

基于 WinCE 的智能车载仪表设计

引言

随着高性能电子显示技术的发展,汽车仪表电子化的程度越来越高。国内外已开发出了多功能全电子显示仪表、平视显示仪表、汽车导航系统、行车记录仪等高技术产品。未来,车用电子化嵌入式仪表具有以下优点:提供大量复杂的信息,使汽车的电子控制程度越来越高;满足小型、轻量化的要求,使有限的驾驶空间更人性化;高精度和高可靠性实现汽车仪表的电子化,降低了故障的发生率;设有在线故障诊断系统,一旦汽车发生故障,可以找到故障来源,方便维修;外形设计自由度高,汽车仪表盘造型美观。基于以上优点,汽车会越来越多地采用各种用途的电子化仪表。造型新颖、功能强大的嵌入式电子化仪表将是今后车用仪表的发展趋势和潮流。

1 智能车载仪表系统结构

本智能车载仪表拥有大多数传统车载仪表所拥有的功能,驾驶员可以通过车载仪表的显示界面获取当前汽车的状态信息,例如车速、油压、油温、水温、机油油压力或者电瓶电量。

传统车辆仪表直接与车辆的传感器相连,仪表系统经由传感器的模拟量得到汽车当前状态,精确性不高。本文设计的智能车载仪表并不是简单地与传感器相连,而是通过 CAN 控制器将整车连接成一个网络结构。车辆部件配以 CAN 控制器,通过双绞线将车辆部件连接起来形成一个网络体系,实现部件的电子化。同时,车载仪表和汽车部件的电子化也提高了汽车的精准度和可靠性,降低故障发生率。

车载智能仪表主要分为基于 S3C2440 处理器的硬件系统和 WinCE 环境下的软件系统两大部分。硬件系统为整个控制系统提供基础,负责 CAN 总线通信。软件系统提供 CAN 总线的硬件驱动以及在 WinCE 下的仪表上位应用程序。

2 硬件设计

硬件系统以 S3C2440 为核心, RAM 内存、NOR Flash 和 NAND Flash 作为存储介质,扩展部分外围设备以负责系统信息的输入与输出,如 CAN 总线通信单元、LCD 显示、触摸屏、通用串行口、USB 设备、以太网接口等。系统硬件结构如图 1 所示。

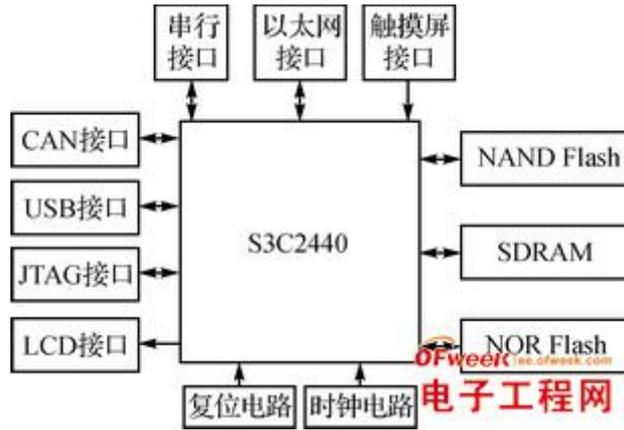


图 1 系统硬件结构框图

在众多接口中，CAN 总线通信单元是在整车通信过程中的关键部分。在汽车各个重要部件中，配置相应的 CAN 控制单元，由双绞线将各个 CAN 总线控制单元连接起来。汽车的各个部件将该部件的当前状态信息由 CAN 控制单元发送出去，经双绞线发送到智能车载仪表的 CAN 单元当中，经过系统的 CAN 接口将数据发送到系统中。车载仪表系统得到数据后，经过数据处理得到汽车部件的当前状态信息。

CAN 总线接口电路如图 2 所示。采用 Microchip 公司的 CAN 总线控制器 MCP2515。MCP2515 完全支持 CAN 2.0A/B 技术规范，速度达到 1 Mbps；SPI 的接口标准使得它与 S3C2440 的连接更加简单；能发送和接收标准和扩展数据帧以及远程帧；自带 2 个验收屏蔽寄存器和 6 个验收滤波寄存器，可以过滤掉不想要的报文，减少了微处理器的开销。CAN 总线收发器采用 TJA1050，该器件提供了 CAN 控制器与物理总线之间的接口以及对 CAN 总线的差动发送和接收功能。

