ART7003 数字多用表卡

WIN2000/XP 驱动程序使用说明书

阿尔泰科技发展有限公司

产品研发部修订

请您务必阅读《使用纲要》，它会使您事半功倍!

目　录

第一章 版权信息

第二章 使用纲要

第一节、使用上层用户函数，高效、简单

如果您只关心通道及频率等基本参数，而不必了解复杂的硬件知识和控制细节，那么我们强烈建议您使用上层用户函数，它们就是几个简单的形如Win32 API的函数，具有相当的灵活性、可靠性和高效性。而底层用户函数如WriteRegisterULong、ReadRegisterULong、WritePortByte、ReadPortByte……则是满足了解硬件知识和控制细节、且又需要特殊复杂控制的用户。但不管怎样，我们强烈建议您使用上层函数（在这些函数中，您见不到任何设备地址、寄存器端口、中断号等物理信息，其复杂的控制细节完全封装在上层用户函数中。）对于上层用户函数的使用，您基本上不必参考硬件说明书，除非您需要知道板上插座等管脚分配情况。

第二节、如何管理设备

由于我们的驱动程序采用面向对象编程，所以要使用设备的一切功能，则必须首先用CreateDevice函数创建一个设备对象句柄hDevice，有了这个句柄，您就拥有了对该设备的绝对控制权。然后将此句柄作为参数传递给相应的驱动函数，最后可以通过ReleaseDevice将hDevice释放掉。

第三节、哪些函数对您不是必须的

公共函数如CreateFileObject，WriteFile，ReadFile等一般来说都是辅助性函数，除非您要使用存盘功能。如果您使用上层用户函数访问设备，那么GetDeviceAddr，WriteRegisterByte，WriteRegisterWord，WriteRegisterULong，ReadRegisterByte，ReadRegisterWord，ReadRegisterULong等函数您可完全不必理会，除非您是作为底层用户管理设备。而WritePortByte，WritePortWord，WritePortULong，ReadPortByte，ReadPortWord，ReadPortULong则对ART用户来讲，可以说完全是辅助性，它们只是对我公司驱动程序的一种功能补充，对用户额外提供的，它们可以帮助您在NT、Win2000 等操作系统中实现对您原有传统设备如ISA卡、串口卡、并口卡的访问，而没有这些函数，您可能在基于Windows NT架构的操作系统中无法继续使用您原有的老设备。

第三章 ART 设备操作函数接口介绍

第一节、设备驱动接口函数总列表（每个函数省略了前缀“ART7003_”）

函数名	函数功能	备注
① 设备对象操作函数		
CreateDevice	创建 ART 设备对象(用设备逻辑号)	上层及底层用户
GetDeviceCurrentBaseAddr	取得当前设备基地址	上层及底层用户
ReleaseDevice	关闭设备，且释放 ART 总线设备对象	上层及底层用户
② 设备功能控制函数		
ResetDevice	复位设备	上层用户
SetDeviceFuntion	万用表功能档选择	上层用户
SetDeviceSecFuntion	选择万用表第二测量功能	上层用户
GetDeviceSecFuntion	获取万用表第二测量功能	上层用户
ReadDeviceData	读取设备采集数据	上层用户
GetDeviceInputRange	获取各功能档下设备输入量程	上层用户
SetInputRangeAdd	执行该函数，量程加 1 档	上层用户
SetInputRangeDec	执行该函数，量程减 1 档	上层用户
③ 设备校准函数		
WriteCalibrationData	写入校准数据	上层用户
ReadCalibrationData	读出校准数据	上层用户
④ 硬件参数操作函数		
LoadBaseAddr	将基地址从系统中读出	上层用户
SaveBaseAddr	将基地址保存至系统中	上层用户

使用需知：

Visual C++：

要使用如下函数关键的问题是：

首先，必须在您的源程序中包含如下语句：

#include “C:\Art\ART7003\INCLUDE\ART7003.H”

注：以上语句采用默认路径和默认板号，应根据您的板号和安装情况确定 ART7003.H 文件的正确路径，当然也可以把此文件拷到您的源程序目录中。

另外，要在 VB 环境中用子线程以实现高速、连续数据采集与存盘，请务必使用 VB5.0 版本。当然如果您有 VB6.0 的最新版，也可以实现子线程操作。

LabVIEW/CVI :

LabVIEW 是美国国家仪器公司(National Instrument)推出的一种基于图形开发、调试和运行程序的集成化环境，是目前国际上唯一的编译型的图形化编程语言。在以 PC 机为基础的测量和工控软件中，LabVIEW 的市场普及率仅次于 C++/C 语言。LabVIEW 开发环境具有一系列优点，从其流程图式的编程、不需预先编译就存在的语法检查、调试过程使用的数据探针，到其丰富的函数功能、数值分析、信号处理和设备驱动等功能，都令人称道。关于 LabView/CVI 的进一步介绍请见本文最后一部分关于 LabView 的专述。其驱动程序接口单元模块的使用方法如下：

