CAN-X-XX-XX
PXI/CPCI/PCI 总线
10/4 路 CAN 总线通讯接口卡

用户手册

目 录

第一章　概述

1.1 关于本手册

本手册适用于下列产品型号：

- **CAN-PXI-10** **3U** PXI 总线，包含 10 通道 CAN 收发通讯、16 路 DIO、时标、16 路内/外部触发
- **CAN-PCI-10** PCI 总线，包含 10 通道 CAN 收发通讯、16 路 DIO、时标、16 路内/外部触发
- **CAN-CPCI-10** **3U** CPCI 总线，包含 10 通道 CAN 收发通讯、16 路 DIO、时标、16 路内/外部触发
- **CAN-CPCI-10-10** **6U** CPCI 总线，前出线，包含 10 通道 CAN 收发通讯、16 路 DIO、时标、16 路内/外部触发
- **CAN-CPCI-10-11** **6U** CPCI 总线，后出线，包含 10 通道 CAN 收发通讯、16 路 DIO、时标、16 路内/外部触发
- **CAN-CPCI-4** **3U** CPCI 总线，包含 4 通道 CAN 收发通讯、16 路 DIO、时标、16 路内/外部触发
-

本手册是关于上述产品的完全使用指南。以下各章节提供了关于该产品更详细的信息，包括产品的功能特性、安装使用、硬件和软件说明等内容。

本手册的电子版本，您可以在购买产品的配套光盘中获得。

注意

在使用该产品之前，请您详细阅读本手册各章节的内容。

1.2 产品描述

CAN-X-XX 是一款包含多种通道配置的 CAN 通讯板卡，其功能能够满足用户的通讯测试需求，良好的兼容性适用于各类系统配置。

1.2.1 特性

- PXI/CPCI/PCI 总线
- 每路发送通道带 64KB×32 位 FIFO；
- 每路接收通道带 1MB×32 位 FIFO；
- 发送和接收 FIFO 复位；
- 波特率 10Kbps～1000Kbps，接收及发送可分别设置；
- 添加时间标签功能；
- 支持触发发送功能；
- 接收标号过滤；
- 标准 CAN 数据格式，接收及发送可分别设置；

1.2.2 详细描述

- 支持 CAN 通讯的各种帧格式：标准数据帧、标准遥传帧、扩展数据帧、扩展远程帧。
- 支持波特率 10kbps、20kbps、50kbps、100kbps、125kbs、200kbps、400kbps、800kbps、1000kbps。
- 支持帧屏蔽过滤功能。
- 发送各通道 fifo 大小为 64KB×32 位，接收各通道 fifo 大小为 1MB*32bit。
- 支持带触发发送、带时标接收功能。
- 支持缓冲区满中断。
- 支持 16 路 DIO，其中前 8 路可用于外部触发。

1.2.3 软件支持

- Windows (标配)：Win2000，WinXP/Win7 (X86，X64)
- RTX (定制)：5.5，7.1，8.1，9.0
- Vxworks (定制)：X86-V5.5，X86-V6.8，PPC603-Vx5.5，PPC603-Vx6.8
- QNX (定制)：X86-V6.5
- Linux (定制)：2.4，2.6，NeoKylin5
- Labview (定制)：RT

1.2.4 一般规格

- 物理尺寸：标准 PXI/CPCI 3U 卡 160mm×100mm×4HP，公差小于 0.2mm，带 3U 助拔器

  标准 PCI 卡 175mm×106mm，公差小于 0.2mm

  标准 CPCI 6U 卡 233.35mm×100mm×4HP，公差小于 0.2mm，带 6U 助拔器
- 连 接 器：前出线 SCSI68 孔式连接器

  后出线 3U J2 连接器

  后出线 6U J4,J5 连接器
- 工作电源：5V
- 相对湿度：5～95%，无凝结

1.3 产品安装

1.3.1 安装之前的准备

1. 在您安装产品之前请检查包装是否完好，以确定产品在运输的过程中没有遭到损坏。如果包装发现有破损，请您马上与运输商联系。
2. 在打开包装后请检查产品以及配件的完整性。打开产品外包装后，您应该发现如下产品
   - CAN-X-XX 板卡
   - 产品合格证
   - 产品配套光盘
   - 标配连接器

   如有规格不符，请您立刻联系我们，我们将负责维修或者更换。
3. 如果有可能，请您准备防静电工作台并佩戴防静电腕带。如果不具备以上静电防护装备，请您接触计算机设备的导地部分，例如机箱壳金属部分，以释放身体上的静电。

现在您可以准备安装 CAN-X-XX 板卡了。

1.3.2 硬件安装

第一步： 打开板卡的防静电包装袋，取出板卡。

注意

手持板卡时，请您尽量只接触板卡的边缘。在板卡安装到您的计算机设备之前，请将板卡平放置于防静电包装袋中，这样有利于保护板卡不受静电损伤。取出板卡后，请您保留产品的防静电和防震包装，以便在您不使用时产品可以妥善存放。

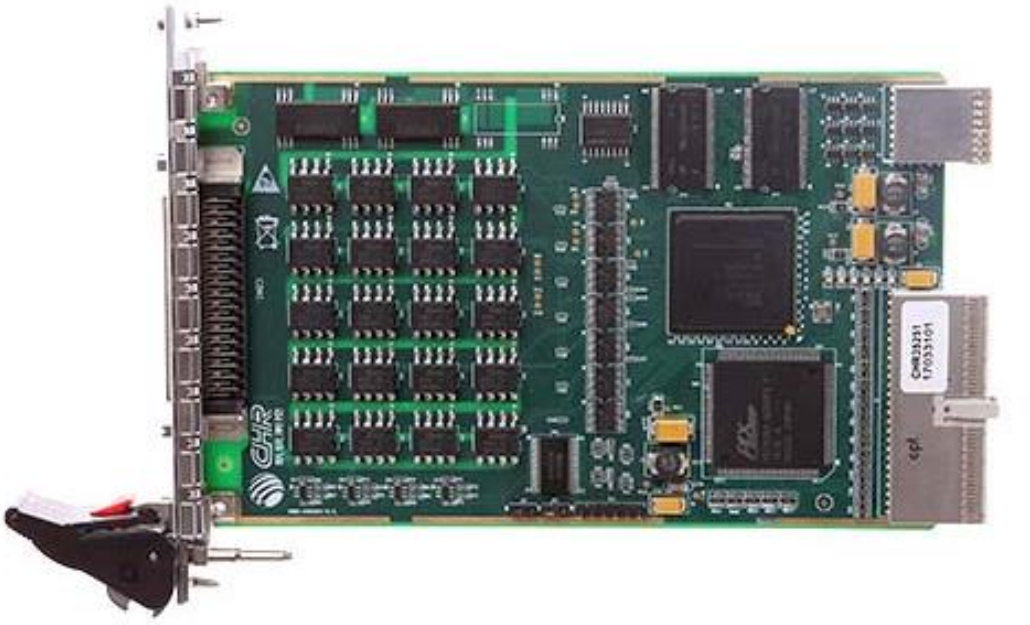


图 1-1a CAN-CPCI-10 产品图片示例

第二步： 关闭计算机设备的电源，将板卡安装到您的计算机机箱内。

第三步： 将配套的连接器或连接电缆插到板卡的连接器接口上。

关于连接电缆的制作请参照 2.2 节的内容。

开启计算机，系统提示发现新硬件，然后安装产品的驱动。

1.3.3 驱动安装

1）Windows XP 安装驱动说明

在产品配套光盘的“驱动安装”目录中，可以找到 **CAN-X-XX** 板卡的驱动。请按如下步骤安装产品的驱动：

1、 将设备插入工控机 CPCI/PCI 插槽，系统检测到新硬件，这时选择“从列表或指定位置安装”，点击“下一步”继续安装。

2、 选择“不要搜索。我要自己选择要安装的驱动程序”，点击“下一步”继续安装。

3、 选择“显示所有设备”，点击“下一步”继续安装。

4、 选择“从磁盘安装”。

5、 选择“浏览”在硬盘或光盘中找到驱动程序 CANX20.inf 文件所在的目录

6、点击下一步，直到驱动程序安装成功

在完成 **CAN-X-XX** 板卡驱动安装后，您可以通过计算机系统的“设备管理器”来确认板卡驱动是否正确安装。访问“设备管理器”可以通过“控制面板”/“系统”/“设备管理器”。如果板卡驱动正确安装，您可以在“设备管理器”的设备列表中看到 **CAN-X-XX** 板卡设备项，如图 1-2 所示。

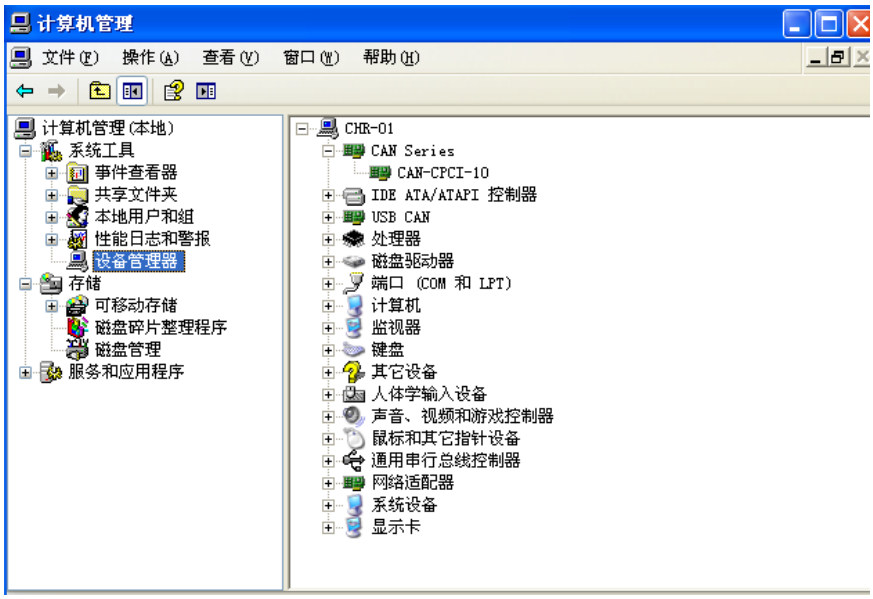
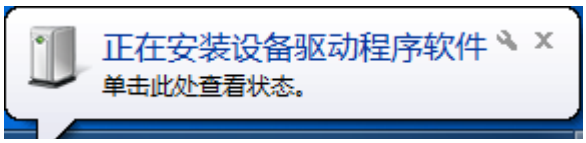
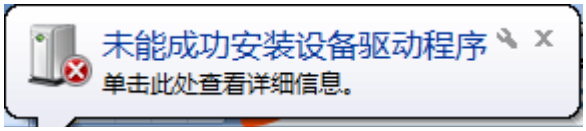


图 1-2 CAN-CPCI-10 板卡在设备管理器中的设备项

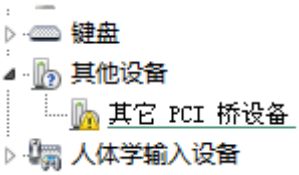
2）Windows 7 安装驱动说明

将设备插入计算机，右下角会弹出如下提示。

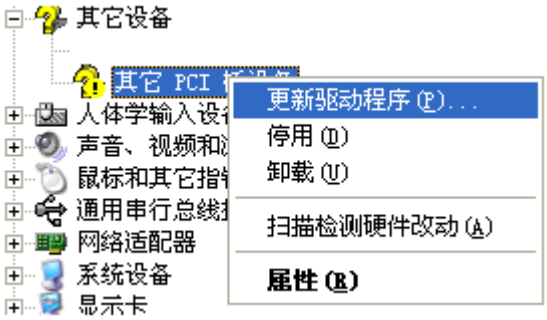




此时电脑中的设备管理器中如下图显示设备驱动为未安装状态。



右键更新驱动程序软件，手动安装方法如 XP，可供参考。



1.3.4 演示应用软件安装

在安装完 CAN-X-XX 板卡驱动之后，可使用配套光盘中附带的功能演示软件进行板卡的测试。演示软件可以满足基本的产品测试和演示功能。具体使用方法，请参考第 4 章节的内容。

