

扬州市职业大学电气与汽车工程学院

毕业设计说明书(论文)

作 者： 郭 力 学 号： 100201210

教 研 室： 机电教研室

专 业： 机电一体化

题 目： 无线视频监控智能小车

指导者： 于 标

评阅者： _____

2013 年 5 月

扬州市职业大学电气与汽车工程学院

毕业设计（论文）评语

学生姓名： 郭 力 班级、学号： 10 机电 2 班 100201210

题 目： 无线视频监控智能小车

综合成绩： _____

指导者评语：

指导者(签字)： _____

年 月 日

毕业设计（论文）评语

评阅者评语：

评阅者(签字): _____
年 月 日

答辩委员会（小组）评语：

答辩委员会负责人(签字): _____
年 月 日

专科毕业设计说明书（论文）

毕业设计说明书（论文）中文摘要

摘要：无线传输视频监控智能车专业知识涉及控制、模式识别、传感技术、无线通讯、电气、计算机、机械等诸多学科。在整个智能车控制系统中，如何准确地运用无线通讯及时的对智能车的速度和方向进行控制是整个控制系统的关键。

本文首先对智能车的硬件进行设计，达到了低重心、大前瞻、高稳定性的目标。其次对系统的软件部分进行设计。WIFI 智能小车由电机、小车车体、单片机控制芯片、WIFI 收发模块、电机驱动、舵机、电源、摄像头等主要部件以及灯光、蜂鸣器、电平转换等辅助模块构成。WIFI 智能小车利用笔记本或手机等能连接无线路由器的终端智能设备连接到路由器，通过应用软件显示路由器上摄像头上采集到的视频信号，再由这些智能的终端设备发送控制指令到无线路由器，通过无线路由器将指令传送给单片机进行处理。然后经过单片机控制电机驱动驱动电机转动、舵机转动，从而实现控制小车的运动及视频采集。

整个系统涉及车体机械结构调整、传感器电路设计及信号处理、控制算法和策略优化等多个方面。经过大量测试，最终确定了现有的系统结构和各项控制参数。

关键词：智能车，路由器，wifi，单片机

目录

第一章 绪论	1
第二章 总体方案设计	2
2.1 设计方案概述	2
2.1.1 控制系统	2
2.1.2 电源系统	3
2.1.3 整车布局	4
2.1.4 智能车相关性能设计思想	5
2.2 智能小车设计的主要内容	5
第三章 硬件设计	7
3.1 路由器的选型及安装方案设计	7
3.1.1 路由器的选型	7
3.1.2 路由器的供电电路	10
3.1.3 路由器刷机固件介绍	11
3.1.4 刷机步骤	11
3.1.5 路由测试	13
3.2 摄像头的选型和安装方案设计	14
3.2.1 摄像头简介	14
3.2.2 摄像头的分类	14
3.2.3 摄像头的工作原理	15
3.2.4 摄像头的安装设计	15
3.3 智能车车体设计	17
3.3.1 车体布局	17
3.3.2 摄像头云台的设计安装	17
3.3.3 主板安装	19
3.4 电路设计	19
3.4.1 电源模块	21
3.4.2 最小系统模块	23
3.4.3 时钟电路	26
3.4.4 复位电路	27
3.4.5 无线通讯模块	27
3.4.6 串口模块	30
3.4.7 电机驱动模块	31
3.4.8 抗干扰处理	35
3.5 本章小结	36
第四章 软件设计	37
4.1 视频信号采集	37
4.1.1 系统程序简介	37
4.2 程序流程图	37
4.2.1 主程序流程图	37
4.2.2 串口中断子函数流程图	39

4.2.3 定时器 2 中断子函数流程图	40
4.2.4 避障子函数流程图	41
4.3 PWM 调速	41
4.3.1 直流电机调速原理	41
4.3.2 PWM 基本原理及其实现方法	43
4.4 本章小结	46
第五章 智能车的开发与调试	47
5.1 编译环境	47
5.1.1 仿真测试	48
5.2 实物调试	48
5.3 单片机的资源分配	49
5.4 本章小结	50
第六章 总结与展望	51
致 谢	52
参考文献	53
附件 A: 智能车程序代码	54
附件 B: 智能车控制板原理图	73

第一章 绪论

智能车即轮式移动机器人，是一种集环境感知、决策规划、自动行驶等功能于一体的综合智能系统，智能车集中地运用了自动控制、模式识别、传感器技术、汽车电子、电气、计算机、机械等多个学科的知识。随着控制技术、计算机技术和信息技术的发展，智能车在工业生产和日常生活中已经扮演了非常重要的角色。近年来，智能车在野外、道路、现代物流及柔性制造系统中都有广泛运用，已成为人工智能领域研究和发展的热点。目前，智能车领域的研究已经能够在具有一定标记的道路上为司机提供辅助驾驶系统甚至实现无人驾驶，这些智能车的设计通常依靠特定道路标记完成识别，通过推理判断模仿人工驾驶进行操作。本文所述智能车就是一种自动导引小车，能够在给定的区域内沿着轨迹自动进行行进。

1953年，美国Barrett Electric公司制造了世界上第1台采用埋线电磁感应方式跟踪路径的自动导向车，也被称作“无人驾驶牵引车”（automated guided vehicle，简称AGV）。这些自动导向车主要用于自动化仓贮系统和柔性装配系统的物料运输。20世纪60年代和70年代初，AGV仍采用这种导向方式。但是，20世纪70年代中期，具有载货功能的AGV在欧洲得到了应用并被引入到美国。这些自动导向车主要用于自动化仓储系统和柔性专配系统的物料运输。在20世纪70年代和80年代初，“智能车”的应用领域扩大而且工作条件也变得多样化，因此，新的导向方式和技术得到了更广泛的研究与开发。自动导向无轨行走车辆常用蓄电池作为动力源，它是机电一体化的典型。AGV技术在汽车工业上有着广泛的用途，欧洲和美、日等国的汽车生产工业在80年代就开始大量使用AGV技术，现已成为比较成熟的技术。

智能车有着极为广泛的应用前景。结合传感器技术和自动驾驶技术可以实现汽车的自适巡航并把车开得又快又稳、安全可靠；汽车夜间行驶时，如果装上红外摄像头，就能实现夜晚汽车的安全辅助驾驶；他也可以工作在仓库、码头、工厂或危险、有毒、有害的工作环境里，此外他还能担当起无人值守的巡逻监视、物料的运输、消防灭火等任务。在普通家庭轿车消费中，智能车的研发也是有价值的，比如雾天能见度差，人工驾驶经常发生碰撞，如果用上这种设备，激光雷达会自动探测前方的障碍物，电脑会控制车辆自动停下来撞车就不会发生了。

第二章 总体方案设计

2.1 设计方案概述

本课题选取路由器作为无线通讯设备，最终要实现智能车的远程控制及监控功能。本文主要对智能车的控制系统和机械结构进行设计，研究内容就是基于单片机设计一部 WIFI 智能小车，小车最终实现视频采集、无线通讯、自动避障，远程控制。下面将对智能车的整个系统进行概述。

2.1.1 控制系统

控制核心单片机的选择

方案一：

采用ATmega16L单片机作为核心处理器，Atmega16L是基于增强的AVR RISC（精简指令集）结构的低功耗 8 位CMOS MCU。其数据吞吐率可高达 1Mips/MHz,具有缓减系统功耗与处理速度之间矛盾的优点。ATmega16L具有 16K字节的系统内可编程Flash(具有同时读写的能力，即RWW)，512 字节EEPROM，1K 字节SRAM，32 个通用I/O 口线，32 个通用工作寄存器，用于边界扫描的JTAG 接口，支持片内调试与编程，三个具有比较模式的灵活的定时器/计数器(T/C),片内/外中断，可编程串行USART，有起始条件检测器的通用串行接口,8 路 10 位具有可选差分输入级可编程增益(TQFP 封装) 的 ADC ，具有片内振荡器的可编程看门狗定时器，一个SPI 串行端口，以及六个可以通过软件进行选择的省电模式。

方案二：

采用STC12C5A60S2 单片机作为核心处理器，STC12C5A60S2 由宏晶公司推出，是一种低功耗，高性能的 8 位CMOS MCU，具有 8K在系统可编程Flash存储器。与工业 C51 系列指令和引脚完全兼容。片上Flash允许程序存储器在系统可编程，亦适于 常规编程器。在单芯片上，拥有灵巧的 8 位CPU 和在系统 可编程Flash，使得AT89S52 为众多嵌入式控制应用系统提 供高灵活、超有效的解决方案。 STC12C5A60S2 具有以下标准功能： 8k字节Flash，256 字节RAM， 32 位I/O 口线，看门狗定时器，2 个数据指针，三个 16 位 定时器/计数器，一个 6 向量 2 级中断结构，全双工串行口， 片内晶振及时钟电路。

鉴于目前对单片机的掌握熟悉程度，本系统使用 C51 系列兼容的 STC12C5A60S2 单片机作为系统核心处理器。

智能车系统包括路由器、单片机最小系统、电机驱动电路、电机、电平转换电路、电源电路、舵机、摄像头、蜂鸣器电路、灯光电路等，布局如图 2.1 所示。

路由器用于接收电脑等终端设备发送的指令和将摄像头采集到的视频信号传送到电脑等终端设备；单片机最小系统是整个小车的控制核心，控制着各个模块统一协调工作；电机驱动电路用于驱动电机转动，可以使电机产生正转、反转，从而使车体产生前进、后退、转弯等动作；电机的作用就是使整个车体运动；电平转换电路是将路由器输出的 3.3V 串口电平转换为单片机能判断的高低电平；电源电路作用是为整个小车系统供电；舵机制作成云台，然后装上摄像头，使摄像头可实现上下、水平方向 180 度旋转调节；摄像头用于采集视频信号；蜂鸣器电路用于报警提示；照明电路为小车在夜间行驶提供照明。

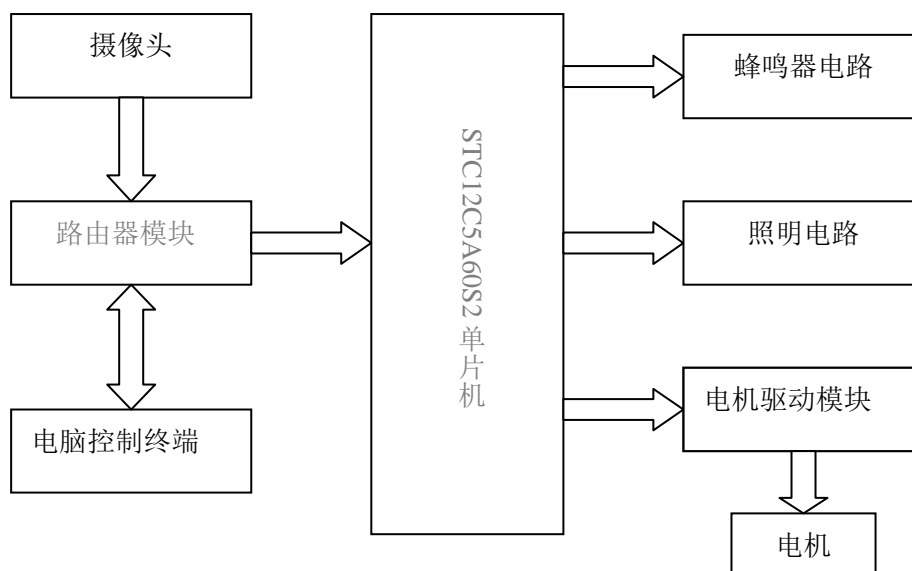


图 2.1 智能车系统模块连接图

2.1.2 电源系统

智能车在电源设计上，设计了两个电源输入接口，使系统板控制小车的电源，与供电给路由的电源独立分开。一方面可以为路由提供稳定的电压，另一方面可以减少路由与单片机相互工作间的干扰 如图2.2所示

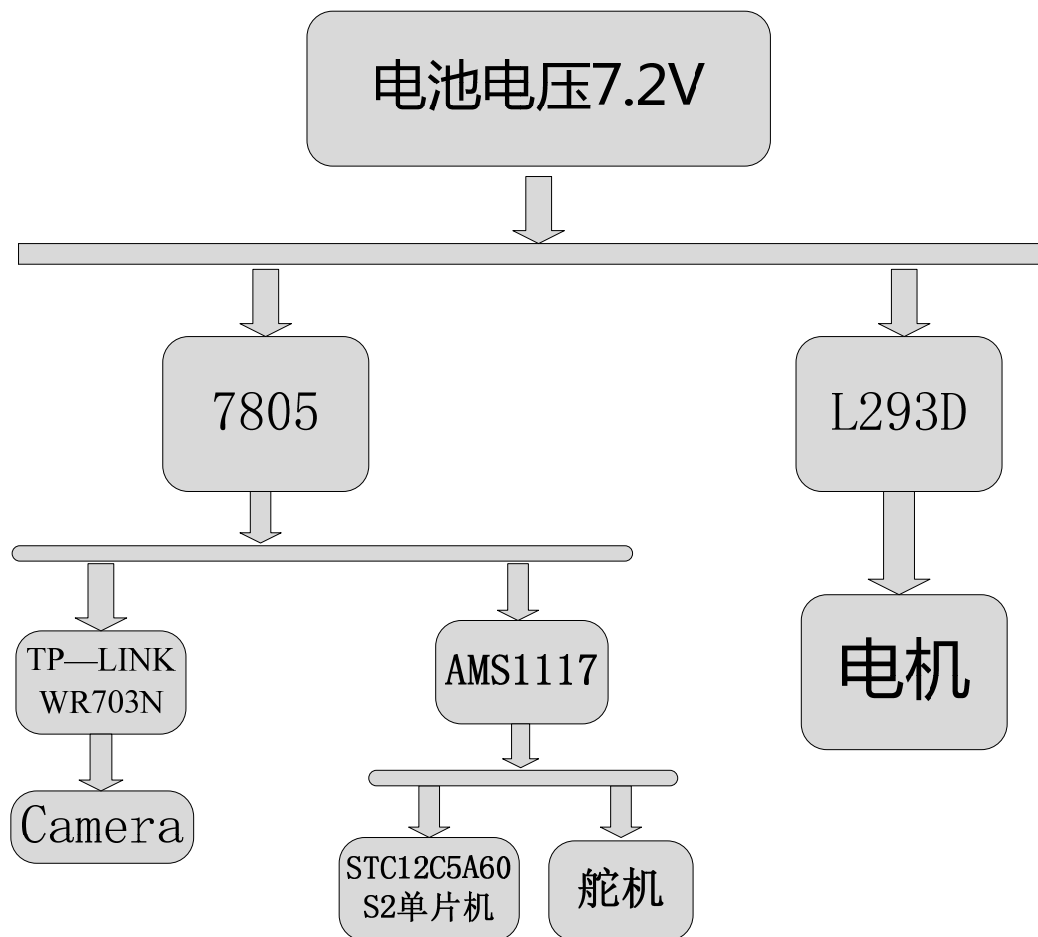


图2.2 智能车的电源系统

2.1.3 整车布局

智能车采用低重心紧凑型设计，并本着轻质量、大前瞻的原则，经过将近三个月的时间，比较多套方案，采用了低位主板的布局以降低重心，同时设计了两个自由度的云台系统，为了保证视频信号的实时传送及缩小空间采用了 TP-linkWR703N 路由器，最终设计的车形如图 2.3 所示

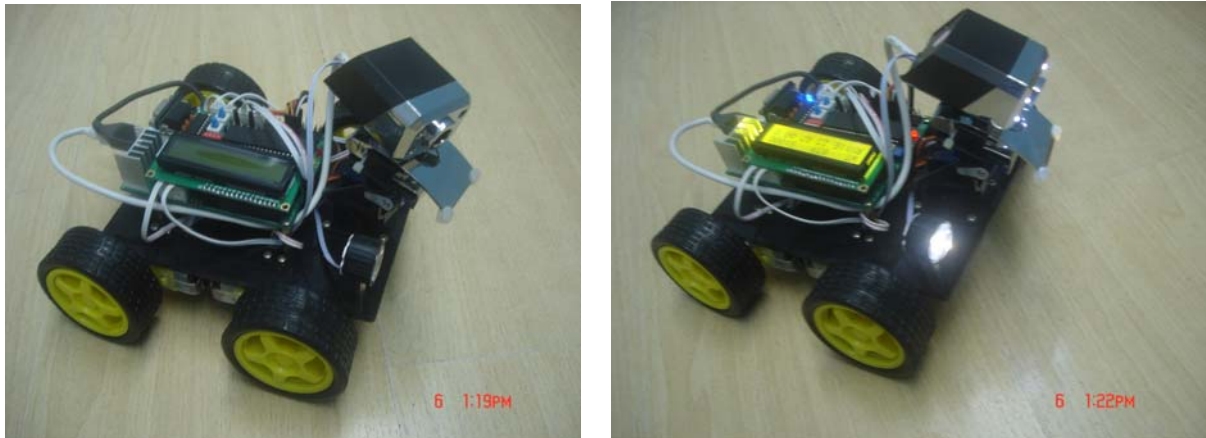


图 2.3 整车布局

2.1.4 智能车相关性能设计思想

智能车在设计过程中除了考虑无线通讯、操纵稳定性、视频信号等关键指标外，还应充分考虑开发、调试过程中可能遇到的种种问题。在智能车的设计、制造和配套设施及保证可靠性上做出如下要求：

- 1、保障智能车的可靠性，增加保护设计；
- 2、提高智能车的维护性，如预留备件、拆装尽可能方便、舵机重复定位性好等；
- 3、提高智能车和电池的续航能力，提高调试效率；
- 4、建立合理的调试辅助装备配套机制：维修工具、备件等；
- 5、建立智能车调试文档，包括硬件使用、调整说明、软硬件调试记录等。

2.2 智能小车设计的主要内容

在整个智能车控制系统中，如何准确地运用无线通讯及时传输视频信号并对智能车的速度和方向进行控制是整个控制系统的关键。

智能车系统主要由无线监测终端和 PC 上位机两部分组成。无线终端的小车车体是利用亚克力材料设计组装完成的。小车前端的摄像头对周围的环境进行检测，同时把视频信号传回给 PC 上位机。通过 PC 上位机可以随时控制小车的行进方向，而且可以操作摄像头随意上下、左右移动，扩大小车拍摄视野。另外在小车上加装了红外壁障传感器，小车可以对周围的障碍物进行探测，同时上位机上实时显示检测到的数据。

视频信号的采集常用的装置器件主要是摄像头。与图像传感器相比，摄像头的实时性更好，能反应所检测的整体信息。图像传感器的前瞻性有了较大提高，不过在检测整体信息方面摄像头还是占优势。总体而言，摄像头与图像传感器比其来，摄像头要功耗小而且方便的多，故采用摄像头作为视频信号采集传感器。

根据智能车常用的单片机的资源配置和开发方法，设计电路图和相应的系统软件，构建完整的无线视频监控小车系统。目的是为了让智能车以最短的响应时间，实现无线监控和实时控制。研究内容主要有以下几个方面：

首先，介绍了研究背景，阐述了控制系统的资源配置、资源需求与分配和核心处理器的寄存器，STC12C5A60S2 单片机寄存器资源。相比于其它类型的单片机，STC12C5A60S2 的功能更加强大，能够很好地满足智能车控制系统的需要。

其次，设计了智能车控制系统的硬件电路，包括各个模块的电路设计方案以及相关电路。采用的方案以STC12C5A60S2单片机为核心，包括总体控制系统的设计，各部件需要的供电电源设计，传感器电路设计，驱动电路的设计等。

然后，阐述了智能车机械结构调整和车体的安装。车体设计的核心思想是低重心，轻质量，提升智能车实时传输性和开阔的视野性能。智能车机械结构调整主要是调节车的重心、前轮、后轮，使智能车在行走时，更加稳定。

最后，进行了软件和算法的设计。根据视频传感器采集的信息，经无线通讯传回PC上位机之后，通过PC上位机给以单片机控制器指令，调节驱动电机和舵机动作，最后实现智能车实时监控的远程操作。

