napięcia

+5V/10A; +12V/2A; +15V/1A; -15V/1A; -12V/0,1A. Napięcia wyprowadzone są na magistralę BUSMAT II.

Dodatkowo część cyfrowa zestawu mieści pakiety wejść i wyjść dyskretnych (OBI16 oraz OBO16). Wejścia i wyjścia dwystanowe są izolowane od pozostałych wejść i wyjść oraz od wspólnego punktu zasilania systemu komputerowego.

#### Pakiety części analogowej:

- SG01

Są to cztery separatory zasilane z zasilacza +24V. Dwa separatory (SEP1 i SEP2) wykorzystywane są do przetwarzania sygnałów pomiarowych zestawu 0...10V oraz 0...20mA na sygnał 0...10V (standardowy sygnał przetwornika A/C). Dwa pozostałe (SEP3 i SEP4) przetwarzają sygnał standardowy przetwornika C/A 0...10V na sygnał napięciowy 0...10V oraz prądowy 0...20mA. Błąd przetwarzania separatorów < 0,5%, stała czasowa filtracji 0,5 sek.

- ADF jest pakietem 12 wejść analogowych z quasiseparacją, przetwarzający sygnał 0...20mA, 0...5mA, 0...5V, 0...10V na standardowy sygnał przetwornika A/C 0...10V. Błąd podstawowy < 0,6%.

Quasiseparacja galwaniczna umożliwia prawidłowy pomiar przy obecności napięcia zakłócającego wspólnego o wartości 220V/50Hz. Pakiet zasilany jest z zasilacza obiektowego  $\pm 15V$ .

Pakiety części analogowej zasilane są z zasilacza impulsowego +24V (napięcie to może być wykorzystane do zasilania obwodów wejść i wyjść dyskretnych) oraz zasilacza impulsowego  $\pm 15V$  (który jest wykorzystywany także do zasilania części analogowych pakietów przetworników A/C i C/A).

W załączniku 1 przedstawiono pomiary charakterystyk pakietów separatorów oraz pakietu ADF.

W części analogowej zestawu mieści się także tablica krosowa z łączówkami zaciskowymi umożliwiającą konfigurację zestawu. Wyprowadzenia sygnałów na zaciski przedstawiono na rys.2.

Wyprowadzone sygnały zostały podzielone na grupy funkcjonalne.

I tak kolejno na pierwszą listwę wyprowadzone są sygnały przetworników A/C i C/A, na drugą standardowe sygnały wyjść przetworników (tzn. separatorów pomiarowych SEP1 i SEP2 oraz pakietu ADF) oraz standardowe sygnały wejściowe separatorów wyjściowych (SEP3 i SEP4)

dzenia sygnałów na zaciski przedstawiono na rys.2.

Wyprowadzone sygnały zostały podzielone na grupy funkcjonalne.

I tak kolejno na pierwszej listwę wyprowadzone są sygnały przetworników A/C i C/A, na drugą standardowe sygnały wyjść przetworników (tzn. separatorów pomiarowych SEP1 i SEP2 oraz pakietu ADF) oraz standardowe sygnały wejściowe separatorów wyjściowych (SEP3 i SEP4) Listwy 3 i 4 zawierają dwuprzewodowe sygnały obiektowe wejściowych separatorów i pakietu ADF oraz dwuprzewodowe wyjścia separatorów wyjściowych. Na listwę 5 i 6 wyprowadzono wejścia i wyjścia dyskretne oraz pomocnicze napięcie zasilające 24V.



### 3. Oprogramowanie i instrukcja posługiwania się programem.

Oprogramowanie zestawu zostało napisane w języku C i zainstalowane na komputerze IBM 386.

Całość oprogramowania można podzielić na dwie części. Program CHARA.EXE jest programem działającym w czasie rzeczywistym. Z częstotliwością 1Hz pobierane są próbki 6 kanałów wejść analogowych oraz ustawiane 2 kanały wyjść analogowych. W programie istnieje możliwość ustawienia rodzaju sygnału wyjściowego (oddzielnie dla kanału 0 i kanału 1).

Do wyboru są następujące opcje :

- sygnał zerowy/niezerowy;
- sygnał skokowy;
- sygnał narastający liniowo;

Zgodnie z normą IEC 546 wyznaczenie wartości nastaw dynamicznych regulatorów wymaga tego rodzaju sygnałów na wejścia regulatora. Dla wyznaczenia stałej czasowej całkowania wartość mierzona powinna być stała w czasie, wówczas po zadziałaniu części różniczkującej algorytmu PID sygnał wyjściowy regulatora jest sygnałem jednostajnie narastającym. Dla wyznaczenia nastawy stałej czasowej różniczkowania sygnał wejściowy regulatora powinien być jednostajnie narastający.

Oprócz rodzaju sygnału wyjściowego wprowadza się parametry tego sygnału tzn. czas opóźnienia, wartość przyrostu na sekundę w procentach zakresu 0...10V lub amplitudę skoku sygnału.

Dodatkowo w kanale 0 został zainstalowany algorytm regulatora PID o ustawianych parametrach  $K_p$ ,  $T_i$ ,  $T_d$  i  $T_f$ .

Pobrane próbki sygnałów analogowych wejściowych są zapisywane do zbioru w sformatowanej postaci :

- Czas od początku eksperymentu,
- 2 kanały wyjść analogowych,
- 6 kanałów wejść analogowych.

W czasie trwania eksperymentu na monitorze są wyświetlane wartości w postaci numerycznej umożliwiające śledzenie eksperymentu.

Powstały w ten sposób zbiór jest źródłem dla drugiej części programu. Program CHARPRN.EXE umożliwia przedstawienie powstałej charakterystyki w postaci wydruku, a następnie wyznaczenie nastaw regulatorów.

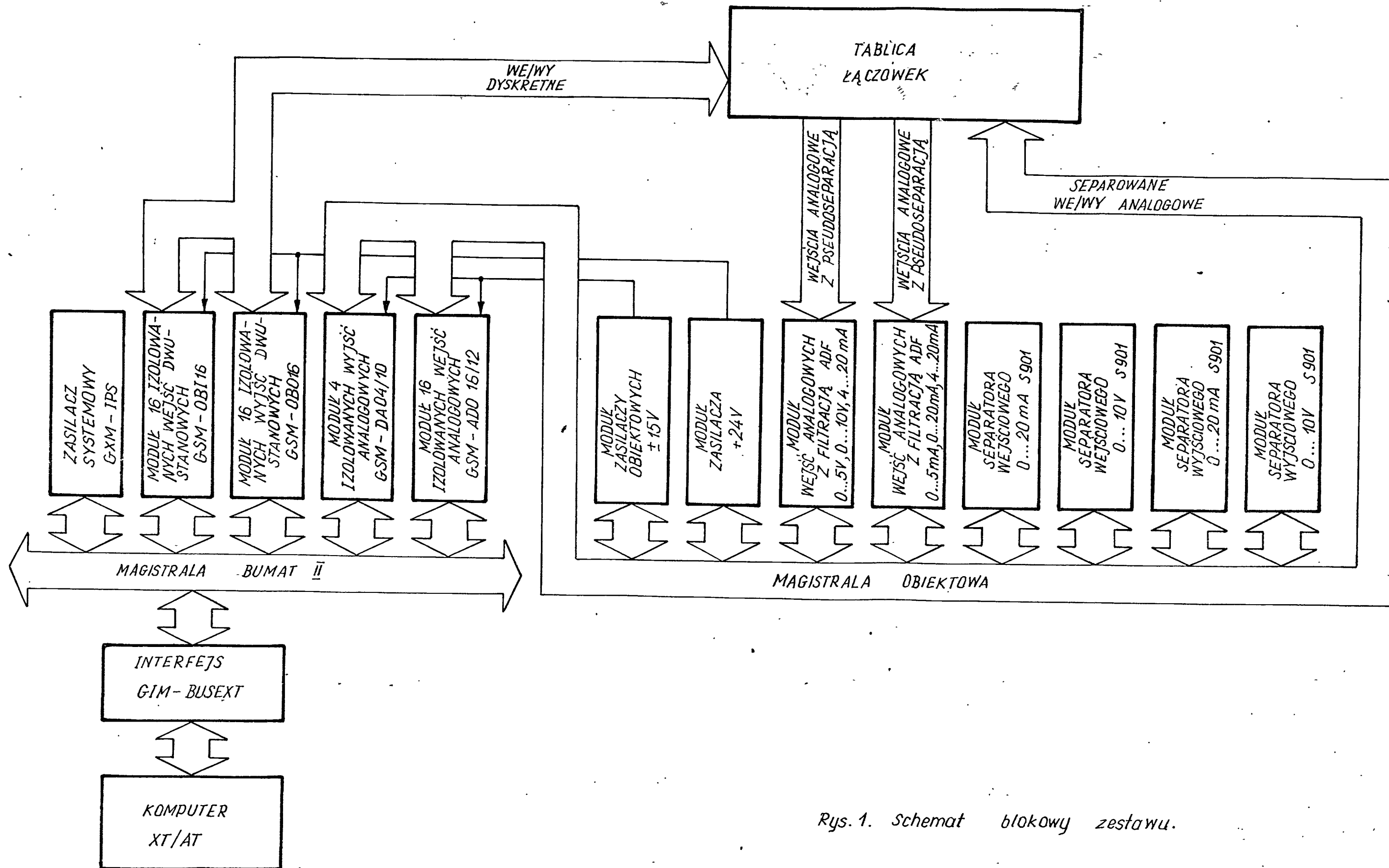
Po uruchomieniu programu CHARPRN wybierane są kanały, które mają być wydrukowane oraz rodzaj sygnału zerowy/niezerowy. W załączniku 2 przedstawione zostały listyngi programów CHARA.EXE oraz CHARPRN.EXE oraz przykładowe wydruki powstałych charakterystyk w postaci tabelarycznej i wykresu.

