

HK—CAN 10S

智能 PC 总线通讯板

(PC ISA 总线)

目 录

一、概述	1
二、性能及技术指标	1
2.1 性能	1
2.2 应用	2
2.3 物理尺寸和工作环境条件	2
三、工作原理	2
3.1 工作原理框图	2
3.2 工作原理概述	2
3.2.1 HK-CAN10S 板控制部分	3
3.2.2 ISA 总线接口部分	3
3.2.3 CAN 通讯部分	3
四、主要元件位置图、信号输入/输出插座和开关选择定义	3
4.1 主要元件位置图	3
4.2 信号输入/输出插座定义	4
4.3 开关及跳线选择	4
4.3.1 板基地址选择	4
4.3.2 中断级别选择	5
4.3.3 通讯口的定义	6
五、安装、拆除方法	6
5.1 安装步骤	7
5.2 拆除步骤	7
六、使用中出现问题解决	7
6.1 HK-CAN10S 板不能通讯	7
6.2 HK-CAN10S 板通讯失败次数多	8
6.3 HK-CAN10S 板的自测试	8
七、应用注意事项	8
7.1 注意事项	8
八、驱动程序	9
8.1 结构体定义	错误! 未定义书签。

8.1.1 CAN 帧数据结构:	错误! 未定义书签。
8.1.2 设备控制参数结构:	错误! 未定义书签。
8. 2 驱动函数说明.....	错误! 未定义书签。
8. 2. 1 HKCanOpen ()	错误! 未定义书签。
8. 2. 2 HKCanClose ()	错误! 未定义书签。
8. 2. 3 HKCanGetBDVersion ()	错误! 未定义书签。
8. 2. 4 HKCanInitState ()	错误! 未定义书签。
8. 2. 5 HKCanSendFrame ()	错误! 未定义书签。
8. 2. 6 HKCanReadFrame ()	错误! 未定义书签。
8. 2. 7 HKCanReadFrameEx ()	错误! 未定义书签。
8. 2. 8 HKCanGetLastError ()	错误! 未定义书签。
8. 3 CAN 信息包格式说明.....	错误! 未定义书签。
九 驱动的安装.....	9

HK—CAN10S 智能 PC 总线通讯板

一、概述

北京华控技术有限责任公司的 HK—CAN10S 智能 PC 总线通讯板，是一种将 CAN (Control Area Network) 通讯协议与 PC 机 ISA 总线标准相连接的智能 PC 插卡，智能 CAN 卡可以充分满足 CAN 网络高实时性的要求，通过该卡可对工业现场具有 CAN 通讯接口的仪表和控制设备进行监控。

CAN 是一种全数字化的通讯网络，在国外，已经安装了大量具有 CAN 通讯接口的传感器、执行器、马达等工业设备；在国内，许多具有 CAN 接口的工控设备也逐渐得到广大用户的认可。CAN 最大的优点在于它的实时性很高，由于它采用位仲裁方式进行网络分配，因此可以最大限度的保证系统对紧急事件的响应；CAN 另外一个优点是具有很高的可靠性，它有五种方法判断和纠正数据在传输中可能发生的错误，所以 CAN 可以应用在对可靠性要求较高的系统中。

二、性能及技术指标

2.1 性能

- ISA 总线数据宽度：8 位
- 通讯符合 ISO11898 2.0A 标准
- ISA 总线中断设定范围：IRQ2, IRQ3, IRQ4, IRQ5, IRQ7
- HK—CAN10S 板控制 CPU: Intel 80C188EC
- CAN 网络通讯最高速率：1Mbit/s
- CAN 网络接口控制器：Philips SJA1000T 或兼容芯片
- CAN 网络收发器：Philips 82C250 或兼容芯片
- CAN 网络连接器：DB-9，针式。
- 隔离耐压：1000V_{DC}
- 隔离方式：CAN 通讯口之间，CAN 通讯口与 PC 机之间全隔离。

※ 驱动程序名称定义：

- 虚拟设备驱动(VXD) 应用于 Windows95/98
- 内核式驱动(SYS) 应用于 Windows 98/NT/2000
- DOS 驱动程序: 适用于 DOS 环境

