目录

[软件说明书 1](#_Toc31621574)

[第1章 认识Mixly 1](#_Toc31621575)

[1.1安装Mixly 1](#_Toc31621576)

[1.2认识Mixly 5](#_Toc31621577)

[1.3测试软件能否正常工作 9](#_Toc31621578)

[1.4任务：点亮一盏单色LED灯并闪烁 9](#_Toc31621579)

[1.4.1硬件连接 9](#_Toc31621580)

[1.4.2画流程图 10](#_Toc31621581)

[1.4.3开始编程 10](#_Toc31621582)

[1.4.4程序下载 11](#_Toc31621583)

[1.5安装Arduino IDE 12](#_Toc31621584)

[第2章 Mixly系统介绍 16](#_Toc31621585)

[2.1程序结构 16](#_Toc31621586)

[2.2输入输出 16](#_Toc31621587)

[2.2.1数字输入 17](#_Toc31621588)

[2.2.2数字输出 18](#_Toc31621589)

[2.2.3模拟输入 18](#_Toc31621590)

[2.2.4模拟输出 19](#_Toc31621591)

[2.2.5中断函数 20](#_Toc31621592)

[2.2.6其它模块简介 21](#_Toc31621593)

[2.3控制 22](#_Toc31621594)

[2.3.1初始化 23](#_Toc31621595)

[2.3.2延时 23](#_Toc31621596)

[2.3.3“如果”程序块 23](#_Toc31621597)

[2.3.4 switch语句 24](#_Toc31621598)

[2.3.5 for语句 25](#_Toc31621599)

[2.3.6重复执行 25](#_Toc31621600)

[2.2.7停止程序简介 25](#_Toc31621601)

[2.2.8跳出循环简介 26](#_Toc31621602)

[2.2.9程序运行时间（计时器1）简介 26](#_Toc31621603)

[2.3.10定时器模块简介 26](#_Toc31621604)

[2.4流程图表示程序 26](#_Toc31621605)

[2.4.1基本流程图介绍 26](#_Toc31621606)

[2.4.2顺序结构 27](#_Toc31621607)

[2.4.3分支/选择结构 28](#_Toc31621608)

[2.4.4循环结构 30](#_Toc31621609)

[2.4.5三种结构共同特点： 31](#_Toc31621610)

[2.5串口通信 31](#_Toc31621611)

[2.5.1“打印（自动换行）”程序块 32](#_Toc31621612)

[2.5.2“软串口初始化”程序块 32](#_Toc31621613)

[2.5.3其它模块简介 33](#_Toc31621614)

[2.6数学 33](#_Toc31621615)

[2.6.1“计算”程序块 33](#_Toc31621616)

[2.6.2“转换”程序块 33](#_Toc31621617)

[2.6.3“随机数”程序块 34](#_Toc31621618)

[2.6.4“约束”程序块 34](#_Toc31621619)

[2.6.5“映射”程序块 34](#_Toc31621620)

[2.6.6“位运算”程序块 35](#_Toc31621621)

[2.6.7“函数运算”程序块 35](#_Toc31621622)

[2.6.8其它模块简介 36](#_Toc31621623)

[2.7逻辑 36](#_Toc31621624)

[2.7.1“比较”程序块 36](#_Toc31621625)

[2.7.2“逻辑选择”程序块 37](#_Toc31621626)

[2.7.3“逻辑非”程序块 37](#_Toc31621627)

[2.7.4其它模块简介 37](#_Toc31621628)

[2.8文本 37](#_Toc31621629)

[2.8.1字符串程序块简介 38](#_Toc31621630)

[2.8.2字符程序块简介 38](#_Toc31621631)

[2.8.3字符串连接程序块简介 39](#_Toc31621632)

[2.8.4转换数值类型程序块简介 39](#_Toc31621633)

[2.8.5其它程序块简介 39](#_Toc31621634)

[2.9 数组 40](#_Toc31621635)

[2.9.1 初始化数组 40](#_Toc31621636)

[2.9.2 获取数组长度 42](#_Toc31621637)

[2.9.3 获取数组的某一项 42](#_Toc31621638)

[2.10 通信 42](#_Toc31621639)

[2.10.1 红外接收 43](#_Toc31621640)

[2.10.2 红外发射 43](#_Toc31621641)

[2.11执行器 43](#_Toc31621642)

[2.11.1 舵机 44](#_Toc31621643)

[2.11.2蜂鸣器 44](#_Toc31621644)

[2.12显示器 45](#_Toc31621645)

[2.12.1 LCD1602显示屏 45](#_Toc31621646)

[2.12.2 RGB灯 46](#_Toc31621647)

[2.12.3其它模块简介 47](#_Toc31621648)

[2.13变量 48](#_Toc31621649)

[2.13.1“变量声明”程序块 48](#_Toc31621650)

[2.13.2“变量赋值”程序块 49](#_Toc31621651)

[2.13.3“变量值类型”程序块 49](#_Toc31621652)

[2.14函数 50](#_Toc31621653)

[2.14.1“函数准备”程序块 50](#_Toc31621654)

[2.14.2“参数”程序块 50](#_Toc31621655)

[2.14.3“函数执行”程序块 51](#_Toc31621656)

[第3章 青橙创客模块区 51](#_Toc31621657)

[3.1传感器 51](#_Toc31621658)

[3.1.1数字开关类传感器 51](#_Toc31621659)

[3.1.2模拟型传感器 52](#_Toc31621660)

[3.1.3其他传感器 53](#_Toc31621661)

[3.2执行器 54](#_Toc31621662)

[3.2.1声类 54](#_Toc31621663)

[3.2.2光类 55](#_Toc31621664)

[3.2.3动类 56](#_Toc31621665)

[3.3通信 58](#_Toc31621666)

[3.3.1蓝牙 58](#_Toc31621667)

[3.3.2红外通信 59](#_Toc31621668)

[3.3.3PS2手柄 60](#_Toc31621669)

[3.4时钟 61](#_Toc31621670)

[3.4.1设置时间 61](#_Toc31621671)

[3.4.2显示时间 62](#_Toc31621672)

[3.5计时器 62](#_Toc31621673)

[3.5.1“间隔执行”程序块 63](#_Toc31621674)

[3.5.2“计时开始”程序块 63](#_Toc31621675)

[3.5.3“超时”程序块 64](#_Toc31621676)

[3.5.4“持续时间”程序块 64](#_Toc31621677)

[3.6人工智能 64](#_Toc31621678)

[3.6.1语音识别 64](#_Toc31621679)

[3.6.2 OpenMV模块 65](#_Toc31621680)

[3.6.3 MU魔图识别 65](#_Toc31621681)

[第4章 甜橙示例程序 70](#_Toc31621682)

[项目一 Hello,World! 72](#_Toc31621683)

[任务 1——点亮真实 LED 灯 72](#_Toc31621684)

[任务2——让LED越闪越快 75](#_Toc31621685)

[项目二 S.O.S. 78](#_Toc31621686)

[任务 1——用 LED 发出求救信号 78](#_Toc31621687)

[项目三 神奇的按钮 81](#_Toc31621688)

[任务1——简易延时灯 82](#_Toc31621689)

[任务2——使用按钮模拟开关(2) 85](#_Toc31621690)

[项目四 简易入侵检测仪 88](#_Toc31621691)

[任务1——简单入侵检测仪 88](#_Toc31621692)

[任务2——记录入侵 89](#_Toc31621693)

[项目五 可调灯 91](#_Toc31621694)

[任务1——简易呼吸灯 92](#_Toc31621695)

[任务2——制作5档可调灯 97](#_Toc31621696)

[任务3——制作旋钮可调灯 98](#_Toc31621697)

[项目六 智能灯 100](#_Toc31621698)

[任务1——声控灯 101](#_Toc31621699)

[任务2——制作楼道灯 103](#_Toc31621700)

[任务3——制作电子蜡烛 105](#_Toc31621701)

[任务4——制作简易门铃 107](#_Toc31621702)

[项目七 制作一个温湿度仪 109](#_Toc31621703)

[任务1——制作一个温度仪 109](#_Toc31621704)

[项目八 小小作曲家 113](#_Toc31621705)

[任务1——简单音阶 113](#_Toc31621706)

[任务2——简单乐曲 116](#_Toc31621707)

[项目九 噪音计 117](#_Toc31621708)

[任务1——制作一个噪音计 118](#_Toc31621709)

[任务2——探测最大噪音 120](#_Toc31621710)

[项目十 遥控灯 122](#_Toc31621711)

[任务1——制作一个红外遥控灯 123](#_Toc31621712)

[项目十一 超声波测距仪 124](#_Toc31621713)

[任务1——超声波模块测量距离 125](#_Toc31621714)

[项目十二 减速电机 128](#_Toc31621715)

[任务1——实现电机的正反转 128](#_Toc31621716)

[项目十三 蓝牙通信 130](#_Toc31621717)

[任务1——蓝牙控制单色灯亮灭 130](#_Toc31621718)

[项目十四 节日闪烁彩灯 133](#_Toc31621719)

[任务1——WS2812 全彩 LED 灯 133](#_Toc31621720)

[项目十五 DIY海盗船系列 135](#_Toc31621721)

[任务1——气压海拔检测 135](#_Toc31621722)

[任务2——车辆测速与倒地报警仪 137](#_Toc31621723)

软件说明书

# 第1章 认识Mixly

近年来，创客教育不断普及，以Arduino为基础的创意电子类创客教育课程正逐步在中小学中推广开来。然而，由于受到其本身技术门槛的限制，创意电子类课程的开设给广大师生提出了巨大的挑战（Arduino需要通过C/C++语言编程实现创意，这对低年龄段的学生来说几乎是不可能的）。创客教育需要更好的图形化编程工具，Mixly（米思齐）因此应运而生。Mixly，中文名为米思齐，全称为Mixly\_Arduino，是一款由北京师范大学教育学部创客教育实验室傅骞教授团队开发的图形化编程软件。

Mixly可以看作是介于普通用户与Arduino IDE之间桥梁，通过这个桥梁，即使用户不懂C语言的语法，也可以利用图形化程序编写Arduino程序。Mixly的基本原理是将图形化程序转化成C语言，再利用Arduino IDE上传到硬件中。

## 1.1安装Mixly

1、打开应用程序：安装包解压后如图1.1-1所示，然后打开文件夹选择Mixly应用程序，如图1.1-2所示，打开应用程序，如图1.1-3所示：



图1.1-1



图1.1-2

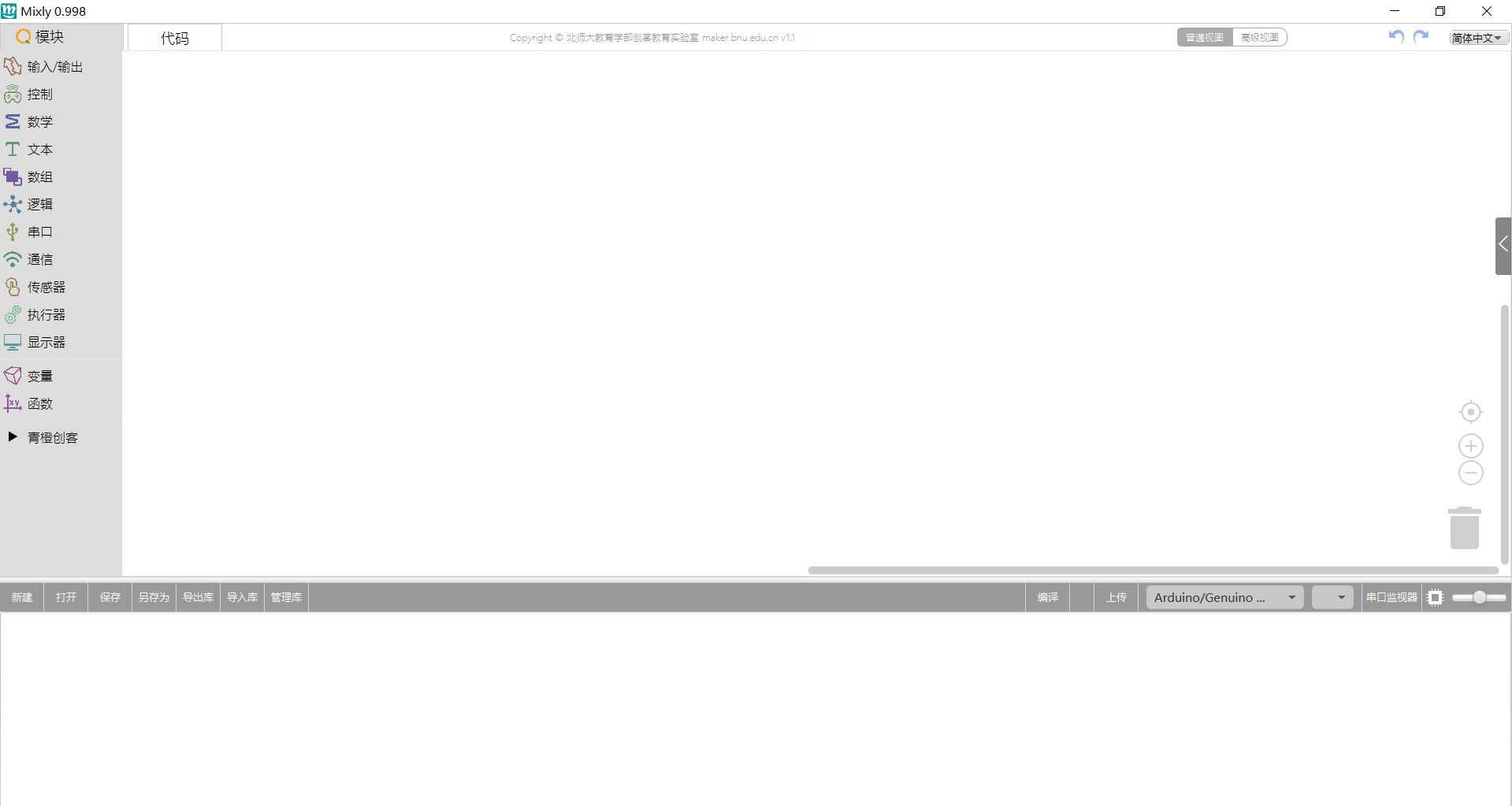


图1.1-3

2、安装驱动：端口驱动一般会包含在解压后的文件夹里，如图1.1-4所示，安装驱动前，需要知道待安装的电脑是32位的，还是64位的操作系统，可打开电脑“设置”，查看“系统”选项中的“关于”，如图1.1-5，可看到该电脑是64位的操作系统；或者打开“我的电脑”，点击右上角“属性”按钮，如图1.1-6会看到如图图1.1-7所示查看；或者直接在桌面右键“我的电脑”--选择“属性”按钮，也可以看到图图1.1-7所示界面。

安装端口驱动时，如图1.1-8，如果是64位的，就选择64位的安装包；然后点击打开，会弹出如图图1.1-9所示窗口，点击下一步，会弹出如图1.1-10所示窗口，点击完成；此时端口驱动程序安装完成。

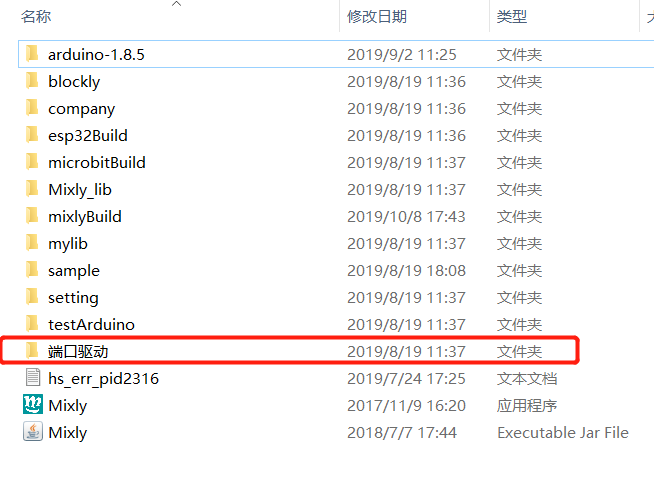


图1.1-4



图1.1-5

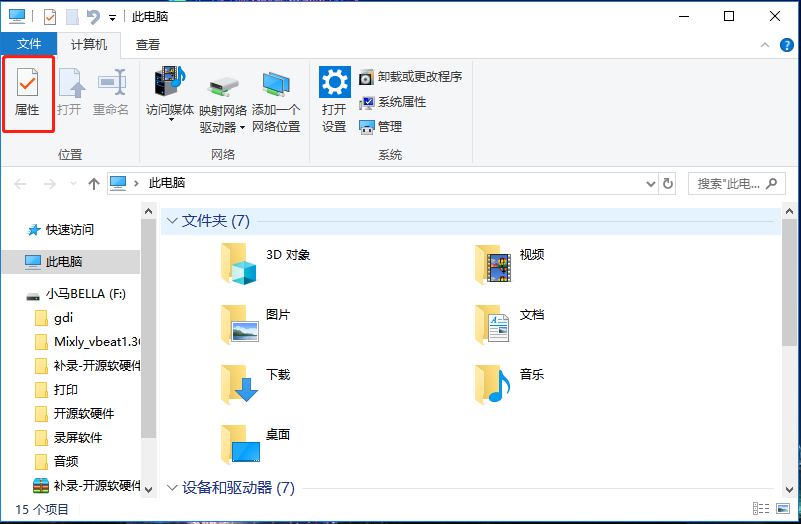
5

图1.1-6

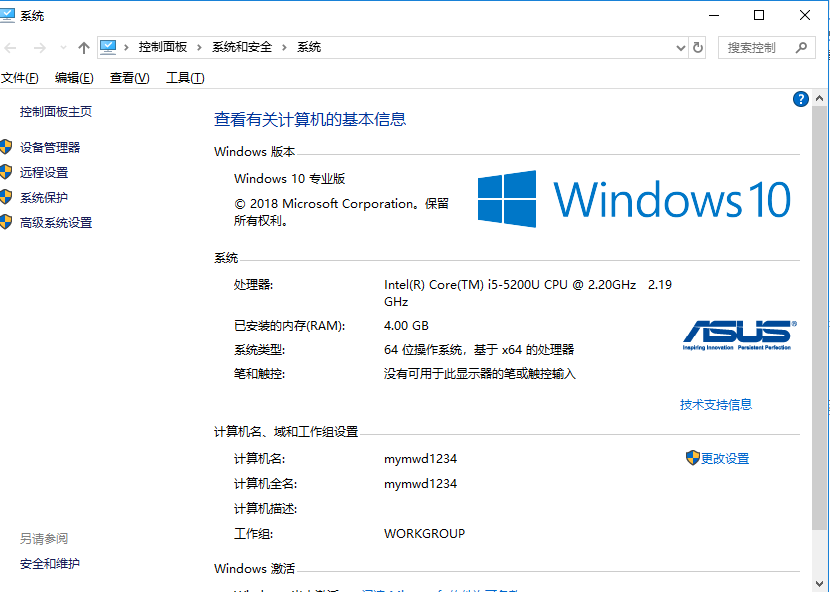


图1.1-7



图1.1-8



图1.1-9

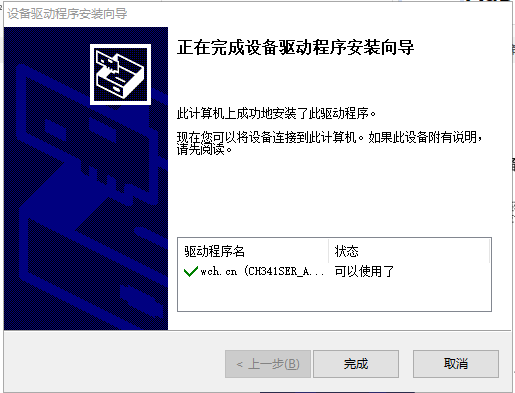


图1.1-10

3、若环境搭建完成，查看你的版型是否选对，甜橙应选第一个Arduino/Genuino Uno，如图1.1-11所示：

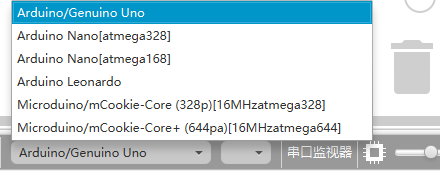


图1.1-11

4、检查板子的端口，Windows 笔记本电脑的Mixly软件在你将主控板成功连接到电脑时会自动识别端口，此时点开端口如图1.1-12所示：



图1.1-12

4、如果Mixly软件没有自动识别端口， 可首先打开端口观察插上USB线连接好之后那个端口新出现的，然后选择；

## 1.2认识Mixly

Mixly分为6个区域，分别是基本模块区、青橙模块区、程序构建区、基本功能区、显示信息区、源代码显示区域（平时缩在右边界，点击箭头会显示出来）。

1、Mixly在正常情况下，只能使用图形化编成区域来编写程序，源代码显示区域是没法修改的。当然，你可以点击图形化编程区域上端的“代码”进入到代码编写模式如图1.2-1所示。



图1.2-1

在代码编写模式下，你可以修改和编写代码，不过这里编写好代码后，图形化编程区域根本不会显示，所以除非你全部想使用代码来编写，不然还是建议不要直接修改代码。

2、Mixly在正常情况下，只能使用图形化编成区域来编写程序，源代码显示区域是没法修改的。当然，你可以点击图形化编程区域上端的“代码”进入到代码编写模式如图1.2-2所示。



图1.2-2

3、在代码编写模式下，你可以修改和编写代码，不过这里编写好代码后，图形化编程区域根本不会显示，所以除非你全部想使用代码来编写，不然还是建议不要直接修改代码。Mixly的软件的五部分，分别为基本模块区、青橙模块区、程序构建区、基本功能区、显示信息区如图1.2-3所示。



图 1.2-3

4、基本模块区就像画家的颜料盒，里面有我们需要的各种各样的程序块，包括输入输出、控制、数学、文本、数组、逻辑、串口、通信、传感器、执行器、显示器、变量、函数等。如图1.2-4所示：

