

# 智能流水线装调

实践教学

前言.....	4
第一节课 机器人的零件.....	5
第 1 课时 认识零件.....	5
第 2 课时 基本结构.....	11
第二节课 控制的基础.....	14
第 1 课时 主控板简介.....	14
第 2 课时 配置编程环境.....	20
第三节课 简单实验操作.....	27
第 1 课时 控制电机.....	27
第 2 课时 白标传感器测值.....	32
第四节课 第一道工序—铸锭送料.....	36
第 1 课时 铸锭送料机构.....	36
第 2 课时 组装铸锭送料机构.....	38
第五课 第二道工序—传送带.....	42
第 1 课时 传送带原理.....	42
第 2 课时 组装传送带.....	44
第六节课 第三道工序—搬运机构.....	46
第 1 课时 搬运机构.....	46
第 2 课时 组装搬运机构.....	49
第七课 第四道工序—模拟加工机械臂.....	52
第 1 课时 模拟加工机械臂.....	52
第 2 课时 组装模拟加工机械臂.....	55
第八节课 第五道工序—码垛.....	57

第 1 课时 机械手爪.....	57
第 2 课时 组装机械手.....	60
第九课 3 自由度机械臂.....	62
第 1 课时 3 自由度机械臂.....	62
第 2 课时 组装 3 自由度机械臂.....	63
第十课 调试 3 自由度机械臂 ( 2 课时 ) .....	65
第十一课 ECT-IAM 智能流水线 ( 2 课时 ) .....	71
第十二课 调试流水线 ( 2 课时 ) .....	73

# 前言

流水线又称为装配线，一种工业上的生产方式，指每一个生产单位只专注处理某一个片段的工作，以提高工作效率及产量，整合生产工艺，可在流水线上布置多种工位，满足生产需求，它的可扩展性高，可根据工厂需求，设计符合产品生产需求的流水线，这样可以节约工厂生产成本，可一定程度上节约生产工人数量，实现一定程度的自动化生产，前期投入不大，回报率 率高。

ECT-IAM1105 是一款组件式的智能流水线设计平台，至少可组装一个 5 工序的流水线，并可根据每个模块的功能结合整体方案设计柔性生产线。产品提供了足够的机器零件和电子部件等，也提供了详细的组装方案，保证每台机器都可以独立运转，而控制系统可以保证生产下协调运作，从而营造出一个流水线模型。

ECT-IAM1105 可以让学生认识和操作流水线更好的锻炼学生实操经验，随意拆装、改造、设计柔性生产线，提高学生工程创新和应用能力。学校可根据不同年级，不同层次和不同的课程目的开设不同的实验课程，也可根据需要扩展其他配件，扩展实验平台的性能和功能。

# 第一节课 机器人的零件

## 第 1 课时 认识零件

“探索者”零件系统由一组经过高度综合与抽象的几何元素构成，可根据需要构建“点、线、面、体”，从而设计丰富多彩的机械结构。核心零件总数约 30 种，非常方便记忆和调用。“探索者”零件的材质是铝镁合金，是一种广泛应用于航空器制造的材料。特点是重量轻、硬度高、延展性好、可用于制作承力结构。采用冲压和折弯工艺，外表喷砂氧化，不易磨损，美观耐用。

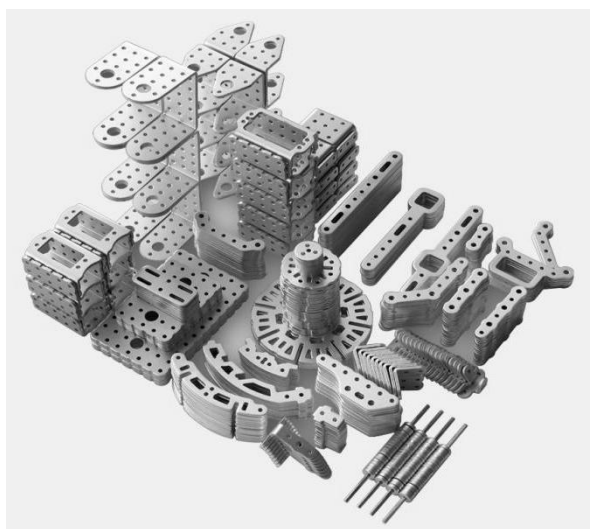


图 1.1 探索者零件体系

### 连杆类零件

连杆类零件提供了“线”单位。连杆类零件（图 1.2）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，构成桁架。

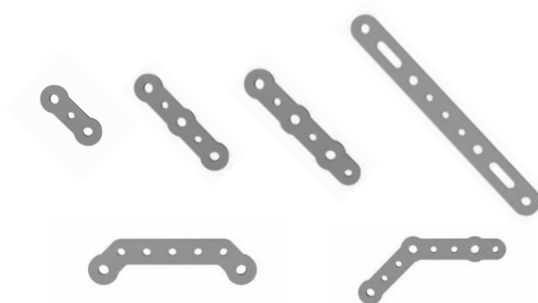


图 1.2 连杆类零件

## 平板类零件

平板类零件（图 1.3）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”。



图 1.3 平板类零件

## 框架类零件

框架类零件（图 1.4）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”。



图 1.4 框架类零件

## 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。

## 1. 常规传动零件

常规传动零件（图 1.5）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。

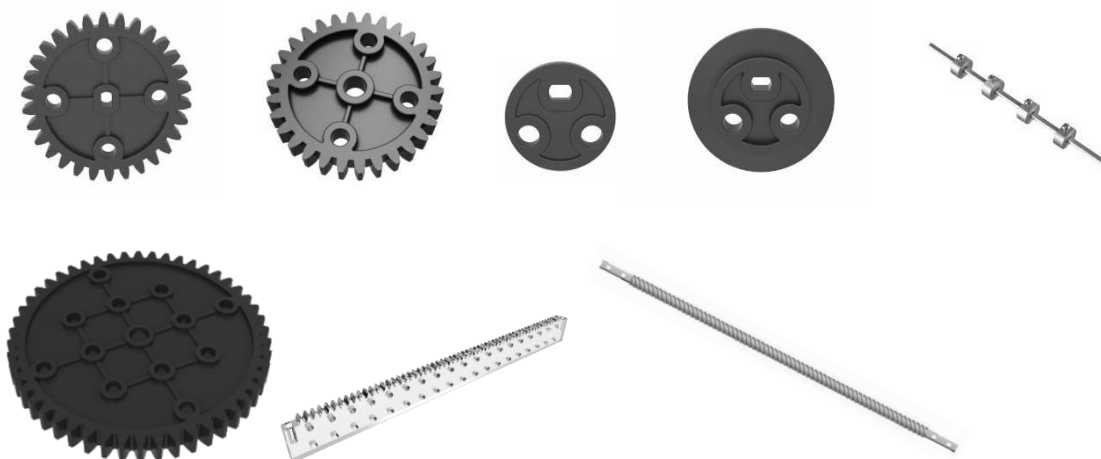


图 1.5 常规传动零件

## 2. 偏心轮连杆

偏心轮连杆（图 1.6）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。



图 1.6 偏心轮连杆

## 3. 电机相关零件

电机相关零件（图 1.7），电机周边的辅助零件包括电机支架、输出头和 U 型支架等。



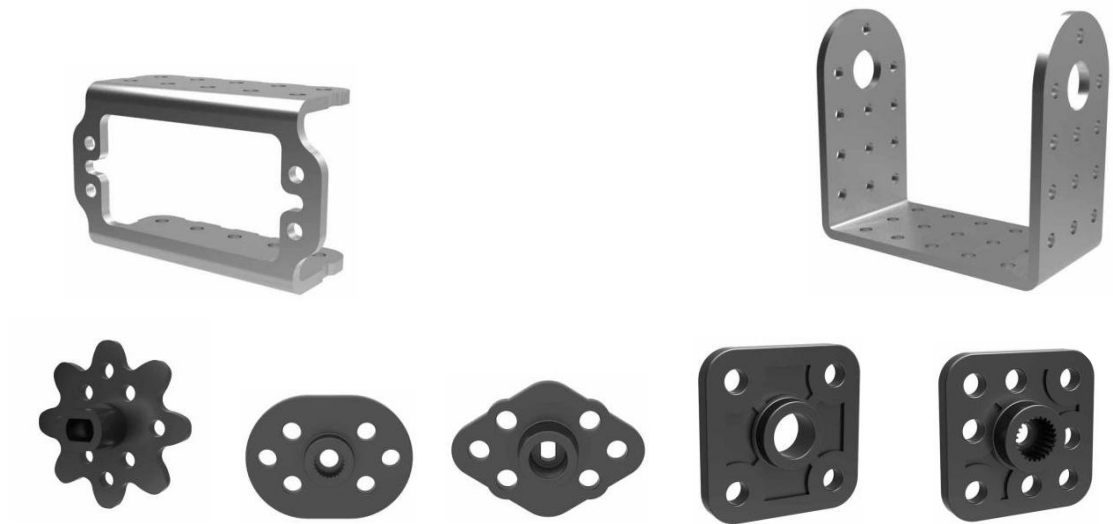


图 1.7 电机相关零件

#### 4. 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 1.8 所示。



图 1.8 轮胎相关零件

#### 5. 标准 5 金件

“探索者”所用连接件如螺丝、螺母等均为标准五金零件（图 1.9），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。

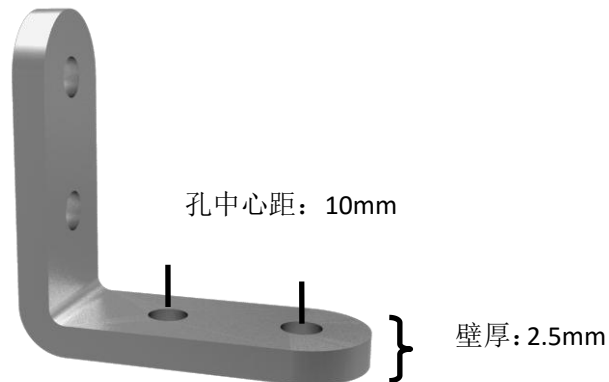


图 1.9 标准 5 金件



## 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 $\times$ 4”。如图 1.10 所示。



A 3D rendering of an L-shaped metal part. The part has a vertical flange on the left and a horizontal base on the right. Two circular holes are located on the horizontal base. Dimension lines indicate that the distance between the centers of these two holes is 10mm. A bracket on the right side of the horizontal base indicates that the thickness of the part is 2.5mm.

图 1.10 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 1.11。

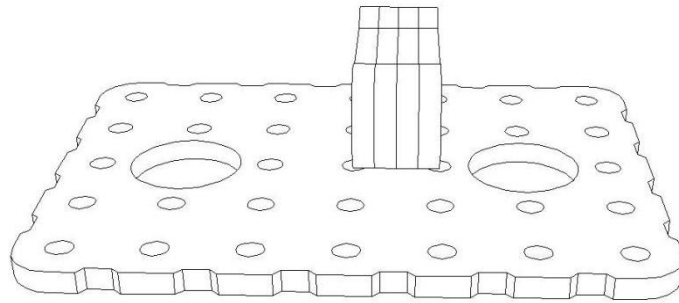


图 1.11 零件空间关系示意图

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 1.12）

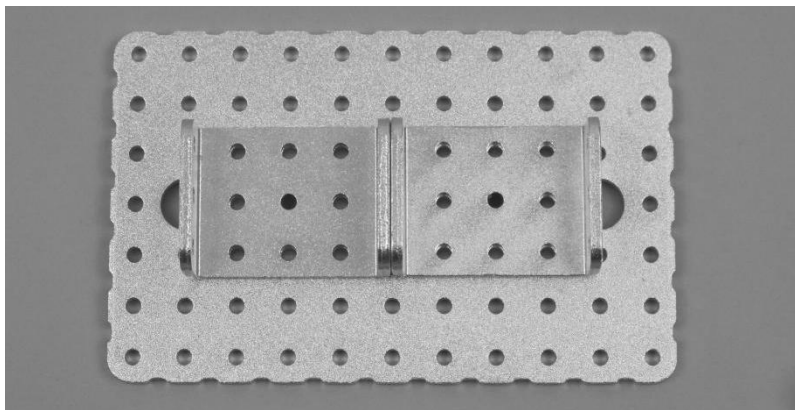


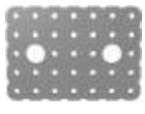







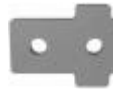










图 1.12 零件孔位置配合关系

表 1.1 核心零件列表

				
J01	J02	J03	J04	J05
10mm 滑轨	3×5 双折面板	5×7 孔平板	7×11 孔平板	90 度支架
				
J07	J08	J09	J11	J12
大轮	垫片 10	垫片 20	机械手 20mm	机械手 40mm
				
J13	J14	J15	J17	J18
机械手驱动	机械手指	轮支架	输出支架	双足大腿
				
J19	J20	J21	J22	J23
双足脚	双足连杆	双足腿	双足小腿	双足支杆
				
J24	J25	J27	J28	J29
四足连杆	小轮	31 度球片	72 度球片	球支架片
				
J30				
11*25 孔平板				

## 第 2 课时 基本结构

零件体系的特性大家已经有所了解，我们再来看看这些零件在实际组装中的表现。我们把可组装的基本结构分为刚体结构、可动结构两类。当然这只是一种粗略的划分，有助于大家整理思路，快速把握“探索者”的设计要领，实际组装中并不需要区分的那么严格。

### 刚体结构

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（图 1.13）。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。



图 1.13 零件的固定

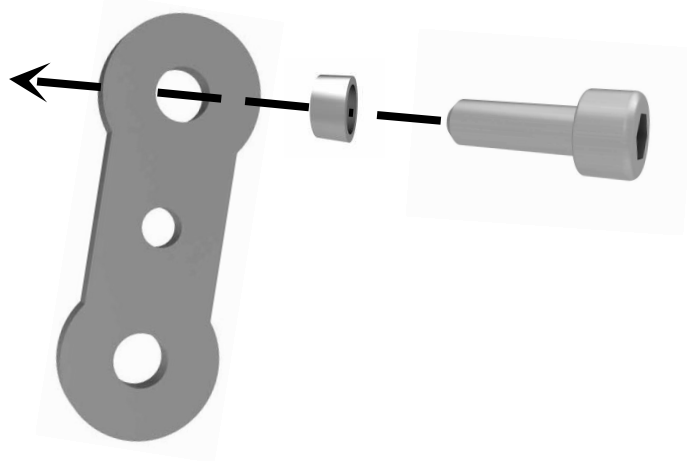
做一做

参照下图，利用所学知识组装一个小椅子。



## 可动结构

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，如图 1.14。



长短不同的轴套，它主要起到铰接的作用，可以让两个连接的零件转动起来。



图 1.14 轴套的用法

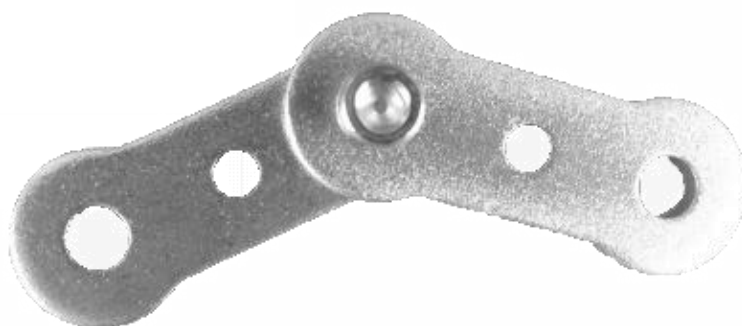
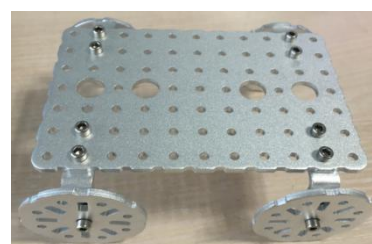


图 1.15 可动结构的外观

### 做一做

参照右图，同学们自己动手组装一个平板车，实现用手推小车，小车能够向前运动。

提示：注意轴套的使用。



## 问题与练习

- 1.想一想，在组装的过程中，什么时候可以用到轴套，什么时候可以用不同轴套？
- 2.螺丝的长度你是如何选择？
- 3.尝试组装一个自平衡双轮代步车：



## 第二节课 控制的基础

### 第 1 课时 主控板简介

介绍 Basra 主控板之前有必要介绍一下 Arduino——这个在全世界掀起了“硬件复兴”运动、导致“创客”群体崛起，引领智能设备开发潮流的神奇硬件。

Arduino 是一个能够用来感应和控制现实物理世界的一套工具。它由一个基于单片机并且开放源码的硬件平台，和一套为 Arduino 板编写程序的开发环境组成。它的编程语言就像似在对一个类似于物理的计算平台进行相应的连线，它基于处理多媒体的编程环境。

有很多的单片机和单片机平台都适合用做交互式系统的设计。例如：ParallaxBasicStamp，Netmedia' sBX-24，Phidgets，MIT' sHandyboard 和其它等等提供类似功能的。所有这些工具，你都不需要去关心单片机编程繁琐的细节，提供给你的是一套容易使用的工具包。Arduino 同样也简化了同单片机工作的流程，但同其它系统相比，Arduino 在很多地方更具有优越性：便宜、跨平台、简易的编程环境、软件开源并可扩展、硬件开源并可扩展。

