界面

Doany 介绍

Doany 是一款基于 Scratch 2.0 开发的图形化编程软件,它拥有简单的程序编写方式,可以兼容 Arduino 的主板并加以控制。

😚 Doany(v1.0.0) - 没有连接串口 - 未保存			
文件 编辑 连接 控制版 扩展 语言 帮助	× 1	省令方块区	
🖃 Untitled 🎽 🍋	脚本 造型 声音	1 + X X	BH4K
Ret	动作 毎件 外況 独樹 声音 (公測 副考 数字和正磁量正算 改画和指令 机器人提块 移動 10 步 (○古磁转 へ 15 度) 向左磁转 へ 15 度 (○方向) 面向 600 方向 (□	 说 你好! 2 秒 将器型切线力 Rabbit 将背景切线为 皆景1 * 	
	移到 X: 66 Y: 23 移到 成府指計 V 在 1 秒内滑行到 X: 66 Y: 22 将X坐标磁加 10 将X坐标磁加 10 将X坐标磁加 10	根木区	Q = Q



图形化编程方式

Doany 的编程方式非常简单,只需将积木拖拽到脚本区即可:

😚 Doany(v1.0.0) - 没有连接串口 - 未保存	0. 1. 4	and the second s	
文件 编辑 连接 控制板 扩展 语言 帮助			
📜 Untitled 🍋 🍋	脚本 造型 声音	± 4 ⋈ X	
	动作 事件 介潤 控制 方言 公園 回答 数字印逻辑运算 数面印旨令 机器人模块 移动 10 岁 0 向右旋转 10 5 度 向左旋转 2 15 度 向左旋转 2 15 度 面向 20 方向 面向 1 移到 x: 167 y: 458	修助 10 步	\$ x 167 y -68
x: 240 y: -75 角色 新建角色: ◆ / 企 @	移到 段标指計 在 ① 砂内滑行到 x: 167 y: -68 将x坐标滑加 ① 将x坐标滑加 ① 将y坐标滑加 10		Q = Q

然后通过积木之间相互卡合的方式来实现程序块之间的连接:

🛞 Doany(v1.0.0) - 没有连接串口 - 未保存		
文件 编辑 连接 控制版 扩展 语言 帮助		
📜 Untitled 🔪 🏲 🔴	脚本 造型 声音	1 + X X
	功作 事件 分苑 空樹 声音 (公規) 回答 数字和思辑运算 数面和目前令 机器人機块 移動 10 步 の右旋转 (* 15 度) 向右旋转 (* 15 度) 原 向左旋转 (* 15 度) 原 面向 90*万向 面向 • 移到 xx 157 yx 659	ید 167 ۲68 بوان و اور اور اور اور اور اور اور اور اور ا
x: 240 y:-8 ● 角色 新建角色: ◆ / ④ ④ 写合 習合 調合 調合 調合 調合 通合	 移到 税标溢計 ▼ 在 1 秒内滑行到 x: 167 y: 68 将x坐标着加 10 将x坐标燈定为 () 将y坐标着加 10 	Q = Q

积木的形状



触发式积木:任何脚本的第一块积木由事件触发执行 命令积木:执行相关的命令 参数模块:不能单独使用,需要将其放入其他积木中 C形积木:内部可以卡和其他模块,实现其特有功能

参数的形状与数据类型



第一块积木中的圆角矩形里的参数在设置时只能放入数字,不能放入其他参数。

第二块积木中的六边形框内只能放形状为六边形的积木模块。六边形积木返回的布尔值,就 是真和假。

第三块积木中的矩形框内的参数既可以放数字也可以放字符串(即数字,字母,字符)。

Doany 的程序结构

Doany 程序要么是纯软件的形式让很多角色在小舞台里面互动,要么是通过脚本以硬件的形式实现其功能,或者两种都有。一个 Doany 程序可以拥有众多的角色,每一个都拥有其自己独特的脚本、声音;每个程序还可以设置一个背景,背景也可拥有声音。脚本是积木的结合,是实现每一个程序的核心,因为它实现了硬件和软件的逻辑。

角色与造型的创建

Doany 的角色与造型的创建既可用导入图片的形式,也可根据自己的想象绘制。GIF 文件包含许多帧,我们既可以直接将其作为角色也可以将这些帧导入到造型中:

	角色	新建角色:	🔶 / 🖆 🔯 👘
舞台 1 背景 新建背景 四 / 스 6	Rabbit		从本地文件中上传角色
脚本造	型声音	1 + X X	
新建造型	Rabbit-a	•	清除 导入
从本地文件中	上传造型		
Rabbit-a 163x224	J		
还可以根	据自己的想法绘制造	ī型:	

新建造型	造型1	19 (°	清除 导入	1-60
绘制新造型				<u>ک</u>
诰型1				
1x1				
				O
				▲
				L

动作

动作指令区

角色

Rabbit

舞台 1 背景

下面红框区域就是 Doany 的动作指令:

😚 Doany(v1.0.0) - 没有连接串口 - 未保存	0 0 C man		
文件编辑 连接 控制板 扩展 语言 帮助			
🔄 Untitled 🎽 🏲 🔴	脚本 造型 声音	white a constant and a constant	
	动作事件		ES
	外观 控制		¥
	回笔 数字和逻辑运算		x: 5 y: 22
	数据和指令 机器人模块		
ρ_{\frown}	移动 10 步		
	向右旋转 (* 15 度	and the state of the state of the state	
(*_ *)	向左旋转 り 15 度		
	面向 907 方向		
e	面向一		
	移到 x: 5 y: 22		
x: 240 y: -74	移到 鼠标指针 🖌		
	在 1 秒內滑行到 x: 5 y: 22		
舞台 1 苔目	将X坐标增加 10		
新建料果	将x坐标设定为 ①		Q = Q
	将y坐标增加 10		
😚 Doany(v1.0.0) - 没有连接串口 - 未保存	NY-1-B Same And	and a state	
文件 编辑 连接 控制板 扩展 语言 帮助			
🖃 Untitled 🎽 🦰 🧶	脚本道型声音	T K X H	
	动作 事件		
\mathbf{O} =	71% 52m (近辺) 72m (近辺)		ж -3
	画笔 数字和逻辑运算		y: -18
6.	数据和指令 机器人模块		
	在 1 秒内滑行到 x: -3 y: -18		
The second se			
8	将x坐标增加 10		
	将x坐标设定为 🕕		
	将y坐标增加 10		
	挖		

