

终稿_陈俊洪论文

【原文对照报告-大学生版】

报告编号: 414edee72a335f7e

检测时间: 2019-05-06 09:54:04

检测字数: 14,503字

作者名称: 终稿

所属单位: 宜宾学院

检测范围:

- | | | |
|------------------|-----------------|-------------------|
| ◎ 中文科技期刊论文全文数据库 | ◎ 中文主要报纸全文数据库 | ◎ 中国专利特色数据库 |
| ◎ 博士/硕士学位论文全文数据库 | ◎ 中国主要会议论文特色数据库 | ◎ 港澳台文献资源 |
| ◎ 外文特色文献数据全库 | ◎ 维普优先出版论文全文数据库 | ◎ 互联网数据资源/互联网文档资源 |
| ◎ 高校自建资源库 | ◎ 图书资源 | ◎ 古籍文献资源 |
| ◎ 个人自建资源库 | ◎ 年鉴资源 | ◎ IPUB原创作品 |

时间范围: 1989-01-01至2019-05-06

检测结论:

全文总相似比 = 复写率 + 他引率 + 自引率 + 专业术语
11.85% = **10.95%** + **0.91%** + **0.00%** + **0.00%**

其他指标:

自写率: 88.15%

专业用语: 0.00%

高频词: 模块, 设计, 数据, 视频, 进行

典型相似性: 无

指标说明:

复写率: 相似或疑似重复内容占全文的比重

他引率: 引用他人的部分占全文的比重, 请正确标注引用

自引率: 引用自己已发表部分占全文的比重, 请正确标注引用

自写率: 原创内容占全文的比重

专业用语: 公式定理、法律条文、行业用语等占全文的比重

典型相似性: 相似或疑似重复内容占互联网资源库的比重, 超过60%可以访问

总相似片段: 85

期刊: 10 博硕: 48 外文: 0 综合: 1 自建库: 0 互联网: 26

颜色标注说明:

- 自写片段
- 复写片段 (相似或疑似重复)
- 引用片段
- 引用片段(自引)
- 专业用语 (公式定理、法律条文、行业用语等)

宜宾学院

YIBIN UNIVERSITY

本科生毕业论文 (设计)



题 目 基于WIFI视频采集智能小车系统的设计

院 别 物理与电子工程学院

专 业 机械电子工程

学生姓名 陈俊洪

学 号 150306020 年级 2015

指导教师 王立刚 职称 助教

独创性声明

本人声明所呈交的学位论文是本人在指导老师指导下进行的研究工作及取得的研究成果。据我所知,除了文中特别加以标注和致谢的地方外,论文中不包含其他人已经发表或撰写过的研究成果,也不包含为获得宜宾学院或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名: 日期: 年 月 日

论文使用授权

本学位论文作者完全了解宜宾学院有关保留、使用学位论文的规定,有权保留并向国家有关部门或机构送交论文的复印件和磁盘,允许论文被查阅和借阅。本人授权宜宾学院可以将学位论文的全部或部分内容编入有关数据库进行检索,可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后应遵守此规定)

作者签名: 指导教师签名:

日期： 年 月 日

摘 要

现有的智能监控设备大多为有线数据传输，定点监控，难于操作等缺陷。本次设计的WIFI视频采集智能小车具有可移动，无线传输，易操作等特点。本次设计主要是WIFI视频采集智能小车系统的软件设计。首先，在linux系统下编写WIFI视频采集智能小车的单片机控制代码，包括电机控制、视屏数据采集、无线传输控制的代码。其次，在windows系统下用QT编写WIFI视频采集智能小车的终端软件，包括界面设计，数据传输控制的代码。最后，将Linux系统下编写好的单片机控制代码移植到小车单片机中，同时也将QT上编写好的软件代码移植到Android设备上。实现了在Android设备上控制WIFI视频采集智能小车的移动，并将WIFI视频采集智能小车采集到的视频据通过无线传输发送到Android设备上，实时显示WIFI视频采集智能小车采集到的视频等功能。由于本次设计的WIFI视频采集智能小车具有可移动，无线传输和易操作等特点，将大大节约资源和时间，对未来社会的发展具有深远的意义。

关键词：Linux；视频传输；WIFI；QT

ABSTRACT

At present most of the intelligent monitoring device for fixed-point monitoring, data transmission cable, difficult to drawbacks such as the operation. And the design of WIFI smart car has a mobile video acquisition, wireless transmission, easy to operate etc. This design is mainly responsible for WIFI video collection and intelligent car system software design. First of all, under Linux system WIFI video acquisition the single-chip microcomputer control of intelligent car code, including motor control, video data acquisition and wireless transmission control code. Second, under the Windows system written in QT WIFI smart car video acquisition terminal software, including interface design, data transmission control code. Then, will write good under Linux system transplanted into the car in the MCU single-chip microcomputer control code, at the same time will also be QT writing good software code ported to the Android devices. Finally achieved on the Android device to control the movement of WIFI smart car video acquisition, and WIFI video collection of smart car collected video according to sent via wireless transmission to the Android devices, real-time display of WIFI video sampling smart cars to the video, and other functions. Due to the design of WIFI smart car has a mobile video acquisition, wireless transmission and easy operation etc., will greatly save resources and time, has the profound significance to the development of the future society.

Keywords: Linux; Video transmission; WIFI; Qt

目 录

1.1 研究工作的背景与意义	2
1.2 国内外研究历史与现状	2
1.3 本文的主要贡献与创新	2
1.4 本论文的结构安排	3
第二章 方案的选取与分析	3
2.1 软件工具的选取与分析	3
2.2 操作系统的选取与分析	4
2.3 视频采集的选取与分析	5
2.4 电机驱动的选取与分析	6
2.5 无线传输的选取与分析	7
2.6 本章小结	7
第三章 结构设计	7

3.1 主体结构介绍	7
3.2 结构设计实物	7
3.3 本章小结	8
第四章 Linux模块设计过程	9
4.1 环境搭建	9
4.2 摄像头模块	10
4.3电机模块	12
4.6 WIFI模块	15
4.5系统使用模块	17
4.6本章小结	18
第五章 Qt模块设计过程	19
5.1 UI模块	19
5.2 tcp客户端模块	21
5.3移植Android模块	21
5.4本章小结	22
第六章 全文总结与展望	23
6.1全文总结	23
6.2后续工作展望	24
致 谢	25
参考文献	26
附 录	27

第一章 绪 论

1.1 研究工作的背景与意义

目前社会所需要的资源也越来越多,我们必须改进当前社会的一些设施,才能满足当前社会的需求。WIFI视频采集智能小车是一个智能控制、无线传输和视屏传输为一体的多功能集成系统。现代智能设备远程无线控制已经成为一个发展趋势,我们可以使用无线设备控制生活中的各种电器。WIFI视频采集智能小车告别传统的设计理念,把多种功能集中在一起,节约了大量的资源和时间。WIFI视频采集智能小车的研究和开发也将有利于我国在智能化领域的技术发展[1]。因此,WIFI视频采集智能小车研究将会对人们生活水平的提高具有深远的意义。

1.2 国内外研究历史与现状

目前许多国家都在研究智能设备,WIFI视频采集智能小车涉及到生活中的各个部分,如矿井检测、人口控制、智能识别等。WIFI视频采集智能小车具有智能控制、视频显示、无线传输的三大功能,有一个非常广阔的市场和应用前景[2]。在今天的许多大企业、街道和学校等都是使用固定的安全监控系统,使得很多地方不可以得到很好的监控,为了达到监控的完美性,把监控设备安装在很多地方,在资源上会造成很大的损失,并且在实际操作上又显得很实用。所以为了社会的发展,我们必须不断提高设备的高效化,这将极大地提高人类的经济效益,提高了人类社会的发展步伐。

1.3 本文的主要贡献与创新

本次设计主要解决目前监控设备上存在的不足,目前的监控设备,大多为定点监控,这样不易于调整摄像头的位置,不能全方位进行监控。其次监控设备多为有线传输,需要消耗大量的电器设备,还有不能在手机终端上直接显示视屏,难以移植等问题。为了解决这些缺陷,本次设计,首先,在linux系统下编写WIFI视频采集智能小车的单片机控制代码,包括电机控制、视频数据采集、无线传输控制的代码。实现数据的传输和摄像头的移动功能。其次,在QT上编写WIFI视频采集智能小车的终端软件设计,包括界面设计,数据

传输控制的代码。然后，将Linux系统下编写好的单片机控制代码移植到小车单片机中，同时也将QT上编写好的软件代码移植到Android设备上[3]。实现了在Android设备上控制WIFI视频采集智能小车的移动，并将WIFI视频采集智能小车采集到的视频数据通过无线传输发送到Android设备上，实时显示WIFI视频采集智能小车采集到的视频等功能。

