

# PCICAN-9820

非智能CAN-bus接口卡

WWW.ITEK.NET.CN

产品用户手册

类别	内容
关键词	CAN-bus接口卡、CAN数据采集卡、PCI转CAN
摘要	



修订历史

版本	日期	原因
V1.00	2011/10/09	创建文档

## 目 录

---

第 1 章 产品简介 .....	4
1.1 概述 .....	4
1.2 产品特性 .....	5
第 2 章 硬件安装与接线 .....	6
2.1 硬件安装 .....	6
2.2 接口定义 .....	6
2.3 终端电阻 .....	7
第 3 章 驱动程序安装 .....	9
3.1 驱动程序安装 .....	9
3.2 驱动程序检查 .....	12
附录 1 CAN2.0B 协议帧格式 .....	13
附录 2 标准波特率设置 .....	15
附录 3 CAN 报文滤波器设置 .....	16
附录 4 CAN 总线通信距离（参考值） .....	21
附录 5 销售清单 .....	22

## 第 1 章 产品简介

通过本章，您可以了解北京爱泰益科科技有限公司出品的PCICAN-9820 CAN-bus接口卡的基本特性。

本章内容提要

- ✓ 概述
- ✓ 产品特性

### 1.1 概述

PCICAN-9820 CAN-bus 接口卡是北京爱泰益科科技有限公司开发的一款工业级计算机 PCI 总线转 CAN 总线的数据传输卡，其集成两路 CAN-bus 接口。PCICAN-9820 接口卡支持 32 位 33MHz PCI 数据总线，兼容 PCI2.2 规范，采用通用 PCI 连接器，用户利用它可以轻松完成 CAN-bus 网络和 PC 之间的互连互通，可以方便、高速的采集 CAN-bus 数据。

PCICAN-9820 为工业级产品，可以工作在 $-20^{\circ}\text{C}\sim+70^{\circ}\text{C}$ 的温度范围内。CAN 口通信波特率可以在 5K~1Mbps 内任意设定。为保证良好的 EMC 和 EMI 性能，PCICAN-9820 采用完全独立的 CAN-bus 通道，有效避免 PC 遭受地环流的影响。同时，两路 CAN 接口均集成 CAN-bus 专用共模滤波器和 TVS 总线保护电路，为工作在电磁环境较复杂的工业场合提供安全保障。

PCI-9820 接口卡提供 API 和多语言版本的简单应用例程，并提供调试软件，方便用户分析、控制、测试 CAN-bus 总线上的传输数据。

## 1.2 产品特性

- PC接口：32位33M PCI数据总线，符合PCI2.2标准，即插即用；
- CAN控制器：NXP SJA1000；
- CAN收发器：Microchip MCP2551；
- CAN协议：CAN 2.0B规范（兼容CAN 2.0A）；
- CAN通信波特率：5K~1Mbps；
- CAN通讯接口：DB9针型插座，符合DeviceNet和CANopen标准；
- 隔离耐压：磁隔离 DC 2500V；
- 工作温度：-20℃~+70℃；
- 储存温度：-40℃~+85℃；
- 物理尺寸：标准PCI短卡130mm×90mm。

## 第 2 章 硬件安装与接线

本章内容主要介绍PCICAN-9820的安装和接口端子定义。

本章内容提要

- ✓ 硬件安装
- ✓ 接口定义
- ✓ 终端电阻

### 2.1 硬件安装

为保证 PCICAN-9820 接口卡能够正常使用，在安装与拆卸时请确认计算机处于关机且断电状态。PCICAN-9820 接口卡是静电敏感型板卡，请在安装与拆卸时注意静电防护，可佩戴防静电手套或手持办卡边缘，避免直接接触元器件。

具体安装步骤如下：

- 1) 关闭PC 电源；
- 2) 打开PC 的盖子；
- 3) 将PCI-9820 接口卡插入空闲的PCI 插槽；
- 4) 拧紧固定板卡的螺钉；
- 5) 打开PC 电源，此时BIOS 会自动给PCI-9820 接口卡分配中断和I/O 地址。

