

基于 CAN 总线的嵌入式汽车仪表设计

王璐, 潘明

(桂林电子科技大学 电子工程与自动化学院, 广西 桂林 541004)

摘要: 在 CAN 总线技术基础上, 设计了一款 CAN 总线嵌入式汽车仪表。该仪表采用 Samsung 公司的 S3C6410 处理器, 利用 Nokia 公司开发的 Qt Quick 及 QML 语言制作美观大方的人机交互界面以实时显示车辆工作情况。给出了系统软硬件结构设计方案。检测结果表明, 该设计方案满足了汽车仪表指示精度与稳定性要求。

关键词: QML; CAN 总线; S3C6410; GUI

中图分类号: TP274

文献标识码: A

文章编号: 1674-7720(2013)07-0020-04

Design of embedded automobile instrument based on CAN bus

Wang Lu, Pan Ming

(School of Electrical Engineering and Automation, Guilin University of Electronic and Technology, Guilin 541004, China)

Abstract: An embedded automobile instrument is designed based on CAN bus in this paper. This instrument adopts S3C6410 ARM CPU, and it gives the design scheme of the system's hardware and software. The condition of automobile can be displayed beautiful human-computer interaction interface, which is made by using the Qt Quick and the QML language which developed by Nokia. The test result shows that the design can meet the automobile instrument's indicating precision and stability requirements.

Key words: QML; CAN bus; S3C6410; GUI

中国汽车工业强劲发展, 产销能力排行世界第一, 但绝大部分国产汽车仍搭载步进电机式汽车仪表。随着电子技术的迅猛发展, 汽车信息容量与系统复杂程度产生了很大的变化。目前, 国外各大汽车公司都在研究全数字式汽车仪表, 汽车仪表正加速向智能化、数字化和网络化方向发展。

本文设计了一种基于 CAN (Controller Area Network) 总线的汽车智能仪表系统。设计思路是将获取的行驶速度、发动机转速、油量、里程表、冷却液和警示灯等数据显示在人机交互界面上, 并将巡航控制、导航、视频和音乐等功能集成到该仪表系统当中, 所有信息都直观显示在大屏幕 TFT-LCD 上, 触摸屏技术给驾驶员与乘员带来更为有趣的行驶过程。

Qt 是一个跨平台的 C++ 图形用户界面库, 它基于面向对象的思想, 扩展容易且可移植性好, 其良好的封装机制使得 Qt 模块化程度很高, 用户开发十分方便, 并具有优良的跨平台特性。Qt 包括多达 250 个以上的 C++ 类, 支持 2D/3D 图形渲染和 OpenGL, 集成 Webkit 引擎。

本文采用 Nokia 最新开发的 Qt Quick 高级用户界面技术, 为嵌入式设备快速轻松地创建动态触摸式界面和轻量级应用程序。在 Qt Quick 中, 新增加了一种简便易学的语言 QML, 这种语言能让 Qt 应用程序开发变得更为方便快捷。

1 汽车仪表总体结构

汽车仪表主要由数据接收发送、数据处理与存储和触摸屏显示 3 个模块组成, 如图 1 所示。

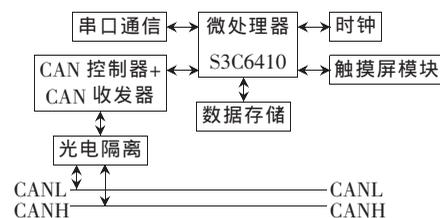


图 1 嵌入式汽车仪表硬件系统

数据接收发送模块负责接收 CAN 总线传输的各种数据, 如速度、转速和温度等, 并将处理后的数据发送到微处理器; 或者将微处理器发送的数据进行处理之后发

送至 CAN 总线。微处理器接收到需要的数据后,按照预定的算法和要求对数据进行处理,并将结果输出。人机交互界面程序接收到微处理器发送的数据后进行相应的显示,并将触摸事件反馈给微处理器。