1.4 本论文的结构安排

本次论文根据自身所学的内容解决现实生活中的一些问题，首先介绍本次设计的选题背景和研究意义，然后根据本次设计需要解决的问题，在原理上进行分析，然后确定出本次设计的主要内容，进行方案的选取的论述，得出最后的方案，其次介绍整个方案的设计过程，最后总结此次设计中遇到的问题和所需要注意的地方。

第二章 方案的选取与分析

本章主要介绍本次设计方案的选取与分析，首先针对本次设计的需求，从原理上进行分析，找出本次设计所存在的问题；其次对方案进行可行性的分析和优缺点的比较，从而确立本机设计所需要的方案，保证设计的合理性和先进性。

2.1 软件工具的选取与分析

智能控制软件的制作，选用Qt来设计软件。Qt是一款以C++为基础的编程软件，主要是关于图形用户界面的应用软件。一款面向对象的编程软件，使用特别的代码生成函数和一些宏，Qt扩展非常简单，并且允许真正的组件编程。可以全面支持iOS、Android、WP等多个终端系统，它能提供应用软件编写人员图形界面所需的所有功能[4]。是一款很全面的编程软件，被程序员广泛使用。

在Qt上进行终端软件的编写，WIFI部分主要是构建Tcp客户端，接收来自tcp服务器的图像数据，发送按键部分的指令给tcp服务器；按键部分主要是编写按键模块的代码，将按键产生的指令传输给WIFI部分；视频部分主要是接收tcp客户端的数据，实时显示图像；移植Android模块主要是将Qt上编写的软件移植到Android设备上，使用Android设备来代替电脑运行软件，控制小车，Qt模块结构如图2-1所示。

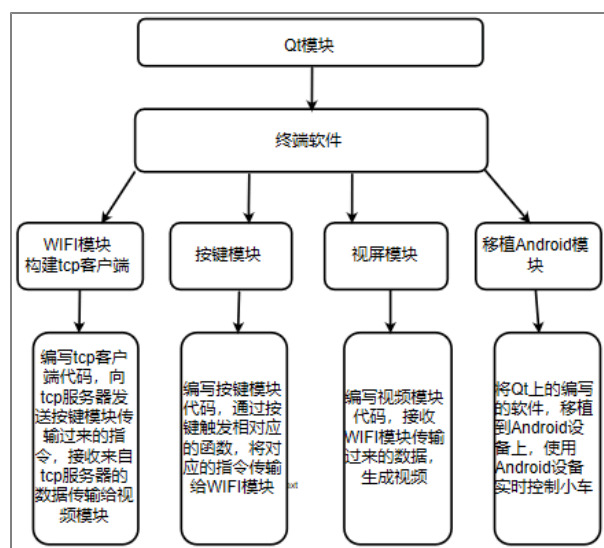
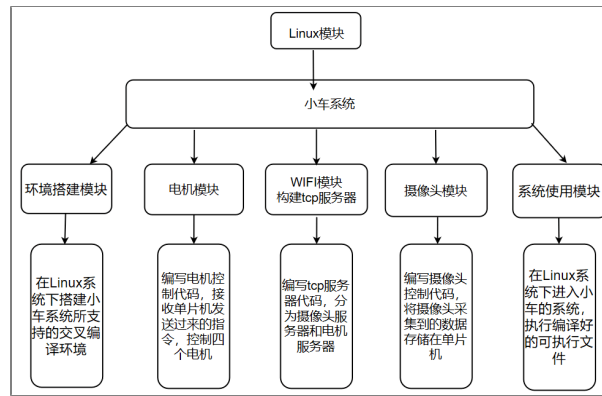


图2-1 QT模块示意图

2.2 操作系统的选取与分析

操作系统有选择linux操作系统，Linux非常的方便，容易使用，配置要求很低，无需花费，支持各个终端系统，安全性很高，支持各种网络方式、方便的控制台操作等优点。而windows是一款收费很高的软件，不开源，代码难以理解，不好修改操作系统自身的代码，并且漏洞比较多。和linux系统相比自身的函数不可以修改，网络功能不完善，很不利于程序员的编程和开发。所以本次设计选择Linux操作系统[5]。



在linux系统下，编写小车单片机的控制代码，电机部分主要是编写电机的控制代码；执行单片机发送过来的指令，控制四个电机的转速，实现小车的移动。WIFI部分主要是编写tcp服务器的代码；接收tcp客户端传送过来的信息放在单片机中，然后将信息发送给电机的驱动部分；同时将摄像头部分采集到的图像数据通过tcp服务器发送给tcp客户端；摄像头部分主要是编写摄像头的控制代码，将外界的环境信息采集到单片机中存储起来，提供给tcp服务器使用。linux模块结构如图2-2所示。

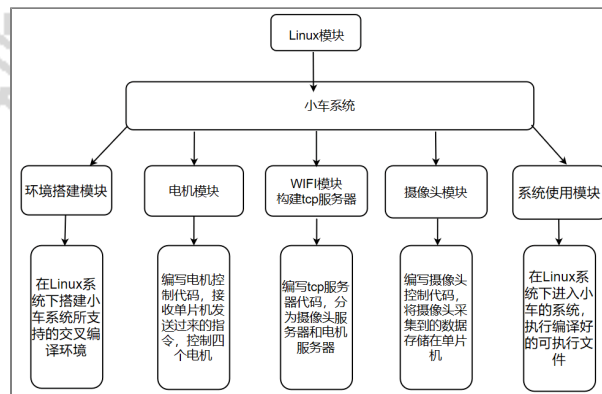


图2-2 Linux模块示意图

2.3 视频采集的选取与分析

视频采集模块，选用V4L2的视频驱动结构。首先，V4L2是针对linux系统设计的视频框架，其主体架构在linux的内核层，并且是linux系统上的视频捕获的驱动。

V4L2驱动架构已经使用几十年，常常在生活中被运用。生活中的立体设备、移动设备、固定设备等都会用到这个框架来进行视频的采集。并且在网络上关于V4L2的学习资料也比较齐全，方便开发者寻找资料，易于解决实际操作中遇到的各种问题。所以本次视频模块采用V4L2的驱动架构[6]。

2.4 电机驱动的选取与分析

本次设计主要采用ESP8266芯片作为智能小车的系统主控，在进行数据处理的同时负责WiFi信息的传输。并且L298N芯片设计电机驱动电路，具有足够大的动力足以驱动小车运行。

QT编写客户端软件用于连接小车的wifi，如图2-3所示，客户端发送驱动协议信息经WiFi传输到达ESP8266芯片后进行处理，控制小车移动。

通信协议包定义					
下行数据协议格式					
包头	类型位	命令位	数据位	包尾	备注
停止	FF	00	00	00	FF
前进	FF	00	01	00	FF
后退	FF	00	02	00	FF
左转	FF	00	03	00	FF
右转	FF	00	04	00	FF

图2-3 小车驱动通信协议

2.5 无线传输的选取与分析

无线传输模块，选用TCP/IP协议。TCP协议，提供完善的字节流服务，是一种面向对象的连接方式。当两边相交换信息时，需要先在两方建立一个TCP连接，建立之后才能进行数据的传输[6]。TCP能在超过时间后再次发送，能无误的检测每一个数据，能控制流量的使用大小，能把重复数据进行删除等，从而保证数据传输的正确性，TCP传输数据过程如图2-4所示。

UDP用户数据报协议是一种不可靠的协议，它仅仅是把应用层发送给IP层的数据报传送出去，是一种简易的运输层协议，并不能准确的保证数据能传送到目的地。UDP在发送数据报的时候，不会再次发送，也不会客户端和服务端之间搭建关系。

