

CHR32XXX-ARINC429

PXI/CPCI/PCI 总线 ARINC429 板



用户手册

声 明

本文档中介绍的产品（包括硬件、软件和文档本身）版权归西安凯锐测控科技有限公司所有，保留所有权利。未经西安凯锐测控科技有限公司书面授权，任何人不得以任何方式复制本文档的任何部分。

对于本文档所有明示或暗示的条款、陈述和保证，包括任何针对特定用途的适用性或不侵害知识产权的暗示保证，均不提供任何担保，除非此类免责声明的范围在法律上视为无效。西安凯锐测控科技有限公司不对任何与性能或使用本文档相关的伴随或后果性损害负责。本文档所含信息如有更改，恕不另行通知。

CHR32XXX 用户手册

文档版本：V1.0-3

发布日期：2019-6-21

目 录

第一章 概述	1
1.1 关于本手册	2
1.2 产品描述	2
1.2.1 特性	2
1.2.2 详细描述	3
1.2.3 一般规格	3
1.3 产品安装	4
1.3.1 安装之前的准备	4
1.3.2 硬件安装	4
1.3.3 驱动安装	5
1.3.4 演示应用软件安装	6
第二章 硬件说明	7
2.1 功能结构图	8
2.2 连接器和信号定义	8
第三章 驱动程序编程接口	13
3.1 动态库 DLL	14
3.1.1 适用编程工具	14
3.1.2 需要引用的文件	14
3.1.3 引用结构说明	14
3.1.3.1 板卡设备参数信息结构	14
3.1.3.2 板卡设备总线信息结构	15
3.1.3.3 429 配置字结构	15
3.1.3.4 定时发送结构	16
3.1.3.5 S/D 标号过滤结构	16
3.2 驱动程序函数功能	17
3.2.1 主要功能	17
3.2.2 功能设计	17
3.3 驱动软件接口函数说明	17
3.3.1 CHR429_Open	17
3.3.2 CHR429_Close	18
3.3.3 CHR429_Reset	18
3.3.4 CHR429_GetDevParInfo	18
3.3.5 CHR429_GetDevBusInfo	18
3.3.6 CHR429_StartTimeTag	18
3.3.7 CHR429_Ch_Reset	19
3.3.8 CHR429_Ch_SetWorkMode	19
3.3.9 CHR429_Ch_Init	19
3.3.10 CHR429_Ch_ResetFIFO	20
3.3.11 CHR429_Ch_WordConvertEnable	20
3.3.12 CHR429_ChTx_TimerCFG	21
3.3.13 CHR429_ChTx_Data	21
3.3.14 CHR429_ChTx_RdFIFOCnt	22

3.3.15 CHR429_ChTx_RdFIFOSTR.....	22
3.3.16 CHR429_ChTx_Start.....	22
3.3.17 CHR429_ChRx_AddTimeTag.....	23
3.3.18 CHR429_ChRx_LabelFilter.....	23
3.3.19 CHR429_ChRx_SetInt.....	24
3.3.20 CHR429_ChRx_GetInt.....	24
3.3.21 CHR429_ChRx_CreateIntEvt.....	25
3.3.22 CHR429_ChRx_WaitIntEvt.....	25
3.3.23 CHR429_ChRx_CloseIntEvt.....	26
3.3.24 CHR429_ChRx_Start.....	26
3.3.25 CHR429_ChRx_RdFIFOCnt.....	26
3.3.26 CHR429_ChRx_RdFIFOStatus.....	27
3.3.27 CHR429_ChRx_Data.....	27
3.4 错误码列表.....	28
3.5 驱动接口函数调用步骤.....	28
3.5.1 板卡操作.....	28
3.5.2 通道公共参数配置.....	28
3.5.3 数据的接收.....	28
3.5.4 数据的发送.....	29
3.5.5 关闭板卡.....	29
第四章 功能演示软件.....	30
4.1 使用环境.....	31
4.1.1 系统要求.....	31
4.1.2 操作系统.....	31
4.2 主要功能.....	31
4.3 图形化接口设计.....	31
4.3.1 菜单.....	31
4.3.1.1 板卡配置（包括复位，字格式转换，自检工作）.....	32
4.3.1.2 通道配置（接收和发送）.....	32
4.3.2 界面.....	34
4.4 功能描述.....	35
4.4.1 接收功能设计.....	35
4.4.2 发送功能设计.....	36
附录 (ARINC429 通讯中奇偶校验位).....	37
A 原理.....	37
B 举例.....	39

第一章 概述

1.1 关于本手册

本手册适用于下列产品型号：

- **CHR321XX** 32 位/33MHz PXI 总线 非隔离
- **CHR321XX-G** 32 位/33MHz PXI 总线 隔离型
包含 2 收 2 发 ARINC429[CHR32102]、4 收 4 发 ARINC429[CHR32104]、8 收 8 发 ARINC429[CHR32108]、16 收 16 发 ARINC429[CHR32116]的配置。

- **CHR322XX** 32 位/33MHz CPCI 总线 非隔离
- **CHR322XX-G** 32 位/33MHz CPCI 总线 隔离型
包含 2 收 2 发 ARINC429[CHR32202]、4 收 4 发 ARINC429[CHR32204]、8 收 8 发 ARINC429[CHR32208]、16 收 16 发 ARINC429[CHR32216]的配置。

- **CHR323XX** 32 位/33MHz PCI 总线 非隔离
- **CHR323XX-G** 32 位/33MHz PCI 总线 隔离型
包含 2 收 2 发 ARINC429[CHR32302]、4 收 4 发 ARINC429[CHR32304]、8 收 8 发 ARINC429[CHR32308]、16 收 16 发 ARINC429[CHR32316]的配置。

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

CHR32XXX 是一款包含多种通道配置的 ARINC429 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

- 32 位/33MHz CPCI 总线
- 最大 16 路发送通道，每路带 511×32 位 FIFO；
- 最大 16 路接收通道，每路带 (1M-1)×32 位 FIFO；
- 发送和接收 FIFO 复位；
- 具有可设的消息间隔、字间隔和发送次数设置；
- 添加时间标签功能；
- 接收标号过滤；
- 接收 FIFO 触发深度可设功能；
- 波特率 10K~150Kbps，接收及发送可分别设置；
- 标准 ARINC429 数据格式，接收及发送可分别设置；
- 8 通道为一组，组间隔离（CHR32XXX-G 产品支持）；
- 隔离电压 1500VDC（CHR32XXX-G 产品支持）

1.2.2 详细描述

- 接收和发送通道的配置
 - 2 路接收、2 路发送配置
 - 4 路接收、4 路发送配置
 - 8 路接收、8 路发送配置
 - 16 路接收、16 路发送配置
- 发送通道
 - ▶ 每路发送通道 FIFO 大小为 511×32 位。
 - ▶ 发送 FIFO 可设置复位
 - ▶ 可设置消息间隔，字间隔和发送帧的预定数量。
 - ▶ 发送波特率 150Kbps、100Kbps、50Kbps、48Kbps、12.5Kbps、10Kbps 可设置。
 - ▶ 标准 ARINC429 字格式转换与否可选择。
- 接收通道
 - ▶ 每路接收 FIFO 大小 $(1M-1) \times 32$ 位。
 - ▶ 接收使能和接收 FIFO 复位功能。
 - ▶ 接收 FIFO 触发深度可设功能；
 - ▶ 接收标号过滤功能。
 - ▶ 接收添加时标功能。
 - ▶ 接收波特率 150Kbps、100Kbps、50Kbps、48Kbps、12.5Kbps、10Kbps 可设置。
 - ▶ 标准 ARINC429 字格式转换与否可选择。
- 8 通道为一组，组间隔离（CHR32XXX-G 产品支持）
- 隔离电压：1500VDC（CHR32XXX-G 产品支持）

1.2.3 一般规格

- 物理尺寸：标准 PXI/CPCI 3U 卡尺寸， $160\text{mm} \times 100\text{mm} \times 20\text{mm}$ ，公差小于 0.2mm
标准 PCI 尺寸 $175\text{mm} \times 106\text{mm}$ ，公差小于 0.2mm
- 连接器：SCSI68 母座
- 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

- CHR32XXX 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 CHR32XXX 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。

