

机器人感知与实践

实训指导书

2016.9

序言

正是在当下国家产业发展规划背景下，传统高校工程教育已经难以满足创新人才政策的培养需求，且工业 4.0 时代下，如“创客 Maker”、“机器人”、“模块化”、“智能制造工厂”、“智能硬件”、“创新创业”、“互联网信息化”、“校企拟合”等新兴理念大量涌入高校工程教育改革当中，创新工程教育以及创新人才培养，不只是单一学科单一方向的教学改革，而是涵盖多方面多角度的综合应用型创新体系，不只是传统高校教学实践内容，还包含了企业真实环境下的创新设计思维教育等等。

本实训指导书是适合大一新生的入门教材。涵盖机构设计和电子设计。基于机器人创新组件及开源 Arduio 开源环境，运用 ArduBlock 可视化编程进行电子功能设计。按照由浅入深编排各实验，为适应不同层次的学生，在各实验中增加实验扩展内容，鼓励学生在完成基本实验内容基础上，进行创新实践。

目录

第一课 零件的认知与使用.....	1
一、零部件的清点.....	1
二、机械零件认知.....	1
三、刚体结构连接.....	11
四、可动结构连接.....	11
第二课 简单结构的组装.....	12
实验一 使用三维软件辅助组装.....	12
实验二 组装滑板小车.....	14
实验三 组装双轮万向小车.....	15
实验四 组装创意造型.....	20
第三课 控制的基础知识.....	21
实验五 配置编程环境实验.....	24
实验六 blink.....	27
实验七 电机入门操作.....	29
实验八 伺服电机入门操作.....	31
第四课 控制双轮万向小车.....	34
实验十 双轮万向小车运动.....	36
实验十一 开关启动机器人.....	38
第五课 传感器的高级用法.....	40
实验十二 数字量传感器测值.....	40
实验十三 模拟量传感器测值.....	42
实验十四 超声波传感器测距算法.....	44
实验十五 超声波避障功能方案.....	48
第六课 舵机和履带的使用.....	49
实验十六 简易排爆车.....	49
第七课 机器人避障.....	54
实验十七 避障实验.....	54
第八课 机器人简单循迹.....	55
实验十八 小车循迹实验.....	55
第九课 机器人编程框架.....	59
实验十九 控制双轮循迹小车.....	66
第十课 机器人识别悬崖.....	67
实验二十 悬崖的躲避与巡检.....	67
第十一课 蓝牙遥控.....	68
实验二十一 蓝牙遥控机器人.....	68
第十二课 排爆机器人汇报表演.....	71

第一课 零件的认知与使用

一、零部件的清点

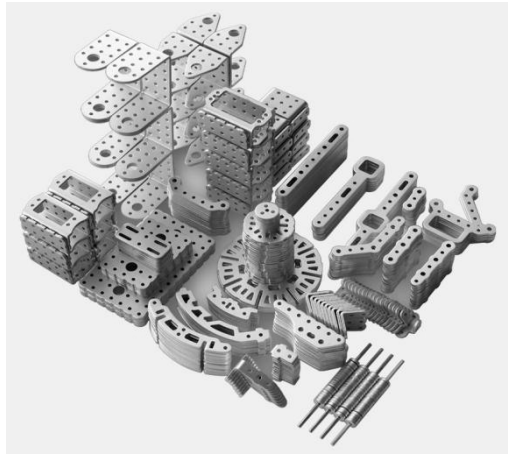
请大家阅读设备里附带的装箱单，清点设备里的零部件，在清点的过程中，我们会认识这些零部件的名字、外观、位置、大致用途，为后面的内容做好准备。

 C13 Bbers控制板 2	 C49 BigFish2.0 3	 C44 Mehran控制板 1	 C50 Birdmen扩展板 1	 C01 触碰传感器 3	 C02 胡须传感器 2	 C06 闪烁传感器 1	 C07 声控传感器 1	 C51 语音识别HBR740 1	 C17 白标传感器 2
 C20 颜色识别传感器 1	 C22 摄像头 1	 C03 光谱传感器 1	 C33 灰度传感器 3	 C26 近红外传感器 4	 C28 加速度传感器 1	 C34 超声波传感器 1	 C35 温湿度传感器 1	 C39 编码器 2	 C10 LED模块 2
 C11 语音模块 1	 C40 OLED 1	 C31/C46 蓝牙串口套装 1	 C30 NRF无线模块 2	 C18 无线路由器 1	 M01 标准伺服电机 8	 M04 大标准伺服电机 4	 M06 双轴直流电机 10	 P01 扳手 3	 P06 螺丝刀 3
 P07 内六角扳手 3	 P13 镊子 3	 P09 排线_4芯输出 4	 P10 排线_4芯输入 12	 P15 1拖2直流扩展线 2	 P20 单芯杜邦线 20	 P25 USB线 3	 C50 2510通信转接板 1	 P02 充电器 1	 P03 锂电池 3
 A10 橡胶垫 8	 A01s 偏心轮3mm 6	 A02s 挡片偏心轮 4	 A03 马达后盖输出头 8	 A04 30齿齿轮 4	 A04s 随动齿轮 4	 A05 输出头 8	 A06 直流电机输出头 10	 A07 大后盖输出头 4	 A08 大输出头 4
 D03 防滑垫 100	 A16 履带片 100	 P22 1:10模型轮胎 6	 P24 联轴器 6	 A19 硅胶轮胎 8	 J01 10mm滑轨 2	 J02 3x5双折面板 8	 J03 5x7孔平板 3	 J04 7x11孔平板 4	 J05 90度支架 12

二、机械零件认知

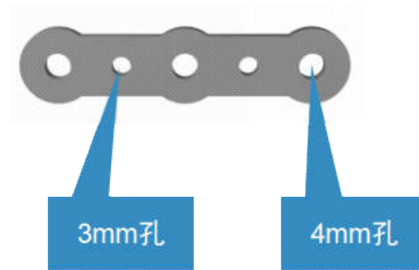
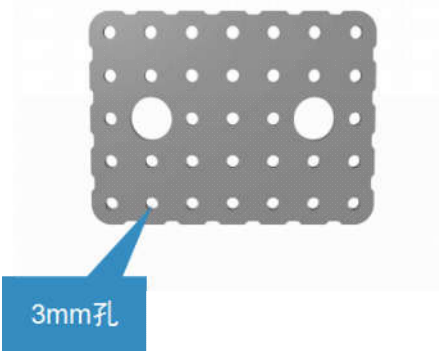
“探索者”的主要零件采用铝镁合金材质，零件上开有通孔，用不锈钢螺丝螺母组装，如图 2.1 所示。孔的作用是约束零件。常见的零件孔有直径 3mm 的孔和 4mm 孔，常见孔的中心距为 10mm。“探索者”的零件孔可看做几何中的“点”，是空间几何造型的基本单位。另外，还有提供特定机械运动的各种方孔、大方孔、长圆孔、大圆孔、异型孔等。

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。

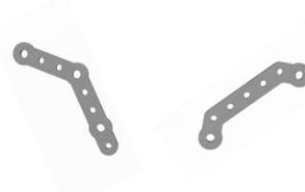


2. 连杆类零件

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”。



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。



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。



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。



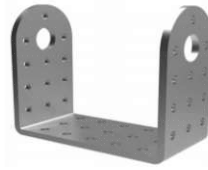
直流马达支架



马达支架



大舵机支架



舵机双折弯

大舵机U型支架

轮胎相关零件

轮胎需要联轴器才能和电机的输出头相连，如下图所示。



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 $\phi 3$ 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。

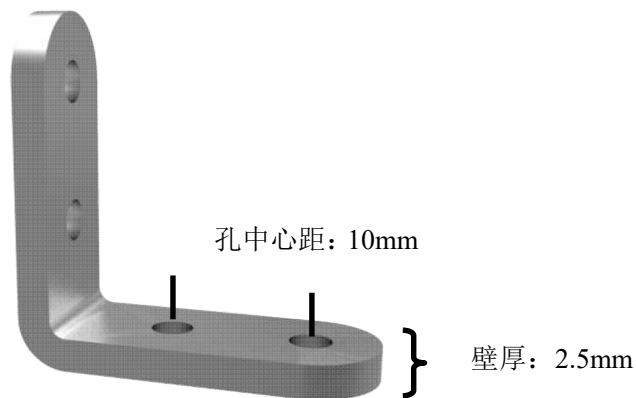
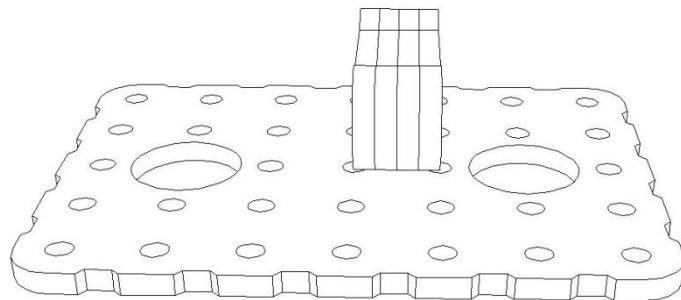


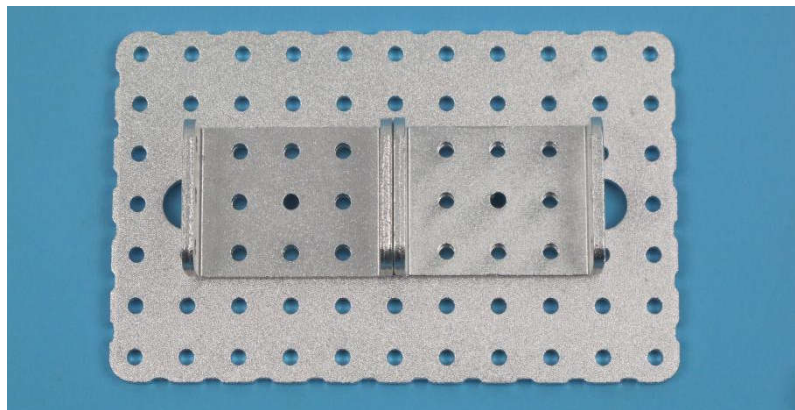
图 2.10 零件的壁厚与孔距

也就是说，四个零件叠加的厚度，正好等于两个孔的中心距。



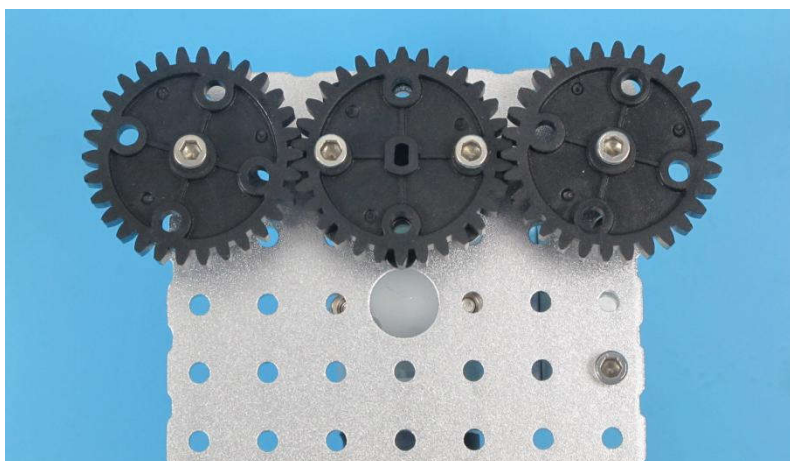
零件空间关系示意图

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。



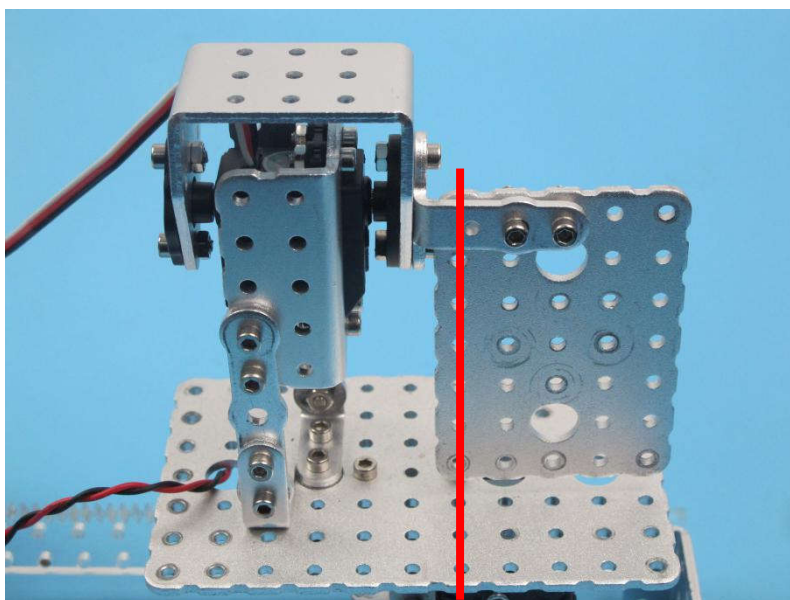
零件孔位置配合关系

齿轮等辅助性零件也遵循这个规律而设计。



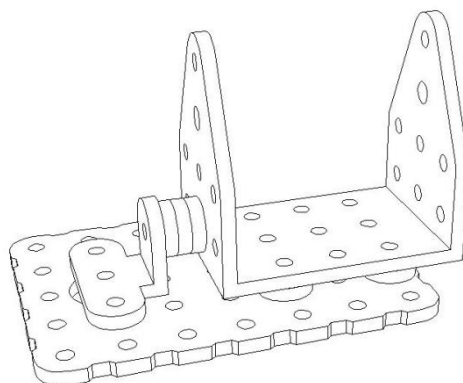
齿轮位置配合关系

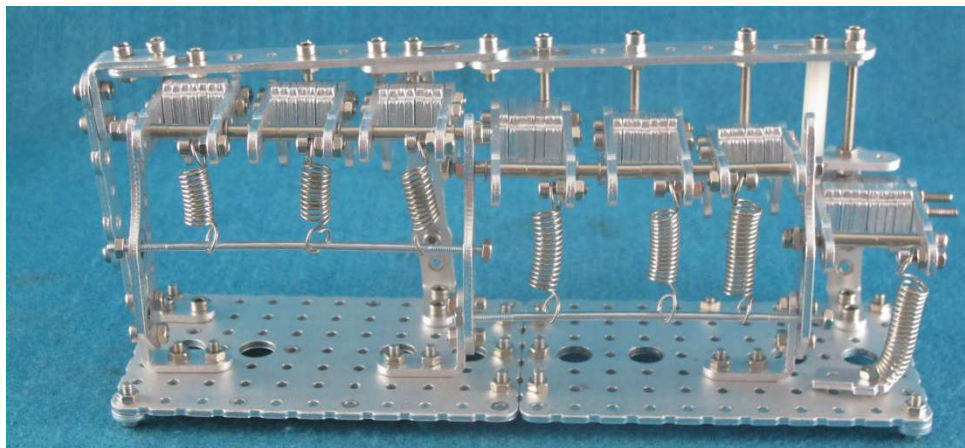
对电机及其相关部件同样适用。



电机位置配合关系

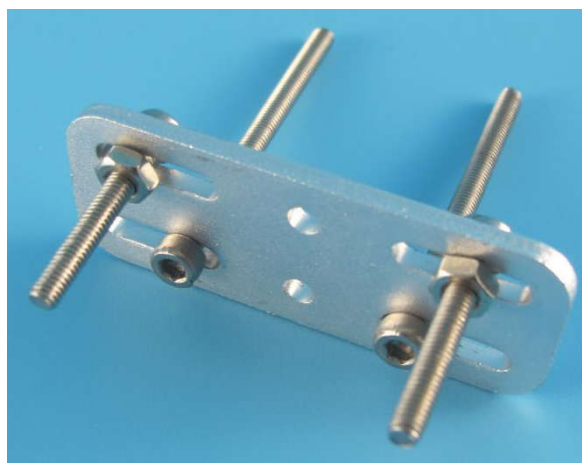
这个原则保证了“探索者”零件系统在空间造型和传动设计的工作中，无论构型多么复杂，总能够良好匹配。因此，如果遇到两个零件孔无法匹配的状况时，我们只需要利用零件、螺母、螺柱、轴套等补充出 2.5mm 整数倍的距离，就可以匹配了。





“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。

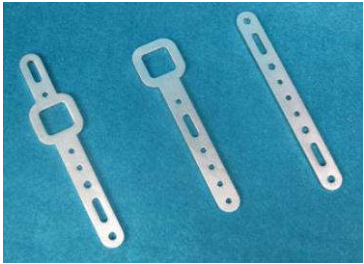
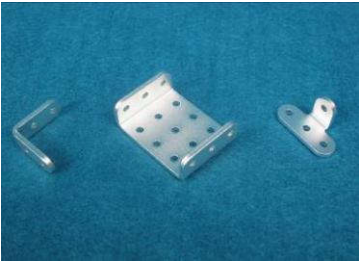

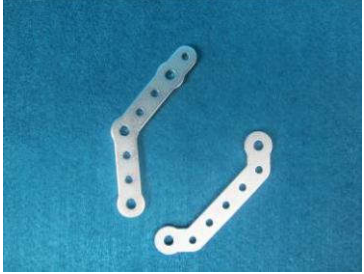

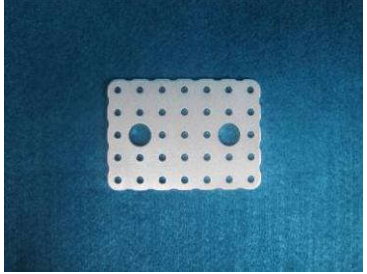
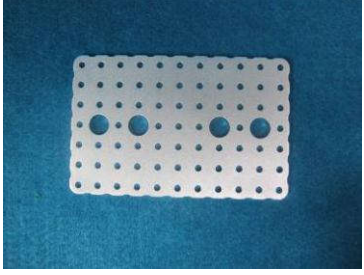




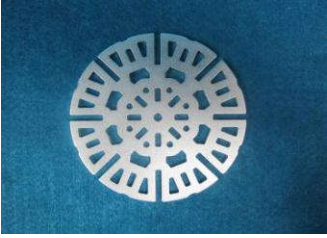
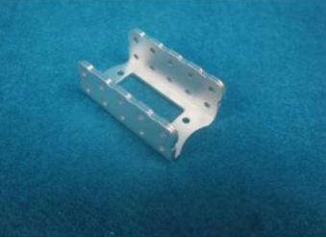


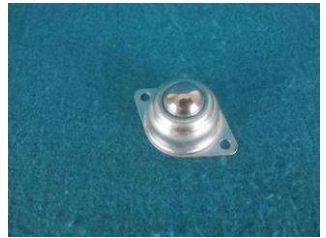
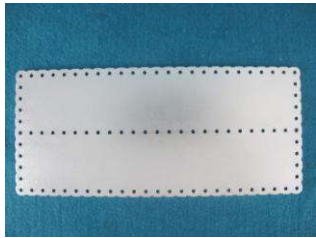



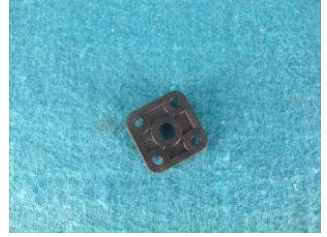


借助可调节孔的零件做转接



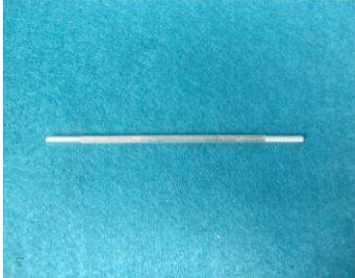
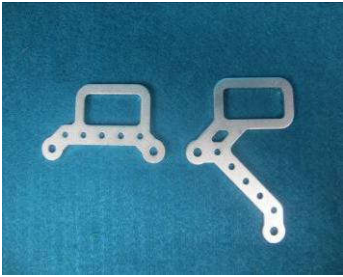





