UHRwin: Time application for MS Windows

Dietmar G. Schrausser

2023

Overview

Time application for MS Windows (Schrausser, 2023).

C++ Source of main functions

```
//-------------------------------------------------------------------------| UHRwinDLG.cpp
//
//                                                                         | UHRwin4, System
Zeit und Datum (deutsch)     //
//                                                                         | von Dietmar
Schrausser, (C) SCHRAUSSER 2011  //
//

#include "stdafx.h"
#include "UHRwin.h"
#include "UHRwinDlg.h"
#include "UHRwinEinst.h"
#include "math.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

//-----------------------------------------------------------| variablen deklaration
int fb_hg, fb_hg_0;
int fb_in, fb_in_0;
int fb_um, fb_um_0;
int fb_uh, fb_uh_0;
int fb_sz, fb_sz_0;
int fb_sl, fb_sl_0;
int fb_mz, fb_mz_0;
int fb_ml, fb_ml_0;
int fb_kz, fb_kz_0;
int fb_kl, fb_kl_0;
int sw_uhr, sw_uhr_0;
int sw_stzg, sw_stzg_0;
int sw_mnzg, sw_mnzg_0;
int sw_skzg, sw_skzg_0;

int sw_mv_u=0, sw_mv_i=0;

//-----------------------------------------------------------| variableninitialisierung CAboutDlg
CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
	//{{AFX_DATA_INIT(CAboutDlg)
	//}}AFX_DATA_INIT
}

//-----------------------------------------------------------| steuerelement-initialisierung
CAboutDlg
void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	//{{AFX_DATA_MAP(CAboutDlg)
	//}}AFX_DATA_MAP
}
```

```
//-------------------------------------------------------| ereignis-deklaration CAboutDlg
BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
	//{{AFX_MSG_MAP(CAboutDlg)
	//}}AFX_MSG_MAP
END_MESSAGE_MAP()

//-------------------------------------------------------| variableninitialisierung
CUHRwinDlg
CUHRwinDlg::CUHRwinDlg(CWnd* pParent)
	: CDialog(CUHRwinDlg::IDD, pParent)
{
	//{{AFX_DATA_INIT(CUHRwinDlg)
	//}}AFX_DATA_INIT
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
	m_Csr1  = AfxGetApp()->LoadCursor(IDC_CURSOR1);
}

//-------------------------------------------------------| steuerelement-initialisierung
CUHRwinDlg
void CUHRwinDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	//{{AFX_DATA_MAP(CUHRwinDlg)
	//}}AFX_DATA_MAP
}


//-------------------------------------------------------| ereignis-deklaration CUHRwinDlg
BEGIN_MESSAGE_MAP(CUHRwinDlg, CDialog)
	//{{AFX_MSG_MAP(CUHRwinDlg)
	ON_WM_SYSCOMMAND()
	ON_WM_PAINT()
	ON_WM_QUERYDRAGICON()
	ON_WM_SIZE()
	ON_WM_CLOSE()
	ON_WM_MOVE()
	ON_WM_TIMER()
	ON_WM_MOUSEWHEEL()
	ON_WM_MOUSEMOVE()
	ON_WM_LBUTTONDBLCLK()
	ON_WM_LBUTTONDOWN()
	ON_WM_LBUTTONUP()
	ON_WM_RBUTTONDOWN()
	ON_WM_RBUTTONUP()
	ON_WM_RBUTTONDBLCLK()
	//}}AFX_MSG_MAP
END_MESSAGE_MAP()


//-------------------------------------------------------| bei dialogstart CUHRwinDlg
BOOL CUHRwinDlg::OnInitDialog()
{
	CDialog::OnInitDialog();

	c0="0";   //display memory variable
	szsw=0;  //schalter window size

	SetTimer(1,50,0);



	CWinApp*  pApp = AfxGetApp(); // ini profil lesen

	dlg0.x  = pApp->GetProfileInt("Fensterposition","x",100);
	dlg0.y  = pApp->GetProfileInt("Fensterposition","y",100);
	dlg_.x  = pApp->GetProfileInt("Fenstergrösse","x",200);
	dlg_.y  = pApp->GetProfileInt("Fenstergrösse","y",100);
	d_      = pApp->GetProfileInt("Uhr","Radius",3);
	sw      = pApp->GetProfileInt("Uhr","Modus",5);
	swd     = pApp->GetProfileInt("Uhr","Zahlen",-1);