2 汽车仪表硬件设计

本仪表采用 Samsung 公司的 S3C6410 处理器,它是一款低功耗、高性价比的 32 bit RSIC 处理器,其处理器在视频媒体编解码、2D/3D 图形加速、显示处理和缩放等方面有很大的性能优势。S3C6410 有很多硬件功能外设,如 TFT 真彩色 LCD 控制器,4 通道的 UART,4 通道定时器,通用 I/O 口,FS 总线,PC 总线,USB Host,高速 USB OTG、SD Host、MMC 卡接口以及内部的 PLL 时钟发生器等。该处理器具有很大的扩展空间,在后续开发中潜力巨大。Qt Quick 开发出美观炫丽的用户图形界面,需要选择图形处理能力较强的处理芯片,因此选择 S3C6410 处理器。

S3C6410 处理器没有内置 CAN 控制器模块,需要外接一个 CAN 控制器和一个 CAN 收发器。该仪表选用美国微芯科技有限公司生产的 MCP2510 作为 CAN 控制器。MCP2510 是带有 SPI 接口的独立的 CAN 控制器,它支持 CAN 2.0A/B,含有 3 个发送缓冲器和两个接收缓冲器,可以使 MCP2510 发送和接收标准帧和扩展帧。本设计采用飞利浦公司生产的 TJA1050 作为 CAN 收发器,如图 2 所示。将 MCP2510 的数据输出 TXCAN 和 RXCAN 引脚通过 6N137 光耦接到 TJA1050 的 TXD 和 RXD 引脚上,MCP2510 的 SPI 接口的 SCK、SI、SO 引脚用导线引出,S3C6410 通过 SPI 接口与 MCP2510 进行实时通信,MCP2510 将数据转换成 CAN 报文发送到 TJA1050,TJA1050 把 MCP2510 传输的信号转换为差分电压信号发送到 CAN 总线上。

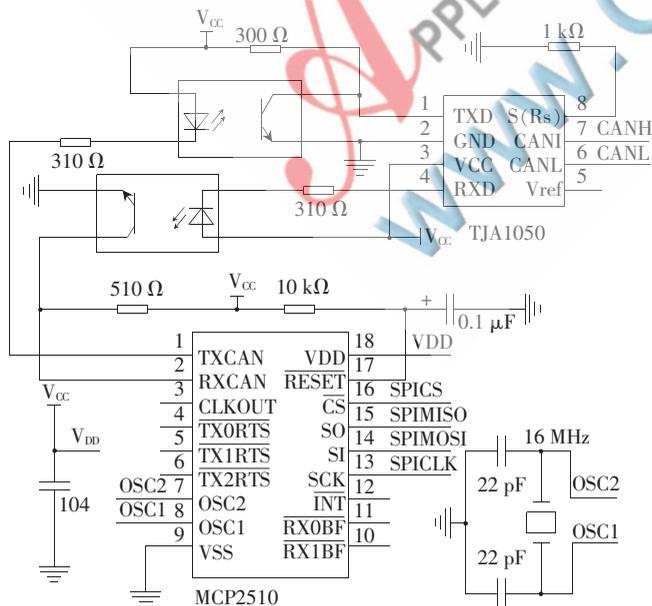


图2 CAN 控制器与收发器接口电路

3 汽车仪表软件设计

3.1 CAN 总线通信

德国 Bosch 公司开发的 CAN 总线是为解决现代汽车中大量的电控单元而开发的一种串行通信协议,在车载各电子控制单元 ECU (Electronic Control Unit) 之间交换信息,形成了重要的汽车电子控制网络。CAN 技术规范遵照 ISO/OSI 标准定义的 7 层基准参考模型,按照这个标准模型,最早的 CAN 规范只定义了数据链路层和物理层,用户需要自己制定相应的应用层协议。J1939 协议是美国汽车工程协会 SAE (Society of Automotive Engineers) 根据 CAN 2.0B 为核心协议制定的协议。J1939 协议制定了应用层协议,明确规定了 ECU 地址、报文发送优先级、扩展帧格式和通信方式等。由 CAN 总线高速网络连接车载电子装置,实现了高速数据共享。因此本文采用 SAE J1939 协议。

