

---

# USB2UIS 用户手册

版本： 6.0

发布： 7/15/2016

OneasyB

---

## 目录

USB2UIS 用户手册 .....	1
一、 简介 .....	3
二、 硬件系统 .....	4
三、 附带文件说明 .....	9
四、 驱动程序安装 .....	10
4.1> 安装 USB2ish 驱动.....	10
4.2> 安装 USB 转 UART 驱动程序.....	19
五、 使用简介 .....	20
5.1> UART 口使用.....	20
5.1.1> window 下使用: .....	20
5.1.2> linux 下使用: .....	21
5.2> I2c 使用.....	21
5.2.1> 选择设备 .....	23
5.2.2> 设置 .....	24
5.2.3> 操作选择 .....	25
5.2.4> 读写操作 .....	25
A. 立即读模式.....	26
B. 复合读模式.....	27
C. 写模式.....	28
5.2.5> 特殊功能(分段写) .....	30
5.3> SPI 使用.....	30
5.3.1> 作为主设备使用.....	31
A. 设置.....	31
B. 自检.....	32
C. 操作选择.....	34
D. 读写操作.....	34
E. 特殊功能.....	38
F. SPI flash 读写 .....	38
5.3.2> 作为从设备使用 (此功能只有扩展型才有) .....	38
5.4> GPIO 使用 (此功能只有扩展型才有) .....	39
5.4.1> 数字 IO 和 ADC 采样端接口定义 .....	41
5.4.2> ADC 使用说明 .....	41
5.4.2> 数字 IO 的使用 .....	42
5.4.3> PWM 使用.....	42
5.5> 文件操作.....	43
5.6> 弹出菜单.....	44
5.7> 固件更新.....	45
六、 LINUX 下使用 I2c,SPI 和 PWM 功能.....	47
七、 常见问题及解决方法 .....	47
八、 维护 .....	49

---

## 一、 简介

USB2UIS 是一款实用方便的多功能转换板，它集成了 USB 转 UART, USB 转 I2C, USB 转 SPI 接口，扩展型还附有 USB 转 ADC, USB 转 PWM, USB 转 GPIO 接口。通过上位机软件很方便地同带有 UART, I2C, SPI 等接口的设备通讯。

### 系统特征:

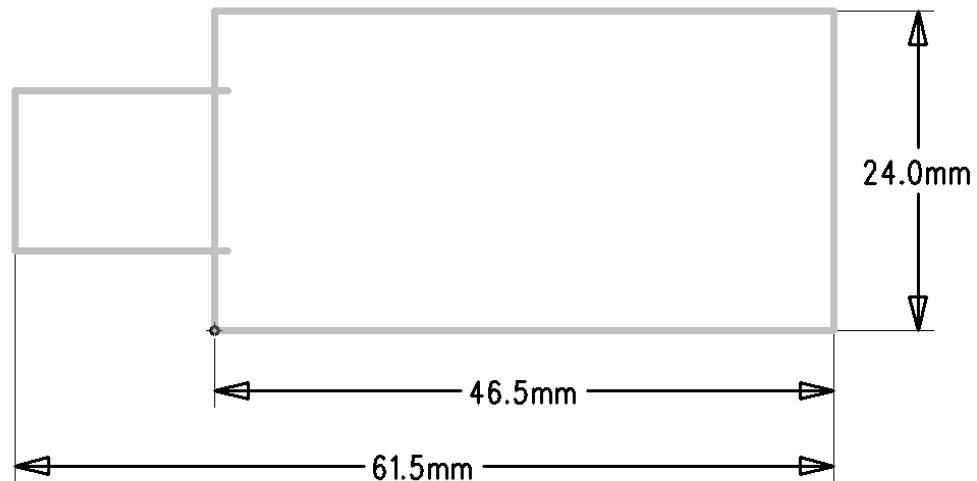
- 1> 可选的 3.3V 和 5V 输出电压。
- 2> 自恢复熔丝，防止过流。
- 3> USB2.0 全速通讯速率。
- 4> 支持 I2c 设备地址自动侦测和 SPI 自检测试。
- 5> 提供 DLL 接口供第三方开发,附带 VB, VC, labview, QT, c++builder 参考程序。
- 6> 唯一的设备序列号，可同时打开多个设备。
- 7> 支持 24,25 序列器件的读写。
- 8> 支持 Hex 扩展名文件的解码，以及文本和 16 进制混合编辑模式。
- 9> 支持 win XP 以上系统(32bit,64bit)，支持 linux 系统。
- 10> 可通过电脑 USB 线更新固件，软件升级，解决维护烦恼,对于特殊应用，还可定制。

### 电气特征:

- 1> 所有的数字端口电压高电平在 2.7~3.6V，低电平在-0.3~0.3V。
- 2> 所有的数字端口最大驱动电流 10MA。
- 3> **UART 波特率最大可达 230400。**
- 4> I2C 有五档频率设置，分别是 100K, 200K, 300K, 400K,800K。
- 5> SPI 波特率最大可设置 12M，有九档设置，分别是 200K,400K,600K,800K,1M,2M,4M,6M,12M。扩展型还支持带中断的读写，可工作在从模式。
- 6> 扩展型带 10bit ADC 采样，四个通道可以配置为单端和差分输入。
- 7> 扩展型带 8 个数字 IO 可任意配置的输入和输出，还有一个扩展 IO 输出和一个中断 IO 输入。
- 8> 扩展型带四路 PWM 同频输出，频率最大可达 100K 占空比任意可调。

## 二、硬件系统

1> 外形尺寸:



2> 外貌，两种类型(基本型和扩展型)

图左是基本型转接板，2.54mm 的间距的排针，图右扩展型转接板，是 2.0mm 间距的排



基本型

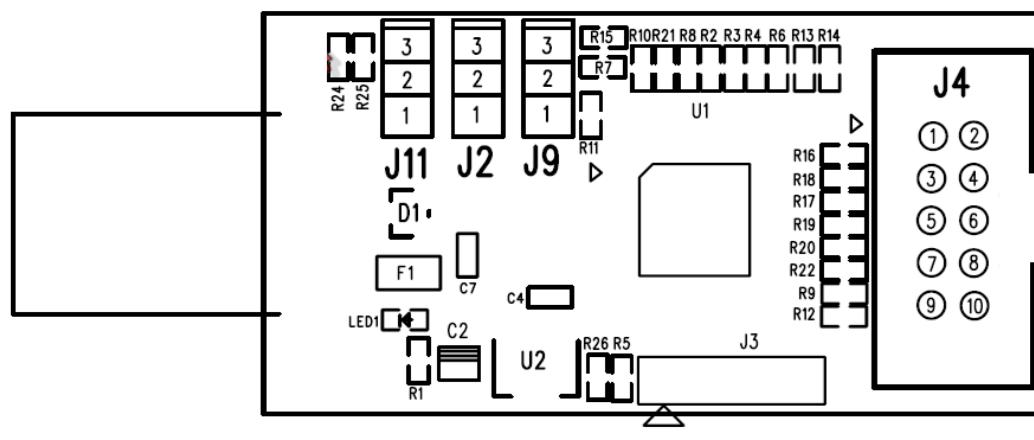


扩展型

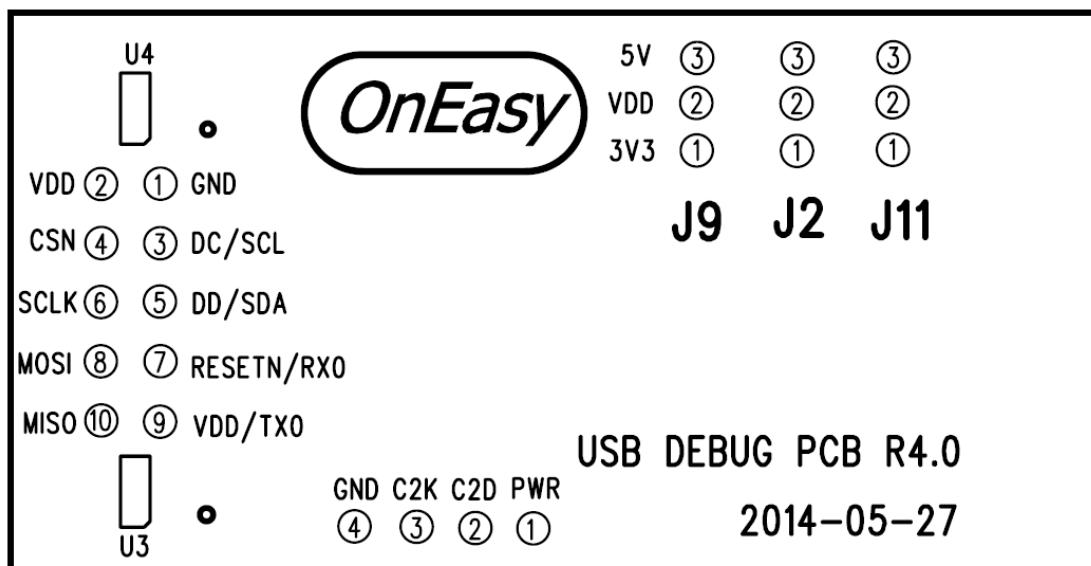
3> 接口布局位置及端口定义说明:

接口布局如下图:

1>基本型



基本型正面丝印图

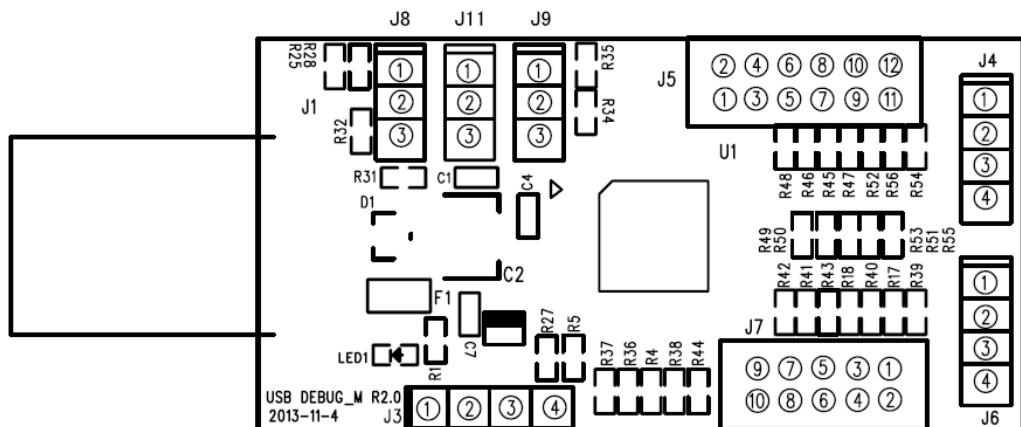


基本型背面丝印图

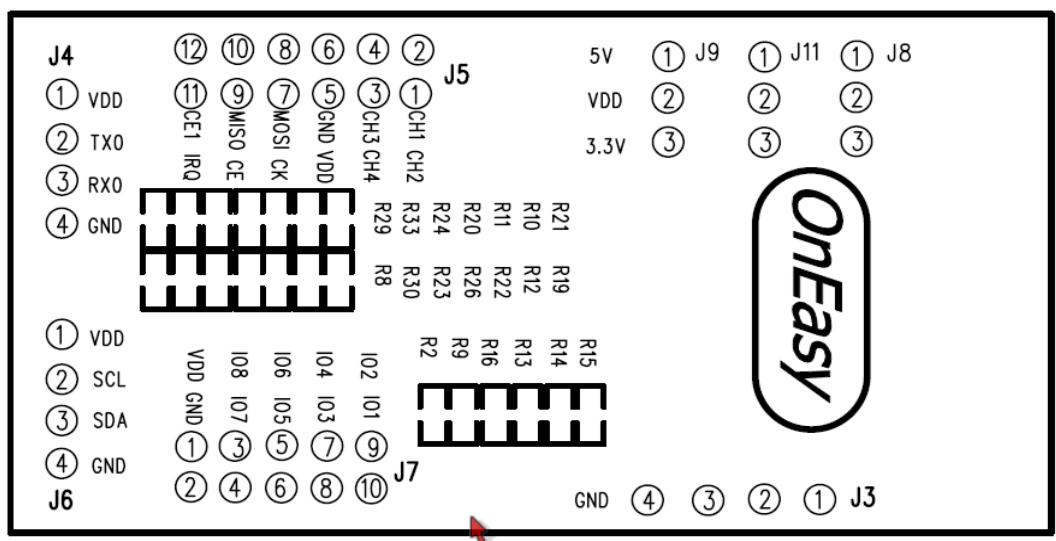
基本型端口说明见下表:

接口编号	用途	备注
J4	供电	J4-1 GND J4-2 VDD(输出)
	I2C	J4-3 SCL J4-5 SDA
	SPI	J4-4 CSN J4-6 CLK J4-8 MOSI J4-10 MISO
	UART	J4-7 RX J4-9 TX
J2	功能选择跳线	1,2 短接, UART 功能允许; 2,3 短接, I2C, SPI 功能允许
J9	电压选择跳线	1,2 短接, 输出 VDD 为 3.3V; 2,3 短接, 输出 VDD 为 5V
J11	升级选择跳线	1,2 短接, 固件升级模式; 2,3 短接, 由 J2 决定功能

2>扩展型



扩展型正面丝印图



扩展型背面丝印图

扩展型端口说明见下表：

接口 编号	功能	引脚定义					
		引脚定义	丝印	软件功能	引脚定义	丝印	软件功能
J1	USB						
J3	未用						
J4	UART	J4-1	VDD	电源正	J4-2	TXD	发送端
		J4-3	RXD	接收端	J4-4	GND	电源地
J5	PWM	J5-1	CH2	二号通道	J5-2	CH1	一号通道
		J5-3	CH3	四号通道	J5-4	CH4	三号通道
	SPI	J5-5	VDD	电源正	J5-6	GND	电源地
		J5-7	CK	时钟端	J5-8	MOSI	主出从入端
		J5-9	CE	片选端	J5-10	MISO	主入从出端
		J5-11	IRQ	IO 中断端	J5-12	CE1	扩展片选
J6	I2C	J6-1	VDD	电源正	J6-2	SCL	时钟端
		J6-3	SDA	数据端	J6-4	GND	电源地
J7	GPIO	J7-1	VDD	电源正	J7-2	GND	电源地
		J7-3	IO8	Bit7	J7-4	IO7	Bit6
		J7-5	IO6	Bit5	J7-6	IO5	Bit4
		J7-7	IO4	Bit3	J7-8	IO3	Bit2
		J7-9	IO2	Bit1	J7-10	IO1	Bit0
J8	功能选择 跳线	1,2 短接, UART 功能允许; 2,3 短接, IIC, SPI, GPIO 功能允许					
J9	电压选择 选择	1,2 短接, 输出 VDD 为 5V; 2,3 短接, 输出 VDD 为 3.3V					
J11	升级选择 选择	1,2 短接, 固件升级模式; 2,3 短接, 由 J8 决定功能					

