

USB总线转I2C总线接口芯片

USB2I2C-S/T

中文手册版本：3.0A

<http://www.usb-i2c-spi.com>

1、USB2I2C概述

USB2I2C是一个USB总线的转I2C总线的专用接口芯片。通过USB2I2C芯片用户可以非常方便地实现PC机USB总线和下位机端各种I2C/IIC设备（如，ATMEL公司的AT24CX系列EEPROM；I2C总线8位并行I/O口扩展芯片PCF8574/JLC1562；I2C接口实时时钟芯片DS1307/PCF8563/SD2000D/M41T80/ME901/ISL1208/；I2C数据采集ADC芯片MCP3221（12bitADC）/ADS1100（16bitADC）/ADS1112（16bitADC）/MAX1238（12bitADC）/MAX1239（12bitADC）；I2C接口数模转换DAC芯片DAC5574（8bitDAC）/DAC6573（10bitDAC）/DAC8571（16bitDAC）/；I2C接口温度传感器TMP101/TMP275/DS1621/MAX6625，等）之间的通信。

USB2I2C芯片上位机PC端提供简单易用的USBIOX.DLL动态库调用，可以方便地被VB, VC, Delphi等上位机开发工具调用。相关例程在公司网站<http://www.usb-i2c-spi.com/cn/down.htm>可以找到，您也可以通过代理商或经销商得到这些例程。

2、USB2I2C功能特点

2.1. 概述

- 全速USB设备接口，兼容USBV2.0。
- 直接实现I2C总线与上位PC机的连接，无需辅助MCU。
- 片内置了USB上拉电阻，UD+和UD-引脚可直接连接到PC机USB总线上。
- 外围元器件只需要一个12M晶体和两个电容。
- 作为I2C总线的Host/Master主机端。
- I2C接口提供SCL和SDA信号线，支持SCL时钟4种不同传输速度：20KHz/100KHz/400KHz/750KHz。
- 可以通过外部的低成本串行EEPROM定义厂商ID、产品ID、序列号等。
- 低成本，SOP-28封装和小型的SSOP-20封装。

2.2. 注意

- 由于是通过USB转换的接口，所以只能做到应用层兼容，而无法绝对相同。

3、USB2I2C引脚及封装

		USB2I2C				USB2I2C			
1				20	5	1	ACT	VCC	28
2	ACT	VCC		19	6	2	RST#	NC.	27
3	NC.	NC.		18	7	3	NC.	NC.	26
4	NC.	NC.		17	8	4	NC.	NC.	25
5	NC.	NC.		16	9	5	NC.	SCL	24
6	INT	SCL		15	10	6	NC.	SDA	23
7	V3	SDA		14	11	7	INT	NC./D7	22
8	UD+	NC.		13	12	8	NC.	NC./D6	21
9	UD-	VCC		12	13	9	V3	NC./D5	20
10	XI	GND		11	14	10	UD+	NC./D4	19
	XO	GND				11	UD-	NC./D3	18
						12	GND	NC./D2	17
						13	XI	NC./D1	16
						14	XO	NC./D0	15

封装形式	塑体宽度		引脚间距		封装说明	订货型号
SOP-28	7.62mm	300mil	1.27mm	50mil	标准的28脚贴片	USB2I2C-S
SSOP-20	5.30mm	209mil	0.65mm	25mil	超小型20脚贴片	USB2I2C-V

4、引脚及功能说明

4.1. 一般说明

USB2I2C-V采用SSOP-20密脚封装，是USB2I2C-S的筒装版，两者相同名称的引脚具有相同的功能。下面的引脚说明针对SOP-28封装的USB2I2C-S，除了引脚号不同之外都适用于USB2I2C-V。

