

ARINC429-PCI-42

用户手册

V1.0

西安方元明科技发展有限公司

目录

第一章 概述.....	1
1.1 功能.....	1
1.2 基本工作环境.....	1
1.3 安装.....	1
第二章 硬件使用说明.....	2
2.1 功能说明.....	2
2.2 板卡功能结构图如下所示:	2
2.3 配置与安装.....	2
2.3.1 配置方法.....	2
2.3.2 安装步骤.....	2
2.4 硬件结构.....	3
2.5 硬件通讯接口定义.....	3
2.6 可编程功能寄存器描述.....	3
2.6.1 COMREG.....	4
2.6.2 SETREG.....	4
2.6.3 TRIMTn	5
2.6.4 TRIMRn	5
2.6.5 FIFOSTATE	5
2.6.6 STTREGn.....	6
2.6.7 EnReadFIFO	6
2.6.8 BTTREGn	6
2.6.9 FIFOTCOUNT	7
2.6.10 FIFORCOUNT.....	7
2.6.11 CFDATA0.....	7
2.6.12 CFDATA1	8
2.6.13 STR	9
第三章 驱动接口说明.....	11
3.1 功能说明.....	11
3.2 运行软件环境描述.....	11
3.3 驱动程序文件说明.....	11
3.4 安装.....	11
3.5 接口说明.....	12
3.5.1. 结构与定义.....	12
3.5.2. 函数接口描述.....	13
3.6 操作建议.....	23
第四章 应用程序说明	26

4.1 软件简介.....	26
4.2 运行环境.....	26
4.3 安装.....	26
4.4 软件界面.....	27
4.4.1 板卡号选择窗口.....	27
4.4.2 应用程序主窗口.....	27
4.4.3 板卡设置窗口.....	28
4.5 菜单功能.....	28
4.6 操作建议.....	28
4.6.1 开启应用程序.....	28
4.6.2 板卡号选择.....	29
4.6.3 设置板卡.....	29
4.6.4 数据通讯操作.....	29
4.6.5 退出应用程序.....	29
4.7 使用说明.....	30
4.7.1 硬件设置.....	30
4.7.2 数据的接收与发送.....	34
4.8 文件格式.....	36
4.9 编程举例.....	36
附录A: 429 数据格式转换.....	37

手册内容简介

本手册主要提供了四个章节，分别为概述、硬件使用说明、驱动接口说明和应用程序说明。

第一章—概述，是对硬件功能与使用方法的整体概述，更多的细节会在第二章—硬件使用说明、第三章—驱动接口说明和第四章—应用程序说明中进行详细描述。

第二章—硬件使用说明，描述了硬件的使用环境、配置安装方法、硬件结构、寄存器描述及用户在实际可能用到的硬件接口。

第三章—驱动接口说明，该部分内容是本手册的重点内容，它详细描述了接口函数的功能及使用方法，通过这一章节的了解，用户可以进行产品的应用程序开发。

第四章—应用程序说明，介绍了提供给用户的应用程序的使用方法，通过现提供的应用程序，用户不需要编程而直接进行一些基本的板卡操作，达到快速应用的目的。另外，在该章节中，也提供了驱动接口使用例程，为用户开发适合个人要求的应用程序提供参考。

第一章 概述

1.1 功能

- 1) PCI 总线接口;
- 2) 2 路发送通道, 每路带 127x32-Bit FIFO;
- 3) 4 路接收通道, 每路带 255x32-Bit FIFO;
- 4) 具有定时发送功能;
- 5) 标号过滤功能;
- 6) 添加时间标签功能;
- 7) FIFO 触发深度可设功能;
- 8) 波特率 100K、48K、12.5K 可设置等功能;

1.2 基本工作环境

操作系统: Windows 98/2000/NT/XP

工作温度: $-40 \sim +85^{\circ}\text{C}$

相对湿度: 0 ~ 95%

1.3 安装

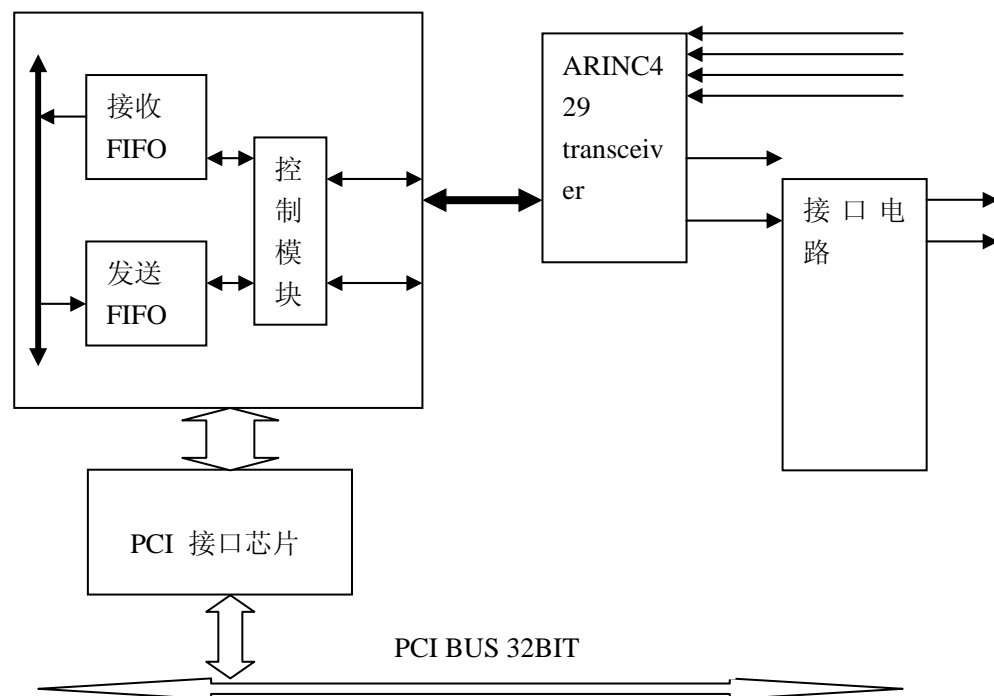
- 硬件安装: 将板卡插入PCI主机箱的设备插槽(插槽有板卡导槽并且有防反插功能), 直至将板卡完全插入插槽。
- 驱动程序安装: 详见3.4节
- 应用程序安装: 详见4.3节

第二章 硬件使用说明

2.1 功能说明

- 1) PCI 总线接口;
- 2) 2 路发送通道, 每路带 127x32-Bit FIFO;
- 3) 4 路接收通道, 每路带 255x32-Bit FIFO;
- 4) 具有定时发送功能;
- 5) 标号过滤功能;
- 6) 添加时间标签功能;
- 7) FIFO 触发深度可设功能;
- 8) 波特率 100K、48K、12.5K 可设置等功能;

2.2 板卡功能结构图如下所示:



2.3 配置与安装

2.3.1 配置方法

本板卡在使用之前不需要进行硬件配置, 所有的初始化都由软件操作

2.3.2 安装步骤

2.4 硬件结构

硬件的外观如下所示：

暂无

2.5 硬件通讯接口定义

外部设备通讯连接头：J2 SCSI36 说明如下

管脚	名称	说明	管脚	名称	说明
1	IN4A	第四路输入 A 端	13	IN2A	第二路输入 A 端
2	RES	保留	14	RES	保留
3	IN4B	第四路输入 B 端	15	IN2B	第二路输入 B 端
4	RES	保留	16	RES	保留
5	GND	系统地	17	GND	系统地
6	GND	系统地	18	GND	系统地
7	IN3A	第三路输入 A 端	19	IN1A	第一路输入 A 端
8	OUT02A	第二路输出 A 端	20	OUT01A	第一路输出 A 端
9	IN3B	第三路输入 B 端	21	IN1B	第一路输入 B 端
10	OUT02B	第二路输出 B 端	22	OUT01B	第一路输出 B 端
11	GND	系统地	23	GND	系统地
12	GND	系统地	24	GND	系统地

