

TE3000 激光控制卡

使用说明书

2006/05

一 概述

TE3000 是一款针对激光控制行业设计的多功能 DA 输出卡 ,为用户提供了 4 路模拟信号输出通道 ,2 路 PWM 数字信号输出通道 ,26Bit TTL 数字量输入通道, 26Bit TTL 数字量输出通道。内建的 32Bit “解码引擎”可以主动从 PC 的内存中取得指令,并以每秒 10 万次的速度解析用户指令,从而实现高速,连续,精细的控制输出。

二、性能和技术指标

2.1 性能

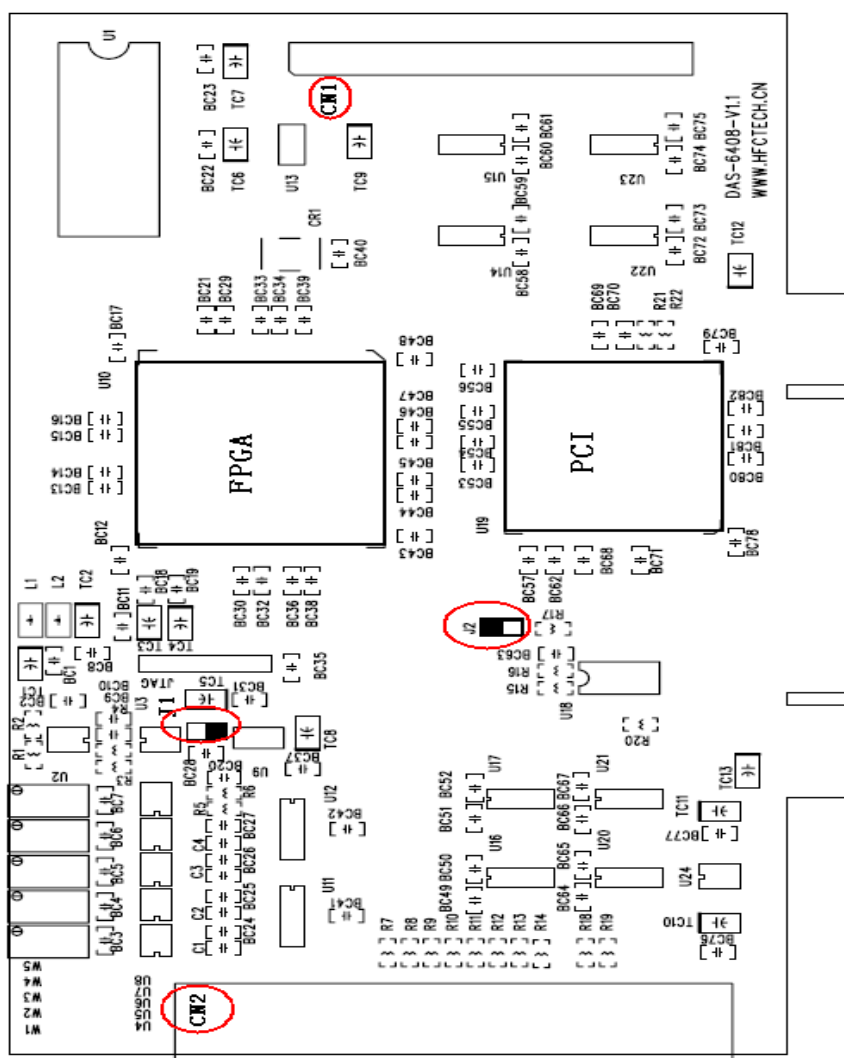
- 4 路模拟信号输出通道
- 模拟信号输出分辨率 16Bit
- 模拟信号输出范围 $\pm 10V$ 或 $\pm 5V$
- PWM 输出可调范围 16bit
- PWM 输出可调精度 0.1 μ S
- PWM 输出首脉冲抑制
- 26Bit DI TTL 兼容数字量输入
- 26Bit DO TTL 兼容数字量输出,可位操作。
- 内建 32Bit 解码引擎
- 解码器缓存 4K BYTE
- 解码指令周期 100nS
- PCI 总线兼容 PCI2.2 规范
- 5V, 3.3V PCI 环境兼容
- PCIX 环境兼容 (33M)
- PCI 主控传输方式, BUSMASTER
- 链表 DMA, Scatter /gather DMA
- 兼容 Windows2000, XP
- VC 兼容 DLL, LIB

