

440

ZESPÓŁ AUTOMATYKI ELEKTRONICZNEJ

BE40

Nazwa ONBIZNB

Główny wykonawca

mgr inż. Elżbieta Jachczyk

Wykonawcy:

mgr inż. Tadeusz Goszczyński

tech. Andrzej Kulik

Wykonanie modelu komputerowego stanowiska KAL 401  
do sprawdzania charakterystyk par czujników temperatury.

Etap 3. Skorygowanie oprogramowania stanowiska KAL 401  
zgodnie z wytycznymi wyznaczania błędów pary  
czujników temperatury sformułowanymi przez  
Kierownika Laboratorium Pomiarów Ciepła GUM.

(Tytuł pracy, numer i tytuł etapu)

Zleceniodawca

praca statutowa

Kierownik Pracowni

mgr inż. Tadeusz Goszczyński

Z-ca Dyrektora  
d/s Bad.-Rozwojowych

dr inż. Jan Jąbłkowski

Kierownik Zespołu

doc.dr inż. J. Korytkowski

30.12.1996r.

Pracę zakończono dnia .....

Nr arch. 7377

Nr zlecenia S1677

## Analiza deskrytorowa

STANOWISKO BADAWCZE : POMIAR + KOMPUTERY + PRZETWORNIKI  
POMIAROWE

## Abstrakt

Przedstawiono skorygowany program komputerowy w języku „C” dokonujący analizy 3 punktów pomiarowych (temperatura , rezystancja) zgodnie z wytycznymi GUM dotyczącymi wyznaczania błędów pary czujników temperatury.

## Tytuły poprzednich sprawozdań

1. Wykonanie modelu komputerowego stanowiska KAL401 dla 15 czujników jako stanowiska na Targi Poznańskie wraz z demonstracyjnym programem pracy stanowiska na komputer PC.
2. Opracowanie instrukcji obsługi stanowiska KAL401 dla użytkownika.
3. Uruchomienie stanowiska KAL401, wykonanie badań funkcjonalnych i metrologicznych.

## Rozdzielnik

Egz. 1. .... OIN

Egz. 2. .... ZAE - 3

Egz. 3. .... ZAE - 1

# **Skorygowanie oprogramowania stanowiska KAL401 zgodnie z wytycznymi wyznaczania błędów pary czujników temperatury sformułowanymi przez kierownika Laboratorium Pomiarów Ciepła GUM**

## **1. Wstęp**

W GUM trwają dopiero prace nad polską wersją normy dotyczącej ciepłomierzy i uzupełniającymi tę normę instrukcjami. Przewiduje się, że norma wejdzie w życie od 1.01.1998, a jej redakcja jest spodziewana w połowie 1997 r. W tej sytuacji nie ma możliwości formalnego uzgodnienia metodyki badań par czujników temperatury na stanowisku KAL401. W związku z tym wyznaczanie błędów pary czujników na podstawie trzech pomiarów temperatury i rezystancji, realizowane przez oprogramowanie stanowiska KAL401, opiera się na wytycznych uzyskanych od mgr inż. Małgorzaty Danikiewicz, Kierownika Laboratorium Pomiarów Ciepła GUM. Wytyczne te są fragmentem projektu instrukcji opracowywanej przez GUM.

Wytyczne wyznaczania błędów pary czujników temperatury znajdują się w **ZAŁĄCZNIKU 1**, a skorygowane zgodnie z nimi oprogramowanie sterownika KAL401 (część oprogramowania o nazwie **pomiar.cpp** w języku C++) w **ZAŁĄCZNIKU 2**.

## **2. Wynik korekcji oprogramowania**

Wynik korekcji oprogramowania uwidocznił się w raporcie z badania, tworzonym przez stanowisko KAL401. Na podstawie wyznaczonych przez program stanowiska charakterystyk par czujników wyznaczane są błędy bezwzględne pomiaru temperatury w całym zakresie temperatur w °C i porównywane z błędem dopuszczalnym  $\pm 2^{\circ}\text{C}$ . Dla pary czujników wyliczany jest błąd maksymalny pomiaru różnicy temperatur dla każdego z przedziałów  $\Delta t$  i porównywany z granicznym błędem dopuszczalnym pary czujników temperatury. Ilustruje to podany na str. 2 wydruk raportu z badania zrealizowanego przez KAL401.