### 3. Wnioski.

Zestaw pakietów wejść/wyjść analogowych i dyskretnych obsługiwany programowo przez komputer z dodatkowymi pakietami separatorów sygnałów analogowych może służyć nie tylko do testowania regulatorów, lecz może być także bardzo przydatny do symulacji obiektowej, doboru nastaw regulatorów, sprawdzania algorytmów regulacji. Napisane oprogramowanie może być w dalszym ciągu modyfikowane i akceptowane do innych celów.

Aktualne dość bogate wyposażenie kasety np. w dodatkowe pakiety wejść/wyjść dyskretnych umożliwi rozszerzenie przyszłych zastosowań zestawu.

W szczególności wydaje się celowe opracowanie oprogramowania stanowiącego bibliotekę obiektowych programów symulacyjnych do sprawdzania algorytmów regulatorów oraz do doboru konfiguracji i nastaw.



Rys. 1. Schemat blokowy zestawu.

IN KAN 0	1	1	OUT SEP 1	IN SEP 1 (+)	1	IN ADF 9 (+)	OUT DSK 0 (+)	1	IN DSK 0 (+)
IN KAN 1	2	2	OUT SEP 2	IN SEP 1 (-)	2	IN ADF 9 (-)	OUT DSK 0 (-)	2	IN DSK 0 (-)
IN KAN 2	3	3	OUT ADF 1	IN SEP 2 (+)	3	IN ADF 10 (+)	OUT DSK 1 (+)	3	IN DSK 1 (+)
IN KAN 3	4	4	OUT ADF 2	IN SEP 2 (-)	4	IN ADF 10 (-)	OUT DSK 1 (-)	4	IN DSK 1 (-)
IN KAN 4	5	5	OUT ADF 3	IN ADF 1 (+)	5	IN ADF 11 (+)	OUT DSK 2 (+)	5	IN DSK 2 (+)
IN KAN 5	6	6	OUT ADF 4	IN ADF 1 (-)	6	IN ADF 11 (-)	OUT DSK 2 (-)	6	IN DSK 2 (-)
IN KAN 6	7	7	OUT ADF 5	IN ADF 2 (+)	7	IN ADF 12 (+)	OUT DSK 3 (+)	7	IN DSK 3 (+)
IN KAN 7	8	8	OUT ADF 6	IN ADF 2 (-)	8	IN ADF 12 (-)	OUT DSK 3 (-)	8	IN DSK 3 (-)
IN KAN 8	9	9	OUT ADF 7	IN ADF 3 (+)	9		OUT DSK 4 (+)	9	IN DSK 4 (+)
IN KAN 9	10	10	OUT ADF 8	IN ADF 3 (-)	10		OUT DSK 4 (-)	10	IN DSK 4 (-)
IN KAN 10	11	11	OUT ADF 9	IN ADF 4 (+)	11		OUT DSK 5 (+)	11	IN DSK 5 (+)
IN KAN 11	12	12	OUT ADF 10	IN ADF 4 (-)	12		OUT DSK 5 (-)	12	IN DSK 5 (-)
IN KAN 12	13	13	OUT ADF 11	IN ADF 5 (+)	13		OUT DSK 6 (+)	13	IN DSK 6 (+)
IN KAN 13	14	14	OUT ADF 12	IN ADF 5 (-)	14		OUT DSK 6 (-)	14	IN DSK 6 (-)
IN KAN 14	15	15	+15V	IN ADF 6 (+)	15		OUT DSK 7 (+)	15	IN DSK 7 (+)
IN KAN 15	16	16	GND	IN ADF 6 (-)	16		OUT DSK 7 (-)	16	IN DSK 7 (-)
OUT KAN 0	17	17	GND	IN ADF 7 (+)	17	OUT SEP 3 (+)	+24V	17	+24V
OUT KAN 1	18	18	-15V	IN ADF 7 (-)	18	OUT SEP 3 (-)	+24V	18	+24V
OUT KAN 2	19	19	IN SEP 3	IN ADF 8 (+)	19	OUT SEP 4 (+)	-24V	19	-24V
OUT KAN 3	20	20	IN SEP 4	IN ADF 8 (-)	20	OUT SEP 4 (-)	-24V	20	-24V

Sygnaty wejściowe/wyjściowe przetworników A/C i C/A

Sygnaty wyjściowe separatorów 0...10V

Sygnaty obiektowe separatorów

Wyjścia obiektowe dyskretne

Wejścia obiektowe dyskretne

Rys. 2. Schemat wyprowadzeń sygnałów na tablicę krosową z zestawu.

Z A Ł Ą C Z N I K 1

BADANIA KANAŁÓW ANALOGOWYCH ZESTAWU DO 6-CIO  
KANAŁOWEJ REJESTRACJI SYGNAŁÓW WOLNOZMIENNYCH

W A R S Z A W A GRUDZIEŃ 1991

16

1. SEPARATOR S-901N Nr fabr. 323/91 0V ... 10V / 0V ... 10V

sygnał wej [V]	sygnał wyj [V]	$\Delta$ [V]	$\delta$ [%]
-0,001	0,047	0,048	0,48
2,510	2,578	0,068	0,68
5,021	5,088	0,067	0,67
7,530	7,592	0,062	0,62
10,039	10,091	0,052	0,52

$$\delta_{\max} = 0,68\%$$

2. SEPARATOR S-901N Nr fabr. 322/91 0V ... 10V / 0V ... 10V

sygnał wej [V]	sygnał wyj [V]	$\Delta$ [V]	$\delta$ [%]
0,00	0,003	0,003	0,03
2,513	2,536	0,023	0,23
5,026	5,052	0,026	0,26
7,538	7,563	0,025	0,25
10,051	10,068	0,017	0,17

$$\delta_{\max} = 0,26\%$$

3. SEPARATOR S-901N Nr fabr. 321/91 0V ... 10V / 0mA ... 20mA

sygnał wej [V]	sygnał wyj [mA]	$\Delta$ [mA]	$\delta$ [%]
0,00	0,1	0,1	0,5
2,517	5,18	0,146	0,73
5,032	10,20	0,136	0,68
7,546	15,20	0,108	0,54
10,060	20,18	0,07	0,35

$$\delta_{\max} = 0,73\%$$

4. SEPARATOR S-901N Nr fabr. 320/91 0mA ... 20mA / 0V ... 10V

sygnał wej [mA]	sygnał wyj [V]	$\Delta$ [V]	$\delta$ [%]
20,01	10,054	0,049	0,49
15,01	7,557	0,052	0,52
10,01	5,062	0,057	0,57
5,014	2,567	0,06	0,6
0,01	0,052	0,047	0,47

