



研究生公选课 机器人设计与制作(I)

电子科技大学 机械电子工程学院 秦东兴、骆德渊

2010-05-01

Contents



Introduction of Robot

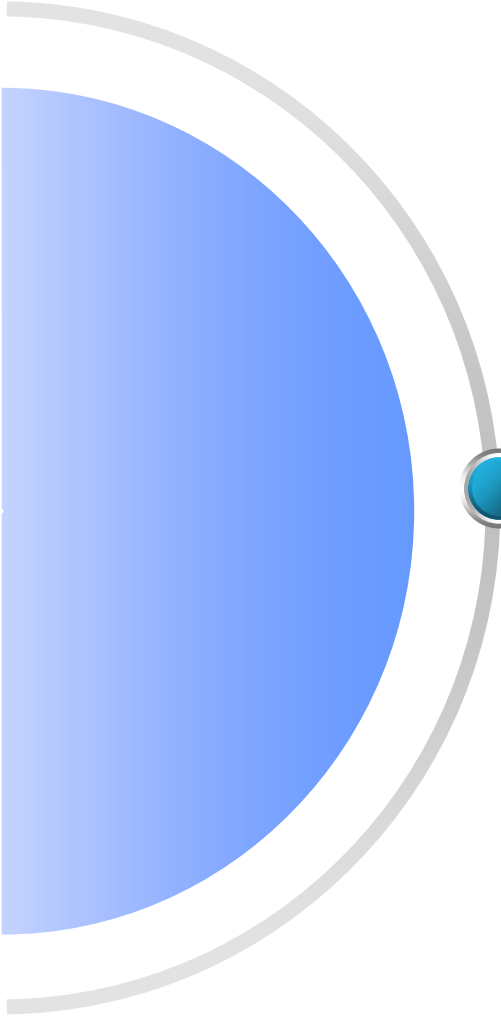
Locomotion Mechanism

Design & Fabrication

Assembly & Programming

Improvement & Optimization

Contents



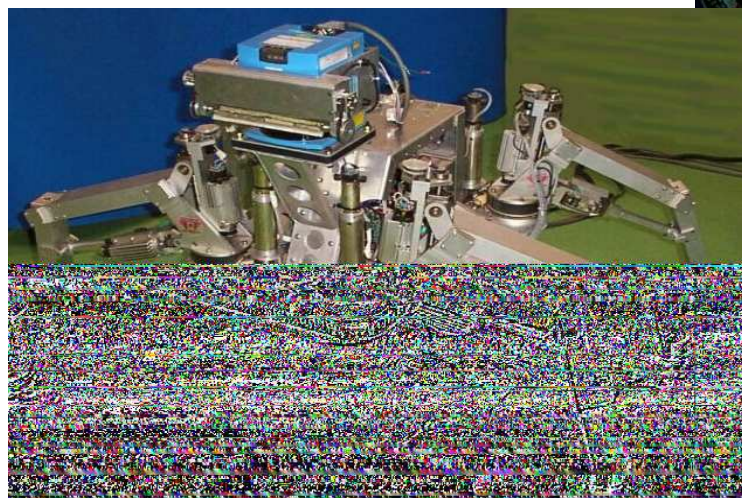
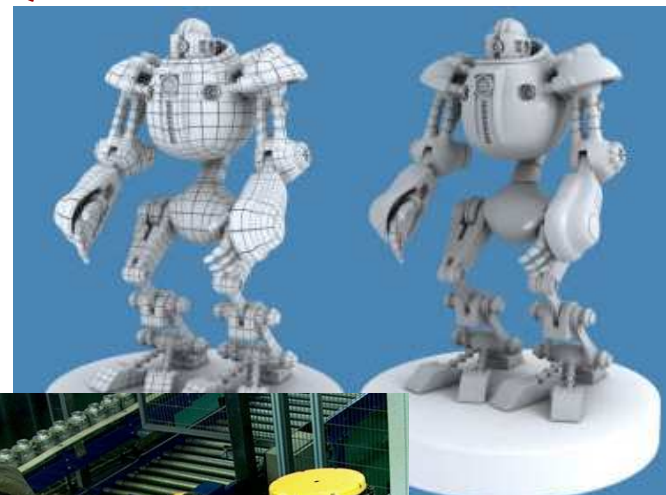
**Introduction
of Robot**

机器人的特点

机器人：“机器的构造” + “人的行为模式”

行为特点：感知---决策----行为

构造特点：感觉器官----大脑---肢体



机器人的特点

机器人的构造

感觉器官-----传感器

大脑-----中央处理器

神经-----电信号线

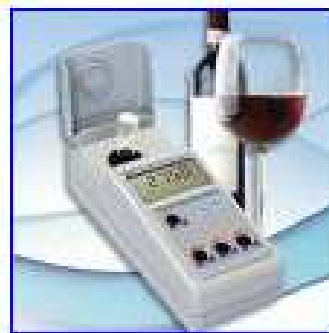
肌肉和肢体-----电机及其它执行器

人类靠“生物能”感觉和执行的，但机器人依靠电源

机器人的特点

机器人的构造

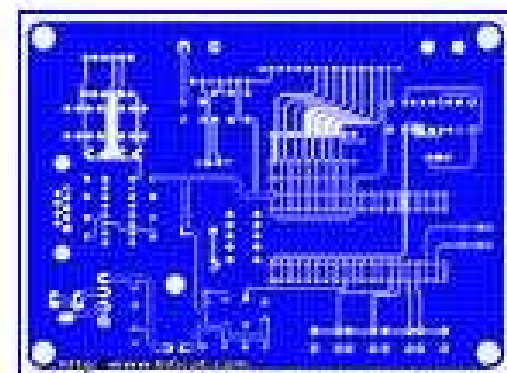
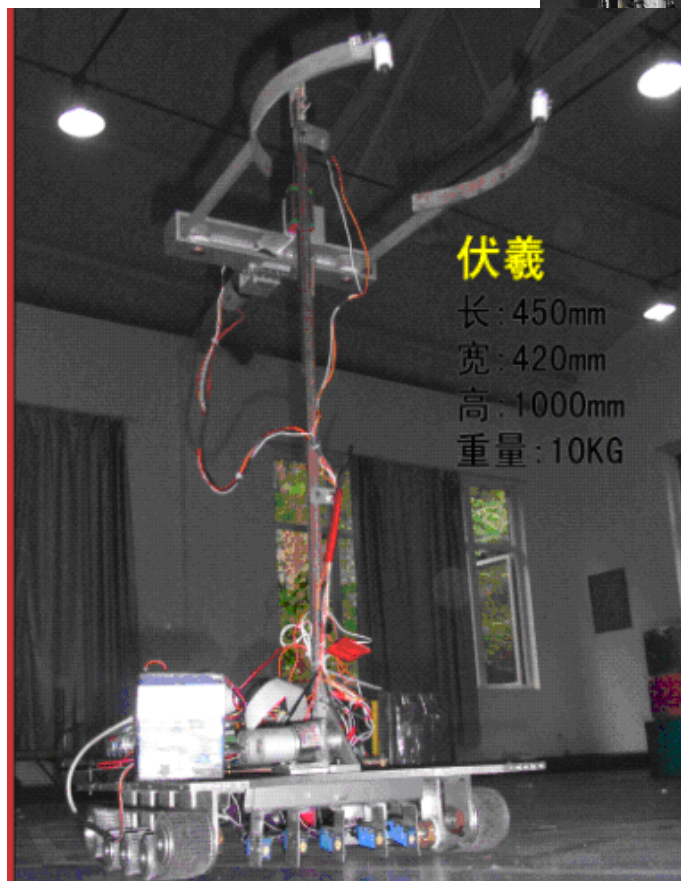
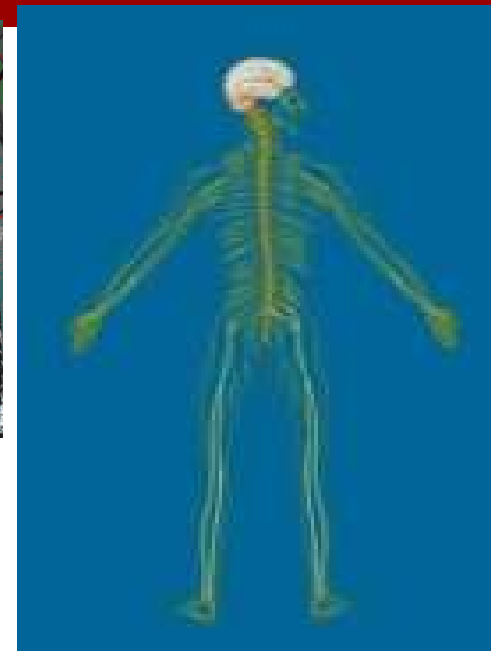
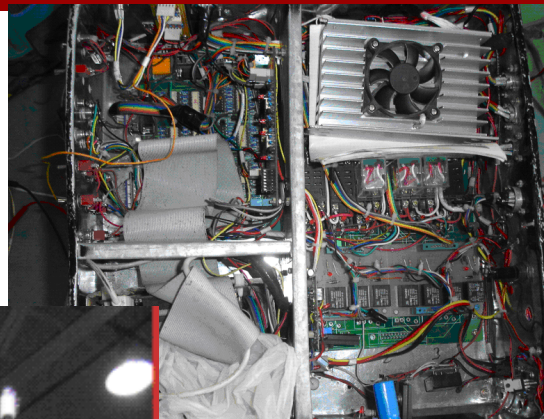
感觉器官=传感器



机器人的特点

机器人的构造

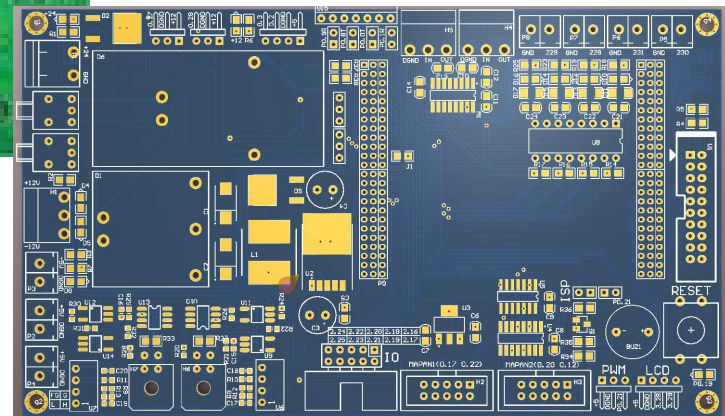
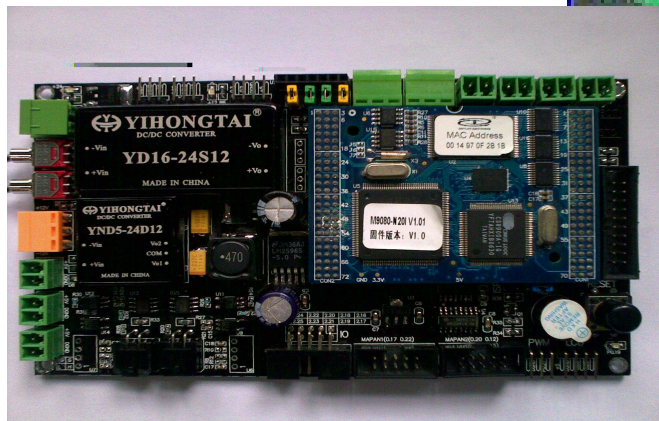
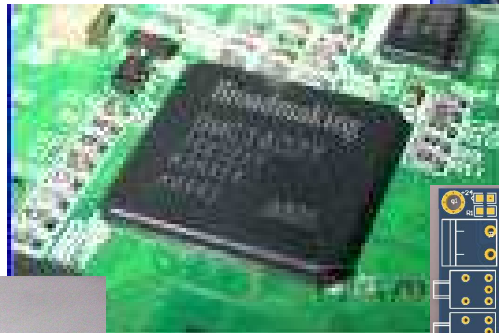
神经 = 电信号传输线



机器人的特点

机器人的构造

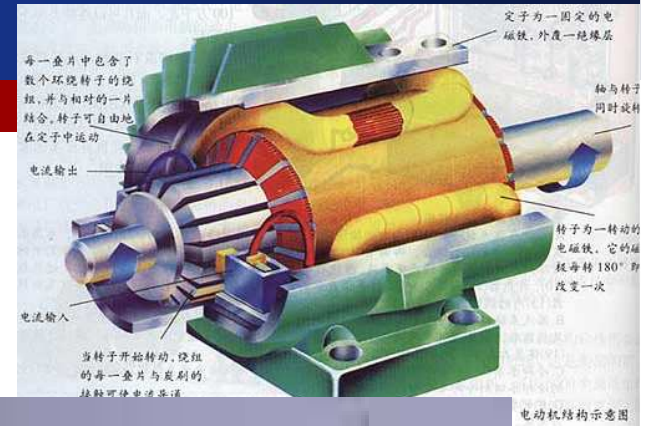
大脑 = 中央处理器



机器人的特点

机器人的构造

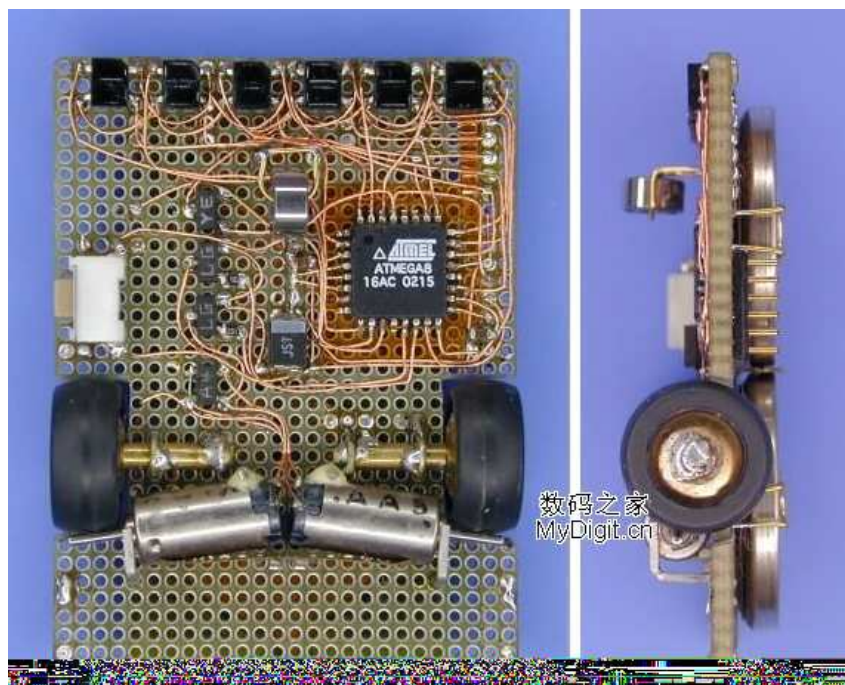
肌肉和四肢 = 执行器



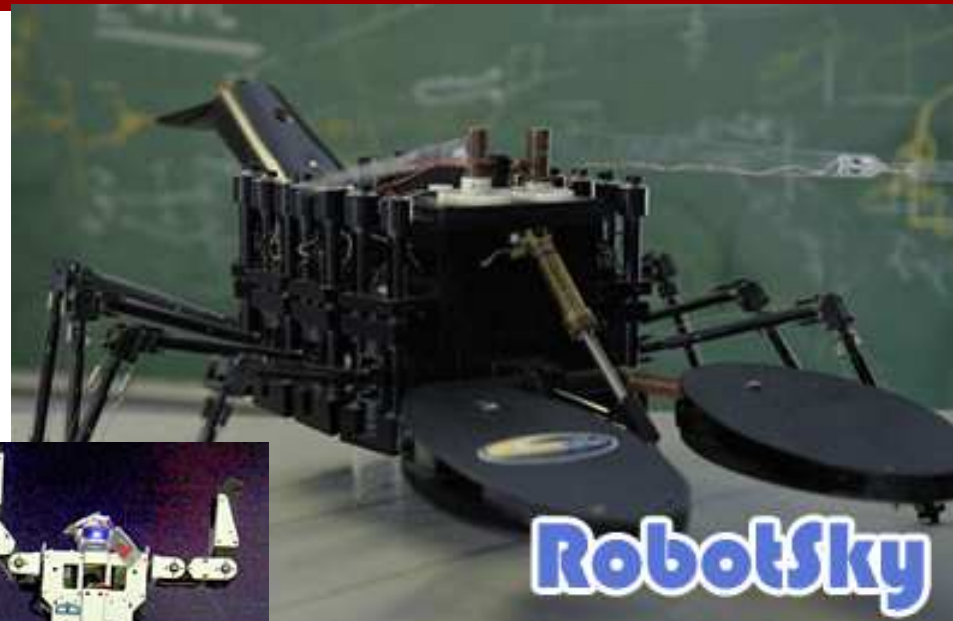
机器人的特点

机器人的构造

能量从哪里来？

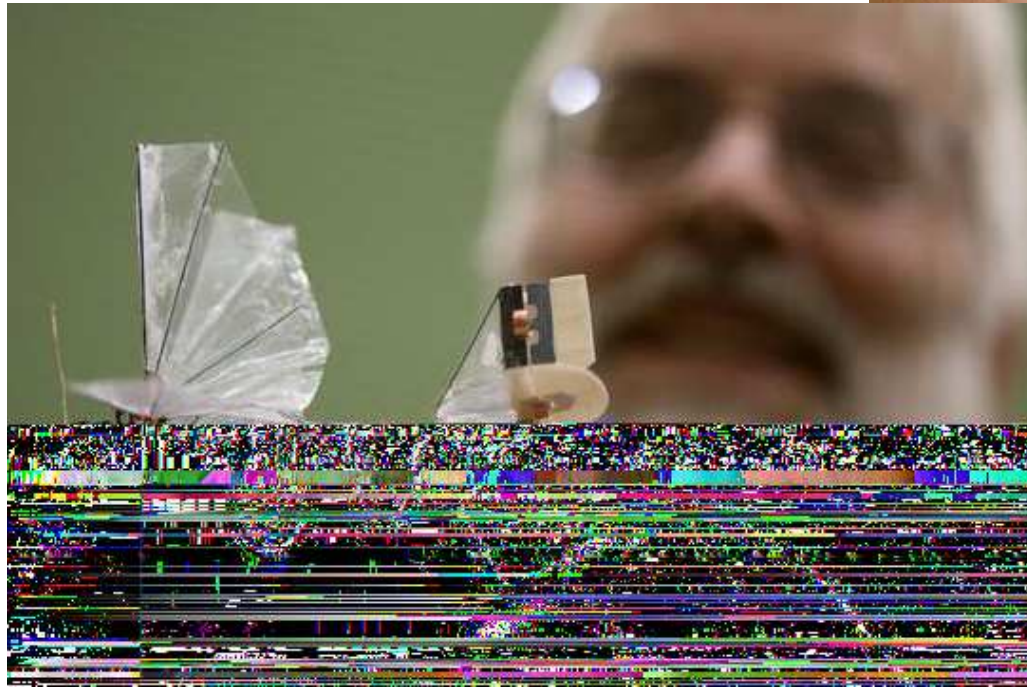


机器人的特点



机器人的特点

机器昆虫：也是机器人



机器人的特点

在月球上行走的机器人----月球车



NASA的勇气号火星车



机器人分类

广义上说，机器人技术是一种综合性的机电一体化技术，包括传动机构、伺服控制、数据处理、人机对话以及与机器人工作性质对应的执行与控制功能。

机器人是可以在现场替代人类执行规定操作的智能机器。目前可归类为：

- 1) 工业机器人；
- 2) 特种作业机器人，如水下作业机器人，辐射环境作业机器人等；
- 3) 竞技机器人；
- 4) 娱乐机器人；
- 5) 家务机器人可能在不久的将来出现。

机器人分类

工业机器人



特种作业机器人



机器人分类

娱乐机器人



家务机器人可能在
不久的将来大量出现



竞技机器人

竞技机器人设计与制作

- 创新是科技发展的灵魂，是高素质人才最本质的能力。机器人教育是培养综合创新能力的重要途径之一。
- 机器人涉及机械与机构学、传感与检测、控制理论、通讯和应用微电子学等诸多领域的前沿技术。竞技项目是集高技术、娱乐和比赛于一体的对抗性活动，比赛种类多，项目丰富多彩，大学生参与热情很高。
- 通过开展机器人设计与制作课程，培养研究生的分析能力、动手能力、思维能力和团队合作能力，提高创新能力，激发创新意识。

竞技机器人

竞技机器人需要自己设计制作与编程开发

- 需要研究生综合应用微电子学、自动控制、计算机、机械工程、传感器、嵌入式系统、软件编程等单元技术。
- 整合广博的知识，进行系统地分析与综合，进行设计与创新
- 项目管理与自主时间管理的能力
- 有些竞技项目设计与制作时间长达12个月

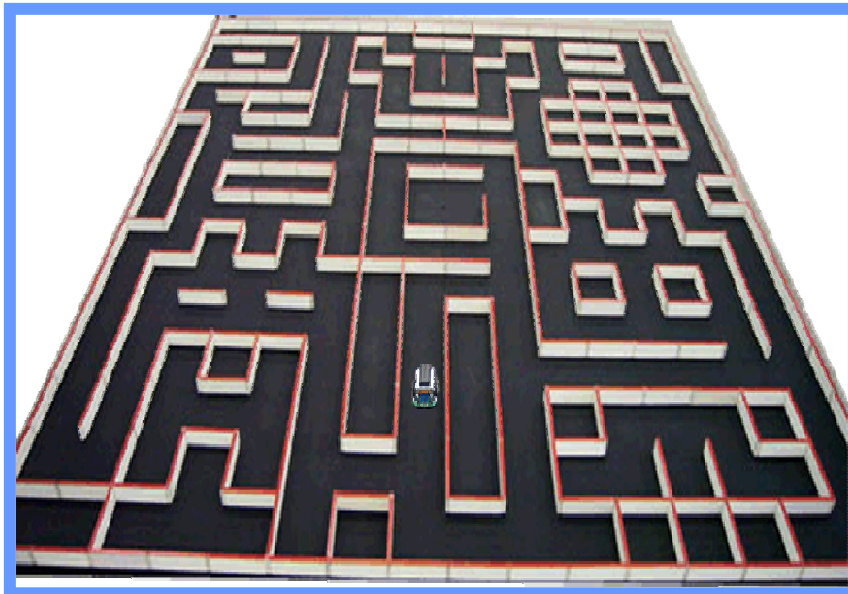
Contents



**Mobile Robots and
Locomotion Mechanism**

Locomotion of the Robot

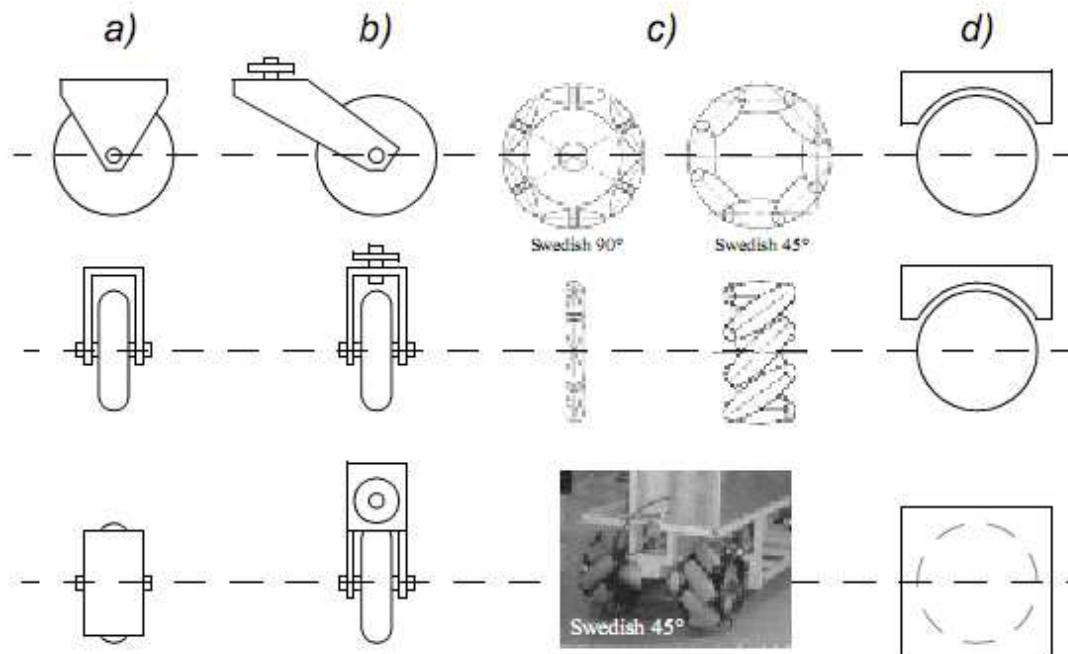
- The first challenge of mobile robots is locomotion. Robots can move unsupervised through real world environments to fulfill its task.
- How should a mobile robot move? And what is it about a particular locomotion mechanism that make it superior to alternative ones when moving on the specific ground?