TABELA WYNIKOW CZUJNIKA WODY GORACEJ

| Lp. | t   | Robl    | Rnorm   | dR     | Et     | Edop | ocena |
|-----|-----|---------|---------|--------|--------|------|-------|
| -   | °C  | om      | om      | om     | °C     | °C   |       |
| 1   | 20  | 107.689 | 107.793 | -0.104 | -0.267 | 2.00 | DOBRY |
| 2   | 56  | 121.703 | 121.703 | -0.000 | -0.001 | 2.00 | DOBRY |
| 3   | 92  | 135.520 | 135.463 | 0.058  | 0.152  | 2.00 | DOBRY |
| 4   | 128 | 149.142 | 149.072 | 0.070  | 0.187  | 2.00 | DOBRY |
| 5   | 164 | 162.568 | 162.531 | 0.037  | 0.101  | 2.00 | DOBRY |
| 6   | 200 | 175.799 | 175.840 | -0.041 | -0.111 | 2.00 | DOBRY |

TABELA WYNIKOW CZUJNIKA WODY ZIMNEJ

| Lp. | t   | Robl    | Rnorm   | dR     | Et     | Edop | ocena |
|-----|-----|---------|---------|--------|--------|------|-------|
| -   | °C  | om      | om      | om     | °C     | °C   |       |
| 1   | 20  | 107.710 | 107.793 | -0.083 | -0.214 | 2.00 | DOBRY |
| 2   | 56  | 121.739 | 121.703 | 0.036  | 0.094  | 2.00 | DOBRY |
| 3   | 92  | 135.565 | 135.463 | 0.102  | 0.269  | 2.00 | DOBRY |
| 4   | 128 | 149.188 | 149.072 | 0.116  | 0.308  | 2.00 | DOBRY |
| 5   | 164 | 162.607 | 162.531 | 0.076  | 0.205  | 2.00 | DOBRY |
| 6   | 200 | 175.823 | 175.840 | -0.016 | -0.044 | 2.00 | DOBRY |

TABELA WYNIKOW PARY CZUJNIKOW

| Lp. | t1  | t2  | t1-t2 | Elt    | Edop  | ocena |
|-----|-----|-----|-------|--------|-------|-------|
|     | °C  | °C  | °C    | %      | %     |       |
| 1   | 23  | 20  | 3     | 0.877  | 3.500 | DOBRY |
| 2   | 34  | 31  | 3     | 1.494  | 3.500 | DOBRY |
| 3   | 46  | 43  | 3     | 2.057  | 3.500 | DOBRY |
| 4   | 57  | 54  | 3     | 2.562  | 3.500 | DOBRY |
| 5   | 69  | 66  | 3     | 3.007  | 3.500 | DOBRY |
| 6   | 80  | 77  | 3     | 3.389  | 3.500 | DOBRY |
| 7   | 30  | 20  | 10    | -0.325 | 2.500 | DOBRY |
| 8   | 64  | 54  | 10    | 0.368  | 2.500 | DOBRY |
| 9   | 98  | 88  | 10    | 0.910  | 2.500 | DOBRY |
| 10  | 132 | 122 | 10    | 1.292  | 2.500 | DOBRY |
| 11  | 166 | 156 | 10    | 1.504  | 2.500 | DOBRY |
| 12  | 200 | 190 | 10    | 1.539  | 2.500 | DOBRY |
| 13  | 40  | 20  | 20    | -0.543 | 1.250 | DOBRY |
| 14  | 72  | 52  | 20    | -0.078 | 1.250 | DOBRY |
| 15  | 104 | 84  | 20    | 0.325  | 1.250 | DOBRY |
| 16  | 136 | 116 | 20    | 0.664  | 1.250 | DOBRY |
| 17  | 168 | 148 | 20    | 0.934  | 1.250 | DOBRY |
| 18  | 200 | 180 | 20    | 1.133  | 1.250 | DOBRY |
| 19  | 180 | 20  | 160   | -0.147 | 1.250 | DOBRY |
| 20  | 200 | 40  | 160   | 0.053  | 1.250 | DOBRY |

WYNIK BADANIA JEST POZYTYWNY

## ZAŁĄCZNIK 1

## WIADOMOŚĆ

od: Małgorzaty DANKIEWICZ, GUM, M2  
dla: p. Tadeusza GOSZCZYŃSKIEGO, PIAP, Zespół Automatyki Elektronicznej,  
Warszawa, fax 863 88 64

2 str.

### Wyznaczanie błędów pary czujników temperatury

§ 1.1. Błąd pary czujników temperatury  $E_T$ , wyrażony w procentach, należy wyznaczyć w następujący sposób:

1) zmierzyć opór (w  $\Omega$ ) obu czujników, wchodzących w skład pary czujników temperatury, w trzech różnych temperaturach (w  $^{\circ}\text{C}$ ):

- a) w temperaturze  $t_1$ :  $R_{1t1}$  - opór czujnika 1,  $R_{2t1}$  - opór czujnika 2,
- b) w temperaturze  $t_2$  -  $R_{1t2}$  - opór czujnika 1,  $R_{2t2}$  - opór czujnika 2,
- c) w temperaturze  $t_3$  -  $R_{1t3}$  - opór czujnika 1,  $R_{2t3}$  - opór czujnika 2.