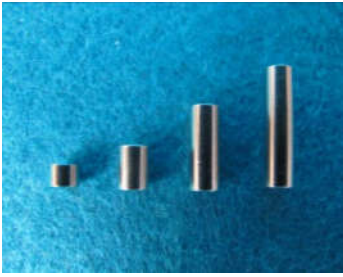
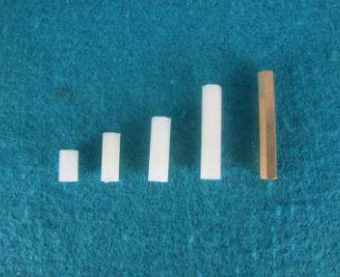

“探索者”的核心零件，即连杆类、平板类和框架类三种，多年来有增有减，但数量始终保持在 30 种左右。也就是说，与其他积木类创新组件不同，“探索者”零件库其实很小，它采用的是类似“七巧板”的创新思路，可以用有限的、精选的、通用性极强的零件，搭建千变万化的结构。

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。

常用零件列表

		
<p>双足腿、四足连杆、双足支杆</p>	<p>90 度支架、3×5 折弯、输出支架</p>	<p>机械手 20mm、机械手 40mm、机械手 40mm 驱动</p>
<p>曲柄滑块机构的主要零件，可用于搭建机器人行走机构。</p>	<p>三种折弯件，可搭建机构支架，连接不同平面</p>	<p>三种长短不同的连杆，可搭建四连杆结构，伸缩机械手等</p>
		
<p>机械手指、双足连杆</p>	<p>垫片 10、垫片 20、轮支架、10mm 滑轨</p>	<p>5×7 孔平板</p>
<p>两种带角度的连杆，可搭建机械手爪，腿部结构等。</p>	<p>四种小金属件，主要起调节机构层次的作用</p>	<p>长边 7 个孔，短边 5 个孔，可用作小型搭载平台</p>
		
<p>7×11 孔平板</p>	<p>小轮</p>	<p>舵机双折弯</p>
<p>长边 11 个孔，短边 7 个孔，可用作大型搭载平台</p>	<p>可用作履带、滚筒的骨架</p>	<p>可用作机器人关节摆动部件</p>

		
大轮	马达支架	大舵机支架
可用做大轮子，机架、半球结构、球结构等	小型舵机使用，连接小型舵机与其他零件	连接大舵机与其他零件
		
大舵机 U 型支架	牛眼万向轮	11×25 孔平板
用于大舵机组装关节式结构	国际标准零件	可用作大型机架平台
		
输出头	马达后盖输出头	大舵机输出头
		
大舵机后盖输出头	直流马达输出头	履带片

		
三种球形件	直流马达支架	传动轴
可用于翅膀、腿、轮足等 仿生机构的搭建	直流马达的支架，连接直流电机与其他零件	不锈钢传动部件，可 连接齿轮等，两端是扁的
		
双足大腿、双足小腿	双足脚	两种偏心轮
可组装特殊的曲柄滑块， 用于机器人行走机构。	可作为脚使用， 也可用于其它功能	可组装偏心轮机构，代替凸轮，代替曲柄等
		
30 齿齿轮	随动齿轮	联轴器、1:10 模型轮胎
		
不锈钢轴套 轴套 2.7、轴套 5.4、轴套 10.4、轴套 15.4	国际标准尼龙螺柱 螺柱 10、螺柱 15、螺柱 20、螺柱 30； 35mm 金属螺柱	国际标准 M3 不锈钢螺丝、螺母

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。

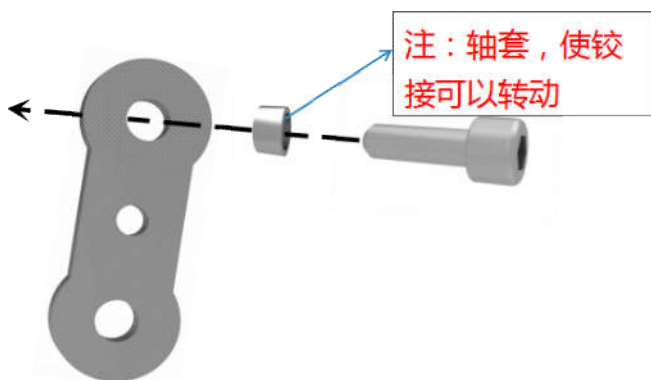
最基础的刚体组装至少需要 2 颗螺丝。这对应了“经过两点有一条直线, 并且只有一条直线”的几何定理。刚体连接一般利用 3mm 零件孔。



请大家取两个零件，用螺丝将他们固定在一起。

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。



请大家取两个杆件，组装成一个铰链结构。

第二课 简单结构的组装

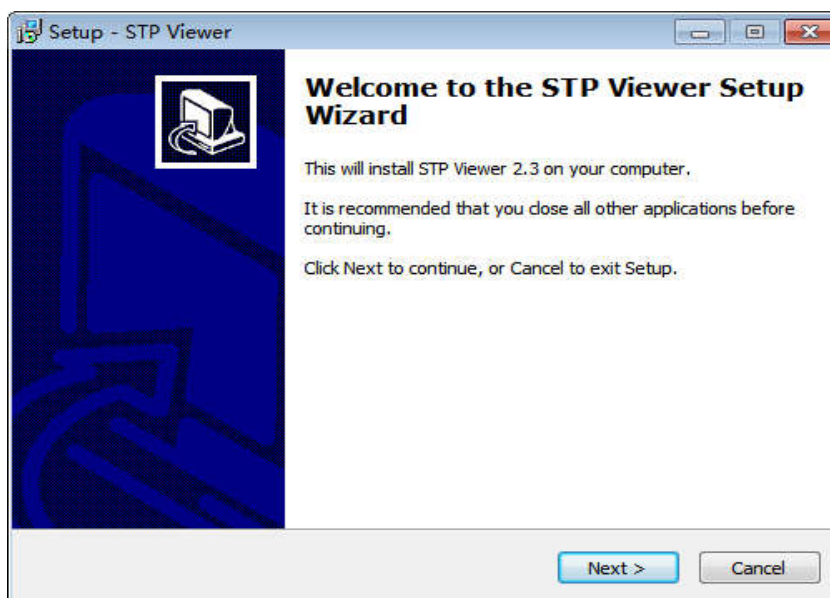
实验一 使用三维软件辅助组装

1. 认识 STPViewer 软件

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

找到 STPViewer setup.exe 文件，双击安装 STPViewer

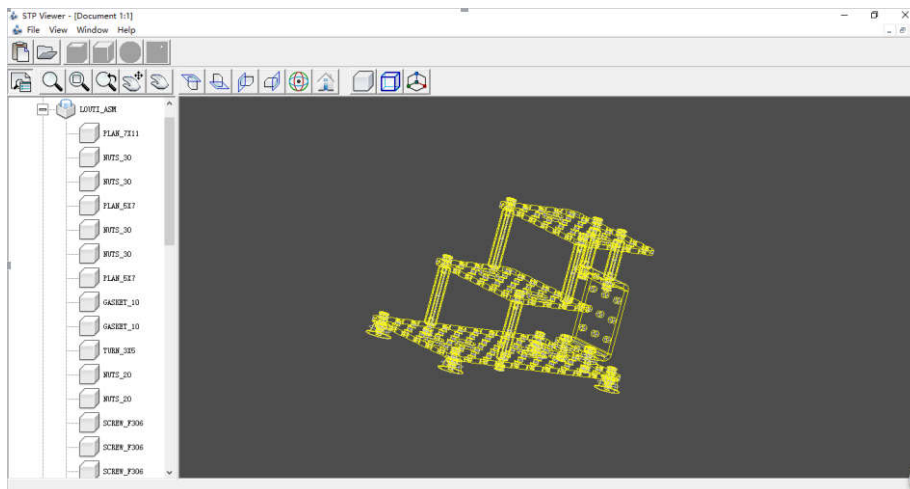


3. 浏览文件

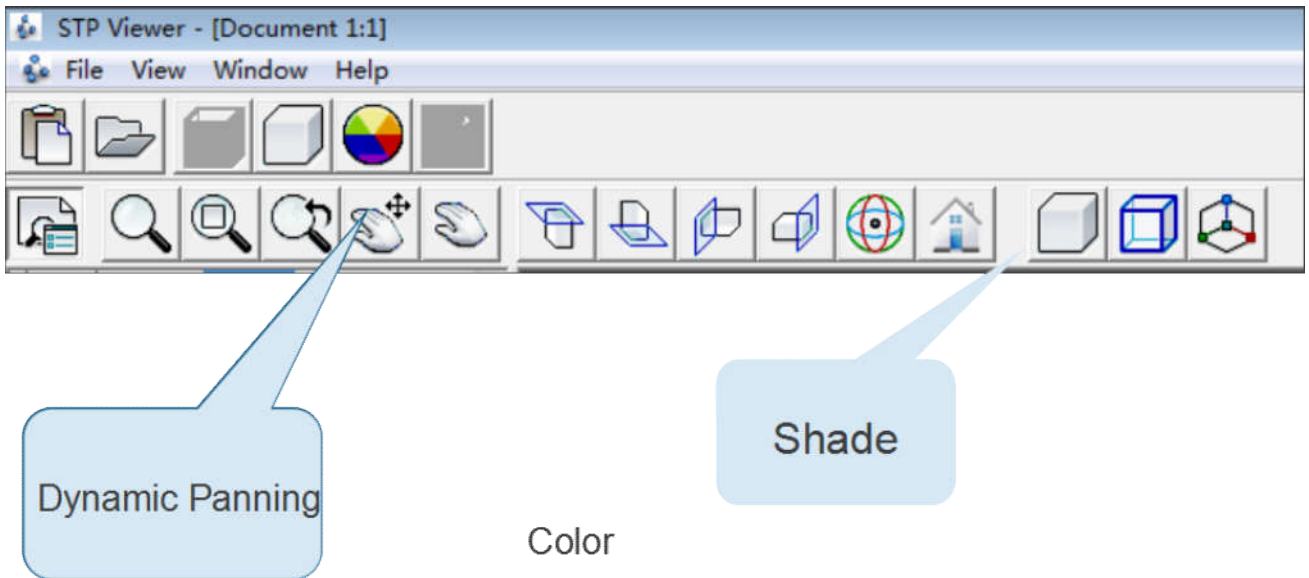
找到名为“STP-class1”的文件夹，将此文件夹拷贝至某盘根目录。

在“STP-class1”文件夹中找到“stairs.stp”文件，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。）。

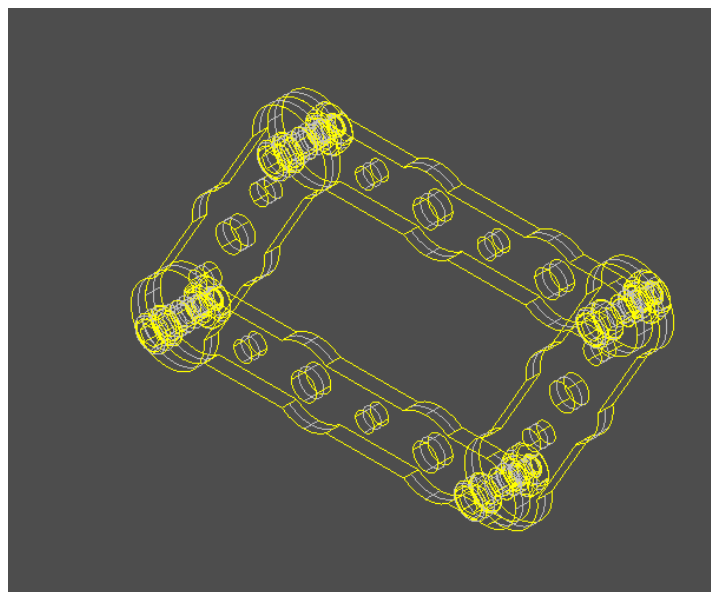
打开文件后，首先看到的是 3D 线框图



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。

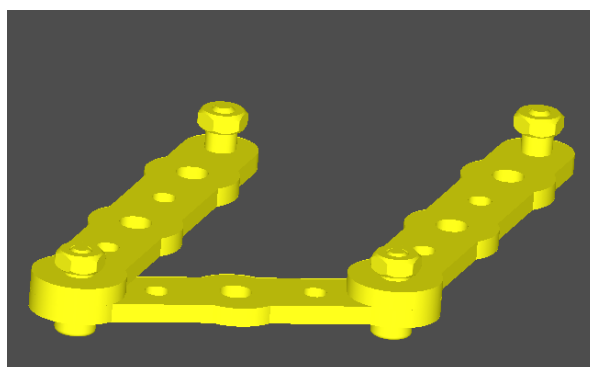


4. 在“STP-class1”文件夹中找到“quadrilateral.stp”文件，双击打开。



5. STPViewer 还有一个“隐藏零件”的功能：选中一个“机械手 40”，然后可以看到在软件界面左侧零件树列表中，对应的零件名也被选中了。

在该零件名上点击右键，选择“hide”。“机械手 40”即可隐藏，里面的轴套就能看到了。



实验二 组装滑板小车

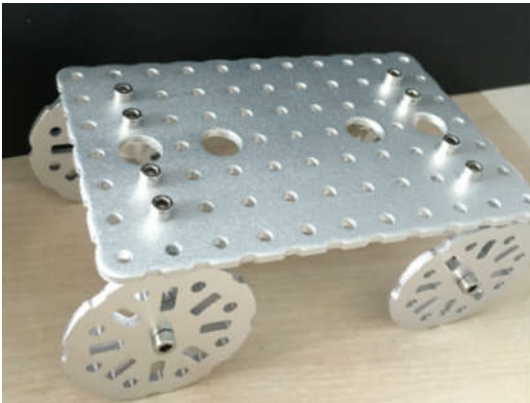
实验目的：1. 学会使用 STP Viewer 看图；2. 综合练习固定、铰接的方法；3. 巩固组装过程中的要点，如层次、螺丝、轴套等的长度选择；4. 孔位的选择与干涉的避免；5. 机构外观造型与功能的协调设计。

实验性质：验证型实验

实验课时：1 课时

实验内容：

找到“smallcar.stp”文件，参照该 stp 图，组装平板车。请自己寻找相应的零件，螺丝长度不用拘泥，自己选择合适的即可。



提示：四个轮子都是铰接的，能够灵活转动，放在桌面轻推即可前行。

实验三 组装双轮万向小车

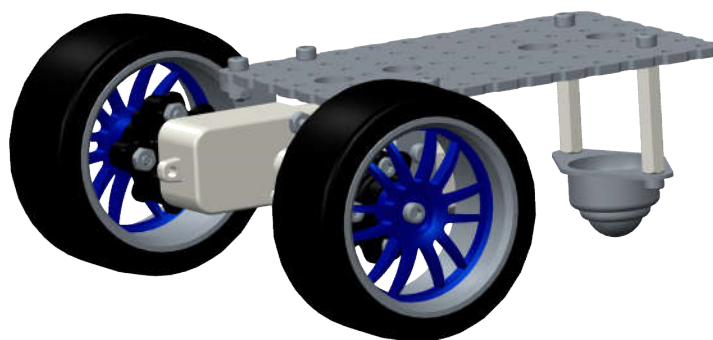
实验目的：1. 掌握驱动模块的组装方法；2. 了解模块化组装机器人；

实验性质：验证型实验

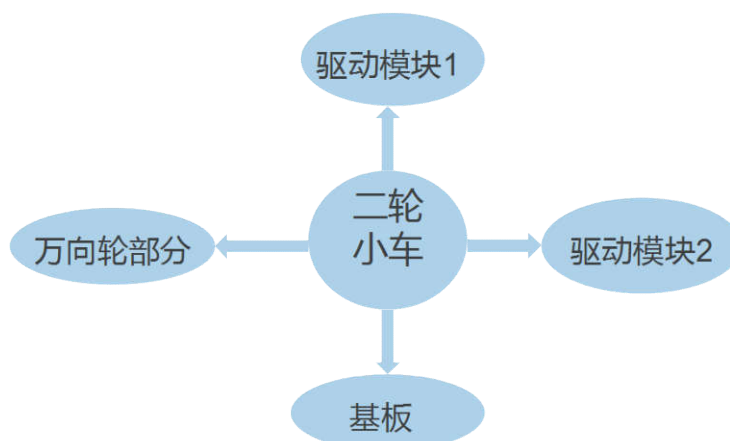
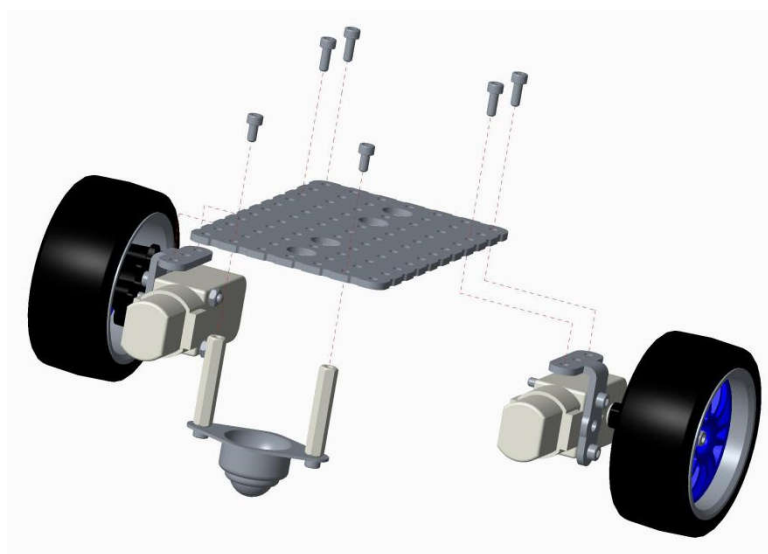
实验课时：1 课时

实验内容：

1. 首先**观察**一个二轮驱动小车模型的图片：



2. 将一个二轮驱动的小车拆分不同的模块：



3. 根据以上拆分，所以组装一个双轮小车，将做以下步骤：

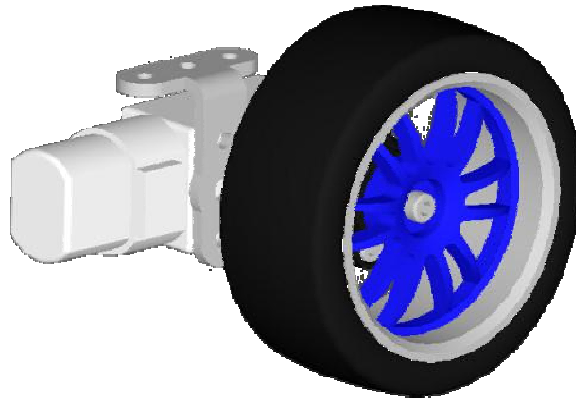
双轮小车模型目测拆分为：两个驱动轮模块+底板+ 万向轮部分；

分别组装各个模块—整合模块—完成；

应用：整机—目测分解为模块—分别组装各个模块—整合模块—完成；

（提示：模块均以电机为中心，所以每次都可以从电机开始着手组装）。

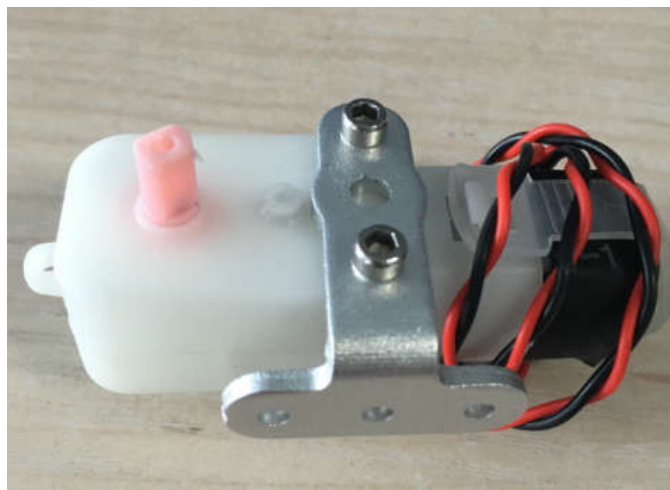
4. 组装一个驱动轮模块：



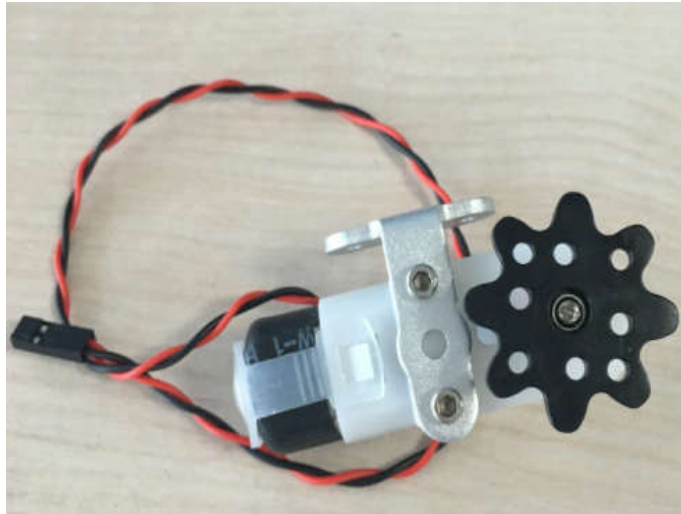
5. 驱动模块所需器材：直流电机、直流马达输出头、直流电机支架、轮胎、螺柱 15、联轴器、螺丝 F325、螺丝 F310、螺丝 F2510H、螺母。



