

# 工程越野主题机器人创新教学与实践教程

## 目录

第一章 零件的认知与使用·····	1
-------------------	---

一、零部件的清点	1
二、机械零件认知	2
三、刚体结构连接	9
四、可动结构连接	10
第二章 控制的基础知识	11
一、主控板简介	11
二、Bigfish 扩展板简介	17
三、安装编程环境-Basra	19
第三章 底盘结构	23
使用三维软件辅助组装—STP Viewer	23
1.认识 STP Viewer	23
2.安装 STPViewer	23
3.浏览文件	23
第一课 驱动轮	26
第二课 差速底盘（一）	30
第三课 差速底盘（二）	32
第四课 转向机构	34
第五课 悬挂机构	37
第六课 履带机构	40
第四章 自主控制	42
第七课 传感器采值	42
第八课 条件语句	46

第九课 条件嵌套语句.....	53
第十课：有限状态机.....	60
第五章 实战.....	75
第十一课 窄桥.....	75
第十二课 管道.....	78
第十三课 台阶.....	80
第十四课 栅格.....	82
第十五课 通过全程.....	84
第十六课 全地形比赛.....	94

# 第一章 零件的认知与使用

## 一、零部件的清点

请大家阅读设备里附带的装箱单，清点设备里的零部件，在清点的过程中，我们会认识

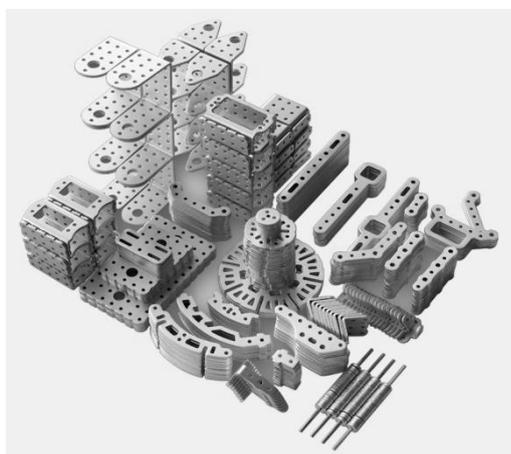
这些零部件的名字、外观、位置、大致用途，为后面的内容做好准备。

产品名称: Rino-AT-1356配置清单										
 A03 三爪自紧输出头 2	 A04 30齿齿轮 1	 A04a 传动齿轮 2	 A05 输出头 2	 A06 高转速机输出头 2	 D03 冲管轴 100	 A15 双轴承 2	 A16 轴承片 80	 A19 碎屑齿轮 4	 A17 1:10锥齿轮 2	 J01 10mm滑动 4
 J02 3x5双折支撑 2	 J03 5x7折支撑 4	 J04 7x11折支撑 2	 J05 90度弯角 2	 J06 传动轴 1	 J07 大垫片 2	 J08 垫片10 10	 J09 垫片20 2	 J10 舵机双折角 2	 J12 机械垫40mm 2	 J13 机械垫双折 2
 J14 机械垫 4	 J17 输出弯角 6	 J19 双足脚 6	 J20 双足支撑 4	 J23 双足支撑 2	 J25 小垫片 12	 J26 三爪弯角 2	 J29 球支撑片 16	 J31 高转速机弯角 6	 D01 小垫片 50	 P19 半圆万向轮 1
 F306 螺丝6mm 20	 F308 螺丝8mm 20	 F310 螺丝10mm 20	 F316 螺丝16mm 10	 F320 螺丝20mm 20	 F325 螺丝25mm 20	 F340 螺丝40mm 10	 F206 舵机螺丝 4	 F2510H 高转速螺丝 10	 M3 螺母 100	 Z10 螺母10mm 20
 Z15 螺母15mm 20	 Z20 螺母20mm 10	 Z30 螺母30mm 4	 T027 轴套2.7mm 20	 T054 轴套5.3mm 10	 T104 轴套10.4mm 10	 T154 轴套15.4mm 10	 M01 标准倒角电机 2	 M06 双轴高转速电机 6	 P10 排线 4芯输入 4	 P15 1排2高转速排线 2
 P25 滤芯杆 20	 P25 USB线 1	 P02 充电器 1	 P03 锂电池 1	 P01 螺丝 10	 P06 螺丝刀 1	 P07 内六角螺丝 1	 P13 镊子 1	 U01 轴承传感器 4	 C02 转速传感器 2	 C26 红外传感器 4
 C33 灰度传感器 4	 C34 声音识别传感器 1	 C42 Bearr控制板 1	 C54 Jieflash2.1板 1	 光盘 1	 教材 1					

## 二、机械零件认知

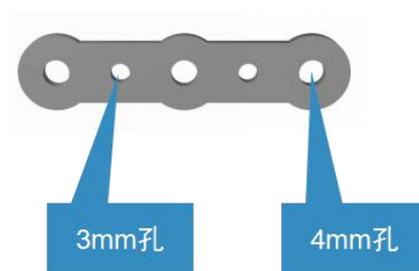
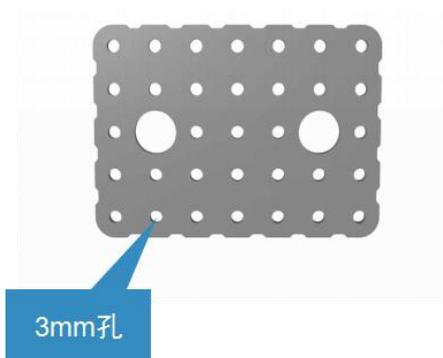
“探索者”的主要零件采用铝镁合金材质，零件上开有通孔，用不锈钢螺丝螺母组装，如下图所示。孔的作用是约束零件。常见的零件孔有直径 3mm 的孔和 4mm 孔，常见孔的中心距为 10mm。“探索者”的零件孔可看做几何中的“点”，是空间几何造型的基本单位。另外，还有提供特定机械运动的各种方孔、大方孔、长圆孔、大圆孔、异型孔等。

丰富的扩展孔是“探索者”零件体系的一大特征，使整个零件体系具有极高的扩展性。



### 1.零件孔

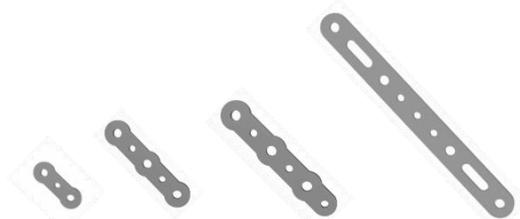
零件孔提供了“点”单位。最常用的零件孔是“3mm 孔”和“4mm 孔”，通过紧固件（螺丝、螺母等）可以将零件组装在一起。



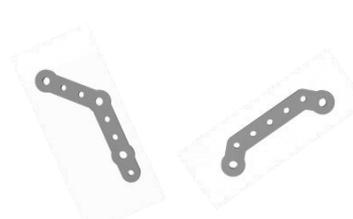
### 2.连杆类零件

连杆类零件提供了“线”单位。连杆类零件可用于组成平面连杆机构或空间连杆机构。

杆与杆相连可以组成更长的杆，或构成桁架。



例：四种长度不同的杆件。从左至右依次为：“机械手20”、“机械手40”、“机械手40驱动”、“双足支杆”



例：两种带角度的杆件，可用于需要角度变化的结构。从左至右依次为：“机械手指”、“双足连杆”

### 3. 平板类零件

这类零件适合做为“面”单位参与组装，从而底板、立板、背板、基座、台面、盘面等。

同时平板与平板之间的连接可以组成更大的“面”，或者不同层次的“面”。



例：两种矩形平板件，从左至右依次为：“5X7孔平板”、“7X11孔平板”，可用作底板、背板、台面等搭载平台



例：两种圆形平板件，从左至右依次为：“小轮”、“大轮”。可用作轮子、履带轮、滚筒的圆面、半球结构圆面、球结构圆面等。

### 4. 框架类零件

框架类零件的参与，使线和面可以连接成“体”。框架类零件多用于转接，连接不同的“面”零件和“线零件”，组成框架、外壳等。框架零件本身是钣金折弯件，有一定的立体特性，甚至可以独立成“体”。



90度支架



输出支架



3x5折弯

三种折弯件，可搭建机构支架，连接不同平面。

### 5. 辅助类零件

辅助类零件是通用性较弱，而专用性较强的零件。前面讲过的连杆、平面、框架类零件

的通用性极强，可以执行“像素”式的组合，而辅助类零件的用途往往比较单一。它们虽然也开有很多的扩展孔，在某些时候也可以用在其他地方，但是，适用范围却小很多。他们的存在，弥补了通用零件“泛而不精”的组装特性，可以大大降低某些机构的组装难度。从一定意义上来说，只要通用型零件足够多，所有的结构都可以实现，只是大与小、多与少的区别。但是受制于成本和组装难度，我们无法使用“足够多”的零件去组装，所以我们需要一些特定用途的零件。

### 常规传动零件

常规传动零件以齿轮为代表，提供常见的传动机构的元件，它们基本没有通用性，但是某些特殊机构必须用到。



30齿齿轮



随动齿轮



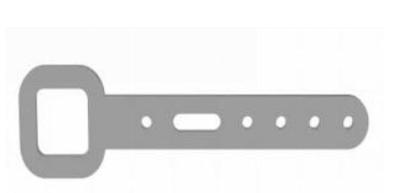
两种偏心轮，可组装偏心轮机构，代替凸轮，代替曲柄等。



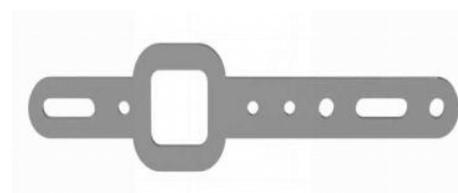
传动轴

### 偏心轮连杆

偏心轮连杆专门用于和偏心轮组合的连杆，在实际组装中，连杆件组成的曲柄摇杆结构可以替代偏心轮，但是使用偏心轮可以避免死点问题。



四足连杆



双足腿

### 曲柄滑块机构的主要零件，可用于搭建机器人行走机构

### 电机相关零件

电机相关零件，电机周边的辅助零件包括电机支架、输出头和 U 型支架等。



直流马达支架



马达支架



大舵机支架



大舵机U型支架

舵机双折弯

### 轮胎相关零件

轮胎需要联轴器才能和电机的输出头相连，如图所示。



### 标准五金件

“探索者”所用连接件如螺丝、螺母等均为标准五金零件，而且与其他标准五金零件的兼容度非常高，在使用中可以自己购买各种φ3接口的五金零件，将它们搭配在一起使用。



## 6.空间关系

“探索者”零件的中心孔距是 10mm，而壁厚基本都是 2.5mm，这个数据意味着“中心孔距=壁厚×4”。如图 2.10 所示。

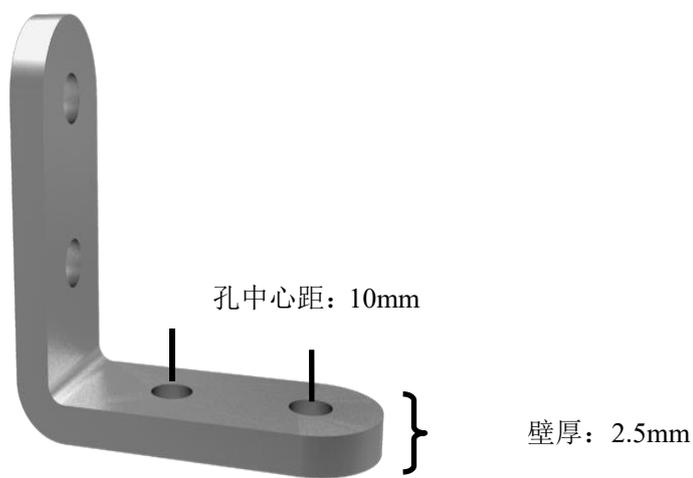


图 2.10 零件的壁厚与孔距

也就是说，四个零件叠加的厚度，正好等于两个孔的中心距。如图 2.11。

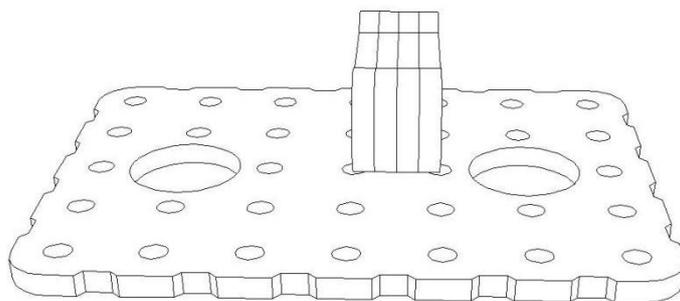


图 2.11 零件空间关系示意图

可以看出，“探索者”系统的最小组装单位是 2.5mm，常用组装单位是 10mm。

在这个原则之下，不同类型零件的孔总是可以良好匹配（图 2.12）。

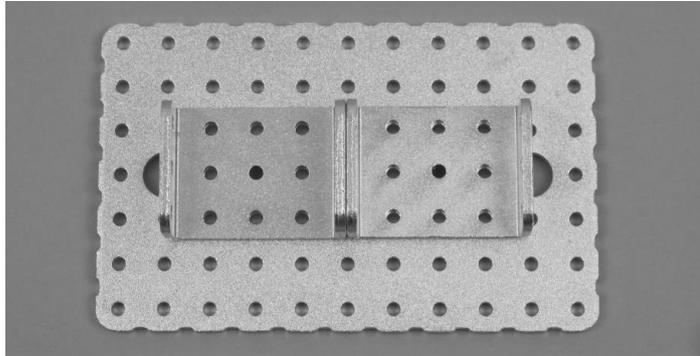


图 2.12 零件孔位置配合关系

齿轮等辅助性零件也遵循这个规律而设计，如图 2.13。

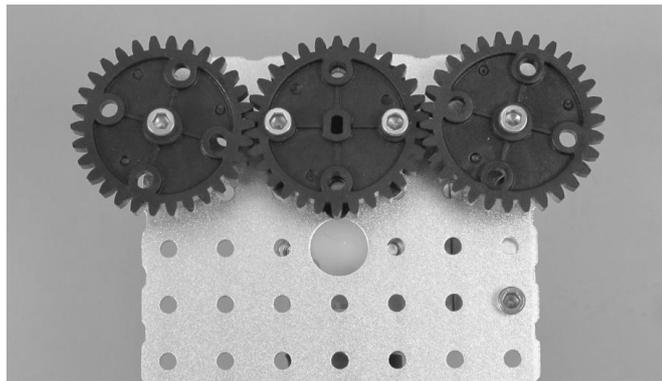


图 2.13 齿轮位置配合关系

对电机及其相关部件同样适用，如图 2.14。

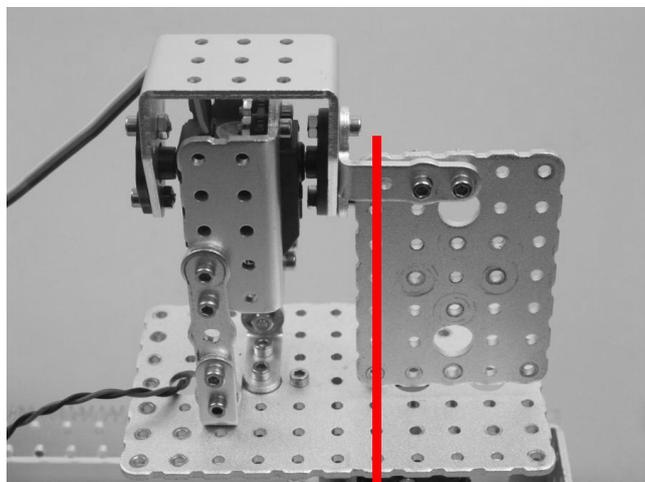
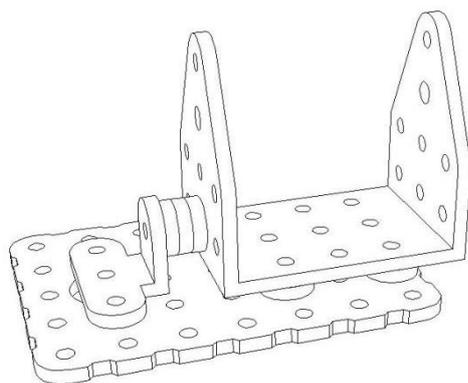
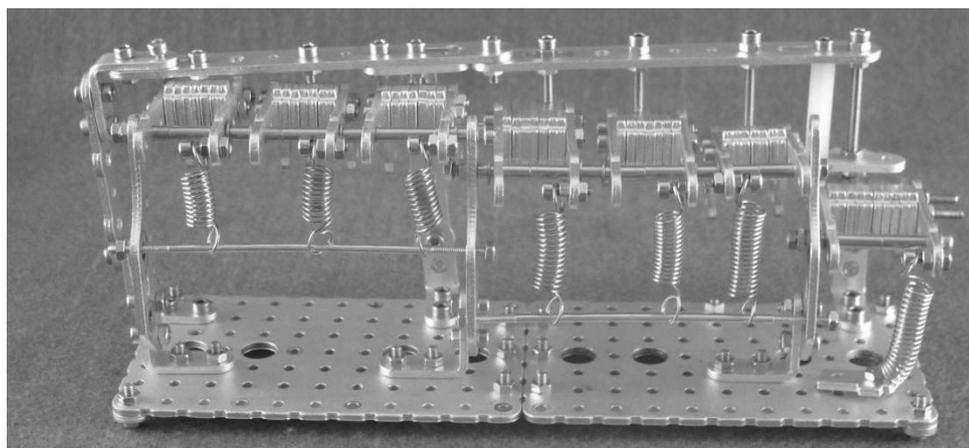


图 2.14 电机位置配合关系

这个原则保证了“探索者”零件系统在空间造型和传动设计的工作中，无论构型多么复杂，总能够良好匹配。因此，如果遇到两个零件孔无法匹配的状况时，我们只需要利用零件、螺母、螺柱、轴套等补充出 2.5mm 整数倍的距离，就可以匹配了。“探索者”零件匹配举例，如图 2.15 所示。



(a)



(b)

图 2.15 “探索者”零件匹配举例

如果在设计中遇到了实在不容易匹配的部件，我们可以借助一些带有可调节孔的零件来做转接（图 2.16）。

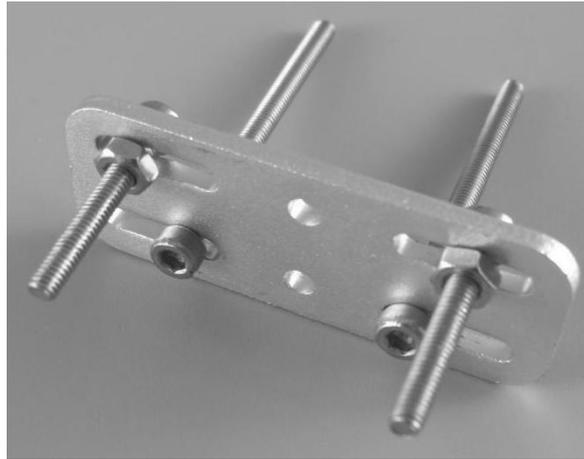


图 2.16 借助可调节孔的零件做转接

“探索者”的核心零件，即连杆类、平板类和框架类三种，多年来有增有减，但数量始终保持在 30 种左右。也就是说，与其他积木类创新组件不同，“探索者”零件库其实很小，它采用的是类似“七巧板”的创新思路，可以用有限的、精选的、通用性极强的零件，搭建千变万化的结构。

与“七巧板”一样，其缺点是某些造型不够仿真，只能抽象示意，色彩也不够丰富，但是这种设计，却能准确的反映出一个模型的结构和运动原理，同时大大减轻了使用者的选择负担，使他们不会迷失于庞大的零件海洋。经过多年的验证，“探索者”零件系统几乎可以搭建所有的常见机器人结构和传动模型，对于某些需要特殊零部件的机构，也完全可以搭建出替代机构，从而模拟其运动。

### 三、刚体结构连接

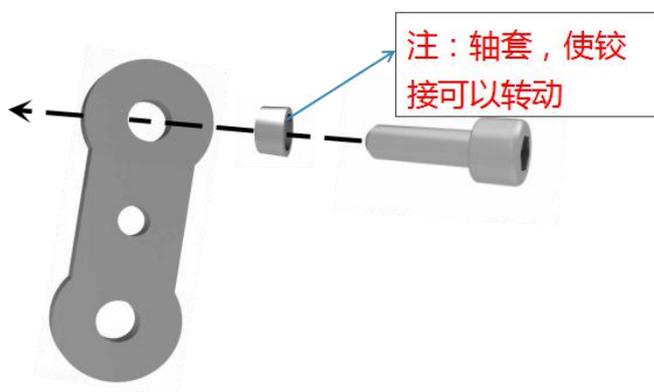
这里说的刚体指拿组件组装的一些连接点固定的造型，如平面、组合型平面、平台、组合型平台、框架、外壳造型等。

最基础的刚体组装至少需要 2 颗螺丝。这对应了“经过两点有一条直线,并且只有一条直线”的几何定理。刚体连接一般利用 3mm 零件孔。



#### 四、可动结构连接

可动结构，相对刚体而言，指带有铰接的结构，如轴、连杆组、滑块、不带电机的传动构造等。最基础的可动模型是铰链结构，利用 4mm 零件孔，并利用轴套起到轴承的作用，使铰链可以转动。