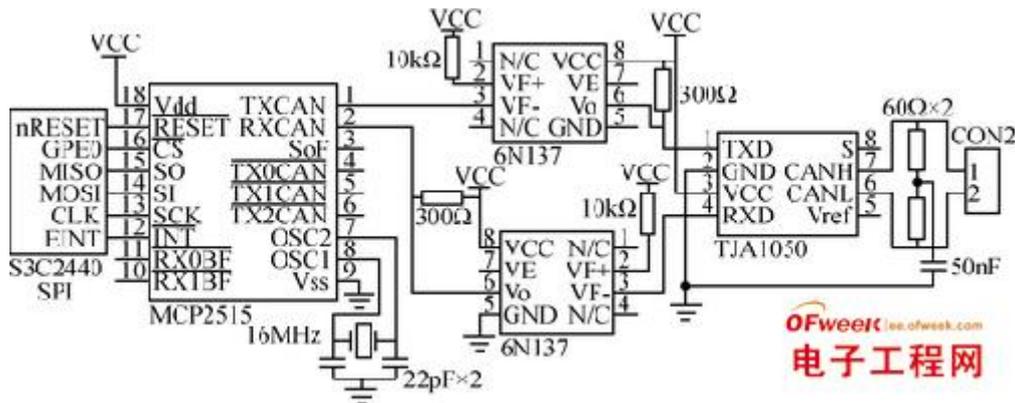


图 2 CAN 总线接口电路

为了增强 CAN 总线节点的抗干扰能力，提高系统的稳定性，在 CAN 控制器与 CAN 收发器之间加入了光耦隔离器 6N137，而不是使 TXCAN 和 RXCAN 端直接与收发器相连，这样就实现了总线上各 CAN 节点之间的电气隔离。同时，这也解决了 MCP2515 与 TJA1050 之间电平兼容的问题，还可以抑制 CAN 网络中的尖峰脉冲及

噪声干扰。光耦部分电路所采用的两个电源必须完全隔离，否则也就失去了意义。电源的隔离可以采用小功率的电源隔离模块或者多带 5 V 隔离输出的开关电源模块实现。这些部分虽然增加了接口电路的复杂性，但是却提高了节点的稳定性和安全性。

在 CAN 接口处，CAN 通信线上的 2 个 $60\ \Omega$ 电阻（总计 $120\ \Omega$ ），起到增大负载、减少回波反射作用，是一种阻抗匹配的补救措施。2 个 $60\ \Omega$ 的中间部分与地端之间连接一个电容以抗干扰。

3 软件设计

软件的整体环境为 WinCE 编程环境。针对本车载智能仪表硬件系统定制相应的 WinCE 操作系统，实现对硬件的驱动。再编写应用程序，通过对应用程序的具体操作实现对系统硬件的操作，即实现系统的功能。其中非常关键的是编写 CAN 控制器的驱动。CAN 驱动实现应用软件对 CAN 控制单元的操作，以及读取 CAN 控制单元中的数据代码。

3.1 系统开发和移植

嵌入式系统开发就是系统驱动层的设计，其中最主要就是 BSP 的开发和调试。所以智能车载仪表底层驱动的开发就显得尤为重要。

由于使用的是 WinCE 操作系统，所以使用 Platform Builder 定制 WinCE 操作系统镜像。在 Platform Builder 中，可以添加系统部分硬件（如液晶屏、RAM）的驱动，这些驱动已经由微软公司编写好。然后启动 Bootloader，把镜像文件下载到 Flash 存储器中，并配置操作系统启动文件 boot.ini。

3.2 CAN 总线驱动开发

由于 CAN 是外部设备，所以需要将 CAN 的驱动以流接口驱动方式编写。流接口驱动函数被设计来与通常的文件系统 API（如 ActivateDevice、ReadFile、WriteFile 和 IOControl 等）紧密匹配，即流接口驱动在应用程序中表现为一个系统文件，应用程序通过对系统文件的特殊文件进行操作从而完成对设备的操作。编写流接口文件主要用到流接口函数，也就是流接口驱动的入口点，如 XXX_Init、XXX_Read 和 XXX_Open 等。这些流接口文件与相应的 API 函数对应，使应用程序由相应的函数可以访问到外部设备。