6. 支架安装：使用直流电机支架、螺丝：F325：



7. 输出头安装：安装于直流电机粉色端，中心需安装 F2510H:



8. 轮模块组装：在联轴器里放置一个 15mm 的螺柱:



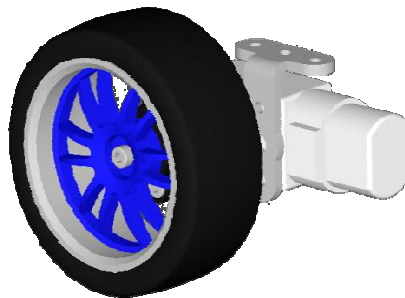
9. 轮模块组装：联轴器与直流电机输出头锁死，使用螺丝 F310:



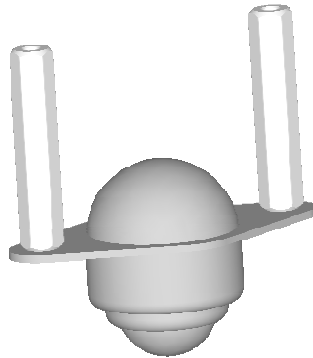
10. 锁上轮胎:



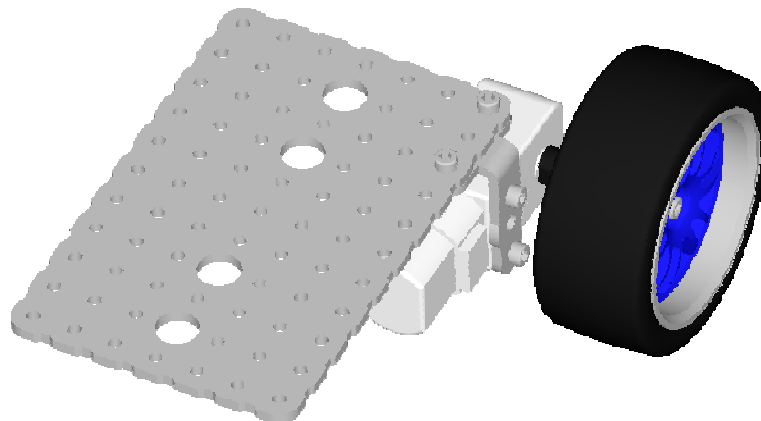
11. 再组装一个对称的驱动轮模块:



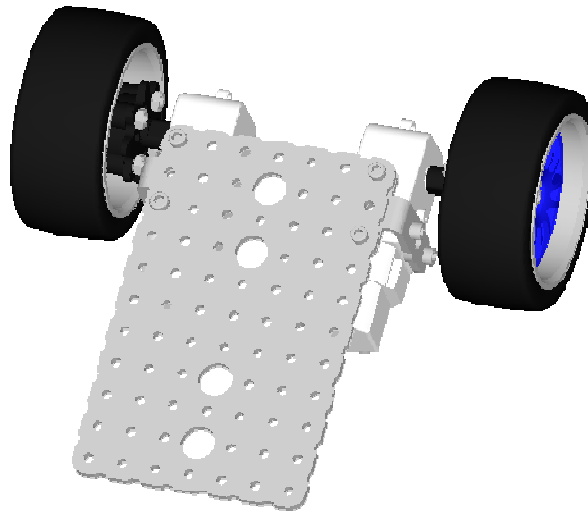
12. 找到牛眼万向轮, 选择 6mm 螺丝将两个 30mm 长的螺柱与牛眼万向轮组装在一起:



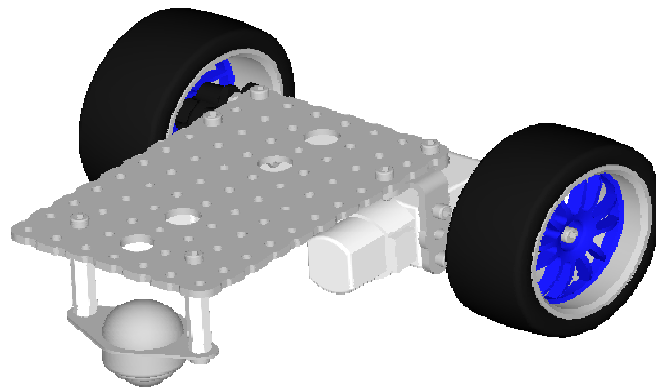
13. 选择一块 7×11 孔的平板, 利用 2 个 8mm 的螺丝固定一个驱动轮模块:



14. 在平板对称位置重复上一步骤，再固定一个驱动轮模块：



15. 最后在平板的边缘中心位置选择合适的空位将万向轮模块组装完成（提示：螺丝长度的选择只要长度合适，不影响结构运动，均可使用）：



实验四 组装创意造型

实验目的：1. 熟悉探索者零件的使用；2. 锻炼学生创新能力；

实验性质：设计型实验

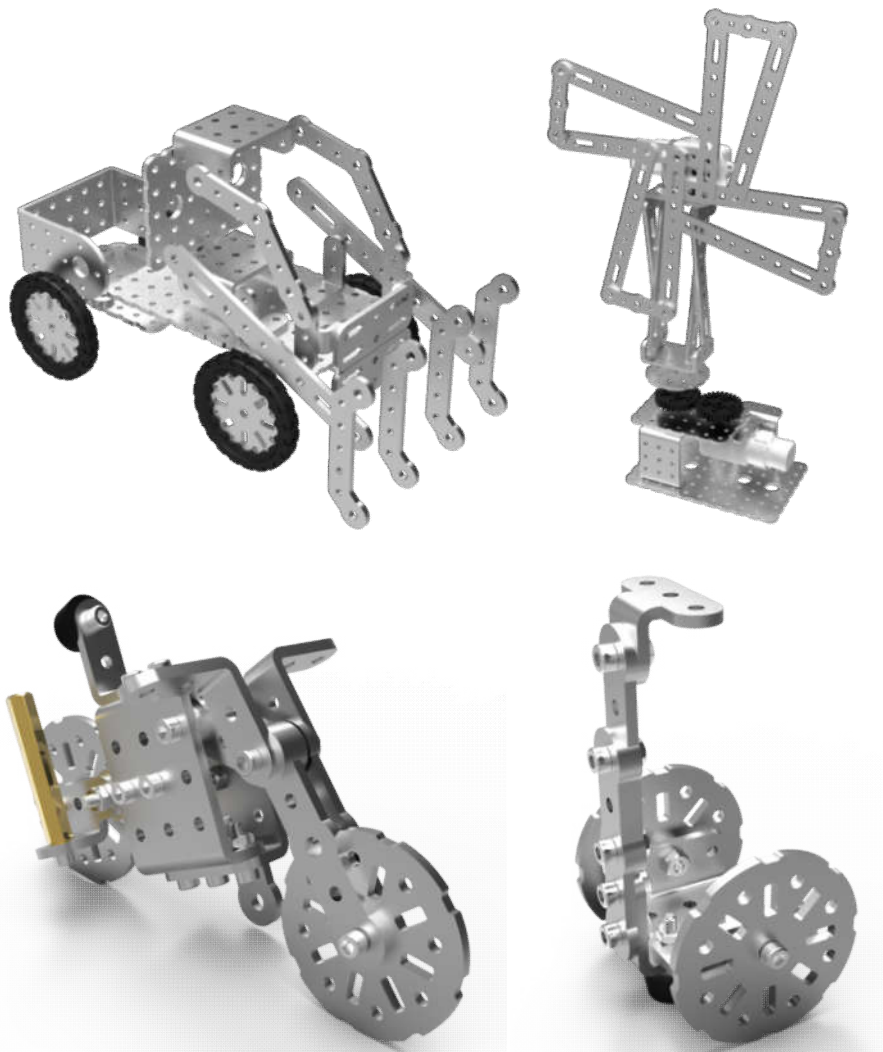
实验课时：1 课时

实验内容：

根据之前学过的内容，请从以下题目中选择一些进行搭建，你可以参考图片，也可以自己发挥，搭建简单的积木造型，不用带电机。

- (1) 家具造型：桌、椅、板、凳、柜子、带合页的门、安全门等；
- (2) 几何造型：立方体、棱柱、圆柱、梯形体、雪花体、八边形体等；
- (3) 轮类造型：滑板、轮滑鞋、自行车、汽车、飞机、拖拉机、挖掘机等；
- (4) 机械造型：等臂四杆、不等臂四杆、六杆、摆角放大机构、曲柄滑块等；
- (5) 工具造型：手机支架、开瓶器、书本支架、笔筒、抽屉锁、梯子等；
- (6) 武器造型：刀枪剑戟斧钺勾叉等；枪械、坦克、火炮等；
- (7) 动物造型：狗、鸟、恐龙、鱼、蛇、昆虫等；
- (8) 建筑造型：平房、楼房、桥梁、风车等。

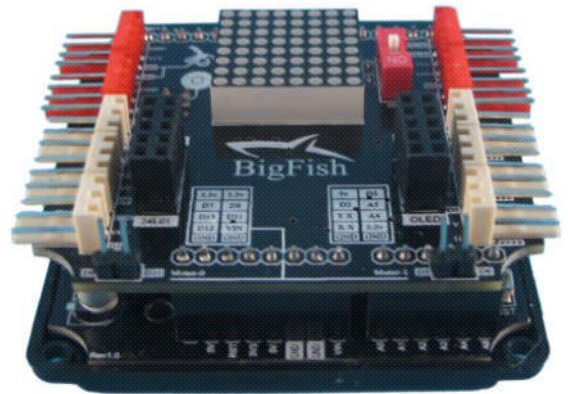
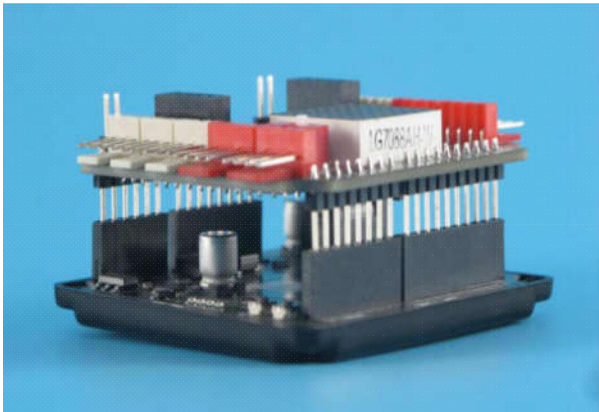
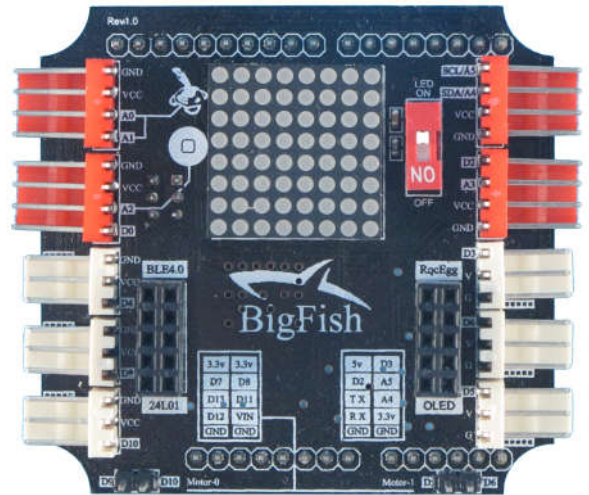
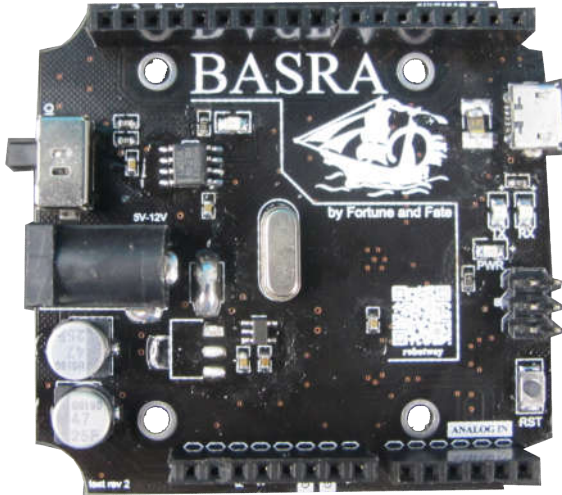
参考示例如下图：



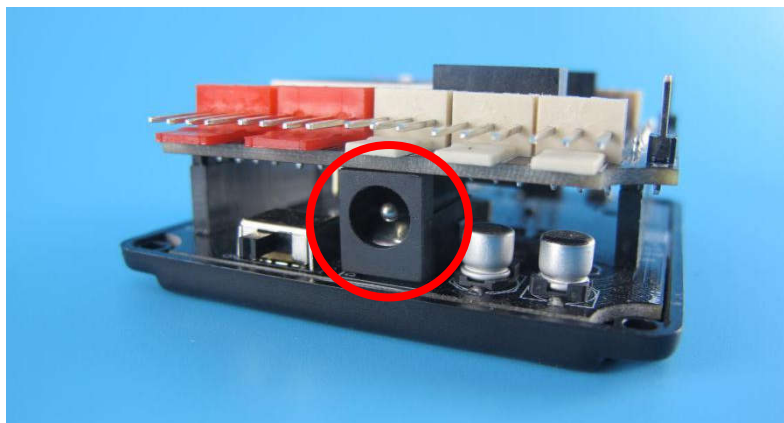
第三课 控制的基础知识

电路连接

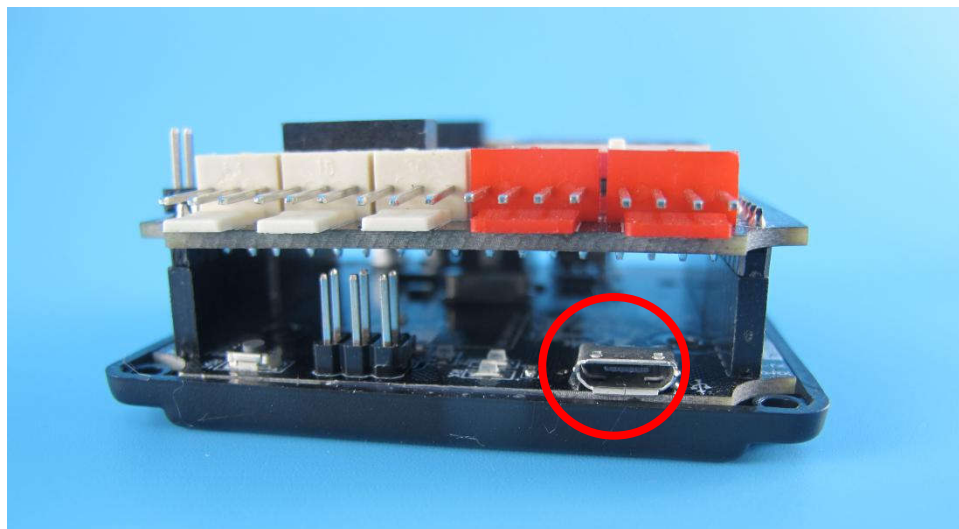
1. Bigfish 扩展板与 Basra 主控板堆叠连接，如下：



2. 与电池连接：找到锂电池，接在 Basra 的电池接口上（使用时打开电源，不使用时记得关闭）：



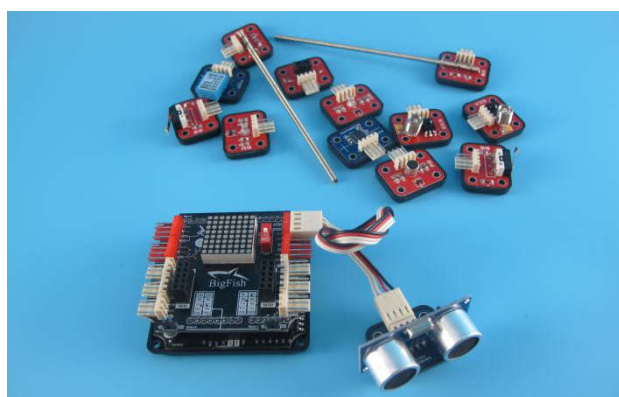
3. USB 连接：下载程序时，需将 USB 线接在 mini USB 口上：



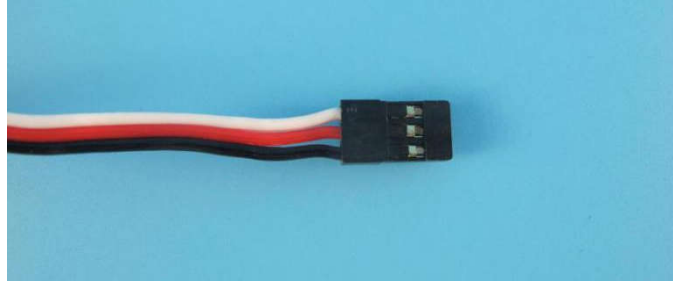
4. Bigfish 扩展板与常规传感器连接：找到 4 芯输入线：



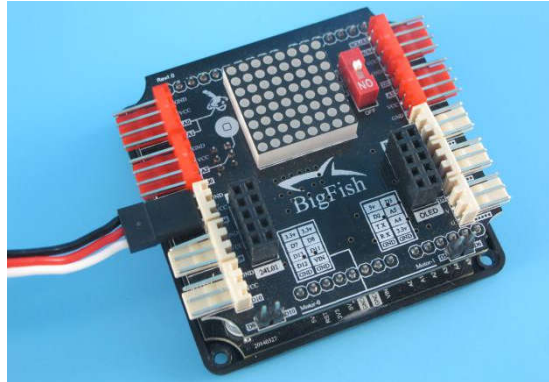
5. Bigfish 扩展板与常规传感器连接：4 芯输入线一端连传感器，另一端连在 Bigfish 红色 4 针接口上（一般可连接 4 个）：