2) dla każdego z czujników ułożyć trzy równania z trzema niewiadomymi wg PN-83/M-53852 Termometry elektryczne. Charakterystyki termometryczne oporników (rezystorów) termometrycznych, równanie (Z-1):

a) dla czujnika 1 (niewiadome:  $R_{01}, A_1, B_1$ )

$$R_{1t1} = R_{01} (1 + A_1 \cdot t_1 + B_1 \cdot t_1^2)$$

$$R_{1t2} = R_{01} (1 + A_1 \cdot t_2 + B_1 \cdot t_2^2)$$

$$R_{1t3} = R_{01} (1 + A_1 \cdot t_3 + B_1 \cdot t_3^2)$$

b) dla czujnika 2 (niewiadome:  $R_{02}, A_2, B_2$ )

$$R_{2t1} = R_{02} (1 + A_2 \cdot t_1 + B_2 \cdot t_1^2)$$

$$R_{2t2} = R_{02} (1 + A_2 \cdot t_2 + B_2 \cdot t_2^2)$$

$$R_{2t3} = R_{02} (1 + A_2 \cdot t_3 + B_2 \cdot t_3^2)$$

3) rozwiązując oba układy równań wyznaczyć wartości:  $R_{01}, A_1$  i  $B_1$  oraz  $R_{02}, A_2$  i  $B_2$ .

4) ułożyć równania obu czujników z wartościami stałych obliczonych w pkt. 3:

$$R_{1t} = R_{01} (1 + A_1 \cdot t + B_1 \cdot t^2)$$

$$R_{2t} = R_{02} (1 + A_2 \cdot t + B_2 \cdot t^2)$$

gdzie:  $R_{1t}$  - opór czujnika 1 w temperaturze  $t$ ,  $R_{2t}$  - opór czujnika 2 w temperaturze  $t$ ,

5) ułożyć równanie czujnika "idealnego" wg PN-83/M-53852:

$$R_t = R_0 (1 + A \cdot t + B \cdot t^2)$$

gdzie:

$R_t$  - opór czujnika "idealnego" w temperaturze  $t$ ,

$$R_0 = 100 \Omega \text{ dla czujników Pt100} \quad A = 3,90802 \cdot 10^{-3} \text{ } ^{\circ}\text{C}^{-1}$$

$$R_0 = 500 \Omega \text{ dla czujników Pt500} \quad B = -5,802 \cdot 10^{-7} \text{ } ^{\circ}\text{C}^{-2}$$

$$R_0 = 1000 \Omega \text{ dla czujników Pt1000}$$

6) obliczyć błędy bezwzględne pomiaru temperatury dla obu czujników ( $e_{t1}$  i  $e_{t2}$ , w  $\Omega$ ) w całym zakresie temperatur; przeliczyć je na  $^{\circ}\text{C}$  i sprawdzić, czy nie przekraczają wartości  $\pm 2 \text{ } ^{\circ}\text{C}$ ,

7) obliczyć błąd pary czujników temperatury  $E_T$  wg wzoru:

$$E_T = e_{T\max} / (R_{t1} - R_{t2}) \cdot 100 \%$$

gdzie:

$e_{Tmax}$  - wartość maksymalna błędu bezwzględnego pomiaru różnicy temperatur (w  $\Omega$ ) dla każdego z przedziałów  $\Delta t$ , określonych w ust. 3,

$R_{t1}$ ,  $R_{t2}$  - opór czujnika "idealnego" w temperaturze odpowiednio  $t_1$  i  $t_2$ , dla których wystąpił błąd  $e_{Tmax}$  w każdym z przedziałów  $\Delta t$ .

g) stwierdzić, czy  $|E_T| \leq |E_{Td}|$  gdzie  $E_{Td}$  - graniczny błąd dopuszczalny pary czujników temp.

2. Para czujników temperatury powinna być sprawdzona w każdym z trzech poniższych zakresów temperatury  $t$ :

1)  $35\text{ }^\circ\text{C} \leq t \leq 45\text{ }^\circ\text{C}$  lub

$t_{min} \leq t \leq t_{min} + 10\text{ }^\circ\text{C}$  (jeżeli określona w decyzji o zatwierdzeniu typu wartość  $t_{min} < 20\text{ }^\circ\text{C}$ )

2)  $75\text{ }^\circ\text{C} \leq t \leq 85\text{ }^\circ\text{C}$

3)  $t_{max} - 30\text{ }^\circ\text{C} \leq t \leq t_{max}$

gdzie  $t_{min}$ ,  $t_{max}$  - dolna i górna granica zakresu temperatury, określone w decyzji o zatwierdzeniu typu ciepłomierza lub pary czujników temperatury.

