

基于 CH365 设计 PCI 总线的 CAN 接口卡

1. 概述

CH365 是一款连接PCI 总线的通用接口芯片，应用广泛，主要用于制作低成本的基于PCI 总线的计算机板卡，以及将原先基于ISA 总线的板卡移植到PCI 总线上。

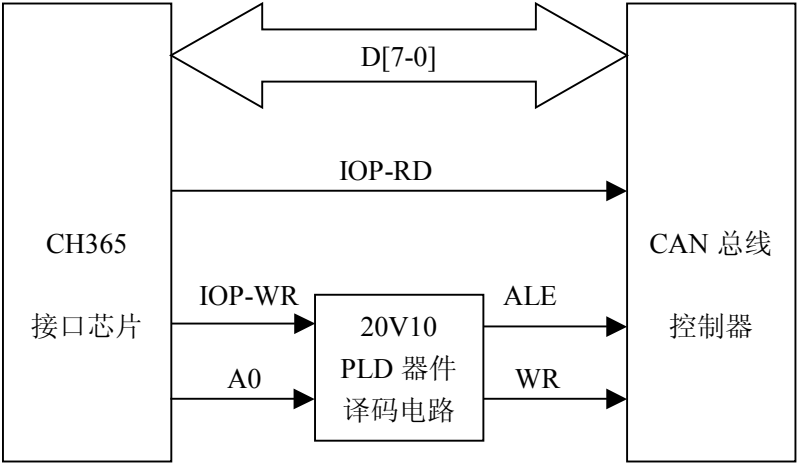
本方案是基于CH365设计PCI总线的CAN 接口卡。

2. 硬件电路设计

2.1 基本电路（也可以采用其它 CAN 芯片代替本设计）

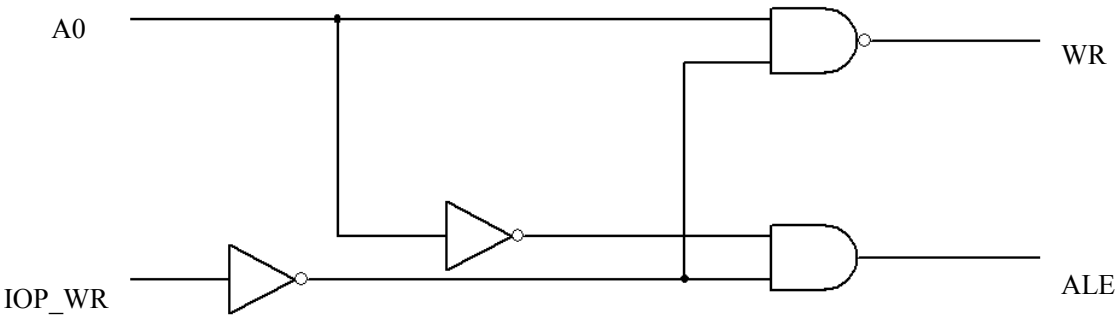
- CAN 控制器采用 PHILIPS 的 SJA1000，工作于 PeliCAN 模式，其它芯片类似。
- CAN 收发器采用 PHILIPS 的 TJA1050，其它芯片类似。
- 采用光耦 6N137 来实现收发器和控制器之间的电气隔离，保护控制系统电路。

简图如下所示（具体电路原理图见文末）



2.2 PLD 译码电路

- 1、CAN 总线控制器的 CS 片选信号由 CH365 的 A1 或者 A7 等控制，或者直接接地；如果有多片 CAN 总线控制器，则各 CS 可以由 CH365 的地址线译码产生。
- 2、CAN 总线控制器的 ALE 和 WR 的译码电路如下：



CH365 的 IOP_WR 通过 PLD22V10 产生 ALE 和 WR 传输给 CAN 控制器。

- $ALE = \overline{SA0} \cdot \overline{IOP_WR}$
- $\overline{WR} = SA0 \cdot \overline{IOP_WR}$

2.3 CH365 中的 D0~D7 信号线复用

CH365 上的 D0~D7 在这里作为地址/数据信号线复用，具体的复用过程为：
当 PLD20V10 中的 A0 为 0，即地址锁存信号 ALE 有效时，CH365 中 D0~D7 作为地址线输出地址。当 PLD22V10 中的 A0 为 1，即数据锁存信号 WR 有效时，CH365 中 D0~D7 作为数据线输出数据。

