KOD PROGRAM PEMANTAUAN MASA NYATA

C1. Aturcara Program EX92026 bagi Sistem BAF dalam Talian

```
// EX92026Dlg.cpp : implementation file
//

#include "stdafx.h"
#include "EX92026.h"
#include "EX92026Dlg.h"
#include "PCIDAQ.h"
#include ".\ex92026dlg.h"
#include <iostream>
#include <fstream>
using namespace std;
 #include <time.h>
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CEX92026Dlg dialog

CEX92026Dlg::CEX92026Dlg(CWnd* pParent /*=NULL*/)
	: CDialog(CEX92026Dlg::IDD, pParent)
	//, m_EDIT1(0)
	//, m_EDIT2(0)
	//, m_EDIT3(0)
	//, m_EDIT4(0)
	//, m_EDIT5(0)
	, m_ST1(0)
	, m_ST2(0)
	, m_EDIT6(0)
	, m_EDIT7(0)
{
	//{{AFX_DATA_INIT(CEX92026Dlg)
	m_cardno = -1;
	m_adrange = 5;
	m_endchannel = 0;
	m_ST1 = 8;
	m_ST2 = 6.5;
	//}}AFX_DATA_INIT
```

```
	// Note that LoadIcon does not require a subsequent DestroyIcon in Win32
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}
void CEX92026Dlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	//{{AFX_DATA_MAP(CEX92026Dlg)
	DDX_Control(pDX, IDC_List_Int_Data, m_List_Int_Data);
	DDX_Control(pDX, IDC_CARD_NO, m_ctl_cardno);
	DDX_CBIndex(pDX, IDC_CARD_NO, m_cardno);
	DDX_CBIndex(pDX, IDC_ADRANGE, m_adrange);
	DDX_CBIndex(pDX, IDC_ENDCHANNEL, m_endchannel);
	//}}AFX_DATA_MAP
	DDX_Control(pDX, IDC_CWGRAPH1, m_CWGraph1);
	DDX_Control(pDX, IDC_CWGRAPH2, m_CWGraph2);
	DDX_Control(pDX, IDC_CWGRAPH3, m_CWGraph3);
	DDX_Control(pDX, IDC_CWGRAPH4, m_CWGraph4);
	DDX_Control(pDX, IDC_CWGRAPH5, m_CWGraph5);
	DDX_Text(pDX, IDC_STATIC1, m_EDIT1);
	DDX_Text(pDX, IDC_STATIC2, m_EDIT2);
	DDX_Text(pDX, IDC_STATIC3, m_EDIT3);
	DDX_Text(pDX, IDC_STATIC4, m_EDIT4);
	DDX_Text(pDX, IDC_STATIC5, m_EDIT5);
	DDX_Text(pDX, IDC_EDIT1, m_ST1);
	DDX_Text(pDX, IDC_EDIT2, m_ST2);
	DDX_Text(pDX, IDC_STATIC6, m_EDIT6);
	DDX_Text(pDX, IDC_STATIC7, m_EDIT7);
	//DDX_Control(pDX, IDC_LIST1, m_LIST1);
}

BEGIN_MESSAGE_MAP(CEX92026Dlg, CDialog)
	//{{AFX_MSG_MAP(CEX92026Dlg)
	ON_WM_PAINT()
	ON_WM_QUERYDRAGICON()
	ON_BN_CLICKED(IDC_OPEN, OnOpen)
	ON_BN_CLICKED(IDC_CLOSE, OnClose)
	ON_BN_CLICKED(IDC_Soft_AD_Aquire, OnSoftADAquire)
	//}}AFX_MSG_MAP
	ON_WM_TIMER()
	ON_BN_CLICKED(IDC_BUTTON1, OnBnClickedButton1)
	ON_BN_CLICKED(IDC_BUTTON2, OnBnClickedButton2)
	ON_BN_CLICKED(IDC_BUTTON3, OnBnClickedButton3)
	ON_BN_CLICKED(IDC_BUTTON4, OnBnClickedButton4)
	ON_BN_CLICKED(IDC_BUTTON5, OnBnClickedButton5)
	ON_BN_CLICKED(IDC_BUTTON6, OnBnClickedButton6)
	ON_BN_CLICKED(IDC_BUTTON7, OnBnClickedButton7)
```

```
	//ON_LBN_SELCHANGE(IDC_LIST1, OnLbnSelchangeList1)
END_MESSAGE_MAP()

////////////////////////////////////////////////////////////////////////////
// CEX92026Dlg message handlers

BOOL CEX92026Dlg::OnInitDialog()
{
	CDialog::OnInitDialog();

	// Set the icon for this dialog.  The framework does this automatically
	//  when the application's main window is not a dialog
	SetIcon(m_hIcon, TRUE);			// Set big icon
	SetIcon(m_hIcon, FALSE);		// Set small icon

	// TODO: Add extra initialization here
	Form_Load();
	return TRUE;  // return TRUE  unless you set the focus to a control
}

// If you add a minimize button to your dialog, you will need the code below
//  to draw the icon.  For MFC applications using the document/view model,
//  this is automatically done for you by the framework.

