

# 智能舞蹈机器人设计与制作

作者：张建高

## 引言

30年前，比尔·盖茨毅然弃学，创立微软，成为个人电脑普及革命的领军人物；30年后的今天，他预言，机器人即将重复个人电脑崛起的道路。点燃机器人普及的“导火索”，这场革命必将与个人电脑一样，彻底改变这个时代的生活方式。

机器人作为人类20世纪最伟大的发明之一，在短短的几十年内发生了日新月异的变化。近几年机器人已成为高技术领域内具有代表性的战略目标。机器人技术的出现和发展，不但使传统的工业生产面貌发生根本性变化，而且将对人类社会产生深远的影响。随着社会生产技术的飞速发展，机器人的应用领域不断扩展。从自动化生产线到海洋资源的探索，乃至太空作业等领域，机器人可谓是无处不在。目前机器人已经走进人们的生活与工作，机器人已经在很多的领域代替着人类的劳动，发挥着越来越重要的作用，人们已经越来越离不开机器人帮助。机器人工程是一门复杂的学科，它集工程力学、机械制造、电子技术、技术科学、自动控制等为一体。目前对机器人的研究已经呈现出专业化和系统化，一些信息学、电子学方面的先进技术正越来越多地应用于机器人领域。

目前机器人行业的发展与30年前的电脑行业极为相似。今天在汽车装配线上忙碌的一线机器人，正是当年大型计算机的翻版。而机器人行业的利基产品也同样种类繁多，比如协助医生进行外科手术的机械臂、在伊拉克和阿富汗战场上负责排除路边炸弹的侦察机器人、以及负责清扫地板的家用机器人，还有不少参照人、狗、恐龙的样子制造机器人玩具。舞蹈机器人（如图1）具有人类外观特征、可爱的外貌、又兼有技术含量，极受青少年的喜爱。我从前年开始机器人方面的研究，在这过程中尝试过很多次的失败，也感受到了无比的乐趣。

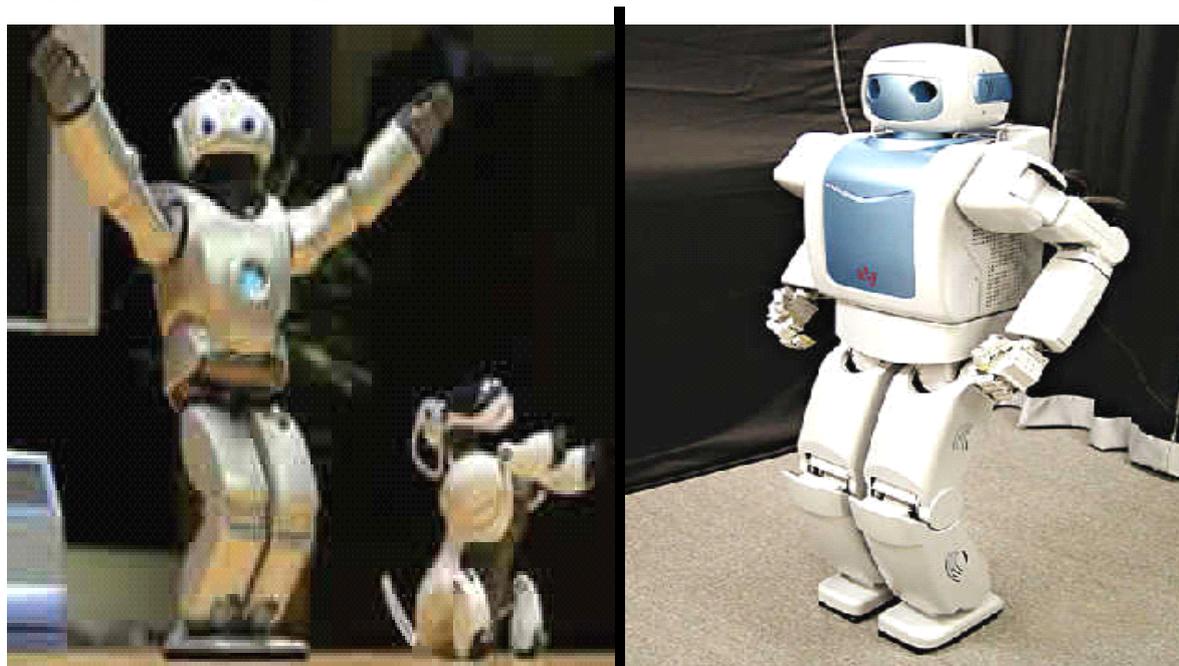


图 1.1、舞蹈机器人

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

## 1 绪论

机器人技术作为 20 世纪人类最伟大的发明之一，自 20 世纪 60 年代初问世以来，经历 40 余年的发展已取得长足的进步。未来的机器人是一种能够代替人类在非结构化环境下从事危险、复杂劳动的自动化机器，是集机械学、力学、电子学、生物学、控制论、计算机、人工智能和系统工程等多学科知识于一身的高新技术综合体。走向成熟的工业机器人，各种用途的特种机器人的多用化，昭示着机器人技术灿烂的明天。

### 1.1 国内外机器人技术发展的现状

为了使机器人能更好的应用于工业，各工业发达国家的大学、研究机构和大工业企业对机器人系统开发投入了大量的人力财力。在美国和加拿大，各主要大学都设有机器人研究室，麻省理工学院侧重于制造过程机器人系统的研究，卡耐基—梅隆机器人研究所侧重于挖掘机器人系统的研究，而斯坦福大学则着重于系统应用软件的开发。德国正研究开发“MOVE AND PLAY”机器人系统，使机器人操作就像人们操作录像机、开汽车一样。

从六十年代开始日本政府实施一系列扶植政策，使日本机器人产业迅速发展起来，经过短短的十几年。到 80 年代中期，已一跃而为“机器人王国”。其机器人的产量和安装的台数在国际上跃居首位。按照日本产业机器人工业会常务理事米本完二的说法：“日本机器人的发展经过了 60 年代的摇篮期。70 年代的实用期。到 80 年代进入普及提高期。”并正式把 1980 年定为产业机器人的普及元年”。开始在各个领域内广泛推广使用机器人。

中国机器人的发展起步较晚，1972 年我国开始研制自己的工业机器人。“七五”期间，国家投入资金，对工业机器人及其零部件进行攻关，完成了示教再现式工业机器人成套技术的开发，研制出了喷涂、点焊、弧焊和搬运机器人。1986 年国家高技术研究发展计划(863 计划)开始实施，智能机器人主题跟踪世界机器人技术的前沿，经过几年的研究，取得了一大批科研成果，成功地研制出了一批特种机器人。20 世纪 90 年代，我国的工业机器人又在实践中迈进一大步，先后研制出了点焊、装配、喷漆、切割、搬运等各种用途的工业机器人，并实施了一批机器人应用工程，形成了一批机器人产业化基地，为我国机器人产业的腾飞奠定了基础。

### 1.2 机器人技术的市场应用

机器人融入我们日常生活的步伐有多快？据国际机器人联盟调查，2004 年，全球个人机器人约有 200 万台，到 2008 年，还将有 700 万台机器人投入运行。按照韩国信息通信部的计划，到 2013 年，韩国每个家庭都能拥有一台机器人；而日本机器人协会预测，到 2025 年，全球机器人产业的“蛋糕”将达到每年 500 亿美元的规模(现在仅有 50 亿美元)。与 20 世纪 70 年代 PC 行业的情况相仿，我们不可能准确预测出究竟哪些用途将推动这个新兴行业进入临界状态。不过看起来，机器人很可能在护理和陪伴老年人的

工作上大展宏图，或许还可以帮助残疾人四处走走，并增强士兵、建筑工人和医护人员的体力与耐力。

目前，我国从事机器人研发和应用工程的单位 200 多家，拥有量为 3500 台左右，其中国产占 20%，其余都是从日本、美国、瑞典等 40 多个国家引进的。2000 年已生产各种类型工业机器人和系统 300 台套，机器人销售额 6.74 亿元，机器人产业对国民经济的年收益额为 47 亿元，我国对工业机器人的需求量和品种将逐年大幅度增加。

### 1.3 机器人技术的前景展望

机器人是人类的得力助手，能友好相处的可靠朋友，将来我们会看到人和机器人会存在一个空间里边，成为一个互相的助手和朋友。机器人会不会产生饭碗的问题。我们相信不会出现“机器人上岗，工人下岗”的局面，因为人们随着社会的发展，实际上把人们从繁重的体力和危险的环境中解放出来，使人们有更好的岗位去工作，去创造更好的精神财富和文化财富，机器人来做这些危险环境的工作，展望 21 世纪机器人将是一个与 20 世纪计算机的普及一样，会深入地应用到各个领域，所以很多专家预测，在 21 世纪的前 20 年是机器人从制造业走向非制造业的发展一个重要时期，也是智能机器人发展的一个关键时期，目前国际上很多国家，也对机器人对人类社会的影响的估计提出了新的认识，同时，我们也可以看到机器人技术，涉及到多个学科，机械、电工、自动控制、计算机测量、人工智能、传感技术等等，它是一个国家高技术实力的一个重要标准。

语音识别处理是语音功能的一个重要方面，目前计算机语音识别处理过程基本上一致，是一种基于统计模式识别的理论。我国的语音识别研究起步于五十年代，近年来发展很快，其研究水平基本与国外同步，在汉语语音识别技术上还有自己的特点和优势。可以预计，语音技术的发展前景无限。

## 2 舞蹈机器人设计的内容和要求

### 2.1 舞蹈机器人设计的内容

随着现代科技的发展，机器人技术已广泛应用于人类社会生活的各个领域，特别是舞蹈机器人具有人类外观特征、可爱的外貌、又兼有技术含量，极受青少年的喜爱。本课题要求设计一具有简单人体功能的、模拟舞蹈动作的类人型机器人，完成简单人体舞蹈的基本动作：可以前进后退，左右侧行，左右转弯和前后摆动手臂，行走频率为每秒两步，举手投足、转圈、头部动作灵活、并具备的语音功能。通过语音识别技术，可以对小机器人进行语音控制，通过发出语音命令，控制机器人的。机器人包括底座、头部、上身、下肢、以及电路控制板，分别控制手臂、头部和底盘运动的电机及传动机构等。通过电路控制和机械传动，可使机器人动作。知识范围涉及机构学、力学、电子学、自动控制、计算机、人工智能等。

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

具体任务：

- 1、深入了解类人型机器人的功能及工作要求，查找与课题有关的文献资料及参考书目；
- 2、学习掌握机构创新设计的基本知识和设计方法，了解控制对象舞蹈机器人的工作原理、动作过程，进行简单舞蹈动作及相应机构设计；
- 3、根据机器人构成、工作原理、主要特点和技术指标，分析比较，加以论证，确定舞蹈机器人运动控制最终方案，完成硬件电路设计，单片机控制程序设计；
- 4、制作舞蹈机器人模型，完成各种运动、动作模拟，调试成功。
- 5、规定的翻译、论文工作。

## 2.2 舞蹈机器人设计的数据和要求

- 1、机器人身高 80~120 cm，表演时机器人随音乐翩翩起舞，动作协调、灵活；
- 2、表演各种舞蹈的基本动作，具体动作可自行设计。涵盖行进、转圈、举手投足、头部等动作；
- 3、整套动作持续 4 分钟左右；
- 4、机器人结构紧凑、体积小，重量轻；
- 5、灵活的多功能机器手臂：活动空间大，可 360 度旋转。
- 6、采用电机驱动，运动准确可靠。

## 3 方案设计及方案论证

### 3.1 设计制作与调试流程

设计就是根据题目的要求而对硬件和软件进行规划，并选择最合适的硬件电路和软件程序来达到目的。

硬件设计是通过分析设计要求的分析，对各种元器件的了解，而得出分立元件与集成块的某些连接方法，以达到设计的功能要求。并且把这些元器件焊接在一块电路板上。它包括对各种元器件的功能和接法的了解，以及对各种元器件的选择和设计方案的选。软件设计是分析设计的硬件用程序实现其功能，并且调试优化产品功能。

机器人的设计首先需作总体方案设计，包括机械和电气两部分。根据机械基础知识，从整体上来讲，机械结构设计必须与机器人所要完成的功能相适应。机械部分设计主要包括底盘设计、尺寸选择、驱动方式设计、电机选择等；电气是机器人最重要的部分，直接影响着机器人功能的实现与否。舞蹈机器人要求完全自动控制，必须采用单片机为控制核心，它类似于机器人的大脑，接收和处理所有外界信息，指挥并控制机器人的所有动作。语音识别等功能是制作机器人硬件的难点，它要求机器人具有一定的感觉系统。目前，机器人技术已日趋成熟，机器人感觉系统可通过各种各样相应的传感器技术即可

实现。传感器把接收到的外部信息输入到单片机，再通过软件进行控制，从而单片机发出命令指挥机器人动作。软件编程可以丰富机器人的功能，使机器人动作更加完善。总体方案设计框图如图 3.1 所示。

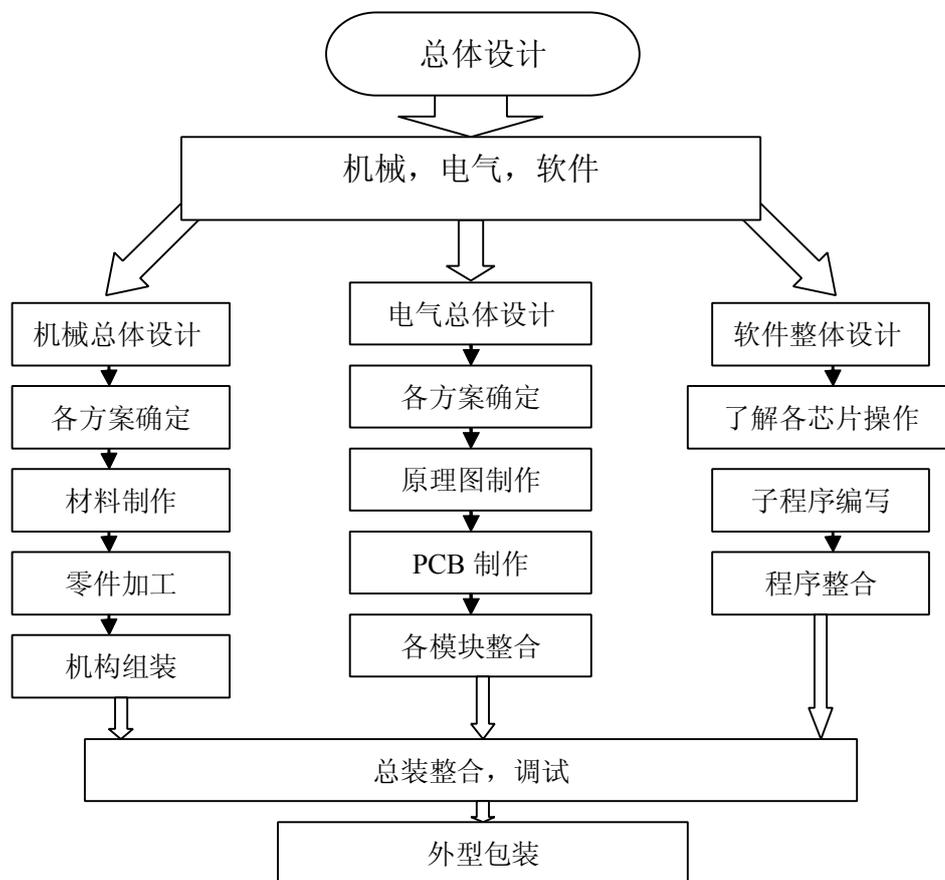


图3.1 机器人总体制作过程

### 3.2 方案的比较论证及方案确定

根据以上的制作流程图，我们把设计分为两大部分——机械部分和电气部分进行论证，对多种方案进行比较论证后，选出一个最佳的方案，最后再进行软件部分的设计。

#### 3.2.1 机械部分

机械部分包括：底盘机构的设计、弯腰机构的设计、转身机构的设计、手臂机构的设计、转头机构的设计、及材料与型材选择。

##### 3.2.1.1 底盘机构的设计

(1) 行走式简图如下

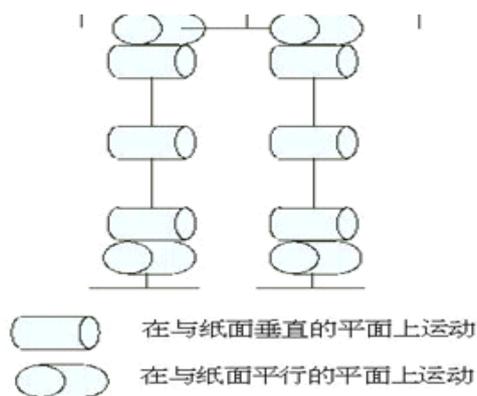


图 3.2 行走机器人简图

行走式，能够在平地上、凹凸不平的地上步行，能上下台阶，其粗糙路面性能较好，好的关节，稳定好的牵引力。它的缺点是不能原地转动，速度慢，控制复杂，移动困难，操纵性差，机械复杂，电机性能要求严格，价格昂贵。

(2) 移动方式为履带式简图如下

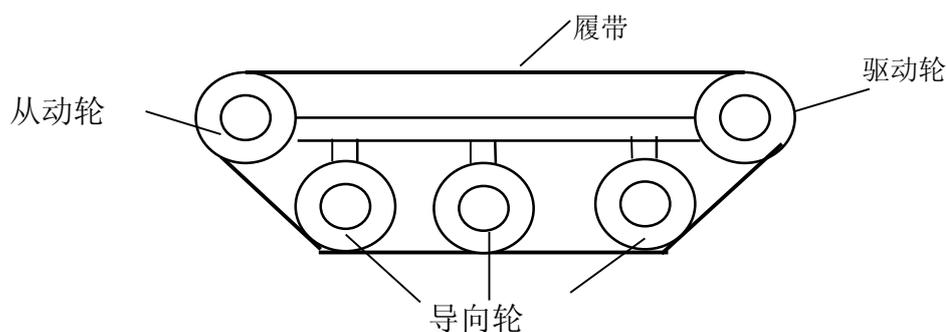


图 3.3 移动方式图

履带式移动方式的特点是运动平稳，适合爬坡度比较大的地方，这是轮式所做不的。但缺点是加工制作比较麻烦，履带一般是需要到市场上购买的，且摩擦力很大、能量损耗大

(3) 前轮偏转后轮驱动移动式简图如下

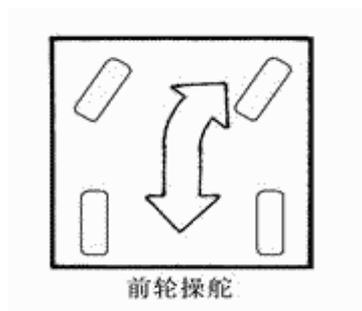


图 3.4 底盘移动方式

可以实现车体任意的定位和定向，可以做到零回转半径。特别是能通过将轮向正交

速度方向实现急停。车底盘、四轮、驱传动装置、舵机和控制盒的行走信道共同组成行走系统。但由于前后启动惯性不一样，导致机器人的不平衡，特别是重心高的机器人。

#### (4) 轮式差速驱动

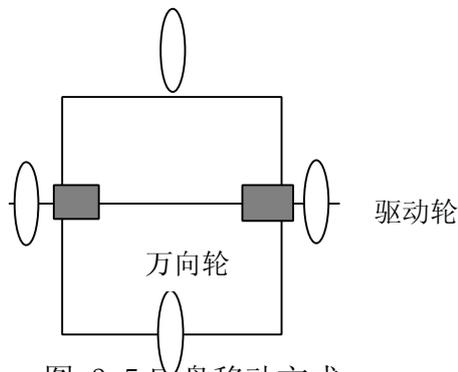


图 3.5 底盘移动方式

采用两个驱动轮，两个万向轮，万向轮用于平衡。机构简单，容易制造及装配；能在原地旋转以获得不同的方向。因此它比较适合的一种底盘机构。

#### 3.2.1.2 弯腰机构的设计

腰部的设计思路是，作为人形机器人的腰部，主要做要是承接了上部的重量，设计的时候腰部能做到水平的转动最好，如果不能将会极大地影响整个机器人的平衡；所以尽量减少可变的环节，故在腰部转动的设计中不采用电机再经传动其他机构最后带动腰部的转动方式，而是直接使用电机实现弯腰。

(1) 齿轮机构简图如下

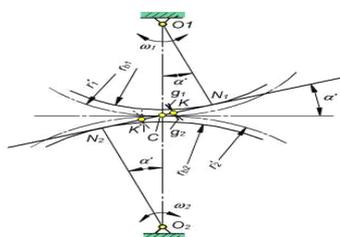


图 3.6 齿轮传动方式

齿轮机构的优点是平稳，自锁性好，精度高。但加工比较困难，且不利于固定，造价也很高。

(2) 连杆机构简图如下

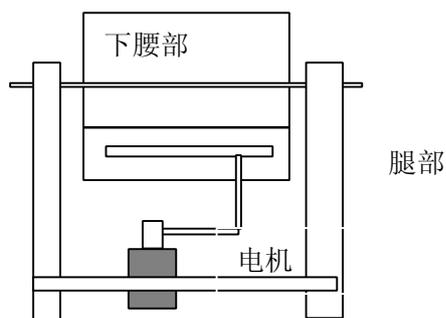


图 3.7 连杆传动方式

结构简单，容易实现，更重要的是它的运动规律正好符合人体运动的原理。我们的舞蹈机器人就是采用这样的机构来实现弯腰的。

(3) 连杆机构简图二

具体实现方式如下：先通过一根轴穿过鞠躬机构与机器人上半身的两只手臂固定，然后电机通过一个摇杆连接到弯腰机构的一个长方形槽中，在电机转动的过程中，带动鞠躬机构做以与手臂固定轴为中心的前后弯腰鞠躬动作，所需的零件包括一根轴，一个带有与轴相同大小的孔和在孔同一侧的开有长方形槽的铝条。并且考虑在这个机构上，能够与手臂机构相固定。

机构如图 2.3 所示：

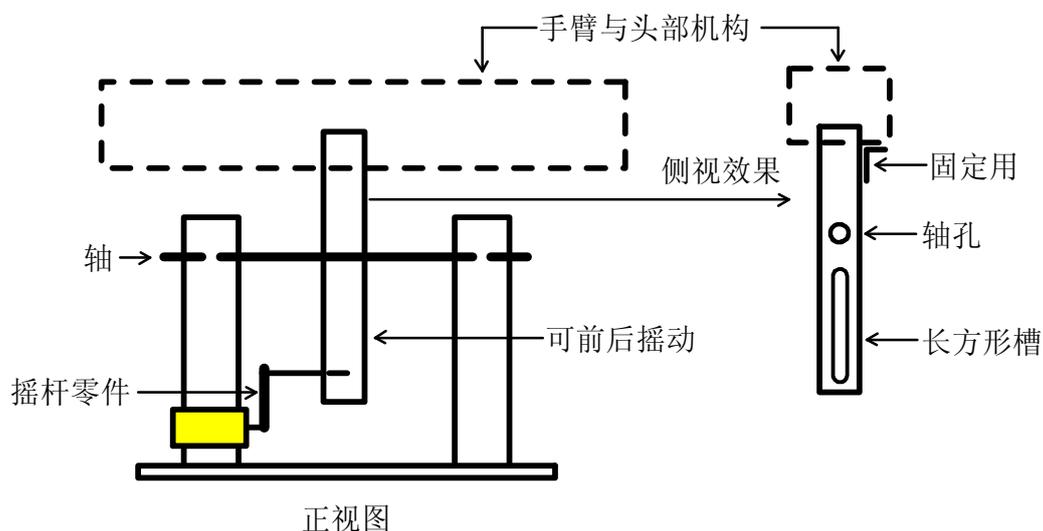


图 3.8 机器人鞠躬机构

鞠躬机构的装配顺序如下：

- (1) 取适当长度的铝条打轴孔和磨长方形槽，在上身支架两端打相同的孔，轴穿过该铝条后穿过两个支架并固定好
- (2) 加工可固定于电机轴上的摇杆，一端可固定住电机，另一端固定摇杆的轴并与加工好的铝条的长方形槽相结合
- (3) 将电机固定在上身左边支架的一侧
- (4) 为了防止弯腰机构前倾，用两根橡胶带做一定的固定，以使电机拉住机构向下弯腰时更加稳定

