

EzLIB使用手冊

(Version 4.3)

EzProg



ICP DAS CO., LTD.

泓格科技股份有限公司

Warranty

All products manufactured by ICPDAS Inc. are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICPDAS Inc. assumes no liability for damages consequent to the use of this product. ICPDAS Inc. reserves the right to change this manual at any time without notice. The information furnished by ICPDAS Inc. is believed to be accurate and reliable. However, no responsibility is assumed by ICPDAS Inc. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright 1997-2009 by ICPDAS Inc., LTD. All rights reserved worldwide.

Trademark

The names used for identification only maybe registered trademarks of their respective companies.

License

The user can use, modify and backup this software on a single machine. The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

技術支援

如果您有任何使用泓格產品問題，請用以下電子郵件信箱聯繫：

Service@icpdas.com

目錄

| | |
|--|-----------|
| 1 前言 | 6 |
| 1.1 使用EzLIB程序開發使用說明 | 6 |
| 1.2 目前EzLIB支援函式如下四大類 | 6 |
| 1.3 使用方法 | 11 |
| 2 FORMAT TRANSFORMATION格式轉換 | 12 |
| 2.1 CString和Float資料格式的轉換..... | 12 |
| 2.1.1 將字串CString轉為浮點數值Float | 12 |
| 2.1.2 將浮點數值Float轉為字串CString | 12 |
| 2.2 DWORD和Float資料格式的轉換 | 13 |
| 2.2.1 將DWORD轉為浮點數值Float | 13 |
| 2.2.2 將浮點數值Float轉為DWORD | 13 |
| 2.3 CString和Double資料格式的轉換 | 14 |
| 2.3.1 將字串CString轉為浮點數值Double..... | 14 |
| 2.3.2 將浮點數值Double轉為字串CString..... | 14 |
| 2.4 Double和DWORD資料格式的轉換..... | 15 |
| 2.4.1 將Double值轉為雙DWORD資料格式 | 15 |
| 2.4.2 將雙DWORD資料格式轉為Double值 | 15 |
| 2.5 Double和BYTE資料格式的轉換..... | 16 |
| 2.5.1 將Double值轉為 8 BYTE資料格式..... | 16 |
| 2.5.2 將 8 BYTE資料格式轉為Double值..... | 16 |
| 2.6 使用檔案作Array資料的存取 | 17 |
| 2.6.1 開啓欲讀寫的檔案 | 17 |
| 2.6.2 將陣列寫入檔案 | 17 |
| 2.6.3 將讀回檔案填回陣列..... | 18 |
| 2.6.4 查詢已開啓的檔案大小 | 19 |
| 2.7 呼叫PAC其它應用程式 | 20 |
| 2.8 讀取本地的日期和時間 | 21 |
| 2.8.1 讀取日期/時間..... | 21 |
| 2.8.2 讀取日期..... | 21 |
| 2.8.3 讀取時間..... | 22 |
| 3 CONTEXT DRAWING本文畫圖 | 23 |
| 3.1 BMP的CDC界面 | 24 |
| 3.1.1 開啓BMP檔 | 24 |
| 3.1.2 儲存BMP檔 | 27 |
| 3.1.3 取得BMP的高度(像素)..... | 27 |

| | | |
|----------|-------------------------------------|-----------|
| 3.1.4 | 取得BMP的寬度(像素)..... | 27 |
| 3.1.5 | 詢問BMP是否已成功開啓..... | 28 |
| 3.1.6 | 將記憶體BMP圖檔顯示到CDC指定區域中..... | 28 |
| 3.1.7 | 將記憶體圖檔顯示到CDC指定位置..... | 29 |
| 3.1.8 | 取得記憶體BMP的CDC指標..... | 29 |
| 3.2 | Monitor的CDC界面..... | 30 |
| 3.2.1 | 取得對話框(畫面)的CDC指標..... | 30 |
| 3.2.2 | 畫面的CDC繪圖範例..... | 31 |
| 4 | ETHERNET CONNECTION網路連結..... | 32 |
| 4.1 | FTP網路連結..... | 32 |
| 4.1.1 | 初始化FTP..... | 32 |
| 4.1.2 | 關閉FTP連線..... | 33 |
| 4.1.3 | 改變遠端路徑..... | 33 |
| 4.1.4 | 改變近端路徑..... | 33 |
| 4.1.5 | 傳送檔案到遠端..... | 34 |
| 4.1.6 | 在近端接收檔案..... | 34 |
| 4.2 | TCP/IP網路連結..... | 35 |
| 4.2.1 | 初始化TCP/IP Socket..... | 35 |
| 4.2.2 | 關閉TCP/IP連線..... | 35 |
| 4.2.3 | 傳送封包到遠端..... | 36 |
| 4.2.4 | 在近端自動接收封包..... | 36 |
| 5 | TREND 趨勢圖..... | 38 |
| 5.1 | 使用CTRENDA類別開發使用說明..... | 38 |
| 5.2 | 基本設定功能..... | 39 |
| 5.2.1 | CTRENDA 建構..... | 39 |
| 5.2.2 | 設定放置CTRENDA的CDC指標..... | 41 |
| 5.2.3 | 設定CTRENDA的顯示上下限..... | 42 |
| 5.2.4 | 設定CTRENDA的顯示座標軸屬性..... | 43 |
| 5.2.5 | 設定CTRENDA的曲線顯示屬性..... | 44 |
| 5.2.6 | 設定CTRENDA的圖例顯示屬性..... | 45 |
| 5.3 | 資料與顯示處理..... | 46 |
| 5.3.1 | 新增一筆記錄..... | 46 |
| 5.3.2 | 清除所有記錄..... | 47 |
| 5.3.3 | Trend 繪圖更新作業..... | 47 |
| 5.4 | 變更顯示範圍與狀態..... | 48 |
| 5.4.1 | 趨勢線顯示..... | 48 |
| 5.4.2 | 變更Y軸顯示上下限..... | 48 |

| | |
|--------------------------|----|
| 5.4.3 變更X軸顯示格與顯示點數 | 49 |
| 5.5 範例 | 49 |
| 附錄一 錯誤代碼表 | 52 |

1 前言

1.1 使用EzLIB程序開發使用說明

近年來工業控制器 PAC 的發展逐漸成熟，應用也越來越廣，使用者迅速增加所需功能，且自動化設備需求亦日益蓬勃，為協助新使用者對 PAC 控制器能輕鬆入門，因此 ICPDAS 泓格科技針對使用者常用功能，配合 MFC 開發此套 EzLIB 軟體函式庫，主要設計目標，是為了降低工業自動化系統的開發門檻及縮短開發時程。

EzLIB 主要分為下列四大類：

1. **Format Transformation:** 格式轉換，主要為一些資料轉換的功能函式庫，資料存檔及讀回到變數函式庫。
2. **Context Drawing:** 本文畫圖，讀取 BMP 檔，在其中繪圖，及顯示，存檔等函式庫。
3. **Ethernet Connection :** 網路連結，包含 TCP/IP Client 和 FTP Client 的各項功能。
4. **Trend :** 趨勢圖顯示功能，每一物件包含最多 8 條趨勢線，及提供各項功能。

1.2 目前 EzLIB 支援函式如下四大類

- **Format Transformation**
- **CEzBMP Context Drawing**
- **Ethernet Connection**
- **Trend**

