

074

A

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

Ośrodek Badań Niezawodności i Jakości.

Pracownia Badań Robotów.

Główny wykonawca

Wykonawcy mgr inż. J. Skrzeczkowski, mgr J. Molak,
 tech. Z. Leszczyński, tech. E. Król.

Konsultant

Nr zlecenia K-28

Adaptacja skomputeryzowanych stanowisk badawczych do badań robotów przemysłowych zg. z wymaganiami normy ISO/DIS 9283.
 Etap 3. Uruchomienie układów pomiarowych i opracowanie programów użytkowych oraz wyznaczenie charakterystyk wybranego robota w celu sprawdzenia poprawności działania stanowisk.

Zleceniodawca kontynuacja CPBR 7.1

Pracę rozpoczęto dnia.....Zakończono dnia..91.12.15.

Kierownik Pracowni

mgr inż. J. Skrzeczkowski

Kierownik OBN

mgr inż. K. Majdan

Z-ca Dyrektora d/s
 Badawczo-Rozwojowych

dr inż. J. Jabikowski

Praca zawiera:

stron 3

rysunków

fotografii

tabel

tablic

załączników 1

arkuszy

Rozdzielnik - ilość egz.:

Egz. 1

BOINTE

Egz. 2

OBN

Egz. 3

OBN

Egz. 4

Egz. 5

Egz. 6

Egz. 7

Nr rejestr. 6785

OBN 91

1

Analiza deskryptorowa

ROBOTY PRZEMYSŁOWE + BADANIA

Analiza dokumentacyjna

Sprawozdanie zawiera opisy programów użytkowych dla robota IRb-60 i mikrokomputera IBM PC, MSA-80.3100 oraz wydruki programów i wyniki sprawdzeń charakterystyk zgodnie z założeniami zawartymi w normie ISO 9283.

Tytuły poprzednich sprawozdań

SPIS TREŚCI

1. Wstęp
2. Zakres pracy
3. Programy użytkowe dla IRb-60
4. Programy użytkowe dla mikrokomputera IBM-PC i MSA-80.3100
 - 4.1. Założenia
 - 4.2. Zakres pracy
 - 4.3. Wydruki programów oraz wyniki sprawdzeń charakterystyk
5. Wnioski

1. WSTEP

W ramach realizacji CPBR 7.1 "Roboty przemysłowe", celu 28 pt. "Opracowanie nowoczesnych metod badań robotów przemysłowych oraz wykonanie stanowisk badawczych", opracowano i wykonano skomputeryzowane stanowiska do prowadzenia badań robotów przemysłowych.

Stanowiska zapewniały mechanizację i automatyzację badań wykonanych zgodnie z wymaganiami określonymi w odpowiednich normach ZN. W celu stworzenia możliwości badania charakterystyk robotów, zgodnie z normą ISO 9283, wykonano adaptację stanowisk. Międzynarodowa norma ISO 9283 zachowuje zakres i przebieg badań oraz kryteria oceny i metody badań normy ISO/DIS 9283. Zmienione zostały jedynie oznaczenia niektórych parametrów. W opracowanych programach użytkowych na mikrokomputer IBM-PC przyjęto obowiązujące oznaczenia zgodne z normą ISO 9283.

2. ZAKRES PRACY

Praca obejmuje uruchomienie układów pomiarowych, opracowanie i uruchomienie programów użytkowych dla wybranego robota (IRb-60) oraz mikrokomputera IBM-PC i MSA-80.3100, jak również, w celu sprawdzenia poprawności działania stanowisk, wyznaczenie następujących charakterystyk zgodnie z wymaganiami normy ISO 9283:

- dokładność pozycjonowania wielokierunkowego
- dokładność i powtarzalność odtwarzania odcinka
- czas stabilizacji położenia
- przeregulowanie położenia
- dryft położenia
- dokładność i powtarzalność odtwarzania torów o kształtach prostokątnym i kołowym w płaszczyźnie ściśle określonej w przestrzeni
- długość odcinka stabilizacji
- dokładność i powtarzalność prędkości

3. PROGRAM UŻYTKOWY DLA ROBOTA IRb-60

W celu sprawdzenia poprawności działania adaptowanych stanowisk opracowano program użytkowy dla robota IRb-60.

Program użytkowy zapisany na kasecie zawiera pełny zestaw podprogramów sekwencji ruchów robota dla sprawdzeń wykonanych zgodnie z wymaganiami określonymi w normie ISO 9283.

W tabeli 1 zamieszczono wykaz podprogramów oraz informacje o warunkach badań i uruchomieniu podprogramów przy obsłudze ręcznej

i automatycznej układu sterowania robota.

Realizacja podprogramów użytkowych rozpoczyna się i kończy w położeniu zerowym części manipulacyjnej robota. Układ sterowania badanego robota przyłącza się do systemu pomiarowego SUM-R poprzez dwustawne układy we i wy (z separacją galwaniczną).

Wszystkie podprogramy są uruchamiane automatycznie przez mikrokomputer nadrzędny IBM-PC przez ustawienie odpowiedniej kombinacji sygnałów wejściowych. Po osiągnięciu położenia zadanego robot wyzwała pomiar ustawiając odpowiedni stan wyjścia 1 lub 5 układu sterowania.

4. PROGRAMY UŻYTKOWE DLA MINIKOMPUTERA IBM PC i MSA-80.3100

4.1. ZAŁOŻENIA

Przyjęto założenie, że oprogramowanie skomputeryzowanych stanowisk badawczych do badań robotów przemysłowych (SUM-R), służące do wyznaczania nowych charakterystyk robota zgodnie z wymaganiami normy ISO 9283, będzie stanowić wraz z istniejącymi już programami jednolity system pomiarowy.

W związku z tym przyjęto wszystkie założenia techniczne i programowe zgodne z opracowaniem zawartym w sprawozdaniu nr rej. 6473 (30.06.90) p.t. "Opracowanie założeń i programów użytkowych na badania z wykorzystaniem systemu MSA-80 i IBM-PC..." (Zad. 4.5), uwzględniając wymagania zawarte w opracowaniu "Adaptacja skomputeryzowanych stanowisk badawczych do badań robotów przemysłowych zgodnie z wymaganiami normy ISO/DIS 9283" et.1. "Opracowanie założeń technicznych na układy i programy użytkowe (dla układu sterowania robota mikrokomputera IBM i MSA) do wyznaczania nowych charakterystyk: dokładność pozycjonowania wielokierunkowego, ...".

4.2. ZAKRES PRACY

Wykonana praca obejmuje adaptację już istniejących programów systemu SUM-R oraz wykonanie nowych programów dla dodatkowych sprawdzeń.

Oprogramowanie dla komputera IBM wykonano w języku C (turbo C v.5.0) z nakładkami w języku Basic (GW) dla obsługi urządzenia pomiarowego firmy Hottinger.

Oprogramowanie komputera MSA zostało zmodernizowane tak, że programy obsługi poszczególnych urządzeń zostały zastąpione jednym programem MSA.BAS, którego poszczególne funkcje są wybierane odpowiednim sygnałem z komputera IBM.

Adaptacja dotyczy programów zarządzających systemem, w

szczegółności program MENU-2 został rozszerzony o nowe funkcje i obecnie umożliwia wyświetlanie na monitorze dwustronnicowego zestawu dostępnych sprawdzeń (z możliwością przechodzenia między stronicami w obu kierunkach).

Wykonano następujące nowe programy:

- S10.C wraz z nakładką HBM-S10.BAS obsługuje sprawdzenie dokładności pozycjonowania wielokierunkowego,
- S11.C wraz z nakładką HBM-S11.BAS obsługuje sprawdzenie dokładności i powtarzalności odtwarzania odcinka,
- S12.C wraz z nakładką HBM-S12.BAS obsługuje sprawdzenie czasu stabilizacji i przeregulowania położenia ,
- S14.C wraz z nakładką HBM-S14.BAS obsługuje sprawdzenie dryftu położenia,
- S15.C wraz z nakładką HBM-S15.BAS obsługuje sprawdzenie dokładności i powtarzalności odtwarzania torów o kształtach prostokątnym i kołowym,
- S16.C wraz z nakładką HBM-S16.BAS obsługuje sprawdzenie długości odcinka stabilizacji,
- S17.C obsługuje sprawdzenie dokładności i powtarzalności prędkości, współpracując z programem MSA.BAS kontrolującym pracę czujnika Ft-5.

4.3. WYDRUKI PROGRAMÓW ORAZ WYNIKI SPRAWDZEŃ CHARAKTERYSTYK.
(patrz załącznik).

5. WNIOSKI

Opracowane w ramach realizacji zlecenia K-28 metody, układy pomiarowe i osprzęt umożliwiają wyznaczenie charakterystyk badanego robota zgodnie z wymaganiami obowiązującej normy ISO 9283.

COBN*91

```
#include <conio.h>
#include <stdio.h>
#include <process.h>
#include <dir.h>
```

```
#define LW      11
#define Cz      0x70          /*czarne*/
#define DOWN    0x50          /* kursor w dol*/
#define UP      0x48          /*kursor w gore*/
#define BACK    0x4b          /*powrot do poprzedniego ekranu*/
#define CR      0xd
#define ESC     0x1b
```

```
static int xb,yb,xp;
static char *typ;
static char **tabnaz;
```

```
static char *tabnaz_1[]=
{
    "S1 Spr. przestrzeni roboczej",
    "S2 Spr. powtarzalnosci pozycjonowania",
    "S3 Spr. dokladnosci odtwarzania toru ruchu",
    "S4 Spr. sztywnosci",
    "S5 Spr. predkosci i przyspieszenia",
    "S6 Spr. czasu rozruchu i hamowania",
    "S7 Spr. napiecia pradnicy tachometrycznej ukladow napedowych",
    "S8 Spr. pradu silnikow napedowych",
    "S9 Spr. stalosci parametrow w czasie",
    "S10Spr. dokladnosci pozycjonowania wielokierunkowego",
    "B      nastepna strona ..."
};
```

```
static char *tabnaz_2[]=
{
    "A      ... poprzednia strona",
    "S11Spr. dokladnosci i powtarzalnosci odtwarzania odcinka",
    "S12Spr. czasu stabilizacji polozenia",
    "S13Spr. przeregulowania polozenia",
    "S14Spr. dryftu polozenia",
    "S15Spr. dokl. i powt. odtwarzania torow kolowych i prostokatnych",
    "S16Spr. dlugosci odcinka stabilizacji",
    "S17Spr. dokladnosci i powtarzalnosci predkosci",
    "SS Spr. srodowiska",
    "Sn      NAGLOWEK /informacje ogolne/",
    "E      KONIEC PRACY",
};
```

```
main()                                     /*** MENU-2.C ***/
{
```

```
char c;
```

```
int i,kier;
char cat[4];
```

7

```

tabnaz = tabnaz_1;
ekran();

sw:    if((c=getch())==NULL)
        switch(getch())
        {
            case DOWN:
                kier=1;
                kursor(kier);
                break;
            case UP:
                kier=-1;
                kursor(kier);
                break;
            case BACK:
                cs();
                chdir("\\tc5\\mol");
                execl("MENU-1", "MENU-1", NULL);
                break;
            default:
                goto sw;
        }
    else
        switch(c)
        {
            case CR:
                cs();
                for(i=0; i<3; i++)
                {
                    cat[i]=*(typ-3+i);
                }
                if(cat[0]=='E')
                    exit(0);
                if(cat[0]=='A')
                {
                    tabnaz = tabnaz_1;
                    ekran();
                    break;
                }
                if(cat[0]=='B')
                {
                    tabnaz = tabnaz_2;
                    ekran();
                    break;
                }

                cat[3]='\0';
                execl(cat, cat, NULL);
                printf("%s", cat);

            /*
                exit(0);
                break;
            case ESC:
                cs();
                exit(0);
                break;
            default:
                goto sw;
        }
    goto sw;
}

kursor(k)
int k;
{
    normvideo();
    cprintf("%-70s", typ);

    if(((yb==1)&&(k==-1)) || ((yb==LW)&&(k==1)))
        yb+=k+(-1)*k*LW;
}