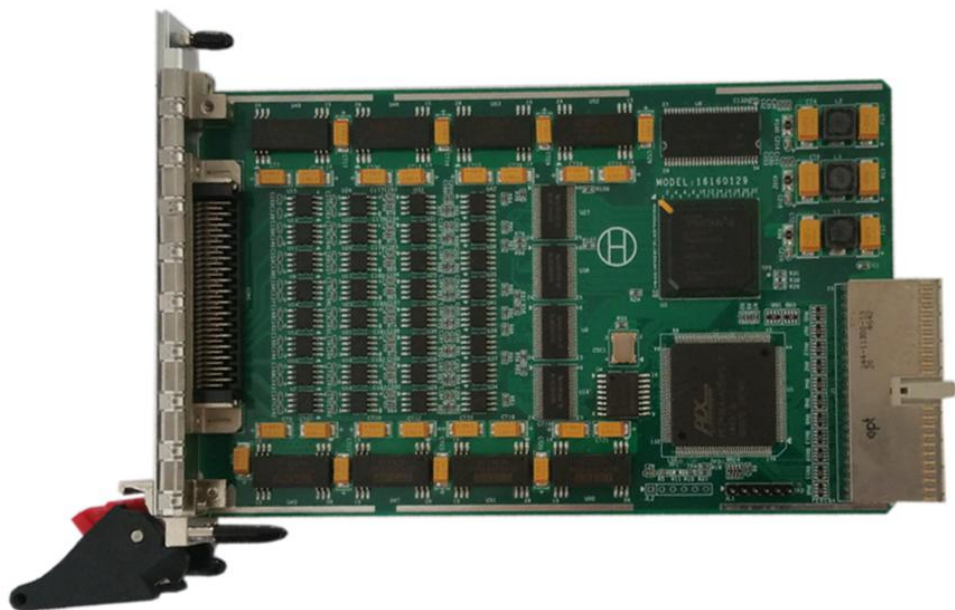


图 1-1a CHR322XX 产品图片示例

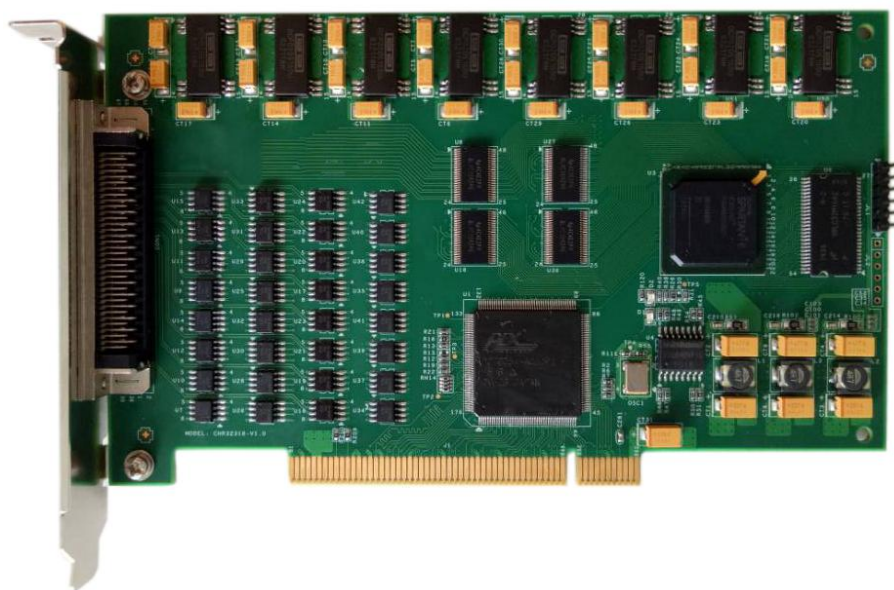


图 1-1b CHR323XX 产品图片示例

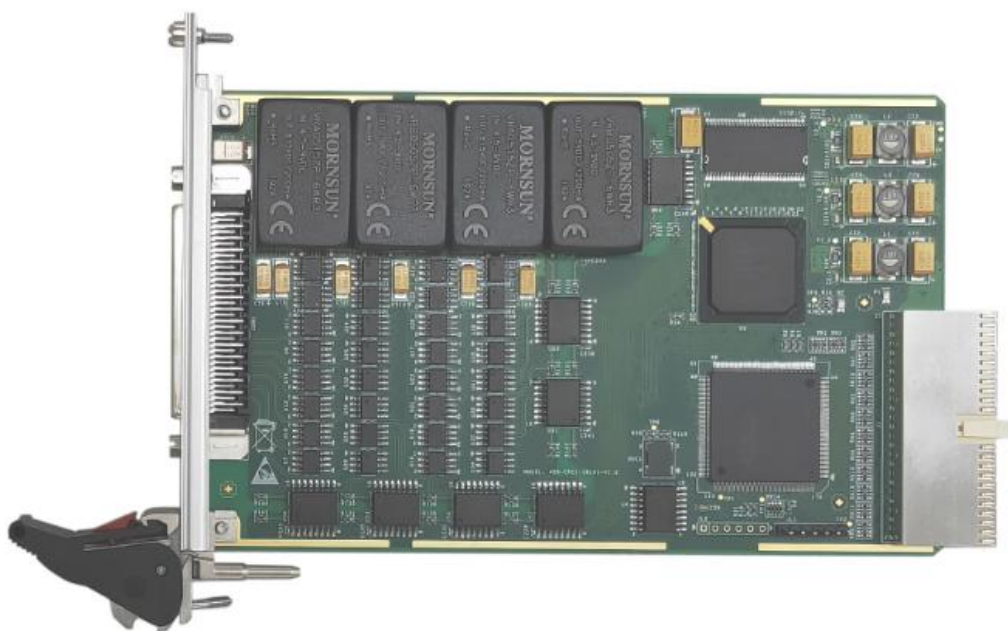


图 1-1c CHR322XX-G 产品图片示例

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 [2.3](#) 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

在产品配套光盘的“驱动安装”目录中，可以找到 **CHR32XXX** 板卡的驱动。请按如下步骤安装产品的驱动：

- 1、将板卡插入工控机相应的插槽
- 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，确定驱动程序所在的目录
- 4、 点击下一步，直到驱动程序安装成功

在完成 CHR32XXX 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 CHR32XXX 板卡设备项，如图 1-2 所示。CHR32XXX(2 表示通道数目)



图 1-2 CHR32XXX(2)板卡在设备管理器中的设备项

1.3.4 演示应用软件安装

在安装完 CHR32XXX 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 CHR32XXX 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

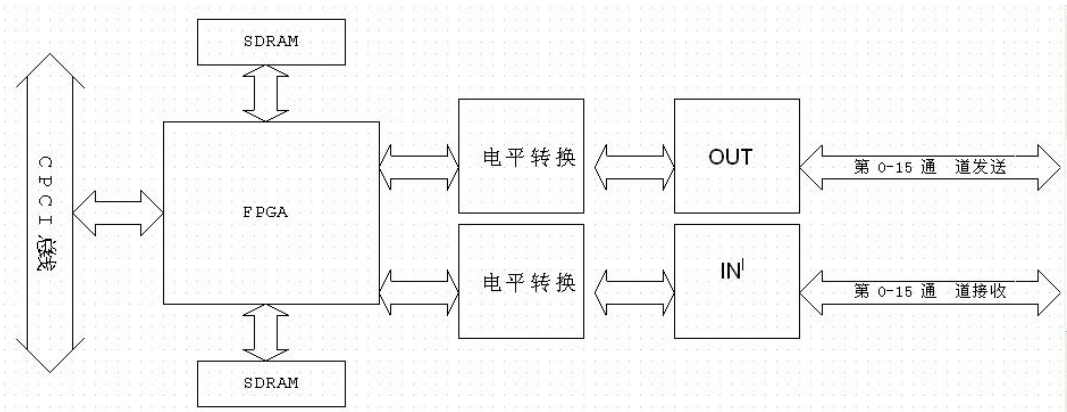


图 2-1a CHR32XXX 功能结构图

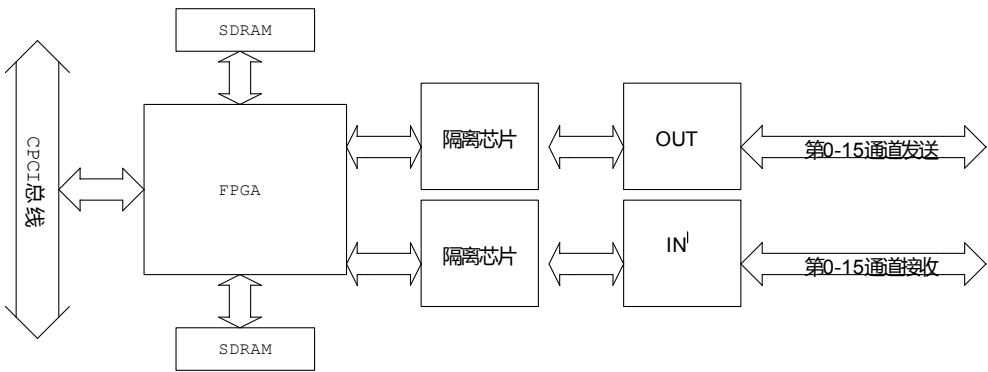


图 2-1b CHR32XXX-G 功能结构图

2.2 连接器和信号定义

各通道类型板卡点号定义

本板卡采用 SCSI68 标准接头。点号定义如下表（空白部分不连接）：

型号：CHR32X02

管脚	名称	说明	管脚	名称	说明
1	429_INB0	第 0 路 429 输入负	35	429_OUTB0	第 0 路 429 输出负
2	429_INA0	第 0 路 429 输入正	36	429_OUTA0	第 0 路 429 输出正
3	429_INB1	第 1 路 429 输入负	37	429_OUTB1	第 1 路 429 输出负
4	429_INA1	第 1 路 429 输入正	38	429_OUTA1	第 1 路 429 输出正
33	GND	参考地	67	GND	参考地
34	GND	参考地	68	GND	参考地

型号：CHR32X02-G

管脚	名称	说明	管脚	名称	说明
1	429_INB0	第 0 路 429 输入负	35	429_OUTB0	第 0 路 429 输出负
2	429_INA0	第 0 路 429 输入正	36	429_OUTA0	第 0 路 429 输出正
3	429_INB1	第 1 路 429 输入负	37	429_OUTB1	第 1 路 429 输出负
4	429_INA1	第 1 路 429 输入正	38	429_OUTA1	第 1 路 429 输出正

第二章 硬件说明

CHR32XXX 用户手册

33	GND_I1	隔离参考地	67		
34	GND_I1	隔离参考地	68		

型号：CHR32X04

管脚	名称	说明	管脚	名称	说明
1	429_INB0	第 0 路 429 输入负	35	429_OUTB0	第 0 路 429 输出负
2	429_INA0	第 0 路 429 输入正	36	429_OUTA0	第 0 路 429 输出正
3	429_INB1	第 1 路 429 输入负	37	429_OUTB1	第 1 路 429 输出负
4	429_INA1	第 1 路 429 输入正	38	429_OUTA1	第 1 路 429 输出正
5	429_INB2	第 2 路 429 输入负	39	429_OUTB2	第 2 路 429 输出负
6	429_INA2	第 2 路 429 输入正	40	429_OUTA2	第 2 路 429 输出正
7	429_INB3	第 3 路 429 输入负	41	429_OUTB3	第 3 路 429 输出负
8	429_INA3	第 3 路 429 输入正	42	429_OUTA3	第 3 路 429 输出正
33	GND	参考地	67	GND	参考地
34	GND	参考地	68	GND	参考地

型号：CHR32X04-G

管脚	名称	说明	管脚	名称	说明
1	429_INB0	第 0 路 429 输入负	35	429_OUTB0	第 0 路 429 输出负
2	429_INA0	第 0 路 429 输入正	36	429_OUTA0	第 0 路 429 输出正
3	429_INB1	第 1 路 429 输入负	37	429_OUTB1	第 1 路 429 输出负
4	429_INA1	第 1 路 429 输入正	38	429_OUTA1	第 1 路 429 输出正
5	429_INB2	第 2 路 429 输入负	39	429_OUTB2	第 2 路 429 输出负
6	429_INA2	第 2 路 429 输入正	40	429_OUTA2	第 2 路 429 输出正
7	429_INB3	第 3 路 429 输入负	41	429_OUTB3	第 3 路 429 输出负
8	429_INA3	第 3 路 429 输入正	42	429_OUTA3	第 3 路 429 输出正
33	GND_I1	隔离参考地	67		
34	GND_I1	隔离参考地	68		

