# 按钮

按钮使能：添加类变量m\_btn\_open，

m\_btn\_open.EnableWindow(0); //按钮不可用

m\_btn\_open.EnableWindow(1); //按钮可用

设置按钮文本：

GetDlgItem(IDC\_BUTTON1)->SetWindowTextW(\_T("xxx"));

# Combo BOX

选择某一列：

CComboBox\* combobox;

combobox = (CComboBox\*)GetDlgItem(IDC\_COMBO1);

combobox->SetCurSel(0);//第0列

# 数据转换

1. string to CString

  CString.format("%s",string.c\_str());

1. CString to string

string str(CString.GetBuffer(str.GetLength()));

1. string to char \*

char \*p=string.c\_str();

1. char \* to string

string str(char\*);

1. CString to char \*

strcpy(char,CString,sizeof(char));

1. char \* to CString

CString.format(\_T("%s"),char\*);

 CString的format方法是非常好用的。string的c\_str()也是非常常用的，但要注意和char \*转换时，要把char定义成为const char\*，这样是最安全的。

以上函数UNICODE编码也没问题：unicode下照用，加个\_T()宏就行了,像这样子\_T("%s")

1. CString to int

int num1;

CString str1;

num1 = \_ttoi(str1);

CString to int（可转十六进制）

CString str;

char temp[50];

::wsprintfA(temp, "%ls", (LPCTSTR)str);

w\_Data = (uint32\_t)strtoul(temp,NULL,16); //16为16进制

1. int to CString

int num1=1;

CString str1;

str1.Format(\_T("%d"),num1);

str1.Format(\_T("%.1f"), i); //保留1位小数

# 定时器

类向导->消息->WM\_TIEMR 添加。



启动定时器

SetTimer(0, 500, NULL); // 设置定时器周期500MS

停止定时器

kelltimer(0);

# 5.文件操作

if (-1 == GetFileAttributes(\_T("./Convert\_Data.bin"))) //如果文件不存在 ./为当前目录

GetFileAttributes返回-1则代表文件不存在。

feof(FILE \*p)。

站在光标所在位置，向后看看还有没有字符。如果有，返回0；如果没有，返回非0。

fopen()

字符 含义

"r" 打开文字文件只读

"w" 创建文字文件只写

"a" 增补, 如果文件不存在则创建一个

"r+" 打开一个文字文件读/写

"w+" 创建一个文字文件读/写

"a+" 打开或创建一个文件增补

"b" 二进制文件(可以和上面每一项合用)

"t" 文这文件(默认项)

　文件使用方式 　　　　　　　意 义

“rt”　　　　　　只读打开一个文本文件，只允许读数据

“wt”　　　　　　只写打开或建立一个文本文件，只允许写数据

“at”　　　　　　追加打开一个文本文件，并在文件末尾写数据

“rb”　　　　　　只读打开一个二进制文件，只允许读数据

“wb”　　　　 　 只写打开或建立一个二进制文件，只允许写数据

“ab” 　　　　 　追加打开一个二进制文件，并在文件末尾写数据

“rt+”　　　　　 读写打开一个文本文件，允许读和写

“wt+”　　　　　 读写打开或建立一个文本文件，允许读写

“at+”　　　　　 读写打开一个文本文件，允许读，或在文件末追加数 据

“rb+”　　　　　 读写打开一个二进制文件，允许读和写

“wb+”　　　　　 读写打开或建立一个二进制文件，允许读和写

“ab+” 　　　　　读写打开一个二进制文件，允许读，或在文件末追加数据

对于文件使用方式有以下几点说明：

1. 文件使用方式由r,w,a,t,b，+六个字符拼成，各字符的含义是：

r(read): 读

w(write): 写

a(append): 追加

t(text): 文本文件，可省略不写

b(banary): 二进制文件

+: 读和写

# 6.ASCII码转十六进制

ASCII码 = HEX + 0X30;

HEX码 = ASCII - 0x30;

# 函数内调用控件

函数定义：static UINT receive\_thread( LPVOID pParam );

调用:

UINT Chex\_convert\_binDlg::recv\_thread(LPVOID pParam) //串口接收数据线程

{

Chex\_convert\_binDlg \*test\_dlg = (Chex\_convert\_binDlg \*)pParam;

::SetWindowText(::GetDlgItem(test\_dlg->m\_hWnd, IDC\_STATIC1), \_T("开始更新数据"));

}

# 通用数据格式转换

swscanf\_s 将CString转为其他格式

swscanf\_s((LPCTSTR)CString,\_T(“%f”),&float);

可连续转换

UINT R,G,B;

CString szColor(\_T("#FE00FA"));

swscanf\_s(szColor,\_T("#%02x%02x%02x"),&R,&G,&B);

将CString转为char\* 数组

CString strDate;

char cNowTime[400];

swscanf\_s((LPCTSTR)strDate,\_T("%10c"),cNowTime);

# 获取窗口指针

本地直接使用this代表当前窗口指针，如果在其他窗口调用 p->Debug\_Instance.m\_hWnd 获取，m\_hWnd代表指针窗口指针。

# 读写ini配置文件

利用CFileFind类。

WritePrivateProfileStringW 写配置文件

将lpString(CString型)变量存入lpFileName(Cfg.ini)文件里面，按照lpAppName字段进行分类索引。

BOOL WritePrivateProfileString(

       LPCTSTRlpAppName,

       LPCTSTRlpKeyName,

       LPCTSTRlpString,

       LPCTSTRlpFileName

       );

//其中各参数的意义:

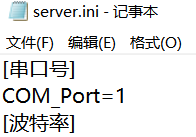
LPCTSTR lpAppName; //是INI文件中的一个字段名.