第二章　硬件说明

本章描述了 CAN-X-XX 板卡硬件信息，包括硬件设置、I/O 连接器和信号定义等。

2.1 功能结构图

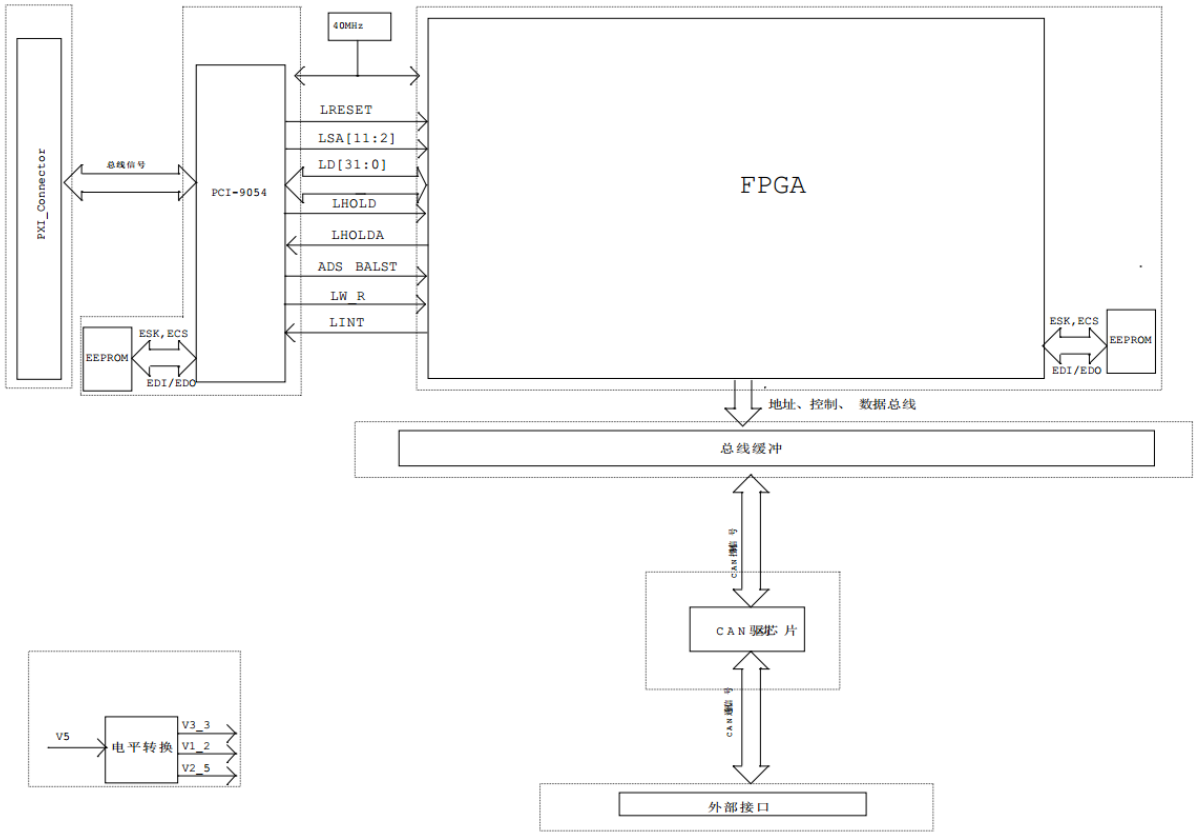


图 2-1 CAN-X-10 功能结构图

2.2 印制板示意图

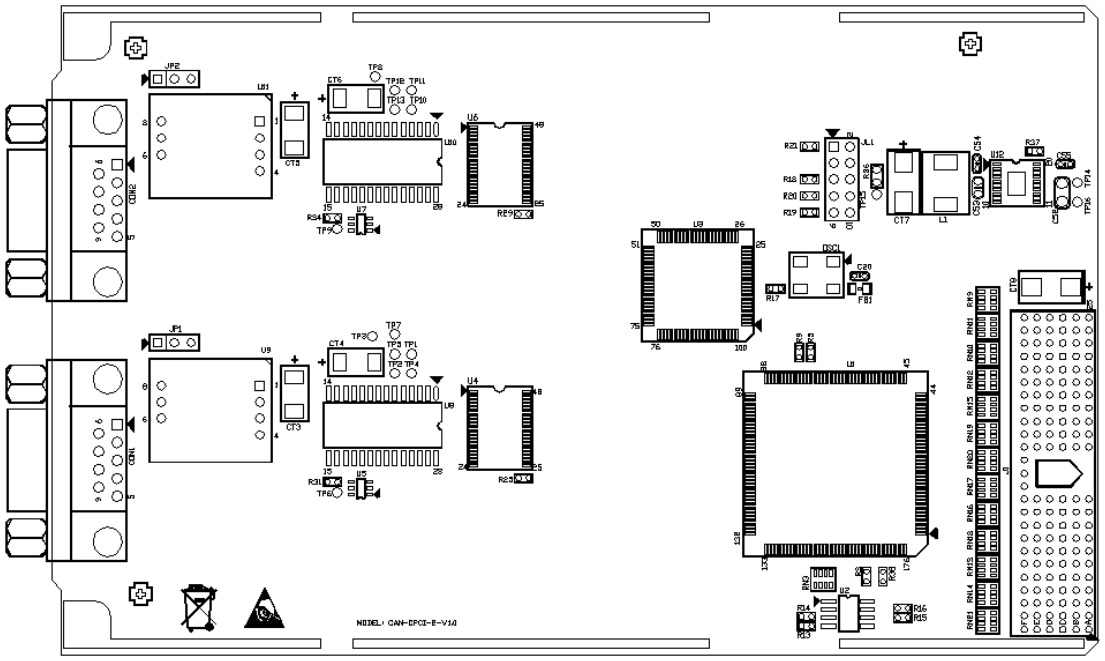


图 2-2 CAN-X-10 印制板示意图

2.3 连接器和信号定义

CAN-X-XX 板卡采用了一个 SCSI68 母座用于信号输入输出。

2.3.1 前面板与连接器

图 2-3 描述了 CAN-X-XX 板卡连接器在前面板的位置。

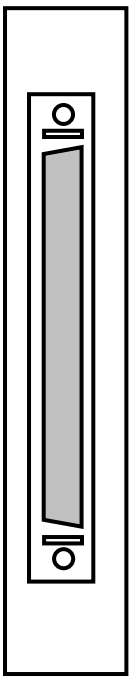


图 2-3 CAN-X-10 前面板示意图

2.3.2 点号定义

图 2-4 描述了 CAN-X-XX 板卡连接器的点号定义。

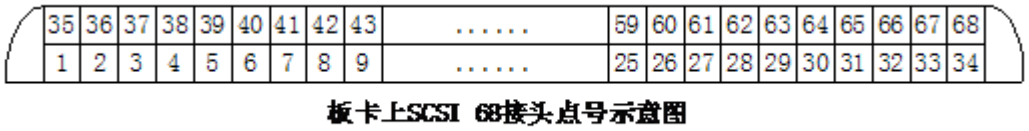


图 2-4 连接器点号定义

2.3.3 连接器和信号定义

各通道类型板卡点号定义

CAN-CPCI-10,CAN-PCI-10,CAN-PXI-10 板卡采用 SCSI-68 标准接头。点号定义如下表（空白部分不连接）：

管脚	名称	说明	管脚	名称	说明
1			34		
2			35		
3			36	CAN_GND	CAN 参考地
4			37		
5			38	GND	参考地

6			39	DIO8	数字量 8 通道
7			40	DIO9	数字量 9 通道
8			41	DIO10	数字量 10 通道
9	CAN_H4	CAN 第 4 通道 H 线	42	DIO11	数字量 11 通道
10	CAN_L4	CAN 第 4 通道 L 线	43	DIO12	数字量 12 通道
11	CAN_H5	CAN 第 5 通道 H 线	44	DIO13	数字量 13 通道
12	CAN_L5	CAN 第 5 通道 L 线	45	DIO14	数字量 14 通道
13	CAN_H6	CAN 第 6 通道 H 线	46	DIO15	数字量 15 通道
14	CAN_L6	CAN 第 6 通道 L 线	47		
15	GND	参考地	48		
16	DIO0	数字量 0 通道	49		
17	DIO1	数字量 1 通道	50		
18	DIO2	数字量 2 通道	51		
19	DIO3	数字量 3 通道	52		
20	DIO4	数字量 4 通道	53		
21	DIO5	数字量 5 通道	54		
22	DIO6	数字量 6 通道	55		
23	DIO7	数字量 7 通道	56		
24	CAN_H7	CAN 第 7 通道 H 线	57	CAN_H3	CAN 第 3 通道 H 线
25	CAN_L7	CAN 第 7 通道 L 线	58	CAN_L3	CAN 第 3 通道 L 线
26			59	CAN_H2	CAN 第 2 通道 H 线
27	CAN_H8	CAN 第 8 通道 H 线	60	CAN_L2	CAN 第 2 通道 L 线
28	CAN_L8	CAN 第 8 通道 L 线	61	CAN_H1	CAN 第 1 通道 H 线
29			62	CAN_L1	CAN 第 1 通道 L 线
30	CAN_H9	CAN 第 9 通道 H 线	63	CAN_H0	CAN 第 0 通道 H 线
31	CAN_L9	CAN 第 9 通道 L 线	64	CAN_L0	CAN 第 0 通道 L 线
32			65		
33			66	CAN_GND	CAN 参考地
67	GND	参考地	68		

CAN-CPCI-10-01 板卡采用 SCSI-68 标准接头。

管脚	名称	说明	管脚	名称	说明
1	DIO0	数字量 0 通道	35	DIO8	数字量 8 通道
2	DIO1	数字量 1 通道	36	DIO9	数字量 9 通道
3	DIO2	数字量 2 通道	37	DIO10	数字量 10 通道
4	DIO3	数字量 3 通道	38	DIO11	数字量 11 通道
5	DIO4	数字量 4 通道	39	DIO12	数字量 12 通道
6	DIO5	数字量 5 通道	40	DIO13	数字量 13 通道
7	DIO6	数字量 6 通道	41	DIO14	数字量 14 通道
8	DIO7	数字量 7 通道	42	DIO15	数字量 15 通道
9	GND	参考地	43	GND	参考地
10	GND	参考地	44	GND	参考地
11			45	GND	参考地

12			46	GND	参考地
13			47		
14			48		
15			49		
16			50		
17			51		
18			52		
19	CAN_H4	CAN 第 4 通道 H 线	53		
20	CAN_L4	CAN 第 4 通道 L 线	54		
21	CAN_H5	CAN 第 5 通道 H 线	55		
22	CAN_L5	CAN 第 5 通道 L 线	56		
23	CAN_H6	CAN 第 6 通道 H 线	57	CAN_H3	CAN 第 3 通道 H 线
24	CAN_L6	CAN 第 6 通道 L 线	58	CAN_L3	CAN 第 3 通道 L 线
25	CAN_H7	CAN 第 7 通道 H 线	59	CAN_H2	CAN 第 2 通道 H 线
26	CAN_L7	CAN 第 7 通道 L 线	60	CAN_L2	CAN 第 2 通道 L 线
27	CAN_H8	CAN 第 8 通道 H 线	61	CAN_H1	CAN 第 1 通道 H 线
28	CAN_L8	CAN 第 8 通道 L 线	62	CAN_L1	CAN 第 1 通道 L 线
29	CAN_H9	CAN 第 9 通道 H 线	63	CAN_H0	CAN 第 0 通道 H 线
30	CAN_L9	CAN 第 9 通道 L 线	64	CAN_L0	CAN 第 0 通道 L 线
31			65		
32			66		
33	CAN_GND	CAN 参考地	67	CAN_GND	CAN 参考地
34	CAN_GND	CAN 参考地	68	CAN_GND	CAN 参考地

CAN-CPCI-10-11 板卡配合 CHR97010-1（6U 后走线转接板），采用 SCSI-68 标准接头。

管脚	名称	说明	管脚	名称	说明
1	DIO0	数字量 0 通道	35	DIO8	数字量 8 通道
2	DIO1	数字量 1 通道	36	DIO9	数字量 9 通道
3	DIO2	数字量 2 通道	37	DIO10	数字量 10 通道
4	DIO3	数字量 3 通道	38	DIO11	数字量 11 通道
5	DIO4	数字量 4 通道	39	DIO12	数字量 12 通道
6	DIO5	数字量 5 通道	40	DIO13	数字量 13 通道
7	DIO6	数字量 6 通道	41	DIO14	数字量 14 通道
8	DIO7	数字量 7 通道	42	DIO15	数字量 15 通道
9	GND	参考地	43	GND	参考地
10	GND	参考地	44	GND	参考地
11			45	GND	参考地
12			46	GND	参考地
13			47		
14			48		
15			49		
16			50		
17			51		
18			52		

19	CAN_H4	CAN 第 4 通道 H 线	53		
20	CAN_L4	CAN 第 4 通道 L 线	54		
21	CAN_H5	CAN 第 5 通道 H 线	55	CAN_H3	CAN 第 3 通道 H 线
22	CAN_L5	CAN 第 5 通道 L 线	56	CAN_L3	CAN 第 3 通道 L 线
23	CAN_H6	CAN 第 6 通道 H 线	57		
24	CAN_L6	CAN 第 6 通道 L 线	58		
25	CAN_H7	CAN 第 7 通道 H 线	59	CAN_H1	CAN 第 1 通道 H 线
26	CAN_L7	CAN 第 7 通道 L 线	60	CAN_L1	CAN 第 1 通道 L 线
27	CAN_H8	CAN 第 8 通道 H 线	61	CAN_H2	CAN 第 2 通道 H 线
28	CAN_L8	CAN 第 8 通道 L 线	62	CAN_L2	CAN 第 2 通道 L 线
29	CAN_H9	CAN 第 9 通道 H 线	63	CAN_H0	CAN 第 0 通道 H 线
30	CAN_L9	CAN 第 9 通道 L 线	64	CAN_L0	CAN 第 0 通道 L 线
31			65		
32			66		
33	CAN_GND	CAN 参考地	67	CAN_GND	CAN 参考地
34	CAN_GND	CAN 参考地	68	CAN_GND	CAN 参考地