型号：CHR32X08

管脚	名称	说明	管脚	名称	说明
1	429_INB0	第 0 路 429 输入负	35	429_OUTB0	第 0 路 429 输出负
2	429_INA0	第 0 路 429 输入正	36	429_OUTA0	第 0 路 429 输出正
3	429_INB1	第 1 路 429 输入负	37	429_OUTB1	第 1 路 429 输出负
4	429_INA1	第 1 路 429 输入正	38	429_OUTA1	第 1 路 429 输出正
5	429_INB2	第 2 路 429 输入负	39	429_OUTB2	第 2 路 429 输出负
6	429_INA2	第 2 路 429 输入正	40	429_OUTA2	第 2 路 429 输出正
7	429_INB3	第 3 路 429 输入负	41	429_OUTB3	第 3 路 429 输出负
8	429_INA3	第 3 路 429 输入正	42	429_OUTA3	第 3 路 429 输出正
9	429_INB4	第 4 路 429 输入负	43	429_OUTB4	第 4 路 429 输出负
10	429_INA4	第 4 路 429 输入正	44	429_OUTA4	第 4 路 429 输出正

11	429_INB5	第 5 路 429 输入负	45	429_OUTB5	第 5 路 429 输出负
12	429_INA5	第 5 路 429 输入正	46	429_OUTA5	第 5 路 429 输出正
13	429_INB6	第 6 路 429 输入负	47	429_OUTB6	第 6 路 429 输出负
14	429_INA6	第 6 路 429 输入正	48	429_OUTA6	第 6 路 429 输出正
15	429_INB7	第 7 路 429 输入负	49	429_OUTB7	第 7 路 429 输出负
16	429_INA7	第 7 路 429 输入正	50	429_OUTA7	第 7 路 429 输出正
33	GND	参考地	67	GND	参考地
34	GND	参考地	68	GND	参考地

型号：CHR32X08-G

管脚	名称	说明	管脚	名称	说明
1	429_INB0	第 0 路 429 输入负	35	429_OUTB0	第 0 路 429 输出负
2	429_INA0	第 0 路 429 输入正	36	429_OUTA0	第 0 路 429 输出正
3	429_INB1	第 1 路 429 输入负	37	429_OUTB1	第 1 路 429 输出负
4	429_INA1	第 1 路 429 输入正	38	429_OUTA1	第 1 路 429 输出正
5	429_INB2	第 2 路 429 输入负	39	429_OUTB2	第 2 路 429 输出负
6	429_INA2	第 2 路 429 输入正	40	429_OUTA2	第 2 路 429 输出正
7	429_INB3	第 3 路 429 输入负	41	429_OUTB3	第 3 路 429 输出负
8	429_INA3	第 3 路 429 输入正	42	429_OUTA3	第 3 路 429 输出正
9	429_INB4	第 4 路 429 输入负	43	429_OUTB4	第 4 路 429 输出负
10	429_INA4	第 4 路 429 输入正	44	429_OUTA4	第 4 路 429 输出正
11	429_INB5	第 5 路 429 输入负	45	429_OUTB5	第 5 路 429 输出负
12	429_INA5	第 5 路 429 输入正	46	429_OUTA5	第 5 路 429 输出正
13	429_INB6	第 6 路 429 输入负	47	429_OUTB6	第 6 路 429 输出负
14	429_INA6	第 6 路 429 输入正	48	429_OUTA6	第 6 路 429 输出正
15	429_INB7	第 7 路 429 输入负	49	429_OUTB7	第 7 路 429 输出负
16	429_INA7	第 7 路 429 输入正	50	429_OUTA7	第 7 路 429 输出正
33	GND_I1	隔离参考地	67		
34	GND_I1	隔离参考地	68		

型号：CHR32X16

管脚	名称	说明	管脚	名称	说明
1	429_INB0	第 0 路 429 输入负	35	429_OUTB0	第 0 路 429 输出负
2	429_INA0	第 0 路 429 输入正	36	429_OUTA0	第 0 路 429 输出正
3	429_INB1	第 1 路 429 输入负	37	429_OUTB1	第 1 路 429 输出负
4	429_INA1	第 1 路 429 输入正	38	429_OUTA1	第 1 路 429 输出正
5	429_INB2	第 2 路 429 输入负	39	429_OUTB2	第 2 路 429 输出负
6	429_INA2	第 2 路 429 输入正	40	429_OUTA2	第 2 路 429 输出正
7	429_INB3	第 3 路 429 输入负	41	429_OUTB3	第 3 路 429 输出负
8	429_INA3	第 3 路 429 输入正	42	429_OUTA3	第 3 路 429 输出正
9	429_INB4	第 4 路 429 输入负	43	429_OUTB4	第 4 路 429 输出负
10	429_INA4	第 4 路 429 输入正	44	429_OUTA4	第 4 路 429 输出正

11	429_INB5	第 5 路 429 输入负	45	429_OUTB5	第 5 路 429 输出负
12	429_INA5	第 5 路 429 输入正	46	429_OUTA5	第 5 路 429 输出正
13	429_INB6	第 6 路 429 输入负	47	429_OUTB6	第 6 路 429 输出负
14	429_INA6	第 6 路 429 输入正	48	429_OUTA6	第 6 路 429 输出正
15	429_INB7	第 7 路 429 输入负	49	429_OUTB7	第 7 路 429 输出负
16	429_INA7	第 7 路 429 输入正	50	429_OUTA7	第 7 路 429 输出正
17	429_INB8	第 8 路 429 输入负	51	429_OUTB8	第 8 路 429 输出负
18	429_INA8	第 8 路 429 输入正	52	429_OUTA8	第 8 路 429 输出正
19	429_INB9	第 9 路 429 输入负	53	429_OUTB9	第 9 路 429 输出负
20	429_INA9	第 9 路 429 输入正	54	429_OUTA9	第 9 路 429 输出正
21	429_INB10	第 10 路 429 输入负	55	429_OUTB10	第 10 路 429 输出负
22	429_INA10	第 10 路 429 输入正	56	429_OUTA10	第 10 路 429 输出正
23	429_INB11	第 11 路 429 输入负	57	429_OUTB11	第 11 路 429 输出负
24	429_INA11	第 11 路 429 输入正	58	429_OUTA11	第 11 路 429 输出正
25	429_INB12	第 12 路 429 输入负	59	429_OUTB12	第 12 路 429 输出负
26	429_INA12	第 12 路 429 输入正	60	429_OUTA12	第 12 路 429 输出正
27	429_INB13	第 13 路 429 输入负	61	429_OUTB13	第 13 路 429 输出负
28	429_INA13	第 13 路 429 输入正	62	429_OUTA13	第 13 路 429 输出正
29	429_INB14	第 14 路 429 输入负	63	429_OUTB14	第 14 路 429 输出负
30	429_INA14	第 14 路 429 输入正	64	429_OUTA14	第 14 路 429 输出正
31	429_INB15	第 15 路 429 输入负	65	429_OUTB15	第 15 路 429 输出负
32	429_INA15	第 15 路 429 输入正	66	429_OUTA15	第 15 路 429 输出正
33	GND	参考地	67	GND	参考地
34	GND	参考地	68	GND	参考地

型号: CHR32X16-G

管脚	名称	说明	管脚	名称	说明
1	429_INB0	第 0 路 429 输入负	35	429_OUTB0	第 0 路 429 输出负
2	429_INA0	第 0 路 429 输入正	36	429_OUTA0	第 0 路 429 输出正
3	429_INB1	第 1 路 429 输入负	37	429_OUTB1	第 1 路 429 输出负
4	429_INA1	第 1 路 429 输入正	38	429_OUTA1	第 1 路 429 输出正
5	429_INB2	第 2 路 429 输入负	39	429_OUTB2	第 2 路 429 输出负
6	429_INA2	第 2 路 429 输入正	40	429_OUTA2	第 2 路 429 输出正
7	429_INB3	第 3 路 429 输入负	41	429_OUTB3	第 3 路 429 输出负
8	429_INA3	第 3 路 429 输入正	42	429_OUTA3	第 3 路 429 输出正
9	429_INB4	第 4 路 429 输入负	43	429_OUTB4	第 4 路 429 输出负
10	429_INA4	第 4 路 429 输入正	44	429_OUTA4	第 4 路 429 输出正
11	429_INB5	第 5 路 429 输入负	45	429_OUTB5	第 5 路 429 输出负
12	429_INA5	第 5 路 429 输入正	46	429_OUTA5	第 5 路 429 输出正
13	429_INB6	第 6 路 429 输入负	47	429_OUTB6	第 6 路 429 输出负
14	429_INA6	第 6 路 429 输入正	48	429_OUTA6	第 6 路 429 输出正
15	429_INB7	第 7 路 429 输入负	49	429_OUTB7	第 7 路 429 输出负
16	429_INA7	第 7 路 429 输入正	50	429_OUTA7	第 7 路 429 输出正

17	429_INB8	第 8 路 429 输入负	51	429_OUTB8	第 8 路 429 输出负
18	429_INA8	第 8 路 429 输入正	52	429_OUTA8	第 8 路 429 输出正
19	429_INB9	第 9 路 429 输入负	53	429_OUTB9	第 9 路 429 输出负
20	429_INA9	第 9 路 429 输入正	54	429_OUTA9	第 9 路 429 输出正
21	429_INB10	第 10 路 429 输入负	55	429_OUTB10	第 10 路 429 输出负
22	429_INA10	第 10 路 429 输入正	56	429_OUTA10	第 10 路 429 输出正
23	429_INB11	第 11 路 429 输入负	57	429_OUTB11	第 11 路 429 输出负
24	429_INA11	第 11 路 429 输入正	58	429_OUTA11	第 11 路 429 输出正
25	429_INB12	第 12 路 429 输入负	59	429_OUTB12	第 12 路 429 输出负
26	429_INA12	第 12 路 429 输入正	60	429_OUTA12	第 12 路 429 输出正
27	429_INB13	第 13 路 429 输入负	61	429_OUTB13	第 13 路 429 输出负
28	429_INA13	第 13 路 429 输入正	62	429_OUTA13	第 13 路 429 输出正
29	429_INB14	第 14 路 429 输入负	63	429_OUTB14	第 14 路 429 输出负
30	429_INA14	第 14 路 429 输入正	64	429_OUTA14	第 14 路 429 输出正
31	429_INB15	第 15 路 429 输入负	65	429_OUTB15	第 15 路 429 输出负
32	429_INA15	第 15 路 429 输入正	66	429_OUTA15	第 15 路 429 输出正
33	GND_I1	0-7 通道隔离参考地	67	GND_I2	8-15 通道隔离参考地
34	GND_I1	0-7 通道隔离参考地	68	GND_I2	8-15 通道隔离参考地

