

1. 认识积木

1.1 大家好，我是小白，不姓江

2. 星星点灯

小结：

3. 会呼吸的灯

4. 开关控制LED

4.1 开关控制1个灯

4.2 开关控制灯的亮度

5. 能不能变成声音？

5.1 我有一个好主意-音乐盒

6. LED点阵-T_T我是显示屏

6.1 点亮1盏灯

6.2 点亮1列/1行灯

6.3 全亮

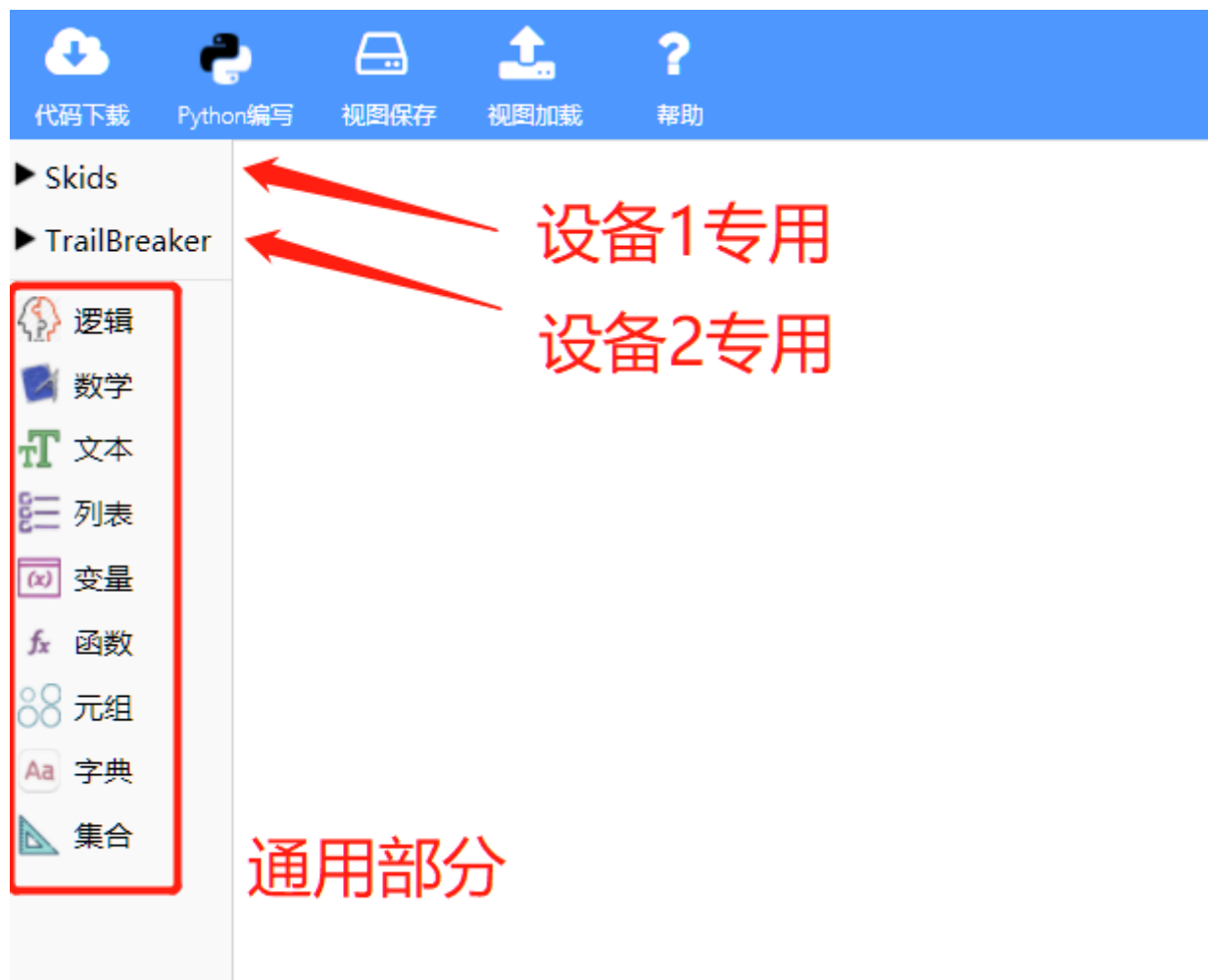
6.4 画个图

Blockly网址<http://www.neuai.net/neuaibit/#>

1. 认识积木

Skids/TB是与硬件相关的模块

逻辑、数学、文本、列表、变量、函数



...

1.1 大家好，我是小白，不姓江

大家在学新东西时，总会觉得有难度，因为没有接触过，因为是小白。这里想说的是不能被所学的**内容吓到**，而是去寻找**兴趣点**，大家都在说**兴趣是最好的老师**。

我们会通过一个个小小的实例的实现过程，从“我认识它，它不认识我”，转变到大部分功能都掌握。

2. 星星点灯

怎么点亮一盏灯？
案。

选中后看答

我们从积木中找到一个指示灯的“打开”。



终端：

复位：CTRL+C(先按下ctrl并保持，然后按下C)

启动：CTRL+D(先按下ctrl并保持，然后按下D)

但是LED并没有点亮，还有报错，为什么？

先来创建1个灯。



先创建，后使用

然后我们把灯关掉，直接拖拽1个“关闭LED”，可以发现LED直接灭了，并没有看到亮。为什么？

(选中看)