TCP是面向连接的，有比较高的可靠性。本次设计中数据的传输量不是很大，但是要求数据传输必须准确，因此选取TCP/IP协议来进行无线传输。

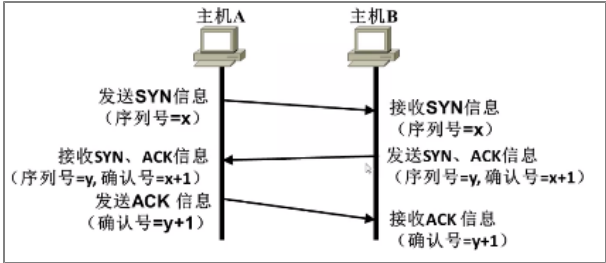


图2-4 TCP数据传输示意图

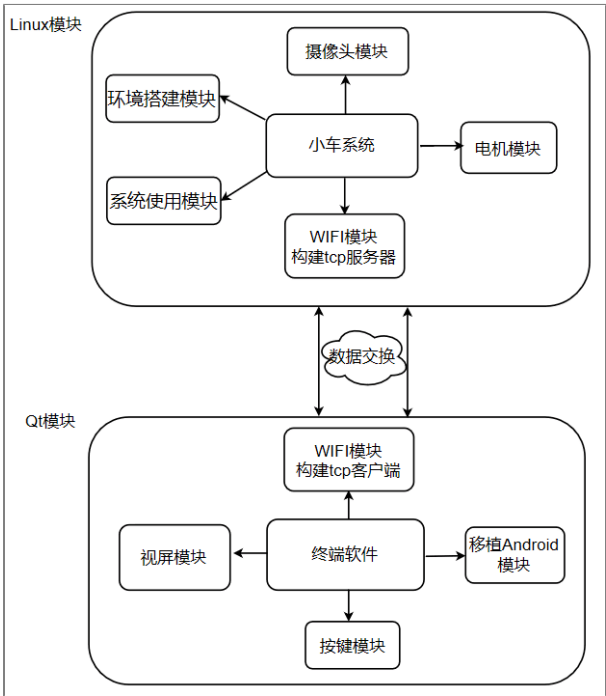
2.6 本章小结

通过以上的分析和论证，最终确立本次设计的方案。本次设计的方案主要分为两个大的模块：第一个大的模块在Linux系统上进行，主要负责小车单片机控制系统代码的编写，由环境搭建模块、系统使用模块、摄像头模块、电机控制模块和Tcp服务器模块组成。第二大的个模块在windows系统上利用QT进行编程，主要负责Android设备上软件的编写，由视频显示模块、按键控制模块、移植Android模块和Tcp客户端模块组成。在这两个大的模块下，有很多的小的模块，需要先从小模块开始，一步一步的逐个击破，将小的模块进行一一实现，然后再实现大的模块，最后将两个大的模块进行联合调试，才能达到最终的结果，完成本次的设计。

第三章 结构设计

本章主要介绍本次设计硬件的主体结构，首先从本次设计的功能上入手，分析所需要的硬件有哪些，然后选择适合的硬件，进行整体结构的组装，保障本次设计功能的可靠性。

3.1 主体结构介绍



本次设计使用Qt作为客户端的界面，总体设计结构如图3-1所示。Linux编写服务端程序，利用TCP/IP协议将小车上摄像头采集到的视频数据传送到指定的地方，再将数据信息通过TCP/IP协议发送到小车的wifi模块中，然后再使用串口将接收到的控制数据发送到核心控制板上，最终实现对方向的控制以及摄像头采集数据的功能[7]。

图3-1 设计方案示意图

3.2 结构设计实物

智能小车结构如图3-2、3-3所示，车身采用汽车式的结构，以此保证装置的稳定性。整个板子搭建在四轮小车上，无线模块处于小车的前半段，后半段是电池和一些串口，摄像头处于整个装置的最前面，便于采集数据。

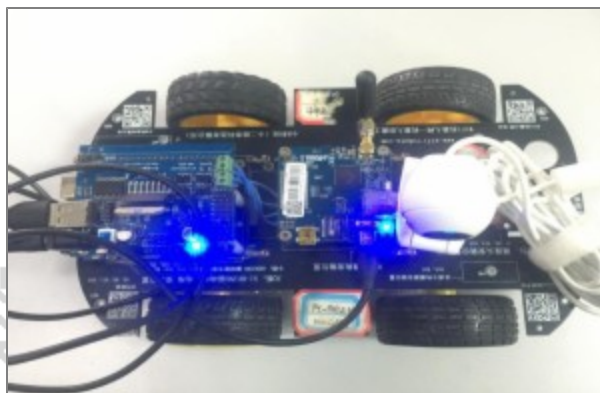


图3-2 智能小车实物图

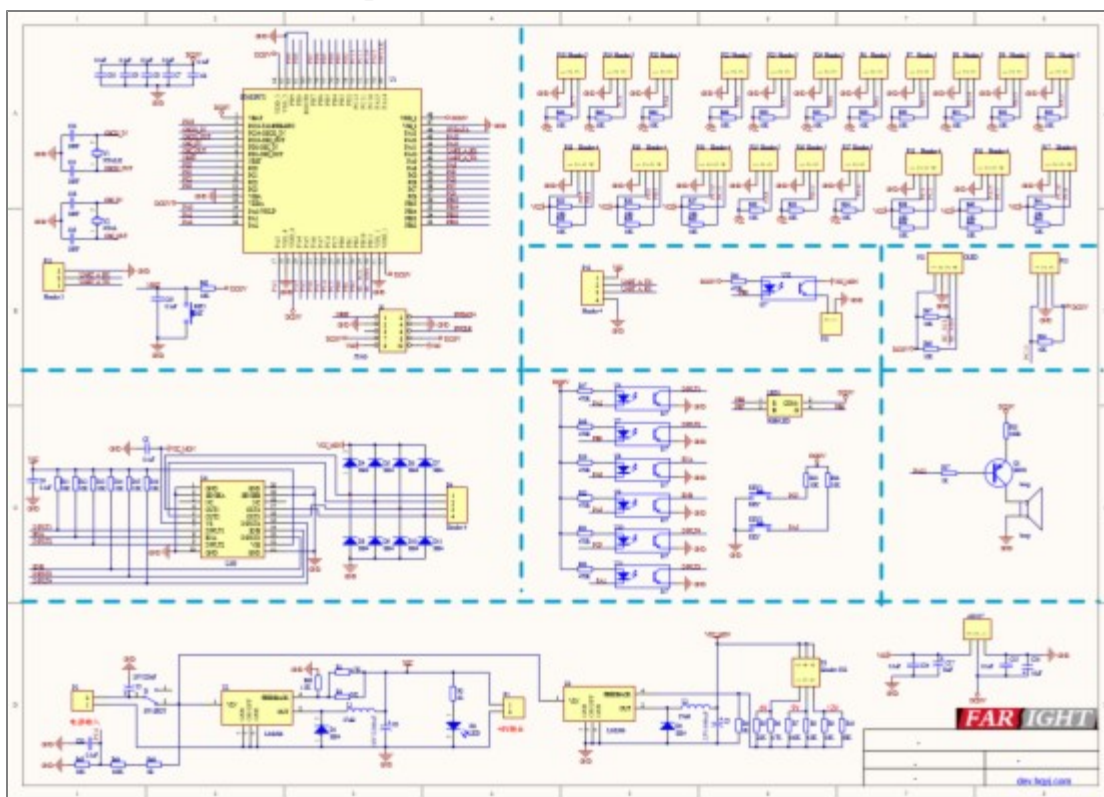


图3-3 智能小车电路原理图

3.3 本章小结

本次结构设计首先从原理上进行分析，然后结合自身所需的功能，进行反复的安装与调试，最终将各个模块进行了合理的安装，实现了各个模块之间的数据传输，协调工作，保证了本次设计的完整性。

第四章 Linux模块设计过程

本章主要介绍linux模块的设计过程，linux模块主要负责小车单片机控制系统代码的编写，前期需要将小车所需的开发环境搭建成功，然后进行摄像头模块、电机模块和Wifi模块代码的编写，最后进入小车的系统，执行编译好的可执行程序，实现小车的正常工作。

4.1 环境搭建

因为不同的系统对应的编译器是不一样的，所以首先应该在Linux下安装小车系统所支持的编译器，用该编译器编译出来的代码，才能在小车上运行。将交叉编译器在Linux系统下进行解压，然后再设置对应的环境变量，保存环境变量信息，检查是否生效这4个步骤后，就可以使用该交叉编译器，环境搭建流程图如图4-1所示。



图4-1 环境搭建示意图

环境搭建具体实施步骤：

1、解压交叉编译器

```
tar jxvf Open Wrt1- SDK -ar 711 x x- f o r- l i n u x- i 6 8 6- g c c- 4 . 8- l i n a r o_ u C l i b c- 0.9.33.22.tar.bz2
```

2、设置环境变量

```
sudo vim /etc/bash.bashrc
```

在最后一行添加：

```
export PATH=/usr/bin:/home/farsight/vidoes/OpenWrt-SDK-ar71xx-for-linux1-i686-gcc-4.8-linaro_uClibc-0.9.33.22/staging_dir/toolchain-mips_34kkc_gcc-4.8-linaro_uClibc-0.9.33.22/bin:$PATH
export STAGING_DIR=/home/farsight/vidoes/OpenWrt-SDK-ar71xx-for-linxs-i686-gcc-4.8-linaro_uClibc-0.9.33.22/staging_dir/toolchain-mips_34kc_gcc-4.8-linaro_uClibc-0.9.33.2/mips-openwrt-linux/staging_dir
```