3. Błąd graniczny dopuszczalny pary czujników temperatury  $E_{\Delta t d}$ , obliczony względem wartości poprawnej różnicy temperatur  $\Delta t$ , wynosi:

a)  $E_{Td} = \pm 3,5\%$  dla  $\Delta t < 10\text{ }^\circ\text{C}$

b)  $E_{Td} = \pm 2,5\%$  dla  $10\text{ }^\circ\text{C} \leq \Delta t < 20\text{ }^\circ\text{C}$

c)  $E_{Td} = \pm 1,25\%$  dla  $\Delta t \geq 20\text{ }^\circ\text{C}$

7

**ZAŁĄCZNIK 2**



```
tmax=tmaxi;
dt_min=tdmini;
dt_max=tdmaxi;
r_n=100.0;
if(typ_czuj==1)
r_n=500.0;
if(typ_czuj==2)
r_n=1000.0;
for( int i=0; i<3; i++)
{
m_d2g[i] = sG[i];
m_d2z[i] = sZ[i];
}
SprawdzCzuj1();
SprawdzCzuj2();
ObliczPary();
return wyniki;
```

```
}
/////////////////////////////////////////////////////////////////
/*****/
```

```

//printf("\n %8f Ohms", atof(instr));

    *wynik = atof(instr);

#endif

#ifdef DEMO
double czujnika = 1;
if(typ_cz==1)
    czujnika=5;
if(typ_cz==2)
    czujnika=10;

*wynik = (100 + 19.4*m_nNrZlacza)* czujnika;
#endif
}

////////////////////////////////////
void CPomiar::PrzelaczNaCzujnik(WORD nNrZlacza, WORD nNrCzujnika)
{
    int adres = 0x300;
#ifdef DEMO
    WORD multiplex = (nNrCzujnika) + 16 * (nNrZlacza-1);
    multiplex = ~multiplex;
    unsigned char a=(unsigned char)multiplex;

    _outp(adres,a);
    czas_start = CTime::GetCurrentTime();
//TGDEMO
    Delay(1);
    WORD stan=_inp(adres);
    WORD maska = 1<<(nNrZlacza-1);
    stan=stan&maska;
    if(stan==0){
        sprintf(ok,"ZLY NR ZLACZA ");
// CParyDoc::printf(" BLAD MULTIPLEXERA zlacze nr %d", nNrZlacza );
    }
#endif
}

////////////////////////////////////
void CPomiar::PobierzWartosci(WORD NrZlacz,WORD NrCzujnika, double* dTemperatura, double* dRezystancja)
{
    int wzorcowy = 15;
    double dWynik;
    double dTemperatur;
    m_nNrZlacza=NrZlacz;
    m_nNrCzujnika= NrCzujnika;
// PrzelaczNaCzujnik(m_nNrZlacza, wzorcowy);
#ifdef DEMO
    czas_start = CTime::GetCurrentTime();
    Delay(15);
#endif
    PomiarIEC( & dWynik);
    TempWzorc(m_nNrZlacza, dWynik, &dTemperatur);
    *dTemperatura = dTemperatur;
// PrzelaczNaCzujnik(m_nNrZlacza, m_nNrCzujnika);
    czas_start = CTime::GetCurrentTime();
    Delay(1);
    PomiarIEC(& dWynik);
    *dRezystancja = dWynik;
}

////////////////////////////////////

////////////////////////////////////

const char* CPomiar::Oblicz(struct D2 *sG, struct D2 *sZ, double tmini, double tmaxi, double tmini, double tmaxi,WORD typ_czuj)
{
    typ_cz = typ_czuj;
    tmin=tmini;