怎么解决呢

(选中看)



程序执行是飞快的，有多快？开灯不到1ms

我来试一试

(选中看):

灯同时打开，同时关闭

一个一个打开，一个一个关闭

小作业：4个灯，同时只亮一个，依次从左到右，然后从右到左。

我有一个好主意

(选中看):

逐个点亮

小结:

终端的字是怎么回事? 程序猿会说, **日志**

这个日志就是工作日志, 不同的是这个是机器的,
是我们写的代码的

补充:

用日志的方式做数学运算:

$1+1$, $457687+12879812$

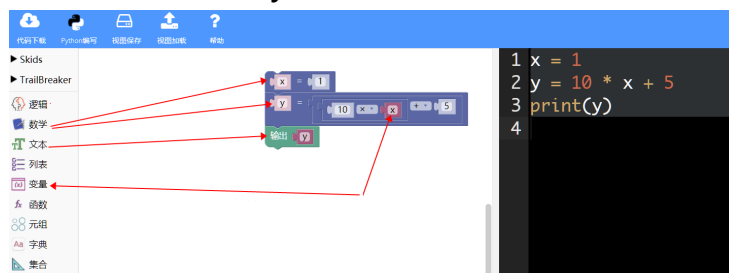
$1>1$,

用日志的方式做文字的运算:

拼接

用积木求解方程:

$y = a * x + b$, 已知 x , 求 y , 其中 $a=10$, $b = 5$



3. 会呼吸的灯

亮一次、灭一次、亮一次、灭一次...

咦, 教程没了...

教程还是有的, 只不过之后的每一步将都是从无到有的创造过程, 通过从无到有的过程来感受程序员的思

维。

当亮、灭的时间和小于25ms的时候，灯变的不闪了，这是因为“余晖效应”

LED亮的时间在25ms中占的比例越大，灯就越亮。

目标：实现一个呼吸灯

让LED灯逐渐变亮，然后逐渐变灭。

如何实现：

让LED灯亮的时间逐渐增加，让灭的时间逐渐减少。当时总的时间应当是25ms。

1. 创建1个x，让 $x=1$
2. 让灯亮x毫秒，灭 $(25-x)$ 毫秒
3. 让x变大
4. 重复2-3



现象1：LED等在很短的时间内就从最暗，变到了最亮。这个时间大约是650毫秒。

问题1：时间太短，达不到呼吸灯效果。希望从最暗到最亮的这个过程的时间为5秒。

分析、解决问题：可以让x在某个值的时候多执行几个循环（灯亮x毫秒，灭（25-x）毫秒），5秒=5000毫秒，一个亮灭周期是25毫秒，整个从暗到亮的变化过程持续5秒，需要在每个x值位置停留200毫秒，即执行8次（灯亮x毫秒，灭（25-x）毫秒）循环。

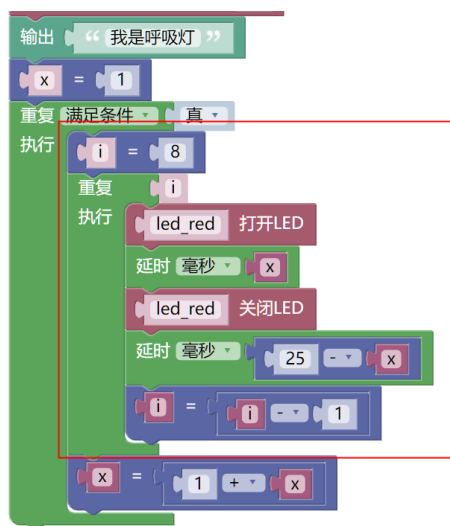
1. 创建1个x，让x=1

2. 让灯亮x毫秒，灭（25-x）毫秒

2.1 重复2，8次

3. 让x变大

4. 重复2-3



知识点：重复执行 i 的逻辑是当i不等于0时重复执行，当i等于0时不再重复。

注意点：i和x都是可以改变的，叫做变量，变量需要先定义，后使用（先创建，后使用）。

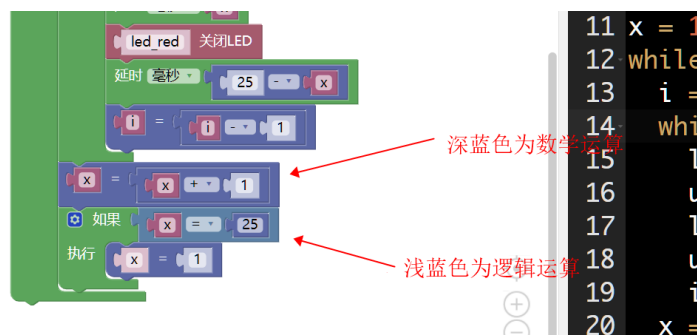
现象2：LED灯的由灭变亮的过程只执行了1次。

问题2：达不到呼吸灯效果，希望从灭到亮，然后从亮到灭。至少从灭到亮，然后再从灭到亮。

分析与解决：当x变大到25时，让它从1重新开始。

1. 创建1个x，让x=1
2. 让灯亮x毫秒，灭 (25-x) 毫秒
- 2.1 重复2，8次
3. 让x变大
- 3.1 当x>24时，让x=1
4. 重复2-3