4.2. 标准引脚

引脚号	引脚名称	类型	引脚说明
1	ACT	输出	USB设备配置完成状态输出，低电平有效
2	RSTI	输入	外部复位输入，高电平有效，内置下拉电阻
9	V3	电源	3.3V供电时连接VCC外部电源，在5V供电时外接0.01uF退耦电容
10	UD+	双向三态	直接连到USB总线的D+数据线，内置上拉电阻
11	UD-	双向三态	直接连到USB总线的D-数据线
12	GND	电源	公共接地端，直接连到USB总线的地线
13	XI	输入	晶体振荡的输入端，需要外接晶体及振荡电容
14	XO	输出	晶体振荡的反相输出端，需要外接晶体及振荡电容
28	VCC	电源	正电源输入端，需要外接0.1uF电源退耦电容

4.3. I2C总线功能引脚（适用于USB2I2C-S和USB2I2C-V芯片）

引脚号	引脚名称	类型	引脚说明
7	INT	输入	中断请求输入，上升沿有效，内置上拉电阻
23	SDA	开漏输出及输入	I2C总线数据输入输出，内置上拉电阻
24	SCL	开漏输出	I2C总线时钟输出，内置上拉电阻
26	RST	输出	复位输出，低电平有效
其它引脚	NC.	NC.	不用

5、功能说明

本手册中的数据，后缀B为二进制数，后缀H为十六进制数，否则为十进制数。

5.1. 硬件说明

USB2I2C芯片的所有类型为三态输出的引脚，都内置了上拉电阻，在芯片复位完成后作为输出引脚，而在芯片复位期间三态输出被禁止，由内置的上拉电阻提供上拉电流。如果必要，外部电路可以在电路中再提供外置的上拉电阻或者下拉电阻，从而设定相关引脚在USB2I2C芯片复位期间的默认电平，外置上拉电阻或者下拉电阻的阻值通常在 $2K\Omega \sim 5K\Omega$ 之间。

USB2I2C芯片的ACT引脚用于USB设备配置完成状态输出。当USB设备尚未配置或者取消配置后，该引脚输出高电平，当USB设备配置完成后，该引脚输出低电平。ACT引脚可以外接串了限流电阻的发光二级管LED，用于指示USB设备的配置完成状态。

USB2I2C芯片内置了USB上拉电阻，UD+和UD-引脚应该直接连接到USB总线上。

USB2I2C芯片内置了电源上电复位电路。RSTI引脚用于从外部输入异步复位信号；当RSTI引脚为高电平时，USB2I2C芯片被复位；当RSTI引脚恢复为低电平后，USB2I2C会继续延时复位20ms左右，然后进入正常工作状态。为了在电源上电期间可靠复位并且减少外部干扰，建议在RSTI引脚与VCC之间跨接一个容量为0.47uF左右的电容。

USB2I2C芯片正常工作时需要外部向XI引脚提供12MHz的时钟信号。一般情况下，时钟信号由USB2I2C内置的反相器通过晶体稳频振荡产生。外围电路只需要在XI和XO引脚之间连接一个12MHz的晶体，并且分别为XI和XO引脚对地连接振荡电容。

USB2I2C芯片使用5V电源电压时，V3引脚应该外接容量为0.01 μ F左右的电源退耦电容。3.3V供电时外部电源VCC直接连接V3引脚。

5.2. 功能配置

USB2I2C芯片可以通过配置连接外部的串行EEPROM芯片，通过EEPROM芯片定义芯片功能、厂商ID、产品ID等。配置芯片应该选用7位地址的24CXX系列芯片，例如：24C01A、24C02、24C04、24C16等。其特点是：可以灵活地定义芯片功能和USB产品的各种常用识别信息。通过Windows下的工具软件USB2I2CCFG.EXE（公司网站可以找到，您也可以通过代理商或经销商得到），可以随时在线修改串行EEPROM中的数据，重新定义USB2I2C芯片的各种识别信息。