void CEX92026Dlg::OnPaint()
{
	if (IsIconic())
	{
		CPaintDC dc(this); // device context for painting

		SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

		// Center icon in client rectangle
		int cxIcon = GetSystemMetrics(SM_CXICON);
		int cyIcon = GetSystemMetrics(SM_CYICON);
		CRect rect;
		GetClientRect(&rect);
		int x = (rect.Width() - cxIcon + 1) / 2;
		int y = (rect.Height() - cyIcon + 1) / 2;

		// Draw the icon
		dc.DrawIcon(x, y, m_hIcon);
	}
	else
	{
		CDialog::OnPaint();
```

```
	}
}
// The system calls this to obtain the cursor to display while the user drags
//  the minimized window.
HCURSOR CEX92026Dlg::OnQueryDragIcon()
{
	return (HCURSOR) m_hIcon;
}
WORD ErrCode=0; //used for returned ErrCode of EX92026 functions
char MsgBuf[50];                                    //used to show Error String
char NsgBuf[50];                                    //used to show Error String
char OsgBuf[50];                                    //used to show Error String
char PsgBuf[50];                                    //used to show Error String
char QsgBuf[50];                                    //used to show Error String
char RsgBuf[50];                                    //used to show Error String
char SsgBuf[50];                                    //used to show Error String
char TsgBuf[50];                                    //used to show Error String
char UsgBuf[50];                                    //used to show Error String
char VsgBuf[50];                                    //used to show Error String
char WsgBuf[50];                                    //used to show Error String
char XsgBuf[50];                                    //used to show Error String
char dateStr [9];                                   //used to show current date
char timeStr [9];                                   //used to show current time

void CEX92026Dlg::Form_Load()                       //FORM LOAD
{
	long ID(4);
	//long status;
	WORD Card_Num;                                  //card no. installed

	ErrCode=W_2026_Open(&Card_Num);                 //Initialize Pcx-2026 card.
	Card_Num=1;

	if(ErrCode==0 && Card_Num>0)                    //if open cards success
	{
		char str[3];
		WORD ID[4];
		W_2026_GetCardsID(ID);
		for(int i=0;i<Card_Num;i++)
		{
			sprintf(str,"%d",ID[i]);
			m_ctl_cardno.AddString(str);        //Fill the list with card number
		}
		m_ctl_cardno.SetCurSel(0);
	}
	else		//if open cards fail
```

```
	{
		sprintf (MsgBuf,"Open Device Failed!\n\rERROR Code= %d",ErrCode);
		AfxMessageBox(MsgBuf);		// show error messagebox
	}
	//write to text file
	ofstream myfile;				//open text file

	myfile.open ("UKM Biological Aerated Filter Control System.txt", ios::out | ios::app);
	myfile <<"\n";
	myfile <<"\n";
	myfile << "DO\t";
	myfile << "ORP\t";
	myfile << "PH\t";
	myfile << "NH4\t";
	myfile << "NO3\t";
	myfile << "TIME\t";
	myfile << "\tDATE\t";
	myfile <<"\n";

}
void CEX92026Dlg::OnOpen()	//SET MIN PH
{
	UpdateData(TRUE);		//update data
	m_EDIT7=m_ST2;			//SET MIN PH
	UpdateData(FALSE);		 //update data
}
void CEX92026Dlg::OnClose()	 // CLOSE ALL CARDS AND WINDOW
{
	//ErrCode = W_2026_Set_Do_Bit(1,0);
	//ErrCode = W_2026_Set_Do_Bit(1,1);
	ErrCode = W_2026_Reset_Do_Bit(1,3);
	W_2026_Close();			//Close all cards
	CDialog::OnOK();		//Close the Window
}