$$\delta_{\max} = 0,6\%$$

5. 1 kanał 12-sto kanałowego pakietu wejść analogowych ADF  
0V ... 5V / 0V ... 10V

sygnał wej [V]	sygnał wyj [V]	$\Delta$ [V]	$\delta$ [%]
-0,007	-0,021	0,014	0,14
1,000	1,994	-0,006	-0,06
2,001	3,996	-0,006	-0,06
3,001	5,996	-0,006	-0,06
4,001	7,995	-0,007	-0,07
5,009	10,017	-0,001	-0,01

$$\delta_{\max} = 0,14\%$$



Z A Ł A C Z N I K 2

WYDRUKI PROGRAMÓW I PRZYKŁADY ZDJĘTYCH CHARAKTERYSTYK

W A R S Z A W A GRUDZIEŃ 1991

17

```

/* Program ustawiania i zadawania 2 wejosc analogowych oraz
   odczytu i rejestracji w zbiorze 6 wyjsc analogowych
   13.12.1991 \

#include <dos.h>
#include <stdio.h>
#include <string.h>

#define NAZWA 20 /* max rozmiar nazwy zbioru */
#define TIME 9999
#define IL_KANAL 8
/*-----*/
int adres_da = 0xfd40;
int adres_ad = 0xfd10;

unsigned int ti; /* licznik sekund */
double z_v_0,z_v_1,zv0,zv1; /* wartosci zadawane napieci X.XXXV */
double m_v_0,m_v_1,m_v_2,m_v_3,m_v_4,m_v_5,m_v_6; /* wartosci mierzone napieci x.xxxV */

double sum,wejpop,delta_pop,kp,tii,td,tf;
int arg_c;
char *arg_v[NAZWA];
int dp; /* dysk poczatkowy */
FILE *out; /* nazwa otwartego zbioru */
/*-----*/

void main(argc,argv)
int argc;
char *argv[];
{
/*-----*/
unsigned int kan_0,kan_1,tau_0,tau_1,czas_globalny;
char p;
double delta_0,delta_1,nzer(),reg();
struct time sek;
extern unsigned int ti;
extern int dp;
int sek_pop,sek_akt;
z_v_0 = z_v_1 = delta_0 = delta_1 = zv0 = zv1 = 0;
m_v_0 = m_v_1 = m_v_2 = m_v_3 = m_v_4 = m_v_5 = m_v_6 = 0;
sek_pop = sek_akt = 0;
kan_0 = kan_1 = tau_0 = tau_1 = 0;
sum = 0.0;
wejpop = 0.0;
delta_pop = 0.0;
/*=====*/
/* Naglowek */
printf("PROGRAM ZADAWANIA 2 WYJSC I REJESTRACJI 6 WEJSC ANALOGOWYCH\n");
printf("\t MERA PIAP\t Zaklad Automatyki Elektrycznej\t G.K.\n\n");

/*=====*/
/* Ustawianie parametrow sygnalow zadawanych */
czytaj:
printf("Czas trwania eksperymentu w sekundach: ");
scanf("%d",&czas_globalny);

```

```

printf("### KANAL 0 ###\n\tKanal wykorzystywany (T/N?)\n");
if(( p = getch()) == 't' || p == 'T')
{
printf("\tWyjscie zerowe/niezerowe (Z/N?)\n");
if(( p = getch()) == 'z' || p == 'Z')
kan_0 = 10;
else
kan_0 = 20;
printf("\tRodzaj zadawanego sygnalu\n");
printf("\t\tSygnal skokowy ..... 1\n");
printf("\t\tSygnal liniowy ..... 2\n");
printf("\t\tRegulator PID ..... 3\n");
if ((p = getch()) == '1')
{
kan_0 = kan_0 + 1;
printf("\t\tCzas opoznienia w sekundach ");
scanf("%d",&tau_0);
printf("\n\t\tWartosc sygnalu w procentach zakresu ");
scanf("%lf",&delta_0);
}
else
if( p == '2')
{
kan_0 = kan_0 + 2;
printf("\t\tCzas opoznienia w sekundach ");
scanf("%d",&tau_0);
printf("\n\t\tPrzyrost sygnalu w procentach zakresu na sekunde ");
scanf("%lf",&delta_0);
}
else
{
printf("\n\tSygnal skokowy czy liniowy (S/L)?\n");
if((p=getch()) == 's' || p == 'S')
kan_0 = kan_0 + 3;
else
kan_0 = kan_0 + 4;
printf("\t\tCzas opoznienia w sekundach ");
scanf("%d",&tau_0);
printf("\n\t\tKp = ");
scanf("%lf",&kp);
printf("\n\t\tTi = ");
scanf("%lf",&tii);
printf("\n\t\tTd = ");
scanf("%lf",&td);
printf("\n\t\tTf = ");
scanf("%lf",&tf);
printf("\n\t\tWartosc sygnalu w procentach zakresu ");
scanf("%lf",&delta_0);
}
}
else
{
kan_0 = 0;

```

```

        tau_0 = 0;
        delta_0 = 0.0;
    }

/*-----*/
    printf("### KANAL 1 ###\n\tKanal wykorzystywany (T/N?)\n");
    if(( p = getch()) == 't' || p == 'T')
    {
        printf("\tWyjście zerowe/niezerowe (Z/N?)\n");
        if(( p = getch()) == 'z' || p == 'Z')
            kan_1 = 10;
        else
            kan_1 = 20;
        printf("\tRodzaj zadawanego sygnału\n");
        printf("\t\tSygnał skokowy ..... 1\n");
        printf("\t\tSygnał liniowy ..... 2\n");
        if (getch() == '1')
        {
            kan_1 = kan_1 + 1;
            printf("\t\tCzas opóźnienia w sekundach   ");
            scanf("%d",&tau_1);
            printf("\n\t\tWartość sygnału w procentach zakresu   ");
            scanf("%lf",&delta_1);
        }
        else
        {
            kan_1 = kan_1 + 2;
            printf("\t\tCzas opóźnienia w sekundach   ");
            scanf("%d",&tau_1);
            printf("\n\t\tPrzyrost sygnału w procentach zakresu na sekundę   ");
            scanf("%lf",&delta_1);
        }
    }
    else
    {
        kan_1 = 0;
        tau_1 = 0;
        delta_1 = 0.0;
    }

/*=====*/
    printf("\nSygnały wyjściowe:\n");
    printf("kanal 0 = sygnał ");
    if(kan_0 == 0)
        printf("niewykorzystany\n");
    else
    {
        if(kan_0 == 13 || kan_0 == 14)
            printf("wyjściowy regulatora ");
        if(kan_0 == 11 || kan_0 == 12 || kan_0 == 13 || kan_0 == 14)
            printf("zerowy ");
        else
            printf("niezerowy ");
        if(kan_0 == 13 || kan_0 == 14)
            printf("o wejściu ");
        if(kan_0 == 11 || kan_0 == 21 || kan_0 == 13)
            printf("skokowy");
    }

```

```

        else
            printf("liniowy");
            if(kan_0 == 13 || kan_0 == 14)
                printf("m ");
            printf(" o parametrach:\n");
            printf("\t\t\tCzas opoznienia = %d sek, delta = %4.2f%%",tau_0,delta_0);
            if(kan_0 == 12 || kan_0 == 22 || kan_0 == 14)
                printf(" /sek\n");
            else
                printf("\n");
            if(kan_0 == 13 || kan_0 == 14)
                printf("\nNastawy regulatora Kp=%3.1f\t\n"
                    "Ti=%3.1f\tTd=%3.1f\tTf=%3.1f\n",kp,tii,td,tf);
        }
    printf("kanal 1 = sygnał ");
    if(kan_1 == 0)
        printf("niewykorzystany\n");
    else
        {
            if(kan_1 == 11 || kan_1 == 12)
                printf("zerowy ");
            else
                printf("niezerowy ");
            if(kan_1 == 11 || kan_1 == 21)
                printf("skokowy ");
            else
                printf("liniowy ");
            printf("o parametrach:\n");
            printf("\t\t\tCzas opoznienia = %d sek, delta = %4.2f%%",tau_1,delta_1);
            if(kan_1 == 12 || kan_1 == 22)
                printf(" /sek\n");
            else
                printf("\n");
        }

    /*-----*/
    printf("Potwierdz zgodność (T/N)?\n");
    if(( p = getch() == 'N' || p == 'n')
        {
            printf("Powtórz wprowadzanie parametrów !\n");
            goto czytaj;
        }

    /*=====*/
        /* OTWIERANIE ZBIORU DO ZAPISU DANYCH */
    arg_c = argc;
    if(argc > 1)
        strcpy(arg_v[1],argv[1]);
    if(otworz()==1)
        {
            printf("Nie można utworzyć pliku danych !");
            exit();
        }
    /*-----*/

    for(ti = 0; ti<=czas_globalny+1;)