	u_vs.x  = pApp->GetProfileInt("Uhr","Verschub_x",0);
	u_vs.y  = pApp->GetProfileInt("Uhr","Verschub_y",0);
	i_vs.x  = pApp->GetProfileInt("Info","Verschub_x",0);
	i_vs.y  = pApp->GetProfileInt("Info","Verschub_y",0);
	fb_hg   = pApp->GetProfileInt("Farbe","Hintergrund",13357270);
	fb_in   = pApp->GetProfileInt("Farbe","Info",8421504);
	fb_um   = pApp->GetProfileInt("Farbe","Uhrrahmen",16777215);
	fb_uh   = pApp->GetProfileInt("Farbe","Uhrhintergrund",13357270);
```

```
        fb_sz   = pApp->GetProfileInt("Farbe","Stundenzeiger",16777215);
        fb_sl   = pApp->GetProfileInt("Farbe","Stundenzahl",0);
        fb_mz   = pApp->GetProfileInt("Farbe","Minutenzeiger",16777215);
        fb_ml   = pApp->GetProfileInt("Farbe","Minutenzahl",0);
        fb_kz   = pApp->GetProfileInt("Farbe","Sekundenzeiger",16777215);
        fb_kl   = pApp->GetProfileInt("Farbe","Sekundenzahl",0);
        sw_uhr  = pApp->GetProfileInt("Darstellung","Uhr",1);
        sw_stzg = pApp->GetProfileInt("Darstellung","Stundenzeiger",1);
        sw_mnzg = pApp->GetProfileInt("Darstellung","Minutenzeiger",1);
        sw_skzg = pApp->GetProfileInt("Darstellung","Sekundenzeiger",1);

        fb_hg_0=fb_hg;     // einstellungs memory variablen
        fb_in_0=fb_in;     //
        fb_um_0=fb_um;     //
        fb_uh_0=fb_uh;     //
        fb_sz_0=fb_sz;     //
        fb_sl_0=fb_sl;     //
        fb_mz_0=fb_mz;     //
        fb_ml_0=fb_ml;     //
        fb_kz_0=fb_kz;     //
        fb_kl_0=fb_kl;     //
        sw_uhr_0=sw_uhr;   //
        sw_stzg_0=sw_stzg; //
        sw_mnzg_0=sw_mnzg; //
        sw_skzg_0=sw_skzg; //

        if(dlg_.x ==0)SetWindowPos(&wndTop, 100,100, 200+8,100+27, SWP_SHOWWINDOW);
        if(dlg_.x !=0)SetWindowPos(&wndTop, dlg0.x,dlg0.y, dlg_.x+8,dlg_.y+27, SWP_SHOWWINDOW);


        ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
        ASSERT(IDM_ABOUTBOX < 0xF000);

        CMenu*  zgr = GetSystemMenu(0);

                        zgr->AppendMenu(MF_SEPARATOR);
                        zgr->AppendMenu(MF_STRING, 0x0005, "analog");
                        zgr->AppendMenu(MF_STRING, 0x0010, "+/-Zeit");
                        zgr->AppendMenu(MF_SEPARATOR);
                        zgr->AppendMenu(MF_STRING, 0x0012, "+digital");
                        zgr->AppendMenu(MF_STRING, 0x0015, "+Datum");
                        zgr->AppendMenu(MF_STRING, 0x0020, "+Tag/Monat");
                        zgr->AppendMenu(MF_STRING, 0x0030, "+Julianisch");
                        zgr->AppendMenu(MF_STRING, 0x0040, "+time(0)");
                        zgr->AppendMenu(MF_SEPARATOR);
                        zgr->AppendMenu(MF_STRING, 0x0045, "Einstellungen...");
                        zgr->AppendMenu(MF_SEPARATOR);
                        zgr->AppendMenu(MF_STRING, 0x0050, "Information");

        MenueMod();

        SetIcon(m_hIcon, 1);SetIcon(m_hIcon, 0);return 1;
}

//-------------------------------------------------------------| system-menue-punkte CUHRwinDlg
void CUHRwinDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
        CMenu*  zgr = GetSystemMenu(0);

        if (nID == 0x0005){  sw=5;   MenueMod();OnTimer(0); RedrawWindow();   }
        if (nID == 0x0010){  swd*=-1;MenueMod();OnTimer(0); RedrawWindow();   }
        if (nID == 0x0012){  sw=0;   MenueMod();OnTimer(0); RedrawWindow();   }