#### 主控板简介

Basra (图 2.1) 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。

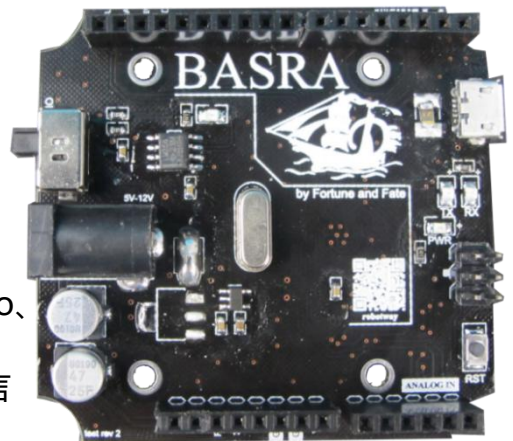


图 2.1 Basra

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。

Basra 可作为机器人的控制核心，用于控制舵机、直流电机、传感器、输出模块、通信模块等。其接口如图 2.2

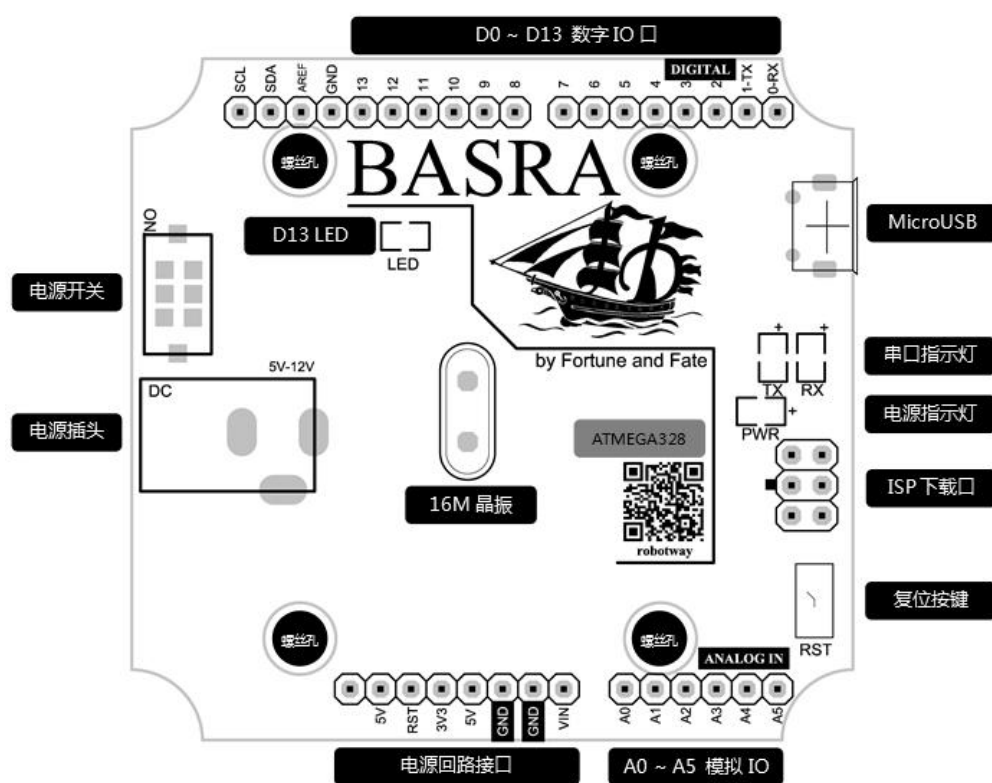


图 2.2 Basra 接口图

## Bigfish 扩展板简介

Basra 是开源的控制板，非常适合爱好电子制作的朋友制作互动作品，但对于一些不熟悉电子技术的爱好者，要在控制板上添加电路是一件比较麻烦的事，所以我们设计了一批专用于简单机器人的扩展板：Bigfish 综合扩展板、Birdmen 手柄扩展板、Seahorse 电机扩展板等，能将大部分传感器、电子模块、电机等轻松的和 arduino 控制板连接。其中 Bigfish 功能最综合，最适合日常使用，尤

其是入门学习使用。

通过 BigFish 扩展板（图 2.3）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，可以说的 arduino 控制板的必备配件。通常与 Basra 主控板堆叠连接，如（图 2.4）

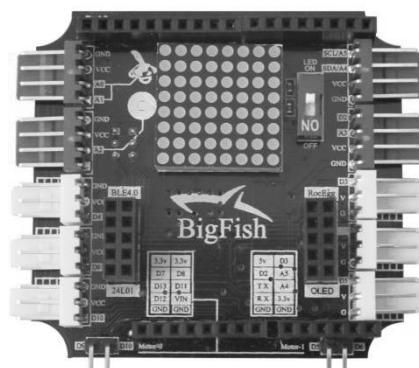


图 2.3 Bigfish 扩展板。

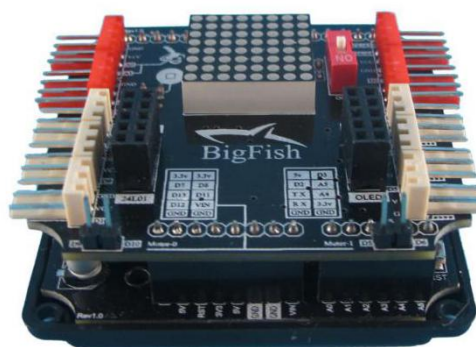
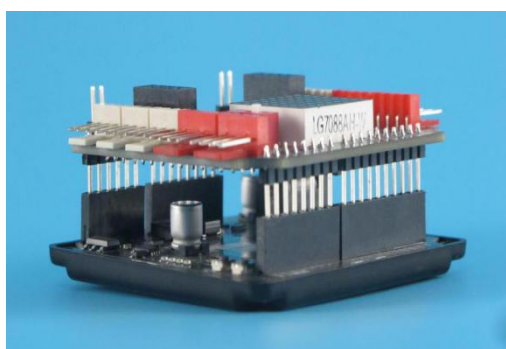


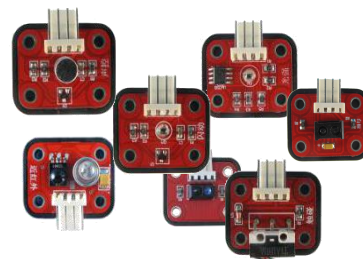
图 2.4 Basra 主控板与 BigFish 扩展板堆叠

## 传感器简介

传感器是一种检测装置，可以用来检测外部环境，获取相应的参数，按一定规律转换成电信号或其他所需形式的信息输出，以满足信息的传输、处理、存储、显示、记录和控制等要求，相当于机器的感官。

传感器可粗略分为数字量传感器和模拟量传感器。

数字型传感器，只能返回 0 或 1，也就是高电平信号或者低电平信号，比较类似一个电源的开或关，所以也被称作开关量传感器。这类传感器都是低电平触发，也就是说，触发时产生一个低电平信号；换句话说，传感器发出低电平信号时，主控板将这个信号标为 1，高电平时为 0。





模拟量传感器能够检测到连续信号，给出一大串数字。有了一大串数字，那么我们就可以设置传感器更多的触发条件，比如某个传感器可以检测到 0~1023 的连续数值，那么我们可以设置 0~100 之间触发功能 1，100~500 之间触发功能 2，500~1024 之间触发功能 3。因此模拟量传感器使用起来功能更加强大。

## 白标传感器

白标传感器（如图 2.5）既是数字量传感器又是模拟量传感器，可以帮助进行白线的跟踪，可以识别黑色背景中的白色区域，或悬崖边缘。

寻线信号可以提供稳定的输出信号，使寻线更准确更稳定。有效距离在 0.7cm~3cm 之间。

工作电压：4.7~5.5V，工作电流：1.2mA。

①固定孔，便于用螺丝将模块固定于机器人上

②四芯输入线接口，连接四芯输入线

③白标传感器元件，用于检测白线信号

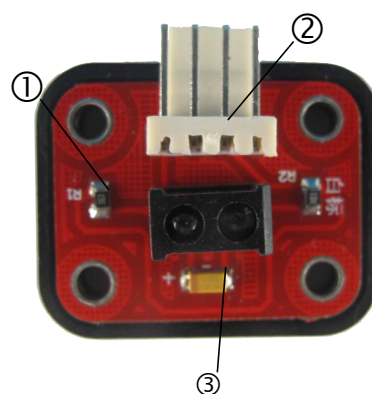
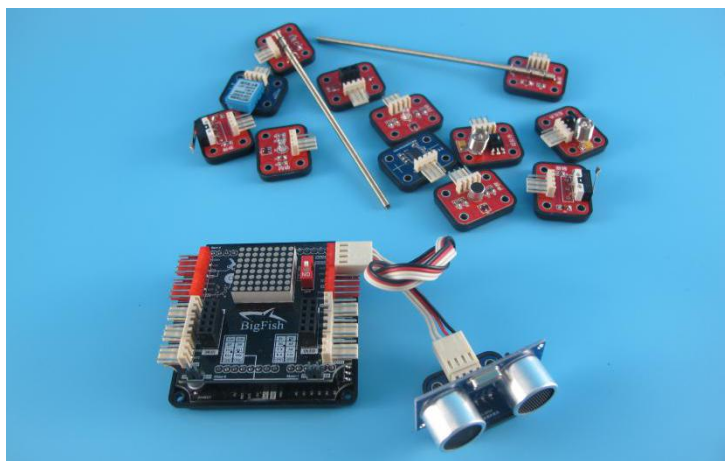


图 2.5 白标传感器

**工作原理：**图为灰度传感器，③号元件是一个红外发射/接收管（蓝发黑收），可以发射红外线并接收反射的红外线，如果目标颜色较深，红外线就会被吸收，从而触发。因此，如果目标是开阔空间，没有红外线反射回来，也会触发。白标触发原理与灰度相反。

**注意事项：**白标传感器的安装应当贴近地面且与地面平行，使用前最好测试一下触发距离，这样才能更加灵敏并且有效的检测到信号。



Bigfish 扩展板与常规传感器连接：用 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）。

## 电机简介

电机用于提供动力，“探索者”最常用电机包括双轴直流电机和标准伺服电机（舵机）。直流电机提供圆周运动，伺服电机提供摆动。

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。

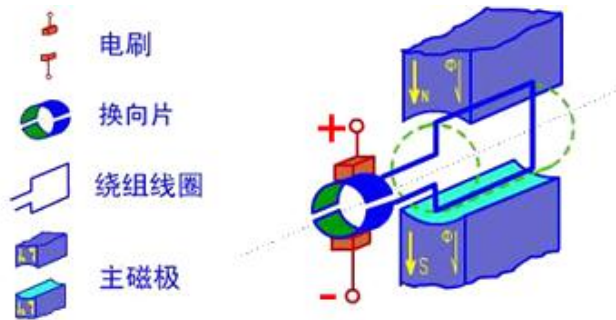


图 2.6 直流电机工作原理

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 2.6）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。

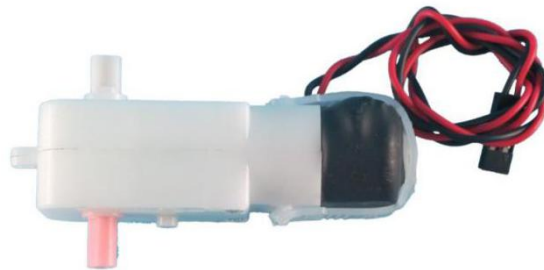


图 2.7 直流电机

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

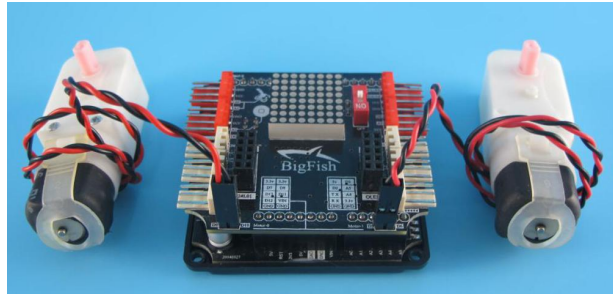


图 2.8 直流电机连接扩展板

“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（图 2.8），标准舵机的用途是制作摆动机构，如关节模块等。参数见表 2.1。

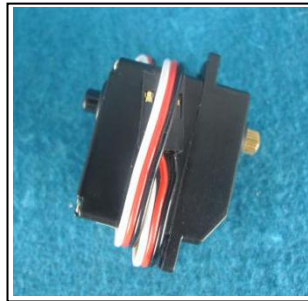
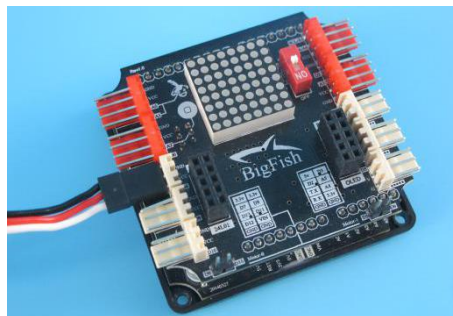


图 2.9 标准舵机

表 2.1 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

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。



## 第 2 课时 配置编程环境

编程环境（IDE）简单说就是用来写程序的软件，我们采用的是 arduino 官方 IDE。这个 IDE 有很多版本，我们选择最后一个绿色安装版本 1.5.2（图 2.9）进行说明，该软件直接拷贝即可使用。其他的版本大家可以自己尝试。

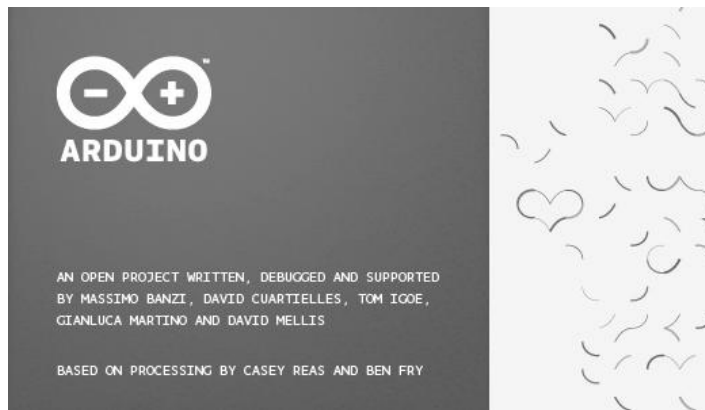


图 2.9 Arduino 官方的编程 IDE 绿色安装版本 1.5.2

### 安装驱动

由于已经将 Arduino IDE 安装好了，下面就将 Basra 主控板与计算机连接来。首先，你需要用一根 USB 线，一头接 Basra 主控板，一头插入计算机的 USB 接口中。在将 USB 线插入开启的计算机的时候，计算机就开始为 Basra 板供电，同时，板子上的 LED 灯会发亮。

将 Basra 控制板通过 USB 数据线与 PC 连接，初次连接时会出现 Windows 自动安装、弹出硬件更新向导、看不到任何现象等情况。当出现弹出硬件更新向导请按照以下步骤完成安装驱动的操作。



选择“否，暂时不”



选择“从列表或指定位置安装（高级）”

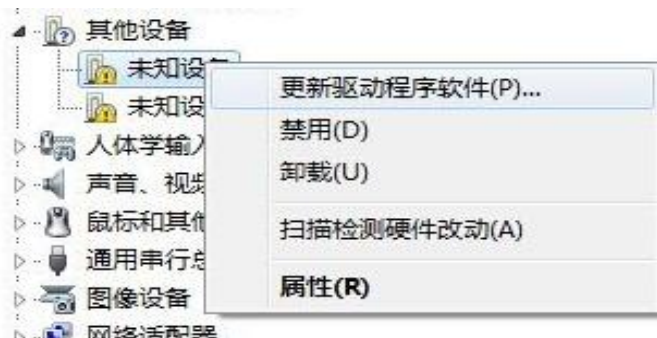


安装路径选择“arduino-1.5.2\drivers”，选中“FTDI USB Drivers”文件夹，点击“确定”



点击“完成”如果感觉没有安装成功，就重新安装一遍。

当 Basra 控制板通过 USB 数据线与 PC 连接，看不到任何现象时，请在“我的电脑”上点右键，选择“管理”，在“管理”中打开“设备管理器”。在端口列表中，会看到黄色感叹号（图 2.10）。在黄色感叹号设备上点右键，选择“更新驱动程序软件”，接下来的做法与弹出更新向导的步骤相同。



完成安装后，打开设备管理器，在端口（COM 和 LPT）列表中，出现 USB Serial Port (COMx)（x 是数字用来表示端口号）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 COM8。



## 设置选项参数

Arduino IDE 支持各种不同型号的板子，所以，你需要告诉 IDE 你所用于程序开发的是那种型号。图 2.11 给出了完成单片机板选择操作的菜单命令（Tools→Board）。在这实例中，选择的是 Arduino Uno。

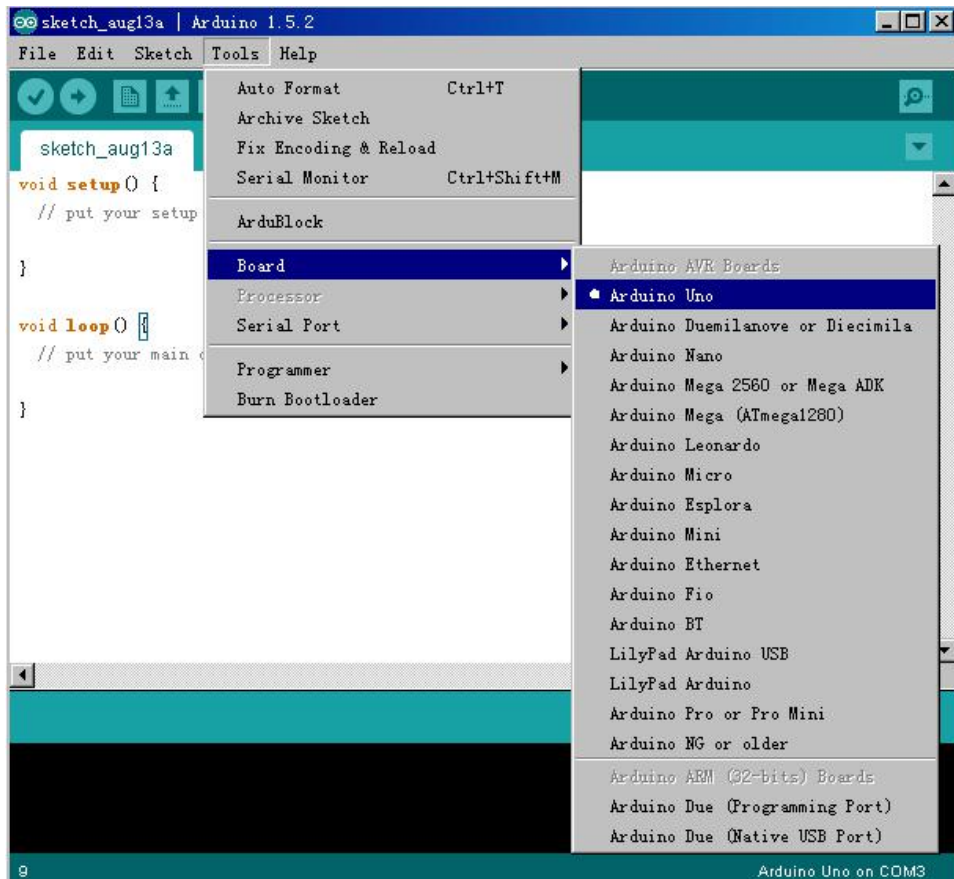


图 2.10 选择你的单片机型号

你所指定的单片机板类型需要与你实际使用的相匹配。如果今后更换了单片机板，只需要再次使用这个菜单进行切换即可。

IDE 能够很好地识别你所选择用来为单片机板供电并与只进行通信的 USB 端口。要确定使用了那个端口，只需要执行菜单命令 Tools→Serial Port 即可，如图 2.13 所示，在这个实例中，用于与单片机板进行通信的是 COM 端口 3。