第三章 硬件设计

3.1 路由器的选型及安装方案设计

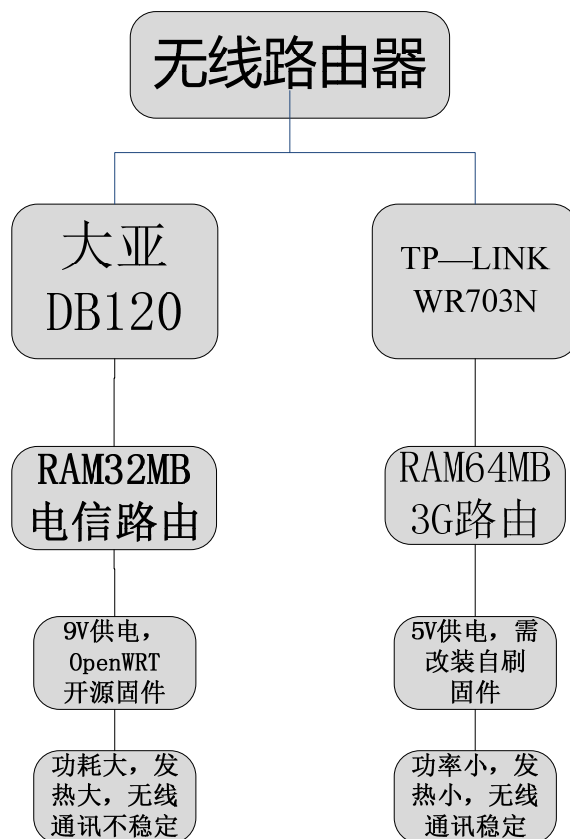


图 3.1 DB120 和 TP-LinK 路由器的比较

3.1.1 路由器的选型

市场上常见的能挂载摄像头路由器有大亚DB120和TP-LinkWR703两类，不管DB120或TP-LinkWR703，基本上两者都是基于Linux系统，通过开源的系统刷新路由的固件。这种刷入开源OpenWrt系统的路由就相当于一个运行了Linux系统的小型电脑。同时在路由中刷入mjpg-streamer程序，通过此程序将USB摄像头的视频进行编码，然后通过wifi返回给上位机，PC机就会出现来自智能车的视频了。比较大亚DB120和TP-LinkWR703的结构，两者的体积和功耗是最大的不同。

关于路由器的选择我们从以下几个方面分析：

- 1、重量和体积



图 3.2 大亚 DB120 路由器



图 3.3 TP-LinkWR703 路由器

图 3.2 是大亚 DB120 路由器，图 3.3 TP-LinkWR703 路由器。很明显，TP-LinkWR703 体积可以做到非常小，而大亚 DB120，因为存在复杂的电路，PCB 面积需要很大的空间。重量上，因为体积的区别，TP-LinkWR703 路由器也优于大亚 DB120，大约有 50%-70% 的重量差别。这个重量的差异，随路由器安装的位置的不同而对整个智能车的稳定性有不同的影响。

2、可操作性

相比于大亚DB120路由器，TP-LinkWR703路由器预留有TTL串口和USB摄像头挂载口，它在重量和体积上都要优于大亚DB120，因为其引出了TTL串口线，可以方便对路由进行调试和刷机，增强了路由的可操作性。另外通过安装在路由里面的Ser2net软件，能把来自wifi信道的指令转到串口输出，并于单片机芯片进行通讯，做出相应动作，还可以通过安装不同的软件来实现更多的功能。

因此可以在路由器选型时，可操作性作为优先级较高的一个参考标准。而在其他可操作性要求不高的情况下，两种路由器的操作性差别，可以不作为主要考虑因素。

3、电能和功耗

影响路由器无线通讯的一个重要原因，就是供电和功耗。首先，大亚DB120需要9V供电，对智能车而言，这就意味着需要额外给大亚DB120设置一个升压电路。这意味着成本和体积的增加。还好，越来越多的设计者在智能车电机驱动部分开始采用MOS管驱动的方式，而MOS管栅极电压很多都需要9V，因此两路电源合二为一，提高了电源的利用率。不能否认，大亚DB120确实是耗电大户，工作时散发的热量足以烫手。对于电池供电的系统来说，每一点电能的节约都至关重要。从节能角度来说，大亚DB120真的有点不适合，随着技术的发展，目前路由器需要的电流已经从开始的150mA降到了现在的50mA，但是别忘记，大亚DB120的供电是9V，这么看来，功耗依然很大。而 TP-LinkWR703路由器用5V供电和微功耗，在功耗方面与大亚DB120比起来则有很大优势。

综合上述几个方面，可以看出可操作性，电能和功耗是我们着重考虑的几个方面。选型时，考虑电机高速运行时，功耗较大，故节省电能成为要考虑的一个重要问题。虽然 TP-LinkWR703 需要自主刷固件，但就是因为可以刷很多开源的固件，可以让路由装载很多实用的软件，丰富智能车的功能。综合考虑，我选择的型号为 TP-LinkWR703 无线路由器

TP-LinkWR703 无线路由器具有以下的特点和优势：

- (1) 电路板的尺寸 40*40mm。
- (2) 5V 供电电压。
- (3) 输出 TTL 串口线，分别是 5V、RX、TX、GND, 可以通过 TTL 刷机或通过 TTL 与更多的硬件连接，完成别的应用；
- (4) 升级原有的4MB SPI FLASH，可以升级到8MB，更大的 Flash 容量可以使 Flash 里装更多的程序；
- (5) 升级 RAM，从32MB 升级到64MB。这样在带更多客户端时，CPU 的压力会减小很多；
- (6) 主要是通过重新编译 OpenWRT 代码来实现；
- (7) 引出 SMA 天线插座。外接天线，可以在一定程度上扩大这个路由器的无线覆盖范围；
- (8) 外扩 USB HUB 芯片，通过自带的 USB 接口，再加上 USB HUB 芯片，可以扩充这

个路由器的 USB 接口数量，以便同时接更多的 USB 外设。这个功能也可以通过直接外接成品 USB HUB 来实现。USB 接口丰富之后，可以接更多的 USB 设备，以完成更多的功能，如脱机下载、网络监控等。



图 3.4 TP-LinkWR703 电路板

路由器安装在智能车最小系统与底板之间，方便与单片机进行通讯，及视频信号的实时传送。

3.1.2 路由器的供电电路

型号为TP-LinkWR703路由器需要给其提供稳定的5V电源，由于L7805的稳压的线性度非常好，所以选用L7805单独对其进行供电，L7805具有纹波小、电路结构简单的优点，选用L7805单独对其进行供电一方面可以为路由提供稳定的电压，另一方面可以减少路由与单片机相互工作间的干扰。如图3.5所示。

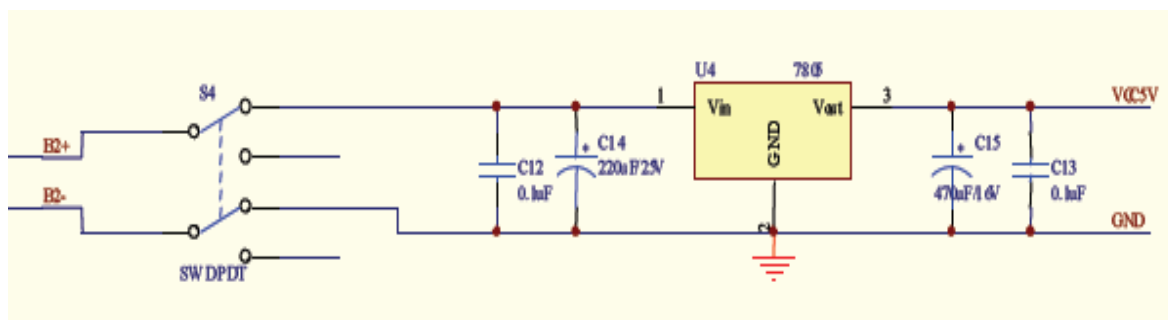


图 3.5 路由器的供电电路

3.1.3 路由器刷机固件介绍

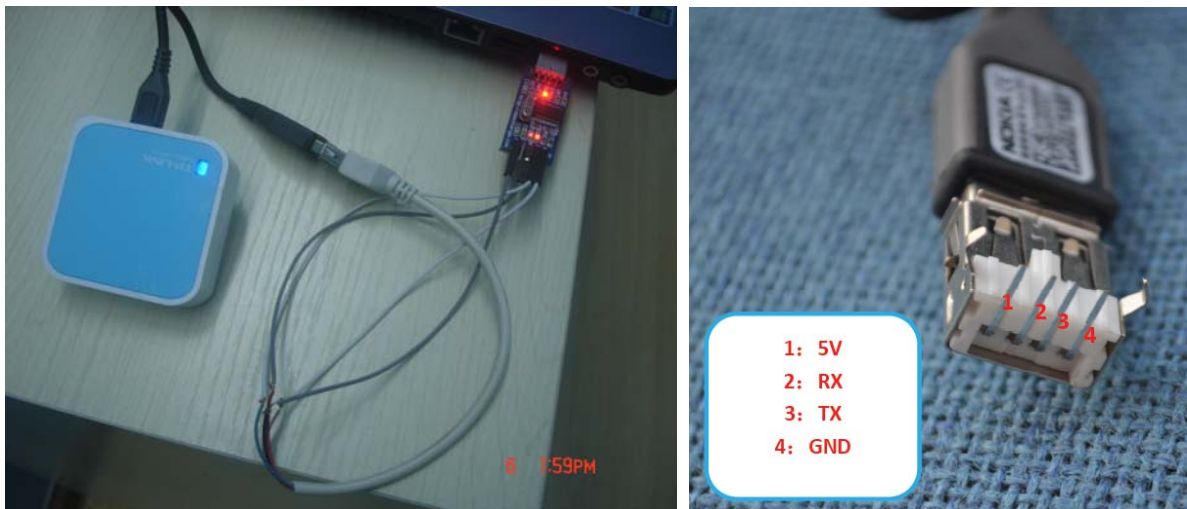


图 3.6 路由器刷机硬件连接图

固件 WT-wr703n-v1-squashfs-sysupgrade 固件说明：

1、Ser2net 已设置开机启动

2、mjpg_streamer 开机启动。摄像头只有 2 种驱动，uvc 和 301，目前开机 mjpg_streamer 缺省启动了 301 摄像头，301 摄像头无需设置直接跳到第 3 步，uvc 需要修改下即可 ssh 登陆后，vi etc/init.d/wificar，找到摄 301 像头参数 mjpg_streamer -b -i "input_uvc.so -r 640x480 -f 30" -o "output_http.so -p 8080 -w /web" 修改为 uvc 参数 mjpg_streamer -b -i "input_uvc.so -r 352x288 -f 15 -y" -o "output_http.so -p 8080 -w /web" 然后在 putty 命令行下输入 chmod 777 /etc/init.d/wifirobot 更改属性输入/etc/init.d/wificar enable 启动该配置文件重启路由器，301 摄像头设置完成。

3、支持 Luci/web 管理, 错误设置后不能进入的，可以长按复位键回初始模式

4、网络设置，目前是 AP 模式，IP:192.168.1.1。

3.1.4 刷机步骤

1、网线连接电脑和路由器的 LAN1（靠近电源那个网口，升级只能使用此口，有的版本丝印不是 LAN1，反正记住靠近电源的那个口即可），电脑 IP 设置为 192.168.1.169/255.255.255.0，192.168.1.1；设置完成点击网络邻居，选择属性，选择升级用的有线网口，右键点修复。

2、用 USB TTL 线接好路由的 GND、RXD、TXD 三条线（VCC 电源千万别接，会烧路由），打开 putty，选好端口号，端口号请查看电脑设备管理器，设置波特率为 115200，

点击下面的 Open。

3、路由器通电，接线且设置正确的话，putty 窗口应该出现路由器的信息，在通电 3 秒内，按下键盘的回车键，阻止路由器 CFE 启动，这时候应出现 CFE>。

4、输入小写 e a，e a 之间有个回车键，然后再按下 y，点击回车，路由器开始擦除清除原系统固件。清除之后，路由会持续输出很多命令行，且不会停止，我们不用管，关掉路由电源。

5、重新打开路由电源，和第四步一样，通电 3 秒内按下回车键阻止 CFE 继续启动打开浏览器，输入 192.168.1.1，应看到上传固件的的选择文件，putty 窗口也会出现路由器升级进度信息，此时千万不要断电，理论上此时断电会刷坏路由器，升级完成后，耐心等待 5 分钟，路由器会自动重启。

6、等待路由启动后，等待几分钟再次重启一次路由，打开网页 192.168.1.1，输入密码登陆进入路由器的设置，将路由器的无线网络开启，只有这样电脑等终端设备才能通过 wifi 信号和路由器连接。

7、用 putty 登陆路由，转到/etc/init.d/目录，可以看到一个 ser2net.conf 文件，打开它可以看到路由串口的配置信息，默认配置的波特率是 9600,8 位数据位，1 位停止位，无校验位。完成之后需要设定 mjpg-streamer 和 ser2net 随机启动，不然没法正常工作。

8、由于我使用的摄像头是 301 芯片的，在电脑上显示会花屏，所以还需要对脚本进行修改，其操作步骤如下：

具体步骤：

a、登陆路由器：telnet 192.168.1.1 (IP 地址根据你的路由器实际地址而定)。

b、编辑脚本/etc/init.d/wificar：# vi /etc/init.d/wificar （在 START=80 后面增加一行 killall mjpg_streamer，修改后的脚本看下面内容。）

```
#!/bin/sh /etc/rc.common
```

```
START=80
```

```
killall mjpg_streamer
```

```
start() {
```

```
mjpg_streamer -b -i "input_uvc.so -r 640x480 -f 15" -o "output_http.so -p 8080 -w /web"
```

```
# mjpg_streamer -b -i "input_uvc.so -r 352x288 -f 15 -y" -o "output_http.so -p 8080 -w /web"
```

```

ser2net
}
stop(){
killall mjpg_streamer
killall ser2net
}

```

c、保险起见，修改 wificar 脚本权限。

```
# chmod -777
```

d、重启路由器。

```
# sync; sync; reboot
```

9、插上 301 芯片的摄像头，在电脑上打开我们的一个 wifi 小车的控制界面软件，选择 wifi 控制模式，然后打开视频，这时我们可以看到一个稳定视频信号通过路由器无线传送到我们的电脑终端，至此说明我们路由器刷机及其配置完成。

3.1.5 路由测试

路由器通电之前，先把摄像头连接到无线路由模块。从 USB 线中接好 TTL 电平到电脑或者设备。注意是否采用交叉方式。路由开机等待 45 秒之后用开发的上位机软件进行测试，如图 3.7 所示。

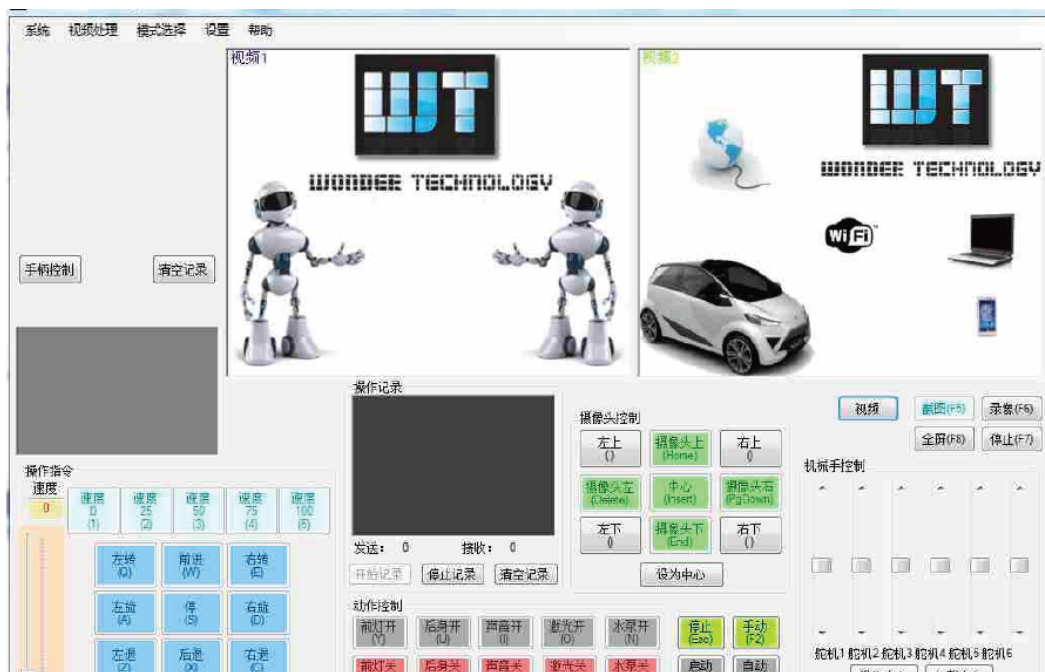


图 3.7 路由器上位机测试界面

点击视频，即可以看到无线路由模块连接摄像头视频信号。

点击软件中的启动后，即可以向无线路由模块的串口发送数据。另外还可以采用 TCP232 对此模块进行测试，如图 3.8 所示。

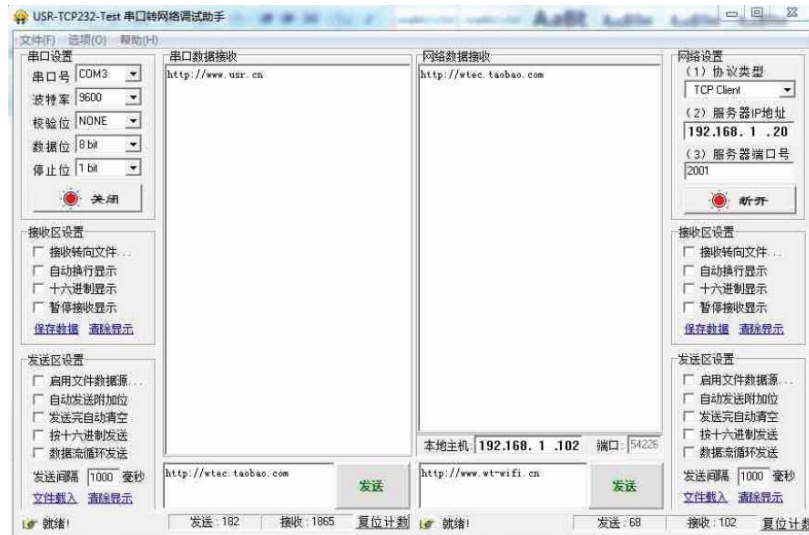


图 3.8 串口转网络调试助手界面

串口端采用 USB 赚串口（TTL 电平），通过交叉的方式与电脑端连接。PC 端通过网线或无线 WIFI 连接到路由模块，采用 TCP Client 模式，输入 IP 地址：192.168.1.20，端口：2001，即可从串口与网络相互发信息。

3.2 摄像头的选型和安装方案设计

本项目中摄像头实现实时监测小车周边情况，使之成为一个无线可控移动的视频监控器，也用于判断小车当前路况。

3.2.1 摄像头简介

摄像头(CAMERA)又称为电脑相机、电脑眼等，它作为一种视频输入设备，在过去被广泛的运用于视频会议、远程医疗及实时监控等方面。近年以来，随着互联网技术的发展，网络速度的不断提高，再加上感光成像器件技术的成熟并大量用于摄像头的制造上，这使得它的价格降到普通人可以承受的区间。普通的人也可以彼此通过摄像头在网络进行有影像、有声音的交谈和沟通，另外，人们还可以将其用于当前各种流行的数码影像、影音处理。

3.2.2 摄像头的分类

摄像头分为数字摄像头和模拟摄像头两大类。模拟摄像头可以将视频采集设备产生的模拟视频信号转换成数字信号，进而将其储存在计算机里。模拟摄像头捕捉到的视频

信号必须经过特定的视频捕捉卡将模拟信号转换成数字模式，并加以压缩后才可以转换到计算机上运用。数字摄像头可以直接捕捉影像，然后通过串、并口或者 USB 接口传到计算机里。现在电脑市场上的摄像头基本以数字摄像头为主，而数字摄像头中又以使用新型数据传输接口的 USB 数字摄像头为主，目前市场上可见的大部分都是这种产品。除此之外还有一种与视频采集卡配合使用的产品，但目前还不是主流。由于个人电脑的迅速普及，模拟摄像头的整体成本较高等原因，USB 接口的传输速度远远高于串口、并口的速度，因此现在市场热点主要是 USB 接口的数字摄像头。以下主要是指 USB 接口的数字摄像头。

