基于 i. MX6 的 LED 异步控制系统软件设计

杨 博[1]，李可生[1]，何书专[1,2]，李 伟[1,2]，李 丽[1,2]，潘红兵[1,2]

（1. 南京大学 电子科学与工程学院，江苏 南京 210046；
2. 南京大学 江苏省光电信息功能材料重点实验室，江苏 南京 210093）

摘 要：为满足户外 LED 屏幕对异步控制的需求，利用面向对象的设计方法，设计实现一款基于 i. MX6 处理器的 LED 全彩异步控制系统嵌入式软件。提出基于多线程的事件队列和基于插件的 GStreamer 流媒体播放系统等技术方案；利用嵌入式 Linux 环境下可移植库的特性，实现对上位机发送的数据进行灵活的动态化处理，以及对多种格式的多媒体节目播放的异步控制；在性能上优化开机时间和播放策略。测试结果表明，该方案拥有较好用户体验，满足了设计需求。

关键词：面向对象；多线程；流媒体播放；异步控制；嵌入式软件
中图法分类号：TP311 **文献标识号**：A **文章编号**：1000-7024（2016）06-1478-07
doi：10. 16208/j. issn1000-7024. 2016. 06. 011

Design of embedded software used on i. MX6 based asynchronous LED display control system

YANG Bo[1], LI Ke-sheng[1], HE Shu-zhuan[1,2], LI Wei[1,2], LI Li[1,2], PAN Hong-bing[1,2]

(1. School of Electronic Science and Engineering, Nanjing University, Nanjing 210046, China; 2. Jiangsu Provincial Key Laboratory of Photonic and Electronic Materials Science and Technology, Nanjing University, Nanjing 210093, China)

Abstract: Aiming at asynchronous control requirement of outdoor LED screen, the design and implementation of embedded software was put forward, which could be used on the i. MX6 processor based asynchronous LED display control system. An event queue method based on multithreading in the announced embedded software was adopted, accompanied with the utilization of the GStreamer based streaming media player system. Using the characteristics of portable library in embedded Linux environment, the data from the upper computer were processed dynamically and asynchronous control of multimedia programs in multiple formats was realized. Considering the functionality of the system, boot time and playing strategy were optimized for the performance. Results of several tests show that the proposed methods have better user experiences and satisfying results.

Key words: object oriented; multithreading; streaming media player; asynchronous control; embedded software

0 引 言

LED（light emitting diode）异步控制系统又称作脱机控制系统，需要显示的多媒体数据由上位机终端编辑，并通过串口、网口等接口预先传入异步控制系统中的存储器中，然后由控制系统控制，使之循环的播放多媒体信息。相较于同步控制系统必须要有一台实时同步工作的终端这一特点，异步控制系统具有的特点是操作简单，价格不高，应用场合更加广泛[1]。鉴于异步控制系统脱机工作的优势，本设计选用异步控制系统的方案。利用拥有强大多媒体处

收稿日期：2015-06-25；**修订日期**：2015-08-27
基金项目：国家自然科学基金项目（61176024、61006018、61370040、61376075）；高等学校博士学科点专项科研基金项目（20120911110029）；江苏省产学研联合创新资金-前瞻性联合研究基金项目（BY2013072-05）
作者简介：杨博（1989-），男，江苏南京人，硕士研究生，研究方向为 VLSI 设计、嵌入式系统开发；李可生（1991-），男，江苏连云港人，硕士研究生，研究方向为 VLSI 设计、嵌入式系统开发；何书专（1974-），男，海南琼海人，工程师，研究方向为模拟集成电路；李伟（1975-），男，黑龙江齐齐哈尔人，工程师，研究方向为多核处理器软件；李丽（1975-），女，山东威海人，博士，教授，博士生导师，研究方向为 VLSI 设计、多核处理器芯片体系架构及实现方法、SoC-IP 设计；潘红兵（1971-），男，江苏常熟人，博士，教授，博士生导师，研究方向为 VLSI 设计、多核处理器软件、并行计算、霍尔传感器。E-mail：y131j@qq. com

理能力的 i. MX6 系列处理器和其相关的 BSP 开发套件，基于嵌入式 Linux 操作系统，自主开发了基于多线程的 LED 异步控制系统嵌入式软件，为 LED 全彩屏的异步控制提供了一套可靠解决方案。