2.2 应用

- 各种集散式控制系统
- 要求抗干扰能力强的通讯网络中

2.3 物理尺寸和工作环境条件

- 外形尺寸: 173mm × 119mm
- 工作温度范围: 0℃ ~ +50℃
- 贮藏温度范围: -25℃ ~ +85℃
- 湿度范围: 90% (不结露)
- HK-CAN10S 板功耗 (典型值): +5V、490mA

三、工作原理

3.1 工作原理框图

图 3.1 为 HK-CAN10S 板的工作原理框图。

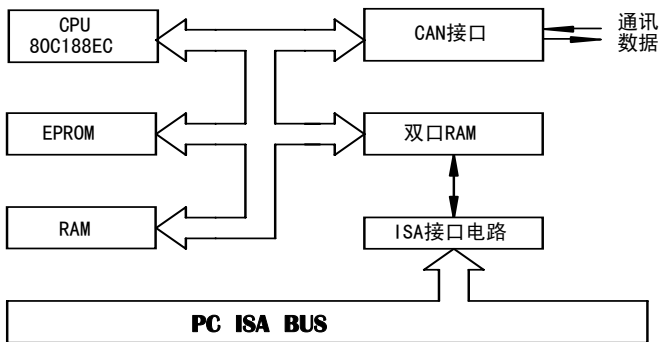


图 3.1 HK-CAN10S 板工作原理框图

3.2 工作原理概述

从功能上, HK-CAN10S 智能 PC 总线通讯板可分为三个部分: HK-CAN10S 板控制部分、ISA 总线接口部分和 CAN 网络通讯部分。

3.2.1 HK-CAN10S 板控制部分

Intel 公司的 80C188EC 微处理器及其周围的静态存储器、程序存储器和双口 RAM 构成 HK-CAN10S 智能 PC 总线通讯板控制电路的核心。

HK-CAN10S 板内部控制电路的主要功能，是对 CAN 通讯接口和 ISA 总线接口进行控制，从而实现 PC 机的 ISA 总线与 CAN 之间的数据交换，换句话说，HK-CAN10S 板控制电路是 ISA 总线与 CAN 之间的桥梁。

3.2.2 ISA 总线接口部分

ISA 总线接口部分是 HK-CAN10S 板和 PC 机之间交换数据的桥梁，HK-CAN10S 板和 PC 机之间的数据交换是通过 ISA 总线电路中的双口 RAM 实现的。

3.2.3 CAN 通讯部分

该部分实现了 CAN 物理层和数据链路层协议，这部分功能受 HK-CAN10S 板内的控制电路控制。

四、主要元件位置图、信号输入/输出插座和开关选择定义

4.1 主要元件位置图

图 4.1 为 HK-CAN10S 智能 PC 总线通讯板的主要元件位置图，此元件位置图上的开关和跳线设置为出厂标准设置。

设置为：板基地址=640，中断为 IRQ7，使用匹配电阻。

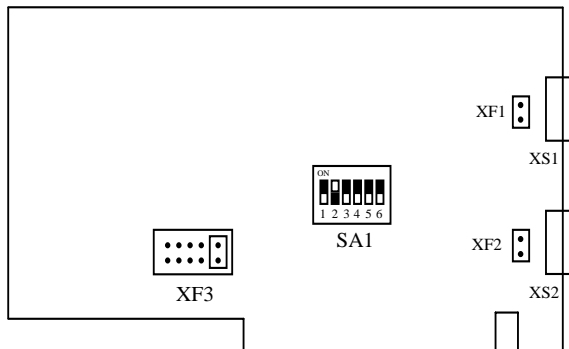


图 4.1 HK-CAN10S 板的主要元件位置图

- XS1: CAN 总线通讯端口 1
 XS2: CAN 总线通讯端口 2
 XF1: 通讯端口 1 匹配电阻的选择跳线
 XF2: 通讯端口 2 匹配电阻的选择跳线
 SA1: 板基地址选择开关
 XF3: 中断级别选择跳线