3.2.3 摄像头的工作原理

摄像头的工作原理大致为：景物通过镜头(LENS)生成的光学图像投射到图像传感器表面上，然后转为电信号，经过 A/D(模数转换)转换后变为数字图像信号，再送到数字信号处理芯片(DSP)中加工处理，再通过 USB 接口传输到电脑中处理，通过显示器就可以看到图像了。我们这里是将摄像头连接在路由器的 USB 接口上，通过路由器上的摄像头驱动软件使摄像头工作，并由路由器通过 WIFI 将视频信号发送出去。电脑等终端接收设备，接收到视频信号后通过控制软件的界面显示图像。

3.2.4 摄像头的安装设计

从摄像头的工作原理就可以列出摄像头的主要结构和组件：1、主控芯片（详情请参阅下面介绍）2、感光芯片（详情请参阅下面介绍）3、镜头（详情请参阅下面介绍）4、电源。摄像头内部需要两种工作电压：3.3V 和 2.5V，因此好的摄像头内部电源也是保证摄像头稳定工作的一个因素。

在本次 wifi 小车的制作中我们使用的天敏 S606 的摄像头，如图 3.9 所示。

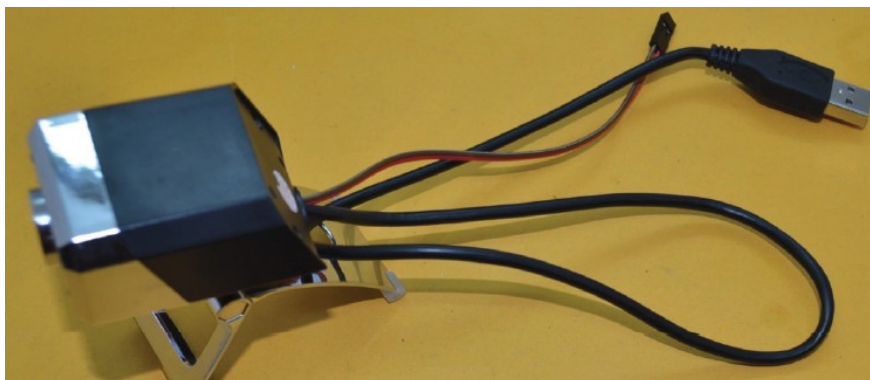


图 3.9 天敏 301 摄像头

对于摄像头的安装位置主要有安装高度和俯视角度两个关键参数。增加安装高度同时增大俯视角度可以在提高摄像头的前瞻性，使视野更开阔，便于单片机识别，保证识别稳定且有足够的前瞻。所以把摄像头的位置安装的很高是提高智能车前瞻性的一种很有效的方式，但是，在保证前瞻性的前提下还要低重心、高稳定性，综合两个方面的特点最终设计如图 3.10 所示



图 3.10 云台系统设计图

综合考虑各方面的因素，最终安装高度确定为 5cm，俯视角度约为 90 度。为了开阔摄像头的视野性能，我专门设计了两个自由度能 180 度旋转的云台系统，而且采用了黑色碳纤材料，摄像头改装之后安装在云台系统上，使安装的零件质量控制在 60g 以下。最终效果如图 3.11。

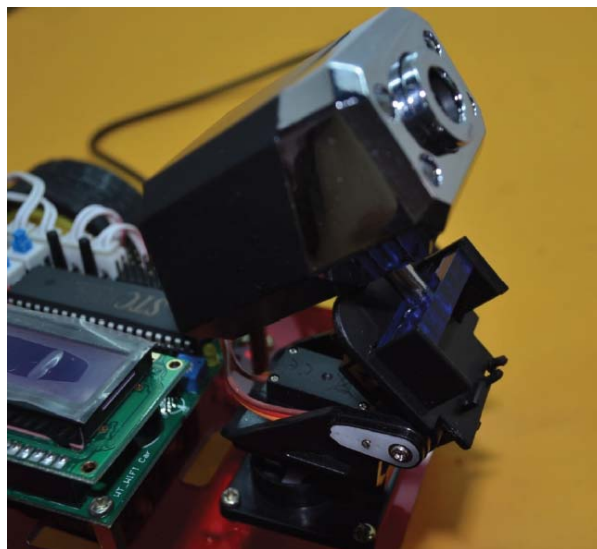


图 3.11 摄像头实际安装效果图

3.3 智能车车体设计

车体采用硬度较大、容易操作的亚克力板材拼合制成，车体更牢固。采用双舵机和云台来固定摄像头，摄像头转动灵活。云台选用的是质地轻、刚性大，直径小而非常坚固耐用的黑色碳纤材料。此云台灵活性能、开阔性非常好。

3.3.1 车体布局

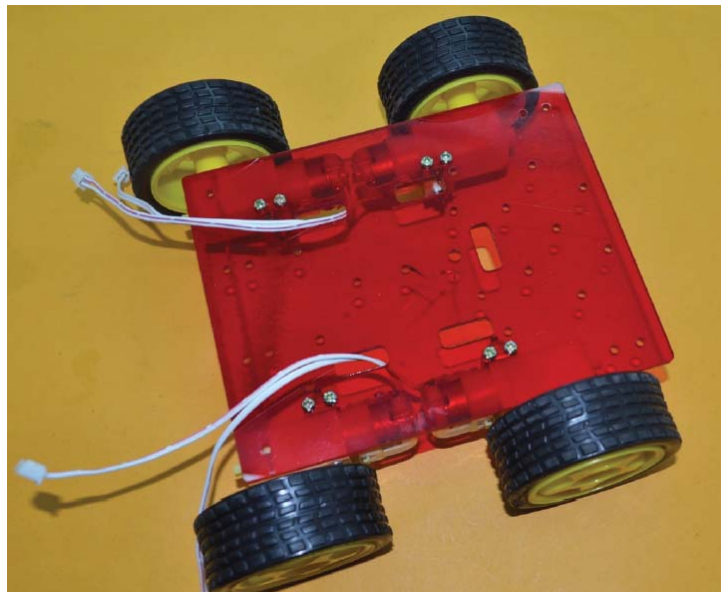


图 3.12 智能车车体

为了提高智能车行走情况下的操纵稳定性，在整车布局上主要在做了三方面工作：低位主板安装、降低摄像头安装高度和紧凑安装的轻量化设计、控制整车质心尽量靠近中心转向点。经过上述设计的智能车质心基本在中心转向点附近，所以车体在极恶劣的情况下也能完成视频信号的实时监控。

3.3.2 摄像头云台的设计安装

为了降低整车重心，需严格控制摄像头及其云台系统的重量。首先是设计了轻巧的云台系统，并采用了黑色碳纤材料作为安装摄像头的云台支架，这样可以获得最大的刚度质量比。在摄像头安装参数的可调整性设计上，采用了双舵机结构。这样不仅使摄像头能上下滑动调整高度，而且能够使摄像头左右横摆调整水平宽度。而且这个定位关系是从底座安装孔获得并以较高定位精度向上传递的，从而获得良好的重复定位精度。使得能够方便的将云台系统拆下和装上，不必顾虑摄像头的位置会发生改变。使摄像头及云台系统便于拆卸和维修，具有快速保障能力。舵机如图3-13所示。



图 3.13 舵机

舵机的输入线共有三条，红色中间，是电源线，一边黑色的是地线，这两根线给舵机提供最基本的能源保证，主要是电机的转动消耗。电源有两种规格，一是 4.8V，一是 6.0V，分别对应不同的转矩标准，即输出力矩不同，6.0V 对应的要大一些，具体看应用条件；另外一根线是控制信号线，Futaba 的一般为白色，JR 的一般为桔黄色。另外要注意一点，SANWA 的某些型号的舵机引线电源线在边上而不是中间，需要辨认。但记住红色为电源，黑色为地线，一般不会有错。

舵机的控制：

舵机的控制信号为周期是 20ms 的脉宽调制（PWM）信号，其中脉冲宽度从 0.5ms-2.5ms，相对应舵盘的位置为 0—180 度，呈线性变化。也就是说，给它提供一定的脉宽，它的输出轴就会保持在一个相对应的角度上，无论外界转矩怎样改变，直到给它提供一个另外宽度的脉冲信号，它才会改变输出角度到新的对应的位置上。舵机内部有一个基准电路，产生周期 20ms，宽度 1.5ms 的基准信号，有一个比较器，将外加的 PWM 信号与基准信号相比较，判断出方向和大小，从而产生电机的转动信号。由此可见，舵机是一种位置伺服的驱动器，转动范围不能超过 180 度，适用于那些需要角度不断变化并可以保持的驱动当中。比方说机器人的关节、飞机的舵面等。

舵机的作用：

在本次 wifi 小车中我们利用两个舵机做成一个云台（如图 3.14），将摄像头固定在云台上，通过 89C52 单片机控制舵机旋转一定的角度，把我们想要角度的图像通过摄像头传送到电脑等终端设备。

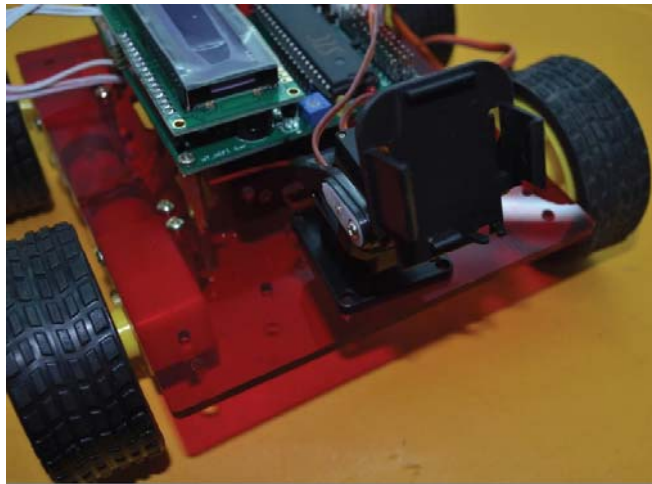


图 3.14 舵机云台安装

3.3.3 主板安装

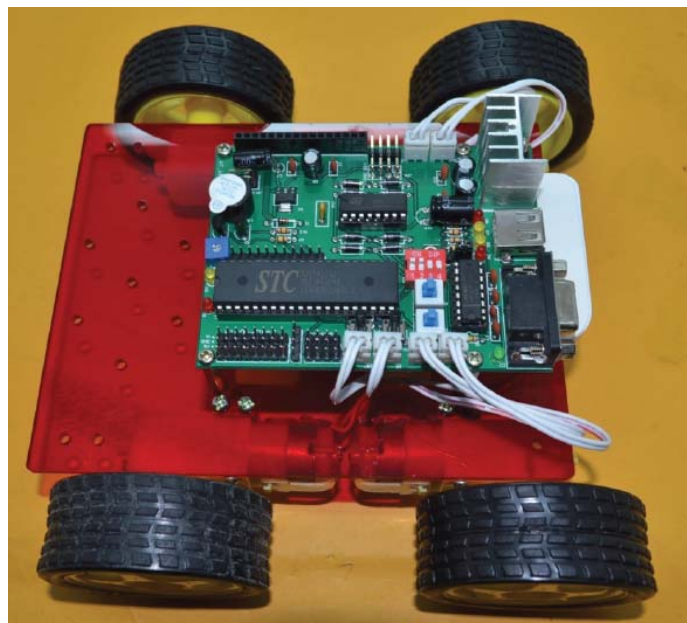


图3.15 主板安装示意图

如图 3.15 所示，主板安装紧贴车体底盘，放在底板的上方，用铜柱固定。这样做的目的就是控制车体的重心，提高稳定性。电路板的形状是 100×70 的长方形万能板，安装时非常方便。各电路板之间通过杜邦线连接，用扎带整齐地布在车体上。

3.4 电路设计

该智能车系统包括以下几个模块：最小系统板模块、电源模块、红外壁障模块、串口模块、视频采集模块、无线通讯模块、电机驱动模块、照明电路模块。智能车的电路模块主要集中在一张单片机最小系统板上，无线通讯模块安装在最小系统的下方，视频采集模块和照明电路模块安装在智能车的车头部分。红外壁障模块安装在智能车地盘正

前方。各电路板之间通过杜邦线连接，安装拆卸方便，能够快速组装。

灯光、蜂鸣器及电平转换电路都是通过三极管的开关作用实现的，如图 3.16、3.17、3.18 所示。

灯光电路和蜂鸣器电路都是通过单片机引脚输出高低电平到三极管的基极，使三极管导通或截止，从而控制 LED 的点亮熄灭，控制蜂鸣器的鸣叫。其中灯光电路中的电阻用于限流保护发光二极管。

电平转换电路中将 5V 转化为 3V 是通过滑动电阻器降压实现的，将 3V 转换为 5V 是通过 3V 的信号控制三极管的截止，导通进行控制从而使三极管的发射极对应输出高低电平用于输入单片机进行处理。

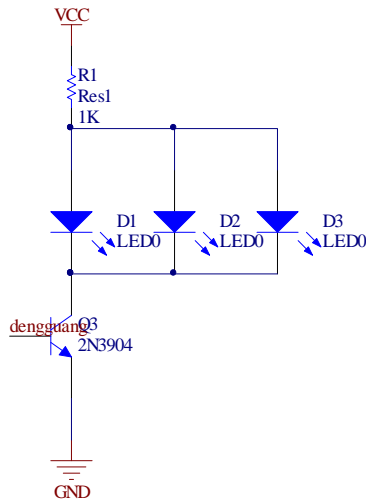


图 3.16 灯光控制电路

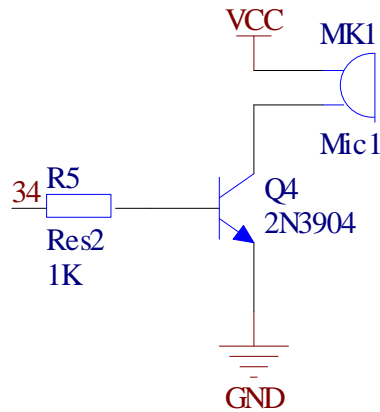


图 3.17 蜂鸣器控制电

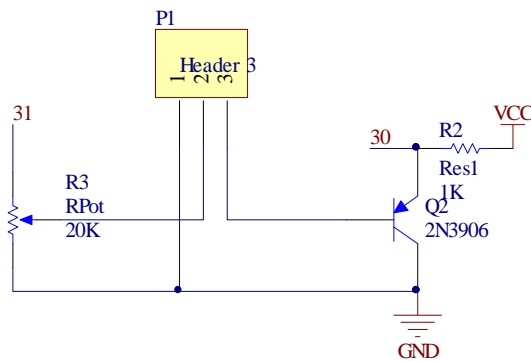


图 3.18 电平转换电路

本方案将MCU 最小系统运行在单片模式下，它由以下模块组成：供电电路、时钟电路、复位电路、下载口，如图3.19所示。

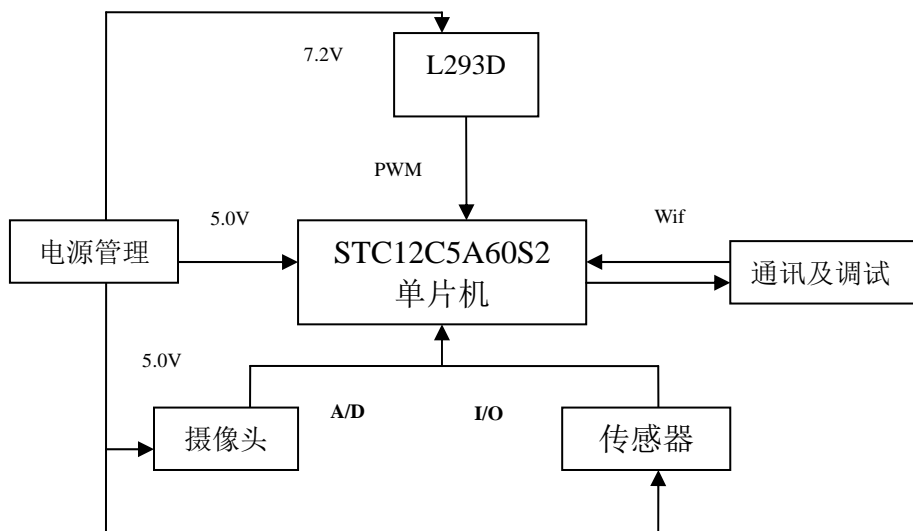


图 3.19 系统电路的硬件结构

3.4.1 电源模块

该小车由 7.2V 电池供电，但是该单片机需要的是+5V 电源。单片机 I/O 模块的供电是+5V 电压，而单片机片内使用 2.5V 电压，片外 I/O 使用 5V 电压，较低的片内电压使 CPU 运算速度快、功耗低；较高的 I/O 电平有利于抗外界干扰，所以系统使用+5V 电压供电，能够工作在复杂的环境之中。该系列的单片机内部集成了电压调整器模块，该模块产生单片机内部需要的其他电压，为了稳定+5V 电压，克服电机运行所产生的电源电压变化，该供电电路加上一些储能电容和去耦电容。此外给电源系统接一个 LED 指示灯，是为了以后能够正确地观察电路工作情况。



图 3.20 电源与充电器

电源模块为系统其它各个模块提供所需要的电源，最小系统板模块、电源模块、红外壁障模块、串口模块、视频采集模块、无线通讯模块、电机驱动模块、照明电路模块。供电电压有 7.2V、5V 两种电压值。设计中，除了要考虑到电压范围和电流容量等基本参数之外，还要在电源转换效率、降低噪声、防止干扰和电路简易程度等方面进行优化。在电源设计上，设计了两个电源输入接口，使系统板控制小车的电源，与供电给路由的电源独立分开。一方面可以为路由提供稳定的电压，另一方面可以减少路由与单片机相互工作间的干扰。此两个电源同时设计了两个独立的开关，可以方便调试WIFI 小车。其实物图和原理图如图3.21,3.22所示：