## 第二章 控制的基础知识

### 一、主控板简介

#### 1. Basra 主控板（本教材大部分实验例程基于 Basra 编写）

##### 概述

Basra 是一款基于 Arduino 开源方案设计的一款开发板，通过各种各样的传感器来感知环境，通过控制灯光、马达和其他的装置来反馈、影响环境。板子上的微控制器可以在 Arduino、eclipse、Visual Studio 等 IDE 中通过 c/c++ 语言来编写程序，编译成二进制文件，烧录进微控制器。Basra 的处理器核心是 ATmega328，同时具有 14 路数字输入/输出口（其中 6 路可作为 PWM 输出），6 路模拟输入，一个 16MHz 晶体振荡器，一个 USB 口，一个电源插座，一个 ICSP header 和一个复位按钮。

主 CPU 采用 AVR ATMEGA328 型控制芯片，支持 C 语言编程方式；该系统的硬件电路包括：电源电路、串口通信电路、MCU 基本电路、烧写接口、显示模块、AD/DA 转换模块、输入模块、IIC 存储模块等其他电路模块电路。控制板尺寸不超过 60\*60mm，便于安装。CPU 硬件软件全部开放，除能完成对小车控制外，还能使用本实验板完成单片机所有基础实验。供电范围宽泛，支持 5v~9v 的电压，干电池或锂电池都适用。编程器集成在控制板上，通过 USB 大小口的方式与电脑连接。下载程序。开放全部底层源代码。控制板含 3A6V 的稳压芯片，可为舵机提供 6v 额定电压。板载 8\*8led 模块采用 MAX7219 驱动芯片。板载 2 片直流电机驱动芯片 L9170，可驱动两个直流电机。板载 USB 驱动芯片及自动复位电路，烧录程序时无需手动复位。2 个 2\*5 的杜邦座扩展坞，方便无线模块、OLED、蓝牙等扩展模块直插连接，无需额外接线。

## 特点

- 开放源代码的电路图设计，程序开发接口免费下载，也可依需求自己修改。
- 可以采用 USB 接口供电，不需外接电源，也可以使用外部 DC 输入。
- 支持 ISP 在线烧，可以将新的“bootloader”固件烧入芯片。有了 bootloader 之后，可以在线更新固件。
- 支持多种互动程序，如：Flash、Max/Msp、VWV、PD、C、Processing 等。
- 具有宽泛的供电范围，电源电压可任选 3v~12v 的电源
- 采用堆叠设计，可任意扩展
- 主控板尺寸不超过 60mm\*60mm，便于给小型机电设备安装
- 板载 USB 驱动芯片及自动复位电路，烧录程序时无需手动复位

## 参数

- 处理器 ATmega328
- 工作电压 5V
- 输入电压（推荐） 7-12V
- 输入电压（范围） 6-20V
- 数字 IO 脚 14 (其中 6 路作为 PWM 输出)
- 模拟输入脚 6
- IO 脚直流电流 40 mA
- 3.3V 脚直流电流 50 mA
- Flash Memory 32 KB (ATmega328, 其中 0.5 KB 用于 bootloader)
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- 工作时钟 16 MHz

## 电源

Basra 可以通过 3 种方式供电，而且能自动选择供电方式

- 外部直流电源通过电源插座供电。
- 电池连接电源连接器的 GND 和 VIN 引脚。
- USB 接口直接供电。

电源引脚说明

VIN --- 当外部直流电源接入电源插座时，可以通过 VIN 向外部供电；也可以通过此

引脚向 UNO 直接供电；VIN 有电时将忽略从 USB 或者其他引脚接入的电源。

5V --- 通过稳压器或 USB 的 5V 电压，为 UNO 上的 5V 芯片供电。

3.3V --- 通过稳压器产生的 3.3V 电压，最大驱动电流 50mA。

GND --- 地脚。

## 存储器

ATmega328 包括了片上 32KB Flash，其中 0.5KB 用于 Bootloader。同时还有 2KB SRAM 和 1KB EEPROM。

## 输入输出

14 路数字输入输出口：工作电压为 5V，每一路能输出和接入最大电流为 40mA。每一路配置了 20-50K 欧姆内部上拉电阻（默认不连接）。除此之外，有些引脚有特定的功能

- 串口信号 RX (0 号)、TX (1 号)：与内部 ATmega8U2 USB-to-TTL 芯片相连，提供 TTL 电压水平的串口接收信号。
- 外部中断 (2 号和 3 号)：触发中断引脚，可设成上升沿、下降沿或同时触发。
- 脉冲宽度调制 PWM (3、5、6、9、10、11)：提供 6 路 8 位 PWM 输出。
- SPI (10(SS), 11(MOSI), 12(MISO), 13(SCK))：SPI 通信接口。
- LED (13 号)：Arduino 专门用于测试 LED 的保留接口，输出为高时点亮 LED，反之输出为低时 LED 熄灭。

6 路模拟输入 A0 到 A5：每一路具有 10 位的分辨率（即输入有 1024 个不同值），默认输入信号范围为 0 到 5V，可以通过 AREF 调整输入上限。除此之外，有些引脚有特定功能

- TWI 接口 (SDA A4 和 SCL A5)：支持通信接口（兼容 I2C 总线）。

AREF：模拟输入信号的参考电压。

Reset：信号为低时复位单片机芯片。

## 通信

- 串口：ATmega328 内置的 UART 可以通过数字口 0 (RX) 和 1 (TX) 与外部实现串口通信；ATmega16U2 可以访问数字口实现 USB 上的虚拟串口。
- TWI (兼容 I2C) 接口
- SPI 接口

## 下载程序

Basra 上的 ATmega328 已经预置了 bootloader 程序，因此可以通过 Arduino 软件直

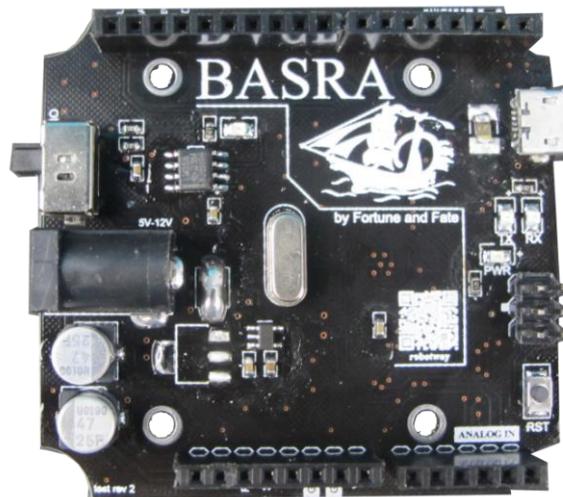
接下载程序到主控板中。

可以直接通过主控板上 ICSP header 直接下载程序到 ATmega328。

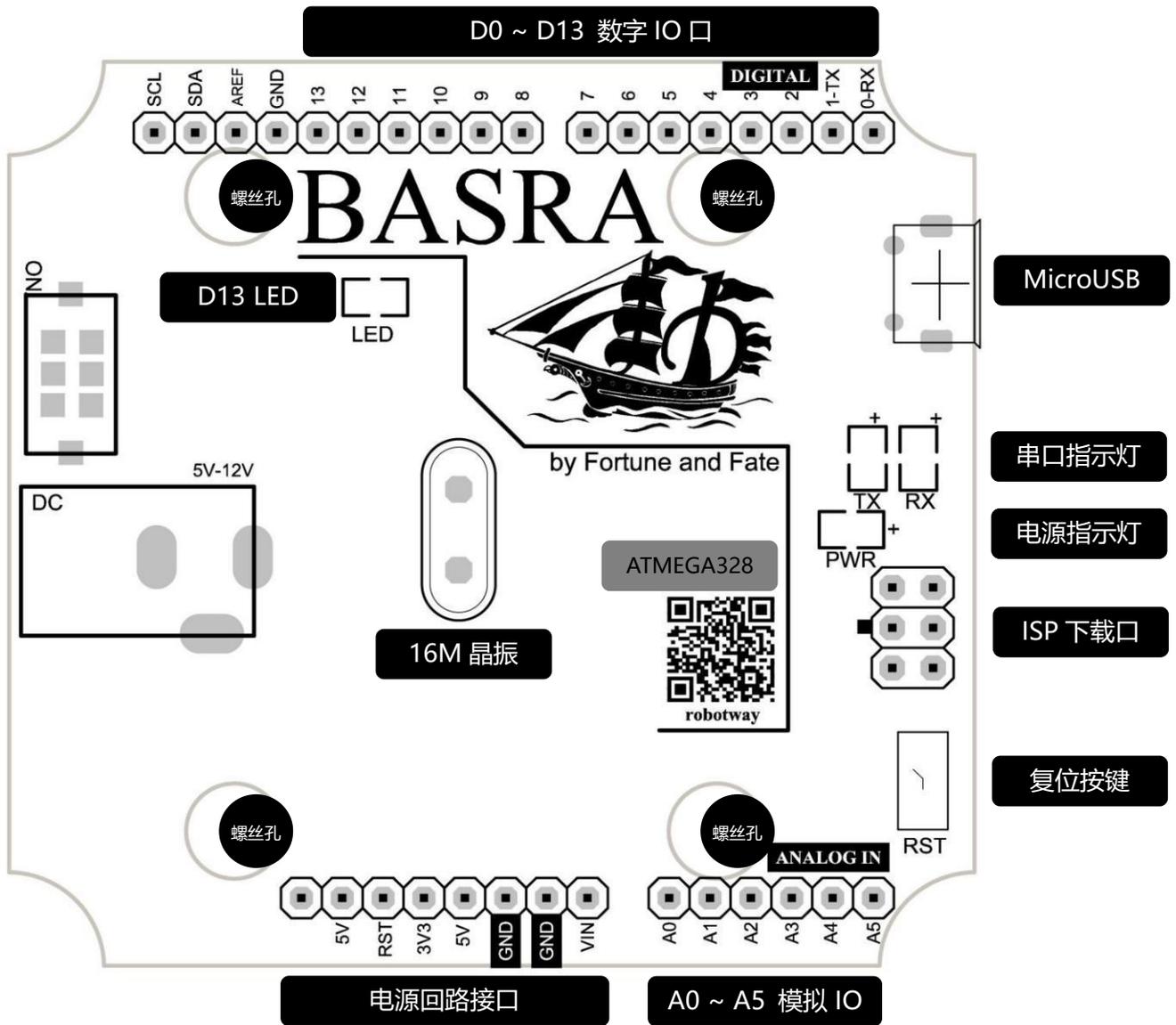
### 注意事项

- USB 口附近有一个可重置的保险丝，对电路起到保护作用。当电流超过 500mA 时会断开 USB 连接。
- 主控板提供了自动复位设计，可以通过主机复位。这样通过 Arduino 软件下载程序到主控板中时，软件可以自动复位，不需要在复位按钮。

### 实物图片与接口



D0 ~ D13 数字 IO 口



电源开关

电源插头

MicroUSB

串口指示灯

电源指示灯

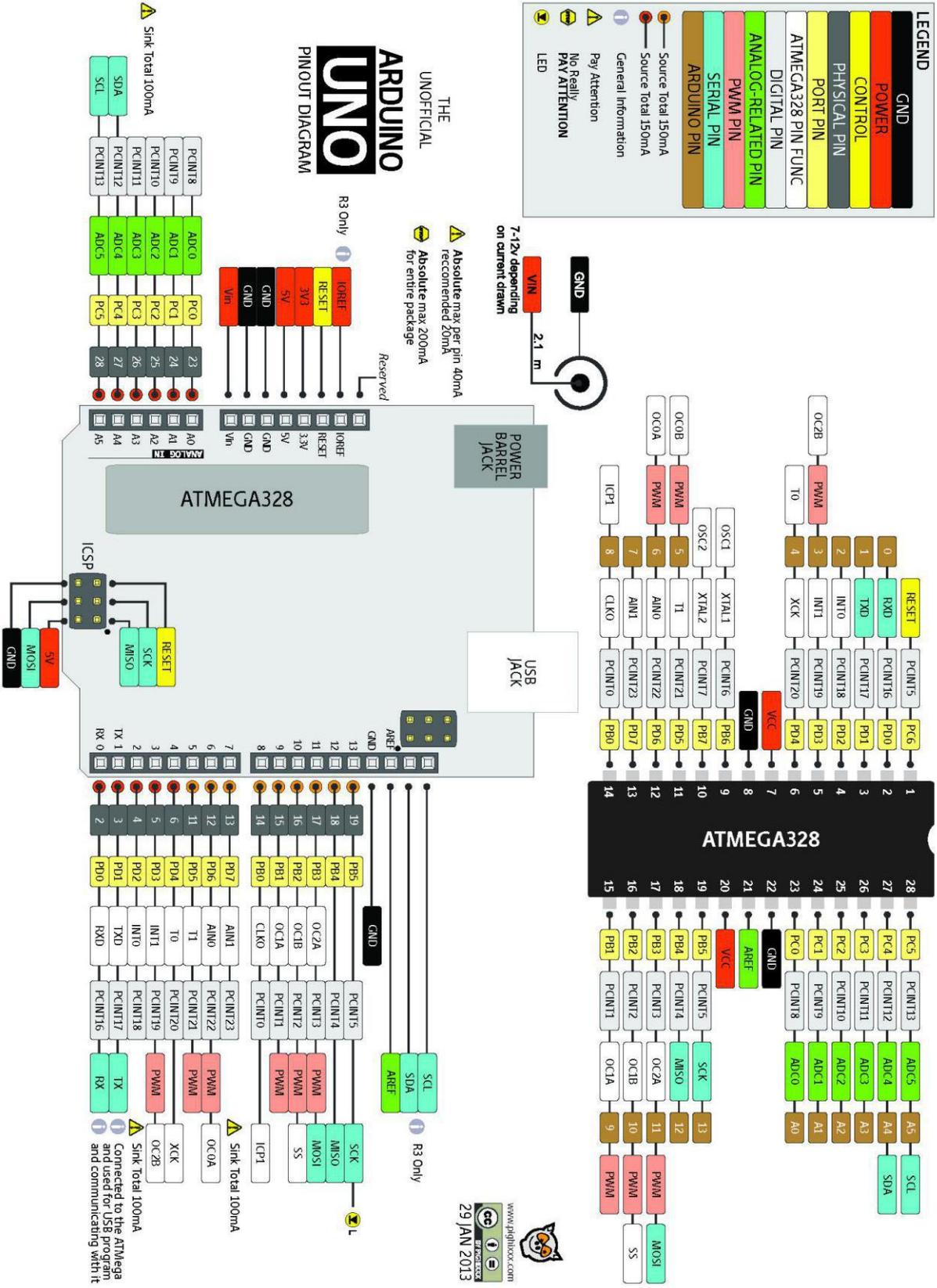
ISP 下载口

复位按键

电源回路接口

A0 ~ A5 模拟 IO

# 芯片与接口引脚



## 二、Bigfish 扩展板简介

通过 BigFish 扩展板连接的电路可靠稳定，上面还扩展了伺服电机接口、8\*8Led 点阵、直流电机驱动以及一个通用扩展接口，可以说控制板的必备配件。

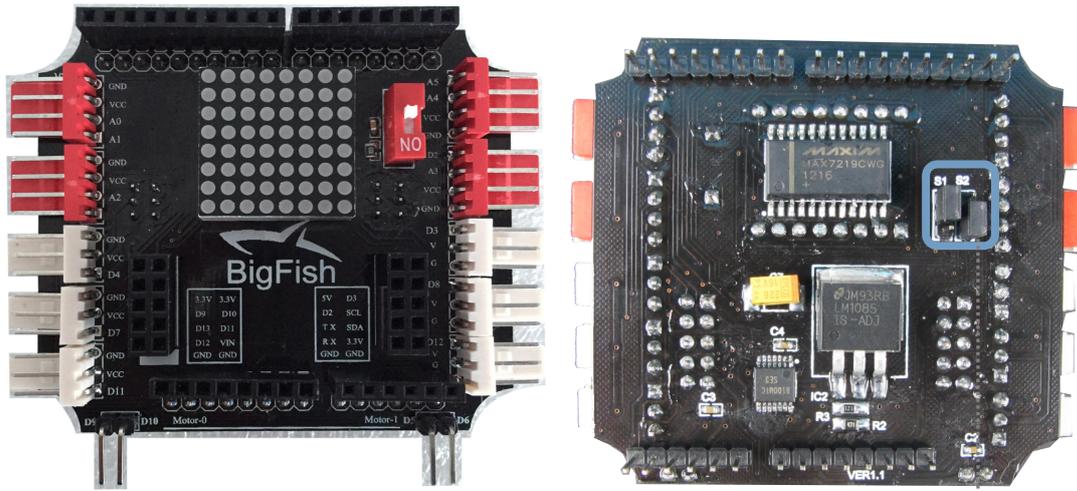
### 特点

- (1) 完全兼容 Basra、Mehran 控制板接口
  - (2) 彩色分组插针，一目了然
  - (3) 全部铜制插针，用料考究，电器性能稳定
  - (4) 优秀 PCB 设计，美观大方
  - (5) 多种特殊接口设计，兼容所有探索者电子模块，使用方便
- 所有 3P、4P 接口采用防反插设计，避免电子模块间连线造成的误操作
  - 板载舵机接口、直流电机驱动芯片、MAX7219LED 驱动芯片，可直接驱动舵机、直流电机、数码管等机器人常规执行部件，无需外围电路
  - 具有 5v、3.3v 及 vin 3 种电源接口，便于为各类扩展模块供电

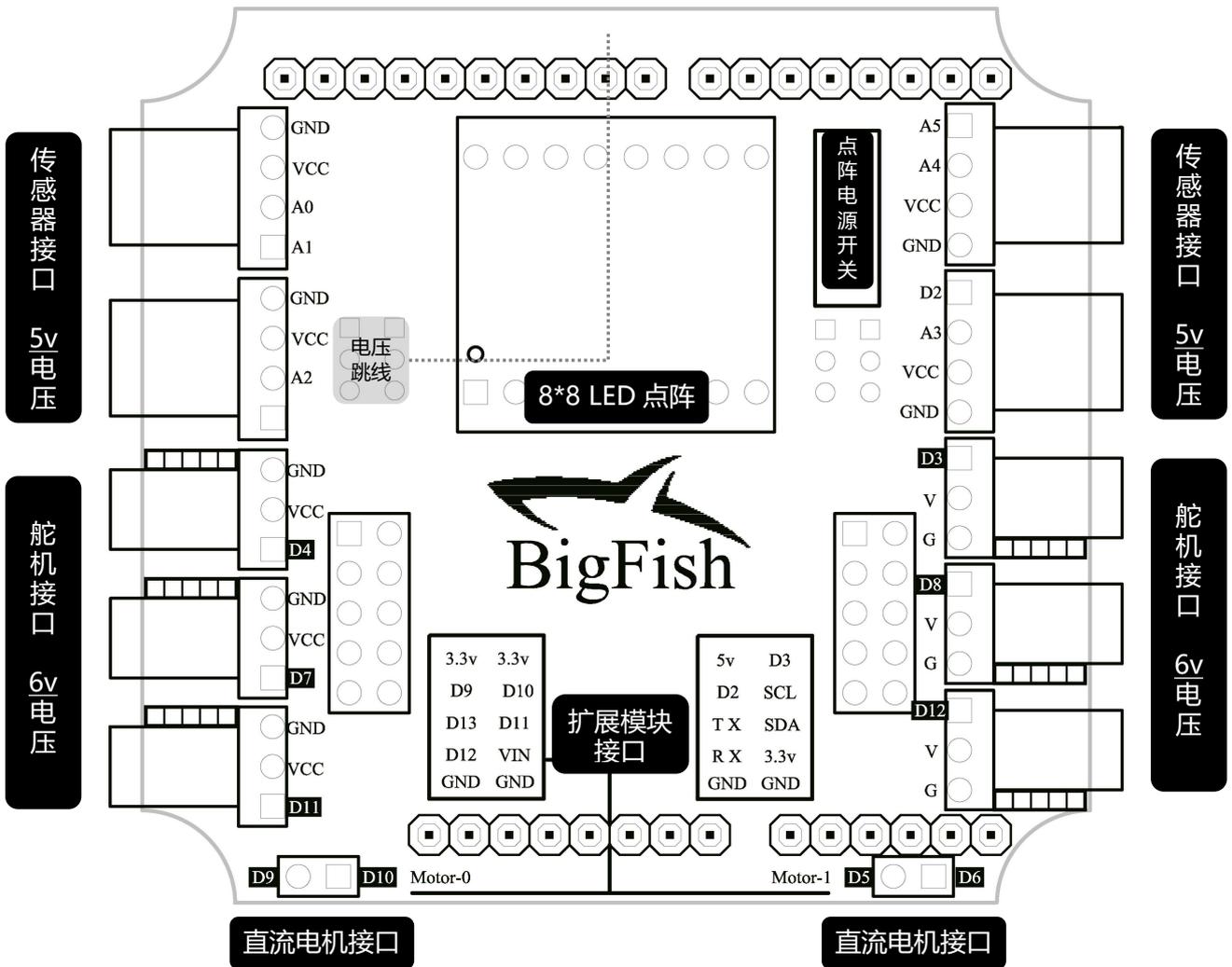
### 参数

- 4 针防反插接口供电 5v
- 舵机接口使用 3A 的稳压芯片 LM1085ADJ，为舵机提供 6v 额定电压
- 8\*8led 模块采用 MAX7219 驱动芯片
- 板载 2 片直流电机驱动芯片 L9170，支持 3v~15v 的 vin 电压，可驱动两个直流电机。
- 2 个 2\*5 的杜邦座扩展坞，方便无线模块、OLED、蓝牙等扩展模块直插连接，无需额外接线

