

# 基于 Qt/Embedded 嵌入式数字兆欧表 GUI 的开发与实现<sup>①</sup>

连文生, 索强强, 崔向阳

(北方民族大学 电气信息工程学院, 银川 750021)

**摘要:** 基于 Qt/Embedded 设计了一款嵌入式数字兆欧表 GUI, 与触摸屏结合取代了原有的机械开关, 并建立数据库实现了历史数据保存、打印等现有数字兆欧表没有的功能。

**关键词:** Qt/Embedded; GUI; 触摸屏; 数据库

## Development and Implementation of Embedded Digital Megger GUI Based on Qt/Embedded

LIAN Wen-Sheng, SUO Qiang-Qiang, CUI Xiang-Yang

(College of Electrical and Information Engineering, North University for Ethnicity, Yinchuan 750021, China)

**Abstract:** Based on the Qt/Embedded, this paper designs a kind of embedded digital megger GUI. It uses touch screen instead of original mechanical switch, establishes database storage and finally realizes some functions which current digital megger lacks, such as the storage and printing of history data the history printing.

**Key words:** Qt/Embedded, GUI Touchscreen, database

随着微电子、计算机等技术的发展, 嵌入式系统已应用于我们生活的各个领域<sup>[1]</sup>. 而实用美观的图形用户界面(GUI)已成为决定嵌入式产品性能的指标之一. 在嵌入式系统中, Linux 操作系统因其源码完全开放、可移植性强, 支持多任务多用户等特点得到广泛的应用. 本设计基于嵌入式 Linux 系统, 对 Qt/Embedded 进行了裁剪和移植. Qt/Embedded 是 QT 的嵌入式版本, 包含有完整的图形用户界面和 C++类库, 并且具有完全面向对象、容易扩展、提供丰富窗口部件的特点<sup>[5]</sup>. 本设计采用交叉编译的方法实现 GUI 的开发, 首先在宿主机上设计界面并做初步的调试, 然后通过交叉编译器编译生成可执行文件下载到目标板进行最终调试.

### 1 数字兆欧表原理

由于在嵌入式数字兆欧表 GUI 的开发过程中, 需要对一些设备驱动文件和 IO 端口进行操作, 所以我们简单的介绍一下数字兆欧表的工作原理. 数字兆欧表(即绝缘材料测量仪)主要用来测量绝缘材料的绝

缘性能, 被广泛的应用于电力设备的安装调试中. 整个系统原理框图如图 1 所示, 具体工作流程为: 12V 直流电源通过直流负高压生成模块产生所需的高电压, 再经过模拟开关加到采样电阻两端, 采样电阻的电信号经过 12 位高速 AD 转换器转换为数字信号, S3C2440 微处理器通过计算将数字信号转化为绝缘阻值, 并根据不同的绝缘阻值自动切换采样电阻<sup>[2]</sup>.

由图 1 可以看出在整个系统中 S3C2440 微处理器需要对 LCD、触摸屏、模拟开关、打印机等模块进行读写操作, 因此, 在嵌入式 Linux 操作系统的定制时需要加入相应的驱动程序.

### 2 Qt/Embedded在嵌入式系统中的应用

Qt/Embedded 是 Trolltech 公司开发的面向嵌入式系统的 Qt 版本, 是一个为嵌入式应用定制的用于多种平台的图形界面程序开发工具包, 采用 C++封装, 具有丰富的控件资源和较好的可移植性. Qt/Embedded 的设计原则是不依赖于 X Server 或者 Xlib 库, 而是直接访问帧缓存(FrameBuffer), 这样提高了显示速度并