### 三、附带文件说明

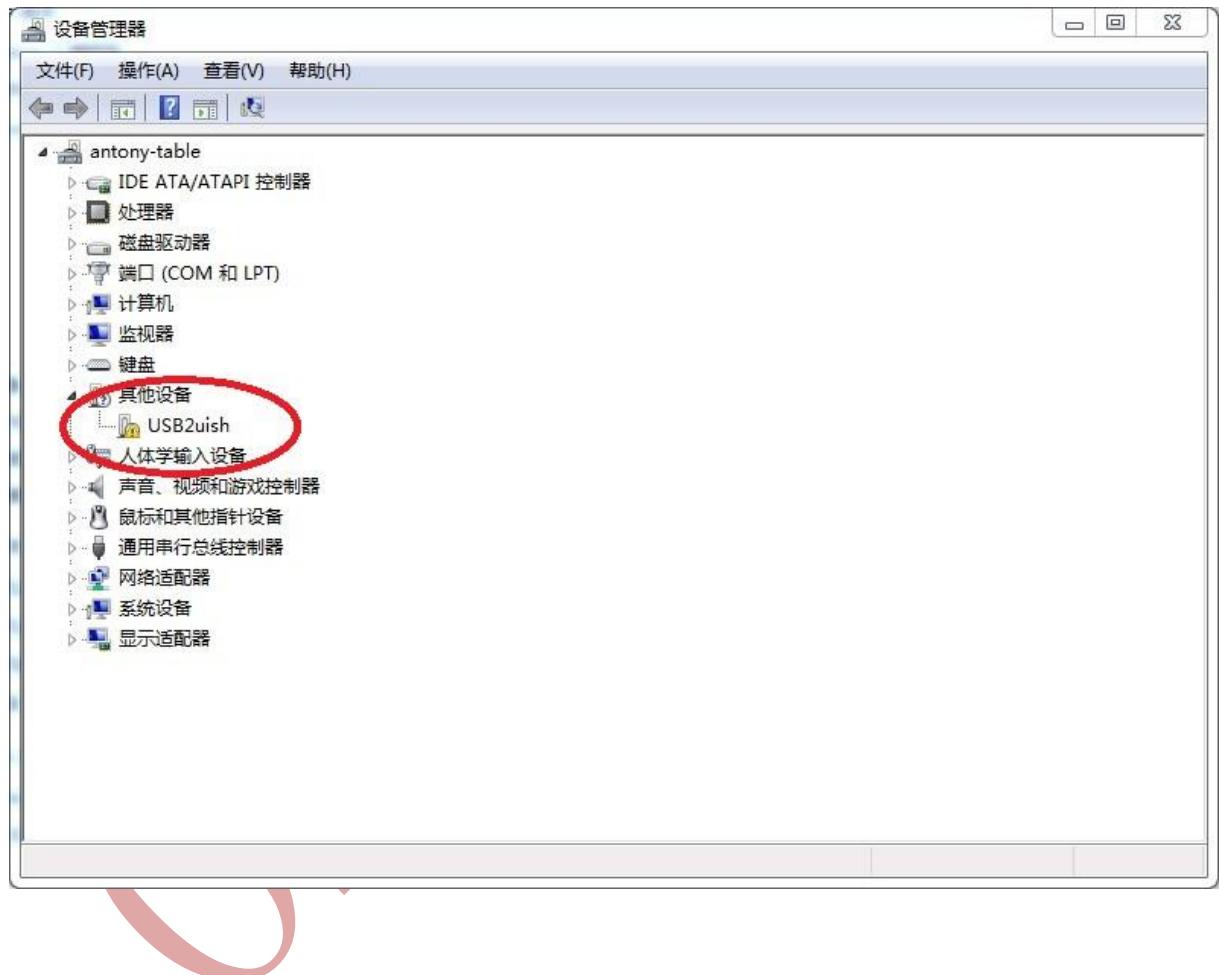
文件夹	内容	适用系统	用 途
APP	Usb2ish_pro.exe	Window (32bit) 系统	Usb 转接板上位机软件(中文)
	Usb2ish_pro_en.exe	Window (32bit) 系统	Usb 转接板上位机软件(英文)
	HybridEdit.ocx, Usb2ish.dll	Window (32bit) 系统	上位机软件运行时的库文件
LIB	libUSB2UIS.so	linux	Linux 下的库文件
	usb2uis.dll, usb2uis.lib	Window(32bit/64bit)	Window 下的库文件
DOC	Usb2uis 转接板用户手册.pdf	不限	操作说明书(中文)
	USB2UIS user manul.pdf	不限	操作说明书(英文)
	Lib user manual.pdf	不限	库函数使用说明(英文)
DRIVER	Ish	Window 系统	Ish 是 usb2ish 设备驱动程序
	Uart	Window 系统	Usb2uart 设备驱动程序
FIRMWARE	USB2ISH_FM_BXX.bin	不限	基本型板子固件升级目标文件
	USB2ISH_FM_MXX.bin	不限	扩展型板子固件升级目标文件
DEMO	cbc2009	Window xp 系统	C++builder 下的参考代码，针对 nRF2401 的读写参考程序
	Labview2012	Window 系统	Labivew 下的参考代码，含 i2c, spi, gpio 的读写控制
	Qt-creator301	Ubuntu14LTS 系统	linux 下的 Qt 参考代码，含 i2c, spi, gpio, pwm 的读写控制
	VS2010/vc	Window 系统	VS2010 下的 VC 参考代码，含 i2c, spi, gpio 的读写控制
	Vs2010/vb	Window 系统	VS2010 下的 VB 参考代码，含 SPI 和 I2C 的读写控制
OTHER	uartassist.rar	Window 系统	USB 转 UART 口测试程序

## 四、 驱动程序安装

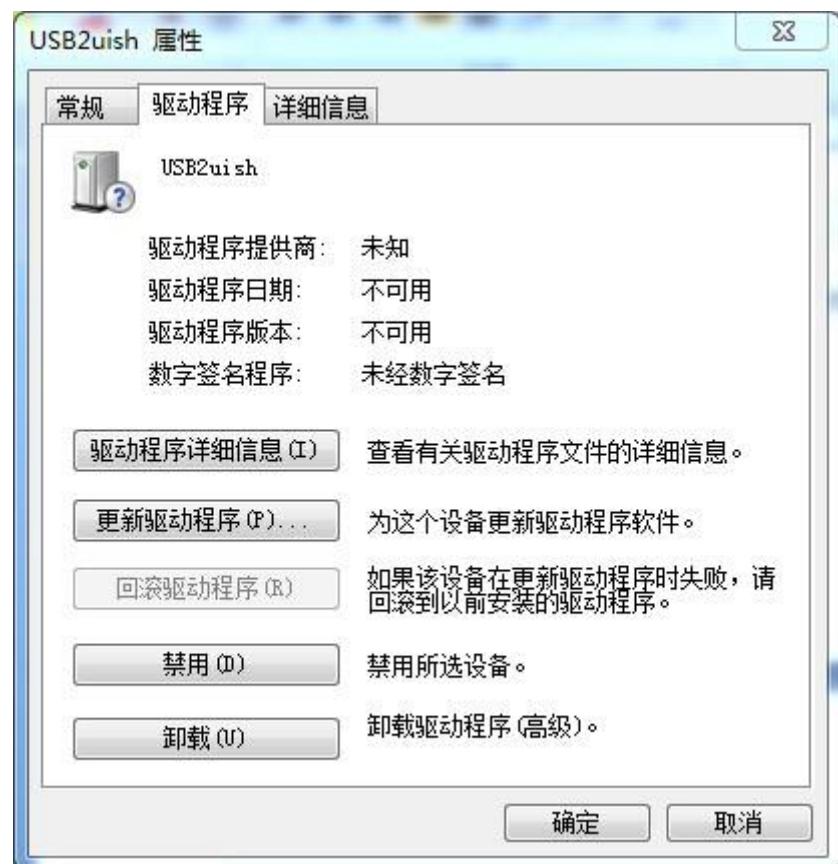
下面讲述扩展型的转接板在 window7 上的驱动程序的安装步骤

### 4.1> 安装 USB2ish 驱动

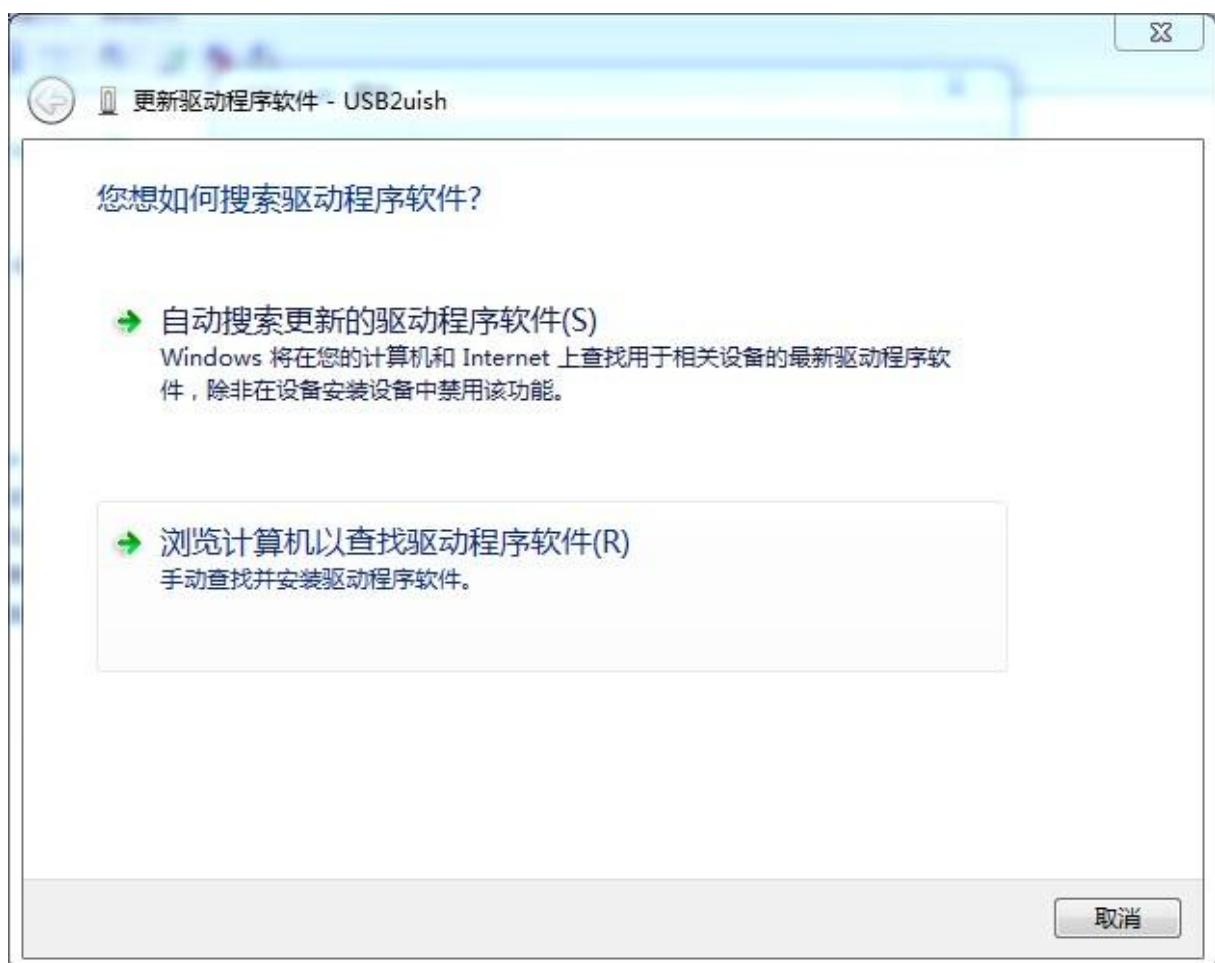
A>确保功能选择跳线和升级选择跳线都为 2-3 短接后，插入 USB 转接板，跳过 window7 的驱动程序自动搜索，打开设备管理器，将会看到



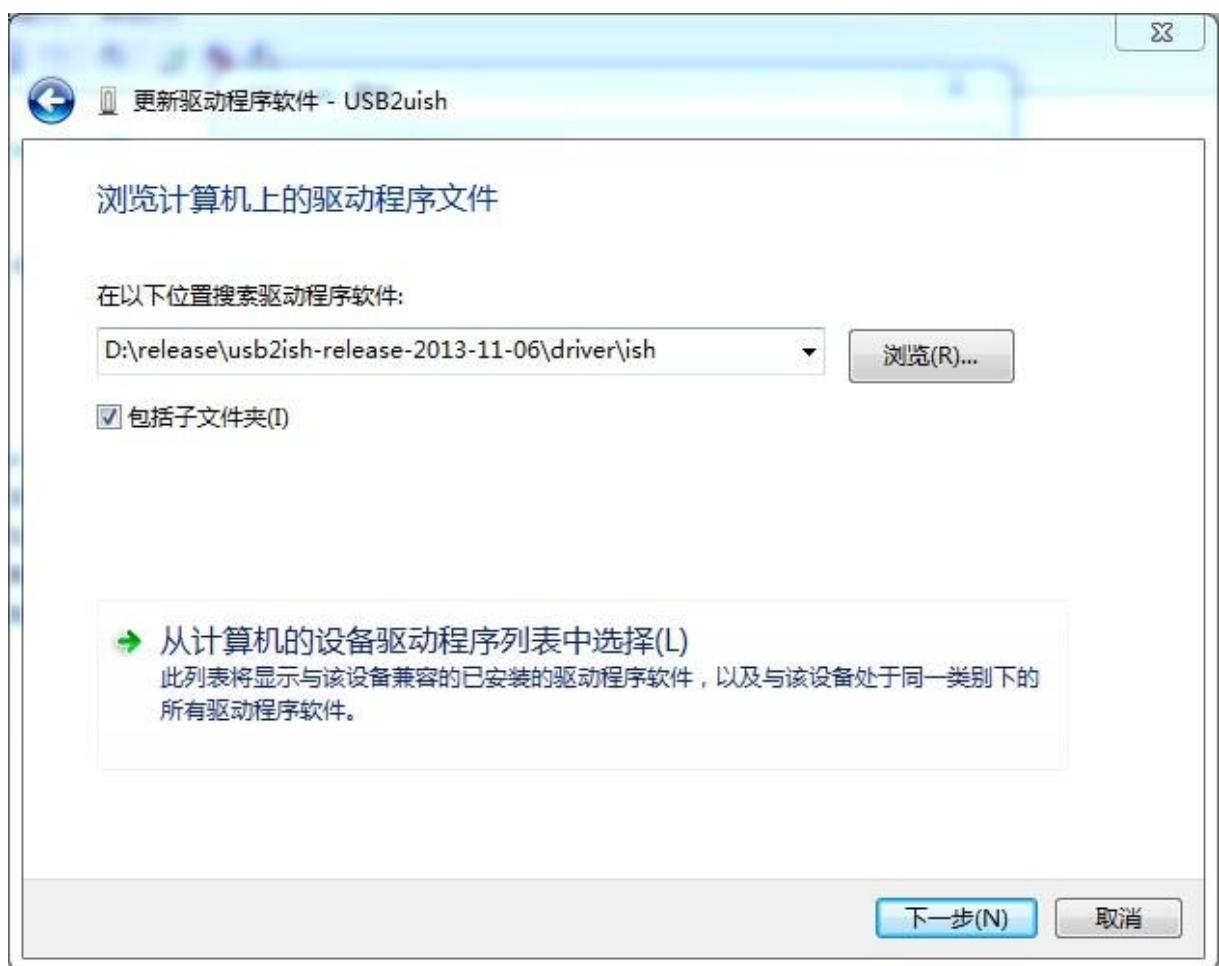
B. 双击黄色带感叹号的设备 USB2uishi, 弹出 USB2uishi 设备属性对话框, 如下图



