《嵌入式系统工程实践》

实验报告

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **组号** | **学号** | | **姓名** | | **学院** | 软件学院 | | |
| 9 | 55190506 | | 王思扬 | | **学期** | 2021－2022 | | |
| 55190525 | | 王泽卿 | | **专业** | 软件工程 | | |
|  | |  | | **递交日期** | | 2022.12.01 | |
| **题目** | 电子广告牌系统设计 | | | | | | | |
| **满分** | | 30分 | | **得分** | | | |  |

1. 需求分析
2. 功能性需求:

电子广告牌系统将实现的基本功能包括：

a、能够通过在PC机上的配置客户端（以下简称客户端）配置当前滚动显示广告语的内容，并使得发光LED能够根据给定的内容滚动显示广告语。

b、能够通过客户端配置当前滚动显示广告语的滚动方向，并使得发光LED能够根据给定的显示方向滚动显示广告语。

c、能够通过客户端配置当前滚动显示广告语的速度，并使得发光LED能够根据给定的显示速度滚动显示广告语，

d、能够通过客户端暂停广告语的滚动。

e、能够通过客户端实现对暂停广告语的重新播放。

f、能够在客户端中添加预设广告语（以实现快捷设置显示广告内容）。

g、能够在客户端中删除预设的广告语。

1. 运行环境需求：

系统的运行环境需求如下：

a、系统在室外及室内环境下均可运行。

b、系统要求环境温度介于-10℃~45℃之间。

c、系统要求环境湿度介于30%~80%之间。

d、系统要求环境中不能有强电磁干扰。

e、系统要求环境中不能有大量扬尘。

1. 数据需求：

系统产生的数据主要有广告语的内容（字符串形式）、广告语的显示速度（编码形式）以及广告语的显示方向（编码形式），其由用户输入，并从配置客户端传递至开发板上运行的服务器程序，并由服务器程序以文件形式存储在开发板上以便于LED驱动程序后续使用；系统使用的数据包括字符编码数据库（文件，存储字符与发光LED上编码的对应关系）。

1. 系统需求：

系统要求联网，以实现在配置客户端上远程对发光LED的显示内容、显示速度以及显示方向进行配置。

1. 应用需求：

用户可通过在任意PC上运行配置客户端实现对发光LED显示广告语的内容以及滚动方向和速度进行配置。用户首先在配置客户端上输入开发板的ip地址，随后输入期望显示的广告内容并选择期望的显示速度与滚动方向，最后点击设定运行按钮，所输入的内容即可在发光LED上按照给定的速度与方向进行滚动显示。

此外，用户可以通过按下开发板上的按钮对系统进行复位，使其按照预设配置显示预设的内容。

1. 约束需求：

系统使用了pThread实现多线程，以及Qt实现配置客户端的GUI，因此应具备相应的依赖库。

1. 业务流程需求：

a、在发光LED上显示广告：用户首先在配置客户端上输入开发板（发光LED）的ip地址（必要步骤，之后省略），随后输入期望显示的广告内容并选择期望的显示速度与滚动方向，最后点击设定运行按钮，所输入的内容即可在发光LED上按照给定的速度与方向进行滚动显示。

b、对系统进行复位：用户按下在开发板上的按钮，即可完成对系统的复位。

c、暂停广告语的滚动：用户在配置客户端上选择“暂停”,之后应用变更，即可实现对广告语的暂停。

d、恢复暂停广告语的滚动：用户在配置客户端上选择一个期望的速度(1x~3x)，之后应用变更，即可恢复暂停广告语的滚动。

e、变更广告语的滚动方向：用户在配置客户端上选择一个期望的滚动方向，之后应用变更，即可实现对广告语滚动方向的修改。

f、添加预设广告语：用户在配置客户端上输入广告语，并点击添加按钮，即可实现对添加预设广告语。

g、删除预设广告语：用户在配置客户端上选中一条预设广告语，随后点击删除按钮，即可实现对预设广告语的删除。

1. 软硬件开发技术（技术理论部分、编程实现部分）

2.1 界面编程技术

2.1.1 技术简介

界面采用Qt实现。界面负责与用户直接进行交互，需要能够直观的显示、反馈用户的操作效果，该部分需要完成的任务如下：

1. 存储用户设定的广告语，并支持添加删除功能。
2. 根据用户当前设定信息，通过Socket通信，向服务器发送广告语运行配置，包含速度、方向、显示内容。

2.1.2 编程技术

2.1.2.1 文件存储与读取

界面采用一个文本文件来管理广告语。用户添加的广告语将会保存在该文件中，而界面当中的广告语列表将会从该文本文件中逐行读取。

1. 写入文件操作步骤：

调用fstream函数创建输入流。

按行从界面的widgettable中读入广告语，并采用输入流将其输入到文件当中。

1. 读入文件操作步骤：

调用istream函数创建输出流，读取设定好的文件（如果不存在，则创建空文件）。随后按行输出文本中的广告语，并将其添加到界面的表格当中。

2.1.2.2 Socket通信

界面端Socket通信与服务器端不同之处在于，界面通信只需要处理发送即可，不需要考虑接收信息。对于Socket发送信息来说，其操作如下：

1. 根据输入IP以及约定端口创建Socket连接。
2. 根据约定协议构建发送内容。
3. 将发送内容从文字转为字节流。
4. 将字节流通过Socket发送至服务器。