一、在 LabView 中打开 ART7003.VI 文件，用鼠标单击接口单元图标，比如 CreateDevice 图标

然后按 Ctrl+C 或选择 LabView 菜单 Edit 中的 Copy 命令，接着进入用户的应用程序 LabView 中，按 Ctrl+V 或选择 LabView 菜单 Edit 中的 Paste 命令，即可将接口单元加入到用户工程中，然后按以下函数原型说明或演示程序的说明连接该接口模块即可顺利使用。

二、根据 LabView 语言本身的规定，接口单元图标以黑色的较粗的中间线为中心，以左边的方格为数据输入端，右边的方格为数据的输出端。

三、在单元接口图标中，凡标有“I32”为有符号长整型 32 位数据类型，“U16”为无符号短整型 16

位数据类型，“［U16］”为无符号 16 位短整型数组或缓冲区或指针，“［U32］”与“［U16］”同理，只是位数不一样。

第二节、设备对象管理函数原型说明

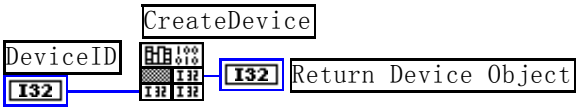
♦ 创建设备对象函数（逻辑号）

函数原型：

Visual C++：

HANDLE CreateDevice (WORD wBaseAddress)

LabVIEW：



功能：该函数使用逻辑号创建设备对象，并返回其设备对象句柄 hDevice。只有成功获取 hDevice，您才能实现对该设备所有功能的访问。

参数：

返回值：如果执行成功，则返回设备对象句柄；如果没有成功，则返回错误码 INVALID_HANDLE_VALUE。由于此函数已带容错处理，即若出错，它会自动弹出一个对话框告诉您出错的原因。您只需要对此函数的返回值作一个条件处理即可，别的任何事情您都不必做。

相关函数： CreateDevice GetDeviceCurrentBaseAddr ReleaseDevice

Visual C++程序举例

```
    :
HANDLE hDevice;     // 定义设备对象句柄
int DeviceLgcID = 0;
hDevice = ART7003_CreateDevice (DeviceLgcID);   // 创建设备对象,并取得设备对象句柄
if(hDevice == INVALIDE_HANDLE_VALUE);     // 判断设备对象句柄是否有效
  {
    return;     // 退出该函数
 }
    :
```

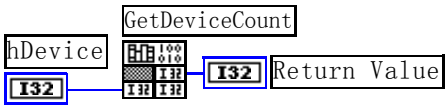
♦ 取得当前设备基地址

函数原型：

Visual C++:

WORD GetDeviceCurrentBaseAddr (HANDLE hDevice)

LabVIEW：



功能：取得当前设备基地址。

参数：

hDevice设备对象句柄，它应由CreateDevice创建。

返回值：成功，则返回 TRUE，否则返回 FALSE。

相关函数： CreateDevice GetDeviceCurrentBaseAddr ReleaseDevice

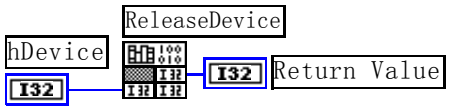
♦ 释放设备对象所占的系统资源及设备对象

函数原型：

Visual C++:

BOOL ReleaseDevice (HANDLE hDevice)

LabVIEW：



功能：释放设备对象所占用的系统资源及设备对象自身。
参数：
hDevice设备对象句柄，它应由CreateDevice创建。
返回值：若成功，则返回TRUE，否则返回FALSE，用户可以用GetLastErrorEx捕获错误码。
相关函数： CreateDevice　　GetDeviceCurrentBaseAddr　　ReleaseDevice

应注意的是，CreateDevice必须和ReleaseDevice函数一一对应，即当您执行了一次CreateDevice后，再一次执行这些函数前，必须执行一次ReleaseDevice函数，以释放由CreateDevice占用的系统软硬件资源，如DMA控制器、系统内存等。只有这样，当您再次调用CreateDevice函数时，那些软硬件资源才可被再次使用。

第三节、设备功能控制函数原型说明

- **复位设备**

函数原型：
Visual C++:
BOOL ResetDevice (HANDLE hDevice)

LabVIEW:
请参考相关演示程序。

功能：复位设备。
参数：
hDevice 设备对象句柄，它应由CreateDevice创建。
返回值：若调用成功则返回 TRUE，否则返回 FALSE。
相关函数： CreateDevice　　GetDevWorkSts　　ResetDevice
SetDeviceFunction　　SetDeviceSecFunction　　GetDeviceSecFunction
ReadDeviceData　　GetDeviceInputRange　　SetInputRangeAdd
SetInputRangeDec　　ReleaseDevice

- **万用表功能档选择**

函数原型：
Visual C++:
BOOL SetDeviceFunction (HANDLE hDevice,
LONG lFunction)

LabVIEW:
请参考相关演示程序。

功能：万用表功能档选择。
参数：
hDevice 设备对象句柄，它应由 CreateDevice 创建。
lFunction 功能档选择。
返回值：若调用成功则返回 TRUE，否则返回 FALSE。
相关函数： CreateDevice　　GetDevWorkSts　　ResetDevice
SetDeviceFunction　　SetDeviceSecFunction　　GetDeviceSecFunction