当x>24时可描述为**如果x大于24**让**x=1**，这里x大于2是执行x=1的**条件为真**。



问题3：如何实现从灭到亮，然后从亮到灭。先实现从亮到灭。

分析与解决：前面已经实现了从灭到亮，如何实现从亮到灭呢？

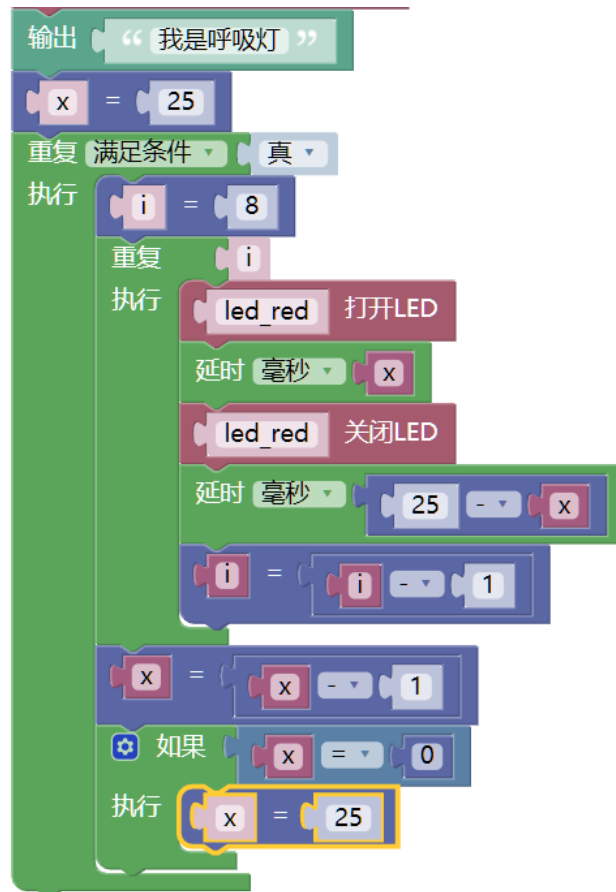
1. 创建1个x，让x=25
2. 让灯亮x毫秒，灭 (25-x) 毫秒

2.1 重复2, 8次

3. 让x变~~小~~

3.1 当x等与0时, 让x=25

4. 重复2-3



分析与解决：如何把逐渐变亮和逐渐变暗这2个过程拼接起来。

思路1：把逐渐变亮和逐渐变暗进行拼接。



思路2：上山与下山的思路，如果有办法让我们知道灯是在变亮还是在变暗，那我们就可以根据这个开控制x是变大，还是变小。

1. 创建1个x，让x=25

1.1 创建1个dir，dir=1，表示加1；dir=0，表示减1

2. 让灯亮x毫秒，灭（25-x）毫秒

2.1 重复2，8次

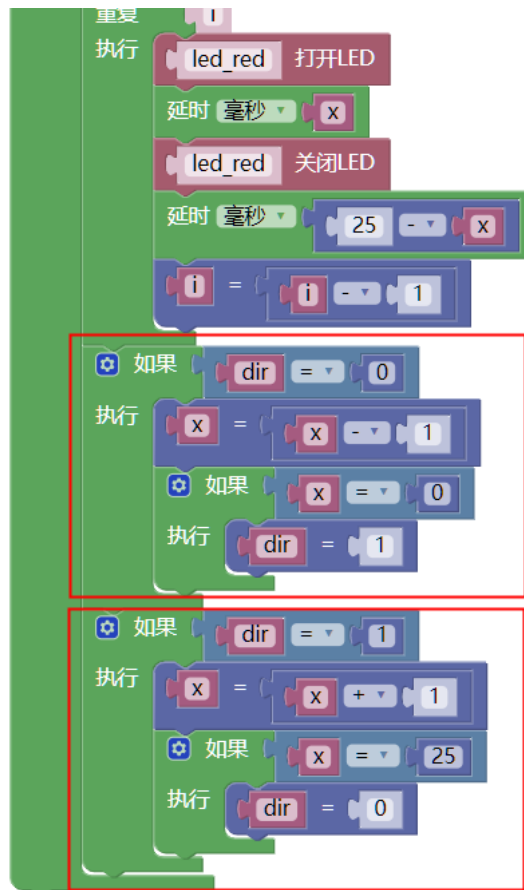
3. 如果dir是0，让x变小

3.1 如果x等于0，改变方向，让dir=1

4. 如果dir是1，让x变大

4.1 如果x等于25，改变方向，让dir=0

5. 重复2-4



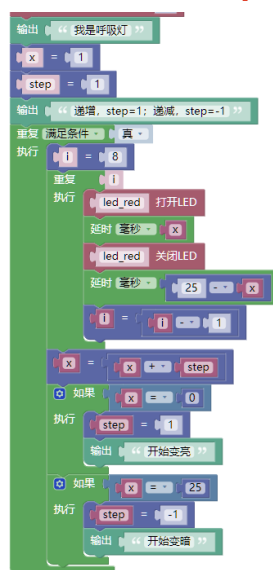
这里的dir在程序中是一个会变化的量，叫做变量，还有哪些是可以变化的呢？

思路3：用1个变量step来表示x的变化量，那么 $x = x + \text{step}$ ，我们可以控制step为1或者-1，进而控制灯变亮还是变暗。

3. 让x变化step ($x = x + \text{step}$)