在弯腰机构中，如果将机构的位置和方向调换过来，动作就可变为肩的左右摇摆，因为设计水平有限，没有将两种动作结合在一起的机构设计出来。

3.2.1.3 转身机构的设计

(1) 连杆原理实现的转身机构

转身机构的设计原理和弯腰机构的设计原理差不多，在此就不做多的比较论证了。我们还是采用连杆机构，但具体的设计又和前面稍微不同。

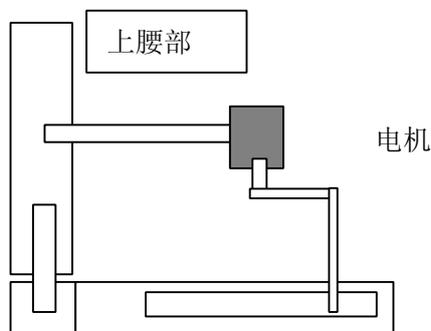


图 3.9 转身机构

(2) 电机做转轴的转身机构

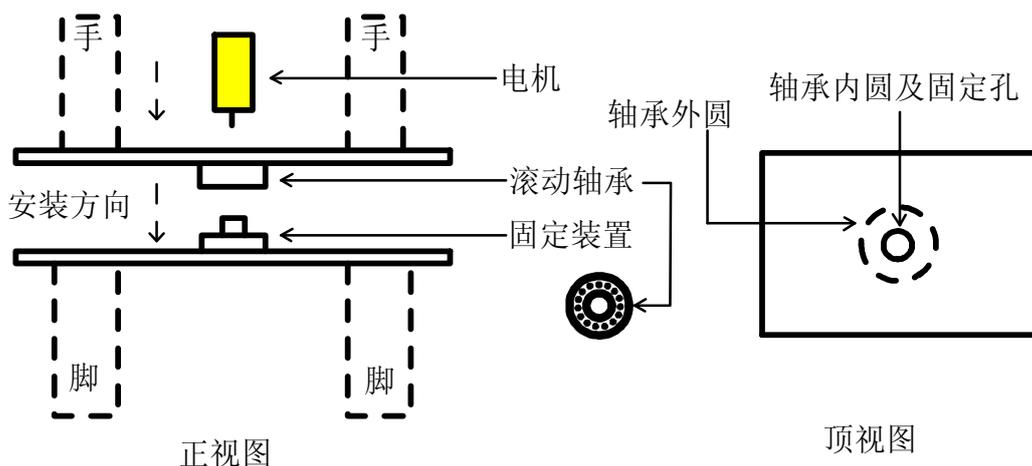


图 3.10 机器人的腰部机构

机器人的腰部机构用于实现机器人左右转身的动作，主要部件包括一个滚动轴承、一个自行加工的可固定滚动轴承内部圆的零件，以及必备的其它固定零件。

滚动轴承的特点在于当内圆固定时，外圆可以转动，反之亦然。机器人的腰部机构正是从这个特点出发制作的，要实现机器人腰部左右转动，就让滚动轴承内部圆与可固定电机轴的传动零件固定，同时该零件与机器人下身机构固定，滚动轴承与机器人上身机构固定，再将电机与传动零件固定后又与上身机构固定，这样，在电机转动的过程中，利用轴承的作用，机器人下身机构不动时，由于上身机构与轴承外圆固定，电机旋转带动了轴承外圆的旋转，于是，机器人上身就绕着轴承中心作转身运动。

这种机构的缺点是电机轴受力较大，电机容易损坏，同时要求电机有足够的转矩。

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

### 3.2.1.4 手臂和头部机构的设计

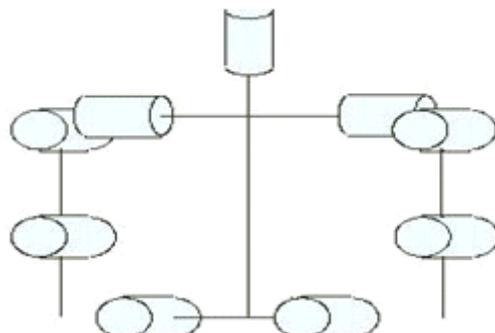


图 3.11 手臂机构

手臂机构的是机器人整体比较重要的环节，手臂自由度较多，机器人的很多动作都通过手臂体现出来。由于头部机构比较简单，在此和手部一起画出来，因为设计要求没有点头动作，我们只要一个电机来达到头部动作就可以了。

手臂机构就是机器人的两只手臂，在本次设计中将机器人的手部的两个电机与带动机器人头部转动的电机固定在了同一个机构上，同时，将该手臂与前后弯腰的机构相固定，构成机器人的上半身机构的一部分，然后该整体又与机器人的左右支架和腰部机构相结合，一起构成机器人的上半身机构。

参看做的很好的机器人的例子，机器人的手臂是机器人复杂程度的重要标志之一。手臂所具有的自由度越多，动作时活动的空间就越大，所做的动作就越精细，完成的任务就越复杂。

本次设计机器人的手臂由一根横杆构成，左右两端分别固定左右手的两个电机，可做前后 360 度的旋转，中间固定机器人的头部电机，可带动头部做 360 度的旋转，其中手臂的机构还可以继续扩展，在肩部上和臂末端可再扩展两个电机，实现三个自由度的动作。如图 2.4 所示：

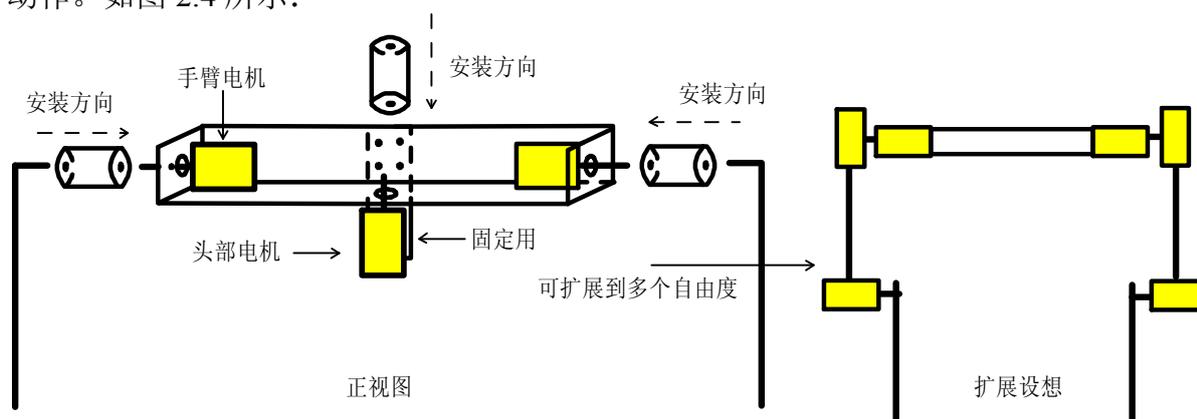


图 3.12 机器人的手臂机构

同时，为了防止手臂旋转过度使动作错乱，同左右转腰的机构一样，可在机器人手臂转动限定角度内装上限位开关，以达到动作的一致性。

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

装配顺序：

- (1) 选择合适钢性材料钻好孔后将左右手两个电机和头部电机固定住
- (2) 将上述做好的机构通过一块钢片钻孔后用螺钉固定在前后弯腰的机构上
- (3) 将两个传动杆固定到左右两个手臂电机上，在传动杆的另一端固定直径大小适合的两根钢轴作为机器人的手臂用
- (4) 将一个传动杆固定在机器人的头部电机的轴上，作为机器人可 360 度转动的脖子

### 3.2.1.5 材料与型材选择

机器人制作前，必须选择用于机器人本体的主要结构材料。在进行机器人制作时，特别要确定它自身的大小、轻重等，这就需要选择合适的材料来满足要求。对机器人本体材料选择，不仅仅有助于确定它的强度、重量、尺寸，而且确定了整体结构的难度水平、可能需要的特殊工具以及达到可接受结果所需要的技术水平。因此在选择材料时，一般是根据机器人以下原则进行挑选：

- (1) 零部件的形状、大小有何要求
- (2) 零部件需要多大的强度；
- (3) 要选择多大重量的零部件

下表列出机器人结构中使用的材料，比较它们的优缺点。

表 1 结构材料优缺点比较

材料	优点	缺点
木材	易于获得；价格较低，使用日常工具易于加工；硬木和层压板非常强硬而且坚固	不如塑料或者金属强度大；受潮容易变形；在压力下会有裂缝和碎裂
塑料	强度大而且耐用；有多种形式可选；包括板材和挤压型材；容易获得	遇高温会发生溶化或下陷；某些塑料收到冲击可能会碎裂，有些材料难以获得；性能较好的塑材比较昂贵
金属	强度非常高；铝材有各种方便使用的形状；甚至在高载荷以及高温下也可以保持全方位的稳定性	是所有材料中最重的；制作过程需要电动工具以及锋利的锯条和钻头才能完成；更难于加工；价格比较高

根据以上分析：选择机器人的本体材料，应该从机器人的性能要求出发，满足机器人的设计和制作要求，所以应在充分掌握机器人特性和各组成部分的基础上，从设计思想出发。机器人本体是被用来支承、连接、固定机器人的各部分，也包括机器人的运动部分。伴舞机器人臂部是运动的，其整体也是运动的，所以，机器人运动部分的材料质量要轻。伴舞机器人的结构设计成左右对称，高度适当，上下半身比例协调，确保结构紧凑。关节连接件的设计合适，既要保证机器人行动起来灵活自如，动作顺滑平稳，又要具有一定的承受负载能力。所以选用了密度较小的铝制作，因为铝易于加工，质量又轻，不会增加机器人的承载负担。

伴舞机器人是仿人机器人，材料截面对构件质量和刚度施加重要的影响，因此通过合理选择构件的截面可以较好地满足机器人的使用要求，如空心圆截面，空心矩形截面，工字截面等。

若空心矩形截面是边长为  $a$ ，壁厚为  $t$  的正方形，空心圆截面的外圆直径也为  $a$ ，壁厚也为  $t$ ，且令  $t=0.2a$ ，通过计算可以得出，在相同壁厚的条件下，正方形空心截面比空心圆截面的惯性矩高 69%~84%，而质量仅增加 27%。壁厚越薄，则效果越明显。若比较条件改为空心矩形截面和空心圆截面型材的截面相等，且  $D_0=a$ ，设空心圆截面壁厚  $t_D=0.2a$ ，可以计算出正方形空心截面的壁厚为  $t_F=0.147a$ 。此时，正方形空心截面比空心圆截面的刚度提高了 40%~60%。因此无论从何角度来衡量，空心矩形截面都比空心圆截面的刚性优越。从设计，装配的角度来看，矩形截面也有一定方便之处。

综上所述，机器人结构设计为空心矩形截面，由铝材加工而成。其基本框架是使用连接件和角撑支架来搭建的。而常见的机器人联接结构有扣接零件、胶接、胶带、铆接、焊接等，下面通过表格逐一比较各种结构技术的应用，从而确定最后方案。

表 2 机器人常用联接结构

结构	应用场合
扣接零件	小型玩具机器人
胶接	格局所需要的胶粘剂，可用于重量达数磅的机器人上
胶带	暂时性结构、轻型材料和元件的机器人上
铆接	小型到中等尺寸的机器人，以及轻型到中等重量的机器人
焊接	大型和重型机器人，特别用于恶劣环境的机器人

伴舞机器人负载小，自重轻，对寿命要求不高，因此除特别讲究强度、刚度以及抗摩擦磨损性的机构，机械连接上，应使机器人具有较好的坚固性，在机器人做舞蹈动作或行进过程中，各种接插件不能松动、脱落，满足合适的控制方案，所以选择铆接方式。

### 3.2.2 电气部分

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

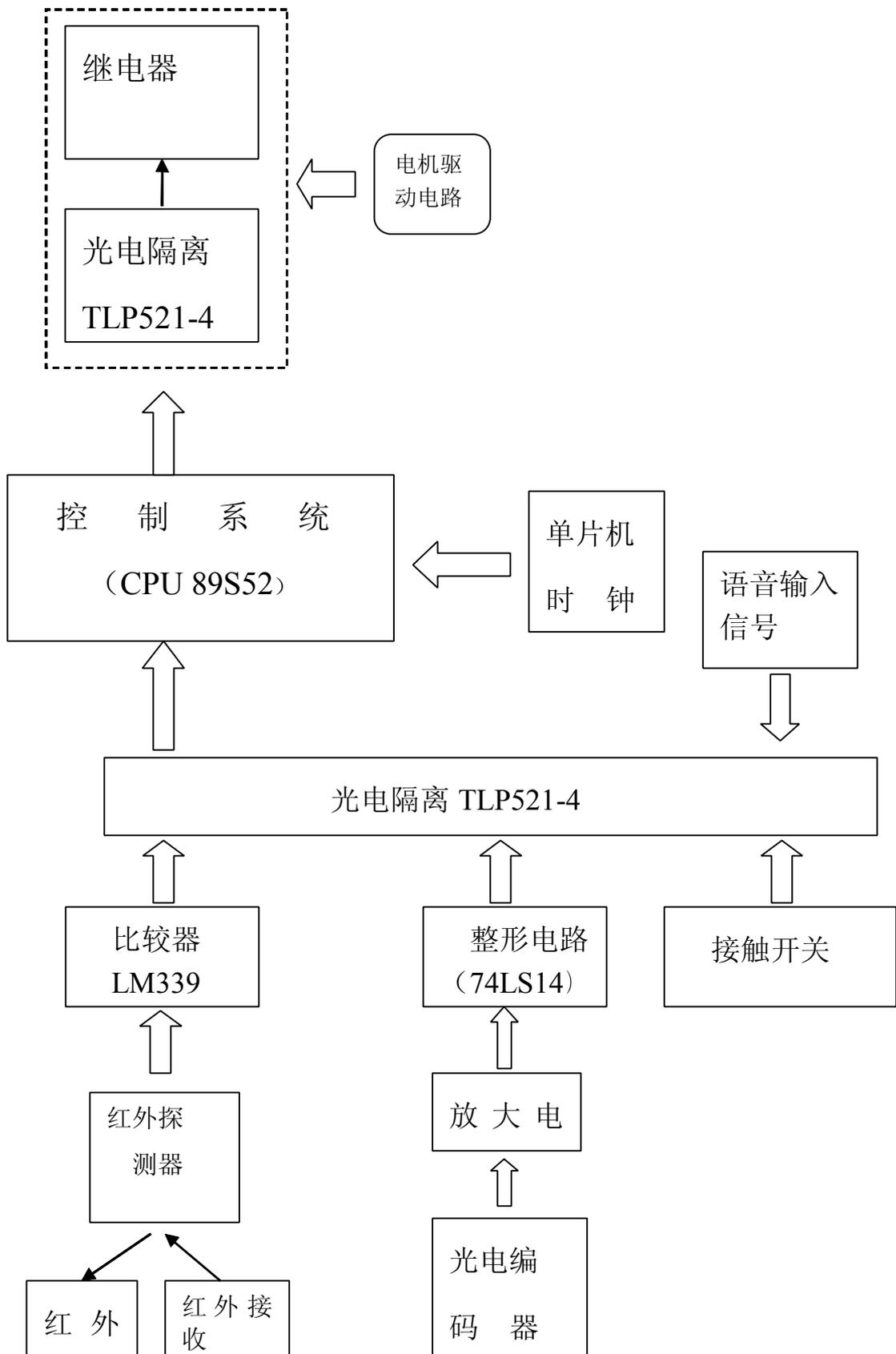
(1) 方案 1

电机驱动电路采用继电器构成桥式换向电路，继电器和主控制电路间用光电隔离器。考虑到电机的起动电流和制动时比较大，会造成电源电压不稳定容易对单片机和传感器的工作产生干扰，所以，电机驱动电路和单片机以及传感器电路用 TLP521-4 光耦隔离。提高电路的可靠性。同时还可以采用相互独立的电源供电。以确保单片机正常运行，进而以保证整个系统正常工作。

在机器人的控制系统中最主要的就是传感器电路，传感器是机器人感知外界事物的关键。机器人通过传感器收集外界信息，通过电路把这种信息转变成 CPU 能识别的数字信号，再通过软件算法对所收集的信息进行处理后，形成相应的控制信号，通过端口去控制相关的电机动作。

传感器采用多种传感器相互配合使用。使用光电编码器判断角位移和行程位移。主要用在机械手臂的转动上，使机械手臂动作更准确。行程开关用于位置的检测，主要用在机器人，或机械手臂上。当机器人行走，或机械手臂动作到达某种位置时给单片机一个电信号。让单片机发出相宜的控制信号。

此方案还待解决的就是传感器的抗干扰能力还不够高，其中主要是红外光电器件容易受外界光的干扰。光电编码器提高可靠性主要取决于光电编码器的加工精度，光源和接收管。一般对于比赛机器人的精度要求不是很高，编码器可以采用自制的就可以达到效果，而光电检测（对射式光偶）器已经可以从市场上很容易的购买到，所以要提高光电编码器的抗干扰能力并不难办到。机械开关的可靠性就更高了，他简单实用。但是对机械开关安装的位置一定要准确，牢固。如果安装不到位就无法触碰就检测不到电信号，又因为机器人经常运动如果安装不牢固开关就容易松动。对于提高红外传感器的抗干扰能力可以从以下几个角度去加强：



[点击这里](#) 获赠价值数万元的 200 多套原创技术方

图 3.13 方案一的电路方框图

(2) 方案 2

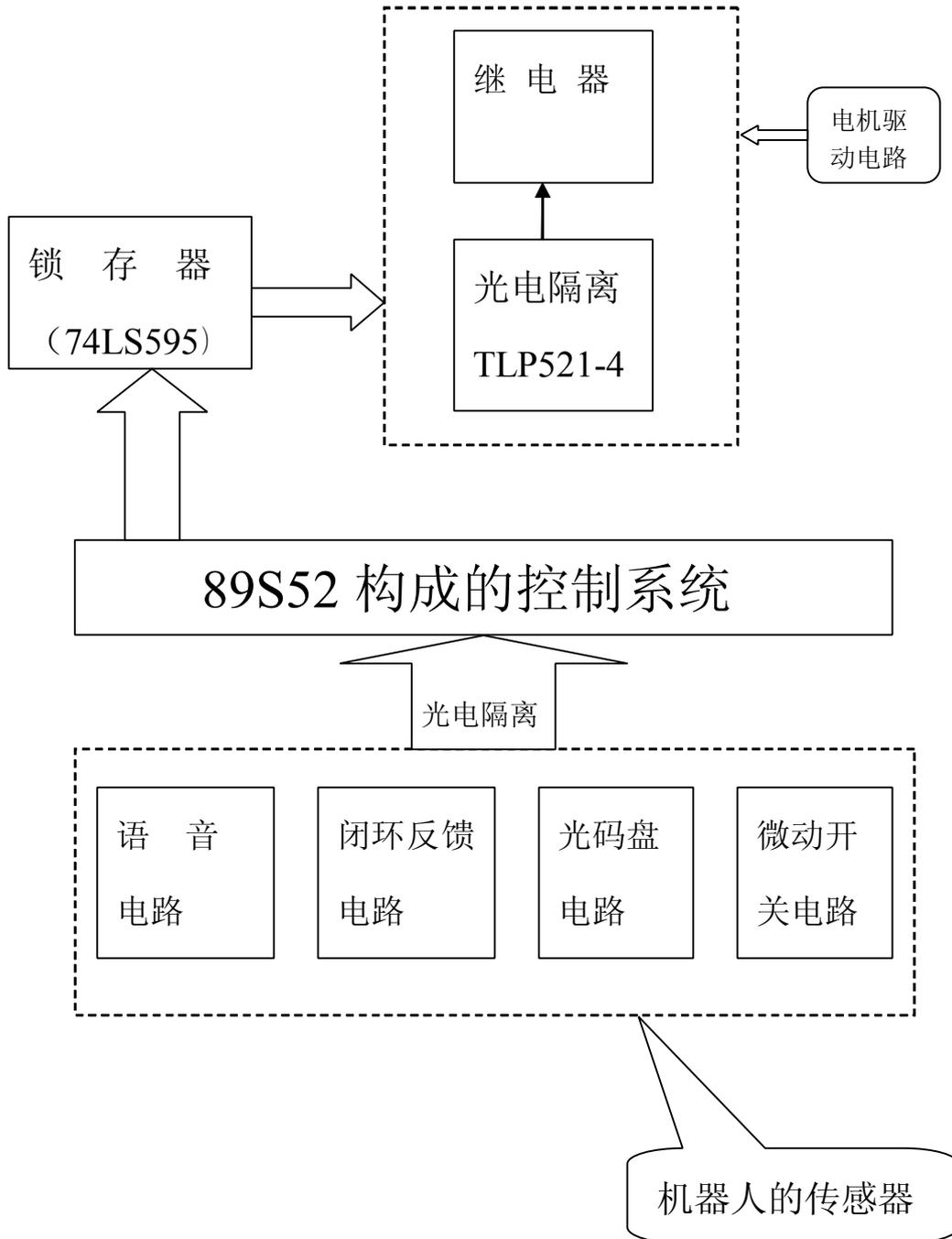


图 3.14 方案二的电路方框图

方案二的电气图基本不变,只是把 H 桥换成了继电器,把红外信号输入换成光码盘。继电器就有其自身的优胜,他相当于短接没有压降,能承受的电流也可以很大,可以把电机电枢短路,即相当于刹车。

方案二的电气方框图中,机器人的传感器系统由光电编码器,微动开关,闭环电路,语音电路构成。可见光探头用于检测白色引导线,光电编码器,微动开关和方案一的功能类似。

### 3.2.3 电方案的比较与选择

#### 3.2.3.1 电气方案比较

(1) 首先我们采用直流电机。电机的选取以方案(1)的普通直流电机为最佳选择,其控制比较简单,功率相对机器人比赛是够的;而步进电机功率不够,伺服电机太贵了;在电机驱动电路的选择上我认为用继电器构成开关控制,因为对电机的控制要求不一样所以对不同的电机采用不同的控制方法。三极管构成的驱动电路其优点是能够进行脉宽调速,缺点是三极管有压降。用于行走的电机采用此种方法驱动的力度不够大,对完成动作的电机可以采用由两个继电器实现其正反转和启动停止。

(2) 传感器选用以闭环传感器为主,其他传感器为辅,其中包括光码盘,触碰开关。

(3) 单片机控制系统采用单机形式。其可靠性能更高。

综上所述,电气方案选择最佳电气方案(2)

#### 3.2.3.2 机械方案比较

通过以上各个机械方案的提出并描述了各自的优缺点,比较得出所要设计的机器人的结论为:

(1) 移动方式以方案(1)的四轮移动为最佳选择,其速度要比履带式移动快,且其能量耗损相对较少;

(2) 机械手的设计模型已和人的手臂很相似,不会有很大的改动。

(3) 转身和弯腰机构则是连杆机构较为合理,其它机构不是成本高,就是不易于实现,在此我们选用以设计好的连杆机构。

(4) 机器人的材料我们以铝材为主,

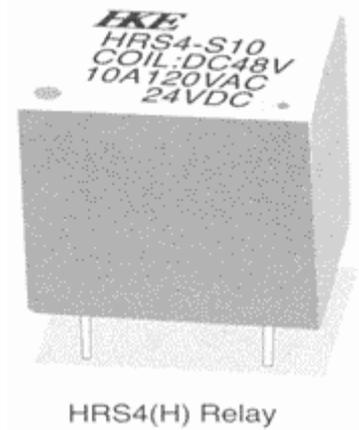
## 4 硬件电路设计

### 4.1 电机驱动电路

继电器是本系统的执行机构,本系统选用D4810型固态继电器,固态继电器(SSR)与机电继电器相比,是一种没有机械运动,不含运动零件的继电器,但它具有与机电继电器本质上相同的功能。SSR是一种全部由固态电子元件组成的无触点开关元件,他利用

电子元器件的点，磁和光特性来完成输入与输出的可靠隔离，利用大功率三极管，功率场效应管，单项可控硅和双向可控硅等器件的开关特性，来达到无触点，无火花地接通和断开被控电路。 固态继电器有三部分组成:输入电路，隔离(耦合)和输出电路。按输入电压的不同类别，输入电路可分为直流输入电路，交流输入电路和交直流输入电路三种。有些输入控制电路还具有与TTL/CMOS兼容，正负逻辑控制和反相等功能。固态继电器的输入与输出电路的隔离和耦合方式有光电耦合和变压器耦合两种。固态继电器的输出电路也可分为直流输出电路，交流输出电路和交直流输出电路等形式。交流输出时，通常使用两个可控硅或一个双向可控硅，直流输出时可使用双极性器件或功率场效应管。固态继电器有寿命长、可靠性高、灵敏度高、控制功率小、电磁兼容性好、快速转换、电磁干扰小的优点。D4810型继电器输入电流为5-40mA，电压3-30V，额定输出电流10A，输出电压范围宽为20-220V，满足项目的要求。