3.1.1 SAE J1939 协议数据帧格式

SAE J1939 协议数据帧以协议数据单元 PDU 为单位,如图 3 所示。PDU 包含优先级(P)、保留位(R)、数据页(DP)、PDU 格式(PF)、PDU 细节(PS)、源地址(SA)以及数据域(Data Field)。指定 PDU (PS): 当 PF 值为 0~239 时,PS 是目标地址 DA; 当 PF 值为 240~255 时,PS 为组扩展(GE)。CAN 扩展帧中,SOF、SRR、IDE、RTR 位、控制域场、CRC、ACK 和 EOF 场没有包括在 PDU 内,这些场由 CAN 规范定义,未被 SAE J1939 修改。



图3 SAE J1939 协议数据帧格式

3.1.2 CAN 通信软件设计

本设计中 S3C6410 通过 SPI 总线与 CAN 控制器 MCP2510 通信,CAN 通信可分为各个独立的程序模块,包括 SPI 初始化、MCP2510 的初始化和发送接收子程序等。在 CAN 总线工作开始时,首先需要对 S3C6410 微处理器的 SPI 进行初始化操作,设置相应寄存器,然后对 MCP2510 进行初始化操作。在对 MCP2510 进行读写操作时,必须严格遵守 SPI 指令。SPI 指令包括复位(11000000)、读(00000011)、写(00000010)、发送请求(10000000)、状态读(10100000)和位修改(00000101)。

MCP2510 只有在配置模式下才能对器件进行正常的初始化。其中包括 CAN 通信的波特率、MCP2510 的配置寄存器、滤波/屏蔽寄存器和中断使能寄存器等。在进

行完初始化之后, 必须将 MCP2510 置为 Normal 模式。CAN 控制器 MCP2510 初始化流程如图 4 所示。

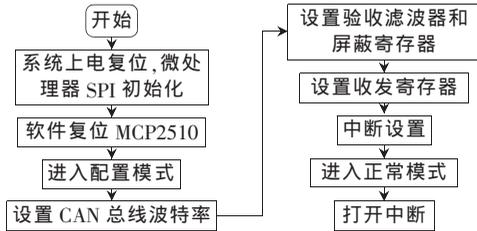


图 4 MCP2510 初始化流程图

3.2 人机交互界面设计

Qt Quick 提供了一套高动态、丰富的 QML 元素来制定用户图形界面的框架以及 Qt Quick Designer 开发工具。QML 语言是一种描述性语言, 用于描述用户界面的外观和行为。QML 中采用 JavaScript 作为脚本语言, 用户界面被规定为树形结构。

3.2.1 QML 语言模块以及语法

QML 已经定义了许多功能强大的界面元素, 如 Item、Rectangle、Image、Text 和 ListView 等, 其中 Item 是 QML 中可视元素的基础。import 语句导入包含所有标准 QML 元素的 Qt 模块, 其中一些元素可以作为其他元素 (children) 的容器 (parent)。每个元素具有唯一的 id 属性, 此 id 允许对象被其他的对象引用。属性以 name: value 形式指定。State、property、Animation、Transition、Timer 和 Connection 等元素用于描述应用程序的行为。在 QML 中常常用到的 Anchors 描述的是当前元素的位置和其他元素的相对关系。一个元素通过形式为 Type.property 的属性附加到另一个元素当中, 其中 Type 是附加 property 元素的类型。在 QML 中, 状态 (States) 用于管理有 id 的元素, 它由多个 state 元素构成, 每个元素都可以定义自己的不同状态, 当元素进入某个状态时, 状态所对应的属性将被设置修改。在设计中通过判断条件来决定是否激活一个状态, 使用 when 属性, 返回值为 true 或者 false。在 QML 语言描述一个按钮的单击功能时, 可以使用 QML 事件处理功能, 这个功能是非常类似于 Qt 的 Signal/Slot 机制。