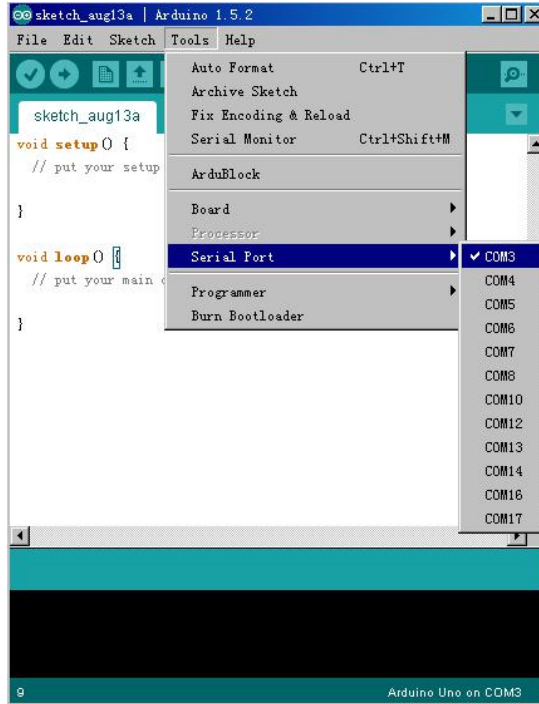
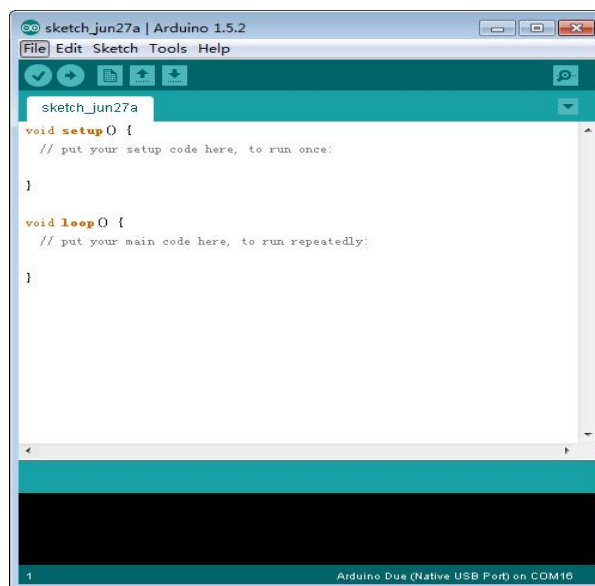


图 2.11 端口的选择

## 认识编程界面

Arduino 有两个编程界面，一个是 C 语言界面，一个是图形化界面。打开 Arduino1.5.2 后就是的 C 语言界面，如图 2.14 所示，由于已经做好了编程模板，所以看不见 main ( ) 函数，界面上所显示的代码都在 main ( ) 函数里面。初始化部分 setup ( ) 和循环程序部分 loop ( ) 的框架也已经存在了。



Arduino 的 IDE 是编译和烧录一体化的，因此我们不再需要专门的烧录软件去烧录 hex，只要点击顶部菜单栏左下方的√按钮，是编译，→按钮是编译+生成+烧录。

在 C 语言界面上点击菜单栏的 Tools→Ardublock 即可打开，这是一个由国内创客开发的插件，中文界面，使用起来非常方便。与常见的流程图编程方式不同，这个图形化界面是严格的 C 语言结构，如图 2.12。



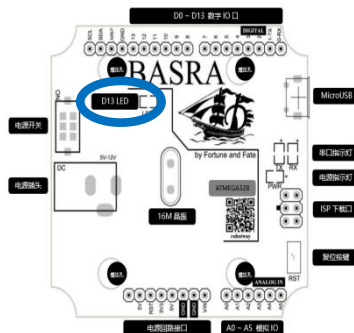
可以从左侧的菜单栏中拖出需要的程序图块，像拼图一样把程序拼接出来。  
不需要的图块，拖回左侧即可删除。  
点击上传到 Arduino 进行烧录。

图 2.12 图形化界面

## 做一做

实验操作：Blink（控制 LED 灯闪烁）

本次实验控制的 LED 位于如图所示位置，针脚号为 D13。我们要编程使其闪烁。





## 操作步骤

- 1.连接：将 Basra 主控板通过 USB 线连接到 PC。
- 2.编程：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。

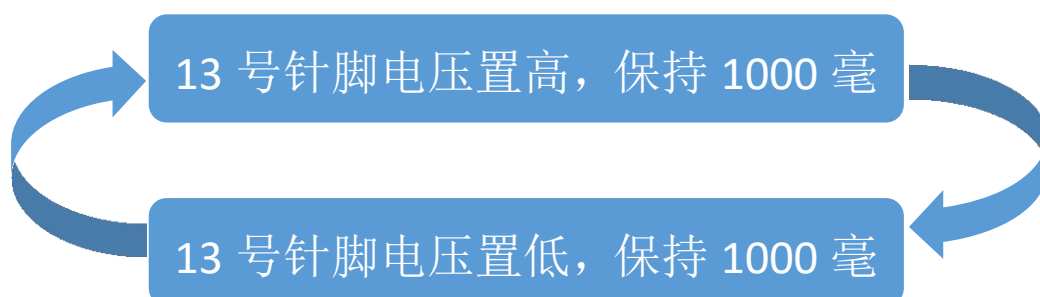


## 3.烧录

图形化程序拼接完成后，确认已选择好 Board 和 Serial Port ，点击“上载到 Arduino”按钮或 C 语言界面 `upload` ，即  编译并烧录。同时，还会在 C 语言界面生成对应的 C 语言的代码，供使用者对比学习

## 4.烧录成功后观察效果。

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 秒之后再去做”，而是说该语句上面所设定的状态，即“引脚 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。



### 思考与练习

- 1.想一想，使用 C 语言界面编程，该如何进行烧录？
- 2.尝试改变延迟参数，说说你看到的效果？
- 3.要想使 LED 灯闪烁的频率不同，该如何编程，试着自己进行编程，完成该实验操作。

## 第三节课 简单实验操作

### 第 1 课时 控制电机

#### 实验操作一：控制直流电机

在这个实验中，我们首先将直流马达输出头安装在直流电机的粉色输出轴上，利用 Basra 主控板、BigFish 扩展板搭建控制电路，使用图形化界面编程或 C 语言界面编程，从而控制直流电机的转动。

将主控板连上 PC，将直流电机连在 5/6 或 9/10 端口上（如图 3.1）。下面的例程将以 9/10 端口为例。

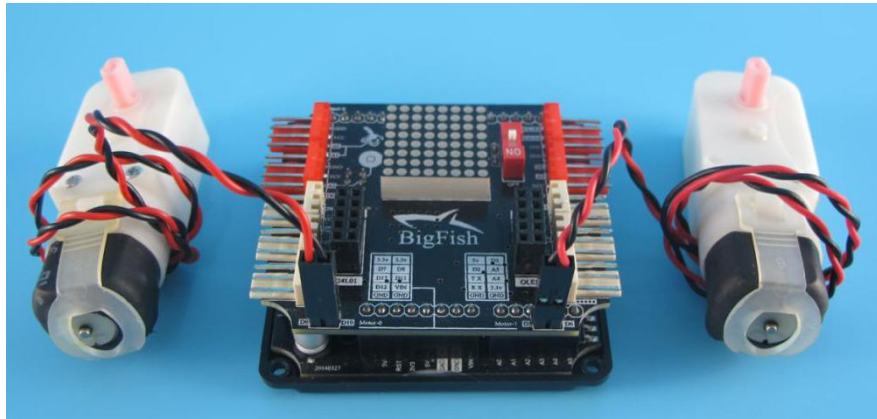
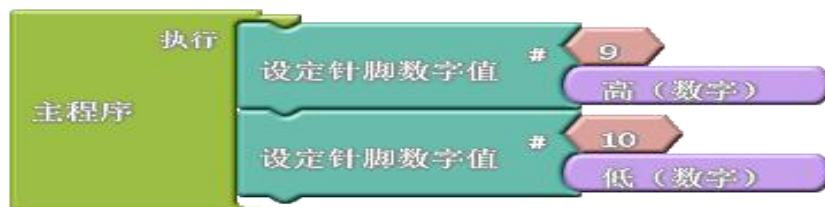


图 3.1 直流电机连接扩展板 5/6、9/10 端口上

#### 参考例程一



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。

对应的 C 语言程序如下：

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

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

### 思考与讨论

- 1.参考例程中是针脚 9 高 10 低，如果反过来 9 低 10 高效果会如何，同高同低效果又是什么？观察并记录直流电机的转动方向。
- 2.请尝试加入延迟语句，直流电机转动的效果是怎样的？

参考例程二



对应的 C 语言程序如下：

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

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

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 D9/D10，那么程序就可以有“9 高 10 低”、“9 低 10 高”、“9 低 10 低”，分别对应“转”、“反转”、“停”。而“9 高 10 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。

## 实验操作二：控制伺服电机

在这个实验中，我们首先将输出头安装在伺服电机的输出轴上，利用 Basra 主控板、BigFish 扩展板搭建控制电路，使用图形化界面编程或 C 语言界面编程，从而控制伺服电机的转动。

将主控板连上 PC，将直流电机连在 4 端口上（如图 3.2）。下面的例程将以 4 号端口为例。

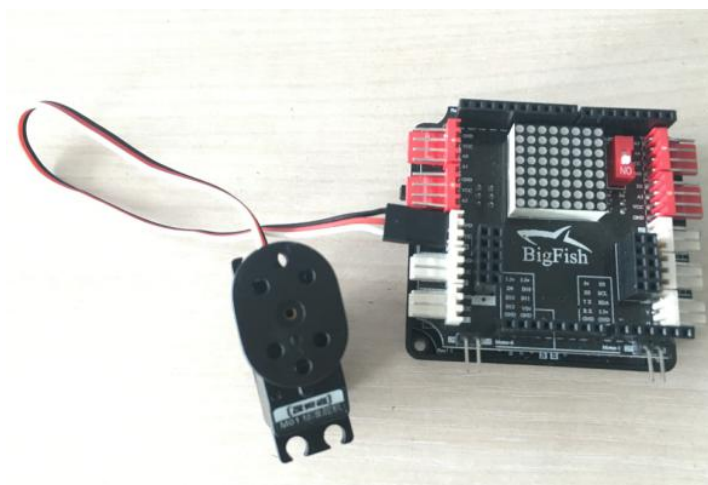


图 3.2 伺服电机连接在扩展板 4 号端口

## 参考例程



上图程序将实现的功能：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。

对应的 C 语言程序如下：

```
#include <Servo.h>
```

```
Servo servo_pin_4;
```

```
void setup()
```

```
{
```

```
    servo_pin_4.attach(4);
```

```
}
```

```
void loop()
```

```
{
```

```
servo_pin_4.write( 120 );  
  
delay( 1000 );  
  
servo_pin_4.write( 60 );  
  
delay( 1000 );  
  
}
```

