《嵌入式系统与接口技术》

姓 名： 周羲阳 班 号： 193122

学 号：20121001542 院（系）： 计算机学院

专 业： 网络工程 指导教师： 陈小岛

目录

[第一章 需求分析 3](#_Toc421597104)

* 1. [问题背景 3](#_Toc421597105)
  2. [功能需求 3](#_Toc421597106)

[第二章 规格说明 3](#_Toc421597107)

* 1. [系统基本组成 3](#_Toc421597108)
  2. [工作框图 3](#_Toc421597109)

[第三章 硬件设计 4](#_Toc421597110)

* 1. [硬件需求 4](#_Toc421597111)
     1. [树莓派 4](#_Toc421597112)
     2. [sd卡 4](#_Toc421597113)
     3. [无线网卡 5](#_Toc421597114)
     4. [L298N电机驱动板 5](#_Toc421597115)
     5. [红外线遥控器及接收器 6](#_Toc421597116)
     6. [小车底盘 6](#_Toc421597117)
     7. [其他 6](#_Toc421597118)
  2. [成本 6](#_Toc421597119)
  3. [接线图解 7](#_Toc421597120)

[第四章 软件设计及实现步骤 8](#_Toc421597121)

* 1. [配置无线网卡 8](#_Toc421597122)
  2. [修改/boot/config.txt 8](#_Toc421597123)
  3. [更新源列表 9](#_Toc421597124)
  4. [安装wiringPi 9](#_Toc421597125)
  5. [安装lirc 9](#_Toc421597126)
  6. [设置lirc 9](#_Toc421597127)
  7. [记录按键 10](#_Toc421597128)
  8. [测试按键 10](#_Toc421597129)
  9. [终端命令控制小车运动 10](#_Toc421597130)
     1. [小车前进 11](#_Toc421597131)
     2. [小车后退 11](#_Toc421597132)
     3. [小车左转 11](#_Toc421597133)
     4. [小车右转 11](#_Toc421597134)
     5. [小车停止 11](#_Toc421597135)
  10. [遥控器按键控制小车运动 12](#_Toc421597136)
  11. [进一步优化 13](#_Toc421597137)
      1. [start.sh 13](#_Toc421597138)
      2. [setup.sh 13](#_Toc421597139)
      3. [Irwgo.sh 13](#_Toc421597140)
      4. [forward.sh 13](#_Toc421597141)
      5. [backward.sh 13](#_Toc421597142)
      6. [left.sh 14](#_Toc421597143)
      7. [right.sh 14](#_Toc421597144)
      8. [stop.sh 14](#_Toc421597145)
      9. [go.sh 14](#_Toc421597146)
  12. [集成对树莓派的配置操作 14](#_Toc421597147)
      1. [install.sh 14](#_Toc421597148)
  13. [进一步简化操作 16](#_Toc421597149)

# 第一章 需求分析

## 问题背景

树莓派(Raspberry Pi)是一种便宜的卡片式 Linux 电脑，风靡全球。它是世界上众多廉价又成熟的电脑之一，而且十分容易上手。树莓派有成千上万种玩法，例如[备份邮件](http://pyropus.ca/software/getmail/)、[搭建个人 RSS 阅读器](http://magnatecha.com/replacing-google-reader-with-rss2email/)、搭建网站、家用网络存储、站点监控、事件提醒、家庭相册、点唱机、安全摄像头等等。

## 功能需求

实现小车的前进、后退、左右转向

# 第二章 规格说明

## 系统基本组成

红外线树莓派小车主要由4部分组成：红外模块、树莓派、电源和小车。红外模块通过向树莓派发出红外信号间接控制小车，树莓派根据接收到不同的红外信号，进行相应的处理，向小车输出特定的电平，直接控制小车，电源这向小车提供能量。