1　LED 异步控制系统硬件设计方案

1. 1　整体结构设计

本设计由上、下位机协作工作，整个 LED 全彩异步控制系统由控制主卡和接收卡控制系统组成，把微处理器作为控制核心，和显示控制终端通信，控制 FPGA（field-programmable gate array）完成数据处理、显示屏刷新等任务。这种方案可以充分发挥微处理器和可编程逻辑器件各自的特点，设计的集成度高，维护调试方便，性能比较强[2]。

LED 全彩异步控制系统是 LED 显示屏上的重要组成部分。如图 1 所示的是本文所设计的 LED 全彩异步控制系统结构框架图，从左往右分为 3 个部分，依次是上位机，控制主卡和接收卡。首先上位机负责编辑图片或视频等多媒体数据、制作节目单和播放参数等，通过网线与控制主卡通信。然后，控制主卡是由两部分组成：ARM 部分和 FPGA 部分，两部分通过 LVDS（low voltage differential signaling）和 I^2C（inter-integrated circuit）接口通信。其中 ARM 部分是带有 ARM 架构处理器的核心板，负责把接收到的上位机数据存储在 FLASH 中，保证掉电后数据不会丢失，并由其嵌入式软件负责控制解析多媒体数据，经过硬件解码把数据发送给 FPGA 部分。FPGA 部分是发送主卡 FPGA，负责对来自 ARM 的流媒体信号和配置信号的处理，实现灰度的控制，完成 LED 显示屏的扫描驱动电路，并通过千兆网络把数据发送到接收卡 FPGA。最后是接收卡 FPGA，负责接收主卡的数据并完成显示器的驱动，因为单个模组尺寸较小需要把多个小块接收卡 FPGA 以及显示模组级联，经过拼接还原成大尺寸 LED 显示屏，具体数量依据实际 LED 屏幕尺寸确定。

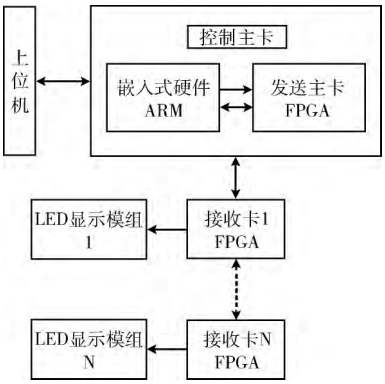

图 1　LED 异步控制系统总体结构框架

1. 2　下位机嵌入式硬件系统

控制主卡即下位机硬件系统，主要由以下三大部分组成：ARM（i. MX6s）核心板、XC6SLX16 FPGA 显示模块和其它外围接口器件。其中 ARM 核心板是本设计的嵌入式硬件系统，控制主卡的总体构成如图 2 所示。

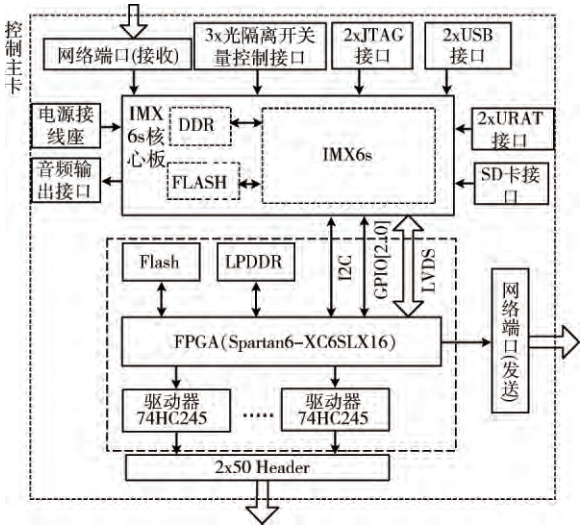

图 2　控制主卡硬件结构

ARM 部分嵌入式硬件，主要的应用需求是多媒体信息处理，本文选用的 ARM 的处理器是基于 ARM Cortex-A9 架构[3]，飞思卡尔的 i. MX6 系列应用处理器。其拥有尖端 3D 和 2D 图形处理能力和高清视频功能，同时拥有丰富的接口和较低的功耗，在多媒体方面表现出色。

2　LED 异步控制系统嵌入式软件设计方案

LED 异步控制系统嵌入式软件在整个系统里又称作下位机软件。本设计采用的是基于 UML 的面向对象框架标准建模[4]，利用面向对象的分析（OOA）方法，把复杂的系统抽象为一个个对象以及其属性和服务来分解任务。有利于贯穿软件生命周期全过程的一致性。