        if (nID == 0x0015){  sw=1;   MenueMod();OnTimer(0); RedrawWindow();   }
        if (nID == 0x0020){  sw=2;   MenueMod();OnTimer(0); RedrawWindow();   }
        if (nID == 0x0030){  sw=3;   MenueMod();OnTimer(0); RedrawWindow();   }
        if (nID == 0x0040){  sw=4;   MenueMod();OnTimer(0); RedrawWindow();   }
        if (nID == 0x0045){  UHRwinEinst o; o.DoModal();          }
        if (nID == 0x0050){
                                m_InfoDlg.DestroyWindow(); //<--

                                m_InfoDlg.Create(IDD_ABOUTBOX );
                                m_InfoDlg.ShowWindow(SW_SHOW);
                                m_InfoDlg.BringWindowToTop();         }

        else{CDialog::OnSysCommand(nID, lParam);}
}

void CUHRwinDlg::MenueMod()
```

```
{
         CMenu*  zgr = GetSystemMenu(0);

                 zgr->CheckMenuItem(0x0010, MF_UNCHECKED);
                    zgr->CheckMenuItem(0x0012, MF_UNCHECKED);
                    zgr->CheckMenuItem(0x0015, MF_UNCHECKED);
                    zgr->CheckMenuItem(0x0020, MF_UNCHECKED);
                    zgr->CheckMenuItem(0x0030, MF_UNCHECKED);
                    zgr->CheckMenuItem(0x0040, MF_UNCHECKED);

        if(swd==1)zgr->CheckMenuItem(0x0010, MF_CHECKED);
        if(sw==0) zgr->CheckMenuItem(0x0012, MF_CHECKED);
        if(sw==1) zgr->CheckMenuItem(0x0015, MF_CHECKED);
        if(sw==2) zgr->CheckMenuItem(0x0020, MF_CHECKED);
        if(sw==3) zgr->CheckMenuItem(0x0030, MF_CHECKED);
        if(sw==4) zgr->CheckMenuItem(0x0040, MF_CHECKED);
}

//-------------------------------------------------------------| OnPaint CUHRwinDlg
void CUHRwinDlg::OnPaint()
{
        UpdateWindow();
        GetWindowRect(&coord); //fensterrechteckkoordinaten in RECT strukturpointer coord

        CPaintDC ooo(this);

                ooo.SetBkColor(fb_hg);//hintergrundfarbe

                                CRect or(0, 0, dlg_x, dlg_y);//hintergrund
                        ooo.FillSolidRect(or, fb_hg );

                                 CPen oo;
                                oo.CreatePen(PS_SOLID,3,fb_um); //linien -breite,-farbe
                        ooo.SelectObject(&oo);

                                  CBrush o1;
                                o1.CreateSolidBrush(fb_uh);//füllfarbe
                        ooo.SelectObject(&o1);

                                       CFont o; //schriftart
                                    o.CreateFont(13,4,0,0,400,0,0,0,0,OUT_DEFAULT_PRECIS,

CLIP_DEFAULT_PRECIS,

                                DEFAULT_QUALITY,

                                DEFAULT_PITCH,

                                "Arial" );
                        ooo.SelectObject(&o);

                                     x0_=dlg_x/d_;
                                             y0_=dlg_y/d_;

                                               x_=x0_*(d_-1);
                                              y_=y0_*(d_-1);
                        if(sw_uhr==1)
                        ooo.RoundRect(x0_ +u_vs.x, y0_ +u_vs.y,
                                         x_  +u_vs.x, y_  +u_vs.y,
                                                x_  +u_vs.x, y_  +u_vs.y); //uhrkörper


                        //uhr zeiger

                                                 CPen oo1;
                                        oo1.CreatePen(PS_SOLID,1,fb_kz); //linien -breite,-farbe
sekundenzeiger