第三章 驱动程序编程接口

本章主要讲述了如何使用 CHR32XXX 板卡的驱动程序接口,为用户编程提供参考。CHR32XXX 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

CHR32XXX 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 CHR32XXX 板卡配套光盘获取。

3.1.1 适用编程工具

运行环境: Windows 98/2000/2003/xp/Win7 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

当您进行程序开发时,需要引用下列文件:

库文件:

CHR32XXX.dll 和 CHR429.lib

函数库头文件:

CHR429_lib.h

3.1.3 引用结构说明

3.1.3.1 板卡设备参数信息结构

```
typedef struct _CHR_DEVPARST_  
{  
    DWORD dwCardType;  
    DWORD dwhwVersion;  
    DWORD dwdvrVersion;  
    DWORD dwlibVersion;  
    DWORD dwBoardID;  
    DWORD dwSN;  
    DWORD dwChMax;  
} CHR_DEVPARST, *pCHR_DEVPARST;
```

dwCardType: 板卡类型

如 0x2216 表示 16 收 16 发 429 板卡 CPCI 总线

如 0x2116 表示 16 收 16 发 429 板卡 PXI 总线类

如 0x2204 表示 4 收 4 发 429 板卡 CPCI 总线类

如 0x2304 表示 4 收 4 发 429 板卡 PCI 总线类

dwhwVersion: 硬件版本号

dwdvrVersion: 驱动版本号

dwlibVersion: 库版本号

注：高 16 位为主版本号，低 16 位为次版本号，版本号编码方式为 BCD 码

dwBoardID: 拨码开关 ID (板卡不支持)

dwSN: 序列号

dwChMax: 通道数目

3.1.3.2 板卡设备总线信息结构

```
typedef struct _CHR_DEVBUSST_
{
    WORD wdDevID;
    WORD wdVenID;
    WORD wdSubDevID;
    WORD wdSubVenID;
    WORD wdBusNum;
    WORD wdDevNum;
    WORD wdFunNum;
    WORD wdIrqNum;
    ULONG dwMemBase[6][2];
} CHR_DEVBUSST, *pCHR_DEVBUSST;
```

wdDevID: 设备 DevID

wdVenID: 设备 VenID

wdSubDevID: 设备 SubDevID

wdSubVenID: 设备 SubVenID

wdBusNum: 总线号

wdDevNum: 总线号

wdFunNum: 功能号

wdIrqNum: 中断号

dwMemBase: 资源信息[1]表示起始地址,[0]表示长度(长度为 0 表示没有)

3.1.3.3 429 配置字结构

```
typedef struct _CHR429_CFGST_
{
    BYTE btTR;
    BYTE btDataNum;
    BYTE btParity;
    BYTE btBaudRate;
} CHR429_CFGST, *pCHR429_CFGST;
```

参数说明:

btTR: 收发选择

1: 接收 2: 发送

btBaudRate: 通道波特率, 取值 0(表示 150000bps)、1(表示 100000 bps)、2(表示 50000 bps)、3(表示 48000 bps)、4(表示 12500 bps)、5(表示 10000bps)

btDataNum: 数据位长度, 取值 25 或 32

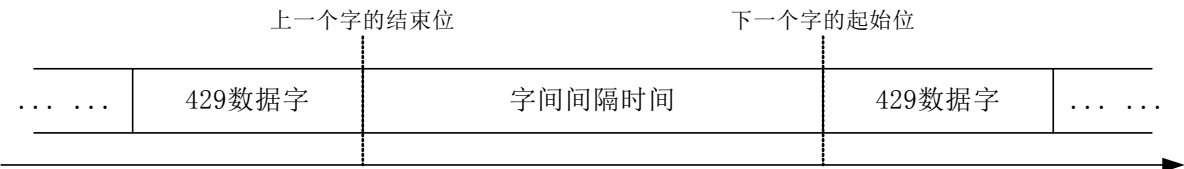
btParity: 校验方式, 取值范围为 0~2, 对应校验方式如下

取值	校验
0	无校验
1	奇校验
2	偶校验

3.1.3.4 定时发送结构

```
typedef struct _CHR429_TXTMST_  
{  
    BYTE    btEnable;  
    DWORD   dwTxPeriod;  
    WORD    wdTxNum;  
    WORD    wdGap;  
} CHR429_TXTMST, *pCHR429_TXTMST;
```

参数说明：
btEnable: 1:定时发送模式使能,0:定时发送模式不使能(即单次发送模式,后面的其他参数无效,)
dwTxPeriod: 定时发送消息(一组 429 数据字为一个消息)周期,分辨率为 10us, 为一个 429 数据的开始到下一个 429 数据的开始之间的间隔
wdTxNum: 定时发送的次数(Max=65535), 0 为无限次发送
wdGap: 字间间隔, 取值范围为 4 ~ 65535。间隔大小由发送波特率大小确定。分辨率为 0.5 个 429 数据位发送时间; 当此值小于 4 时, 函数自动设置为 4; 字间间隔为上一个字的结尾到下一个字的开头, 如下图所示:



3.1.3.5 S/D 标号过滤结构

```
typedef struct _CHR429_LABFILTST_  
{  
    BYTE btFilterAry[4][256];  
} CHR429_LABFILTST, *pCHR429_LABFILTST;
```

参数说明：
btFilterAry: S/D 标号过滤数组, 数组的行坐标值代表 S/D 号, 列坐标值代表 Label 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: 标号过滤使能(见 [3.3.18 CHR429_ChRx_LabelFilter](#))
btFilterAry [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
btFilterAry [1][2]=0: 表示 S/D 为 1 且 Label 为 2 的 429 数据参与过滤, 即不被接收
标号过滤不使能(见 [3.3.18 CHR429_ChRx_LabelFilter](#))
btFilterAry [1][2]=1: 表示 S/D 为 1 且 Label 为 2 的 429 数据不参与过滤, 即被接收
btFilterAry [1][2]=0: 表示 S/D 为 1 且 Label 为 2 的 429 数据不参与过滤, 即被接收
SD 和 LABEL 见[图 3-1](#)和[图 3-2](#)定义。

3.2 驱动程序函数功能

3.2.1 主要功能

驱动软件功能主要是 16 路 429 功能。

3.2.2 功能设计

- 打开/复位/关闭板卡
- 读取板卡序列号/版本号
- 接收和发送通道的配置
 - 16 路接收、16 路发送配置
 - 每通道独立设置
 - 标准 ARINC429 字格式转换与否可选择
- 发送通道
 - ▶ 每路发送通道 FIFO 大小为 511×32 位
 - ▶ 发送 FIFO 可设置复位
 - ▶ 具有内部时钟触发功能
 - 可设置消息(一组 429 数据字为一个消息)周期, 字(429 数据字)间隔和发送消息的预定数量, 实现定时发送。
 - ▶ 发送波特率 150K bps、100Kbps、50Kbps、48Kbps、12.5Kbps、10Kbps 可设置。
- 接收通道
 - ▶ 每路接收 FIFO 大小 $(1M-1) \times 32$ 位
 - ▶ 接收使能和接收 FIFO 复位功能。
 - ▶ 接收标号过滤功能。
 - ▶ 接收添加时标功能。
 - ▶ 接收波特率 150K bps、100Kbps、50Kbps、48Kbps、12.5Kbps、10Kbps 可设置。

3.3 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 CHR429_Open

函数原型: int __stdcall *CHR429_Open* (OUT HANDLE*phDevice,IN BYTE btCardId)

函数功能: 打开板卡, 找到板卡资源的入口, 并分配驱动接口所需要的资源

参数说明: phDevice: 板卡句柄的指针

btCardID: 板卡号, 取值范围为 0 ~ 127, 当主机中只插入一块卡时, 该值为 0; 当主机中同时插入了多块卡时, 需要通过指定不同的板卡号对板卡进行识别

返回值: 如果函数执行成功, 函数返回值为 CHR_SUCCESS; 否则为错误代码

3.3.2 CHR429_Close

函数原型: BOOL __stdcall **CHR429_Close** (IN HANDLE hDevice)

函数功能: 关闭板卡, 释放资源

参数说明: hDevice: 板卡的句柄

返回值: 若板卡关闭成功, 返回值为真; 否则为假

3.3.3 CHR429_Reset

函数原型: BOOL __stdcall **CHR429_Reset** (IN HANDLE hDevice)

函数功能: 复位板卡, 板卡将清空接收 FIFO; 停止发送通道的数据发送; 标号过滤模式不使能。
内、外触发不使能; 内触发周期、触发延迟时间、字间间隔、消息定时发送的次数清 0。板卡的设置都将清除。应户需要重新设置接收和发送通道的配置字。

参数说明: hDevice: 板卡的句柄

返回值: 若板卡复位成功, 返回值为真; 否则假

3.3.4 CHR429_GetDevParInfo

函数原型: BOOL __stdcall **CHR429_GetDevParInfo** (IN HANDLE hDevice,
OUT pCHR_DEVPARST pstParinfo)

函数功能: 读取板卡的硬件参数信息

参数说明: hDevice: 板卡的句柄

pstParinfo: 设备参数结构信息的结构体指针

返回值: 若操作成功, 返回值为真 (TRUE=1); 否则假 (FALSE=0)

3.3.5 CHR429_GetDevBusInfo

函数原型: BOOL __stdcall **CHR429_GetDevBusInfo** (IN HANDLE hDevice,
OUT pCHR_DEVBUSST pstBusinfo)

函数功能: 读取板卡的硬件参数信息

参数说明: hDevice: 板卡的句柄

pstBusinfo: 设备总线结构信息的结构体指针

返回值: 若操作成功, 返回值为真 (TRUE=1); 否则假 (FALSE=0)

