

天津大学仁爱学院

毕业设计（论文）任务书

题目：语音控制机械臂自动取物车设计

系 名 信息工程系

专 业 电子信息工程

年 级 12 级

姓 名 谢博翔

学 号 6012202093

指导教师 吉建华

职 称 副教授

2015 年 12 月 8 日

一、原始依据（包括设计或论文的工作基础、研究条件、应用环境、工作目的等。）

1、工作基础：在掌握一定的控制器处理器的基础下，设计语音控制的机械臂自动取物车，并能够在此基础上进行扩展，实现红外检测、报时、显示等功能。

2、研究条件：Windows 操作系统的 PC 机，KEIL 软件，相关外设电路。

3、开发工具：程序调试软件 KEIL、电路图绘制软件 Altium Designer，PROTUES 仿真器。

4、工作目的：熟悉 STC12C5A60S2 芯片工作环境，了解一些外设电路，并能设计出语音控制的机械臂自动取物车，通过相应简单的软件进行测试。

二、参考文献

- [1] 李朝青. 单片机原理与接口技术[M]. 北京：北京航空航天大学出版社，1997.
- [2] 杨继生, 刘芬. 霍尔传感器A44E在车轮测速中的应用研究[J]. 电子测量技术, 2009, (10).
- [3] 陈永普. 新编555集成电路应用800例[M]. 电子工业出版社，2001, 1.
- [4] 王倩, 许永生. 出租车计价器的硬件设计[J]. 计算机工程应用技术, 2010,02.
- [5] 谭浩强. C程序设计（第二版）[M]. 清华大学出版社，2003.
- [6] 徐良雄. 出租车计价器的设计与分析[J]. 仪器仪表用户, 2012,19(3).
- [7] 肖卫初, 陈伟宏, 杨明. 基于单片机的多功能出租车计价器的设计[J]. 新型元器件与应用, 2008,16.
- [8] 袁益林, 韩云东等. 利用89S51单片机的出租车计价器设计[J]. 汇编语言, 2011,9（002）.
- [9] 陈渭红. 多功能的出租车计价器的设计与实现[J]. 高校理科研究, 2011,22.
- [10] 王素芹. 基于单片机的出租车计价器的设计[J]. 设计应用, 2011,14.
- [11] 刘尧, 白福忠, 兰月政. 基于AT89S51单片机出租车计费器的研究[J]. 仪表与传感器, 2013,32(11).
- [12] Jose Villar, Adoifo Otero, Jose Otero, Luciano Sanchez. Taximeter verification using imprecise data from GPS[J]. Engineering Applications of Artificial Intelligence, 2008.

三、设计（研究）内容和要求（包括设计或研究内容、主要指标与技术参数，并根据课题性质对学生提出具体要求。）

1、研究内容及实现目标：

语音控制的机械臂自动取物车，能在很大程度上解决人力投入问题，主要由循迹车体、机械臂、语音识别部分和单片机控制部分组成。本产品能在管理员发出语音命令后，通过合泰单片机接受指令，小车会驶向相应的位置，通过调动机械臂取放相应的物品并回到起点。由此只需要很少的管理人员和维护人员，就可实现长期的低投入高效率的货物运输过程，为企业节约人力资金投入，让企业更快发展。

由于科学技术的发展和进步,工厂的运作正在快速地由手工操作向自动化和智能化方向过渡。语音循迹自动取物车模拟了实际自动化工厂中小车的搬运和机器人的自动分拣功能。此设计针对操作者的指令,对不同物品进行识取,进而让小车将所需物品按类别取回。

掌握 STC12C5A60S2 芯片工作原理,进一步扩展到相应外设,通过单片机的处理结合必要的外围电路设计。利用电路图/PCB 绘制软件画出相应的 PCB 板。在硬件实现的基础下,利用 KEIL 软件编写简单测试代码加以对系统测试。

2、主要指标及技术参数:

- (1) 单片机准确接受语音指令并处理来控制小车和机械臂运做。
- (2) 利用红外对管检测黑线与障碍物,并以单片机为控制芯片来控制小车的速度及转向,从而实现自动循迹并到达指定位置的功能。其中小车驱动 L298N 驱动电路完成,速度由单片机输出的 PWM 波控制。
- (3) 通过单片机控制完成机械臂安全平稳地取放物品。
- (4) 显示取放物品的名称和次数,并且显示检测室内温湿度。
- (5) 通过颜色识别模块,识取出所需要的物品。

指导教师(签字)

年 月 日

审题小组组长(签字)

年 月 日

天津大学仁爱学院本科生毕业设计（论文）开题报告

课题名称	语音控制机械臂自动取物车设计		
系 名	信息工程系	专 业	电子信息工程
学生姓名	谢博翔	指导教师	吉建华

一、课题来源及意义

随着工业自动化的普及和发展，控制器的需求量逐年增大，其主要在汽车、电子、机械加工、食品、医药等领域的生产流水线或货物装卸调运。而机械手技术的运用可以更好地节约能源和提高运输设备或产品的效率，满足现代经济发展的要求。

如今制造业发展迅速，需要投入大量的人力资源来完成工作，其中很大一部分需要人力去传送原料和物料等资源，为此，声控技术和机械臂的结合无疑为企业带来便利。

二、国内外发展现状

目前，国内现有的产品只能做到循迹定点取物，但是真实情况下物品和物品存放的位置不会是一尘不变的。与之单一的定点循迹取物相比，本产品能做到根据语音的内容取回指定的物品，物品可以存放在地图内的任意位置。物品选择的变化只需要在语音控制端和主控单片机单方面增加控制选项，便能在地图中增加寻找目标选项，内容升级十分方便。有着较高的推广价值。

三、研究目标

本课题旨在设计出一个语音控制的机械臂自动取物车，当在管理员发出对应语音指令后，小车会在任意地图内寻找指令中对应的物体并通过机械臂夹取的。该系统包括 5 个部分：语音识别部分（LD3320 语音芯片由 STC11L08XE 控制。），红外传感循迹部分，机械臂控制部分，颜色识别部分（除语音识别部分外均由 STC12C5A60S2 单片机及其各自外围电路组成）。

该系统成功实现了语音控制小车识别、取物、返回等功能。极大的提高了物料的传送效率，提高了企业的生产力并减少了长期的人力财力投入。

四、研究内容

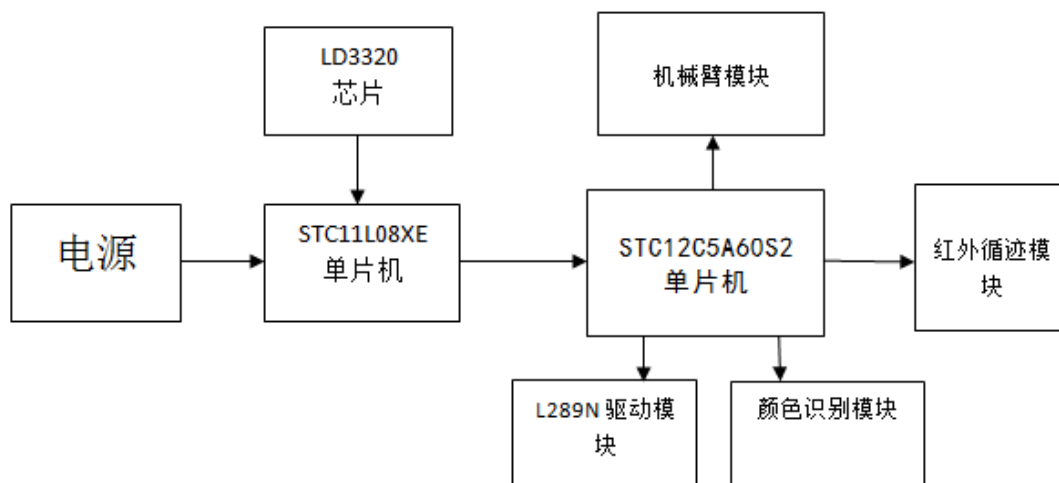
（1）收集该课程设计所用到的资料及文献

- (2) 初步了解和研究该课程设计所用到的元器件及功能介绍
- (3) 熟悉 Altium Designer 开发环境，绘制原理图
- (4) 熟悉 KEIL C51 开发环境，编写源程序
- (5) 掌握语音识别模块的原理和应用
- (6) 实现机械臂简单的抓取动作
- (7) 实现小车自动循迹的功能

五、研究方法手段

(一)硬件电路模块

- 1、电源部分
- 2、语音识别模块
- 3、机械臂控制电路
- 4、L298N 电机驱动电路
- 5、红外循迹模块
- 颜色识别模块



为实现低成本，我们采用 LD3320 芯片的语音识别模块，我们需用 STC11L08XE 单片机通过设置 LD3320 芯片的寄存器，把诸如"你好"这样的识别关键词的内容动态地传入芯片中，芯片识别出这样设定的关键词语，再反馈给来控制小车和机械臂。

采用五只红外对管，一只置于轨道中间，四只置于轨道外侧，当小车脱离轨道时，等待外面任一只检测到黑线后，做出相应的转向调整，直到中间的光电开关重新检测到黑线（即回到轨道）再恢复正向行驶。在循迹小车的基础上，加装光电检测器，实现对电动车的速度、位置、运行状况的实时测量，并将测量数据

传送至合泰单片机进行处理，然后由单片机根据所检测的各种数据实现对小车的智能控制。这种方案能实现对循迹小车的运动状态进行实时控制，有控制灵活、可靠，精度高等特点，可满足对系统的各项要求。

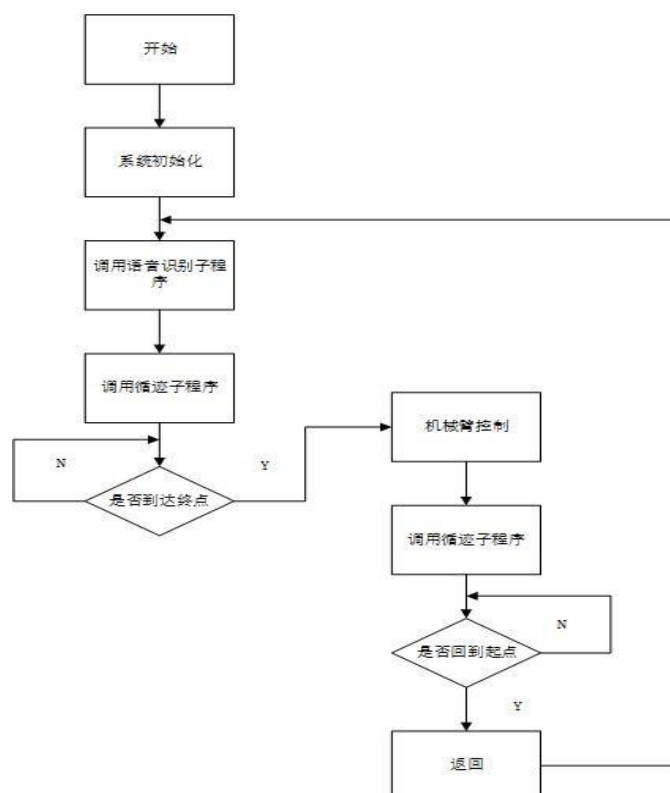
用 STC12C5A60S2 单片机输出四路 PWM 脉冲分别控制机械臂的四个舵机，实现具有四个自由度的机械臂。本次设计所使用的单片机性能较强大，完全可以满足设计需要。用单片机通过产生不同频率及占空比的 PWM 波来控制驱动电路调整电机的速率，实现对当前速度的调整。

通过 GY-31 颜色识别模块对物体颜色的识别，完成对不同颜色物体的识别，反馈给合泰单片机后为之后机械臂抓取提供抓取前条件。

以上各模块通过处理器相互协调最终实现小车的智能循迹。

(二)源程序的编写及调试

- 1.熟悉 KEIL C51 开发环境，编写源程序
- 2.查看 LD3320 的芯片手册，编写程序
- 3.学习舵机及电机的控制方法，实现机械臂和小车的控制
- 4.查看颜色识别模块的资料，实现颜色识别



六、进度安排

1. 2015.12.07-2016.03.15: 搜集毕业设计的相关资料, 阅读文献, 完成开题报告;
2. 2016.03.16-2016.03.30: 熟悉 STC12C5A60S2 芯片内部结构和外部引脚的功能, 内部硬件资源的具体使用方法, 熟悉 Cadence Allegro 开发环境;
3. 2016.03.30-2016.04.02: 设计电路, 焊接完成硬件基础;
4. 2016.04.02-2016.05.10: 软件编程设计, 程序调试, 实现实验目的;
5. 2016.05.10-2016.06.07: 撰写论文, 准备答辩。

七、方案可行性分析

1. 语音控制的智能取物小车在近几年竞赛中获得了广泛的关注, 有许多专家学者正对此进行着深入的研究和探讨, 并且已经得出了许多有意义的结论, 有比较丰富的成果可以借鉴, 研究智能循迹小车也具有重要的现实意义。

2. 学校图书馆有丰富的文献资料可供参考, 还有完善的电子信息平台提供技术支持。

八、主要参考文献

- [1] 谭浩强. C 程序设计[M]. 北京: 清华大学出版社, 2003.
- [2] 张伟等. Protel DXP 高级应用[M]. 北京: 人民邮电出版社, 2004.
- [3] 侯明, 付兴建, 吴迎年等. 基于 LabVIEW VISA 智能车无线调试系统[J]. 国外电子测量技术, 2012, (1): 67-69, 88.
- [4] 葛亚明, 刘涛, 王宗义. 视频同步分离芯片 LM1881 及其应用[J]. 应用科技, 2007, 31(9): 6-7.
- [5] 黄开胜, 陈宋. 汽车理论与智能模型车机械结构调整方法[D]. 清华大学汽车安全与节能国家重点实验室, 2006.
- [6] 谢文策. 线阵 CCD 图像不失真采集算法及实现[J]. 信息通信, 2009, 9(4): 4-6.
- [7] 张军. AVR 单片机应用系统开发典型实例[M]. 北京: 中国电力出版社, 2005.
- [8] 朱广, 周程. CCD 图像传感器在路径识别系统中的应用[J]. 工业控制计算机. 2009, (3): 50-51.
- [9] 黄开胜, 金华民, 蒋狄南著. 韩国智能模型车技术方案分析[M]. 北京: 北京航空航天大学

学出版社, 2007.45~51.

[10] 吴振宇, 赵亮, 冯林等. 基于分数阶 PID 控制器的智能车控制[J]. 控制工程, 2011, 18(3): 401-404.

[11] 阮毅, 陈伯时. 电力拖动自动控制系统[M]. 北京: 机械工业出版社, 2010.

[12] 李晨, 宓超. 基于飞思卡尔单片机 MC9S12XS128 的智能车设计[J]. 上海海事大学学报, 2012, 33(1): 82-84.

[13] Julio E. Normey-Rico. Mobile robot path tracking using a robust PID controller [J], Control Engineering Practice, 2001, 9(11):1209-1214.

[14] Freescale Semiconductor Inc.S12PWM8B8CV1/D Block User Guide V01.17[Z].2004.1-51.

[15] Julio E,Normey-Rico.Mobile robot path tracking using a robust PID controller [J], Control Engineering Practice,2001,9(11):1209-1214.

选题是否合适: 是 ☐ 否 ☐

课题能否实现: 能 ☐ 不能 ☐

指导教师 (签字)

年 月 日

选题是否合适: 是 ☐ 否 ☐

课题能否实现: 能 ☐ 不能 ☐

审题小组组长 (签字)

年 月 日

天津大学仁爱学院

毕业设计（论文）说明书



题目：语音控制机械臂自动取物车设计

系 名 信息工程系

专 业 电子信息工程

学 号 6012202093

学生姓名 谢博翔

指导教师 吉建华

2016 年 6 月 7 日

摘 要

“语音控制的机械臂自动取物车”，能在很大程度上解决人力投入问题，主要由单片机控制部分、颜色识别部分、循迹车体、机械臂和语音识别部分组成。本产品能在管理员发出语音命令后，通过单片机接受指令，小车会驶向相应的位置，通过调动机械臂取放相应的物品并回到起点。由此只需要很少的管理人员和维护人员，就可实现长期的低投入高效率的货物运输过程，为企业节约人力资金投入，让企业更快发展。

该系统包括五部分，语音识别部分（LD3320 语音芯片由 STC11L08XE 控制。），红外传感循迹部分，机械臂控制部分，颜色识别部分（除语音识别部分外均由 STC12C5A60S2 单片机及其各自外围电路组成）五部分组成。该系统成功实现了语音控制小车识别、取物、返回的功能，在管理员发出对应语音指令后，小车会在任意地图内寻找指令中对应的物体，并返回。极大的提高了物料的传送效率，提高了企业的生产力并减少了长期的人力财力投入。

关键词： 语音识别；颜色识别；机械臂

ABSTRACT

"Voice control of the manipulator to automatically extract car", in great extent solve the manpower problem, consists essentially of by the MCU control part, color recognition, tracking a vehicle body, a mechanical arm and speech recognition part. This product can be issued by the administrator voice commands, through the microcontroller to accept the instructions, the car will sail to the corresponding position, through the mobilization of the mechanical arm to take the corresponding items and return to the starting point. Thus, only a few managers and maintenance personnel, can achieve long-term low input and high efficiency of the cargo transport process, for the enterprise to save human capital investment, so that enterprises can develop faster.

The system consists of five parts, the speech recognition part (LD3320 voice chip is controlled by STC11L08XE). , infrared sensor tracking and mechanical arm control part, color recognition part (except the part of speech recognition are by STC12C5A60S2 microcontroller and its peripheral circuit) of five parts. The system successfully realized the voice control car identification, fetch, return function, the administrator issued the corresponding voice commands, the car will be in any map to find the corresponding objects in the map, and return. Greatly improve the transmission efficiency of the material, improve the productivity of enterprises and reduce the long-term investment in human and financial resources.

Key words: Speech recognition; Color recognition; Manipulator arm

目 录

第一章	绪论	1
1.1	研究背景和动机	1
1.2	国内外发展现状	1
第二章	系统总体设计	2
2.1	系统总体设计思路和框图	2
2.2	硬件设计思路	3
2.3	软件设计思路	3
第三章	系统的硬件设计	5
3.1	STC12C5A60S2 最小系统	5
3.2	系统稳压电源模块	7
3.3	语音识别模块	9
3.4	小车循迹原理与设计	12
3.5	舵机模块工作原理	15
3.6	颜色识别模块工作原理	16
第四章	系统的软件设计与实现	19
4.1	KEIL C51 开发环境的使用	19
4.2	软件系统总体设计	21
第五章	系统调试与总结	27
5.1	小车调试方法设计	27
5.2	测试环境和方法设计	27

5.3 工作总结	27
第六章 结束语	29
参考文献	30
附录	31
外文资料	
中文翻译	
致 谢	

第一章 绪论

本章主要介绍研究背景及国内外的发展现状。

1.1 研究背景和动机

随着工业自动化的普及和发展，控制器的需求量逐年增大，其主要在汽车、电子、机械加工、食品、医药等领域的生产流水线或货物装卸调运。而机械手技术的运用可以更好地节约能源和提高运输设备或产品的效率，满足现代经济发展的要求。如今制造业发展迅速，需要投入大量的人力资源来完成工作，其中很大一部分需要人力去传送原料和物料等资源，为此，声控技术和机械臂的结合无疑为企业带来便利。

“语音控制的机械臂自动取物车”，能在很大程度上解决人力投入问题，主要由合泰单片机控制部分、颜色识别部分、循迹车体、机械臂和语音识别部分组成。本产品能在管理员发出语音命令后，通过单片机接受指令，小车会驶向相应的位置，通过调动机械臂取放相应的物品并回到起点。由此只需要很少的管理人员和维护人员，就可实现长期的低投入高效率的货物运输过程，为企业节约人力资金投入，让企业更快发展。

1.2 国内外发展现状

汽车在走过的 100 多年的历史中，从没停止过智能化的步伐，进入 20 世纪 90 年代以来，随着汽车市场竞争激烈程度的日益加剧和智能运输系统（ITS）的兴起，国际上对于智能汽车及其相关技术的研究越拉越重视，一大批有实力有远见的大公司、大学和研究机构开展了这方面的研究。美国、日本和欧洲的很多国家都十分重视并积极发展智能车系统，并进行了相关实验，取得了许多成就。

目前，国内现有的产品只能做到循迹定点取物，但是真实情况下物品和物品存放的位置不会是一尘不变的。与之单一的定点循迹取物相比，本产品能做到根据语音的内容取回指定的物品，物品可以存放在地图内的任意位置。物品选择的变化只需要在语音控制端和主控单片机单方面增加控制选项，便能在地图中增加寻找目标选项，内容升级十分方便。有着较高的推广价值。

第二章 系统总体设计

本章就智能小车的总体概述及各部分的布局进行了简要的介绍。

2.1 系统总体设计思路和框图

为实现低成本,我们采用 LD3320 芯片的语音识别模块,我们需用 STC11L08XE 单片机通过设置 LD3320 芯片的寄存器,把诸如“你好”这样的识别关键词的内容动态地传入芯片中,芯片识别出这样设定的关键词语,再反馈给来控制小车和机械臂。

采用五只红外对管,一只置于轨道中间,四只置于轨道外侧,当小车脱离轨道时,等待外面任一检测到黑线后,做出相应的转向调整,直到中间的光电开关重新检测到黑线(即回到轨道)再恢复正向行驶。在循迹小车的基础上,加装光电检测器,实现对电动车的速度、位置、运行状况的实时测量,并将测量数据传送至合泰单片机进行处理,然后由单片机根据所检测的各种数据实现对小车的智能控制。这种方案能实现对循迹小车的运动状态进行实时控制,有控制灵活、可靠,精度高等特点,可满足对系统的各项要求。