通过上面这些指令, Doany 会做出与之相对应的动作, 比如移动十步, 移动到 XY(-3, -18) 所在位置。

设定为 左-右翻转 🔻

Q = **Q**

y坐标 方向

x: 240 y: -143 新建角色: � / 🏜 🙆



选择不同的方向,舞台中的角色就会根据你的选择实现面朝哪边,向哪边旋转:

旋转模式

角色的旋转模式有三种,可以在两个地方进行设置,第一处是在角色列表中,用鼠标点 击角色左上角的感叹号,就可以选择不同的旋转模式:

	Rabbit
Rabbit	x: -16 y: 17 方向: 90° 旋转模式: ♥ ↔ ● 可以在播放器中拖动: ■ 显示: ■
另外一处就是使用积木块了	
将旋转模式证	設定为 左-右翻转 左-右翻转 不旋转 任意

它的旋转模式含义:

- 左-右翻转就是角色只能左右翻转,即 90 和-90 度。
- 不旋转就是无论给角色设定什么值,角色的方向不做任何变化,始终面朝 90 度方向。
- 任意即可以做任何方向的旋转。



一个角色的每一个造型都有一个中心点,在造型页面中可以进行打开并设置:

角色的移动

角色有许多种的移动方式,常见的有:水平位置随机运动,垂直位置随机运动,移动到 舞台任意位置:





通过以上按键的方法控制角色时无法实现组合按键(即上移和右移一起按下)且会有一 点延时,如有需要,可以参考下面的方式进行移动。

重复执	行 · · · · · · · · · · · · · · · · · · ·
如果	< 按键 上移鍵 ▼ 是否按下? → 那
×	Y坐标增加 10
如果	
×	y坐标增加 -10
如果	★ 按键 左移键 ▼ 是否按下? 那
×	x坐标增加 -20
如果	★按键 右移键 ▼ 是否按下? 那
米	x坐标增加 20

外观

外观指令区

下图中红色区域就是要介绍的外观指令:



说和思考



直说你好,要将内容清除就在矩形框内将其内容删除执行就好了,而 积木将会在规定时间后自动清除效果。

角色的放大和缩小

改变角色的方法有以下三种:

• 造型里面的绘图编辑器

• 工具栏上的 🚽 🔨 🔯 🔣 图标,使用会改变角色中所有造型的大小

• 使用积木进行放大和缩小

第一种方法:首先看造型处于何种方式,造型处于矢量模式,就选用选择工具,在点击 矢量图即可;若造型处于位图模式,也选择使用工具,选中希望改变大小的区域即可。



第二种方法:用鼠标点击选择工具栏上的放力	、和缩小选项,	再点击对应	的角色即可:
文件编辑 连接 控制板 扩展 语言 帮助			
📃 Untitled 🎽 🖊 🖊	脚本 造型 声音		1 ≁ ⊠ <u>×</u>
	动作	事件	
	外观	控制	
	声音	侦测	
	画笔	数字和逻辑运算	
\sim	数据和指令	机器人模块	
	说 你好! 2 秒		19 No 19
	10 Jatz 1		
X	W DIRT:		
	思考 呃 2 秒		
S S	思考 呃		
G			
	显示		
第 三 种 方 法 · 前 是 使 田 外观	植块里的	将角色的大	小增加 10 和
	一次八王的		14
将角色的大小设定为 100			
就可以在脚本里动态	的实现改变造	型的大小,设	と定参数 100 为
-			
原始大小,参数 150 就是原来的 15 倍。使用	本小 积木	可以在程序中	口动态的获得角
		4 9 (PE / LE / 1	
色当前的大小。			

造型与动画

每一个角色都至少有一个造型,在任意时刻都要处于一个造型之中。与造型相关的积木 如下所示:



第一个积木可以直接切换到自己想要的造型,第二个积木会自动切换到下个造型,如果 当前处于最后一个造型的话,那么自动切换到第一个造型,第三个积木则会告诉你当前处于 第几个造型。

角色的造型可以在造型里面添加,修改,删除:



我们可以制作一个简单的动画来展示效果:



背景与场景

任何角色,包括舞台本身都可以切换背景,相关积木如下:



第一个积木可以切换到具体的背景;第二个积木可以获得当前背景的名称。 舞台的背景往往是和背景有关的,下面演示一下简单的场景切换,首先要从素材库中添加两 个背景,也可以自己绘制,然后再编写脚本。



脚本背景	声音	
新建背景	castle3	
1		
背景1 480x360		
castle3 480x359		
		将背景切换为 castle3 ▼ 说 出去玩累了,准备回家睡觉 2 秒
bedroom2 480x360		将背景切换为 bedroom2 ▼

图形特效

图形特效是为了让角色更加的生动,我们可以更改外观上的效果,一些相关的积木如下:

清除所有图形特效									
将	颜	色 🔻	特效增加	10 25					
		将	颜色▼	特效设定	≣为 🖸				
			颜色						
		1	超广	角镜头					
			旋转						
		÷.	像素	滤镜					
		2	马赛	克	1.1				
		2	亮度		1.1				
		12	虚像		1.1				
			-						

将颜色特效设定值改为0就是还原到最初的效果,也可以使用"清除所有图形特效"积 木,就可以还原。各个效果如下图所示:

























当有多个角色在一起时,就会发生重叠,这就涉及到了谁在谁上面的情况:



声音

外观指令区



下图中的红框区域就是相对应的声音指令:

每一个角色都拥有声音,你可以在声音页面里自己录制声音,导入外部声音或者从库中 选取声音都可以:



个积木的差别在于第一个积木播放声音后会进行下个积木的内容,而第二个积木要等到声音 播放完毕后才能进入下一个积木。我们可以动手通过下面的例子来说明:



如上面的脚本的效果可以看出第一个脚本的效果是一遍播放声音一边移动,第二个脚本 的效果是声音播放完毕后才开始移动。



乐器

循环音乐 打击乐 声乐 birthday bells

ut D

i (÷ D

dance chill out

bubbles

computer beeps2 cough-female cough-male

i (÷ 🕞

buzz whir

I(€) **●**

吉咅库									×
/→ ロ/ + 分类									
全部									Ш
动物 效果	arro string	allen creak1	allen creakz	Deat DOX1	beat box2	bell toll	bira	birthday	Ш
电子声	u(÷ D	I ({ €	u(€)⊳	u(÷ D	iii (÷)⊳	u(÷)⊳	u(€)⊳	III € ●	

car passing

I(;))

cricket

ut D

dance funky dance head nod dance slow mo dance snare beat dance space

chomp

I({ €

cymbal

i (÷ 🗩

cave

I({ €

crickets

I({ €

chord

i (÷ D

dog1

computer beeps1

I({ €

dog2

确定 取消

cymbal crash dance celebrate

除了一些自己制作的声音外,还可以从库中调取,来制作适合程序的音乐:

