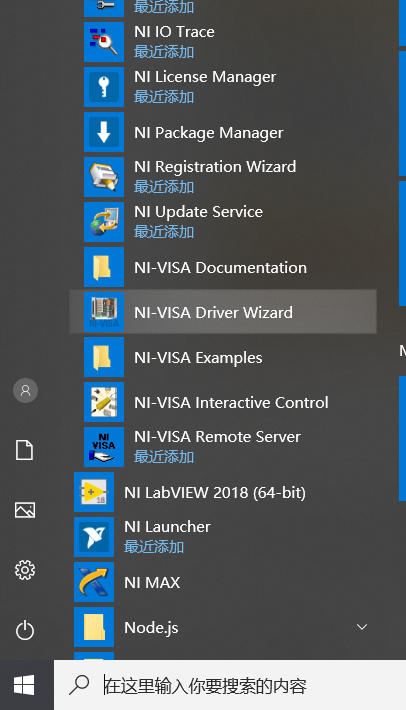
**一、前期准备**

    首先介绍一下USB设备（STM32 USB），设备通过一个接口2个端点与主机通信，两个端点分别为端点1输入和端点1输出。两个端点均为中断端点。

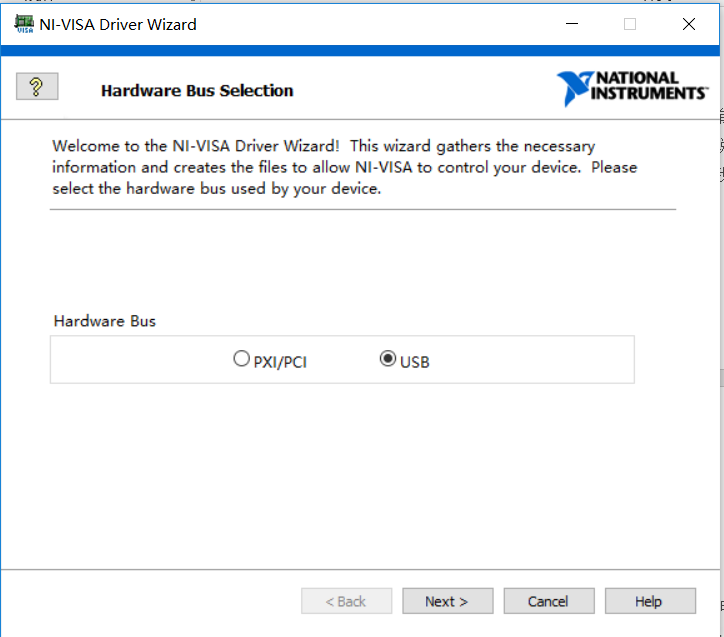
    好，下面来正式介绍一下基于labview的USB调试小助手。

    USB口的HID类能轻易的被windows系统识别，因为它内置了USB HID驱动。但是这个驱动是不被labview承认的。要想被labview承认，需要安装VISA生成的USB驱动。（个人感觉这个是labview比较不好的地方）具体的操作方法是：

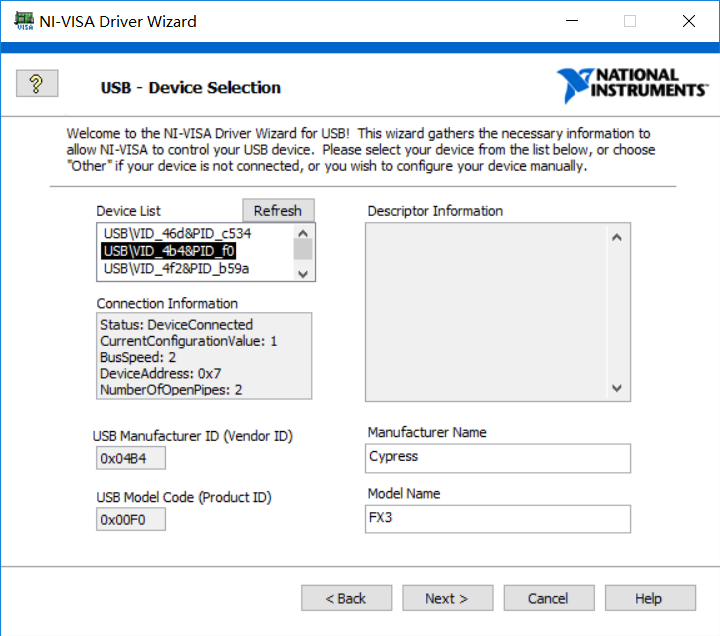
打开NI-VISA driver wizard



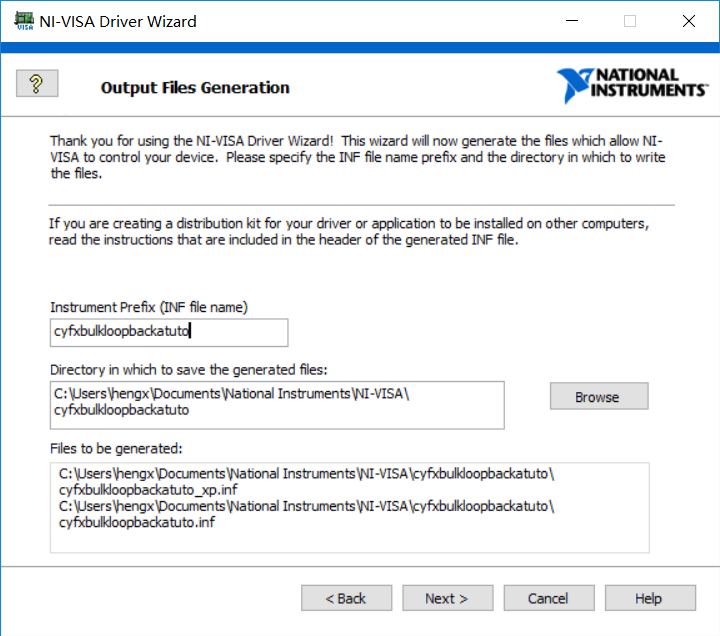
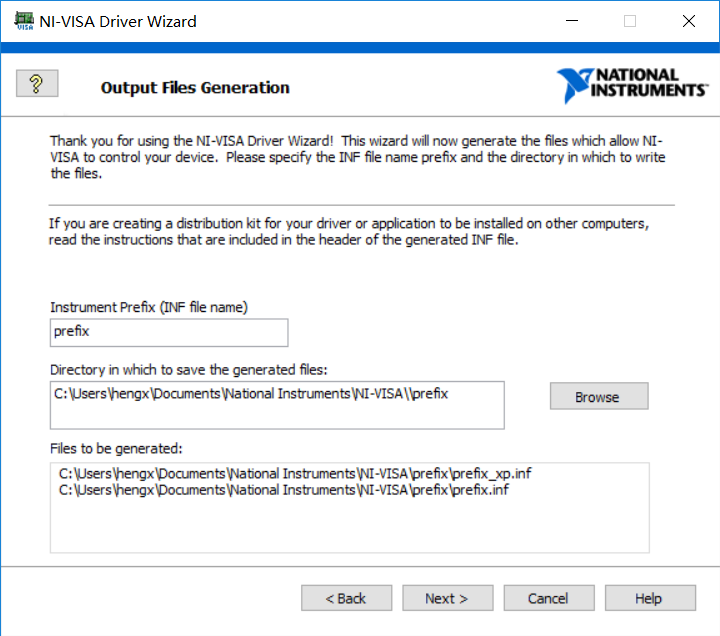
选择USB，点NEXT