#### 注意事项

- 1.舵机的黑线一定要接在 GND 口！不要带电操作！
- 2.由于电脑 USB 口电压不足，所以本实验请用电池供电。

#### 练习

1. 请尝试改变伺服电机的端口或延迟参数，观察伺服电机的转动情况。

## 第 2 课时 白标传感器测值

白标传感器既是数字量传感器又是模拟量传感器，可以检测数字量（两个值，0 或 1）和简单模拟量，复杂传感器可以检测复杂的模拟量数据，如距离、温度、角度、颜色等。

传感器检测到的数字量和模拟量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。

将主控板连上 PC，将白标传感器接在 A0 端口上 ( 如图 3.3 )。下面的例程将以 A0 引脚为例。

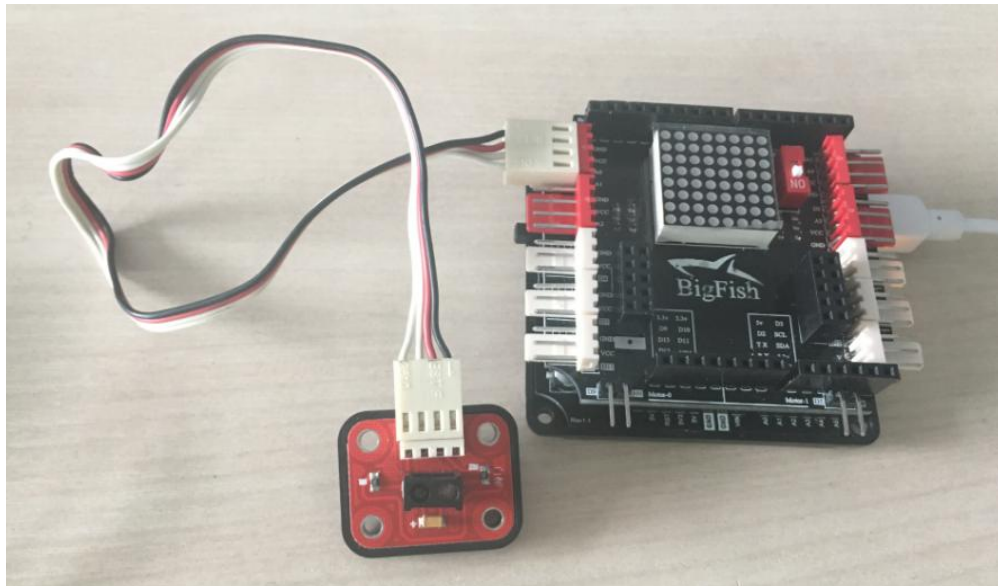


图 3.3 传感器的连接

### 检测数字量

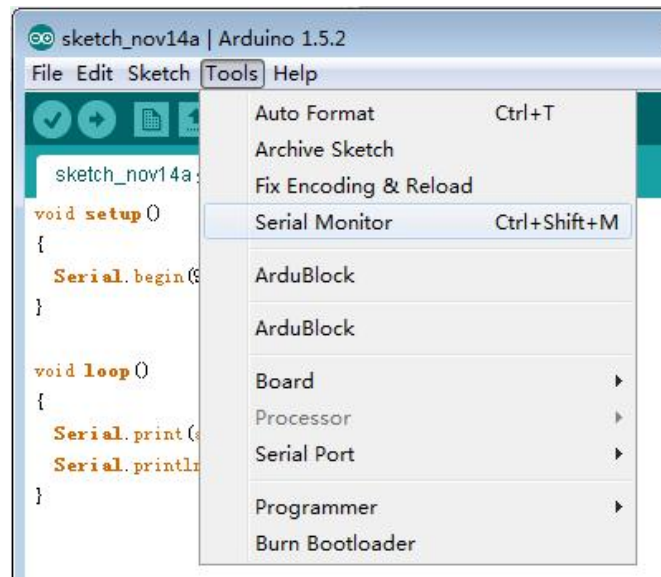


程序烧录后，在图形化界面的上方最右侧，点击 serial monitor 按钮，可打开串口监视器。

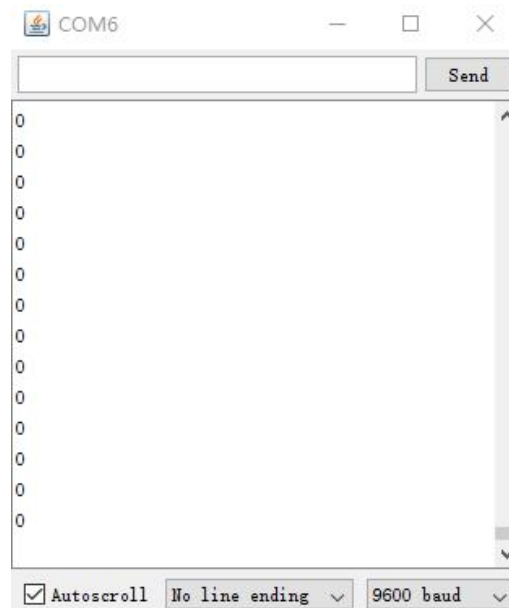


在 C 语言界面的 Tools 菜单里面，也可以找到 serial monitor 选项。





通过该窗口我们即可看我们检测的数据，同学们请自己动手操作，然后观察检测到的数据。



在编程过程中特别需要注意的是，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（如图 3.4）

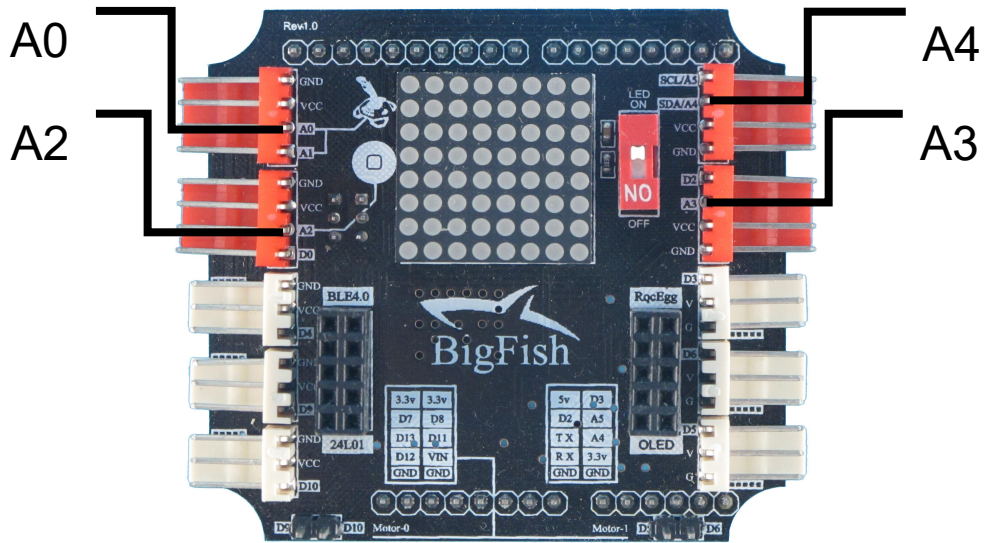


图 3.4 传感器对应的端口号

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  $3+14=17$

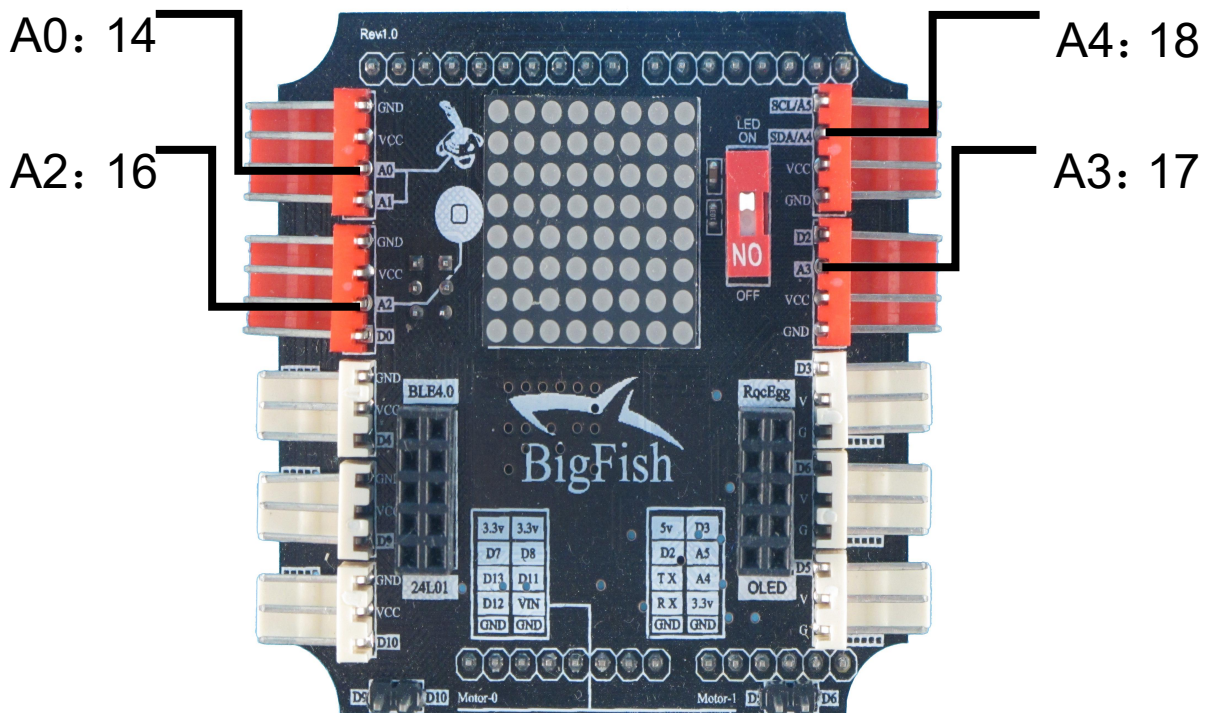
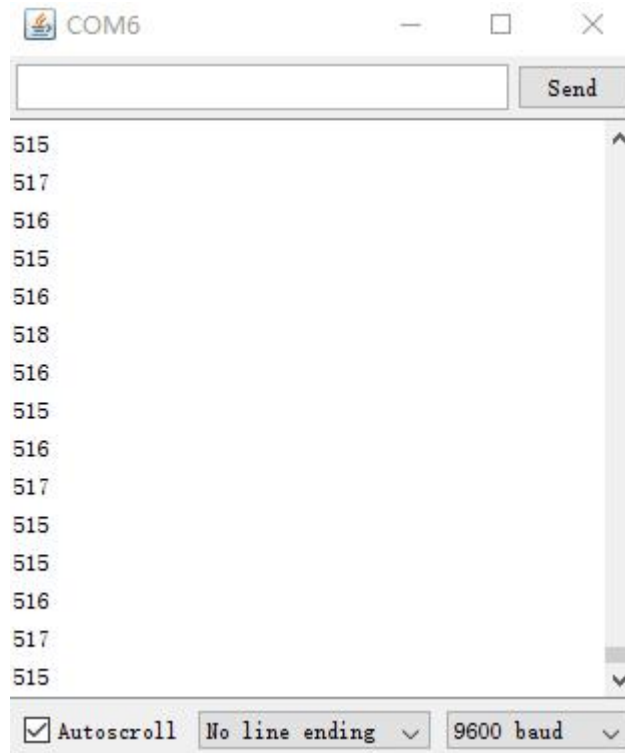


图 3.5 图形化编程对应端口号

## 检测模拟量



程序烧录后，打开串口监视器，即可看到检测到的数据。



读取模拟量时可以获得物体的灰度（深浅不一的灰色）参数。

通过该窗口我们即可看我们检测的数据，同学们请自己动手操作，然后观察检测到的数据。

## 第四节课 第一道工序—铸锭送料

### 第 1 课时 铸锭送料机构

ECT-IAM 是一款组件式的智能流水线设计平台，至少可组装一个 5 工序的流水线，并可根据每个模块的功能结合整体方案设计柔性生产线。每台机器都可以独立运转，而控制系统可以保证生产下协调运作，从而营造出一个流水线模型（如图 4.1）。

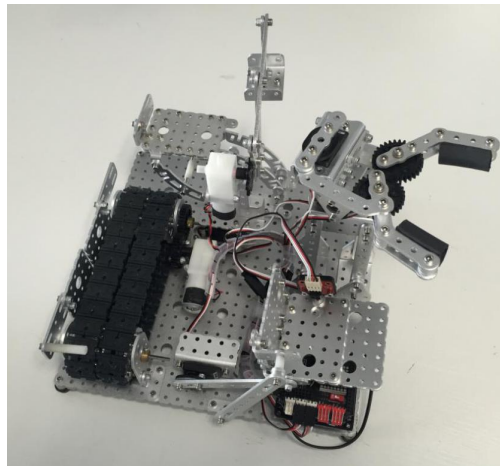


图 4.1 流水线模型

#### 铸锭送料结构

铸锭送料结构（如图 4.2）是 ECT-IAM 智能流水线的第一道工序，可以独立运转，其工作特点是为粗实线位置为炉铸锭进入装料器 4 中，装料器 4 即为双摇杆机构 ABCD 中的连杆 BC，当机构运动到传送位置时，转料器翻转把铸锭卸放到下一道工序。

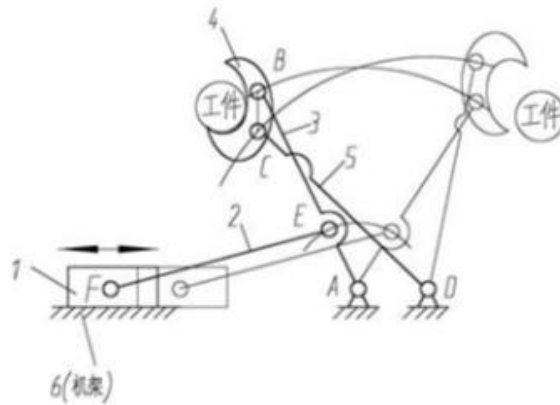


图 4.2 铸锭送料结构的结构图

结构说明：平动转摆动——这是一个滑块四连杆机构，滑块 A 作为驱动部件，水平移动，通过连杆 AD 将运动传递给 CE 杆，当滑块 A 到一定位置时，CE 杆的位置和 BF 杆的位置交换，这样实现了 E 点和 F 点的位置交换，最终输出杆 EF 表现为摆动。

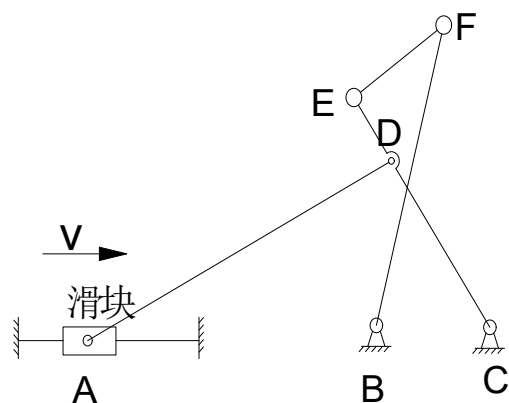
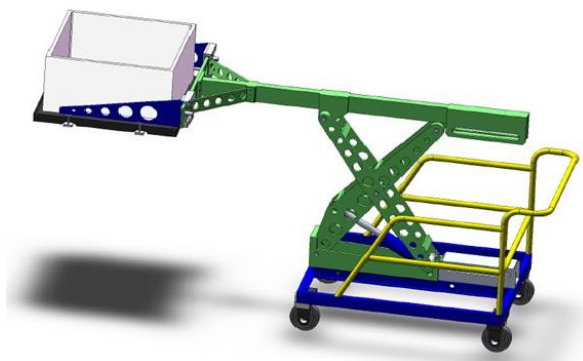


图 4.3 铸锭送料机构原理图

## 铸锭送料机构的应用

铸锭送料机构在工业生产中有广泛应用。加热炉出料设备、加工机械的上料设备等都应用了此技术。这一技术的应用为生产、加工带来了很大的便利，极大的提高了工作效率、减少了人工成本、同时也能减少工人在工作中的危险系数。



## 第 2 课时 组装铸锭送料机构

铸锭送料机构（如图 4.4）的工作特点及原理同学们已经有所了解，本节课的主要任务就是将其组装出来。对于 ECT-IAM 流水线的学习我们是通过模块化思维，一道工序一道工序的认识，那么对于铸锭送料机构的组装我们同样可以应用这种思维。

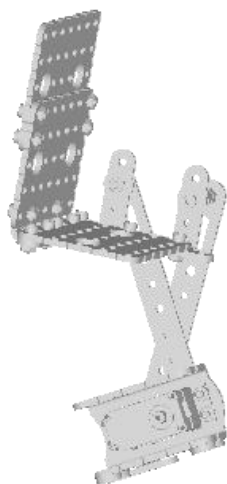
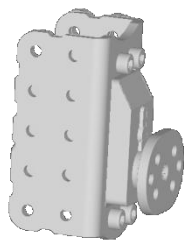


图 4.4 铸锭送料机构

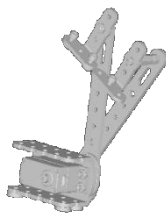
### 思考与讨论

- 1.想一想，组装铸锭送料机构，你想从哪部分开始？
- 2.讨论，你想讲铸锭送料机构分为几部分进行组装？

参照 stp 图，搭建一个铸锭送料结构。提示：根据模块化思维，我们可以将其分为电机、双摇杆机构、装料器三部分进行组装，有想法的同学也可以根据自己的思路进行组装。



安装伺服电机



组装双摇杆



组装装料器

通过前面所学的知识，我们已经知道如何控制伺服电机的转动,但是其运行效果正如我们之前所观察的较生硬，我们可以用“重复”语句进行优化。

“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 );  
}
```

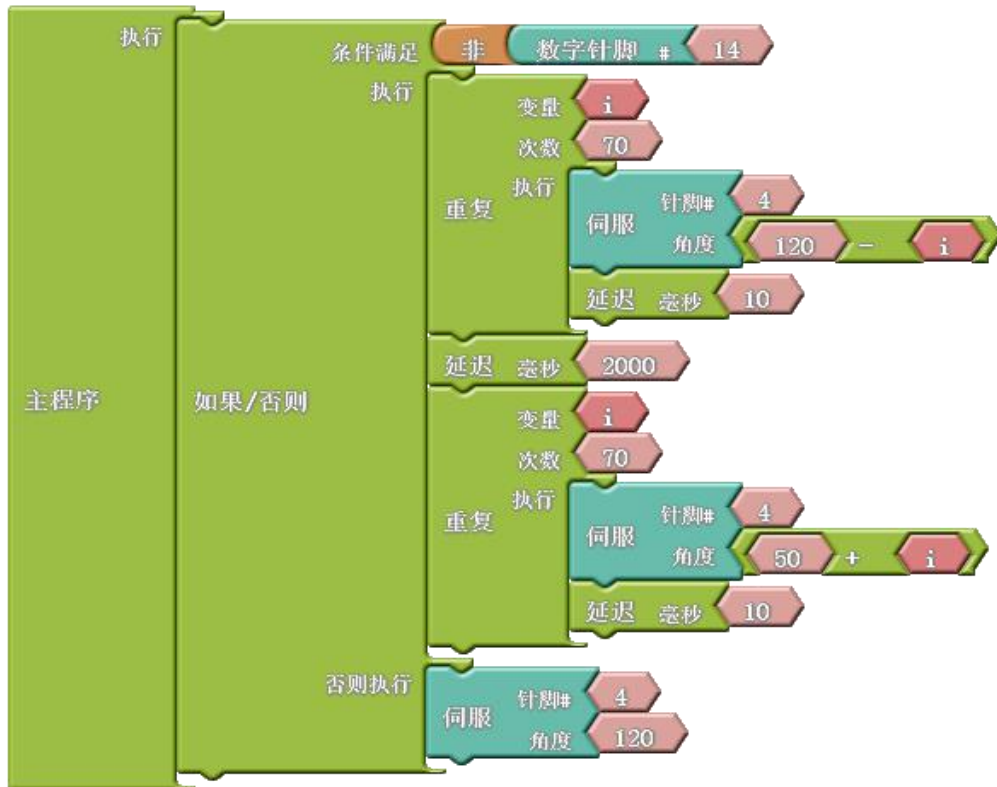
由于自动生成的 C 语言例程变量名比较长，为了方便阅读，上边的 C 语言例程做了简化。

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。

# 小实验

实验内容：控制铸锭送料机构转动（利用 for 循环）

参考例程：



对应的 C 语言程序：