```



```

        yb+=k;

        gotoxy(xp,yb);
        typ=(*(tabnaz+yb-1))+3;
        textattr(Cz);
        cprintf("%-70s",typ);
        gotoxy(xp,yb);
return;

}
cs()
{
window(1,1,80,25);
normvideo();
clrscr();
return;
}

/***** ekran() *****/
ekran()
{
int x1,y1,x2,y2;
int i;

clrscr();
printf(" * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *");
gotoxy(1,23);
printf("                                \21\55 - powrot do TYF ROBOTA\n");
printf("CR - potwierdzenie wyboru          ESC - przerwanie pracy\n");
printf("\30 - kursor w gore                \31 - kursor w dol\n");

xb=22;
yb=3;                                /*xb,yb - biezacy kursor*/
gotoxy(xb,yb);
highvideo();
cprintf(" ** SPRAWDZENIA ROBOTA  IRb(p) - 60 **");
yb++;
xb=22;
gotoxy(xb,yb);
cprintf("_____");
normvideo();
x1=5;
y1=8;                                /*x,y - okienko textowe*/
x2=75;
y2=y1+LW;
window(x1,y1,x2,y2);
xp=1;
yb=1;
xb=xp;
for (yb=1,i=0;yb<=LW;yb++,i++)
{
gotoxy(xb,yb);
typ=(*(tabnaz+i))+3;
cprintf("%-70s",typ);
}

yb=1;
gotoxy(xb,yb);
typ=(*(tabnaz))+3;
textattr(Cz);
cprintf("%-70s",typ);
gotoxy(xb,yb);
return;
}

/***** koniec ekran() *****/

```

```

#include <stdio.h>
#include <conio.h>
#include <process.h>
#include "carex.h"
#include <dos.h>

#define BACK 0x4b

main()
{
    int i;
    char *tekst[] =
    {
        "10.SPRAWDZENIE DOKLADNOSCI POZYCJONOWANIA WIELOKIERUNKOWEGO",
        " ",
        " - przygotowac robota zgodnie z DTR,",
        " - zainstalowac na laczu mechanicznym robota obciaznik",
        " wg ark. 3.1 i dlowice GP-TCP wg ark. 4.1,",
        " - zainstalowac na stanowisku pomiarowym IIa (patrz DTR",
        " nr rej. 6463 rys.1) glowice pomiarowa GP-3L wg ark.4.1,",
        " - polaczyc uk lad pomiarowy wg ark. 5.1.1 (bez czujnika Ft-5)",
        " - zsynchronizowac robota,",
        " - wczytac program uzytkowy z kasety dla uk ladu sterowania robota
(USR)",
        " - uruchomic podprogram POZ3 dla wybranego punktu (P1,P2,P3) w tr
ybie",
        " pracy KROKOWO (z panelu proramowania)",
        " - ustawic i wyzerowac wskazania czuników w p-kcie pomiarowm wg a
rk.1.1,",
        " - wykonac instrukcje ZAL WY 1 i sprawdzic poprawnosc rejestracji
,",
        " - wystartowac program USR w trybie AUTO,",
        " wcisnij klawisz",
    };

    printf("* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M
- R *");
    gotoxy(1,5);
    for(i=0;i<16;i++)
        printf("%s\n",tekst[i]);
    gotoxy(40,25);
    printf("\21\55 - powrot do menu SPRAWDZENIA");
    sw: if((getch())==NULL)
        switch(getch())
        {
            case BACK:
                clrscr();
                execl("MENU-2", "MENU-2", NULL);
                break;
            default:
                goto sw;
        }
    else

```

```

        clrscr();
        strona();
        system("POZ_S10.BAT"); /*wykonanie HBM-S10.BAS i MENU-2*/
        clrscr();
    }

}

/***** koniec main *****/
/*----- naglowek strony -----*/
strona()
{
    float Temp;
    FILE *fd;
    struct date Dzień;
    struct time Czas;

    if((fd=fopen(4,"w"))==NULL)
    {
        printf("Błąd drukarki !");
        exit(0);
    }

    getdate(&Dzień);
    gettimeofday(&Czas);
    Temp=carex(6);
    fprintf(fd,"\f* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * *  
S U M - R *\n");
    fprintf(fd,"Data spr. %d.%02d.%02d          Temp. %4.1f          Go  
dz. %d.%02d.%02d\n\n\n",\
            Dzień.da_year,Dzień.da_mon,Dzień.da_day,Temp,Czas.ti_hour,Czas.ti_min,Cz  
as.ti_sec);
    return;
}

float carex(nrk)

char nrk;

{
    unsigned pa,pb,kanal,K;
    int port_ab;
    int i;
    long proba=0;
    float temp;
    float Sens=20.48;                /* czulosć div/stop.C */

    nrk=nrk-1;                        /*nr kanalu */
    K=TEMP!nrk;

    outportb(PCW,T3);                 /* ustaw tryb T3 , zeruj */
    outportb(PC,K);                   /*wyslij do PC symbol prog. 0x4nrk t.j. uruchom  
.MSA*/

next:
while((!inportb(PA))&&(!inportb(PB))) /*oczekiwanie na pomiar */
;
pa=inportb(PA);                      /* odczyt portow A i B */
pb=inportb(PB);
port_ab=((pb&M)<<8)!pa;               /* sklejenie bitow B+A */
kanal=pb>>4;                         /* wyodrebnienie nr kanalu */
if(kanal!=nrk)
{
    if(proba<100000)
    {
        proba++;
        goto next;
    }

else
{

```

```
return NULL;
}

}
outportb(PC, 0x0);          /* stop dla MSA */
temp=(port_ab-ZERO)/Sens;    /* przeliczenie na stopnie C */

for(i=1; i<15000; i++)      /* opoznienie */
;
return temp;
}
```

```

10 10 REM***** HBM-S10 (3 czujniki) *****
20 MASK=&H10
100 PCW=&H13E3:PA=&H13E0:PB=&H13E1:PC=&H13E2
101 OUT PCW,&H92
250 CLS
260 PRINT"*** DOKLADNOSC POZYCJONOWANIA (WIELOKIERUNKOWA) ***":PRINT:PRINT
270 PRINT"U W A G A ! - STOP POMIARU = dowolny klawisz (max.30 cykli pom.)"
310 LPRINT"*** DOKLADNOSC POZYCJONOWANIA (WIELOKIERUNKOWA) ***"
340 LPRINT:LPRINT
350 REM *****
360 IC=3
370 L=0
380 IP=90:REM ***** 30 CYKLI PO 3 POM. (Z KIER. X,Y,Z )kazdy pomiar odbywa
    sie 3 czujnikami *****
390 DIM WRD$(IP)
400 OPEN "COM1:9600,E,7,2" AS #1:ON ERROR GOTO 830
410 PRINT #1,CHR$(18);
420 PRINT #1,"CPB78;TEX*"
430 PRINT #1,"IDY"
445 LINE INPUT#1,A$:PRINT A$
450 PRINT"PODAJ NR PUNKTU POMIAROWEGO : "
452 PRINT "0 - punkt pomiarowy P1"
454 PRINT "1 - punkt pomiarowy P2"
456 PRINT "2 - punkt pomiarowy P4"
460 INPUT PP
465 REM **** HEX A = 'DO ROBOTA ' , HEX E = 'PROGR. POZ3' *****
470 MSA=&HAE-PP
480 OUT PC,MSA
490 ROB=INP(PA)
495 IF INKEY$<>"" THEN GOTO 570
500 IF ((ROB AND MASK)<>MASK) GOTO 490
505 OUT PC,&H0
510 PRINT #1,"MSV 12,14,0"
520 LINE INPUT#1,A$:PRINT "Pomiar nr : ",L+1:PRINT A$
525 FOR I=1 TO 78:PRINT"-";:NEXT I:PRINT"- "
530 WRD$(L+1)=A$
540 L=L+1
550 IF INKEY$="" THEN GOTO 560 ELSE 570
560 IF L<=IP THEN GOTO 480
570 IP1=L
575 OUT PC,&H0
580 REM*****KONIEC POMIAROW*****
590 DIM WART(IC-1,IP1)
600 FOR Y=1 TO IP1
610 FOR X=0 TO IC-1
620 WART(X,Y)=VAL(MID$(WRD$(Y),(6+(X*15)),10))
630 NEXT X
640 NEXT Y
1240 REM*****
1300 REM***** OBLICZ R *****
1305 REM----- OBLICZ SREDNIA -----
1310 DIM WS(2,3)
1320 FOR X=0 TO IC-1
1325 FOR I=1 TO 3
1330 FOR Y=1 TO IP1 STEP 3
1350 WS(X,I)=WS(X,I)+WART(X,Y)
1370 NEXT Y
1375 WS(X,I)=WS(X,I)/(INT(IP1/3))
1378 NEXT I
1380 NEXT X

```

```

1410 LPRINT "WSPOLRZEDNE WEKTOROW DOKLADNOSCI POZYCJONOWANIA WYNOSZA : "
1420 LPRINT "    AP_1 ("; WS(0,1); WS(1,1); WS(2,1); ")"
1430 LPRINT "    AP_2 ("; WS(0,2); WS(1,2); WS(2,2); ")"
1440 LPRINT "    AP_3 ("; WS(0,3); WS(1,3); WS(2,3); ")"
1500 REM-----
1520 REM----- OBLICZ MAX vAP -----
1530 DIM AP_TAB(2)
1550 AP_TAB(0) = SQR(ABS( (WS(0,1)-WS(0,2))^2 + (WS(1,1)-WS(1,2))^2 + (WS(2,1)-W
S(2,2))^2)
1560 AP_TAB(1) = SQR(ABS( (WS(0,2)-WS(0,3))^2 + (WS(1,2)-WS(1,3))^2 + (WS(2,2)-W
S(2,3))^2)
1570 AP_TAB(2) = SQR(ABS( (WS(0,3)-WS(0,1))^2 + (WS(1,3)-WS(1,1))^2 + (WS(2,3)-W
S(2,1))^2)
1600 REM----- MAX AP_i -----
1605 MAX=AP_TAB(0)
1610 FOR I=0 TO 2
1620 IF MAX < AP_TAB(I) THEN MAX = AP_TAB(I)
1630 NEXT I
1640 LPRINT "WIELOKIERUNKOWA DOKLADNOSC POZYCJONOWANIA WYNOSI : "
1650 LPRINT "          vAP = "; MAX
1900 SYSTEM

```

10 k1 *** DOKLADNOSC POZYCJONOWANIA (WIELOKIERUNKOWA) ***

WSPOLRZEDNE X , Y , Z WEKTOROW DOKLADNOSCI POZYCJONOWANIA WYNOSZA :

AP_1 (-3.9972 -1.4112 -5.2128)

AP_2 (-3.9972 -1.4112 -5.2128)

AP_3 (-3.9972 -1.4112 -5.2128)

