

USBMOD6 高速 USB2.0 传输模块使用说明

——V2.2 版

简介：

USBMOD6 模块是北京华诺铭科电子科技有限公司设计的一款高速 USB2.0 数据传输模块，有效数据实时稳定传输速度达 32M 字节/秒。更高传输速度可以定制，最高可达 44M 字节/秒。该模块完全屏蔽了复杂的 USB2.0 协议，将电脑的 USB2.0 接口转换为一个透明的并行总线数据传输接口和一个普通的串口。可以方便的跟 FPGA、CPLD、DSP、ARM、单片机等外部控制器连接。在电脑主机上，提供专用动态库，方便被各种编程语言调用。FPGA 程序和 VC 应用程序开源设计，利用该模块，您可以在最短的时间内完成高速的 USB2.0 数据传输功能。

**功能特点：**

- FPGA 程序和 VC 应用程序开源设计，用户可以根据自己的需要修改源码；
- 提供多种应用实例源码，如普通收发实例、AD 采集实例、图像采集实例、逻辑分析仪实例等；
- 独家提供稳定高效的 SRAM 虚拟 FIFO 技术，可将普通 SRAM 虚拟成高速大容量的 FIFO 来使用，免去使用两片 SRAM 做乒乓缓存，即可以降低成本，又可以使设计变得更简单；
- 实时传输速度快，有效数据可达 32M 字节/秒，更高速度可以定制，最高可达 44M 字节/秒；
- 数据双向传输，可以同时上传和下传；
- 硬件接口简单，可以方便的跟 FPGA、CPLD、DSP、ARM、单片机等控制器连接；

- 大缓存硬件设计，容量高达 512K 字节；
- 外扩独立的全双工异步串口，串口波特率可设，最大波特率为 115200bps；
- 提供专用动态库，可以方便的被各种高级编程语言调用，并提供相应的动态库调用实例；
- 提供 FPGA、单片机等控制器的接口通讯源代码；
- 模块电路板采用多层板设计，采用全板沉金的表面处理工艺，信号线等长设计，并用专业的电路板信号仿真软件对信号进行仿真，充分保证了信号传输的稳定可靠；
- 模块尺寸：72×50×15mm。

模块应用：

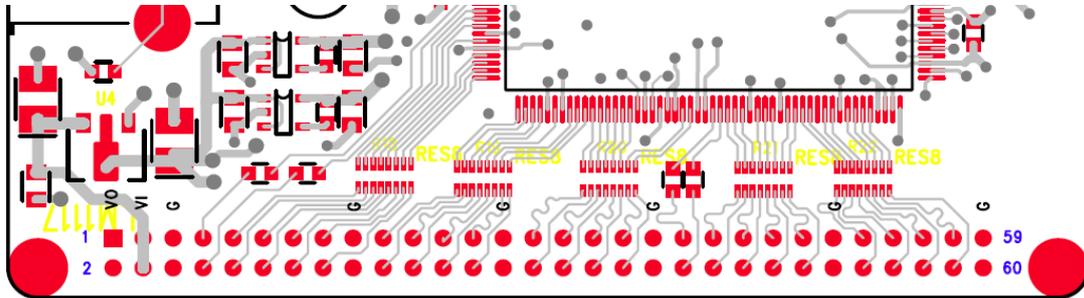
- USB 数据采集系统；
- USB 逻辑分析仪；
- USB 高速图像采集系统；
- USB 数字存储示波器；
- USB 工业控制系统；

产品清单：

- 1、USBMOD6 模块一块；
- 2、USB 双层屏蔽线一条；
- 3、光盘 1 张（内含动态库、相关源码、使用说明、原理图和 PCB 库、工具软件等）；

管脚说明：

USBMOD6 模块的各引脚的功能描述如下：



管脚号	信号名称	信号描述
1	+5Vout	USB 口电源输出管脚，输出电压为：5.0V，电流 \leq 500mA。
2	+5Vout	
3	+5Vin	USBMOD6 电源输入管脚，输入电压范围：4.5V~5.5V。 不能悬空，必须外接电源，可以将该引脚跟 1、2 引脚相连。
4	+5Vin	
5	GND	电源地。
6	GND	
7	PE0	备用管脚。悬空，不接任意信号。
8	PE1	备用管脚。悬空，不接任意信号。
9	PE2	备用管脚。悬空，不接任意信号。
10	PE3	备用管脚。悬空，不接任意信号。
11	PE4	备用管脚。悬空，不接任意信号。
12	PE5	备用管脚。悬空，不接任意信号。
13	PE6	备用管脚。悬空，不接任意信号。
14	PE7	备用管脚。悬空，不接任意信号。
15	PE8	备用管脚。悬空，不接任意信号。
16	PE9	备用管脚。悬空，不接任意信号。
17	GND	电源地。
18	GND	
19	PD0	备用管脚。悬空，不接任意信号。
20	PD1	备用管脚。悬空，不接任意信号。
21	PD2	备用管脚。悬空，不接任意信号。
22	PD3	备用管脚。悬空，不接任意信号。
23	PD4	备用管脚。悬空，不接任意信号。
24	PD5	备用管脚。悬空，不接任意信号。
25	PD6	备用管脚。悬空，不接任意信号。
26	PD7	备用管脚。悬空，不接任意信号。
27	GND	电源地。
28	GND	
29	RXD/PC0	异步串口输入引脚，为 3.3V LVTTTL 电平。
30	TXD/PC1	异步串口输出引脚，为 3.3V LVTTTL 电平。