在直流电机电枢上加上一定幅度的直流电压，电机便开始旋转；改变电压方向，电机转向也随之改变。如果电机以恒速运动，不调速可以采用一个简单的电路来控制正/反向，电路原理如图 3.1 所示。它采用两个继电器作为控制弱电到驱动电机的强电之间的转换器。正反转控制开关 RELAY1 断开，电源开关 RELAY3 闭合时，电机正转；反之电机反转。两开关同时断开，电机停转。两开关同时闭合是不允许的，从电路可看出，这会导致电源短路。



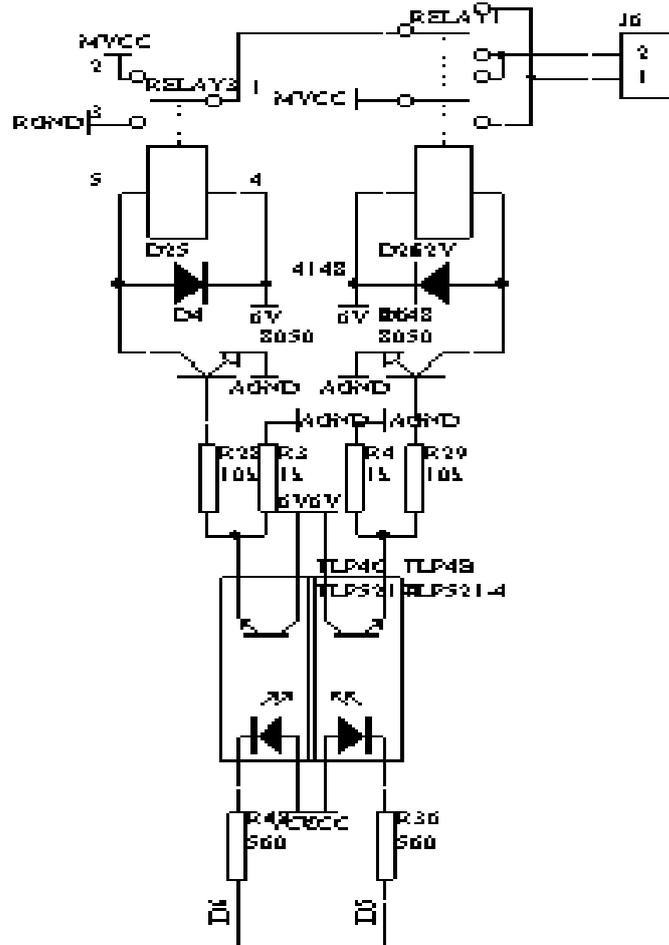


图 4.1 直流电机的简单双向控制及继电器图

电路原理图如下：

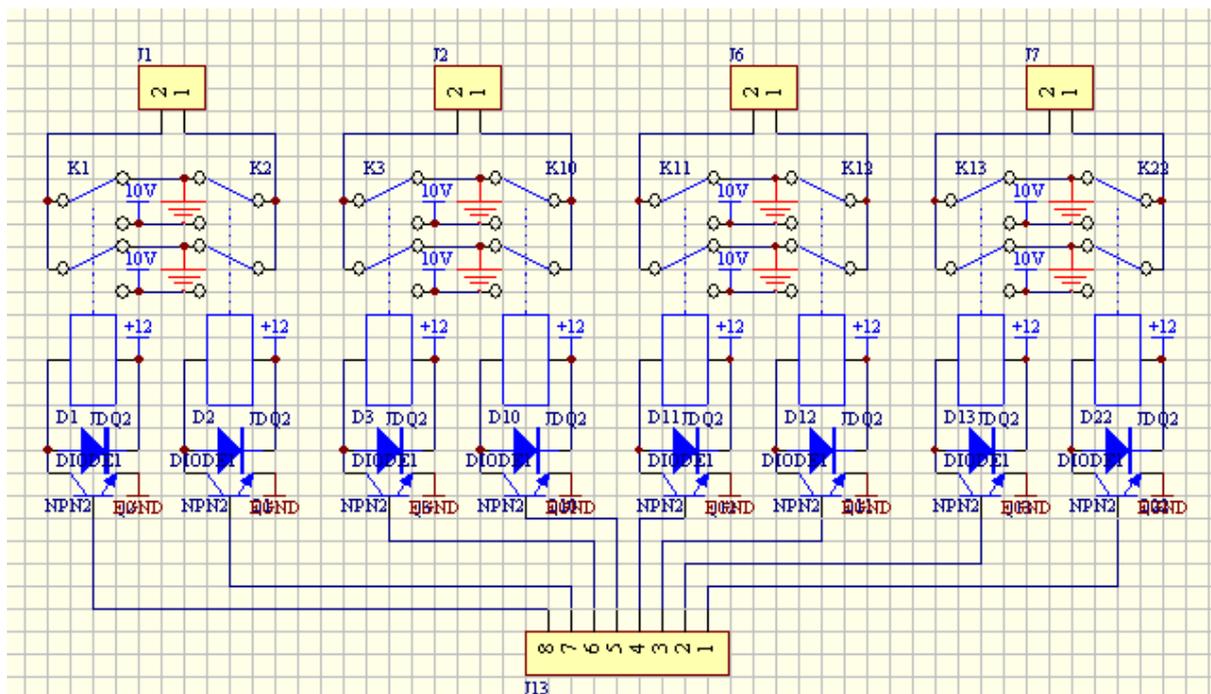


图 4.2 继电器电路图

## 4.2 光电编码器

编码器是一种简单的数字式位移传感器，它能够把角位移或线位移经过简单的变换变成效字量。目前，编码器精度、分辨力和可靠性都优于相同尺寸的模拟式传感器，但价格一般比较高。根据编码器检测原理和检测对象的不同，可以把编码器分为两种类型。用于检测线位移的编码器称为直线位移编码器；用于检测角位移的编码器称为角度数字编码器。

角度数字编码器由敏感元件和码盘组成。码盘是一种经过精细加工的薄圆盘，它的材料根据与之配套的敏感元件不同而不同。角度数字编码器有增量式码器和绝对式编码器两种基本类型。绝对式编码器能给出与每个角位置相对应的数字量输出，增量式编码器则只输出角度的变化增量光学编码器。这种编码器由光源、码盘和光电敏感元件组成。图 3.13 表示了一种五位光学编码器。光学编码器的码盘是在一个基体上形成透明和不透明码区，代表了二进制码 0 和 1。编码器的性能好坏主要取决于码盘的质量。光学编码器通常采用照相技术和光刻技术制作。把同心分布的多因二进制码复制在码盘上。光源是光学编码器中最容易发生故障的元件。

通常不但要求光源具有较高的可靠性，而且要求光源的光谱要相光电敏感元件相适应。光源的工作温度范围要宽。光电敏感元件可以采用光电二极管、光电晶体管或硅光电池。为了提高输出逻辑电压，还需要加接各种电压放大器，而且每个轨道对应的光电敏感元件要接一个电压放大器。电压放大器通常由集成电路高增益差分运算放大器组

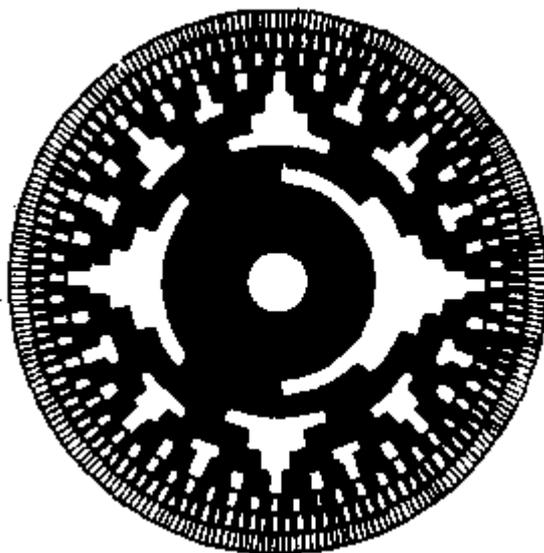


图 4.3 光电编码盘

成。为了减小声的影响，在光路中要加入透镜和狭缝装置。应当注意的是，狭缝不能太窄，要保证所有轨道的光电敏感元件的敏感区都处于狭缝内。光学编码器的基本构成如图 3.14 所示。

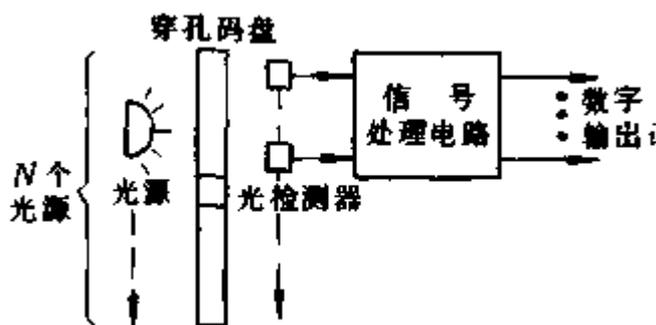


图 4.4 光学编码器的基本构成

绝对式光学编码器通常采用的是格雷(Gray)码。它是一种非标准的二进制码。Gray 码与标准二进制码的区别，在 Gray 码中，相邻两个二进制码的变化只能有一位。所以，在连续的两个数码中，若发现数码的变化超过一位，就认为是非法的读数，从而具有一定的纠错能力，可提高可靠性。光电编码器可以检测机器人各个关节的运动状态。若电机电流大，但编码器无输出，说明机器人撞到对方而无退让，己方机器人就须要调整战术。若电机欲前进，编码器反转，说明机器人受对方推堵而退让。最近内藏整形电路直接以矩形波输出的编码器已问世，可以直接接入计算机。在我们的机器人中均采用自制的编码器。在带有明暗方格的码盘两侧分别安置发光元件和光敏元件，随着码盘旋转，光敏元件接收的光通量与方格的间隔同步变化。光敏元件随着输出经过整形的脉冲。根据脉冲计数，可以知道与码盘固接的旋转轴的角度位移。

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

下图是设计中用到的编码器电路图（图 3.15）：

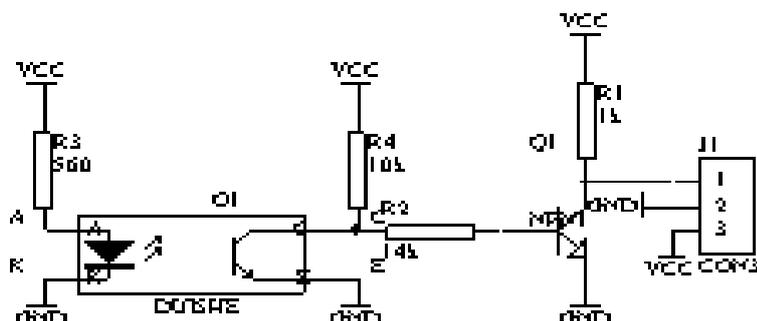


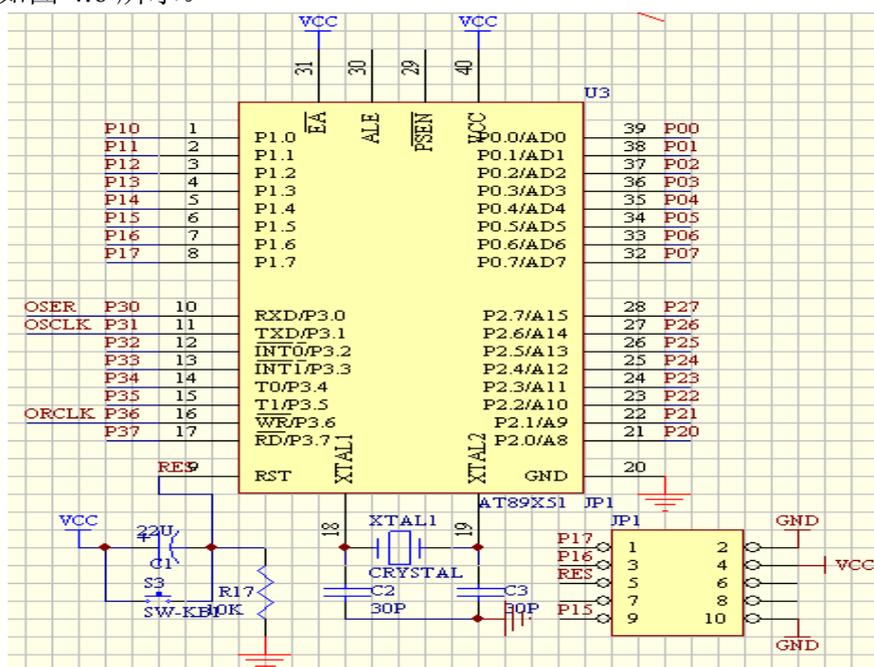
图 4.5 编码器电路图

### 4.3 开关

比赛机器人常用的开关用按钮开关，数字开关和接触开关几种。按钮开关在机器调试阶段可以方便地提供人工输入。数字开关一般用于向 CPU 的输入选择，如速度等级，行走模式，策略等。接触开关则用于检测是否触碰，检测的对象有场地的道具，环境界限等，一般微动开关即可。

### 4.4 单片机控制系统

采用单片机完成控制的系统，在系统中 P1 口作为光电传感器的输入端口，P0.0——P0.3 作为行走电机的控制端口。P2 作为动作电机的控制端口。P3.4——P3.5 作为光电编码器的记数口。P3 口的另外一些都用来接行程开关。另外留出串行通信的端口以备以后使用，如图 4.6 所示。



[点击这里](#) 获赠价值数万元的 200 多套原创技术方

图 4.6 控制系统的的电路图

当 P0 口作为一般的 I/O 口用时我们要接上一个排阻作为上拉电阻。这是由单片机的内部特性所决定的。

单片机及其外围接口电路包括单片机、时钟电路、复位电路、外部接口扩展电路等。单片机采用 AT89S51 系列单片机，它是十六位的高性能嵌入式控制器，具有 40 个可编程的 I/O 口，价格便宜，但功能强大且支持在线编程功能，使用十分方便。时钟电路给单片机系统提供时间基准设计时采用 12MHz 晶振。复位电路用于上电后使系统回复到初始状态开始运行。整个单片机控制系统框图如下：

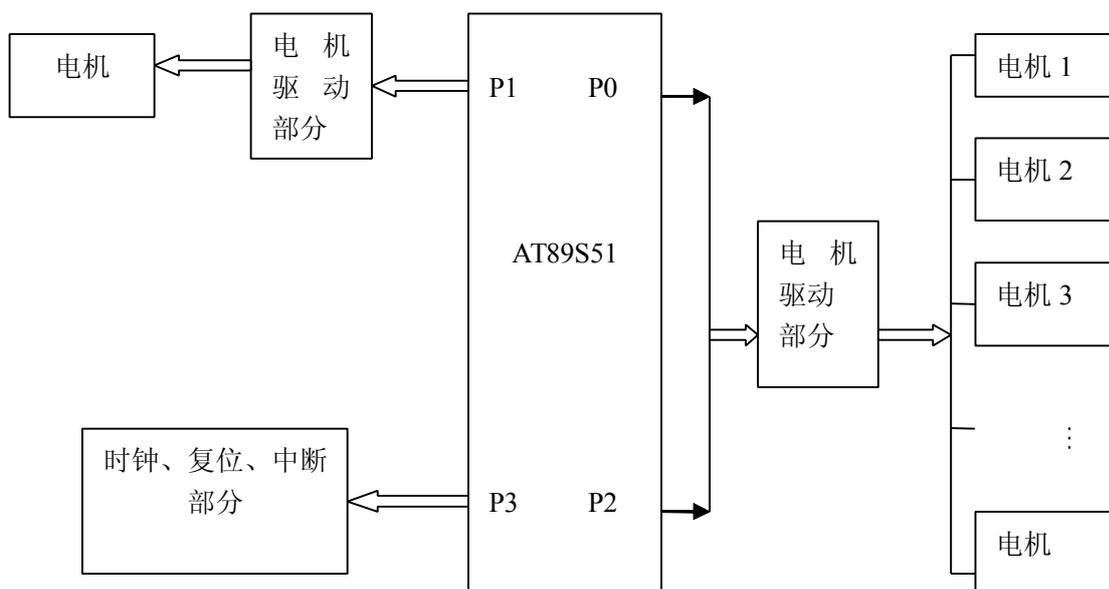


图 4.7 单片机控制系统框图

#### 4.5 信号隔离模块

在实际的电子电路系统中，不可避免地存在各种各样的干扰信号，若电路的抗干扰能力差将导致测量、控制准确性的降低。因此在硬件上可以采用一些设计技术，破坏干扰信号进入测控系统的途径，从而有效地提高系统的抗干扰能力。

常见的信号隔离方法有光电耦合和变压器隔离等，本系统设计采用光电耦合方式对电机控制信号进行隔离，以消除模拟信号对数字信号的干扰。使用的光耦型号为 TLP521 - 2 及 TLP521 - 4。信号隔离最重要的一点就是将数字与模拟或强电与弱电部分在供电上要完全隔开，否则就不能达到真正意义上的隔离。为了更好的防止干扰，电路中使用了光电耦合器，一般由三部分组成：光的发射、光的接收及信号放大。输入的电信号驱动发光二极管（LED），使之发出一定波长的光，被光探测器光敏三极管接收而产生光

电流，再经过进一步放大后输出，这就完成了电—光—电的转换。由于光耦合器的输入端和输出端之间通过光信号来传输，因而两部分之间在电气上完全隔离，没有电信号的反馈和干扰，故性能稳定，抗干扰能力强。发光管和光敏管之间的耦合电容小（2pf 左右）耐压高(2.5KV 左右)，故共模抑制比很高。输入和输出间的电隔离度取决于两部分供电电源间的绝缘电阻。此外，因其输入电阻小（约 10Ω），对高内阻源的噪声相当于被短接。因此，由光耦合器构成的模拟信号隔离电路具有优良的电气性能。实际中，光耦两侧使用了两组不同的电源。光耦电路如下图所示：

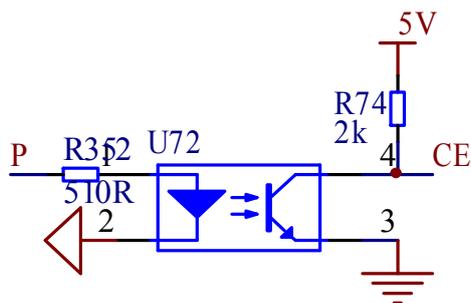


图 4.8 光耦隔离电路

光耦隔离电路图如下：

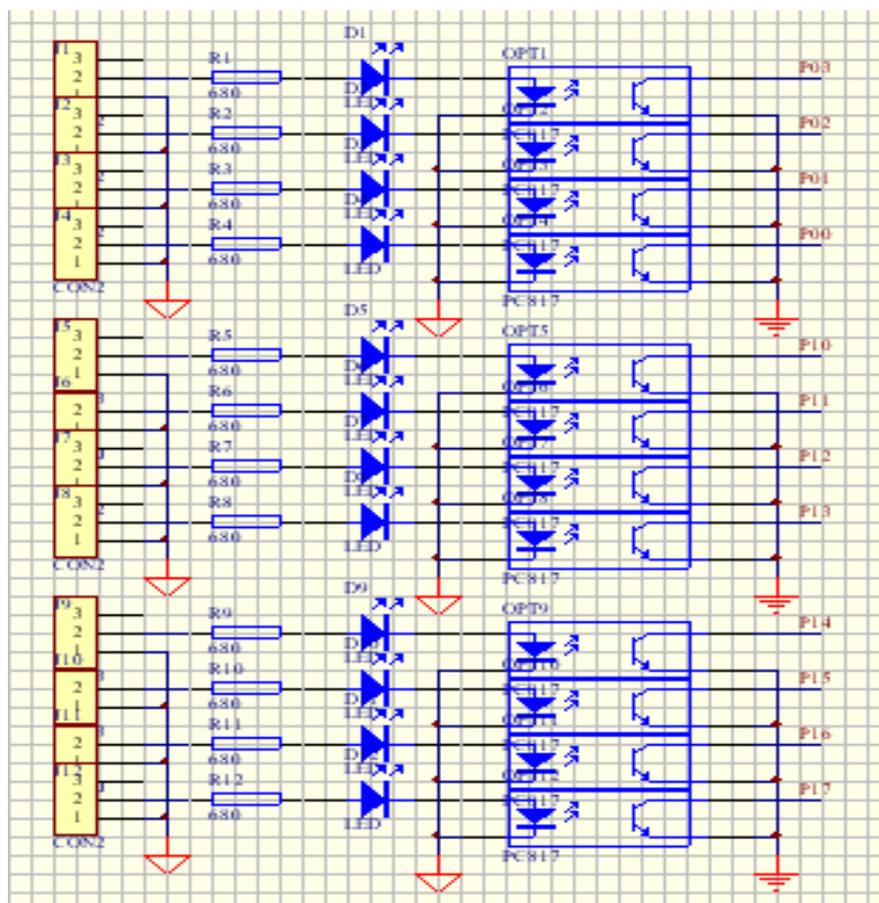


图 4.9 光耦隔离电路图

#### 4.6 锁存器 74HC595 模块

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

74HC595 的工作原理，锁存器一般是由 D 触发器集成的，可以记住一个信号，直到下一个触发信号到来为止。

74HC595 具有一个 8 位串行输入并行输出的移位寄存器和一个 8 位输出锁存器。由于占用 MCU 的 I/O 口线少，硬件结构简单，因此应用广泛。

#### (1) 应用原理

以点亮 LED 数码管显示“0”为例！

1. 并口输出： 共用 P1 八个口，输出用一条指令 `MOV P1, #3FH` 即可。
2. 串并输出： 共用 P1.0-P1.2 三个口。但输出必须先将 3FH 逐位移入 595 移位寄存器内(即串入)，再锁存输出(即并出)。