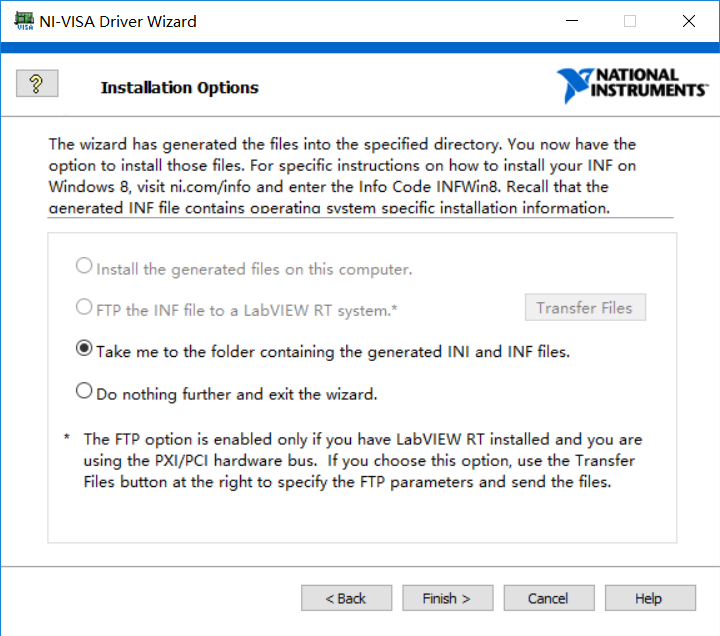
    因为我的USB设备一直在电脑上插着，所以VISA安装向导识别了它，我们直接选中这个设备。这个时候你会发现，都自动填装好了，不用我们自己再填写。 包括Vendor ID、Product ID、Descriptor Information（可以查看各种描述符），Model Name（设备管理器要显示的名字）我们点击NEXT，进行下一步（点完后可能会跳出小窗口，我们一路点Yes，通过）