2. 1　软件需求

面向上位机，本嵌入式软件设计了四大基本功能。①查询功能：包括对 FPGA 反馈的 LED 坏点、下位机工作状态、日志等内容的查询。②升级功能：包括对嵌入式软件的升级和 FPGA 的升级。③设置功能：对于下位机本身，包含了系统时间和 IP 的设置；对于 LED 显示屏，包含灰度级、扫描方式、亮度、模组尺寸等显示参数设置；对于播放，包含节目单设置和插播的设置；④播放功能：包括根据节目单的设置内容播放多媒体数据和根据上位机指令控制本嵌入式软件停止和继续播放，播放节目支持的数据类型：图像、文字和视频。

面向基本用户，本嵌入式软件还提供对即插即用设备的响应，即将设置好的播放数据存放在即插即用设备中（如 U 盘、SD 卡、TF 卡），插入下位机后，嵌入式软件会自行播放设备中设置好的内容。

根据软件需求设计的软件用例图如图 3 所示。

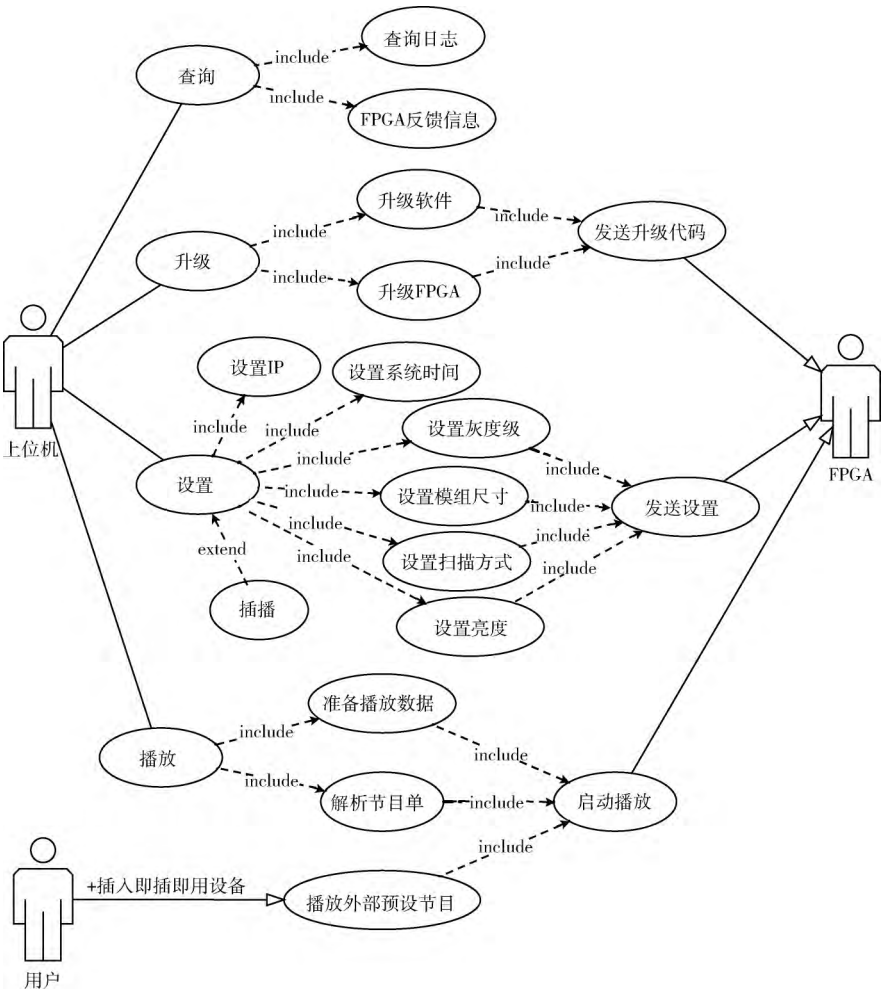

图 3 嵌入式软件用例

2.2 软件模块设计

通过对软件需求的分析，根据面向对象设计方法，可以把本嵌入式软件抽象为 6 个基本模块包括：网络模块，线程控制模块，总控制模块，播放模块，日志模块和设置模块。这些模块是按照最粗粒度方式抽象的对象名称，他们不属于同一个抽象层次，但不同的对象之间互相关联，此阶段需要完善各个模块属性和职责，自上而下的设计整个软件。

网络模块：实现上位机（PC）与下位机（ARM）软件的通信。编程时使用 SOCK _STREAM 流套接口即 TCP（transmission control protocol）协议，保证数据有序和无错。把下位机设置为小型 FTP（file transfer protocol）服务器，用 XML 格式文件来保存节目单增强跨平台工作特性。