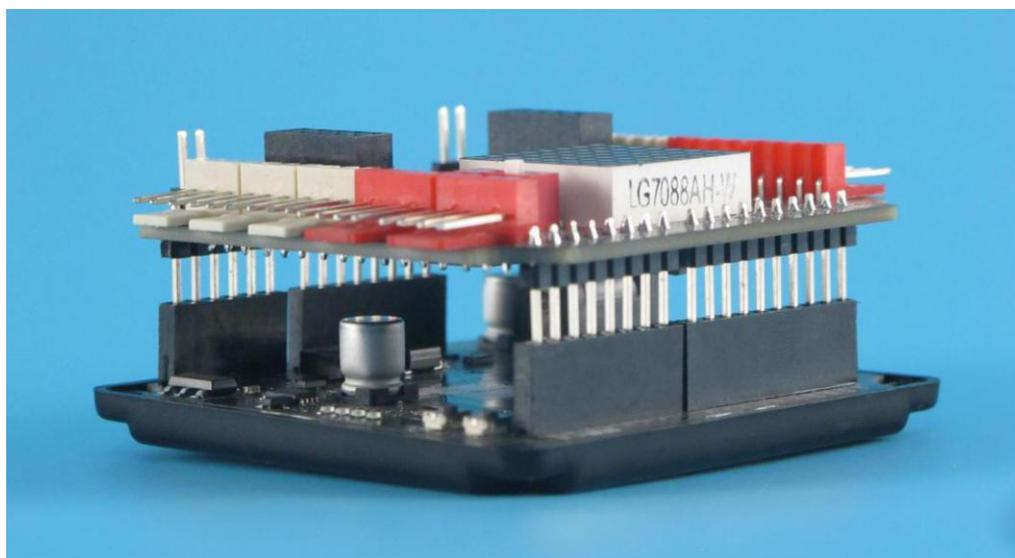
## 实物图片与接口



传感器 VCC	舵机 VCC
5V	5V
GND	GND
vin	6V



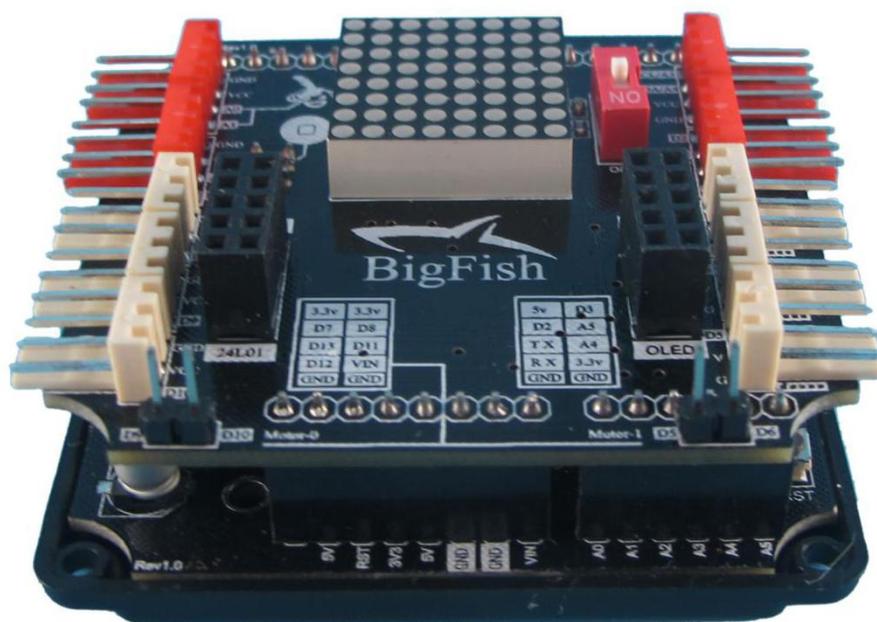
与控制板堆叠连接



BigFish 扩展板



控制板



### 三、安装编程环境-Basra

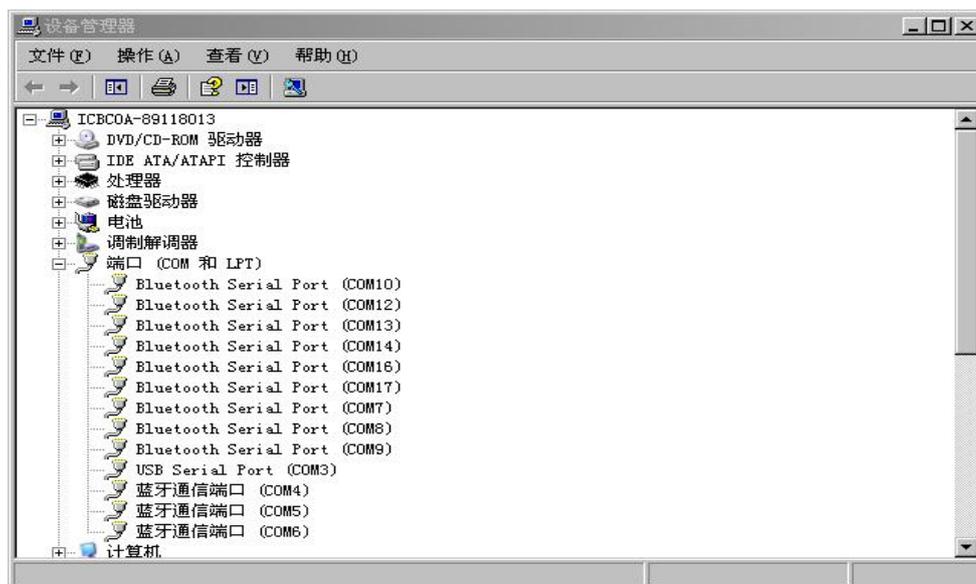
步骤一：拷贝..\Basra 控制板\arduino-1.5.2 目录至本机位置。

步骤二：将 Basra 控制板通过 miniUSB 数据线与 PC 连接，初次连接时会弹出驱动安

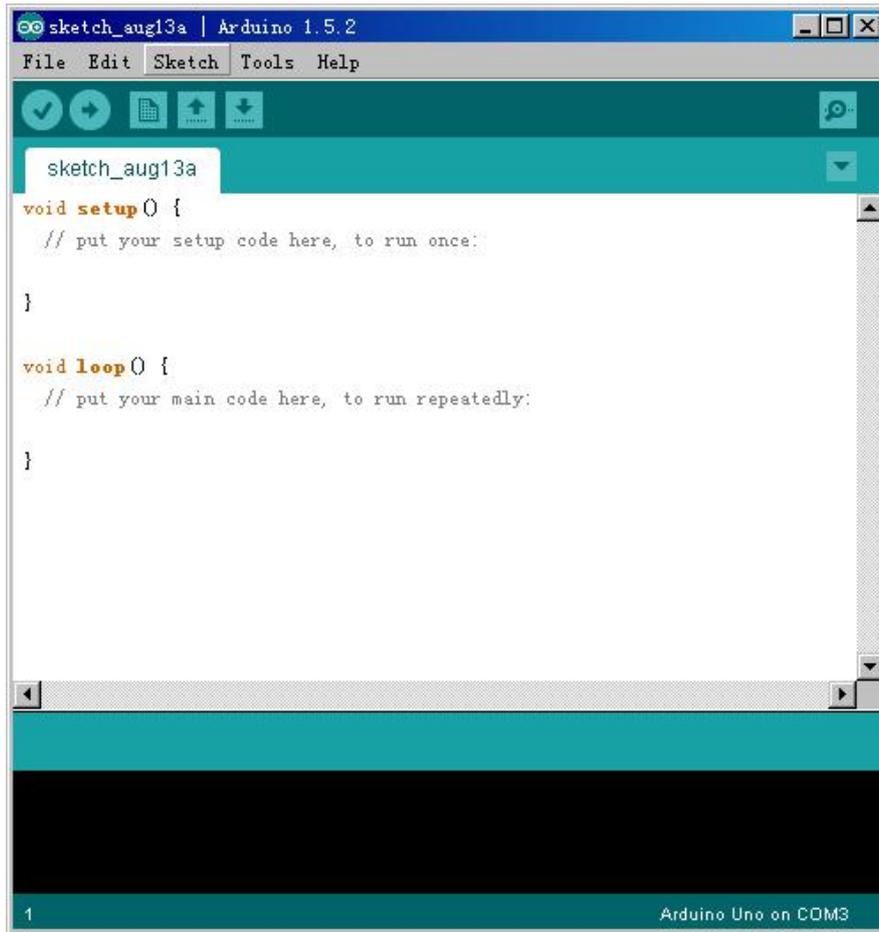
装提示。选择.\Basra 控制板\arduino-1.5.2\drivers\FTDI USB Drivers 目录安装驱动。



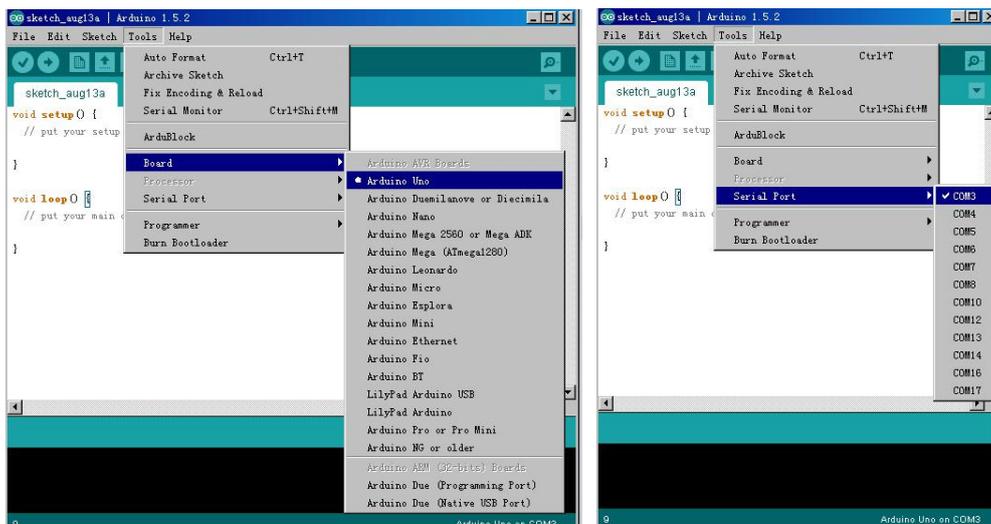
步骤三：打开设备管理器，在“端口(COM和LPT)”列表中，出现 USB Serial Port (COMx)，表示驱动安装成功。请记录下这个 COM 端口号 x，下图中端口号为 COM3。



步骤四：在本机上运行 arduino-1.5.2 目录下的 arduino.exe，显示如下界面：



步骤五：在 **Tools** 菜单下，依次选择 **Board** 里的 **Arduino Uno** 项，以及 **Serial Port** 里的 **COM3**（COM3 为步骤 3 里记录下的端口号）。此时在界面右下角显示 **Arduino Uno on COM3**。

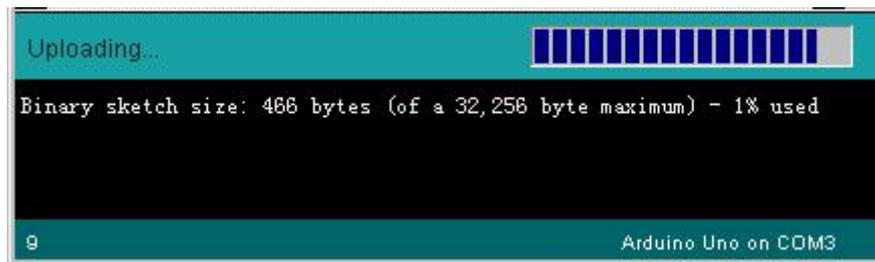


步骤六：点击 **upload** 按钮 ，一个空白的程序将自动烧录进 Basra 控制板。具体

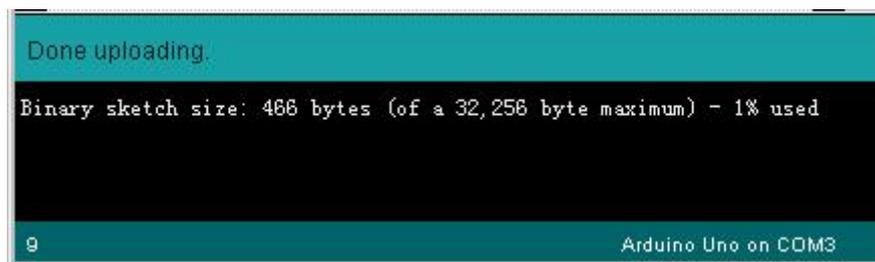
过程如下图所示：



1、开始编译代码



2、开始向 Basra 控制板烧录程序，烧录过程中控制板上的 TX/RX 指示灯闪动



3、烧录成功

## 第三章 底盘结构

### 使用三维软件辅助组装—STP Viewer

#### 1.认识 STP Viewer

STP 是一种通用的 3D 文件格式，可以在几乎所有的 3D 设计软件中打开。STP Viewer 是一款小体量的、针对 STP 格式文件的看图软件，可以打开和观看 STP 格式 3D 文件，方便参照 3D 图组装。

#### 2.安装 STPViewer

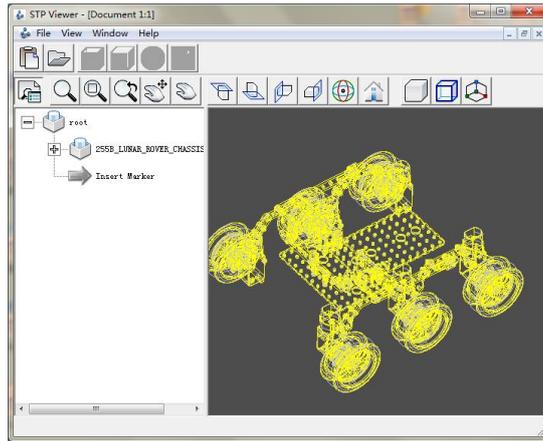
在光盘/软件套装目录下找到 STPViewer setup.exe 文件，双击安装 STPViewer



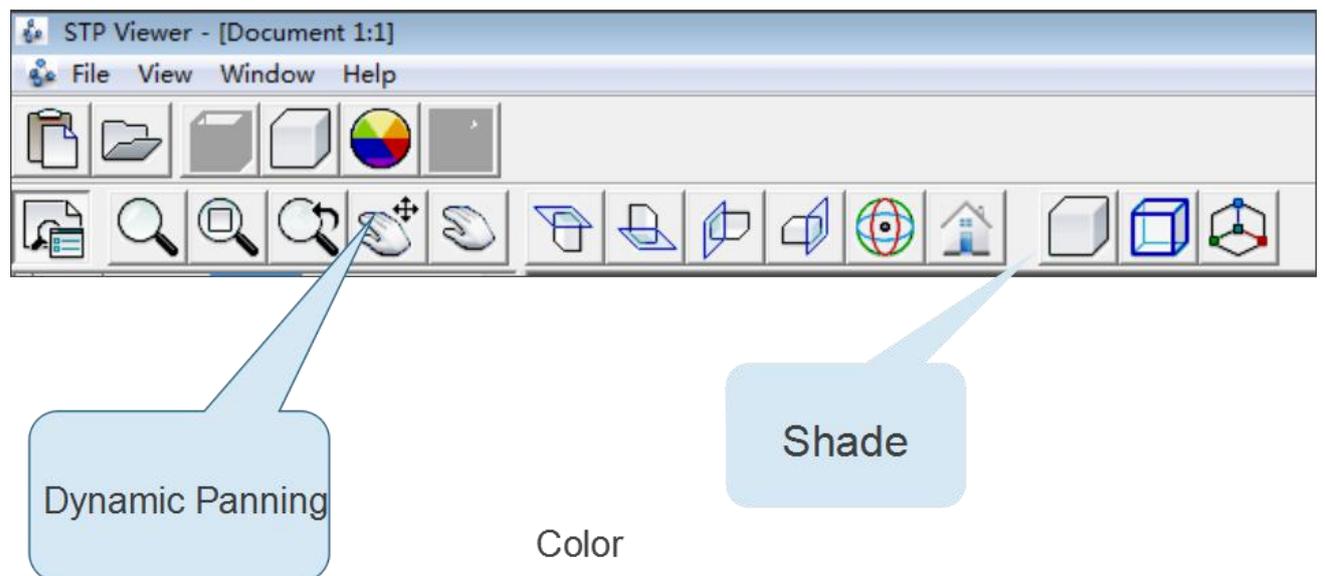
#### 3.浏览文件

在光盘/STP 目录下找到名为 “255c\_lunar\_rover\_chassis\_6m.stp” 的文件，将此文件拷贝至某盘根目录，双击打开（提示：STP 文件存储路径中不能有中文字符或特殊符号，否则软件无法读取。某些操作系统桌面也不识别。）。

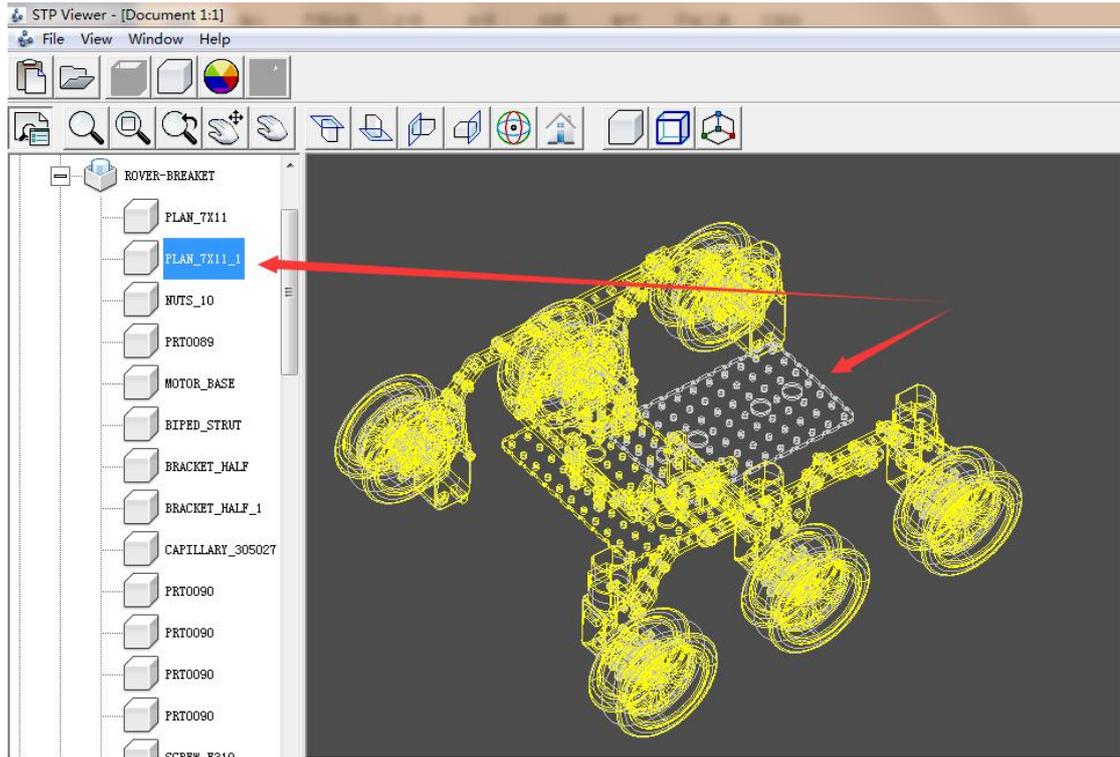
打开文件后，首先看到的是 3D 线框图



大部分时候,我们只需要实用 Dynamic Panning (动态规划,简单说就是平移)、Shade (塑形,简单说就是填充)、Color (着色) 等功能,而“旋转”和“缩放”功能用鼠标就可以实现。大家可以在界面上任意操作熟悉上面这些基本指令。



4.STPViewer 还有一个“隐藏零件”的功能:选中一个“零件”,然后可以看到在软件界面左侧零件树列表中,自动跳转到对应的零件名被选中。



在该零件树类别对应的零件名上点击右键，选择“hide”。对应的零件即可隐藏。有些结构由多个零件叠加组成，通过图片很难知道准确的零件组成，可以通过这个功能找到对应的零件。

## 第一课 驱动轮

**实验目的：**1. 通过组装驱动轮模块，熟悉部分“探索者”零部件，掌握“驱动轮”的基本组装技法、组装工具使用方法，以及 STPViewer 三维浏览软件的使用方法；2. 通过搭建控制电路，掌握“探索者”基本电路的连接方法；3. 通过控制驱动轮模块，掌握 Arduino 软件的基本使用方法，尝试图形化编程，C 语言编程；4. 认识 digitalWrite 函数，analogWrite 函数；5. 熟悉驱动轮模块 module003 的运动性能。

**实验性质：**验证型实验

**实验课时：**2 课时

**参考资料：**请参考**光盘**中的 STP 装配文件，000\_module003\_dw.stp，以及光盘**课件**“基础零件组装”部分相关 PPT，尤其是“**直流电机的安装**”和“**橡胶轮胎与联轴器的安装**”，以及“软件的安装与使用”部分的“**STPViewer 三维浏览软件的使用**”。你需要用到直流电机、直流电机支架、直流电机输出头、联轴器、1:10 模型轮胎、螺丝螺母，以及组装工具等。

你还需要用到光盘中的“**探索者**”**软件套装**文件夹中的软件。请参考课件中“基础电路连接”部分的“**Bigfish 连接直流电机**”、“软件的安装与使用”部分的“**Basra 在 Windows 环境下配置与烧录**”，连好部件，设置好软件。