CAN-CPCI-4 板卡采用 SCSI-68 标准接头。点号定义如下表（空白部分不连接）：

管脚	名称	说明	管脚	名称	说明
1			34		
2			35		
3			36	CAN_GND	CAN 参考地
4			37		
5			38	GND	参考地
6			39	DIO8	数字量 8 通道
7			40	DIO9	数字量 9 通道
8			41	DIO10	数字量 10 通道
9			42	DIO11	数字量 11 通道
10			43	DIO12	数字量 12 通道
11			44	DIO13	数字量 13 通道
12			45	DIO14	数字量 14 通道
13			46	DIO15	数字量 15 通道
14			47		
15	GND	参考地	48		
16	DIO0	数字量 0 通道	49		
17	DIO1	数字量 1 通道	50		
18	DIO2	数字量 2 通道	51		
19	DIO3	数字量 3 通道	52		
20	DIO4	数字量 4 通道	53		
21	DIO5	数字量 5 通道	54		
22	DIO6	数字量 6 通道	55		
23	DIO7	数字量 7 通道	56		
24			57	CAN_H3	CAN 第 3 通道 H 线
25			58	CAN_L3	CAN 第 3 通道 L 线

26			59	CAN_H2	CAN 第 2 通道 H 线
27			60	CAN_L2	CAN 第 2 通道 L 线
28			61	CAN_H1	CAN 第 1 通道 H 线
29			62	CAN_L1	CAN 第 1 通道 L 线
30			63	CAN_H0	CAN 第 0 通道 H 线
31			64	CAN_L0	CAN 第 0 通道 L 线
32			65		
33			66	CAN_GND	CAN 参考地
67	GND	参考地	68		

第三章　驱动程序编程接口

本章主要讲述了如何使用 CAN-X-XX 板卡的驱动程序接口，为用户编程提供参考。CAN-X-XX 驱动程序提供了丰富的接口函数，能满足用户对板卡的操作需求；具有良好的兼容性，能适用于多种编程环境；操作简单方便，可以大大缩短用户的开发周期。

3.1 动态库 DLL

CAN-X-XX 驱动程序接口函数按 ANSI C 标准编写，以动态链接库 DLL 形式提供给用户。您可以在 CAN-X-XX 板卡配套光盘中获取。

3.1.1 适用编程工具

运行环境：Windows：Win2000，WinXP/Win7 (X86，X64)操作系统

开发工具：

- Visual C++
- Visual Basic
- C++ Builder
- Delphi
- Labview
- Labwindows/CVI

3.1.2 需要引用的文件

库文件：

CANX20.dll

CANX20.lib

函数库头文件：

CANX20_lib.h

3.2 驱动程序函数功能

3.2.1 CAN 功能

在 CAN 功能应用中，需要用户编写应用程序，调用 CAN 接口函数，可实现通用 CAN 实现的功能。串口接口函数可实现如下功能：

- 打开、关闭板卡
- 获取板卡信息
- 复位板卡
- 板卡自检
- CAN 功能

  初始化 CAN

  使能、禁用接收时标

  数据接收

  数据发送
- 触发功能

  复位触发

  触发时钟源选择

  配置触发输出

  配置触发输入

  获取触发输出个数

  开启、停止触发输出

获取当前触发输出状态

- 数字量 I/O

设置数字量输入模式
设置数字量输出模式
数字量输入的触发配置
数字量输出的触发配置
使能、禁用时标
输出使能控制
设置方向（输入、输出）
复位输入 FIFO
复位输出 FIFO
获取输入状态
按通道获取输入状态
按通道设置输出状态
按通道获取输出状态

- 时标功能

设置时标功能
设置时标计数初值
获取当前时标计数值
触发开启配置
开启、停止时标计数

3.2.2CAN 功能结构定义

3.2.2.1 CAN 帧结构

```
typedef struct   _STMCPCAN_FRAME_
{
    CCAN_UINT8 FrameType;
    CCAN_UINT8 RequestType;
    CCAN_UINT8 DataCnt;
    CCAN_UINT8 DataBuf[16];
    CCAN_UINT32 Id;
    CCAN_UINT64 ulltmtg;
} STMCPCAN_FRAME, *pSTMCPCAN_FRAME;
```

结构参数说明：

FrameType：帧类型

4：扩展帧，帧标识符为 29 位

2：标准帧，帧标识符为 11 位

RequestType：帧请求类型

1：远程帧

0：数据帧

Id：帧标识符

DataCnt：帧数据字节的个数，取之范围 0 ~ 8

DataBuf：存放帧的数据字节

Ulltmtg：时标

3.2.2.2 CAN 滤波屏蔽设置结构

```
typedef struct  _STMCPCAN_FMCFG_
{
    CCAN_UINT8 FrameType;
    CCAN_UINT8 AcceptFilterMode;
    CCAN_UINT32 dwFilter[2];
    CCAN_UINT32 dwMASK;
}STMCPCAN_FMCFG, * pSTMCPCAN_FMCFG;
```

FrameType：帧类型

8：扩展帧和标准帧

4：扩展帧

2：标准帧

AcceptFilterMode：验收滤波器模式设置

1：单个验收滤波器，对应 dwFilter[0], (dwFilter[1]默认值为 0)

0：两个验收滤波器，对应 dwFilter[0]和 dwFilter[1]

dwFilter：过滤位

dwMASK: 屏蔽位

屏蔽位 n	过滤位 n	报文标示符位	接受或拒绝位 n
0	x	x	接受
1	0	0	接受
1	0	1	拒绝
1	1	0	拒绝
1	1	1	接受

表 3-2 滤波/屏蔽真值表

注：屏蔽位为假，X=任意值，报文接受。

屏蔽位为真，进行过滤，且过滤位与报文标示符位相同接受，否则拒绝。

3.2.2.3 触发输出配置结构

```
typedef struct _TRIGGER_OUT_CONFIG_STRUCT_
{
    CCAN_BOOL   TrigEnable;
    CCAN_BOOL   OutEnable;
    CCAN_UINT32 TrigLevel;
    CCAN_UINT32 HighWidth;
    CCAN_UINT32 LowWidth;
    CCAN_UINT32 LevelDefault;
    CCAN_UINT32 TrigOutCnt;
```

} TRIGOUT_CONFIG_STRUCT, *pTRIGOUT_CONFIG_STRUCT;

结构参数:

TrigEnable：使能、禁用触发输出，值为 1 表示使能，值为 0 表示禁用

OutEnable：使能、禁用输出到外部连接器

TrigLevel：触发输出有效电平选择，取值 0~3，定义如下：

取值	触发线选择
0	保留（无效）
1	上升沿有效
2	下降沿有效
3	保留（无效）

HighWidth：触发输出高电平宽度，取值为 0~0x00FFFFFF，分辨率为 1us

LowWidth：触发输出低效电平宽度，取值为 0~0x00FFFFFF，分辨率为 1us

LevelDefault：触发输出空闲电平，取值为 0 时表示“低电平”，取值为 1 时表示“高电平”

TrigOutCnt：触发输出个数，取值 0~0xFFFFFFFF，0 值时表示无限输出，否则按个数输出

3.2.2.4 触发输入配置结构

```
typedef struct
{
    CCAN_BOOL    TrigEnable;
    CCAN_UINT32 TrigSrc;
    CCAN_UINT32 TrigLevel;
    CCAN_UINT32 FilterPeriod;
    CCAN_UINT32 WorkDelay
} TRIGIN_CONFIG_STRUCT, *pTRIGIN_CONFIG_STRUCT;
```

结构参数:

TrigEnable：使能、禁用触发输入，值为 1 表示使能，值为 0 表示禁用

TrigSrc：触发线选择，取值范围为 0~15，取值定义如下：

取值	触发线选择
0	PXI 触发线 0 有效
1	PXI 触发线 1 有效
2	PXI 触发线 2 有效
3	PXI 触发线 3 有效
4	PXI 触发线 4 有效
5	PXI 触发线 5 有效
6	PXI 触发线 6 有效
7	PXI 触发线 7 有效
8	TTL 外触发线 0 有效
9	TTL 外触发线 1 有效
10	TTL 外触发线 2 有效
11	TTL 外触发线 3 有效
12	TTL 外触发线 4 有效

13	TTL 外触发线 5 有效
14	TTL 外触发线 6 有效
15	TTL 外触发线 7 有效

TrigLevel：触发输入有效电平选择，取值 0~3，定义如下：

取值	触发线选择
0	保留（无效）
1	上升沿有效
2	下降沿有效
3	保留（无效）

FilterPeriod：触发输入滤波周期宽度，取值为 0~65535，分辨率为 1us

WorkDelay：触发输入延迟工作时间，分辨率为 1us

3.2.2.5 数字量输入数据结构

```
typedef struct
{
    CCAN_UINT32 Status;
    CCAN_UINT64 TimeTag;
} IODATA_STRUCT, *pIODATA_STRUCT;
```

结构参数:

Status：输入状态，第 0~15 位分别代表第 0~15 通道，位值为 0 表示“低电平”，位值为 1 表示“高电平”

TimeTag：时标，分辨率为 1us，低 48 位有效，此参数只有在使能数字量输入的时标功能后有效

3.3 CAN 功能驱动软件接口函数说明

本节内容详细描述了 API 函数的调用原形，函数功能、参数说明和返回值。

3.3.1 板卡功能

3.3.1.1 CANX20_Open

函数原型：CCAN_INT32 __stdcall ***CANX20_Open*** (CCAN_UINT16 Bus, CCAN_UINT16 Dev, CCAN_UINT16 Type, CCAN_HANDLE *pHandle);

函数功能：打开板卡。

参数说明：

Bus：总线号，取值范围为 0 ~ 255；

Dev：设备号，取值范围为 0 ~ 255；

Type：本类板卡型号编码，取值范围为 0 ~ 255；

pHandle：用于返回打开板卡的句柄。

返 回 值：返回值为 0 表示函数执行成功。

备　　注：“总线号”与“设备号”可从“设备管理器”中的板卡驱动属性对话窗口里“常规”页面中的“位置”处找到。“本类板卡型号编码”可从用户手册中找到，或从开发制造商处获取。

3.3.1.2 CANX20_Close

函数原型：CCAN_INT32 __stdcall ***CANX20_Close*** (CCAN_HANDLE Handle);

函数功能：关闭板卡。

参数说明：**Handle**：板卡的句柄。

返 回 值：返回值为 0 表示函数执行成功。

备　　注：关闭板卡时，此函数会对板卡进行复位操作。

3.3.1.3 CANX20_GetDevBusInfo

函数原型：CCAN_INT32 __stdcall ***CANX20_GetDevBusInfo*** (CCAN_HANDLE Handle,

pDEVICEBUSINFO_STRUCT pBusinfo);

函数功能：获取板卡信息。

参数说明：

Handle：板卡的句柄；

pBusinfo：设备信息结构体指针。

```
typedef struct
{
    WORD wdDevID;                   //厂商号
    WORD wdVenID;                   //设备号
    WORD wdSubDevID;        //SUB
    WORD wdSubVenID;        //SUB
    WORD    wdBusNum;       //总线号
    WORD    wdDevNum;       //设备号
    WORD    wdFunNum;       //功能号
    WORD    wdIrqNum;       //中断号
    DWORD dwSN;             //序列号
    DWORD dwVer;                    //硬件号
    ULONG dwMemBase[6][2];//[1]表示起始地址,[0]表示长度(长度为0表示没有)
}DEVICEBUSINFO_STRUCT, *pDEVICEBUSINFO_STRUCT;
```

返 回 值：返回值为 0 表示函数执行成功。

3.3.1.4 CANX20_Reset

函数原型：CCAN_INT32 __stdcall ***CANX20_Reset*** (CCAN_HANDLE Handle);

函数功能：复位板卡。

参数说明：**Handle**：板卡的句柄。

返 回 值：返回值为 0 表示函数执行成功。

注：复位板卡后，重新对模块功能进行配置！

3.3.2 CAN 功能接口

3.3.2.1 CANX20_SetBaud

函数原型：CCAN_INT32 __stdcall ***CANX20_SetBaud*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch,

BYTE BaudCode);