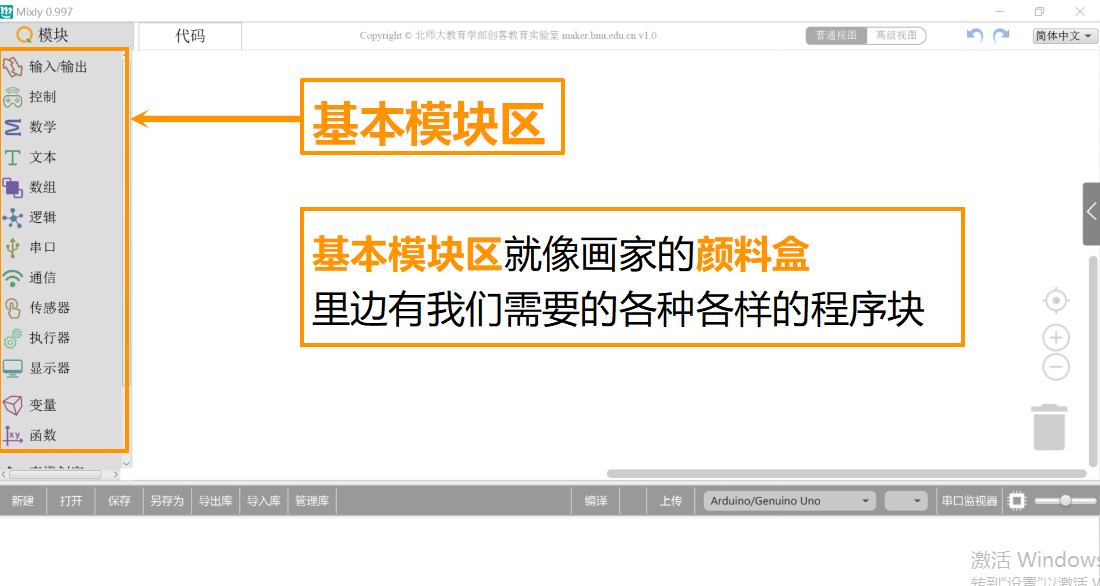


图 1.2-4

5、程序构建区就像画家的画布，将程序块在编辑区合理搭配，就能实现各种各样的功能。我们从模块区拖出的程序块就在这里被编辑、搭配，以实现各种各样的功能。

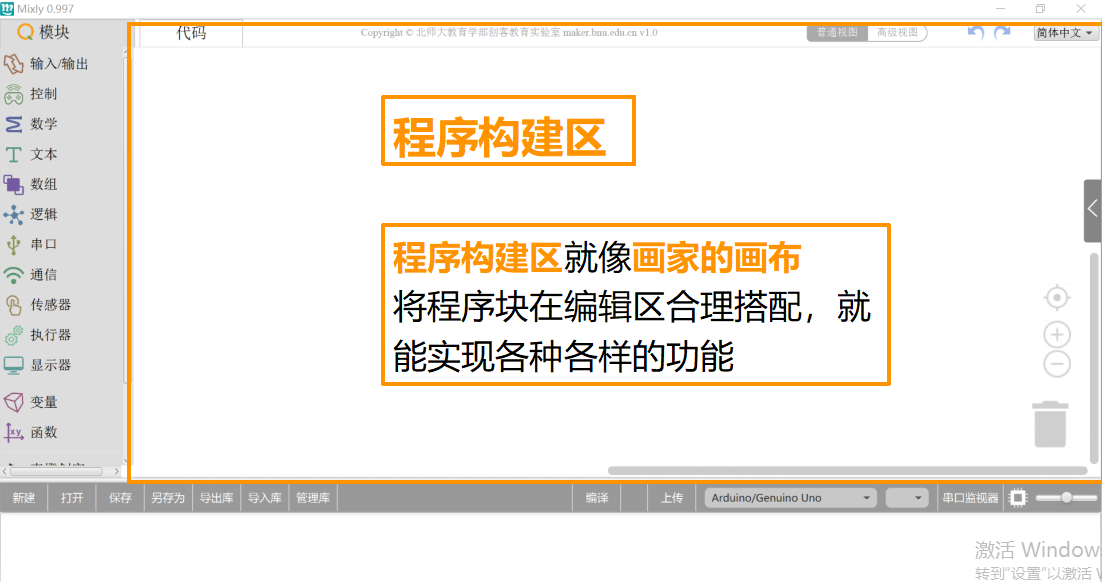


图 1.2-5

6、垃圾桶，将编辑区的程序块拖入垃圾桶就可以删除了，同时，将程序块拖到基本模块区和青橙模块区也可以删除掉了。

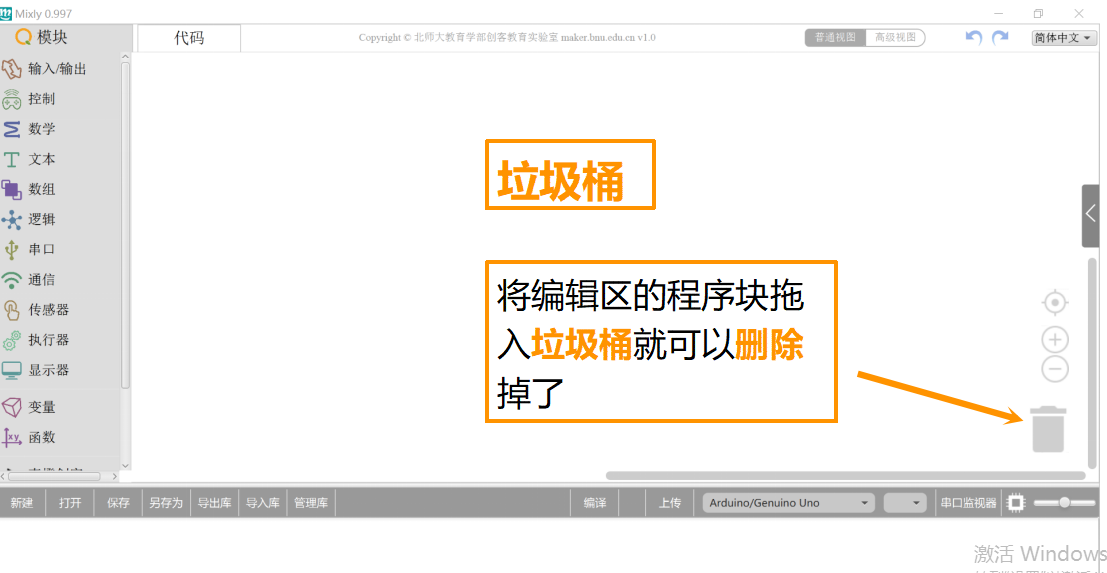


图 1.2-6

7、基本功能区，就像我们的管家，负责所有设计程序之外的事，如新建、打开、另存为、导出库、导入库、管理库、编译、上传、保存、串口监视、选择主板、选择串口等。

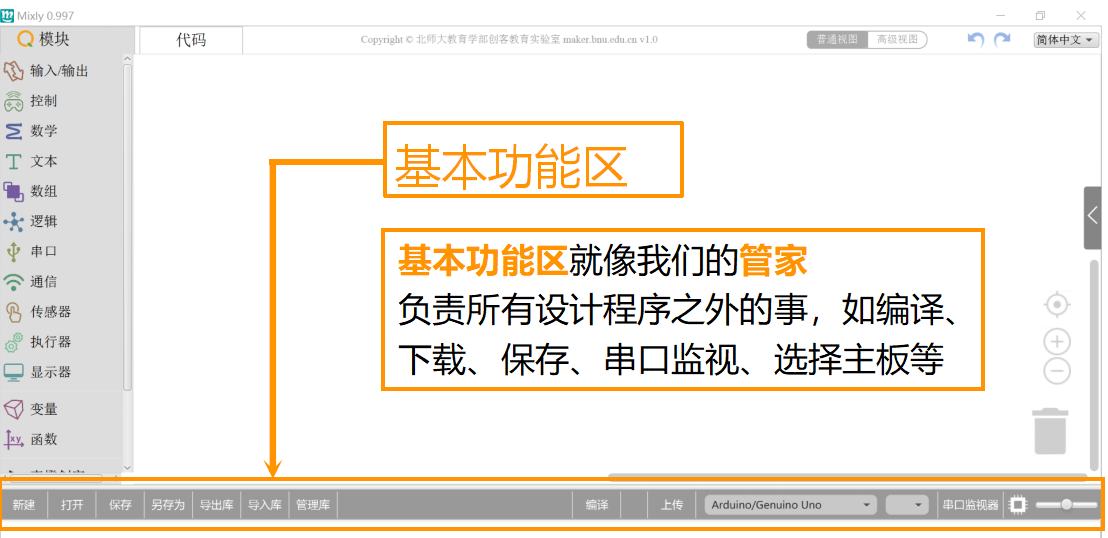


图 1.2-7

## 1.3测试软件能否正常工作

**主要分为三个步骤：**

1）连接硬件---能否识别自动端口；如上图1.3-1所示；



图1.3-1

2）修改控制芯片版型----Arduino/Genuino Uno（甜橙）；如图1.3-2所示：

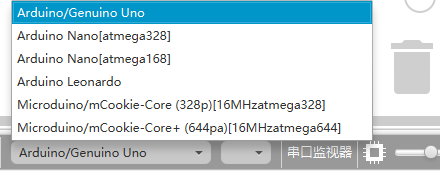


图1.3-2

3）连接硬件编写一个程序，如下图1.3-3，并上传，若上传成功，如图1.3-4所示，则证明软件已经安装好；

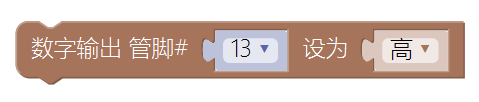


图1.3-3



图1.3-4

## 1.4任务：点亮一盏单色LED灯并闪烁

### 1.4.1硬件连接

将 LED 灯与 10 号管脚相连。注意插线时的颜色对应。

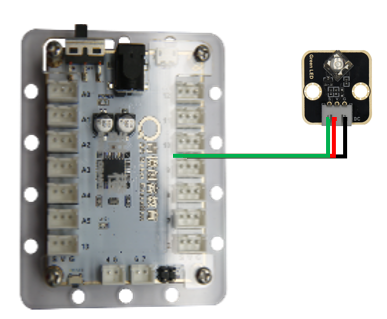


图1.4-1

### 1.4.2画流程图

这是一个顺序结构，可以看到：板载 LED 灯在熄灭 1 秒后又重新亮了起来，1 秒钟后又熄灭，并且，灯总是亮 1 秒，灭 1 秒，如此重复下去，这是因为，这些程序的模块是按它排列的顺序执行的，主控板不会先执行第一个模块，然后跳过延时的模块，直接去执行第三个模块，或者是按任何与程序不一样的顺序执行。

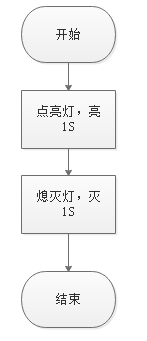


图1.4-2

### 1.4.3开始编程

（1）双击图1.4-3中的 Mixly.vbs 文件，即可打开 Mixly 软件：



图1.4-3

（2）在左侧“输入输出”模块菜单中找到“数字输出”模块，点击并拖动至右侧空白处，如图1.4-4所示。



图1.4-4

（3）在左侧“控制”菜单中找到“延时模块”，拖动至空白处并与“数字输出”模块拼接，修改数字输出的管脚为10，数字输出的输出电平设为一高一低。



图1.4-5

### 1.4.4程序下载

编写好程序后，单击下图中的上传按钮，将程序上传到 Arduino 主控板上。（注意，在上传程序之前，要设置好主控板型号和 COM 接口号，点击上传按钮右侧的下拉菜单即可设置）

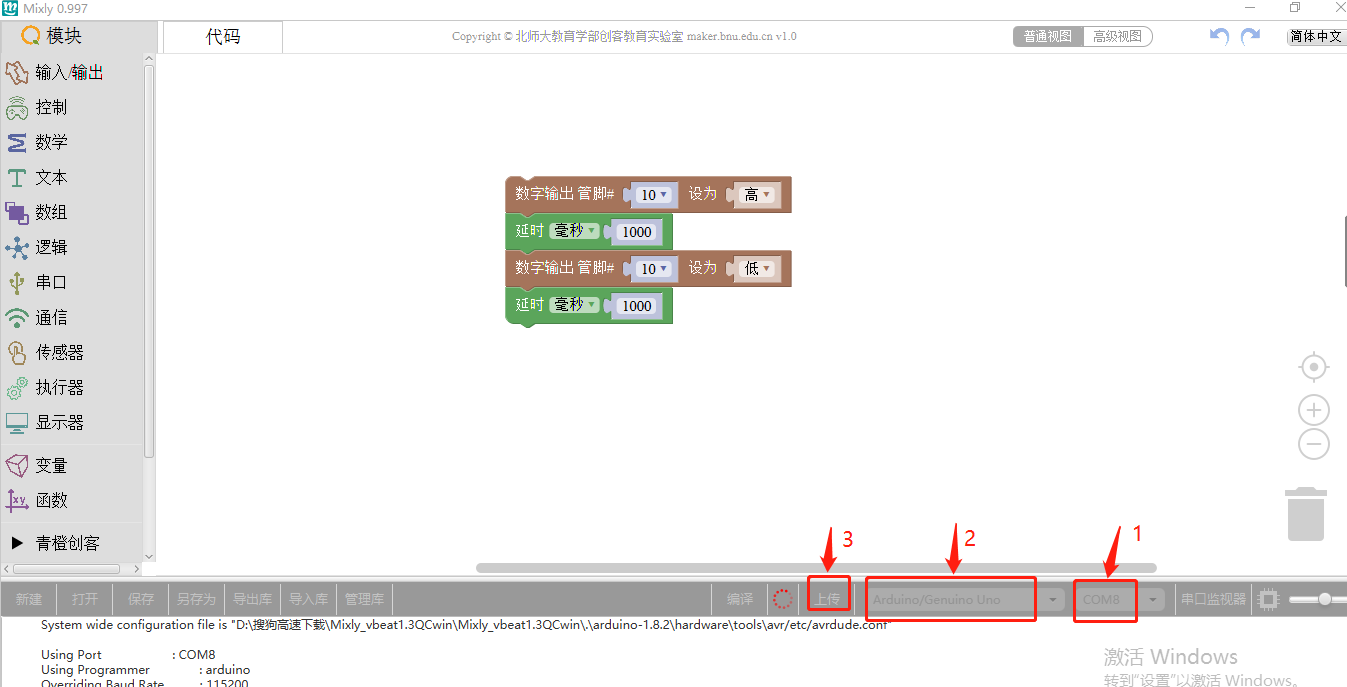


图1.4-6

上传之后，所有的按钮都变为灰色，无法点击，以保证上传过程不被干扰。

这里，我们见到了两个非常常用的模块——**数字输出和延时**。

数字输出是 Arduino 主控板对原件的控制方式之一。它向输出的电路传送数字信号——0和 1。0 意味着输出低电平，电路不会接通；1 则是输出指高电平，电路接通。上面的程序中，将 10 号管脚的数字输出设为高，与其连接的板载 LED 灯便会被点亮。经过 1 秒钟的延时（延时过程中，硬件保持延时开始时的状态，直到设定的时间结束），数字输出变为低，灯就会熄灭，之后保持熄灭状态 1 秒钟。

可以看到，板载 LED 灯在熄灭 1 秒后又重新亮了起来，1 秒钟后又熄灭，如此重复下去。

这是因为，Mixly 和 Arduino 默认这段程序是重复执行的。如果没有其他干预，程序便会一直重复执行。并且，灯总是亮 1 秒，灭 1 秒，这个重复不会发生变化。

## 1.5安装Arduino IDE

给Arduino写入程序需要使用Arduino ide软件，通过类C语言来编写，后来，随着Arduino的流行，这种直接编写代码的方式使零基础的初学者感到了困难，所以有爱好者还有一些教育机构开发了很多图形化的编程软件。Mixly是北京师范大学教育学部创客教育实验室开发的，是给Arduino写入程序的图形化编程软件，它适合刚接触Arduino和编程的入门学习者。

Mixly可以看作是介于普通用户与Arduino IDE之间桥梁，通过这个桥梁，即使用户不懂C语言的语法，也可以利用图形化程序编写Arduino程序。Mixly的基本原理是将图形化程序转化成C语言，再利用Arduino IDE上传到硬件中。

1）安装包解压后如图1.5-1所示，然后打开文件夹选择第一个文件夹，打开后如图1.5-2所示，打开文件夹后找到arduino 应用程序，如图1.5-3所示，双击，按照步骤安装完成；安装完成打开时，打开后的界面如图1.5-4所示。



图1.5-1



图1.5-2

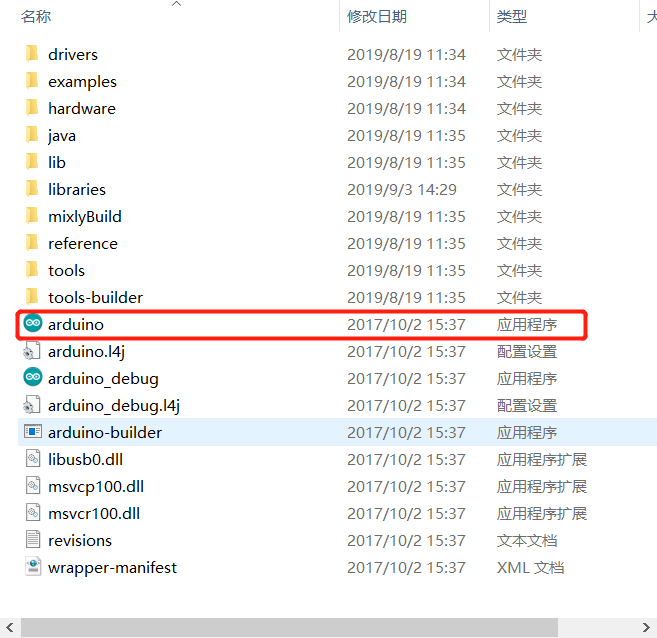


图1.5-3

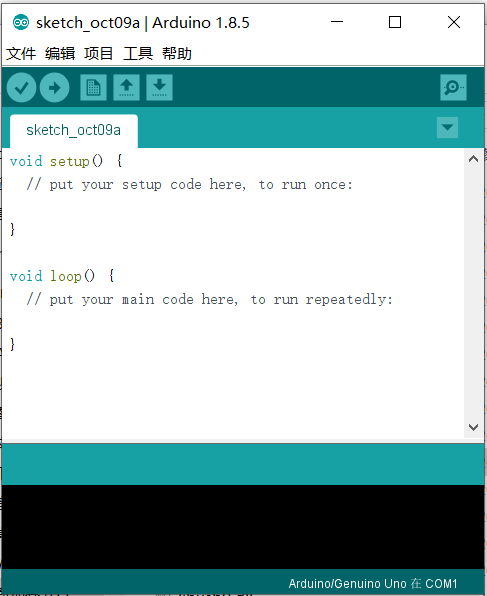


图1.5-4

注意，输入代码时，要切换到英文输入法的模式。下面黑色的区域是消息提示区，会显示编译或者下载是否通过。

2）工具栏介绍

从左到右依次是：验证、上传、新建、打开、保存。还有最右边的是串口监视器。

3）菜单栏介绍

1、文件：跟文件有关的操作，比如新建、保存、另存为、示例程序、最近文件等；其中示例程序是一些简单的、经典的程序；

2、编辑： 包括复制、还原、粘贴、查找、跳转等都是对当前文件的操作。

3、项目：是对当前文件的操作，包括编译也就是验证、上传、加载库文件等；

4、工具：常用的有自动格式化、串口监视器和串口绘图器、开发板、端口等；

自动格式化是对编写的代码的一些对齐和空格等问题的自动调整；

串口监视器也就相当于Mixly的串口监视器的文本格式；

串口绘图器也就是Mixly的串口监视器的绘图模式；

开发板也就是选择的开发版型号，包括Arduino UNO，Arduino nano，Arduino Mega等等；

端口就是当前arduino与电脑进行通信的接口，详情我们下面会提到。

5、帮助：帮助用户了解arduino的一些信息，比如运行环境、常见故障和arduino的版本信息等。

4）编程区介绍

Arduino 控制器的基本结构由setup（）和loop（）这两个函数组成。

Arduino控制器通电或复位后，机会执行setup（）函数中的程序，且只会执行一次，完成初始化设置，如配置I/O口状态和初始化串口等操作；对应Mixly图形化编程的“电机准备”、“波特率9600”、“彩灯准备”等内容。

Setup（）函数执行完毕后，接着执行loop（）函数中的程序，loop（）函数是一个死循环，其中的程序会不断地重复运行，完成程序的主要功能，如驱动各种模块和采集数据等。

5）信息显示区

直观表现编译、连接的情况，当程序上传成功后，会出现“上传成功！”字样。

3、如果 Mixly 软件没有自动识别端口， 可首先打开端口观察插上 USB 线连接好之后那个端口新出现的，然后选择它；

4、我们也可以在 ArduinoIDE 中选择主控板类型和 COM 口，步骤如下：

打开 ArduinoIDE，其编译环境如图 1.5-5所示

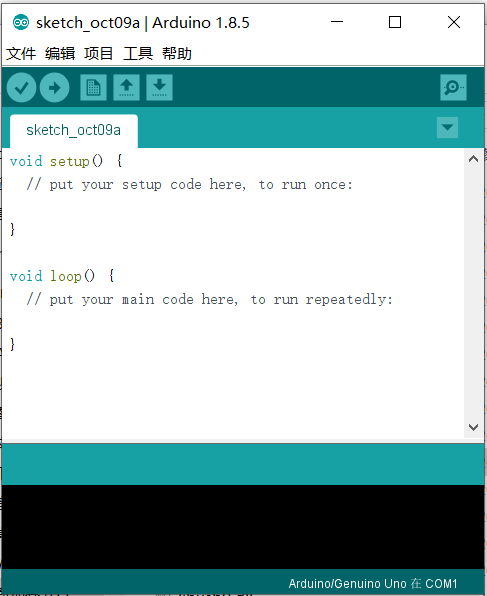


图 1.1-7

* + 1. 在 ArduinoIDE 中选取甜橙 2.0 主控板型号为 Arduino/Genuino Uno，如图 1.1-8。

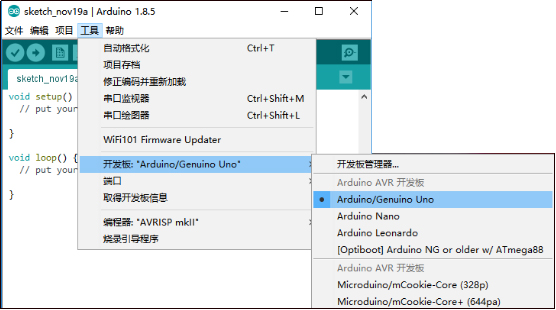


图 1.1-8

* + 1. 通过 MicroUSB 线将主控板连接到电脑的 USB 端口，在 ArduinoIDE 中选取与主控板对应的 COM 端口如图 1.1-9 所示，示例中主控板对应的端口为 COM6。

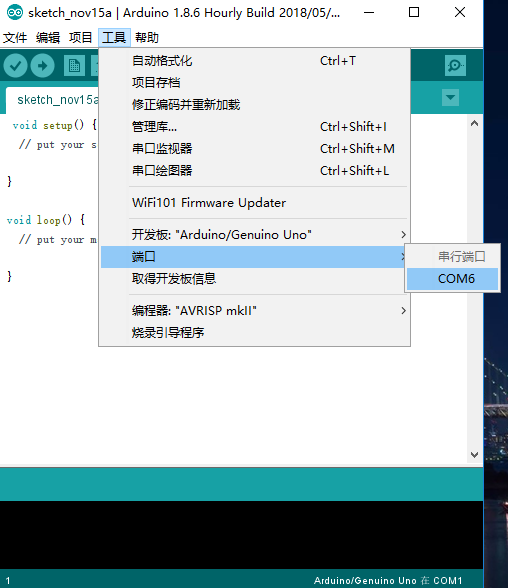


图 1.1-9

注意：连接同一块甜橙板时，arduino IDE和Mixly识别的是同一个端口，故我们不能同时使用两个软件上传使用Mixly上传时，必须保证Arduino IDE的软件没有正在使用该端口上传程序，同理，也不能在arduino IDE上传程序时在Mixly上上传程序。

# 第2章 Mixly系统介绍

## 2.1程序结构

Arduino 控制器的基本结构由setup（）和loop（）这两个函数组成。

Arduino控制器通电或复位后，机会执行setup（）函数中的程序，且只会执行一次，完成初始化相关设置，如配置I/O口状态和初始化串口等操作；对应Mixly图形化编程的“电机准备”、“波特率9600”、“彩灯准备”等内容。

Setup（）函数执行完毕后，接着执行loop（）函数中的程序，loop（）函数是一个死循环，其中的程序会不断地重复运行，完成程序的主要功能，如驱动各种模块和采集数据等。

## 2.2输入输出

打开 Mixly 编译环境，在 Mixly 左侧模块栏中选择 输入/输出，如图 2.2-1所示。

这部分功能包括：电平高（低）、数字输出、数字输入、模拟输入、模拟输出。

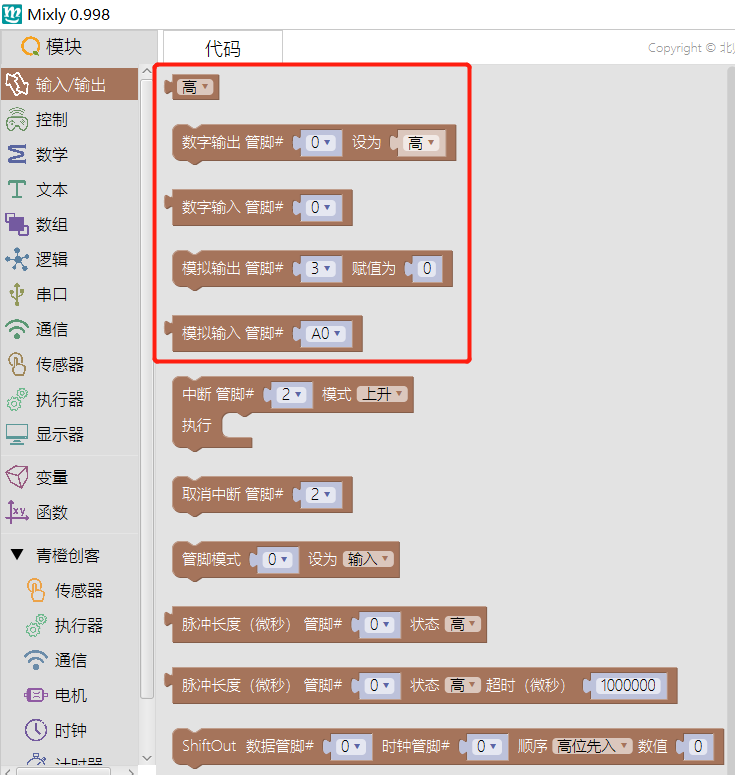


图2.2-1

### 2.2.1数字输入

数字输入表示：读取相应的管脚的电平高低，读取的信号是数字的。

1. 打开 Mixly 编译环境，在 Mixly 左侧模块栏中选择 输入/输出 →数字输入，如图 2.2-2所示。



图 2.2-2

2、修改数字输入管脚，根据主控板接入传感器的引脚选择正确的管脚号，如图2.2-3 所示修改为2号管脚。



图 2.2-3

### 2.2.2数字输出

数字输出表示：写入相应的管脚的电平高低，写入的信号是数字的。

在 Mixly 左侧模块栏中选择 输入/输出 →数字输出，如图 2.2-4。然后将管脚号改为 13，电平设为高，如图 2.2-5 所示。



图 2.2-4



图 2.2-5

### 2.2.3模拟输入

模拟输入表示：读取相应管脚的模拟信号，模拟信号转化为电压读入主控板，读取的信号是模拟的。

1、打开 Mixly 编译环境，在 Mixly 左侧模块栏中选择 输入/输出 →模拟输入，如图 2.2-6所示。



图2.2-6

2、修改模拟输入管脚，根据主控板接入传感器的引脚选择正确的管脚号，如图2.2-3 所示修改为A0管脚。



图2.2-7

注意：模拟输入管脚为A0-A7。

### 2.2.4模拟输出

模拟输出表示：写入相应的管脚的电压信号，电压信号转化为模拟信号来控制设备，写入管脚的信号是模拟的。

1、在 Mixly 的左侧模块栏中选择 输入/输出 →模拟输出，如图 2.2-8所示：



图2.2-8

2、修改模拟输出管脚，根据主控板接入传感器的引脚选择正确的管脚号，如图2.2-9所示：



图2.2-9

注意：模拟输出管脚在甜橙版上为3/5/6/9/10/11，赋值范围为0-255。

### 2.2.5中断函数

**中断**过程是指，在程序顺序执行的过程中，当主控板收到特定的数字输入信号时，会立即暂停当前执行的程序，转而去执行中断部分的程序，当中断部分执行完毕，再回到刚才暂停的地方继续执行原来的程序。

1、在 Mixly 的左侧模块栏中选择 输入/输出 →中断，如图 2.2-10所示，中断功能包括中断执行和取消中断两种，首先将中断执行程序块拖出右侧，如图2.2-11所示：