### 2.2 接口定义

PCICAN-9820集成两路CAN通道（通道定义顺序如图2-1），通过DB9针型连接器与CAN-bus网络进行连接。DB9 针型连接器的引脚信号定义如图 2-2 所示。引脚定义符合DeviceNet和CANopen标准。

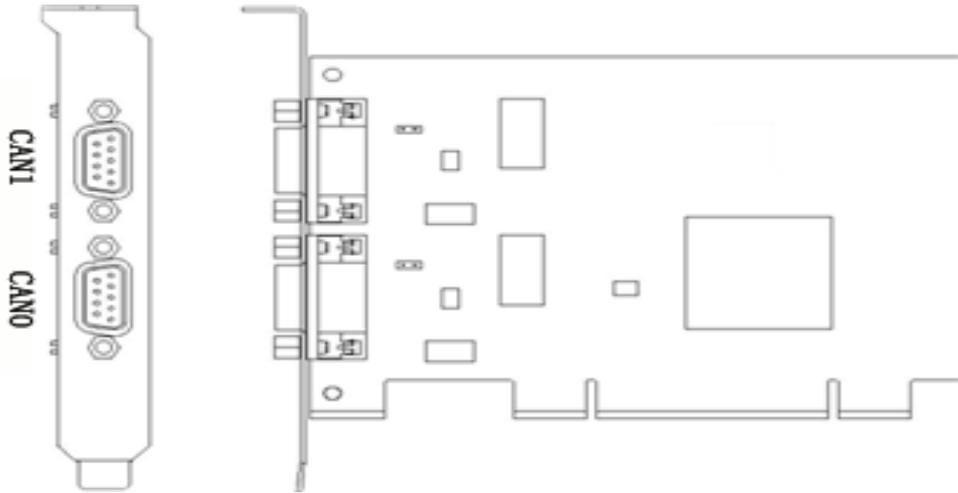


图 2-1 CAN 通道顺序定义



图 2-2 CAN 接口端子定义

### 2.3 终端电阻

如CAN网络采用直线拓扑结构，总线的2个终端需要安装120Ω的终端电阻。如果节点数目大于2，中间节点不需要安装120Ω的终端电阻如图2-3。

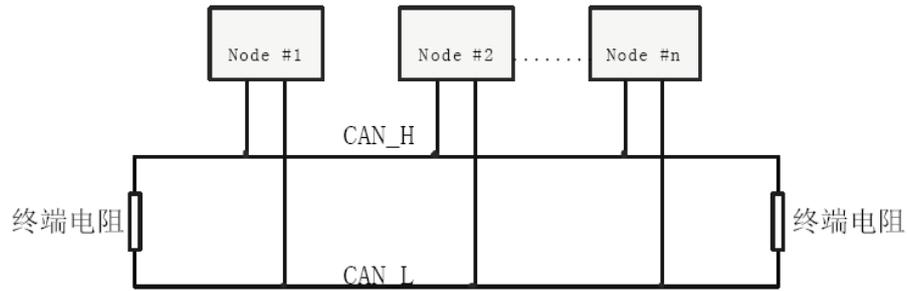


图2-3 总线拓扑结构

PCICAN-9820接口卡的每路CAN通道板上都集成了120Ω终端电阻，可通过拨码开关S1来设定对应的CAN通道是否使用该电阻。如果PCICAN-9820接口卡的对应CAN通道位于一个CAN网络的端点，请将对应通道的拨码开关处于ON的状态。出厂时，两通道拨码开关默认连接状态，即使用集成120Ω终端电阻。