31	DIR/PC2	数据总线方向控制引脚。高电平为输出，低电平为输入。
32	PC3	备用管脚。悬空，不接任意信号。
33	WR/PC4	数据总线写信号，高电平有效。当 WRfull 电平为低时，可以向 USBMOD6 模块写入要发送到电脑的数据。
34	WRfull/PC5	数据总线写满信号，高电平有效。外部控制器向 USBMOD6 模块写数据，当模块内的数据未能被电脑及时取走，使模块内部的缓存被写满时，该信号变为高电平，此时不能再向模块写数据，否则会造成数据丢失。
35	RD/PC6	数据总线读信号，高电平有效。当 RDempty 电平为低时，可以从模块读出电脑发送出来的数据。
36	RDempty/PC7	数据总线读空信号，高电平有效。外部控制器从 USBMOD6 模块读数据，当电脑发送到模块内部的数据被读完时，该信号变为高电平，此时不能再发读信号，否则会造成数据出错。
37	GND	电源地。
38	GND	
39	CLK-	备用管脚。悬空，不接任意信号。
40	CLK+	备用管脚。悬空，不接任意信号。
41	D0/PB0	D0，数据总线最低位 (LSB)。
42	D1/PB1	D1，数据总线。
43	D2/PB2	D2，数据总线。
44	D3/PB3	D3，数据总线。
45	D4/PB4	D4，数据总线。
46	D5/PB5	D5，数据总线。
47	D6/PB6	D6，数据总线。
48	D7/PB7	D7，数据总线。当数据总线宽度为 8 时，该位为最高位 (MSB)。
49	GND	电源地。
50	GND	
51	D8/PA0	D8，数据总线。当数据总线宽度为 8 时，该位无效。
52	D9/PA1	D9，数据总线。当数据总线宽度为 8 时，该位无效。
53	D10/PA2	D10，数据总线。当数据总线宽度为 8 时，该位无效。
54	D11/PA3	D11，数据总线。当数据总线宽度为 8 时，该位无效。
55	D12/PA4	D12，数据总线。当数据总线宽度为 8 时，该位无效。
56	D13/PA5	D13，数据总线。当数据总线宽度为 8 时，该位无效。
57	D14/PA6	D14，数据总线。当数据总线宽度为 8 时，该位无效。
58	D15/PA7	D15，数据总线。当数据总线宽度为 8 时，该位无效。 当数据总线宽度为 16 位，该位为最高位 (MSB)。
59	GND	电源地。
60	GND	

备注： 1、如果需要模块独立供电，则需要将 1、2 引脚跟 3、4 引脚短接起来。
2、所有数字引脚信号均为 3.3V LVTTL 电平。

DC 电气参数:

信号名称	说明	最小值	典型值	最大值	单位
VI	模块输入电源电压	4.5	5.0	5.5	V
Iin	模块输入电源电流	200	300	400	mA
VO	模块输出电源电压	4.0	5.0	5.5	V
Iout	模块输出电源电流	100	200	500	mA
VOL	IO 输出低电平电压	0	0.05	0.3	V
VOH	IO 输出高电平电压	3.0	3.3	3.6	V
VIL	IO 输入低电平电压	0	0.05	0.8	V
VIH	IO 输入高电平电压	2.7	3.3	5.0	V

AC 电气参数:

信号名称	说明	最小值	典型值	最大值	单位
T1	DIR 改变前, WR 或 RD 由有效变成无效的时间			75	ns
T2	DIR 改变前, WRFULL 或 REMPTY 由有效变成无效的时间			50	ns
T3	DIR 改变后, WRFULL 或 REMPTY 由无效变成有效的的时间			50	ns
T4	DIR 改变后, WR 或 RD 由无效变成有效的的时间			35	ns
T5	DIR 改变后, 数据总线方向切换的时间			35	ns
T10	读数据时钟高电平持续时间	20			ns
T11	读数据时钟周期	40			ns
T12	数据呈现在数据总线上到时钟上升沿的时间	5			ns
T13	读数据输出保持时间			35	ns
T14	从读最后一个数据时钟上升沿到输出 REMPTY 有效的的时间			35	ns
T15	REMPTY 无效到读时钟上升沿时间	5			ns
T20	写数据时钟高电平持续时间	20			ns
T21	写数据时钟周期	40			ns
T22	写数据有效到写数据时钟上升沿的时间	0			ns
T23	写数据时钟上升沿后数据保持时间	25			ns
T24	模块内部 FIFO 中, 最后一个数据 n 被写入时, 从写数据时钟上升沿到输出 WRFULL 有效的的时间			35	ns

备注: 更多波形数据细节可以参考 FPGA 测试工程 UsbDev_FPGA_Q91_Test 下的 SignalTap II Logic Analyzer 文件 stp.stp (20MHz 工作时钟, 200MHz 采样时钟)。

数据总线输入输出切换时序图：

V2.2 版本 USBMOD6 模块内部数据发送和接收采用两个独立的缓存，出厂默认程序 (UsbDev_FPGA_Q91_sram_fifo) 的缓存配置为：模块至 PC 采用大缓存 (SRAM 缓存, 512KBytes)，而 PC 至模块则采用小缓存 (FPGA 片上 RAM, 8KBytes)。当客户需要交换这两种缓存的配置时，直接下载对应的 FPGA 程序 (UsbDev_FPGA_Q91_sram_fifo_out) 即可。当 DIR 切换时，不会造成缓存内部的数据丢失。