WIELOKIERUNKOWA DOKLADNOSC POZYCJONOWANIA WYNOSI :

vAF = 0

L P		X	Y	Z
		kanal 12	kanal 13	kanal 14
1	k1	-3.331	-1.176	-4.344
	k2	-3.331	-1.176	-4.344
	k3	-3.331	-1.176	-4.344
4	k1	-3.331	-1.176	-4.344
	k2	-3.331	-1.176	-4.344
	k3	-3.331	-1.176	-4.344
7	k1	-3.331	-1.176	-4.344
	k2	-3.331	-1.176	-4.344
	k3	-3.331	-1.176	-4.344
10	k1	-3.331	-1.176	-4.344
	k2	-3.331	-1.176	-4.344
	k3	-3.331	-1.176	-4.344
13	k1	-3.331	-1.176	-4.344
	k2	-3.331	-1.176	-4.344
	k3	-3.331	-1.176	-4.344
16	k1	-3.331	-1.176	-4.344
	k2	-3.331	-1.176	-4.344


```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include "carex.h"
#include <dos.h>
```

```
#define BACK 0x4b
```

```
main() /* ** S11 ** */
{
    int i;
    char *tekst[]=
    {
        "11.SPRAWDZENIE DOKŁADNOŚCI I POWTARZALNOŚCI ODTWARZANIA ODCINKA",
        " ",
        " - przygotować robota zgodnie z DTR,",
        " - zainstalować na łączu mechanicznym robota obciążnik",
        " wg ark. 3.1 i dłowice GP-TCP wg ark. 4.1,",
        " - zainstalować na dwóch stanowiskach pomiarowych IIa (patrz DTR"
        " nr rej. 6463 rys.1) głowice pomiarowa GP-3L wg ark.4.1,",
        " - połączyć układ pomiarowy wg ark. 5.1.1 (bez czujnika Ft-5)",
        " - zsynchronizować robota,",
        " - wczytać program użytkowy z kasety dla układu sterowania robota
        (USR)",
        " - uruchomić podprogram ODC dla wybranego punktu (P1,P2,P3) w try
        bie",
        " pracy KROKOWO (z panelu programowania)",
        " - ustawić i wyzerować wskazania czuników w p-kcie P2 i P4 wg ark
        .1.1,",
        " - wykonać instrukcje ZAL WY 1 i sprawdzić poprawność rejestracji
        ",
        " - wystartować program USR w trybie AUTO,",
        " wcisnij klawisz",
    };
```

```
printf("* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M
- R *");
gotoxy(1,5);
for(i=0;i<16;i++)
    printf("%s\n",tekst[i]);
gotoxy(40,25);
printf("\21\55 - powrót do menu SPRAWDZENIA");
sw: if((getch())==NULL)
    switch(getch())
    {
        case BACK:
            clrscr();
            execl("MENU-2", "MENU-2", NULL);
            break;
        default:
            goto sw;
```

```

else
{
clrscr();
strona();
system("ODC_S11.BAT"); /*wykonanie HBM-S11.BAS i MENU-2*/
clrscr();
}

}

/***** koniec main *****/
/*----- naglowek strony -----*/
strona()
{
float Temp;
FILE *fd;
struct date Dzień;
struct time Czas;

if((fd=fopen(4,"w"))==NULL)
{
printf("Bład drukarki !");
exit(0);
}

getdate(&Dzien);
gettext(&Czas);
Temp=carex(6);
fprintf(fd,"\f* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * *  
S U M - R *\n");
fprintf(fd,"Data spr. %d.%02d.%02d Temp. %4.1f Go  
dz. %d:%02d:%02d\n\n\n",\  
Dzien.da_year,Dzien.da_mon,Dzien.da_day,Temp,Czas.ti_hour,Czas.ti_min,Cz  
as.ti_sec);
return;
}

float carex(nrk)

char nrk;

{
unsigned pa,pb,kanal,K;
int port_ab;
int i;
long proba=0;
float temp;
float Sens=20.48; /* czulosc div/stop.C */

nrk=nrk-1;
K=TEMP!nrk; /*nr kanalu */

outportb(PCW,T3); /* ustaw tryb T3 , zeruj */
outportb(PC,K); /*wyslij do PC symbol prog. 0x4nrk t.j. uruchom  
.MSA*/

```

```

10 REM***** HBM-S11 (2 X 3 czujniki) *****
20 MASK=&H10
30 PCW=&H13E3:PA=&H13E0:PB=&H13E1:PC=&H13E2
40 OUT PCW,&H92
50 CLS
60 PRINT"*** DOKLADNOSC I POWTARZALNOSC ODTWARZANIA ODCINKA ***":PRINT:PRINT
70 PRINT"U W A G A ! - STOP POMIARU = dowolny klawisz (max.30 cykli pom.)"
80 LPRINT"*** DOKLADNOSC I POWTARZALNOSC ODTWARZANIA ODCINKA ***"
90 LPRINT:LPRINT
100 REM *****
110 IC=3
120 L=0
130 IP=30:REM ***** 30 CYKLI x DWIE GLOWICE kazdy pomiar odbywa sie 3 czuj
nikami *****
140 DIM WRD$(IP)
150 DIM WRD2$(IP)
160 OPEN "COM1:9600,E,7,2" AS #1:ON ERROR GOTO 1000
170 PRINT #1,CHR$(18);
180 PRINT #1,"CFB7B;TEX*"
190 PRINT #1,"IDY"
200 LINE INPUT#1,A$:PRINT A$
210 PRINT"PUNKT POMIAROWY P2 - P4"
220 REM **** HEX A = 'DO ROBOTA ' , HEX B = 'PROGR. ODC' *****
230 MSA=&HAB
240 OUT PC,MSA
250 ROB=INP(PA)
260 IF INKEY$<>" " THEN GOTO 450
270 IF ((ROB AND MASK)<>MASK) GOTO 250
280 OUT PC,&H0
290 PRINT #1,"MSV 12,14,0"
300 LINE INPUT#1,A$:PRINT "Pomiar w P2 nr :",L+1:PRINT A$
310 FOR I=1 TO 78:PRINT "-";:NEXT I:PRINT "-"
320 WRD$(L+1)=A$
325 REM ***** P2 / P4 *****
330 OUT PC,MSA
340 ROB=INP(PA)
350 IF INKEY$<>" " THEN GOTO 450
360 IF ((ROB AND MASK)<>MASK) GOTO 340
370 OUT PC,&H0
380 PRINT #1,"MSV 15,17,0"
390 LINE INPUT#1,A$:PRINT "Pomiar w P4 nr :",L+1:PRINT A$
400 FOR I=1 TO 78:PRINT "-";:NEXT I:PRINT "-"
410 WRD2$(L+1)=A$
420 L=L+1
430 IF INKEY$="" THEN GOTO 440 ELSE 450
440 IF L<=IP THEN GOTO 240
450 IP1=L
460 OUT PC,&H0
470 REM*****KONIEC POMIAROW*****
480 DIM WART(IC-1,IP1)
490 FOR Y=1 TO IP1
500 FOR X=0 TO IC-1
510 WART(X,Y)=VAL(MID$(WRD$(Y),(6+(X*15)),10))
520 NEXT X
530 NEXT Y
540 REM*****
541 DIM WART2(IC-1,IP1)
542 FOR Y=1 TO IP1
543 FOR X=0 TO IC-1
545 WART2(X,Y)=VAL(MID$(WRD2$(Y),(6+(X*15)),10))
547 NEXT X

```

```

550 REM***** OBLICZ R *****
560 REM----- OBLICZ SREDNIA -----
570 FOR Y=1 TO IP1
580 ADX=(ADX+(WART(O,Y)+WART2(O,Y)))
590 NEXT Y
595 ADX=ADX/IP1
598 PRINT "ADx = ";ADX
600 FOR Y=1 TO IP1
610 ADY=(ADY+(WART(1,Y)+WART2(1,Y)))
620 NEXT Y
630 ADY=ADY/IP1
635 PRINT "ADy = ";ADY
650 FOR Y=1 TO IP1
660 ADZ=(ADZ+(WART(2,Y)+WART2(2,Y)))
670 NEXT Y
680 ADZ=ADZ/IP1
685 PRINT "ADz = ";ADZ
690 ADD=SQR((ADX^2)+(ADY^2)+(ADZ^2))
695 PRINT "AD_ = ";ADD
700 LPRINT "DOKLADNOSC ODTWARZANIA ODCINKA WYNOSI AD = ";ADD
720 LPRINT "                w osi X          ADx = ";ADX
730 LPRINT "                w osi Y          ADy = ";ADY
740 LPRINT "                w osi Z          ADz = ";ADZ
745 LPRINT :LPRINT :LPRINT :LPRINT
750 REM *****
760 PRINT "*** POMIAR POWTARZALNOSCI ODTWARZANIA ODCINKA ***"
770 FOR Y=1 TO IP1
780 RDX=RDX+((WART(O,Y)+WART2(O,Y)-ADX)^2)
790 NEXT Y
800 RDX=3*SQR(RDX/(IP1-1))
810 PRINT "RDx = +-" ;RDX
815 REM -----
820 FOR Y=1 TO IP1
830 RDY=RDY+((WART(1,Y)+WART2(1,Y)-ADY)^2)
840 NEXT Y
850 RDY=3*SQR(RDY/(IP1-1))
860 PRINT "RDy = +-" ;RDY
865 REM -----
870 FOR Y=1 TO IP1
880 RDZ=RDZ+((WART(2,Y)+WART2(2,Y)-ADZ)^2)
890 NEXT Y
900 RDZ=3*SQR(RDZ/(IP1-1))
910 PRINT "RDz = +-" ;RDZ
915 REM -----
920 FOR Y=1 TO IP1
930 DDJ=DDJ+SQR(((WART(O,Y)+WART2(O,Y))^2)+((WART(1,Y)+WART2(1,Y))^2)+((WART(2,Y)+WART2(2,Y))^2))
940 RDD=RDD+(DDJ-ADD)^2
950 NEXT Y
960 RDD=3*SQR(RDD/(IP1-1))
970 PRINT "RD_ = +-" ;RDD
980 LPRINT "POWTARZALNOSC ODTWARZANIA ODCINKA WYNOSI RD = ";RDD
982 LPRINT "                w osi X          RDx = ";RDX
984 LPRINT "                w osi Y          RDy = ";RDY
986 LPRINT "                w osi Z          RDz = ";RDZ
1000 SYSTEM

```

*** DOKLADNOSC I POWTARZALNOSC ODTWARZANIA ODCINKA ***

DOKLADNOSC ODTWARZANIA ODCINKA WYNOŚI: AD = 4.330457

w osi X ADx = -3.318001

w osi Y ADy = -1.1843

w osi Z ADz = -2.518166

POWTARZALNOSC ODTWARZANIA ODCINKA WYNOŚI: RD = 88.11032

w osi X RDx = 1.494121E-06

w osi Y RDy = 3.735302E-07

w osi Z RDz = 1.732057E-03

L	P	X	Y	Z
		KANAL 12/15	KANAL 13/16	KANAL 14/17
1	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.826
2	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.826
3	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.826
4	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.826
5	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.827
6	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.826
7	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.826
8	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.825
9	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.825
10	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.826
11	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.826
12	P2	-3.331	-1.176	-4.344
	P4	0.013	-0.008	1.825

```

#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <fcntl.h>
#include <graphics.h>
#include <dos.h>
#include "carex.h"

#define BACK      0x4b

char *wc;
FILE *fp;
FILE *fd;

cz_zb_d();
float carex();

float wynik[16];
float czas[16];

main()
{
double pix_x=10
    ,pix_y=10;          /* wartosc parametru na 1 pixel */
int yop=47,
    yok=347,
    xop=60,
    xok=660,
    xoy;                /* wspolrzedne osi x i y oraz zero ukladu */

int Driv=DETECT;
int Mod;
int w;

int ,xx,yy;            /* wspolrzedne wykresu */

int i;

char *tekst[]=
{
    "12.SPRAWDZENIE CZASU STABILIZACJI I PRZEREGULOWANIA.",
    "    - przygotowac robota zgodnie z DTR,",
    "    - zainstalowac na zlaczu mechanicznym obciaznik wg ark.3.1,",
    "    - zainstalowac na stanowisku pomiarowym IIa glowice GP-1L wg.ark.4.1,",
    "    - polaczyc uklad pomiarowy wg ark. 5.1.1 (bez czujnika Ft-5),",
    "    - zsynchronizowac robota,",
    "    - wczytac program uzytkowy z kasety dla ukladu sterowania robota(USR)",
    "    - uruchomic z panelu programowania robota podprogram FOL (KROKOWO)",
    "    - ustawic i wyzerowac wskazania czujnika w wybranym punkcie ",
    "    - (P1,P2,P3,P4,P5) wg ark1.1,",
    "obsługa :      * zakonczenie pomiaru          - dowolny klawisz,",
    "              * ustawienie kursora A          - strzalka lewo,pravo,",
    "              * wydruk (zakonczenie progr.) - klawisz End",
    "              nacisnij klawisz"
};

/*=====*/

clrscr();
printf("% S'U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *");
gotoxy(1,5);
for(i=0;i<13;i++)
    printf("%s\n",tekst[i]);

gotoxy(40,25);
printf("\21\55 - powrot do menu SPRAWDZENIA");
sw:    if((getch())==NULL)
        switch(getch())
        {
            case BACK:
                clrscr();
                execl("MENU-2", "MENU-2", NULL);
                break;
            default:
                goto sw;
        }
    else
    {
        clrscr();
        strona();
        system("STAB_S12.BAT"); /*wykonanie GWBASIC HBM_S12 i MENU-2*/
        exit(1);
        clrscr();
    }

cz_zb_d();

while(!kbhit());
getch();

pix_x=600/(fabs(maxi(czas)));
pix_y=300/(fabs(maxi(wynik)-mini(wynik)));