### 第 3 章 驱动程序安装

本章通过 Windows XP 系统介绍 PCICAN-9820 接口卡的驱动程序如何安装。

本章内容提要

- ✓ 设备驱动程序安装
- ✓ 设备驱动程序检查

#### 3.1 驱动程序安装

1) PC 安装好 PCICAN-9820 后，启动 PC 电源；

2) 系统启动后将会提示发现新硬件，并弹出驱动安装引导窗口，如图 3-1。如未弹出驱动安装引导窗口，请右键点击<我的电脑>，选择<属性>，在弹出窗口的<属性>菜单中点击<设备管理器>，选择<其他 PCI 桥设备>，右键菜单中选择<更新驱动程序>（如图 3-2 所示），会弹出图 3-1 的对话框；



图 3-1 驱动安装引导窗口



图 3-2 更新驱动

3) 在图 3-1 窗口选择<从列表或指定位置安装>, 点击<下一步>, 将安装路径指向驱动文件所在的文件夹, 如图 3-3 所示;

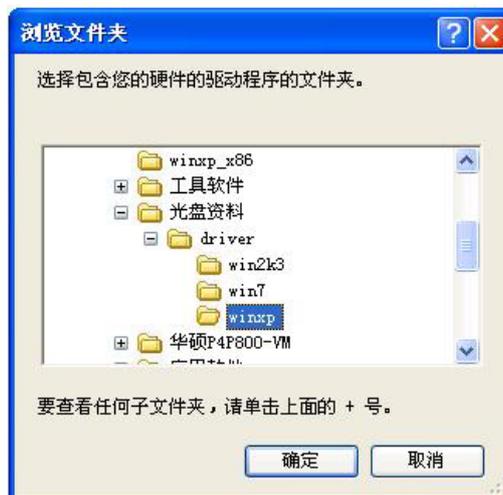


图 3-3 驱动文件路径选择窗口

4) 在图 3-3 窗口点击<确定>按钮, 会弹出驱动正在安装界面, 如图 3-4 所示。安装完毕后系统会提示驱动安装成功, 如图 3-5 所

示；



图 3-4 驱动安装进度界面



图 3-5 驱动安装完成界面

5) 出现图 3-5 的界面后, 代表驱动已经安装完, 点击<确定>按钮后, PCICAN-9820 就可以正常使用了。

### 3.2 驱动程序检查

下面以WinXP操作系统为例，说明如何检查PCICAN-9820接口卡驱动是否正确安装。

#### 1) 打开WINDOWS 设备管理器

鼠标右击桌面上<我的电脑>图标，从下拉菜单中选取<属性>选项；然后，选择<硬件>标签，鼠标单击<设备管理器>按钮打开当前硬件设备列表，如图 3-6 所示。



图3-6 驱动安装状态界面

#### 2) 检查驱动安装状态

检查<iTEKCAN>设备类是否已经在当前硬件列表中, 并且设备描述符为<PCICAN-9820 Interface Card>。设备图标上没有黄色“！”符号或红色“X”符号存在，表示设备驱动安装正确并可以使用。

## 附录 1 CAN2.0B 协议帧格式

### CAN2.0B 标准帧：

CAN 标准帧信息为 11 个字节，包括两部分：信息和数据部分。

前 3 个字节为信息部分。

	7	6	5	4	3	2	1	0
字节 1	FF	RTR	×	×	DLC(数据长度)			
字节 2	ID. 10	ID. 9	ID. 8	ID. 7	ID. 6	ID. 5	ID. 4	ID. 3
字节 3	ID. 2	ID. 1	ID. 0	×	×	×	×	×
字节 4	数据 1							
字节 5	数据 2							
字节 6	数据 3							
字节 7	数据 4							
字节 8	数据 5							
字节 9	数据 6							
字节 10	数据 7							
字节 11	数据 8							

### 说明：

字节 1 为帧信息，第 7 位（FF）表示帧格式，在标准帧中，FF=0；第 6 位（RTR）表示帧的类型，RTR=0 表示为数据帧，RTR=1 表示为远程帧；DLC 表示在数据帧时实际的数据长度。