3.1 如果x等于0，让step等于1

3.2 如果x等于25，让step等于-1



平冤：

妻子对丈夫说，下班时路上**买1斤**黄瓜，**如果**看到的西瓜**买1个**。

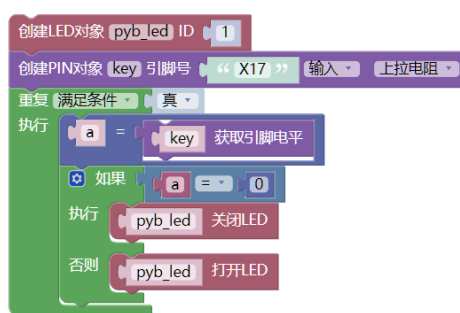
小结：原来灯泡是可以这样调亮度的，难怪有闪频

4. 开关控制LED

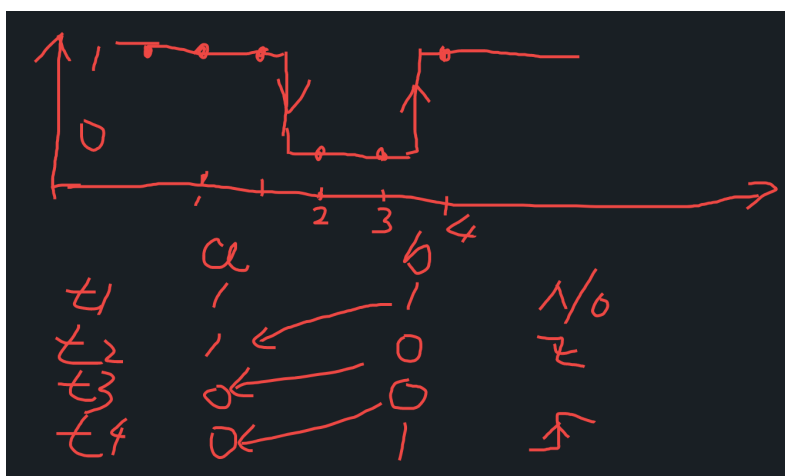
4.1 开关控制1个灯

实现步骤：

1. 创建1个灯，创建1个开关/按键
2. 看按键是不是被按下
3. 如果按下，关灯
4. 否则，开灯



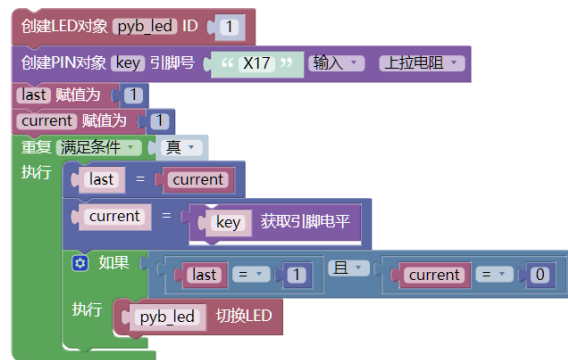
问题：如何用按键按1次改变1下灯的亮灭，当灯亮时，按下开关灯变灭，当灯灭时，按下开关灯变亮。



如何判断按键被按下？

1. 定义2个变量a,b (a=last, b=current)

2. a总是按键的上一个状态，b总是按键的当前状态
3. 当a等于1且b等于0，可识别为按键被按下



4.2 开关控制灯的亮度

5. 能不能变成声音？

想法：既然能控制灯的亮度，和闪烁快慢，那么是不是可以用同样的方法控制来控制1个发声装置，从而实现一个音乐盒呢？

概念1：闪烁的快慢可以用频率表示

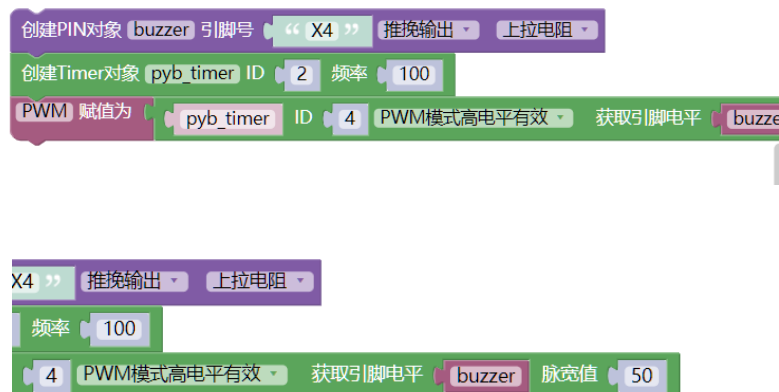
概念2：频率相同，改变亮灭时间的不同比例，这个叫做PWM调试

实现思路：不同的频率对应不同的声音，只要在适当的时候改变为特定的频率即可实现音乐盒。

5.1 我有一个好主意-音乐盒

实现步骤：

1. 创建1个管脚
2. 创建定时器
3. 改变频率



改变频率试一试：

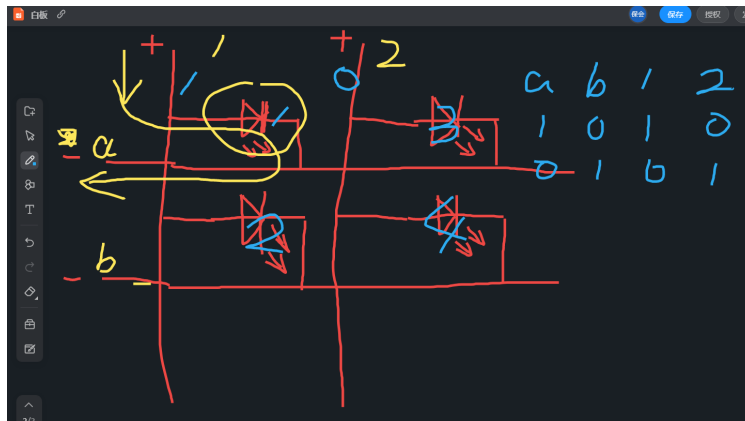
1234567的频率分别是261.63Hz 293.66Hz 329.63Hz 349.23Hz 392.00Hz 440.00Hz 493.88Hz