用 STC12C5A60S2 单片机输出四路 PWM 脉冲分别控制机械臂的四个舵机,实现具有四个自由度的机械臂。本次设计所使用的单片机性能较强大,完全可以满足设计需要。用单片机通过产生不同频率及占空比的 PWM 波来控制驱动电路调整电机的速率,实现对当前速度的调整。

通过颜色传感器 TCS3200 对物体颜色的识别,完成对不同颜色物体的识别,反馈给合泰单片机后为之后机械臂抓取提供抓取前条件。

以上各模块通过处理器相互协调最终实现小车的智能循迹。如图 2-1 所示。

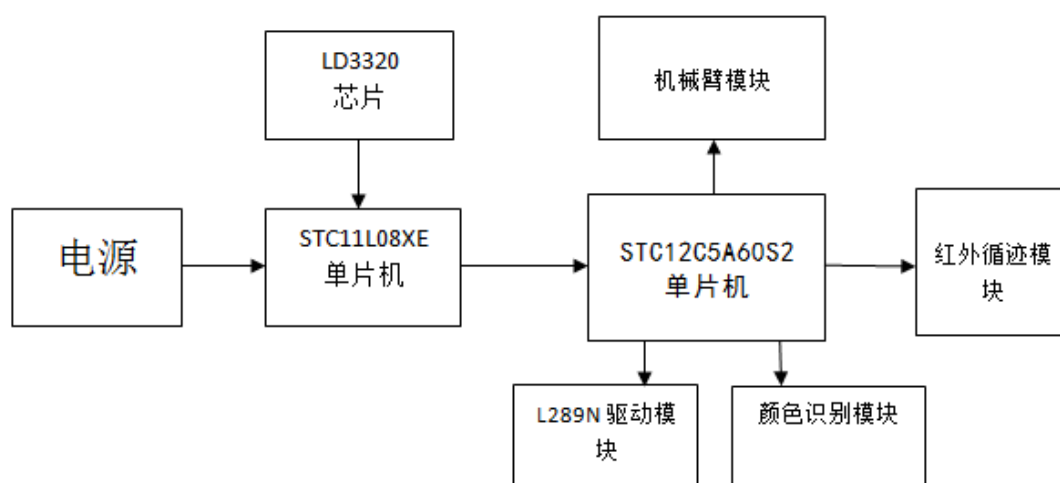


图2-1 系统总体结构

各模块的作用如下：

（1）STC12C5A60S2 单片机最小系统：作为整个智能车的“大脑”，采集并处理红外循迹模块、颜色识别模块输入的信号，根据控制算法做出控制决策，驱动直流电机和机械臂上面的舵机完成智能车相应的动作。

（2）语音识别模块：语音模块用 STC1108XE 通过设置 LD3320 芯片的寄存器，把识别关键词的内容动态地传入芯片中，芯片识别出设定的关键词语，再反馈给合泰单片机。

（3）红外循迹模块：为整个系统提供合适而又稳定的电源。

（4）机械臂控制模块：驱动直流电机和舵机完成智能车的加减速控制和转向控制。

（5）L298N 驱动模块：检测反馈智能车后轮的转速，用于速度的闭环控制。

（6）颜色识别模块：主要用于智能车系统的功能调试、智能车状态监控等方面。

（7）电源模块：给电路供电。

2.2 硬件设计思路

系统单片机程序设计主要包括：STC11L08XE 单片机控制的语音识别模块；STC12C5A60S2 单片机控制的红外循迹模块、控制机械臂模块、L298N 驱动之流电机电路和最后颜色识别模块。

当语音识别模块接收到语音指令后，通过 STC11L08XE 单片机将所需的物品的特征传递个 STC12C5A60S2 单片机，同时在 STC12C5A60S2 单片机控制下，小车通过循迹，寻找到小车通过的路径，小车电机又 L298N 模块来驱动电机，最后到达物品前，机械臂调整姿势，并通过粘贴在机械臂上的颜色识别模块，感知，是否为所需的物品，确定后，机械臂抓取物品，并返回起始点。

2.3 软件设计思路

本系统单片机程序设计主要包括：STC11L08XE 单片机控制的语音识别程序；STC12C5A60S2 单片机控制的红外循迹程序；控制机械臂程序；L298N 驱动之流电机的程序和最后颜色识别的程序。

主程序是对整个系统框架的描述。该系统的主程序作用就是上电后，完成对系统的初始化，而系统初始化包括：单片机初始化和各个模块的初始化。其单片机的初始化主要是对串口和定时器寄存器进行设置，模块的初始化是通过单片机发送 AT 指令对各个模块进行设置。

STC11L08XE 单片机对语音识别模块的控制也是通过 AT 指令完成的，通过设置 LD3320 芯片的寄存器，识别关键词的内容动态地传入芯片中，再反馈给单片机，并由语音识别模块对指令响应值作出相应的反应。对语音识别模块指令的接收方式，单片机采用 I/O 口检测方式，当指定的 I/O 口电平发生变化时，便判

断是否为新消息提示，从而进行下一步动作。

对于红外循迹和 L298N 电机驱动部分，通过红外传感器检测黑线传给单片机，经过处理，然后由单片机根据所检测的各种数据分析处理后，通过对 L298N 驱动小车各个电机，实现对小车的智能控制。

对于机械臂抓取物品和颜色识别部分，当颜色识别模块检测的颜色与索取物品一致时，给予单片机一个工作信号，机械臂接受指令，通过调整姿态，抓取物品，等待下一条指令。

单片机系统全部程序使用 C 语言作为系统开发设计语言，C 语言是一种计算机程序设计语言。它既具有高级语言的特点，又具有汇编语言的特点，它可以作为工作系统设计语言，编写系统应用程序，也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。具有简洁紧凑、灵活方便、便于移植等优点。

第三章 系统的硬件设计

本章主要介绍组成智能车硬件模块的电路设计及各模块的基本原理。

3.1 STC12C5A60S2 最小系统

STC12C5A60S2 最小系统为本系统的主控部分。

3.1.1 STC12C5A60S2 芯片介绍

STC12C5A60S2 芯片介绍如下。

3.1.1.1 STC12C5A60S2 芯片简介

- 1、增强型 8051CPU，1T（1024G），单时钟/机器周期
- 2、工作电压 5.5-3.5V
- 3、1280 字节 RAM
- 4、通用 I/O 口，复位后为：准双向口/弱上拉

可设置成四种模式：准双向口/弱上拉，强推挽/强上拉，仅为输入/高阻，开漏
每个 I/O 口驱动能力均可达到 20mA，但整个芯片最大不要超过 120mA

- 5、有 EEPROM 功能
- 6、看门狗
- 7、内部集成 MAX810 专用复位电路
- 8、外部掉电检测电路
- 9、时钟源：外部高精度晶体/时钟，内部 R/C 振荡器

常温下内部 R/C 振荡器频率为：5.0V 单片机为：11~17MHz 3.3V 单片机为：
8~12MHz

10、4 个 16 位定时器两个与传统 8051 兼容的定时器/计数器，16 位定时器
T0 和 T1

11、3 个时钟输出口，可由 T0 的溢出在 P3.4/T0 输出时钟，可由 T1 的溢出
在 P3.5/T1 输出时钟，独立波特率发生器可以在 P1.0 口输出时钟

12、外部中断 I/O 口 7 路，传统的下降沿中断或电平触发中断，并新增支持
上升沿中断的 PCA 模块，Power Down 模式可由外部中断唤醒，INT0/P3.2，
INT1/P3.3，T0/P3.4，T1/P3.5，Rx/D/P3.0，CCP0/P1.3，CCP1/P1.3

13、PWM2 路

14、A/D 转换，10 位精度 ADC，共 8 路，转换速度可达 250K/S

15、通用全双工异步串行口（UART）

16、双串口，Rx/D2/P1.2，Tx/D2/P1.3

17、工作范围：-40~85

18、封装：LQFP-48，LQFP-44，PDIP-40，PLCC

3.1.1.2 STC12C5A60S2 管脚说明

P0.0~P0.7 P0: P0 口既可以作为输入/输出口,也可以作为地址/数据复用总线使用。当 P0 口作为输入/输出口时, P0 是一个 8 位准双向口, 内部有弱上拉电阻, 无需外接上拉电阻。当 P0 作为地址/数据复用总线使用时, 是 8 位地址线 A0~A7, 数据线 D0~D7

P1.0/ADC0/CLKOUT2 标准 IO 口、ADC 输入通道 0、独立波特率发生器的时钟输出

P1.1/ADC1、P1.2/ADC2/ECI/RxD2 标准 IO 口、ADC 输入通道 2、PCA 计数器的外部脉冲输入脚, 第二串口数据接收端

P1.3/ADC3/CCP0/TxD2 外部信号捕获, 高速脉冲输出及脉宽调制输出、第二串口数据发送端

P1.4/ADC4/CCP1/SS 非 SPI 同步串行接口的从机选择信号

P1.5/ADC5/MOSI SPI 同步串行接口的主出从入(主器件的输入和从器件的输出)

P1.6/ADC7/SCLK SPI 同步串行接口的主入从出

P2.0~P2.7 P2 口内部有上拉电阻, 既可作为输入输出口 (8 位准双向口), 也可作为高 8 位地址总线使用。

P3.0/RxD 标准 IO 口、串口 1 数据接收端

P3.1/INT0 非 外部中断 0, 下降沿中断或低电平中断

P3.3/INT1 、P3.4/T0/INT 非/CLKOUT0 定时器计数器 0 外部输入、定时器 0 下降沿中断、定时计数器 0 的时钟输出

A/D 转换器的结构, STC12C5A60AD/S2 系列带 A/D 转换的单片机的 A/D 转换口在 P1 口, 有 8 路 10 位高速 A/D 转换器, 速度可达到 250KHz (25 万次/秒)。8 路电压输入型 A/D, 可做温度检测、电池电压检测、按键扫描、频谱检测等。上电复位后 P1 口为弱上拉型 IO 口, 用户可以通过软件设置将 8 路中的任何一路设置为 A/D 转换, 不须作为 A/D 使用的口可继续作为 IO 口使用。

单片机 ADC 由多路开关、比较器、逐次比较寄存器、10 位 DAC、转换结果寄存器以及 ADC_CONTER 构成。

该单片机的 ADC 是逐次比较型 ADC。主次比较型 ADC 由一个比较器和 D/A 转换器构成, 通过逐次比较逻辑, 从最高位 (MSB) 开始, 顺序地对每一输入电压与内置 D/A 转换器输出进行比较, 经过多次比较, 使转换所得的数字量逐次逼近输入模拟量对应值。逐次比较型 A/D 转换器具有速度快, 功耗低等优点。需作为 AD 使用的口先将 P1ASF 特殊功能寄存器中的相应位置为 '1', 将相应的口设置为模拟功能

3.1.2 STC12C5A60S2 最小系统介绍

最小系统电路是整个系统的核心，它主要完成各种信号的处理、舵机电机的控制等工作。STC12C5A60S2 是 STC 生产的单时钟/机器周期（1T）的单片机，是高速、低功耗、超强抗干扰的新一代 8051 单片机，指令代码完全兼容传统 8051，但速度快 8-12 倍。内部集成 MAX810 专用复位电路，2 路 PWM，8 路高速 10 位 A/D 转换，针对电机控制，强干扰场合。^[1]

本文中选购 STC12C5A60S2 最小系统板。该最小系统板稳定可靠，所有功能引脚均已引出。包括各路 IO 接口、USB 转 TTL 电路、电源滤波电路、时钟电路、复位电路、电源指示灯、TX/RX 指示灯、程序下载接口和 P4 接口扩展（STC12 贴片芯片比直插 DIP 封装多了 P4.0—P4.7 IO 口）。最小系统板实物图如图 3-1 所示。

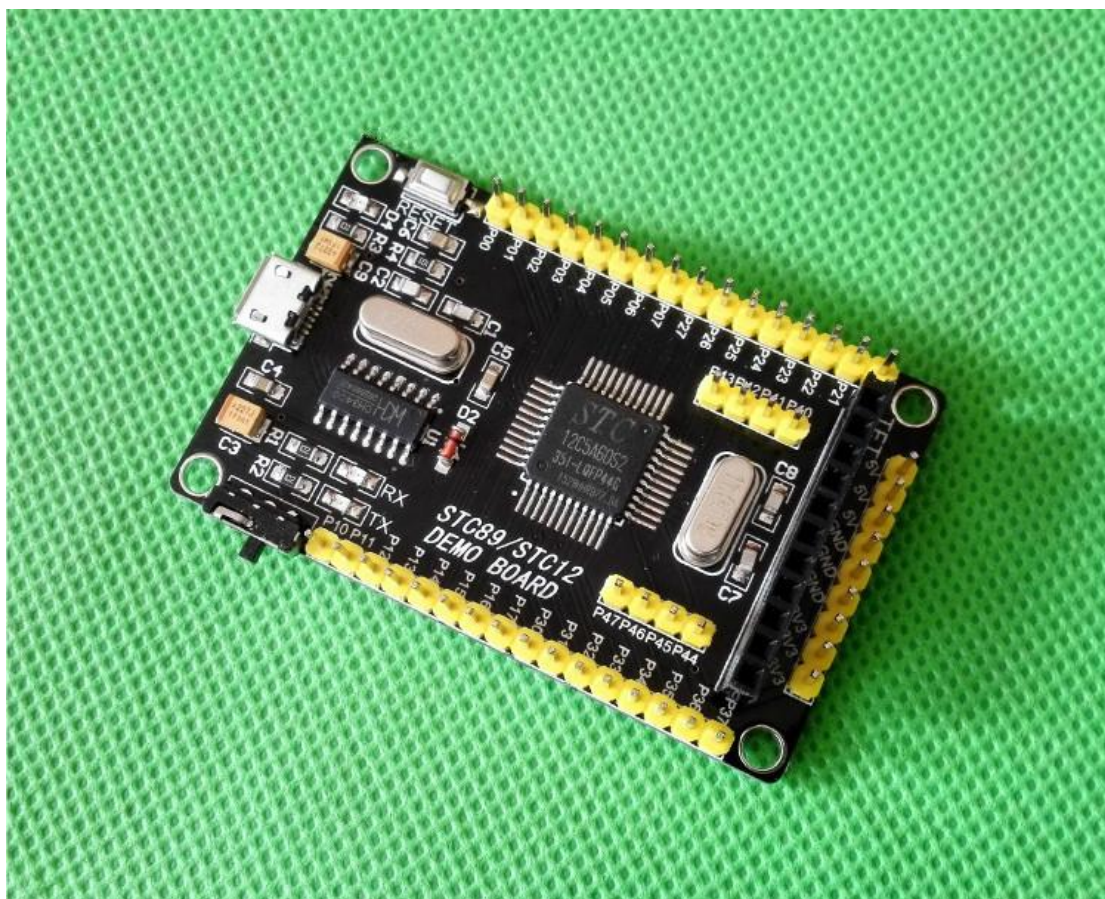


图3-1 最小系统板

3.2 系统稳压电源模块

电源是一个系统正常工作的基础，电源模块为系统其他各个模块提供所需要的能源保证，因此电源模块的设计至关重要。模型车系统中接受供电的部分包括：

传感器模块、单片机模块、电机驱动模块、舵机模块等。其硬件结构原理图如图 3-2 所示。

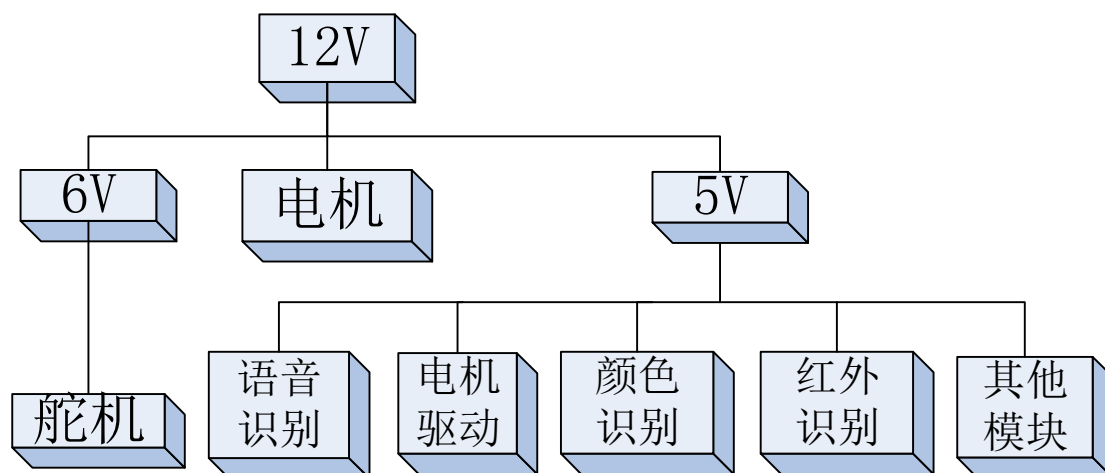


图 3-2 系统电源模块

3.2.1 5V 稳压电源设计

常用的电源有串联型线性稳压电源和开关型稳压电源两大类。前者具有纹波小、电路结构简单的优点，但是效率较低，功耗大；后者功耗小，效率高，但电路却比较复杂，电路的纹波大。

5V 主要用于最小系统、编码器和摄像头的供电。最小系统和摄像头对电源要求比较高，电源纹波必须要小。鉴于开关电源纹波比较大，而线性稳压电源纹波很小，所以本文选择使用线性电源。线性稳压芯片 LM7805 具有纹波小、电路结构简单的优点，所以选用 LM7805 单独对其进行供电，使得系统能够稳定地工作[2]。其典型应用电路图如图 3-3 所示。

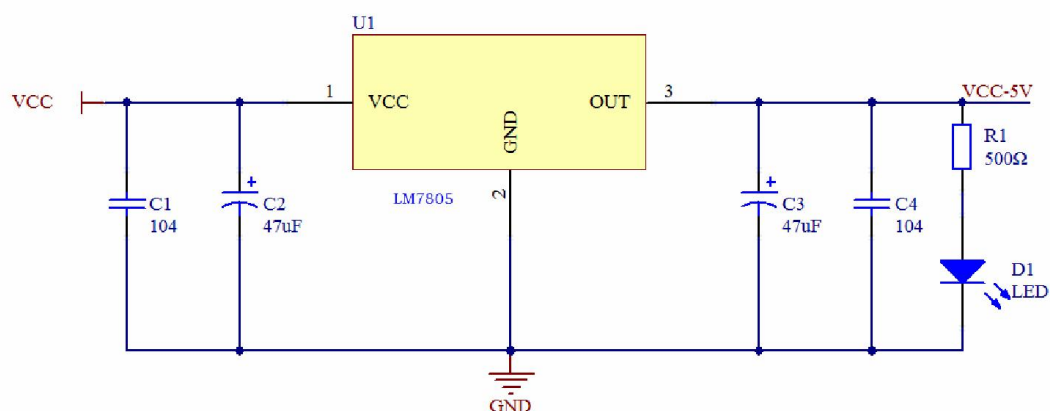


图 3-3 LM7805-5V 稳压模块

3.2.2 6V 稳压电源设计

6V 主要用于舵机的供电。舵机电源需要稳定电压供电，所以我们选择舵机的标准电压 6V 供电，稳压芯片选择 LM7806，电路结构简单并且完全满足课题的需要。其典型应用电路图如图 3-4 所示。

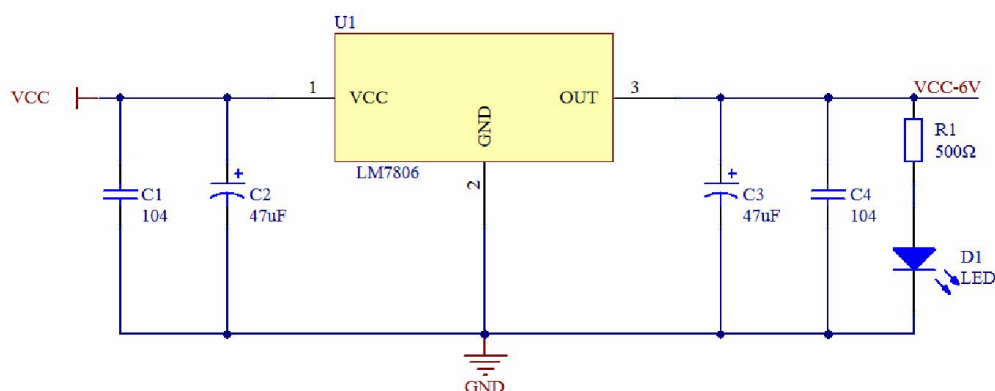


图 3-4 LM7806-6V 稳压模块

3.3 语音识别模块

3.3.1 特定人语音识别技术及原理

特定人语音识别(ASR, Auto Speech Recognition)技术是基于“关键词语列表”的识别技术，它是对大量的语音数据(相当于对数千人采集的数万小时的有效声音数据)经语言学家语音模型分析，建立数学模型，并经过反复训练提取基元语音的细节特征，以及提取各基元间的特征差异，得到在统计概率最优化意义上的各个基元语音特征，最后才由资深工程师将算法以及语音模型转换成硬件芯片并应用在嵌入式系统中。ASR 技术每次识别的过程就是把用户说出的语音内容，通过频谱转换为语音特征，再将这个转换后的语音特征和“关键词语列表”中的条目一一进行匹配，最优匹配的一条即作为识别结果。比如 ASR 技术在语音控制的手机应用中，这个“关键词语列表”的内容就是电话本中的人名、手机的菜单命令或手机存储卡中的歌曲名字。不论这个列表的条目内容是什么，只需要用户设置相关的寄存器，就可以把相应的待识别条目内容以字符形式传递给识别引擎。

由此可见，语音识别芯片完成的工作就是：把 MIC(麦克风)输入的声音进行频谱分析后提取语音特征，再和关键词语列表中的关键词语进行对比匹配，最后出得分最高的关键词语作为识别结果输出。