注意：只有数据锁存后，3FH 才出现在 595 的 Q1-Q7 上，LED 数码管显示“0”。

移位方式：1-1-1-1-1-1-0-0，由低到高。

#### (2) 主要引脚

- 14 脚： 串行数据输入端接 MCU-P1.0。
- 11 脚： 移位时钟接 MCU-P1.2。
- 12 脚： 输出锁存器的送入信号 MCU-P1.1。

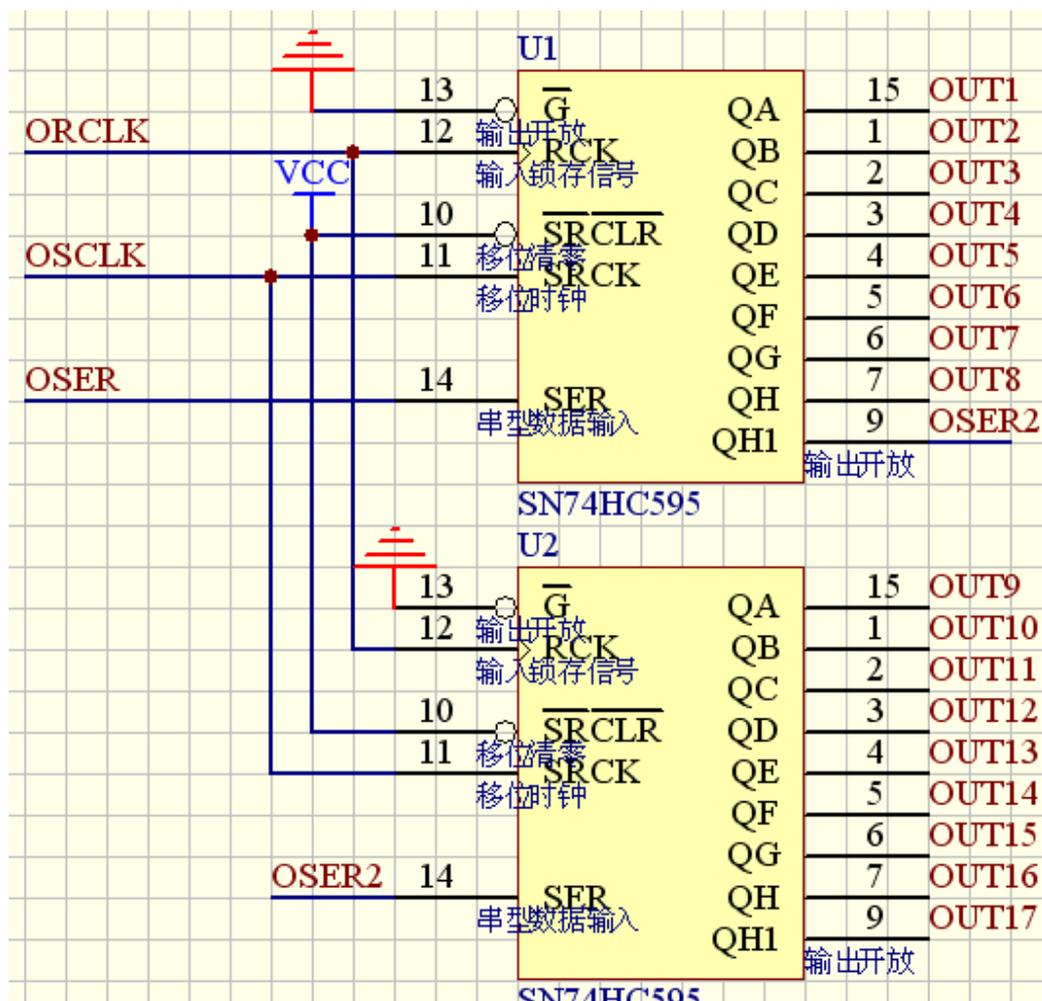


图 4.10 74HC595 的原理图

### (3) 应用三部曲

第一步：目的：将要准备输入的位数据移入 74HC595 数据输入端上。

方法：送位数据到 P1.0。

第二步：目的：将位数据逐位移入 74HC595，即数据串入

方法：P1.2 产生一上升沿，将 P1.0 上的数据移入 74HC595 中。从低到高。

第三步：目的：并行输出数据。即数据并出

方法：P1.1 产生一上升沿，将由 P1.0 上已移入数据寄存器中的数据送入到输出锁存器。

说明：从上可分析：从 P1.2 产生一上升沿（移入数据）和 P1.1 产生一上升沿（输出数据）是二个独立过程，实际应用时互不干扰。即可输出数据的同时移入数据。

### (4) 级联的应用

74HC595 主要应用于点阵屏，以 16\*16 点阵为例：传送一行共二个字节（16



图 4.11 61 单片机内部结构图及引脚图

计合理，性能稳定，单独引出的IO口引脚方便开发，所以，本次设计直接采用了凌阳61板作为语音功能和控制功能部分的模块电路使用，这样，大大节省了硬件电路的开发时间，能将精力主要集中于实现语音功能的软件开发上，也能有更充分的时间深入了解硬件在现代相关音频处理技术方面的一些基本原理和问题。

下图是61板的实物图：

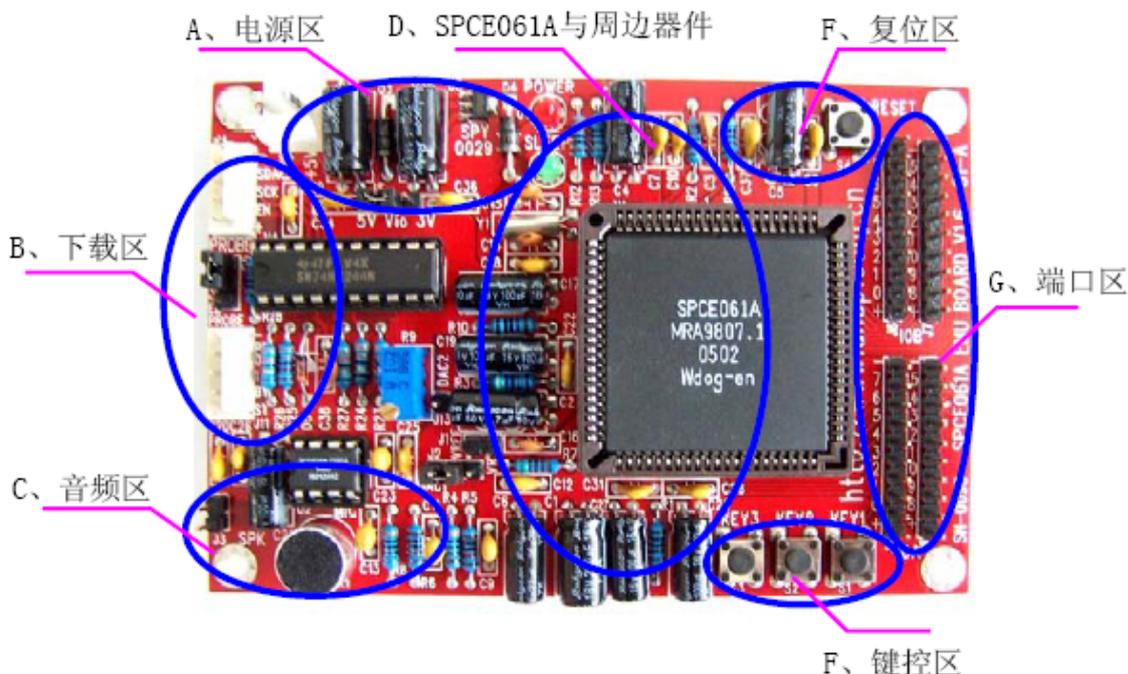


图4.12 61板实物图

61板各模块电路原理图如下：

放音电路：

在放音电路中，61板采用了SPY0030功放芯片，和LM386工作电压需在4V 以上相比，SPY0030 仅需2.4V即可工作；LM386 输出功率100mW 以下，SPY0030 约700mW。这样，在给61板5V的工作电压下，音量可达到较好效果，并且在运放前端加上了1K的精密可调电阻可调节音量的大小。

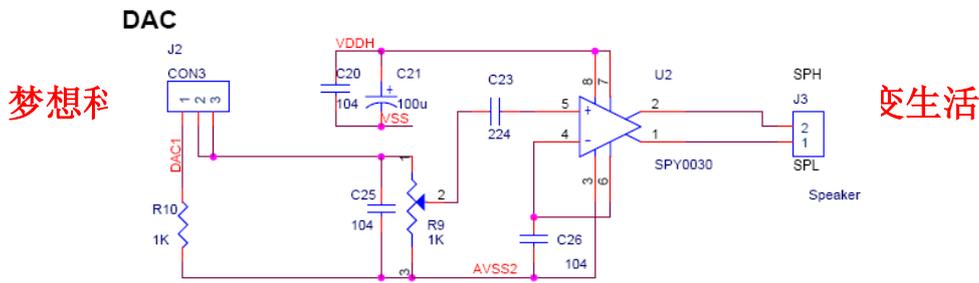


图4.13 61板DAC电路

电源电路:

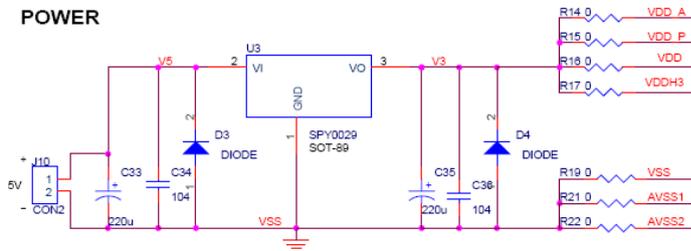


图 4.14 61 板电源电路

61 板电源由 SPY0029 稳压提供。

程序下载电路:

该电路以 74HC244 实现，同一般单片机下载电路，在此不再详述。

麦克风输入电路:

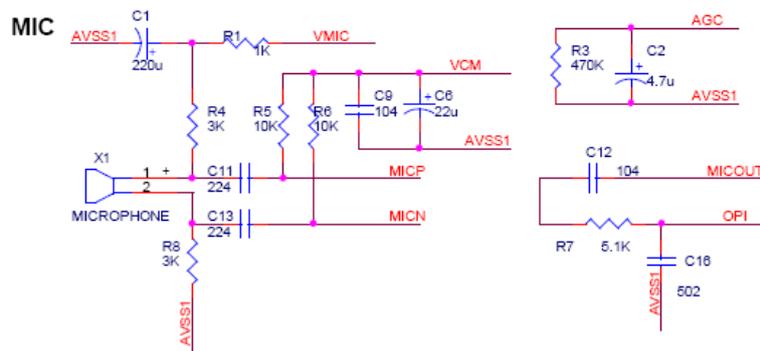


图 4.15 61 板麦克风输入电路

麦克风输入电路对于语音识别的效果有最直接最重要的影响，在 61 板的麦克风输入电路中，加入了许多滤波措施，并加入了 AGC 增益，在对 61 板语音输入和语音识别的测试中，即使是最普通最低廉的麦克风，录入语音都比较清晰，语音识别也非常敏感和正确，这也说明了 61 板电路设计的合理性。

61 单片机最小系统:

SPCE061A 的最小系统包括系统锁相环 PLL、晶振电路和复位电路，可按照 61 单片机的引脚说明进行连接，这里不再详述，可参看附录的 61 板原理图。

## 5. 电机和能源的选择

驱动系统是机器人的手和脚，即运动机构和执行机构的基本组成部分。驱动系统的

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

设计与制作是比赛机器人创意与实践的重要内容。它的任务不仅是向机器人传递动力，而且要提供准确的运动定位和灵活的操作，是机器人制胜的法宝之一，因此其重要性是不言而喻的。通常机器人驱动有电气，液压，气压三种方式。其中以电气驱动最常见，它按供电的方式又可以分为直流电机和交流电机。机器人比赛规则通常规定采用电池供电，即直流电。采用直流电供电的电机按照其内部结构又可分为直流电机和步进电机。直流电机外部接通直流电就能旋转；步进电机的驱动原理复杂些，必须按照一定的通电顺序给电机供电，电机才能旋转。

## 5.1 直流电机的特性

### 5.1.1 直流电机的结构特点

直流电机的结构形式很多，但基本的结构是相同的，即必须有定子，转子和换向器。定子是固定在机身的圆桶状部分，一般由永磁材料或能产生磁场的线圈制成。转子由一根旋转轴及固定在轴上的硅钢片构成，是把电能转换成机械能的部分。转子的外沿有槽，槽内有电流通过是，旋转轴便开始旋转。为了在电机连续旋转条件下电流能顺利通过转子，旋转轴尾部设置有换向器，它始终与定子的电刷保持接触，给电机绕组供电。

直流电机有普通直流电机和直流伺服电机之分，两者的区别主要体现在性能方面。伺服电机在调速范围，机械特性和调节特性的线性度，响应快速性等方面均占优，且当控制电压改变为零时能立即停止转动；普通电机则做不到。从结构上比较，两者的区别在于电枢铁心的长度与直径之比，伺服电机较大，而气隙较小。因此在精密定位和宽调速范围的应用场合，一般都选用伺服电机。

### 5.1.2 直流电机的运行特性

对一般机器人用户来说，驱动的任务仅仅是在确定了机械结构并计算出所需要的速度和驱动力矩之后选择合适的电机和相应的控制方法，这就涉及到电机的运行特性问题。

机器人比赛规则往往对机器人的质量有严格的限制，所以应选用质量小，功率适宜的永磁电机。它的励磁磁通是恒定的。在控制系统中，通常用电枢电压作为控制信号，电压的大小与电机转速成正比，改变电压极性，电机运转方向也随之改变。图 4.1 是电枢控制的原理图。根据电枢反应原理，电机转速  $n$  的计算公式为

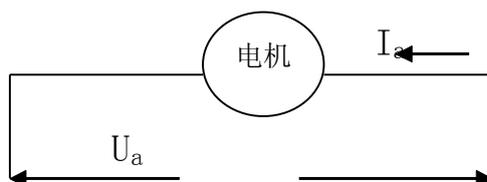


图 5.1 电枢控制的原理图

$$n = n_0 - K_h T_e m \quad (5.1)$$

式中  $n_0 = U_a / K_t$ ,  $K_h = R_a / K_t K_e$ , 而  $U_a$  为电枢电压,  $K_t$  为转矩常数,  $R_a$  为电枢电阻,  $K_e$  为电势常数, 电机的电磁转矩  $T_{em}$  应为

$$T_{em} = C_t \phi I_a = K_t I_a \quad (5.2)$$

式中  $I_a$  为电枢电流,  $C_t$  为常数,  $\phi$  为磁通量。

式 (5.1) 是电枢控制的静态特性方程, 即稳定运行状态方程, 由此可得直流电机的机械特性和调节特性。

### (1) 机械特性

机械特性指控制电压恒定时, 电机转速随转矩变化的关系。由转速公式可得到直流电机的机械特性, 如图 5.2 所示。机械特性是线性的, 当电压一定时, 转速升高则转矩下降。特性曲线与纵轴的交点即为电磁转矩等于零时的理想空载转速  $n_0$ , 即

$$n_0 = U_a / K_e$$

可以看出, 空载时电机转速与电枢电压成正比。但实际电机空载运行中, 即使电机轴上不带负载, 因为自身的空载损耗, 电磁转矩并不为零。机械特性曲线与横轴的交点为电机堵转时 ( $n=0$ ) 的转矩, 即所谓的电机堵转转矩  $T_d$ ,

$$T_d = K_t U_a / R_a$$

在图 5.2 中, 机械特性曲线斜率  $k$  的绝对值为

$$k = \tan \alpha = n_0 / T_d = R_a / K_t K_e$$

它表示机械特性的硬度, 即电机的转速随转矩  $T_{em}$  的改变而变化的程度。由图 5.2 可以看出, 随着控制电压  $U_a$  增大, 电机的机械特性曲线平行地向转速和转矩增加的方向移动, 但斜率不变, 所以电枢控制时电机的机械特性是一组平行线。

### (2) 调节特性

电机的调节特性指电磁转矩恒定时, 转速随控制电压变化的规律, 如图 5.3 所示。调节特性与横轴的交点, 表示在某一电磁转矩是电机的启动电压。转矩一定时, 若电机的控制电压大于相应的启动电压, 电机便能启动并达到所要求的转速; 反之, 最大电磁转矩小于要求的转矩值, 电机就无法启动。调节特性曲线上从原点至启动电压点的这一段横坐标所示的区域被称为伺服电机某一电磁转矩值所对应的死区。死区电压的大小与电磁转矩的大小成正比。

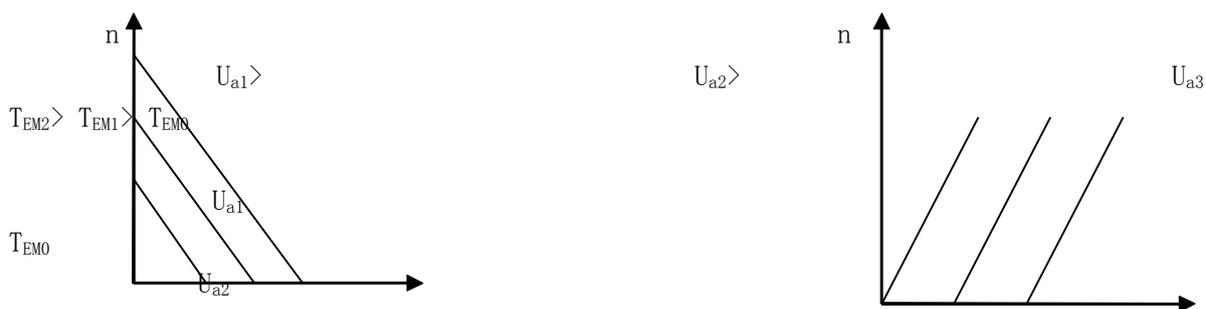




图 5.2 直流电机的机械特性

图 5.3 直流电机的调节特性

### 5.1.3. 直流电机的动态特性

在直流电机的电枢上加阶跃电压，转子速度随时间的变化规律称作直流电机的动态特性。动态特性用  $n=f(t)$  表示。动态特性本质上是由过渡过程来描述的。直流电机产生过渡过程的原因在于电机存在两种惯性，即机械惯性和电磁惯性。为了满足自动控制系统的快速响应的需要，直流电机的机电过渡过程应尽可能短，以便转速的变化能跟上控制信号的改变。为了更好的理解直流电机的过渡过程，可将电机电枢回路等效于图 5.4 来进行分析。他通常可以看成是一个惯性环节来建立传递函数。转子的角速度  $\omega$  的公式为

$$\omega(t) = U_a(1 - e^{-(t/\tau_m)}) / U_d = (1 - e^{-(t/\tau_m)}) \quad (5.6)$$

(a) 式中  $\omega_0 = U_a / U_d$  为电机理想空载角速度， $\tau_m$  为电机的机械时间常数， $\tau_e$  为电机的电气时间常数。通常因为电枢组的电感很小， $\tau_e$  比  $\tau_m$  小得多，往往可以忽略不计。

电机角速度随时间变化的关系如图 5.4 所示。从式(5.6)知，当时，电机的角速度上升到稳定角速度的 0.632 倍；当时间  $t = 4\tau_m$  时，电机角速度  $\omega = 0.985\omega_0$ ，一般可认为

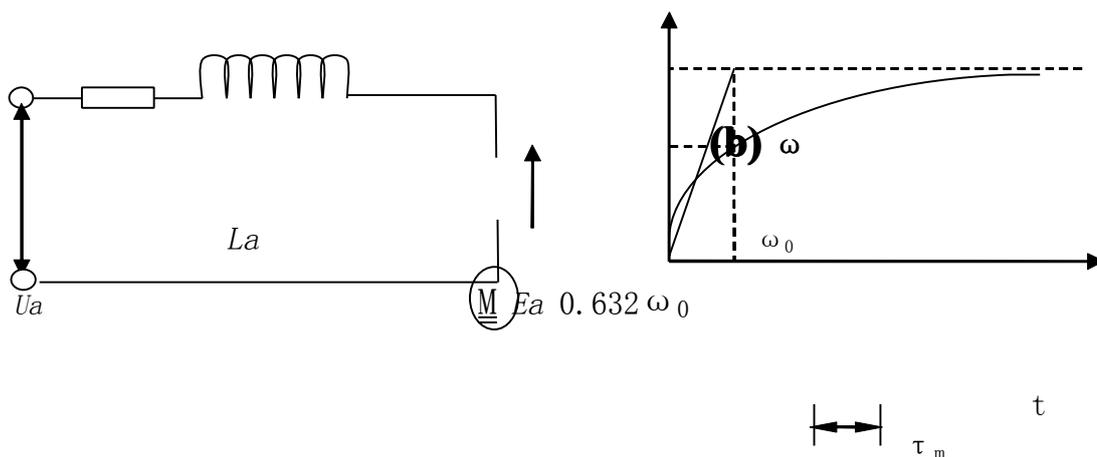


图 5.4 直流电机等效电路

电机的过渡过程已结束。因而可将电机的机械时间常数  $\tau_m$  定义为：电枢外加阶跃电压后，电机转速从 0 上升到稳定转速的 63.2%所需的时间。他是衡量过渡过程的一个

重要指标。

## 5.2 能源选取

不需要太大的输出扭矩，必须能灵活操控，而且要求体积小，重量轻、方便检修，毫无疑问是蓄电池提供的直流电源外加直流电机。对于需要大驱动力低转速场合和场合，可以用齿轮链、同步皮带等装置处理。但他有其缺点就是蓄电池的重量太大，在要求不大的情况下我们可以采用锂电池。他的优点就是容量大，体积小，过电流大，重量轻。另外机器人还需要一类共给数字 IC，单片机，传感器工作的电源，通常为+5v，约数十到数百 mA。他必须和电机的电源分开，形成单独供电。这样可以使 CPU 供电更加稳定。

电源的电路原理图如下：

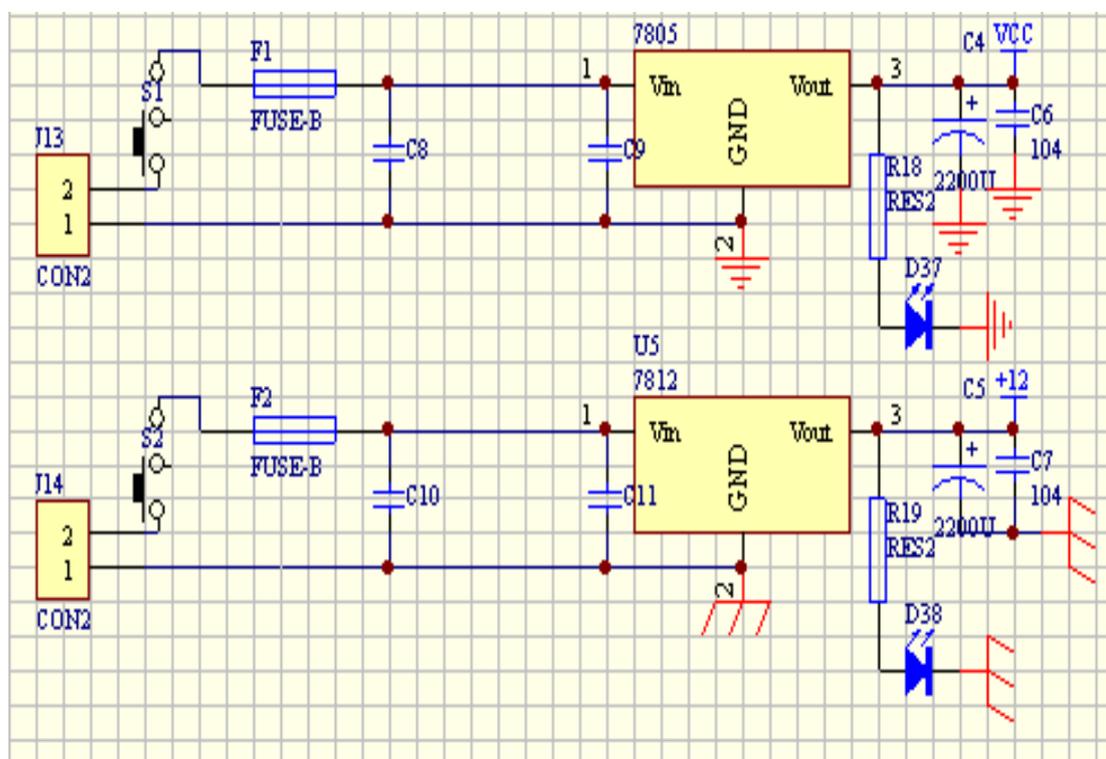


图 5.5 电源模块电路图

## 6 机器人的完整图

舞蹈机器人由以上机械本体和底盘设计、材料的选择和连接方式的分析最后制作的机器人整体图如下：

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

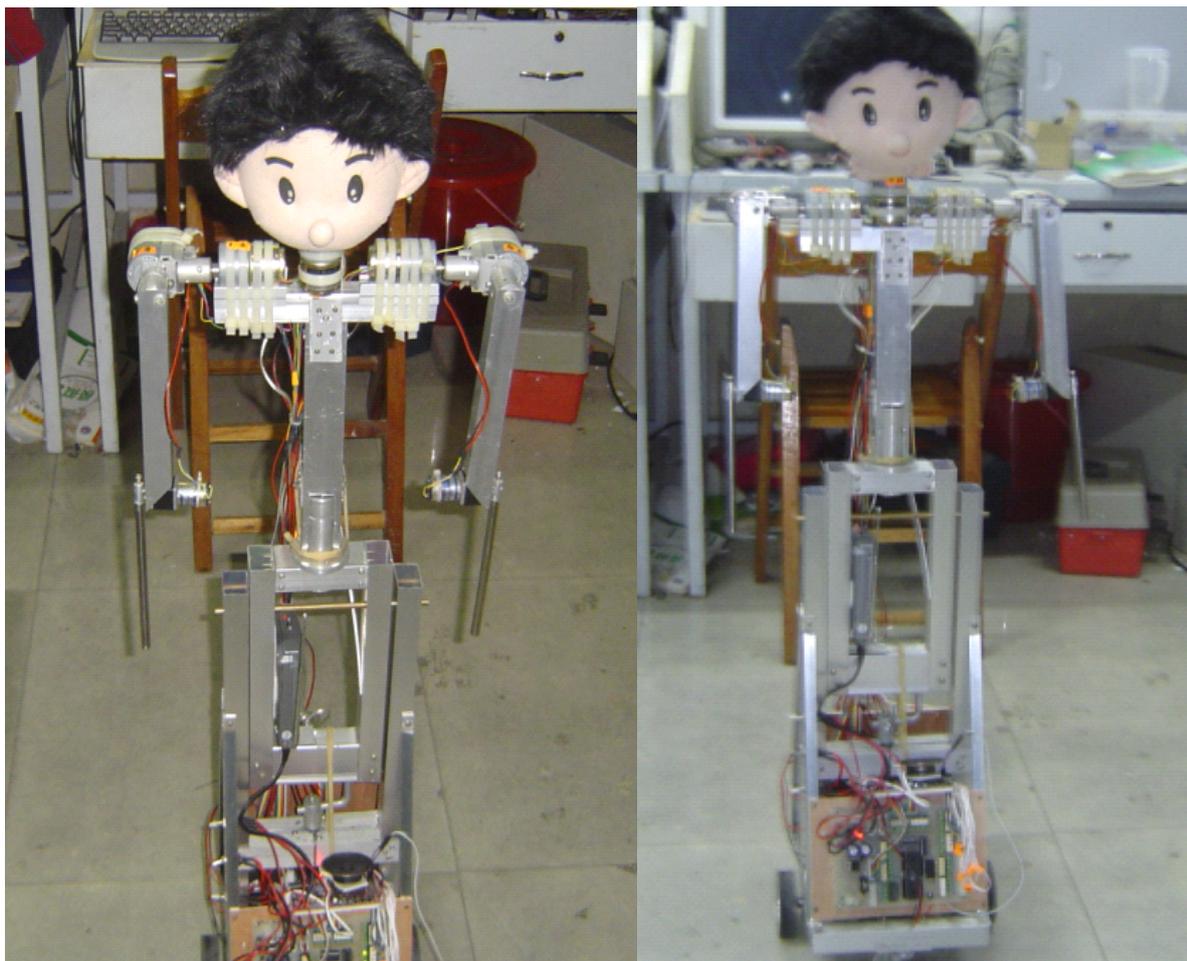


图 6.1 机器人整体效果图

## 7 系统的软件设计

在完成了硬件的设计之后,需要进行相应的软件设计,对机器人的控制的思想,是直接通过软件程序表达的。因此,软件设计是整个设计过程中最重要的部分之一,它关系到舞蹈动作的编辑、存储和执行,关系到舞蹈机器人舞姿的“优美”与否。