<sup>①</sup> 收稿时间:2012-07-27;收到修改稿时间:2012-09-10

大大减少了内存的消耗。

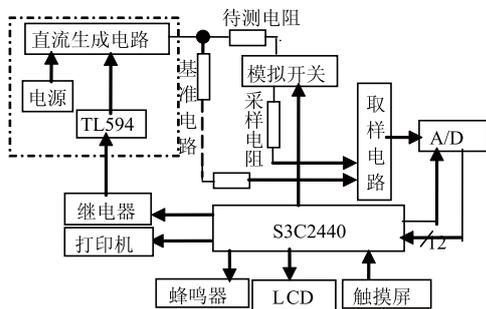


图 1 数字兆欧表的原理框图

Qt/Embedded 在 Linux 系统中的结构如图 2 所示<sup>[3]</sup>, Qt/E 基于客户/服务器架构, 有一个服务器(Server) 端程序. 该程序负责监听系统事件(Event), 如鼠标和键盘输入事件、屏幕显示等. 每一个 Qt/E 应用程序都需要一个服务端程序, 或者本身就作为一个服务端程序运行. 系统中产生的所有事件, 都会先传到服务端, 然后再由服务端根据情况分发给相对应的客户端应用程序. 另外 Qt/E 区别于其它工具的重要地方是信号(SIGNAL) 和槽(SLOT) 机制, 它是 Qt/E 编程的基础. 通过信号和槽, 能够使 Qt/E 各组件或对象在不知道对方的情况下进行通信. 信号和槽是一种高级接口, 它是 Qt/E 的核心特性.

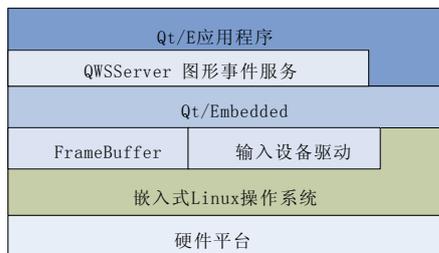


图 2 Qt/Embedded 结构

### 3 嵌入式数字兆欧表 GUI 的开发

在进行图形用户界面(GUI) 开发之前, 首先要搭建好宿主机的开发环境, 本设计采用的操作系统是 Fedora, 使用 qt-sdk-linux-x86-opensource-2010.05.1.bin 软件包作为图形界面开发工具. 搭建好环境后在 QT Creator 中建立 Digital\_Megger 工程, 基类选择为 QWidget.

#### 3.1 主窗口界面

接下来在 Designer 视图界面中设计主窗口界面并添加函数实现其功能. 整个主窗口的设计如图 3 所示,

最顶层是菜单栏, 下面是工具栏和操作界面, 在操作界面的下面留有一定的空间来设计状态栏. 下面说明主窗口中主要部件的实现方法:

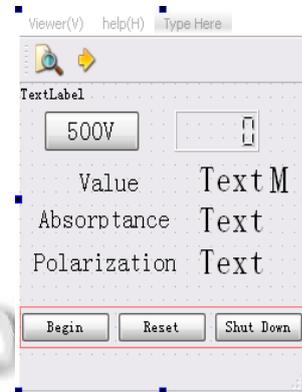


图 3 主窗口设计界面

#### (1) 系统时间显示

工具栏下面的第一个标签(项目名 time) 我们用来显示系统时间. 在 mainwindow.h 中添加槽函数声明: private slots: void timeUpdate() 切换到 mainwindow.cpp 中编写实现代码. 为使系统时间实时显示, 首先建立一个 QTimer 类定时器对象并以秒为启动单位发送信号, 然后将定时器信号与前面声明的槽函数关联起来: connect(timer1, SIGNAL(timeout()), this, SLOT(timeUpdate()));

系统时间更新函数 timeUpdate() 的实现分三步: 获取系统时间、设置显示格式、标签显示. QT 中提供了一个用于时间的类 QDateTime, 用它来获取系统时间, 实现代码如下:

```
QDateTime time = QDateTime::currentDateTime(); //获取系统时间
QString str = time.toString("yyyy-MM-dd hh:mm:ss dddd"); //设置系统时间显示格式
ui->time->setText(str); //在标签上显示时间
```

