

# 基于嵌入式 Linux 的智能家庭原理与实践

## The theory and implementation of embedded Linux – based smart home

赵卫星 李敏 董溯前 (东华大学信息科学与技术学院 上海 200051)

**摘要:** MisterHouse 是基于 X10 协议的可通过 Web 控制家电的自动化服务器软件。Geebox 是采用 LZMA 算法压缩的适合家庭娱乐的嵌入式 Linux 发行版。本文分析了 X10 协议,首次集成 MisterHouse 到 Geebox 中,并对 Geebox 进一步优化,初步搭建出集娱乐、管理、控制为一体的智能家庭操作系统,提供了一种基于 Linux 的家庭自动化实现方案。

**关键词:** MisterHouse Linux 智能家庭 X10

### 1 引言

Linux 是 Linus Torvalds 于 1991 年以开源形式公布的类 Unix 操作系统。经过世界各地计算机爱好者的努力,现已成为十分稳定和高效的操作系统。Linux 以其优异的稳定性、安全性和低廉的成本在全世界迅速发展,使用人数越来越多。目前主要的商业或开源发行版有: Fedora, Ubuntu, SUSE 等。还有很多嵌入式 Linux 发行版例如: Geebox, Damn small linux, uClinux 等。

家庭自动化和网络化是信息技术发展的趋势。1. 家电的功能越来越多、越来越复杂,家电种类越来越多;2. 家中的计算机越来越多;3. 家中各种网络如电话网、有线电视网、计算机网及未来的智能家电网络。所有这些都需要一套高效、安全、稳定的操作系统来统一管理,这些互相独立的“信息孤岛”必将组成一张全功能的智能家庭网络。基于 Linux 的智能家庭是 Linux 在家庭自动化中的一个新应用方向。

#### 1.1 Geebox

本文采用的嵌入式 Linux 发行版是 Geebox。Geebox 是开源的系统,大小只有 8.4MB,可从光驱启动,也可安装到硬盘上。它本身就是一个适合家庭娱乐的 Media center,可以播放电影、音乐,浏览照片等。它的核心部分文件用 7-zip 的高级压缩算法 LZMA 来压缩,使 Geebox 变得很小,很适合用在较低配置的末来家庭控制器(如低端的 PC)上;当然,在现在主流的 PC 上能更快速地运行。

#### 1.2 MisterHouse

MisterHouse 是为控制 X10 而写的基于 Linux 的开源家庭自动化软件。在 Linux 的开源世界,有很多类似项目,如: MisterHouse, Jhome, Heyu 等;其中 MisterHouse 项目成立于 1998 年,它的代码用 Perl 语言编写,现已发展得比较成熟。主要特点有:(1) 用户控制界面基于 Web,可远程控制家电。(2) 基于 X10 协议。(3) 语音控制。安装相应的语音识别软件后,可用语音来控制各种家电。(4) 用 MisterHouse 不光可以控制家里的电灯,摄像头,喷泉,电视机等各种电器;还可以打视频电话,收发 Email,看天气预报,看电视,看电影,听音乐,浏览照片,管理约会等。用作者 Bruce Winter 的话说:MisterHouse 能做的事情取决于你的想象力[1]。

### 2 X10 协议和系统原理

#### 2.1 X10 协议

X10 是国际通用的智能家居电力载波协议。它是第一个利用电力线来控制电子电器设备的标准。电力线使用 X10 协议在提供电流的同时可像网线一样将控制命令传送到被控制的家电上,从而实现网络化的控制。

X10 是以 50Hz(或 60Hz)为载波,将 120kHz 音频信号限制在一个很小的带宽内,形成一个音频脉冲,然后以 120kHz 调变信号在每一正弦波周期上有和无和交流电信号的过零点检测很好地结合在一起,再加以调

制形成良好的方波电平信号,即电源线调变信号取代数据 0 与 1,由此在一个限定个数的周期内得到了一个简单的数字信号组合,通过对该数字信号组合进行规范,形成了一个简单的通信协议。音频信号通过耦合变压器加载到电力线上从而将控制指令通过电力线传送到目的地[2]。