从伴舞机器人的执行过程来看,它由单片机的初始化程序,包括外部中断、定时器中断、串行中断、舞蹈动作初始化;舞蹈动作的读取、执行和避障控制程序两大部分组成。其中串行通信的开关控制量,不等于 1,表示单片机进入串行中断等待,等于 1,程序往下执行;机器人开始跳舞的开关,等于 1,表示机器人开始跳舞,否则,机器人保持原来状态。

### 7.1 舞蹈动作的编辑

舞蹈最显著的特征是步调、节奏和动作幅度,反映在机器人上就应该控制机器人的各个关节的电机转动快慢、时间和角度。要使机器人跳出让观众赏心悦目的舞蹈,就应该很好的编排舞蹈动作,充分利用机器人的特点来展现其动作之美。因而,如何使伴

舞机器人的动作快慢变化错落有致并富有节奏感将是控制系统必须解决的关键问题。要设计舞蹈机器人的舞蹈动作，首先需要选择合适的曲子，然后根据曲子的节拍进行动作编排。

一整套的舞蹈动作其实是由一个一个的单独的舞姿构成的，所以设计舞蹈动作就必须先设计好每个舞姿，这时就需要通过电机的滑块来实时观察关节的位置是否合适，若是合适的话，就把这个动作数据添加到数据库模块的动作表中，接着再调试下一个动作如此反复直到把所有的动作都编排好了，那么舞蹈动作部分也就制作好了。这时可以把动作数据表下载到单片机中，由单片机来控制各个关节，顺序执行动作表中的各个动作。同时控制音乐播放器播放舞蹈音乐，那么我们就可以直观观看一到舞蹈机器人在音乐声中翩翩起舞了。要设计出一套完整的舞蹈动作，使机器人能够像舞蹈演员那样在舞台上表演是相当繁琐的过程，需要反复调试，还要通过对人类动作行为作出的细致入微的观察，并能根据机械结构，把人类动作转变成对机器人关节的控制，才能获得完美的控制效果。

## 7.2 主程序设计

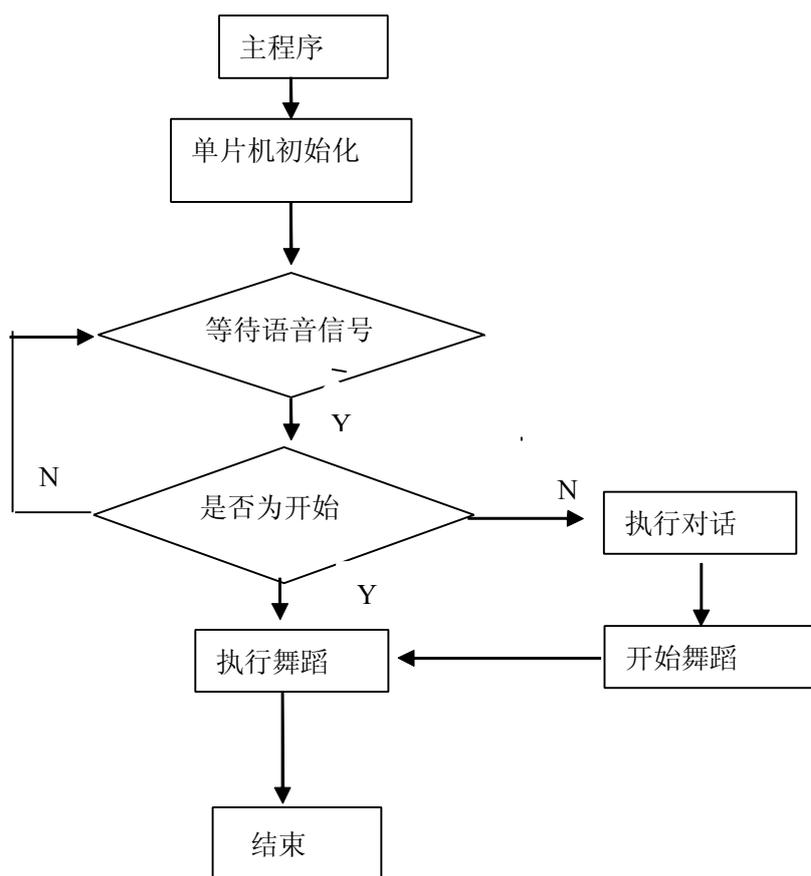


图 7.1 主程序流程图

根据前述的机械结构设计与电路设计，单片机上的控制程序主要包括一个主程序和

相应的中断服务子程序。本设计采用 C 语言按模块化的编程思想编写。所有机器人的动作实现部分均在子函数中完成主函数中只负责对子函数的调用如此便保证了良好的维护性和清晰的代码结构。主程序中，首先对单片机进行初始化，接着调用脉宽调速函数，使相应电机获得应有的速度。然后进入程序中的主循环，判断当前系统的工作模式是自动状态还是手动状态(设置变量用于标识系统运行的状态，在机器人刚通电时默认进入手动控制状态)。主程序主要负责设置堆栈指针和中断向量.对所用到的特殊寄存器进行初始化和舞蹈动作初始化.主程序软件框图如下：

### 7.3 子程序设计

最主要的是考虑到电机控制信号的产生，采用定时器每隔一定的时间中断以产生控制信号和加载舞蹈动作。利用定时器中断程序可以很容易的产生直流电动机的控制信号。通过外部中断，对机器人前后左右四个方向上的碰撞做出处理，以使机器人不会因为碰撞影响以后动作的执行。对碰撞的处理，实际上就是改变保存舞蹈动作的寄存器，由于一定时间读取一次动作数据，可以理解，对碰撞的处理仅在碰撞后的这个规定时间内。只要对直流电机的控制数据设置恰当，就完全可以满足要求，使其不影响以后动作的执行。以下为电机驱动的子程序和 74HC595 的应用子程序：

```
/******  
/*将显示数据送入 74HC595 内部移位寄存器 */  
/******  
void WR_595(void)  
{  
    uchar j ;  
    temp_595=temp;  
    for (j=0 ;j<16 ;j++)  
    {  
        temp_595=temp_595<<1 ;  
        SDATA_595=CY ;  
        SCLK_595=1 ;                //上升沿发生移位  
        SCLK_595=0 ;  
    }  
}  
void OUT_595(void)  
{  
    RCK_595=0 ;  
    _nop_() ;  
    RCK_595=1 ;                //上升沿将数据送到输出锁存器
```

```
RCK_595=0 ;
}
void start_595(void)
{
    WR_595();
    OUT_595();
    _nop_();
}
/*****/
/* 0--关电机 //关某位用 1 或, 其他位为 0 */
/* 1--正转 //开某位用 0 与, 其他位为 1 */
/* 2--反转 */
/*****/
void moto0(unsigned char s0)
{
    switch(s0)
    {
        case 0: {temp_0=temp; temp_0|=0x0003; temp=temp_0;} break ; //关电机
        case 1: {temp_0=temp; temp_0&=0xfffe; temp=temp_0;} break ; //正转
        case 2: {temp_0=temp; temp_0&=0xfffd; temp=temp_0;} break ; //反转
    }
}
/*****/
//反馈信号读入
void scan(void)
{
    state0_B=state0_A;
    state0_A=in0;
    if(state0_A!=state0_B) //判断是否有电平跳变
    {
        count_0++;
        if(count_0>num_0)
        {
            moto0(0);
            start_595();
            count_0=0;
            num--;
        }
    }
}
```

```
        num_0=0xffff;  
    }  
}  
}
```

## 7.4 语音系统程序设计

### 7.4.1 语音系统程序流程

在具有语音功能的智能机器人的软件设计中，主要有几大方面的内容：语音训练和语音识别、语音模型的存储和机器人的动作控制，其中还包括一些必要的中断服务程序和其它程序，机器人主程序流程图如图 7.2 所示。

在具有语音功能的智能机器人的软件设计中，主要有几大方面的内容：语音训练和语音识别、语音模型的存储和机器人的动作控制，其中还包括一些必要的中断服务程序和其它程序，机器人主程序流程图如图 7.2 所示。

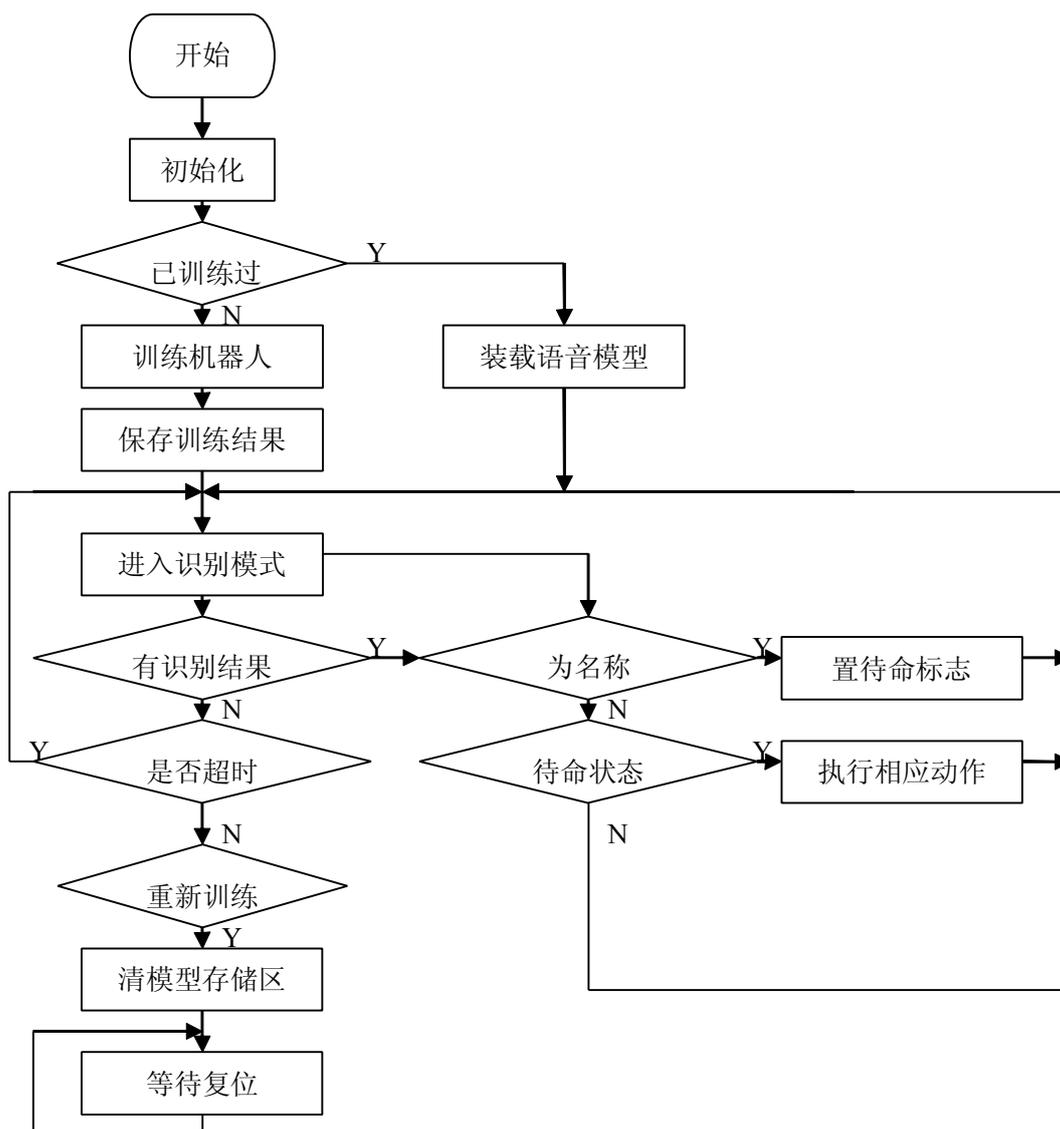


图 7.2 主程序流程图

程序的主要执行过程通过主程序流程图可以清楚看出，下面就对各子程序的编写进行分述。

#### 7.4.2 编程准备

由于控制机器人动作的电机共有 7 个，将占用单片机的 14 个 IO 口，控制 ISD2590 的 IO 口为三个，用于重新训练的按键也占用 1 个 IO 口，所以一共要用单片机的 18 个 IO 口，另外还应考虑机器人可扩展的传感器模块等其它功能模块接口，所以首先对单片机 IO 口进行分配，在分配过程中要注意以下问题：

- (1) 光耦电路输入引脚为每六个脚一组，对应三个电机，因为与单片机 IO 口引脚为排线相连，所以应考虑连线位置
- (2) 检测重新训练键是否按下用到单片机的 IOA 口的 3 脚，所以 IOA 口 0-7 脚只作输入的 IO 口用
- (3) 控制 ISD2590 的三个 IO 口也应考虑连线过程中不与上面两者冲突

所以单片机 IO 口具体分配见表 7.1：

表 7.1：IO 口分配表

IOA 15	ISD2590 启动
IOA 14	ISD2590 停止
IOA 7	ISD2590 的 EOM 输入信号
IOA 2	重新训练按键

#### 7.4.3 子程序的编写

##### 7.4.3.1 语音识别程序

语音识别可分为特定语音识别和非特定语音识别两种，特定语音识别需要使用者先进行训练，非特定语音识别可识别任何人的声音，鉴于开发时间和难度，本次设计采用特定语音识别的方式。语音识别的原理如图 7.3。

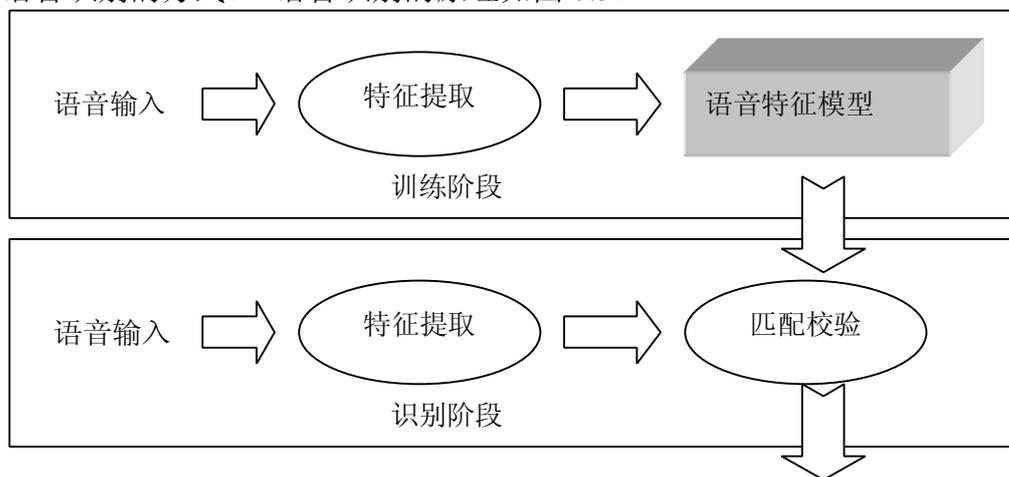


图 7.3 语音识别原理

61 单片机实现语音识别的过程包括训练部分与识别部分，以及在训练、识别过程中的中断的情况，可见图 7.4。

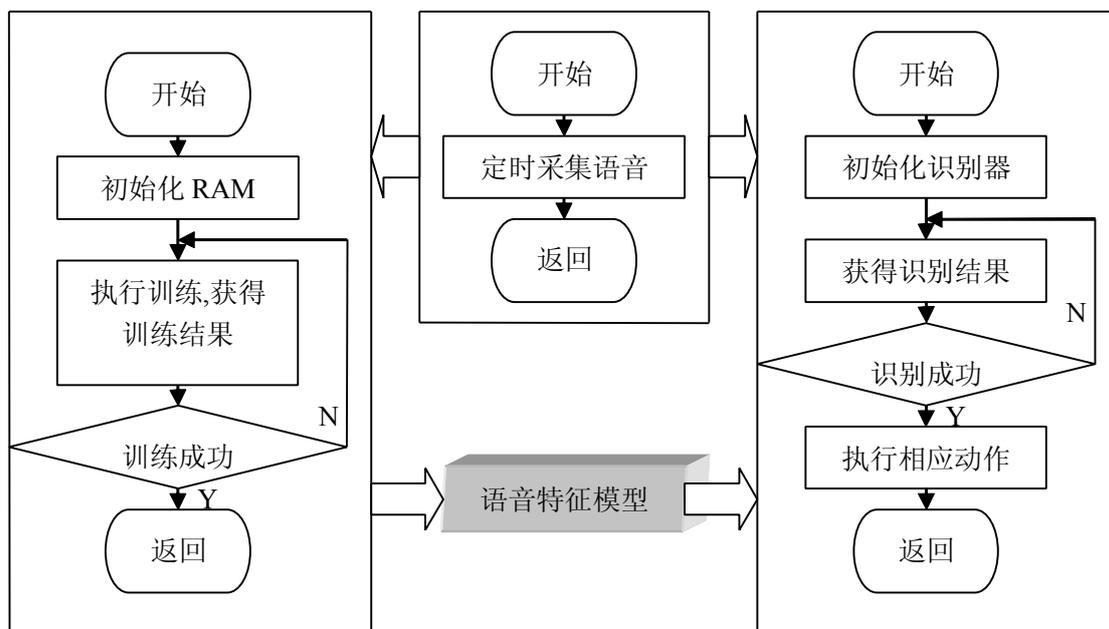


图 7.4 61 单片机识别流程

实现语音功能程序的编写主要依靠凌阳公司提供的 API，使用该 API 需要在程序中链接语音库文件：“bsrv222SDL.lib”和包含头件：“bsrSD.h”。

在语音训练的程序中主要用到两个 API 函数，分别为：

- (1) BSR\_Train();
- (2) BSR\_ExportSDWord();

BSR\_Train 函数在训练语音时使用，带有两个参数：所训练的该条语音的序号和需训练的次数。BSR\_ExportSDWord 函数在每五条语音命令（为一组）训练成功后调用，用于将这一组语音命令的特征模型数据导出到一个临时自动创建为 100 个字节的数组 BSR\_SDModel[] 中，从而可通过这个数组将语音特征模型数据写入 Flash 单元进行存储。

语音识别的程序主要用四个 API 函数，分别为：

- (1) BSR\_InitRecognizer();
- (2) BSR\_GetResult();
- (3) BSR\_DeleteSDGroup(int SDGroupNo);
- (4) BSR\_ImportSDWord();

其中，BSR\_InitRecognizer 函数用于初始化辨识器，该函数在主程序中被调用时，辨识器便打开 8K 采样率的 FIQ\_TMA 中断并开始将采样的语音数据填入辨识器的数据

队列中。BSR\_DeleteSDGroup 用于将 RAM 空间中所有的特征模型数据删除，即清除内存。BSR\_ImportSDWord 的作用同 BSR\_ExportSDWord 相类似，是将已训练过的一组语音特征模型数据导入到数组 BSR\_SDMModel[] 中，采样结果在这里与原特征模型数据进行匹配校验。

在 BSR\_SDMModel[] 中所存储数据为 100 字节，所以 61 单片机在语音识别时一次最多只能同时识别五条语音命令，但 61 单片机的 flash 闪存为 32K，可储存较多的语音模型数据，所以在本次设计中希望可以使机器人能识别更多条的语音命令，这里采用了一个比较巧妙的方法来实现，首先将命令分为 5 条一组（共多组）存入 flash 闪存中，第一组命令作为触发命令组，在第一次识别时根据识别的命令载入相应的组的特征模型数据，在随后的第二次识别中在所载入的这一组数据中进行识别比较，流程如图 7.5 所示。

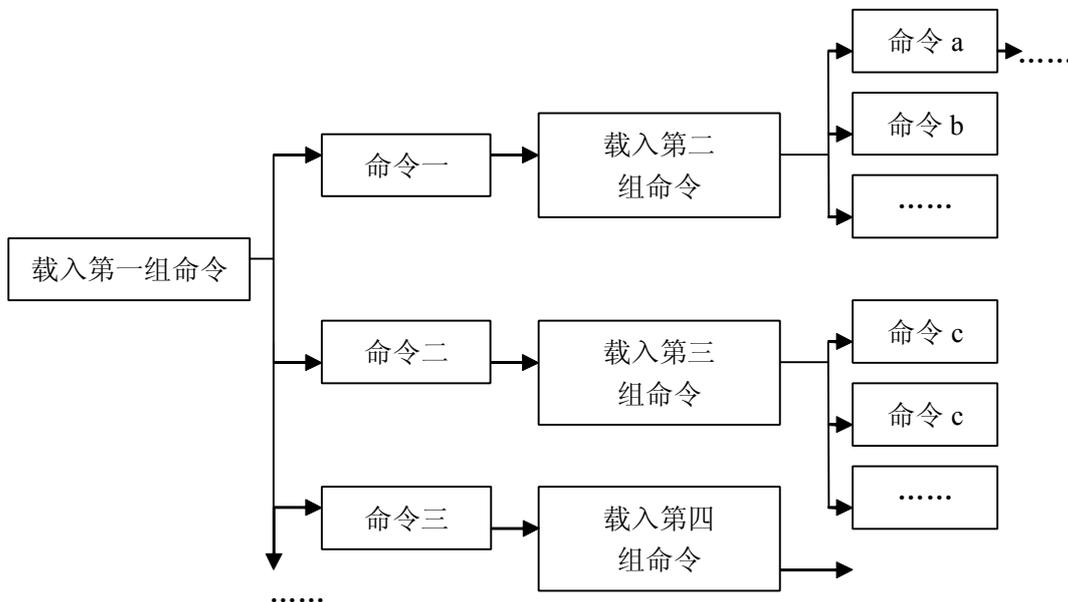


图 7.5 多条语音识别流程

这种通过不同的触发命令载入相应的分组数据进行识别的方法在理论上可以达到识别任意条语音命令，本次设计中做了 15 条语音命令的识别。15 条语音命令分为三组存放，第一组的第一条命令是机器人的待命状态指令，第二条命令即为导入第二组数据，第三条命令为导入第三组数据。

