

HK-CAN40A PC/104 总线隔离型 CAN 通讯模块

一、概述

HK-CAN40A 非智能隔离型 CAN 总线通讯模块，是一种采用 PC/104 总线标准的隔离型 CAN 通讯模块，该模块采用的栈接总线结构符合 IEEE P961.1 标准，可方便的为具有 PC/104 总线接口的嵌入式系统扩充 CAN 通讯功能。由于 CAN 总线通讯协议简单，工作可靠性高，通讯距离长，编程方便，系统造价低廉等特点，采用该标准的 HK-CAN40 模块特别适合用于构成分布式测控系统及现场总线自动化仪器仪表系统。

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 性能与特点

- 通讯协议符合 CAN 2.0A/B (ISO/DIS 11898)国际标准
- PC/104 总线标准在电气指标方面与 IBM PC 总线 100% 兼容，其工作原理易于理解，便于使用
- 采用了 CAN 总线高速光电隔离技术，使 CAN 网络系统中各节点间实现了电气隔离，提高了整个网络系统的通讯可靠性
- CAN 通讯接口使用 9 针 D 型连接器，网络连接方便可靠
- 可支持 DeviceNet, CANopen 应用层协议

• 备有在 DOS、Windows、Windows 95/98、操作系统环境下运行的驱动程序及例程

2.2 技术指标

- PC/104 总线数据宽度：8 位
- PC/104 总线中断设定范围：IRQ3，IRQ4，IRQ5，IRQ6，IRQ7，IRQ9，IRQ10，IRQ11，IRQ14，IRQ15
- CAN 网络通讯最高速率：1Mbit/s
- CAN 网络接口控制器：Philips SJA1000T
- CAN 网络收发器：Philips 82C250
- CAN 网络连接器：DB-9，孔式。
- DOS 驱动程序
- Windows 驱动程序
- Windows 95/98 驱动程序
- 隔离耐压：1000V_{DC}

2.3 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

2.4 物理尺寸和工作环境条件

- 外形尺寸：90mm×96mm
- 工作温度范围：0℃～+60℃
- 贮藏温度范围：-25℃～+85℃
- 湿度范围：90%（不结露）
- HK-CAN40A 模块功耗（典型值）：+5V、200mA

三、工作原理

3.1 工作原理概述

从功能上，HK-CAN40A 非智能隔离型 CAN 总线通讯模块可分为二个部分：PC/104 总线接口部分和 CAN 网络通讯部分。

3.2.1 PC/104 总线接口部分

PC/104 总线接口部分是 HK-CAN40A 模块和 PC/104 总线 CPU 模块之间交换数据的桥梁，HK-CAN40A 模块和 PC/104 总线 CPU 之间的数据交换是通过 PC/104 总线电路实现的。接口采用的 PC/104 总线数据宽度为 8 位。

3.2.2 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-CAN40A 模块内的控制电路控制。

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-CAN40A 非智能隔离型 CAN 总线通讯模块的主要元件位置图，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：模块基地址=640，中断为 IRQ11。

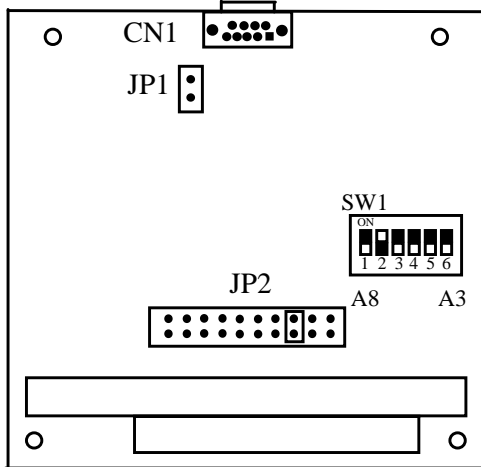


图 4.1 HK-CAN40A 模块的主要元件位置图

CN1: CAN 通讯端口

JP1: 通讯端口 1 匹配电阻的选择跳线

SW1: 模块基地址选择开关

JP2: 中断级别选择跳线

4.2 信号输入/输出插座定义

4.2.1 HK-CAN40A信号输入/输出插座定义

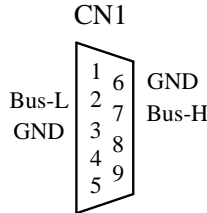


图 4.2.1 为 HK-CAN40A 模块信号输入/输出插座定义图。

CN1 信号定义说明:

