

北京邮电大学	创新创业实践课程 实验指导书	
计算机学院	BUPT SCST 智能机器人实验	



戴志涛等编
二〇一八年九月

目录

目录	i
导论	1
一、 智能机器人发展背景	1
二、 Arduino 开源硬件平台	2
三、 课前自学要求	3
实验一：LED 控制	4
一、 实验目的	4
二、 实验内容	4
三、 背景知识	4
3.1 LED 元件及其工作原理	4
3.2 ArduBlock	5
四、 实验步骤	6
五、 实验现象及结果	8
实验二：异步串口通信	9
一、 实验目的	9
二、 实验内容	9
三、 背景知识	9
3.1 串口及其工作原理	9
3.2 串口收发流程与判断逻辑	10
3.3 多重判断逻辑	11
四、 实验步骤	12
五、 实验现象及结果	13
实验三：扬声器发声与音乐播放	15
一、 实验目的	15
二、 实验内容	15
三、 背景知识	15
3.1 GPIO 及扬声器发声原理	15
3.2 音乐播放原理	16
四、 实验步骤	18
实验四：舵机控制	20
一、 实验目的	20
二、 实验内容	20
三、 背景知识	20
四、 实验步骤	21

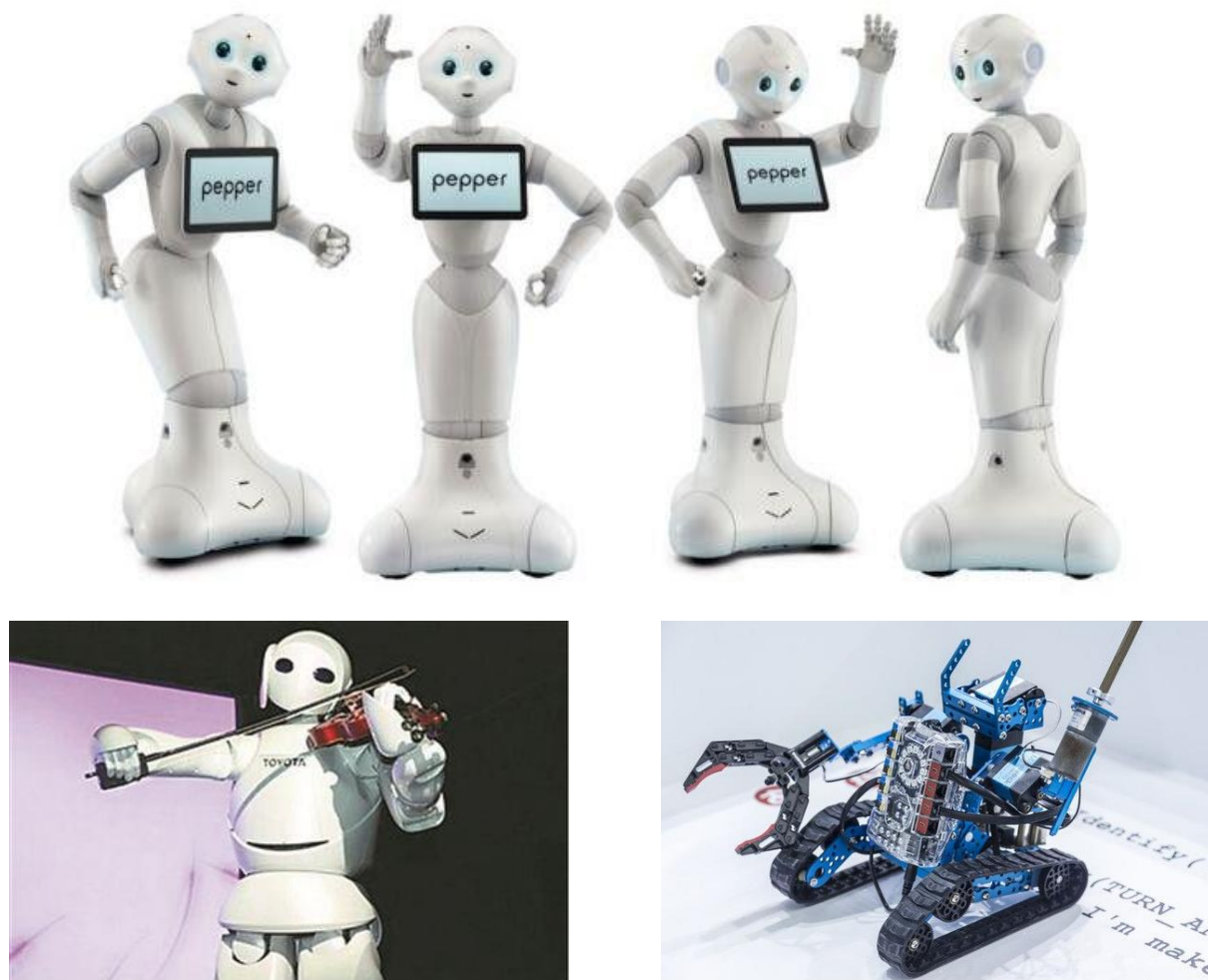
扩展实验一：机械臂抓物品游戏.....	23
一、 实验目的	23
二、 实验内容	23
三、 背景知识	23
四、 实验步骤	23
扩展实验二：机械臂之舞.....	24
五、 实验目的	24
六、 实验内容	24
七、 背景知识	24
八、 实验步骤	24
扩展实验三：自主实验（选做）	25
一、 实验目的	25
二、 实验内容	25
课后作业与实验报告	26
一、 课后作业	26
二、 分组讨论参考题目	26
三、 实验报告要求	26
附录《ArduBlock 教育版》使用帮助	27

导论

一、 智能机器人发展背景

我们从广泛意义上理解所谓的智能机器人，它给人的最深刻的印象是一个独特的进行自我控制的“活物”。其实，这个自控“活物”的主要器官并没有像真正的人那样微妙而复杂。

智能机器人具备形形色色的内部信息传感器和外部信息传感器，如视觉、听觉、触觉、嗅觉。除具有感受器外，它还有效应器，作为作用于周围环境的手段。这就是筋肉，或称自整步电动机，它们使手、脚、长鼻子、触角等动起来。由此也可知，智能机器人至少要具备三个要素：感觉要素、反应要素和思考要素。



我们称这种机器人为自控机器人，以便使它同前面谈到的机器人区分开来。它是控制论产生的结果，控制论主张这样的事实：生命和非生命有目的的行为在很多方面是一致的。正像一个智能机器人制造者所说的，机器人是一种系统的功能描述，这种系

统过去只能从生命细胞生长的结果中得到，现在它们已经成了我们自己能够制造的东西了。

智能机器人能够理解人类语言，用人类语言同操作者对话，在它自身的“意识”中单独形成了一种使它得以“生存”的外界环境——实际情况的详尽模式。它能分析出现的情况，能调整自己的动作以达到操作者所提出的全部要求，能拟定所希望的动作，并在信息不充分的情况下和环境迅速变化的条件下完成这些动作。当然，要它和我们人类思维一模一样，这是不可能办到的。不过，仍然有人试图建立计算机能够理解的某种“微观世界”。

二、 Arduino 开源硬件平台

Arduino 是一款便捷灵活、方便上手的开源电子原型平台。包含硬件（各种型号的 Arduino 板）和软件（Arduino IDE）。由一个欧洲开发团队于 2005 年冬季开发。

它构建于开放原始码 simple I/O 介面版，并且具有使用类似 Java、C 语言的 Processing/Wiring 开发环境。主要包含两个主要的部分：硬件部分是可以用来做电路连接的 Arduino 电路板；另外一个则是 Arduino IDE，你的计算机中的程序开发环境。你只要在 IDE 中编写程序代码，将程序上传到 Arduino 电路板后，程序便会告诉 Arduino 电路板要做些什么了。



Arduino 实验平台具有以下特点：

- 跨平台：Arduino IDE 可以在 Windows、Macintosh OS X、Linux 三大主流操作系统上运行，而其他的大多数控制器只能在 Windows 上开发。
- 简单清晰：Arduino IDE 基于 processing IDE 开发。对于初学者来说，极易掌握，同时有着足够的灵活性。Arduino 语言基于 wiring 语言开发，是对 avr-gcc 库的二次封装，不需要太多的单片机基础、编程基础，简单学习后，你也可以快速的进行开发。
- 开放性：Arduino 的硬件原理图、电路图、IDE 软件及核心库文件都是开源的，在开源协议范围内里可以任意修改原始设计及相应代码。
- 发展迅速：Arduino 不仅仅是全球最流行的开源硬件，也是一个优秀的硬件开发平台，更是硬件开发的趋势。Arduino 简单的开发方式使得开发者更关注创意与实现，更快的完成自己的项目开发，大大节约了学习的成本，缩短了开发的周期。