2.2 模拟信号输出技术指标

- | | |
|-------------|-----------------------|
| • 模拟信号输出通道: | 4 |
| • 模拟信号输出分辨率 | 16 位 |
| • 模拟信号输出范围: | $\pm 10 V$ 或 $\pm 5V$ |
| • 数模转换器件 | DAC5547 |
| • 连续控制周期 | 100ns |

- 建立时间: 2uS
 - 精度: 优于 $\pm 0.003\%$ 满量程
 - 线性度: $\pm 1/2$ LSB
 - 复位状态: 0 输出
 - 输出负载能力: $\pm 5\text{mA}$
- 2.3 PWM 输出
- PWM 输出通道: 2 路反相
 - 脉冲周期范围: 0.1uS – 6.5535mS
 - 高电平输出范围: 0.0uS – 6.5535mS
 - 分辨率: 0.1uS
 - 建立时间: 0.1uS
 - 输出电平: TTL 兼容
 - 低电平输出: 灌电流 20mA 时, 最大输出 0.5 V
 - 高电平输出: 源电流 20mA 时, 最小输出 2.4V
 - 复位输出: PWMA 输出 0
- 2.4 数字量输出
- 通道: 26 路
 - 输出电平: TTL 兼容
 - 低电平输出: 灌电流 20mA 时, 最大输出 0.5 V
 - 高电平输出: 源电流 20mA 时, 最小输出 2.4V
 - 复位输出: 输出 0
- 2.5 数字量输入
- 通道: 26 路
 - 输入电平: TTL 兼容
 - 低电平输入: 最大 0.8V
 - 高电平输入: 最小 2.0V
- 2.6 其他参数
- 工作温度 : 0 — 60
 - 储存温度 : -40 — 80
 - 最大功耗 : 500mA

三. 板卡布局



接口定义:

CN2 : DB37 接口, 模拟量输出, PWM , DIO 输出

CN1 : IDC40 接口, 辅助数字接口 EDI0

JP1 : 选择模拟量输出范围

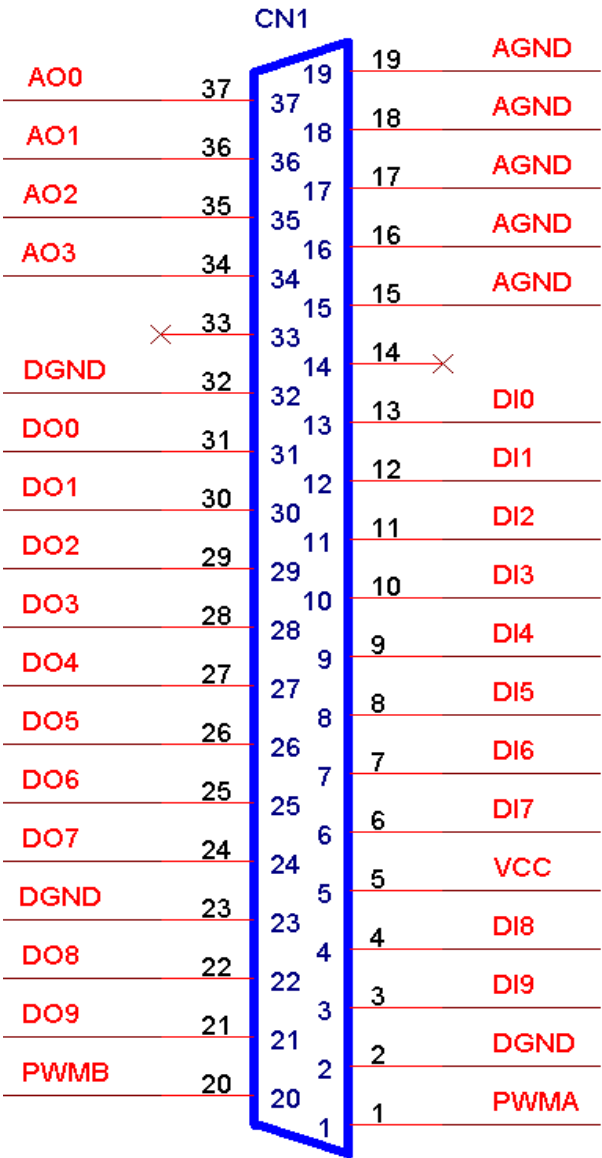
OPEN ± 10 V : CLOSE ± 5 V

JP2 : 预激励 PWM 输出控制

OPEN 无输出 : CLOSE 有输出

四、引脚描述

4.1 CN2



模拟量输出引脚 : AO (0..2)
数字量输出引脚 : DO (0..9)
数字量输入引脚 : DI (0..9)
PWM 输出引脚 : PWMA
开激光 (LaserON) : PWMB
数字地 : DGND
模拟地 : AGND