2.2 服务器编程技术

2.2.1技术简介

消息接收与程序调用服务器：作为系统中间层，该部分需要完成的任务如下：

（1）首先，该中间层使用网络通信技术与配置端建立连接，并持续监控配置端消息；

（2）当配置端消息传来，按照约定的解码格式对消息进行解码，从而获取启动点阵发光程序需要的参数；

（3）当获取到新的配置参数后，该中间层会检查点阵发光程序是否被启动，若没有启动，则使用多线程启动程序，否则杀死已经被启动的线程，根据当前配置参数重启一个新线程。

2.2.2编程技术

2.2.2.1 Socket通信

（1）对于服务器，其通信流程有如下步骤：

调用socket函数创建socket，这一步会创建一个文件描述符FD。

调用bind函数将socket（也就是FD）绑定到某个ip和端口的二元组上。

调用listen函数开启侦听端口。

调用accept阻塞等待接受连接，当有客户端请求连接上来后，产生一个新的socket（客户端socket）。

基于新产生的socket调用send或recv函数开始与客户端进行数据通信。

通信结束后，调用close函数关闭客户端socket。

（2）对于客户端，其通信流程有如下步骤：

调用socket函数创建客户端socket。

调用connect函数尝试连接服务器。

连接成功以后调用send或recv函数开始与服务器进行数据通信。

通信结束后，调用close函数关闭socket。

2.2.2.2 多线程

（1）当有新配置消息到来时创建线程启动显示程序

int pthread\_create(pthread\_t \* tid, const pthread\_attr\_t \* attr, void \* ( \* func) (void \* ), void \* arg);

（2）点阵显示线程从内部结束线程

void pthread\_exit (void \*status);

（3）当有新配置消息到来时终止正在运行的点阵显示线程

int pthread\_kill(pthread\_t threadId,int signal);

2.3发光LED驱动编程技术

2.3.1技术简介

该程序通过完成如下任务以控制点阵LED按照预定的配置显示广告语：

首先，该程序从文件（从配置客户端中接收并由后端服务器写入开发板上）中读取广告语的内容（字符串形式）以及广告语的显示配置（滚动速度，滚动方向）。

之后，该程序查字符编码数据库文件（预先写在开发板上）得到字符在发光LED上的显示编码。

最后，根据配置的显示速度与方向执行相应算法，向开发板Linux系统中代表点阵LED的特殊文件中按照给定的配置写入每一个显示图案的编码，控制广告语在发光LED上的滚动显示。

2.3.2编程技术：

延时：通过传入函数参数数字的大小来控制jmdelay函数中循环的执行次数，从而实现可控时长的延时。

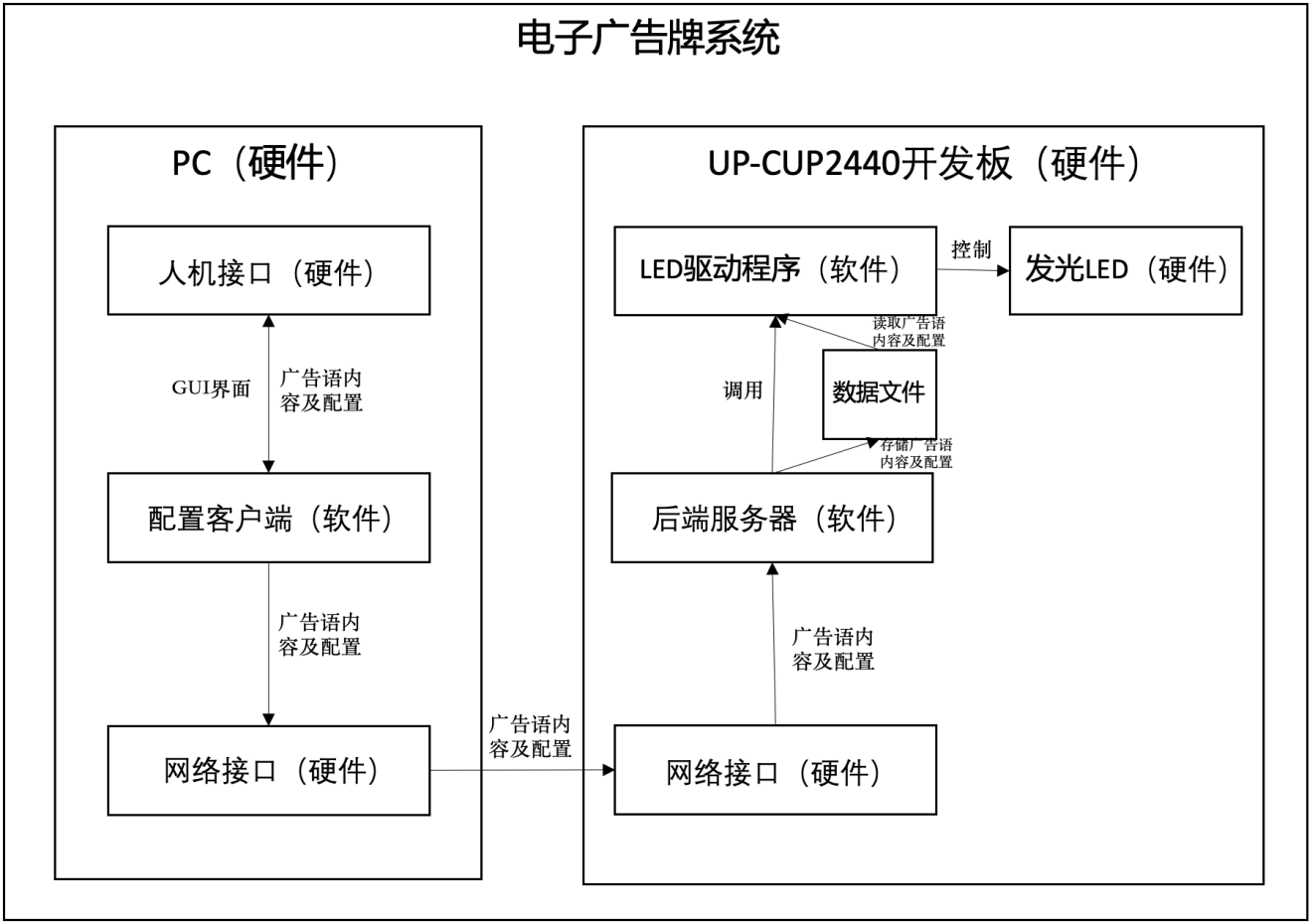
读取文件：通过open函数打开文件得到文件描述符，并执行read函数实现对文件的读取。