6. Bigfish 扩展板与伺服电机连接：伺服电机都是 3 根线，黑色为地线（GND），红色为电源线（VCC），白色为信号线（D*）：



7. Bigfish 扩展板与伺服电机连接：伺服电机可接在白色 3 针伺服电机口上，注意观察接口上的 GND 针的位置，不要插反了，简单来说，露出金属的那一面朝下：



实验五 配置编程环境实验

实验目的：学会配置 Arduino 编程环境, 为编程做准备；

实验性质：验证实验

实验学时：1 课时

实验内容

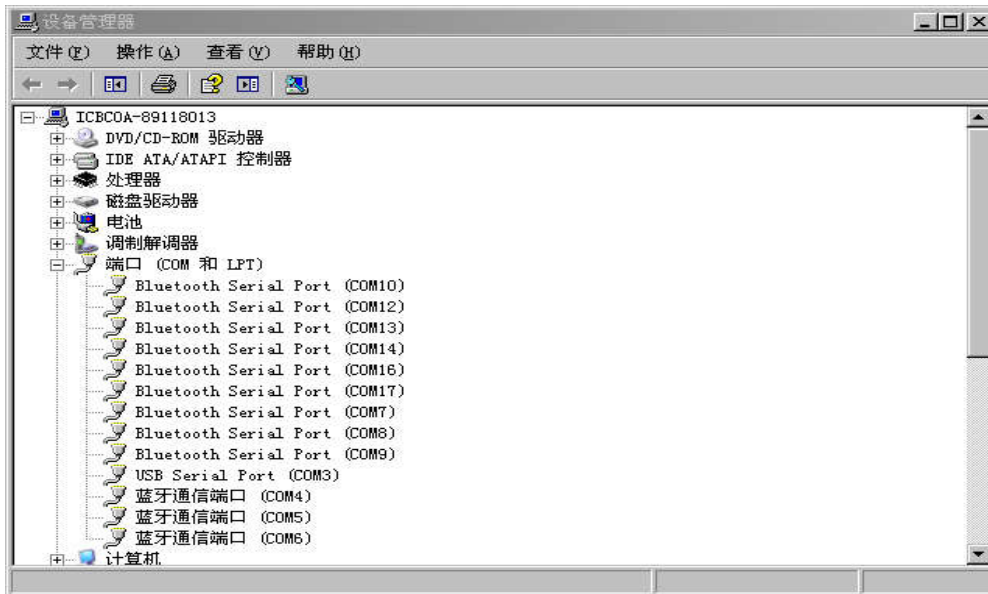
安装编程环境-Basra 控制板

步骤一：拷贝 `arduino-1.5.2` 目录至本机位置。

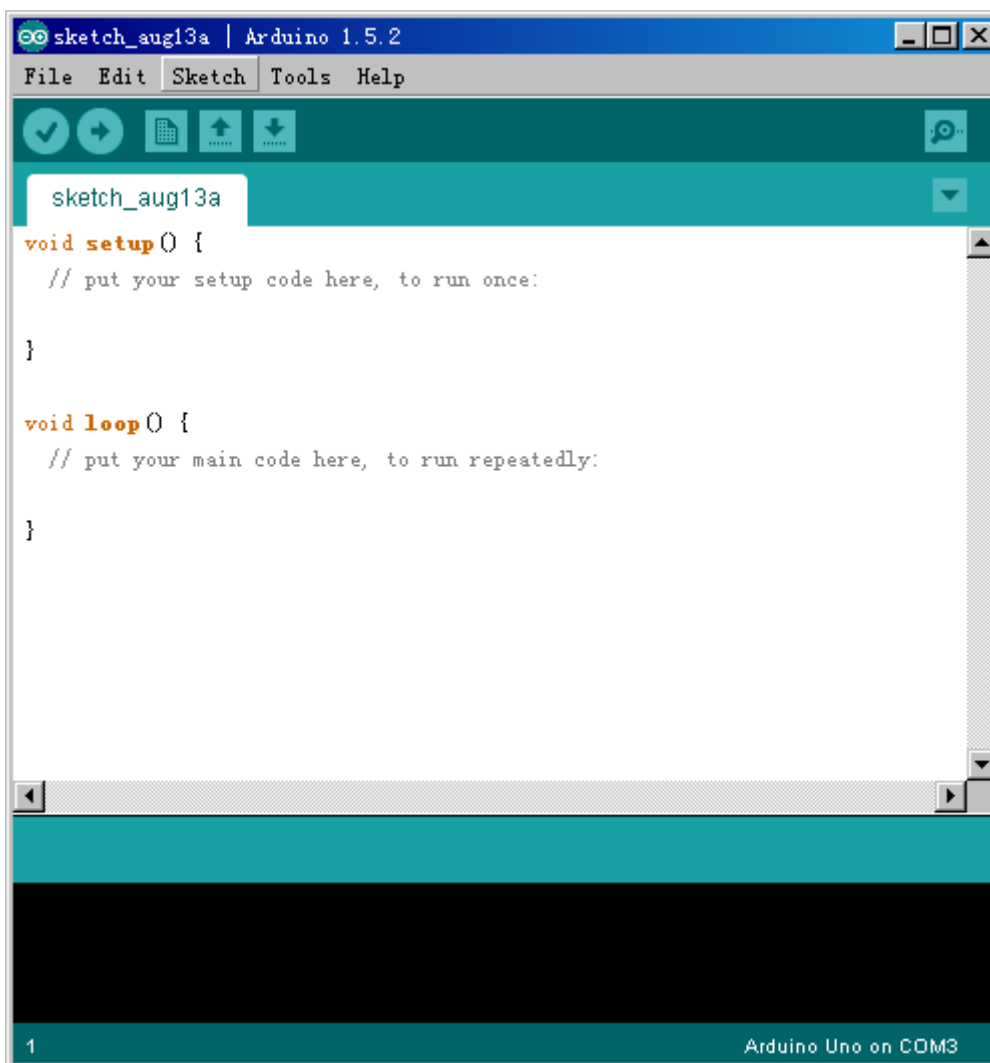
步骤二：将 Basra 控制板通过 miniUSB 数据线与 PC 连接，初次连接时会弹出驱动安装提示。选择 `arduino-1.5.2\drivers\FTDI USB Drivers` 目录安装驱动。



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。

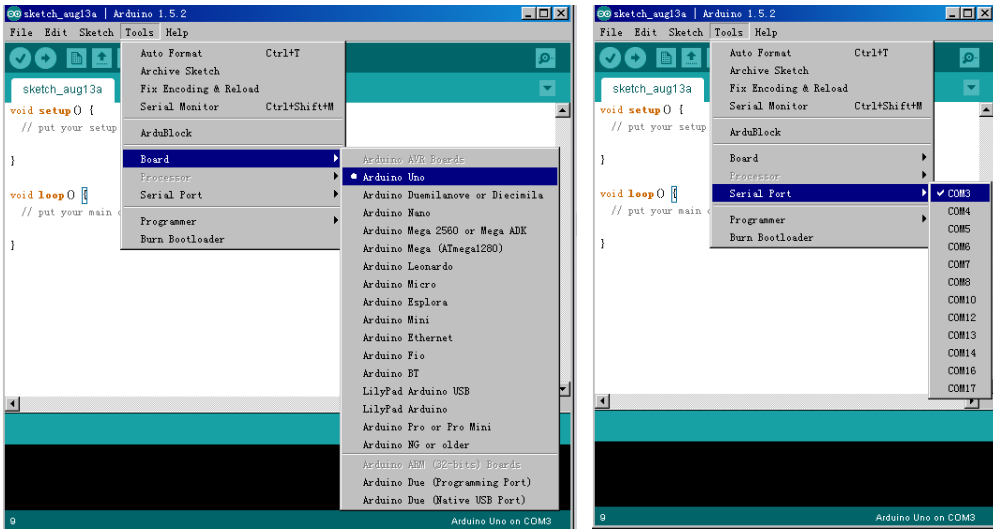



步骤四：在本机上运行 arduino-1.5.2 目录下的 `arduino.exe`，显示如下界面：



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)

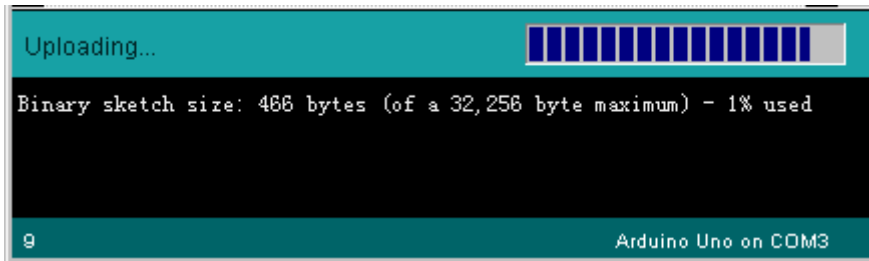
为步骤 3 里记录下的端口号)。此时在界面右下角显示 Arduino Uno on COM3。



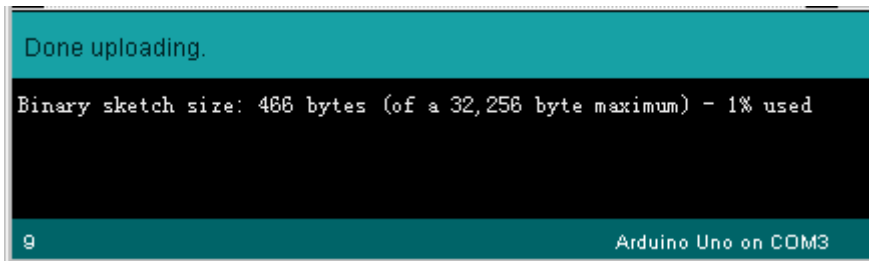
步骤六：点击 upload 按钮 ，一个空白的程序将自动烧录进 Basra 控制板。具体过程如下图所示：



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

实验六 blink

实验目的：学习使用延时语句，理解程序的顺序执行

实验性质：验证型实验、设计型实验

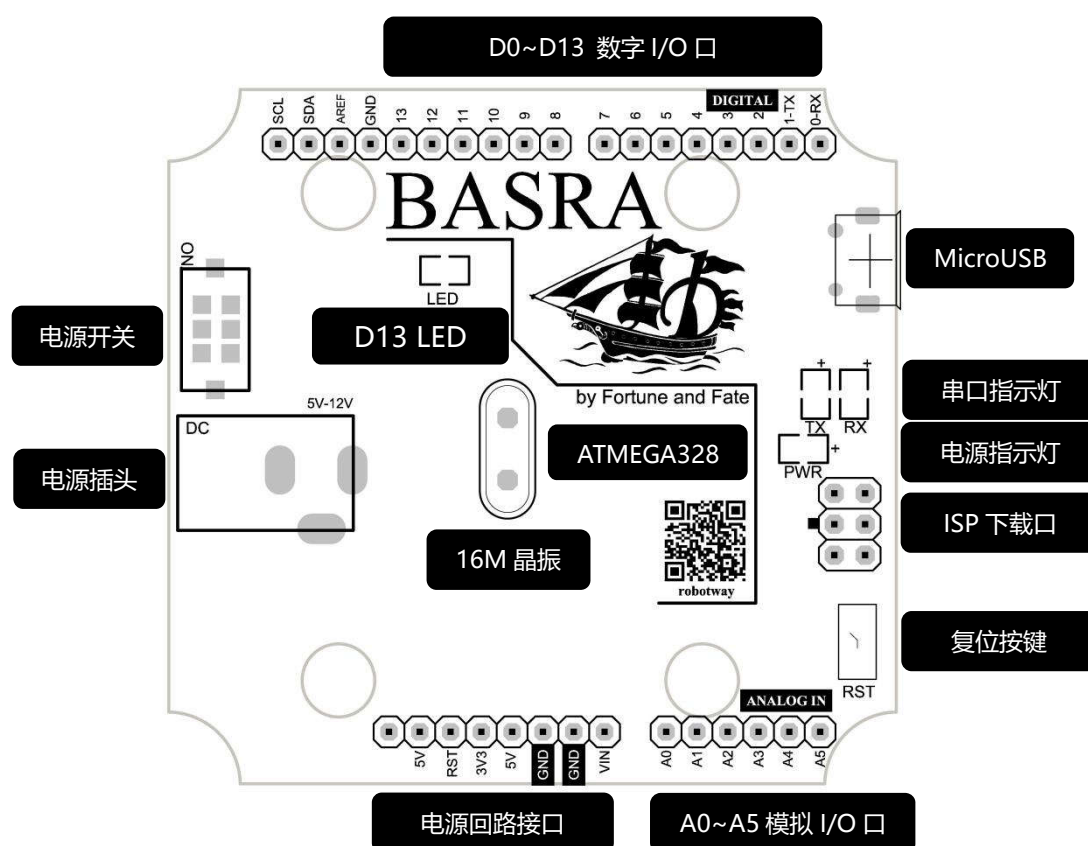
实验课时：1 课时

实验内容：

摘掉 Bigfish 扩展板，只使用 Basra 控制板。

1. 13 号端口 LED 灯橙色闪烁

在下图找到 D13 端口，并在 Basra 主控板上找到对应的 LED 灯。



从图形化界面左侧菜单栏拖选语句图形至右侧，完成下图所示语句



烧录，观察这段程序实现的效果是什么。

在这段程序中，我们使用了”延迟“语句，设定的参数是 1000，也就是 1000 毫秒，或者说 1 秒。”延迟 1000 毫秒“的意思**不是**说”1 秒之后再做“，而是说该语句上面所设定的状态，即”针脚 13 置高“，要保持 1 秒。这样，我们就能理解这段程序的意思了。

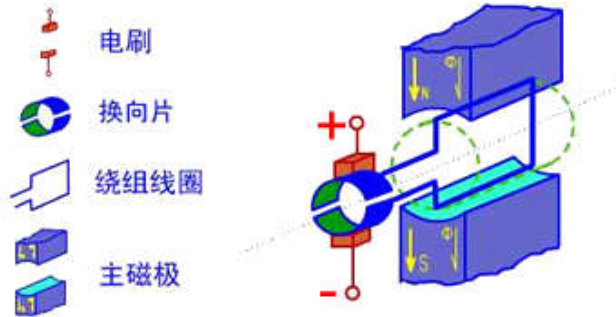
和传统的 C 语言一样，程序是从上到下顺序运行的，然后不断地循环。

2. 接下来请大家实现改变闪烁的频率。

实验七 电机入门操作

在进行电机实验之前我们先了解下该创新套件中直流电机的参数和控制方法。

直流电机是利用电磁感应工作的执行器，通过给线圈通电，使其能够在两片磁铁之间旋转。

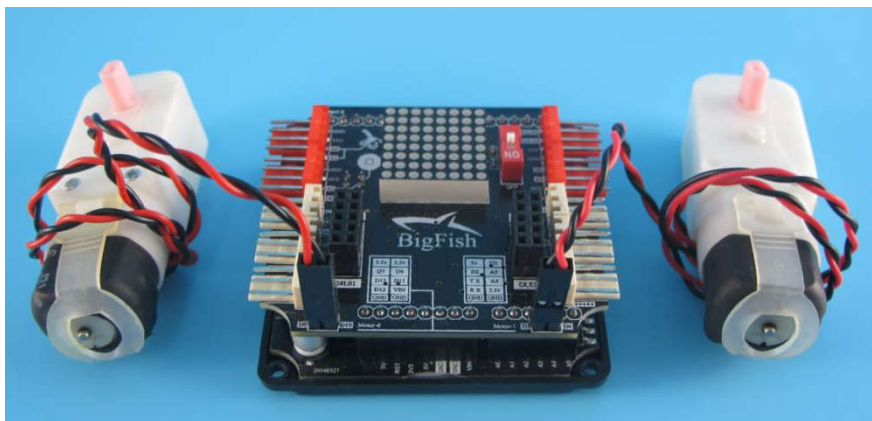


直流电机是制作驱动轮的常见部件，同时可以驱动一切圆周运动的机构，如曲柄摇杆、曲柄滑块等。这种直流电机（图 2.1）已经加装了减速箱，减速比为 87，速度较快，力量较小，参数见表 2.1。



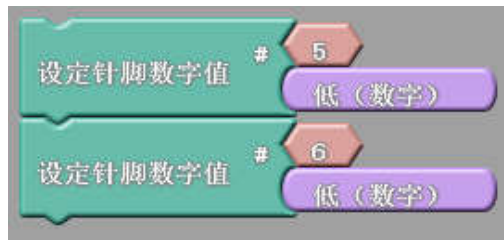
直流电机参数

减速比	额定电压	额定电流	扭力	RPM
87	4.5v	180mA	5kgf · cm	69



直流电机连接扩展板

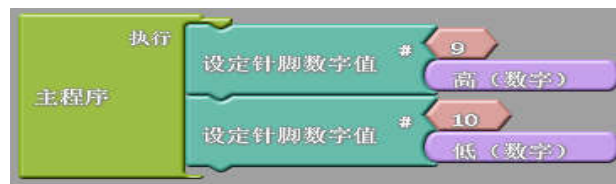
使用的图块为：



函数为 digitalwrite() 图块

由于直流电机有两个针脚，所以要写出两个图块（程序简单的话只写一个也行），比如 Bigfish 有个直流口两根针分别接主控板的 D5/D6，那么程序就可以有“5 高 6 低”、“5 低 6 高”、“5 低 6 低”，分别对应“转”、“反转”、“停”。而“5 高 6 高”这种模式是刹车功能，实际使用时很少用到。因此使用的时候只要认清针脚号就可以了。

再比如 Bigfish 扩展板有两个直流口两根针分别为 D9/D10，那么控制程序可以这么写：



D9/D10 控制程序图块

还有一个语句是“设定针脚模拟值”，其实就是设置针脚的电压参数，这个参数的范围是 0~255 之间，0 相当于“低”，255 相当于“高”，可以取它们之间的数值，说明可以改变直流电机的速度。我们可以把 D6 设置成低电平，而 D5 供电参数设为 200，那么电机的转速将会变慢。

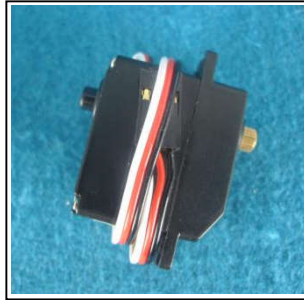


D5/D6 控制程序图块

实验八 伺服电机入门操作

简介

“探索者”器材中常见的伺服电机，是常用于航模舵位控制的小型伺服电机，一般也被称作“舵机”，标准舵机如。



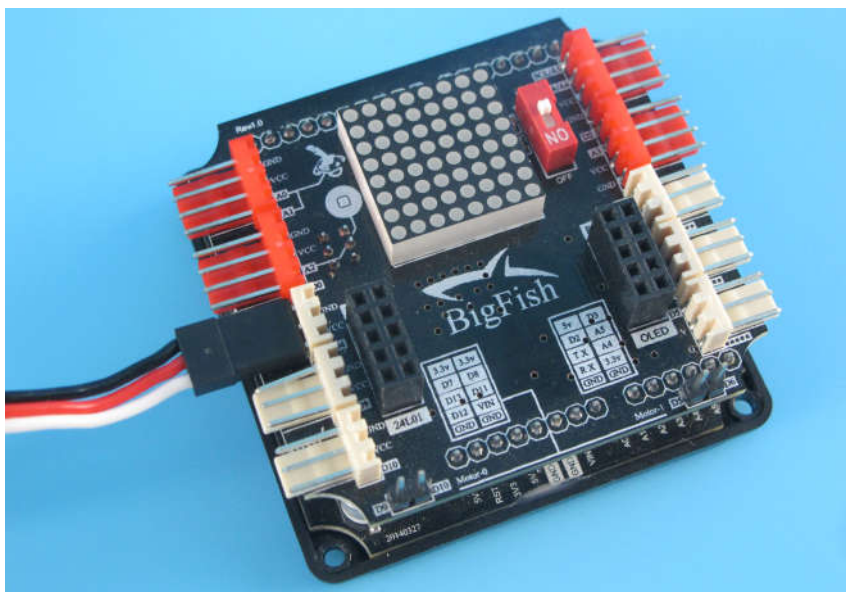
小型伺服电机
小型伺服电机参数

	速度	扭力		转动角度	额定电压
	Sec/60°	kg · cm	oz · in		V
标准舵机	0.13	2.9	40.3	±90°	6

标准舵机的用途是制作摆动机构，如关节模块等。

从器材中找到：Basra 主控板、Bigfish 扩展板、USB 数据线、标准伺服电机和输出头。并把输出头安装在电机输出轴上，注意安装的时候要找准中间位置。可以左右拧动输出头来确认中间位置，左右可拧动的幅度基本一致即可。

