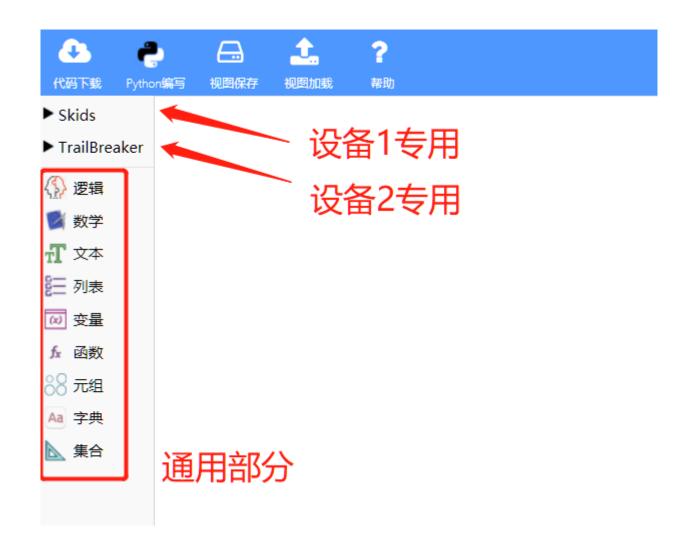
1. 认识积木						
1.1 大家好,我是小白,不姓江						
2. 星星点灯						
小结:						
3. 会呼吸的灯						
4.1 开关控制1个灯						
4.2 开关控制灯的亮度						
5. 能不能变成声音?						
5.1 我有一个好主意-音乐盒						
6. LED点阵-T_T我是显示屏						
6.1 点亮1盏灯						
6.2 点亮1列/1行灯						
6.3 全亮						
6.4 画个图						

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?

选中后看答

案。

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

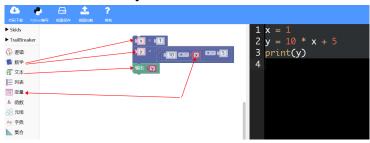
1>1,

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)。

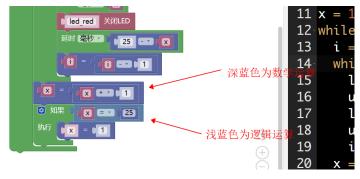
现象2: LED灯的由灭变亮的过程只执行了1次。

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

```
#出 ** 我是呼吸灯 **

** ** **

** **

** **

** **

** **

** **

** **

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**

**
```

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

```
[led_red] 打开LED
    延时(毫秒▼)
             X
           关闭LED
    led red
    延时(毫秒・
             25 - V (X)
     i = (
            i = 1
        dir = v ( 0
执行
     x =
            X - 1 1
    ◎ 如果
            x = v = 0
    执行 (dir) = (1
        dir = v (1)
执行
     X = (
            x + v (1
    □ 如果
            x = v 25
    执行 dir = 00
```

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+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+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<u>买1斤</u>黄瓜,如果看到的 西瓜<u>买1</u>个。

小结:原来灯泡是可以这样调亮度的,难怪有闪频

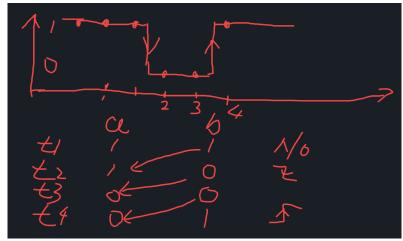
4. 开关控制LED

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

```
创建LED对象 pyb_led ID 1
创建PIN对象 key 引脚号 ( X X17 ) 输入  L拉电阻  L拉电阻  Line  Li
```

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:

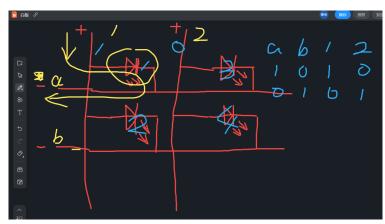
1234567的频率分别是261.63Hz 293.66Hz 329.63Hz 349.23Hz 392.00Hz 440.00Hz 493.88Hz

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

```
総出 ( 全部点 7 )

対 ( y 中的每个项目 ( i 设置引脚电平 ( 0 )

対 ( x 中的每个项目 ( i 设置引脚电平 ( 0 )

延时 ( i 设置引脚电平 ( 0 )

延时 ( i 设置引脚电平 ( 0 )

延时 ( i 设置引脚电平 ( 1 )

扱行 ( i 设置引脚电平 ( 1 )

扱行 ( i 设置引脚电平 ( 1 )
```

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。

```
編出 (** 全部点亮-方案2 **)
輸出 (** 1. 列开关打开, y=0 **)

対 (** 1. 列开关打开, y=0 **)

対 (** 2. 遂行开关开启20ms **)

(** 4. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. ) (** 2. )
```

求解1~100之间的素数:

素数一般指<mark>质数。质数</mark>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

```
a = 12
                                              3 a = 2
                                              4 print('1')
車复 満足条件 ▼ 真 ▼
                                              5 while True:
执行 max2 = 取整(四舍五入) 和 a ÷ v 2
                                                  max2 = round(a / 2)
   num = (2)
   重复 num
执行 ② 如果 a % num
                                                  while num:
                                                   if a % num == 0:
                                              9
                                              10
                                                     break
       执行 跳出 工循环
                                                   else:
                                             12
                                                     num += 1
       if num > max2:
                                                     print(a)
       ② 如果 | num > max2
                                             14
                                             15
                                                     num = 0
       执行 输出 (a)
                                              16
       num 赋值为(0
                                                  if a > 100:
                                             17
                                                   print('end')
                                             18
   a += v 1
                                             19
                                                    break
   9 如果 (a > 100)
执行 输出 ("end"
                                             20
      跳出 • 循环
```