Locomotion of the Robot

Wheeled mobile robots:

- By far the most popular locomotion mechanism in man-made vehicles in general. It can achieve very good efficiencies when moving on the structural ground.
- With a relatively simple mechanical implementation, and balance is usually not a research problem.



Locomotion of the Robot


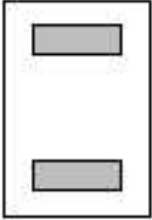
Wheeled mobile robots:

- Three wheels are sufficient to guarantee stable balance, two wheels robots can also be stable. When more than three wheels are used, a suspension system is required to allow all wheels to maintain ground contact when the robot encounters uneven terrain.



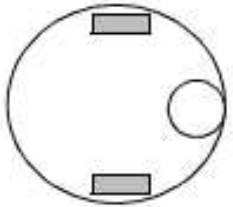
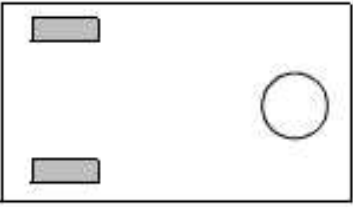
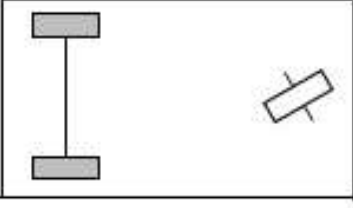
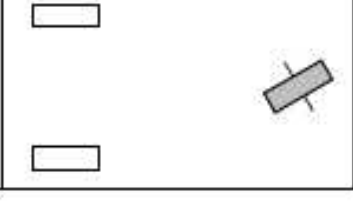
Locomotion of the Robot

Two-Wheeled mobile robots:

	One steering wheel in the front, one traction wheel in the rear	Bicycle, motorcycle
	Two-wheel differential drive with the center of mass (COM) below the axle	Cye personal robot

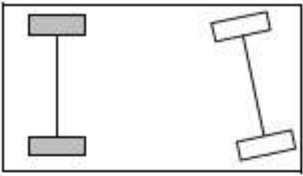
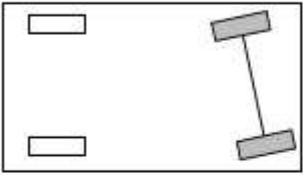
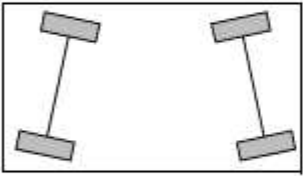
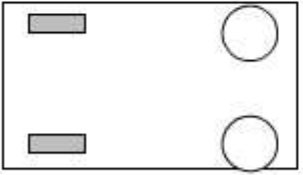
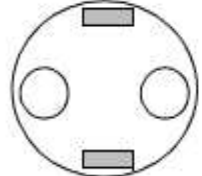
Locomotion of the Robot

Three-Wheeled mobile robots:

	<p>Two-wheel centered differential drive with a third point of contact</p>	<p>Nomad Scout, smartRob EPFL</p>
	<p>Two independently driven wheels in the rear/front, 1 unpowered omnidirectional wheel in the front/rear</p>	<p>Many indoor robots, including the EPFL robots Pygmalion and Alice</p>
	<p>Two connected traction wheels (differential) in rear, 1 steered free wheel in front</p>	<p>Piaggio minitrucks</p>
	<p>Two free wheels in rear, 1 steered traction wheel in front</p>	<p>Neptune (Carnegie Mellon University), Hero-1</p>

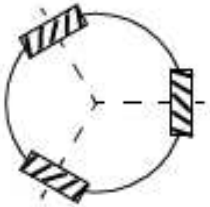



Locomotion of the Robot

Four-Wheeled mobile robots:

	<p>Two motorized wheels in the rear, 2 steered wheels in the front; steering has to be different for the 2 wheels to avoid slipping/skidding.</p>	<p>Car with rear-wheel drive</p>
	<p>Two motorized and steered wheels in the front, 2 free wheels in the rear; steering has to be different for the 2 wheels to avoid slipping/skidding.</p>	<p>Car with front-wheel drive</p>
	<p>Four steered and motorized wheels</p>	<p>Four-wheel drive, four-wheel steering Hyperion (CMU)</p>
	<p>Two traction wheels (differential) in rear/front, 2 omnidirectional wheels in the front/rear</p>	<p>Charlie (DMT-EPFL)</p>
	<p>Two-wheel differential drive with 2 additional points of contact</p>	<p>EPFL Khepera, Hyperbot Chip</p>

Locomotion of the Robot

Swedish-Wheeled mobile robots:

	<p>Three motorized Swedish or spherical wheels arranged in a triangle; omnidirectional movement is possible</p>	<p>Stanford wheel Tribolo EPFL, Palm Pilot Robot Kit (CMU)</p>
	<p>Four omnidirectional wheels</p>	<p>Camegie Mellon Uranus</p>
	<p>Four omnidirectional wheels</p>	<p>UESTC Robot team</p>
	<p>Four omnidirectional wheels</p>	<p>UESTC Robot team</p>

Locomotion of the Robot

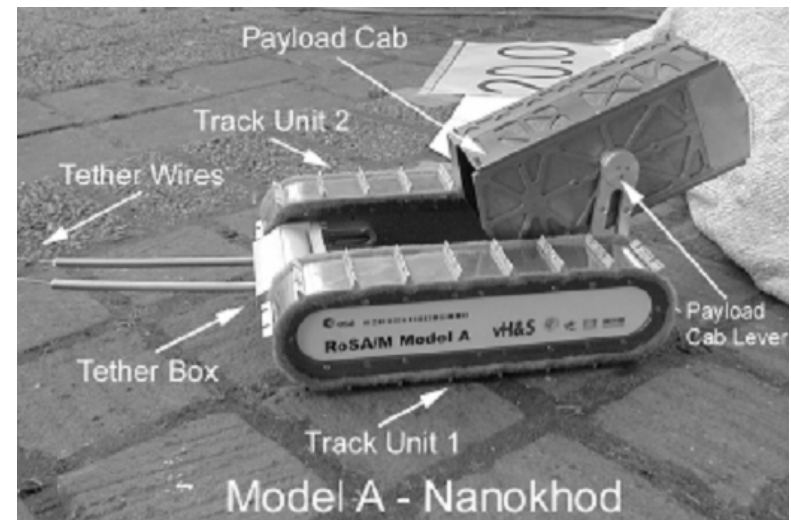
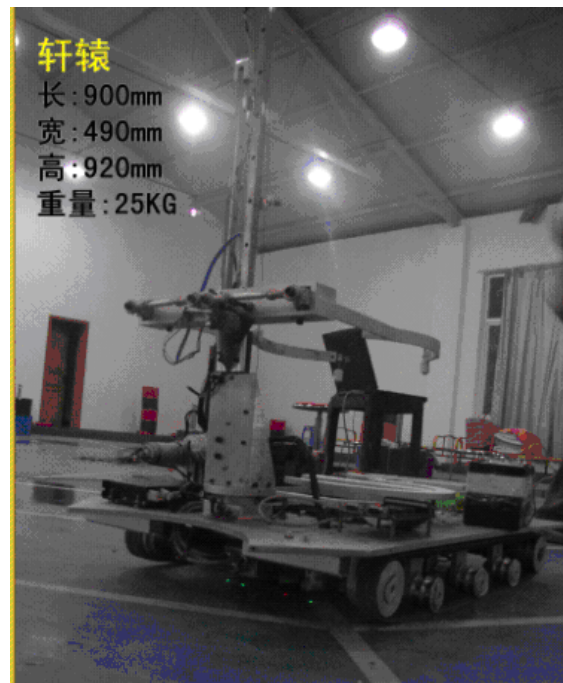
Tracked locomotion robots:

- Robot can make use of tread have much larger ground contact patches, the maneuverability can be improved significantly in loose terrain compared to conventional wheeled designs.
- Due to large ground contact patch, changing the orientation of the robots requires a skidding turn, wherein a large portion of the track must slide against the terrain.
- The exact center of the rotation of the robot is hard to predict and the exact change in position and in orientation is also subject to variations depending on the ground friction.

Locomotion of the Robot

Tracked locomotion robots:

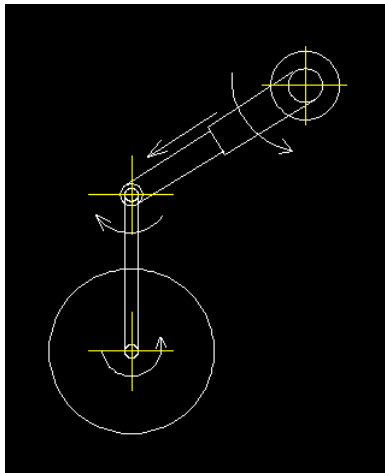
- The microrover Nanokhod: developed for European Space Agency
- Designed in 2007 by UESTC robot team



Locomotion of the Robot

Walking wheels robots:

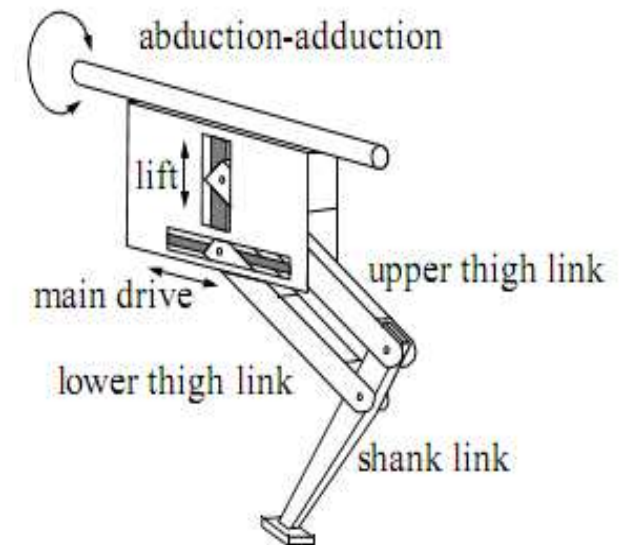
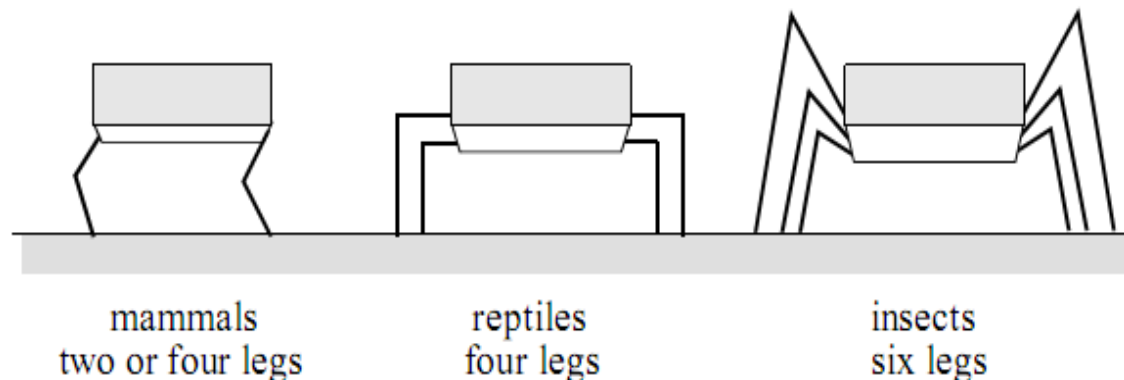
- Offer the best maneuverability in rough terrain, but inefficient on flat ground and need sophisticated control. Hybrid solution, combining the adaptability of legs with the efficiency of the wheels, offer an interesting compromise.
- Solution that passively adapt to the terrain are of particular interest for field and space robotics.



Locomotion of the Robot

Legs robots:

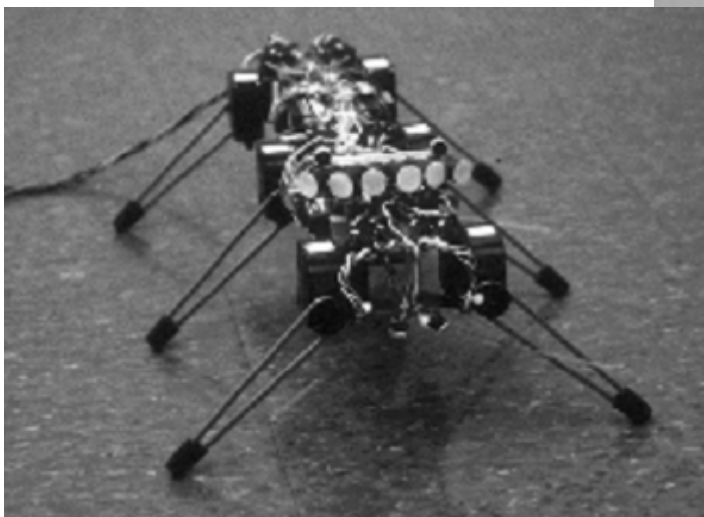
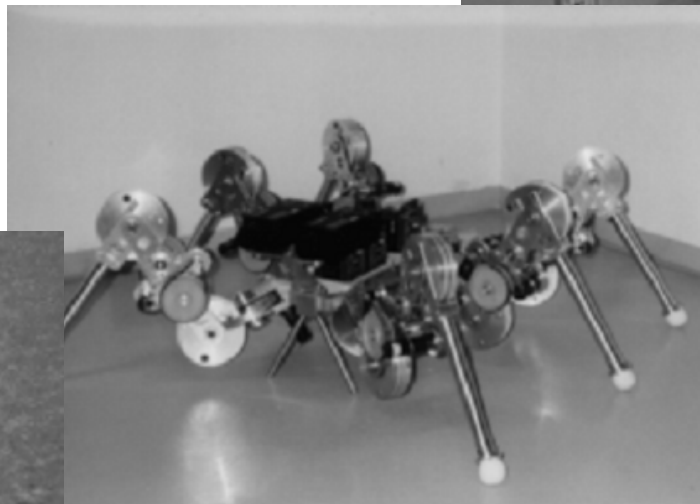
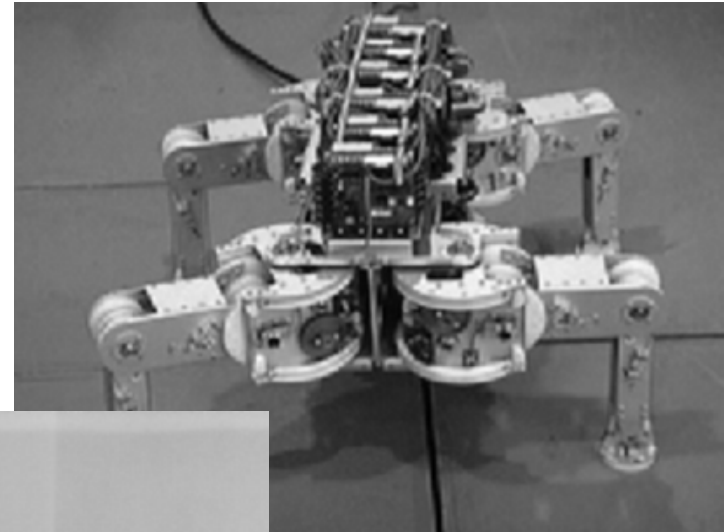
- Numbers of legs: Insects and spiders are immediately able to walk when born. Humans, with 2 legs, can not stand in one place with static stability. Infants requires months to study stand and walk.
- Stable balance.



Locomotion of the Robot

Legs robots:

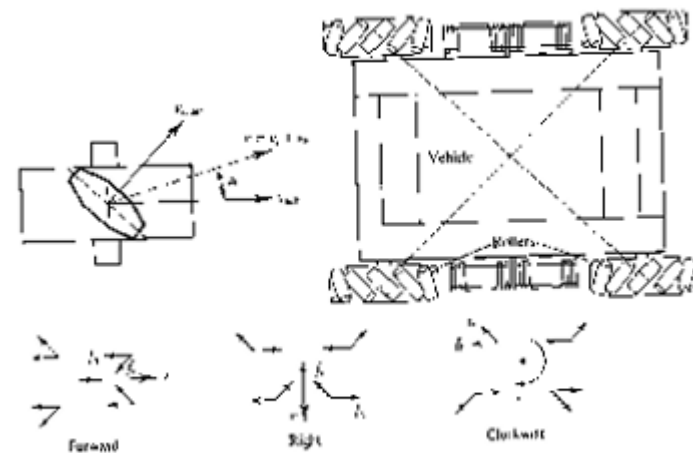
- Tokyo institute of tech: four legs
- University of Karlsruhe, Germany: six legs
- MIT:genghis



Locomotion of the Robot

Controllability:

- There is an inverse correlation between controllability and maneuverability. For example, the four wheel omnidrive such as Uranus Robot, which has four swedish wheels, the problem is even harder because all four wheels must be driven at exactly the same speed to travel in a perfectly straight line.



Locomotion of the Robot

Controllability:

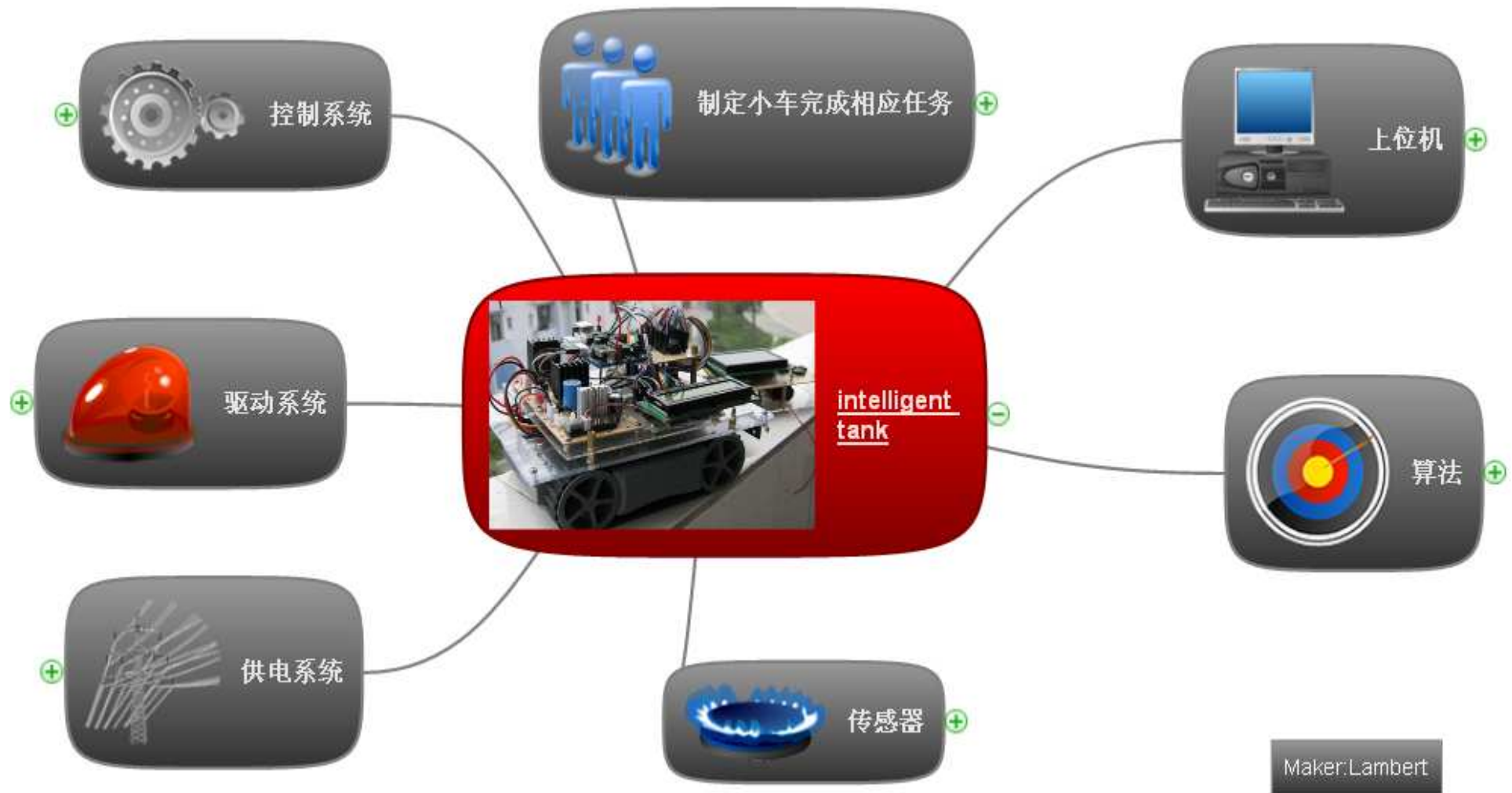
- In summary, there is no “ideal” drive configuration simultaneously maximizes stability, maneuverability and controllability.
- Each robot places unique constraints on the design problem, the designer’s task is to choose the most appropriate drive configuration possible from among this space of compromises.