C>选择“更新驱动程序(P)...”,弹出更新驱动程序对话框, 如下图:



d>选择“浏览计算机以查找驱动程序软件(R)”后，点击下一步，进入下图：



---

E>点击“浏览 (R)”选择驱动程序文件下载的 DRIVER\ish\(window 平台)\usb2ish.inf 所在的文件夹，点击下一步，开始加载驱动程序，如下图：



Only



One

F>加载完成后，选择关闭，回到 USB2ish 设备属性驱动程序标签页，可以看到设备驱动安装信息和公司数字签名的信息，如下图：



G>在 USB2ish 设备属性详细信息标签页，可以看到设备唯一的序列标识号，如下图：

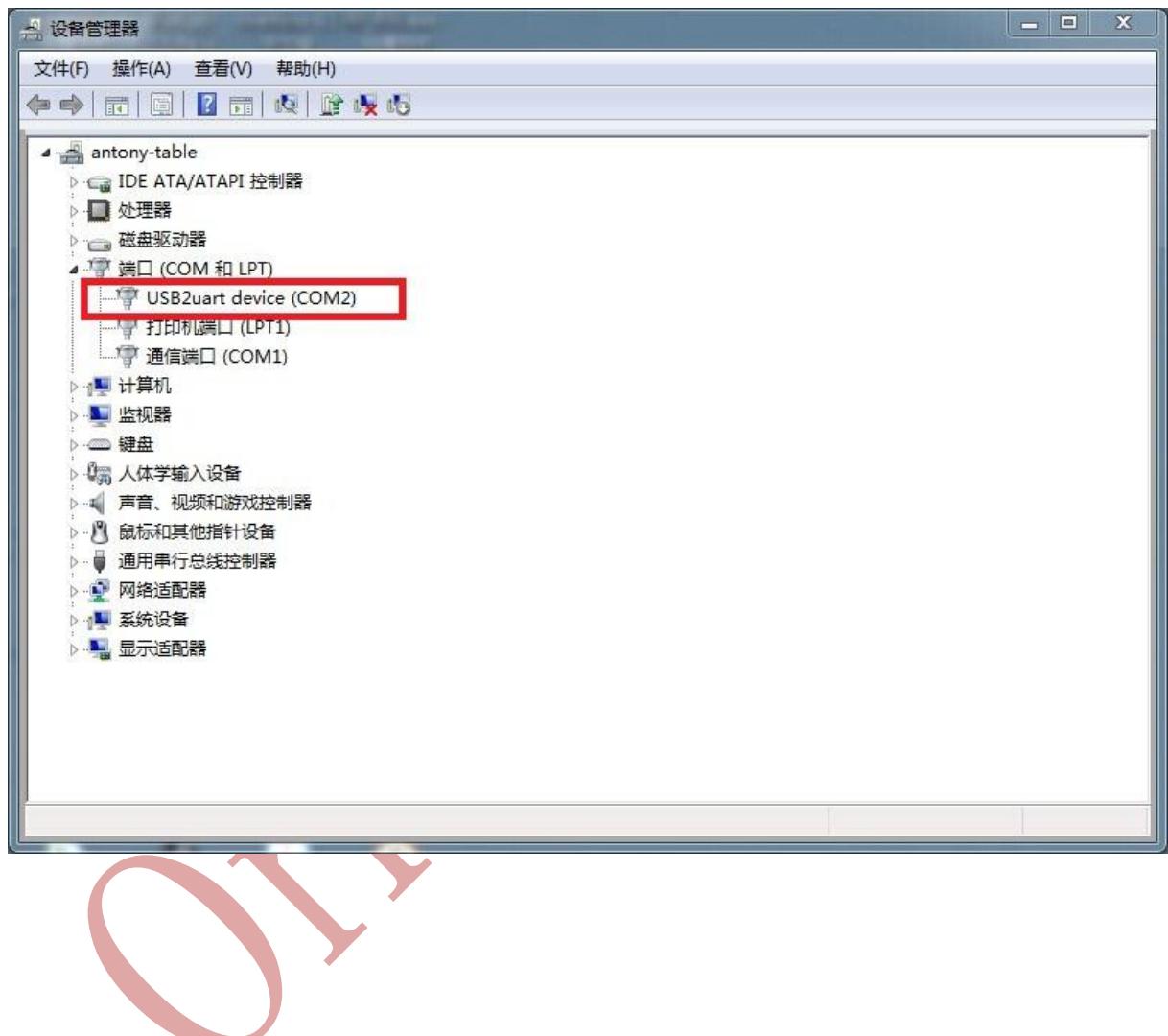


H>回到设备管理器,展开通用串行总线控制器前面的三角符号,可以看到在下面多了一个新的设备,这样证明驱动 程序安装完毕。如下图:



## 4.2> 安装 USB 转 UART 驱动程序

步骤同 1>中的步骤 ABCDE(功能选择跳线要改为 1-2), 只不过在步骤 D 中要选择的驱动程序文件是 DRIVER\uart\(window 平台)\usb2uart.inf 所在的文件夹。安装完成后, 打开系统属性->硬件->设备管理器, 展开端口(COM 和 LPT)的三角符号, 可以看到在下面多了一个新的端口, 如下图:



## 五、 使用简介

### 5.1> UART 口使用

确保功能选择跳线是 1-2 短接，升级选择跳线 2-3 短接。

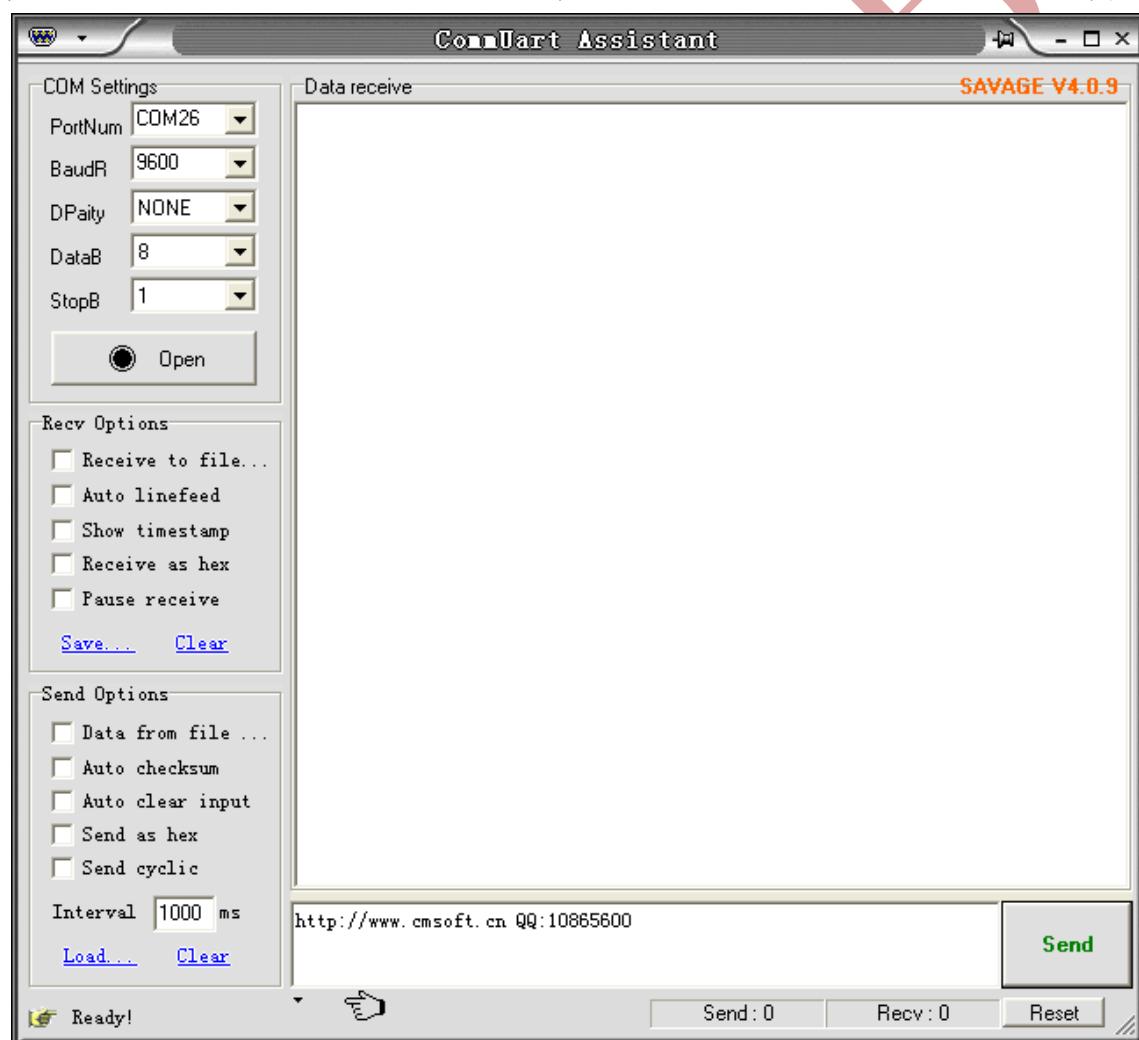
#### 5.1.1> window 下使用：

打开软件，打开 UartAssist.exe，选择 com，就可以进行 com 通讯了。

如

下

图

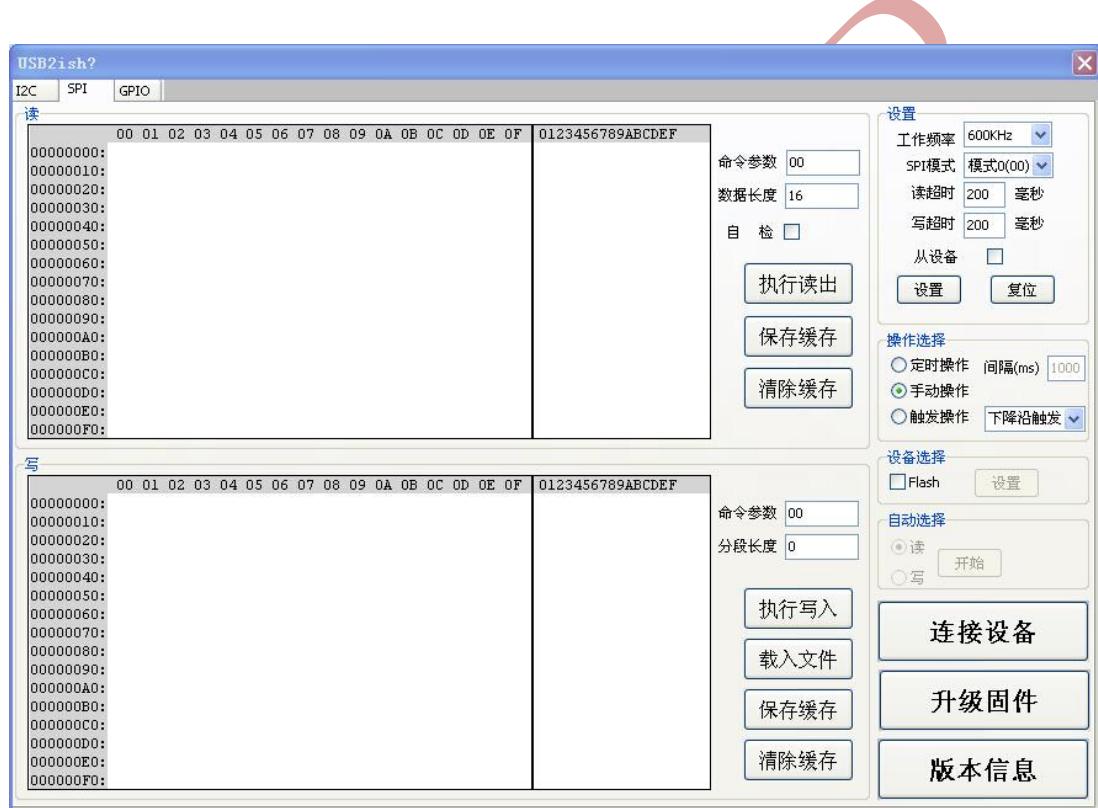


### 5.1.2> linux 下使用:

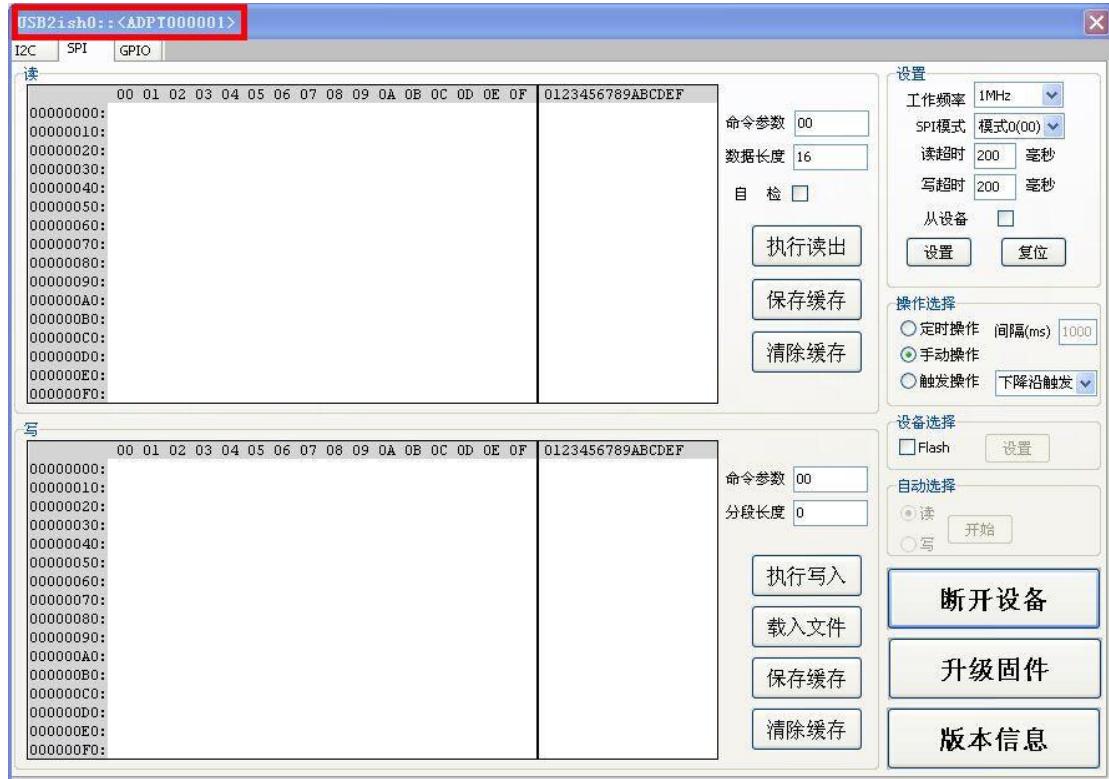
插上设备，在控制台输入 `ls /dev/ttyACM*`,如果发现有新设备 `ttyACM0~9` 出现，则证明可以在 linux 下进行 com 通讯了。