2.6 可编程功能寄存器描述

名称	描述	读写	位宽	地址偏移
COMREG	命令寄存器	写	32	0x00
SETREG	设置寄存器	读、写	32	0x04
RFIFOREG1	第 1 路接收 FIFO	读	32	0x08
RFIFOREG2	第 2 路接收 FIFO	读	32	0x0c
RFIFOREG3	第 3 路接收 FIFO	读	32	0x10
RFIFOREG4	第 4 路接收 FIFO	读	32	0x14
TRIMT1	第 1 路发送触发深度设置寄存器	读、写	8	0x18
TRIMT2	第 2 路发送触发深度设置寄存器	读、写	8	0x1a
TRIMR1	第 1 路接收触发深度设置寄存器	读、写	8	0x1c
TRIMR2	第 2 路接收触发深度设置寄存器	读、写	8	0x1d
TRIMR3	第 3 路接收触发深度设置寄存器	读、写	8	0x1e
TRIMR4	第 4 路接收触发深度设置寄存器	读、写	8	0x1f
FIFOSTATE	FIFO 状态寄存器	读	32	0x20
STTREG1	第 1 路字发送定时时间设置寄存器	写	16	0x20
STTREG2	第 2 路字发送定时时间设置寄存器	写	16	0x24

FIFOTCOUNT	实际发送数据深度寄存器	读	32	0x24
FIFORCOUNT	实际接受数据深度寄存器	读	32	0x28
CFDATA0	配置寄存器 0	写	16	0x28
CFDATA1	配置寄存器 1	写	16	0x2A
STR	状态寄存器	读	16	0x2e
TFIFOREG1	第 1 路发送 FIFO	写	32	0x30
TFIFOREG2	第 2 路发送 FIFO	写	32	0x38
EnReadFIFO	使能读接收 FIFO	读、写	16	0x4c
BTTREG1	第 1 路群发送定时时间设置寄存器	写	16	0x50
BTTREG2	第 2 路群发送定时时间设置寄存器	写	16	0x54
TABW	标号过滤设置寄存器	写	32	0X2000-0X2FFF
TABR	标号过滤状态寄存器	读	32	0X3000-0X3FFF

2.6.1 COMREG

命令寄存器 16 位 写 00H

0X 00	cfstarta=0	0X01	cfstarta=1	配置接收 1、2 路和发送 1 路	复位 1
0X02	cfstartb=0	0x03	cfstartb=1	配置接收 3、4 路和发送 2 路	复位 1
0x04	entimer1=0	0x05	entimer1=1	使能 1 路字定时发送	复位 0
0x08	entimer2=0	0x09	entimer2	使能 2 路字定时发送	复位 0
0x0c	tstarta=0	0x0d	tstarta=1	接收 1、2 路时标开始	复位 0
0x0e	tstartb=0	0x0f	tstartb=1	接收 3、4 路时标开始	复位 0
0x10	sysrst=0	0x11	sysrst=1	系统软件复位	复位 0
0x12	t1ten=0	0x13	t1ten=1	使能 1 路群定时发送	复位 0
0x16	t2ten=0	0x17	t2ten=1	使能 2 路群定时发送	复位 0

2.6.2 SETREG

设置寄存器 32 位 读、写 04H

D31	D30	D29	D28	D27	D26	D25	D24
0	0	0	ren4	ren3	ren2	ren1	X
D23	D22	D21	D20	D19	D18	D17	D16
X	X	X		X	X	X	X
D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	clksetb	clkseta
D7	D6	D5	D4	D3	D2	D1	D0
entlabel4	entlabel3	entlabel2	entlabel1	Enfilterr4	Enfilterr3	Enfilterr2	enfilterr1

X: 系统保留 写入 0

D0:	enfilterr1 使能接收 1 路标号过滤	设置为 1 时, 使能标号过滤	复位 0
D1:	enfilterr2 使能接收 2 路标号过滤	设置为 1 时, 使能标号过滤	复位 0
D2:	enfilterr3 使能接收 3 路标号过滤	设置为 1 时, 使能标号过滤	复位 0
D3:	enfilterr4 使能接收 4 路标号过滤	设置为 1 时, 使能标号过滤	复位 0
D4:	entlabel1 使能接收 1 路时标	设置为 1 时 时标添加	复位 0
D5:	entlabel2 使能接收 2 路时标	设置为 1 时 时标添加	复位 0
D6:	entlabel3 使能接收 3 路时标	设置为 1 时 时标添加	复位 0
D7:	entlabel4 使能接收 4 路时标	设置为 1 时 时标添加	复位 0
D8:	clkseta 接收 1、2 路 发送 1 路波特率选择		复位 0
D9:	clksetb 接收 3、4 路 发送 2 路波特率选择		复位 0

注: Clkseta,clksetb 设置为 0 时, 波特率为高速模式 (100k、12.5k) 1 位低速模式 (48k)

D25:	ren1 接收 1 路使能	1: 使能接收 0: 不使能接收	复位为 0
D26:	ren2 接收 2 路使能	1: 使能接收 0: 不使能接收	复位为 0
D27:	ren3 接收 3 路使能	1: 使能接收 0: 不使能接收	复位为 0
D28:	ren4 接收 4 路使能	1: 使能接收 0: 不使能接收	复位为 0

2.6.3 TRIMTn

第 n 路发送深度设置寄存器 8 位 读、写

每路发送深度设置寄存器的值可以设置为 0 到 127 之间的数据。

2.6.4 TRIMRn

第 n 路接收深度设置寄存器 8 位 读、写

一路接收深度设置: bit7→bit0

二路接收深度设置: bit15→bit8

三路接收深度设置: bit23→bit16

四路接收深度设置: bit31→bit24

2.6.5 FIFOSTATE

32 位 FIFO 状态寄存器 读 20H

D31	D30	D29	D28	D27	D26	D25	D24
X	X	X	X	TFU2	TEM2	TOV2	TTR2
D23	D22	D21	D20	D19	D18	D17	D16
X	X	X	X	TFU1	TEM1	TOV1	TTR1
D15	D14	D13	D12	D11	D10	D9	D8

RFU4	REM4	ROV4	RTR4	RFU3	REM3	ROV3	RTR3
D7	D6	D5	D4	D3	D2	D1	D0
RFU2	REM2	ROV2	RTR2	RFU1	REM1	ROV1	RTR1

n 代表 1 到 4 X: 系统保留

RTRn: n 路接收 FIFO 触发发生 当接收的数据大于数据深度设置寄存器时, 为 1。

ROVn: n 路接收 FIFO 溢出错误 当为 1 时, 接收 FIFO 溢出错误

REMn: n 路接收 FIFO 空, 当 1 时指示空, 否则为非空

RFUn: n 路接收 FIFO 满, 当 1 时指示满, 否则为不满。

TTRn: n 路发送 FIFO 触发发生 当发送的数据小于发送数据深度设置寄存器时, 为 1。

TOVn: n 路发送 FIFO 溢出错误 当为 1 时, 发送 FIFO 溢出错误。

TEMn: n 路发送 FIFO 空标志 当 1 时指示 FIFO 空, 否则非空

TFUn: n 路发送 FIFO 满标志 当 1 时指示 FIFO 满, 否则不满

2.6.6 STTREGn

第 n 路字发送定时时间设置寄存器 16 位 20H-26H

定时说明: 定时分辨率是 50Us

定时最大时间是 $50Us \times 65535 = 3.27675$

定时时间: $50US \times$ 发送第 n 路定时时间设置寄存器值

2.6.7 EnReadFIFO

16 位 使能读接收 FIFO 读、写 4cH

D15	D14	D13	D12	D11	D10	D9	D8
D7	D6	D5	D4	D3	D2	D1	D0
				EnReadFIFO4	EnReadFIFO3	EnReadFIFO2	EnReadFIFO1

D0: EnReadFIFO1 使能读第 1 路接收 FIFO

D1: EnReadFIFO2 使能读第 2 路接收 FIFO

D2: EnReadFIFO3 使能读第 3 路接收 FIFO

D3: EnReadFIFO4 使能读第 4 路接收 FIFO

1: 使能读接收 FIFO 0: 禁止读接收 FIFO

2.6.8 BTTREGn

第 n 路群发送定时时间设置寄存器 16 位 50H-56H

定时说明：定时分辨率是 50Us

定时最大时间是 $50\text{Us} \times 65535 = 3.27675$

定时时间： $50\text{US} \times$ 发送第 n 路定时时间设置寄存器值

2.6.9 FIFOTCOUNT

实际发送数据深度寄存器 32 位 读 24H

一路实际发送深度：bit6→bit 0;