制作乐曲

制作一个乐曲除了需要设置音符和节拍外,还需要考虑音量的大小与速度,使用的音色等:



设置弹奏有音符的音色,设置停止音符的时间



通过下面脚本了解 bpm:





当 bpm 为 120, 1 分钟演奏 120 拍, 4 拍时间为 2S

下面动手制作粉刷匠的主奏:



画笔

外观指令区

下图中红色区域就是要介绍的画笔指令:

😤 Doany(v1.0.0) - 没有连接串口 - 未保存	T PACE DES	
文件 编辑 连接 控制板 扩展 语言 帮助		
📃 Untitled 🎽 🖊	脚本 造型 声音	サスズ
R	动作 事件 分況 控制 声音 仮則 三 章 奴判 三 章 数学和逻辑运算 政振和指令 机器人模块	ې ۲ - 26 ۲ - 26
	濟立 回幸 家ぞ	
x: 240 yr - 133	拍笔 将圈笔的颜色设定为 将画笔的颜色慎增加 10	
角色 新建角色: ◆ / ▲ O 新建角色: ◆ / ▲ O 目台 3 預泉	将國笔的颜色设定为 ① 将國笔的色度增加 ① 将画笔的色度设定为 50	
#### ☑ / ▲ @	将画笔的大小增加 1 将画笔的大小设定为 1	Q = Q

绘图功能

将画笔的颜色设定为 🕕

每个角色都自带了一个隐形的画笔, 画笔的笔尖就是改造型的中心点, 而画笔只有两种



将画笔的色度设定为 50

将画笔的大小设定为 1

以上的积木,通过改变这些属性我们可以绘制出非常漂亮的图案:



											2
当	■ 被占	÷									3
面同	90 万	回									
将画望	笔的大小	设定为	1								
将画望	笔的颜色	设定为									
络丽谷	的品度	设定为	50								
	CHICIS	KX AE 79	50								
落笔	14 - 14 - 14 - 14 - 14 - 14 - 14 - 14 -										
清空											
重复排	机行 4	次									
将回	画笔的色	度增加	10								
将匪	。 画笔的大	小设定	为在	1	到(10 阔	酮随机	ī选-	-个	数	
移ž	动 200	步			1						
向	左旋转 ▶	90	度								
	· · · ·	<u></u>	1 50 为在1到10间随机选一个数 度								

图章

图章和画笔一样,都可以在舞台中留下印记,不同的是,图章留下来的是角色当前造型 本身,而不是一条线,并且不需要执行落笔和抬笔的操作,因为图章积木本身就包含了落笔 和抬笔的操作,我们可以通过例子来了解图章的这个功能:



通过上图可以看出,图章就是复制了角色,如果使用画笔比较麻烦的话,可以先在编辑器中将角色绘制好,然后在使用图章积木进行绘制。

数据和指令

外观指令区

下图中红色区域就是要介绍的数据和指令:

2件 编辑 连接 控制板 扩展 语言 帮助		
📃 Untitled	▶ ● 脚本 造型 声音	サスズ
■」 Untitled		

变量

变量是什么?简单的说,我们可以把变量看做是一个盒子,可以将一些物品放在盒子中。



新致	建变量
变量名:	
● 适用于所有角色	○ 仅适用于当前角色
确定	取消

在弹出的窗口中可以添加需要的变量名,其中"适用于所有角色"和"仅适用于当前角 色"的区别在于前者的变量是所有角色所通用的,即全局变量。而后者仅仅可以在当前角色 内部中使用,并且也可以作为克隆体的变量使用,即局部变量。还有一个区别则在于重命名 的问题,全局变量的命名是唯一的,不能重复,而局部变量是在角色内部唯一的,其他的角 色可以拥有相同的名称。

再输入变量名确定后,积木区就会出现新的积木,同时舞台上出现"变量值显示器":



"将变量设定为"积木可以给变量一个初始的值,"将变量的值增加"积木可以对变量 增加或减少一定的数值,最后"显示和隐藏变量"就是将舞台中的变量显示器进行隐藏和显示。

舞台上的变量显示器有三种展示的方式:



默认的就是正常尺寸,有时候会调成大尺寸,也会调成滑杆,这样对程序的交互会有很大的帮助,在滑杆模式下还可以点击右键进行范围的调整,默认范围是{0,100},还可以显示小数形式:

正常尺寸 大尺寸	滑杆值范围
净杆	最小:
设置滑块最大最小值	最大:
隐藏	4070
	确定取消

变量名也可以进行操作,不需要了可以将它删掉,名字也可以进行修改,右键变量名可以进行修改和删除。

一个变量对程序的使用会很重要,使用变量的次数会很频繁,例如记录步数,时间,进行一些运算啊,可以看下面的一些程序对于变量的使用:



链表

链表是变量的容器,每个变量在容器中都有一个唯一的编号。第一个容器中的变量,第 二个是 2 号变量,第三个是 3 号变量,以此往后类推,如果现在 1 号变量不需要,将它删除, 那么2号变量就自动变成1号变量,3号变成2号变量,以此往上类推。我们都知道变量的 引用方法是通过变量名,而链表中变量的引用就是通过唯一的编号。换言之,链表可以在程 序运行时动态的管理变量。

	新建链表
	链表名称:
	● 适用于所有角色 ○ 仅适用于当前角色
鼠标点击新建链表 新建链表 新建链表:	确定取消

其中"适用于所有角色"和"仅适用于当前角色"的区别跟"变量"是一样的。添加过后积 木区会出现相关积木,舞台模块出现链表值显示器:

新建链表	
south	
将 thing 加到链表 south ▼ 末尾	
删除第 1™ 项,从链表: south ▼	
插入: thing 位置: 1 到链表: south T	
替换位置: 1▼ 链表: south ▼ 内容: thing	south
第 1Y项 south ▼	
链表 south ▼ 的长度	
south ▼ 包含 thing ?	(empty)
显示链表 south ▼	
隐藏链表 south ▼	+ 长度:0