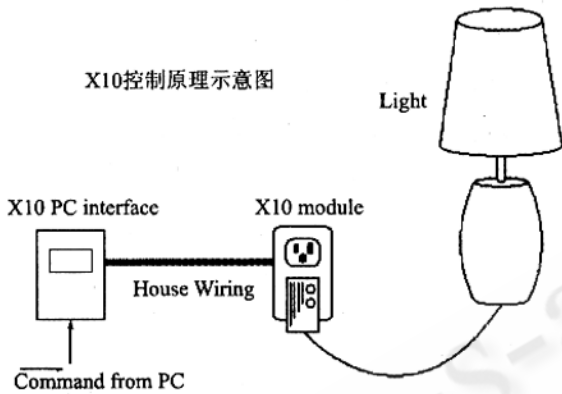


图 1 X10 控制原理图

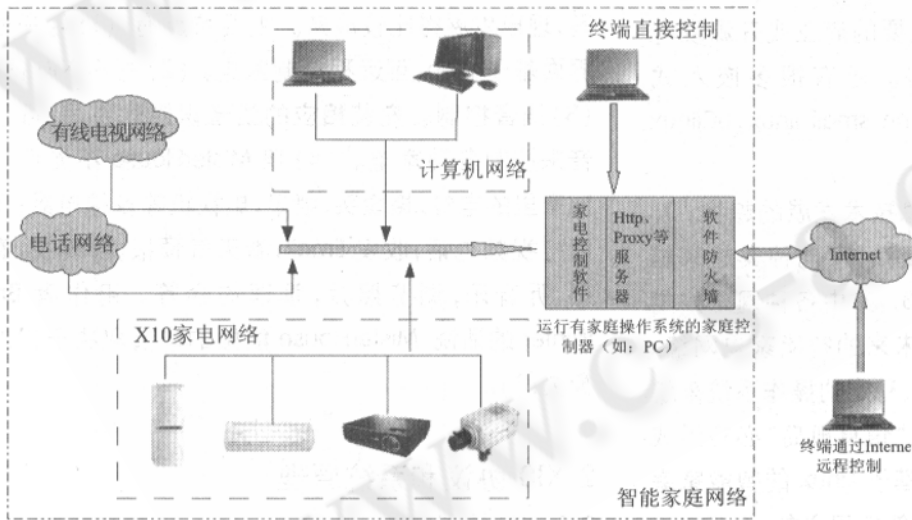


图 2 本文系统原理图

图 1 为本系统所用的 X10 控制原理图。控制信号从运行着支持 X10 协议软件 (MisterHouse) 的 PC 上发出,通过与 PC 相连的 X10 PC 接口模块,将信号通过电线发送给信号中指定的模块来控制电灯。其中:(1) X10 PC 接口模块使用的是 CM11A,它是一个 Transceiver,即可以接收和发送 X10 信号,用 RS232 口和 PC 相连。(2) X10 接收器 (Receiver) 模块,通常被连接到墙

壁上的电源插座或者嵌入到电器的开关中,它们会监听接收控制信号并做出必要的响应,将对应的灯具和家电开启或关闭 (on/off)。本文使用的是 AM486,这是一种只能控制开/关的简单 X10 模块。(3) 交流电线即家中常见的 220V 电线,主要是将信号从 X10 PC 接口模块通过电线送至连接有灯具等家电的 X10 接收器模块。[3]

### 2.2 系统原理

图 2 为本文的系统原理图。如图所示,本文只针对智能家庭网络中的 X10 家电网络,来构建家庭控制器上的小型家庭操作系统。在此操作系统中集成家电控制软件 - MisterHouse,使得用户可通过本地和远程 (Internet) 的终端来控制家电。[4]

## 3 Linux 智能家庭实践

### 3.1 系统软硬件环境

表 1 为实现本系统所用的软件和硬件环境。本文采用普通 PC 作为本系统的硬件平台。

### 3.2 实现原理和实现步骤

通过解压分析 initrd (Geexbox 的 initrd 文件采用 ext2 文件系统做成 image 格式,并用 gzip 压缩) 中的 linuxrc 文件,知道系统启动的过程中会解压 bin. tar. lzma、建立文件系统的其它目录,并将这些东西复制进内存,以建立 Linux 根文件系统,最后要运行 /sbin/init; 解读 /sbin/init 文件 (注:此文件为 Geexbox 中的 /sbin 目录中,而不是工作平台的 /sbin 中) 知道系统将依次执行 /etc/init. d 中的脚本,想让 MisterHouse 在启动时运