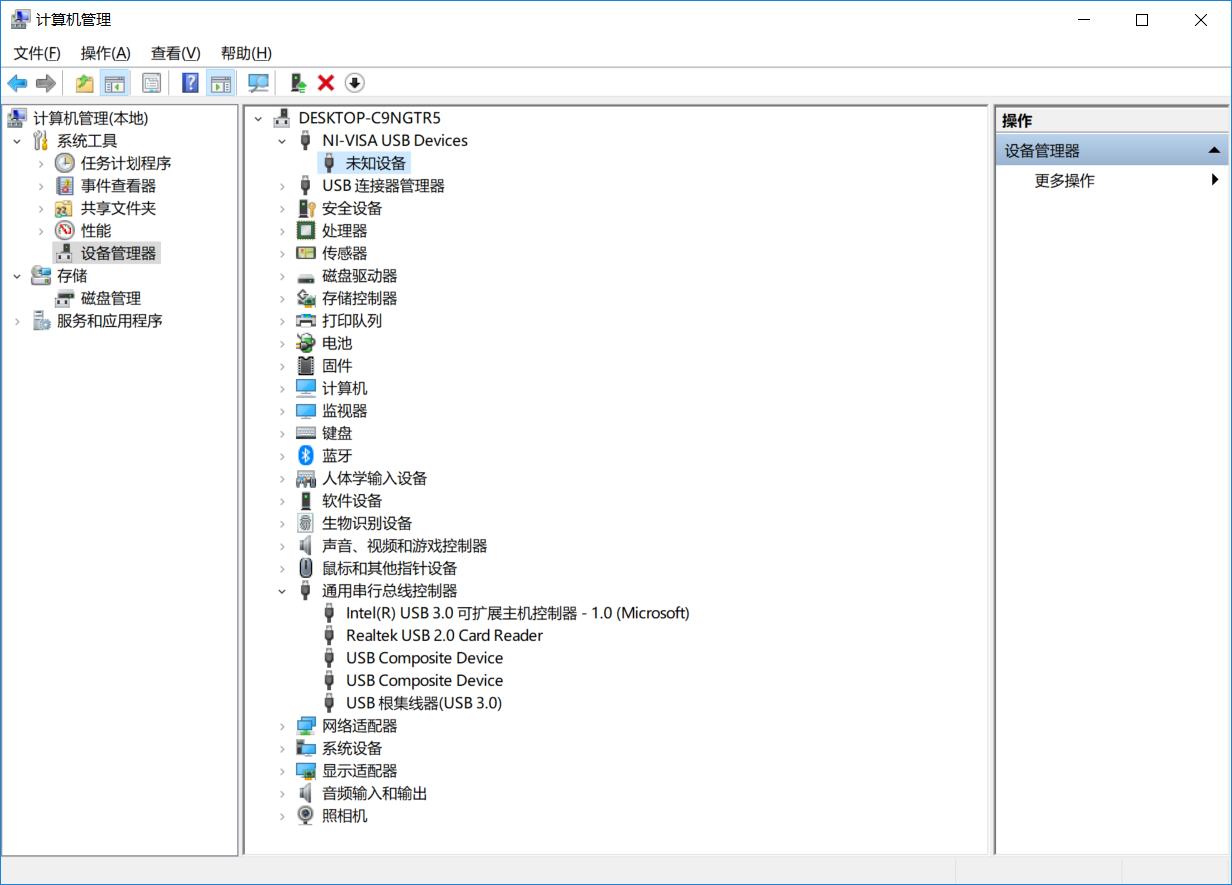
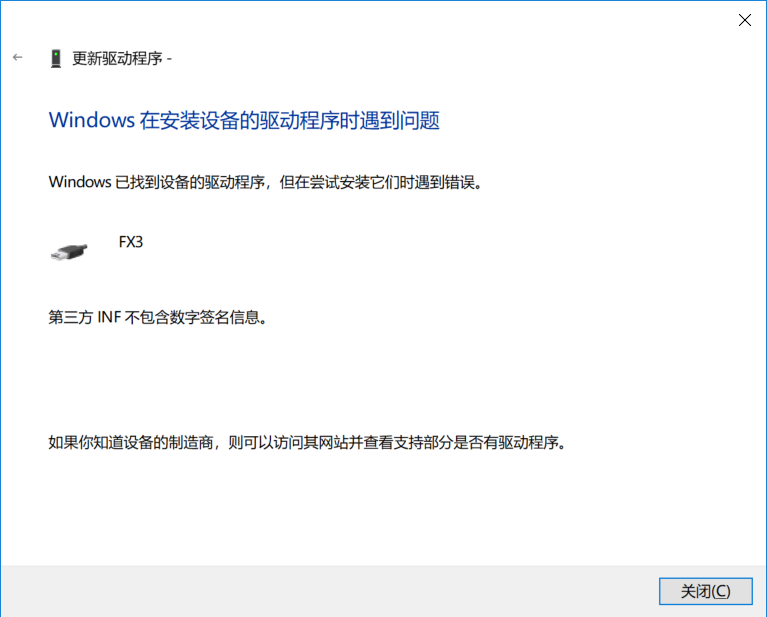
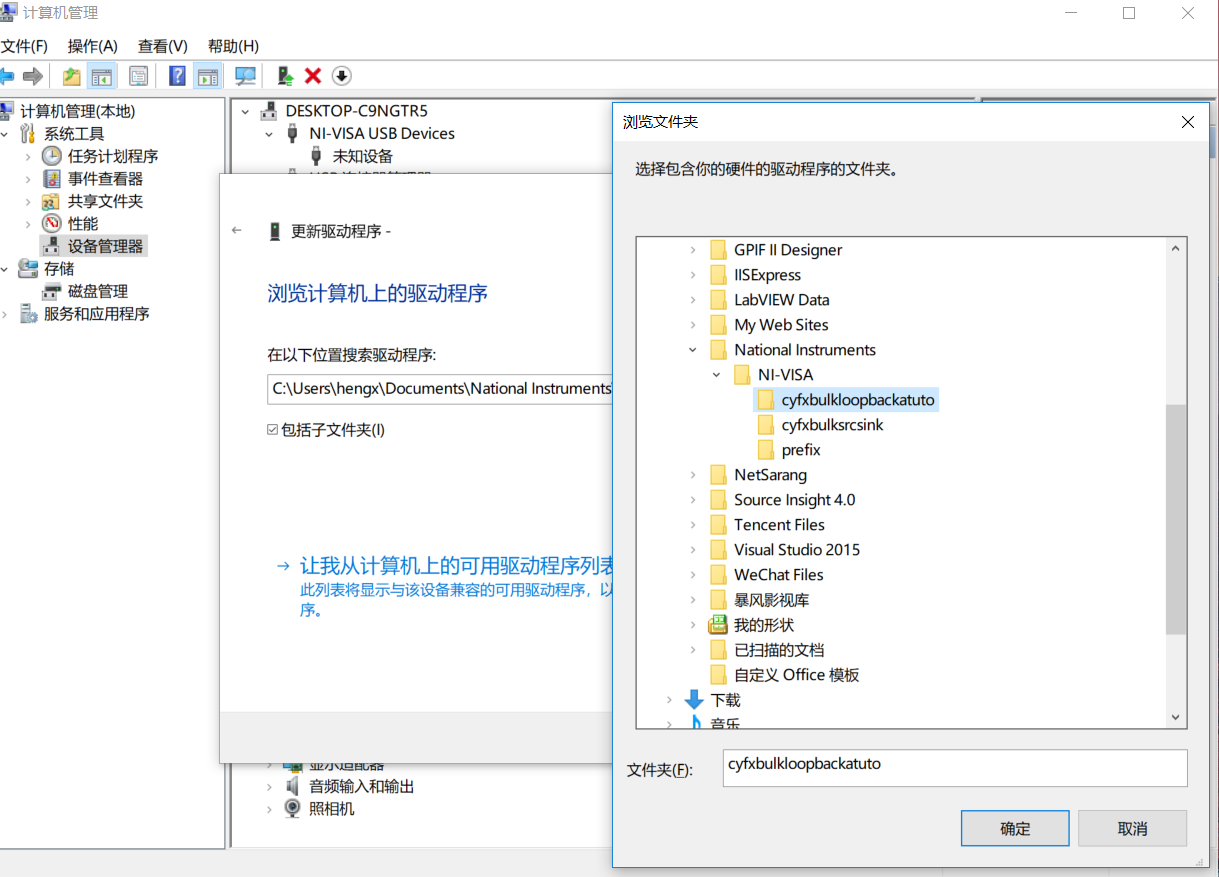
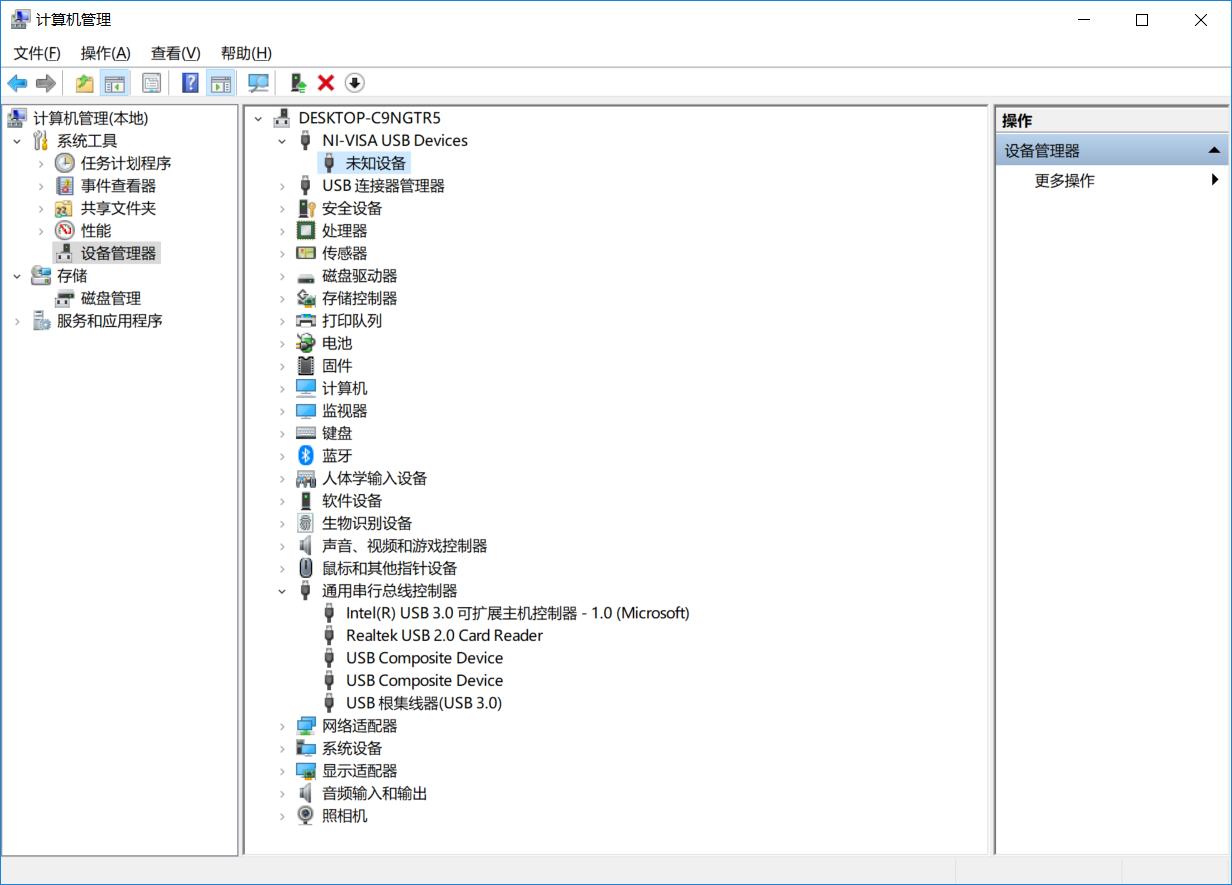
下一步是Output Files Generation，我们可以不做任何修改直接next