```

```

    r001=25.51304;
#ifdef DEMO
*temp = 50.0;
#endif
}
if( nNrZlacza==2) {
    awz=-0.00026383;
    bwz=-0.00004098;
    //r001=100.0;
    r001=25.51304;
#ifdef DEMO
*temp = 100.0;
#endif
}
if( nNrZlacza==3) {
    awz=-0.00026383;
    bwz=-0.00004098;
    //r001=100.0;
    r001=25.51304;
#ifdef DEMO
*temp = 150.0;
#endif
}

double Wt = rezyst/r001;
double Wrt = Wt- awz*(Wt-1) - bwz*(Wt-1)*(Wt-1);
double ul = (Wrt-2.64)/1.64;
#ifdef DEMO
*temp = d0+d1*ul+d2*ul*ul+d3*pow(ul,3)+d4*pow(ul,4)+d5*pow(ul,5)+d6*pow(ul,6)+d7*pow(ul,7)+d8*po ,8
)+d9*pow(ul,9);
#endif
}
//////////////////////////////////////////////////////////////////
void CPomiar::PomiarIEC(double * wynik)
{
//////////////////////////////////////////////////////////////////

    char instr[100];
    int a,dlug,przep;
    char oustr[100];
#ifdef DEMO
    if(te!=7);
    {
        ini_karty();
        te=7;
    }
    strcpy(oustr,"*RST");
    dlug=strlen(oustr);
    a=ieoutput(16,oustr,dlug);
    strcpy(oustr,":conf:fres");
    dlug=strlen(oustr);
    a=ieoutput(16,oustr,dlug);
    strcpy(oustr,":sens:fres:nplc 8; dig 7.5; ocom on; aver:coun 5; tcon rep; stat on");
//strcpy(oustr,":sens:fres:nplc 8; dig 7.5");
    dlug=strlen(oustr);
    a=ieoutput(16,oustr,dlug);

    strcpy(oustr,":form:elem read");
    dlug=strlen(oustr);
    a=ieoutput(16,oustr,dlug);

    strcpy(oustr,":stat:meas:PTR 1; NTR 0");
    dlug=strlen(oustr);
    a=ieoutput(16,oustr,dlug);
    strcpy(oustr,":read?");
    dlug=strlen(oustr);
    a=ieoutput(16,oustr,dlug);

    a=ieenter(16,instr,100);
    strcpy(oustr,":stat:meas:even?");
    dlug=strlen(oustr);
    a=ieoutput(16,oustr,dlug);
    a=ieenter(16,oustr,100);
    przep=atoi(oustr);
    if (przep&&1==1)
//printf("przep = %d\n",przep);
    ;
    else
    ;
//printf("\n instr = %s\n",instr);

```

M

```

}
sprintf(wyn," %-6.3lf |%s|\r\n",blad_dp,ok);
wyniki += wyn;
sprintf(wyn, "|-----|-----|-----|-----|-----|-----|-----|\r\n");
wyniki += wyn;

```

```

////////////////////////////////////
tt1 = tmax ;
rez= rezyst(tt1,m_dRg,m_dAg,m_dBg);
TempNorma( rez, &tg);

```

```

////////////////////////////////////
tt2= tmax - dt_max;
blad_dp= 1.5;
rez = rezyst(tt2,m_dRz,m_dAz,m_dBz);
TempNorma( rez, &tz);
dt=tt1-tt2;
bl =100*(dt - (tg-tz))/(tg-tz);

```

```

if(fabs(bl) - blad_dp >0)
{
    sprintf(ok," ZLY ");
    m_nOKey = 0;
}
else
    sprintf(ok, " DOBRY ");
++poz;
sprintf(wyn,"| %-4d| %-4.0lf| %-4.0lf| %-4.0lf|",poz,tt1,tt2,dt);
wyniki += wyn;
if(fabs(bl)<1.0E3) { /* 1.0E3 */
    sprintf(wyn," %-9.3lf|",bl);
    wyniki += wyn;
}
else {
    sprintf(wyn," error |");
    wyniki += wyn;
}
sprintf(wyn," %-6.3lf |%s|\r\n",blad_dp,ok);
wyniki += wyn;
sprintf(wyn, "|-----|-----|-----|-----|-----|-----|-----|\r\n");
wyniki += wyn;

```

```

////////////////////////////////////
if(m_nOKey) {
    sprintf(wyn,"\r\n WYNIK BADANIA JEST POZYTYWNY ");
    wyniki += wyn;
}
else {
    sprintf(wyn,"\r\n WYNIK BADANIA JEST NEGATYWNY ");
    wyniki += wyn;
}
// CParyDoc::printf("%s",wyniki);

```

```

////////////////////////////////////
void CPomiar::TempNorma(double rezys, double* tem)
{
    //wsp. rownania r=r0*(1+a*t+b*t*t)
    double del=r_n*r_n*an*an-4*r_n*bn*(r_n-rezys);//wyznosc trojmianu
    *tem = (sqrt(del) - r_n*an)/(2 *r_n*bn);
}

```

```

////////////////////////////////////
double CPomiar::RezNorma(double t)
{
    double rezystancja = r_n*(1+an*t+bn*t*t);
    return rezystancja;
}

```

```

////////////////////////////////////
void CPomiar::TempWzorc(int nNrZlacza, double rezyst, double* temp)
{
    //ze swiadczenia czujnika
    double awz, bwz, r001;

    if( nNrZlacza==1) {
        awz=-0.00026383;
        bwz=-0.00004098;
        //r001=100.0;
    }
}

```

12

```

wyniki += wyn;
}
else {
    sprintf(wyn," error  |");
    wyniki += wyn;
}
sprintf(wyn," %-6.3lf |%s|\r\n",blad_dp.ok);
wyniki += wyn;
sprintf(wyn, "|-----|-----|-----|-----|-----|-----|-----|\r\n");
wyniki += wyn;
}