**光盘中的 PPT 资料给出了非常详细的基础操作说明，请大家在课下预览，上课时方便查阅，在以后的实验项目中，我们只给出简单的参考资料提示，不再详细展开。**

**实验步骤：**

1. 组装一个驱动轮模块（编号：000\_module003\_dw）；

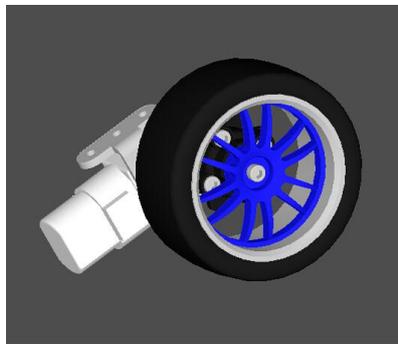


图 3.10 驱动轮模块 module003

**结构说明：**这种驱动轮模块原理非常简单，就是将 1:10 模型轮胎通过联轴器安装在 1:87 直流减速电机的输出头上，初次组装有一定难度。

**运动特性：**输出转矩，轮子转动的角速度、转动方向与电机一致，线速度一般，力量一般，摩擦力较好。

2. 将 Basra 主控板、Bigfish 扩展板、锂电池和驱动轮模块连接成电路；
3. 在图形化编程界面 Ardublock 中编写以下程序并烧录；



图 3.12 驱动轮供电转动写法 1

当直流电机连在 D9/D10 针脚（Bigfish 下方左侧的直流接口）时，可以通过把 D9 或 D10 置高来供电。

单向转动时，图 3.12 程序的写法和图 3.13 程序的写法是等价的。需要正反转时，只能使用图 3.13 的完整写法。



图 3.13 驱动轮供电转动写法 2

你还会发现，按过“上载到 Arduino”按钮之后，C 语言界面上自动生成了 C 语言代码。对应生成的 C 语言代码分别为：

```
void setup()
{
  pinMode( 9 , OUTPUT);
}

void loop()
{
  digitalWrite( 9 , HIGH );
}
```

和

```
void setup()
{
  pinMode( 9 , OUTPUT);
  pinMode( 10 , OUTPUT);
}

void loop()
{
  digitalWrite( 9 , HIGH );
  digitalWrite( 10 , LOW );
}
```

digitalWrite 有两个参数，很容易掌握，请对应图形程序观察、学习。

4. 请分别更改供电端口号为 10、5、6，观察驱动轮模块转动的情况，以顺时针或逆时针来记录；

5. 在图形化编程界面 Ardublock 中编写以下程序并烧录；



图 3.14 驱动轮转速的控制

生成的 C 语言代码为：

```
void setup()
{
  pinMode( 9, OUTPUT);
}

void loop()
{
  analogWrite(9 , 255);
}
```

在这种写法下，可以利用 analogWrite 函数，通过改变 PWM 占空比来改变电机的转动速度。analogWrite 函数通过 PWM 的方式在引脚上输出一个模拟量，较多的应用在 LED 亮度控制、电机转速控制等方面。analogWrite 有两个参数 pin 和 value，参数 pin 表示所要设置的引脚，只能选择函数支持的引脚；参数 value 表示 PWM 输出的占空比，范围在 0 ~ 255 的区间，对应的占空比为 0% ~ 100%。

请大家修改 value，观察模块运动的变化情况。在实际中，由于有负载，当 value 低于某值时，就已经带不动电机了，不需要 value 取 0

## 第二课 差速底盘（一）

**实验目的：**1. 组装通过组装 092e\_twowheel\_2m 机器人，熟悉轮型底盘的一些基础制作方法；2. 通过控制机器人运动，了解差速运动的特性和程序书写方法。

**实验性质：**设计型实验

**实验课时：**2 课时

**参考资料：**092e\_twowheel\_2m.stp；实验 1。

**实验步骤：**

1. 组装一个 092e 号机构，双轮加支点型底盘；

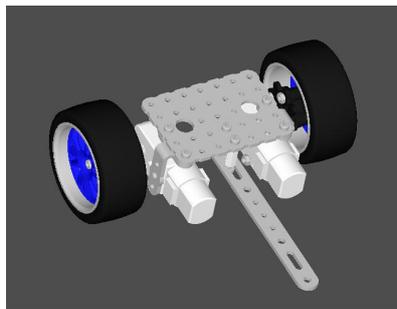


图 3.20 双轮加支点型机器人 092e

**结构说明：**这种机构由两个驱动轮模块构成，加长尾的目的是防止电机与地面摩擦，并提供一定的平衡性。可以说是最简单的具备完整运动能力的底盘。

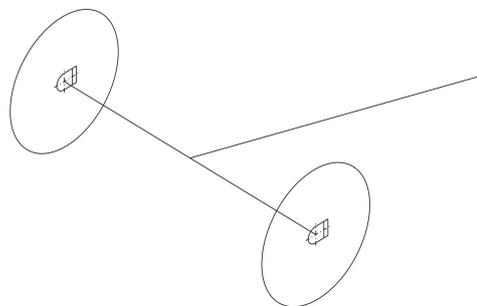


图 3.21 092e 的机构简图

**运动特性：**能完成前进、后退、左转、右转、原地旋转等动作，但底盘不水平，也不太稳定，前进并急刹时车尾容易翘起，后退时遇到地面坎坷容易被干扰，转动时采用的是差速转动。所谓“差速转动”，指主要依靠“两个轮子转动的**方向**和**速度**的各种搭配”完成各种动作。

关系如下图所示：

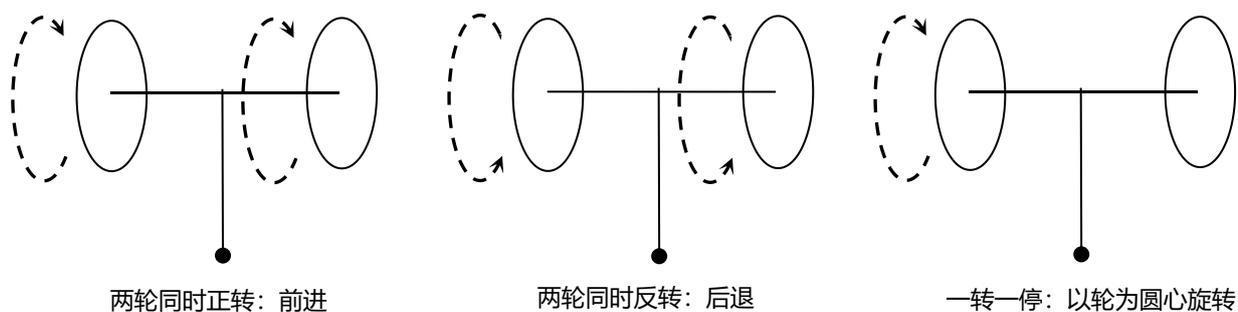


图 3.22 差速运动图解

2. 编程，实现机器人前进动作；
3. 编程，实现机器人后退动作；
4. 编程，实现机器人左转和右转动作；

更换一部分车身零件，或改变现有零件的组装方式，使机器人在同样的结构原理下达到不同的外观效果。

## 第三课 差速底盘（二）

**实验目的：**1. 熟悉四驱车型机器人的运动特性；2. 学会使用 1 拖 2 电机扩展线并联电机；

**实验性质：**验证型实验

**实验课时：**2 课时

**参考资料：**022\_4x4s car\_4m.stp

**实验步骤：**

1. 组装四驱车型机器人 022\_4x4s car\_4m；

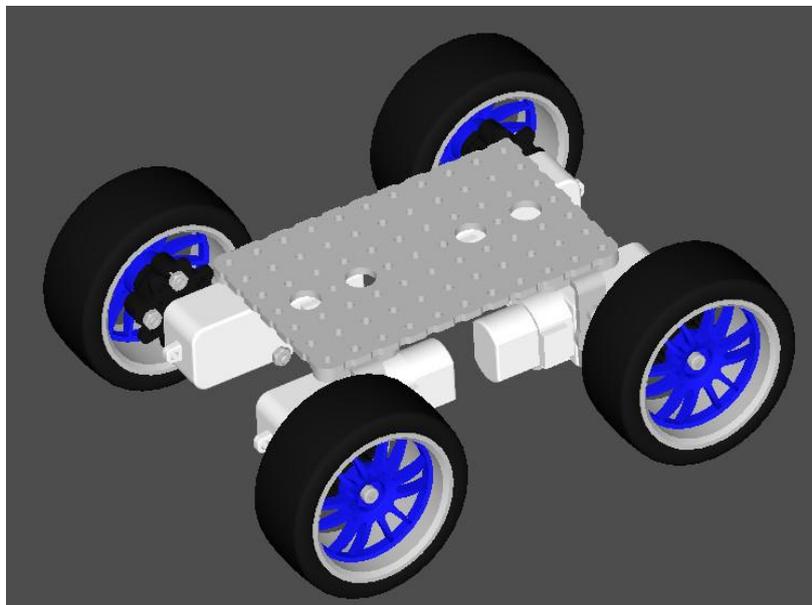


图 3.3 022 号机构——四驱车型机器人

**结构说明：**四驱车型机器人 022\_4x4s car\_4m 由四个驱动轮模块构成，其中有两个冗余的驱动轮，结构非常稳定，能提供更大的动力，从而在同样的负载下达到更高的速度。

**运动特性：**能完成前进、后退、左转、右转、原地旋转等动作。运动时同侧的电机必须同步转动，否则转速不同或方向相反会严重影响运动效果，因此，分别控制 4 个电机并不是一个好方案，可以将同侧驱动滚轮并联，用一个端口来控制两个驱动轮。

2. 用 2 根 1 拖 2 扩展线将驱动轮并联起来，并接上 Bigfish；



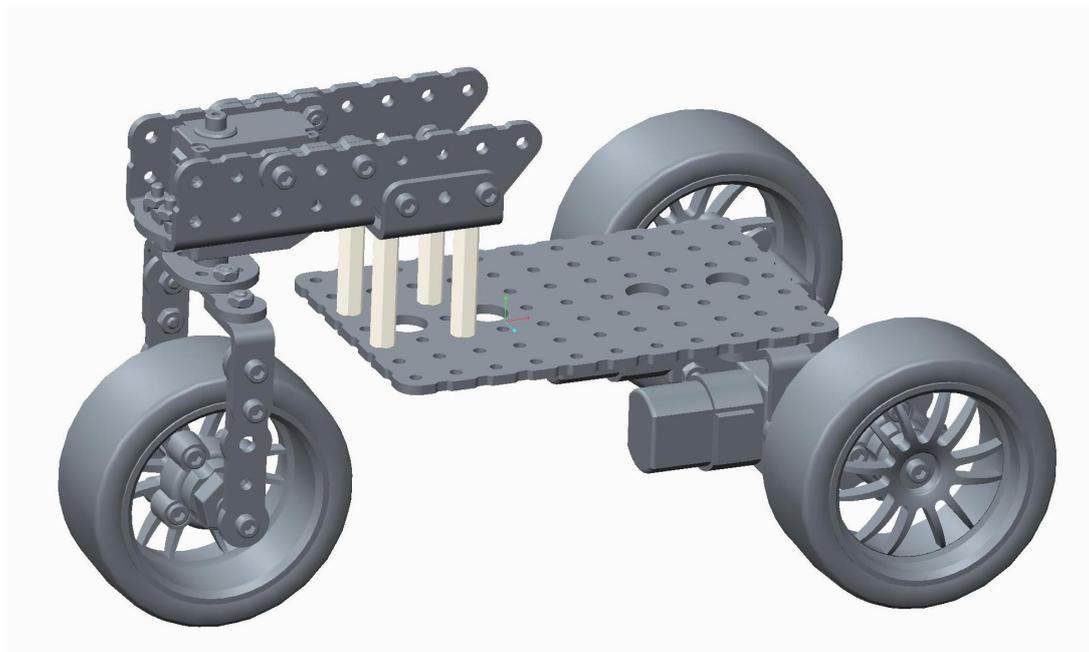
3. 烧录一段运动程序，观察车轮运动情况，如有异常，可利用改变杜邦接头正反的方式调整，保证同侧车轮运动一致。

4. 更换一部分车身零件，或改变现有零件的组装方式，使机器人结构在同样原理下达到不同的外观效果。

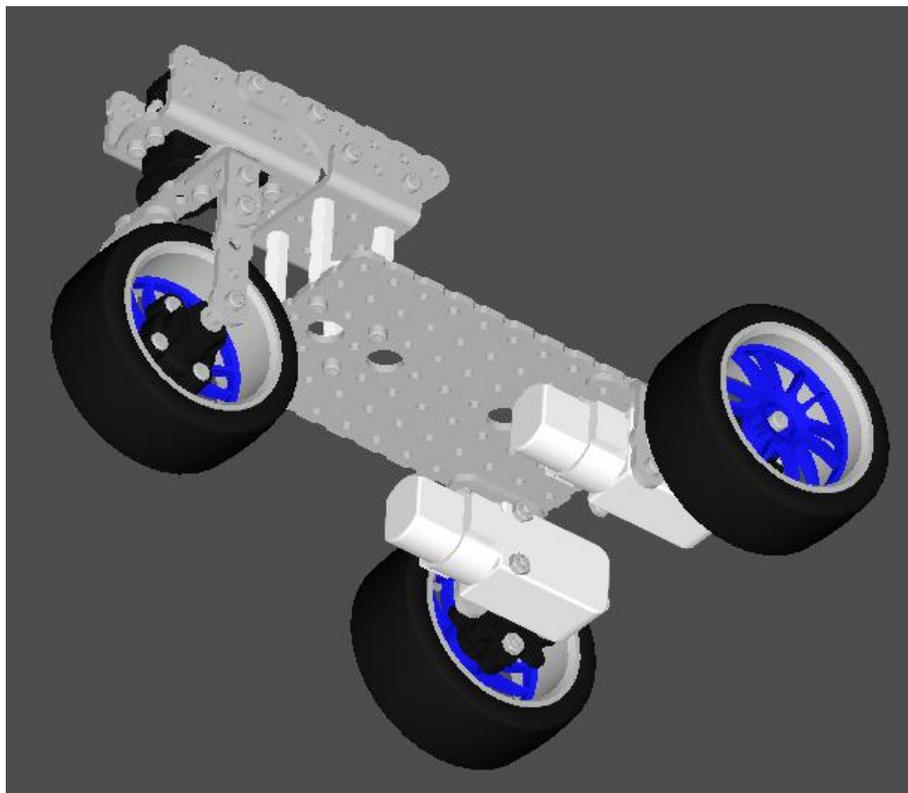
## 第四课 转向机构

转向机构可以让转向的控制更加精确。尤其是采用伺服电机控制转向的时候，可以进行特定角度的转向。但是不能原地旋转。

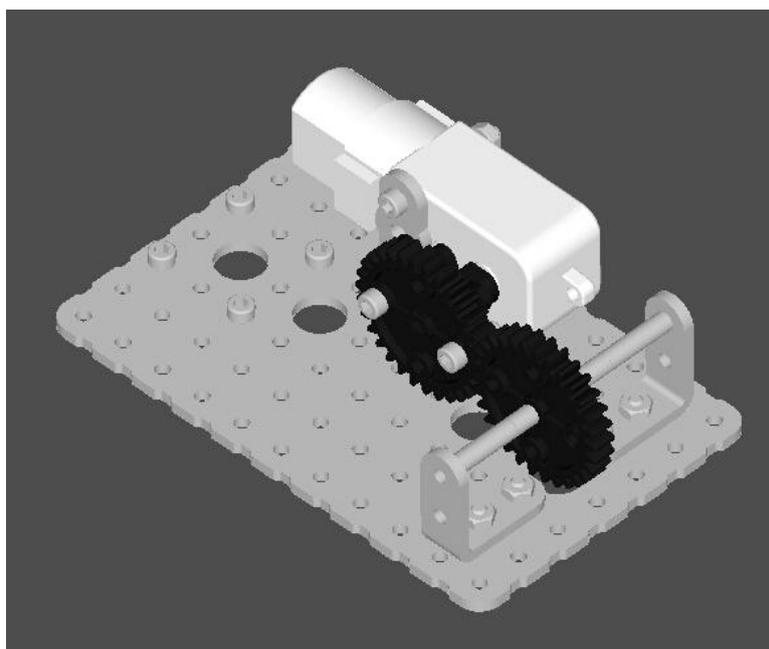
比如下图的三轮车，该机构是铰接式转向，它的转向机构由伺服电机驱动，可以直接输入伺服电机的角度位来决定转弯的角度值。这个机构可以分为转向模块化和后轮驱动模块两部分，两部分完全可以独立设计。



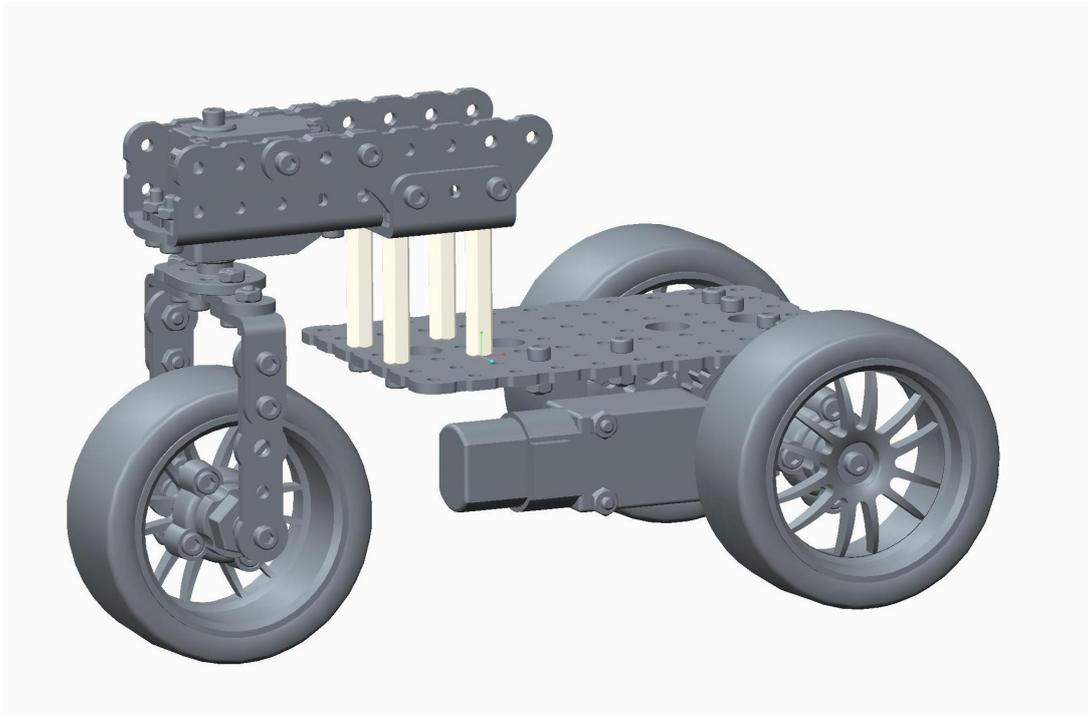
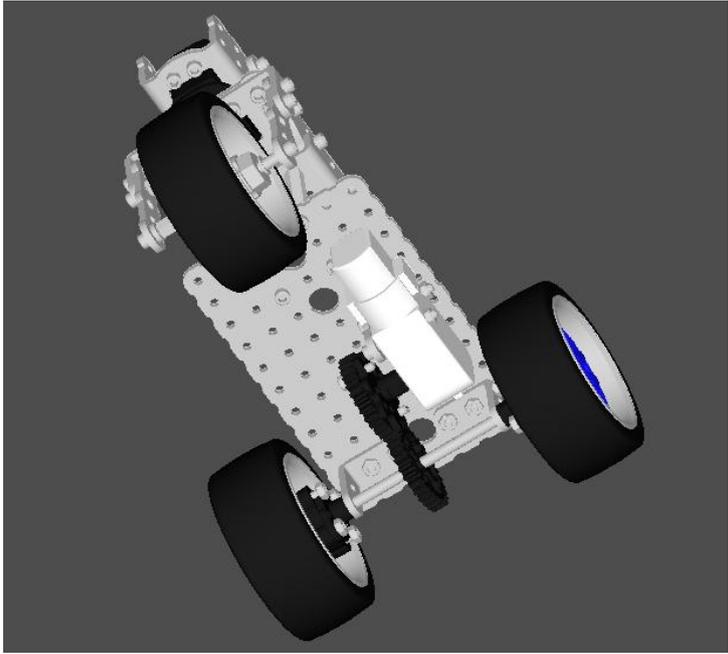
后轮驱动部分最简单的可以采用两个驱动轮模块，通过共用一个直流接口进行控制。