总控制模块：用于协调网络模块和线程控制模块工作，解析节目单，是负责连接与反馈的中间模块。查询功能与升级功合并在总控制模块里，为后续功能升级留出扩展。

线程控制模块：采用单例的设计模式，负责四大线程（查询、设置、升级和播放）的多线程多任务并行管理，实现任务的创建，执行，关闭以及反馈。

播放模块：负责读取节目单，从播放队列中领取播放任务，播放流媒体节目。

日志模块：用来写本地日志的工具，具备使用简单、结构简单、实现高效和线程安全等特点。支持在日志行前添加日期、时间、线程号和日志级别；日志最大数量可调整；支持 debug、info、error、fatal 等多种级别日志。

设置模块：负责保存上位机发送的设置信息，保存在本地的 SQlite 数据库中，方便查询，并且把设置信息发送给 FPGA。

在设计出嵌入式软件各模块后，需要使用类图的形式表现各个模块的结构关系[5,6]。在 UML 设计里，类图是静态视图，将行为实体描述成离散的模型元素，但不包含它们的动态行为细节。其关键元素是类元以及他们之间的关系，常用的类元有类、接口和数据类型。而最终我们设计的对象是类元的实例，是具有特定身份的类。本设计类图如图 4 所示。

2.3 软件流程

根据上一节嵌入式软件模块的设计，本节结合各模块静态视图以及软件需求来整体描述软件的动态工作流程。

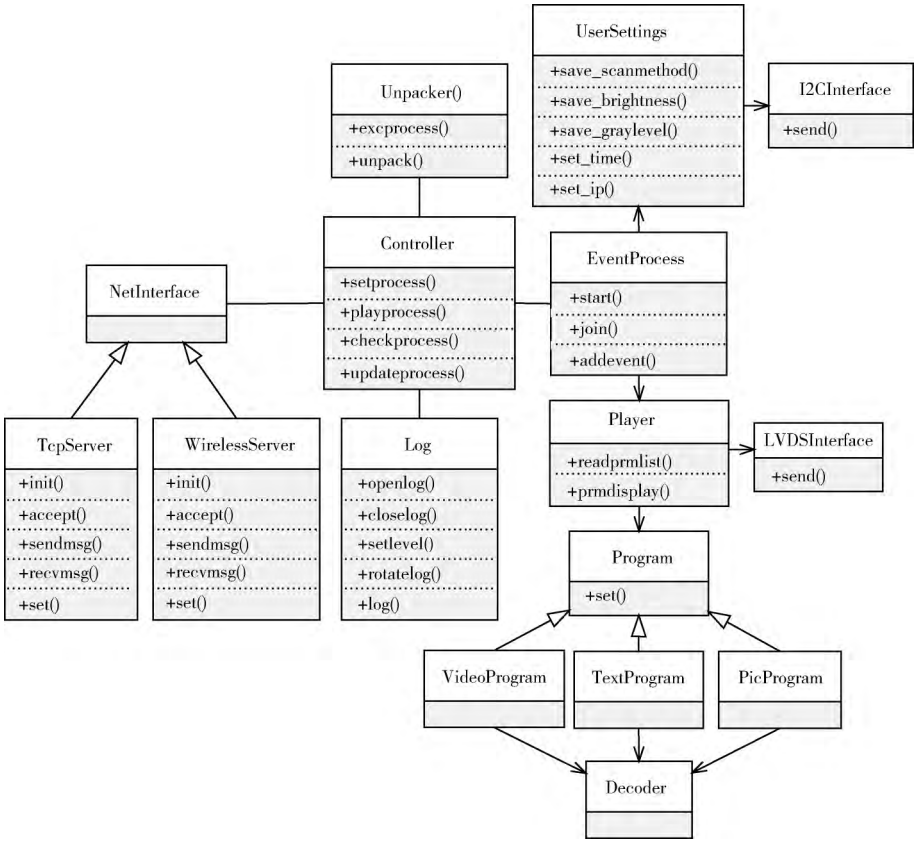

图 4　嵌入式软件类

按照软件的生命周期各模块行为顺序把运行时状态分为控制层、功能层和 IO 层。最上层的控制层由网络模块、总控制模块和线程控制模块构成，是整个系统的核心层，负责软件数据流控制，协调模块之间通信，为下层模块添加新的任务。具体为上位机终端在用户配置下，生成配置信息，并通过 RJ45 等网络接口传送至下位机，网络模块会进行解包分析，如果收到的信息不全或者格式不正确，会要求上位机重发送数据包。在确定接收信息无误后，总控制模块会调用解析节目单子模块和线程控制模块，把新的播放任务添加进各线程的任务队列等待领取。中间的功能层是由播放模块，设置模块和日志模块构成，负责从对应线程的任务队列里领取任务完成并把数据交给下一层。具体为设置模块的线程在任务队列里领取到新的设置任务，调用 SQLite 数据库保存在本地，播放模块的线程在任务队列里领取到新的播放任务，由解码子模块根据节目单内容选择合适的解码单元对流媒体数据进行解码，完成任务后更新节目单。