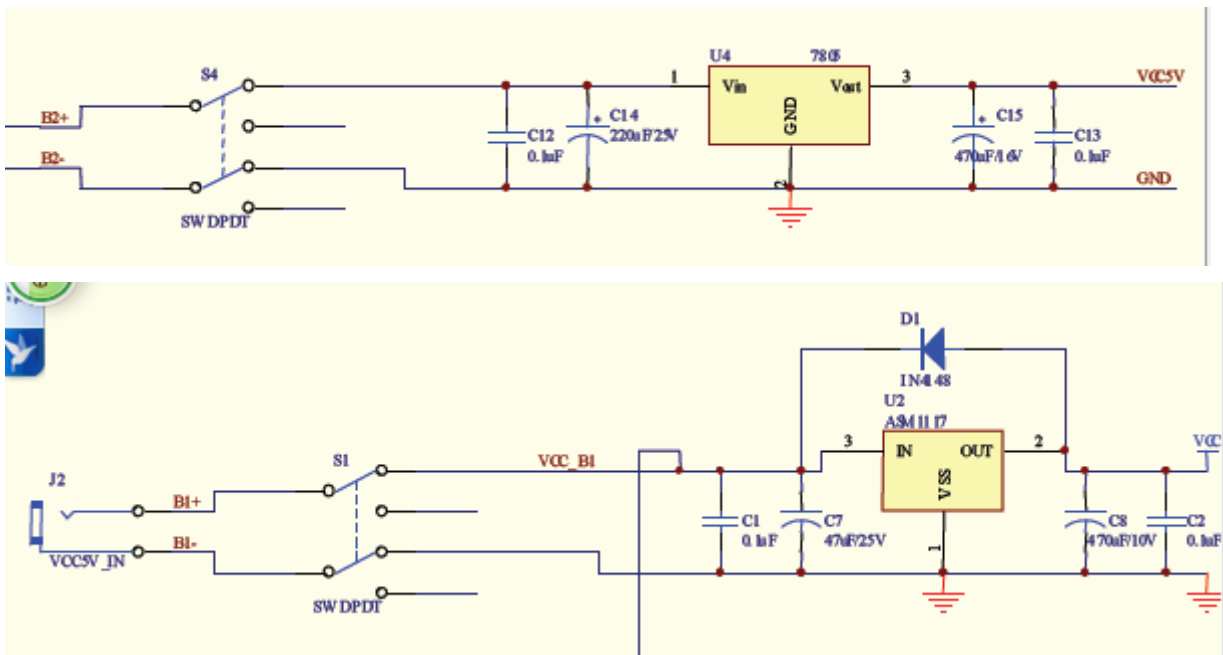


图3.21 稳压电源原理图

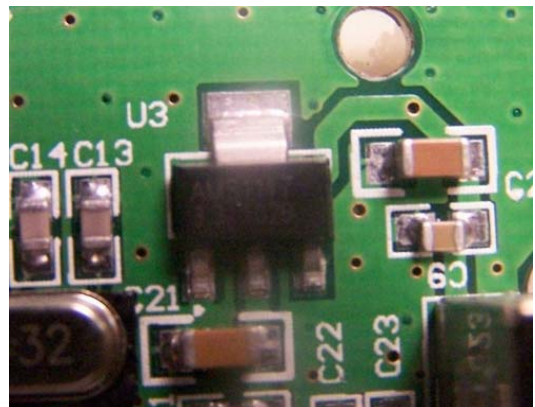
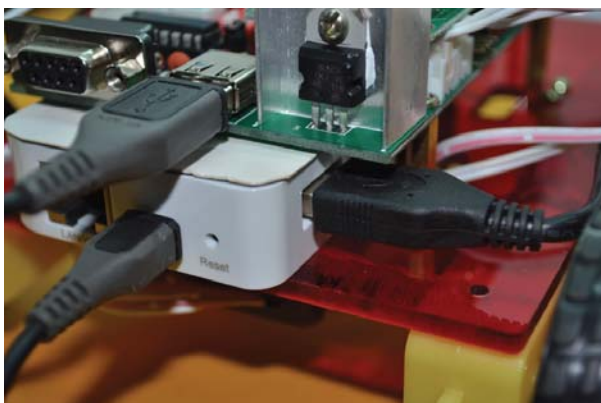


图 3.22 稳压电源实物图

3.4.2 最小系统模块

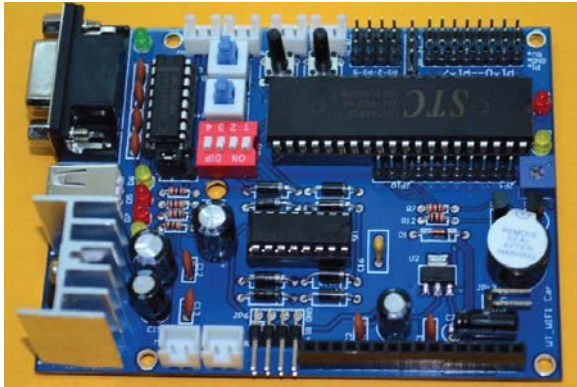


图 3.23 最小系统板

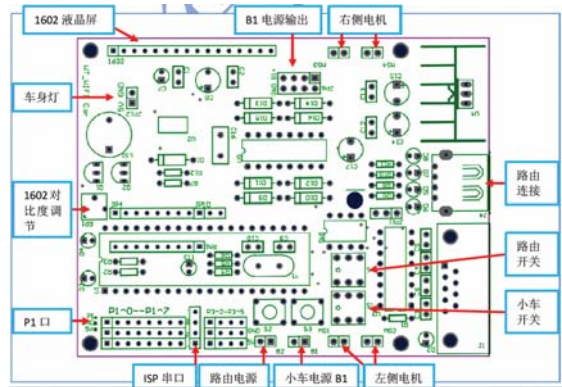


图 3.24 印刷电路板

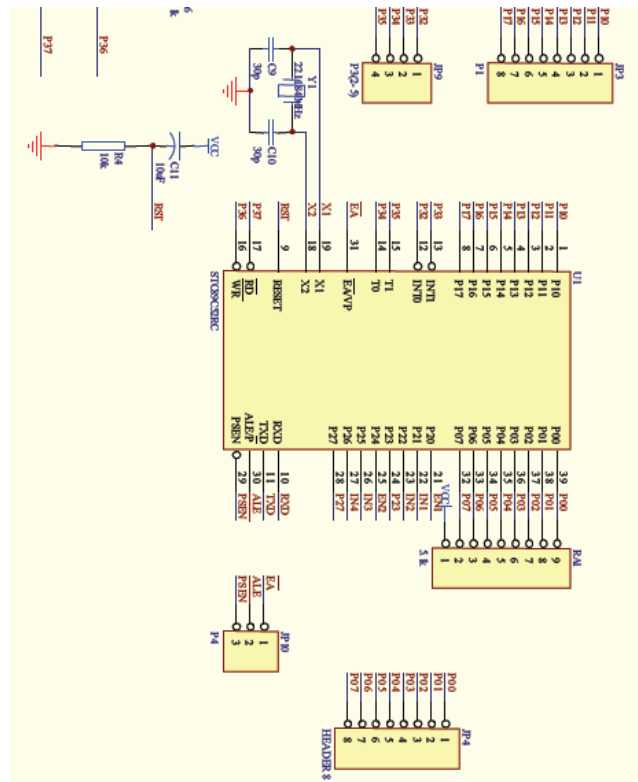


图 3.25 最小系统板引脚分布图

STC12C5A60S2 是宏晶公司MCS-51 系列单片机中基本的产品，它采用STC公司可靠的CHMOS工艺技术制造的高性能 8 位单片机，属于标准的MCS-51 的HCMOS产品。它结合了HMOS的高速和高密度技术及CHMOS的低功耗特征，它基于标准的MCS-51 单片机体系结构和指令系统，属于 80C51 增强型单片机版本，集成了时钟输出和向上或向下计数器等更多的功能，适合于类似马达控制等应用场

合。STC12C5A60S2 内置 8 位中央处理单元、256 字节内部数据存储器RAM、8k片内程序存储器（ROM）32 个双向输入/输出(I/O)口、3 个 16 位定时/计数器和 5 个两级中断结构，一个全双工串行通信口，片内时钟振荡电路。此外，STC12C5A60S2 还可工作于低功耗模式，可通过两种软件选择空闲和掉电模式。在空闲模式下冻结CPU而RAM定时器、串行口和中断系统维持其功能。掉电模式下，保存RAM数据，时钟振荡停止，同时停止芯片内其它功能。STC12C5A60S2 有PDIP(40pin)和PLCC(44pin)两种封装形式。

主要功能特性：

1、标准 MCS-51 内核和指令系统。2、32 个双向 I/O 口。3、3 个 16 位可编程定时/计数器。4、向上或向下定时计数器。5、6 个中断源。6、全双工串行通信口。7、一帧错误侦测。8、一自动地址识别。9、空闲和掉电节省模式。10、片内 8kROM（可扩充 64kB 外部存储器）。11、256x8bit 内部 RAM（可扩充 64kB 外部存储器）。12、时钟频率 3.5-12/24/33MHz。13、改进型快速编程脉冲算法。14、5.0V 工作电压。15、布尔处理器。16、4 层优先级中断结构。17、兼容 TTL 和 CMOS 逻辑电平。18、PDIP(40)和 PLCC(44)封装形式。

管脚说明：

VCC：供电电压。 GND：接地。

P0 口：P0 口为一个 8 位漏级开路双向 I/O 口，每脚可吸收 8TTL 门电流。当 P1 口的管脚第一次写 1 时，被定义为高阻输入。P0 能够用于外部程序数据存储器，它可以被定义为数据/地址的第八位。在FIASH编程时，P0 口作为原码输入口，当FIASH进行校验时，P0 输出原码，此时P0 外部必须被拉高。

P1 口：P1 口是一个内部提供上拉电阻的 8 位双向 I/O 口，P1 口缓冲器能接收输出 4TTL 门电流。P1 口管脚写入 1 后，被内部上拉为高，可用作输入，P1 口被外部下拉为低电平时，将输出电流，这是由于内部上拉的缘故。在 FLASH 编程和校验时，P1 口作为第八位地址接收。

P2 口：P2 口为一个内部上拉电阻的 8 位双向 I/O 口，P2 口缓冲器可接收，输出 4 个 TTL 门电流，当 P2 口被写“1”时，其管脚被内部上拉电阻拉高，且作为输入。并因此作为输入时，P2 口的管脚被外部拉低，将输出电流。这是由于内部上拉的缘故。P2 口当用于外部程序存储器或 16 位地址外部数据存储器进行存取时，P2 口输出地址的高八位。在给出地址“1”时，它利用内部上拉优势，当对外部八位

地址数据存储器进行读写时，P2 口输出其特殊功能寄存器的内容。P2 口在 FLASH 编程和校验时接收高八位地址信号和控制信号。

P3 口：P3 口管脚是 8 个带内部上拉电阻的双向 I/O 口，可接收输出 4 个 TTL 门电流。当 P3 口写入“1”后，它们被内部上拉为高电平，并用作输入。作为输入，由于外部下拉为低电平，P3 口将输出电流（ILL）这是由于上拉的缘故。P3 口作为 STC89C52 的一些特殊功能口，管脚备选功能：

P3.0 RXD（串行输入口）

P3.1 TXD（串行输出口）

P3.2 /INT0（外部中断 0）

P3.3 /INT1（外部中断 1）

P3.4 T0（记时器 0 外部输入）

P3.5 T1（记时器 1 外部输入）

P3.6 /WR（外部数据存储器写选通）

P3.7 /RD（外部数据存储器读选通）

RST：复位输入。当振荡器复位器件时，要保持 RST 脚两个机器周期的高电平时间。

ALE/PROG：当访问外部存储器时，地址锁存允许的输电平用于锁存地址的地位字节。

在 FLASH 编程期间，此引脚用于输入编程脉冲。在平时，ALE 端以不变的频率周期输出正脉冲信号，此频率为振荡器频率的 1/6。因此它可用作对外部输出的脉冲或用于定时目的。然而要注意的是：每当用作外部数据存储器时，将跳过一个 ALE 脉冲。如想禁止 ALE 的输出可在 SFR8EH 地址上置 0。此时，ALE 只有在执行 MOVX，MOVC 指令是 ALE 才起用。另外，该引脚被略微拉高。如果微处理器在外部执行状态 ALE 禁止，置位无效。

/PSEN：外部程序存储器的选通信号。在由外部程序存储器取指期间，每个机器周期两/PSEN 有效。但在访问外部数据存储器时，这两次有效的/PSEN 信号将不出现。

/EA / VPP：当/EA 保持低电平时，则在此期间外部程序存储器（0000H-FFFFH），不管是否有内部程序存储器。注意加密方式 1 时，/EA 将内部

锁定为 RESET；当/EA 端保持高电平时，此间内部程序存储器。在 FLASH 编程期间，此引脚也用于施加 12V 编程电源（VPP）。

XTAL1：反向振荡放大器的输入及内部时钟工作电路的输入。

XTAL2：来自反向振荡器的输出。

振荡电路：

外接石英晶体或者陶瓷谐振器以及电容 C1、C2 接在放大器的反馈回路中构成并联振荡电路。为了使装置能够被外部时钟信号激活，XATL1 应该有效，而 XTAL2 应该被悬空。由于输入到内部的时钟信号电路通过了一个二分频的信号，外部信号的工作周期比没有别的要求，但是最大值和最小值的大小可以在数据表上观察出来。

当正常工作时，外部振荡器可以计算出 XTAL1 上的电容，最大可达到 100pF。这是由于振荡器电容和反馈电容之间的相互作用。当外部信号是标准高电平或者低电平时，电容不会超过 20pF。

空置模式：

用户的软件都可以调用空置模式。当单片机出于这种模式，耗能就会自然降低。特殊功能端和板子上的随机存储器在空置状态保持各自的电平。但是处理器阻止装置执行指令。空置模式会被激活如果端口处于复位状态或者中断系统有效。

3.4.3 时钟电路

单片机的时钟信号为单片机芯片内部的各种操作提供时间基准，时钟电路为单片机产生时钟脉冲序列。作为单片机工作的时间基准，典型的晶体振荡频率为 12MHz。

MCS-51 系列单片的时钟信号可以由两种方式产生：一种为内部时钟方式，利用芯片内部的振荡电路；另一种为外部时钟方式。其两种电路如图 3.26 所示。

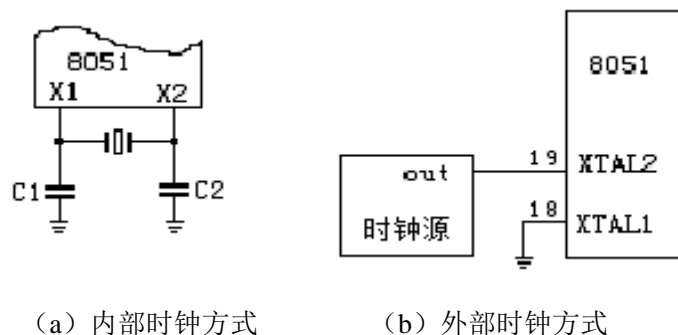
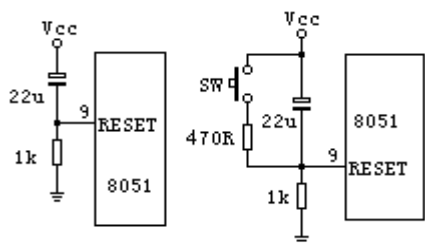


图 3.26 时钟电路

3.4.4 复位电路

单片机复位是使 CPU 和系统中的其他功能部件都恢复到一个确定的初始状态，并从这个状态开始工作。当在 MCS-51 系列单片的 RST 引脚处引入高电平并保持 2 个机器周期，单片机内部就执行复位操作。单片机常见的复位电路有两种基本形式：一种是上电自动复位，另一种是手动复位。其两种电路方式如图 3.27 (a)、(b) 所示。

由于 STC12C5A60S2 单片机芯片内有时钟振荡电路，所以此系统单片机均采用内部时钟方式。只需在单片机的 XTAL1 和 XTAL2 引脚外接石英晶体和微调电容，就构成了自激振荡器并在单片机内部产生时钟信号脉冲信号。同时，也采用手动复位电路。其具体电路设计如图 3.28 所示。



(a) 上电自动复位 (b) 手动复位

图 3.27 单片机复位电路示意图

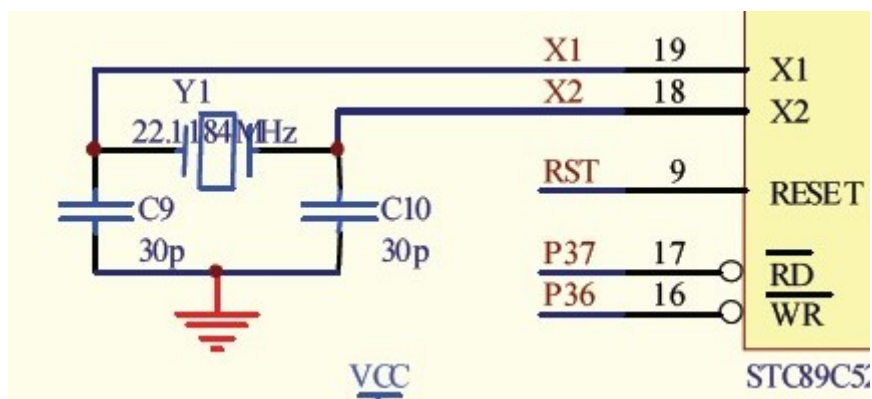


图 3.28 智能车单片机具体复位电路

在此图中，C1、C2 电容的作用的是稳定频率和快速起振，其值为 5~30pF，在此选择 30pF；晶振 X1 的振荡频率范围在 3.5~33MHz 之间选择，因为需要使用串通讯涉及波特率，所以在此选择 22.1184MHz。

3.4.5 无线通讯模块

基于 IEEE802.11 系列协议的 WLAN 是一种能支持较高数据传输速率 (1~54Mbit/s)，采用微蜂窝、微微蜂窝结构，自主管理的计算机局域网络。其关键技术大致有 3 种，直

序列扩频调制技术 (DSSS:Direct Sequence Spread Spectrum) 及补码键控 (CCK: Complementary Code Keying) 技术、包二进制卷积 (PBCC: Packet Binary Convolutional Code) 和正交频分复用技术 (OFDM:Orthogonal Frequency Division Mustiplexing)。

WiFi 技术是一种无线联网的技术，用无线路由器通过无线电磁波来进行网络互联代替有线的链接方式，在这个无线路由器的电波覆盖的有效范围都可以采用 wifi 连接方式进行联网，如果无线路由器连接了一条 ADSL 线路或者别的上网线路，则又被称为“热点”。目前常用的无线路由器多为 54Mbps 的传输速率，还有部分 108Mbps 或以上传输速率的设备。对于 802.22b 来说，在网络建设完备的情况下，其工作距离可达到 100m 以上。

WiFi 技术作为高速有线接入技术的补充，具有为可移动性、价格低廉的优点，WIFI 技术广泛应用于有线接入需无线延伸的领域。IEEE 802.11n 协议采用 MIMO 与 OFDM 相结合，并在无线局域网中引入智能天线和软件无线电等相关技术，使数据速率成倍提高，这些都不断提升着 WiFi 的性能。

一般来说，WiFi 产品的射频部分可由五大部分组成：无线收发器、功率放大器、低噪声放大器、收发切换模块和天线。

射频设计框图如图 3.29 所示

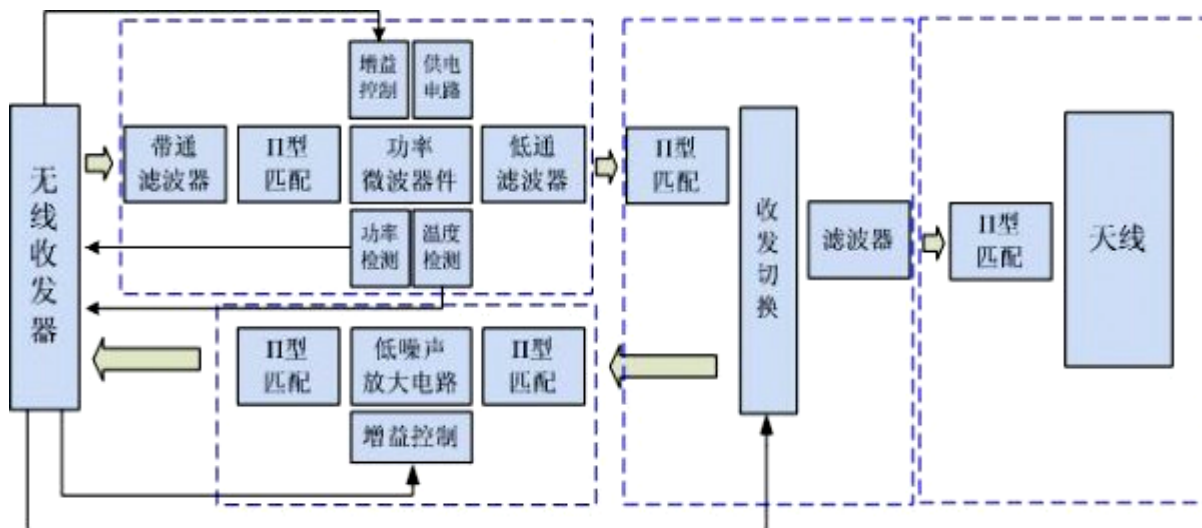


图 2.29 射频设计框图

无线收发器是核心器件之一，其重要性决定了它的外围电路很复杂，一般包含有几个重要的管脚：电源管脚，数字地，模拟地，射频输出，功率放大器增益控制，功率检测，温度检测，射频输入，低噪声放大器增益控制，发射、接收切换等管脚。无线收发

器芯片的技术参数中会首先提到其支持的协议、频率、通道数、传输速率，这些参数给出了无线收发芯片的基础可用范围，与设计方案密切相关。无线收发器的调制方式与传输速率是密切相关的，不同的传输速率对应着不同的调制方式。其他参数还包括：时钟频率，输出功率，接收灵敏度，射频接口，供电电压与功耗。