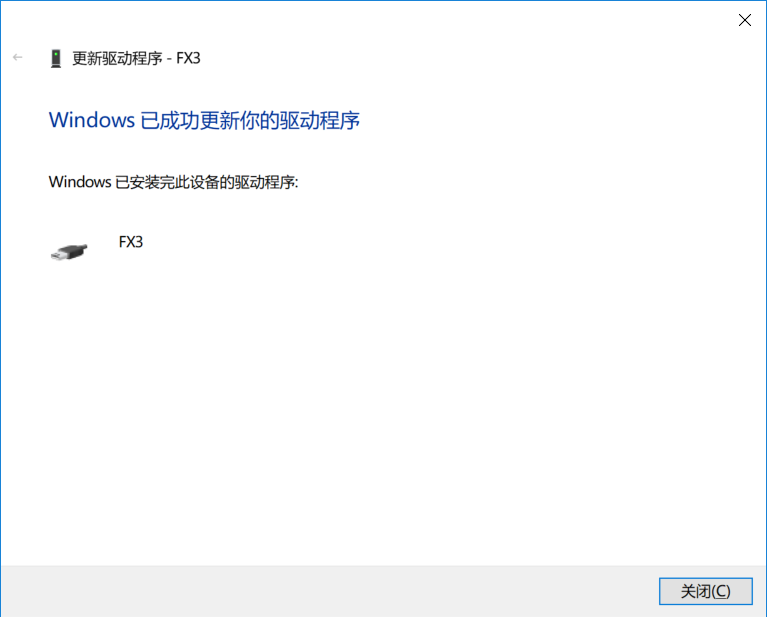
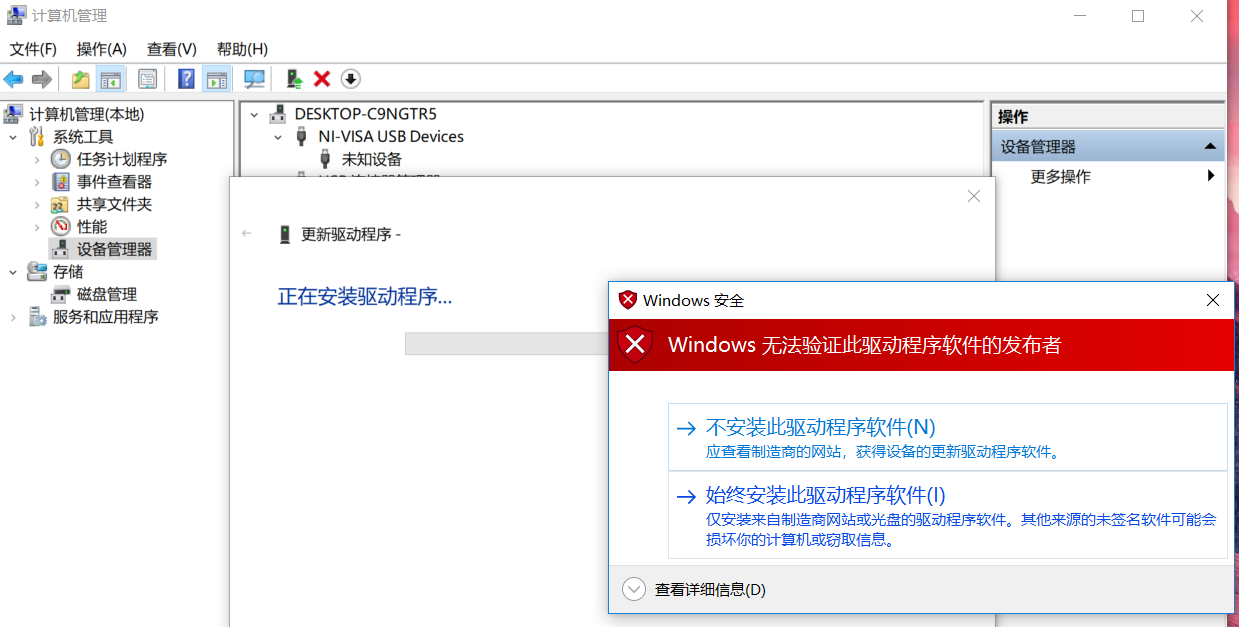
最后一步是Installation Option，我们选第一项安装，然后点Finish。