将主控板连上 PC，将伺服电机接在白色 3 针伺服电机口上，注意观察板子上的针脚名称（下图接口针脚号为 D4），不要插反了，简单来说，露出金属的那一面朝下。



舵机的控制方法

Basra 编程时，舵机控制如下。



舵机控制图块

我们可以发现，它有两个参数，

第一个参数为插接在主控制板上的输出端口的序号或引脚号；

第二个参数的范围在 0~180 之间，该参数对标准舵机而言，对应的是标准舵机的转动角度为 0~180 度，标准舵机的默认角度（复位角度）为 90 度；

在图形化编程界面中编写以下程序并烧录。



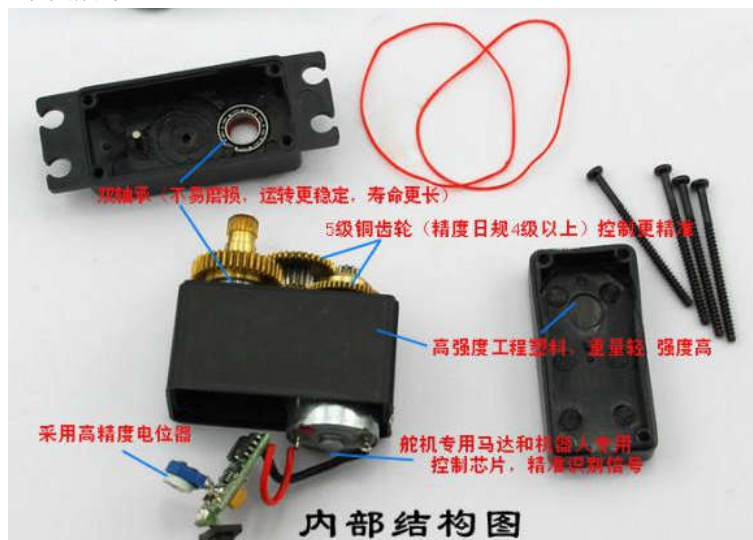
上图程序将实现这样一个功能：接在 D4 端口的标准伺服电机摆动到 120° 的位置，保持 1000ms，再摆动到 60° 的位置，保持 1000ms，循环。

①可以发现，标准伺服电机将以中轴为参照，左右各摆动 30 度左右。也就是说，角度参数 90 对应于伺服电机的中轴。

②加入更多语句，并调整延时参数，将执行效果改变为“初始位置在 90 度，摆动到 30 度，再摆动回到 90 度，再摆动到 150 度，再摆动回到 90 度”的循环。

舵机的原理

舵机的结构组成如下图所示：



舵机的结构组成

舵机的三根线分别为 GND（黑色）、VCC（供电，红色）、信号线（白色）。

标准舵机是比较严格的伺服电机，它具有反馈功能，但是这种小型伺服一般不安装反馈数据线。

由于标准舵机具有准确的舵位控制能力，因此在关节型机器人的设计中应用非常普遍，我们常说的“自由度”概念，就与它相关，一般有多少个伺服电机，就有多少个自由度。



舵机在关节机器人设计中的应用

第四课 控制双轮万向小车

三种基本程序结构

几乎所有编程语言，包括 C 语言，都会有三种基本的程序结构：顺序结构、选择结构和循环结构。

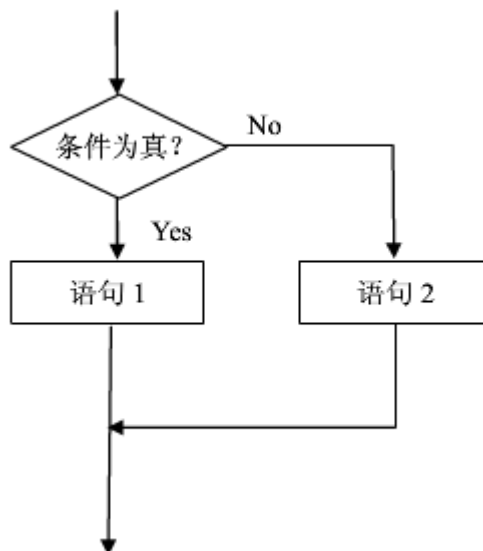
顺序结构

顺序结构是最简单最基本的程序控制结构，程序中的语句将自顶向下逐条执行，即按语句的排列顺序从第一条顺序执行到最后一条，如果要执行第 n 条语句，必须首先执行第 n-1 条语句，且每个语句只执行一边，无需专门的控制语句实现。顺序结构的流程图如下：



条件判断 (if...else)

条件判断就是 if 或者 if……else 语句，多用于传感器触发条件比较简单，触发效率不需要太高的地方；或者判断串口数据的地方。其流程图如下：

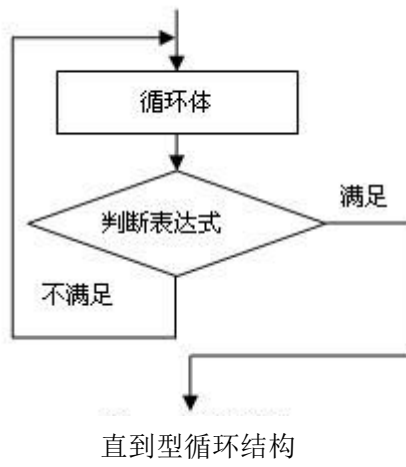
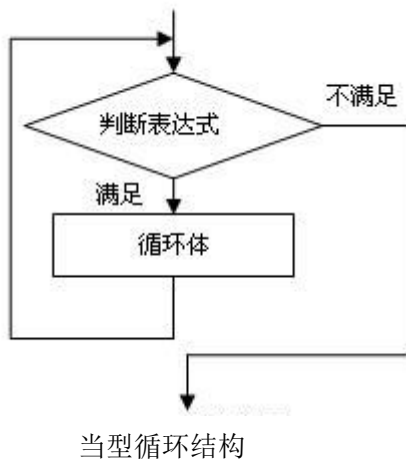


循环结构

循环用于依次执行。比如可以用于依次读取端口状态等。

在处理实际问题时，往往需要重复执行相同的操作，即对一段程序进行循环执行，这种结构即称为循环结构，被重复执行的语句序列称为循环体。C 语言中提供了所有程序设计语言都具有的 WHILE 条件循环和 For 步长循环等训话内结构。该结构中需要根据循环条件是否成立，来决定某些语句是否重复执行。循环结

构有两种：一种是当型循环，即当条件为真时，反复执行循环体，条件为假时，结束循环，当型循环结构流程图如下图所示。另一种是直到型循环，先执行循环体，再判断条件，条件为真时继续循环，直到条件为假时结束循环，直到型循环结构流程图如下：



实验十 双轮万向小车运动

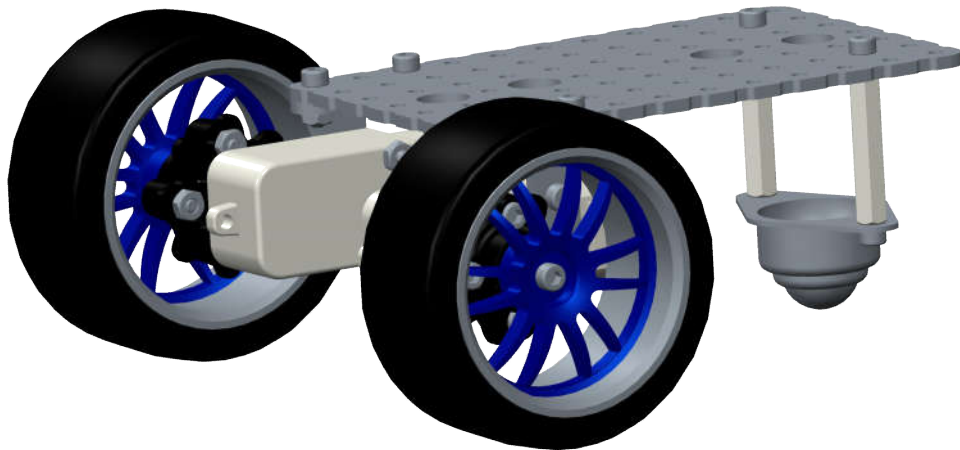
实验目的：1. 具备多个电机的机构的调试；2. 差速双轮底盘的差速运动控制；3. 学习子程序的编写；

实验性质：验证型实验，设计型实验

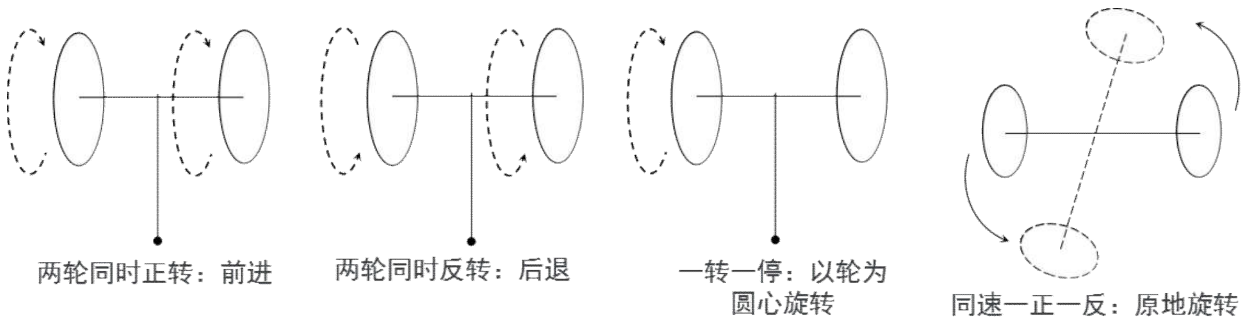
实验课时：1 课时

实验内容：

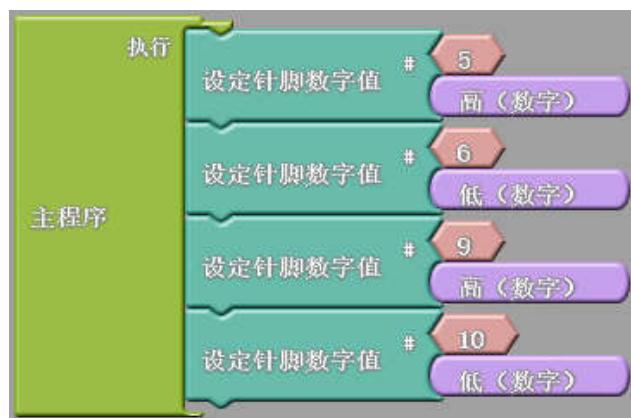
1. 编写程序实现双轮底盘前进、后退、转向、原地旋转 4 个动作：



2. 四种运动形式的图示：



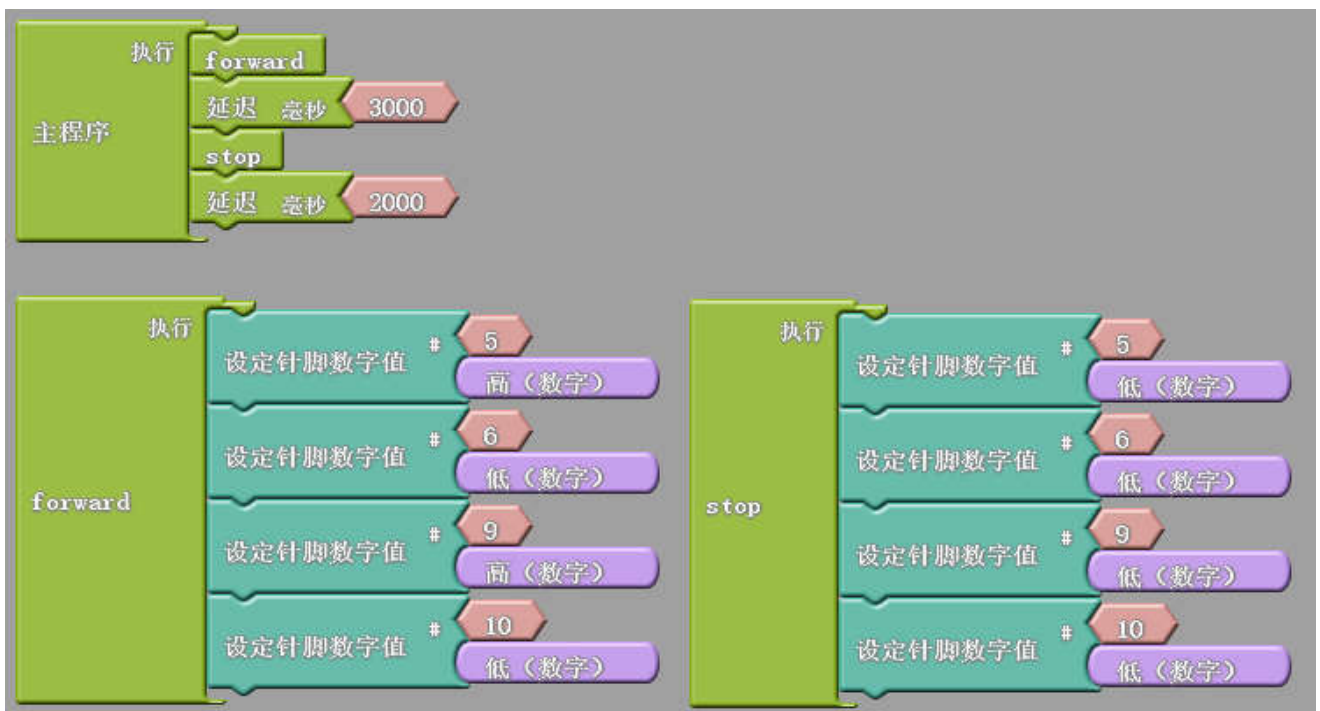
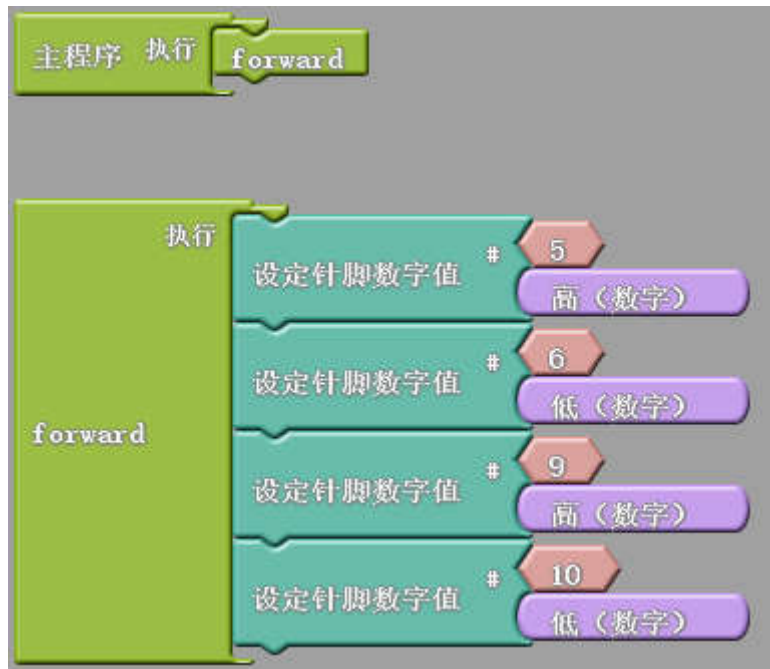
3. 示例程序：



4. 该底盘的运动不仅仅和引脚的电平高低组合有关系，还和电机插针的正反有关系。当发现轮的行进方向与预想不同时可以“保持电路不变而调整程序”，也可以“保持程序不变而调整电路”。

子程序模块:

如下图的两个程序，主程序里调用子程序，子程序的具体内容在模块中编写，使主程序简洁明了。



请大家编程并调试，实现前面说的四个功能，用子程序的方式编写。

实验十一 开关启动机器人

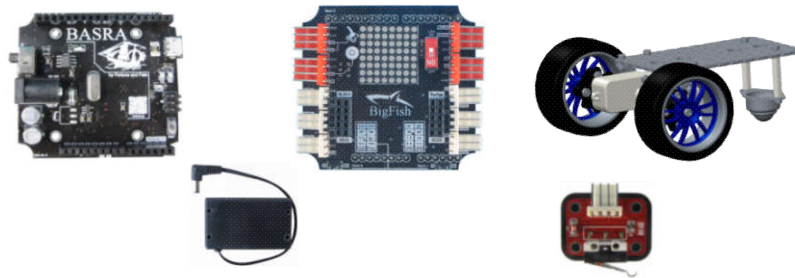
实验目的：1. 使用简单传感器的数字量（即开关量，0 或 1）检测功能；2. 掌握触碰传感器的特性和使用方法；3. 通过搭建检测电路，掌握“探索者”基本检测电路的连接方法；4. 通过编写触碰开关程序，掌握图形化编程，C 语言编程；5. 学会使用“如果/否则”逻辑；

实验性质：验证型实验

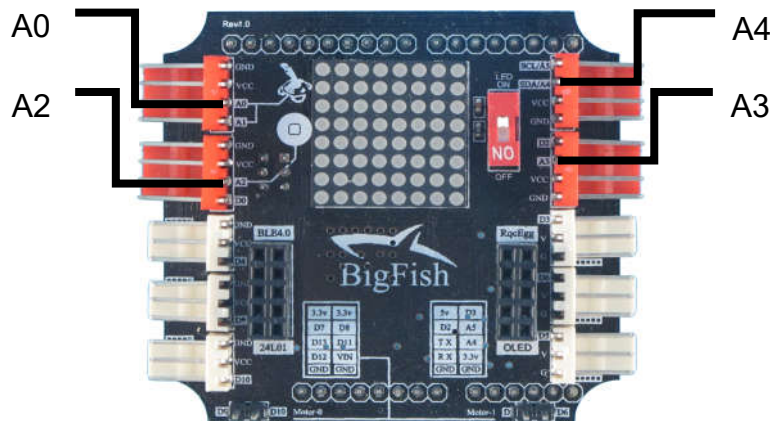
实验课时：1 课时

实验内容：

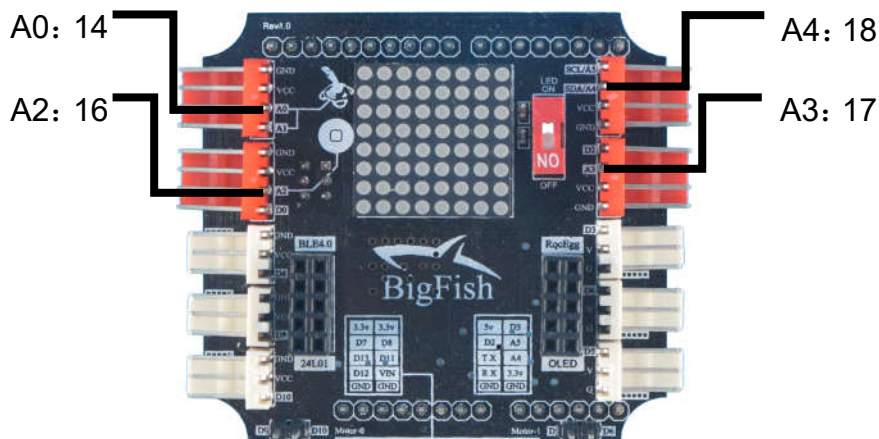
1. 将主控板、电池、触碰传感器固定在小车上，将扩展板堆叠在主控板上。编写程序：如果传感器触发，小车前进 3 秒，否则小车不动。