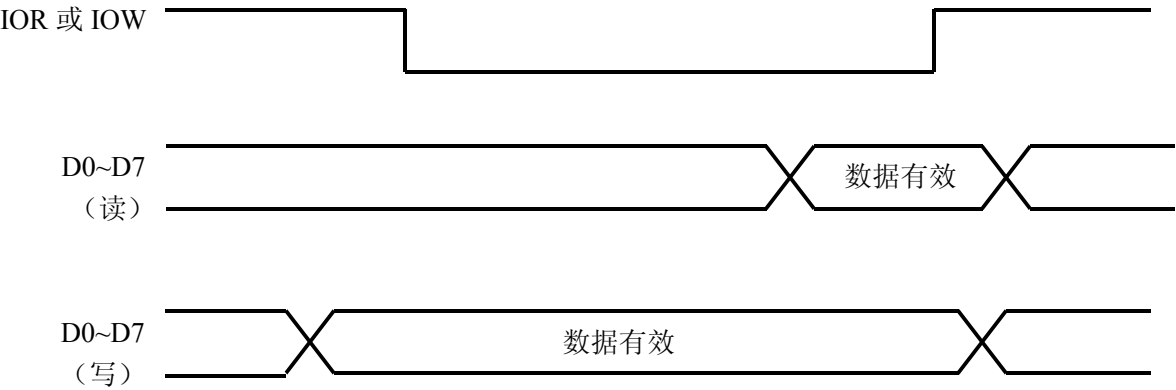
```
//CH365 向 CAN 控制寄存器中（地址为 iAddr 的单元）写字节
CH365mWriteIoByte(iDevice, & m_IoBase->mCh365IoPort[0], iAddr);
CH365mWriteIoByte(iDevice, & m_IoBase->mCh365IoPort[1], iData);
//CH365 向 CAN 控制寄存器中（地址为 iAddr 的单元）读字节
CH365mWriteIoByte(iDevice, & m_IoBase->mCh365IoPort[0], iAddr);
CH365mReadIoByte (iDevice, & m_IoBase->mCh365IoPort[1], iData);
```