## 8 系统功能测试

### 8.1 硬件检测

做好机器人的硬件部分和软件部分之后，要让机器人实现软硬结合，就必须经过调试，才能够达到课题的要求。在进行总体调试之前，我进行了硬件的检测：

(1) 首先是采用万用表进行硬件的检查，由于本硬件需要焊接的器件较多，在焊接

的过程中可能没有连接上，元器件的好坏，是否正常工作等。

(2) 由于单片机的工作电压是 5V，所以先接入 5V 的电压看硬件电路板中的发光二极管是否亮来判断电路是否接好。该部分主要是判断单片机能否驱动直流电机。为了提高效率，我先编简单的程序只控制直流电机前进跟后退。先判断光电耦合器是否工作。调试时先不接电机驱动部分，而是把光耦的输出接到 LED 发光二极管，看 LED 是否会闪动。这部分调试过程中发现有一个亮着的 LED 不会闪动，检查了整个电路没有发现问题，于是我把几个光耦对换了位置，发现原来不闪动的 LED 会闪了，但是对换光耦位置的另一个 LED 出现了同样的问题。于是我判断是光耦的问题，换了一个新的，通电后十二个 LED 都会闪动，说明电路正常了，接上电机驱动部分，把直流电机的接线接好，通电后直流电机成功实现正、反转。而电机的驱动电压是 12V，因此在使用之前也要用 12V 的电源接上电机，判断电机是否可用。

## 8.2 软件调试

完成以上的检测之后就进行软件的调试。调试伴舞机器人是一个相当繁琐的过程，需要反复调试对机器人多个关节直流电机的控制，而编排舞蹈则是把调试好的舞蹈动作按一定顺序组合起来。因此在调试过程中把编好的每个子程序分别进行调试。

本设计我选择的是单一的电源供电，但是在调试初期我先双电源开始调试，发现所有的开关器件很难稳定地给单片机送正确值，只有当更换新电池或机器人刚刚启动时才会很准确，经过思考，这都是开关元件消耗电量很大的原因，一旦电池电量不足其工作将会萎靡不振，于是我们将所有开关器件划成一个部分，由一块电池专门供电。

在所有电源开启后，系统即进入给定时间倒计时状态，而倒计时完毕之后，有时电机不能正常启动。经分析，这是属于正常情况，因为其启动需要克服很大的惯性，而其电源又不能在瞬间提供很大的电流，所以就没有动静。此时，关掉单片机电源，再重新打开后即可进入正常状态去正常运行。

从调试过程中我深深认识到科学的严谨，任何细微的错误都可能导致最后的失败。因此我们必须对每一个环节都要求完美。

最后就是连起来所有的子程序，整体调试。

## 9 结论

机器人学已经发展成为一门综合运用机械、电气、计算机、产业和制造业等多种工程学科基础知识和创造能力的举足轻重的学科。无需夸张其特殊地位，机器人学也不愧为引人入胜、前途无量的研究领域。机器人设计是对各项科学技术的综合运用。在长达十周的机器人机械设计及制作过程中，我深刻体会到科技技术的无穷魅力，在浩瀚的知识中遨游才感觉自己的才疏学浅。通过对以往的各项科学技术的资料的阅览，加上平时在日常生活中对一些机械机构的经验以及自己一些有创造性的机械设计理念，完成了实

物制作和电气控制。

“掌握信息，实事求是，实践中创新”，这是我在这次毕业设计的体会。作为一名设计者，首要目的就是要清楚设计的任务是什么，有什么样的信息资料作为设计背景，是否有参照，只有充分掌握这些信息，才能有的放矢，达到设计目的。作为工程技术类的设计不仅是理论上的论证而且应当考虑实际的条件能否达到设计要求。空中造楼阁只是彩泡，绚丽却不实用。在零件加工装配过程中，有的问题在设计中没有预期到的，如何去处理？返工重来？成本？工期？能否成功？一串问题会出现。当我转换模式，局部改动设计，经过分析计算，在不改变部件功能的情况下，局部修改尺寸或添加辅助零件有时能获得更好的效果。我想这也是创新，不拘泥。

在机器人的制作过程中，几乎所有的东西从头来过，从方案论证到具体实施，经验是我们本科生最稀缺的资源。有些设计上，所以采取了降低了理论高度，采取简单易行的方案，量体裁衣来实现任务。很多现实的问题也阻碍了设计的难度提高。比如电机性能与价格是成正比的，考虑到成本问题，只能选择性价比适中的电机，并且加辅助调速的光码盘来提高走路的精度。由于是实物设计，所以我在设计的过程中考虑最多的是实际制造的问题，因此在设计理念上受到了诸多的限制。同时自身的知识和经验储备不足，无法将自己的设计理念完全体现在实物上。但是正是这些诸多的因素是我懂得了，作为一名设计者如何去做设计，如何来实现设计要求。

## 谢 辞

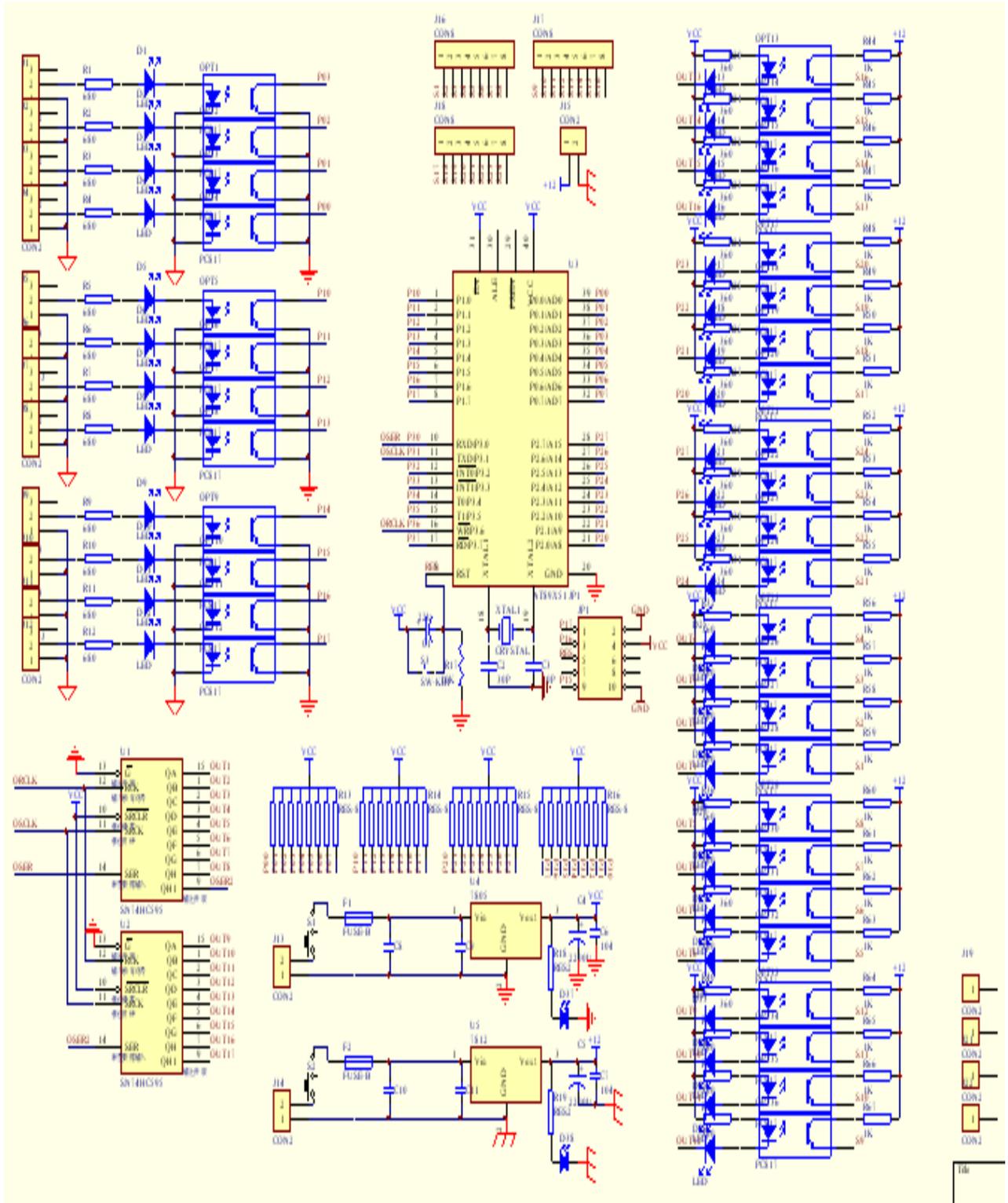
在此设计制作过程中，得到了我的指导画龙点睛式的指导，使我在设计过程中开阔了思路，少走了许多不必要的弯路。感谢张震在整个制作过程的合作。在零件加工过程中，实验室的唐亮给予了热情的帮助，在仪器设备上提供了最大程度的支持。在此，对以上老师的指导和关怀表示感谢和敬意。感谢所有参加机器人大赛的同学在合作的过程中给我信心和勇气，使我跨过一道道难关，顺利完成了这个设计任务。感谢领导给予机器人大赛组的关怀和鼓励。感谢这么多年来教育我的老师们，是他们的谆谆教导使我有机会用自己的学识来展示我的实力和智慧，使我对走向工作岗位的能力更有信心。

## 参考文献

- [1] 黄冰. 微机原理及应用[M]. 重庆: 重庆大学出版社, 2001.
- [2] 蔡自兴. 机器人学[M]. 北京: 清华大学出版社, 2000. 9.
- [3] 周远清, 张自兴, 许万雍, 贾培发等. 智能机器人系统[M]. 北京: 清华大学出版社, 1988.
- [4] 计算机科学与技术百科全书(选编本)[J]. 清华大学出版社. 2002.
- [5] 侯朝帧. 微机与单片机应用基础[M], 北京: 北京理工大学出版社, 1995.
- [6] 余永权. 单片机应用系统的功率接口技术[M]. 北京: 北航出版社, 1991.
- [7] 罗宜晶. 多媒体计算机软件大全[J]. 成都: 电子科技大学出版, 1991.
- [8] 高海生等. 单片机应用技术大全[M]. 成都: 西南交大出版社, 1996.
- [9] 何立民. 单片机应用技术选编 1/2/3[M]. 北京: 北航出版社, 1992.
- [10] DALAS Semiconductor. Atomic Identification Data Book[M]. 1995.
- [11] 李秉操. 单片机接口技术及其在单片机的应用[M]. 陕西: 陕西电子编辑部, 1993.
- [12] 陈汝全. 微机与单片机接口及应用[M]. 成都: 电子科技大学出版, 1996.
- [13] 林建, 李湍. 实用电子电路大全(一 二 三)[D]. 电子工业出版社, 1993.
- [14] 刘甘娜. 微机原理及接口技术[M]. 西安: 西安交通大学出版社, 1996.
- [15] 谢自美. 电子线路设计. 实验. 测试[M]. 湖北: 华中理工大学出版社; 2000
- [16] 马忠梅. 单片机的C语言应用程序设计[M]. 北京: 北京航空航天大学出版社, 2000
- [17] 徐惠民. 单片微型计算机原理接口及应用[M]. 北京: 北京航空航天大学出版社, 2001
- [18] 张培仁. MCS-51C 语言编程[M]. 北京: 清华大学出版社, 2000