3.3.6 CHR429_StartTimeTag

函数原型: BOOL __stdcall **CHR429_StartTimeTag** (IN HANDLE hDevice,
IN BYTE btEnabled);

函数功能：开启/停止时标计数

参数说明：hDevice：板卡的句柄

btEnable：时标计数器开启控制，1 表示开启时标计数器；0 表示停止时标计数器

返回值：若板卡复位成功，返回值为真；否则假

注：CHR429_StartTimeTag 使能开启后，硬件时标已经开启。ZH16_ChRx_AddTimeTag 函数仅告诉对应通道接收数据时添加时标

3.3.7 CHR429_Ch_Reset

函数原型：int __stdcall *CHR429_ARINC42_Ch_Init* (IN HANDLE hDevice,
IN BYTE btCh);

函数功能：429 通道复位

参数说明：hDevice：板卡的句柄

btCh：通道号，取值范围为 0 ~ 15

返回值：如果函数执行成功，函数返回值为 CHR_SUCCESS；否则为错误代码

3.3.8 CHR429_Ch_SetWorkMode

函数原型：int __stdcall *CHR429_Ch_SetWorkMode* (IN HANDLE hDevice,
IN BYTE btCh, IN BYTE btNormal);

函数功能：设置 429 工作模式

参数说明：hDevice：板卡的句柄

btCh：通道号，取值范围为 0 ~ 15

btNormal：429 工作模式选择位

1：正常工作模式

0：自检工作模式

在自检模式下，发送通道的输出在内部被连接到接收通道。发送通道仍正常输出；

发送与接收通道的波特率、数据位长度必须一致

返回值：若 429 工作模式设置成功，函数返回值为 CHR_SUCCESS；否则为错误代码

3.3.9 CHR429_Ch_Init

函数原型：int __stdcall *CHR429_ARINC42_Ch_Init* (IN HANDLE hDevice,
IN BYTE btCh, IN pCHR429_CFGST pstChCfg);

函数功能：初始化 429 配置信息

参数说明：hDevice：板卡的句柄

btCh: 通道号, 取值范围为 0 ~ 15

pstChCfg: 429 初始化结构

返回值: 如果函数执行成功, 函数返回值为 CHR_SUCCESS; 否则为错误代码

3.3.10 CHR429_Ch_ResetFIFO

函数原型: int __stdcall *ZHI6D_Ch_ResetFIFO* (IN HANDLE hDevice,
IN BYTE btCh,IN BYTE btTR);

函数功能: 清除接收/发送 FIFO

参数说明: hDevice: 板卡的句柄

btCh: 通道号, 取值范围为 0 ~ 15

btTR: 值为 1 时, 接收缓冲区清除; 值为 2 时, 表示发送缓冲区清除

返回值: 如果函数执行成功, 函数返回值为 CHR_SUCCESS; 否则为错误代码

3.3.11 CHR429_Ch_WordConvertEnable

函数原型: int __stdcall *CHR429_Ch_WordConvertEnable* (IN HANDLE hDevice,
IN BYTR btCh,IN BYTE btEnable);

函数功能: 使能/禁用 429 字格式转换

参数说明: hDevice: 板卡的句柄

btCh: 通道号, 取值范围为 0 ~ 15

btEnable: 转换使能位 1: 转换接收和发送 0: 不转换

返回值: 如果函数设置成功, 函数返回值为 CHR_SUCCESS; 否则为错误代码

备 注:

	转换	不转换
32 位发送	PC 数据与 429 总线数据都为格式 2。串行数据最高位可以设置成校验位或数据位。	PC 数据: 格式 1 429 总线数据: 格式 2
32 位接收	PC 数据与 429 总线数据都为格式 2。接收最高位是奇偶校验位。	
25 位发送	PC 数据低 25 位与 429 总线数据都为格式 2。串行数据最高位可以设置成校验位或数据位。	PC 数据: 格式 1 429 总线数据: 格式 2
25 位接收	PC 数据低 25 位与 429 总线数据都为格式 2。接收 bit24 是奇偶校验位。	

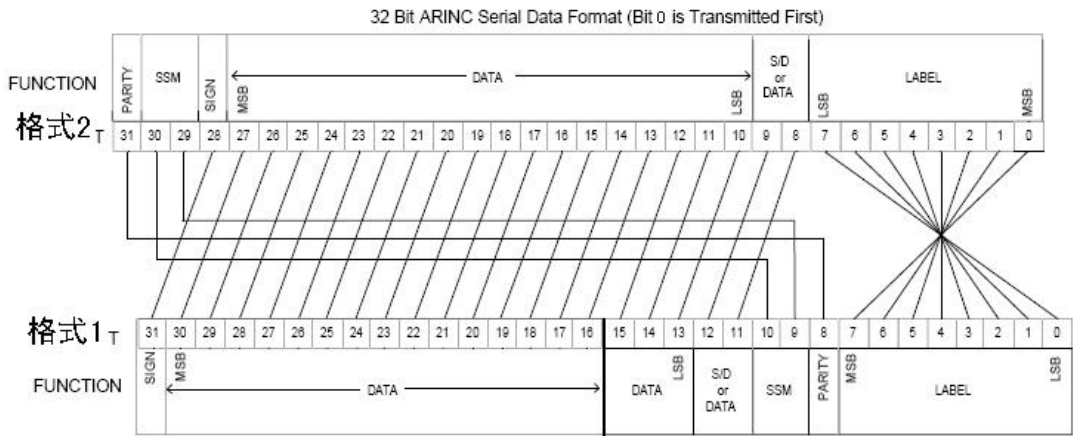


图 3-1 32 位转换

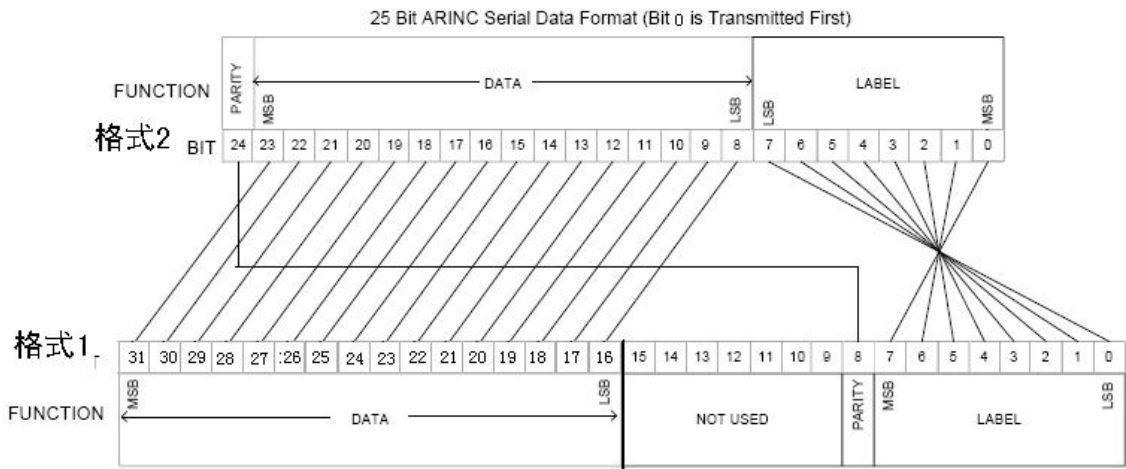


图 3-2 25 位转换

3.3.12 CHR429_ChTx_TimerCFG

函数原型: `int __stdcall CHR429_ChTx_TimerCFG (IN HANDLE hDevice,`
`IN BYTE btCh, IN pCHR429_TXTMST pstTxTmCfg);`

函数功能: 429 定时发送功能配置

参数说明: hDevice: 板卡的句柄

Ch: 通道号, 取值范围为 0 ~ 15

pstTxTmCfg: 429 定时发送配置结构

返回值: 若函数执行成功, 函数返回值为 CHR_SUCCESS; 否则为错误代码

3.3.13 CHR429_ChTx_Data

函数原型: `int __stdcall CHR429_ChTx_Data (IN HANDLE hDevice, IN BYTE btCh,`
`IN WORD wdTxLen, OUT DWORD *pdwBuf,`
`OUT WORD *wdResult);`

函数功能: 向发送 FIFO 写入数据(没有设置定时发送, 调用该函数后立即直接发送, 否则定时发送, 需要调用 CHR429_ChTx_Start 才能发送)

参数说明: hDevice: 板卡的句柄

btCh: 发送通道号, 取值 0 ~ 15

wdTxLen: 待发送的 429 数据量, 最多为 511 个 429 数据

pdwBuf: 存放待发送的 429 数据, 数组的有效长度为 TxLen

wdResult: 返回实际发送的数据量

返回值: 若函数执行成功, 函数返回值为 CHR_SUCCESS; 否则为错误代码

3.3.14 CHR429_ChTx_RdFIFOCnt

函数原型: int __stdcall *CHR429_ChTx_GetFIFOCnt* (IN HANDLE hDevice,
IN BYTE btCh,OUT DWORD *dwFIFOCnt);

函数功能: 读取发送 FIFO 里的数据量

参数说明: hDevice: 板卡的句柄

btCh: 发送通道号, 取值 0 ~ 15

dwFIFOCnt: 发送 FIFO 里的数据量

返回值: 发送 FIFO 里的数据量, 最多为 511, 单位 DWORD(32 位)

3.3.15 CHR429_ChTx_RdFIFOSTR

函数原型: int __stdcall *CHR429_RdFIFOSTR* (IN HANDLE hDevice,
IN BYTE ChanNo,OUT BYTE *btFIFOStr);

函数功能: 读取发送 FIFO 的状态

参数说明: hDevice: 板卡的句柄

ChanNo: 发送通道号, 取值 0 ~ 15

btFIFOStr: FIFO 状态标志

值	定义	描述
0x00	FIFONENF	FIFO 非空非满
0x01	FIFOEMPTY	FIFO 空
0x02	FIFOFULL	FIFO 满

返回值: 若函数设置成功, 函数返回值为 CHR_SUCCESS; 否则为错误代码

3.3.16 CHR429_ChTx_Start

函数原型: int __stdcall *CHR429_ChTx_Start* (IN HANDLE hDevice,
IN BYTE btCh, IN BYTE btEnable);

函数功能: 开始/停止数据的定时发送 (定时发送使用该函数, 单次发送不需要该函数)