通常基于 ASR 技术的语音识别芯片能在两种情况下给出识别结果：

1、外部送入预定时间的语音数据（比如 5s 的语音数据），芯片对这些语音数据运算分析后，给出识别结果。

2、外部送入语音数据流，语音识别芯片通过端点检测(VAD, Voice Activity Detection) 技术检测出用户停止说话，把用户开始说话到停止说话之间的语音数据进行运算分析后，给出识别结果。

对于第一种情况，可以理解为设定了一个定时录音(如 5s 的语音数据)，芯片在 5s 后会停止把声音送入识别引擎，并且根据已送入引擎的语音数据计算出识别结果。

对于第二种情况，需要了解 VAD 的工作原理：VAD 技术是在一段语音数据流中，判断出哪个时间点是人声音的开始，哪个时间点是人声音的结束。判断的依据是，在背景声音的基础上有了语音发音，则视为声音的开始。而后，检测到一段持续时间的背景音(比如 600 ms)，则视为人声说话结束。通过 VAD 判断出人声说话的区域后，语音识别芯片会把这期间的声音数据进行识别处理，计算出识别结果。

除了以上两种情况外，语音识别算法无法“主动”地判断出是否识别出了一个结果。这是因为，在计算过程中的任何时刻，语音识别器都会对已送入识别芯片的声音数据进行分析，并根据匹配程度为识别列表中的关键词语进行打分，最匹配的打分最高。但是，由于识别算法不知道用户后面是否还继续说话，所以无法主动地判断已经识别出的结果。

3.3.2 语音识别芯片 LD3320 的工作原理

LD3320 提供了真正的单芯片语音识别解决方案。LD3320 集成了高精度的 A/D 和 D/A 接口，不再需要外接辅助的 Flash 和 RAM，即可以实现语音识别/声控/人机对话功能。并且，识别的关键词语列表是可以动态编辑的。基于 LD3320，可以在任何的电子产品中，甚至包括最简单的 51 作为主控芯片的系统中，轻松实现语音识别/声控/人机对话功能。为所有的电子产品增加 VUI (Voice User Interface) 语音用户操作界面。

主要特色功能：

非特定人语音识别技术：不需要用户进行录音训练可动态编辑的识别关键词语列表：只需要把识别的关键词语以字符串的形式传送进芯片，即可以在下次识别中立即生效。比如，用户在 51 等 MCU 的编程中，简单地通过设置芯片的寄存器，把诸如“你好”这样的识别关键词的内容动态地传入芯片中，芯片就可以识别这样设定的关键词语了。真正单芯片解决方案：不需要任何外接的辅助 Flash 和 RAM，真正降低系统成本。

内置高精度 A/D 和 D/A 通道：不需要外接 AD 芯片，只需要把麦克风接在芯片的 AD 引脚上；可以播放声音文件，并提供 550mW 的内置放大器。高准确度和实用的语音识别效果支持使用自由编辑 50 条关键词语在同一时刻，最多在 50 条关键词语中进行识别，终端用户可以根据场景需要，随时编辑和更新这 50

条关键词语的内容。

LD3320 语音识别芯片采用的就是 ASR 技术，图 3-5 就是由 LD3320 和单片机(或嵌入式系统)组成的语音识别系统原理框图。图中给出了 LD3320 的内部原理图结构，本文中选用的 MCU 是 STC11L08XE 单片机。如图 3-5 所示。

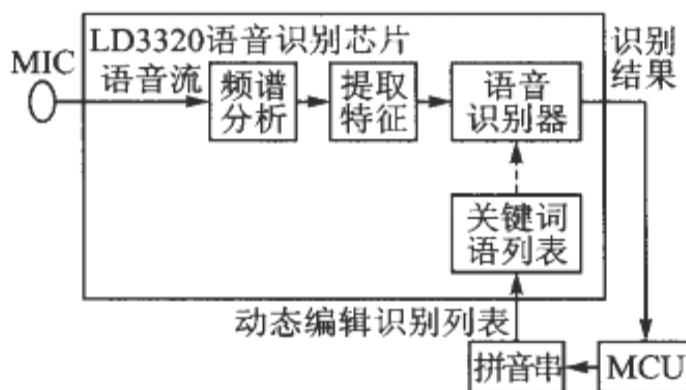


图 3-5 语音识别系统原理框图

语音识别芯片 LD3320 是 IC Route 公司的产品，它采用 ASR 技术，提供了一种脱离按键、键盘、鼠标、触摸屏等 GUI 操作方式且基于语音的用户界面 VUI(Voice UserInterface)使得用户对该系统的操作更简单、快速和自然。

用户只需要把识别的关键词语以字符串的形式传送进芯片，即可以在下次识别中立即生效。比如，用户在 51 等主控 MCU 的编程中，简单地通过设置芯片的寄存器，把诸如“你好”这样的识别关键词语的内容动态地传入芯片中，芯片就可以识别所设定的关键词语了。每个关键词语可以是单字、词组、短句或者任何的中文发音的组合。基于 LD3320 的语音识别系统可以随着使用流程，在运行时动态地更改关键词语列表的内容，这样可以用一个系统支持多种不同的场景，同时也不需要用户作任何的录音训练。

3.3.3 语音识别系统硬件设计

由 LD3320 组成的语音识别系统硬件有单片机及 LD3320。图 3-6 是由单片机 STC11L08XE 构成的主控芯片和由 LD3320A 构成的语音识别主系统。语音模块是由 STC11L08XE 单片机通过设置 LD3320 芯片的寄存器，把识别关键词的内容动态地传入芯片中，芯片识别出设定的关键词语，再反馈给单片机，如图 3-6。

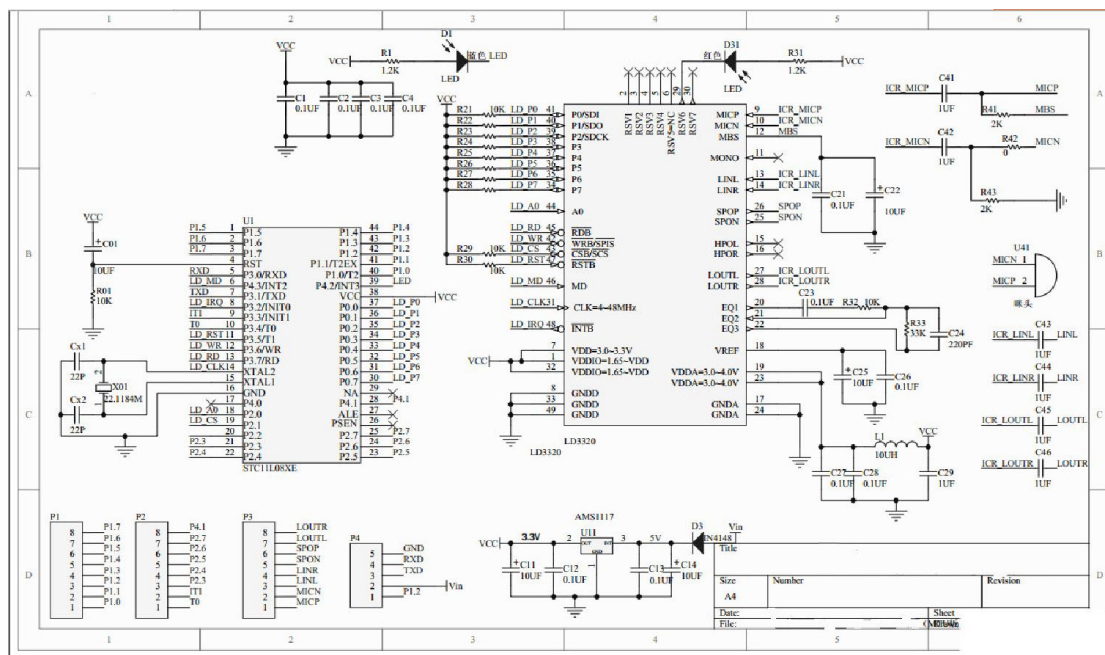


图 3-6 语音识别模块电路图

3.4 小车循迹原理与设计

3.4.1 红外循迹模块工作原理

红外循迹模块是为智能小车、机器人等自动化机械装置提供一种多用途的红外线探测系统的解决方案。该传感器模块对环境光线适应能力强，其具有一对红外线发射与接收管，发射管发射出一定频率的红外线，当检测方向遇到障碍物（反射面）时，红外线反射回来被接收管接收，经过比较器电路处理之后，同时信号输出接口输出数字信号（一个低电平信号），可通过电位器旋钮调节检测距离，有效距离范围 2~60cm，工作电压为 3.3V-5V。该传感器的探测距离可以通过电位器调节、具有干扰小、便于装配、使用方便等特点，可以广泛应用于机器人避障、避障小车、流水线计数及黑白线循迹等众多场合。

这里的循迹是指小车在白色地板上循黑线行走，通常采取的方法是红外探测法。红外探测法，即利用红外线在不同颜色的物体表面具有不同的反射性质的特点，在小车行驶过程中不断地向地面发射红外光，当红外光遇到白色纸质地板时发生漫反射，反射光被装在小车上的接收管接收；如果遇到黑线则红外光被吸收，小车上的接收管接收不到红外光。单片机就是否收到反射回来的红外光为依据来确定黑线的位置和小车的行走路线。红外探测器探测距离有限，一般最大不应超过 15cm。对于发射和接收红外线的红外探头，可以自己制作或直接采用集成式红外探头。

1、自制红外探头电路如图 3-7 所示，红外光的发送接收选用型号为

TCRT5000 的对管。当小车在白色地面行驶时，装在车下的红外发射管发射红外线信号，经白色反射后，被接收管接收，一旦接收管接收到信号，那么图中光敏三极管将导通，比较器输出为低电平；当小车行驶到黑色引导线时，红外线信号被黑色吸收后，光敏三极管截止，比较器输出高电平，从而实现了通过红外线检测信号的功能。将检测到的信号送到单片机 I/O 口，当 I/O 口检测到的信号为高电平时，表明红外光被地上的黑色引导线吸收了，表明小车处在黑色的引导线上；同理，当 I/O 口检测到的信号为低电平时，表明小车行驶在白色地面上。此种方法简单，价格便宜，灵敏度可调，但是容易受到周围环境的影响，特别是在图 3-7 较强的日光日光灯下，对检测到的信号有一定的影响。

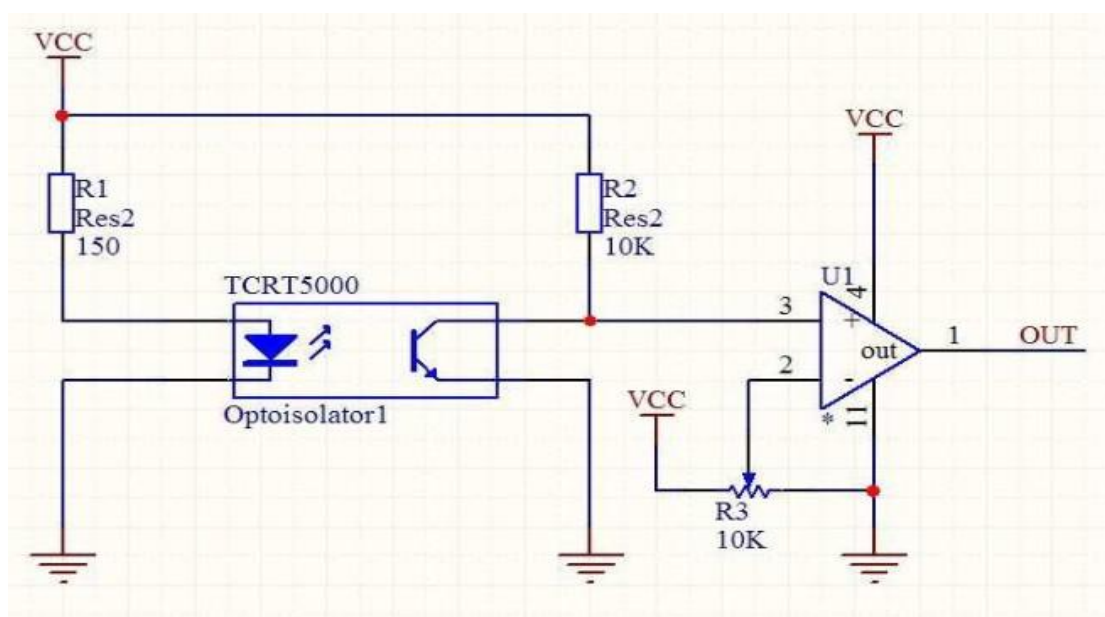


图 3-7 红外对管原理图

2、集成式红外探头可以采用型号为 E3F-DS10C4 集成断续式光电开关探测器，它具有简单、可靠的工作性能，只要调节探头上的一个旋钮就可以控制探头的灵敏度。该探头输出端只有三根线（电源线、地线、信号线），只要将信号线接在单片机的 I/O 口，然后不停地对该 I/O 口进行扫描检测，当其为高电平时则检测到白纸，当为低电平时则检测到黑线。此种探头还能有效地防止普通光源（如日光灯等）的干扰。其缺点则是体积比较大，占用了小车有限的空间。

3.4.2 电机驱动模块工作原理

常用的电机驱动有两种方式：一、采用集成电机驱动芯片。二、采用 N 沟道 MOSFET 和专用栅极驱动芯片设计。为了减少车体复杂程度，我们采用了较为简单的集成的电机驱动芯片 L298N 作为电机的驱动。

L298N是ST公司生产的一种高电压、大电流电机驱动芯片。该芯片采用15脚封装。主要特点是：工作电压高，最高工作电压可达46V；输出电流大，瞬间峰值电流可达3A，持续工作电流为2A；额定功率25W。内含两个H桥的高电压大电流全桥式驱动器，可以用来驱动直流电动机和步进电动机、继电器线圈等感性负载；采用标准逻辑电平信号控制；具有两个使能控制端，在不受输入信号影响的情况下允许或禁止器件工作有一个逻辑电源输入端，使内部逻辑电路部分在低电压下工作；可以外接检测电阻，将变化量反馈给控制电路。使用L298N芯片驱动电机，该芯片可以驱动一台两相步进电机或四相步进电机，也可以驱动两台直流电机。

L298N可以驱动两台直流电机。分别为M1和M2。引脚A，B可用于输入PWM脉宽调制信号对电机进行调速控制。（如果无须调速可将两引脚接5V，使电机工作在最高速状态，既将短接帽短接）实现电机正反转就更容易了，输入信号端IN1接高电平输入端IN2接低电平，电机M1正转。（如果信号端IN1接低电平，IN2接高电平，电机M1反转）控制另一台电机是同样的方式，输入信号端IN3接高电平，输入端IN4接低电平，电机M2正转。（反之则反转），PWM信号端A控制M1调速，PWM信号端B控制M2调速。参考如图3-8。

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

图3-8 L298N控制电机引脚图

小车使用L298N作为小车驱动电机的芯片。OUT1、OUT2和OUT3、OUT4之间分别接2个电动机。IN1、IN2、IN3、IN4引脚从单片机接输入控制电平，控

制电机的正反转，ENA，ENB接控制是能端，控制电机的转挺。具体的电路如图 3-9:

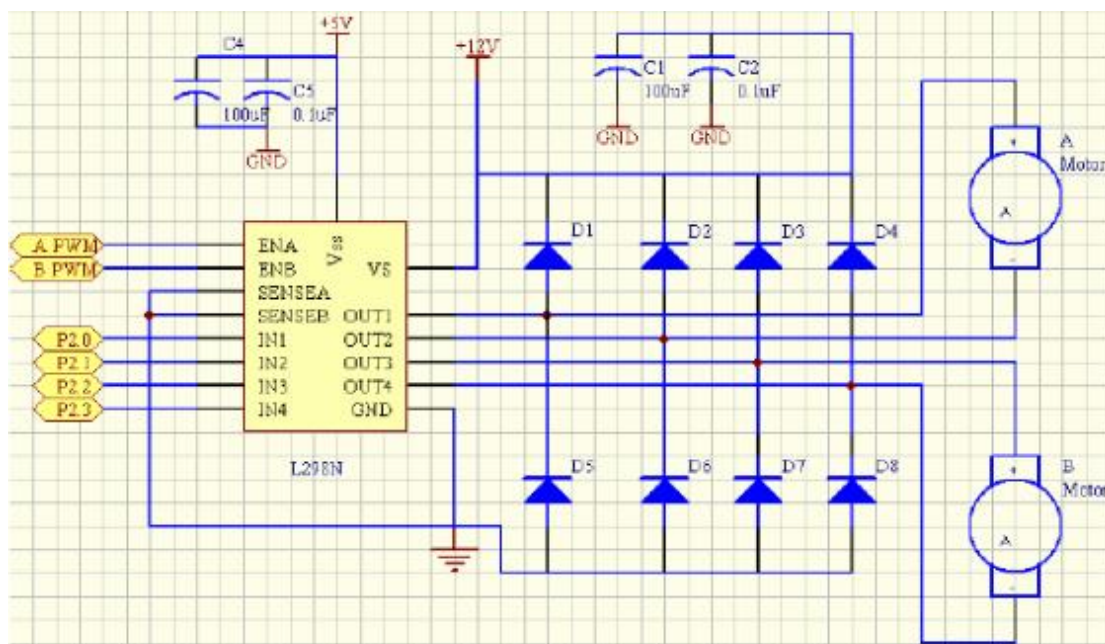


图 3-9 L298N 驱动原理图

3.5 舵机模块工作原理

3.5.1 舵机的概念

在机器人机电控制系统中，舵机控制效果是性能的重要影响因素。舵机可以在微机电系统和航模中作为基本的输出执行机构，其简单的控制和输出使得单片机系统与之接口非常容易。舵机是一种位置（角度）伺服的驱动器，适用于那些需要角度不断变化并可以保持的控制系统。目前在高档遥控玩具，如航模，包括飞机模型，潜艇模型；遥控机器人中已经使用得比较普遍。舵机是一种俗称，其实是一种伺服马达。

其工作原理是：控制信号由接收机的通道进入信号调制芯片，获得直流偏置电压。它内部有一个基准电路，产生周期为20ms，宽度为1.5ms的基准信号，将获得的直流偏置电压与电位器的电压比较，获得电压差输出。最后，电压差的正负输出到电机驱动芯片决定电机的正反转。当电机转速一定时，通过级联减速齿轮带动电位器旋转，使得电压差为0，电机停止转动。

3.5.2 舵机的控制

舵机的控制一般需要一个20ms左右的时基脉冲，该脉冲的高电平部分一般为0.5ms-2.5ms范围内的角度控制脉冲部分。以180度角度伺服为例，那么对应的控制关系是这样的：

0.5ms-----0 度；

1.0ms-----45 度；

1.5ms-----90 度；

2.0ms-----135 度；

2.5ms-----180 度；

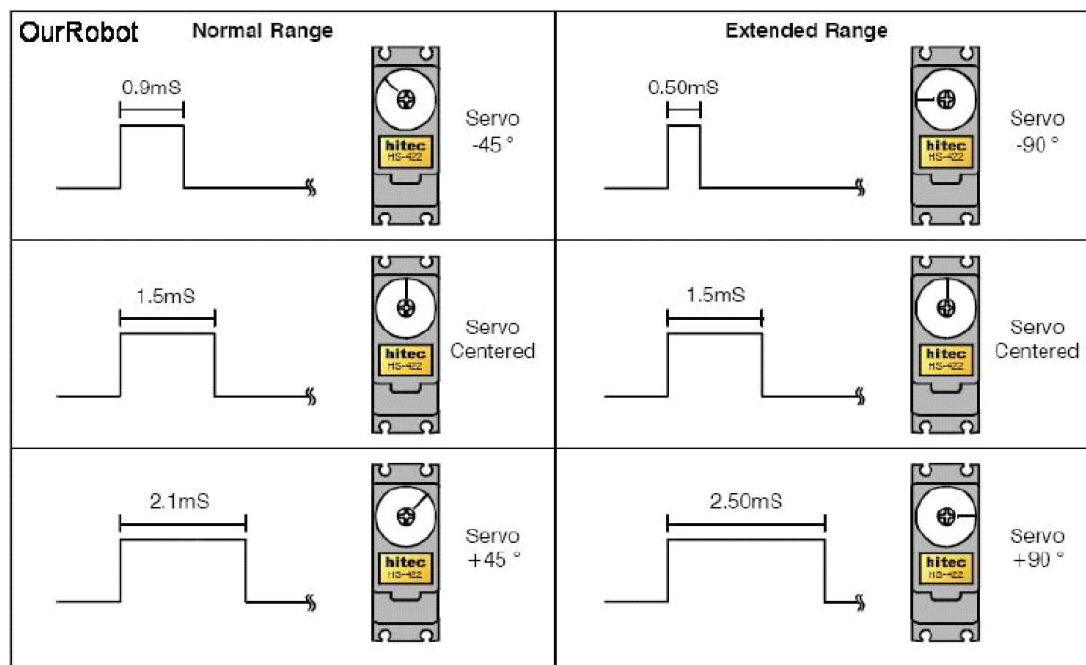


图 3-10 舵机控制原理

3.6 颜色识别模块工作原理

3.6.1 TCS3200 芯片介绍

TCS3200 是 TAOS 公司推出的可编程彩色光到频率的转换器，它把可配置的硅光电二极管与电流频率转换器集成在一个单一的 CMOS 电路上，同时在单一芯片上集成了红绿蓝（RGB）三种滤光器，是业界第一个有数字兼容接口的 RGB 彩色传感器，TCS3200 的输出信号是数字量，可以驱动标准的 TTL 或 CMOS 逻辑输入，因此可直接与微处理器或其他逻辑电路相连接，由于输出的是数字量，并且能够实现每个彩色信道 10 位以上的转换精度，因而不需要 A/D 转换电路，使电路变得更简单。