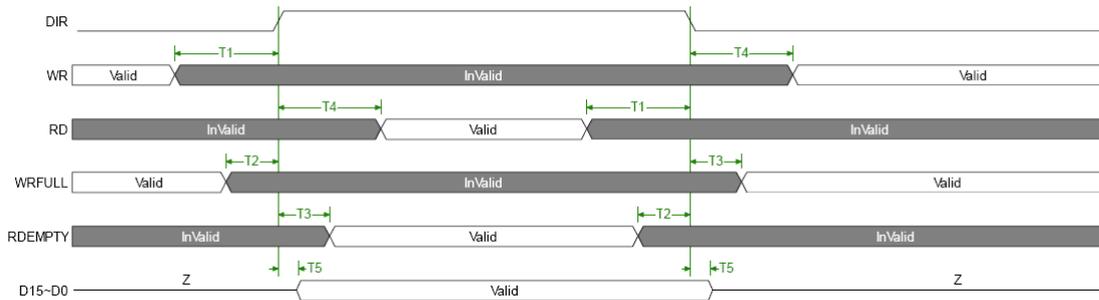


图 1 DIR 切换时序图

写时序图：

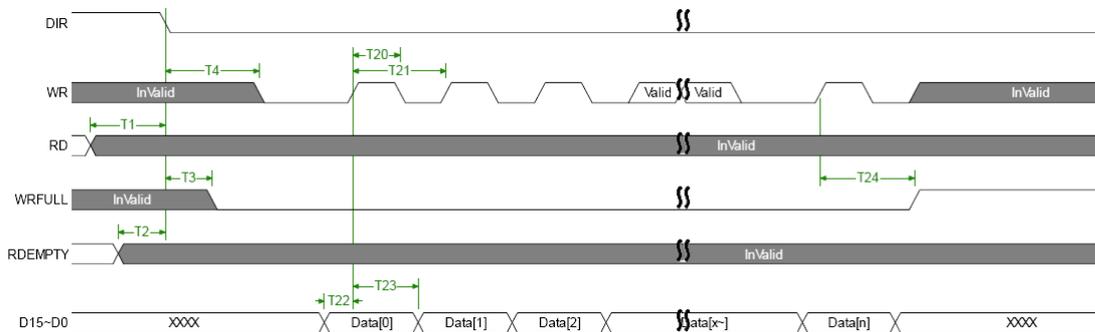


图 2 写时序图

读时序图：

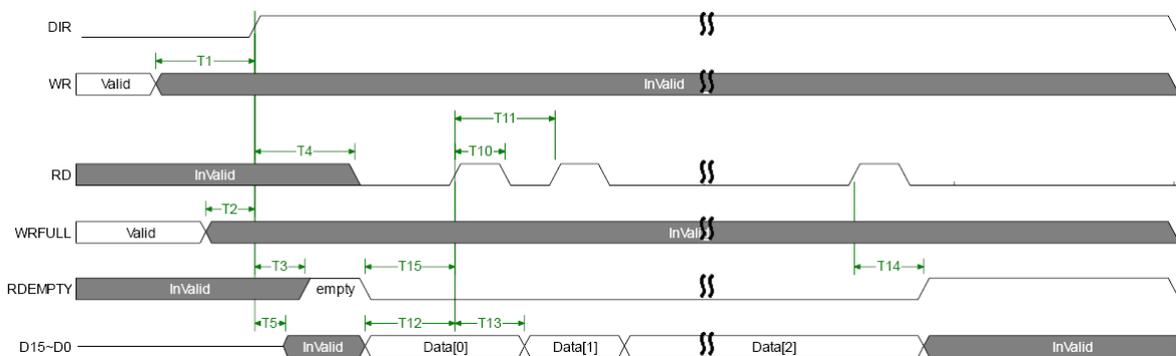


图 3 读时序图

典型电路：

USBMOD6 自供电电路如图 4 所示：

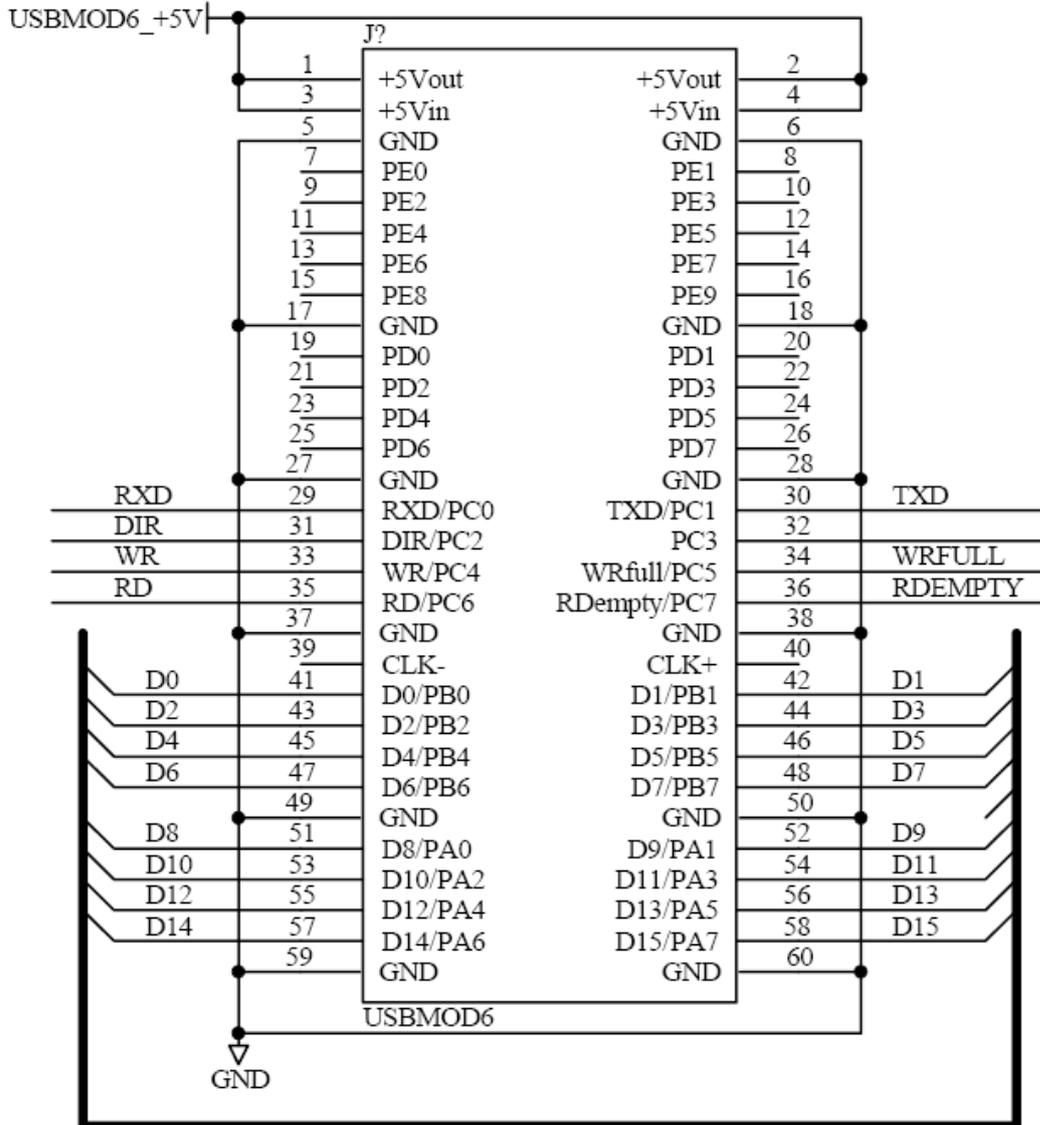


图 4 模块自供电电路