字节 2、3 为报文识别码，11 位有效。

字节 4~11 为数据帧的实际数据，远程帧时无效。

## CAN2.0B 扩展帧

CAN 扩展帧信息为 13 个字节，包括两部分，信息和数据部分。

前 5 个字节为信息部分。

	7	6	5	4	3	2	1	0
字节 1	FF	RTR	×	×	DLC(数据长度)			
字节 2	ID. 28	ID. 27	ID. 26	ID. 25	ID. 24	ID. 23	ID. 22	ID. 21
字节 3	ID. 20	ID. 19	ID. 18	ID. 17	ID. 16	ID. 15	ID. 14	ID. 13
字节 4	ID. 12	ID. 11	ID. 10	ID. 9	ID. 8	ID. 7	ID. 6	ID. 5
字节 5	ID. 4	ID. 3	ID. 2	ID. 1	ID. 0			
字节 6	数据 1							
字节 7	数据 2							
字节 8	数据 3							
字节 9	数据 4							
字节 10	数据 5							
字节 11	数据 6							
字节 12	数据 7							
字节 13	数据 8							

说明：

字节 1 为帧信息，第 7 位（FF）表示帧格式，在扩展帧中，FF = 1；第 6 位（RTR）表示帧的类型，RTR=0 表示为数据帧，RTR=1 表示为远程帧；DLC 表示在数据帧时实际的数据长度。

字节 2~5 为报文识别码，其高 29 位有效。

字节 6~13 为数据帧的实际数据，远程帧时无效。

## 附录 2 标准波特率设置

	波特率	BTR0	BTR1
1	5	BF	FF
2	10*	31	1C
3	20*	18	1C
4	40	87	FF
5	50*	09	1C
6	80	83	FF
7	100*	04	1C
8	125*	03	1C
9	200	81	FA
10	250*	01	1C
11	400	80	FA
12	500*	00	1C
13	666	80	B6
14	800*	00	16
15	1000*	00	14

注：带\*号的是 CiA 协会推荐的波特率。

### 附录 3 CAN 报文滤波器设置

---

转换器的 CAN 报文滤波器是基于 PHILIPS 公司 CAN 控制器 SJA1000 的 Pelican 模式来进行设计的。SJA1000 的滤波器由 4 组（4 字节）验收代码寄存器（ACR）和 4 组（4 字节）验收屏蔽寄存器（AMR）构成。ACR 的值是预设的验收代码值，AMR 值是用来表征相对应的 ACR 值是否用作验收滤波。

滤波的一般规则是：每一位验收屏蔽分别对应每一位验收代码，当该位验收屏蔽位为1的时候（即设为无关），接收的相应帧ID 位无论是否和相应的验收代码位相同均会表示为接收；但是当验收屏蔽位为0 的时候（即设为相关），只有相应的帧ID 和相应的验收代码位值相同的情况才会表示为接收。并且只有在所有的位都表示为接收的时候，CAN 控制器才会接收该帧报文。

滤波的方式上又分“单滤波”和“双滤波”两种。并且在标准帧和扩展帧情况下滤波又略有不同。在配置软件的“自定义过滤屏蔽码”的情况下开放滤波器所有功能。现阐述如下：

#### 1. 单滤波配置

这种滤波器配置方式可以定义成一个长滤波器。滤波器字节和信息字节之间位的对应关系取决于当前接收帧格式。

**标准帧：**在帧格式为标准帧时，在验收滤波中仅使用ACR前两个字节（ACR3和ACR4）中的部分数据（低11位）来存放过滤验收码。同样，过滤屏蔽码也只采用AMR3和AMR4的低11位。在AMR的位为0时（意

为相关)，当ACR的相对应位（如ACR1.0对应AMR1.0，同时也和ID.00相对应）和接收帧标识的对应位值相同时，表现为“可接收”（逻辑1）；当两者不等时表现为“不接收”（逻辑0）。或者当AMR的位为1时，无论ACR的相对应位和接收帧标识的对应位值是否相同，均表现为“可接收”（逻辑1）。