图2.2-10



图2.1-11

注意：1、甜橙主控板上只有 2 号和 3 号管脚支持外部中断功能。

1. 中断模式有三种，分别是：上升沿、下降沿、电平改变。
2. 中断程序块一般用在初始化程序块中。

2、在 Mixly 的左侧模块栏中选择 输入/输出 →取消中断，如图 2.2-12所示，取消中断可以解放被占用的管脚，对应管脚取消中断后可以用于其它功能。

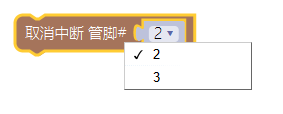


图2.1-12

### 2.2.6其它模块简介

如图2.2-13所示，从上到下依次是：管脚模式设置、脉冲长度（数字）设置、脉冲长度（数字）超时设置、shiftout移位函数。



图2.2-13

## 2.3控制

打开 Mixly 编译环境，在 Mixly 左侧模块栏中选择 控制，如图 2.3-1所示。

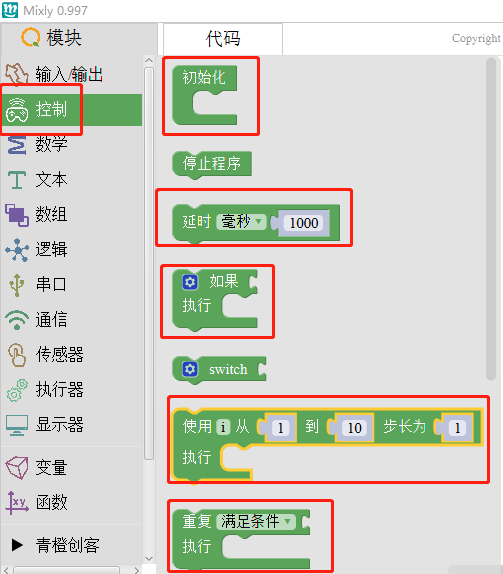


图2.3-1

### 2.3.1初始化

包括端口的状态、变量的声明、彩灯准备、电机准备、LCD162初始化等内容均属于初始化内容，内部的程序仅执行一次。



图2.3-2

### 2.3.2延时

延时程序可以让程序在原地踏步一定时间以后再执行下一步程序（延时的常用单位是毫秒），可编辑延时的数值大小。

注：1秒=10^3毫秒=10^6微秒。

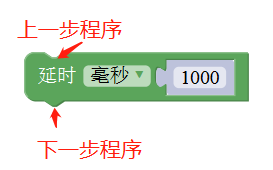


图2.3-3

### 2.3.3“如果”程序块

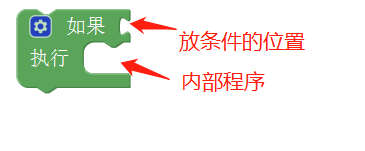


图2.3-4

若条件成立，将内部程序执行一次，然后执行下一步程序；

若条件不成立，跳过内部程序，执行下一步程序。

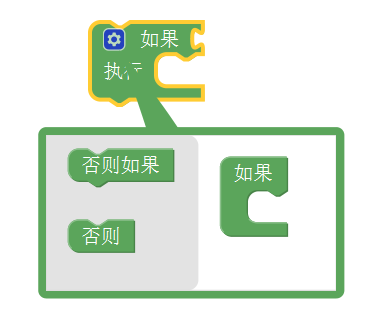


图2.3-5

“如果”语句分支结构，可以是单分支，也可以是多分支结构。

在“如果”模块中，点击“”图标，选择“否则”，将它拖到“内部程序”处，即可得到“如果否则”程序块。

当如果程序块需要判断多个条件时，就会用到“否则如果”标签，这个标签可以有无限个，程序进入程序块后，按从上到下的顺序判断各个标签的条件，如果条件符合，则执行该标签内部程序，如果条件不符合则判断下一个标签的条件，如果所有条件都不符合，则执行“否则”标签内部的程序，只要成功执行了任何一个内部程序，则跳出程序块。

### 2.3.4 switch语句

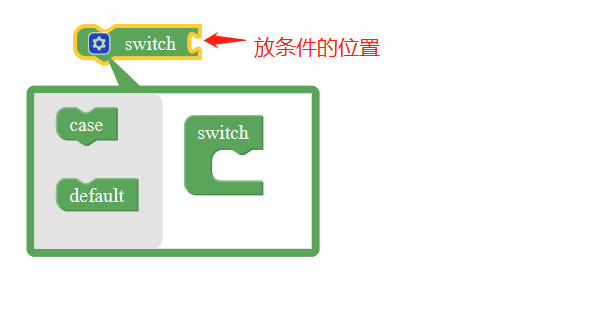


图2.3-6

Switch语句是多分支结构，与“如果”模块类似，switch语句常与case一起使用，default表示不满足条件时，执行语句体，default常用于switch语句的最后一层判断；相当于“如果”程序块的“否则”。

### 2.3.5 for语句

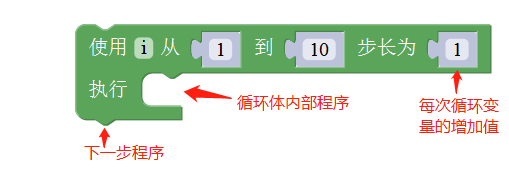


图2.3-7

循环的次数通过变量 i 的大小来控制。所谓变量，就是在程序运行过程中大小发生变化的量。循环执行过程中，变量 i 从 1 开始，每次增加 1，直到等于10，然后循环结束，执行下一步程序。每次循环，都会执行一次包在其中的程序。

for循环语句可以编辑变量的名称i，也可以改变变量的初值、终值和步长。

### 2.3.6重复执行

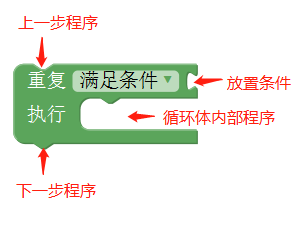


图2.3-8

重复执行，当满足某一条件时，执行内部程序。也叫“当型”while循环语句。可以改变循环模式，如图2.3-9所示，选择“不满足条件”。

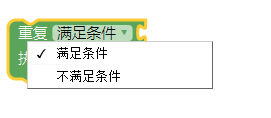


图2.3-9

### 2.2.7停止程序简介

停止循环程序，实际上这是一个死循环程序，让芯片反复的做无用的空指令，所以与延时程序块类似，使程序不向下执行，宏观上来看，就是程序停止了执行。



图2.2-10

### 2.2.8跳出循环简介

跳出循环分为两种，一种是“跳出循环”，函数为“break”，实际上是跳出循环结构；而另一种是“跳到下一个循环”，函数为“continue”，意思是跳出本次循环，却没有脱离循环结构。

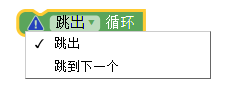


图2.2-11

### 2.2.9程序运行时间（计时器1）简介

该指令可以读取从代码编程开始到此时的系统时间。



图2.2-12

### 2.3.10定时器模块简介

人类最早使用的定时工具是**沙漏或水漏**，但在钟表诞生发展成熟之后，人们开始尝试使用这种全新的计时工具来改进定时器，达到准确控制时间的目的。控制模块中，还有定时器语句。MS2定时器利用系统内部的中断，优先级高于外部中断,在mixly中只有该定时器是内部中部。如图2.3-13所示。每次使用该定时器时需要先使用程序块“MsTimer2 启动”，使用完要使用程序块“MsTimer2 停止”，定时器的执行模块，可以调整间隔的ms数。

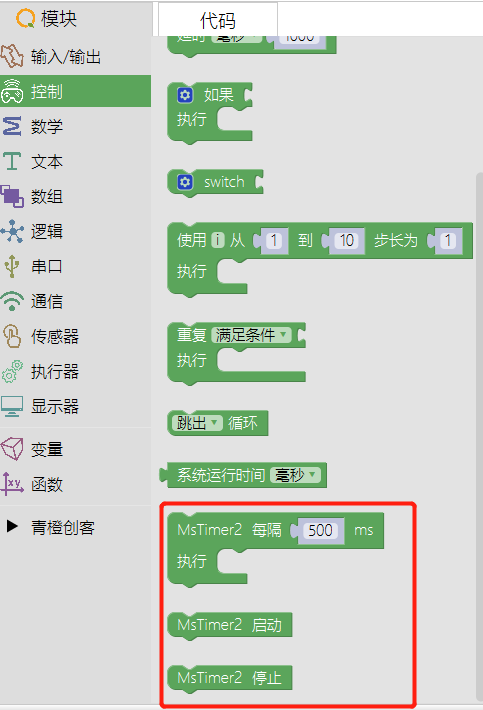


图2.3-13

## 2.4流程图表示程序

### 2.4.1基本流程图介绍

|  |  |  |
| --- | --- | --- |
| 名称 | 图片 | 用途 |
| 开始或结束 |  | 表示流程图的开始或结束 |
| 流程 |  | 执行或处理 |
| 判定 |  | 对某一条件进行判断真假 |
| 数据 |  | 表示数据的输入或结果的输出 |
| 数据流 |  | 箭头指向流程图执行的方向 |
| 连接符号 |  | 用于连接到另一页；  避免流程图交叉；  避免流程图太长 |

### 2.4.2顺序结构

功能描述：按照时间先后，依次执行语句A，语句B；是最为常见的一种结构，如图2.4-1所示。

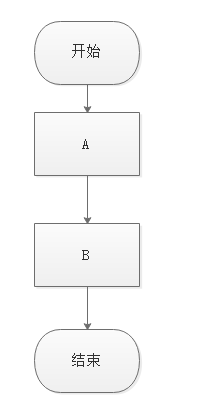


图2.4-1

顺序结构举例：点亮一盏灯，亮1S后熄灭，程序如图2.4-2所示，

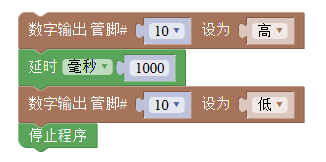


图2.4-2

此程序执行顺序为：从上到下，假如管脚10与单色LED灯相连，那么灯亮1S然后熄灭，程序结束。

### 2.4.3分支/选择结构

#### 1）单分支结构

程序执行至某一处时，出现分支，需要判断真假，如果条件成立，则为真，执行语句A，然后接着向下执行程序；如果不成立，则为假，执行语句B，随后程序向下执行。如图2.4-3所示：

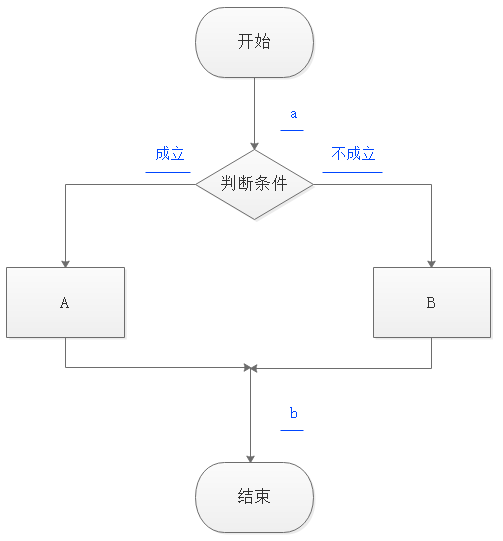


图2.4-3

分支结构举例：开关按下，单色灯点亮，松开后单色灯熄灭。程序如图2.4-4所示。



图2.4-4

#### 2）多分支结构

功能描述：如果表达式的值为真，则执行语句1，然后退出if选择语句不执行下面的语句；否则，判断表达式2，如果表达式2的结果为真，则执行语句2，然后退出if选择语句，不执行下面的语句，同样，如果表达式的值为假则判断表达式3，以此类推，最后如果表达式n不成立，则执行else后面的语句。如图2.4-5所示。

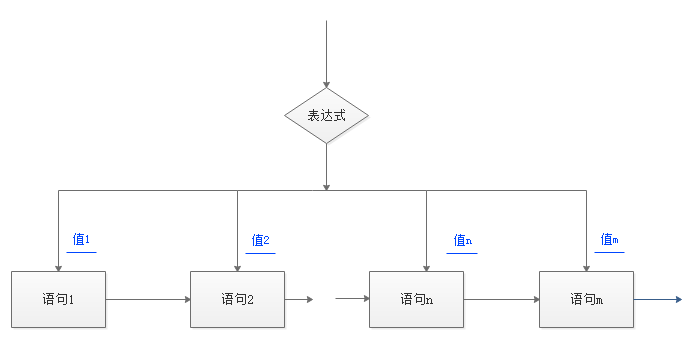


图2.4-5

注：一个判断框有两个出口，而分支结构只有一个出口，切记不要混为一谈。

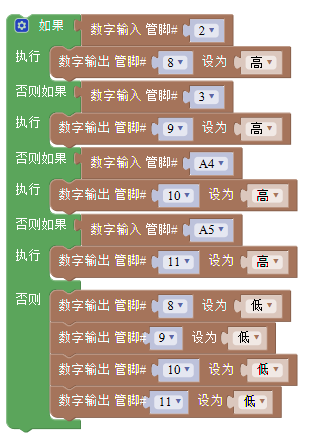


图2.4-6

如图2.4-6所示，为多分支结构，假设管脚2、3、A4、A5均与按钮开关相连，管脚8/9/10、11均与单色LED灯相连，那么此程序执行顺序为：

如果2号管脚相连的按钮按下，8号管脚相连的LED灯点亮；如果3号管脚相连的按钮按下，那么与9号管脚相连的LED灯点亮；如果A4号管脚相连的按钮按下，那么与10号管脚相连的LED灯点亮；如果A5号管脚相连的按钮按下，那么与11号管脚相连的LED灯点亮；如果没有按钮按下，那么所有的LED灯都熄灭。

### 2.4.4循环结构

首先判断表达式的真假，如果表达式为真，执行语句，再次判断表达式的真假，执行语句 ....... 直到表达式为假，此时跳出循环，执行程序下一步的程序。如图2.4-7所示。

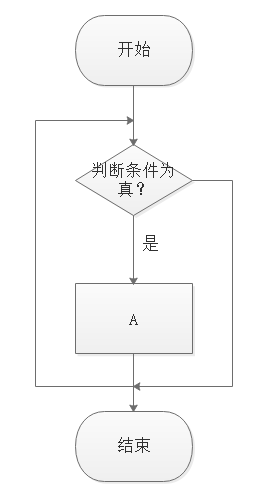


图2.4-7

举例：按下按键开关，单色灯点亮，松开后，单色灯熄灭。

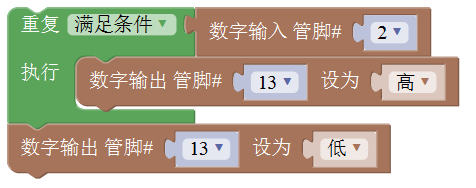


图2.4-8

假设管脚2相连的设备为按钮开关，13号管脚接的是单色LED灯，此程序的执行顺序为：

如果按钮按下，点亮LED灯，接着判断按钮的状态，如果还是按住状态，还是执行点亮LED灯....只有当按钮状态为松开时，熄灭LED灯。

#### 2）For语句

执行步骤：首先求解表达式1，第二步判断表达式2，如果表达式2为真，执行语句，然后求解表达式3，再来判断表达式2，如果表达式2为假，跳出循环，执行程序余下的语句。

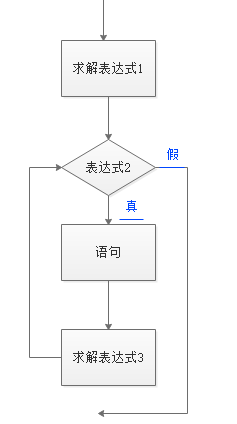


图2.4-10

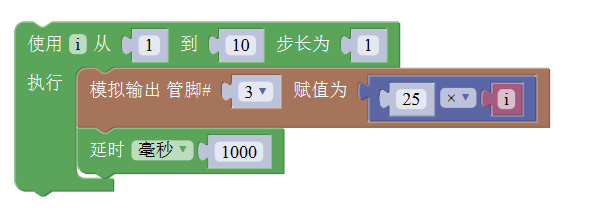


图2.4-11

假设管脚3连接单色灯，灯光由弱变强，每次变化灯光效果持续一秒，当灯光达到最亮后，持续一秒跳出循环，执行下一步程序。

### 2.4.5三种结构共同特点：

1）结构内每一部分都有机会被执行到；

2）结构内不存在无法跳出的循环；

3）只有一个入口，也只有一个出口。

## 2.5串口通信

甜橙的D0、D1为arduino自带的串口，称为硬件串口，D0为RX，为接收信号的端口；D1为TX，为发送信号的端口。在进行串口通信时，两个串口设备间需要发送端（TX）与接收端（RX）交叉相连，并共用电源地（GND），而使用SoftwareSerial类库模拟成的串口称为软件模拟串口（简称软串口）。



图2.5-1

### 2.5.1“打印（自动换行）”程序块

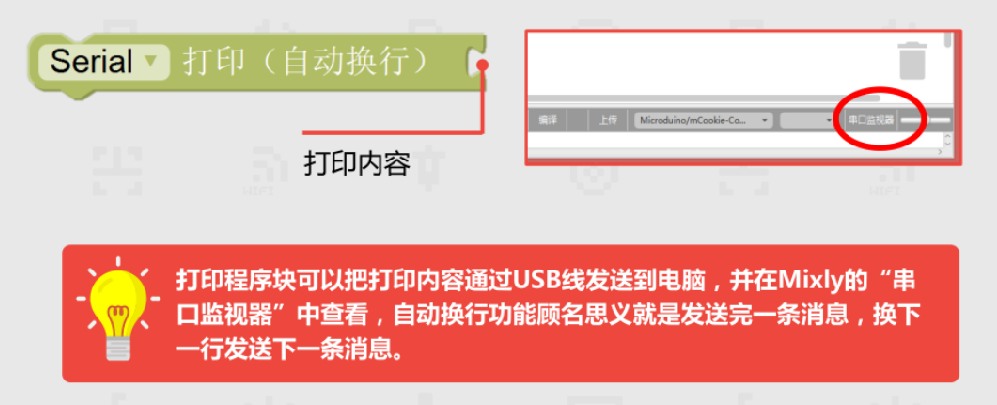


图2.5-2

### 2.5.2“软串口初始化”程序块

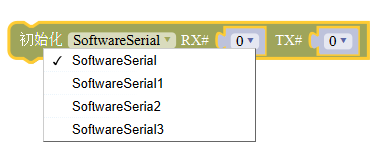


图2.5-3

注意：通常在连接外部设备时尽量避免使用0（RX）、1（TX）这组串口。

### 2.5.3其它模块简介

串口模块还包括：串口波特率设置、串口原始输出、串口打印、串口打印（十六进制）、串口有数据可读吗？、串口读取字符串、串口读取、串口清空缓存区数据、串口中断等。

串口波特率用于设置串口和软串口的波特率；

串口原始输出用于写入本串口的数据；

串口打印功能是打印出传入串口的数据，打印的数据在一行排列；

串口打印（十六进制/自动换行）功能是打印出传入串口的十六进制数据，并且自动换行；

## 2.6数学

数据的数学关系，包括加减乘除等确定的数学运算，还包括取随机数这样的得到数学数据。

### 2.6.1“计算”程序块

计算程序块主要用于加、减、乘、除、求余数、乘方的计算用途，选择运算符，将A数和B数进行需要的预算即可。

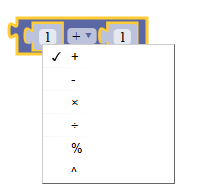


图2.6-1

### 2.6.2“转换”程序块

通常该函数被广泛应用于数论，函数绘图和计算机领域，这是对数据的处理，可分为六种方式，分别是：四舍五入、取上限、取下限、取绝对值、平方、平方根。

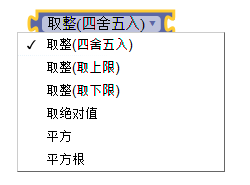


图2.6-2

### 2.6.3“随机数”程序块

随机数程序块，就好像自己做了一颗骰子，骰子最小可以摇摆到设置的下限，最高可以摇摆到设置的上限-1，例如以上程序可以在1-99之间取一个随机数。



图2.6-3

### 2.6.4“约束”程序块

“约束”就是将一个数字约束在一定范围之内，如果数字在范围之内，那么直接输出数字，如果数字低于下限，则输出下限值，如果数字高于上限，则输出上限值。



图2.6-4

### 2.6.5“映射”程序块

“映射”程序块可以自动将输入的数值从范围按比例换算到输出范围。

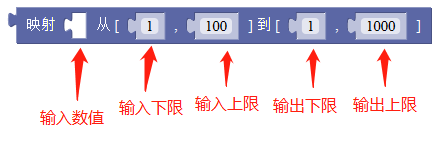


图2.6-5

映射的功能：是将前面集合中的元素对应到后面集合中的元素，若是模拟输入传感器，映射模块前面集合中的元素为主控制器从传感器采集到的数据（0～1023），后面集合中的数据设置为0～255，0～255 是模拟输出的限定范围；

### 2.6.6“位运算”程序块



图2.6-6

位运算：二进制数之间的运算

与：&，两边同为真，结果才为真；

或：|，两边任意一边为真，结果为真；

异或：一边为真，一边为假，结果才为真；

左移：<<，每左移一位，数值变为原来的一半；

右移：>>，每右移一位，数值变为原来的两倍；

### 2.6.7“函数运算”程序块

正弦、余弦、正切、反正弦、反余弦、反正切、对数函数、底数为e的对数函数、底数为e和10的幂函数、自增、自减、按位取反函数。

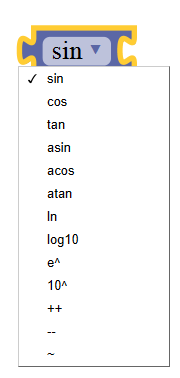


图2.6-7

### 2.6.8其它模块简介

数学模块里还包括数值、取最值、初始化随机数三种。“数值”模块可以取任意的数值； “取最值”可以取数据的最大值或最小值；“初始化随机数”模块可用来产生随机种子。单独使用random()函数所产生的随机数，在每一次程序重新启动后，总是重复同一组随机数字。如果希望程序重新启动后产生的随机数值与上一次程序运行时的随机数不同，则需要使用randomSeed()函数。

## 2.7逻辑

逻辑关系：比较、与、或、非、空、条件命题；



图2.7-1

### 2.7.1“比较”程序块

可以按照设定规则比较两边数字的大小，比较规则有：大于，小于，等于，如果两边数字的大小符合规则，则成立，如果不符合则不成立。

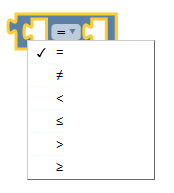


图2.7-2

### 2.7.2“逻辑选择”程序块

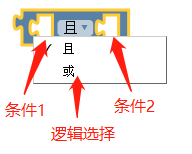


图2.7-3

### 2.7.3“逻辑非”程序块

用于将其连接的条件取反，如果条件成立，取反后条件不成立；如果条件不成立，取反后条件成立。

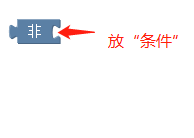


图2.7-4

### 2.7.4其它模块简介

“逻辑值”程序块，包括两个选项：真或假，也是布尔变量的逻辑值；“逻辑空”程序块，一般的，我们把研究对象统称为元素(element)，把一些元素组成的总体叫做集合(set)(简称为集)。对于两个集合A，B，如果A中任意一个元素都是B中的元素，我们就说这两个集合有包含关系，称集合A是集合B的子集(subset)（集合 A的空子集定义为不含A的元素的子集。任何集合的空子集相等，这时把空子集称为空集(empty set)，记为∅或{}. 一个至少含一个元素的子集叫做非空子集，而空集是非空子集的真子集。 注：1.空集中没有元素（即空集里元素的个数为0）；2.空集是集合论的一个重要概念，不包含任何元素的集合称为空集。

## 2.8文本

文本部分一来可以作为显示的内容，例如LCD1602显示器显示的内容、OLED显示器显示的内容还可以是串口打印的内容；二来，可以作为显示内容和显示器之间的桥梁，比如转换显示器显示的内容的类型，还可以获取显示内容的某一项内容。比较常用的功能分别是：字符串、字符、字符串连接和转换数值类型。如图2.8-1所示。



图2.8-1

### 2.8.1字符串程序块简介

通常作为显示器打印的内容，打印的内容类型为字符串，用“”括起来的数据类型。其中，可以对“hello”进行编辑。



图2.8-2

### 2.8.2字符程序块简介

通常作为显示器打印的内容，打印的内容类型为字符，用’ ’括起来的数据类型。其中，可以对 ’a’ 进行编辑。

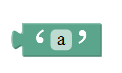


图2.8-3

### 2.8.3字符串连接程序块简介

通常作为显示器打印的内容，打印的内容类型为字符，用“”括起来的数据类型。其中，可以对“Hello”和“Mixly”进行编辑。



图2.8-4

### 2.8.4转换数值类型程序块简介

通常可以作为显示内容和显示器之间的桥梁，后面需要连接一项内容，类型可以为字符串。其中，既可以将显示的字符串转换为整数，也可以转换为小数。

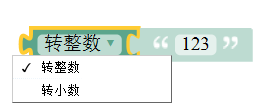


图2.8-5

### 2.8.5其它程序块简介

其它程序块从上到下依次有：

1. 数字转ASCII码：将十进制数转成ASCII码值；
2. 字符转ASCII码：将字符类型的数据转成ASCII值；
3. 进制数转字符串：可将二进制或八进制或十进制或十六进制数转成字符串的数据类型；
4. 获取字符串长度：后接一个确定的字符串，可获取其长度；
5. 获取某个字符：前面接确定的字符串，可获得第几个字符；
6. 如图２.８－６所示，字符串结尾于：用于测 试字符串是否以指定的后缀结束。如果参数表示的字符序列是此对象表示的字符序列的后缀，则返回 true；否则返回 false。