TCS3200 采用 8 引脚的 SOIC 表面贴装式封装，在单一芯片上集成有 64 个光电二极管，这些二极管分为四种类型，其 16 个光电二极管带有红色滤波器；16 个光电二极管带有绿色滤波器；16 个光电二极管带有蓝色滤波器，其余 16 个不带有任何滤波器，可以透过全部的光信息，这些光电二极管在芯片内是交叉排列的，能够最大限度地减少入射光辐射的不均匀性，从而增加颜色识别的精确度；

另一方面，相同颜色的 16 个光电二极管是并联连接的，均匀分布在二极管阵列中，可以消除颜色的位置误差。工作时，通过两个可编程的引脚来动态选择所需要的滤波器，该传感器的典型输出频率范围从 2Hz—500kHz，用户还可以通过两个可编程引脚来选择 100%、20% 或 2% 的输出比例因子，或电源关断模式。输出比例因子使传感器的输出能够适应不同的测量范围，提高了它的适应能力。例如，当使用低速的频率计数器时，就可以选择小的定标值，使 TCS3200 的输出频率和计数器相匹配。

当入射光投射到 TCS3200 上时，通过光电二极管控制引脚 S2、S3 的不同组合，可以选择不同的滤波器；经过电流到频率转换器后输出不同频率的方波（占空比是 50%），不同的颜色和光强对应不同频率的方波；还可以通过输出定标控制引脚 S0、S1，选择不同的输出比例因子，对输出频率范围进行调整，以适应不同的需求。

3.6.2 TCS3200 识别颜色的原理

（1）三原色的感应原理通常所看到的物体颜色，实际上是物体表面吸收了照射到它上面的白光（日光）中的一部分有色成分，而反射出的另一部分有色光在人眼中的反应。白色是由各种频率的可见光混合在一起构成的，也就是说白光中包含着各种颜色的色光（如红 R、黄 Y、绿 G、青 V、蓝 B、紫 P）。根据德国物理学家赫姆霍兹（Helmholtz）的三原色理论可知，各种颜色是由不同比例的三原色（红、绿、蓝）混合而成的。

（2）TCS3200 识别颜色的原理由三原色感应原理可知，如果知道构成各种颜色的三原色的值，就能够知道所测试物体的颜色。对于 TCS3200 来说，当选定一个颜色滤波器时，它只允许某种特定的原色通过，阻止其他原色的通过。例如：当选择红色滤波器时，入射光中只有红色可以通过，蓝色和绿色都被阻止，这样就可以得到红色光的光强；同时，选择其他的滤波器，就可以得到蓝色光和绿色光的光强。通过这三个值，就可以分析投射到 TCS3200 传感器上的光的颜色。

（3）白平衡和颜色识别原理白平衡就是告诉系统什么是白色。从理论上讲，白色是由等量的红色、绿色和蓝色混合而成的；但实际上，白色中的三原色并不完全相等，并且对于 TCS3200 的光传感器来说，它对这三种基本色的敏感性是不相同的，导致 TCS3200 的 RGB 输出并不相等，因此在测试前必须进行白平衡调整，使得 TCS3200 对所检测的“白色”中的三原色是相等的。进行白平衡调整是为后续的颜色识别作准备。在本装置中，白平衡调整的具体步骤和方法如下：将空的试管放置在传感器的上方，试管的上方放置一个白色的光源，使入射光能够穿过试管照射到 TCS3200 上；根据前面所介绍的方法，依次选通红色、绿色和蓝色滤波器，分别测得红色、绿色和蓝色的值，然后就可计算出需要的 3 个调整参数。

当 TCS3200 识别颜色时，就用这 3 个参数对所测颜色的 R、G 和 B 进行调整。

这里有两种方法来计算调整参数：1、依次选通三颜色的滤波器，然后对 TCS3200 的输出脉冲依次进行计数。当计数到 255 时停止计数，分别计算每个通道所用的时间，这些时间对应于实际测试时 TCS3200 每种滤波器所采用的时间基准，在这段时间内所测得的脉冲数就是所对应的 R、G 和 B 的值。2、设置定时器为一固定时间（例如 10ms），然后选通三种颜色的滤波器，计算这段时间内 TCS3200 的输出脉冲数，计算出一个比例因子，通过这个比例因子可以把这些脉冲数变为 255。在实际测试时，室外同样的时间进行计数，把测得的脉冲数再乘以求得的比例因子，然后就可以得到所对应的 R、G 和 B 的值。颜色识别模块内部电路图，如图 3-11。

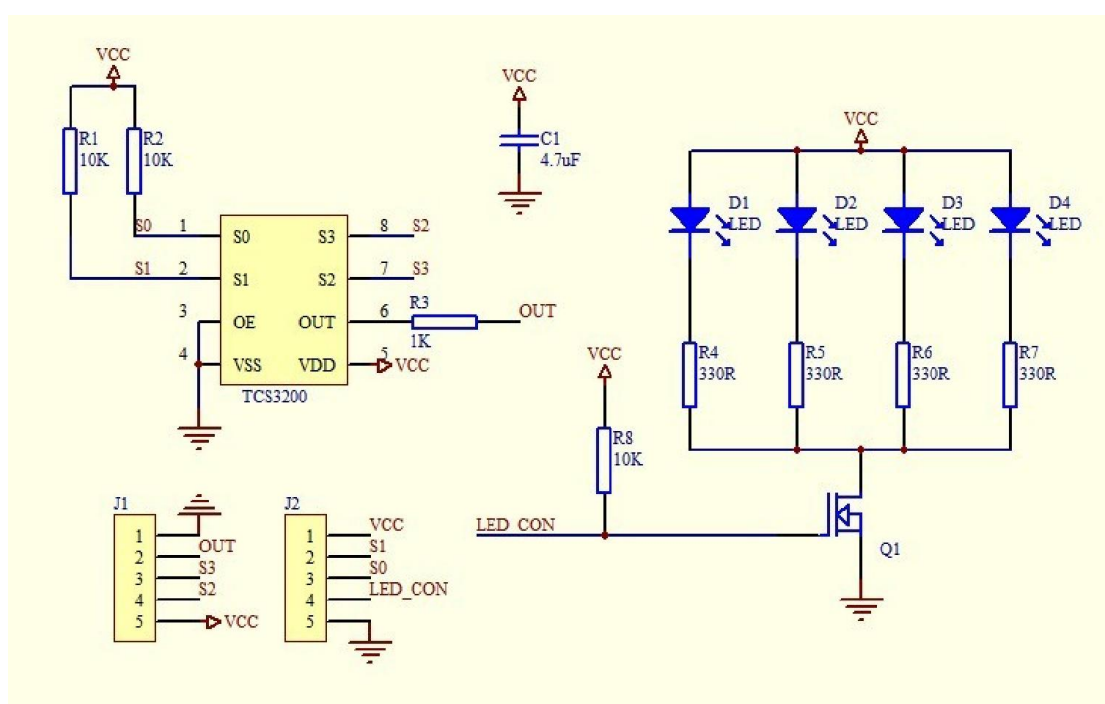


图 3-11 颜色识别模块内部电路图

第四章 系统的软件设计与实现

程序是一个系统的灵魂，本章就系统开发环境的使用，程序设计等展开论述。

4.1 KEIL C51 开发环境的使用

4.1.1 KEIL C51 开发环境的安装

登陆 Keil Software 官方网站打开 KEIL 下载界面，单击图中的灰色阴影处。页面跳转到软件版本的选择页面，单击红色区域的下载后开始下载适用于 STC12C5A60S2 的 KEIL C51 版本。如图 4-1 所示。

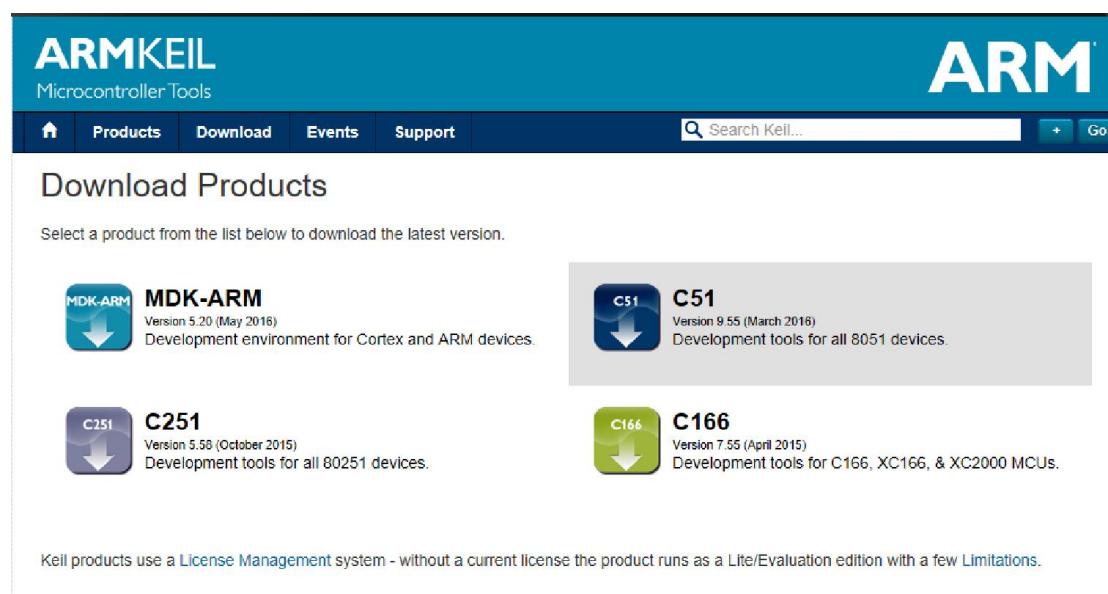


图 4-1 软件下载

下载完成后进入到安装界面。直接单击下一步。由于 KEIL C51 软件本身与中文的兼容性问题，在选择安装路径的时候要保持全英文路径，选好路径之后单击下一步。稍等片刻之后 KEIL C51 的图标会在桌面显示。如图 4-2 所示。

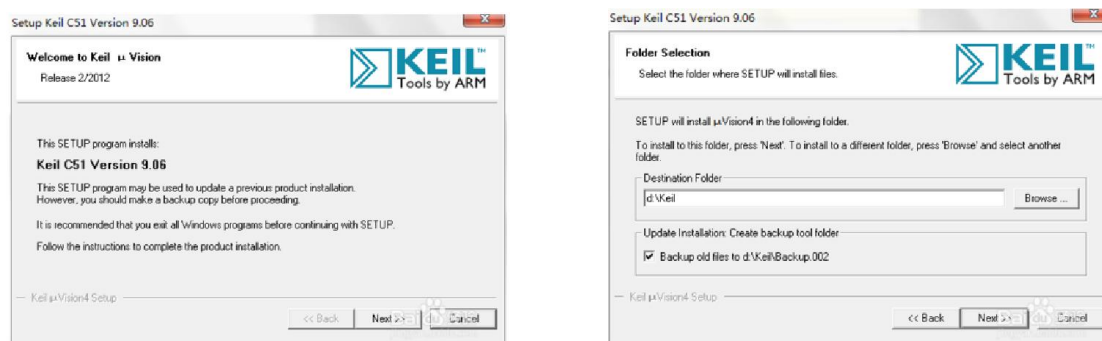


图 4-2 软件安装

4.1.2 KEIL C51 创建新工程

在打开 KEIL C51，选择菜单栏中的 project，新建一个新的 project。然后为工程命名，并选择工程保存的路径。如图 4-3 所示。

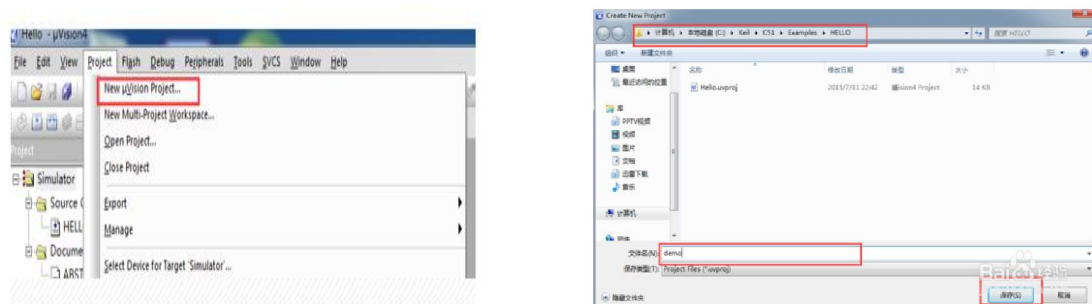


图 4-3 创建新工程

选择本次设计所需要的芯片号 STC12C5A60S2。如图 4-4 所示：

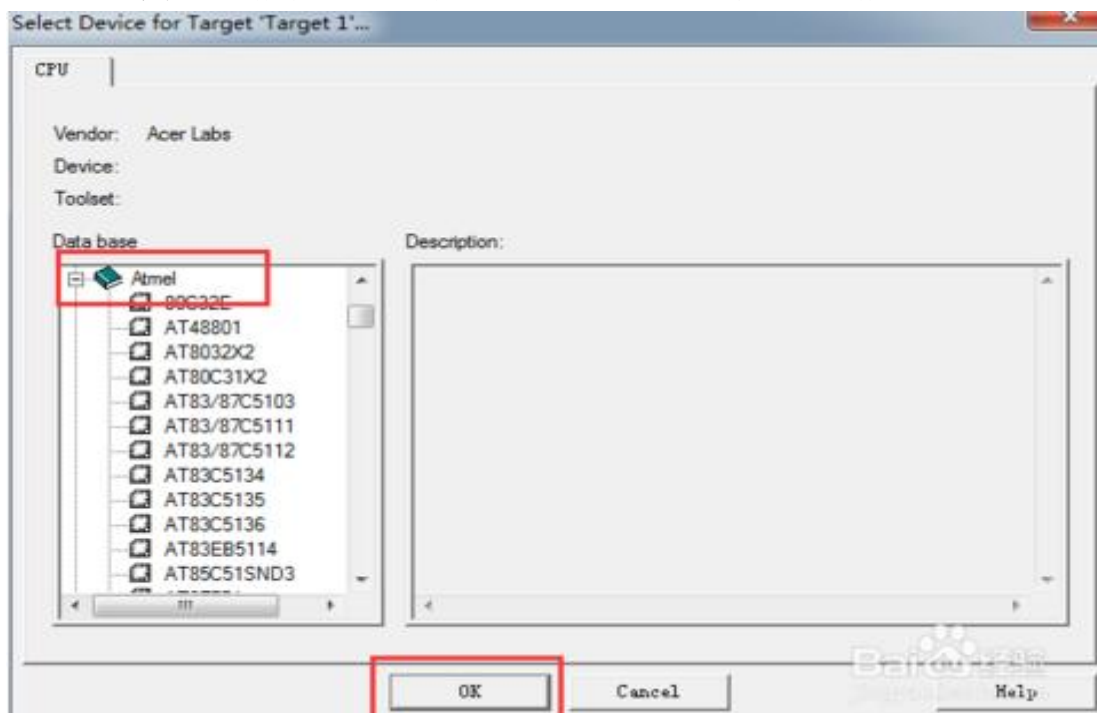




图 4-4 芯片号的选择

4.1.3 程序下载与调试

编写好程序之后将最小系统用串口线与 PC 链接，然后在 KEIL C51 的左侧找到下载图标（）。之后单击工具栏中的 Debug 按键（），弹出程序下载界面。在程序下载界面选择好相应的 .hex 文件，如图 4-5 所示。之后单击打开。这样就实现了程序的下载。

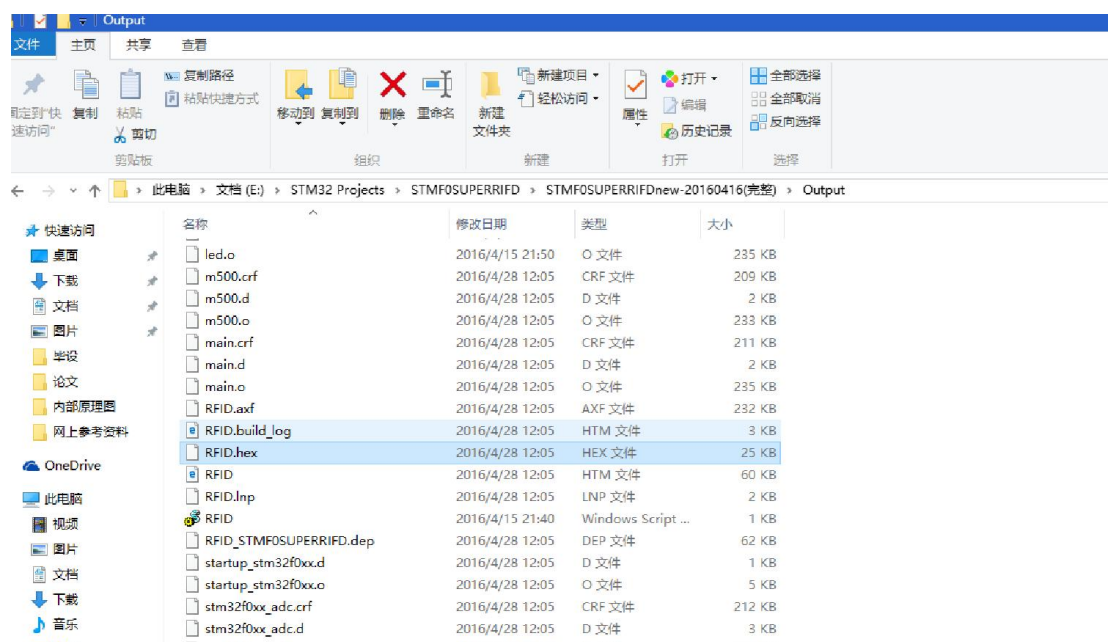


图 4-5 程序下载与调试

4.2 软件系统总体设计

在智能车的软件设计中，用到了单片机的 L298N 模块，用于电机的正反转和舵机的转向角控制；用到了红外循迹 模块，用于对小车行迹方向作为判断；用到了语音识别模块，用于对获取操作人员的指令；用到了颜色识别模块，通过颜色识别筛选出需要的物品；也涉及到了机械臂控制，用于抓取物品。

4.2.1 系统主程序设计

系统通过在主程序内循环调用机械臂控制、红外循迹、颜色识别和语音识别等功能子模块，程序的执行为先对各个模块进行初始化，然后在主程序中完成相应的功能。高效稳定的软件程序是智能车平稳快速寻线的基础。主程序主要用于对系统的初始化和对各个子函数的调用。主程序流程图如图 4-6。

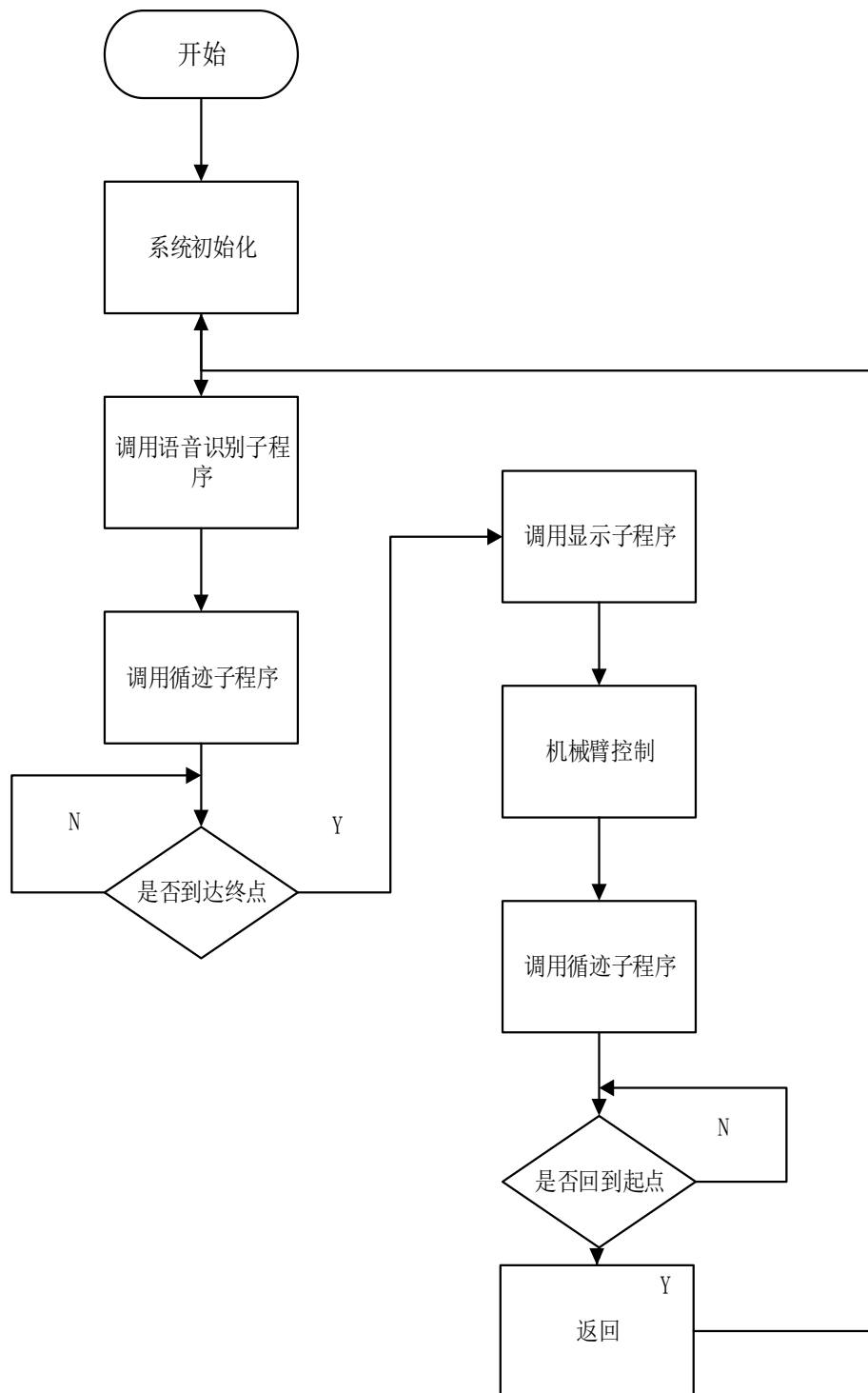


图 4-6 主程序流程图

4.2.2 语音识别程序设计

通过设置 LD3320 芯片的寄存器，识别关键词的内容动态地传入芯片中，再反馈给单片机。两个单片机之间实现通信，实现语音控制小车取物的过程。语音识别程序流程图，如图 4-7 所示。