行,须在此目录中建立相应的脚本。

通过分析 Geexbox 系统原理,本文提出下列两种方案。

方案 1: MisterHouse 直接放在 Geexbox 的 /usr/share 中

实施过程中发现:由于 Geexbox 光盘启动采用

isolinux 技术,其最多支持的文件夹层次为 6 层。所以此方法不行。

方案 2: MisterHouse 集成入 bin. tar. lzma 文件

由于 LZMA 压缩算法,可大大减小最终系统的大小;另外,这一方案解决了方案 1 中的 isolinux 支持目录限制问题。本文采用这一方案。

表 1 系统软硬件环境

	类别	名称	版本/参数
软件环境:	工作平台操作系统	openSUSE	10.2
	工具软件	mkisofs	2.01.01
		LZMA	4.42
		Perl	5.8.8
		Geexbox	1.1rc1
		MisterHouse	2.103
硬件环境:	工作平台 PC	CPU	2.1GHz X86
		内存	512MB

实现具体步骤:(注:本文使用的命令均在 root 权限下执行)

### (1) 加载 Geexbox

加载 .iso 文件到临时目录。

```
#mount -o loop geexbox-1.1rc1-en.i386.iso /mnt/temp/
#ls
```

可看到其中有文件夹 boot, codecs, etc, firmwares, sbin, usr, var 和文件 bin. tar. lzma。其中文件 bin. tar. lzma 中是使用 LZMA 高效率压缩算法压缩的 lib, usr/bin 和 usr/lib 目录。因为这三个目录比较大,所以 Geexbox 使用了 LZMA 来压缩,使得 Geexbox 只有不到 9MB,这对硬件资源有限的嵌入式应用无疑是非常有利的。同时,这也是本文将 MisterHouse 也集成入此文件中的原因。

复制 Geexbox 全部文件到工作目录(/mnt/workspace),并复制 bin. tar. lzma 到临时目录,作修改、剖析用:

```
#cp -rf /mnt/temp/* /mnt/workspace
#cp /mnt/temp/bin.tar.lzma /tmp/bin-with-mh
```

### (2) 安装 lzma

lzma 是用来生成和解压 lzma 格式的压缩文件的,可到 [www.7-zip.org](http://www.7-zip.org) 下载 lzma-sdk;解压后,进 C/7zip/Compress/LZMA\_Alone 目录,运行 make -C ./ -f makefile.gcc 编译后将 lzma 复制到工作平台(openSUSE-10.2)的/usr/bin中。

```
#tar jxvf lzma-4.42.tar.bz2
#cd lzma-4.42/C/7zip/Compress/LZMA_Alone
#make -C ./ -f makefile.gcc
#cp lzma /usr/bin
```

### (3) 解压 bin. tar. lzma

安装软件 lzma 后,就可用下列命令来解压 bin. tar. lzma 文件,解压后得到 lib 和 usr 两个目录。

```
#cd /tmp/bin-with-mh
#lzma d bin.tar.lzma bin.tar
#tar xf bin.tar
```

### (4) 安装 Perl

Perl 是一种很适合网络应用的语言, MisterHouse 是用 Perl 编写的,在运行 MisterHouse 时系统中要安装 Perl,但在 Geexbox 中没有 Perl,所以要自己安装。从 Perl 官方网站 [www.perl.org](http://www.perl.org) 下载最新版 perl-5.8.8-10.tar.bz2。

编译、安装 Perl 如下。

```
#tar jxvf perl-5.8.8-10.tar.bz2
#cd perl-5.8.8
#./Configure -Dprefix=/usr
#make install DESTDIR=/tmp/bin-with-mh
#rm -rf /tmp/bin-with-mh/usr/man
```