功率放大器，Power Amplifier，俗称PA，主要的作用就是将无线收发器（Radio Transceiver）送来的射频信号进行功率放大，保证有足够大的输出功率满足设计需求。功率放大器的设计会考虑很多参数，但主要分为三类：增益，噪声，非线性。增益，和最终的输出功率有关，噪声和非线性关系着信号质量。

低噪声放大器位于收发切换器（Transmit/Receive Switch）和无线收发器（Radio Transceiver）之间，对天线感应到的信号进行放大，使无线收发器进行有效的处理。低噪声放大器的性能直接影响着整个设计的灵敏度。低噪声放大器有四个部分组成：输入回路，输出回路，放大电路，增益控制。

收发切换电路实现的功能就是进行发射与接收的切换，使用切换芯片完成，切换芯片实际上是一个单刀双掷的开关，开关掷向哪一边决定于加在控制管脚上的电压。天线回路在设置时一定要注意匹配网络的设计。

该系统采用基于 Broadcom BCM5354 芯片的无线路由器作为无线通信模块核心，选用华硕 WL-520GU 无线路由，USB2.0 连接带 USB 接口的摄像头用以采集和传输视频信息。

路由是指信息从前端穿过网络传递到目的地的行为，路由器是工作在OSI参考模型的第三层——网络层的数据包设备，通过转发数据包来实现网络互联。路由器内部由CPU、内存RAM、接口和总线四部分组成，用来完成路由功能。此外还具有各类插槽，用以扩充Flash、RAM、接口和总线。无线路由器（Wireless Router）将单纯性无线AP和宽带路由器合二为一，其具备单纯性无线AP所有功能如支持DHCP客户端、支持VPN、防火墙、支持WEP加密等，还包括网络地址转换（NAT）功能，可支持局域网用户的网络连接共享。可实现家庭无线网络中的Internet连接共享，实现ADSL和小区宽带的无线共享接入。

BCM5354 具有高集成度的特点，集成了在单芯片上设计路由器所需的所有组件采用了 Broadrange 技术优化的 802.11g 路由器芯片，它集成了一个高性能的 240MHz MIPS32 处理器（具有 16KB 指令缓存，16KB 数据缓存，以及 1KB 预取缓存）、兼容 IEEE 802.11 b/g MAC/PHY、2.4-GHz 直接转换射频、USB 2.0 主控制器、SDRAM 控制器，以

及一个可配置的 5 端口快速以太网（FE）交换机，支持 125Mbps 的 HSM（High-speed Performance Mode）。集成了在上百万设备中发运的高效 CMOS 功率放大器，提供与具有外部功率放大器一样高的输出功率。可支持 Linux，VxWorks 等多方操作系统。

BCM5354 通过其可配置的介质接口支持 WAN 连接，每端口可编程的 4 优先级排列为有保证的宽带应用、DiffServ/TOS 和 L2/L3 IGMP 监听保证了服务质量(IEEE 802.1p)。IEEE 802.1Q VLAN 灵活地实现了 VLAN 分组和 WAN 端口隔离。支持 BroadRange 技术，集成了先进的数字信号处理技术。BCM5354 芯片解决方案改善了器件的接收发送数据的能力，在低信号强度下依然能够保持比较高传输速率。此外 BCM5354 还集成了 Wi-Fi Multimedia 技术，能够对多媒体文件传送进行了特别优化，使得高品质 DVD 可以在无线网络顺畅在线播放。

通过 openWrt 的软件刷机，可提高路由器固件性能，实现流畅的 WiFi 信号传输，并驱动摄像头，实时回传视频信息。无线路由器的 TTL 端口与控制中心相连，接收到信息后经由 MAX 模块解码为 RS232 串口信号，输入给单片机，通过单片机解码为相应的控制指令，控制报警车的电机，实现摄像头转向和报警车的行进路线控制。另一方面，由各个传感器采集的火势信息和视频信息，经由相反的路径有无线路由再传回给控制终端 PC 机，实现远程监测功能

3.4.6 串口模块

本系统采用 MAX232 芯片完成 TTL 与 EIA RS-232C 间进行双向电平转换，由 RS-232C 正负电压表示的逻辑状态转换为 TTL 高低电平表示的逻辑状态，从而实现与带 TTL 接口的无线路由直接相连。

MAX232 芯片说明：

MAX232 为电脑的 RS-232 标准串口设计的单电源电平转换芯片，符合 RS-232C 技术标准，需单 +5V 电源供电；片载电荷泵具有升压、电压极性反转能力，能够产生 +10V 和 -10V 电压；功耗低，典型供电电流 5mA；内部集成 2 个 RS-232C 驱动器；内部集成两个 RS-232C 接收器

第一部分是电荷泵电路。由 1、2、3、4、5、6 脚和 4 只电容构成。功能是产生 +12v 和 -12v 两个电源，提供给 RS-232 串口电平的需要。

第二部分是数据转换通道。由 7、8、9、10、11、12、13、14 脚构成两个数据通道。

其中 13 脚 (R1IN)、12 脚 (R1OUT)、11 脚 (T1IN)、14 脚 (T1OUT) 为第一数据通道。8 脚 (R2IN)、9 脚 (R2OUT)、10 脚 (T2IN)、7 脚 (T2OUT) 为第二数据通道。TL/CMOS 数据从 T1IN、T2IN 输入转换成 RS-232 数据从 T1OUT、T2OUT 送到电脑 DB9 插头;DB9 插头的 RS-232 数据从 R1IN、R2IN 输入转换成 TTL/CMOS 数据后从 R1OUT、R2OUT 输出。

第三部分是供电。15 脚 GND、16 脚 VCC (+5v)。

MAX232 实物图和转换电路如图 3.30 所示

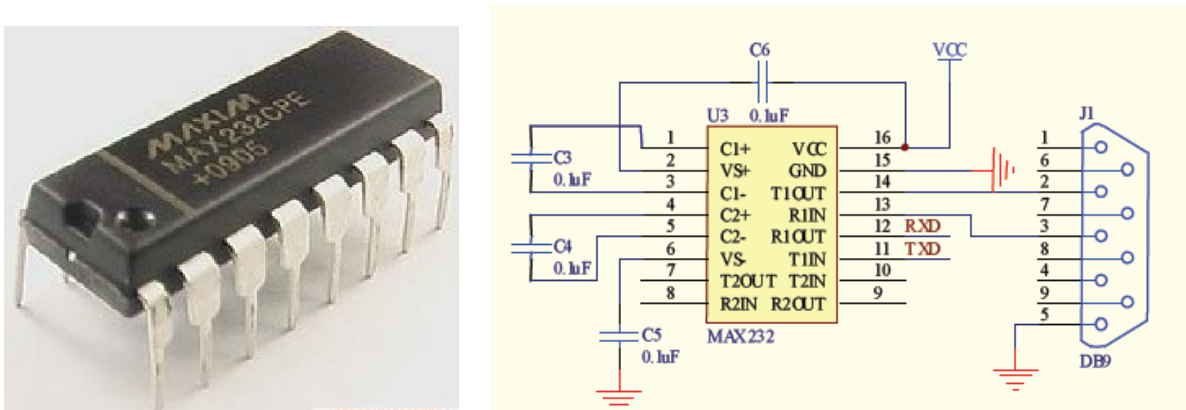


图 3.30 MAX232 转换电路

3.4.7 电机驱动模块

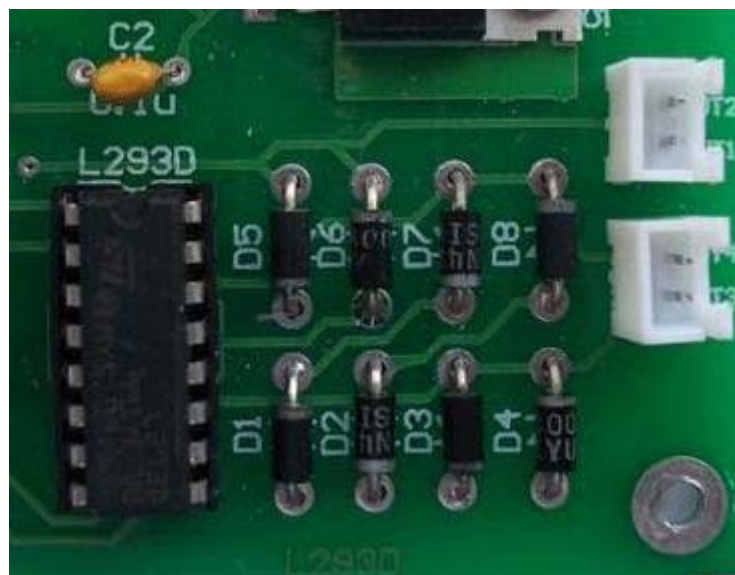


图 3.31 电机驱动实物图

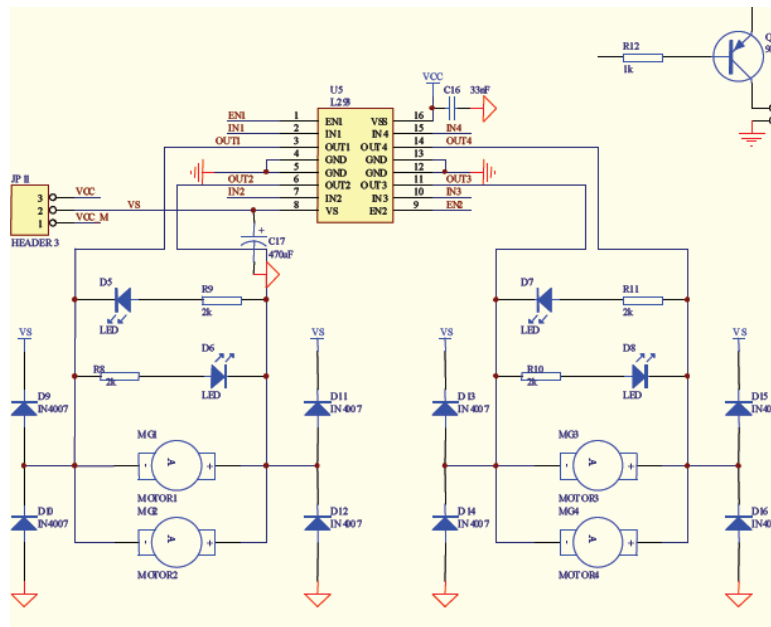


图 3.32 电机驱动原理图

表 3-1 电机驱动状态表

电机	旋转方式	控制端 IN1	控制端 IN2	控制端 IN3	控制端 IN4	输入 PWM 信号 改变脉宽可调速	
						调速 端 A	调速端 B
M1	正转	高	低	/	/	高	/
	反转	低	高	/	/	高	/
	停止	低	低	/	/	高	/
M2	正转	/	/	高	低	/	高
	反转	/	/	低	高	/	高
	停止	低	低	/	/	/	高

电动小车的电机驱动及控制

一个电动小车整体的运行性能，首先取决于它的电池系统和电机驱动系统。电动小车的驱动系统一般由控制器、功率变换器及电动机三个主要部分组成。电动小车的驱动不但要求电机驱动系统具有高转矩重量比、宽调速范围、高可靠性，而且电机的转矩-转速特性受电源功率的影响，这就要求驱动具有尽可能宽的高效率区。我们所使用的电机一般为直流电机，主要用到永磁直流电机、伺服电机及步进电机三种。直流电机的控制很简单，性能出众，直流电源也容易实现。本文即主要介绍这种直流电机的驱动及控制。

电机驱动的选择

针对不同的电机，应该选择相对应驱动。简单地来说，功率大的电机应该选用内阻小，电流容许大的驱动，功率小的电机就可以选用较低功率的驱动。电机驱动较常规的方法是采用 PWM 控制。驱动电路既可以直接采用 MC33886 电机驱动芯片，也可以采用大功率 MOS 管来自行设计电机驱动电路，还可以用场效应管搭建 H 桥电路。

方案一：

采用大功率 MOS 管组成电机驱动电路；用这个方法电路非常简单，控制只需要一路 PWM，在管子上消耗的电能也比较少，可以有效地避免多片 MC33886 并联时由于芯片分散性导致的驱动芯片某些片发热某些不发热的现象。但是缺点是不能控制电机的电流方向，在小车的刹车的性能的提升上明显有弱势，而且电流允许值也比较小。

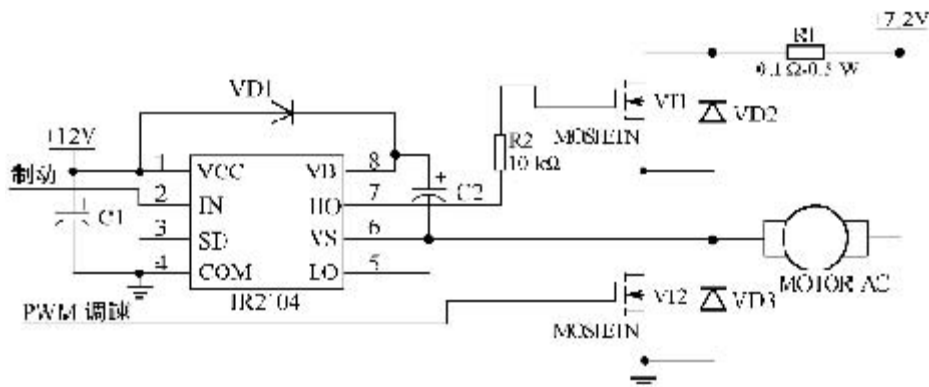


图 3.33 采用大功率 MOS 管组成电机驱动电路

方案二：

通过电机驱动模块，控制驱动电机两端电压来对模型车加速运行，或对其进行制动，采用飞思卡尔半导体公司的集成桥式驱动芯片 MC33886。MC33886 最大驱动电流为 5A，导通电阻为 140 毫欧姆，PWM 频率小于 10KHz，具有短路保护、欠压保护、过温保护等功能。体积小，使用简单，但由于是贴片的封装，散热面积比较小，长时间大电流工作时，温升较高，如果长时间工作必须外加散热器，而且 MC33886 的工作内阻比较大，又有高温保护回路，使用不方便。

方案三：

图 3.34 和图 3.35 就是一种简单的 H 桥电路，它由 2 个 P 型场效应管 Q1、Q2 与 2 个 N 型场效应管 Q3、Q3 组成，P 型管在栅极低电平时导通，高电平时关闭；N 型管在栅极高电平时导通，低电平时关闭，场效应管是电压控制型元件，栅极通过的电流几乎为

“零”。正因为这个特点，在连接好下图电路后，控制臂 1 置高电平（ $U=VCC$ ）、控制臂 2 置低电平（ $U=0$ ）时，Q1、Q4 关闭，Q2、Q3 导通，电机左端低电平，右端高电平，所以电流沿箭头方向流动。设为电机正转。

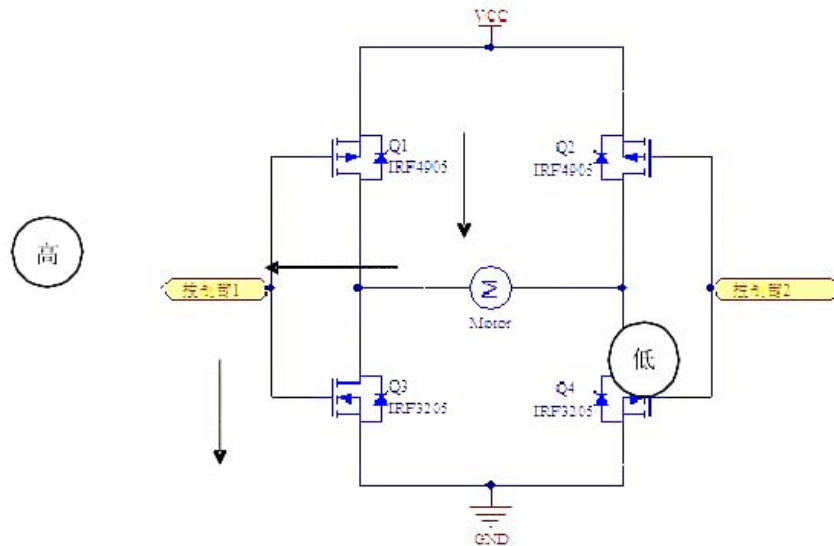


图 3.34 H 桥电路（高）

控制臂 1 置低电平、控制臂 2 置高电平时，Q2、Q3 关闭，Q1、Q4 导通，电机左端高电平，右端低电平，所以电流沿箭头方向流动。设为电机反转。

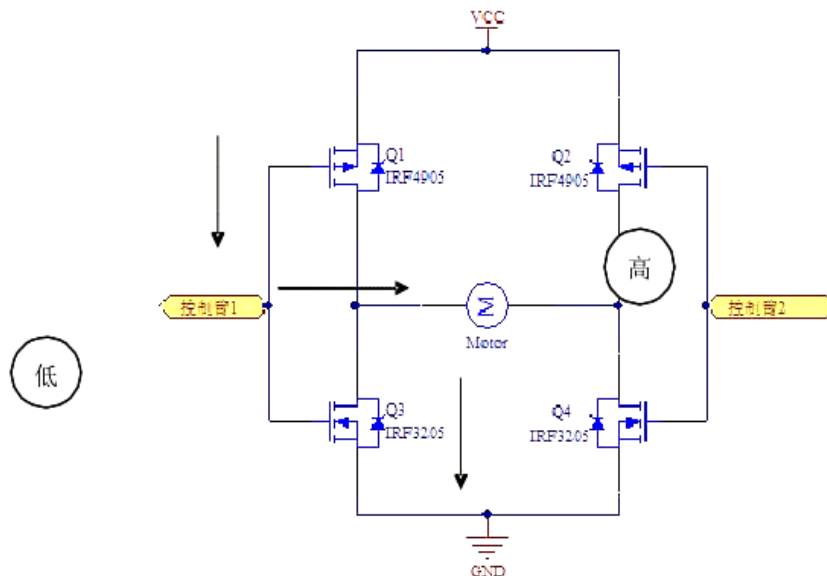


图 3.35 H 桥电路（低）

当控制臂 1、2 均为低电平时，Q1、Q2 导通，Q3、Q4 关闭，电机两端均为高电平，电机不转；当控制臂 1、2 均为高电平时，Q1、Q2 关闭，Q3、Q4 导通，电机两端均为低

电平，电机也不转。所以，此电路有一个优点就是无论控制臂状态如何，H 桥都不会出现“共态导通”（短路），很适合我们使用。

方案四：

采用两片 BTS7970B, BTS7970, BTN7960, BTN7971 搭成 H 桥来驱动电机，原理跟 MOS 管搭建 H 桥相似，此种驱动方法驱动电流大，可以达到 18A 左右。