对于一个成功接收的信息所有单个位的比较后都必须发出接收信号。如图 1所示。

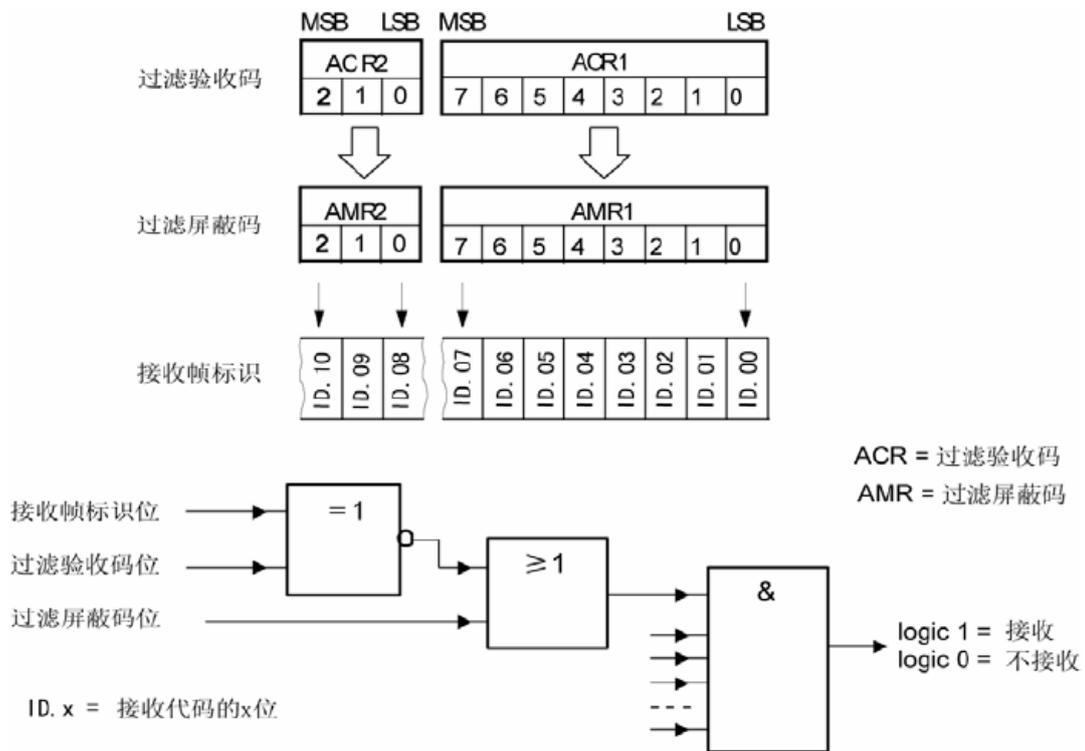


图 1 标准帧单滤波示意图

**扩展帧：**在帧格式为扩展帧时，由于帧标识是29位，所以在验收滤波中使用ACR的四个字节中的部分数据（低29位）来存放过滤验收码。同样，过滤屏蔽码也只采用AMR的低29位。

接收逻辑关系和标准帧相同，逻辑表示如图 2 所示。