一般情况下，复位后USB2I2C首先通过SCL和SDA引脚查看外部配置芯片中的内容。

下表是外部串行EEPROM配置芯片中的内容。

字节地址	简称	说明	默认值	
00H	SIG	外部配置芯片有效标志，首字节必须是53H，其它值则配置数据无效，使用直接组合配置	53H	
02H	CFG	芯片的具体配置，参考下表按位说明	USB2I2C-S	FEH
			USB2I2C-V	FAH
03H		（保留单元，必须为00H或者0FFH）	00H	
05H~04H	VID	VendorID，厂商识别码，高字节在后，任意值	4348H	
07H~06H	PID	ProductID，产品识别码，高字节在后，任意值	55??H	
09H~08H	RID	ReleaseID，产品版本号，高字节在后，任意值	0100H	
17H~10H	SN	SerialNumber，产品序列号字符串，长度为8	12345678	
其它地址		（保留单元）	00H或FFH	

6、参数

6.1. 绝对最大值（临界或者超过绝对最大值将可能导致芯片工作不正常甚至损坏）

名称	参数说明	最小值	最大值	单位
TA	工作时的环境温度	-20	70	°C
TS	储存时的环境温度	-55	125	°C
VCC	电源电压（VCC接电源，GND接地）	-0.5	6.5	V
VIO	输入或者输出引脚上的电压	-0.5	VCC+0.5	V

6.2. 电气参数（测试条件：TA=25°C，VCC=5V，不包括连接USB总线的引脚）

名称	参数说明	最小值	典型值	最大值	单位
VCC	电源电压（V3引脚不连VCC引脚）	4.5	5	5.3	V
ICC	工作时总电源电流		15	30	mA
ISLP	USB挂起时的总电源电流		0.5		mA
VIL	低电平输入电压	-0.5		0.7	V
VIH	高电平输入电压	2.0		VCC+0.5	V
VOL	低电平输出电压（4mA吸入电流）			0.5	V
VOH	高电平输出电压（4mA输出电流）（芯片复位期间仅100 μ A输出电流）	VCC-0.5			V
IUPs	SCL和SDA引脚的高电平输出电流	100	200	500	μ A

IUP	内置上拉电阻的输入端的输入电流	40	80	160	uA
IDN	内置下拉电阻的输入端的输入电流		-50		uA
VR	电源上电复位的电压门限	2.3	2.6	2.9	V

名称	参数说明	最小值	典型值	最大值	单位
FCLK	XI引脚的输入时钟信号的频率	11.98	12.00	12.02	MHz
TPR	电源上电的复位时间		20	40	mS
TRI	外部复位输入的有效信号宽度	100			nS
TRD	外部复位输入后的复位延时		30		mS

名称	参数说明	最小值	典型值	最大值	单位
TWA	接收应答ACK的低电平有效宽度	100			nS
TAK	接收应答ACK上升沿到STB有效	400			nS

7、应用设计

7.1. 基本连接（图1）

I2C总线支持多个设备的地址识别，采用数据流方式读写数据，支持一次读写较大的数据块。USB2I2C的两线串口支持20KHz/100KHz/400KHz/750KHz的速度，与具有硬件两线串口的设备连接时可以选择较高的速度，与软件模拟两线串口的单片机连接时只能选择较低的速度（例如20KHz）。

图1中使用电阻R4强制ACT#引脚在USB2I2C功能配置期间为低电平，从而禁止USB2I2C配置时访问2线同步串口而影响总线上的其它设备，如果不需要LED显示，那么可以省去电阻R3和发光管L2，并且可以将R4接RST#端改为接GND。

P7是USB端口，USB总线包括一对5V电源线和一对数据信号线，通常，+5V电源线是红色，接地线是黑色，D+信号线是绿色，D-信号线是白色。USB总线提供的电源电流最大可以达到500mA，一般情况下，USB2I2C芯片和低功耗的USB产品可以直接使用USB总线提供的5V电源。如果USB产品通过其它供电方式提供常备电源，那么USB2I2C也应该使用该常备电源，如果需要同时使用USB总线的电源，那么可以通过阻值约为1Ω的电阻连接USB总线的5V电源线与USB产品的5V常备电源，并且两者的接地线直接相连接。