函数功能：初始化串口。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

BaudCode：波特率代码

表 3-1 波特率与时间参数寄存器

BaudCode	波特率
0	10Kbps
1	20Kbps
2	50Kbps
3	100Kbps
4	125Kbps
5	250Kbps
6	500Kbps
7	800Kbps
8	1000Kbps

返 回 值：返回值为 0 表示函数执行成功。

3.3.2.2 CANX20_ZeroFIFO

函数原型：CCAN_INT32 __stdcall ***CANX20_ZeroFIFO*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch);

函数功能：发送、接收 FIFO 清零。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

返 回 值：返回值为 0 表示函数执行成功。

3.3.2.3 CANX20_TxMode

函数原型：CCAN_INT32 __stdcall ***CANX20_TxMode*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_UINT8 TxMode);

函数功能：设置 CAN 数据发送模式（“静态发送”、“动态发送”）。动态发送：（触发定时）更新发送所有缓冲区帧数据。

注：需要先停止定时发送，然后向发送缓冲区中写入定时发送的数据，然后再开启定时发送。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

TxMode：发送模式，值为 1 时，表示触发定时发送，值为 0 时，表示普通发送。

返 回 值：返回值为 0 表示函数执行成功。

备　　注：

静态发送：硬件将发送缓存中数据发送一次。

动态发送：触发定时发送，每次触发发送一帧数据，直到发送 FIFO 为空。

3.3.2.4 CANX20_TxTrigInConfig

函 数 原 型： CCAN_INT32 __stdcall ***CANX20_TxTrigInConfig*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, pTRIGIN_CONFIG_STRUCT pstCfgInput);

函数功能：触发发送配置。当触发输入有效时，CAN 触发发送。要求配置发送为“动态发送”模式，调用此函数使能触发发送后，CAN 才会按触发进行发送。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

pstCfgInput：指定触发输入配置结构的地址。

返 回 值：返回值为 0 表示函数执行成功。

备　　注：触发发送：在触发脉冲有效边沿时刻，将发送缓存中的数据发送出去。触发信号每次到来时发送一帧数据。

3.3.2.5 CANX20_TxFIFOStatus

函数原型：CCAN_INT32 __stdcall ***CANX20_TxFIFOStatus*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_UINT8 *pStatus);

函数功能：获取硬件发送 FIFO 的状态。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

pStatus：返回硬件发送缓冲区的状态，如下表所示：

位	Bit 7~3	Bit 2	Bit 1	Bit 0

定义	0	FIFOA 满	FIFO 满	FIFO 空

第 7~2 位的值恒为 0；

FIFOA 满标志：该位为 1 时，表示 FIFOA 已满，为 0 时，表示不满；

FIFO 空标志：该位为 1 时，表示缓冲区为空，即没有数据，为 0 时，表示不空；

FIFO 满标志：该位为 1 时，表示缓冲区已满，为 0 时，表示不满。

注： FIFOA 满或者 FIFO 满，都不能调用 CANX20_TxWrite 写入数据帧。

返 回 值：返回值为 0 表示函数执行成功。

3.3.2.6 CANX20_TxFIFOCnt

函 数 原 型： CCAN_INT32 __stdcall ***CANX20_GetTxFIFOCnt*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_UINT32 *pCount);

函数功能：获取发送 FIFO 的数据量。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

pCount：返回当前发送 FIFO 中的数据量。

返 回 值：返回值为 0 表示函数执行成功。

3.3.2.7 CANX20_TxWrite

函数原型：CANX20_INT32 __stdcall ***CANX20_TxWrite*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, pSTMCPCAN_FRAME pTxFrame);

函数功能：数据帧发送，向发送 FIFO 缓冲区中的写入数据帧。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

pTxFrame：存放需写入发送 FIFO 中的数据帧；

返 回 值：返回值为 0 表示函数执行成功。

3.3.2.8 CANX20_RxAddTimeTag

函数原型：CCAN_INT32 __stdcall ***CANX20_RxAddTimeTag*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch,CCAN_BOOL Enabled);

函数功能：使能、禁用接收时标。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

Enabled：值为 1 时，表示使能接收时标，值为 0 时，表示禁用接收时标。

返 回 值：返回值为 0 表示函数执行成功。

3.3.2.9 CANX20_RxMode

函数原型：CCAN_INT32 __stdcall ***CANX20_RxMode*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_UINT8 Mode, pSTMCPCAN_FMCFG pfmcfg);

函数功能：滤波和屏蔽设置。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

Mode:

0: 关闭屏蔽滤波功能，接收所有报文(此时 pfmcfg 参数无意义)

1:接收符合滤波条件的扩展帧或标准帧

2:只接收符合滤波条件的标准帧，不管是标准数据帧还是标准远程帧

3:只接收符合滤波条件的扩展帧，不管是扩展数据帧还是扩展远程帧

pfmcfg：滤波和屏蔽配置结构。

返 回 值：返回值为 0 表示函数执行成功。

备 注：接收时标为 48 位数据（即 6 个 8 位字节），当使能接收时标后，每个串口数据后都会跟随有 6 个字节的时标，所以此时读取一个数据后还需要应用程序再从接收缓冲区中再多读取 6 个字节的时标。时标的分辨率为 1us，串口数据后的时标按“先低字节，后高字节”的顺序排列。

3.3.2.10 CANX20_RxFIFOStatus

函数原型：CCAN_INT32 __stdcall ***CANX20_RxFIFOStatus*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_UINT8 *pStatus);

函数功能：获取接收 FIFO 的状态。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

pStatus：返回硬件接收缓冲区的状态，如下表所示：

位	Bit 7~3	Bit 2	Bit 1	Bit 0

定义	0	帧到达标志	满标志	空标志

第 7~3 位的值恒为 0；

帧达标志：当缓冲区（FIFO）中有数据时，该位为 1，否则该位为 0；（暂未实现）

满标志：该位为 1 时，表示缓冲区已满，为 0 时，表示不满；

空标志：该位为 1 时，表示缓冲区为空，即没有数据，为 0 时，表示不空。

返 回 值：返回值为 0 表示函数执行成功。

3.3.2.11 CANX20_RxFIFOCnt

函数原型：CCAN_INT32 __stdcall ***CANX20_RxFIFOCnt*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_UINT32 *pCount);

函数功能：非中断接收下,获取接收 FIFO 里的数据量。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

pCount：返回当前接收缓冲区（FIFO）中的数据量。

返 回 值：返回值为 0 表示函数执行成功。

3.3.2.12 CANX20_RxRead

函数原型：CANX20_INT32 __stdcall ***CANX20_RxRead*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, pSTMCPCAN_FRAME pRxFrame);

函数功能：数据接收，从接收 FIFO 缓冲区中读取数据帧。当使能时标时，两个数据之间会一个 48 位的时标（共 6 个字节），每一个时标均占用 FIFO 中 4*32bit。

无时标 FIFO 格式		有时标 FIFO 格式	
帧数据	32bit 数据 A	帧数据	32bit 数据 A
	32bit 数据 B		32bit 数据 B
	32bit 数据 C		32bit 数据 C
	32bit 的高 8bit 为数据 D		32bit 的高 8bit 为数据 D
帧数据	32bit 数据 A	帧时标	无效
	32bit 数据 B		无效
	32bit 数据 C		32bit 的低 16bit 为时标 E
	32bit 的高 8bit 为数据 D		32bit 时标 F（48 位时标的低 32 位）

- 数据 A

ID 信息和帧类型

D31	D30	D29	D28	D27	D26	D25	D24
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
D23	D22	D21	D20	D19	D18	D17	D16
SID2	SID1	SID0	SRR	IDE	0	EID17	EID16
D15	D14	D13	D12	D11	D10	D9	D8
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
D7	D6	D5	D4	D3	D2	D1	D0
EID7	EID6	EID5	EID4	EID3	EID2	EID2	EID0

SRR=1，表示收到标准远程帧；SRR=0，表示收到标准数据帧。

IDE=1，表示收到的是扩展帧；IDE=0，表示收到的是标准帧。

SID 为收到的标准 ID；EID 为收到的扩展 ID;

- 数据 B

帧类型、数据长度

D31	D30	D29	D28	D27	D26	D25	D24
0	RTR	1	0	DLC3	DLC2	DLC1	DLC0
D23	D22	D21	D20	D19	D18	D17	D16
RXB0D0	RXB0D0	RXB0D0	RXB0D0	RXB0D0	RXB0D0	RXB0D0	RXB0D0
D15	D14	D13	D12	D11	D10	D9	D8
RXB0D1	RXB0D1	RXB0D1	RXB0D1	RXB0D1	RXB0D1	RXB0D1	RXB0D1
D7	D6	D5	D4	D3	D2	D1	D0
RXB0D2	RXB0D2	RXB0D2	RXB0D2	RXB0D2	RXB0D2	RXB0D2	RXB0D2

RTR=1 表示收到扩展远程帧；RTR=0 表示收到扩展数据帧；

- 数据 C

接收缓冲区

D31	D30	D29	D28	D27	D26	D25	D24
RXB0D3	RXB0D3	RXB0D3	RXB0D3	RXB0D3	RXB0D3	RXB0D3	RXB0D3
D23	D22	D21	D20	D19	D18	D17	D16
RXB0D4	RXB0D4	RXB0D4	RXB0D4	RXB0D4	RXB0D4	RXB0D4	RXB0D4
D15	D14	D13	D12	D11	D10	D9	D8
RXB0D5	RXB0D5	RXB0D5	RXB0D5	RXB0D5	RXB0D5	RXB0D5	RXB0D5
D7	D6	D5	D4	D3	D2	D1	D0
RXB0D6	RXB0D6	RXB0D6	RXB0D6	RXB0D6	RXB0D6	RXB0D6	RXB0D6

RXB0D<0：3>:接收缓冲区低四字节

- 数据 D

接收缓冲区

D31	D30	D29	D28	D27	D26	D25	D24
RXB0D7	RXB0D7	RXB0D7	RXB0D7	RXB0D7	RXB0D7	RXB0D7	RXB0D7
D23	D22	D21	D20	D19	D18	D17	D16
0	0	0	0	0	0	0	0

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	0
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0

RXB0D<4：7>:接收缓冲区高四字节

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

pRxFrame：用于存储从接收 FIFO 中读出的帧。

返 回 值：返回值为 0 表示函数执行成功。

3.3.2.13 CANX20_RxIntEnable

函数原型：CCAN_INT32 __stdcall CANX20_RxIntEnable (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_BOOL Enabled);

函数功能：使能/禁用接收中断。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

Enabled：值为 1 时，表示使能接收中断；值为 0 时，表示禁用接收中断。

返 回 值：返回值为 0 表示函数执行成功。

3.3.2.14 CANX20_GetIntRxCnt

函数原型：CCAN_INT32 __stdcall CANX20_GetIntRxCnt (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_UINT32 *pCount);

函数功能：中断接收下,获取接收 FIFO 里的数据量。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~9，分别代表第 0~9 通道；

pCount：返回当前接收缓冲区中的数据量。

返 回 值：返回值为 0 表示函数执行成功。

3.3.3 触发功能接口

板卡 PXI 总线背板上有 8 根触发线，另外，外部连接器上有 8 路 TTL 电平触发线，共有 16 根触发线。其中 TTL 外触发线 0~7 与 32 路数字量 I/O 的第 0 组（即第 0~7 通道）复用。

配置触发输出时，触发功能接口中各函数的 Ch 参数取值 0~15，分别对应 16 根触发线，如下 所示：

Ch	触发线选择
0	PXI 触发线 0 有效
1	PXI 触发线 1 有效
2	PXI 触发线 2 有效
3	PXI 触发线 3 有效
4	PXI 触发线 4 有效
5	PXI 触发线 5 有效
6	PXI 触发线 6 有效
7	PXI 触发线 7 有效
8	TTL 外触发线 0 有效
9	TTL 外触发线 1 有效
10	TTL 外触发线 2 有效
11	TTL 外触发线 3 有效
12	TTL 外触发线 4 有效
13	TTL 外触发线 5 有效
14	TTL 外触发线 6 有效
15	TTL 外触发线 7 有效

触发输入引入各个功能模块，如：CAN、I/O、时标功能模块。触发线选择通过触发输入配置结构中 TrigSrc 参数进行配置。

触发配置的相关参数信息，详见“触发输入、输出配置结构”相关说明。16 根触发线中，每一根只能选择为输入或输出的其中一种。

触发应用要求：

- “PXI 触发线”与“TTL 外触发线”作为输出时，允许多个触发功能选择同一个触发线作为“触发输入”，但不允许多个触发功能选择同一个触发线同时作为“触发输出”。
- 只要配置使能了任意一根“TTL 外触发线”，则 32 路数字量 I/O 的第 0 组（即第 0~7 通道）将不可用于数字量输入或输出，即此时数字量输入、输出的第 0~7 通道无效。
- 允许配置相同触发线同时为触发输出和触发输入。当某根触发线被配置为触发输出后，某功能模块又配置选择这根触发线为触发输入，则开启触发输出后，触发输出信号如果符合触发输入的条件，则同样会触发某功能模块。即各功能块的触发输入可以以某触发输出作为自己的触发信号源。
- PXI 总线背板上的每根触发线可单独设置工作模式为输入或输出，而 8 路 TTL 电平触发线只能全部同时为同一方向。
- 当配置 TTL 外触发线触发功能中，存在部分通道为触发输入，部分通道为触发输出时，则最终硬件会自动将所有通道处理应用为触发输入，即 TTL 触发输入配置优先。举例