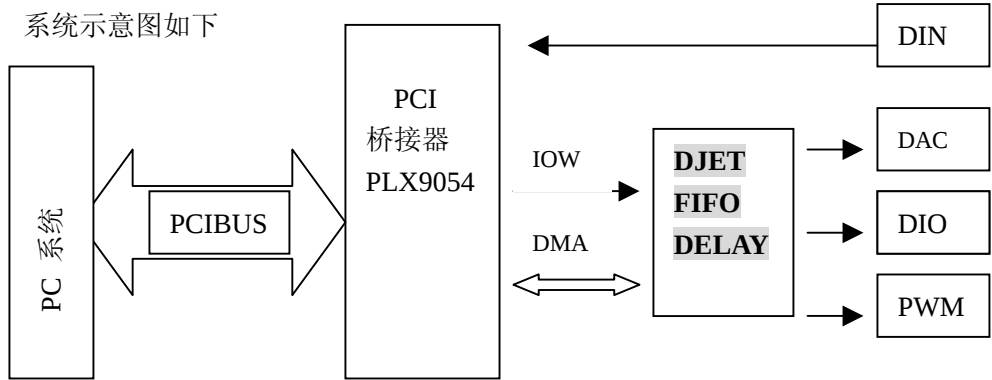
4. 2 辅助数字量引脚 CN2

VCC	1	2	VCC
EDO0	3	4	EDI0
EDO1	5	6	EDI1
EDO2	7	8	EDI2
EDO3	9	10	EDI3
EDO4	11	12	EDI4
EDO5	13	14	EDI5
EDO6	15	16	EDI6
EDO7	17	18	EDI7
EDO8	19	20	EDI8
EDO9	21	22	EDI9
EDO10	23	24	EDI10
EDO11	25	26	EDI11
EDO12	27	28	EDI12
EDO13	29	30	EDI13
EDO14	31	32	EDI14
EDO15	33	34	EDI15
GND	35	36	GND
GND	37	38	GND
GND	39	40	GND

数字输出	:	EDO【0: 15】
数字输入	:	EDI0【0: 15】
电源	:	VCC=5V
数字地	:	DGND

五. 工作原理:

系统示意图如下



TE3000 的工作核心是在 FPGA 中内建的 32bit 解码引擎 (DJET)。DJET 以 100ns 的速度解析用户给定的指令，实现 DA 的输出，数字输出，PWM 输出，延时等功能。用户应用程序通过 DMA 传输或直接命令方式同 DJET 进行数据沟通。**直接命令**是由应用程序通过 IO 写的方式通知 DJET，DJET 会立刻执行直接命令，如果 DJET 正在 DMA 传输方式，会在结束当前指令后暂停 DMA 指令的读取，立刻执行直接命令；DMA 传输方式是一种智能的控制方式，应用程序设置好相应的参数后 DJET 会自动从 PC 的内存中取出指令；DMA 传输方式时指令大小可以是无穷大，也可以是指定的内存块的大小。