Function List

| Format Transformation |
|---|
| float CString_To_Float(CString CStr) |
| CString 轉 Float |
| CString Float_To_CString(float fData, long DotNo) |
| Float 轉 CString |
| float DWORD_To_Float(DWORD dwData) |
| DWORD 轉 Float |
| DWORD Float_To_DWORD(float fData) |
| Float 轉 DWORD |
| void DOUBLE_To_DWORDS (DWORD dwDATA[2] , double DData) |
| Double 轉 DWORD |
| double DWORDS_To_DOUBLE (DWORD dwDATA[2]) |
| DWORD 轉 Double |
| void DOUBLE_To_BYTES (BYTE BDATA[8] , double DData) |
| Double 轉 8 BYTE |
| double BYTES_To_DOUBLE (BYTE BDATA[8]) |
| 8 BYTE 轉 Double |
| long Start_FileIO(LPCTSTR lpszFileName, bool ReadWrite, BYTE OpenMode) |
| 開啟欲讀寫的檔案 |
| long Array_To_File(LPCVOID lpBuffer, DWORD nBytesToWrite, bool Flag) |
| 將陣列寫入檔案 |
| long File_To_Array(LPVOID lpBuffer, DWORD nBytesToRead, bool Flag) |
| 讀回檔案填回陣列 |
| void Call_Shell(WORD ShellNo) |
| 開啟 PAC 的外部應用程式 |
| void Get_DT(TCHAR tcDT[30], bool Year, bool Month, bool Week, bool Day, bool Hour, bool Minute, bool Second) |
| 讀取日期和時間 |
| void Get_Date(TCHAR tcDate[15]) |
| 讀取日期 |
| void Get_Time(TCHAR tcTime[15]) |
| 讀取時間 |
| EzBMP Context Drawing |

| |
|--|
| bool Open_BMP(LPCTSTR lpszFileName) |
| 開啟 BMP 檔案到記憶體 |
| bool Save_BMP(LPCTSTR lpszFileName) |
| 將記憶體圖檔存成 BMP 檔案 |
| long Get_BMP_Height() |
| 取得記憶體圖檔的高(像素) |
| long Get_BMP_Width() |
| 取得記憶體圖檔的寬(像素) |
| bool Get_BMP_OK() |
| 查詢是否已經載入 BMP 圖檔 |
| bool Draw_BMP_Fit(CDC* pDC, CPoint ptDest, CSize size, BOOL bForceBackground) |
| 將記憶體圖檔顯示到 CDC 指定區域中 |
| bool Draw_BMP(CDC* pDC, CPoint ptDest, BOOL bForceBackground) |
| 將記憶體圖檔顯示到 CDC 中(1:1) |
| CDC* Get_Monitor_DC(CDC* pDC) |
| 取得 Monitor 的 CDC HANDLE |
| CDC* Get_BMP_DC(CDC* pDC) |
| 取得記憶體圖檔的 CDC HANDLE |
| |
| |

| Ethernet Connection | |
|--|--------------------------------------|
| BOOL FtpInitial(LPCTSTR FtpAddress, long FtpPort, LPCTSTR UserName, LPCTSTR Password) | |
| | FTP Client 初始化設定 |
| void FtpClose() | |
| | 關閉 FTP Client |
| BOOL FtpCwd(LPCTSTR Directory) | |
| | Change Working Directory，改變 FTP 工作路徑 |
| BOOL FtpCld(LPCTSTR Directory) | |
| | Change Local Directory，改變 FTP 本地路徑 |
| BOOL FtpPut(LPCTSTR LocalFile, LPCTSTR RemoteFile) | |
| | 傳送檔案從 FTP Client 到 Server |
| BOOL FtpGet(LPCTSTR RemoteFile, LPCTSTR LocalFile) | |
| | 傳送檔案從 FTP Server 到 Client |
| BOOL SocketInitial(short SocketNumber, LPCTSTR IpAddress, long IpPort) | |
| | Ethernet Socket 初始化 |
| BOOL SocketClose(short SocketNumber) | |
| | 關閉 Ethernet Socket |
| BOOL SocketSend(short SocketNumber, LPCTSTR String) | |
| | 傳送資料到 Server 端 |
| BOOL SocketReceiveOn(short SocketNumber) | |
| | 開啟自動接收資料的 CallBack Function |
| BOOL SocketReceiveOff(short SocketNumber) | |
| | 關閉 CallBack Function |

| |
|---|
| Trend |
| Create(LPCRECT Rect,int iLeft,int iRight,int iTop,int iBottom,COLORREF RGB,WORD RecordDataNo,WORD XDir) |
| 將 CTRENDA 物件初始化 |
| void SET_CDC (CDC* pDC) |
| 設定欲放置顯示 CTRENDA 的對話框(Dialog)CDC 指標 |
| void Set_Rang(float fMin_X,float fMin_Y,float fMax_X,float fMax_Y,int iDevN_X,int iDevN_Y) |
| 設定 CTRENDA 的顯示上下限 |
| void SetFrameP(BYTE Target,int nPenStyle, int nWidth,COLORREF crColor) |
| 設定 CTRENDA 的顯示座標軸屬性 |
| void SetCurveP(BYTE Cno,int nPenStyle, int nWidth,COLORREF crColor,bool IS_Used) |
| 設定 CTRENDA 的曲線顯示屬性 |
| void AddRecord(double fData[8], CString Dtime) |
| void AddRecord(double fData0,double fData1,double fData2,double fData3,double fData4,double fData5,double fData6,double fData7 ,CString Dtime) |
| 新增一筆記錄到 Trend 中，本指令提供兩種參數輸入方法，一為陣列式，另一為列舉式，可以依需求使用 |
| void ClearRecord() |
| 將 Trend 記錄的所有記錄清除 |
| void Paint() |
| 將記錄的 Trend 繪製到先前指定的 CDC 中 |
| void SetVisible(BYTE Cno,bool Visible) |
| 利用此函式，設定使用中的趨勢線是否要顯示 |
| void ChangeRangY(double fMin_Y, double fMax_Y) |
| 變更 Y 軸顯示上下限 |
| void ChangeRangX(int iDevN_X,WORD RecordDataNo); |
| 變更 X 軸顯示分割格子數與顯示資料點數，變更 X 軸顯示會清除原先所有記錄，重新顯示資料 |

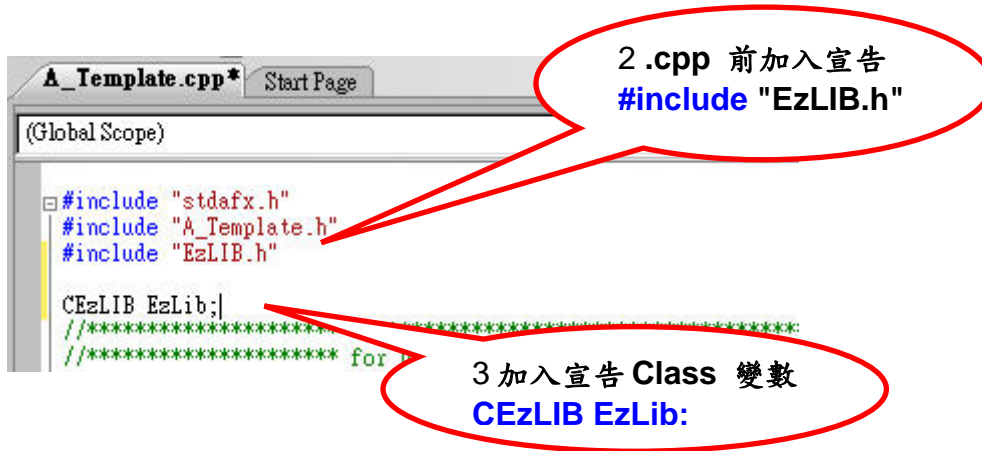
1.3 使用方法

EzLIB 為靜態函式庫，檔案如下：

EzLIB.h: 主要為一些資料轉換的功能函式庫的宣告。

EzLIB.lib & EzLIB_D.lib: 連結時需使用的函式庫。

1. **EzLIB.h** 要在欲引用 **EzLIB Function** 函式的 **.cpp** 前加入宣告：
#include "EzLIB.h"。
2. 加入宣告 **Class** 變數 **CEzLIB EzLib**，如下圖



The screenshot shows a code editor window titled "A_Template.cpp". The code includes the following lines:

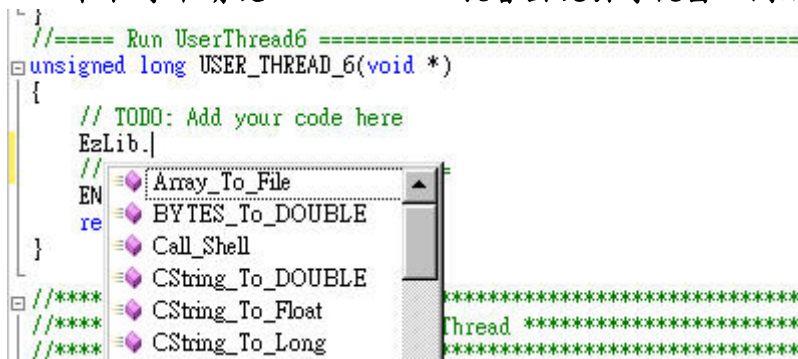
```
#include "stdafx.h"
#include "A_Template.h"
#include "EzLIB.h"

CEzLIB EzLib;
//*****
//***** for n
```

Two red callout boxes highlight specific parts of the code:

- Box 1: "2 .cpp 前加入宣告 #include "EzLIB.h"" points to the `#include "EzLIB.h"` line.
- Box 2: "3 加入宣告 Class 變數 CEzLIB EzLib:" points to the `CEzLIB EzLib;` line.

