扩展函数库列表

完整支持 Lua 语言 Ver5.3.3

系统扩展函数库有 sys,window,app,usb,keyboard,mouse,event

程序入口是 _main 函数

技术支持 support@ethese.com

SYS库函数列表

sys.Beep
sys.Clear
sys.DrawViewText
sys.DebugPrint
sys.DebugClear
sys.MsgBox
sys.Print
sys.RunFlag
sys.SaveScreen
sys.Sleep
sys.SendMessage;

声明:sys.Sleep(毫秒数)

说明:阻塞主线程延时函数。

参数:毫秒数
返回值:无。

例子: sys.Sleep(1000) 延时一秒;

声明:sys.DebugPrint(参数)

说明:在输出对话框打印信息。

参数:需要打印的数组，字符串，数组等，以逗号分隔.第一个参数为字符串的时候且格式为{,}可以为控制输出字符串
″ 返回值:无。

例子1: sys.DebugPrint(″Hello word\\r\\n″)
例子2: sys.DebugPrint(″{,}″,1,2,3,4,5)
例子3: sys.DebugPrint(″{;}″,1,2,3,4,5);

声明:sys.Print(参数)

说明:在输出对话框打印信息。

参数:需要打印的数组，字符串，数组等，以逗号分隔.第一个参数为字符串的时候且格式为{,}可以为控制输出字符串
返回值:无。

例子1: sys.Print(″Hello word\\r\\n″)
例子2: sys.Print(″{,}″,1,2,3,4,5)
例子3: sys.Print(″{;}″,1,2,3,4,5);

声明:sys.DebugClear()

说明:清除输出对话框的所有信息。

参数:无
返回值:无。

例子: sys.DebugClear();

声明:sys.Clear()

说明:清除输出对话框的所有信息。

参数:无
返回值:无。

例子: sys.Clear();

声明:返回值 = sys.MsgBox(文字,标题栏文字,类型)

说明:显示一个消息对话框并阻塞线程。

参数:

文字:待显示的消息字符串。
标题栏文字:消息对话框的标题栏文字，如果设置NULL则显示"错误"。
类型:组合类型参数。

返回值:根据类型不同而不同。

例子:

```
sys.MsgBox("Box string","title string",MB_OK)";
```

声明:返回值 = sys.MessageBox(文字,标题栏文字,类型)

说明:显示一个消息对话框并阻塞线程。

参数:

文字:待显示的消息字符串。
标题栏文字:消息对话框的标题栏文字，如果设置NULL则显示"错误"。
类型:组合类型参数。

返回值:根据类型不同而不同。

例子:

```
sys.MessageBox("Box string","title string",MB_OK)";
```

声明:返回值 = sys.SendMessage(句柄,命令,高位参数,低位参数)

说明:给指定句柄发送消息。

参数:

句柄:一个窗体的句柄,如果为NULL，则发消息给桌面。

命令:待发送的命令。

高位参数:参数。

低位参数:参数。

返回值:根据消息不同而不同。

例子:

```
sys.SendMessage(0,WM_LBUTTONDOWN,0,0)
Send left mouse button click down.";
```

声明:返回值 = sys.Beep(频率,时长)

说明:让电脑哔一声。

参数:

频率:哔的频率，37~32767。

时长:哔的时长，毫秒为单位。

返回值:成功则非0值。

例子:

sys.Beep(500,1000);

声明:sys.DrawViewText(视图序号,文字,颜色)

说明:模拟敲击键盘一个字。

参数:

视图序号:子视图序号，0左上角，1右上角，2左下角，3右下角。

文字:待显示的文字。

颜色:24bit颜色值，可用RGB(r,g,b)定义颜色值。

返回值:无

例子:

sys.DrawViewText(0,"FAIL",RGB(255,0,0))"

在左上角视图显示红色FAIL;

声明1:Flag标志 = sys.RunFlag(无参数)

声明2:sys.RunFlag(Flag标志)

声明3:sys.RunFlag(Flag标志,按钮使能状态)

声明4:sys.RunFlag(Flag标志,按钮使能状态,按钮文本)

说明:获取工具栏运行控制按钮状态。

声明1:无参数。
声明2:需要写入的Flag标志状态。
声明3:
Flag标志:需要写入的Flag标志状态。
按钮使能状态:按钮状态。
声明4:
Flag标志:需要写入的Flag标志状态。
按钮使能状态:按钮状态。
按钮文本:显示在按钮的文本。

返回值:bool类型工具栏运行控制按钮状态。声明2，3，4无参数返回。

例子:
f = sys.RunFlag()
返回工具栏运行控制按钮状态。;

声明:sys.SaveScreen(filename)

说明:截屏函数并保存到指定文件名（可以包含路径）。

参数:filename
返回值:无。

例子: sys.SaveScreen("test.bmp") ;

keyboard库函数

keyboard.Press
keyboard.Text

声明:keyboard.Press(键盘码,风格码)

说明:模拟敲击键盘一个字。

参数:

键盘码:虚拟按键码，必须在1~254之间。

风格码:控制不同的敲击风格

KEYEVENTF_EXTENDEDKEY = 如果设置，扫描码将设置为固定的0xE0 (224)。

KEYEVENTF_KEYUP = 如果设置，模拟抬起.如果未设置，将模拟抬起动作。

KEYEVENTF_KEYPRESSED = 如果设置，模拟按钮按下并不抬起。

返回值:无

例子:

keyboard.Press(VK_F4,KEYEVENTF_KEYUP)"
模拟敲击键盘F4。;