ReadDeviceData GetDeviceInputRange SetInputRangeAdd
SetInputRangeDec ReleaseDevice

♦ **选择万用表第二测量功能**

函数原型：

Visual C++:

BOOL SetDeviceSecFunction (HANDLE hDevice)

LabVIEW:

请参考相关演示程序。

功能：选择万用表第二测量功能。

参数：

hDevice 设备对象句柄，它应由 CreateDevice 创建。

返回值：若调用成功则返回 TRUE，否则返回 FALSE。

相关函数： CreateDevice GetDevWorkSts ResetDevice
SetDeviceFunction SetDeviceSecFunction GetDeviceSecFunction
ReadDeviceData GetDeviceInputRange SetInputRangeAdd
SetInputRangeDec ReleaseDevice

♦ **获取万用表第二测量功能**

函数原型：

Visual C++:

BOOL GetDeviceSecFunction (HANDLE hDevice,
LONG lFunction,
PLONG plSecondFunction)

LabVIEW:

请参考相关演示程序。

功能：获得万用表的第二测量功能。

参数：

hDevice 设备对象句柄，它应由 CreateDevice 创建。

lFunction 功能档位的选择。

plSecondFunction 第二测量功能的选择。

返回值：若调用成功则返回 TRUE，否则返回 FALSE。

相关函数： CreateDevice GetDevWorkSts ResetDevice
SetDeviceFunction SetDeviceSecFunction GetDeviceSecFunction
ReadDeviceData GetDeviceInputRange SetInputRangeAdd
SetInputRangeDec ReleaseDevice

♦ **读取设备采集数据**

函数原型：

Visual C++:

BOOL ReadDeviceData (HANDLE hDevice，
PLONG plADData)

LabVIEW:

请参考相关演示程序。

功能：读取设备采集的数据。

参数：

hDevice 设备对象句柄，它应由CreateDevice创建。

plADData 采集数据。

返回值：若调用成功则返回 TRUE，否则返回 FALSE。

相关函数： CreateDevice GetDevWorkSts ResetDevice SetDeviceFunction SetDeviceSecFunction GetDeviceSecFunction ReadDeviceData GetDeviceInputRange SetInputRangeAdd SetInputRangeDec ReleaseDevice

◆ **获取各功能档下设备输入量程**

函数原型：

Visual C++:

```
BOOL GetDeviceInputRange (HANDLE hDevice,
                          LONG lFunction,
                          PLONG plInputRange)
```

LabVIEW:

请参考相关演示程序。

功能：获取各功能档下设备输入量程。

参数：

hDevice 设备对象句柄，它应由 CreateDevice 创建。

lFunction 功能档。

plInputRange 设备输入量程。

返回值：如果初始化设备对象成功，则返回 TRUE，否则返回 FALSE。

相关函数： CreateDevice GetDevWorkSts ResetDevice SetDeviceFunction SetDeviceSecFunction GetDeviceSecFunction ReadDeviceData GetDeviceInputRange SetInputRangeAdd SetInputRangeDec ReleaseDevice

◆ **执行该函数，量程加 1 档**

Visual C++:

```
BOOL SetInputRangeAdd (HANDLE hDevice)
```

LabVIEW:

请参考相关演示程序。

功能：执行该函数，量程加 1 档。

参数：

hDevice设备对象句柄，它应由CreateDevice创建。

返回值：如果初始化设备对象成功，则返回 TRUE，否则返回 FALSE， 用户可用 GetLastErrorEx 捕获当前误码，并加以分析。

相关函数： CreateDevice GetDevWorkSts ResetDevice SetDeviceFunction SetDeviceSecFunction GetDeviceSecFunction ReadDeviceData GetDeviceInputRange SetInputRangeAdd SetInputRangeDec ReleaseDevice

◆ **执行该函数，量程减 1 档**

Visual C++:

```
BOOL SetInputRangeDec (HANDLE hDevice)
```

LabVIEW:

请参考相关演示程序。

功能：执行该函数，量程减 1 档。

参数：

hDevice设备对象句柄，它应由CreateDevice创建。

返回值：如果初始化设备对象成功，则返回 TRUE，否则返回 FALSE， 用户可用 GetLastErrorEx 捕获当前误码，并加以分析。

相关函数： CreateDevice GetDevWorkSts ResetDevice
SetDeviceFunction SetDeviceSecFunction GetDeviceSecFunction
ReadDeviceData GetDeviceInputRange SetInputRangeAdd
SetInputRangeDec ReleaseDevice

第四节、设备校准函数原型说明

◆ **写入校准数据**

函数原型：

Visual C++:

```
BOOL WriteCalibrationData (HANDLE hDevice,
                           LONG lfunction,
                           LONG lSecondFunction,
                           LONG lInputRange,
                           LONG lCalMode,
                           LONG lCalData)
```

LabVIEW:

请参考相关演示程序。

功能：写入校准数据。

参数：

hDevice设备对象句柄，它应由CreateDevice创建。

lFunction 功能档选择，此处只有电压、电流和电阻档选择。

lSecondFunction 第二测量功能（当功能档为选择电流档时，此参数需要选择直流或交流）。

lInputRange 量程选择。

lCalMode 校准模式，0 为零点校准，1 为满度校准。

lCalData 校准值的范围为 0-65535。

返回值：若成功，返回 TRUE，否则返回 FALSE。

相关函数： CreateDevice WriteCalibrationData ReadCalibrationData
ReleaseDevice

◆ **读出校准数据**

函数原型：

Visual C++:

```
BOOL ReadCalibrationData (HANDLE hDevice,
                          LONG lfunction,
                          LONG lSecondFunction,
                          LONG lInputRange,
                          LONG lCalMode,
                          PLONG PlCalData)
```

LabVIEW:

请参考相关演示程序。

功能：读出校准数据。
参数：
hDevice设备对象句柄，它应由CreateDevice创建。
lFunction 功能档选择，此处只有电压、电流和电阻档选择。
lSecondFunction 第二测量功能（当功能档为选择电流档时，此参数需要选择直流或交流）。
lInputRange 量程选择。
lCalMode 校准模式，0 为零点校准，1 为满度校准。
PlCalData 校准值的范围为 0-65535。
返回值：若成功，返回 TRUE，否则返回 FALSE。
相关函数: CreateDevice WriteCalibrationData ReadCalibrationData ReleaseDevice

第五节、硬件参数操作函数

◆ **将基地址从系统中读出**

函数原型：
Visual C++:
BOOL LoadBaseAddr (HANDLE hTDevice, PWORD pBaseAddr)
LabVIEW:
请参考相关演示程序。

功能：将基地址从系统中读出。
参数：
hDevice设备对象句柄，它应由CreateDevice创建。
返回值：若成功，返回 TRUE，否则返回 FALSE。
相关函数: CreateDevice LoadBaseAddr SaveBaseAddr ReleaseDevice

◆ **将基地址保存至系统中**

函数原型：
Visual C++:
BOOL SaveBaseAddr (HANDLE hTDevice, WORD wBaseAddr)
LabVIEW:
请参考相关演示程序。

功能：将基地址保存至系统中。
参数：
hDevice设备对象句柄，它应由CreateDevice创建。
返回值：若成功，返回 TRUE，否则返回 FALSE。
相关函数: CreateDevice LoadBaseAddr SaveBaseAddr ReleaseDevice

第四章 功能参数选择结构

函数 ART7003_SetDeviceFunction 中的 lFunction 参数所使用功能挡选项如下：

常量名	常量值	功能定义
ART7003_FUNCTIONG_DCmV	0x00	直流 mV 电压档
ART7003_FUNCTIONG_DCV	0x01	直流 V 电压档
ART7003_FUNCTIONG_ACV	0x02	交流 V 电压档
ART7003_FUNCTIONG_OHM	0x03	电阻测量/通断测量功能

ART7003_FUNCTIONG_mA	0x04	mA 电流档
ART7003_FUNCTIONG_uA	0x05	uA 电流档
ART7003_FUNCTIONG_DIODE	0x06	测量二极管正向压降

函数 ART7003_GetDeviceSecFunction 中的 lFunction 参数选择为 A、mA、uA 档时，lSecondFunction 参数所使用选项如下：

常量名	常量值	功能定义
ART7003_SECFUNCTIONG_DCA	0x00	直流电流档
ART7003_SECFUNCTIONG_ACA	0x01	交流电流档

函数 ART7003_GetDeviceSecFunction 中的 lFunction 参数选择为电阻测量/通断测量功能档时，lSecondFunction 参数所使用选项如下：

常量名	常量值	功能定义
ART7003_SECFUNCTIONG_RESISTANCE	0x00	电阻测量
ART7003_SECFUNCTIONG_ONOFF	0x01	通断测量

函数 ART7003_GetDeviceInputRange 中的 lFunction 参数选择直流 mV 电压档时，参数 plInputRange 获得的量程选项如下：

常量名	常量值	功能定义
ART7003_DCmV_INPUT_N0_P500mV	0x00	0～500mV 量程，增益为 1 倍
ART7003_DCmV_INPUT_N0_P50mV	0x01	0～50mV 量程，增益为 0.1 倍

函数 ART7003_GetDeviceInputRange 中的 lFunction 参数选择直流 V 电压档时，参数 plInputRange 获得的量程选项如下：

常量名	常量值	功能定义
ART7003_DCV_INPUT_N0_P1000V	0x00	0～1000V 量程，增益为 2000 倍
ART7003_DCV_INPUT_N0_P500V	0x01	0～500V 量程，增益为 1000 倍
ART7003_DCV_INPUT_N0_P50V	0x02	0～50V 量程，增益为 100 倍
ART7003_DCV_INPUT_N0_P5V	0x03	0～5V 量程，增益为 10 倍