#### (2) 档位选择按钮

本兆欧表中有四个档位可供选择, 通过一个 Push Button 的 clicked() 响应函数来实现, 在切换档位之前先判断系统状态, 如果正处于测量状态则通过 QMessageBox 提醒用户先停止测量. 对于档位的判断, 设有一个在头文件中已声明的 int dangwei 变量, 每点击一次按钮 dangwei 加 1, 计满 4 次后重至为 0. 部分代码如下:

```

if(b) //b 是 bool 型测量判断标志
    QMessageBox::warning(this,tr("Warning"),tr("Please stop
measuring first!")); //警告
else
    { dangwei++; //mainwindow.h 中声明的全局变量
      if(dangwei==1)
        ui->dangwei->setText(tr("1000V")); //档位切换
        ... }

```

### (3) 测量计时

测量计时与系统时间的实现相似, 这里设有一个对应多个槽函数的定时器 timer, 时间通过 LCD Number 显示. 在构造函数中定义定时器、设置 LCD Numbe 显示格式、关联定时器和槽函数等. 在测量计时更新槽函数 update\_lcdnum()中, 通过信号计数的方式统计时间, 最后使用下面代码显示时间:

```

QString str; //建立 QString 对象
str.sprintf("%0.1d:%0.1d:%0.1d",hex,min,sec); //将显示数据
按格式打印进 str
ui->lcdNumber->display(str); //显示

```

### (4) 状态栏

状态栏的设计采用代码方式, 首先建立 QStatusBar 类指针, 再依次获取状态栏、新建标签、设置标签大小形状显示效果和初始化等.

### (5) 开始、复位、关闭按钮

开始按钮主要是用来启动和停止前面提到的定时器 timer, 同时改变状态栏的一些显示信息; 复位按钮用来初始化一些参数如: 计时变量、显示等; 关闭按钮实现系统的软关闭, 调用系统函数 system("shutdown -s\n").

### (6) 测量值、吸收比、极化系数显示

测量结果的显示依赖于外围测量电路的信息采集, 而在 Linux 系统中对所有硬件模块的操作都可以看作普通文件来对待, 这就是设备文件. 测量阻止的计算是一个比较复杂的过程, 首先判断数模转换的值是否在一定范围内, 如果不在通过设备文件控制相应的 GPIO 口切换模拟开关, 从而改变采样电阻. 从 ADC 采集回来的数据还要经过一系列复杂的运算(可参考相关文献)才能转化为阻值显示. 吸收比是 60 秒与 15 秒测量阻值的比值; 极化系数是 600 秒与 60 秒测量阻值的比值. 为说明如何在 QT 应用程序中操作硬件设备, 下面以数模转化为例.

跟普通文件一样, 要操作设备文件必须先打开文件:

```

int fd = ::open("/dev/adc", 0);
然后对文件进行读写操作:
int len = ::read(fd, buffer, sizeof buffer - 1);
最后关闭设备文件: ::close(fd);.

```

## 3.2 历史数据查询对话框

历史数据的查询是建立在数据库基础上的, 所以在开发查询对话框之前要先建立数据库. 添加 database.h 头文件在其中实现 createConnection()函数, createConnection()中的主要代码如下:

```

QSqlDatabase db =
QSqlDatabase::addDatabase("QSQLITE"); //添加数据库驱动
db.setDatabaseName("database.db"); //数据库连接命名
if(!db.open()) //打开数据库
return false;

```

为建立数据库连接通常在 main()中调用 create Connection(), 这时就可以向数据库中存取数据了, 本设计中每隔 10 秒钟向数据库中存入测量时间和阻值, 主要利用 QSqlQuery 类执行 SQL 语句完成<sup>[4]</sup>.

历史数据查询对话框通过菜单栏和工具栏中的图标来触发, 因此需编写相应的触发响应函数. 在触发函数中以 MainWindow 为父窗口新建一个对话框, 设置其标题、添加一些部件, 并以绝对位置法排放各部件. 查询功能通过 show\_findValue()函数实现. 对话框运行效果如图 4 所示.