因为 Arduino 的种种优势，越来越多的专业硬件开发者已经或开始使用 Arduino 来开发他们的项目、产品；越来越多的软件开发者使用 Arduino 进入硬件、物联网等开发领域。

另外，Arduino 有一款支持图形化积木式编程的工具 ArduBlock。ArduBlock 软件

是 Arduino 官方编程环境的第三方软件，目前必须依附于 Arduino 软件下运行，区别于 Arduino 文本式编程环境，ArduBlock 是以图形化积木搭建的方式编程的，这样的方式会使编程的可视化和交互性加强，编程门槛降低，即使没有编程经验的人也可以尝试给 Arduino 控制器编写程序。所以，我们本次实践创新课程会基于 Arduino 平台，并且用到 ArduBlock。

三、 课前自学要求

- 阅读实验指导书，查阅相关文献，了解简单控制系统的基本原理
- 阅读实验指导书，查阅相关文献，了解 Arduino 核心板的工作原理

实验一：LED 控制

一、 实验目的

- 了解 Arduino 开源硬件平台
- 熟悉开发环境，建立相应开发工程
- 理解程序设计过程及程序基本结构
- 了解发光二极管（LED）控制基本原理

二、 实验内容

将 PC 机与实验板通过 USB 串口线连接，打开 Arduino IDE 及 ArduBlock。使用 ArduBlock 可视化编程方式设计程序令 LED 闪烁。

三、 背景知识

3.1 LED 元件及其工作原理

本实验的主要操作对象是 LED 灯，LED 灯的电路图如下：



当管脚（又称引脚、针脚）PIN13 输出高电压时，LED 灯就会亮；输出低电压时，LED 灯就会灭。如果想要让灯闪，那么需要亮一会儿，灭一会儿。

所以我们的目的就是让管脚 13 输出高电压一段时间，输出低电压一段时间，然后再输出高电压，再输出低电压，这样循环起来。

3.2 ArduBlock

如图所示，首先打开 ArduinoIDE 开发环境，并从“工具”菜单中选择 ArduBlock。

打开 ArduBlock 后，点击左侧的积木区“控制”组，就会弹出很多黄色的积木，其中有一个“主程序”，代表 Arduino 程序运行的主循环，拖动出来就是这样：



在这个黄色积木区里面，可以加很多东西，比如让某个管脚的电压升高、降低等等，实际上，ArduBlock 有一个积木是专门用来操作 LED 灯的，为 DFRobot 组中的 LED 模块：



本 Arduino 上 LED 的针脚号为 13，故需要将针脚值改为 13，下面状态有两种，ON 和 OFF，对应 LED 灯的开和关，把鼠标放在那个淡紫色的小积木上，小积木的右上方就会出现一个小三角，点击那个小三角，就能改变状态了（ON 和 OFF）。这一积木的功能可以用引脚中的“设定数字针脚值”积木取代。

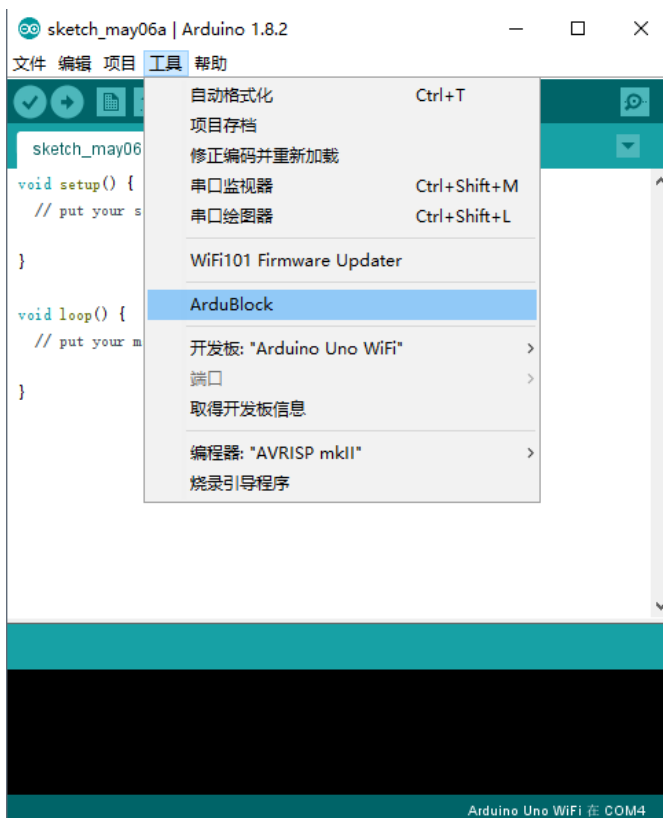
另外，控制组里也有表示延时的积木，延时是指等待，比如把灯打开，然后延时 1 秒，再把灯关闭，我们就是通过这种方式来告诉 Arduino 该等待多长时间的。将这个积木插到哪里就是指哪里需要延时，延时的时间可以在积木上输入：



切记不可想当然地直接把 LED 亮、暗的逻辑连起来但是不加入延时。如果这样做，在你最终看到的结果上就是 LED 灯亮着但是不闪，原因是 Arduino 的程序执行速度太快了，导致灯非常高速地闪以至于肉眼分辨不清。一般肉眼可分辨的 LED 闪的频率对应的最小延时为几十毫秒。

注意只有主程序框里面的积木是有效的，外面的积木无效，要执行的逻辑要连接此积木。编程区的积木拖动到左侧积木区，它们就会在编程区消失。

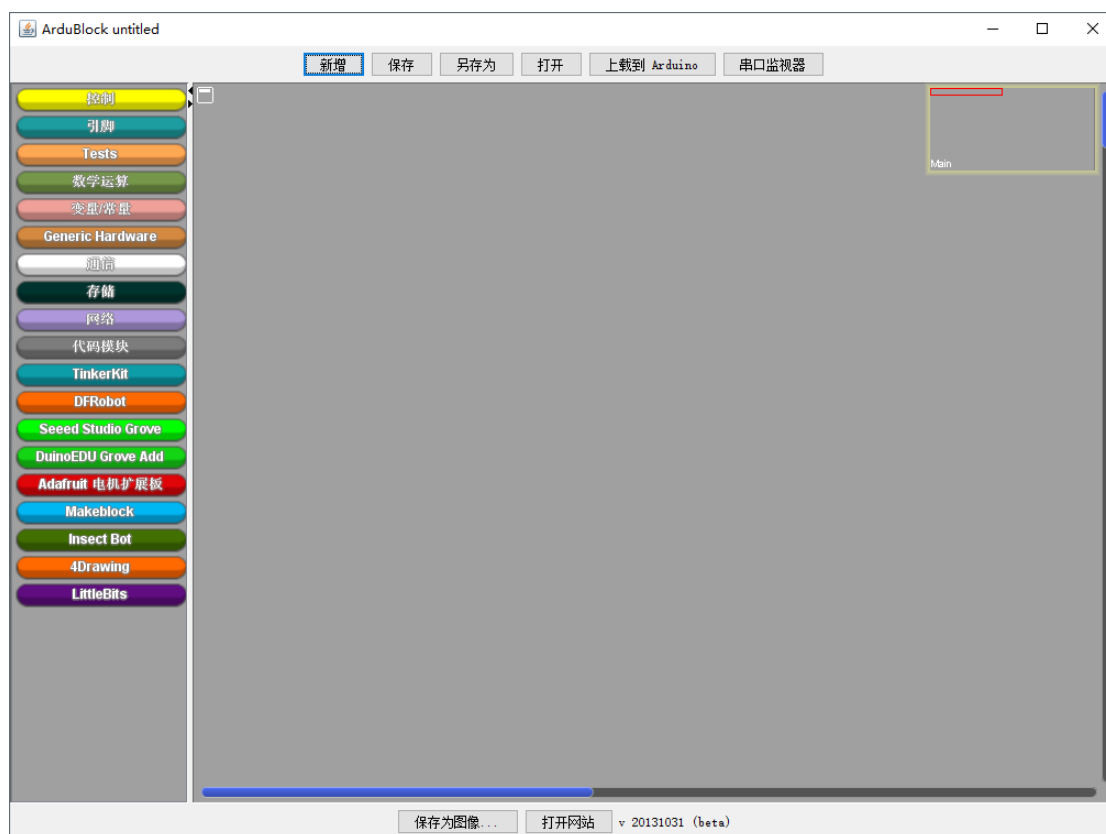
ArduBlock 的细节可以参考附录《ArduBlock 教育版》使用帮助。





四、 实验步骤

- 将蓝色的 USB 线一端连接 Arduino 实验板，另一端插入电脑的 USB 口；
- 打开 Arduino IDE，依次点击工具->ArduBlock，打开 ArduBlock。