C13和C14是独石或高频瓷片电容，C13容量为1000pF到0.01μF，用于USB2I2C内部电源节点退耦，C14容量为0.1μF，用于外部电源退耦。晶体X3、电容C11和C12用于时钟振荡电路。X3的频率是12MHz，C11和C12是容量为15pF的独石或高频瓷片电容。

如果USB产品使用USB总线的电源，并且在VCC与GND之间并联了较大的电容C15，使得电源上电过程较慢并且电源断电后不能及时放电，那么USB2I2C将不能可靠复位。建议在RSTI引脚与VCC之间跨接一个容量为0.47μF的电容延长复位时间。

在设计印刷电路板PCB时，需要注意：退耦电容C13和C14尽量靠近USB2I2C的相连引脚；使D+和D-信号线贴近平行布线，尽量在两侧提供地线或者覆铜，减少来自外界的信号干扰；尽量缩短XI和X0引脚相关信号线的长度，为了减少高频干扰，可以在相关元器件周边环绕地线或者覆铜。

限流电阻R5是可选器件，通常被省去。

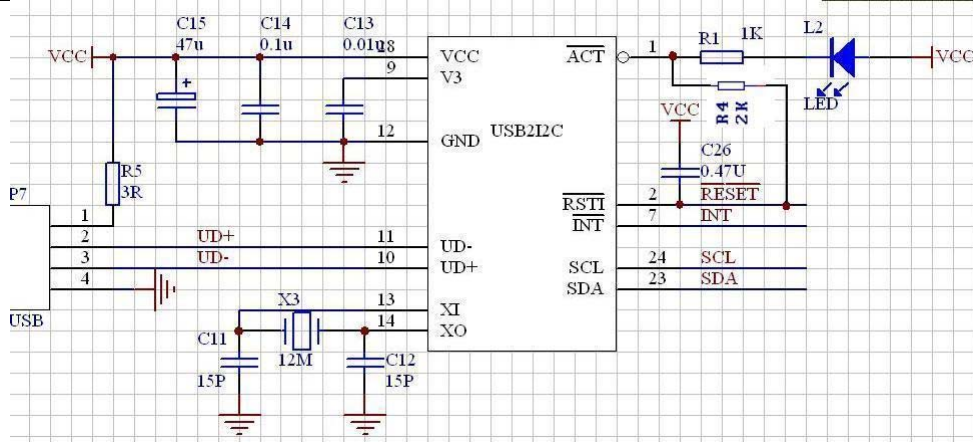


图 1

7.2. I2C总线介绍

I2C总线的主要引脚包括SCL引脚、SDA引脚。SCL用于单向输出同步时钟，开漏输出且内置上拉电阻，SDA用于准双向数据输入输出，开漏输出及输入且内置上拉电阻。

I2C总线的基本操作元素包括：起始位、停止位、位输出、位输入。起始位定义为当SDA为高电平时，SCL输出下降沿（从高电平切换为低电平）。停止位定义为当SDA为高电平时，SCL输入上升沿（从低电平切换为高电平）。位输出定义为当SCL为低电平时，SDA输出位数据，然后SCL输出高电平脉冲。位输入定义为SCL输出高电平脉冲，在下降沿之前从SDA输入位数据。字节输出定义为8个位输出及1个位输入用于应答。字节输入定义为8个位输入及1个位输出用于应答。2线串口的数据输入和输出以字节为单位，每个字节含8个位，高位在前。

USB2I2C支持常用的I2C/IIC设备（如，ATMEL公司的AT24CX系列EEPROM；I2C总线8位并行I/O口扩展芯片PCF8574/JLC1562；I2C接口实时时钟芯片DS1307/PCF8563/SD2000D/M41T80/ME901/ISL1208/；I2C数据采集ADC芯片MCP3221（12bitADC）/ADS1100（16bitADC）/ADS1112（16bitADC）/MAX1238（12bitADC）/MAX1239（12bitADC）；I2C接口数模转换DAC芯片DAC5574（8bitDAC）/DAC6573（10bitDAC）/DAC8571（16bitDAC）/；I2C接口温度传感器TMP101/TMP275/DS1621/MAX6625/，等）之间的通信。