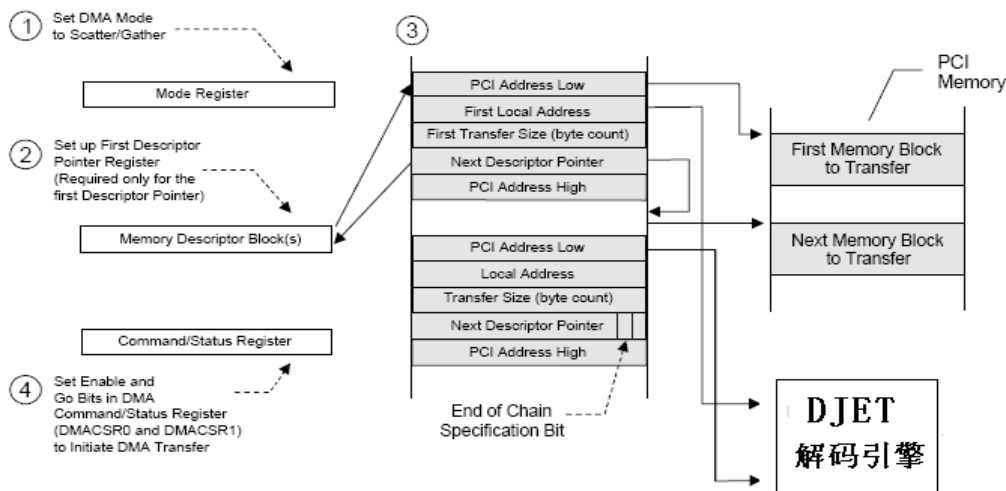
我们为客户提供简单函数接口来完成相应的控制：

TE3000_writeIoPort :实现**直接命令**控制。

TE3000_SglTransfer :实现块 DMA 传输控制。

TE3000_ShuttleTransfer:实现循环 DMA 传输控制

· 关于本卡片设计使用主控链表式 DMA 的介绍



Windows 操作系统的管理内存机制使的应用程序很难在运行时申请到连续的大块的物理内存，由于 DMA 是直接对物理内存操作，如果不用链表 DMA 方式会造成每次 DMA 数据块大小受连续内存块大小限制。链表 DMA 方式是使用链表来管理不连续的物理内存，从而使应用程序在运行时可以使用整个空闲内存，提高内存的使用效率，链表是由驱动程序设置，通过硬件自动管理，应用程序不必关心实现细节。只需用数组或 malloc 申请需要的内存大小。(链表管理最大内存值受限于注册表 TE3000 项目 // MaxSglTransferSize，你需要修改此项以申请合适的缓冲区)。

如果我们将链表首尾相连，就可以实现连续的 DMA 传输，传输的数据大小可以是无穷大，我们提供的函数会在指定的块传输完成后，产生一个事件，应用程序可以在此时更新其中的内容。从而实现连续的解码控制。

六. 解码引擎 (DJET) 指令简介

解码引擎接受 32BI 指令，通常情况下指令包括控制命令和参数两个部分，（只有在连续坐标输出时，指令包含两个模拟通道输出的参数）；控制命令固定占用最高一个字节，根据控制命令不同可以带一个或两个不同意义的参数，参数长度也会不同。

6. 1 指令列表:

控制命令	参数 1	参数 2	说明
0X10	连续坐标输出	PWM 控制输出	系统工作控制命令
0X12	延时时间	无	解码器延时
0X16	等待条件	无	解码器等待
0X20	DA0 输出值	延时时间	模拟量 0 通道绝对输出
0X21	DA1 输出值	延时时间	模拟量 1 通道绝对输出
0X22	DA2 输出值	延时时间	模拟量 2 通道绝对输出
0X23	DA3 输出值	延时时间	模拟量 3 通道绝对输出
0X28	DA0 输出值	选择开关	模拟量 0 通道选择输出
0X29	DA1 输出值	选择开关	模拟量 1 通道选择输出
0X30	D0 输出值	延时时间	数字量输出 D0 【9: 0】
0X31	ED0 输出值	延时时间	数字量输出 ED0 【15: 0】
0X34	D0 位选择	延时时间	数字量位置高
0X35	D0 位选择	延时时间	数字量位清零
0X40	脉宽初始宽度	无	脉宽初始值设置
0X42	脉宽调整步长	无	脉宽调整步长设置
0X44	步长调整次数	无	脉宽调整次数设置
0X46	周期值	无	PWM 输出周期值
0X48	延时时间	无	开激光延时设置
0X50	位置次数低位	无	连续位置输出次数设置低 16 位
0X51	位置次数高位	无	连续位置输出次数设置高 13 位
0X52	延时时间	无	连续位置输出延时设置
0X70	脉宽	无	预激励脉冲宽度
0X71	周期	无	预激励 PWM 周期
其他			保留功能