二路实际发送深度：bit22→bit16

其它位保留

2.6.10 FIFORCOUNT

实际接受数据深度寄存器 32 位 读 28H

一路实际接收深度：bit7→bit0

二路实际接收深度：bit15→bit8

三路实际接收深度：bit23→bit16

四路实际接收深度：bit31→bit24

2.6.11 CFDATA0

配置寄存器 16 位 写 28H

D15	D14	D13	D12	D11	D10	D9	D8
X	Rclksel12	Tclksel1	Parity1	R2B10	R2B9	Rdec2	R1B10
D7	D6	D5	D4	D3	D2	D1	D0
R1B9	Rdec1	Self test	Paren1	X	X1	X	X

X：系统保留位,写入 0。

D4: paren1 第一路发送奇偶校验使能，1 bit31 是校验位，0 bit31 位是数据位

D5: sleftest 第一路发送、第一路接收和第二路接收自检使能

1 正常工作状态 0 自检工作状态

在自检状态下，第 1 路发送通道的输出，在内部被连接到第 1、2 路接收通道。

其中，数据在进入第 2 路接收通道前被反相，以便与第 1 路接收通道区分。自检状态下，发送通道仍然正常输出。

D6: rdec1 第 1 路接收通道 S/D 码检测使能。

1： 使能第 1 路接收通道，源/目标 译码器。 0： 关闭第 1 路接收通道，源/目标 译码器。

D7、D8: R1B9、R1B10 第 1 路接收通道 S/D 码。

如果 rdec1=1,第 1 路接收通道将收到 ARINC429 数据的 S/D 与 R1B9,R1B10 进行比较,相同则接纳,不相同则丢弃。其中,R1B9 与串行数据 bit9 比较,R1B10 与串行数据 bit10 比较。

D9: rdec2 第 2 路接收通道 S/D 码检测使能。

1: 使能第 1 路接收通道,源/目标 译码器。 0: 关闭第 1 路接收通道,源/目标 译码器。

D10、D11: R2B9、R2B10 第 2 路接收通道 S/D 码。

如果 rdec2=1,第 1 路接收通道将收到 ARINC429 数据的 S/D 与 R2B9,R2B10 进行比较,相同则接纳,不相同则丢弃。其中,R2B9 与串行数据 bit9 比较,R2B10 与串行数据 bit10 比较。

D12: parity1 发送 1 路校验模式选择

0 奇校验 1 偶校验

D13: tclksel1 发送 1 路波特率模式选择

0 100k 1 12.5k

注: 复位后全为 0

2.6.12 CFDATA1

配置寄存器 16 位 写 0X2AH

D15	D14	D13	D12	D11	D10	D9	D8
X	Tclksel34	Tclksel3	Parity3	R4B10	R4B9	Rdec4	R3B10
D7	D6	D5	D4	D3	D2	D1	D0
R3B9	Rdec3	Self test	Paren3	X	X	X	X

D4: paren3 第二路发送奇偶校验使能 1 bit31 是校验位, 0 bit31 位是数据位

D5: slefetest 第二路发送、第三路接收和第四路接收自检使能

1 正常工作状态 0 自检工作状态

在自检状态下,第二路发送通道的输出,在内部被连接到第 3、4 路接收通道。其中,数据在进入第 4 路接收通道前被反相,以便与第 3 路接收通道区分。自检状态下,发送通道仍然正常输出。

D6: rdec3 第 3 路接收通道 S/D 码检测使能。

1: 使能第 1 路接收通道,源/目标 译码器。 0: 关闭第 1 路接收通道,源/

目标 译码器。

D7、D8: R3B9、R3B10 第 3 路接收通道 S/D 码。

如果 rdec3=1, 第 3 路接收通道将收到 ARINC429 数据的 S/D 与 R3B9,R3B10 进行比较, 相同则接纳, 不相同则丢弃。其中, R3B9 与串行数据 bit9 比较, R3B10 与串行数据 bit10 比较

D9: rdec4 第 4 路接收通道 S/D 码检测使能。

1: 使能第 1 路接收通道, 源/目标 译码器。 0: 关闭第 1 路接收通道, 源/目标 译码器。

D10、D11: R4B9、R4B10 第 4 路接收通道 S/D 码。

如果 rdec4=1, 第 4 路接收通道将收到 ARINC429 数据的 S/D 与 R4B9,R4B10 进行比较, 相同则接纳, 不相同则丢弃。其中, R4B9 与串行数据 bit9 比较, R4B10 与串行数据 bit10 比较。

D12: parity3 发送 2 路校验模式选择

0 奇校验 1 偶校验

D13: tclksel3 发送 2 路波特率模式选择

0 100k 1 12.5k

D14: rclksel34 接收 3 路、4 路波特率模式选择

0 100k 1 12.5k

2.6.13 STR

状态寄存器 16 位 读 2CH

D15	D14	D13	D12	D11	D10	D9	D8
0	X	X	X	t3ten	X	t1ten	sysrst
D7	D6	D5	D4	D3	D2	D1	D0
tstartb	tstarta	X	entimer3	X	entimer1	cfstartb	cfstarta

X: 系统保留位, 写入 0

D0: 配置接收 1、2 路 发送 1 路数据格式开始 把 CFDATA 低两字节写到 3282 中的控制字中

D1: 配置接收 3、4 路 发送 2 路数据格式开始 把 CFDATA 高两字节写到 3282 中的控制字中

D2: 使能 1 路定时字发送 1 1 路定时发送模式 0 1 路普通模式发送

D4: 使能 2 路定时字发送 1 2 路定时发送模式 0 3 路普通模式发送

D6: tstarta 接收 1、2 路时标开始

D7: tstartb 接收 3、4 路时标开始

D8: 系统复位指示 1 复位状态 0 工作状态

D9: 使能 1 路群定时发送 1 1 路群定时发送 0 1 路普通模式发送

D11: 使能 2 路群定时发送 1 2 路群定时发送 0 1 路普通模式发送

第三章 驱动接口说明

3.1 功能说明

- PCI 的数据接口总线
- ARINC429 通讯板
- 32 位总线接口
- 4 路接收通道，2 路发送通道
- 采用 FIFO 的方式接收数据
- FIFO 模式：