函数 ART7003_GetDeviceInputRange 中的 lFunction 参数选择交流 V 电压档时，参数 plInputRange 获得的量程选项如下：

常量名	常量值	功能定义
ART7003_ACV_INPUT_N0_P1000V	0x00	0～1000V 量程，增益为 2000 倍
ART7003_ACV_INPUT_N0_P500V	0x01	0～500V 量程，增益为 1000 倍
ART7003_ACV_INPUT_N0_P50V	0x02	0～50V 量程，增益为 100 倍
ART7003_ACV_INPUT_N0_P5V	0x03	0～5V 量程，增益为 10 倍

函数 ART7003_GetDeviceInputRange 中的 lFunction 参数选择电阻测量/通断测量功能时并且函数 ART7003_SetDeviceSecFunction 中的 lSecondFunction 参数选择电阻测量功能时，参数 plInputRange 获得的量程选项如下：

常量名	常量值	功能定义
ART7003_OHM_INPUT_N0_P50M	0x00	0～50MΩ 量程，Ref=10M
ART7003_OHM_INPUT_N0_P5M	0x01	0～5MΩ 量程，Ref=10M
ART7003_OHM_INPUT_N0_P500K	0x02	0～500KΩ 量程，Ref=1.1111M
ART7003_OHM_INPUT_N0_P50K	0x03	0～50KΩ 量程，Ref=100.01K
ART7003_OHM_INPUT_N0_P5K	0x04	0～5KΩ 量程，Ref=10.01K
ART7003_OHM_INPUT_N0_P500OHM	0x05	0～500Ω 量程，Ref=1.001K

函数 ART7003_GetDeviceInputRange 中的 lFunction 参数选择 mA 电流档时，参数 plInputRange 获得的量程选项如下：

常量名	常量值	功能定义
ART7003_mA_INPUT_N0_P500mA	0x00	0～500mA 量程
ART7003_mA_INPUT_N0_P50mA	0x01	0～50mA 量程

函数 ART7003_GetDeviceInputRange 中的 lFunction 参数选择 uA 电流档时，
参数 plInputRange 获得的量程选项如下：

常量名	常量值	功能定义
ART7003_uA_INPUT_N0_P5000uA	0x00	0～5000uA 量程
ART7003_uA_INPUT_N0_P500uA	0x01	0～500uA 量程

第五章 共用函数介绍

这部分函数不参与本设备的实际操作，它只是为您编写数据采集与处理程序时的有力手段，使您编写应用程序更容易，使您的应用程序更高效。

第一节、公用接口函数总列表（每个函数省略了前缀“ART7003_”）

函数名	函数功能	备注
①ISA 总线 I/O 端口操作函数		
WritePortByte	以字节(8Bit)方式写 I/O 端口	用户程序操作端口
WritePortWord	以字(16Bit)方式写 I/O 端口	用户程序操作端口
WritePortULong	以无符号双字(32Bit)方式写 I/O 端口	用户程序操作端口
ReadPortByte	以字节(8Bit)方式读 I/O 端口	用户程序操作端口
ReadPortWord	以字(16Bit)方式读 I/O 端口	用户程序操作端口
ReadPortULong	以无符号双字(32Bit)方式读 I/O 端口	用户程序操作端口
②文件对象操作函数		
CreateFileObject	初始设备文件对象	适用于所有设备
WriteFile	请求文件对象写用户数据到磁盘文件	适用于所有设备
ReadFile	请求文件对象读数据到用户空间	适用于所有设备
SetFileOffset	设置文件指针偏移	适用于所有设备
GetFileLength	取得文件长度	适用于所有设备
ReleaseFile	释放已有的文件对象	适用于所有设备
GetDiskFreeBytes	取得指定磁盘的可用空间(字节)	适用于所有设备
③ 辅助操作函数		
GetDevVersion	获取设备固件及程序版本	底层用户
CreateSystemEvent	创建系统内核事件对象	用于线程同步
ReleaseSystemEvent	释放系统内核事件对象	用于线程同步
CreateVBThread	创建 VB 子线程	用于线程同步
TerminateVBThread	释放 VB 子线程	用于线程同步
DelayTimeUs	微秒级延时函数	

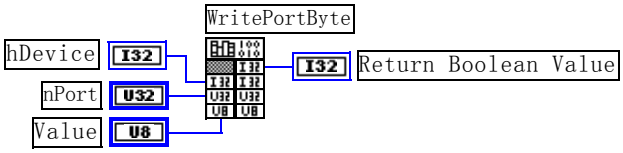
第二节、IO 端口读写函数原型说明

- **以单字节(8Bit)方式写 I/O 端口**

Visual C++:

```
BOOL WritePortByte (HANDLE hDevice,
                    UINT nPort,
                    BYTE Value)
```

LabVIEW:



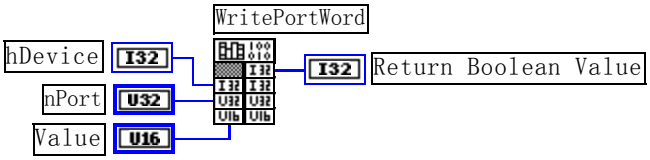
功能：以单字节(8Bit)方式写 I/O 端口。
参数：
hDevice 设备对象句柄，它应由CreateDevice创建。
nPort 指定寄存器的物理基地址。
OffsetBytes 相对于物理基地址的偏移位置(字节)。
Value 写入由 nPort 指定端口的值。
返回值：若成功，返回 TRUE，否则返回 FALSE。
相关函数： CreateDevice WritePortByte WritePortWord
WritePortULong ReadPortByte ReadPortWord

- **以双字(16Bit)方式写 I/O 端口**

Visual C++:

```
BOOL WritePortWord (HANDLE hDevice,
                    UINT nPort,
                    WORD Value)
```

LabVIEW:



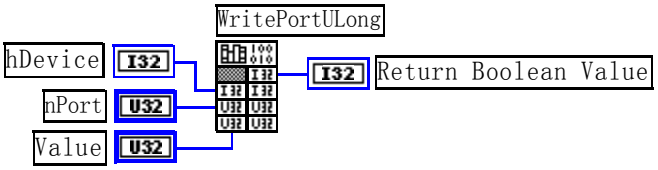
功能：以双字(16Bit)方式写 I/O 端口。
参数：
hDevice设备对象句柄，它应由CreateDevice创建。
nPort 指定寄存器的物理基地址。
Value 写入由 nPort 指定端口的值。
返回值：若成功，返回 TRUE，否则返回 FALSE。
相关函数： CreateDevice WritePortByte WritePortWord
WritePortULong ReadPortByte ReadPortWord

- **以四字节(32Bit)方式写 I/O 端口**

Visual C++:

```
BOOL WritePortULong (HANDLE hDevice,
                     UINT nPort,
                     ULONG Value)
```

LabVIEW:



功能：以四字节(32Bit)方式写 I/O 端口。
参数：
hDevice 设备对象句柄，它应由CreateDevice创建。

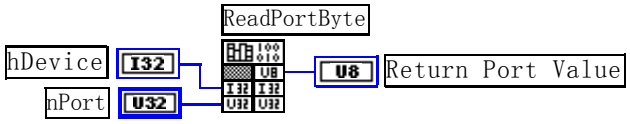
nPort 指定寄存器的物理基地址。
Value　写入由 nPort 指定端口的值。
返回值：若成功，返回 TRUE，否则返回 FALSE。
相关函数： CreateDevice　WritePortByte　WritePortWord
WritePortULong　ReadPortByte　ReadPortWord

- **以单字节(8Bit)方式读 I/O 端口**

Visual C++:

BYTE ReadPortByte (HANDLE hDevice,
UINT nPort)

LabVIEW:



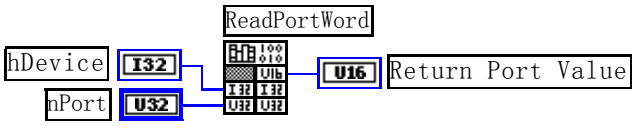
功能：以单字节(8Bit)方式读 I/O 端口。
参数：
hDevice设备对象句柄，它应由CreateDevice创建。
nPort 指定寄存器的物理基地址。
返回值：返回由 nPort 指定的端口的值。
相关函数： CreateDevice　WritePortByte　WritePortWord
WritePortULong　ReadPortByte　ReadPortWord

- **以双字节(16Bit)方式读 I/O 端口**

Visual C++:

WORD ReadPortWord (HANDLE hDevice,
UINT nPort)

LabVIEW:



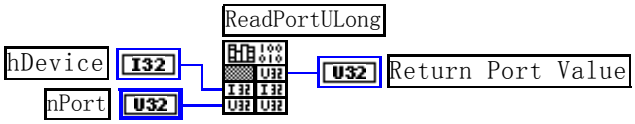
功能：以双字节(16Bit)方式读 I/O 端口。
参数：
hDevice设备对象句柄，它应由CreateDevice创建。
nPort 指定寄存器的物理基地址。
返回值：返回由 nPort 指定的端口的值。
相关函数： CreateDevice　WritePortByte　WritePortWord
WritePortULong　ReadPortByte　ReadPortWord

- **以四字节(32Bit)方式读 I/O 端口**

Visual C++:

ULONG ReadPortULong (HANDLE hDevice,
UINT nPort)

LabVIEW:



功能：以四字节(32Bit)方式读 I/O 端口。
参数：
hDevice设备对象句柄，它应由CreateDevice创建。
nPort 指定寄存器的物理基地址。
返回值：返回由 nPort 指定端口的值。
相关函数： CreateDevice WritePortByte WritePortWord WritePortULong ReadPortByte ReadPortWord

第三节、辅助操作函数原型说明

- **获取设备固件及程序版本**

函数原型：
Visual C++:

```
BOOL GetDevVersion (HANDLE hDevice,
                    PULONG pulFmwVersion
                    PULONG pulDriverVersion)
```