```

```

xoy=(yop+floor(maxi(wynik)/pix_y));
xoy=(xoy < 347) ? xoy : 347;

/***** GRAFIKA *****/
initgraph(&Driv,&Mod,"");
if((w=graphresult())!=grOk)
{
    printf("\nBLAD KARY GRAFICZNEJ... %d\n",w);
    execl("MENU-2","MENU-2",NULL);
    exit(0);
}

line(xop,yop,xop,yok);
line(xop,xoy,xok,xoy);          /* osie wykresu */
moveto(xop,xoy);
for(i=1;i<=wynik[0];i++)
{
    xx=xoy+floor(wynik[i]/pix_y);
    yy=floor(czas[i]/pix_x);
    lineto(xx,yy);
}
while(!kbhit());
getch();
closegraph();

}
/***** czytaj zb. dyskowy *****/

cz_zb_d()
{

int i;
printf("WYNIKI POMIAROW \n");

fp=fopen("WYNIKI.S12","rt");

wynik[0]=atof(fgets(wc,15,fp));
czas[0]=atof(fgets(wc,15,fp));

for(i=1;i<16;i++)
{
    wynik[i]=atof(fgets(wc,15,fp));
    czas[i]=atof(fgets(wc,15,fp));
    czas[i]=czas[i]-czas[0];
}
for(i=0;i<16;i++)
{
    printf("%d   %7.3f   %7.3f\n",i,wynik[i],czas[i]);
}

fclose(fp);
return;
}

/*****~::~::~::~::~::~::~::~::~::~::~::~::~::~::~::~*****/
maxi(tabela)

float tabela[];
{
int i;
float max=tabela[1];

for(i=1;i<=tabela[0];i++)
{
    max=(max > tabela[i]) ? max : tabela[i];
}

return(max);
}
/*****~::~::~::~::~::~::~::~::~::~::~::~::~::~::~::~*****/

```



```

    }
    else
    {
        outportb(PC,0x0);          /* stop dla MSA */
        return NULL;
    }
}
outportb(PC,0x0);                /* stop dla MSA */
temp=(port_ab-ZERO)/Sens;        /* przeliczenie na stopnie C */

for(i=1;i<15000;i++)            /* opoznienie */
;
return temp;
}

/*****/

```



```

505 OUT PC,&H0
510 PRINT #1,"MSV 12,,0"
515 VTP(L+1)=TIMER
520 LINE INPUT#1,A$:PRINT "Pomiar nr :",L+1:PRINT A$
525 FOR I=1 TO 78:PRINT"-";:NEXT I:PRINT"- "
530 WRD$(L+1)=A$
540 L=L+1
550 IF INKEY$="" THEN GOTO 560 ELSE 570
560 IF L<IP THEN GOTO 510:REM ***** PO SYG. OD ROBOTA - 15 POMIAROW **
570 IP1=L
575 OUT PC,&H0
580 REM*****KONIEC POMIAROW*****
600 FOR Y=1 TO IP1
620 WART(Y)=WART(Y)+VAL(MID$(WRD$(Y),6,10))
640 NEXT Y
643 L=0
645 NEXT CYKL
650 REM***** KONIEC PETLI CYKLE*****
660 FOR Y=1 TO IP1
670 WART(Y)=WART(Y)/3
680 NEXT Y
800 OPEN "WYNIKI.S12" FOR OUTPUT AS #3
810 WART(0)=IP1
820 FOR I=0 TO IP1
840 PRINT #3,WART(I)
850 PRINT #3,VTP(I)
860 NEXT I
900 CLOSE #3
950 LPRINT "PRZEREGULOWANIE      CZAS"
955 LPRINT "      /mm/      /s/":LPRINT
1000 FOR I=1 TO IP1
1010 LPRINT USING "      ###.###      ";WART(I);VTP(I)-VTP(1)
1020 NEXT I
2000 END

```

* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *
Data spr. 1991.12.30 Temp. 0.0 Godz. 13:23:28

*** POMIAR CZASU STABILIZACJI I PRZEREGULOWANIA POLOZENIA ***

OBCIAZENIE ROBOTA :100%
PRĘDKOŚĆ ROBOTA :50%

PRZEREGULOWANIE /mm/	CZAS /s/
-3.349	0.000
-3.349	0.602
-3.349	1.199
-3.349	1.809
-3.349	2.410
-3.349	3.020
-3.349	3.621
-3.349	4.230
-3.349	4.832
-3.349	5.379
-3.349	5.980
-3.349	6.590
-3.349	7.191
-3.349	7.801
-3.349	8.402

```

#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <fcntl.h>
#include <graphics.h>
#include <dos.h>
#include "carex.h"

#define BACK 0x4b

char *wc;
FILE *fp;
FILE *fd;

cz_zb_d();
float carex();

float wynik[16];
float czas[16];

main()
{
double pix_x=10
    ,pix_y=10;    /* wartosc parametru na 1 pixel */
int yop=47,
    yok=347,
    xop=60,
    xok=660,
    xoy;    /* wspolrzedne osi x i y oraz zero ukladu */

int Driv=DETECT;
int Mod;
int w;

int xx,yy;    /* wspolrzedne wykresu */

int i;

char *tekst[]=
{
    "13.SPRAWDZENIE CZASU STABILIZACJI I PRZEREGULOWANIA.",
    "    - przygotowac robota zgodnie z DTR,"

```

```

"      - zainstalować na stanowisku pomiarowym IIa głowice GP-1L wg.ark.4.1,"
"      - połączyć układ pomiarowy wg ark. 5.1.1 (bez czujnika Ft-5),,"
"      - zsynchronizować robota,"
"      - wczytać program użytkowy z kasety dla układu sterowania robota(USR)",
"      - uruchomić z panelu programowania robota podprogram POL (KROKOWO),"
"      - ustawić i wyzerować wskazania czujnika w wybranym punkcie ",
"      - (P1,P2,P3,P4,P5) wg ark1.1,"
"obsługa :      * zakończenie pomiaru      - dowolny klawisz,"
"              * ustawienie kursora A      - strzałka lewo,pravo,"
"              * wydruk (zakończenie progr.) - klawisz End",
"              nacisnij klawisz"
};

```

```

/*=====*/

```

```

clrscr();
printf(" * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * ");
gotoxy(1,5);
for(i=0;i<13;i++)
    printf("%s\n",tekst[i]);

```

```

gotoxy(40,25);
printf("\21\55 - powrot do menu SPRAWDZENIA");
sw:    if((getch())==NULL)
        switch(getch())
        {
            case BACK:
                clrscr();
                execl("MENU-2","MENU-2",NULL);
                break;
            default:
                goto sw;
        }
    else
    {
        clrscr();
        strona();
        system("STAB_S12.BAT"); /*wykonanie GWBASIC HBM_S12 i MENU-2*/
        exit(1);
        clrscr();
    }

```

```

cz_zb_d();

```

```

while(!kbhit());
getch();

```

```

pix_x=600/(fabs(maxi(czas)));
pix_y=300/(fabs(maxi(wynik)-mini(wynik)));

```

```

printf("%f    %f\n",pix_x,pix_y);    /*-----*/

```

```

xoy=(yop+floor(maxi(wynik)/pix_y));
xoy=(xoy < 347) ? xoy : 347;

```

```

/***** GRAFIKA *****/
initgraph(&Driv,&Mod,"");
if((w=graphresult())!=grOk)
{
    printf("\nBLAD KARY GRAFICZNEJ... Zd\n",w);
    execl("MENU-2","MENU-2",NULL);
    exit(0);
}

```

```

line(xop,yop,xop,yok);
line(xop,xoy,xok,xoy);    /* osie wykresu */
moveto(xop,xoy);
for(i=1;i<=wynik[0];i++)
{

```

```

        yy=floor(czas[i]/pix_x);
        lineto(xx,yy);
    }
    while(!kbhit());
    getch();
    closegraph();

}
/***** czytaj zb. dyskowy *****/

cz_zb_d()
{

    int i;
    printf("WYNIKI POMIAROW \n");

    fp=fopen("WYNIKI.S12","rt");

    wynik[0]=atof(fgets(wc,15,fp));
    czas[0]=atof(fgets(wc,15,fp));

    for(i=1;i<16;i++)
    {
        wynik[i]=atof(fgets(wc,15,fp));
        czas[i]=atof(fgets(wc,15,fp));
        czas[i]=czas[i]-czas[0];
    }
    for(i=0;i<16;i++)
    {
        printf("%d %7.3f %7.3f\n",i,wynik[i],czas[i]);
    }

    fclose(fp);
    return;
}

/*****~~~~~*****/
maxi(tabela)

float tabela[];
{
    int i;
    float max=tabela[1];

    for(i=1;i<=tabela[0];i++)
    {
        max=(max > tabela[i]) ? max : tabela[i];
    }

    return(max);
}
/*****~~~~~*****/

mini(tabela)

float tabela[];

{
    int i;
    float min=tabela[1];

    for(i=1;i<=tabela[0];i++)
    {
        min=(min < tabela[i]) ? min : tabela[i];
    }

    return(min);
}
/*****~~~~~*****/

strona()

```



```

float Temp;
struct date Dzień;
struct time Czas;

if((fd=fopen(4,"w"))==NULL)
{
    printf("Błąd drukarki !");
    exit(0);
}
getdate(&Dzień);
gettime(&Czas);
Temp=carex(6);
fprintf(fd,"\\f\\ S U M - R \\f\\ S U M - R \\f\\ S U M - R \\f\\ S U M - R \\f\\ S U M - R \\f\\n");
fprintf(fd,"Data spr. %d.%02d.%02d          Temp. %4.1f          Godz. %d:%02d:%02d\\n\\n\\n",\\
        Dzień.da_year,Dzień.da_mon,Dzień.da_day,Temp,Czas.ti_hour,Czas.ti_min,Czas.ti_sec);
return;
}

```

/*****/

```
float carex(nrk)
```

```
char nrk;
```

```

{
    unsigned pa,pb,kanal,K;
    int port_ab;
    int i;
    long proba=0;
    float temp;
    float Sens=20.48;          /* czułość div/stop.C */

    nrk=nrk-1;
    K=TEMP*nrk;                /* nr kanału */

    outportb(PCW,T3);          /* ustaw tryb T3 , zeruj */
    outportb(PC,K);            /* wyslij do PC symbol prog. 0x4nrk t.j. uruchom.MSA */