```
#include <Servo.h>
```

```
int _ABVAR_1_i = 0 ;
```

```
Servo servo_pin_4;
```

```
void setup()
```

```
{  
  pinMode( 14, INPUT);  
  servo_pin_4.attach(4);  
}
```

```
void loop()
```

```
{  
  if (!( digitalRead(14) ))
```



```

{
  for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 70 ); _ABVAR_1_i++ )
  {
    servo_pin_4.write( ( 120 - _ABVAR_1_i ) );
    delay( 10 );
  }
  delay( 2000 );
  for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 70 ); _ABVAR_1_i++ )
  {
    servo_pin_4.write( ( 50 + _ABVAR_1_i ) );
    delay( 10 );
  }
}
else
{
  servo_pin_4.write( 120 );
}
}

```

该例程实现的效果：让舵机从 120° 起，角度参数每 10 毫秒减 1，直到 50°。保持 2s 后在让舵机从 50° 起，角度参数每 10 毫秒加 1，直到 120°。

传感器的触发条件是“低电平”，而  语句意思是

“14 号数字针脚获得高电平”，因此要在前面加一个逻辑运算符



同学们可以根据舵机转动的效果，自行改变角度、次数等参数使铸锭送料结构转动到你认为合适的位置。

## 第五课 第二道工序—传送带

### 第 1 课时 传送带原理

传动装置在工业、民用、工程中的应用都很广泛，常见的传动方式有链条传动、摩擦传动、液压传动、齿轮传动以及皮带式传动等。ECT-IAM 流水线采用的是皮带式传动。

#### 机构原理

带传动是一个将电机的转动转化为平动的机构，该机构的移动稳定性很强，但是由于皮带或履带的都是柔性的，强度较小，所以不能承受太大力，适合用于配备重量不大的物体的运输机构中，或者只是作为一个运动传递的机构也是不错选择。

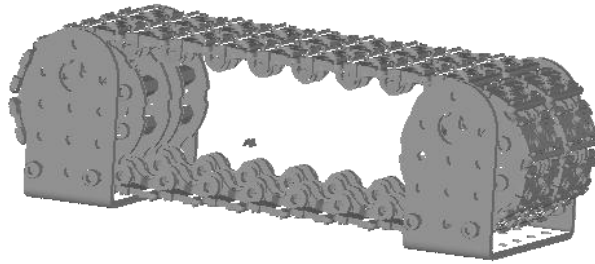


图 5.1 履带传送

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 5.1 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。

## 传送带的应用

传送带在生活中的应用比比皆是，无论是工业、农业、制造业还是日常生活所需，传送带都发挥着重要的作用。例如：商场的自动扶梯，机场的自动人行道、码头上自动装卸货传送带、工厂生产流水线，农业机械中（联合收割机、插秧机）都有应用。



传送带的应用为人们的生活和生产带来了很大的便利，在生活中，节约了人们的时间、精力；在工业生产系统，传送带的应用不仅节约了劳动力，提高了生产效率，而且降低了生产成本，在工业生产中发挥了巨大的作用。未来传送带设备的将向着大型化发展、扩大使用范围、物料自动分拣、降低能量消耗、减少污染等方面发展。

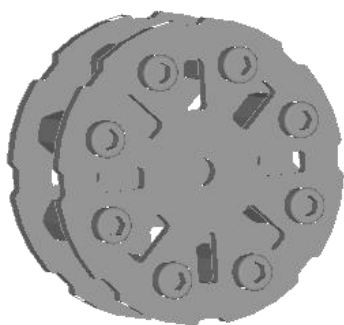


## 第 2 课时 组装传送带

传送带的工作特点及原理同学们已经有所了解，本节课的主要任务就是将其组装出来。

参考提供的 stp 图，搭建一个传送带结构。提示：根据模块化思维，我们可以按照电机驱动、传动轮、履带三部分进行组装，有想法的同学也可以根据自己的思路进行组装。

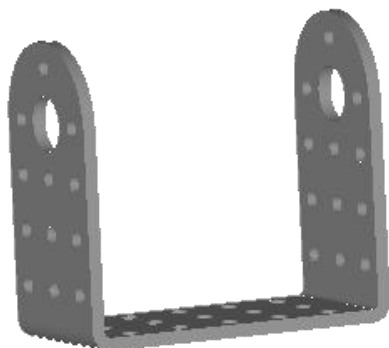
### 组装步骤参考



组装传动轮



安装履带



大舵机 U 型支架



传动轴



马达后盖输出头

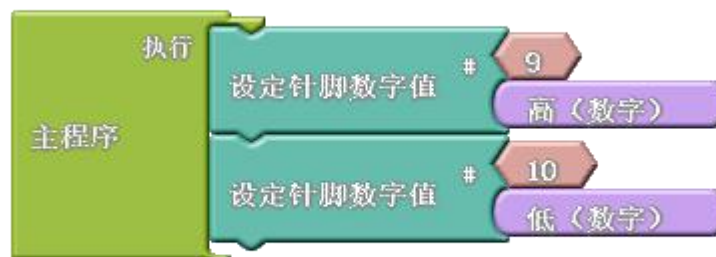
最后用 U 型支架和传动轴将传动轮和履带固定，详细步骤请参照 stp 图。

# 小实验

我们通常采用数字量和模拟量控制传送带的运动，数字量只有 0 和 1(即未触发和触发)，用它来控制电机，则电机将以一定的速度转动；模拟量是指变量在一定范围连续变化的量也就是在一定范围内可以取任意值，所以用它来控制电机，电机则能够以不同的速度进行转动。

实验内容：1、用数字量控制直流电机的运动

参考例程：



2、用模拟量控制直流电机的运动

参考例程：



将上述程序烧录到主控板上，观察传送带的运动。请尝试改变模拟量的参数，观察传送带的运动情况，可以设置多组参数进行对比，观察其速度的改变情况。

## 第六节课 第三道工序—搬运机构

### 第 1 课时 搬运机构

搬运机构是 ECT-IAM 流水线的第三道工序，为后面的模拟加工做准备。搬运机构主要应用了四连杆机构（如图 6.1），所谓四连杆机构是由四个构件通过低副联接而成的平面连杆机构。它是平面连杆机构中最常见的形式，也是组成多连杆机构的基础。

#### 平面四连杆机构

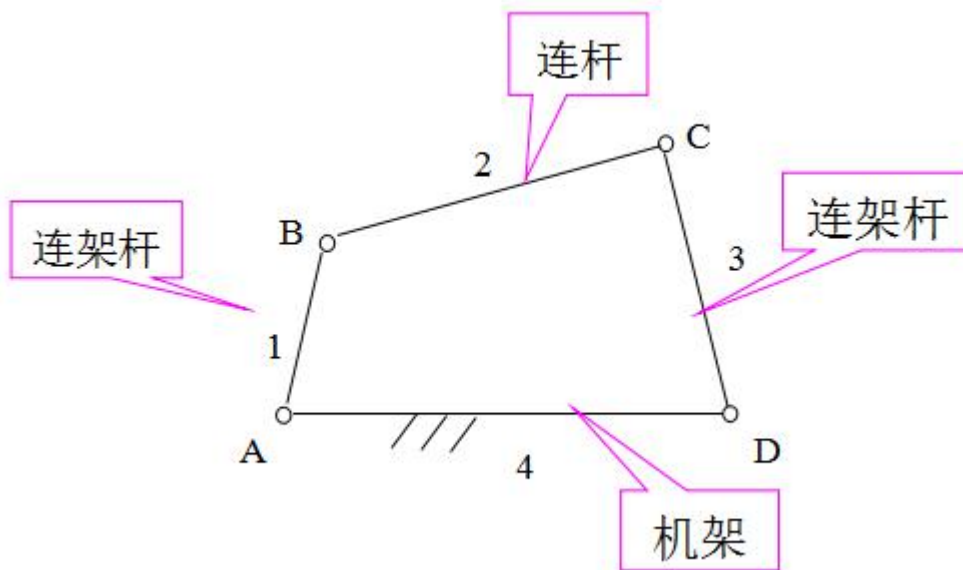
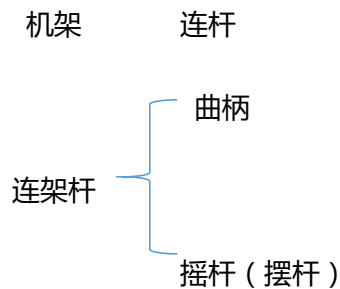


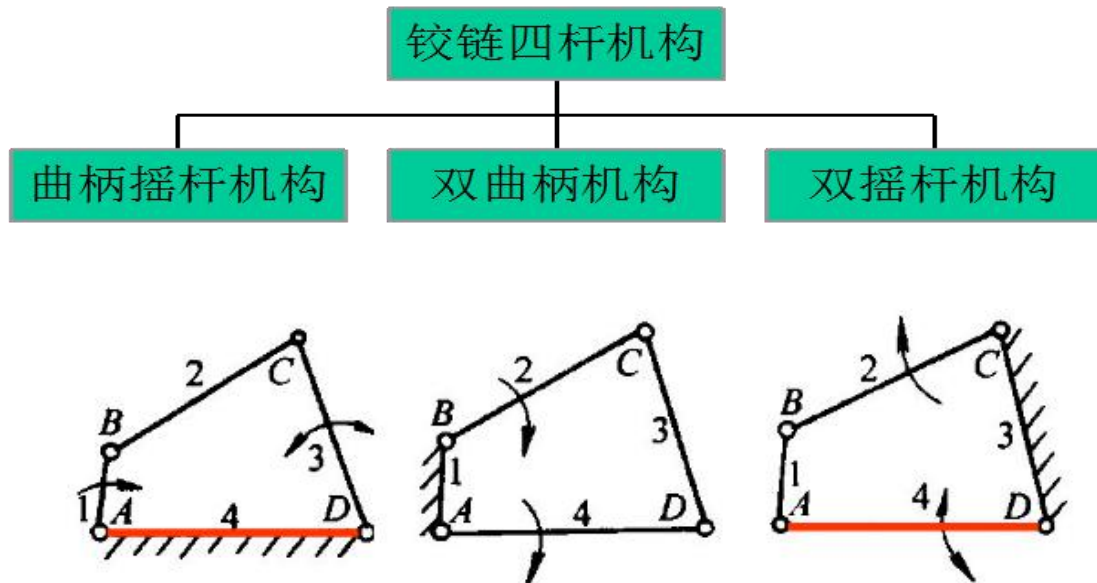
图 6.1 平面连杆机构

如图所示的曲柄摇杆机构中，主动连架杆(曲柄)AB 连续转动可以带动从动连架杆（摇杆）CD 作往复摆动。

平面四连杆的构成:



## 平面四杆机构的类型



曲柄摇杆机构：具有一个曲柄和一个摇杆的铰链四杆机构称为曲柄摇杆机构。通常，曲柄为主动件且等速转动，而摇杆为从动件作变速往返摆动，连杆作平面复合运动。曲柄摇杆机构中也有用摇杆作为主动构件，摇杆的往复摆动转换成曲柄的转动。

双曲柄机构：在铰链四杆机构中，若两连架杆均为曲柄，此四连杆机构称为双曲柄机构。

双摇杆机构：铰链四杆机构中两两连架杆均为摇杆，称为双摇杆机构。机构中两摇杆可以分别为主动件。当连杆与摇杆共线时，为机构的两个极限位置。

平面四杆结构常见的类型是有曲柄摇杆机构、双曲柄机构、双摇杆机构，而我们的第三道工序-搬运机构正是应用了双曲柄机构中平行四边形机构。

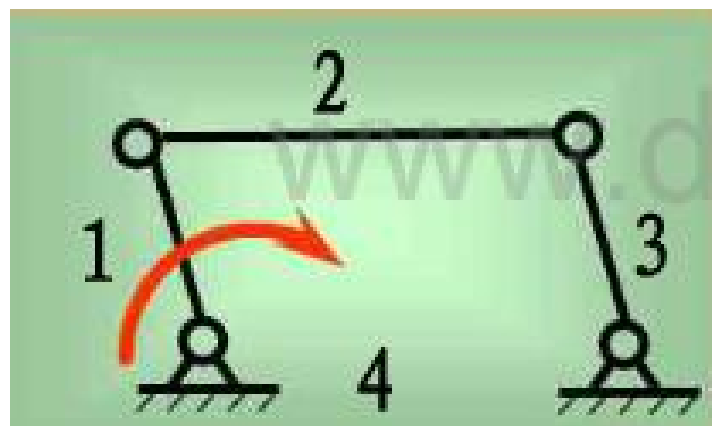


图 6.1 平行四连杆机构

机构特点：平行曲柄四连杆机构，1、3 两个平行曲柄杆总是以相同的角速度转动，但当这两根杆运动到水平时无法确定 1、3 杆的运动方向，所以无法保证 2 杆保持水平，故当 1、3 杆水平时是死点位置。如果要避免出现死点，可以多加一根与 1 杆平行的连杆。

## 连杆机构的优点

连杆机构的特点：面接触、承载能力高、耐磨损；易于制造和获得较高的精度。但效率低，会产生较大的运动误差。

连杆机构的用途：可以实现运动形式的转换、实现一定的动作、实现一定的轨迹。



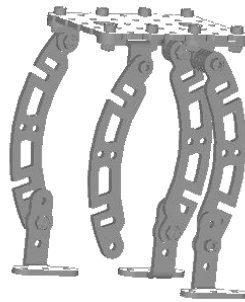
## 第 2 课时 组装搬运机构

搬运机构的工作特点及原理同学们已经有所了解,本节课的主要任务就是将其组装出来。参考提供的 stp 图,搭建一个搬运结构。

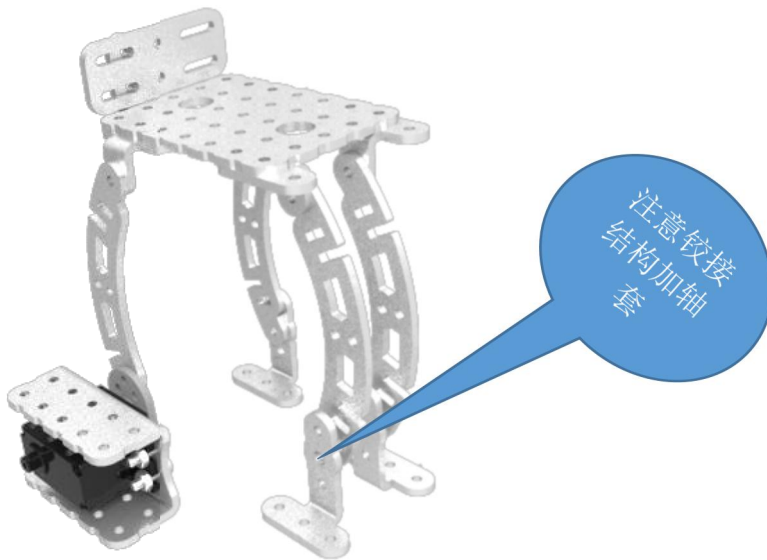
### 组装步骤参考



安装伺服电机



组装搬运机构

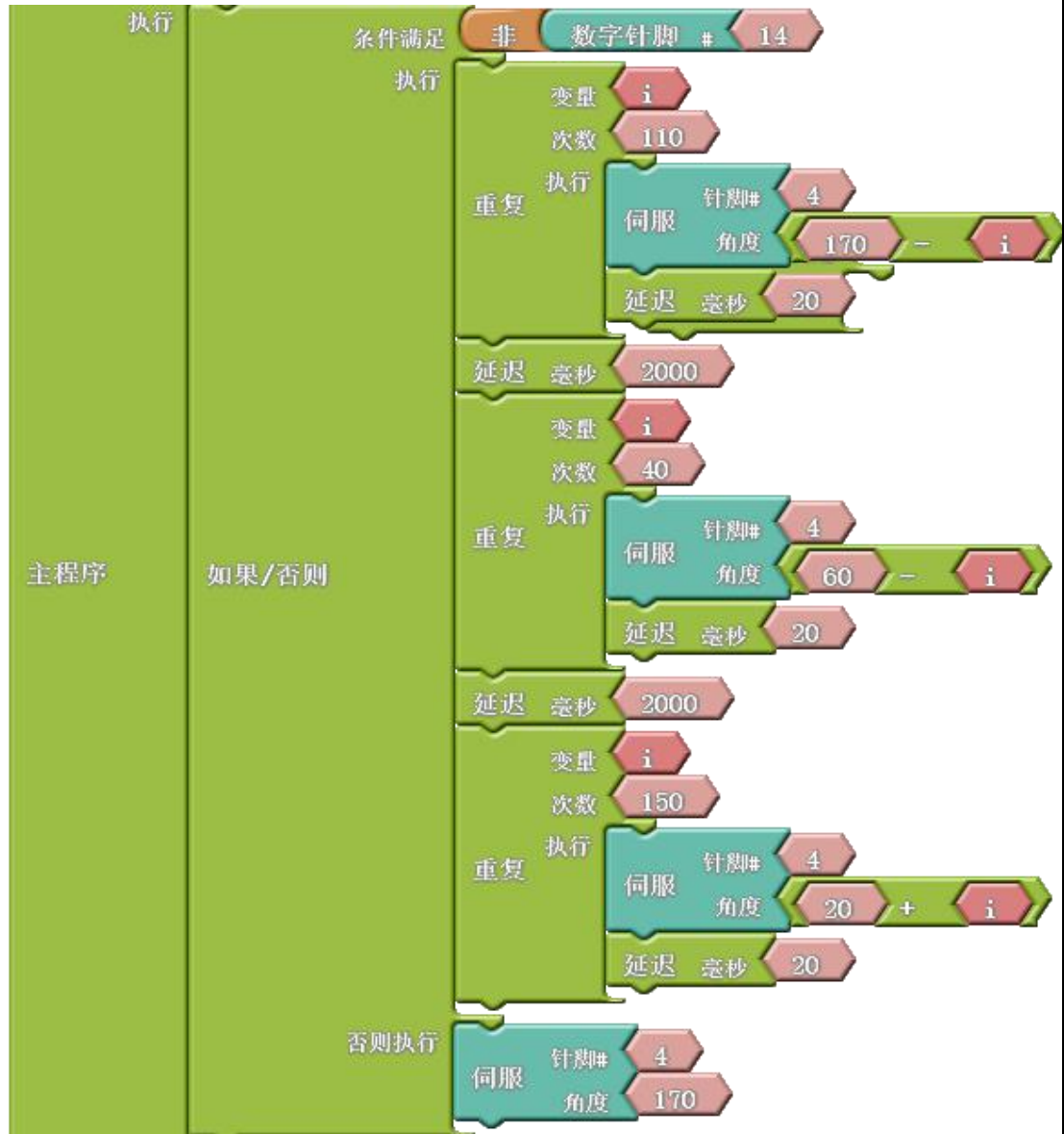


搬运结构完成

# 小实验

实验内容：控制搬运机构转动

参考例程：



对应的 C 语言程序为：