Contents



Design & Fabrication

Robot Design



Robot Design

作为竞技项目的参与者，我们追求胜利。与其它类型的机器人相比，竞技机器人有些独特的竞技环境：

- 1) 设计的目标是在对抗中获胜，有针对性的战术策略设计；
- 2) 对抗作业的范围大：包括电磁对抗、结构对抗（可靠性、强度）、策略的对抗；
- 3) 干扰难以事先预知，要求机器人有智能反应能力；
- 4) 对机器人的结构尺寸与重量有限制；
- 5) 对动作的完成时间有限制，追求速度。

Robot Design

竞技机器人上述特点，要求我们在设计机器人的时候，要重点考虑以下几个机器人的实现特点：

- 1) 深刻理解竞技主题，紧扣对抗特点设计竞技动作；
- 2) 策略实现应考虑并善于利用竞技场地环境；
- 3) 深入研究游戏规则与得分规则；
- 4) 追求出其不意、出奇制胜的对抗策略，敢于采用新技术新方法；
- 5) 在对抗训练中的不断优化改进设计；
- 6) 设计中应注重干扰对抗；
- 7) 设计要求：体现“快、准、稳”。

Robot Design

与其它类型机器人一样，竞技机器人是典型的机电一体化产品。其技术构成包括：

- 1) 执行单元：含得分动作执行机构、传动机构与动作元件；
- 2) 控制单元：电路 + CPU + controlled I/O
- 3) 控制软件：程序 + ? Embed OS
- 4) 电源单元；
- 5) 干扰单元；
- 6) 通信单元。

Robot Design

2006 第一届飞思卡尔大学生智能车竞赛主题——弯曲跑道赛车。

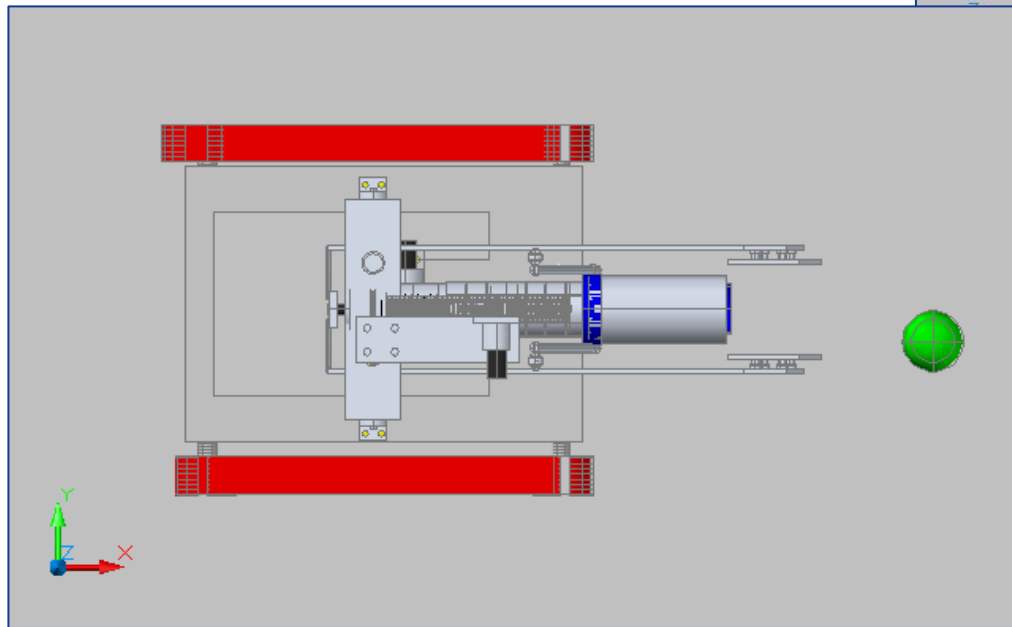
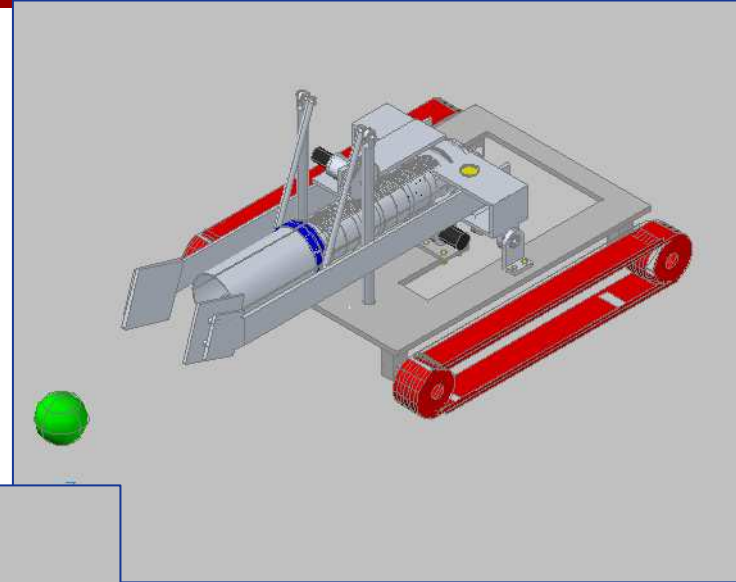
- 1) 采用统一车模，底盘结构、轮距、轮径及轮胎相同；
- 2) 禁止改动驱动电机与传动比，伺服电机数量不超过3个；
- 3) 禁止改动舵机与滚动轴承；
- 4) 统一提供电池，禁止采用DC-DC升压电路；
- 5) 限定采用飞思卡尔MC9S12DG128单片机作为唯一控制处理器，核心模块采用DG128MCU，传感器数量不得超过16个；
- 6) 全部电容容量和不得大于2000uF，最高充电电压不得大于25V；
- 7) 整个车模尺寸250mm×400mm，高度不限。
- 8) 评分：最快单圈时间 + 技术报告得分 + 赛车冲出赛道次数

可见，传感检测、信号处理与软件算法是竞技重点。

Robot Design

2007挑战杯机器人竞赛主题：

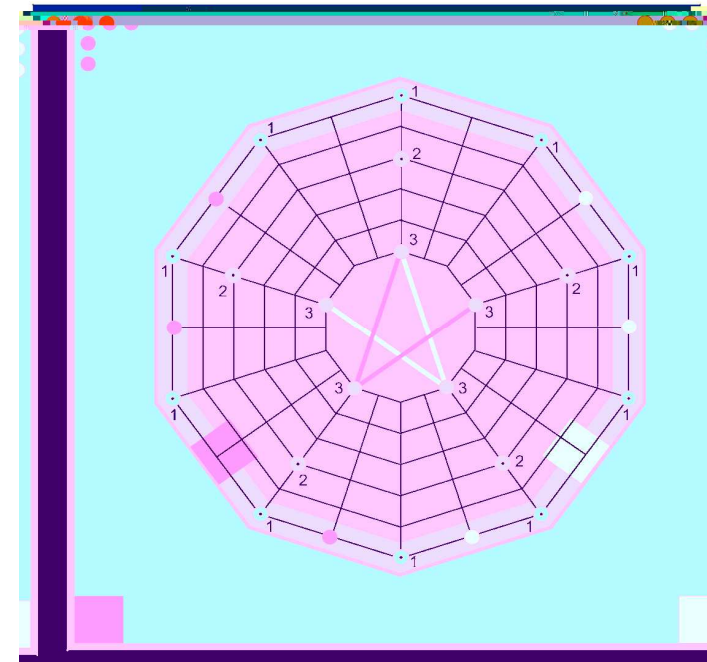
- (1) 码放带标号的方块；
- (2) 走直折赛道；
- (3) 抛带颜色的球得分。



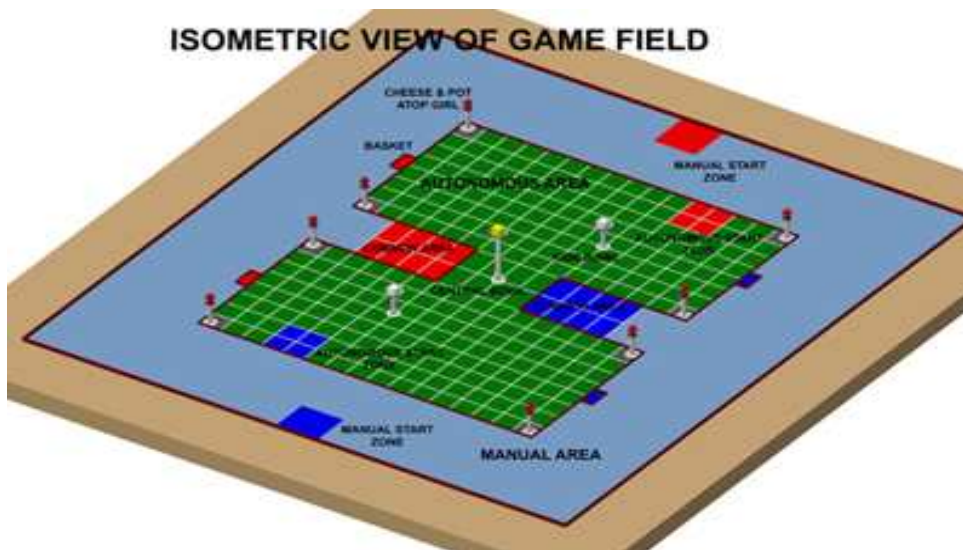
Robot Design

2007 亚太大学生机
器人竞赛主题

2008 亚太大学生机
器人竞赛主题

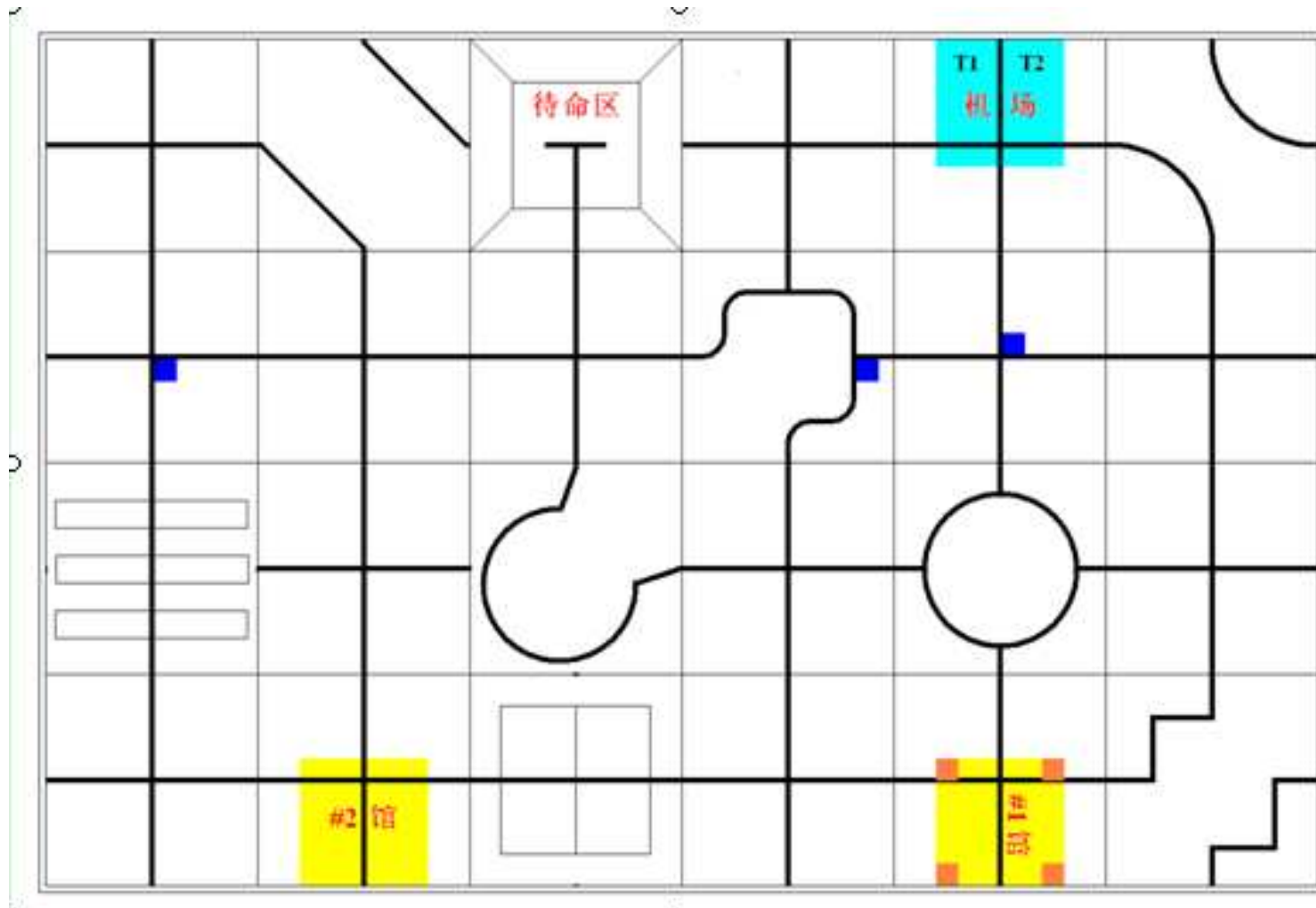


ISOMETRIC VIEW OF GAME FIELD



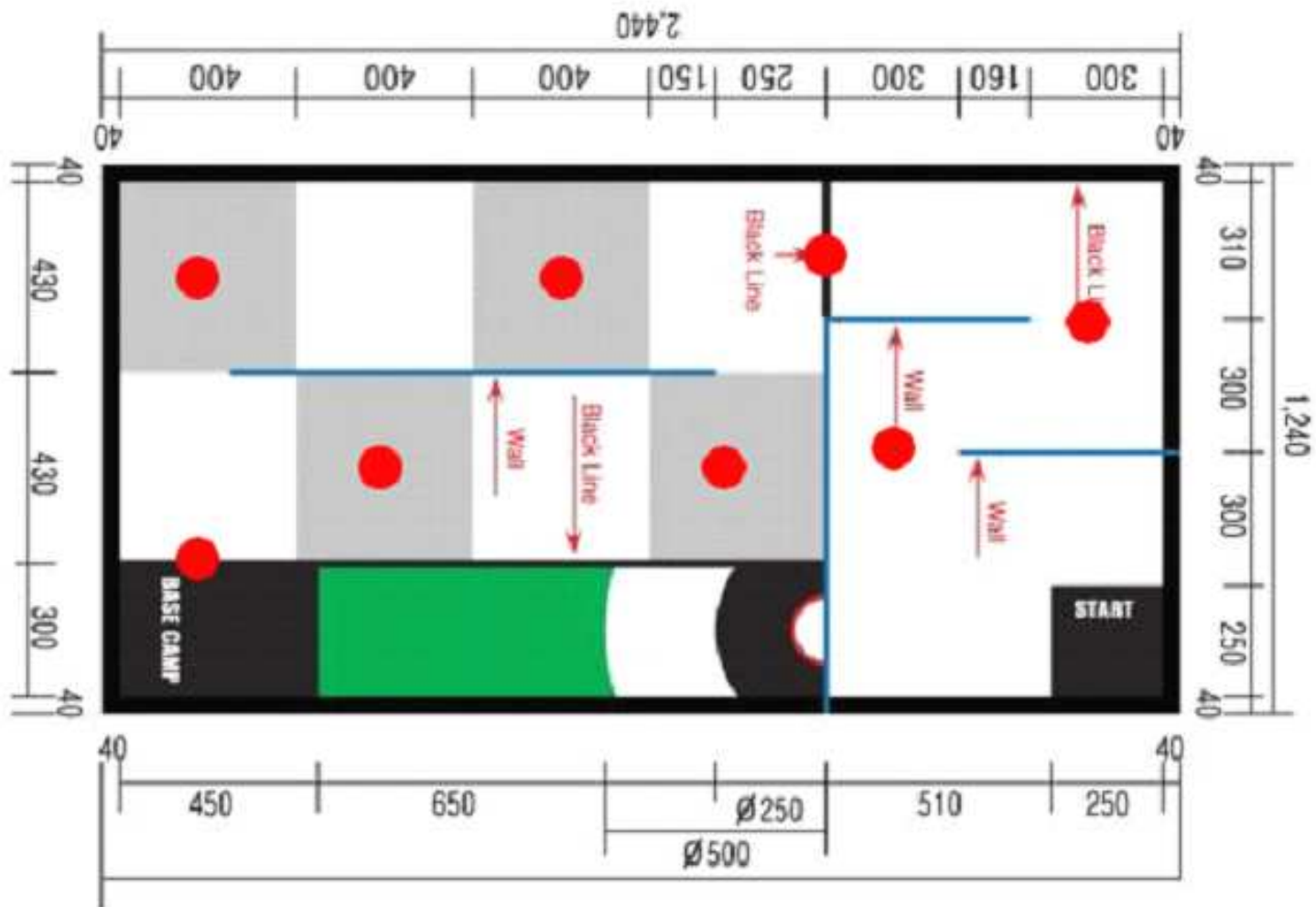
Robot Design

寻线机器人竞赛主题的一个例子



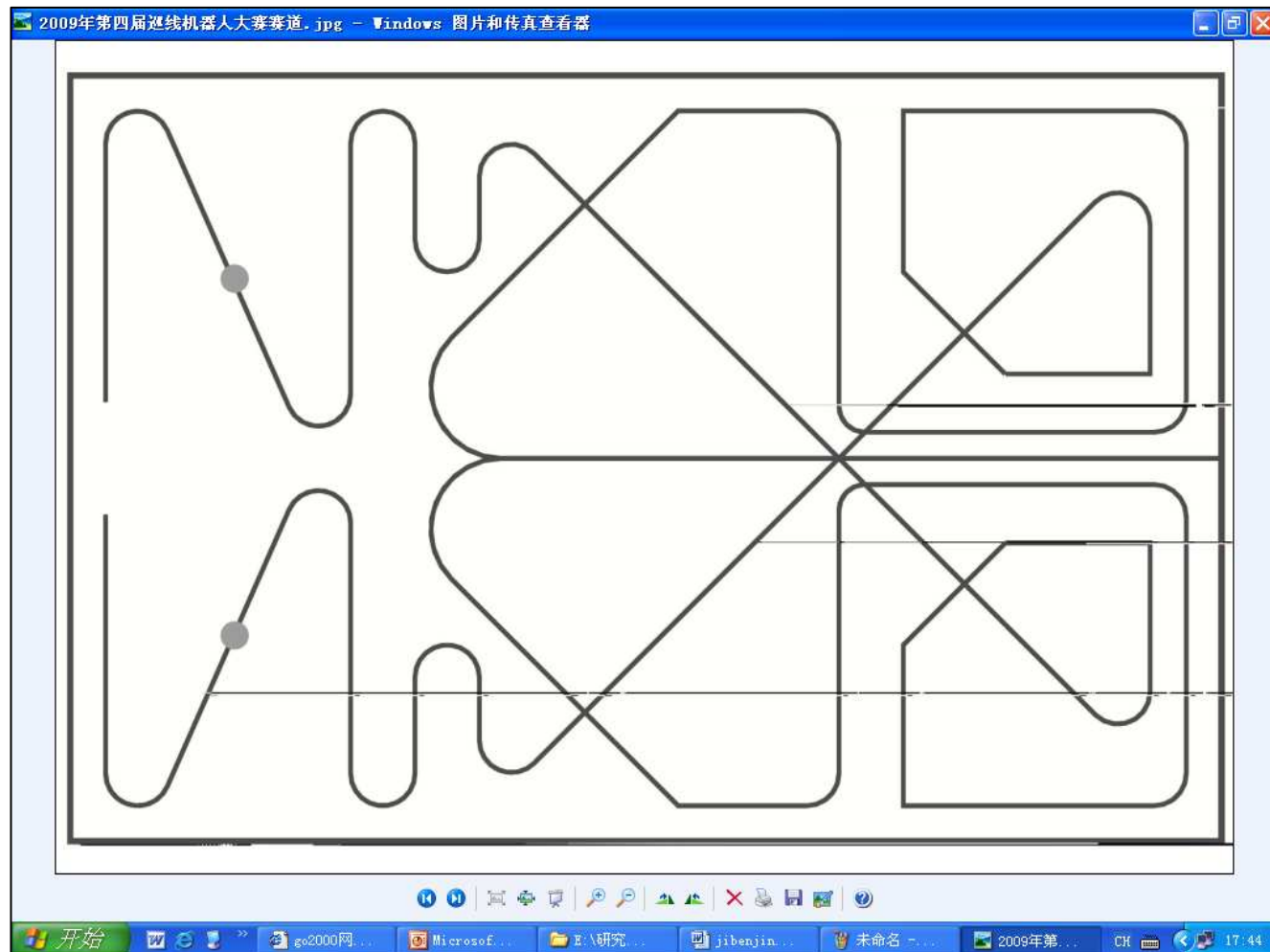
Robot Design

机器人竞赛主题的一个例子



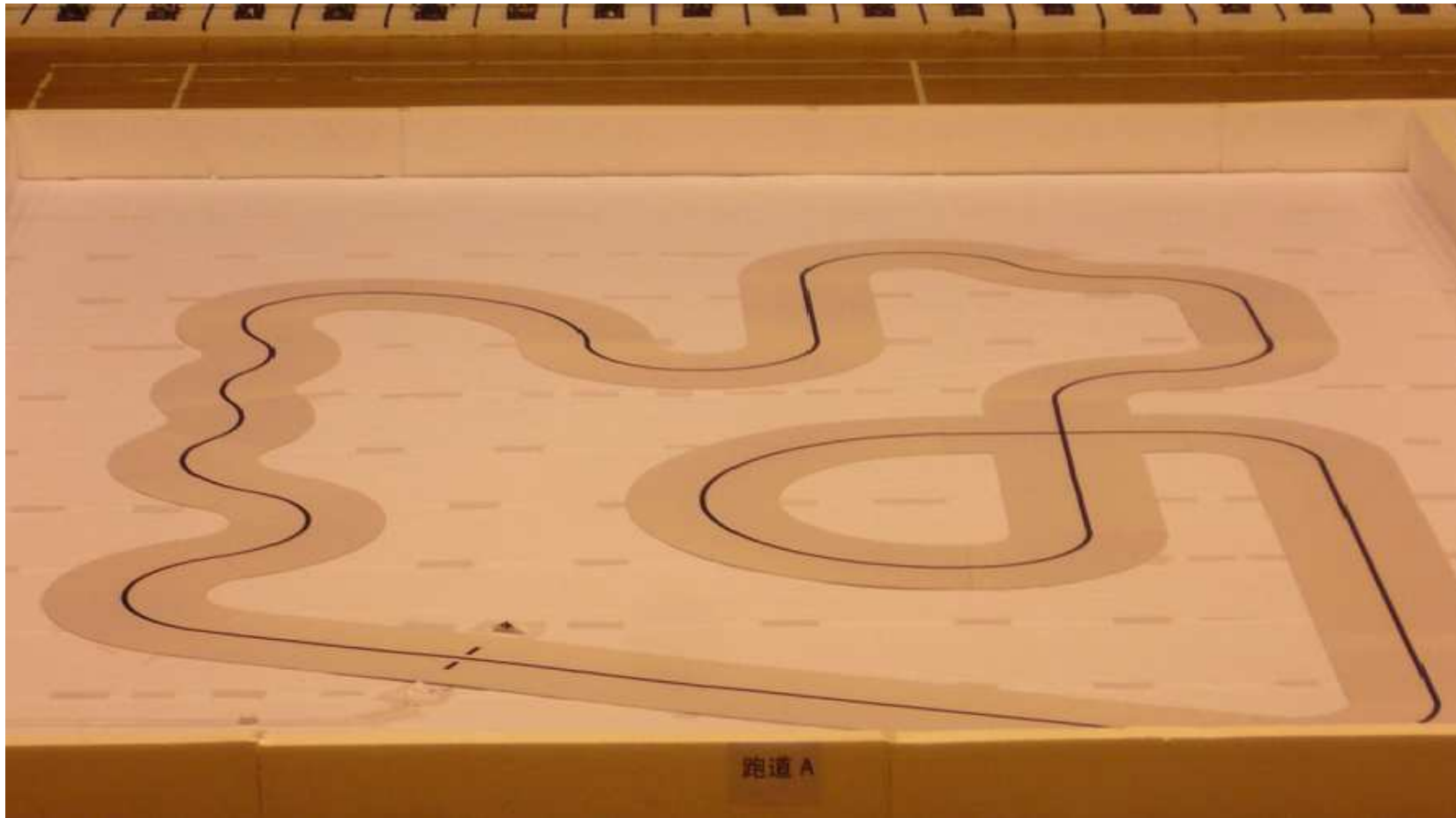
Robot Design

2009寻线机器人竞赛主题



Robot Design

2010寻线机器人竞赛主题



Robot Design

通过主题，我们应深刻理理解规则所包含的竞技要点，从中提取技术设计要素：

- 1) 飞思卡尔智能车：竞技要素集中于传感检测、信号处理与软件算法，需要确保检测灵敏、速度快、技术报告漂亮；
- 2) 电脑鼠走迷宫：稳定、准确、快速的行走能力；正确判断环境的能力，记忆路径的能力；
- 3) 挑战杯：只有尺寸与重量限制，是技术与操控的全面比较，速度很关键；
- 4) 亚太大学生机器人：有尺寸与重量限制，与对手同台竞技，是技术与操控的全面比较，智能性、可靠性和情景反应能力是关键。