3、保存环境变量：

```
source /etc/bash.bashrc
```

4、检查一下是否生效：

```
mips-openwrt-linux-gcc
```

4.2 摄像头模块

摄像头模块采用V4L2的架构和USB的通信方式；主要负责将外界环境信息采集到内核空间，再将数据传送到用户空间，最后提供给应用程序使用。linux下摄像头数据采集原理图如图4-2所示

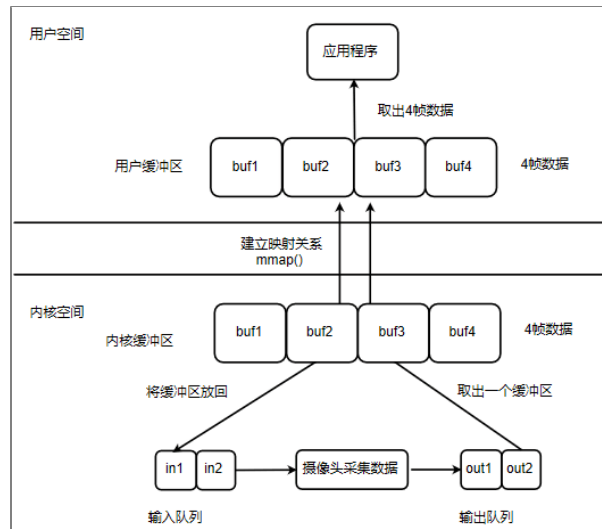


图4-2 数据采集示意图

摄像头初始化过程：

摄像头初始化分为九个步骤，第一步，以可读写的方式打开摄像头；第二步，显示设备驱动对应的信息，驱动的名字，设备的名字等，第三步，显示摄像头所支持的格式，摄像头格式有两种格式，一种是yuyv格式，一种是mjpeg格式；第四步，设置视频格式，视频的类型、宽度、高度、格式等，这里采用的格式为mjpeg格式；第五步，设置帧缓冲参数，帧的个数，帧的类型，帧的存储方式；第六步，申请缓冲队列，用队列的方式，对数据进行出队和入队；第七步，获取帧缓冲的地址；第八步，将内核空间映射到用户空间节约数据传输的时间[10]；第九步将数据放入缓存队列，进行数据的读写，摄像头初始化流程如图4-3和4-4所示。

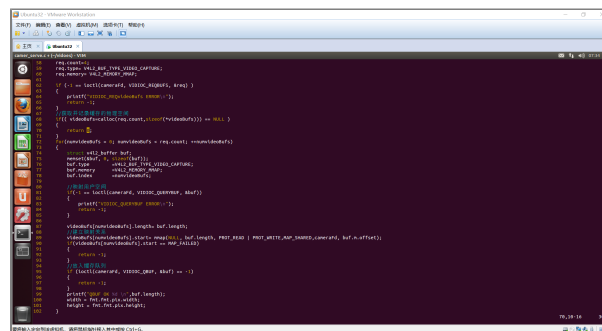


图4-3 摄像头初始化示意图

图4-4 摄像头代码图

摄像头初始化具体实施步骤：

- 1、打开视频文件，调用openr()函数
- 2、显示设备信息，调用ioctl()函数
- 3、显示设备所支持的格式令
- 4、设置视频捕获格式
- 5、设置帧缓冲参数，调用ioctl()函数，执行VIDIOC_REQBUFS命令
- 6、申请缓冲队列，调用calloc()函数，申请动态内存

8、从系统映射到用户，调用相应的库函数

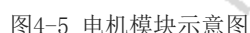
9、将数据放入缓存队列，调用*ioctl()*函数，执行VIDIOC_QBUF指令

摄像头模块小结:

摄像头模块是智能小车的核心模块之一，该模块主要应该决解以下两点问题：1、摄像头初始化函数的编写，配置摄像头的各个参数并了解各个参数代表的含义；2、了解摄像头数据循环采集过程，如何从内核空间传递到用户空间的过程[11]。

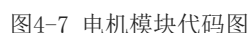
4.3电机模块

电机模块使用四个电机和USB的通信模式，主要负责接收应用程序发送过来的指令，控制各个电机的转速，实现小车的移动；首先接收Android设备发送过来的数据，将数据发送给用户程序，用户程序将对应的数据转码成电机设备能识别的指令，再将指令发送给电机设备，电机设备接收到指令后，开始驱动电机，控制小车的移动[12]。电机模块流程图如图4-5所示



电机编码原理:

电机模块主要是控制小车的移动，数据采用USB方式进行传输；电机初始化分为九个步骤：第一，打开电机设备，获取文件描述符；第二，获取终端属性；第三，设置适应的波特率，两者应该保持一致；第四，设置控制模式，保证程序不占用串口和保证程序可以从串口中读取出来；第五，设置数据格式，第六，设置输出模式和本地模式为原始模式；第七，设置时间和最小字符数，一般时间设置为0，最小字符数设置为1；第八，数据溢出处理，如果发生数据溢出，只接收数据，但不进行读操作；第九，激活配置，进行数据的传送；电机初始化流程如图4-6所示，电机模块代码图如图4-6所示。



具体实施步骤如下：

- 1、打开设备文件
- 2、获取终端属性
- 3、设置波特率
- 4、设置控制模式
- 5、设置数据格式

6、设置本地模式和输出模式

7、设置最少接收字符数和等待时间

8、数据溢出处理

9、激活配置

4.4数据转码

首先将发送过来的数据和特定的字符串进行比较，根据小车驱动的通信协议生成相关的指令代码，然后将相应的指令发送给电机驱动，控制小车的移动数据转码流程图如图4-8所示。



图4-8 数据转码示意图

小车驱动通信包定义如图4-9所示：

通信协议包定义					
下行数据协议格式					
	包头	类型位	命令位	数据位	包尾
停止	FF	00	00	00	FF
前进	FF	00	01	00	FF
后退	FF	00	02	00	FF
左转	FF	00	03	00	FF
右转	FF	00	04	00	FF

图4-9 小车驱动通信协议

4.5电机模块小结

电机模块是智能小车的核心模块之一，主要负责小车的移动控制，电机模块应注意以下两点：1、电机初始化函数的编写，了解各个函数和参数的意义；2、了解各个电机驱动代码是让电机如何工作的。

4.6 WIFI模块

WIFI模块主要负责智能小车与Android终端设备之间的数据传输，智能小车系统主要负责搭建tcp服务器，分为摄像头服务器和电机服务器。首先初始化tcp服务器，调用tcp服务器初始化函数；tcp服务器建立好后，就开始接收和发送数据给单片机内部，单片机将发送过来的数据传输给摄像头和电机，WIFI模块流程图如图4-10所示，WiFi模块代码图如图4-11所示。

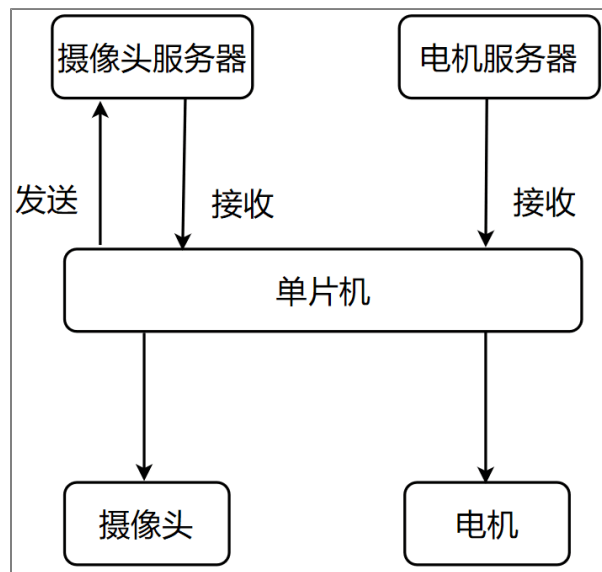


图4-10 WIFI模块示意图

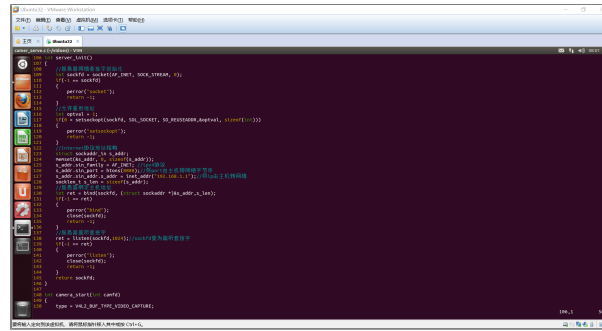


图4-11 WIFI模块代码示意图

具体实施过程：

- 1、建立tcp服务器，调用tcp服务器初始化函数
- 2、服务器接收和发送数据，调用send()和recv()函数
- 3、服务器结束，调用close()函数