4.2 信号输入/输出插座定义

图 4.2 为 HK-CAN10S 板信号输入/输出插座定义图。

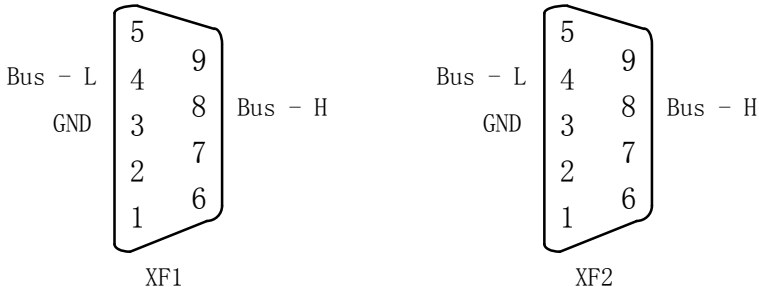


图 4.2 HK-CAN10S 板信号输入/输出插座定义

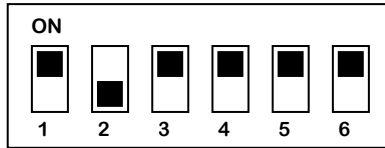
- Bus - H: CAN 通讯信号高电平
 Bus - L: CAN 通讯信号低电平
 GND: 地 (可以不使用)

4.3 开关及跳线选择

4.3.1 板基地址选择

HK-CAN10S 智能 PC 总线通讯板基地址通过对 SA1 开关的设置来选择, 板基地址可设置成 200H~3F8H 任何二进制码的组合, HK-CAN10S 板将占用从基地址起的连续 8 个 I/O 地址。

板基地址选择开关 SA1 示意如图 4.3.1。



地址线:	A8	A7	A6	A5	A4	A3
十进制:	256	128	64	32	16	08
十六进制:	100	80	40	20	10	08

图 4.3.1 HK—CAN10S 板基地址选择

板基地址范围 512~1023 (0200H~03F8H)，开关置“OFF”有效，置“ON”无效。

基地址计算公式：

基地址=512 (0200H) + 所有有效位之和。

例如：图 4.3.1 开关设置板基地址计算如下。

$$\begin{aligned} \text{板基地址} &= 512 + 128 = 640 \\ &= 0200\text{H} + 80\text{H} = 0280\text{H} \end{aligned}$$

4.3.2 中断级别选择

在设定 HK—CAN10S 板的中断请求跳线以前，为了防止发生中断占用冲突，请检查您的 PC 机中现有硬件设备占用中断的情况，HK—CAN10S 板可以使用的中断请求为 IRQ2、IRQ3、IRQ4、IRQ5、IRQ7，测试程序和出厂检验时 HK—CAN10S 板中断请求设定为 IRQ7。

中断的设定方法：

图 4.3.2 为中断跳线器 XF3 的定义。用户在相应的位置上跳线即可，当 XF3 跳线不插时，则无中断级别选中。

例如：选择 IRQ7



图 4.3.2 中断跳线器的定义

4.3.3 通讯口的定义

通讯端口采用 DB-9 针式插座，用户可采用 DB-9 孔式插头与之配合连接，DB-9 插座的引脚定义如下：

通讯口 1 的 DB-9 针式插座定义	通讯口 2 的 DB-9 针式插座定义
PinSignal	PinSignal
8 Bus-H	8 Bus-H
4 Bus-L	4 Bus-L
3 GND	3 GND

CAN 是一种半双工通讯协议，所以用户只需一对导线就可实现数据的发送和接收，但这一对导线是有极性的，如果网络上有多台设备连接，请将它们的 Bus-H 连接在一起，Bus-L 连接在一起（参见图 4.3.3）。