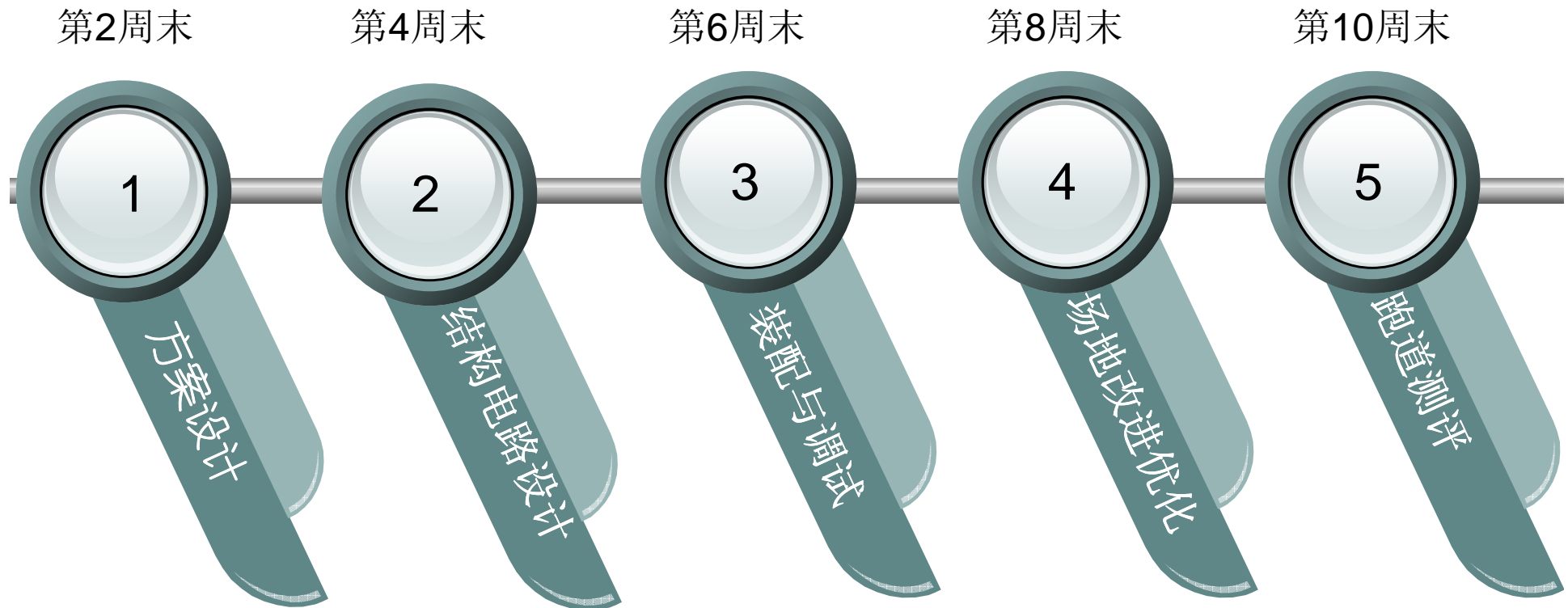
Robot Design

巡线机器人的技术实现包括：



Robot Design

巡线机器人设计制作的时间节点：



Robot Design

方案设计对技术实现与机器人性能影响很大，应注重创新。输出与评估应考虑赛场环境：

- 1) 采用三维模型将结构特点表现出来；
- 2) 采用运动仿真将控制动作 + 运动定位的特点表现出来；

应注意：

- 3) 采用一致的方案设计软件：如PRO/E, AutoCAD
- 4) 描述不必过分详细，说明方案意图与独特竞技优势即可；
- 5) 方案论证应关注可出奇制胜的方案，不以实现难度为理由抛弃方案。

Robot Design

方案设计应考虑规则约束

1) 运动元件:

- 电机（直流、交流、步进等）；
- 差速方法；
- 气动或液压元件（相应能源）

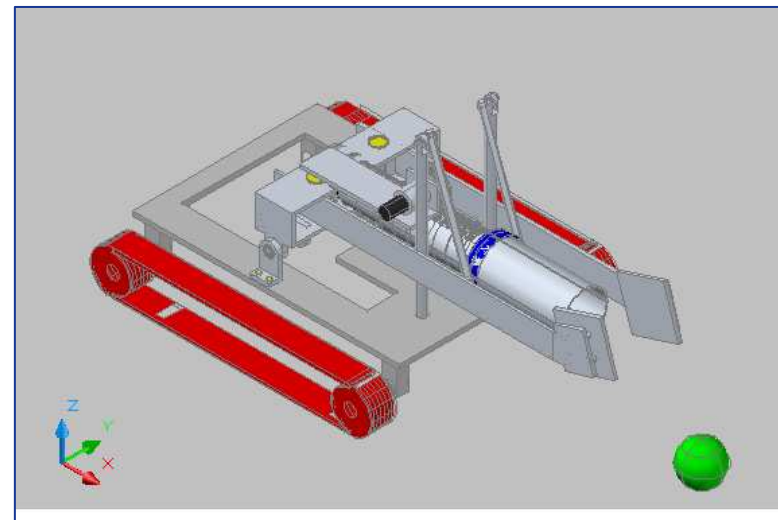
2) 运动转化与传递部件:

- 变速与提高转矩：齿轮机构、杠杆机构；
- 旋转变直线运动：丝杆导轨模组、拉绳 + 滑轨、凸轮 + 顶杆机构、连杆 + 滑块机构、同步带；
- 运动支撑：滚动轴承、滑动轴承、直线轴承；
- 缓冲机构：空气阻尼、液压缓冲器、弹簧、耗能材料；
- 运动定位：舵机、轮式底盘（双驱）、履带式底盘、多足步进爬行式、飞行式、两足步行式、喷气动量式等等

Robot Design

方案设计时，应分解机器人的动作，并通过基本运动实现。

- 1) 大家想象一下采用XYZ直线运动模块来到达空间指定位置与采用XY直线运动模块和X轴旋转模块相同吗？
- 2) 动作综合：就是通过选定的机构的顺序运动或同时运动，完成机器人应完成的所有动作。
- 3) 好的设计没有多余的机构与结构，多余意味这复杂和重量配置的低效。



Robot Design

巡线机器人结构设计：

- 考虑车底板结构强度与刚度。采用（硬铝合金）弯边板材，既可保证强度要求，加工时间短，成本低。注意结构变形较大可能会引起机构卡死。
- 重量分配应合理。重心的位置——不佳时会在高速转弯或撞击中倾倒。
- 要避免结构在冲击时诱发振动，使控制性能变坏。
- 驱动电机如何与轮子连接？底板距离地面多高？四个轮子如何保证同时着地？其他部件如何固定？
- 关键配合设计需要有公差标注：如轴孔配合等，每个需要加工制作的零件都需要图纸，应考虑结构件的加工工艺要简单，工艺性要好。

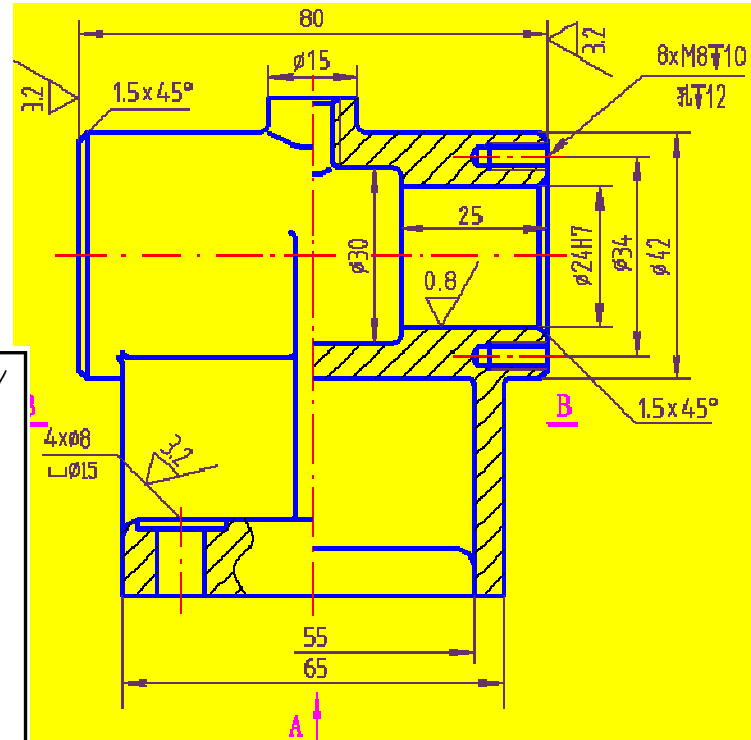
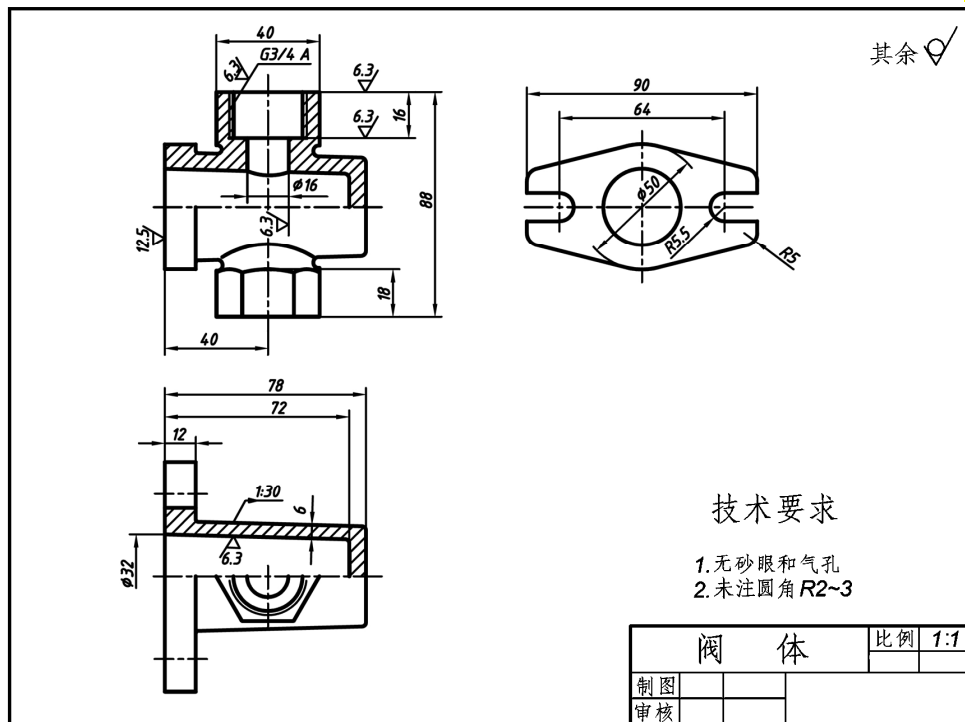
Robot Design

巡线机器人结构设计的基本原则：

- 从总体上建立起结构设计和零部件设计的总体概念，即从寻线机器人的总体要求出发，引出对结构零部件的要求，根据零件的功能和失效形式，拟定设计准则，用一定的设计方法来设计零部件。
- 掌握机械零件材料的选用原则和结构工艺性设计的基本要求，学会采用标准。
- 掌握关于机械零件的载荷与应力的基本概念，尤其要熟练掌握变应力、极限应力、许用应力与安全系数、接触应力的基本概念。

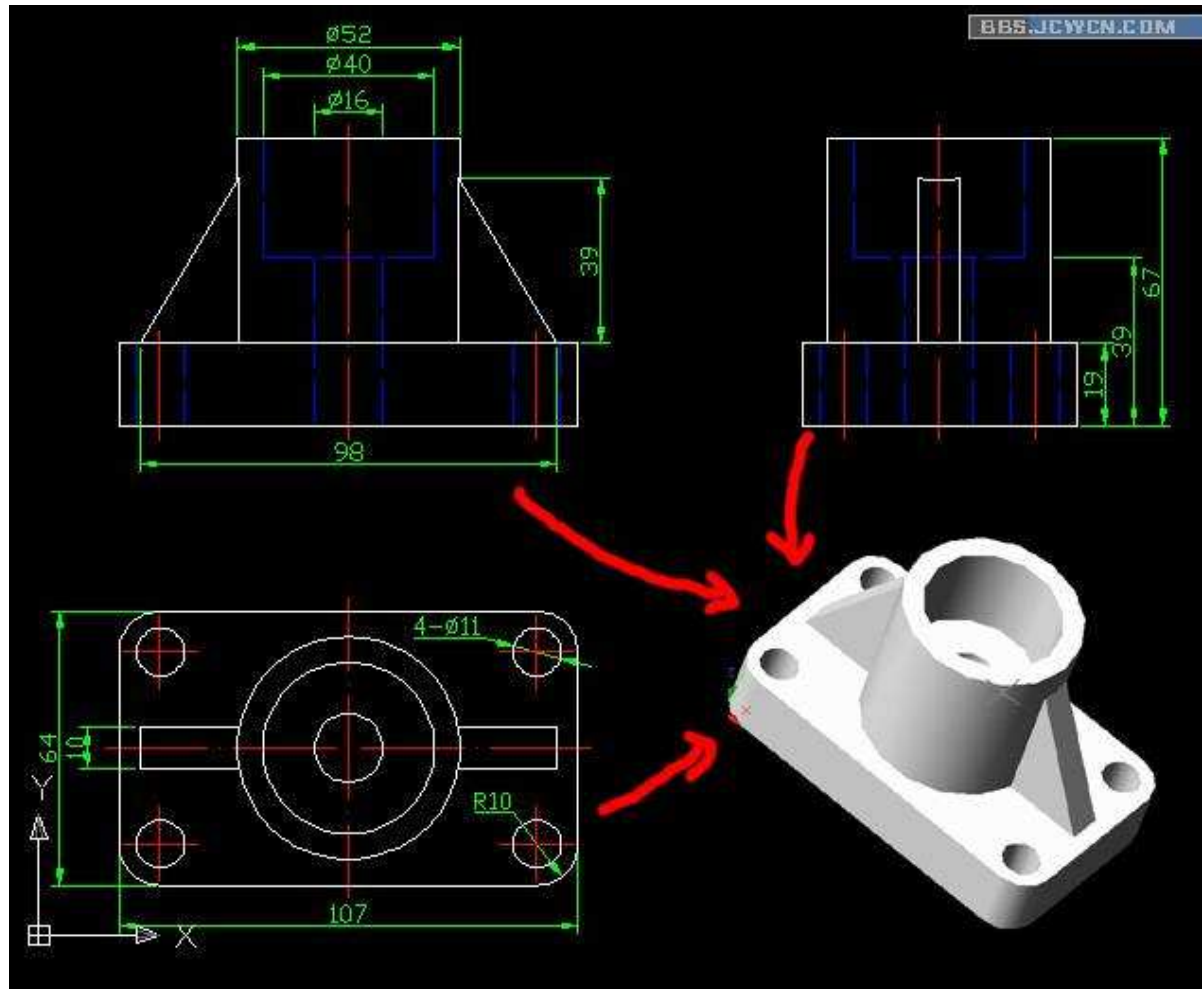
Robot Design

结构设计图 (2D) 实例:



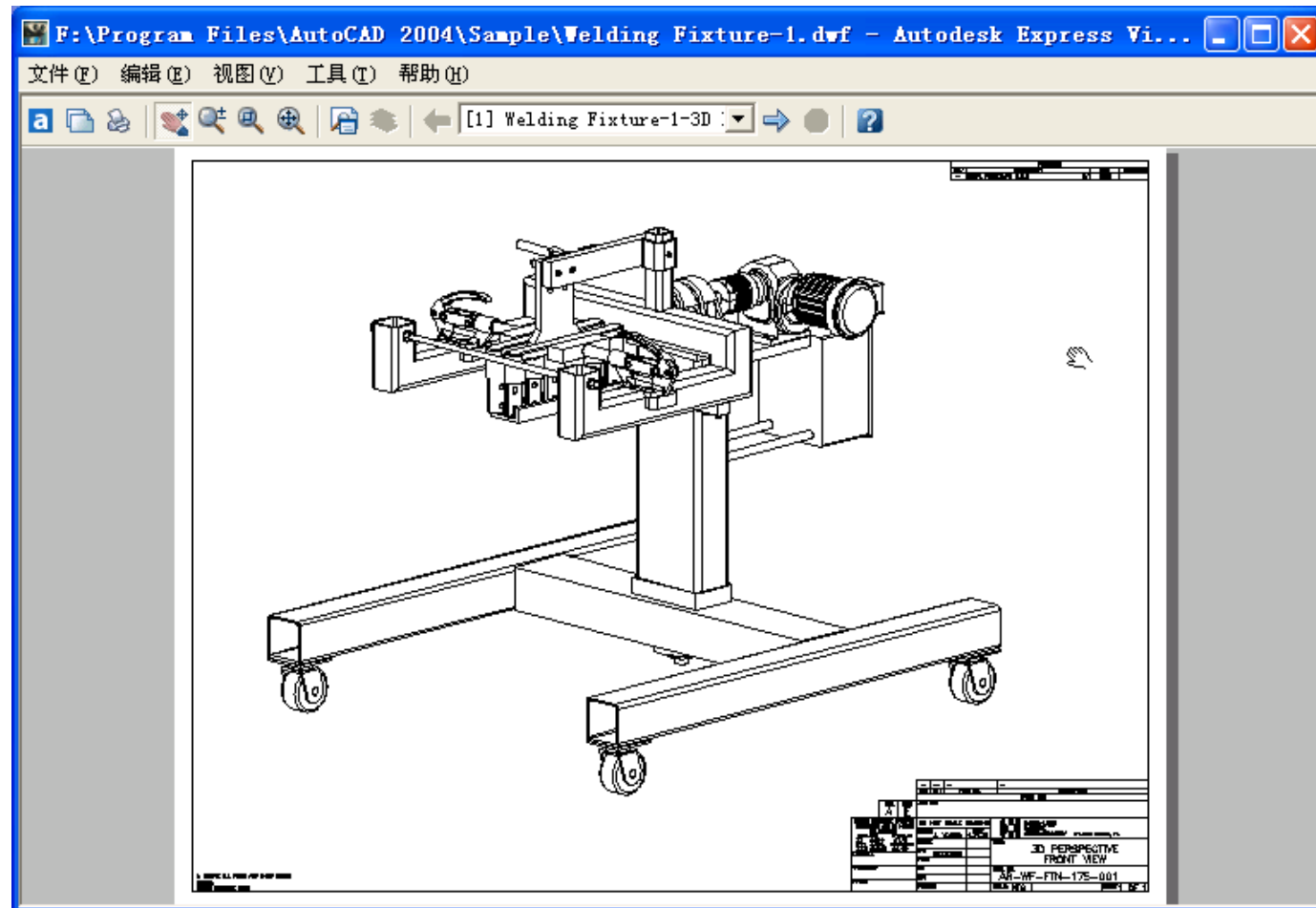
Robot Design

结构设计图（2D）实例：



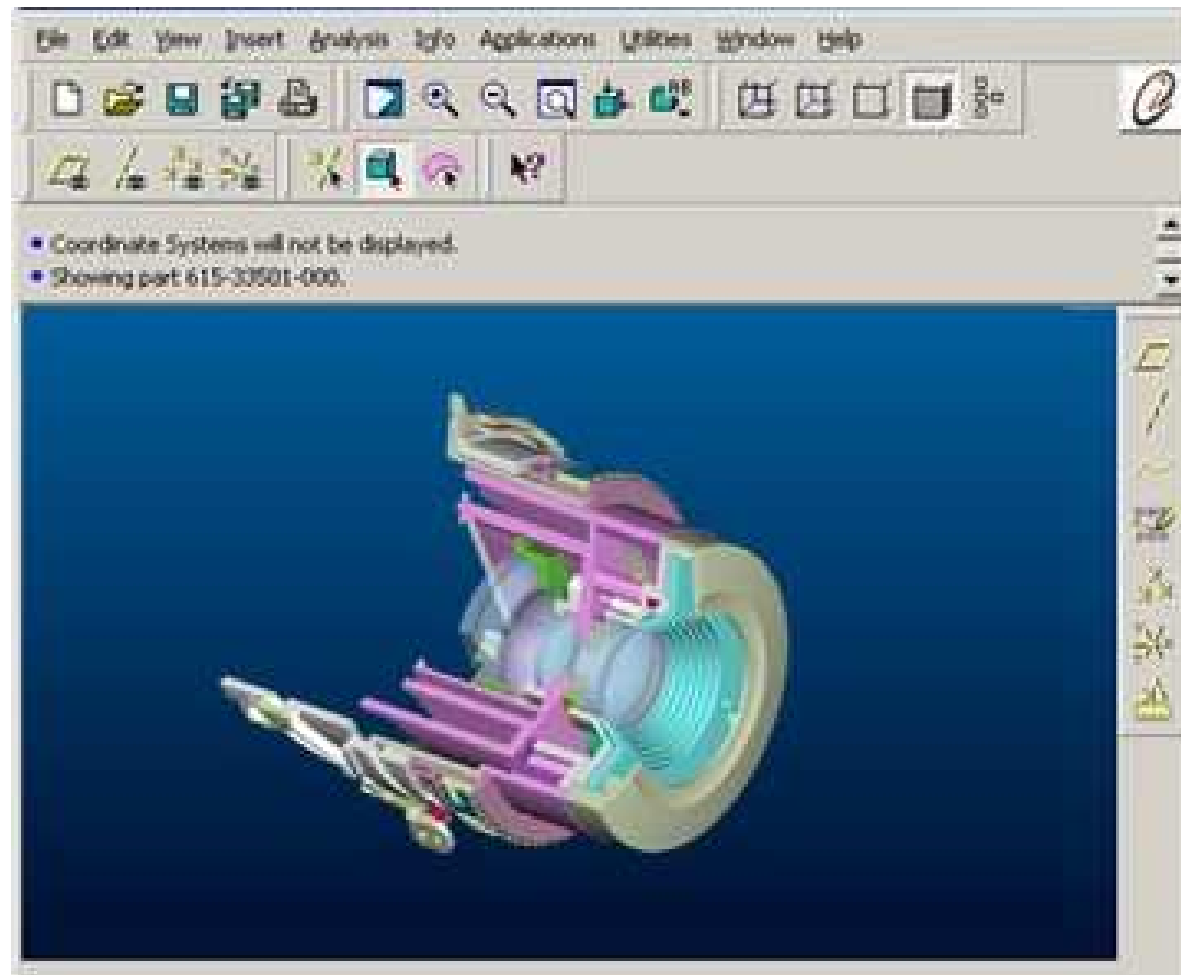
Robot Design

结构设计图（2D）实例：



Robot Design

结构设计图（PRO/E 3D）实例：



Contents



Control Design

Control System Design

❖ 什么是控制器？

控制器是机器人的核心，相当于人的大脑，它处理机器的外部信息和控制机器人的动作。

一般用在寻线机器人上的控制器是集成了CPU和一些CPU外部组件的集成块。

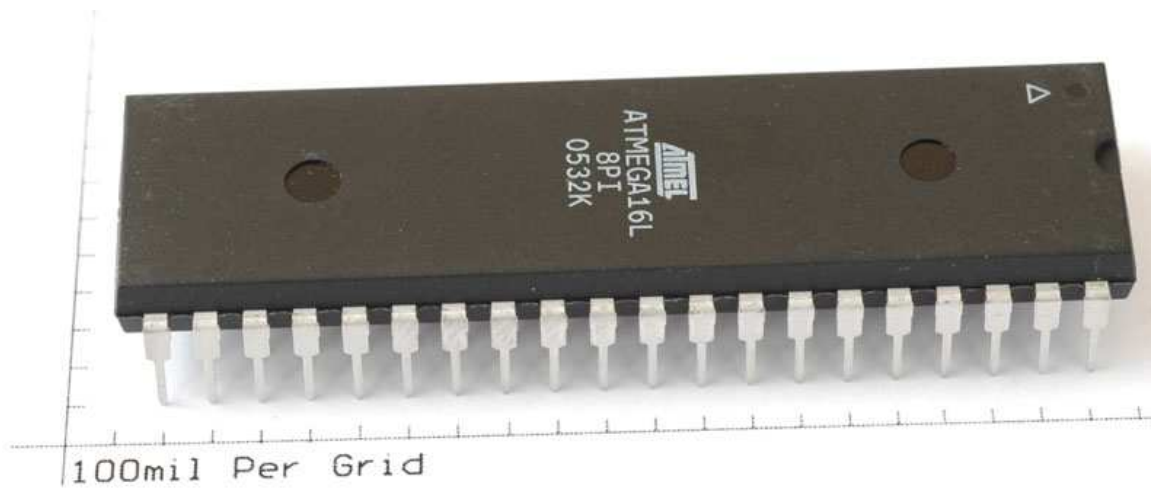
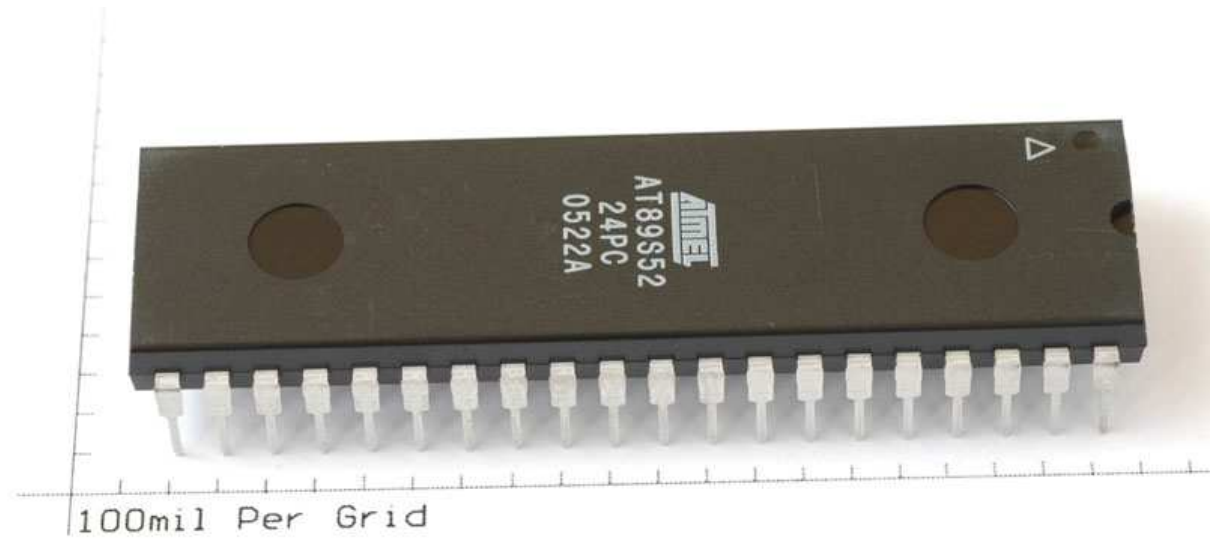
Control System Design