4.7 tcp服务器

第一步利用库函数创建服务器的套接字，第二步利用库函数进行绑定，第三步利用库函数监听是否有客户端连接过来，第四步利用库函数接受客户端的连接请求；tcp服务器初始化流程图如图4-12所示。

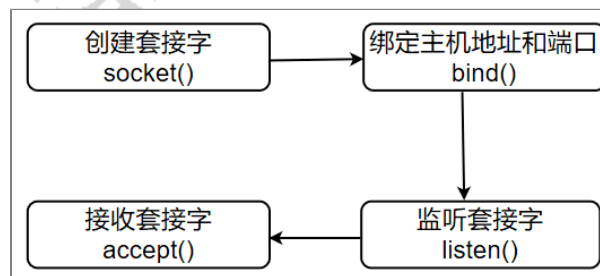


图4-12 tcp服务器示意图

tcp服务器初始化具体实施过程：

- 1、创建套接字
- 2、绑定主机地址和端口
- 3、监听是否有客户端连接
- 4、接受客户端的连接请求

WIFI模块小结：

WIFI模块是智能小车的核心模块之一，主要是解决数据的无线传输，WIFI模块主要应解决以下两点：1、tcp服务器初始化的原理，各个函数对应的功能；2、数据传输的原理，数据传输函数的使用；

4.5系统使用模块

本环节主要介绍如何从linux系统下进入小车的系统，以及如何将可执行文件下载到小车的系统中，然后在小车系统中执行可执行程序，系统使用模块流程图如图4-13所示。

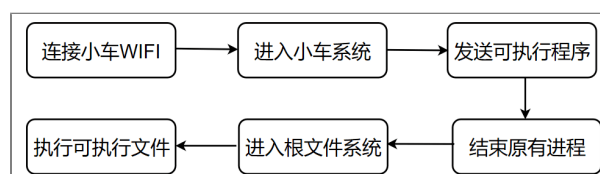


图4-13 系统使用示意图

小车系统流程图具体实施步骤：

1、连接小车的WIFI；

2、进入小车系统，代码如下：

```
sudo ssh 192.168.1.1 密码: admin;
```

3、将Linux下编译好的可执行代码发送到小车系统中，代码如下：

```
sudo scp camer 192.168.1.1:/ 密码: admin
```

4、结束小车系统原有的摄像头进程，代码如下：

```
ps
```

```
kill 846852; //小车摄像头相关进程号
```

5、进入小车根文件系统，代码如下：

```
cd /
```

6、在小车根文件系统中执行可执行文件，代码如下：

```
./filename(可执行文件名)
```

4.6本章小结

Linux模块主要是进行智能小车单片机控制系统的编写，由五个小的模块组成，分别是环境搭建模块，系统使用模块，摄像头模块，电机模块，WIFI模块；前两个是通用模块，后三个模块是Linux模块中的核心模块，设计时需要重点解决这三个模块。再代码的实际编程中，首先要养成良好的编码习惯，可以避免很多语法上的错误，节约大量的编程时间，其次了解每个小的模块的编程原理，理解代码中代表的实际意义，才能快速的进行代码的编写，然后再根据实际所需要的功能进行相应代码的编写，最后将编写好的5个小的模块联合起来进行相应的调试，找出存在的问题，经行相应的调试，完成Linux模块的设计。

第五章 Qt模块设计过程

本章主要介绍Qt模块的设计过程，Qt模块主要负责终端软件的编写，在Qt软件上，首先使用UI设计出软件的界面，其次搭建tcp客户端，然后将UI设计的各个对象进行相应的槽连接，编写相应的槽函数，最后将Qt上调式好的软件移植到Android手机，Qt设计总体界面图如图5-1所示[12]。

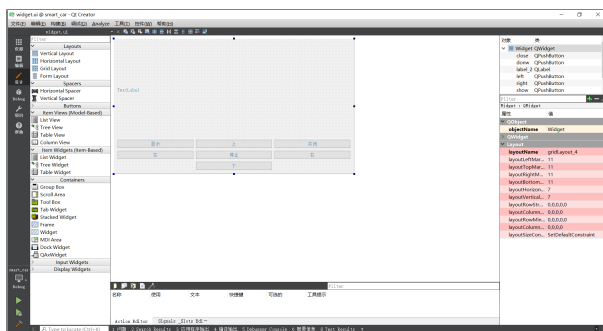


图5-1 UI界面图

5.1 UI模块

UI模块主要是对软件界面进行设计，UI中的对象有很多，根据自己软件所需要的功能选择相对应的对象进行设计，然后进行相应槽函数的编写；本次UI模块主要设计按键模块和视频模块的功能，UI设计流程图如图5-2所示。

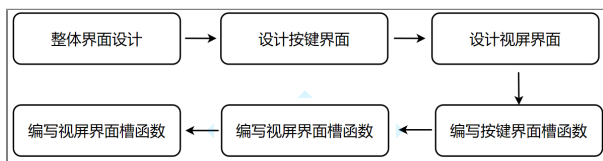


图5-2 UI模块示意图

UI设计实施过程：

首先根据本次设计的要求，分析得出软件的功能分为两点：1、通过按键控制小车的移动。2、通过标签循环显示图片；然后根据软件的功能，设计出按键和标签后，进行槽函数的连接，槽函数编写每个按键对应的功能函数[13]。

按键界面设计：

按键模块使用UI设计出的按键界面如图5-3所示。

图5-3 按键模块示意图

按键模块具体实施步骤：

- 1、编写前进功能按键函数
- 2、编写后退功能按键函数
- 3、编写左转功能按键函数
- 4、编写右转功能按键函数
- 5、编写视频显示功能按键函数
- 6、编写视频关闭功能按键函数

视频界面设计：

使用UI设计出视频显示的界面如图5-4所示

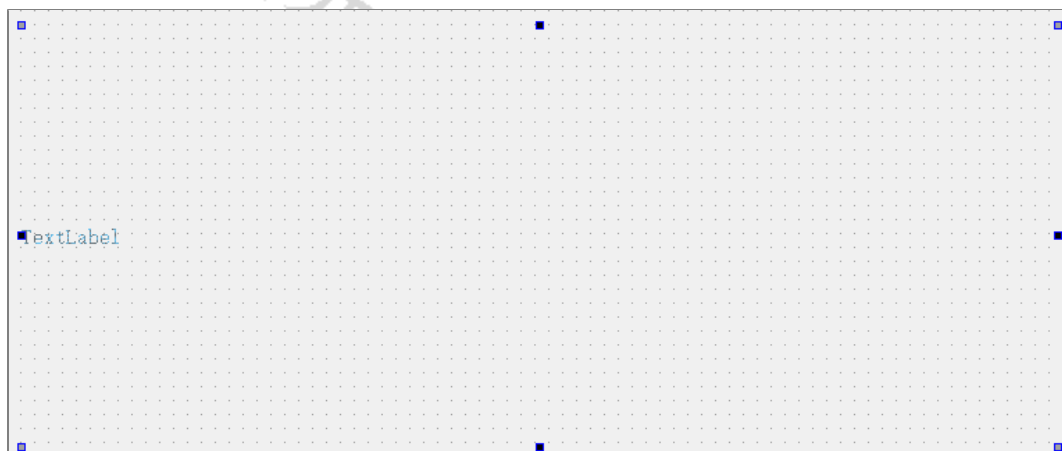


图5-4视频显示示意图

视频显示模块具体实施步骤：

- 1、设计视频界面
- 2、将按键和标签进行槽函数的链接，编写槽函数

5.2 tcp客户端模块

客户端主要是连接服务器，然后与服务器进行数据的交换[14]；客户端将软件产生的数据发送给小车上的服务器，客户端流程图如图5-5所示。

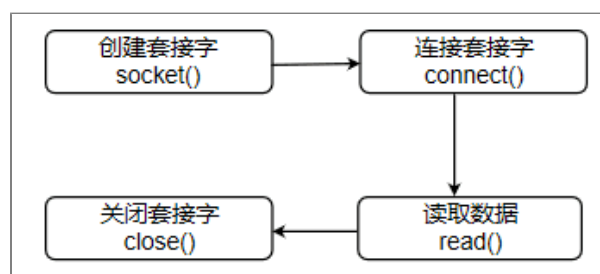


图5-5 tcp客户端示意图

tcp客户端模块具体实施过程：

- 1、ip设置
- 2、端口号设置
- 3、套接字设置
- 4、监测和接受消息设置

5.3移植Android模块

Android环境搭建，需要分别下载所需的安装包，然后将下载后的安装包进行安装[15]，将QT上编写好的界面程序使用USB连接下载到Android手机上，移植后Android手机上程序界面显示如图5-6所示。