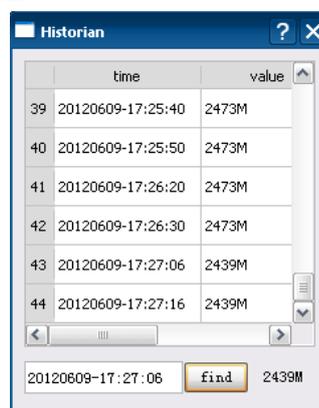


图 4 历史数据查询对话框

## 3.3 打印实现

QT 中打印的实现有一个常用步骤: 首先创建一

个当作绘制设备的 QPainter; 这时需弹出 QPrintDialog 对话框, 允许用户设置打印选项; 然后在创建的 QPainter 上操作 QPainter, 使用 QPainter 绘制一页, 再调用 QPainter::newPage() 绘制下一页. 打印代码可按照以上过程编写, 具体操作可查看参考文献<sup>[4]</sup>.

#### 4 嵌入式数字兆欧表 GUI 的实现

前面讲述的嵌入式数字兆欧表 GUI 开发是在宿主机的 QT 集成开发环境中进行的, 然而要想实现 GUI 在目标板运行, 还需搭建宿主机 Qt/E 编译环境和目标板的 Qt/E 运行环境. 下载 arm-qte-4.6.3.tar 版本, 解压后设置一些相关的环境变量, 然后编译、安装<sup>[4]</sup>. 搭建 Qt/E 运行环境时, 要对 Qt/E 进行适当的裁剪, 不然生成的库会很大; 配置时选择 ARM 平台, Qt/E 库为 release 版本, 添加 MySQL 数据库插件, 使用触摸屏支持库等; 最后编译、安装<sup>[5]</sup>.



图 5 实际运行效果图

在 Qt/E 环境搭建好后, 就可以编译 QT 应用程序了, 将先前开发好的 GUI 原文件复制到 Qt/E 安装目录下, 使用 qmake-project 生成项目文件, 再 qmake 后生成 Makefile 文件, 最后 make 生成可执行文件. 这时可将生成的可执行文件下载到目标板调试运行.

为使 GUI 开机自启动, 本设计还编写了一个 shell 脚本用于启动驱动和配置 QT 与触摸屏的运行环境. 实际运行效果如图 5 所示.

#### 5 结语

本文介绍了嵌入式数字兆欧表 GUI 开发与实现的整个过程, 对其他嵌入式产品 GUI 的开发具有一定的参考意义. 使用 QT 集成开发环境与交叉编译 Qt/E 程序是一种非常高效的嵌入式 GUI 开发方法, 在目前得到广泛的应用.

#### 参考文献

- 1 田福英. 基于 Linux Qt 的软键盘设计与实现. 计算机与现代化, 2011(12): 179-181.
- 2 虎恩典. 单片机控制的子兆欧表. 测控技术, 2004, 23(12): 71-73.
- 3 方世烟, 林东. 基于 QT 的电纸图形用户界面设计. 计算机系统应用, 2012, 21(3): 87-90.
- 4 Blanchette J, Summerfird M. C++ GUI QT4 编程. 第 2 版. 北京: 电子工业出版社, 2008.
- 5 倪继利. QT 及 Linux 操作系统窗口设计. 北京: 电子工业出版社, 2006.

(上接第 177 页)

- 检索方法. 计算机应用, 2009, 12(29): 39-42.
- 4 李敏, 罗红艳. 一种改进的最大类间方差图像分割法. 南京理工大学学报, 2012, 4(36): 2-6.
- 5 郝颖明, 朱枫. 二维 Otsu 自适应阈值的快速算法. 中国图象图形学报, 2005, 10(4): 484-488.
- 6 郑宏, 潘励. 基于遗传算法的图像阈值的自动选取. 中国图象图形学报, 1999, 4(4): 245-249.
- 7 Gonzales RC, Woods RE. Digital image processing. Beijing:

Publishing House of Electronics Industry, 2002: 460-507.

- 8 秦开怀, 王海颖, 郑辑涛. 一种基于 Hough 变换的圆和矩形的快速检测方法. 中国图象图形学报, 2010, 15(1): 109-115.
- 9 陈芳艳. 禁令标志和警告标志检测技术研究. 长沙: 中南大学, 2010.
- 10 刘哲. 基于信息融合的遥感图像处理方法研究. 西安: 西北工业大学, 2002.