"将...加到链表末尾"积木直接将变量添加到容器的末尾,"删除第 n 项"积木除了可以删

刪除第	17顶,	从链表:	s	outh	n *	
	1			1.1		
	末尾	100				
	全部					

除某一项之外,还可以清空整个链表的变量:

"链表的长度"积木可以得到容器中变量的个数,"包含"积木可以检测容器中是否包含 某些变量。其他积木块都很好理解,就不在一一描述了。链表值显示器支持导入导出文本文

件,每行代表一个变量:



添加和删除链表除了使用积木添加变量,还可以使用链表值显示器中链表功能非常强大,可以说是高级程序中必不可少的要素,即使程序关闭链表中的数据也不会被删除,因此完成 存档的功能,使数据不会丢失。

新建模块指令

新建功能块是一个非常重要的功能,不仅因为其功能强大,更重要的是它的一种编程思想就是模块化,这种思维方式不仅在编程中有体现,而且生活中也很常见,如我们做事一样会有一定的计划,分为几个步骤,每个步骤就是一个模块。不仅可以将脚本分为模块化,也可以将相同的脚本整合在一起。下面让我们来建立一个模块,点击新建模块指令会出现如下对话框:

	新建模块指令	
 □ □	确定取消	

每一个模块需要一个名称来说明它的主要作用,像绘制五角星,输入文字,点击确定后, 在脚本区和积木区都会出现相对应的模块:



这样我们就可以直接使用并绘制正方形了,但是提前还需要完成"定义"部分:



这样就绘制成功了,如何更加智能的绘制两个不同颜色的正方形呢,我们还可以使用参数,右键"绘制五角星积木"或者点击"定义绘制五角星积木"选择编辑,打开选项,添加一个数字参数,将参数修改为"颜色",最后点击确定:



这时软件中的积木就会发生对应的变化:



这时我们只需要将程序修改为下图所以就可以:



还可以根据自己的需要在添加一些模块,方法跟上面介绍的一样,就不在操作了。下面 介绍一下"运行时不刷新屏幕"的作用,所谓的不刷新屏幕就是指舞台直接展现出每个模块 运行到最后的效果,下面举例说明一下:

	编辑功能块		
绘制	区 颜色 颜色 ●		
▼选项			
	添加一个数字参数:		
	添加一个字符串参数:		
	添加一个布尔参数:		
	添加文本标签:	text	定义移动
	■ 运行时不刷新屏幕	1	重复执行 10 次
	确定取消		移动 10 步

运行移动模块,此角色不会依次向前移动,而是直接移动到最终位置,如果某个模块运 行时间过长,就可以使用此选项。

事件

何为事件

所谓的事件是指被动的接受由外部触发的事件,这是一种接近人类的思维方式。例如, 大多数人都是通过一大早起床看天气穿什么衣服,下雨天带伞,天气就是一个外部事件,导 致我们是否带伞,穿什么衣服的这种行为。就是被动的接受一个事件并做出行为。生活中有 很多事情都是被动事件。那么在我们 Doany 软件中,事件是怎么样的,在软件中事件都是 由一块块上方半圆形的积木构成:

1 Doany(v1.0.0) - 没有连接串口 - 未保存	SW-7-8 monthalt	4 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
文件 编辑 连接 控制板 扩展 语言 帮助			
📃 Untitled 🍋 🕨	脚本造型声音	L + X X	
K: 240 y: 120 商商 新建角色: ◆ / ④ @	 助作 9件 外规 控制 控制 位列 团站 政步和逻辑运算 机器人模块 当 / ● 秋点击 当将是、初块到 發展1 * 当 / ● 印度 * > 10		£ x 4 y: 49
智 舞台 Rabbit	自接收到 message1 ▼		
	广播 message1 ▼ 并等待		Q = Q

上图中的这些积木都是外部事件,因此他们都是脚本的第一个起始积木,由他们作为触 发执行脚本,比如,当绿旗被击中时,它的外部事件就是舞台上的小绿旗,一旦点击就触发 这块积木,"当按下"积木,它的外部条件就是用户按下了键盘上的某个键,来使脚本运行, 鉴于事件的含义比较好理解和掌握,这里就介绍两三个积木,其他的积木性质都是一样的, 使用方法也差不多。

广播

广播我相信大家都知道,就是一个角色把一个消息传递给每一个角色,让别人都知道, 当别人接收到这条消息时,就会做出相应的反应。因此,广播就包含了两个部分,发送消息 和接收消息。下面通过角色对话的例子来给大家介绍下这两个阶段:



后就会执行下面积木的程序,而后者在广播之后会等待所有人接收到消息后在执行下面积 木的程序。如果你写的程序需要严格的协调,同步角色间的行为,则必须使用后者,如果 没严格要求,使用前者积木就好。

控制

控制指令区

下图中的红框区域就是下面要介绍的控制指令区:





分之结构

分支结构涉及到的积木如下所示:



如果结构是我们编程过程中必不可少的一部分,非

常重要,因为它是一种用来判断的结构,如果这样做,否则那样做。第一块积木为单分支结构,第二块积木为双分支结构,两者之间还可以进行组合,形成多分支结构,如下图所示:



循环的使用



循环的积木有三个:

分别介绍下,第一块积木是无限重复,专业的叫法叫做死循环,通常用在持续的检测某些事件,注意这块积木的下方是平的,说明它的下面不能连接任何积木。第二块积木是次数重复,适用于知道次数的情况下进行重复执行,第三块积木是直到型循环,或者说是有条件的重复,适用于知道某个条件去不知道次数的情况下使用。下面介绍一些例子来说明这三种循环模块





停止

有的时候,程序运行过程中需要停止,就要用到停止模块:



停止模块它有三个选项, 第一个选项就是停止整个项目的全部

程序,第二个选项就是停止当前积木的脚本,第三个选项就是停止本段脚本之外的所有脚本。

克隆

有的时候,在程序运行时如要复制角色,可以在角色列表或者舞台中右键进行复制,但 是这种做法是在程序运行之前就确定数量的,如果在这之前,复制的数量无法确定时,那么





每当击中绿旗后,先隐藏自己,然后重复的移动到某个随机的位置,克隆自己。

	х: y:
当作为克隆体启动时	
显示	
里夏执行	
将y坐标增加 -10	
如果 碰到 边缘 了 那么	
删除本克隆体	

当每个克隆体启动时,首先将自身显示出来,然后不断地减少 y 轴下降的过程,同时判断是否碰到边缘,碰到边缘就删除克隆体。

侦测

控制指令区

下图中的红框区域就是下面要介绍的侦测指令区:

1 Doany(v1.0.0) - 没有连接串口 - 未保存	D-BBW-F-B manufact	A AND	
文件 编辑 连接 控制板 扩展 语言 帮助			
🗔 Untitled	脚本 造型 声音	サイズズ	
	动作事件		67
	外观 控制		53
0	PE UXX 回笔 数字和逻辑运算		x: -64 y: 23
	数据和指令 机器人模块		
6.5	/ 術到 ▼ ?)		
	磁到颜色 2		
4	施田 ● 従利 ■ 2		
6 62 8	到了的距离		
	询问 你叫什么名字 并等待		
		instantin an an an an an an an an	
	按键 空格键 ▼ 是否按下?		
▲ 新建角色: ◆ /			
	鼠标的×坐标		
1. Alexandre and the second se	鼠标的y坐标		
舞台 Rabbit			
上问题	49.18	the top of all all all all all all	
₩/ ± 0	□ 视频侦测 动作 ▼ 在 角色 ▼ 上		Q = Q
1 Boany(v1.0.0) - 没有连接串口 - 未保存	DOBERT OF LOUGH	S. S. Lawson	
⑦ Doany(v1.0.0) - 没有连接串□ - 未保存 文件 编辑 连接 控制版 扩展 语言 帮助		1 2 57 57	
 ③ Doany(v1.0.0) - 没有塗接車口 - 未保存 文件 編輯 差接 控制版 扩属 语言 報助 □ Untitled 		1 + X X	- • • • ×
3 Doany(v1.0.0) - 没有強振申口 - 未保存 文件 編編 连接 控制版 扩展 语言 帮助 日 - 和		1 + X X	
 ⑦ Doany(v1.0.0) - 没有塗銀車口 - 未保存 文件 編編 连接 控制版 扩展 语言 帮助 ① Untitled 		1 + X X	بر الم
⑦ Doany(v1.0.0) - 没有道接車口 - 未保存 文件 編編 连接 控制版 扩属 语言 帮助 ① Untitled	脚本 造型 声音 功作 事件 分別 一次例 方面 (公園) 西端 数字和逻辑运算	± + X X	م ال
⑦ Doany(v1.0.0) - 没有连接車口 - 未保存 文件 編編 连接 控制版 扩展 店首 帮助 ① Untitled	脚本 造型 声音 动作 事件 外观 空句 声音 公元 公元 声音 公元 公元 支管 功能 第 支管 功能 公元 支管 功能 公元	1 + X X	ب ال
 ⑦ Doany(v1.0.0) - 没有连接乗□ - 未保存 文件 編編 连接 控制版 扩展 店首 帮助 ③ Untitled 	脚本 道型 声音 动作 事件 分元 空荷 声音 公式 三端 数字和逻辑运算 数据和田田会 机器人模块 既特的发生特	1 + X X	ب ۵ س ر ۲ -64 ۲ -23
⑦ Doany(v1.0.0) - 没有違接準□ - 未保存 文件 編編 连接 控制版 扩属 语言 帮助 □ Untitled	脚本 速型 声音 动作 事件 分元 一章件 方言 公式 百零 公式 武術和音 公式 武術和音 公式 武術和音 机晶人種块 原作的文型修 原作的文型修	1 + X X	
 ③ Doany(v1.0.0) - 没有違接準日 - 未保存 文件 編編 连接 控制版 扩属 语言 報助 ④ Untitled 	脚本 速型 声音 动作 事件 分元 売荷 万音 公式 西省 公式 武術石田協会 机器人提供 原作的火生時 原作的火生時 原作的火生時	1 + X X	№ № № № № № № №
⑦ Doany(v1.0.0) - 没有違接乗日 - 未保存 文件 編編 连接 控制版 扩展 语言 帮助 ○ ● </th <th>製本 速型 声音 动作 事件 分元 資効 声音 公式 声音 公式 三零 公式 政策和出版会 机器人提供 原作的文生标 回應</th> <th>1 + X X</th> <th>№ № № № № № № № </th>	製本 速型 声音 动作 事件 分元 資効 声音 公式 声音 公式 三零 公式 政策和出版会 机器人提供 原作的文生标 回應	1 + X X	№ № № № № № № №
⑦ Doany(v1.0.0) - 没有違接乗日 - 未保存 文件 編編 连接 控制版 扩展 雪富 報助 ○ ● </th <th>製本 速型 声音 动作 事件 分別 資料 声音 分別 一般 和 日 (秋時的) 型标 日 (税券) 动作、生 日 (税券) 动作、生</th> <th>1 + X X</th> <th>№ № № № № № № № </th>	製本 速型 声音 动作 事件 分別 資料 声音 分別 一般 和 日 (秋時的) 型标 日 (税券) 动作、生 日 (税券) 动作、生	1 + X X	№ № № № № № № №
 ⑦ Doany(v1.0.0) - 没有接接集□ - 未保存 文件 編編 连接 控制版 扩展 香油 報助 □ Untitled 	野本 速型 声音 动作 事件 分別 資料 声音 公式 三号 数字和逻辑运算 社会的 机器人提供 採杯的业标 限标的业标 目 範囲 「採焼炊剤」 动作 * 在 角色 * 上 「将鍋盒头 开启 *	1 + X X	№ № № № № № № №
⑦ Doany(v1.0.0) - 没有違接乗日 - 未保存 文件 等構 连接 控制版 扩展 香油 報助 □ Untitled	野本 速型 声音 动作 事件 分风 投制 声音 公式 三号 双示印服令 武林的公告 机器人提供 採杯的公告 机器人提供 採杯的公告 日 保護体測 以前作 * 在 角色 * 上 将机像点明 探信 子の * 10 %	1 + X X	x -64 y: 23
⑦ Doeny(v1.0.0) - 没有違接申□ - 未保存 文件 等構 差接 控制版 扩展 语言 帮助 ① Untitled	製本 速型 声音 动作 事件 分風 受効 声音 公式 高考 双字和逻辑运算 就能和当你 机器人提供 積積的坐标 机器人提供 積積的坐标 一個度 複類体則 动作 * 在 角色 * 上 停锅像头 开启 * 一 脊視频透明度设置为 *30 % %		₩ 0 ₩ × -64 9. 23
⑦ Doeny(v1.0.0) - 没有違接申□ - 未保存 文件 編輯 差接 控制版 扩属 语言 帮助 ○ Untitled ★ 249	製本 速型 声音 动作 季件 分规 声音 动作 季件 分元 三零 数字和逻辑运算 鼓簧和当零 机器人提块 桶杯的业标 桶杯的业标 机器人提块 一時度 一 1 一 報知休期 动作 「在 角色 」上 一 特報像夫 开启 」 - 特報感情明度设置为 50 % 9 Y: -180 「計開器 1		x -64 y 23
⑦ Doeny(v1.0.0) - 没有違接申□ - 未保存 文件 編編 強接 控制版 扩展 语言 稀助 ① Untitled ● 角色			x -64 y -23
⑦ Doeny(v1.0.0) - 没有違接申□ - 未保存 文件 編編 強接 控制版 扩展 语言 帮助 文件 編編 強接 控制版 扩展 语言 帮助 ① ① ① ① ① ① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	野本 透型 声音 一部件 分別 子市 44 子市 424 日本 日本 三名 秋季和思報法書 秋季和思報法書 秋季和思報法書 武徳和道士 11器人提供 日本 11器人提供 「秋村的工業長 「秋村的工業長 「 11器人提供 「秋村的工業長 「 11器人提供 11器人提供 「秋村的工業長 「 11器人提供 11器人提供 「秋村的工業長 「 11器 11 「 (11) (11) 11 11 「 (11) (11) (11) 11 11 「 (11) (11) (11) (11) 11		2.23
⑦ Doeny(v1.0.0) - 没有違接申□ - 未保存 文件 編編 连接 控制版 扩展 语言 帮助 文件 編編 连接 控制版 扩展 语言 帮助 ① ②	野本 透型 声音 一部件 予約 空時 声音 設計 事件 子言 公式 公式 三名 放き和認識認識 引品人提供 代格的文型体 「日本 一部 「秋秋的文型体 「日本 「日本 「秋秋前秋型体 「日本 「日本 「秋秋秋風田居设置力 ⑤ % 「日日四 「日田四 「日田四 「日田四 「日田四 「日田四 「日本 「日本		μ. D
⑦ Doeny(v1.0.0) - 没有違接申□ - 未保存 文件 編輯 连接 控制版 扩展 语言 帮助 文件 編輯 连接 控制版 扩展 语言 帮助 ① Untitled ● ① ① 角色 新建角色: ◆ / 育合 1 背梁	野本 送型 声音 功作 事件 分別 二 二 一 一 二 一 一 二 一 一 二 一 一 二 一 一 二 一 一 二 一 日 二 一 日 二 一 日 二 一 日 二 一 日 二 一 日 二 一 日 二 一 日 二 一 日 二 日 日 二 1 日 二 1 日 二 1 日 二 1 日 二 1 日 二 1 日 二 1 日 二 1 日 1		₩ - 64 ¥. 23
⑦ Doany(v1.0.0) - 没有違振申□ - 未保存 文件 編編 连接 控制版 扩展 雪音 帮助 ① Untitled / / / / / / / / / / / / / / / / / / /	▶ 認型 声音 予件 予約 声音 ● 予件 予約 一 第件 予約 一 第件 予約 一 第件 予約 一 第件 予約 で 数字和変級選邦 可認 一 報題 一 報 一 報 一 報 一 報 一 報 一 報 一 報 一 報		μ D Δ Δ μ - 64 μ - 23