声明:keyboard.Text(字符串,延时)

说明:模拟敲击键盘输入文字。

参数:

字符串:待模拟敲击的英文字符串

延时:每敲击一个字符的延时

返回值:无

例子:

```
keyboard.Text("Hello",10)
```

模拟敲击键盘输入文字Hello。;

Mouse库函数

mouse.Click

mouse.Move

mouse.WindowClick

声明:mouse.Click()

说明:模拟鼠标点击。

参数:无
返回值:无

例子:
mouse.Click()
模拟鼠标在当前位置点击。;

声明:mouse.Move(x偏移量,y偏移量)

说明:模拟鼠标移动。

参数:
x偏移量:待鼠标移动的x距离。
y偏移量:待鼠标移动的y距离。

返回值:无

例子:

mouse.Move(-10,10)

模拟鼠标移动(-10,10)左下。;

声明:mouse.WindowClick(句柄)

说明:模拟鼠标按键点击 句柄 的窗体。

参数:

句柄:待鼠标按下的窗体的句柄。

返回值:无

例子:

mouse.WindowClick(hwnd)

模拟鼠标按一下hwnd窗体。

Window库函数

window.DrawText

window.EnumControl

window.FillOpenDlg

window.FillOpenDlgCancel

window.WaitingPreFillOpenDlg
window.isPreFillOpenDlgThreadStillWorking
window.GetDesktopHwnd
window.FindText
window.FitView
window.GetText
window.GetPixel
window.Visible
window.Enable
window.Size
window.SetText
window.SetForeground
window.SetBackground
window.Show
window.SetView;

声明:执行成功标志位,句柄 = window.FindText(父窗口句柄,兄弟窗口句柄,类名,窗体文字)

说明:按文字查找窗体的句柄。

参数:

父窗口句柄:待查找窗体的父窗体句柄，如果为NULL则从桌面查起。

兄弟窗口句柄:待查找窗体的兄弟窗体句柄，如果要查找的在同一级窗体，该参数设置为同一级窗体的上一个句柄。

类名:要查找的类别，按钮为Button，文本框为Edit。
窗体文字:指定的窗体的文字。

返回值:
执行成功标志位:执行成功返回 1，反之返回 0。
句柄:执行成功返回句柄，反之返回nil。

例子:
```
iswindow,hwnd = window.FindText(hwnd,NULL,"Button","OK")
```
按文字查找窗体的句柄查找以hwnd为父句柄的窗体内的OK按钮，执行结果保存在fresult和hwnd。;

声明:执行成功标志位,数据表 = window.EnumControl(待查找的窗口句柄,类名,窗体文字)

说明:按文字查找窗体的句柄。

参数:
待查找的窗口句柄:待查找窗体的句柄，不可以为NULL。
类名:待查找窗体的类名，支持正则表达式匹配。
窗体文字:待查找窗体的文字，支持正则表达式匹配。

返回值:
执行成功标志位:执行成功返回 1，反之返回 0。
数据表:执行成功返回数据表，反之返回nil。
数据表Hwnd :返回查找到的句柄数组。

数据表Sub　　:返回查找到的句柄所在的层数。
数据表Class　　:返回查找到的类名数组。
数据表TextLength:返回查找到的控件的文本长度的数组集合。
数据表Text　　:返回查找到的控件的文本的数组集合。

例子:

```
finded,tab = window.EnumControl(hwnd,"^.{4}$","") -- 所有四个字符类名的控件，正则表达式"^.{4}$\
```

查找hwnd窗口的所有四个字符类名的控件且查找任何文本的控件,数据集合在tab。
数组大小在　　#tab.Hwnd，方便后面的循环数组下标是从1开始，具体看下面代码示例。
句柄数组在　　tab.Hwnd
层数数组在　　tab.Sub
类名数组在　　tab.Class
文本长度数组在 tab.TextLength
文本数组在　　tab.Text

查找本应用程序所有控件的例子:

```
f,x = window.EnumControl(MAIN_HWND,"","") --枚举窗体的所有控件
sys.Print("当前窗体查到了",#x.Hwnd,"个控件\\r\\n")
for i=1,#x.Hwnd do --lua数组下标从1开始
substr = (string.len(x.Text[i])>20) and "..." or " \
sys.Print("\\r\\n",
            string.rep(" \\\\_",x.Sub[i]), --格式化输出
            "HWND = ",string.format("%08X",x.Hwnd[i]),  --显示句柄
            ";\\tClass Name = ",string.format("%s",x.Class[i]),  --显示类名
            ";\\tTextLength = ",string.format("%d",x.TextLength[i]), --显示控件文本长度
            ";\\tText = ",string.sub(x.Text[i],0,20)..  substr)  --显示控件文本缩略，只显示20个字符以内
```

```
end;
```

声明:执行成功标志位,文本 = window.SetText(句柄,字符串)

说明:设置窗口的文本。

参数:

句柄:待设置文本的窗口的句柄。

字符串:待设置的文本。

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

```
fresult = window.SetText(hwnd,"Hello Word!")
```

设置hwnd窗口的文字，执行结果保存在fresult。;

声明:执行成功标志位,文本 = window.GetText(句柄)

说明:获取窗口的文本。

参数:

句柄:待获取文本的窗口的句柄。

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

```
fresult,str = window.GetText(hwnd)
```