6.2 系统工作控制命令 offset=0x10

位域	31: 24	23:4	3	2:0
定义	0X10	保留	开始连续位置输出	PWM 输出控制

说明： BIT【3】由低到高，开始连续位置输出功能。以后的指令为连续的位置指令，每个指令包含 DA0, DA1 值。（参考连续位置输出介绍）

BIT【0】软件控制激光开关，高电平立刻开激光 LaserON, PWM 输出

BIT【1】硬件控制激光开关，DI【9】为低，开激光 LaserON, PWM 输出

BIT【2】智能控制激光开关，如果连续输出进行中开激光 LaserON, PWM 输出
参考 PWM 输出功能介绍

6.3. 解码器延时控制 offset=0x10

位域	31: 24	23:16	15:0
定义	0X12	保留	延时长度 T

说明： 参数 T 是需要延时的大小，单位是 0.1uS。

由于指令解析需要 0. uS, 所以实际延时是 (T+1) X 0.1uS

此命令只在 DMA 续解码时有用。

6.4. 解码器等待控制 offset=0x16

位域	31: 24	23:7	6	5	4:0
定义	0X10	保留	等待使能 Y	等待条件	位置选择 SEL

说明： 当使能为高时，解码器等待 SEL【4: 0】指定的 DI 位，同 Y 相同时继续执行以后的指令，否则一直等待。

SEL【4】=0 则 SEL【3: 0】选择 DI【9: 0】

SEL【4】=1 则 SEL【3: 0】选择 EDI【15: 0】

例如： 0x10000060 会等待 DI【0】为低时继续执行。

6.5 直接模拟量输出 offset=0x20, 0x21, 0x22, 0x23

位域	31: 24	23:16	15: 0
模拟量 0 通道	0x20	延时长度 M	DA 输出值 VALUE
模拟量 1 通道	0x21	延时长度 M	DA 输出值 VALUE
模拟量 3 通道	0x22	延时长度 M	DA 输出值 VALUE
模拟量 3 通道	0x23	延时长度 M	DA 输出值 VALUE

说明： 将给定的值写入对应的数模转换器，并执行延时 M 个延时单位

VALUE 范围 0000—0xffff 跳线选择对应±10 V 或 ±5V。

M 范围是 0—255，延时单位是 0.8u；

如“TEST01”跳线选择±5V

0x20190000; 设置 DA0=-5V, 延时 20uS

0x20194000; 设置 DA0=-2.5V, 延时 20uS

0x20198000; 设置 DA0=0V, 延时 20uS

0x2019c000; 设置 DA0=2.5V, 延时 20uS

0x2019ffff; 设置 DA0=5V, 延时 20uS

0x100003e7; 延时 100uS

0x20008000 设置 DA0=0V

6.6 选择模拟量输出 offset=0x28, 0x29

位域	31: 24	23:16	15: 0
模拟量 0 通道	0x28	选择值 S	DA 输出值 VALUE
模拟量 1 通道	0x29	选择值 S	DA 输出值 VALUE

说明：如果 EDI 【7: 0】同选择值相同，输出 VALUE 到 DA 通道。

否则忽略从命令。

6.7 数字量输出 offset=0x30, 0x31

位域	31: 24	23:16	15: 0
数字量 DO	0x30	延时大小 M	DO 输出值 VALUE
数字量 EDO	0x31	延时大小 M	EDO 输出值 VALUE

说明：将给定的值 VALUE 输出到数字输出，并执行延时 T 个延时单位

DO 范围是 【9: 0】，EDO 范围是 【15: 0】；

M 范围是 0—255，延时单位是 0.8uS

6.8 数字量位操作 offset=0x34, 0x35

位域	31: 24	23:8	7:5	4: 0
置高	0x34	延时大小 T	保留	位选择 S
清零	0x35	延时大小 T	保留	位选择 S

说明：对特定的位操作，并且延时 T 时间

S 【4】=0 选择 DO 端口，S 【3: 0】选择 0—9

S 【4】=2 选择 EDO 端口，S 【3: 0】选择 0—15

T 占用 16bit 范围 【0—65535】延时单位为 0.1uS

6.9 PWM 输参数设置 offset=0x40, 0x42, 0x44, 0x46, 0x48

位域	31: 24	23:16	15: 0
脉宽初始值设置	0x40	保留	PStartValue
脉宽调整步长设置	0x42	保留	PStepValue
脉宽调整次数设置	0x44	保留	PStepCounr
PWM 输出周期值	0x46	保留	PATime
开激光延时设置	0x48	保留	PSync

说明：以上参数占有 16 位宽度，时间单位 0.1uS

由于一些激光器需要在开激光头几个脉冲宽度小于正常的宽度。

这就需要设置脉宽调整步长，和调整次数。

开激光延时设置：激光打开操作到实际的 LaserOn 的时间。

例子如下

0x4000000a 设置脉宽初始值设置 1uS

0x4200000a 脉宽调整步长设置 1us
 0x44000002 脉宽调整次数 2 次;
 0x44000032 PWM 周期为 5uS
 0480000064 开激光延时 10uS
 0x10000001 打开 PWM

产生的周期为 5us 的 PWM 信号,
 第一个脉冲在 0x10000001 指令 10uS 出现;
 第一个脉宽 1uS, 第二个脉宽 2uS,
 第三个脉宽 3uS, 以后都为 3uS。

6.10 连续位置输出参数设置 offset=0x50, 0x51, 0x52

位域	31: 24	23:16	15: 0
连续位置输出次数设置低 16 位	0x40	保留	PStartValue
连续位置输出次数设置高 13 位	0x42	保留	PStepValue
连续位置输出延时设置	0x44	保留	PStepCounr