### 5.2> I2c 使用

确保功能选择跳线和升级选择跳线都为 2-3 短接。打开软件 `app/Usb2ish_pro.exe`, 如下图：



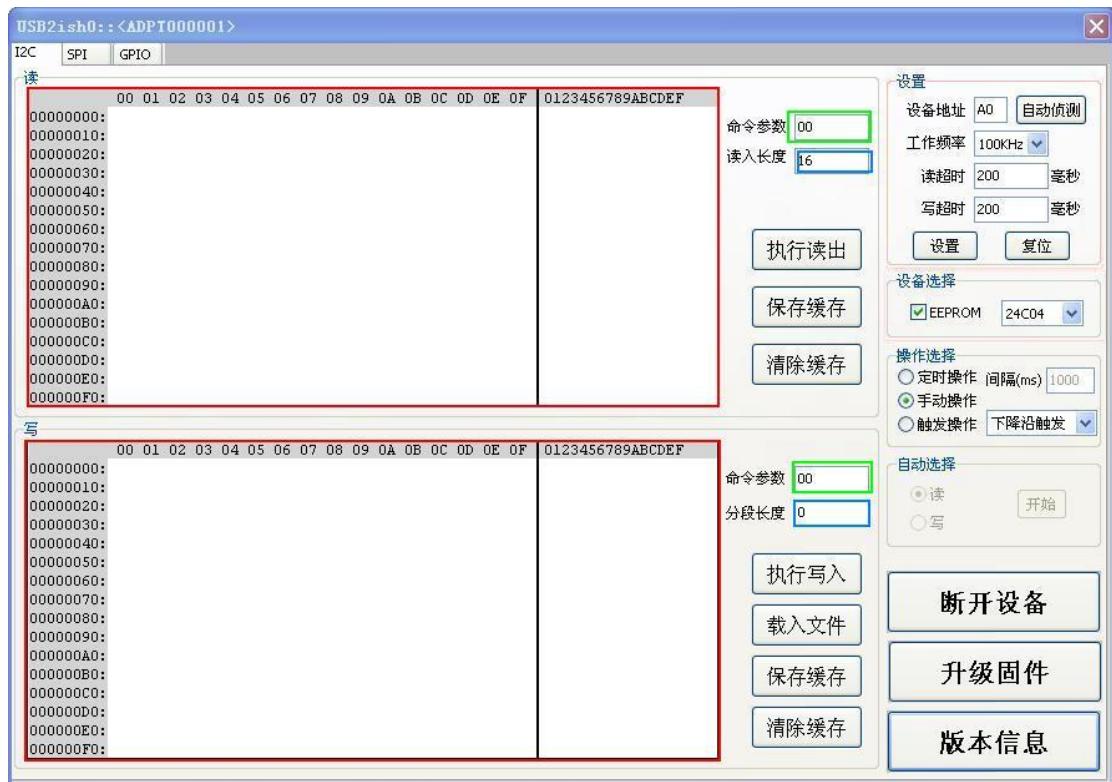
此时设备还没有连接，插上 USB 转接板，点击下边的“连接设备”按钮，可以看到连接成功后窗口标题的变化，多了设备索引号和设备序列号，如下图：



继续点击版本信息按钮，可以看到各个软件的信息，如下图：



连接成功后，切换到 I2c 页面，如下图



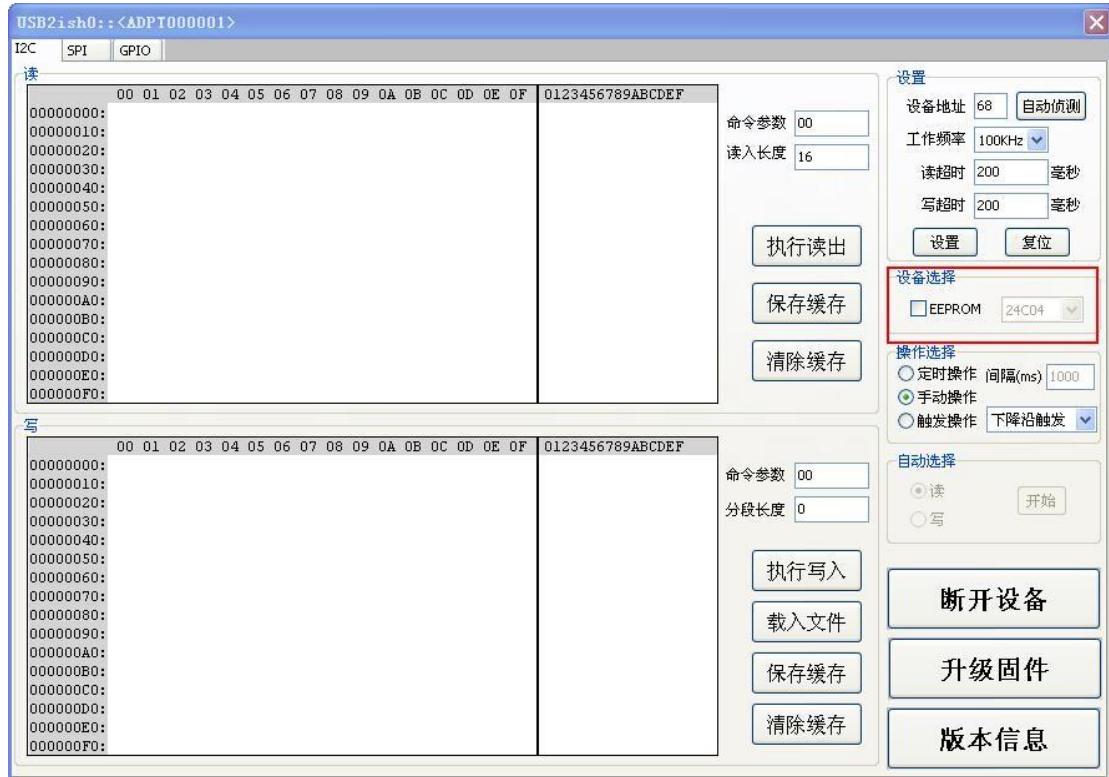
红色的区域是内容框，用于显示读写的数据，该框分为左右两部分，同步显示。左边是 hex 格式模式，右边是文本格式模式。

绿色的框为 hex 格式的输入框，用于填写 i2c 设备的命令参数或寄存器地址，可以为空，由读写的 i2c 设备决定。

蓝色的框用于填写数据长度，输入的是十进制的数字。

### 5.2.1> 选择设备

USB 转 I2C 默认的设备选择是 EEPROM，如果读写的设备不是 EEPROM 时，则必须在设备选择项里去掉 EEPROM 前面的勾选。如下图：



### 5.2.2> 设置

进行读写操作前先设置好 I2C 设备地址，通讯频率和读写超时参数，按设置按钮，直到弹出设置 OK 对话框证明设置成功。复位按钮恢复初始设置。如果不知道设备地址，可以点击自动侦测按钮，可以尝试找到连接的从设备地址，然后点击设置按钮，直到弹出设置 OK 对话框证明设置成功。如下图：



注意：

- 设备地址编辑框是 16 进制字符，最多 2 个字符，默认是 A0。
- 设备地址的放在 bit7~bit1, bit0 是读写 bit，固定是 0。
- 设置地址设置成功后，只要不断电，读写时不用反复设置。

### 5.2.3> 操作选择

操作选择里头有三种操作模式可供选择：

定时操作：每隔一定的时间进行一次读或写操作。

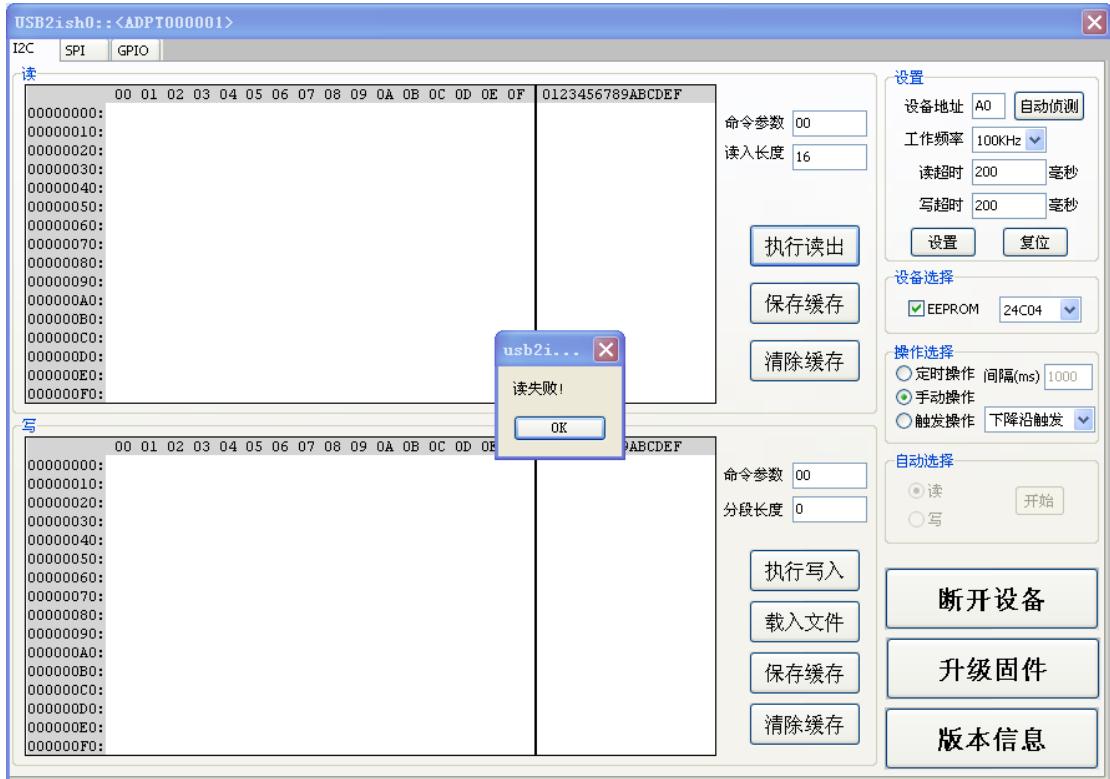
手动操作：必须点击执行读出或执行写入按钮进行一次操作。

触发操作：检测 IRQ 脚上的电平变化进行一次读或写操作。

无论是定时操作，触发还是手动操作，都只能是读或写操作。

### 5.2.4> 读写操作

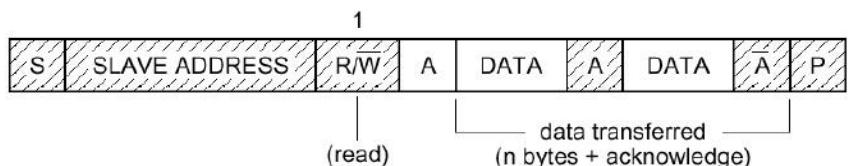
无论是读还是写，只要 I2c 读写失败，则会弹出读写失败对话框，成功则不会有任何提示。  
如下图：



I2c 设备读写有三者种格式，具体使用哪种模式，由连接的 I2c 设备决定。  
分别论述如下：

### A. 立即读模式

此种模式不常用，格式如下：



from master to slave

A = acknowledge (SDA LOW)

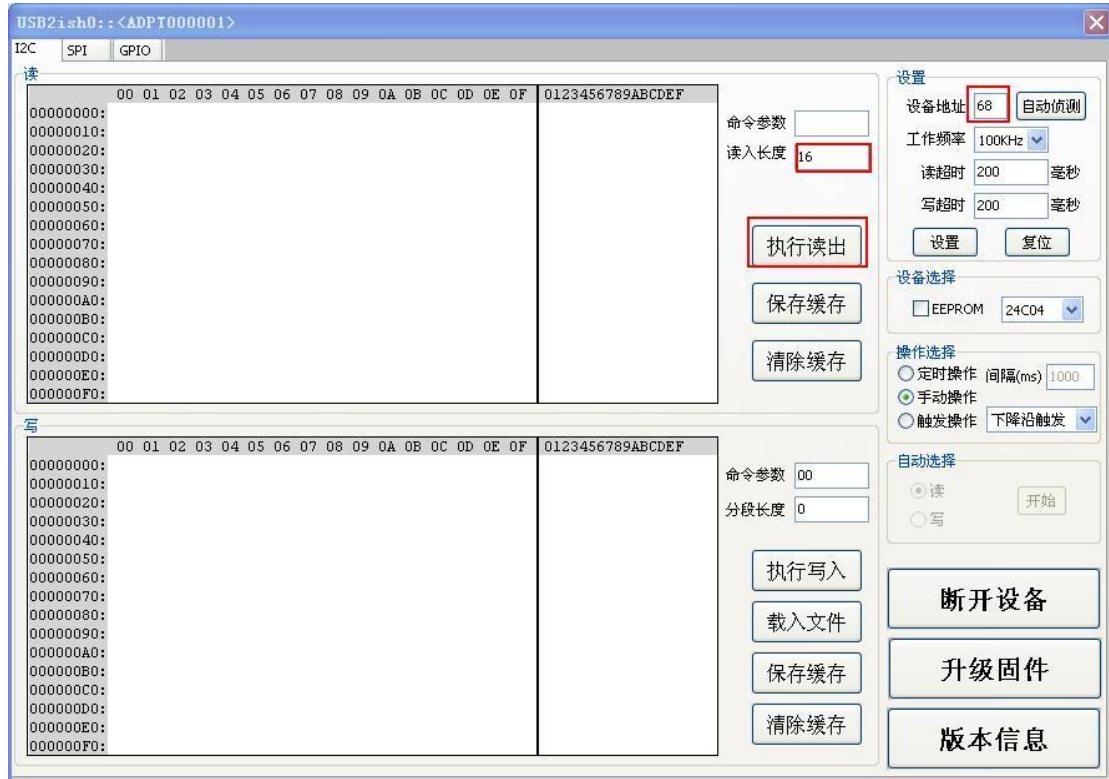
from slave to master

A = not acknowledge (SDA HIGH)