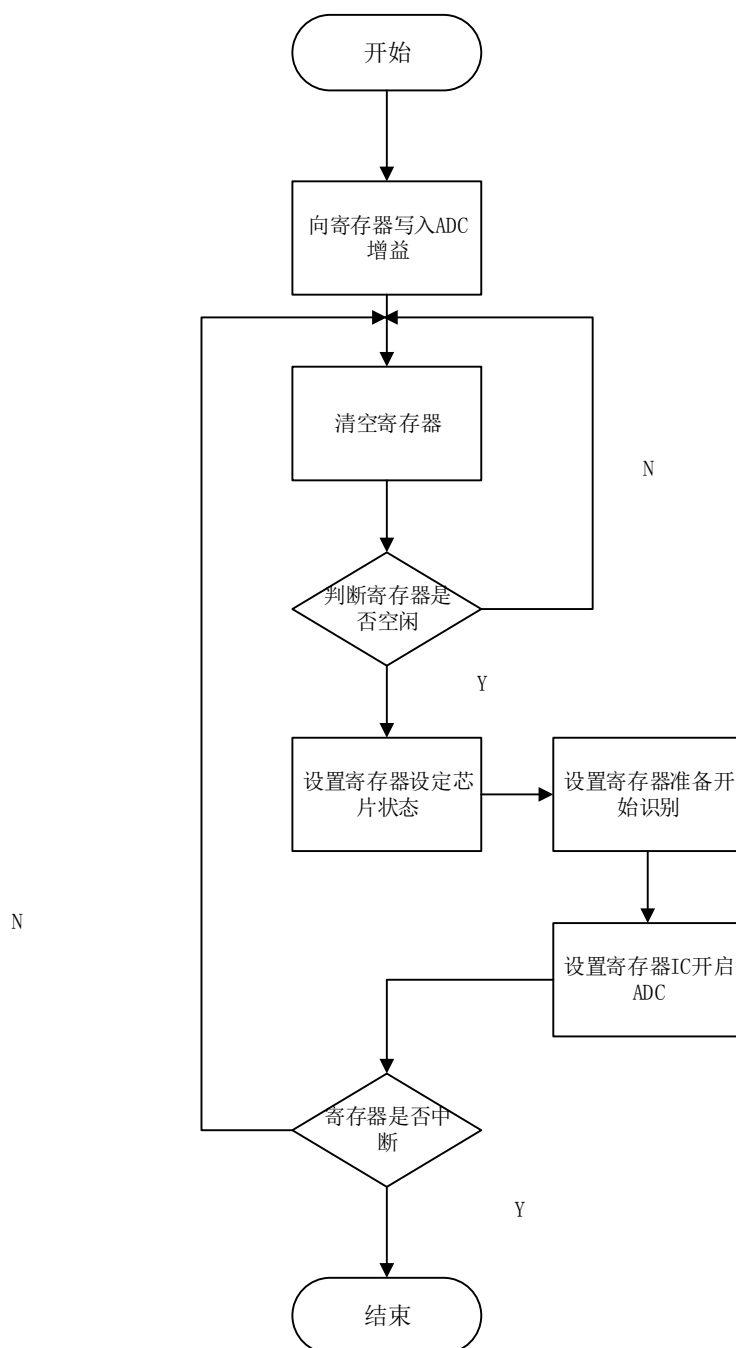


图 4-7 语音识别程序流程图

4.2.3 红外循迹程序设计

通过红外传感器检测黑线传给单片机，经过处理，然后由单片机根据所检测的各种数据实现对小车的智能控制，如图 4-8 所示。

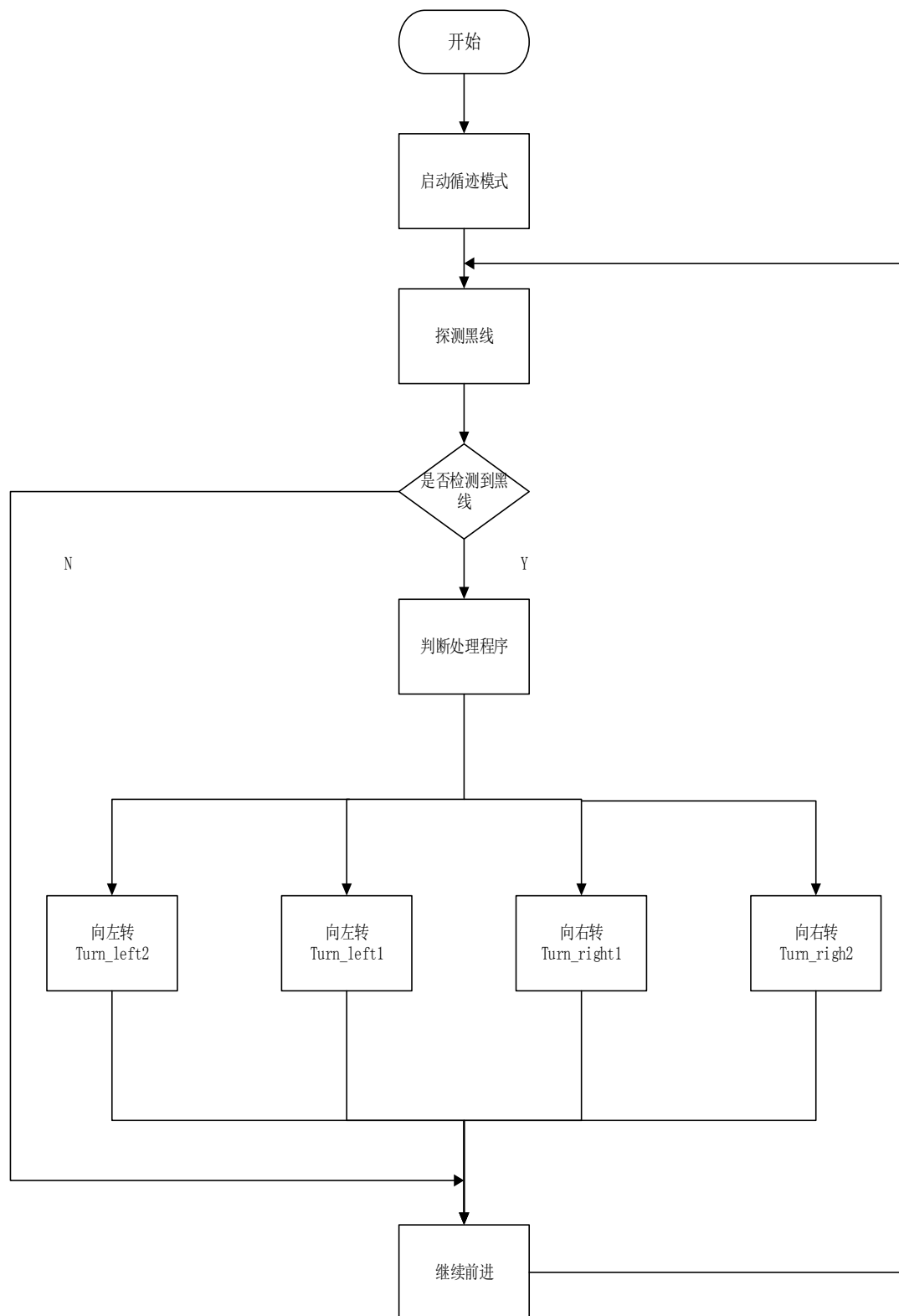


图 4-8 红外循迹程序流程图

4.2.4 机械臂控制程序设计

单片机待机等待指令，接受指令后，通过调整姿态，抓取物品放回后，重新等待下一条指令，如图 4-9 所示。

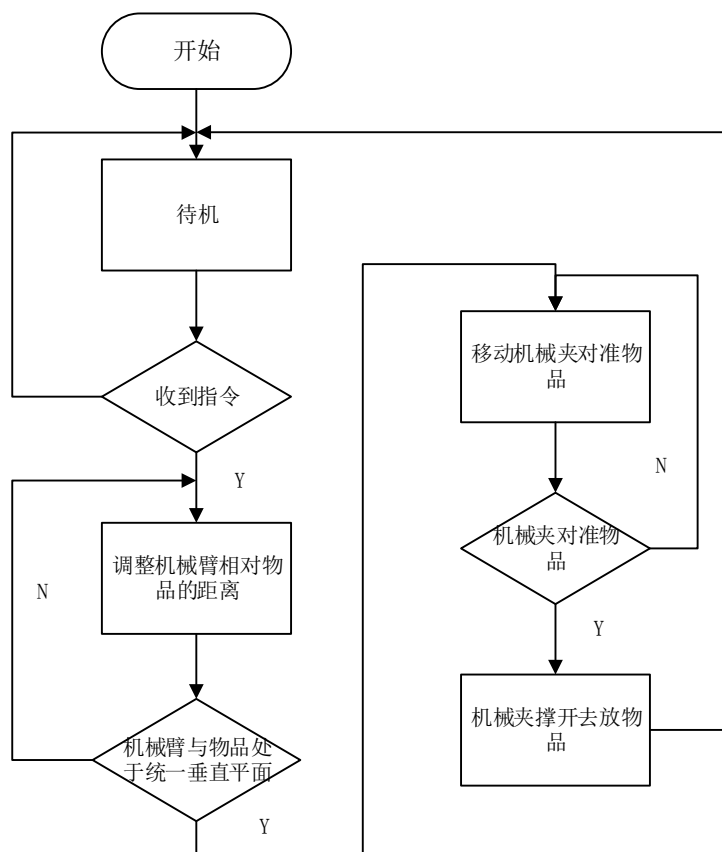


图 4-9 机械臂控制程序流程图

4.2.5 颜色识别程序设计

在程序流程中：系统初始化负责设置单片机的定时器的工作方式，选择 TCS3320 的输出比例因子，使能输出引脚及通信参数的设置。初始化完成后，检测是否需要进行白平衡调整。如有白平衡子程序；否则转跳到下一步，检测是否需要进行颜色识别。如不需要颜色识别，返回；如需颜色识别，调用颜色识别子程序，直到颜色识别完毕，如图 4-10 所示。

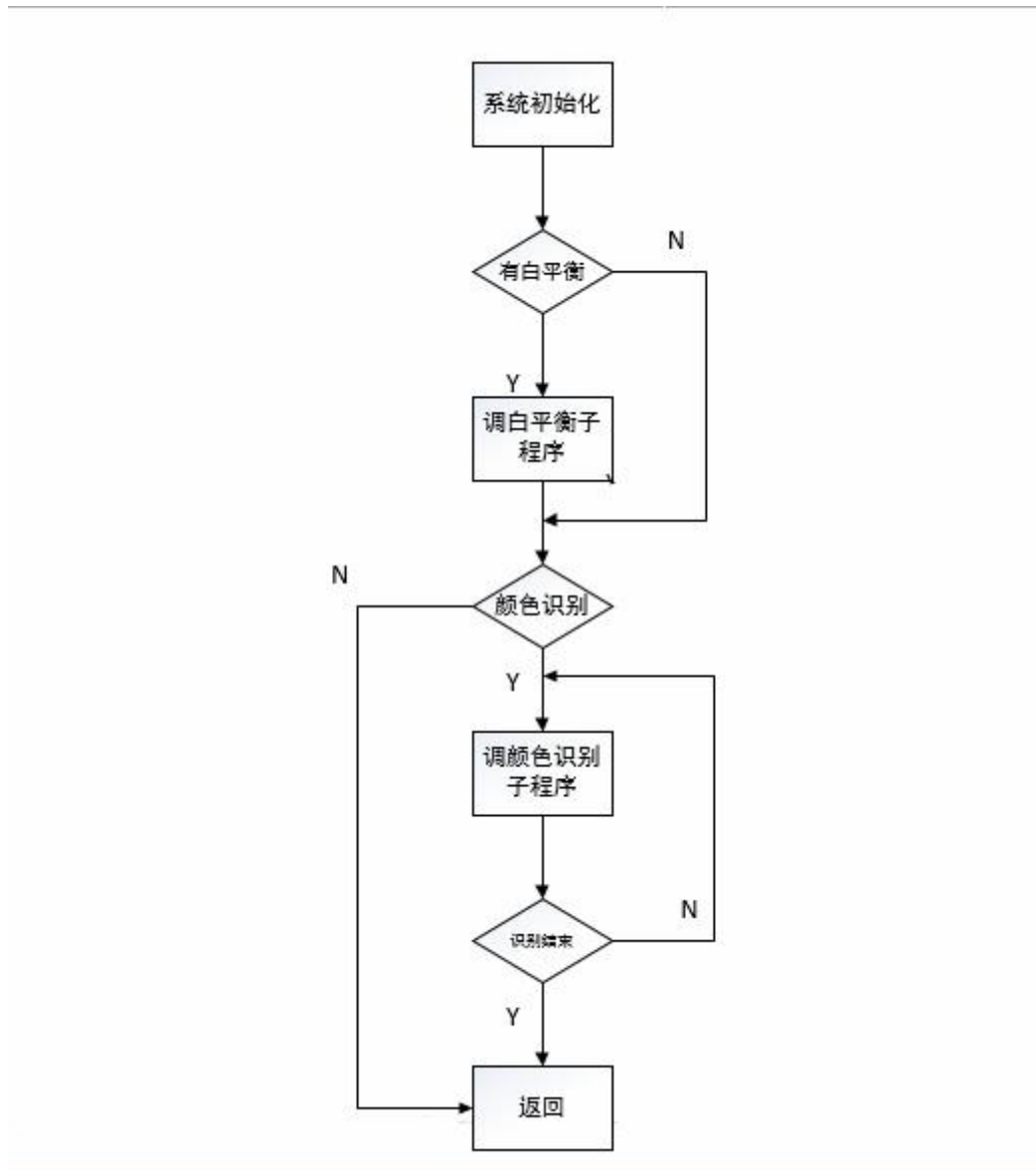


图 4-10 颜色识别程序流程图

第五章 系统调试与总结

5.1 小车调试方法设计

智能小车采用模块化设计思想。然后将各模块拆分为软硬两部分，在完成硬件电路和软件的编写调试后，最后将软硬件结合进行测试。主要利用了 STC12C5A60S2 单片机的中断，定时器等功能。在调试小车循迹行走时，小车调整轨迹不准确，通过对算法和程序的改进，最后使小车的准确无误的完成各个动作状态。

测试主要用的的工具设备主要有：SAMSUNG 笔记本电脑、示波器、万用表、串口线。在 SAMSUNG 笔记本电脑上安装 STC12C5A60S2 单片机开发软件 KEIL C51 以提供软件测试环境，同时安装 Altium Designer 作为开发硬件模块的工具。

在进行硬件电路的实验调试时，首先检测电路板各个主要模块之间的连通性，既保证硬件电路焊接及连接无误，是否有短路、虚焊现象；对系统各硬件电路进行检测，保证各模块可以正常工作，同时保证各主要硬件电路可以有主控 STC12C5A60S2 单片机进行控制；对系统主控芯片 STC12C5A60S2 单片机的通信进行测试；结合软件 KEIL C51 在线调试，利用硬件平台进行功能性检测，最终验证系统软硬件综合功能的实现。

5.2 测试环境和方法设计

用纸板划定一定范围用黑胶条做网格提供小车循迹，放置一些不同颜色的小球在网格的交叉点上。

测试方法：

1、分模块测试

红外循迹小车模块：开启电源，在按下复位键后能否自动沿黑胶条行进，设置路障观察小车运行情况。机械臂舵机控制模块：在单片机控制下输入抓取信号，观察机械臂抓取物品情况。语音识别模块：将语音识别模块与主控芯片相连的输出端连接至示波器，在下达不同语音指令后观察示波器电位变化。

2、整体测试

将物品在地图上随机放置。将小车放置到开始区域，给智能小车供电，打开电源，按下复位键，测试人员发出语音指令。测试小车整体运行情况，然后再针对各个模块进行微调。

5.3 工作总结

本设计重点在于构建一个完整的智能车系统。所以对于有些部分并未进行详尽的说明和研究。而本设计最主要的目标就是实现小车正常行驶在道路中心。而智能车路面的各种情况复杂多变，对于道路的分析，颜色的处理方面还需要更加深入的研究。对于如何在较短的时间内安全平稳地行驶，如何选择最优化的路径

方面也是值得改进和研究的。希望在日后的学习和工作工程中积累更多的经验和知识，从而使本设计更加的完善稳定。

第六章 结束语

经过一段时间的学习和研究，终于使语音控制的智能取物小车能够实现其基本功能，并且已经实际测试成功。实践证明通过语音控制与小车的相连可以利用电脑技术的尖端优势和强大的功能使得整个系统的本身的功能提升不少，在人机对话方面更具方便性和快捷性。

本文主要介绍了语音控制的智能车的总体设计方案，包括硬件电路设计和软件的设计。在设计和制作智能车的过程中，翻阅了大量的书籍和资料，结合前人的经验也进行了相应的改进，并取得了一定的成果。在整个设计过程中不断发现问题，及时分析问题，克服各种困难最终完成了这辆智能车的制作。智能车是一个多学科、综合性的项目，其中设计涉及了控制算法、传感技术、电子信息、模式识别和机械等多个学科，在整个准备的过程中，不仅仅需要把所学的理论知识应用于实际，还需要自学了大量的新知识。不仅开拓了视野，而且还提高了动手能力、运用知识的能力、分析解决问题的能力。

参考文献

- [1]张伟等.Protel DXP 高级应用[M]. 北京:人民邮电出版社,2004.
- [2]袁明新,李勇,张勇.汽车状态远程监测中计算机与手机通讯的实现[J].工业控制计算机,2003,16(3):53-55.
- [3]阮毅,陈伯时.电力拖动自动控制系统[M].北京:机械工业出版社,2010.
- [4]谭浩强著.C 程序设计[M].北京:清华大学出版社,2003.
- [5]齐保良,杜文洪.利用全球移动通信系统的公交车预报与调度系统[J].电子技术,2003,11(2):27-30.
- [6]王倩,许永生.出租车计价器的硬件设计[J].计算机工程应用技术, 2010,02.
- [7]葛亚明,刘涛,王宗义.视频同步分离芯片 LM1881 及其应用[J].应用科技,2007,31(9):6-7.
- [8]倪建军,范跃祖,李松林.利用 GSM 短信息业务实现 GPS 车辆监控[J].电子技术,2003,26(2):27-30.
- [9]谢文策.线阵 CCD 图像不失真采集算法及实现[J].信息通信,2009,9(4):4-6.
- [10]叶常春,罗金平,周兴铭.一种交通信息服务系统设计[J].电子技术应用,2002,16(2):47-49.
- [11]陆健贤.移动通信分布系统原理与工程设计[M].北京:机械工业出版社,2003:124-127.
- [12]John Scourias.Overview of the Global System for Mobile Communications[J]. Oxford University Press ,2000,24(13):78-87.
- [13]Keith Clarke. Getting Started with Geographical InformationSystem.Prentice Hall[J]. Oxford University Press, 2001,23(20):23-30.
- [14]Freescale Semiconductor Inc.S12PWM8B8CV1/D Block User Guide V01.17[Z].2004.1-51.
- [15]Julio E,Normey-Rico.Mobile robot path tracking using a robust PID controller [J], Control Engineering Practice,2001,9(11):1209-1214.