参数说明: hDevice: 板卡的句柄

btCh: 发送通道号, 取值 0 ~ 15

btEnable: 发送控制位

1: 开始数据的定时发送

0: 停止数据的定时发送

返回值: 若函数设置成功, 函数返回值为 CHR_SUCCESS; 否则为错误代码

3.3.17 CHR429_ChRx_AddTimeTag

函数原型: int __stdcall *CHR429_ChRx_AddTimeTag* (IN HANDLE hDevice,
IN BYTE btCh, IN BYTE btEnable);

函数功能: 使能/禁用时间标签功能

参数说明: Handle: 板卡的句柄

Ch: 通道号, 取值范围为 0 ~ 15

Enable: 时标使能控制位

1: 使能时标, 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

0: 不使能

返回值: 若时间标签功能设置成功, 函数返回值为 CHR_SUCCESS; 否则为错误代码

3.3.18 CHR429_ChRx_LabelFilter

函数原型: int __stdcall *CHR429_ChRx_LabelFilter* (IN HANDLE hDevice,
IN BYTE btCh, IN BYTE btEnable,
IN pCHR429_LABFILTST pstLabFilter);

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

参数说明: hDevice: 板卡的句柄

btCh: 通道号, 取值范围为 0 ~ 15

btEnable: 标号过滤表使能位

1: 接收通道将只接收标号和 S/D 码符合标号过滤表结构中的 429 数据,
标号过滤表通过 pLabFilter 设置

0: 接收所有标号的 429 数据 (注: btFilterAry 设置无效)

pstLabFilter: S/D 标号过滤表, 包含一个 BYTE btFilterAry[4][256] 二维数组, 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: FilterAry [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

返回值： 若标号过滤表设置成功，函数返回值为 CHR_SUCCESS；否则为错误代码

3.3.19 CHR429_ChRx_SetInt

函数原型： int __stdcall *CHR429_ChRx_SetInt* (IN HANDLE hDevice,
IN BYTE btCh, IN BYTE btIntEnable,
IN WORD wdDepth);

函数功能： 接收中断使能，触发深度，及触发事件设置

参数说明： hDevice: 板卡的句柄

btCh: 通道号，取值范围为 0 ~ 15

btIntEnable: 接收 FIFO 到达触发深度中断使能

1: 中断使能，当接收 FIFO 里的 429 数据量大于或等于用户设置的触发深度值时，将会触发一个中断

0: 禁止中断

wdDepth: FIFO 触发深度值(btIntEnable=1 即中断使能时该参数有效)

取值 0~32767（接收**有时标**）

取值 0~65535（接收**无时标**）

若接收 FIFO 里的 429 数据量大于或等于设置的触发深度值时，则有中断发生。

若已调用 *CHR429_ChRx_CreateIntEvt* 创建中断事件，则中断事件有信号。

注意： 如果数据的流量比较大，触发深度需根据实际情况而定。因为这时系统会反复产生中断，

中断的优先级比较高，会占用太多的系统资源，效率反而降低。

返回值： 若函数执行成功，函数返回值为 CHR_SUCCESS；否则为错误代码

3.3.20 CHR429_ChRx_GetInt

函数原型： int __stdcall *CHR429_ChRx_GetInt* (IN HANDLE hDevice,
IN BYTE btCh, OUT BYTE *btIntEnable,
OUT WORD *wdDepth);

函数功能： 接收中断使能，触发深度，及触发事件设置

参数说明： hDevice: 板卡的句柄

btCh: 通道号，取值范围为 0 ~ 15

btIntEnable: 接收 FIFO 到达触发深度中断使能

1: 中断使能，当接收 FIFO 里的数据量大于或等于用户设置的触发深度值时，将会触发一个中断

0: 禁止中断

dwDepth: FIFO 触发深度值(btIntEnable=1 即中断使能时该参数有效)

取值 0~32767 (接收有时标)

取值 0~65535 (接收无时标)

若接收 FIFO 里的数据量大于或等于设置的触发深度值时, 则有中断发生。

若已调用 *CHR429_ChRx_CreateIntEvt* 创建中断事件, 则中断事件有信号。

注意: 如果数据的流量比较大, 触发深度需根据实际情况而定。因为这时系统会反复产生中断,

中断的优先级比较高, 会占用太多的系统资源, 效率反而降低。

返回值: 若函数执行成功, 函数返回值为 CHR_SUCCESS; 否则为错误代码

3.3.21 CHR429_ChRx_CreateIntEvt

函数原型: int __stdcall *CHR429_ChRx_CreateIntEvt* (IN HANDLE hDevice,
OUT HANDLE *phEvt)

函数功能: 接收中断关闭

参数说明: hDevice: 板卡的句柄

*phEvt: 接收中断事件句柄

返回值: 若函数执行成功, 函数返回值为 CHR_SUCCESS; 否则为错误代码

内部部分代码 *phEvt =CreateEvent(NULL, FALSE/*自动事件*/, FALSE/*开始无信号*/, NULL);

3.3.22 CHR429_ChRx_WaitIntEvt

函数原型: DWORD __stdcall *CHR429_ChRx_WaitIntEvt* (IN HANDLE hDevice,
IN HANDLE hEvt)

函数功能: 等待中断事件 (需要在线程中等待)

参数说明: hDevice: 板卡的句柄

phEvt: 接收中断事件句柄数组

返回值: 返回值每一位表示对应通道事件有信号(1),无信号 (0), 见下表描述

D31	D30	D29	D28	D27	D26	D25	D24
D23	D22	D21	D20	D19	D18	D17	D16
D15	D14	D13	D12	D11	D10	D9	D8
通道 15 1 有事件 0 无事件	通道 14 1 有事件 0 无事件	通道 13 1 有事件 0 无事件	通道 12 1 有事件 0 无事件	通道 11 1 有事件 0 无事件	通道 10 1 有事件 0 无事件	通道 9 1 有事件 0 无事件	通道 8 1 有事件 0 无事件

D7	D6	D5	D4	D3	D2	D1	D0
通道 7	通道 6	通道 5	通道 4	通道 3	通道 2	通道 1	通道 0
1 有事件	1 有事件	1 有事件	1 有事件	1 有事件	1 有事件	1 有事件	1 有事件
0 无事件	0 无事件	0 无事件	0 无事件	0 无事件	0 无事件	0 无事件	0 无事件

注：如果没有事件到来，该函数在线程一直等待，具体参考 MSDN 的 **WaitForSingleObject**

内部部分代码 `WaitForSingleObject(hEvt, INFINITE);`

3.3.23 CHR429_ChRx_CloseIntEvt

函数原型：int __stdcall **CHR429_ChRx_CloseIntEvt**(IN HANDLE hDevice,
IN HANDLE hEvt)

函数功能：接收中断关闭

参数说明：hDevice：板卡的句柄

hEvt：接收中断事件句柄

返回值：若函数执行成功，函数返回值为 CHR_SUCCESS；否则为错误代码
内部部分代码 `CloseHandle(hEvt);`

3.3.24 CHR429_ChRx_Start

函数原型：int __stdcall **CHR429_ChRx_Start** (IN HANDLE hDevice,
IN BYTE btCh, IN BYTE btEnable);

函数功能：开始和停止数据的接收

参数说明：hDevice：板卡的句柄

btCh：通道号，取值范围为 0 ~ 15

btEnable：429 数据开始接收控制位

1：开始 429 数据的接收 0：停止 429 数据的接收

返回值：若函数执行成功，函数返回值为 CHR_SUCCESS；否则为错误代码

3.3.25 CHR429_ChRx_RdFIFOCnt

函数原型：int __stdcall **CHR429_ChRx_RdFIFOCnt** (IN HANDLE hDevice,
IN BYTE btCh , OUT DWORD *pdwFIFOCnt);

函数功能：获取接收 FIFO 里的数据量。每路接收缓冲区最多可容纳 1M 个 429 数据字

参数说明：hDevice：板卡的句柄

btCh：通道号，取值范围为 0 ~ 15

dwFIFOCnt：接收 FIFO 里的数据量(包含时标)

返回值：若函数执行成功，函数返回值为 CHR_SUCCESS；否则为错误代码

注：若接收有时标，获取的数据量为 4 时，真实数据描述如下：

429 数据	时标	429 数据	时标
--------	----	--------	----

若接收无时标，获取的数据量为 4 时，真实数据描述如下：

429 数据	29 数据	429 数据	29 数据
--------	-------	--------	-------

3.3.26 CHR429_ChRx_RdFIFOStatus

函数原型：int __stdcall *CHR429_ChRx_RdFIFOStatus* (IN HANDLE hDevice,
IN BYTE btCh, OUT BYTE *pbtFIFOStr);

函数功能：获取接收 FIFO 的状态

参数说明：hDevice：板卡的句柄

btCh：通道号，取值范围为 0 ~ 15

btFIFOStr: FIFO 状态标志

值	定义	描述
0x00	FIFONENT	FIFO 非空非满
0x01	FIFOEMPTY	FIFO 空
0x02	FIFOFULL	FIFO 满

返回值： 若函数执行成功，函数返回值为 CHR_SUCCESS；否则为错误代码

3.3.27 CHR429_ChRx_Data

函数原型：int __stdcall *CHR429_ChRx_Data* (IN HANDLE hDevice,
IN BYTE btCh, IN WORD wdRxLen,
OUT DWORD*pdwBuf, OUT WORD *pwdResult);

函数功能：数据的接收

参数说明：hDevice：板卡的句柄

btCh：接收通道号，取值 0 ~ 15

wdRxLen：待接收的数据量(从 CHR429_ChRx_RdFIFOCnt 获取数据量)

pdwRxBuf：存放接收到的 429 数据 ，最多接收 65535 个（429 数据,32bit 时标）

无时标：最多接收 65535 个 429 数据

有时标：最多接收 32767 个 429 数据和 32767 个时标

有时标，获取的数据及时标如下：

429 数据	时标	...	429 数据	时标
--------	----	-----	--------	----

无时标，获取的数据如下：

429 数据	29 数据	...	429 数据	29 数据
--------	-------	-----	--------	-------

pwdResult：实际接收到的数据量(包含时标)