CH365 的 8 位 I/O 工作时序如下：

- 输出地址在 CAN 控制器端的时序

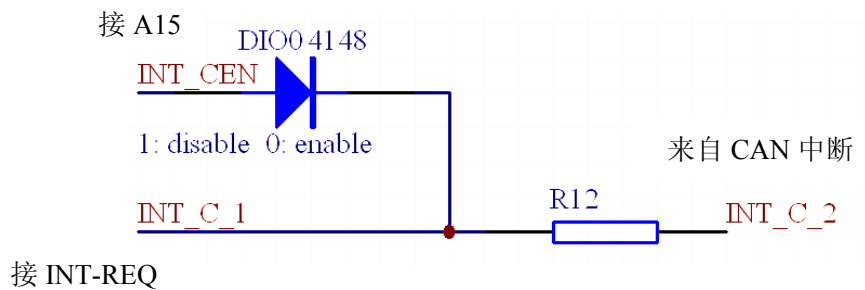


- 读出数据或者写入数据在 CAN 控制器端的时序



2.4 CH365 的中断控制

中断部分的控制电路如下图所示：

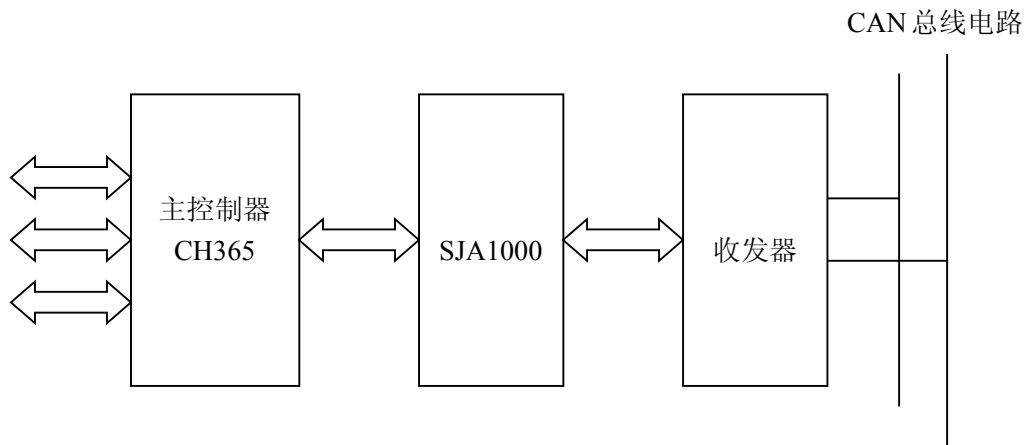


CH365 中断控制电路图

其中，INT_CEN 连接 CH365 的 A15 输出端口，INT_C_2 连接 SJA1000 的中断输出端口，INT_C_1 连接 CH365 中断输入端口，构造成简易或门电路。如果 A15 输出高电平，则 CH365 中断端口保持高电平不变，相当于取消外部电路对 CH365 的中断请求。如果 A15 输出低电平，则 CH365 中断输入端口的电平由 SJA1000 中断输出端口的电平决定，相当于使能外部电路的中断请求。

3. 软件设计

对于图中所示的CH365 与CAN 控制器SJA1000 的接口电路，其控制软件应按模块化设计，一般由以下几个部分组成



下面的接口软件用 C 语言编写的，目的是简洁、可读性好。

所有软件源程序在 CAN 资料光盘中，含 WDM 驱动程序源程序和 DLL 源程序及应用程序的示例。

1. CH365 对 SJA1000 的访问

访问过程分两次 I/O 操作完成，第一次往地址端口送地址，第二次才是对数据端口进行访问。

```
//CH365 向 CAN 配置寄存器中（地址为 iAddr ）写字节
VOID    CH365_CanBus_WriteByte(
    USHORT    iDevice                      ,//设备号
    mPCH365_IO_REG  m_IoBase  ,
```

```

    UCHAR    iAddr          ,//CAN 配置寄存器地址
    UCHAR    iData          ,//发送数据
//CH365 向 CAN 配置寄存器中（地址为 iAddr ）读字节
    UCHAR    CH365_CanBus_ReadByte(
        USHORT    Device          ,//设备号
        mPCH365_IO_REG    m_IoBase ,
        UCHAR    iAddr          ,//数据地址
        UCHAR    * iData          ,//数据存放缓冲
    )

```

2. CH365 初始化

初始化主要包括工作方式的设置、中断线程的设置等。

```

    BOOL CH365_INT_Init (
        PUSHORT    iDevice      ,
        mPCH365_IO_REG    m_IoBase ,
        PULONG     m_IntLine    ,
        mPCH365_INT_ROUTINE    iIntRoutine );

```

3. SJA1000 初始化

初始化主要包括工作方式的设置、接收滤波方式的设置、接收屏蔽寄存器和接收代码寄存器的设置、波特率参数设置和中断允许寄存器的设置等。

```

    BOOL CH365_CanBus_Init(
        USHORT    iDevice      ,
        mPCH365_IO_REG    m_IoBase,
        UCHAR     AMR_Code[]   ,//AMR 掩码
        UCHAR     ACR_Code[]   ,//ACR 识别码
        UCHAR     m_Time1      ,//
        UCHAR     m_Time2      ,//波特率设定
        USHORT     DebugMode    );//调试模式

```

4. CAN 报文发送

CAN 报文发送函数负责 CH365 报文的发送。发送时用户只需将待发送的数据按特定格式组合成一帧报文，送入 CAN 适配卡上的 SJA1000 发送缓冲区中，然后启动 SJA1000 发送即可。

```

    BOOL CH365_CanBus_Send(
        USHORT    iDevice      ,//设备号
        mPCH365_IO_REG    m_IoBase ,//I/O 地址
        UCHAR     iRegion      ,//域名代码
        UCHAR     iSubId        ,//子类代码
        UCHAR     iFunc         ,//功能代号
        UCHAR     iExFunc       ,//功能扩展
        PUCHAR     m_Data       ,//数据区
        USHORT     DebugMode    ,//调试模式
        UCHAR     iMode         ,//iMode=0:数据帧；1: 远程正帧
    )

```

5. CAN 报文接收

CAN 报文发送函数负责 CH365 报文的接收以及其他情况处理。CAN 报文的接受主要有两种方式：中断接收方式和查询接收方式。下面提供的是以中断方式接收报文的函数。