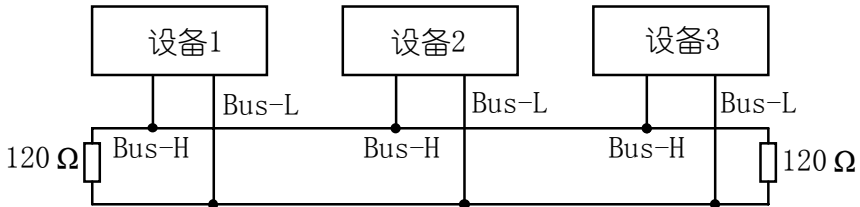


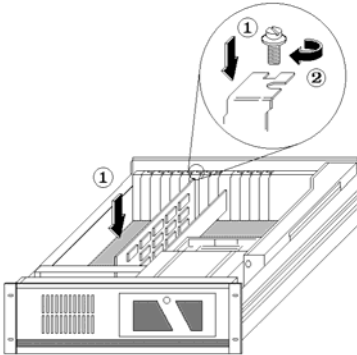
图 4.3.3

由于 CAN 是一种全数字的通讯，信号在通讯电缆传送过程中存在反射，因此，在配置 CAN 时要加匹配电阻（参见图 4.3.3），电阻值为 $120\ \Omega$ 。本卡上已经安装了匹配电阻，用户可根据现场实际需要，通过跳线器 XF1（通讯端口 1）、XF2（通讯端口 2）选择是否要匹配电阻。（跳线器插上为选择要匹配电阻，出厂时选择要匹配电阻）

五、安装、拆除方法

HK-CAN10S 板禁止带电插拔。接插和拔下引线插头，开关的设置及跳线器选择都必须在断电的情况下进行。

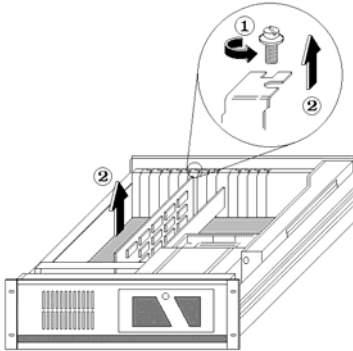
5.1 安装步骤



① 把模板垂直对准任一扩展槽，以手向下用力将该板插入扩展槽。

② 拧紧螺钉，连接输入/输出电缆。

5.2 拆除步骤



① 断开输入/输出信号电缆，拧下螺钉。

② 双手向上垂直用力，将该板从扩展槽中拔出。

六、使用中出现问题解决

HK-CAN10S 板在出厂前都是经过检验的，确保交到用户手中产品符合设计指标要求。但用户的使用环境千差万别，难免在使用过程中出现一些问题，下面是一些常见问题的解决方法。

6.1 HK-CAN10S 板不能通讯

HK-CAN10S 板不能通讯是一个复杂的问题，这即可能硬件配置问题，也可能是软件运行环境配置问题。从 HK-CAN10S 板的结构可以知道，实现一个完整的数据发送过程是将数据由 PC 机的 ISA 总线送到 HK

—CAN10S 板的双口 RAM 中，再由 HK—CAN10S 板上的微处理器将双口 RAM 中的数据送到 CAN 收发器中发送出去；接收过程也是一样的，只是这一过程是逆向的。

因为用户只根据驱动程序编写应用程序，因此问题一般出现在 PC 机 ISA 总线与 HK—CAN10S 板上双口 RAM 的数据交换上，即用户不能与 HK—CAN10S 板进行正确的数据交换。用户可采取如下措施：

① 检查 PC 机的硬件设备配置情况，特别是硬件设备的中断使用情况，如果 HK—CAN10S 板的中断和现有设备的中断冲突，请改变 HK—CAN10S 板的中断号。

② 如果 PC 机中的 CONFIG.SYS 文件加载了 EMM386.EXE 扩展内存管理程序，请在加载语句后添加 X=m000—mFFF，其中 m 可取值 D、E 和 F。（Windows NT 下没有作用）

6.2 HK—CAN10S 板通讯失败次数多

HK—CAN10S 板通讯失败次数多可通过如下措施解决。

- ① 使用 HK—CAN10S 板上的匹配电阻。
- ② 选择合适的导线。
- ③ 根据 CAN 网络协议，检查网络结构是否合理。

6.3 HK—CAN10S 板的自测试

如果用户通过上述努力仍不能使用 HK—CAN10S 板，请使用该板自测试程序，测试方法为：

- ① 将 HK—CAN10S 板的基地址地址设为 280H，中断设为 IRQ7；
- ② 将 HK—CAN10S 板的通讯口 1、通讯口 2 用测试电缆连接；
- ③ 运行测试程序。

七、应用注意事项

7.1 注意事项

在公司售出的产品包装中，用户将会找 HK—CAN10S 智能 PC 总线通讯板一块，通讯测试电缆一条，HK—CAN10S 板驱动程序及测试程序（光盘）一套。同时还有产品质保卡。产品质保卡请用户务必妥善保存，当该产品出现问题需要维修时，请用户将产品质保卡同产品一起，寄回本公司，以便我们能尽快的为您解决问题。