参考电路图如图 3.36 所示

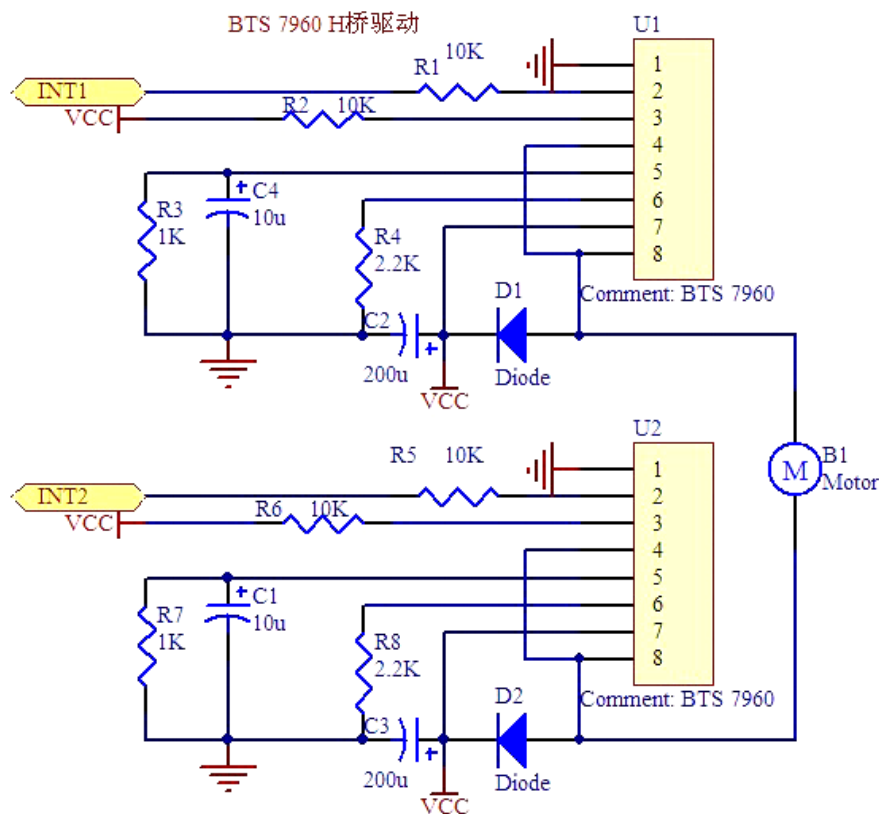


图 3.36 BTS7970B 驱动电路

目前，在智能车的应用上，方案三、方案四比较常用。

总结

以上主要分析了电机的全桥式驱动电路，这是直流电机调速使用最多的调速方法。目前市场上有很多种电机驱动的集成电路，效率高，电路简单，使用也比较广泛，但是其驱动方法大多与全桥式驱动一样。PWM 控制方法配合桥式驱动电路，是目前直流电机调速最普遍的方法。

3.4.8 抗干扰处理

在电子系统设计中，应充分考虑并满足抗干扰性的要求，避免在设计完成后再去进

行抗干扰的补救措施。

（1）抑制干扰源

抑制干扰源就是尽可能的减小干扰源的 du/dt 错误！未找到引用源。、 di/dt 错误！未找到引用源。，这是抗干扰设计中最优先考虑和最重要的原则。减小干扰源的 du/dt 错误！未找到引用源。主要是通过干扰源两端并联电容来实现。减小干扰源的 di/dt 错误！未找到引用源。则是在干扰源回路串联电感或电阻以及增加续流二极管来实现。

本设计中所采取的措施有：电机两端增加续流二极管，消除断开线圈时产生的反电动势干扰；同时给电机加滤波电路；7.2V 电源两端并联两个电感，起到滤波的作用，其中一个大电容，大小为 $470\ \mu\text{F}$ ，一个小电容，大小为 $0.01\ \mu\text{F}$ （常用的 103）。布线时避免 90° 折线，减少高频噪声发射。

（2）切断干扰传播路径

干扰的传播路径可分为传导干扰和辐射干扰两类，所谓传导干扰是指通过导线传播到敏感器件的干扰。高频干扰噪声和有用信号的频带不同，可以通过在导线上增加滤波器的方法切断高频干扰噪声的传播，有时也可加隔离光耦来解决。电源噪声的危害最大，要特别注意。所谓辐射干扰是指通过空间辐射传播到敏感器件的干扰。一般的解决方法是增加干扰源与敏感器件的距离，用地线把它们隔离和在敏感器件上加蔽罩。

本设计中所采取的措施有：充分考虑电源对单片机的影响，给单片机电源加滤波电路，同时对单片机的 5V 电源单独供电；单片机的 I/O 口与外部器件（控制电机、传感器、开关等）之应加光隔隔离；电路板合理分区（如强、弱信号，模拟、数字信号分开）。

（3）提高敏感器件的抗干扰性

提高敏感器件的抗干扰性是指从敏感器件这边考虑尽量减少对干扰噪声的拾取，以及从不正常状态尽快恢复的方法。本设计中所采取的措施有：布线时，电源线和地线尽量粗；尽量降低单片机的晶振和选用低速数字电路；IC 器件尽量直接焊在电路板上。

经过长时间的运行和改进，系统稳定性已经基本达到了我们的要求。

3.5 本章小结

硬件设计在智能车整个系统的设计中非常重要。因为车体的硬件性能直接影响到智能车运动的极限参数，如最快响应时间、最大视野范围等。为了提高硬件的性能主要从三个方面进行设计，即车模调整、采集传感器的安装、电机的驱动电路设计。本章主要内容就是介绍所述智能车的整体硬件设计方案，最主要的工作是对 STC12C5A60S2 的最小系统板的开发。

第四章 软件设计

基于摄像头的智能车的软件设计主要有以下几个方面：视频信号采集、无线通讯、电机控制。其中视频信号的采集是最先要考虑的，如果视频信号采集不稳定的话，则直接影响到后面的控制结果，造成智能车的不稳定，严重影响智能车运行。

对于电机的控制则是最终体现车体稳定运行的关键，上一章所述的硬件设计师软件设计的基础。只有硬件设计过硬才能得到一个比较好的控制实体，之后才能更容易的控制。

4.1 视频信号采集

视频信号的采集是下面各控制算法的基础，良好的视频采集程序能够提高软件运行的稳定性。

4.1.1 系统程序简介

硬件平台结构一旦确定，大的功能框架即形成，软件在硬件平台上构筑，完成各部分硬件的控制和协调工作。系统功能是由软硬件共同实现的，由于软件的可伸缩性，最终实现的系统功能可强可弱，差别可能很大。因此，软件是系统的灵魂。软件采用模块化设计方法，不仅易于编程和调试，也可减小软件故障率和提高软件的可靠性。

我们的小车控制是通过 STC12C5A60S2 单片机控制驱动电路和给舵机送控制信号，然而这些控制信号的命令又是电脑等终端设备通过无线路由器串口传送给单片机的，所以在程序中我们需要设计到串口的使用、定时器使用、I/O 口的使用。

4.2 程序流程图

智能 WIFI 小车的软件编写主要分为命令接收部分、命令执行部分、以及具体的信号的采集和处理。

4.2.1 主程序流程图

系统主程序模块主要完成对系统中各模块电路的初始化等工作，主要包括对定时器、串口中断、外部中断的初始化，同时执行电脑等终端设备所发送的命令，等待外部中断以及根据所需要的功能进行相应操作。软件总体设计及程序流程如下图 4.1 所示。

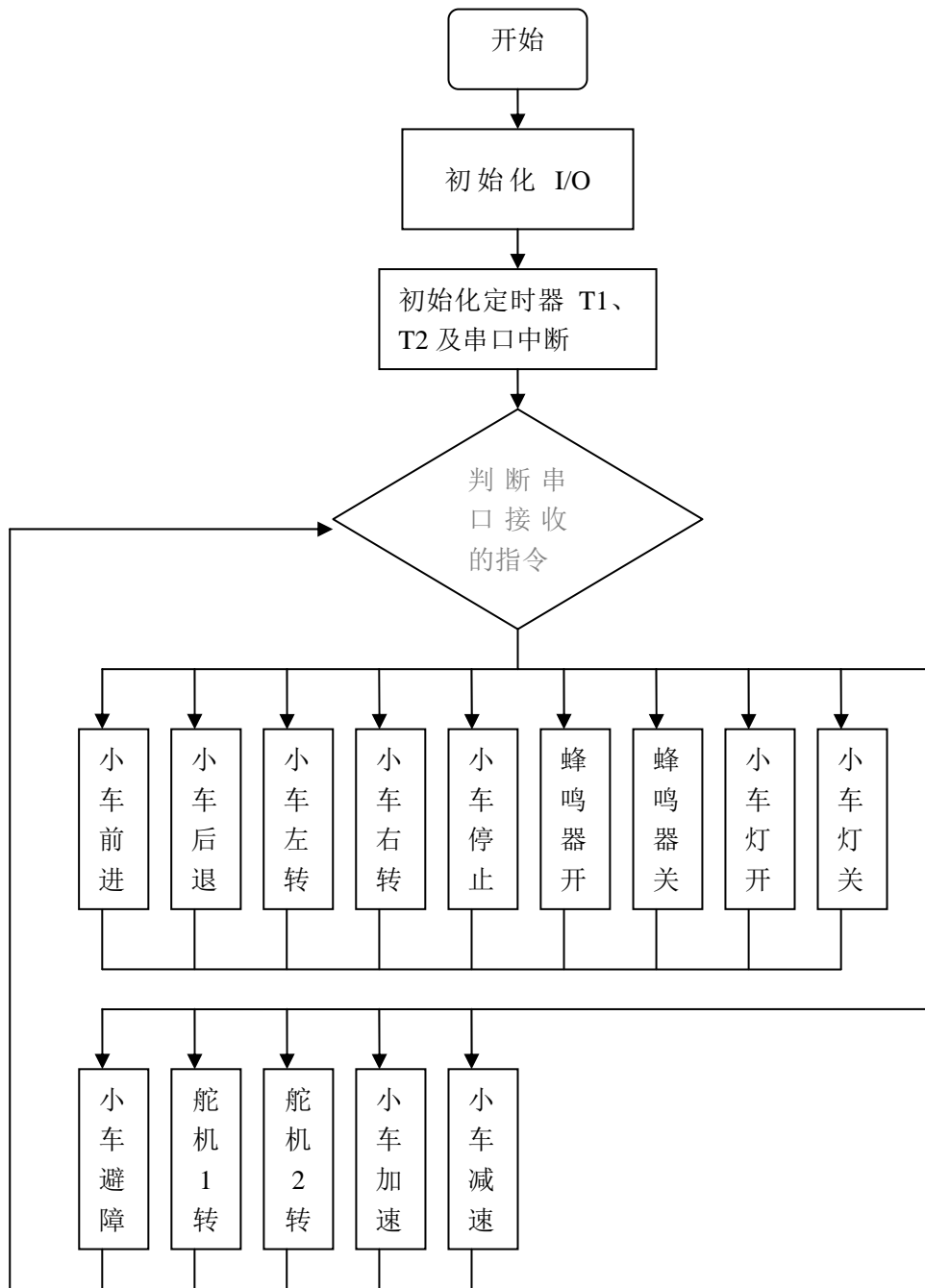


图 4.1 主函数流程图

4.2.2 串口中断子函数流程图

通过串口中断，实现单片机和路由器进行通讯。进入中断后关掉中断，避免数据信号重复引起中断。根据命令数据发送的规律，将命令解码储存在相应的二维数组当中，方便在主函数中调用。具体流程图如下图 4.2 所示。

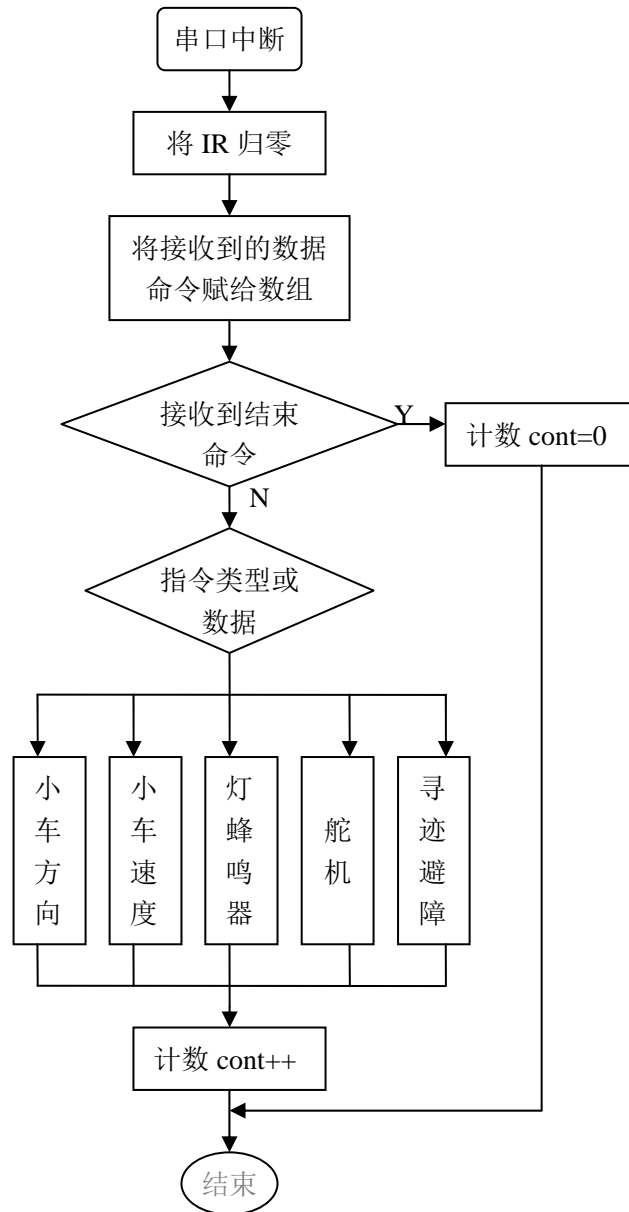


图 4.2 串口中断流程图

4.2.3 定时器 2 中断子函数流程图

通过定时器的初始值和循环次数，确定出舵机 PWM 信号的 20Ms 的周期，通过控制高电平持续的中断次数，来确定具体的 PWM 信号。其流程图如下图 4.3 所示。

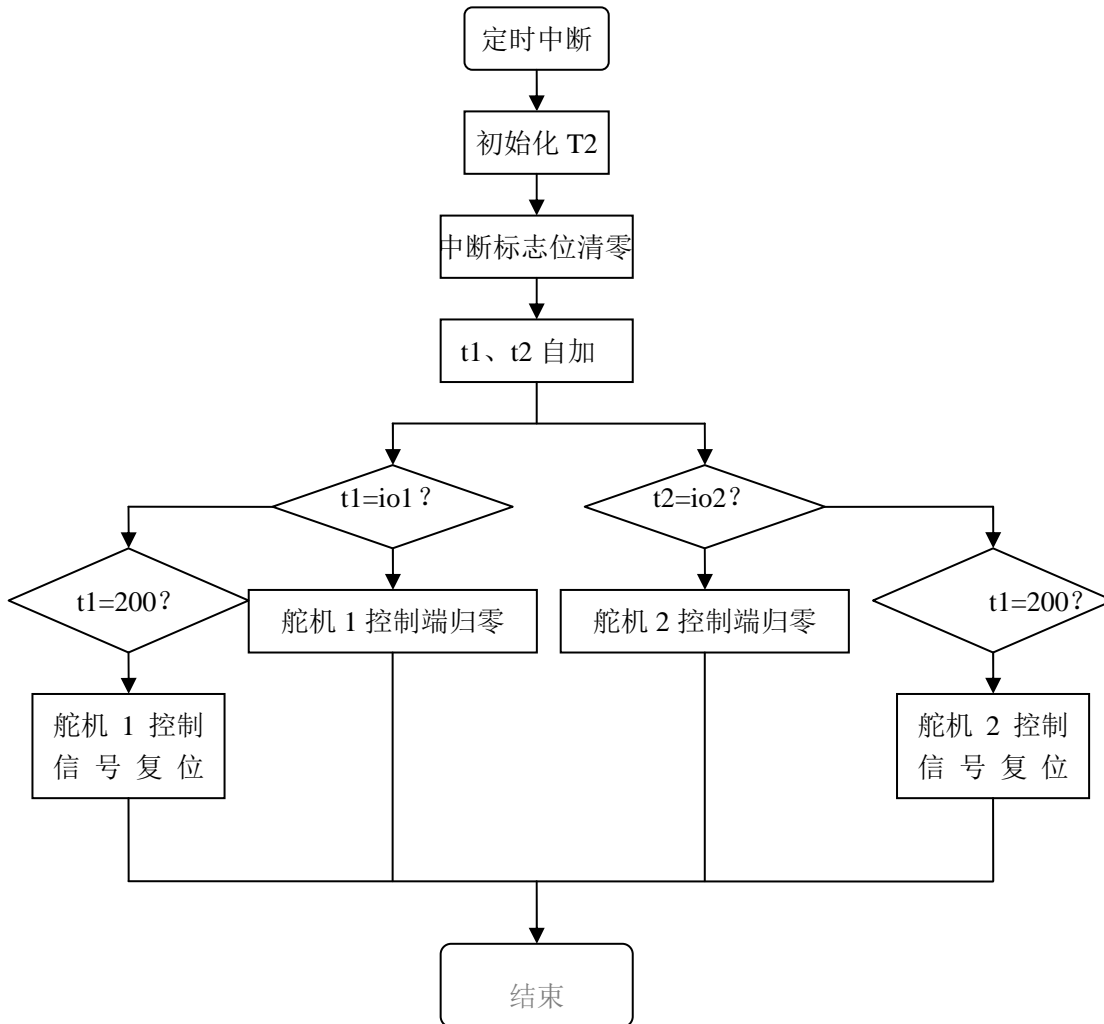


图 4.3 定时中断函数流程图

4.2.4 避障子函数流程图

根据传感器探测到小车四周障碍物状态，从而确定驱动几个电机的旋转方向，也就确定了小车的运动状态。其流程图如下图 4.4 所示。

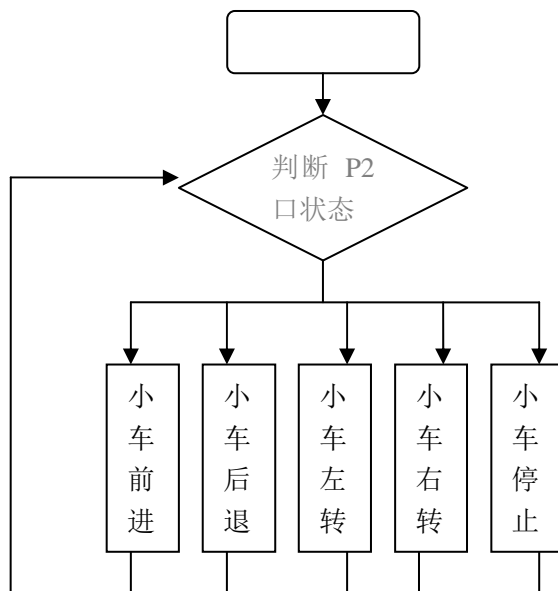


图 4.4 避障子函数流程图

4.3 PWM 调速

脉宽调制（PWM: (Pulse Width Modulation)）是利用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术，广泛应用在从测量、通信到功率控制与变换的许多领域中。PWM 是一种对模拟信号电平进行数字编码的方法。通过高分辨率计数器的使用，方波的占空比被调制用来对一个具体模拟信号的电平进行编码。PWM 信号仍然是数字的，因为在给定的任何时刻，满幅值的直流供电要么完全有 (ON)，要么完全无 (OFF)。电压或电流源是以一种通 (ON) 或断 (OFF) 的重复脉冲序列被加到模拟负载上去的。通的时候即是直流供电被加到负载上的时候，断的时候即是供电被断开的时候。只要带宽足够，任何模拟值都可以使用 PWM 进行编码。

4.3.1 直流电机调速原理

直流电机调速原理根据励磁方式不同，直流电机分为自励和他励两种类型。不同励

磁方式的直流电机机械特性曲线有所不同。对于直流电机来说，人为机械特性方程式为：

$$n = \frac{U_N}{K_e \phi_N} - \frac{R_{ad} + R_a}{K_e K_t \phi_N^2} T = n_0 - \Delta n$$

式中， U_N 、 ϕ_N ——额定电枢电压、额定磁通量；

K_e 、 K_t ——与电机有关的常数；

R_{ad} 、 R_a ——电枢外加电阻、电枢内电阻；

n_0 、 Δn ——理想空载转速、转速降。

分析公式可得，当分别改变 U_n 、 ϕ_n 和 R_{ad} 时，可以得到不同的转速凡，从而实现
对速度的调节。由于 $\phi = F(I_f)$ ，当改变励磁电流 I_f 时，可以改变磁通量西的大小，从
而达到变磁通调速的目的。但由于励磁线圈发热和电动机磁饱和的限制，电动机的励磁
电流 I_f 和磁通量 ϕ 只能在低于其额定值的范围内调节，故只能弱磁调速。而对于调节
电枢外加电阻 R_{ad} 时，会使机械特性变软，导致电机带负载能力减弱。