                        ooo.SelectObject(&oo1);

                        //sekundenzeiger koordinaten berechnung
                        ooo.MoveTo(dlg_x/2 +u_vs.x,
                                      dlg_y/2 +u_vs.y);

                                                  a_= cos((3.1415926535897931/180)* (450 -
(360/60)*atoi(c)));
                                                  b_= sin((3.1415926535897931/180)* (450 -
(360/60)*atoi(c)));
```

```
                                        x_= a_*(1.0*dlg_x/dlg_.x)*(dlg_.x/( d_/  ((d_-2)
/2.0) ));
                                        y_= b_*(1.0*dlg_y/dlg_.y)*(dlg_.y/( d_/  ((d_-2)
/2.0) ));

                        if(sw_skzg==1)
                        ooo.LineTo((dlg_x/2 +x_ +u_vs.x),
                                    (dlg_y/2 -y_ +u_vs.y)); //sekunden zeiger


                        ooo.SetTextColor(fb_kl); //textfarbe sekunden zahl
                        ooo.SetBkMode(TRANSPARENT);

                                                itoa(atoi(c),ccc,10);
                                                cc=ccc;
                        if(swd==1)
                        ooo.TextOut(dlg_x/2 +x_ +u_vs.x,
                                    dlg_y/2 -y_ +u_vs.y, cc); //sekunden zahl

                                            CPen oo2;
                                oo2.CreatePen(PS_SOLID,3,fb_mz); //linien -breite,-farbe
minutenzeiger
                        ooo.SelectObject(&oo2);

                        //minutenzeiger koordinaten berechnung
                        ooo.MoveTo(dlg_x/2 +u_vs.x,
                                    dlg_y/2 +u_vs.y);

                                                                a_=
cos((3.1415926535897931/180)* (450 - (360/60)*atoi(cm)));
                                                                b_=
sin((3.1415926535897931/180)* (450 - (360/60)*atoi(cm)));

                                                            x_=
a_*(1.0*dlg_x/dlg_.x)*(dlg_.x/( d_/  ((d_-2) /2.0) ));
                                                        y_= b_*(1.0*dlg_y/dlg_.y)*(dlg_.y/(
d_/  ((d_-2) /2.0) ));

                        if(sw_mnzg==1)
                        ooo.LineTo((dlg_x/2 +x_/1.2) +u_vs.x,
                                    (dlg_y/2 -y_/1.2) +u_vs.y);//minuten zeiger

                        ooo.SetTextColor(fb_ml); //textfarbe minuten zahl

                        if(swd==1)
                        ooo.TextOut((dlg_x/2+x_/1.2)+u_vs.x,(dlg_y/2-y_/1.2)+u_vs.y,cm);
//minuten zahl

                                    CPen oo3;
                                oo3.CreatePen(PS_SOLID,4,fb_sz); //linien -breite,-farbe
stundenzeiger
                        ooo.SelectObject(&oo3);

                         //stundenzeiger koordinaten berechnung
                        ooo.MoveTo(dlg_x/2 +u_vs.x,
                                    dlg_y/2 +u_vs.y);

                                                                a_=
cos((3.1415926535897931/180)* (450 - (360/12)*atoi(cs)));
                                                            b_= sin((3.1415926535897931/180)*
(450 - (360/12)*atoi(cs)));

                                                            x_=
a_*(1.0*dlg_x/dlg_.x)*(dlg_.x/( d_/  ((d_-2) /2.0) ));
                                                        y_= b_*(1.0*dlg_y/dlg_.y)*(dlg_.y/(
d_/  ((d_-2) /2.0) ));
                        if(sw_stzg==1)