但是即使两个同型号的直流电机也会存在物理上的误差，因此我们可以用传动机构让两个轮子的协同性更好。比如，可以采用齿轮传动轴模块。



从而构建一个更合理的底盘机构。

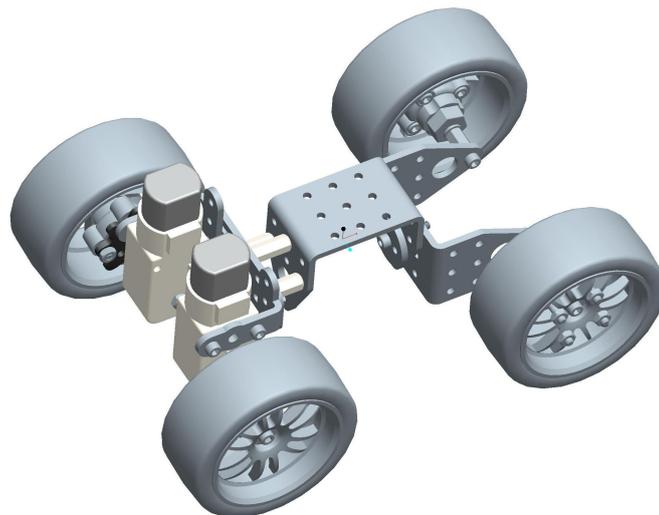
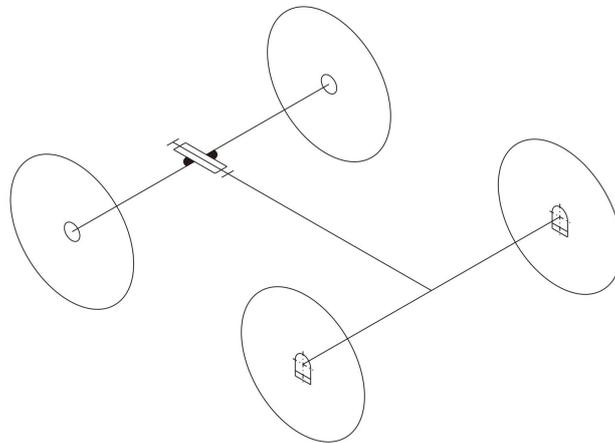


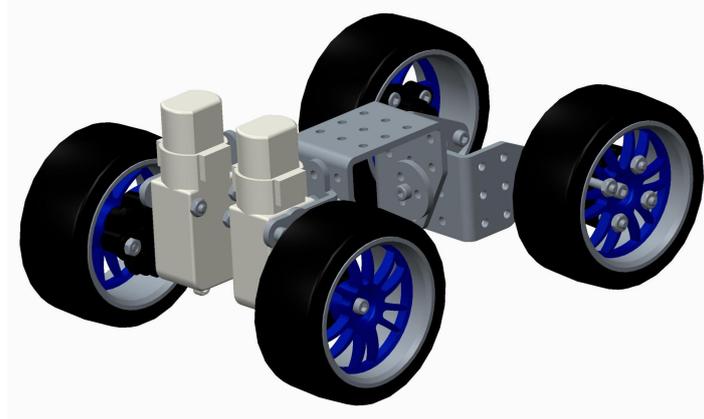
## 第五课 悬挂机构

悬挂系统是汽车的车架与车桥或车轮之间的一切传力连接装置的总称,其作用是传递作用在车轮和车架之间的力和力矩,并且缓冲由不平路面传给车架或车身的冲击力,并衰减由此引起的震动,以保证汽车能平稳地行驶。

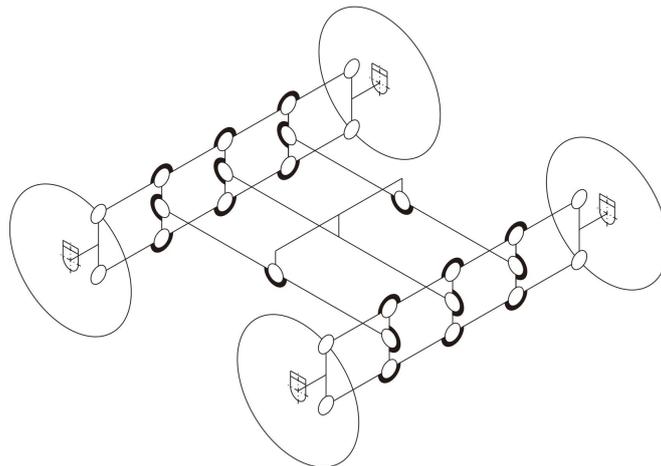
以下是两种带有悬挂系统的全地形机器人案例:

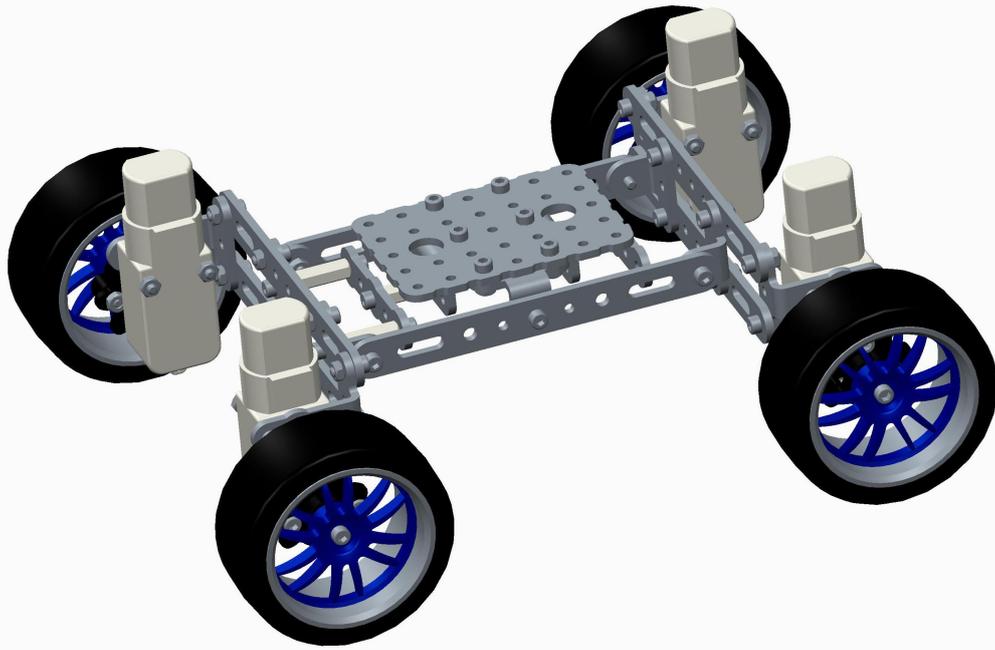
(1) 这是一个单点悬挂底盘方案,前、后轮系统利用一个铰链结构进行连接,遇到普通障碍时前后轮发生相对扭转,保证起码 3 个轮子不会脱离地面。





下图是一个利用连杆组设计的底盘悬挂方案，前轮、后轮均安装在四边形连杆组上，中间车架也是一个四边形连杆组，从而构成了一个空间连杆组。遇到普通障碍时连杆组发生平行形变，从而保证起码 3 个轮子不会脱离地面。





悬挂系统不仅仅可以用于越障，还可以提升整个车体的抓地性能，避免打滑、增加爬坡能力、减小车身颠簸幅度等。

汽车的悬挂系统习惯性区分为独立悬挂和非独立悬挂。按此标准，上面两种方案均属于非独立悬挂。在现实中，采用非独立悬挂和独立悬挂的底盘都很常见。另外，悬挂机构的设计思路其实还有很多，大家可以根据情况提出自己的设计，不求完整模仿某种已有典型方案，但求用最经济合理的方式做出自己的创新。

## 第六课 履带机构

**实验目的：**1. 熟悉履带模块在底盘机构中的应用。

**实验性质：**验证型实验

**实验课时：**2 课时

**参考资料：**000\_module005\_cb.stp；026a\_Tracked vehicle\_2m.stp；026b\_Tracked vehicle\_2m；履带的组装与拆卸.ppt

**实验步骤：**

### 1. 组装履带机器人

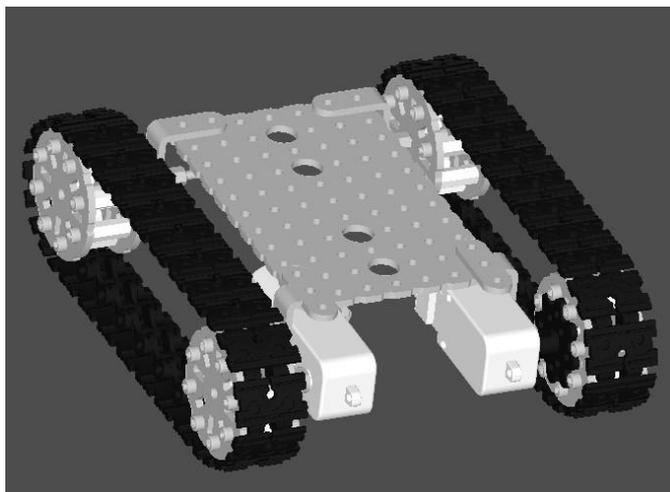


图 3.60 履带车 026a

**结构说明：**履带可以传送转矩到随动轮，从而让前后两组轮子都能驱动。运动能力更强，爬坡、翻越障碍的能力更强。可拆卸的履带片方便使用者调整设计方案，从而构造不同长度和造型的履带。图中履带更像是传送带，对于模型底盘、小型工程机械已经够用了，但是在实际工程中，往往需要增加更多的带轮、负重轮或者张紧轮，或者更换履带材质。

**运动特性：**转动灵活，摩擦力好，地形适应能力强。

2. 尝试控制它运动，控制方式和控制一个双轮车是一样的。
3. 调整履带长度，组装一个迷你履带车，并尝试控制它运动

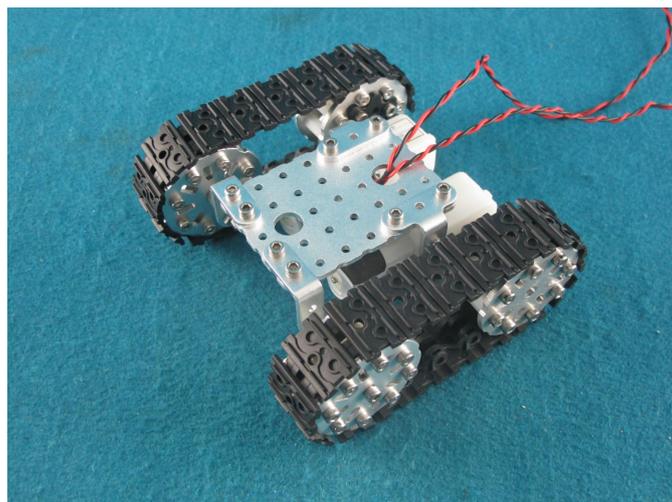


图 3.61 迷你履带车 026b

## 第四章 自主控制

### 第七课 传感器采值

**实验目的：**1. 了解传感器测值的概念；2. 学会利用 serial monitor 监测模拟量传感器的数值采集；3. 学会使用图形化程序编写测值程序；4.了解数字量传感器和模拟量传感器的不同。

**实验性质：**验证型实验

**实验课时：**2 课时

传感器可以大体上分为数字量传感器和模拟量传感器。我们之前在设置针脚值得时候接触过类似的概念，比如“设置针脚数字值”和“设置针脚模拟值”。传感器都可以从外界环境中检测到一些信号，数字量传感器检测到的信号是 0 或者 1（即“未触发”和“触发”），而模拟量传感器检测到的信号则是一个范围内的许多数值，这些数值都是电信号。

传感器的数值都可以通过一个叫做 serial monitor（串口监视器）的工具获取。我们只要编写一段代码就可以实现这个功能。

#### （一）数字量传感器测值

1. 数字量传感器可以用这种方法监测触发方式和触发条件，但是效果其实和“触发点亮 LED” 什么区别。我们可以用的语句是：



```
void setup()
{
  pinMode( 14, INPUT);
  Serial.begin(9600);
}
```

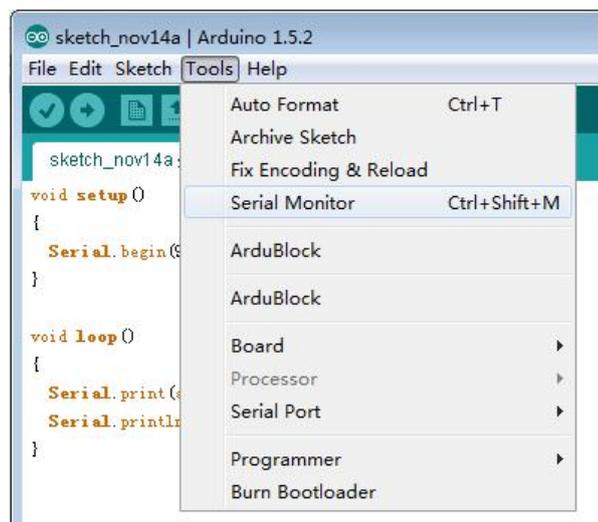
```
void loop()
{
  Serial.print(! ( digitalRead(14) ));
  Serial.println();
}
```

2. 在“实用命令”菜单里可以找到“串口打印加回车”这个图形语句。“串口打印”意思就是在串口监视器里显示，对应的函数是 Serial.print ()。我们把这个程序烧录之后，就可以接上黑标/白标传感器，并打开 serial monitor 查看检测到的数据。

在图形化界面的上方最右侧，有 serial monitor 按钮。



在 C 语言界面的 Tools 菜单里面，也可以找到 serial monitor 选项



另外，C 语言界面上方最右侧的“放大镜”按钮，也是 serial monitor。



打开之后即可看到传感器获取到的数据。

在监测过程中，主控板必须烧录上文提到的程序，且必须始终连在电脑上。

## (二) 模拟量传感器测值

模拟量传感器检测到的数值是某个范围内的许多数值，那么这个数据是如何被测到的呢？

1. 我们仍然是通过 serial monitor (串口监视器) 获取。请编写以下程序并烧录。



```
void setup()
{
  Serial.begin(9600);
}

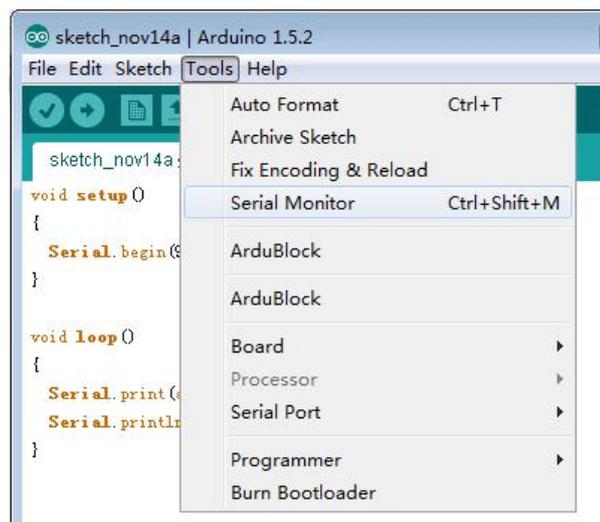
void loop()
{
  Serial.print(analogRead(14));
  Serial.println();
}
```

2. 在“实用命令”菜单里可以找到“串口打印加回车”这个图形语句。“串口打印”意思就是在串口监视器里显示，对应的函数是 Serial.print ()。我们把这个程序烧录之后，就可以接上黑标/白标传感器，并打开 serial monitor 查看检测到的数据。

在图形化界面的上方最右侧，有 serial monitor 按钮。



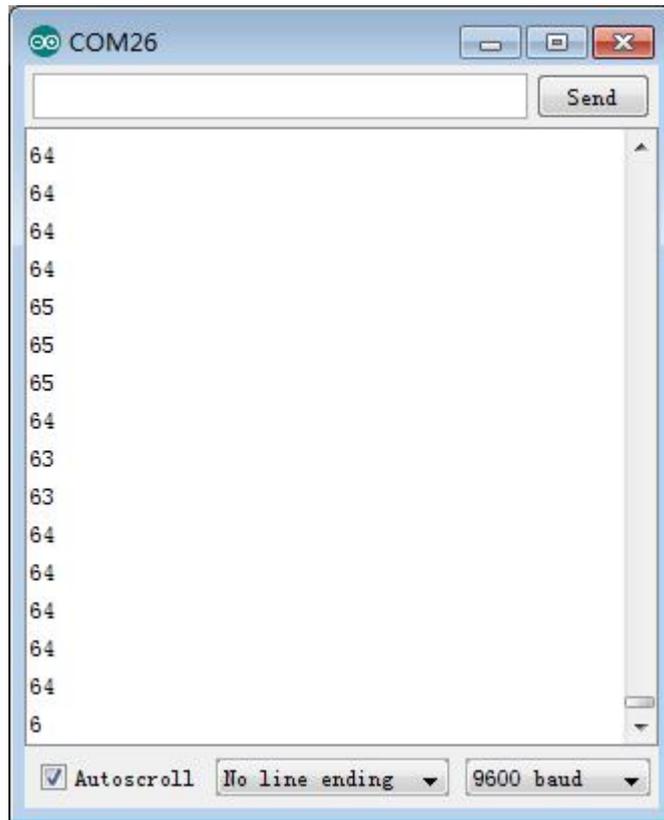
在 C 语言界面的 Tools 菜单里面，也可以找到 serial monitor 选项



另外，C 语言界面上方最右侧的“放大镜”按钮，也是 serial monitor。



打开之后即可看到传感器获取到的数据



在监测过程中，主控板必须烧录上文提到的程序，且必须始终连在电脑上。

学会了这个技能，我们就可以利用 serial monitor 去了解各种传感器的触发条件，以及检测到的数据情况了。如果数据太乱，就可以加上延迟语句，隔一段时间监测一次。

## 第八课 条件语句

**实验目的：**1. 了解一种避障功能的实现方式；2. 了解条件语句的概念，学习编写条件语句。

**实验性质：**设计型实验

**实验课时：**2 课时

**参考资料：**请参考**光盘**中的“电子元件资料”，Basra 控制板，Bigfish 扩展板；Bigfish 连接常规传感器.ppt；实验 1；

**主要器材：**触碰传感器×1

**实验步骤：**

- 1.在 023 号机构上的适当位置安装触碰传感器，连好电路；
- 2.阅读、编写并烧录下面的程序，它将实现这样一个功能：小车持续前进，遇到障碍后后退并转向另一个方向行驶，以躲避障碍。



生成的 C 语言代码为：

```
void setup()
{
  pinMode( 14, INPUT);

  pinMode( 9 , OUTPUT);

  pinMode( 10 , OUTPUT);

  pinMode( 5 , OUTPUT);

  pinMode( 6 , OUTPUT);
```

```
}  
  
void loop()  
{  
  if (!( digitalRead(14) ))  
  {  
    digitalWrite( 9 , LOW );  
    digitalWrite( 10 , HIGH );  
    digitalWrite( 5 , LOW );  
    digitalWrite( 6 , HIGH );  
    delay( 1000 );  
    digitalWrite( 9 , HIGH );  
    digitalWrite( 10 , LOW );  
    digitalWrite( 5 , LOW );  
    digitalWrite( 6 , HIGH );  
    delay( 1000 );  
  }  
  else  
  {  
    digitalWrite( 9 , HIGH );  
    digitalWrite( 10 , LOW );  
    digitalWrite( 5 , HIGH );  
  }  
}
```

```

digitalWrite( 6 , LOW );

delay( 1000 );

}
}

```

3.这个例程比较繁琐，下面我们来试试用设置子程序的方法来改写一下。

学习并编写以下程序，烧录进主控板，理解子程序的概念，并了解其在程序书写上的优势：

程序结构更规范，各个功能模块更容易维护和管理，出错方便排查。子程序图标为：

**子程序** 和 **子程序 执行** ，双击改名。



生成的 C 语言代码为：

```

void turnright();

void back();

```

```
void forwards();

void setup()
{
  pinMode( 14, INPUT);
  pinMode( 9 , OUTPUT);
  pinMode( 10 , OUTPUT);
  pinMode( 5 , OUTPUT);
  pinMode( 6 , OUTPUT);
}

void loop()
{
  if (!( digitalRead(14) ))
  {
    back();
    delay( 1000 );
    turnright();
    delay( 1000 );
  }
  else
  {
```

```
forwards();

delay( 1000 );

}

}

void turnright()
{

digitalWrite( 9 , HIGH );

digitalWrite( 10 , LOW );

digitalWrite( 5 , LOW );

digitalWrite( 6 , HIGH );

}

void back()
{

digitalWrite( 9 , LOW );

digitalWrite( 10 , HIGH );

digitalWrite( 5 , LOW );

digitalWrite( 6 , HIGH );

}

void forwards()
```

```
{  
  
digitalWrite( 9 , HIGH );  
  
digitalWrite( 10 , LOW );  
  
digitalWrite( 5 , HIGH );  
  
digitalWrite( 6 , LOW );  
  
}
```

## 第九课 条件嵌套语句

**实验目的：** 1.了解条件嵌套语句的写法，学习编写条件嵌套语句； 2.实现小车循迹。

**实验性质：** 验证型实验

**实验课时：** 2 课时

**参考资料：** 请参考 Bigfish 连接常规传感器.ppt； 视频： 后轮随动黑标小车.mpg

**主要器材：** 灰度传感器×2

**实验步骤：**

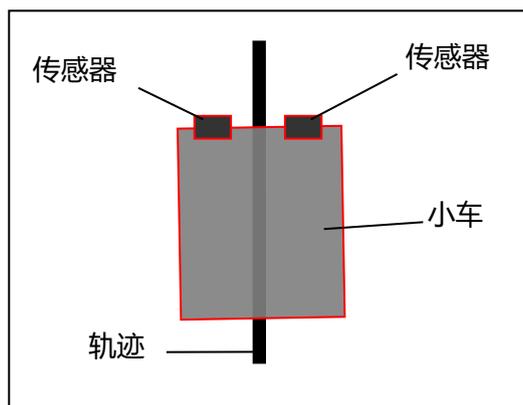
1. 将两个灰度传感器安装在 023 号机构的底部前端，传感器距离车轮越远效果越好，具体位置请自己尝试，连好电路；
2. 在白色场地上用黑色绝缘胶带（在两个灰度传感器之间，越宽效果越好）铺设一条轨迹，直线、弧线、圆均可；