返回值： 若函数执行成功，函数返回值为 CHR_SUCCESS；否则为错误代码

3.4 错误码列表

错误定义	错误返回值	说明
CHR_SUCCESS	0x00000000	函数执行正确
CHR_ERRCODE1	(CHR_SUCCESS+1)	配置数据丢失
CHR_ERRCODE2	(CHR_SUCCESS+2)	设备句柄无效
CHR_ERRCODE3	(CHR_SUCCESS+3)	非法输入参数
CHR_ERRCODE4	(CHR_SUCCESS+4)	寄存器写操作失败
CHR_ERRCODE5	(CHR_SUCCESS+5)	寄存器读操作失败
CHR_ERRCODEF	(CHR_SUCCESS+0x0F)	未知其他错误
...	...	...
CHR_ERRCODE_F0	(CHR_SUCCESS+0xF0)	硬件内部通道数不兼容(BAR0 的 0xFC 与 EEPROM 邮箱不同)
CHR_ERRCODE_F1	(CHR_SUCCESS+0xF1)	CHR429_ChRx_CreateIntEvt 内部创建事件失败

3.5 驱动接口函数调用步骤

3.5.1 板卡操作

- 打开板卡 (CHR429_Open)
- 获取板卡信息 (CHR429_GetDevInfo)
- 复位板卡 (CHR429_Reset)
- 开启板卡时标 (CHR429_StartTimeTag)
- 创建接收中断事件 (CHR429_ChRx_CreateIntEvt)

3.5.2 通道公共参数配置

- 通道复位 (CHR429_Ch_Reset)
- 通道工作模式[自环/正常] (CHR429_Ch_SetWorkMode)
- 通道初始化参数配置 (CHR429_Ch_Init)
- 通道复位 FIFO[发送/接收] (CHR429_Ch_ResetFIFO)
- 通道 429 字转换使能<可选> (CHR429_Ch_WordConvertEnable)

3.5.3 数据的接收

- 时标使能<可选> (CHR429_ChRx_AddTimeTag)
- SD 标号过滤功能设置<可选> (CHR429_ChRx_LabelFilter)
- 中断使能<可选> (CHR429_ChRx_SetInt)
- 接收使能 (CHR429_ChRx_Start)
- 查询接收
 - ✧ 读取接收 FIFO 的数据量 (CHR429_ChRx_RdFIFOCnt)
 - ✧ 若接收 FIFO 的数据量不为 0, 则读取数据 (CHR429_ChRx_Data)
- 中断接收
 - ✧ 等待中断事件<可选> (CHR429_ChRx_WaitIntEvt)
 - ✧ 读取接收 FIFO 的数据量 (CHR429_ChRx_RdFIFOCnt)
 - ✧ 若接收 FIFO 的数据量不为 0, 则接收数据 (CHR429_ChRx_Data)
- 接收禁止 (CHR429_ChRx_Start)
- 中断禁止 (CHR429_ChRx_SetInt)

注：CHR429_ChRx_WaitIntEvt 使用前需要使用 CHR429_ChRx_CreateIntEvt 创建事件

3.5.4 数据的发送

➤ 非定时发送

- ✧ a 发送模式（将函数 CHR429_ChTx_TimerCFG 的使能 btEnable =0）
- ✧ b 读取发送 FIFO 的状态（CHR429_ChTx_RdFIFOSTR）
- ✧ c 若发送 FIFO 不满，则发送数据（CHR429_ChTx_Data）
- ✧ 重复 b~c 两步

➤ 定时发送

- ✧ 发送模式（将函数 CHR429_ChTx_TimerCFG 的 btEnable =1）
 - ✓ 设置消息周期
 - ✓ 设置消息发送次数
 - ✓ 设置 429 字间间隔
- ✧ 停止数据的发送（CHR429_ChTx_Start(FALSE) ）
- ✧ 写入待发送的 429 数据（CHR429_ChTx_Data ）
- ✧ 开始数据的发送（CHR429_ChTx_Start(TRUE) ）

注：在定时发送的过程中若需要修改下次发送的数据，调用 CHR429_TxWrite

3.5.5 关闭板卡

关闭接收中断事件（CHR429_ChRx_CloseIntEvt）

应用程序退出时，复位板卡（CHR429_Reset），并关闭板卡（CHR429_Close）

第四章 功能演示软件

CHR32XXX 应用程序包含 16 路接收通道，16 路发送通道。本应用程序适用于 16 路接收通道、16 路发送通道；8 路接收通道、8 路发送通道；4 路接收通道、4 路发送通道；2 路接收通道、2 路发送通道。

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 98/2000/2003/xp/Win7 操作系统

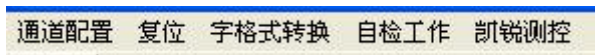
4.2 主要功能



图 4.2.1 应用程序主界面

4.3 图形化接口设计

4.3.1 菜单



4.3.1.1 板卡配置（包括复位，字格式转换，自检工作）

✧ 复位

复位

板卡复位

所有通道复位

当前通道复位

- ✓ 板卡复位
整个板卡复位=所有通道复位+时标使能复位
- ✓ 所有通道复位
每个通道的参数复位
- ✓ 当前通道复位
当前通道复位，取决于用户选择的当前通道页面

✧ 字格式转换

字格式转换 当前通道使能	字格式转换 ✓ 当前通道使能
------------------------------------	--------------------------------------

- ✓ 当前通道(字转换)使能
设置当前通道字转换使能和不使能，同时可以记住当前通道是否为字转换模式还是不转换模式

✧ 自检工作

自检工作 当前通道使能	自检工作 ✓ 当前通道使能
-----------------------------------	-------------------------------------

- ✓ 当前通道(自检)使能
设置当前通道自检使能和不使能，同时可以记住当前通道是否为自检模式还是正常模式

4.3.1.2 通道配置（接收和发送）

通道配置

接收参数

发送参数

✧ 接收配置

- ✓ 波特率（支持 150K，100K，50K，48K，12.5K，10Kbps）
- ✓ 429 字长（25 位，32 位）
- ✓ 校验（无校验，奇校验，偶校验）
- ✓ 接收方式（查询，中断）
- ✓ 时标（无，有）
- ✓ SD+Label 使能（使能，禁止）
- ✓ SD+Label 具体设置项(如: 0-00;0-01 表示 SD=0, LABEL=0x00 和 SD=0, LABEL=0x01)



图 4.3.1.2.1 通道接收参数配置窗口

SD 和 LABEL 过滤设置，每个通道包含 SD0~3 和 LABEL00~FF，配置完一个通道，需要选择保存按钮，再进行下一个通道的配置，否则上一个通道的配置将丢失。

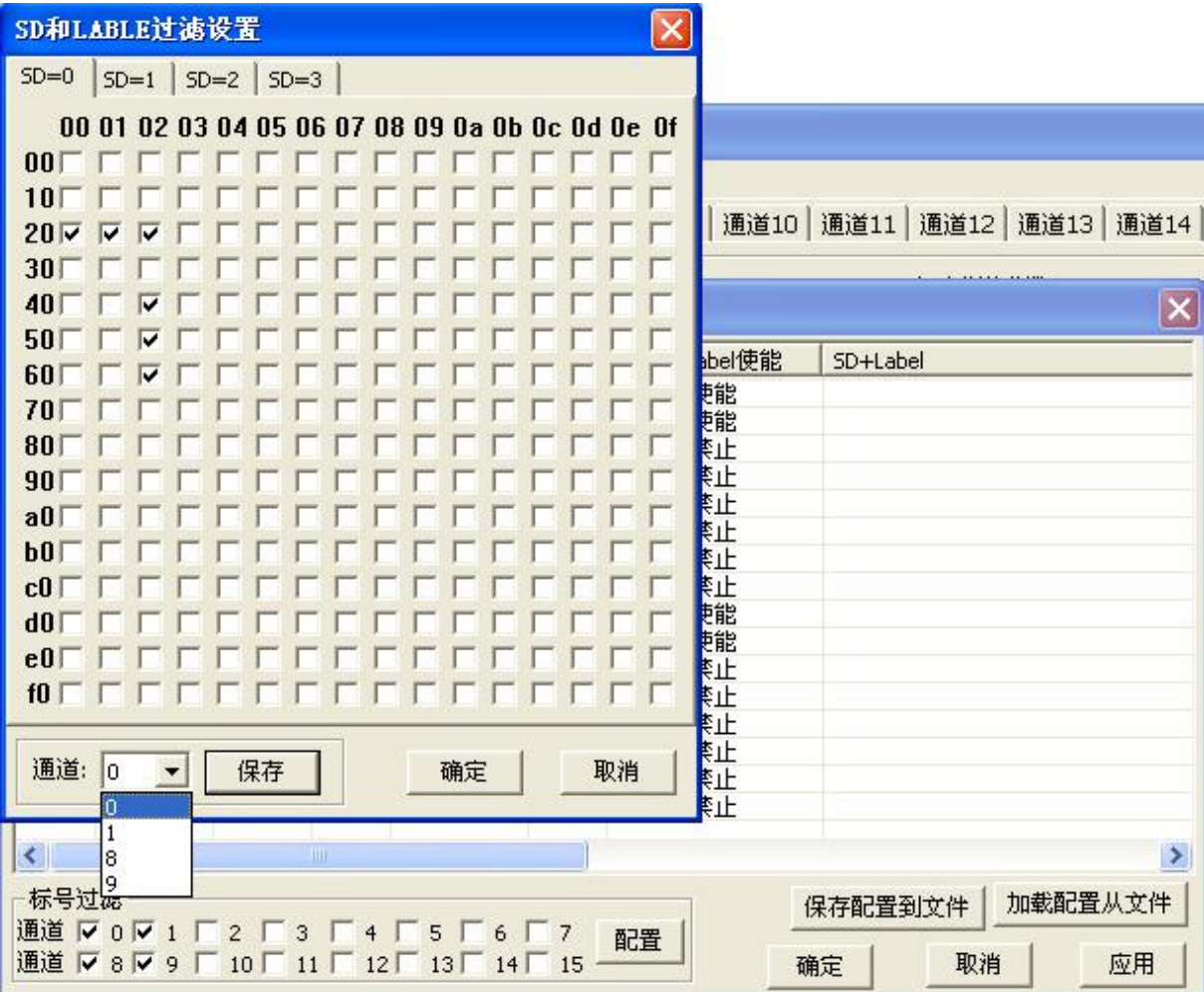


图 4.3.1.2.2 通道接收参数 SD+LABEL 过滤设置窗口

- ✧ 发送配置
 - ✓ 波特率（支持 150K，100K，50K，48K，12.5K，10Kbps）
 - ✓ 429 字长（25 位，32 位）
 - ✓ 校验（无校验，奇校验，偶校验）
 - ✓ 发送方式（单次，定时）