```

```

(
    gettime(&sek);
    if((sek_akt = sek.ti_sec) != sek_pop)
    {
        /*=====*/
/* Przeliczenie wyjsc */
/* kanal 0 */
if( kan_0 == 0 )
    zv0 = 0.0; /* wyjscie niewykorzystane */
else
    {
        if ( ti >= tau_0)
        {
            if( kan_0 == 11 || kan_0 == 21)
            {
                /* sygnal skokowy */
                zv0 = delta_0/10.0;
            }
            else
            {
                if(kan_0 == 12 || kan_0 == 22)
                {
                    /* sygnal liniowy */
                    zv0 = zv0 + delta_0/10.0;
                }
                else
                {
                    if(kan_0 == 13)
                        zv0 = reg(delta_0/10.0);
                    else
                        zv0 = reg(wejpop+delta_0/10.0);
                }
            }
        }
    }
if( kan_0 == 21 || kan_0 == 22 || kan_0 == 23)
    z_v_0 = nzer(zv0);
else
    z_v_0 = zv0;
/*=====*/
/* kanal 1 */
if( kan_1 == 0 )
    zv1 = 0.0; /* wyjscie niewykorzystane */
else
    {
        if ( ti >= tau_1)
        {
            if( kan_1 == 11 || kan_1 == 21)
            {
                /* sygnal skokowy */
                zv1 = delta_1/10.0;
            }
            else
            {
                /* sygnal liniowy */
                zv1 = zv1 + delta_1/10.0;
            }
        }
    }
if( kan_1 == 21 || kan_1 == 22)
    z_v_1 = nzer(zv1);

```

```

        else
            z_v_1 = z_v1;
        /*=====*/
/* ograniczenie sygnalow */
        if(zv0 <= 0.0)
            z_v_0 = 0.0;
        if(zv0 >= 9.9999)
            z_v_0 = 9.99999;
        if(zv1 <= 0.0)
            z_v_1 = 0.0;
        if(zv1 >= 9.9999)
            z_v_1 = 9.99999;
        /*=====*/
        set_out();
        set_in();
        zapis();
        ti++;
        sek_pop = sek_akt;
    /*=====*/
    }
}
/*=====*/
zamknij();
}
/*=====*/
/* otwieranie zbioru do zapisu */
otworz()
{
extern int arg_c;
extern char *arg_v[];
extern dp;
char name[NAZWA];
char *roz = ".TAB";
char *pkt;
extern FILE *out;
int dd,i;

    /* WYBOR DYSKU */
    dp = getdisk();
    dd = setdisk();
    printf(" Wybierz dysk: ");
    for(i='A'; i<=('A'-2+dd); i++)
        printf(" %c,",i);
    printf("...\n");
    if((dd=toupper(getch())) >= 'A' && dd <='F')
        setdisk(dd-'A');
    else
        dd = dp;
    /* nazwa zbioru do zapisu */
    if(arg_c > 1)
        strcpy(name,arg_v[1]);
    else
        {

```

```

        printf(" Podaj nazwe zbioru ... ");
        scanf("%12s",&name);
    }
   strupr(name);
    if((pkt=strchr(name,'.')) == NULL)
        strcat(name,roz);
    else
        strcpy(pkt,roz);

    if((out=fopen(name,"wt+")) == NULL)
        return(1);
    printf(" Zapis do zbioru Zc: \\%s\n",dd,name);
    /* naglowek do zbioru */
    fprintf(out,"czas\tout1\tout2\tin1\tin2\tin3\tin4\tin5\tin6\n");
}
/*-----*/
/* ZAMKNIECIE ZBIOROW */

zamknij()
{
fclose(out);
setdisk(dp);
}
/*-----*/
/* ustawianie wyjsc analogowych */

set_out()
{
unsigned int    bity;
unsigned int    znak,ml_bajt,st_bajt;
                outportb(adres_da,0);
                bity = z_v_0*1024/10;
                ml_bajt = bity % 256;
                st_bajt = ((bity / 256) & 0x3) | 0x4;
                outportb(adres_da+2,st_bajt);
                outportb(adres_da+3,ml_bajt);

                outportb(adres_da,1);
                bity = z_v_1*1024/10;
                ml_bajt = bity % 256;
                st_bajt = ((bity / 256) & 0x3) | 0x4;
                outportb(adres_da+2,st_bajt);
                outportb(adres_da+3,ml_bajt);
}
/*-----*/
/* pomiary wejsc analogowych */

set_in()
{
unsigned char wynik_ml,wynik_st;
int n,bity;
double bity_double;
    /* kanal 0 */
    outport(adres_ad,0);
    delay(1);
    outport(adres_ad+1,1);
    for(n=0; (n<100) && ((inportb(adres_ad)) & 1) == 0; n++)
        ;
}

```



```

wynik_st = inportb(adres_ad+1);
wynik_ml = inportb(adres_ad+1);
bity = (int)((int)wynik_ml /256*(int)wynik_st);
bity_double = (double)bity;
m_v_0 = bity_double*10/4096;
/* kanal 1 */
outport(adres_ad,1);
delay(1);
outport(adres_ad+1,1);
for(n=0; (n<100) && ((inportb(adres_ad)) & 1) == 0; n++)
;
wynik_st = inportb(adres_ad+1);
wynik_ml = inportb(adres_ad+1);
bity = (int)((int)wynik_ml +256*(int)wynik_st);
bity_double = (double)bity;
m_v_1 = bity_double*10/4096;

/* kanal 2 */
outport(adres_ad,2);
delay(1);
outport(adres_ad+1,1);
for(n=0; (n<100) && ((inportb(adres_ad)) & 1) == 0; n++)
;
wynik_st = inportb(adres_ad+1);
wynik_ml = inportb(adres_ad+1);
bity = (int)((int)wynik_ml +256*(int)wynik_st);
bity_double = (double)bity;
m_v_2 = bity_double*10/4096;

/* kanal 3 */
outport(adres_ad,3);
delay(1);
outport(adres_ad+1,1);
for(n=0; (n<100) && ((inportb(adres_ad)) & 1) == 0; n++)
;
wynik_st = inportb(adres_ad+1);
wynik_ml = inportb(adres_ad+1);
bity = (int)((int)wynik_ml +256*(int)wynik_st);
bity_double = (double)bity;
m_v_3 = bity_double*10/4096;
}
/*-----*/
/* zapis linii do zbioru */
zapis()
{
extern FILE #out;

fprintf(out, "%04d\t%4.3f\t%4.3f\t%4.3f\t%4.3f\t%4.3f\t%4.3f\t%4.3f\n",
ti, z_v_0, z_v_1, m_v_0, m_v_1, m_v_2, m_v_3, m_v_4, m_v_5, m_v_6);
printf("%4d\t%4.3f\t%4.3f\t%4.3f\t%4.3f\t%4.3f\t%4.3f\t%4.3f\n",
ti, z_v_0, z_v_1, m_v_0, m_v_1, m_v_2, m_v_3, m_v_4, m_v_5, m_v_6);
}
/*-----*/
/* przeliczanie sygnalow niezerowych */
double nzer(wyj)