底层的 IO 层由播放模块和设置模块的子模块 LVDSInterface 和 I2CInterface 组成，显然是两个数据接口模块，播放模块对应的接口是高速并行单向接口 LVDS，在硬件解码完毕后数据通过这个接口发送给 FPGA；设置模块对应的接口是双向接口 I2C，在播放配置设置完毕后通过 I2C 接口与 FPGA 通信。当量个接口的数据都发送完毕后下位机的一整套流程也运行结束，循环往复等待上位机新的消息。据此嵌入式软件的流程如图 5 所示。

3　LED 异步控制系统嵌入式软件关键部分实现

3.1　开发环境以及开发组件

本设计的硬件环境是基于 i. MX6 处理器的开发板。软件环境是开源嵌入式 Linux 操作系统，版本号为 i. MX 6 Linux 3. 0. 35 _4. 1. 0 release，编程语言是 C＋＋。在源主机上开发的程序需要通过交叉编译工具（飞思卡尔的 fsl-linaro-toolchain 工具链）在 X86 的 Linux 平台上生成基于 ARM 体系结构目标板可执行的二进制文件，通过网络文件系统 NFS，将主机文件系统共享到目标板上，实现目标板和主机之间的通讯，把可执行二进制文件发送到目标板文件系统中调试运行程序。

该嵌入式操作系统遵循 GPL 开源协议，有大量的免费的开发插件和函数库可以使用[7]，利用这些成熟并且可复用的库可以更加专注在软件本身，减少开发时间，提高开发效率。在 usr/lib 和 usr/local/lib 这两个目录下包含非系统运行时所需要的库和第三方应用程序的库。而本次开发所使用的库见表 1。

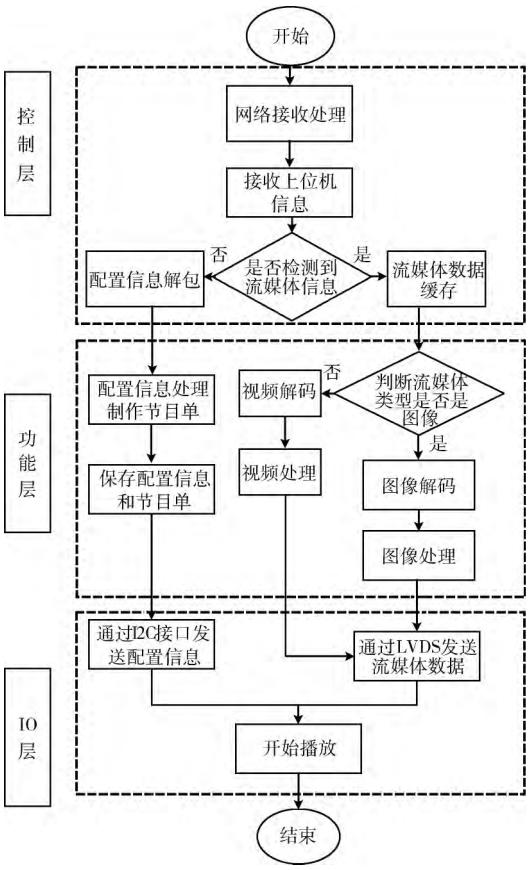

图 5 嵌入式软件工作流程

表 1 嵌入式软件应用库

库名	解释
SQLite	本地型嵌入式数据库
Glib-2.0	C 语言函数库
GStreamer-0.10	多媒体框架库
Libxml2	XML 程序库
Freetype-2.6	可移植字体引擎库

3.2 线程控制模块

本软件在网络传输，播放节目，设置配置等方面存在并行的需求，例如在播放阶段，通过以太网通讯的上位机在控制下位机停止播放时，单线程操作就会使得整个软件处于假死状态，一直等到播放任务结束，软件才会响应上位机操作，影响整体效果和性能。本设计提出了一种多线程的解决方案来避免这种尴尬。