1. 261.63Hz

每秒钟改变一次频率，从1~7循环

6. LED点阵-T_T我是显示屏

原理：



1. 每次点亮一行，保持20ms，然后熄灭
2. 再点亮下一行，保持20ms，然后熄灭
3. 不断循环这个过程，根据闪屏情况可减少保持时长。

前一天涉及内容巩固：

先创建后使用

逻辑运算

数学运算

条件执行：如果。。。就。。。否则。。。； 左转/右转/直行，还是停止、条件是红绿灯

循环执行：不停执行。。。 while/for

执行多少次：

变量：线性方程求解

求解0~100之间的素数

百元买鸡问题

6.1 点亮1盏灯

1. 创建引脚
2. 引脚设置初值，即**初始化**，就像运动员预备状态

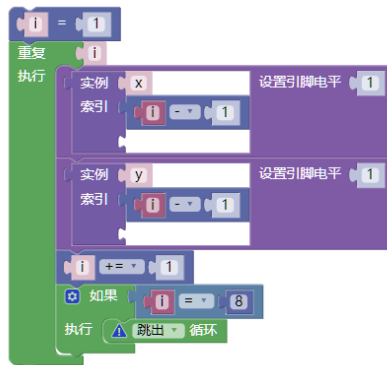
3. 第一行，第一列分别设置为0

我完成1行的点亮，学生完成1列的点亮

6.2 点亮1列/1行灯



6.2.1 初始化部分优化1



6.2.2 初始化部分优化2



6.3 全亮

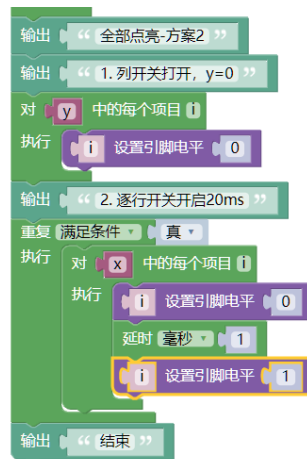
方案1：所有开关闭合，即所有引脚配置为0



问题：处理器带负载能力不足，长时间工作会有风险。

分析与解决：利用余晖效应，实现一开始的方案

1. 每次点亮一行，保持 1ms，然后熄灭
2. 再点亮下一行，保持 1ms，然后熄灭
3. 不断循环这个过程，根据闪屏情况可减少保持时长。



求解1~100之间的素数：

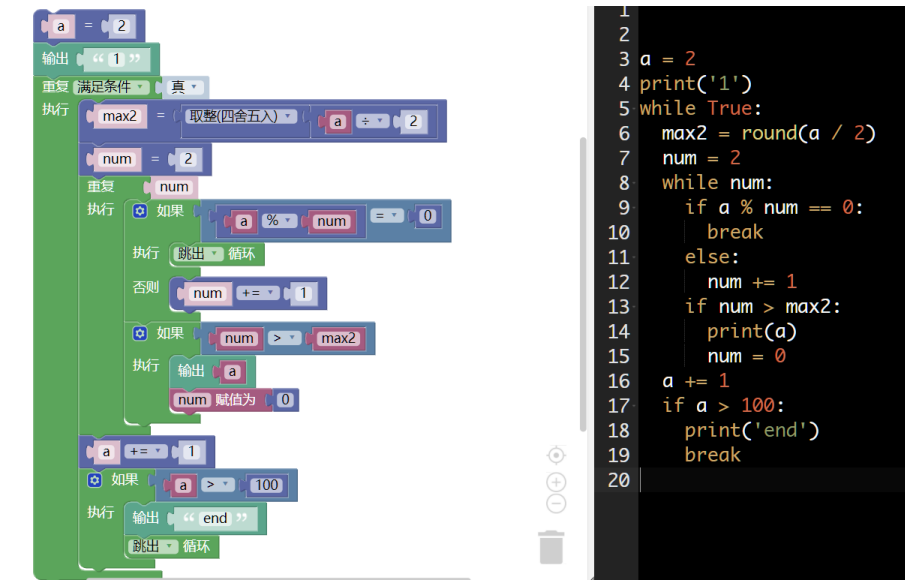
素数一般指**质数**。**质数**是指在大于1的自然数中，除了1和它本身以外不再有其他因数的自然数。