C 语言编程时需要用到端口号，传感器对应的端口号要看传感器端口 VCC 针脚旁边的编号，即：A0，A2，A3，A4。

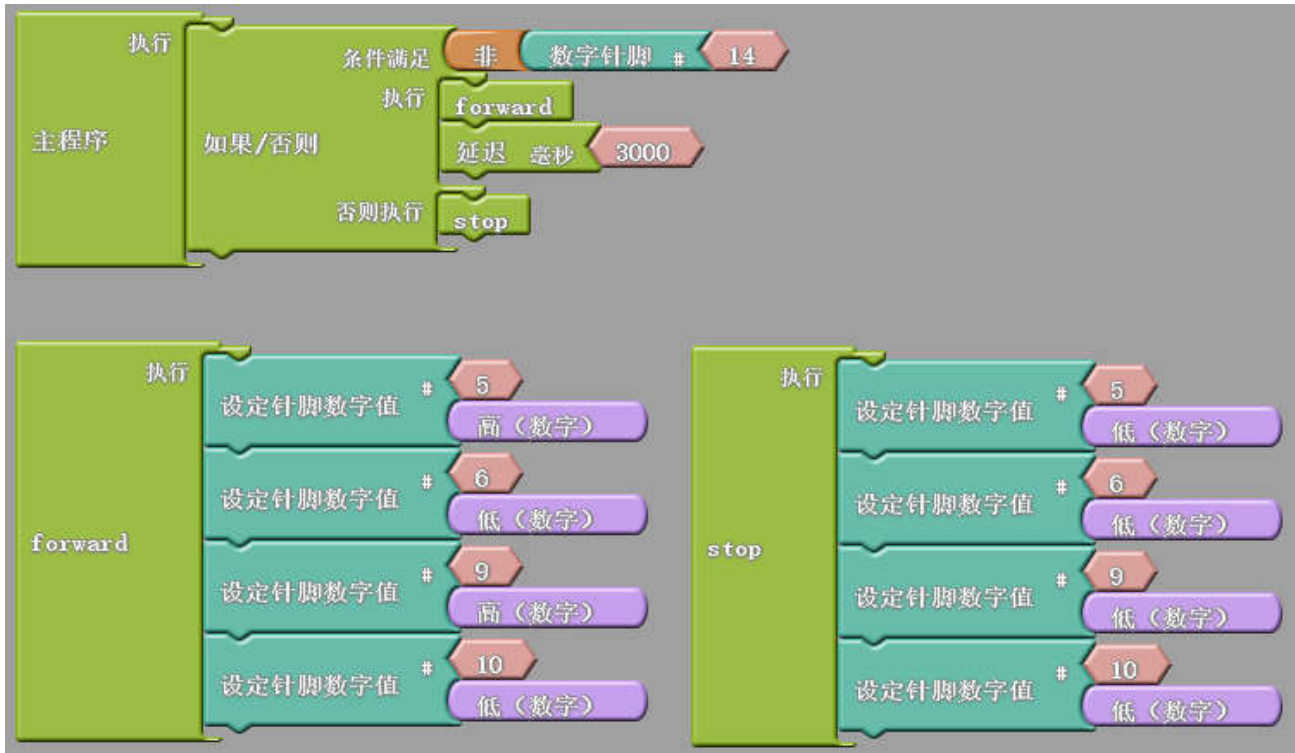


图形化编程时也需要用到端口号，图形化程序对端口号认定的方法比较特殊，对于 Basra，A 后面的数字加上 14 即可，如 A3 端口号为 3+14=17。

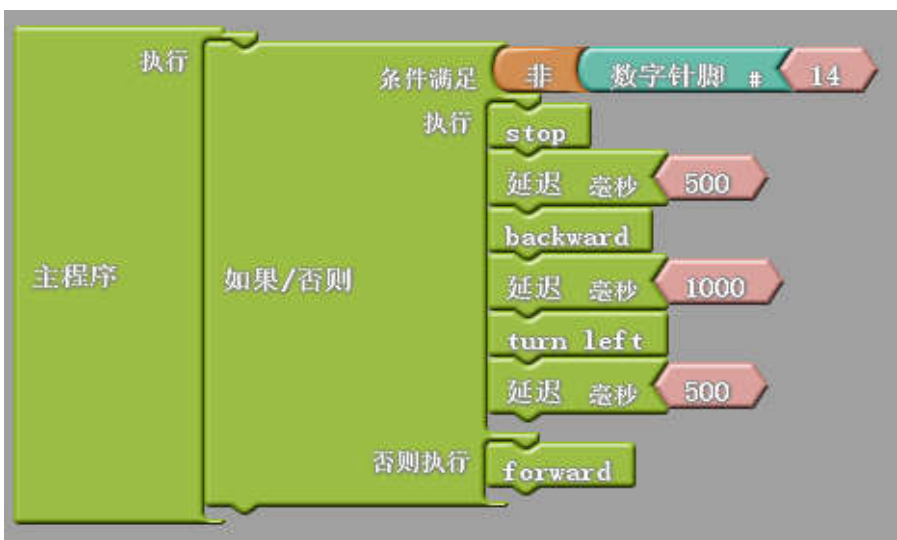


2. 在图形化编程界面中参考以下程序并编写烧录，自己完成调试（传感器的触发条件是“低电平”，而

语句  意思是“14号数字引脚获得高电平”，因此要在前面加一个逻辑运算符 )：



3. 拓展：1. 改用近红外传感器实现该功能；2. 使用近红外传感器实现避障功能，即“如果传感器触发，小车停止、后退、转向；否则，小车前进”，请自己编写子程序并完成调试。



第五课 传感器的高级用法

实验十二 数字量传感器测值

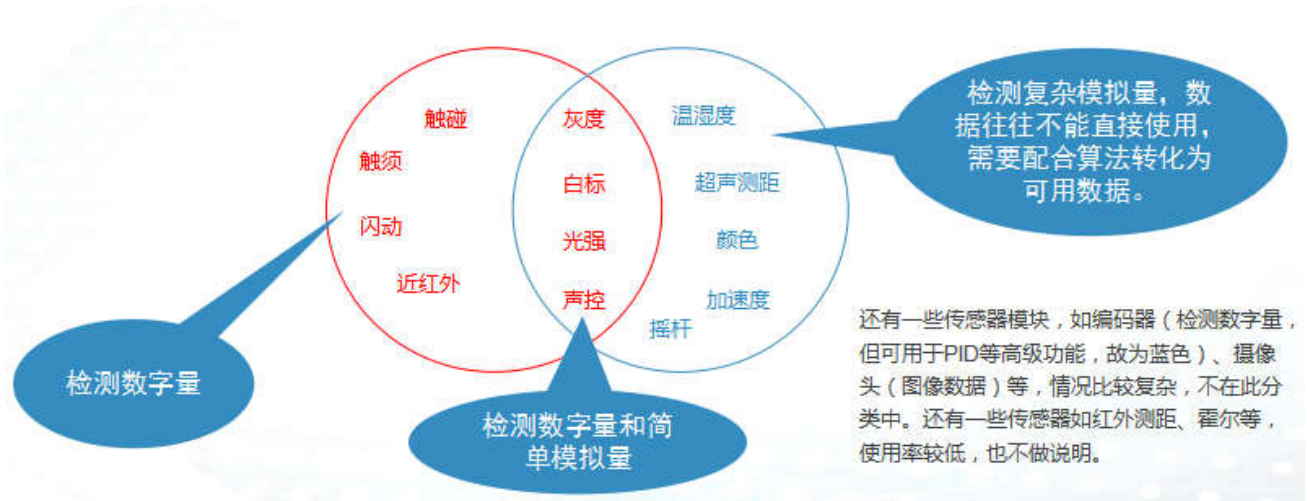
实验目的：1. 了解数字量传感器和模拟量传感器的概念和区别；2. 学会利用串口监视器监测数字量传感器采集到的数值；3. 熟悉 serial 系列函数。

实验性质：验证型实验

实验课时：2 课时

实验步骤：

传感器可以大体上分为数字量传感器和模拟量传感器。我们之前在设置引脚值得时候接触过类似的概念，比如“设置引脚数字值”和“设置引脚模拟值”。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。

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()
{
```

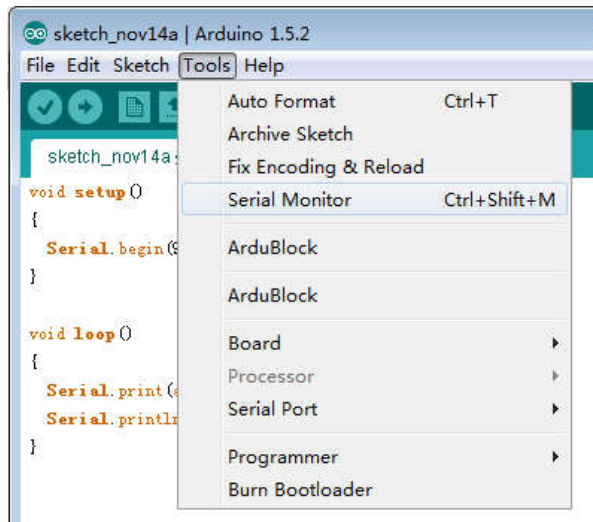
```
Serial.print(! ( digitalRead(14) ));  
Serial.println();  
}
```

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。

3. 搭建电路，分别监测触碰、触须、近红外、光强、黑标、声控这 6 种传感器。

串口监视器其实有很多，大家可以在网上搜索、下载各种版本的串口监视器，以开阔思路。

实验十三 模拟量传感器测值

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

实验步骤:

实验 5 中我们讲到, 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);
}

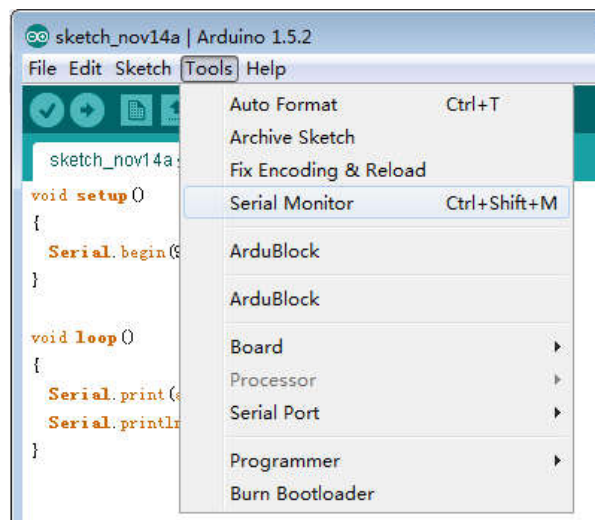
void loop()
{
  Serial.print(analogRead(14));
  Serial.println();
}
```

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。



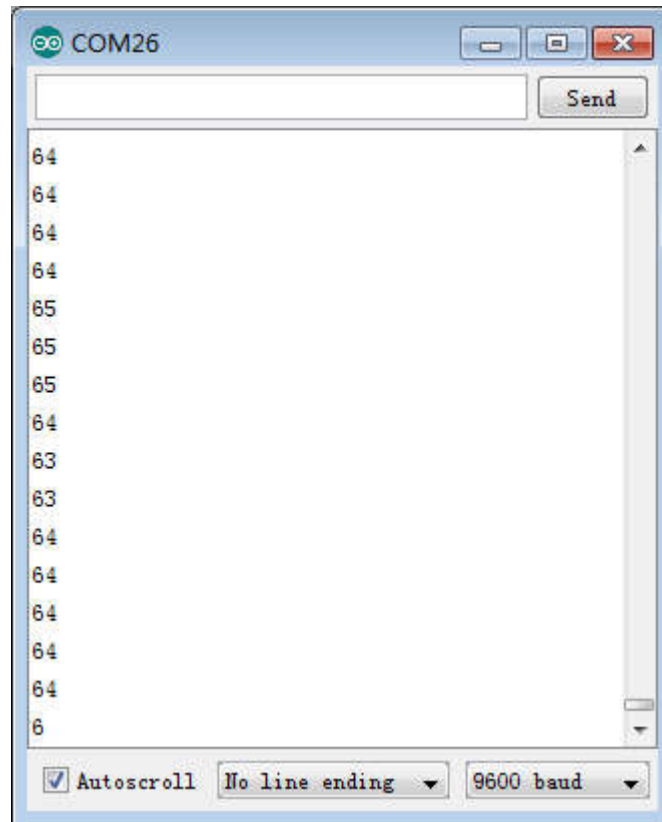
在 C 语言界面的 Tools 菜单里面, 也可以找到 serial monitor 选项



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。

3. 搭建电路，监测光强、黑标、声控这 3 种传感器。上个实验我们已经监测过这两个传感器了，为什么现在又用到了呢？因为这三种传感器同时具备数字量和模拟量的功能。你可以使用手电筒、遮挡来测试光强；用不同深浅的色彩来测试黑标（此时黑标就有灰度的作用了）；自己发声来测试声控。

4. 那么其他的传感器呢？比如超声波、温湿度等等。这些传感器功能比较复杂，不能直接监测，后面我们会陆续讲到它们的监测方法。

实验十四 超声波传感器测距算法

实验目的：1. 练习用串口监视器监测超声波传感器；2. 理解超声测距传感器的；3. 理解超声测距算法的数学意义。

实验性质：验证型实验

实验课时：2 课时

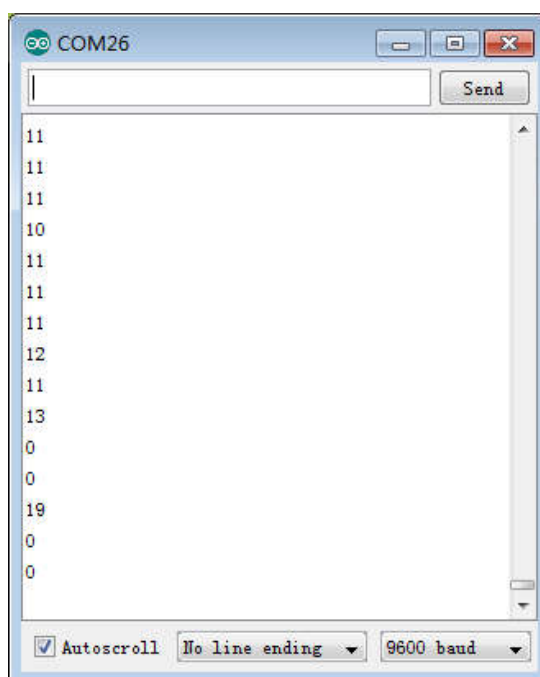
实验步骤：

1. 串口监视器测值

超声波有两个数据针脚，我们可以用到超声波专用的接口图块来完成监测



监测到数据如下：



如果我们拿一个尺子并在超声波传感器前面移动障碍物，我们会发现这个监测到的数据确实是比较准确的距离值，单位是 cm。

超声测距传感器是有图块的，而像温湿度、颜色识别、红外编码器传感器没有专用的图块，直接读的话，读出来的数据又看不懂，这时候我们就需要研究一下这个传感器了。

2. 测距算法

超声波的例程中，查看生成的 C 语言，我们会发现源代码比较复杂：

```
int ardublockUltrasonicSensorCodeAutoGeneratedReturnCM(int trigPin, int echoPin)
{
    long duration;
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    digitalWrite(trigPin, LOW);
```

```

    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(20);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    duration = duration / 59;
    if ((duration < 2) || (duration > 300)) return false;
    return duration;
}

void setup()
{
    Serial.begin(9600);
    digitalWrite( 15 , LOW );
}

void loop()
{
    Serial.print( ardublockUltrasonicSensorCodeAutoGeneratedReturnCM( 15 , 14 ));
    Serial.println();
}

```

以上这段 C 代码是自动生成的（其实是 Ardublock 的开发者写的）。

如果我们打开器材配套的资料，在电子模块资料，超声测距传感器的资料文件中可以发现，“探索者”的工程师也写了一个例程：

```

#define ECHOPIN 14
#define TRIGPIN 15
void setup()
{
    Serial.begin(9600);
    pinMode(ECHOPIN, INPUT);
    pinMode(TRIGPIN, OUTPUT);
}

void loop()
{
    digitalWrite(TRIGPIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIGPIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGPIN, LOW);
    float distance = pulseIn(ECHOPIN, HIGH);
    distance= distance/58;
    Serial.println(distance);
    delay(500);
}

```

这两段代码功能是一样的，大家要对比阅读。它们的思路是一样的，在数据处理上细节有些小差异，

比如在计算距离数值时，一个是拿 59 做除数，一个是拿 58 做除数。这两段代码无所谓哪个更权威，代码都是普通人写的，所以用户不要迷信“例程”，要根据自己的需要，对代码进行选择使用或者改写。

里面有一个陌生的函数 pulseIn()，我们可以上网查询这个函数的功能，由于 Arduino 是一个国际通用的开源体系，所以用户量非常庞大，互联网上有海量的共享资源和交流人群，所以一定能搜索到。

经过查询我们可以知道 pulseIn() 的功能是获取两个信号的时间差，即发出超声波到收到反射回来的超声波的时间差，单位是微秒。有了这个时间差，那么中学物理知识就派上用场了：

$$\text{距离 } D = \text{声速 } v \times \text{往返时间差 } t / 2$$

根据中学物理知识，我们知道声波在空气中的传播速度是 340m/s，但是这个数据是有前提的：1 个标准大气压和 15℃ 的条件下。经再次搜索可知，超声波在 1 个标准大气压和 20℃ 的条件下速度为 344m/s。如果改成 cm/μs 为单位，那么

$$D = 344 \times 10^2 \times 10^{-6} \times t / 2 = 0.0172t \approx t / 58$$

所以超声波测距算法约定俗成的写作：

```
distance = pulseIn(ECHOPIN, HIGH);
distance = distance / 58;
```

3. 知道了这些原理，我们就可以写出超声波测距功能程序。假设我们选择一个直流电机，设定这样一个功能：当目标离它 10cm 以下时，不动，超过 10cm 时，开始转动，那么程序可以写作：



看懂了这个程序后，请大家改写这个程序，用 023 号机构或其他机构实现超声避障或跟随功能。

4. 这样，我们就从原理到实践，完全掌握了一个陌生的，比较复杂的传感器。

从上面的案例中，我们大致可以总结出一套通用的学习传感器的方法：

(1) 当我们遇到一个新的传感器时，首先阅读它的例程和资料，从而知道它的工作原理、信号输出引脚、信号类型、涉及到的语句或函数，有不清楚的就去查询；

(2) 进而通过串口监视器监测它的工作状态，验证它的工作方式和数据生成情况等。

(3) 根据它的工作方式，将它安排在机身的合适位置，根据它的数据生成情况，编写合适的程序语句。

阅读例程和学会看电路（能看引脚功能就行）是非常重要的。能够熟练地查询相关资料也是很重要的。

现在是信息爆炸的时代，这种开源硬件、常用电子模块和编程的资料、教程、案例等都非常容易查到。用户要尽量自己查询，自己摸索，实在解决不了可以找技术人员询问**参考资料**的地址，切莫做伸手党——直接要答案。

“探索者”器材里涉及到的所有较复杂的传感器，如超声测距、温湿度、红外编码器、都给出了串口监视例程。请大家学习的时候自己查询使用。

实验十五 超声波避障功能方案

实验目的：1. 了解一种避障功能的实现方式；2. 了解超声测距传感器的工作特性及安装注意事项；3. 练习C语言编程；4. 学会使用超声波距离测量函数。

实验性质：设计型实验

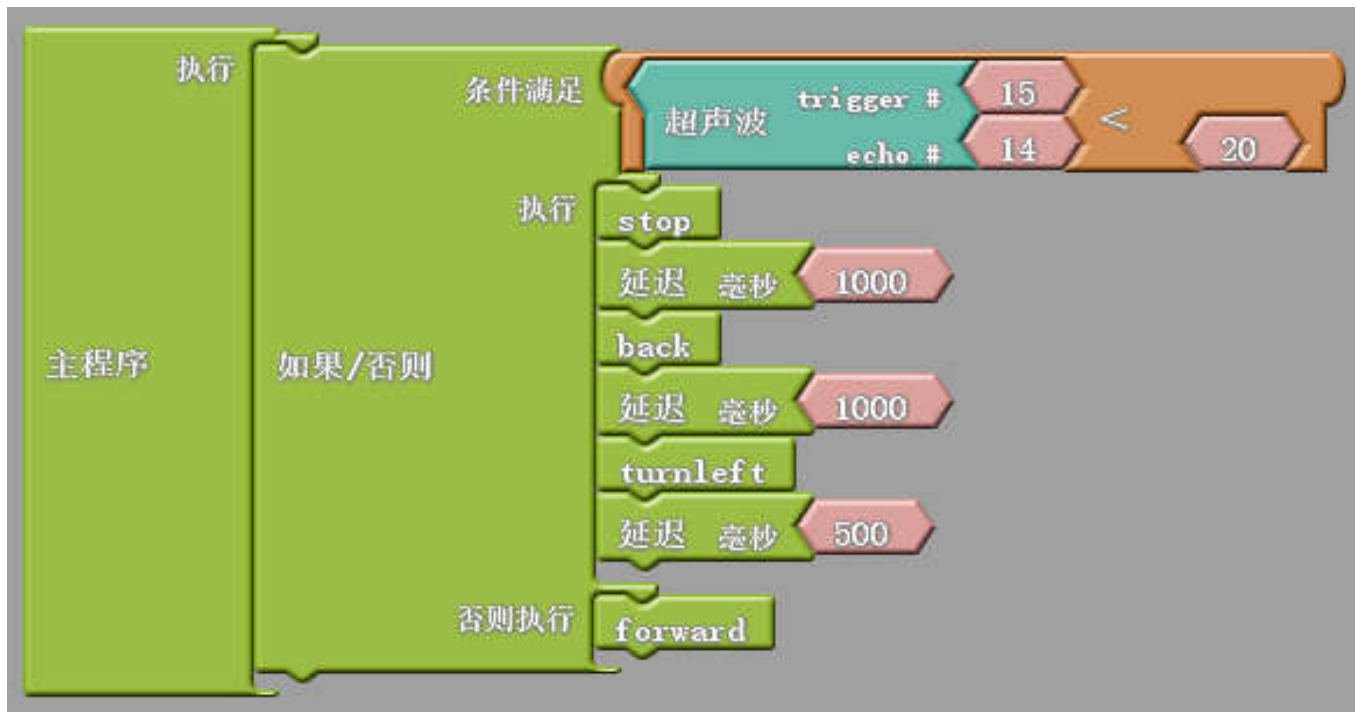
实验课时：2 课时

参考资料：请参考光盘中的“电子元件资料”中的“HC-SR04 超声波测距模块”资料；Bigfish 连接常规传感器.ppt；实验 4；

主要器材：超声测距传感器×1

实验步骤：

1. 将 1 个超声测距传感器安装在合适的位置，连好电路；
2. 阅读、编写并烧录下面的程序（编号：exp10_1），它将实现这样一个功能：小车持续前进，距离障碍 10cm 即后退并转向另一个方向行驶，以躲避障碍。