Bus - H: CAN 通讯信号高端

Bus - L: CAN 通讯信号低端

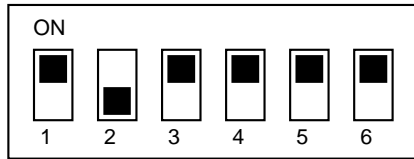
GND: 地

4.3 开关及跳线选择

4.3.1 模块基地址选择

HK-CAN40A 通讯模块基地址通过对 SW1 开关的设置来选择, 模块基地址可设置成 200H~3F8H 任何二进制码的组合, HK-CAN40A 模块将占用从基地址起的连续 8 个 I/O 地址。

模块基地址选择开关示意如图 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-CAN40A 模块基地址选择

模块基地址范围 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 开关设置模块基地址计算如下：

模块基地址=512+128=640

或=0200H+80H=0280H

4.3.2 中断级别选择

在设定 HK-CAN40A 模块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/104 系统中现有硬件设备占用中断的情况，HK-CAN40A 模块可以使用的中断请求为 IRQ3、IRQ4、IRQ5、IRQ6、IRQ7、IRQ9、IRQ10、IRQ11、IRQ14、IRQ15，测试程序和出厂检验时 HK-CAN40A 模块中断请求设定为 IRQ11。

中断的设定方法：

图 4.3.2 为中断跳线器 JP2 的定义。用户在相应的位置上跳线即可，当 JP2 跳线不插时，则无中断级别选中。

例如：选择 IRQ11

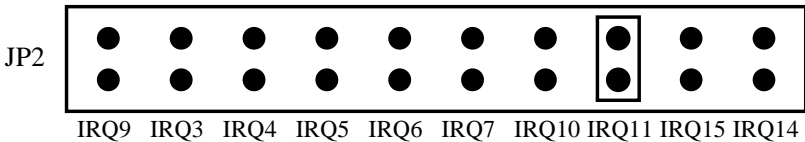


图 4.3.2 中断跳线器的定义

4.3.3 通讯口的设置

通讯端口采用 DB-9 孔式插座，用户可采用 DB-9 针式插头与之配合连接，DB-9 插座的引脚定义参见 4.2 节

使用 HK-CAN40A 模块时，两根通信线分别接 CN1 口的 BUS_H 和 BUS_L。

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）。

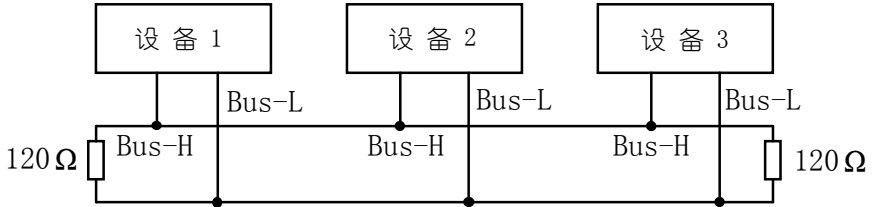


图 4.3.3 多台设备连接示意图

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 JP1 选择是否要匹配电阻。（跳线器插上为选择要匹配电阻）

五、安装、拆除方法

HK-CAN40A 模块禁止带电插拔。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 把模块安装到 PC/104 系统中时，把 PC 扩展总线连接器的插针垂直插入到总线连接器的插座里。要确保每根插针都插入对应孔里。
- ② 模块间用提供的螺柱固定。