S = START condition

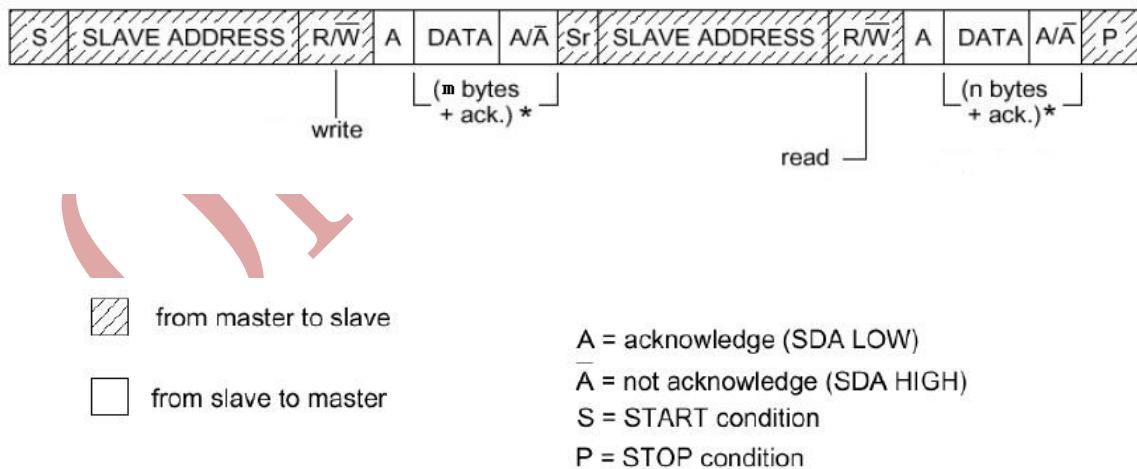
P = STOP condition

例如：设备地址是 0x68，n=16，立即读 16 个 bytes，应先设置设备地址 68，点击设置按钮，  
接着在读入的数据长度填 16，命令参数不填，然后点击执行读出，如下图：

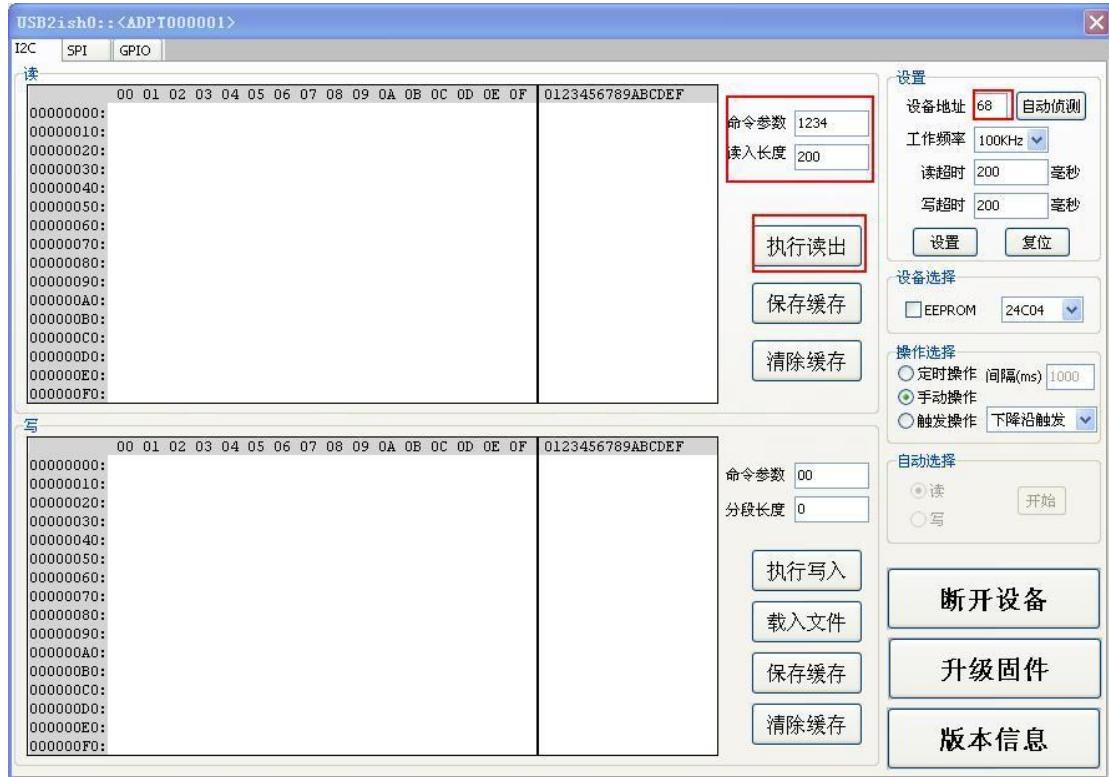


## B. 复合读模式

此种模式读用的最多，格式如下：



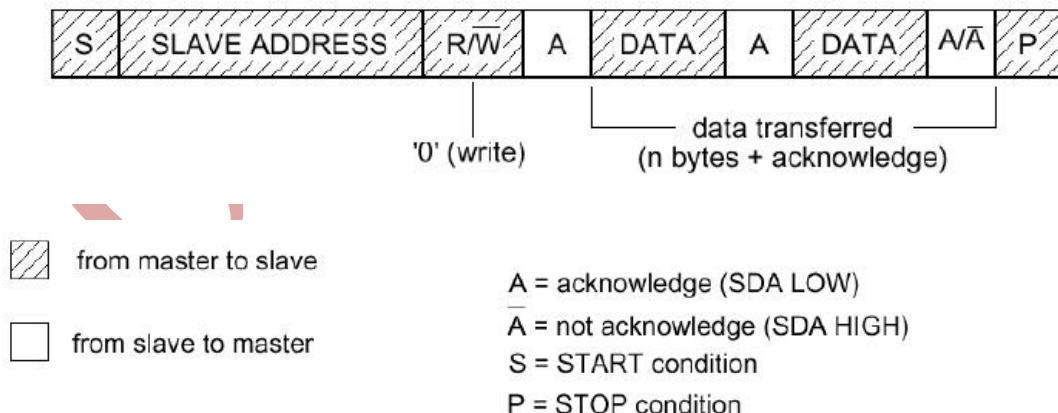
例如：设备地址是 0x68, m = 2, n = 200, m 区的 data 是 0x1234, 对应于命令参数, 按下图设置：



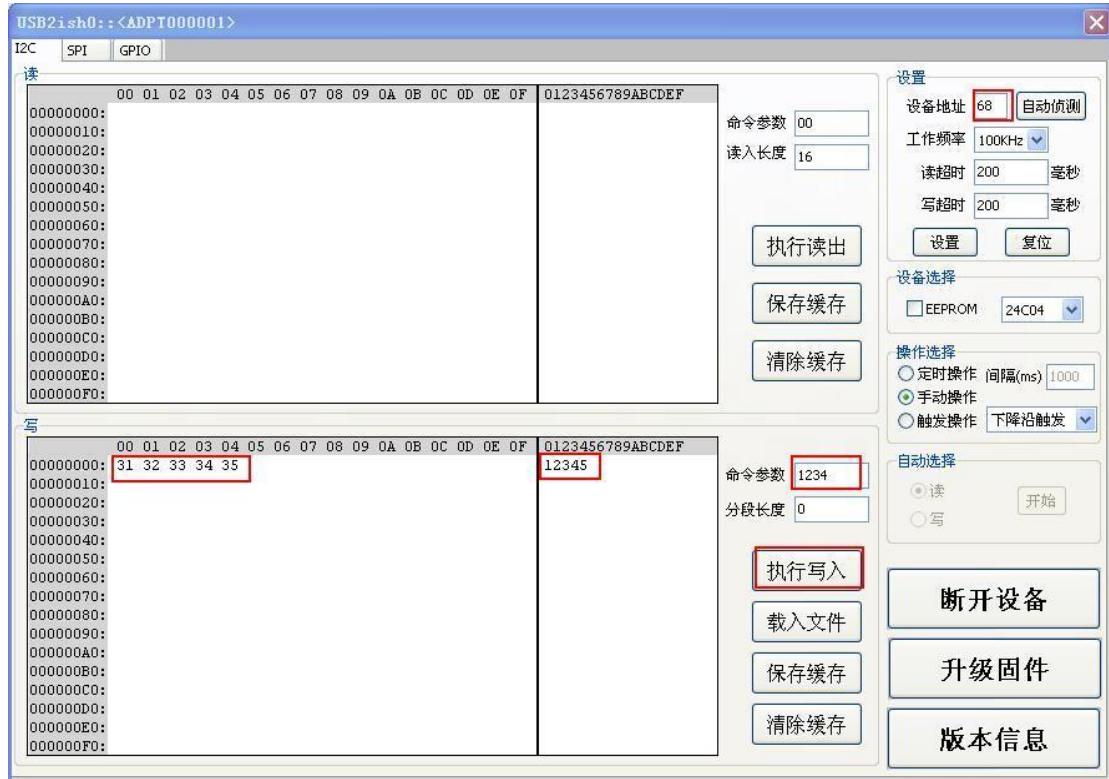
点击执行读出，即执行复合读操作。

### C. 写模式

格式如下：



例如：向 0x1234 处写入 0x31, 0x32, 0x33, 0x34, 0x35，按下图设置：



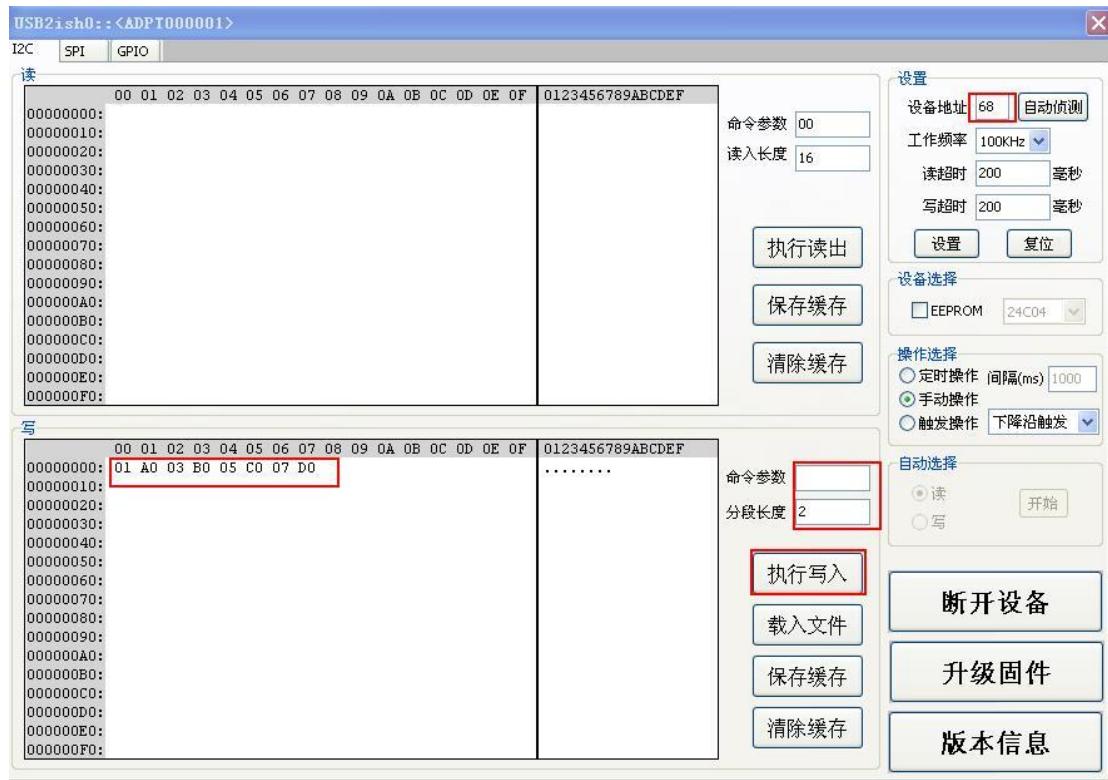
或者



点击执行写入后都是一样的效果。

### 5.2.5> 特殊功能(分段写)

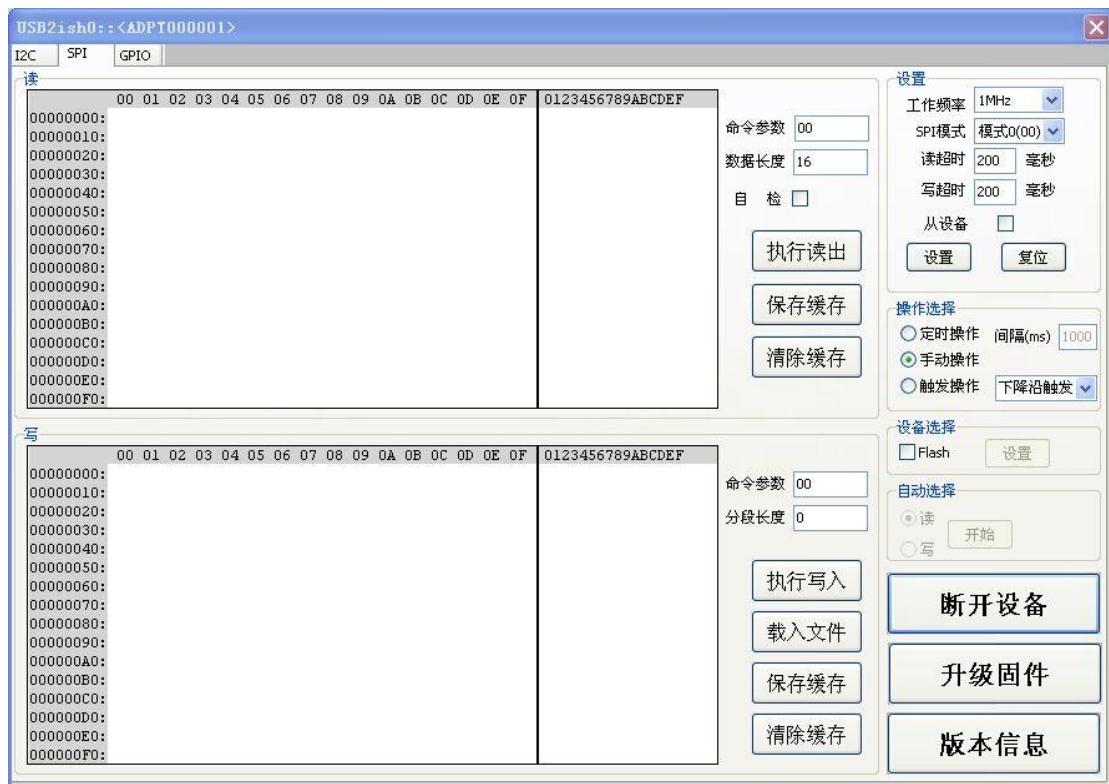
例如要向寄存器 0x01, 0x03, 0x05, 0x07 分别写入数据 0xA0, 0xB0, 0xC0, 0xD0, 则可以用分段写功能, 先设置分段写长度为 2, 写命令参数空白, 写区间配置如下图:



最后点击执行写入,则会按设置的长度分段写入,直到写完为止。

### 5.3> SPI 使用

确保功能跳线和升级跳线都为 2-3 短接。打开软件 `usb2uis.exe`. 如果不是 SPI 页面, 点击 SPI 标签, 如下图

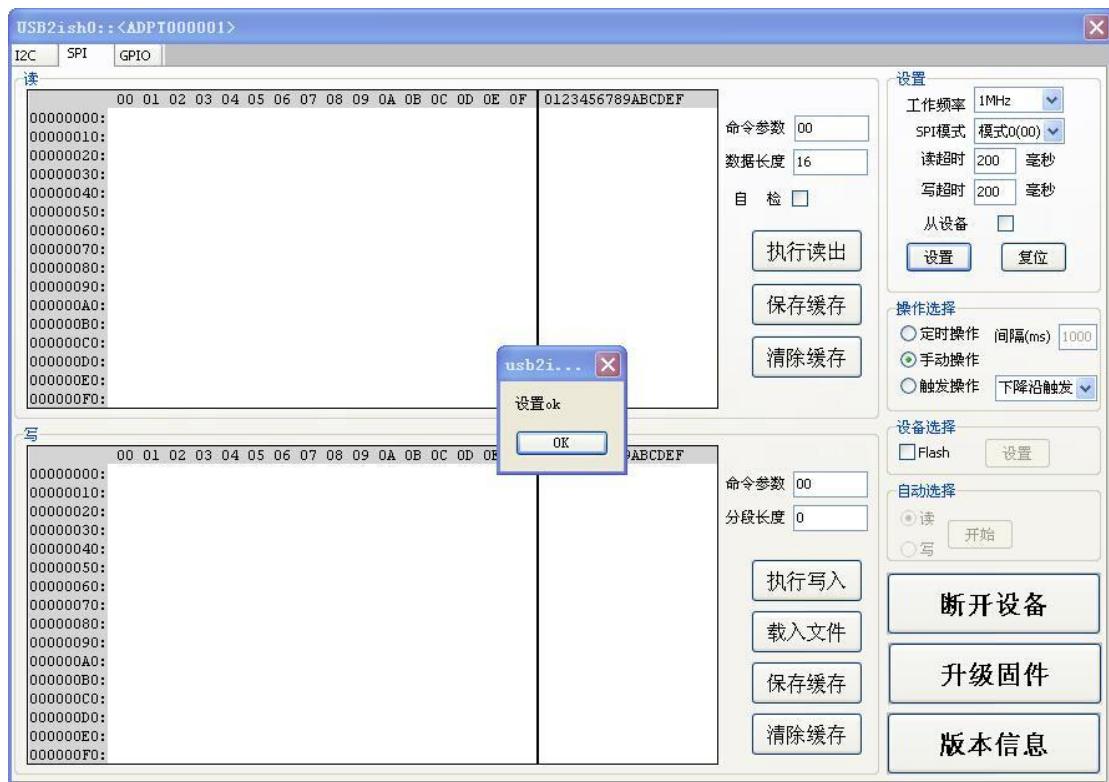


USB 转 SPI 既可以作为主设备使用，又可以作为从设备使用。默认是主设备。

### 5.3.1> 作为主设备使用

#### A. 设置

进行读写操作前先设置好 SPI 通讯频率和读写超时参数，按设置按钮，直到弹出设置 OK 对话框证明设置成功。复位按钮恢复初始设置。如下图：



### SPI 模式说明:

模式 00: SCK 在空闲状态时处于低电平, 在 SCK 周期的第一个边沿采样数据

模式 01: SCK 在空闲状态时处于高电平, 在 SCK 周期的第一个边沿采样数据

模式 10: SCK 在空闲状态时处于低电平, 在 SCK 周期的第二个边沿采样数据

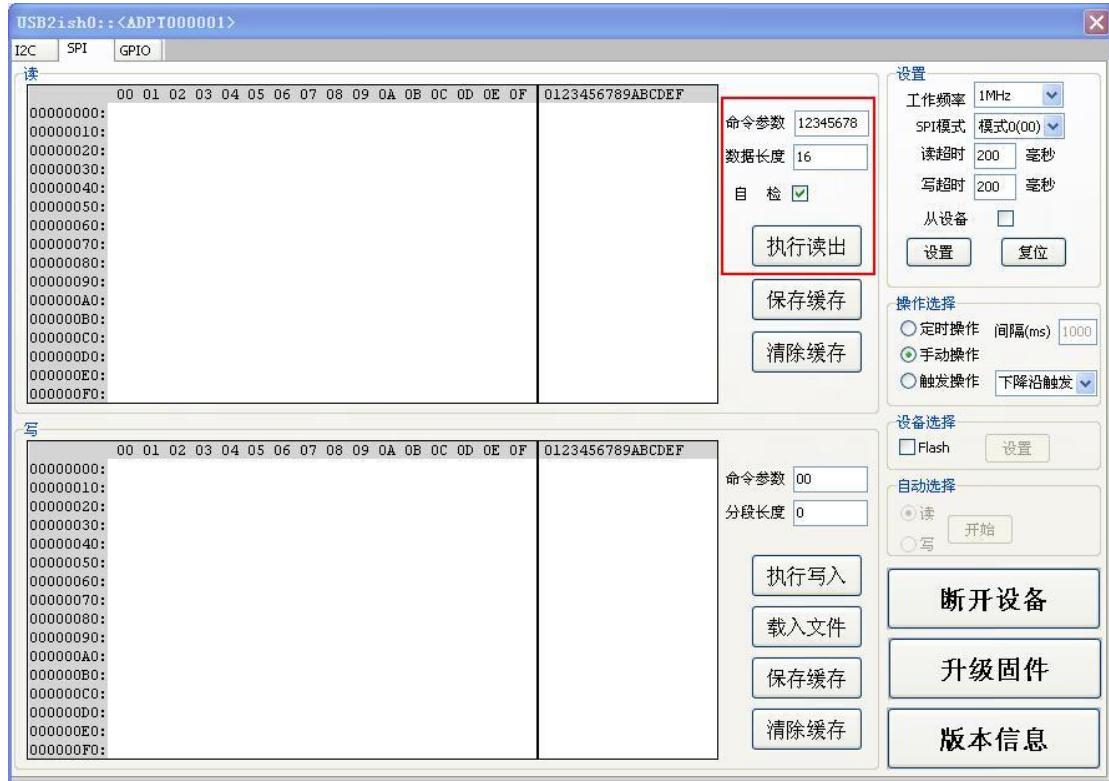
模式 11: SCK 在空闲状态时处于高电平, 在 SCK 周期的第二个边沿采样数据

### SPI 发送顺序:

MSB 优先发送

## B. 自检

将 SPI 的 MISO 和 MOSI 短接在一起, 勾选自检, 如下图填入命令参数 12345678



然后点击执行读出，则会在内容框中显示读到的数据，跟命令框一致证明 SPI 功能正常，如下图



## C. 操作选择

操作选择里头有三种操作模式可供选择:

定时操作: 每隔一定的时间进行一次读或写操作。

手动操作: 必须点击执行读出或执行写入按钮进行一次操作。

触发操作: 检测 IRQ 脚上的电平变化进行一次读或写操作。

无论是定时操作, 触发还是手动操作, 都只能是读或写操作。

## D. 读写操作

SPI 读写也有三种格式:

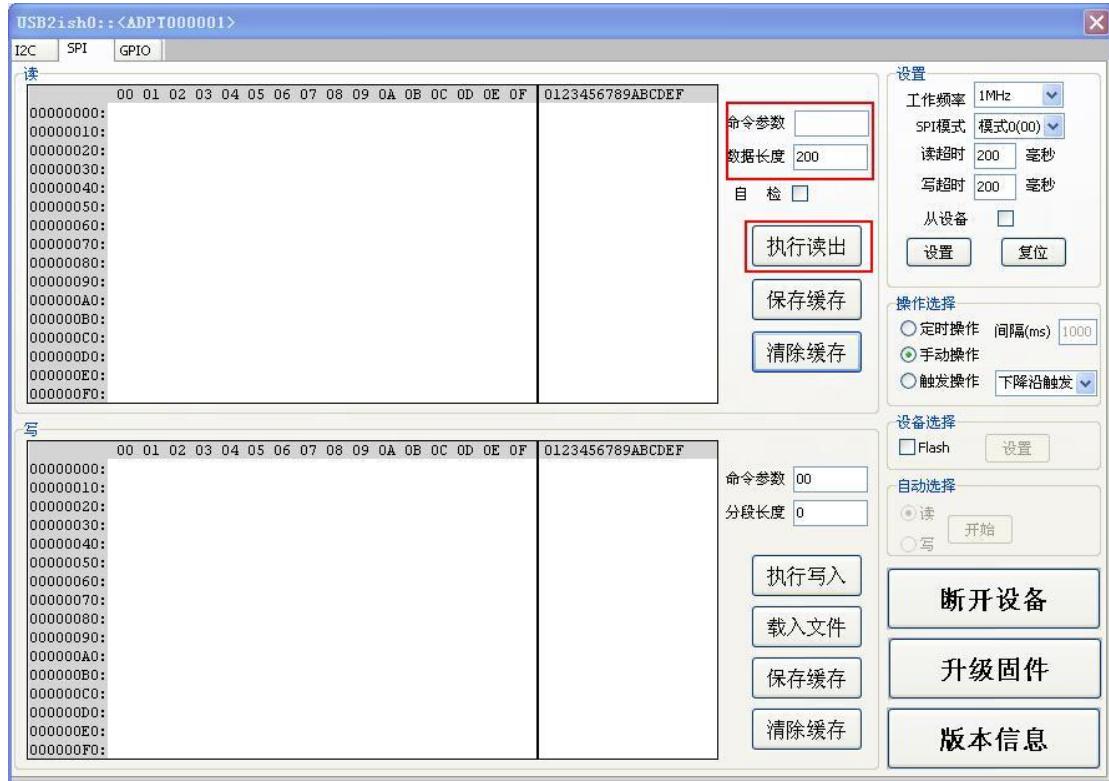
### 1. 无命令参数的读模式

如下表:

BYTE	First	Second	Third	.....	Last
MOSI	0x00	0x00	0x00	.....	0x00
MISO	Data1	Data2	Data3	.....	DataN
CSN	LOW				

如 N = 200, 通过 SPI 读 200 个 byte, 读区间配置如下图:

.然后点击执行读出。



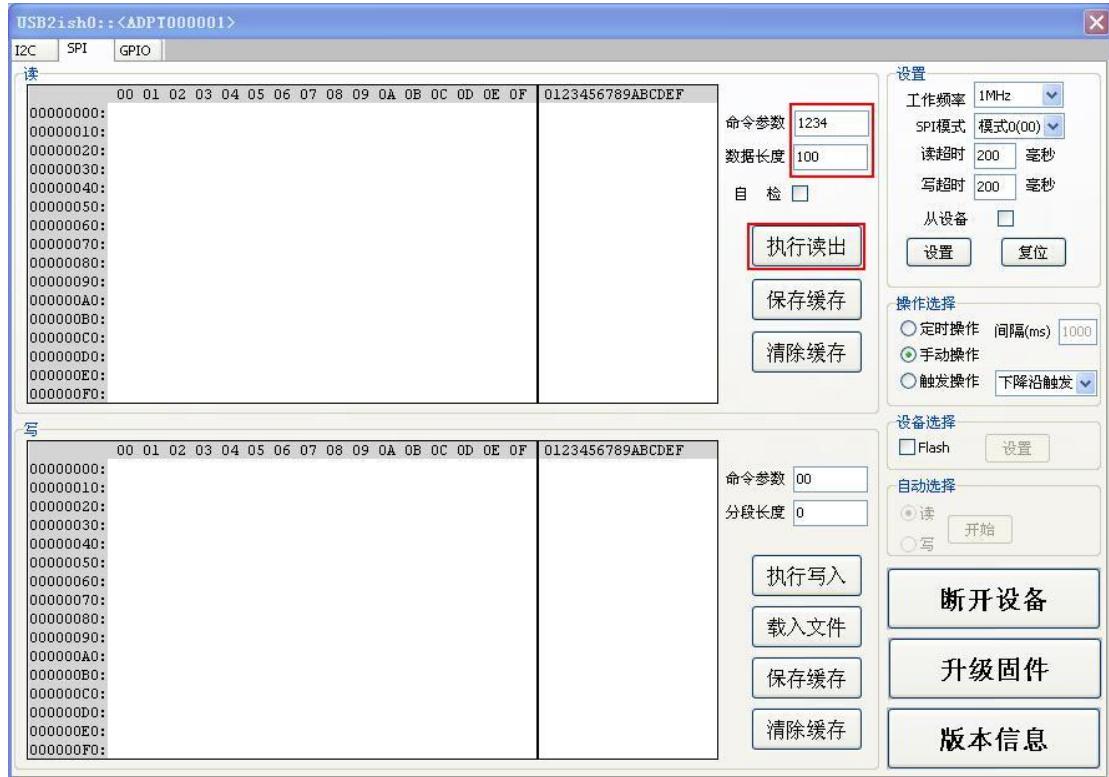
## 2. 有命令参数的读模式

如下表：

BYTE	First	Second	Third	.....	Last
MOSI	COMMAND1	COMMAND2	0x00	.....	0x00
MISO	Don't care	Don't care	Data1	.....	DataN
CSN	LOW				

这儿 COMMAND 最长可以是 4 个 byte。

如 command 是 0x1234, N = 100, 通过 SPI 读 100 个 byte, 读区间配置如下图:



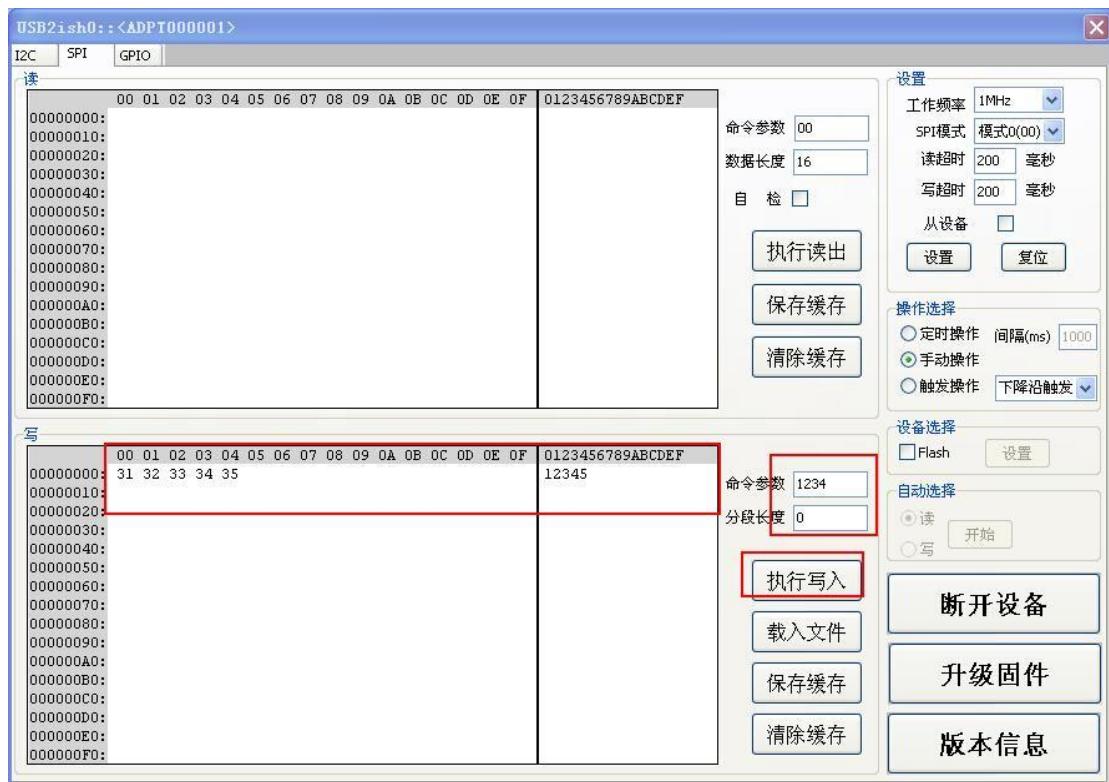
然后点击执行读出。

### 3. 写模式

BYTE	First	Second	Third	.....	Last
MOSI	COMMAND1	COMMAND2	DATA1	.....	DATAN
MISO	Don't care	Don't care	Don't care	.....	Don't care
CE	LOW				

这个 COMMAND 最长可以是 4 个 byte。

如 command 是 0x1234, 写入的数据是 0x31, 0x32, 0x33, 0x34, 0x35, 写区间配置如下图:



或者



然后点击执行写入后都是一样的效果。

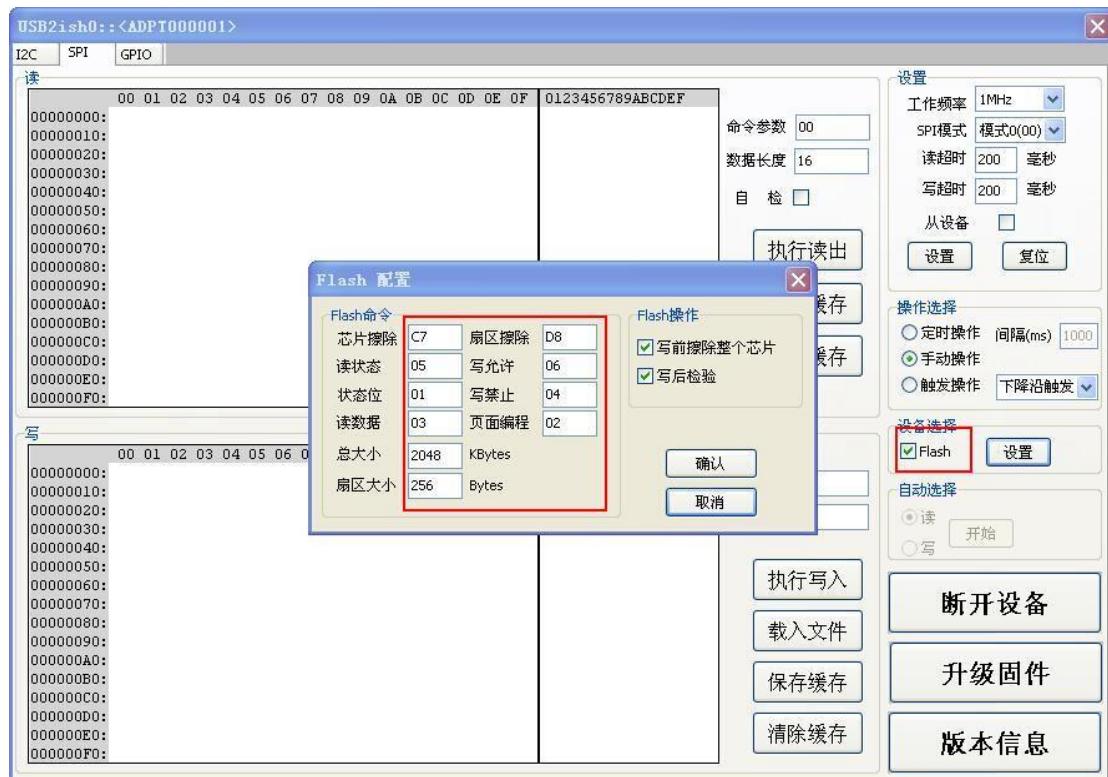
## E. 特殊功能

如 I2c 写一样，SPI 写也有分段写功能，操作方式请参考 I2c 的分段写。

## F. SPI flash 读写

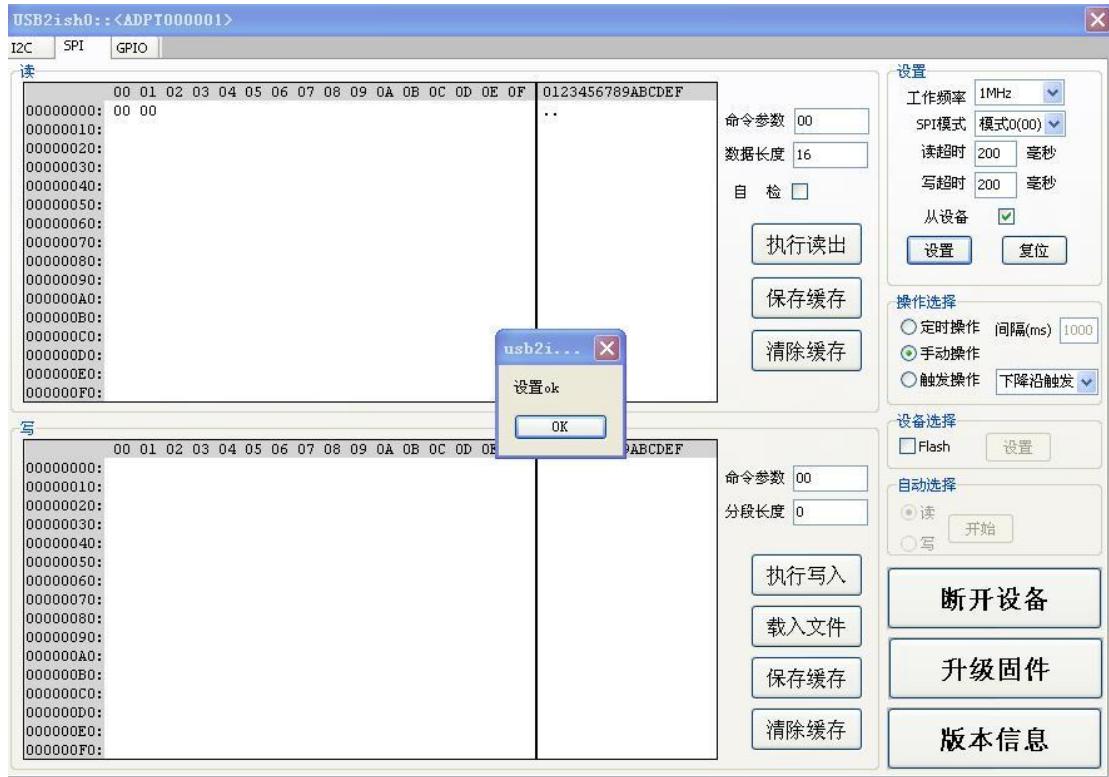
对于 SPI flash 设备的读写，可以选择勾选 flash，设置好 flash 的参数后进行读写。

命令参数就是 SPI flash 的偏移地址。



### 5.3.2> 作为从设备使用（此功能只有扩展型才有）

要 SPI 工作为从设备，需要勾选从设备，并点击设置按钮，直到弹出设置 ok 对话框指示设置成功。如下图：



作为从设备工作时,功能有如下限制:

- 1> 从设备只能接收 MOSI 上的数据, MISO 上的数据不确定。
- 2> 主设备最大的时钟频率不能超过 4M。
- 3> 主设备单次发送的数据包长度不能大于 128 BYTES。
- 4> 主设备发送的数据包间隔应该长于 5ms。

作为从设备时, 接收的数据显示在读数据区域。

## 5.4> GPIO 使用 (此功能只有扩展型才有)

如 I2C 使用那样, 连接成功后, 切换到 GPIO 页面, 如下图:



此页面功能分为三个部分：ADC，数字IO和PWM，如下图：



### 5.4.1> 数字 IO 和 ADC 采样端接口定义

标号	丝印	软件功能配置	功能说明
J7-1	VDD		电源正
J7-2	GND		电源地
J7-3	IO8	J7-03	ADC 采用通道或数字 IO
J7-4	IO7	J7-04	ADC 采用通道或数字 IO
J7-5	IO6	J7-05	ADC 采用通道或数字 IO
J7-6	IO5	J7-06	ADC 采用通道或数字 IO
J7-7	IO4	J7-07	ADC 采用通道或数字 IO
J7-8	IO3	J7-08	ADC 采用通道或数字 IO
J7-9	IO2	J7-09	ADC 采用通道或数字 IO
J7-10	IO1	J7-10	ADC 采用通道或数字 IO

### 5.4.2> ADC 使用说明

10位的分辨率，最大200 ksps采样频率，支持单端/差分输入。带有内部参考电压和一个温度传感器。J7-3和J7-10可以配置为任何通道的正极或负极输入。最多可以四个通道轮流采样。当**GND**被选择为负极输入时，**ADC**工作在单端方式；否则，**ADC**工作在差分方式。

#### A、使用采样功能

勾选所用的ADC通道，选择每个通道的正负极输入选择，按设置按钮，知道弹出设置OK对话框，确认后ADC采样设置成功。

选择采样的操作模式(分为手动，定时和触发三种模式)后，按开始采样按钮，进行采样操作。触发模式和定时采样模式启动后，要停下的话需要再按开始采样按钮(此时的按钮标题已变成停止采样)。采样值按开启的通道顺序显示在数据框里头，而且可以文本文件的形式保存在指定的地方。

#### B、采样电压计算

定义**Value** 为采样读取的值, **V<sub>REF</sub>** 为参考电压，为2.44V，

则采样的电压值**V<sub>S</sub>**计算如下：

单端方式  $V_S = Value * V_{REF} / 1023$ ; ( $V_S$  范围在0~  $V_{REF}$ 之间)

差分方式  $V_S = Value * V_{REF} / 511$ ; ( $Value$ 的高位Bit9~15全为0)

---

$$V_S = -(Value \& 0xFE) * V_{REF} / 511; \text{ ( Value 的高位 Bit9~15 全为 1)}$$

C、内部温度传感器计算

$$Temp = (V_S - 0.776) * 1000 / 2.86 \text{ 单位 } ^\circ\text{C}$$

### 5.4.2> 数字 IO 的使用

当J7-03~J7-10不被用作ADC通道使用时，可以配置为数字IO使用。

1> 在数字IO配置栏目中勾选相应的IO口，则配置为输入，不勾选的配置为输出，选择好后要按设置按钮使设置生效，弹出设置OK对话框后证明设置成功，相应的数字IO读写栏目里头的电平会重新刷新一次。

2> 数字IO读写栏目里头的读出和写入按钮用于刷新或变更数字IO的端口电平。

配置为ADC采样选择的IO口不能配置为数字IO，相应数字IO配置栏里头的勾选变为灰色。配置为ADC采样选择或配置为数字IO输入的IO口在数字IO读写栏里头变为灰色，只能读出，不能写入。且配置为ADC采样选择的IO在读入时其电平总是0.

### 5.4.3> PWM 使用

本设备最多可以输出四路，100K的PWM，每个PWM的输出频率都是一样的且同时开启和关闭。

占空比则可以单独调整。通道数为0是没有PWM输出，为1时一号通道输出，为2时，一二号通道输出，以此类推。

选择好通道数，频率和占空比后，按设置按钮，弹出设置OK对话框后证明设置成功。

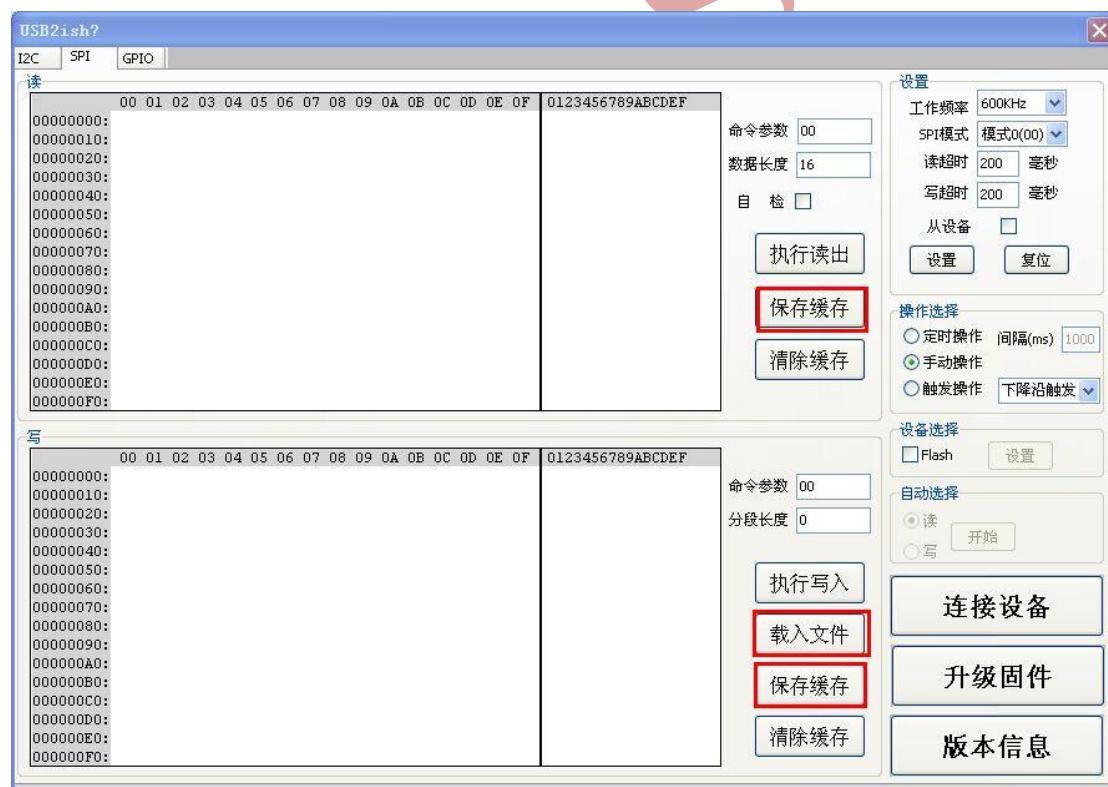
由于软件的运行误差，最大占空比有一点点限制，跟PWM频率有一定的关系。

实测最大占空比见下表：

频率(KHz)	最小占空比	最大占空比	占空比为0时输出	超过最大占空比时输出
1	0.1%	99.9%	高电平	低电平
2	0.1%	99.8%	高电平	低电平
4	0.1%	99.7%	高电平	低电平
6	0.1%	99.6%	高电平	低电平
8	0.1%	99.5%	高电平	低电平
10	0.1%	99.3%	高电平	低电平
20	0.1%	98.7%	高电平	低电平
40	0.1%	97.5%	高电平	低电平
60	0.1%	96.2%	高电平	低电平
80	0.1%	95.0%	高电平	低电平
100	0.1%	93.7%	高电平	低电平