1. 请自行调试机器人，根据需要调整代码或者机构，使其较好地完成任务；
2. 改写该程序，使其表现出不同的避障效果。
3. 换另外一个具备相似运动能力的机器人，给它也增加这个功能。

第六课 舵机和履带的使用

实验十六 简易排爆车

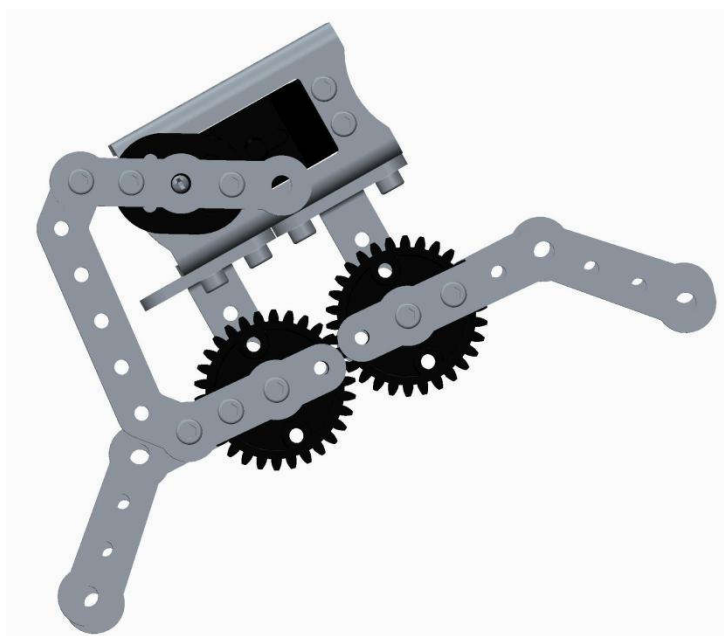
实验目的：1.熟悉探索者零件使用；2.熟悉舵机控制；3.了解排爆机器人；

实验性质：验证型实验，设计型实验

实验课时：2 课时

实验内容：

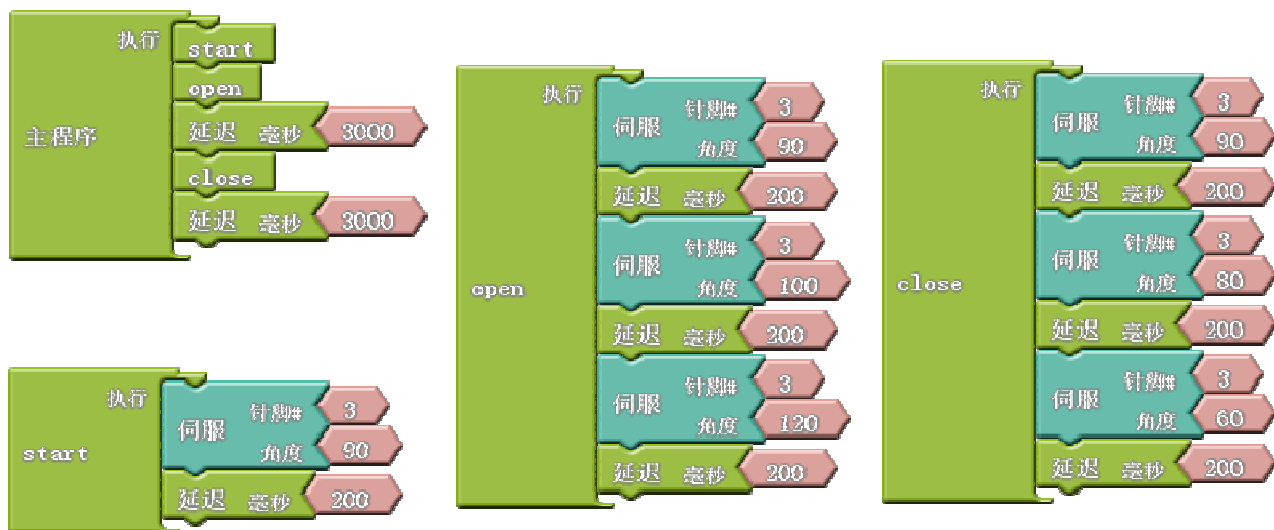
1.参考 stp-class3 文件夹里的 3D 文件组装一个机械爪模块，将机械爪的舵机连在 bigfish 上，参考实验八编程实现开合动作：



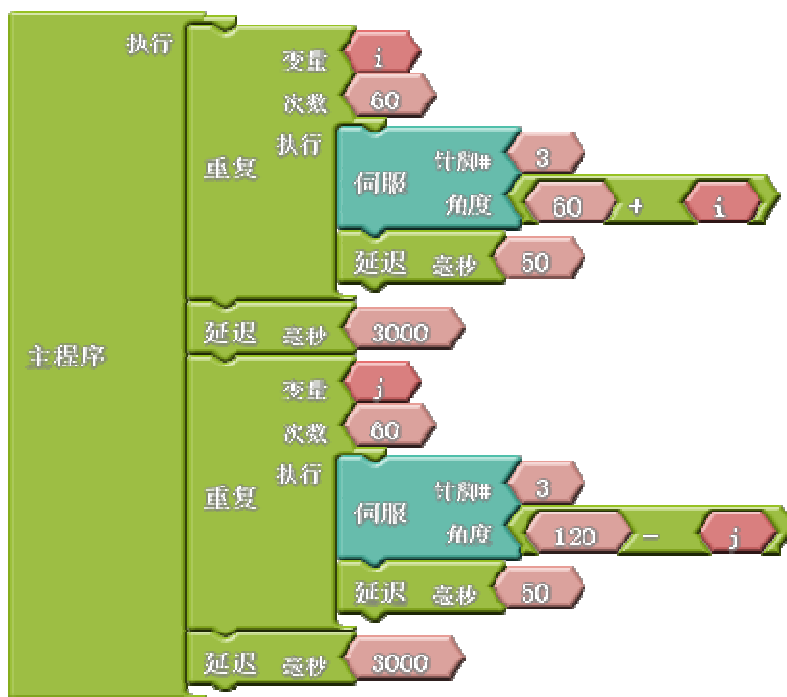
2.参考例程：



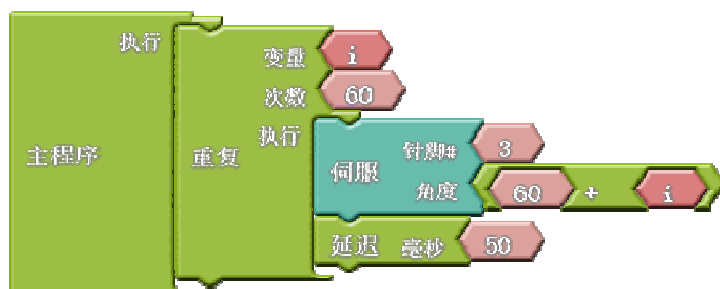
3.上面那段例程的运行效果比较生硬，可以通过增加动作停留点的方法将动作变得柔和。比如在 90、120、60 之间再加入 100、80 等；甚至每隔 10 度增加一个动作点。



4.这样的程序非常笨拙，而且容易出错。还可以用“重复”语句来进行优化。



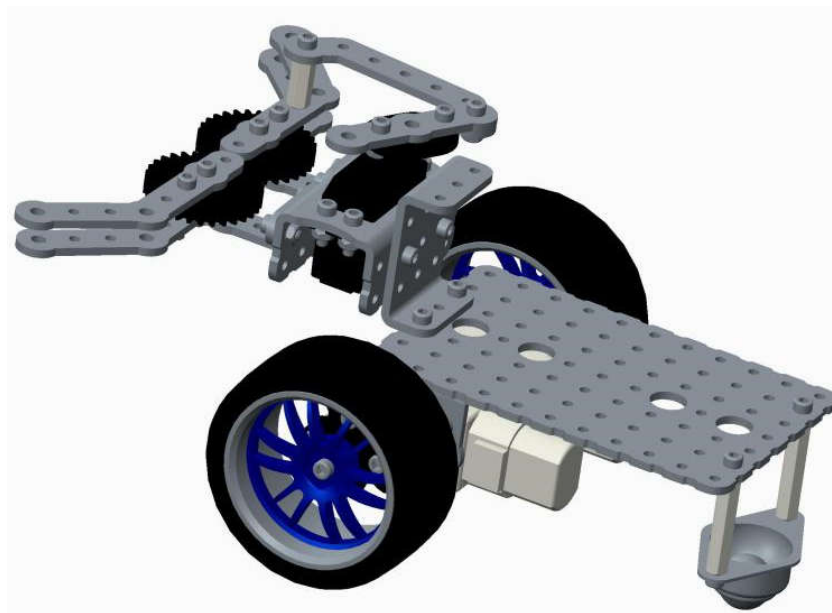
“重复”语句的后台是 for 循环，如下图的简单 for 循环，它设置了一个变量 i,让舵机的角度值与 i 一致，从而让舵机从 60° 起，角度参数每 50 毫秒加 1，直到 120°。



```
for (i= 1; i<=60; i++)  
{  
servo_pin_3.write(60 + i);  
delay( 50 );  
}
```

由于自动生成的 C 语言例程变量名比较长，为了方便阅读，上文的 C 语言例程做了简化。由于图形化默认的 i 初始值为 1，因此我们需要安排一个 (60+i) 的操作。如果用 C 语言写，直接写成 i=60;i<=120;即可。i++ 的意思是 i 的值每循环 1 次+1。

5.将机械爪安装于超声波小车上，编写程序，实现“小车向前走到某个特定位置时，停止运动，并抓取某个物体”的功能。你可以用手充当超声波传感器的标志物，让小车能够识别与手的距离，并让机械爪做出抓取动作即可：

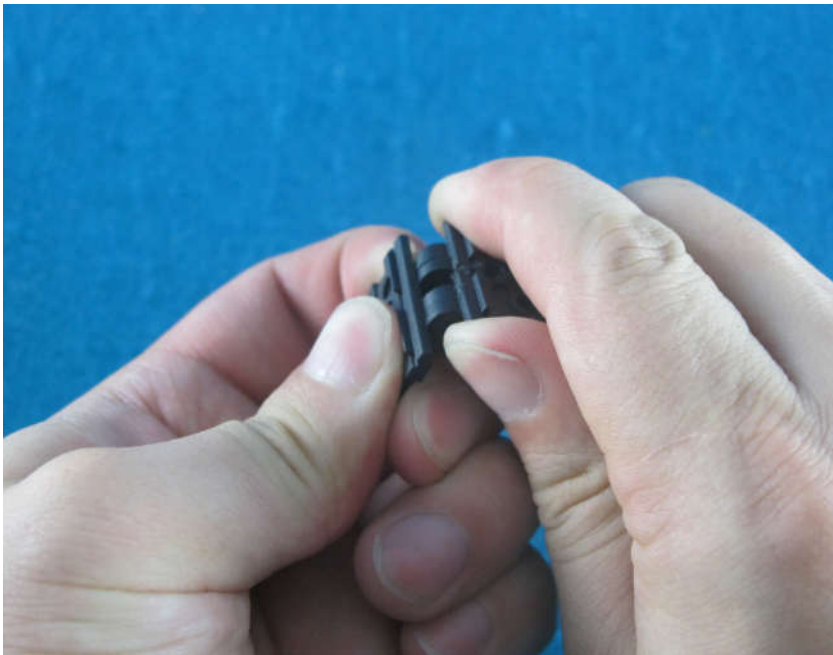


履带的组装与拆卸



组装技巧：

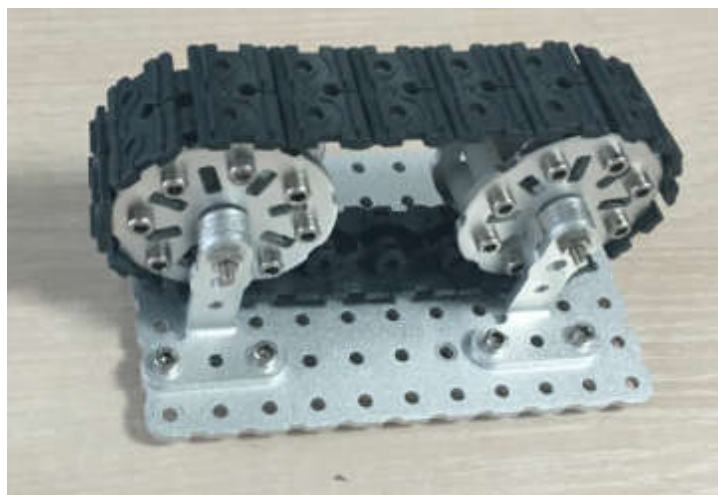
组装时，两个履带片成锐角时，两个履带片的大口和小口最容易匹配，便于安装。



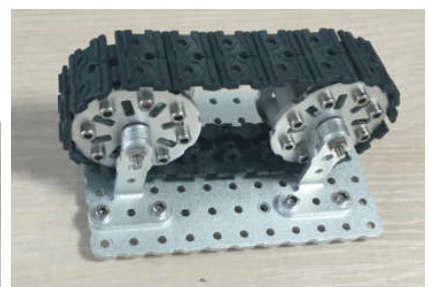
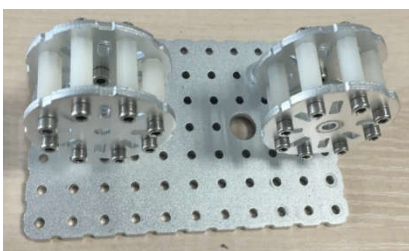
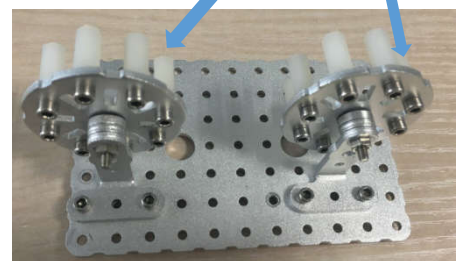
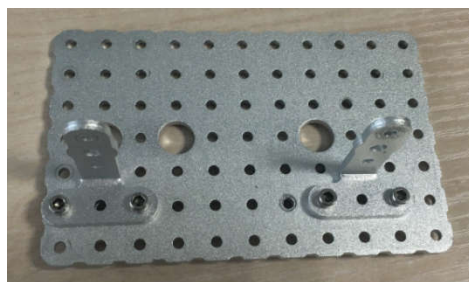
拆卸技巧:

履带片拆卸时, 同样两个履带片成锐角时方便拆卸。

参照下面图片组装一个传送带:



注意加轴套



提示

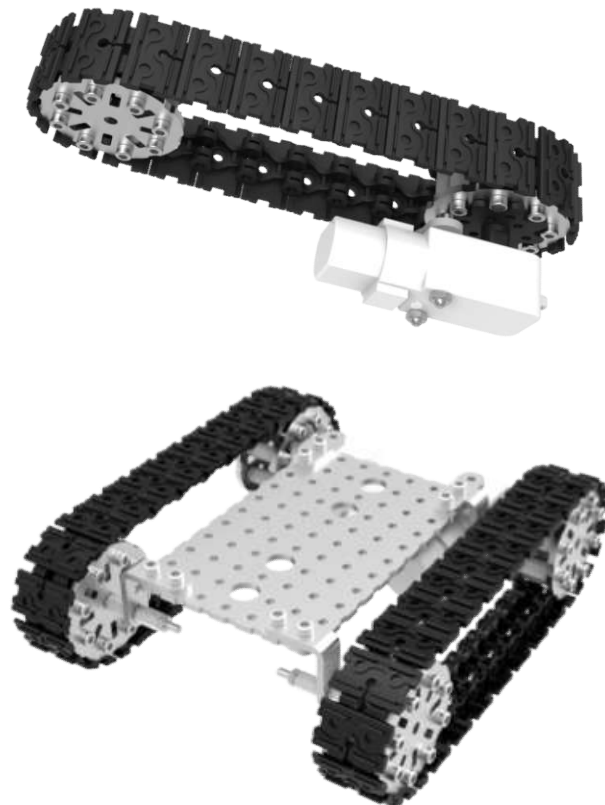


螺丝一穿到底：强度高，费力



短螺丝两头对锁：强度低，方便

尝试组装一个履带底盘



第七课 机器人避障

实验十七 避障实验

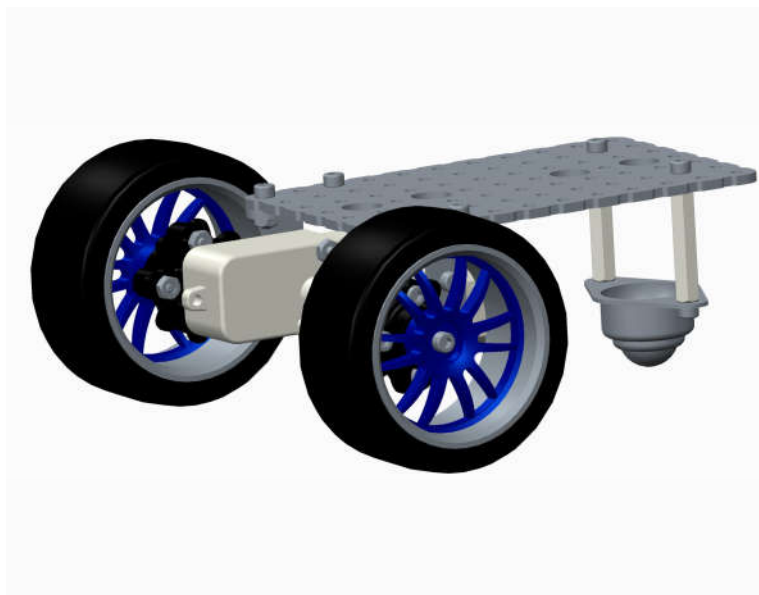
实验目的： 1. 了解几种避障功能的实现方式； 2. 掌握传感器的工作特性；

实验性质： 设计型实验

实验课时： 4 课时

实验步骤：

1. 在 023 号机构上的适当位置安装传感器，连好电路；可以选择触碰、触须、近红外、超声测距等。
2. 要求至少使用 2 个传感器。



可以参考实验十一“拓展部分”的代码。

第八课 机器人简单循迹

实验十八 小车循迹实验

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。

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：

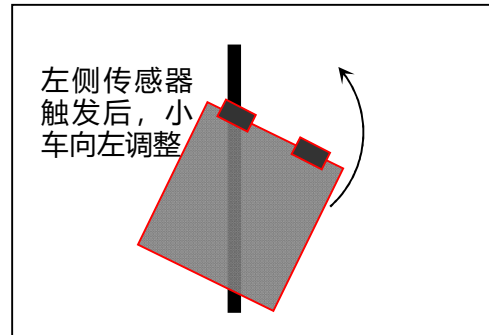
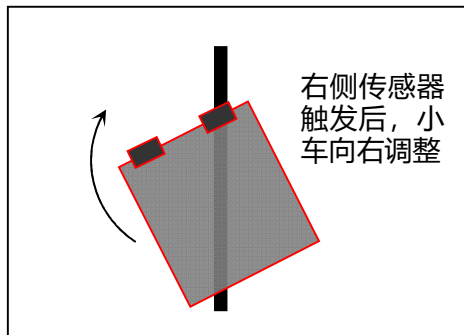
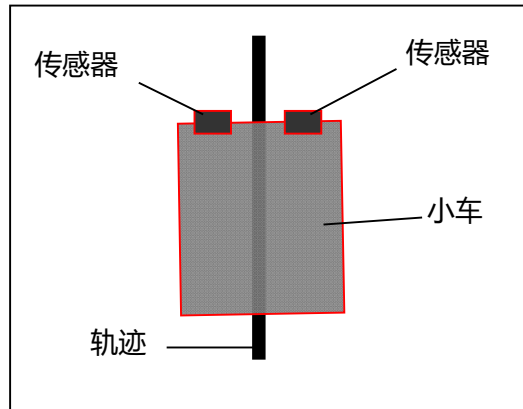
将两个灰度传感器安装在 023 号机构的底部前端，传感器距离车轮越远效果越好，具体位置请自己尝试，连好电路；