如下：

例 1：配置 TTL-0 触发线为触发输出，配置串口某通道触发发送选择 TTL-1 触发线（即为串口的触发输入），配置 TTL-3 触发线为触发输出，则此时 8 根 TTL 触发线的触发方向均为“触发输入”。

例 2：基于例 1 的基础上，如果 TTL-1 触发线即配置为串口某通道所需的触发输入，又配置为触发输出，则 8 根 TTL 触发线的触发方向均为“触发输出”。

3.3.3.1 CANX20_TrigReset

函数原型：CANX20_INT32 __stdcall ***CANX20_TrigReset*** (CCAN_HANDLE Handle);

函数功能：复位触发。

参数说明：**Handle**：板卡的句柄。

返 回 值：返回值为 0 表示函数执行成功。

3.3.3.2 CANX20_TrigClkSrc

函数原型：CANX20_INT32 __stdcall ***CANX20_TrigClkSrc*** (CCAN_HANDLE Handle, CCAN_UINT8 ClkSrc);

函数功能：触发时钟源选择。

参数说明：

Handle：板卡的句柄；

ClkSrc：0 表示“系统时钟”，1 表示“PXI 背板时钟”。

返 回 值：返回值为 0 表示函数执行成功。

3.3.3.3 CANX20_TrigCfgOutput

函数原型：CANX20_INT32 __stdcall ***CANX20_TrigCfgOutput*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, pTRIGOUT_CONFIG_STRUCT pstCfgOutput);

函数功能：配置触发输出。

参数说明：

Handle：板卡的句柄；

Ch：触发线号，取值为 0~15；

pstCfgOutput：指定触发输出配置结构的地址。

返 回 值：返回值为 0 表示函数执行成功。

3.3.3.4 CANX20_TrigGetOutputCnt

函数原型：CANX20_INT32 __stdcall ***CANX20_TrigGetOutputCnt*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_UINT32 *pCnt);

函数功能：获取触发输出个数。

参数说明：

Handle：板卡的句柄；

Ch：触发线号，取值为 0~15；

pCnt：返回当前触发输出个数。

返 回 值：返回值为 0 表示函数执行成功。

3.3.3.5 CANX20_TrigStartOutput

函数原型：CANX20_INT32 __stdcall ***CANX20_TrigStartOutput*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_BOOL Enabled);

函数功能：开启、停止触发输出。

参数说明：

Handle：板卡的句柄；

Ch：触发线号，取值为 0~15；

Enabled：值为 1 时，表示开启触发输出；值为 0 时，表示触发输出。

返 回 值：返回值为 0 表示函数执行成功。

3.3.3.6 CANX20_TrigGetOutputStatus

函数原型：CANX20_INT32 __stdcall ***CANX20_TrigGetOutputStatus*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_BOOL *pStatus);

函数功能：获取当前触发输出状态。

参数说明：

Handle：板卡的句柄；

Ch：触发线号，取值为 0~15；

pStatus：返回当前触发输出状态，值为 1 时，表示正在输出；值为 0 时，表示输出停止。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4 数字量 I/O 功能接口

板卡共 16 路数字量通道，每 8 路分为一组，可通过函数参数 GrpNo 进行选择。每组只能配置

为“输入”或“输出”，不能同时存在两个应用方向。

其中第 0 组（即第 0~7 路）I/O 与触发功能复用。

同步采样或同步刷新模式下，输入或输出状态存储采用 FIFO 方式，最大容量为 32 位×8K-1。

3.3.4.1 CANX20_IOSetInputMode

函数原型：CANX20_INT32 __stdcall ***CANX20_IOSetInputMode*** (CCAN_HANDLE Handle, CCAN_UINT8 Mode);

函数功能：设置数字量输入模式。

参数说明：

Handle：板卡的句柄；

Mode：0 表示“普通模式”，1 表示“同步采样模式”。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.2 CANX20_IOSetOutputMode

函数原型：CANX20_INT32 __stdcall ***CANX20_IOSetOutputMode*** (CCAN_HANDLE Handle, CCAN_UINT8 Mode);

函数功能：设置数字量输出模式。

参数说明：

Handle：板卡的句柄；

Mode：0 表示“普通模式”，1 表示“同步刷新模式”。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.3 CANX20_IOInputTrigInConfig

函数原型：CANX20_INT32 __stdcall ***CANX20_IOInputTrigInConfig*** (CCAN_HANDLE Handle, pTRIGIN_CONFIG_STRUCT pstCfgInput);

函数功能：数字量输入的触发配置。当触发输入有效时，同步采样输入触发开启。

参数说明：

Handle：板卡的句柄；

pstCfgInput：指定触发输入配置结构的地址。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.4 CANX20_IOOutputTrigInConfig

函数原型：CANX20_INT32 __stdcall ***CANX20_IOOutputTrigInConfig*** (CCAN_HANDLE Handle,

pTRIGIN_CONFIG_STRUCT pstCfgInput);

函数功能：数字量输出的触发配置。当触发输入有效时，同步刷新输出触发开启。

参数说明：

Handle：板卡的句柄；

pstCfgInput：指定触发输入配置结构的地址。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.5 CANX20_IOAddTimeTag

函数原型：CANX20_INT32 __stdcall ***CANX20_IOAddTimeTag*** (CCAN_HANDLE Handle, CCAN_BOOL Enabled);

函数功能：使能、禁用时标，只有在“同步采样模式”下有效。

参数说明：

Handle：板卡的句柄；

Enabled：值为 1 时，表示使能时标；值为 0 时，表示禁用时标。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.6 CANX20_IOEnable

函数原型：CANX20_INT32 __stdcall ***CANX20_IOEnable*** (CCAN_HANDLE Handle, CCAN_UINT8 GrpNo, CCAN_BOOL Enabled);

函数功能：输出使能控制。

参数说明：

Handle：板卡的句柄；

GrpNo：组号，取值 0~1 分别代表第 0~1 组；

Enabled：值为 1 时，表示输出使能；值为 0 时，表示输出禁止。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.7 CANX20_IOSetDir

函数原型：CANX20_INT32 __stdcall ***CANX20_IOSetDir*** (CCAN_HANDLE Handle, CCAN_UINT8 GrpNo, CCAN_BOOL Input);

函数功能：设置方向，“输入”或“输出”。

参数说明：

Handle：板卡的句柄；

GrpNo：组号，取值 0~1 分别代表第 0~1 组；

Input：值为 1 时，表示此组的 8 路数字量通道均为“输入”；值为 0 时，表示此组的 8 路数字量通道均为“输出”。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.8 CANX20_IOResetInputFIFO

函数原型：CANX20_INT32 __stdcall ***CANX20_IOResetInputFIFO*** (CCAN_HANDLE Handle);

函数功能：复位输入 FIFO，只有在“同步采样模式”下有效。

参数说明：**Handle**：板卡的句柄。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.9 CANX20_IOResetOutputFIFO

函数原型：CANX20_INT32 __stdcall ***CANX20_IOResetOutputFIFO*** (CCAN_HANDLE Handle);

函数功能：复位输出 FIFO，只有在“同步刷新模式”下有效。

参数说明：**Handle**：板卡的句柄。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.10 CANX20_IOGetInputFIFOStatus

函数原型：CANX20_INT32 __stdcall ***CANX20_IOGetInputFIFOStatus*** (CCAN_HANDLE Handle, CCAN_UINT8 *pStatus);

函数功能：获取输入 FIFO 的状态，只有在“同步采样模式”下有效。

参数说明：

Handle：板卡的句柄；

pStatus：返回硬件数字量输入 FIFO 的状态，如下表所示：

位	Bit 7~2	Bit 1	Bit 0
定义	0	满标志	空标志

第 7~2 位的值恒为 0；

满标志：该位为 1 时，表示缓冲区已满，为 0 时，表示不满；

空标志：该位为 1 时，表示缓冲区为空，即没有数据，为 0 时，表示不空。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.11 CANX20_IOGetInputFIFOCnt

函数原型：CANX20_INT32 __stdcall ***CANX20_IOGetInputFIFOCnt*** (CCAN_HANDLE Handle, CCAN_UINT32 *pCount);

函数功能：获取输入 FIFO 里的数据量，只有在“同步采样模式”下有效。

参数说明：

Handle：板卡的句柄；

pCount：返回硬件数字量输入 FIFO 里的状态数据量或帧数量。

返 回 值：返回值为 0 表示函数执行成功。

备　　注：当硬件被配置为“同步采样模式”且“使能时标”时，pCount 返回值为帧数量，一帧包括“32 位数据 + 48 位时标”；当硬件被配置为“同步采样模式”且“禁用时标”时，pCount 返回值为状态数据量，即 32 位状态数据个数。

3.3.4.12 CANX20_IOGetOutputFIFOStatus

函数原型：CANX20_INT32 __stdcall ***CANX20_IOGetOutputFIFOStatus*** (CCAN_HANDLE Handle, CCAN_UINT8 *pStatus);

函数功能：获取输出 FIFO 的状态，只有在“同步刷新模式”下有效。

参数说明：

Handle：板卡的句柄；

pStatus：返回硬件数字量输出 FIFO 的状态，如下表所示：

位	Bit 7~2	Bit 1	Bit 0
定义	0	满标志	空标志

第 7~2 位的值恒为 0；

满标志：该位为 1 时，表示缓冲区已满，为 0 时，表示不满；

空标志：该位为 1 时，表示缓冲区为空，即没有数据，为 0 时，表示不空。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.13 CANX20_IOGetOutputFIFOCnt

函数原型：CANX20_INT32 __stdcall ***CANX20_IOGetOutputFIFOCnt*** (CCAN_HANDLE Handle, CCAN_UINT32 *pCount);

函数功能：获取输入 FIFO 里的数据量，只有在“同步刷新模式”下有效。

参数说明：**hDev**：板卡的句柄。

参数说明：

Handle：板卡的句柄；

pCount：返回硬件数字量输出 FIFO 里的状态数据量。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.14 CANX20_IOGetInputStatus

函数原型：CANX20_INT32 __stdcall ***CANX20_IOGetInputStatus*** (CCAN_HANDLE Handle, CCAN_UINT32 Length, pIODATA_STRUCT pstIoDataBuf, CCAN_UINT32 *prtLength);

函数功能：获取输入状态。

参数说明：

Handle：板卡的句柄；

Length：在“普通模式”时，此值需指定为 1；在“同步采样模式”时，此值为 pstIoDataBuf 指向结构数组中单元数量；

pstIoDataBuf：在“普通模式”时，此指针应指向数字量输入数据结构的地址；在“同步采样模式”时，此指针指向数字量输入数据结构数组的首址，此时 Length 应小于等于结构数组中的单元数量；

prtLength：在“普通模式”时，此值返回为 1；在“同步采样模式”时，此值返回实际从硬件 FIFO 中读出的数量（即 pstIoDataBuf 中返回的单元数量）。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.15 CANX20_IOGetInputStatusSingle

函数原型：CANX20_INT32 __stdcall ***CANX20_IOGetInputStatusSingle*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_BOOL *pStatus);

函数功能：按通道获取输入状态。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~31，分别代表第 0~31 通道；

pStatus：返回当前通道的输入状态，值为 1 时，表示输出为“高电平”；值为 0 时，表示输出为“低电平”。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.16 CANX20_IOSetOutputStatus

函数原型：CANX20_INT32 __stdcall ***CANX20_IOSetOutputStatus*** (CCAN_HANDLE Handle, CCAN_UINT32 Length, CCAN_UINT32 *pstIoDataBuf, CCAN_UINT32 *prtLength);

函数功能：设置输出状态。

参数说明：

Handle：板卡的句柄；

Length：在“普通模式”时，此值需指定为 1；在“同步刷新模式”时，此值应小于等于同步刷新 FIFO 还可容纳的数量；

pstIoDataBuf：在“普通模式”时，此指针应指向一个 32 位的数字量输出数据的地址；在“同步刷新模式”时，此指针指向一个 32 位数字量输出数据数组的首址，此时 Length 应小于等于数组中的单元数量；pstIoDataBuf 中的 32 位数据第 0~31 位分别表示第 0~31 数字量输出通道，位值为 0 表示输出低电平，位值为 1 表示输出高电平；

prtLength：在“普通模式”时，此值返回为 1；在“同步刷新模式”时，此值返回实际写入硬件 FIFO 中的状态数据量。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.17 CANX20_IOSetOutputStatusSingle