每个接收通道有 255*32bit 的接收 FIFO；

每个发送通道有 127*32bit 的发送 FIFO；

可设置接收 FIFO 的触发深度，最大为 254，单位 32bit

可设置发送 FIFO 的触发深度，最大为 126，单位 32bit

- 3 种波特率：100K、48K、12.5K
- 有自检功能，奇偶校验软件可设
- 支持标号和 SD 过滤功能
- 支持时间标签功能
- 支持定时发送功能

3.2 运行软件环境描述

Windows 98/2000/xp。

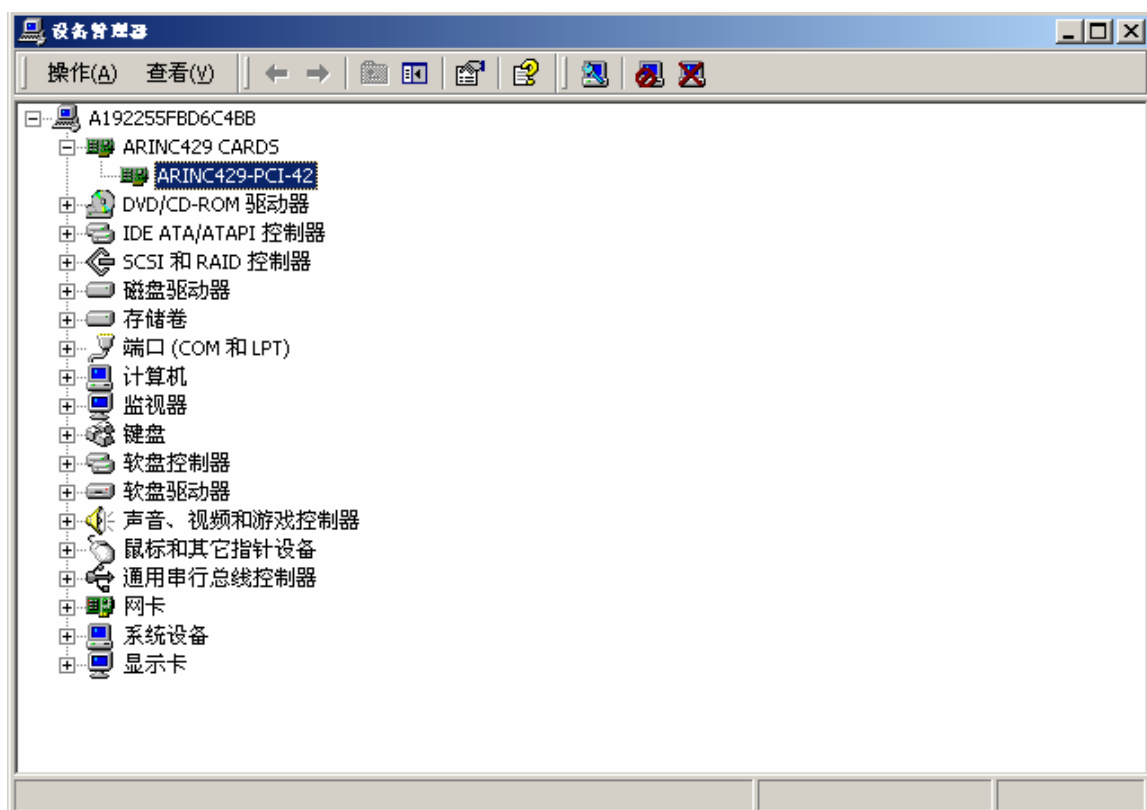
3.3 驱动程序文件说明

- 1、库文件：DLL429D42.dll 和 DLL429D42.lib
- 2、函数库头文件：DLL429D42_lib.h

3.4 安装

1. 将板卡插入PC机CPCI插槽；

2. 系统提示找到新硬件，需要安装它的软件；
3. 点击硬件安装的下一步，若PC机是win2000/98的操作系统，请指向光盘中“驱动安装文件\driver4win2k”目录；若PC机是XP操作系统，请指向光盘中“驱动安装文件\driver4winxp”目录，选择PCI_CARD.inf文件。
4. 点击下一步，直到驱动程序安装成功。
5. 安装完毕，在设备管理器会看到如下的信息（下图中的蓝色区域）。



3.5 接口说明

3.5.1. 结构与定义

3.5.1.1. FIFO 触发深度设置结构

```
typedef struct TriggerDepth
```

```
{
```

```
    BYTE Chan0Depth_I;-----第 1 路接收 FIFO 触发深度寄存器
```



```

    BYTE Chan1Depth_I;-----第 2 路接收 FIFO 触发深度寄存器
    BYTE Chan2Depth_I;-----第 3 路接收 FIFO 触发深度寄存器
    BYTE Chan3Depth_I;-----第 4 路接收 FIFO 触发深度寄存器
    BYTE Chan0Depth_O;-----第 1 路发送 FIFO 触发深度寄存器
    BYTE Chan1Depth_O;-----第 2 路发送 FIFO 触发深度寄存器
} TriggerDepth_STRUCT;

```

注：4 路接收 FIFO 的触发深度的有效位为 8，2 路发送 FIFO 的触发深度的有效位为 7，收 FIFO 里的数据量大于设置的触发深度值，则产生一个触发标志；若发送 FIFO 里的数据量小于设置的触发深度值，则产生一个触发标志。

3.5.1.2. 标号过滤表设置结构

```

typedef struct LabelTable
{
    BYTE LabFilterChan0 [4][256];-----第 1 路标号过滤表设置
    BYTE LabFilterChan1 [4][256];-----第 2 路标号过滤表设置
    BYTE LabFilterChan2 [4][256];-----第 3 路标号过滤表设置
    BYTE LabFilterChan3 [4][256];-----第 4 路标号过滤表设置
} LabelTable_STRUCT;

```

注：LabelFilterChanX：数组的行坐标值代表 S/D 号，列坐标值代表标号；若数组对应元素的值不为零，则 S/D 号对应的标号参与过滤。例：LabelFilterChanX[1][2]=1：表示 S/D 为 1 且 Label 为 2 的 429 数据参与过滤，即被接收

3.5.2. 函数接口描述

3.5.2.1. Card_Open

函数原型：BOOL __stdcall Card_Open (HANDLE *phCardHandle,BYTE CardId);

函数功能：打开 429 板卡，分配板卡资源

参数说明：phCardHandle：板卡的句柄

CardId：板卡编号，取值为 0~255（若 PC 机中同时插 256 块 ARINC429-PCI-42 板卡，板卡的编号按板卡所在的插槽离 CPU 的距离由近到远依次编号 0, 1,..255；若 PC 机中同时只插一块板，板卡编号为 0）

返回值：若找到 429 板卡且板卡资源分配成功，返回值为真；否则为假。

3.5.2.2.Card_Close

函数原型：BOOL __stdcall Card_Close(HANDLE hCardHandle);

函数功能：关闭板卡，释放板卡资源

参数说明：hCardHandle：板卡的句柄

返回值：如果板卡关闭成功，返回值为真；否则为假

3.5.2.3. ResetCard

函数原型：BOOL __stdcall **ResetCard** (HANDLE hCardHandle);

函数功能：板卡复位函数，板卡将清空发送和接收 FIFO；停止接收数据；停止二路发送通道的定时发送；停止时标；清除时间标签、定时发送、标号过滤模式。且板卡处于不通讯状态，此时用户需重新设置波特率。

参数说明：hCardHandle：板卡的句柄

返回值：若板卡复位成功，返回置为真；否则为假

3.5.2.4. EnableBaudrate_48K

函数原型：void __stdcall **EnableBaudrate_48K** (HANDLE hCardHandle, BYTE ChannelNo, BOOL Enable);

函数功能：48K 波特率使能函数

参数说明：hCardHandle：板卡的句柄

ChannelNo：取值为：

0—第 1, 2 路接收通道和第 1 路发送通道 48K 波特

率使能

1—第 3, 4 路接收通道和第 2 路发送通道 48K 波特

率使能

Enable: 48K 波特率使能位。

TRUE: 使能 48K 波特率

FALSE: 不使能 48K 波特率, 波特率由函数 SetConfigureWord
决定

返回值: 空

3.5.2.5. SetConfigureWord

函数原型: BOOL __stdcall **SetConfigureWord** (HANDLE hCardHandle,
BYTE ChannelNo, WORD CFGWord);

函数功能: 设置 429 配置字, 包括波特率和数据位长度及发送奇偶校验设置

参数说明: hCardHandle: 板卡的句柄

ChannelNo: 通道号, 取值范围为 0~1

0---第 1, 2 路接收通道和第 1 路发送通道波特率,
数据位长度及发送奇偶校验设置

1---第 3, 4 路接收通道和第 2 路发送通道波特率,
数据位长度及发送奇偶校验设置

CFGWord: 配置字, 结构见下表:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	RCVSEL	TXSEL	PARCK	Y2	X2	SDENB2	Y1	X1	SDENB1	SLFTST	PAREN	X	X	X	X

PAREN: [发送通道](#)校验使能

1: bit32为校验位 0: bit32为数据位

SLFTST: 自检使能

1: 正常工作状态 0: 自检状态

在自检状态下, 第一路发送通道的输出, 在内部被连接到第一
和第二接收通道。其中, [数据在进入第二接收通道前被反相](#),

以便与第一接收通道区分。自检状态下，发送通道仍然正常输出。

SDEN1: 第一路接收通道S/D码检测使能。

1: 使能第一路接收通道，源/目标 译码器 0: 关闭第一路接收通道，源/目标 译码器

X1, Y1: 第一路接收通道S/D码。

如果SDEN1=1，第一路接收通道将收到ARINC429数据的S/D与X1, Y1进行比较，相同则接纳，不相同则丢弃。其中，X1与串行数据bit9比较，Y1与串行数据bit10比较。

SDEN2: 第二路接收通道S/D码检测使能。

1: 使能第二路接收通道，源/目标 译码器 0: 关闭第二路接收通道，源/目标 译码器

X2, Y2: 第二路接收通道S/D码。

如果SDEN2=1，第二路接收通道将收到ARINC429数据的S/D与X2, Y2进行比较，相同则接纳，不相同则丢弃。其中，X2与串行数据bit9比较，Y2与串行数据bit10比较。

PARCK: 校验检测使能。

1: 将发送通道校验位取反（使用偶校验），以测试校验电路。 0: 使用正常的奇校验。

TXSEL: 第一路发送通道波特率选择。

1: 低速（12.5 K bps） 0: 高速（100 K bps）

RCVSEL: 第一，二路接收通道波特率选择。

1: 低速（12.5 K bps） 0: 高速（100 K bps）

这样，在自检模式下，所有配置字的D5位设置为0，则第1路数据的发送，将连接接收第1、2路数据的接收；第2路数据的发送，将连接接收第3、4路数据的接收；但第2、4、接收通道数据相对于发送通道1、2的数据，将被反相（即取反）。