碰撞模块

碰撞设计是一些游戏中必不可少的功能,在软件中对应的积木有以下几个:



第一块积木可以侦测是否碰到鼠标,边缘,其他角色,第二块积木侦测是否碰到什么颜 色,第三块积木侦测本角色的颜色是不是碰到其他的颜色,第四块积木可以算出到鼠标指针 或其他角色的距离。

用户输入

所谓用户输入就是根据程序要求,输入一些信息,例如人机交互时,我们可以用下面的



输入的内容会被保存在"回答"的积木中,下面就是程序的显示结果:





按键鼠标

在一些程序的运行时,可以通过键盘输入,用鼠标进行交互,在 Doany 软件中的积木 有以下几个:



第一块积木是侦测用户是否按下某个按键,第二块积木侦测到鼠标的左键和右键,最后 两块积木显示鼠标的当前所在位置。

音频和视频

Doany 的交互还可以使用音频和视频(需要硬件支持):



第一块积木会获得麦克风的响度, 第二块是视频交互的关键, 它有四种排列方式:



动作是视频在舞台/角色中移动幅度的表现,移动幅度越大,动作值越大,方向是视频 在舞台/角色中移动方向的表现,右方向和上方向是正数,左方向和下方向是负数。 第三块积木可以将摄像头进行开启和关闭,第四块积木设置舞台中摄像头的透明度,0为完 全不透明,100完全透明。

计时器

计时器就是任何编程都必须提供的功能了, Doany 的计时器功能有点偏弱, 不仅只有一个计时器, 而且无法停止, 当然它作为入门级的编程语言, 仅使用以下这两块积木是可以满



使用计时器通常要将它归零, 你会

发现每次将其归零,都会从0开始计时,因为无法停止,所以需要使用变量记录时间或者使



获取属性和时间

每一个角色都有许多属性,如坐标位置,方向大小,当前造型等。如果需要从一个角色 去访问另一个角色中的属性,就可以使用下面的积木来完成:



一些时间相关的积木如下所示:



积木本身很简单,基本看一眼就知道怎么去用,可以做一些跟时间相关的程序,如日历 和钟表等。但是需要注意的一点就是当前时间信息需要减一才符合中国的习惯。

数字和逻辑运算

数字和逻辑运算指令区

下图红色区域就是数字和逻辑运算指令:

1 Doany(v1.0.0) - 没有连接串口 - 未保存	a - a - C - an -	8 5 6 8488 -	Can Don / - 1 - Dan	
文件 编辑 连接 控制板 扩展 语言 帮助				
📃 Untitled	► ●	脚本 造型 声音	サンズ	
		 动作 事件 外观 控制 		ģ
~		声音 侦测		x: -64 v: 23
		回号 0.5-102 H C H 数据和指令 机器人模块		
()		<u>••</u>		
8-8				
U -		在 1 到 10 间随机洗一个数		
	x: 212 y: 67	T		
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	新建角色: 💙 / 🖴 🖸			
100 N				
舞台 1 背景		不成立		
		合并 hello 与 world		Q = Q
ICIA T				
⑦ Doany(v1.0.0) - 没有连接串□ - 未保存	1111-11-11-11	STATE SHOULD	10 8 8 M	- • ×
⑦ Doany(v1.0.0) - 没有连接申口 - 未保存 文件 编辑 连接 控制板 扩展 语言 帮助	4437-2-2-8		1+XX	
砂 Doany(v1.0.0) - 没有连接申日 - 未保存 文件 编辑 连接 控制版 扩属 语言 帮助 Image: Comparison of the state of the	~●	 脚本 造型 声音 動作 	1 + X X	- C ×
 3 Doany(v1.0.0) - 没有法規準日 - 未保存 文件 編編 连接 控制版 扩展 语言 等助 (一) Untitled 	× • • •	脚本 遠迎 声音 助作 事件 外风 控制 声音 仮測	1 4 X X	
 ⑦ Doany(v1.0.0) - 没有连接車口 - 未保存 文件 編編 连接 控制版 扩展 语言 等助 ④ Untitled 	~ •	評本 透型 声音 动作 事件 外观 控制 声音 侦测 声音 侦测 声音 ····································	± + X X	x -64 y 23
⑦ Doony(v1.0.0) - 没有法想申□ - 未保存 文件 編編 连接 控制版 扩展 语言 帮助 □ Untitled	~ •	輝本 道型 声音 动作 事件 外況 注列 声音 位則 一番 位則 ● 日 ● 日 ● 日 ● 日 ● 日 ● 日 ● 日 ● 日 ● 日 ● 日 ● 日 ● 日 ● 日 ● 日 </td <td>1 + X X</td> <td>x -64 y: 23</td>	1 + X X	x -64 y: 23
 Doany(v1.0.0) - 没有法規準日 - 未保存 文件 編編 連接 控制版 扩展 语言 等助 ① Untitled 	~ •	脚本 道型 声音 动作 事件 分成 控制 声音 (成画) 影響 1000000000000000000000000000000000000	± + X X	рокалана ж64 у. 23
 Doany(v1.0.0) - 没有连续单口 - 未保存 文件 編編 連接 控制版 扩展 语言 等助 Untitled 	~ •	脚本 速型 声音 功作 事件 分別 位利 声音 位利 声音 位利 声音 人名中 数面 人名中 改進 小園 支援 小園 支援 小園 支援 小園	± + × ×	2 - 64 y: -64 y: 23
 Doany(v1.0.0) - 没有连接車口 - 未保存 文件 編編 連接 控制版 扩展 语言 等助 ① Untitled 	~~●	輝本 道型 声音 功作 事件 分和 逆刺 声音 近期 声音 近期 声音 近期 声音 近期 声音 近期 直端 数字和2出版算 改動和2当後 第二年 支援和2当後 1 直 重 支 1 直 重 支 1	1 + X X	x -64 y; 23
⑦ Doany(v1.0.0) - 没有法規準□ - 未保存 文件 編編 連接 控制版 扩展 语言 奉助 ● Untitled	~ •	脚本 道型 声音 功作 事件 分パ 空利 声音 (近周) 声音 (近周) 三端 (女子和送福伝会) 武法和指令 引品人提供 1 日 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 東 - 市 - 東 - 東 - 東 - 東 - 東 - 東 - 日 - </th <th>* + X X</th> <th>x -64 y: 23</th>	* + X X	x -64 y: 23
 Doany(v1.0.0) - 没有连续单口 - 未保存 文件 编辑 连接 控制版 扩展 语言 等助 ① Untitled 	~ •	レージー 第年 初作 事件 分元 空句 声音 (公司) 三昭 分子和送信公式 文式研究日報令 引起人機快 3 ・ 日 ・ 万 ・ 日 ・ ○ ・ </th <th>1 + X X</th> <th>2 - 64 y: 23</th>	1 + X X	2 - 64 y: 23
 Doany(v1.0.0) - 没有连接車口 - 未保存 文件 编辑 连接 控制版 扩展 语言 等助 ① Untitled 	× 240 tr -150	野本 送型 声音 功作 事件 分元 位刻 声音 公別 三部 石字40送相区封 文第 石泉山 夏 夏 夏 <t< th=""><th>1 + X X</th><th>\$ x -64 y 23</th></t<>	1 + X X	\$ x -64 y 23
 Doany(v1.0.0) - 没有连接車口 - 未保存 文件 編編 連接 控制版 扩展 语言 等助 ① Untitled 	★ 240 y: -168 新建角色: ◆ / ▲ @	評本 道型 声音 功作 事件 分元 空利 分元 空利 第 (分) 第 (分) 第 (分) 1 (分)	▲ ← X X	x -64 y: 23
 Doany(v1.0.0) - 没有连接車口 - 未保存 文件 编辑 连接 控制版 扩展 语言 蒂助 ① Untitled 	×: 240 y: -168 新建角色: ◆ / 🏠 🙆	レージャン・ 正正 声音 功作 事件 分元 空話 分元 空話 第二 (1) 2 1) 2 1) 2 1) 2 1) 2 1) 2 1) 2 1) 2 1) 2 1) 2 1) 2 1) 3) (1) 4) (1) 4) (1) 5) (1) 4) (1) 4) (1) 4) (1) 5) (1) 5) (1) 5) (1) 5) (1) 5) (1) 5) (1) 5) (1) 5) (1) 5) (1) 5) (1) 5) (1)	★ ★ X X	x -64 y 23
 Doany(v1.0.0) - 没有连续单目 - 未保存 文件 編編 連接 控制版 扩展 语言 等助 ① Untitled 	x: 240 y: -168 新建角色: ◆ / 🏠 💽	レージー 正常 初作 事件 分元 空向 声音 (公司) 声音 (公司) 声音 (公司) 支援 (公司) 支援 (公司) 支援 (公司) 支援 (公司) 支援 (公司) (公司) (公司)	± + x x	2 - 64 5 - 23
 Doany(v1.0.0) - 没有连续单目 - 未保存 文件 编辑 连接 控制版 扩展 语言 导助 ① Untitled 	★: 240 y: -168 新建角色: ◆ / ▲ ○	詳本 透照 声音 功作 事件 分元 位利 声音 位利 声音 位利 方面 位利 支援 分子和反告に広身 支援 万子和反告に広身 支援 万子和反告に広身 支援 万子和反告に広身 支援 万余石: 支援 小子石: 大学石: 小子石: 支援 小子石: 支援 小子石: 支援 小子石: 支援 小子石: 支援 小子石: 支援 小子石: 大学石: 小子石: 大学石: 小子石: 大学石: 小子石: 大学石: 小子石: 大学石: 小子石:	▲ + X X	x -64 y 23