USBMOD6 外部供电典型电路如图 5 所示：

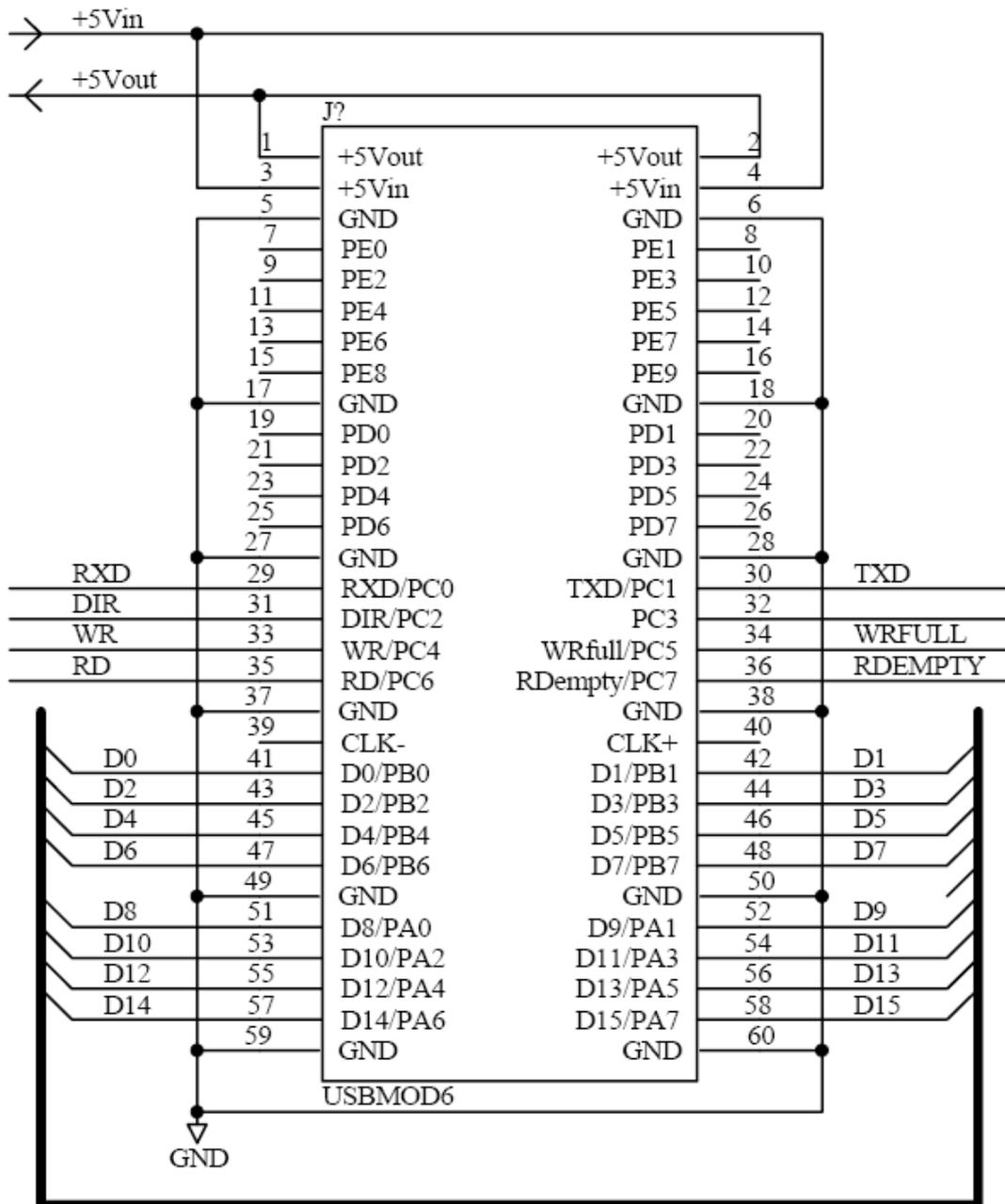


图 5 模块外部供电电路

- 备注：
- 1、+5Vout 为 USB 总线电源，实际输出电流大小视电脑而定，最大 500mA。
 - 2、当数据总线为 8 位总线时，只接 D7~D0，而 D15~D8 悬空不接。
 - 3、如果串口功能没使用，建议将 RXD 引脚通过 10K 欧电阻上拉到 +3.3V 电源，TXD 引脚悬空。
 - 4、图中没有使用的引脚都保持悬空。