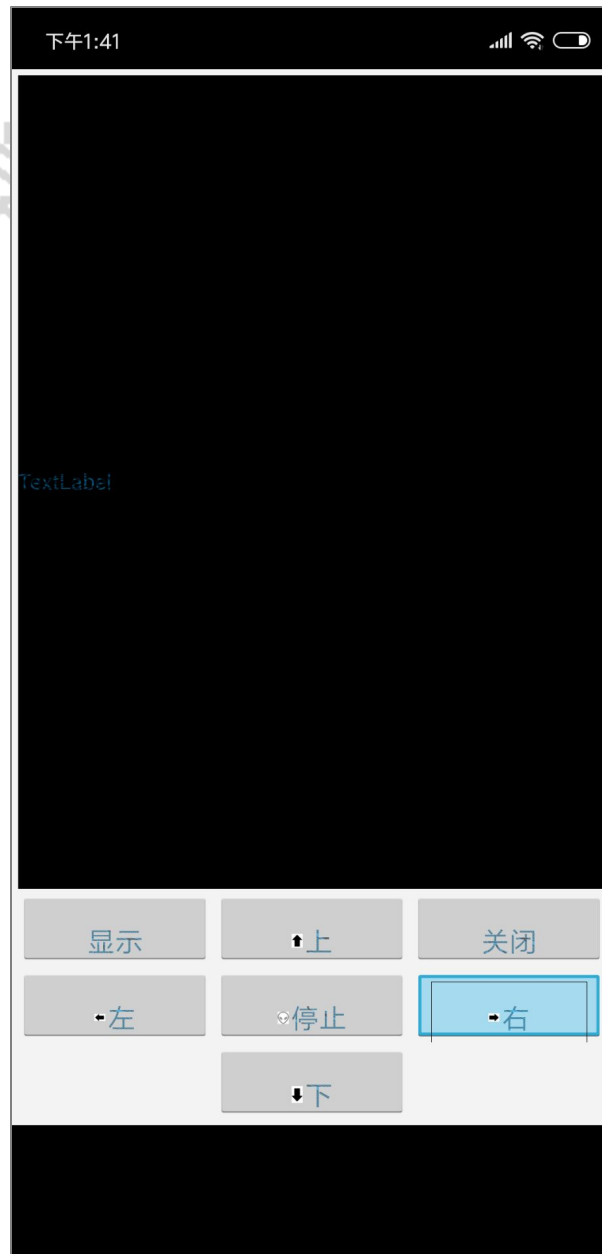


图5-6软件示意图

5.4本章小结

QT模块是负责终端软件的编写，需要熟练使用QT软件，第一步，了解软件所需要的功能，根据功能找出QT上对应的函数，了解函数各个参数代表的含义，然后进行相应的配置，将各个模块实现后，进行相应的调试，协调好各个模块之间数据的接受和处理和移植过程

中电脑端和手机端的适应性问题等等，最终完成QT模块的设计。

第六章 全文总结与展望

6.1 全文总结

在本次设计中，需要运用到很多的知识，在Linux系统上进行编程，需要我们熟悉Linux系统的操作，了解Linux系统的工作原理，熟悉各个驱动设备之间的协议，例如视频的驱动协议和电机的驱动协议，还要了解网络之间如何进行通信的等一些内容才能完成小车自身代码的编程；在QT上，我们需要熟悉QT各个库函数所对应的功能，如何去配置那些参数才能达到自己的目的，还要明白如何把QT上制作的软件转移到安卓设备上等内容，才能完成终端软件的制作。智能小车的设计将我大学四年中的知识很好的应用到设计中，通过查询很多的资料，询问老师和同学，让我增长了许多知识，提高了我的代码能力和动手能力，从各个分模块的调试到整机调试这一过程遇到了很多的难题，不过最终都得到了解决，完成了本次的设计，如图6-1所示。不过在代码的优化和硬件的选择上仍有很多需要完善之处。



图6-1 设计实物图

6.2 后续工作展望

随着我国科学技术的不断进步，人们对社会的需要不断增多，这需要我们不断完善现有的设备，将设备的使用效率不断的提高，才能满足人们的需求和社会的发展，WIFI视频采集智能小车在原有的基础上进行了改进，使它的先进性、可靠性、实用性不断升高。但是由于时间和知识水平有限，在WIFI视频采集智能小车系统的设计上，存在着许多的不足和缺点，还有很多方面需要进一步的改善。今后，随着自身知识的不断提高，找出本次设计中缺点的解决办法，不断的完善这些缺点，使本次设计变的越来越好。

致 谢

在攻读本科学士学位期间，首先衷心感谢我的论文指导老师王立刚。王老师对我的悉心教导让我在遇到很多问题是能够豁然开朗，能够找到解决的办法，让我不气馁，最终走向成功。

另外还要感谢我们15级6班的同学，在作品的制作过程中，也得到了许多同学的宝贵建议，在此致以诚挚的谢意。感谢所有帮助过我的朋友。

最后，向在百忙中抽出时间对本文进行评审并提出宝贵意见的各位领导老师表示衷心地感谢！由于自身能力有限，设计中一定存在很多不足之处，敬请各位老师批评指正。

参考文献

- [1] 赵映琳. 浅析智能电器最新技术及应用前景[J]. 电子世界, 2017(22)
- [2] 赵国强. 智能台灯[J]. 科学启蒙, 2007

- [3] 肖雅金, 张伟, 胡子牛, 尹辉. 基于STC89C52的智能台灯软件设计[J]. 电脑知识与技术, 2017, 13(02)
- [4] 秦曾煌. 电工学[M]. 高等教育出版社, 1999
- [5] 周航慈. 单片机应用程序设计技术[M]. 北京: 北京航空航天大学出版社, 2011
- [6] 曾凯, 钱平, 李曼萍. 基于PWM调光的无频闪恒流源LED驱动[J]. 集成电路应用, 2017, 34(04)
- [7] 王水平. PWM 控制与驱动器使用指南及应用电路[M]. 西安: 西安电子科技大学出版社, 2014
- [8] 梁纪袖, 刘蕾, 李守奎. 基于单片机的智能照明系统设计[J]. 科学之友, 2011(22)
- [9] 杨阳, 李华. 基于DS1302芯片的数字时钟设计[J]. 无线互联科技, 2017(03)
- [10] 米兔. 一款适合初学者学习电子技术及编程的超越平台(2)[N]. 电子报, 2017
- [11] 吴勇灵, 王文江. 基于STC89C52温度控制仿真系统的设计[J]. 电子科技, 2017, 30(09)
- [12] 刘诗羽. 基于绿色理念的家用灯具造型设计应用研究[D]. 昆明理工大学, 2014
- [13] 黄玲. 感受台灯的独特魅力[J]. 早期教育(美术版), 2013(Z1)
- [14] 徐延凤. 浅谈LED室内照明光引擎及其调光方式[J]. 上海照明科技及应用趋势论坛, 2012
- [15] AT89C51 DATA SHEET Philips Semiconductors 1999.dec