如图 6 所示，本文设计的嵌入式软件使用了队列和多线程机制来组建。①使用多线程编程使不同线程共享一块数据空间和地址空间，在不同线程间切换和通信所花费的时间很短，满足并行性运行的要求[8]。②网络模块在这里主要开辟了接收和发送线程，分别用来接收上位机所产生的待处理多媒体任务和发送下位机处理完的反馈任务。线程控制模块开辟了显示，设置，查询，升级等线程，总控制模块开辟了一个独立的日志线程用来实时记录软件的状态。这 4 个主要线程独立拥有自己的事件队列，在软件初始化完毕后可以同时存在。控制层和功能层之间的通信是通过线程队列来实现的，控制层在接收到上位机终端传送来的数据后形成待完成队列（TodoList），由功能层各模块里的线程去待完成队列里领取任务并执行，待各模块执行完毕后会形成一个待发送队列（ToSendList）统一由网络通信模块反馈给上位机。这样只要处理待完成队列就可以实时的更新播放状态，在结构上实现了更新节目单、即插即用设备播放等拓展功能。③在开辟 4 个主要线程和队列的时候使用了单例设计模式，保证系统中只有一个实例并且该实例易于被外界访问。其单例伪代码为：

```
//声明
static EventProcess* GetIntance();
//实现
EventProcess* EventProcess::Getinstance() {
加互斥锁
  ptr__=new EventProcess;
释放互斥锁
Return ptr__;
}
```

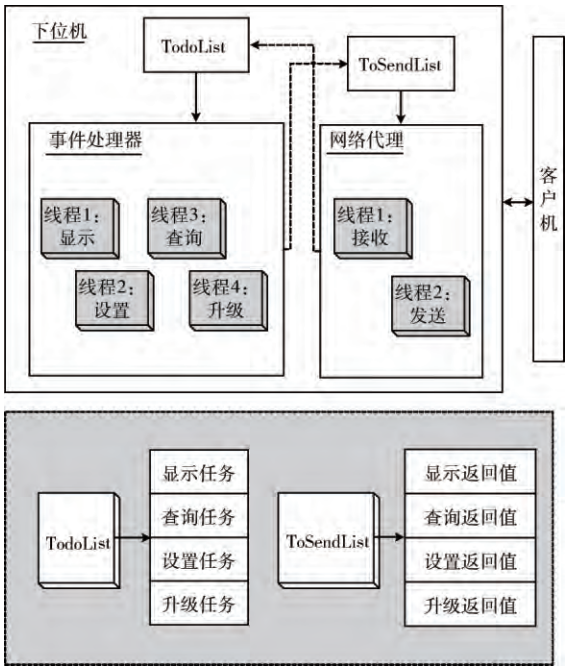

图 6 线程控制模块体系结构

3.3 设置模块

与下位机 I2C 接口相连的是 FPGA 中的 LED 驱动模块，LED 驱动模块的功能是给 LED 模组发送配置参数与同步信号，这是一种参数可配置的驱动模块。其构成包括参数配置单元，数据分割单元，行驱动单元、列驱动单元等模块。其中参数配置单元是直接和 I2C 交互的模块，用来

接收和发送各种参数。本设计所包含的参数有两种：第一种是发送给 FPGA 驱动模块的配置参数，第二种是接收 FPGA 自检芯片传送给核心板的回传参数。具体的两种参数信息如图 7 所示。

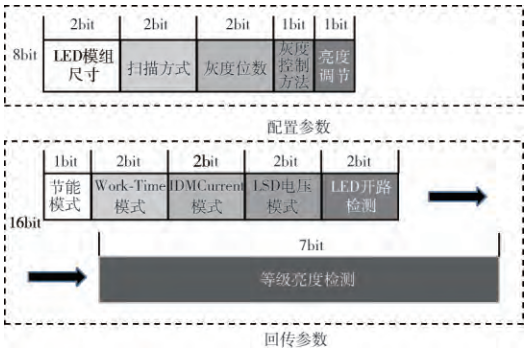

图 7　I2C 接口传输数据格式

据图 7 结构，在 LED 异步控制系统嵌入式软件中，从属于功能层设置模块需要对上位机数据分析解包，把配置信息解析出来。这里发送方的配置参数是 8 bit 数据，接收方的回传参数是 16 bit 数据，按照两种参数的不同比特格式设计了专用的数据结构保存并通过 I2C 接口通讯。本设计在保存配置参数时引入了 SQLite 轻量关系型数据库，其专业使用领域为嵌入式软件，对资源的占用非常低。整个数据库都存储在宿主主机上的一个单一文件中，在设置处理模块中调用 SQLite 的 API 函数把每次上位机发送的配置指令保存在数据库中，当上位机的配置信息发生变动或者有即插即用设备需要更新节目单时，就可以调用数据库中的配置信息，重新修改后更新数据库。