- 积木式编程，操作位于管脚 13 的 LED（LED 灯的积木位于 DFRobot 分组中）灯开一秒、关一秒的循环。点击“上载到 Arduino”，正常条件下可以看到在 Arduino IDE 的状态条上依次会出现“上载成功”字样然后等待片刻，直到在 Arduino IDE 的状态条上出现“正在编译”、“上传”、“上传成功”等字样，在 Arduino IDE 代码区还可以看到自动生成的 C 代码。观察实验板上的 LED 闪烁现象。
- 具备 C 语言或其他高级语言编程基础者可自行选择直接使用高级语言编程。

搭建的 ArduBlock 积木参考程序：



在 Arduino IDE 中可以看到自动生成的 C 代码：

```
void __ardublockDigitalWrite(int pinNumber, boolean status)
{
    pinMode(pinNumber, OUTPUT);
    digitalWrite(pinNumber, status);
}

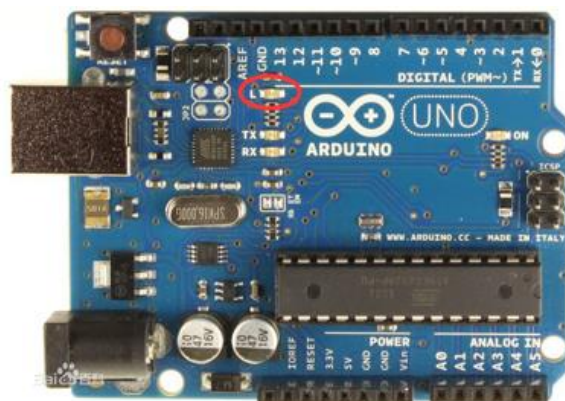
void setup()
{
}

void loop()
{
    __ardublockDigitalWrite(13, HIGH);
    delay( 1000 );
    __ardublockDigitalWrite(13, LOW);
    delay( 1000 );
}
```

五、 实验现象及结果

LED 灯按照预期方式闪烁，亮暗间隔为 1 秒。

- 修改程序，降低延时时间，调整闪烁频率，重复上述步骤。观察延时时间下降到何时会看到常亮的 LED 灯。



实验二：异步串口通信

一、实验目的

- 掌握基本的程序设计逻辑
- 了解异步串口通信的基本原理
- 锻炼快速学习和知识整合能力

二、实验内容

1. 让 PC 机与实验板建立串口链接，PC 机通过串口监视器向实验板的串口发送字符，实验板将收到的字符回发给 PC 机。
2. 让 PC 机与实验板建立串口链接，PC 机向实验板的串口发送指令，控制 LED 灯的亮暗：发送 1 则点亮 LED；发送 0 则熄灭 LED。

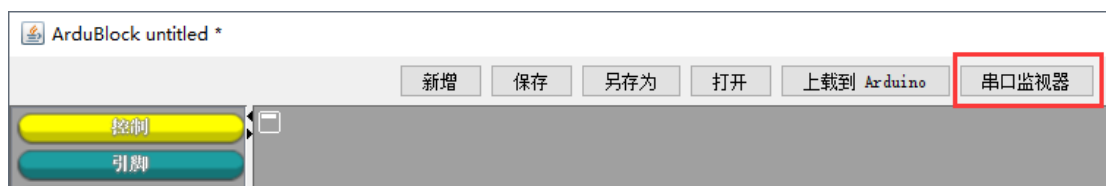
三、背景知识

3.1 串口及其工作原理

串口，即串行接口 (Serial Interface)。串行是指数据一位一位地顺序传送，其特点是通信线路简单，只要一对传输线就可以实现双向通信（可以直接利用电话线作为传输线），从而大大降低了成本，特别适用于远距离通信，但传送速度较慢。一条信息的各位数据被逐位按顺序传送的通讯方式称为串行通讯。串行通讯的特点是：数据位的传送，按位顺序进行，除信号回路外，最少只需一根传输线即可完成；成本低但传送速度慢。串行通讯的距离可以从几米到几千米，最常见的串口即异步串口。

串口是 Arduino 与其它设备进行通信的接口之一。Arduino 的串口操作是双向的，即既可以从 PC 上通过 Arduino 串口向 Arduino 实验板发送数据，也可以从 Arduino 实验板向 PC 发送数据。当你通过 USB 线缆使用串口功能的时候，串口已经被占用了，这就意味着你不能再使用板上相应的串口引脚了。

在 Arduino 控制器上，串口对应 Rx 和 Tx 两个引脚。在 UNO 板上，Arduino 的 USB 端口通过一个 USB-串口转换芯片与这两个串口引脚连接。在 PC 计算机上会通过 USB 接口虚拟一个用于 Arduino 通信的串口。在 ArduBlock 工具区点击“串口监视器”按钮即可打开串口监视器，通过虚拟串口与实验板双向通信：



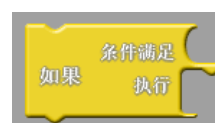
此监视器可以接收 Arduino 实验板向 PC 发送的消息，并通过串口向实验板发送信

息：



3.2 串口收发流程与判断逻辑

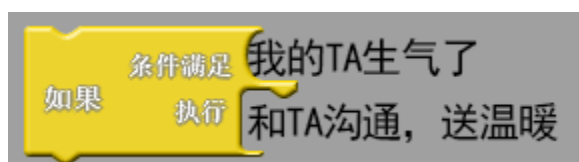
Arduino 串口接收操作过程：先判断串口数据是否已经到达，如果到达则可以读串口，并对串口数据进行处理。用于判断逻辑的积木如右图：



而判断串口数据是否已经到达的积木如下：



最外边橙色的方块是条件，中间是两个等于号，两侧各有两个洞，用来放小块积木，这个积木的意思是，当前的情况是不是串口（serial）数据（data）是可用（available）这一条件是真的（==1），这个积木就可以整块放进黄色的条件满足里面，意思是如果条件满足这个橙色积木所表示的意思，就执行“执行”后面的逻辑，如：



上面的积木翻译成我们平时说的话（自然语言）就是：如果我的 TA 生气了，那么就 就和 TA 沟通，送温暖；反之，如果 TA 不生气，就不执行“和 TA 沟通，送温暖”的逻辑。

然后需要定义字符变量，把串口读到的值赋值给字符变量，然后就可以通过此变量访问串口读到的字符。定义字符变量的积木如下：



字符就是如 ‘a’，‘b’，‘1’，‘3’，这种形式的数据，定义字符变量的意思就是造一个容器，并给这个容器取一个名字，把一个字符放进去，以后想找这个字符就可以通过这个容器的名字来找。上面那个绿色积木的意思是定义一个变量，取名为 HoHoHo，并将里面放入 ‘A’ 这个字符，一般而言，这种定义之后立刻进行赋值的行为叫做初始化，我们可以把变量初始化为串口发过来的数据，然后这个数据被存在变量里了：