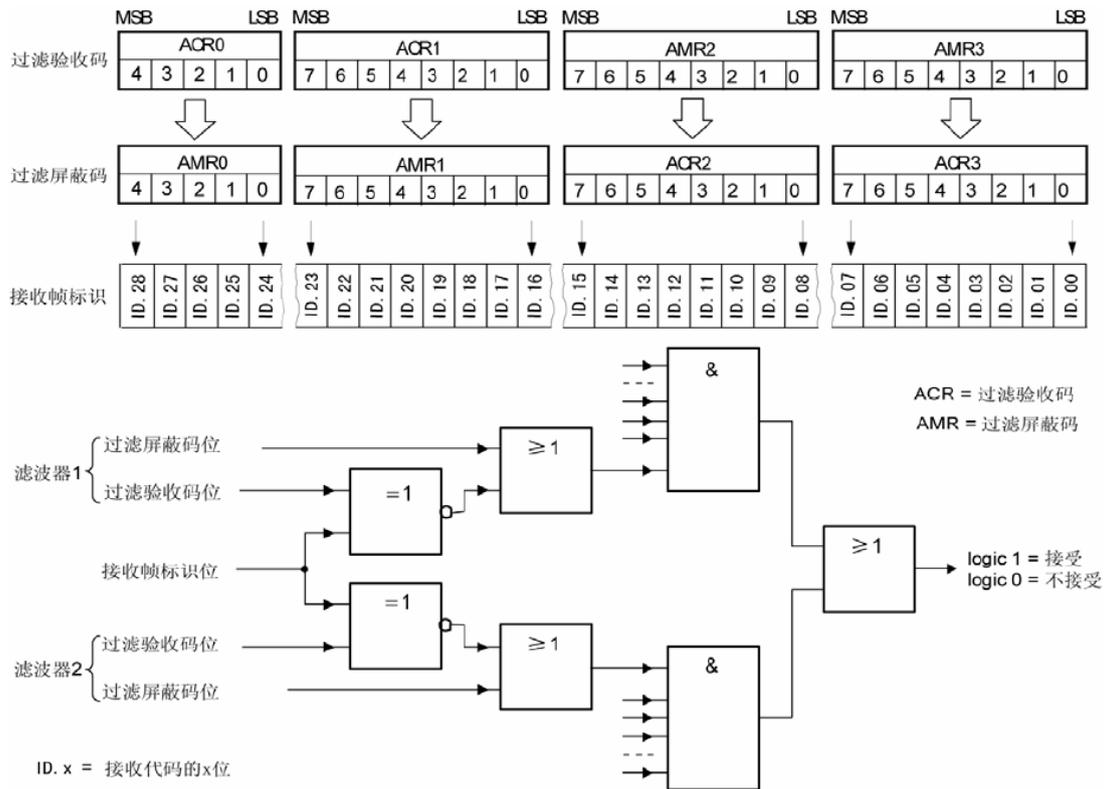


图 2 扩展帧单滤波示意图

## 2. 双滤波配置

这种配置可以定义两个短滤波器。一条接收的信息要和两个滤波器比较来决定是否放入接收缓冲器中。至少有一个滤波器发出接受信号，接收的信息才有效。滤波器字节和信息字节之间位的对应关系取决于当前接收的帧格式。