3.4　播放模块

GStreamer 是一款功能强大的通用流媒体应用开发框架。采用基于插件和管道的体系架构，能够实现插件间的无缝融合。在飞思卡尔公司提供的嵌入式 Linux 源码中就拥有 GStreamer 动态库，所以采用 GStreamer 框架搭建图片和视频播放器是最佳选择。本设计利用 GStreamer 实现多媒体播放的主要工作流程为[9,10]：

选定元件：首先根据实际需要选择 GStreamer 的元件，元件都是以插件的形式绑定在管道中。本设计基于 i. MX6 平台，除了 GStreamer 通用的元件以外，采用 vpudec 进行视频解码，采用 mfw _ipucsc 进行格式以及分辨率的转换，并将 mfw _isink 作为视频接收元件。

设置元件属性：安装 GStreamer 之后，在终端输入“gst-inspect＋某个特定的元件”则会显示该元件的具体信息，包括解释性质的信息以及开发者可以具体设置的信息(element properties)。例如设置 filesrc 元件的 location 属性用于提供需要播放的视频或图片存放的位置信息。

连接管道：如图 8 是 i. MX6 中的 GStreamer 元件所连成的管道，该管道适用于视频播放。在使用 GStreamer 进行编程时，最先的工作就是搭建可以正确运行的管道，利用 gst-launch 命令可以方便地在终端进行管道的搭建。因为终端搭建的管道只能够从头至尾执行，缺乏必要的控制功能，所以要在能够顺利工作以后再利用相应的库函数将管道翻译成程序。

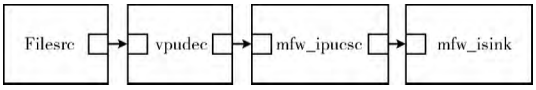

图 8　元件连接

控制状态：管道连接后，可以通过 GStreamer 中的函数对其状态进行控制。管道拥有 4 种状态：null，ready，playing，paused。配合 GStreamer 中的消息监听函数和键盘处理函数等，实现流媒体的播放、暂停、加减速等功能。

释放资源：所有的插件资源，管道资源使用结束后，必须利用 GStreamer 中特有的函数对内存进行释放。

除了视频和图片的显示，本软件还需要显示文字。而 GStreamer 搭建的播放器只支持把文字作为视频的字幕添加在屏幕上，这种限制是无法完成设计时提出的文字动态效果显示任务。所以在文字显示上又提出了 FreeType＋framebuffer 的方案，首先需要把 FreeType 库移植到嵌入式系统中，调用 FreeType 库提供的统一接口访问字体格式文件，生成相应的 bitmap 数据，最后再调用输出帧缓存设备(framebuffer) 的 API 函数把文字显示在屏幕上，其支持例如缩放，旋转等动态效果。如图 9 所示调用 FreeType 库，在字符终端模拟 LED 屏幕文字显示效果。

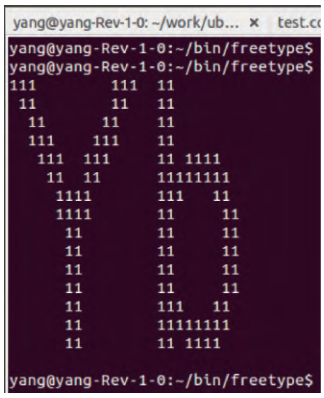

图 9　字符终端模拟字库显示

4　LED 异步控制系统嵌入式软件优化方案与结果分析

4.1　开机时间优化

没有优化过的系统从上电到加载用户完成需要 20 s 以上的时间，作为工业设计的系统开机时间过长会导致较差的用户体验，需要优化。在调试终端下逐条查看开机的打印信息发现操作系统加载 uboot 后需要等待 3 s 判断是否进入 uboot 设置界面，加载内核时启动了很多本应用不必要

的程序，例如蜂鸣器，hdmi 接口，摄像头等。这些情况都会延长启动时间，据此本设计的优化策略是：①优化 uboot 结构，提升拷贝数据速度；②裁剪内核，移除在启动时所不需要的程序；③在内核加载完毕后立即运行本设计软件。