3. 在程序中請鍵入 "EzLIB." 就會出現引導視窗，例如下圖：



The screenshot shows a code editor window with a function definition:

```
unsigned long USER_THREAD_6(void *)
{
    // TODO: Add your code here
    EzLib.|
    //
    EN
    re
}
//*****
//*****
//*****
```

A dropdown menu is open over the `EzLib.|` line, showing a list of functions with a search icon on the left:

- Array_To_File
- BYTES_To_DOUBLE
- Call_Shell
- CString_To_DOUBLE
- CString_To_Float
- CString_To_Long

The background code is partially obscured by the dropdown menu.

如此可以方便始用者輸入正確函式。

2 Format Transformation 格式轉換

2.1 CString 和 Float 資料格式的轉換

2.1.1 將字串 CString 轉為浮點數值 Float

- **float** `CString_To_Float (CString CStr)`

功能： 將CString轉為float。

參數： **CStr** 欲轉換CString字串

回應： 轉換浮點值

範例：
`DWORD dwAA= 0;`
`float fDATA= 0.55f;`
`float fRET= 0;`
`CString CS;`
`dwAA= EzLIB.Float_To_DWORD(fDATA);`
`fRET= EzLIB.DWORD_To_Float(dwAA);`
`CS= EzLIB.Float_To_CString(fDATA,5);`
`fRET= EzLIB.CString_To_Float(CS);`

2.1.2 將浮點數值 Float 轉為字串 CString

- **CString** `Float_To_CString (float fData, long DotNo)`

功能： 將float轉為CString。

參數： **fData** 欲轉換浮點值
DotNo 指定小數位數(0 為不指定轉換小數位)

回應： 轉換後CString字串

範例： 請參考 2.1.1

2.2 DWORD 和 Float 資料格式的轉換

2.2.1 將 DWORD 轉為浮點數值 Float

- **float** `DWORD_To_Float` (`DWORD dwDATA`)

功能： 將存為DWORD 格式的float轉回為float值。

參數： `dwDATA` 欲轉換DWORD資料

回應： 轉換浮點值

範例： 請參考 2.1.1

2.2.2 將浮點數值 Float 轉為 DWORD

- `DWORD Float_To_DWORD`(`float fData`)

功能： 將float轉為DWORD格式資料。

參數： `fData` 欲轉換浮點值

回應： 轉換後DWORD

範例： 請參考 2.1.1

2.3 CString 和 Double 資料格式的轉換

2.3.1 將字串 CString 轉為浮點數值 Double

- **double** CString_To_DOUBLE(CString CStr)

功能： 將CString轉為double。

參數： **CStr** 欲轉換CString字串

回應： 轉換浮點值

範例：
double dDATA= 0.55;
double dRET= 0;
CString CS;
CS= EzLIB.DOUBLE_To_CString(dDATA,5);
dRET= EzLIB.CString_To_DOUBLE(CS);

2.3.2 將浮點數值 Double 轉為字串 CString

- CString DOUBLE_To_CString (**double** DData, **long** DotNo)

功能： 將double轉為CString。

參數： **DData** 欲轉換浮點值
DotNo 指定小數位數(0 為不指定轉換小數位)

回應： 轉換後CString字串

範例： 請參考 2.3.1

2.4 Double 和 DWORD 資料格式的轉換

2.4.1 將 Double 值轉為雙 DWORD 資料格式

- **void** DOUBLE_To_DWORDS (DWORD *dwDATA*[2], **double** *DData*)

功能： 將**double**轉存為兩個DWORD 資料。

數參： *dwDATA*[2] 欲轉換出DWORD資料變數(回傳值)
DData 欲轉換 **double** 值

回應： 無

範例：

```
double Da=25245245.23525;  
double DaRET=0;  
DWORD DArray[2]={0,0};  
EzLIB.DOUBLE_To_DWORDS(DArray,Da);  
DaRET=EzLIB.DWORDS_To_DOUBLE(DArray);
```

```
BYTE BArray[8]={0,0,0,0,0,0,0,0};  
DaRET=0;  
EzLIB.DOUBLE_To_BYTES(BArray,Da);  
DaRET=EzLIB.BYTES_To_DOUBLE(BArray);
```

2.4.2 將雙 DWORD 資料格式轉為 Double 值

- **double** DWORDS_To_DOUBLE (DWORD *dwDATA*[2])

功能： 將雙DWORD資料格式轉回為**double**值。

參數： *dwDATA*[2] 欲轉換DWORD格式資料

回應： 轉換後**double** 值

範例： 請參考 2.4.1

2.5 Double 和 BYTE 資料格式的轉換

2.5.1 將 Double 值轉為 8 BYTE 資料格式

- **void** DOUBLE_To_BYTES (BYTE *BDATA*[8], **double** *DData*)

功能： 將**double**轉為 8 BYTE格式 。

參數： *BDATA*[8] 欲轉換出BYTE資料變數(回傳值)
DData 欲轉換 **double** 值

回應： 無

範例： 請參考 2.4.1

2.5.2 將 8 BYTE 資料格式轉為 Double 值

- **double** BYTES_To_DOUBLE (BYTE *BDATA*[8])

功能： 將 8 Byte 料格式轉為**double**值 。

參數： *BDATA*[8] 欲轉換BYTE格式資料

回應： 轉換後**double** 值

範例： 請參考 2.4.1

2.6 使用檔案作 Array 資料的存取

2.6.1 開啟欲讀寫的檔案

- **long** Start_FileIO(LPCTSTR *lpszFileName*, **bool** *ReadWrite*, **BYTE** *OpenMode*)

功能： 開啟欲讀寫的檔案。

參數：
lpszFileName 欲開啟檔案名稱
ReadWrite 檔案的讀寫模式(true=寫入,false=讀出)
OpenMode 檔案開啟模式
1:開啟新檔
2:開啟檔案(如存在開啟舊檔,如不存在開啟新檔)
3:開啟舊檔(檔案必須存在)

回應： **0** 執行成功
其他 執行失敗(請參考錯誤碼)

範例： 請參考 2.6.3

2.6.2 將陣列寫入檔案

- **long** Array_TO_File(LPCVOID *lpBuffer*, **DWORD** *nBytesToWrite*, **bool** *Flag*)

功能： 將陣列資料存到檔案。

參數：
lpBuffer 欲輸出的陣列
nBytesToWrite 欲輸出陣列的 BYTE 數(容量)
Flag 是否要同時關閉檔案(寫入及關閉檔案)
true=寫入及關閉檔案
false=寫入但不關閉檔案(可以繼續寫入同一檔案)

回應： **0** 執行成功
其他 執行失敗(請參考錯誤碼)

範例： 請參考 2.6.3

2.6.3 將讀回檔案填回陣列

● **long** File_TO_Array (LPVOID *lpBuffer*, DWORD *nBytesToRead*, **bool** *Flag*)

功能： 讀回檔案中陣列資料。

參數：
lpBuffer 欲讀入的陣列
nBytesToRead 欲讀入陣列的 BYTE 數(容量)
Flag 是否要同時關閉檔案(讀入及關閉檔案)
true=讀入及關閉檔案
false=讀入但不關閉檔案(可以繼續讀入同一檔案)

回應： 0 執行成功
其他 執行失敗(請參考錯誤碼)

範例：
//1 變數宣告:(在.cpp檔頭)
typedef struct **//== define =====**
{
 BYTE TYPE;
 long PARA;
 long BB;
 long WORK;
 BYTE ARRAY[100];
}DEMO_Data;
DEMO_Data DArray[100];
TCHAR PATH[100]= TEXT("\\ArrayFile.dat");

//2 陣列給值:
void OnFillArrayData()
{
 // TODO: Add your control notification handler code here
 int i,j;
 for (j=0; j<100; j++)
 {
 for (i=0; i<100; i++)
 DArray[j].ARRAY[i]= i;
 }
}

```

//3 將陣列寫入檔案:
void OnWriteArrayToFile()
{
    // TODO: Add your control notification handler code here
    if (EzLIB.Start_FileIO(PATH,true,1)==0)//Open New file for write
    {
        EzLIB.Array_To_File(&DArray,sizeof(DArray),true);
    }
}

//4 將陣列從檔案讀回:
void CRW_ArrayFileDialog::OnReadArrayFromFile()
{
    // TODO: Add your control notification handler code here
    if (EzLIB.Start_FileIO(PATH,false,3)==0)//Open old file for read
    {
        EzLIB.File_To_Array(&DArray,sizeof(DArray),true);
    }
}