8、上位机端的软件设计

在计算机端的Windows操作系统下，USB2I2C的驱动程序USBIOX.SYS和动态链接库USBIOX.DLL向应用程序提供了应用层接口，包括：设备管理API、I2C总线数据传输API、中断处理API。有关API参数的说明请参考USBIOX.H，主要API如下。

8.1. 设备管理API

`USBIO_OpenDevice`(//打开USB2I2C设备, 返回句柄, 出错则无效

`ULONGiIndex`); //指定USB2I2C设备序号, 0对应第一个设备将USB2I2C作为设备, 使用前必须先打开, 然后才能使用

`USBIO_CloseDevice`(//关闭USB2I2C设备

`ULONGiIndex`); //指定USB2I2C设备序号用完USB2I2C后, 或者应用程序退出前, 应该关闭USB2I2C设备

`USBIO_SetDeviceNotify`(//设定设备事件通知程序

`ULONGiIndex`, //指定USB2I2C设备序号, 0对应第一个设备

`PCHARiDeviceID`, //可选参数, 指向字符串, 指定被监控的设备的ID, 字符串以\0终止

`mPUSB2I2C_NOTIFY_ROUTINEiNotifyRoutine`); //指定设备事件回调程序

用于应用程序监控USB2I2C设备的插拔事件，确保应用程序随时知道USB设备是否存在，防止在USB设备拔出后收发数据，并及时响应USB设备的插入

```
USBIO_GetStatus(//通过USB2I2C直接输入数据和状态，类似的API还有USBIO_GetInput
    ULONGiIndex, //指定USB2I2C设备序号
    PULONGiStatus); //指向一个双字单元, 用于保存状态数据
    获取的状态数据中: 位10对应USB2I2C的INT#引脚, 位11对应USB2I2C的SLC引脚, 位23对应USB2I2C
    的SDA引脚
```

```
USBIO_SetOutput(//设置USB2I2C的I/O方向, 并通过USB2I2C直接输出数据//谨慎使用该API, 防止修改
    I/O方向使输入引脚变为输出导致与其它输出引脚之间短路而损坏
```

```
    ULONGiIndex, //指定USB2I2C设备序号
    ULONG iEnable, //数据有效标志
    ULONG iSetDirOut, //设置I/O方向, 位清0则对应引脚为输入, 位置1则对应引脚为输出
    ULONG iSetDataOut); //输出数据, 如果I/O方向为输出, 那么位数据将通过引脚输出
```

上述的I/O方向和输出数据以32位数据表示, 其中: 位10对应USB2I2C的INT#引脚; 另外, 以下引脚只能输出, 不考虑I/O方向: 位16对应USB2I2C的RESET#引脚, 位18对应USB2I2C的SCL引脚, 位29对应USB2I2C的SDA引脚

8.2. 中断处理API

```
USBIO_SetIntRoutine(//设定中断服务程序
    ULONGiIndex, //指定USB2I2C设备序号
    mPUSBIO_INT_ROUTINEiIntRoutine); //指定中断服务程序, 为NULL则取消中断服务
```

设置USB2I2C的中断服务程序, iIntRoutine是一个符合mPUSBIO_INT_ROUTINE格式的子程序, 当USB2I2C的INT#引脚出现上升沿时, DLL自动调用iIntRoutine, 并向其提供一个引脚状态参数, 引脚状态参数中, 位为1则说明对应的引脚为高电平, 位为0则说明对应的引脚为低电平。

例如: 主程序

```
main{
    .....
    USBIO_OpenDevice(0); //打开设备, 针对0#设备, 如果有多个, 可以计数
    USBIO_SetIntRoutine(0, myInterruptEvent); //设置中断服务程序
    ..... 读写数据, 或者在接收到中断服务程序的通知后处理中断
    USBIO_CloseDevice(0); //用完后关闭设备
}
```