## 5.5> 文件操作

在 I2c 和 SPI 标签页有载入文件和保存缓存实现文件的读和写。

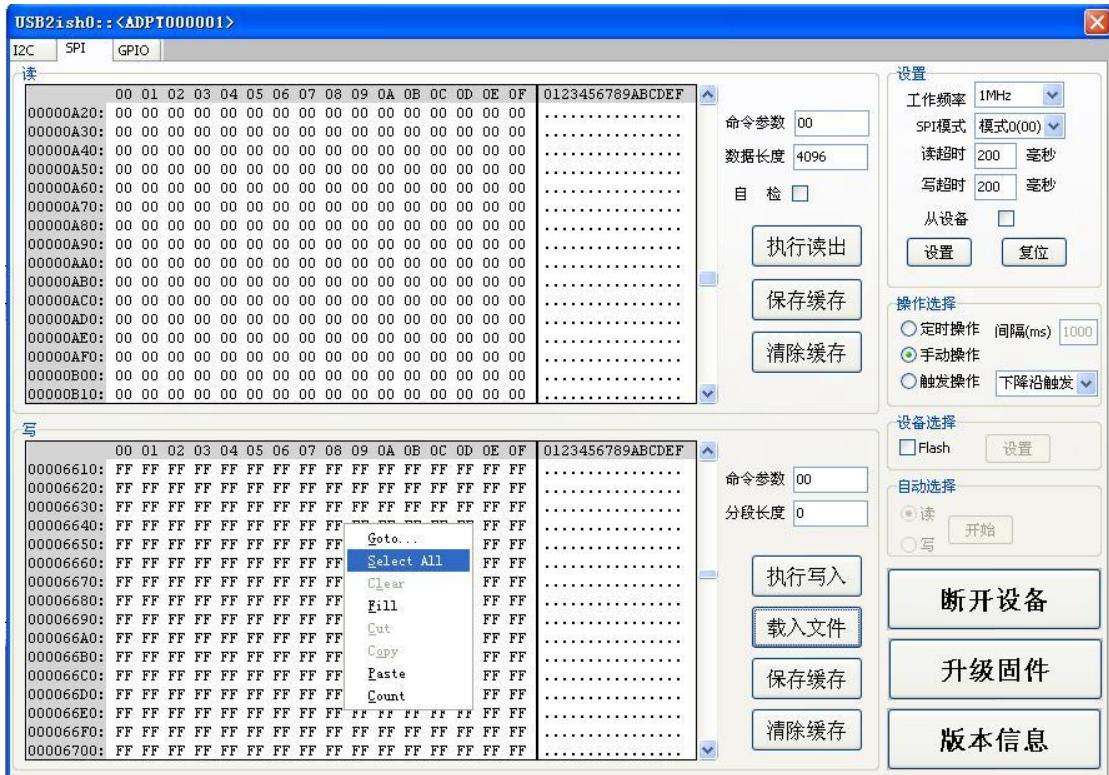


载入文件：任何格式的文件都可以载入进来，只有扩展名为 hex 的文件会进行解码，其他格式的文件一律按文本文件的格式载入缓存，文件大小跟缓存数据长度一致。非显示字符在文本模式显示下 ‘.’。在 hex 模式下显示其十六进制值。

保存缓存：将内容框里的数据按文本文件的格式保存在电脑中，文件大小跟缓存数据长度一致。

## 5.6> 弹出菜单

讲鼠标移动至在内容框范围内，点击鼠标右键，会弹出一个菜单见下图：



菜单说明：

Goto...: 将编辑光标定位到所在偏移地址

Select All: 全选所有缓存内容

Clean: 将所选内容删除

Fill: 将某块区域内容全部改为相同的字节

Cut: 将所选内容删除，并 copy 到粘贴板

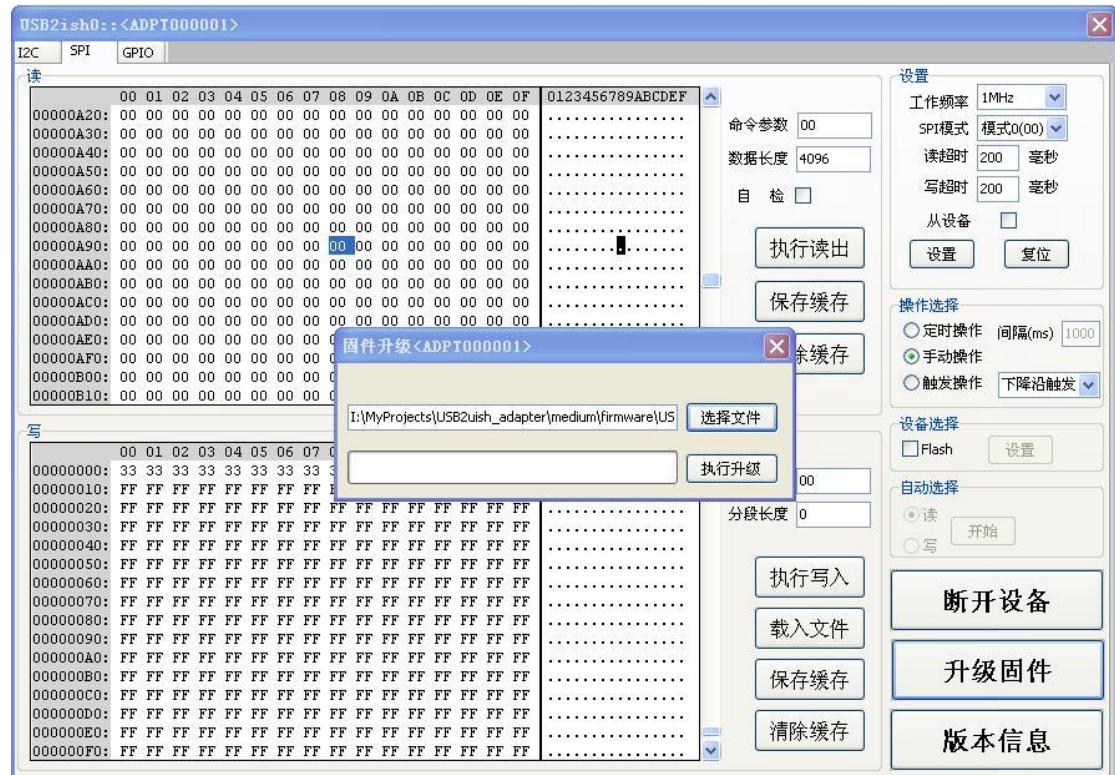
Copy: 将所选内容 copy 到粘贴板

Paste: 将粘贴板内容插入到编辑光标所在位置

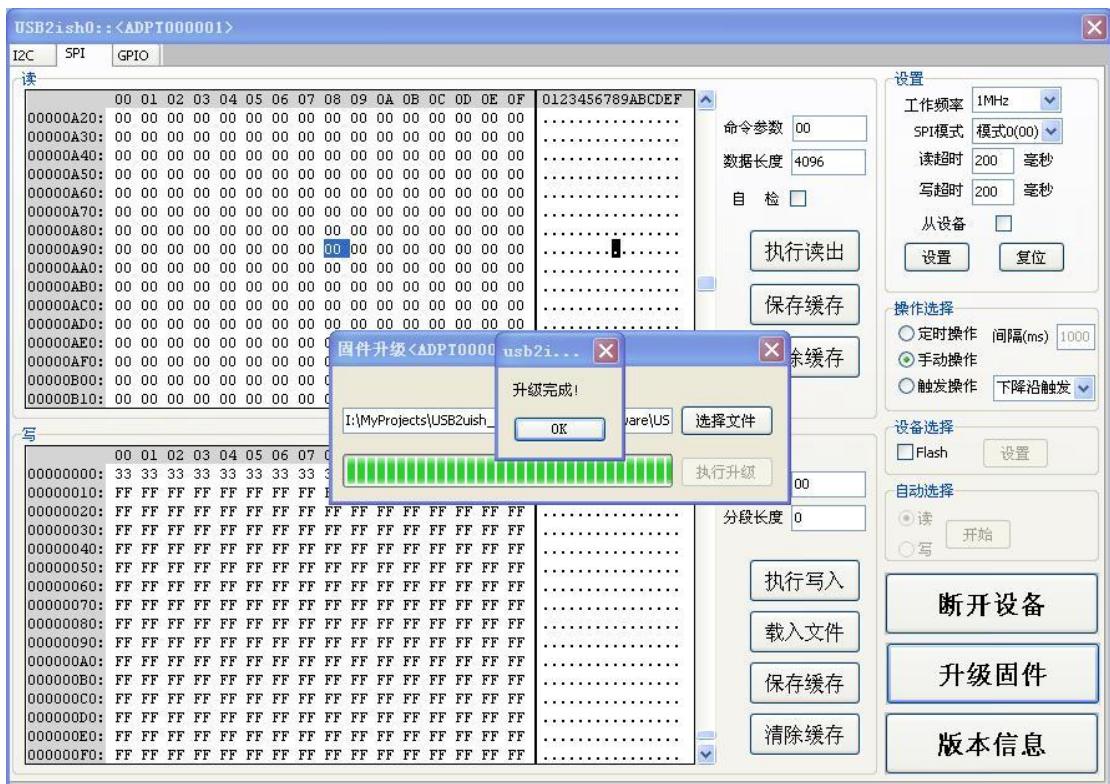
Count: 统计内容框缓存的长度，单位是 byte

## 5.7> 固件更新

确保升级选择跳线为 1-2 短接，插入 USB2ish 设备，成功连接设备后，点击固件升级按钮，如下图：



点击选择文件后，再按执行，成功后如下图：



不同的软件有不同的版本和发行时间，更新成功后，将升级选择跳线改回 2-3 短接，插入 USB2ish 转接板，成功连接设备后点击版本信息，可以查看固件版本和发行时间，检测是否更新成功。如下图：



版本如有变化，以实际使用为准。

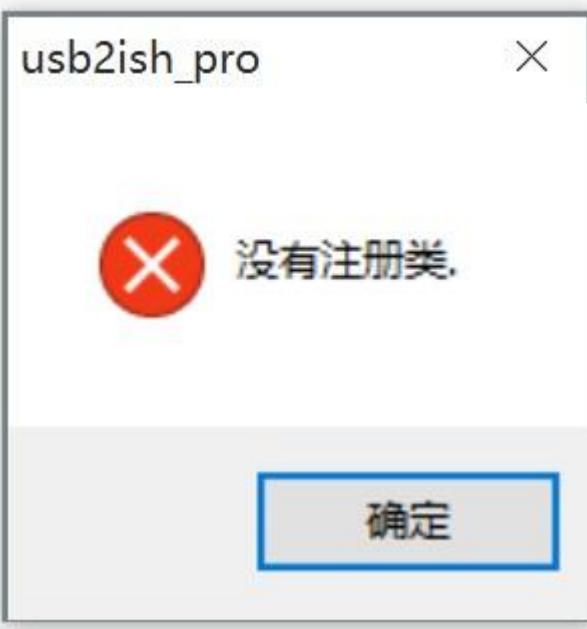
## 六、 Linux 下使用 I2c,SPI 和 PWM 功能

- 1> 确保 linux 下已安装 libusb 软件包和 qt-creator。
- 2> 打开 linux 下的 terminal 软件，进入软件包的 lib/linux 目录。
- 3> 输入命令 sudo sh install.sh 安装库文件
- 4> 用 qt-creator 打开软件包 DEMO/qt-creator301/下的 I2c\_RW.pro 或者 SPI\_RW.pro 项目。
- 5> 编译运行即可进行 I2c 或 SPI 的读写操作。

注意：库文件和 DEMO 程序在 ubutun14.04(64bit)测试通过。  
Qt 项目必须不是 root 账户开发。

## 七、 常见问题及解决方法

	故障现象	可能原因	解决方法
1	拔插 USB，设备管理器没有反应	PC USB 端口坏	更换至其他 USB 端口
		驱动程序僵死	重新启动电脑
		USB 转接板坏	更换转接板
2	读不到软件版本信息，如下图：很多问号	DLL 调用出错	退出应用程序，拔下板子，再插上，重开应用程序
		驱动程序僵死	重新启动电脑
3	Usb2ish_pro.exe 运行时，弹出如下提示窗口	应用程序没有执行权限	右键选择以管理员权限运行

			
4	驱动程序安装失败	检测跳线，驱动文件夹选择是否正确	
		360 杀毒软件	关闭 360 后再安装驱动
5	安装 UART 驱动失败问题,如下图: 	<p>1&gt; window 系统 inf 文件夹下，没有 mdmcpq.inf 文件，或者 system32\dirver\ 没有 usbser.sys 文件</p> <p>2&gt; mdmcpq.inf 文件段中的语法错</p>	<p>1&gt; 网上下载相应系统版本的 mdmcpq.inf 和 usbser.sys 到指定的文件夹</p> <p>2&gt; 将 :\Windows\inf\mdmcpq.inf 中 [FakeModemCopyFileSection] usbser.sys,,0x20 改为 [FakeModemCopyFileSection] ;usbser.sys,,,0x20</p>

---

## 八、 维护

- 1> 一年的非人为硬件损坏，可以免费更换。
- 2> 终身包修，只需承担材料费和运费。
- 3> 特殊要求的可定制。

OneasvB