**标准帧：**对于标准帧，那么则相当于有两个单滤波情况下的滤波器对接收帧标识进行滤波。接收逻辑如图 3 所示。

为了能成功接收信息，一组滤波器的单个位的比较时均要表示为接收。

两组滤波器至少有一组表示接收该帧才会被接收。

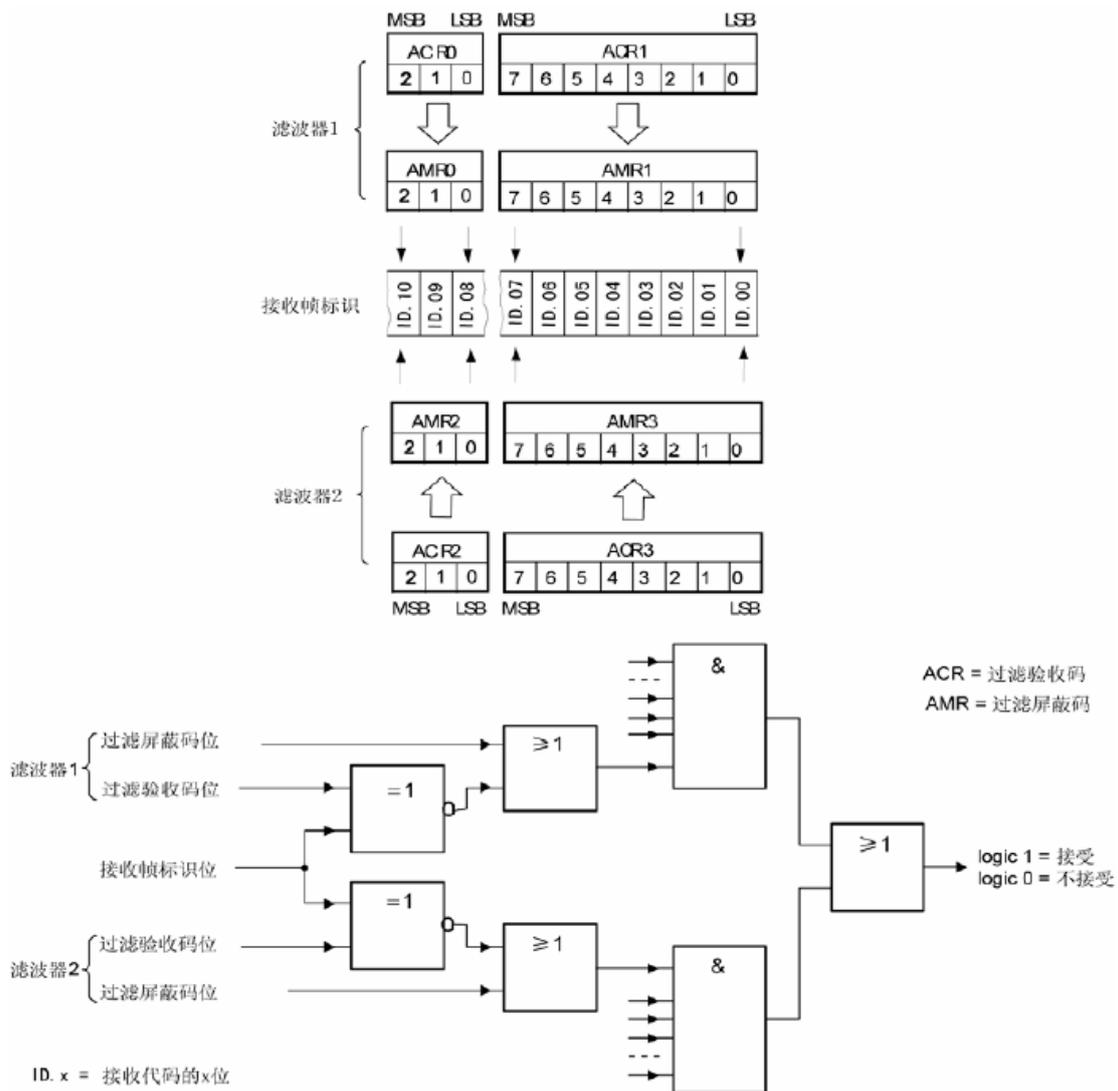


图 3 标准帧双滤波示意图

**扩展帧：**对于扩展帧，定义的两个滤波器是相同的。两个滤波器都只比较扩展识别码的前两个字节——ID. 28到ID. 13，而不是全部的29位标识。如图 4 所示。为了能成功接收信息，一组滤波器的单个位的比较时均要表示为接收。

两组滤波器至少有一组表示接收该帧才会被接收。