控制发光LED：通过open函数打开表示LED的特殊文件得到文件描述符，执行write函数对其写入需显示的图案编码，也即控制发光LED的显示。

打印状态或错误信息：使用printf语句向控制台打印状态或错误信息。

1. 应用模型（图示、说明）

系统应用模型与数据流向如图所示，实现的功能为通过配置客户端获取广告语的显示内容及配置，并在发光LED上显示。



系统硬件由PC及UP-CUP2440开发板两大模块组成，其中PC模块硬件部分又包含人机接口及网络接口；UP-CUP2440开发板模块硬件部分又包含网络接口及发光LED。系统软件由配置客户端、后端服务器以及LED驱动程序三大模块组成，其中配置客户端运行在PC模块上，后端服务器及LED驱动程序运行在UP-CUP2440开发板上。

PC模块中的人机接口模块主要负责与用户的交互，通过键盘鼠标采集用户的输入，并通过显示器输出信息给用户。

PC模块中的配置客户端模块主要负责提供友好的GUI，通过用户的输入，得知需要显示内容及配置；同时通过网络接口模块将显示内容及配置发送给UP-CUP2440开发板。

PC模块与UP-CUP2440开发板模块中的网络接口模块主要负责PC与UP-CUP2440开发板之间的通信，传输的数据为用户通过配置客户端配置的广告语显示内容及配置。

UP-CUP2440开发板模块中的后端服务器模块主要负责通过网络模块接收PC模块上运行的配置客户端模块发送的广告语显示内容及配置，并将其存储为文件以备LED驱动程序的使用；同时，在适当时候调用LED驱动程序控制发光LED对广告语的显示。

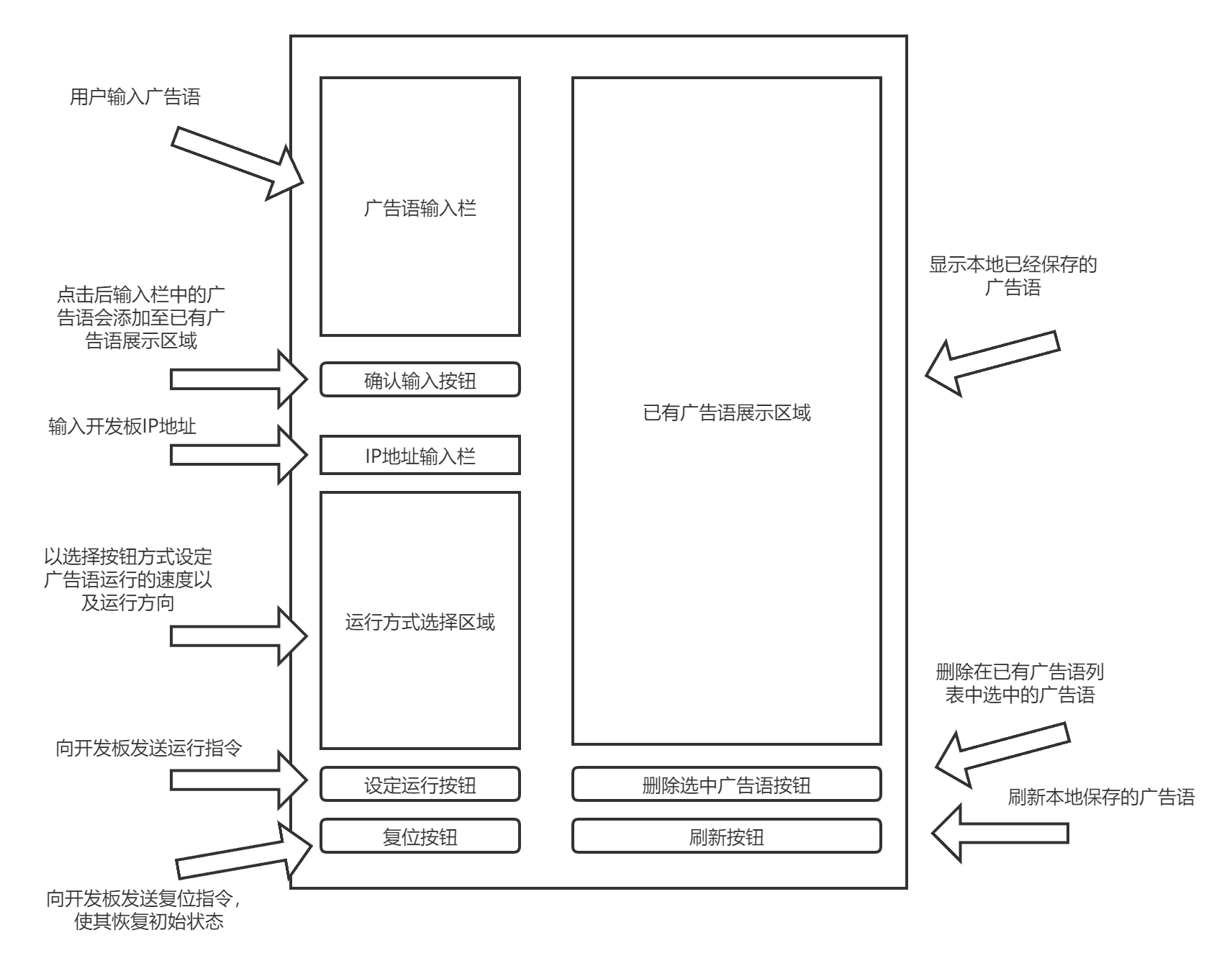
UP-CUP2440开发板模块中的LED驱动程序模块主要负责读取后端服务器模块存储的广告语内容与配置文件，并根据其控制发光LED滚动显示广告语。