```

```
double wyj;
{
double out;
out = wyj + (10.0-wyj)/5;
return(out);
}
/*-----*/
/* Dodatkowy program symulacyjny algorytmu regulatora PID */
/* Algorytm numeryczny z książki "Digital Control Systems" */
/* z dodatkowym algorytmem filtracji składowej różniczki */
double reg(wej)
double wej;
{
int i;
extern double tf,td,tii,kp;
double wyj;
double delta_wej=0.0;
extern double sum,wejpop,delta_pop;
sum = sum + wejpop;
delta_wej = ((wej-wejpop) + delta_pop*(tf-1))/tf;
wyj = kp*(wej + sum/tii + td*delta_wej);
wejpop = wej;
delta_pop = delta_wej;
return(wyj);
}
/*=====*/
```

```

#include <dos.h>
#include <stdio.h>
#include <string.h>

#define MAXX 55 /*ilosc znakow z odczytu zbioru */
#define MAX 132 /*ilosc znakow w linii*/
#define YY 129 /* szerokosc wydruku */
#define NAZWA 20 /* max rozmiar nazwy zbioru */
#define TIME 9999
#define IL_KANAL 8
/*-----*/
char line[MAX];
int arg_c;
char *arg_v[NAZWA];
int dp; /* dysk poczatkowy */
FILE *out; /* nazwa otwartego zbioru */

/*-----*/

void main(argc,argv)
int argc;
char *argv[];
{
/*-----*/
extern FILE *out;
int kan_0,kan_1,kan_2,kan_3,kan_4,kan_5,kan_00,kan_11;
int i,j,ti,l_p,licznik;
double zv1,zv2,mv1,mv2,mv3,mv4,mv5,mv6,nzero();
char p;
char lone[MAX],linia[MAX];

/* OTWIERANIE ZBIORU DO CZYTANIA DANYCH */
arg_c = argc;
if(argc > 1)
strcpy(arg_v[1],argv[1]);
if((otworz()==1)
{
printf("Nie ma takiego pliku danych !");
exit();
}

/*-----*/
kan_0 = kan_1 = kan_2 = kan_3 = kan_4 = kan_5 = kan_00 = kan_11 = 0;
/* ustawianie parametrow wykresu */
printf("\nRejestracja kanalu ### 0 ### (T/N)");
if(( p = getch()) == 'T' || p == 't')
{
printf("\n\t\t\tWejscie zerowe/niezerowe ? (Z/N)");
if(( p = getch()) == 'n' || p == 'N')
kan_0 = 11;
else
kan_0 = 10;
}
else

```

```
        kan_0 = 0;
printf("\nRejestracja kanalu *** 1 *** (T/N)");
if(( p = getch()) == 'T' || p == 't')
{
    printf("\n\t\t\tWejscie zerowe/niezerowe ? (Z/N)");
    if(( p = getch()) == 'n' || p == 'N')
        kan_1 = 11;
    else
        kan_1 = 10;
}
else
    kan_1 = 0;
printf("\nRejestracja kanalu *** 2 *** (T/N)");
if(( p = getch()) == 'T' || p == 't')
{
    printf("\n\t\t\tWejscie zerowe/niezerowe ? (Z/N)");
    if(( p=getch()) == 'n' || p == 'N')
        kan_2 = 11;
    else
        kan_2 = 10;
}
else
    kan_2 = 0;
printf("\nRejestracja kanalu *** 3 *** (T/N)");
if(( p = getch()) == 'T' || p == 't')
{
    printf("\n\t\t\tWejscie zerowe/niezerowe ? (Z/N)");
    if(( p = getch()) == 'n' || p == 'N')
        kan_3 = 11;
    else
        kan_3 = 10;
}
else
    kan_3 = 0;
printf("\nRejestracja kanalu *** 4 *** (T/N)");
if(( p = getch()) == 'T' || p == 't')
{
    printf("\n\t\t\tWejscie zerowe/niezerowe ? (Z/N)");
    if(( p = getch()) == 'n' || p == 'N')
        kan_4 = 11;
    else
        kan_4 = 10;
}
else
    kan_4 = 0;
printf("\nRejestracja kanalu *** 5 *** (T/N)");
if(( p = getch()) == 'T' || p == 't')
{
    printf("\n\t\t\tWejscie zerowe/niezerowe ? (Z/N)");
    if(( p = getch()) == 'n' || p == 'N')
        kan_5 = 11;
    else
        kan_5 = 10;
}
else
```

```

        kan_5 = 0;
printf("\n\t### Rejestracja sygnalu zadajacego 0 ### ");
if(( p = getch()) == 'T' || p == 't')
    kan_00 = 10;
else
    kan_00 = 0;
printf("\n\t### Rejestracja sygnalu zadajacego 1 ### ");
if(( p = getch()) == 'T' || p == 't')
    kan_11 = 10;
else
    kan_11 = 0;
l_p = 1;
printf("\nPodstawa czasu ( w sekundach ) ");
scanf("%d",&l_p);

/*-----*/
fprintf(stdprn, "\033\063\0a"); /* przesuw papieru */
fprintf(stdprn, "\033\017"); /* gestosc wydruku */
/*-----*/

/* rysowanie osi Y */

for(i=0; i<=100;)
{
    fprintf(stdprn, "%-4.1d ", i);
    i += 10;
}
fprintf(stdprn, "\n\0xc9");
for(i=1; i<=YY; i++)
{
    if(i%13 == 0)
        fprintf(stdprn, "\0xd1");
    else
        fprintf(stdprn, "\0xcd");
}
fprintf(stdprn, "\n");
/*-----*/
fgets(lone, MAX, out);
licznik = 0;
while( fgets (lone, MAX, out) != NULL)
{
    strcpy(line, lone);
    sscanf(line, "%d %f %f %f %f %f %f %f", &ti, &zv1, &zv2, &mv1, &mv2, &mv3, &mv4, &mv5, &mv6);
    if (ti%l_p == 0 && ti != 0)
    {
        /* drukowanie osi czasu ze znacznikami co 20 razy*/

        if( (ti%(l_p*20) == 0) && (ti != 0)) /* co 20 razy ciagla linia */
        {
            linia[0] = '\0xc7';
            for(i=1; i<=MAX-3; i++)
                if( i%13 == 0)
                    linia[i] = '\0xc5';
            else
                linia[i] = '\0xc4';
            linia[MAX-2] = '\0xb6';

```

```

        linia[MAX-1] = '\n';
        linia[MAX] = NULL;
    }
else
    {
        if(( tiZ(l_p*10) == 0) && (ti != 0)) /* co 10 razy linia przerywana */
        {
            linia[0] = '\xc7';
            for(i=1; i<= MAX-3; i++)
                if( i%13 == 0)
                    linia[i] = '\xc5';
                else
                    if( i%2 == 0)
                        linia[i] = '\x20';
                    else
                        linia[i] = '\xc4';

            linia[MAX-2] = '\xb6';
            linia[MAX-1] = '\n';
            linia[MAX] = NULL;
        }
    }
else
    {
        /* co 5 razy linia przerywana */
        if((tiZ(l_p*5) == 0) && (ti != 0))
        {
            linia[0] = '\xc7';
            for(i=1; i<=MAX-3; i++)
                if( i%13 == 0)
                    linia[i] = '\xc5';
                else
                    linia[i] = '\x20';
            linia[MAX-2] = '\xb6';
            linia[MAX-1] = '\n';
            linia[MAX] = NULL;
        }
    }
else
    {
        /* co 2 razy linia pusta */
        if(tiZ(l_p*2) == 0)
        {
            linia[0] = '\xba';
            for(i=1; i<=MAX-3; i++)
                linia[i] = '\x20';
            linia[MAX-2] = '\xba';
            linia[MAX-1] = '\n';
            linia[MAX] = NULL;
        }
        else
        {
            for(i=0; i<=MAX-2; i++)
                linia[i] = '\x20';
            linia[MAX-1] = '\n';
            linia[MAX] = NULL;
        }
    }
}
}