软件接口说明:

```
DWORD UsbDev_EnumNumber( void );
```

//返回连接到当前系统的 USBMOD6 模块的个数。

//如果个数为 0 说明没有 USBMOD6 模块连接到当前系统。

```
DWORD UsbDev_Init( DWORD DevSel);
```

//如果成功与指定的设备建立连接成功则返回 1 或者 2，如果连接失败则返回 0。

//如果返回 1，

//则说明模块工作于全速模式模块可能被接到一个 USB1.1，

//总线上这时将不能发挥模块的高速传输性能。

//如果返回 2，

//则说明模块工作于高速模式，

//设备被连接到一个 USB2.0 以上版本的 USB 总线并被成功识别

//入口参数////////////////////////////////////

//DevSel 第几个连接到当前系统的模块，

//DevSel=0 表示第 1 个连接到当前系统的模块。

```
void UsbDev_Close(DWORD DevSel);
```

//断开与模块的连接。

//DevSel 第几个连接到当前系统的模块，

//DevSel=0 表示第 1 个连接到当前系统的模块。

```
BOOL UsbDev_OnHighSpeed(DWORD DevSel);
```

//模块是否工作于高速模式，

//如果模块连接到一个 USB2.0 总线上，而且成功的枚举成高速工作模式，

//则返回 TRUE，否则返回 FALSE。

//DevSel 第几个连接到当前系统的模块，

//DevSel=0 表示第 1 个连接到当前系统的模块。

```
DWORD UsbDev_Write(DWORD DevSel, unsigned char *pBuffer, DWORD Size,  
                    DWORD Address, BOOL AddressInc, DWORD TimeOut_ms);
```

//USBMOD6 作为主设备，主动向外部设备发起写操作信号（USBMOD6 为输出）。

//写成功返回实际发送的数据大小，失败返回 0。

//DevSel 第几个连接到当前系统的模块，

//DevSel=0 表示第 1 个连接到当前系统的模块。

//pBuffer，写数据缓存。

//Size，写数据大小，最大为 65024（127*512 字节）（16 位地址范围）。

//Address，USBMOD6 的写起始地址，支持 16 位的地址范围。

//AddressInc，USBMOD6 向外发起写操作时，地址是否递增，

//即每发一个数，地址是否加一，

//递增为 1，保持不变为 0。

//TimeOut_ms，超时退出，当程序执行到指定时间未完成时，

```
//函数强行退出并返回实际完成的数据量。  
//大小计算，一般 64K 字节的数据对应 10ms，  
//当然实际应用中根据情况调节，可大可小。  
//当前版本 V2.2 不支持 Address 、 AddressInc 、 TimeOut_ms 这三项功能，  
//需要将它们都设为 0。
```

```
DWORD UsbDev_Read(DWORD DevSel, unsigned char *pBuffer, DWORD Size,  
                  DWORD Address, BOOL AddressInc, DWORD TimeOut_ms);  
//USBMOD6 作为主设备，主动向外部设备发起读操作信号（USBMOD 为输入）。  
//读成功返回实际接收的数据大小，失败返回 0。  
//DevSel 第几个连接到当前系统的模块，  
//DevSel=0 表示第 1 个连接到当前系统的模块。  
//pBuffer，读数据缓存，实际大小至少比 Size 大 512 字节。  
//Size，读数据大小，最小为 512，最大为 65024（127*512 字节）（16 位地址范围）。  
//Address，USBMOD6 的读起始地址，支持 16 位的地址范围。  
//AddressInc，USBMOD6 向外发起读操作时，地址是否递增，  
//即每读一个数，地址是否加一，  
//递增为 1，保持不变为 0。  
//TimeOut_ms，超时退出，当程序执行到指定时间未完成时，  
//函数强行退出并返回实际完成的数据量。  
//大小计算，一般 64K 字节的数据对应 10ms，  
//当然实际应用中根据情况调节，可大可小。  
//当前版本 V2.2 不支持 Address 、 AddressInc 、 TimeOut_ms 这三项功能，  
//需要将它们都设为 0。
```

```
BOOL UsbDev_SetBaudRate(DWORD DevSel, DWORD BaudRate );  
//设定串口波特率。  
//成功返回 TRUE，失败返回 FALSE。  
//DevSel 第几个连接到当前系统的模块，  
//DevSel=0 表示第 1 个连接到当前系统的模块。  
//BaudRate 串口波特率设定值，默认为 9600 bps，
```

```
BOOL UsbDev_UartTx(DWORD DevSel, unsigned char *pBuffer, DWORD Size);  
//写数据到串口，成功返回 TRUE，失败为 FALSE。  
//DevSel 第几个连接到当前系统的模块，  
//DevSel=0 表示第 1 个连接到当前系统的模块。  
//pBuffer，为串口发送数据缓存指针。  
//Size，串口发送数据大小。
```

```
DWORD UsbDev_UartRx(DWORD DevSel, unsigned char *pBuffer, DWORD Size);  
//读取串口数据，成功返回实际接收数据大小，否则为 0。  
//DevSel 第几个连接到当前系统的模块，  
//DevSel=0 表示第 1 个连接到当前系统的模块。
```

//pBufffer, 为串口接收数据缓存指针。

//Size, 串口准备接收的数据大小。

```
BOOL UsbDev_SetTest(DWORD DevSel, unsigned char cDirOut );
```

//设定测试模式。

//只有测试模块的时候调用，正常使用时不用，也不用关心它处于什么状态，

//它只控制 FPGA 程序中测试部分的功能，不影响正常的数据传输。

//成功返回 TRUE，失败返回 FALSE。

//DevSel 第几个连接到当前系统的模块，

//DevSel=0 表示第 1 个连接到当前系统的模块。

//cTest, 测试模式设定：0—>电脑为输入模式，1—>电脑为输出模式。

外形结构图:

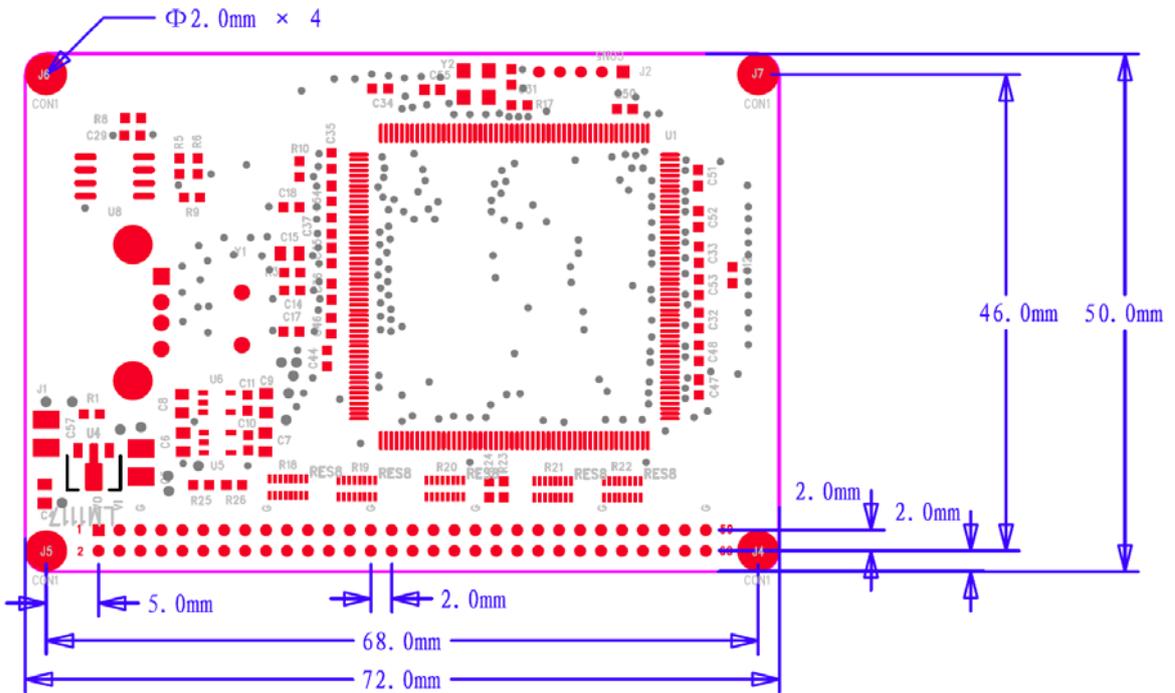


图 6 尺寸及顶视图

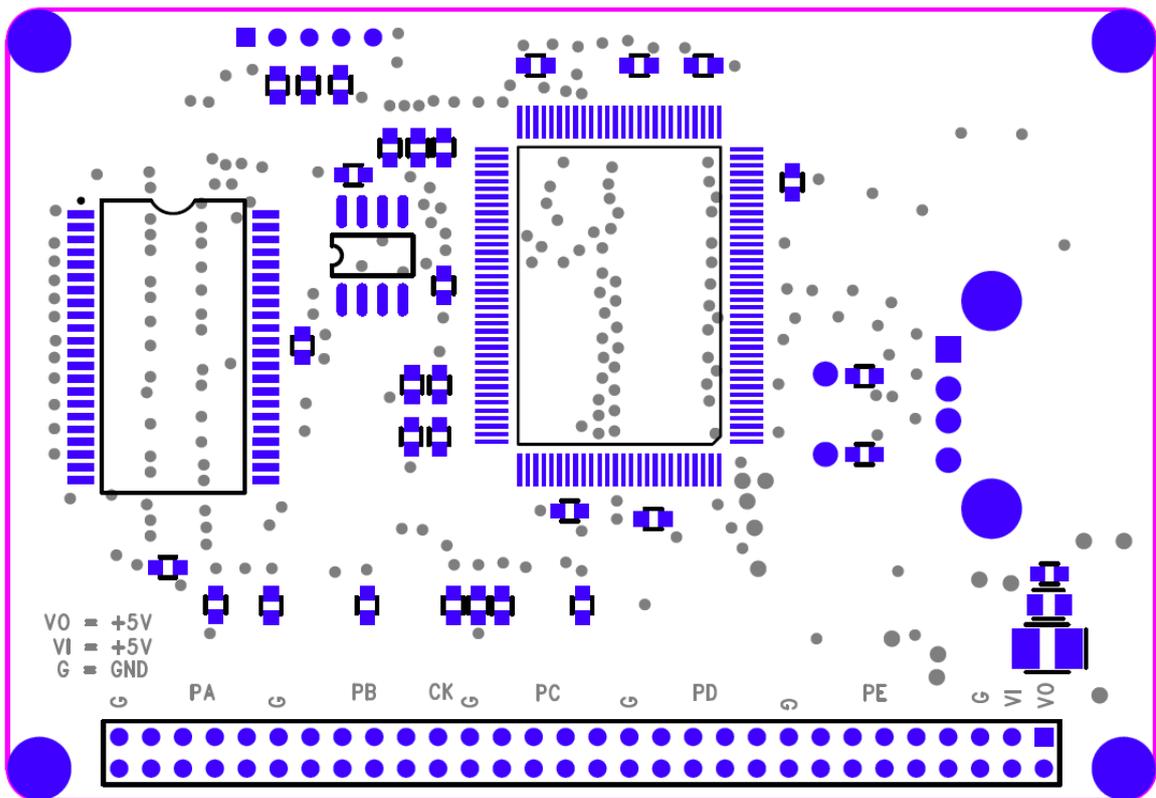
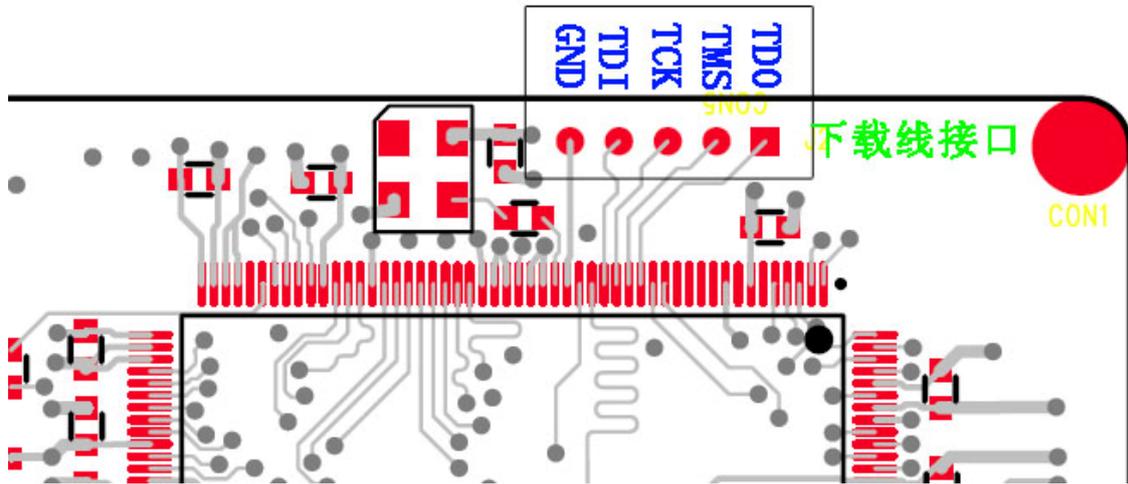