思路：

1. 判断1个数是不是素数

1.1 如何判断，假如判断的数字是a，a对2~a之间的每个数进行取余，如果有余数为零则此数不是素数。

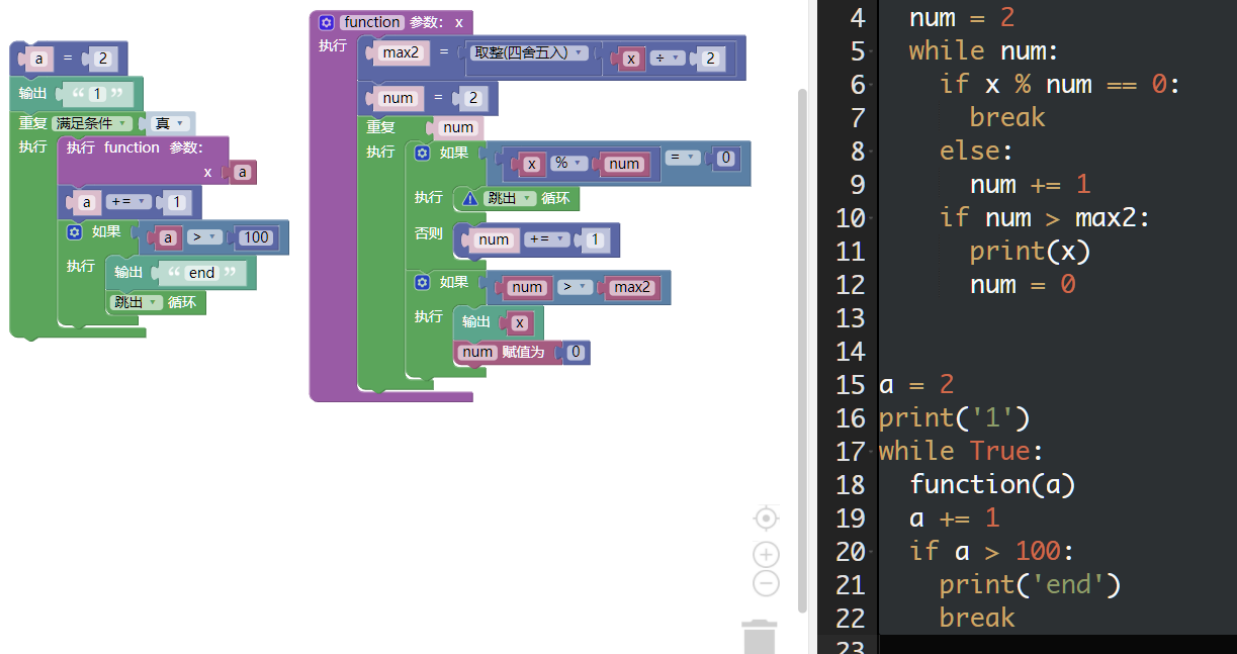
2. 判断1~100之间的每个数



思考：如何找出0~100之间的偶数，如何找出0-100之间的奇数

函数的输入和输出：就像是让小孩打酱油，**输入参数**是10块钱，**输出返回值**是1瓶酱油，打酱油这个过程可认为是函数。

函数的使用：把能够完成特定功能的积木块/程序放在一起，形成一个新的积木块，可在其他地方使用。



基本的打法：

1. 逻辑选择，条件执行/条件语句
2. 循环执行，满足条件时，不断执行
3. 数学运算，逻辑运算（真假判断）

4. 变量，由大小写字母与数字或下划线组合而成，第一个不能是数字。如 9ab是错误的，a9b和ab9都是对的。常数，一个具体的数字

抽奖：

1. 获取随机数
2. 避免重复，新随机数与老随机数进行不对，如果相同丢弃。
3. 判断已抽出人数

百元买鸡问题

公鸡一只5元，母鸡一只3元，小鸡3只1元。

如何用100元买100只鸡。其中公鸡，母鸡，小鸡的数量各是多少

优化：公鸡最多： $100/5=20$ 只

优化：母鸡最多： $100/3 = 33$ 只

设定3个变量：x,y,z分别代表公鸡、母鸡、小鸡的数量。

人民币总和= $5*x + 3*y + z/3$

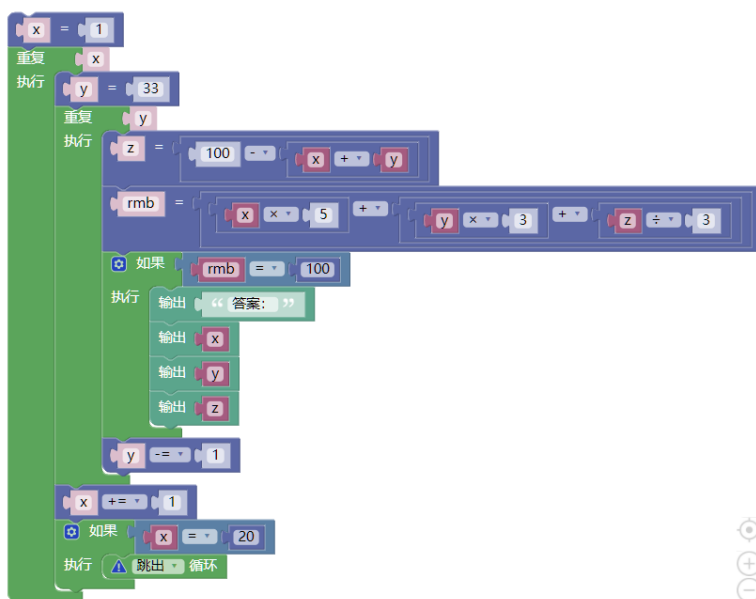
$x+y+z = 100$

当 $x = 1$ 时 $y = 33$ 时 $z = 100 - 1 - 1$ 判断金额是否为100

当 $y = 32$ 时

...

当 $x = 2$ 时 $y = 33$...