```
#include <Servo.h>
```

```
int _ABVAR_1_i = 0 ;
```

```
Servo servo_pin_4;
```

```
void setup()
```

```

{
  pinMode( 14, INPUT);
  servo_pin_4.attach(4);
}

void loop()
{
  if (!( digitalRead(14) ))
  {
    for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 110 ); _ABVAR_1_i++ )
    {
      servo_pin_4.write( ( 170 - _ABVAR_1_i ) );
      delay( 20 );
    }
    delay( 2000 );
    for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 40 ); _ABVAR_1_i++ )
    {
      servo_pin_4.write( ( 60 - _ABVAR_1_i ) );
      delay( 20 );
    }
    delay( 2000 );
    for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 150 ); _ABVAR_1_i++ )
    {
      servo_pin_4.write( ( 20 + _ABVAR_1_i ) );
      delay( 20 );
    }
  }
  else
  {
    servo_pin_4.write( 170 );
  }
}

```

该例程实现的效果：让舵机从 170° 起，角度参数每 20 毫秒减 1，直到 60°。保持 2s 后在让舵机从 60° 起，角度参数每 20 毫秒减 1，直到 20°，保持 2s 后舵机从 20° 起，角度参数每 20 毫秒加 1，直到 170°。

同学们可以根据舵机转动的效果，自行改变角度、次数等参数使搬运结构转动到你认为合适的位置。

# 第七课 第四道工序—模拟加工机械臂

## 第 1 课时 模拟加工机械臂

模拟加工这道工序在整个流水线中起到至关重要的作用，它很好的模拟了工业生产、工厂加工的过程，前面同学们对四连杆机构已经有初步的了解，本节课所学的模拟加工机械臂同样应用了四连杆机构，而它应用的则是其最基本的形式—曲柄摇杆机构。

### 曲柄摇杆

具有一个曲柄和一个摇杆的铰链四杆机构称为曲柄摇杆机构（图 7.1）。通常，曲柄 1 为主动件且等速转动，而摇杆 3 为从动件作变速往返摆动，连杆 2 作平面复合运动。曲柄摇杆机构中也有用摇杆作为主动构件，摇杆的往复摆动转换成曲柄的转动。

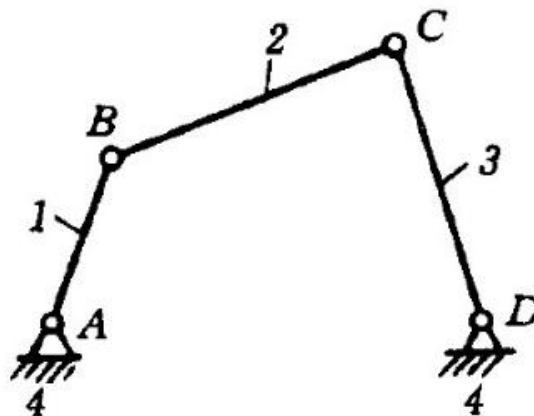


图 7.1 曲柄摇杆机

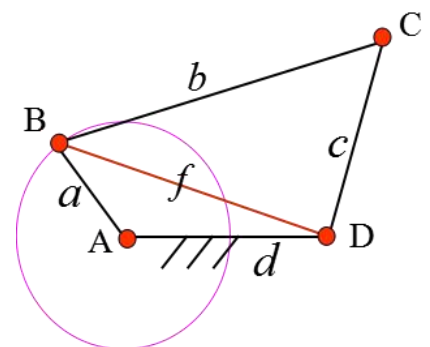
### 曲柄存在的条件（以曲柄摇杆机构为例）

设 AB 为曲柄，且  $a < d$

由  $\triangle BCD$ ：

$$b + c > f \quad , \quad b + f > c \quad , \quad c + f > b$$

以  $f_{\max} = a + d$  ，  $f_{\min} = d - a$



代入并整理得：

$$b + c > a + d$$

$$a < b$$

$$b+d > a+c \quad \text{并可得:} \quad a < c$$

$$c+d > a+b \quad \quad \quad a < d$$

曲柄存在的条件：

- (1)最短杆与最长杆长度之和小于或等于其余两杆长度和。
- (2)最短杆是连架杆或机架。

推论 1：

当  $L_{\max} + L_{\min} \leq L$  (其余两杆长度之和) 时

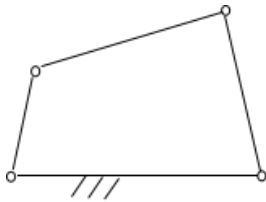
最短杆是连架杆之一——曲柄摇杆机构

最短杆是机架——双曲柄机构

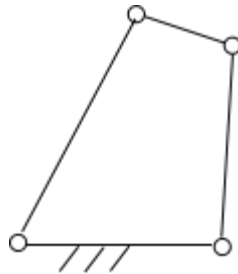
最短杆是连杆——双摇杆机构

推论 2：

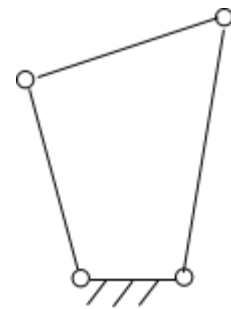
当  $L_{\max} + L_{\min} > L$  (其余两杆长度之和) 时 ——双摇杆机构



曲柄摇杆机构



双曲柄机构



双摇杆机构

模拟加工机械臂的结构图如下：

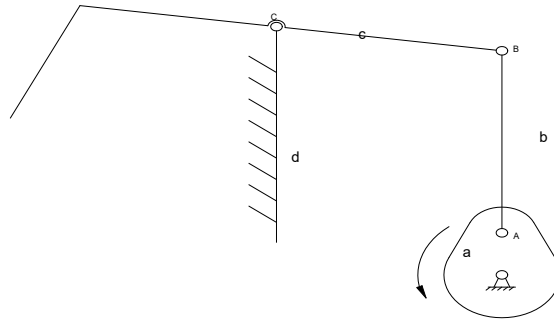


图 7.2 模拟加工机械臂结构图

机构特点：圆周转摆动——从图中不难看出这也是一个四杆机构，代替连杆的凸轮 a、连杆 b、支撑杆 d、摆动杆 c。凸轮 a 是驱动源，做圆周运动，连杆 b 将运动传递给摆动杆 c，由于 d 杆的固定了 c 杆的 C 点位置，且 c 杆与 b 杆连接，所以 c 杆的末端 B 点只能做摆动，最终 C 杆实现摆动。

工业及现实生活中有大量的设备需要用到曲柄摇杆机构，如牛头刨床进给机构、雷达调整机构、缝纫机脚踏机构、复摆式腭式破碎机、钢材输送机等

## 第 2 课时 组装模拟加工机械臂

模拟加工机械臂的工作特点及原理同学们已经有所了解,本节课的主要任务就是将其组装出来。参考提供的 stp 图,搭建一个模拟加工机械臂。

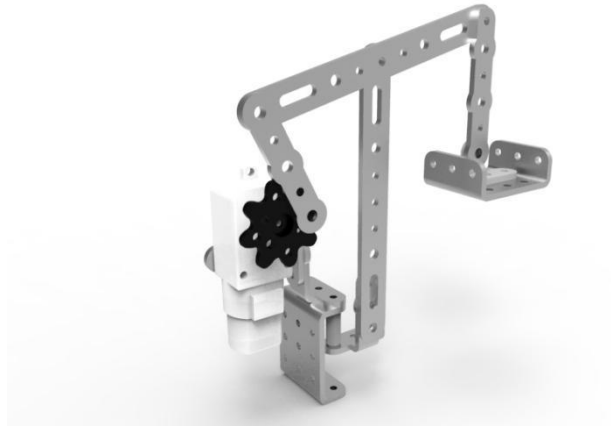
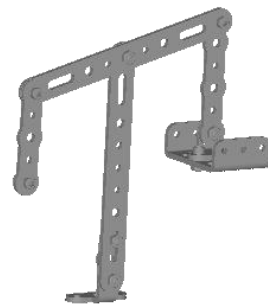


图 7.3 模拟加工机械臂

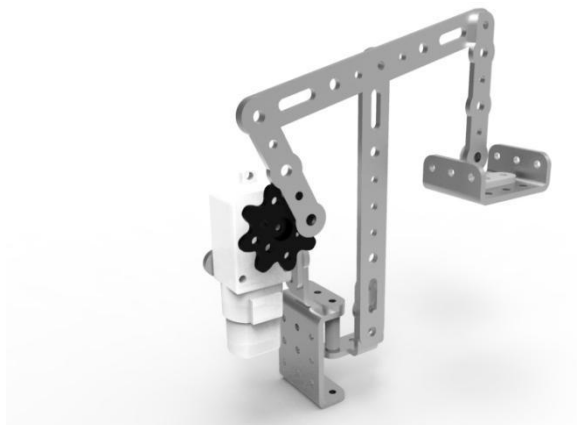
### 组装步骤参考



安装直流电机



组装连杆机构



模拟加工机械臂完成

## 小实验

实验内容：控制加工机械臂

参考例程



对应 C 语言程序；

```
void setup()
```

```
{
```

```
pinMode( 9 , OUTPUT);
```

```
pinMode( 10 , OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
digitalWrite( 9 , HIGH );
```

```
digitalWrite( 10 , LOW );
```

```
}
```



## 第八节课 第五道工序—码垛

### 第 1 课时 机械手爪

ECT-IAM 流水线的最后一道工序—码垛，它主要是由一个 3 自由度机械臂组成，本节课我们首先来研究机械手模块（图 8.1），机械手模块由 1 个标准伺服电机驱动，通过连杆结构和齿轮组传动来达到夹取效果，机械手的设计方案有很多种，这只是其中一种。



图 8.1 机械手模块

运动特性：开合角度比较大，夹具顶端的运动轨迹简单稳定，夹具顶端非平行开合，比较适合于“握”住曲面物体或者柔软的物体。夹具顶点可以再安装一对带铰接的小平板零件，从而适合夹取立方体形状的物体。

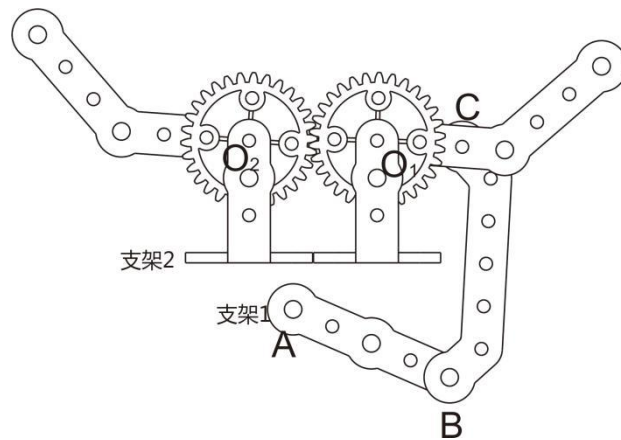


图 8.2 机构原理图

机构原理：该机械手包含一个 4 连杆结构和齿轮传动。其中杆 AB、BC、CO1 以及支架组成一个曲柄摇杆机构（如图 8.2 机构原理图），杆 AB 作为驱动杆，通过连杆 BC 将运动传递给从动杆 CO1，在该机械手中表现的运动形式为右边机械手指（杆 CO1）实现摆动。杆 CO1 连接一个齿轮，与另一边的机械手指所连接的齿轮啮合，形成一个齿轮传动。这两个齿轮参数一致，所以可实现等速传动。在这个齿轮传动中，右边的齿轮作为主动轮，左边的齿轮作为从动轮。因为两个齿轮做相对运动，所以最终机械手爪的表现形式为左右两边的手指做相对等速运动。

## 齿轮传动

在第二道工序中我们已经了解一种常见的传动方式—带传动，这节课我们将了解另一种常见的传动方式—齿轮传动。齿轮传动是机械传动中应用最广的一种传动形式。它的传动比较准确，效率高，结构紧凑，工作可靠，寿命长。

### 等速传递

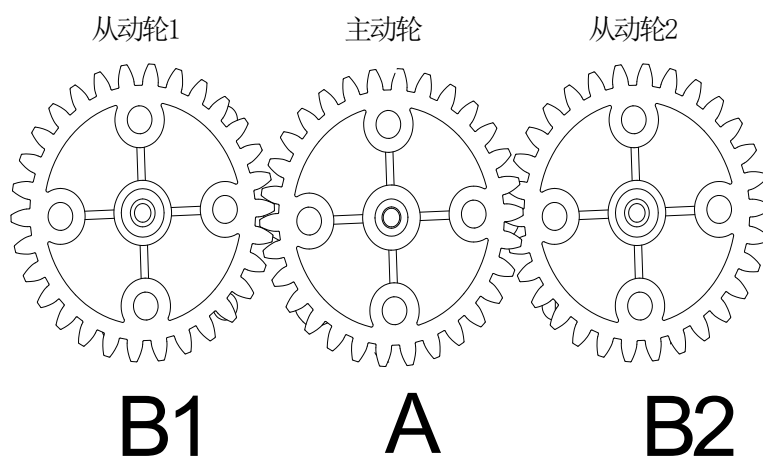


图 8.3 等速传递示意图

齿轮机构：等速传递—这是一个齿轮传动机构，三个齿轮的所有参数一致，所以主动轮 A 将运动传递给从动轮 B1 和 B2 时，除了 A 轮与齿轮 B1、B2 运动方向相反，运动的速度大小一致，实现等速传递。

## 差速传递

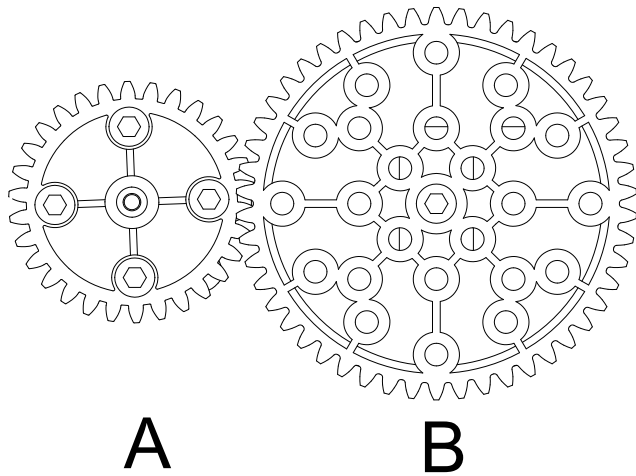


图 8.4 差速传递示意图

齿轮组：差速传动—这是由两个模数和压力角一样，分度圆直径不同的齿轮组成；A 轮作为驱动轮，B 轮是从动轮，B 轮的分度圆直径较大，所以两轮啮合开始运动之后，B 轮的角速度小于 A 轮，运动表现为从动轮 B 转速小于驱动轮 A，形成差速运动特征。

## 第 2 课时 组装机械手

机械手的工作特点及原理同学们已经有所了解，本节课的主要任务就是将其组装出来。

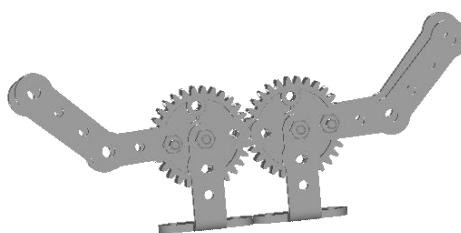
参考提供的 stp 图，搭建一个机械手。

### 组装步骤参考

我们可以将机械手分为伺服电机、手爪两部分进行组装。



安装伺服电机



组装机械手爪



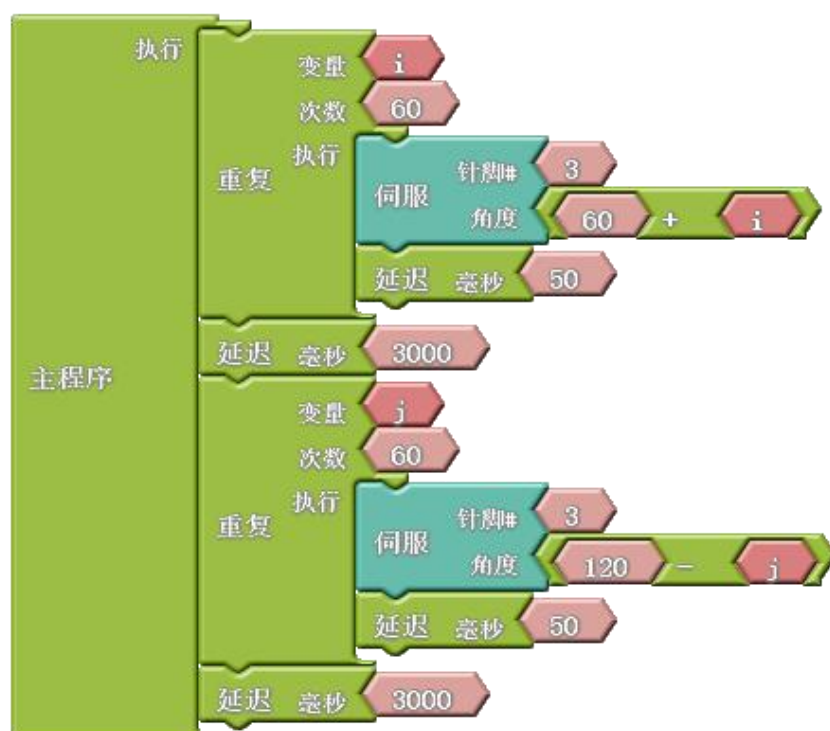
最后用双足连杆、机械手 40mm 将两部分连接完成机械手的安装，提示：铰接结构注意轴套的添加。



## 小实验

实验内容：控制机械手

参考例程：



观察机械手的运动效果，同学们可以根据自己的预期效果自行改变角度、延迟等参数。

## 第九课 3 自由度机械臂

### 第 1 课时 3 自由度机械臂

机械手爪的运动特性和机构原理同学们已经熟悉，本节课主要介绍 3 自由度机械臂相关知识。

#### 关节模块

这种关节模块在机器人的结构设计中非常常用，思路也非常简单，就是将多个关节模块串联累加，构建多自由度的机器人，每一个关节为一个自由度，从而实现机械臂、人形、多足仿生等机器人结构，如图 9.1。



图 9.1 关节模块

运动特性：能较好地模仿生物的运动形态，只要自由度足够多，几乎什么动作都能做到，常见于人形机器人舞蹈表演，或者学习机器人运动学规划等。但是结构缺少优化，缺少传动，能耗较大，做简单运动时冗余的自由度较多。在小尺度上用途非常广泛，常见于玩具、模型、教具等，但是无法承担一定级别以上的尺寸和重量，所以工程中和工业中用途较少。

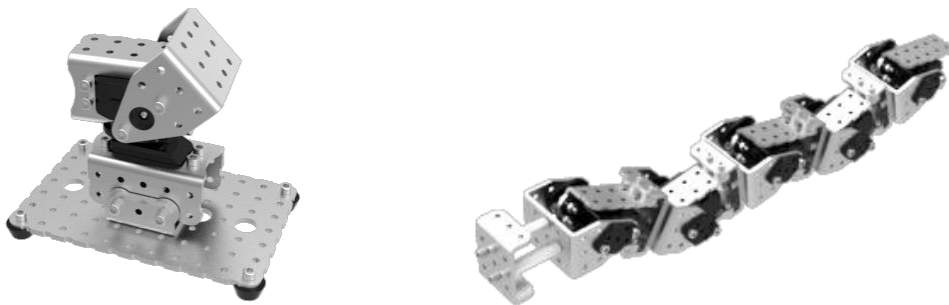


图 3.10 关节模块的应用示例

#### 3 自由度机械臂

机械手的每一个自由度是由其操作机的独立驱动关节来实现的。所以在应用中,关节和自由度在表达机械手的运动灵活性方面是意义相通的。又由于关节在实际构造上是由回转或移动的轴来完成的,所以又习惯称之为轴。因此,就有了6自由度、6关节或6轴机械手的命名方法。它们都说明这一机械手的操作有6个独立驱动关节结构,能在其工作空间中实现抓取物件的任意位置和姿态。

那么3自由度机械臂(如图9.2.3自由度机械臂)则是由3个关节模块串联累加,构建成的一个3自由度机械手爪。

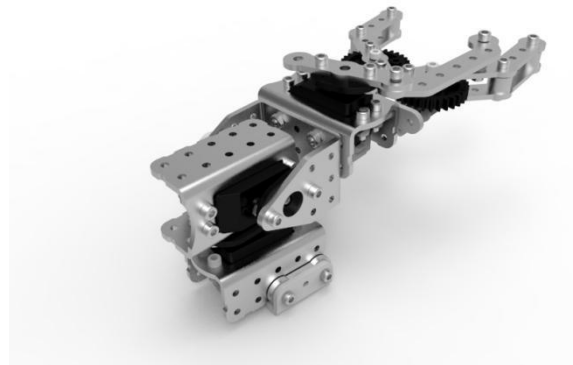


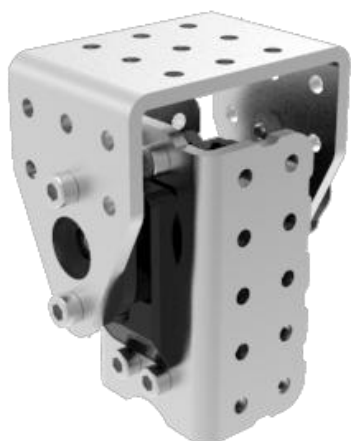
图 9.2.3 自由度机械臂

## 第 2 课时 组装 3 自由度机械臂

3 自由度的机械臂的工作特点及原理同学们已经有所了解，本节课的主要任务就是将其组装出来。参考提供的 stp 图，利用已经搭建好的机械手组装一个 3 自由度机械臂的。

### 组装步骤参考

我们可以将 3 自由度机械臂拆分为关节模块、机械手爪、伺服电机驱动三部分进行组装。



组装关节模块



安装伺服电机



组装机手爪

最后将关节模块、伺服电机、机械手爪三个模块整合，则组成了一个完整的 3 自由度机械臂。



## 第十课 调试 3 自由度机械臂 ( 2 课时 )

调试 3 自由度的机械臂则是对 3 个伺服电机进行控制，从而使 3 个伺服电机相互配合、转动不同的角度，达到最终的工作目的。同学们已经对机械手爪的控制非常熟悉，这节课请同学利用之前所学的知识，尝试自己进行编程，实现 3 自由度机械臂转动。

1.连接：将主控板连上 pc，将 3 自由度机械臂连在扩展板的对应端口。

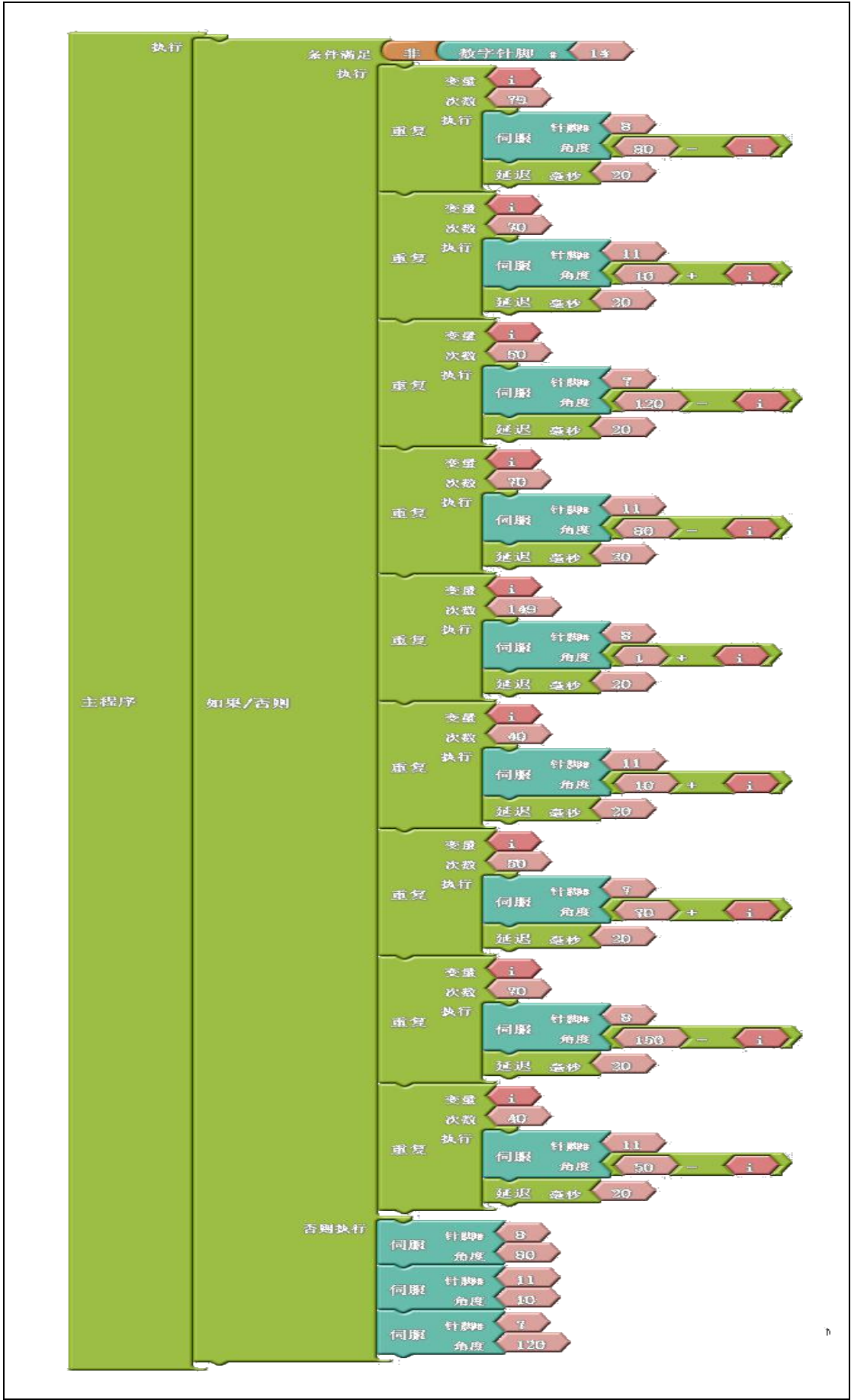
同样对伺服电机的控制我们仍利用重复语句：

“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 );  
}
```