函数原型：CANX20_INT32 __stdcall ***CANX20_IOSetOutputStatusSingle*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_BOOL Status);

函数功能：按通道设置输出状态。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~31，分别代表第 0~31 通道；

Status：输出状态控制，值为 1 时，表示输出为“高电平”；值为 0 时，表示输出为“低电平”。

返 回 值：返回值为 0 表示函数执行成功。

3.3.4.18 CANX20_IOGetOutputStatusSingle

函数原型：CANX20_INT32 __stdcall ***CANX20_IOGetOutputStatusSingle*** (CCAN_HANDLE Handle, CCAN_UINT8 Ch, CCAN_BOOL *pStatus);

函数功能：按通道获取输出状态。

参数说明：

Handle：板卡的句柄；

Ch：通道号，取值为 0~31，分别代表第 0~31 通道；

pStatus：返回当前通道的输出状态，值为 1 时，表示输出为“高电平”；值为 0 时，表示输出为“低电平”。

返 回 值：返回值为 0 表示函数执行成功。

3.3.5 时标功能接口

3.3.5.1 CANX20_ConfigTimeTag

函数原型：CANX20_INT32 __stdcall ***CANX20_ConfigTimeTag*** (CCAN_HANDLE Handle, CCAN_UINT8 ClkSrc, CCAN_UINT8 TrigMode);

函数功能：设置时标的时钟源和计数开启模式。

参数说明：

Handle：板卡的句柄；

ClkSrc：0 表示“系统时钟”，1 表示“PXI 背板时钟”；

TrigMode：0 表示“内部触发模式”，1 表示“外部触发模式”。

返 回 值：返回值为 0 表示函数执行成功。

3.3.5.2 CANX20_TimeTagTrigInConfig

函数原型：CANX20_INT32 __stdcall ***CANX20_TimeTagTrigInConfig*** (CCAN_HANDLE Handle, pTRIGIN_CONFIG_STRUCT pstCfgInput);

函数功能：触发开启配置。当触发输入有效时，时标计数器触发开启。

参数说明：

Handle：板卡的句柄；

pstCfgInput：指定触发输入配置结构的地址。

返 回 值：返回值为 0 表示函数执行成功。

3.3.5.3 CANX20_SetTimeTag

函数原型：CANX20_INT32 __stdcall ***CANX20_SetTimeTag*** (CCAN_HANDLE Handle, CCAN_UINT64 InitVal);

函数功能：设置时标计数初值。

参数说明：

Handle：板卡的句柄；

InitVal：时标计数初值，低 48 位有效，分辨率为 1us。

返 回 值：返回值为 0 表示函数执行成功。

3.3.5.4 CANX20_GetTimeTag

函数原型：CANX20_INT32 __stdcall CANX20_GetTimeTag (CCAN_HANDLE Handle, CCAN_UINT64 *pCurVal);

函数功能：获取当前时标计数值。

参数说明：

Handle：板卡的句柄；

pCurVal：返回当前时标计数值，低 48 位有效，分辨率为 1us。

返 回 值：返回值为 0 表示函数执行成功。

3.3.5.5 CANX20_StartTimeTag

函数原型：CANX20_INT32 __stdcall ***CANX20_StartTimeTag*** (CCAN_HANDLE Handle, CCAN_BOOL Enabled);

函数功能：开启、停止时标计数。

参数说明：

Handle：板卡的句柄；

Enabled：值为 1 时，表示开启时标计数；值为 0 时，表示停止时标计数。

返 回 值：返回值为 0 表示函数执行成功。

备 注：时标计数器如果选择为“外部触发模式”，要先调用此函数将计数开启，外部触发脉冲到达时，硬件才会够控制计时器开启。

3.4 CAN 功能驱动接口函数调用步骤

以下调用步骤中的程序代码仅供参考，不作为最终程序可执行代码。

3.4.1 打开板卡

1. 打开板卡（CANX20_Open）
2. 获取板卡信息（CANX20_GetDevBusInfo）
3. 复位板卡（CANX20_Reset）

3.4.2 CAN 模块

3.4.2.1 配置 CAN

1. 设置波特率（CANX20_SetBaud）
2. 滤波和屏蔽模式配置（CANX20_RxMode）
3. 添加时标（CANX20_RxAddTimeTag）
4. 中断接收（CANX20_RxIntEnable）
5. 清空缓冲区（CANX20_ZeroFIFO）

注：配置时，需先设置波特率，再配置滤波和屏蔽模式，重新设置波特率，需对滤波和屏蔽模式重新进行配置！

3.4.2.2 发送数据

- 普通发送

1. 设置 CAN 数据帧发送模式为“普通发送”（CANX20_TxMode(Handle, Ch, 0)）
2. 判断发送 FIFO 状态（CANX20_TxFIFOStatus）
3. 发送 FIFO 不满，向缓冲区写入数据（CANX20_TxWrite）
4. 重复第 2-3 步

- 触发定时发送
  1. 设置 CAN 数据帧发送模式为“定时发送”（CANX20_TxMode (Handle, Ch, 1)）
  2. 向缓冲区写入数据（CANX20_TxWrite）
  3. 重复第 2 步，完成数据写入
  4. 配置 CAN 触发发送（CANX20_TxTrigInConfig）

3.4.2.3 接收数据

- 非中断接收
  1. 读取缓冲区数据量（CANX20_RxFIFOCnt）
  2. 无时标数据量不小于 4（有时标数据量不小于 8）时，读取缓冲区数据（CANX20_RxRead）
  3. 重复第 1-2 步
- 中断接收
  1. 读取缓冲区数据量（CANX20_GetIntRxCnt）
  2. 无时标数据量不小于 4（有时标数据量不小于 8）时，读取缓冲区数据（CANX20_RxRead）
  3. 重复第 1-2 步

3.4.3 触发

- 复位触发（CANX20_TrigReset）
- 触发时钟源选择（CANX20_TrigClkSrc）
- 触发输出
  - 配置并使能触发输出（CANX20_TrigCfgOutput）
  - 结束时停止触发输出和禁用触发

注：触发输出时，可获取触发输出个数（CANX20_TrigGetOutputCnt）和获取触发输出状态（CANX20_TrigGetOutputStatus）

3.4.4 数字量输入、输出

设置方向为“输入”或“输出”（CANX20_IOSetDir）

3.4.4.1 数字量输入（普通模式）

1. 设置数字量输入模式（CANX20_IOSetInputMode(Handle, 0)）
2. 使能输入（CANX20_IOEnable(Handle, GrpNo, 1)）
3. 获取输入状态（CANX20_IOGetInputStatus(Handle, 1, pstIoDataBuf, prtLength)）
4. 重复第 3 步，结束时需禁止输入（CANX20_IOEnable(Handle, GrpNo, 0)）

3.4.4.2 数字量输入（同步采样模式）

1. 设置数字量输入模式（CANX20_IOSetInputMode(Handle, 1)）
2. 复位输入 FIFO（CANX20_IOResetInputFIFO）
3. 如需时标时，使能时标（CANX20_IOAddTimeTag）
4. 数字量输入的触发配置（CANX20_IOInputTrigInConfig）
5. 使能输入（CANX20_IOEnable(Handle, GrpNo, 1)）
6. 读取输入状态数据（CANX20_IOGetInputStatus(Handle, Length, pstIoDataBuf, prtLength)）
7. 重复第 6 步，结束时需禁止输入（CANX20_IOEnable(Handle, GrpNo, 0)）

3.4.4.3 数字量输出（普通模式）

1. 使能输出（CANX20_IOEnable(Handle, GrpNo, 1)）
2. 输出时可调用下列函数进行输出状态的控制与读取：
   CANX20_IOSetOutputStatusSingle
   CANX20_IOGetOutputStatusSingle
3. 结束时需禁止输出（CANX20_IOEnable(Handle, GrpNo, 0)）

3.4.4.4 数字量输出（同步刷新模式）

1. 设置数字量输出模式（CANX20_IOSetOutputMode(Handle, 1)）
2. 复位输出 FIFO（CANX20_IOResetOutputFIFO）
3. 数字量输出的触发配置（CANX20_IOOutputTrigInConfig）
4. 使能输出（CANX20_IOEnable(Handle, GrpNo, 1)）
5. 向 FIFO 中写入状态数据（CANX20_IOSetOutputStatus(Handle, Length, pstIoDataBuf, prtLength)）
6. 输出时可继续向 FIFO 中写入状态数据（CANX20_IOSetOutputStatus(Handle, Length, pstIoDataBuf, prtLength)）
7. 禁止输出（CANX20_IOEnable(Handle, GrpNo, 0)）

3.4.5 时标

1. 设置时标时钟源和计数开启模式（CANX20_ConfigTimeTag）
2. 设置时标计数初值（CANX20_SetTimeTag）
3. 触发开启配置（C CANX20_TimeTagTrigInConfig）
4. 开启时标计数（CANX20_StartTimeTag(Handle, 1)）
5. 可随时获取当前时标计数值（CANX20_GetTimeTag）
6. 结束时停止时标计数（CANX20_StartTimeTag(Handle, 0)）

3.4.6 关闭板卡

建议在关闭板卡前做好以下工作后，再关闭板卡：

- 停止 CAN 触发定时发送，停止接收
- 禁用数字量输入或输出
- 停止时标计数
- 关闭板卡（CANX20_Close）

第四章　功能演示软件

CAN-X-10 应用程序用于实现本公司 CAN-X-XX 板卡的 CAN 通讯的基本操作。本软件为了解与使用 CAN-X-XX 板卡的 CAN 通讯、数字量输入输出、触发输出、时标等功能提供了方便的途径，利用本软件，可轻松地对板卡进行操作。

4.1 使用环境

4.1.1 系统要求

- 内存：256M 以上
- CPU：800MHz 以上
- 显示分辨率：800×600 以上

4.1.2 操作系统

Windows：Win2000，WinXP/Win7 (X86，X64)操作系统

4.2 开发工具

Microsoft Visual Studio C++ 7.1

4.3 界面说明

4.3.1 启动应用程序

- 功能主程序

  程序名称：CANX10App.exe

  运行“CANX10App.exe”，将显示 CAN-X-10 所有板卡及板卡基本信息。

  双击打开对应板卡，将进入“CAN-X-10App 界面”，如下图 4-3

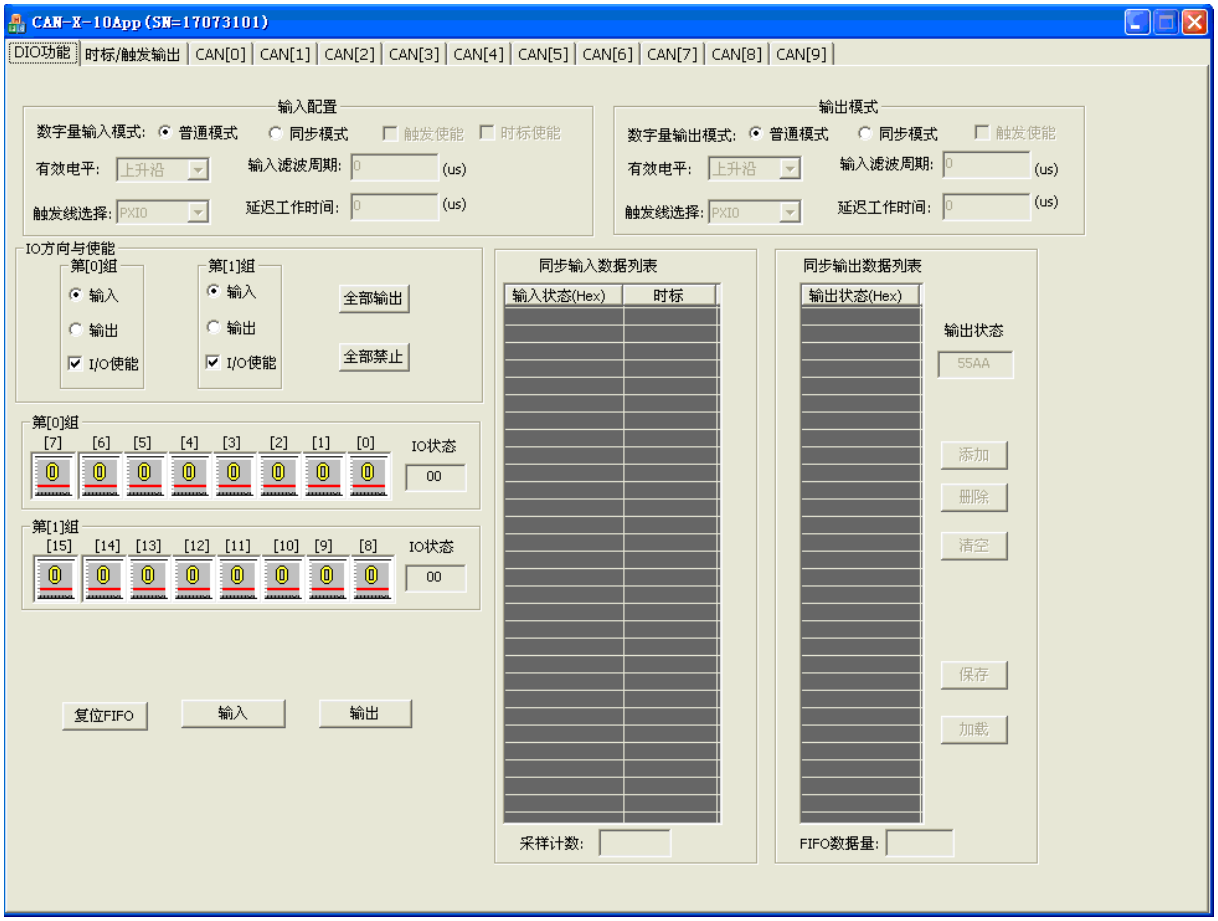


图 4-3 CAN-X-10 功能主程序界面

4.3.2 CAN 通讯部分

CAN 通讯界面主要供用户使用该 CAN 设备进行 CAN 通讯操作，主要包含数据的发送和数据的接收功能，CAN0~ 9 表示 10 个 CAN 功能窗口，如下图