QML 中用户界面创建数据模型非常重要, 具有资源利用率高、数据传输快、数据存储空间合理和用户界面流畅等优点。模型为其他部件提供数据接口, 在设计中使用 ListView 能快速建立菜单栏列表类型的界面。

QML 元素 ListModel 可以从内部创建这些 Model, 或使用 QAbstractListModel 继承的 C++ Model 类, ListItem 在 ListModel 中定义数据项, 列表 ListView 显示 Model 的数据, Delegate 定义如何显示数据。通过 ListView 定义 Items 的布局为水平布局和垂直布局来显示 Items 的布局方式。ListModel、Delegate 和 ListView 之间的关系如图 5 所示。



图 5 ListModel、Delegate 和 ListView 关系

3.2.2 QML 与 C++ 交互

Qt Quick 包含 QtDeclarative C++ 模块, 并且被整合进 Qt creator IDE 当中。QML API 有 QDeclarativeEngine、QDeclarativeComponent 和 QDeclarativeContext 3 个主要成员: QDeclarativeEngine 提供在 C++ 中 QML 应用的运行库以及 QML 内容与 Qt/C++ 程序中的接口; QDeclarativeComponent 加载 QML 文档, 每个 QDeclarativeComponent 实例对应一个 QML 文档, 文档 URL 可以是本地文件系统 URL 或任何 QNetworkAccessManager 支持的 URL; QDeclarativeContext 暴露数据给 QML 实例。

在 Qt/C++ 中使用 QML 开发的组件, 必须先让 QML 运行的环境被应用程序加载调用, 创建一个 QDeclarativeEngine 实例, 使用 QDeclarativeEngine 加载 QML 组件并对所有的 QML 组件进行全局配置。通过 QDeclarativeComponent::create() 方法创建 QML 组件实例。下面的代码演示了如何加载一个 QML Document 并创建一个实例。

```

QDeclarativeEngine*engine=new QDeclarativeEngine(parent);
QDeclarativeComponent component (engine, QUrl::
fromLocalFile("main.qml"));
QObject*Object=component.create();
  
```

为了把 QML 组件嵌入到 Qt/C++ 应用程序中, 需要使用 Qt 4.7 以上的 SDK。

3.2.3 人机交互界面开发

在本设计中设计人机交互界面主要涉及布局、Signal/Slot 通信机制的运用、状态转换、Transition 过渡效果、ListModel 与 ListView 元素等内容。通信方式使用 QML 发送 Signal, Qt/C++ 中定义用 Slot 接收。

在人机交互主界面中, 使用 Scale 缩放仪表盘图片, 使其适合屏幕, 设置 Rotate 来旋转仪表盘指针, 旋转默认是以图片中心为转轴, 但是可以通过 transformOrigin 属性来设置转轴。转向灯使用 State 元素, State 通过 PropertyChanges 修改属性, 当总线传来开启转向灯信号时, 图片透明度从较低值变换到较高值, 当接收到总线传来的关闭转向灯信号时, 图片透明度从较高值变换到较低值, 从而实现了警示灯的亮与灭。警示灯状态变化要迅速直观, 减少驾驶员对警示灯误判, 因此并没有采用 Transition 元素为状态转换间提供动画动态过渡, 同理, 其他警示灯也采用这种方法实现。状态转换同样用于仪表中菜单栏图标的显示中, 例如浏览器网页缩略图模式与浏览模式切换, 代码如下:

```

State{
name: "Internet"
  
```

```
PropertyChanges{
target : bookmark
y : parent.height
opacity : 0
}
PropertyChanges{
target : browser
opacity : 1
height : mainView.height
}
}
```

仪表各分界面菜单使用 ListView 显示, 例如定义一个菜单栏, 代码为:

```
ListView{id : buttons ; x : 10 ; model : {} ; delegate : ItemDrawer ;
orientation : "Horizontal" ;
width : container.width ;
height : container.height ;
spacing : 15 ; interactive : false }
```