void CEX92026Dlg::OnSoftADAquire() //SET MAX PH
{
	UpdateData(TRUE);		//update data
	m_EDIT6=m_ST1;			//SET MAX PH
	UpdateData(FALSE);		//update data
}
BEGIN_EVENTSINK_MAP(CEX92026Dlg, CDialog)
	ON_EVENT(CEX92026Dlg, IDC_CWBOOLEAN1, 1, ValueChangedCwboolean1,
VTS_BOOL)
END_EVENTSINK_MAP()

void CEX92026Dlg::ValueChangedCwboolean1(BOOL Value)//START AND STOP BUTTON
```

```
{
	ErrCode = W_2026_Set_Do_Bit(1,3);

	if (Value)
	{
		SetTimer(0,15000,0);			//set timer
	}
	else
	{
		KillTimer(0);				//stop timer
		ErrCode = W_2026_Set_Do_Bit(1,0);
		ErrCode = W_2026_Set_Do_Bit(1,1);
	}
}
void CEX92026Dlg::OnTimer(UINT nIDEvent)		//TIMER
{
//--------------------------------------TIMER1-------------------------------------------
	float DissolvedOxygen;
	float ORPValue;
	float pHValue;
	float Ammonium;
	float Nitrate;
	char str[3];
	float avgdata1;
	float avgdata2;
	float avgdata3;
	float avgdata4;
	float avgdata5;
	CString ADDataStr;
	avgdata1=0;
	avgdata2=0;
	avgdata3=0;
	avgdata4=0;
	avgdata5=0;
	float k;
	float p;
	float l;
	float m;
	float a;
	float b;
	float c;
	float d;
	float e;
	float f;
	float g;
	//float DissolvedOxygen2;
```

```
double phmax;
double phmin;

UpdateData(TRUE);
m_ctl_cardno.GetLBText(m_cardno,str);
//W_2026_AD_Acquire_Initial(atoi(str),m_adrange);
W_2026_AD_Acquire_Initial(1,5);
//Single channel AD conversion
for (int l=0;l<50;l++)
{
ErrCode=W_2026_SingleChannel_AD_Acquire(1,2,&DissolvedOxygen);
avgdata1=avgdata1+DissolvedOxygen;
}
avgdata1=avgdata1/50;

for (int s=0;s<30;s++)
{
ErrCode=W_2026_SingleChannel_AD_Acquire(1,1,&ORPValue);
avgdata2=avgdata2+ORPValue;
}
avgdata2=avgdata2/30;

for (int h=0;h<5;h++)
{
ErrCode=W_2026_SingleChannel_AD_Acquire(1,0,&pHValue);
avgdata3=avgdata3+pHValue;
}
avgdata3=avgdata3/5;

for (int a=0;a<30;a++)
{
ErrCode=W_2026_SingleChannel_AD_Acquire(1,5,&Nitrate);
avgdata5=avgdata5+Nitrate;
}
avgdata5=avgdata5/30;

for (int b=0;b<50;b++)
{
ErrCode=W_2026_SingleChannel_AD_Acquire(1,3,&Ammonium);
avgdata4=avgdata4+Ammonium;
}
avgdata4=avgdata4/50;

for (int j=0;j<10;j++)
{
}
```

```
//-----------------------------------------------------DissolvedOxygen
//UpdateData(TRUE);//update data
//sprintf (OsgBuf,"%.2f",(20.3*avgdata1)-0.44);
sprintf (OsgBuf,"%.2f",(10.15*avgdata1)-0.205);
//sprintf (OsgBuf,"%.2f",(10.09*avgdata1)-0.174);
m_EDIT1=OsgBuf;
c=atof(OsgBuf);
UpdateData(FALSE);//update data

//-----------------------------------------------------ORPValue
//UpdateData(TRUE);//update data
sprintf (NsgBuf,"%.f",(159.8*avgdata2)-217.5);
m_EDIT2=NsgBuf;
d=atof(NsgBuf);
UpdateData(FALSE);//update data

//-----------------------------------------------------pHValue
//UpdateData(TRUE);//update data
sprintf (MsgBuf,"%.2f",(2.8*avgdata3)-0.026);
//m_EDIT3=(2.8*avgdata3)-0.026;
m_EDIT3=MsgBuf;
e=atof(MsgBuf);
UpdateData(FALSE);//update data

//-----------------------------------------------------Ammonium
//UpdateData(TRUE);//update data -4.7959*Ammonium
sprintf(PsgBuf,"%.3f",avgdata4);//display voltage
k=atof(PsgBuf);                 //voltage value
p=(((10.94*k)-4.519)-0.04)*2;//EQUATION 50mg/l
//p=((21.91*k)-9.069);          //EQUATION 1000mg/l
sprintf(SsgBuf,"%.2f",p);       //display value
m_EDIT4=SsgBuf;
f=atof(SsgBuf);
UpdateData(FALSE);              //update data

//-----------------------------------------------------Nitrate
//UpdateData(TRUE);//update data -1.698*Nitrate