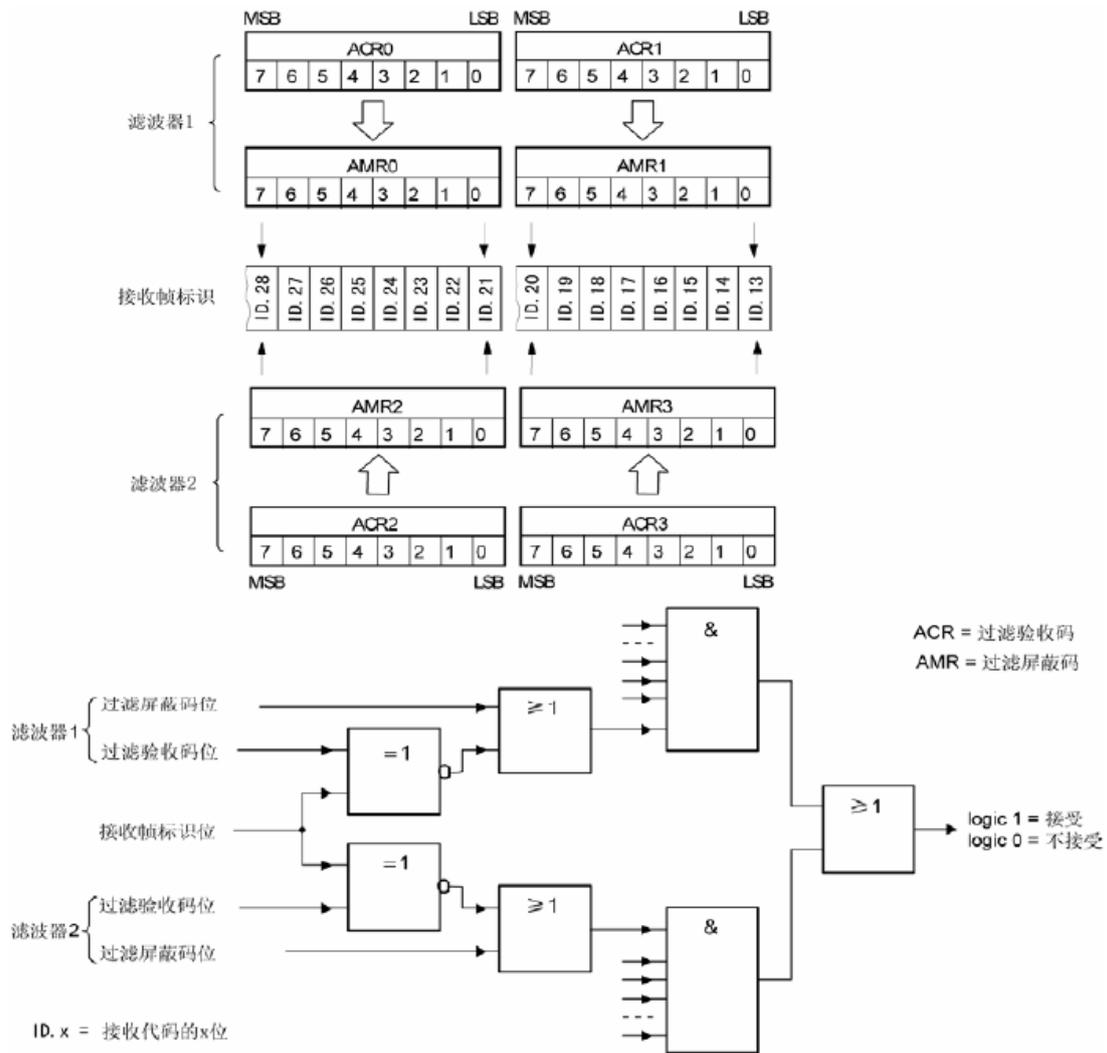


图 4 扩展帧双滤波示意图

附录 4 CAN 总线通信距离 (参考值)

波特率 (kbps)	最大通信距离 (m)
1000	40
500	130
250	270
125	530
100	620
50	1300
20	3300
10	6700
5	10000

附录 5 销售清单

序号	名称	数量	单位	说明
1	PCICAN-9820 接口卡	1	套	
2	配套光盘	1	套	包括使用手册、设备驱动、 函数库等