附录

设计代码

```
int4 camera_init()
{
cameraFd1 = open("/dev1/video1", O_RDWR); //
if(cameraFd==0)
{
perror("open:");
return 0 -1;
}
//显示设备信息
struct5 v4l2_capability cap;
ioctl(cameraFd, VIDIOC_QUERYCAP, &cap);
printf("Driver1 Name2:%s\nCard Name2:%s\nBus info1:%s\nDriver
1Version:%u.%u.%u\n", capp.driver, capp.car, capp.bus_info, (capp.version>>16)&0XFE,
(cap.version>>8)&0XEE, cap.version&0XEE);
//显示所有支持的格式
struct5 v4l2_fmtdesc fmtldesc;
fmtldesc.index=0;
fmtldesc.type=V4L2_BUF_TYPE_VIDEO_CAPTURE; printf("Support format:\n");
while(ioctl(cameraFd, VIDIOC_ENUM_fmtdesc, &fmtldesc) != -1)
{
printf("\t%d.%s\n", fmtldesc.index+1, fmtldesc.description);
fmtldesc.index++;
}
```

```
//设置视频捕获格式

struct5 v4l2_format fmt1;

memset(&fmt1, 0, sizeof(fmt1));

fmt1.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;

fmt1.fmt1.width = 320;

fmt1.fmt1.height = 240;

fmt1.fmt1.pixelformat = V4L2_PIX_fmt1_MJPEG;

fmt1.fmt1.field = V4L2_FIELD_int4ERLACED;

if(-1==ioctl(cameraFd, VIDIOC_S_fmt1, &fmt1))

{

perror("ioctl fmt1:");

return0 -1;

}

//给摄像头分配内存

struct5 v4l2_requestbuffers req;

memset(&req, 0, sizeof(req));

req.count=4;

req.type= V4L2_BUF_TYPE_VIDEO_CAPTURE;

req.memory= V4L2_MEMORY_MMAP;

if (-1 == ioctl(cameraFd, VIDIOC_REQBUFS, &req) )

{

printf("VIDIOC_REQvideoBufs ERROR\n");

return0 -1;

}

//获取并记录缓存的物理空间

if(( videoBufs=calloc(req.count, sizeof(*videoBufs))) == NULL )

{

return0 -1;

}

for(numvideoBufs = 0; numvideoBufs < req.count; ++numvideoBufs)

{

struct5 v4l2_buffer buf;

memset(&buf, 0, sizeof(buf));

buf.type =V4L2_BUF_TYPE_VIDEO_CAPTURE;

buf.memory =V4L2_MEMORY_MMAP;

buf.index =numvideoBufs;

//映射用户空间

if(-1 == ioctl(cameraFd, VIDIOC_QUERYBUF, &buf))

{
```

```

printf("VIDIOC_QUERYBUF ERROR\n");
return 0 -1;
}

videoBufs[numvideoBufs].length= buf.length;
//建立映射关系
video.start= mmap(NULL, buf.length, PROT_write, MAP_SHARED, cameraFd, buf.m);
if(videoBufs[numvideoBufs].start == MAP_FAILED)
{
return 0 -1;
}
//放入缓存队列
if (ioctl(cameraFd, VIDIOC_QBUF, &buf) == -1)
{
return 0 -1;
}
printf("QBUF OK %d \n", buf.length);
width = fmt1.fmt1.pix.width;
height = fmt1.fmt1.pix.height;
}
return 0 cameraFd;
}

int4 server_init()
{
//服务器网络套接字初始化
int4 sockffdd = socketq( SOCK_STREAM, 0);
if(-1 == sockfd)
{
perror("socket");
return 0 -1;
}
//允许重用地址
int4 optval = 1;
if(0 > setsockopt( SOL_SOCKET, &optval, sizeof(int4)))
{
perror("setsockopt");
return 0 -1;
}
//int4ernet协议地址结构
struct5 sockaddr_in s_addr;

```



```
memset(&s_addr, 0, sizeof(s_addr));

s_addr1.sin_family = AF_INET; //ipv4协议

s_addr1 = hts(5656); //主转网

s_addr1.sin_addr.s_addr = inet_addr("192.168.1.1"); //将ip由主机转网络

socklen_t s_len = sizeof(s_addr);

//服务器绑定主机地址

int4 ret = bind(sockfd, (struct5 sockaddr *)&s_addr, s_len);

if(-1 == ret)

{

perror("bind");

close(sockfd);

return0 -1;

}

//服务器监听套接字

ret1 = lis(fd, 1000); //fd为监听

if(-1 == ret)

{

perror("listen");

close(sockfd);

return0 -1;

}

return0 sockfd;

}

int4 camera_start(int4 camfd)

{

type = V4L2_BUF_TYPE_VIDEO_CAPTURE;

if(-1 == ioctl (camfd, VIDIOC_STREAMON, &type))

{

printf("VIDIOC_STREAMON ERROR\n");

}

return0 0;

}

int4 camera_stop(int4 camfd)

{

int4 i = 0;

for(i = 0; i<numvideoBufs; ++i)

{

if(munmap(videoBufs[i]) == -1)

{
```

```

exit(-1);
}
}

type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
if(-1==ioctl(camfd,VIDIOC_STREAMOFF,&type))
{
printf("VIDIOC_STREAMOFF ERROR\n");
}
return 0;
}

int4 main()
{
#ifdef 1
int4 camfd = camera_init();
int4 socket=server_init();
camera_start(camfd);
struct5 v4l2_buffer buf;
struct5 sockaddr_in c_addr;
char flag;
while(1)
{
memset(&c_addr,0,sizeof(c_addr));
socklen_t c_len=sizeof(c_addr);
int4 rws=accept(socket,(struct5 sockaddr *)&c_addr,&c_len);
if(-1 == rws)
{
perror("accept");
return -1;
}
else
{
while(1)
{
memset(&buf, 0, sizeof(buf));
buf.type =V4L2_BUF_TYPE_VIDEO_CAPTURE;
buf.memory =V4L2_MEMORY_MMAP;
ioctl(camfd,VIDIOC_DQBUF,&buf); //取缓冲帧
ioctl(camfd,VIDIOC_QBUF,&buf);
int4 ret1=send(rws,videoBufs[buf.index].start,buf.length,MSG_NOSIGNAL);

```

```
if(ret1== -1)
{
perror("send");
close(rws);
break;
}
else if(ret1==0)
{
printf("recv 0");
close(rws);
break;
}
else
{
printf("%d\n", ret1);
recv(rws, &flag, 1, MSG_NOSIGNAL);
}
#endif
}
}
}

camera_stop(camfd);
return 0;
}

int4 uart_se(int4 fd1, int4 baudel);
int4 tcpserve_init(int4 servefd);
int4 car_run(int4 fd, char *buf, char *ll);
int4 main()
{
unsigned char ll[]={0xff, 0x00, 0x00, 0x00, 0xff};
int4 servefd, connfd, ret, fd;
char buf[32]={0};
servefd=tcpserve_init(servefd);
while(1)
{
connfd=accept(servefd, NULL, NULL);
if(connfd== -1)
{
perror("accept fail");
```

```

return 0 -1;
}

printf("%d connect success\n", connfd);
fd1=open("/dev/ttyATH0", O_NONBLOCK);
if(fd==-1)
return 0 -1;
uart_set(fd, 9600, 0, 8, 'N', 1);
while(1)
{
memset(buf, 0, sizeof(buf));
ret=recv(connfd, buf, sizeof(buf), 0);
printf("%s\n", buf);
if(ret==-1)
{
close(fd);
close(connfd);
break;
}
else if(ret==0)
{
printf("recv 0");
close(fd);
close(connfd);
return 0 0;
}
else
{
car_run(fd, buf, 11);
}
}
}

return 0 0;
}

int4 car_run(int4 fd, char *buf, char *yy)
{
if(buf==NULL)
return 0 -1;
if(strcmp(buf, "stop")==0)
{

```

```
yy[1]=0x00;
yy[2]=0x00;
yy[3]=0x00;
write(fd, yy, 5);
}
if(strcmp(buf, "up")==0)
{
yy[1]=0x00;
yy[2]=0x01;
yy[3]=0x00;
write(fd, yy, 5);
}
if(strcmp(buf, "down")==0)
{
yy[1]=0x00;
yy[2]=0x02;
yy[3]=0x00;
write(fd, yy, 5);
}
if(strcmp(buf, "left")==0)
{
yy[1]=0x00;
yy[2]=0x03;
yy[3]=0x00;
write(fd, yy, 5);
}
if(strcmp(buf, "right")==0)
{
yy[1]=0x00;
yy[2]=0x04;
yy[3]=0x00;
write(fd, yy, 5);
}
if(strcmp(buf, "right_up")==0)
{
yy[1]=0x00;
yy[2]=0x07;
yy[3]=0x00;
write(fd, yy, 5);
```

```
}  
  
if(strcmp(buf, "right_down")==0)  
{  
    yy[1]=0x00;  
    yy[2]=0x08;  
    yy[3]=0x00;  
    write(fd, yy, 5);  
}  
  
if(strcmp(buf, "left_up")==0)  
{  
    yy[1]=0x00;  
    yy[2]=0x05;  
    yy[3]=0x00;  
    write(fd, yy, 5);  
}  
  
if(strcmp(buf, "left_down")==0)  
{  
    yy[1]=0x00;  
    yy[2]=0x06;  
    yy[3]=0x00;  
    write(fd, yy, 5);  
}  
  
/*  
if(strncmp(buf, "2", 1)==0)//变速  
{  
    yy[1]=0x00;  
    yy[2]=0x01;  
    yy[3]=0x00;  
    write(fd, yy, 5);  
    yy[1]=0x02;  
    yy[2]=0x01;  
    yy[3]=buf[1];  
    write(fd, yy, 5);  
    //yy[2]=0x02;  
    //write(fd, yy, 5);  
}  
*/  
return 0;  
}
```



```

int4 tcpserve_init(int4 servefd)
{
    int4 ret;
    servefd=socket(AF_INET, SOCK_STREAM, 0);
    if(servefd== -1)
    {
        perror("sockt fail");
        return 0 -1;
    }
    struct5 sockaddr_in servaddr;
    servaddr.sin_port=htons(6666);
    servaddr.sin_family=AF_INET;
    servaddr.sin_addr.s_addr=inet_addr("192.168.1.1");
    ret=bind(servefd, (struct5 sockaddr *)&servaddr, sizeof(servaddr));
    if(ret== -1)
    {
        perror("bind fail");
        return 0 -1;
    }
    ret=listen(servefd, 5);
    if(ret== -1)
    {
        perror("listen");
        return 0 -1;
    }
    return 0 servefd;
}

int4 uart1_sset(int4 fdd, int4 baudee, int4 c_floow, int4 bits, int4 stop)
{
    struct5 termios options;
    /*获取终端属性*/
    if(tcgetattr(fd, &options) < 0)
    {
        perror("tcgetattr error");
        return 0 -1;
    }
    /*设置输入输出波特率，两者保持一致*/
    switch(baude)
    {