如果需要从 Arduino 向 PC 发送数据，则可以使用“串行打印”积木：



上面这个积木就是串口打印 XiYangYang 并且另起一行。



上面这个积木就是串口打印变量“HoHoHo”并且另起一行。

注：由于软件设计问题，如果要使用串口打印功能，程序中请务必出现以下两个积木之一，有了它们之一串口才能成功初始化：



3.3 多重判断逻辑

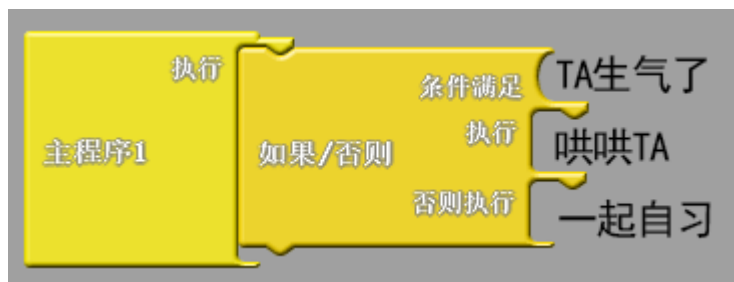
第二个实验内容是前面实验的综合，需要用到稍微复杂一点的逻辑，像下图这样：



这种逻辑为判断“TA 是否生气”，根据 TA 是否处于生气的状态，决定下一步该做什么，上面的积木表示的意思是如果 TA 生气了，那么就哄哄 TA，如果没生气就不用哄，

一起自习就好。

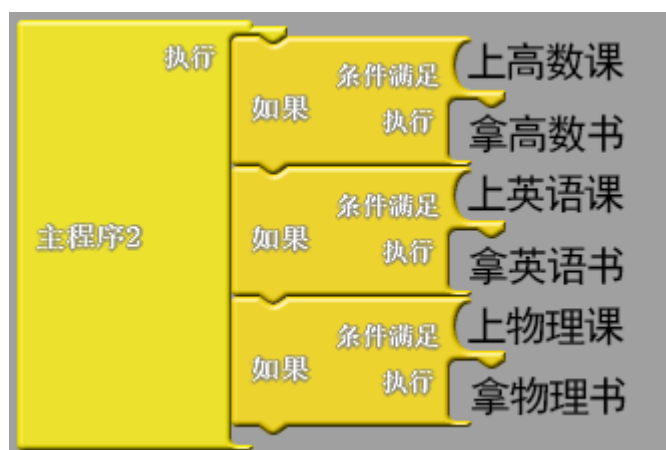
上面这种逻辑还可以使用另一种积木结构来表示：



此积木的逻辑和上面相同，如果/否则积木块的意义为如果满足某种条件，则执行第一个逻辑块，否则执行第二个逻辑块。另外，这种逻辑是可以嵌套的：



上面的逻辑和下面这种逻辑的效果相似：



推荐使用上面的逻辑。

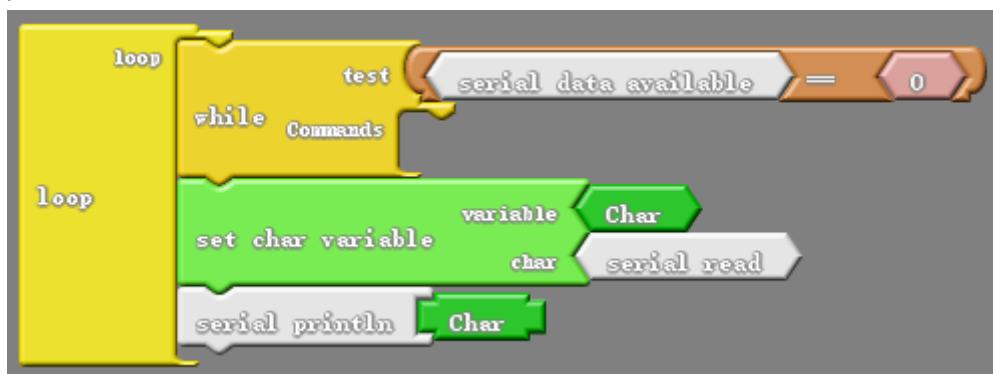
四、 实验步骤

- 将蓝色的 USB 线一端连接 Arduino 实验板，另一端插入电脑的 USB 口
- 打开 Arduino IDE，依次点击工具->ArduBlock，打开 ArduBlock。
- 积木式编程后，点击 ArduBlock 面板上的“上载到 Arduino”。
- 点击 ArduBlock 面板上的串口监视器，测试串口读写，观察 LED 灯的状况。

五、 实验现象及结果

向串口发送信息，能得到 Arduino 实验板串口的预期回应。

参考积木程序：



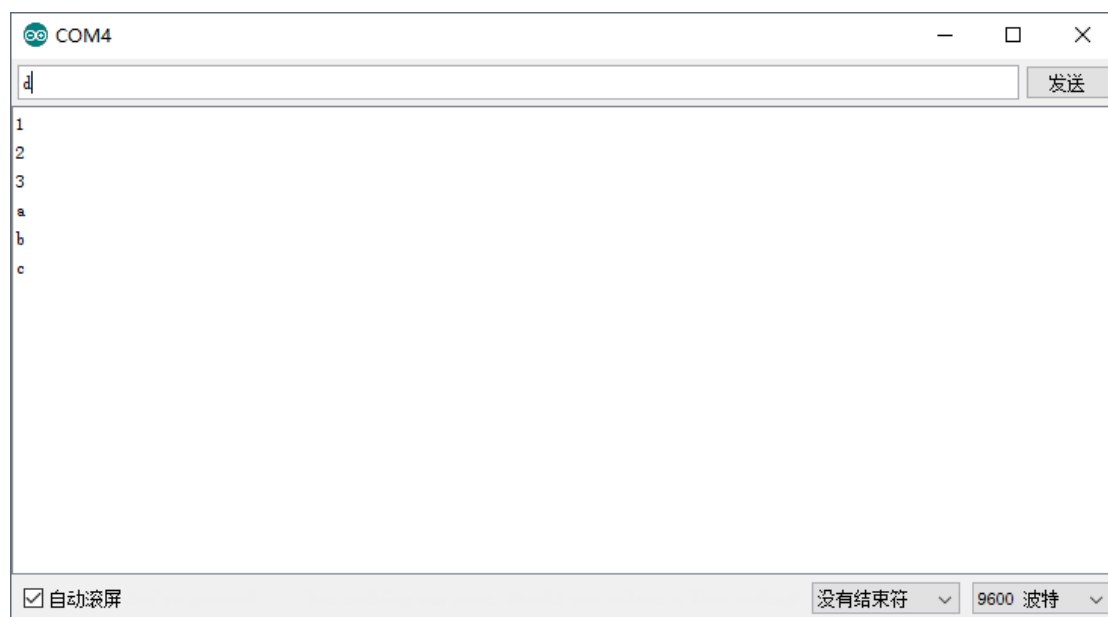
观察自动生成的源代码：

```
char Char = ' ' ;

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

void loop()
{
    while ( ( ( Serial.available() ) == ( 0 ) ) )
    {
    }

    Char = 'A';
    Serial.println(Char);
}
```



向串口发送信息，能按照预期控制 LED 灯。请自行设计程序逻辑。

实验三：扬声器发声与音乐播放

一、实验目的

- 了解扬声器（或蜂鸣器）发声原理
- 对方波有感性的认识，了解扬声器发声和播放音乐的基本原理
- 了解通过通用输入/输出接口（GPIO）控制外部设备的方式

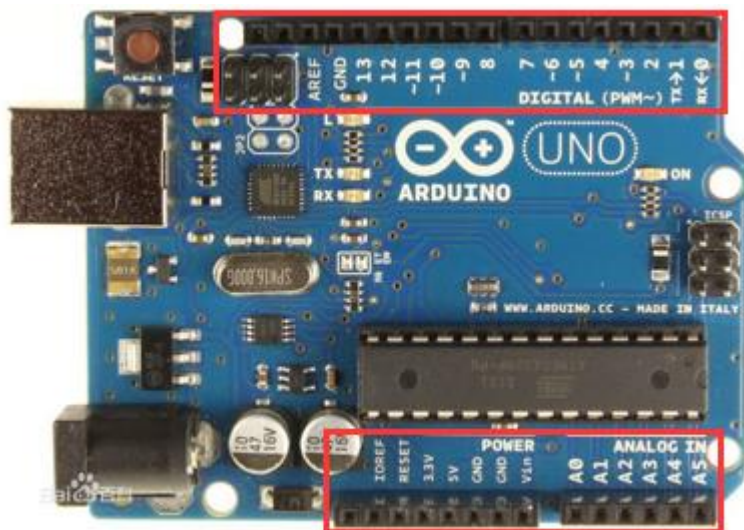
二、实验内容

1. 让扬声器（蜂鸣器）按照指定的频率发声。
2. 使用串口发送 1-7 分别使扬声器（蜂鸣器）发出 D0-TI 的音调（选做）。
3. 使用扬声器（蜂鸣器）发声和延时，播放乐曲。