```

////////////////////////////////////

```

for( i=0; i<6; i++)
{
    temp_char= tmin + i * ((tmax-tmin)/5);
    tt1 = temp_char + 20;
    rez= rezyst(tt1,m_dRg,m_dAg,m_dBg);
    TempNorma( rez, &tg);
    //////////////////////////////////
    tt2= temp_char ;
    blad_dp= 1.5;
    rez = rezyst(tt2,m_dRz,m_dAz,m_dBz);
    TempNorma( rez, &tz);
    dt=tt1-tt2;
    bl =100*(dt - (tg-tz))/(tg-tz);

    if(fabs(bl) - blad_dp >0)
    {
        sprintf(ok," ZLY  ");
        m_nOKey = 0;
    }
    else
    sprintf(ok, " DOBRY  ");
    ++poz;
    sprintf(wyn,"| %-4d| %-4.0lf| %-4.0lf| %-4.0lf|",poz,tt1,tt2,dt);
    wyniki += wyn;
    if(fabs(bl)<1.0E3) { /* 1.0E3 */
        sprintf(wyn," %-9.3lf|",bl);
        wyniki += wyn;
    }
    else {
        sprintf(wyn," error  |");
        wyniki += wyn;
    }
    sprintf(wyn," %-6.3lf |%s|\r\n",blad_dp.ok);
    wyniki += wyn;
    sprintf(wyn, "|-----|-----|-----|-----|-----|-----|-----|\r\n");
    wyniki += wyn;
}

```

////////////////////////////////////

```

tt1 = tmin + dt_max;
rez= rezyst(tt1,m_dRg,m_dAg,m_dBg);
TempNorma( rez, &tg);
    //////////////////////////////////
    tt2= tmin ;
    blad_dp= 1.5;
    rez = rezyst(tt2,m_dRz,m_dAz,m_dBz);
    TempNorma( rez, &tz);
    dt=tt1-tt2;
    bl =100*(dt - (tg-tz))/(tg-tz);

    if(fabs(bl) - blad_dp >0)
    {
        sprintf(ok," ZLY  ");
        m_nOKey = 0;
    }
    else
    sprintf(ok, " DOBRY  ");
    ++poz;
    sprintf(wyn,"| %-4d| %-4.0lf| %-4.0lf| %-4.0lf|",poz,tt1,tt2,dt);
    wyniki += wyn;
    if(fabs(bl)<1.0E3) { /* 1.0E3 */
        sprintf(wyn," %-9.3lf|",bl);
        wyniki += wyn;
    }
    else {
        sprintf(wyn," error  |");
        wyniki += wyn;
    }

```

13

```

double tt1, tt2, dt, bl, blad_dp;
double rez, tg, tz;
printf(wyn, "\r\n TABELA WYNIKOW PARY CZUJNIKOW \r\n");
wyniki += wyn;
printf(wyn, "\r\n|-----|-----|-----|-----|-----|-----|-----|\r\n");
wyniki += wyn;
printf(wyn, "| Lp. | t1 | t2 | t1-t2 | Elt | Edop | ocena |\r\n");
wyniki += wyn;
printf(wyn, "|-----|-----|-----|-----|-----|-----|-----|\r\n");wyniki += wyn;
printf(wyn, "| °C | °C | °C | %% | %% | |\r\n");wyniki += wyn;
;
printf(wyn, "|-----|-----|-----|-----|-----|-----|-----|\r\n");wyniki += wyn;

```

```

////////////////////////////////////
for(int i=0; i<6; i++)
{
temp_char= tmin + i * ((80-dt_min)-tmin)/5);
tt1 = temp_char + dt_min;
rez= rezyst(tt1,m_dRg,m_dAg,m_dBg);
TempNorma( rez, &tg);
////////////////////////////////////
tt2= temp_char ;
blad_dp= 3.5;
rez = rezyst(tt2,m_dRz,m_dAz,m_dBz);
TempNorma( rez, &tz);
dt=tt1-tt2;
bl =100*(dt - (tg-tz))/(tg-tz);
if(fabs(bl) - blad_dp >0)
{
printf(ok, " ZLY ");
m_nOKey = 0;
}
else
printf(ok, " DOBRY ");
++poz;
printf(wyn, "| %-4d| %-4.01f| %-4.01f| %-4.01f|",poz,tt1,tt2,dt);
wyniki += wyn;
if(fabs(bl)<1.0E3) { /* 1.0E3 */
printf(wyn, " %-9.31f|",bl);
wyniki += wyn;
}
else {
printf(wyn, " error |");
wyniki += wyn;
}
printf(wyn, " %-6.31f |%s|\r\n",blad_dp,ok);
wyniki += wyn;
printf(wyn, "|-----|-----|-----|-----|-----|-----|-----|\r\n");
wyniki += wyn;
}