```

```

next:
while((!inportb(PA))&&(!inportb(PB))) /* oczekiwanie na pomiar */
;
pa=inportb(PA);                /* odczyt portów A i B */
pb=inportb(PB);
port_ab=((pb&M)<<8)!pa;        /* sklejenie bitów B+A */
kanal=pb>>4;                  /* wyodrębnienie nr kanału */
if(kanal!=nrk)
{
    if(proba<100000)
    {
        proba++;
        goto next;
    }

    else
    {
        outportb(PC,0x0);      /* stop dla MSA */
        return NULL;
    }
}

outportb(PC,0x0);              /* stop dla MSA */
temp=(port_ab-ZERO)/Sens;      /* przeliczenie na stopnie C */

for(i=1;i<15000;i++)          /* opóźnienie */
;
return temp;
}

```

/*****/

```

10 REM***** HBM-S13 (1 czujnik) *****
20 MASK=&H10
100 PCW=&H13E3:PA=&H13E0:PB=&H13E1:PC=&H13E2
101 OUT PCW,&H92
250 CLS
260 PRINT"*** POMIAR CZASU STABILIZACJI I PRZEREGULOWANIA POLOZENIA ***":PRINT:P
RINT
270 PRINT"U W A G A !      -      STOP POMIARU = dowolny klawisz (max.150 pomiarow)"
310 LPRINT"*** POMIAR CZASU STABILIZACJI I PRZEREGULOWANIA POLOZENIA ***"
340 LPRINT:LPRINT
345 INPUT"PODAJ WARTOSC OCIAZENIA : ",OB$:LPRINT "OBCIAZENIE ROBOTA : ";OB$
347 INPUT "PODAJ PREDKOSC ROBOTA : ",PR$:LPRINT "PREDKOSC ROBOTA   : ";PR$
350 REM *****
355 LPRINT :LPRINT
360 IC=1
370 L=0
380 IP=15
385 DIM WART(IP)
390 DIM WRD$(IP)
392 TIMER ON
394 DIM VTP(IP)
400 OPEN "COM1:9600,E,7,2" AS #1:ON ERROR GOTO 2000
410 PRINT #1,CHR$(18);
420 PRINT #1,"CPB7B;TEX*"
430 PRINT #1,"IDY"
445 LINE INPUT#1,A$:PRINT A$
450 PRINT"PODAJ NR PUNKTU POMIAROWEGO : "
451 PRINT "      1 - punkt pomiarowy P1"
452 PRINT "      2 - punkt pomiarowy P2"
453 PRINT "      3 - punkt pomiarowy P3"
455 PRINT "      4 - punkt pomiarowy P4"
456 PRINT "      5 - punkt pomiarowy P5"
460 INPUT PP
470 MSA=&HAA-PP+1
473 REM ***** 3 CYKLE *****
475 FOR CYKL=1 TO 3
480 OUT PC,MSA
490 ROB=INP(PA)
495 IF INKEY$<>" " THEN GOTO 570
500 IF ((ROB AND MASK)<>MASK) GOTO 490
505 OUT PC,&H0
510 PRINT #1,"MSV 12,,0"
515 VTP(L+1)=TIMER
520 LINE INPUT#1,A$:PRINT "Pomiar nr : ",L+1:PRINT  A$
525 FOR I=1 TO 78:PRINT"-";:NEXT I:PRINT"- "
530 WRD$(L+1)=A$
540 L=L+1
550 IF INKEY$="" THEN GOTO 560 ELSE 570
560 IF L<IP THEN GOTO 510:REM ***** PO SYG. OD ROBOTA - 15 POMIAROW **
570 IP=L
575 OUT PC,&H0
580 REM*****KONIEC POMIAROW*****
600 FOR Y=1 TO IP1
620 WART(Y)=WART(Y)+VAL(MID$(WRD$(Y),6,10))
640 NEXT Y
643 L=0

```

```

640 NEXT Y
650 REM***** KONIEC PETLI CYKLE*****
660 FOR Y=1 TO IP1
670 WART(Y)=WART(Y)/3
680 NEXT Y
800 OPEN "WYNIKI.S12" FOR OUTPUT AS #3
810 WART(0)=IP1
820 FOR I=0 TO IP1
840 PRINT #3,WART(I)
850 PRINT #3,VTP(I)
860 NEXT I
900 CLOSE #3
950 LPRINT "PRZEREGULOWANIE      CZAS"
955 LPRINT "      /mm/      /s/";LPRINT
1000 FOR I=1 TO IP1
1010 LPRINT USING "      ###.###      ";WART(I);VTP(I)-VTP(1)
1020 NEXT I
2000 END

```

* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *
Data spr. 1991.12.30 Temp. 0.0 Godz. 13:30:51

*** POMIAR CZASU STABILIZACJI I PRZEREGULOWANIA POLOZENIA ***

OBCIAZENIE ROBOTA :77
PRĘDKOŚĆ ROBOTA :88

PRZEREGULOWANIE /mm/	CZAS /s/
-3.349	0.000
-3.349	0.602
-3.349	1.211
-3.349	1.813
-3.349	2.410
-3.349	3.020
-3.349	3.621
-3.349	4.230
-3.349	4.781
-3.349	5.383
-3.349	5.980
-3.349	6.590
-3.349	7.191
-3.349	7.801
-3.349	8.402

```

#include <stdio.h>
#include <conio.h>
#include <process.h>
#include "carex.h"
#include <dos.h>

```

```

#define BACK 0x4b

```

```

main()
{
char c;
int i;
char *tekst[]=
{
    "14.SPRAWDZENIE DRYFTU POLOZENIA.",
    "",
    "    - przygotowac robota zgodnie z DTR,",
    "    - zainstalowac stanowisko pomiarowe IIa lub IIb (patrz DTR",
    "      nr rej.6463 rys.1) w strefie roboczej robota,",
    "    - zainstalowac glowice pomiarowa GP-3L (lub GP-1L) wg ark.4.1,",
    "    - dla glowicy GP-3L na koncowce kolnierzowej robota ",
    "      zainstalowac glowice GP-TCP wg ark.4.1,",
    "    - polaczyc uk lad pomiarowy wg ark.5.1,",
    "    - zsynchronizowac robota,",
    "    - wczytac program uzytkowy z kasety dla uk ladu sterowania robota(USR)",
    "    - uruchomic podprogram POZ1(lub POZ2.1 lub POZ2.2) w trybie",
    "      pracy KROKOWO symulujac z panelu programowania odp.wejsca,",
    "    - ustawic i wyzerowac wskazania czunikow w p-ktach pomiarowych,",
    "    - wystartowac program USR w trybie AUTO,",
    "      wcisnij klawisz",
};

```

```

gotoxy(1,5);
for(i=0;i<16;i++)
    printf("%s\n",tekst[i]);
gotoxy(40,25);
printf("\21\55 - powrot do menu SPRAWDZENIA");
sw:    if((c=getch())==NULL)
        switch(getch())
        {
            case BACK:
                clrscr();
                execl("MENU-2","MENU-2",NULL);
                break;
            default:
                goto sw;
        }
    else
    {
        clrscr();
        strona();
        system("DRYF.BAT"); /*wykonanie HBM-S14.BAS i MENU-2*/
        clrscr();
    }

}

/***** koniec main *****/
/*----- naglowek strony -----*/
strona()
{
    float Temp;
    FILE *fd;
    struct date Dzień;
    struct time Czas;

    if((fd=fopen(4,"w"))==NULL)
    {
        printf("Błąd drukarki !");
        exit(0);
    }

    getdate(&Dzień);
    gettime(&Czas);
    Temp=carex(6);
    fprintf(fd,"\f* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *\n");
    fprintf(fd,"Data spr. %d.%02d.%02d          Temp. %4.1f          Godz. %d:%02d:%02d\n\n\n",\
        Dzień.da_year,Dzień.da_mon,Dzień.da_day,Temp,Czas.ti_hour,Czas.ti_min,Czas.ti_sec);
    return;
}

float carex(nrk)

char nrk;

{
    unsigned pa,pb,kanal,K;
    int port_ab;
    int i;
    long proba=0;
    float temp;
    float Sens=20.48;          /* czulosc div/stop.C */

    nrk=nrk-1;
    K=TEMP!nrk;                /*nr kanalu */

    outportb(PCW,T3);          /* ustaw tryb T3 , zeruj */
    outportb(PC,K);            /*wyslij do PC symbol prog. 0x4nrk t.j. uruchom.MSA*/

next:
    while((!inportb(PA))&&(!inportb(PB))) /*oczekiwanie na pomiar */
        ;
    temp=(float)inportb(PA);
    temp=(temp-0.5)/Sens;
}

```

```

pb=inportb(PB);
port_ab=((pb&M)<<8)!pa;          /* sklejenie bitow B+A */
kanal=pb>>4;                     /* wyodrebnienie nr kanalu */
if(kanal!=nrk)
{
    if(proba<100000)
    {
        proba++;
        goto next;
    }

    else
    {
        outportb(PC,0x0);          /* stop dla MSA */
        return NULL;
    }
}
outportb(PC,0x0);                /* stop dla MSA */
temp=(port_ab-ZERO)/Sens;         /* przelczenie na stopnie C */

for(i=1;i<15000;i++)             /* opoznienie */
;
return temp;
}

```

```

10 REM***** HBM-S14 (3 czujniki) *****
20 MASK=&H10
100 PCW=&H13E3:PA=&H13E0:PB=&H13E1:PC=&H13E2
101 OUT PCW,&H92
250 CLS
260 PRINT"*** POMIAR DRYFTU POLOZENIA ***":PRINT:PRINT
270 PRINT"U W A G A ! - STOP POMIARU = dowolny klawisz (max.150 pomiarow)"
310 LPRINT"*** POMIAR DRYFTU POLOZENIA ***"
340 LPRINT:LPRINT
350 REM *****
360 IC=3
370 L=0
380 IP=150
390 DIM WRD$(IP)
400 OPEN "COM1:9600,E,7,2" AS #1:ON ERROR GOTO 830
410 PRINT #1,CHR$(18);
420 PRINT #1,"CPB7B;TEX*"
430 PRINT #1,"IDY"
445 LINE INPUT#1,A$:PRINT A$
470 MSA=&HAS
480 OUT PC,MSA
490 ROB=INP(PA)
495 IF INKEY$<>"" THEN GOTO 570
500 IF ((ROB AND MASK)<>MASK) GOTO 490
505 OUT PC,&H0
510 PRINT #1,"MSV 12,14,0"
520 LINE INPUT#1,A$:PRINT "Pomiar nr : ",L+1:PRINT A$
525 FOR I=1 TO 78:PRINT"-";:NEXT I:PRINT"- "
530 WRD$(L+1)=A$
540 L=L+1
550 IF INKEY$="" THEN GOTO 560 ELSE 570
560 IF L<=IP THEN GOTO 480
570 IP=L
575 OUT PC,&H0
580 REM*****KONIEC POMIAROW*****
590 DIM WART(IC-1,IP1)
600 FOR Y=1 TO IP1
610 FOR X=0 TO IC-1
620 WART(X,Y)=VAL(MID$(WRD$(Y),(6+(X*15)),10))
630 NEXT X
640 NEXT Y
650 REM*****
660 GOSUB 850
670 GOSUB 920
680 GOSUB 1010
690 GOSUB 920
700 GOSUB 850
710 GOSUB 1060
720 REM***** MAX *****
730 LPRINT:LPRINT
740 CZ=0
750 LPRINT"Maksymalna odchyłka w osi X wynosi - MAX X = ";
760 GOSUB 1160
770 CZ=1
780 LPRINT"Maksymalna odchyłka w osi Y wynosi - MAX Y = ";
790 GOSUB 1160
800 CZ=2
810 LPRINT"Maksymalna odchyłka w osi Z wynosi - MAX Z = ";
820 GOSUB 1160
825 GOSUB 1300
830 CLS
840 SYSTEM

