

# Qt/Embedded 触摸屏的智能家居无线传输系统设计

李新春 徐英华 (辽宁工程技术大学 电子信息与工程学院 辽宁 葫芦岛 125105)

**摘要:** 设计了一个基于 Qt/Embedded 触摸屏的智能家居控制平台, 并通过 zigbee 组网的方式, 实现家居的智能化控制。首先, 智能家居的界面用 Qt/Embedded 设计, 然后, 界面被移植到以 ARM9 为微处理器的控制平台上, 并实现界面的触摸屏功能; 最后, 通过串口驱动将设计的具有触摸屏功能的界面与 zigbee 组网的协议连接起来, 实现对智能家居控制的无线传输。

**关键词:** Qt/Embedded; ARM; zigbee; 触摸屏; GUI

## Design of Intelligence Home Wireless Transmission System Based on Qt / Embedded of Touch Screen

LI Xin-Chun, XU Ying-Hua (Department of Electronic Information and Engineering, Liaoning Engineering Technology University, Huludao 125105, China)

**Abstract:** This paper designs the intelligence home control platform based on Qt/Embedded of touch screen, realizing the intelligence control of home by way of zigbee networking. Interface designer of intelligence home uses Qt/Embedded. At the same time, interface will be transplanted to the control platform for microprocessor of ARM9, and achieve the touch-screen features of interface. Finally, the paper connects the interface of a touch-screen features with zigbee networking through the serial port driver, to realize wireless transmission about the intelligence home.

**Keywords:** Qt/Embedded; ARM; zigbee; touch screen; GUI

### 1 引言

随着生活质量的提高, 家居智能化已经成为当今时代的一种主流。如何更好的设计智能家居的整体性控制就显得越来越重要。由于 ARM 和触摸屏的迅速发展, 采用更好的 GUI, 实现友好的界面控制成为了可能; 同时由于 zigbee 无限传输技术的广泛应用, 通过设计 zigbee 的协议组网, 使 ARM 能轻易的通过的串口驱动程序实现智能家居的控制; 同时, 通过 ARM 对万能遥控器的识别码的学习, 实现了 ARM 在智能家居上的多功能应用和智能家居的一体化控制。

### 2 系统设计概述

基于嵌入式智能家居的设计结构框图如图 1 所示。主要包括以下几部分。Qt/E 及 linux 系统、ARM9

微处理器、zigbee 模块、语音模块、GSM 模块、串口和网络模块以及万能遥控器。

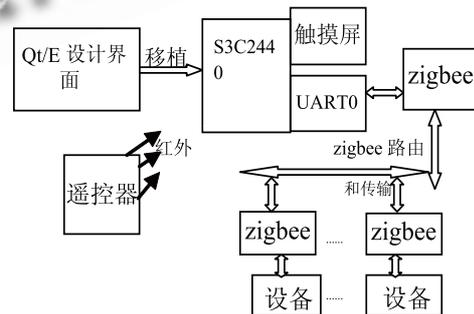


图 1 智能家居设计结构框图

ARM9 微处理器采用 Samsung S3C2440 CPU 开发板, 系统选择嵌入式 linux 系统, 其内核使用

收稿时间:2010-03-13;收到修改稿时间:2010-04-06

2.6.25.8 的内核, nandflash 采用 256M 的内存。且由于 arm 内核集成了多种外设控制模块的驱动程序, 所以要对其进行必要的裁剪, 串口用于连接 zigbee 模块。

Zigbee 采用 TI 公司的 CC24430 无线射频芯片, zigbee 有显著的低成本、低消耗、网络节点多等的无线传输功能, 所以可以将一个 zigbee 作为路由, 通过一个 zigbee 向多个 zigbee 传输的功能, 实现 zigbee 的组网的, 也有效的延长了 zigbee 的传输距离。

网络模块采用 CS8900A, 通过连接路由器, 实现远程登入。

### 3 系统软件设计实现

系统采用 linux 作为操作系统, 首先基于 Qt/Embedded 进行图形界面的设计, 再在宿主机上建立交叉编译环境, 并制作 yaffs 操作系统, 最后将嵌入式 linux 系统移植到 ARM9 开发板上。系统通过设计 GUI 界面, 实现通过触摸屏对智能家居的一体化控制<sup>[1]</sup>。文章软件部分包括嵌入式 linux 操作系统的移植、系统界面设计、串口驱动和 zigbee 组网设计。在 ARM9 和 zigbee 的传输过程中, 有一个智能家居的单元端接口设置。为了区别于不同家庭之间的传输, 首先 ARM9 会有一个各自家庭的主机 ID 号设置。在一个家庭的智能化控制上, 给路由功能的 zigbee 设计八个端口, 每个端口有一个 ID 号, 然后其余的连接路由的 zigbee 通过接受各自 ID 号的分别传输信息, 最后, 这些 zigbee 又分别设置各自的设备号, 这样就去除了干扰, 实现智能家居的单独控制。