                        ooo.LineTo((dlg_x/2 +x_/1.7) +u_vs.x,
                                    (dlg_y/2 -y_/1.7) +u_vs.y);//stunden zeiger

                        ooo.SetTextColor(fb_sl); //textfarbe stunden zahl

                        if(swd==1)
                        ooo.TextOut(dlg_x/2 +u_vs.x -5,
                                    dlg_y/2 +u_vs.y -5,cs1); //stunden zahl

                        if(szsw==0) { dlgx0=dlg_x;dlgy0=dlg_y; } //bei !windowsize
```

```
                        if(sw!=5) //info ausgabe
                        {
                                ooo.SetTextColor(fb_in);

                                ooo.TextOut(     1.0*dlg_x/20 +i_vs.x*(1.0*dlg_x/dlgx0),
                                                18*(1.0*dlg_y/20)+i_vs.y*(1.0*dlg_y/dlgy0)
,cd);
                        }

        CDialog::OnPaint();
}


HCURSOR CUHRwinDlg::OnQueryDragIcon(){return (HCURSOR) m_hIcon;}

//-------------------------------------------------------------| timer ereignisse CUHRwinDlg
void CUHRwinDlg::OnTimer(UINT nIDEvent)
{
        // zeit und datums berechnung

        double tag_1, tag;
        int tagnr;

        char zeit[9], datum[9], std[2], min[2], sec[2], mon[2], ccc[1000];

        _strdate( datum );
        _strtime( zeit );

        strcpy(sec,"");
        strcpy(min,"");

        sprintf(std, "%c%c", zeit[0], zeit[1]);
        sprintf(min, "%c%c", zeit[3], zeit[4]);
        sprintf(sec, "%c%c", zeit[6], zeit[7]);
        sprintf(mon, "%c%c", datum[0], datum[1]);

                  tag_1=time(0)-(atoi(std)*3600 + atoi(min)*60 + atoi(sec))+7200;
            tag = tag_1/(86400*7);
         tag = floor((tag-floor(tag))*100);

        if (tag != 0 &&
                tag != 14 &&
                tag != 28 &&
                tag != 42 &&
                tag != 57 &&
                tag != 71 &&
                tag != 85 )
        {
                          tag_1=time(0)-(atoi(std)*3600 + atoi(min)*60 + atoi(sec))+3600;
                tag = tag_1/(86400*7);
                tag = floor((tag-floor(tag))*100);
        }
                         tagnr=(tag_1/86400)+2440588;
        if(atoi(std)<12)tagnr=tagnr-1;

                 cs ="";
                 cs +=zeit[0];
                 cs +=zeit[1];

                 cm  =zeit[3];
                 cm +=zeit[4];

                 c  =zeit[6];
                 c +=zeit[7];

                cs1=cs;
        if(atoi(cs)>12) //pm
        {
                         tmp=atoi(cs);
                         tmp-=12;
                itoa(tmp,ccc,10);
                          cs=ccc;
        }

        if(sw==0) //digitalzeit
        {
                        cd ="";
                 cd +=zeit[0];
```

```
        cd +=zeit[1];
               cd +=":";
        cd +=zeit[3];
        cd +=zeit[4];
               cd +=":";
        cd +=zeit[6];
        cd +=zeit[7];
}

if(sw==1) //datum
{
       cd = datum[3];
       cd +=datum[4];
       cd +="-";
       cd +=datum[0];
       cd +=datum[1];
       cd +="-20";
       cd +=datum[6];
       cd +=datum[7];
}

if(sw==2) //wochentag und monat
{
       if(tag == 0)       cd ="Donnerstag,";
       if(tag == 14)      cd ="Freitag,";
       if(tag == 28)         cd ="Samstag,";
       if(tag == 42)      cd ="Sonntag,";
       if(tag == 57)      cd ="Montag,";
       if(tag == 71)      cd ="Dienstag,";
       if(tag == 85)      cd ="Mittwoch,";