## 工作框图

# 第三章 硬件设计

## 硬件需求

### 树莓派



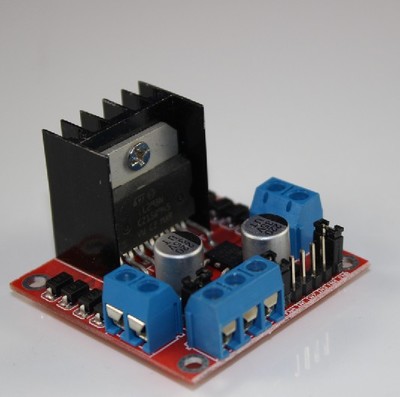
### sd卡



### 无线网卡



### L298N电机驱动板



此驱动板含有4个in口in1、in2分别接收高电平和低电平电机1正转，反之亦然，

In1、in2都低电平这电机不转；in3、in4类似。

### 红外线遥控器及接收器



### 小车底盘



### 其他

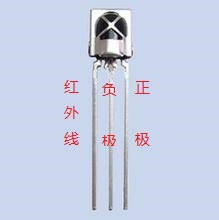
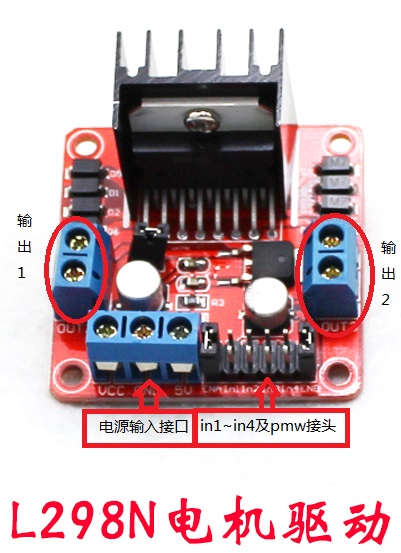
杜邦线（公对公、母对母、公对母）、铜柱、电池组（或移动电源+USB转接线）

## 成本

树莓派（约200）+sd卡（30）+无线网卡（33）+驱动板（9×3）+遥控器及接收器（5×2）+底盘、电机及铜柱（38）+杜邦线（5+5+10）+邮费（8+10+5+8）=200+189

## 接线图解





# 第四章 软件设计及实现步骤

## 配置无线网卡

使用vi编辑器,打开interfaces

sudo vi  /etc/network/interfaces

将内容：

auto lo

iface lo inet loopback

iface eth0 inet dhcp

auto wlan0

allow-hotplug wlan0

iface wlan0 inet dhcp

wpa-roam /etc/wpa\_supplicant/wpa\_supplicant.conf

改为：

auto lo

iface lo inet loopback

iface eth0 inet dhcp

auto wlan0

allow-hotplug wlan0

iface wlan0 inet dhcp

wpa-ssid "ssid"

wpa-psk "key"

#wpa-roam /etc/wpa\_supplicant/wpa\_supplicant.conf

Reboot后连接相应的无线网络

## 修改/boot/config.txt

使用vi编辑器,打开config.txt

Sudo vi /boot/config.txt

在其最后添加：

dtoverlay=lirc-rpi

保存后重启生效

注：如果没有此操作，后面的modprobe lirc\_rpi会报错

## 更新源列表

nameserver 114.114.114.114 > /etc/resolv.config

deb http://mirrors.aliyun.com/raspbian/raspbian/ wheezy main non-free contrib > /etc/apt/sources.list

deb-src http://mirrors.aliyun.com/raspbian/raspbian/ wheezy main non-free contrib >> /etc/apt/sources.list

apt-get update

配置dns服务器地址后，在aliyun上跟新源列表

## 安装wiringPi

cd /opt

rm -rf wiringPi

git clone git://git.drogon.net/wiringPi

cd wiringPi

./build

gpio –v

## 安装lirc

apt-get install lirc

## 设置lirc

modprobe lirc\_rpi gpio\_in\_pin=18

sed -i "4cLIRCD\_ARGS=\"--uinput\"" /etc/lirc/hardware.conf

sed -i "16cDRIVER=\"default\"" /etc/lirc/hardware.conf

sed -i "18cDEVICE=\"/dev/lirc0\"" /etc/lirc/hardware.conf

第一个命令是将BCM编码方式下的GPIO 18口作为输入口（联系连线图解可知，该口是作为红外信号输入口），设置后可输入mode2 -d /dev/lirc0命令进行测试，如果按遥控器任意按键后出现若干行含pluse和space的输出，就说明设置成功。