对于他励直流电机来说，当改变电枢电压“时，分析人为机械特性方程式，得到人
为特性曲线

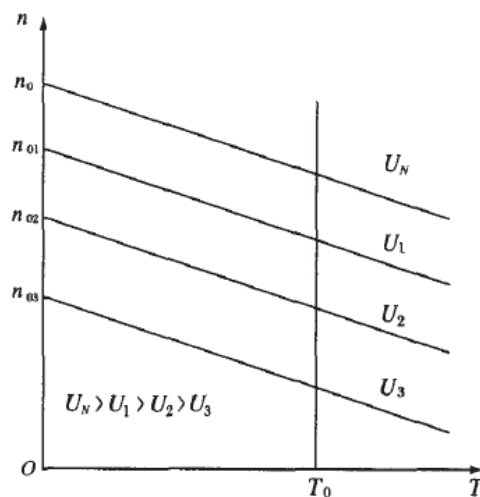


图 3.5 直流电机机械特性曲线

如图 3.5 所示。理想空载转速 n_0 随电枢电压升降而发生相应的升降变化。不同电
枢电压的机械特性曲线相互平行，说明硬度不随电枢电压的变化而改变，电机带负载能
力恒定。当我们平滑调节他励直流电机电枢两端电压时，可实现电机的无级调速。

基于以上特性，改变电枢电压，实现对直流电机速度调节的方法被广泛采用。改变
电枢电压可通过多种途径实现，如晶闸管供电速度控制系统、大功率晶体管速度控制系

统、直流发电机供电速度控制系统及晶体管直流脉宽调速系统等。

4.3.2 PWM 基本原理及其实现方法

PWM 是通过控制固定电压的直流电源开关频率，从而改变负载两端的电压，进而达到控制要求的一种电压调整方法。PWM 可以应用在许多方面，如电机调速、温度控制、压力控制等。

在 PWM 驱动控制的调整系统中，按一个固定的频率来接通和断开电源，并根据需要改变一个周期内“接通”和“断开”时间的长短。通过改变直流电机电枢上电压的“占空比”来改变平均电压的大小，从而控制电动机的转速。因此，PWM 又被称为“开关驱动装置”。

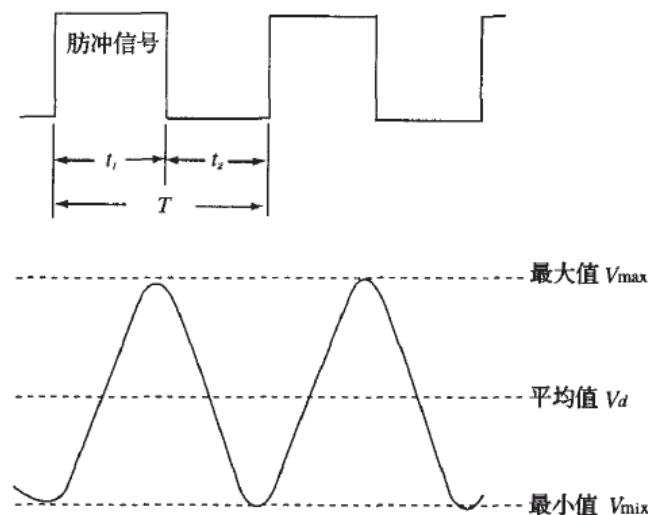


图 4.6 电枢电压“占空比”与平均电压关系图

如图 4.6 所示，在脉冲作用下，当电机通电时，速度增加；电机断电时，速度逐渐减少。只要按一定规律，改变通、断电的时间，即可让电机转速得到控制。

设电机始终接通电源时，电机转速最大为 V_{max} ，设占空比为 $D=t_1 / T$ ，则电机的平均速度为

$$V_d = V_{max} \cdot D$$

式中， V_d ——电机的平均速度；

V_{max} ——电机全通电时的速度(最大)；

$D = t_1 / T$ ——占空比。

由公式可见，当我们改变占空比 $D=t_1 / T$ 时，就可以得到不同的电机平均速度 V_d ，从而达到调速的目的。严格地讲，平均速度 V_d 与占空比 D 并不是严格的线性关系，在一般的应用中，可以将其近似地看成线性关系。

PWM 信号的产生通常有两种方法：一种是软件的方法；另一种是硬件的方法。硬件方法的实现已有很多文章介绍，这里不做赘述。本文主要介绍利用单片机对 PWM 信号的软件实现方法。

MCS-51 系列典型产品 8051 具有两个定时器 T₀ 和 T₁。通过控制定时器初值 T₀ 和 T₁，从而可以实现从 8051 的任意输出口输出不同占空比的脉冲波形。由于 PWM 信号软件实现的核心是单片机内部的定时器，而不同单片机的定时器具有不同的特点，即使是同一台单片机由于选用的晶振不同，选择的定时器工作方式不同，其定时器的定时初值与定时时间的关系也不同。因此，首先必须明确定时器的定时初值与定时时间的关系。如果单片机的时钟频率为 f，定时器 / 计数器为 N 位，则定时器初值与定时时间的关系为：

$$t_1 = (2^n - T_w) \times \frac{N}{f \times 10^6}$$

式中, T_w ——定时器定时初值;
 N ——一个机器周期的时钟数。

N 随着机型的不同而不同。在应用中，应根据具体的机型给出相应的值。这样，我们可以通过设定不同的定时初值 T_w ，从而改变占空比 D，进而达到控制电机转速的目的。

控制程序的设计有两种方法：软件延时法和计数法。软件延时法的基本思想是：首先求出占空比 D，再根据周期 T 分别给电机通电 M 个单位时间 t_0 ，所以 $M=t_1 / t_0$ 。然后，再断电 M 个单位时间，所以 $M=t_2 / t_0$ 。改变 M 和 M 的值，从而也就改变了占空比 D。计数法的基本思想是：当单位延时个数 M 求出之后，将其作为给定值存放在某存储单元中。在通电过程中，对通电单位时间 t_0 的次数进行计数，并与存储器的内容进行比较。若不相等，则继续输出控制脉冲，直到计数值与给定值相等，使电机断电。

以 89C51 单片机为核心的油辊电机控制系统控制简图如图 4.7 所示。单片机按预定算法计算出定时初值，由软件转换成 PWM 信号，并由 P3.0 输出，经放大器及固态继电器输出给电机，从而控制电机得电与失电。

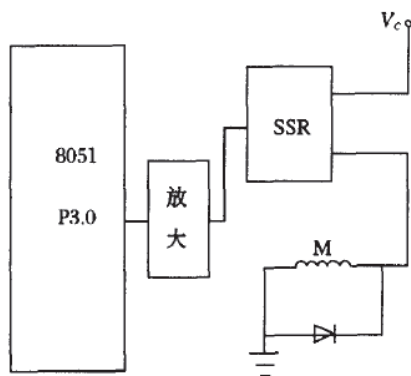


图 4.7 油辊电机控制系统控制简图

软件采用计数法加软件延时法进行设计。如图4.8所示，单片机上电后，系统进入准备状态。当按动启动按钮后，判断P 3.0是否高电平。然后，根据P 3.0分别进入计数值与预定值比较环节，当计数值与预定值相等时，清计数寄存器，并置P 3.0为相反状态，进入延时程序。根据不同的加减速按钮，调整P3.0输出高低电平时的预定值，从而可以控制P 3.0输出高低电平时的延时时间，进而控制电压的大小。控制程序成功的应用于纺织机油辊电机的加减速控制。

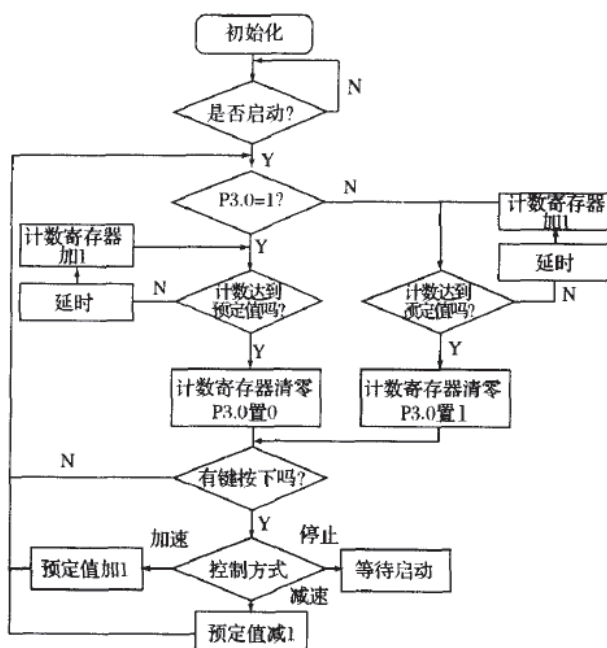


图 4.8 流程图

通过单片机来实现电机调整有多种途径。相对于其他用硬件或者硬软结合的方法实现对电机进行调整，采用PWM用纯软件的方法来实现调速过程，具有更大的灵活性和更低的成本，能够充分发挥单片机的效能，对于简易速度控制系统的实现提供了一种有效的途径。对于软件，采用计数法加软件延时法进行设计的思路，为采用纯软件对电机速

度的平滑调节提供了一种不错的解决方案，经过在“油辊电机控制系统”中的实际应用证明，能够取得满意的效果。

4.4 本章小结

本章针对智能车的软件系统设计展开叙述，首先介绍了视频信号的采集，其次是无
线通讯及路由的固件刷机方法，对于电机的驱动程序则采用了速度 PWM 控制，采用 PWM
软件方法来实现的调速过程具有更大的灵活性和更低的成本，它能够充分发挥单片机的
效能，对于简易速度控制系统的实现提供了一种有效的途径。而在软件方面，采用 PLD
算法来确定闭环控制的补偿量也是由数字电路组成的直流电机闭环调速系统所不能及
的。曾经也试过用单片机直接产生 PWM 波形，但其最终效果并不理想，在使用了少量的
硬件后，单片机的压力大大减小，程序中有充足的时间进行闭环控制的测控和计算，使
得软件的运行更为合理可靠。

第五章 智能车的开发与调试

智能车的调试属系统设计的最后一个阶段。本章主要对本系统的开发环境及调试工具进行介绍。智能车的控制核心为 STC12C5A60S2 位单片机，用 keil 进行编译开发。程序的下载采用 STC 软件包自带的下载工具 STC.exe。。

5.1 编译环境

1. Keil 软件

Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统，与汇编相比，C 语言在功能上、结构性、可读性、可维护性上有明显的优势，因而易学易用。Keil 提供了包括 C 编译器、宏汇编、连接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个集成开发环境（uVision）将这些部分组合在一起。运行 Keil 软件需要 WIN98、NT、WIN2000、WINXP 等操作系统。如果你使用 C 语言编程，那么 Keil 几乎就是你的不二之选，即使不使用 C 语言而仅用汇编语言编程，其方便易用的集成环境、强大的软件仿真调试工具也会令你事半功倍。其软件界面如图 5.1 所示。

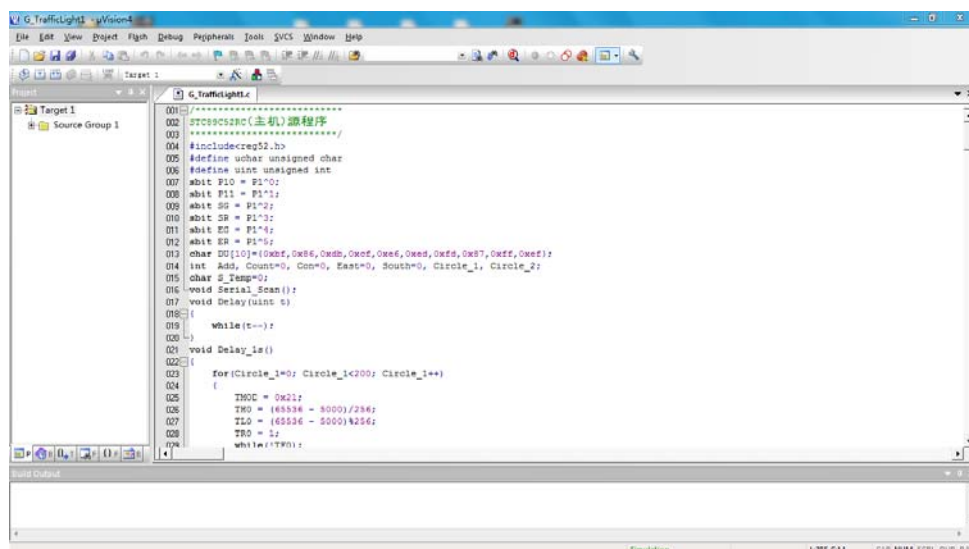


图 5.1 Keil 软件界面图

2. Proteus 软件

Proteus 软件是英国 Labcenter electronics 公司出版的 EDA 工具软件。它不仅具有其它 EDA 工具软件的仿真功能，还能仿真单片机及外围器件。它是目前最好的仿真单片机及外围器件的工具。虽然目前国内推广刚起步，但已受到单片机爱好者、从事单片机教学的教师、致力于单片机开发应用的科技工作者的青睐。Proteus 是世界上著名的 EDA

工具(仿真软件), 从原理图布图、代码调试到单片机与外围电路协同仿真, 一键切换到 PCB 设计, 真正实现了从概念到产品的完整设计。是目前世界上唯一将电路仿真软件、PCB 设计软件和虚拟模型仿真软件三合一的设计平台, 其处理器模型支持 8051、HC11、PIC10/12/16/18/24/30/DsPIC33、AVR、ARM、8086 和 MSP430 等, 2010 年即将增加 Cortex 和 DSP 系列处理器, 并持续增加其他系列处理器模型。在编译方面, 它也支持 IAR、Keil 和 MPLAB 等多种编译。其软件界面如图 5.2 所示。

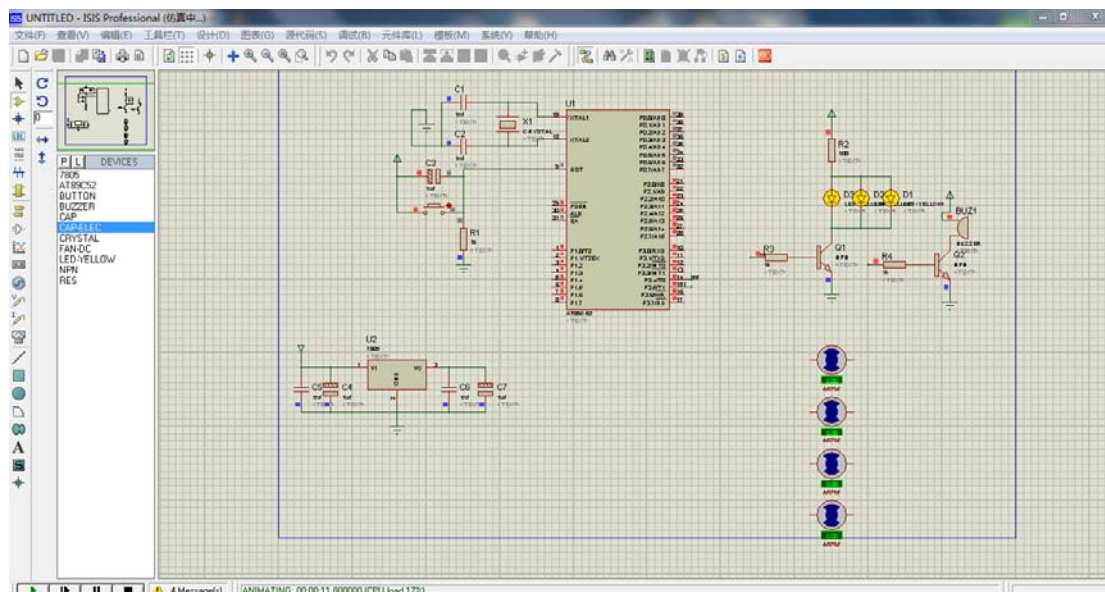


图 5.2 Proteus 软件界面

5.1.1 仿真测试

1. 系统初始化测试: 主要完成对系统中各模块电路的初始化等工作, 如在 Proteus 里面对软件的一些参数进行调试。将 Proteus 里的单片机 89C52 的晶振设置为我们实物中使用的参数 22.1184MHz, 将 T0、T1 和 T2 定时器开启并使之从单片机的 I/O 口输出, 我们通过 Proteus 软件里的示波器, 就可看出我们设置的定时器初始值是否合适, 如果不合适我们就能根据 I/O 口的波形, 修改我们的参数直到合适为止。

2. 分模块对软件进行测试, 在 Proteus 软件中我们把我们的寻迹、避障模块程序分别进行调试, 我将传感器的状态在软件中用开关替代, 在输出 I/O 引脚上加上发光二极管, 用观察输出状态。这样我们就可将外部的输入状态和输出进行对比, 判断出个模块软件的编写是否正常。

5.2 实物调试

根据系统设计方案, 此系统调试可分为硬件调试、软件调试和软硬件联调三个部分。

1. 硬件调试

由于此设计实物的焊接、装配工作量非常大，所以在电路安装完成后，首先应对系统进行整体检查，确认电路无虚焊、短路、断路等错误，然后应该对电路各功能模块进行分级调试，逐步实现系统的整体功能。

2. 软件调试

此系统的软件程序，均使用 C 语言编写。首先选用单片机开发板作为程序调试的载体，逐步对各模块子程序进行调试，然后再将各子程序模块进行有机组合、联调，最终完成整个系统的软件调试。

3. 软硬联调

系统软硬件联调的主要任务是检测系统中各模块的功能和效果，同时需要校正系统的相关数据，按实际测量数据进行调试，直至数据结果准确为止。

5.3 单片机的资源分配

表 5.1 系统资源分配表

直流电机	速度控制	2 路 16 位 PWM 输出，ENA 接 PW1，ENB 接 PW2 方向控制 IN1 接 PM0，IN2 接 PM1，IN3 接 PM2，IN4 接 PM3
	工作电压	7.2V
编码器	速度反馈	接中断口 PT7
	工作电压	5V
摄像头	图像采集	视频数据线接路由 USB 串口
	工作电压	5V
串口	与 PC 进行数据通信	接收 RXD 和发送 TXD，属 SCI, USB 转串口，把芯片的 P3.1 与 TXD 接通，P3.0 与 RXD 接通，共地线。
控制板	电源开关	1 个
	复位按钮	1 个
	指示灯	2 个

智能车的控制系统主要有视频信号采集模块，无线通讯模块，电机驱动模块及串口

模块等。为了让这些模块能正常工作就要对单片机的引脚进行合理分配。根据这些模块的功能和特点，现将所需要的资源总结如表 4.1 所示。

5.4 本章小结

调试属于智能车设计的最后一步，也是关键的一步，是决定设计成败的关键。在调试过程中主要对硬件电路及软件程序进行逐步的改动以达到预期效果。本章主要介绍了智能车的开发调试环境、调试的方法和需要改进的地方。

第六章 总结与展望

本课题对智能车的硬件和软件进行了设计，最终达到让智能车视频信号采集和速度方向精确控制的目的。本文研究工作总结如下：

- 1、设计开发了 STC12C5A60S2 单片机的最小系统板。由于其特殊的外形，更适合在智能车上安装。
- 2、使用了能刷多种开源固件的路由，大幅度增加了智能车的可操作性。
- 3、采用了两个自由度的云台系统，能自由灵活的控制摄像头的事业范围。
- 4、用 PC 上位机经路由进行无线通讯来控制智能车运行及采集视频信号，单片机最小系统能实现准确的控制。

对于我们而言，远程监控并不是最终目的，在无外界条件干扰的情况下，尽可能准确实时的传回视频信号，智能车能适应更加恶劣的环境才是我们所追求的。从这个意义上说，红外臂章是我们为了保证智能车完整运行而采取的辅助手段。

但是由于水平有限，系统中尚存在许多问题有待改进：

- 1、使用不同摄像头时路由器要进行怎样的一个处理。
- 2、控制算法的改进。本系统采用了单闭环加前馈的控制方法，这在控制算法中是比较复杂的，实际应用也验证了效果的良好，但其参数调节太过复杂，需要太长的时间调节。
- 3、电机功率太低，不能充分完成智能车稳定运行，同时要对电机驱动的选择进行改进，提高运行稳定性。