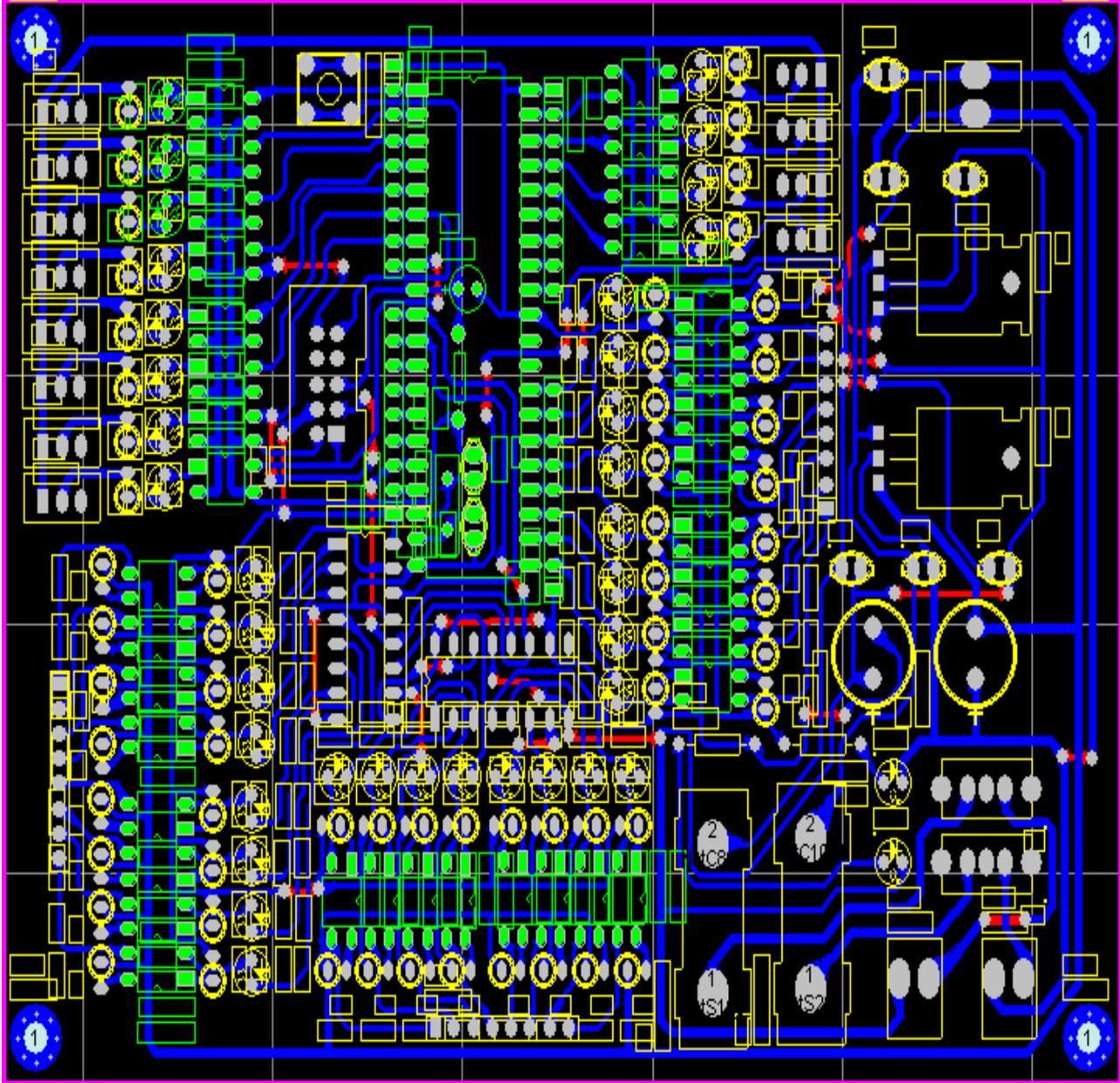
## 附录

附录 1 主控板电路图



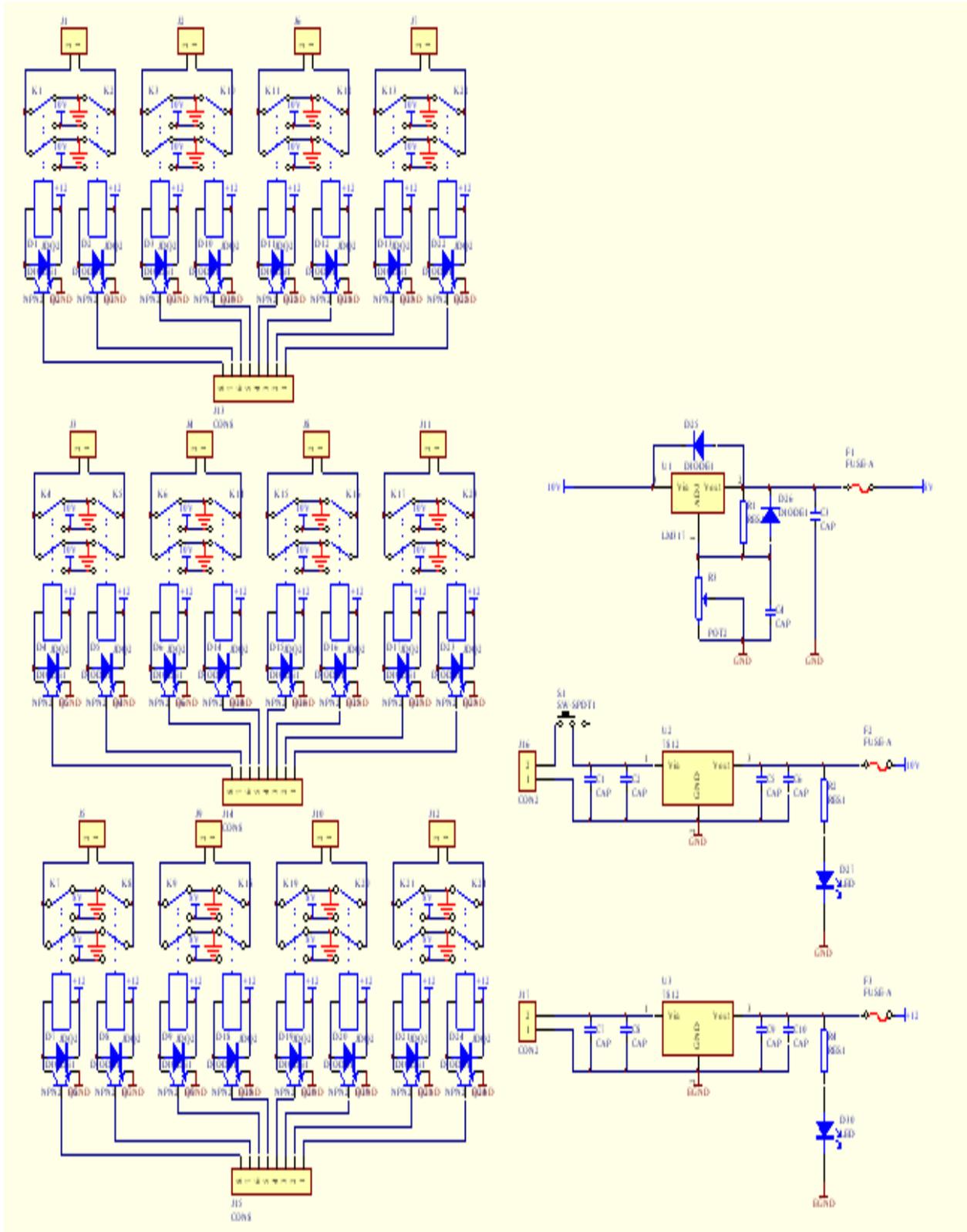
[点击这里](#) 获赠价值数万元的 200 多套原创技术方

附录 2 主控板 PCB 图



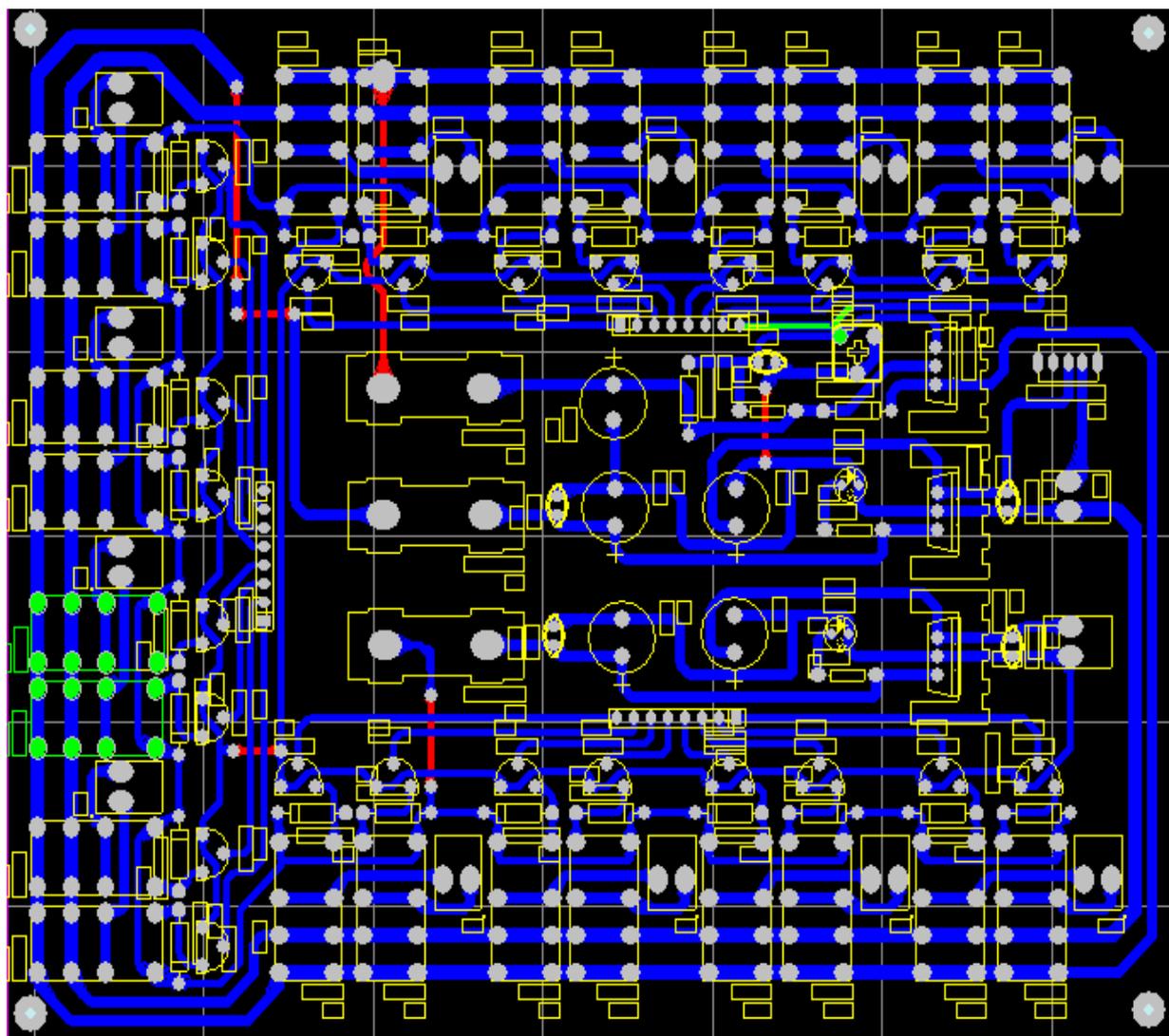
[点击这里](#) 获赠价值数万元的 200 多套原创技术方

附录 3 继电器电路图



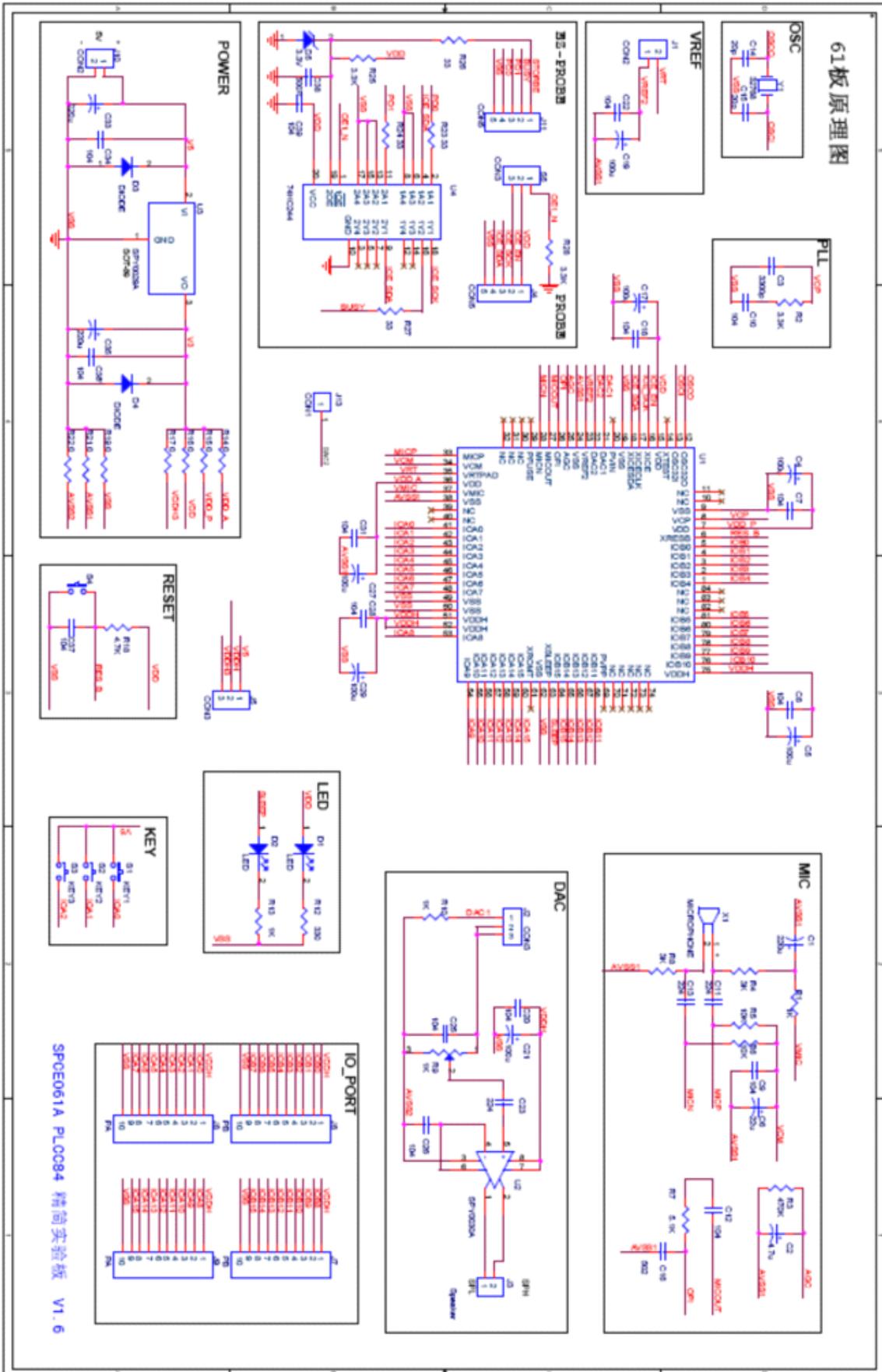
[点击这里](#) 获赠价值数万元的 200 多套原创技术方

附录3 继电器 PCB 图



[点击这里](#) 获赠价值数万元的 200 多套原创技术方

附录 4 61 板电路原理图



点击这里 获赠价值数万元的 200 多套原创技术方

## 附录 5 程序清单

```
/*
#include <AT89X55.H>
//51 芯片管脚定义头文件
#include <intrins.h>
//内部包含延时函数 _nop_()
#define uchar unsigned char
#define uint unsigned int

sbit SDATA_595=P3^0; //串行数据输入
sbit SCLK_595 =P3^1; //移位时钟脉冲
sbit RCK_595 =P3^6; //输出锁存器控制脉冲
uchar run=0; //步骤标记
uchar num=0; //同时工作的电数
uchar i;
sbit in0=P1^0;
sbit in1=P1^1;
sbit in2=P1^2;
sbit in3=P1^3;
sbit in4=P1^4;
sbit in5=P1^5;
sbit in6=P1^6;
sbit in7=P1^7;
sbit in8=P0^0;
unsigned char state0_A=0; //定义电机 0 的状态 A
unsigned char state0_B=0; //定义电机 0 的状态 B
unsigned char state1_A=0; //定义电机 1 的状态 A
unsigned char state1_B=0; //定义电机 1 的状态 B
unsigned char state2_A=0; //定义电机 2 的状态 A
unsigned char state2_B=0; //定义电机 2 的状态 B
unsigned char state3_A=0; //定义电机 3 的状态 A
unsigned char state3_B=0; //定义电机 3 的状态 B
unsigned char state4_A=0; //定义电机 4 的状态 A
unsigned char state4_B=0; //定义电机 4 的状态 B
unsigned char state5_A=0; //定义电机 5 的状态 A
unsigned char state5_B=0; //定义电机 5 的状态 B
unsigned char state6_A=0; //定义电机 6 的状态 A
unsigned char state6_B=0; //定义电机 6 的状态 B
//unsigned char state7_A=0; //定义电机 7 的状态 A
//unsigned char state7_B=0; //定义电机 7 的状态 B
//unsigned char state8_A=0; //定义电机 8 的状态 A
//unsigned char state8_B=0; //定义电机 8 的状态 B
*/
```

```
unsigned int temp=0xffff;
unsigned int temp_595;
unsigned int temp_0;
unsigned int temp_1;
unsigned int temp_2;
unsigned int temp_3;
unsigned int temp_4;
unsigned int temp_5;
unsigned int temp_6;
unsigned int temp_7;
unsigned char P17_temp;
unsigned char P17_flag;
unsigned int num_0=0xffff;
unsigned int num_1=0xffff;
unsigned int num_2=0xffff;
unsigned int num_3=0xffff;
unsigned int num_4=0xffff;
unsigned int num_5=0xffff;
unsigned int num_6=0xffff;
//unsigned int num_7=0xffff;
//unsigned int num_8=0xffff;
unsigned int count_0=0;
unsigned int count_1=0;
unsigned int count_2=0;
unsigned int count_3=0;
unsigned int count_4=0;
unsigned int count_5=0;
unsigned int count_6=0;
//unsigned int count_7=0;
//unsigned int count_8=0;
```

```
void WR_595();
void OUT_595();
void scan();
void start_595(void);
void stop(void);
void moto0(unsigned char s0);
void moto1(unsigned char s1);
void moto2(unsigned char s2);
void moto3(unsigned char s3);
void moto4(unsigned char s4);
void moto5(unsigned char s5);
```

```
void moto6(unsigned char s6);
void moto7(unsigned char s7);
void scan0(unsigned int n0);
void scan1(unsigned int n1);
void scan2(unsigned int n2);
void scan3(unsigned int n3);
void scan4(unsigned int n4);
void scan5(unsigned int n5);
void scan6(unsigned int n6);
void scan7(unsigned int n7);
```

```
void delay50ms(int ms);
```

```
/******  
/*将显示数据送入 74HC595 内部移位寄存器 */  
/******
```

```
void WR_595(void)
{
    uchar j ;
    temp_595=temp;
    for (j=0 ;j<16 ;j++)
    {
        temp_595=temp_595<<1 ;
        SDATA_595=CY ;
        SCLK_595=1 ;           //上升沿发生移位
        _nop_() ;
        _nop_() ;
        SCLK_595=0 ;
    }
}
```

```
/******  
/*将移位寄存器内的数据锁存到输出寄存器并显示 */  
/******
```

```
void OUT_595(void)
{
    RCK_595=0 ;
    _nop_() ;
    _nop_() ;
    RCK_595=1 ;           //上升沿将数据送到输出锁存器
    _nop_() ;
    _nop_() ;
}
```

```
_nop_();
RCK_595=0;
}

/*****
/* 595 开启子程序 */
*****/
void start_595(void)
{
    WR_595();
    OUT_595();
    _nop_();
}
/*****
/* 停止子程序 */
*****/
void stop(void)
{
    temp=0xffff;           //取数据
    WR_595();             //把数据写入 595 的寄存器
    OUT_595();
    _nop_();
}
/*****
/* 延时子程序 */
*****/
void delay50ms(int ms)
{
    int k;
    while(ms--)
    {
        for(k=12500; k>0; k--)
        {
            _nop_();
            _nop_();
            _nop_();
            _nop_();
        }
    }
}
/*****
/* 开关电机 */
/* 0--关电机 //关某位用 1 或, 其他位为 0 */
*****/
```

```
/* 1--正转 //开某位用 0 与, 其他位为 1 */
/* 2--反转 */
/*****/
void moto0(unsigned char s0)
{ switch(s0)
  { case 0: {temp_0=temp; temp_0|=0x0003; temp=temp_0;} break ; //关电机
    case 1: {temp_0=temp; temp_0&=0xfffe; temp=temp_0;} break ; //正转
    case 2: {temp_0=temp; temp_0&=0xfffd; temp=temp_0;} break ; //反转
  }
}
void moto1(unsigned char s1)
{ switch(s1)
  { case 0: {temp_1=temp; temp_1|=0x000c; temp=temp_1;} break ; //关电机
    case 1: {temp_1=temp; temp_1&=0xffb; temp=temp_1;} break ; //正转
    case 2: {temp_1=temp; temp_1&=0xff7; temp=temp_1;} break ; //反转
  }
}
void moto2(unsigned char s2)
{ switch(s2)
  { case 0: {temp_2=temp; temp_2|=0x0030; temp=temp_2;} break ; //关电机
    case 1: {temp_2=temp; temp_2&=0xffef; temp=temp_2;} break ; //正转
    case 2: {temp_2=temp; temp_2&=0xffd; temp=temp_2;} break ; //反转
  }
}
void moto3(unsigned char s3)
{ switch(s3)
  { case 0: {temp_3=temp; temp_3|=0x00c0; temp=temp_3;} break ; //关电机
    case 1: {temp_3=temp; temp_3&=0xffb; temp=temp_3;} break ; //正转
    case 2: {temp_3=temp; temp_3&=0xff7; temp=temp_3;} break ; //反转
  }
}
void moto4(unsigned char s4)
{ switch(s4)
  { case 0: {temp_4=temp; temp_4|=0x0300; temp=temp_4;} break ; //关电机
    case 1: {temp_4=temp; temp_4&=0xffe; temp=temp_4;} break ; //正转
    case 2: {temp_4=temp; temp_4&=0xffd; temp=temp_4;} break ; //反转
  }
}
void moto5(unsigned char s5)
{ switch(s5)
  { case 0: {temp_5=temp; temp_5|=0x0c00; temp=temp_5;} break ; //关电机
    case 1: {temp_5=temp; temp_5&=0xffb; temp=temp_5;} break ; //正转
    case 2: {temp_5=temp; temp_5&=0xff7; temp=temp_5;} break ; //反转
  }
}
```

```
    }
}
void moto6(unsigned char s6)
{   switch(s6)
    {   case 0: {temp_6=temp; temp_6|=0x3000; temp=temp_6;} break ; //关电机
        case 1: {temp_6=temp; temp_6&=0xefff; temp=temp_6;} break ; //正转
        case 2: {temp_6=temp; temp_6&=0xdfff; temp=temp_6;} break ; //反转
    }
}
void moto7(unsigned char s7)
{   switch(s7)
    {   case 0: {temp_7=temp; temp_7|=0xc000; temp=temp_7;} break ; //关电机
        case 1: {temp_7=temp; temp_7&=0xbfff; temp=temp_7;} break ; //正转
        case 2: {temp_7=temp; temp_7&=0x7fff; temp=temp_7;} break ; //反转
    }
}

/*****
/*   脉冲电机控制                               */
/*   num_i--输入电机的转动角数                 */
/*   state0_B--前一个电平状态                  */
/*   state0_A--当前电平状态                    */
*****/
void scan0(unsigned int n0)
{   num_0=n0; }
void scan1(unsigned int n1)
{   num_1=n1; }
void scan2(unsigned int n2)
{   num_2=n2; }
void scan3(unsigned int n3)
{   num_3=n3; }
void scan4(unsigned int n4)
{   num_4=n4; }
void scan5(unsigned int n5)
{   num_5=n5; }
void scan6(unsigned int n6)
{   num_6=n6; }
/* void scan7(unsigned int n7)
{   num_7=n7; }
void scan8(unsigned int n5)
{   num_5=n5; } */

void scan(void)
```

```
{
/***** 第 0 个扫描 *****/
state0_B=state0_A;
_nop_();
state0_A=in0;
if(state0_A!=state0_B) //判断是否有电平跳变
{
count_0++;
if(count_0>num_0)
{ moto0(0);
start_595();
count_0=0;
num--;
num_0=0xffff;
}
}
/***** 第一个扫描 *****/
state1_B=state1_A;
_nop_();
state1_A=in1;
if(state1_A!=state1_B) //判断是否有电平跳变
{ count_1++;
if(count_1>num_1)
{ moto1(0);
start_595();
count_1=0;
num--;
num_1=0xffff;
}
}
/***** 第二个扫描 *****/
state2_B=state2_A;
_nop_();
state2_A=in2;
if(state2_A!=state2_B) //判断是否有电平跳变
{ count_2++;
if(count_2>num_2)
{ moto2(0);
start_595();
count_2=0;
num--;
num_2=0xffff;
}
}
```

```
    }
    /******* 第三个扫描 *****/
    state3_B=state3_A;
    _nop_();
    state3_A=in3;
    if(state3_A!=state3_B) //判断是否有电平跳变
    { count_3++;
      if(count_3>num_3)
      { moto3(0);
        start_595();
        count_3=0;
        num--;
        num_3=0xffff;
      }
    }
    /******* 第四个扫描 *****/
    state4_B=state4_A;
    _nop_();
    state4_A=in4;
    if(state4_A!=state4_B) //判断是否有电平跳变
    { count_4++;
      if(count_4>num_4)
      { moto4(0);
        start_595();
        count_4=0;
        num--;
        num_4=0xffff;
      }
    }
    /******* 第五个扫描 *****/
    state5_B=state5_A;
    _nop_();
    state5_A=in5;
    if(state5_A!=state5_B) //判断是否有电平跳变
    { count_5++;
      if(count_5>num_5)
      { moto5(0);
        start_595();
        count_5=0;
        num--;
        num_5=0xffff;
      }
    }
  }
```

```
/****** 第六个扫描 ******/
state6_B=state6_A;
_nop_();
state6_A=in6;
if(state6_A!=state6_B) //判断是否有电平跳变
{
    count_6++;
    if(count_6>num_6)
    {
        P2=0xff;
        count_6=0;
        num--;
        num_6=0xffff;
    }
}
/****** 第七个扫描 ******/
state7_B=state7_A;
_nop_();
state7_A=in7;
if(state7_A!=state7_B) //判断是否有电平跳变
{
    count_7++;
    if(count_7>num_7)
        start_595();
    {
        moto7(0);
        count_7=0;
        num--;
        num_7=0xffff;
    }
}
/****** 第八个扫描 ******/
state8_B=state8_A;
_nop_();
state8_A=in8;
if(state8_A!=state8_B) //判断是否有电平跳变
{
    count_8++;
    if(count_8>num_5)
    {
        P2=0xff;
        count_8=0;
        num--;
        num_5=0xffff;
    }
}
}

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

```
/* 主程序 */
/*****/
void main(void)
{
    SCLK_595=0 ;
    RCK_595=1 ;
    stop();
    run=0;
    P17_flag=1;
    while(1)
    {
        while(P17_flag)
        {
            P17_temp=P1;
            P17_temp |=0x7f;
            if(P17_temp==0x7F)
                P17_flag=0;
        }

        switch(run)
        {
            case 0: //0-1 秒,双手向前伸直, 与身体成 90 度
                num=2;
                moto0(1); scan0(950); // 电机 0 正转 950
                moto1(1); scan1(950); // 电机 1 正转 950
                start_595();
                while(num>0) scan();
                delay50ms(1);
                num=2; //2-4 秒,双手向侧展开与肩水平
                moto2(1); scan2(950); // 电机 2 正转 950
                moto3(2); scan3(950); //电机 3 反转 950
                start_595();
                while(num>0) scan();
                delay50ms(1);
                run++;
                break ;

            case 1: //把手放平
                num=2; //正在工作的电数 2
                moto0(2); scan0(950);
                moto1(2); scan1(950);
                start_595();
                while(num>0) scan();
                delay50ms(1);
        }
    }
}
```

```
num=1; //放下右手
moto3(1); scan3(950);
start_595();
while(num>0) scan();
delay50ms(1);
num=1; //放下左手
moto2(2); scan2(950);
start_595();
while(num>0) scan();
delay50ms(1);
run++;
break ;
case 2: //5-8 秒, 右手向前伸直, 与身体成 90 度,同时头向
右 90 度
num=1;
moto0(1); scan0(950);
start_595();
while(num>0) scan();
delay50ms(1);
run++;
break ;
case 3: //右手向侧展开与肩水平, 然后右手再向前伸直同
时头向左 90 度
num=2;
moto3(2); scan3(950);
moto4(2); scan4(600); // 头向右
start_595();
while(num>0) scan();
delay50ms(1);
num=2;
moto1(1); scan1(950);
moto4(1); scan4(600); //向左 90 度
start_595();
while(num>0) scan();
delay50ms(1);
num=2;
moto2(1); scan2(950);
moto4(1); scan4(600); //向左 90 度
start_595();
while(num>0) scan();
delay50ms(1);
run++;
break ;
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

case 4: //9-10 秒，保持前面动作同时左手向后伸，与身体成 60 度

```
num=3;
moto1(2); scan1(1450);
moto2(2); scan2(950);
    moto3(1);    scan3(950);
// moto4(2);    scan4(600);
start_595();
while(num>0) scan();
delay50ms(1);
run++;
    break ;
```

case 5: //11-14 秒，原地转逆时针一圈

```
moto6(1); //scan6(5000);
    moto7(2);    //scan7(5000);
start_595();
//delay50ms(37);
for(i=3;i>0;i--)
{
P2=0xeb;    //小手往下
delay50ms(3);
P2=0xff;
// delay50ms(1);
P2=0xd7;    //小手往上
delay50ms(6);
P2=0xff;
P2=0xeb;    //小手往下
delay50ms(3);
P2=0xff;
// delay50ms(1);
}
P2=0xf7;    //小手往上
delay50ms(1);
P2=0xff;
delay50ms(1);
num=1;
    moto4(2);    scan4(600);
moto6(0);
moto7(0);
start_595();
while(num>0) scan();
delay50ms(1);
num=2;    //15-17 秒，全部复位
```

```
moto0(2); scan0(950);
  moto1(1); scan1(500);
start_595();
while(num>0) scan();
delay50ms(1);
run++;
  break ;
```

case 6:

```
num=2; //双手做叉腰状
moto2(1); scan2(400);
  moto3(2); scan3(400);
start_595();
while(num>0) scan();
delay50ms(1);
num=1; //上身向左扭 45 度
moto5(1); scan5(1100);
start_595();
while(num>0) scan();
delay50ms(2);
num=1; // 然后再向右扭 90 度
moto5(2); scan5(2200);
start_595();
while(num>0) scan();
delay50ms(2);
num=1; // 然后再向左扭 90 度
moto5(1); scan5(1200);
start_595();
while(num>0) scan();
delay50ms(2);
run++;
  break ;
```

case 7:

//18-19 秒, 全部复位

```
num=2; //双手叉腰状复位
moto2(2); scan2(400);
  moto3(1); scan3(400);
start_595();
while(num>0) scan();
delay50ms(3);
run++;
  break ;
```

case 8:

//20-24 秒, 双手向两侧打开, 稍向后, 同时做弯腰

```
num=4;
```

状

圈

```
moto2(1); scan2(970);
  moto3(2); scan3(970);
moto0(1); scan0(970);
  moto1(1); scan1(970);
start_595();
while(num>0) scan();
delay50ms(1);
num=1; //弯腰状（如飞翔的姿势），身体顺时针旋转 2
```

```
P2=0xfd; scan6(800);
while(num>0) scan();
  //delay50ms(2);
P2=0xeb; //小手往下
delay50ms(2);
P2=0xff;
num=1; // 转头
  moto4(2); scan4(600);
start_595();
while(num>0) scan();
moto6(2); //scan6(5000); //顺时针旋转 1 圈
  moto7(1); //scan7(5000);
start_595();
delay50ms(36);
num=1; // 转头
  moto4(1); scan4(600);
start_595();
while(num>0) scan();
moto6(0);
moto7(0);
start_595();
delay50ms(1);
num=1; //弯腰状复位
P2=0xfe; scan6(800);
while(num>0) scan();
  //delay50ms(2);
P2=0xd7; //小手往上
delay50ms(2);
P2=0xff;
run++;
break ;
```

case 9: // 25-28 秒，双手放下，左手向前伸直，与身体成 90 度，然后向侧展开与肩水平同时把头向左 90 度，然后放下

```
num=4; // 双手放下
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
moto2(2); scan2(950);
  moto3(1); scan3(930);
moto0(2); scan0(940);
  moto1(2); scan1(940);
start_595();
while(num>0) scan();
delay50ms(1);
  num=1; //左手向前伸直与身体成 90 度
moto1(1); scan1(950);
start_595();
while(num>0) scan();
delay50ms(1);
  num=2; //然后向侧展开与肩水平同时把头向左 90 度
moto2(1); scan2(950);
moto4(1); scan4(600);
start_595();
while(num>0) scan();
delay50ms(1);
  num=3; //然后放下
moto2(2); scan2(950);
moto4(2); scan4(600);
moto1(2); scan1(950);
start_595();
while(num>0) scan();
delay50ms(1);
  num=1; //左手向前伸直与身体成 90 度
moto0(1); scan0(950);
start_595();
while(num>0) scan();
delay50ms(1);
  num=2; //然后向侧展开与肩水平同时把头向左 90 度
moto3(2); scan3(950);
moto4(2); scan4(600);
start_595();
while(num>0) scan();
delay50ms(1);
  num=3; //然后放下
moto3(1); scan3(940);
moto4(1); scan4(600);
moto0(2); scan0(950);
start_595();
while(num>0) scan();
delay50ms(1);
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
run++;  
break ;
```

case 10: //29-33 秒，双手向侧展开与肩水平，然后双手向前伸直，与身体成 90 度，再双手向侧展开与肩水平，然后双手向前伸直与身体成 90 度，然后全部复位。

```
num=4; //双手向侧展开与肩水平  
moto2(1); scan2(970);  
moto3(2); scan3(970);  
moto0(1); scan0(980);  
moto1(1); scan1(980);  
start_595();  
while(num>0) scan();  
delay50ms(2);  
num=2; //然后双手向前伸直，与身体成 90 度  
moto2(2); scan2(950);  
moto3(1); scan3(950);  
start_595();  
while(num>0) scan();  
delay50ms(2);  
num=2; //再双手向侧展开与肩水平  
moto2(1); scan2(950);  
moto3(2); scan3(950);  
start_595();  
while(num>0) scan();  
delay50ms(2);  
num=4; //然后双手向前伸直与身体成 90 度 然后全部
```