    然后系统就开始安装驱动啦。我遇到的情况是安装完成，它弹出一个小窗口说安装不成功，我们不用管它直接close就好。然后这个时候我们再去设备管理器中看，已经产生可以被labview识别的驱动了。

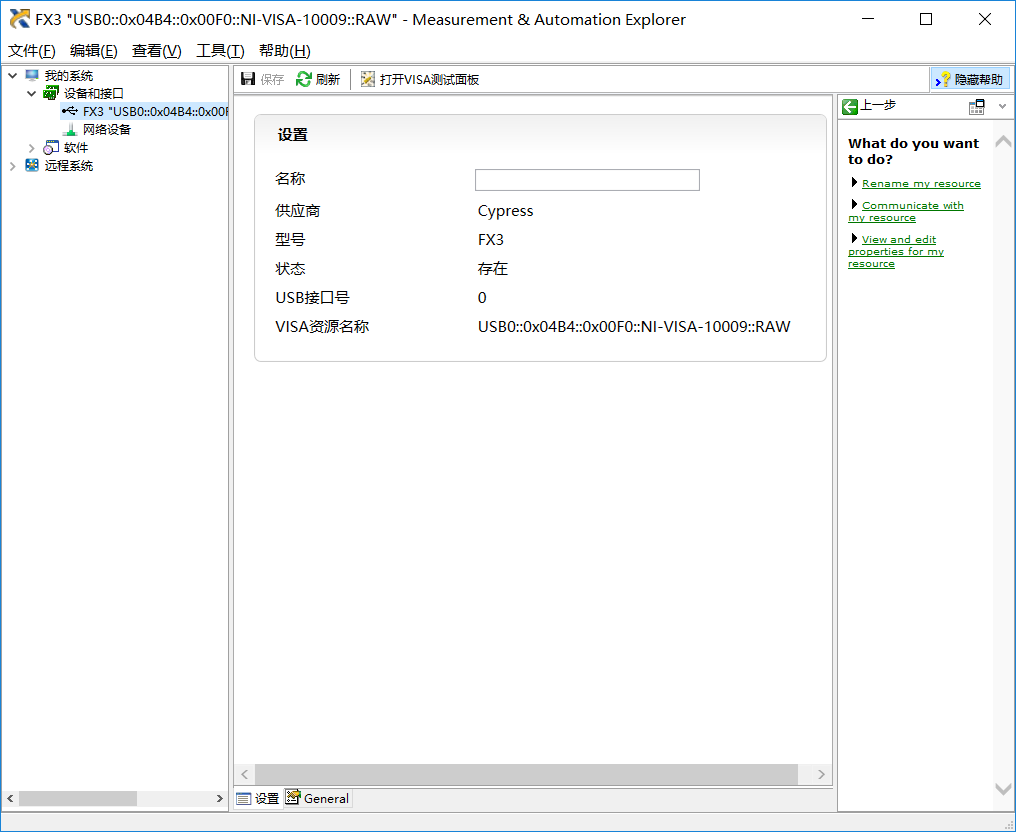
  


给问题处理请参考“Win10怎么禁用驱动程序强制签名”，按照步骤操作完之后，就可以安装USB驱动程序。



    那么如何查看我们的USB能被labview识别了呢。我们打开NI-MAX，在设备和接口处如果能看到我们的USB设备，说明已经被识别了。（我一般是看0x0413::0x5724来识别我的USB口，因为这两个参数是我自己设的。。）