CPU需要选用，应注意的问题：

1. 选用的这个芯片主频有多高？（速度有多快？）
2. 这个芯片的片上资源有什么？比如：IO口数量、内部定时器、中断资源等
3. 使用这个芯片时，自己搭测试电路板方便吗？
4. 芯片价格如何？能否方便买到？
5. 芯片的封装是怎样的？
6. 这个芯片好开发吗？
7. 在网上能找到多少有关这个芯片的资料？比如：应用报告、使用范例

推荐芯片：AT89S52、ATmega16

Control System Design



Control System Design

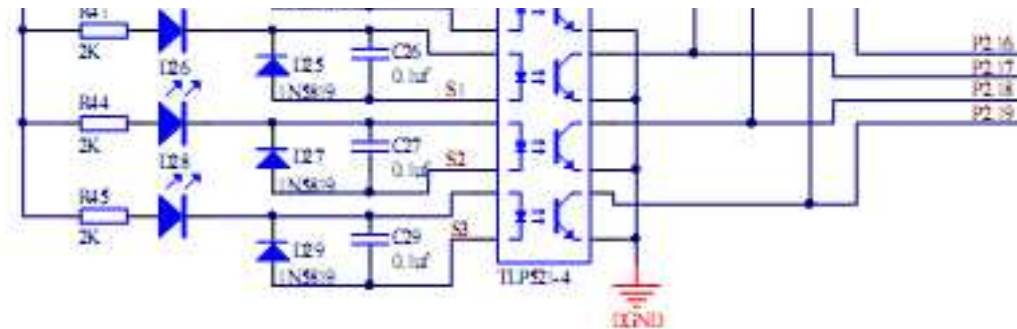
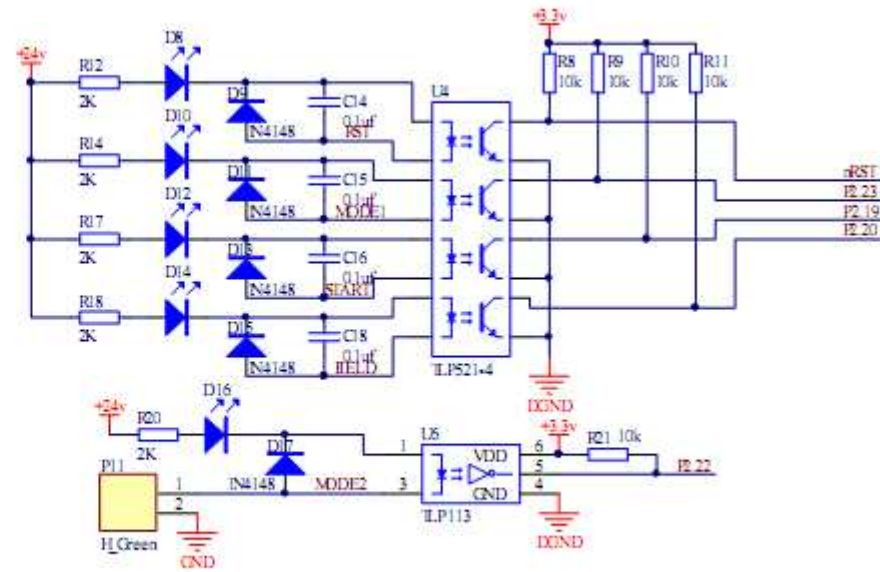
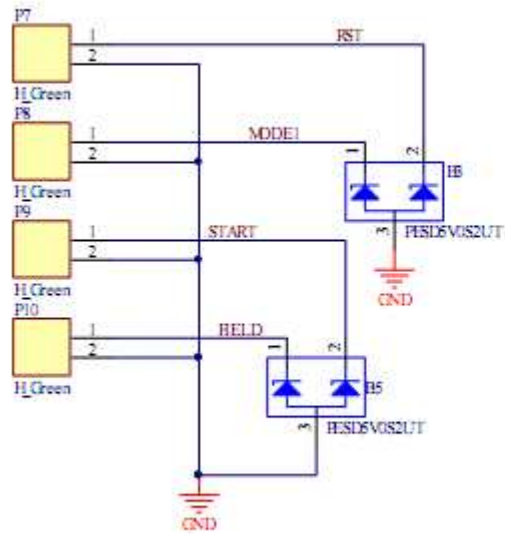
单片机编译器推荐

- ❖ Keil2.0（51系列）
- ❖ ICCAVR6.3（AVR系列）
- ❖ WinAVR（AVR系列）
- ❖ 程序下载软件推荐：[ispdown.exe](#)

Control System Design

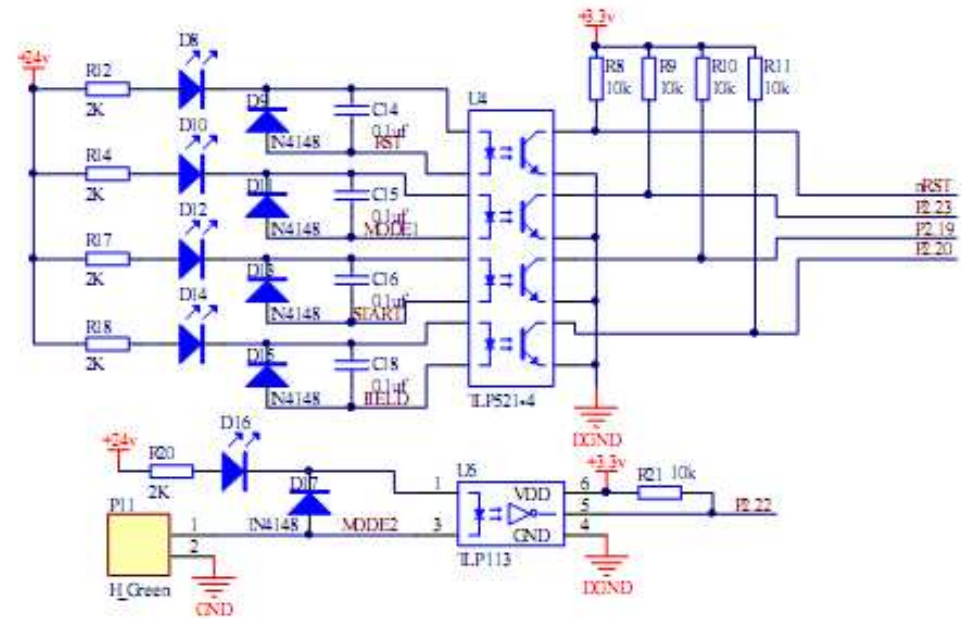
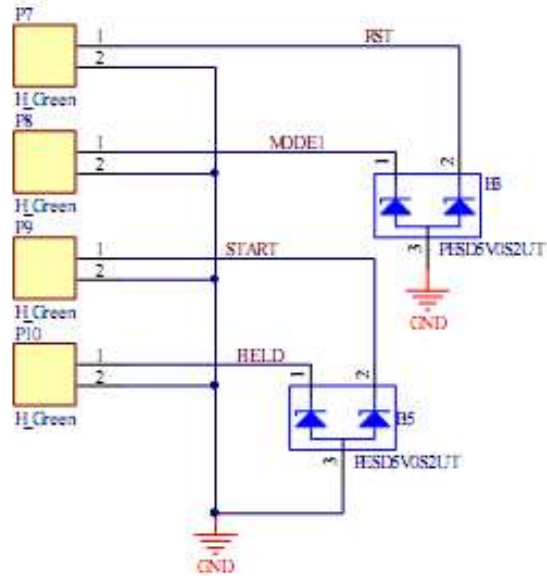
I/O资源分配

- ◆
- ◆
- ┆
- ◆
- ◆



Control System Design

接口电路的另一个例子



Control System Design

电机为机器人机动提供动力来源

电机一般分为：

直流电机（重点讲述）

交流电机

步进电机

伺服电机

Control System Design

直流电机的一些基本参数：

- ❖ 额定功率
- ❖ 额定电压和电流
- ❖ 最高转速
- ❖ 最大输出转矩

Control System Design

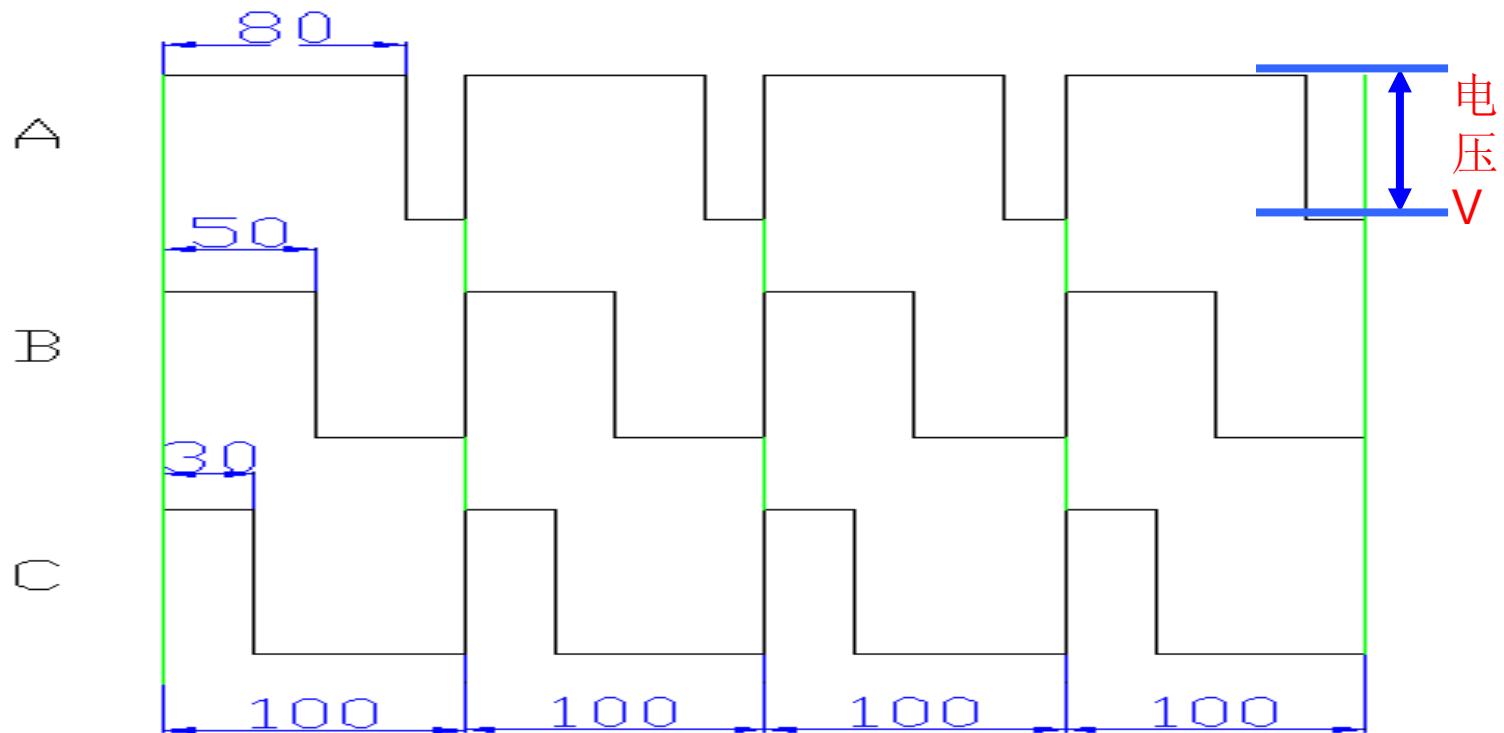
一般来说，电机的转速跟电压**U**有关，转矩跟电流**I**有关。

U 越高，速度越快； **I**越大，转矩越大。

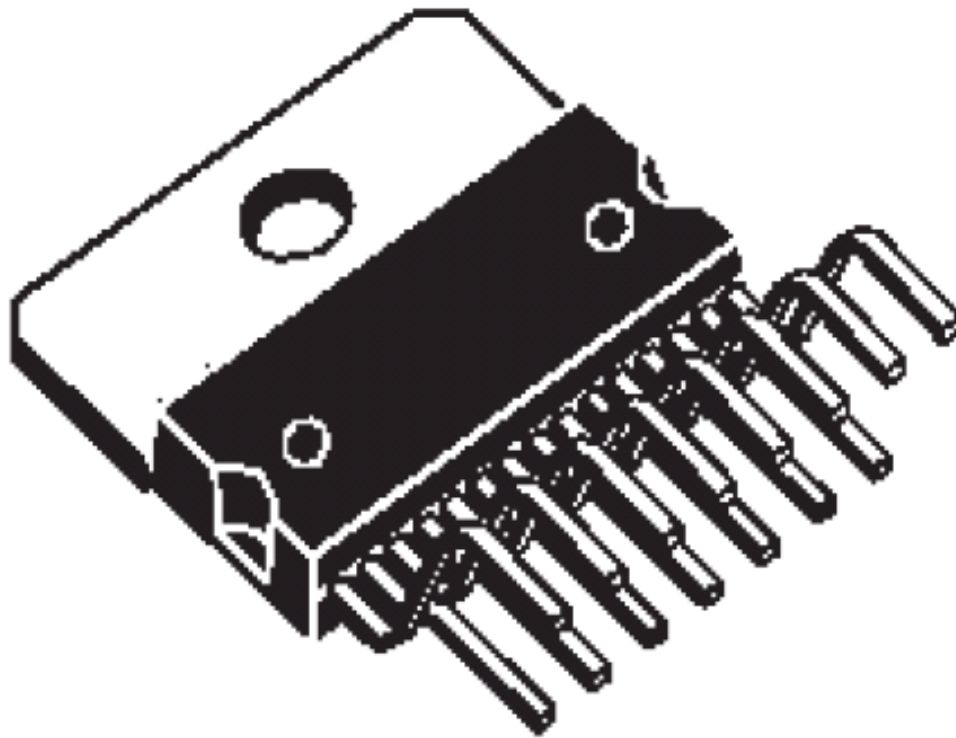
电机的调速一般我们采用PWM（脉冲宽度调制）方式。

Control System Design

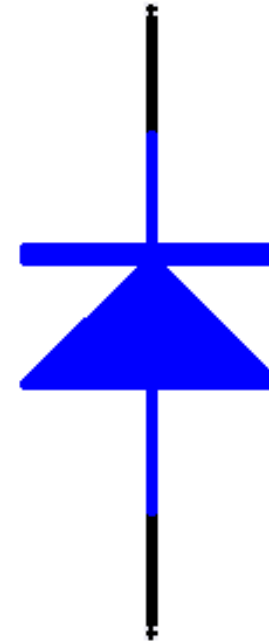
❖ PWM（脉冲宽度调制）：



Control System Design



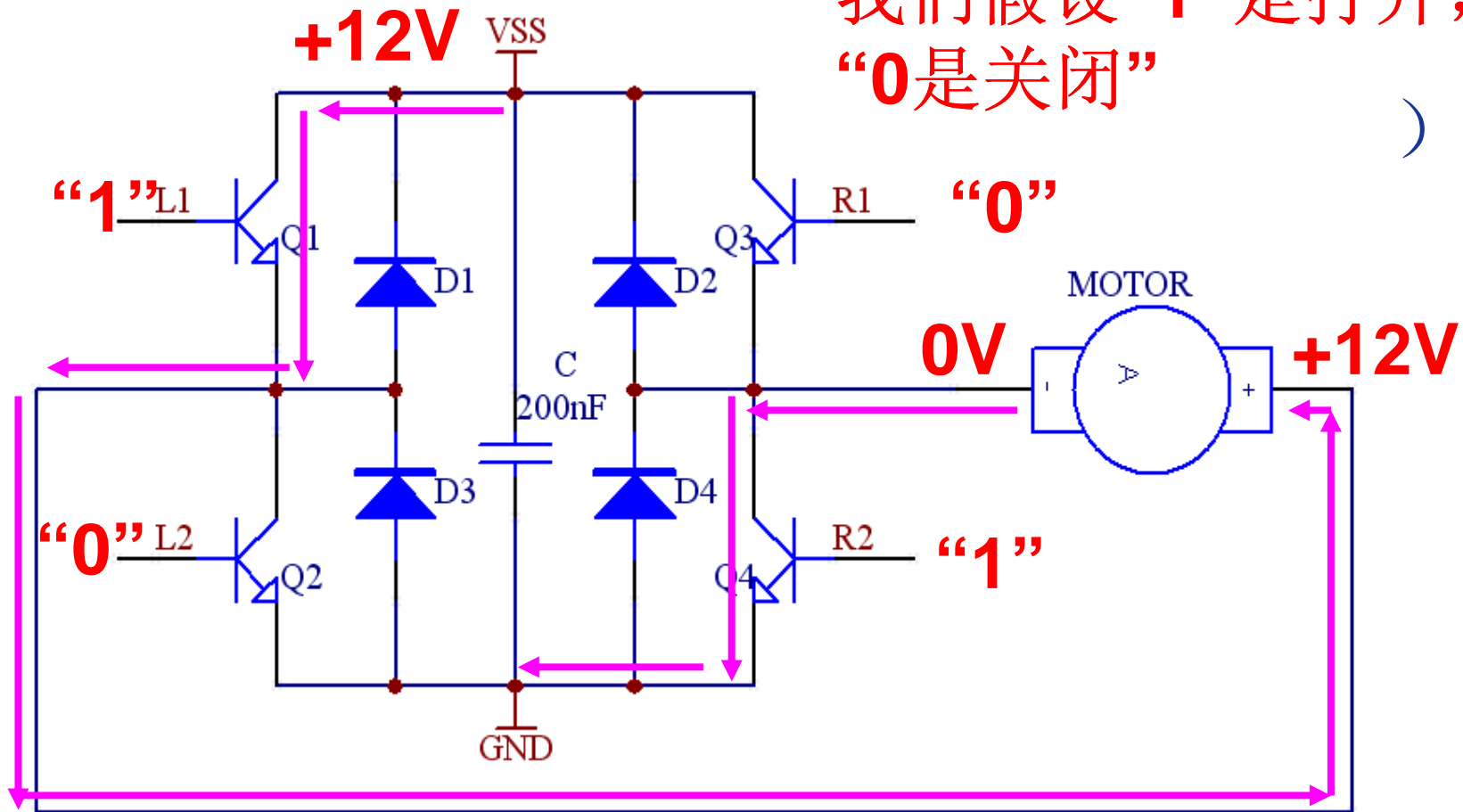
Multiwatt15



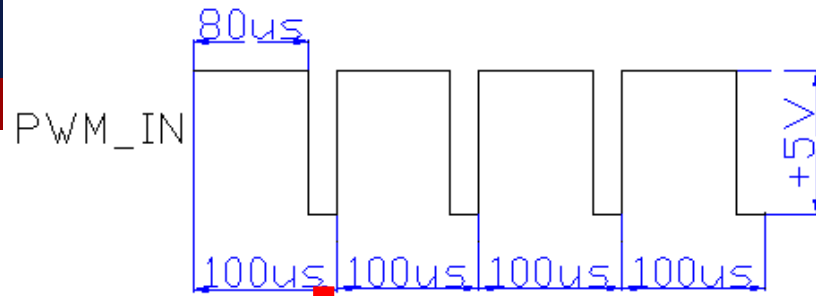
Control System Design

❖ 具体如何实现PWM? (方法一)

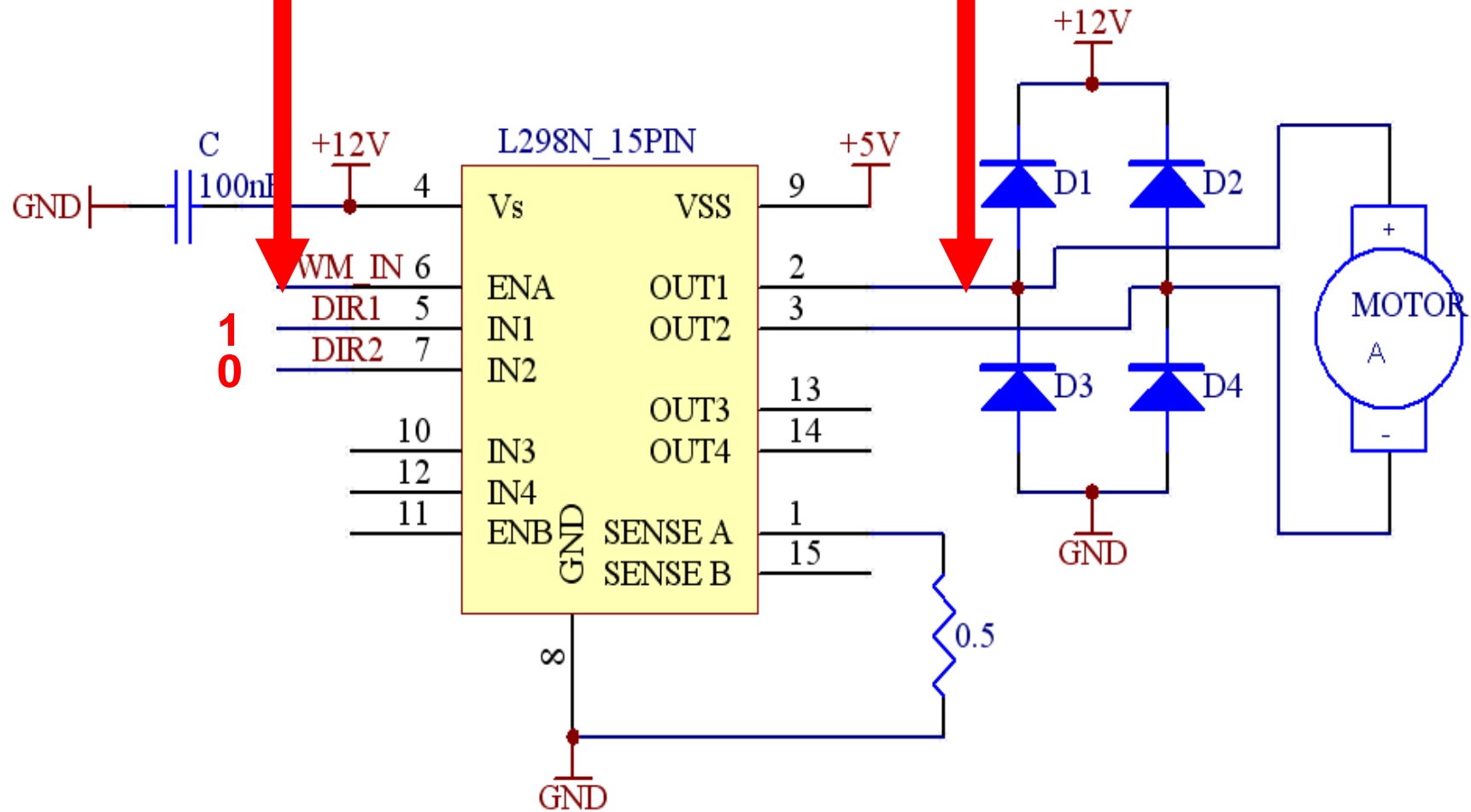
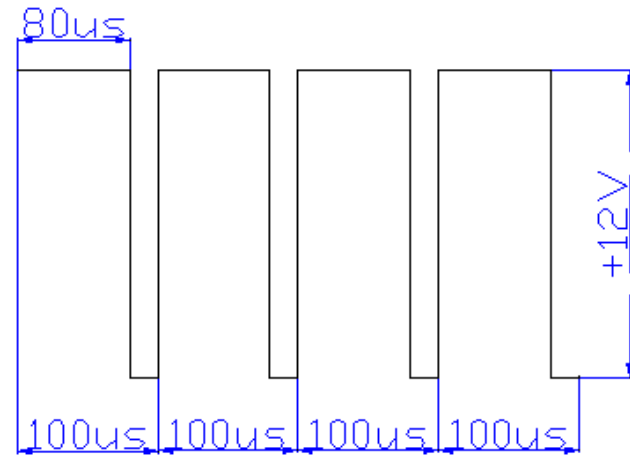
我们假设“1”是打开，
“0”是关闭