基本的菜单栏都类似于上面的代码。在仪表系统中, 实现数字、英文字母与符号输入使用 Grid 元素来布局, spacing 属性控制网格单元之间的距离。代码例子如下:

```
Grid{
id : keyboard
spacing : 2
columns : 12
KeyButton{operation : "Q"}
.....
KeyButton{operation : "Enter" ; width : 50}
.....
KeyButton{operation : "1"}
.....
}
```

4 开发环境编译与移植

Qt Creator 是跨平台的 Qt IDE, 提供了图形化的调试窗口和 qmake 工具, Qt Creator 代码编辑窗口具有非常人性化的设计, 具有对 C++ 及 QML 语言表达式检查, 代码语句整齐排列, 代码自动补全, 对类与函数及变量作上下文关联分析, 告知函数声明和调用的位置。Qt Demo 提供大量的 Qt 实例并提供在 Qt Linguist 下的源代码链接, 通过 Qt Demo 可以快速学习 Qt 和 QML 编程。

要在 ARM 上运行 Qt 程序, 需要移植 ARM 版本的 Qt/E。进入到 ARM 版 Qt/E 目录当中, 执行配置源码命令 `echo yes | ./configure -opensource -embedded arm -xplatform qws/linux-arm-g++-webkit-qt-libtiff-qt-libmng-qt-mouse-tslib-no-mouse-linux-tp-declarative-no-qt3support -nomake examples -nomake demos -nomake docs -importdir/usr/local/Trolltech/QtEmbedded-4.8.0-arm/`

`qml - fast`。

在上面的语句当中, 最后的语句 `-importdir/usr/local/Trolltech/QtEmbedded-4.8.0-arm/qml` 是将 qml 相关库添加到 Qt/E 库当中, 使用 `arm-linux` 交叉编译器进行编译, `-xplatform qws/linux-arm-g++-qt-mouse-tslib` 命令为使用 `tslib` 来驱动触摸屏。编写好配置命令后, 使用 `make` 与 `make install` 命令进行编译安装, 然后将文件打包下载到 ARM 板并解压缩, 这样 Qt/E 的 ARM 环境就移植完毕。

5 汽车仪表测试

仪表系统是驾驶过程中非常重要的系统之一, 在此选用较深色背景作为 LCD 终端显示背景, 避免驾驶员出现视觉疲劳现象, 背景也可以根据用户喜好而改变。汽车车速表、转速表、燃油表与温度表都以圆盘式指针表显示, 这样的设计比较符合驾驶员对传统汽车仪表的使用习惯。菜单栏位于屏幕上部, 便于乘员对仪表盘进行操作。



图6 汽车仪表效果图

使用嵌入式 Linux 作为车载仪表系统是比较流行的解决方案之一, 其中嵌入式 GUI 能给用户提供良好的操作界面, 而 Qt Quick 能迅速开发美观易用的嵌入式 GUI, 在今后的实际应用中有着巨大的潜力, 有比较大的参考价值。

参考文献

- [1] 饶运涛, 邹继军, 郑勇芸. 现场总线 CAN 原理与应用技术(第2版)[M]. 北京: 北京航空航天大学出版社, 2007.
- [2] 廖乐平, 杨松, 高琛, 等. 基于 SAE J1939 的车身控制系统设计[J]. 汽车实用技术, 2011(5): 31-33.
- [3] 宋晓东, 王建. CAN 总线汽车仪表研究[J]. 电子设计工程, 2010(4): 35-37.
- [4] 李宏梅, 佟为明, 程树康. CAN 总线全数字式汽车仪表[J]. 电子器件, 2010(5): 646-650.
- [5] 肖楚海, 梁杰申, 黄炜中. 基于 CAN 总线的汽车仪表的设计[J]. 信息技术, 2010(2): 80-82.

(收稿日期: 2012-10-12)

作者简介:

王璐, 男, 1987年生, 硕士, 主要研究方向: 汽车电子。