附 录

智能循迹小车设计程序源文件

////////////////////////////////////小车循迹////////////////////////////////////

```
#include "reg52.h"
```

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
/******直流电机 PWM******/
```

```
uint i_a,i_b;
```

```
uint j_a,j_b;
```

```
sbit PWM_1=P2^0;
```

```
sbit PWM_2=P2^1;
```

```
/******L298N 芯片定义******/
```

```
sbit L1_A=P2^2;
```

```
sbit L2_A=P2^3;
```

```
sbit L3_B=P2^4;
```

```
sbit L4_B=P2^5;
```

```
/******° 红外循迹口******/
```

```
sbit ss1=P1^0;
```

```
sbit ss2=P1^1;
```

```
sbit ss3=P1^2;
```

```
sbit ss4=P1^3;
```

```
sbit ss5=P1^4;
```

```
uchar flag,flag1;
```

```
uchar flag2,flag3;
```

```
/******° 红外探头.******/
```

```
sbit HW5=P2^6;
```

```
sbit HW6=P2^7;
```

```
/******×Ó³ÌÐò******/
```

```
void Car_go(); //直走
```

```
void Car_right(); //右转
```

```
void Car_left(); //左转
```

```
void Car_right_1(); //右转
```

```
void Car_left_1();    //左转
//void Car_round();  //掉头
void Car_stop();
void delay(uint ms);
void delay1(uint ms);

void system_start()
{
    TMOD=0x01;
    TH0=(65535-100)/256;
    TL0=(65535-100)%256;
    EA=1;
    ET0=1;
    TR0=1;
}

void CarGoLine()      //go in black line
{
    if(ss2==0) {Car_right_1(); delay1(1);}
    if(ss1==0) {Car_right();delay(24); }
    if(ss4==0) {Car_left_1();delay1(1); }
    if(ss5==0) {Car_left(); delay(24); }
    if(ss3==0) Car_go();
}

uchar X_one(void)
{
    if((s1==0)&&(s2==0))
        if((s4==1)&&(s5==1))
            flag++;
    return flag;
}

uchar X_two(void)
{
```

```
    if((s4==0)&&(s5==0))
        if((s1==1)&&(s2==1))
            flag1++;
    return flag1;
}

uchar Y(void)
{
    if((s4==0)&&(s5==0))
        if((s1==0)&&(s2==0))
            flag2++;
    return flag2;
}

void dic_2()
{
    if((s4==1)&&(s5==1))
        if((s1==1)&&(s2==1))
            if(s3==0)
                Car_go();
}

void dic()
{
    X_one();
    X_two();
    Y();
    if(flag==2) {Car_left();delay(200);dic_2();flag=0;}
    if(flag1==2) {Car_right();delay(200);dic_2();flag1=0;}
    if(flag2==3) {Car_right();delay(200);dic_2();flag2=0;++flag3;}
    while(flag3==2)
    {
        Car_go();
        if((s4==0)&&(s5==0))
            if((s1==0)&&(s2==0))
```

```

        Car_stop();
    }
}

void main()
{
    system_start();
    Car_stop();
    while(1)
    {
        CarGoLine();
//        dic();

    }
}

/*****小延时*****/
void delay1(uint ms)
{
    uint t;
    while(ms--)
        for(t=0;t<2;t++);
}

/*****大延时*****/
void delay(uint ms)
{
    uint t;
    while(ms--)
        for(t=0;t<120;t++);
}

/*****小车运行状态*****/
void Car_stop()
{
    L1_A=0;

```

```
    L2_A=0;
    L3_B=0;
    L4_B=0;
}

void Car_go()
{
    L1_A=0;
    L2_A=1;
    L3_B=0;
    L4_B=1;
}

void Car_right()
{
    L1_A=0;
    L2_A=1;
    L3_B=1;
    L4_B=0;
}

void Car_left()
{
    L1_A=1;
    L2_A=0;
    L3_B=0;
    L4_B=1;
}

void Car_right_1()
{
    L1_A=0;
    L2_A=1;
    L3_B=0;
    L4_B=0;
}

void Car_left_1()
{
```



```

    L1_A=0;
    L2_A=0;
    L3_B=0;
    L4_B=1;
}

void Car_round()
{
    L1_A=0;
    L2_A=1;
    L3_B=1;
    L4_B=0;
}

/*****TIME0 为舵机产生 PWM*****/

void T0_time() interrupt 1
{
    TH0=(65536-100)/256;    //100
    TL0=(65536-100)%256;

    i_a++;
    j_a++;
    if(j_a==111){PWM_1=0;}    //左电机调速
    if(i_a==190){PWM_1=1;i_a=0;j_a=0;}    //20ms 一个周期

    i_b++;
    j_b++;
    if(j_b==100){PWM_2=0;}    //右电机调速
    if(i_b==200){PWM_2=1;i_b=0;j_b=0;}    //20ms 一个周期
}

//////////////////////////////// TCS3200 颜色识别////////////////////////////////
sbit    LCM_RS=P2^0;    //LCD1602 数据端口
sbit    LCM_RW=P2^1;    //LCD1602 控制端口

```

```

sbit    LCM_EN=P2^2;    //LCD1602 控制端口
sbit s2=P1^1;           //TCS3200 S2
sbit s3=P1^0;           //TCS3200 S3
                        //TCS3200 S0 模块内部默认上拉
                        //TCS3200 S1 模块内部默认上拉
                        //TCS3200 OE 模块内部接地
sbit test_pin=P1^2;    //用示波器看这个引脚，可知道定时器中断频率
//变量、常量定义
uchar ge,shi,bai ;
uchar rp=2,gp=2,bp=2; //定义比例因子，具体环境可以修改
uchar count;           //颜色标志位（0：红 1：绿 2：蓝）

//显示数组
uchar disp_R[3]; //红
uchar disp_G[3]; //绿
uchar disp_B[3]; //蓝

//*****定义函数*****
void    delay(unsigned int k);
void    InitLcd();
void    WriteDataLCM(uchar dataW);
void    WriteCommandLCM(uchar CMD,uchar Attribc);
void    DisplayOneChar(uchar X,uchar Y,uchar DData);

//*****LCD1602 初始化*****
void InitLcd()
{
    WriteCommandLCM(0x38,1);
    WriteCommandLCM(0x08,1);
    WriteCommandLCM(0x01,1);
    WriteCommandLCM(0x06,1);
    WriteCommandLCM(0x0c,1);
}

//*****检测忙信号*****

```

```
void WaitForEnable(void)
{
    DataPort=0xff;
    LCM_RS=0;LCM_RW=1;_nop_();
    LCM_EN=1;_nop_();_nop_();
    while(DataPort&0x80);
    LCM_EN=0;
}

//*****写命令至 LCD*****
void WriteCommandLCM(uchar CMD,uchar Attribute)
{
    if(Attribute)WaitForEnable();
    LCM_RS=0;LCM_RW=0;_nop_();
    DataPort=CMD;_nop_();
    LCM_EN=1;_nop_();_nop_();LCM_EN=0;
}

//*****写数据至 LCD*****
void WriteDataLCM(uchar dataW)
{
    WaitForEnable();
    LCM_RS=1;LCM_RW=0;_nop_();
    DataPort=dataW;_nop_();
    LCM_EN=1;_nop_();_nop_();LCM_EN=0;
}

//*****写一个字符数据到指定的目标*****
void DisplayOneChar(uchar X,uchar Y,uchar DData)
{
    Y&=1;
    X&=15;
    if(Y)X|=0x40;
    X|=0x80;
    WriteCommandLCM(X,0);
}
```

```
WriteDataLCM(DData);
}

//*****延时函数*****
void delay(unsigned int k)
{
    unsigned int i,j;
    for(i=0;i<k;i++)
    {
        for(j=0;j<121;j++)
        {;}
    }
}

/*****
* 函数名称: t0_init()
* 函数功能: 定时器 0 初始化
*****/
void t0_init()
{
    TMOD=0x51;          //T1 计数 T0 定时 工作方式 1

    TH1=0x00;           //技术初值
    TL1=0x00;

    TH0=0xE0;
    TL0=0x00;           //11.0592M 晶振 10ms
    EA=1;               //开中断

    ET0=1;
    TR0=1;              //启动
    TR1=1;
}

//*****
```

```
//数值转换出个十百千的 ASCII 码
//*****

void conversion(uint temp_data)
{
    bai=temp_data/100+0x30 ;
    temp_data=temp_data%100;    //取余运算
    shi=temp_data/10+0x30 ;
    ge=temp_data%10+0x30;        //取余运算
}

/*****
* 函数名称: c10ms_out()
* 函数功能: 定时中断 0 服务程序
            修改颜色标志 disp tc (0: 红 1: 绿 3: 蓝)
            设置 s0 s1 s2 选择滤波器
            计算脉冲, 读取色值
* 入口参数: 无
* 出口参数: 无
/*****/

void c10ms_out() interrupt 1
{
    uint temp;

    test_pin=!test_pin; //测试定时器中断频率引脚, 可以用示波器观察
    TR0=0;                //关闭定时
    TR1=0;                //关闭计时
    // count+1 实现先检查绿色, 在检测蓝色, 然后检测红色, 循环检测
    if(count==0)
    {
        count++;
        s2=1;s3=1;        //选择滤波器为蓝色
        temp=(8<<TH1)+TL1; //计算这段时间内 TCS3320 的输出脉冲数
        temp/=rp;
        conversion(temp);
        disp_R[2]=ge;      //因为这次的中断, 是上一次选择滤波器的数值
        disp_R[1]=shi;
        disp_R[0]=bai;
    }
}
```

```
}

else if(count==1)
{
    count++;
    s2=0;s3=1;           //选择滤波器为红色

    temp=(8<<TH1)+TL1;   //计算这段时间内 TCS230 的脉冲数
    temp/=gp;
    conversion(temp);
    disp_G[2]=ge;         //因为这次的中断，是上一次选择滤波器的数值
    disp_G[1]=shi;
    disp_G[0]=bai;
}

else if(count==2)
{
    count=0;
    s2=0;s3=0;           //选择滤波器为蓝色

    temp=(8<<TH1)+TL1;   //计算这段时间内 TCS230 的脉冲数
    temp/=bp;
    conversion(temp);
    disp_B[2]=ge;         //因为这次的中断，是上一次选择滤波器的数值
    disp_B[1]=shi;
    disp_B[0]=bai;
}

//定时器计数器的赋初值
TH0=0xE0;
TL0=0x00; //11.0592M 晶振，10ms
TL1=0x00; //计数器清零
TH1=0x00; // 计数器清零
TR0=1;    //打开定时器
```

```
    TR1=1;    //打开定时器
}

/*****
* 函数名称: main()
*****/
void main()
{

    delay(10);
    InitLcd();    //lcd 初始化
    s2=0;        //初始设定 S2 引脚
    s3=0;        //初始设定 S3 引脚
    t0_init();    //定时计数初始化

    while(1)
    {
        DisplayOneChar(0,0,'G');
        DisplayOneChar(1,0,':');
        DisplayOneChar(2,0,disp_G[0]);
        DisplayOneChar(3,0,disp_G[1]);
        DisplayOneChar(4,0,disp_G[2]);    //绿

        DisplayOneChar(6,0,'B');
        DisplayOneChar(7,0,':');
        DisplayOneChar(8,0,disp_B[0]);
        DisplayOneChar(9,0,disp_B[1]);
        DisplayOneChar(10,0,disp_B[2]);    //蓝

        DisplayOneChar(0,1,'R');
        DisplayOneChar(1,1,':');
        DisplayOneChar(2,1,disp_R[0]);
        DisplayOneChar(3,1,disp_R[1]);
```

```

        DisplayOneChar(4,1,disp_R[2]);        //红
        delay(100);
    }
}

////////////////////////////////机械臂控制程序////////////////////////////////
//舵机控制变量
bit            motor1_level;        //舵机 1 电平标志
sbit           motor1_ctrl=P1^0;
bit            motor2_level;        //舵机 2 电平标志
sbit           motor2_ctrl=P1^1;
//定时器初值
unsigned int    init_value0_H;        //舵机 0° 高电平期间的定时器初值
unsigned int    init_value90a_H;    //舵机-90° 高电平期间的定时器初值
unsigned int    init_value90b_H;    //舵机 90° 高电平期间的定时器初值
unsigned int    init_value0_L;        //舵机 0° 低电平期间的定时器初值
unsigned int    init_value90a_L;    //舵机-90° 低电平期间的定时器初值
unsigned int    init_value90b_L;    //舵机 90° 低电平期间的定时器初值
unsigned int    init_value;        //舵机电平持续期间定时器初值
unsigned char*   p;                //从整形数中取出高低字节的指针
unsigned char    TH,TL;            //舵机电平持续期间的定时器初值高低字节

//动作间隔计数
//每 50 个 PWM 周期完成 1 次转动，即 1 秒摆动 1 哥角度
//每 PWM 周期触发两次定时器，实现 PWM 波高低电平转化
unsigned int     T0_counter=0;
unsigned int     T1_counter=0;
//其他
unsigned int     i;

/*-----
控制角度转换为定时器初值函数
-----*/

// 计算 PWM 周期内高电平持续时间对应的定时器初值函数
//输入控制角度和脉宽当量，输出高电平持续时间的定时器初值

```



```

unsigned int TimerInitValueH(float degree,float us_per_degree)
{
    //根据角度计算脉宽
    unsigned int pulse_width=(unsigned int)(1500.0+degree*us_per_degree);//us
    unsigned int init_value_H=65536-pulse_width;
    return init_value_H;
}

// 计算 PWM 周期内低电平持续时间对应的定时器初值函数
//输入控制角度和脉宽当量，输出高电平持续时间的定时器初值
unsigned int TimerInitValueL(unsigned int init_value_H)
{
    unsigned int init_value_L=65536-init_value_H+45536;
    return init_value_L;
}

/*-----
程序
功能：配置定时器模式、调用辅助函数计算定时参数、启动定时器
-----*/

void main()
{
    //配置定时器工作模式寄存器
    TMOD=0x11;
    EA=1;
    ET0=1;
    ET1=1;

    //计算各角度高电平和低电平持续时间的定时器初值
    init_value0_H=TimerInitValueH(0.0,2100.0/180.0);
    init_value90a_H=TimerInitValueH(-90.0,2100.0/180.0);
    init_value90b_H=TimerInitValueH(90.0,2100.0/180.0);
    init_value0_L=TimerInitValueL(init_value0_H);
    init_value90a_L=TimerInitValueL(init_value90a_H);
    init_value90b_L=TimerInitValueL(init_value90b_H);

```

```
//设置 0° 高电平持续时间的定时器初值高低字节
p=(unsigned char*)&init_value0_H;
TH0=*p;TH1=*p;
p++;
TL0=*p;TL1=*p;

//两个舵机设置高电平
motor1_ctrl=motor1_level=1;
motor2_ctrl=motor2_level=1;

//启动定时器 0，等待一段时间后启动定时器 1
TR0=1;
for(i=0;i<1000;i++)
{
}
TR1=1;

//保持程序时钟运行
while(1);
}

//定时器 0 中断服务程序
void T0_Server() interrupt 1
{
//1、条件判断和计算准备
//每次计数电平取反
T0_counter++;
motor1_level=!motor1_level;

//根据定时器触发次数和电平高低，确定电平持续时间
if(T0_counter<100)
{
if(motor1_level)
init_value=init_value0_H;
```

```
        else
            init_value=init_value0_L;
    }
    else if(T0_counter<200)
    {
        if(motor1_level)
            init_value=init_value90a_H;
        else
            init_value=init_value90a_L;
    }
    else if(T0_counter<300)
    {
        if(motor1_level)
            init_value=init_value0_H;
        else
            init_value=init_value0_L;
    }
    else if(T0_counter<400)
    {
        if(motor1_level)
            init_value=init_value90b_H;
        else
            init_value=init_value90b_L;
    }
    else
    {
        T0_counter=0;
        init_value=init_value0_H;
    }

    p=(unsigned char*)&init_value;
    TH=*p;
    p++;
    TL=*p;
    motor1_ctrl=motor1_level;
```

```
    TH0=TH;
    TL0=TL;
}
void T1_Server() interrupt 3
{
    T1_counter++;
    motor2_level=!motor2_level;
    if(T1_counter<100)
    {
        if(motor2_level)
            init_value=init_value0_H;
        else
            init_value=init_value0_L;
    }
    else if(T1_counter<200)
    {
        if(motor2_level)
            init_value=init_value90a_H;
        else
            init_value=init_value90a_L;
    }
    else if(T1_counter<300)
    {
        if(motor2_level)
            init_value=init_value0_H;
        else
            init_value=init_value0_L;
    }
    else if(T1_counter<400)
    {
        if(motor2_level)
            init_value=init_value90b_H;
        else
            init_value=init_value90b_L;
    }
}
```

```

else
{
    T1_counter=0;
    init_value=init_value0_H;
}
p=(unsigned char*)&init_value;
TH=*p;
p++;
TL=*p;
motor2_ctrl=motor2_level;    //µçÆ½Ëì·
TH1=TH;
TL1=TL;
}

////////////////////////////////////语音识别模块程序////////////////////////////////////
#include "config.h"

/*****
*****/

//  nAsrStatus 用来在 main 主程序中表示程序运行的状态，不是 LD3320 芯片内部的状态寄存器
//  LD_ASR_NONE:      表示没有在做 ASR 识别
//  LD_ASR_RUNING:     表示 LD3320 正在做 ASR 识别中
//  LD_ASR_FOUNDOK:    表示一次识别流程结束后，有一个识别结果
//  LD_ASR_FOUNZERO:   表示一次识别流程结束后，有一个没有识别结果
//  LD_ASR_ERROR:      表示一次识别流程中 LD3320 芯片内部出现不正确的状态

/*****
*****/

uint8 idata nAsrStatus=0;
extern void _nop_ (void);
void MCU_init();
uint8 RunASR();
void ProcessInt0(); //识别处理函数
void delay(unsigned long uldata);

```

```

/*****
****
功能描述：主函数程序入口
入口参数： none
返回值： none
其他说明： none
*****/

void main()
{
    uint8 idata nAsrRes;
    uint8 i=0;

    MCU_init();
    LD_Reset();
    UartIni();//串口初始化
    nAsrStatus = LD_ASR_NONE; //初始状态：没有在作 ASR
    PrintCom("串行运动中...\n"); /*text.....*/
    PrintCom("口令： 1、代码测试\n"); /*text.....*/
    PrintCom(" 2、开发板验证\n"); /*text.....*/
    PrintCom(" 3 检测完毕\n"); /*text.....*/

    while(1)
    {
        switch(nAsrStatus)
        {
            case LD_ASR_RUNING:
            case LD_ASR_ERROR:
                break;
            case LD_ASR_NONE:
            {
                nAsrStatus=LD_ASR_RUNING;
                if (RunASR()==0) /*
                {
                    nAsrStatus = LD_ASR_ERROR;

```

```

    }
    break;
}

case LD_ASR_FOUNDOK: /*
{
    nAsrRes = LD_GetResult();
    PrintCom("\n 识别码");
    UARTSendByte(nAsrRes+0x30);
    switch(nAsrRes)
    {
        case CODE_DMCS:
            PrintCom("代码测试\n");
            break;

        case CODE_KFBYZ:
            PrintCom("\n"); /
            break;

        case CODE_JCWB:

            PrintCom("\n");
            break;

        default:break;
    }

    nAsrStatus = LD_ASR_NONE;
    break;
}
case LD_ASR_FOUNZERO:
default:
{
    nAsrStatus = LD_ASR_NONE;
    break;
}
} // switch
} // while

```

```

}

/*****
****
功能描述：单片机初始化
入口参数：
返回值：
其他说明：
*****/

void MCU_init()
{
    P0 = 0xff;
    P1 = 0xff;
    P2 = 0xff;
    P3 = 0xf7;

#ifdef (SOFT_SPI_PORT)
    LD_MODE = 1;
#elif defined (HARD_SPI_PORT)
    LD_MODE = 1;
#else
    LD_MODE = 0;
#endif
    IE0=1;
    EX0=1;
    EA=1;
}

/*****
****
功能描述：运行 ASR 识别流程
入口参数： none
返回值： asrflag: 1->启动成功，0->启动失败
其他说明： 识别顺序如下：

```


- 1、 RunASR()函数实现了一次完整的 ASR 语音识别流程
- 2、 LD-AsrStart()函数实现了添加关键词语到 LD3320 芯片中
- 3、 LD-AsrAddFixed()函数实现了田间关键词语到 LD3320

```

*****
*****/
void delay(unsigned long uldata)
{
    unsigned int j = 0;
    unsigned int g = 0;
    for (j=0;j<5;j++)
    {
        for (g=0;g<uldata;g++)
        {
            _nop_();
            _nop_();
            _nop_();
        }
    }
}
/*****
****
*****
*****/
uint8 RunASR()
{
    uint8 i=0;
    uint8 asrflag=0;
    for (i=0; i<5; i++)
    {
        LD_AsrStart();
        delay(100);
        if (LD_AsrAddFixed()==0)
        {
            LD_Reset();
            delay(100);
        }
    }
}

```

```

        continue;
    }
    delay(10);
    if (LD_AsrRun() == 0)
    {
        LD_Reset();
        delay(100);
        continue;
    }
    asrflag=1;
    break;
}

return asrflag;
}
/*****
****
功能描述： 中断处置函数
入口参数：
返回值：
其他说明：
*****/
void ExtInt0Handler(void) interrupt 0
{
    ProcessInt0();
    PrintCom("/n");
}
#ifdef HARD_PARA_PORT

#define LD_INDEX_PORT      (*((volatile unsigned char
xdata*)(0x8100)))
#define LD_DATA_PORT      (*((volatile unsigned char
xdata*)(0x8000)))

void LD_WriteReg( unsigned char address, unsigned char dataout )

```

```
{
    LD_INDEX_PORT = address;
    LD_DATA_PORT = dataout;
}
unsigned char LD_ReadReg( unsigned char address )
{
    LD_INDEX_PORT = address;
    return (unsigned char)LD_DATA_PORT;
}
#endif
#ifdef SOFT_PARA_PORT

#define DELAY_NOP    _nop();_nop();_nop();_nop();_nop();
sbit LD_WR = P3^6;
sbit LD_RD = P3^7;
sbit LD_CS = P2^1;
sbit LD_A0 = P2^0;
void LD_WriteReg( unsigned char address, unsigned char dataout )
{
    P0 = address;
    LD_A0 = 1;
    LD_CS = 0;
    LD_WR = 0;
    DELAY_NOP;

    LD_WR = 1;
    LD_CS = 1;
    DELAY_NOP;

    P0 = dataout;
    LD_A0 = 0;
    LD_CS = 0;
    LD_WR = 0;
    DELAY_NOP;
```

```

        LD_WR = 1;
        LD_CS = 1;
        DELAY_NOP;
    }

unsigned char LD_ReadReg( unsigned char address )
{
    unsigned char datain;

    P0 = address;
    LD_A0 = 1;
    LD_CS = 0;
    LD_WR = 0;
    DELAY_NOP;

    LD_WR = 1;
    LD_CS = 1;
    DELAY_NOP;

    LD_A0 = 0;
    LD_CS = 0;
    LD_RD = 0;
    DELAY_NOP;

    datain = P0;
    LD_RD = 1;
    LD_CS = 1;
    DELAY_NOP;

    return datain;
}
#endif
#ifdef SOFT_SPI_PORT
#define DELAY_NOP    _nop();_nop();_nop();

```

```
sbit SCS=P2^1;
sbit SDCK=P0^2;
sbit SDI=P0^0;
sbit SDO=P0^1;
sbit SPIS=P3^5;

void LD_WriteReg(unsigned char address,unsigned char dataout)
{
    unsigned char i = 0;
    unsigned char command=0x04;
    SPIS =0;
    SCS = 0;
    DELAY_NOP;

    //write command
    for (i=0;i < 8; i++)
    {
        if ((command & 0x80) == 0x80)
            SDI = 1;
        else
            SDI = 0;