作为流接口驱动程序，CAN 总线驱动程序也有一组标准函数，可完成 I/O 操作。这些函数提供给 WinCE 操作系统内核使用，它们都是流接口驱动程序的 DLL 文件。动态加载 CAN 总线驱动程序时，系统要进行注册。CAN 驱动程序注册表信息：

```
WCHAR * szRegKey=L "DriversBuiltInCAN" ;
```

```
WCHAR * szPrefix=L "CAN" ;
```

```
WCHAR * szDLL=L “can.dll” ;
```

```
WCHAR * szFriendlyName=L “CANDriver” ;
```

```
DWORD * DrvIndex=1;
```

```
DWORD * DrvOrder=0;
```

通过以上代码完成注册之后，调用 ActivateDeviceEx（）函数进行加载驱动。

3.3 CAN 总线驱动通信流程

CAN 总线控制器驱动程序主要用于设置 MCP2515 的波特率、MCP2515 的验收过滤器、CAN 消息传送模式和 CAN 收发数据方式。驱动程序是连接上位应用程序和硬件的一个中间纽带。智能车载仪表系统在 EVC++环境下进行 CAN 总线开发。

流式驱动以文件的方式打开函数 CreateFile（）：驱动成功打开后，CreateFile（）函数返回不为 INVALID_HANDLE_VALUE 的句柄，此句柄也是关闭驱动函数 CloseHandle（）的参数。CAN 设置函数 DeviceIoControl（）接收由 CreateFile（）函数获得的句柄及控制命令码，设置输入/输出缓冲区以及缓冲区的大小。函数执行完成后返回实际输出的缓冲区大小等。CAN 总线发送与接收函数 WriteFile（）和 ReadFile（），利用驱动文件句柄、发送和接收缓冲区等参数完成数据的发送和接收，操作成功后返回实际发送和接收的字节数。

为了不让主线程一直处于等待数据到来的状态，让主线程拥有时间令牌来操作其他事情，程序采用多线程设计，创建一个接收数据线程。接收数据线程是一个无限循环，它不断查询 CAN 总线数据退出标志，如果退出标志有效，则该循环结束退出。驱动程序流程如图 3 所示。

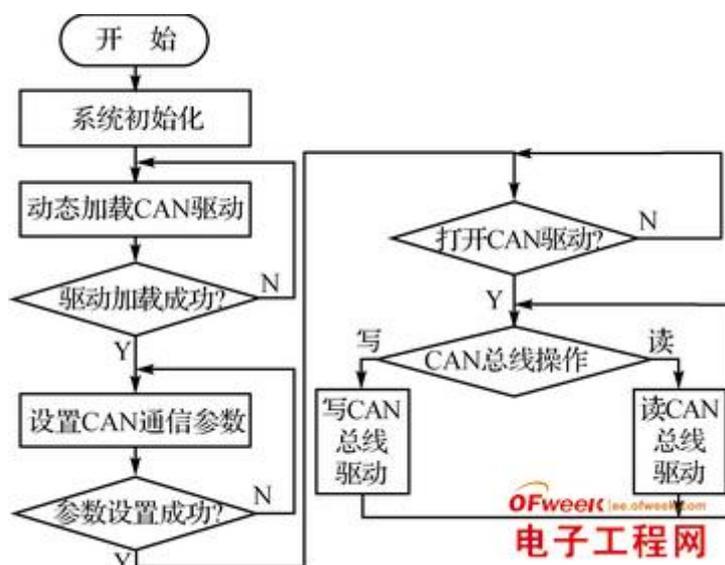


图 3 CAN 总线驱动通信流程

结语

目前，基于嵌入式技术的车载仪表逐渐开始流行。本文使用基于 WinCE 的 ARM9 微处理器作为开发平台，处理速度快，功能强大；使用 CAN 总线技术进行信号的相互传输，CAN 总线实时接收汽车部件发送来的信息，并进行处理与分析；使用 WinCE 操作系统，界面友好。本智能车载仪表系统可以节约成本、降低功耗，并且可维护性强，便于扩展和升级。