优化之后上电加载时间缩减到 12 s，其中约有 4 s 时间是加载文件系统。

4.2 播放策略优化

播放指标主要为功能和性能两个部分：功能，指的是否可以正常播放多媒体文件；性能，这也是此设计的重点指标，性能越好则意味着在操作系统下 CPU 的占用率越低，出现卡顿的现象越少。

由于 i.MX6 拥有优秀的图形处理单元 VPU，可以对外完成多媒体的集中处理任务，在芯片内部通过 AXI 和 AHB 总线与 ARM Cortex A9 多核平台相连，通过操作系统提供的 VPU API 函数就可以加速图形图像的处理；性能上优化，主要是比较使用 ffmpeg 软件解码和使用 VPU 硬件加速模块解码播放视频的 CPU 占用率以及他们的加速比。在嵌入式 Linux 环境下运行程序，同时选择 GStreamer 元件中不同的解码元件，其中软件解码元件是 fftdec _h264、VPU 解码元件是 vpudec，表 2 是部分测试结果。

表 2 嵌入式软件性能优化测试结果

	软件解码	VPU 解码
avi 格式视频源（h.264 编码）	89%～96%	2.6%～7.2%
mkv 格式视频源（h.264 编码）	45%～53%	1%～2%

经测试实验可以发现，使用本设计中硬件加速播放器播放多媒体文件消耗的资源更少，保证了整个嵌入式软件的实时性能，但是使用硬件解码加速优化并不支持所有的格式的视频编码，而且不同视频源编码优化后效果也不同，所以考虑硬件解码优化方案的同时需要考虑所要使用多媒体视频源的编码格式是否适合本优化方案。

5 结束语

本文设计的基于 i.MX6 处理器的 LED 全彩异步控制系统软件，从原理与技术上提出了一套完整的嵌入式异步控制软件方案。创新性的使用 Gstreamer＋Freetype 结构来完成播放功能，可以实现图片，视频和文字的多种效果播放。通过硬件加速，最大程度上发挥 i.MX6 多媒体播放性能。在以后开发中，考虑到电子商业化需求的增大，必须以性能和用户体验为根本，继续优化软件，增加市场竞争力。

参考文献：

[1] FAN Yu, GONG Wei, LI Nong. Asynchronous control system of LED lattice displayer based on ARM [J] Journal of Changzhou University (Natural Science Edition), 2013, 25(4): 88-92 (in Chinese). [范宇，龚伟，李农. 基于 ARM 的 LED 显示屏异步控制系统 [J]. 常州大学学报：自然科学版，2013，25(4)：88-92.]

[2] LIU Jianwei. Research on asynchronous control system for traffic guiding display [D]. Xi'an: Xidian University, 2010 (in Chinese). [刘建伟. 交通诱导显示屏异步控制系统的研究 [D]. 西安：西安电子科技大学，2010.]

[3] Koppanalil J, Yeung G, O'Driscoll D, et al. A 1.6 GHz dual-core ARM Cortex A9 implementation on a low power high-K metal gate 32nm process [C] //International Symposium on VLSI Design, Automation and Test. IEEE, 2011: 1-4.

[4] Weilkiens T. Systems engineering with SysML/UML: modeling, analysis, design [M]. Morgan Kaufmann, 2011.

[5] Evans A, France R, Lano K, et al. Developing the UML as a formal modelling notation [J]. arXiv preprint arXiv: 1409.6928, 2014.

[6] Gomaa H. Software modeling and design: UML, use cases, patterns, and software architectures [M]. Cambridge University Press, 2011.

[7] ZHAO Yinghui, XIE Hua, XIAO Yindong. Implementing method of CLIPS shared object library on Linux [J]. Electronic Measurement Technology, 2010 (9): 48-51 (in Chinese). [赵英辉，谢华，肖寅东. Linux 环境下 CLIPS 动态链接库的实现方法 [J]. 电子测量技术，2010 (9)：48-51.]

[8] Iannucci RA, Guang R Gao, Halstead Jr, et al. Multithreaded computer architecture: A summary of the state of the art [M]. Springer Science & Business Media, 2012.

[9] Burks S D, Doe J M. GStreamer as a framework for image processing applications in image fusion [C] //SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, 2011: 80640M-80640M-7.

[10] TA Burks S D, DOE J M. GStreamer as a framework for image processing applications in image fusion [C] //SPIE Defense, Security, and sSensing. International Society for Optics and Photonics, 2011: 80640M-80640M-7.