                                              cd+=" ";
       if(datum[3] !='0')  cd+= datum[3];
                                              cd+= datum[4];
                                              cd+=". ";
       if(atoi(mon) ==  1) cd+="J\x84nner";
       if(atoi(mon) ==  2) cd+="Februar";
       if(atoi(mon) ==  3) cd+="M\x84rz";
       if(atoi(mon) ==  4) cd+="April";
       if(atoi(mon) ==  5) cd+="Mai";
       if(atoi(mon) ==  6) cd+="Juni";
       if(atoi(mon) ==  7) cd+="Juli";
       if(atoi(mon) ==  8) cd+="August";
       if(atoi(mon) ==  9) cd+="September";
       if(atoi(mon) == 10) cd+="Oktober";
       if(atoi(mon) == 11) cd+="November";
       if(atoi(mon) == 12) cd+="Dezember";
}

if(sw==3) //julianisch
{
              itoa(tagnr,ccc,10);
                   cc=ccc;
                      cd = "julianisch: ";
                      cd +=cc;
}

if(sw==4) //time(0)
{
         itoa(time(0),ccc,10);
                   cc=ccc;
                      cd = "time(0): ";
                      cd +=cc;
}



if(c!=c0){RedrawWindow();c0=c;}  //bei zeitänderung


if(fb_hg_0 !=fb_hg ||            //bei einstellungsänderung
   fb_in_0 !=fb_in ||
   fb_um_0 !=fb_um ||
   fb_uh_0 !=fb_uh ||
   fb_sz_0 !=fb_sz ||
   fb_sl_0 !=fb_sl ||
   fb_mz_0 !=fb_mz ||
   fb_ml_0 !=fb_ml ||
   fb_kz_0 !=fb_kz ||
```

```
           fb_kl_0 !=fb_kl ||
           sw_uhr_0 !=sw_uhr ||
           sw_stzg_0 !=sw_stzg ||
           sw_mnzg_0 !=sw_mnzg ||
           sw_skzg_0 !=sw_skzg
           )
        {
               fb_hg_0= fb_hg;
               fb_in_0= fb_in;
               fb_um_0= fb_um;
               fb_uh_0= fb_uh;
               fb_sz_0= fb_sz;
               fb_sl_0= fb_sl;
               fb_mz_0= fb_mz;
               fb_ml_0= fb_ml;
               fb_kz_0= fb_kz;
               fb_kl_0= fb_kl;
               sw_uhr_0 =sw_uhr;
               sw_stzg_0 =sw_stzg;
               sw_mnzg_0 =sw_mnzg;
               sw_skzg_0 =sw_skzg;

               RedrawWindow();
        }

        CDialog::OnTimer(nIDEvent);
}

//-------------------------------------------------------------| click ereignisse CUHRwinDlg
void CUHRwinDlg::OnLButtonDblClk(UINT nFlags, CPoint point)
{
        SetCursor(m_Csr1);

        swd*=-1; MenueMod(); RedrawWindow();

        CDialog::OnLButtonDblClk(nFlags, point);
}

void CUHRwinDlg::OnRButtonDblClk(UINT nFlags, CPoint point)
{
        SetCursor(m_Csr1);
           sw++;
        if(sw==6)sw=0;

        MenueMod(); RedrawWindow();

        CDialog::OnRButtonDblClk(nFlags, point);
}

void CUHRwinDlg::OnLButtonDown(UINT nFlags, CPoint point){ sw_mv_u=1; SetCursor(m_Csr1);
CDialog::OnLButtonDown(nFlags, point);}
void CUHRwinDlg::OnLButtonUp(UINT nFlags, CPoint point)  { sw_mv_u=0; SetCursor(m_Csr1);
CDialog::OnLButtonUp(nFlags, point);  }
void CUHRwinDlg::OnRButtonDown(UINT nFlags, CPoint point){ sw_mv_i=1; SetCursor(m_Csr1);
CDialog::OnRButtonDown(nFlags, point);}
void CUHRwinDlg::OnRButtonUp(UINT nFlags, CPoint point)  { sw_mv_i=0; SetCursor(m_Csr1);
CDialog::OnRButtonUp(nFlags, point);  }

//
void CUHRwinDlg::OnMove(int x, int y)
{
        CDialog::OnMove(x, y);

        RedrawWindow();
}

//-------------------------------------------------------------| OnSize ereignis CUHRwinDlg
void CUHRwinDlg::OnSize(UINT nType, int cx, int cy)
{
        CDialog::OnSize(nType, cx, cy);

        szsw=1;