字符串开始于于用于检测字符串是否以指定的前缀开始，如果字符串以指定的前缀开始，则返回 true；否则返回 false。

字符串等于：注意，如果参数是空字符串，或者等于此 String 对象（用 equals(Object) 方法确定），则结果为 true。

1. 字符串比较：返回值是整型，它是先比较对应字符的大小（ASCII码顺序），如果第一个字符和参数的第一个字符不等，结束比较，返回他们之间的差值。如果第一个字符和参数的第一个字符相等，则以第二个字符和参数的第二个字符作比较，以此类推，直至比较的字符或被比较的字符有一方全比较完，这时就比较字符的长度。

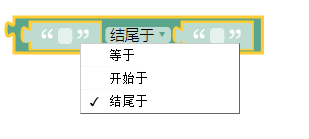


图2.8-6

## 2.9 数组

数组是为了便于代码编写时所采用的的一种数字块，具体包括：**定义数组、取数组值、改数组值。**

数组像一个大盒子，可以储存一定个数的数字（第一个指令）或字符串（第二个指令）。



图2.9－1

### 2.9.1 初始化数组

1）如图2.9-2所示：可以设置数组的元素的数据类型，可以设为整型、长整数、小数、字符字节、字符串等几种，也可以更改数组的名称，当前数组的名称为：mylist；可以储存一定个数的数。



图2.9－2

2）可修改数组的名称和组成的元素行数，可点击蓝色的小滚轮，添加或者删除行数，如图2.9-3所示。

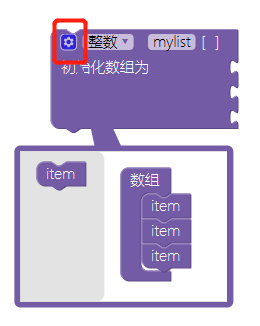


图2.9－2

3）如图2.9-3所示：可以设置数组的元素的数据类型，可以设为整型、长整数、小数、字符字节、字符串等几种, 也可以更改数组的名称，当前数组的名称为：mylist；储存一定个数的字符串。注意：双引号间应该写数字，并且每个用逗号隔开。

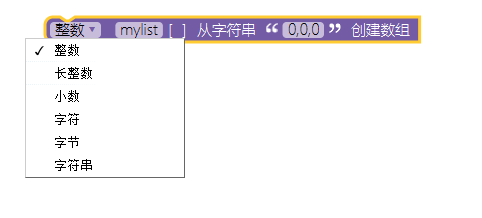


图2.9－3

### 2.9.2 获取数组长度

如图2.9-4所示，获取的是字符串mylist的长度。



图2.9－4

### 2.9.3 获取数组的某一项

1）如图2.9-5所示，获取的是字符串mylist的第1项。



图2.9－5

2）如图2.9-6所示，将字符串mylist的第一项赋值为某一数据类型，可以设为整型、长整数、小数、字符字节、字符串等几种。



图2.9－6

## 2.10 通信

通信是为了方便人与计算机交互而采用的一种特殊通信方式，具体包括：串口通信（新增串口选择和波特率设置）、红外通信、IIC通信、SPI通信。

在Mixly中，我们从基本模块区---通信中，可以看到：红外接收模块和红外发射模块，如图2.10-1所示，不过我们通常使用的是另一个模块，青橙创客---通信模块，因为包含了“红外接收恢复”模块。

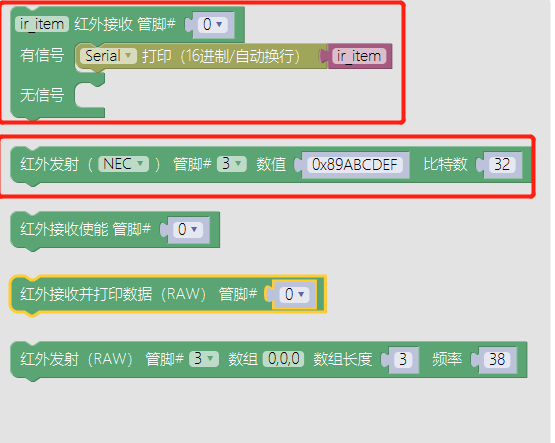


图2.10－1

### 2.10.1 红外接收

红外接收模块，可以修改管脚号，当有信号时，串口打印出十六进制并自动换行。常与“红外接收恢复模块”一起使用。

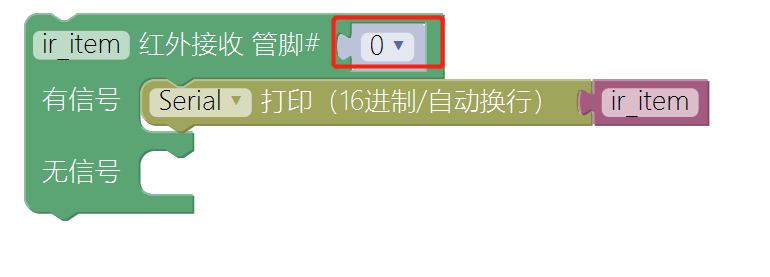


图2.10－2

### 2.10.2 红外发射

红外发射程序块对应开源硬件就是红外发射传感器，且默认与3号管脚相连；由于我们使用的硬件的型号，所以红外发射的类型必须为**NEC**，数值就是红外信号的类型，如图2.9-4所示，此处为十六进制信号。比特数表示数据传输的速率，为与红外接收模块配对，故**比特数**也是不变的。



图2.10－3

## 2.11执行器

执行器模块主要功能有：舵机、读取舵机角度、蜂鸣器播放和蜂鸣器结束声音。如图2.11-1所示。

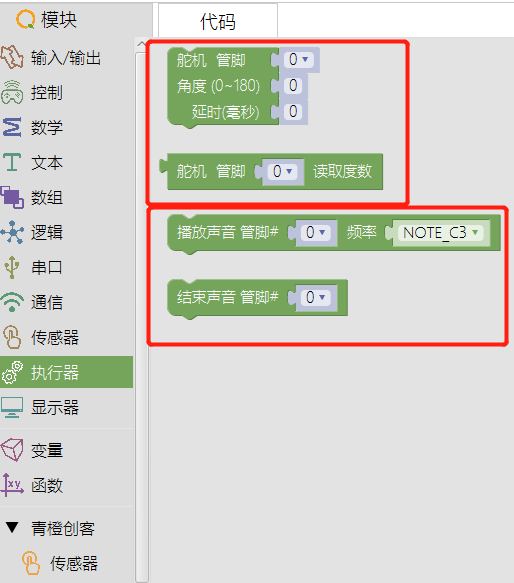


图2.11－1

### 2.11.1 舵机

在Mixly的基本模块区---执行器中，如图2.11-2所示，可以设置舵机的管脚号和转动的角度和转动角度需要的时间。

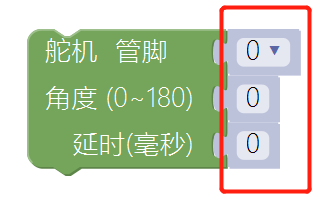


图2.11－2

如图2.11-3所示，可以读取舵机的角度数。

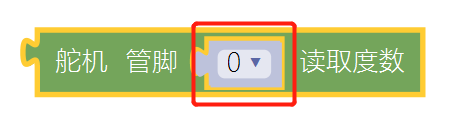


图2.11－3

### 2.11.2蜂鸣器

如图2.11-4所示，可以控制蜂鸣器发出不同频率的声音，也可以修改控制蜂鸣器的管脚。



图2.11－4

如图2.11-5所示，可以控制蜂鸣器结束发声。



图2.11－5

## 2.12显示器

显示器模块分为液晶显示屏1602、RGB灯、四位数码管、8\*8点阵显示屏的相关初始化、控制程序块。

### 2.12.1 LCD1602显示屏

如图2.12-1所示，LCD1602的地址是默认的0x27，设备名称为mylcd, 在Mixly中不能修改。



图2.12-1

如图2.12-2所示，LCD1602显示屏打印的内容可以是数字、字符、字符串等多种文本内容，显示格式可根据需要修改。



图2.12-2

如图2.12-3所示，LCD1602显示屏可以选择将打印的内容显示在想要的行和列的位置，LCD1602显示屏可打印2行，列数为16个字符。



图2.12-3

如图2.12-4所示，为液晶显示屏的清屏程序块，可以清除LCD1602显示屏上面显示的内容；也可以使用背光、光标闪烁等功能。



图2.12-4

### 2.12.2 RGB灯

如图2.12-5所示，RGB灯的准备程序块可以编辑管脚号和灯数。

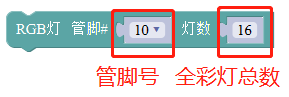


图2.12-5

如图2.12-6所示，管脚号的设置与上述步骤一致，灯号按照实际需要进行设置，示例中设置为 1，R、G、B 颜色设置的数值范围为 0～255，数值越大全彩 LED 灯的亮度越大。

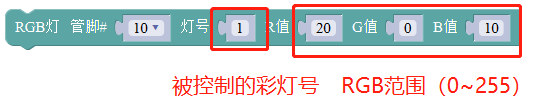


图2.12-6

如图2.12-7所示，设置全彩 LED 灯颜色的程序除图2.12-6 以外，还有另一种方法，在 Mixly 左侧模块栏的显示器选项中选择带有选色器的程序模块，颜色的选择可打开选色器，如图2.12-7所示。

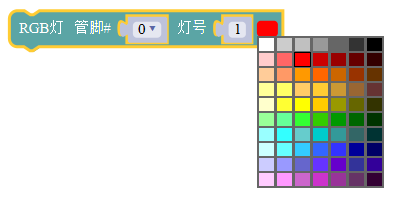


图2.12-7

### 2.12.3其它模块简介

如图2.12-8所示，四位数码管型号为TM1650，包括：清屏、显示字符串、点亮小数点三种；TM1637共阳极驱动数码管，包括：管脚初始化、设置亮度（0-9）、滚动显示字符串、显示时间等；8\*8点阵显示屏，包括：初始化、点阵屏显示、屏幕旋转、显示字符串、显示图案、显示清屏等；



图2.12-8

## 2.13变量

变量均需先声明再使用，且声明部分在程序中只执行一次；

### 2.13.1“变量声明”程序块

“变量”就像个记东西的小纸条，变量在使用之前要“声明”，就是告诉大家这张纸条叫什么（与其他变量区分），记录什么类型的东西（比如整数、小数或英文等），纸条上最初写着什么（初始值）。如图2.13-1所示，我们可以对变量进行声明，此时变量的名称为“item”，变量的数据类型为“整数”，变量初值默认为0。



图2.13-1

### 2.13.2“变量赋值”程序块

如图2.13-2所示，变量赋值的意思是赋予它一个数值，比如在纸条上写100，就是一次数值。



图2.13-2

### 2.13.3“变量值类型”程序块

如图2.13-3所示，可以将变量的数据类型进行强制转换，此时是转换为整数。还可以转换成长整数、小数、布尔、字节、字符、字符串等数据类型。



图2.13-3

## 2.14函数

函数与自然函数相同，需要设置参数，完成特定的函数功能。

### 2.14.1“函数准备”程序块

函数好像一个程序包，其中包含了一些制定好的程序，在执行函数的时候，其实是执行了函数内部制定好的程序，同样的函数执行多少次结果都一样，就好像是命令厨师蒸馒头的程序，每次蒸馒头的步骤肯定一样，蒸出的馒头肯定也一个样子。

如图2.14-1所示，函数的名称为“procedure”,函数内容为空。



图2.14-1

### 2.14.2“参数”程序块

如果命令厨师蒸包子，那就需要告诉厨师包子馅是什么（X），其他的程序还是一样的（和面、包包子、蒸熟），这时候单纯的函数就无法胜任了，需要在执行函数的同时，将参数X（包子馅）设定好。

如图2.14-2所示，添加参数的步骤如下：从左侧模块区拖出函数准备程序块，点击蓝色齿轮，将参数修改为X，将参数移动到输入中，就完成了参数的添加，最终如图2.14-3所示。



图2.14-2

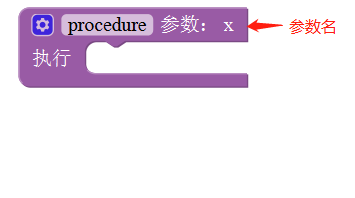


图2.14-3

### 2.14.3“函数执行”程序块

“函数执行”程序块在编程界面中就像是普通的程序块，只不过在实际执行的时候执行的是“函数准备”中制定的程序，并且需要设定好参数，参数可以是数字也可以是文本，参数实际上就是变量，只是因为位置特殊，所以叫“参数”。

如图2.14-4所示，为“函数执行”程序块，此时参数为X，可以编辑传入的参数的值。



图2.14-4

# 第3章 青橙创客模块区

## 3.1传感器

人们为了从外界获取信息，必须借助于感觉器官。而单靠人们自身的感觉器官，在研究[自然现象](https://baike.baidu.com/item/%E8%87%AA%E7%84%B6%E7%8E%B0%E8%B1%A1)和规律以及生产活动中它们的功能就远远不够了。为适应这种情况，就需要传感器。因此可以说，传感器是人类五官的延长，又称之为电五官。

传感器的存在和发展，让物体有了触觉、味觉和嗅觉等感官，让物体慢慢变得活了起来。通常根据其基本感知功能分为热敏元件、光敏元件、气敏元件、力敏元件、磁敏元件、湿敏元件、声敏元件、放射线敏感元件、色敏元件和味敏元件等十大类。

传感器的用法与原理详情请见《硬件说明书3.0》。

### 3.1.1数字开关类传感器

**数字开关类传感器**，是以高低电平为输出信号的一类传感器，向主控制器输出信号“1”代表高电平，输出信号“0”代表低电平，而且只有这两种状态。

如图3.1-1所示，常见的数字开关类传感器有：按键开关、碰撞开关、触摸开关、倾斜开关、拨动电平开关、红外巡线传感器、红外接近开关、磁力开关、火焰传感器、人体感应传感器等。



图3.1-1

### 3.1.2模拟型传感器

模拟量在时间上或数值上都是连续的；我们把表示模拟量的信号叫**模拟信号**。例如：生活中常见的声音，声音的大小都是连续的，这样的量就叫模拟量，我们能听到声音，是因为有声音信号通过空气或者其他的介质（也叫媒介）传给我们的耳朵，我们才能听到声音，这种声音信号就是模拟信号。

如图3.1-2所示，常见的模拟量传感器有：模拟声音传感器、模拟光线传感器、旋钮开关、烟雾传感器、土壤湿度传感器、水分传感器、酒精传感器、水蒸气传感器等。



图3.1-2

### 3.1.3其他传感器

如图3.1-3所示，综合类传感器有：摇杆、DS18B20水温传感器、BMP180气压传感器、MPU6050三轴加速度传感器、超声波传感器和DHT11温湿度传感器。



图3.1-3

摇杆模块可分解为两个模拟输出和一个数字输出；

DS18B20水温传感器模块可改变控制管脚，测量出的水温的单位为摄氏度；

气压传感器通过IIC与主控板通信，分为初始化和数据获取两个程序块，初始化包括传感器控制管脚、串口波特率等一系列设置，数据获取模块可获得气压、温度、海拔高度三项数据；

三轴加速度传感器通过IIC与主控板通信，也包括初始化和数据获取两部分，初始化就是加速度角速度收集模块，里面设置了管脚、波特率等内容，数据获取可以获得模块自身三个方向的加速度和角速度等数据。

超声波测距模块，可以修改发射和接收的管脚，Trig为发射，Echo为接收管脚；

DHT11温湿度传感器，其中，DHT11为传感器的型号，可修改控制的管脚，也可以获取环境的温度和湿度值。

## 3.2执行器

通俗来说，执行器就是通过控制器发出的信号来完成某项任务的执行装置，比如，想要点亮LED灯，需要甜橙主控板发出信号，来点亮LED灯，其中LED灯就是执行器；再比如，想要播放音乐，必须要有蜂鸣器来发出声音，蜂鸣器就是执行器；此外，常见的执行器还有振动马达、全彩灯、铜线灯串（与转接口一起使用）、减速电机、舵机还有模块化风扇。

### 3.2.1声类

声音类执行器有：蜂鸣器和带功放喇叭两种，喇叭没有专门的模块来控制，蜂鸣器有。如图3.2-1所示，蜂鸣器有三种控制方式，分别是频率（0-2000）、旋律和歌曲，频率控制声音的频率范围为0-2000；旋律就是音乐的七个音阶，如图3.2-2所示；歌曲是提前封装好的一些旋律，另外还有“蜂鸣器关”程序块，可以关闭对应管脚链接的蜂鸣器。



图3.2-1



图3.2-2

### 3.2.2光类

常见的光类执行器有：单色灯、LCD1602显示屏、OLED屏幕、灯串、全彩灯等。

1. 单色灯

如图3.2-2所示，包括单色灯、灯串、逐渐点亮和逐渐熄灭的呼吸灯四种。单色灯是数字输出；灯串是模拟输出，可连接3、5、6、9、10、11等6个管脚，灯的亮度范围为0-255；呼吸灯的程序块也是模拟输出控制，控制管脚可以是3、5、6、9、10、11等，时间的单位为秒；



图3.2-2

1. 全彩灯

如图3.2-3所示，彩灯的程序块分为彩灯准备、彩灯颜色控制、彩灯呼吸。

彩灯准备程序块可以为设备分配地址，可以编辑彩灯的控制管脚和彩灯总数，这里的彩灯默认是串联；图中彩灯总数为4。

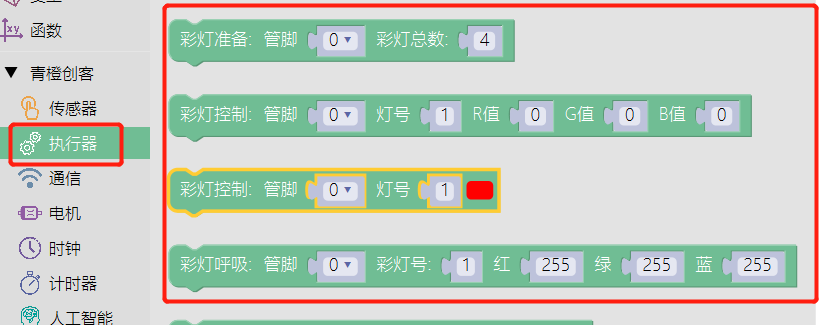


图3.2-3

彩灯控制程序块，可以编辑彩灯的控制管脚和彩灯号，控制方式有两种，一种是利用RGB值来调颜色，颜色的数值范围为0-255；另一种是从调色盘里选择需要的颜色，如图3.2-4所示；



图3.2-4

彩灯呼吸程序块，可以编辑彩灯的控制管脚和彩灯号，也是利用RGB值来调颜色，进而逐渐点亮和熄灭。颜色的数值范围同样为0-255。如图3.2-5所示。



图3.2-5

1. OLED屏幕

如图3.2-6所示，OLED屏幕相关的程序块包括：

OLED初始化，包括库文件的导入，管脚的设置，变量的设置等，这些都是必须的；

OLED打印 程序块，不支持汉字打印，默认从第一行第一列开始打印字符串；

打印程序块，可以选择打印的位置，OLED的规格是128\*64，每个字符占据的大小不一，每个字符高约8，y的范围是0-32。

OLED显示屏清屏　程序块，可以清除OLED屏幕打印的所有内容。



图3.2-6

### 3.2.3动类

　包括：电机、舵机、水泵、模块化风扇、振动马达和电磁铁等。

　如图３.２－７所示，模块化风扇为数字输出控制执行器，振动马达为模拟输出类执行器，可以编辑振动的速度数据，范围为０－２５５。



图３.２－７

　如图３.２－８所示， 为舵机控制的程序块，可以编辑控制的管脚号，转动的角度与转速，与２.１１章舵机的控制不同的是这里控制的是电机的转动速度，上面控制的是舵机的转动时间。

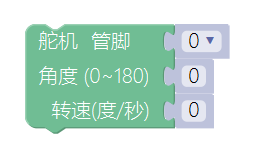


图３.２－８

　如图３.２－９所示， 当驱动的舵机较多时，为了节省主控板的管脚并对舵机更高效的控制，可以使用舵机驱动，当使用舵机驱动来控制舵机时，可以使用如图３.２－９所示的程序块，舵机驱动准备初始化了舵机的管脚、通信方式等；舵机驱动可以修改舵机的通道数（０－７）、转动的角度（０－１８０）和延时时间（单位：ｍｓ）。

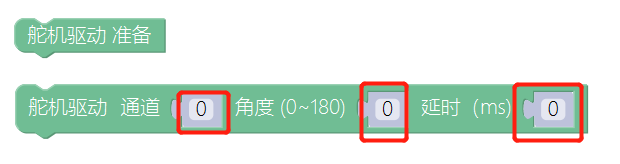


图３.２－９

　如图３.２－１０所示， 电机程序块分为电机初始化和电机选择。电机初始化定义了管脚等内容，电机选择　可以控制两路电机，分别是M１和M２，M１对应管脚４和５，M２对应管脚６和７，电机速度范围为－２５５到２５５。

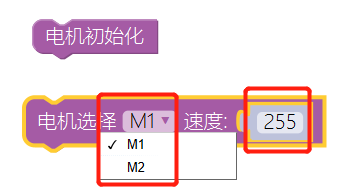


图３.２－１０

　如图3.2-11所示，电磁铁与水泵的控制与电机的控制语句相似，首先都是设备准备，然后选择控制端口和大小。吸力大小范围都是0-255。



图３.２－１1

## 3.3通信

通信是为了方便人与计算机交互而采用的一种特殊通信方式，具体包括：串口通信（新增串口选择和波特率设置）、红外通信、IIC通信、SPI通信。

### 3.3.1蓝牙

本节通过设置软串口来进行蓝牙通信。

1）“蓝牙准备”程序块

该程序块写入了关于蓝牙通信的波特率、端口信息还定义了变量，我们可以自己设置蓝牙接收信息与发送信息的端口号，是蓝牙通信的必要准备。



图3.3-1

2）“蓝牙接收”程序块

该程序块是一个判断条件，表示：如果接收到蓝牙信号？执行倒U型里的语句体；



图3.3-2

3）“蓝牙接收的数据”程序块

表示传入蓝牙的信息，它的数据类型为字符型。

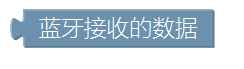


图3.3-3

4）“蓝牙发送数据”程序块

表示蓝牙传入其它设备的信息，它的数据类型为字符串。



图3.3-4

5）“读取串口数据”程序块

对应

读取其它设备对应的串口传入的信息。



图3.3-5

### 3.3.2红外通信

1）“红外发射”程序块

NEC为红外发射器的型号，数值是发射出的红外信号，既可以为十六进制，以0X开头，也可以发射十进制、八进制、二进制数，比特数32，为数据的传输速率。

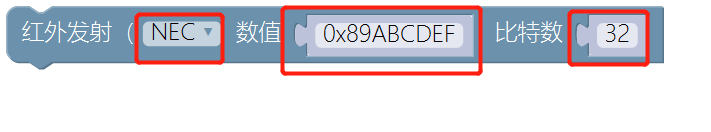


图3.3-6

2）“红外接收”程序块

该模块定义了隐性变量ir\_item来存储接收到的红外信号，可以选择红外接收的控制管脚，当有信号是，执行Serial 打印语句。



图3.3-7

3）“红外接收恢复”程序块

红外接收传感器每次接收到红外信号后，需要释放内存，才能为下一次接收信号做好准备。控制管脚与图3.3-7所示的管脚对应，并且该模块经常放入图3.3-7所示的有信号后执行的语句体的最末尾。



图3.3-8

### 3.3.3PS2手柄

1）“初始化PS2”

可以选择主控板的版型，UNO或者Mega2560，初始化PS2定义了控制的管脚、导入库和定义变量等信息。

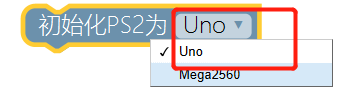


图3.3-9

2）“按键”

手柄的所有按键都可以选择，按键的状态也可以选择，当按键状态为“按下”时，表示对应按键按下时才会触发。

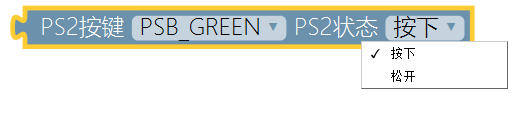


图3.3-10

3）“摇杆”模块

两边摇杆分别是R和L，也就是左和右，每边摇杆可以控制X、Y两个方向的运动，该模块可以选择控制四个变量的任一个。

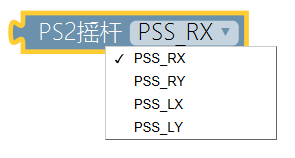


图3.3-11

## 3.4时钟

时钟模块主要指DS1307，它有两个最主要的功能，一个是：设置指令的时间与日期；另一个是：显示北京时间当前的时间与日期。如图3.4-1所示



图3.4-1

### 3.4.1设置时间

1）“初始化时钟模块”

如图3.4-2所示，该程序块表示与主控板通过IIC进行通信，“myRTC”为时钟模块的名称，不可更改，SDA为数据传输端，默认与主控板A4连接，SCL为时钟控制端，默认与主控板A5连接。

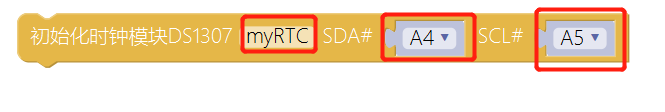


图3.4-2

2）“设置日期”

如图3.4-3所示，可以设置RTC的日期，有关年月日的信息，日期符合一年12个月，每月的天数也与公历纪年相同。



图3.4-3

3）“设置时间”