Control System

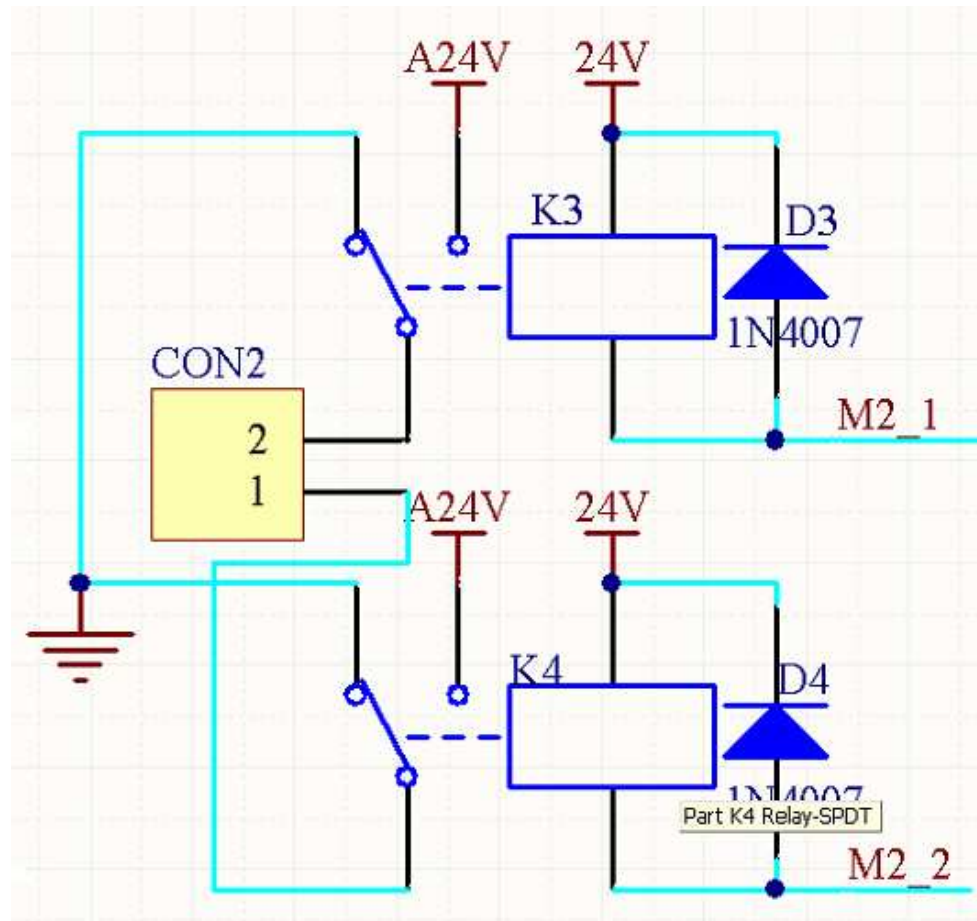


(方)



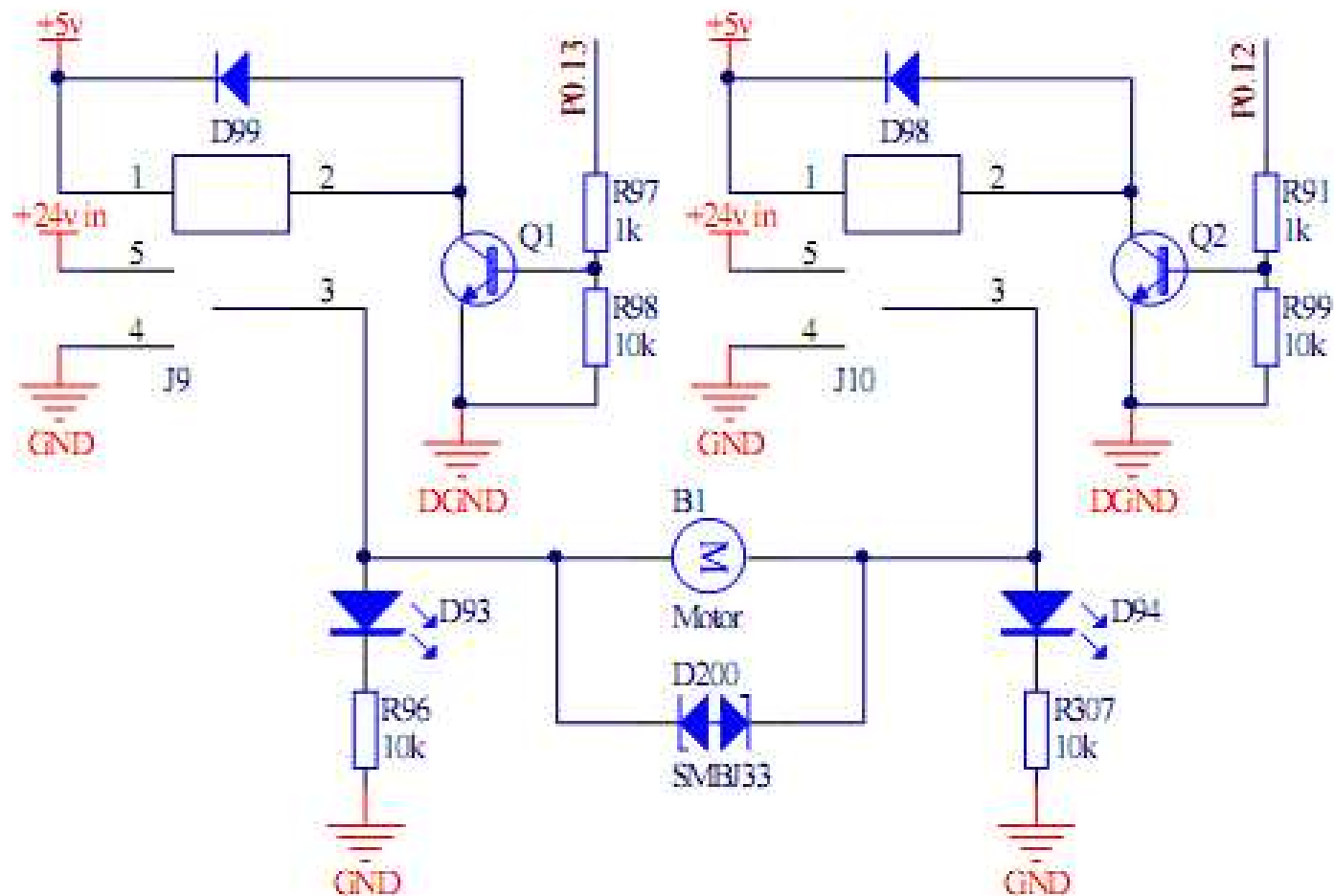
Control System Design

继电器控制电机的电路例子（08 robocon）



Control System Design

继电器控制电机的电路例子（09 robocon）



How to Follow Line

算法是整个控制系统的重要部分。

❖ 什么是算法？

说得通俗点就是完成一个任务的方法。

比如：。。。。。

❖ 为什么算法这么重要？

CPU处理任务就像人做事情一样，要有先后轻重、方法等等。

How to Follow Line

在寻线小机器人中，算法主要与下面几个有关：

- ❖ 电路硬件
- ❖ 传感器摆放位置（**最主要**）
- ❖ 机械结构（驱动轮的位置）

Contents



Sensor

Control System Design

❖ 什么是传感器？

传感器是机器人感知外部世界的器件，相当于人的眼耳鼻。

❖ 在比赛机器人上，可能用到的传感器种类

红外线传感器（如：ST188，TCRT5000）

触碰开关

CCD图像处理器（如果你的技术水平高的话）

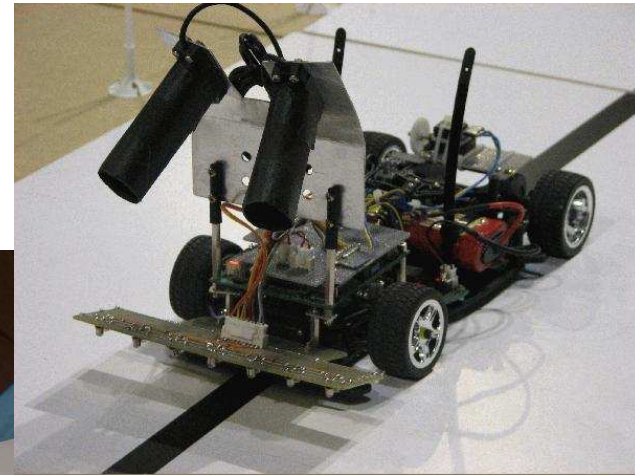
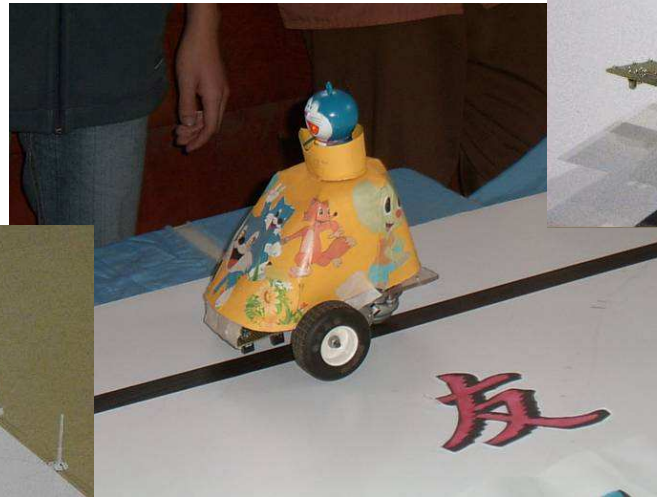
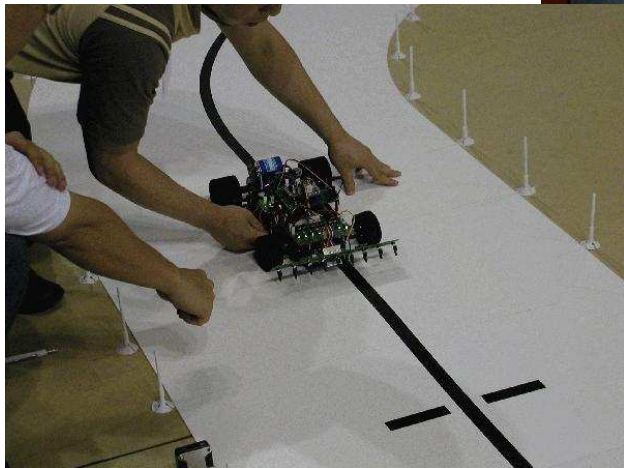
Control System Design

❖ 选用传感器时应注意的一些问题：

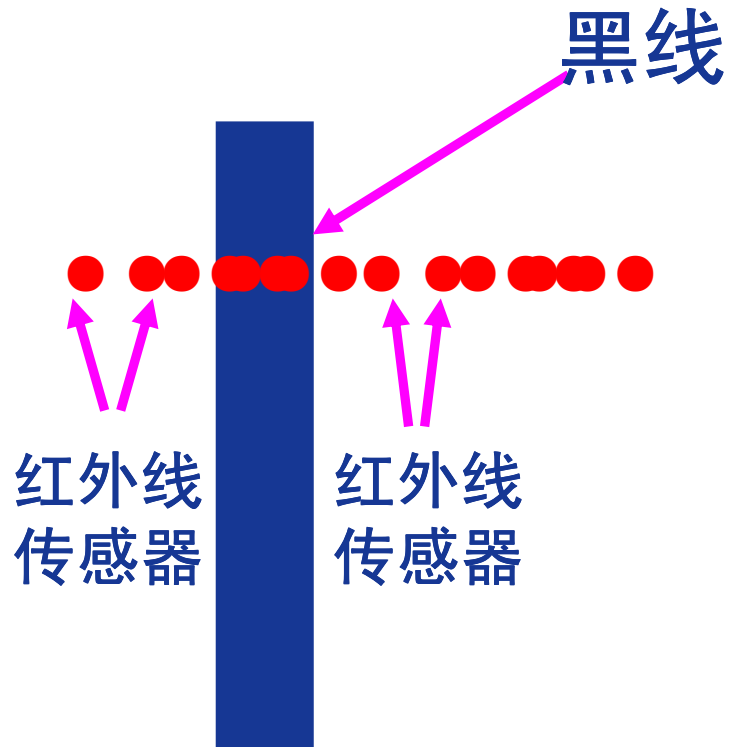
- 测量范围
- 供电电压和电流
- 允许的最大电压和电流
- 响应时间
- 输出量的形式
- 体积大小
- 价格

Control System Design

使用传感器时，一般都应有接口电路，要有抗干扰设计(可以在软件上实现，也可以在硬件上实现)。



How to Follow Line



传感器状态

1	2	3	4	
0	0	0	0	
0	1	0	0	右偏小
1	1	0	0	右偏中
1	0	0	0	右偏大

0 代表传感器不在线上

1 代表传感器在线上

How to Follow Line

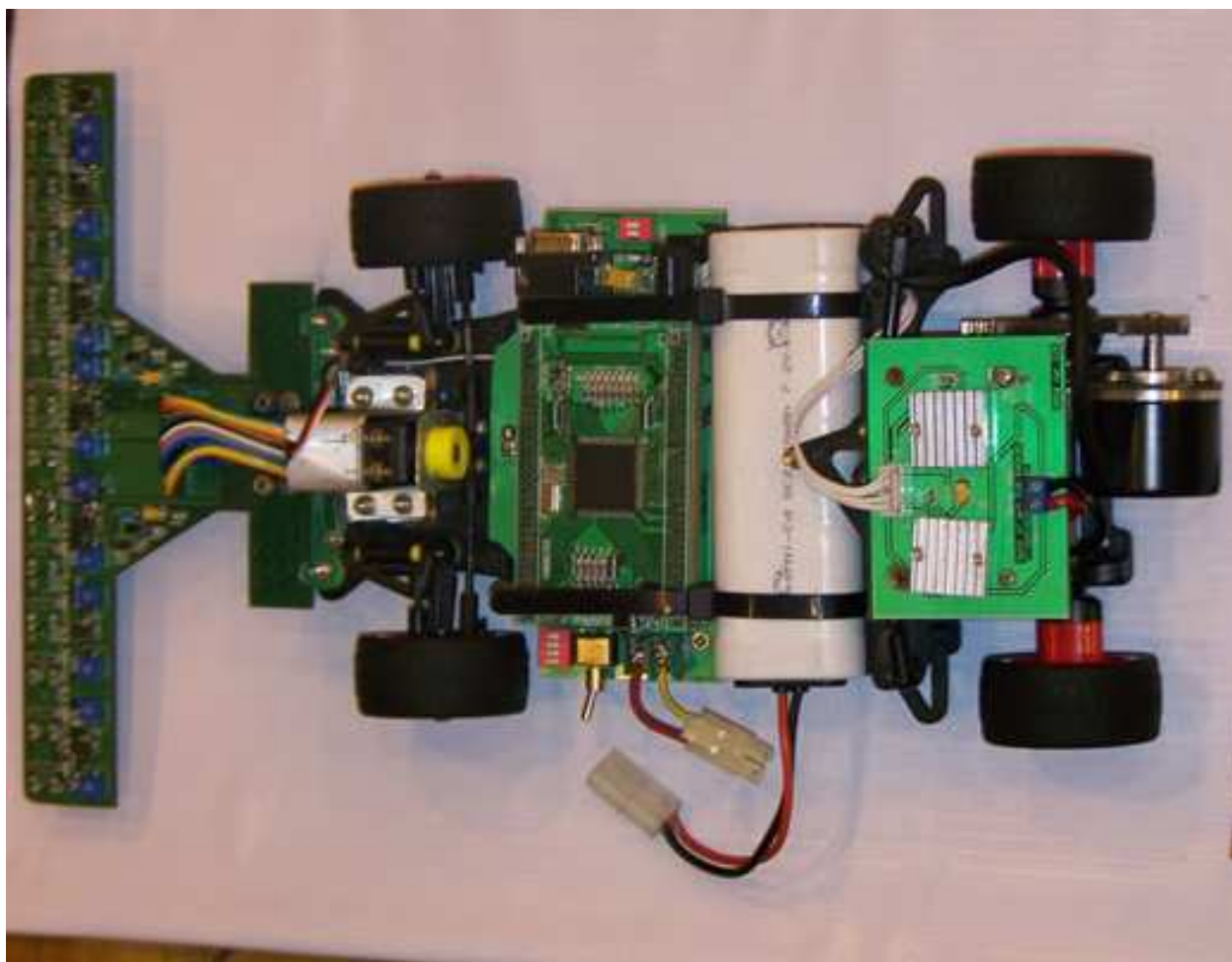
如果CPU知道机器人偏离了黑线，就要调整电机。

例如：机器人右偏时，就要向左转弯。

机器人左偏时，就要向右转弯。

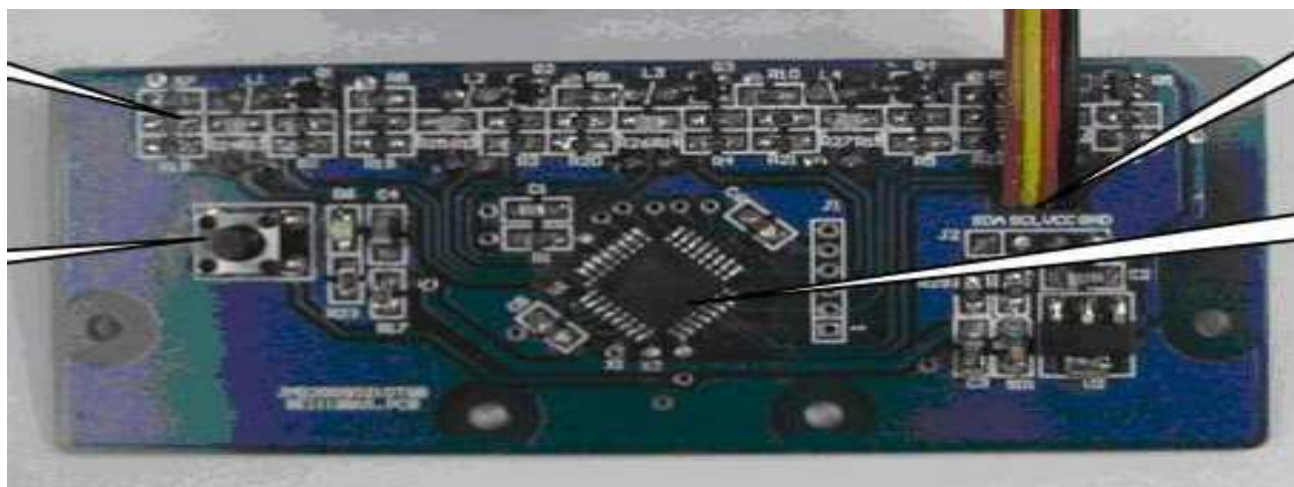
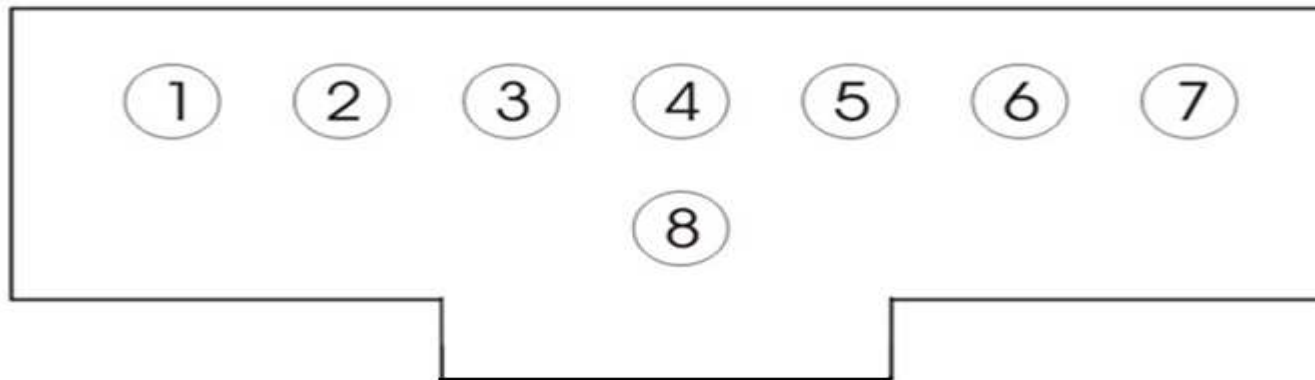
How to Follow Line

一个寻线传感器实例



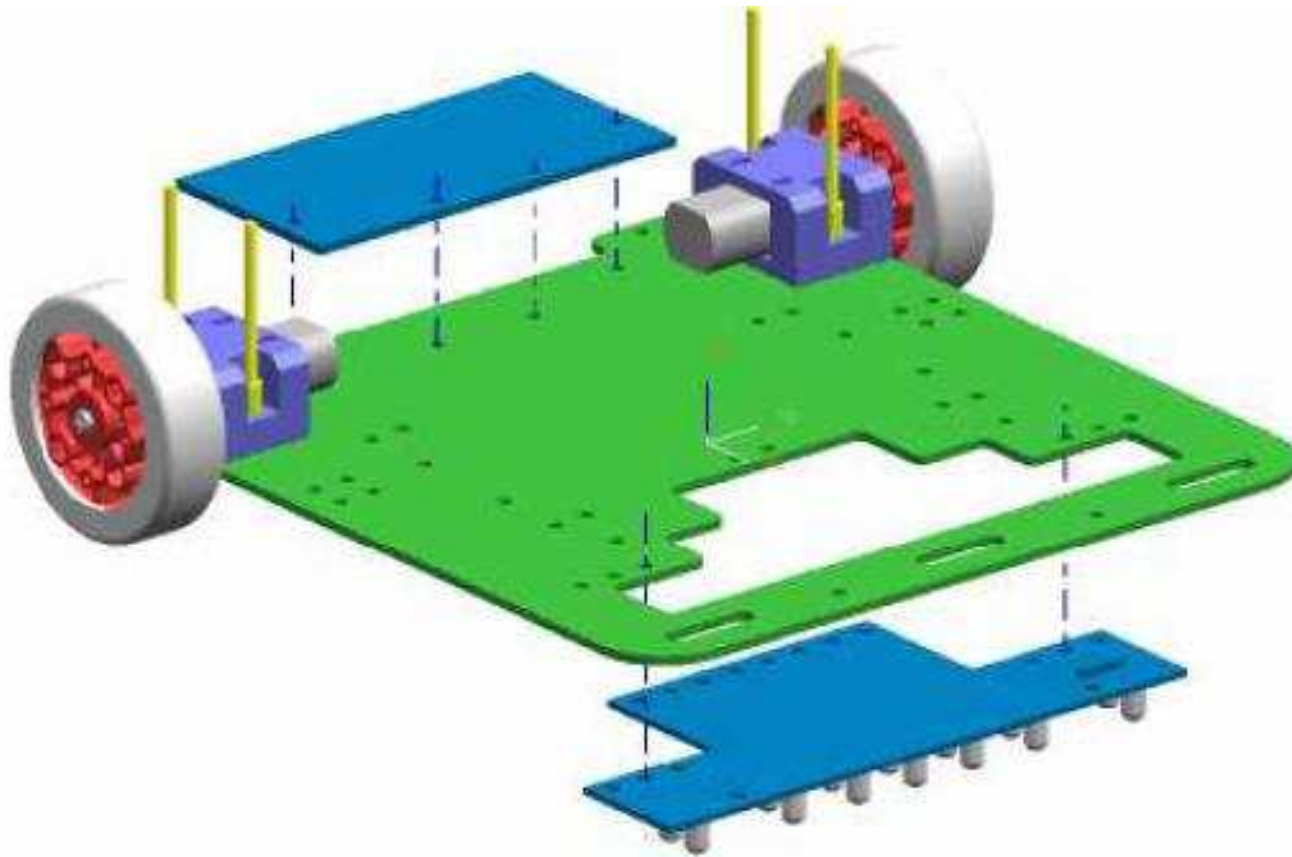
How to Follow Line

一个寻线模块的传感器布置方式(BE-1118循迹模块)