        DELAY_NOP;
        SDCK = 0;
        command = (command << 1);
        DELAY_NOP;
        SDCK = 1;
    }
    //write address
    for (i=0;i < 8; i++)
    {
        if ((address & 0x80) == 0x80)
            SDI = 1;
        else
            SDI = 0;
```

```

        DELAY_NOP;
        SDCK = 0;
        address = (address << 1);
        DELAY_NOP;
        SDCK = 1;
    }
    //write data
    for (i=0;i < 8; i++)
    {
        if ((dataout & 0x80) == 0x80)
            SDI = 1;
        else
            SDI = 0;
        DELAY_NOP;
        SDCK = 0;
        dataout = (dataout << 1);
        DELAY_NOP;
        SDCK = 1;
    }
    DELAY_NOP;
    SCS = 1;
}

unsigned char LD_ReadReg(unsigned char address)
{
    unsigned char i = 0;
    unsigned char datain = 0;
    unsigned char temp = 0;
    unsigned char command = 0x05;
    SPIS = 0;
    SCS = 0;
    DELAY_NOP;

    //write command
    for (i=0;i < 8; i++)

```

```
{
    if ((command & 0x80) == 0x80)
        SDI = 1;
    else
        SDI = 0;
    DELAY_NOP;
    SDCK = 0;
    command = (command << 1);
    DELAY_NOP;
    SDCK = 1;
}
```

```
//write address
for (i=0;i < 8; i++)
{
    if ((address & 0x80) == 0x80)
        SDI = 1;
    else
        SDI = 0;
    DELAY_NOP;
    SDCK = 0;
    address = (address << 1);
    DELAY_NOP;
    SDCK = 1;
}
DELAY_NOP;
```

```
//Read data
for (i=0;i < 8; i++)
{
    datain = (datain << 1);
    temp = SDO;
    DELAY_NOP;
    SDCK = 0;
    if (temp == 1)
```

```

        datain |= 0x01;
        DELAY_NOP;
        SDCK = 1;
    }

    DELAY_NOP;
    SCS = 1;
    return datain;
}

#endif

void ProcessInt0(void);

/*****
*****
*****/
void LD_Reset()
{
    RSTB=1;
    delay(1);
    RSTB=0;
    delay(1);
    RSTB=1;

    delay(1);
    CSB=0;
    delay(1);
    CSB=1;
    delay(1);
}

/*****
*****
*****/

```



```
*****/
void LD_Init_Common()
{
    LD_ReadReg(0x06);
    LD_WriteReg(0x17, 0x35);
    delay(10);
    LD_ReadReg(0x06);

    LD_WriteReg(0x89, 0x03);
    delay(5);
    LD_WriteReg(0xCF, 0x43);
    delay(5);
    LD_WriteReg(0xCB, 0x02);

    /*PLL setting*/
    LD_WriteReg(0x11, LD_PLL_11);

    LD_WriteReg(0x1E, 0x00);
    LD_WriteReg(0x19, LD_PLL_ASR_19);
    LD_WriteReg(0x1B, LD_PLL_ASR_1B);
    LD_WriteReg(0x1D, LD_PLL_ASR_1D);
    delay(10);

    LD_WriteReg(0xCD, 0x04);
    LD_WriteReg(0x17, 0x4c);
    delay(5);
    LD_WriteReg(0xB9, 0x00);
    LD_WriteReg(0xCF, 0x4F);
    LD_WriteReg(0x6F, 0xFF);
}

/*****
****
*****/
```

```

void LD_Init_ASRO()
{
    LD_Init_Common();

    LD_WriteReg(0xBD, 0x00);
    LD_WriteReg(0x17, 0x48);
    delay( 10 );

    LD_WriteReg(0x3C, 0x80);
    LD_WriteReg(0x3E, 0x07);
    LD_WriteReg(0x38, 0xff);
    LD_WriteReg(0x3A, 0x07);

    LD_WriteReg(0x40, 0);
    LD_WriteReg(0x42, 8);
    LD_WriteReg(0x44, 0);
    LD_WriteReg(0x46, 8);
    delay( 1 );
}

/*****
****
*****/

void ProcessInt0(void)
{
    uint8 nAsrResCount=0;

    EX0=0;
    ucRegVal = LD_ReadReg(0x2B);
    LD_WriteReg(0x29,0) ;
    LD_WriteReg(0x02,0) ;
    if((ucRegVal & 0x10) &&
        LD_ReadReg(0xb2)==0x21 &&
        LD_ReadReg(0xbf)==0x35)          /*Ê±³£¼¶*/

```

```

{
    nAsrResCount = LD_ReadReg(0xba);
    if(nAsrResCount>0 && nAsrResCount<=4)
    {
        nAsrStatus=LD_ASR_FOUNDDOK;
    }
    else
    {
        nAsrStatus=LD_ASR_FOUNDDZERO;
    }
}
/*Ã»ÓÐÊ¶±ð½á¹û*/
else
{
    nAsrStatus=LD_ASR_FOUNDDZERO;
}

LD_WriteReg(0x2b, 0);
LD_WriteReg(0x1C,0);/*Ð'0:ADC²»¿ÉÓÃ*/

LD_WriteReg(0x29,0) ;
LD_WriteReg(0x02,0) ;
LD_WriteReg(0x2B, 0);
LD_WriteReg(0xBA, 0);
LD_WriteReg(0xBC,0);
LD_WriteReg(0x08,1); /*Çâ³ýFIFO_DATA*/
LD_WriteReg(0x08,0); /*Çâ³ýFIFO_DATA°ó ÔÙ'ÎÐ'0*/

EX0=1;
}

/*****
****
*****/

```

```
uint8 LD_Check_ASRBusyFlag_b2()
{
    uint8 j;
    uint8 flag = 0;
    for (j=0; j<10; j++)
    {
        if (LD_ReadReg(0xb2) == 0x21)
        {
            flag = 1;
            break;
        }
        delay(10);
    }
    return flag;
}

/*****
****
*****/

void LD_AsrStart()
{
    LD_Init_ASR();
}

/*****
****
*****/

uint8 LD_AsrRun()
{
    EX0=0;
    LD_WriteReg(0x35, MIC_VOL);
    LD_WriteReg(0x1C, 0x09);
    LD_WriteReg(0xBD, 0x20);
    LD_WriteReg(0x08, 0x01);
    delay( 1 );
}
```

```

LD_WriteReg(0x08, 0x00);
delay( 1 );

if(LD_Check_ASRBusyFlag_b2() == 0)
{
    return 0;
}
LD_WriteReg(0x1C, 0x0b);
LD_WriteReg(0xB2, 0xff);
delay( 1);
LD_WriteReg(0x37, 0x06);
delay( 1 );
LD_WriteReg(0x37, 0x06);
delay( 5 );
LD_WriteReg(0x29, 0x10);
LD_WriteReg(0xBD, 0x00);
EX0=1;
return 1;
}
/*****
****
*****/
uint8 LD_AsrAddFixed()
{
    uint8 k, flag;
    uint8 nAsrAddLength;
    uint8 code sRecog[4][20] = {"", "kai fa ban yan zheng", "dai ma ce shi", "jian cha
wan bi"};
    uint8 code pCode[4] = {0, CODE_KFBYZ, CODE_DMCS, CODE_JCWB};
    flag = 1;
    for (k=0; k<4; k++)
    {
        if(LD_Check_ASRBusyFlag_b2() == 0)
        {

```

```

        flag = 0;
        break;
    }
    LD_WriteReg(0xc1, pCode[k] );
    LD_WriteReg(0xc3, 0 );
    LD_WriteReg(0x08, 0x04);
    delay(1);
    LD_WriteReg(0x08, 0x00);
    delay(1);
    for (nAsrAddLength=0; nAsrAddLength<20; nAsrAddLength++)
    {
        if (sRecog[k][nAsrAddLength] == 0)
            break;
        LD_WriteReg(0x5, sRecog[k][nAsrAddLength]);
    }
    LD_WriteReg(0xb9, nAsrAddLength);
    LD_WriteReg(0xb2, 0xff);
    LD_WriteReg(0x37, 0x04);
}
return flag;
}