获取hwnd窗口的文字，执行结果保存在fresult,获取的文本保存在str。执行失败str值为nil;

声明:执行成功标志位 = window.SetForeground(句柄)

说明:设置窗口显示在最顶层。

参数:

句柄:待更改状态的句柄。

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

```
fresult = window.SetForeground(hwnd)
```

设置hwnd窗口显示状态为最顶层显示，执行结果保存在fresult;

声明:执行成功标志位 = window.SetBackground(句柄)

说明:设置窗口显示在最底层。

参数:

句柄:待更改状态的句柄。

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

fresult = window.SetBackground(hwnd)

设置hwnd窗口显示状态为最底层显示，执行结果保存在fresult;

声明:执行成功标志位 = window.Show(句柄,命令)

说明:设置窗口显示状态。

参数:

句柄:待更改显示状态的句柄。

命令:待显示的命令，显示用SW_SHOW，一般以SW_开头，可以用|进行组合。

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

fresult = window.Show(hwnd,SW_MAXIMIZE)

设置窗口显示状态为最大化显示，执行结果保存在fresult;

声明:线程是否还在运行标志 = window.isPreFillOpenDlgThreadStillWorking()

说明:查询线程是否还在运行。

参数:无

返回值:线程是否还在运行标志，还在运行返回1，没有运行返回0

例子:

fresult = window.isPreFillOpenDlgThreadStillWorking()

查询线程是否还在运行，并将执行结果返回到result;

声明:window.WaitingPreFillOpenDlg()

说明:等待运行中的打开对话框线程。

参数:无

返回值:无

例子:

window.WaitingPreFillOpenDlg()

等待运行中的打开对话框线程。;

声明: window.FillOpenDlgCancel()

说明:终止打开文件对话框线程。

参数:无

返回值:无

例子1:

window.FillOpenDlgCancel()

终止打开文件对话框的线程，并不触发回掉事件。;

声明1:线程创建执行成功标志位 = window.FillOpenDlg(标题栏文字,文件名,按下打开按钮标志位,等待时间)

声明2:线程创建执行成功标志位 = window.FillOpenDlg(标题栏文字,文件名,按下打开按钮标志位,等待时间,回调函数)

说明:等待打开文件对话框，并填写文件路径并关闭对话框。

参数:

标题栏文字:打开文件对话框的标题栏文字，默认为空字符串。

文件名:待填写的文件路径字符串。

按下打开按钮标志位:填完后是否按下打开按钮标志。

等待时间:等待打开文件对话框显示的时间长度，超时返回失败。

回调函数:函数线程执行结果所执行的函数。如果该参数为空，则会删除掉之前FillOpenDlg的所有回调函数。

返回值:线程创建成功标志位，创建成功返回1，失败返回0

例子1:

```
fresult = window.FillOpenDlg("","d:\text.txt",1,5000)
```

执行上一个动作后，等待5000毫秒打开文件对话框，如5000毫秒内对话框打开了，

填写"d:\text.txt"文件路径并按下打开按钮，并将执行结果返回到result

例子2:

```
--打开文件对话框事件响应函数，第一个参数是数字型返回值，第二个值是返回值的详细说明
function event_fill_opendlg_exit(exitcode,exitcodestring)
    sys.Print("\\r\\n FillOpenDlg函数执行结果 = ",exitcodestring,"(",exitcode,")")
    --event.StringOfFillOpendlgCallBackExitCode(exitcode) --返回值转换成说明性的文字，方便阅读
end"

--分配一个回调函数给FillOpenDlg的返回事件(可选)
--event.Assign(EVENT_FILLOPENDLG_CALLBACK,event_fill_opendlg_exit)

fresult = window.FillOpenDlg("","d:\text.txt",1,5000,event_fill_opendlg_exit) --效果与上一个函数相同
```

执行上一个动作后，先将线程创建结果返回到fresult。

然后等待5000毫秒打开文件对话框，如5000毫秒内对话框打开了，填写"d:\text.txt"文件路径并按下打开按钮，
执行完毕后调用event_fill_opendlg_exit函数并结果在函数内显示。;

声明:执行成功标志位 = window.FitView(句柄,主窗口序号)

说明:设置窗口适应相应的视图。

参数:

句柄:需要操作的窗口句柄

主窗口序号:需要调整到的主界面的视图序号，0~3，0左上，1右上，2左下，3右下。

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

result = window.FitView(handle,0)

设置handle的窗体适应相应的左上视图，并将执行结果返回到result;

声明:执行成功标志位 = window.SetView(高度比例,宽度比例)

说明:设置窗口的视图比例。

参数:

高度比例:0~100。

宽度比例:0~100。

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

result = window.SetView(10,20)

设置窗体子视图的纵横比，并将执行结果返回到result;

声明:执行成功标志位,R,G,B = window.GetPixel(句柄,窗口x偏移位置，窗口y偏移位置)

说明:读取窗口的一个像素点的颜色。

参数:

句柄:窗口的句柄。

窗口x偏移位置:相对位置，可以是正值，可以是负值，原点在改句柄所在窗口的左上角。

窗口y偏移位置:相对位置，可以是正值，可以是负值，原点在改句柄所在窗口的左上角。。

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

RGB:颜色值

例子:

result,R,G,B = window.GetPixel(buttonhwnd,2,2)

读取buttonhwnd窗口2,2位置的一个像素点的颜色值，并将执行结果返回到result;