UP-CUP2440开发板模块中的发光LED模块主要负责广告语的显示。其受LED驱动程序的控制，根据给定广告语显示内容及配置滚动显示广告语。

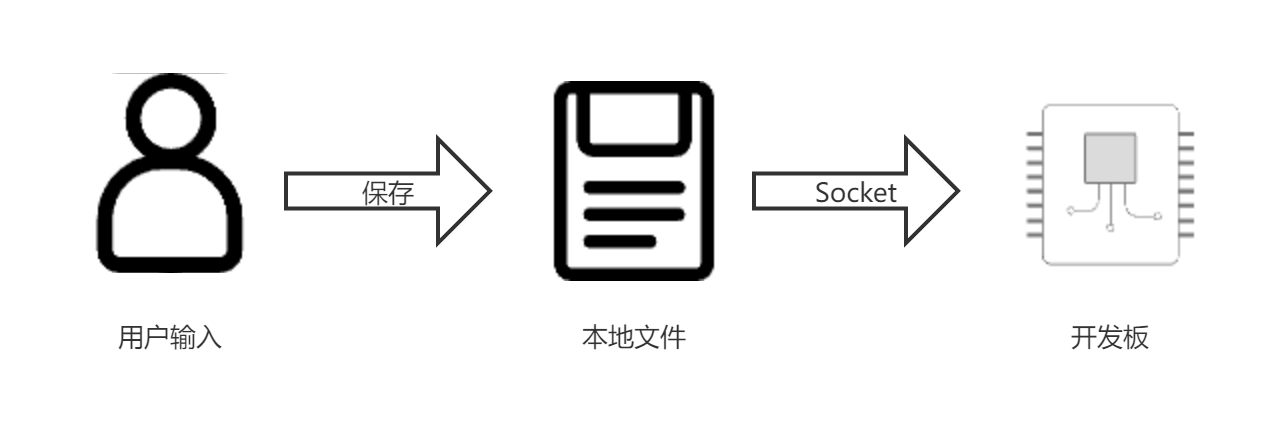
1. 界面设计

界面设计以及各个模块功能如下图所示。

界面的具体实现由Qt实现。其实现效果在之后报告中会进行阐述。

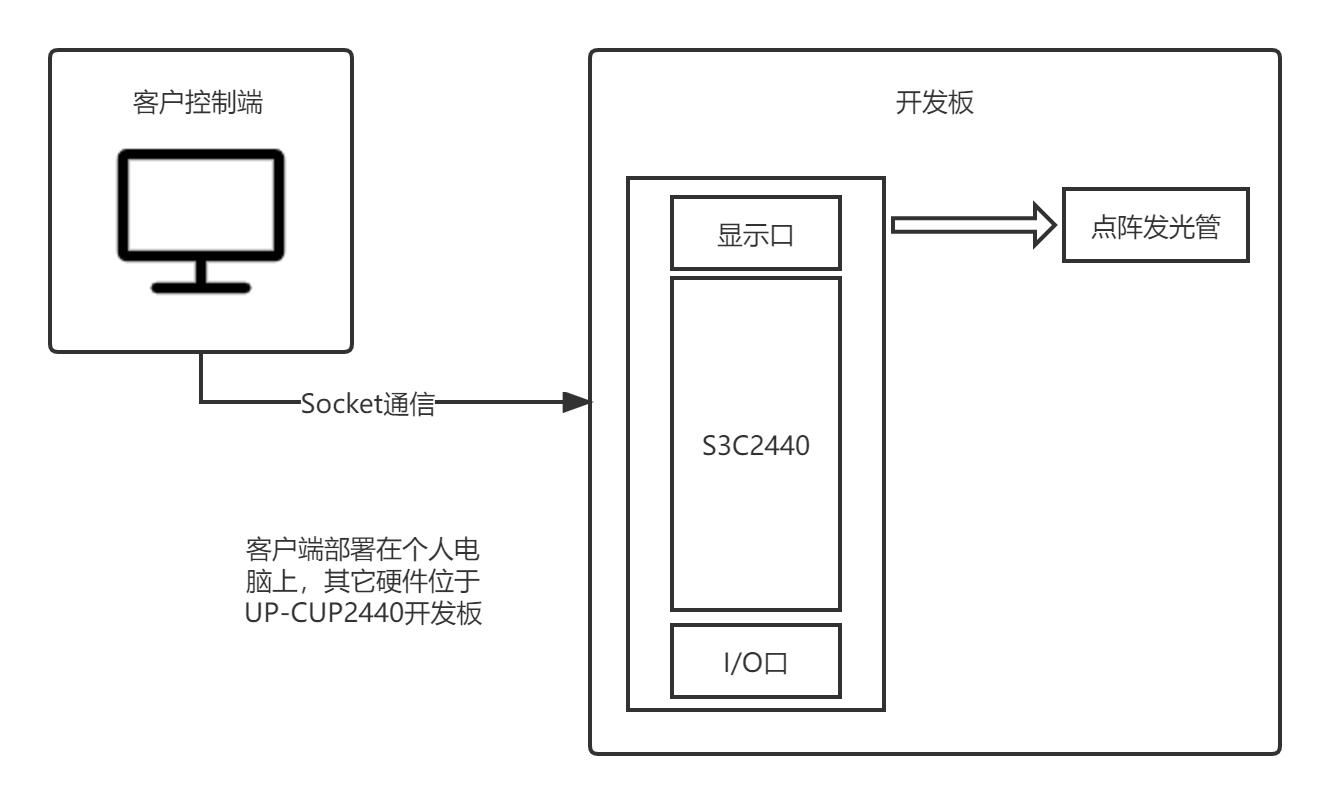


在界面设计中，界面负责将广告语从用户传递到开发板上，并将其运行。这一消息的消息流向图如下图所示。



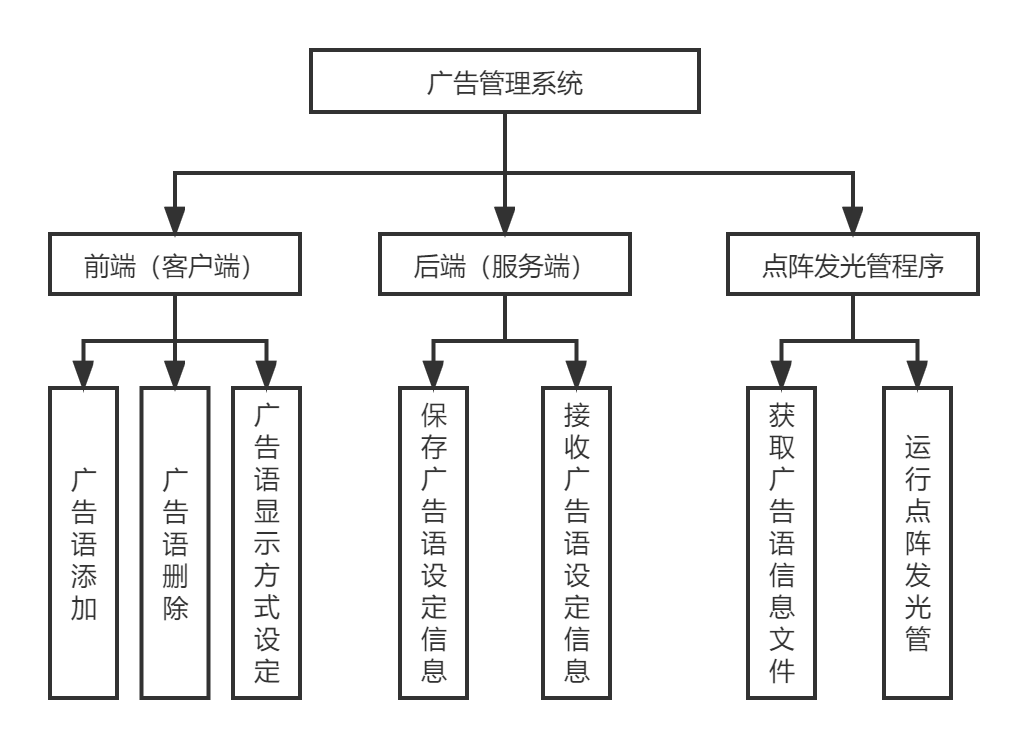
1. 硬件设计（系统硬件模块图、以及硬件模块说明）

硬件设计由两部分组成，分别是用户用于控制UP-CUP2440的客户端电脑，以及用于运行服务端程序以及显示广告语的UP-CUP2440开发板。其硬件模块如下。



1. 软件设计(软件模块组成、以及模块功能说明)

软件设计可以分前端，即用户界面程序、后端，即服务器以及点阵发光管程序，即控制点阵发光管运行。其软件模块图如下。



1. 开发计划：

1、时间：第3次课

任务：基本划分软件各个模块，进行分工。

人员：

程玮斌：编写LED驱动程序

王泽卿：编写Qt界面

王思扬：编写后端通信程序

2、时间：第4次课

任务：分工编写各个模块程序。

人员：

程玮斌：编写LED驱动程序

王泽卿：编写Qt界面槽函数

王思扬：编写后端多线程程序

3、时间：第5次课

任务：分工调试完善各个模块程序。

人员：

程玮斌：调试完善LED驱动程序。

王泽卿：完善Qt界面槽函数

王思扬：完善后端多线程控制。

4、时间：第6次课

任务：完成基本各模块主主体编写工作，开始进行系统集成

人员：

程玮斌：调试完善LED驱动程序，编写复位程序

王泽卿：调试与后端的网络通信

王思扬：调试与前端通信以及驱动程序之间的交互

5、时间：第7次课

任务：基本完成系统集成。