```

```
        /* zaznaczenie 0 kanalu out */
if(kan_00 != 0)
{
    i = 0;
    while((zv1*YY/10.0)>i++);
    liniafil = '\xf8';
}
/* zaznaczenie 1 kanalu out */
if(kan_11 != 0)
{
    i = 0;
    while((zv2*YY/10.0)>i++);
    liniafil = '\xf8';
}
/* zaznaczenie 0 kanalu in */
if(kan_0 != 0)
{
    if(kan_0 == 11)
        mv1 = nzero(mv1);
    i = 0;
    while((mv1*YY/10.0)>i++);
    liniafil = '\xfe';
}
/* zaznaczenie 1 kanalu in */
if(kan_1 != 0)
{
    if(kan_1 == 11)
        mv2 = nzero(mv2);
    i = 0;
    while((mv2*YY/10.0)>i++);
    liniafil = '\xfe';
}
/* zaznaczenie 2 kanalu in */
if(kan_2 != 0)
{
    if(kan_2 == 11)
        mv3 = nzero(mv3);
    i = 0;
    while((mv3*YY/10.0)>i++);
    liniafil = '\xfe';
}
/* zaznaczenie 3 kanalu in */
if(kan_3 != 0)
{
    if(kan_3 == 11)
        mv4 = nzero(mv4);
    i = 0;
    while((mv4*YY/10.0)>i++);
    liniafil = '\xfe';
}
/* zaznaczenie 4 kanalu in */
if(kan_4 != 0)
{
    if(kan_4 == 11)
```

```

        mv5 = nzero(mv5);
        i = 0;
        while((mv5*YY/10.0)>i++);
        linia[i] = '\xfe';
    }
    /* zaznaczenie 5 kanalu in */
    if(kan_5 != 0)
    {
        if(kan_5 == 11)
            mv6 = nzero(mv6);
        i = 0;
        while((mv6*YY/10.0)>i++);
        linia[i] = '\xfe';
    }

    fputs(linia, stdprn);
}

    licznik++;

}

/*-----*/
zanknij();
}
/*-----*/
double nzero(wej)
double wej;
{
    double ina;
    ina = wej - (10.0-wej)/4;
    return(ina);
}
/*-----*/
/* otwieranie zbioru do odczytu */
otworz()
{
    extern int arg_c;
    extern char *arg_v[];
    extern dp;
    char name[NAZWA];
    char *roz = ".TAB";
    char *pkt;
    extern FILE *out;
    int dd,i;

    /* WYBOR DYSKU */
    dp = getdisk();
    dd = setdisk();
    printf(" Wybierz dysk: ");
    for(i='A'; i<=('A'-2+dd); i++)
        printf(" %c,",i);
    printf("...?\n");
    if((dd=toupper(getch())) >= 'A' && dd <='F')
        setdisk(dd-'A');
    else

```



```
dd = dp;
/* nazwa zbioru do odczytu */
if(arg_c > 1)
    strcpy(name,arg_v[1]);
else
{
    printf(" Podaj nazwe zbioru ... ");
    scanf("%12s",&name);
}
strupr(name);
if((pkt=strchr(name,',')) == NULL)
    strcat(name,roz);
else
    strcpy(pkt,roz);

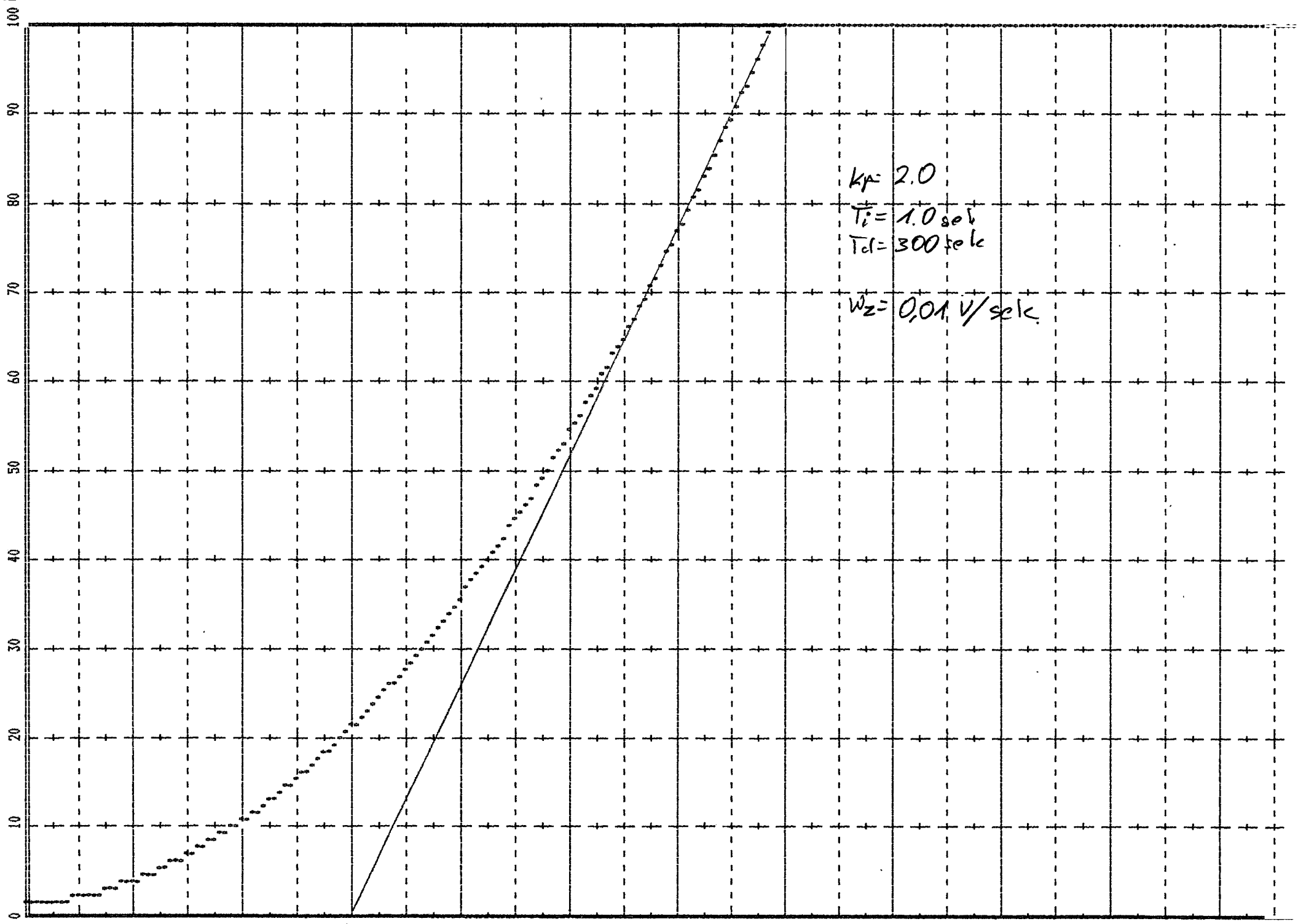
if((out=fopen(name,"rt+")) == NULL)
    return(1);
printf(" Dane ze zbioru %c: \\%s\n",dd,name);
/* naglowek do zbioru */

}
/*-----*/
/* ZAMKNIECIE ZBIOROW */
zamknij()
{
fclose(out);
setdisk(dp);
}
/*-----*/
```