说明：连续位置输出在一个指令中输出 DA0 和 DA1 的值，
 可以节省内存，提高编程效率；
 输出次数是 29 位的整数，即最大可以是 512M 个【DA0，DA1】组合
 输出延时指两个输出间的延时:0—65535，时间单位 0.1uS

例如：0x50000006 ；
 0x51000000 ；需要连续输出 6 次
 0x52000064 输出延时 10uS
 0x10000008 开始连续位置输出；
 0x0000ffff DA0 输出 5V，DA1 输出—5V，延时 10uS
 0xffff0000 DA0 输出—5V，DA1 输出 5V，延时 10uS
 0x0000ffff DA0 输出 5V，DA1 输出—5V，延时 10uS
 0xffff0000 DA0 输出—5V，DA1 输出 5V，延时 10uS
 0x0000ffff DA0 输出 5V，DA1 输出—5V，延时 10uS
 0xffff0000 DA0 输出—5V，DA1 输出 5V，延时 10uS

6.11 预激励 PWM 参数设置 offset=0x70, 0x71

位域	31: 24	23:16	15: 0
脉宽	0x70	无	预激励脉冲宽度
周期	0x71	无	预激励 PWM 周期

说明：当没有控制激光开始时，可以由跳线选择是否预激励，
 脉宽和周期 16bit 的整数，单位时间 0.1uS；
 上电时初始化脉宽=1.6uS ；周期=50uS

七 应用程序接口函数简介:

T3000_GetCardCount(U32 *CardCount);

功能: 得到在系统中 TE3000 的卡数。

T3000_Open(U32 CardIndex, HANDLE *pDevice);

功能: 对准备操作的卡初始化, 得到句柄

T3000_Reset(hDevice);

功能: 复位 TE3000, 清空硬件缓冲区

T3000_Close(HANDLE hDevice)

功能: TE3000 操作结束, 释放系统资源

T3000_ReadIoPort(HANDLE hDevice, U32 Offset, U32 *pData);

功能: 读出特定偏移位置的值

说明: TE3000 使用此函数读取 DI, EDI 的值, Offset=0;

DI【9: 0】占用*pData【9: 0】

EDI【15: 0】占用*pData【31: 16】

T3000_WriteIoPort (HANDLE hDevice, U32 Offset, U32 Data)

功能: 写特定偏移位置的值

说明: TE3000 使用此函数对解码引擎直接控制, Offset=0;

T3000_SglTransfer(

HANDLE hDevice, U32 *pBuffer,U32 Length,

BUSWIDTH BusWidth, U32 IopToPci, BOOLEAN bSync)

功能: 开始指定的缓冲区输出, 块传输

说明: *pBuffer 缓冲区开始位置

Length 缓冲区大小 (BYTE 计算)

BSync 同步/异步传输, 同步指结束传输后函数返回

异步指函数立即返回, 以后查询传输是否结束

本卡固定值 BusWidth=3, IopToPci=0;

T3000_SglStopTransfer(HANDLE hDevice)

功能: 立刻结束块传输

T3000_SglCompleteEventGet(HANDLE hDevice, HANDLE *pEvent)

功能: 异步块传输时, 得到传输结束事件

T3000_IsSglComplete(HANDLE hDevice, BOOLEAN *pbDone)

功能: 异步块传输时，查询传输是否结束

T3000_ShuttleBufferSetup

(HANDLE hDevice, void *Buffer, U32 Count, U32 *BufferId);

功能: 建立循环缓冲区链表;

说明: *Buffer 为当前块指针;
Count 为当前块大小 (BYTE 记)
BufferId 当前块编号

注意

每块大小不能大于 64M;

T3000_ShuttleTransfer

(HANDLE hDevice, BUSWIDTH BusWidth, U32 IopToPci);

功能: 开始将缓冲区循环输出;

说明: 对于本卡 BusWidth=3, IopToPci=0;

T3000_ShuttleStopTransfer(HANDLE hDevice);

功能: 结束缓冲区循环输出

T3000_BlockFreeEventGet (HANDLE hDevice, HANDLE *pEvent)

功能: 循环输出时，得到块结束事件

T3000_BlockFreeIdGet (HANDLE hDevice, U32 *pId);