如图3.4-4所示，可以设置RTC的时间，有关时分秒的信息，时（0-24）分（0-59）秒（0-59）。

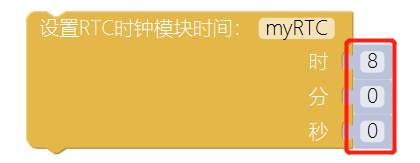


图3.4-4

4）“获取时间与日期”

如图3.4-5所示，该模块可以从RTC模块获取设置好的时间信息，包括年、月

日、时、分、秒等6种时间信息。



图3.4-5

### 3.4.2显示时间

1）“初始化DS1307”

与设置时间不同的初始化，主要区别在于程序调用的库不同，使用的变量等也不相同。如图3.4-6所示，该程序块是获取北京时间的初始化模块。



图3.4-6

2）“获取北京时间”

如图3.4-7所示，该模块可以从RTC模块获取标准北京时间信息，包括年、月

日、时、分、秒等6种时间信息。



图3.4-7

## 3.5计时器

Timer1计时器模块是利用软件来计时，并不涉及中断，所以与MsTimer2 定时器工作原理不同。该模块与其它模块一样从上到下执行。如图3.5-1所示，计时器共有四个程序块，分别是：间隔执行、计时器开始、超时、持续时间。

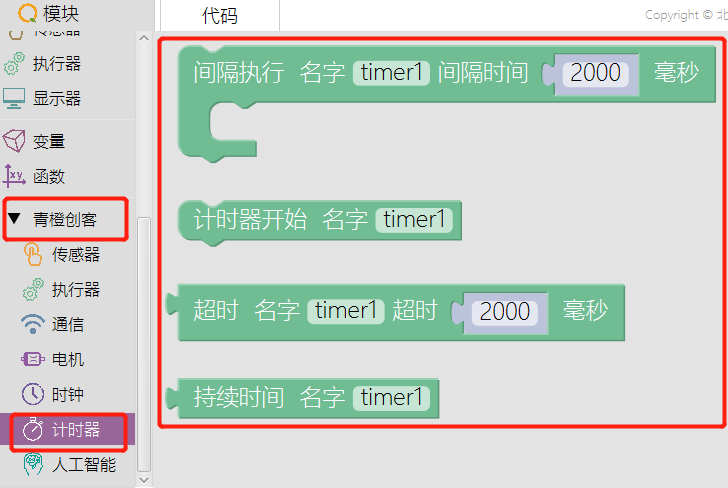


图3.5-1

### 3.5.1“间隔执行”程序块

如图3.5-2所示，该程序块定义了计时器的名称为timer1，间隔时间为2000毫秒，可以修改。每隔2000毫秒执行语句体1，执行完语句体1后执行语句体2。

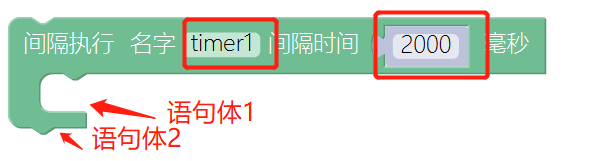


图3.5-2

### 3.5.2“计时开始”程序块

如图3.5-3所示，该指令可以从程序运行的任意时刻开始计时。



图3.5-3

### 3.5.3“超时”程序块

如图3.5-4所示，该指令可在到达指令的时间后，再过2S触发。



图3.5-4

### 3.5.4“持续时间”程序块

如图3.5-5所示，该模块常与“计时开始”程序块一起使用，表示从计时开始模块后程序运行到此模块共持续了多长时间。



图3.5-5

## 3.6人工智能

人工智能（Artificial Intelligence），英文缩写为AI。它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。人工智能是计算机科学的一个分支，它企图了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器，该领域的研究包括机器人、语言识别、图像识别、自然语言处理和专家系统等。本节共涵盖了三个模块，分别是：语音识别、OpenMV模块和MU魔图识别模块；语音识别属于人工智能的语言识别方向，OpenMV模块和MU魔图识别模块属于图像识别方向。

### 3.6.1语音识别

语音模块的设置仅指主控板端的信号处理的程序，语音模块本身也有工作模式与工作内容。例如：语音模块的对话内容、工作模式（共四种）等。如图3.6-1所示，语音模块相关的程序块共三条。



图3.6-1

语音指令设置，包含调用库、定义变量等内容；

语音管脚设置，可设置软串口通信的管脚和波特率，默认均是9600。RX为串口通信的接收管脚，TX为串口通信的发送管脚。注意：一般不与硬件串口相同。

读取语音数据，指的是语音模块发送到主控板的数据。

### 3.6.2 OpenMV模块

OpenMV模块的设置仅指主控板端的信号处理的程序，OpenMV模块本身也是一种单片机，这里的程序块是当与甜橙主控板通信的时候，OpenMV模块这时相当与图像传感如图3.6-2所示，OpenMV模块相关的程序块共三条。



图3.6-2

OpenMV模块软串口设置，该指令与语音识别模块的管脚设置类似，是OpenMV模块与主控板通信的串口管脚设置、波特率设置、定义变量的程序块；

OpenMV有信号，指主控板接收到来自OpenMV的信号后，将要执行的语句写入倒U型中；

OpenMV读取数据，指的是OpenMV模块发送到主控板的数据。

### MU魔图识别

小 MU 视觉传感器是一款用于图像识别的传感器，其内置的深度学习引擎可以识别多种目标物体，例如颜色检测，球体检测，人体检测，卡片识别等。检测结果可以通过 UART 或 I2C 进行输出，体积小巧，功耗低，所有算法本地处理，无须联网，可广泛应用于智能玩具、人工智能教具、创客等产品或领域。

**（1）设置模块**

如图3.6-3所示，魔图识别模块的设置，包括：初始化MU、设置MU、恢复到出厂设置、LED灯、颜色识别、设置算法、数码变焦、设置串口波特率、设置摄像头白平衡、高帧率模式等。注意：除了初始化MU，其它的设置模块都需要放到设置MU模块里。



图3.6-3

1）“初始化MU”

如图3.6-4，可以选择mu的号码，默认为00，如果连接了不止一个魔图识别模块，可以设置为Mu01或者Mu10或者Mu11；也可以设置端口，通信方式可以设置为串口通信或者I2C通信，我们通常使用IIC的通信模式。

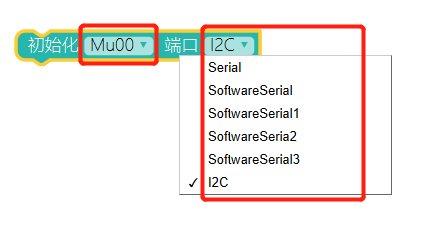


图3.6-4

2）设置Mu

如图3.6-5所示，可以选择Mu的号码该模块与基本模块区-控制里的“初始化”模块相似，都只是对程序的一种限定，并没有具体的指令。所有的魔图识别模块的设置都要在里面。



图3.6-5

3）“LED灯”

如图3.6-6所示，魔图识别模块上共有两个LED灯，可以选择1号或者2号灯，LED灯的亮度分为0-15，15最亮，0为熄灭，图中的亮度为1，另外，我们还可以改变颜色，如图所示，共有6种颜色，黑色为熄灭状态。



图3.6-6

4）“算法”

如图3.6-7所示，该指令可以“开启”或者“关闭”算法，算法类型共有七类，分别为：颜色识别、颜色检测、球、人体、形状卡片、交通卡片、数字卡片。

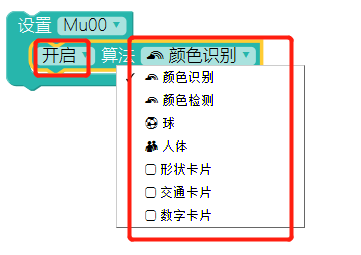


图3.6-7

如图3.6-8所示，设置算法的类别如图3.6-7所示共有七种；性能共有四种，分别是：自动、速度优先、性能均衡、准确率优先。

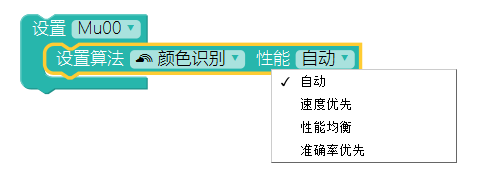


图3.6-8

5）“数码变焦“

如图3.6-9所示，数码变焦利用软件对已有像素周边的色彩进行判断，并根据周边的色彩情况插入经特殊算法加入的像素，数码变焦是以牺牲分辨率和图像质量为代价的变焦，尽量不要使用该功能。

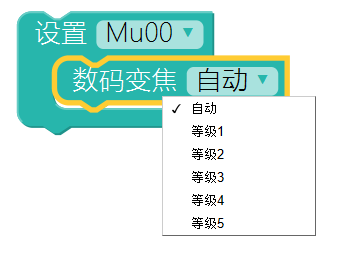


图3.6-9

6）“设置镜头白平衡”

如图3.6-10所示，摄像头白平衡共有四种：自动、锁定白平衡、白光模式和黄光模式，白平衡会对颜色检测产生明显的影响，建议使用自动锁定白平衡模式，将摄像头面对白纸进行测光，然后再识别颜色；

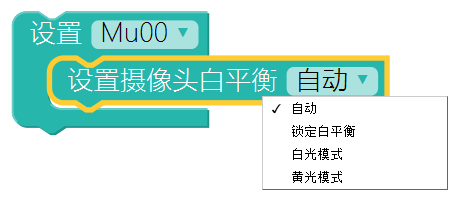


图3.6-10

**（2）运行模块**

如图3.6-11所示，运行模块都是不可以独立运行的模块，都是必须与其它模块配合使用。

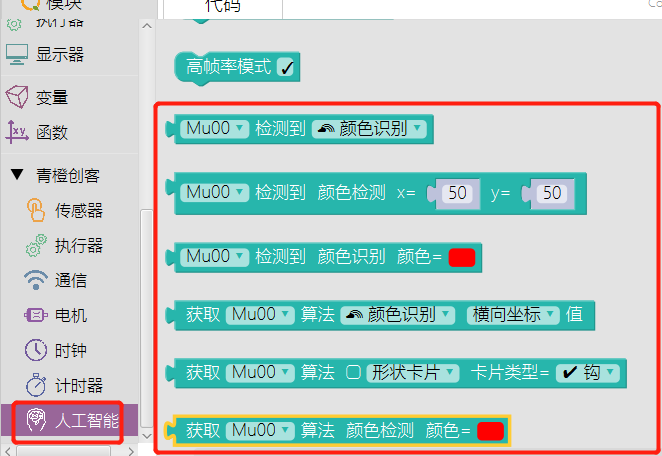


图3.6-11

如图3.6-12所示，与“数字输入”程序块类似，算法类型分为7类。



图3.6-12

如图3.6-13所示，颜色检测： x为水平中心坐标；y为垂直中心坐标。



图3.6-13

如图3.6-14所示，获取算法，有三类算法可用。其中，形状卡片分为图3.6-15所示的五种类型；交通卡片对应向前、向左、向右、掉头四种类型；数字卡片对应0-9共十个数字。

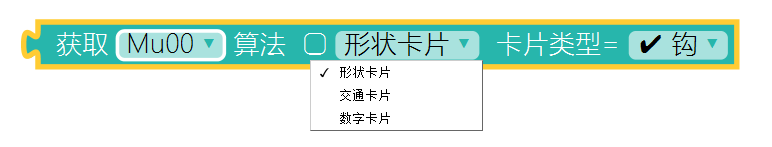


图3.6-14



图3.6-15

如图3.6-16所示，width：物体边缘宽度；height：物体边缘高度，label：分类标签编号，部分算法适用。



图3.6-16

# 第4章 甜橙示例程序

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目号 | 项目名称 | 任务 | 软件知识点 | 硬件元器件 |
| 1 | Hello ，World | 任务一：点亮真实的单色LED灯 | 数字输出模块  延时模块  顺序结构 | 1、单色LED灯 |
| 任务二：让LED灯循环闪烁 | 循环结构  变量的使用  顺序结构 | 1、单色LED灯 |
| 2 | S.O.S. | 任务：s.o.s.求救信号 | 数字输出  延时模块  循环模块 | 单色LED灯 |
| 3 | 神奇的按钮 | 任务一：简易延时灯 | 数字输入  逻辑  条件结构 | 按钮传感器  单色LED灯 |
| 任务二：使用按钮模拟开关1 | 声明变量  初始化  条件结构 | 按钮开关  单色LED灯 |
| 任务三：使用按钮模拟开关2 | 中断  条件结构 | 按钮  单色LED灯 |
| 4 | 简易入侵检测仪 | 任务一：简易入侵仪 | 条件结构  数字输入  逻辑非  数字输出 | 红外接近开关  单色LED灯 |
| 任务二：记录入侵 | 初始化  数字输入  中断  条件结构 | 按钮开关  红外接近开关  单色LED灯 |
| 5 | 可调灯 | 任务一：简易呼吸灯 | 函数程序块  循环结构  模拟输出  数学计算 | 1、单色LED灯 |
| 任务二：制作五档可调灯 | 初始化  变量声明与赋值  逻辑比较  条件结构  模拟输出 | 单色LED灯  按钮开关 |
| 任务三：制作旋钮可调灯 | 模拟输出  映射程序块  初始化  变量声明与赋值  中断  模拟输入  自定义函数 | 模拟角度电位器  按钮开关  单色LED灯 |
| 6 | 智能灯 | 任务一：声控灯 | 串口打印  模拟输入  条件结构  数学比较 | 模拟声音传感器  单色LED灯 |
| 任务二：制作楼道灯 | 串口打印  模拟输入  条件结构  逻辑 | 单色LED灯  模拟声音传感器  光线传感器 |
| 任务三：制作电子蜡烛 | 1、串口打印  2、模拟输入  3、条件结构  4、逻辑比较  5、循环结构 | 1、单色LED灯  2、模拟声音传感器  3、光线传感器 |
|  |  | 任务四：制作简易门铃 | 自定义函数  蜂鸣器程序块  条件结构  数字输入  5、逻辑非 | 1、蜂鸣器  2、按钮开关 |
| 7 | 温湿度 | 任务：串口打印温度值 | 变量的使用  串口打印  顺序结构 | 1、温湿度传感器 |
| 8 | 小小作曲家 | 任务一：简单音阶 | 顺序结构  蜂鸣器 | 1、蜂鸣器 |
| 任务二：简单乐曲 | 顺序结构  蜂鸣器 | 1、蜂鸣器 |
| 任务三：自编旋律 | 顺序结构  蜂鸣器  延时 | 1、蜂鸣器 |
| 9 | 噪音计 | 任务一：制作一个噪音计 | 舵机模块  映射 | 模拟声音传感器  舵机 |
| 任务二：探测最大噪音 | 中断  变量声明与赋值  初始化  条件结构  映射  数字输出 | 模拟声音传感器  按钮  舵机 |
| 10 | 遥控灯 | 任务：制作一个红外遥控灯 | 红外接收程序块  条件结构  变量 | 红外接收模块  遥控器  LED灯 |
| 11 | 超声波测距 | 任务：制作一个超声波测距仪，且在显示屏显示距离 | 初始化  LCD初始化  液晶显示屏  变量声明与赋值  显示屏清屏 | LCD1602显示屏  超声波 |
| 12 | 减速电机 | 任务：电机正反转 | 电机初始化  电机选择  顺序结构 | HR8833电机  2、外接电源 |
| 13 | 蓝牙通信 | 任务：蓝牙与PC通信 | AT模式设置  蓝牙 | HC05  PC、单色灯 |
| 14 | 节日闪烁彩灯 | 任务：ws2812全彩灯 | RGB灯准备  RGB灯控制  顺序结构 | RGB灯 |
| 14 | DIY海盗船系列 | 任务1：气压海拔检测  任务2：车辆测速与倒地报警仪 | 判断结构及其嵌套结构  OLED程序块的使用 | OLED屏幕； |

## 项目一 Hello,World!

### 任务 1——点亮真实 LED 灯

#### 1、任务目标

通过编写程序，控制一盏单色灯点亮1S后熄灭。

#### 2、流程图

编写程序的思路如图4.1-1所示：

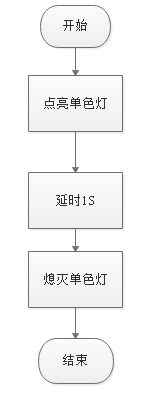


图4.1-1

#### 3、硬件连接

将 LED 灯与 10 号管脚相连。

注意：下文如无特殊说明，数字如“10”代表数字接口D10管脚，“3”代表D3。

#### 4、程序编写

1）从左侧 基本模块区—输入输出 中拖出“数字输出”程序块，如图4.1-2所示；

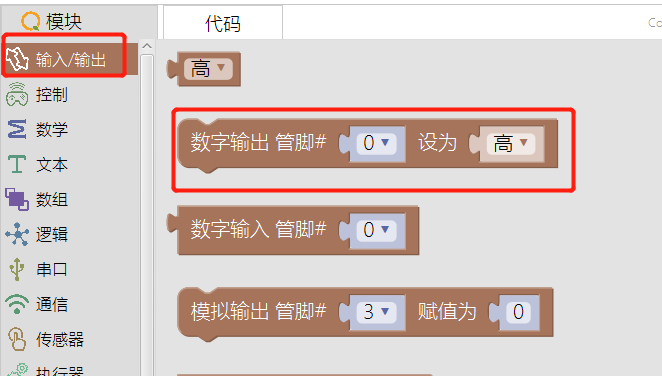


图4.1-2

2）从左侧 基本模块区—控制 中拖出“延时”程序块和“停止程序”程序块，如图4.1-3所示；



图4.1-3

3）将程序如图4.1-4所示；

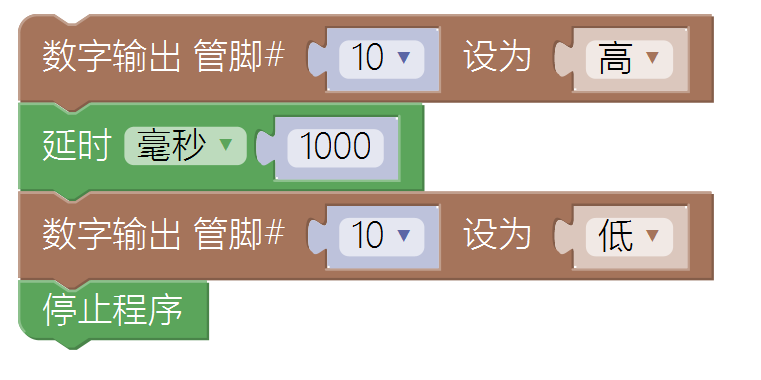


图4.1-4

#### 5、Q&A

**Q：单色灯是如何接线的？**

A：首先检查接线是否正确，注意，单色灯的S端与甜橙板的S端相连，V与甜橙板的V管脚相连，G与甜橙板的G端相连。

**Q：如何上传程序？**

A：程序编写完成后，点击“基本功能区”的“上传”按钮，观察“信息显示区”，当观察到“上传成功！”字样时，程序上传完成。

**Q：停止程序有什么用？**

A：停止程序意思是使程序结束运行，按下复位按键可以使程序从头开始运行。

#### 6、拓展

**1、知识点总结**

1）单色灯接线

2）数字输出程序块

3）延时程序块

4）停止程序程序块

**2、相关案例：**

1）闪烁灯（任务2）

2）S.O.S.求救信号（项目2任务1）

### 任务2——让LED越闪越快

#### 1、任务目标

通过编写程序控制一盏单色LED灯使其越闪越快，第一次点亮后灯保持1S然后熄灭100ms，第二次点亮后保持900ms,然后熄灭，以此类推，第十次点亮100ms，熄灭100ms，然后开始点亮1S后熄灭，重复执行。

#### 2、流程图

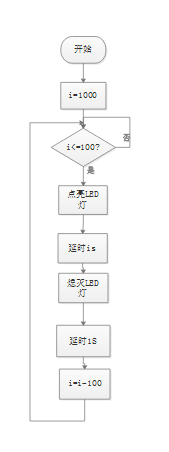


图4.1-5

循环的次数通过变量 i 的大小来控制。所谓变量，就是在程序运行过程中大小发生变化的 量。循环执行过程中，变量 i 从 1000 开始，每次减小 100，直到减小到 100，然后循环 结束。每次循环，都会执行一次包在其中的程序(这里就是控制灯亮灭的程序)。

#### 3、硬件连接

将 LED 灯与 10 号管脚相连。

#### 4、程序编程

1）首先来看这样一段程序，如图4.1-6所示:

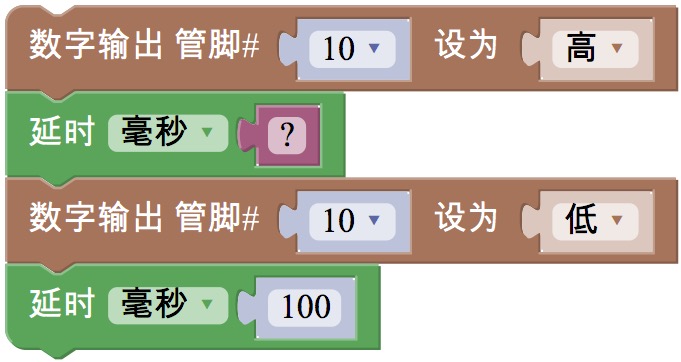


图4.1-6



图4.1-7

不难看出，图4.1-7这段程序是由图4.1-6所示的一小段不断重复而成的:

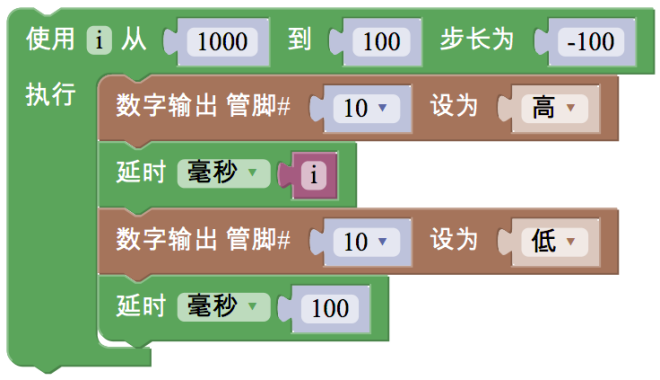
2）按照之前讲过的顺序执行原则，这一组模块就会一直重复下去。如果我们想让它执行很多次(比如 100 次)，一种办法就是，把 100 个这样的模块组前后连接在一起。

可想而知，如图4.1-7所示，编写的程序会非常长。并且，如果想知道它被重复执行了多少次，数起来也非常麻烦。那么，有没有一种模块，可以自动实现这个重复的过程，并且让人不怎么费力就知道重复执行了多少次呢?

从左侧模块区 控制 中找到如图4.1-8所示的程序块，



4.1-8



4.1-9

上面图4.1-9这段程序，实现了“重复”这个想法。它就是程序中常用的**循环结构**。如何理解这个循环结构呢?“使用 i 从 1000 到 100 步长为 -100”这句话是什么意思呢?