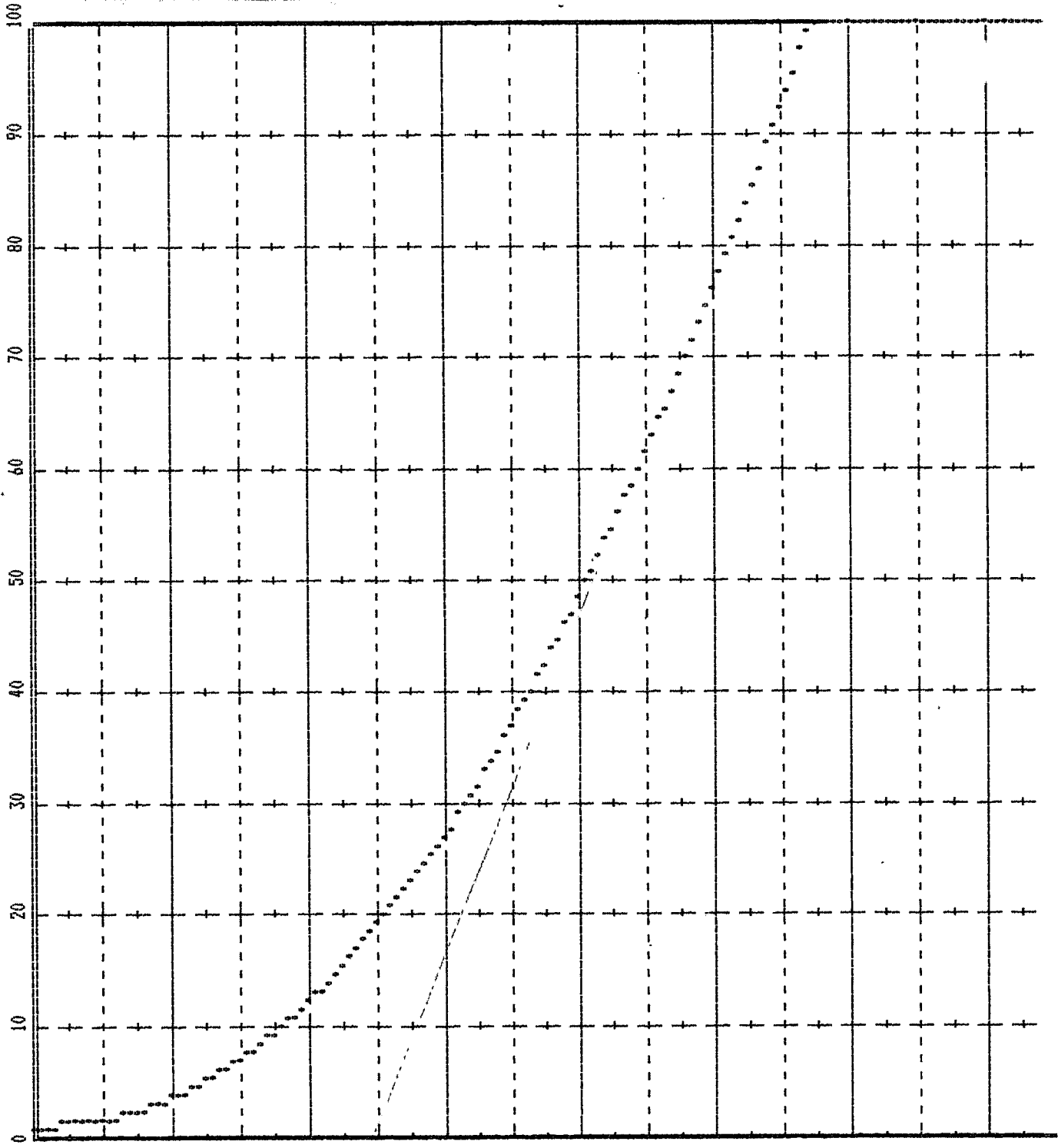
czas	out1	out2	in1	in2	in3	in4	in5	in6
0000	0.000	0.000	-0.002	-0.005	-0.005	-0.002	0.000	0.000
0001	0.000	0.000	-0.002	-0.002	-0.002	-0.002	0.000	0.000
0002	0.000	0.000	-0.002	-0.005	-0.002	-0.002	0.000	0.000
0003	0.000	0.000	-0.005	-0.005	-0.002	-0.002	0.000	0.000
0004	0.000	0.000	-0.005	-0.005	-0.005	-0.002	0.000	0.000
0005	0.000	0.000	-0.012	-0.002	-0.002	-0.002	0.000	0.000
0006	0.000	0.000	-0.002	-0.005	-0.005	-0.007	0.000	0.000
0007	0.000	0.000	-0.002	-0.002	-0.005	-0.002	0.000	0.000
0008	0.000	0.000	-0.005	-0.005	-0.005	-0.002	0.000	0.000
0009	0.000	0.000	-0.002	-0.005	-0.002	-0.002	0.000	0.000
0010	0.000	0.000	-0.002	-0.002	-0.005	-0.002	0.000	0.000
0011	0.000	0.000	-0.002	-0.005	-0.002	-0.002	0.000	0.000
0012	0.000	0.000	-0.002	-0.002	-0.005	-0.002	0.000	0.000
0013	0.000	0.000	-0.002	-0.005	-0.002	-0.002	0.000	0.000
0014	0.000	0.000	-0.002	-0.007	-0.005	-0.002	0.000	0.000
0015	0.000	0.000	-0.005	-0.005	-0.002	-0.002	0.000	0.000
0016	0.000	0.000	-0.002	-0.007	-0.002	-0.002	0.000	0.000
0017	0.000	0.000	-0.005	-0.002	-0.002	-0.005	0.000	0.000
0018	0.000	0.000	-0.002	-0.002	-0.007	-0.002	0.000	0.000
0019	0.000	0.000	-0.002	-0.007	-0.002	-0.002	0.000	0.000
0020	6.250	0.000	6.299	-0.002	-0.002	6.296	0.000	0.000
0021	6.171	0.000	6.196	-0.002	-0.002	6.216	0.000	0.000
0022	6.093	0.000	6.121	-0.002	-0.005	6.125	0.000	0.000
0023	6.017	0.000	6.055	-0.002	-0.005	6.057	0.000	0.000
0024	5.942	0.000	5.981	-0.002	-0.005	5.984	0.000	0.000
0025	5.868	0.000	5.896	-0.005	-0.002	5.901	0.000	0.000
0026	5.795	0.000	5.825	-0.005	-0.002	5.828	0.000	0.000
0027	5.724	0.000	5.754	-0.005	-0.002	5.771	0.000	0.000
0028	5.654	0.000	5.681	-0.005	-0.002	5.691	0.000	0.000
0029	5.586	0.000	5.615	-0.005	-0.002	5.613	0.000	0.000
0030	5.518	0.000	5.562	-0.005	-0.002	5.569	0.000	0.000
0031	5.452	0.000	5.483	-0.005	-0.005	5.493	0.000	0.000
0032	5.387	0.000	5.415	-0.002	-0.002	5.417	0.000	0.000
0033	5.323	0.000	5.359	-0.005	-0.005	5.371	0.000	0.000
0034	5.260	0.000	5.286	-0.005	-0.005	5.291	0.000	0.000
0035	5.198	0.000	5.229	-0.005	-0.005	5.222	0.000	0.000
0036	5.138	0.000	5.181	-0.005	-0.002	5.178	0.000	0.000
0037	5.078	0.000	5.112	-0.005	-0.002	5.115	0.000	0.000
0038	5.020	0.000	5.078	-0.005	-0.002	5.061	0.000	0.000
0039	4.962	0.000	5.000	-0.002	-0.005	4.998	0.000	0.000
0040	4.906	0.000	4.946	-0.002	-0.005	4.944	0.000	0.000
0041	4.851	0.000	4.883	-0.002	-0.007	4.885	0.000	0.000
0042	4.796	0.000	4.834	-0.005	-0.002	4.832	0.000	0.000
0043	4.743	0.000	4.771	-0.005	-0.005	4.773	0.000	0.000
0044	4.690	0.000	4.727	-0.005	-0.002	4.719	0.000	0.000
0045	4.639	0.000	4.673	-0.002	-0.005	4.688	0.000	0.000
0046	4.588	0.000	4.614	-0.002	-0.002	4.619	0.000	0.000
0047	4.539	0.000	4.565	-0.002	-0.002	4.573	0.000	0.000
0048	4.490	0.000	4.519	0.01	-0.012	4.519	0.000	0.000
0049	4.442	0.000	4.470	-0.002	-0.002	4.475	0.000	0.000
0050	4.395	0.000	4.434	-0.005	-0.002	4.436	0.000	0.000
0051	4.349	0.000	4.380	-0.002	-0.007	4.380	0.000	0.000
0052	4.303	0.000	4.331	-0.005	-0.002	4.338	0.000	0.000
0053	4.259	0.000	4.299	-0.005	-0.005	4.290	0.000	0.000