sprintf (QsgBuf,"%.2f",((20.11*avgdata5)+0.159)*2);
//m_EDIT5=((20.11*avgdata5)+0.159);
//sprintf (QsgBuf,"%.5f",avgdata5);
m_EDIT5=QsgBuf;
g=atof(QsgBuf);
UpdateData(FALSE);//update data

COleVariant vOptional((long)DISP_E_PARAMNOTFOUND, VT_ERROR);
float data;       //DissolvedOxygen
```

```
	float data1;	//ORPValue
	float data2;	//pHValue
	float data3;	//Ammonium
	float data4;	//Nitrate
	COleVariant kSec = 5.0;
	//COleVariant kSec = 0.00001157407;

	for (int i=0;i<1;++i)
	{
		//data = (10.15*avgdata1)-0.205;//
//DissolvedOxygen
		data = c;
		m_CWGraph1.ChartY (COleVariant(data),kSec,vOptional);//DissolvedOxygen
graph

		//data1 = (141.2*avgdata2)-126;
//ORPValue
		data1 = d;
		m_CWGraph2.ChartY (COleVariant(data1),kSec,vOptional);//ORPValue graph

		//data2 = (2.8*avgdata3)-0.026;
//pHValue
		data2 = e;
		m_CWGraph3.ChartY (COleVariant(data2),kSec,vOptional);//pHValue graph

		//data3 = ((10.94*avgdata4)-4.519)-0.1;
//Ammonium
		data3 = f;
		m_CWGraph4.ChartY (COleVariant(data3),kSec,vOptional);//Ammonium graph

		//data4 = (20.11*avgdata5)+0.159;
//Nitrate
		data4 = g;
		m_CWGraph5.ChartY (COleVariant(data4),kSec,vOptional);//Nitrate graph
	}
//=============================================================
	_strdate( dateStr);				//date
	_strtime( timeStr );				//time

	ofstream myfile;				//open file
	myfile.open ("UKM Biological Aerated Filter Control System.txt", ios::out | ios::app);
					//file name
	myfile <<"\n";
   myfile << OsgBuf <<"\t";			//write do
	myfile << NsgBuf <<"\t";			//write orp
	myfile << MsgBuf <<"\t";			//write ph
```

```
	myfile << SsgBuf <<"\t";			//write ammonium
	myfile << QsgBuf <<"\t";			//write nitrate
	myfile << timeStr <<"\t";			//write time
	myfile << dateStr <<"\t";			//write date
	myfile.close();
//--------------------------------------TIMER2-----------------------------------------

	if(c<2.9)
	{
		ErrCode = W_2026_Reset_Do_Bit(1,3);	//air pump ON
	}
	else if (c>3.1)
	{
		ErrCode = W_2026_Set_Do_Bit(1,3);	//air pump OFF
	}
	else
	{

	}
//--------------------------------------TIMER3-----------------------------------------
	/*phmax=m_EDIT6;
	phmin=m_EDIT7;

	if (data2>phmax)//if alkali
	{
		ErrCode = W_2026_Reset_Do_Bit(1,1);
		ErrCode = W_2026_Set_Do_Bit(1,0);
	}
	else if (data2<phmin)//if acid
	{
		ErrCode = W_2026_Reset_Do_Bit(1,0);
		ErrCode = W_2026_Set_Do_Bit(1,1);
	}
	//if (phmax>data2>phmin)//if in the middle
	else
	{
		ErrCode = W_2026_Set_Do_Bit(1,1);
		ErrCode = W_2026_Set_Do_Bit(1,0);
	}*/

	CDialog::OnTimer(nIDEvent);
}
void CEX92026Dlg::OnBnClickedButton1()		//PUMP 1 ON
{
	ErrCode = W_2026_Set_Do_Bit(1,0);
}
```

```
void CEX92026Dlg::OnBnClickedButton2()          //PUMP 1 OFF
{
        ErrCode = W_2026_Reset_Do_Bit(1,0);
}
void CEX92026Dlg::OnBnClickedButton3()          //PUMP 2 ON
{
        ErrCode = W_2026_Set_Do_Bit(1,1);
}
void CEX92026Dlg::OnBnClickedButton4()          //PUMP 2 OFF
{
        ErrCode = W_2026_Reset_Do_Bit(1,1);
}
void CEX92026Dlg::OnBnClickedButton5()          //AIR PUMP ON
{
        ErrCode = W_2026_Reset_Do_Bit(1,3);
}

void CEX92026Dlg::OnBnClickedButton6()          //AIR PUMP OFF
{
        ErrCode = W_2026_Set_Do_Bit(1,3);
}
void CEX92026Dlg::OnBnClickedButton7()
{
}

//void CEX92026Dlg::OnLbnSelchangeList1()
//{
        // TODO: Add your control notification handler code here
//}
```