功能: 循环输出时，得到结束块的编号

八. 软件安装

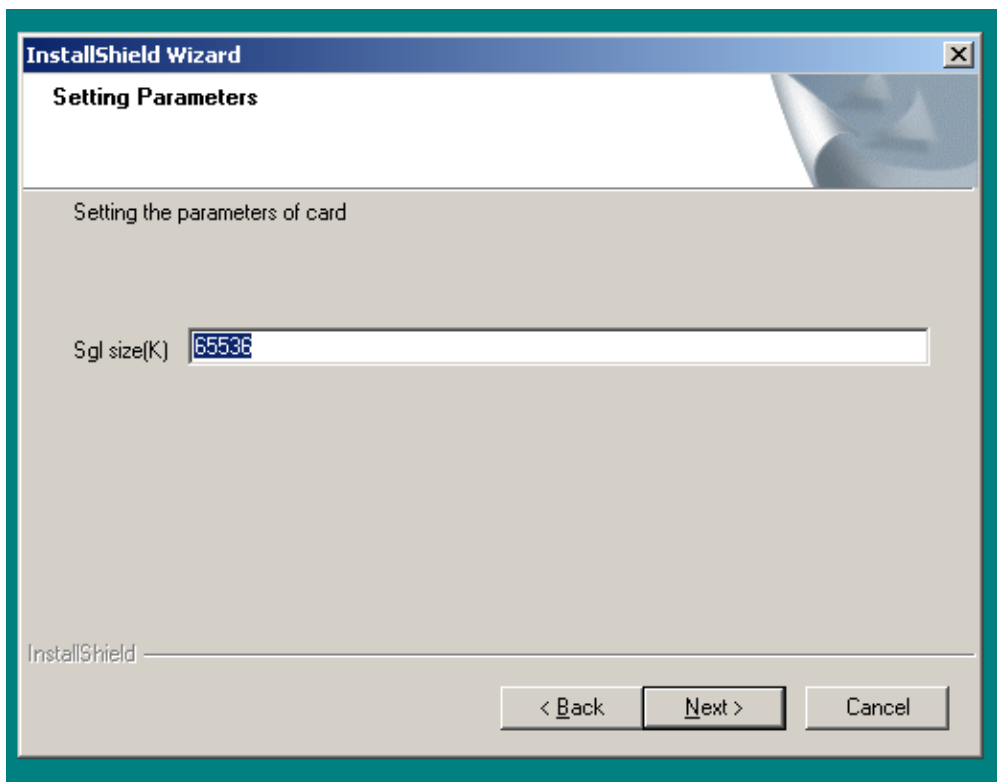
8. 1 软件列表

随机的软件包括驱动程序，应用程序开发库，例子程序。

驱动程序	.\Driver\Te3000.sys .\Driver\Te3000.inf .\Driver\Te3000.dll
头文件	.\include\Te3000Api.h .\include\Te3000Erro.h .\include\Te3000Type.h
库文件	.\Lib\Te3000.Lib
说明书	.\DOC\Te3000.pdf
例子程序	.\LaserSample*.*
安装文件	.\Setup*.*

8.2 驱动安装

执行 Setup\setup.exe 安装程序，按提示一步步操作



当提示输入 SGL 缓存最大值大小时，填入适当的值，默认值 64M，如果你的物理内存比较大可以选择更大的 SGL 缓存。

当所有的文件复制完成后，选择重新启动电脑。

九. 硬件安装

1. 关掉计算机及所有外设电源。
2. 打开机盖，选择一个空槽，拧开档条的固定螺丝。
3. 小心插入 TE3000 卡，上紧档条上的螺丝。
4. 盖上机盖，连接好外围设备，打开电源。

十、 编程指南

例 1：通用操作流程

```
:
T3000_GetCardCount (&Count);      //得到系统中 TE3000 卡数
T3000_Open(Index, &hDevice);      //打开指定的卡操作
:
T3000_WritelPort (hDevice, 0, 0x20004000) ; //DA0 输出-2. 5V.
T3000_WritelPort (hDevice, 0, 0x2100C000) ; //DA1 输出 2. 5V.
T3000_WritelPort (hDevice, 0, 0x31005555) ; //ED0[15:0]输出 0x5555.
T3000_WritelPort (hDevice, 0, 0x34000000) ; //D0[0]=1.
T3000_ReadIoPort(hDevice, 0, U32 *DI);      //读出 DI, EDI 数字输入的值
:
T3000_Close(hDevice);
```