三、背景知识

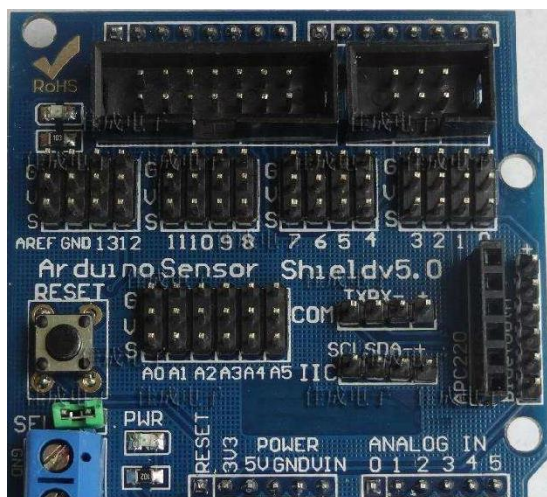
3.1 GPIO 及扬声器发声原理

在 Arduino 实验板上，有很多的针脚，针脚旁边都有对应的标号：

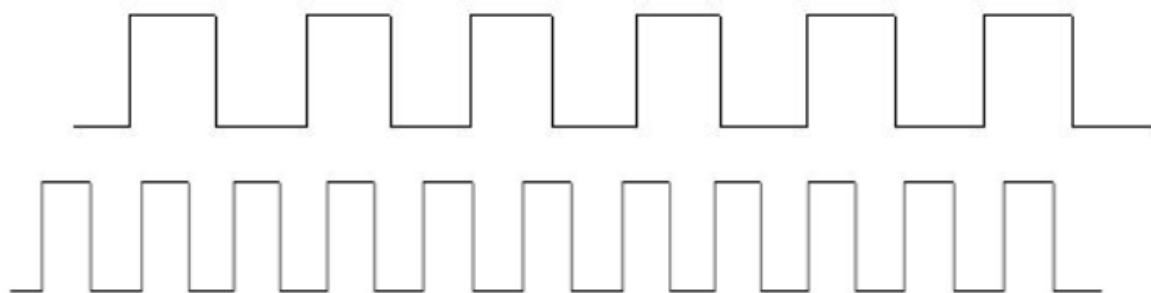


标号为数字的一般为通用输入/输出接口（GPIO）端口，标为 GND 的是地线，标为 3.3V 的表示 Arduino 实验板上电后此针脚电压一直是 3V，标为 5V 的同理。

本实验的实验设备在 Arduino 实验板的上面加了一层扩展板，Arduino 实验板上的很多针脚和接口都被连接到扩展板上了，扩展板如下图所示。



本实验需要用到扬声器（喇叭）（或者蜂鸣器），需要将扬声器（蜂鸣器）正极接到信号源，负极接到 GND。对于扬声器（蜂鸣器）来说，输入的波形频率越大，它发出的声音就越高：



例如上面的波形对应的声音较低，下面的波形对应的声音较高。

我们需要用 GPIO 来生成这种波形。

Arduino 中已经有专门为扬声器（蜂鸣器）发声而生成各种方波的积木：

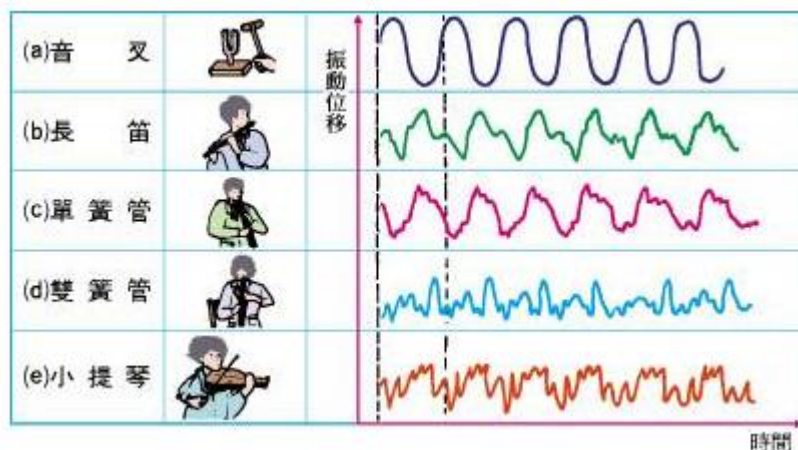


上面的积木有效的前提是扬声器（蜂鸣器）的正极接 Arduino 实验板的针脚 8，在针脚 8 输出频率为 440Hz 的方波。

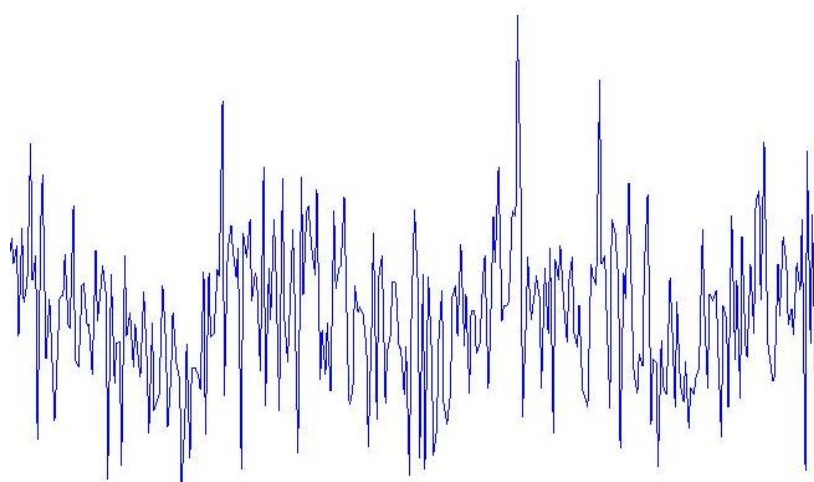
3.2 音乐播放原理

本实验涉及简单乐理知识。首先，噪声也有高低之分，也有 do re mi fa so la si，声音的高低只与波形的频率有关，但一般情况下乐音比噪声悦耳（锣鼓等打击乐器除外），原因在于乐音的音色相比于噪声而言更加有优势。

乐音：



噪声：



如果我们传方波，声音的音色应该在传统意义上的噪声和乐音之间，并且扬声器会比蜂鸣器好听一些。为了简单，本实验还是采取播放方波的方式，有兴趣的同学可以在后续学习过程中挑战正弦波。

基本的控制方式和实验步骤 1 类似，利用延时来控制节奏，将不同的频率发声连接起来，组成乐曲。

音调和频率的对应关系：

音调	简谱符号	低音频率	中音频率	高音频率
Do	1	262	523	1046
Re	2	294	587	1175
Mi	3	330	659	1318
Fa	4	349	698	1397
So	5	392	784	1568
La	6	440	880	1760
Ti	7	494	988	1967

四、 实验步骤

- 如果扩展板上没有连接扬声器或蜂鸣器，需首先将蜂鸣器与 Arduino 主板连接。**断开实验板电源及 USB 连接线**，将扬声器（蜂鸣器）的正极接到 8 号口的 S 端，负极连接到 8 号口的 G 端，如右图所示。
- 将蓝色的 USB 线一端连接 Arduino 实验板，另一端插入电脑的 USB 口。
- 打开 Arduino IDE，依次点击工具 ->ArduBlock，打开 ArduBlock。
- 积木式编程让实验板发出单一频率的声音，点击 ArduBlock 面板上的“上载到 Arduino”。
- 听扬声器（蜂鸣器）的声音。
- 调整输出频率，重复执行程序。
- 修改程序，使用串口发送 1-7 分别使扬声器（蜂鸣器）发出 D0-SI 的音调（选做）。
- 自行选择歌曲，修改程序，按照乐谱连续发出相应的频率的声音并延迟，奏出乐曲。



粉刷匠

1=F $\frac{2}{4}$
中速 欢快地

5 3	5 3	5 3	1	2 4	3 2	5	-
我是一个 粉刷匠， 粉刷本领强，							
5 3	5 3	5 3	1	2 4	3 2	1	-
我要把那 新房子 刷得很漂亮。							
2 2	4 4	3 1	5	2 4	3 2	5	-
刷了房顶 又刷墙， 刷子飞舞忙，							
5 3	5 3	5 3	1	2 4	3 2	1	-
哎呀我的 小鼻子 变呀变了样。							

〔波〕 佳基洛夫斯卡词
列申斯卡曲
曹永生译配

生日快乐

1=G $\frac{3}{4}$

米尔彻丽特
帕丽·希尔词曲

5 5 | 6 5 i | 7 0 5 5 | 6 5 2 | i - 5 5 | 5 3 i |
 Happy birth-day to you! Happy birth-day to you! Happy birth-day
 祝你生日快乐! 祝你生日快乐! 祝你生日快

7 6 - | 6 0 4 4 | 3 i 2 | 1. i - (5 5) :|| 2. i - 0 ||
 to my dear Happy birth day to you! you!
 乐我亲爱的, 祝你生日快乐! 乐!