/*****
****
*****/

uint8 LD_GetResult()
{
    return LD_ReadReg(0xc5 );
}

```

外文资料

Speed Control System Based on Improved Fuzzy-PID hybrid Control for Direct Current Motor

Abstract—This paper describes a new hybrid speed regulating method based on improved Fuzzy-PID, which is used in the speed loop for speed and current closed loop control direct current (DC) motor control system. The specific way of the hybrid method depends on the value of the error between reference speed and actual speed, when it is bigger than threshold value, the fuzzy control is used, when it reduces to below the threshold value, the PID adjustment is used. The simulation research on the new hybrid speed regulating system of DC motor is carried out. The results of simulation show that the system based on improved fuzzy-PID hybrid control has better dynamic and static performance than the system based on conventional fuzzy-PID hybrid control. A new hybrid speed control system of DJ23 DC motor of DDSZ-1 electrical machinery and electrical technology test installation is designed. The system based on this new hybrid speed regulating method has a control unit of the single-chip Atmega128. Controlling the PWM duty cycle is equivalent to controlling the motor armature voltage, which in turn adjusts the motor speed. The results of the system experiment show that the system is of better robustness and stronger practicality.

Keywords- DC Motor; improved Fuzzy-PID; hybrid control; speed regulating

I. INTRODUCTION

DC motor Speed control system is a complex system with multi-variable, strong coupling, nonlinear and time-varying. The traditional PI or PID control can not meet the requirements of dynamic response, high-performance and accuracy[1-4]. With the advantages of being independence on mathematical models, better robustness and simplicity in controlling, fuzzy controller is quite popular in motor control field [5-8].

Various parameter self-adjusting fuzzy logic PID control methods of design and simulation for DC Motor have been proposed[5-6], which optimize PI or PID controller parameters by fuzzy control to improve the precision of system, the method can meet the practical requirements of production, but is complex to control. Some fuzzy control methods of design and simulation for DC Motor have been introduced[7-8], while the general fuzzy control is equal to PID, does not have the Integral, and is difficult to eliminate the static error completely. There also will be a small oscillation at the equilibrium point when the number of variable classification is small [5-6].

Considering the high precision and better steady performance of PID control, the easiness of realization and independence on the mathematical model of fuzzy control, a new hybrid speed regulating method based on improved Fuzzy-PID is introduced designed in this paper. This new hybrid speed control method is applied to DJ23 DC motor of DDSZ-1 electrical machinery and electrical technology test installation which is produced by Hangzhou Tianhuang Technology Industry Co., Ltd. The feasibility of this DC motor hybrid control system is proved by the laboratory experiments.

II. IMPROVED FUZZY-PID HYBRID CONTROL

A. Hybrid control theory

The main control mode of DC motor Speed control system is cascade control, which contains speed loop (external ring) and current loop (internal ring), changing the armature voltage to control speed. The actual speed and armature current are often used as the feedback variable in the system[5].

In this design, DC motor control system is a speed and current closed loop control system, the PI adjustment is used in the current loop and the improved fuzzy-PID hybrid control is used in speed loop. The specific way of the improved fuzzy-PID hybrid method in speed loop depends on the value of the error between reference speed and actual speed. When it is bigger than threshold value, the fuzzy control is used, when the error reduces to below the threshold value, the PID adjustment is used. The improved fuzzy-PID hybrid Control structure for DC motor is shown in Fig.1, where $K=20\text{r/min}$.

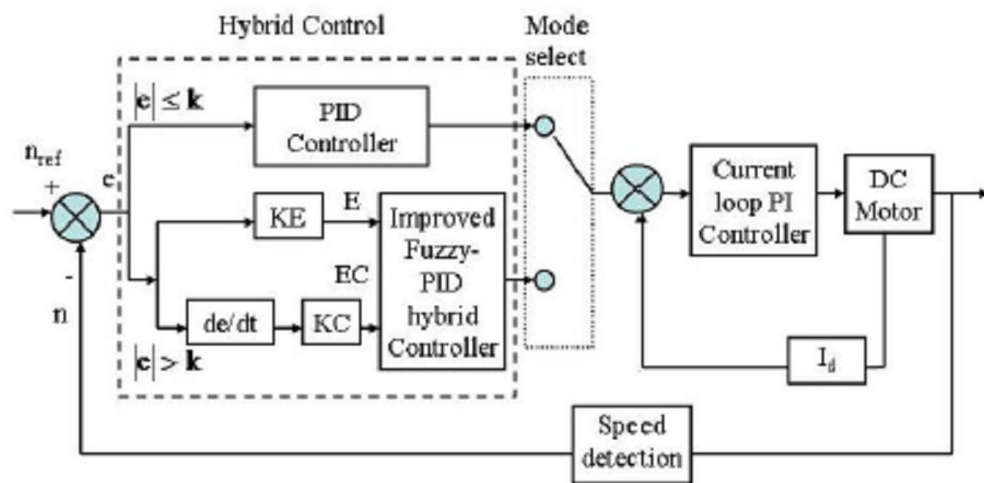


Figure 1. Fuzzy-PID hybrid control structure for DC motor

B. Fuzzy-PID hybrid Control of speed loop

If $|e| > k (|e| > 20 \text{ r/min})$, the fuzzy control is used in speed loop for speed and current closed loop control system.

A contrast study on conventional fuzzy control method and intelligent weight function fuzzy Control method[9] is carried out here.

The inputs of fuzzy controller are E and EC, the output is U. Set the domain of quantification $E = \{-9, 9\}$, $EC = \{-1, 0, 1\}$, $U = \{-9, 9\}$. The fuzzy language subset of E、EC and U are { NB(negative big), NM(negative middle), NS(negative small), ZO(zero), PS (positive small), PM(positive middle), PB(positive big) }.

(1) Conventional FuzzyControl

Based on the variable language quantization level and the practical experience of expert, the fuzzy control inference rules are built as Table 1.

TABLE 1. INFERENCE RULES

U \ EC								
	NB	NM	NS	ZO	PS	PM	PB	
E \								
NB	PB	PB	PB	PB	PM	PS	ZO	
NM	PB	PB	PB	PB	PM	ZO	ZO	
NS	PM	PM	PM	PM	PS	ZO	NS	
ZO	PM	PM	PS	ZO	NS	NS	NM	
PS	PS	PS	NS	NM	NM	NM	NM	
PM	ZO	ZO	NM	NB	NB	NB	NB	
PB	ZO	ZO	NM	NB	NB	NB	NB	

(2) Intelligent Weight Function Fuzzy Control

The objective of control method is to adjust the inference rules to improve the performance in a real time application. The inference rule is described as (1).

$$u = \frac{|E|}{|E| + |EC|} E + \frac{|EC|}{|E| + |EC|} EC \quad (1)$$

Where U is the output control variables, E and EC are the values of error and error cycle after fuzzy processing.

The typical characteristics of the control algorithm is that weight function is only related to the input variables, and control system can adjust the weight function of the control rules automatically on the basis of the value of the input variables, the control rules can then be adjusted online. The method does not require membership functions through a complex transformation to calculate the control rules, but only needs to transform the input error and the error cycle from the basic domain into the fuzzy domain, then calculate the weight function of error and error cycle, and finally you can calculate the fuzzy control output by equation (1), but this output is still fuzzy

output variable, which still needs to be transformed into the actual control variable through the scale factor.

C. Simulation of fuzzy-PID Hybrid Control

Hybrid control system of DC motors is simulated by writing M-file of Matlab, which aims to carry out a contrast study on the capabilities of conventional fuzzy-PID hybrid control and the improved fuzzy-PID hybrid control in speed loop.

(1) Speed and load Variation simulation of conventional fuzzy-PID hybrid control

In the simulation test, stair case speed transients from 300 to 750 r/min at the 2nd second, then stair case speed transients from 750 to 1350 r/min at the third second, and load torque transients from 0 to 10 % rated load torque at the first second. The simulation result of the control system based on conventional fuzzy-PID is shown in Fig. 2, where 'nref' is the reference speed, and 'n' is the actual speed.

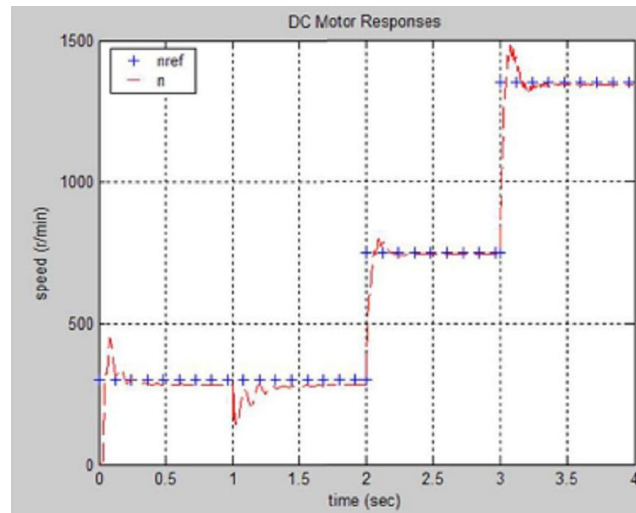


Figure 2 The response of conventional fuzzy-PID hybrid control system

(2) Speed and load Speed Variation of improved fuzzy-PID hybrid control

Under the same simulation conditions, the simulation result of the control system based on improved fuzzy-PID is shown in Fig. 3.

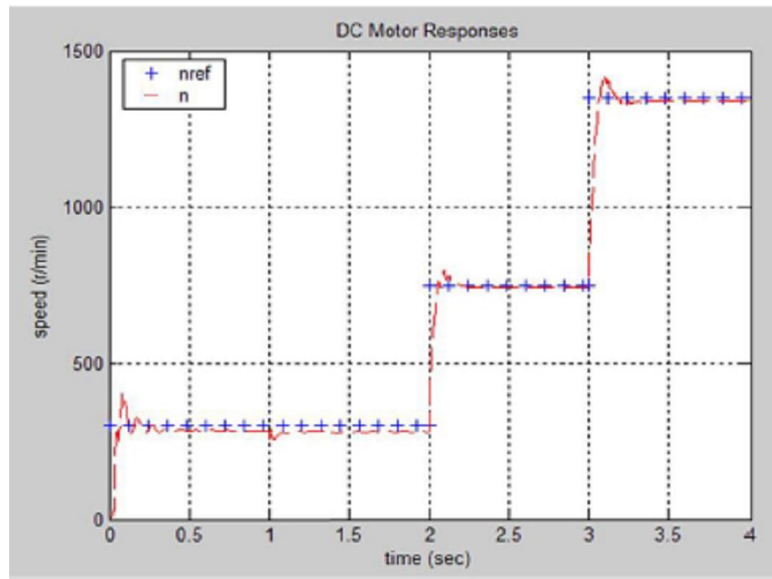


Figure 3. The response of improved fuzzy-PID hybrid control system

The two simulation results indicate that system based on improved fuzzy-PID hybrid control has better performance ,such as the small overshoot, the short time of reaching steady,and the improved dynamic performance, robustness.

III. DC MOTOR HYBRID CONTROL SYSTEM

A. The composition of dc motor hybrid control System

This system based on improved fuzzy-PID hybrid control produces PWM signal from Atmega128. The control object is a DJ23 DC motor of DDSZ-1 electrical machinery and electrical technology test installation. The motor parameters are: $U_N=220V$, $I_N=2.2A$, $P_N=355W$, $n_N=1500r/min$. This speed control system has some functions such as speed control, forward and reverse rotation control, brake control. The system composition is described as Fig. 4.

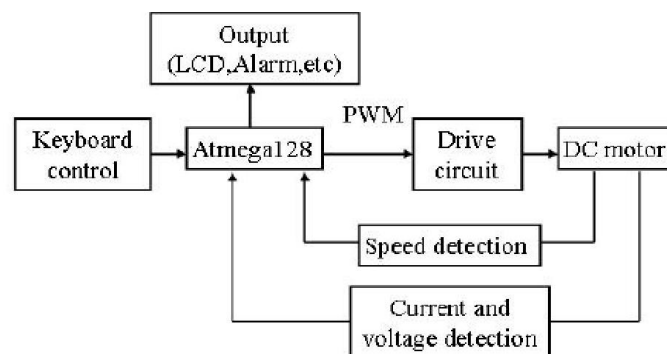


Figure 4. Control System composition

B. Software design

The software consists of main program, interrupt service program and other related subprograms.

The important interrupt program is A / D interrupt program of speed control, the flow chart is shown in Fig. 5.

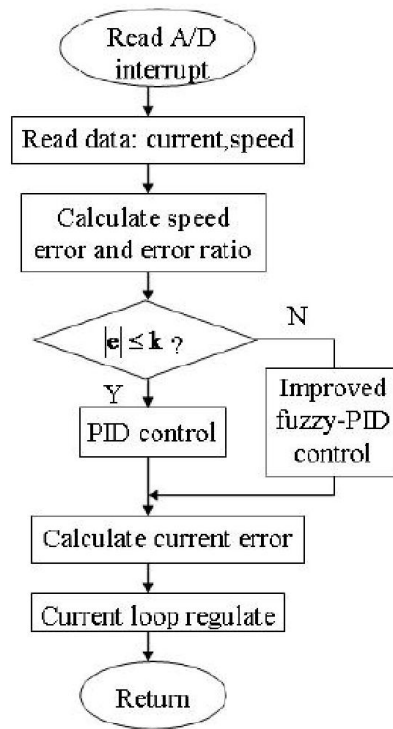


Figure 5. Flow chart of interrupt program of speed control

C. System experiment

(1) PWM waveform measurement

a. Instruments: digital oscilloscope, digital multimeter, 220V AC power, speed control system

b. The speed of DC motor can be controlled by adjusting the duty cycle of the pulse-width modulation (PWM), so it is necessary to make a test for PWM waveforms before the whole machine test. Oscilloscope is connected to the D5 pin of single chip computer, waveform duty cycle is regulated by the key, and different waveform data is recorded. As shown in Fig.6, when numerical value of liquid crystal display(LCD) is 512, the duty cycle of waveform oscilloscope display is 50%. “ 512 ” – the value of the Atmega128 0CR1A output compare register can vary from 0 to 1024, and represents the high level portion in the whole cycle.

$$\text{Duty cycle} = \frac{512}{1024} \times 100\% = 50\%$$

The PWM waveform test results are good.

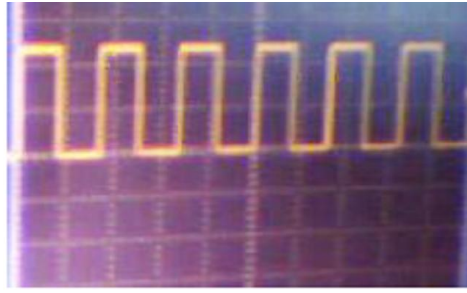


Figure 6. PWM waveform when LCD display is 512

(2)DC motor control system experiment

a. Test instruments: 355W DC motor, digital multimeter, 220V DC power, 20V AC power, speed control system

b. Test procedure: The control circuit board, drive circuit board and DC motor are properly connected in the motor testing Room. Connect excitation system to 220V DC .

In order to test the stability of the system, the motor speed should be changed manually. The motor speed follows the change of the duty cycle, that is to say, the system is stable.

In the automatic control stage, enter the reference speed, press the 'OK' key, the system can automatically adjust the duty cycle and change the motor speed. when the error between reference speed and actual speed is below 3r/min ,the system gives success sign. Fig. 7 is experimental system of DC motor.

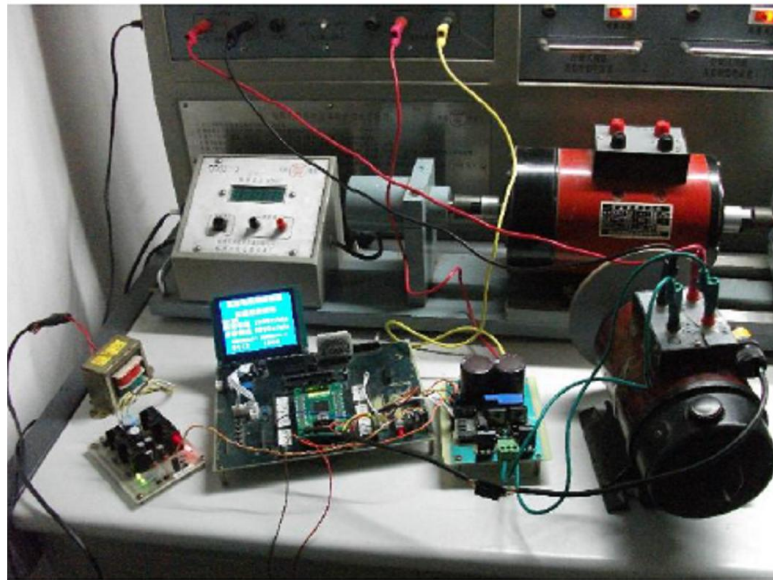


Figure 7. Experimental system of DC motor

Fast response, stable operation and good robustness are shown in the experiment .

Some results of experiment are given in table 2. Table 2 shows that the error between reference and actual speed is small.

TABLE 2 EXPERIMENT RESULTS

Test number	Reference speed (r/min)	Actual speed (r/min)
Test 1	1000	998
Test 2	1300	1300
Test 3	600	599

IV. CONCLUSION

In this paper, a new hybrid speed control method based on improved fuzzy-PID control is applied to the speed loop for speed and current closed loop control DC motor control system. The simulation results show that the method based on this hybrid speed regulating has better dynamic and more static performance than the method based on conventional fuzzy-PID hybrid control. A DC motor speed control system based on this improved fuzzy-PID method is introduced. The results of experiment show that the system is of better robustness and stronger practicality.

中文翻译

基于改进的模糊-PID 复合控制直流电机调速系统

摘要：本文介绍了一种基于改进的模糊 PID，这是在速度循环控制速度和电流闭环控制直流（DC）电机控制系统的一种新混合速度调节方法。混合方法的具体方式取决于参考速度和实际速度之间的误差的值，当它大于阈值使用模糊控制，当它降低到低于该阈值时，PID 调节被使用。仿真研究在直流电动机的新型混合调速系统上进行。仿真结果表明，基于改进的模糊 PID 混合控制系统比基于传统模糊 PID 混合控制系统具有更好的动态和静态性能。DDSZ - 1 电机及电气技术实验装置的 DJ23 直流电动机的一种新的混合速度控制系统是基于这种新的混合速度调节方法设计的。该系统由 Atmega128 芯片作为主控制单元。控制 PWM 占空比，相当于控制电动机的电枢电压，从而调节系统实验的马达的速度。结果表明，该系统具有更好的鲁棒性和更强的实用性。

关键词：直流电机;改进模糊 PID;混合控制;调速

I. 引言

直流电动机调速系统是一个复杂的系统，具有多变量，强耦合，非线性和时变性。传统的 PI 或 PID 控制不能满足动态响应的要求，高性能和精度[1-4]。由于模糊控制具有独立的数学模型，较好的鲁棒性和简单的控制优势，因此模糊控制器在电机控制领域[5-8]受到了广泛的应用。

直流电机的各种参数自调整模糊 PID 控制方法的设计与仿真被提出[5-6]，它通过模糊控制优化 PI 或 PID 控制器参数，以提高系统的精度，该方法能满足实际需求生产的，但需要复杂的控制。设计和仿真的直流电机的一些模糊控制方法已经介绍[7-8]，而一般的模糊控制等于 PID 控制没有积分项，并且很难完全消除稳态误差。并且在时变量分类的数量很小时在平衡点也会产生一个小的振荡[5-6]。

由于 PID 控制的高精度和高稳定性能，所以其更容易实现和独立于数学模型，本文介绍了一种基于改进模糊 PID 的新型混合速度调节方法。这种新的混合速度控制方法被应用于由杭州天煌科技实业有限公司开发生产 DDSZ-1 电机及电气技术实验装置的 DJ23 直流电动机，作为此直流电动机混合动力控制系统证明方案的可行性。

II. 改进的模糊-PID复合控制

A. 混合控制理论

直流电动机调速系统的主要控制方式为级联控制，其中包含速度环（外环）和电流环（内环），通过改变电枢电压来实现速度控制，实际速度和电枢电流通常被用作控制系统的反馈变量[5]。

在此设计中，直流电动机的控制系统是一个速度和电流的闭环控制系统，PI 调节用于在电流回路和改进的模糊-PID 复合控制被用在速度环。在速度回路改进的模糊-PID 混合方法的具体方式取决于参考速度和实际速度。当其之间的误差值大于阈值的情况下使用模糊控制，当误差减小到低于阈值时 PID 调节被使用。改进的模糊 PID 混合控制结构的直流电动机如图 1 所示，其中 $K=20\text{r/min}$ 。

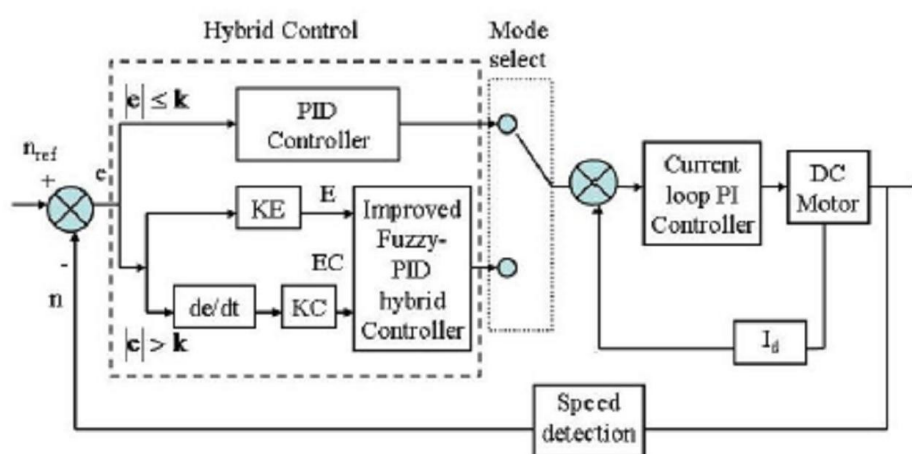


图 1 直流电机的模糊 PID 混合控制结构

B. 模糊 PID 混合控制速度环

如果 $|E| > K$ ($|E| > 20$ 转/分)，模糊控制采用的是速度循环控制的速度和电流闭环控制系统。在传统的模糊控制方法及智能权函数模糊控制方法的对比研究[9]进行了这里。模糊控制器的输入为 e 和 ec ，输出为 U 套装量化 E 的域 $=\{-9,9\}$ ， $EC = \{-1,0,1\}$ ， $U = \{-9,9\}$ 。 E ， EC 的模糊语言子集和 U 分别为 $\{NB$ （负大）， NM （负中）， NS （负小）， $Z0$ （零）， PS （正小）， PM （正中）， PB （正大） $\}$ 。

(1) 传统的模糊 PID 混合控制

基于变量量化的语言水平和专家的实践经验，模糊控制推理规则都是建立如表 1。

TABLE 1. INFERENCE RULES

U \ EC								
	NB	NM	NS	ZO	PS	PM	PB	
E								
	NB	PB	PB	PB	PB	PM	PS	ZO
	NM	PB	PB	PB	PB	PM	ZO	ZO
	NS	PM	PM	PM	PM	PS	ZO	NS
	ZO	PM	PM	PS	ZO	NS	NS	NM
	PS	PS	PS	NS	NM	NM	NM	NM
	PM	ZO	ZO	NM	NB	NB	NB	NB
	PB	ZO	ZO	NM	NB	NB	NB	NB

(2) 智能权函数模糊控制

控制方法的目的是调整推理规则，以改善其在实时系统中的应用表现。该推理规则的描述见下 (1)。

$$u = \frac{|E|}{|E| + |EC|} E + \frac{|EC|}{|E| + |EC|} EC \quad (1)$$

其中 U 是输出控制变量，E 和 EC 是误差和误差周期值模糊化处理后。

控制算法的典型特征是，权重函数是仅与输入变量相关，并且控制系统可以在输入变量的值的基础上自动调整控制规则的权重函数，控制规则可以进行在线调整。该方法不通过复杂的转换要求从属关系函数来计算控制规则，但仅需要从基本域变换的输入误差和误差周期到模糊域，然后计算误差和误差周期的权重函数，最后由方程 (1) 可以计算出模糊控制的输出，但该输出仍然是模糊输出变量，它仍然需要通过比例因子转化为实际的控制变量。

C. 模糊 PID 混合控制下的仿真

直流电动机的混合控制系统是通过 matlab 编写的 M 文件，其目的是对常规模糊 PID 混合控制和速度环的改进的模糊 PID 复合控制的功能进行对比研究。

(1) 常规模糊 PID 混合控制的速度和负载变化模拟

在模拟试验中，在第 2 秒时速度跳变从 300 转/分跳变至 750 转/分，然后在第 3 秒从 750 转/分跳变至 1350 转/分，在第 1 秒时负载转矩产生从 0 至 10% 的瞬变。基于传统的模糊 PID 控制系统的仿真结果示于图 2，其中‘NREF’是参考速度，和‘n’是实际速度。

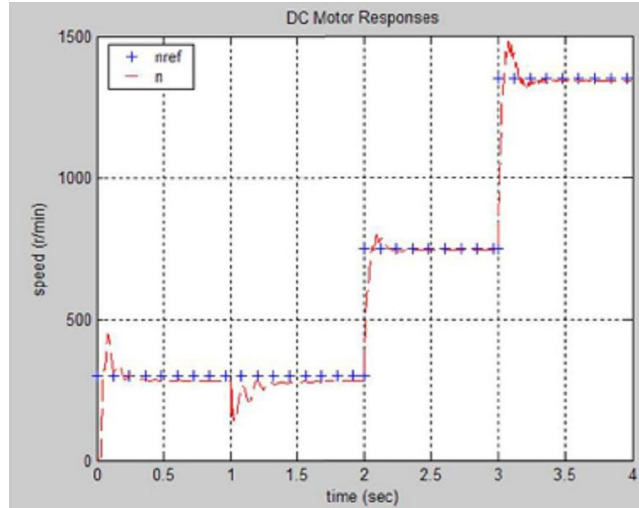


图2 传统的模糊 PID 混合控制系统的响应

(2) 速度和负载改进的模糊 PID 复合控制的速度变化

在相同的仿真条件下，基于改进的模糊 PID 控制系统的仿真结果示于图 3。

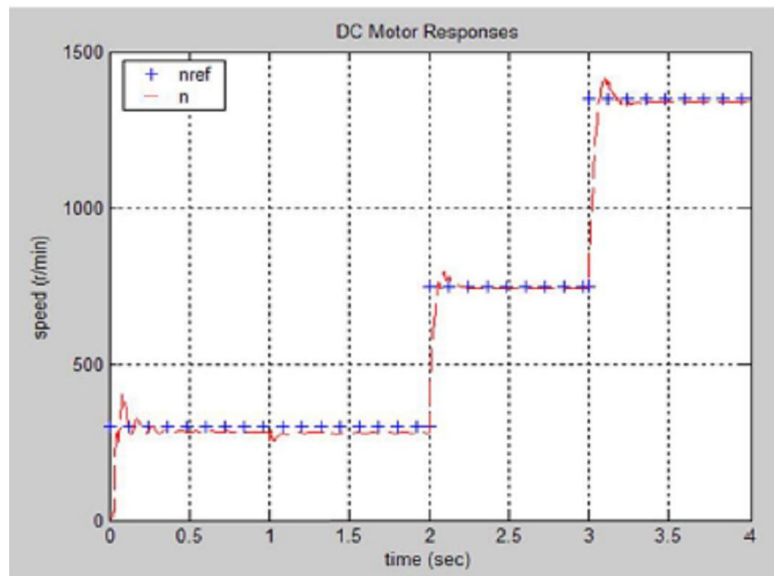


图3 改进的模糊-PID 复合控制系统的响应

这两个仿真结果表明，基于改进的模糊 PID 混合控制系统具有更好的性能，如超调量小，达到稳定的时间短，并且改善了动态性能和鲁棒性。

III. 直流电动机混合控制系统

A. 直流电动机混合控制系统的组成

基于改进的模糊 PID 混合控制该系统从 Atmega128 产生 PWM 信号。控制对象是 DDSZ-1 电机及电气技术实验装置的 DJ23 直流电动机。电机参数为：UN=220V，IN=2.2A，PN=355W，NN=1500r/min。该调速系统具有一定的功能，如速度控制，正反转控制，制动控制。该系统的组成被描述为图 4。

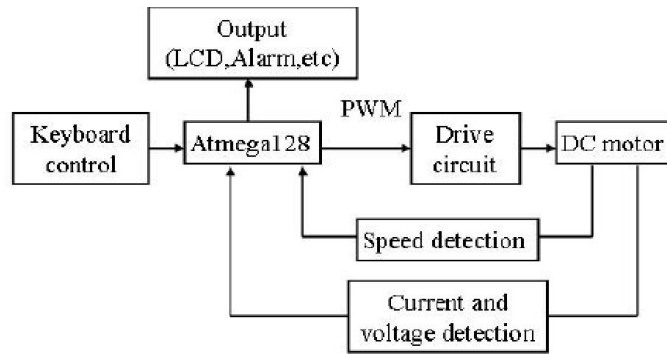


图 4 控制系统的组成

B. 软件设计

该软件由主程序，中断服务程序和其他相关子程序。最重要的中断程序是速度控制的 A/D 转换中断程序，流程图如图 5 所示。

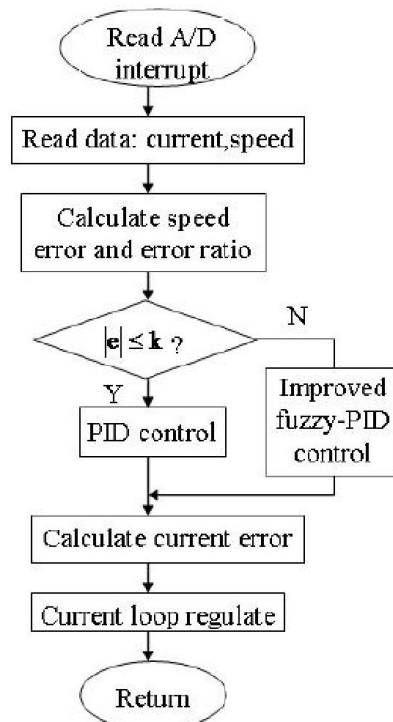


图 5 速度控制的中断程序的流程图

C. 系统实验

(1) PWM 波形的测量

a. 仪器仪表：数字示波器，数字万用表，220V 交流电源，速度控制系统

b. 直流电动机的转速可通过调节脉冲宽度调制（PWM）的占空比进行控制，因此，有必要在整机测试之前对 PWM 波形进行一个测试。示波器被连接到单芯片计算机的 D5 针，波形的占空比由按键调节，并且记录不同的波形数据。如图 6 所示，当液晶显示器（LCD）的数值是 512，波形的示波器显示的占空比为 50%。“512”- ATmega128 的 OCR1A 输出比较寄存器的值可以从 0 到 1024，并代表整个周期的高电平部分。

$$\text{占空比} = \frac{512}{1024} \times 100\% = 50\%$$

PWM 波形的测试结果不错。

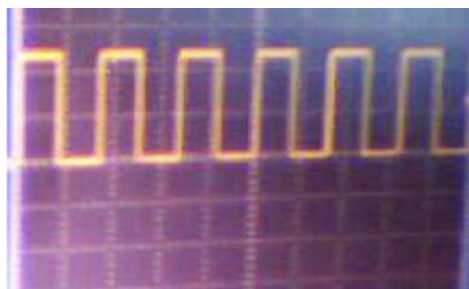


图 6 当液晶显示屏为 512 时的 PWM 波形

(3) 直流电机控制系统实验

- a. 测试工具：355W 直流电机，数字万用表，220V 直流电源，20V 交流电源，速度控制系统
- b. 测试步骤：控制电路板，驱动电路板和直流电动机的连接是否正确链接。励磁系统连接到 220V 直流。

为了测试该系统的稳定性，电机的转速应手动改变。电机的转速随着占空比的变化而变化，也就是说，该系统是稳定的。

在自动控制阶段，输入参考速度，按下“确定”键后，系统能自动调节占空比和改变电机的转速。当参考速度和实际速度之间的误差小于 3r/min，系统会给出成功的标志。图 7 是直流电动机的实验系统。

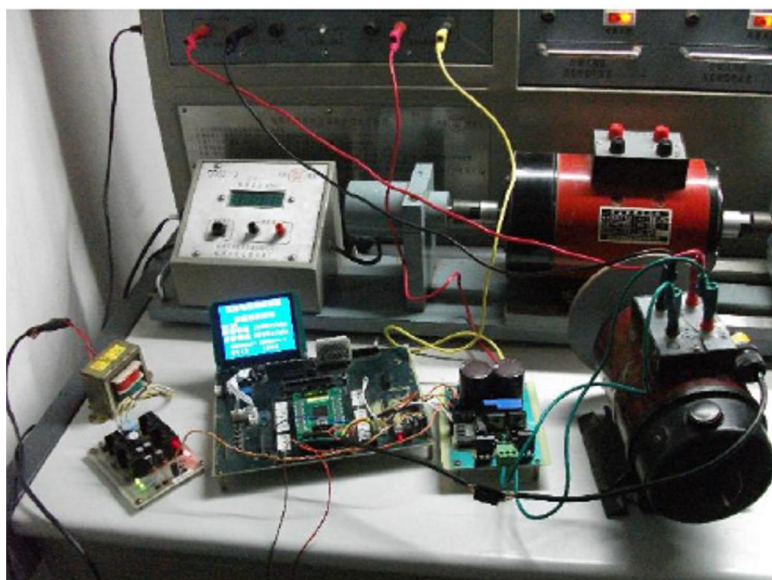


图 7 直流电动机的实验系统

实验显示了改进的模糊 PID 算法响应速度快，运行稳定，鲁棒性好。

实验的一些结果在表 2 中给出。表 2 表明参考速度与实际速度之间的误差较小。

TABLE 2 EXPERIMENT RESULTS

Test number	Reference speed (r/min)	Actual speed (r/min)
Test 1	1000	998
Test 2	1300	1300
Test 3	600	599

IV. 总结

在本文中,对于速度循环控制的速度和电流闭环控制直流电机系统采用了一种基于改进的模糊 PID 控制一种新型混合速度控制方法。仿真结果表明,基于这种混合调速方法比基于传统领域的模糊 PID 混合控制有更好的动态和静态的性能。介绍在此基础上改进的模糊 PID 方法的直流电机调速系统。实验结果表明,该系统是较好的鲁棒性和更强的实用性。

致 谢

随着本论文的完成，大学四年的学习生活即将划上圆满的句号。回首四年的大学生活，是在老师和同学们的教导和帮助下走过来的，在此过程中，我也学到了使我受益终生的人身哲理和知识。我大学的一切几乎都是实验室给我的！所以我也很庆幸我能在实验室度过我大学的一段时光。因此我要向一直关心和帮助我的老师和同学道一声谢谢！

衷心感谢吉建华老师在毕业设计期间给予的悉心指导和帮助！在毕业设计期间，吉建华老师严谨的作风一直影响着我们，并且在具体问题上给予我们指导，教会了我们很多知识，也让我们学会了做学问应有的态度和方法。

我还要感谢信息系的各位老师们在毕业设计期间所作的大量工作和给予的大量帮助，没有信息系的各位老师的帮助，本课题不会圆满完成。再次感谢各位老师们的所给予的帮助！

此外，还要感谢吉建华老师对于我论文撰写所提出的宝贵意见。

我还要感谢身边的同学，在我毕业设计期间对我极大的支持和鼓励。

另外，我也要感谢实验室一直以来对我学业的支持！