中断服务程序, 当USB2I2C的INT#引脚出现上升沿时, DLL会自动调用该子程序

```
voidCALLBACKmyInterruptEvent(unsignedlongPinStatus)
    {if(PinStatus&mStateBitERR)printf(“发生中断时ERR#引脚为高电平”);
    elseprintf(“发生中断时ERR#引脚为低电平”);..... 自己处理或者通知主程序处理
}
```

8.3. I2C总线数据传输API

```
USBIO_ReadI2C(//从两线串口读取一个字节数据, 仅适用于7位地址的设备
    ULONG iIndex, //指定USB2I2C设备序号
    UCHAR iDevice, //低7位指定设备地址
    UCHAR iAddr, //指定数据单元的地址
    PCHAR oByte); //指向一个字节单元, 用于保存读取的字节数据
```

USBIO_WriteI2C(//向两线串口写入一个字节数据, 仅适用于7位地址的设备

ULONGiIndex, //指定USB2I2C设备序号
UCHARiDevice, //低7位指定设备地址
UCHARiAddr, //指定数据单元的地址
UCHARiByte); //待写入的字节数据

USBIO_WriteRead(//执行数据流命令, 先输出再输入

ULONGiIndex, //指定USB2I2C设备序号
ULONGiWriteLength, //写长度, 准备写出的长度
PVOIDiWriteBuffer, //指向一个缓冲区, 放置准备写出的数据
ULONGiReadStep, //准备读取的单个块的长度, 总长度为(iReadStep*iReadTimes)
ULONGiReadTimes, //准备读取的次数
PULONGoReadLength, //指向长度单元, 返回后为实际读取的长度
PVOIDoReadBuffer); //指向一个足够大的缓冲区, 用于保存读取的数据
先输出数据再输入数据, 执行数据流命令, 适用于同步串口等。

USBIO_SetStream(//设置同步串口流模式

ULONGiIndex, //指定USB2I2C设备序号
ULONGiMode); //指定模式, 见下行
//位1位0: I2C速度/SCL频率, 00=低速20KHz, 01=标准100KHz, 10=快速400KHz, 11=高速750KHz, 其它保留, 必须为0

USBIO_StreamI2C(//处理两线串口的数据流, 适用于所有两线串口的设备

ULONGiIndex, //指定USB2I2C设备序号
ULONGiWriteLength, //准备写出的数据字节数
PVOIDiWriteBuffer, //指向缓冲区, 放置准备写出的数据, 首字节是设备地址及读写位
ULONGiReadLength, //准备读取的数据字节数
PVOIDoReadBuffer); //指向缓冲区, 返回后是读入的数据对两线串口设备进行操作。

例如, 从24C256中3200H开始的地址读出256字节的数据:

UCHAROutBuf[5], InBuf[300]; //待写数据缓冲区, 读出数据缓冲区
OutBuf[0]=0xA1; OutBuf[1]=0x32; OutBuf[2]=0x00; //待写数据: 设备地址及单元地址
USBIO_StreamI2C(0, 3, OutBuf, 256, InBuf); //针对0#设备处理两线串口的数据流

USBIO_ReadEEPROM(//从EEPROM中读取数据块, 速度约56K字节

ULONGiIndex, //指定USB2I2C设备序号
EEPROM_TYPEiEepromID, //指定EEPROM型号
ULONGiAddr, //指定数据单元的地址
ULONGiLength, //准备读取的数据字节数
PUCHARoBuffer); //指向一个缓冲区, 返回后是读入的数据读写EEPROM的API支持从24C01到24C16和从24C32到24C4096的各种型号的EEPROM存储器。

USBIO_WriteEEPROM(//向EEPROM中写入数据块

ULONGiIndex, //指定USB2I2C设备序号
EEPROM_TYPEiEepromID, //指定EEPROM型号
ULONGiAddr, //指定数据单元的地址
ULONGiLength, //准备写出的数据字节数
PUCHARiBuffer); //指向一个缓冲区, 放置准备写出的数据

有关USBIOX.DLL中各个API的使用实例请参考USB2I2C评估板资料中的各个源程序及例子。