复位

```
moto2(2); scan2(950);  
moto3(1); scan3(950);  
moto0(2); scan0(950);  
moto1(2); scan1(950);  
start_595();  
while(num>0) scan();  
delay50ms(2);  
run++;  
break ;
```

case 11: //34-38 秒，双手向侧展开与肩水平然后上身向右扭 45 度，再向左扭 90 度，重复一次

```
num=2; //再双手向侧展开与肩水平  
moto2(1); scan2(600);  
moto3(2); scan3(600);  
start_595();  
while(num>0) scan();
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
delay50ms(1);
num=1; //上身向左扭 45 度
moto5(2); scan5(1200);
start_595();
while(num>0) scan();
delay50ms(2);
num=1; // 然后再向右扭 90 度
moto5(1); scan5(2400);
start_595();
while(num>0) scan();
delay50ms(2);
num=1; // 然后再向左扭 45 度
moto5(2); scan5(1300);
start_595();
while(num>0) scan();
delay50ms(2);
run++;
break ;
case 12: //39-43 秒，全部复位，然后双手向侧展开与肩水平，放下，
双手向侧展开与肩水平，放下重复，同时身体逆时针旋转
num=2; //双手放下
moto2(2); scan2(590);
moto3(1); scan3(580);
// moto6(1); //逆时针旋转
// moto7(2);
start_595();
while(num>0) scan();
// delay50ms(2);
moto6(1); //逆时针旋转
moto7(2);
start_595();
for(i=3;i>0;i--)
{
num=2; //再双手向侧展开与肩水平
moto2(1); scan2(800);
moto3(2); scan3(800);
moto6(1); //逆时针旋转
moto7(2);
start_595();
while(num>0) scan();
delay50ms(3);
num=2; //双手放下
moto2(2); scan2(800);
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
        moto3(1);    scan3(800);
    moto6(1);        //逆时针旋转
        moto7(2);
    start_595();
    while(num>0) scan();
    delay50ms(3);
    }
    moto6(0);
    moto7(0);
    start_595();
    delay50ms(1);
    run++;
        break ;
case 13:                //44-47 秒，双手向后伸，与身体成 60 度，然后
前进 2 秒
    num=3;                //
    moto0(2); scan0(500);
        moto1(2);    scan1(500);
    P2=0xfd; scan6(450);
    moto6(2);            //前进
        moto7(2);
    start_595();
    while(num>0) scan();
    P2=0xeb;
    delay50ms(2);
    P2=0xff;
    delay50ms(18);
    num=3;                //弯腰状复位
    P2=0xfe; scan6(450);
    moto0(1); scan0(500);
        moto1(1);    scan1(500);
    moto6(0);
    moto7(0);
    start_595();
    while(num>0) scan();
    P2=0xd7;
    delay50ms(2);
    P2=0xff;
    run++;
        break ;
case 14:                //48-53 秒，双手复位，双手向侧展开与肩水平
同时抬头，然后后退 2 秒
    num=3;                //再双手向侧展开与肩水平
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
    moto2(1); scan2(1000);
    moto3(2); scan3(1000);
    P2=0xfd; scan6(700); //弯腰状
    moto6(1);           //后退
    moto7(1);
    start_595();
    while(num>0) scan();
    delay50ms(20);
    num=1;               //弯腰状复位
    P2=0xfe; scan6(700);
    moto6(0);
    moto7(0);
    start_595();
    while(num>0) scan();
    delay50ms(1);
    run++;
    break ;
case 15:                //54-58 秒，双手复位抬头顺时针转一圈
```

```
    num=2;
    moto2(2); scan2(940); //双手复位
    moto3(1); scan3(920);
    moto6(2);           //顺时针旋转
    moto7(1);
    start_595();
    while(num>0) scan();
    delay50ms(2);
    for(i=3;i>0;i--)
    {
    num=2;              //再双手向侧展开与肩水平
    moto2(1); scan2(800);
    moto3(2); scan3(800);
    moto6(2);           //顺时针旋转
    moto7(1);
    start_595();
    while(num>0) scan();
    delay50ms(2);
    num=2;              //双手放下
    moto2(2); scan2(800);
    moto3(1); scan3(800);
    moto6(2);           //逆时针旋转
    moto7(1);
    start_595();
```

```
while(num>0) scan();  
delay50ms(2);  
}  
moto6(0);  
moto7(0);  
start_595();  
while(num>0) scan();  
//delay50ms(1);  
run++;  
break ;
```

case 16: //59-1 分 03 秒左手向前伸直与身体成 90 度，然后向侧展开后放下，然后左手向前伸直，与身体成 90 度，然后向侧展开，重复

```
num=1;  
moto1(1); scan1(950); //左手向前伸直  
start_595();  
while(num>0) scan();  
delay50ms(1);  
num=1;  
moto2(1); scan2(950); //侧展开  
start_595();  
while(num>0) scan();  
delay50ms(1);  
num=2;  
moto2(2); scan2(930); //放下  
moto1(2); scan1(950);  
start_595();  
while(num>0) scan();  
delay50ms(1);  
num=1; //右手向前伸直与身体成 90 度  
moto0(1); scan0(950);  
start_595();  
while(num>0) scan();  
delay50ms(1);  
num=1; //然后向侧展开与肩水平同时把头向左 90 度  
moto3(2); scan3(950);  
// moto4(2); scan4(600);  
start_595();  
while(num>0) scan();  
delay50ms(1);  
num=2; //然后放下  
moto3(1); scan3(920);  
// moto4(1); scan4(600);  
moto0(2); scan0(950);
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
start_595();
while(num>0) scan();
delay50ms(1);
run++;
break;
case 17: //1 分 04 秒-1 分 06 秒, 头向左摆 45 度然后向
右 90 度, 然后再向左 90 度, 重复
num=1;
moto4(2); scan4(500); //头向左摆 45 度
start_595();
while(num>0) scan();
delay50ms(1);
num=1;
moto4(1); scan4(1000); //头向右 90 度
start_595();
while(num>0) scan();
delay50ms(1);
num=1;
moto4(2); scan4(500); //头向左摆 45 度
start_595();
while(num>0) scan();
delay50ms(1);
run++;
break ;
case 18: //1 分 09 秒-1 分 13 秒, 双手向两侧打开, 稍向后,
同时做弯腰状, (如飞翔的姿势), 身体顺时针旋转 2 圈
num=4;
moto2(1); scan2(1000);
moto3(2); scan3(1000);
moto0(1); scan0(1000);
moto1(1); scan1(1000);
start_595();
while(num>0) scan();
delay50ms(1);
num=1; //弯腰状 (如飞翔的姿势), 身体顺时针旋转 2
圈

P2=0xfd; scan6(700);
while(num>0) scan();
delay50ms(1);
moto6(2); //scan6(5000); //顺时针旋转 2 圈
moto7(1); //scan7(5000);
start_595();
for(i=3;i>0;i--)
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
{
P2=0xeb;          //小手往下
delay50ms(3);
P2=0xff;
// delay50ms(1);
P2=0xd7;          //小手往上
delay50ms(6);
P2=0xff;
P2=0xeb;          //小手往下
delay50ms(3);
P2=0xff;
// delay50ms(1);
}
P2=0xf7;          //小手往上
delay50ms(1);
P2=0xff;
delay50ms(1);
moto6(0);
moto7(0);
start_595();
delay50ms(1);
num=1;            //弯腰状复位
P2=0xfe; scan6(720);
while(num>0) scan();
delay50ms(1);
run++;
break ;

case 19:          //1分14秒-1分17秒，全部复位然后双手向后
伸，与身体成60度同时抬头，然后前进2秒
num=4;           //双手放下
moto2(2); scan2(940);
moto3(1); scan3(910);
moto0(2); scan0(940);
moto1(2); scan1(930);
start_595();
while(num>0) scan();
delay50ms(1);
num=3;           //后双手向后伸，与身体成60度同时，然后前
进2秒
moto0(2); scan0(500);
moto1(2); scan1(500);
P2=0xfd; scan6(700);
moto6(2);        //前进
```

[点击这里](#) 获赠价值数万元的200多套原创技术方

```
        moto7(2);
        start_595();
        while(num>0) scan();
        delay50ms(20);
        num=3;                //弯腰状复位
        P2=0xfe; scan6(700);
        moto0(1); scan0(500);
        moto1(1); scan1(500);
        moto6(0);
        moto7(0);
        start_595();
        while(num>0) scan();
        delay50ms(1);
        run++;
        break ;
    case 20:                    //1 分 18 秒-1 分 23 秒，双手向侧展开与肩水平
    同时抬头，然后后退 2 秒
        num=2;
        moto2(1); scan2(950); //双手向侧展开与肩水平
        moto3(2); scan3(950);
        moto6(1);            //后退 2 秒
        moto7(1);
        start_595();
        while(num>0) scan();
        delay50ms(20);
        num=2;
        moto2(2); scan2(960); //双手复位
        moto3(1); scan3(940);
        moto6(0);
        moto7(0);
        start_595();
        while(num>0) scan();
        delay50ms(1);
        run++;
        break ;
    case 21:                    //1 分 24 秒-1 分 27 秒，右手向前伸直，与身体成 90 度，
    然后向侧展开与肩水平同时把身子向右 60 度然后放下手
        run++;
        break ;
    case 22:                    //1 分 28 秒-1 分 32 秒，左手向前伸直，与身体成 90 度，
    然后向侧展开与肩水平同时把身子向左 60 度然后放下手
        num=1;
        moto1(1); scan1(960); //左手向前伸直
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
start_595();
while(num>0) scan();
delay50ms(1);
num=3;
moto2(1); scan2(950); // 侧展开与肩水平
moto5(2); scan5(500); //向左 60 度
moto4(1); scan4(500); //头向左摆 45 度
start_595();
while(num>0) scan();
delay50ms(1);
num=4;
moto2(2); scan2(960); // 放下手
moto5(1); scan5(500);
moto1(2); scan1(950);
moto4(2); scan4(500); //头向右摆 45 度
start_595();
while(num>0) scan();
delay50ms(1);
//delay50ms(1);
run++;
break ;
```

case 23: //右手向前伸直，与身体成 90 度，同时头向左 90 度，然后右手向开与肩水平，然后右手再向前伸直同时头向右 90 度，重复一次，后复位

```
num=1;
moto0(1); scan0(950); //右手向前伸直
start_595();
while(num>0) scan();
delay50ms(1);
num=3;
moto3(2); scan3(950); // 侧展开与肩水平
moto5(1); scan5(1000); //向右 60 度
moto4(2); scan4(500); //头向右摆 45 度
start_595();
while(num>0) scan();
delay50ms(1);
num=4;
moto3(1); scan3(940); // 放下手
moto5(2); scan5(1000);
moto0(2); scan0(950);
moto4(1); scan4(500); //头向左摆 45 度
start_595();
while(num>0) scan();
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
delay50ms(1);  
//delay50ms(1);  
run++;  
break ;
```

case 24: //1 分 45 秒-1 分 47 秒, 双手向侧展开与肩水平然后弯曲右前臂与肩成 90 度, 然后伸直右手, 把右前臂弯曲与肩成 90 度, 然后再弯曲右前臂重复几次

```
num=4; //双手向侧展开与肩水平  
moto2(1); scan2(990);  
moto3(2); scan3(990);  
moto0(1); scan0(990);  
moto1(1); scan1(990);  
start_595();  
while(num>0) scan();  
delay50ms(1);  
num=2; //然后双手向前伸直, 与身体成 90 度  
moto2(2); scan2(950);  
moto3(1); scan3(940);  
start_595();  
while(num>0) scan();  
delay50ms(1);  
num=2; //然后全部复位  
moto0(2); scan0(960);  
moto1(2); scan1(940);  
start_595();  
while(num>0) scan();  
delay50ms(1);  
run++;  
break ;
```

case 25: //54-58 秒, 双手复位抬头顺时针转一圈

```
moto6(1); //逆时针旋转  
moto7(2);  
start_595();  
while(num>0) scan();  
delay50ms(38);  
moto6(0);  
moto7(0);  
start_595();  
while(num>0) scan();  
delay50ms(1);  
run++;  
break ;
```

case 26: //59-1 分 03 秒左手向前伸直与身体成 90 度, 然后向侧展开后放下, 然后左手向前伸直, 与身体成 90 度, 然后向侧展开, 重复

```
run++;  
break;
```

case 27: //1 分 04 秒-1 分 06 秒, 头向左摆 45 度然后向右 90 度, 然后再向左 90 度, 重复

```
run++;  
break;
```

case 28: //1 分 09 秒-1 分 13 秒, 双手向两侧打开, 稍向后, 同时做弯腰状, (如飞翔的姿势), 身体顺时针旋转 2 圈

```
num=4;  
moto2(1); scan2(970);  
moto3(2); scan3(970);  
moto0(1); scan0(950);  
moto1(1); scan1(960);  
start_595();
```

```
while(num>0) scan();
```

```
// delay50ms(1);
```

```
num=1;
```

//弯腰状 (如飞翔的姿势), 身体顺时针旋转 2

圈

```
P2=0xfd; scan6(700);
```

```
while(num>0) scan();
```

```
delay50ms(1);
```

```
moto6(2); //scan6(5000); //顺时针旋转 1 圈
```

```
moto7(1); //scan7(5000);
```

```
start_595();
```

```
for(i=3;i>0;i--)
```

```
{
```

```
P2=0xeb; //小手往下
```

```
delay50ms(3);
```

```
P2=0xff;
```

```
// delay50ms(1);
```

```
P2=0xd7; //小手往上
```

```
delay50ms(6);
```

```
P2=0xff;
```

```
P2=0xeb; //小手往下
```

```
delay50ms(3);
```

```
P2=0xff;
```

```
// delay50ms(1);
```

```
}
```

```
delay50ms(1);
```

```
moto6(0);
```

```
moto7(0);
```

```
start_595();
```

```
delay50ms(1);
```

```
num=1; //弯腰状复位
P2=0xfe; scan6(710);
while(num>0) scan();
P2=0xf7; //小手往上
delay50ms(1);
P2=0xff;
run++;
break ;
case 29: //1 分 14 秒-1 分 17 秒, 全部复位然后双手向后伸, 与身体成 60 度同时
抬头, 然后前进 2 秒
num=4; // 双手放下
moto2(2); scan2(960);
moto3(1); scan3(930);
moto0(2); scan0(940);
moto1(2); scan1(930);
start_595();
while(num>0) scan();
delay50ms(1);
num=3; //后双手向后伸, 与身体成 60 度同时, 然后前
进 2 秒
moto0(2); scan0(500);
moto1(2); scan1(500);
P2=0xfd; scan6(700);
moto6(2); //前进
moto7(2);
start_595();
while(num>0) scan();
P2=0xeb; //小手往下
delay50ms(3);
P2=0xff;
// delay50ms(1);
P2=0xd7; //小手往上
delay50ms(6);
P2=0xff;
P2=0xeb; //小手往下
delay50ms(3);
P2=0xff;
delay50ms(8);
num=3; //弯腰状复位
P2=0xfe; scan6(720);
moto0(1); scan0(500);
moto1(1); scan1(500);
moto6(0);
```

```
    moto7(0);
    start_595();
    while(num>0) scan();
    P2=0xf7;          //小手往上
    delay50ms(1);
    P2=0xff;
    run++;
    break ;
case 30: //1 分 18 秒-1 分 23 秒，双手向侧展开与肩水平同时抬头，然后后退 2 秒
    num=2;
    moto2(1); scan2(950); //双手向侧展开与肩水平
    moto3(2); scan3(960);
    moto6(1);          //后退 2 秒
    moto7(1);
    start_595();
    while(num>0) scan();
    P2=0xeb;          //小手往下
    delay50ms(3);
    P2=0xff;
    // delay50ms(1);
    P2=0xd7;          //小手往上
    delay50ms(6);
    P2=0xff;
    P2=0xeb;          //小手往下
    delay50ms(3);
    P2=0xff;
    delay50ms(8);
    num=2;
    moto2(2); scan2(940); //双手复位
    moto3(1); scan3(930);
    moto6(0);
    moto7(0);
    start_595();
    while(num>0) scan();
    delay50ms(1);
    run++;
    break ;
case 31: //1 分 24 秒-1 分 27 秒，右手向前伸直，与身体成 90 度，然后向侧展开
与肩水平同时把身子向右 60 度然后放下手
    num=1;
    moto0(1); scan0(950); //右手向前伸直
    start_595();
    while(num>0) scan();
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
delay50ms(1);
num=2;
moto3(2); scan3(960); // 侧展开与肩水平
moto5(1); scan5(1000);
start_595();
while(num>0) scan();
delay50ms(1);
num=3;
moto3(1); scan3(950); //
moto5(2); scan5(1000);
moto0(2); scan0(960);
start_595();
while(num>0) scan();
delay50ms(1);
run++;
break ;
```

case 32: //1 分 28 秒-1 分 32 秒，左手向前伸直，与身体成 90 度，然后向侧展开与肩水平同时把身子向左 60 度然后放下手

```
num=1;
moto1(1); scan1(960); //左手向前伸直
start_595();
while(num>0) scan();
// delay50ms(1);
num=2;
moto2(1); scan2(950); // 侧展开与肩水平
moto5(2); scan5(1000); //向左 60 度
start_595();
while(num>0) scan();
delay50ms(1);
num=3;
moto2(2); scan2(960); // 放下手
moto5(1); scan5(1000);
moto1(2); scan1(940);
start_595();
while(num>0) scan();
delay50ms(1);
run++;
break ;
```

case 33: //右手向前伸直，与身体成 90 度，同时头向左 90 度，然后右手向开与肩水平，然后右手再向前伸直同时头向右 90 度，重复一次，后复位

```
num=1;
moto0(1); scan0(960); //右手向前伸直
start_595();
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
while(num>0) scan();  
// delay50ms(1);  
num=2;  
moto3(2); scan3(950); // 侧展开与肩水平  
moto5(1); scan5(1000); //向右 60 度  
start_595();  
while(num>0) scan();  
delay50ms(1);  
num=3;  
moto3(1); scan3(940); // 放下手  
moto5(2); scan5(1000);  
moto0(2); scan0(950);  
start_595();  
while(num>0) scan();  
delay50ms(1);  
run++;  
break ;
```

case 34: //1 分 45 秒-1 分 47 秒, 双手向侧展开与肩水平然后弯曲右前臂与肩成 90 度, 然后伸直右手, 把右前臂弯曲与肩成 90 度, 然后再弯曲右前臂重复几次

```
num=4; //双手向侧展开与肩水平  
moto2(1); scan2(970);  
moto3(2); scan3(970);  
moto0(1); scan0(1000);  
moto1(1); scan1(990);  
start_595();  
while(num>0) scan();  
delay50ms(1);  
num=2; //然后双手向前伸直, 与身体成 90 度  
moto2(2); scan2(960);  
moto3(1); scan3(950);  
start_595();  
while(num>0) scan();  
delay50ms(1);  
num=2; //然后全部复位  
moto0(2); scan0(950);  
moto1(2); scan1(950);  
start_595();  
while(num>0) scan();  
delay50ms(1);  
run++;  
break ;
```

case 35: //1 分 33 秒-2 分 37 秒, 全部复位, 顺时针一圈, 逆时针一圈  
moto6(2); //scan6(5000); //顺时针旋转圈

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
    moto7(1);    //scan7(5000);
start_595();
delay50ms(38);
moto6(0);
moto7(0);
start_595();
delay50ms(1);
moto6(1); //scan6(5000); //逆时针一圈
    moto7(2);    //scan7(5000);
start_595();
delay50ms(38);
moto6(0);
moto7(0);
start_595();
delay50ms(1);
run++;
    break ;
```

case 36: //2 分 42 秒最后，左手向前伸直，与身体成 90 度然后双手向侧展开与肩水平然后放下

```
    num=1;
moto1(1); scan1(950); //左手向前伸直
start_595();
while(num>0) scan();
delay50ms(1);
    num=1;
moto0(1); scan0(950); //手向前伸直
start_595();
while(num>0) scan();
delay50ms(1);
    num=3;
moto2(1); scan2(960); //双手向侧展开与肩水平
    moto3(2); scan3(960);
P2=0xfd; scan6(800);
start_595();
while(num>0) scan();
delay50ms(5);
    num=5;
moto2(2); scan2(1050); //敬礼
    moto3(1); scan3(1030);
moto1(2); scan1(930);
moto0(2); scan0(940);
P2=0xfe; scan6(800);
start_595();
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
        while(num>0) scan();
        // delay50ms(10);
        run++;
        break ;
    }

    // }

}
while(1);
}
```

## 附录 6 语音程序清单

```
#include "bsrsd.h"
#include "SPCE061V004.H"

#define NAME_ID      0x100
#define COMMAND_ONE_ID  0x101
#define COMMAND_TWO_ID  0x102
#define RSP_INTR      0
#define RSP_NAME      1
#define RSP_FIRE      2
#define RSP_GUARD     3
#define RSP_AGAIN     4
#define RSP_NOVOICE   5
#define RSP_NAMEDIFF  6
#define RSP_CMDDIFF   7
#define RSP_STAR      8
#define RSP_MASTER    9
#define RSP_HERE     10
#define RSP_GUNSHOT   0
#define RSP_PATROL    11
#define RSP_READY     12
#define RSP_COPY      13
#define RSP_NOISY     14
#define RSP_NIHAO     15

//.....全程变量.....
int gActivated = 0; //用于检测是否有触发命令，当有识别出语句为触发名称则该位置 1

int gTriggerRespond[] = {RSP_MASTER, RSP_HERE, RSP_MASTER}; //第一条命令应答
int gComm2Respond[] = {RSP_PATROL, RSP_READY, RSP_COPY}; //第二条命令应答

//用于写入 FLASH 时用到的全局变量
extern unsigned int BSR_SDMoDel[100];
unsigned int commandID; //用来写命令进 FLASH 时用的
unsigned int g_Ret;

extern void ClearWatchDog();

int PlayFlag = 0;
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
void PlayRespond2(int Result)//枪声放音子程序
{
    BSR_StopRecognizer();
    SACM_A2000_Initial(1);
    SACM_A2000_Play(Result, 3, 3);
    SACM_A2000_Volume(15);
    while((SACM_A2000_Status() & 0x0001) != 0)
    {
        SACM_A2000_ServiceLoop();
        ClearWatchDog();
    }
    SACM_A2000_Stop();
    BSR_InitRecognizer(BSR_MIC);
    BSR_EnableCPUIndicator();
}
```

```
void PlayRespond(int Result)//放音子程序
{
    BSR_StopRecognizer();
    SACM_S480_Initial(1);
    SACM_S480_Play(Result, 3, 3);
    SACM_S480_Volume(15);
    while((SACM_S480_Status() & 0x0001) != 0)
    {
        SACM_S480_ServiceLoop();
        ClearWatchDog();
    }
    SACM_S480_Stop();
    BSR_InitRecognizer(BSR_MIC);
    BSR_EnableCPUIndicator();//启动实时监控
}
```

```
int TrainWord(int WordID, int RespondID)//命令训练
{
    int res;
    PlayRespond(RespondID);
    while(1)
    {
        res = BSR_Train(WordID, BSR_TRAIN_TWICE);
        if(res == 0) break;
        switch(res)
        {
```

```
case -1:                                     //没有检测出声音
    PlayRespond(RSP_NOVOICE);
    return -1;
case -2:                                     //需要重新训练一遍
    PlayRespond(RSP_AGAIN);
    break;
case -3:                                     //环境太吵
    PlayRespond(RSP_NOISY);
    return -1;
case -4:                                     //数据库满
    return -1;
case -5:                                     //检测出声音不同
    if(WordID == NAME_ID)    PlayRespond(RSP_NAMEDIFF); //两次输入名称
不同
    else                    PlayRespond(RSP_CMDDIFF); //两次输入命令不同
    return -1;
case -6:                                     //序号错误
    return -1;
}
}
return 0;
}

int main()
{
    int *p, BS_Flag=0, ulAddr, res, Ret, jk, k, i;          //写 FLASH 时用到的变量
    int timeCnt=0, random_no=0;

    p=0xfc00;
    BS_Flag=*p;
    BSR_DeleteSDGroup(0);                                // 初始化存储器 RAM
    if (BS_Flag == 0xFFFF)
    {
        PlayRespond(RSP_INTR);
        PlayRespond(RSP_INTR);                            //播放开始训练的提示音

        //..... 训练名称.....
        while(TrainWord(NAME_ID, 1) != 0) ;
        //..... 训练第一条命令.....
        while(TrainWord(COMMAND_ONE_ID, 2) != 0) ;
        //..... 训练第二条命令.....
        while(TrainWord(COMMAND_TWO_ID, 3) != 0) ;
    }
}
```

```
BS_Flag=0xaaaa;
F_FlashWrite1Word(0xfc00, 0xaaaa);
F_FlashErase(0xf700);
F_FlashErase(0xf800);
F_FlashErase(0xf900);

ulAddr=0xf700;//*****
for(commandID=0x100;commandID<0x103;commandID++)
{
    do
    {
        g_Ret=BSR_ExportSDWord(commandID);
    }while(g_Ret!=0);

    for(i=0;i<100;i++)
    {
        F_FlashWrite1Word(ulAddr, BSR_SDModel[i]);
        ulAddr+=1;
    }
}
}
if(BS_Flag==0xaaaa)
{
    BSR_DeleteSDGroup(0);
    p=0xf700;

    for(jk=0;jk<3;jk++)
    {
        for(k=0;k<100;k++)
        {
            Ret=*p;
            BSR_SDModel[k]=Ret;
            p+=1;
        }
        g_Ret=BSR_ImportSDWord();
        while(g_Ret!=0)
            g_Ret=BSR_ImportSDWord();
    }
}

//..... 开始识别命令.....
BSR_InitRecognizer(BSR_MIC); //辨识器初始化
BSR_EnableCPUIndicator(); //启动实时监控
```

[点击这里](#) 获赠价值数万元的 200 多套原创技术方

```
//PlayRespond (RSP_STAR); // 播放开始辨识的提示音

while(1)
{
    random_no++;
    if(random_no >= 3) random_no = 0;
    res = BSR_GetResult();

    if(res > 0) //识别出命令
    {
        if(gActivated)
        {
            timeCnt = 0;
            switch(res)
            {
                case NAME_ID:
                    PlayRespond(gTriggerRespond[random_no]);
                    break;
                case COMMAND_ONE_ID:
                    PlayFlag = 1;
                    PlayRespond2(RSP_GUNSHOT);
                    PlayFlag = 0;
                    gActivated = 0;
                    break;
                case COMMAND_TWO_ID:
                    PlayRespond(gComm2Respond[random_no]);
                    gActivated = 0;
            }
        }
        else
        {
            if(res == NAME_ID)
            {
                PlayRespond(gTriggerRespond[random_no]);
                gActivated = 1;
                timeCnt = 0;
            }
        }
    }
    else if (gActivated)
    {
        if (++timeCnt > 450) //超出定时
```

```
{
    PlayRespond(RSP_NOVOICE); //在设定时间内没有检测出声音

    gActivated = 0;
    timeCnt = 0;
}
}
}
```

## 与朋友分享

读完这本书，你一定会大有收获。这不但是一个价值连城的礼物，还是关爱的礼物，因为你之所以得到它，也是因为朋友的分享。

让我们将这种关爱和分享传递下去，现在请通过 QQ，电子邮箱，MSN，聊天室，博客，论坛将本资料分享给你的朋友、同事和同学、、、、、、