这种循迹方案的运动原理：

要想识别地面上的黑线或者白线，很容易可以想到使用灰度传感器，而且至少要有 2 个灰度传感器。当只安装一个传感器时，一旦小车偏离轨迹就不好办了，所以还要想办法在小车快要离开轨迹的时候把它拉回来，这样就需要另外一个传感器。

所以我们最少要用到两个灰度传感器，一个安装在车头左侧，一个安装在车头右侧，如果左侧传感器检测到轨迹，说明小车右偏，就向左行驶来纠正；同理，如果右侧传感器检测到轨迹，说明小车左偏，就向右行驶来纠正。这样就保证轨迹始终在两个传感器之间。

如下图所示：



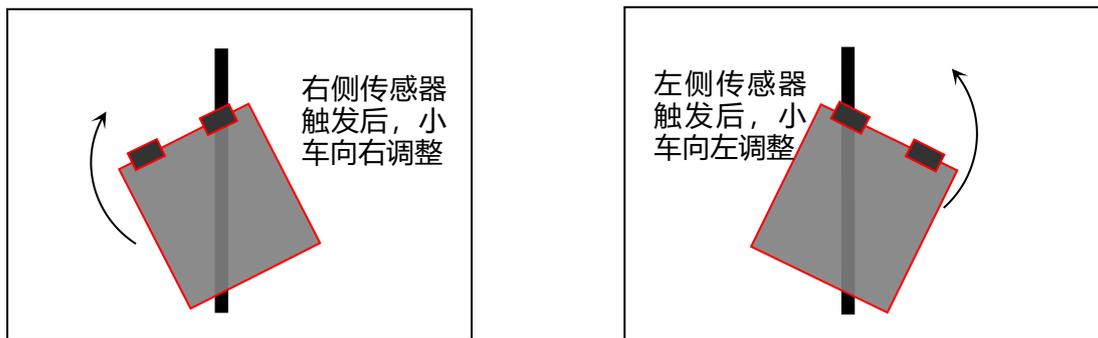


图 9.1 检测原理及程序设计设想

3. 阅读、编写并烧录下面的程序，它将实现这样一个功能：把双轮万向小车放置在如图 9.2 所示的场地上，小车能够自动沿着黑线行驶。



图 9.2 黑线场地

传感器触发情况、小车行驶状态、对应行为策略表：

传感器 1	传感器 2	小车状态	动作
0	1	小车主偏	向右调整
1	0	小车右偏	向左调整
1	1	到达终点	停止
0	0	正常	前进

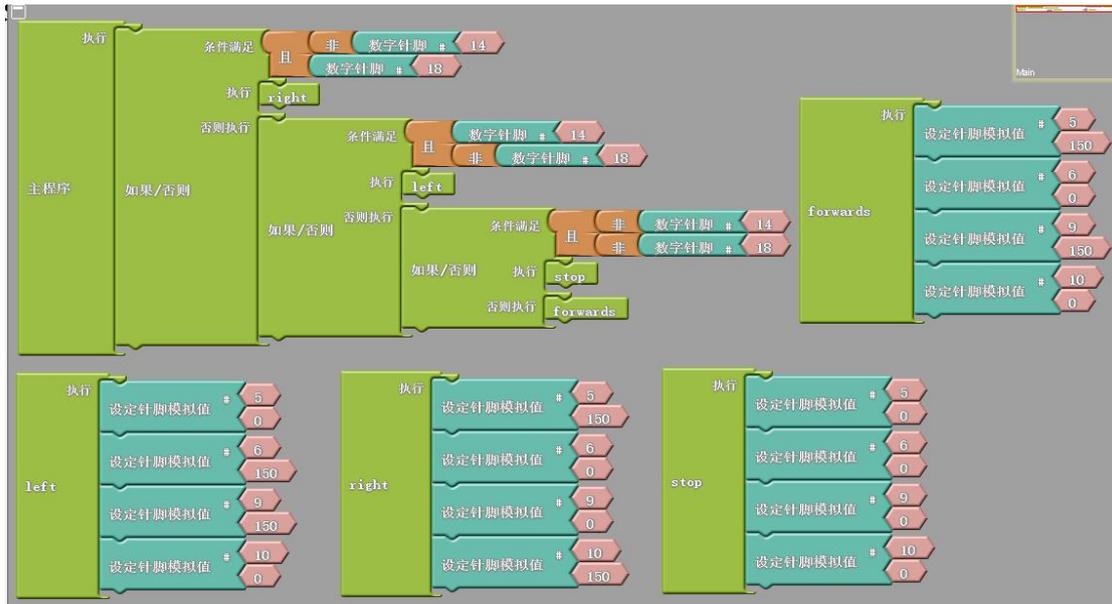


图 9.3 小车沿黑线轨迹行驶程序

生成 C 语言代码为:

```

void stop();

void left();

void right();

void forwards();

void setup()
{
    pinMode( 18, INPUT);

    pinMode( 14, INPUT);

    pinMode( 10, OUTPUT);

    pinMode( 6, OUTPUT);

    pinMode( 5, OUTPUT);

    pinMode( 9, OUTPUT);
}

```

```
}  
  
void loop()  
{  
  if (( !( digitalRead(14) ) && digitalRead(18) ))  
  {  
    right();  
  }  
  else  
  {  
    if (( digitalWrite(14) && !( digitalWrite(18) ) ))  
    {  
      left();  
    }  
    else  
    {  
      if (( !( digitalWrite(14) ) && !( digitalWrite(18) ) ))  
      {  
        stop();  
      }  
      else  
      {
```

```
        forwards();  
    }  
}  
}  
}  
}  
}  
  
void stop()  
{  
  
    analogWrite(5 , 0);  
  
    analogWrite(6 , 0);  
  
    analogWrite(9 , 0);  
  
    analogWrite(10 , 0);  
}  
  
void right()  
{  
  
    analogWrite(5 , 150);  
  
    analogWrite(6 , 0);  
  
    analogWrite(9 , 0);  
  
    analogWrite(10 , 150);  
}
```

```
void forwards()
{
    analogWrite(5 , 150);
    analogWrite(6 , 0);
    analogWrite(9 , 150);
    analogWrite(10 , 0);
}

void left()
{
    analogWrite(5 , 0);
    analogWrite(6 , 150);
    analogWrite(9 , 150);
    analogWrite(10 , 0);
}
```

#### 4. 重要提示:

小车结构、场地、电机速度等因素，对循迹效果影响非常明显。以下要点请大家注意。

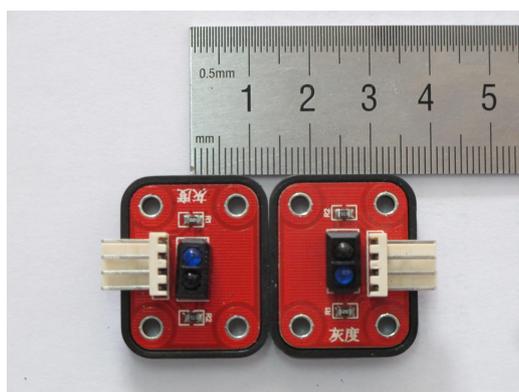
- (1) 传感器与小车的车轮，距离要尽量远，否则会造成小车转弯角度过大，可以用杆件等加长距离或改变电机的安装位置；
- (2) 如果地面不是很平整，尽量不要把万向轮装在前进方向上，容易与地面干涉；
- (3) 传感器与地面距离要在 1~3 厘米以内，简单来说，023 机构的主平板和传感器之间用 30 的螺柱就差不多了。烧录以下程序，在连接 USB 的状态下，打开 Tools→Serial Monitor，

监测传感器是否可以正确触发，帮助确定正确的传感器安装位置以及场地。



这个程序可以调用串口显示功能，在 Serial Monitor 中显示 A0 端口传感器触发状态，未触发显示 0，触发显示 1。

(4) 灰度场地背景颜色要尽量浅，最好就是白色，要尽量平整。对于 2 个传感器的小车来说，黑线的宽度很重要（2 个以上传感器，黑线的宽度就无所谓了）。由于两个“探索者”黑标传感器的检测头最近也要有 2.5cm 左右的距离，因此黑线不应小于 2.5cm，否则会造成小车转弯角度太大。



(5) 两个传感器安装时尽量靠近，传感器检测头的距离应和黑线宽度相当。

(6) 循迹是一种非常综合的机器人实验。对综合调试能力要求很高。就这个实验来说，程序很简单，但是小车要走好却很难。小车的结构、传感器安装、传感器触发条件、场地状况的综合调试，比程序本身重要。大家在这个实验中要把这些程序外因素的调试训练到位。

5. 改写程序，使机器人实现更好的循迹效果。

## 第十课：有限状态机

**实验目的：**1. 了解一种循迹功能的实现方式；2. 了解灰度传感器的工作特性及安装注意事项。3. 掌握 switch 语句的写法。4. 了解“有限状态机”的概念，掌握一种常用的编程框架，学会分析传感器触发组合状态。

**实验性质：**验证型实验

**实验课时：**4 课时

**参考资料：**请参考 Bigfish 连接常规传感器.ppt，实验 9；“探索者”编程框架.ppt

**主要器材：**灰度传感器×3

有限状态机 (Finite-state machine) 简称 FSM，表示有限个状态以及在这些状态之间的转移和动作等行为的数学模型。它把复杂的控制逻辑分解成有限个稳定状态，在每个状态上判断事件。由于有限状态机有有限个状态，因此可以在实际中实现。有限状态机可以广泛的应用于机器人多个传感器触发组合状态的判断，大大提高检测效率。

状态表

之前我们了解过条件判断，机器人的传感器触发一般用条件判断来做。

这时机器人程序的一般思路是：

```
如果
机器人的某几个传感器触发了；
机器人的某几个电机做个什么事；
做多久；
如果
机器人的另外某几个传感器触发了；
机器人的某几个电机做个什么事；
做多久；
```

所以我们总是要用到大量的 if 语句。比如双轮小车的某个功能：

如果  
机器人的 1 号传感器触发了;  
机器人的左侧电机顺时针转;  
机器人的右侧电机逆时针转;  
持续 5 秒;  
如果  
机器人的 2 号传感器触发了;  
机器人的左侧电机逆时针转;  
机器人的右侧电机顺时针转;  
持续 5 秒;  
否则  
都不转

用伪码写出来就是:

```
if { Sensor(端口 a,触发); //传感器触发时此句为真, 否则为假 }  
{  
    Motor(L,顺);  
    Motor(R,逆);  
    Delay 5;  
}  
if { Sensor(端口 b,触发); }  
{  
    Motor(L,逆);  
    Motor(R,顺);  
    Delay 5;  
}  
else  
{  
    Motor(L,停);  
    Motor(R,停);  
}
```

无论是 ARM7 还是 Mehran 都很容易实现。

在只有一个传感器的情况下, 我们假设这是个数字量传感器。那么我们可以得到一个状态表格:

状态序号	传感器 1
1	1
2	0

这个传感器有两个状态。

而当有两个传感器时，则有四个状态。

状态序号	传感器 1	传感器 2
1	1	1
2	1	0
3	0	1
4	0	0

如果我们用 if 语句写这四个状态，就显得比较长。

状态序号	传感器 1	传感器 2	伪码
1	1	1	<pre> if {   Sensor(1,1);   Sensor(2,1);   ..... } </pre>
2	1	0	<pre> if {   Sensor(1,1);   Sensor(2,0);   ..... } </pre>
3	0	1	<pre> if {   Sensor(1,0);   Sensor(2,1);   ..... } </pre>
4	0	0	<pre> else ..... </pre>

在编程的时候，状态罗列的越全，未来机器人的 bug 越少。但是随着传感器的增多，状态数量按 2 的 N 次幂增加，大量的 if 语句使执行效率变得很低，经常出现识别不灵的情况。我们有必要换一种高效写法。

多个确定数量的传感器的触发组合，符合有限状态机的概念，有限状态机一般是用 Switch 语句来实现。如：

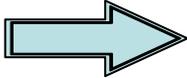
```
switch(s)
{
    case 1 : {动作 1;}break;
    case 2 : {动作 2;}break;
    case 3 : {动作 3;}break;
    case 4 : Act_Stop();break;
    default;;break;
}
```

不难发现，这段语句实现的关键，就是识别出上页表中的 1、2、3、4，四个状态序号。

那么问题就来了：我们如何让机器人知道自己传感器的触发组合对应于 1、2、3、4 的哪个序号呢？

## 二进制状态表

下面，我们把每组传感器返回值看成一个二进制数值。

传感器 1	传感器 2		二进制结果	十进制结果
1	1		11	3
1	0		10	2
0	1		01	1
0	0		00	0

结果我们发现了一种新的、可计算的编码方式：

新序号	传感器 1	传感器 2
0	0	0
1	0	1
2	1	0
3	1	1

于是，只要我们知道传感器的触发状态，也就知道了序号；知道了序号，也就知道了传感器的触发状态。用这个序号去写 switch 语句，再合适不过了。下面我们要做的是，用一种算法，让机器人能够返回自己接收到的传感器组合值的二进制数据。

### 算法精解

我们可以使用以下算法来实现：

- 首先设置一个变量  $s$ ，这个  $s$ ，将存储传感器组的二进制状态序号。

- 我们还需要用到一个重要的运算符 “<<” ，这个运算符的意义是：左移  
如：1<<n，意思是 1 向左移动 n 位，空出来的数位用 0 填补。

如：1<<1，结果就是 10；1<<2，结果就是 100；101<<1，结果就是 1010

- 只要让机器人依次返回各个传感器的状态数值，最早获取的，移到最左；第二获得的，移到“倒数第二左”，……，以此类推。即可获得。

如两个传感器均触发：

先获得 1 号的数值（真）并左移 0 位，得 

0	1
---	---

再获得 2 号的数值（真）并左移 1 位，得 

1	0
---	---

两数值取“或”，即可得 11

数学问题解决了，很容易就可以转化为程序语句：

```
s=0;

for(i=0;i<2;i++) //因为此例中有 2 个传感器，i 取 2

{

s=s|(Servo(i+1, 触发判断)<<i); //获得传感器值，移位，或运算

}
```

于是 switch 语句可以写为：

```
switch(s)

{

case 0x00 : {动作 0;}break; //序号也可以写作 16 进制数值

case 0x01 : {动作 1;}break;

case 0x02 : {动作 2;}break;

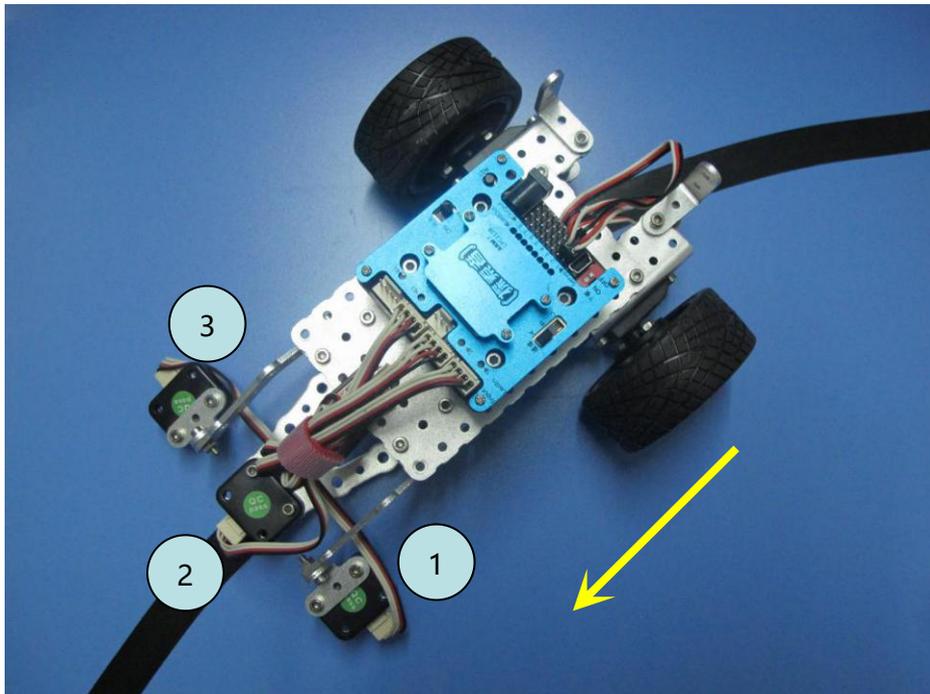
case 0x03 : {动作 3;}break;
```

```
default;;break;
```

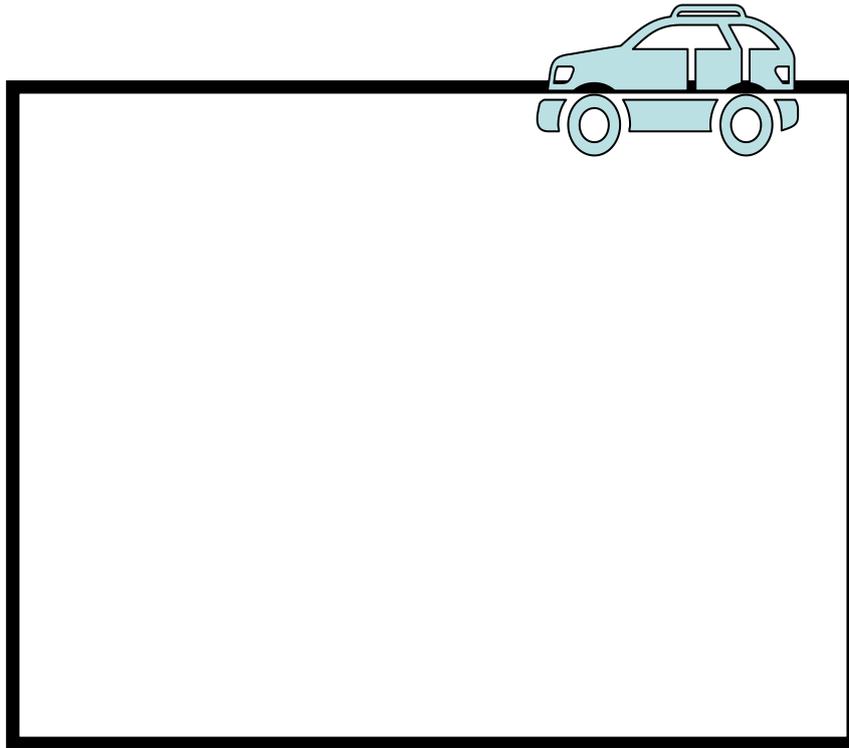
```
}
```

## 策略表

下面我们以一个“三传感器双轮循迹小车”程序为例，再来推导一遍：



假设跑道是矩形，行进方向如图：



传感器触发情况、小车行驶状态、对应行为策略表如下：

传感器 1	传感器 2	传感器 3	序号	小车状态	动作
0	0	0	0	都没触发，可能是跑偏了	后退，转向
0	0	1	1	小车左偏	左轮逆时针转，向右调整
0	1	0	2	小车正中	左轮逆时针转，右轮顺时针转，前进
0	1	1	3	在这个行进方向上不可能	无
1	0	0	4	小车右偏	右轮顺时针转，向左调整
1	0	1	5	在此跑道上不可能	无
1	1	0	6	遇到转角	右轮顺时针转，左转
1	1	1	7	在此跑道上不可能	无

伪码如下：

```

s=0;

for(i=0;i<3;i++)

{

s=s|(Input(i+1,1)<<i);

}

switch(s)

{

case 0x00 : 停;break;

case 0x01 : {Motor(L,逆);Motor(R,停);}break;

case 0x02 : {Motor(L,逆);Motor(R,顺);}break;

case 0x04 : {Motor(L,停);Motor(R,顺);}break;

case 0x06 : {Motor(L,停);Motor(R,顺);}break;

default;;break;

}

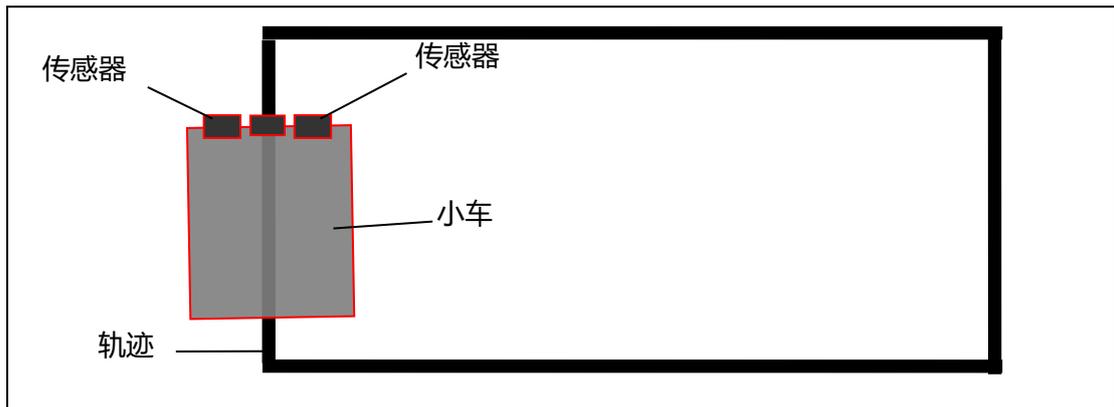
```

