CH341A的I2C接口Labview应用说明

注： 按照下面方法生成的子VI需要重新处理，才可以在所有的labview版本中使用；

处理方法：

打开子VI，－程序面板――打开‘节点调用’－将调用规范由原来的C调用更改为stdcall ――然后保存。

<http://wch.cn/>

CH341A 可以做同步串口、异步串口、并口等使用。

一，基础操作

A ，相关文件下载

1，在官网下载CH341PRT.zip（并口＋同步串口驱动）CH341SER.EXE(异步串口驱动)

2，在官网上面下载动态库文件和相应的头文件：CH341dll.dll（CH341dll.h针对并口、同步串口）,CH341PT.dll（CH341PAR.h针对异步串口）,USBMON.dll（针对USB设备自我测试） 共3个dll文件。

下载方法：进官网－在线下载－关键字搜索（CH341）－下载需要的文件

针对并口和同步串口的，需要下载CH341PAR.ZIP；其他请咨询客服

B，驱动安装，成功安装后将相应的硬件CH341插入电脑，此时PC显示有新硬件接入，并自动更新驱动。

C，硬件查看：桌面－我的电脑－右键－设备管理器－外部接口－USB-EPP/I2CCH341 ，如果没有则需要重新安装驱动等，请咨询客服。

D，硬件识别：

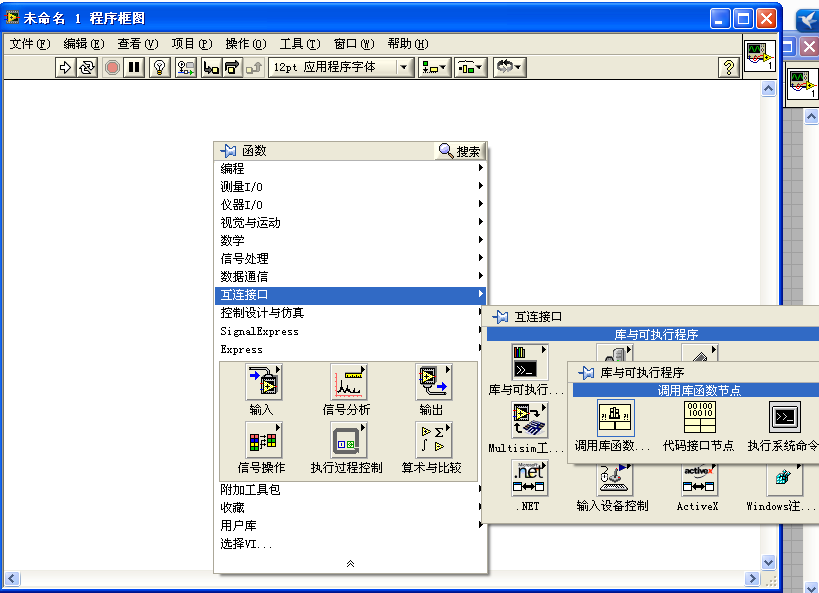
可以通过Labview 的Visa-Driver Wizard 来给CH341A做USB的驱动，但是在具有dll文件时，不适应。如此操作后，Labview可以识别CH341，但是Labview编译环境下面通过DLL生成的VI函数反而不认识CH341了。



调用CH341的VI时，VI需要提供一个iIndex的参数，这个参数一般为默认值0，即PC上面只有1个CH341时，VI自动将该设备的标识号设定为0，如果有多个，则自动根据插入的先后顺序，按照0,1,2~n的序列给各个VI提供iIndex参数。

二、Labview dll库处理

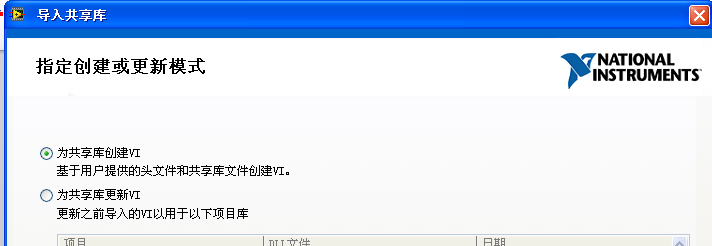
Labview有调用库函数节点操作，但该操作需要自己设置函数参数的类型等，不适合初用者。这部分的操作本文不予说明。故我们常使用labview的导入共享库操作，该操作需要dll文件和相应的\*.h 头文件。具体操作如下



1，打开Labview －工具－导入－共享库（dll）



2，选择为共享库创建VI

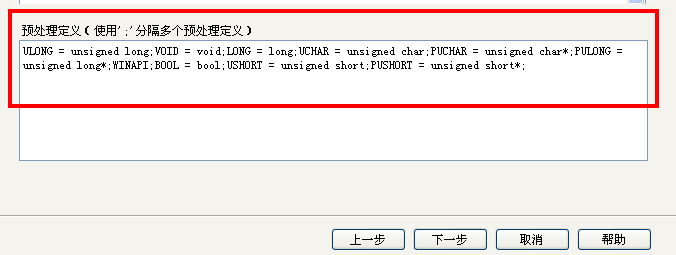


3，导入相应的dll文件和h文件



4，下一步的配置包括路径和预处理定义栏做如下处理：

ULONG = unsigned long;VOID = void;LONG = long;UCHAR = unsigned char;PUCHAR = unsigned char\*;PULONG = unsigned long\*;WINAPI;BOOL = bool;USHORT = unsigned short;PUSHORT = unsigned short\*;