在使用 HK-CAN10S 智能 PC 总线通讯板时，应注意以下问题：

HK-CAN10S 智能 PC 总线通讯板正面 IC 芯片不要用手去摸，防止芯片受到静电的危害。

八、驱动程序

为了让用户方便灵活的使用 HK-CAN10S 板，我们为用户编制了 Windows 98/NT/2000/XP 下的驱动。以下是用户接口函数的结构定义说明。

8.1 结构体定义

8.1.1 CAN 帧数据结构：

```
typedef struct _HKCANFRAME {  
    UCHAR    nPort;  
    UCHAR    mFrame[10];  
} HKCANFRAME, *PHKCANFRAME;
```

成员含义：

nPort : 端口号

mFrame[10] : 数据帧，按照 CAN 标准数据帧定义。

8.1.2 设备控制参数结构：

```
typedef struct _HKCANHANDLE {  
    HANDLE    mHandle;  
    DWORD    mGuid;  
    WORD     mClass;  
    WORD     mDev;  
    WORD     mType;  
    WORD     mState;  
} HKCANHANDLE, *PHKCANHANDLE;
```

成员含义：

mHandle : 设备句柄。

mGuid : 设备标志。

mClass : 设备类型。

0	HK-CAN10S 板卡
1	HK-CAN20C 板卡
2	HK-CAN30B 板卡
4	HK-CAN50 模块

mDev : 设备数量。

一部机器内, 最多可插 4 块 CAN10 板卡设备序号取值为 0-3

mType : 驱动类型。

驱动类型有内核式、虚拟设备驱动。内核式驱动用于 windows 98/2000 虚拟设备驱动用于 windows 95 和 windows98。

DRIVER_SYS	0	内核式驱动
DRIVER_VXD	1	虚拟设备驱动

mState : 设备状态。

指示该设备驱动是否已正常工作。1 为正常, 0 为不正常, 可能是硬件参数冲突、端口未初始化、设备未打开等错误。

8.2 驱动函数说明

8.2.1 HKCanOpen ()

函数原型 :

```

BOOL HKCanOpen(
                PHKCANHANDLE mDevHandle,
                char          *InDriverName,
                int           Type,
                int           nDev
                )

```

功能: 打开设备。

返回值:

TRUE 设备打开成功。

FALSE 设备打开错误。

参数:

mDevHandle: 设备状态控制结构体 (OUT)

InDriverName: 设备名称 (IN)

HK-CAN10S 板卡——“HKCAN10”

HK-CAN20C 板卡——“HKCAN20”

HK-CAN30B 板卡——“HKCAN30”

HK-CAN50 模块——“HKCAN50”

nDev : 设备序号 (0 - 3) (IN)

Type : 设备驱动类型 (IN)

DRIVER_SYS 0 内核式驱动

DRIVER_VXD 1 虚拟设备驱动

8.2.2 HKCanClose ()

函数原型:

```
BOOL HKCanClose(  
                PHKCANHANDLE mDevHandle  
                )
```

功能: 关闭设备。

返回值:

TRUE 设备正确关闭。

FALSE 设备关闭错误。

参数:

mDevHandle : 设备状态控制结构体 (IN)

8.2.3 HKCanGetBDVersion ()

函数原型:

```
BOOL HKCanGetBDVersion(  
                PHKCANHANDLE mDevHandle,  
                char *sVersion,  
                int buflength  
                )
```

功能: 取得当前驱动版本号。

返回值:

TRUE 函数操作正确

FALSE 函数操作错误

参数:

mDevHandle: 设备状态控制结构体 (IN)

sVersion: 返回设备驱动版本字符串指针 (长度不小于 2 字节) (OUT)

buflen: 字符缓冲区长度 (小于 16 会导致调用错误) (IN)

8.2.4 HKCanInitState ()

函数原型:

```

BOOL HKCanInitState(
                                PHKCANHANDLE mDevHandle,
                                Int nPort,
                                UINT CAN_bps,
                                UCHAR CAN_StationAddress,
                                UCHAR CAN_Mask,
                                HANDLE mRxEvent
                                );