这段代码中的动作，完全由策略表分析获得，因此，当状态比较多时，用户要学会利用策略表进行分析，从而确定机器人的动作策略，而不是凭空想象。

#### **实验步骤：**

1. 将 3 个灰度传感器安装于 023 机构的底部前端，连接好电路；
2. 请大家学习以下程序，根据自己的机构实际情况调整参数，实现稳定的 3 传感器巡线。

该段程序为，有 90 度转角的，黑色轨迹上的，3 传感器巡线。



为了让程序结构更规范, 我们可以为小车编写子函数, 大家可以根据下面的子函数的写法来调整电机的接线, 务必让子函数的定义和小车现实中运动是相符的。

左转:

```
void Left()
{
    digitalWrite( 5 , LOW );
    digitalWrite( 6 , HIGH);
    digitalWrite( 9 , HIGH );
    digitalWrite( 10 , LOW );
}
```

右转:

```
void Right()
{
    digitalWrite( 5 , HIGH );
    digitalWrite( 6 , LOW );
    digitalWrite( 9 , LOW );
}
```

```
digitalWrite( 10 , HIGH );  
  
}
```

前进:

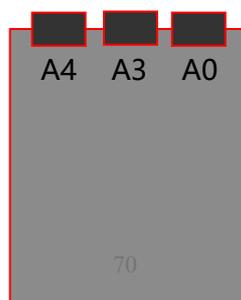
```
void Forwards()  
  
{  
  
digitalWrite( 5 , HIGH );  
  
digitalWrite( 6 , LOW );  
  
digitalWrite( 9 , HIGH );  
  
digitalWrite( 10 , LOW );  
  
}
```

后退:

```
void back()  
  
{  
  
digitalWrite( 5 , LOW );  
  
digitalWrite( 6 , HIGH );  
  
digitalWrite( 9 , LOW );  
  
digitalWrite( 10 , HIGH );  
  
}
```

电路连接: 当小车车顶如下图所示对着你时, 从右到左的传感器分别连接 A0、A3、A4 口。

电机的连接方式请大家自己摸索。



完整程序为：

```
int pin[3] = {A0, A3, A4};           //按车头前进方向，从右至左定义，后面经过公  
式计算，会转化为从左至右的顺序  
  
int s;  
  
void setup()  
{  
  pinMode( 5 , OUTPUT);  
  pinMode( 6 , OUTPUT);  
  pinMode( 9 , OUTPUT);  
  pinMode( 10 , OUTPUT);  
}  
  
void loop()  
{  
  s = 0;  
  for(int i=0; i<3; i++)           //循环获取三个传感器的值  
  {  
    s|= (!digitalRead(pin[i]) << i); //经过左移运算和或运算后,按照 A0、  
A3、A4 的顺序产生一个三位 2 进制数值，表示 3 个传感器的组合触发状态  
  }  
  switch (s)  
  {
```

```

    case 0x00: //三个均未触发

    back();

    Left();

    break;

    case 0x01: //右侧传感器触发, 直线上摆动或遇到右转弯

    Right();

    break;

    case 0x02: //中间传感器触发, 直线上直行

    Forwards();

    break;

    case 0x04: //左侧传感器触发, 直线上摆动或遇到左转弯

    Left();

    break;

    case 0x06: //左侧两个触发, 遇到左转弯

    Left();

    break;

    default::break;

    }

}

void Left()

{

```

```
digitalWrite( 5 , LOW );  
  
digitalWrite( 6 , HIGH);  
  
digitalWrite( 9 , HIGH );  
  
digitalWrite( 10 , LOW );  
  
}
```

```
void Right()  
  
{  
  
digitalWrite( 5 , HIGH );  
  
digitalWrite( 6 , LOW );  
  
digitalWrite( 9 , LOW );  
  
digitalWrite( 10 , HIGH );  
  
}
```

```
void Forwards()  
  
{  
  
digitalWrite( 5 , HIGH );  
  
digitalWrite( 6 , LOW );  
  
digitalWrite( 9 , HIGH );  
  
digitalWrite( 10 , LOW );  
  
}
```

```
void back()
{
digitalWrite( 5 , LOW );
digitalWrite( 6 , HIGH );
digitalWrite( 9 , LOW );
digitalWrite( 10 , HIGH );
}
```

## 第五章 实战

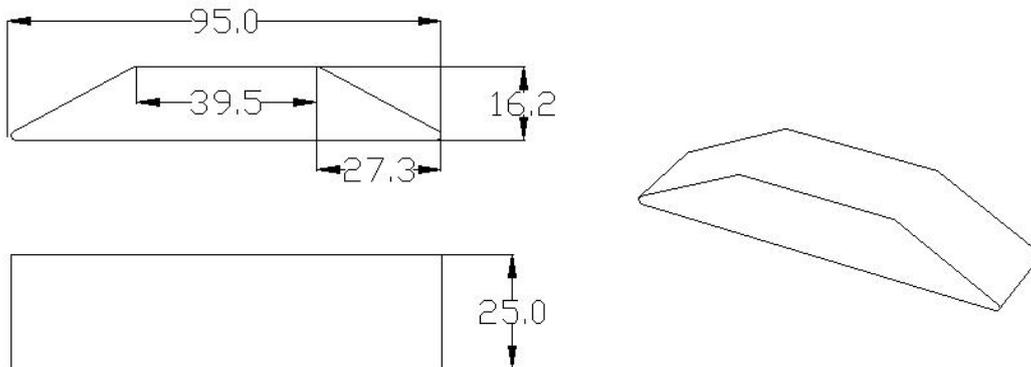
### 第十一课 窄桥

窄桥尺寸图：

单位：cm

材料：发泡 EVA

颜色：黑色



#### 斜坡

当我们的车子遇到斜坡时，一般会出现两种情况：

- 1、爬不上去；
- 2、能爬上去但走不直。

首先我们来分析一下爬不上去的原因：

- 1、是因为车子摩擦力不够导致车子在斜坡上打滑；
- 2、是因为车子动力不够导致车子静止在斜坡上；
- 3、是因为车子的初始速度太小导致车子达到斜坡时的惯性太小冲不上去。

就这三点原因我们给出了几种解决的办法：

1、增加轮胎的摩擦力，我们可以通过给轮胎缠绕履带（图 1）或者在履带上附加螺钉（图 2）。

2、提高小车动力，我们可以增加电机来做到动力的提升（图 3）。

3、提高小车的速度，我们可以增加轮胎的直径来提高速度，比如用履带缠绕轮胎或者使用大履带轮胎（图 4）。



图 1



图 2



图 3



图 4

现在我们在分一下走不直的问题：

1、是因为轮胎在空间上的不对称导致各个轮胎的旋转中心轴交叉以至于小车不能走直线。

2、是因为每个电机的转速不一致，导致有的快有的慢以至于小车不能走直线。

3、是因为轮胎的大小不一致，存在细微的误差。

就这三点原因我们给出了几种解决的办法：

1、调整结构：支架组装拧紧、联轴器和圆周舵机的输出头拧紧、更换使用长久导致零件磨损变形的零件、竖直方向不对称可以增加纸张、塑料带等不超出比赛规则的薄片。

2、调整电机转速，可以通过程序微调。

3、如果小车始终向某侧偏斜，可以尝试调换左右边的一两个轮胎，以平衡误差。

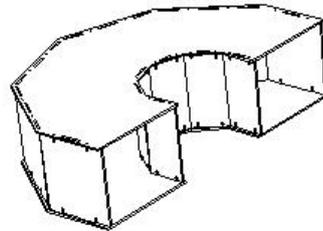
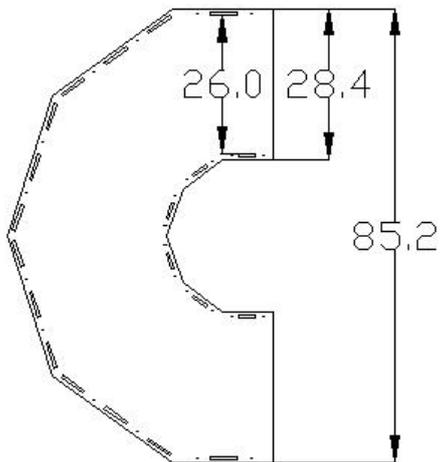
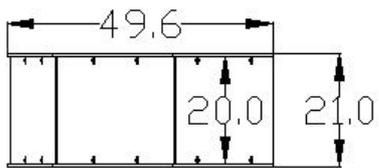
## 第十二课 管道

管道尺寸图：

单位：cm

材料：亚克力

颜色：透明



### 隧道

当我们的车遇到隧道时，一般只会出现一种情况：就是小车冲进去撞在隧道里并不拐弯，以至于卡死在里边。所以我们可以增加一些辅助机构来纠正小车以使小车进行转向，比如导轮（如图 5）。

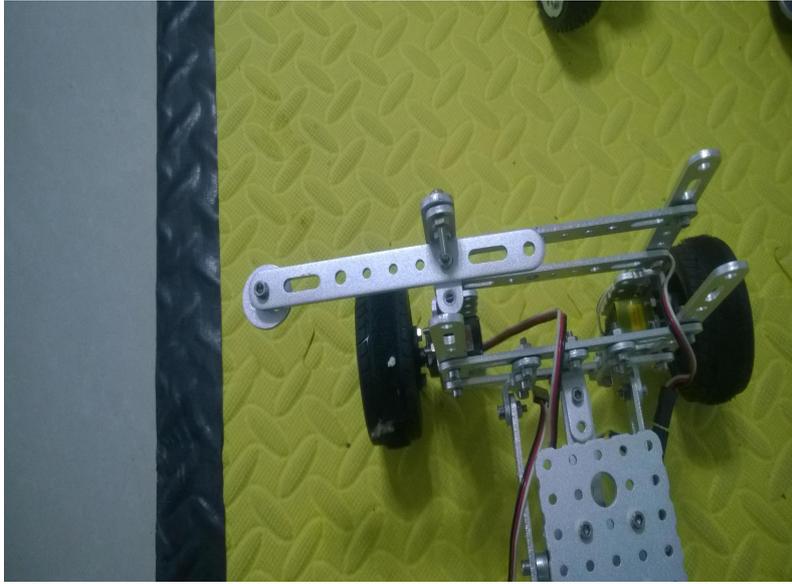


图 5

从控制的角度考虑，可以使用传感器来辅助转向。

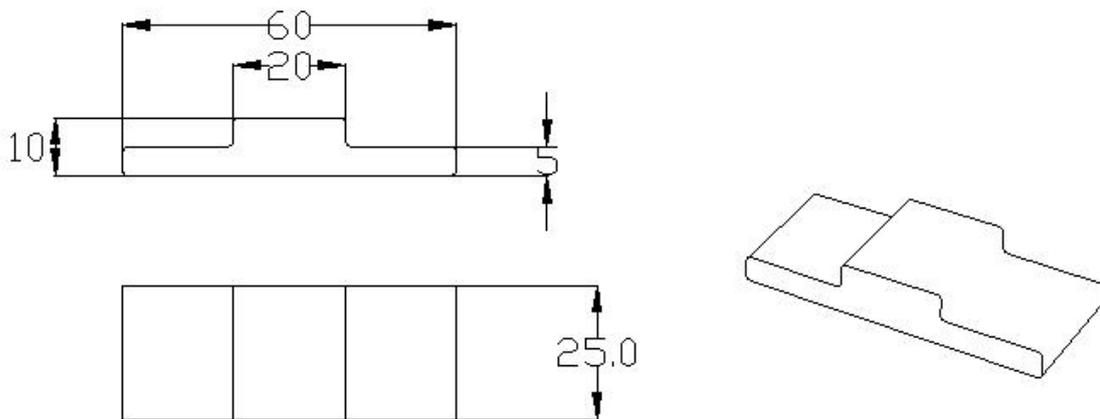
## 第十三课 台阶

台阶尺寸图：

单位：cm

材料：发泡 EVA

颜色：黑色



### 台阶

当我们的车遇到台阶时，可能爬不上去：

所以我们给出了以下几个方案：

- 1、增加轮胎的摩擦力，比如缠绕履带或者履带附加螺丝（方法如上图 1、2）；
- 2、增加悬挂机构，可提高小车的可变型性和减震效果，但同时也可提高抓地力；
- 3、增加辅助机构，比如添加辅助机械杆（如图 6）；
- 4、使用大直径车轮，以至于小车可以轻易爬上第一阶梯。



图 6

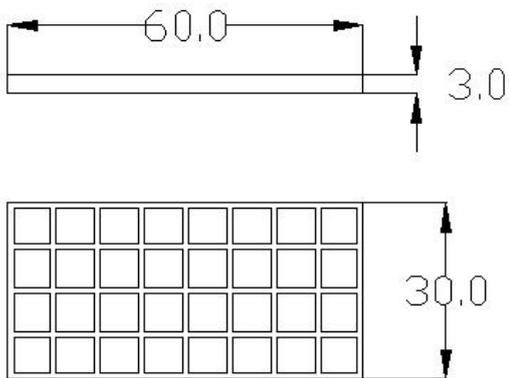
## 第十四课 栅格

栅格地面尺寸图：

单位：cm

材料：工程塑料

颜色：蓝色



### 栅格

当我们的车子遇到栅格时，一般会出现两种情况：

- 1、凹陷；
- 2、卡底盘。

首先我们来分析一下凹陷的问题：

- 1、是因为栅格每个格子的尺寸太大；
- 2、是因为轮胎太小；

就这两点原因我们给出了几种解决的办法：

- 1、缩小栅格尺寸（这是不可能的）；
- 2、提高轮胎直径，可以在轮胎上缠绕履带或者使用大履带轮（方法图片如上图 1、4）；
- 3、增加轮胎宽度，可以将多个轮胎并排组装在一起等（如图 7）；

4、放弃轮胎改用履带车。



图 7

现在我们在分一下卡底盘的问题：

卡底盘是因为小车底盘太低了，所以我们可以通过三种办法解决这个问题：

- 1、调整机械结构，提高底盘与地面的距离；
- 2、增加轮胎的直径，以达到提高底盘的效果（方法如上图 1、4）；
- 3、使用履带车。

## 第十五课 通过全程

### 一、循迹相关问题

现象 1：程序没有问题，但小车无视黑线，完全没有循线。

分析问题：

小车无视黑线，可能原因是：

- ①传感器坏了；
- ②传感器安装位置不合适，导致传感器没有触发；
- ③程序中直行时 delay 时间太长导致没能及时检测到传感器状态；

解决办法：

(1) 我们可以先单独测试一下传感器是否损坏，用串口测值或点亮 LED 的方法进行测试，如果确认传感器坏了，更换新传感器。

(2) 如果传感器是好的那么可能是传感器的安装位置不合适，先不要将传感器固定在机器人身上，而用手持的方法，利用用串口测值或点亮 LED 的方法进行测试，找到合适的安装位置，再进行安装。

(3) 缩短 delay()时长，提高检测频率。

现象 2：循线时，传感器检测到黑线，但小车发生“蛇行”；或紧接着识别黑线不灵。

分析问题：

- (1) 可能是小车两侧轮子差速过大，导致转弯半径太小，从而“蛇行”；
- (2) 可能是转弯动作 delay 时间过长，导致下次识别不灵。
- (3) 可能是小车速度太快，外加程序执行效率不高，导致未能识别到黑线。

解决办法：

- (1) 转弯时减小电机转动差速，或用转向机构替代差速转向；
- (2) 缩短 delay 时间，或不使用 delay；
- (3) 更换转速慢的电机或提高程序执行效率。

现象 3：小车在翻越障碍时，传感器误触发。

分析问题：

误触发的原因可能是：

(1) 遇到障碍，轮子被抬起，车身倾斜，传感器与地面间的距离变化，导致传感器误触发。

(2) 窄桥和楼梯是黑色的，且长度不同，当共用一种 delay 时长时，有可能误触发传感器。

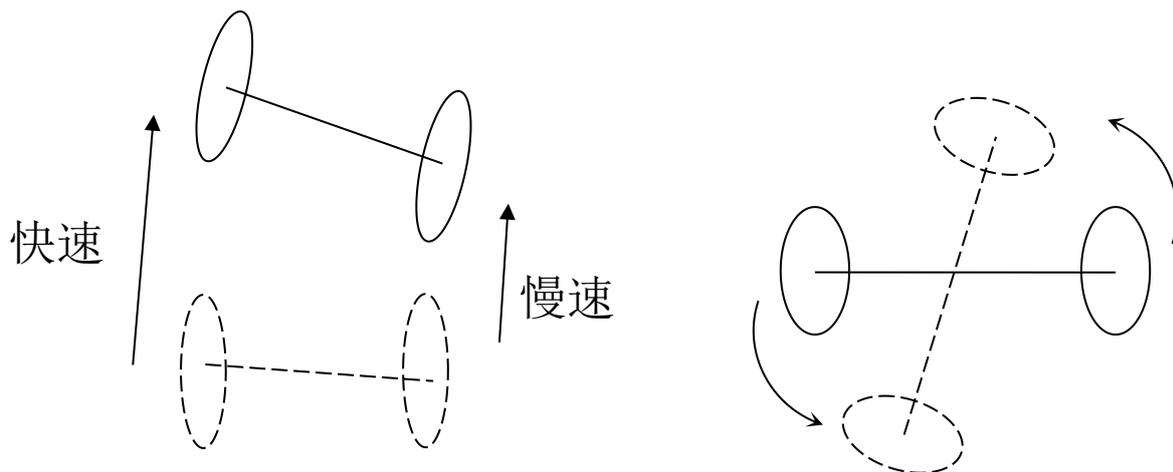
解决办法：

- (1) 调整 delay 的时间长度，尽量降低误触发引起的动作影响。
- (2) 通过对横线计数，为窄桥和楼梯分别安排合适的 delay 时长。

## 二、循迹直角转弯问题

在直角位置转弯时，一般的转弯动作容易偏离。需要左右反转，原地转向。

但循迹时原地转向的动作会造成行进速度变慢，因此，最好是遇到路口再触发一个短暂的原地转向动作。



同向一快一慢：转弯

同速一正一反：原地旋转

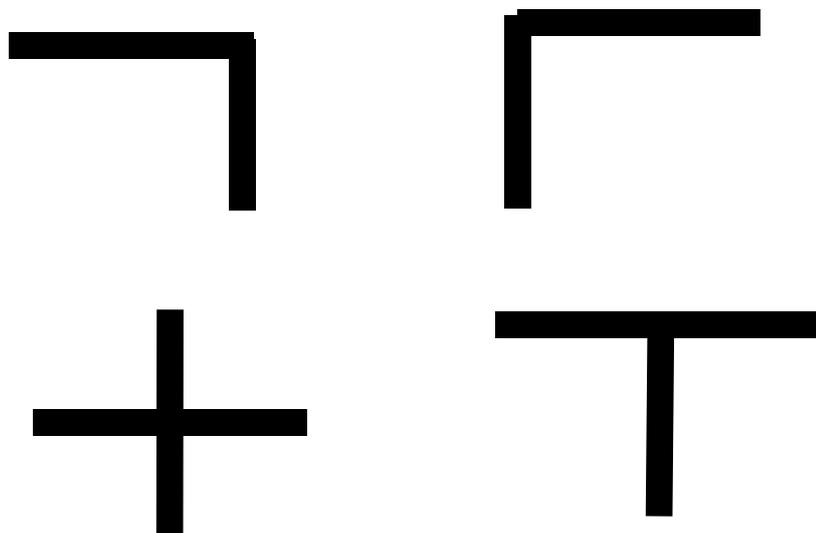
### 三、传感器触发实时监测问题

机器人在工作过程中出现异常，传感器是否正常触发了？这是困扰许多参赛队的问题。

我们平时学习和做实验时，往往利用串口监视器监测传感器的触发情况，但是这总是离不开电脑，如何更方便的在机器人工作过程中监测呢？其实利用 LED 点阵上的亮点即可监测。

### 四、路口识别问题

场地上有几种路口：左转角、右转角、十字路口、丁字路口



1、两个黑标传感器可以区分左、右转角。10 左, 01 右;

2、三个黑标传感器可以区分左转角、右转角、十字/丁字路口, 110 左、011 右、111 十字/丁字路口;

3、四个黑标传感器 (第四个装在前端) 可以区分以上四种路口, 1100 左、0110 右、1111 十字、1110 丁字

## 五、路口计数问题

场地上共有十字路口 6 个, 丁字路口 2 个, 转角 9 个。在同一种路口可能有不同的动作, 那么机器人如何知道在哪个路口做哪个动作呢?