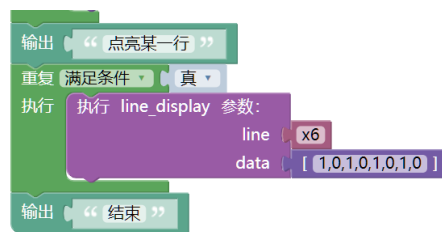
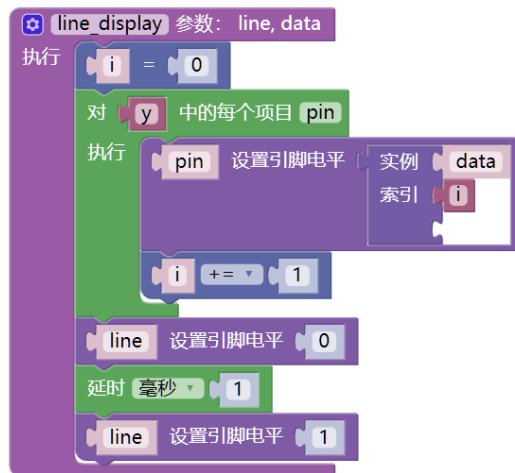
6.4 画个图

如果全部点亮LED的做法是：

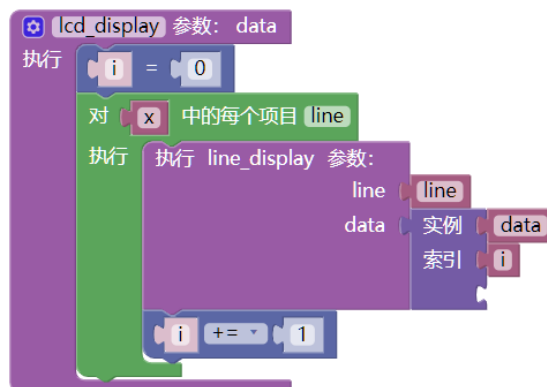
1. 所有行、所有列断开，这个过程叫做初始化
2. 所有列闭合，打开某一行，并保持1ms，这里称作“行扫描”
3. 用2的方法重复每一行

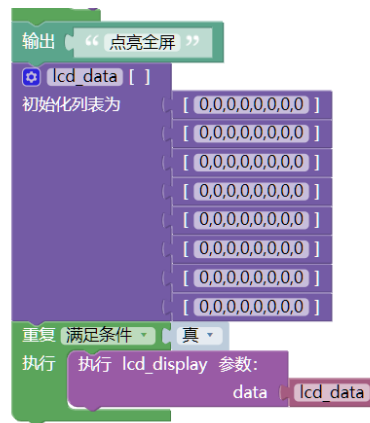
上面的第2条是最关键的，如果不同行用不同的内容进行扫描，这里的内容是指不同列。比如，第一行让第一列亮，第二行让第二列亮。。。

6.4.1 line_display



6.4.2 能不能用同样的方法完成全屏刷新





6.5 让画面动起来

思路：

把第二行内容给第一行，第三行内容给第二行，如此循环，第一行给第8行

实现：

1. 用“对 x 中的每一个项目 i 执行”模块循环处理每一个列表
2. 循环把 i 给到上一个列表元素

错误分析

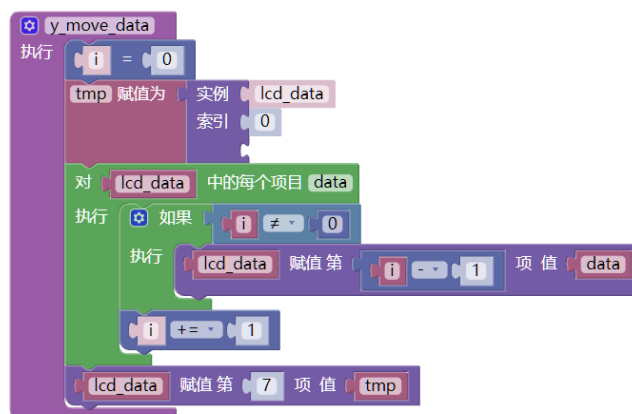
第一次：data = lcd_data[0]，赋值给了lcd_data[1]

第二次：data = lcd_data[1]，赋值给了lcd_data[2]