```

功能: 初始化设备端口

返回值:

TRUE 初始化设备端口成功。

FALSE 初始化设备端口失败。

参数:

mDevHandle: 设备状态控制结构体 (IN)

nPort: 设备端口号 (取值为 0、1) (IN)

CAN_bps : HK-CAN10S 通讯板的波特率。现有五档波特率, 请参见波特率设置表。如果用户欲使用其它波特率, 请参照有关 CAN 总线原理的资料自行计算。

波特率	CAN_bps
1M	0xCOA3

500K	0xC1A3
250K	0xC3A3
125K	0xC7A3
50K	0xC7AF

CAN_StationAddress: 为本站的站地址 (IN)

CAN_Mask: 为通讯板的接收屏蔽字, 与 CAN_StationAddress 共同作用决定本站可接收信息包 (IN)。

判定公式如下:

$$ID \mid \text{CAN_MASK} = \text{CAN_MASK} \mid \text{CAN_StationAddress}$$

其中: ID 为信息包标识符的高 8 位。

例如: 当 CAN_MASK=0xFF 时, 则接收网络上的所有信息包, 当 CAN_MASK=0 时, 则只接收 ID 与 CAN_StationAddress 相等的信息包。

中断屏蔽字设置实例:

假设 CAN 总线上现共有三个站点, 站地址分别设为: 10、12、13, 其中地址为 10 的站点需要接收标识为 10 和 11 的信息包, 地址为 12 的站点只关心标识为 12 的信息包, 地址为 13 的站点欲接收所有的信息包。因为 10 和 11 只在最低位不同 (00001010 和 00001011), 所以站点 10 的中断屏蔽字应设置为 00000001, 既 0x01; 站点 12 只关心与本身有关的信息包, 其中断屏蔽字应为 0x00; 而站点 13 则应设置为 0xFF。

mRxEvent : 数据帧到达核心对象句柄, 一般为事件。如果采用查询方式读取数据, 可将此参数设为 NULL (IN)。

8.2.5 HKCanSendFrame ()

函数原型:

```
int HKCanSendFrame (
                                PHKCANHANDLE mDevHandle,
                                PHKCANFRAME pSendFrame
                                )
```

功能: 发送一帧数据

返回值:

- 0 设备正确发送数据
- 5 设备未初始化
- 6 设备发送忙
- 9 事件句柄为空, DLL 中事件初始化失败
- 13 设备发送超时
- 106 设备正在发送中

参数:

- mDevHandle: 设备状态控制结构体 (IN)
- pSendFrame: 发送数据帧指针 (IN)

8.2.6 HKCanReadFrame ()

函数原型:

```
int HKCanReadFrame(  
                    PHKCANHANDLE mDevHandle,  
                    Int           nPort,  
                    PHKCANFRAME  pReadFrame  
                    )
```

功能: 读取一帧数据

返回值:

- >=0: 该值为缓冲区中剩余帧数 (等于 0 表示缓冲区已读空)。
- <0: 读取操作失败

参数:

- mDevHandle: 设备状态控制结构体 (IN)
- nPort: CAN 端口号 (IN)
- pReadFrame: 接收数据帧指针 (OUT)

8.2.7 HKCanReadFrameEx ()

函数原型:

```
int HKCanReadFrameEx(  
                    PHKCANHANDLE mDevHandle,  
                    int           nPort,  
                    PHKCANFRAME  pReadFrame,  
                    int           *pReadnum
```

)

功能：读取多个数据帧

返回值：

>=0：该值为缓冲区中剩余帧数（等于 0 表示缓冲区已读空）。

<0：读取操作失败

参数：

mDevHandle：设备状态控制结构体（IN）

nPort：CAN 端口号（IN）

pReadFrame：接收数据帧指针（大小应足够容纳指定帧数
sizeof(PHKCANFRAME) * pReadnum）（OUT）

pReadnum：读取帧数（IN）

8.2.8 HKCanAbortSend（）

函数原型：

```

BOOL HKCanAbortSend(
    PHKCANHANDLE mDevHandle,
    int nPort)