循环的次数通过变量 i 的大小来控制。所谓变量，就是在程序运行过程中大小发生变化的量。循环执行过程中，变量 i 从 1000 开始，每次减小 100，直到减小到 100，然后循环 结束。每次循环，都会执行一次包在其中的程序(这里就是控制灯亮灭的程序)。

写好程序后，单击上传按钮，将程序上传到甜橙主控板上，参考图 4.1-9。

注意：在上传程序之前，要设置好主控板型号和 COM 接口号，点击“上传”按钮右侧的下拉菜单即可设置。

#### 5、Q&A

**Q：程序上传成功后，单色灯常亮，没有闪烁？**

A1：检查程序，循环结构的步长是否为-100；

A2：检查程序，对否有单色灯熄灭和延时的程序块。

A3：检查硬件连接，线色的对应。

#### 6、拓展

**1、知识点总结**

1）顺序结构；

2）循环结构；

3）单色灯的接线：注意线色对应。

**2、相关案例：**

1）点亮真实的LED灯（任务1）；

## 项目二 S.O.S.

S.O.S.可以被展开为:

Save Our Souls (拯救我们的灵魂)

Save Our Ship (拯救我们的船)

Send Our Succour (速来援助)

Saving Of Soul (救命)

这是全球通用的求救信号。当你身在异乡，突遇险情，SOS 求救信号或许可以救你一命。

### 任务 1——用 LED 发出求救信号

#### 1、任务内容

实现功能：通过程序控制一盏单色灯点亮的时间，分别控制灯点亮的时间间隔为“三短三长三短”时，表示求救信号成功发出。

#### 2、流程图

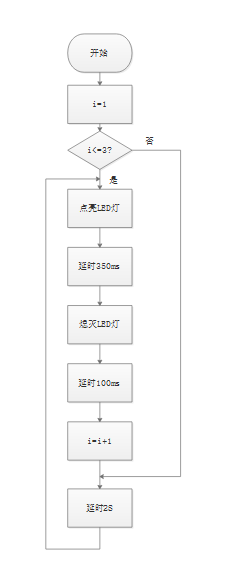


图 4.2-1

#### 3、硬件连接

将 LED 灯与 10 号管脚相连。注意插线时的颜色对应。

#### 4、程序编程

观察摩尔斯电码 S.O.S 的表示形式(“...---...”)，其中长短依次重复出现，可以采用上一任务中的循环结构:



图 4.2-2

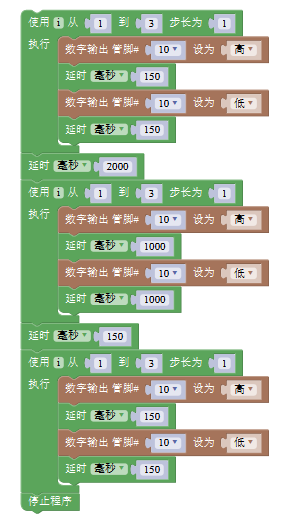


图 4.2-3

#### 5、 Q&A

**Q：单色灯出现点亮两短三长四短等不符合的规律，怎么办？**

A：此时，应明确，硬件接线、板型和端口号均无误；反复检查所写的程序，延时程序块的位置（循环里面还是外面）和时间值。

#### 6、拓展

**1、知识点总结**

1）单色灯的接线，注意线色对应。

2）程序结构的设计，循环结构，注意循环体执行的次数和条件。

**2、相关案例**

1）项目一任务2---让LED灯越闪越快。

## 项目三 神奇的按钮

按钮是我们在这套元件中接触的第一个输入设备，具有按下（高）和抬起（低）两种状

态。默认状态为抬起。生活中的按钮可以说无处不在，遥控器、计算器、手机、电脑等各种电子设备上的按键，都是按钮。

当按钮被按下时，向 2 号管脚输入低电平，此时 10 号管脚应输出高电平;

当按钮被抬起时，向 2 号管脚输入高电平，此时 10 号管脚应输出低电平。

程序如下图4.3-1和图4.3-2所示：



图4.3-1



图4.3-2

数字输入:有两种状态，即“高”和“低”。按钮按下为高，抬起为低。支持数字输入的管脚为:2~13， A0~A5(其实 0 号和 1 号管脚也支持数字输入，但因为程序上传过程中需占用0和1管脚，故一般情况下不建议使用)。

### 任务1——简易延时灯

#### 1、任务内容

实现功能：通过按下按钮开关后，单色LED灯被点亮保持3 秒后灯熄灭。

#### 2、流程图

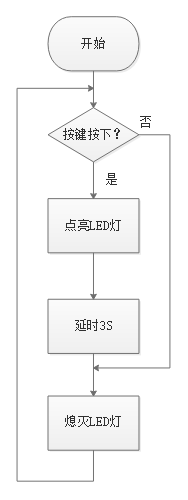


图4.3-3

#### 3、硬件连接

按钮开关----2；单色 LED 灯---10；注意插线时的颜色对应。

#### 4、程序编写



图4.3-4



图4.3-5



图4.3-6

这里我们看到了一个新的结构——**条件结构**。它是如何应用的呢?请看下图:



图4.3-7

如果当前状态符合判断条件的要求，判断模块内(红色边框内的)的程序将会被执行。如果不符合，则那部分代码将会被跳过，直接执行接下来黑色框中的程序。

#### 5、Q&A

**Q：程序上传成功后，按下按键单色灯不熄灭？**

A：观察熄灭灯的程序块的位置，应在条件结构外面。

**Q：按键开关可以连接任意的管脚吗？**

A：是的，但是，如果设计连接管脚为D0或D1时，需注意：先上传程序后，再连接硬件。

#### 6、拓展

**1、知识点总结**

1）按钮的工作原理为：按下按键，电路接通，此时按钮输出为低电平。如图3.3-7所示两程序完全相同：按下开关，单色灯被点亮；松开按键开关，灯熄灭。

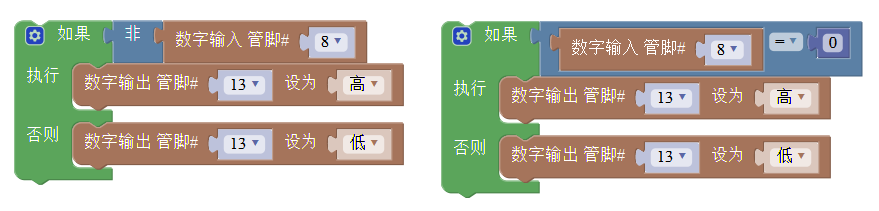


图4.3-8

2）按钮开关具有机械抖动的缺点，在使用代码编程时，需添加防抖动的程序。

3）条件结构，注意判断的条件与可执行语句的区别

4）数字输入程序块

**2、相关案例**

1）使用按钮模拟开关（任务2、任务3）；

### 任务2——使用按钮模拟开关(2)

#### 1、任务目标

实现功能：通过控制按钮开关来控制单色灯的状态，当按下按钮时，单色LED灯被点亮，保持3 秒后灯熄灭。

#### 2、流程图

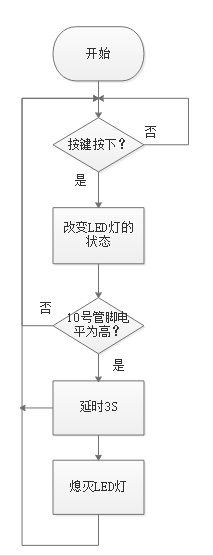


图4.3-9

#### 3、硬件连接

硬件连接：按钮开关----2；单色LED灯----10；注意插线时的颜色对应。

#### 4、程序编程



图4.3-10



图4.3-11



图4.3-12

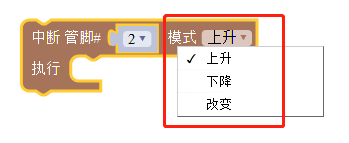


图4.3-13

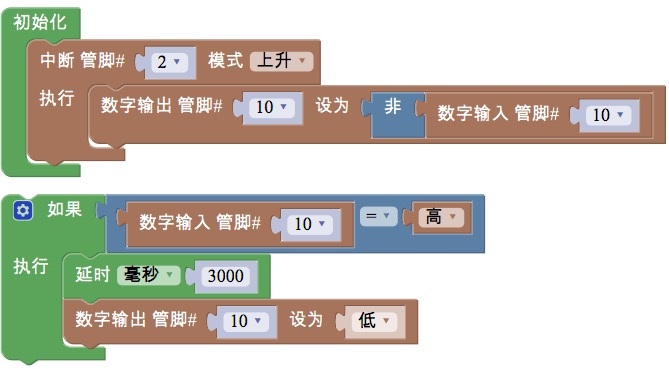


图4.3-14

这段程序中，我们使用了一个新的模块——**中断**。

**中断**过程是指，在程序顺序执行的过程中，当主控板收到规定的某些数字输入信号时，会立即暂停当前执行的程序，转而去执行中断部分的程序，当中断部分执行完毕，再回到刚 才暂停的地方继续执行原来的程序。

注意:主控板上只有 2 号和 3 号管脚支持中断功能。

举一个简单的例子：比如你在家看电视，突然门铃响了（家人回来了），那么你

不得不停下看电视先去开门，之后你又可以继续看电视啦！在整个过程中接电

话就是一个中断过程，门铃响就是中断的标志，即触发中断的条件。

上面的程序中，当人按下按钮时，便会暂停当前程序进入中断，执行中断中的模块。

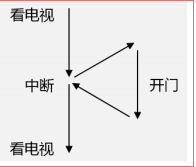


图4.3-15

#### 5、Q&A

**Q：为什么数字输入的管脚同时也可以是输出管脚？**

A：数字输入只是表示所对应的电平的高低，而不执行相应的语句。

#### 6、拓展

**1、知识点总结**

1）复位按钮，按下后，程序恢复刚上传成功的状态；

2）中断程序块；

3）延时程序块；

**2、相关案例：**

项目三的任务1和任务2；

## 项目四 简易入侵检测仪

### 任务1——简单入侵检测仪

#### 1、任务目标

当有人走近红外开关的检测范围时，灯被点亮，人走远一定距离后熄灭灯。

#### 2、流程图

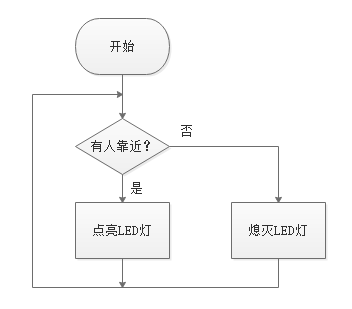


图4.4-1

#### 3、硬件连接

硬件连接：红外入侵检测仪——3；LED ——10。注意红外接近开关的插线：棕对红，蓝对黑，黑对黄。

#### 4、程序编程

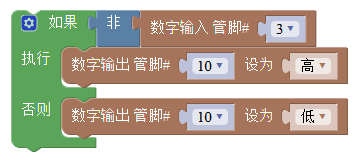


图4.4-3

#### 5、Q&A

**Q：人靠近红外接近开关没有使灯点亮？**

A：将红外开关正对障碍物。

#### 6、拓展

**1、知识点总结**

1）红外接近开关是一种数字输入设备。具有未入侵(高)和入侵(低)两种状态。默认状态为未入侵(高)。

2）红外接近开关:红外线属于一种电磁射线。可见光波长是 380nm-780nm，红外线波长为 780nm-1mm，而红外接近开关使用的是接近可见光波长的近红外线。

3）利用被检测物体对红外光束的遮光(位于红外发射器另一侧的接收器接收不到红外线)或反射(位于 红外发射器同侧的接收器接收到物体反射回来的红 外线)，检测物体的有无。

4）红外接近开关对所有能反射光线的物体均可检测。

5）使用数字输入管脚，读取输入口数值 1/0 代表有无接收红外信号。

6）条件结构；

2、相关案例：任务2。

### 任务2——记录入侵

#### 1、任务目标

红外接近开关控制点亮灯，按钮控制熄灭灯。当障碍物走进红外开关的检测范围时，灯被点亮，通过按下按钮熄灭灯。

#### 2、流程图

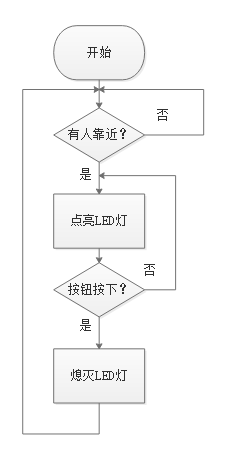


图4.4-3

#### 3、硬件连接

硬件连接：红外入侵检测仪——3；LED ——10；按钮——2。注意红外接近开关的插线：棕对红，蓝对黑，黑对黄。

#### 4、程序编程

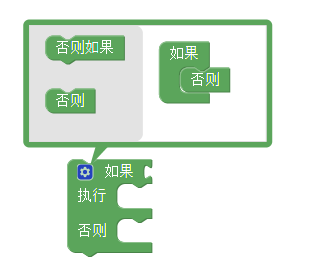


图4.4-4

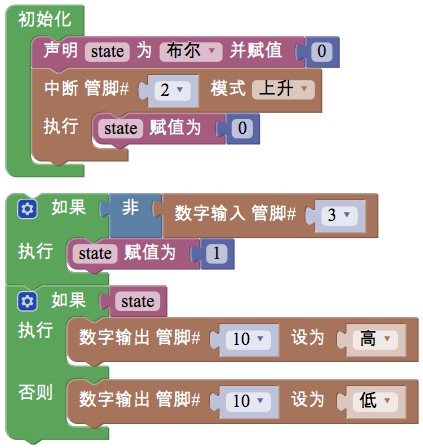


图4.4-5

#### 5、Q&A

**Q：红外开关检测到物体后，灯没有点亮？**

A：检查线色的对应情况，按下复位按钮，重新检测；

**Q：按下按钮开关后，灯没有熄灭？**

A：将中断程序块放在初始化模块中，检查STATE的变量类型。

#### 6、拓展

**1、知识点总结**

1）红外开关的原理；

2）按钮开关；

3）布尔变量的转换机制；

**布尔变量**：Boolean Variable (布尔型变量) 是有两种逻辑状态的变量，它包含两个值：真和假。如果在表达式中使用了布尔型变量，那么将根据变量值的真假而赋予整型值1或0。要把一个 整型变量转换成布尔型变量，如果整型值为0，则其布尔型值为假；反之如果整型值为非0，则其布尔型值为真。布尔型变量在运行时通常用做标志，比如进行逻辑测试以改变程序流程。

**2、相关案例：**项目四的任务1；

## 项目五 可调灯

灯光在微电脑控制之下完成由暗到亮再由亮到暗的逐渐变化，感觉像是在呼吸，所以称为**呼吸灯**。呼吸灯广泛应用于手机、无线路由器之上，如你的手机里面有未处理的通知，比 如说未接来电，未查收的短信等等，呼吸灯就会由暗到亮的变化，像呼吸一样那么有节奏，起到一个通知提醒的作用。

### 任务1——简易呼吸灯

#### 1、任务目标

在程序的控制下，灯的亮度首先逐渐变亮，达到最亮后，逐渐变暗，完全熄灭后，再次逐渐点亮，依次循环。

#### 2、流程图



图4.5-1

#### 3、硬件连接

硬件连接：LED ——10。注意插线时的颜色对应。

#### 4、程序编程

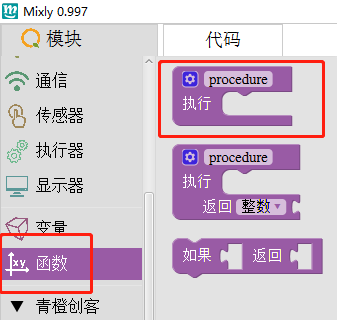


图4.5-2

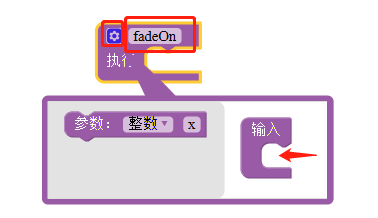


图4.5-3

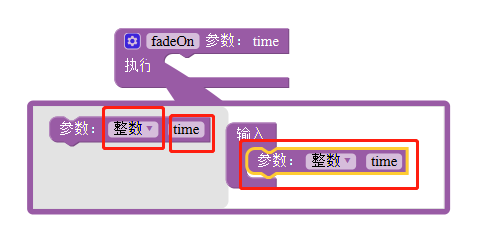


图4.5-4



图4.5-5

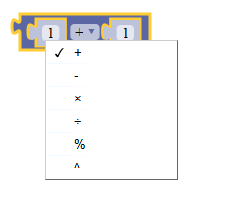


图4.5-6



图4.5-7

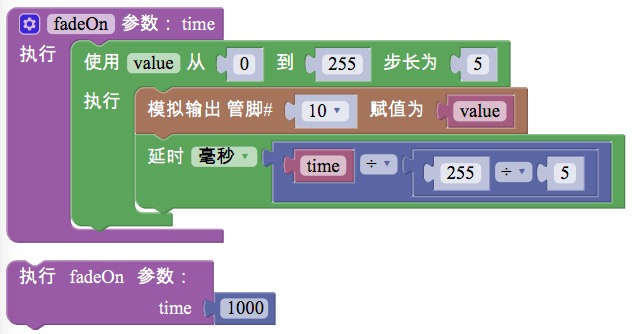


图4.5-8

不难看出，上面这个函数的功能是让 LED 灯逐渐变亮。 要实现任务要求，还需要一个让 LED 逐渐变暗的函数。 完整的程序如下图：



图4.5-9

这段程序中我们接触到了两个新的模块——**函数以及模拟输出。**

左侧两个紫色块内的程序需要自行编写，它们的功能是定义两个函数。程序的主体是右侧的两个模块，它们是对左侧的函数的调用。

#### 5、Q&A

**Q：灯逐渐点亮后，并没有逐渐熄灭？**

A：检查fadeOff函数的步长为-5，而不是5。

#### 6、拓展

**1、知识点总结**

1）函数：我们最初见到函数是在数学当中。y=f(x)是函数的一种一般形式，它接受变量 x 的值，经过对应法则 f 的处理，向人返回结果值 y。Mixly 程序中的函数，可以类比理解。程序中的 fadeOn 函数接受了变量 time 的值，对其进行了操作，并向程序的其余部分返回一个值。

有区别的是，这里的变量 time，我们称之为“参数”，函数返回的值，是“空值”，因为这个函数的目的，不在于返回一个数值，而在于对于硬件进行输出操作。另外，函数的内部，也可以定义变量（如图3.5-3中的 value），这个变量只在函数执行的时候存在，一旦函数执行结束，value 也就不存在了。函数的一次执行叫做函数的调用，在一个函数中，可以调用其他函数，甚至可以调用自己。

2）模拟输出：脉宽调制（PWM：Pulse Width Modulation）输出：它是一种对模拟信号电平进行数字编码的方法，简单来说就是通过一个时钟周期内高低电平的不同占空比来表征模拟信号，如图3.5-6所示就是一个具体的编码样例。

Arduino 使用 analogWrite(int value)输出 PWM 信号，其中的value 取值范围是 0-255，效果如图3.5-5所示。Arduino 主控板只有有限个 GPIO 管脚支持 PWM。观察一下 Arduino 板，查看数字引脚，你会发现其中 6 个引脚（3、5、6、9、10、11）旁标有“~”，这些引脚不同于其他引脚，因为它们可以输出 PWM 信号。

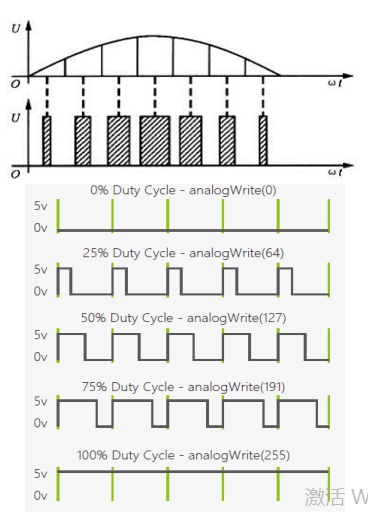


图4.5-10

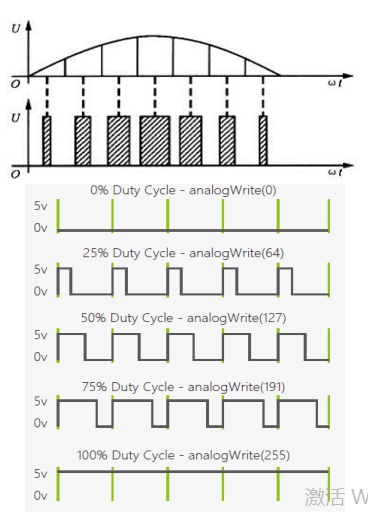


图4.5-11

**2、相关案例：**项目5的任务1；

### 任务2——制作5档可调灯

#### 1、任务目标

通过按钮调节单色灯的亮度，从单色灯熄灭，按下按钮，灯被点亮，再次按下按钮，单色灯更亮一点，第五次按下按钮后，单色灯最亮，第六次按下按钮，灯熄灭。

#### 2、流程图

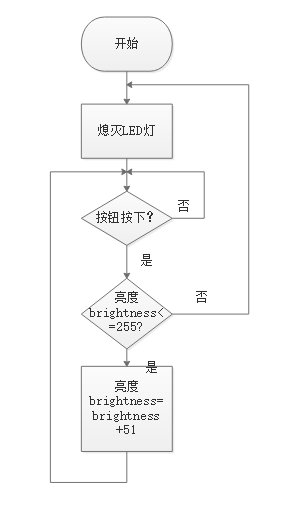


图4.5-12

#### 3、硬件连接

硬件连接：LED ——10；按钮——2。注意插线时的颜色对应。

#### 4、程序编程

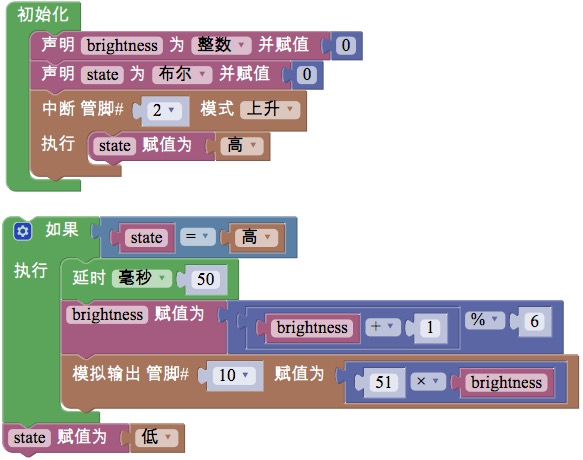


图4.5-13

程序中“brightness 赋值为”一句里，对变量进行了运算操作:(brightness+1)%6。这里的“%”是一个运算符号，叫做“取余”，这个式子的结果是 brightness+1 后的数 除以 6 得到的余数。

#### 5、Q&A

**Q：单色灯的亮度分为五档，但是却对6取余，而不是5，为什么？**

A：取余的右方应写6，因为还有0挡，也就是灯完全熄灭的时刻。

#### 6、拓展

**1、知识点总结**

1）按钮的按下与松开，分别代表输出的电平高与低。

2）模拟输出的范围为0-255；

3）变量赋值使用时，需先声明，且声明放入“初始化”程序块。

**2、相关案例**

任务1--简易呼吸灯；

任务3--制作旋钮可调灯；

### 任务3——制作旋钮可调灯

这里，我们用到了一个新的元件——模拟角度电位器，也叫 “滑动变阻器”。通过调节旋钮，可以改变它接入电路的阻值大小。将其连到主控板支持模拟输入的接口上，就可以把阻值作为模拟信号输入到主控板上。主控板根据输入值的大小，确定输出的值(在这里，输入值大，输出值也大;也可能另外一些程序希望输出值随着输入值变大而减小。)

#### 1、任务目标

按下按钮开关，灯被点亮，此时可以使用旋钮调节LED灯的亮度强弱，当单色灯点亮时再次按下按钮，灯熄灭。

#### 2、流程图

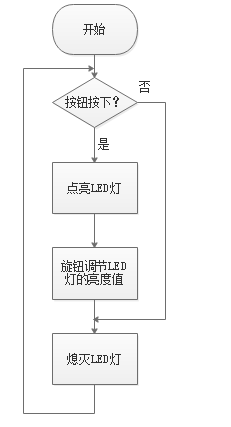


图4.5-14

#### 3、硬件连接

硬件连接：LED ——10；模拟角度电位器——A0；按钮——2。注意插线时的颜色对应。

#### 4、程序编程

需要注意的是，主控板支持的模拟输入信号的大小范围是 0~1023。然而，模拟输出大小是 0~255。因此，模拟输入的数值，不能直接进行模拟输出，我们需要一种办法，能够把 0~1023 内的数，按比例缩小，转化成 0~255 之间的数，再模拟输出。方法如下:



图4.5-15

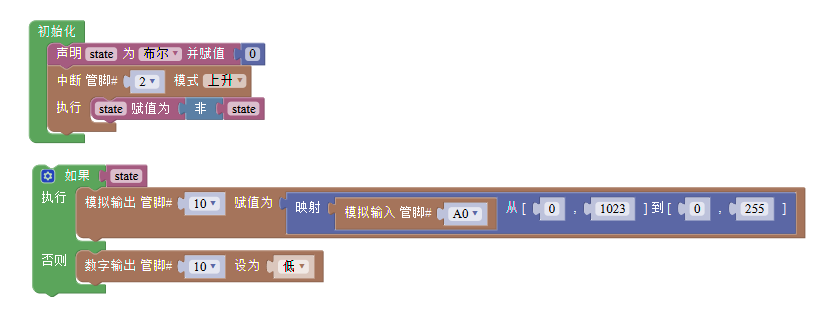


图4.5-16

#### 5、Q&A

**Q：旋钮从最小阻值到最大阻值，经历了四次亮灭，为什么？**

A：查看程序是否使用的“映射”程序块，而不是“约束”程序块，其次查看映射的取值范围，模拟输入的范围为0-1023，输出的范围为0-255。

#### 6、拓展

**1、知识点总结**

1）旋钮的原理，与滑动变阻器类似；

2）“映射”程序块，表达一种输出的对应关系；

3）中断，切换state变量的状态，同时切换灯的状态；

4）条件结构；

**2、相关案例**

1）本项目任务1；

2）本项目任务2；

## 项目六 智能灯

智能灯是以控制、灯光效果、创作、分享、光与音乐互动、光提升健康和幸福为特点的新型智能设备。开关灯的控制，不但可以通过控制器，手动遥控灯的开关，还可以进行声控开关灯的控制。

我们需要通过声音代替按钮形式的开关。如何识别声音信号呢?我们需要使用 一个新的原件——模拟声音传感器。

模拟声音传感器可以将声音的响度转化成模拟信号。在 Arduino 主控板上， 仍然是输入 0~1023 的数值。

### 任务1——声控灯

#### 1、任务目标

用串口打印出通过模拟声音传感器采集的环境中的声音值，当有响声且大于某设定值时，灯被点亮并且持续一段时间，当检测到声音的数值小于某设定值时，灯熄灭。

#### 2、流程图

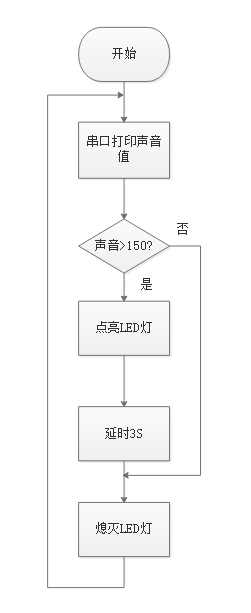


图4.6-1

#### 3、硬件连接

硬件连接：LED ——10；模拟声音传感器——A0。注意插线时的颜色对应。

#### 4、程序编程

1）我们看到，程序中使用了一个新的模块——**串口打印**



图4.6-2



图4.6-3

2）模拟输入的信号是通过串口传给 Arduino 主控板的，使用“串口打印”可以把当前通过串口的数据显示出来。显示在串口监视器里。



图4.6-4

3）单击串口监视器，会弹出一个新窗口，这个窗口中显示上传的数据:

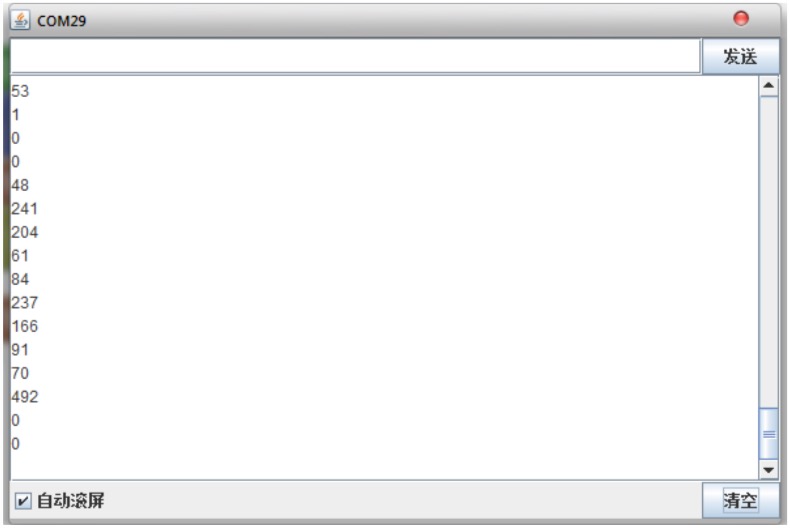


图4.6-5

#### 5、Q&A

**Q：程序上传完成后，打开串口，没有检测到数据输出？**

A：检查串口号是否为当前使用的端口，检查的方法为：点击端口的下拉按钮，是否有多个端口，如果是，拔掉甜橙版，重新插入，会重新出现的端口号为目标端口号。

#### 6、拓展

**1、知识点总结**

1）串口打印

2）模拟传感器

3）逻辑比较

4）条件结构

**2、相关案例**

1）本项目任务2；

2）本项目任务3；

### 任务2——制作楼道灯

#### 1、任务目标

程序首先通过模拟声音传感器和模拟光线传感器分别打印出检测到的声音值和光强值，然后根据这两个值作为条件进行判断，当声音值大于某设定值并且同时光强值小于某设定值时，点亮单色LED灯，并且保持3S，当没有同时满足这两个条件时，熄灭单色LED灯。

#### 2、流程图

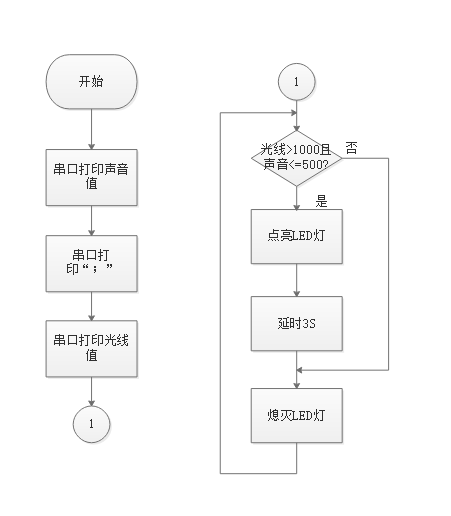


图4.6-6

#### 3、硬件连接

硬件连接：LED ——10；模拟声音传感器——A0；模拟光线传感器——A1。注意插线时的颜色对应。

#### 4、程序编程

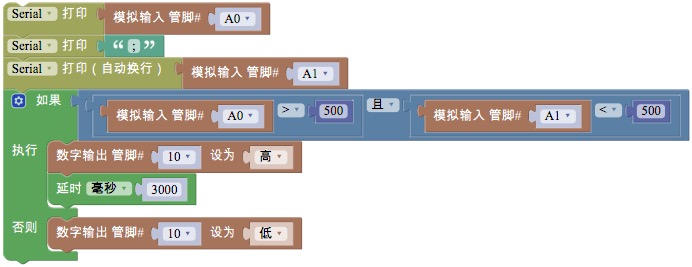


图4.6-7

#### 5、Q&A

**Q：“如果”程序块的条件怎么编写的？**

A：首先，明确条件由两部分组成，分别是光强和声音，这两部分必须同时满足，故逻辑关系为“且”，声音和光强的每部分内部都是“逻辑比较”的关系，故编写的顺序为：从基本模块区--逻辑 中拖出“逻辑选择”程序块，再拖出“逻辑比较程序块”，再从基本模块区--输入/输出 中拖出“模拟输入”程序块，再从基本模块区--数学 中拖出“数值”，修改“数值”的值为500，组成如图3.6-10所示的程序块；“逻辑选择”程序块的右侧的编写与左侧同理。



图4.6-8

#### 6、拓展

**1、知识点**

1）“逻辑选择”与“逻辑比较”的嵌套使用；

2）串口打印；

3）条件结构；

**2、相关案例**

1）本项目任务1；

### 任务3——制作电子蜡烛

#### 1、任务目标

首先打开Mixly的串口监视器，可以分别看到模拟声音传感器和模拟光线传感器检测到的声音强度值和光强值；当亮度暗时，蜡烛“点亮”，火苗闪烁，有吹蜡烛的声音的时候，蜡烛熄灭。

#### 2、流程图

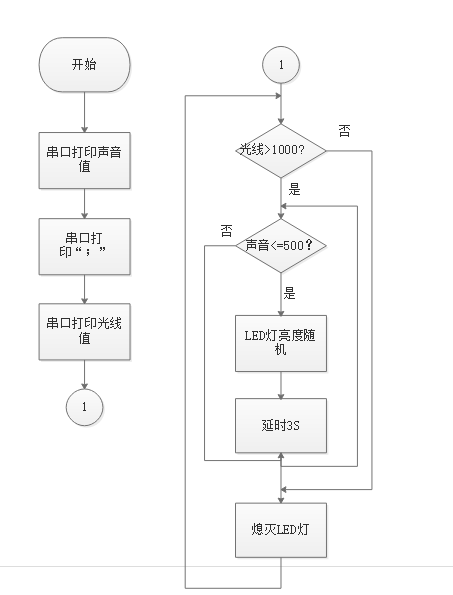


图4.6-9

#### 3、硬件连接

硬件连接：LED ——10；模拟声音传感器——A0；模拟光线传感器——A1。注意插线时的颜色对应。

#### 4、程序编程

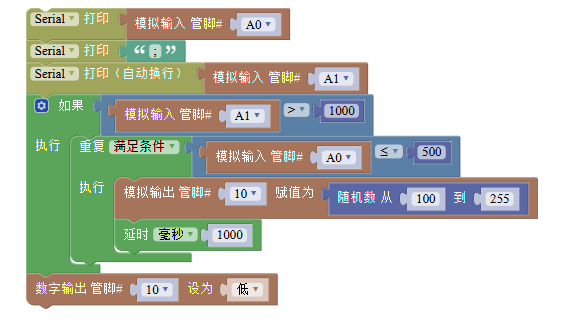


图4.6-10

#### 5、Q&A

**Q：为什么程序中要设定光强大于1000时，在限定声音值小于500时，蜡烛开始闪烁？**

A：光强大于1000的时刻，代表点蜡烛的瞬间，光线最强。

#### 6、拓展

**1、知识点总结**

1）“随机数”程序块；

2）串口打印；

3）循环结构；

**2、相关案例**

1）本项目任务2和任务1；

### 任务4——制作简易门铃

#### 1、任务目标

按下按钮开关，蜂鸣器开始播放两段频率不同的音乐，播放完毕后停止。

#### 2、流程图

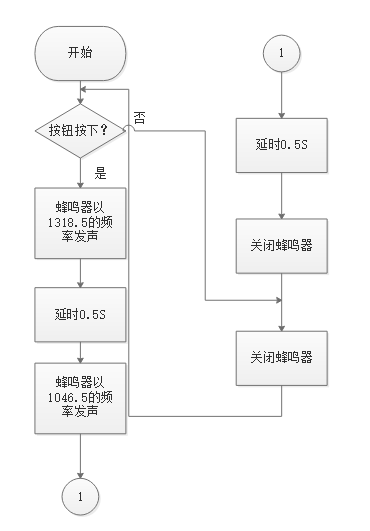


图4.6-11

#### 3、硬件连接

硬件连接： 按钮——2；蜂鸣器——8。注意插线时的颜色对应。

#### 4、程序编程

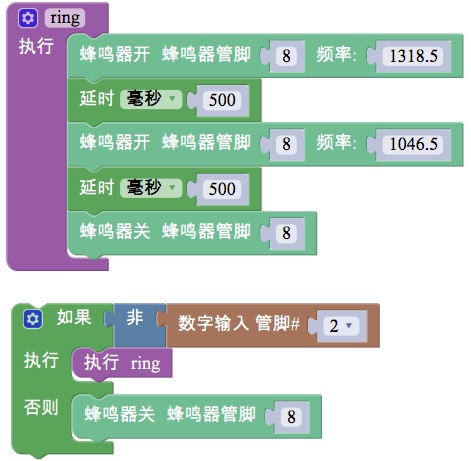


图4.6-12

#### 5、Q&A

**Q：按钮按下没有使音乐播放停止？**

A：此程序中，按钮开关的作用，只能播放音乐，而不能使音乐声停止。

#### 6、拓展

**1、知识点总结**

1）自定义函数；

2）条件结构

3）“蜂鸣器开”与“蜂鸣器关”的配合使用；

**2、相关案例**

1）本项目任务1；

## 项目七 制作一个温湿度仪

### 任务1——制作一个温度仪

#### 1、任务目标

在串口打印出温湿度传感器测出的温度值和湿度值。

#### 2、流程图

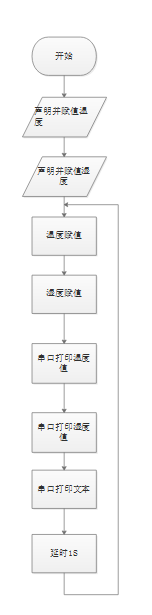


图4.7-1

#### 3、硬件连接

硬件连接：DH11---8。

#### 4、程序编程

1）打开 Mixly,在左侧模块栏选择 传感器 →DHT11，如图 4.7-2 所示，示例中温湿度传感器数据输出引脚连接到主控板的 D8 端口，因此在程序中将管脚号改为 8，如图4.7-3。



图4.7-2



图 4.7-3

2）在模块栏中分别选择 串口 →Serial 波特率以及 Serial 打印（自动换行），如图 4.7-4 所示。



图 4.7-4

3）从模块栏中分别选择声明变量、文本框和延时模块，按照图 4.7-5所示连接。

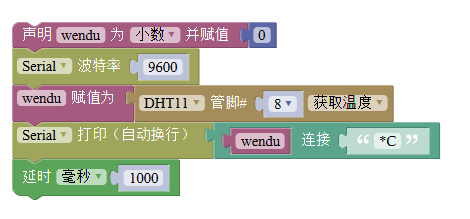


图 4.7-5

4）上传程序到主控板，打开串口监视器，当改变环境温度时（可将温湿度传感器放到手心或者靠近装热水的水杯），可以从串口监视器中观察到温度逐步上升，如图 4.7-6 所示。

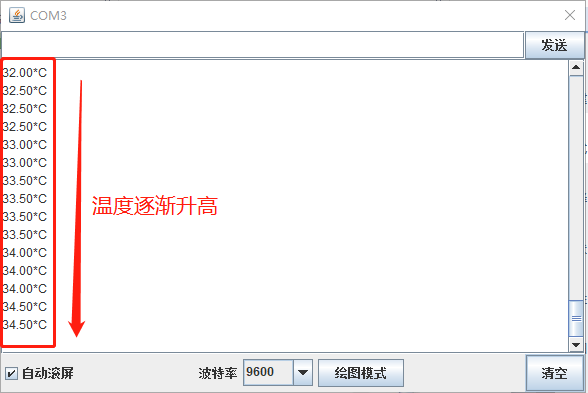


图 4.7-6

5）当需要同时读取温度和湿度值时，可以参考如图 4.7-7 所示的程序，变量的声明和使用以及文本框的调用可参照本章节温度传感器在 Mixly 中使用示例的步骤。当环境温湿度改变时，可通过串口监视器观察到 DHT11 返回的温湿度变化，如图 4.7-8 所示。



图 4.7-7

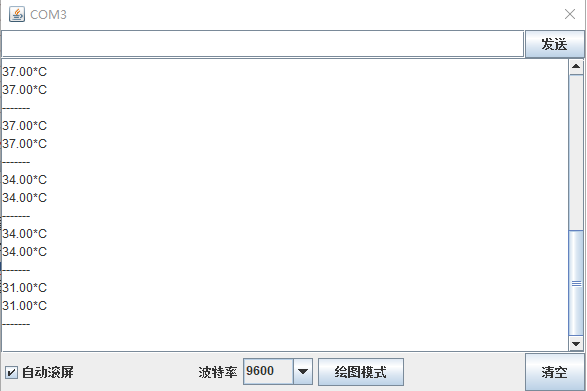


图 4.7-8

6）在 Mixly 示例程序文件中打开“温湿度传感器 DHT11”示例文件，将此示例上传到主控板后，可从串口监视器中观察到温湿度传感器返回的数据，改变环境温湿度，可观察到串口打印的温湿度数据也逐渐随之改变。

#### 5、Q&A

**Q：靠近热源后，串口打印的温度值没有变化？**

A：最好声明一个变量，用来存储检测到的温度值。

#### 6、拓展

**1、知识点总结**

1）“DHT11温湿度传感器”程序块的使用；

2）温湿度传感器的接线；

3）变量的使用，先声明，在赋值使用；

**2、相关案例**

温湿度传感器DHT11与温度传感器DS18B20的使用方法的异同。

## 项目八 小小作曲家

本节中，我们将向你介绍一种用Mixly播放音乐的方法。这一节中，你能够了解到一 些有关音乐的基本知识，以及它们如何与 Arduino 和 Mixly 结合，并试着自己编一段小旋律。

### 任务1——简单音阶

#### 1、任务目标

这段程序实现了一个简单的功能:蜂鸣器能按顺序播放一组音阶(do, re, mi, fa, sol, la, si)。程序 的开头定义了一个含有 7 个元素的数组，每一个元素对应着一个音的频率。

#### 2、流程图

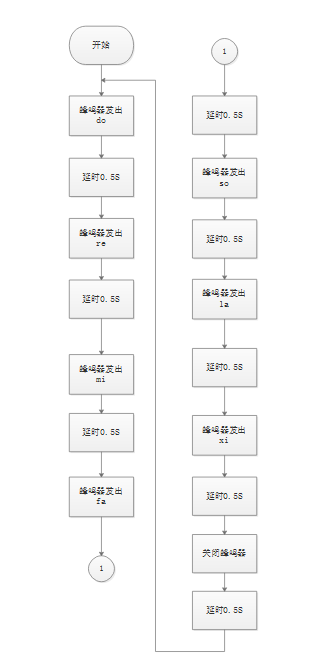


图4.8-1

#### 3、硬件连接

硬件连接： 蜂鸣器——8。注意插线时的颜色对应。

#### 4、程序编程



图4.8-2

#### 5、Q&A

**Q：蜂鸣器播完一组音阶后，没有停顿，重复开始执行？**

A：没有加入“蜂鸣器关”程序块。

#### 6、拓展

**1、知识点总结**

1）“蜂鸣器开”与“蜂鸣器关”的配合使用；

2）顺序结构；

**2、相关案例**

1）项目七的任务1；

2）本项目的任务2；

### 任务2——简单乐曲

#### 1、任务目标

用蜂鸣器播放简单乐曲。

#### 2、流程图

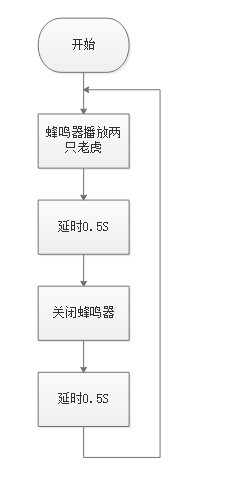


图4.8-3

#### 3、硬件连接

硬件连接： 蜂鸣器——8。注意插线时的颜色对应。

#### 4、程序编程



图4.8-4

#### 5、Q&A

**Q：程序上传成功后，蜂鸣器不响？**

A：检查连线，蜂鸣器的IN与甜橙板D8管脚相连，注意线色对应。

#### 6、拓展

**1、知识点总结**

1）关于音乐的基本知识

一般来讲，一个音有以下要素：音高、音色、响度。在乐曲中，时值也是一个重要的要素。

声音与物体的震动：声音是由物体的震动产生的。

音高与频率：一个音的音高，由物体振动的频率决定。振动频率指物体振动的快慢，频率越高，物体振动越快，音高越高。

音色与材质：不同的物体的材质不同，振动时产生的声音的“色彩”不相同。

响度与振幅：一个音色响度，由物体振动的幅度决定。物体振动幅度越大，响度越大。

音名：前面已经知道，音调的高低是由频率决定的。对于音乐来说，人们已经发现，用一些特定的频率的音，演奏出来的曲子比较悦耳。人们也将这些频率用英文字母代替，方便记忆。这个英文字母，就是它代表的那个音的音名。（例如，规定 A=440Hz，那么，A 就是一个音名）音名是唯一的，任何情况下，一个音名只能代表同一个频率。（如 A 永远代表 440Hz）。

唱名：音乐课上，你肯定听到过 do re mi fa sol la si 吧？这些就是唱名。因为唱出来比较简单。”do ”可以对应不同的频率，可以是 440Hz，也可以是 880Hz，也可以是……。所以可以看出音名和唱名的区别——一个音名值对应一个频率，一个唱名可以对应多个频率。

时值：指一个音的长短。

2）“蜂鸣器开”与“蜂鸣器关”的配合使用；

**2、相关案例**

1）项目七的任务1；

2）项目把的任务1；

## 项目九 噪音计

生活中我们可以看到一些仪器上指针——时钟、电压表、电流表、汽车上的转速表......甜橙 套件中也有一个元件可以模拟指针，它就是**舵机**。

舵机是由直流电机、减速齿轮组、传感器和控制电路组成的一套自动控制系统通过发送信号，指定输出轴旋转角度套件中舵机最大旋转角度:180°。

这是 Mixly 软件中控制舵机的程序块:

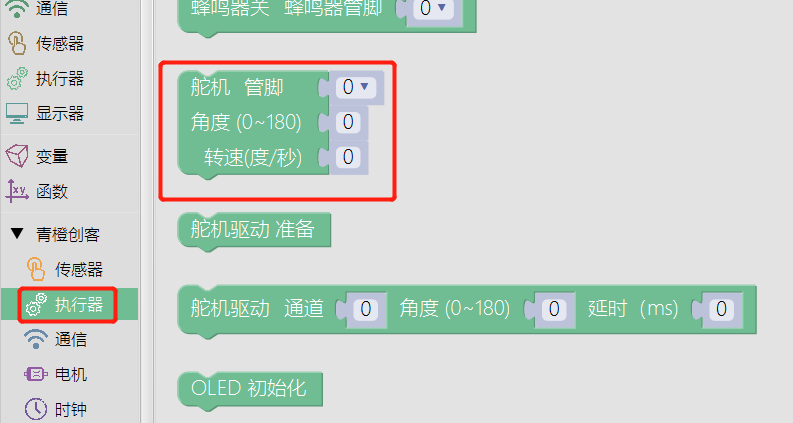


图4.9-1

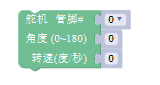


图4.9-2

### 任务1——制作一个噪音计

#### 1、任务目标

根据模拟声音传感器检测到的噪音的大小，控制舵机转动到不同的角度。

#### 2、流程图

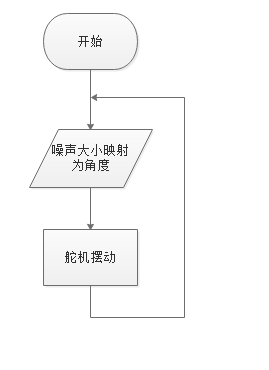


图4.9-3

#### 3、硬件连接

硬件连接： 舵机——9；模拟声音传感器——A0。注意插线时的颜色对应。

#### 4、程序编程



图4.9-4

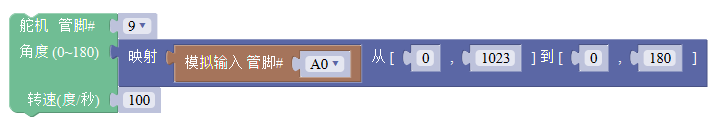


图4.9-5

#### 5、Q&A

**Q：无论怎么调节噪声，舵机都没有转动到180度位置？**

A：查看程序中是否正确使用了“约束”程序块，二看延时的时间，是否太小，因为舵机的转动需要一定时间。

#### 6、拓展

**1、知识点总结**

1）“舵机”程序块；

2）“映射”与“模拟输入”程序块的结合；

**2、相关案例**

1）本项目任务2；

### 任务2——探测最大噪音

#### 1、任务目标

模拟声音传感器用来探测一段时间内噪音的最大值，用户按下按钮开关后模拟声音传感器开始工作， 只记录最大值。再次按下按钮开关停止记录，并回到初始位置。

#### 2、流程图

思路解析：

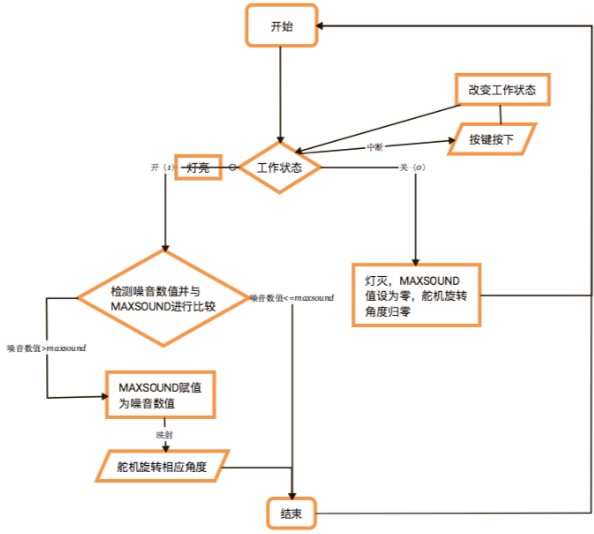


图4.9-6

如何使按键随时改变工作状态——**中断函数。**

在这个程序中，我们主要需要用一个 state变量来判断工作状态，若 state 值为1(即开) 各部分按要求工作，若 state值为0(即关)则所有部分归零;其次，在工作过程中我们还 需随时检测按键有没有被按下以改变state的值;噪音计和舵机的工作状态与按键状态是 两个可以独立运行的部分(没有什么明显的逻辑关系语句可以把两个部分建立联系)，因此，我们考虑用中断函数，即在工作或停止工作过程中随时检测按键状态(在用按键控制 复杂的元件的开关状态时经常使用中断函数)。

#### 3、硬件连接

硬件连接： 舵机——9；按钮——2；模拟声音传感器——A0。注意插线时的颜色对应。

#### 4、程序编程

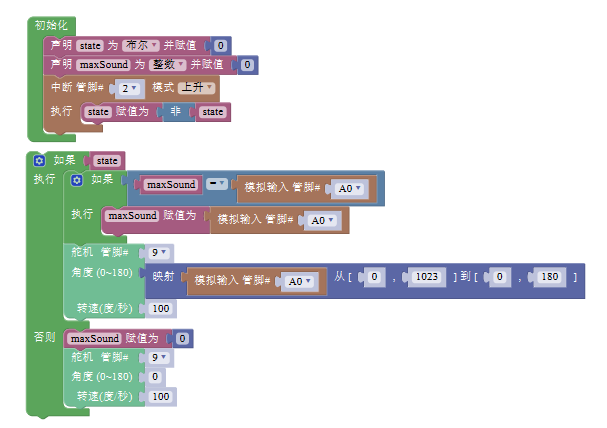


图4.9-7

#### 5、Q&A

**Q：按下按钮开关后，舵机没有开始摆动或摆动角度很小？**

A：模拟传感器的输入值较小，此时可以修改映射的输出范围；

#### 6、拓展

**1、知识点总结**

1）“舵机”程序块与“映射”、“模拟输入”程序块的结合使用；

2）条件结构；

3）变量的使用；

**2、相关案例**

本项目的任务1；

项目五的任务2；

## 项目十 遥控灯

想象自己躺在床上，该睡觉了，灯的开关却在几米外的门口，你不愿离开温暖的被窝，在寒冷中跋涉漫长的几米去关灯。你需要一个遥控器。 现实世界的大多数遥控器都是发出红外的信号，如电视机遥控器，机顶盒遥控器等。

任何一个遥控系统都由发射器和接收器两部分组成。

每个按钮都有一个特定的 16 进制代码，都以 FD 开头，在接收过程中有可能出错。

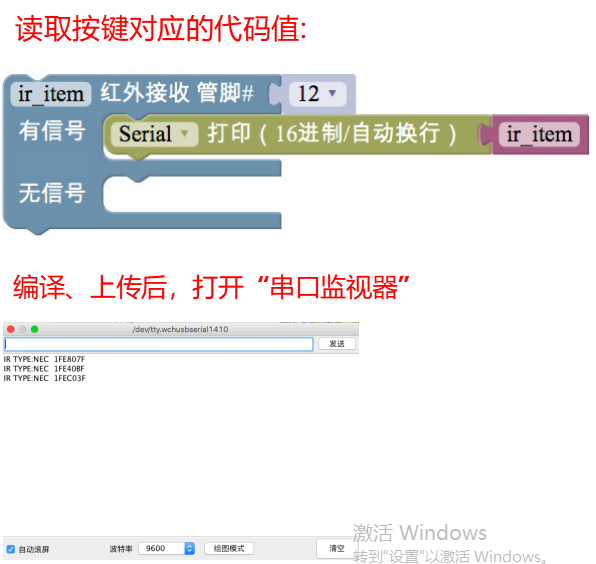


图4.10-1

### 任务1——制作一个红外遥控灯

#### 1、任务目标

通过红外遥控器发出红外信号控制灯的状态，当按下一个开关键，灯由点亮变为熄灭，再次按下，由熄灭变为点亮。

#### 2、流程图

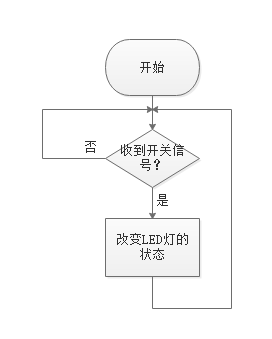


图4.10-2

#### 3、硬件连接

硬件连接：LED——10；红外遥控模块——12（主板自带）。注意插线时的颜色对应。

#### 4、程序编程

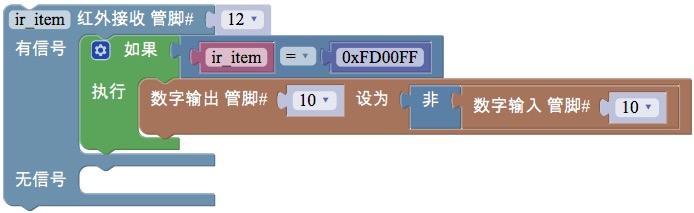


图4.10-3

#### 5、Q&A

**Q：是怎样接收遥控器发出的红外信号的？**

A：甜橙板上自带红外接收传感器模块，并且默认与D12管脚相连。

#### 6、拓展

**1、知识点总结**

1）“红外接收”程序块；

2）条件结构；

**2、相关案例**

## 项目十一 超声波测距仪

在上一个电子琴任务中，我们通过距离的变化，控制琴的音高。相应地，我们也能通过音高的变化，大致感受手与元件 的距离。不过，我们能不能直接看到距离是多少呢?怎样把测得的数字显示出来呢?这就需要一个液晶显示屏了。