我们选打开VISA测试面板，可以看到一些USB状态，这也证明USB设备的一些信息，继续往下走。



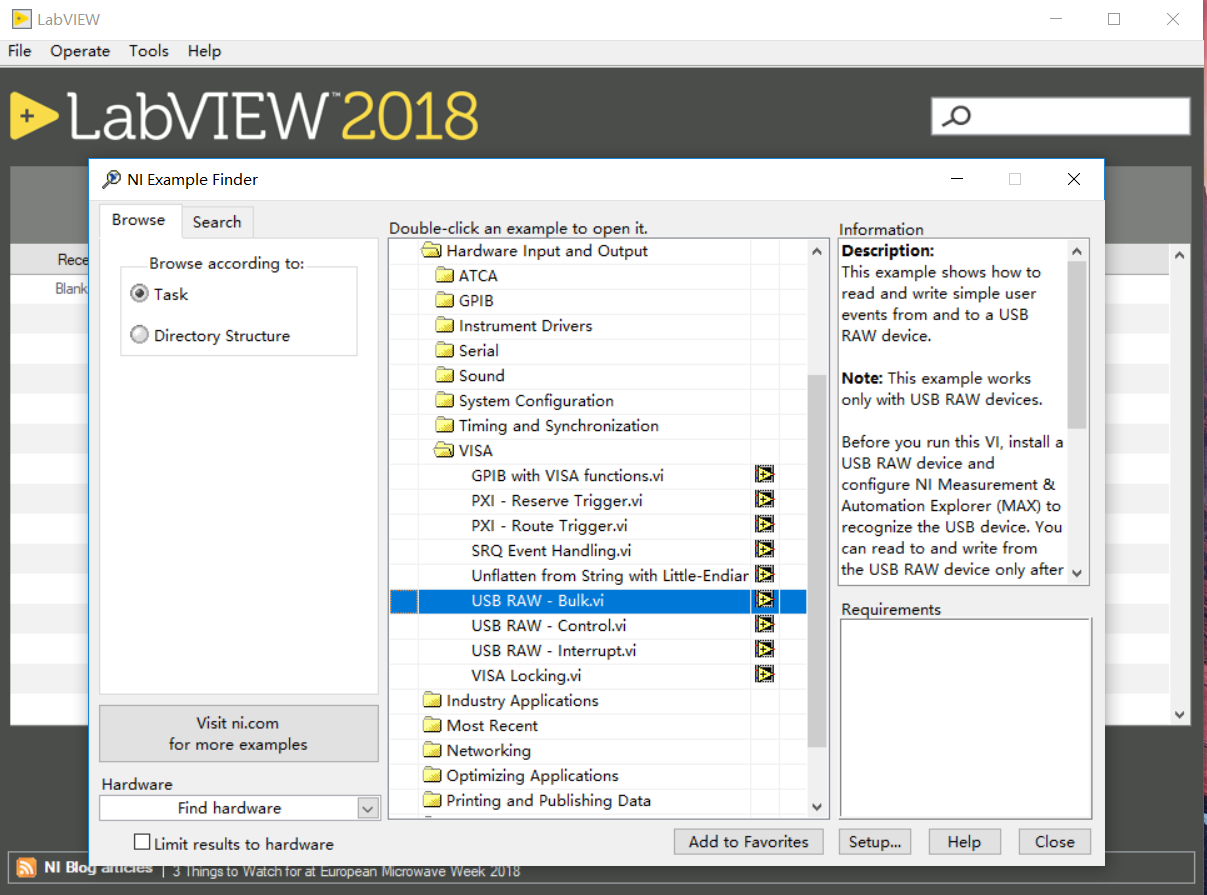
上面我们已经验证了NI可以识别USB口，我们开始编写小助手程序吧.