1. 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，就向右行驶来纠正；同理，如果右侧传感器检测到轨迹，就向左行驶来纠正。这样就保证轨迹始终在两个传感器之间。如下图所示：



2. 阅读、编写并烧录下面的程序，它将实现这样一个功能：把小车放置在黑线上，小车能够自动沿着黑线行驶。



沿黑线行驶程序

生成 C 语言代码为：

```

void setup()
{
  pinMode( 18, INPUT);
  pinMode( 14, INPUT);
  pinMode( 5 , OUTPUT);
  pinMode( 9 , OUTPUT);
}

void loop()
{
  if (!( digitalRead(14) ))
  {
    digitalWrite( 5 , HIGH );
  }
  if (!( digitalRead(18) ))
  {
    digitalWrite( 9 , HIGH );
  }
}

```

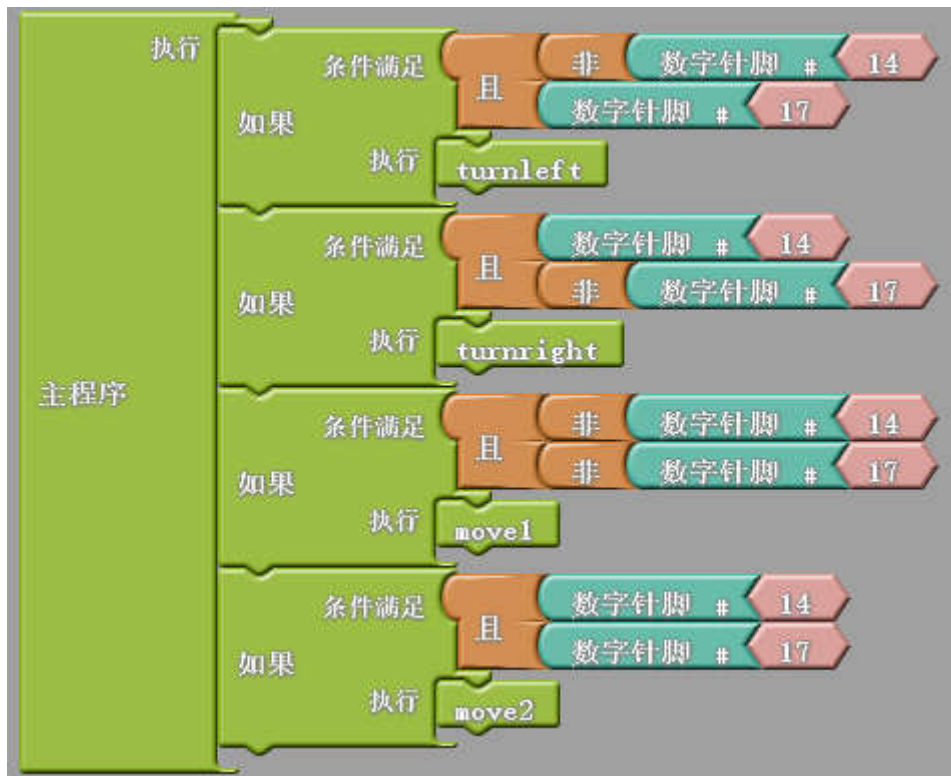
3. 在白色场地上用黑色绝缘胶带铺设一条轨迹，先尝试直线，熟练后可尝试弧线、圆、矩形等；

4. 上文示例程序能够循迹，但是大家应该会发现它的运行并不稳定，经常出现识别不到黑线的情况。我们可以改写程序，使机器人实现更好的循迹效果。

当使用两个传感器的时候，传感器的触发总共有 4 种状况：

- 1、左侧未触发，右侧未触发
- 2、左侧触发、右侧未触发
- 3、左侧未触发、右侧触发
- 4、左侧右侧同时触发

由于上文示例程序仅仅涵盖了 2、3 两种状况，一旦出现 1、4 状况，机器人就会“无所适用”，就会发生动作混乱，因此，我们应该把四种情况全部列出，为机器人指定相应的动作策略。



至于 move1 和 move2 是什么动作，有大家自行决定。

5. 重要提示：

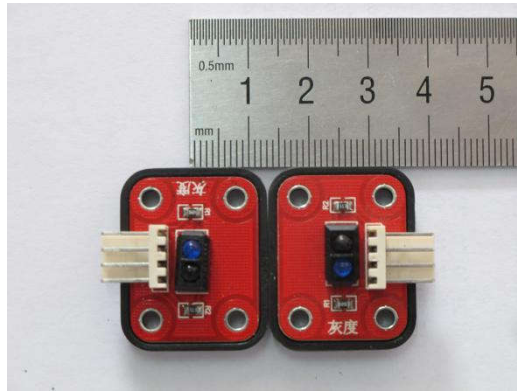
小车结构、场地、电机速度等因素，对循迹效果影响非常明显。以下要点请大家注意。

- (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) 黑标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。

第九课 机器人编程框架

有限状态机算法解析：

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。

(1) 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);
}
```

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，那么这个传感器有两个状态。我们可以得到一个状态表格：

状态序号	传感器 1
1	1
2	0

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（实验 18 的做法）。

状态序号	传感器 1	传感器 2	伪码
------	-------	-------	----

1	1	1	if { Sensor (1,1); Sensor (2,1); }
2	1	0	if { Sensor (1,1); Sensor (2,0); }
3	0	1	if { Sensor (1,0); Sensor (2,1); }
4	0	0	else

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？

(2) 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。

(3) 算法

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： $1 \ll n$ ，意思是 1 向左移动 n 位，空出来的数位用 0 填补。

如： $1 \ll 1$ ，结果就是 10； $1 \ll 2$ ，结果就是 100； $101 \ll 1$ ，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：

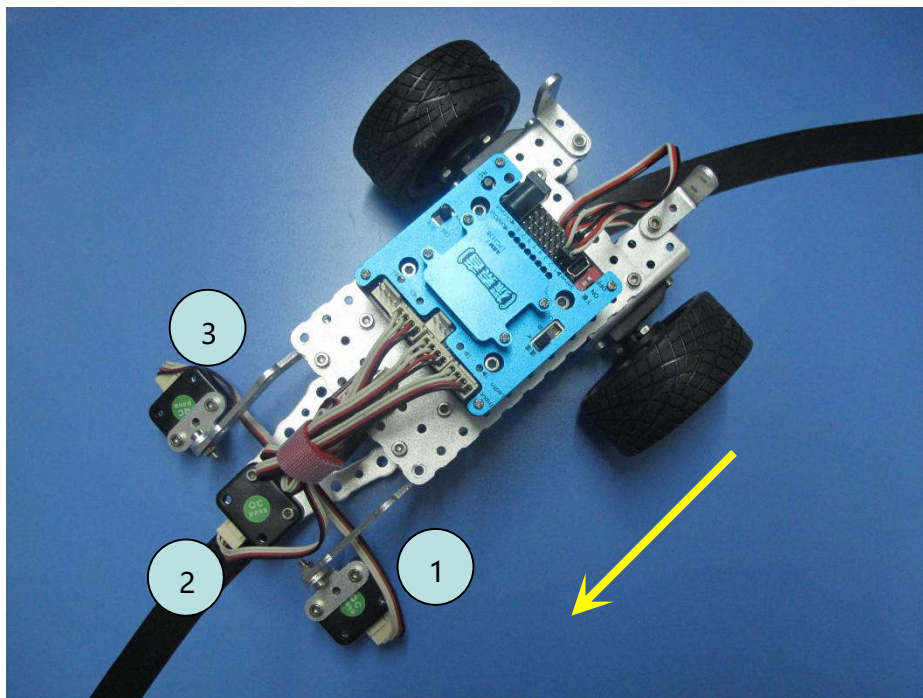
```
s=0;
for(i=0;i<2;i++) //因为此例中有 2 个传感器，i 取 2
{
s=s|(Servo(i+1, 触发判断)<<i); //获得传感器值，移位，或运算
}
```

于是 switch 语句可以写为：

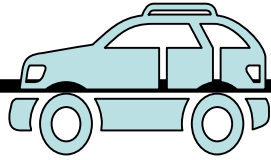
```
switch(s)
{
case 0x00 : {动作 0;}break; //序号也可以写作 16 进制数值
case 0x01 : {动作 1;}break;
case 0x02 : {动作 2;}break;
case 0x03 : {动作 3;}break;
default;;break;
}
```

(4) 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（结合场地分析）	动作策略（可自定义）
0	0	0	0	都没触发，可能是跑偏了	停
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。

实验十九 控制双轮循迹小车

实验目的

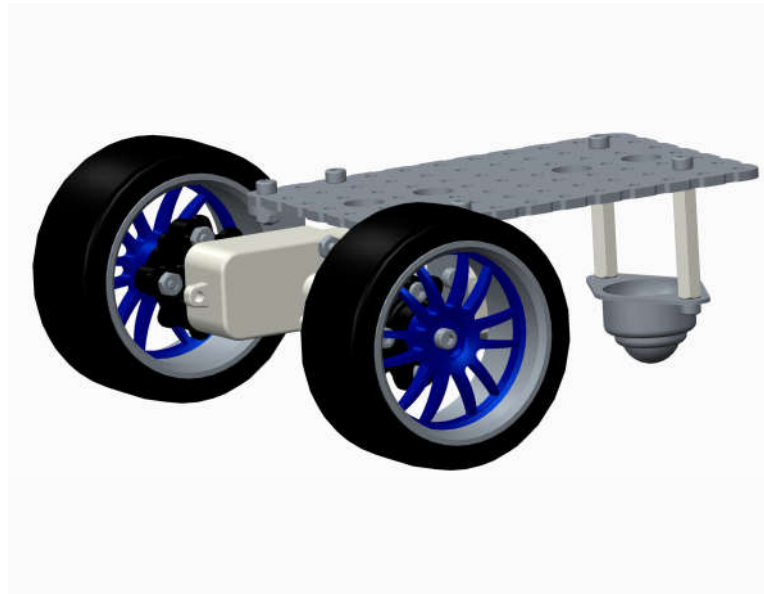
掌握循迹小车的设计及控制思路，综合使用前面学到的有限状态机算法；体会优化的机构设计对程序执行效果的影响；体会场地环境对程序执行效果的影响。

实验性质：设计型实验

实验学时：2 学时

实验步骤

为小车安装 3 或 4 个传感器，根据前面学过的知识，自己编写有限状态机循迹程序。



让小车沿着矩形场地行走



第十课 机器人识别悬崖

实验二十 悬崖的躲避与巡检

实验目的：了解一种有趣的悬崖巡检功能的实现方式。

实验性质：验证型实验

实验课时：2 课时

主要器材：近红外传感器，灰度传感器

实验步骤：

1. 在双轮万向车的机构上安装一个检测装置，它可以由近红外传感器或者灰度传感器和一系列机械零件组成；安装位置请自己思考尝试。
 2. 近红外传感器可以识别桌面，灰度传感器可以识别“悬崖”。灰度传感器发射红外线，并接收反射回来的红外线，可以检测出反射回来的红外线的强度数据，目标颜色越深则红外线被吸收的越多，因此可以反映目标颜色的灰阶。如果遇上“悬崖”，则也会产生反射回来的红外线减少甚至消失的情况。因此可以用于识别“悬崖”。
- 1、请参考实验十八编写程序，让机器人在桌面上行驶，不会掉下去。
 - 2、请参考实验十八编写程序，让机器人沿着“悬崖”巡逻，既不远离，也不掉下去。



第十一课 蓝牙遥控

实验二十一 蓝牙遥控机器人

实验目的：了解移动设备与机器人蓝牙通信的一种实现方式，熟悉安卓手机与蓝牙串口模块连接；

实验性质：验证型实验、设计型实验

实验课时：2 课时

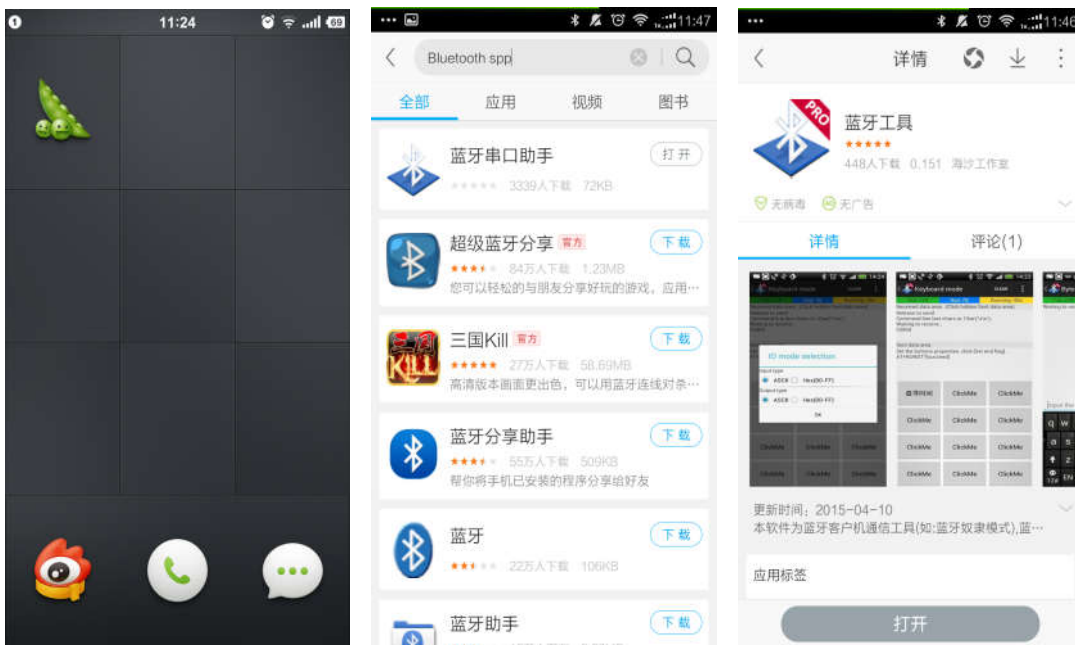
主要器材：蓝牙串口模块×1，自己的安卓手机

实验步骤：

1. 在安卓手机上按如下步骤操作：

(1) 安装蓝牙串口助手 app

将..\“探索者”软件套装\android\蓝牙串口助手.apk 安装到手机里，或者在 google play、豌豆荚中搜索 bluetooth spp 安装，并升级，如下图所示。



安装蓝牙串口助手 app，并升级

(2) 设置蓝牙串口助手

打开蓝牙串口助手 app，按如下步骤进行配置

<p>点击搜索到的 HC-05 设备，初次连接需输入密码 1234 进行配对</p>	<p>配对完成后选择操作模式为“键盘模式”</p>	<p>就能够看到键盘界面</p>

<p>点击菜单键，选择“配置键盘值”</p>	<p>将两个按钮的名字分别设置为“正转”，“反转”；发送的数据分别为 1、2</p>	<p>一会儿就可以用这两个按钮控制机构了</p>

配置蓝牙串口助手 app

2. 将蓝牙模块连接到 Bigfish 扩展板上，并将 Bigfish 插到控制板上。接下来只要我们把串口数据和机器人要做的动作匹配起来，就可以实现用串口控制机器人运动了。用图形化程序也可以实现该功能，阅读、学习并烧录以下程序，实现用安卓手机 APP 通过串口控制 023 号小车运动。下载以下程序时，先不要堆叠蓝牙串口模块，因为会占用串口，造成下载失败。



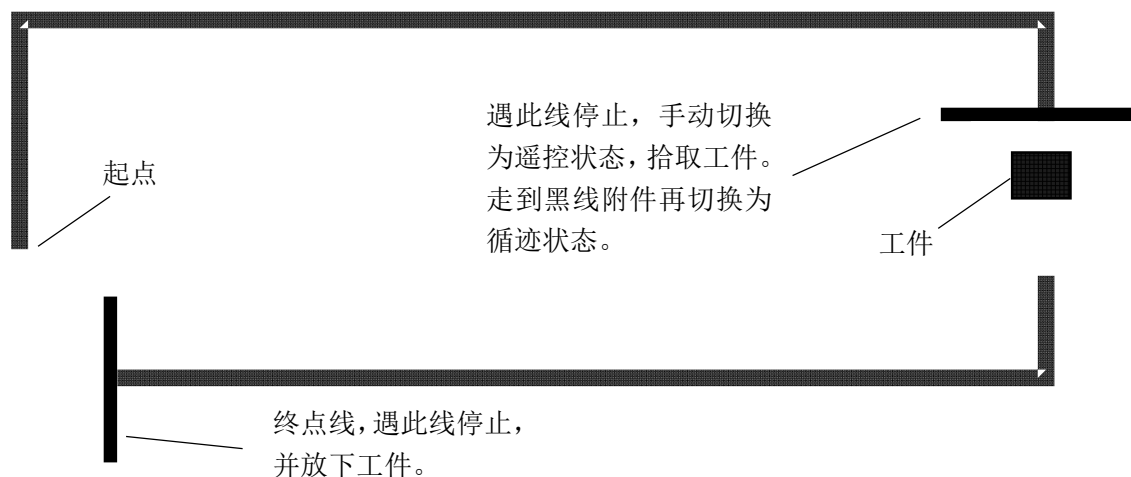
很容易看明白，串口命令“1”、“2”分别对应小车左转和右转，那么根据这个原理我们可以很容易实现用手机遥控小车前进、后退、左转、右转、停止等。同理也可以控制其他更加复杂的机构运动。

3. 自己尝试设置蓝牙按键和编程，实现更复杂的程序功能。

第十二课 排爆机器人汇报表演

表演内容：学生参考排爆机器人机构自行组装一个排爆机器人，独立程序设计完成指定场景任务。

场地场景内容：一个黑胶带围成的矩形场地（胶带宽度越宽越好，4cm~6cm 为宜）；一个方形或圆柱形工件（尺寸自由选择，边长或直径 3cm~5cm 为宜）。



比赛规则：1. 逐个机器人分别进行，老师也可根据场地情况重新确定机器人运行起止点；

2. 机器人从起点出发，沿黑线行驶，遇中线停止，然后手动切换为遥控状态，拾取工件。遥控继续向前走到黑线附件，再切换为循迹状态。遇终点线停止，并放下工件；

3. 在满足以上规则的前提下，在最短时间内机器人跑完全程者为第一名；

参考机构如下：