声明1:执行成功标志位,窗口可见状态 = window.Visible(句柄)
声明2:执行成功标志位 = window.Visible(句柄,窗口可见状态)

说明:查询或者设置窗口是否可见。

参数:句柄

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

result,s = window.Visible(handle)

查询handle的窗体是否可见，执行结果返回到result,状态返回s;

声明1:窗口使能状态 = window.Enable(句柄)
声明2:执行成功标志位 = window.Enable(句柄,窗口使能状态)

说明:查询或者设置窗口是否可用。

参数:句柄

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

s = window.Enable(handle)

查询handle的窗体的可用性，状态返回s。;

声明1:执行成功标志位,左边位置,上边位置,宽度,高度 = window.Size(句柄)

声明2:执行成功标志位 = window.Size(句柄,左边位置,上边位置,宽度,高度)

声明3:执行成功标志位,{左边位置,上边位置,宽度,高度} = window.Size(句柄,{})

声明4:执行成功标志位 = window.Size(句柄,{左边位置,上边位置,宽度,高度})

说明:设置或者读取窗口位置以及大小。

参数:句柄

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

result= window.Size(handle,{10,20,200,300})

设置handle的窗体的位置和大小，并将执行结果返回到result。;

声明:执行成功标志位 = window.DrawText(句柄,需要显示的文字)

说明:在指定窗体上显示文字。

参数:句柄，需要显示的文字，空值显示当前句柄。

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

```
result = window.DrawText(handle,"")
```

在handle窗体上显示句柄文字，并将执行结果返回到result;

App库函数列表

app.AppExecute
app.AppShellExecute
app.AppClose
app.AppKill;

声明:执行成功标志位,句柄 = app.AppExecute(文件路径)

说明:打开应用程序。

参数:文件路径

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

句柄:执行成功返回 句柄，反之返回 nil

例子:

```
result,h = app.AppExecute("c:\\test.exe")
```

打开应用程序，并将执行结果返回到result,句柄返回到h;

;

声明:执行成功标志位 = app.AppShellExecute(文件路径,参数)

说明:打开应用程序。

参数:文件路径，参数

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

```
result = app.AppExecute("NotePad","c:\\test.txt")
```

打开记事本应用程序，并将c:\\test.txt载入进来，执行结果返回到result;

声明:执行成功标志位 = app.AppClose(句柄)

说明:关闭应用程序。

参数:句柄

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

result = app.AppClose(hwnd)

关闭hwnd的应用程序，执行结果返回到result;

声明1:执行成功标志位 = app.AppKill(句柄)

声明2:执行成功标志位 = app.AppKill(句柄数组)

说明:强制关闭应用程序。

参数:句柄或者句柄数组。

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

result = app.AppKill(hwnd)

强制关闭hwnd的应用程序，执行结果返回到result;

Usb库函数列表

usb.OpenByIndex

usb.Close

usb.isDeviceConnected

usb.Reset

usb.GetLibraryVersion

usb.GetConnectedDevices

usb.GetFactorySerialNumber

usb.I2cCancelCurrentTransfer

usb.I2cRead

usb.I2cWrite

usb.SetAdvancedCommParams

usb.SetSpeed

usb.SmbusWriteByte

usb.SmbusReadByte

usb.SmbusWriteWord

usb.SmbusReadWord
usb.SmbusBlockWrite
usb.SmbusBlockRead
usb.SmbusSendByte
usb.SmbusReceiveByte
usb.GetHwFwRevisions
usb.GetLastError
usb.ErrorCode2String
usb.SetResetPinStatus
usb.GetResetPinStatus
usb.GetInterruptPinFlag
usb.GetInterruptEdgeSetting
usb.SetInterruptEdgeSetting
usb.ClearInterruptPinFlag
usb.EEPROMRead
usb.EEPROMWrite;

声明1:句柄,连接成功标志位 = usb.OpenByIndex(无参数)
声明2:句柄,连接成功标志位 = usb.OpenByIndex(插入的设备序号)

说明:按设备顺序号打开USB Bridge。

参数:无或者插入的设备序号。

返回值:

句柄:执行成功返回 设备的句柄，反之返回 -1

连接成功标志位:执行成功返回 1，反之返回 0

例子:

result,isConnedted= usb.OpenByIndex()

读取当前设备的连接状态，并将执行结果返回到result,isConnedted;

声明1:执行成功标志位 = usb.Close(无参数)

声明2:执行成功标志位 = usb.Close(指定待关闭设备的句柄)

说明:关闭USB Bridge。

参数:无或者指定待关闭设备的句柄。

返回值:

执行成功标志位:执行成功返回 1，反之返回 0

例子:

result= usb.Close()

声明1:USBBridge是否已经连接标志位 = usb.isDeviceConnected(无参数)

说明:获取USB Bridge的连接状态。

参数:无

返回值:

是否已经连接标志位:连接上返回 1，反之返回 0

例子:

isConnected = usb.isDeviceConnected()

读取设备的连接状态，并将执行结果返回到isConnected;

声明1:执行结果志位 = usb.Reset(无参数)

说明:将连接上的USB Bridge的连接状态重启复位。

参数:无

返回值:

是否已经连接标志位:成功返回 0，失败范围非0。

例子:

ret = usb.Reset()

将设备重启，并将执行结果返回到ret;

声明1:文本类型的库的版本号 = usb.GetLibraryVersion(无参数)

说明:获取运行库的文本类型的版本号。

参数:无

返回值:

文本类型的库的版本号。

例子:

str = usb.GetLibraryVersion()

获取运行库的文本类型的版本号，并将执行结果返回到str;

声明1:USBBridge数量,执行成功标志位 = usb.GetConnectedDevices(无参数)

说明:获取插入USB总线的USBBridge数量。

参数:无

返回值:

USBBridge数量:返回当前插入USB总线的USBBridge数量。

执行成功标志位:成功返回 0，失败范围非0。。

例子:

num, ret = usb.GetConnectedDevices()

获取插入USB总线的USBBridge数量，并将执行结果返回到 num, ret。;

声明1:序列号,执行成功标志位 = usb.GetFactorySerialNumber(无参数)

说明:获取设备序列号。

参数:无

返回值:

序列号:返回当前插入USB总线的USBBridge文本格式序列号。

执行成功标志位:成功返回 0，失败范围非0。。

例子:

serial, ret = usb.GetFactorySerialNumber()

获取插入USB总线的USBBridge设备序列号，并将执行结果返回到 serial, ret 。;

声明1:执行成功标志位 = usb.I2cCancelCurrentTransfer(无参数)

说明:中断当前IIC传输(每当执行IIC读写不成功，强烈建议执行该语句，以防止IIC死锁)。

参数:无

返回值:

执行成功标志位:成功返回 0，失败范围非0。。

例子:

ret = usb.I2cCancelCurrentTransfer()

中断当前IIC传输，并将执行结果返回到 ret 。;

声明1:{数据},执行成功标志位 = usb.I2cRead(bytesToRead,slaveAddress,use7bitAddress)

说明:IIC读操作。

参数:

bytesToRead:需要读出字节数的数量。

slaveAddress:从设备的器件地址。

use7bitAddress:从设备的器件地址是7bit地址写本参数写1,从设备的器件地址是8bit地址写本参数写0。

返回值:

{数据}:读出的数据，数组形式。

执行成功标志位:成功返回 0，失败范围非0。。

例子:

data,ret = usb.I2cRead(2,0x27,1)

读取二代扩展板的供电电平状态存储到data，并将执行结果返回到 ret 。;

声明1:执行成功标志位 = usb.I2cWrite({待写入的数据数组},slaveAddress,use7bitAddress)

说明:IIC读操作。

参数:

{待写入的数据数组}:把需要写入的数据存到数组内。

slaveAddress:从设备的器件地址。

use7bitAddress:从设备的器件地址是7bit地址写本参数写1,从设备的器件地址是8bit地址写本参数写0。

返回值:

执行成功标志位:成功返回 0，失败范围非0。。

例子:

ret = usb.I2cWrite({0x00,0xaa},0x27,1)

设置二代扩展板的供电电平状态为奇数点亮，并将执行结果返回到 ret 。;

声明1:执行成功标志位 = usb.SetAdvancedCommParams(超时时长,最大尝试次数)

说明:设置IIC高级参数。

参数:

超时时长:IIC总线读写失败时，自动尝试的间隔时长。

最大尝试次数:IIC总线读写失败时，自动尝试的最大尝试次数。

返回值:

执行成功标志位:成功返回 0，失败范围非0。。

例子:

ret = usb.SetAdvancedCommParams(3,5)

设置IIC总线读写失败时重试超时为3ms，最大尝试5次，并将执行结果返回到 ret 。;

声明1:执行成功标志位 = usb.SetSpeed(速度)

说明:设置IIC的CLK的速度。

参数:

速度:IIC总线的CLK速度，46875~500000。

返回值:

执行成功标志位:成功返回 0，失败范围非0。。

例子:

ret = usb.SetSpeed(400000)

设置IIC总线读写速度为400k，并将执行结果返回到 ret 。;

声明1:Returns = usb.SmbusWriteByte(slaveAddress,use7bitAddress,usePec,command,data)

Description: SMBus write byte. The first byte of a Write Byte operation is the command code.