```

2.6.4 查詢已開啟的檔案大小

● **DWORD Get_File_Length(DWORD *HighDWORD)**

功能： 讀回檔案大小。

參數： *HighDWORD* 檔案大小的高雙字元

回應： **DWORD** 檔案大小的低雙字元

範例： 無

2.7 呼叫 PAC 其它應用程式

void Call_Shell(WORD ShellNo)

功能： 開啟PAC的其它應用程式。

參數： *ShellNo* 欲開啟PAC的其它應用程式號碼

| 參數 | 程式號碼 | 說明 |
|---------|------|---------|
| ShellNo | 0 | - |
| | 1 | - |
| | 2 | 鍵盤內容 |
| | 3 | 密碼內容 |
| | 4 | 密碼內容 |
| | 5 | - |
| | 6 | 系統內容 |
| | 7 | 顯示器內容 |
| | 8 | 滑鼠內容 |
| | 9 | - |
| | 10 | - |
| | 11 | 輸入窗格內容 |
| | 12 | 移除程式 |
| | 13 | 日期/時間內容 |
| | 14 | 憑證 |
| | 15 | - |
| | 16 | - |

回應： 無

範例： *//開啟PAC的日期/時間內容*
CEzLIB EzLIB;
EzLIB.Call_Shell(13);

2.8 讀取本地的日期和時間

2.8.1 讀取日期/時間

**void Get_DT(TCHAR tcDT[30],
 bool Year, bool Month, bool Week, bool Day,
 bool Hour, bool Minute, bool Second, bool mSecond)**

功能： 讀取日期和時間。

| | | |
|-----|-----------------|------------------|
| 參數： | <i>tcDT[30]</i> | 存放日期和時間的TCHAR陣列 |
| | <i>Year</i> | 年(true=讀取) |
| | <i>Month</i> | 月(true=讀取) |
| | <i>Week</i> | 週(true=讀取) |
| | | 0: 週日、1: 週一..... |
| | <i>Day</i> | 日(true=讀取) |
| | <i>Hour</i> | 時(true=讀取) |
| | <i>Minute</i> | 分(true=讀取) |
| | <i>Second</i> | 秒(true=讀取) |
| | <i>mSecond</i> | 微秒(true=讀取) |

回應： 無

範例： 請參考 2.8.3

2.8.2 讀取日期

void Get_Date(TCHAR tcDate[15])

功能： 讀取日期(年/月/日)。

參數： *tcDate[15]* 存放日期的TCHAR陣列

回應： 無

範例： 請參考 2.8.3

2.8.3 讀取時間

void Get_Time(TCHAR tcTime[15])

功能： 讀取時間(時:分:秒)。

參數： *tcTime[15]* 存放時間的TCHAR陣列

回應： 無

範例： **CEzLIB EzLIB;**
TCHAR tcTime[15];
EzLIB.Get_Time(tcTime);

TCHAR tcDate[15];
EzLIB.Get_Date(tcDate);

TCHAR tcDT[30];
EzLIB.Get_DT(tcDT, true, true, true, true, true, true, true, true);

3 Context Drawing 本文畫圖

本章節提供之函式，主要利用 MFC CDC 相關函式 Function 作為圖型處理，因此提供取得 BMP File 與 Monitor 及 Printer 的 CDC 界面，讓使用者可以方便使用，本章節範例取自 CDC_DEMO，並共用 CDC WorkCDC 變數繪圖，詳細請參考 CDC_DEMO 應用範例。

```
CEzBMP m_BMP; // 宣告一個 BMP 類別變數 =====

CDC WorkCDC;

// 主要繪圖程序 本章節範例都會 CALL 這個程序=====
void Draw_Work()
{
    //===== Draw Lines =====
    WorkCDC.MoveTo(0,0);
    WorkCDC.LineTo(300,200);
    WorkCDC.MoveTo(300,0);
    WorkCDC.LineTo(0,200);
    //===== Select font =====
    CFont font;
    VERIFY(font.CreatePointFont(240, _T("Arial"), &WorkCDC));
    CFont* def_font = WorkCDC.SelectObject(&font);
    font.DeleteObject();

    //===== Set TEXT Color=====
    WorkCDC.SetTextColor(RGB(0, 0, 0));
    //===== Draw TEXT =====
    WorkCDC.DrawText("泓格科技 繪圖測試", CRect(50, 300, 1200, 400),
0);

    WorkCDC.SelectObject(def_font);
}
```

3.1 BMP 的 CDC 界面

需先加入宣告 CEzBMP Class 變數=> CEzBMP m_BMP;

3.1.1 開啟 BMP 檔

- **bool** Open_BMP(LPCTSTR *lpszFileName*)

功能： 開啟一BMP檔案到記憶體。

參數： *lpszFileName* 欲開啟的檔案名稱

回應： **0** 執行成功
其他 執行失敗(請參考錯誤碼)

範例：

//在開啟 BMP 檔 =====

```
void CCDC_DEMODlg::OnGetBMPCDC()
{
    // TODO: Add your control notification handler code here
    //== Select file and load file =====
    CString strFilter = "Bitmap Files (*.bmp)|*.bmp|All Files (*.*)|*.*|";
    CFileDialog dlg(TRUE, NULL, NULL,
    OFN_HIDEREADONLY|OFN_EXPLORER, strFilter, NULL);
    if (dlg.DoModal() == IDOK)
    {
        if (m_BMP.Open_BMP(dlg.GetPathName()))
        {
            BMP_OK=true;
            Printer_OK=false;
            Monitor_OK=false;
            this->SetWindowText(dlg.GetPathName());
        }
        else
        {
            BMP_OK=false;
            AfxMessageBox(_T("Unable to open bitmap file"));
        }
    }
    //== GET the BMP CDC =====
    WorkCDC.Attach(*m_BMP.Get_BMP_DC(this->GetDC()));
}
```



```

//在 BMP 檔繪圖 =====
void CCDC_DEMODlg::OnBMPDraw()
{
    // TODO: Add your control notification handler code here
    if (WorkCDC)
    {
        Draw_Work();
    }
}

//將 BMP 另存新檔 =====
void CCDC_DEMODlg::OnBMPSave()
{
    // TODO: Add your control notification handler code here
    //== Select a file =====
    CString strFilter = "Bitmap Files (*.bmp)|*.bmp|All Files (*.*)|*.*|";
    CFileDialog dlg(false, NULL, NULL,
    OFN_HIDEREADONLY|OFN_EXPLORER, strFilter, NULL);
    if (dlg.DoModal() == IDOK)
    {
        CString m_FileName= dlg.GetPathName();
        if (!m_FileName.IsEmpty())
            //== Save BMP to file =====
            m_BMP.Save_BMP(m_FileName);
    }
}

```

```

//將 BMP 另存新檔 =====
void CCDC_DEMODlg::OnDrawBMP()
{
    // TODO: Add your control notification handler code here
    m_BMP.Draw_BMP(this->GetDC(), CPoint(0,0),0);
}

//將 BMP 顯示在螢幕的位置 =====
void CCDC_DEMODlg::OnDrawBMP()
{
    // TODO: Add your control notification handler code here
    m_BMP.Draw_BMP(this->GetDC(), CPoint(0,0),0);
}

//將 BMP 縮放顯示在螢幕的固定區域內 =====
void CCDC_DEMODlg::OnDrawBMPFit()
{
    // TODO: Add your control notification handler code here
    CRect rect(0, 0, 800, 550);
    m_BMP.Draw_BMP_Fit(this->GetDC(), CPoint(0,0), rect.Size(),0);
}

```

3.1.2 儲存 BMP 檔

- **bool** Save_BMP(LPCTSTR *lpzFileName*)

功能： 儲存記憶體BMP到檔案中。

參數： *lpzFileName* 欲儲存的檔案名稱

回應： 0 執行成功
其他 執行失敗(請參考錯誤碼)

範例： 請參考 3.1.1 章節

3.1.3 取得 BMP 的高度(像素)

- **long** Get_BMP_Height()

功能： 取得記憶體圖檔的高(像素)。

參數： 無

回應： BMP檔高(像素)

範例： 請參考 3.1.1 章節

3.1.4 取得 BMP 的寬度(像素)

- **long** Get_BMP_Width()

功能： 取得記憶體圖檔的寬(像素)。

參數： 無

回應： BMP檔寬(像素)

範例： 請參考 3.1.1 章節

3.1.5 詢問 BMP 是否已成功開啟

- **bool** Get_BMP_OK()

功能： 詢問BMP檔案是否已成功開啟。

參數： 無

回應： **true**: 已經開啟成功
false: 尚未開啟成功

範例： 請參考 3.1.1 章節

3.1.6 將記憶體 BMP 圖檔顯示到 CDC 指定區域中

- **bool** Draw_BMP_Fit(CDC* *pDC*, CPoint *ptDest*, CSize *size*,
BOOL *bForceBackground*)