实验四：舵机控制

一、实验目的

- 了解舵机的基本控制原理，熟悉舵机的控制逻辑
- 熟悉机械臂的运动规律

二、实验内容

- 使用串口控制机械爪的开合。
- 通过程序自定义机械臂动作，使机械臂在各个方向运动到最大范围，观察动作和想象中的差距，并分析原因。
- 编程实现机械臂“跳舞”（机械臂有规律地重复动作）。（可选）。

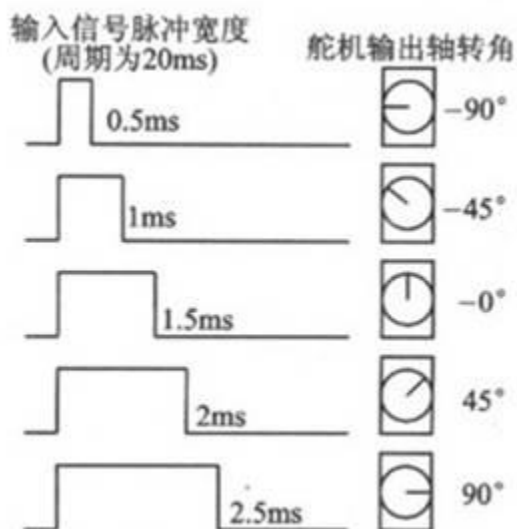
三、背景知识

舵机也叫伺服电机，最早用于船舶上实现其转向功能，由于它具有体积小、力矩大、外部机械设计简单、稳定性高，并可以通过程序连续控制其转角等特点，因而被广泛应用智能小车以实现转向以及机器人各类关节运动中。

舵机的输入线共有三条，如图所示，红色是电源线，黑色的是地线，这根线给舵机提供最基本的能源保证，主要是电机的转动消耗。电源有两种规格，一是 4.8V，一是 6.0V，分别对应不同的转矩标准，即输出力矩不同，6.0V 对应的要大一些，具体看应用条件；另外一根线是控制信号线，一般为白色或桔黄色。



舵机的控制信号周期为 20ms 的脉宽调制（PWM）信号，其中脉冲宽度从 0.5-2.5ms，相对应的舵盘位置为 0-180 度，呈线性变化。也就是说，给它提供一定的脉宽，它的输出轴就会保持一定对应角度上，无论外界转矩怎么改变，直到给它提供一个另外宽度的脉冲信号，它才会改变输出角度到新的对应位置上。舵机内部有一个基准电路，产生周期为 20ms、宽度 1.5ms 的基准信号，有一个比较器将外加信号与基准信号相比较，判断出方向和大小，从而生产电机的转动信号。由此可见，舵机是一种位置伺服驱动器，转动范围不能超过 180 度，适用于那些需要不断变化并可以保持的驱动器中，比如说机器人的关节、飞机的舵面等。



Ardublock 中的舵机积木经过高度封装，我们只需选择舵机端口，然后指定它转到相应角度就可以了，PWM 的操作 Ardublock 已经帮我们做了：



上面的积木意为将连接到引脚 9 的舵机转到 90 度。

本实验采用的机械臂未经调零，故同样的代码在不同的机械臂上运行的具体动作表现可能不同，但是依旧不影响操作。

四、 实验步骤

- 如果主板上没有连接扩展板，需首先将扩展板与 Arduino 主板连接。**断开实验板电源及 USB 连接线**，确保扩展板正面右下角的 ANALOG IN 插头与主板上的 ANALOG IN 插针座完全对齐，不能错位，将扩展板轻轻按压牢固。在扩展板上确认已经按照下面的规范连接了机械臂的舵机控制信号线：
 - ✧ 数字 I/O 引脚 9：机械爪舵机
 - ✧ 数字 I/O 引脚 11：底座旋转舵机
 - ✧ 数字 I/O 引脚 13：右侧小臂舵机
 - ✧ 数字 I/O 引脚 12：左侧大臂舵机
- 将蓝色的 USB 线一端连接 Arduino 实验板，另一端插入电脑的 USB 口。如果 USB 供电不足，可以将电源适配器与 Arduino 实验板连接。
- 打开 Arduino IDE，依次点击工具->ArduBlock，打开 ArduBlock。
- 积木式编程，通过串口控制机械臂的机械爪开合：发送 1 则机械爪打开；发送 0 则机械爪闭合。

- 点击 ArduBlock 面板上的“上载到 Arduino”。
- 修改程序，控制舵机在各个方向上运动，摸索机械臂的运动范围极限。注意：
 - ✧ 控制舵机连续动作时，注意每次动作后需要延时 2 秒（2000ms）以上。
 - ✧ 机械臂向下运动时可能会低于桌面，务必将装置放置在桌面边缘位置，并令机械臂伸展到桌面边缘之外，以免机械臂在运动时碰触桌面。
- 编程实现机械臂“跳舞”（可选）。

扩展实验一：机械臂抓物品游戏

一、 实验目的

- 了解机器人的控制方式，体会机器人在工业生产中的工作方式
- 培养对智能机器人的兴趣

二、 实验内容

编写程序令机械臂完成抓取轻物品的操作。推荐物品为纸巾团、橡皮擦等。

三、 背景知识

在工业生产过程中，经常会出现重复性较高的工作，这种工作包括手机装配、汽车生产等等，一些劳动力比较昂贵的国家经常会使用机械臂来代替人力完成重复性的生产加工操作。

四、 实验步骤

- 如果主板上没有连接扩展板，需首先将扩展板与 Arduino 主板连接。**断开实验板电源及 USB 连接线**，确保扩展板正面右下角的 ANALOG IN 插头与主板上的 ANALOG IN 插针座完全对齐，不能错位，将扩展板轻轻按压牢固。在扩展板上确认已经按照下面的规范连接了机械臂的舵机控制信号线：
 - ✧ 数字 I/O 引脚 9：机械爪舵机
 - ✧ 数字 I/O 引脚 11：底座旋转舵机
 - ✧ 数字 I/O 引脚 13：右侧小臂舵机
 - ✧ 数字 I/O 引脚 12：左侧大臂舵机
- 将蓝色的 USB 线一端连接 Arduino 实验板，另一端插入电脑的 USB 口。如果 USB 供电不足，可以将电源适配器与 Arduino 实验板连接。
- 打开 Arduino IDE，依次点击工具->ArduBlock，打开 ArduBlock。
- 积木式编程，通过程序控制机械臂的机械爪抓取物体移动到指定位置。
- 点击 ArduBlock 面板上的“上载到 Arduino”。
- 调整被抓取物的位置和对机械臂的指令，体会控制精度对工业机械臂性能的重要性。

扩展实验二：机械臂之舞

五、 实验目的

- 深入理解机器人的控制方式，体会机器人在节奏中的美感
- 将所有基础实验的内容结合，并对 Arduino 系列实验进行总结

六、 实验内容

通过串口以数字的形式一次性输入一段短暂的旋律，机器人能够实现读取并播放，并且每一个音高播放时对应一种动作，同时使用 LED 灯闪烁表示节拍，从而实现可编程的机械动作舞蹈。