人员：

程玮斌：review调试各模块代码，指导进行系统集成工作。

王泽卿：完善Qt界面逻辑。

王思扬：完善后端线程控制部分。

团队：进行系统集成。

6、时间：第8次课

任务：完成系统集成，提交检查，书写实验报告

人员：

程玮斌：辅助组员调试模块代码，书写实验报告第一、三、八节，第二、七节LED驱动程序部分。

王泽卿：继续完善前后端通信以及界面逻辑，书写实验报告第二、七节以及使用说明书。

王思扬：完善后端线程控制部分，书写实验报告第二、七节以及使用说明书。

团队：收尾系统集成，分工书写实验报告。

1. 程序设计

7.1 界面程序设计

7.1.1 问题分析

界面承担系统与用户的交互的责任，该部分需要能够实现与用户之间良好的交互。因此该部分的设计目标是实现一个简洁、易用、完备的用户界面。

7.1.2 程序设计

界面程序可以分为如下四个部分：

1. 广告语添加。

在TextEdit中输入想要保存的广告语，点击添加广告按钮，即可将其保存至本地，并且添加至右侧的已有广告语展示栏。

1. 已有广告语展示。

界面右侧的广告语展示栏会展示当前保存在客户端的广告语，并根据用户的添加、删除实时更新。

1. 广告语删除。

用户选中广告语展示栏中的广告语，随后点击删除选中广告语按钮，即可删除该广告语。

1. 设定广告语显示配置。

在界面左下方，用户通过选中按钮的方式来设定广告语展示的方式，设定完毕后点击设定运行，即可将配置发送到服务器端。

其界面如下图所示。



7.2 服务器程序设计

7.2.1问题分析

服务器承担着系统中间层的责任，该部分需要完成的任务主要是两个方面，首先是与显示配置端进行消息的交互，然后根据交互过程中得到的指令控制发光点阵的显示内容与显示方式。对于与配置端的交互，由于点阵只有一个，所以消息交互方式只用考虑一对一通信。通信的内容是更新的显示内容或显示方式，需要设计规则进行区分。当有新的配置信息传来时，点阵的状态需要根据配置信息变化，因此需要考虑怎样让运行中的点阵发光程序更改显示内容或显示方式。