功能： 將記憶體BMP圖檔顯示到 CDC 指定區域中，並會依區域的大小自動調整顯示比例。

參數： *pDC* 欲顯示CDC的指標
ptDest 欲顯示 BMP 的開始點座標
size 欲顯示 CDC 的大小
bForceBackground 是否強制設為背景

回應： **0** 執行成功
其他 執行失敗(請參考錯誤碼)

範例： 請參考 3.1.1 章節

3.1.7 將記憶體圖檔顯示到 CDC 指定位置

- **bool Draw_BMP(CDC* pDC, CPoint ptDest, BOOL bForceBackground)**

功能： 將記憶體圖檔顯示到 CDC 指定位置，並會已 1:1 比例顯示。

參數： *pDC* 欲顯示CDC的指標
ptDest 欲顯示 BMP 的開始點座標
bForceBackground 是否強制設為背景

回應： 0 執行成功
 其他 執行失敗(請參考錯誤碼)

範例： 請參考 3.1.1 章節

3.1.8 取得記憶體 BMP 的 CDC 指標

- **CDC* Get_BMP_DC(CDC* pDC)**

功能： 取得記憶體圖檔的 CDC 指標，使用者可以利用CDC 的function直接在記憶體圖檔上做繪圖及文字顯示等工作。

參數： *pDC* CDC的指標請用this->GetDC()

回應： BMP CDC的指標

範例： 請參考 3.1.1 章節

3.2.2 畫面的 CDC 繪圖範例

```
//在畫面上繪圖(=====
void CCDC_DEMODlg::OnMonitorDraw()
{
    // TODO: Add your control notification handler code here
    if (WorkCDC && Monitor_OK)
        Invalidate();//呼叫 MFC 螢幕更新
}
```

```
// MFC 螢幕更新動作,在畫面上繪圖=====
void CCDC_DEMODlg::OnPaint()
{
    CPaintDC dc(this); // device context for painting
    // TODO: Add your message handler code here
    if (WorkCDC && Monitor_OK)
    {
        Draw_Work();//呼叫 主要繪圖程序
    }
    // Do not call CDialog::OnPaint() for painting messages
}
```

4 Ethernet Connection 網路連結

本章節提供之函式，主要包含TCP/IP Client和FTP Client的各項功能，連線到遠端的FTP Server並登錄連結，或建立一個TCP/IP Client Socket連結遠端的TCP/IP Server。收送封包

4.1 FTP 網路連結

4.1.1 初始化 FTP

- **long** FtpInitial(LPCTSTR *FtpAddress*, **long** *FtpPort*, LPCTSTR *UserName*, LPCTSTR *Password*)

功能： 連線到遠端的FTP Server。並登錄連結，

參數： *FtpAddress*: FTP Server的 IP 位址
FtpPort: FTP Server的 Port：0 ~ 65535
UserName: 登錄伺服端的使用者名稱
Password: 使用者密碼名稱

回應： 0 執行成功
其他 執行失敗(請參考錯誤碼)

範例： CEzLIB EzLIB;
// =====
void CEzLIB_TESTDlg::OnFTPTest()
{
 // TODO: Add your control notification handler code here
 BOOL rec;
 rec= EzLIB.FtpInitial
 (_T("10.0.0.110"), 21, _T("Anonymous"), _T("0"));
 rec= EzLIB.FtpCwd(_T("\\Temp"));
 rec= EzLIB.FtpCld(_T("\\Temp"));
 rec= EzLIB.FtpPut(_T("test.BMP"), _T("test.BMP"));
 rec= EzLIB.FtpGet(_T("test.BMP"), _T("test.BMP"));
 EzLIB.FtpClose();
}

4.1.2 關閉 FTP 連線

- **void** FtpClose()

功能： 關閉與FTP Server 的連線

參數：

回應：

範例： 請參考 4.1.1

4.1.3 改變遠端路徑

long FtpCwd(LPCTSTR *Directory*)

功能： 改變遠端FTP Server 的工作路徑

參數： *Directory*: 名稱路徑

回應： 0 執行成功
其他 執行失敗(請參考錯誤碼)

範例： 請參考 4.1.1

4.1.4 改變近端路徑

long FtpCld(LPCTSTR *Directory*)

功能： 改變近端(FtpClient) 的工作路徑

參數： *Directory*: 名稱路徑

回應： 0 執行成功
其他 執行失敗(請參考錯誤碼)

範例： 請參考 4.1.1

4.1.5 傳送檔案到遠端

long FtpPut(LPCTSTR *LocalFile*, LPCTSTR *RemoteFile*)

功能： 傳送檔案到遠端的FTP Server。

參數： *LocalFile*: FTP Client傳送的檔案名稱
RemoteFile: FTP Server接收的檔案名稱

回應： 0 執行成功
其他 執行失敗(請參考錯誤碼)

範例： 請參考 4.1.1

4.1.6 在近端接收檔案

long FtpGet(LPCTSTR *RemoteFile*, LPCTSTR *LocalFile*)

功能： 從遠端的FTP Server。接收檔案，

參數： *RemoteFile*: FTP Server傳送的檔案名稱
LocalFile: FTP Client的檔案名稱

回應： 0 執行成功
其他 執行失敗(請參考錯誤碼)

範例： 請參考 4.1.1

4.2 TCP/IP 網路連結

4.2.1 初始化 TCP/IP Socket

- **long SocketInitial**(**short** *SocketNumber*, **LPCTSTR** *IpAddress*,
long *IpPort*)

功能： 建立一個TCP/IP Client Socket連結遠端的，TCP/IP Server。收送封包

參數： **SocketNumber:** **Socket**：編號0 ~ 15
IpAddress: **TCPIP Server IP**位址
IpPort: **TCPIP Server Port**：0 ~ 65535

回應： **0** 執行成功
 其他 執行失敗(請參考錯誤碼)

範例： 請參考 4.2.4

4.2.2 關閉 TCP/IP 連線

- **long SocketClose**(**short** *SocketNumber*)

功能： 關閉TCP/IP Client Socket。

參數： **SocketNumber:** **Socket**：編號0 ~ 15

回應： **0** 執行成功
 其他 執行失敗(請參考錯誤碼)

範例： 請參考 4.2.4

4.2.3 傳送封包到遠端

- **long SocketSend(short SocketNumber, LPCTSTR String)**

功能： 傳送封包到遠端的TCPIP Server。

參數： **SocketNumber:** **Socket**：編號0 ~ 15
String: 傳送的字串

回應： **0** 執行成功
 其他 執行失敗(請參考錯誤碼)

範例： 請參考 4.2.4

4.2.4 在近端自動接收封包

- **long SocketReceiveOn(short SocketNumber, ptTSRFunc2 SRF)**

功能： 啟動CallBack Function自動從遠端的，TCPIP Server，接收封包一次最大為 2048 位元組 如果封包很長時，會多次引發CallBack Function，應用時請多注意。

參數： **SocketNumber:** **Socket**：編號0 ~ 15
SRF: 傳送的字串

回應： **0** 執行成功
 其他 執行失敗(請參考錯誤碼)

範例：

CEzLIB EzLIB;

//User define callback function

```
void SocketReceive0(CString AA,BYTE * Data,long Lenth)
{
    BYTE Re[5000]={0};
    memcpy( Re, Data,Lenth );
}
```

```

// =====
void OnEthernetTest()
{
    // TODO: Add your control notification handler code here
    BOOL rec;
    rec= EzLIB.SocketInitial(0, _T("10.0.0.20"), 60000);
    rec= EzLIB.SocketSend(0, _T("WinCon_IP_10.0.0.26"));
    rec= EzLIB.SocketReceiveOn
        (0, &(ptTSRFunc2)SocketReceive0);
}

void OnEthernetClose()
{
    // TODO: Add your control notification handler code here
    BOOL rec;
    rec= EzLIB.SocketReceiveOff(0);
    rec= EzLIB.SocketClose(0);
}

```

- **long** SocketReceiveOff(**short** SocketNumber)

功能： 關閉CallBack Function，結束。自動接收封包

參數： **SocketNumber:** **Socket**：編號0 ~ 15

回應： **0** 執行成功
 其他 執行失敗(請參考錯誤碼)

範例： 請參考如上

5.2 基本設定功能

5.2.1 CTRENDA 建構

- **Create**(LPCRECT *Rect*,int *iLeft*,int *iRight*,int *iTop*,int *iBottom*,COLORREF *RGB*,WORD *RecordDataNo*,WORD *XDir*);