## 记录按键

/etc/init.d/lirc stop

irrecord -n -d /dev/lirc0 /etc/lirc /lircd.conf

/etc/init.d/lirc start

第二行为记录命令，在此之前须要先关闭lirc，记录结束后再启动。这个过程比较复杂，首先是要求按住遥控器上任意一个按键不放，直到屏幕上出现一整行点，然后要求用一个比较合适的频率按遥控器上任意一个按键，直到屏幕上出现一整行点。

之后要求输入一个按键名称（如：forward），回车后按一下遥控器上的一个按键（如：2）

以此循环直到完成所有按键的记录。

注：我们一共记录了2、4、6、8、5共五个按键，分别对应forward、left、right、backward和stop。

## 测试按键

Irw

输入此命令后按遥控器上的2、4、6、8、5按键，终端分别会输出类似：

Xxxxxxxxxxxxxxxxxxxxx 00 forward /boot/lircd.conf

Xxxxxxxxxxxxxxxxxxxxx 00 left /boot/lircd.conf

Xxxxxxxxxxxxxxxxxxxxx 00 right /boot/lircd.conf

Xxxxxxxxxxxxxxxxxxxxx 00 backward /boot/lircd.conf

Xxxxxxxxxxxxxxxxxxxxx 00 stop /boot/lircd.conf

主要是每行第三个内容有区别，该内容也是我们在之后的步骤中需要的

**到此我们在树莓派上的配置就基本结束了**

我们还可以将如下内容写到/etc/rc.local里面，便于下次使用。

modprobe lirc\_rpi gpio\_in\_pin=18 gpio\_out\_pin=2

/etc/init.d/lirc start

cd /opt/smartcar-shell/bin

./start.sh \&

## 终端命令控制小车运动

/usr/local/bin /gpio mode 0 out

/usr/local/bin /gpio mode 6 out

/usr/local/bin /gpio mode 4 out

/usr/local/bin /gpio mode 5 out

这里是将wpi编码方式下的GPIO 0、6、4、5作为输出口

### 小车前进

/usr/local/bin/gpio write 0 0

/usr/local/bin/gpio write 6 1

/usr/local/bin/gpio write 4 0

/usr/local/bin/gpio write 5 1

### 小车后退

/usr/local/bin/gpio write 0 1

/usr/local/bin/gpio write 6 0

/usr/local/bin/gpio write 4 1

/usr/local/bin/gpio write 5 0

### 小车左转

/usr/local/bin/gpio write 0 1

/usr/local/bin/gpio write 6 0

/usr/local/bin/gpio write 4 0

/usr/local/bin/gpio write 5 1

### 小车右转

/usr/local/bin/gpio write 0 0

/usr/local/bin/gpio write 6 1

/usr/local/bin/gpio write 4 1

/usr/local/bin/gpio write 5 0

### 小车停止

/usr/local/bin/gpio write 0 0

/usr/local/bin/gpio write 6 0

/usr/local/bin/gpio write 4 0

/usr/local/bin/gpio write 5 0

**以上组五组命令分别给wpi编码方式下的GPIO 0、6、4、5接口相应的高电平或者低电平实现小车的运动（结合L298N电机驱动板和接线图解不难理解）**

## 遥控器按键控制小车运动

首先我们将上一步骤的命令做成shell文件，以前进为例

shell名：

forward.sh

shell内容：

/usr/local/bin/gpio write 0 0

/usr/local/bin/gpio write 6 1

/usr/local/bin/gpio write 4 0

/usr/local/bin/gpio write 5 1

注：其他方向同理分别为backward.sh、left.sh、right.sh和stop.sh

然后终端输入

irw | `./$3.sh`

联系之前的按键记录和irw命令可知 当我们按下按键“2时”终端会输出：

Xxxxxxxxxxxxxxxxxxxxx 00 forward /boot/lircd.conf

而此时因为“|”命令，只会将其作为`./$3.sh`的输入，将$3替换成forward后，执行的命令为./forward.sh，所有此时小车会前进。

同理当我们按下按钮4、6、8、5是小车分别会左转、右转、后退、停止。

## 进一步优化

生成下来9个shell文件：

### start.sh

./setup.sh

irw | xargs -l ./irwgo.sh

### setup.sh

gpio\_path=/usr/local/bin

#right wheel

$gpio\_path/gpio mode 0 out

$gpio\_path/gpio mode 6 out

#left wheel

$gpio\_path/gpio mode 4 out

$gpio\_path/gpio mode 5 out