//SJA1000 发送缓冲区接收报文

```

BOOL CH365_CanRece(
    USHORT      iDevice ,
    mPCH365_IO_REG m_IoBase,
    p_RingChain volatile ReceBuff);

```

//系统接收报文

```

BOOL CH365_SysRece(
    p_RingChain volatile ReceBuff,
    p_Frame_ExF          p_Frame );

```

● 应用层接口子程序功能说明

- (1) 获得驱动程序版本号,返回版本号,出错则返回 0

```
ULONG WINAPI CH365_CAN_GetDrvVersion();
```

- (2) 打开 CH365 设备

```
HANDLE WINAPI CH365_Can_mOpenDevice();
```

- (3) 获取中断号

```

BOOL WINAPI CH365_Can_mGetIntLine(
    PULONG      oIntLine); // 保存中断号的单元地址

```

- (4) 设定中断号

```

BOOL WINAPI CH365_Can_mSetIntLine(
    ULONG      iIntLine); // 指定中断号,为 0 则关闭中断,为-1 则自动检测并设定

```

- (5) CAN 设备初始化

```

BOOL WINAPI CH365_CanBus_Init(
    UCHAR      AMR_Code[], //AMR 掩码
    UCHAR      ACR_Code[], //ACR 识别码
    UCHAR      m_Time1 , //
    UCHAR      m_Time2 , //波特率设定
    UCHAR      DebugMode); //调试模式

```

验收代码位 (AC.7~AC.0) 和报文标识码的高 8 位 (ID.10~ID.3) 必须相等,而这些位是被验收屏蔽位 (AM.7~AM.0) 的相应位屏蔽的。即如果条件满足下列两等式之一,则报文被接受。

$$(ID.10 \sim ID.3) \equiv (AC.7 \sim AC.0)$$

$$(AM.7 \sim AM.0) \equiv 11111111$$

m_Time1 (CAN 地址 6) 包括波特率预置器位域和同步跳转宽度位域; m_Time2 (CAN 地址 7) 包括采样位以及时间段 1 和时间段 2 位域。

调试模式采用自接收模式,利用自接收请求命令,即使没有接收后的应答,CAN 控制器也会成功发送。

- (6) 获取 I/O 端口的基址

```

BOOL WINAPI CH365mGetIoBaseAddr(
    mPCH365_IO_REG *oIoBaseAddr); // 保存 I/O 端口基址的单元地址

```

(7) 获取存储器的基址（不会用到）

```
BOOL WINAPI CH365mGetMemBaseAddr(
    mPCH365_MEM_REG * oMemBaseAddr ); // 保存存储器基址的单元地址
```

(8) CAN 设备发送

```
BOOL WINAPI CH365_CanBus_Send(
    UCHAR      iRegion    ,//域名代码
    UCHAR      iSubId     ,//子类代码
    UCHAR      iFunc      ,//功能代号
    UCHAR      iExFunc     ,//功能扩展
    PCHAR      m_Data     ,//数据区
    UCHAR      I Mode     );//iMode=0:数据帧; 1: 远程帧
```

发送缓冲器的整体布局：描述符区占 5 个字节，其中第一个字节是帧模式字节，说明是数据帧或者远程帧，后面 4 个字节分别为域名代码字节、子类代码字节、功能代号字节和功能扩展字节，可根据用户需要自己设定；数据区占 8 个字节。发送缓冲器总长 13 个字节。

(9) CAN 设备接收

```
BOOL WINAPI CH365_CanRece(p_Frame my_Frame);
```

(10) 设定中断服务程序

```
BOOL WINAPI CH365_CanIntRoutine(
    mPCAN_INT_ROUTINE iIntRoutine, // 指定中断服务程序,为 NULL 则取消中断服务,
    否则在中断时调用该程序
    LONG              ServerPTR ) ;//等待类型
```

其中等待类型有两种：发送等待 mFuncWaitingSend，先发送，等待发送成功中断再返回；接收等待 mFuncWaitingReceive，等待接收中断确定接收。两种等待类型只能选其中一种。

(11) 放弃等待中断

```
BOOL WINAPI CH365_CanAbortWaitInt();
```

(12) 关闭 CH365 设备

```
VOID WINAPI CH365mCloseDevice();
```

附录 CH365 与 CAN 控制器 SJA1000 接口电路原理图

（阅读此图时，请逆时针旋转 90°）