```



```

860 FOR I=1 TO ((IC*11)+6)
870 LPRINT "=";
880 NEXT I
890 LPRINT
900 RETURN
910 REM*****
920 REM***** I I I I *****
930 LOCATE,1
940 LPRINT"I";SPACE$(5);
950 FOR I=1 TO IC
960 LPRINT"I";SPACE$(10);
970 NEXT I
980 LPRINT
990 RETURN
1000 REM*****
1010 REM***** OPIS GLOWKI *****
1020 LPRINT" L P";SPACE$(7);"X";SPACE$(10);"Y";SPACE$(10);"Z"
1030 LPRINT SPACE$(8);"kanal 12   kanal 13   kanal 14"
1040 RETURN
1050 REM*****
1060 REM ***** WYDRUK WYNIKOW *****
1070 FOR Y=1 TO IP1
1080 LPRINT USING"   ###";Y;
1090 FOR X=0 TO IC-1
1100 LPRINT USING"  ##.###  ";WART(X,Y);
1110 NEXT X
1120 LPRINT
1130 NEXT Y
1140 RETURN
1150 REM*****
1160 REM***** SUB MAX *****
1170 MAX=ABS(WART(CZ,1))
1180 FOR I=1 TO IP1-1
1190 AKT=ABS(WART(CZ,I+1))
1200 IF AKT>MAX THEN MAX=AKT
1210 NEXT I
1220 LPRINT "WARTOSC ABSOLUTNA ";MAX
1230 RETURN
1240 REM*****
1300 REM***** OBLICZ R *****
1305 REM----- OBLICZ SREDNIA -----
1310 DIM WS(IC-1)
1320 FOR X=0 TO IC-1
1330 FOR Y=1 TO IP1
1350 WS(X)=WS(X)+WART(X,Y)
1370 NEXT Y
1375 WS(X)=WS(X)/IP1
1380 NEXT X
1400 REM----- OBLICZ Dj -----
1410 DIM DJ(IC-1)
1420 FOR X=0 TO IC-1
1440 FOR Y=1 TO IP1
1450 WART(X,Y)=ABS(WART(X,Y)-WS(X))
1455 DJ(X)=DJ(X)+WART(X,Y)
1460 NEXT Y
1470 DJ(X)=DJ(X)/IP1
1480 NEXT X
1500 REM----- OBLICZ Sd -----
1510 DIM SD(IC-1)
1520 FOR X=0 TO IC-1
1530 FOR Y=1 TO IP1
1540 SD(X)=SD(X)+((WART(X,Y)-DJ(X))^2)
1550 NEXT Y
1560 SD(X)=SD(X)/(IP-1)
1565 SD(X)=SQR(SD(X))
1570 NEXT X
1580 REM----- OBLICZ rx,ry,rz -----
1600 RX=DJ(0)+3*SD(0)
1605 IF RX<.0001 THEN RX=0
1610 RY=DJ(1)+3*SD(1)
1615 IF RY<.0001 THEN RY=0
1620 RZ=DJ(2)+3*SD(2)
1625 IF RZ<.0001 THEN RZ=0
1630 REM----- WYDRUK Rx,Ry,Rz -----
1640 LPRINT
1650 LPRINT "WARTOSC APx = ";RX
1660 LPRINT "WARTOSC APy = ";RY
1670 LPRINT "WARTOSC APz = ";RZ
1700 RETURN
1701 REM *****

```

* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *
 Data spr. 1991.12.30 Temp. 0.0 Godz. 14:15:25

*** POMIAR DRYFTU POLOZENIA ***

```

=====
I      I      I      I
L P      X      Y      Z
      kanal 12  kanal 13  kanal 14
I      I      I      I
=====
 1 -3.343 -1.176 -4.344
 2 -3.343 -1.176 -4.344
 3 -3.343 -1.176 -4.344
 4 -3.343 -1.176 -4.344
 5 -3.343 -1.176 -4.344
 6 -3.343 -1.176 -4.344
 7 -3.343 -1.176 -4.344
 8 -3.343 -1.176 -4.344
 9 -3.342 -1.176 -4.344
10 -3.342 -1.176 -4.344
11 -3.343 -1.176 -4.344
12 -3.343 -1.176 -4.344
13 -3.343 -1.176 -4.344
14 -3.342 -1.176 -4.344
15 -3.343 -1.176 -4.344
16 -3.342 -1.176 -4.344
17 -3.343 -1.176 -4.344
18 -3.342 -1.176 -4.344
19 -3.342 -1.176 -4.344
20 -3.343 -1.176 -4.344
21 -3.344 -1.176 -4.344
22 -3.343 -1.176 -4.344
23 -3.343 -1.176 -4.344
24 -3.343 -1.176 -4.344
25 -3.343 -1.176 -4.344
26 -3.342 -1.176 -4.344
27 -3.343 -1.176 -4.344
28 -3.343 -1.176 -4.344
29 -3.343 -1.176 -4.344
30 -3.343 -1.176 -4.344
  
```

Maksymalna odchyłka w osi X wynosi - MAX X = WARTOSC ABSOLUTNA 3.344
 Maksymalna odchyłka w osi Y wynosi - MAX Y = WARTOSC ABSOLUTNA 1.176
 Maksymalna odchyłka w osi Z wynosi - MAX Z = WARTOSC ABSOLUTNA 4.344

WARTOSC APx = 7.711874E-04
 WARTOSC APy = 0
 WARTOSC APz = 0

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include "carex.h"
#include <dos.h>
```

```
#define BACK 0x4b
```

```
main()
{
    char c;
    int i;
    char *tekst[] =
    {
        "15.SPRAWDZENIE DOKLADNOSCI ODTWARZANIA TOROW KOLOWYCH I PROSTOKATNYCH.",
        " ",
        " - przygotowac robota zgodnie z DTR,",
        " - zainstalowac na laczu mechanicznym robota obcaznik wg ark.3.1",
        " i glowice pomiarowa GP-2L lub GP-TCP (odp.do ksztaltu toru,",
        " - zainstalowac na stanowisku IIa linial pomiarowy lub glowice ",
        " GP-2L/k/1 (dla toru kolowego) wraz z plansza do synchronizacji, ",
        " - polaczyc ukklad pomiarowy wg ark.5.1.1, 5.6,",
        " - zsynchronizowac robota,",
        " - wczytac program uzytkowy z kasety dla ukkladu sterowania robota(USR)",
        " - uruchomic podprogram TOR 1.1(lub TOR 2.1 ) w trybie",
        " pracy KROKOWO symulujac z panelu proramowania odp.wejsca,",
        " - ustawic i wyzerowac wskazania czunikow w p-ktach pomiarowych,",
        " - wystartowa program USR w trybie AUTO,",
        " wcisnij klawisz",
    };

    printf("% S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *");
    gotoxy(1,5);
    for(i=0;i<15;i++)
        printf("%s\n",tekst[i]);
    gotoxy(40,25);
    printf("\21\55 - powrot do menu SPRAWDZENIA");
    sw: if((c=getch())==NULL)
        switch(getch())
        {
            case BACK:
                clrscr();
                execl("MENU-2","MENU-2",NULL);
                break;
            default:
                goto sw;
        }
    else
    {
        clrscr();
        strona();
        system("TOR_s15.BAT"); /*wykonanie HBM-S15.BAS i MENU-2*/
        clrscr();
    }
}
```

```

/***** koniec main *****/
/*----- naglowek strony -----*/
strona()
{
float Temp;
FILE *fd;
struct date Dzień;
struct time Czas;

if((fd=fopen(4,"w"))==NULL)
{
printf("Błąd drukarki !");
exit(0);
}

getdate(&Dzień);
gettime(&Czas);
Temp=carex(6);
fprintf(fd,"\\f* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *\\n");
fprintf(fd,"Data spr. %d.%02d.%02d          Temp. %4.1f          Godz. %d:%02d:%02d\\n\\n\\n",\\
        Dzień.da_year,Dzień.da_mon,Dzień.da_day,Temp,Czas.ti_hour,Czas.ti_min,Czas.ti_sec);
return;
}

float carex(nrk)

char nrk;

{
unsigned pa,pb,kanal,K;
int port_ab;
int i;
long proba=0;
float temp;
float Sens=20.48;          /* czułość div/stop.C */

nrk=nrk-1;
K=TEMP!nrk;                /*nr kanalu */

outportb(PCW,T3);          /* ustaw tryb T3 , zeruj */
outportb(PC,K);            /*wyslij do PC symbol prog. 0x4nrk  t.j. uruchom.MSA*/

next:
while((!inportb(PA))&&(!inportb(PB)))    /*oczekiwanie na pomiar */
;
pa=inportb(PA);            /* odczyt portow A i B */
pb=inportb(PB);
port_ab=((pb&M)<<8)!pa;      /* sklejenie bitow B+A */
kanal=pb>>4;               /* wyodrebnienie nr kanalu */
if(kanal!=nrk)
{
if(proba<100000)
{
proba++;
goto next;
}

else
{
outportb(PC,0x0);          /* stop dla  MSA */
return NULL;
}
}

outportb(PC,0x0);          /* stop dla  MSA */
temp=(port_ab-ZERO)/Sens;   /* przeliczenie na stopnie C */

for(i=1;i<15000;i++)        /* opoznienie */
;
return temp;
}

```

```

10 REM***** HBM-S15 (2 czujniki) *****
20 MASK=&H10
100 PCW=&H13E3:PA=&H13E0:PB=&H13E1:PC=&H13E2
101 OUT PCW,&H92
250 CLS
260 PRINT"*** POMIAR DOKLADNOSCI ODTWARZANIA TORU RUCHU ***":PRINT:PRINT
270 PRINT"U W A G A !      -      STOP POMIARU = dowolny klawisz (max.150 pomiarow)"
310 LPRINT"*** POMIAR DOKLADNOSCI ODTWARZANIA TORU RUCHU ***"
340 LPRINT:LPRINT
350 REM *****
360 IC=2
370 L=0
380 IP=150
390 DIM WRD$(IP)
400 OPEN "COM1:9600,E,7,2" AS #1:ON ERROR GOTO 830
410 PRINT #1,CHR$(18);
420 PRINT #1,"CPB78;TEX*"
430 PRINT #1,"IDY"
445 LINE INPUT#1,A$:PRINT A$
450 PRINT "WYBOR TORU : "
452 PRINT "          0 - tor prostokatny (P2-P5-P2)"
454 PRINT "          1 - tor prostokatny (P2-P3-P2)"
456 PRINT "          2 - tor kolowy      (r max)"
458 PRINT "          3 - tor kolowy      (r min)"
460 INPUT PP
470 MSA=&HAE-PP
480 OUT PC,MSA
490 ROB=INP(PA)
495 IF INKEY$<>" " THEN GOTO 570
500 IF ((ROB AND MASK)<>MASK) GOTO 490
505 OUT PC,&H0
510 PRINT #1,"MSV 12,13,0"
520 LINE INPUT#1,A$:PRINT "Pomiar nr : ",L+1:PRINT A$
525 FOR I=1 TO 78:PRINT"-";:NEXT I:PRINT"- "
530 WRD$(L+1)=A$
540 L=L+1
550 IF INKEY$="" THEN GOTO 560 ELSE 570
560 IF L<=IP THEN GOTO 480
570 IP1=L
575 OUT PC,&H0
580 REM*****KONIEC POMIAROW*****
590 DIM WART(IC-1,IP1)
600 FOR Y=1 TO IP1
610 FOR X=0 TO IC-1
620 WART(X,Y)=VAL(MID$(WRD$(Y),(6+(X*15)),10))
630 NEXT X
640 NEXT Y
650 REM*****
660 GOSUB 850
670 GOSUB 920

```