The next one is the data to be written.。

```
    Parameters:
        slaveAddress:7bit or 8bit SMBus slave address, depending on the value of the "use7bitAddress" flag.
                     For 8 bit addresses, the R/W LSB of the address is set to 0 inside the function.
        use7bitAddress:if > 0 - 7 bit address will be used for the slave. If 0 - 8 bit is used.
        usePec:if > 0 Packet Error Checking (PEC) will be used. A PEC byte containing the CRC8
                 value for the sent message is appended after the data byte.
        command:The command code byte.
        data:The data byte.

    Returns:
        0 for success; error code otherwise.

Example:
    ret = usb.SmbusWriteByte(0x21,1,0,0x10,0x55)
    Set the bus type to smbus ,and send the 0x10 command with data 0x55 write to device address 0x21,result returned to 'ret' 。;
```

```
声明1:readByte,Returns = usb.SmbusReadByte(slaveAddress,use7bitAddress,usePec,command)

    Description: SMBus Read Byte. First Write the command byte to the slave, then read one data byte back.

    Parameters:
        slaveAddress:7bit or 8bit SMBus slave address, depending on the value of the "use7bitAddress" flag.
                     For 8 bit addresses, the R/W LSB of the address is set to 0 inside the function.
```

```
        use7bitAddress:if > 0 - 7 bit address will be used for the slave. If 0 - 8 bit is used.
        usePec:if > 0 Packet Error Checking (PEC) will be used. A PEC byte containing the CRC8
                 value for the sent message is appended after the data byte.
        command:The command code byte.



    readByte:The readed data byte.
    Returns:
        0 for success; error code otherwise.

Example:
    data, ret = usb.SmbusReadByte(0x21,1,0,0x10)
    Set the bus type to smbus ,and get the 0x10 command data frome the device address 0x21,result returned to data,'ret' 。;
```

```
声明1:Returns = usb.SmbusWriteWord(slaveAddress,use7bitAddress,usePec,command,wdata)

    Description: SMBus write word. The first byte of a Write Byte operation is the command code,
    followed by the data_byte_low then data_byte_high.

    Parameters:
        slaveAddress:7bit or 8bit SMBus slave address, depending on the value of the "use7bitAddress" flag.
                     For 8 bit addresses, the R/W LSB of the address is set to 0 inside the function.
        use7bitAddress:if > 0 - 7 bit address will be used for the slave. If 0 - 8 bit is used.
        usePec:if > 0 Packet Error Checking (PEC) will be used. A PEC byte containing the CRC8
                 value for the sent message is appended after the data byte.
```

```
        command:The command code byte.
        wdata:The data word.

    Returns:
        0 for success; error code otherwise.

Example:
    ret = usb.SmbusWriteWord(0x21,1,0,0x10,0x55AA)
    Set the bus type to smbus ,and send the 0x10 command with data 0x55AA write to device address 0x21,result returned to 'ret' 。;
```

```
声明1:readWord,Returns = usb.SmbusReadWord(slaveAddress,use7bitAddress,usePec,command)

    Description: SMBus Read Word. First Write the command byte to the slave, then read one data byte back.

    Parameters:
        slaveAddress:7bit or 8bit SMBus slave address, depending on the value of the "use7bitAddress" flag.
                     For 8 bit addresses, the R/W LSB of the address is set to 0 inside the function.
        use7bitAddress:if > 0 - 7 bit address will be used for the slave. If 0 - 8 bit is used.
        usePec:if > 0 Packet Error Checking (PEC) will be used. A PEC byte containing the CRC8
                 value for the sent message is appended after the data byte.
        command:The command code byte.


    readWord:The readed data word.
    Returns:
```

```
        0 for success; error code otherwise.

Example:
    data, ret = usb.SmbusReadWord(0x21,1,0,0x10)
    Set the bus type to smbus ,and get the 0x10 command data frome the device address 0x21,result returned to data,'ret' 。;

声明1:Returns = usb.SmbusBlockWrite(slaveAddress,use7bitAddress,usePec,command,{byteArry})

    Description: SMBus Block Write. The first byte of a Block Write operation is the command code,
    followed by the number of data bytes, then data bytes.

    Parameters:
        slaveAddress:7bit or 8bit SMBus slave address, depending on the value of the "use7bitAddress" flag.
                     For 8 bit addresses, the R/W LSB of the address is set to 0 inside the function.
        use7bitAddress:if > 0 - 7 bit address will be used for the slave. If 0 - 8 bit is used.
        usePec:if > 0 Packet Error Checking (PEC) will be used. A PEC byte containing the CRC8
                 value for the sent message is appended after the data byte.
        command:The command code byte.
        {byteArry}:Array containing the data bytes to be sent to the slave.

    Returns:
        0 for success; error code otherwise.

Example:
    ret = usb.SmbusBlockWrite(0x21,1,0,0x10,{0x55,0xAA})
    Set the bus type to smbus ,and send the 0x10 command with data 0x55AA write to device address 0x21,result returned to 'ret' 。;
```

声明1:{byteArry},Returns = usb.SmbusBlockRead(slaveAddress,use7bitAddress,usePec,command,byteCount)

Description: SMBus Read Word. First Write the command byte to the slave, then read one data byte back.

Parameters:

slaveAddress:7bit or 8bit SMBus slave address, depending on the value of the "use7bitAddress" flag.
For 8 bit addresses, the R/W LSB of the address is set to 0 inside the function.

use7bitAddress:if > 0 - 7 bit address will be used for the slave. If 0 - 8 bit is used.

usePec:if > 0 Packet Error Checking (PEC) will be used. A PEC byte containing the CRC8
value for the sent message is appended after the data byte.

command:The command code byte.

byteCount:(block size) the number of data bytes that the slave will send to the master.
Valid range is between 1 and 255 bytes. If there is a mismatch between this value
and the byteCount the slave reports that it will send, an error will be returned.

{byteArry}:The readed data byte.

Returns:0 for success; error code otherwise.

Example:

data, ret = usb.SmbusBlockRead(0x21,1,0,0x10,10)

Set the bus type to smbus ,and get the 0x10 command 10 datas frome the device address 0x21,result returned to data,'ret' 。;