```

```

////////////////////////////////////
for( i=0; i<6; i++)
{
temp_char= tmin + i * ((tmax-tmin)/5);
tt1 = temp_char + 10;
rez= rezyst(tt1,m_dRg,m_dAg,m_dBg);
TempNorma( rez, &tg);
////////////////////////////////////
tt2= temp_char ;
blad_dp= 2.5;
rez = rezyst(tt2,m_dRz,m_dAz,m_dBz);
TempNorma( rez, &tz);
dt=tt1-tt2;
bl =100*(dt - (tg-tz))/(tg-tz);
if(fabs(bl) - blad_dp >0)
{
printf(ok, " ZLY ");
m_nOKey = 0;
}
else
printf(ok, " DOBRY ");
++poz;
printf(wyn, "| %-4d| %-4.01f| %-4.01f| %-4.01f|",poz,tt1,tt2,dt);
wyniki += wyn;
if(fabs(bl)<1.0E3) { /* 1.0E3 */
printf(wyn, " %-9.31f|",bl);

```

14

```

        wyniki += wyn;
    }
    else {
        sprintf(wyn,"  error  |");
        wyniki += wyn;
    }
    sprintf(wyn,"  %-6.2lf |%s|\r\n",blad_dp,ok);
    wyniki += wyn;
    sprintf(wyn, "|-----|-----|-----|-----|-----|-----|-----|-----|\r\n");
    wyniki += wyn;
}

```

////////////////////////////////////

```

void CPomiar::SprawdzCzuj2()
{
    double temp_char, rez, tn, rn,dr, bl, blad_dp;

    ObliczRAB( m_d2z, &m_dRz,&m_dAz,&m_dBz);

    sprintf(wyn,"\r\n  TABELA WYNIKOW CZUJNIKA WODY ZIMNEJ \r\n");
    wyniki += wyn;
    sprintf(wyn,"\r\n|-----|-----|-----|-----|-----|-----|-----|-----|\r\n");
    wyniki += wyn;
    sprintf(wyn,"|  Lp. |  t  | Robl | Rnorm |  dR  |  Et  |  Edop | ocena |\r\n");
    wyniki += wyn;
    sprintf(wyn, "|-----|-----|-----|-----|-----|-----|-----|-----|\r\n");wyniki
+= wyn;
    sprintf(wyn, "|  -  |  °C  |  om  |  om  |  om  |  °C  |  °C  |  |\r\n");wyniki
+= wyn;
    sprintf(wyn, "|-----|-----|-----|-----|-----|-----|-----|-----|\r\n");wyniki
+= wyn;

    for(int i=0; i<6; i++)
    {
        temp_char= tmin + i * ((tmax-tmin)/5);
        rez = rezyst(temp_char,m_dRg,m_dAg,m_dBg);
        TempNorma( rez, &tn);
        rn=RezNorma(temp_char);
        dr=rez-rn;
        bl = tn - temp_char;
        blad_dp = 2.0;
// (0.30+0.005*temp_char)*r0n*(an+2*bn*temp_char);
        if(fabs(bl) - blad_dp >0)
        {
            sprintf(ok,"  ZLY  ");
            m_nOKey = 0;
        }
        else
        sprintf(ok, "  DOBRY  ");
        sprintf(wyn,"|  %-4d|  %-4.0lf |%-6.2lf |%-6.2lf |  %-6.2lf |",i+1,temp_char,rez,rn,dr);
        wyniki += wyn;
        if(fabs(bl)<1.0E3) { /* 1.0E3 */
            sprintf(wyn,"  %-9.3lf|",bl);
            wyniki += wyn;
        }
        else {
            sprintf(wyn,"  error  |");
            wyniki += wyn;
        }
        sprintf(wyn,"  %-6.2lf |%s|\r\n",blad_dp,ok);
        wyniki += wyn;
        sprintf(wyn, "|-----|-----|-----|-----|-----|-----|-----|-----|\r\n");
        wyniki += wyn;
    }
}

```

////////////////////////////////////

```

void CPomiar::ObliczPary()
{
    double temp_char;
    int poz = 0;