```

690 GOSUB 920
700 GOSUB 850
710 GOSUB 1060
720 REM***** MAX *****
825 GOSUB 1300
830 CLS
840 SYSTEM
850 REM***** ===== *****
860 FOR I=1 TO ((IC*11)+6)
870 LPRINT"";
880 NEXT I
890 LPRINT
900 RETURN
910 REM*****
920 REM***** I I I I *****
930 LOCATE,1
940 LPRINT"I";SPACE$(5);
950 FOR I=1 TO IC
960 LPRINT"I";SPACE$(10);
970 NEXT I
980 LPRINT
990 RETURN
1000 REM*****
1010 REM***** OPIS GLOWKI *****
1020 LPRINT" L P";SPACE$(7);"1";SPACE$(10);"2"
1030 LPRINT SPACE$(8);"kanal 12   kanal 13 "
1040 RETURN
1050 REM*****
1060 REM ***** WYDRUK WYNIKOW *****
1070 FOR Y=1 TO IP1
1080 LPRINT USING"   ###";Y;
1090 FOR X=0 TO IC-1
1100 LPRINT USING"   ##.###   ";WART(X,Y);
1110 NEXT X
1120 LPRINT
1130 NEXT Y
1140 RETURN
1150 REM*****
1240 REM*****
1300 REM***** OBLICZ R *****
1305 REM----- OBLICZ SREDNIA -----
1310 DIM WS(IC-1)
1320 FOR X=0 TO IC-1
1330 FOR Y=1 TO IP1
1350 WS(X)=WS(X)+WART(X,Y)
1370 NEXT Y
1375 WS(X)=WS(X)/IP1
1380 NEXT X
1400 REM----- OBLICZ Dj -----
1410 DIM DJ(IC-1)
1420 FOR X=0 TO IC-1
1440 FOR Y=1 TO IP1
1450 WART(X,Y)=ABS(WART(X,Y)-WS(X))
1455 DJ(X)=DJ(X)+WART(X,Y)
1460 NEXT Y
1470 DJ(X)=DJ(X)/IP1
1480 NEXT X
1500 REM----- OBLICZ Sd -----
1510 DIM SD(IC-1)
1520 FOR X=0 TO IC-1
1530 FOR Y=1 TO IP1
1540 SD(X)=SD(X)+((WART(X,Y)-DJ(X))^2)
1550 NEXT Y
1560 SD(X)=SD(X)/(IP-1)
1565 SD(X)=SQR(SD(X))
1570 NEXT X
1580 REM----- OBLICZ rx,ry,rz -----
1600 RX=DJ(0)+3*SD(0)
1605 IF RX<.0001 THEN RX=0
1610 RY=DJ(1)+3*SD(1)
1615 IF RY<.0001 THEN RY=0

```

```
1640 LPRINT
1650 AT=SQR(RX^2+RY^2)
1660 LPRINT "DOKLADNOSC ODTWARZANIA TORU WYNOSI : AT = ";AT
1700 RETURN
1701 REM *****
```

* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *
Data spr. 1991.12.30 Temp. 0.0 Godz. 16:01:12

*** POMIAR DOKLADNOSCI ODTWARZANIA TORU RUCHU ***

```
=====
I      I      I
L P      1      2
      kanal 12  kanal 13
I      I      I
=====
1 -3.335 -1.176
2 -3.335 -1.176
3 -3.335 -1.176
4 -3.335 -1.176
5 -3.335 -1.176
6 -3.335 -1.176
7 -3.335 -1.176
8 -3.335 -1.176
9 -3.335 -1.176
10 -3.336 -1.176
11 -3.336 -1.176
12 -3.336 -1.176
13 -3.335 -1.176
```

DOKLADNOSC ODTWARZANIA TORU WYNOSI : AT = 5.561657E-04

47


```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include "carex.h"
#include <dos.h>
```

```
#define BACK 0x4b
```

```
main()
{
char c;
int i;
char *tekst[]=
{
    "16.SPRAWDZENIE DŁUGOŚCI ODCINKA STABILIZACJI.",
    "    ",
    "    - przygotować robota zgodnie z DTR,",
    "    - zainstalować na łączu mechanicznym robota obciążnik wg ark.3.1",
    "    i głowice pomiarowa GP-TCP,",
    "    - zainstalować na stanowisku IIa głowice GP-2L/k/1",
    "    wraz z planszą do synchronizacji,",
    "    - połączyć układ pomiarowy wg ark.5.1.1, 5.6,",
    "    - zsynchronizować robota,",
    "    - wczytać program użytkowy z kasety dla układu sterowania robota(USR)",
    "    - uruchomić podprogram SPL w trybie pracy KROKOWO symulując",
    "    z panelu programowania odp.wejścia,",
    "    - ustawić i wyzerować wskazania czuników w p-ktach pomiarowych,",
    "    - wystartować program USR w trybie AUTO,",
    "    wcisnąć klawisz",
};
```

```
printf("S U M - R S U M - R S U M - R S U M - R S U M - R");
gotoxy(1,5);
for(i=0;i<15;i++)
    printf("%s\n",tekst[i]);
gotoxy(40,25);
printf("\21\55 - powrót do menu SPRAWDZENIA");
sw: if((c=getch())==NULL)
    switch(getch())
    {
        case BACK:
            clrscr();
            execl("MENU-2","MENU-2",NULL);
            break;
        default:
            goto sw;
    }
else
{
    clrscr();
    strona();
    system("SPL_s16.BAT"); /*wykonanie HBM-S16.BAS i MENU-2*/
}
```

```
}
```

```

/***** koniec main *****/
/*----- naglowek strony -----*/
strona()
{
float Temp;
FILE *fd;
struct date Dzień;
struct time Czas;

if((fd=fopen(4,"w"))==NULL)
{
printf("Bład drukarki !");
exit(0);
}
getdate(&Dzien);
gettime(&Czas);
Temp=carex(6);
fprintf(fd,"\\f$ S U M - R $ $ S U M - R $ $ S U M - R $ $ S U M - R $ $ S U M - R $\\n");
fprintf(fd,"Data spr. %d.%02d.%02d Temp. %4.1f Godz. %d:%02d:%02d\\n\\n",\
Dzien.da_year,Dzien.da_mon,Dzien.da_day,Temp,Czas.ti_hour,Czas.ti_min,Czas.ti_sec);
return;
}

```

```
float carex(nrk)
```

```
char nrk;
```

```

{
unsigned pa,pb,kanal,K;
int port_ab;
int i;
long proba=0;
float temp;
float Sens=20.48;          /* czulosc div/stop.C */

nrk=nrk-1;
K=TEMP!nrk;                /*nr kanalu */

outportb(PCW,T3);          /* ustaw tryb T3 , zeruj */
outportb(PC,K);            /*wyslij do PC symbol prog. 0x4nrk t.j. uruchom.MSA*/

```

```

next:
while((!inportb(PA))&&(!inportb(PB))) /*oczekiwanie na pomiar */

```

```

;
pa=inportb(PA);            /* odczyt portow A i B */
pb=inportb(PB);
port_ab=((pb&M)<<8)!pa;    /* sklejanie bitow B+A */
kanal=pb>>4;              /* wyodrebnienie nr kanalu */
if(kanal!=nrk)

```

```

{
if(proba<100000)
{
proba++;
goto next;
}

```

```

else
{
outportb(PC,0x0);          /* stop dla MSA */
return NULL;
}
}

```

```

outportb(PC,0x0);          /* stop dla MSA */
temp=(port_ab-ZERO)/Sens;   /* przelczenie na stopnie C */

```

```

for(i=1;i<15000;i++)       /* opoznienie */

```

```
return temp;  
}
```

```

10 REM***** HBM-S16 (2 czujniki) *****
20 MASK=&H10
100 PCW=&H13E3:PA=&H13E0:PB=&H13E1:PC=&H13E2
101 OUT PCW,&H92
250 CLS
260 PRINT"*** POMIAR ODCINKA STABILIZACJI ***":PRINT:PRINT
270 PRINT"U W A G A ! - STOP POMIARU = dowolny klawisz (max.150 pomiarow)"
310 LPRINT"*** POMIAR DLUGOSCI ODCINKA STABILIZACJI DLA TORU LAMANEGO ***"
340 LPRINT:LPRINT
350 REM *****
360 IC=2
370 L=0
380 IP=150
390 DIM WRD$(IP)
400 OPEN "COM1:9600,E,7,2" AS #1:ON ERROR GOTO 830
410 PRINT #1,CHR$(18);
420 PRINT #1,"CPB78;TEX*"
430 PRINT #1,"IDY"
445 LINE INPUT#1,A$:PRINT A$
470 MSA=&HAA
480 OUT PC,MSA
490 ROB=INP(PA)
495 IF INKEY$<>"" THEN GOTO 570
500 IF ((ROB AND MASK)<>MASK) GOTO 490
505 OUT PC,&H0
510 PRINT #1,"MSV 12,13,0"
520 LINE INPUT#1,A$:PRINT "Pomiar nr : ",L+1:PRINT A$
525 FOR I=1 TO 78:PRINT"-";:NEXT I:PRINT"- "
530 WRD$(L+1)=A$
540 L=L+1
550 IF INKEY$="" THEN GOTO 560 ELSE 570
560 IF L<=IP THEN GOTO 480
570 IP=L
575 OUT PC,&H0
580 REM*****KONIEC POMIAROW*****
590 DIM WART(IC-1,IP1)
600 FOR Y=1 TO IP1
610 FOR X=0 TO IC-1
620 WART(X,Y)=VAL(MID$(WRD$(Y),(6+(X*15)),10))
630 NEXT X
640 NEXT Y
650 REM*****
660 GOSUB 850
670 GOSUB 920
680 GOSUB 1010
690 GOSUB 920
700 GOSUB 850
710 GOSUB 1060
720 REM***** MAX *****
825 GOSUB 1300
830 CLS
840 SYSTEM
850 REM***** ===== *****
860 FOR I=1 TO ((IC*11)+6)
870 LPRINT"=";
880 NEXT I
890 LPRINT
900 RETURN
910 REM*****
920 REM***** I I I I *****
930 LOCATE,1
940 LPRINT"I";SPACE$(5);
950 FOR I=1 TO IC
960 LPRINT"I";SPACE$(10);

```