7.2.2 程序设计

1. 消息交互

采样socket进行通信，服务端启动后等待配置端连接，当连接建立后就一直循环监听消息，当连 接断开时，若点阵仍有显示，则维持直到手动关闭，若点阵已关闭，则服务端程序也退出。

1. 点阵显示更新

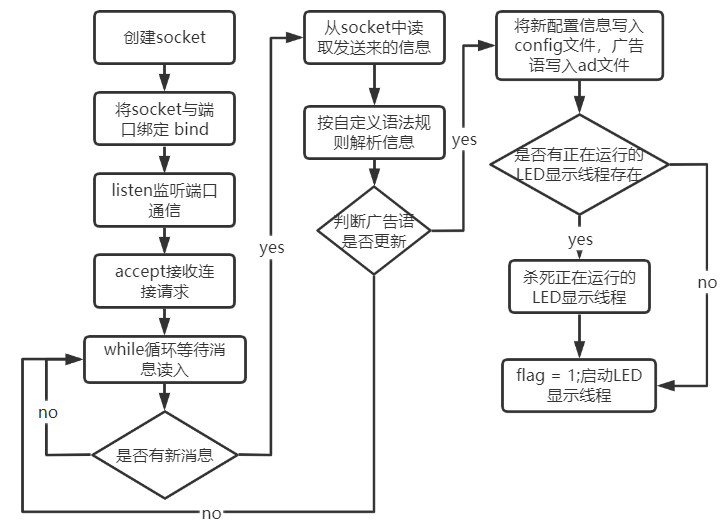
首先对接收到的配置消息按一定格式进行分割，分割规则如下：

收到消息的格式为{dir}${speed}${cttldx}${ad},其中dir为广告语运行方向，speed为广告语运行速度，ad为需要显示的广告语，以$作为分隔符进行分隔，按照这个规则分割得到的配置参数保存在config文件中，得到的广告语保存在ad文件中，这两个文件供点阵 发光程序使用。其中config文件存储格式如下：

Dir: [ ] \n

Speed: [ ] \n

配置信息保存在文件中后，开启线程调用点阵发光程序。需要注意，此时需要判断点阵发光程序 是否第一次被调用，若为第一次调用，则直接启动新线程，线程中运行点阵发光程序；若已经有 点阵发光程序在运行，则需要先杀死正在运行的线程，然后启动新线程，让点阵更新显示内容。



7.2.3 程序编写

程序编写时涉及到的关键函数和用法如下：

7.2.3.1. 通信

（1）socket

int socket(int domain, int type, int protocol);

* domain：指定通信协议族。常用的协议族有AF\_INET、AF\_UNIX等，对于TCP协议，该字段应为AF\_INET（ipv4）或AF\_INET6（ipv6）。
* type：指定socket类型。常用的socket类型有SOCK\_STREAM（TCP）、SOCK\_DGRAM（UDP）等。
* protocol：指定socket所使用的协议，一般我们平常都指定为0，使用type中的默认协议。严格意义上，IPPROTO\_TCP（值为6）代表TCP协议。
* 返回值：成功为文件描述符FD，失败未-1。

（2）bind

int bind(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);

* sockfd：一般为调用socket函数返回的文件描述符。
* addr：地址的格式与协议有关，ipv4查看man 7 ip，ipv6查看man 7 ipv6。
* addrlen：\*addr的长度。
* 返回值：0为成功，-1为失败。

（3）listen

int listen(int sockfd, int backlog);

* sockfd：一般为调用socket函数返回的文件描述符。
* backlog：待连接的等待队列的最大值。
* 返回值：0为成功，-1为失败。

（4）accept

int accept(int sockfd, struct sockaddr \*addr, socklen\_t \*addrlen);

* sockfd：一般为调用socket函数返回的文件描述符。
* addr：存储客户端的信息（如ip和端口），传入一个空的结构体，便于函数内部为这个结构体赋值，这样外部就能访问了。
* addrlen：\*addr的长度。
* flags：c语言不能像java中一样方法重载，所以只能用个新的方法accept4加个flags参数，这个参数也是用来设置非阻塞SOCK\_NONBLOCK的。
* 返回值：成功则返回与客户端建立连接的socket，也就是一个文件描述符FD，失败则返回-1。

（5）connect

int connect(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);

* sockfd：一般为调用socket函数返回的文件描述符。
* \*addr：服务器端的地址，格式参考bind函数。
* addrlen：\*addr的长度。
* 返回值：0为成功，-1为失败。

（6）send

ssize\_t send(int sockfd, const void \*buf, size\_t len, int flags);

* sockfd：文件描述符。
* \*buf：要发送的内容。
* len：发送的内容的长度
* flags：标志位，默认为0，具体参考man 2 send。
* 返回值：成功则返回发送的内容的长度，失败则返回-1。

（7）write函数原型如下：

ssize\_t write(int fd, const void \*buf, size\_t count);

7.2.3.2 多线程

（1）创建线程

int pthread\_create(pthread\_t \* tid, const pthread\_attr\_t \* attr, void \* ( \* func) (void \* ), void \* arg);

* 线程变量名：pthread\_t \*类型，是标示线程的id，一般是无符号整形，这里也可以是引用类型，目的是用于返回创建线程的ID
* 线程的属性指针：制定线程的属性，比如线程优先\*级，初始栈大小等，通常情况使用的都是指针。
* 创建线程的程序代码：一般是函数指针，进程创建后执行该函数指针只想的函数。
* 程序代码的参数：若线程执行的函数包含由若干个参数，需要将这些参数封装成结构体，并传递给它指针。

（2）结束线程

void pthread\_exit (void \*status);

参数是指针类型，用于存储线程结束后返回状态。

（3）终止线程

该函数可以用于向指定的线程发送信号：

int pthread\_kill(pthread\_t threadId,int signal);