接收恒定采用奇校验

3.5.2.6. SetLabelFilter

函数原型: void __stdcall **SetLabelFilter** (HANDLE hCardHandle,
LabelTable_STRUCT *Label);

函数功能: 设置标号过滤表, 即某种标号和 S/D 码的 429 数据将被接收

参数说明: hCardHandle: 板卡的句柄

Label: 指向标号过滤表设置结构的指针

返回值: 无

3.5.2.7. StartLabelFilter

函数原型: BOOL __stdcall **StartLabelFilter** (HANDLE hCardHandle,
BYTE ChannelNo, BOOL Enable);

函数功能: 接收通道 429 数据的标号和 S/D 码参与过滤使能函数

参数说明: hCardHandle: 板卡的句柄

ChannelNo: 接收通道号, 取值范围为 0~3, 分别对应接收通道
1~4

Enable: 使能位;

TRUE: 接收通道将只接收标号和 S/D 码符合标号过滤表中
的 429 数据

FALSE: 接收通道将接收所有标号和 S/D 码的 429 数据

返回值: 若标号过滤功能使能成功, 返回值为真; 否则为假

3.5.2.8. AddTimeTag

函数原型: BOOL __stdcall **AddTimeTag** (HANDLE hCardHandle,
BYTE ChannelNo, BOOL Enable);

函数功能: 设置时间标签模式

参数说明: hCardHandle: 板卡的句柄

ChannelNo: 接收通道号, 取值范围为 0~3, 分别对应接收通道 1~4

Enable: 模式使能位;

TRUE: 使能, 接收到的 429 数据将带一个时间标签, 表明

数据接收到的时间

FALSE: 不使能

返回值: 若时标模式设置成功, 返回值为真; 否则为假

3.5.2.9. StartTimeTag

函数原型: void __stdcall **StartTimeTag** (HANDLE hCardHandle,
BOOL Enable, LPSYSTEMTIME CurSysTime);

函数功能: 接收方式下时标开始计数函数, 否则接收的时标为 0

参数说明: hCardHandle: 板卡的句柄

Enable: 使能位; TRUE: 使能, 时标开始计数

FALSE: 不使能, 是标停止计数

CurSysTime: 系统当前时间

```
typedef struct _SYSTEMTIME {  
    WORD wYear;  
    WORD wMonth;  
    WORD wDayOfWeek;  
    WORD wDay;  
    WORD wHour;  
    WORD wMinute;  
    WORD wSecond;  
    WORD wMilliseconds;
```

```
} SYSTEMTIME, *LPSYSTEMTIME;
```

wYear : 年

WMonth : 月; January = 1, February =2, 以此类推

wDayOfWeek: 星期; Sunday = 0, Monday = 1, 以此类推

wDay : 日.

wHour : 小时

wMinute : 分.

wSecond : 秒

wMilliseconds : 毫秒

返回值: 无

3.5.2.10. SetTimerIntervalB

函数原型: BOOL __stdcall **SetTimerIntervalB** (HANDLE hCardHandle,
BYTE ChannelNo, WORD Time);

函数功能: 设置群定时发送时间间隔。群定时发送的时间间隔为两次启动群定时发送数据的起始时间的时间差; 群定时发送的时间应该保证发送完所有的数据, 否则发送的数据会不全。

参数说明: hCardHandle: 板卡的句柄

ChannelNo: 通道号, 取值范围为 0~1, 分别对应发送 1~2 路

Time: 定时时间, 单位为 50 微秒

返回值: 若定时时间设置成功, 返回值为真; 否则为假

3.5.2.11. SetTimerIntervalS

函数原型: BOOL __stdcall **SetTimerIntervalS** (HANDLE hCardHandle,
BYTE ChannelNo, WORD Time);

函数功能: 设置字定时发送时间间隔。字定时发送的时间间隔为发送两个 32 位数据的起始时间的时间差。

参数说明: hCardHandle: 板卡的句柄

ChannelNo: 通道号, 取值范围为 0~1, 分别对应发送 1~2 路

Time: 定时时间, 单位为 50 微秒

返回值: 若定时时间设置成功, 返回值为真; 否则为假

3.5.2.12. StartTimerB

函数原型: BOOL __stdcall **StartTimerB** (HANDLE hCardHandle,
BYTE ChannelNo, BOOL Enable);

函数功能: 启动或停止定时发送

参数说明: hCardHandle: 板卡的句柄

ChannelNo: 通道号, 取值为 0~1, 分别对应发送 1~2 路

Enable: 使能位;

TRUE: 若设置了群定时发送时间, 硬件将按定时时间间隔重复发送 FIFO 里的数据

FALSE: 硬件将停止数据的发送

返回值: 若定时器启动成功, 返回值为真; 否则为假

3.5.2.13. StartTimerS

函数原型: `BOOL __stdcall StartTimerS (HANDLE hCardHandle, BYTE ChannelNo, BOOL Enable);`

函数功能: 启动或停止定时发送

参数说明: hCardHandle: 板卡的句柄

ChannelNo: 通道号, 取值为 0~1, 分别对应发送 1~2 路

Enable: 使能位;

TRUE: 若设置了字定时发送时间, 硬件将按定时时间间隔每次发送 FIFO 里的一个 32 位字数据, 发完将停止

FALSE: 硬件将停止数据的发送

返回值: 若定时器启动成功, 返回值为真; 否则为假

3.5.2.14. SetTriggerDepth

函数原型: `void __stdcall SetTriggerDepth (HANDLE hCardHandle, TriggerDepth_STRUCT *Depth);`

函数功能: 设置 FIFO 触发深度

参数说明: hCardHandle: 板卡的句柄

Depth: FIFO 触发深度结构指针

返回值: 无

3.5.2.15. IsFIFOTriggered_R

函数原型: `BOOL __stdcall IsFIFOTriggered_R (HANDLE hCardHandle,`

BYTE ChannelNo);

函数功能：判断接收 FIFO 是否到达触发深度（每一路接收通道带一个接收 FIFO，用来存放接收到的数据）

参数说明：hCardHandle: 板卡的句柄

ChannelNo: 接收通道号，取值范围为 0~3，分别对应接收通道 1~4

返回值：若对应路 FIFO 到达触发深度，返回值为真；否则为假

3.5.2.16. IsFIFOTriggered_S

函数原型：BOOL __stdcall **IsFIFOTriggered_S** (HANDLE hCardHandle, BYTE ChannelNo);

函数功能：判断发送 FIFO 是否到达触发深度（每一路发送通道带一个发送 FIFO，用来存放待发送的数据）

参数说明：hCardHandle: 板卡的句柄

ChannelNo: 发送通道号，取值范围为 0~1，分别对应发送通道 1~2

返回值：若对应路 FIFO 到达触发深度，返回值为真；否则为假

3.5.2.17. ReadFIFOStatus_R

函数原型：BYTE __stdcall **ReadFIFOStatus_R** (HANDLE hCardHandle, BYTE ChannelNo);

函数功能：读取接收 FIFO 的状态，包括接收 FIFO 的空、满、溢出标志

参数说明：hCardHandle: 板卡的句柄

ChannelNo: 接收通道号，取值范围为 0~3，分别对应接收通道 1~4

返回值：返回值见下表：

二进制值	符号名	意义
0x10	FIFOEmpty	FIFO 空
0x11	FIFOFull	FIFO 满
0x12	FIFONotFull	FIFO 不满，但也不空
0x13	FIFOError	FIFO 溢出
0x14	ErrIndex	错误通道号

3.5.2.18. ReadFIFOStatus_S

函数原型: BYTE __stdcall **ReadFIFOStatus_S** (HANDLE hCardHandle,
BYTE ChannelNo);

函数功能: 读取发送 FIFO 的状态, 包括发送 FIFO 的满、不满标志

参数说明: hCardHandle: 板卡的句柄

ChannelNo: 发送通道号, 取值范围为 0~1, 分别对应发送通道 1~2

返回值: 返回值见下表

二进制值	符号名	意义
0x10	FIFOEmpty	FIFO 空
0x11	FIFOFull	FIFO 满
0x12	FIFONotFull	FIFO 不满, 但也不空
0x14	ErrIndex	错误通道号

3.5.2.19. EnableReadFIFO

函数原型: BOOL __stdcall EnablReadFIFO(HANDLE hCardHandle,
BYTE ChannelNo);