5.2 拆除步骤

- ① 松开模块间用提供的螺柱。
- ② 双手向上垂直用力，把模块从 PC/104 系统总线连接器的插座中拔出。

六、使用中出现问题解决

HK-CAN40A 模块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-CAN40A 模块不能通讯

HK-CAN40A 模块不能通讯是一个复杂的问题，这既可能是硬件配置问题，也可能是软件运行环境配置问题。用户可采取如下措施：

- ① 检查地址开关的设置是否和其它设备冲突；
- ② 检查 HK-CAN40A 模块的中断设置是否和现有设备的中断冲突，请改变 HK-CAN40A 模块的中断号。

6.2 HK-CAN40A 模块通讯失败次数多

HK-CAN40A 模块通讯失败次数多可通过如下措施解决。

- ① 使用 HK-CAN40A 模块上的匹配电阻；
- 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。

七、应用注意事项

7.1 注意事项

在公司售出的产品包装中，用户将会找到这本说明书和 HK-CAN40A 通讯模块一块，驱动程序盘（光盘）一张。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-CAN40A 通讯模块时，应注意以下问题：

HK-CAN40A 通讯模块正面 IC 芯片不要用手去摸，防止芯片受到静电的危害。

八、驱动软件

为了让用户方便灵活的使用 HK-CAN40A 模块，我们为用户编制了 DOS 驱动程序、和 WINDOWS 98/NT/2000/XP 驱动程序。HK-CAN40 的驱动采用的是 HK-CAN20 的驱动，因此具体使用方法请参见 HK-CAN20 驱动的使用说明。

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

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           buflen  
                )
```

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：

$$ID \mid CAN_MASK == CAN_MASK \mid CAN_StationAddress$$

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。

九 驱动的安装

字节1	7	6	5	4	3	2	1	0
字节2	标识符（高八位）							
字节3	标识符		RTR	DLC				
字节4	数 据							
字节5	数 据							
字节6	数 据							
字节7	数 据							
字节8	数 据							
字节9	数 据							
字节10	数 据							

HK-CAN20C 通讯卡如想正常工作，必须正确安装驱动程序。HK-CAN20C 通讯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 (HKCAN20), 单击“驱动安装”按钮, 在弹出的打开对话框中找到光盘\DRIVERS\HKCAN20c 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 HKCAN20.Sys 文件)。然后单击打开, 界面应如图 9.1 所示:



图 9.1

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。
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。

附录：CAN40A模块驱动函数使用说明

一、概述

为了让用户方便灵活的使用 CAN40A 模块，我们为用户编制了 CAN40A 模块驱动函数，用户可以用 C 语言调用 CAN40A 模块的驱动函数。

用户的驱动程序磁盘中，有在 DOS 和 WINDOWS 下驱动函数如何调用的例程，请参考。

希望使用 CAN40A 模块驱动函数的用户请向本公司购买。

二、DOS下的通讯驱动程序说明

DOS 下的驱动程序共有五个函数：

- InstallCANDriver()
- UninstallCANDriver()
- SendCANFrame()
- ReadCANFrame()
- ClearError()

2.1 InstallCANDriver()