**二、程序编写**

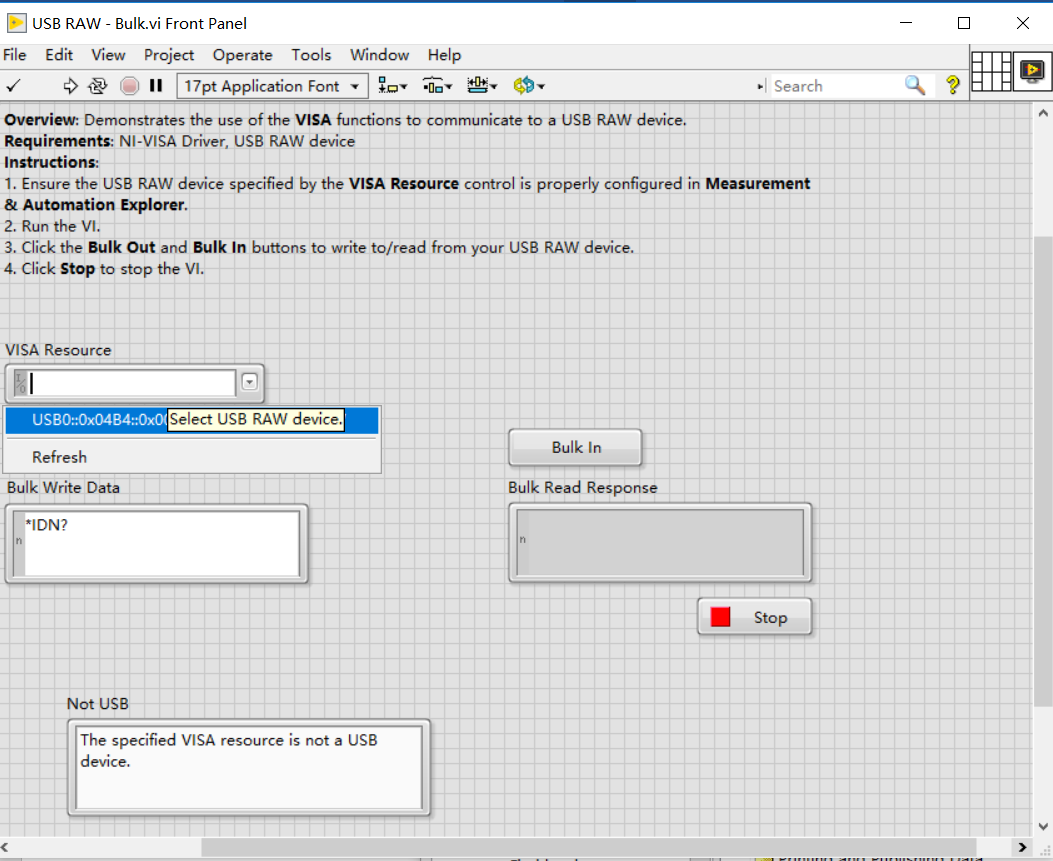
新建VI文件。

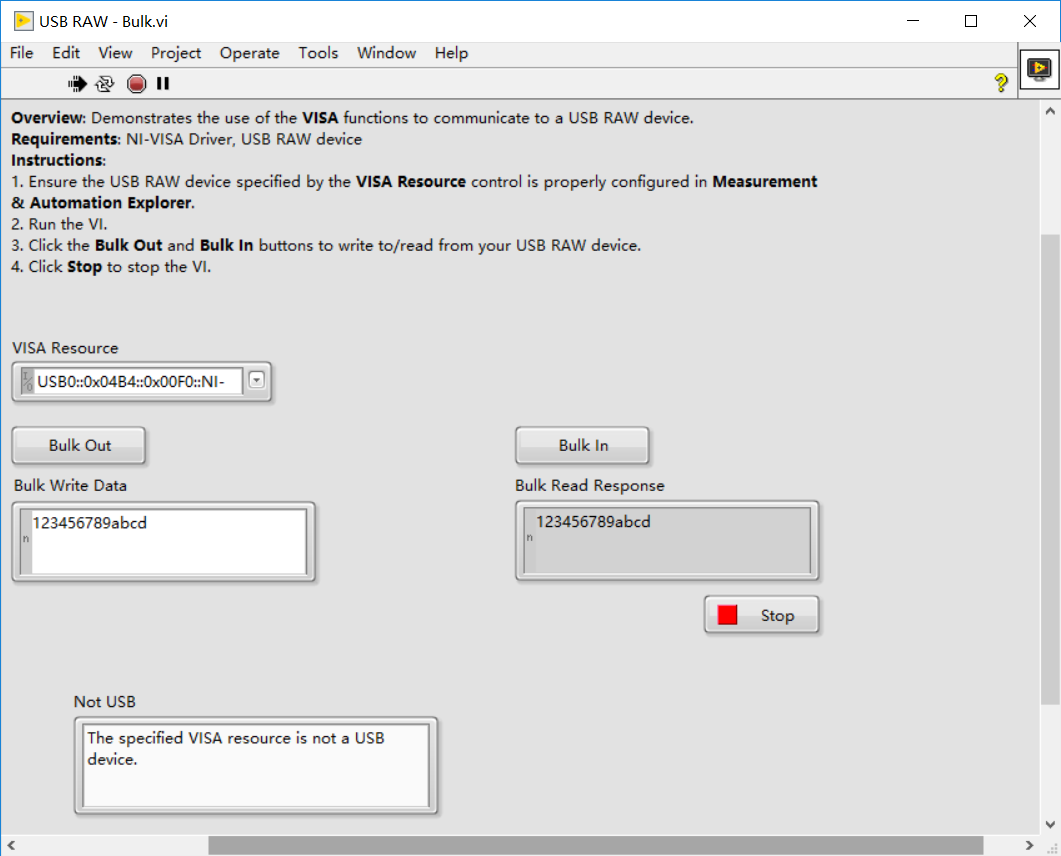
因为使用VISA通信，所以我们想到的肯定是去VISA里找驱动。先把几个常用的拖出来。（它们的都在仪器I/O—>VISA里，具体哪个在什么位置我就不说了）VISA打开、VISA写入、VISA读取、VISA关闭。于是我们连接了这样的结构

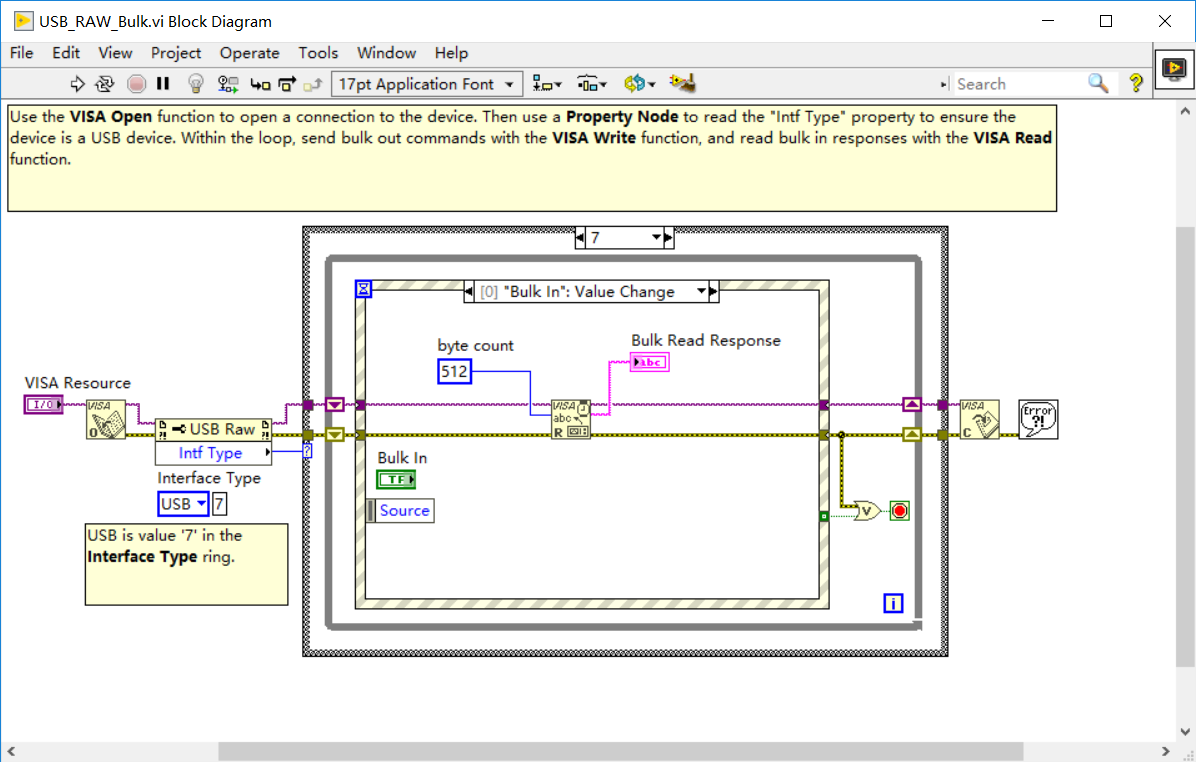
在VI例子中找到USB RAW



结果，我打开VISA资源口发现是COM4，懵逼了呀，上面才说的USB口识别了呀。







    研究一番（实际上是忘记了从哪里查的了）问题出在VISA资源名称，这个控件上，我们需要设备它识别的VISA类为USB。具体设置方式是在控件上右键—>选择VISA—>I/O session-->USB Raw

Ok设置完成我们就能找到USB设备啦！我们运行一下看看效果，结果出来这个鬼东西。

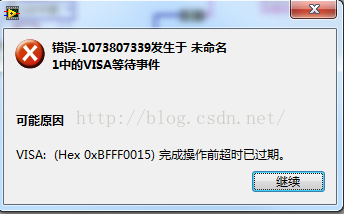
    上网查了一下，网上有大神已经针对这个问题给出了解决办法：  
http://bbs.elecfans.com/jishu\_515959\_1\_1.html  
    我大概说一下大概就是，就是HID类只支持仅支持控制传输和中断传输。需要主机发送命令后，USB设备返回信息，这样才能完成读取。所以我们需要使用中断方式进行数据读取。按照上面给出的解决办法（非常详细）我们做出修改。于是程序又变成了这个样子。