函数功能: 使能读 FIFO 数据

参数说明: hCardHandle: 板卡的句柄

ChannelNo: 待接收数据的通道号, 取值范围为 0~3, 分别对应接收通道 1~4

返回值: 若使能成功则返回真, 否则返回假

3.5.2.20. DisableReadFIFO

函数原型: BOOL __stdcall DisablReadFIFO(HANDLE hCardHandle);

函数功能: 禁止读 FIFO 数据

参数说明: hCardHandle: 板卡的句柄

返回值: 若禁止成功则返回真, 否则返回假

3.5.2.21. ReceiveData

函数原型: DWORD __stdcall **ReceiveData** (HANDLE hCardHandle,
BYTE ChannelNo);

函数功能: 接收数据函数

参数说明: hCardHandle: 板卡的句柄

ChannelNo: 待接收数据的通道号, 取值范围为 0~3, 分别对应接收通道 1~4

返回值: 接收到的数据

3.5.2.22. Enable_Rev

函数原型: void __stdcall Enable_Rev (HANDLE hCardHandle, BOOL Enable);

函数功能: 使能数据接收

参数说明: hCardHandle: 板卡的句柄

在复位板卡的时候禁止接收数据, 所以要在所有的设置完成后调用此函数来使能数据接收

Enable: 使能位;

TRUE: 使能硬件接收数据

FALSE: 硬件将停止接收数据

返回值: 无

3.5.2.23. SendData

函数原型: BOOL __stdcall **SendData** (HANDLE hCardHandle, BYTE ChannelNo, BYTE Len, DWORD *DataAry);

函数功能: 数据发送函数

参数说明: hCardHandle: 板卡的句柄

ChannelNo: 待发送数据的通道号, 取值范围为 0~1, 分别对应发送通道 1~2

Len: 待发送数据的长度, 最大为 127, 单位: 32-bit

DataAry: 存放待发送的数据

返回值: 若数据发送成功, 返回值为真; 否则为假

3.6 操作建议

1. 打开板卡 (Card_Open)
2. 复位板卡 (ResetCard)
3. 设置数据格式

- 1) 设置配置字 (SetConfigureWord)
 - 2) 启动或停止 48K 波特率 (EnableBaudrate_48K)
 4. 设置 FIFO 触发深度 (SetTriggerDepth)
 5. 标号过滤功能的设置
 - 1) 设置待接收数据的标号 (SetLabelFilter)
 - 2) 启动标号过滤功能 (StartLabelFilter)
 6. 时间标签功能设置
 - 1) 设置时间标签模式 (AddTimeTag)
 - 2) 启动时标 (StartTimeTag)
 7. 定时发送功能设置
 - 1) 按群定时发送
 - (a) 设置帧定时发送的时间间隔 (SetTimerIntervalB)
 - (b) 启动定时发送模式 (StartTimerB)
 - 2) 按字定时发送
 - (a) 设置字定时发送的时间间隔 (SetTimerIntervalS)
 - (b) 启动定时发送模式 (StartTimerS)
- 注：二者可同时设置，此时需保证群定时时间>字定时时间*字的个数，否则发送数据不完全。
8. 使能数据接收 (Enable_Rev)
设置完成时要使能数据接收。
 9. 数据的接收
 - 按触发深度的方式读取数据
 - 1) 判断接收 FIFO 是否触发 (IsFIFOTriggered_R)
 - 2) 若 FIFO 触发，使能读 FIFO 数据 (EnableReadFIFO)
 - 3) 读取数据 (ReceiveData)，可连续读取触发深度+1 次
 - 4) 禁止读 FIFO 数据 (DisableReadFIFO)
 - 5) 重复以上 3 步

- 普通方式接收数据

- 1) 判断接收 FIFO 是否为空 (ReadFIFOStatus_R)
- 2) 若不为空且没有溢出, 使能读 FIFO 数据 (EnableReadFIFO)
- 3) 读取数据 (ReceiveData)
- 4) 禁止读 FIFO 数据 (DisableReadFIFO)
- 5) 重复以上 3 步

说明: 若设置了时间标签模式, 则接收数据时, 首先收到的是一个 429 数据 (记为 data429), 接下来的是该 429 数据的时标 (记为 TimeTag)。**data429 接收到的时间=时标开始计数时的系统时间+TimeTag*时标分辨率**, 时标开始计数系统时间可由函数 StartTimeTag 获得。

10. 数据的发送

- ◆ 使用定时发送

- 1) 停止定时器 (StartTimerB 和 StartTimerS)
- 2) 启动定时器 (StartTimerB 和 StartTimerS)
- 3) 读取发送 FIFO 的状态 (ReadFIFOStatus_S)
- 4) 若发送 FIFO 不为满 (不为 FIFOFull), 则发送数据 (SendData)

- ◆ 不使用定时发送

- 1) 读取发送 FIFO 的状态 (ReadFIFOStatus_S)
- 2) 若发送 FIFO 不为满 (不为 FIFOFull), 则发送数据 (SendData)

11. 应用程序退出时, 关闭板卡 (Card_Close)

第四章 应用程序说明

4.1 软件简介

ARINC429-PCI-42 应用程序是本公司针对 PCI 接口 2 发 4 收 429 板卡产品开发的程序，目的是为了使用户很快地对硬件的功能有所了解。

本应用程序通过调用驱动接口程序，可以对硬件板卡提供的所有功能进行设置，还可以进行简单的数据通讯操作。

4.2 运行环境

硬件需求：

可支持 PCI 总线的计算机

至少 128M 内存

显示器分辨率最低可设为 800*600

ARINC429-PCI-42 通讯板

软件需求：

Windows 98/2000/XP 操作系统

ARINC429-PCI-42 硬件驱动程序

4.3 安装

执行光盘中所带有的应用程序安装包 ARINC429-PCI-42.exe，按照安装提示进行程序安装，完成安装后，用户可通过执行任务栏中的“开始”->“程序”->“ARINC429-PCI-42”->“ARINC429-PCI-42”来运行应用程序。

4.4 软件界面

4.4.1 板卡号选择窗口



图 I-1（板卡号选择窗口）

4.4.2 应用程序主窗口

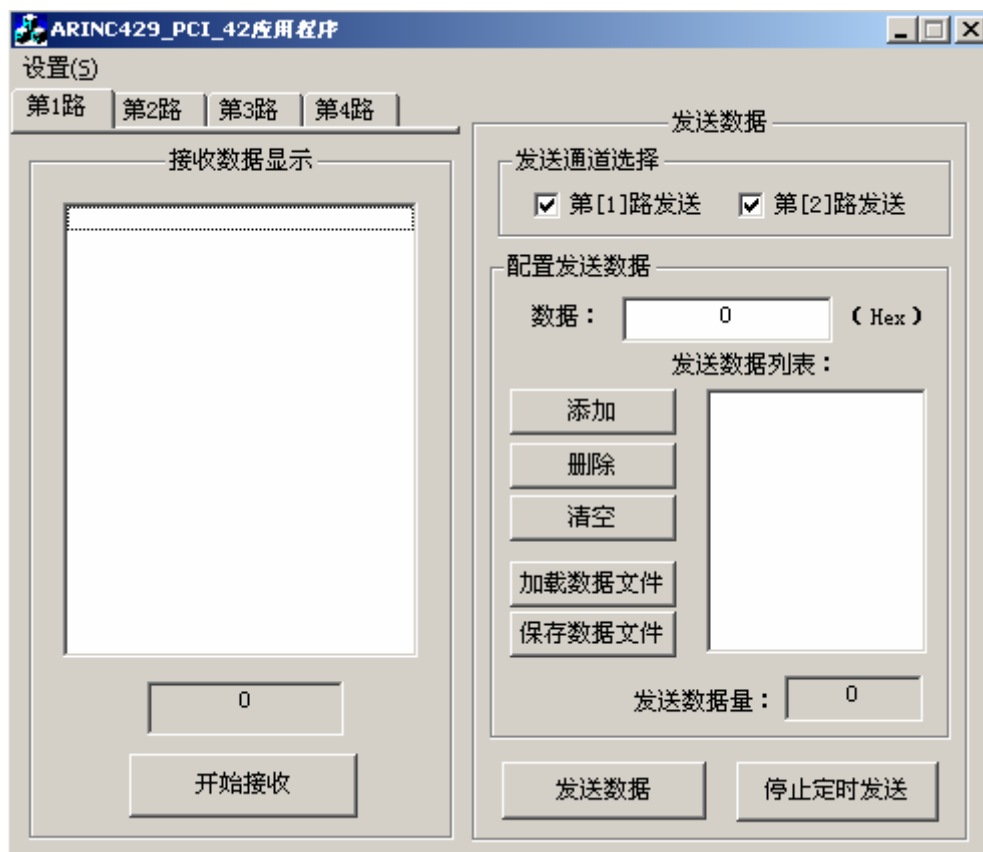


图 I-2（主窗口）

4.4.3 板卡设置窗口



图 I-3（设置窗口）

4.5 菜单功能

设置-----显示板卡设置窗口（图 I-3 所示）

4.6 操作建议

4.6.1 开启应用程序

用户可通过执行任务栏中的“开始”->“程序”->“ARINC429-PCI-42”->“ARINC429-PCI-42”来运行应用程序。

4.6.2 板卡号选择

应用程序启动后，需要做的第一件事就是进行板卡号选择（如图 I-1 所示）。当用户的计算机中同时插有多个同类型的板卡时，“选择板卡号”就用来指定用户当前操作的是哪块板卡，它的输入范围是从 0 到 255 的十进制数。