图 4-4 CAN 通讯界面

操作步骤：

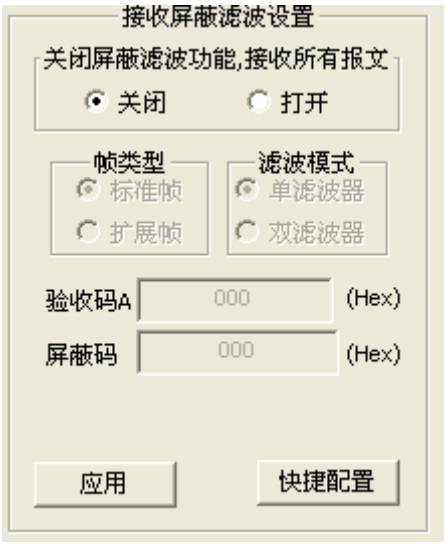
- 波特率设置，选择 

- 发送

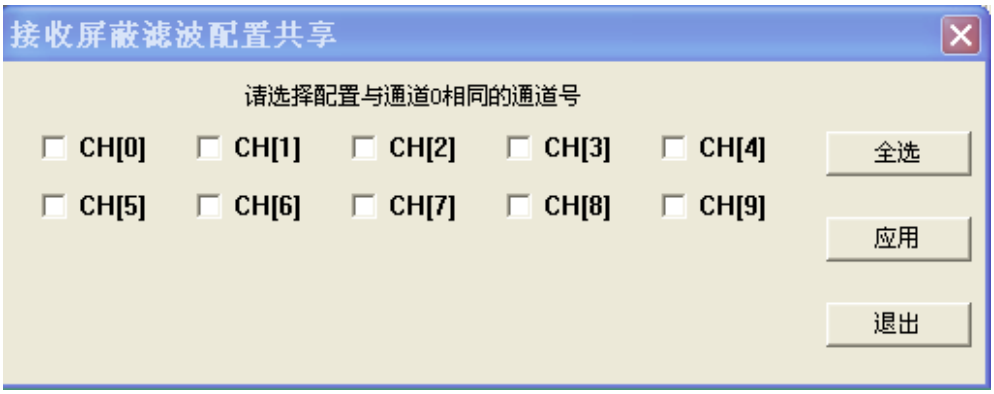
  发送区域如下图：

  可选择帧类型，帧格式，输入帧 ID，选择数据量，输入数据，添加帧到发送帧列表，点击“发送”按钮将发送帧列表中的帧依次发送。
- 接收
  - 接收屏蔽滤波设置

设置界面如下图



1) 关闭屏蔽滤波功能，接收所有报文；
2) 打开屏蔽滤波功能，选择帧类型为扩展帧：只接收符合滤波条件的扩展帧，不管是扩展数据帧还是扩展远程帧；
3) 打开屏蔽滤波功能，选择帧类型为标准帧：只接收符合滤波条件的标准帧，不管是标准数据帧还是标准远程帧；
4) 快捷配置



若要对多个 CAN 通道进行相同配置，可以使用快捷配置功能：完成当前 CAN 通道的配置后，单击“快捷配置”按钮打开快捷配置界面，选择采用相同配置的通道号后，单击“应用”按钮，则所选的通道均采用与当前通道相同的配置；

- 接收帧

可直接接收帧，或者选择时标使能/中断使能后再接收。注:时标为帧接收到的时间标签。

4.3.3 DIO 功能

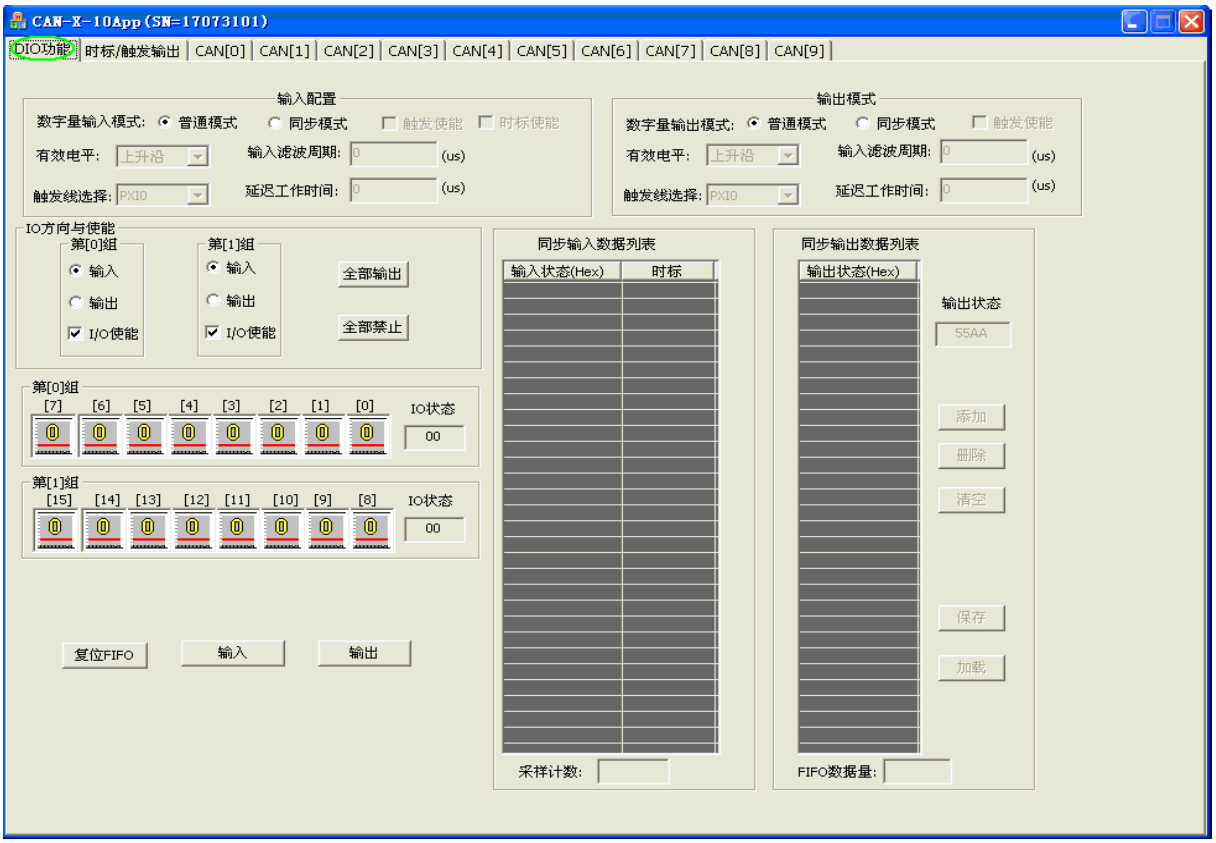
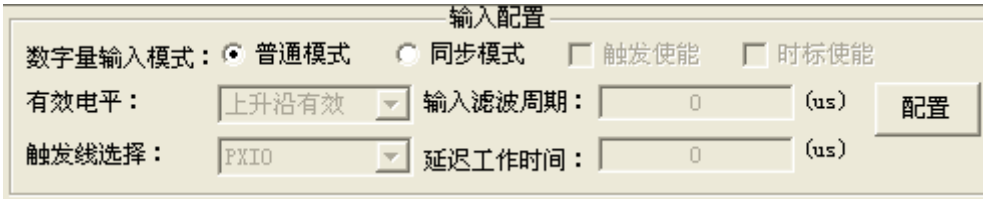


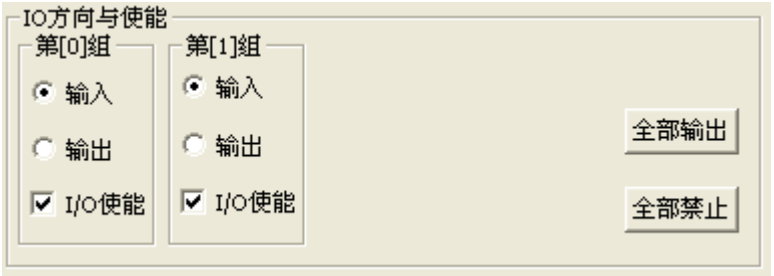
图 4-6 DIO 功能窗口

- **普通输入：**

1、 将数字量输入模式设置为普通模式；



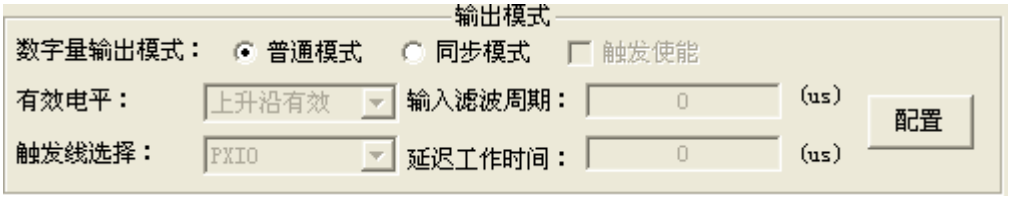
2、 分别设置 2 组的输入输出模式，将需要输入的组设置成输入，设置 0~1 组的 I/O 使能；



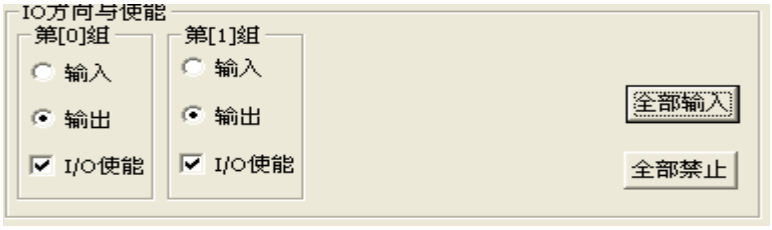
3、 点击输入按钮进行输入。

➢ 普通输出：

1、 将数字量输出模式设置为普通模式；



2、 分别设置 4 组的输入\输出模式，将需要输出的组设置成输出，设置 0~3 组的 I/O 使能；

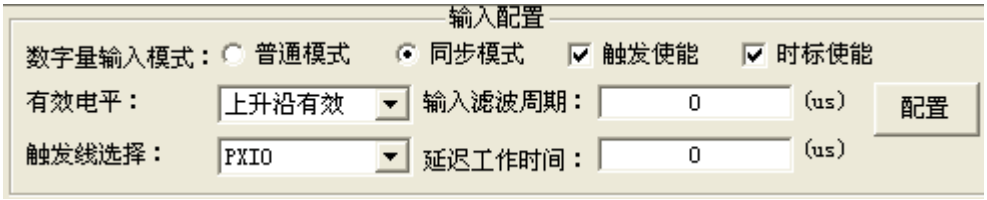


3、 设置输出组各个通道状态，点击通道下的图片进行高低的切换，   ，注：只有设置成输出的组，且 I/O 使能开启的组才能进行高低电平的切换；

4、 点击输出按钮进行输出。

➢ 同步输入：

1、 将数字量输入模式设置为同步模式；



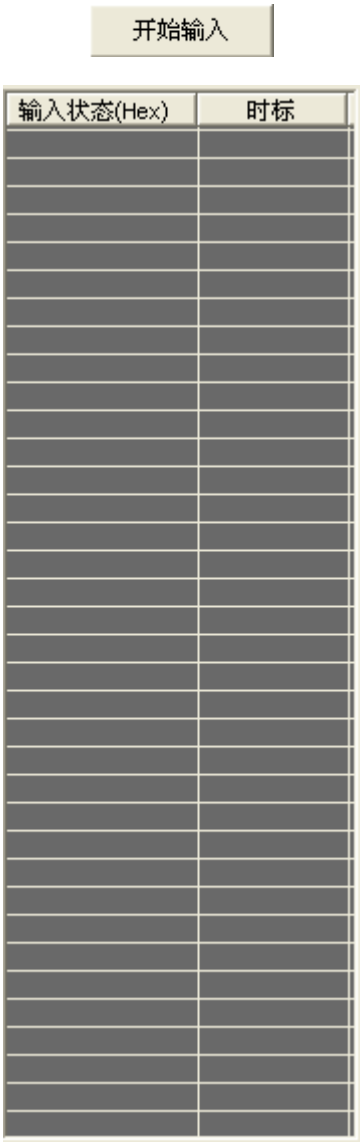
2、 同步模式需要设置触发使能、有效电平（上升沿有效/下降沿有效可选）、触发线选择（PXI0~7/TTL0~7 可选）、滤波周期（单位 1us）、延迟工作时间（单位 1us）；