七、 背景知识

在市场上有很多电子类玩具，通常是结合音乐进行相关的舞蹈表演，本次实验就是要实现类似的一个玩具，考察大家对前几个实验中所学到的知识的整合及逻辑设计能力。

八、 实验步骤

- 如果主板上没有连接扩展板，需首先将扩展板与 Arduino 主板连接。**断开实验板电源及 USB 连接线**，确保扩展板正面右下角的 ANALOG IN 插头与主板上的 ANALOG IN 插针座完全对齐，不能错位，将扩展板轻轻按压牢固。在扩展板上确认已经按照下面的规范连接了机械臂的舵机控制信号线：
 - ◇ 数字 I/O 引脚 9：机械爪舵机
 - ◇ 数字 I/O 引脚 11：底座旋转舵机
 - ◇ 数字 I/O 引脚 13：右侧小臂舵机
 - ◇ 数字 I/O 引脚 12：左侧大臂舵机
- 按照实验三的提示连接扬声器。
- 将蓝色的 USB 线一端连接 Arduino 实验板，另一端插入电脑的 USB 口。如果 USB 供电不足，可以将电源适配器与 Arduino 实验板连接。
- 打开 Arduino IDE，依次点击工具->ArduBlock，打开 ArduBlock。
- 积木式编程，通过程序控制机器人通过扬声器播放音乐，同时机器人配合音乐的旋律做出预先设计的动作。
- 点击 ArduBlock 面板上的“上载到 Arduino”。
- 调整输入的旋律，体会电子类玩具的制作原理。

扩展实验三：自主实验（选做）

一、 实验目的

- 培养自主创新意识和自主学习能力、资源整合能力

二、 实验内容

利用现有的机械臂硬件，你可以实现哪些自主创新应用？
工作量大的复杂实验内容可以多人分工合作协作完成。

课后作业与实验报告

一、 课后作业

1. 查询参考资料，了解开源创客项目及开源硬件和开源软件的基本概念与发展现状。
2. 进一步完善自主设计的智能应用场景，总结设计与实现过程中遇到的问题，分析其原因与改进办法。

二、 分组讨论参考题目

1. 总结智能硬件与开源创客发展的历史与现状，以及对未来发展的展望。
2. 智能机器人的应用场景有哪些？机器人能替代哪些工种的人类工作？

三、 实验报告要求

将实验内容总结成一份实验报告。基本内容要求：

1. 封面。参考内容如下：

北京邮电大学实践课程实验报告

课程名称	创新创业实践	学 院	计算机学院	指导教师	(空白)
知识模块	智能机器人	完成时间		201x 年 x 月 x 日	
班 级	班内序号	学 号	学生姓名	成绩	
XXX	XXX	XXX	XXX	(空白)	

2. 针对实验三至实验五（实验六）的每个实验，分别说明实验的基本内容、基本原理，给出可视化编程程序和自动生成的高级语言源代码（或者直接用高级语言编写的源代码）、实验过程总结（思维过程、走过的弯路、遇到的难题、解决的办法等）。
3. 思考与收获。
4. 课程调查。说明自己具备哪些计算机相关技术或技能基础、是否学过/用过编程语言，对本实验课程的建议（如哪部分比较难，需要引导，你建议完成哪些新的实验题目等等）。

附录 《ArduBlock 教育版》使用帮助

ArduBlock 是 Arduino IDE 的可视化编程插件，它必须依附于 Arduino IDE 环境启动，是最受欢迎的 Arduino 编程入门工具之一。ArduBlock 教育版是开源课程《Arduino 创意机器人》所使用的编程工具，是著名创客何琪辰为中小学 Arduino 课程定制的版本。

ArduBlock 教育版对 Arduino IDE 版本则要求是 1.5 或更高版本。软件安装和简单，只要将下载的压缩包（下载地址：http://blog.sina.com.cn/s/blog_6611ddcf0101kfs7.html）中 libraries 和 tools 文件夹，复制到 Arduino 的安装目录下，重新运行 Arduino IDE 后，点击“工具”中的“ArduBlock”便可以启动，如图 1 所示。

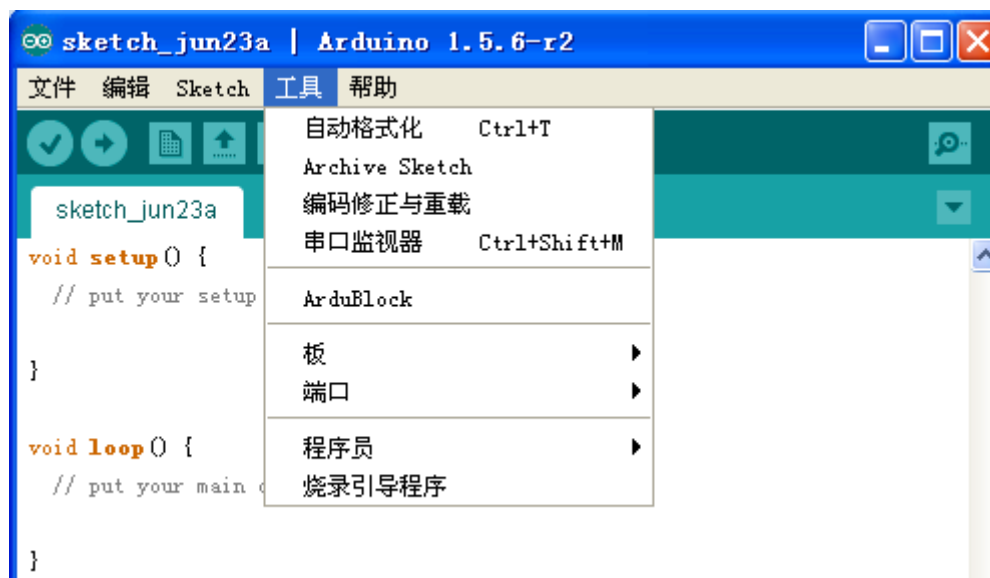


图 1 启动 ArduBlock

启动 ArduBlock 之后，我们会发现它的界面（如图 2 所示）主要分为三大部分：工具区（上），积木区（左），编程区（右）。其中，工具区主要包括保存、打开、下载等功能，积木区主要是用到的一些积木命令，编程区则是通过搭建积木编写程序的区域。下面将分别介绍这三个区域。

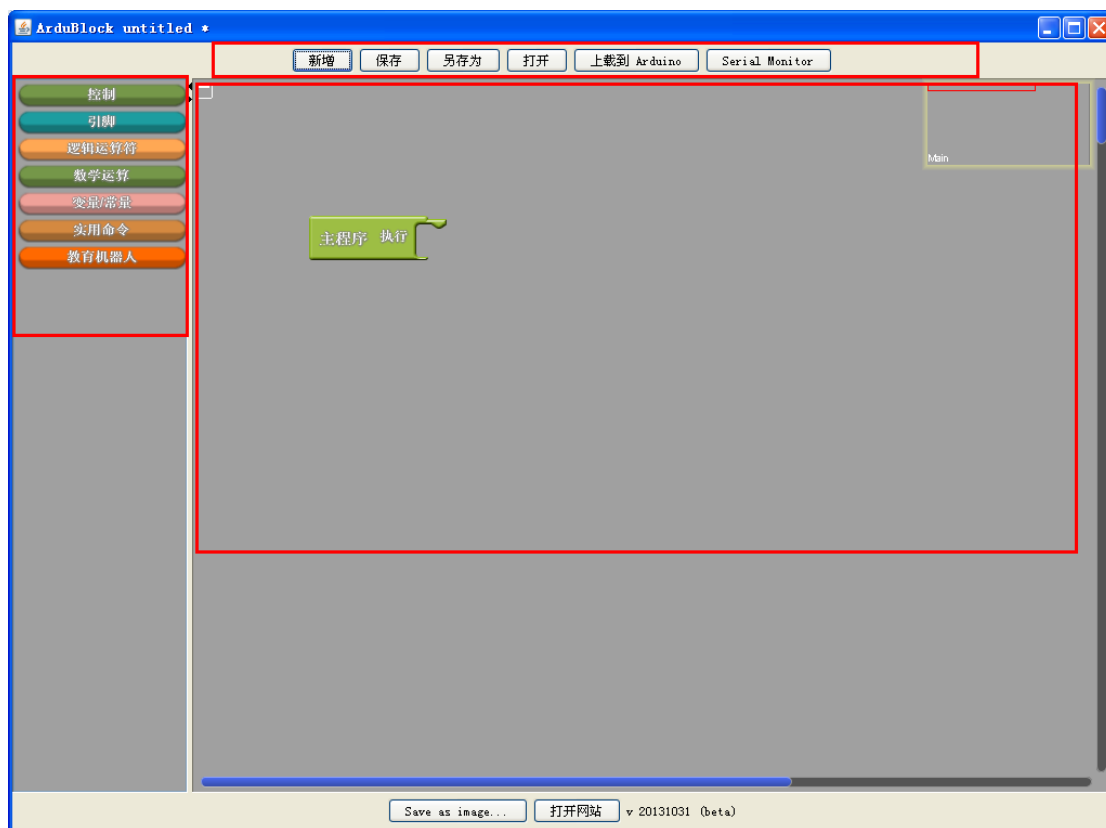


图 2 ArduBlock 界面

一、工具区

工具区包括“新增”“保存”“另存为”“打开”“上载到 Arduino”“Serial Monitor”，“新增”就是新建，“保存”“另存为”“打开”也都是其它软件的常用工具，这里就不做介绍了。点击“上载到 Arduino”，Arduino IDE 将生成代码，并自动上载到 Arduino 板子，需要注意的是在上载 Arduino 之前，要查看一下端口号和板卡型号是否正确。在点击“上载到 Arduino”之后，我们可以打开 Arduino IDE 查看程序是否上载成功。“Serial Monitor”则是打开串口监视器，串口监视器只有在电脑中有 Arduino 端口时才能打开。