（板卡的编号按板卡所在的插槽离 CPU 的距离由近到远依次编号 0,1,...255；若 PC 机中同时只插一块板，板卡编号为 0）。

如果指定序号的板卡存在，则直接进入主窗口（如图 I-2）。

4.6.3 设置板卡

在进入主窗口后，程序会自动对板卡进行默认配置，该配置结果允许用户可以进行最基本的数据发送与接收操作，即无触发深度、无定时发送、无时间标签，也没有标号过滤。要进行更多的功能配置，还需要进一步配置板卡。

执行菜单“设置”，程序会显示设置窗口（如图 I-3 所示），这个窗口中包含了硬件所有可被设置的属性。

完成设置后，单击“确定”按钮向板卡提交设置，也可以单击“取消”结束本次操作。

4.6.4 数据通讯操作

应用程序提供了简单的数据通讯操作，如数据的接收与发送。程序会将收到的数据显示在接收数据列表中。用户可以通过窗口界面配置发送数据，还可以从文件中加载发送数据，并通过软件将数据写入发送缓冲区，再由硬件发送出去。

4.6.5 退出应用程序

单击“主窗口”右上角的“X”按钮，便可退出应用程序，退出程序时有提示，确认后即可退出。

4.7 使用说明

4.7.1 硬件设置

在应用程序开启后，用户可以通过执行主窗口菜单中“设置”，显示硬件板卡设置对话框（如图 I-3 所示）。

4.7.1.1 设置配置字

配置字包括：比特率、数据位和校验，如下表所示：

配置字内容	说明		
比特率	100K	48K	12.5K
校验	奇校验	偶校验	无

第 1 路发送与第 1、2 路接收通道采用相同的配置字，第 2 路发送与第 3、4 路接收通道采用相同的配置字。在软件中配置[1]与配置[2]的配置字相同，对应第 1 路发送和第 1、2 路接收的配置字，配置[3]与配置[4]的配置字相同，对应第 2 路发送和第 3、4 路接收的配置字，改变其中一个另外一个也跟着变化。

注：发送校验的用法：发送使用奇校验时，校验位取值取决于除它自己之外的 31 位数据中“1”的个数，若为奇数个，则校验位为“0”，否则为“1”；发送使用偶校验时，校验位取值取决于除它自己之外的 31 位数据中“1”的个数，若为偶数个，则校验位为“0”，否则为“1”。

接收校验的用法：接收恒采用奇校验。接收数据的校验位由接收的 32 位数据决定，如果有奇数个“1”，校验位为“0”，否则为“1”。

举例如下：当发送数据为“12350248”时，其发送和接收的数据如下所示：

发送			接收
校验位	待发送数据	校验后数据	
奇校验	12350248	12350248	12350248
偶校验	12350248	12350348	12350348
无校验	12350248	12350248	12350248

4.7.1.2 设置触发深度

缓冲区名称	触发深度取值范围
发送缓冲区	0 到 126
接收缓冲区	0 到 254

触发深度的设置，会直接影响到 FIFO 缓冲区状态，如果设置值为 0，表示不采用触发深度功能，如果设置值大于 0，则表示采用触发深度。

例如在接收时，需要定时接收定长的数据，可以将数据长度设置好，设置值为待接收长度-1。例如，当需要一次接收至少 6 个数据时，则触发深度可设置为 5，待程序判断缓冲区状态为到达触发深度时，便可将定长的数据一并接收。需要注意的是，如果用户设置了时间标签功能，则时间标签也会被算为数据中的一个长度。

建议用户在使用该深度时最好不要超过整体深度设置值的三分之二。发送缓冲区可不必设置触发深度。

4.7.1.4 设置定时发送

定时发送有两种模式，分别为群定时发送和字定时发送。群定时发送是指将缓冲区内的数据按指定的时间间隔发送出去。字定时发送是指将缓冲区内的数据按指定的时间间隔每次发送一个 32 位数据，发完就停止。本程序提供了对二路发送通道的定时发送功能的设置，其中配置[1]和配置[2]配置第 1 路发送通道，配置[3]和配置[4]配置第 2 路发送通道，二者自动采用一致的设置。

定时时间的基本单位为 50 微秒，当输入值为 20000 时，则表示实际定时发送的时间间隔为 1 秒。具体公式如下：

$$\text{实际时间间隔} = \text{时间间隔输入值} \times \text{单位时间}$$

经过如上公式，我们可以算出实际的时间间隔值：

$$\begin{aligned}\text{实际时间间隔} &= 20000 \times 50 \\ &= 1000000 \text{ (微秒)} = 1 \text{ (秒)}\end{aligned}$$

定时发送值设置为 0 时，则表示不采用定时发送功能。

二者可以同时设置，但需满足群定时发送时间 > 字定时发送时间 * 缓冲区中字的个数，否则数据将发不完全。

4.7.1.5 添加时间标签

时间标签是指在接收到的数据后面增加一个接收到的时间，这个时间就称为时间标签，该标签的作用是用来描述当前接收到的这个数据的到达时间。其单位是 50 微秒，求得实际到达时间的方法与计算定时发送时间间隔相同，但这个时间标签的定时是从设置该功能的时候开始的，它的计时完全由硬件来实现。

4.7.1.6 设置接收标号过滤

标号过滤实际上就是，通知硬件只接收那些用户想收到的数据。该功能的实现主要是依据 429 标准，通过识别数据的标号实现数据过滤功能，标号指的是 ARINC429 数据中的第 0 位到第 7 位表示的一个 8 位二进制数据。

当数据位设置为 32 位时，SD 值也参与过滤，SD 值指 ARINC429 数据中的第 11、12 两位，SD 下拉列表中的数据为 0 到 3，描述二进制数据的两个数据位。

当数据位设置为 25 位时，SD 值则不参与过滤，软件中每一个 SD 值都对应着 256 个标号。设置标号过滤时，与 32 位的有些区别，需要将 4 个 SD 值对应的 4 组标号均设置为相同的，而在 32 位数据通讯时，由于 SD 参与了过滤，则每个 SD 值都可以对应不同的标号设置。

在接收标号过滤设置中，共有 256 个标号可设置，分别是 0 到 255。且每一个 SD 值都有 256 个标号。标号里横向的是低 4 位，纵向的是高 4 位，如果用户要设置一个 SD 为 2 标号为 45H（69D）的数据过滤时，需要先将 SD 选择在 2 上，然后再在相应的标号上选中即可，操作结果如下所示：

接收数据标号过滤

SD: ☒ 允许标号过滤

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
a	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
b	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
c	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
d	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
e	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
f	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

在设置 256 个标号时当用户选中了其中的某一项，就表示对该标号的数据进行过滤，也就是会接收到该标号的数据。

单配置了标号与 SD 是不够的，只有将“允许标号过滤”选中，才可以便板卡具有标号过滤功能。

4.7.1.7 板卡工作方式

方式	说明
自环	在该方式下，板卡实现内部自环功能，主要表现为，板卡可以在不与外界通讯时，便可接收到自己发送的数据，ARINC429-PCI-42 板卡具有 4 路接收与 2 路发送的能力，通过本程序可以从 4 路接收通道接收到发送的数据，但需要注意的是，此时 1、3 接收通道收到的是原数据，2、4 接收通道接收到的数据为原数据的取反值。该工作方式下，从第 1 路发送的数据会由 1、2 路接收收到，第 2 路发送的数据会由 3、4 路接收收到。
正常	在该方式下，板卡可保证正常通讯。