如果线程内不对信号进行处理，则调用默认的处理程式，如SIGQUIT会退出终止线程，SIGKILL 会杀死线程等等,可以调用signal(SIGQUIT, sig\_process\_routine); 来自定义信号的处理程序。

pthread\_kill 返回值如下：

0: 调用成功。

ESRCH：线程不存在。

EINVAL：信号不合法。

7.3 LED驱动程序

7.3.1问题分析

该程序负责根据用户给定的配置与广告语的内容，控制发光LED滚动显示广告语。为了达成此目的，其需要完成的主要任务如下：

a、实现从文件中读取配置。

b、实现查字符编码数据库文件，将字符形式的广告语转换为能在发光LED上显示的编码 形式的功能。

c、实现按照给定配置，将发光LED编码形式的广告语在发光LED上水平滚动显示的功能。

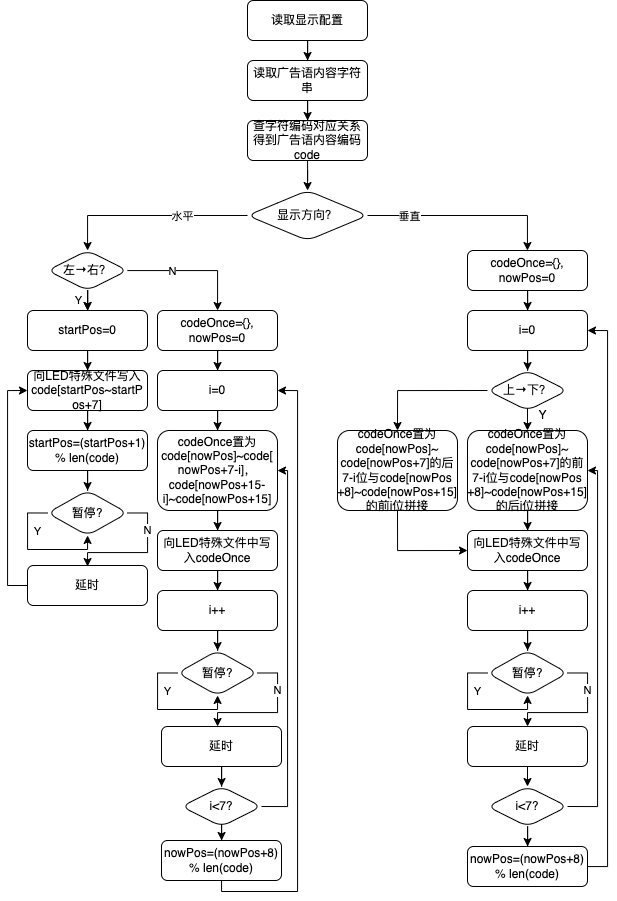
d、实现按照给定配置，将发光LED编码形式的广告语在发光LED上垂直滚动显示的功能。