声明1:Returns = usb.SmbusSendByte(slaveAddress,use7bitAddress,usePec,data)

```
    Description:SMBus Send byte. Sends one data byte.

    Parameters:
        slaveAddress:7bit or 8bit SMBus slave address, depending on the value of the "use7bitAddress" flag.
                    For 8 bit addresses, the R/W LSB of the address is set to 0 inside the function.
        use7bitAddress:if > 0 - 7 bit address will be used for the slave. If 0 - 8 bit is used.
        usePec:if > 0 Packet Error Checking (PEC) will be used. A PEC byte containing the CRC8
                value for the sent message is appended after the data byte.
        data:The data byte.

    Returns:
        0 for success; error code otherwise.

Example:
    ret = usb.SmbusSendByte(0x21,1,0,0x55)
    Set the bus type to smbus ,and send the data 0x55 write to device address 0x21,result returned to 'ret' 。;
```

```
声明1:readByte,Returns = usb.SmbusReceiveByte(slaveAddress,use7bitAddress,usePec)

    Description: SMBus Read Byte. First Write the command byte to the slave, then read one data byte back.

    Parameters:
        slaveAddress:7bit or 8bit SMBus slave address, depending on the value of the "use7bitAddress" flag.
                    For 8 bit addresses, the R/W LSB of the address is set to 0 inside the function.
```

```
        use7bitAddress:if > 0 - 7 bit address will be used for the slave. If 0 - 8 bit is used.
        usePec:if > 0 Packet Error Checking (PEC) will be used. A PEC byte containing the CRC8
                value for the sent message is appended after the data byte.


    readByte:The readed data byte.
    Returns:0 for success; error code otherwise.

Example:
    data, ret = usb.SmbusReceiveByte(0x21,1,0)
    Set the bus type to smbus ,and get the 1 byte frome the device address 0x21,result returned to data,'ret' 。;
```

```
声明1:Returns = usb.GetHwFwRevisions(hardwareRevision, firmwareRevision)

    Description: Reads the hardware and firmware revision values from the device.

    Parameters:
        hardwareRevision - will contain the hardware revision string.
        firmwareRevision - will contain the firmware revision string.

    Returns:0 for success; error code otherwise.

Example:
    s1, s2 = usb.GetHwFwRevisions(0x21,1,0)
    Get the hard firmware version types ,result returned to data,'ret' 。;
```

声明1:lasterrorcode = usb.GetLastError()

Description: Gets the last error value. Used only for the Open methods.

Parameters: no

Returns:The value for the last error code.

Example:

```
err = usb.GetLastError();
```

声明1:stringerrorcode = usb.ErrorCode2String(errorcode)

Description: Cover number errorcode to string errorcode .

Parameters: The value for the last error code

Returns:The value for the string name of last error code.

Example:

```
errstr = usb.ErrorCode2String(-1);
```

声明1:ret = usb.SetResetPinStatus(status) Description: Set the reset pin status . Parameters: The value for the rst pin, 0 is high logic level. Returns:0 for success; error code otherwise. Example: ret = usb.SetResetPinStatus(0);
声明1:status,ret = usb.GetResetPinStatus() Description: Get the reset pin status . Parameters: none Returns:status is the value for the rst pin, 0 is high logic level, ret 0 for success; error code otherwise. Example: status,ret = usb.GetResetPinStatus();

声明1:flag,ret = usb.GetInterruptPinFlag()

Description: Get the interrupt pin flag .

Parameters: none

Returns:flag is the value for the interrupt pin, 1 is logic level changed, ret 0 for success; error code otherwise.

Example:

```
flag,ret = usb.GetInterruptPinFlag();
```

声明1:interruptPinMode,ret = usb.GetInterruptEdgeSetting()

Description: Gets the interrupt pin trigger configuration. .

Parameters: none

Returns:

interruptPinMode:

0 - none

1 - positive edge

2 - negative edge

```
                              3 - both
    ret: 0 for success; error code otherwise.

Example:
    interruptPinMode,ret = usb.GetInterruptEdgeSetting();

声明1:ret = usb.SetInterruptEdgeSetting(interruptPinMode)

    Description: Sets the interrupt pin trigger configuration. .

    Parameters:interruptPinMode
                              0 - none
                              1 - positive edge
                              2 - negative edge
                              3 - both

    Returns:
    ret: 0 for success; error code otherwise.

Example:
    ret = usb.SetInterruptEdgeSetting(interruptPinMode);
```

```
声明1:ret = usb.ClearInterruptPinFlag()
```

```
    Description: Clears the interrupt pin flag of a device.

    Parameters:none

    Returns:
    ret: 0 for success; error code otherwise.

Example:
    ret = usb.ClearInterruptPinFlag();
```

```
Type 1:{pData},ret = usb.EEPROMRead(nStart,nDataLength)
Type 2:{pData},ret = usb.EEPROMRead(bAddress,nStart,nDataLength)

    Description: Read AT24XXX serial EEPROM.

    Parameters:
             nStart:Start to read position.
             nDataLength:Read length.
             bAddress:The user EEPROM address. Default is internal EEPROM.

    Returns:
    {pData}:returned data arry.
    ret: 0 for success; error code otherwise.
```

Example:

```
pData,ret = usb.EEPROMRead(0,10);
```

Type 1:ret = usb.EEPROMWrite({pData},nStart)
Type 2:ret = usb.EEPROMWrite({pData},nStart,bAddress)

Description: Write AT24XXX serial EEPROM.

Parameters:

{pData}:To Write data arry.
nStart:Start to read position.
bAddress:The user EEPROM address. Default is internal EEPROM.

Returns:

ret: 0 for success; error code otherwise.