    然后再运行程序，完成啦.配合上一博文写的USBHID通讯的历程，我们通过上位机把字符5通过主机发给设备，然后设备，把收到的字符发给主机，我们也确实读到了字符5！！！到此简单的USB通讯已经可以了但是，我们是要做通讯小助手的人呀，不能就此止步呀。  
    我们继续往下修改，上面的网址中文章提到过”要注意必须在visa写入之后才可以读取usb中断返回的数据。不写入或是写入的数据错误导致无返回值，就会产生超时错误 VISA:  (Hex 0xBFFF0015) 完成操作前超时已过期。”  
    经过实测，确实是这样样子。于是我们通过一个条件结构来解决，当需要发送数据时，点发送按键，条件为真，labview将发送写入缓冲区的数据，当不需要发送时，条件为假，系统送0字节

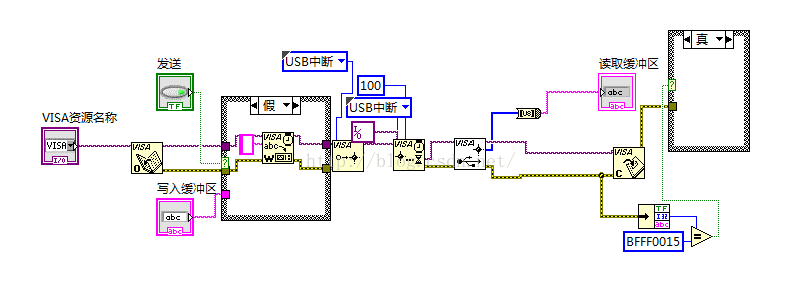
我们点击运行看一下效果。结果系统又报错了。

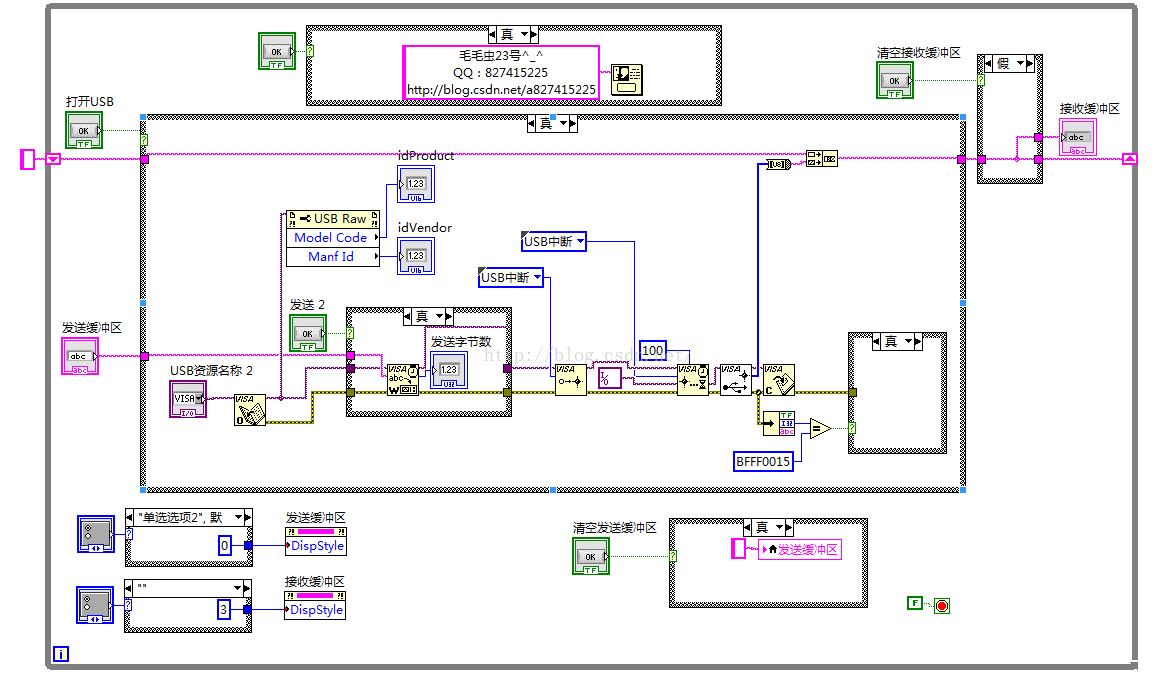
    说系统完成操作前超时，问题出在VISA等待事件，这是为什么呢？原来我们设置启用中断传输时间，结果这个中断传输一直没发生，没有数据通过中断传输读取过来。所以系统报错了。但是仔细想想，USB设备想发数据就发数据，没有要发送的数据，你偏要它发点有效数据，也是无理的要求呀（USB设备内心：你自己报错，怪我咯）怎么办呢？既然我们理解了超时的原因，他对于我们USB通信也无大碍，那我们就忽略它好了！具体的操作方式是：

    获取错报代码（0xbfff0015）然后当发现是超时报错时我们不处理它，不用弹出报错窗口



所以后来程序又变成了这样



这个时候程序已经可以正常运行了，主体部分完成，下面再介绍几个细枝末节的部分  
1、 使用移位寄存器实现读取缓冲区的的不覆盖显示。  
2、 使用条件结构控制USB口的打开与关闭  
3、 调用发送缓冲区与接收缓冲区的显示样式的属性节点控制属性  
最后的最后程序是这样的  


前面板是这样的



至此USB通信小助手完成

参考文献：

|  |  |  |
| --- | --- | --- |
| 1 | Win10怎么禁用驱动程序强制签名 | <https://jingyan.baidu.com/article/624e74594dbc8d34e8ba5aa6.html> |
| 2 | 使用LabVIEW开发STM32 USB HID驱动和上位机软件成功，有图有真相 | <http://bbs.elecfans.com/jishu_454527_1_1.html> |
| 3 | Labview 与USB设备（HID）通过visa通信。VISA: (Hex 0xBFFF003A) 由于设置无效（导致属性被设置为不一致状态） | <http://bbs.elecfans.com/jishu_515959_1_1.html> |
| 4 | 基于labview的USB调试小助手 | <https://blog.csdn.net/a827415225/article/details/51745789> |
| 5 |  |  |