### Irwgo.sh

`./$3.sh`

### forward.sh

./go.sh 0 1 0 1

### backward.sh

./go.sh 1 0 1 0

### left.sh

./go.sh 1 0 0 1

### right.sh

./go.sh 0 1 1 0

### stop.sh

./go.sh 0 0 0 0

### go.sh

gpio\_path=/usr/local/bin

#right wheel

$gpio\_path/gpio write 0 $1

$gpio\_path/gpio write 6 $2

#left wheel

$gpio\_path/gpio write 4 $3

$gpio\_path/gpio write 5 $4

## 集成对树莓派的配置操作

### install.sh

echo config dns

echo nameserver 114.114.114.114 > /etc/resolv.config

echo config apt source aliyun

echo deb http://mirrors.aliyun.com/raspbian/raspbian/ wheezy main non-free contrib > /etc/apt/sources.list

echo deb-src http://mirrors.aliyun.com/raspbian/raspbian/ wheezy main non-free contrib >> /etc/apt/sources.list

echo apt update

apt-get update

echo all install to /opt directory

echo instal gpio wiringPi

cd /opt

rm -rf wiringPi

git clone git://git.drogon.net/wiringPi

cd wiringPi

./build

gpio -v

echo install lirc

apt-get install lirc -y

echo config gpio use pin 18

modprobe lirc\_rpi gpio\_in\_pin=18 gpio\_out\_pin=2

#echo test

#echo mode2 -d /dev/lirc0

echo config

sed -i "4cLIRCD\_ARGS=\"--uinput\"" /etc/lirc/hardware.conf

sed -i "16cDRIVER=\"default\"" /etc/lirc/hardware.conf

sed -i "18cDEVICE=\"/dev/lirc0\"" /etc/lirc/hardware.conf

echo record buttons

/etc/init.d/lirc stop

rm ~/lircd.conf

irrecord -n -d /dev/lirc0 ~/lircd.conf

mv ~/lircd.conf /etc/lirc/lircd.conf

/etc/init.d/lirc start

#echo test

#echo irw

echo install car

cd /opt

echo start smart car

cd smartcar-shell/bin

chmod +x \*.sh

echo start car on boot

sed -i '$d' /etc/rc.local

echo start car on boot >> /etc/rc.local

echo modprobe lirc\_rpi gpio\_in\_pin=18 gpio\_out\_pin=2 >> /etc/rc.local

echo /etc/init.d/lirc start >> /etc/rc.local

echo cd /opt/smartcar-shell/bin >> /etc/rc.local

echo ./start.sh \& >> /etc/rc.local

echo exit 0 >> /etc/rc.local

./start.sh

## 进一步简化操作

将前面两个步骤中中生成的shell文件分别装入bin文件夹和install文件夹，在将它们装入smartcar-shell文件夹中上传到github上，之后如果用另一台树莓派制作小车将变得非常简单。

制作流程如下

组装

1. git clone git://github.com/…… /smartcar-shell/
2. sudo chmod +x ~~/smartcar-shell/instell/install.sh 注：“~~”代表smartcar-shell所在目录
3. sudo ~~/smartcar-shell/instell/install.sh 注：此过程需要记录按键
4. cd ~~/smartcar-shell/bin/
5. ./start.sh

# 总结

本次课程设计，我们主要是一起合作，购买硬件、共同组装，软件安装、基本设置、接线等等。本小组在小车的软件安装过程中遇到了几个问题，首先遇到的问题在输入modprobe lirc\_rpi gpio\_in\_pin=XX（将树莓派相应的gpio口设置为红外信号接收口）时报错。反复在网上查阅资料后发现需要在此前修改/boot/config.txt，在里面加一行 dtoverlay=lirc-rpi，具体原因还不清楚，但却是可以解决报错的问题。然后在记录按键的时候由于操作比较复杂，也是尝试了多次后成功的。