0054	4.215	0.000	4.236	-0.005	-0.005	4.246	0.000	0.000
0055	4.172	0.000	4.192	-0.002	-0.002	4.194	0.000	0.000
0056	4.130	0.000	4.165	-0.005	-0.005	4.163	0.000	0.000
0057	4.089	0.000	4.111	-0.002	-0.002	4.124	0.000	0.000
0058	4.048	0.000	4.080	-0.005	-0.002	4.075	0.000	0.000
0059	4.008	0.000	4.033	-0.005	-0.005	4.036	0.000	0.000
0060	3.969	0.000	3.994	-0.002	-0.005	4.004	0.000	0.000
0061	3.931	0.000	3.953	-0.005	-0.002	3.953	0.000	0.000
0062	3.893	0.000	3.916	-0.005	-0.002	3.914	0.000	0.000
0063	3.856	0.000	3.887	-0.005	-0.005	3.875	0.000	0.000
0064	3.820	0.000	3.840	-0.005	-0.002	3.848	0.000	0.000
0065	3.784	0.000	3.801	-0.002	-0.005	3.811	0.000	0.000
0066	3.750	0.000	3.765	-0.002	-0.002	3.772	0.000	0.000
0067	3.715	0.000	3.740	-0.002	-0.005	3.738	0.000	0.000
0068	3.682	0.000	3.699	-0.002	-0.005	3.699	0.000	0.000
0069	3.649	0.000	3.667	-0.005	-0.005	3.682	0.000	0.000
0070	3.616	0.000	3.638	-0.005	-0.002	3.645	0.000	0.000
0071	3.584	0.000	3.606	-0.005	-0.002	3.608	0.000	0.000
0072	3.553	0.000	3.567	-0.005	-0.002	3.577	0.000	0.000
0073	3.523	0.000	3.555	-0.002	-0.005	3.545	0.000	0.000
0074	3.492	0.000	3.508	-0.002	-0.005	3.513	0.000	0.000
0075	3.463	0.000	3.491	-0.005	-0.005	3.484	0.000	0.000
0076	3.434	0.000	3.452	-0.005	-0.002	3.459	0.000	0.000
0077	3.406	0.000	3.423	-0.005	-0.002	3.425	0.000	0.000
0078	3.378	0.000	3.403	-0.005	-0.002	3.391	0.000	0.000
0079	3.351	0.000	3.376	-0.002	-0.005	3.374	0.000	0.000
0080	3.324	0.000	3.350	-0.005	-0.005	3.347	0.000	0.000
0081	3.298	0.000	3.325	-0.002	-0.005	3.323	0.000	0.000
0082	3.272	0.000	3.293	-0.005	-0.002	3.303	0.000	0.000
0083	3.247	0.000	3.267	-0.002	-0.005	3.269	0.000	0.000
0084	3.222	0.000	3.242	0.01	-0.007	3.245	0.000	0.000
0085	3.198	0.000	3.201	-0.005	-0.002	3.218	0.000	0.000
0086	3.174	0.000	3.196	-0.002	-0.002	3.196	0.000	0.000
0087	3.151	0.000	3.169	-0.007	-0.002	3.162	0.000	0.000
0088	3.128	0.000	3.142	-0.005	-0.005	3.152	0.000	0.000
0089	3.105	0.000	3.120	-0.002	-0.005	3.123	0.000	0.000
0090	3.083	0.000	3.096	-0.002	-0.005	3.108	0.000	0.000
0091	3.062	0.000	3.081	-0.002	-0.005	3.088	0.000	0.000
0092	3.041	0.000	3.066	-0.005	-0.005	3.069	0.000	0.000
0093	3.020	0.000	3.042	-0.002	-0.005	3.040	0.000	0.000
0094	3.000	0.000	3.027	-0.002	-0.002	3.030	0.000	0.000
0095	2.980	0.000	3.003	-0.005	-0.005	3.010	0.000	0.000
0096	2.961	0.000	2.993	-0.005	-0.005	2.976	0.000	0.000
0097	2.941	0.000	2.964	-0.005	-0.002	2.966	0.000	0.000
0098	2.923	0.000	2.942	-0.005	-0.005	2.952	0.000	0.000
0099	2.905	0.000	2.927	-0.002	-0.002	2.922	0.000	0.000
0100	2.887	0.000	2.905	-0.005	-0.005	2.913	0.000	0.000
0101	2.869	0.000	2.886	-0.005	-0.005	2.878	0.000	0.000
0102	2.852	0.000	2.876	-0.005	-0.005	2.878	0.000	0.000
0103	2.835	0.000	2.866	-0.005	-0.005	2.859	0.000	0.000
0104	2.819	0.000	2.847	-0.005	-0.005	2.839	0.000	0.000
0105	2.802	0.000	2.822	-0.005	-0.005	2.820	0.000	0.000
0106	2.787	0.000	2.800	-0.005	-0.005	2.815	0.000	0.000
0107	2.771	0.000	2.793	-0.005	-0.002	2.795	0.000	0.000
0108	2.756	0.000	2.778	-0.005	-0.005	2.776	0.000	0.000

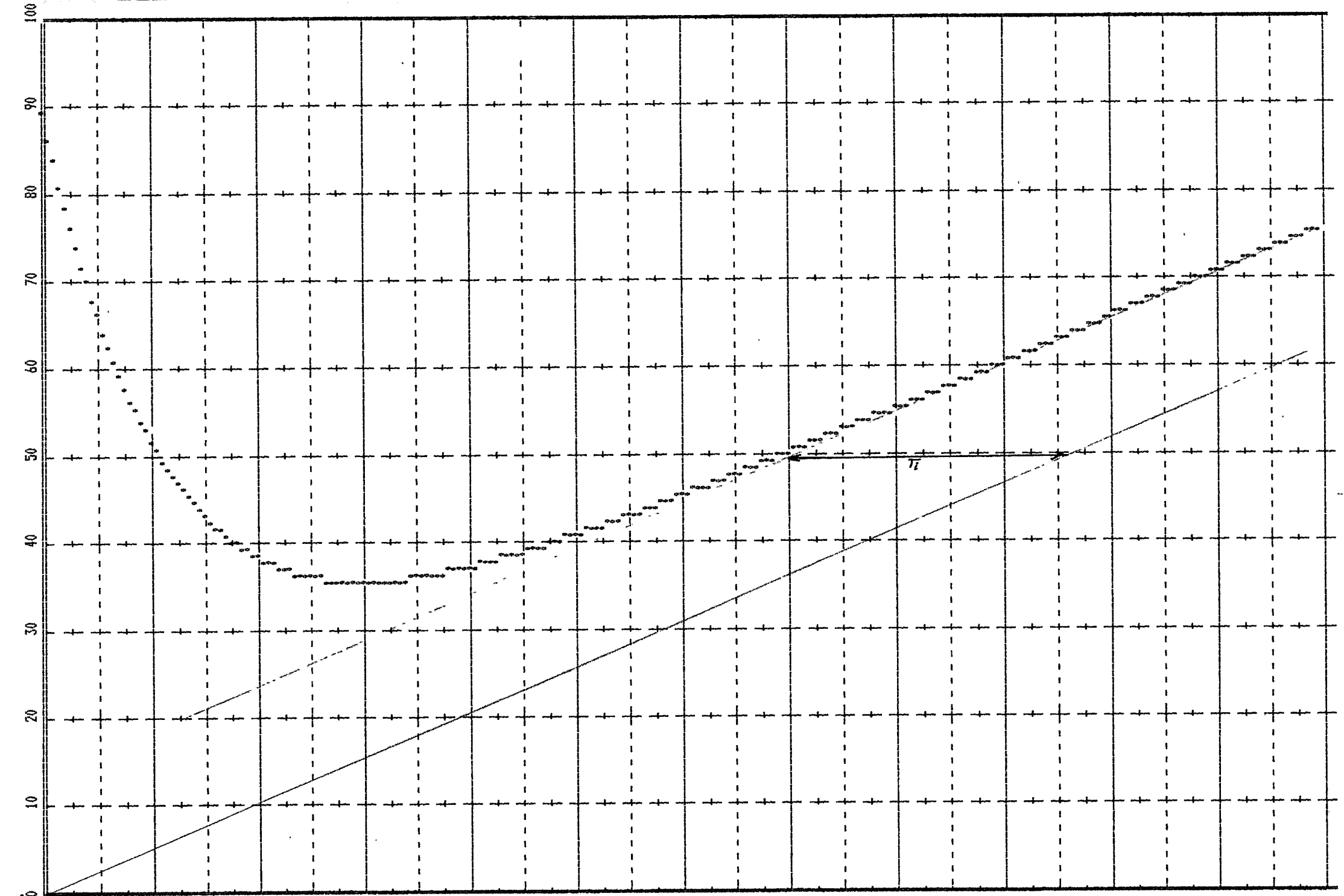
36



37-



$K_p = 2.0$   
 $T_i = 1.0 \text{ sek}$   
 $T_d = 240 \text{ sek}$   
 $W_z = 0,001 \text{ \% / sek}$



880

$K_p = 5.0$ ;  $T_i = 4 \text{ min}$ ;  $T_d = 12.6 \text{ min}$ ;  $\alpha = 6.3$

$\Delta t = 50 \text{ sek}$