```

15

```
170.0, 70.0, 1.25
};
```

```
struct D2 {
double t;
double r;
} m_d2g[3];
// struct D2 m_d2g[3];
struct D2 m_d2z[3];

double te;

void CPomiar::ObliczRAB(struct D2 *s ,double* R0, double* A, double* B)
{
double w1=(s[0].r*s[1].t-s[1].r*s[0].t)*(s[1].r*s[2].t*s[2].t-s[2].r*s[1].t*s[1].t)-(s[0].r*s[1].t*s[1].t-s[1].r*s[0].t*s[0].t)*(s[1].r*s[2].t-s[2].r*s[1].t);//
double w2=(s[1].t-s[0].t)*(s[2].t-s[1].t)*(s[2].t-s[0].t); //warunek na trzy rozne punkty t
if (w1*w2==0)
return;
double AA=(s[1].r-s[0].r)*(s[1].r*s[2].t*s[2].t-s[2].r*s[1].t*s[1].t)-(s[0].r*s[1].t*s[1].t-s[1].r*s[0].t*s[0].t)*(s[2].r-s[1].r);
AA=AA/w1;
double BB=(s[0].r*s[1].t-s[1].r*s[0].t)*(s[2].r-s[1].r)-(s[1].r*s[2].t-s[2].r*s[1].t)*(s[1].r-s[0].r);
BB=BB/w1;
double RR=s[0].r/(1+AA*s[0].t+BB*s[0].t*s[0].t);
*A = AA;
*B = BB;
*R0= RR;
}

////////////////////////////////////
void CPomiar::SprawdzCzujl()
{
CString wyniki_new;
wyniki = wyniki_new;
sprintf(ok," ZLY ");
m_nOKey =1;
double temp_char, rez, tn, rn,dr, bl, blad_dp;

ObliczRAB( m_d2g, &m_dRg,&m_dAg,&m_dBg);

sprintf(wyn, "\r\n TABELA WYNIKOW CZUJNIKA WODY GORACEJ \r\n");
wyniki += wyn;
sprintf(wyn, "\r\n|-----|-----|-----|-----|-----|-----|-----|-----|\r\n");
wyniki += wyn;
sprintf(wyn, "| Lp. | t | Robl | Rnorm | dR | Et | Edop | ocena |\r\n");
wyniki += wyn;
sprintf(wyn, "|-----|-----|-----|-----|-----|-----|-----|-----|\r\n");wyniki
+= wyn;
sprintf(wyn, "| - | °C | om | om | om | °C | °C | |\r\n"); ki
+= wyn;
sprintf(wyn, "|-----|-----|-----|-----|-----|-----|-----|-----|\r\n");wyniki
+= wyn;

for(int i=0; i<6; i++)
{
temp_char= tmin + i * ((tmax-tmin)/5);
rez = rezyst(temp_char,m_dRg,m_dAg,m_dBg);
TempNorma( rez, &tn);
rn=RezNorma(temp_char);
dr=rez-rn;
bl = tn - temp_char;
blad_dp = 2.0;
// (0.30+0.005*temp_char)*rn*(an+2*bn*temp_char);
if(fabs(bl) - blad_dp >0)
{
sprintf(ok," ZLY ");
m_nOKey = 0;
}
else
sprintf(ok, " DOBRY ");
sprintf(wyn, "| %-4d| %-4.0lf |%-6.2lf |%-6.2lf | %-6.2lf |",i+1,temp_char,rez,rn,dr);
wyniki += wyn;
if(fabs(bl)<1.0E3) { /* 1.0E3 */
sprintf(wyn, " %-9.3lf|",bl);
}
}
}
```

16



```

/* pomiar.cpp */
////////////////////////////////////

#include "stdafx.h"
#include "pomiar.h"
#include "math.h"
#include "conio.h"

/*
extern "C" int ieseg(int);
extern "C" void ieinit(int,int,int);
extern "C" int ieoutput(int,char*,int);
extern "C" int ieabort(void);
extern "C" int ieenter(int,char*,int);
*/
#include "iewinlib.h"

#define bn      -5.802e-7
#define an      0.00390802

#define d0      439.932854
#define d1      472.418020
#define d2      37.684494
#define d3      7.472018
#define d4      2.920828
#define d5      0.005184
#define d6      -0.963864
#define d7      -0.188732
#define d8      0.191203
#define d9      0.049025

/*****/
void ini_karty(void)
{
int a;
ieseg(0xD000);
ieinit(0x2b8,21,0x100);
ieabort();
}
////////////////////////////////////
void CPomiar::Delay(int czas)
{
CTime poczatek = CTime::GetCurrentTime();
//CParyDoc::printf("\r\n Czekaaj %d sek. *",czas);
while(1)
{
CTime teraz= CTime::GetCurrentTime();
CTimeSpan roznica = teraz - poczatek;
if( czas<= (int) roznica.GetTotalSeconds() )
break;
while(1)
{
CTime sek= CTime::GetCurrentTime();
roznica = sek - teraz;
if( 1 <= (int) roznica.GetTotalSeconds() )
; //CParyDoc::printf("*");
break;
}
}
}
////////////////////////////////////
////////////////////////////////////
#define rezyst(t,RO,A,B) (RO*(1+A*t+B*t*t))

struct
{
double temp_g;
double temp_z;
double bl_dop;
} oblp[6]= { 33.0, 30.0, 0.105,
45.0, 40.0, 0.165,
80.0, 70.0, 0.25,
104.0, 85.0, 0.475,
150.0, 110.0, 0.5,

```

17