注意：系统中可以有多个 TE3000 卡，应用程序支持对多卡操作。

例 1：块传输操作流程同步传输

```
i=0 ; DmaOutBuffer[i]=0x34006400; //SET_D0[0]=1 延时 10uS
i++ ; DmaOutBuffer[i]=0x35006400; //CLR_D0[0]=0 延时 10uS
i++ ; DmaOutBuffer[i]=0x34006400; //SET_D0[0]=1 延时 10uS
i++ ; DmaOutBuffer[i]=0x35006400; //CLR_D0[0]=0 延时 10uS
i++ ; DmaOutBuffer[i]=0x34006400; //SET_D0[0]=1 延时 10uS
i++ ; DmaOutBuffer[i]=0x35006400; //CLR_D0[0]=0 延时 10uS
i++ ; DmaOutBuffer[i]=0x34006400; //SET_D0[0]=1 延时 10uS
// 填充数据
3000_SglTransfer(hDevice, DmaOutBuffer, i*4 , BusWidth32, 0, FALSE);
T3000_SglStopTransfer(hDevice);
// SGL 块传输，同步传输，传输结束后函数返回
// 注意传输完成时，最后的数据进如解码器 FIFO，解码器解析指令还要时间
// 所以不能执行 FIFO 清空的函数。
```

例 3：块传输异步等待结束

```
T3000_SglTransfer(hDevice, DmaOutBuffer, i*4, BusWidth32, 0, TRUE);
bDone = FALSE;
while(!bDone)
{
    rc = T3000_IsSglComplete(hDevice, &bDone);
}
T3000_SglStopTransfer(hDevice);
//软件查询传输结束
```

例 4：块传输异步事件通知结束

```
rc = T3000_SglCompleteEventGet(hDevice, &hEvent); //安装事件
rc = T3000_SglTransfer(hDevice, DmaOutBuffer, i*4, BusWidth32, 0, FALSE);
dwRet = WaitForSingleObject(hEvent, 1000);
if(WAIT_TIMEOUT == dwRet)
    {printf("\n ERROR: T3000_SglTransfer time out!\n");}
T3000_SglStopTransfer(hDevice);
```

例 4：循环块传输流程

```
T3000_ShuttleBufferSetup(
    hDevice,
    Buffer0_start_adress,
    Buffer0_SIZE,
    &BufferId);
}; //安装第一个数据块
T3000_ShuttleBufferSetup(
    hDevice,
    Buffer1_start_adress,
    Buffer1_SIZE,
    &BufferId);
}; //安装第二个数据块
rc = T3000_ShuttleTransfer(hDevice, BusWidth32, 0);
time = n; //循环输出 n 次
while(time--)
{
    dwRet = WaitForSingleObject(hEvent, INFINITE);
    //等待数据块传输结束事件
    rc = T3000_BlockFreeIdGet(hDevice, &BufferId);
    //取得传输结束块的编号
    rc = T3000_BlockFreeEventGet(hDevice, &hEvent);
    //设置等待下次传输结束事件
    //应用程序在此更新数据块的内容
}
rc = T3000_ShuttleStopTransfer(hDevice);
// 注意：所安装的数据块小于 8 个；
// 注意：数据块总和大小不能超过注册本项目的限制值
// 注意：每个数据块大小要求小于 64M
// 注意：数据块太小会需要应用程序过多干预
```

十一、校准

交付到客户的每一块 TE3000 都经过校准，只有在极其少数情况下你才有必要重新对板卡进行校准。

11.1 可调电阻

对 TE3000 各模拟量输出通道的偏移和满度的校准，是通过一组可调电阻进行的，列表如下。

可调电阻编号	功能
VR1	模拟量输出 0 通道偏移校准
VR2	模拟量输出 1 通道偏移校准
VR3	模拟量输出 2 通道偏移校准
VR4	模拟量输出 3 通道偏移校准
VR5	模拟量输出满度校准（+）

11.2 校准过程

1) 模拟通道满度校准；

模拟通道 0 写入 0XFFF, 调节 VR5, 在输出端得到电压 10V。

满度校准是所有的输出通道共用，所以只需校准一个通道。

2) 模拟量输出 0 通道偏移校准

模拟通道 0 写入 0x8000, 调节 VR1, 在输出端得到电压 0V。

2) 模拟量输出 1 通道偏移校准

模拟通道 1 写入 0x8000, 调节 VR2, 在输出端得到电压 0V。

2) 模拟量输出 2 通道偏移校准

模拟通道 2 写入 0x8000, 调节 VR3, 在输出端得到电压 0V。

2) 模拟量输出 2 通道偏移校准

模拟通道 2 写入 0x8000, 调节 VR4, 在输出端得到电压 0V。