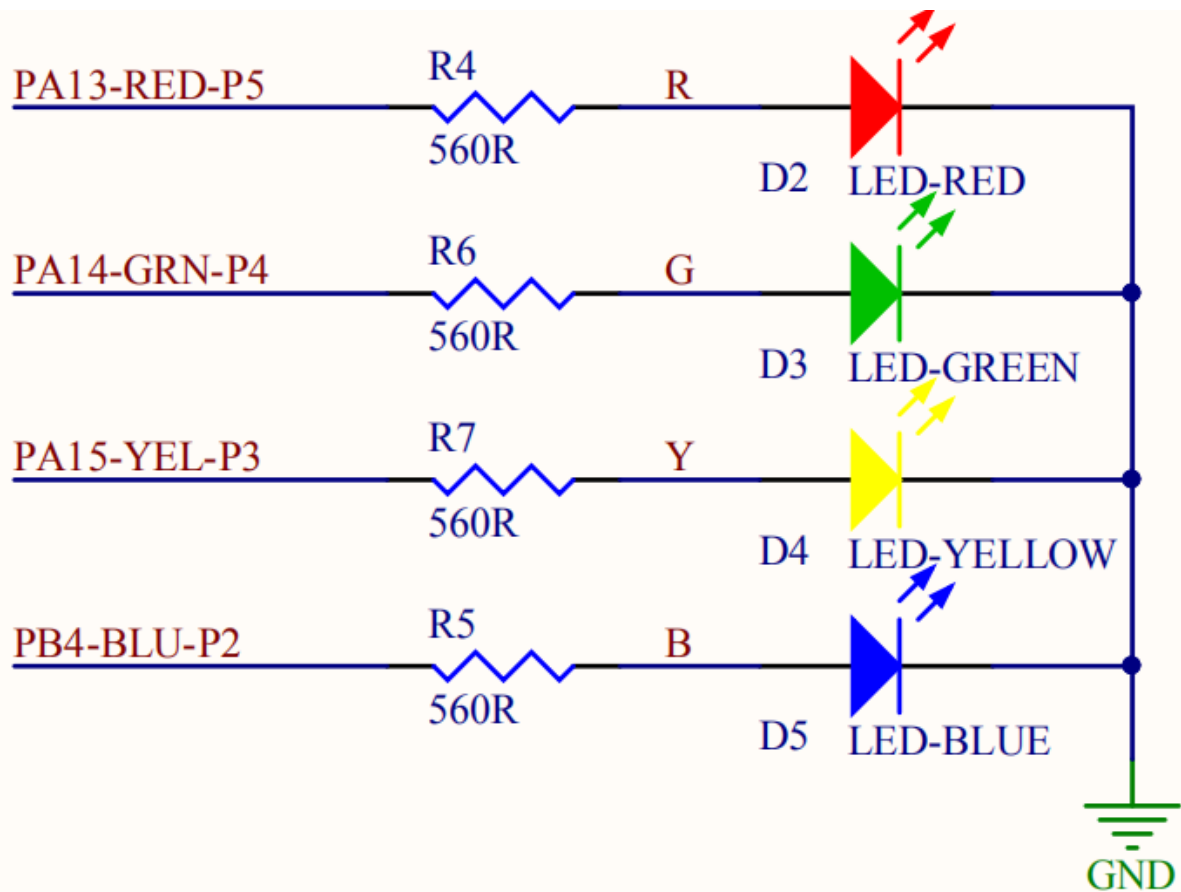
错误解决：

把后面的给前面

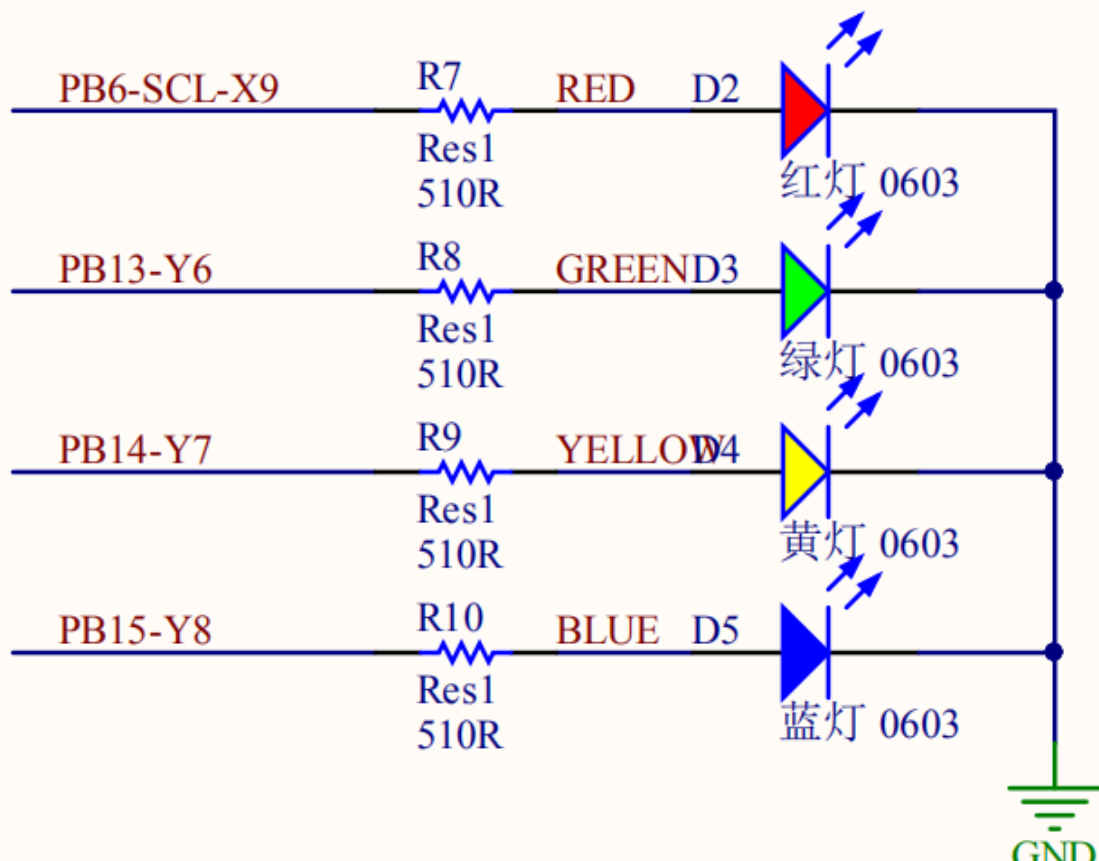
y轴滚动



x轴滚动



LED Board



Skids