功能： 將CTRENDA 物件初始化。

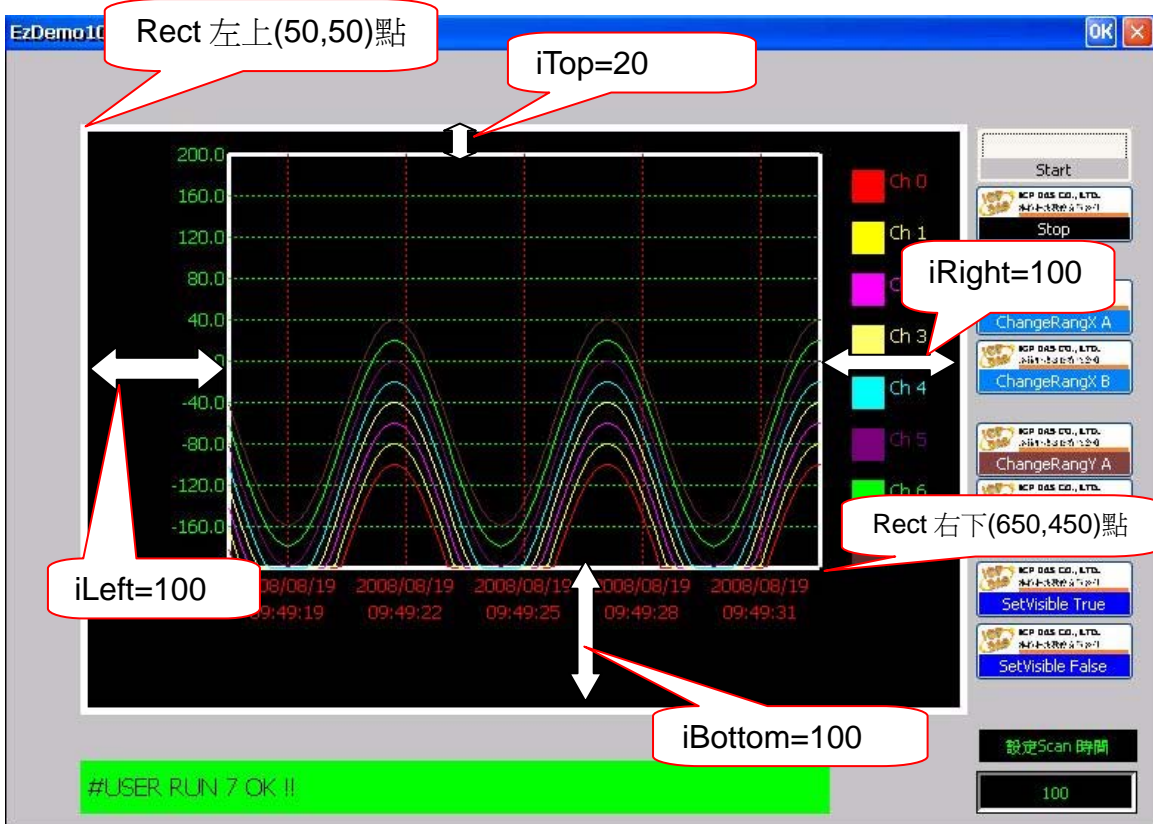
參數：

| | |
|------------------------------------|--|
| <i>Rect</i> | 欲建立Trend在目標對話框的顯示位置。 |
| <i>iLeft</i> | 指定 Trend 左邊的預留標示空間(畫素) (1~200)。 |
| <i>iRight</i> | 指定 Trend 右邊的預留標示空間(畫素) (1~200)。 |
| <i>iTop</i> | 指定 Trend 上緣的預留標示空間(畫素) (1~200)。 |
| <i>iBottom</i> | 指定 Trend 下緣的預留標示空間(畫素) (1~200)。 |
| <i>RGB</i> | 指定 Trend 的背景顏色。 |
| <i>iRecordDataNo</i> (10~1024)。 | 指定 Trend 一頁面的 X 軸顯示資料點數 |
| <i>[XDir]</i> | 非必要參數, 指定Trend顯示繪圖方向。 XDir_Right_Lift =0: , 繪圖方向由右到左(預設值) XDir_Lift_Right =1: , 繪圖方向由左到右 |

回應： 無

範例:

```
CRect rect(50, 50, 650, 450); //宣告在目標對話框的顯示位置  
MyTrend.Create(rect, 100, 100, 20, 100, RGB(0,0,0), 100, XDir_Right_Lift);  
; //宣告 TTrend 在目標對話框的顯示位置
```



5.2.2 設定放置 CTRENDA 的 CDC 指標

● **void SET_CDC (CDC* pDC)**

功能： 設定欲放置顯示CTRENDA的對話框(Dialog)CDC指標。

參數： *pDC* CDC的指標請用this->GetDC()

回應： 無

範例： `MyTrend.SET_CDC(this->GetDC());`

5.2.3 設定 CTRENDA 的顯示上下限

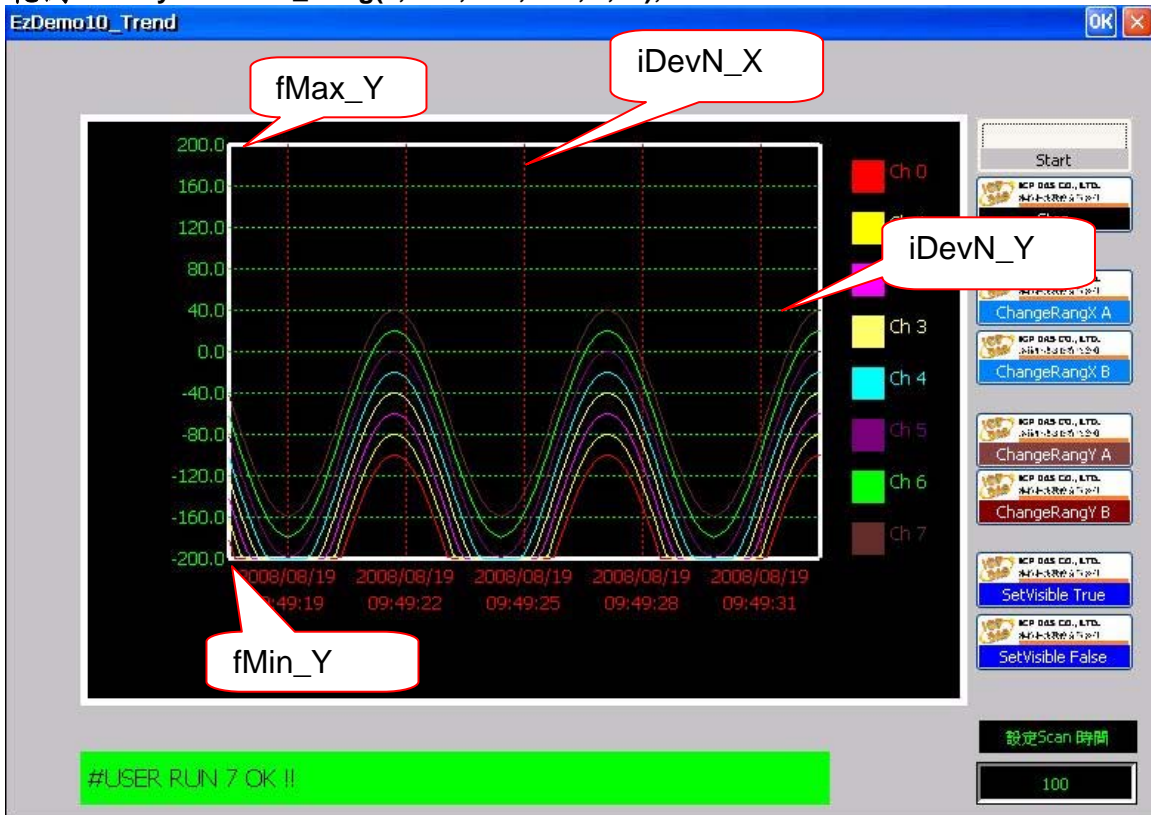
- `void Set_Rang(float fMin_X,float fMin_Y,float fMax_X,float fMax_Y,int iDevN_X,int iDevN_Y);`

功能： 設定CTRENDA的顯示上下限。

| | | |
|-----|----------------------|---------------|
| 參數： | <code>fMin_X</code> | X軸的最小值(參考用) |
| | <code>fMax_X</code> | X軸的最大值(參考用) |
| | <code>fMin_Y</code> | Y軸的最小值 |
| | <code>fMax_Y</code> | Y軸的最大值 |
| | <code>iDevN_X</code> | X軸的分割格數(3~20) |
| | <code>iDevN_Y</code> | Y軸的分割格數(3~20) |