二、积木区


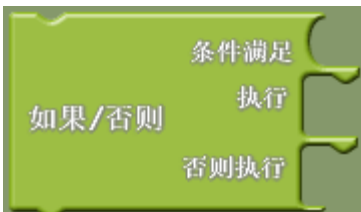


积木区包含了《Arduino 创意机器人》课程中用到的所有模块，还包括有一些课程中没有涉及但经常用到的模块。积木区的积木共分为七大部分：控制，引脚，逻辑运算符，数学运算，变量/常量，实用命令，教育机器人。

1.控制

控制中的各个模块都是一些最基本的编程语句，只要接触过编程的人都会很容易理解这里的各个模块的含义。控制中各模块释义如表 1 所示。

表 1 控制中各模块释义

模块	释义
	程序中只允许有一个主程序，主程序能够调用子程序，但不能被子程序调用

	这里的程序也是主程序，但不同于上一个的是，这里的“设定”和“循环”分别表示 IDE 中的 setup 和 loop 两个函数
	选择结构，如果条件满足……，执行……
	选择结构，如果条件满足……，执行……；否则执行……
	循环结构，当条件满足……，执行……，直到条件不满足时跳出循环
	循环结构，可设定循环的次数，然后执行……
	强制退出循环
	编写子程序
	调用子程序

2.引脚

引脚中的各个模块是针对 Arduino 板的引脚（也称针脚）所设计的，主要是数字针脚和模拟针脚，也包括一些常见的使用，比如舵机，超声波等。引脚中各模块释义如表 2 所示。

表 2 引脚中各模块释义

模块	释义
	读取数字针脚值（取值为 0 或 1）
	读取模拟针脚值（取值在 0-1023 之间）
	设定一般数字针脚的值（0 或 1）


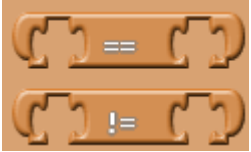





	设定支持 PWM 的数字引脚的值（0-255 之间） 以 UNO 为例，支持 PWM 的数字引脚有 3, 5, 6, 9, 10, 11
	设定舵机（又称伺服电机）的引脚和角度 Arduino 中能够连接舵机的引脚只有 9 和 10
	专门针对 360 度的舵机，设定其引脚和角度
	设定超声波传感器的 trig 和 echo 的引脚 trig 为发射端，echo 为接收端
	读取 Dht11 温度和湿度的值
	设定扬声器（蜂鸣器）的引脚和频率
	设定扬声器（蜂鸣器）的引脚、频率和持续时间
	设定扬声器（蜂鸣器）为无声

3.逻辑运算符

逻辑运算符主要包括常见的“且”“或”“非”，还包括比较运算符，如数字值、模拟值和字符的各种比较。逻辑运算符中各模块释义如表 3 所示。

表 3 逻辑运算符中各模块释义

模块	释义
	模拟值和实数的比较，比较的两个值为模拟类型或实数类型，包括大于、小于、等于、大于等于、小于等于、不等于

	数字值的比较，比较的两个值为数字类型，包括等于、不等于
	字符的比较，比较的两个值为字符类型，包括等于、不等于
	逻辑运算符，也称“与”，上下两个语句都为真时整体（复合语句）为真，否则为假
	逻辑运算符，上下两个语句都为假时整体为假，否则为真
	逻辑运算符，表示对后面语句的否定
	比较字符串是否相等，比较的两个值为字符串类型
	判断字符串是否为空

4.数学运算

数学运算主要是 Arduino 中常用的基本运算，包括四则运算、三角函数、函数映射等。数学运算中各模块释义如表 4 所示。

表 4 数学运算中各模块释义

模块	释义
	四则运算，包括加、减、乘、除，要求符号两边为模拟值
	取模运算，又称取余或求余，要求符号两边为模拟值
	求绝对值
	乘幂运算，又称乘方运算
	求平方根

	三角函数，包括正弦、余弦、正切
	求随机数，随机数的范围在“最小值”和“最大值”之间
	映射，将一个数值（变量或常量）从一个范围映射到另一个范围

5. 变量/常量

变量/常量主要包括数字变量、模拟变量、字符变量、字符串变量以及它们对应的各种常量。变量/常量中各模块释义如表 5 所示。

表 5 变量/常量中各模块释义

模块	释义
	模拟常量
	给模拟变量赋值
	设定模拟变量（名），如果没有赋值，默认值为 0
	给数字变量赋值
	设定数字变量（名），如果没有赋值，默认值为 false (0)
	数字常量，高低电平值
	数字常量，真假值
	设定实数变量（名），如果没有赋值，默认值为 0.0

	给实数变量赋值
	实数常量，圆周率
	给字符变量赋值
	设定字符变量（名）
	设定字符串变量（名）
	字符串常量

6.实用命令

实用命令是常用到的一些命令，包括延迟、串口监视器的操作、红外遥控的操作等。实用命令中各模块释义如表 6 所示。

表 6 实用命令中各模块释义

模块	释义
 	延迟函数，单位是毫秒或微秒
	记录 Arduino 上电后到当前为止运行的时间
	读取串口的值
	通过串口打印并换行
	将字符串和模拟量结合，即将模拟量转换为字符串形式
	将字符串和数字量结合，即将数字量转换为字符串形式
	设定红外接收头的针脚
	获取红外遥控的指令
	读取 I2C，需要设备地址和寄存器地址

读取I2C是否正确	判断是否正确读取 I2C
-----------	--------------

三、编程区

编程区是程序编写的舞台，可以通过拖动右边和下边的滚动条来查看编程区。启动 ArduBlock 后，编程区会默认地放入一个主程序模块，因为主程序有且只能有一个，所以不能再继续往里面添加主程序模块了，如果再拖进去主程序模块的话，下载程序的时候会提示“循环块重复”。

除子程序执行模块外，所有积木模块都必须放在主程序内部。当搭建积木编写程序时，要注意把具有相同缺口的积木模块搭在一起，成功时会发出“咔”的一声。我们还可以对积木模块进行克隆或添加注释语句，只要选中该模块，右击就可以实现对该模块的克隆和添加注释操作；其中子程序执行模块还有另外一个功能就是创建引用，即点击之后会自动弹出调用该子程序的模块。

要删除某些积木吗？只要选择不需要的积木块，拖拽到积木区就不见了。

四、常见问题解答

1、问：子程序模块如何使用？

答：子程序中直接搭建程序模块，主程序中用到该子程序时直接调用即可，模块范例如图 3 所示。假如数字针脚 3 连接 LED，子程序则是实现 LED 的闪烁。



图 3 子程序使用范例

2、问：ArduBlock 提供了两个主程序模块，它们在使用时有什么区别？

答：两个主程序模块的使用范例如下（假设数字针脚 3 连接 LED）。



图 4 主程序 1 使用范例

如图 4，该程序在运行时先执行一次“设定”中的语句，再重复循环地执行“循环”中的语句。程序运行之后，LED 会亮 1 秒，之后一直保持灭的状态。



图 5 主程序 2 使用范例

如图 5，该程序在运行时会重复“执行”中的语句。程序运行之后，LED 会亮 1 秒，灭 1 秒，这样保持闪烁。