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。

```
function 参数: x
                            max2 = 取整(四舍五入) *
(a) = (2)
                                                x ÷ • 2
                            ( num = (2
重复(满足条件 ) 真 ,
执行 function 参数:
                                num
                                       (X % * (num) = * (0)
              x a
                                执行 🛕 跳出 🔻 循环
   (a += 2 ) 1
    否则 (num += x ) 1
   执行 输出 ( " end "
                               ② 如果 ( num >  max2)
       跳出 * 循环
                                执行 输出 X
                                  num 赋值为 ( 0
```

```
2 def function(x):
 3
     max2 = round(x / 2)
     num = 2
 4
 5
     while num:
 6
       if x \% num == 0:
 7
         break
 8
       else:
 9
         num += 1
10
       if num > max2:
         print(x)
11
12
         num = 0
13
14
15 a = 2
16 print('1')
17 while True:
18
     function(a)
19
     a += 1
20
     if a > 100:
21
       print('end')
22
       break
```

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

```
line_display 参数: line, data
执行
i = 0

对 y 中的每个项目 pin
执行
pin 设置引脚电平 实例 data
索引 i

line 设置引脚电平 0

延时 毫秒 1

line 设置引脚电平 0
```

6.4.2 能不能用同样的方法完成全屏刷新

```
(cd_display) 参数: data
执行 (i = 00)
对 (x 中的每个项目 line
执行 kh行 line_display 参数:
line (line data 实例 data 索引 (i)
```

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

函数调用

```
输出 ( ※ | ※ | ※ | ※ |
o (lcd_data) [ ]
初始化列表为
                     [ (1,1,1,1,1,1,1,1) ]
                     [ (1,1,1,1,1,1,1,1) ]
                    [ (1,1,1,1,1,1,1,1) ]
                     [ (1,1,1,0,0,1,1,1 )
                    [ (1,1,1,0,0,1,1,1) ]
                     [ (1,1,1,1,1,1,1,1) ]
                    [ (1,1,1,1,1,1,1,1) ]
                    [ (1,1,1,1,1,1,1,1) ]
更复 满足条件 · 真 ·
拱行 times 賦值为 100
重复 times
サイプ サイデ はcd_display 参数:
                                  data | Icd_data
              times == 1
      执行 x_move_data
```

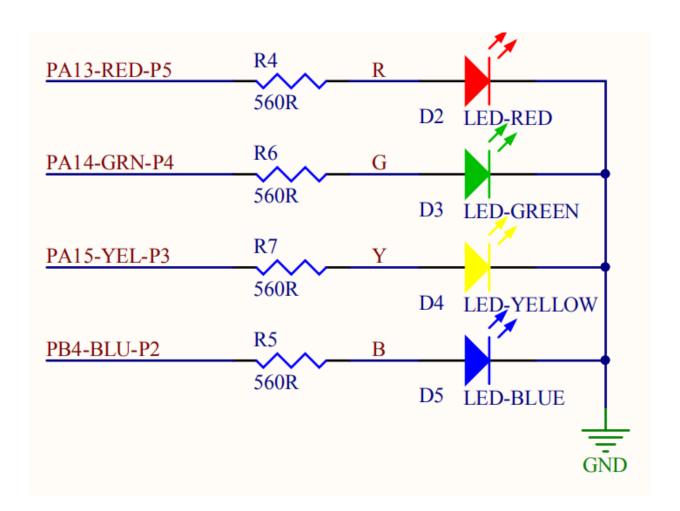
抽奖问题, 我们也来做个抽奖算法

用Blockly计算出0~100之间的素数, 0~1000之间的素数

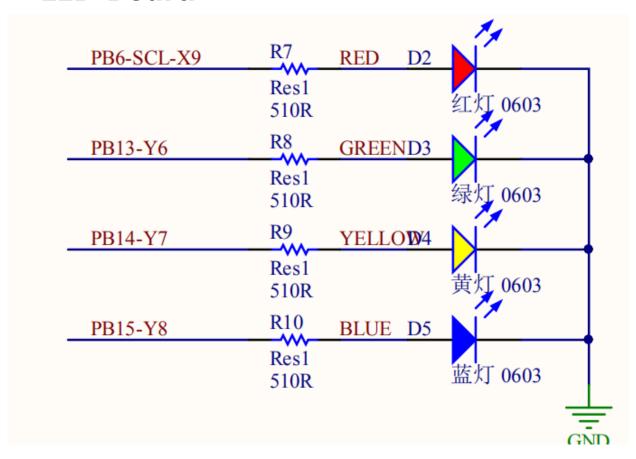
公鸡一只5元,母鸡一只3元,小鸡3只1元。如何用100元买100只鸡。其中公鸡,母鸡,小鸡的数量各是多少

分析:公鸡最多买:20只,母鸡最多买:33只。设:公鸡x只,母鸡y只,小鸡z只

TB



LED Board



Skids

3V3									
J	- V4	2	A.	1 1	RED	R18 0603	1K	IO17_LED1_SDCD_RAMSCLK_3glNT2	
	V5	2		1	GRN	R20 0603	1K	IO22_LED2_ZGB_RX1	
1			1					_	
	V6	2	A.	1	YEL	R22 0603	1K	IO21_LED3_ZGB_TX1	
1	V7	2	\times	1	BLU	R24 0603	1K	IO26_LED4_SDA	