```

功能：停止发送

返回值：

TRUE 函数执行正确，

FALSE 函数执行错误。

参数：

mDevHandle：设备状态控制结构体（IN）

nPort：CAN 端口号（IN）

8.2.9 HKCanGetLastError（）

函数原型：

```

BOOL HKCanGetLastError(
    PHKCANHANDLE mDevHandle,
    PULONG pError)

```

功能：获取当前设备状态

返回值：

TRUE 函数执行正确，

FALSE 函数执行错误。

参数:

mDevHandle: 设备状态控制结构体 (IN)

pError : 返回数据为设备端口状态 (OUT)

d0-d7: 核心对象状态 (一般为事件)

d8-d15: 设备错误状态

d16-d23: 端口二当前接收数据帧数 (未读取, 在驱动缓冲区内)

d24-d31: 端口一当前接收数据帧数 (未读取, 在驱动缓冲区内)

核心对象状态 (d0-d7):

HKCAN_EVENT_ERROR	1	设备操作失败
HKCAN_EVENT_SEND	2	设备发送完成
HKCAN_EVENT_READ	4	接收到数据帧
HKCAN_EVENT_INIT	8	端口初始化完成

设备错误状态 (d8-d15):

	0	为设备操作正确
INIT_SUCCESS	0x01	//初始化成功
INIT_ERROR	0x02	//初始化错误
RECEIVE_QUE_OVERFLOW	0x03	//接收数据区溢出
SENDFRAME_FAILED	0x04	//发送出错 (失败)
SEND_QUE_OVERFLOW	0x05	//发送数据区溢出
DEV_INT_ERROR	0x06	//CAN 设备出错中断
UNINIT_ERROR	0x07	//CAN 没有初始化
UNEXIST_BOARD_ERROR	0x08	//CAN10 板不存在
RECEIVE_ERROR	0x09	//接收出错 (失败)
RECEIVE_QUE_EMPTY	0x0a	//接收数据区空

8.3 CAN 信息包格式说明

CAN 信息包分为两部分：

信息部分和数据部分。头两个字节为信息部分，其前十一位为标识符。然后是一位 RTR 位（应设为 0），最后是四位的 DLC（数据长度位，即所发数据的实际长度，单位：字节）。其余八个字节是数据部分，存有实际要发的数据。详见右表。

字节1

字节2

字节3

字节4

字节5

字节6

字节7

字节8

字节9

字节10

7	6	5	4	3	2	1	0
标识符（高八位）							
标识符		RTR		DLC			
数 据							
数 据							
数 据							
数 据							
数 据							
数 据							
数 据							

九 驱动的安装

HK-CAN10S 通讯卡如想正常工作，必须正确安装驱动程序。本卡的驱动有两种类型：内核式驱动（SYS）、虚拟设备驱动（VXD）。

内核式驱动适用于 Win95/Win98/NT/2000。虚拟设备驱动只适用于 Win95/98。用户可根据需要自己选择驱动的种类。需要注意的是，如果在同一计算机中安装有多块华控 CAN 系列通讯卡，驱动类型必须统一，即全部安装成 SYS 驱动或全部安装成 VXD 驱动。

安装步骤：

1. 在随卡光盘 SetupControl 目录下找到 Setup.exe，双击运行，安装程序将在控制面板中生成华控驱动控制，进入控制面板双击运行。
2. 点击相应的单选按钮选择驱动类型，然后在左边列表框选中欲安装的设备（HKCAN10），单击“驱动安装”按钮，在弹出的打开对话框中找到光盘\DRIVERS\HKCAN10S 目录下相应驱动类型文件夹，选择驱动文件（例如：选择了 SYS 驱动，则应该选择 SYS 文件夹中的 HKCAN10.Sys 文件）。然后单击打开，界面应如图 9.1 所示：
3. 设置参数。通过双击参数值，选择与板卡设置一致的基地址、中断资源。

4. 如果同一系统中安装多块同种板卡，可以单击添加设备按钮，然后设置各参数即可（板卡各参数不能相同）。在使用时只需注意设备号即可。



图 9.1

5. 如果有不需要的设备，选中该设备后单击删除设备即可。
6. 安装完成后退出程序，然后重新启动计算机。

注：DOS、WINDOWS 是美国 Microsoft 公司的注册商标