### (5) MisterHouse 集成

Geexbox 中的 runlevels 分为: geexbox, debug, install 三个。分析/sbin/init 发现 runlevel 应该在 init.d 目录脚本中的"# runlevels:"后面设置。编写 shell 脚本 83\_misterhouse.sh 如下:

```
#!/bin/sh
# runlevels: geexbox, debug, install
/usr/mh/bin/mh&
exit 0
```

修改 83\_misterhouse.sh 权限,使其有可执行权

限,并复制到 Geebox 的/etc/init.d 中。

```
#chmod +x 83_misterhouse.sh
#cp 83_misterhouse.sh /mnt/workspace/GEEXBOX/etc/init.d
```

将 misterhouse 的 tar.gz 文件解压,复制到解压 bin.tar.lzma 得到的 usr 中。

```
#tar xzvf misterhouse-2.103.tar.gz
#cp -rf misterhouse-2.103/tmp/bin -with -mh/usr/mh
```

#### (6) 完善 Geebox

添加 Mplayer 的 win32 codecs:到 www.mplayerhq.hu 下载相关的 codecs,解压后复制到/mnt/workspace/GEEXBOX/codecs 即可。

#### (7) 生成新的 bin.tar.lzma

```
#tar cf - lib usr/bin usr/lib usr/mh | lzma e bin.tar.lzma -si -o2
```

#### (8) 生成新的 iso 文件

```
#mkisofs -quiet -p "The GeeXboX team (www.geebox.org)"
-o /home/neil/Geebox+Misterhouse.iso -no-emul -boot -
boot-info-table -boot-load-size 4 -b boot/isolinux.bin -c
boot/boot.catalog -D -r -J /mnt/workspace
```

#### (9) 测试

可将生成的 iso 文件刻录到光盘,并从光驱启动。按 F1,输入 debug 进入 debug 模式。启动完毕后,可在 Web 浏览器 (IE/Firefox) 地址栏输入运行 MisterHouse 的 PC 的 IP 地址,就会出现如图 2 的基于 Web 的用户控制界面。也可在 VMware for Linux 中用同样方法测试。

### 3.3 系统运行

#### 3.3.1 系统启动

系统启动时,最先调用 isolinux.bin,此文件是 isolinux 技术中的核心程序,它会分别在光盘的/boot/isolinux、/isolinux 及根目录等目录下查找 isolinux 的配置文件 -isolinux.cfg。isolinux.cfg 文件中指定了 Linux

内核映像文件 vmlinuz 和 initrd.gz 文件的位置。找到后,将它们复制到内存中,并运行 Linux 内核 vmlinuz, vmlinuz 会解压和调用 initrd.gz,initrd 加载驱动模块,生成系统最终的文件系统,它最后会调用/sbin/init。/sbin/init 负责系统最终应用程序及系统服务 (Daemon) 的加载运行。系统启动顺序如下:

isolinux.bin (读 isolinux.cfg) -> vmlinuz (调用 initrd) -> initrd.gz (运行 linuxrc) -> /sbin/init (解压 bin.tar.lzma) -> mplayer, misterhouse 等应用程序或系统服务。

#### 3.3.2 用户控制界面

用 Firefox2.0 打开 MisterHouse 的用户控制界面,通过此界面,用户可安全地控制系统内的家电。

## 4 结论

本文提供了一种基于 Linux 的家庭自动化解决方案。分析了 X10 协议,将 MisterHouse 集成到 Geebox 中,对 Geebox 进行优化,初步完成了一个集 Geebox 娱乐功能和 MisterHouse 控制管理功能为一体的家庭操作系统。本文使用的软件 (主要是 MisterHouse 和 Geebox) 都是高性能的开源软件,使得构建出来的系统成本低,性能高。

虽然已经取得初步成功,但还需要进一步优化,如安全方面加入 Firewall;及时更新 MisterHouse;加强硬件驱动支持;加入更多的媒体解码器等。

#### 参考文献

- 1 Bruce Winter, MisterHouse Docs, misterhouse project, 2005.
- 2 杨琳, X10 协议与智能家居, 中国多媒体通信, 2005 -9.
- 3 Neil Cherry, Linux. Smart. Homes. For. Dummies, Wiley Publishing, Inc. Aug. 2006, 248 ~ 300.
- 4 Aurelien Jacobs, Geebox README, geebox.org, 732 ~ 784.