#### 3.1 触摸屏和 Qt/Embedded 交叉编译环境的建立

在建立交叉编译环境之前先要进行 linux 系统移植, Linux 系统的移植主要主要包括 BootLoad 移植、内核移植与裁剪以及文件系统移植等部分<sup>[2]</sup>。文章选用 u-boot 作为 BootLoad 引导程序。Linux 内核由于其公开性, 很容易实现其移植和裁剪。文件系统这采用 yaffs 根文件系统。

触摸屏和 Qt/Embedded 的交叉编译环境的建立是 linux 系统移植的关键部分。其步骤如下。

第一步是进行触摸屏校正, 设计中采用 tslib-1.4 作为触摸屏的校正。其分为两步:(1)编译 tslib, 产生校正文件。(2)是下载校正文件和库文件到 arm 板, 实现五点校正。其编译过程如下。

```
#tar xvf tslib-1.4.tar.bz2
#cd tslib-1.4
#./autogen.sh
#./configure--prefix=/usr/local/tslib/
--host=arm-linux
ac_cv_func_malloc_0_nonnull=yes?
--enable-inputapi=no
#make
#make install
```

第二步是 Qt/E 的交叉编译。设计中 Qt/E 采用 qt-embedded-linux-opensource-src-4.5.0。同时为了方便程序开发, 需要安装 X86 版和 ARM 版。先在 PC 上使用 x86 版和 qvfb 开发, 完成以后再使用 ARM 交叉编译放到开发版上运行。所以需要先解压一次, 然后重命名为 qt-embedded-linux-opensource-src-4.5.0-x86; 再解压一次, 并重命名为 qt-embedded-linux-opensource-src-4.5.0-arm。Qt/E 的编译过程如下。

```
# tar xvf qt-embedded-linux-opensource-src-4.5.0.tar.bz2
#./configure-release-shared-fast-qt-zlib-qt-gif-qt-libtiff-qt-libpng-qt-libmng-qt-libjpeg-makelibs-no-make-tools-nomake-examples-nomake-docs-no-make-demo-xplatform-qws/linux-arm-g++-embedded-arm-little-endian-qt-freetype-depths-16,18-qt-gfx-linuxfb-qt-kbd-usb-qt-mouse-tslib-l/home/cnnbboy/opt/include-L/home/cnnbboy/opt/lib
#make
#make install
```

最后, 设置交叉编译环境变量, 进入系统控制平台 vi /etc/profile. 写入环境变量如下:

```
export TSLIB_ROOT=/usr/local/tslib
export TSLIB_TSDEVICE=/dev/input/event0
export TSLIB_CALIBFILE=/etc/pointercal
export TSLIB_CONFFILE=$TSLIB_ROOT/etc/ts.conf
export TSLIB_PLUGINDIR=$TSLIB_ROOT/lib/ts
export TSLIB_CONSOLEDEVICE=none
export TSLIB_FBDEVICE=/dev/fb0
export QWS_MOUSE_PROTO=tslib:/dev/input/event0
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$TSLIB_ROOT/lib
```

```
exportQTDIR=/usr/local/Trolltech/qt-embedded-4.5.0
export PATH=$QTDIR/bin:$PATH
exportLD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
```

### 3.2 Qt/Embedded 编程

打开 Qt Creator, 进入 Qt4 GUI Application, 分别建立了 pro、mai.cpp、dialog.cpp 和 dialog.h 文件<sup>[3]</sup>。智能家居的图形用户界面图 2, 分为如图 2(a) 和图 2(b) 所示, 下面先说明图 2(a)。quankai—全开按钮主要是完成控制屋内所有灯光的打开; quanguan—全关按钮主要是完成控制屋内所有灯光的关闭; woshi—卧室、chufang—厨房、keting—客厅、卫生间—卫生间主要是当按下按钮时, 会进入分别控制它们安装的具体设备的窗口; tianjia—添加、shanchu—shanchu 按钮是当添加新的设备和拆除设备时的控制功能。再介绍图 2(b)。下面设置了一些设备, 如 deng—灯、chuanglian—窗帘、mencimen—门磁、meiqifa—煤气阀、kongti—空调, dianshi—电视、bingxiang—冰箱、xiyiji—洗衣机, 当按下如图 b 的按钮时, 按钮会通过槽输出一个 16 进制, 这个 16 进制将通过串口传输给 zigbee, 然后触发 zigbee 中的控制程序, 实现控制设备的作用。下面对几个重要的连接进行说明。图 2(a) 的说明如下:



(a) 首页效果图 (b) 控制页面效果图

图 2 智能家居界面效果图

(1) connect(PushButton. SIGNAL(clicked()), this, SLOT(open()))

当 PushButton 按钮按下时, zhinengjiaju 对话框接受信号, 并执行 open() 函数, open() 函数中写如程序, 让其进入下一个对话框--zhinengjiaju1。下面是部分程序:

```
#include "zhinengjiaju.h"
#include "zhinengjiaju1.h"
```

```
void zhinengjiaju::open()
{
    zhinengjiaju1 *two=new zhinengjiaju1;
    two->show();
}
```

图 2(b) 的说明如下:

(2) connect(PushButton . SIGNAL(clicked()), this, SLOT(newslot()))

当按下按钮 PushButton 时, zhinengjiaju1 对话框接受一个信号, 使 newslot() 函数发出一个 16 进制。下面是部分程序:

```
#include <iostream>
#include <stdio.h>
using namespace std;
void zhinengjiaju1::newslot()
{
    unsigned char i=0x01; //开
    cout<<hex<<i<<endl;
    unsigned char i=0x02; //关闭
    cout<<hex<<i<<endl;
}
```

### 3.3 串口传输驱动程序

ARM 和 zigbee 主要是通过串口实现传输。界面设计完成后, 通过触摸触发按钮, 使按钮输出出一个 16 进制, zigbee 通过串口驱动接受到按钮传输出的数据, 使 zigbee 中的程序工作。下面是部分串口传输驱动程序源代码。

```
void UART_RXdata(void)
{
    P2SEL&=~0x01; //串口接收 0x01 这个 16 进制
    if((UartFrame[1]== 0x01)&&(UartFrame[2] == 0x0D)&&(UartFrame[3] == 0x0A))
    {
        SET_LED1();
        lcd_clear();
        LCD_write_english_string(0,1,"LED_ON");
        //串口接收到 0x01, 触发灯亮
    }
    Else if((UartFrame[1] == 0x02)&&(UartFrame[2] == 0x0D)&&(UartFrame[3] == 0x0A))
    {
```

```

        CLR_LED1();
//0x02 触发灯关闭
        lcd_clear();

LCD_write_english_string(0,1,"LED_OFF");
    }
}

3.4 zigbee 组网设计
    Zigbee 的组网主要是由 zigbee 的协调器和向 zigbee 的节点相互传输组成的。主要是用一个 zigbee 作为路由,向这个 zigbee 下载协调器的程序;再向其余的 zigbee 下载节点程序,从而实现一个 zigbee 控制多个 zigbee 的传输过程。以下是部分的智能家居所有编写的的 zigbee 的组网应用程序。
static void  zb_BuildNet(void); //建立网络
static void  Choose_ControlTpye(void);
//选择系统控制模式:手动/自动
static void  Auto_Open_Light(void); //点亮灯
static void  Auto_Close_Light(void); //熄灭灯
static void  Auto_Open_Curtain(void); //开窗帘
static void  Auto_Close_Curtain(void); //关窗帘
static void  Auto_Open_Condition(void); //开空调
static void  Auto_Close_Condition(void); //关空调
static void  Process_Coor_Cmd( uint8 show_type );
//处理协调器命令
static void  Close_Alarm(void); //关闭报警器
static void  Process_Temp( float temp );
//自动控制状态下温度处理
static void  Process_Illuminance( uint8 illuminance);
//自动控制状态下照度处理
static void  Show_CoorInfo(void);

```

```

//显示协调器信息
static void  Show_Temp(void); //显示温度
static void  Show_Illuminance(void); //显示照度
static void  Show_CurtainState(void);
//显示窗帘的状态
static void  Show_ConditionState(void);
//显示空调的状态
static void  Show_WlecomeInfo(void);
//显示门迎信息
static void  Show_InbreakInfo(void);
//显示防盗器信息
static void  Show_GasInfo(void);
//显示煤气检测信息

```

#### 4 结 语

在智能家居的设计中,一个友好的系统界面(GUI)是非常必要的,友好的 GUI 界面更加方便用户的使用,也促进了系统的推广。通过 ARM 和 zigbee 对智能家居的控制,很有效的实现无线传输控制功能,且由于 ARM 可以对万能遥控器识别码的学习,可以很容易的设计出一种配合 ARM 和 zigbee 的遥控器,更家的完善了对智能家居的无线控制功能,更大的适应广大用户的需求。

#### 参 考 文 献

- 1 Blanchette J. C + + GUI Programming with Qt 4. Trolltech AS. 2004.
- 2 张方辉,王建群.Qt/Embedded 在嵌入式 linux 上的移植.计算机技术与发展,2006,16(7):65 - 66.
- 3 张娟,张雪兰.基于嵌入式 linux 的 GUI 应用程序的实现.计算机应用,2003,23(4):115 - 117.