4.7.1.8 配置的保存与加载

在每次应用程序打开后，程序会自动按照默认值对板卡进行配置，实现基本的数据通讯能力，但往往用户都需要将板卡的功能配置成自己想要的功能，这样就需要经常对板卡进行配置操作，在设置窗口中的“加载配置”与

“保存配置”可以提供些方便。当用户进入到配置窗口后，首次对板卡通讯进行自定义配置，配置完成后，可以执行“保存配置”，这时程序会将程序中当前的配置保存到配置文件中，以便下次重新启动应用程序后，用户可以通过执行“加载配置”操作重新获取之前保存的配置。

4.7.2 数据的接收与发送

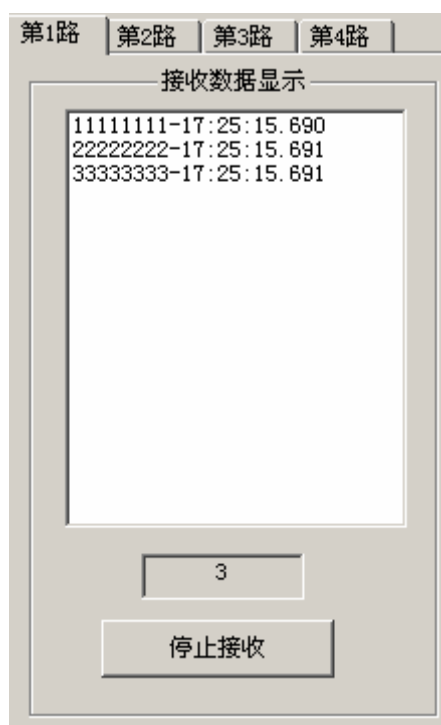
该板卡有 4 路接收和 2 路发送通道，主窗口用于实现数据通讯操作（如图 I-2 所示）。

4.7.2.1 接收数据

需要进行数据接收时，执行“开始接收”就可以了，执行它后，该按钮会自动变为“停止接收”按钮，终止接收操作时，只需再次点击一下该按钮就可以了，当接收缓冲区溢出时，应用程序停止接收，此时需复位板卡（点击“设置”菜单或重新启动应用程序）才能进行下一次接收。

程序会将接收到的数据分别显示在列表窗口中，接收到的数据量会显示在列表窗口下方的文本窗口中。查看接收数据可以通过选择接收通道选项卡实现。

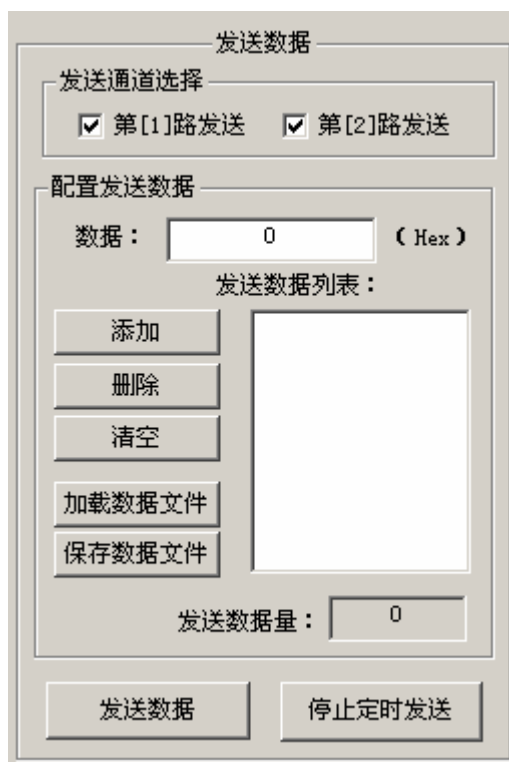
接收数据时，如果添加了时标功能，在接收列表中的数据显示会有所变化。在每一个数据后面都会加上“时:分:秒.毫秒”的显示，用来表示当前数据接收到的时间。如下图所示：



4.7.2.2 发送数据

发送数据时可以通过执行“发送数据”进行数据发送，如果配置了定时发送功能，则可通过执行“停止定时发送”停止定时发送，停止定时发送包括停止群定时发送和停止字定时发送，停止群定时发送时停止定时发送缓冲区内的数据，若停止字定时发送，则以普通模式发送缓冲区中的数据。

程序总是会将发送列表中的数据发送出去，所示要保证发送数据列表不为空。添加发送数据到列表有两个方法，一个就是通过“添加”、“删除”、“清空”操作将数据文本窗口中的数据添加到列表，输入数据时要以十六进制的输入方式进行输入，列表中显示的数据也都是十六进制的。另一个方法是通过“加载数据文件”和“保存数据文件”进行发送列表的配置，“加载数据文件”会指定一个数据文件，并将里面的数据以续加的方式添加到列表中，不会将列表中原有的数据清空。“保存数据文件”会将列表中的数据保存到文件中。两种配置发送数据列表的方法可以混合使用。



注：“删除”操作会将发送列表中被选中的数据删除掉，当列表中没有数据被选中时，将会从列表中的最后一个数据开始删除。

4.8 文件格式

当完成安装后，在安装目录中应有如下文件：

APP429.exe 应用程序

DLL429D42.dll 驱动程序接口

Fym429cfg.ini 配置文件，用于保存用户对板卡的配置

data.sdt 数据文件，可由用户自定义修改，里面存放发送数据。

注：除数据文件可以由用户通过软件修改，其它三个文件不可以由用户自行修改。

4.9 编程举例

为方便开发个性化的应用程序，我公司提供了应用程序例程，该例程可从交付客户的光盘中获得。例程采用的是 VC 6.0 工具进行编写，所用的编程语言是 C 语言。这里我们仅是提供了一个简单的例子，但我们的动态链接库是可以被 Visual C++、Visual Basic、Delphi、LabView 以及 CVI 等开发工具调用的。

附录 A：429 数据格式转换

图上面的为 429 总线上的数据格式，而下面的为计算机系统软件的数据格式。即发送 FIFO 发送的数据与 429 串行数据总线上的数据进行下面的格式转换。其中 Word1 是系统数据低 16 位，而 Word2 是系统数据高 16 位。

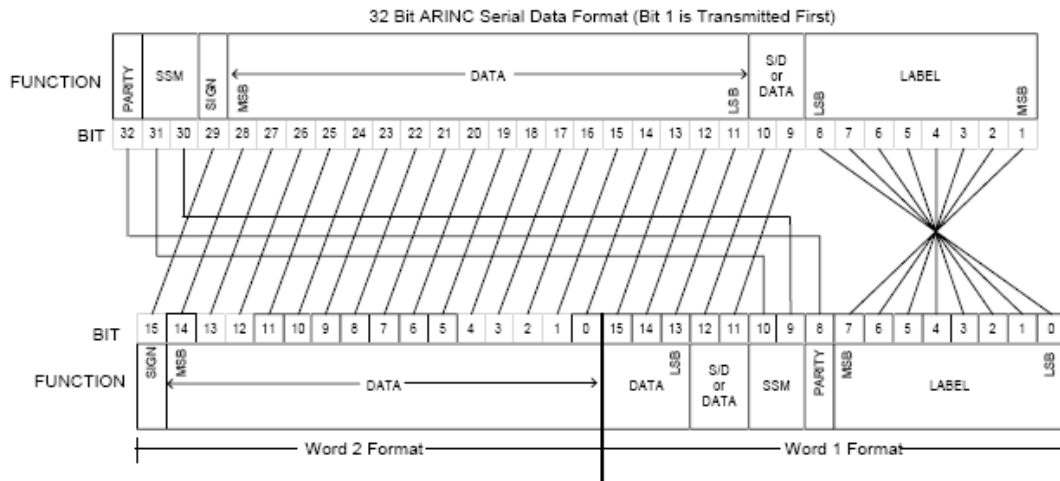


Figure 2: Mapping of Serial Data to/from Word 1 and Word 2 in 32 bit format.

例如：系统软件写数据 12345678 到发送 FIFO，奇偶校验设置位奇。则系统数据为

Word1=0X5678， Word2=0X1234。

429 串行数据格式为： 0X62468A1E