这样做的原因请参阅附件文章‘Labview编程调用DLL时遇见的问题’（可baidu）

5，然后一直‘下一步’直到完成。记住生成的VI放置路径，以便后续调用这些VI。

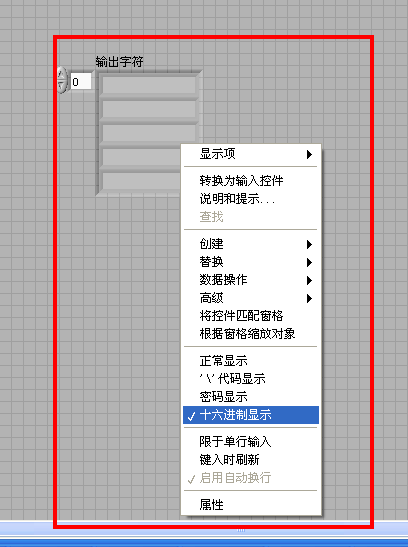
三、创建labview 应用程序

A ，CH341的应用序列为： CH341Open Device―相关应用函数（CH341Write I2C.vi等）－CH341Close Device

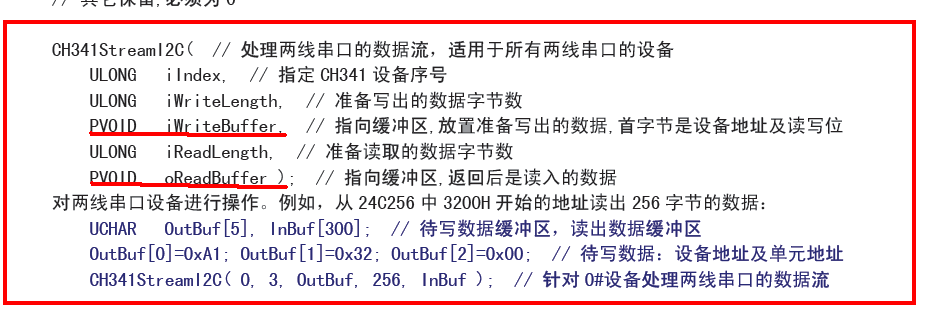
可以参考附件实例‘CH341A-a~d.VI’这些例子以USB转I2C为主

B，部分子VI处理：

1. CH341Read I2C.vi : 该VI的输出为字符格式，与我们的常规应用不一样，这时可以做如下操作：（注：此时显示的是字符ASCII码的数字，其中0为空格

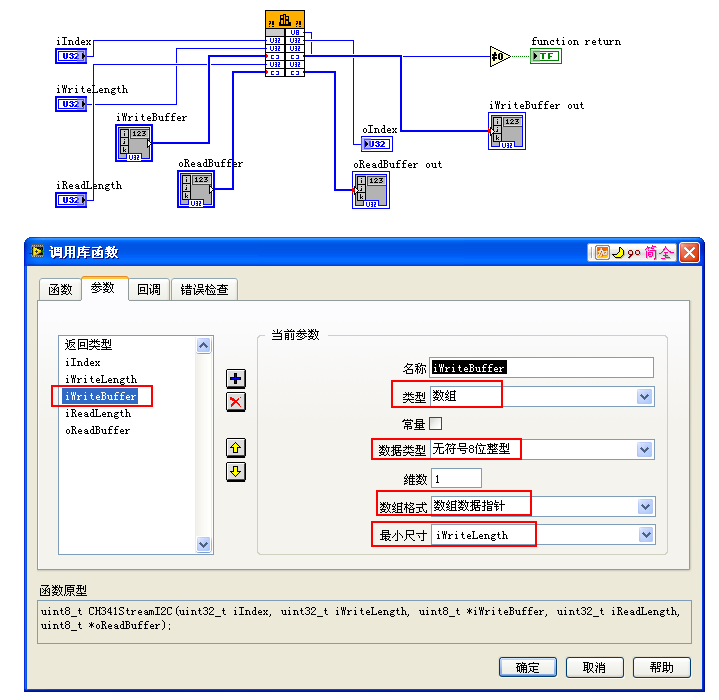


2 ，CH341Stream I2C.vi ，该VI的输入参数如下图：



其中iWriteBuffer和OReadBuffer为PVOID类型，其函数本身传递的值为数字（指针数值），但在Labview应用中却应该为数组。系统自动生成的VI实际使用时会报错。因此需要对该VI的库节点调用做修正，如下图所示：，将iWriteBuffer和OReadBuffer的类型修改为数组，并根据实际应用环境设置数据类型（对于一般的I2C，常为无符号的8位整形）、数组格式选择数组数据指针，最小尺寸必须设置（不设定就会报错）。

建议：其他函数中的PVOID也做相应处理。



3，为了适应labview的数据流传递方式，建议将设备的iIndex连线端修改为输入输出端 ,这样的话，VI就可以顺序执行了。