e、实现广告语滚动显示的暂停与恢复功能。

7.3.2算法设计

算法使用的协议格式如上小节所示。

程序算法流程图如下：



1. 测试报告：

**测试项1：通过客户端配置当前滚动显示广告语的内容**

测试计划（测试步骤和测试指标）：

a.双击打开配置客户端（验证能否正常打开配置客户端）。

b.在广告语文本框中输入广告文本字符串，点击“添加广告”按钮（验证能否正常输 入及添加广告语）。

c.在IP地址文本框中输入IP地址（验证能否正常输入广告语）。

d.选择“速度”下的单选框（验证单选框能否正常选择且是否仅能选择一个选项）。

e.选择“显示方式”下的单选框（验证单选框能否正常选择且是否仅能选择一个选项)。

f.单击“设定运行”按钮，观察点阵LED（验证能够按照所给配置及广告语内容正确 显示广告语）。

测试环境：

配置客户端运行在Windows 7操作系统上，LED驱动程序以及后端服务器运行在 UP-CUP2440开发板上。

测试操作：

a.双击打开配置客户端。

b.在广告语文本框中输入广告文本字符串“TEST”，点击“添加广告”按钮。

c.在IP地址文本框中输入IP地址“192.168.24.18”。

d.选择“速度”下的单选框“1x”。

e.选择“显示方式”下的单选框“上->下”。

f.单击“设定运行”按钮。

测试结果与分析：

测试结果显示，配置客户端能够正常打开；广告语能够成功输入且添加；能够正常配置广告语显示配置；设定后点阵LED能够按照给定配置正确显示广告语。

总结：

系统中通过客户端配置当前滚动显示广告语内容的功能正常。

**测试项2：通过客户端配置当前滚动显示广告语的滚动方向**

测试计划（测试步骤和测试指标）：

a.更改“显示方式”下选择的单选框（验证单选框能否正常选择且是否仅能选择一个 选项）。

b.单击“设定运行”按钮，观察点阵LED（验证点阵LED能够根据更改后的显示方向 滚动显示广告语）。

c.依次更改“显示方式”下选择的单选框，重复上述步骤（验证广告语在各个方向均 可正常滚动显示）。

测试环境：

配置客户端运行在Windows 7操作系统上，LED驱动程序以及后端服务器运行在 UP-CUP2440开发板上。

测试操作：

a.选择“显示方式”下的单选框“下->上”。

b.单击“设定运行”按钮。

c.依次选择“显示方式”下的单选框“左->右”、“右->左”，重复上述步骤。

测试结果与分析：

测试结果显示，在各个方向上，均能够在客户端上正常更改广告语显示方式配置；更改显示方式并设定后广告语能够按照配置正常在点阵LED上滚动显示。

总结；

系统中通过客户端配置当前滚动显示广告语的滚动方向的功能正常。

**测试项3：通过客户端配置当前滚动显示广告语的速度**

测试计划（测试步骤和测试指标）：

a.更改“速度”下选择的单选框（验证单选框能否正常选择且是否仅能选择一个选项）。

b.单击“设定运行”按钮，观察点阵LED（验证点阵LED能够根据更改后的显示速度 滚动显示广告语）。

c.依次更改“显示速度”下选择的单选框，重复上述步骤（验证广告语能够按照所配 置的不同速度滚动选择广告语）。

测试环境：

配置客户端运行在Windows 7操作系统上，LED驱动程序以及后端服务器运行在 UP-CUP2440开发板上。

测试操作：

a.选择“速度”下的单选框“2x”。

b.单击“设定运行”按钮。

c.选择“速度”下的单选框“3x”，重复上述步骤。

测试结果与分析：

测试结果显示，在不同速度下，均能够在客户端上正常更改广告语显示速度配置；更改显示速度并设定后广告语能够按照配置正常在点阵LED上滚动显示。

总结：

系统中通过客户端配置当前滚动显示广告语的速度的功能正常。

**测试项4：通过客户端暂停广告语的滚动**

测试计划（测试步骤和测试指标）：

a.在“速度”下选择单选框“暂停”（验证单选框能否正常选择且是否仅能选择一个 选项）。

b.单击“设定运行”按钮，观察点阵LED，（验证点阵LED能够暂停广告语的滚动显示）。

测试环境：

配置客户端运行在Windows 7操作系统上，LED驱动程序以及后端服务器运行在 UP-CUP2440开发板上。

测试操作：

a.选择“速度”下的单选框“暂停”。

b.单击“设定运行”按钮。

测试结果与分析：

测试结果显示，能够在客户端上的显示速度一栏下正常选择暂停选项；选择暂停后并设定后点阵LED上显示的广告语暂停滚动。

总结：

系统中通过客户端暂停广告语的滚动的功能正常。

**测试项5：通过客户端实现对暂停广告语的重新播放**

测试计划（测试步骤和测试指标）：

a.在“速度”下选择单选框“3x”（验证单选框能否正常选择且是否仅能选择一个选 项）。

b.单击“设定运行”按钮，观察点阵LED（验证点阵LED能够恢复广告语的滚动显示）。

测试环境：

配置客户端运行在Windows 7操作系统上，LED驱动程序以及后端服务器运行在 UP-CUP2440开发板上。

测试操作：

a.选择“速度”下的单选框“3x”。

b.单击“设定运行”按钮。

测试结果与分析：

测试结果显示，能够在客户端上的显示速度一栏下正常选择非暂停的选项；选择非暂停选项并设定后点阵LED上显示的广告语恢复滚动。

总结;

系统中通过客户端实现对暂停广告语的重新播放的功能正常。

**测试项6：在客户端中添加预设广告语**

测试计划（测试步骤和测试指标）：

a.在配置客户端左上角“广告语”下侧的文本框中输入广告语（验证能够正常输入广 告语）。

b.单击“添加广告”按钮，观察右侧“已有广告语”（验证能够正常添加预设的广告 语）。

c.输入其他广告语，重复如上步骤（验证能够正常添加多条预设广告语）。

d.重启客户端（验证当客户端关闭后之前预设的广告语仍然保留）。

测试环境：

配置客户端运行在Windows 7操作系统上，LED驱动程序以及后端服务器运行在 UP-CUP2440开发板上。

测试操作：

a.在配置客户端左上角“广告语”下侧的文本框中输入字符串“TESTA”。

b.单击“添加广告”按钮。

c.依次输入广告语“TESTB”、“TESTC”、“TESTD”，重复如上步骤。

d.重启客户端后观察右侧“已有广告语”。

测试结果与分析：

测试结果显示，能够在客户端上正常输入且添加多条广告语；在客户端重启后之前预设的广告语仍然保存。

总结：

系统中在客户端中添加预设广告语的功能正常。

**测试项7：在客户端中删除预设广告语**

测试计划（测试步骤和测试指标）：

a.在客户端右侧“已有广告语”中单击选中一个预设的广告语（验证能够正常选中预 设的广告语且一次仅可选中一个）。

b.单击“删除选中广告语”按钮，观察右侧“已有广告语”（验证能够正常删除预设 的广告语）。

c.选中其他预设广告语，重复如上步骤（验证能够正常删除多条预设广告语）。

d.重启客户端（验证删除的持久性）。

测试环境：

配置客户端运行在Windows 7操作系统上，LED驱动程序以及后端服务器运行在 UP-CUP2440开发板上。

测试操作：

a.在配置客户端右侧“已有广告语”中单击选中预设广告语“TESTA”。

b.单击“删除选中广告语”按钮。

c.依次选中广告语“TESTB”、“TESTC”、“TESTD”，重复如上步骤。

d.重启客户端后观察右侧“已有广告语”。

测试结果与分析：

测试结果显示，对多条不同广告语，均能够在客户端上正常选中一条广告语；能够正常删除选中的广告语。在重启客户端后，之前删除的广告语具有持久性。

总结：

系统中在客户端中删除预设广告语的功能正常。

1. 总结

1、如何依据理论指导项目开发

若我们在课堂上的理论学习与项目开发的结果无关，则所能体验到的仅仅是动作的不断重复，却无法体验到行动与结果之间的关系。依据理论指导项目开发，在实际操作时是先明确项目开发的目标，然后根据目标细化内容，根据内容明确所需要的的开发技术，然后再学习并掌握开发技术的相关理论。只有通过理论理解技术实现过程，在开发时才能做到有序和规范。以我们小组项目的实现为例，首先进行需求分析，明确要实现的功能，然后各成员根据各自要实现的功能学习相关的技术理论，掌握相关理论后进行程序的实现设计，然后根据程序的实现设计进行程序开发。

1. 开发项目过程中的得与失
2. 收获

提升了了linux命令行操作的熟练度，在此次团队嵌入式开发的过程中，掌握了进程相关的内存共享技术和管道技术，以及多线程相关操作。由于与此次项目的队友是第一次合作，整个合作过程一定程度上锻炼了我的交流沟通能力。

1. 不足

小组合作方面，由于前期小组成员之间没有过合作经历，所以沟通上不是很流畅，导致前期开发浪费了一些时间；功能实现方面，由于开发时间有限，最后实现的功能上还比较简单。