Example:

```
ret = usb.EEPROMWrite({1,2,3,4},0);
```

Event库函数列表

event.Assign

event.Unassign
event.SyncCallBack
event.Notify
event.TextChanged
event.MonitorClose
event.StringOfInterruptCode
event.StringOfFillOpendlgCallBackExitCode;

声明:event.StringOfFillOpendlgCallBackExitCode(退出码)

说明:将打开文件对话框的事件响应回调函数的退出码翻译为文本，方便打印输出。

参数:退出码。

返回值:对应的字符串;

声明:event.StringOfInterruptCode(中断号)

说明:将中断号翻译为文本，方便打印输出。

参数:中断号。

EVENT_FILLOPENDLG_CALLBACK :打开对话框线程回掉事件
EVENT_MAIN_WINDOW_SIZED :主窗口大小改变事件
EVENT_MAIN_WINDOW_MOVED :主窗口移动事件

返回值:对应的字符串;

声明:event.MonitorClose(监视器序号)

说明:关闭一个监视器，释放内部资源，不再响应事件。
重要说明:最多只能关闭8个监视器，分别对应MONITOR_1~MONITOR_8

参数:监视器序号MONITOR_1~MONITOR_8，必须用程序内置的定义值。否则会引发不可预知的错误。

返回值:无

例子:

```
--关闭一个监视器。
event.MonitorClose(MONITOR_1);
```

声明:event.TextChanged(句柄,事件发生类型,需要比对的字符串,监视器序号,事件响应函数)

说明:创建一个监视器，监视有效句柄对应的窗体文字，例如按钮的文字或者文本框的文字，
当有对应的事件发生类型，则触发一次事件响应函数。
如果不执行MonitorClose直接重新调用TextChanged会触发一次MONITOR退出事件，
因为内部要先销毁线程再新创建线程。
重要说明:最多只能创建8个监视器，分别对应MONITOR_1~MONITOR_8

参数1:句柄。需要监测的句柄，必须是一个有效的句柄，否则将会创建监视器不成功。

参数2:事件发生类型

TEXT_CHANGED :文本只要发生改变即触发事件

TEXT_COMMON :文本完全和比对的字符串完全相等才触发事件

TEXT_DIFFERENT :文本只要和和比对的字符串不相等就触发事件

TEXT_LIKE :指窗体文本包含有比对的字符串就触发事件

参数3:监视器序号MONITOR_1~MONITOR_8，必须用程序内置的定义值。否则会引发不可预知的错误。

参数4:事件响应函数，如有对应的事件发生则调用该函数。

返回值:创建成功与否

例子:

```
--创建一个监视器1，监视本程序的标题栏文字，如果有文字发生变动，则调用txt_changed1函数。
event.TextChanged(MAIN_HWND,TEXT_CHANGED," ",MONITOR_1,txt_changed1)

--事件响应函数
function txt_changed1(code,str) --第一个参数是返回的事件类型码，第二个参数是监控的句柄的改变后的字符串
    sys.Print("\r\n function txt_changed1:  " .. string.format("code = %d,%s",code,str))
end;
```

声明:event.Notify(中断号)

说明:主动触发一次中断号对应的中断函数，如果未分配中断函数则不会执行。

参数:中断号

EVENT_FILLOPENDLG_CALLBACK :打开对话框线程回掉事件
EVENT_MAIN_WINDOW_SIZED :主窗口大小改变事件
EVENT_MAIN_WINDOW_MOVED :主窗口移动事件

返回值:无

例子:

```
--主动触发一次EVENT_MAIN_WINDOW_MOVED对应的中断函数，如果未分配中断函数则不会执行。
event.Notify(EVENT_MAIN_WINDOW_MOVED);
```

声明:event.Unassign(中断号)

说明:剥离中断响应函数，再发生中断时不再响应之前分配的函数。

参数:中断号

EVENT_FILLOPENDLG_CALLBACK :打开对话框线程回掉事件
EVENT_MAIN_WINDOW_SIZED :主窗口大小改变事件
EVENT_MAIN_WINDOW_MOVED :主窗口移动事件

返回值:无

例子:

```
--剥离中断响应函数，再发生中断时不再响应之前分配的EVENT_MAIN_WINDOW_MOVED中断函数
event.Unassign(EVENT_MAIN_WINDOW_MOVED);
```

声明:event.SyncCallBack(延时,回调函数)

说明:异步延时调用一个回调函数，在回调之前不会响应新的回调发起。

参数:延时，毫秒数
回调函数 :需要自定义回调的函数名称

返回值:是否设置成功

例子:

```
--1秒异步延时调用一个回调函数
event.SyncCallBack(1000,test_sync_recall);
```

声明:event.Assign(中断号,中断响应函数)

说明:分配一个中断响应函数给一个中断号，当中断发生的时候会调用该中断响应函数。

参数:中断号
EVENT_FILLOPENDLG_CALLBACK :打开对话框线程回掉事件
EVENT_MAIN_WINDOW_SIZED :主窗口大小改变事件
EVENT_MAIN_WINDOW_MOVED :主窗口移动事件
参数:中断响应函数

自己定义的一个函数

返回值:无

例子:

```
function interrupt_windowmoved()    --窗体移动
    print "\r\n interrupt_windowmoved"
end
--分配interrupt_windowmoved          函数给窗体移动中断
event.Assign(EVENT_MAIN_WINDOW_MOVED,interrupt_windowmoved)
--分配以后，当有中断EVENT_MAIN_WINDOW_MOVED发生时，在程序运行期间，则会调用interrupt_windowmoved函数。；
```

扩展事件函数

点击运行按钮调用的事件
OnEventPreRunProgram()

点击暂停按钮调用的事件
OnEventPauseProgram()

点击继续按钮调用的事件

OnEventResumeProgram()

点击停止按钮调用的事件
OnEventStopProgram()

点击标志位按钮调用的事件
OnEventFlagChanged();