LabVIEW:
请参考相关演示程序。

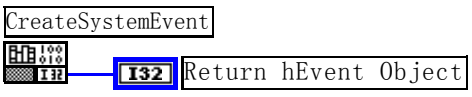
功能：获取设备固件及程序版本。
参数：
hDevice设备对象句柄，它应由CreateDevice创建。
pulFmwVersion 固件版本。
pulDriverVersion 驱动版本。
返回值：若成功，返回 TRUE，否则返回 FALSE。
相关函数： CreateDevice GetDevVersion CreateSystemEvent ReleaseSystemEvent CreateVBThread TerminateVBThread DelayTimeUs ReleaseDevice

- **创建内核系统事件**

函数原型：
Visual C++:

```
HANDLE CreateSystemEvent (void)
```

LabVIEW:



功能：创建系统内核事件对象，它将被用于中断事件响应或数据采集线程同步事件。
参数：无任何参数。
返回值：若成功，返回系统内核事件对象句柄，否则返回－1(或 INVALID_HANDLE_VALUE)。

- **释放内核系统事件**

函数原型：
Visual C++:

BOOL ReleaseSystemEvent (HANDLE hEvent)

LabVIEW:

请参见相关演示程序。

功能：释放系统内核事件对象。

参数：

hEvent　被释放的内核事件对象。它应由CreateSystemEvent成功创建的对象。

返回值：若成功，则返回 TRUE。

相关函数： CreateDevice　GetDevVersion　CreateSystemEvent
ReleaseSystemEvent　CreateVBThread　TerminateVBThread
DelayTimeUs　ReleaseDevice

♦在 VB 环境中，创建子线程对象，以实现多线程操作

Visual C++:

BOOL CreateVBThread (HANDLE* hThread, LPTHREAD_START_ROUTINE RoutineAddr)

功能：该函数在 VB 环境中解决了不能实现或不能很好地实现多线程的问题。通过该函数用户可以很轻松地实现多线程操作。

参数：

Thread　若成功创建子线程，该参数将返回所创建的子线程的句柄，当用户操作这个子线程时将用到这个句柄，如启动线程、暂停线程以及删除线程等。

RoutineAddr 作为子线程运行的函数的地址，在实际使用时，请用 AddressOf 关键字取得该子线程函数的地址，再传递给 CreateVBThread 函数。

返回值：当成功创建子线程时，返回 TRUE，且所创建的子线程为挂起状态，用户需要用 Win32 API 函数 Resume Thread 函数启动它。若失败，则返回 FALSE，用户可用 GetLastError 捕获当前错误码。

相关函数： CreateDevice　GetDevVersion　CreateSystemEvent
ReleaseSystemEvent　CreateVBThread　TerminateVBThread
DelayTimeUs　ReleaseDevice

♦ 在 VB 中，释放子线程对象

Visual C++:

BOOL TerminateVBThread (HANDLE hThread)

功能：在 VB 中释放由 CreateVBThread 创建的子线程对象。

参数：

Thread　指向需要释放的子线程对象的句柄，它应由 CreateVBThread 创建。

返回值：当成功释放子线程对象时，返回 TRUE，否则返回 FALSE，用户可用 GetLastError 捕获当前错误码。

相关函数： CreateDevice　GetDevVersion　CreateSystemEvent
ReleaseSystemEvent　CreateVBThread　TerminateVBThread
DelayTimeUs　ReleaseDevice

♦ 微秒级延时函数

Visual C++:

BOOL DelayTimeUs (HANDLE hDevice, LONG nTimeUs)

功能：微秒级延时函数。

参数：

返回值：当成功释放子线程对象时，返回 TRUE，否则返回 FALSE。

相关函数： CreateDevice　GetDevVersion　CreateSystemEvent

ReleaseSystemEvent CreateVBThread TerminateVBThread
DelayTimeUs ReleaseDevice

第四节、文件对象操作函数原型说明

♦ **创建文件对象**

函数原型：

Visual C++:

```
HANDLE CreateFileObject (HANDLE hDevice,
                         LPCTSTR NewFileName,
                         int Mode)
```

LabVIEW:

请参见相关演示程序。

功能：初始化设备文件对象， 以期待 WriteFile 请求准备文件对象进行文件操作。

参数：

hDevice设备对象句柄，它应由CreateDevice创建。

NewFileName 新文件名。

Mode 文件操作方式，所用的文件操作方式控制字定义如下(可通过或指令实现多种方式并操作)：

常量名	常量值	功能定义
ART7003_modeRead	0x0000	只读文件方式
ART7003_modeWrite	0x0001	只写文件方式
ART7003_modeReadWrite	0x0002	既读又写文件方式
ART7003_modeCreate	0x1000	如果文件不存在可以创建该文件， 如果存在，则重建此文件，且清 0
ART7003_typeText	0x4000	以文本方式操作文件

返回值：若成功，则返回文件对象句柄。

相关函数： CreateDevice CreateFileObject WriteFile
ReadFile ReleaseFile ReleaseDevice
SetFileOffset GetFileLength GetDiskFreeBytes