```
函数原形：int InstallCANDriver(  
    unsigned int  CAN_BaseAddress,  
    unsigned char CAN_IntNo,  
    unsigned int  CAN_bps,  
    unsigned char CAN_StationAddress,  
    unsigned char CAN_Mask,  
    void (__far *Callback)(int Index)  
);
```

功能：初始化 CAN40A 通讯模块的各个寄存器，设置中断向量和回调函数，为其正常通讯做准备。

参数：

CAN_BaseAddress 为模块基地址，与模块上的地址拨动开关对应。

CAN_IntNo 为 CAN40A 通讯模块的中断号。

计算公式为：CAN_IntNo = IRQ 号 - 8 + 112

CAN_bps 为 CAN40A 通讯模块的波特率。现有五档波特率，请参见表 2.1。如果用户欲使用其它波特率，请参照有关 CAN 总线原理的书籍自行计算。

表 2.1 波特率设置表

波特率	CAN_bps
1M	0xC0A3
500K	0xC1A3
250K	0xC3A3
125K	0xC7A3
50K	0xC7AF

CAN_StationAddress 为本站的站地址。

CAN_Mask 为通讯模块的接收屏蔽字，与 CAN_StationAddress 共同作用决定本站可接收信息包，判定公式如下：

$ID \mid CAN_MASK = CAN_MASK \mid CAN_StationAddress$

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。

Callback 为指向用户编写的回调函数的指针，该回调函数的原形为：void (_far *CallbackFunctionName)(int Index);

其中：

Index=1: 当接收缓冲区已空又有新的信息包到来时, 中断服务程序以该值调用回调函数, 用户应在该处调用 `ReadCANFrame()` 函数接收信息包。(请参见 `ReadCANFrame()` 函数说明)

Index=2: 当接收缓冲区已满时, 中断服务程序以该值调用回调函数, 用户可自行处理。

该指针也可为空, 由用户直接调用 `ReadCANFrame()` 从缓冲队列中读取信息包。

返回值:

- 0 —— 安装成功;
- 1 —— 安装失败, 调用接口的参数不对;
- 2 —— 安装失败, 写 CAN 寄存器失败;

2.2 UninstallCANDriver()

函数原形: `void UninstallCANDriver(void);`

功能: 在程序退出之前释放 CAN 驱动程序所占用的系统资源。

参数: 无。

返回值: 无。

2.3 SendCANFrame()

函数原形: `int SendCANFrame(unsigned char _far *pFrame);`

功能: 为发送 CAN 通讯包。

参数:

`pFrame` 为指向存有要发送的通讯包的缓冲区的远指针, 该缓冲区的长度应为 10 个字节。

返回值: 1 —— 发送成功; 0 —— 发送失败。

2.4 ReadCANFrame()

函数原形: `int ReadCANFrame(unsigned char __far *pFrame);`

功能: 从缓冲队列中接收 CAN 通讯包。

参数:

`pFrame` 为指向接收通讯包的缓冲区的远指针, 该缓冲区的长度应为 10 个字节。

返回值:

> 0 —— 缓冲队列仍有未读出的信息包。

<= 0 —— 缓冲队列读空，可返回。

注意：当该接口在回调函数对 Index=1 的处理代码中调用时，必须反复调用直到接口返回值小于等于零，即将缓冲队列中所有的信息包读出，否则中断服务程序不会再调用回调函数，直到缓冲区满后用 Index=2 再次调用回调函数。该接口也可以单独调用。

2.5 ClearError()

函数原形：int ClearError(void);

功能：将 CAN40A 模块的各个寄存器重新设置，清除某些偶然错误。

参数：无

返回值：

1 —— 除错成功；

0 —— 有严重错误，不能除错。

注：DOS、WINDOWS 是美国 Microsoft 公司的注册商标。

HK-CAN40A

PC/104 总线隔离型 CAN 通讯模块

目 录

一、概述	1
二、性能及技术指标	1
2.1 性能与特点	1
2.2 技术指标	2
2.3 应用	2
2.4 物理尺寸和工作环境条件	2
三、工作原理	2
3.1 工作原理概述	2
3.2.1 PC/104 总线接口部分	3
3.2.2 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
4.2.1 HK-CAN40A 信号输入/输出插座定义	4
4.3 开关及跳线选择	4
4.3.1 模块基地址选择	4
4.3.2 中断级别选择	5
4.3.3 通讯口的设置	5
五、安装、拆除方法	6
5.1 安装步骤	6
5.2 拆除步骤	6
六、使用中出现问题解决	6
6.1 HK-CAN40A 模块不能通讯	7
6.2 HK-CAN40A 模块通讯失败次数多	7
七、应用注意事项	7
7.1 注意事项	7
八、驱动软件	7

8.1 结构体定义	7
8.2 驱动函数说明	9
8.3 CAN 信息包格式说明	15
九 驱动的安装	15
附录：CAN40A 模块驱动函数使用说明	18
一、概述	18
二、DOS 下的通讯驱动程序说明	18
2.1 InstallCANDriver()	18
2.2 UninstallCANDriver()	20
2.3 SendCANFrame()	20
2.4 ReadCANFrame()	20
2.5 ClearError()	21