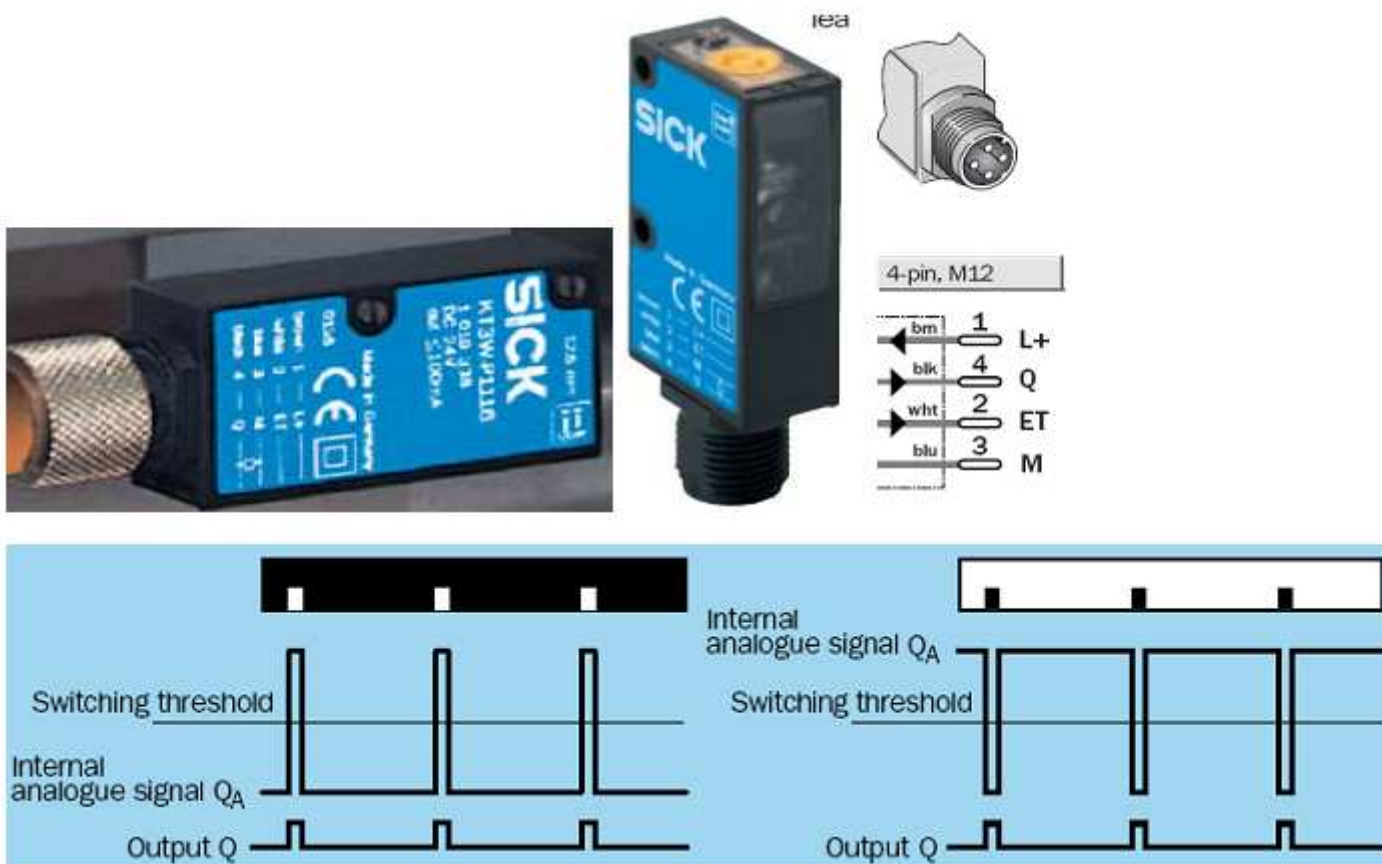
How to Follow Line

一个寻线模块的传感器布置方式(BE-1118循迹模块)



How to Follow Line

色标传感器



How to Follow Line

一些技术网站可能会找到你需要的资料

❖ www.robotdiy.com(有关机器人技术)

❖ www.ouravr.com(有关AVR单片机)

❖ www.21ic.com(有关器件信息)

Contents



Power Design

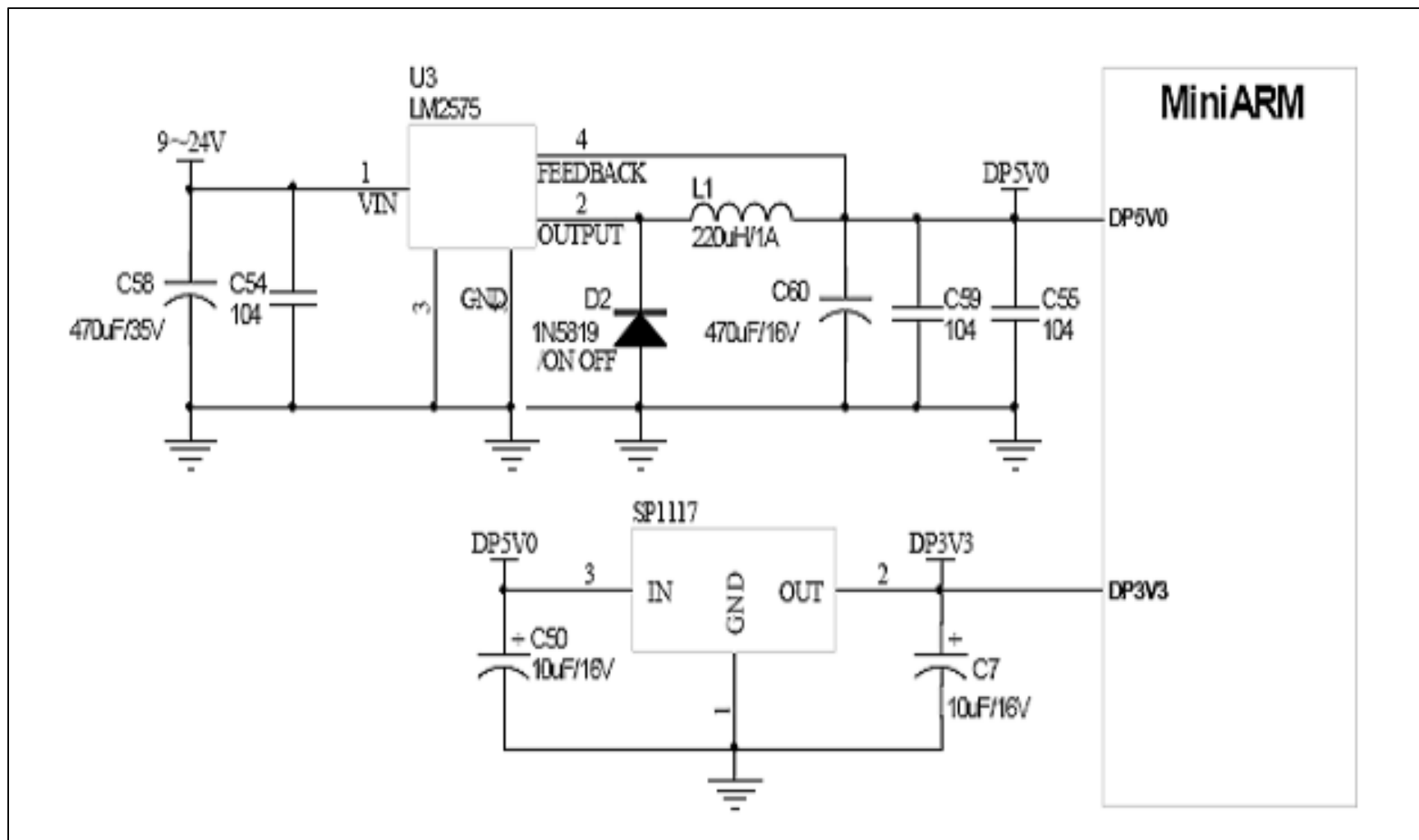
Robot Design

电源模块应采用瞬间过流大的充电电池（如镍氢电池等）通过电源变换电路，向电机和控制板供电。

- 为电机和控制系统提供电源，电池应分开。驱动回路的干扰可以导致 CPU Reset
- 瞬间大过流可保证电机迅速启动与强力对抗；
- 应有低压监测电路对低电压报警；
- 如采用高压脉冲干扰，应采用大电容充放电实现；
- 可通过逆变器为交流电机供电，或提升电压；
- 技术成熟，但电池选用应注意重量。

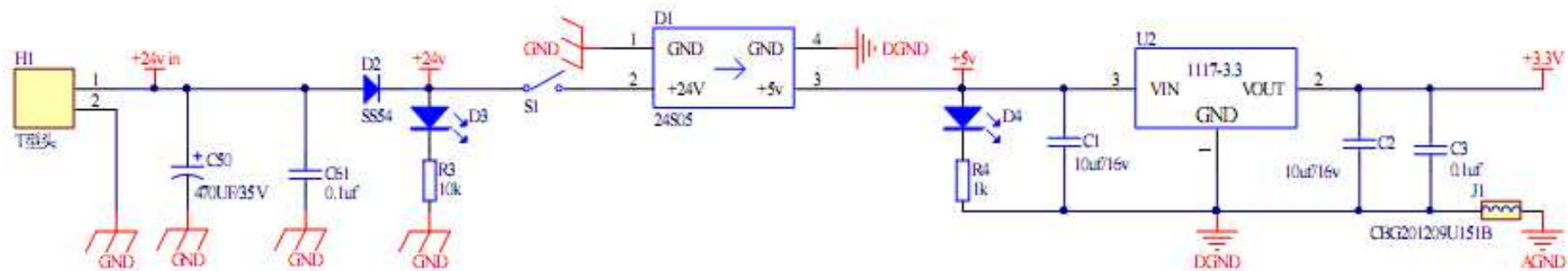
Robot Design

电源模块设计实例 (08 ROBOCON)



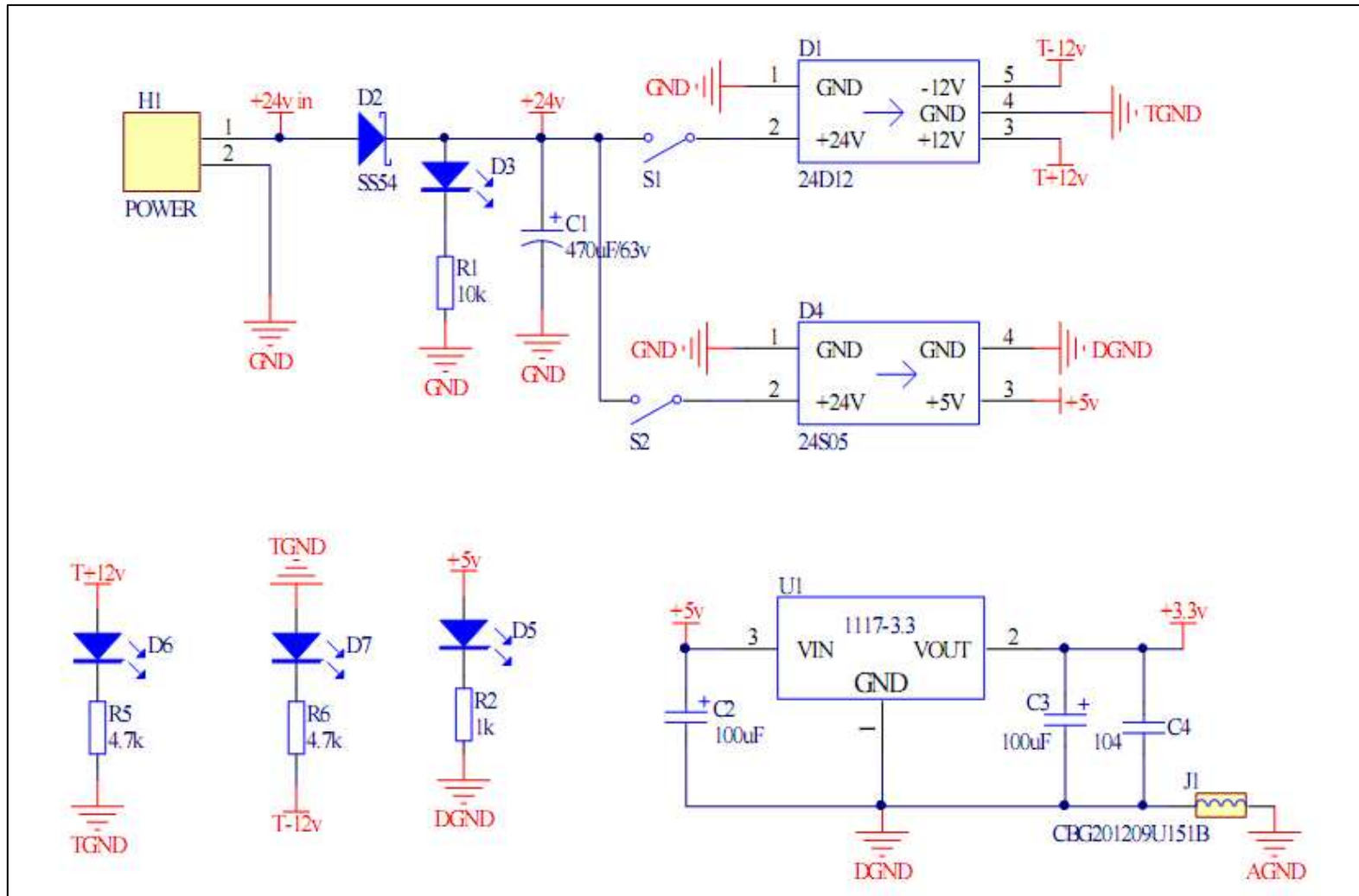
Robot Design

电源模块设计实例 (09 ROBOCON)



Robot Design

电源模块设计实例 (09 ROBOCON)



Contents



Programming

Programming

- 一. 硬件平台介绍 (ARM7TDMI-S)
- 二. 软件平台介绍
- 三. 软硬件平台的综合
- 四. 机器人控制系统
 1. 感知系统
 2. 控制系统
- 五. 机器人控制系统的综述

Programming

ARM7TDMI—S处理器是ARM7处理器系列成员之一，是目前应用最广的32位高性能嵌入式RISC处理器。ARM7TDMI—S采用3级流水线，以提高处理器指令的处理速度。ARM7TDMI—S核采用的是冯·诺依曼架构，也就是使用同一条32位长的总线对指令和数据进行寻址操作。ARM7TDMI—S有2个指令集：32位ARM指令集和16位Thumb指令集。

Programming

ARM7TDMI-S嵌入高速Flash存储器，采用3级流水线技术，取指、译码和执行同时进行，能够并行处理指令，提高CPU运行速度。由于具有非常小的尺寸和极低的功耗，抗干扰能力强，适用于各种工业控制。

机器人所用内核LPC2290，具有16KB片上SRAM，多种通信机制协议（如CAN，UART，I²C，SPI等），2个32位定时计数器、6路PWM输出和76个通用I/O口，使它特别适用于对环境要求较低的工业控制和小型智能机器人系统。因此选用该系列的芯片为主控制器，设计结构简单、性能稳定的智能机器人控制系统。

Programming

μ c/os-II 由Micrium公司提供，是一个可移植、可固化的、可裁剪的、占先式多任务实时内核，它适用于多种微处理器，微控制器和数字处理芯片（已经移植到超过100种以上的微处理器应用中）。同时，该系统源代码开放、整洁、一致，注释详尽，适合系统开发。

Programming

μ c/os-II 是一个嵌入式实时操作系统内核，包含了任务调度、文件管理、时间管理、内存管理和任务间的通信与同步等基本功能。 **μ c/os-II** 进行任务调度的时候，会把当前任务的CPU寄存器存放到该任务的堆栈中，然后再从另一个任务的堆栈中恢复原来的工作寄存器，继续运行另一个任务。

Programming

μ c/os-II 的几个特性：

- 1) 、 **可剥夺性的资源独享**
- 2) 、 **多任务调度管理**
- 3) 、 **中断服务**

Programming

μ c/os-II 操作系统采用的是可剥夺性的资源独享调度，任务先根据优先级排位，资源总是被较高优先级的任务独自占用。在任务执行过程中，较高优先级的任务一旦就绪，总能得到CPU的控制权。当一个比当前运行着的任务优先级高的任务进入了就绪态，当前任务的CPU使用权就被剥夺了，或者说被挂起了，那个高优先级的任务立刻得到了CPU的控制权。

Programming

μ c/os- II 的几个特性：

- 1) 、可剥夺性的资源独享
- 2) 、多任务调度管理
- 3) 、中断服务

Programming

多任务运行的实现实际上是靠CPU(中央处理单元)在许多任务之间转换、调度。CPU只有一个，轮番服务于一系列任务中的某一个。多任务运行很像前后台系统，但后台任务有多个。多任务运行使CPU的利用率得到最大的发挥，并使应用程序模块化。在实时应用中，多任务化的最大特点是，开发人员可以将很复杂的应用程序层次化。使用多任务，应用程序将更容易设计与维护。

Programming

μ c/os-II 的几个特性：

- 1) 、可剥夺性的资源独享
- 2) 、多任务调度管理
- 3) 、中断服务

Programming

中断是一种硬件机制，用于通知CPU有个异步事件发生了。中断一旦被识别，CPU保存部分（或全部）现场（Context）即部分或全部寄存器的值，跳转到专门的子程序，称为中断服务子程序（ISR）。对可剥夺型内核而言，中断就是先打断正在执行的较低优先级的任务，让进入就绪态的较高优先级的任务开始运行，在完成较高优先级的任务后再回到原来的任务执行进度，继续未完成任务。中断特性更好的体现了“重要的事情先处理”的原则。

Programming

软硬件平台的综合实际就是将 $\mu\text{C}/\text{OS-II}$ 嵌入式操作系统移植到ARM7平台上，完成软硬件的结合，以便我们可以通过软件编程来实现相关控制的功能。在此过程中就要注意移植的相关问题，如ARM内核提供的指令集， $\mu\text{C}/\text{OS-II}$ 的文件管理和进程管理那些与底层硬件有关等。

Programming

1. 操作系统的移植指的是使实时系统的内核能在微处理器上运行。

2. 在 μ C/OS-II 的移植过程中，处理器必须满足以下要求：

- 1)、处理器的C编译器能产生可重入型代码
- 2)、处理器支持中断，并且能产生定时中断
- 3)、用C语言就可以开/关中断
- 4)、处理器能支持一定数量的数据存储硬件堆栈
- 5)、处理器有将堆栈指针以及其他CPU寄存器的内容

Programming

实际上， μ C/OS-II 可以简单地看作是一个多任务调度器，在这个任务调度器上添加了与多任务操作系统相关的一些系统服务，如信号量、中断等。其90%的代码是用C语言写的，可以直接移植到有C语言编译器的处理器上。移植主要都集中在多任务切换的实现上，因为这部分代码用来保存和恢复CPU现场（即写 / 读相关寄存器），不能用C语言，只能使用汇编语言完成，即编写OS_CPU_A.S文件。另外还需要修改与ARM体系结构相关的OS_CPU.H文件和用户规定任务栈初始化结构的OS_CPU_C.C文件。

Programming

OS_CPU.H 这个头文件包括了用#define语句定义的与处理器相关的常数/宏以及类型等，如：

1) 临界段代码宏定义：OS_ENTER_CRITICAL()和OS_EXIT_CRITICAL()，可以提供三种办法开关中断；2) 、堆栈段生长方向：OS_STK_GROWTH，常数定义了堆

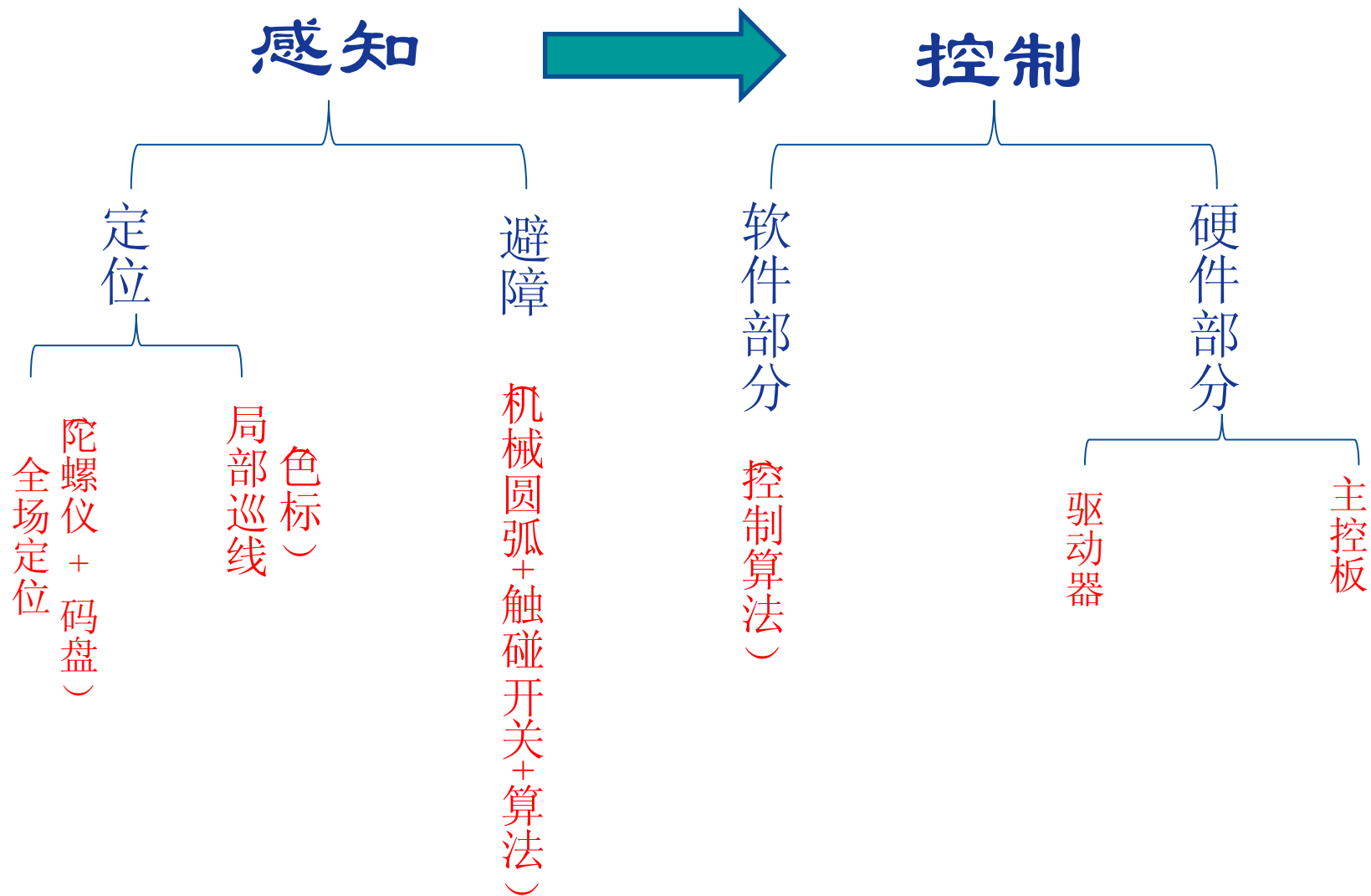
栈的方向，值0时表示堆栈从下往上递增，值1时表示堆栈从上往下递减；

3) 任务调度函数宏定义：OS_TASK_SW()，是在操作系统从低优先级切换到高优先级任务时用到的。

Programming

移植过程中应注意一系列数据类型的定义，以适应不同微处理器的字长。uC/OS-II的代码中不使用C语言的short, int等数据类型，因为这些数据类型是与处理器相关，不可移植的。程序代码中定义的数据类型字长应根据处理器C编译器规定的数据类型字长而定。上面的数据类型移植是根据ADS编译器对C语言各类型的字长要求定义的。