```

770  RETURN
980  LPRINT
990  RETURN
1000 REM*****
1010 REM***** OPIS GLOWKI *****
1020 LPRINT" L P";SPACE$(7);"1";SPACE$(10);"2"
1030 LPRINT SPACE$(8);"kanal 12   kanal 13 "
1040 RETURN
1050 REM*****
1060 REM ***** WYDRUK WYNIKOW *****
1070 FOR Y=1 TO IP1
1080 LPRINT USING"   ###";Y;
1090 FOR X=0 TO IC-1
1100 LPRINT USING"  ##.###  ";WART(X,Y);
1110 NEXT X
1120 LPRINT
1130 NEXT Y
1140 RETURN
1150 REM*****
1240 REM*****
1300 REM***** OBLICZ R *****
1305 REM----- OBLICZ SREDNIA -----
1310 DIM WS(IC-1)
1320 FOR X=0 TO IC-1
1330 FOR Y=1 TO IP1
1350 WS(X)=WS(X)+WART(X,Y)
1370 NEXT Y
1375 WS(X)=WS(X)/IP1
1380 NEXT X
1400 REM----- OBLICZ Dj -----
1410 DIM DJ(IC-1)
1420 FOR X=0 TO IC-1
1440 FOR Y=1 TO IP1
1450 WART(X,Y)=ABS(WART(X,Y)-WS(X))
1455 DJ(X)=DJ(X)+WART(X,Y)
1460 NEXT Y
1470 DJ(X)=DJ(X)/IP1
1480 NEXT X
1500 REM----- OBLICZ Sd -----
1510 DIM SD(IC-1)
1520 FOR X=0 TO IC-1
1530 FOR Y=1 TO IP1
1540 SD(X)=SD(X)+((WART(X,Y)-DJ(X))^2)
1550 NEXT Y
1560 SD(X)=SD(X)/(IP-1)
1565 SD(X)=SQR(SD(X))
1570 NEXT X
1580 REM----- OBLICZ rx,ry,rz -----
1600 RX=DJ(0)+3*SD(0)
1605 IF RX<.0001 THEN RX=0
1610 RY=DJ(1)+3*SD(1)
1615 IF RY<.0001 THEN RY=0
1630 REM----- WYDRUK Rx,Ry,Rz -----
1640 LPRINT
1650 AT=SQR(RX^2+RY^2)
1660 LPRINT "DLUGOSC ODCINKA STABILIZACJI WYNOSI : SFL = ";AT
1700 RETURN
1701 REM *****

```

* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *
 Data spr. 1991.12.30 Temp. 18.6 Godz. 16:47:25

*** POMIAR DLUGOSCI ODCINKA STABILIZACJI DLA TORU LAMANEGO ***

```
=====
I      I      I
L P      1      2
      kanal 12  kanal 13
I      I      I
=====
1 -3.331 -1.176
2 -3.332 -1.176
3 -3.332 -1.176
4 -3.331 -1.176
5 -3.331 -1.176
6 -3.332 -1.176
7 -3.332 -1.176
8 -3.332 -1.176
9 -3.332 -1.176
10 -3.331 -1.176
11 -3.332 -1.176
12 -3.332 -1.176
13 -3.332 -1.176
14 -3.332 -1.176
15 -3.332 -1.176
16 -3.332 -1.176
17 -3.331 -1.176
```

DLUGOSC ODCINKA STABILIZACJI WYNOSI : SPL = 6.054879E-04

```

#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <dos.h>
#include "carex.h"
#include <math.h>

#define BACK 0x4b

int v_p[7];          /* tabela wyników pomiarów predkosci [impulsy] */
double v_m[7],v_k[7];
double v_ms,v_ks;    /* srednie */
double pasek,wspol;  /*dl.paska pom.,wspolczynnik do wzoru przelicz.*/
struct time Czas;
struct zeit          /*czas docelowy*/
{
    unsigned m;      /*min*/
    unsigned s;      /*sek*/
    unsigned ss;     /*sek/100*/
}d_Czas;
double przysp[300][4];
int LPP;              /*liczba pomiarów przyspieszenia*/
FILE *fd;
int Fc=5000;

main()                /* ** S17 ** */
{
    predkosc();
    clrscr();
    execl("MENU-2","MENU-2",NULL);

}                    /*koniec main*/

/*===== pomiar predkosci =====*/
predkosc()
{
    int i;
    char c,n_os;
    unsigned os;

    char *tekst1[]=
    {
        "17.SPRAWDZENIE DOKŁADNOŚCI I POWTARZALNOŚCI PRĘDKOŚCI.",
        "    - przygotować robota zgodnie z DTR,",
        "    - zainstalować na łączu mechanicznym robota obciążnik wg ark.3.1 ,",
        "    i czujnik Ft-5,",
        "    - zainstalować na stanowisku pomiarowym IIa /patrz DTR nr rej.6463/",
        "    liniał pomiarowy i plansze pomiarowa dla toru o kształcie kołowym,",
        "    - połączyć układ pomiarowy w/g ark.5.1.1,",
        "    - wczytać program użytkowy z kasety dla układu sterowania robota/USR/",
        "    - uruchomić program VAR w trybie pracy KROKOWO symulując z panelu",
        "    programowania odpowiednie wejścia,",
        "    - ustawić czujnik Ft-5 oraz liniał lub plansze pomiarowa,",
        "    - uruchomić program pomiaru prędkości dla określonych stopni swobody,",
        "    naciśnij klawisz",
    };
}

```

```

clrscr();
printf("* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *");
gotoxy(1,5);
for(i=0; i<13; i++)
    printf("%s\n", tekst1[i]);
gotoxy(40,25);
printf("\21\55 - powrot do menu SPRAWDZENIA");
sw:    if((c=getch())==NULL)
        switch(getch())
        {
            case BACK:
                clrscr();
                execl("MENU-2", "MENU-2", NULL);
                break;
            default:
                goto sw;
        }
    else
    {
        clrscr();
        printf("Podaj numer osi:\n");
        printf("      1 - fi\n      2 - teta\n      3 - alfa\n      4 - t\n      5 - v\n");
        printf("os nr - ");
        printf("%c\n", n_os=getch());
        switch(n_os)
        {
            case '1':
                os=0xae;
                pasek=200;
                wspol=7;
                break;
            case '2':
                os=0xad;
                pasek=24.6;
                wspol=3.54;
                break;
            case '3':
                os=0xac;
                pasek=56.5;
                wspol=3.24;
                break;
            case '4':
                os=0xab;
                pasek=400;
                wspol=79;
                break;
            case '5':
                os=0xaa;
                pasek=200;
                wspol=183.4;
                break;
            default:
                printf("\nBledny nr osi !\n");
                goto sw;
        }
        /*koniec switch*/
    }
    /*koniec else*/

printf("\nOczekiwanie na sygnal od robota...\n");
do_robot(os);
ft(v_p);
for(i=0; i<7; i++)
{
    printf("pomiar nr %d      * %d * impulsow\n", i+1, v_p[i]);
}
for(i=1; i<7; i++)
{
    v_m[i]=pasek/(v_p[i]*(0.0002)); /*przeliczenie na mm/s*/
    v_k[i]=wspol*Fc/v_p[i];        /*przeliczenie na o/s*/
    v_ms+=v_m[i];
    v_ks+=v_k[i];
}

```



```

return;
}

/*----- pomiar predkosci ----- ft -----*/

int ft(port_ab)          /* port_ab -tabela wyników 7 el-tow */

int port_ab[7];
{

int pa,pb;
int i;
char nrk=0x80;

outportb(PCW,T3);          /* ustaw tryb T3 , zeruj */
outportb(PC,nrk);          /*wyslij do PC symbol prog. 0xXnrk  t.j. uruchom.MSA*/
for(i=0;i<7;i++)          /* 7 pomiarow - impulsy * [0.2 ms]*/
{
while(inportb(PA))          /*oczekiwanie na zero */
;
while(!(inportb(PA)&&inportb(PB)))  /*oczekiwanie na pomiar */
;
pa=inportb(PA);          /* wartosc licznika L3 */
pb=inportb(PB);          /* wartosc licznika H3 */
port_ab[i]=((255-pb)*256)+(255-pa);
/*printf("%d  L3= %d    H3= %d\n",port_ab[i],pa,pb);*/
}
outportb(PC,0x0);          /* stop dla  MSA */
return;
}

/*----- do robota i od robota -----*/

do_robot(nrk)

char nrk;
{
unsigned pa;
int i;

outportb(PCW,T3);          /* ustaw tryb T3 , zeruj */
outportb(PC,nrk);          /*wyslij do PC symbol prog. 0xXnrk  t.j. uruchom.MSA*/

while(!(pa=inportb(PA)))    /*oczekiwanie na sygnal od robota */
;
outportb(PC,0x0);          /* stop dla  MSA */
for(i=1;i<32000;i++)
;
return pa;
}

/*----- naglowek strony -----*/
strona()
{
float Temp;
struct date Dzień;
struct time Czas;

if((fd=fopen(4,"w"))==NULL)
{
printf("Bład drukarki !");
exit(0);
}

getdate(&Dzień);
gettime(&Czas);
Temp=carex(6);

fprintf(fd,"\f* S U M - R * * * S U M - R * * * S U M - R * * * S U M - R * * * S U M - R *\n");
fprintf(fd,"Data spr.  %d.%02d.%02d          Temp.  %4.1f          Godz  %c.%02d.%02d\n\n" \

```

```

    Dzień.da_year,Dzień.da_mon,Dzień.da_day,Temp,Czas.ti_hour,Czas.ti_min,Czas.ti_sec);
return;
}

/*----- pomiar z CAREXu -----*/

float carex(nrk)

char nrk;

{
    unsigned pa,pb,kanal,K;
    int port_ab;
    int i;
    long proba=0;
    float temp;
    float Sens=20.48;          /* czulosc div/stop.C */

    nrk=nrk-1;
    K=TEMP!nrk;                /*nr kanalu */

    outportb(PCW,T3);          /* ustaw tryb T3 , zeruj */
    outportb(PC,K);            /*wyslij do PC symbol prog. 0x4nrk  t.j. uruchom.MSA*/

    next:
    while((!inportb(PA))&&(!inportb(PB))) /*oczekiwanie na pomiar */
        ;
    pa=inportb(PA);            /* odczyt portow A i B */
    pb=inportb(PB);
    port_ab=((pb&M)<<8)!pa;      /* sklejanie bitow B+A */
    kanal=pb>>4;               /* wyodrebnienie nr kanalu */
    if(kanal!=nrk)
    {
        if(proba<100000)
        {
            proba++;
            goto next;
        }

        else
        {
            outportb(PC,0x0);    /* stop dla MSA */
            return NULL;
        }
    }

    outportb(PC,0x0);          /* stop dla MSA */
    temp=(port_ab-ZERO)/Sens;   /* przelczenie na stopnie C */

    for(i=1;i<15000;i++)        /* opoznienie */
        ;
    return temp;
}

/*----- wydruk predkosci -----*/

wydruk(axi)
char axi;
{
    int i;

    strona();
    fprintf(fd,"                SPRAWDZENIE PREDKOSCI osi nr %c\n\n",axi);
    fprintf(fd,"Nr pomiaru          [ mm/s ]          [ stopni/s ]\n\n");
    for(i=1;i<7;i++)
    {
        fprintf(fd,"          %d                %6.1f                %5.1f\n",i,v_m[i],v_k[i]);
    }
    fprintf(fd,"\n  srednia                %6.1f                %5.1f\n",v_ms/6,v_ks/6);
    return;
}

```

```

/*----- od robota -----*/

od_robot()

{

unsigned pa;
int i;

outportb(PCW,T3);          /* ustaw tryb T3 ; zeruj */
outportb(PC,0xb0);         /*wyslij do PC symbol prog. 0xXnrk  t.j. uruchom.MSA*/

while(!(pa=inportb(PA)))    /*oczekiwanie na sygnal od robota */
;

outportb(PC,0x0);          /* stop dla  MSA */
for(i=1;i<32000;i++)
;
return pa;
}

/*----- wydruk przyspieszenia -----*/
wydruk1(axi)
char axi;
{
int i;
double max=0;
for(i=0;i<LPP;i++)
    max=(fabs(przysp[i][3])>fabs(max)?przysp[i][3]:max;    /*oblicz maximum*/

strona();
fprintf(fd,"          SPRAWDZENIE PRZYSPIESZENIA osi nr %c  [m/s2]  \n\n",axi);
fprintf(fd,"Nr pomiaru      kanal 3          kanal 4          kanal 5          srednia\n\n");
for(i=0;i<LPP;i++)
{
fprintf(fd,"      %3d          %6.2f          %6.2f          %6.2f          %6.2f\n",i+1,przysp[i][0],przysp[i][1],przysp[i][2],
przysp[i][3]);
}

fprintf(fd,"\nwart. maksymalna      -          -          -          %6.2f\n",max);
return;
}

```