算数运算

算数运算估计就是加减乘除和取余数:



比较简单就不在叙述了。

逻辑运算符

我们的生活中,无时无刻都充满了逻辑运算,本软件有三个逻辑:



且运算就是要两个都成立才能进行下面积木的程序,或运算就是只要一个成立就可以进 行下面的积木运算,不成立运算就是如果需要按下,而你没按下就成立,会进行下面积木的 运算。可以通过下面程序进行了解。



关系运算符

关系运算符比较重要,在一些逻辑逻辑转换到计算机语言中,本质上都在比较数值的大小,等于关系。以下为关系运算符的积木,比较简单,就不详细阐述了。



数学函数

数学函数对于程序非常重要,我们可以使用它进行很多计算:



随机模块的应用场景比较多,如设置随机时间,随机号码等,此积木可以生成连续的随 机整数,小数等,比较简单。

四舍五入模块,使用非常简单,默认把小数四舍五入成整数位:



像一些数学函数用的最多的是第三块模块:



字符串处理

字符串即数字,字母,符号,的集合,字符串也可以进行许多操作,可以计算字符串的 长度啊,字符串之间可以进行比较,下面讲一下字符串的应用。



字符串的长度即字符串中包含字符的数量,当我们需要遍历字符串中的每一个字符时就 需要使用第二和第三块积木了,字符串的合并也比较简单,如下所示:

				hello world
合并	hello	与	world	

机器人模块

机器人模块指令区

下图中红色区域就是接下来要介绍本软件的最后一个机器人模块指令:

😚 Doany(v1.0.0) - 没有连接串口 - 未保存	· K K & T - 2-2-21	SW-T-B SHOP AN	at a second the second second	
文件 编辑 连接 控制板 扩展 语言 帮助	助			
Untitled	> •	御本 造型 声音	1 + X X	
R		 动作 事件 外別 空制 声音 (効測) 画画 数字和逻辑运 数葉和指令 机器人感染 Rainbot ▼ 	H ■ ■	фт ж. 64 у. 23
		Rainbot 主程序 前进 ¥ 转速为 ① 设置电机 电机接口17 转速为 ①		
8.6.	x: 240 y: -37	(注意184、(福二)→(福美)→ 用度 CO (注意(表式LED (福二)) 全部) 紅色 OP # 色 (後面LED (福二)) 全部) 紅色 OP # (冷声灯音 (福二)) (金部) 紅色 OP #		
海台 1 背景 新連辞業	anserto. Y / W	播放音调为 C47 节拍力 C分2-7 表情面板 (2011) 显示数字: 0 表情面板 (2011) エテ数字: 0 素情面板 (2011) x: 0 y: 0 显示	×2: 11	
		表情面被 (据11) 显示时间: 10 时	: 【20 分	q = q



连接

我们需要将硬件和软件进行连接,连接方式有 USB 数据线,蓝牙,和 2.4G,首先选择主 控板和电脑连接,选择 COM 端口。



如果设备端口找不到端口号,先安装 Arduino 驱动后在查找并选择端口:



Doany 成功的连接了主控板,但是他们之间还不能进行通信,所以我们需要将协议安装 在主板上,选择当前主控板,在选择安装固件:



下面讲解蓝牙的连接方法,打开电脑的蓝牙开关,断开之前的 COM 端口连接,在菜单 栏中选择蓝牙,发现,然后出现蓝牙列表,选择对应的蓝牙,连接就好了(当蓝牙未连接时, 蓝牙模块蓝灯闪烁,连接成功后,蓝牙模块蓝灯常亮)。

2.4G 目前仅支持 Rainbot 小车,其操作简单,将 2.4G 模块连接到主板上。 2.4G 有两种模式

第一种是慢闪模式: 2.4G 连接到主控板,上电,2.4G 模块蓝灯会缓慢闪烁,说明只能 跟上一次适配器配对,不能跟新的配对。

第二种是快闪模式: 2.4G 连接到主控板,上电,2.4G 模块蓝灯会快速闪烁,就会立即 连接,比较方便。

配对成功后, 蓝灯会常亮, 选择 2.4G 就可以无线通信了。

👌 Doany(v1.0.0)	- 没有连接串口	- 未保存					
文件 编辑	員」注	妾 控制板 扩展	建 语言	帮助				
· .		串口		+		日本 御本	造型	声音
	Ur	蓝牙			r			
		2.4G无线串口		•	连接	云乃代年		争件
		网络		· [外观		控制
		安装固件 恢复出厂程序 设置固件模式 查看源代码 安装Arduino题	রকা)- -		声音 画笔 数据 音里 限位 温度	和指令 专感器(接 开关(接口 专感器(接	(次列) 数字和逻辑运算 机器人模块 1137 119 (插头17) 1137 (插头17) °C
		~ \$				人体 陀螺 温湿 火焰 气体 电子	红外传感: 文 文袖 夏 传感器 专感器 接 要 集 集 集 集 集 集 集 集 集 集 集 集 集	器 (接口2) 角度 (接口1) (温度) (注3)

编程

机器人模块中的个部分指令都是在菜单中选择的,下图选择了 Doany 主控板:

B Doany(v1.0.0) - 没有连接串口 - 未保存				
文件编辑 连接 控制板 打	展 语言 帮助			
Untitled	扩展管理器 Ctrl+Shift+T 还原出厂扩展 清空缓存			
	微软认知服务设置			
<u></u>	Microsoft Cognitive Services Arduino Joystick(Arduino Mode Only)			
	Doany			
	Communication			

还可以选择一些 Arduino 系列的主控板, Joystick 系列的, 根据自己的需求进行编程, 达到比较好的效果。

上传到主板

每一个程序编程过后就需要将程序上传到主板,这样就能够进行脱机工作了,主要过程就是将积木模块的程序转化成 Arduino 代码,直接烧录到主板上,就可以了。



上面就是将程序上传到主板了,非常的简单,希望大家能够多多学习。