        dlg_x=cx; dlg_y=cy;

        RedrawWindow();
}
```

```
//-------------------------------------------------------------| OnMouseWheel CUHRwinDlg

BOOL CUHRwinDlg::OnMouseWheel(UINT nFlags, short zDelta, CPoint pt) //uhr umfang
{
	SetCursor(m_Csr1);

	if(zDelta< 0 && d_ <=15)d_*=1.17;
	if(zDelta>=0 && d_ >=3) d_/=1.17;

	RedrawWindow();

	return CDialog::OnMouseWheel(nFlags, zDelta, pt);
}

//-------------------------------------------------------------| OnMouseMove CUHRwinDlg

void CUHRwinDlg::OnMouseMove(UINT nFlags, CPoint point) //verschub
{
	SetCursor(m_Csr1);

	if(nFlags==MK_LBUTTON && sw_mv_u==1) //uhr verschub
	{
		u_vs.x= point.x-dlg_x/2.0;
		u_vs.y= point.y-dlg_y/2.0;

		RedrawWindow();
	}

	if(nFlags==MK_RBUTTON && sw_mv_i==1) //info verschub
	{

		i_vs.x=  ( point.x-(    1.0*dlg_x/20 ) )/(1.0*dlg_x/dlgx0);
		i_vs.y=  ( point.y-(18*(1.0*dlg_y/20)) )/(1.0*dlg_y/dlgy0);

		RedrawWindow();
	}

	CDialog::OnMouseMove(nFlags, point);
}

//-------------------------------------------------------------| OnClose CUHRwinDlg
void CUHRwinDlg::OnClose()
{
	CWinApp* pApp = AfxGetApp(); // ini profil schreiben

		 pApp->WriteProfileInt("Fensterposition","x",coord.left);
			pApp->WriteProfileInt("Fensterposition","y",coord.top);
		pApp->WriteProfileInt("Fenstergrösse","x",dlg_x);
		pApp->WriteProfileInt("Fenstergrösse","y",dlg_y);
			pApp->WriteProfileInt("Uhr","Radius",d_);
			pApp->WriteProfileInt("Uhr","Modus",sw);
		pApp->WriteProfileInt("Uhr","Zahlen",swd);
		pApp->WriteProfileInt("Uhr","Verschub_x",u_vs.x);
		pApp->WriteProfileInt("Uhr","Verschub_y",u_vs.y);
		pApp->WriteProfileInt("Info","Verschub_x",i_vs.x);
		pApp->WriteProfileInt("Info","Verschub_y",i_vs.y);
			pApp->WriteProfileInt("Farbe","Hintergrund",fb_hg);
			pApp->WriteProfileInt("Farbe","Info",fb_in );
			pApp->WriteProfileInt("Farbe","Uhrrahmen",fb_um);
		   pApp->WriteProfileInt("Farbe","Uhrhintergrund",fb_uh);
			pApp->WriteProfileInt("Farbe","Stundenzeiger",fb_sz);
		pApp->WriteProfileInt("Farbe","Stundenzahl",fb_sl);
			pApp->WriteProfileInt("Farbe","Minutenzeiger",fb_mz);
			pApp->WriteProfileInt("Farbe","Minutenzahl",fb_ml  );
			pApp->WriteProfileInt("Farbe","Sekundenzeiger",fb_kz);
			pApp->WriteProfileInt("Farbe","Sekundenzahl",fb_kl);
			pApp->WriteProfileInt("Darstellung","Uhr",sw_uhr);
			pApp->WriteProfileInt("Darstellung","Stundenzeiger",sw_stzg);
			pApp->WriteProfileInt("Darstellung","Minutenzeiger",sw_mnzg);
		pApp->WriteProfileInt("Darstellung","Sekundenzeiger",sw_skzg);

	CDialog::OnClose();
}

//-------------------------------------------------------------| einstellungsvariablenübernahme
funktionen
void CUHRwinDlg::fhg_(int fhg){fb_hg=fhg;}
void CUHRwinDlg::fin_(int fin){fb_in=fin;}
```

```
void CUHRwinDlg::fum_(int fum){fb_um=fum;}
void CUHRwinDlg::fuh_(int fuh){fb_uh=fuh;}
void CUHRwinDlg::fsz_(int fsz){fb_sz=fsz;}
void CUHRwinDlg::fsl_(int fsl){fb_sl=fsl;}
void CUHRwinDlg::fmz_(int fmz){fb_mz=fmz;}
void CUHRwinDlg::fml_(int fml){fb_ml=fml;}
void CUHRwinDlg::fkz_(int fkz){fb_kz=fkz;}
void CUHRwinDlg::fkl_(int fkl){fb_kl=fkl;}
void CUHRwinDlg::uhr_(int uhr){sw_uhr=uhr;}
void CUHRwinDlg::stzg_(int stzg){sw_stzg=stzg;}
void CUHRwinDlg::mnzg_(int mnzg){sw_mnzg=mnzg;}
void CUHRwinDlg::skzg_(int skzg){sw_skzg=skzg;}
```

References

Schrausser, D. G. (2023). *Schrausser/UHRwin: Time application for MS Windows (v4.0.0)*. Zenodo. DOI:10.5281/zenodo.7654048