图 4.3.1.2.3 通道发送参数配置窗口

4.3.2 界面



图 4.3.2.1 应用程序主界面

4.4 功能描述

4.4.1 接收功能设计



图 4.4.1.1 应用程序主界面

- 数据接收及显示区，红色标记处，当设置为无时表示，仅显示数据列，有时表示数据列和是表列都显示。
- 当接收参数配置为中断使能时，中断触发深度编辑框为可用，同时使用人员知道该通道正在使用中断方式接收数据。
- 实时显示复选框，选中表示接收的数据显示（数据接收及显示区），否则不显示接收到的数据，仅有数据量框显示实时接收到的数据量。
- 保存数据，可以保存 TXT 文本文件，或者保存为 bin 二进制文件，如下图所示

☒ 保存数据(文件保存在exe目录的Save\Ch0.txt)

☒ 保存数据(文件保存在exe目录的Save\Ch0.bin)

- 复位接收 FIFO，复位硬件接收 FIFO。
- 接收，开始接收数据和停止接收数据。

4.4.2 发送功能设计

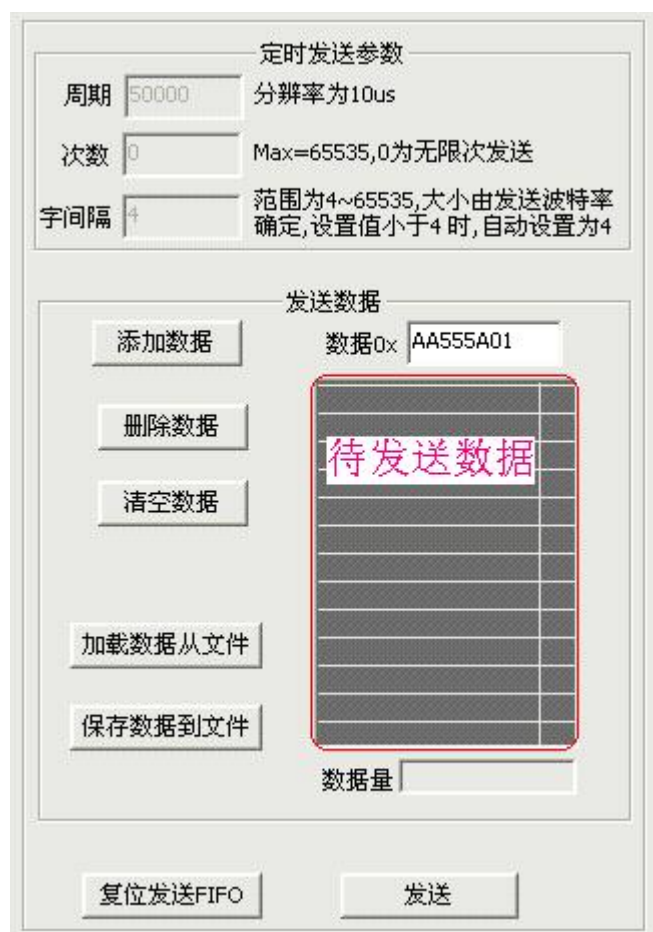


图 4.4.2.1 应用程序主界面

- 待发送数据（数据即 429 字），红色标记处，待发送数据不为空才能发送数据，否则不能进行发送。待发送数据最多为 511 个，即 1~511 个数据可以组成一个消息，允许重复 LABEL 号数据。
- 当发送参数配置为定时发送时，定时发送参数中的编辑框为可用，定时发送参数包括消息周期，消息发送次数，429 字间间隔。
- 添加数据，添加待发送数据。
- 删除数据，删除待发送数据。
- 清空数据，清空待发送数据。
- 加载数据从文件，加载用户以前保存的待发送数据。
- 保存数据到文件，保存待发送数据到制定文件，为下一次“加载数据从文件”使用。
- 数据量，待发送数据的个数，即消息包含多少个 429 字，最大值为 511。
- 复位发送 FIFO，发送前应该清除硬件发送 FIFO。当定时发送过程中，选择“复位发送 FIFO”，定时发送因为 FIFO 中没有数据，接收方将接收不到数据。
- 发送，设置为单次发送时，选择一次发送一次；设置为定时发送（周期发送）时，选择“发送”后，将变为“停止定时发送”，选择“停止定时发送”即可停止正在周期发送的数据，同时恢复“发送”。

附录 (ARINC429 通讯中奇偶校验位)

A 原理

A.1 接收恒定为奇校验时，发送校验方式为无校验，奇校验，偶校验

- 当发送为无检验时：
 - ✧ 判断发送的数据中“1”的个数，偶数个“1”时奇偶校验位恒为 1；奇数个“1”时奇偶校验位恒为 0。
- 当发送为奇检验时：
 - ✧ 无需判断发送的数据中“1”的个数，奇偶校验位恒为 0。
- 当发送为偶检验时：
 - ✧ 无需判断发送的数据中“1”的个数，奇偶校验位恒为 1。

举例：

发送数据 0x00000001，在发送为无校验时接收到 0x 00000001

在发送为奇校验时接收到 0x 00000001

在发送为偶校验时接收到 0x 00000101

发送数据 0x 00000003，在发送为无校验时接收到 0x 00000103

在发送为奇校验时接收到 0x 00000003

在发送为偶校验时接收到 0x 00000103

A.2 接收恒定为偶校验时，发送校验方式为无校验，奇校验，偶校验

- 当发送为无检验时：
 - ✧ 判断发送的数据中“1”的个数，偶数个“1”时奇偶校验位恒为 0；奇数个“1”时奇偶校验位恒为 1。
- 当发送为奇检验时：
 - ✧ 无需判断发送的数据中“1”的个数，奇偶校验位恒为 1。
- 当发送为偶检验时：
 - ✧ 无需判断发送的数据中“1”的个数，奇偶校验位恒为 0。

举例：

发送数据 0x 00000001，在发送为无校验时接收到 0x 00000101

在发送为奇校验时接收到 0x 00000101

在发送为偶校验时接收到 0x 00000001

发送数据 0x 00000003，在发送为无校验时接收到 00000003

在发送为奇校验时接收到 00000103

在发送为偶校验时接收到 00000003

A.3 接收恒定为无校验时，发送校验方式为无校验，奇校验，偶校验

- 当发送为无检验时：
 - ✧ 接收数据为发送数据。
- 当发送为奇检验时：
 - ✧ 接收数据为发送的奇校验数据。
- 当发送为偶检验时：
 - ✧ 接收数据为发送的偶校验数据。

举例：

发送数据 0x00000001，在发送为无校验时接收到 0x00000001

在发送为奇校验时接收到 0x00000001

在发送为偶校验时接收到 0x00000101

发送数据 0x 00000003，在发送为无校验时接收到 0x 00000003

在发送为奇校验时接收到 0x 00000103

在发送为偶校验时接收到 0x 00000003

B 举例

B.1 发送的原始数据->发送校验后的数据->接收数据

序号	接收设置	发送设置	发送的原始数据	发送校验后的数据	接收数据
1	接收恒为奇校验	发送无校验	0x00000001	0x00000001	0x 00000001
			0x00000003	0x00000003	0x 00000103
		发送奇校验	0x00000001	0x00000001	0x00000001
			0x00000003	0x00000103	0x00000003
		发送偶校验	0x00000001	0x00000101	0x00000101
			0x00000003	0x00000003	0x00000103
2	接收恒为偶校验	发送无校验	0x00000001	0x00000001	0x 00000101
			0x00000003	0x00000003	0x00000003
		发送奇校验	0x00000001	0x00000001	0x00000101
			0x00000003	0x00000103	0x00000103
		发送偶校验	0x00000001	0x00000101	0x00000001
			0x00000003	0x00000003	0x00000003
3	接收恒为无校验	发送无校验	0x00000001	0x00000001	0x00000001
			0x00000003	0x00000003	0x00000003
		发送奇校验	0x00000001	0x00000001	0x00000001
			0x00000003	0x00000103	0x00000103
		发送偶校验	0x00000001	0x00000101	0x00000101
			0x00000003	0x00000003	0x00000003

B.2 发送的原始数据->发送校验后的数据(校验规则)

注：<发送的原始数据>的校验位如果有数据将视为无效数据

发送设置	发送的原始数据	发送校验后的数据	校验规则
发送无校验	0x00000001	0x00000001	<原始数据>和 <校验后的数据>相同
	0x00000003	0x00000003	

发送奇校验	0x00000001	0x00000001	<校验后的数据> “1”的个数为奇数
	0x00000003	0x00000103	
发送偶校验	0x00000001	0x00000101	<校验后的数据> “1”的个数为偶数
	0x00000003	0x00000003	

B.3 发送校验后的数据->接收数据(校验规则)

接收设置	发送设置	发送校验后数据	接收数据	校验规则
接收恒为奇校验	发送无校验	0x00000001	0x 00000001	<接收数据>“1”的个数，偶数个“1”时校验位恒为 1；奇数个“1”时校验位恒为 0
		0x00000003	0x 00000103	
	发送奇校验	0x00000001	0x00000001	<接收数据>校验位恒为 0
		0x00000103	0x00000003	
	发送偶校验	0x00000101	0x00000101	<接收数据>校验位恒为 1
		0x00000003	0x00000103	
接收恒为偶校验	发送无校验	0x00000001	0x 00000101	<接收数据>“1”的个数，偶数个“1”时校验位恒为 0；奇数个“1”时校验位恒为 1
		0x00000003	0x00000003	
	发送奇校验	0x00000001	0x00000101	<接收数据>校验位恒为 1
		0x00000103	0x00000103	
	发送偶校验	0x00000101	0x00000001	<接收数据>校验位恒为 0
		0x00000003	0x00000003	
接收恒为无校验	发送无校验	0x00000001	0x00000001	<发送校验后数据>和<接收数据>相同
		0x00000003	0x00000003	
	发送奇校验	0x00000001	0x00000001	
		0x00000103	0x00000103	
	发送偶校验	0x00000101	0x00000101	
		0x00000003	0x00000003	