液晶显示模块:每行16个字符，共2行若出现乱码，可在每次输出之前清屏。



图4.11-1



图4.11-2

### 任务1——超声波模块测量距离

#### 1、任务目标

将超声波传感器检测到的距离显示在LCD1602液晶显示屏上。

#### 2、流程图

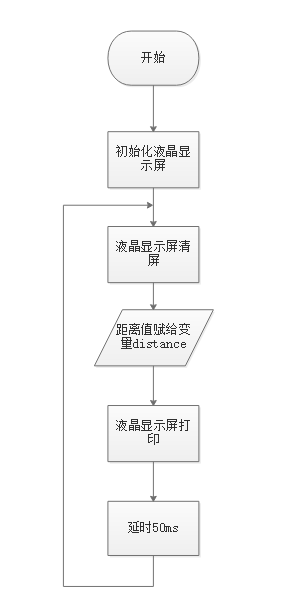


图4.11-3

#### 3、硬件连接

硬件连接：超声波——表1；液晶显示模块——表2。注意插线时的颜色对应。

**表1**

|  |  |
| --- | --- |
| 超声波 | 甜橙板 |
| VCC | 5V |
| Trig | 数字口2 |
| Echo | 数字口3 |
| GND | 地 |

**表2**

|  |  |
| --- | --- |
| LCD1602显示屏 | 甜橙板 |
| SDA | 模拟口A4 |
| SCL | 模拟口A5 |
| VCC | 5V |
| GND | GND |

#### 4、程序编程



图4.11-4

请你观察上面的程序是否能正确输出?如果不能，想一想应如何修改。

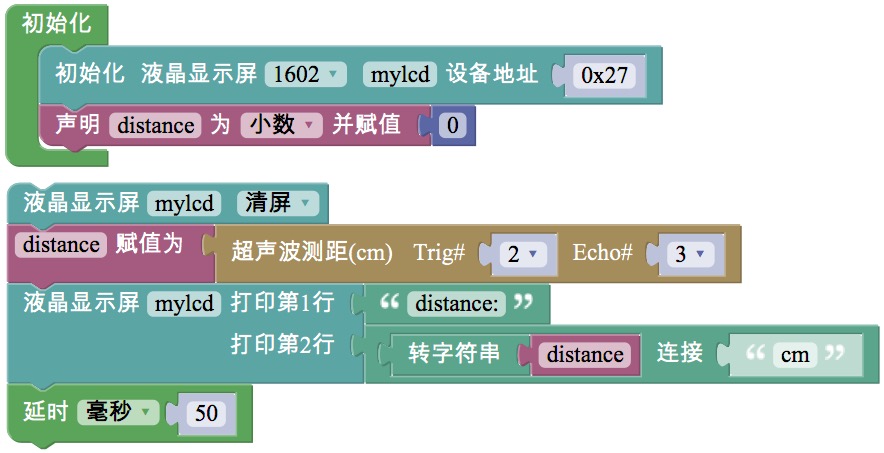


图4.11-5

#### 5、Q&A

**Q：显示屏上没有示数或者示数不变？**

A：没有将超声波测量的值赋给变量，首先需要声明变量，赋值变量在初始化程序块外面。

#### 6、拓展

**1、知识点总结**

1）“初始化”程序块中的程序仅执行一次；

2）LCD1602液晶显示屏的“清屏”程序块；

3）LCD1602液晶显示屏的打印；

4）变量的使用；

**2、相关案例**

1）项目十一的任务1；

## 项目十二 减速电机

### 任务1——实现电机的正反转

#### 1、任务目标

通过外接电源供电，程序控制实现HR8833减速电机正转和反转。

#### 2、流程图

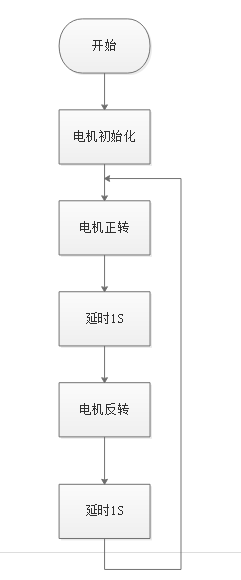


图4.12-1

#### 3、硬件连接

硬件连接：红色线---D4；黑色线---D5。

外接电源：DC接口连接外接的9V电源，并将甜橙板的电源从“off”拨到“on”。

#### 4、程序编程



图4.12-2

程序模块中的 left 和right 分别对应主控板上的 M1 和 M2 两个接线端子连接的电机，速度范围为-255-255，负值越小为电机反转速度越快，正值越大为电机正转速度越快，速度 0 为电机停止。



图4.12-3

#### 5、Q&A

**Q：程序上传成功后，电机并没有开始转动；**

A：检查外接电源的开关是否拨到“ON”,甜橙板外接电源处是否拨到“ON”端。

**Q：电机不转，但是一直发热，这是怎么回事？**

A：电机已损坏，或者是甜橙版的接线不良。

**Q：程序编译失败；**

A：未加“电机初始化”。

#### 6、拓展

**1、知识点总结**

1）HR8833的接线；

2）“电机初始化”程序块；

3）left电机对应M1的接口，Right电机对应M2两个接口；

4）外接电源供电；

**2、相关案例**

电机需要**外接电源**供电，用六节5号电池，DC接口；在程序上传成功后，外接电源盒，打开电池盒和甜橙版的电源开关，电机开始转动。

## 项目十三 蓝牙通信

蓝牙：是一种无线技术标准，可实现固定设备、移动设备和楼宇个人域网之间的短距离数据交换（使用2.4—2.485GHz的ISM波段的UHF无线电波）。

首先设置AT模式，参考硬件说明书第四章，4.6蓝牙模块。

### 任务1——蓝牙控制单色灯亮灭

实现功能：蓝牙主机发送的数据能控制蓝牙从机模块的单色灯。

#### 1、AT模式设置

在连接蓝牙的时候，要注意线的连接是否正确，检查清楚再进行通电。

1）Arduino HC05 的AT模式接线

|  |  |
| --- | --- |
| **蓝牙HC-05** | **Arduino** |
| TXD | 10(RX) |
| RXD | 11(TX) |
| VCC | V |
| GND | G |

2）Arduino 进入 AT 模式代码

接下来，我们需要为使用 Arduino 设置蓝牙模块 AT 模式编写程序，这个程序是让我们可以通过 Arduino IDE 提供的串口监视器来设置蓝牙模块。

Arduino IDE界面如图6.3-1所示：

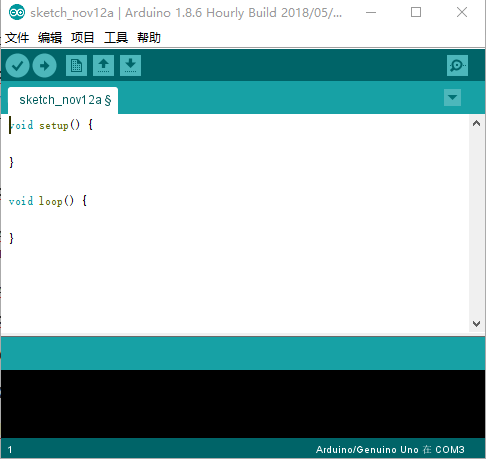


图4.13-1

详细的 Arduino 代码如下：

#include <SoftwareSerial.h>

SoftwareSerial BT(10, 11); // Pin10为RX，接HC05的TXD； Pin11为TX，接HC05的RXD

char val;

void setup() {

Serial.begin(38400);

Serial.println("BT is ready!");

BT.begin(38400); // HC-05默认，38400

}

void loop() {

if (Serial.available()) {

val = Serial.read();

BT.print(val);

}

if (BT.available()) {

val = BT.read();

Serial.print(val);

}

}

**3）利用 Arduino IDE 串口监视器进行调试**

首先，将 Arduino 断电，然后按着蓝牙模块上的复位按钮（reset键），再让 Arduino 通电，如果蓝牙模块指示灯按2秒的频率闪烁，表明蓝牙模块已经正确进入 AT 模式，此时松开黑色按钮。 打开 Arduino IDE 的串口监视器，选择正确的端口，将输出格式设置为 Both: NL & CR ，波特率设置为 38400 ，可以看到串口监视器中显示 BT is ready! 的信息。

然后，输入 AT ，如果一切正常，串口显示器会显示 OK。 接下来，我们即可对蓝牙模块进行设置，常用 AT 命令如下：（注意要英文输入）

|  |  |
| --- | --- |
| AT+ORGL | 恢复出厂模式 |
| AT+NAME=<Name> | 设置蓝牙名称 |
| AT+ROLE=0 | 设置蓝牙为从模式 |
| AT+ROLE=1 | 设置蓝牙为主模式 |
| AT+CMODE=1 | 设置蓝牙为任意设备连接模式 |
| AT+PSWD=<Pwd> | 设置蓝牙匹配密码 |

正常情况下，命令发送后，会返回 OK ，如果没有返回任何信息，请检查接线是否正确，蓝牙模块是否已经进入 AT 模式，如果上述两点都没有问题，可能是蓝牙模块的问题，可以找蓝牙模块供应商咨询。

设置完毕后，断开电源，再次通电，这时，蓝牙模块指示灯会快速闪烁，这表明蓝牙已经进入正常工作模式，两个模块会自动配对然后连接，状态灯会出现慢闪烁指示。

#### 2、蓝牙与PC间数据传输

如果要对蓝牙与PC间进行通信，需要将蓝牙设备在AT模式时设置为从模式，AT模式设置好之后蓝牙模块进入正常工作模式，将PC连接蓝牙模块连接。

准备两块甜橙主控板、两个HC-05蓝牙模块，一个单色LED灯，1根3pin线，两根4pin线，两根USB数据线。

**主机**：发送数据到从机。

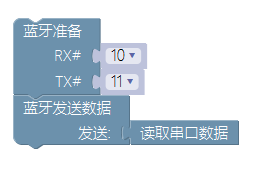


图4.13-2

**从机**：从主机接收数据，从串口监视器查看接收到的数据。

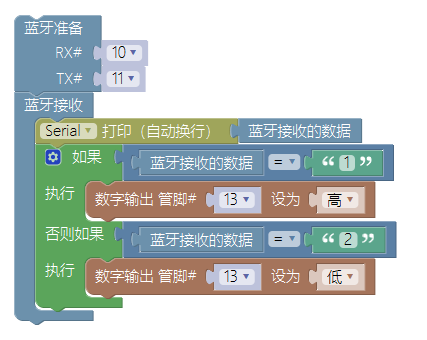


图4.13-3

注意，读取的数据类型为字符，需要从文本中，拖出“字符”程序块。

由主机向从机发送信息，并且打印在从机的串口，当从机收到“1”时，点亮与从机相连主控板的LED灯，当从机收到“2”时，熄灭LED灯。

#### 3、Q&A

**Q：如何确定蓝牙进入AT模式？怎么退出AT模式？**

A：蓝牙模块指示灯由2次/秒快速闪烁进入2秒/次的频率闪烁，表明蓝牙已进入AT模式。

设置完毕后，断开电源，再次通电，这时，蓝牙模块指示灯会快速闪烁，这表明蓝牙已经进入正常工作模式，两个模块会自动配对然后连接，状态灯会出现慢闪烁指示。

**Q：蓝牙配对完成后，并没有收到发送的信号？**

A：检查硬件接线，RX、TX是否已经交叉连接；查看蓝牙模块的工作状态，指示灯是否慢闪烁。

#### 4、拓展

**1、知识点总结**

1）蓝牙模块的接线；

2）蓝牙AT模式的设置；

3）蓝牙与PC的配对与连接；

## 项目十四 节日闪烁彩灯

### 任务1——WS2812 全彩 LED 灯

#### 1、任务目标

将三个全彩LED灯颜色设置依次是红黄绿。

#### 2、流程图

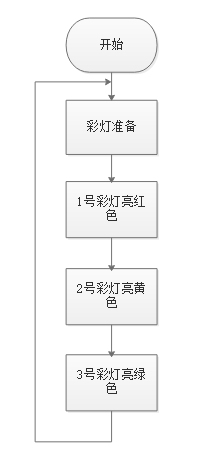


图4.14-1

#### 3、硬件连接

硬件连接：IN端控制灯颜色，OUT端连接下一个全彩LED灯；D3连接1号彩灯的IN控制端。

注：观察单个全彩灯模块，有电容的一角为IN，也可以看灯壳，缺角的一侧为IN端，另一端为OUT。

#### 4、程序编程

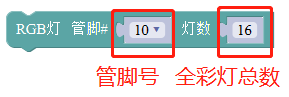


图4.14-2

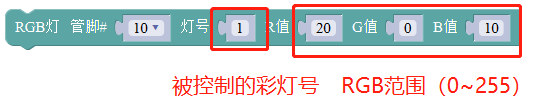


图4.14-3

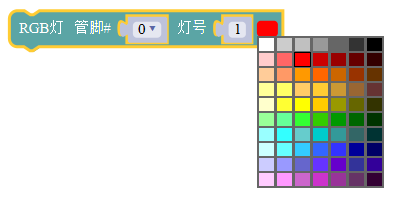


图4.14-4



图4.14-5

#### 5、Q&A

**Q：程序上传成功后，灯不亮？**

A：查看甜橙版是否连接的是全彩灯的IN端，第二个全彩灯的IN端。

#### 6、拓展

**1、知识点总结**

1）全彩灯的接线，控制端为IN，OUT端与下一个全彩灯的IN连接；

2）全彩灯的控制，灯号没有重复；

3）全彩灯阵的控制，可以通过单个的全彩灯来控制；

全彩灯阵的控制，IN端在左端时，左上角第一个灯为1号灯，其右方为2号灯，第二行第一个为5号灯。

**2、相关案例**

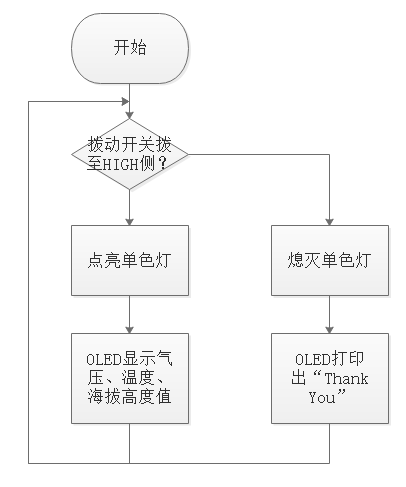
## 项目十五 DIY海盗船系列

### 任务1——气压海拔检测

#### 任务目标

拨动开关HIGH侧，单色灯点亮，气压传感器检测此时此地的海拔高度、温度和气压，OLED屏幕显示。松开触摸开关后，OLED屏幕显示“Thank You”，单色灯熄灭。

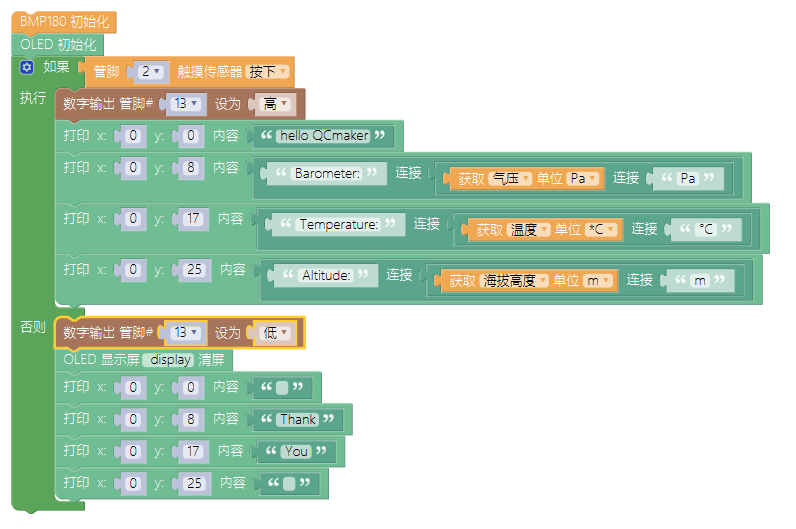
#### 流程图



#### 硬件连接

气压BMP180与OLED屏幕为IIC通信，通过IIC扩展板与A4、A5相连；拨动开关—2；单色灯—13。

#### 编程



#### 5、Q&A

**Q：为什么OLED显示的数据一直闪烁？**

A：OLED清屏 程序块目的是清除命令行的所以显示内容，所以会清除一些数据的内容。

#### 6、拓展

**1、气压传感器有什么用呢？**

气压传感器应用广泛 它可以用于手机GPS测海拔高度。

BMP180气压传感器（型号GY-521）它测量周围空气的绝对压力。它的测量范围为300hPa至1100hPa，精度低至0.02hPa。它还可以测量高度和温度。BMP180气压传感器通过I2C接口进行通信。这意味着它仅使用2个引脚与Arduino通信。

气压传感器首次在智能手机上使用是在Galaxy Nexus上，而之后推出的一些Android旗舰手机里也包含了这一传感器，像Galaxy SIII、Galaxy Note2也都有。

对于喜欢登山的人来说，都会非常关心自己所处的高度。海拔高度的测量方法，一般常用的有2种方式，一是通过GPS全球定位系统来计算表海拔，但是由于存在十米左右的较大误差，以及GPS卫星信号接收不能够保障等问题，二是通过测出大气压，然后根据气压值计算出海拔高度。气压的方式可选择的范围也更广些，而且可以把成本可以控制在比较低的水平。另外像Galaxy Nexus等手机的气压传感器还包括温度传感器，它可以捕捉到温度来对结果进行修正，以增加测量结果的精度。所以在手机原有GPS的基础上再增加气压传感器的功能，可以让的三维定位更加精准。

如果说用压力传感器来计算海拔算是一项不错的应用，那么利用压力传感器来辅助导航，你是不是会觉得惊讶呢？由于导航仪市场较为混乱，产品质量良莠不齐，因此经常会出现导航仪瞎指挥的现状。如你在高架桥上时GPS却可能会指挥你转弯，但其实并没有转弯出口。这往往是由于GPS存在误差，不能够判断车子在高架桥上还是桥下所致。但如果再加上气压传感器，测量出所处的高度，就能够将误差降低到1米左右，随着精度提升导航也将变得更加精确。

同时当用户处于楼宇内时，内置感应器可能会无法接收到GPS信号，从而不能够识别地理位置。配合气压传感器、加速计、陀螺仪等就能够实现精确定位。这样当你在商场购物时，你能够更好找到目标商品。

1. **知识点总结**
2. IIC通信的原理与管脚连接；
3. OLED屏幕程序块的使用；
4. 气压传感器BMP180的使用；
5. 文本程序块的使用；
6. 触摸传感器程序块与数字输入程序块使用方法一样；
7. OLED屏幕显示的位置，X\Y的范围，Y为（0-32）。
8. 数字输出程序块的使用。
9. **相关案例**

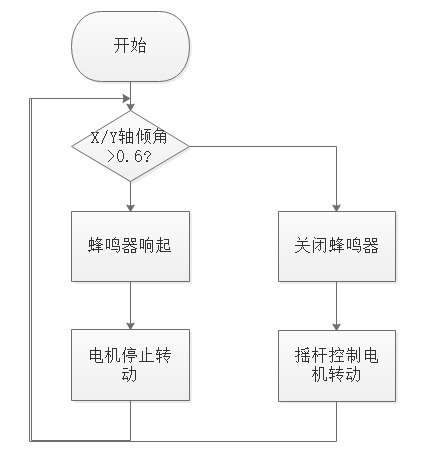
见任务2

### 任务2——车辆测速与倒地报警仪

#### 任务目标

使用摇杆控制车辆的运动，X轴方向控制发动机前进和后退，Y轴控制电机2前进和后退；加速度传感器检测车辆的三轴加速度和角速度，当X方向或Y方向的倾角大于60度时，蜂鸣器倒地报警并且车辆停止。

#### 流程图

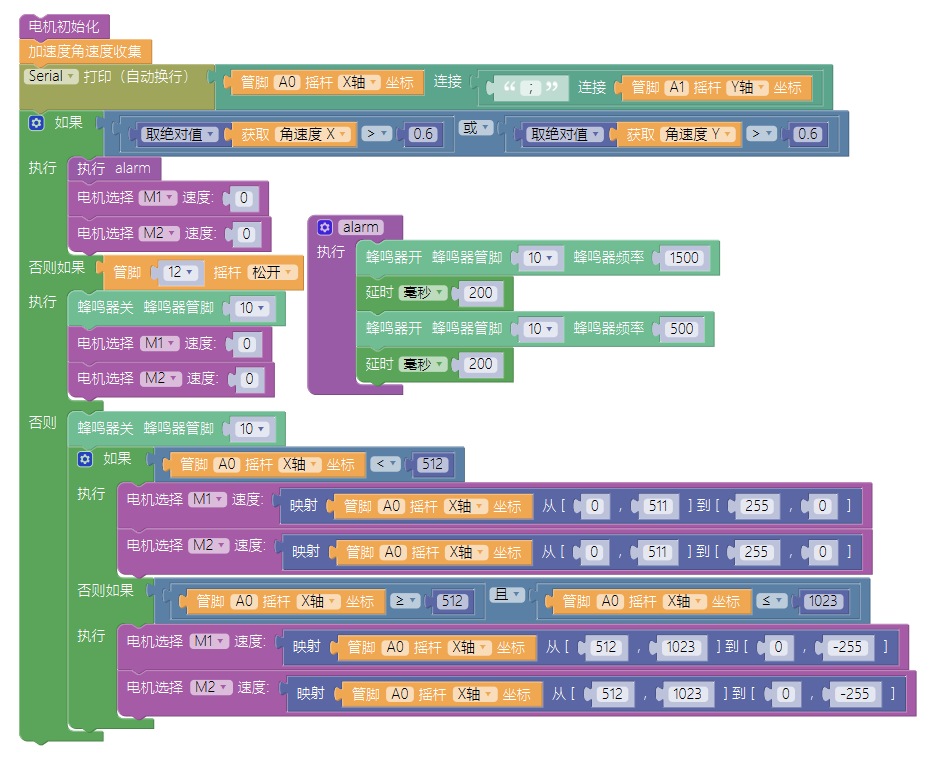


#### 硬件连接

首先准备好硬件：1个加速度传感器、1个蜂鸣器、2个N20电机、1个摇杆模块、1块7.4V锂电池、2个IIC扩展板、两根2pin线、6根3pin线和两根4PIN线；一根USB数据线、一个主控板。

加速度模块通过IIC扩展板与主控板的A4、A5相连；摇杆模块X轴与A0相连，Z轴与12号管脚相连；蜂鸣器与10号管脚相连；两个N20电机与4、5和6、7相连；7.4V锂电池与主控板的DC电源接口相连。

#### 编程



#### Q&A

**Q：为什么不用X轴和Y轴分别控制小车的前进和后退？**

A：因为操作摇杆时，时常会同时改变XY轴的值。而小车不能同时既前进又后退，又因为摇杆静止时，默认XY轴方向的值均是512左右，向右与向左可以分为两个动作，能达到控制小车方向的目的。

#### 拓展

**1、知识拓展**

MPU6050是一种非常流行的空间运动传感器芯片，可以获取器件当前的三个加速度分量和三个旋转角速度。由于其体积小巧，功能强大，精度较高，不仅被广泛应用于工业，同时也是航模爱好者的神器，被安装在各类飞行器上驰骋蓝天。

加速度传感器已被广泛应用于汽车电子领域，主要集中在车身操控、安全系统和导航。

广泛应用于游戏控制、手柄振动和摇晃、汽车制动启动检测、地震检测、工程测振、地质勘探、振动测试与分析以及安全保卫振动侦察等多种领域。

在UNO板子上，SDA接口对应的是A4引脚，SCL对应的是A5引脚。MPU6050需要5V的电源，可由UNO板直接供电。

该摇杆模块有 3 路输出端口，其中两路为 x 轴和 y 轴的模拟量电压信号输出端口，一路为按键的数字电平信号输出端口。摇杆模块的 x/y 轴由电位器组成，在没有外力作用时，摇杆会自动回中，当摇杆转到不同位置时，x/y 轴对应输出信号的大小也不同，主控板通过模拟输入端口读取摇杆模块 x/y 输出端口的值，便可知道摇杆当前位置。

**2、知识点总结**

1）摇杆的使用

2）加速度传感器MPU6050的使用；

3）蜂鸣器的使用

4）电机的使用

5）函数的定义与使用

**3、相关案例**

见任务1