Programming



Programming

感知系统对于一个机器人就相当于人类的眼耳等器官，是对环境的感受和反应的基础。在机器人中完成感知功能的是各类内外的传感器。通过各类相关的测量，来衡量各类相关指标，以判断相关的机器人的位置、状态等。在我们的机器人中主要是要知道机器人运动的位置等参数，以作出实时的控制和调整。所以我们主要用到了全场定位和局部巡线组成的简单感知系统。

Programming

由于机器人是一个对外封闭的系统，对外界的环境是未知的，所以需要通过传感器来测定其自身的位置，以确定自身在环境中的运动状态。全场定位是一种以某一固定参考点为原点建立直角坐标系，以 (X, Y, θ) 的形式表示实时运动相对位置的定位方法。常用角度和距离的积分实时累积来完成坐标的转换。角度常用陀螺仪量定，距离则用码盘量定。

Programming

由于我们机器人的功能的特殊性，在直线段有机器人上轿的任务，这时产生的外界扰动（如冲击，晃动等）可能会对原始的全场定位产生影响，以至于对后续路径不可控，所以需要借助相应的修正措施。在直线上我们就考虑到采用色标来进行局部巡线以达到完成修正的目的。

Programming

避障是通过机械上的圆弧栏杆和控制上的场势效应算法来实现的。机械圆弧栏杆安放在自动轿夫的前端，在其末端有触碰开关连工控板。通过其物理顺滑特性和触碰开关引导，来完成一定的避障功能。控制上的场势效应算法是来源于电磁场效应抽象的算法。先将障碍信息坐标化，并以此建立效应圈，通过根据定位系统的自身坐标信息的反馈，当机器人离障碍越近时，产生的向外排斥的趋势越大，这样就可以比较好的完成避障的任务。

Programming

机器人的控制系统就是控制机器人怎样运动，怎样完成相应功能等具体行为的软硬件集合，是机器人行为指挥系统，具有中枢的作用。其中涉及到软硬件的综合和优化，怎样搭配和使用软硬件资源以使整个系统的达到功能输出的最优化。在我们的机器人中，就涉及到怎样设置和使用EPOS，怎样进行软件上的路径优化控制，怎样使系统有更稳定的输出。

Programming

驱动器是产生电流以驱动电机运转的装置，我们采用的是MAXON公司的EPOS，该设备可以通过编程来精确的控制电流，以达到控制电机的功率输出。工控板是控制的核心硬件基础，是控制系统的中枢神经，装载有程序，通过实时的运行和计算，完成具有控制功能指令的输出。控制系统先进行工控板的计算，并发出相应指令，再通过CAN总线传达EPOS，EPOS转化为相应的电流输出到电机，以完成整个控制操作的完成。

Programming

我们机器人在设计上具有独立的驱动配置（EPOS+电机+驱动全向轮），这样可以在控制中独立的控制各个轮的运动方向以控制机器人的运动方向。这样设计，在算法上就有了更高的要求，在顶层机器人的算法综合效果上表现为唯一性，但在底层驱动全向轮的控制算法上表现为多样性。所以算法也是根据各种需求，在调试过程中融合相关算法，其中主要的还是反馈和PID算法。

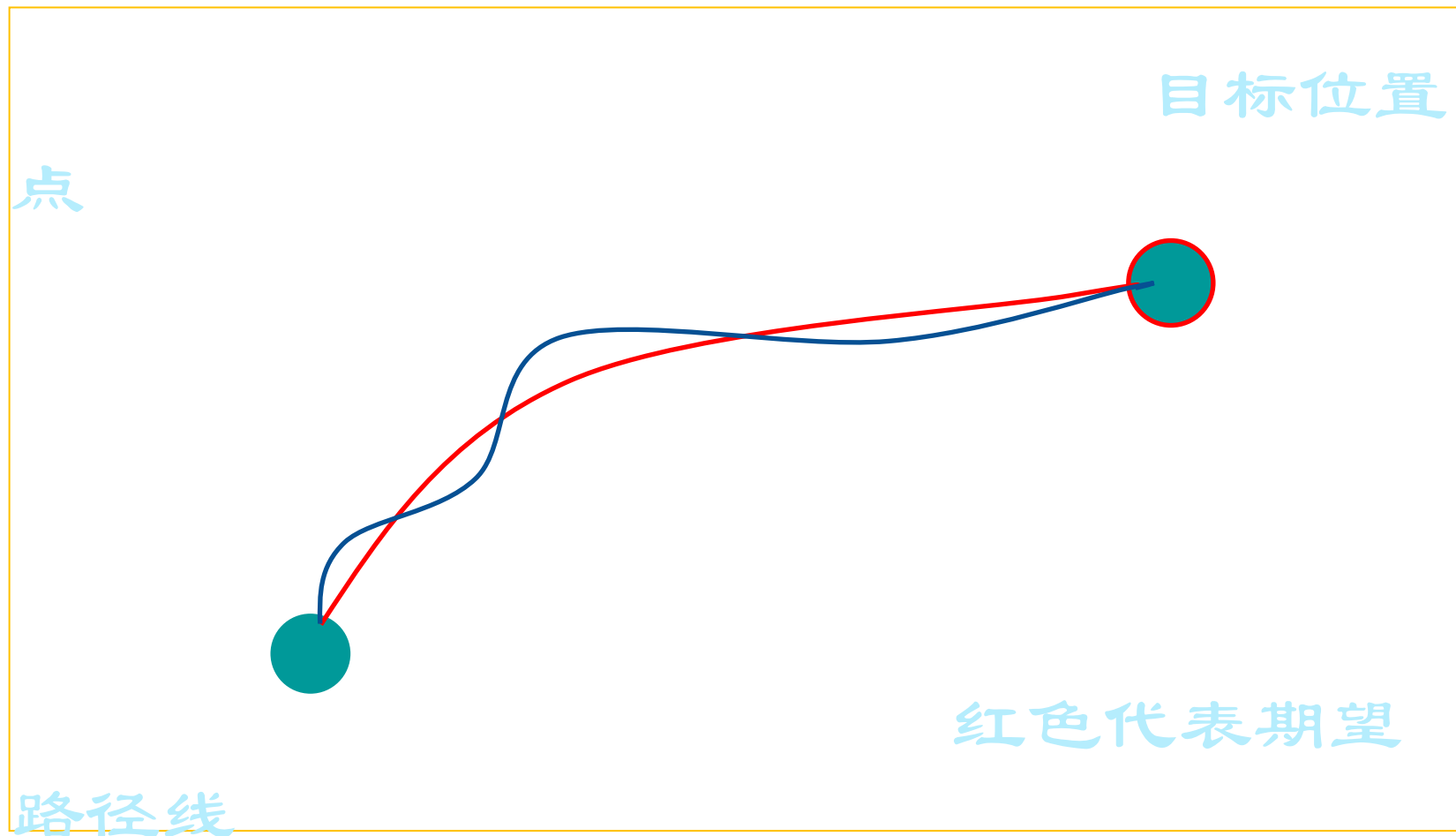
Programming

鉴于我们机器人任务和要求的特殊性，场地的多样和复杂性，并且没有现成固定的控制算法适应我们的机器人，所以我们在机器人控制过程中需要根据实际的需求来融合现有的算法，以达到机器人的较精确控制。在此过程中我们分段分任务设计算法，直线段用反馈+PID算法控制，过树林段用反馈算法+PID算法+势场效应法控制。同时，我们设定一个算法综合的标准，以完成不同算法间的过渡和联系，这样以达到控制效果的连续性和系统的稳定性。

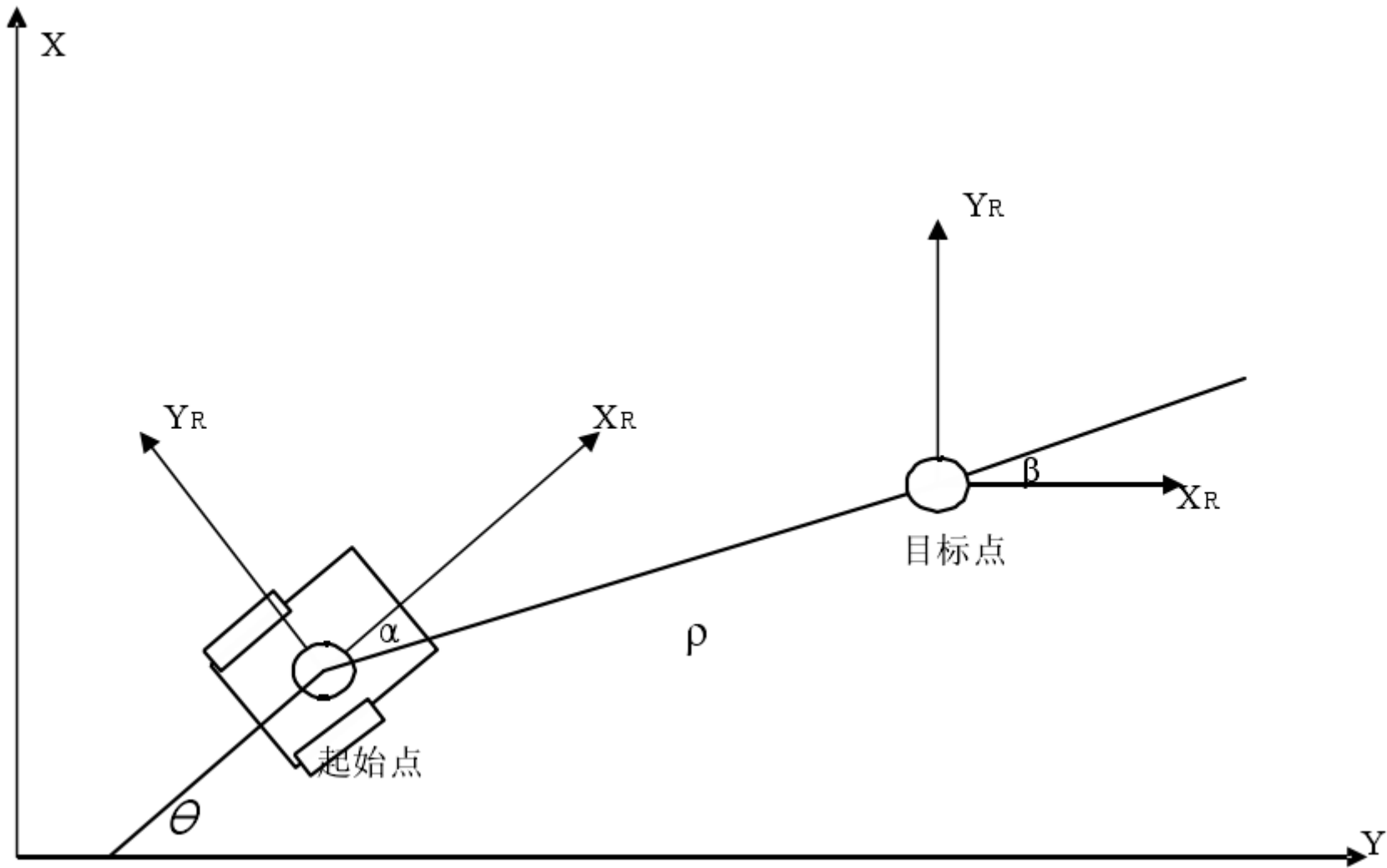
Programming

反馈控制算法是一种点镇定的路径控制方法。所谓点镇定，是一种路径控制的方法，即为已知两终端的状态，按一定的规律来实时生成参数以控制中间的状态。通俗上说就是终端已知，中段未知的控制方法。

Programming



Programming



Programming

设计控制信号 v , w (线速度, 角速度), 以把它从实际位置以曲线驱动到目标位置。机器人参考框架 $\{X_R, Y_R, \theta\}$, 和全局框架 $\{X, Y, \theta\}$ 。可以设计如下控制率:

$$\begin{cases} V = k_{\rho} \bullet \rho \\ W = k_{\alpha} \bullet \alpha + k_{\beta} \bullet \beta \end{cases}$$

式中 k_{ρ} , k_{α} , k_{β} 为控制参数; α , β 表示在 $(-\pi, \pi)$ 中;

Programming

控制参数的稳定性：运动路径依赖于参数 k_ρ ， k_α ， k_β ，要使机器人路径收敛于目标点，则可以证明参数需满足：

$$\begin{cases} k_\rho \geq 0, \\ k_\beta \leq 0, \\ k_\alpha - k_\rho \geq 0 \end{cases}$$

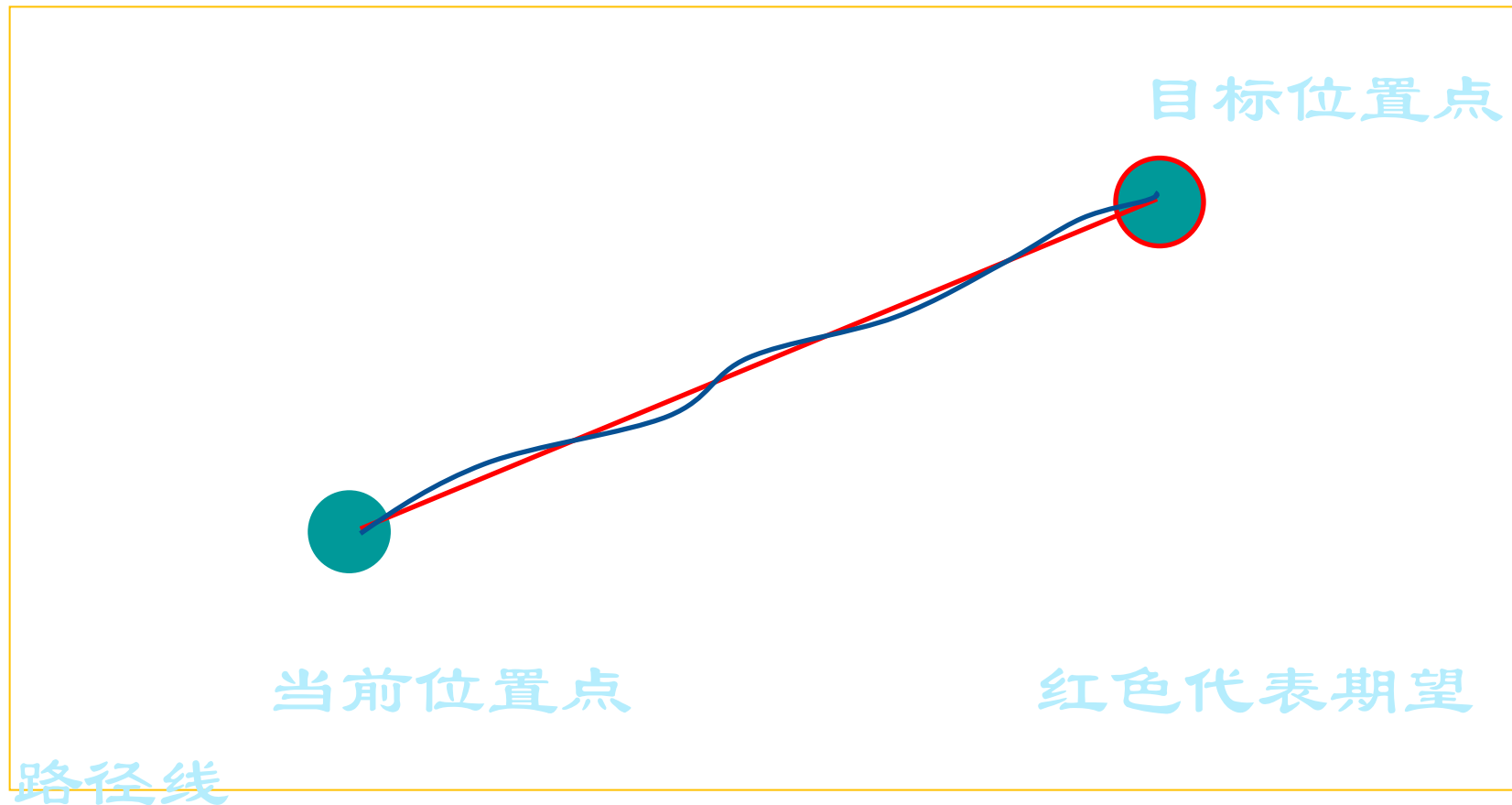
为鲁棒位置控制，建议用强的稳定性条件，保证在机器人到达目标时，不改变方向：

$$\begin{cases} k_\rho \geq 0, \\ k_\beta \leq 0, \\ k_\alpha + 5/3 k_\beta - 2/\pi k_\rho \geq 0 \end{cases}$$

Programming

PID控制算法是一种路径实时控制的过程控制方法，在控制过程中，按实时偏差的比例（P）、积分（I）和微分（D）进行相应的参数选择，来完成过程的精确调整和控制。其中P参数使系统保持与预定相符的运动趋势，I参数使系统稳态误差逐渐消除，但会加剧系统的振荡，D参数使系统的振荡减小。合适的P、I、D参数可以使系统在预定的运动趋势上逐渐靠拢预定的轨迹。

Programming



Programming

控制机器人以直线行走，假设机器人在全局坐标系中的位置为 (x, y) ，目标坐标为 (x_0, y_0) ，两点可确定一条直线，设直线方程为 $AX + BY + C = 0$ ，系数 A, B, C 可以由下式得出：

$$\left\{ \begin{array}{l} A = y_0 - y; \\ B = x - x_0; \\ C = x_0 * y - x * y_0; \end{array} \right.$$

要驱动机器人沿此直线运动，假设机器人沿此直线运动的过程中坐标为 (X, Y) ，由于控制有误差， (X, Y) 不是精确在此直线上，那么可以求出点 (X, Y) 到直线的距离，也就是运动的误差

$$\left\{ \begin{array}{l} \text{ERROR} = (AX + BY + C) / E \\ E = \text{sqrt}(A * A + B * B); \end{array} \right.$$

Programming

在机器人控制过程中，我们主要采用增量PID算法控制此误差. 偏差量

$$e(k)=\text{Error}=(AX+BY+C)/E;$$

$$\left\{ \begin{array}{l} \Delta e(k) = e(k) - e(k-1); \\ \Delta e(k-1) = e(k-1) - e(k-2); \end{array} \right.$$

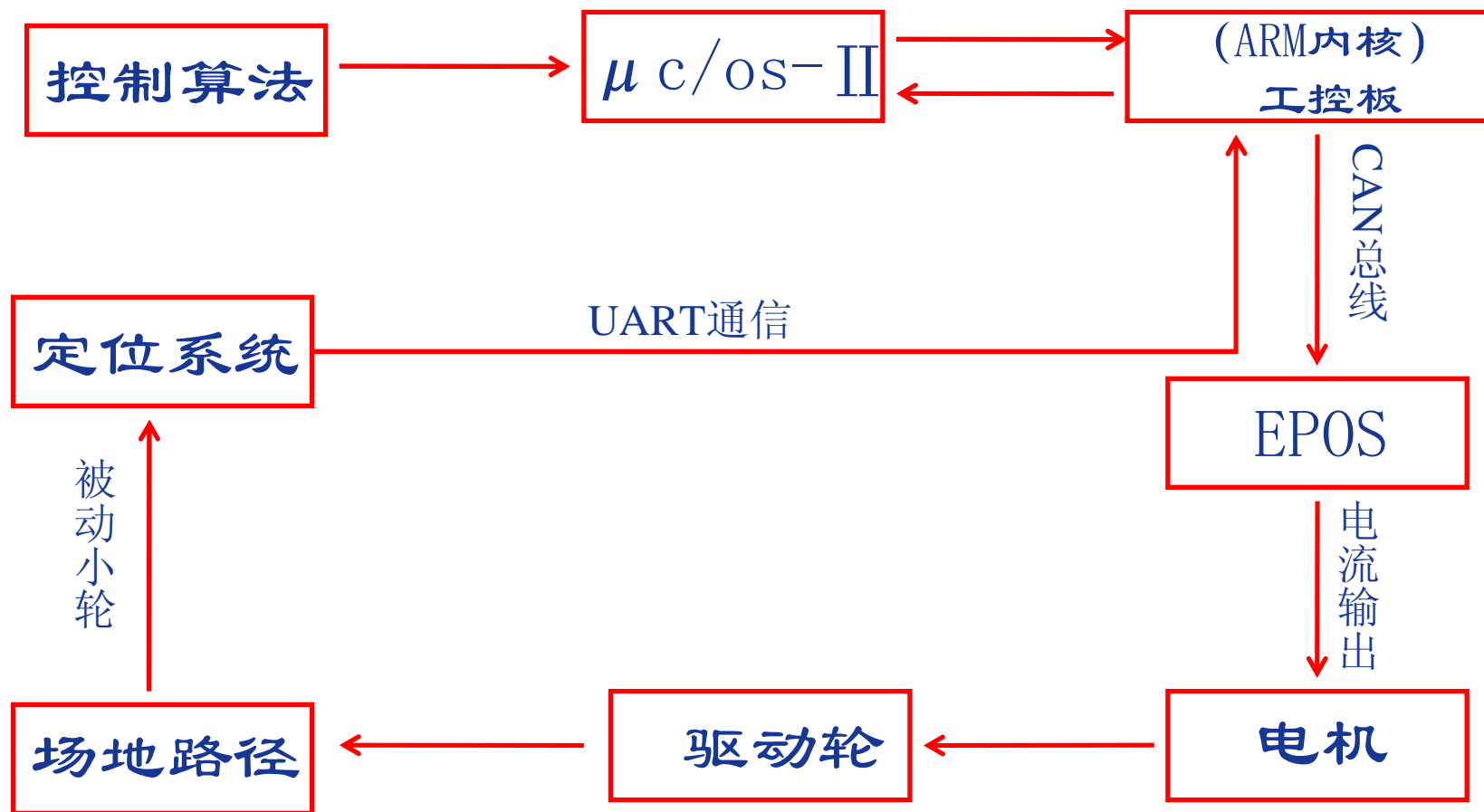
设 $u(k) = u(k-1) + k_p * \Delta e(k) + k_i * e(k) + k_d * (\Delta e(k) - \Delta e(k-1))$;

其中 $u(k)$ 为输出控制量， k_p 为比例系数， k_i 为积分系数， k_d 为微分系数；

Programming

在机器人控制过程中，要将机器人从任意位置任意方向驱动到目标点，我们采用的是首先将机器人以较大的转向速度和较小的前进速度，转向正对目标点，再以较大的前进速度和较小的转向速度到达目标点的可接受范围。在点镇定的前段路径，我们以反馈算法控制，在点镇定的后段路径，我们以强PID减弱反馈的算法进行控制。其中PID的控制具体行为如下：给左右两轮一恒定的线速度 v ，根据在运动过程中的机器人坐标，计算出误差，通过PID计算出控制量 $u(k)$ ，控制机器人转向角速度 w ，使机器人能够在有干扰的情况下沿预定直线运动到目标点。

Programming





Thank You!