### 一、PC机程序与USB设备通信过程介绍

PC端C#程序与USB设备通信过程如下图所示：



PC端C#程序在设计时使用到了LibUsbDotNet类库（在07-Lib文件夹下的LibUsbDotNet.dll中），里面包括许多USB操作的类和一些事件，在具体程序设计时用到的类有UsbRegistry、UsbDevice、UsbEndpointReader、UsbEndpointWriter等，事件有IDeviceNotifier、DataReceived等。PC端程序与USB程序交互步骤如下描述：

（1）PC端获取目前系统中全部USB设备信息，通过比较VID、PID以及SerialString判断是否为本程序可操作目标设备，如是则加入下拉列表中；

（2）根据USB设备的UsbRegistry信息，使用的Open方法创建USB对象，打开设备，并设置读写端点reader，writer；

（3）添加OnDeviceNotify、USBDataRecieved事件处理过程，在程序运行期间如USB设备有变化，会触发OnDeviceNotify事件，如收到USB数据触发USBDataRecieved事件，对应事件触发后后执行绑定的函数；

（4）PC端程序通过writer端点向USB设备发送数据，USB设备产生中断接收数据，在main函数中进行处理，如是块数据则存入Flash中，如是取数据指令0XAA，则将数据放入USB发送缓冲区；

（5）PC端驱动不断向USB请求数据，如此时USB设备发送缓冲区不空，则将数据发送至PC端，此时PC收到数据放入缓冲区，PC端C#程序产生USB数据接收中断USBDataRecieved，从缓冲区将数据读出并显示。

如USB设备在PC端程序运行前已经上电且与PC端USB驱动程序枚举完成，则在PC程序运行之初会将设备信息加入列表中，并显示相关信息，如果USB设备在PC端程序运行后才上电，则会触发PC端的OnDeviceNotify事件，重新加载设备列表，并打开USB设备。

### 二、发送及接收数据格式封装

帧共两部分组成：命令（1字节）+数据（<=31字节），数据长度不超过31字节，超过31字节的数据在拆分发送时需自行组帧，并重新定义帧格式。

从底层返回的数据要求返回时附带命令，表示这个数据是由该命令请求得到的。

|  |  |
| --- | --- |
| 命令 | 数据区 |
| 1字节 | <=31字节 |

### 三、程序修改指南

（1）如需增加帧类型，需在PublicProperty类中的命令枚举中增加一个新的命令。并**注意同时修改底端MCU程序，否则可能会无法完成所需功能**。

（2）本程序在区分USB设备时使用SerialString字段判别（由于VID和PID在程序中无需修改，在此处就不作为判断依据了），因此**如在实际项目中加入了某个设备，将其SerialString添加至USBTest类中mySerialStringArray数组里即可正常识别**，在系统中不同的USB设备的SerialString属性可以相同。

（3）如需对点击发送按钮时完成的功能修改，请修改btnSend\_Click函数部分代码。

（4）如需对点击接收按钮式完成的功能修改，请修改btnRecv\_Click函数部分代码。

（5）如需修改接收到数据后完成的功能，例如对新添加的命令进行解析等，请修改OnRxEndPointData函数。

### 四、注意事项

**程序中的VID和PID不能修改，否则将不能识别目标设备。**