2.参考例程：( 本例程— 7、8、11 号端口为例 )



对应的 C 语言程序：

```
#include <Servo.h>
```

```
int _ABVAR_1_i = 0;
```

```
Servo servo_pin_8;
```

```
Servo servo_pin_11;
```

```
Servo servo_pin_7;
```

```
void setup()
```

```
{
```

```
  pinMode( 14, INPUT);
```

```
  servo_pin_8.attach(8);
```

```
  servo_pin_11.attach(11);
```

```
  servo_pin_7.attach(7);
```

```
}
```

```
void loop()
```

```
{
```

```
  if (!( digitalRead(14) ))
```

```
  {
```

```
    for (_ABVAR_1_i= 1; _ABVAR_1_i<= ( 79 ); _ABVAR_1_i++)
```

```
    {
```

```
servo_pin_8.write( ( 80 - _ABVAR_1_i ) );  
  
delay( 20 );  
  
}  
  
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 70 ); _ABVAR_1_i++ )  
  
{  
  
servo_pin_11.write( ( 10 + _ABVAR_1_i ) );  
  
delay( 20 );  
  
}  
  
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 50 ); _ABVAR_1_i++ )  
  
{  
  
servo_pin_7.write( ( 120 - _ABVAR_1_i ) );  
  
delay( 20 );  
  
}  
  
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 70 ); _ABVAR_1_i++ )  
  
{  
  
servo_pin_11.write( ( 80 - _ABVAR_1_i ) );  
  
delay( 20 );  
  
}  
  
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 149 ); _ABVAR_1_i++ )  
  
{  
  
servo_pin_8.write( ( 1 + _ABVAR_1_i ) );  
  
delay( 20 );
```

```
}  
  
for (_ABVAR_1_i= 1; _ABVAR_1_i<= ( 40 ); _ABVAR_1_i++)  
  
{  
  
    servo_pin_11.write( ( 10 + _ABVAR_1_i ) );  
  
    delay( 20 );  
  
}  
  
for (_ABVAR_1_i= 1; _ABVAR_1_i<= ( 50 ); _ABVAR_1_i++)  
  
{  
  
    servo_pin_7.write( ( 70 + _ABVAR_1_i ) );  
  
    delay( 20 );  
  
}  
  
for (_ABVAR_1_i= 1; _ABVAR_1_i<= ( 70 ); _ABVAR_1_i++)  
  
{  
  
    servo_pin_8.write( ( 150 - _ABVAR_1_i ) );  
  
    delay( 20 );  
  
}  
  
for (_ABVAR_1_i= 1; _ABVAR_1_i<= ( 40 ); _ABVAR_1_i++)  
  
{  
  
    servo_pin_11.write( ( 50 - _ABVAR_1_i ) );  
  
    delay( 20 );  
  
}  
  
}
```

```
else
{
    servo_pin_8.write( 80 );
    servo_pin_11.write( 10 );
    servo_pin_7.write( 120 );
}
}
```

观察 3 自由度机械臂的转动效果，同学们可以根据自己想要实现的效果，进行改变角度、延迟等参数。

## 第十一课 ECT-IAM 智能流水线（2 课时）

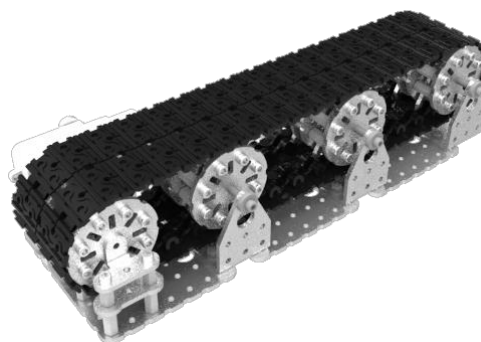
ECT-IAM 智能流水线设计平台的 5 道工序同学们已经有所了解，每道工序都发挥着各自的作用、相辅相成、独立运转，而控制系统可以保证生产下协调运作，从而营造一个流水线模型。本节课的主要内容是将每个模块的结合整体方案设计柔性生产线。产品提供了足够的机器零件和电子部件等，也提供了详细的组装方案。

### 组装步骤参考

参考提供的 stp 图，利用已经搭建好的各个工序组装一个 ECT-IAM 智能流水线。



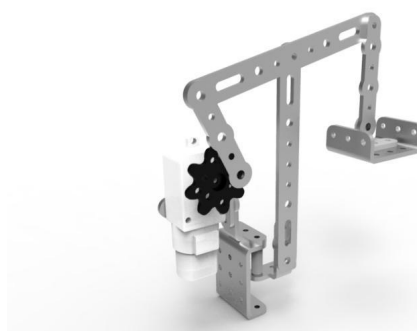
铸锭送料结构



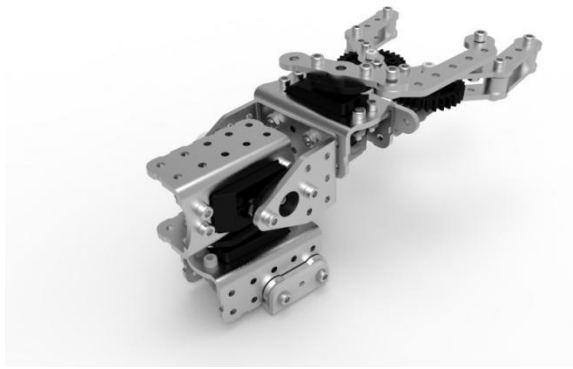
传送机构



搬运机构

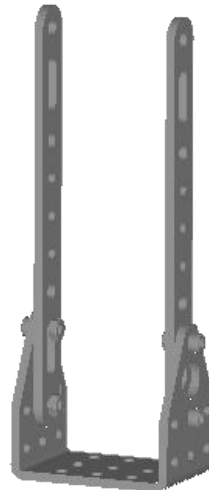
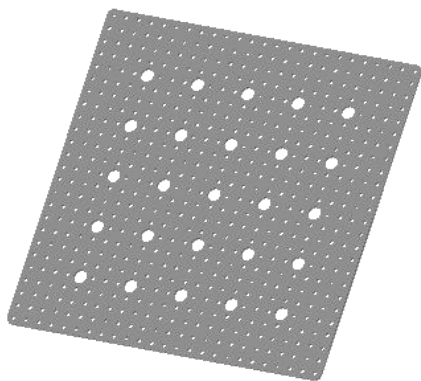


模拟加工

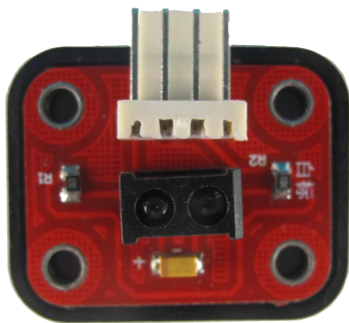


### 3 自由度机械臂机构

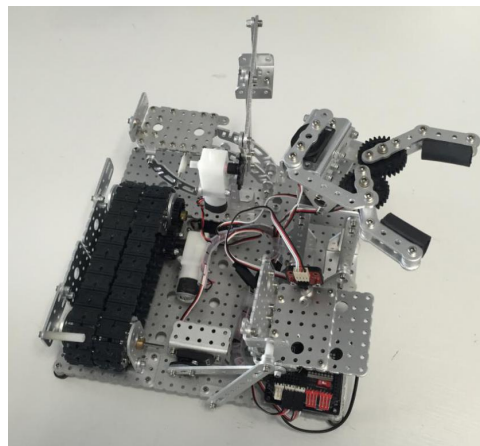
将以上 5 道工序安装在下图大底盘上，参照 stp 图，在添加一些机架以及外观件完成整个流水线的搭建。



将白标传感器安装在流水线的合适位置，整个流水线的搭建完成。



白标传感器



流水线搭建完成

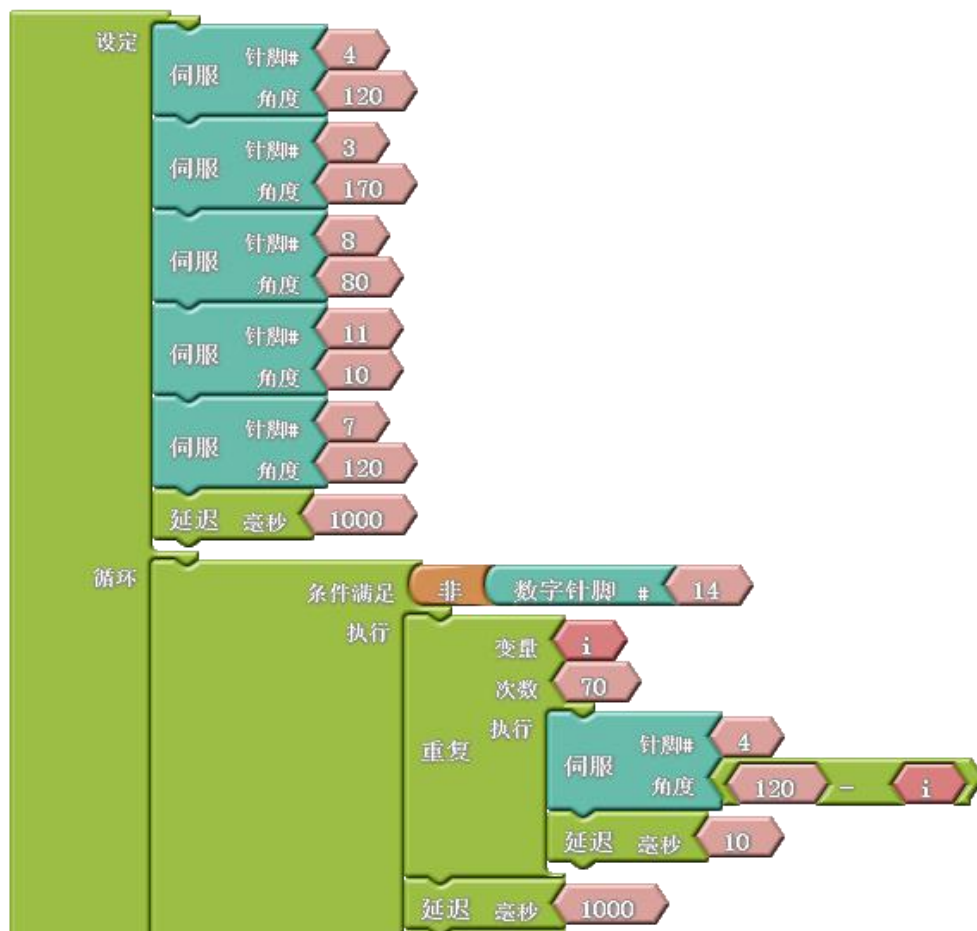


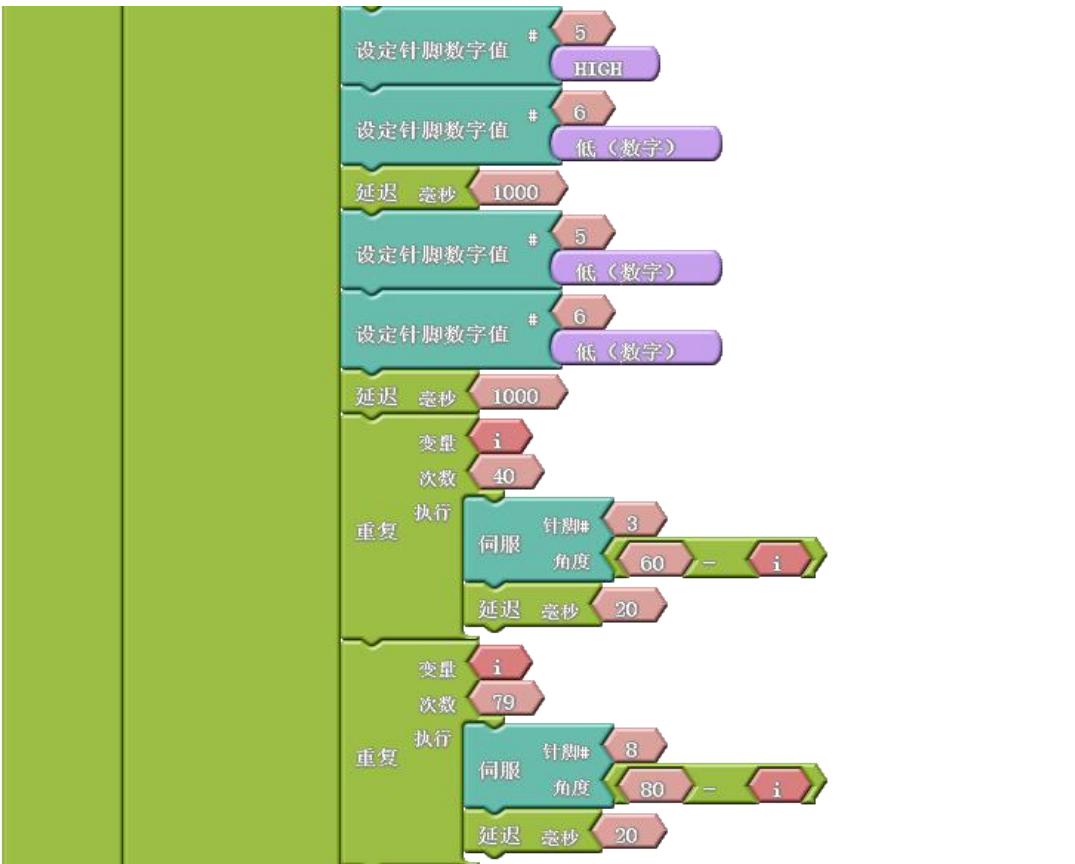
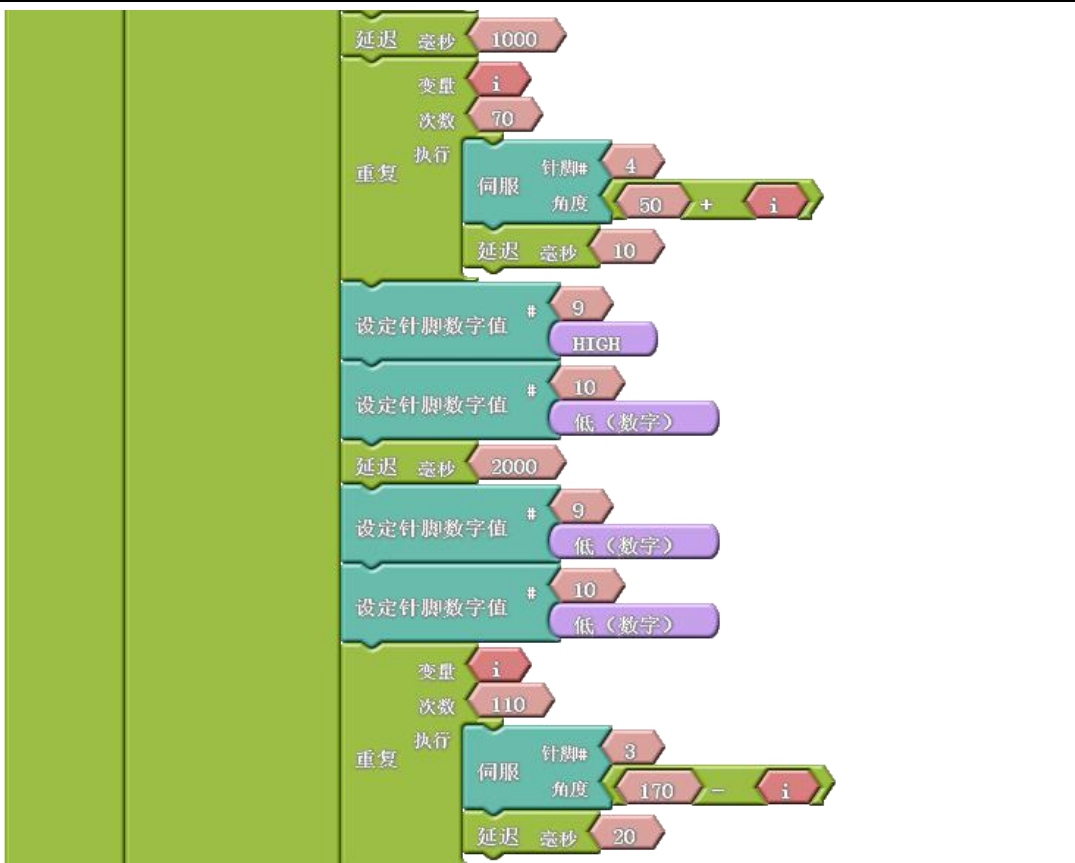
## 第十二课 调试流水线 ( 2 课时 )