LPCTSTR lpKeyName;//是lpAppName下的一个键名,通俗讲就是变量名.

LPCTSTR lpString; //是键值, 也就是变量的值,不过必须为LPCTSTR型或CString型的.

LPCTSTR lpFileName;//是完整的INI文件名.

WritePrivateProfileStringW(\_T("串口号"),\_T("COM\_Port"),\_T("1"),File\_Path); // 域名 变量名 数据 路径



GetPrivateProfileInt 读配置文件

参数类型及说明：

int a= GetPrivateProfileInt("b", "c", 1, "c:\\d.ini");

定义变量a 查找c盘d.ini配置文件下的b小段目录下的，c参数，如果找不到使用默认值1

CString File\_Path = \_T("..\\x.ini"); //文件

CFileFind finder;

BOOL ifFind = finder.FindFile(File\_Path);

WritePrivateProfileStringW(\_T("串口号"),\_T("COM\_Port"),\_T("1"),File\_Path); //

# Check-Box控件

int state =((CButton \*)GetDlgItem(IDC\_CHECK1))->GetCheck();

返回值说明：

　　if state == 0，按钮处于未选中状态。

　　if state == 1，按钮处于选中状态。

　　if state == 2，按钮状态不确定。

可使用下方函数进行状态设置：

((CButton \*)GetDlgItem(IDC\_CHECK1))->SetCheck(int nCheck);

参数 nCheck 指定按钮控件的选中状态。可能的取值如下：

　　if nCheck == 0，把按钮设为未选中状态。

　　if nCheck == 1，把按钮设为选中状态 。

if nCheck == 2，把按钮设为不确定状态。仅当按钮的风格是BS\_3STATE或BS\_AUTO3STATE时有效

# CListCtrl控件

CListCtrl\* pList=GetListCtrl();

pList->DeleteAllItems(); // 全部清空

pList->DeleteColumn(i); // 清空第i行

# 动态创建控件

## 13.1 获取窗口坐标

GetWindowRect得到的是相对于当前界面的整个窗口左上角的坐标，比如一个对话框程序，那就是相对于对话框左上角的坐标，从左往右是X方向，从上往下是Y方向。

注意：这个函数的相对坐标原点分两种情况：

1 窗口还没有初始化完成时：原点是整个窗口的左上角

2 初始化完成后，原点是设备屏幕左上角

GetClientRect得到的是相对于窗口客户区左上角的坐标。

ClientToScreen是把坐标从当前窗体转化成全屏幕的。

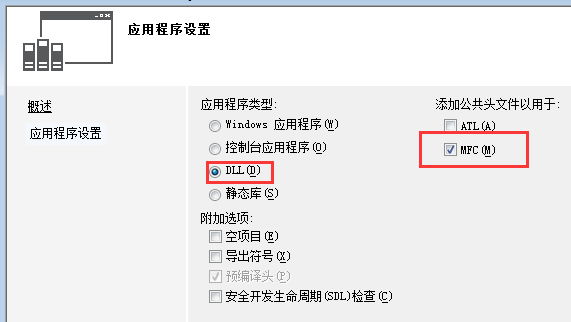
ScreenToClient是把屏幕坐标转化成相对当前窗体的坐标。

|  |
| --- |
| CRect m\_rect\_curve;  GetDlgItem(IDC\_STATIC\_oldshow)->GetWindowRect(m\_rect\_curve); //获取界面上的图片控件位置  ScreenToClient(&m\_rect\_curve);//屏幕坐标转换为其父窗口即对话框的客户区坐标  CRect m\_rect\_sourceData=CRect(CPoint(m\_rect\_curve.left,m\_rect\_curve.top+25),  CSize(m\_rect\_curve.Width() - 5, m\_rect\_curve.Height()-27)); |

# 创建MFC动态库

## 14.1 通用动态库

创建新项目，Win32->Win32项目。



## 14.2 MFC专用动态库

创建新项目，MFC->MFC DLL ，然后选择MFC DLL扩展库。

# 相对路径设置

..\:表示工程文件路径所在目录，..\\表示工程文件路径所在目录再向上一层目录；

指.vcxproj文件所在目录。

# 错误

1. 提示控件未定义标识符 "IDC\_COMBO\_addr" 。

添加头文件#include "resource.h"

1. 静态文本赋值错误

ID不能用下划线，应该IDC\_STATIC1。

1. 串口发送失败：下面2个变量需要设置为全局变量。

COMSTAT Comstat;

OVERLAPPED overlapped;

1. “CWnd::KillTimer”: 非静态成员函数的非法调用

函数声明的时候加static

static UINT recv\_thread(LPVOID pParam)；

调用时候定时窗口变量来调用

Chex\_convert\_binDlg test\_dlg;

test\_dlg.KillTimer(0);

1. VS2013打开串口关闭了串口无法再次打开。

关闭串口需要调用CloseHandle(hCom);关闭，而不是FindClose(hCom);// 关闭串口的句柄

1. CString.GetLength返回值为0

代码加上s.ReleaseBuffer();后再s.GetLength()就OK了。

1. CString转16进制提示不存在从cstring到const char的适当转换函数

char temp[50];

CString st = L"20";

::wsprintfA(temp, "%ls", (LPCTSTR)st);

int n = (unsigned int)strtoul(temp, NULL, 16);//strtol