回應： 無

範例： `MyTrend.Set_Rang(0,-200, 100, 200, 5,10);`



5.2.4 設定 CTRENDA 的顯示座標軸屬性

- `void SetFrameP(BYTE Target,int nPenStyle, int nWidth,COLORREF crColor);`

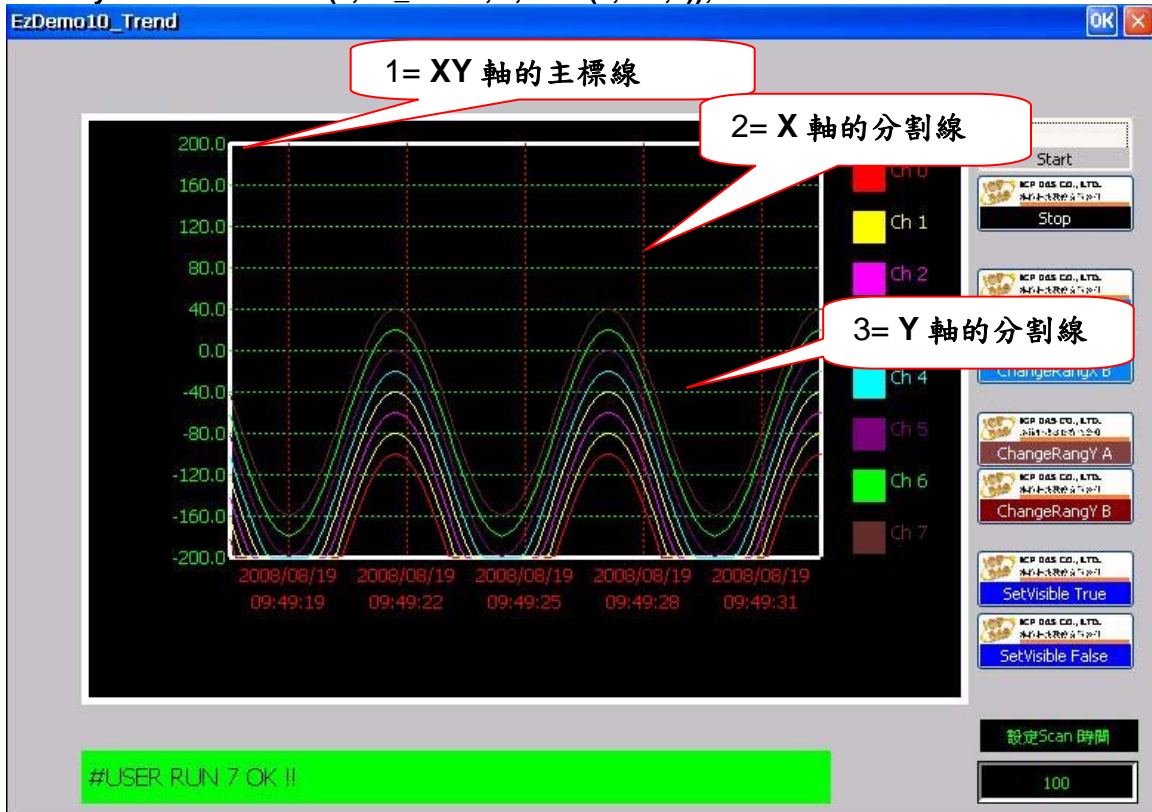
功能： 設定CTRENDA的顯示座標軸屬性。

| | | |
|-----|------------------|--|
| 參數： | Target | 設定對象 1:XY 軸的主標線 2:X 軸的分割線 3:Y 軸的分割線 |
| | nPenStyle | 畫筆型式 PS_SOLID:實心線 PS_DASH:虛線 |
| | nWidth | 畫筆線寬(1~10) |
| | crColor | 指定線的顏色 |

回應： 無

範例：

```
MyTrend.SetFrameP(1,PS_SOLID, 3,RGB(255,255,255));  
MyTrend.SetFrameP(2,PS_DASH, 1, RGB(255,0,0));  
MyTrend.SetFrameP(3,PS_DASH, 1, RGB(0,255,0));
```



5.2.5 設定 CTRENDA 的曲線顯示屬性

- `void SetCurveP(BYTE Cno,int nPenStyle, int nWidth,COLORREF crColor,bool IS_Used);`

功能： 設定CTRENDA的曲線顯示屬性。

| | | |
|-----|------------------|------------------------------------|
| 參數： | Cno | 設定曲線對象(0~7) |
| | nPenStyle | 畫筆型式 PS_SOLID:實心線 PS_DASH:虛線 |
| | nWidth | 畫筆線寬(1~5) |
| | crColor | 指定線的顏色 |
| | IS_Used | 指定是否要使用(顯示) |

回應： 無

範例：

```
MyTrend.SetCurveP(0,PS_SOLID,1,RGB ( 255 , 0 , 0 ),true);  
MyTrend.SetCurveP(1,PS_SOLID,1,RGB ( 255 , 255 , 0 ),true);  
MyTrend.SetCurveP(2,PS_SOLID,1,RGB ( 255 , 0 , 255 ),true);  
MyTrend.SetCurveP(3,PS_SOLID,1,RGB ( 255 , 255 , 111 ),true);  
MyTrend.SetCurveP(4,PS_SOLID,1,RGB ( 0 , 255 , 255 ),true);  
MyTrend.SetCurveP(5,PS_SOLID,1,RGB ( 127 , 0 , 127 ),true);  
MyTrend.SetCurveP(6,PS_SOLID,1,RGB ( 0 , 255 , 0 ),true);  
MyTrend.SetCurveP(7,PS_SOLID,1,RGB ( 100 , 44 , 44 ),true);
```



5.2.6 設定 CTRENDA 的圖例顯示屬性

● **void SetLegend(bool ShowFlag,WORD XSpace);**

功能： 設定CTRENDA的圖例顯示屬性。

參數： **ShowFlag** 設定是否顯示圖例
 true: 顯示圖例
 false: 不顯示圖例
XSpace 設定顯示圖例與 Trend 主體間距
 1~200

回應： 無

範例： 無

5.3 資料與顯示處理

5.3.1 新增一筆記錄

- `void AddRecord(double fData[8], CString Dtime);`
- `void AddRecord(double fData0,double fData1,double fData2,double fData3,double fData4,double fData5,double fData6,double fData7 ,CString Dtime);`

功能： 新增一筆記錄到Trend中，本指令提供兩種參數輸入方法，一為陣列式，另一為列舉式，可以依需求使用。

參數： `fData[8]` 新增一筆記錄的陣列資料。
`fData0~7` 新增一筆記錄的列舉資料。
`Dtime` 新增一筆記錄 X 的時間資料。

回應： 無

範例：

```
CString DateTime;  
//A: 陣列式  
int i;  
double fData[8];  
for(i=0;i<8;i++)  
    fData[i]=High*sin(Data_count*DtoGrade*10)+25*i-150;  
if (Data_count<360)  
    Data_count++;  
else  
    Data_count=1;  
EzLIB.Get_Date(DateTime);  
EzLIB.Get_Time(tcTime);  
wcscat(DateTime,CRLF);  
wcscat(DateTime,tcTime);  
MyTrend.AddRecord(fData,DateTime);
```

```
//B: 列舉式配合 EzProg-I GET_F()  
MyTrend.AddRecord(GET_F(0), GET_F(1), GET_F(2), GET_F(3), GET_F(4),  
GET_F(5), GET_F(6), GET_F(7) ,DateTime);
```

```
//C: 列舉式配合 EzProg-I IN_AI()  
MyTrend.AddRecord(IN_AI(0), IN_AI(1), IN_AI (2), IN_AI (3), IN_AI (4), IN_AI (5),  
IN_AI (6), IN_AI (7) ,DateTime);
```

```
MyTrend.Paint();
```