可设置几个变量, 如 i、j、k....., 分别对应几种路口, 过一个路口, 变量+1, 利用变量值为路口计数, 从而在不同的路口做不同的动作。

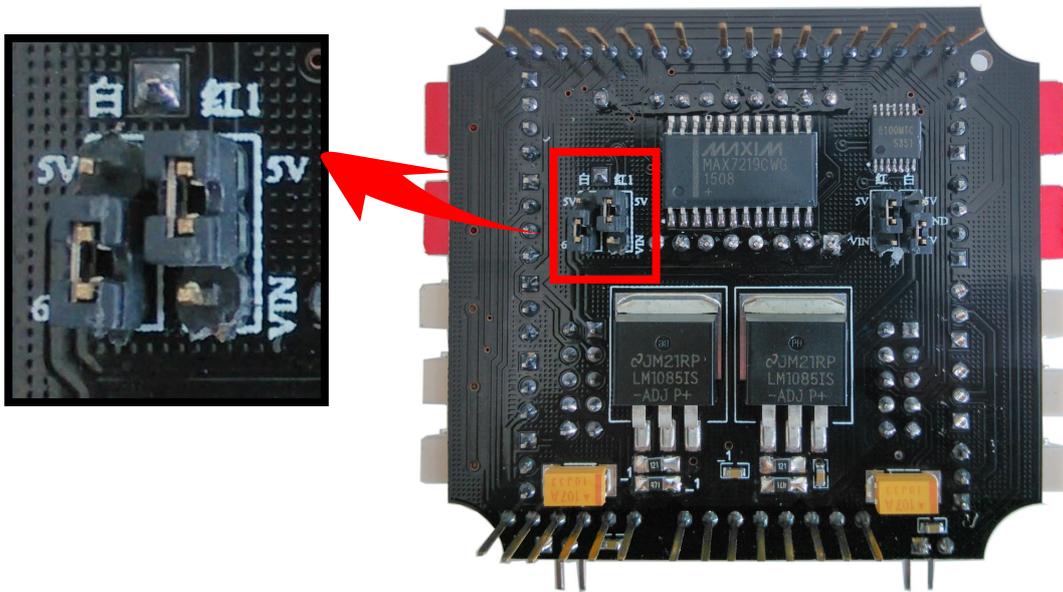
## 六、程序切换问题

场地上共有 11 个障碍, 一次性全部走完的可能性很低。有时候我们需要有几套程序, 用于不同的运行路线。

利用传感器即可完成程序的切换。首先编写针对不同路线的程序, 再利用简单的 if 语句即可管理。如传感器触发 1 次, 则执行第一套程序; 触发两次, 则执行第一套程序.....同时利用 LED 点阵显示标示程序序号。

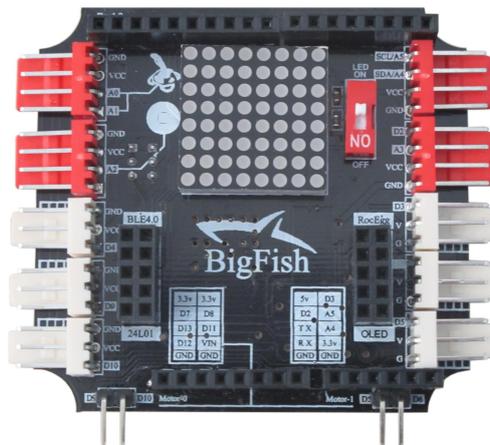
## 七、如何增加传感器接口?

对于 Bigfish 扩展板来说, 首先可以利用背后的跳线帽, 将舵机接口更改为数字传感器接口。



还有能扩展出更多传感器口的方法。

Basra 主控板总共有 20 个 I/O 口，理论上能同时连接的传感器上限是 20 个。



驱动数字量传感器和简单的模拟量传感器，只需要 3 根针：

GND、VCC、信号针（D 口或 A 口）。

因此我们可以把红色接口的两个信号针，用杜邦线单独引出来用。

或者从板子上下的横排座坞引出；

或者从下方的两个 2×5 座坞引出。

这种灵活的做法，可以最大限度的利用针脚，并且尽量减少对舵机数量的影响。

### 通过全程参考例程:

```
#include"LedControl.h"

LedControl lc=LedControl(12,11,13,1);

#define right  1

#define left   0

#define dt     800

int speedd = 255;

int j=0;

int way[10]={1,0,0,0};

int Sensor[4] = {A0,A2,A3,A4};

int Motor[4] = {5,6,9,10};

int state[4] = {1};

void setup()

{

    Serial.begin(9600);

    for(int i=0;i<4;i++)

    {

        pinMode(Sensor[i],INPUT);

        pinMode(Motor[i],OUTPUT);

    }

}
```

```

    lc.shutdown(0, false);

    lc.setIntensity(0,15);

    lc.clearDisplay (0);

    lc.setRow(0,0,11111111);

    lc.setRow(0,7,11111111);

    delay(1000);

    lc.clearDisplay (0);
}

void loop()
{
    lc.clearDisplay (0);

    //循环读取传感器值
    for(int i=1;i<3;i++)
    {
        state[i] = digitalRead(Sensor[i]);

        lc.setColumn(0, 2*i, byte(255*(!state[i])));
    }

    //如果遇到提示条
    if(!state[0]&!state[3])
    {
        zhangai(way[j]);
    }
}

```

```
        j++;  
    }  
    xunji();  
    delay(5);  
}  
  
void go()  
{  
    analogWrite(Motor[0],speedd);  
    digitalWrite(Motor[1],LOW);  
    analogWrite(Motor[2],speedd);  
    digitalWrite(Motor[3],LOW);  
}  
  
void wheel(int w)  
{  
    if (w==left)  
    {  
        digitalWrite(Motor[0],HIGH);  
        digitalWrite(Motor[1],HIGH);  
        analogWrite(Motor[2],speedd);
```

```

        digitalWrite(Motor[3],LOW);

    }

    else

    {

        analogWrite(Motor[0],speedd);

        digitalWrite(Motor[1],LOW);

        digitalWrite(Motor[2],HIGH);

        digitalWrite(Motor[3],HIGH);

    }

}

void xunji()

{

    if(!state[1]&!state[2])

        go();

    else if( (state[1]==1)&&(state[2]==0) )

        wheel(right);

    else if( (state[1]==0)&&(state[2]==1) )

        wheel(left);

}

```

```
void zhangai(int w)
```

```
{
```

```
}
```

## 第十六课 全地形比赛

比赛场地尺寸说明 (单位: cm)

比赛场地及四种障碍物 (栅格地面, 管道, 窄桥, 楼梯) 尺寸标记 (含引导黑线、比赛起点和终点)。

(1) 场地整体图:

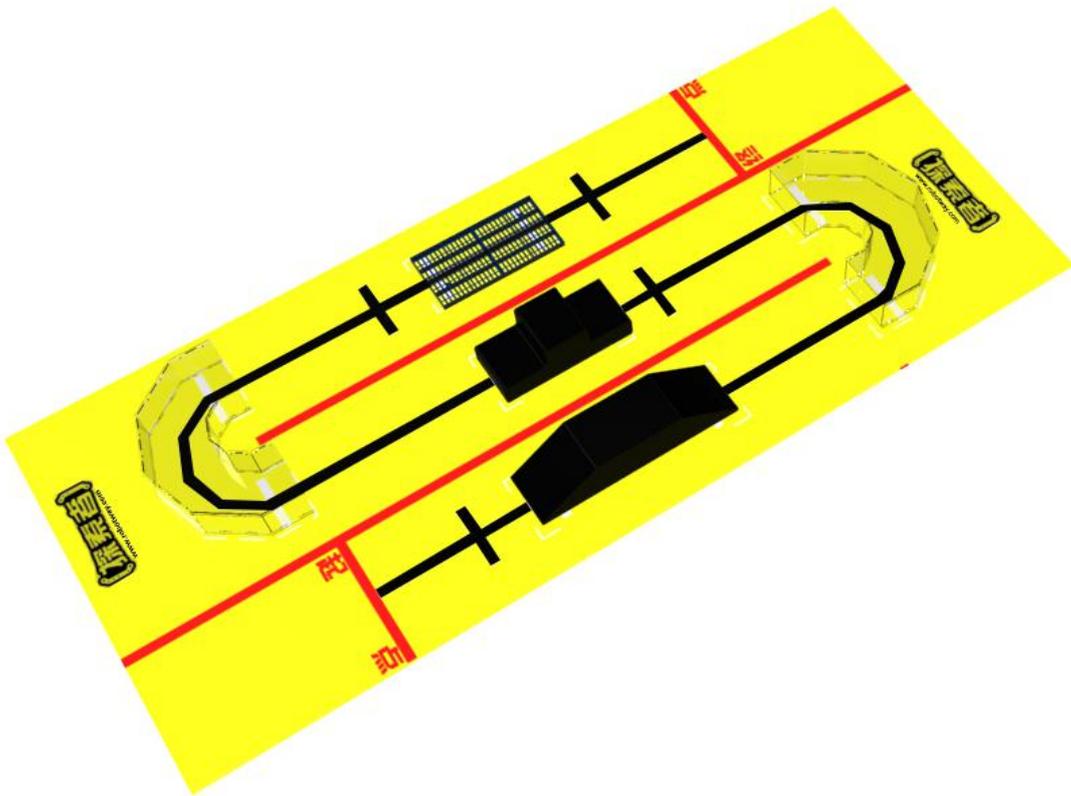


图 1: 场地整体效果图

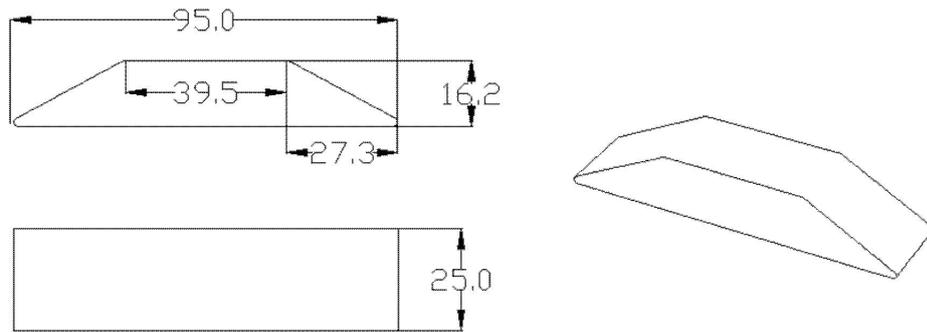


(2) 窄桥

单位: cm

材料: 发泡 EVA

颜色: 黑色



备注:

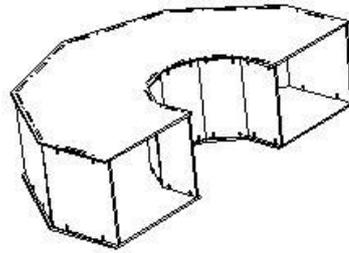
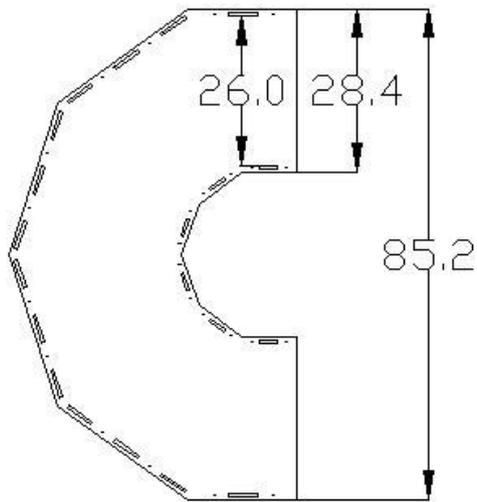
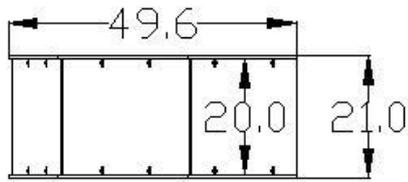
1. 单位: mm, 厚度250mm, EVA泡棉;
2. 未注倒角 $0.2 \times 45^\circ$ ;
3. 所有尺寸公差 $\pm 5\text{mm}$ 。

(3) 管道

单位: cm

材料: 亚克力

颜色: 透明

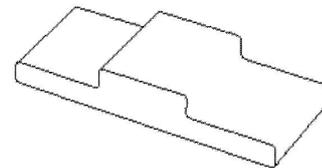
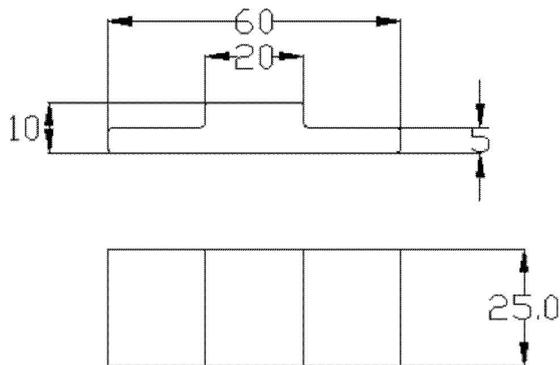


(4)台阶

单位: cm

材料: 发泡 EVA

颜色: 黑色



备注:

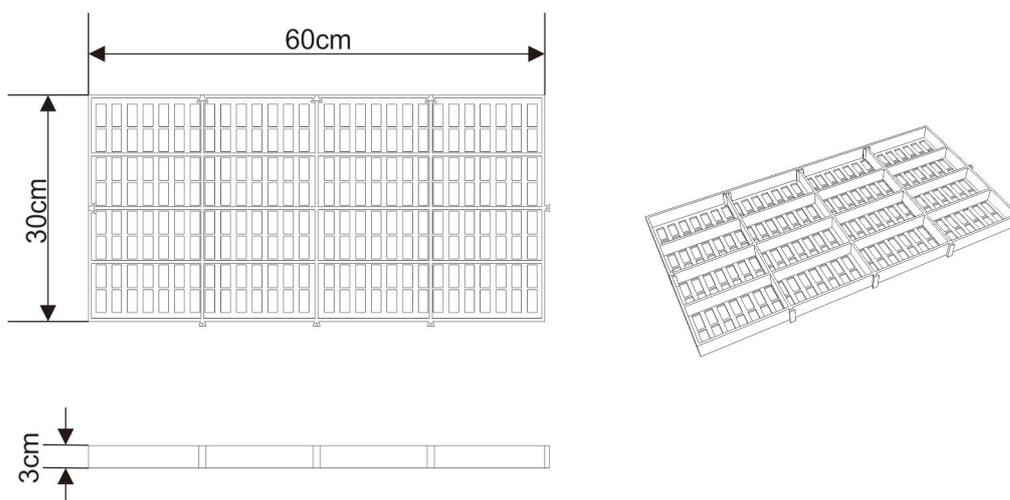
1. 单位: mm, 厚度250mm, EVA泡棉;
2. 未注倒角 $0.2 \times 45^\circ$ ;
3. 所有尺寸公差 $\pm 5\text{mm}$ 。

(5)栅格地面尺寸图:

单位: cm

材料: 工程塑料

颜色: 蓝色



备注: 所有尺寸公差 $\pm 5\text{mm}$ 。

## 探索者全地形机器人自主创新设计大赛

### 比赛规则

开设“探索者”全地形机器人自主创新设计比赛,是为了鼓励和推动在校大学生开展机器人项目的自主创新设计活动,使用“探索者创新套件”统一设计平台,以统一的创新主题“全地形机器人自主创新设计”,其目的是使参赛高校都能在统一的设计平台上完成统一的设计主题,设计制作各种新颖的机器人项目,实现各类机器人的设计、应用等方面的创新,从而推动机器人应用技术的不断发展,同时也便于统一展示创新设计的成果,合理公正评判大学生自主创新设计的成果。

#### 1.关于探索者全地形机器人的设计和制作要求

参赛队应根据大赛组委会提供的具有4种不同特性障碍物及比赛要求,采用“探索者模

块化机器人平台”设计制作全地形机器人。构成作品的零部件的类型不能超出“探索者模块化机器人平台（高级版）”（含A3002, Rino-MX2、Rino-MX201）和组委会提供的“扩展配件包”的范围，符合条件的零部件的使用数量不限。

仅允许使用以下辅助材料参与制作：打印用纸、塑料布、透明胶带、绝缘胶带、双面胶带、魔术贴、束线带、螺丝胶、止松垫、防滑螺母。

## 2.关于全地形机器人比赛障碍场地的设定

场地中设定四种五个不同特点、不同难度的障碍物，每种障碍物均有一定的分值，参赛队根据比赛规则自主设计制作机器人，完成穿越各个障碍物的比赛。障碍物分别为窄桥、管道（2个）、楼梯、模拟工业用栅格地毯，各障碍物由黑色引导线连接，形成完整的比赛赛道，并设置比赛起点和终点，比赛场地由组委会统一布置。

## 3. 比赛场地尺寸说明（单位：cm）

比赛场地及四种障碍物（栅格地面，管道，窄桥，楼梯）尺寸标记（含引导黑线、比赛起点和终点）。

场地整体图：

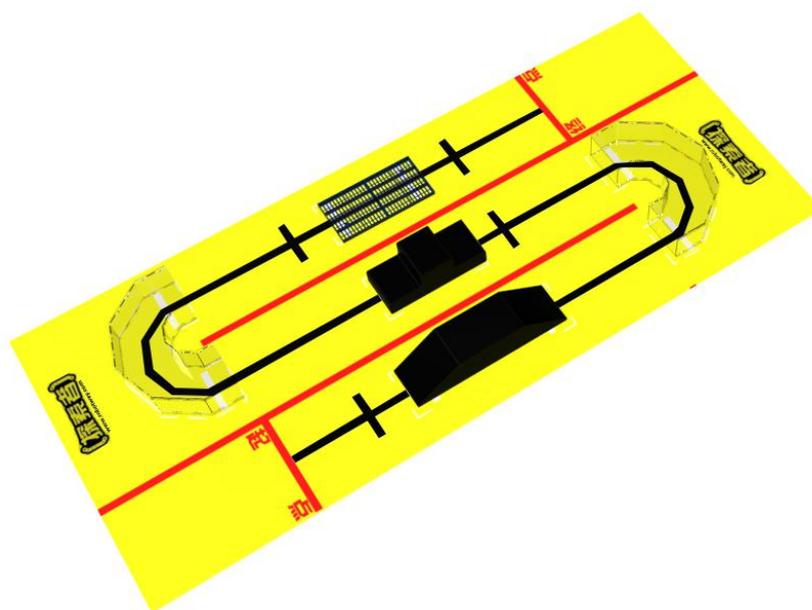


图1： 场地整体效果图

#### 4. 比赛流程

- (1) 参赛队伍在完成签到后，由大赛组委会组织统一抽签，参赛队按照抽签顺序进行比赛。
- (2) 每支队伍仅允许拥有1台机器人作品。
- (3) 上场前须对作品称重，并记录重量。
- (4) 参赛作品应自主控制，不允许任何形式的远程控制干预。
- (5) 比赛开始后，每个参赛队可指派一名选手持作品进入场地，该选手需使用鞋套，避免接触障碍。每个作品现场运行总时间限定在5分钟内，从作品首次启动开始计时。裁判席放置计时器公开倒计时，比赛过程中除非发生极端情况，否则不暂停计时。
- (6) 首次运行时，作品必须从出发区起跑，以作品任何一部分接触终点线作为跑完全程的标志。作品正常运行过程中选手不得接触作品。
- (7) 作品在挑战某个障碍失败时，允许重新运行。重新运行时，选手须将作品移至该障碍头部附近，作品投影不得与障碍有任何重合。
- (8) 已通过前一个障碍，在行驶过程中出现故障或路线偏离，作品投影未能与下一个障碍头部有重合，需要重新运行时，需从前一个障碍尾部附近重新运行。
- (9) 重新运行前，选手可以对作品进行调整，但不得将作品带出场地。
- (10) 发生以下情况之一时比赛终止：
  - ①作品跑完全程，比赛终止；
  - ②5分钟时间耗尽，比赛终止；
  - ③选手向裁判申请结束挑战，裁判判定比赛终止；
  - ④现场发生裁判认为必须终止比赛的情况，比赛终止。

#### 五、评分规则：

比赛作品综合得分C，满分100分，由：障碍完成分（75分）、计时分（10分），创新

得分（15分）构成。组委会根据各队得分高低评出一、二、三等奖（各奖项数量由大赛组委会决定）。

#### ①障碍完成分

此项成绩记为I：评分依据为障碍通过情况。按照通过障碍的数量计分，每个障碍15分。以“作品投影与障碍头部边界有重合并完全进入障碍，作品投影与障碍尾部边界有重合并完全离开障碍”为通过标准。计算公式为：

$$I=15\times\text{障碍完成数量}$$

#### ②计时分

此项成绩记为II，各队成绩按比赛终止时的剩余时间（精确至秒）转换为数字值，代入公式计算。计算公式为：

$$II=\text{剩余时间值}\times 2$$

例：假设某队比赛终止时剩余时间为3分25秒，则得到 $3.25\times 2=6.5$ 分。5分钟时间耗尽的得0分；

#### ③创新得分

此项成绩记为III，制作创新评分主要依据为参赛队在截止日期前提交的“探索者”全地形机器人自主创新设计技术报告电子版1份，不提交技术报告的队伍创新得分为0。技术报告内容主要包括作品名称、选手和指导老师的基本情况、作品简介、设计过程、制作过程；以及对作品创意设计及结构设计的说明、电路设计及程序设计的说明；以及该作品的科学性、实用性、新颖性、先进性和实用价值的说明，以及心得体会等内容。

由大赛组委会组织评委组依据下列评分标准对参赛作品综合打分（满分 15分）。

IIIA 设计评价（创新性、结构合理性、先进理论和技术应用）：10分。

IIIB 制作评价（稳定性，可靠性，规整性，美观性）：5分。

$$III=III A+III B$$

## 六、奖项分配：

之后，按以下公式计算比赛作品综合得分：

$$C=I+II+III$$

根据总成绩C排名。若总分C相同，则根据作品跑完全程的耗时决定排名，耗时少的排名靠前，若总分C仍相同，则根据作品重量决定排名，重量轻的排名靠前。

奖项分配方式由组委会决定。奖项公布后，进入30天异议期。

### 4.2 不获奖原则

各参赛队在比赛过程中如“未能完成比赛”，则不参与评奖，即不获奖。视为“未能完成比赛”的情况包括：

- ①损坏比赛场地，引发安全事故；
- ②不遵守赛场纪律，干扰他人参赛；
- ③参赛队员不符合参赛资格；
- ④制作材料不符合比赛要求；
- ⑤裁判专家组判定的其他情况。



**机器时代（北京）科技有限公司**

**ROBOTTIME BEIJING TECHNOLOGY. LTD**

电话：010-52801448

Email：web@robottime.cn

<http://www.robottime.cn>