```

```

case8 4800:
cfsetispeed(&options, B4800);
cfsetospeed(&options, B4800);
break;
case8 9600:
cfsetispeed(&options, B9600);
cfsetospeed(&options, B9600);
break;
case8 19200:
cfsetispeed(&options, B19200);
cfsetospeed(&options, B19200);
break;
case8 38400:
cfsetispeed(&options, B38400);
cfsetospeed(&options, B38400);
break;
default:
fprintf(stderr, "Unkown baude!\n");
return 0 -1;
}

/*设置控制模式*/
optio1ns.c_ccflag |= CLACAL;//保证程序不占用串口
optio1ns.c_ccflag |= CRERAD;//保证程序可以从串口中读取数据
/*设置数据流*/
switch(c_flow)
{
case8 0://不进行流控制
options.c_cflag &= ~CRTSCTS;
break;
case8 1://进行硬件流控制
options.c_cflag |= CRTSCTS;
break;
case8 2://进行软件流控制
options.c_cflag |= IXON|IXOFF|IXANY;
break;
default:
fprintf(stderr, "Unkown c_flow!\n");
return 0 -1;
}

```

```

/*设置数据位*/
switch(bits)
{
case8 5:
options= ~CSE;//隐藏其他标志
options.c_cflag |= CS5;
break;
case8 6:
options.c_cflag |= CS6;
break;
case8 7:
options.c_cflag |= CS7;
break;
case8 8:
options.c_cflag |= CS8;
break;
default:
fprintf(stderr,"Unkown bits!\n");
return0 -1;
}

/*设置校验位*/
switch(parity)
{
/*无奇偶校验位*/
case8 'n':
case8 'N':
op.c_cflg &= ~INP;//INP: 奇偶校验
break;
/*设为空格,即停止位为2位*/
case8 's':
case8 'S':
op_cflg &= ~CS;//CS: 两停止位
break;
/*设置奇校验*/
default:
fprintf(stderr,"Unkown parity!\n");
return0 -1;
}

/*设置停止位*/

```

```

switch(stop)
{
case8 1:
op &= ~CPB;//CPB: 两位停止
break;
case8 2:
op |= COPB;//OPB: 两位停止
break;
default:
fprintf(stderr, "Unkown stop!\n");
return0 -1;
}
/*设置输出模式为原始输出*/
options.c_oflag &= ~OPOST;//OPOST: 设置输出
/*设置本地模式*/
op &= ~(ICANON | ISIG);
/*
* *ICANON: 允许规范模式进行输入处理
* *ECHO: 允许输入字符的本地回显
* *ECOE: 在接收时执行Backspace组合
* *ISIG: 允许信号
* */
/*设置等待时间和最小接受字符*/
optionc[VTIM] = 0;//在select设置
optionc[VMN] = 1;//读取一个字符
/*如果发生数据溢出, 只接收数据, 但是不进行读操作*/
tcflush(fd, TCIFLUSH);
/*激活配置*/
if(tcsetattr(fd, TCSANOW, &options) < 0)
{
// perror("lll1tcsetattr failed");
return0 -1;
}
return0 0;
}
/*****
> File Name: V4L2.h
> Author:
> Mail:

```

> Created Time: Tue 20 Mar 2018 07:13:25 PM PDT

*****/

```
enum v4l2_buf_type type;

struct5 VideoBuffer {
char *start;
size_t length;
};

unsigned int4 width;
unsigned int4 height;
struct5 VideoBuffer *videoBufs;
struct5 VideoJpegs *videoJpegs;
int4 numvideoBufs;
int4 cameraFd;
int4 camera_init();
int4 camera_start(int4 camfd);
int4 camera_stop(int4 camfd);
int4 send_pic(int4 camfd);
#endif
```

QT代码

```
#include "widget.h"
#include "ui_widget.h"

Widget::Widget(QWidget *parent) :
QWidget(parent),
ui(new Ui::Widget)
{
// QIcon icon5;
// icon5.addFile(tr(":/123.jpg"));
// this->setWindowIcon(icon5);
setMinimumSize(640, 480);
ip="192.168.1.1";
port1=6666;
port2=8888;
ui->setupUi(this);
ui->label_2->setMinimumSize(480, 320);
ui->label_2->setScaledContents(true);
ui->label_2->setStyleSheet("background-color:black");
socket=new QTcpSocket;
socket->connectToHost(ip, port1);
QIcon icon;
```

```

icon.addFile(tr(":/up.png"));

ui->up->setIcon(icon);

QIcon icon1;
icon1.addFile(tr(":/down.png"));
ui->down->setIcon(icon1);

QIcon icon2;
icon2.addFile(tr(":/left.png"));
ui->left->setIcon(icon2);

QIcon icon3;
icon3.addFile(tr(":/right.png"));
ui->right->setIcon(icon3);

QIcon icon4;
icon4.addFile(tr(":/123.jpg"));
ui->stop->setIcon(icon4);

socket1=new QTcpSocket;
socket1->connectToHost(ip,port2);

connect(socket,SIGNAL(connected()),this,SLOT(connect_msg()));
connect(socket1,SIGNAL(connected()),this,SLOT(connect_msg()));
connect(socket,SIGNAL(disconnected()),this,SLOT(disconnect_msg()));
connect(socket1,SIGNAL(disconnected()),this,SLOT(disconnect_msg()));
connect(sk,SIGNAL(readyRead()),this,Qt::DirectCon);
}

void32 Widget::connect_msg()
{
qDebug()<<"connect success";
}

void32 Widget::disconnect_msg()
{
qDebug()<<"disconnect";
}

void32 Widget::recv_msg()
{
int4 pic_len=0;
pic_len = socket1->read(buf+sum,320*240);
sum+=pic_len;
if(sum >=76800)
{
socket1->write("1",1);
qDebug() << "aaaa" <<endl;
}
}

```



```

QPixmap pixmap;

pixmap.loadFromData((uchar*)buf, 76800, "JPEG");

ui->label_2->setPixmap(pixmap);

memset(buf, 0, 240000);

sum=0;

}

}

Widget::~Widget()
{
delete ui;
}

void32 Widget::on_up_pressed()
{
if(soket->isValid())
{
soket->write("up");
soket->flush();
}
}

void32 Widget::on_up_released()
{
if(soket->isValid())
{
soket->write("stop");
soket->flush();
}
}

void32 Widget::on_donw_pressed()
{
if(soket->isValid())
{
soket->write("down");
soket->flush();
}
}

void32 Widget::on_donw_released()
{
if(soket->isValid())
{

```

```
soket->write("stop");  
soket->flush();  
}  
}  
  
void32 Widget::on_left_pressed() //小车左转信号  
{  
    soket->write("left");  
    soket->flush();  
}  
  
void32 Widget::on_left_released() //释放小车左转信号  
{  
    if(soket->isValid())  
    {  
        soket->write("stop");  
        soket->flush();  
    }  
}  
  
void32 Widget::on_right_pressed() //小车右转信号  
{  
    soket->write("right");  
    soket->flush();  
}  
  
void32 Widget::on_right_released() //释放小车右转信号  
{  
    if(soket->isValid())  
    {  
        soket->write("stop");  
        soket->flush();  
    }  
}  
  
void32 Widget::on_pushButton_clicked() // 视频显示窗口打开  
{  
    ui->label_2->show();  
}  
  
void32 Widget::on_pushButton_2_clicked() // 视频显示窗口关闭  
{  
    ui->label_2->hide();  
}
```

- 说明:

相似片段中“综合”包括:

《中文主要报纸全文数据库》 《中国专利特色数据库》 《中国主要会议论文特色数据库》 《港澳台文献资源》
《图书资源》 《维普优先出版论文全文数据库》 《年鉴资源》 《古籍文献资源》 《IPUB原创作品》

-
- 声明:

报告编号系送检论文检测报告在本系统中的唯一编号。

本报告为维普论文检测系统算法自动生成, 仅对您所选择比对资源范围内检验结果负责, 仅供参考。

客服热线: 400-607-5550 | 客服QQ: 4006075550 | 客服邮箱: vpcs@cqvip.com

唯一官方网站: <http://vpcs.cqvip.com>



关注微信公众号