图 7 底视图

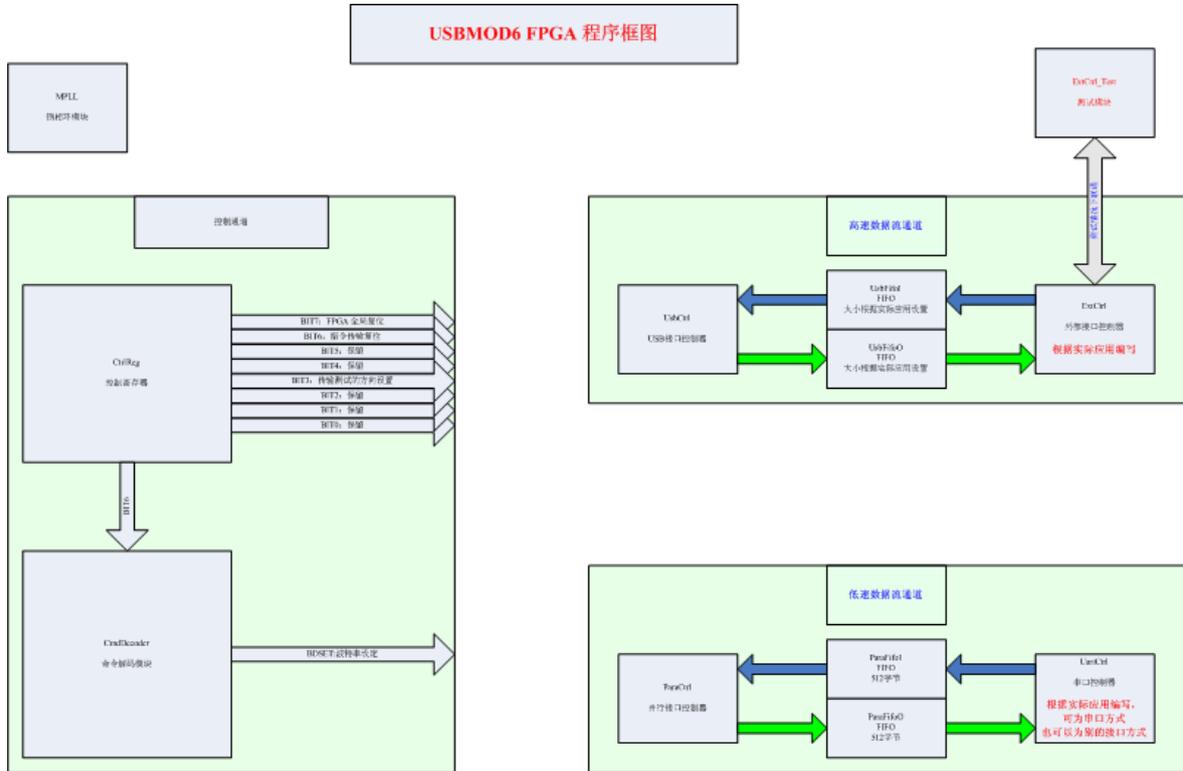
附录一



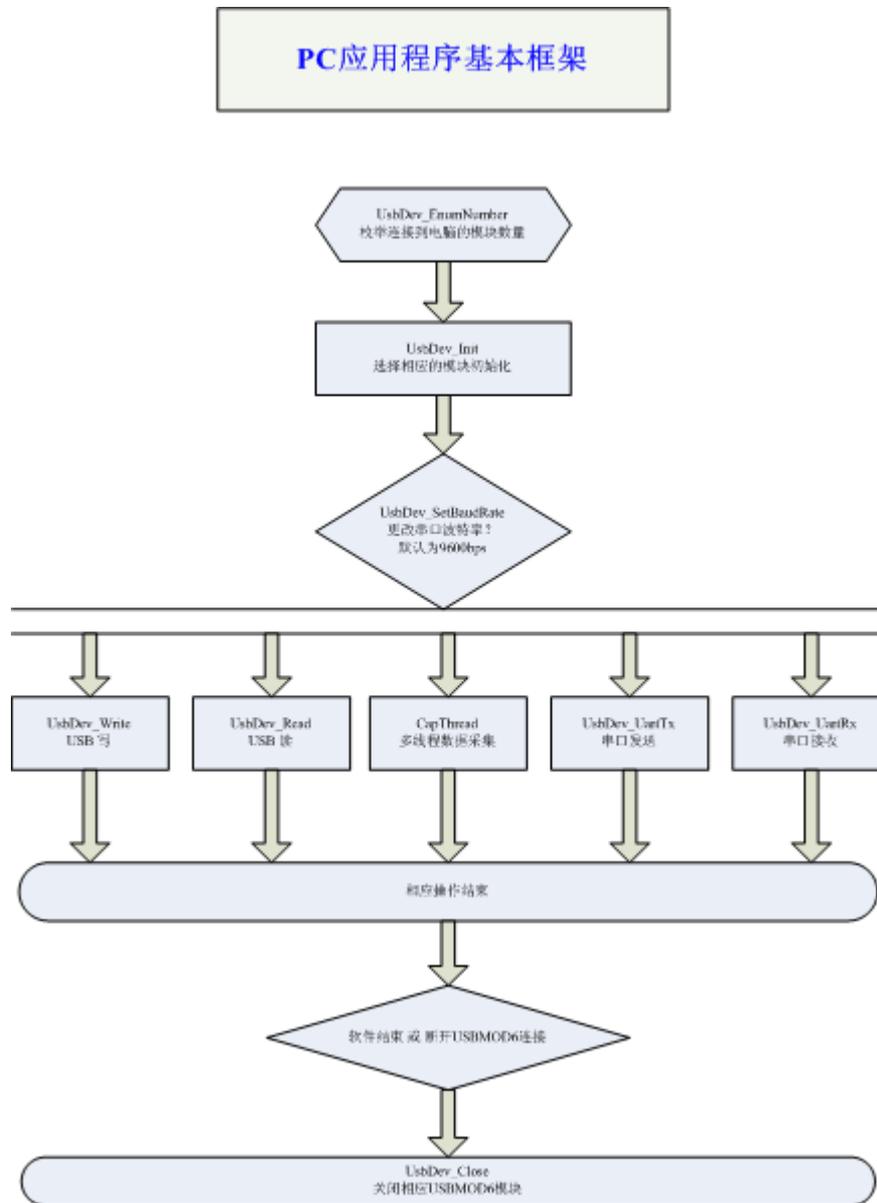
下载线接口定义图

由于下载线接口没有预留电源引脚，所以需要下载线自己跟自己供电。

附录二

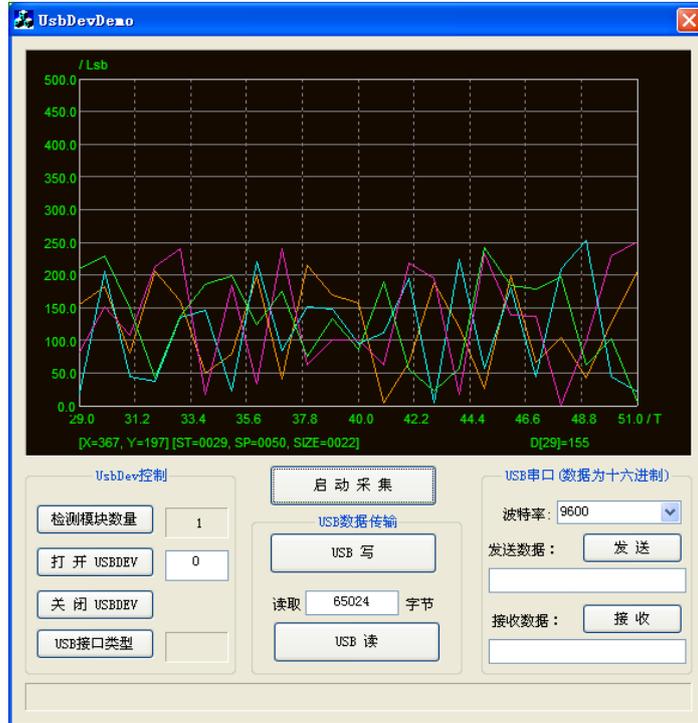


附录三

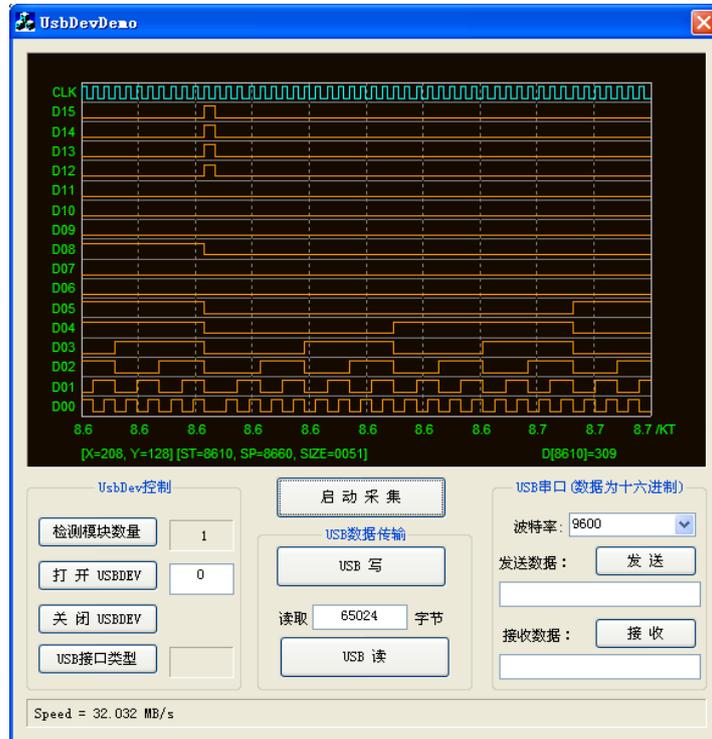


附录四

VC 部分示例贴图



4 通道 AD 采样示例贴图



16 通道逻辑分析仪示例贴图

北京华诺铭科电子科技有限公司

地址：北京市海淀区中关村大街28-1号中海园电子市场BF-198

（位于 海淀剧院旁 或者 中发电子大厦对面）

电话：010-82534219

传真：010-82534219

网址：<http://www.huanor.com>

技术支持：huanor_supports@hotmail.com

销售：huanor_sales@hotmail.com