♦ **通过设备对象，往指定磁盘上写入用户空间的采样数据**

函数原型：

Visual C++:

```
BOOL WriteFile (HANDLE hFileObject,
                PVOID pDataBuffer,
                ULONG nWriteSizeBytes)
```

LabVIEW:

详见相关演示程序。

功能：通过向设备对象发送“写磁盘消息”，设备对象便会以最快的速度完成写操作。注意为了保证写入的数据是可用的，这个操作将与用户程序保持同步，但与设备对象中的环形内存池操作保持异步，以得到更高的数据吞吐量，其文件名及路径应由CreateFileObject函数中的strFileName指定。

参数：

hFileObject 设备对象句柄，它应由CreateFileObject创建。

pDataBuffer 用户数据空间地址，可以是用户分配的数组空间。

nWriteSizeBytes 告诉设备对象往磁盘上一次写入数据的长度(以字节为单位)。

返回值：若成功，则返回TRUE，否则返回FALSE，用户可以用GetLastErrorEx捕获错误码。

相关函数： CreateDevice CreateFileObject WriteFile
ReadFile ReleaseFile ReleaseDevice

SetFileOffset GetFileLength GetDiskFreeBytes

◆ **通过设备对象,从指定磁盘文件中读采样数据**

函数原型：

Visual C++:

```
BOOL ReadFile (HANDLE hFileObject,
               PVOID pDataBuffer,
               ULONG nOffsetBytes,
               ULONG nReadSizeBytes)
```

LabVIEW:

详见相关演示程序。

功能：将磁盘数据从指定文件中读入用户内存空间中，其访问方式可由用户在创建文件对象时指定。

参数：

hFileObject 设备对象句柄，它应由CreateFileObject创建。

pDataBuffer 用于接受文件数据的用户缓冲区指针，可以是用户分配的数组空间。

nOffsetBytes 指定从文件开始端所偏移的读位置。

nReadSizeBytes 告诉设备对象从磁盘上一次读入数据的长度(以字为单位)。

返回值：若成功，则返回TRUE，否则返回FALSE，用户可以用GetLastErrorEx捕获错误码。

相关函数： CreateDevice CreateFileObject WriteFile
ReadFile ReleaseFile ReleaseDevice
SetFileOffset GetFileLength GetDiskFreeBytes

◆ **设置文件偏移位置**

函数原型：

Visual C++:

```
BOOL SetFileOffset (HANDLE hFileObject,
                    ULONG nOffsetBytes)
```

LabVIEW:

详见相关演示程序。

功能：设置文件偏移位置，用它可以定位读写起点。

参数：hFileObject 文件对象句柄，它应由CreateFileObject创建。

返回值：若成功，则返回 TRUE，否则返回 FALSE。

相关函数： CreateDevice CreateFileObject WriteFile
ReadFile ReleaseFile ReleaseDevice
SetFileOffset GetFileLength GetDiskFreeBytes

◆ **取得文件长度（字节）**

函数原型：

Visual C++:

```
ULONG GetFileLength (HANDLE hFileObject)
```

LabVIEW:

详见相关演示程序。

功能：取得文件长度。

参数：hFileObject 设备对象句柄，它应由CreateFileObject创建。

返回值：若成功，则返回>1，否则返回 0，用户可以用GetLastErrorEx捕获错误码。

相关函数： CreateDevice CreateFileObject WriteFile

ReadFile ReleaseFile ReleaseDevice
SetFileOffset GetFileLength GetDiskFreeBytes

♦ **释放设备文件对象**

函数原型：

Visual C++:

BOOL ReleaseFile (HANDLE hFileObject)

LabVIEW:

详见相关演示程序。

功能：释放设备文件对象。

参数：hFileObject 设备对象句柄，它应由CreateFileObject创建。

返回值：若成功，则返回TRUE，否则返回FALSE，用户可以用GetLastErrorEx捕获错误码。

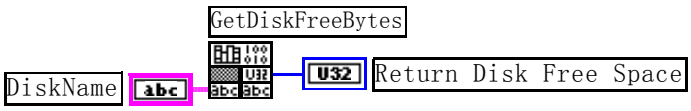
相关函数： CreateDevice CreateFileObject WriteFile
ReadFile ReleaseFile ReleaseDevice
SetFileOffset GetFileLength GetDiskFreeBytes

♦ **取得指定磁盘的可用空间**

Visual C++:

ULONGLONG GetDiskFreeBytes (LPCTSTR DiskName)

LabVIEW:



功能：取得指定磁盘的可用剩余空间(以字为单位)。

参数：DiskName 需要访问的盘符，若为 C 盘为"C:\\"， D 盘为"D:\\"，以此类推。

返回值：若成功，返回大于或等于 0 的长整型值，否则返回零值，用户可用GetLastErrorEx捕获错误码。注意使用 64 位整型变量。

相关函数： CreateDevice CreateFileObject WriteFile
ReadFile ReleaseFile ReleaseDevice
SetFileOffset GetFileLength GetDiskFreeBytes