3、 选择是否使能时标；

4、 分别设置 2 组的输入输出模式，将需要输入的组设置成输入，设置 0~1 组的 I/O 使能；

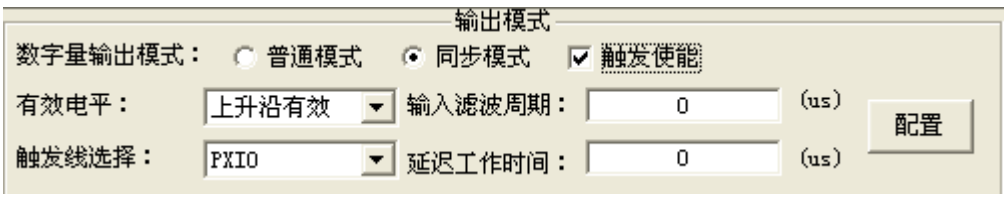
5、 复位同步输入 FIFO；

6、 点击开始输入进行同步输入，当输入开始时“开始输入”按钮变成“停止输入”按钮，点击“停止输入”按钮停止输入，其中同步输入的数据显示在下面的列表中。



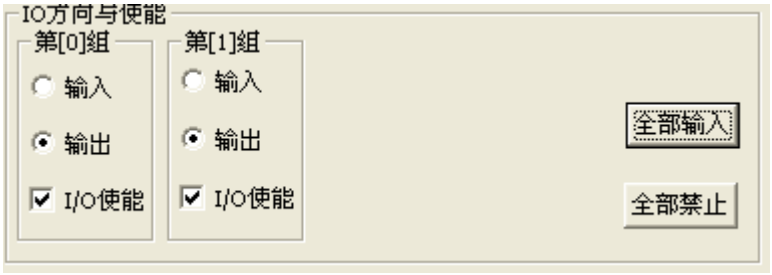
- **同步输出：**

1、 将数字量输出模式设置为同步模式；

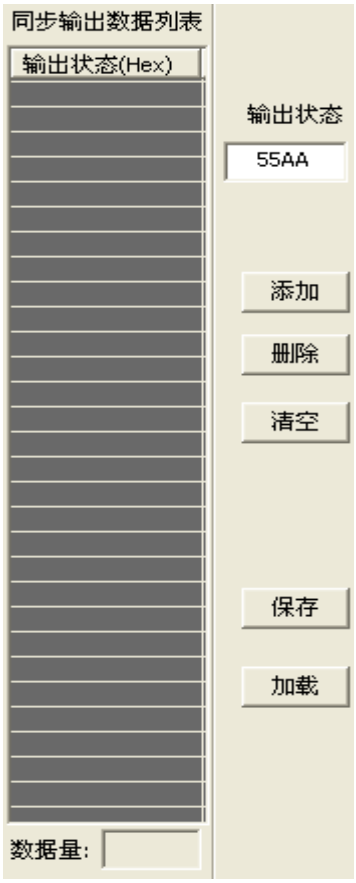


2、 同步模式需要设置触发使能、有效电平（上升沿有效/下降沿有效可选）、触发线选择（PXI0~7/TTL0~7 可选）、滤波周期（单位 1us）、延迟工作时间（单位 1us）；

3、 分别设置 4 组的输入/输出模式，将需要输出的组设置成输出，设置 0~3 组的 I/O 使能；



4、 通过“添加”按钮、“删除”按钮、“清空”按钮、“保存”按钮、“加载”按钮对输出列表进行编辑；



5、 点击输出按钮进行同步输出。

4.3.4 时标

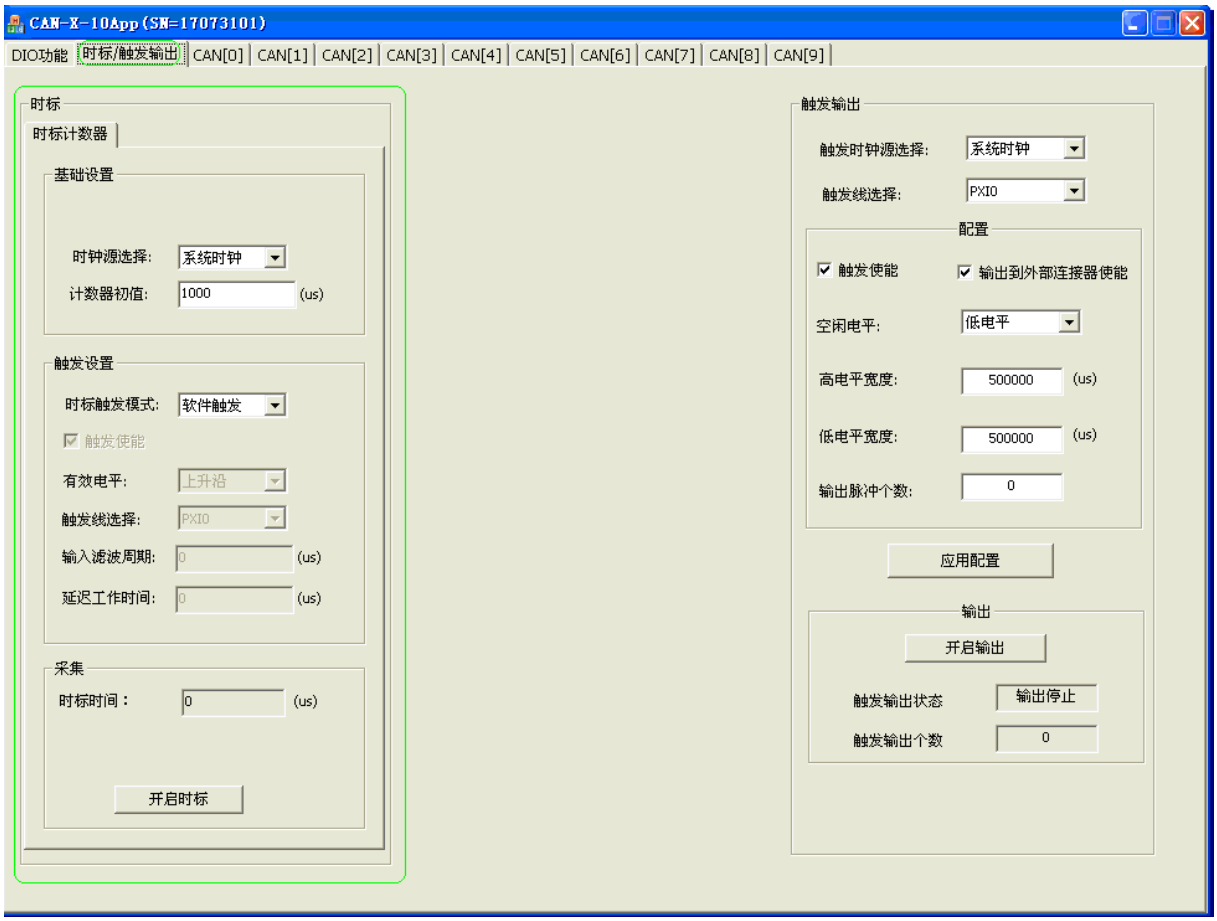


图 4-7 时标功能窗口

➢ 时标操作步骤：

1、 选择时标计数器的时钟源，系统时钟/PXI 背板时钟可选；

2、 设置时标计数器的初值，分辨率为 1us；

3、 设置定时触发模式，软件触发/硬件触发可选；

4、 硬件触发需要设置触发使能、有效电平（上升沿有效/下降沿有效可选）、触发线选择（PXI0~7/TTL0~7 可选）、滤波周期（单位 1us）、延迟工作时间（单位 1us）；

5、 单击“开启时标”按钮可以开启时标计数器的运行，时标时间将实时更新时标计数器的值，单击“停止时标”按钮可以停止时标计数器的运行；

4.3.5 触发输出功能

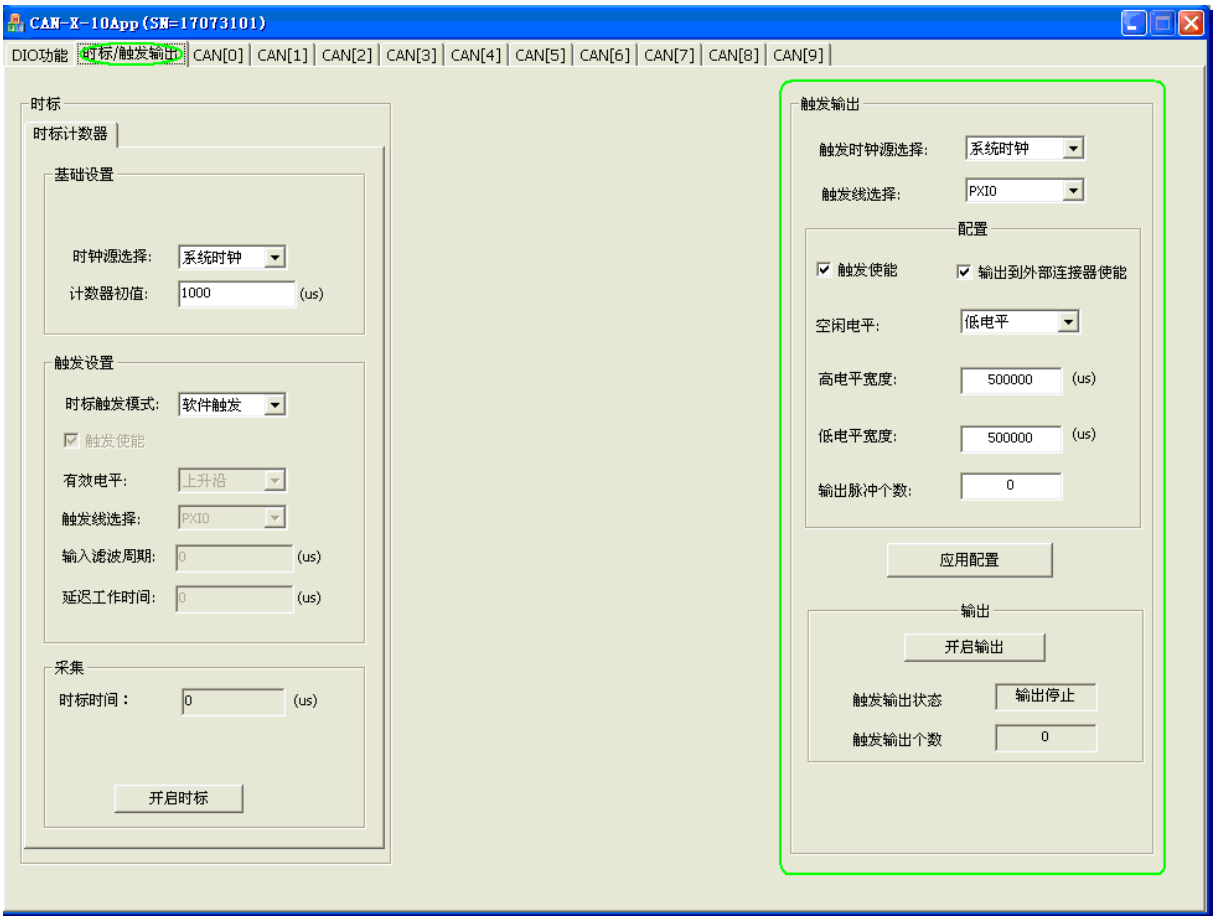


图 4-8 触发输出功能窗口

➢ **触发输出设置步骤：**

1、 选择触发时钟源，系统时钟/PXI 背板时钟可选；

2、 设置触发使能/不使能、是否输出到外部连接器使能、空闲电平（高电平/低电平可选）、高电平宽度（单位 1us）、低电平宽度（单位 1us）、输出脉冲个数；

3、 点击“设置”按钮进行配置。

➢ **触发输出步骤：**

1、 点击“开始输出”按钮进行触发输出、开始输出后“开始输出”按钮会变成“停止输出”按钮，点击“停止输出”按钮停止输出；