致 谢

首先，感谢于标老师的细心指导，由于老师的认真负责，我才能顺利的完成毕业设计。本文的研究、撰写直至论文的定稿，每个环节和步骤，老师都严格把关，精心指导，论文的字里行间都浸透了老师辛勤的汗水，在此向指导老师们表示深深的谢意！

在智能车电路的开发过程中宋老师，曾给了我很多帮助，对整个智能车系统的设计提供了不少的建议，指引着我克服了很多技术问题，在此向这位良师益友表示诚挚的感谢！

另外，对于一直陪同我制作智能车的伙伴，想发自肺腑地说一声谢谢，因为有他们的默默支持才能让我顺利完成毕业设计。

最后，感谢各位评阅老师，想跟你们说声：老师，您辛苦了！

参考文献

- [1]曾星星, 基于摄像头的路径识别智能车控制系统设计[J], 湖北汽车工业学院学报, 2008, 22(2): 73-78
- [2]张永枫. 单片机应用实训教程. 北京: 清华大学出版社, 2008
- [3]冯津. 基于 MC9S12 单片机的摄像头参数测量及模拟视频信号离散化[A]. 智能检测控制技术及仪表装置发展研讨会论文集, 2007: 154-160
- [4]马忠梅、籍顺心、张凯等编著, 单片机的 C 语言应用程序设计, 北京: 北京航空航天大学出版社, 1998.
- [5]谭浩强, C 程序设计, 北京: 清华大学出版社, 1999
- [6]邵贝贝. 单片机嵌入式应用的在线开发方法[M]. 北京: 清华大学出版社, 2004
- [7]刘守义. 单片机应用技术. 西安: 西安电子科技大学出版社, 2007
- [8]Wang Xiuquan. Route Identification and Direction Control of Smart Car Based on CMOS Image Sensor. ISECS International Colloquium on Computing, Communica
- [9]幻德科技 Wifi 使用手册 2012
- [10]百度文库 STC12C5A60S2 超详细中文资料 2012
- [11]孙肖子, 张企民 . 模拟电子技术基础 . 西安:西安电子科技大学出版社, 2000
- [12]孙同景. 陈桂友. 单片机原理及嵌入式开发技术[M]. 机械工业出版社, 2008
- [13]李群芳. 单片微型计算机与接口技术. 北京: 电子工业出版社, 2007
- [14]李伯成. 嵌入式系统可靠性设计. 北京: 电子工业出版社, 2006
- [15]徐 玮. C51单片机高效入门. 北京: 机械工业出版社, 2006
- [13]曹志平. 现代通信原理. 北京: 清华大学出版社, 2003, 4
- [14]南利平. 通信原理简明教程. 北京: 清华大学出版社, 2000, 2

附件 A：智能车程序代码

//Wifi 智能小车

```
#include <STC12C5A.h> //包含 STC12C5A 系列单片机的头文件
```

```
#include "1602.h"
```

```
#include "eeprom.h"
```

```
sbit LED1=P3^0; //定义两只 LED 到 P3^0, P3^1
```

```
sbit LED2=P3^1;
```

```
sbit Light_cam=P3^2;
```

```
sbit Light_B=P2^3; //控制车身的 LED
```

```
sbit beed=P2^7;
```

```
sbit KEY1=P3^6;
```

```
sbit KEY2=P3^7;
```

```
//电机控制端
```

```
sbit EN1=P2^0;
```

```
sbit IN1=P2^1;
```

```
sbit IN2=P2^2;
```

```
sbit EN2=P2^4;
```

```
sbit IN3=P2^5;
```

```
sbit IN4=P2^6;
```

```
// 驱动其它设备，如激光，
```

```
sbit EN1_S=P1^0;
```

```
sbit IN1_S=P1^1;
```

```
sbit IN2_S=P1^2;
```

```
sbit EN2_S=P1^3;
```

```
sbit IN3_S=P1^4;
```

```
sbit IN4_S=P1^5;
```

```
//红外避障传感器
```

```
sbit IROA_R=P3^3;
```

```
sbit IROA_M=P3^4;
```

```
sbit IROA_L=P3^5;
```

```
#define sensor_IROA P3 //跟上面定义一样，取 P3^3-P3^5

//PWM 控制云台
sbit SPWM1=P1^6; //控制摄像头左右活动
sbit SPWM2=P1^7; //控制摄像头上下活动

unsigned long PWM_cnt=0;//

unsigned int PWM_G1=33; //初始化中心值
unsigned int PWM_G2=33; //

int speed_R=0; //定义两个电机的速度，有正负之分，为正是，正转。（范围—
100~100）
int speed_L=0;

int pre_RS=0;
int pre_LS=0;

int set_RS=0;
int set_LS=0;

int speed_A=0;
unsigned int cnt_s=0; //用于计数 PWM 的点空比 0=100;

unsigned long cnt_b=0;
unsigned long cnt_beed=0; //用于控制蜂鸣器
unsigned int flag_beed=1;
unsigned int B_counter=0;

unsigned char R_date[3]={0};
unsigned char RBuff_date[4]={0};

unsigned char pattern=0; //0:stop 1:start

unsigned long cnt_ser=0; //用于舵机直流电机控制
unsigned char flag_ser=0;

unsigned char flag_phone=0; //用于手机控制标记
unsigned char check_phone=0;
unsigned char flag_PC=0;

void delay(unsigned long t) //软件延时带参数 t
```

```
{
    unsigned int j,k;
    for(t;t>0;t--)
    for(j=20;j>0;j--)
    for(k=100;k>0;k--);
}
```

```
unsigned char sen_inp_IROA()
{
    unsigned IROA_date;
    IROA_date=sensor_IROA;
    IROA_date=~IROA_date;
    IROA_date &=0x38;
    IROA_date>>=3;
    return(IROA_date);
}
```

```
void beed_ctrl(unsigned long time , unsigned int num)
{
    beed=0;
    cnt_beed=2*time;
    B_counter=2*num;
    beed=1;
    cnt_b=0;
}
```

```
void display_1602(unsigned int addr,unsigned char temp)
{
    unsigned int i,w[2];
    w[0]=temp/16;
    w[1]=temp%16;

    for(i=0;i<2;i++)
    {
        switch(w[i])
        {
            case 0: print(addr+i,"0"); break;
            case 1: print(addr+i,"1"); break;
            case 2: print(addr+i,"2"); break;
            case 3: print(addr+i,"3"); break;

            case 4: print(addr+i,"4"); break;
        }
    }
}
```

```

        case 5: print(addr+i,"5"); break;
        case 6: print(addr+i,"6"); break;
        case 7: print(addr+i,"7"); break;

        case 8: print(addr+i,"8"); break;
        case 9: print(addr+i,"9"); break;
        case 10: print(addr+i,"A"); break;
        case 11: print(addr+i,"B"); break;

        case 12: print(addr+i,"C"); break;
        case 13: print(addr+i,"D"); break;
        case 14: print(addr+i,"E"); break;
        case 15: print(addr+i,"F"); break;
        default: break;
    }
}

void display_1602_10(unsigned int addr,unsigned char temp)
{
    unsigned int i,w[3];
    w[0]=temp/100;
    w[1]=temp/10%10;
    w[2]=temp%10;

    for(i=0;i<3;i++)
    {
        print2(addr+i,w[i]);
    }
}

/*****采用直流电机控制摄像头上下活动*****/
void ser_up()
{
    EN1_S=1;
    IN1_S=0;
    IN2_S=1;
    flag_ser=1;
}

void ser_down()
{
    EN1_S=1;
    IN1_S=1;

```

```
IN2_S=0;
flag_ser=1;
}

/*****采用舵机控制摄像头上下左右活动*****/
void cam_left()
{
    PWM_G1++;
    if(PWM_G1>53)
    {
        PWM_G1=53;
    }
}

void cam_right()
{
    PWM_G1--;
    if(PWM_G1<13)
    {
        PWM_G1=13;
    }
}

void cam_up()
{
    PWM_G2--;
    if(PWM_G2<13)
    {
        PWM_G2=13;
    }
}

void cam_down()
{
    PWM_G2++;
    if(PWM_G2>53)
    {
        PWM_G2=53;
    }
}

void cam_mid()
{
```



```

    PWM_G1=Byte_Read(0x00);;
    PWM_G2=Byte_Read(0x01);;
}

/*****采用 L293 驱动前车灯*****/
void LED_F_on()
{
    EN2_S=1;
    IN4_S=1;
}
void LED_F_off()
{
    EN2_S=1;
    IN4_S=0;
}
/*****采用 L293 驱动激光灯*****/
void Laser_on()
{
    EN2_S=1;
    IN3_S=1;
}
void Laser_off()
{
    EN2_S=1;
    IN3_S=0;
}

/*****采用继电器驱动水泵*****/

/*****init To*****/
void init_T0(void)
{
    TMOD|=0x01; //定时器工作在模式 1
    TH0=0xff; //定时初值 TH0=0xff TL0=0x49 定时 100uS
    TL0=0x49;
    TR0=1;
    ET0=1;
}

/*****T0 interrupt*****/
void timer0(void) interrupt 1
{

```

```

TH0=0xff;
TL0=0x49;

cnt_s++; //PWM 占空比计数 0-100
if(cnt_s>=speed_R)
{
    EN1=0;
}
if(cnt_s>=speed_L)
{
    EN2=0;
}
if(cnt_s>=100)
{
    EN1=1;
    EN2=1;
    cnt_s=0;
}

}

/*****motor control*****/
void motor( int m_R, m_L)
{
    pre_RS=m_R;
    pre_LS=m_L;

    //right motor
    if(m_R>0)
    {
        IN1=0;
        IN2=1;
        speed_R=m_R;
    }
    else if (m_R<0)
    {
        IN1=1;
        IN2=0;
        m_R=-m_R;
        speed_R=m_R;
    }
}

```

```

else
{
    IN1=0;
    IN2=0;
}

//left motor
if(m_L>0)
{
    IN3=0;
    IN4=1;
    speed_L=m_L;
}
else if(m_L<0)
{
    IN3=1;
    IN4=0;
    m_L=-m_L;
    speed_L=m_L;
}
else
{
    IN3=0;
    IN4=0;
}
}

/*****初始化系统所有的参数*****/
void init_system()
{
    unsigned int i;
    motor(0,0);
    LED_F_off();
    Laser_off();
    beed=1;

    speed_A=0;
    for(i=0;i<3;i++)
    {
        R_date[i]=0;
        RBuff_date[i]=0;
    }
}

/*****速度记录*****/

```

```
int Mark_Speed(int set)
{
    int s_temp;

    if(set==0)
    {
        s_temp=0;
    }
    else if(set<0)
    {
        s_temp=-speed_A;
    }
    else
    {
        s_temp=speed_A;
    }
    return s_temp;
}
```

/******UART*****采用独立波特率发生器*****/

```
void initUart(void) //4800 波特率
{
    SCON=0x50;
    PCON=0xB0;
    BRT=0xfa;
    AUXR=0x11;
    ES=1;
}
```

```
void SCI_TX(unsigned char date)
{
    SBUF=date;
    while(!TI);
    TI=0;
}
```

```
void uart(void) interrupt 4
{

    if(RI)
    {
        RI=0;
        //add your code here.
    }
}
```

```

if(flag_phone==0)
{
    check_phone= RBuff_date[0];
    RBuff_date[0]= RBuff_date[1];
    RBuff_date[1]= RBuff_date[2];
    RBuff_date[2]=SBUF;

    //判断是否是手机连接
    if(check_phone==0x44 && RBuff_date[0]==0x44 && RBuff_date[1]==0x31 &&
RBuff_date[2]==0x0a)
    {
        flag_phone=1;
        pattern=1;
//    print(0x03, "A");
        beed_ctrl(1500, 1);
        speed_A=100;

    }
    //PC 控制
    if(RBuff_date[0]==0x3C && RBuff_date[1]==0x31 && RBuff_date[2]==0x3E)
    {
        pattern=2;
        print(0x4f, "2");
        beed_ctrl(500, 4);
    }

    if(RBuff_date[0]==0x3C && RBuff_date[1]==0x30 && RBuff_date[2]==0x3E)
    {
        pattern=1;
        print(0x4f, "1");
        beed_ctrl(1500, 1);
    }

    if(RBuff_date[0]==0x7B && RBuff_date[1]==0x31 && RBuff_date[2]==0x7D)
    {
        pattern=1;
        print(0x4f, "1");
        print(0xe3, " ");
        beed_ctrl(1500, 1);
    }
    else if(RBuff_date[0]==0x7B && RBuff_date[1]==0x30 &&
RBuff_date[2]==0x7D)
    {
        pattern=0;
    }
}

```

```

        print(0x4f, "0");
        init_system();
        beed_ctrl(500, 2);
    }
    else if( RBuff_date[0]==0x50 && RBuff_date[2]==0x00 )
    {
        speed_A=RBuff_date[1];
        set_RS=Mark_Speed(pre_RS);
        set_LS=Mark_Speed(pre_LS);
        motor( set_LS, set_RS);
    }
    else
    {
        R_date[0]=RBuff_date[0];
        R_date[1]=RBuff_date[1];
        R_date[2]=RBuff_date[2];
    }
    }
    else
    {
        RBuff_date[0]= RBuff_date[1];
        RBuff_date[1]= RBuff_date[2];
        RBuff_date[2]= RBuff_date[3];
        RBuff_date[3]=SBUF;

        R_date[0]=RBuff_date[0];
        R_date[1]=RBuff_date[1];
        R_date[2]=RBuff_date[2];
    }
}
else
TI=0;
}
/*****interrupt T1*****/
void timer1(void) interrupt 3
{
    TH1=0xff;
    TL1=0xdc; //49:100us a5:50us dc:20us ee:10us
    //add your code here.
    PWM_cnt++;
}

```

```

if(PWM_cnt >=PWM_G1 )
{
    SPWM1=0;
}
if(PWM_cnt >=PWM_G2 )
{
    SPWM2=0;
}

if(PWM_cnt >= 500)
{
    PWM_cnt=0;
    SPWM1=1;
    SPWM2=1;
}

cnt_b++;
if(flag_beed==1)
{
    if(B_counter>0)
    {
        if(cnt_b>cnt_beed)
        {
            beed=!beed;
            cnt_b=0;
            B_counter--;
        }
    }
}

/*
if(flag_ser==1)
{
    cnt_ser++;
    if(cnt_ser>100)
    {
        EN1_S=0;
        flag_ser=0;
        cnt_ser=0;
    }
}

```

```

*/

}

/*****init T1*****/
void init_T1(void)
{
    TMOD|=0x10;
    TH1=0xff;
    TL1=0x49;
    TR1=1;
    ET1=1;
}

/*****initServo*****/
void initServo()
{
    PWM_G1=Byte_Read(0x00);
    PWM_G2=Byte_Read(0x01);
}

/*****setServo*****/
void setServo()
{
    Sector_Erase(0x00);
    Byte_Program(0x00,PWM_G1) ;
    Byte_Program(0x01,PWM_G2) ;
}

main(void)
{
    LED1=1;    //高电平 LED 不亮
    LED2=1;    //

    P4SW=0xff;    //定义 ALE PSEN 。 。三个引脚为 P4.4 P4.5 P4.6 用于控制 1602
    显示屏

    P1M0=0xC0;
    P1M1=0x00;

    init_T0();    //初始化定时器。

```



```
init_T1();

CLK_DIV=0x01; //系统分频工作，速度太快，1602 控制不了。
LCM2402_Init(); // 初如化 1602 显示屏

initUart();
initServo();

init_system();

EA=1;

beed_ctrl(800, 3);

print(0x00, "WT"); //在第一行第一位处从左向右打印"WonderTechnology" 字符串
print(0x40, "RX:");
print(0x0B, "S:");

print(0x4f, "0");

while(1)
{
    if(KEY1==0) //按下第一按键发送 "0x55 "
    {
        while(!KEY1);
        SCI_TX(0x55);
        ser_up();
        PWM_G1++;
    }

    if(KEY2==0) //按下第二按键发送 "0xf0 "
    {
        while(!KEY2);
        SCI_TX(0xf0);
        ser_down();
        PWM_G1--;
    }
}

display_1602_10(0x05, PWM_G1);
```

```

display_1602_10(0x0D, speed_A);

display_1602(0x43, RBuff_date[0]);
display_1602(0x46, RBuff_date[1]);
display_1602(0x49, RBuff_date[2]);

display_1602(0x4c, sen_inp_IROA());

if(pattern==1)
{
switch(R_date[0])
{
case 0x44:
switch(R_date[1])
{
case 0x4c:
switch(R_date[2])
{
case 0x31: motor(0, speed_A); break;
case 0x32: motor(-speed_A, speed_A); break;
case 0x33: motor(0, -speed_A); break;
default:
break;
}
break;

case 0x44:
switch(R_date[2])
{
case 0x31: motor(speed_A, speed_A); break;
case 0x32: motor(0, 0); break;
case 0x33: motor(-speed_A, -speed_A); break;
default:
break;
}
break;

case 0x52:
switch(R_date[2])
{
case 0x31: motor(speed_A, 0); break;
case 0x32: motor(speed_A, -speed_A); break;
case 0x33: motor(-speed_A, 0); break;
}
}
}
}

```

```

        default:
            break;
    }
    break;

default:
    break;
}
break;

case 0x46:
    switch(R_date[1])
    {
        case 0x4c:
            switch(R_date[2])
            {
                case 0x31:LED_F_on(); Light_cam=0;    break;//控制前车灯
                case 0x32:LED_F_off();Light_cam=1;    break;
                case 0x33:Light_B=0;    break;          //控制车前灯
                case 0x34:Light_B=1;    break;
                default:
                    break;
            }
            break;

        case 0x42:    //控制蜂鸣器
            switch(R_date[2])
            {
                case 0x31: beed=0; break;
                case 0x30: beed=1; break;
                default:
                    break;
            }
            break;

        case 0x53:    //控制激光
            switch(R_date[2])
            {
                case 0x31: Laser_on(); break;
                case 0x30: Laser_off(); break;
                default:
                    break;
            }
            break;
    }

```

```
        case 0x50:    //控制水泵
            switch(R_date[2])
            {
                case 0x31: break;
                case 0x30: break;
                default:
                    break;
            }
            break;

        default:
            break;
    }
    break;

    default:
        break;
}
}
else if(pattern==2)
{
//红外避障传感器
    switch( sen_inp_IROA() )
    {
        case 0x01:
        case 0x03: motor(-100,100);
                beed_ctrl(100,2);
                delay(20);
                break;

// case 0x02: motor(-100,-100); delay(20); motor(100,0); break;

        case 0x04:
        case 0x06: motor(100,-100);
                beed_ctrl(100,2);
                delay(20);
                break;

        case 0x07: motor(-100,-100);
                beed_ctrl(100,3);
                delay(100);
```

```

        motor(-100, 100);
        delay(150);
        motor(100, 100);
        break;
    default:
        motor(100, 100);
        break;
    }
}
else
{
;
}

if(pattern!=0)
{
switch(R_date[0])
{
    case 0x53:
        switch(R_date[1])
        {
            case 0x53:
                switch(R_date[2])
                {
                    case 0x31: cam_up(); R_date[2]=0; break;
                    case 0x32: cam_down(); R_date[2]=0; break;
                    case 0x33: cam_left(); R_date[2]=0; break;
                    case 0x34: cam_right(); R_date[2]=0; break;
                    //增加四个边动方向
                    case 0x35: cam_up(); cam_left(); R_date[2]=0; break;
                    case 0x36: cam_up(); cam_right(); R_date[2]=0; break;
                    case 0x37: cam_down(); cam_left(); R_date[2]=0; break;
                    case 0x38: cam_down(); cam_right(); R_date[2]=0; break;
                }
            default:
                break;
        }
        break;
    case 0x4d:
        if(R_date[2]==0x30)
        {
            cam_mid();
            R_date[2]=0;
        }
        if(R_date[2]==0x31)

```

```
        {
            setServo();
            beed_ctrl(500, 4);
            R_date[2]=0;
        }
        break;

        default:

            break;
    }
    break;
default:
    break;
}
}
}
```

附件 B：智能车控制板原理图