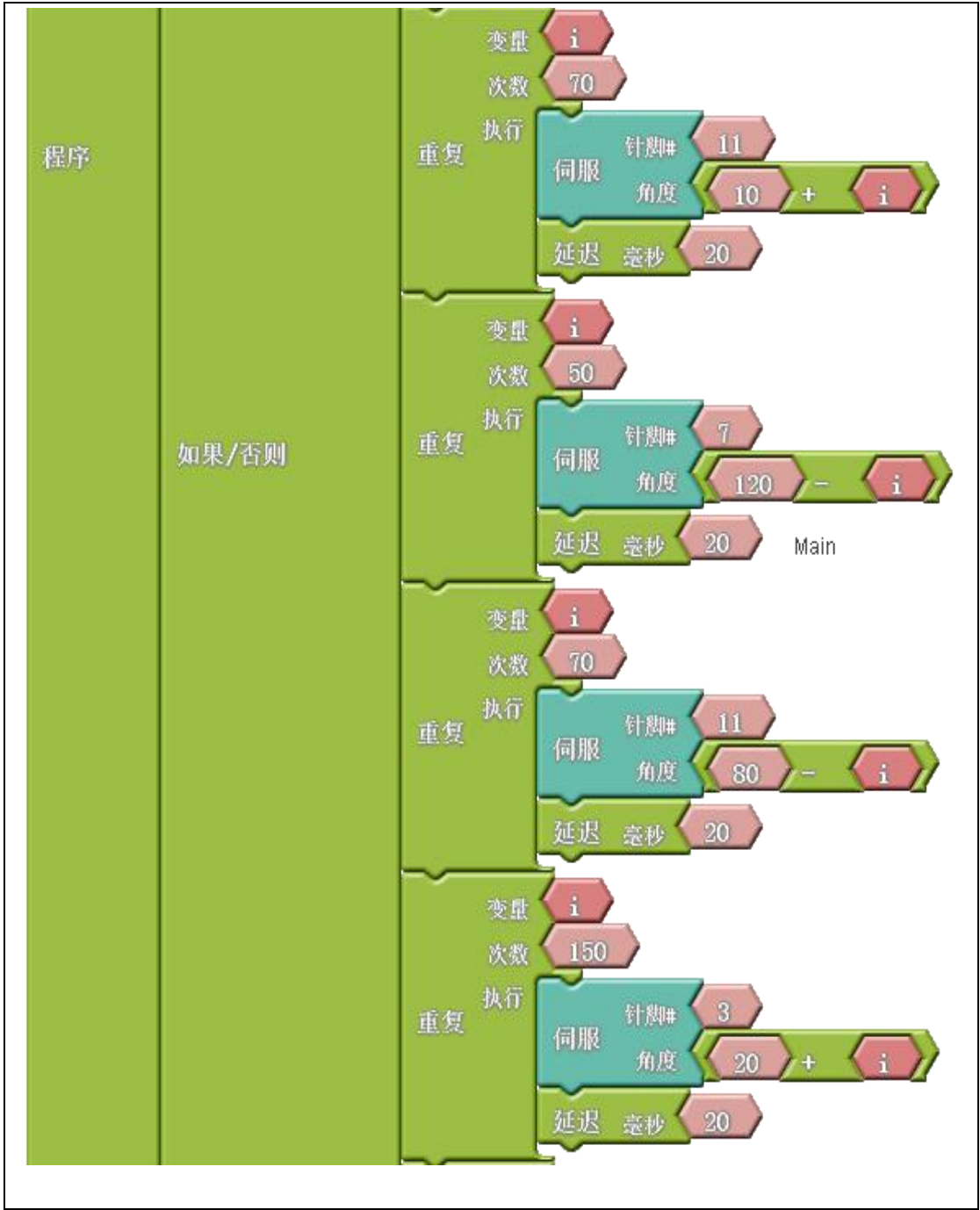
整个流水线已经搭建完毕，本节课的主要内容是对整个流水线进行调试，使各个工序协调运作。

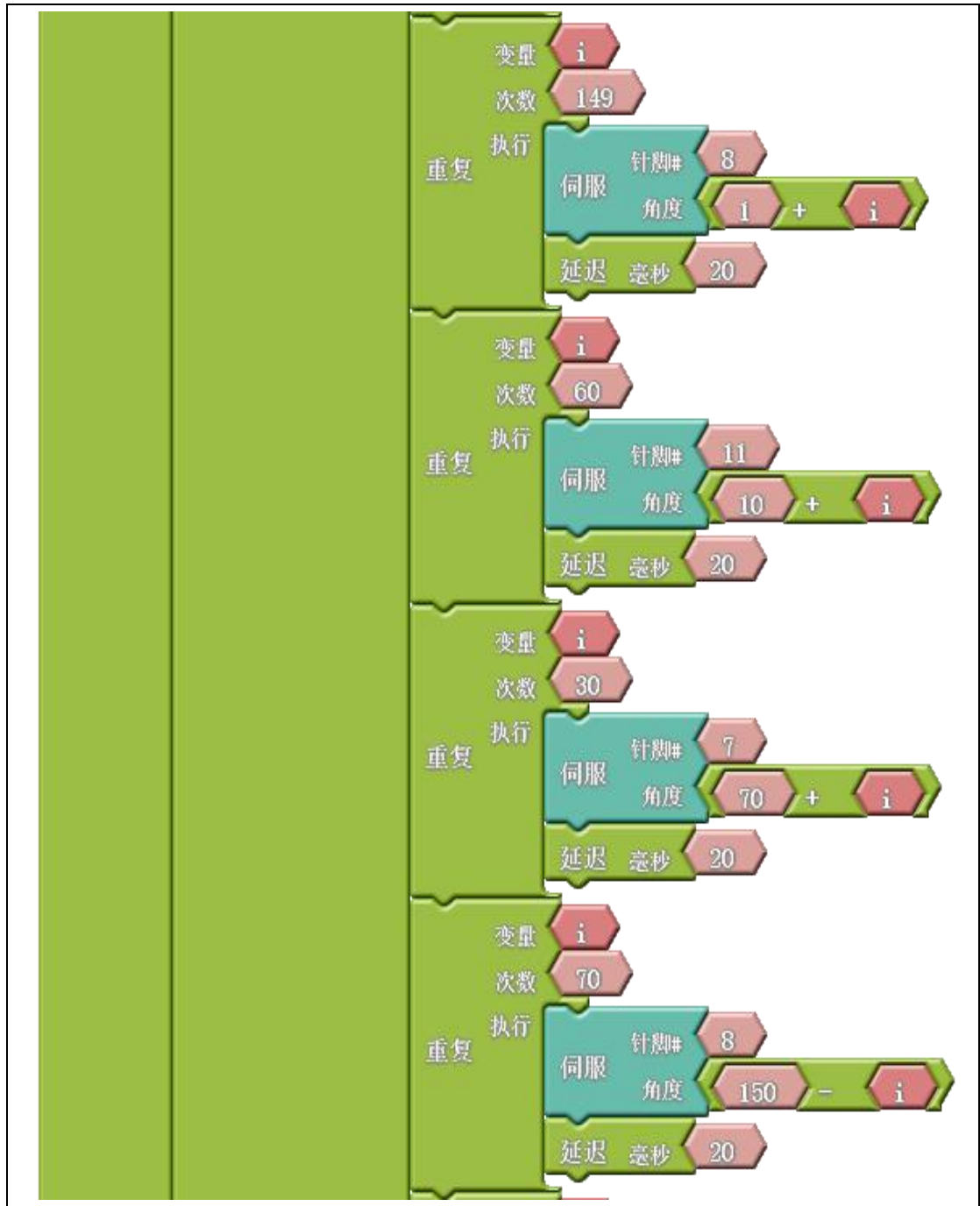
1.连接：将主控板、扩展板、锂电池固定在流水线上；将主控板连上 pc 端；将直流电机、伺服电机、传感器连接到扩展板；

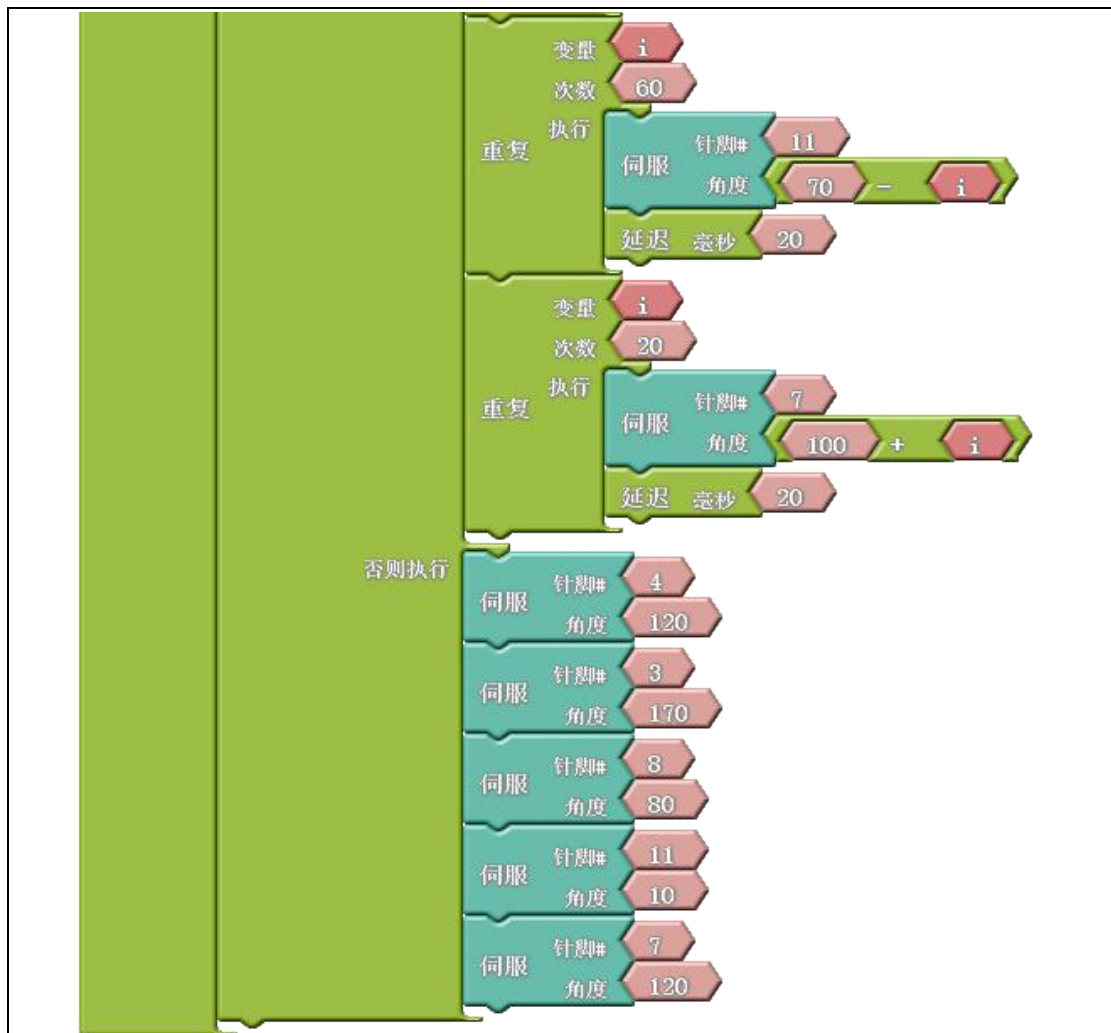
2.参考例程：











对应的 C 语言程序:

```
#include <Servo.h>
```

```
Servo servo_pin_4;
```

```
Servo servo_pin_3;
```

```
Servo servo_pin_8;
```

```
Servo servo_pin_11;
```

```
Servo servo_pin_7;
```

```
int _ABVAR_1_i = 0;
```

```
void setup()
```

```
{
```

```
  pinMode( 14, INPUT);
```

```
  servo_pin_4.attach(4);
```

```
  servo_pin_3.attach(3);
```

```
  servo_pin_8.attach(8);
```

```
  servo_pin_11.attach(11);
```

```
  servo_pin_7.attach(7);
```

```

pinMode( 9 , OUTPUT);
pinMode( 10 , OUTPUT);
pinMode( 5 , OUTPUT);
pinMode( 6 , OUTPUT);
servo_pin_4.write( 120 );

servo_pin_3.write( 170 );

servo_pin_8.write( 80 );

servo_pin_11.write( 10 );

servo_pin_7.write( 120 );

delay( 1000 );
}

void loop()
{
  if (!( digitalRead(14) ))
  {
    for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 70); _ABVAR_1_i++)
    {
      servo_pin_4.write( ( 120 - _ABVAR_1_i ) );
      delay( 10 );
    }
    delay( 1000 );
    for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 70); _ABVAR_1_i++)
    {
      servo_pin_4.write( ( 50 + _ABVAR_1_i ) );
      delay( 10 );
    }
    digitalWrite( 9 , HIGH );
    digitalWrite( 10 , LOW );
    delay( 2000 );
    digitalWrite( 9 , LOW );
    digitalWrite( 10 , LOW );
    for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 110); _ABVAR_1_i++)
    {
      servo_pin_3.write( ( 170 - _ABVAR_1_i ) );
      delay( 20 );
    }
    digitalWrite( 5 , HIGH );

```

```
digitalWrite( 6 , LOW );
delay( 1000 );
digitalWrite( 5 , LOW );
digitalWrite( 6 , LOW );
delay( 1000 );
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 40 ); _ABVAR_1_i++ )
{
    servo_pin_3.write( ( 60 - _ABVAR_1_i ) );
    delay( 20 );
}
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 79 ); _ABVAR_1_i++ )
{
    servo_pin_8.write( ( 80 - _ABVAR_1_i ) );
    delay( 20 );
}
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 70 ); _ABVAR_1_i++ )
{
    servo_pin_11.write( ( 10 + _ABVAR_1_i ) );
    delay( 20 );
}
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 50 ); _ABVAR_1_i++ )
{
    servo_pin_7.write( ( 120 - _ABVAR_1_i ) );
    delay( 20 );
}
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 70 ); _ABVAR_1_i++ )
{
    servo_pin_11.write( ( 80 - _ABVAR_1_i ) );
    delay( 20 );
}
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 150 ); _ABVAR_1_i++ )
{
    servo_pin_3.write( ( 20 + _ABVAR_1_i ) );
    delay( 20 );
}
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 149 ); _ABVAR_1_i++ )
{
    servo_pin_8.write( ( 1 + _ABVAR_1_i ) );
    delay( 20 );
}
for ( _ABVAR_1_i= 1; _ABVAR_1_i<= ( 60 ); _ABVAR_1_i++ )
{
    servo_pin_11.write( ( 10 + _ABVAR_1_i ) );
    delay( 20 );
}
```

```

}
for (_ABVAR_1_i= 1; _ABVAR_1_i<= ( 30); _ABVAR_1_i++)
{
    servo_pin_7.write( ( 70 + _ABVAR_1_i ));
    delay( 20 );
}
for (_ABVAR_1_i= 1; _ABVAR_1_i<= ( 70); _ABVAR_1_i++)
{
    servo_pin_8.write( ( 150 - _ABVAR_1_i ));
    delay( 20 );
}
for (_ABVAR_1_i= 1; _ABVAR_1_i<= ( 60); _ABVAR_1_i++)
{
    servo_pin_11.write( ( 70 - _ABVAR_1_i ));
    delay( 20 );
}
for (_ABVAR_1_i= 1; _ABVAR_1_i<= ( 20); _ABVAR_1_i++)
{
    servo_pin_7.write( ( 100 + _ABVAR_1_i ));
    delay( 20 );
}
}
else
{
    servo_pin_4.write( 120 );
    servo_pin_3.write( 170 );
    servo_pin_8.write( 80 );
    servo_pin_11.write( 10 );
    servo_pin_7.write( 120 );
}
}
}

```

观察效果，如果没有达到预期的效果，请同学们尝试改变例程中的参数，进行调试。





**机器时代（北京）科技有限公司**

**ROBOTTIME BEIJING TECHNOLOGY. LTD**

电话：010-60705356

Email：web@robottime.cn

<http://www.robottime.cn>