5.3.2 清除所有記錄

- **void ClearRecord();**

功能： 將Trend記錄的所有記錄清除。

參數： 無

回應： 無

範例：

```
MyTrend. ClearRecord ();
```

5.3.3Trend 繪圖更新作業

在對話框(Dialog) OnPaint 中需呼叫這一函式，圖形資料更新到畫面到先前指定的對話框(Dialog)CDC 中。

- **void Paint();**

功能： 將記錄的Trend繪製到先前指定的CDC中。

參數： 無

回應： 無

範例：

```
MyTrend.Paint();
```

5.4 變更顯示範圍與狀態

5.4.1 趨勢線顯示

- **void SetVisible(BYTE Cno, bool Visible);**

功能： 利用此函式，設定使用中的趨勢線是否要顯示。

參數： **Cno** 設定曲線對象(0~7)
Visible 指定是(true)否(false)要使用(顯示) 趨勢線

回應： 無

範例：

```
MyTrend.SetVisible(3,true);  
MyTrend.SetVisible(4,true);  
MyTrend.SetVisible(5,true);
```

5.4.2 變更 Y 軸顯示上下限

- **void ChangeRangY(double fMin_Y, double fMax_Y);**

功能： 變更Y軸顯示上下限。

參數： **fMin_Y** Y軸的最小值
fMax_Y Y軸的最大值

回應： 無

範例： **MyTrend.ChangeRangY(-400,400);**
或
MyTrend.ChangeRangY(-200,200);

5.4.3 變更 X 軸顯示格與顯示點數

void ChangeRangX(int iDevN_X,WORD RecordDataNo);

●

功能： 變更X軸顯示分割格子數與顯示資料點數，變更X軸顯示會清除原先所有記錄，重新顯示資料。

參數：

| | |
|------------------------------------|-------------------------|
| <i>iDevN_X</i> | X 軸的分割格數(3~20) |
| <i>iRecordDataNo</i> (10~1024)。 | 指定 Trend 一頁面的 X 軸顯示資料點數 |

回應： 無

範例： MyTrend.ChangeRangX(3,200);
或
MyTrend.ChangeRangX(5,100);

5.5 範例

本範例將展示如何使用 CTRENDA 從宣告到建立與填入資料到顯示資料到螢幕的正常程序。

Example:

```
// STEP 1: Declare the following three global objects:
CTRENDA MyTrend;
CEzLIB EzLIB;
CDC WorkCDC;

void CTrendViewDlg::InitializeTrendView()
{
// STEP 2:
// Declare CRect object
// Initilize the object with the position and size of the graph window
CRect rect(0, 0, 400, 350);

// STEP 3:
// Set the general properties of the graph area like size and color
MyTrend.Creat(rect,50,50,50,50,RGB(0,0,0),100,XDir_Right_Lift);

// STEP 4:
//Attach the trend graph and the global device context to the dialog
// window
```

```

WorkCDC.Attach(*this->GetDC());
MyTrend.SET_CDC(this->GetDC());

// STEP 5:
//Set the line type, width and color of the frame and the grid lines
MyTrend.SetFrameP(1,PS_SOLID, 3,RGB(255,255,255)); //white
MyTrend.SetFrameP(2,PS_DASH, 1, RGB(255,0,0));// red
MyTrend.SetFrameP(3,PS_DASH, 1, RGB(0,255,0));// green

// STEP 6:
// Set the lower and upper limit of the vertical axis. Determine the
// number of gridlines for the vertical and horizontal axis. A label
// is automatically attached to each gridline.
MyTrend.Set_Range(0.0,-60.0, 0.0, 60.0, 3,10);

// STEP 7:
//Set the line type, width and color of the trend lines
MyTrend.SetCurveP(0,PS_SOLID,2,RGB ( 255 , 0 , 0 ),true);
MyTrend.SetCurveP(1,PS_SOLID,2,RGB ( 255 , 255 , 0 ),true);
MyTrend.SetCurveP(2,PS_SOLID,2,RGB ( 255 , 0 , 255 ),true);
MyTrend.SetCurveP(3,PS_SOLID,1,RGB ( 255 , 255 , 111 ),false);
MyTrend.SetCurveP(4,PS_SOLID,1,RGB ( 0 , 255 , 255 ), false);
MyTrend.SetCurveP(5,PS_SOLID,1,RGB ( 127 , 0 , 127 ), false);
MyTrend.SetCurveP(6,PS_SOLID,1,RGB ( 0 , 255 , 0 ), false);
MyTrend.SetCurveP(7,PS_SOLID,1,RGB ( 100 , 44 , 44 ), false);

// STEP 8:
// Draw the frame
MyTrend.Paint();
}

void CTrendViewDlg::DrawTrendView()
{
    CString DateTime;
    wchar_t tcDate[15];
    wchar_t tcTime[15];
    int i;
    static double x = 0;
    double fData[8];

    for(i=0; i<100; i++)
    {
        // STEP 8:
        // Load the data for the y axis (vertical axis)
        fData[0]= 5.0 * x - 60.0 ;
        fData[1]= 2.5 * x - 60.0 ;
        fData[2]= x - 60.0 ;

        if (x <20.0) x++;
        else x=0.0;

        // STEP 9:

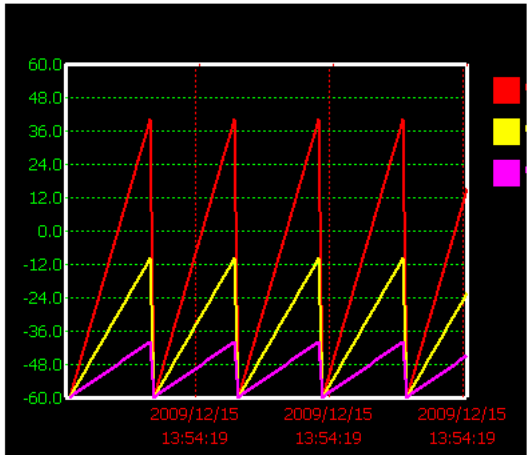
```

```
// Load data for the vertical gridline labels.
// Read the current date and time:
EzLIB.Get_Date(tcDate);
EzLIB.Get_Time(tcTime);
// Copy the date and time to the time CString
DateTime.Format(L"%s\r\n%s", tcDate,tcTime);

// STEP 10:
// Update the graph with the new data and labels
MyTrend.AddRecord(fData,DateTime);

}

// STEP 11:
// Draw the graph
MyTrend.Paint();
}
```



附錄一 錯誤代碼表

| | |
|---------------------------------------|------|
| #define _NO_ERROR | 0 |
| #define _EXEC_ERROR | -1 |
| #define _OPENFILE_ERROR | -2 |
| #define _SETUP_ERROR | -3 |
| #define _FRAM_INIT_ERROR | -4 |
| #define _REGISTER_ERROR | -5 |
| #define _NONE_MAOS_ERROR | -6 |
| | |
| #define _INCORECT_RUN_MODE | -7 |
| | |
| #define _DIVIDE_ZERO_ERROR | -10 |
| | |
| #define _SET_Interrupt_ERROR | -20 |
| | |
| #define I8048_ERROR_NO_MODULE | -31 |
| #define I8048_ERROR_OPEN_DEVICE | -32 |
| #define I8048_ERROR_INVALID_PARAMETER | -33 |
| #define I8048_ERROR_INVALID_HANDLE | -34 |
| #define I8048_ERROR_CALL_IOCTL | -35 |
| #define I8048_ERROR_GET_IST_EVENT | -36 |
| | |
| #define _IN_USE_ERROR | -50 |
| #define _NO_USE_ERROR | -51 |
| #define _OUT_OF_RANGE_ERROR | -52 |
| | |
| #define _AES_NOT_SETKEY_ERROR | -80 |
| #define _AES_CHECK_ERROR | -81 |
| | |
| #define _CREATE_THREAD_ERROR | -90 |
| | |
| #define _INOUT_ERROR | -100 |
| | |
| #define _STP_PARAMETER_ERROR | -150 |

```

#define _SYSTEM_VERSION_ERROR -200
#define _DEVICE_CHECK_ERROR -201
#define _DEVICE_NOT_INIT -202
#define _DEVICE_NOT_WinCon -203

#define _AES_REGCODE_LENTH_ERROR -250
#define _AES_REGMSG_NO_ERROR -251

#define _FILE_NOT_OPEN -300
// the file not open Please open first
#define _READ_FILE_ERROR -310
// Read from file error
#define _WRITE_FILE_ERROR -320
// Write to file error

#define _ETHERNET_CONNECTION_ERROR 1

#define _ALREADY_RUN_WARNING 10

//for i8094H/A
#define _I8094H_TIMEOUT_ERROR 50

//for i8092/F i8094/F i8094H/A
#define _MOTION_NO_REG_ERROR 51

//for i8094/F
#define _MOTION_OPENCONFIG_ERROR 52

//for i8092/F i8094/F i8094H/A
#define _NON_MOTION_ERROR 53
#define _MOTION_INTP_ALREADY_RUN 54

#define _SYSTEM_NOT_READY 100

```